

**IBM DB2 10.1  
for Linux, UNIX, and Windows**

**管理ルーチンおよびビュー  
2013 年 1 月更新版**

**IBM**



**IBM DB2 10.1**  
**for Linux, UNIX, and Windows**

**管理ルーチンおよびビュー**  
**2013 年 1 月更新版**

**IBM**

**ご注意**

本書および本書で紹介する製品をご使用になる前に、1603 ページの『付録 B. 特記事項』に記載されている情報をお読みください。

本書には、IBM の専有情報が含まれています。その情報は、使用許諾条件に基づき提供され、著作権により保護されています。本書に記載される情報には、いかなる製品の保証も含まれていません。また、本書で提供されるいかなる記述も、製品保証として解釈すべきではありません。

IBM 資料は、オンラインでご注文いただくことも、ご自分の国または地域の IBM 担当員を通してお求めいただくこともできます。

- オンラインで資料を注文するには、IBM Publications Center (<http://www.ibm.com/shop/publications/order>) をご利用ください。
- ご自分の国または地域の IBM 担当員を見つけるには、IBM Directory of Worldwide Contacts (<http://www.ibm.com/planetwide/>) をお調べください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： SC27-3865-01  
IBM DB2 10.1  
for Linux, UNIX, and Windows  
Administrative Routines and Views  
Updated January, 2013

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2012.12

© Copyright IBM Corporation 2006, 2013.

# 目次

組み込みルーチンおよびビュー	1
組み込みのルーチンおよびビューをアプリケーションで呼び出す場合のベスト・プラクティス	1
組み込みルーチンとビューを使用するための許可	2
組み込みビューと表関数との比較	3
サポートされる組み込み SQL ルーチンおよび管理ビュー	4
管理ルーチンおよび ADMIN_CMD プロシージャ	25
管理タスク・スケジューラー・ルーチンおよびビュー	311
監査ルーチンおよびプロシージャ	325
自動保守ルーチン	328
共通 SQL API プロシージャ	334
構成ルーチンおよびビュー	388
DB2 pureScale インスタンス情報ルーチンおよびビュー	395
環境ルーチンおよびビュー	404
Explain ルーチン	420
モニター・ルーチン	439
MQSeries ルーチン	783
セキュリティー・ルーチンおよびビュー	806
スナップショット・ルーチンおよびビュー	820
SQL プロシージャ・ルーチン	1094
段階的な再配分ルーチン	1105
ストレージ管理ツール・ルーチン	1115
テキスト検索ルーチン	1119
ワークロード管理ルーチン	1159
その他のルーチンおよびビュー	1205
使用すべきでない SQL 管理ルーチンおよびビュー	1253
ADMIN_GET_DBP_MEM_USAGE 表関数 - インスタンスの合計メモリー消費量の取得	1259
ADMINTABCOMPRESSINFO 管理ビューおよび ADMIN_GET_TAB_COMPRESS_INFO 表関数 (使用すべきでない) - 圧縮された情報を戻す	1261
ADMIN_GET_TAB_COMPRESS_INFO_V97	1266
ADMIN_GET_TAB_INFO_V95 表関数 - 表のサイズおよび状態に関する情報の検索	1273
ADMIN_GET_TAB_INFO_V97 表関数 - 表のサイズおよび状態に関する情報の取得	1280
AM_BASE_RPT_RECOMS - アクティビティ・レポートに関する推奨事項	1288
AM_BASE_RPTS - アクティビティ・イベント・モニター・レポート	1290
AM_DROP_TASK - モニター・タスクの削除	1291
AM_GET_LOCK_CHN_TB - 表形式のアプリケーション・ロック・チェーン・データの検索	1292
AM_GET_LOCK_CHNS - 特定のアプリケーションに関するロック・チェーン情報の検索	1293

AM_GET_LOCK_RPT - アプリケーション・ロックに関する詳細の検索	1294
AM_GET_RPT - アクティビティ・モニター・データの検索	1302
AM_SAVE_TASK - モニター・タスクの作成または変更	1304
APPLICATION_ID	1305
DB_PARTITIONS	1306
GET_DB_CONFIG	1307
GET_DBM_CONFIG	1309
ENV_SYS_RESOURCES 管理ビュー - システム情報を戻す	1310
LOCKS_HELD 管理ビュー - 保持されているロックに関する情報の検索	1313
LOCKWAITS 管理ビュー - 現行のロック待機情報の検索	1316
ヘルス・スナップショット・ルーチン	1320
REG_VARIABLES 管理ビュー - 使用中の DB2 レジストリー設定の検索	1368
SNAPAGENT_MEMORY_POOL 管理ビューおよび SNAP_GET_AGENT_MEMORY_POOL 表関数 - memory_pool 論理データ・グループのスナップショット情報の検索	1370
SNAP_GET_APPL_INFO_V95 表関数 - appl_info 論理データ・グループのスナップショット情報の検索	1375
SNAP_GET_APPL_V95 表関数 - appl 論理データ・グループのスナップショット情報の検索	1382
SNAP_GET_BP_V95 表関数 - bufferpool 論理グループのスナップショット情報の検索	1390
SNAP_GET_CONTAINER_V91 表関数 - tablespace_container 論理データ・グループ・スナップショット情報の検索	1394
SNAPDB_MEMORY_POOL 管理ビューおよび SNAP_GET_DB_MEMORY_POOL 表関数 - データベース・レベルのメモリー使用量情報の検索	1397
SNAP_GET_DBM_V95 表関数 - dbm 論理グループ・スナップショット情報の検索	1402
SNAPDBM_MEMORY_POOL 管理ビューおよび SNAP_GET_DBM_MEMORY_POOL 表関数 - データベース・マネージャー・レベルのメモリー使用量情報の検索	1405
SNAP_GET_DB_V97 表関数 - dbase 論理グループからのスナップショット情報の検索	1409
SNAP_GET_DETAILLOG_V91 表関数 - detail_log 論理データ・グループからのスナップショット情報の取得	1420
SNAP_GET_DYN_SQL_V95 表関数 - dynsql 論理グループのスナップショット情報の検索	1422

SNAPHADR 管理ビューおよび	
SNAP_GET_HADR 表関数 - hadr 論理データ・グループのスナップショット情報の検索 . . .	1426
SNAPLOCK 管理ビューおよび	
SNAP_GET_LOCK 表関数 - lock 論理データ・グループのスナップショット情報の検索 . . .	1431
SNAPLOCKWAIT 管理ビューおよび	
SNAP_GET_LOCKWAIT 表関数 - lockwait 論理データ・グループのスナップショット情報の検索 . . . . .	1438
SNAP_GET_STO_PATHS . . . . .	1445
SNAPSTORAGE_PATHS 管理ビューおよび	
SNAP_GET_STORAGE_PATHS_V97 表関数 - 自動ストレージ・パスの情報の検索 . . . . .	1447
SNAPTAB 管理ビューおよび SNAP_GET_TAB 表関数 - table 論理データ・グループのスナップショット情報の検索 . . . . .	1450
SNAP_GET_TAB_V91 . . . . .	1454
SNAP_GET_TBSP_PART_V97 表関数 - tablespace_nodeinfo 論理データ・グループのスナップショット情報の検索 . . . . .	1457
SNAP_GET_TBSP_V91 . . . . .	1461
SNAPAGENT_MEMORY_POOL 管理ビューおよび SNAP_GET_AGENT_MEMORY_POOL 表関数 - memory_pool 論理データ・グループのスナップショット情報の検索 . . . . .	1466
SNAPDB_MEMORY_POOL 管理ビューおよび SNAP_GET_DB_MEMORY_POOL 表関数 - データベース・レベルのメモリー使用量情報の検索 . . . . .	1471
SNAPDBM_MEMORY_POOL 管理ビューおよび SNAP_GET_DBM_MEMORY_POOL 表関数 - データベース・マネージャー・レベルのメモリー使用量情報の検索 . . . . .	1476
SNAPHADR 管理ビューおよび	
SNAP_GET_HADR 表関数 - hadr 論理データ・グループのスナップショット情報の検索 . . .	1480
SNAPLOCK 管理ビューおよび	
SNAP_GET_LOCK 表関数 - lock 論理データ・グループのスナップショット情報の検索 . . .	1484
SNAPLOCKWAIT 管理ビューおよび	
SNAP_GET_LOCKWAIT 表関数 - lockwait 論理データ・グループのスナップショット情報の検索 . . . . .	1491
SNAPSHOT_AGENT . . . . .	1498
SNAPSHOT_APPL . . . . .	1500
SNAPSHOT_APPL_INFO . . . . .	1505
SNAPSHOT_BP . . . . .	1508
SNAPSHOT_CONTAINER . . . . .	1511
SNAPSHOT_DATABASE . . . . .	1512
SNAPSHOT_DBM . . . . .	1519
SNAPSHOT_DYN_SQL . . . . .	1521
SNAPSHOT_FCM . . . . .	1523
SNAPSHOT_FCMNODE . . . . .	1525
SNAPSHOT_FILEW . . . . .	1526
SNAPSHOT_LOCK . . . . .	1527

SNAPSHOT_LOCKWAIT . . . . .	1529
SNAPSHOT_QUIESCERS . . . . .	1530
SNAPSHOT_RANGES . . . . .	1532
SNAPSHOT_STATEMENT . . . . .	1533
SNAPSHOT_SUBSECT . . . . .	1536
SNAPSHOT_SWITCHES . . . . .	1538
SNAPSHOT_TABLE . . . . .	1540
SNAPSHOT_TBREORG . . . . .	1541
SNAPSHOT_TBS . . . . .	1543
SNAPSHOT_TBS_CFG . . . . .	1546
SNAPSTORAGE_PATHS 管理ビューおよび	
SNAP_GET_STORAGE_PATHS_V97 表関数 - 自動ストレージ・パスの情報の検索 . . . . .	1548
SNAPTAB 管理ビューおよび SNAP_GET_TAB 表関数 - table 論理データ・グループのスナップショット情報の検索 . . . . .	1551
SQLCACHE_SNAPSHOT . . . . .	1555
SYSINSTALLROUTINES . . . . .	1557
WLM_GET_ACTIVITY_DETAILS - 特定のアクティビティに関する詳細情報を戻す . . . . .	1558
WLM_GET_SERVICE_CLASS_AGENTS_V97 - サービス・クラスで実行中のエージェントのリスト . . . . .	1564
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES_V97 - ワークロード・オカレンスのリスト . . . . .	1573
WLM_GET_SERVICE_SUBCLASS_STATS_V97 - サービス・サブクラスの統計を戻す . . . . .	1577
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97 - アクティビティのリストを戻す . . . . .	1583
WLM_GET_WORKLOAD_STATS_V97 - ワークロード統計を戻す . . . . .	1588

**付録 A. DB2 技術情報の概説 . . . . . 1591**

DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式) . . . . .	1592
コマンド行プロセッサから SQL 状態ヘルプを表示する . . . . .	1594
異なるバージョンの DB2 インフォメーション・センターへのアクセス . . . . .	1595
コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの更新 . . . . .	1595
コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新 . . . . .	1597
DB2 チュートリアル . . . . .	1599
DB2 トラブルシューティング情報 . . . . .	1600
ご利用条件 . . . . .	1600

**付録 B. 特記事項 . . . . . 1603**

**索引 . . . . . 1607**

---

## 組み込みルーチンおよびビュー

組み込み管理ルーチンおよびビューは、構造化照会言語 (SQL) を介して DB2® データベースおよびデータベース・オブジェクトを管理および使用するための簡易プログラマチック・インターフェースを提供します。組み込みルーチンには、プロシージャ、スカラー関数、表関数が含まれます。

組み込みルーチンおよびビューを使用して、さまざまな DB2 タスクを実行できます。例えば、組み込みルーチンを使用して、表の再編成、モニター・データのキャプチャーおよび取得、または現行接続のアプリケーション ID の取得が可能です。

これらの組み込みルーチンおよびビューは、SQL ベースのアプリケーション、DB2 コマンド行、またはコマンド・スクリプトから呼び出すことができます。

---

## 組み込みのルーチンおよびビューをアプリケーションで呼び出す場合のベスト・プラクティス

組み込みルーチンおよびビューを適切に使用できるようにするために、特定のコーディング手法が推奨されています。ルーチンはリリースによって変わることがあり、また機能拡張が行われるとフィックスバックなどでリリース内でも変わる可能性があるため、これらの手法は特に重要です。

組み込みルーチンまたはビューを使って情報を取得するために照会を発行するとき、ワイルドカードを使用してすべての列を選択せずに、特定の列を選択してください。例えば、次のような照会を発行しないでください。

```
SELECT * FROM TABLE(MON_GET_UNIT_OF_WORK(NULL,-1)) AS t
ORDER BY total_cpu_time DESC
```

その代わりに、SELECT ステートメントで結果列を指定します。この手法を使用すると、結果列の数とそれらが戻される順序をアプリケーションが制御できるようになります。前の照会を書き直した次の照会では、列が指定されます。

```
SELECT application_handle,
       uow_id,
       total_cpu_time,
       app_rqsts_completed_total,
       rqsts_completed_total
FROM TABLE(MON_GET_UNIT_OF_WORK(NULL,-1)) AS t
ORDER BY total_cpu_time DESC
```

列を指定することで、ルーチン内の列の順序や数が変わる場合の問題を防ぐことができます。ルーチンが戻す結果列の数が増えることがあります。例えば、ルーチンが 6 つの結果列を戻すときにホスト変数を 5 つしか指定していなければ、アプリケーションは中断してしまいます。

さらに、ルーチンの出力パラメーターまたは結果列のタイプやサイズが変わることもあります。例えば、列が VARCHAR(8) から VARCHAR(128) に変わったり、INTEGER 列が BIGINT 列になったりすることがあります。使用する変数が小さすぎる場合は、ルーチンから受け取るデータが切り捨てられる可能性があります。

このような変化から C アプリケーションを保護するために、準備済み (PREPARE) ステートメントを記述 (DESCRIBE) することにより、戻される結果列とそのタイプおよびサイズを判別できます。次の例は、準備済みステートメントを記述する方法を示しています。

```
strcpy(strStmt, "SELECT application_handle, uow_id, total_cpu_time
FROM TABLE(MON_GET_UNIT_OF_WORK(NULL,-1))
AS t ORDER BY total_cpu_time DESC");
EXEC SQL PREPARE stmt FROM :strStmt;
EXEC SQL DESCRIBE stmt INTO :pSqllda;
```

戻される情報を SQL 記述子域 (SQLDA) で使用する方法的例については、`samples/c/tbread.sqc` ファイルにある `RowDatamemoryAlloc` 関数を参照してください。

Java™ および .NET アプリケーションの場合は、プログラムのデータ・タイプおよびサイズを知る必要があります。次の例に示すように、メタデータを使用することにより、戻される結果列とそのタイプおよびサイズを判別できます。

```
ResultSet rs = pstmt.executeQuery();
ResultSetMetaData rsm = rs.getMetaData();
```

結果セットのメタデータを使用する方法的例については、`samples/java/jdbc/Tbread.java` ファイルにある `execPreparedQueryWithUnknownOutputColumn()` メソッドを参照してください。

---

## 組み込みルーチンとビューを使用するための許可

すべての組み込みルーチンとビューの実行には、特定の特権が必要になります。

### 組み込みルーチン

SYSPROC スキーマのすべての組み込みルーチンは、ルーチンに対する EXECUTE 特権を必要とします。次の照会を使用して、許可 ID あるいは属するグループまたはロールが EXECUTE 特権を持っているかどうかを確認できます。

```
SELECT A.SPECIFICNAME, GRANTEE, GRANTEETYPE
FROM SYSCAT.ROUTINEAUTH A, SYSCAT.ROUTINES R
WHERE A.SCHEMA = R.ROUTINESCHEMA
AND A.SPECIFICNAME = R.SPECIFICNAME
AND A.SCHEMA = 'SYSPROC'
AND R.ROUTINENAME = 'routine_name'
AND A.EXECUTEAUTH <> 'N'
```

ここで `routine_name` は、組み込みルーチンの名前です。

ユーザーの許可 ID、またはユーザーが属しているグループあるいはロールが GRANTEE 列にリストされている場合には、指定の組み込みルーチンに対するアクセス権があります。

### 組み込みビュー

SYSIBMADM スキーマ内のすべての組み込みビューについて、ビューに対する SELECT 特権が必要です。次の照会を使用して、許可 ID あるいは属するグループまたはロールが SELECT 特権を持っているかどうかを確認できます。



```
SELECT GRANTEE, GRANTEETYPE
FROM SYSCAT.TABAUTH
WHERE TABSCHEMA = 'SYSIBMADM'
AND TABNAME = 'view_name'
AND SELECTAUTH <> 'N'
```

ここで *view\_name* は組み込みビューの名前です。

ユーザーの許可 ID、またはユーザーが属しているグループあるいはロールが GRANTEE 列にリストされている場合には、指定の組み込みビューに対するアクセス権があります。

---

## 組み込みビューと表関数との比較

組み込みビューは、SQL による DB2 機能への簡易アプリケーション・プログラミング・インターフェースを提供します。

組み込みビューは、以下の 3 つのカテゴリーに分けられます。

- カタログ・ビューを基にしたビュー
- 入力パラメーターを持たない表関数を基にしたビュー
- 1 つ以上の入力パラメーターを持つ表関数を基にしたビュー

表関数は組み込みビューと同じような情報を戻すことができますが、表関数を使用すると、特定のデータベース・パーティション、すべてのデータベース・パーティションのサブセット、またはすべてのデータベース・パーティションのいずれかの特定のデータベースを対象とした情報を取得することができます。

以下の例は、1 つ以上の入力パラメーターを持つ表関数に基づいた組み込みビューを使用する場合と、対応する表関数を使用する場合の違いを示しています。

- 通知ログ・メッセージを検索する **PDLOGMSG\_LAST24HOURS** ビューは、直前の 24 時間のデータへの迅速なアクセスを提供します。一方、**PD\_GET\_LOG\_MSGS** 表関数を使用して、指定した期間のデータを検索できます。
- スナップショット・モニター・ビュー (**SNAP** で始まる名前で識別される) は、各データベース・パーティションのデータへのアクセスを提供します。一方、スナップショット・モニター表関数 (**SNAP\_GET\_** で始まる名前で識別される) には、単一データベース・パーティションからのデータかすべてのデータベース・パーティションからのデータ・サブセットのどちらかを選択するオプションがあります。
- **ADMINTABINFO** ビューは、データベースのすべての表の情報を検索します。この動作は、大規模データベースを使用するアプリケーションのパフォーマンスに大きな影響を及ぼすことがあります。代わりに、**ADMIN\_GET\_TAB\_INFO** 表関数を使用して、スキーマ名または表名、あるいはその両方を入力として指定することにより、パフォーマンスへの影響を軽減できます。

1 つ以上の入力パラメーターを持つ表関数を基にした組み込みビューの場合、組み込みビューと表関数の両方を使用できますが、それぞれは以下のように異なる目標を実現します。

組み込みビューは必ず最新のバージョンの表関数を基にします。ただし、戻される情報の中の列の位置は、新しい情報を戻せるように、リリースによって変わる可能

性があります。したがって、組み込みビューまたは表関数から特定の列を選択する必要があります。あるいは、アプリケーションが `SELECT *` ステートメントを使用する場合は、結果セットを記述する必要があります。

バージョンの接尾部 (`_Vxx`) がある表関数は、推奨されていないか、廃止されています。推奨されない関数は、将来のリリースで廃止される可能性があります。したがって、これらの表関数を使用するアプリケーションやスクリプトを変更し、対応する表関数でバージョンの接尾部がないものを呼び出すようにする必要があります。

## サポートされる組み込み SQL ルーチンおよび管理ビュー

サポートされる組み込み SQL ルーチンと管理ビューについての情報が記載されています。

このトピックには、以下の組み込み SQL ルーチンについての情報が記載されています。

- 管理 SQL ルーチンと ADMIN\_CMD ストアード・プロシージャ: 表 1
- 管理タスク・スケジューラー・ルーチンおよびビュー: 6 ページの表 2
- 監査ルーチンおよびプロシージャ: 6 ページの表 3
- 自動保守組み込み SQL ルーチンおよびビュー: 7 ページの表 4
- 共通 SQL API ストアード・プロシージャ: 7 ページの表 5
- 構成用組み込み SQL ルーチンおよびビュー: 8 ページの表 6
- DB2 pureScale<sup>®</sup> インスタンス用管理ビュー: 8 ページの表 7
- 環境用組み込み SQL ルーチンおよびビュー: 8 ページの表 8
- Explain ルーチン: 9 ページの表 9
- モニター組み込み SQL ルーチン: 10 ページの表 10
- MQSeries<sup>®</sup> 組み込み SQL ルーチン: 15 ページの表 11
- セキュリティー用組み込み SQL ルーチンおよびビュー: 16 ページの表 12
- スナップショット用組み込み SQL ルーチンおよびビュー: 17 ページの表 13
- SQL プロシージャ用組み込みルーチン: 20 ページの表 14
- 段階的再配分用組み込み SQL ルーチン: 21 ページの表 15
- ストレージ管理ツール用組み込み SQL ルーチン: 21 ページの表 16
- テキスト検索組み込み SQL ルーチン: 21 ページの表 17
- ワークロード管理ルーチン: 22 ページの表 18
- その他の組み込み SQL ルーチンおよび管理ビュー: 23 ページの表 19

表 1. 管理 SQL ルーチン

ルーチン名	スキーマ	説明
ADMIN_CMD プロシージャ	SYSPROC	このプロシージャを使用すると、管理者は CALL ステートメントで ADMIN_CMD を実行することにより、管理コマンド (DB2 コマンド行プロセッサ (CLP) コマンドを含む) を実行することができます。

表 1. 管理 SQL ルーチン (続き)

ルーチン名	スキーマ	説明
ADMIN_COPY_SCHEMA プロシージャ	SYSPROC	このプロシージャは、特定のスキーマと、その中に含まれているすべてのオブジェクトをコピーするために使用されます。
ADMIN_DROP_SCHEMA プロシージャ	SYSPROC	このプロシージャは、特定のスキーマと、その中に含まれているすべてのオブジェクトをドロップするために使用されます。
ADMIN_EST_INLINE_LENGTH 関数	SYSIBM	この関数は、XML 列、BLOB 列、CLOB 列、または DBCLOB 列に格納されたデータをインライン化するために必要な、インライン長の見積もりを戻します。
ADMIN_GET_INDEX_COMPRESS_INFO	SYSPROC	この表関数は、非圧縮索引について索引圧縮による節約の可能性を戻すか、カタログからの索引圧縮統計をレポートします。
ADMIN_GET_INDEX_INFO 表関数	SYSPROC	この表関数は、カタログ・ビューで使用できない索引情報を戻します。
ADMIN_GET_INTRA_PARALLEL スカラー関数	SYSPROC	このスカラー関数は、アプリケーションのパーティション内並列処理の現在の状態を戻します。
ADMIN_GET_MEM_USAGE 表関数	SYSPROC	この表関数は、特定のメンバーのメモリー使用量の統計を戻します。
ADMIN_GET_MSGS 表関数	SYSPROC	この表関数は、ADMIN_CMD プロシージャを介して実行されるデータ移動ユーティリティによって生成されるメッセージを取得するのに使用されます。
ADMIN_GET_STORAGE_PATHS 表関数	SYSPROC (表関数)	この管理ビューおよび表関数は、特に db_storage_group 論理データ・グループから、データベースの自動ストレージ・パスのリストを、ストレージ・パスごとのファイル・システム情報を含めて戻します。
ADMIN_GET_TAB_COMPRESS_INFO 表関数	SYSPROC	この表関数は、表、マテリアライズ照会表 (MQT)、および階層表の圧縮見積もりを戻します。
ADMIN_GET_TAB_DICTIONARY_INFO 表関数	SYSPROC	この表関数は、表、マテリアライズ照会表 (MQT)、および階層表のディクショナリー情報を戻します。
ADMIN_IS_INLINED 関数	SYSIBM	この関数は、XML 列、BLOB 列、CLOB 列、または DBCLOB 列のインライン・データに関する状態情報を取得します。
ADMIN_MOVE_TABLE プロシージャ	SYSPROC	このプロシージャは、アクティブな表のデータを、同じ名前の新しい表オブジェクトに移動します。データはオンラインのままなので、引き続きアクセス可能です。
ADMIN_MOVE_TABLE_UTIL プロシージャ	SYSPROC	このプロシージャは、ADMIN_MOVE_TABLE プロシージャによって使用されるユーザー定義可能値を変更します。

表 1. 管理 SQL ルーチン (続き)

ルーチン名	スキーマ	説明
ADMIN_REMOVE_MSGS プロシージャ	SYSPROC	このプロシージャは、ADMIN_CMD プロシージャを介して実行されるデータ移動ユーティリティによって生成されるメッセージをクリーンアップするのに使用されます。
ADMIN_REVALIDATE_DB_OBJECTS プロシージャ	SYSPROC	このプロシージャは、無効なデータベース・オブジェクトを再度有効にします。
ADMIN_SET_INTRA_PARALLEL プロシージャ	SYSPROC	このプロシージャは、データベース・アプリケーションのパーティション内並列処理を使用可能にしたり使用不可にしたりします。
ADMIN_SET_MAINT_MODE プロシージャ	SYSPROC	このプロシージャは、SQL の実行時の SQL コンパイル中にトリガーや制約を展開しないように、現行接続の内部フラグを設定します。
ADMINTABINFO および ADMIN_GET_TAB_INFO	SYSIBMADM (管理ビュー)、 SYSPROC (表関数)	このビューおよび表関数は、表、マテリアライズ照会表 (MQT)、階層表のサイズおよび状態の情報を戻します。
ADMINTEMPCOLUMNS ビューおよび ADMIN_GET_TEMP_COLUMNS 表関数	SYSIBMADM (管理ビュー)、 SYSPROC (表関数)	このビューおよび表関数は、作成済み一時表および宣言済み一時表の列属性に関する情報を取得します。
ADMINTEMPTABLES ビューおよび ADMIN_GET_TEMP_TABLES 表関数	SYSIBMADM (管理ビュー)、 SYSPROC (表関数)	このビューおよび表関数は、作成済み一時表および宣言済み一時表のインスタンスの表属性およびインスタンス生成時間に関する情報を取得します。

表 2. 管理タスク・スケジューラ・ルーチンおよびビュー

ルーチン名またはビュー名	スキーマ	説明
ADMIN_TASK_ADD	SYSPROC	このプロシージャは、管理タスクをスケジュールに入れます。
ADMIN_TASK_LIST	SYSTOOLS	この管理ビューは、スケジューラで定義されている各タスクに関する情報を検索します。
ADMIN_TASK_REMOVE	SYSPROC	このプロシージャは、スケジューラされたタスクまたはタスク状況レコードを除去します。
ADMIN_TASK_STATUS	SYSTOOLS	この管理ビューは、各タスクの状況に関する情報を検索します。
ADMIN_TASK_UPDATE	SYSPROC	このプロシージャは、既存のタスクを更新します。

表 3. 監査ルーチンおよびプロシージャ

ルーチン名またはビュー名	スキーマ	説明
AUDIT_ARCHIVE プロシージャおよび表関数	SYSPROC	このプロシージャおよび表関数は、現在の監査ログをアーカイブします。

表 3. 監査ルーチンおよびプロシージャ (続き)

ルーチン名またはビュー名	スキーマ	説明
AUDIT_DELIM_EXTRACT プロシージャ	SYSPROC	このプロシージャは、バイナリー・アーカイブ・ログからデータを抽出し、それを区切り文字付きファイルにロードします。
AUDIT_LIST_LOGS 表関数	SYSPROC	この表関数は、現行データベースの、指定されたパスにあるアーカイブ監査ログのリストを戻します。

表 4. 自動保守組み込み SQL ルーチンおよびビュー

ルーチン名またはビュー名	スキーマ	説明
AUTOMAINT_GET_POLICY プロシージャ	SYSPROC	このプロシージャは、データベースの現在の自動保守設定を取得します。
AUTOMAINT_GET_POLICYFILE プロシージャ	SYSPROC	このプロシージャは、データベースの現在の自動保守設定を取得します。
AUTOMAINT_SET_POLICY プロシージャ	SYSPROC	このプロシージャは、現在接続中のデータベースの自動保守ポリシー設定を設定します。
AUTOMAINT_SET_POLICYFILE プロシージャ	SYSPROC	このプロシージャは、現在接続中のデータベースの自動保守設定を設定します。

表 5. 共通 SQL API ストアド・プロシージャ

ルーチン名またはビュー名	スキーマ	説明
CANCEL_WORK プロシージャ	SYSPROC	このプロシージャは、指定されたアクティビティをキャンセルします。固有のアクティビティ ID をまったく指定しないと、接続済みアプリケーションのすべてのアクティビティをキャンセルし、システムでそのアプリケーションを強制的に停止します。
DESIGN_ADVISOR プロシージャ	SYSPROC	このプロシージャは、IBM® DB2 10.1 サーバーから設計アドバイザーの推奨事項を取得します。
GET_CONFIG プロシージャ	SYSPROC	このプロシージャは、データ・サーバー構成データを取得します。このデータには、すべてのデータベース・パーティションの nodes.cfg ファイル・データ、データベース・マネージャー構成データ、データベース構成データ、およびレジストリー設定が含まれます。
GET_MESSAGE プロシージャ	SYSPROC	このプロシージャは、簡略メッセージ・テキスト、詳細メッセージ・テキスト、および SQLCODE の SQLSTATE を取得します。

表 5. 共通 SQL API ストアド・プロシージャ (続き)

ルーチン名またはビュー名	スキーマ	説明
GET_SYSTEM_INFO プロシージャ	SYSPROC	このプロシージャは、データ・サーバーに関する情報を取得します。この情報には、システム、現行インスタンス、インストール済み DB2 データベース製品、環境変数、使用可能な CPU、および他のシステム情報が含まれます。
SET_CONFIG プロシージャ	SYSPROC	このプロシージャは、GET_CONFIG プロシージャによって取得した構成パラメータを更新します。

表 6. 構成用組み込み SQL ルーチンおよびビュー

ルーチン名またはビュー名	スキーマ	説明
DBCFCG 管理ビュー	SYSIBMADM	この管理ビューは、データベース構成情報を戻します。
DBMCFG 管理ビュー	SYSIBMADM	この管理ビューは、データベース・マネージャー構成情報を戻します。

表 7. DB2 pureScale インスタンス用管理ビュー

ビュー名	スキーマ	説明
DB_MEMBERS 表関数	SYSIBMADM	この表関数は、DB2 pureScale インスタンスに関する基本的なメンバー情報を返します。
DB2_CLUSTER_HOST_STATE 管理ビュー	SYSIBMADM	DB2_CLUSTER_HOST_STATE 管理ビューと、関連付けられている DB2_GET_CLUSTER_HOST_STATE 表関数は、DB2 pureScale インスタンスの一部になっているホストに関する情報を取得します。
DB2_INSTANCE_ALERTS 管理ビュー	SYSIBMADM	このビューは、DB2 pureScale インスタンス内のアラートに関する情報を提供します。
DB2_MEMBER および DB2_CF 管理ビュー	SYSIBMADM	DB2_MEMBER および DB2_CF 管理ビューと、関連付けられている DB2_GET_INSTANCE_INFO 表関数は、DB2 pureScale インスタンスのメンバーとクラスター・キャッシング・ファシリティに関する情報を戻します。該当する場合には状態情報が含まれます。

表 8. 環境用組み込み SQL ルーチンおよびビュー

ビュー名	スキーマ	説明
ENV_CF_SYS_RESOURCES 管理ビュー	SYSIBMADM	この管理ビューは、システム上のクラスター・キャッシング・ファシリティ (CF と呼ばれる) で使用されるシステム・リソースのリストを戻します。
ENV_FEATURE_INFO 管理ビュー	SYSPROC	この管理ビューは、ライセンスが必要とされる使用可能なすべてのフィーチャーに関する情報を戻します。

表 8. 環境用組み込み SQL ルーチンおよびビュー (続き)

ビュー名	スキーマ	説明
ENV_GET_DB2_SYSTEM_RESOURCES 表関数	SYSPROC	この表関数は、現行インスタンス内の指定したメンバーの CPU 使用量と DB2 プロセスに関する情報を戻します。
ENV_GET_NETWORK_RESOURCES 表関数	SYSPROC	この表関数は、DB2 を実行するホスト・マシン上でアクティブなすべてのネットワーク・アダプターに関する情報を戻します。
ENV_GET_REG_VARIABLES 表関数	SYSPROC	この表関数は、1 つまたはすべてのデータベース・メンバーから DB2 レジストリー設定を戻します。
ENV_GET_SYSTEM_RESOURCES 表関数	SYSPROC	この表関数は、オペレーティング・システム、CPU、およびメモリーなどの、システム上のメンバーに関連した情報を戻します。
ENV_INST_INFO 管理ビュー	SYSIBMADM	この管理ビューは、現在のインスタンスについての情報を戻します。
ENV_PROD_INFO 管理ビュー	SYSIBMADM	この管理ビューは、インストール済みの DB2 データベース製品についての情報を戻します。
ENV_SYS_INFO 管理ビュー	SYSIBMADM	この管理ビューは、システムについての情報を戻します。

表 9. Explain ルーチン

ルーチン名	スキーマ	説明
EXPLAIN_GET_MSGS 表関数	スキーマは Explain 表スキーマと同じです。	この表関数は、EXPLAIN_DIAGNOSTIC および EXPLAIN_DIAGNOSTIC_DATA Explain 表を照会し、定様式メッセージを戻します。
EXPLAIN_FORMAT_STATS スカラー関数	SYSPROC	この新規のスカラー関数は、定様式の統計情報を表示するために使用されます。この情報は構文解析され、特定の照会についてキャプチャーされた Explain スナップショットから抽出されます。
EXPLAIN_FROM_ACTIVITY プロシージャ	SYSPROC	このプロシージャは、アクティビティ・イベント・モニターから得られるセクションの内容を使用して、ステートメントの特定の実行を Explain します。
EXPLAIN_FROM_CATALOG プロシージャ	SYSPROC	このプロシージャは、カタログから得られるセクションの内容を使用してステートメントを Explain します。
EXPLAIN_FROM_DATA プロシージャ	SYSPROC	このプロシージャは、入力セクションの内容を使用してステートメントを Explain します。

表9. Explain ルーチン (続き)

ルーチン名	スキーマ	説明
EXPLAIN_FROM_SECTION プロシージャ	SYSPROC	このプロシージャは、パッケージ・キャッシュまたはパッケージ・キャッシュ・イベント・モニターから得られるセクションの内容を使ってステートメントを Explain します。

表10. モニター SQL ルーチン

ルーチン名	スキーマ	説明
EVMON_FORMAT_UE_TO_TABLES プロシージャ	SYSPROC	このプロシージャは、フォーマットされていないイベント表に保管されているデータを取り出し、その XML 文書を一連のリレーショナル表に移動します。
EVMON_FORMAT_UE_TO_XML 表関数	SYSPROC	この表関数は、バイナリー・イベントを未フォーマット・イベント表から抽出し、XML 文書にフォーマットします。
EVMON_UPGRADE_TABLES プロシージャ	SYSPROC	このプロシージャは、イベント・モニターの作成後に追加された新規モニター・エレメントまたは変更されたモニター・エレメントに対応するように、イベント・モニターのターゲットの SQL イベント表または未フォーマット・イベント表を変更します。
MON_BP_UTILIZATION 管理ビュー	SYSIBMADM	この管理ビューは、現在接続されているデータベース内のすべてのバッファ・プールとすべてのデータベース・パーティションに関するヒット率、平均読み取り/書き込み時間などの主要なモニタリング・メトリックを戻します。
MON_CONNECTION_SUMMARY 管理ビュー	SYSIBMADM	この管理ビューは、現在接続されているデータベース内のすべての接続に関する主要なメトリックを戻します。
MON_CURRENT_SQL 管理ビュー	SYSIBMADM	この管理ビューは、データベースの全メンバーに対してサブMITされた、まだ未完了であるすべてのアクティビティに関する主要なメトリックを戻します。
MON_CURRENT_UOW 管理ビュー	SYSIBMADM	この管理ビューは、データベースの全メンバーに対してサブMITされたすべての作業単位に関する主要なメトリックを戻します。



表 10. モニター SQL ルーチン (続き)

ルーチン名	スキーマ	説明
MON_DB_SUMMARY 管理ビュー	SYSIBMADM	この管理ビューは、現在接続されているデータベース内のすべてのサービス・クラスにわたって集計された主要なメトリックを戻します。
MON_FORMAT_LOCK_NAME 表関数	SYSPROC	この表関数は、内部ロック名をフォーマット設定し、ロックについての詳細情報を行ベースの形式で戻します。
MON_FORMAT_XML_COMPONENT_TIMES_BY_ROW 表関数	SYSPROC	この表関数は、XML メトリック文書に含まれるコンポーネント時間に関するフォーマット設定された行ベースの出力を戻します。
MON_FORMAT_XML_METRICS_BY_ROW 表関数	SYSPROC	この表関数は、XML メトリック文書に含まれるすべてのメトリックに関するフォーマット設定された行ベースの出力を戻します。
MON_FORMAT_XML_TIMES_BY_ROW 表関数	SYSPROC	この表関数は、XML メトリック文書に含まれる待機時間と処理時間の結合された階層に関するフォーマット設定された行ベースの出力を戻します。
MON_FORMAT_XML_WAIT_TIMES_BY_ROW 表関数	SYSPROC	この表関数は、XML メトリック文書に含まれる待機時間に関するフォーマット設定された行ベースの出力を戻します。
MON_GET_ACTIVITY_DETAILS	SYSPROC	この表関数は、アクティビティに関する詳細情報を戻します。これには、一般的なアクティビティ情報と、アクティビティのメトリックの集合が含まれます。
MON_GET_APPL_LOCKWAIT 表関数	SYSPROC	この表関数は、(現在のデータベースに接続されている) 各アプリケーションのエージェントが取得を待機しているすべてのロックについての情報を戻します。
MON_GET_APPLICATION_HANDLE スカラー関数	SYSPROC	このスカラー関数は、呼び出しアプリケーションのアプリケーション・ハンドルを戻します。
MON_GET_APPLICATION_ID スカラー関数	SYSPROC	このスカラー関数は、呼び出しアプリケーションのアプリケーション ID を戻します。
MON_GET_AUTO_MAINT_QUEUE 表関数	SYSPROC	この表関数は、オートノミック・コンピューティング・デーモン (db2acd) で実行するためのキューに現在入っている自動保守ジョブすべてに関する情報を戻します (自動保守キュー上にジョブをサブミットしないリアルタイム統計を除く)。

表 10. モニター SQL ルーチン (続き)

ルーチン名	スキーマ	説明
MON_GET_AUTO_RUNSTATS_QUEUE 表関数	SYSPROC	この表関数は、現在接続しているデータベースでの自動統計収集によって評価するために現在キューに入れられているオブジェクトすべてに関する情報を戻します。
MON_GET_BUFFERPOOL 表関数	SYSPROC	この表関数は、1 つ以上のバッファプール・プールのモニター・メトリックを戻します。
MON_GET_CF 表関数	SYSPROC	この表関数は、DB2 pureScale 環境内の 1 つ以上のクラスター・キャッシング・ファシリティーに関する状況情報を戻します。
MON_GET_CF_CMD 表関数	SYSPROC	この表関数は、クラスター・キャッシング・ファシリティー (CF) コマンドの処理時間に関する情報を戻します。
MON_GET_CF_WAIT_TIME 表関数	SYSPROC	この表関数は、クラスター・キャッシング・ファシリティー (CF) による要求処理を待機するのに費やした合計時間 (マイクロ秒単位) を報告します。この時間には、クラスター・キャッシング・ファシリティーへの関連通信に要した時間が含まれます。
MON_GET_CONNECTION 表関数	SYSPROC	この表関数は、1 つ以上の接続のメトリックを戻します。
MON_GET_CONNECTION_DETAILS 表関数	SYSPROC	この表関数は、1 つ以上の接続の詳細メトリックを戻します。
MON_GET_CONTAINER 表関数	SYSPROC	この表関数は、1 つ以上の表スペース・コンテナのモニター・メトリックを戻します。
MON_GET_EXTENDED_LATCH_WAIT 表関数	SYSPROC	この関数は、拡張ラッチ待機に関係していたラッチに関する情報を戻します。
MON_GET_EXTENT_MOVEMENT_STATUS 表関数	SYSPROC	この表関数は、エクステント移動操作の状況を戻します。
MON_GET_FCM 表関数	SYSPROC	この表関数は、高速コミュニケーション・マネージャー (FCM) に関するメトリックを戻します。
MON_GET_FCM_CONNECTION_LIST 表関数	SYSPROC	この表関数は、指定されたメンバーのすべての高速コミュニケーション・マネージャー (FCM) 接続に関するモニター・メトリックを戻します。
MON_GET_GROUP_BUFFERPOOL 表関数	SYSPROC	この表関数は、GBP_FULL エラーの検出回数を含め、グループ・バッファプールに関する統計を戻します。
MON_GET_HADR 表関数	SYSPROC	この関数は、高可用性災害時リカバリー (HADR) モニター情報を戻します。

表 10. モニター SQL ルーチン (続き)

ルーチン名	スキーマ	説明
MON_GET_INDEX 表関数	SYSPROC	この表関数は、1 つ以上の索引のメトリックを戻します。
MON_GET_INDEX_USAGE_LIST 表関数	SYSPROC	この表関数は、索引に定義されている使用量リストから情報を戻します。
MON_GET_LOCKS 表関数	SYSPROC	この表関数は、現在接続されているデータベース内のすべてのロックのリストを戻します。
MON_GET_MEMORY_POOL 表関数	SYSPROC	この表関数は、メモリ・セット内に含まれるメモリ・プールからメトリックを取得します。
MON_GET_MEMORY_SET 表関数	SYSPROC	この表関数は、メモリ・セット内に含まれるメモリ・プールからメトリックを取得します。
MON_GET_PAGE_ACCESS_INFO 表関数	SYSPROC	この表関数は、指定された表に関する待機中のバッファ・プール・ページに関する情報を戻します。
MON_GET_PKG_CACHE_STMT 表関数	SYSPROC	この表関数は、データベース・パッケージ・キャッシュ内の静的 SQL ステートメントと動的 SQL ステートメントの両方のポイント・イン・タイム・ビューを戻します。
MON_GET_PKG_CACHE_STMT_DETAILS 表関数	SYSPROC	この表関数は、1 つ以上のパッケージ・キャッシュ項目の詳細メトリックを戻します。
MON_GET_REBALANCE_STATUS 表関数	SYSPROC	この表関数は、表スペースに対する再平衡化操作の状況を戻します。
MON_GET_ROUTINE 表関数	SYSPROC	この表関数は、データベースの活動化以降に呼び出されたプロシージャ、外部プロシージャ、コンパイル済み関数、コンパイル済みトリガー、無名ブロックの集約された実行メトリックを返します。
MON_GET_ROUTINE_DETAILS 表関数	SYSPROC	この表関数は、データベースの活動化以降に呼び出されたプロシージャ、外部関数、コンパイル済み関数、コンパイル済みトリガー、無名ブロックの集約された実行メトリックを返します。メトリックは XML 文書で返されます。
MON_GET_ROUTINE_EXEC_LIST 表関数	SYSPROC	この表関数は、データベースの活動化以降に、それぞれのプロシージャ、外部関数、コンパイル済み関数、コンパイル済みトリガー、および無名ブロックによって呼び出されて実行されたすべてのステートメント (セクション) のリストを返します。

表 10. モニター SQL ルーチン (続き)

ルーチン名	スキーマ	説明
MON_GET_RTS_RQST 表関数	SYSPROC	この表関数は、システム内で保留中のすべてのリアルタイム統計要求と、リアルタイム統計デーモンによって (例えば、リアルタイム統計処理キュー上で) 現在処理されている要求のセットに関する情報を戻します。
MON_GET_SECTION_ROUTINE 表関数	SYSPROC	この表関数は、入力セクションの実行中に呼び出される可能性のあるすべてのプロシージャー、外部関数、コンパイル済み関数、およびコンパイル済みトリガーのリストを返します。
MON_GET_SERVERLIST 表関数	SYSPROC	この表関数は、1 つ以上のメンバー上でキャッシュされるサーバー・リストから、現在接続しているデータベースのサーバー・リストに関するメトリックを戻します。
MON_GET_SERVICE_SUBCLASS 表関数	SYSPROC	この表関数は、1 つ以上のサービス・サブクラスのメトリックを戻します。
MON_GET_SERVICE_SUBCLASS_DETAILS 表関数	SYSPROC	この表関数は、1 つ以上のサービス・サブクラスの詳細メトリックを戻します。
MON_GET_TABLE 表関数	SYSPROC	この表関数は、1 つ以上の表のモニター・メトリックを戻します。
MON_GET_TABLESPACE 表関数	SYSPROC	この表関数は、1 つ以上の表スペースのモニター・メトリックを戻します。
MON_GET_TABLE_USAGE_LIST 表関数	SYSPROC	この表関数は、表に定義されている使用量リストから情報を戻します。
MON_GET_TRANSACTION_LOG 表関数	SYSPROC	この表関数は、現在接続しているデータベースに関するトランザクション・ロギング・サブシステムの情報を戻します。
MON_GET_UNIT_OF_WORK 表関数	SYSPROC	この表関数は、1 つ以上の作業単位のメトリックを戻します。
MON_GET_UNIT_OF_WORK_DETAILS 表関数	SYSPROC	この表関数は、1 つ以上の作業単位の詳細メトリックを戻します。
MON_GET_USAGE_LIST_STATUS 表関数	SYSPROC	この表関数は、使用量リストの現在の状況を戻します。
MON_GET_WORKLOAD 表関数	SYSPROC	この表関数は、1 つ以上のワークロードのメトリックを戻します。
MON_GET_WORKLOAD_DETAILS 表関数	SYSPROC	この表関数は、1 つ以上のワークロードの詳細メトリックを戻します。
MON_INCREMENT_INTERVAL_ID プロシージャー	SYSPROC	このプロシージャーは、モニター・インターバルを 1 ずつ増やし、出力引数内に新しい値を戻します。

表 10. モニター SQL ルーチン (続き)

ルーチン名	スキーマ	説明
MON_LOCKWAITS 管理ビュー	SYSPROC	この管理ビューは、現在接続しているデータベースでのロック取得を待機しているアプリケーションのために機能するエージェントについての情報を戻します。
MON_PKG_CACHE_SUMMARY 管理ビュー	SYSIBMADM	この管理ビューは、キャッシュ内の静的 SQL ステートメントと動的 SQL ステートメントの両方に関する主要なメトリックを戻し、データベース・パッケージ・キャッシュの概要情報を提供します。
MON_SAMPLE_SERVICE_CLASS_METRICS 表関数	SYSPROC	この表関数は、1 つ以上のデータベースの 1 つ以上のサービス・クラスに関するシステム・メトリックを、2 つの時点で読み取ります。すなわち、関数が呼び出される時点と、特定の時間が経過した後です。
MON_SAMPLE_WORKLOAD_METRICS 表関数	SYSPROC	この表関数は、1 つ以上のデータベースの 1 つ以上のワークロードに関するシステム・メトリックを、2 つの時点で読み取ります。すなわち、関数が呼び出される時点と、特定の時間が経過した後です。
MON_SERVICE_SUBCLASS_SUMMARY 管理ビュー	SYSIBMADM	この管理ビューは、現在接続されているデータベース内のすべてのサービス・サブクラスに関する主要なメトリックを戻します。
MON_TBSP_UTILIZATION 管理ビュー	SYSIBMADM	この管理ビューは、現在接続されているデータベース内のすべての表スペースとすべてのデータベース・パーティションに関する、ヒット率や使用率パーセンテージなどの主要なモニタリング・メトリックを戻します。
MON_WORKLOAD_SUMMARY 管理ビュー	SYSIBMADM	この管理ビューは、現在接続されているデータベース内のすべてのワークロードに関する主要なメトリックを戻します。

表 11. MQSeries 組み込み SQL ルーチン

ルーチン名	スキーマ	説明
MQPUBLISH スカラー関数	DB2MQ, DB2MQ1C	このスカラー関数は、MQSeries ロケーションに対してデータを公開します。
MQREAD スカラー関数	DB2MQ, DB2MQ1C	このスカラー関数は、MQSeries ロケーションからメッセージを戻します。

表 11. MQSeries 組み込み SQL ルーチン (続き)

ルーチン名	スキーマ	説明
MQREADALL 表関数	DB2MQ, DB2MQ1C	この表関数は、MQSeries ロケーションからメッセージとメッセージ・メタデータを示した表を戻します。
MQREADALLCLOB 表関数	DB2MQ	この表関数は、指定された MQSeries ロケーションから、メッセージとメッセージ・メタデータの入った表を戻します。
MQREADCLOB スカラー関数	DB2MQ	このスカラー関数は、指定された MQSeries ロケーションからメッセージを戻します。
MQRECEIVE スカラー関数	DB2MQ, DB2MQ1C	このスカラー関数は、MQSeries ロケーションからメッセージを戻し、それに関連したキューからメッセージを除去します。
MQRECEIVEALL 表関数	DB2MQ, DB2MQ1C	この表関数は、MQSeries ロケーションからメッセージとメッセージ・メタデータの入った表を戻し、それに関連したキューからメッセージを除去します。
MQRECEIVEALLCLOB 表関数	DB2MQ	この表関数は、指定された MQSeries ロケーションから、メッセージとメッセージ・メタデータの入った表を戻します。
MQRECEIVECLOB スカラー関数	DB2MQ	このスカラー関数は、指定された MQSeries ロケーションからメッセージを戻します。
MQSEND スカラー関数	DB2MQ, DB2MQ1C	このスカラー関数は、MQSeries ロケーションにデータを送信します。
MQSUBSCRIBE スカラー関数	DB2MQ, DB2MQ1C	このスカラー関数は、特定のトピックに関して公開された MQSeries メッセージにサブスクライブします。
MQUNSUBSCRIBE スカラー関数	DB2MQ, DB2MQ1C	このスカラー関数は、特定のトピックに関して公開された MQSeries メッセージからアンサブスクライブします。

表 12. セキュリティー組み込み SQL ルーチンおよびビュー

ルーチン名またはビュー名	スキーマ	説明
AUTH_GET_INSTANCE_AUTHID スカラー関数	SYSPROC	このスカラー関数はインスタンス所有者の許可 ID を戻します。
AUTH_LIST_AUTHORITIES_FOR_AUTHID 表関数	SYSPROC	この表関数は、データベース構成ファイルにあるか、許可 ID に直接付与されたか、あるいはグループまたはロールを介して間接的に付与された許可 ID によって保持されているすべての権限を戻します。
AUTH_LIST_GROUPS_FOR_AUTHID 表関数	SYSPROC	この表関数は、与えられた許可 ID がメンバーになっているグループのリストを戻します。
AUTH_LIST_ROLES_FOR_AUTHID 関数	SYSPROC	この関数は、与えられた許可 ID がメンバーになっているロールのリストを戻します。

表 12. セキュリティー組み込み SQL ルーチンおよびビュー (続き)

ルーチン名またはビュー名	スキーマ	説明
AUTHORIZATIONIDS 管理ビュー	SYSIBMADM	この管理ビューには、現在接続中のデータベースに対して特権または権限を付与された許可 ID のリストが、それらのタイプとともに含まれます。
OBJECTOWNERS 管理ビュー	SYSIBMADM	この管理ビューには、現在接続中のデータベースに関するすべてのオブジェクト所有権情報が含まれます。
PRIVILEGES 管理ビュー	SYSIBMADM	この管理ビューには、現在接続中のデータベースに対する明示的な特権すべてが含まれます。

表 13. スナップショット用組み込み SQL ルーチンおよびビュー

ルーチン名またはビュー名	スキーマ	説明
APPL_PERFORMANCE 管理ビュー	SYSIBMADM	この管理ビューは、選択行数と読み取り行数の比率の情報をアプリケーションごとに表示します。
APPLICATIONS 管理ビュー	SYSIBMADM	この管理ビューは、接続されているデータベース・アプリケーションに関する情報を戻します。
BP_HITRATIO 管理ビュー	SYSIBMADM	この管理ビューは、データベースでのバッファ・プール・ヒット率 (合計、データ、索引) を戻します。
BP_READ_IO 管理ビュー	SYSIBMADM	この管理ビューは、バッファ・プール読み取りパフォーマンス情報を戻します。
BP_WRITE_IO 管理ビュー	SYSIBMADM	この管理ビューは、バッファ・プールあたりのバッファ・プール書き込みパフォーマンス情報を戻します。
CONTAINER_UTILIZATION 管理ビュー	SYSIBMADM	この管理ビューは、表スペース・コンテナと使用率についての情報を戻します。
LOCKS_HELD 管理ビュー	SYSIBMADM	この管理ビューは、現在のロック保持数についての情報を戻します。
LOCKWAITS 管理ビュー	SYSIBMADM	この管理ビューは、付与待機中ロック数についての情報を戻します。
LOG_UTILIZATION 管理ビュー	SYSIBMADM	この管理ビューは、現在接続されているデータベースのログ使用率についての情報を戻します。
LONG_RUNNING_SQL 管理ビュー	SYSIBMADM	この管理ビューは、現在接続されているデータベースで実行時間が最も長い SQL ステートメントを戻します。
QUERY_PREP_COST 管理ビュー	SYSIBMADM	この管理ビューは、ステートメントのリストを、ステートメントの準備に必要な時間に関する情報とともに戻します。

表 13. スナップショット用組み込み SQL ルーチンおよびビュー (続き)

ルーチン名またはビュー名	スキーマ	説明
SNAP_WRITE_FILE プロシージャ	SYSPROC	このプロシージャは、システム・スナップショット・データを、インスタンス・ディレクトリーの tmp サブディレクトリーにあるファイルに書き込みます。
SNAPAGENT 管理ビューおよび SNAP_GET_AGENT 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、アプリケーション・スナップショットから、特に agent 論理データ・グループのエージェント情報を戻します。
SNAPAPPL 管理ビューおよび SNAP_GET_APPL 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、アプリケーション・スナップショットから、特に appl 論理データ・グループのアプリケーション情報を戻します。
SNAPAPPL_INFO 管理ビューおよび SNAP_GET_APPL_INFO 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、アプリケーション・スナップショットから、特に appl_info 論理データ・グループのアプリケーション情報を戻します。
SNAPBP 管理ビューおよび SNAP_GET_BP 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、バッファ・プール・スナップショットから、特に bufferpool 論理データ・グループのバッファ・プール情報を戻します。
SNAPBP_PART 管理ビューおよび SNAP_GET_BP_PART 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、バッファ・プール・スナップショットから、特に bufferpool_nodeinfo 論理データ・グループのバッファ・プール情報を戻します。
SNAPCONTAINER 管理ビューおよび SNAP_GET_CONTAINER 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、tablespace_container 論理データ・グループからの表スペース・スナップショット情報を戻します。
SNAPDB 管理ビューおよび SNAP_GET_DB 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、データベース (dbase) およびデータベース・ストレージ (db_storage_group) 論理グループからのスナップショット情報を戻します。
SNAPDBM 管理ビューおよび SNAP_GET_DBM 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、DB2 データベース・マネージャ (dbm) 論理グループのスナップショット・モニター情報を戻します。
SNAPDETAILLOG 管理ビューおよび SNAP_GET_DETAILLOG 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、detail_log 論理データ・グループからのスナップショット情報を戻します。
SNAPDYN_SQL 管理ビューおよび SNAP_GET_DYN_SQL 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、dynsql 論理データ・グループからのスナップショット情報を戻します。



表 13. スナップショット用組み込み SQL ルーチンおよびビュー (続き)

ルーチン名またはビュー名	スキーマ	説明
SNAPFCM 管理ビューおよび SNAP_GET_FCM 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、データベース・マネージャ・スナップショットから、特に fcm 論理データ・グループの高速コミュニケーション・マネージャ (FCM) 情報を戻します。
SNAPFCM_PART 管理ビューおよび SNAP_GET_FCM_PART 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、データベース・マネージャ・スナップショットから、特に fcm_node 論理データ・グループの高速コミュニケーション・マネージャ (FCM) 情報を戻します。
SNAPLOCK 管理ビューおよび SNAP_GET_LOCK 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、特に lock 論理データ・グループのロック・スナップショット情報を戻します。
SNAPLOCKWAIT 管理ビューおよび SNAP_GET_LOCKWAIT 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、特に lockwait 論理データ・グループのロック待機スナップショット情報を戻します。
SNAPSTMT 管理ビューおよび SNAP_GET_STMT 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、アプリケーション・スナップショットからステートメントに関する情報を戻します。
SNAPSUBSECTION 管理ビューおよび SNAP_GET_SUBSECTION 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、アプリケーション・サブセクション情報として、subsection 論理モニター・グループの情報を戻します。
SNAPSWITCHES 管理ビューおよび SNAP_GET_SWITCHES 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、データベース・スナップショット切り替え状態に関する情報を戻します。
SNAPTAB 管理ビューおよび SNAP_GET_TAB 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、table 論理データ・グループからのスナップショット情報を戻します。
SNAPTAB_REORG 管理ビューおよび SNAP_GET_TAB_REORG 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、表再編成情報を戻します。
SNAPTbsp 管理ビューおよび SNAP_GET_TBSP 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、table space 論理データ・グループからのスナップショット情報を戻します。
SNAPTbsp_PART 管理ビューおよび SNAP_GET_TBSP_PART 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、tablespace_nodeinfo 論理データ・グループからのスナップショット情報を戻します。
SNAPTbsp_QUIESCER 管理ビューおよび SNAP_GET_TBSP_QUIESCER 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、表スペース・スナップショットから、静止プログラムに関する情報を戻します。
SNAPTbsp_RANGE 管理ビューおよび SNAP_GET_TBSP_RANGE 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、範囲スナップショットから情報を戻します。

表 13. スナップショット用組み込み SQL ルーチンおよびビュー (続き)

ルーチン名またはビュー名	スキーマ	説明
SNAPUTIL 管理ビューおよび SNAP_GET_UTIL 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、utility_info 論理データ・グループからのユーティリティーについてのスナップショット情報を戻します。
SNAPUTIL_PROGRESS 管理ビューおよび SNAP_GET_UTIL_PROGRESS 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、特に progress 論理データ・グループのユーティリティー進行状況情報を戻します。
TBSP_UTILIZATION 管理ビュー	SYSIBMADM	この管理ビューは、表スペースの構成および使用率の情報を戻します。
TOP_DYNAMIC_SQL 管理ビュー	SYSIBMADM	この管理ビューは、実行数、平均実行時間、ソート数、またはステートメントあたりのソートによってソートできる動的 SQL ステートメントのうち、上位のものを戻します。

表 14. SQL プロシージャ用組み込みルーチン

ルーチン名	スキーマ	説明
ALTER_ROUTINE_PACKAGE プロシージャ	SYSPROC	このプロシージャは、コンパイル済み SQL ルーチンまたはコンパイル済みトリガーに関連付けられたパッケージに使用される値を、再バインドを必要とせずに変更します。
GET_ROUTINE_NAME プロシージャ	SYSPROC	このプロシージャは、オブジェクトの LIB_ID が指定された SQL PL オブジェクトの名前を戻します。
GET_ROUTINE_OPTS スカラー関数	SYSPROC	このスカラー関数は、現行セッションでの SQL プロシージャの作成に使われる予定のオプションの文字列値を戻します。
GET_ROUTINE_SAR プロシージャ	SYSFUN	このプロシージャは、最低限同じレベルおよびオペレーティング・システムで実行している別のデータベース・サーバーに、同一のルーチンをインストールするのに必要な情報を戻します。
PUT_ROUTINE_SAR プロシージャ	SYSFUN	このプロシージャは、データベース・サーバーで SQL ルーチンを作成したり定義したりするのに必要な情報を渡します。
REBIND_ROUTINE_PACKAGE プロシージャ	SYSPROC	このプロシージャは、SQL プロシージャに関連したパッケージを再バインドします。
SET_ROUTINE_OPTS プロシージャ	SYSPROC	このプロシージャは、現行セッションでの SQL プロシージャの作成に使われる予定のオプションを設定します。

表 15. 段階的な再配分用組み込み SQL ルーチン

ルーチン名	スキーマ	説明
ANALYZE_LOG_SPACE プロシージャ	SYSPROC	このプロシージャは、ログ・スペース分析情報を戻します。
GENERATE_DISTFILE プロシージャ	SYSPROC	このプロシージャは、データ配分ファイルを生成します。
GET_SWRD_SETTINGS プロシージャ	SYSPROC	このプロシージャは、再配分情報を戻します。
SET_SWRD_SETTINGS プロシージャ	SYSPROC	このプロシージャは、再配分レジストリを作成または変更します。
STEPWISE_REDISTRIBUTE_DBPG プロシージャ	SYSPROC	このプロシージャは、データベース・パーティション・グループに属するものを再配分します。

表 16. ストレージ管理ツール用組み込み SQL ルーチン

ルーチン名	スキーマ	説明
CAPTURE_STORAGEMGMT_INFO プロシージャ	SYSPROC	このプロシージャは、所定のルート・オブジェクトのストレージ関連情報を戻します。
CREATE_STORAGEMGMT_TABLES プロシージャ	SYSPROC	このプロシージャは、ストレージ管理表を作成します。
DROP_STORAGEMGMT_TABLES プロシージャ	SYSPROC	このプロシージャは、すべてのストレージ管理表をドロップします。

表 17. テキスト検索用組み込み SQL ルーチン

ルーチン名	スキーマ	説明
SYSTS_ADMIN_CMD ストアード・プロシージャ	SYSPROC	このプロシージャは、SQL CALL ステートメントを使用してテキスト検索管理コマンドを実行します。
SYSTS_ALTER プロシージャ	SYSPROC	このプロシージャは、索引の更新特性を変更します。
SYSTS_CLEANUP プロシージャ	SYSPROC	このプロシージャを使用すると、データベース内の古くなった DB2 Text Search の索引コレクションを除去できます。
SYSTS_CLEAR_COMMANDLOCKS プロシージャ	SYSPROC	このプロシージャは、データベース内の特定のテキスト検索索引またはすべてのテキスト検索索引のすべてのコマンド・ロックを解除します。
SYSTS_CLEAR_EVENTS プロシージャ	SYSPROC	このプロシージャは、管理に使用される索引のイベント表から、索引付けイベントを削除します。
SYSTS_CONFIGURE プロシージャ	SYSPROC	このプロシージャは、テキスト検索サーバーの接続情報をテキスト検索カタログに適用します。

表 17. テキスト検索用組み込み SQL ルーチン (続き)

ルーチン名	スキーマ	説明
SYSTS_CREATE プロシージャ	SYSPROC	このプロシージャは、テキスト検索関数を使用して列データを検索可能にする、テキスト列のテキスト検索索引を作成します。
SYSTS_DISABLE プロシージャ	SYSPROC	このプロシージャは、現行データベースの DB2 テキスト検索を使用不可にします。
SYSTS_DROP プロシージャ	SYSPROC	このプロシージャは、表列と関連付けられた既存のテキスト検索索引をドロップします。
SYSTS_ENABLE プロシージャ	SYSPROC	このプロシージャは、データベース内の表の列に対してテキスト検索索引を作成する前に、正常に実行する必要があります。
SYSTS_UPDATE プロシージャ	SYSPROC	このプロシージャは、索引が関連付けられるテキスト列の現行の内容を反映するテキスト検索索引を更新します。
SYSTS_UPGRADE_CATALOG プロシージャ	SYSPROC	このプロシージャは、管理表や管理ビューを含む DB2 Text Search カタログを最新の製品バージョンにアップグレードします。
SYSTS_UPGRADE_INDEX プロシージャ	SYSPROC	このプロシージャは、テキスト検索カタログ表内の DB2 Text Search 索引情報を更新します。

表 18. ワークロード管理用の組み込み SQL ルーチン

ルーチン名	スキーマ	説明
WLM_CANCEL_ACTIVITY プロシージャ	SYSPROC	このプロシージャは、指定されたアクティビティをキャンセルします。
WLM_CAPTURE_ACTIVITY_IN_PROGRESS プロシージャ	SYSPROC	このプロシージャは、指定されたアクティビティに関する情報をアクティビティ・イベント・モニターに送信します。
WLM_COLLECT_STATS プロシージャ	SYSPROC	このプロシージャは、サービス・クラス、ワークロード、作業クラス、およびしきい値キューの統計を統計イベント・モニターに送信し、その統計のメモリー内コピーをリセットします。
WLM_GET_CONN_ENV 表関数	SYSPROC	この表関数は、特定の接続の、アクティビティ・データおよびセクション実行時統計の収集を制御する設定の値を戻します。
WLM_GET_QUEUE_STATS 表関数	SYSPROC	この表関数は、1 つ以上のしきい値キューの基本統計情報を戻します。

表 18. ワークロード管理用の組み込み SQL ルーチン (続き)

ルーチン名	スキーマ	説明
WLM_GET_SERVICE_CLASS_AGENTS 表関数	SYSPROC	この表関数は、SERVICE_SUPERCLASS_NAME および SERVICE_SUBCLASS_NAME によって指定されたサービス・クラスで実行しているか、または APPLICATION_HANDLE によって指定されたアプリケーションの代わりに実行している、指定されたパーティション上のエージェントのリストを戻します。
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES 表関数	SYSPROC	この表関数は、特定のパーティション上の指定されたサービス・クラスで実行しているすべてのワークロード・オカレンスのリストを戻します。
WLM_GET_SERVICE_SUBCLASS_STATS 表関数	SYSPROC	この表関数は、1 つ以上のサービス・サブクラスの基本統計を戻します。
WLM_GET_SERVICE_SUPERCLASS_STATS 表関数	SYSPROC	この表関数は、1 つ以上のサービス・スーパークラスの基本統計を戻します。
WLM_GET_WORK_ACTION_SET_STATS 表関数	SYSPROC	この表関数は、作業アクション・セット内の作業クラスの基本統計を戻します。
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES 表関数	SYSPROC	この表関数は、指定されたパーティション上の特定のアプリケーションからサブミットされ、また完了していないすべてのアクティビティのリストを戻します。
WLM_GET_WORKLOAD_STATS 表関数	SYSPROC	この表関数は、1 つ以上のワークロードの基本統計を戻します。
WLM_SET_CLIENT_INFO プロシージャ	SYSPROC	このプロシージャは、DB2 データベース・サーバーでの現行接続に関連付けられたクライアント情報を設定します。
WLM_SET_CONN_ENV プロシージャ	SYSPROC	このプロシージャは、特定の接続でのアクティビティ・データの収集とセクション実行時統計の測定を有効にします。

表 19. その他の組み込み SQL ルーチンおよびビュー

ルーチン名またはビュー名	スキーマ	説明
ALTOBJ プロシージャ	SYSPROC	このプロシージャは、入力 CREATE TABLE ステートメントをターゲット表定義として使用して既存の表を変更します。
COMPILATION_ENV 表関数	SYSPROC	この表関数は、コンパイル環境のエLEMENTを戻します。
CONTACTGROUPS 管理ビュー	SYSIBMADM	この管理ビューは、連絡先グループのリストを戻します。
CONTACTS 管理ビュー	SYSIBMADM	この管理ビューは、データベース・サーバーで定義されている連絡先のリストを戻します。

表 19. その他の組み込み SQL ルーチンおよびビュー (続き)

ルーチン名またはビュー名	スキーマ	説明
DB_HISTORY 管理ビュー	SYSIBMADM	この管理ビューは、現在接続されているデータベース・パーティションに関連付けられた履歴ファイルからの情報を戻します。
DBPATHS 管理ビュー	SYSIBMADM	この管理ビューは、スプリット・ミラー・バックアップなどのタスクに必要なデータベース・パスの値を戻します。
GET_DBSIZE_INFO プロシージャ	SYSPROC	このプロシージャは、データベース・サイズと最大容量を計算します。
NOTIFICATIONLIST 管理ビュー	SYSIBMADM	この管理ビューは、インスタンスの状況が通知される連絡先および連絡先グループのリストを戻します。
PD_GET_DIAG_HIST 表関数	SYSPROC	この表関数は、指定された機能からログ・レコード、イベント・レコード、および通知レコードを戻します。
PDLOGMSG_LAST24HOURS 管理ビューおよび PD_GET_LOG_MSGS 表関数	SYSIBMADM (管理ビュー)、SYSPROC (表関数)	この管理ビューおよび表関数は、DB2 通知ログに記録された問題判別ログ・メッセージを戻します。この情報は、データベース管理者とシステム管理者が使用するためのものです。
REORGCHK_IX_STATS プロシージャ	SYSPROC	このプロシージャは、索引統計を調べて、再編成の必要があるかどうかを判別します。
REORGCHK_TB_STATS プロシージャ	SYSPROC	このプロシージャは、表統計を調べて、再編成の必要があるかどうかを判別します。
SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY プロシージャ	SYSPROC	このプロシージャは、システム期間テンポラル表のテンポラル履歴の記録を使用不可にし、特殊テンポラル列に値を指定できるように、内部オプションを設定します。
SQLERRM スカラー関数	SYSPROC	このスカラー関数には 2 つのバージョンがあります。1 つは、メッセージ・トークンの使用や言語選択などを含む、十分に柔軟性をもたせたメッセージ検索を提供します。もう 1 つは、SQLCODE のみを入力パラメーターとして取り、短メッセージを英語で戻すという、インターフェースが簡単なものです。
SYSINSTALLOBJECTS プロシージャ	SYSPROC	このプロシージャは、特定のツールで必要なデータベース・オブジェクトを作成またはドロップします。

## 管理ルーチンおよび ADMIN\_CMD プロシージャ

### ADMIN\_CMD - 管理コマンドの実行

ADMIN\_CMD プロシージャは、SQL CALL ステートメントを使用して管理コマンドを実行するアプリケーションで使用されます。

#### 構文

```
▶▶—ADMIN_CMD—(—command-string—)—————▶▶
```

スキーマは SYSPROC です。

#### プロシージャ・パラメーター

*command-string*

実行する単一のコマンドを指定する、タイプ CLOB (2M) の入力引数。

#### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

#### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

このプロシージャは現在、以下の DB2 コマンド行プロセッサ (CLP) コマンドをサポートしています。

- ADD CONTACT
- ADD CONTACTGROUP
- AUTOCONFIGURE
- BACKUP - オンラインのみ
- DESCRIBE
- DROP CONTACT
- DROP CONTACTGROUP
- EXPORT
- FORCE APPLICATION
- IMPORT
- INITIALIZE TAPE
- LOAD
- PRUNE HISTORY/LOGFILE
- QUIESCE DATABASE

- QUIESCE TABLESPACES FOR TABLE
- REDISTRIBUTE
- REORG INDEXES/TABLE
- RESET ALERT CONFIGURATION
- RESET DATABASE CONFIGURATION
- RESET DATABASE MANAGER CONFIGURATION
- REWIND TAPE
- RUNSTATS
- SET TAPE POSITION
- UNQUIESCE DATABASE
- UPDATE ALERT CONFIGURATION
- UPDATE CONTACT
- UPDATE CONTACTGROUP
- UPDATE DATABASE CONFIGURATION
- UPDATE DATABASE MANAGER CONFIGURATION
- UPDATE HEALTH NOTIFICATION CONTACT LIST
- UPDATE HISTORY

注: コマンドによっては、ADMIN\_CMD プロシージャを介して実行するとき、サポートされる構文が若干異なることがあります。

このプロシージャはまた、CLP ではサポートされていない次のようなコマンドをサポートします。

- GET STMM TUNING
- UPDATE STMM TUNING

## 使用上の注意

コマンドの実行に関する情報の検索:

- ADMIN\_CMD プロシージャはサーバー上で実行されるので、ユーティリティー・メッセージもサーバー上で作成されます。 **MESSAGES ON SERVER** オプション (詳細は、該当するコマンドの項を参照してください) は、メッセージ・ファイルをサーバー上で作成することを指示します。
- コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。
- 管理コマンドの実行が正常に完了した場合に、実行状況以外のものがコマンドから戻されるとしたら、そのような追加情報は、結果セット (最大 2 つまで) の形式で戻されます。例えば、**EXPORT** コマンドが正常に実行された場合、戻される結果セットには、エクスポートされた行の数についての情報が含まれます。一方、**RUNSTATS** コマンドが正常に実行された場合、結果セットは戻されません。結果セットの情報は、対応するコマンドとともに説明されています。
- 管理コマンドの実行が正常に完了しなかった場合、ADMIN\_CMD プロシージャから SQL20397W 警告メッセージが戻されます。それには、管理コマンドが失敗した理由に関する詳細を示した結果セットが伴います。ADMIN\_CMD プロシージャ



ャーを使用するアプリケーションはすべて、このプロシージャから戻される SQLCODE を検査する必要があります。 SQLCODE が  $\geq 0$  の場合、管理コマンドの結果セットを検索する必要があります。 以下の表は、**MESSAGES ON SERVER** オプションを使用した場合としなかった場合に、どのような情報が戻されるかを示しています。

表 20. ADMIN\_CMD プロシージャから戻される SQLCODE および情報

管理コマンドの実行状況	MESSAGES ON SERVER オプションを指定した場合	MESSAGES ON SERVER オプションを指定しなかった場合
成功	戻された SQLCODE が $\geq 0$ の場合: 追加情報 (結果セット) があれば、それが戻されます。	戻された SQLCODE が $\geq 0$ の場合: 追加情報 (結果セット) があれば、それが戻されます。ただし、MSG_RETRIEVAL 列と MSG_REMOVAL 列は NULL です。
失敗	戻された SQLCODE が 20397 の場合: 追加情報 (結果セット) が戻されます。ただし、データが設定されているのは MSG_RETRIEVAL 列と MSG_REMOVAL 列だけです。	戻された SQLCODE が $< 0$ の場合: 追加情報 (結果セット) は戻されません。

- 結果セットを CLP からや、JDBC および CLI アプリケーションなどのアプリケーションから取り出すことはできますが、組み込み C アプリケーションから取り出すことはできません。
- 大文字と小文字を区別する名前および 2 バイト文字セット (DBCS) 名は、¥" MyTabLe ¥" のように円記号および二重引用符で囲む必要があります。

ADMIN\_CMD を介して実行されるすべてのコマンドでは、データベースへの接続を確立したユーザー ID が認証で使用されます。

必要な追加の権限 (例えば、データベース・サーバー上のファイル・システムへのアクセスを必要とするコマンドに必要な追加の権限) については、該当するコマンドを解説している参照情報に記載されています。

ユーザー定義関数 (SQLSTATE 38001) またはトリガーからこのプロシージャを呼び出すことはできません。

#### **ADD CONTACT コマンド (ADMIN\_CMD プロシージャを使用):**

システムでローカルに定義できる連絡先リストかまたはグローバル・リストで定義できる連絡先リストに、連絡先を追加します。連絡先とは、スケジューラーおよびヘルス・モニターなどのプロセスが、メッセージを送信する先のユーザーです。

Database Administration Server (DAS) の **contact\_host** 構成パラメーターの設定により、リストがローカルかグローバルかが決まります。

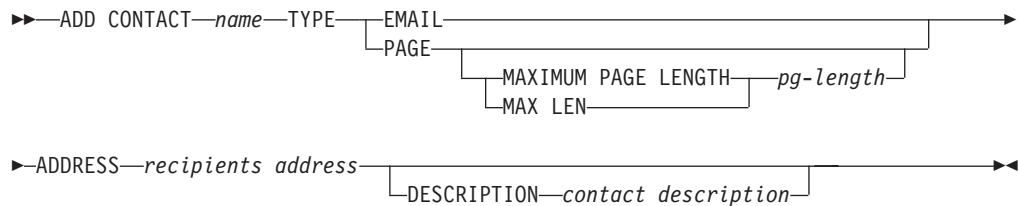
## 許可

なし

## 必要な接続

データベース。 DAS が実行中でなければなりません。

## コマンド構文



## コマンド・パラメーター

### ADD CONTACT *name*

追加される連絡先の名前。デフォルトでは、DB2 Administration Server 構成パラメーター **contact\_host** が別のシステムを指示していない限り、連絡先はローカル・システムに追加されます。

**TYPE** 連絡の方法。次のうちいずれかです。

**EMAIL** この連絡先には、**(ADDRESS)** に E メールで通知します。

**PAGE** この連絡先には、**ADDRESS** にページを送信することで通知します。

### **MAXIMUM PAGE LENGTH** *pg-length*

ページング・サービスにメッセージ長に関する制限がある場合、その制限を文字数で指定します。

通知システムは、SMTP プロトコルを使用して、2DB2 Administration Server 構成パラメーター **smtp\_server** が指定するメール・サーバーに通知を送信します。E メールを送信したり、ページャーを呼び出したりするのは、SMTP サーバーの役割です。

### **ADDRESS** *recipients-address*

宛先の SMTP メールボックス・アドレス。例えば、joe@somewhere.org などです。 **smtp\_server** DAS 構成パラメーターは、SMTP サーバーの名前に設定することが必要です。

### **DESCRIPTION** *contact description*

連絡先のテキスト記述。長さは、最大 128 文字です。

## 例

電子メール・アドレスが testuser@test.com であるユーザー testuser の連絡先を追加します。

```
CALL SYSPROC.ADMIN_CMD  
('ADD CONTACT testuser TYPE EMAIL ADDRESS testuser@test.com')
```

## 使用上の注意

DAS が作成されていて実行中でなければなりません。

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

### ADD CONTACTGROUP コマンド (ADMIN\_CMD プロシージャを使用):

ローカル・システムで定義されたグループのリストに、新しい連絡先グループを追加します。連絡先グループとは、スケジューラおよびヘルス・モニターなどのモニター・プロセスが、メッセージを送信する先のユーザーおよびグループのリストです。

Database Administration Server (DAS) の **contact\_host** 構成パラメーターの設定により、リストがローカルかグローバルかが決まります。

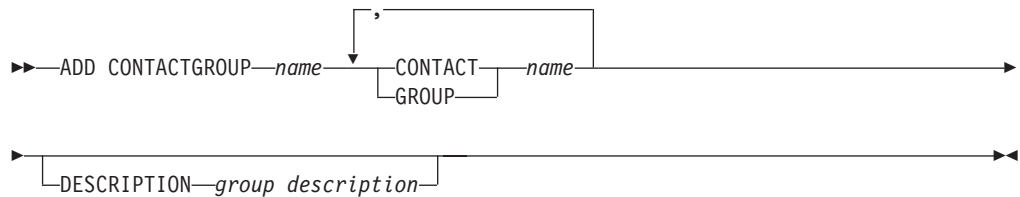
## 許可

なし

## 必要な接続

データベース。DAS が実行中でなければなりません。

## コマンド構文



## コマンド・パラメーター

### ADD CONTACTGROUP *name*

新しい連絡先グループの名前。システム上のグループの集合の中で固有なものでなければなりません。

### CONTACT *name*

グループのメンバーである連絡先の名前。グループに追加された後、**ADD CONTACT** コマンドを使用して連絡先を定義できます。

### GROUP *name*

このグループがメンバーである連絡先グループの名前。

### DESCRIPTION *group description*

オプション。連絡先グループのテキスト記述。

## 例

2 つの連絡先 `cname1` と `cname2` を含む連絡先グループ `gname1` を作成します。

```
CALL SYSPROC.ADMIN_CMD( 'add contactgroup gname1 contact cname1, contact cname2' )
```

## 使用上の注意

DAS が作成されていて実行中でなければなりません。

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

## AUTOCONFIGURE コマンド (ADMIN\_CMD プロシーチャーを使用):

バッファース・プール・サイズ、データベース構成パラメーター、およびデータベース・マネージャー構成パラメーターの初期値を計算して表示します。オプションで、これらの報告値を適用するように指定できます。

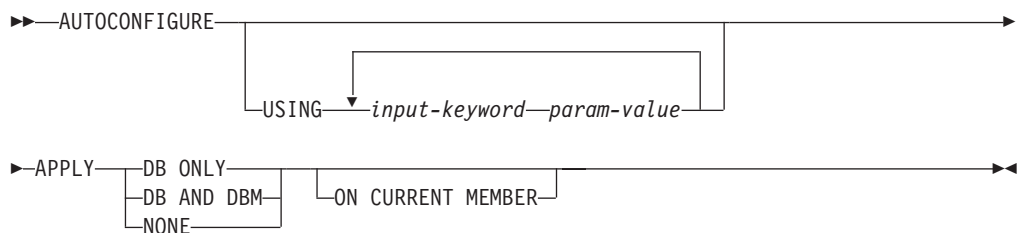
## 許可

SYSADM

## 必要な接続

データベース

## コマンド構文



## コマンド・パラメーター

USING *input-keyword param-value*

表 21. 有効な入力キーワードおよびパラメーター値

キーワード	有効値	デフォルト値	説明
<code>mem_percent</code>	1 から 100	25	データベースに割り当てられているインスタンス・メモリーのパーセンテージ。ただし、 <b>CREATE DATABASE</b> コマンドによって構成アドバイザーが呼び出され、 <code>mem_percent</code> の値を指定しない場合、インスタンスとシステムのメモリー使用量に基づいてパーセンテージが計算され、最大でインスタンス・メモリーの 25% になります。

表 21. 有効な入力キーワードおよびパラメーター値 (続き)

キーワード	有効値	デフォルト値	説明
<b>workload_type</b>	simple、 mixed、 complex	mixed	単純 (simple) ワークロードは入出力集中の傾向があり大部分がトランザクションであるのに対し、複雑 (complex) ワークロードは CPU 集中の傾向があり大部分が照会です。
<b>num_stmts</b>	1 から 1,000,000	10	作業単位ごとのステートメント数
<b>tpm</b>	1 から 200,000	60	1 分ごとのトランザクション
<b>admin_priority</b>	performance、 recovery、 both	both	より良いパフォーマンス (分あたりのより多いトランザクション数) またはより良いリカバリー時間のための最適化
<b>is_populated</b>	yes、 no	yes	データベースにデータを移植するかどうか
<b>num_local_apps</b>	0 から 5,000	0	接続されたローカル・アプリケーションの数
<b>num_remote_apps</b>	0 から 5,000	10	接続されたりリモート・アプリケーションの数
<b>isolation</b>	RR、 RS、 CS、 UR	RR	このデータベースに接続するアプリケーションの最大分離レベル (反復可能読み取り (RR)、読み取り固定 (RS)、カーソル固定 (CS)、非コミット読み取り (UR))。これは、他の構成パラメーターの値を決定するためにのみ使用されます。アプリケーションを特定の分離レベルに制限するよう設定されるものではなく、デフォルト値を使用するのが安全です。
<b>bp_resizeable</b>	yes、 no	yes	バッファー・プールのサイズが変更可能かどうか。

## APPLY

### DB ONLY

現行のデータベース・マネージャーの構成に基づいて、データベース構成およびバッファー・プール設定の推奨値を表示します。データベース構成およびバッファー・プール設定に対して推奨される変更を適用します。

### DB AND DBM

データベース・マネージャー構成、データベース構成、およびバッファー・プール設定に対して推奨される変更を、表示および適用します。

**NONE** 推奨される変更を表示しますが、適用はしません。

## ON CURRENT MEMBER

デフォルトでは、パーティション・データベース環境または DB2 pureScale

環境において、構成アドバイザーによりすべてのメンバー上のデータベース構成が更新されます。**ON CURRENT MEMBER** オプションを指定すると、構成アドバイザーは接続から判別した現在のメンバーに対してメンバー・レベルの構成パラメーターを設定します。一方、グローバル・レベルの構成パラメーターはグローバル・レベルでのみ機能するように構成することができ、設定するとすべてのメンバーに影響を与えます。

バッファ・プールの変更事項は常にシステム・カタログに適用されます。したがってすべてのメンバーが影響を受けます。**ON CURRENT MEMBER** オプションは、バッファ・プールの推奨事項では無視されます。

## 例

ADMIN\_CMD ストアド・プロシージャにより、データベースに対して **AUTOCONFIGURE** を呼び出します。

```
CALL SYSPROC.ADMIN_CMD( 'AUTOCONFIGURE APPLY NONE' )
```

以下は、コマンドによって戻される結果セットの例です。

LEVEL	NAME	VALUE	RECOMMENDED_VALUE	DATATYPE
DBM	ASLHEAPSZ	15	15	BIGINT
DBM	FCM_NUM_BUFFERS	512	512	BIGINT
...				
DB	APP_CTL_HEAP_SZ	128	144	INTEGER
DB	APPGROUP_MEM_SZ	20000	14559	BIGINT
...				
BP	IBMDEFAULTBP	1000	164182	BIGINT

## 使用上の注意

- このコマンドは、現在接続されているデータベースのための推奨構成を作成し、このデータベースがインスタンス上で唯一のアクティブ・データベースであると仮定します。セルフチューニング・メモリー・マネージャーを使用可能にしておらず、インスタンス上に複数のアクティブ・データベースがある場合、データベース・メモリー分散を反映する **mem\_percent** 値を指定してください。例えば、インスタンス・メモリーの 80% を使用しリソースを平等に共有する、2 つのアクティブ・データベースがインスタンスにある場合、40% (80% を 2 データベースで割る) を **mem\_percent** 値に指定します。
- 同じコンピューター上に複数のインスタンスがあり、セルフチューニング・メモリー・マネージャーが使用可能になっていない場合は、各インスタンスで **instance\_memory** に固定値を設定するか、データベース・メモリーの配分を反映する **mem\_percent** 値を指定してください。例えば、すべてのアクティブ・データベースがコンピューター・メモリーの 80% を使用し、それぞれ 1 つのデータベースを持つインスタンスが 4 つある場合は、20% (80% を 4 データベースで割る) を **mem\_percent** 値に指定します。
- AUTOCONFIGURE** コマンドによって構成アドバイザーを明示的に呼び出す場合、**DB2\_ENABLE\_AUTOCONFIG\_DEFAULT** レジストリー変数の設定値は無視されます。
- データベースに対して **AUTOCONFIGURE** コマンドを実行すると、セルフチューニング・メモリー・マネージャーを有効にすることが推奨されます。ただし、**sheapthres** がゼロではないインスタンス内で、あるデータベースに対して **AUTOCONFIGURE** コマンドを実行した場合、ソート・メモリー・チューニング (**sortheap**) は自動的に有効になりません。ソート・メモリー・チューニング

(**sortheap**) を有効にするには、**UPDATE DATABASE MANAGER CONFIGURATION** コマンドを使用して、**sheapthres** をゼロに設定する必要があります。 **sheapthres** の値を変更すると、これまでの既存データベース内のソート・メモリーの使用に影響を与える可能性があることに注意してください。

- コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。
- **AUTOCONFIGURE** コマンドの実行終了時には COMMIT ステートメントが発行されます。タイプ 2 接続では、その結果として ADMIN\_CMD プロシージャから理由コード 2 の SQL30090N が戻されます。

### 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**NODE** および **DBPARTITIONNUM** を **MEMBER** の代わりに指定できます。

### 結果セット情報

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。実行が成功したなら、以下の結果セットによりコマンドから追加情報が戻されます。

表 22. **AUTOCONFIGURE** コマンドから戻される結果セット

列名	データ・タイプ	説明
LEVEL	VARCHAR(3)	パラメーターのレベル。以下のいずれか 1 つ。 <ul style="list-style-type: none"> <li>• BP (バッファー・プール・レベル)</li> <li>• DBM (データベース・マネージャー・レベル)</li> <li>• DB (データベース・レベル)</li> </ul>
NAME	VARCHAR(128)	<ul style="list-style-type: none"> <li>• LEVEL が DB または DBM の場合、これには構成パラメーター・キーワードが含まれています。</li> <li>• LEVEL が BP の場合、この値にはバッファー・プール名が含まれています。</li> </ul>

表 22. AUTOCONFIGURE コマンドから戻される結果セット (続き)

列名	データ・タイプ	説明
VALUE	VARCHAR(256)	<ul style="list-style-type: none"> <li>• LEVEL が DB または DBM で、推奨値が適用された場合、この列の内容は、推奨値適用前に NAME 列の中で示されていた構成パラメーターの値です (つまり古い値が含まれています)。変更が適用されなかった場合、この列の内容は、示されている構成パラメーターについて現在ディスク上にある (据え置き) 値です。</li> <li>• LEVEL が BP で、推奨値が適用された場合、この列の内容は、推奨値適用前に NAME 列の中で示されていたバッファ・プールのサイズ (ページ数) です (つまり古いサイズが含まれています)。変更が適用されなかった場合、この列の内容は、示されているバッファ・プールの現在のサイズ (ページ数) です。</li> </ul>
RECOMMENDED_VALUE	VARCHAR(256)	<ul style="list-style-type: none"> <li>• LEVEL が DB または DBM の場合、この列の内容は、パラメーター列で示されている構成パラメーターの推奨値 (または適用された値) です。</li> <li>• タイプが BP の場合、この列の内容は、パラメーター列で示されているバッファ・プールの推奨サイズ (または適用されたサイズ) です (ページ数)。</li> </ul>
DATATYPE	VARCHAR(128)	パラメーターのデータ・タイプ。

#### BACKUP DATABASE コマンド (ADMIN\_CMD プロシージャを使用):

データベースまたは表スペースのバックアップ・コピーを作成します。

異なるさまざまなオペレーティング・システムおよびハードウェア・プラットフォームの間で DB2 データベース・システムによってサポートされるバックアップ操作の詳細は、『異なるオペレーティング・システムおよびハードウェア・プラットフォーム間のバックアップおよびリストア操作』を参照してください。

#### 有効範囲

パーティション・データベース環境で、データベース・パーティションを指定しない場合、このコマンドはコマンドが実行されたデータベース・パーティションにのみ作用します。



パーティション・バックアップを実行するためのオプションが指定された場合、コマンドを呼び出すことができるのは、カタログ・データベース・パーティションに対してだけです。すべてのデータベース・パーティション・サーバーをバックアップするためのオプションが指定されているなら、それは `db2nodes.cfg` ファイルの中にリストされているすべてのデータベース・パーティション・サーバーに影響を与えます。そうでない場合は、コマンドで指定されたデータベース・パーティション・サーバーに作用します。

## 許可

以下の権限のいずれか。

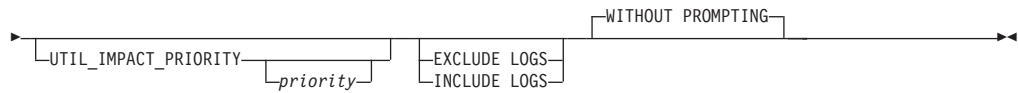
- SYSADM
- SYSCTRL
- SYSMMAINT

## 必要な接続

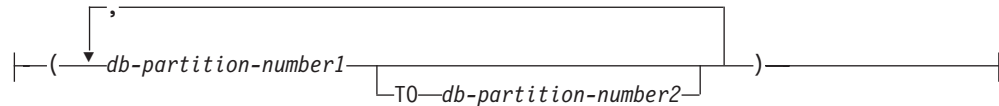
データベース。既存のデータベース接続は、バックアップ操作の完了後はそのまま残ります。

## コマンド構文

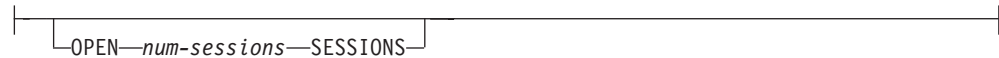




### Partition number(s):



### Open sessions:



### Options:



## コマンド・パラメーター

### DATABASE | DB *database-alias*

バックアップを取るデータベースの別名を指定します。別名は、サーバーで定義されたローカル・データベースでなければならず、また、ユーザーの現在の接続先のデータベース名でなければなりません。データベース別名がユーザーの接続先のものでない場合は、SQL20322N エラーが戻されます。

**ON** データベース・パーティションのセットに対してデータベースをバックアップします。この節は、カタログ・パーティションでのみ指定します。

### DBPARTITIONNUM *db-partition-number1*

データベース・パーティション・リスト内のデータベース・パーティション番号を指定します。

### DBPARTITIONNUMS *db-partition-number1 TO db-partition-number2*

データベース・パーティション番号の範囲を指定します。  
*db-partition-number1* から *db-partition-number2* までの間のすべてのパーティションがデータベース・パーティション・リストに含まれます。

### ALL DBPARTITIONNUMS

db2nodes.cfg ファイルで指定されたすべてのパーティションでデータベースをバックアップすることを指定します。

**EXCEPT** db2nodes.cfg ファイルで指定されたパーティションのうち、データベース・パーティション・リストで指定されたパーティションを除くすべてのパーティションでデータベースをバックアップすることを指定します。

**DBPARTITIONNUM** *db-partition-number1*

データベース・パーティション・リスト内のデータベース・パーティション番号を指定します。

**DBPARTITIONNUMS** *db-partition-number1 TO*

*db-partition-number2*

データベース・パーティション番号の範囲を指定します。*db-partition-number1* から *db-partition-number2* までの間のすべてのパーティションがデータベース・パーティション・リストに含まれます。

**TABLESPACE** *tablespace-name*

バックアップを取る表スペースを指定するときに使用する名前のリスト。

**ONLINE**

オンライン・バックアップを指定します。サポートされているモードは、デフォルトのこれのみです。 **ONLINE** 節を指定する必要はありません。

**INCREMENTAL**

累積 (増分) バックアップ・イメージを指定します。増分バックアップ・イメージとは、正常に実行されたフルバックアップ操作のうち最新のものが実行されて以来変更された、すべてのデータベース・データのコピーです。

**DELTA** 非累積 (差分) バックアップ・イメージを指定します。差分バックアップ・イメージとは、正常に実行された任意のタイプのバックアップ操作のうち最新のものが実行されて以来変更された、すべてのデータベース・データのコピーです。

**USE**

**TSM** バックアップで使用するターゲット・デバイスとして Tivoli® Storage Manager (TSM) を指定します。

**XBSA** XBSA インターフェースを使用するように指定します。バックアップ・サービス API (XBSA) は、バックアップやアーカイブの目的でデータ・ストレージ管理を必要とするアプリケーションまたは機能のための、オープン・アプリケーション・プログラミング・インターフェースです。

**SNAPSHOT**

スナップショット・バックアップを取ることを指定します。

**SNAPSHOT** パラメーターを以下のいずれかのパラメーターと一緒に使用することはできません。

- **TABLESPACE**
- **INCREMENTAL**
- **WITH num-buffers BUFFERS**
- **BUFFER**
- **PARALLELISM**
- **COMPRESS**
- **UTIL\_IMPACT\_PRIORITY**

## • SESSIONS

スナップショット・バックアップのデフォルトの動作は、データベースを構成するすべてのパスの完全データベース・オフライン・バックアップです。つまり、すべてのコンテナ、ローカル・ボリューム・ディレクトリー、データベース・パス (**DBPATH**)、1 次ログとミラー・ログのパスが含まれます (**EXCLUDE LOGS** が明示的に記述されない限り、**INCLUDE LOGS** がすべてのスナップショット・バックアップのデフォルトになります)。

### **LIBRARY** *library-name*

IBM DB2 Server には、以下のストレージ・ハードウェアのための DB2 ACS API ドライバーが組み込まれています。

- IBM TotalStorage SAN ボリューム・コントローラー
- IBM Enterprise Storage Server® Model 800
- IBM Storwize® V7000
- IBM System Storage® DS6000™
- IBM System Storage DS8000®
- IBM System Storage N Series
- IBM XIV®

他のストレージ・ハードウェアを使用していて、そのストレージ・ハードウェア用の DB2 ACS API ドライバーがある場合、**LIBRARY** パラメーターを使用してその DB2 ACS API ドライバーを指定できます。

**LIBRARY** パラメーターの値は、完全修飾ライブラリー・ファイル名です。

## OPTIONS

### **"options-string"**

バックアップ操作で使用するオプションを指定します。ストリングは、二重引用符なしで、入力されたとおりに渡されます。

### **@ file-name**

バックアップ操作で使用するオプションが、DB2 サーバー上のファイルに含まれていることを指定します。このストリングは、ベンダー・サポートのライブラリーに渡されます。ファイル名は完全修飾ファイル名でなければなりません。

**VENDOROPT** データベース構成パラメーターを使用して、スナップショット・バックアップ操作のベンダー固有オプションを指定することはできません。これには、バックアップ・ユーティリティーの **OPTIONS** パラメーターを使用する必要があります。

### **OPEN num-sessions SESSIONS**

DB2 製品と TSM 製品または他のバックアップ・ベンダー製品との間に作成する入出力セッションの数。このパラメーターは、テープ、ディスク、または他のローカル装置にバックアップする場合には効果はありません。オンライン・バックアップで **INCLUDE LOGS** パラメーターを指定した場合、最初のセッションがクローズされた後に、**OPEN num-sessions SESSIONS** パラメー

ターに対して追加のセッションが 1 つ作成されます。 Single System View (SSV) オンライン・バックアップを作成する場合、各ノード・バックアップで、最初のセッションがクローズされた後に **OPEN num-sessions SESSIONS** パラメーターに対して追加のセッションが 1 つ作成されます。 TSM オプションでこのパラメーターを使用すると、履歴ファイルに作成される項目の数が、作成されるセッションの数と等しくなります。

#### **TO** *dir | dev*

ディレクトリーまたはテープ装置名のリストです。ディレクトリーが常駐する絶対パスを指定しなければなりません。このターゲット・ディレクトリーまたは装置は、データベース・サーバー上に存在する必要があります。

パーティション・データベースでは、すべてのデータベース・パーティションにターゲット・ディレクトリーまたは装置が存在する必要があります。オプションで共有パスにすることができます。ディレクトリー名や装置名は、データベース・パーティション式を使用して指定できます。データベース・パーティション式について詳しくは、『自動ストレージ・データベース』を参照してください。

このパラメーターは、バックアップ・イメージが複数の宛先ディレクトリーや装置にわたる場合に、それらを指定するために繰り返すことができます。宛先が複数指定されている場合 (例えば、宛先 1、宛先 2、および宛先 3)、宛先 1 が最初にオープンされます。メディア・ヘッダーおよび特殊ファイル (構成ファイル、表スペース表、および履歴ファイルを含む) は、宛先 1 にあります。他の残りの宛先は、オープンされており、これらはバックアップ操作のときに並列で使用されます。 Windows オペレーティング・システムの場合、汎用テープ装置はサポートされていないので、テープ装置のタイプごとに固有のデバイス・ドライバーが必要です。

テープ装置またはフロッピー・ディスクを使用すると、プロンプトおよびユーザー対話を必要とする可能性があります。その場合は、結果としてエラーが戻されます。

テープ・システムでバックアップ・イメージを固有に参照する機能をサポートしていない場合は、同じテープに同じデータベースの複数のバックアップ・コピーは作成しないことをお勧めします。

#### **LOAD** *library-name*

使用するバックアップおよびリストア I/O ベンダー関数を含む共有ライブラリー (Windows オペレーティング・システムでは DLL) の名前。絶対パスで指定することができます。絶対パスを指定していない場合、デフォルトはユーザー出口プログラムが常駐しているパスになります。

#### **DEDUP\_DEVICE**

データ非重複化をサポートするターゲット・ストレージ・デバイス用にバックアップ・イメージのフォーマットを最適化します。

#### **WITH** *num-buffers* **BUFFERS**

使用するバッファの数です。指定するバッファの数が足りなくて正常にバックアップを作成できない場合、このパラメーターでは、バックアップを完了するために必要な最小値が自動的に選択されます。複数の場所にバックアップする場合は、パフォーマンスを改善するため、より大きな数値をバッファ数に指定することもできます。 **COMPRESS** パラメーターを指定する場

合は、パフォーマンス改善に役立てるため、**PARALLELISM** パラメーターに指定する各表スペース用の追加のバッファを追加できます。

#### **BUFFER** *buffer-size*

4 KB ページごとの単位で表した、バックアップ・イメージを作成する際に使用するバッファのサイズ。値を明示的に指定しない場合、DB2 はこのパラメーターの最適値を自動的に選択します。このパラメーターの最小値は 8 ページです。

さまざまなブロック・サイズのテープを使用する場合は、磁気テープ装置がサポートする範囲内にバッファ・サイズを削減してください。この範囲内でないと、バックアップ操作は正常に実行されることもありますが、作成されたイメージはリカバリー不能になることがあります。

Linux のほとんどのバージョンでは、SCSI テープ装置でバックアップ操作を行うときに、DB2 に用意されたデフォルトのバッファ・サイズを使用すると、エラー SQL2025N、理由コード 75 が表示されます。Linux 内部 SCSI バッファがオーバーフローするのを防ぐには、以下の公式を使用してください。

```
bufferpages <= ST_MAX_BUFFERS * ST_BUFFER_BLOCKS / 4
```

*bufferpages* は **BUFFER** パラメーターと共に使用する値であり、ST\_MAX\_BUFFERS と ST\_BUFFER\_BLOCKS は drivers/scsi ディレクトリー中の Linux カーネルで定義されています。

#### **PARALLELISM** *n*

バックアップ・ユーティリティーによって同時に読み取り可能な表スペースの数を決定します。値を明示的に指定しない場合、DB2 はこのパラメーターの最適値を自動的に選択します。

#### **UTIL\_IMPACT\_PRIORITY** *priority*

バックアップを、指定した優先順位によりスロットル・モードで実行することを指定します。スロットル・モードでは、バックアップ操作によるパフォーマンスの影響を調整できます。優先順位 (*priority*) は 1 から 100 までの範囲の任意の数であり、1 が優先順位最低、100 が優先順位最高を意味します。優先順位の値なしで **UTIL\_IMPACT\_PRIORITY** キーワードが指定された場合は、デフォルトの優先順位 50 でバックアップが実行されます。

**UTIL\_IMPACT\_PRIORITY** を指定しない場合、バックアップは非スロットル・モードで実行されます。バックアップをスロットル・モードで実行するためには、**util\_impact\_lim** 構成パラメーターを設定することによって影響ポリシーが定義されていなければなりません。

#### **COMPRESS**

バックアップを圧縮することを指定します。

#### **COMPRLIB** *name*

圧縮を実行するために使用するライブラリーの名前を指定します (例えば、Windows の場合は db2compr.dll、Linux および UNIX オペレーティング・システムの場合は libdb2compr.so)。この名前は、サーバー上の 1 個のファイルを参照する完全修飾パスでなければなりません。このパラメーターを指定しない場合、デフォルトの DB2 圧縮ライブラリーが使用されます。指定されたライブラリーをロードできない場合、バックアップは失敗します。

## EXCLUDE

圧縮ライブラリーをバックアップ・イメージに格納しないことを指定します。

## COMPROPTS *string*

バイナリー・データのうち、圧縮ライブラリーの初期設定ルーチンに渡すブロックを記述します。DB2はこのストリングをクライアントからサーバーに直接渡すため、バイト反転やコード・ページ変換の問題がある場合は圧縮ライブラリーで処理する必要があります。データ・ブロックの最初の文字が「@」である場合、DB2は、データの残りの部分をサーバー上に存在するファイルの名前と解釈します。その場合DB2は、*string*の内容をそのファイルの内容で置き換え、そのようにして得られる新しい値を初期設定ルーチンに渡します。*string*の最大長は1024バイトです。

## EXCLUDE LOGS

バックアップ・イメージにログ・ファイルをまったく含めないことを指定します。オフライン・バックアップ操作の実行の場合、このオプションが指定されていてもいなくても、ログは除外されます(ただし、スナップショット・バックアップは例外です)。ログは、以下のバックアップ・シナリオにおいて、デフォルトで除外されます。

- 単一パーティション・データベースのオフライン・バックアップ。
- 単一システム・ビュー・バックアップを使用しない場合は、マルチ・パーティション・データベースのオンラインまたはオフラインのバックアップ。

スナップショット・バックアップで **EXCLUDE LOGS** を指定すると、バックアップ中のログ・ファイルへの書き込みが可能になります。これらのログ・ファイルは、デフォルトでスナップショット・バックアップに含まれますが、リカバリーには使用できません。このバックアップをリストアした場合、バックアップからログ・ファイルを抽出してはなりません。バックアップが取られたときにログ・パスがデフォルトに設定されていた場合は、ログ・ファイルをリストアから除外することはできず、バックアップのリストア後にこれらのログ・ファイルを手動で削除する必要があります。ログ・パスがデフォルトではなかった場合、**RESTORE DATABASE** コマンドに **LOGTARGET EXCLUDE** オプションを指定して使用することで、ログ・ファイルをリストア時に除外することができます。

## INCLUDE LOGS

ログ・ファイルのうち、特定の整合ポイント・イン・タイムまでこのイメージをリストアおよびロールフォワードするために必要な範囲をバックアップ・イメージに含めることを指定します。オフライン・バックアップの場合、このオプションは無効です。ただし、スナップショット・バックアップの場合は例外です。**INCLUDE LOGS** は、各データベース・パーティションが独立してバックアップされる(つまり非シングル・システム・ビュー・バックアップ)複数パーティション・オンライン・バックアップの場合を除き、すべてのオンライン・バックアップ操作において常にデフォルト・オプションです。

バックアップが必要とされるいずれかのログ・ファイルが既にバックアップされていて、ログ・パスにはもうない場合、オーバーフロー・ログ・パスが

既に設定されているのであれば、DB2 データベース・マネージャーはそのパスからバックアップ用にログ・ファイルを取り出します。その他の場合には、データベース・マネージャーは現在のログ・パスまたはミラー・ログ・パスから、バックアップ用のログ・ファイルを取り出します。これらのログ・ファイルは、バックアップ完了後にログ・パスから削除されます。

#### WITHOUT PROMPTING

バックアップは、管理されることなく実行されるため、通常はユーザーの介入を必要とするアクションでエラー・メッセージが戻されるように指定されます。これはデフォルトです。

#### 例

以下は、リカバリー可能データベース用の週次の増分バックアップ・ストラテジーのサンプルです。週 1 回の全データベース・バックアップ操作、1 日 1 回の非累積 (差分) バックアップ操作、および週 2 回の累積 (増分) バックアップ操作が含まれています。

```
(Sun) CALL SYSPROC.ADMIN_CMD('backup db sample online use tsm')
(Mon) CALL SYSPROC.ADMIN_CMD
      ('backup db sample online incremental delta use tsm')
(Tue) CALL SYSPROC.ADMIN_CMD
      ('backup db sample online incremental delta use tsm')
(Wed) CALL SYSPROC.ADMIN_CMD
      ('backup db sample online incremental use tsm')
(Thu) CALL SYSPROC.ADMIN_CMD
      ('backup db sample online incremental delta use tsm')
(Fri) CALL SYSPROC.ADMIN_CMD
      ('backup db sample online incremental delta use tsm')
(Sat) CALL SYSPROC.ADMIN_CMD
      ('backup db sample online incremental use tsm')
```

#### 使用上の注意

- バックアップ内のデータは、データベース・サーバーによって保護されるわけではありません。バックアップに LBAC で保護されたデータが含まれる場合は特に、バックアップを適切に保護しておく必要があります。
- テープへのバックアップの場合、現在、可変ブロック・サイズの使用はサポートされていません。そのオプションを使用する必要がある場合は、リカバリーが正常に実行されるように十分にテストしたプロシージャが使用できるようになっていることを確認し、また可変ブロック・サイズを指定して作成されたバックアップ・イメージを使用してください。
- 可変ブロック・サイズを使用する場合、使用している磁気テープ装置の最大限度以下のバックアップ・バッファ・サイズを指定する必要があります。パフォーマンスを最適化するには、使用している装置のブロック・サイズの最大限度と等しい値をバッファ・サイズとして使用しなければなりません。
- スナップショット・バックアップは、ファイラー・システムまたはストレージ・システムで障害が発生した場合に備えて、通常のディスク・バックアップで補完する必要があります。
- 定期的にデータベースをバックアップしていくと、非常に大きなデータベース・バックアップ・イメージ、多くのデータベース・ログ、およびロード・コピー・イメージが累積する場合があります、これらすべてが大量のディスク・スペースを占



めることがあります。これらのリカバリー・オブジェクトの管理方法については、『リカバリー・オブジェクトの管理』を参照してください。

- プロキシ・ノードをサポートする TSM 環境では、**OPTIONS** パラメーターを使用してバックアップ操作を使用可能にすることができます。詳しくは、『Tivoli Storage Manager クライアントの構成』のトピックを参照してください。
- **DB2\_BCKP\_PAGE\_VALIDATION** レジストリー変数を使用して、バックアップ中の DMS および AS ページの妥当性検査を有効にすることができます。
- **DB2\_BCKP\_INCLUDE\_LOGS\_WARNING** レジストリー変数を使用すると、一部のオンライン・バックアップは必要なログすべてが正常に組み込まれていなくても成功できるということを指定できます。

### 結果セット情報

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。実行が成功すると、コマンドは追加情報を戻します。バックアップ操作は、バックアップに含まれる各データベース・パーティションがそれぞれ 1 つの行を構成する、1 つの結果セットを戻します。

表 23. バックアップ操作の結果セット

列名	データ・タイプ	説明
BACKUP_TIME	VARCHAR(14)	バックアップ・イメージに名前を付けるときに使用されたタイム・スタンプ・ストリングに対応します。
DBPARTITIONNUM	SMALLINT	エージェントがバックアップ操作を実行したデータベース・パーティション番号。
SQLCODE	INTEGER	指定されたデータベース・パーティションにおけるバックアップ処理の最終的な SQLCODE 結果。
SQLERRMC	VARCHAR(70)	指定されたデータベース・パーティションにおけるバックアップ処理の最終的な SQLERRMC 結果。
SQLERRML	SMALLINT	指定されたデータベース・パーティションにおけるバックアップ処理の最終的な SQLERRML 結果。

非パーティション・データベースをバックアップする場合や、従来の単一パーティションの構文を使用してパーティション・データベースをバックアップする場合は、結果セットは 1 つの行で構成されます。**DBPARTITIONNUM** には、バックアップされるデータベース・パーティションの ID 番号が含まれます。

SQLCODE、SQLERRMC、および SQLERRML は、指定されたデータベース・パーティションでバックアップによって戻された SQLCA の、同等の名前を持つメンバーを参照します。

**DESCRIBE コマンド (ADMIN\_CMD プロシージャを使用):**

**DESCRIBE** コマンドは、表またはビューの列、索引、およびデータ・パーティションに関するメタデータを表示します。このコマンドは、**SELECT**、**CALL**、または **XQuery** ステートメントの出力に関するメタデータも表示できます。

**DESCRIBE** コマンドを使用して、以下の項目のいずれかに関する情報を表示します。

- **SELECT**、**CALL**、または **XQuery** ステートメントの出力
- 表またはビューの列
- 表またはビューの索引
- 表またはビューのデータ・パーティション

### 許可

必要な許可は、**DESCRIBE** コマンドを使用して表示する情報のタイプによって異なります。

- **SYSTOOLSTMPSPACE** 表スペースが存在する場合、以下の表に示されている権限の 1 つが必要になります。

情報を表示するオブジェクト	必要な特権または権限
<b>SELECT</b> ステートメントまたは <b>XQuery</b> ステートメントの出力	<b>SELECT</b> ステートメント内で参照されている表またはビューごとに、以下のいずれかの特権または権限: <ul style="list-style-type: none"> <li>• <b>SELECT</b> 特権</li> <li>• <b>DATAACCESS</b> 権限</li> <li>• <b>DBADM</b> 権限</li> <li>• <b>SQLADM</b> 権限</li> <li>• <b>EXPLAIN</b> 権限</li> </ul>
<b>CALL</b> ステートメントの出力	以下の特権または権限のいずれか。 <ul style="list-style-type: none"> <li>• <b>DATAACCESS</b> 権限</li> <li>• ストアード・プロシージャでの <b>EXECUTE</b> 特権</li> </ul>

情報を表示するオブジェクト	必要な特権または権限
表またはビューの列	<p>SYSCAT.COLUMNS システム・カタログ表に対して、以下のいずれかの特権または権限:</p> <ul style="list-style-type: none"> <li>• SELECT 特権</li> <li>• ACCESSCTRL 権限</li> <li>• DATAACCESS 権限</li> <li>• DBADM 権限</li> <li>• SECADM 権限</li> <li>• SQLADM 権限</li> </ul> <p><b>SHOW DETAIL</b> パラメーターを使用する場合、SYSCAT.DATAPARTITIONEXPRESSION システム・カタログ表に対しても、これらの特権または権限のいずれかが必要になります。</p> <p>PUBLIC には宣言済み一時表に対するすべての特権が付与されているので、このコマンドを使用して、接続内に存在するすべての宣言済み一時表に関する情報を表示できます。</p>
表またはビューの索引	<p>SYSCAT.INDEXES システム・カタログ表に対して、以下のいずれかの特権または権限:</p> <ul style="list-style-type: none"> <li>• SELECT 特権</li> <li>• ACCESSCTRL 権限</li> <li>• DATAACCESS 権限</li> <li>• DBADM 権限</li> <li>• SECADM 権限</li> <li>• SQLADM 権限</li> </ul> <p><b>SHOW DETAIL</b> パラメーターを使用する場合、GET_INDEX_COLNAMES() UDF に対する EXECUTE 特権も必要です。</p> <p>PUBLIC には宣言済み一時表に対するすべての特権が付与されているので、このコマンドを使用して、接続内に存在するすべての宣言済み一時表に関する情報を表示できます。</p>

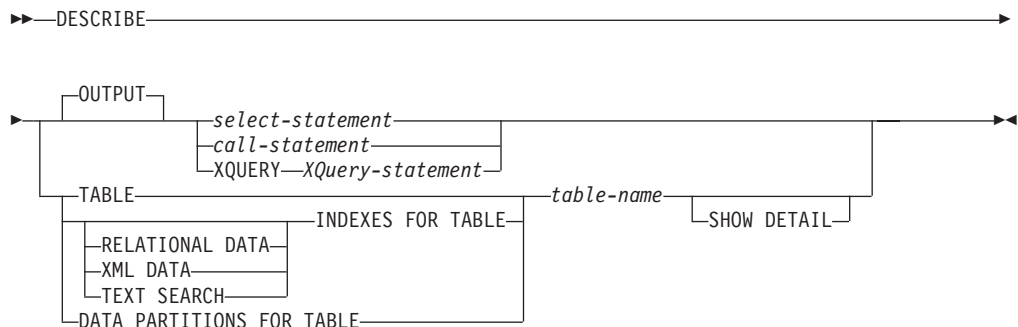
情報を表示するオブジェクト	必要な特権または権限
表またはビューのデータ・パーティション	<p>SYSCAT.DATAPARTITIONS システム・カタログ表に対して、以下のいずれかの特権または権限:</p> <ul style="list-style-type: none"> <li>• SELECT 特権</li> <li>• ACCESSCTRL 権限</li> <li>• DATAACCESS 権限</li> <li>• DBADM 権限</li> <li>• SECADM 権限</li> <li>• SQLADM 権限</li> </ul> <p>PUBLIC には宣言済み一時表に対するすべての特権が付与されているので、このコマンドを使用して、接続内に存在するすべての宣言済み一時表に関する情報を表示できます。</p>

- SYSTOOLSTMPSPACE 表スペースが存在しない場合は、上にリストされた権限のいずれかに加えて SYSADM または SYSCTRL 権限も必要になります。

### 必要な接続

データベース

### コマンド構文



### コマンド・パラメーター

**OUTPUT** ステートメントの出力を記述するよう指示します。このキーワードはオプションです。

*select-statement* | *call-statement* | **XQUERY** *XQuery-statement*

情報が必要なステートメントを識別します。ステートメントは CLP によって自動的に準備されます。XQuery ステートメントであることを示すには、ステートメントの先頭にキーワード **XQUERY** を入れます。DESCRIBE OUTPUT ステートメントで暗黙的な隠し列についての情報が戻されるのは、記述対象の照会の最終的な結果表の SELECT リスト内にその列が明示的に指定されている場合のみです。

## TABLE *table-name*

記述する表またはビューを指定します。 *schema.table-name* 形式の完全修飾名を使用しなければなりません。実際の表の代わりに表の別名を使用することはできません。暗黙的に非表示になっている列に関する情報が戻されますが、どの列が暗黙的に非表示にされているかを示すには SHOW DETAIL を使用する必要があります。

**DESCRIBE TABLE** コマンドは、各列に関する以下の情報をリストします。

- 列名
- タイプ・スキーマ
- タイプ名
- 長さ
- スケール
- NULL 値 (yes/no)

## INDEXES FOR TABLE *table-name*

索引を記述する必要がある表またはビューを指定します。

*schema.table-name* という形式の完全修飾名を使用するか、*table-name* のみを指定してデフォルトのスキーマが自動的に使用されるようにできます。実際の表の代わりに表の別名を使用することはできません。

**DESCRIBE INDEXES FOR TABLE** コマンドは、その表またはビューの各索引に関する以下の情報をリストします。

- 索引スキーマ
- 索引名
- ユニーク規則
- 列の数
- 索引タイプ

**DESCRIBE INDEXES FOR TABLE** コマンドが **SHOW DETAIL** オプションを指定して使用される場合、索引名が 18 バイトより長いと、索引名は切り捨てられます。索引タイプ・オプションが指定されていない場合、すべての索引タイプ (リレーショナル・データ索引、XML データに対する索引、および Text Search 索引) の情報がリストされます。出力には以下の追加情報が含まれます。

- リレーショナル・データ索引の索引 ID、XML パス索引、XML 領域索引、または XML データに対する索引
- XML データに対する索引のデータ・タイプ
- XML データに対する索引のハッシュ
- XML データに対する索引の最大 VARCHAR 長
- XML データに対する索引に指定された XML パターン
- テキスト検索索引のコード・ページ
- テキスト検索索引の言語
- テキスト検索索引に指定された形式
- テキスト検索索引の更新最小数
- テキスト検索索引の更新頻度

- テキスト検索索引のコレクション・ディレクトリー
- 列名
- BUSINESS\_TIME WITHOUT OVERLAPS 節の指定の有無

特定の索引タイプについてだけの情報をリストする索引タイプを指定します。複数の索引タイプの指定はサポートされていません。

#### RELATIONAL DATA

RELATIONAL DATA 索引タイプ・オプションが **SHOW DETAIL** オプションなしで指定されている場合は、以下の情報だけがリストされます。

- 索引スキーマ
- 索引名
- ユニーク規則
- 列の数

**SHOW DETAIL** を指定した場合、列名情報もリストされます。

#### XML DATA

XML DATA 索引タイプ・オプションが **SHOW DETAIL** オプションなしで指定されている場合は、以下の情報だけがリストされます。

- 索引スキーマ
- 索引名
- ユニーク規則
- 列の数
- 索引タイプ

**SHOW DETAIL** を指定した場合、以下の XML データに対する索引の情報もリストされます。

- 索引 ID
- データ・タイプ
- ハッシュ
- 最大 varchar 長
- XML パターン
- 列名

#### TEXT SEARCH

TEXT SEARCH 索引タイプ・オプションが **SHOW DETAIL** オプションなしで指定されている場合は、以下の情報だけがリストされます。

- 索引スキーマ
- 索引名

**SHOW DETAIL** を指定した場合、以下のテキスト検索索引情報もリストされます。

- 列名
- コード・ページ
- 言語
- フォーマット

- 更新最小数
- 更新頻度
- コレクション・ディレクトリー

**TEXT SEARCH** オプションが指定され、テキスト検索オプションがインストールされていないか正しく構成されていない場合、エラー (SQLSTATE 42724) が戻されます。

列にリストされる情報については、DB2 Text Search の情報を参照してください。

#### **DATA PARTITIONS FOR TABLE** *table-name*

データ・パーティションを記述する必要がある表またはビューを指定します。表に含まれるデータ・パーティションごとに表示される情報には、パーティション ID とパーティション・インターバルが含まれます。結果は、パーティション ID の順になっています。 *schema.table-name* 形式の完全修飾名を使用しなければなりません。実際の表の代わりに表の別名を使用することはできません。 *schema* はユーザー名で、その下に表またはビューが作成されます。

**DESCRIBE DATA PARTITIONS FOR TABLE** コマンドの場合、出力に以下の追加情報を内容とする 2 番目の表を含めることを指定します。

- データ・パーティション順序 ID
- SQL でのデータ・パーティション式

#### **SHOW DETAIL**

**DESCRIBE TABLE** コマンドの場合、表のデータ・パーティション式 (表がデータ・パーティション表でなければ 0 行が戻されるかもしれない) を含む 2 番目の結果セットと共に、以下の追加情報を出力に含めることを指定します。

- CHARACTER、VARCHAR または LONG VARCHAR 列のいずれかが FOR BIT DATA として定義されたかどうか
- 列番号
- 分散キー・シーケンス
- コード・ページ
- 非表示属性
- デフォルト
- 表パーティションのタイプ (範囲によってパーティション化されている表の場合、元の出力の後にこの出力が表示される)
- パーティション・キー列 (範囲によってパーティション化されている表の場合、元の出力の後にこの出力が表示される)
- 索引に使用される表スペースの ID
- 表に定義された期間 (テンポラル表の場合、元の出力の後にこの出力が表示される)
- 表でバージョン管理が有効かどうか (テンポラル表の場合、元の出力の後にこの出力が表示される)

## 例

### SELECT ステートメントの出力の記述

次に示すのは、SELECT ステートメントを記述する方法の一例です。

```
CALL SYSPROC.ADMIN_CMD('describe select * from emp_photo')
```

以下はこの SELECT ステートメントの出力例です。

Result set 1

```
-----  
SQLTYPE_ID  SQLTYPE          SQLLENGTH  SQLSCALE  SQLNAME_DATA  ...  
-----  
          452 CHARACTER              6           0  EMPNO         ...  
          448 VARCHAR                10           0  PHOTO_FORMAT  ...  
          405 BLOB                   102400        0  PICTURE       ...
```

3 record(s) selected.

Return Status = 0

この SELECT ステートメントの出力 (続き)。

```
... SQLNAME_LENGTH  SQLDATATYPE_NAME_DATA  SQLDATATYPE_NAME_LENGTH  
... -----  
...              5 SYSIBM .CHARACTER                18  
...             12 SYSIBM .VARCHAR                 16  
...              7 SYSIBM .BLOB                   13
```

### 表の記述

非パーティション表の記述。

```
CALL SYSPROC.ADMIN_CMD('describe table org show detail')
```

以下はこの CALL ステートメントの出力例です。

Result set 1

```
-----  
COLNAME      TYPESCHEMA      TYPENAME      FOR_BINARY_DATA  ...  
-----  
DEPTNUMB     SYSIBM          SMALLINT      N                 ...  
DEPTNAME     SYSIBM          VARCHAR       N                 ...  
MANAGER      SYSIBM          SMALLINT      N                 ...  
DIVISION     SYSIBM          VARCHAR       N                 ...  
LOCATION      SYSIBM          VARCHAR       N                 ...
```

5 record(s) selected.

この CALL ステートメントの出力 (続き)。

```
... LENGTH SCALE NULLABLE COLNO PARTKEYSEQ CODEPAGE DEFAULT  
... -----  
...      2    0 N           0           1           0 -  
...     14    0 Y           1           0          1208 -  
...      2    0 Y           2           0           0 -  
...     10    0 Y           3           0          1208 -  
...     13    0 Y           4           0          1208 -
```

この CALL ステートメントの出力 (続き)。

Result set 2

```
-----  
DATA_PARTITION_KEY_SEQ  DATA_PARTITION_EXPRESSION  
-----
```

0 record(s) selected.

Return Status = 0



パーティション表の記述。

```
CALL SYSPROC.ADMIN_CMD('describe table part_table1 show detail')
```

以下はこの CALL ステートメントの出力例です。

Result set 1

```
-----  
COLNAME      TYPESCHEMA    TYPENAME FOR_BINARY_DATA ...  
-----...- - - - -  
COL1         SYSIBM       INTEGER N          ...
```

1 record(s) selected.

この CALL ステートメントの出力 (続き)。

```
... LENGTH SCALE NULLABLE COLNO PARTKEYSEQ CODEPAGE DEFAULT  
... -----  
...      4      0 N          0          1          0      -
```

この CALL ステートメントの出力 (続き)。

Result set 2

```
-----  
DATA_PARTITION_KEY_SEQ DATA_PARTITION_EXPRESSION  
-----  
1 COL1
```

1 record(s) selected

## 表索引の記述

次に示すのは、表索引を記述する方法の一例です。この呼び出しは表 USER1.DEPARTMENT を記述して、2 つのリレーショナル・データ索引、6 つの XML データ索引、2 つのテキスト検索索引、およびシステム索引をリストします。

```
CALL SYSPROC.ADMIN_CMD('describe indexes for table user1.department')
```

以下はこの CALL ステートメントの出力例です。

Result set 1

```
-----  
INDSCHEMA      INDNAME          UNIQUE_RULE  
-----...- - - - -  
SYSIBM         SQL070531145253450  DUPLICATES_ALLOWED  
SYSIBM         SQL070531145253620  UNIQUE_ENTRIES_ONLY  
USER1          RELIDX1            DUPLICATES_ALLOWED  
USER1          RELIDX2            DUPLICATES_ALLOWED  
SYSIBM         SQL070531145253650  PRIMARY_INDEX  
USER1          XMLIDX1            DUPLICATES_ALLOWED  
SYSIBM         SQL070531154625650  DUPLICATES_ALLOWED  
USER1          XMLIDX2            DUPLICATES_ALLOWED  
SYSIBM         SQL070531154626000  DUPLICATES_ALLOWED  
USER1          XMLIDX3            DUPLICATES_ALLOWED  
SYSIBM         SQL070531154626090  DUPLICATES_ALLOWED  
USER1          XMLIDX4            DUPLICATES_ALLOWED  
SYSIBM         SQL070531154626190  DUPLICATES_ALLOWED  
USER1          XMLIDX5            DUPLICATES_ALLOWED  
SYSIBM         SQL070531154626290  DUPLICATES_ALLOWED  
USER1          XMLIDX6            DUPLICATES_ALLOWED  
SYSIBM         SQL070531154626400  DUPLICATES_ALLOWED  
USER1          TXTIDX1            -  
USER1          TXTIDX2            -
```

19 record(s) selected.

Return Status = 0

この CALL ステートメントの出力 (続き)。

```

... COLCOUNT  INDEXTYPE
... -----
...          -  XML_DATA_REGIONS
...          1  XML_DATA_PATH
...          1  RELATIONAL_DATA
...          2  RELATIONAL_DATA
...          1  RELATIONAL_DATA
...          1  XML_DATA_VALUES_LOGICAL
...          1  XML_DATA_VALUES_PHYSICAL
...          1  XML_DATA_VALUES_LOGICAL
...          1  XML_DATA_VALUES_PHYSICAL
...          1  XML_DATA_VALUES_LOGICAL
...          1  XML_DATA_VALUES_PHYSICAL
...          1  XML_DATA_VALUES_LOGICAL
...          1  XML_DATA_VALUES_PHYSICAL
...          1  XML_DATA_VALUES_LOGICAL
...          1  XML_DATA_VALUES_PHYSICAL
...          1  XML_DATA_VALUES_LOGICAL
...          1  XML_DATA_VALUES_PHYSICAL
...          1  TEXT_SEARCH
...          1  TEXT_SEARCH

```

### データ・パーティションの記述

次に示すのは、データ・パーティションを記述する方法の一例です。

```
CALL SYSPROC.ADMIN_CMD('describe data partitions for table part_table2')
```

以下はこの CALL ステートメントの出力例です。

Result set 1

```

-----
DATA_PARTITION_ID LOW_KEY_INCLUSIVE LOW_KEY_VALUE ...
-----
                0 Y                   1           ...
                1 Y                   10          ...
                2 Y                   20          ...

```

3 record(s) selected.

この CALL ステートメントの出力 (続き)。

```

... HIGH_KEY_INCLUSIVE HIGH_KEY_VALUE
... -----
... N                   10
... N                   20
... N                   40

```

次に示すのは、SHOW DETAIL 節を指定してデータ・パーティションを記述する方法の一例です。

```
CALL SYSPROC.ADMIN_CMD('describe data partitions
for table part_table2 show detail')
```

以下はこの CALL ステートメントの出力例です。

Result set 1

```

-----
DATA_PARTITION_ID LOW_KEY_INCLUSIVE LOW_KEY_VALUE ...
-----
                0 Y                   1           ...
                1 Y                   10          ...
                2 Y                   20          ...

```

3 record(s) selected.

Return Status = 0

この CALL ステートメントの出力 (続き)。

```
... HIGH_KEY_INCLUSIVE HIGH_KEY_VALUE
... -----
... N                    10
... N                    20
... N                    40
```

この CALL ステートメントの出力 (続き)。

Result set 2

```
-----
DATA_PARTITION_ID DATA_PARTITION_NAME TBSPID ...
-----
          0 PART0                3 ...
          1 PART1                3 ...
          2 PART2                3 ...
```

3 record(s) selected.

Return Status = 0

この CALL ステートメントの出力 (続き)。

```
... PARTITION_OBJECT_ID LONG_TBSPID ACCESSMODE STATUS
... -----
...                    15                3 FULL_ACCESS
...                    16                3 FULL_ACCESS
...                    17                3 FULL_ACCESS
```

## 使用上の注意

**DESCRIBE** コマンドが一時表を作成しようとして失敗した場合、SYSTOOLSTMPSPACE の作成が試みられ、その後、再び一時表の作成が、今回は SYSTOOLSTMPSPACE の中で試みられます。SYSCTRL または SYSADM 権限は、SYSTOOLSTMPSPACE 表スペースを作成するために必要です。

## 結果セット情報

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。実行が成功したなら、以下の結果セットによりコマンドから追加情報が戻されます。

- 54 ページの表 24: **DESCRIBE select-statement**、**DESCRIBE call-statement**、および **DESCRIBE XQUERY XQuery-statement** の各コマンド
- 54 ページの表 25: **DESCRIBE TABLE** コマンドの結果セット 1
- 55 ページの表 26: **DESCRIBE TABLE** コマンドの結果セット 2
- 55 ページの表 27: **DESCRIBE INDEXES FOR TABLE** コマンド
- 57 ページの表 28: **DESCRIBE DATA PARTITIONS FOR TABLE** コマンドの結果セット 1
- 58 ページの表 29: **DESCRIBE DATA PARTITIONS FOR TABLE** コマンドの結果セット 2

表 24. DESCRIBE select-statement、DESCRIBE call-statement、および DESCRIBE XQUERY XQuery-statement の各コマンドによって戻される結果セット

列名	データ・タイプ	LOB のみ <sup>1</sup>	説明
SQLTYPE_ID	SMALLINT	いいえ	列のデータ・タイプ。SQL 記述子域 (SQLDA) の SQLTYPE フィールドに示されているのと同じ。
SQLTYPE	VARCHAR (257)	いいえ	SQLTYPE_ID 値に対応するデータ・タイプ。
SQLLEN	INTEGER	いいえ	列の長さ属性。SQLDA の SQLLEN フィールドに示されているのと同じ。
SQLSCALE	SMALLINT	いいえ	10 進 (DECIMAL) 値の場合は小数部分の桁数。それ以外のデータ・タイプの場合は 0。
SQLNAME_DATA	VARCHAR (128)	いいえ	列の名前。
SQLNAME_LENGTH	SMALLINT	いいえ	列名の長さ。
SQLDATA_Typeschema	VARCHAR (128)	はい	データ・タイプ・スキーマ名。
SQLDATA_Typename	VARCHAR (128)	はい	データ・タイプ名。

注: <sup>1</sup>: Yes の場合、記述する LOB データがある場合にのみヌル以外の値が戻されることを示します。

表 25. DESCRIBE TABLE コマンドによって戻される結果セット 1

列名	データ・タイプ	詳細 <sup>2</sup>	説明
COLNAME	VARCHAR (128)	いいえ	列名。
Typeschema	VARCHAR (128)	いいえ	列名が特殊ならスキーマ名が戻され、そうでない場合は 'SYSIBM' が戻されます。
Typename	VARCHAR (128)	いいえ	列タイプの名前。
FOR_BINARY_DATA	CHAR (1)	はい	列が CHAR、VARCHAR、または LONG VARCHAR のいずれかのタイプであり、かつ FOR BIT DATA として定義されているなら 'Y' が戻され、そうでない場合は 'N' が戻されます。
LENGTH	INTEGER	いいえ	データの最大長。DECIMAL (10 進) データの場合、これは精度を示します。特殊タイプの場合、0 が戻されます。
SCALE	SMALLINT	いいえ	DECIMAL (10 進) データの場合、これはスケールを示します。それ以外のすべてタイプの場合、0 が戻されます。
NULLABLE	CHAR (1)	いいえ	以下のいずれか <ul style="list-style-type: none"> <li>列が NULL 可能の場合、'Y'</li> <li>列が NULL 可能でない場合、'N'</li> </ul>

表 25. DESCRIBE TABLE コマンドによって戻される結果セット 1 (続き)

列名	データ・タイプ	詳細 <sup>2</sup>	説明
COLNO	SMALLINT	はい	列の順序。
PARTKEYSEQ	SMALLINT	はい	表のパーティション・キーの中での列の順序。列がパーティション・キーの一部でない場合は NULL または 0 が戻され、副表および階層表の場合は NULL。
CODEPAGE	SMALLINT	はい	列のコード・ページ。以下のうちいずれか。 <ul style="list-style-type: none"> <li>FOR BIT DATA として定義されていない列の場合、データベース・コード・ページの値。</li> <li>グラフィック列の場合、DBCS コード・ページの値。</li> <li>それ以外の場合は 0。</li> </ul>
DEFAULT	VARCHAR (254)	はい	列のデータ・タイプに適した定数、特殊レジスター、または cast 関数で表された表の列のデフォルト値。NULL の場合もあります。

注: <sup>2</sup>: Yes の場合、SHOW DETAIL 節が使用された場合にのみヌル以外の値が戻されることを示します。

表 26. SHOW DETAIL 節が使用された場合に DESCRIBE TABLE コマンドによって戻される結果セット 2

列名	データ・タイプ	説明
DATA_PARTITION_KEY_SEQ	INTEGER	データ・パーティション・キー番号。例えば、最初のデータ・パーティション式の場合は 1、2 番目のデータ・パーティション式の場合は 2。
DATA_PARTITION_EXPRESSION	CLOB (32K)	SQL 構文によるこのデータ・パーティション・キーの式

表 27. DESCRIBE INDEXES FOR TABLE コマンドによって戻される結果セット

列名	データ・タイプ	詳細 <sup>3</sup>	索引タイプ・オプション 4, 5	説明
INDSCHEMA	VARCHAR (128)	いいえ	RELATIONAL DATA XML DATA TEXT SEARCH	索引スキーマ名。
INDNAME	VARCHAR (128)	いいえ	RELATIONAL DATA XML DATA TEXT SEARCH	索引名。
UNIQUE_RULE	VARCHAR (30)	いいえ	RELATIONAL DATA XML DATA	以下の値のいずれか。 <ul style="list-style-type: none"> <li>• DUPLICATES_ALLOWED</li> <li>• PRIMARY_INDEX</li> <li>• UNIQUE_ENTRIES_ONLY</li> </ul>

表 27. DESCRIBE INDEXES FOR TABLE コマンドによって戻される結果セット (続き)

列名	データ・タイプ	詳細 <sup>3</sup>	索引タイプ・オプション 4, 5	説明
INDEX_PARTITIONING	CHAR(1)	いいえ	N/A	索引のパーティション特性を示します。可能な値は以下のとおりです。 <ul style="list-style-type: none"> <li>• N = 非パーティション索引</li> <li>• P = パーティション索引</li> <li>• ブランク = 索引はパーティション表にはない</li> </ul>
COLCOUNT	SMALLINT	いいえ	RELATIONAL DATA XML DATA	キー内の列数と組み込み列 (存在する場合) の数の合計。
INDEX_TYPE	VARCHAR (30)	いいえ	RELATIONAL DATA XML DATA TEXT SEARCH	索引のタイプ: <ul style="list-style-type: none"> <li>• RELATIONAL_DATA</li> <li>• TEXT_SEARCH</li> <li>• XML_DATA_REGIONS</li> <li>• XML_DATA_PATH</li> <li>• XML_DATA_VALUES_LOGICAL</li> <li>• XML_DATA_VALUES_PHYSICAL</li> </ul>
INDEX_ID	SMALLINT	はい	RELATIONAL DATA XML DATA	リレーショナル・データ索引の索引 ID、XML バス索引、XML 領域索引、または XML データに対する索引
DATA_TYPE	VARCHAR (128)	はい	XML DATA	XML データに対する索引に指定された SQL データ・タイプ。以下の値のいずれか。 <ul style="list-style-type: none"> <li>• VARCHAR</li> <li>• DOUBLE</li> <li>• DATE</li> <li>• TIMESTAMP</li> </ul>
HASHED	CHAR (1)	はい	XML DATA	XML データに対する索引の値がハッシュされるかどうかを示します。 <ul style="list-style-type: none"> <li>• 値がハッシュされる場合、'Y'。</li> <li>• 値がハッシュされない場合、'N'。</li> </ul>
LENGTH	SMALLINT	はい	XML DATA	XML データに対する索引の場合、VARCHAR ( <i>integer</i> ) の長さ。その他の場合 0。
PATTERN	CLOB (2M)	はい	XML DATA	XML データに対する索引に指定された XML パターン式。
CODEPAGE	INTEGER	はい	TEXT SEARCH	テキスト検索索引に指定された文書コード・ページ
LANGUAGE	VARCHAR (5)	はい	TEXT SEARCH	テキスト検索索引に指定された文書言語
FORMAT	VARCHAR (30)	はい	TEXT SEARCH	テキスト検索索引に指定された文書フォーマット

表 27. DESCRIBE INDEXES FOR TABLE コマンドによって戻される結果セット (続き)

列名	データ・タイプ	詳細 <sup>3</sup>	索引タイプ・オプション 4, 5	説明
UPDATEMINIMUM	INTEGER	はい	TEXT SEARCH	インクリメンタル更新が実行される前のテキスト検索ログ表内の項目の最小数
UPDATEFREQUENCY	VARCHAR (300)	はい	TEXT SEARCH	テキスト索引に更新を適用するために指定されるトリガー基準
COLLECTION DIRECTORY	VARCHAR (512)	はい	TEXT SEARCH	テキスト検索索引ファイルに指定されたディレクトリー
COLNAMES	VARCHAR (2048)	はい	RELATIONAL DATA XML DATA TEXT SEARCH	列名のリスト。それぞれの前に + が付いている場合は昇順、- が付いている場合は降順。

注: <sup>3</sup>: Yes の場合、**SHOW DETAIL** 節が索引タイプ・オプションの指定なしで使用された場合にのみ、値が戻されることを示します。値は NULL にすることもできます。

注: <sup>4</sup>: **DESCRIBE index-type INDEXES FOR TABLE** を使用した場合に戻される値を示します。例えば、INDEX\_ID 値は、TEXT SEARCH が *index-type* として指定された場合は戻されません。INDEX\_ID 値は、RELATIONAL DATA または XML DATA のいずれかが指定される場合に戻されます。

注: <sup>5</sup>: **DESCRIBE index-type INDEXES FOR TABLE SHOW DETAIL** を使用する場合、値は索引タイプがリストされるときにのみ戻されます。例えば、DATA\_TYPE 値は、XML DATA が *index-type* として指定された場合に戻されます。DATA\_TYPE 値は、TEXT SEARCH または RELATIONAL DATA のいずれかが *index-type* として指定された場合には戻されません。

表 28. DESCRIBE DATA PARTITIONS FOR TABLE コマンドによって戻される結果セット 1

列名	データ・タイプ	詳細 <sup>2</sup>	説明
DATA_PARTITION_ID	INTEGER	いいえ	データ・パーティション ID。
LOW_KEY_INCLUSIVE	CHAR (1)	いいえ	キー値の下限を含む場合は 'Y'、そうでない場合は 'N'。
LOW_KEY_VALUE	VARCHAR (512)	いいえ	このデータ・パーティションのキー値の下限。
HIGH_KEY_INCLUSIVE	CHAR (1)	いいえ	キー値の上限を含む場合は 'Y'、そうでない場合は 'N'。
HIGH_KEY_VALUE	VARCHAR (512)	いいえ	このデータ・パーティションのキー値の上限。

注: <sup>2</sup>: Yes の場合、**SHOW DETAIL** 節が使用された場合にのみヌル以外の値が戻されることを示します。

表 29. SHOW DETAIL 節が使用されている場合に DESCRIBE DATA PARTITIONS FOR TABLE コマンドによって戻される結果セット 2

列名	データ・タイプ	説明
DATA_PARTITION_ID	INTEGER	データ・パーティション ID。
DATA_PARTITION_NAME	VARCHAR (128)	データ・パーティション名。
TBSPID	INTEGER	このデータ・パーティションが格納されている表スペースの ID。
PARTITION_OBJECT_ID	INTEGER	このデータ・パーティションが格納されている DMS オブジェクトの ID。
LONG_TBSPID	INTEGER	LONG データが格納されている表スペースの ID。
INDEX_TBSPID	INTEGER	索引データが格納されている表スペースの ID。
ACCESSMODE	VARCHAR (20)	データ・パーティションのアクセシビリティを定義。以下のいずれかが必要です。 <ul style="list-style-type: none"> <li>• FULL_ACCESS</li> <li>• NO_ACCESS</li> <li>• NO_DATA_MOVEMENT</li> <li>• READ_ONLY</li> </ul>
STATUS	VARCHAR(64)	データ・パーティションの状況。以下のいずれかが必要です。 <ul style="list-style-type: none"> <li>• NEWLY_ATTACHED</li> <li>• NEWLY_DETACHED: MQT メンテナンスが必要。</li> <li>• INDEX_CLEANUP_PENDING: 索引クリーンアップのためのみ</li> </ul> SYSDATAPARTITIONS 内のタプルが維持されている、デタッチされたデータ・パーティション。このタプルは、デタッチされたデータ・パーティションを参照する索引レコードがすべて削除された時点で除去されます。 <p>それ以外の場合、この列はブランクです。</p>

#### DROP CONTACT コマンド (ADMIN\_CMD プロシージャーを使用):

ローカル・システムで定義された連絡先のリストから、連絡先を除去します。連絡先とは、スケジューラーおよびヘルス・モニターがメッセージを送信する先のユーザーです。Database Administration Server (DAS) の **contact\_host** 構成パラメータの設定により、リストがローカルかグローバルかが決まります。

#### 許可

なし

#### 必要な接続

データベース。DAS が実行中でなければなりません。



## コマンド構文

▶▶—DROP CONTACT—*name*————▶▶

### コマンド・パラメーター

**CONTACT** *name*

ローカル・システムからドロップされる連絡先の名前。

### 例

サーバー・システム上の連絡先リストから `testuser` という連絡先をドロップします。

```
CALL SYSPROC.ADMIN_CMD( 'drop contact testuser' )
```

### 使用上の注意

DAS が作成されていて実行中でなければなりません。

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

### **DROP CONTACTGROUP** コマンド (ADMIN\_CMD プロシージャを使用):

ローカル・システムで定義された連絡先のリストから、連絡先グループを除去します。連絡先グループには、スケジューラーおよびヘルス・モニターがメッセージを送信する先のユーザーのリストが入っています。Database Administration Server (DAS) の **contact\_host** 構成パラメーターの設定により、リストがローカルかグローバルかが決まります。

### 許可

なし

### 必要な接続

データベース。DAS が実行中でなければなりません。

## コマンド構文

▶▶—DROP CONTACTGROUP—*name*————▶▶

### コマンド・パラメーター

**CONTACTGROUP** *name*

ローカル・システムからドロップされる連絡先グループの名前。

### 例

連絡先グループ `gname1` をドロップします。

```
CALL SYSPROC.ADMIN_CMD( 'drop contactgroup gname1' )
```

## 使用上の注意

DAS が作成されていて実行中でなければなりません。

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

### EXPORT コマンド (ADMIN\_CMD プロシージャを使用):

データベースから、いくつかある外部ファイル形式のいずれかにデータをエクスポートします。ユーザーは、SQL SELECT ステートメントによって、または型付き表の階層情報によってエクスポートするデータを指定します。データはサーバーにのみエクスポートされます。

67 ページの『エクスポート・ユーティリティー用のファイル・タイプ修飾子』へのクイック・リンク。

### 許可

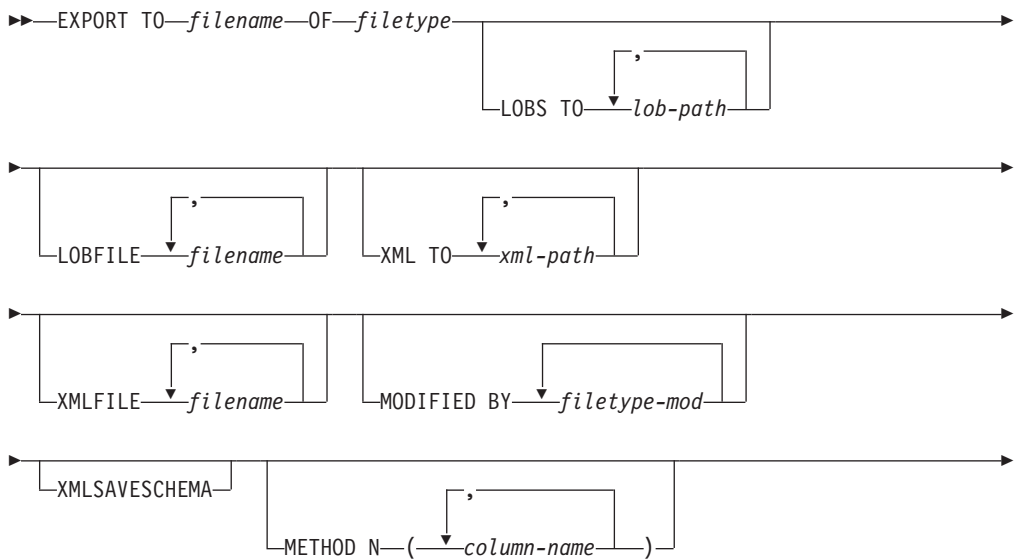
以下の権限のいずれか。

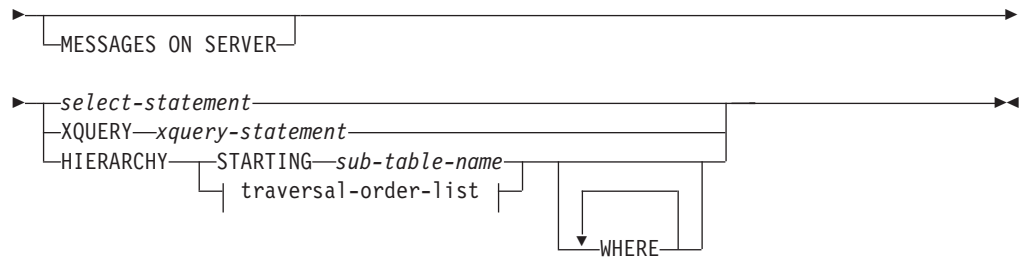
- DATAACCESS 権限
- 関係するそれぞれの表またはビューに対する CONTROL または SELECT 特権

### 必要な接続

データベース。Linux、UNIX、または Windows クライアントから Linux、UNIX、または Windows データベース・サーバーへのユーティリティー・アクセスは、DB2 Connect™ ゲートウェイまたはループバック環境を経由してではなく、エンジンを使用した直接接続でなければなりません。

### コマンド構文





**traversal-order-list:**



**コマンド・パラメーター**

**TO filename**

サーバーにおいてデータのエクスポート先となるファイルの名前を指定します。これは、完全修飾パスでなければならず、サーバー・コーディネーター・パーティション上に存在していなければなりません。

既に存在するファイルの名前を指定した場合、エクスポート・ユーティリティーはファイルの内容を上書きします。つまり情報は追加されません。

**OF filetype**

次のような出力ファイルのデータ・フォーマットを指定します。

- DEL (区切り文字付き ASCII フォーマット)。さまざまなデータベース・マネージャーやファイル・マネージャー・プログラムで使用します。
- IXF (統合交換フォーマット、PC バージョン) は、プロプラエタリー・バイナリー・フォーマットです。

**LOBS TO lob-path**

LOB ファイルが保管される、ディレクトリーへの 1 つ以上のパスを指定します。それらのパスはサーバーのコーディネーター・パーティション上に存在するものでなければならず、また、完全修飾パスでなければなりません。LOB パスごとに少なくとも 1 つのファイルが存在し、各ファイルには少なくとも 1 つの LOB が入ります。指定できるパスの最大数は 999 です。これによって、LOBSINFILE 動作が暗黙的にアクティブ化されます。

**LOBFILE filename**

LOB ファイルに 1 つ以上の基本ファイル名を指定します。最初の名前の名前空間がいっぱいになると、2 番目の名前が使用され、以下 3 番目、4 番目と続きます。これによって、LOBSINFILE 動作が暗黙的にアクティブ化されます。

エクスポート操作中に LOB ファイルを作成するときに、まずこのリストから現行パス (lob-path で指定されたパス) に現行のベース名を追加してから、最初に 3 桁のシーケンス番号、次に 3 文字の ID lob を追加して、ファイル名が構成されます。例えば、現行 LOB パスがディレクトリー /u/foo/lob/path/ で、現行 LOB ファイル名が bar の場合、LOB ファイルは、/u/foo/lob/path/bar.001.lob、/u/foo/lob/path/bar.002.lob (以

下 003、004 と続く) などのように作成されます。LOB ファイル名の 3 桁のシーケンス番号で 999 が使用されてしまうとこの番号は 4 桁に増え、また、9999 が使用されてしまうと 4 桁が 5 桁に増えます。以下同様にシーケンス番号の桁数が増えていきます。

#### **XML TO** *xml-path*

XML ファイルが保管されるディレクトリーを指す 1 つ以上のパスを指定します。XML パスごとに少なくとも 1 つのファイルが存在し、各ファイルには少なくとも 1 つの XQuery データ・モデル (XDM) インスタンスが含まれることとなります。複数のパスが指定された場合、XDM インスタンスはそれらのパスに均等に分散されます。

#### **XMLFILE** *filename*

XML ファイルのための 1 つ以上の基本ファイル名を指定します。最初の名前の名前空間がいっぱいになると、2 番目の名前が使用され、以下 3 番目、4 番目と続きます。

エクスポート操作中に XML ファイルを作成するときに、まずこのリストから現行パス (*xml-path* で指定されたパス) に現行のベース名を追加し、それに 3 桁のシーケンス番号を追加し、さらに 3 文字の ID `xml` を追加したファイル名が構成されます。例えば、現行 XML パスがディレクトリー `/u/foo/xml/path/` で、現行 XML ファイル名が `bar` の場合、XML ファイルは、`/u/foo/xml/path/bar.001.xml`、`/u/foo/xml/path/bar.002.xml` などのように作成されます。

#### **MODIFIED BY** *filetype-mod*

ファイル・タイプ修飾子オプションを指定します。67 ページの『エクスポート・ユーティリティー用のファイル・タイプ修飾子』を参照してください。

#### **XMLSAVESCHEMA**

すべての XML 列について XML スキーマ情報を保存することを指定します。エクスポートされた各 XML 文書のうち、挿入時に XML スキーマに関する妥当性検査が実行されたものについては、そのスキーマの完全修飾 SQL ID が、対応する XML Data Specifier (XDS) 内に SCH 属性として格納されます。エクスポートされた文書に対して XML スキーマに関する妥当性検査が実行されていない場合、またはそのスキーマ・オブジェクトがデータベースにもやば存在しない場合は、対応する XDS に SCH 属性は含められません。

SQL ID のスキーマと名前の各部分は、XML スキーマに対応する `SYSCAT.XSROBJECTS` カタログ表の行の "OBJECTSCHEMA" および "OBJECTNAME" の値として格納されます。

**XMLSAVESCHEMA** オプションには、整形 XML 文書を生成しない XQuery シーケンスとの互換性がありません。

#### **METHOD N** *column-name*

出力ファイルで使用される 1 つ以上の列名を指定します。このパラメーターが指定されない場合、表の列名が使用されます。このパラメーターは IXF ファイルでのみ有効ですが、階層データをエクスポートするときは無効です。

### *select-statement*

エクスポートされるデータを戻す **SELECT** または **XQUERY** ステートメントを指定します。このステートメントによってエラーが発生する場合、メッセージ・ファイル (または標準出力) にメッセージが書き込まれます。エラー・コードが **SQL0012W**、**SQL0347W**、**SQL0360W**、**SQL0437W**、または **SQL1824W** である場合、エクスポート操作は続行します。これ以外のエラー・コードの場合、操作は停止します。

**SELECT** ステートメントが **SELECT \* FROM tablename** という形式で、表に暗黙的な非表示列が含まれている場合には、エクスポート操作に非表示列のデータを含めるかどうかを明示的に指定する必要があります。非表示列のデータを含めるかどうかを示すには、以下のいずれかの方法を使用します。

- 非表示列のファイル・タイプ修飾子のいずれかを使用します。非表示列のデータをエクスポートに含める場合には **implicitlyhiddeninclude** を指定し、エクスポートに含めない場合には **implicitlyhiddenmissing** を指定します。

```
db2 export to t.del of del modified by implicitlyhiddeninclude
select * from t
```

- 暗黙的な非表示列が含まれる表をデータ移動ユーティリティが検出した場合のデフォルト動作を設定するには、**DB2\_DMU\_DEFAULT** レジストリー変数をクライアント・サイドで使用します。

```
db2set DB2_DMU_DEFAULT=IMPLICITLYHIDDENINCLUDE
db2 export to t.del of del select * from t
```

### **HIERARCHY STARTING** *sub-table-name*

デフォルトの走査順序 (**ASC**、または **DEL** ファイルの **OUTER** 順序、または **PC/IXF** データ・ファイルに保管されている順序) を使用して、*sub-table-name* から始まる副階層をエクスポートします。

### **HIERARCHY traversal-order-list**

指定した走査順序を使用して副階層をエクスポートします。すべての副表は、**PRE-ORDER** 方式でリストされていなければなりません。最初の副表名が、**SELECT** ステートメントのターゲット表名として使用されます。

### **MESSAGES ON SERVER**

**EXPORT** コマンドによってサーバー上に作成されるメッセージ・ファイルを保存することを指定します。戻される結果セットには以下の 2 つの列が含まれます。1 つは **MSG\_RETRIEVAL** で、これはこの操作中に発生したすべての警告メッセージおよびエラー・メッセージを取り出すのに必要な **SQL** ステートメントです。もう 1 つは **MSG\_REMOVAL** で、これはメッセージをクリーンアップするために必要な **SQL** ステートメントです。

この節が指定されていない場合は、**ADMIN\_CMD** プロシージャから呼び出し元に戻る時点でメッセージ・ファイルが削除されます。結果セット中の **MSG\_RETRIEVAL** および **MSG\_REMOVAL** 列の内容は **NULL** 値になります。

この節が指定されているかどうかに関係なく、**fenced** ユーザー ID に、**DB2\_UTIL\_MSGPATH** レジストリー変数の示すディレクトリーおよびデータのエクスポート先ディレクトリーの下にファイルを作成するための権限が付与されていることが必要です。

## 例

次に示すのは、SAMPLE データベースにある STAFF 表から、ファイル myfile.ixf に情報をエクスポートする方法の一例です。これは、IXF フォーマットで出力されます。コマンドを発行する前に、SAMPLE データベースと接続していなければなりません。

```
CALL SYSPROC.ADMIN_CMD ('EXPORT to /home/user1/data/myfile.ixf
  OF ixf MESSAGES ON SERVER select * from staff')
```

## 使用上の注意

- **EXPORT** コマンドで使用されるすべてのパスは、サーバー上の有効な完全修飾パスでなければなりません。
- 表に **LOB** 列が含まれている場合、**LOBS TO** および **LOBFILE** 節を使用して、少なくとも 1 つの完全修飾 **LOB** パスと **LOB** 名が指定されていなければなりません。
- エクスポート・ユーティリティーは、操作の開始時に **COMMIT** ステートメントを発行しますが、これによってタイプ 2 接続の場合に、プロシージャーは **SQL30090N**、理由コード 2 を戻します。
- **UCS-2** データベースから、区切り文字で区切られている **ASCII (DEL)** ファイルにエクスポートする場合、すべての文字データは、プロシージャーを実行する場所で有効なコード・ページに変換されます。文字ストリングと **GRAPHIC** ストリング・データのどちらも、サーバーの、同じ **SBCS** または **MBCS** コード・ページに変換されます。
- エクスポート操作を開始する前に、すべての表操作が完了し、すべてのロックが解除されていることを確認してください。これは、**WITH HOLD** でオープンされたすべてのカーソルをクローズした後で **COMMIT** を発行するか、または **ROLLBACK** を発行することによって行うことができます。
- **SELECT** ステートメントでは表の別名を使用できます。
- メッセージ・ファイルに置かれたメッセージには、メッセージ検索サービスから戻される情報が含まれています。各メッセージは新しい行から始まります。
- **PC/IXF** インポートは、データベース間でデータを移動する場合に使用します。行区切り文字を含む文字データが区切り文字付き **ASCII (DEL)** ファイルにエクスポートされ、テキスト転送プログラムによって処理される場合、行区切り文字を含むフィールドは長さが変わることがあります。
- ソースとターゲットのデータベースが両方とも同じクライアントからアクセス可能である場合、ファイルのコピーというステップは必要ありません。
- **DB2 Connect** を使用して、**DB2 for OS/390®**、**DB2 for VM and VSE**、および **DB2 for OS/400®** などの **DRDA®** サーバーから表をエクスポートすることができます。 **PC/IXF** エクスポートだけがサポートされています。
- **IXF** フォーマットにエクスポートする際に、**ID** が **IXF** フォーマットでサポートされる最大サイズを超えている場合、エクスポートは成功しますが、生成されるデータ・ファイルは **CREATE** モードを使用する後続のインポート操作では使用できません。 **SQL27984W** が戻されます。
- **Windows** でディスクットにエクスポートしているときに 1 枚のディスクットの容量を超えるデータを持つ表がある場合、別のディスクットを挿入するようプロンプトが出され、複数パーツ **PC/IXF** ファイル (マルチボリューム **PC/IXF** ファ

イル、または論理的に分割された PC/IXF ファイルとして知られる) が生成されて別々のディスクに保管されます。最後のファイルを除く各ファイルには、そのファイルが論理的に分割されていること、および次のファイルを検索する場所を示すための DB2 CONTINUATION RECORD (または略して "AC" Record) が書き込まれます。その後、ファイルがインポート・ユーティリティーおよびロード・ユーティリティーによって読み取られるように、ファイルを AIX® システムに転送できます。エクスポート・ユーティリティーは、AIX システムから呼び出される場合、複数部分からなる PC/IXF ファイルを作成しません。詳しい使用方法については、**IMPORT** コマンドまたは **LOAD** コマンドを参照してください。

- エクスポート・ユーティリティーは、提供される **SELECT** ステートメントが、**SELECT \* FROM tablename** という形式である場合、IXF ファイルの表の **NOT NULL WITH DEFAULT** 属性を保管します。
- 型付き表をエクスポートする場合、副選択ステートメントは、ターゲット表名と **WHERE** 節を指定することによってのみ表現することができます。階層をエクスポートする場合、全選択と *select-statement* は指定できません。
- IXF 以外のファイル形式の場合は、階層の全探索の方法、およびエクスポートする副表とが DB2 に知らされるよう、全探索順序リストを指定することをお勧めします。このリストを指定しない場合、階層内のすべての表がエクスポートされ、デフォルトの順序は **OUTER** 順序になります。OUTER 関数によって指定されるデフォルトの順序を使うこともできます。
- インポート操作の間も同じ走査順序を使用します。ロード・ユーティリティーでは、階層または副階層のロードはサポートされていません。
- 保護行のある表からデータをエクスポートする場合は、セッション許可 ID の保持する LBAC 信用証明情報のために、エクスポートされる行が制限されることがあります。セッション許可 ID に読み取りアクセスがない行はエクスポートされません。エラーも警告も出ません。
- セッション許可 ID の保持する LBAC 信用証明情報のために、エクスポートに含まれている 1 つ以上の保護列からの読み取りが許可されない場合、エクスポートは失敗し、エラー (SQLSTATE 42512) が戻されます。
- **export** や **db2move** などのデータ移動ユーティリティーを実行する際に、照会コンパイラーが、基礎照会を基本表よりも MQT に対して実行する方が効率が良いと判別することがあります。そのような場合、照会は据え置きリフレッシュの MQT に対して実行され、ユーティリティーの結果は基礎表内のデータを正確に表していない可能性があります。
- エクスポートするパッケージは DATETIME ISO 形式を使用してバインドされるため、ストリング表記へのキャスト時にすべての日付/ 時間/ タイム・スタンプの値は ISO 形式に変換されます。CLP パッケージは DATETIME LOC 形式 (ロケール固有形式) を使用してバインドされるため、CLP DATETIME 形式が ISO と異なる場合に CLP とエクスポート間で矛盾した振る舞いが見られる場合があります。例えば、以下の **SELECT** ステートメントは、以下の期待される結果を戻しません。

```
db2 select col2 from tab1 where char(col2)='05/10/2005';
COL2
-----
05/10/2005
05/10/2005
05/10/2005
3 record(s) selected.
```

しかし、同じ select 節を使用した export コマンドでは、期待される結果が戻されません。

```
db2 export to test.del of del select col2 from test
where char(col2)='05/10/2005';
Number of rows exported: 0
```

ここで、LOCALE 日付形式を ISO 形式に置き換えると、以下のように、期待される結果になります。

```
db2 export to test.del of del select col2 from test
where char(col2)='2005-05-10';
Number of rows exported: 3
```

### 結果セット情報

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。実行が成功したなら、以下の結果セットによりコマンドから追加情報が戻されます。

表 30. EXPORT コマンドから戻される結果セット

列名	データ・タイプ	説明
ROWS_EXPORTED	BIGINT	エクスポートされた行の総数。
MSG_RETRIEVAL	VARCHAR(512)	このユーティリティーによって作成されたメッセージを取り出すために使用する SQL ステートメント。以下に例を示します。  SELECT SQLCODE, MSG FROM TABLE (SYSPROC.ADMIN_GET_MSGS ( '3203498_txu' )) AS MSG
MSG_REMOVAL	VARCHAR(512)	このユーティリティーによって作成されたメッセージをクリーンアップするために使用する SQL ステートメント。以下に例を示します。  CALL SYSPROC.ADMIN_REMOVE_MSGS ( '3203498_txu' )



## エクスポート・ユーティリティ用のファイル・タイプ修飾子

表 31. エクスポート・ユーティリティで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット

修飾子	説明
lobsinfile	<p><i>lob-path</i> には、LOB データの入ったファイルへのパスを指定します。</p> <p>各パスには、データ・ファイル内で LOB ロケーション指定子 (LLS) によって示される 1 つ以上の LOB の入った、少なくとも 1 つのファイルが組み込まれます。LLS は、LOB ファイル・パスに保管されるファイル内の LOB のロケーションのストリング表現です。LLS の形式は、<i>filename.ext.nnn.mmm/</i> です。<i>filename.ext</i> は LOB を収めたファイルの名前、<i>nnn</i> はファイル内の LOB のオフセット (バイト単位)、<i>mmm</i> は LOB の長さ (バイト単位) を表します。例えば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保管される場合、LOB はファイル <i>db2exp.001</i> のオフセット 123 に位置し、456 バイト長です。</p> <p><b>EXPORT</b> の使用時に <b>lobsinfile</b> 修飾子を指定した場合、LOB データは <b>LOBS TO</b> 節に指定されたロケーションに置かれます。指定しない場合、LOB データはデータ・ファイル・ディレクトリーに送られます。<b>LOBS TO</b> 節は、LOB ファイルが保管されるディレクトリーへのパスを 1 つ以上指定します。LOB パスごとに少なくとも 1 つのファイルが存在し、各ファイルには少なくとも 1 つの LOB が入ります。<b>LOBS TO</b> または <b>LOBFILE</b> オプションによって、<b>LOBSINFILE</b> 動作が暗黙的にアクティブ化されます。</p> <p>NULL LOB を指定するには、サイズに -1 と入力します。サイズを 0 と指定すると、長さが 0 の LOB として扱われます。長さが -1 の NULL LOB の場合、オフセットとファイル名は無視されます。例えば、NULL LOB の LLS は、<i>db2exp.001.7.-1/</i> のようになります。</p>
implicitlyhiddeninclude	<p>この修飾子は <b>SELECT *</b> 照会と一緒に使用し、<b>SELECT *</b> 照会の結果に含まれない暗黙的な非表示列のデータもエクスポートすることを指定します。この修飾子は、<b>implicitlyhiddenmissing</b> 修飾子と共に使用することはできません。</p> <p>この修飾子を使用して、照会に <b>SELECT *</b> が含まれていないと、エラーが戻りません (SQLCODE SQL3526N)。</p>
implicitlyhiddenmissing	<p>この修飾子は <b>SELECT *</b> 照会と一緒に使用し、暗黙的な非表示列のデータをエクスポートしないことを指定します。この修飾子は、<b>implicitlyhiddeninclude</b> 修飾子と共に使用することはできません。</p> <p>この修飾子を使用して、照会に <b>SELECT *</b> が含まれていないと、エラーが戻りません (SQLCODE SQL3526N)。</p>
xmlinsefiles	<p>各 XQuery データ・モデル (XDM) インスタンスが別のファイルに書き込まれます。デフォルトでは、同じファイルの中で複数の値が連結されます。</p>
lobsinsefiles	<p>各 LOB 値が別のファイルに書き込まれます。デフォルトでは、同じファイルの中で複数の値が連結されます。</p>
xmlnodeclaration	<p>XDM インスタンスが XML 宣言タグなしで書き込まれます。デフォルトでは、XDM インスタンスのエクスポート時に、エンコード属性を指定した XML 宣言タグが先頭に組み込まれます。</p>
xmlchar	<p>XDM インスタンスが文字コード・ページで書き込まれます。文字コード・ページは <b>codepage</b> ファイル・タイプ修飾子で指定されている値であるか、または指定がない場合はアプリケーションのコード・ページであることに注意してください。デフォルトでは、XDM インスタンスは、Unicode で書き込まれます。</p>

表 31. エクスポート・ユーティリティで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
xmlgraphic	<b>EXPORT</b> コマンドで xmlgraphic 修飾子を指定すると、アプリケーション・コード・ページまたは codepage ファイル・タイプ修飾子にかかわりなく、エクスポート XML 文書は、UTF-16 コード・ページでエンコードされます。

表 32. エクスポート・ユーティリティの有効なファイル・タイプ修飾子: DEL (区切り付き ASCII) ファイル形式

修飾子	説明
chardelx	<p>x は単一文字の文字区切り文字です。デフォルト値は、二重引用符 (") です。文字区切り文字を囲む二重引用符の代わりに指定の文字を使用します。<sup>2</sup> 文字区切り文字として二重引用符を明示的に指定する場合は、以下のように指定します。</p> <pre>modified by chardel""</pre> <p>以下のように、文字区切り文字として単一引用符 (') を指定することもできます。</p> <pre>modified by chardel''</pre>
codepage=x	<p>x は、ASCII 文字区切り文字です。この値は、出力データ・セットに含まれているデータのコード・ページとして解釈されます。エクスポート操作の実行中に、文字区切り文字は、アプリケーション・コード・ページからこのコード・ページに変換されます。</p> <p>DBCS のみ (GRAPHIC)、混合 DBCS、および EUC の場合、文字区切り文字の範囲は x00 から x3F に制限されます。codepage 修飾子を lobsinfile 修飾子と一緒に使用することはできません。</p>
coldelx	<p>x は単一文字の列区切り文字です。デフォルト値はコンマ (,) です。列の終わりを示すコンマの代わりに指定の文字を使用します。<sup>2</sup></p> <p>以下の例では coldel; を指定しているため、エクスポート・ユーティリティは、セミコロン文字 (;) をエクスポート・データの列区切り文字として使用します。</p> <pre>db2 "export to temp of del modified by coldel; select * from staff where dept = 20"</pre>
decplusblank	<p>正符号文字。正の 10 進数の接頭部として、正符号 (+) ではなく空白・スペースを使用します。デフォルトのアクションでは、正の 10 進数の前に正符号 (+) が付けられます。</p>
decptx	<p>x は、小数点文字としてピリオドの代わりに使用される単一の置換文字です。デフォルト値はピリオド (.) です。小数点文字として、ピリオドの代わりに指定の文字を使用します。<sup>2</sup></p>
nochardel	<p>列データが文字区切りで囲まれません。DB2 を使用してデータのインポートまたはロードを実行する場合は、このオプションを指定しないでください。これは、区切り文字を持たないベンダー・データ・ファイルをサポートするために用意されています。不適切に使用すると、データが損失または破壊される場合があります。</p> <p>このオプションを chardelx または nodoubledel と一緒に指定することはできません。これらは、相互に排他的なオプションです。</p>
nodoubledel	<p>二重文字区切りの認識を抑制します。<sup>2</sup></p>



表 32. エクスポート・ユーティリティの有効なファイル・タイプ修飾子: DEL (区切り付き ASCII) ファイル形式 (続き)

修飾子	説明
timestampformat="x"	<p>x は、ソース・ファイルのタイム・スタンプの形式です。<sup>4</sup> 有効なタイム・スタンプ・エレメントは、以下のとおりです。</p> <p><b>YYYY</b> 年 (0000 から 9999 の範囲の 4 桁の数)</p> <p><b>M</b> 月 (1 から 12 の範囲の 1 桁または 2 桁の数)</p> <p><b>MM</b> 月 (01 から 12 の 2 桁の数。M および MMM とは相互に排他的)</p> <p><b>MMM</b> 月 (大文字小文字を区別しない月名の 3 文字の省略形。M と MM とは相互に排他的)</p> <p><b>D</b> 日 (1 から 31 の範囲の 1 桁または 2 桁の数)</p> <p><b>DD</b> 日 (01 から 31 の範囲の 2 桁の数。D とは相互に排他的)</p> <p><b>DDD</b> 元日から数えた日数 (001 から 366 の範囲の 3 桁の数。他の日または月のエレメントとは相互に排他的)</p> <p><b>H</b> 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の範囲の 1 桁または 2 桁の数。)</p> <p><b>HH</b> 時 (12 時間制の場合は 00 から 12、24 時間制では 0 から 24 の範囲の 2 桁の数。H と相互に排他的)</p> <p><b>M</b> 分 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p><b>MM</b> 分 (00 から 59 の範囲の 2 桁の数。M (分) とは相互に排他的)</p> <p><b>S</b> 秒 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p><b>SS</b> 秒 (00 から 59 の範囲の 2 桁の数。S とは相互に排他的)</p> <p><b>SSSSS</b> 夜中の 12 時から数えた秒数 (00000 から 86400 の範囲の 5 桁の数。他の時刻エレメントとは相互に排他的)</p> <p><b>U (1 から 12 時)</b>                      小数秒 (U のオカレンス数は、各桁を 0 から 9 の範囲として、桁数を表します)</p> <p><b>TT</b> 午前/午後の指定子 (AM または PM)</p> <p>タイム・スタンプ・フォーマットの例を以下に示します。</p> <p style="text-align: center;">"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>MMM エレメントは、以下の値を生成します。                      「Jan」、「Feb」、「Mar」、「Apr」、「May」、「Jun」、「Jul」、「Aug」、                      「Sep」、「Oct」、「Nov」、および「Dec」。「Jan」は 1 月と等しく、「Dec」は 12 月と等しいです。</p> <p>以下の例は、「schedule」という表から、ユーザー定義のタイム・スタンプ・フォーマットを示すデータをエクスポートする方法を示しています。</p> <pre>db2 export to delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" select * from schedule</pre>

表 33. エクスポート・ユーティリティで有効なファイル・タイプ修飾子: IXF ファイル・フォーマット

修飾子	説明
codepage=x	<p>x は、ASCII 文字ストリングです。この値は、出力データ・セットに含まれているデータのコード・ページとして解釈されます。エクスポート操作中に、文字データをこのコード・ページからアプリケーション・コード・ページに変換します。</p> <p>DBCS のみ (GRAPHIC)、混合 DBCS、および EUC の場合、区切り文字の範囲は x00 から x3F に制限されます。</p>

**注:**

- サポートされていないファイル・タイプを **MODIFIED BY** オプションで使用しようとしても、エクスポート・ユーティリティは警告を出しません。サポートされていないファイル・タイプを使おうとすると、エクスポート操作は失敗し、エラー・コードが戻されます。
- データ移動のための区切り文字の制約事項に、区切り文字の指定変更として使用できる文字に適用される制限のリストが示されています。
- 通常、エクスポート・ユーティリティは、以下のような形式で書き込みを行います。

- 日付データ: *YYYYMMDD* の形式
- 文字 (日付) データ: *YYYY-MM-DD* の形式
- 時刻データ: *HH.MM.SS* の形式
- タイム・スタンプ・データ: *YYYY-MM-DD-HH.MM.SS.aaaaaa* の形式

エクスポート操作のために **SELECT** ステートメントで指定される日時列に組み込まれたデータも、これらの形式になります。

- タイム・スタンプ・フォーマットの場合、月の記述子と分の記述子のどちらも文字 **M** を使用するため、区別があいまいにならないように注意する必要があります。月のフィールドは、他の日付フィールドと隣接していなければなりません。分フィールドは、他の時刻フィールドに隣接していなければなりません。あいまいなタイム・スタンプ形式の例を以下に示します。

- "M" (月または分のどちらにもとれる)
- "M:M" (月と分の区別がつかない)
- "M:YYYY:M" (両方とも月と解釈される)
- "S:M:YYYY" (時刻値と日付値の両方に隣接している)

あいまいな場合、ユーティリティはエラー・メッセージを報告し、操作は失敗します。

以下に、明確なタイム・スタンプ・フォーマットを示します。

- "M:YYYY" (M (月))
- "S:M" (M (分))
- "M:YYYY:S:M" (M (月)...M (分))
- "M:H:YYYY:M:D" (M (分)...M (月))

- XMLFILE** 節または **XML TO** 節のどちらも指定されていない場合でも、すべての XDM インスタンスは、メイン・データ・ファイルとは別個の XML ファイルに書き込まれます。デフォルトで、XML ファイルはエクスポートされるデータ・ファイルのパスに書き込まれます。XML ファイルのデフォルトのベース名は、エクスポートされるデータ・ファイル名に拡張子 ".xml" を追加したものとなります。

6. XMLNODEDECLARATION ファイル・タイプ修飾子が指定されていない場合は、すべての XDM インスタンスはエンコード属性を含む XML 宣言を先頭に付けて書き込まれます。
7. XMLCHAR または XMLGRAPHIC ファイル・タイプ修飾子が指定されていない場合は、デフォルトで、すべての XDM インスタンスは Unicode で書き込まれます。
8. XML データおよび LOB データのデフォルト・パスは、メイン・データ・ファイルのパスです。デフォルトの XML ファイルのベース名は、メイン・データ・ファイルです。デフォルトの LOB ファイルのベース名は、メイン・データ・ファイルです。例えば、メイン・データ・ファイルが  
/mypath/myfile.del

の場合、XML データおよび LOB データのデフォルト・パスは  
/mypath"

となり、デフォルトの XML ファイルのベース名は  
myfile.del

となり、デフォルトの LOB ファイルのベース名は  
myfile.del

LOB ファイルを生成するには、LOBSINFILE ファイル・タイプ修飾子が指定されている必要があります。

9. エクスポート・ユーティリティは、数値 ID を各 LOB ファイルまたは XML ファイルに付加します。この ID は、0 が埋め込まれた 3 桁のシーケンス値で、  
.001

から始まります。999 番目の LOB ファイルまたは XML ファイルの後は、ID への 0 の埋め込みは行われません (例えば、1000 番目の LOG ファイルまたは XML ファイルの拡張子は

.1000

となります)。数値 ID の後に、データ・タイプを表す次のどちらかの 3 文字のタイプ ID が続きます。

.lob

または

.xml

例えば、生成される LOB ファイルの名前は

myfile.del.001.lob

の形式となり、生成される XML ファイルの名前は

myfile.del.001.xml

の形式となります。

10. XQuery を指定すると、エクスポート・ユーティリティは、整形形式でない文書である XDM インスタンスをエクスポートできます。ただし、XML 列に含め

ることができるのは完全な文書だけなので、これらのエクスポートされた文書を XML 列に直接インポートまたはロードすることはできません。

#### FORCE APPLICATION コマンド (ADMIN\_CMD プロシージャを使用):

システムからローカルまたはリモートのユーザーやアプリケーションを強制終了し、サーバー上での保守を可能にします。

**重要:** 割り込みが許されない操作 (例えば、**RESTORE DATABASE**) を強制終了する場合は、その操作を再び実行して正常終了しなければデータベースは使用可能になりません。

#### 有効範囲

このコマンドは、\$HOME/sql1lib/db2nodes.cfg ファイルにリストされているすべてのデータベース・パーティションに影響を与えます。

パーティション・データベース環境では、このコマンドを発行するのは、強制終了するアプリケーションのコーディネーター・データベース・パーティションからでなくてもかまいません。パーティション・データベース環境ではどのデータベース・パーティション・サーバーからでも発行できます。

#### 許可

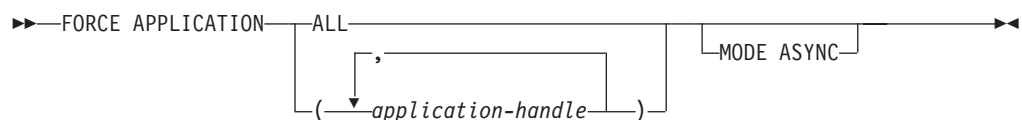
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT

#### 必要な接続

データベース

#### コマンド構文



#### コマンド・パラメーター

##### FORCE APPLICATION

**ALL** すべてのアプリケーションがデータベースから切断されます。これにより、ADMIN\_CMD プロシージャが実行されている接続がクローズされることがあります。その場合、強制終了オペレーションが正常に完了した時点で、ADMIN\_CMD プロシージャについては SQL1224N エラーが戻されることとなります。

##### *application-handle*

エージェントの終了を指定します。 **LIST APPLICATIONS** コマンドを使用して値をリストします。

## MODE ASYNC

コマンドは、指定したすべてのユーザーが終了するのを待たずに戻ってきます。コマンドは、機能を正常に発行するか、またはエラー（無効な構文などの）を発見するとすぐに戻ります。

現在サポートしているモードはこのモードだけです。

## 例

次の例は、*application-handle* の値が 41408 と 55458 の 2 つのユーザーをデータベースから強制的に切断します。

```
CALL SYSPROC.ADMIN_CMD( 'force application ( 41408, 55458 )' )
```

## 使用上の注意

データベース・マネージャーは、**db2start** を必要とせずに、後続のデータベース・マネージャー操作を処理できるようにするため、アクティブなままになっています。

データベースの整合性を確保するため、終了できるのは、アイドル中のユーザー、または割り込み可能なデータベース操作を実行中のユーザーだけです。

以下のタイプのユーザーおよびアプリケーションは、強制終了できません。

- データベースを作成しているユーザー
- システム・アプリケーション

これらのタイプのユーザーおよびアプリケーションを正常に強制終了するには、データベースの非活動化とインスタンスの再始動のいずれかまたは両方を行う必要があります。

**FORCE APPLICATION** が出された後も、データベースはまだ接続要求を受諾します。すべてのユーザーを完全に強制終了するためには、追加の **FORCE** が必要になる場合があります。

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

## ADMIN\_CMD プロシージャを使用する GET STMM TUNING コマンド:

ユーザー設定のセルフチューニング・メモリー・マネージャー (STMM) の調整メンバー番号、および現在の STMM 調整メンバー番号を報告するカタログ表の読み取りに使用します。

## 許可

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかの権限または特権が含まれていなければなりません。

- DBADM
- SECADM
- SQLADM
- ACCESSCTRL



- DATAACCESS
- SYSIBM.SYSTUNINGINFO に対する SELECT

### 必要な接続

データベース

### コマンド構文

▶▶—GET—STMM—TUNING—MEMBER—◀◀

### 例

```
CALL SYSPROC.ADMIN_CMD( 'get stmm tuning member' )
```

以下はこの照会の出力例です。

Result set 1

```
-----
USER_PREFERRED_NUMBER CURRENT_NUMBER
-----
                        2                2
```

1 record(s) selected.

Return Status = 0

### 使用上の注意

- ユーザー設定のセルフチューニング・メモリー・マネージャー (STMM) の調整メンバー番号 (USER\_PREFERRED\_NUMBER) は、ユーザーにより設定され、メモリー・チューナーの実行対象にするメンバーを指定します。データベースの稼働中に、調整メンバーは 1 時間に数度適用されます。結果として、戻される CURRENT\_NUMBER および USER\_PREFERRED\_NUMBER は、ユーザー設定の STMM メンバーの更新後にも同期がとれていない可能性があります。これを解決するために、CURRENT\_NUMBER が非同期に更新されるのを待機するか、またはデータベースをいったん停止してから開始し、CURRENT\_NUMBER の更新を強制します。

### 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** を **MEMBER** の代わりに使用できます。

### 結果セット情報

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。実行が成功した場合、コマンドは以下の結果セットで追加情報を戻します。

表 34. GET STMM TUNING コマンドで戻される結果セット

列名	データ・タイプ	説明
USER_PREFERRED_NUMBER	INTEGER	ユーザー設定のセルフチューニング・メモリー・マネージャー (STMM) の調整メンバー番号。パーティション・データベース環境では、値 -1 は、デフォルトのメンバーの使用を示します。
CURRENT_NUMBER	INTEGER	現在の STMM 調整メンバー番号。値 -1 は、デフォルトのメンバーの使用を示します。

### IMPORT コマンド (ADMIN\_CMD プロシージャーを使用):

外部ファイルのデータを、サポートされているファイル・フォーマットで表、階層、ビュー、またはニックネームに挿入します。LOAD はより高速な代替方法です。しかしロード・ユーティリティーでは、階層レベルのデータのロードはサポートされていません。

93 ページの『インポート・ユーティリティー用のファイル・タイプ修飾子』へのクイック・リンク。

### 許可

- **INSERT** オプションを使用して **IMPORT** する場合、以下の権限のいずれかが必要です。
  - DATAACCESS 権限
  - 関係するそれぞれの表、ビュー、またはニックネームに対する CONTROL 特権
  - 関係するそれぞれの表またはビューに対する INSERT および SELECT 特権
- **INSERT\_UPDATE** オプションを使用して既存の表に **IMPORT** するには、以下の権限のいずれかが必要です。
  - DATAACCESS 権限
  - 関係するそれぞれの表、ビュー、またはニックネームに対する CONTROL 特権
  - 関係するそれぞれの表またはビューに対する INSERT、SELECT、UPDATE、および DELETE 特権
- **REPLACE** または **REPLACE\_CREATE** オプションを使用して既存の表に **IMPORT** するには、以下の権限のいずれかが必要です。
  - DATAACCESS 権限
  - 表またはビューに対する CONTROL 特権
  - 表またはビューに対する INSERT、SELECT、および DELETE 特権
- **CREATE** または **REPLACE\_CREATE** オプションを使用して新規の表に **IMPORT** するには、以下の権限のいずれかが必要です。
  - DBADM 権限

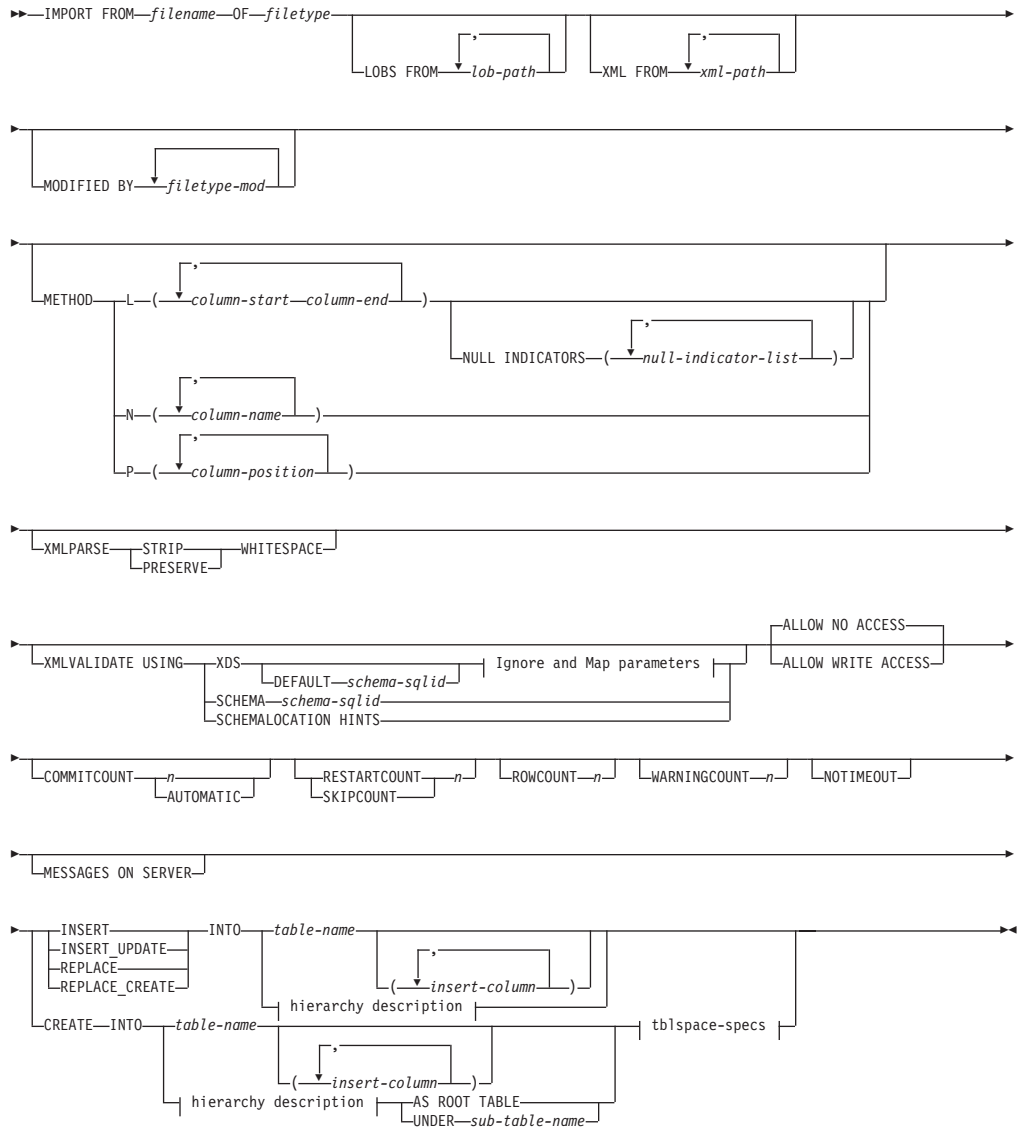
- データベースに対する **CREATETAB** 権限および表スペースに対する **USE** 特権に加えて、以下のいずれかが必要です。
  - データベースに対する **IMPLICIT\_SCHEMA** 権限 (表の暗黙的または明示的スキーマ名が存在しない場合)
  - スキーマに対する **CREATEIN** 特権 (表のスキーマ名が既存のスキーマを指す場合)
- **CREATE** または **REPLACE\_CREATE** オプションを使って、存在しない階層に **IMPORT** するには、以下の権限のいずれかが必要です。
  - **DBADM** 権限
  - データベースに対する **CREATETAB** 権限および表スペースに対する **USE** 特権と、以下のいずれかが必要です。
    - データベースに対する **IMPLICIT\_SCHEMA** 権限 (表のスキーマ名が存在しない場合)
    - スキーマに対する **CREATEIN** 特権 (表のスキーマが存在する場合)
    - 階層全体に対して **REPLACE\_CREATE** オプションが使用されている場合は、階層内のすべての副表に対する **CONTROL** 特権
- **REPLACE** オプションを使用して既存の階層に **IMPORT** するには、以下の権限のどれかが必要です。
  - **DATAACCESS** 権限
  - 階層内のすべての副表に対する **CONTROL** 特権
- 保護列のある表にデータをインポートするには、セッション許可 ID に、その表内のすべての保護列への書き込みアクセスを許可する **LBAC** 信用証明情報が必要です。そうでない場合、インポートは失敗し、エラー (SQLSTATE 42512) が戻されます。
- 保護されている行のある表にデータをインポートするには、セッション許可 ID に、以下の基準を満たす **LBAC** 信用証明情報が必要です。
  - 表を保護しているセキュリティ・ポリシーの一部である
  - 書き込みアクセスに関して、セッション許可 ID に付与された挿入する行のラベル、ユーザーの **LBAC** 信用証明情報、セキュリティ・ポリシー定義、および **LBAC** 規則が行のラベルを決定します。
- **REPLACE** または **REPLACE\_CREATE** オプションが指定された場合、セッション許可 ID には、その表をドロップするための権限が付与されていなければなりません。
- ニックネームにデータをインポートするには、セッション許可 ID には指定したデータ・ソースにパススルー・モードでアクセスおよび使用する特権がなければなりません。
- 行アクセス制御がアクティブになっている表に対して **IMPORT REPLACE** を使用する場合は、その表をドロップするための権限が必要です。具体的には、表に対する **CONTROL** または **DBADM** のいずれかが必要になります。

### 必要な接続

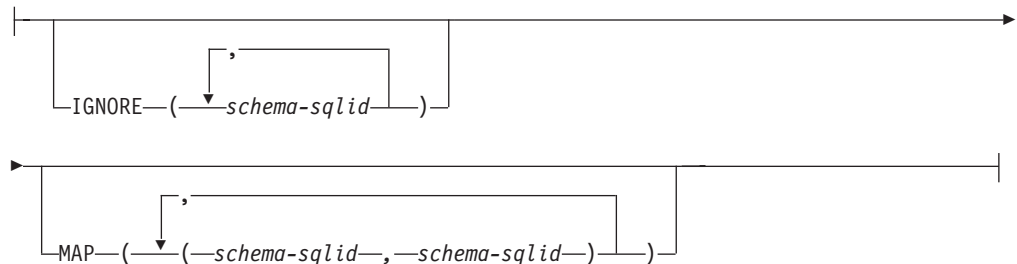
データベース。Linux、UNIX、または Windows クライアントから Linux、UNIX、または Windows データベース・サーバーへのユーティリティ・アクセスは、DB2

Connect ゲートウェイまたはループバック環境を経由してではなく、エンジンを使用した直接接続でなければなりません。

## コマンド構文



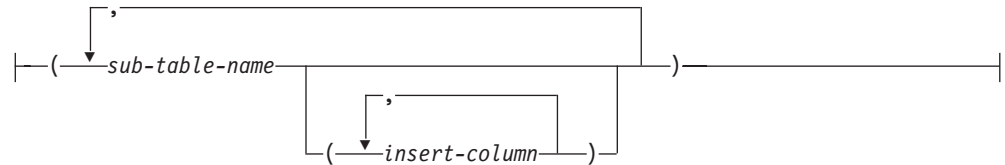
### Ignore and Map parameters:



### hierarchy description:



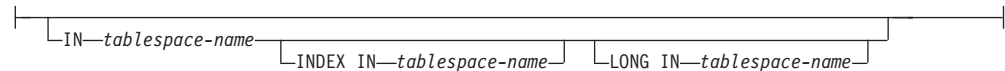
### sub-table-list:



### traversal-order-list:



### tblspace-specs:



## コマンド・パラメーター

### FROM filename

インポートするデータの入ったファイルの名前を指定します。これは完全修飾パスでなければならず、データベース・サーバー上にそのファイルが存在していなければなりません。

### OF filetype

入力ファイル内のデータのフォーマットを指定します。

- ASC (区切りなし ASCII フォーマット)
- DEL (区切り文字付き ASCII フォーマット)。さまざまなデータベース・マネージャーやファイル・マネージャー・プログラムで使用します。
- IXF (統合交換フォーマット、PC バージョン) は、DB2 によって排他的に使用されるバイナリー・フォーマットです。

### LOBS FROM lob-path

LOB ファイルを保管する 1 つ以上の完全修飾パスを指定します。それらのパスは、データベース・サーバーのコーディネーター・パーティション上に存在していなければなりません。LOB データ・ファイルの名前は、メイン・データ・ファイル (ASC、DEL、または IXF) の、LOB 列にロードされる列内に保管されます。指定できるパスの最大数は 999 です。これによって、LOBSINFILE 動作が暗黙的にアクティブ化されます。

ニックネームにインポートするときには、このパラメーターは無効です。

#### XML FROM *xml-path*

XML ファイルが含まれているパスを 1 つ以上指定します。

#### MODIFIED BY *filetype-mod*

ファイル・タイプ修飾子オプションを指定します。93 ページの『インポート・ユーティリティー用のファイル・タイプ修飾子』を参照してください。

#### METHOD

- L** データのインポートを開始する列および終了する列の番号を指定します。列の番号は、データの行の先頭からのバイト単位のオフセットです。この番号は 1 から始まります。

注: このメソッドは、ASC ファイルの場合にのみ使用することができ、そのファイル・タイプに対してのみ有効なオプションです。

- N** インポートするデータ・ファイルの中の列の名前を指定します。これらの列名の大きい文字小文字の区別は、システム・カタログ内の対応する名前の大きい文字小文字の区別と一致しなければなりません。NULL 可能ではない各表の列には、**METHOD N** リスト内に対応する項目が必要です。例えば、データ・フィールドが F1、F2、F3、F4、F5、および F6 であり、表の列が C1 INT、C2 INT NOT NULL、C3 INT NOT NULL、および C4 INT の場合、**method N** (F2, F1, F4, F3) は有効な要求ですが、**method N** (F2, F1) は無効です。

注: この方式は、IXF ファイルの場合にのみ使用することができます。

- P** インポートする入力データ・フィールドのフィールド番号 (1 から始まる) を指定します。NULL 可能ではない各表の列には、**METHOD P** リスト内に対応する項目が必要です。例えば、データ・フィールドが F1、F2、F3、F4、F5、および F6 であり、表の列が C1 INT、C2 INT NOT NULL、C3 INT NOT NULL、および C4 INT の場合、**method P** (2, 1, 4, 3) は有効な要求ですが、**method P** (2, 1) は無効です。この方式を使用できるのは、ファイル・タイプ IXF または DEL の場合のみです。また、DEL ファイル・タイプに対しては、これが唯一有効な方式です。

**method P** によって指定したフィールドごとに、対応する列をアクション・ステートメント内に定義する必要があります。ただし、次の例に示すようにすべての列が対象になる場合、および最初の *x* 列がロードされる場合は必要ありません。

```
db2 load from datafile1.del of del method P(1, 3, 4)
      replace into table1 (c1, c3, c4)
```

#### NULL INDICATORS *null-indicator-list*

このオプションは、**METHOD L** パラメーターを指定した場合にのみ使用できます。つまり、入力ファイルが ASC ファイルの場合です。NULL 標識リストは、コンマで区切られた正の整数のリストで、各 NULL 標識フィールドの列の番号を指定します。列の番号は、データの行の先頭からのバイト単位の、各 NULL 標識フィールドのオフセットです。NULL 標識リストには、**METHOD L** パラメーターで定義された各データ・フィールドに対する 1

つの項目がなければなりません。列の番号がゼロであることは、対応するデータ・フィールド内に必ずデータがあることを示します。

NULL 標識列中の Y の値は、その列データが NULL であることを指定します。NULL 標識列に Y 以外の文字を指定した場合は、列データが NULL ではなく、**METHOD L** オプションで指定された列データがインポートされることを指定することになります。

nullindchar ファイル・タイプ修飾子を指定した **MODIFIED BY** オプションを使用すれば、NULL 標識文字を変更することができます。

#### **XMLPARSE**

XML 文書の解析方法を指定します。このオプションが指定されていない場合、XML 文書の解析の動作は、CURRENT XMLPARSE OPTION 特殊レジスターの値によって決まります。

#### **STRIP WHITESPACE**

XML 文書の解析時に空白文字を除去することを指定します。

#### **PRESERVE WHITESPACE**

XML 文書の解析時に空白文字を除去しないことを指定します。

#### **XMLVALIDATE**

該当する場合に、XML 文書がスキーマに準拠しているかどうかの妥当性検査を実行することを指定します。

#### **USING XDS**

メイン・データ・ファイルの中で XML Data Specifier (XDS) によって指定されている XML スキーマに準拠しているかどうかについて、XML 文書の妥当性検査が実行されます。デフォルトでは、**USING XDS** 節によって **XMLVALIDATE** オプションが呼び出された場合、妥当性検査実行のために使用されるスキーマは、その XDS の SCH 属性によって決まります。XDS の中で SCH 属性が指定されていない場合、**DEFAULT** 節によってデフォルト・スキーマが指定されているのでない限り、スキーマ妥当性検査は実行されません。

**DEFAULT**、**IGNORE**、および **MAP** 節を使用することにより、スキーマ決定の動作を変更することができます。これら 3 つの節はオプションであり、相互に適用されるのではなく XDS の指定に直接適用されます。例えば、**DEFAULT** 節で指定されているためにあるスキーマが選択された場合、それが **IGNORE** 節で指定されていたとしても無視されることはありません。同じように、**MAP** 節のペアの最初の部分で指定されているためにあるスキーマが選択された場合、それが別の **MAP** 節のペアの 2 番目の部分で指定されていたとしても再びマップされることはありません。

#### **USING SCHEMA *schema-sqlid***

指定されている SQL ID の XML スキーマに準拠しているかどうかについて、XML 文書の妥当性検査が実行されます。この場合、すべての XML 列について XML Data Specifier (XDS) の SCH 属性は無視されます。

#### **USING SCHEMALOCATION HINTS**

ソース XML 文書の中で XML スキーマ・ロケーション・ヒントに

よって指定されているスキーマに準拠しているかどうかについて、XML 文書の妥当性検査が実行されます。その XML 文書の中に schemaLocation 属性が指定されていない場合、妥当性検査は実行されません。**USING SCHEMALOCATION HINTS** 節が指定されているなら、すべての XML 列について XML Data Specifier (XDS) の SCH 属性は無視されます。

以下のセクションに示す **XMLVALIDATE** オプションの例を参照してください。

#### **DEFAULT** *schema-sqlid*

このオプションは、**USING XDS** パラメーターを指定した場合にのみ使用できます。**DEFAULT** 節で指定されたスキーマは、インポート対象 XML 文書の XML Data Specifier (XDS) に XML スキーマを指定する SCH 属性が含まれていない場合に、妥当性検査のために使用するスキーマとなります。

**DEFAULT** 節は、**IGNORE** 節および **MAP** 節よりも優先されます。XDS が **DEFAULT** 節を満たすなら、**IGNORE** と **MAP** の指定は無視されます。

#### **IGNORE** *schema-sqlid*

このオプションは、**USING XDS** パラメーターを指定した場合にのみ使用できます。**IGNORE** 節は、SCH 属性によって指定されていても無視するスキーマとして、1 つ以上のスキーマのリストを指定します。インポートする XML 文書の XML Data Specifier の中に SCH 属性が存在し、その SCH 属性によって指定されるスキーマが無視するスキーマ・リストに含まれている場合には、インポートするその XML 文書についてスキーマ妥当性検査は実行されません。

あるスキーマが **IGNORE** 節の中で指定されている場合、**MAP** 節のスキーマ・ペアの左辺にそれを含めることはできません。

**IGNORE** 節は XDS にのみ適用されます。あるスキーマが **IGNORE** 節によって指定されていても、それが **MAP** 節によってマップされているなら、それ以降そのスキーマが無視されることはありません。

#### **MAP** *schema-sqlid*

このオプションは、**USING XDS** パラメーターを指定した場合にのみ使用できます。**MAP** 節は、インポートする各 XML 文書について XML Data Specifier (XDS) の SCH 属性によって指定されるスキーマの代わりに使用する代替スキーマを指定するのに使用します。**MAP** 節には、それぞれがあるスキーマから別のスキーマへのマッピングを表すスキーマ・ペアを 1 つ以上列挙したリストを指定します。ペア中の最初のスキーマは、XDS 内の SCH 属性によって参照されるスキーマを表します。ペア中の 2 番目のスキーマは、スキーマ妥当性検査の実行で使用する必要のあるスキーマを表します。

あるスキーマが **MAP** 節のスキーマ・ペアの左辺で指定されている場合、**IGNORE** 節でさらにそれを指定することはできません。

スキーマ・ペアのマッピングが適用されたなら、その結果は最終的なものです。マッピング操作は推移的ではないため、選択されたスキーマが、それ以降に別のスキーマ・ペアのマッピングに適用されることはありません。



スキーマを複数回マップすることはできません。つまり、複数のペアの左辺に指定することはできません。

#### **ALLOW NO ACCESS**

オフライン・モードでインポートを実行します。行の挿入の前には常に、ターゲット表に排他 (X) ロックがかけられます。これで、同時アプリケーションは表データにアクセスできなくなります。これがデフォルトのインポート動作です。

#### **ALLOW WRITE ACCESS**

オンライン・モードでインポートを実行します。最初の行の挿入時には、ターゲット表に意図的排他 (IX) ロックがかけられます。これで、表データへの同時の読み取りおよび書き出しアクセスが可能になります。オンライン・モードには、**REPLACE**、**CREATE**、または **REPLACE\_CREATE** インポート・オプションとの互換性はありません。オンライン・モードとバッファー挿入との連携はサポートされません。インポート操作によって挿入後のデータが定期的にコミットされるので、表ロックへのロック・エスカレーションが削減され、アクティブなログ・スペースが使い果たされることはなくなります。このようなコミットは、**COMMITCOUNT** オプションを使わなくても実行されます。各コミットごとに、インポートでは IX 表ロックが外されるので、コミットの完了後に再びロックの設定が試みられます。ニックネームにインポートするときにはこのパラメーターが必要で、有効な数値を使って **COMMITCOUNT** を指定する必要があります (**AUTOMATIC** は有効なオプションとは見なされません)。

#### **COMMITCOUNT *n* | AUTOMATIC**

*n* 個のレコードがインポートされるたびに **COMMIT** を実行します。数 *n* を指定すると、インポートでは *n* 個のレコードのインポートの後にそのつど **COMMIT** が実行されます。コンパウンド挿入を使用した場合、ユーザー指定のコミット頻度 *n* は、そのコンパウンド・カウント値に最も近い整数の倍数に切り上げられます。 **AUTOMATIC** を指定すると、コミットの必要時期はインポート操作で内部的に判別されます。次の 2 つのうちのいずれかの理由で、このユーティリティーはコミットを行います。

- アクティブ・ログ・スペースを使いきらないようにするため。
- ロックが行レベルから表レベルにエスカレーションしないようにするため。

**ALLOW WRITE ACCESS** オプションを指定した場合に **COMMITCOUNT** オプションを指定しないと、インポート・ユーティリティーは、**COMMITCOUNT AUTOMATIC** が指定されたものとしてコミットを実行します。

アクティブ・ログ・スペースを使いきらないようにする点でのインポート操作の能力は、以下のように **DB2** レジストリー変数

**DB2\_FORCE\_APP\_ON\_MAX\_LOG** の影響を受けます。

- **DB2\_FORCE\_APP\_ON\_MAX\_LOG** を **FALSE** にして、**COMMITCOUNT AUTOMATIC** コマンド・オプションを指定した場合、インポート・ユーティリティーは自動的にアクティブ・ログ・スペースを使い切らないようにすることができます。
- **DB2\_FORCE\_APP\_ON\_MAX\_LOG** を **FALSE** にして、**COMMITCOUNT *n*** コマンド・オプションを指定した場合、インポート・ユーティリティーはレコードを

挿入または更新しようとして SQL0964C (トランザクション・ログがフル) を検出したら、ログ・フル状態が解決しようとし、無条件のコミットを実行してから、レコードの挿入または更新を再試行します。これで問題が解決しない場合 (ログ・フルがデータベース上の他のアクティビティに起因する場合など)、予期されるように **IMPORT** コマンドは失敗しますが、コミットされる行数は **COMMITCOUNT**  $n$  値の倍数にならないことがあります。インポート操作を再試行する際に既にコミットされた行を処理しないようにするには、**RESTARTCOUNT** または **SKIPCOUNT** コマンド・パラメーターを使用します。

- **DB2\_FORCE\_APP\_ON\_MAX\_LOG** が TRUE (これがデフォルトです) の場合、レコードの挿入または更新中に SQL0964C を検出すると、インポート操作は失敗します。これは、**COMMITCOUNT** **AUTOMATIC** と **COMMITCOUNT**  $n$  のどちらを指定したかに関係なく発生します。

アプリケーションは強制的にデータベースから切断され、現在の作業単位はロールバックされます。インポート操作を再試行する際に既にコミットされた行を処理しないようにするには、**RESTARTCOUNT** または **SKIPCOUNT** コマンド・パラメーターを使用します。

#### **RESTARTCOUNT** $n$

$n + 1$  の位置のレコードからインポート操作を開始するよう指定します。最初の  $n$  個のレコードはスキップされます。このオプションは機能的には **SKIPCOUNT** と同等です。**RESTARTCOUNT** と **SKIPCOUNT** は相互に排他的です。

#### **SKIPCOUNT** $n$

$n + 1$  の位置のレコードからインポート操作を開始するよう指定します。最初の  $n$  個のレコードはスキップされます。このオプションは機能的には **RESTARTCOUNT** と同等です。**SKIPCOUNT** と **RESTARTCOUNT** は相互に排他的です。

#### **ROWCOUNT** $n$

インポート (挿入または更新) するファイル内の物理レコードの数  $n$  を指定します。ユーザーは、**SKIPCOUNT** または **RESTARTCOUNT** オプションで指示されたレコードから始めて、ファイルの  $n$  行だけをインポートすることができます。**SKIPCOUNT** または **RESTARTCOUNT** オプションの指定がないと、最初の  $n$  行がインポートされます。**SKIPCOUNT**  $m$  または **RESTARTCOUNT**  $m$  を指定すると、行  $m+1$  から  $m+n$  がインポートされます。コンパウンド挿入を使用した場合、ユーザー指定の **ROWCOUNT**  $n$  は、そのコンパウンド・カウンタ値に最も近い整数の倍数に切り上げられます。

#### **WARNINGCOUNT** $n$

$n$  個の警告後に、インポート操作を停止します。このパラメーターは、警告は予期されないが、正しいファイルと表が使用されていることを確認するのが望ましい場合に設定してください。インポート・ファイルまたはターゲット表が不適切に指定されると、インポート対象の各行ごとにインポート・ユーティリティーによって警告が生成され、このためにインポートが失敗する可能性があります。 $n$  をゼロにした場合や、このオプションを指定しない場合、発行された警告の回数に関係なくインポート操作は続行します。

#### **NOTIMEOUT**

インポート・ユーティリティーがロックの待機中にタイムアウトしないこと

を指定します。このオプションのほうが、**locktimeout** データベース構成パラメーターより優先されます。他のアプリケーションは影響を受けません。

#### MESSAGES ON SERVER

**IMPORT** コマンドによってサーバー上に作成されるメッセージ・ファイルを保存することを指定します。戻される結果セットには以下の 2 つの列が含まれます。1 つは **MSG\_RETRIEVAL** で、これはこの操作中に発生したすべての警告メッセージおよびエラー・メッセージを取り出すのに必要な SQL ステートメントです。もう 1 つは **MSG\_REMOVAL** で、これはメッセージをクリーンアップするために必要な SQL ステートメントです。

この節が指定されていない場合は、**ADMIN\_CMD** プロシージャーから呼び出し元に戻る時点でメッセージ・ファイルが削除されます。結果セット中の **MSG\_RETRIEVAL** および **MSG\_REMOVAL** 列の内容は **NULL** 値になります。

この節が指定されているかどうかに関係なく、**fenced** ユーザー ID に、**DB2\_UTIL\_MSGPATH** レジストリー変数の示すディレクトリーおよびデータのエクスポート先ディレクトリーの下にファイルを作成するための権限が付与されていることが必要です。

**INSERT** 既存の表データを変更することなく、インポートされたデータを表に追加します。

#### INSERT\_UPDATE

インポートしたデータ行をターゲット表に追加するか、または主キーが一致するものがあればターゲット表の既存行を更新します。

#### REPLACE

データ・オブジェクトを切り捨てることによって表内の既存のデータすべてを削除してから、インポートしたデータを挿入します。表定義および索引定義は変更されません。表がない場合は、このオプションを使用できません。階層間でデータを移動する際にこのオプションを使用する場合は、階層全体に関係したデータだけが置き換えられます。副表は置き換えられません。

ニックネームにインポートするときには、このパラメーターは無効です。

このオプションでは、**CREATE TABLE** ステートメントの **NOT LOGGED INITIALLY (NLI)** 節、あるいは **ALTER TABLE** ステートメントの **ACTIVE NOT LOGGED INITIALLY** 節は考慮されません。

このオプションは、システム期間テンポラル表へのデータのインポートには使用できません。

**NLI** 節が呼び出される **CREATE TABLE** または **ALTER TABLE** ステートメントと同じトランザクションの中で、**REPLACE** オプションの指定されたインポートが実行された場合、インポートにおいてその **NLI** 節は考慮されません。すべての挿入操作がログ対象となります。

#### 予備手段 1

**DELETE** ステートメントを使用して表の内容を削除した後、  
**INSERT** ステートメントによりインポートを呼び出す

## 予備手段 2

表をドロップしてからそれを再作成した後、INSERT ステートメントによってインポートを呼び出す

この制限は、DB2 Universal Database™ バージョン 7 および DB2 バージョン 8 に適用されます。

## REPLACE\_CREATE

**注: REPLACE\_CREATE** パラメーターは推奨されておらず、今後のリリースで除去される可能性があります。さらに詳しくは、『IMPORT コマンドの推奨されなくなったオプション CREATE および REPLACE\_CREATE』を参照してください。

表が既にある場合には、データ・オブジェクトを切り捨てることによって表内の既存のデータすべてを削除し、表定義や索引定義は変えることなく、インポートしたデータを挿入します。

表がまだない場合には、データベースのコード・ページで、表と索引の定義と行の内容を作成します。制約事項のリストは、『インポート済みの表の再作成』を参照してください。

このオプションは、IXF ファイルの場合にのみ使用することができます。階層間でデータを移動する際にこのオプションを使用する場合は、階層全体に関係したデータだけが置き換えられます。副表は置き換えられません。

ニックネームにインポートするときには、このパラメーターは無効です。

## INTO table-name

データのインポート先となるデータベース表を指定します。この表として、システム表、作成済み一時表、宣言済み一時表、またはサマリー表は指定できません。

以前のサーバーの場合を除き、**INSERT**、**INSERT\_UPDATE**、または **REPLACE** オプションには、完全修飾または非修飾の表名を使用しなければならないようなときでも、別名を使用することができます。修飾された表名は、*schema.tablename* の形式になります。 *schema* には、表作成時のユーザー名が入ります。

暗黙的な非表示列がデータベース表に含まれている場合には、インポート操作に非表示列のデータを含めるかどうかを指定する必要があります。非表示列のデータを含めるかどうかを示すには、以下のいずれかの方法を使用します。

- データの挿入先の列を明示的に指定する場合、*insert-column* を使用します。

```
db2 import from delfile1 of del
  insert into table1 (c1, c2, c3,...)
```

- 非表示列のファイル・タイプ修飾子のいずれかを使用します。入力ファイルに非表示列のデータが含まれる場合には **implicitlyhiddeninclude** を指定し、入力ファイルに含まれない場合には **implicitlyhiddenmissing** を指定します。

```
db2 import from delfile1 of del modified by implicitlyhiddeninclude
  insert into table1
```

- 暗黙的な非表示列が含まれる表をデータ移動ユーティリティーが検出した場合のデフォルト動作を設定するには、DB2\_DMU\_DEFAULT レジストリー変数をクライアント・サイドで使用します。

```
db2set DB2_DMU_DEFAULT=IMPLICITLYHIDDENINCLUDE
db2 import from delfile1 of del insert into table1
```

#### *insert-column*

データの挿入先となる表またはビュー内の列名を指定します。

#### **ALL TABLES**

階層専用の暗黙キーワード。階層をインポートする場合、走査順序で指定されるすべての表をインポートすることがデフォルトです。

#### *sub-table-list*

型付き表で **INSERT** または **INSERT\_UPDATE** オプションを指定した場合、データのインポート先副表を指定するために副表名のリストが使われます。

#### **HIERARCHY**

階層データをインポートするよう指定します。

#### **STARTING** *sub-table-name*

階層専用キーワード。 *sub-table-name* から始まるデフォルト順を要求します。 PC/IXF ファイルの場合、デフォルト順は入力ファイルに保管されている順です。 PC/IXF ファイル・フォーマットの場合、デフォルト順は有効な唯一の順序です。

#### *traversal-order-list*

型付き表で **INSERT**、**INSERT\_UPDATE**、または **REPLACE** オプションを指定した場合、インポートする階層内の副表の横断順序を指定するために副表名のリストを使います。

#### **CREATE**

**注:** **CREATE** パラメーターは推奨されておらず、今後のリリースで除去される可能性があります。さらに詳しくは、『**IMPORT** コマンドの推奨されなくなったオプション **CREATE** および **REPLACE\_CREATE**』を参照してください。

データベースのコード・ページで表の定義と行の内容を作成します。 DB2 の表、副表、または階層からエクスポートされたデータの場合、索引も作成されます。このオプションが階層に対するものである場合に、DB2 からデータがエクスポートされると、タイプ階層も作成されます。このオプションは、IXF ファイルの場合にのみ使用することができます。

ニックネームにインポートするときには、このパラメーターは無効です。

**注:** データが MVS™ ホスト・データベースからエクスポートされたもので、ページ・サイズで計算した長さが 254 より大きい LONGVAR フィールドを含んでいる場合、**CREATE** は行が長過ぎるために失敗します。制約事項のリストは、『インポート済みの表の再作成』を参照してください。この場合、その表は手動で作成します。そして、**IMPORT** に **INSERT** を指定して呼び出すか、または **LOAD** コマンドを使用してください。

#### **AS ROOT TABLE**

1 つ以上の副表を、独立した表階層として作成します。

**UNDER** *sub-table-name*

1 つ以上の副表を作成する場合に親表を指定します。

**IN** *tablespace-name*

表を作成する表スペースを指定します。表スペースは存在している必要があり、REGULAR 表スペースでなければなりません。他の表スペースを指定しない場合、すべての表パーツはこの表スペースに保管されます。この節を指定しない場合、表は許可 ID によって作成された表スペース中に作成されます。何も検出されない場合、その表はデフォルト表スペースの USERSPACE1 に入れられます。USERSPACE1 がドロップされていた場合、表作成は失敗します。

**INDEX IN** *tablespace-name*

表の索引を作成する表スペースを指定します。このオプションは、**IN** 節で指定される PRIMARY 表スペースが DMS 表スペースである場合のみ使用できます。指定した表スペースは存在している必要があり、かつ REGULAR または LARGE DMS 表スペースでなければなりません。

**注:** どの表スペースに索引を配置するかは、表を作成するときのみ指定できます。

**LONG IN** *tablespace-name*

ロング列の値 (LONG VARCHAR、LONG VARGRAPHIC、LOB データ・タイプ、またはソース・タイプとしてこれらが指定されている特殊タイプ) を保管する表スペースを指定します。このオプションは、**IN** 節で指定した PRIMARY 表スペースが DMS 表スペースである場合のみ使用できます。指定した表スペースは存在している必要があり、LARGE DMS 表スペースでなければなりません。

**例**

次に示すのは、ファイル myfile.ixf から SAMPLE データベースの STAFF 表に情報をインポートする方法の例です。

```
CALL SYSPROC.ADMIN_CMD
('IMPORT FROM /home/userid/data/myfile.ixf
OF IXF MESSAGES ON SERVER INSERT INTO STAFF')
```

**使用上の注意**

**IMPORT** コマンドで使用されるすべてのパスは、サーバーのコーディネーター・データベース・パーティション上の有効な完全修飾パスでなければなりません。

オプションとして **ALLOW WRITE ACCESS** または **COMMITCOUNT** が指定されている場合、インポート・ユーティリティによってコミットが実行されます。その場合、タイプ 2 の接続では、ADMIN\_CMD プロシージャから、理由コード 1 の SQL30090N エラーが戻されます。

ADMIN\_CMD プロシージャからの結果セットの列に代入される値が、その列のデータ・タイプの最大値より大きい場合、そのデータ・タイプの最大値が代入され、警告メッセージ SQL1155W が戻されます。

インポート操作を開始する前に、すべての表操作が完了し、すべてのロックがペンディング解除になっていることを確認してください。これは、WITH HOLD でオープンされた、すべてのカーソルをクローズした後で COMMIT または ROLLBACK を発行することによって行われます。

インポート・ユーティリティーは、SQL INSERT ステートメントを使用してターゲット表に行を追加します。このユーティリティーは、入力ファイル中の各行のデータにつき 1 つずつ INSERT ステートメントを発行します。INSERT ステートメントが失敗した場合、以下の 2 通りの結果のいずれかになります。

- 後続の INSERT ステートメントが成功すると予測される場合には、警告メッセージがメッセージ・ファイルに書き込まれ、処理が継続されます。
- 後続の INSERT ステートメントが失敗すると予測され、データベースが損傷する可能性がある場合には、エラー・メッセージがメッセージ・ファイルに書き込まれ、処理が停止されます。

このユーティリティーは、REPLACE または REPLACE\_CREATE 操作時に以前の行が削除された後、自動 COMMIT を実行します。したがって、表オブジェクトが切り捨てられた後、システムに障害が起こったり、アプリケーションがデータベース・マネージャーに割り込んだりすると、元のデータがすべて失われてしまいます。これらのオプションを使用する前に、元のデータがもはや必要ないことを確認してください。

CREATE、REPLACE、または REPLACE\_CREATE 操作時にログが満杯になると、このユーティリティーは挿入されたレコードに対して自動 COMMIT を実行します。自動 COMMIT の後に、システムに障害が起こるか、またはアプリケーションがデータベース・マネージャーに割り込むと、部分的にデータの挿入された表はデータベース内に残ります。REPLACE または REPLACE\_CREATE オプションを使用してインポート操作全体をやり直すか、または正常にインポートされる行の数に設定した RESTARTCOUNT パラメーターを指定して INSERT を使用してください。

IMPORT コマンドからの更新は、常に IMPORT タスクの終わりにコミットされます。IMPORT コマンドの実行時に自動コミットを実行して、ロック・リストのサイズとアクティブ・ログ・スペースを減らすこともできます。IMPORT 処理中にアクティブ・ログがいっぱいになると、IMPORT コマンドはロールバックします。

- デフォルトでは、INSERT または INSERT\_UPDATE オプションについては自動コミットは実行されません。しかし、COMMITCOUNT パラメーターがゼロでない場合は実行されます。
- 以下のいずれかの条件が真であると、オフライン・インポートでは自動の COMMIT は実行されません。
  - ターゲットは表ではなくビューである。
  - コンパウンド挿入を使用している。
  - バッファ挿入を使用している。
- デフォルトでは、オンライン・インポートは自動コミットを実行して、アクティブ・ログ・スペースとロック・リストを両方とも解放します。自動コミットが実行されないのは、ゼロの COMMITCOUNT 値を指定した場合のみです。

インポート・ユーティリティーが COMMIT を実行するたびに、2 つのメッセージがメッセージ・ファイルに書き込まれます。一方は、コミットされるレコードの数

を示し、もう一方は、COMMIT の成功後に書き込まれます。障害の後にインポート操作を再開するときには、スキップするレコードの数 (最後の正常なコミットから判別される) を指定してください。

インポート・ユーティリティーでは、多少の非互換性問題がある入力データは受け入れられます (例えば、文字データは埋め込みまたは切り捨てを用いてインポートできます。数値データは異なる数値データ・タイプを用いてインポートできます)。しかし、大きな非互換性問題のあるデータは受け入れられません。

オブジェクト表に何らかの従属 (それ自体への従属は除く) がある場合は、そのオブジェクト表を **REPLACE** または **REPLACE\_CREATE** することはできません。また、オブジェクト・ビューの基本表に何らかの従属 (それ自体への従属を含む) がある場合は、そのオブジェクト・ビューを **REPLACE** または **REPLACE\_CREATE** することはできません。そのような表またはビューを置換するには、以下のとおりに行ってください。

1. その表が親となっているすべての外部キーをドロップします。
2. インポート・ユーティリティーを実行します。
3. 表を変更して、外部キーを再作成します。

外部キーの再作成中にエラーが発生する場合、参照整合性を保守するためにデータを変更してください。

参照制約および外部キー定義は、PC/IXF ファイルから表を再作成する場合は保存されません。(主キー定義は、データが前に SELECT \* を使ってエクスポートされた場合、保存されます。)

リモート・データベースへのインポートでは、サーバーに、入力データ・ファイルのコピー、出力メッセージ・ファイル、およびデータベースのサイズ拡大を見込んだ十分なディスク・スペースが必要とされます。

インポート操作がリモート・データベースに対して実行され、出力メッセージ・ファイルが非常に長くなった (60 KB を超過) 場合、クライアントのユーザーに戻されるメッセージ・ファイルで、インポート操作中にメッセージが欠落する可能性があります。メッセージ情報の最初の 30 KB と最後の 30 KB は、常に保持されます。

PC/IXF ファイルのリモート・データベースへのインポートは、PC/IXF ファイルがディスクにあるときよりも、ハード・ディスクにあるときの方がより速く行うことができます。

データベース表または階層が存在していないと、ASC または DEL ファイル形式のデータをインポートできません。ただし、表が存在していない場合でも、**IMPORT CREATE** または **IMPORT REPLACE\_CREATE** は、PC/IXF ファイルからデータをインポートする際に表を作成します。型付き表の場合、**IMPORT CREATE** はタイプ階層と表階層も作成することができます。

PC/IXF インポートは、データベース間でデータ (階層データも含む) を移動する場合に使用します。行区切り文字を含む文字データが区切り文字付き ASCII (DEL) ファイルにエクスポートされ、テキスト転送プログラムによって処理される場合、行区切り文字を含むフィールドは長さが変わることがあります。ソースとターゲット



のデータベースが両方とも同じクライアントからアクセス可能である場合、ファイルのコピーというステップは必要ありません。

ASC および DEL ファイルのデータは、インポートを実行するクライアント・アプリケーションのコード・ページであると仮定されます。異なるコード・ページのデータをインポートする場合は、異なるコード・ページを使用することのできる PC/IXF ファイルをお勧めします。PC/IXF ファイルとインポート・ユーティリティーが同じコード・ページである場合は、通常のアプリケーションの場合のように処理が行われます。それぞれのコード・ページが異なっており、**FORCEIN** オプションが指定されている場合、インポート・ユーティリティーは、PC/IXF ファイルのデータのコード・ページと、インポートを実行中のアプリケーションのコード・ページが同じであると見なします。この処理は、それら 2 つのコード・ページ用の変換テーブルが存在する場合であっても行われます。それぞれのコード・ページが異なっており、**FORCEIN** オプションが指定されておらず、変換テーブルが存在する場合、PC/IXF ファイルのすべてのデータは、そのファイルのコード・ページからアプリケーションのコード・ページに変換されます。それぞれのコード・ページが異なっており、**FORCEIN** オプションが指定されておらず、変換テーブルが存在しない場合、インポート操作は失敗します。これが該当するのは、AIX オペレーティング・システムの DB2 クライアント上の PC/IXF ファイルの場合だけです。

8 KB ページ上の表オブジェクトの量が 1012 列の制限に近い場合、PC/IXF データ・ファイルをインポートすると、SQL ステートメントの最大サイズを超過するため、DB2 はエラーを戻します。この状態が発生する可能性があるのは、列が CHAR、VARCHAR、または CLOB タイプの場合だけです。DEL または ASC ファイルのインポートには、この制限事項は適用されません。PC/IXF ファイルを使って新しい表を作成している場合、別の方法として、**db2look** を使って表を作成した DDL ステートメントをダンプしてから、そのステートメント CLP から発行する、という方法があります。

DB2 Connect を使用して、DB2 for OS/390、DB2 for VM and VSE、および DB2 for OS/400 などの DRDA サーバーにデータをインポートすることができます。サポートされているのは、PC/IXF インポート (**INSERT** オプション) だけです。**RESTARTCOUNT** パラメーターもサポートされていますが、**COMMITCOUNT** パラメーターはサポートされていません。

型付き表で **CREATE** オプションを使用するときは、PC/IXF ファイルで定義されているすべての副表を作成してください。副表の定義は変更できません。型付き表で **CREATE** 以外のオプションを使用するときは、全探索順序リストによって全探索順序を指定できます。このため、全探索順序リストはエクスポート操作時に使用したものと一致する必要があります。PC/IXF ファイル形式の場合は、ターゲット副表の名前を指定して、ファイルに格納されている全探索順序を使用するだけです。

インポート・ユーティリティーは、以前 PC/IXF ファイルにエクスポートされた表をリカバリーする場合に使用できます。その表は、エクスポート時の状態に戻ります。

システム表、作成済み一時表、宣言済み一時表、またはサマリー表にデータをインポートすることはできません。

インポート・ユーティリティーを介してビューを作成することはできません。

それぞれのパーツが Windows システムから AIX システムにコピーされる、複数のパーツからなる PC/IXF ファイルのインポートがサポートされています。最初のファイルの名前だけ、**IMPORT** コマンドで指定する必要があります。例えば、**IMPORT FROM data.ixf OF IXF INSERT INTO TABLE1** のようにします。ファイル data.002 などが、data.ixf と同じディレクトリーで使用できます。

Windows オペレーティング・システムの場合は、以下のとおりです。

- 論理分割された PC/IXF ファイルのインポートはサポートされていません。
- 不正な形式の PC/IXF ファイルのインポートは、サポートされていません。

内部形式のセキュリティー・ラベルには、改行文字が含まれている可能性があります。DEL ファイル形式を使用してファイルをインポートする場合、それらの改行文字が間違っていて区切りと解釈される可能性があります。この問題が発生する場合は、**IMPORT** コマンドで **delprioritychar** ファイル・タイプ修飾子を指定することによって、区切り文字に関して以前に使用されていた古いデフォルト優先順位を使用してください。

暗黙的な非表示列がデータベース表に含まれている場合には、インポート操作に非表示列のデータを含めるかどうかを指定する必要があります。

**IMPORT** ユーティリティーは、表内の列数とデータ・ファイル内のフィールド数の突き合わせを行いません。このユーティリティーは、データ・ファイル内に十分な量のデータがあるかどうかをチェックします。データ・ファイル内の行に十分なデータ列が含まれていない場合、その行は拒否されて警告メッセージが表示される (表の対応するデータの無い列が **NOT NULL** として定義されている場合) か、警告メッセージは表示されずに正常に挿入されます (表の対応する列が **NULL** として定義されている場合)。一方、行に必要な数より多くの列が含まれている場合、必要な数の列が処理され、残りのデータ列は省略されます。警告メッセージは表示されません。

### フェデレーテッドに関する考慮事項

**IMPORT** コマンドで **INSERT**、**UPDATE**、または **INSERT\_UPDATE** コマンド・パラメーターを使用するときには、関係するニックネームに対する **CONTROL** 特権があることを確認してください。インポート操作で使いたいニックネームが既に存在することを確認する必要があります。そのほかにも、**IMPORT** コマンド・パラメーターのセクションに記載されているようないくつかの制約事項に注意する必要があります。

ODBC などの一部のデータ・ソースでは、ニックネームへのインポートはサポートされていません。

### 結果セット情報

コマンドの実行状況は、**CALL** ステートメントからの結果である **SQLCA** で戻されます。実行が成功したなら、以下の結果セットによりコマンドから追加情報が戻されます。

表 35. *IMPORT* コマンドから戻される結果セット

列名	データ・タイプ	説明
ROWS_READ	BIGINT	インポート中にファイルから読み取られたレコードの数。
ROWS_SKIPPED	BIGINT	挿入または更新を開始する前にスキップしたレコードの数。
ROWS_INSERTED	BIGINT	ターゲット表に挿入された行数。
ROWS_UPDATED	BIGINT	インポートされたレコード (主キーの値が既に表内に存在するレコード) からの情報によって更新された、ターゲット表内の行数。
ROWS_REJECTED	BIGINT	インポートできなかったレコード数。
ROWS_COMMITTED	BIGINT	正常にインポートされ、データベースにコミット済みのレコード数。
MSG_RETRIEVAL	VARCHAR(512)	このユーティリティによって作成されたメッセージを取り出すために使用する SQL ステートメント。以下に例を示します。  <pre>SELECT SQLCODE, MSG FROM TABLE (SYSPROC.ADMIN_GET_MSGS ('1203498_txu')) AS MSG</pre>
MSG_REMOVAL	VARCHAR(512)	このユーティリティによって作成されたメッセージをクリーンアップするために使用する SQL ステートメント。以下に例を示します。  <pre>CALL SYSPROC.ADMIN_REMOVE_MSGS ('1203498_txu')</pre>

### インポート・ユーティリティ用のファイル・タイプ修飾子

表 36. インポート・ユーティリティの有効なファイル・タイプ修飾子: すべてのファイル形式

修飾子	説明
compound= <i>x</i>	<p><i>x</i> は 1 から 100 の数字です。非アトミック・コンパウンド SQL を使用してデータを挿入します。毎回 <i>x</i> 個のステートメントが試行されます。</p> <p>この修飾子が指定され、トランザクション・ログに十分な大きさがいない場合、インポート操作は失敗します。トランザクション・ログは、<b>COMMITCOUNT</b> によって指定された行数か、または <b>COMMITCOUNT</b> が指定されていない場合はデータ・ファイルの行数を入れる十分な大きさが必要です。したがって、トランザクション・ログのオーバーフローを避けるために、<b>COMMITCOUNT</b> オプションを指定することをお勧めします。</p> <p>この修飾子は、<b>INSERT_UPDATE</b> モード、階層表、および修飾子 <b>usedefaults</b>、<b>identitymissing</b>、<b>identityignore</b>、<b>generatedmissing</b>、<b>generatedignore</b> とは互換性がありません。</p>
generatedignore	<p>この修飾子は、インポート・ユーティリティに、すべての生成列のデータはデータ・ファイルに存在するが、それらを見捨てるべきことを知らせます。この結果として、生成列のすべての値は、このユーティリティによって生成されます。この修飾子は、<b>generatedmissing</b> 修飾子と共に使用することはできません。</p>

表 36. インポート・ユーティリティの有効なファイル・タイプ修飾子: すべてのファイル形式 (続き)

修飾子	説明
generatedmissing	この修飾子が指定されている場合、ユーティリティは、生成列のデータが入力データ・ファイルに入っていない (NULL も入っていない) のものと見なし、行ごとに値を生成します。この修飾子は、generatedignore 修飾子と共に使用することはできません。
identityignore	この修飾子は、インポート・ユーティリティに、ID 列のデータはデータ・ファイルに存在するが、それらを見捨てるべきことを知らせます。この結果として、すべて ID 値はこのユーティリティによって生成されます。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT のどちらの ID 列の場合も同じです。つまり、GENERATED ALWAYS 列の場合には、リジェクトされる行はありません。この修飾子は、identitymissing 修飾子とともに使用することはできません。
identitymissing	この修飾子を指定すると、ユーティリティは、ID 列のデータが入力データ・ファイルに入っていない (NULL も入っていない) のものと見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT のどちらの ID 列の場合も同じです。この修飾子は、identityignore 修飾子とともに使用することはできません。
implicitlyhiddeninclude	この修飾子が指定されている場合、ユーティリティは、入力データ・ファイルに暗黙非表示列へのデータが含まれると見なし、そのデータもインポートします。この修飾子は、implicitlyhiddenmissing 修飾子と共に使用することはできません。複数の修飾子が指定されている場合の優先順位については、注: セクションを参照してください。
implicitlyhiddenmissing	この修飾子を指定すると、ユーティリティは入力データ・ファイルに暗黙的な非表示列のデータが含まれないと見なし、ユーティリティがそれらの暗黙的な非表示列の値を生成します。この修飾子は、implicitlyhiddeninclude 修飾子と共に使用することはできません。複数の修飾子が指定されている場合の優先順位については、注: セクションを参照してください。
lobsinfile	<p><i>lob-path</i> には、LOB データの入ったファイルへのパスを指定します。</p> <p>各パスには、データ・ファイル内で LOB ロケーション指定子 (LLS) によって示される 1 つ以上の LOB の入った、少なくとも 1 つのファイルが組み込まれます。LLS は、LOB ファイル・パスに保管されるファイル内の LOB のロケーションのストリング表現です。LLS の形式は、<i>filename.ext.nnn.mmm/</i> です。<i>filename.ext</i> は LOB を収めたファイルの名前、<i>nnn</i> はファイル内の LOB のオフセット (バイト単位)、<i>mmm</i> は LOB の長さ (バイト単位) を表します。例えば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保管される場合、LOB はファイル <i>db2exp.001</i> のオフセット 123 に位置し、456 バイト長です。</p> <p><b>LOBS FROM</b> 節は、「lobsinfile」修飾子が使用されているときの、LOB ファイルの場所を指定します。<b>LOBS FROM</b> 節によって、LOBSINFILE 動作が暗黙的にアクティブ化されます。<b>LOBS FROM</b> 節は、データのインポート中に、<b>IMPORT</b> ユーティリティに LOB ファイルを検索するためのパスのリストを送ります。</p> <p>NULL LOB を指定するには、サイズに -1 と入力します。サイズを 0 と指定すると、長さが 0 の LOB として扱われます。長さが -1 の NULL LOB の場合、オフセットとファイル名は無視されます。例えば、NULL LOB の LLS は、<i>db2exp.001.7.-1/</i> のようになります。</p>

表 36. インポート・ユーティリティの有効なファイル・タイプ修飾子: すべてのファイル形式 (続き)

修飾子	説明
no_type_id	1 つの副表にインポートする場合にのみ有効です。これを使う場合として典型的な例は、REGULAR 表からデータをエクスポートした後、この修飾子を使ってインポート操作を呼び出してそのデータを単一の副表に変換する場合です。
nodefaults	ターゲット表の列に対応するソース列が明示的に指定されていない場合、その表列が NULL 不可能なら、デフォルト値はロードされません。このオプションを指定せず、あるターゲット表列のためのソース列が明示的に指定されていない場合、以下のいずれかになります。 <ul style="list-style-type: none"> <li>• 列にデフォルト値を指定できる場合、そのデフォルト値がロードされます。</li> <li>• 列が NULL 可能で、デフォルト値がその列に指定できない場合、NULL がロードされます。</li> <li>• 列が NULL 不可能で、デフォルト値がその列に指定できない場合、エラーが戻され、ユーティリティは処理を停止します。</li> </ul>
norowwarnings	リジェクトされた行についてのすべての警告を抑止します。
periodignore	この修飾子は、インポート・ユーティリティに対して、期間列へのデータがデータ・ファイル内に存在しても無視するように通知します。この修飾子が指定された場合、すべての期間列の値はユーティリティによって生成されます。この修飾子は、 <b>periodmissing</b> 修飾子と共に使用することはできません。
periodmissing	この修飾子が指定されている場合、ユーティリティは、期間列のデータが入力データ・ファイルに入っていないものと見なします。この修飾子が指定された場合、すべての期間列の値はユーティリティによって生成されます。この修飾子は、 <b>periodignore</b> 修飾子と共に使用することはできません。
rowchangetimestampignore	この修飾子は、インポート・ユーティリティに、行変更タイム・スタンプの列のデータがデータ・ファイルに存在するが、それらは無視すべきことを知らせます。この結果、すべての ROW CHANGE TIMESTAMP はユーティリティによって生成されます。この動作は、GENERATED ALWAYS 列でも GENERATED BY DEFAULT 列でも同じです。つまり、GENERATED ALWAYS 列の場合には、リジェクトされる行はありません。この修飾子は、rowchangetimestampmissing 修飾子とともに使用することはできません。
rowchangetimestampmissing	この修飾子を指定すると、ユーティリティは、行変更タイム・スタンプ列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものと見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS 列でも GENERATED BY DEFAULT 列でも同じです。この修飾子は、rowchangetimestampignore 修飾子と共に使用することはできません。
seclabelchar	入力ソース・ファイル内のセキュリティー・ラベルが、デフォルトのエンコードされた数値形式ではなく、セキュリティー・ラベル値のストリング形式であることを示します。IMPORT は、各セキュリティー・ラベルをロードする際に内部形式に変換します。ストリングが適切な形式ではない場合、行はロードされず、警告 (SQLSTATE 01H53) が戻されます。ストリングが表を保護するセキュリティー・ポリシーの一部である有効なセキュリティー・ラベルを表していない場合、行はロードされず、警告 (SQLSTATE 01H53、SQLCODE SQL3243W) が戻されます。  seclabelname 修飾子が指定されている場合には、この修飾子は指定できません。指定すると、インポートは失敗して、エラー (SQLCODE SQL3525N) が戻されません。

表 36. インポート・ユーティリティの有効なファイル・タイプ修飾子: すべてのファイル形式 (続き)

修飾子	説明
seclabelname	<p>入力ソース・ファイル内のセキュリティ・ラベルが、デフォルトのエンコードされた数値形式ではなく、名前によって指定されることを示します。 <b>IMPORT</b> はその名前を適切なセキュリティ・ラベル (存在する場合) に変換します。表を保護するセキュリティ・ポリシーで、指定された名前のセキュリティ・ラベルが存在しない場合、行はロードされず、警告 (SQLSTATE 01H53、SQLCODE SQL3244W) が戻されます。</p> <p>seclabelchar 修飾子が指定されている場合には、この修飾子は指定できません。指定すると、インポートは失敗して、エラー (SQLCODE SQL3525N) が戻されます。</p> <p><b>注:</b> ファイル・タイプが ASC の場合、セキュリティ・ラベルの名前の後にスペースがあれば、それも名前の一部として解釈されます。これを回避するには、striptblanks ファイル・タイプ修飾子を使用してスペースが除去されるようにします。</p>
transactionidignore	<p>この修飾子は、インポート・ユーティリティに対して、TRANSACTION START ID 列へのデータがデータ・ファイル内に存在しても無視するように通知します。この修飾子が指定された場合、TRANSACTION START ID 列の値はユーティリティによって生成されます。この修飾子は、 <b>transactionidmissing</b> 修飾子と共に使用することはできません。</p>
transactionidmissing	<p>この修飾子が指定されている場合、ユーティリティは、TRANSACTION START ID 列のデータが入力データ・ファイルに入っていないものと見なします。この修飾子が指定された場合、TRANSACTION START ID 列の値はユーティリティによって生成されます。この修飾子は、 <b>transactionidignore</b> 修飾子と共に使用することはできません。</p>
usedefaults	<p>ターゲット表の列のソース列が指定されているが、1 つ以上の行インスタンスのデータが入っていない場合は、デフォルト値がロードされます。欠落データの例を以下に示します。</p> <ul style="list-style-type: none"> <li>• DEL ファイルの場合: 列の値として、2 つの連続した列区切り (,) や、任意の数のスペースで分離した 2 つの連続する列区切り (, ) が指定されている。</li> <li>• DEL/ASC ファイルの場合: 列が不足している行、または元の指定には十分な長さでない行。</li> </ul> <p><b>注:</b> ASC ファイルの場合: NULL 列値は明示的に欠落していると思われず、NULL 列値にはデフォルトが置換されません。数値、日付、時刻、タイム・スタンプの列では、全桁スペース文字で NULL 列値を表記します。また、どのタイプの列でも、NULL INDICATOR を使用すれば、その列が NULL であることを示せます。</p> <p>このオプションが指定されていない場合、行インスタンスのソース列にデータがないと、以下のいずれかの処理が行われます。</p> <ul style="list-style-type: none"> <li>• DEL/ASC ファイルの場合: 列が NULL 可能であれば、NULL がロードされます。列が NULL 可能でない場合、ユーティリティはその行をリジェクトします。</li> </ul>

表 37. インポート・ユーティリティで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL)

修飾子	説明
codepage=x	<p>x は、ASCII 文字ストリングです。この値は、入力データ・セット内のデータのコード・ページとして解釈されます。インポート操作中に、文字データをこのコード・ページからアプリケーション・コード・ページに変換します。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>• 純 DBCS (GRAPHIC)、混合 DBCS、および EUC では、区切り文字は x00 から x3F の範囲に制限されます。</li> <li>• nullindchar では、標準の ASCII セットのコード・ポイント x20 から x7F の範囲に含まれているシンボルを指定する必要があります。これは、ASCII 記号およびコード・ポイントを示します。</li> </ul> <p><b>注:</b></p> <ol style="list-style-type: none"> <li>1. codepage 修飾子を lobsinfile 修飾子と一緒に使用することはできません。</li> <li>2. コード・ページがアプリケーションのコード・ページからデータベースのコード・ページに変換されているときにデータの拡張が発生する場合は、データは切り捨てられ、データの消失が発生する可能性があります。</li> </ol>
dateformat="x"	<p>x はソース・ファイルの日付のフォーマットです。<sup>2</sup> 有効な日付要素は以下のとおりです。</p> <p><b>YYYY</b> 年 (0000 から 9999 の範囲の 4 桁の数)</p> <p><b>M</b> 月 (1 から 12 の範囲の 1 桁または 2 桁の数)</p> <p><b>MM</b> 月 (01 から 12 の 2 桁の数。M とは相互に排他的)</p> <p><b>D</b> 日 (1 から 31 の範囲の 1 桁または 2 桁の数)</p> <p><b>DD</b> 日 (01 から 31 の範囲の 2 桁の数。D とは相互に排他的)</p> <p><b>DDD</b> 元日から数えた日数 (001 から 366 の範囲の 3 桁の数。他の日または月の要素とは相互に排他的)</p> <p>デフォルト値の 1 が、指定されない各要素に割り当てられます。日付形式の例を以下に示します。</p> <p>"D-M-YYYY"  "MM.DD.YYYY"  "YYYYDDD"</p>
implieddecimal	<p>暗黙指定されている小数点の位置が列定義によって決定され、値の終わりにあるとは見なされなくなります。例えば、値 12345 は、12345.00 ではなく、123.45 として DECIMAL(8,2) 列にロードされます。</p>

表 37. インポート・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
timeformat="x"	<p>x はソース・ファイル内の時刻のフォーマットです。<sup>2</sup> 有効な時刻エレメントは以下のとおりです。</p> <p><b>H</b> 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の範囲の 1 桁または 2 桁の数。)</p> <p><b>HH</b> 時 (12 時間制の場合は 00 から 12、24 時間制では 0 から 24 の範囲の 2 桁の数。H と相互に排他的)</p> <p><b>M</b> 分 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p><b>MM</b> 分 (00 から 59 の範囲の 2 桁の数。M とは相互に排他的)</p> <p><b>S</b> 秒 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p><b>SS</b> 秒 (00 から 59 の範囲の 2 桁の数。S とは相互に排他的)</p> <p><b>SSSSS</b> 夜中の 12 時から数えた秒数 (00000 から 86400 の範囲の 5 桁の数。他の時刻エレメントとは相互に排他的)</p> <p><b>TT</b> 午前/午後の指定子 (AM または PM)</p> <p>指定されない各エレメントには、デフォルト値の 0 が割り当てられます。時刻フォーマットの例を以下に示します。</p> <p>"HH:MM:SS"  "HH.MM TT"  "SSSSS"</p>



表 37. インポート・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
timestampformat="x"	<p>x はソース・ファイル内のタイム・スタンプのフォーマットです。<sup>2</sup> 有効なタイム・スタンプ・エレメントは以下のとおりです。</p> <p><b>YYYY</b> 年 (0000 から 9999 の範囲の 4 桁の数)</p> <p><b>M</b> 月 (1 から 12 の範囲の 1 桁または 2 桁の数)</p> <p><b>MM</b> 月 (01 から 12 の 2 桁の数。M および MMM とは相互に排他的)</p> <p><b>MMM</b> 月 (大文字小文字を区別しない月名の 3 文字の省略形。M と MM とは相互に排他的)</p> <p><b>D</b> 日 (1 から 31 の範囲の 1 桁または 2 桁の数)</p> <p><b>DD</b> 日 (01 から 31 の範囲の 2 桁の数。D とは相互に排他的)</p> <p><b>DDD</b> 元日から数えた日数 (001 から 366 の範囲の 3 桁の数。他の日または月のエレメントとは相互に排他的)</p> <p><b>H</b> 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の範囲の 1 桁または 2 桁の数。)</p> <p><b>HH</b> 時 (12 時間制の場合は 00 から 12、24 時間制では 0 から 24 の範囲の 2 桁の数。H と相互に排他的)</p> <p><b>M</b> 分 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p><b>MM</b> 分 (00 から 59 の範囲の 2 桁の数。M (分) とは相互に排他的)</p> <p><b>S</b> 秒 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p><b>SS</b> 秒 (00 から 59 の範囲の 2 桁の数。S とは相互に排他的)</p> <p><b>SSSSS</b> 夜中の 12 時から数えた秒数 (00000 から 86400 の範囲の 5 桁の数。他の時刻エレメントとは相互に排他的)</p> <p><b>U (1 から 12 時)</b>  <small>小数秒 (U のオカレンス数は、各桁を 0 から 9 の範囲として、桁数を表します)</small></p> <p><b>TT</b> 午前/午後の指定子 (AM または PM)</p> <p>YYYY、M、MM、D、DD、または DDD エレメントが指定されていない場合、デフォルト値の 1 が割り当てられます。MMM エレメントが指定されていない場合、デフォルト値の「Jan」が割り当てられます。他のエレメントが指定されていない場合には、デフォルト値の 0 が割り当てられます。タイム・スタンプ・フォーマットの例を以下に示します。</p> <p style="text-align: center;">"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>MMM エレメントの有効な値は、「jan」、「feb」、「mar」、「apr」、「may」、「jun」、「jul」、「aug」、「sep」、「oct」、「nov」、および「dec」です。これらの値では、大/小文字は区別されません。</p> <p>次の例では、ユーザ一定義の日時形式を指示するデータを、schedule という表にインポートする方法を示します。</p> <pre>db2 import from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>

表 37. インポート・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
usegraphiccodepage	<p>usegraphiccodepage が指定された場合、GRAPHIC または 2 バイト文字ラージ・オブジェクト (DBCLOB) データ・フィールドにインポートされるデータは、GRAPHIC コード・ページであると見なされます。データの残りは、文字コード・ページであると見なされます。GRAPHIC コード・ページは、文字コード・ページと関連付けられます。IMPORT は、codepage 修飾子 (指定されている場合)、または codepage 修飾子が指定されていない場合はアプリケーションのコード・ページを介して、文字コード・ページを決定します。</p> <p>この修飾子は、リカバリされている表に GRAPHIC データがある場合のみ、ドロップ済み表のリカバリによって生成された区切りデータ・ファイルとともに使用される必要があります。</p> <p><b>制約事項</b></p> <p><b>EXPORT</b> ユーティリティーで作成された DEL ファイルは、1 つのコード・ページのみでエンコードされたデータを含んでいるため、これらのファイルで usegraphiccodepage 修飾子を指定することはできません。usegraphiccodepage 修飾子はまた、ファイル内の 2 バイト文字ラージ・オブジェクト (DBCLOB) には無視されます。</p>
xmlchar	<p>XML 文書が文字コード・ページでエンコードされていることを示します。</p> <p>このオプションは、指定の文字コード・ページでエンコードされていてもエンコード宣言を含まない XML 文書进行处理するために役立ちます。</p> <p>各文書で、宣言タグが存在してエンコード属性が含まれる場合、そのエンコード方式は文字コード・ページと一致する必要があります。一致しない場合、その文書を含む行はリジェクトされます。文字コード・ページは codepage ファイル・タイプ修飾子で指定されている値であるか、または指定がない場合はアプリケーションのコード・ページであることに注意してください。デフォルトでは、文書は Unicode でエンコードされているか、またはエンコード属性のある宣言タグを含んでいます。</p>
xmlgraphic	<p>XML 文書が指定された GRAPHIC コード・ページでエンコードされていることを示します。</p> <p>このオプションは、特定の GRAPHIC コード・ページでエンコードされていてもエンコード宣言を含まない XML 文書进行处理するために役立ちます。</p> <p>各文書で、宣言タグが存在してエンコード属性が含まれる場合、そのエンコード方式は GRAPHIC コード・ページと一致する必要があります。一致しない場合、その文書を含む行はリジェクトされます。GRAPHIC コード・ページは codepage ファイル・タイプ修飾子で指定されている値のグラフィック・コンポーネントであるか、または指定がない場合はアプリケーションのコード・ページのグラフィック・コンポーネントであることに注意してください。デフォルトでは、文書は Unicode でエンコードされているか、またはエンコード属性のある宣言タグを含んでいます。</p> <p><b>注:</b> xmlgraphic 修飾子が <b>IMPORT</b> コマンドで指定されている場合、インポートされる XML 文書は、UTF-16 コード・ページでエンコードされていなければなりません。そうでない場合、XML 文書は構文解析エラーで拒否されるか、またはデータ破損のある状態で表にインポートされる場合があります。</p>

表 38. インポート・ユーティリティーで有効なファイル・タイプ修飾子: ASC (区切り文字で区切られていない ASCII) ファイル・フォーマット

修飾子	説明
nochecklengths	nochecklengths が指定されていると、ソース・データの列定義がターゲット表の列のサイズを超えるものであっても、各行のインポートが試行されます。このような行が正常にインポートされるのは、コード・ページ変換でソース・データが縮小する場合です。例えば、ソースにある 4 バイトの EUC データがターゲットで 2 バイトの DBCS データに縮小すれば、必要スペースは半分になります。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。
nullindchar=x	$x$ は、単一文字です。NULL 値を示す文字を $x$ に変更します。 $x$ のデフォルト値は $\backslash$ です。 <sup>3</sup>  文字が 1 つの英字である場合を除いて、この修飾子は EBCDIC データ・ファイルで大文字小文字を区別します。例えば、NULL 標識文字を文字 $N$ に指定した場合、 $n$ も NULL 標識と認識されます。
reclen=x	$x$ は、32 767 以下の整数です。各行ごとに $x$ 個の文字が読み取られ、行の終わりを示すのに改行文字は使用されません。
striptblanks	データを可変長フィールドにロードする際に、後書きブランク・スペースを切り捨てます。このオプションを指定しない場合、ブランク・スペースはそのまま保持されます。  次の例の場合、インポート・ユーティリティーは、striptblanks によって後書きブランク・スペースを切り捨てます。  <pre>db2 import from myfile.asc of asc modified by striptblanks method 1 (1 10, 12 15) messages msgs.txt insert into staff</pre> このオプションは、striptnulls と一緒に指定することはできません。これらは、相互に排他的なオプションです。このオプションは、廃止された t オプション (下位互換性のためだけにサポートされる) に代わるものです。
striptnulls	データを可変長フィールドにロードする際に、後書き NULL (0x00 文字) を切り捨てます。このオプションを指定しない場合、NULL はそのまま保持されます。  このオプションは、striptblanks と一緒に指定することはできません。これらは、相互に排他的なオプションです。このオプションは、廃止された padwithzero オプション (下位互換性のためだけにサポートされる) に代わるものです。

表 39. インポート・ユーティリティーで有効なファイル・タイプ修飾子: DEL (区切り文字で区切られている ASCII ファイル・フォーマット)

修飾子	説明
chardelx	<p>x は単一文字のストリング区切り文字です。デフォルト値は二重引用符 (") です。指定した文字は、文字ストリングを囲むために、二重引用符の代わりに使用されます。<sup>34</sup> 文字ストリング区切り文字として明示的に二重引用符を指定する場合、次のように指定します。</p> <pre>modified by chardel"</pre> <p>単一引用符 (') も、文字ストリングの区切り文字として指定できます。以下の例では、chardel' が指定されており、インポート・ユーティリティーは検出するすべての単一引用符 (') を文字ストリングの区切り文字として解釈します。</p> <pre>db2 "import from myfile.del of del modified by chardel'" method p (1, 4) insert into staff (id, years)"</pre>
coldelx	<p>x は単一文字の列区切り文字です。デフォルト値はコンマ (,) です。指定した文字は、列の終わりを表すために、コンマの代わりに使用されます。<sup>34</sup></p> <p>以下の例では、coldel; が指定されており、インポート・ユーティリティーは検出するすべてのセミコロン (;) を列の区切り文字として解釈します。</p> <pre>db2 import from myfile.del of del modified by coldel; messages msgs.txt insert into staff</pre>
decplusblank	<p>正符号文字。正の 10 進値の接頭部として、正符号 (+) ではなくブランク・スペースを使用します。デフォルトのアクションでは、正の 10 進数の前に正符号 (+) が付けられます。</p>
decptx	<p>x は、小数点文字としてピリオドの代わりに使用される単一の置換文字です。デフォルト値はピリオド (.) です。指定した文字は、小数点文字としてピリオドの代わりに使用されます。<sup>34</sup></p> <p>以下の例では、decpt; が指定されており、インポート・ユーティリティーは検出するすべてのセミコロン (;) を小数点として解釈します。</p> <pre>db2 "import from myfile.del of del modified by chardel'" decpt; messages msgs.txt insert into staff"</pre>
delprioritychar	<p>区切り文字の現在のデフォルト優先順位は、(1) レコード区切り文字、(2) 区切り文字、(3) 列区切り文字です。この修飾子を使用すると、区切り文字の優先順位が (1) 区切り文字、(2) レコード区切り文字、(3) 列区切り文字に戻り、以前の優先順位に依存している既存のアプリケーションが保護されます。構文は以下のとおりです。</p> <pre>db2 import ... modified by delprioritychar ...</pre> <p>例えば、以下のような DEL データ・ファイルがあるとします。</p> <pre>"Smith, Joshua",4000,34.98&lt;row delimiter&gt; "Vincent,&lt;row delimiter&gt;, is a manager", ... ... 4005,44.37&lt;row delimiter&gt;</pre> <p>delprioritychar 修飾子を指定しているため、このデータ・ファイルは、2 行だけになります。2 番目の &lt;row delimiter&gt; は 2 番目の行の最初のデータ列の一部と解釈されますが、1 番目と 3 番目の &lt;row delimiter&gt; は実レコードの区切り文字と解釈されます。この修飾子が指定されていない場合、このデータ・ファイルでは 3 行になり、各行は &lt;row delimiter&gt; によって区切られます。</p>

表 39. インポート・ユーティリティで有効なファイル・タイプ修飾子: DEL (区切り文字で区切られている ASCII) ファイル・フォーマット (続き)

修飾子	説明
keepblanks	タイプが CHAR、VARCHAR、LONG VARCHAR、または CLOB の各フィールドの前後の空白を保持します。このオプションを指定しないと、区切り文字で囲まれていないすべての前後の空白は除去され、表のすべての空白・フィールドに NULL が挿入されます。
nochardel	インポート・ユーティリティは、列区切り文字の間にあるすべてのバイトを列のデータの一部であると見なします。文字区切り文字は、列データの一部として構文解析されます。データが DB2 を使用してエクスポートされている場合は、このオプションを指定しないでください (エクスポート時に nochardel が指定されない限り)。これは、区切り文字を持たないベンダー・データ・ファイルをサポートするために用意されています。不適切に使用すると、データが損失または破壊される場合があります。  このオプションを chardelex、delprioritychar または nodoubledel と一緒に指定することはできません。これらは、相互に排他的なオプションです。
nodoubledel	二重文字区切りの認識を抑止します。

表 40. インポート・ユーティリティで有効なファイル・タイプ修飾子: IXF ファイル・フォーマット

修飾子	説明
forcein	コード・ページが不一致でもデータを受け入れ、コード・ページ間の変換を抑止するようにユーティリティに指示します。  固定長ターゲット・フィールドに、そのデータが入るだけの十分な大きさがあるかどうかチェックされます。nochecklengths が指定されていると、チェックは実行されず、各行のインポートが試行されます。
indexixf	既存の表に現在定義されている索引をすべてドロップし、PC/IXF ファイルの索引定義に基づいて新しい索引を作成するようにユーティリティに指示します。このオプションを使用できるのは、表の内容を置換する場合だけです。ビューでは使用できません。また、insert-column が指定されている場合にも使用できません。
indexschema=schema	指定した schema を、索引作成時の索引名として使用します。schema を指定しなかった場合 (しかしキーワード indexschema は指定した 場合) には、接続ユーザー ID が使用されます。このキーワードを指定しない場合、IXF ファイルのスキーマが使用されます。
nochecklengths	nochecklengths が指定されていると、ソース・データの列定義がターゲット表の列のサイズを超えるものであっても、各行のインポートが試行されます。このような行が正常にインポートされるのは、コード・ページ変換でソース・データが縮小する場合です。例えば、ソースにある 4 バイトの EUC データがターゲットで 2 バイトの DBCS データに縮小すれば、必要スペースは半分になります。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。
forcecreate	インポート操作中に SQL3311N が戻された後、欠落している可能性のある、または限られた情報で表が作成されることを指定します。

表 41. codepage および usegraphiccodepage 使用時の IMPORT 動作

codepage=N	usegraphiccodepage	IMPORT 動作
なし	なし	ファイル内のすべてのデータは、アプリケーション・コード・ページであると見なされます。
あり	なし	ファイル内のすべてのデータは、コード・ページ N であると見なされます。  <b>警告:</b> N が 1 バイト・コード・ページの場合、GRAPHIC データをデータベースにインポートすると、壊れます。
なし	あり	ファイル内の文字データは、アプリケーション・コード・ページであると見なされます。 GRAPHIC データは、アプリケーション GRAPHIC データのコード・ページであると見なされます。  アプリケーション・コード・ページが 1 バイトの場合は、すべてのデータはアプリケーション・コード・ページであると見なされます。  <b>警告:</b> アプリケーション・コード・ページが 1 バイトの場合、 GRAPHIC データは、データベースにたとえ GRAPHIC 列が収められていても、データベースにインポートされると壊れます。
あり	あり	文字データは、コード・ページ N であると見なされます。 GRAPHIC データは、N の GRAPHIC コード・ページであると見なされます。  N が 1 バイトまたは 2 バイト・コード・ページの場合は、すべてのデータは、コード・ページ N であると見なされます。  <b>警告:</b> N が 1 バイト・コード・ページの場合、GRAPHIC データをデータベースにインポートすると、壊れます。

注:

1. サポートされていないファイル・タイプを **MODIFIED BY** オプションで使用しようとしても、インポート・ユーティリティは警告を出しません。この場合、インポート操作が失敗し、エラー・コードが戻されます。
2. 日付形式ストリングは必ず二重引用符で囲まなければなりません。フィールド区切り文字には、a から z、A から Z、および 0 から 9 を使用することはできません。フィールド区切り文字として、DEL ファイル・フォーマットの文字区切りまたはフィールド区切りと同じ文字を使用することはできません。エレメントの開始および終了位置が明らかな場合、フィールド区切り文字は任意指定です。あいまいさが生じうるのは、項目の長さが一定でない D、H、M、または S などのエレメントが使用されている場合です (修飾の仕方によって異なります)。

タイム・スタンプ・フォーマットの場合、月の記述子と分の記述子のどちらも文字 M を使用するため、区別があいまいにならないように注意する必要があります。月のフィールドは、他の日付フィールドと隣接していなければなりません。

せん。分フィールドは、他の時刻フィールドに隣接していなければなりません。あいまいなタイム・スタンプ形式の例を以下に示します。

```
"M" (could be a month, or a minute)
"M:M" (Which is which?)
"M:YYYY:M" (Both are interpreted as month.)
"S:M:YYYY" (adjacent to both a time value and a date value)
```

あいまいな場合、ユーティリティーはエラー・メッセージを報告し、操作は失敗します。

以下に、明確なタイム・スタンプ・フォーマットを示します。

```
"M:YYYY" (Month)
"S:M" (Minute)
"M:YYYY:S:M" (Month...Minute)
"M:H:YYYY:M:D" (Minute...Month)
```

二重引用符や円記号などの文字の前には、エスケープ文字 (例えば、¥) を付けるなければなりません。

3. `chardel`、`coldel`、または `decpt` ファイル・タイプ修飾子に提供される文字値は、ソース・データのコード・ページで指定する必要があります。

文字コード・ポイント (文字記号ではない) は、`xJJ` または `0xJJ` という構文で指定することができます (`JJ` はコード・ポイントの 16 進表記)。例えば、列区切りとして `#` 文字を指定するには、以下のステートメントのいずれかを使用します。

```
... modified by codel# ...
... modified by codel0x23 ...
... modified by codelX23 ...
```

4. データ移動のための区切り文字の制約事項に、区切り文字の指定変更として使用できる文字に適用される制限のリストが示されています。
5. 以下のファイル・タイプ修飾子は、ニックネームにインポートするときには使用できません。

- `indexixf`
- `indexschema`
- `dldel filetype`
- `nodefaults`
- `usedefaults`
- `no_type_id filetype`
- `generatedignore`
- `generatedmissing`
- `identityignore`
- `identitymissing`
- `lobsinfile`

6. **CREATE** モードは、XML 列ではサポートされません。

7. すべての XML データは、メイン・データ・ファイルとは別の XML ファイル内に存在する必要があります。XML Data Specifier (XDS) (または NULL 値) が、メイン・データ・ファイル内の XML 列ごとに存在する必要があります。

8. XMLCHAR または XMLGRAPHIC ファイル・タイプ修飾子が指定されていない場合は、XML 文書は Unicode 形式であるか、またはエンコード属性のある宣言タグを含むと想定されます。
9. 整形式でない文書が含まれている行はリジェクトされます。
10. **XMLVALIDATE** オプションが指定されている場合、対応するスキーマに対して正常に妥当性検査された文書は、挿入される際にスキーマ情報がアノテーションとして付加されます。対応するスキーマに対する妥当性検査が失敗した文書を含む行は、リジェクトされます。妥当性検査を正常に行うためには、インポートを起動するユーザーの保持する特権に、次の 1 つ以上が含まれている必要があります。
  - DBADM 権限
  - 妥当性検査に使用する XML スキーマに対する USAGE 特権
11. **ignore**、**include**、**missing**、**override** が接尾部に付された複数の修飾子が指定されると、それらの修飾子はリストされた順に適用されます。以下のステートメントの場合、入力データに ID 列ではない暗黙的な非表示列のデータが含まれます。一方、暗黙的に非表示かどうかに関係なく、ID 列のデータは含まれません。

```
db2 import from delfile1 of del modified by
    implicitlyhiddeninclude identitymissing insert into table1
```

しかし、以下のステートメントにおいてファイル・タイプ修飾子の順序を変更すると、入力データにすべての暗黙的な非表示列のデータ (非表示の ID 列も含む) が含まれることとなります。一方、暗黙的に非表示でない ID 列のデータは含まれません。

```
db2 import from delfile1 of del modified by
    identitymissing implicitlyhiddeninclude insert into table1
```

DB2\_DMU\_DEFAULT レジストリー変数が **IMPLICITLYHIDDENINCLUDE** に設定されている場合:

```
db2set DB2_DMU_DEFAULT=IMPLICITLYHIDDENINCLUDE
db2 import from delfile1 of del modified by identitymissing insert into table1
```

これは、以下と同じ意味になります。

```
db2 import from delfile1 of del modified by
    implicitlyhiddeninclude identitymissing insert into table1
```

#### **INITIALIZE TAPE コマンド (ADMIN\_CMD プロシージャを使用):**

ストリーミング磁気テープ装置へのバックアップおよびリストア操作のためにテープを初期化します。このコマンドは Windows オペレーティング・システムでのみサポートされています。

#### **許可**

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT



## 必要な接続

データベース

## コマンド構文

```
▶▶—INITIALIZE TAPE—┬──┬──▶  
                     └──ON—device──┘ └──USING—blksize──┘
```

## コマンド・パラメーター

### ON *device*

有効なテープ装置名を指定します。デフォルト値は ¥¥.¥TAPE0 です。装置の指定は、サーバーに対する相対指定でなければなりません。

### USING *blksize*

装置のブロック・サイズを指定します (バイト単位)。値が装置のブロック・サイズとしてサポートされている範囲内であれば、装置は指定されたそのブロック・サイズを使用するよう初期化されます。

**BACKUP DATABASE** コマンドおよび **RESTORE DATABASE** コマンドで指定されるバッファ・サイズは、ここで指定されるブロック・サイズで割り切れなければなりません。

このパラメーターに値を指定しなかった場合、装置はデフォルトのブロック・サイズを使用するよう初期化されます。値ゼロを指定した場合は、装置は可変長のブロック・サイズを使用するよう初期化されます。装置が可変長のブロック・モードをサポートしていない場合は、エラーが戻されます。

テープへのバックアップの場合、現在、可変ブロック・サイズの使用はサポートされていません。そのオプションを使用する必要がある場合は、リカバリーが正常に実行されるように十分にテストしたプロシージャが使用できるようになっていることを確認し、また可変ブロック・サイズを指定して作成されたバックアップ・イメージを使用してください。

可変ブロック・サイズを使用する場合、使用している磁気テープ装置の最大限度以下のバックアップ・バッファ・サイズを指定する必要があります。パフォーマンスを最適化するには、使用している装置のブロック・サイズの最大限度と等しい値をバッファ・サイズとして使用しなければなりません。

## 例

2048 バイトの値がその装置でサポートされているブロック・サイズの範囲内であれば、ブロック・サイズとして 2048 バイトを使用するよう、磁気テープ装置を初期化します。

```
CALL SYSPROC.ADMIN_CMD( 'initialize tape using 2048' )
```

## 使用上の注意

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

**LOAD コマンド (ADMIN\_CMD プロシージャを使用):**

データを DB2 表にロードします。

サーバー上に保管するデータは、ファイル、テープ、または名前付きパイプの形式にすることができます。また、現在接続しているデータベースまたは別のデータベースに対して実行する照会で定義したカーソルを使用してデータをロードすることもできます。あるいはユーザー作成のスクリプトまたはアプリケーションを使用して、データをロードすることもできます。表の COMPRESS 属性が YES に設定されている場合、ロードされるデータは、表内にディクショナリーが存在するデータおよびデータベース・パーティションごとに圧縮の対象となります。これには表の XML ストレージ・オブジェクト内のデータが含まれます。

139 ページの『ロード・ユーティリティー用のファイル・タイプ修飾子』へのクイック・リンク。

### 制約事項

ロード・ユーティリティーでは、階層レベルのデータのロードはサポートされていません。ロード・ユーティリティーには、範囲がクラスター化された表との互換性はありません。ロード・ユーティリティーでは、NOT LOGGED INITIALLY パラメーターを CREATE TABLE や ALTER TABLE ステートメントでサポートしていません。

### 有効範囲

このコマンドは、一度の要求で複数のデータベース・パーティションに対して発行できます。

### 許可

以下の権限のいずれか。

- DATAACCESS
- データベースに対する LOAD 権限と以下の特権
  - 表の INSERT 特権 (ロード・ユーティリティーが INSERT モード、TERMINATE モード、または RESTART モードで呼び出される場合 (TERMINATE モードは直前のロード挿入操作を終了するためのもので、RESTART モードは直前のロード挿入操作を再開するためのものです))
  - 表の INSERT および DELETE 特権 (ロード・ユーティリティーが REPLACE モード、TERMINATE モード、または RESTART モードで呼び出される場合 (TERMINATE モードは直前のロード置換操作を終了するためのもので、RESTART モードは直前のロード置換操作を再開するためのものです))
  - 例外表の INSERT 特権 (例外表をロード操作の一部として使用する場合)。
- 保護列のある表にデータをロードするには、その表内のすべての保護列への書き込みアクセスを許可する LBAC 資格情報を、セッション許可 ID が直接保持しているか、あるいはグループまたはロールを介して間接的に保持している必要があります。そうでない場合は、ロードが失敗してエラー (SQLSTATE 5U014) が戻されます。
- 保護された行を持つ表にデータをロードするには、セッション許可 ID が、以下の基準を満たすセキュリティ・ラベルを保持していなければなりません。

- セキュリティー・ラベルは表を保護しているセキュリティ・ポリシーの一部である
- セキュリティー・ラベルは、書き込みアクセスまたはすべてのアクセス用にセッション許可 ID に直接付与されているか、グループまたはロールを介して間接的に付与されている。

こうしたセキュリティ・ラベルをセッション許可 ID が保持していない場合は、ロードが失敗してエラー (SQLSTATE 5U014) が戻されます。セキュリティ・ラベルは、セッション許可 ID LBAC 信用証明情報が、データ内のロードされる行を保護するセキュリティ・ラベルにその許可 ID が書き込むことを許可しない場合に、その行を保護します。ただし、表を保護しているセキュリティ・ポリシーが CREATE SECURITY POLICY ステートメントの RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL オプションを使用して作成されている場合は、その状況にはなりません。その場合は、ロードが失敗してエラー (SQLSTATE 42519) が戻されます。

保護行がある表にデータをロードする場合、ターゲット表にはデータ・タイプ DB2SECURITYLABEL の列が 1 つあります。入力データ行にその列の値が含まれていない場合、その行はリジェクトされます。ただし、LOAD コマンドで `usedefaults` ファイル・タイプ修飾子が指定されている場合は例外です。その場合には、その表を保護するセキュリティ・ポリシーの書き込みアクセス用のセキュリティ・ラベルが使用されます。書き込みアクセス用のセキュリティ・ラベルがない場合、その行はリジェクトされ、次の行に処理が続行されます。

- REPLACE オプションを指定する場合、セッション許可 ID は表をドロップできる権限を持っていないとなりません。
- LOCK WITH FORCE オプションが指定される場合、SYSADM 権限が必要です。
- 行アクセス制御がアクティブになっている表に対して **LOAD REPLACE** を使用する場合は、その表をドロップするための権限が必要です。具体的には、表に対する CONTROL または DBADM のいずれかが必要になります。

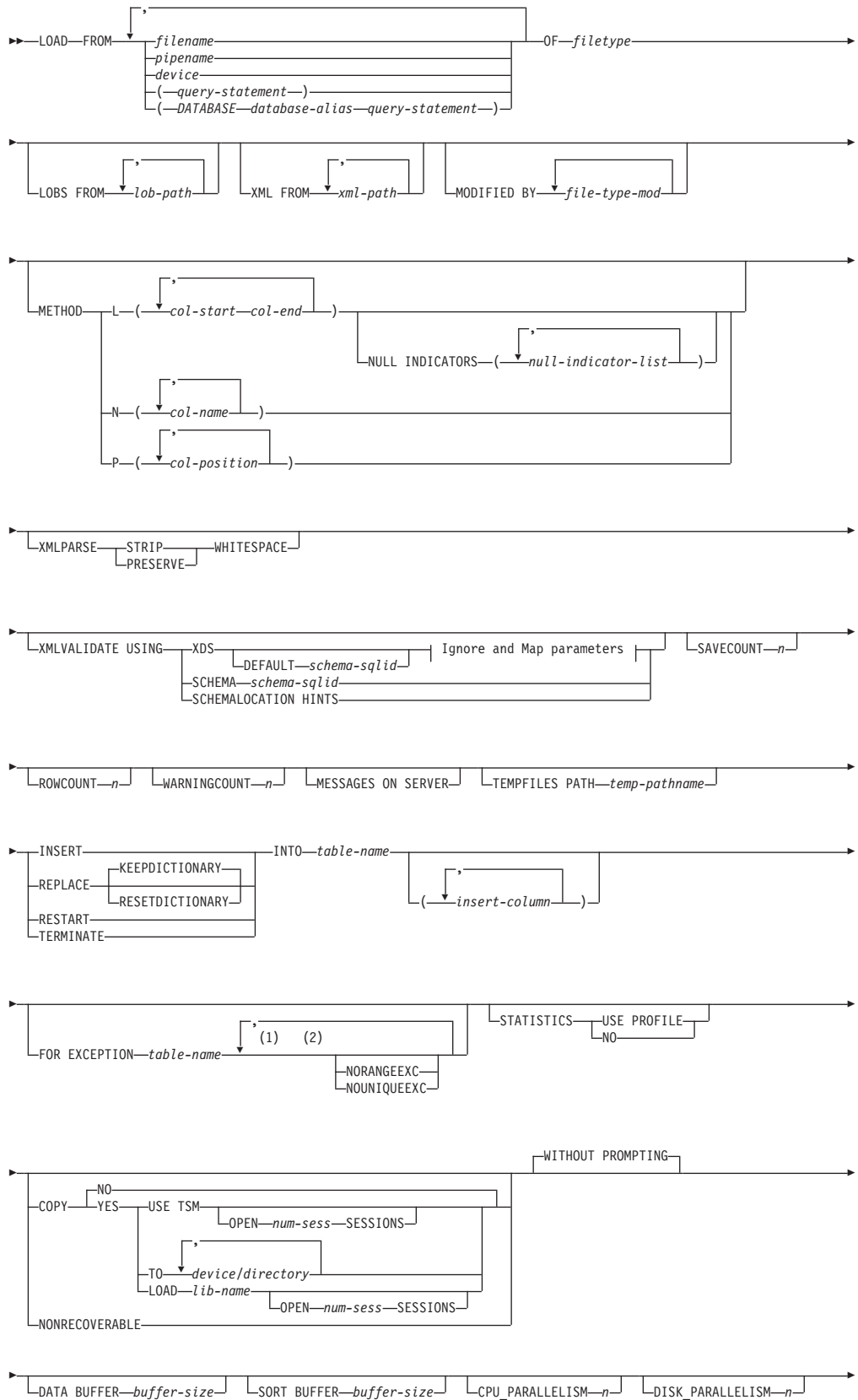
すべてのロード・プロセス (および一般にすべての DB2 サーバー・プロセス) はインスタンス所有者によって所有されており、それらのプロセスすべてにおいて、必要なファイルにアクセスするためにそのインスタンス所有者の ID を使用するため、インスタンス所有者には入力データ・ファイルに対する読み取りアクセス権が必要です。このコマンドをだれが呼び出すかには関係なく、それらの入力データ・ファイルをインスタンス所有者から読むことができなければなりません。

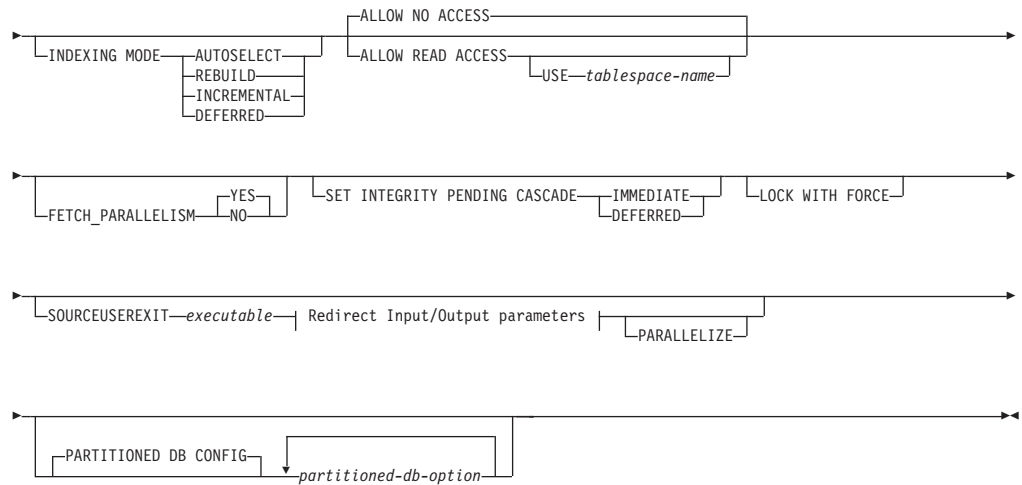
### 必要な接続

データベース。

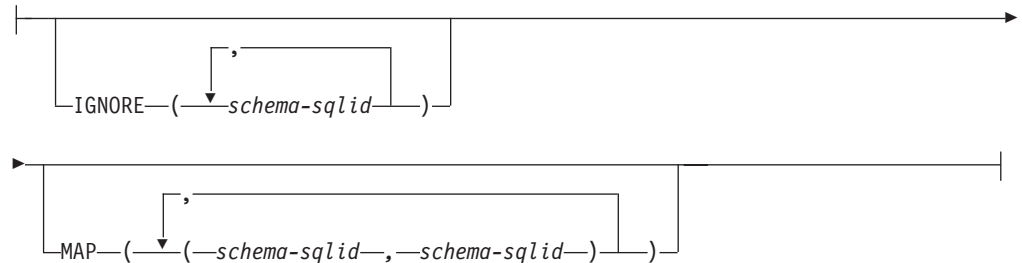
インスタンス。明示的なアタッチは必要ありません。データベースへの接続が確立されている場合には、ローカル・インスタンスへの暗黙的な接続が試みられます。

### コマンド構文

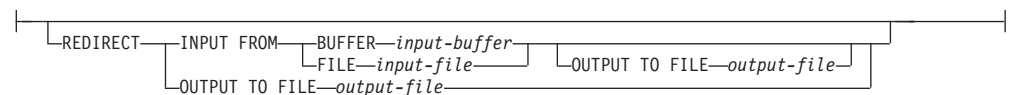




### Ignore and Map parameters:



### Redirect Input/Output parameters:



注:

- 1 これらのキーワードの出現順序は任意です。
- 2 これらのキーワードは、それぞれ 1 回だけ出現できます。

### コマンド・パラメーター

**FROM** *filename* | *pipename* | *device(query-statement)* | (**DATABASE** *database-alias* *query-statement*)

ロードされるデータを含んだ SQL ステートメントを参照するファイル、パイプ、または装置、あるいは、SQL ステートメントそのものと、カーソルからロードする任意指定のソース・データベースを指定します。

*query-statement* オプションは、カーソルから **LOAD** する場合に使用します。これは、括弧で囲んだ照会ステートメントを 1 つだけ含み、VALUES、SELECT、または WITH で始めることができます。例えば、次のようになります。

```
LOAD FROM (SELECT * FROM T1) OF CURSOR INSERT INTO T2
```

括弧で囲んだ照会ステートメントの前に **DATABASE** *database-alias* 節を組み込むと、**LOAD** コマンドは、*query-statement* を使用して、*database-alias* 名で指定された所定のデータベースからのデータのロードを試みます。**LOAD** は、現在接続されているデータベースの明示的に指定されたユーザー ID とパスワードを使用して実行されることに注意してください (暗黙接続は、**LOAD** の失敗を引き起こします)。

入力ソースがファイル、パイプ、または装置である場合、サーバー上のコーディネーター・パーティションからそれらにアクセスできなければなりません。

複数の名前を指定すると、それらは順番に処理されます。最後に指定した項目がテープ装置であり、ユーザーに対してテープを要求するプロンプトが出された場合、**LOAD** は失敗して **ADMIN\_CMD** プロシーチャーはエラーを戻します。

#### 注:

- 完全修飾パスのファイル名を使用し、それがサーバーに存在しなければなりません。
- **ADMIN\_CMD** プロシーチャーを使用した **EXPORT** コマンドを使用してデータをファイルにエクスポートした場合、そのデータ・ファイルは *fenced* ユーザー ID によって所有されます。このファイルは通常、インスタンス所有者がアクセスすることはできません。**LOAD** を **CLP** または **ADMIN\_CMD** プロシーチャーから実行するには、インスタンス所有者 ID はデータ・ファイルにアクセスできなければならないので、データ・ファイルに対する読み取りアクセス権限をインスタンス所有者に付与する必要があります。
- ファイルが物理的には分割されているが論理的には 1 つのファイルである場合には、複数の **IXF** ファイルからのデータのロードがサポートされています。ファイルが論理的にも物理的にも分割されている場合は、サポートされていません。論理的にも物理的にも分割されているファイルを複数指定した場合、最初のファイルより後のファイルはすべて無視されることに注意してください。(複数の物理ファイルがすべて一度の **EXPORT** コマンドの呼び出しで作成された場合、それらは論理的には 1 つであると見なされます。)
- **XML** データをファイルからパーティション・データベース環境にある表にロードする場合、**XML** データ・ファイルはロードが実行されるすべてのデータベース・パーティションから読み取りアクセス可能でなければなりません。

#### **OF** *filetype*

データのフォーマットを指定します。

- **ASC** (区切りなし **ASCII** フォーマット)
- **DEL** (区切り付き **ASCII** フォーマット)
- **IXF** (統合交換フォーマット、**PC** バージョン) は、**DB2** データベースによってのみ使用されるバイナリー・フォーマットです。
- **CURSOR** (**SELECT** または **VALUES** ステートメントに対して宣言されたカーソル)。

注:

- **CURSOR** ファイル・タイプを使用して XML データを分散データベース環境にある表にロードする場合、**PARTITION\_ONLY** および **LOAD\_ONLY** モードはサポートされません。
- **DECLARE CURSOR** ステートメントで **DATABASE** キーワードを指定した **CURSOR** ファイル・タイプを使用してロードを実行する場合、ロードは、データをフェッチするための別のアプリケーションを内部的に生成します。一方、**DATABASE** キーワードを指定しなかった場合、ロードは同じアプリケーション内でデータをフェッチします。この 2 つのケースの違いにより、ロック動作にも違いが生じます。特に、**DATABASE** キーワードを指定しており、現在接続しているデータベースと同じデータベースをそこで使用している場合 (また、ユーザー ID およびパスワードを指定している場合はそれらが現行接続と同じである場合)、ロックの問題 (データベース構成によって異なるが、ロック待機やロック・タイムアウトなど) が発生する可能性があります。これは、**DATABASE** キーワードを省略することで回避できます。

#### **LOBS FROM** *lob-path*

ロードする LOB 値が収められているデータ・ファイルへのパス。パスの最後はスラッシュでなければなりません。パスは完全修飾パスでなければならず、サーバー上のコーディネーター・パーティションからアクセスできなければなりません。LOB データ・ファイルの名前は、メイン・データ・ファイル (ASC、DEL、または IXF) の、LOB 列にロードされる列内に保管されます。指定できるパスの最大数は 999 です。これによって、**LOBSINFILE** 動作が暗黙的にアクティブ化されます。

**CURSOR** ファイル・タイプと併せて指定された場合、このオプションは無視されます。

#### **MODIFIED BY** *file-type-mod*

ファイル・タイプ修飾子オプションを指定します。 139 ページの『ロード・ユーティリティー用のファイル・タイプ修飾子』を参照してください。

#### **METHOD**

- L** データのロードを開始する列および終了する列の番号を指定します。列の番号は、データの行の先頭からのバイト単位のオフセットです。この番号は 1 から始まります。この方式は、ASC ファイルの場合にのみ使用することができ、そのファイル・タイプでは唯一の有効な方式です。

#### **NULL INDICATORS** *null-indicator-list*

このオプションは、**METHOD L** パラメーターを指定した場合だけ使用できます (つまり、入力ファイルが ASC ファイルの場合)。NULL 標識リストは、コンマで区切られた正の整数のリストで、各 NULL 標識フィールドの列の番号を指定します。列の番号は、データの行の先頭からのバイト単位の、各 NULL 標識フィールドのオフセットです。NULL 標識リストには、**METHOD L** パラメーターで定義された各データ・フィールドに対する 1 つの項目がなければなりません。

ん。列の番号がゼロであることは、対応するデータ・フィールド内に必ずデータがあることを示します。

NULL 標識列中の Y の値は、その列データが NULLであることを指定します。NULL 標識列に Y 以外の文字を指定した場合は、列データが NULL ではなく、**METHOD L** オプションで指定された列データがロードされることを指定することになります。

NULL 標識文字は、**MODIFIED BY** オプションを使用して変更できます。

**N** ロードするデータ・ファイルの中の列の名前を指定します。これらの列名の大文字小文字の区別は、システム・カタログ内の対応する名前の大文字小文字の区別と一致しなければなりません。NULL 可能ではない各表の列には、**METHOD N** リスト内に対応する項目が必要です。例えば、データ・フィールドが F1、F2、F3、F4、F5、および F6 であり、表の列が C1 INT、C2 INT NOT NULL、C3 INT NOT NULL、および C4 INT の場合、method N (F2, F1, F4, F3) は有効な要求ですが、method N (F2, F1) は無効です。この方式は、ファイル・タイプ IXF または CURSOR の場合にのみ使用することができます。

**P** ロードする入力データ・フィールドのフィールド番号 (1 から始まる) を指定します。NULL 可能ではない各表の列には、**METHOD P** リスト内に対応する項目が必要です。例えば、データ・フィールドが F1、F2、F3、F4、F5、および F6 であり、表の列が C1 INT、C2 INT NOT NULL、C3 INT NOT NULL、および C4 INT の場合、method P (2, 1, 4, 3) は有効な要求ですが、method P (2, 1) は無効です。この方式は、ファイル・タイプ IXF、DEL、または CURSOR の場合にのみ使用でき、DEL ファイル・タイプに対して有効な唯一の方式です。

method P によって指定したフィールドごとに、対応する列をアクション・ステートメント内に定義する必要があります。ただし、次の例に示すようにすべての列が対象になる場合、および最初の x 列がロードされる場合は必要ありません。

```
db2 import from datafile1.del of del method P(1, 3, 4)
      replace into table1 (c1, c3, c4)
```

#### **XML FROM** *xml-path*

XML ファイルが含まれているパスを 1 つ以上指定します。XDS は、メイン・データ・ファイル (ASC、DEL、または IXF) の、XML 列にロードされる列内に入れられます。

#### **XMLPARSE**

XML 文書の解析方法を指定します。このオプションが指定されていない場合、XML 文書の解析の動作は、CURRENT XMLPARSE OPTION 特殊レジスターの値によって決まります。

#### **STRIP WHITESPACE**

XML 文書の解析時に空白文字を除去することを指定します。



## PRESERVE WHITESPACE

XML 文書の解析時に空白文字を除去しないことを指定します。

## XMLVALIDATE

該当する場合に、XML 文書がスキーマに準拠しているかどうかの妥当性検査を実行することを指定します。

## USING XDS

メイン・データ・ファイルの中で XML Data Specifier (XDS) によって指定されている XML スキーマに準拠しているかどうかについて、XML 文書の妥当性検査が実行されます。デフォルトでは、**USING XDS** 節によって **XMLVALIDATE** オプションが呼び出された場合、妥当性検査実行のために使用されるスキーマは、その XDS の SCH 属性によって決まります。XDS の中で SCH 属性が指定されていない場合、**DEFAULT** 節によってデフォルト・スキーマが指定されているのでない限り、スキーマ妥当性検査は実行されません。

**DEFAULT**、**IGNORE**、および **MAP** 節を使用することにより、スキーマ決定の動作を変更することができます。これら 3 つの節はオプションであり、相互に適用されるのではなく XDS の指定に直接適用されます。例えば、**DEFAULT** 節で指定されているためにあるスキーマが選択された場合、それが **IGNORE** 節で指定されていたとしても無視されることはありません。同じように、**MAP** 節のペアの最初の部分で指定されているためにあるスキーマが選択された場合、それが別の **MAP** 節のペアの 2 番目の部分で指定されていたとしても再びマップされることはありません。

## USING SCHEMA *schema-sqlid*

指定されている SQL ID の XML スキーマに準拠しているかどうかについて、XML 文書の妥当性検査が実行されます。この場合、すべての XML 列について XML Data Specifier (XDS) の SCH 属性は無視されます。

## USING SCHEMALOCATION HINTS

ソース XML 文書の中で XML スキーマ・ロケーション・ヒントによって指定されているスキーマに準拠しているかどうかについて、XML 文書の妥当性検査が実行されます。その XML 文書の中に `schemaLocation` 属性が指定されていない場合、妥当性検査は実行されません。**USING SCHEMALOCATION HINTS** 節が指定されているなら、すべての XML 列について XML Data Specifier (XDS) の SCH 属性は無視されます。

以下のセクションに示す **XMLVALIDATE** オプションの例を参照してください。

## **IGNORE** *schema-sqlid*

このオプションは、**USING XDS** パラメーターを指定した場合にのみ使用できます。**IGNORE** 節は、SCH 属性によって指定されていても無視するスキーマとして、1 つ以上のスキーマのリストを指定します。ロードする XML 文書の XML Data Specifier の中に SCH 属性が存在し、その SCH 属性によ

って指定されるスキーマが無視するスキーマ・リストに含まれている場合には、ロードするその XML 文書についてスキーマ妥当性検査は実行されません。

注:

あるスキーマが **IGNORE** 節の中で指定されている場合、**MAP** 節のスキーマ・ペアの左辺にそれを含めることはできません。

**IGNORE** 節は XDS にのみ適用されます。あるスキーマが **IGNORE** 節によって指定されていても、それが **MAP** 節によってマップされているなら、それ以降そのスキーマが無視されることはありません。

#### **DEFAULT** *schema-sqlid*

このオプションは、**USING XDS** パラメーターを指定した場合にのみ使用できます。**DEFAULT** 節で指定されたスキーマは、ロード対象 XML 文書の XML Data Specifier (XDS) に XML スキーマを指定する **SCH** 属性が含まれていない場合に、妥当性検査のために使用するスキーマとなります。

**DEFAULT** 節は、**IGNORE** 節および **MAP** 節よりも優先されます。XDS が **DEFAULT** 節を満たすなら、**IGNORE** と **MAP** の指定は無視されます。

#### **MAP** *schema-sqlid*

このオプションは、**USING XDS** パラメーターを指定した場合にのみ使用できます。**MAP** 節は、ロードする各 XML 文書について XML Data Specifier (XDS) の **SCH** 属性によって指定されるスキーマの代わりに使用する代替スキーマを指定するのに使用します。**MAP** 節には、それぞれがあるスキーマから別のスキーマへのマッピングを表すスキーマ・ペアを 1 つ以上列挙したリストを指定します。ペア中の最初のスキーマは、XDS 内の **SCH** 属性によって参照されるスキーマを表します。ペア中の 2 番目のスキーマは、スキーマ妥当性検査の実行で使用する必要のあるスキーマを表します。

あるスキーマが **MAP** 節のスキーマ・ペアの左辺で指定されている場合、**IGNORE** 節でさらにそれを指定することはできません。

スキーマ・ペアのマッピングが適用されたなら、その結果は最終的なものです。マッピング操作は推移的ではないため、選択されたスキーマが、それ以降に別のスキーマ・ペアのマッピングに適用されることはありません。

スキーマを複数回マップすることはできません。つまり、複数のペアの左辺に指定することはできません。

#### **SAVECOUNT** *n*

ロード・ユーティリティが *n* 行ごとに整合点を取ることを指定します。この値はページ・カウントに変換され、エクステント・サイズのインターバルに切り上げられます。メッセージは整合点ごとに発行されるので、**LOAD QUERY** を使用してロード操作をモニターする場合には、このオプションを選択する必要があります。*n* の値が十分な大きさでない場合、各整合点で実行される活動の同期化によってパフォーマンスに影響してしまいます。

デフォルト値はゼロですが、それは、必要がなければ整合点は確立されないことを意味します。

**CURSOR** ファイル・タイプと併せて指定された場合、または XML 列を含む表をロードする場合、このオプションは使用できません。

**ROWCOUNT** *n*

ロードするファイル内の物理レコードの数 *n* を指定します。ユーザーはファイル内の最初の *n* 個の行だけをロードできます。

**WARNINGCOUNT** *n*

*n* 個の警告後に、ロード操作を停止します。このパラメーターは、警告は予期されないが、正しいファイルと表が使用されていることを確認するのが望ましい場合に設定してください。ロード・ファイルまたはターゲット表が不適切に指定されると、ロード対象の行ごとにロード・ユーティリティーによって警告が生成され、このためにロードが失敗する可能性があります。*n* がゼロの場合、またはこのオプションが指定されていない場合、何度警告が出されてもロード操作は続行します。警告のしきい値に達したためにロード操作が停止された場合でも、あらためて **RESTART** モードでロード操作を開始できます。ロード操作は、最後の整合点から自動的に続行します。または、入力ファイルの先頭から **REPLACE** モードであらためてロード操作を開始できます。

**MESSAGES ON SERVER**

**LOAD** コマンドによってサーバー上に作成されるメッセージ・ファイルを保存することを指定します。戻される結果セットには以下の 2 つの列が含まれます。1 つは **MSG\_RETRIEVAL** で、これはこの操作中に発生したすべての警告メッセージおよびエラー・メッセージを取り出すのに必要な SQL ステートメントです。もう 1 つは **MSG\_REMOVAL** で、これはメッセージをクリーンアップするために必要な SQL ステートメントです。

この節が指定されていない場合は、**ADMIN\_CMD** プロシージャから呼び出し元に戻る時点でメッセージ・ファイルが削除されます。結果セット中の **MSG\_RETRIEVAL** および **MSG\_REMOVAL** 列の内容は **NULL** 値になります。

この節を使用するしないにかかわらず、**fenced** ユーザー ID は、**DB2\_UTIL\_MSGPATH** レジストリー変数で示されるディレクトリ下にファイルを作成できる権限を持っていないければなりません。

**TEMPFILES PATH** *temp-pathname*

ロード操作時に一時ファイルを作成する場合に使用するパスの名前を指定します。これはサーバー・データベース・パーティションに従って完全に修飾しなければなりません。

一時ファイルは、ファイル・システムのスペースを使用します。場合によっては、このスペースが相当必要になります。以下のリストに示すのは、すべての一時ファイルにどの程度のファイル・システム・スペースを割り振るべきかの見積もりです。

- ロード・ユーティリティーが生成するメッセージごとに 136 バイト
- データ・ファイルに長フィールド・データまたは **LOB** が入っている場合は、15 KB のオーバーヘッド。 **INSERT** オプションを指定した場合で、表の中に多量の長フィールドまたは **LOB** データが既にある場合には、この数値はこれよりもかなり大きくなる場合があります。

**INSERT** ロード・ユーティリティーを実行できる 4 つのモードのうちの 1 つ。既存の表データを変更することなく、ロードされたデータを表に追加します。

## REPLACE

ロード・ユーティリティーを実行できる 4 つのモードのうちの 1 つ。表から既存データをすべて削除し、ロードされたデータを挿入します。表定義および索引定義は変更されません。階層間でデータを移動する際にこのオプションを使用する場合は、階層全体に関係したデータだけが置き換えられます。副表は置き換えられません。

このオプションは、システム期間テンポラル表へのデータのロードには使用できません。

## KEEPDICTIONARY

**LOAD REPLACE** 操作の後も、既存のコンプレッション・ディクショナリーを保持します。表の COMPRESS 属性が YES の場合、新規に置換されるデータは、ロードの呼び出し前に存在していたディクショナリーを使用して圧縮されるという影響を受けます。ディクショナリーが表の中に以前に存在していない場合、表の COMPRESS 属性が YES である限り、置換されて表に入れられるデータを使用して新規ディクショナリーが作成されます。この場合のコンプレッション・ディクショナリーを作成するために必要なデータ量は、ADC のポリシーの影響を受けます。このデータは圧縮解除された状態で表に取り込まれます。ディクショナリーが表にいったん挿入されると、ロード対象の残りのデータは、このディクショナリーを使用して圧縮されるという影響を受けます。これはデフォルト・パラメータです。要約を表 1 に示します。

次の例では、現在、古いディクショナリーが表の中にある場合にはそれを保持します。

```
CALL SYSPROC.ADMIN_CMD('load from staff.del of del replace  
keepdictionary into SAMPLE.STAFF statistics use profile  
data buffer 8')
```

表 42. LOAD REPLACE KEEPDICTIONARY

圧縮	表の行データ・ディクショナリーが存在する	XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	コンプレッション・ディクショナリー	データ圧縮
YES	YES	YES	表の行データ・ディクショナリーおよび XML ディクショナリーを保存します。	ロードされるデータは圧縮の対象になります。
YES	YES	NO	表の行データ・ディクショナリーを保存し、新規 XML ディクショナリーを作成します。	ロードされる表の行データは圧縮の対象になります。XML ディクショナリーが作成された後に、ロードされる残りの XML データは圧縮の対象になります。

表 42. LOAD REPLACE KEEPDICTIONARY (続き)

圧縮	表の行データ・ディクショナリーが存在する	XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	コンプレッション・ディクショナリー	データ圧縮
YES	NO	YES	表の行データ・ディクショナリーを作成し、XML ディクショナリーを保存します。	表の行データ・ディクショナリーが作成された後に、ロードされる残りの表の行データは圧縮の対象になります。ロードされる XML データは圧縮の対象になります。
YES	NO	NO	新規の表の行データおよび XML ディクショナリーを作成します。	ディクショナリーが作成された後に、ロードされる残りのデータは圧縮の対象になります。
NO	YES	YES	表の行データ・ディクショナリーおよび XML ディクショナリーを保存します。	ロードされるデータは、圧縮されません。
NO	YES	NO	表の行データ・ディクショナリーを保存します。	ロードされるデータは、圧縮されません。
NO	NO	YES	表の行ディクショナリーには影響はありません。XML ディクショナリーを保存します。	ロードされるデータは、圧縮されません。
NO	NO	NO	影響なし。	ロードされるデータは、圧縮されません。

注:

1. コンプレッション・ディクショナリーは、XML 列が DB2 パージョン 9.7 以降の表に追加された場合または表が Online Table Move を使用してマイグレーションされた場合にのみ、表の XML ストレージ・オブジェクトに対して作成できます。
2. **LOAD REPLACE KEEPDICTIONARY** 操作が中断された場合、**LOAD RESTART** または **LOAD TERMINATE** のどちらかを実行すると、ロード・ユーティリティをリカバリーすることができます。既存の XML ストレージ・オブジェクト・ディクショナリーは、中断された **LOAD REPLACE KEEPDICTIONARY** 操作をリカバリーした後は、保持されていない可能性があります。**LOAD RESTART** を使用する場合、新しい XML ストレージ・オブジェクト・ディクショナリーが作成されます。

**RESETDICTIONARY**

このディレクティブは、**LOAD REPLACE** 処理に、表の COMPRESS 属性が YES の場合には表データ・オブジェクトの新規ディクショナリーを作成するように指示します。COMPRESS 属性が NO で、ディクショナリーがすでに表にある場合、それは除去され、新規ディクショナリーは表に挿入されません。コンプレッション・デ

イクシヨナリーは 1 つのユーザー・レコードのみを使用して作成できます。ロードするデータ・セットのサイズがゼロの場合は、既存のイクシヨナリーが存在していても、そのイクシヨナリーは保持されません。このディレクティブを使用してイクシヨナリーを作成するために必要なデータ量は、ADC のポリシーの影響を受けません。要約については、表 2 を参照してください。

次の例は現在のイクシヨナリーをリセットし、新規のイクシヨナリーを作成します。

```
CALL SYSPROC.ADMIN_CMD('load from staff.del of del replace
resetdictionary into SAMPLE.STAFF statistics use profile
data buffer 8')
```

表 43. LOAD REPLACE RESETDICTIONARY

圧縮	表の行データ・イクシヨナリーが存在する	XML ストレージ・オブジェクト・イクシヨナリーが存在する <sup>1</sup>	コンプレッション・イクシヨナリー	データ圧縮
YES	YES	YES	新規イクシヨナリーの作成 <sup>2</sup> 。DATA CAPTURE CHANGES オプションが CREATE TABLE または ALTER TABLE ステートメントで有効になっていると、現在の表の行データ・イクシヨナリーが保持されます (これは履歴コンプレッション・イクシヨナリーと呼ばれます)。	イクシヨナリーが作成された後に、ロードされる残りのデータは圧縮の対象になります。
YES	YES	NO	新規イクシヨナリーの作成 <sup>2</sup> 。DATA CAPTURE CHANGES オプションが CREATE TABLE または ALTER TABLE ステートメントで有効になっていると、現在の表の行データ・イクシヨナリーが保持されます (これは履歴コンプレッション・イクシヨナリーと呼ばれます)。	イクシヨナリーが作成された後に、ロードされる残りのデータは圧縮の対象になります。
YES	NO	YES	新規イクシヨナリーの作成。	イクシヨナリーが作成された後に、ロードされる残りのデータは圧縮の対象になります。

表 43. LOAD REPLACE RESETDICTIONARY (続き)

圧縮	表の行データ・ディクショナリーが存在する	XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	コンプレッション・ディクショナリー	データ圧縮
YES	NO	NO	新規ディクショナリーの作成。	ディクショナリーが作成された後に、ロードされる残りのデータは圧縮の対象になります。
NO	YES	YES	ディクショナリーを除去します。	ロードされるデータは、圧縮されません。
NO	YES	NO	表の行データ・ディクショナリーを除去します。	ロードされるデータは、圧縮されません。
NO	NO	YES	XML ストレージ・オブジェクト・ディクショナリーを除去します。	ロードされるデータは、圧縮されません。
NO	NO	NO	影響なし。	すべての表データが圧縮されるわけではありません。

注:

1. コンプレッション・ディクショナリーは、XML 列が DB2 パージョン 9.7 以降の表に追加された場合または表が Online Table Move を使用してマイグレーションされた場合にのみ、表の XML ストレージ・オブジェクトに対して作成できます。
2. ディクショナリーが存在していて圧縮属性が有効であるものの、表パーティションにロードするレコードがない場合、新規ディクショナリーを作成することはできず、RESETDICTIONARY 操作では既存のディクショナリーは保持されません。

**TERMINATE**

ロード・ユーティリティを実行できる 4 つのモードのうちの 1 つ。以前に割り込みを受けたロード操作を終了し、ロード操作が開始された時点まで操作をロールバックします。途中で整合点があっても通過します。その操作に関係する表スペースの状態は通常に戻され、すべての表オブジェクトの整合性が保たれます (索引オブジェクトが無効とマークされる場合がありますが、そのような場合には、次のアクセス時に索引の再作成が自動的に行われます)。終了するロード操作が **LOAD REPLACE** の場合、その表は **LOAD TERMINATE** 操作完了後に空の表まで切り捨てられます。終了するロード操作が **LOAD INSERT** の場合、その表は **LOAD TERMINATE** 操作完了後も元のレコードをすべて保持します。ディクショナリー管理の要約を表 3 に示します。

**LOAD TERMINATE** オプションでは、表スペースの BACKUP PENDING 状態は解除されません。

**RESTART**

ロード・ユーティリティを実行できる 4 つのモードのうちの 1 つ。以前に割り込みを受けたロード操作を再開します。ロード操作は、ロード、作

成、または削除フェーズの最後の整合点から自動的に続行されます。ディクショナリー管理の要約を表 4 に示します。

#### **INTO** *table-name*

データのロード先となるデータベース表を指定します。この表として、システム表、宣言済み一時表、または作成済み一時表は指定できません。別名、完全修飾、または非修飾の表名を指定できます。修飾された表名は、*schema.tablename* の形式になります。非修飾の表名を指定すると、その表は CURRENT SCHEMA で修飾されます。

暗黙的な非表示列がデータベース表に含まれる場合には、ロード操作に非表示列のデータを含めるかどうかを指定する必要があります。非表示列のデータを含めるかどうかを示すには、以下のいずれかの方法を使用します。

- データの挿入先の列を明示的に指定する場合、*insert-column* を使用します。

```
db2 load from delfile1 of del
      insert into table1 (c1, c2, c3,...)
```

- 非表示列のファイル・タイプ修飾子のいずれかを使用します。入力ファイルに非表示列のデータが含まれる場合には **implicitlyhiddeninclude** を指定し、入力ファイルに含まれない場合には **implicitlyhiddenmissing** を指定します。

```
db2 load from delfile1 of del modified by implicitlyhiddeninclude
      insert into table1
```

- 暗黙的な非表示列が含まれる表をデータ移動ユーティリティーが検出した場合のデフォルト動作を設定するには、DB2\_DMU\_DEFAULT レジストリー変数をサーバー・サイドで使用します。暗黙的な非表示列が含まれるとユーティリティーが想定するようにするには **IMPLICITLYHIDDENINCLUDE** を、暗黙的な非表示列は含まれないとユーティリティーが想定するようにするには **IMPLICITLYHIDDENMISSING** を指定します。

```
db2set DB2_DMU_DEFAULT=IMPLICITLYHIDDENINCLUDE
db2 load from delfile1 of del insert into table1
```

#### *insert-column*

データの挿入先となる表の列を指定します。

ロード・ユーティリティーは、1 つ以上のスペースを使った名前前の列を解析できません。例えば、次のようになります。

```
CALL SYSPROC.ADMIN_CMD('load from delfile1 of del noheader
      method P (1, 2, 3, 4, 5, 6, 7, 8, 9)
      insert into table1 (BLOB1, S2, I3, Int 4, I5, I6, DT7, I8, TM9)')
```

は、Int 4 列があるためエラーになります。これは、次のようにして二重引用符で列名を囲むことによって解決できます。

```
CALL SYSPROC.ADMIN_CMD('load from delfile1 of del noheader
      method P (1, 2, 3, 4, 5, 6, 7, 8, 9)
      insert into table1 (BLOB1, S2, I3, "Int 4", I5, I6, DT7, I8, TM9)')
```

#### **FOR EXCEPTION** *table-name*

エラーが発生した行のコピー先となる例外表を指定します。ユニーク索引または主キー索引に違反した行がすべてコピーされます。非修飾の表名を指定すると、その表は CURRENT SCHEMA で修飾されます。



例外表に書き込まれる情報は、ダンプ・ファイルには書き込まれません。パーティション・データベース環境では、ロードする表を定義されたデータベース・パーティションの例外表を定義する必要があります。一方、ダンプ・ファイルには、無効であるか構文エラーであるためにロードできない行が入ります。

XML データのロード時に、**FOR EXCEPTION** 節を使用したロード例外表の指定は、以下の状況ではサポートされません。

- ラベル・ベースのアクセス制御 (LBAC) を使用する場合。
- パーティション表にデータをロードする場合。

#### **NORANGEEXC**

範囲違反のためにリジェクトされた行は、例外表に挿入しないことを指定します。

#### **NOUNIQUEEXC**

ユニーク制約に違反しているためにリジェクトされた行は、例外表に挿入しないことを指定します。

#### **STATISTICS USE PROFILE**

この表で定義されているプロファイルに従ってロード中に統計を収集するようロード操作に指示します。そのプロファイルは、ロードの実行前に作成されていなければなりません。そのプロファイルは、**RUNSTATS** コマンドで作成します。プロファイルが存在しない場合に、プロファイルに従って統計を収集するようロード操作に指示すると、警告メッセージが戻されて統計は収集されません。

ロードの際に、分散統計はタイプ XML の列については収集されません。

#### **STATISTICS NO**

統計データを収集せず、したがってカタログ内の統計データも変更しないことを指定します。これはデフォルトです。

#### **COPY NO**

順方向リカバリーが有効である場合 (つまり、**logarchmeth1** または **logarchmeth2** に OFF 以外の値が設定されている場合)、表が存在している表スペースをバックアップ・ペンディング状態にするよう指定します。**COPY NO** オプションを使用する場合も、表スペース状態は **LOAD IN PROGRESS** になります。これは、一時的な状態であり、ロードが完了するか打ち切られると解除されます。表スペースのバックアップまたはデータベースのフルバックアップを実行しない限り、表スペースのどの表のデータも更新または削除できません。ただし、**SELECT** ステートメントを使用すれば、どの表のデータにもアクセス可能です。

リカバリー可能データベースでの **COPY NO** を指定した **LOAD** は、表スペースを **BACKUP PENDING** 状態のままにします。例えば、**COPY NO** を指定した **LOAD** および **INDEXING MODE DEFERRED** を実行すると、索引はリフレッシュが必要な状態になります。表での照会には、索引スキャンが必要なものがあり、索引がリフレッシュされるまで、成功しません。索引が、バックアップ・ペンディング状態にある表スペース内にある場合、その索引のリフレッシュはできません。この場合、表へのアクセスは、バックアップが行われるまで許可されません。索引リフレッシュは、索引が照会によってアクセスされたときに、データベースによって自動的に行われます。**COPY NO**、**COPY**

**YES、NONRECOVERABLE** のいずれも指定されておらず、データベースがリカバリー可能である場合 (**logarchmeth1** または **logarchmeth2** に OFF 以外の値が設定されている場合)、**COPY NO** がデフォルトです。

#### **COPY YES**

ロードするデータのコピーを保存することを指定します。フォワード・リカバリーが使用不可になっている場合、このオプションは無効です。

#### **USE TSM**

Tivoli Storage Manager (TSM) を使ってコピーを保管することを指定します。

#### **OPEN num-sess SESSIONS**

TSM またはベンダー製品とともに使用する入出力セッションの数です。デフォルト値は 1 です。

#### **TO device/directory**

コピー・イメージを作成する先の装置またはディレクトリーを指定します。

#### **LOAD lib-name**

使用するバックアップおよびリストア I/O ベンダー関数を含む共有ライブラリー (Windows オペレーティング・システムでは DLL) の名前。絶対パスで指定することができます。絶対パスを指定しない場合、デフォルトでユーザー出口プログラムの存在するパスになります。

#### **NONRECOVERABLE**

ロード・トランザクションがリカバリー不能としてマークされており、それ以降のロールフォワード・アクションによってそれをリカバリーさせることは不可能であることを指定します。ロールフォワード・ユーティリティーは、そのトランザクションをスキップし、データのロード先の表に「無効」としてマークします。さらに、ユーティリティーは、その表に対する後続のすべてのトランザクションを無視します。ロールフォワード操作が完了すると、そのような表は、ドロップするか、またはリカバリー不能なロード操作完了後のコミット・ポイントの後に取られたバックアップ (フルバックアップまたは表スペースのバックアップ) からリストアすることのみ可能です。

このオプションを使用すると、表スペースはロード操作後に **BACKUP PENDING** 状態になりません。また、ロード操作中にロードされたデータのコピーが作成される必要もなくなります。 **COPY NO、COPY YES、**

**NONRECOVERABLE** のいずれも指定されておらず、データベースがリカバリー可能ではない場合 (**logarchmeth1** および **logarchmeth2** の両方に OFF が設定されている場合)、**NONRECOVERABLE** がデフォルトです。

#### **WITHOUT PROMPTING**

データ・ファイルのリストにロードするすべてのファイルを含め、しかもリストに入っている装置またはディレクトリーがロード操作全体で十分であるということを指定します。続きの入力ファイルが見つからなかったり、ロード操作が終了する前にコピー先がいっぱいになるとロード操作は失敗し、表は **LOAD PENDING** 状態のままになります。

これはデフォルトです。通常はユーザー介入を必要とするアクションでは、エラー・メッセージが戻されます。

#### DATA BUFFER *buffer-size*

ユーティリティ内でデータを転送するためのバッファ・スペースとして使用する 4 KB ページの数を指定します (並列処理の度合いには依存しません)。指定する値がアルゴリズム上の最小値より小さい場合、最小限必要なりソースが使用され、警告は戻されません。

このメモリーは、ユーティリティ・ヒープから直接に割り当てられ、そのサイズは **util\_heap\_sz** データベース構成パラメーターで修正可能です。バージョン 9.5 以降では、システムにさらに使用可能なメモリーがある場合、**LOAD** コマンドの **DATA BUFFER** オプションの値は、一時的に **util\_heap\_sz** を超えることができます。この場合、ユーティリティ・ヒープは、必要に応じて **database\_memory** 限度に達するまで動的に増加します。このメモリーは、ロード操作が完了すると解放されます。

値が指定されていない場合、実行時にユーティリティによって適切なデフォルトが計算されます。デフォルトは、ローダーのインスタンス生成時にユーティリティ・ヒープで使用できるフリー・スペースの割合と、表の一部の特性に基づいて決まります。

#### SORT BUFFER *buffer-size*

このオプションは、ロード操作時に **sortheap** データベース構成パラメーターをオーバーライドする値を指定します。これは、索引とともに表をロードする場合、また **INDEXING MODE** パラメーターが **DEFERRED** として指定されていない場合にのみ関係があります。指定された値は **sortheap** の値を超えることはできません。このパラメーターは、**sortheap** の値を変更せずに多くの索引を持つ表をロードする際に使用される、ソート・メモリーのスロットルで役に立ちます。これは、一般的な照会処理にも影響を与えます。

#### CPU\_PARALLELISM *n*

表オブジェクトの作成時に、レコードの解析、変換、およびフォーマット設定のためにロード・ユーティリティによって作成されるプロセスまたはスレッドの数を指定します。このパラメーターは、データベース・パーティションごとに実行するプロセスの数を活用するために設計されています。これは、事前にソートされたデータをロードする際に役立ちます (ソース・データのレコード順序が保持されるため)。このパラメーターの値が 0 の場合や、このパラメーターを指定しなかった場合、ロード・ユーティリティは、実行時に自動的に計算された適切なデフォルト値 (通常は使用できる CPU の数に基づく) を使用します。

#### 注:

1. **LOB** または **LONG VARCHAR** フィールドのどちらかの入った表でこのパラメーターを使用する場合、システムの CPU の数またはユーザーが指定した値には関係なく、値は 1 になります。
2. **SAVECOUNT** パラメーターに指定する値が小さいと、データと表のメタデータの両方をフラッシュするために、ローダーがさらに多くの入出力操作を実行することになります。 **CPU\_PARALLELISM** が 1 より大きいなら、フラッシュ操作は非同期になり、ローダーは CPU を活用できます。 **CPU\_PARALLELISM** が 1 に設定されている場合、ローダーは整合点において入出力を待ちます。 **CPU\_PARALLELISM** を 2 に設定し、

**SAVECOUNT** を 10 000 に設定したロード操作は、CPU が 1 つしかなくても、同じ操作で **CPU\_PARALLELISM** を 1 に設定した場合より速く完了します。

#### **DISK\_PARALLELISM *n***

表スペース・コンテナにデータを書き込むためにロード・ユーティリティーが作成するプロセスまたはスレッドの数を指定します。値を指定しない場合、ユーティリティーは表スペース・コンテナの数と表の特性に基づいて、自動的に計算された適切なデフォルトを選択します。

#### **INDEXING MODE**

ロード・ユーティリティーが索引を再作成するのか、それとも索引を増分で拡張するのかを指定します。有効な値は以下のとおりです。

##### **AUTOSELECT**

**REBUILD** モードと **INCREMENTAL** モードのいずれにするかを、ロード・ユーティリティーが自動的に決定します。決定は、ロードされるデータ量と索引ツリーの深さに基づいて行われます。索引ツリーの深さに関連する情報は索引オブジェクトに保管されています。この情報を設定するために、**RUNSTATS** は不要です。**AUTOSELECT** がデフォルトの索引付けモードです。

##### **REBUILD**

すべての索引が再作成されます。古い表データの索引キー部分も、追加される新しい表データの索引キー部分もすべてソートできるようにするため、ロード・ユーティリティーには十分なりソースが必要となります。

**LogIndexBuild** データベース構成パラメーターがオンになっている場合、トランザクション・ログには、各索引ページの作成後にそのページのイメージが含まれます。**LogIndexBuild** データベース構成パラメーターがオフになっている場合、索引マネージャーがログに記録するのは、各ページの割り振りと初期化のみです (各ページの空ではない部分とは異なり、ページ当たり約 250 バイト)。

##### **INCREMENTAL**

索引に新しいデータが取り込まれて拡張します。このアプローチでは、索引のフリー・スペースが消費されます。このアプローチでは、新たに挿入されるレコードの索引キーを追加するためのソート・スペースだけがあれば十分です。この方式がサポートされるのは、索引オブジェクトが有効で、かつロード操作の開始時にアクセス可能な場合だけです (例えば、**DEFERRED** モードが指定されたロード操作の直後では、この方式は無効です)。このモードを指定したものの、索引の状態などの理由でサポートされない場合は、警告が戻され、**REBUILD** モードでロード操作が続行されます。同様に、ロード作成フェーズでロード再開操作を開始した場合も、**INCREMENTAL** モードはサポートされません。

**LogIndexBuild** データベース構成パラメーターがオンになっている場合、DB2 ソフトウェアは、実行されたページ分割のほか、索引への各キーの挿入に関するログ・レコードを生成します。このパラメーターがオフになっている場合 (**HADR** を使用していない場合一般的)、索引マネージャーによって実行される索引ロギングの量は、

ALLOW READ ACCESS オプションが指定されたかどうかに応じて異なります。 ALLOW READ ACCESS オプションが指定されている場合、ページ分割に関するログを含んだログ・レコードが生成されます。 ALLOW READ ACCESS オプションが指定されていない場合、索引マネージャーからのログ・レコードは生成されません。

#### DEFERRED

このモードが指定されている場合、ロード・ユーティリティーは索引の作成を試みません。リフレッシュが必要であることを示すマークが索引に付けられます。ロード操作とは関係のないこのような索引に最初にアクセスするときは、再作成が強制的に実行されたり、データベースの再始動時に索引が再作成されたりする場合があります。このアプローチでは、最も大きい索引のキー部分をすべて処理できるだけのソート・スペースが必要です。索引を作成するためにその後かかる合計時間は、REBUILD モードの場合よりも長くなります。したがって、この索引作成据え置きモードで複数のロード操作を実行する場合、最初の非ロード・アクセス時に索引を再作成できるようにしておくよりも、順序列内の最後のロード操作で索引の再作成を実行できるようにした方が (パフォーマンスの観点から) 賢明であるといえます。

据え置き索引作成がサポートされるのは、非ユニーク索引がある表だけです。そのため、ロード・フェーズで挿入される重複キーがロード操作後は永続的ではなくなります。

#### ALLOW NO ACCESS

ロードを使用すると、ロード中に、排他的アクセスのためにターゲット表がロックされます。ロード中、表の状態は LOAD IN PROGRESS に設定されます。 ALLOW NO ACCESS はデフォルトの動作です。これは、LOAD REPLACE で唯一有効なオプションです。

表に制約があると、表の状態は、ロード中の他に、SET INTEGRITY ペンディングに設定されます。表の SET INTEGRITY ペンディング状態を解除するには、SET INTEGRITY ステートメントを使用する必要があります。

#### ALLOW READ ACCESS

ロードを使用すると、ターゲット表は共有モードでロックされます。表の状態は、LOAD IN PROGRESS および READ ACCESS の両方に設定されます。表のロード中、データの非デルタ部分にアクセスすることができます。つまり、表を読み取る側はロードの開始前に存在していたデータにはアクセスができ、ロード中のデータはロードが完了するまで利用できない、ということです。

**重要:** バージョン 10.1 フィックスパック 1 以降、ALLOW READ ACCESS パラメーターは非推奨となっており、将来のリリースで除去される可能性があります。詳しくは、「DB2 バージョン 10.1 の新機能」の『LOAD コマンドの ALLOW READ ACCESS パラメーターが非推奨になった』を参照してください。

**ALLOW READ ACCESS** ロードの **LOAD TERMINATE** または **LOAD RESTART** はこのパラメーターを使用できますが、**ALLOW NO ACCESS** ロードの **LOAD**

**TERMINATE** または **LOAD RESTART** はこのパラメーターを使用できません。また、ターゲット表上の索引が要再作成のマークが付けられると、このオプションは無効になります。

表に制約があると、表の状態は、**LOAD IN PROGRESS**、および **READ ACCESS** の他に、**SET INTEGRITY PENDING** に設定されます。ロードの終了時に、表の状態 **LOAD IN PROGRESS** は解除されますが、**SET INTEGRITY PENDING** および **READ ACCESS** はそのまま残ります。表の **SET INTEGRITY PENDING** を解除するには、**SET INTEGRITY** ステートメントを使用する必要があります。表が **SET INTEGRITY PENDING** および **READ ACCESS** の状態にある間、データの非デルタ部分には引き続き読み取りアクセスできますが、データの新しい (デルタ) 部分には、**SET INTEGRITY** ステートメントが完了するまでアクセス不能のままになります。ユーザーは、**SET INTEGRITY** ステートメントを発行しないで、同じ表上で複数のロードを実行できます。ただし、元の (チェック済み) データは、**SET INTEGRITY** ステートメントが発行されるまで可視のままです。

**ALLOW READ ACCESS** は、以下の修飾子もサポートします。

#### **USE** *tablespace-name*

索引が再作成される場合、表スペース *tablespace-name* に索引のシャドー・コピーが作成され、ロード終了時の **INDEX COPY PHASE** で、元の表スペース上にコピーされます。**SYSTEM TEMPORARY** 表スペースのみ、このオプションを使用できます。指定されない場合、シャドー索引が、索引オブジェクトと同じ表スペース内に作成されます。シャドー・コピーが索引オブジェクトと同じ表スペース内に作成される場合、古い索引オブジェクトを介したシャドー索引オブジェクトのコピーは瞬時に終了します。シャドー・コピーが索引オブジェクトとは異なる表スペースにある場合、物理コピーが実行されます。これにはかなりの入出力および時間を要します。コピーは、表がオフラインの間、ロード終了時の **INDEX COPY PHASE** で行われます。

このオプションを使用しないと、シャドー索引は元の索引と同じ表スペースに作成されます。デフォルトでは、元の索引とシャドー索引の両方が同時に同じ表スペースに常駐するため、1つの表スペース内に両方の索引を保持するためのスペースが不足する場合があります。このオプションを使用すれば、索引用の十分な表スペースを保持できます。

ユーザーが **INDEXING MODE REBUILD** または **INDEXING MODE AUTOSELECT** を指定しない場合、このオプションは無視されます。このオプションは **INDEXING MODE AUTOSELECT** が選択され、ロードが索引を徐々に更新することを選択した場合にも無視されます。

#### **FETCH\_PARALLELISM YES | NO**

**DATABASE** キーワードを使用してカーソルが宣言されていてカーソルからのロードを実行するとき、または API の `sqlu_remotefetch_entry` メディア項目を使用するとき、このオプションが **YES** に設定されていると、ロード・ユーティリティは、リモート・データ・ソースからのフェッチの並列化を試みます (可能な場合)。 **NO** に設定されている場合、並列フェッチは行われま

せん。デフォルト値は、YES です。詳しくは、『CURSOR ファイル・タイプを使用したデータの移動』を参照してください。

#### SET INTEGRITY PENDING CASCADE

**LOAD** によって表が SET INTEGRITY ペンディング状態になる場合、**SET INTEGRITY PENDING CASCADE** オプションを使用することによって、ユーザーはロードされる表の SET INTEGRITY ペンディング状態を即時にすべての下層 (下層外部キー表、下層即時マテリアライズ照会表、および下層即時ステージング表を含む) にカスケードするかどうか指定することができます。

#### IMMEDIATE

外部キー制約の SET INTEGRITY PENDING 状態が即時にすべての下層外部キー表、下層即時マテリアライズ照会表、および下層ステージング表に拡張されることを示します。**LOAD INSERT** 操作の場合、**IMMEDIATE** オプションが指定されている場合でも、SET INTEGRITY PENDING 状態は下層外部キー表に拡張されません。

後で (SET INTEGRITY ステートメントの IMMEDIATE CHECKED オプションを使用して) ロードされる表の制約違反をチェックする際、SET INTEGRITY PENDING READ ACCESS 状態だった下層外部キー表は、SET INTEGRITY PENDING NO ACCESS 状態になります。

#### DEFERRED

ロードされる表だけが、SET INTEGRITY PENDING 状態になることを示します。下層外部キー表、下層即時マテリアライズ照会表、および下層即時ステージング表は、未変更のままになります。

下層外部キー表は、(SET INTEGRITY ステートメントの IMMEDIATE CHECKED オプションを使用して) その親表の制約違反がチェックされるとき、後で暗黙的に SET INTEGRITY PENDING 状態になる場合があります。下層即時マテリアライズ照会表および下層即時ステージング表は、その基礎表のいずれかの保全性違反がチェックされる際、暗黙的に SET INTEGRITY PENDING 状態になります。照会のアクセス先が、指定された表ではなく、SET INTEGRITY PENDING 状態にない適格なマテリアライズ照会表である場合は、SET INTEGRITY PENDING 状態の表の照会が成功することがあります。下層表が SET INTEGRITY ペンディング状態になったことを示す警告 (SQLSTATE 01586) が出されます。この下層表がいつ SET INTEGRITY PENDING 状態になるかについては、SQL リファレンスにある SET INTEGRITY ステートメントの「注」の項を参照してください。

**SET INTEGRITY PENDING CASCADE** オプションが指定されない場合、次のようになります。

- ロードされる表だけが、SET INTEGRITY PENDING 状態になります。下層外部キー表、下層即時マテリアライズ照会表、および下層即時ステージング表の状態は未変更のままになり、後にロードされた表の制約違反がチェックされる際に、暗黙的に SET INTEGRITY PENDING 状態になる場合があります。

**LOAD** によってターゲット表が **SET INTEGRITY PENDING** 状態にならない場合、**SET INTEGRITY PENDING CASCADE** オプションは無視されます。

#### **LOCK WITH FORCE**

ユーティリティーはロード・プロセス中に、表ロックなどの様々なロックを獲得します。ロックを獲得する際、このオプションを使用すると、ロードは待機することなく、またタイムアウトになることなく、ターゲット表に競合するロックを持つ他のアプリケーションを強制的にオフにします。システム・カタログ表に対する競合するロックを保持するアプリケーションは、ロード・ユーティリティーによって強制的にオフにされることはありません。強制されたアプリケーションは、ロールバックし、ロード・ユーティリティーが必要とするロックをリリースします。その後、ロード・ユーティリティーを続行できます。このオプションは、**FORCE APPLICATIONS** コマンドと同じ権限 (**SYSADM** または **SYSCTRL**) を必要とします。

**ALLOW NO ACCESS** ロードは、ロード操作の開始時に競合するロックを持つアプリケーションを強制的にロールバックさせる場合があります。ロードの開始時に、ユーティリティーは、表の照会または変更を試みているアプリケーションを強制的にロールバックさせる場合があります。

**ALLOW READ ACCESS** ロードは、ロード操作の開始時および終了時に競合するロックを持つアプリケーションを強制的にロールバックさせる場合があります。ロードの開始時に、ロード・ユーティリティーは、表の変更を試みているアプリケーションを強制的にロールバックさせる場合があります。ロード操作の終了時に、ロード・ユーティリティーは、表の照会または変更を試みているアプリケーションを強制的にロールバックさせる場合があります。

#### **SOURCEUSEREXIT** *executable*

このユーティリティーにデータを送るために呼び出される実行可能ファイル名を指定します。

#### **REDIRECT**

##### **INPUT FROM**

###### **BUFFER** *input-buffer*

*input-buffer* で指定されたバイトのストリームが、所定の実行可能ファイルを実行するプロセスの **STDIN** ファイル記述子に渡されます。

###### **FILE** *input-file*

このクライアント・サイドのファイルの内容が、所定の実行可能ファイルを実行するプロセスの **STDIN** ファイル記述子に渡されます。

##### **OUTPUT TO**

###### **FILE** *output-file*

**STDOUT** および **STDERR** ファイル記述子が、指定した完全修飾されたサーバー・サイドのファイルに取り込まれます。

#### **PARALLELIZE**

複数のユーザー出力プロセスを同時に呼び出すことによって、ロード・ユーティリティーへのデータ入力のスループットを高めめます。



このオプションは、複数パーティション・データベース環境でのみ適用でき、単一パーティション・データベース環境では無視されます。

詳細については、『カスタマイズしたアプリケーション (ユーザー出口) を使用したデータの移動』を参照してください。

#### **PARTITIONED DB CONFIG** *partitioned-db-option*

複数のデータベース・パーティションに分散した表へのロードの実行を可能にします。 **PARTITIONED DB CONFIG** パラメーターを使用すると、パーティション・データベース固有の構成オプションを指定することができます。

*partitioned-db-option* の値は、以下のオプションのいずれかになります。

```
PART_FILE_LOCATION x
OUTPUT_DBPARTNUMS x
PARTITIONING_DBPARTNUMS x
MODE x
MAX_NUM_PART_AGENTS x
ISOLATE_PART_ERRS x
STATUS_INTERVAL x
PORT_RANGE x
CHECK_TRUNCATION
MAP_FILE_INPUT x
MAP_FILE_OUTPUT x
TRACE x
NEWLINE
DISTFILE x
OMIT_HEADER
RUN_STAT_DBPARTNUM x
```

これらのオプションの詳細な説明は、『パーティション・データベース環境のロード構成オプション』で扱われています。

#### **RESTARTCOUNT**

非推奨。

#### **USING** *directory*

非推奨。

#### **例**

ファイルにある従業員表のデータに対して置換オプションを指定してロードを発行します。

```
CALL SYSPROC.ADMIN_CMD('LOAD FROM /home/theresax/tmp/emp_exp.dat
  OF DEL METHOD P (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14)
  MESSAGES /home/theresax/tmp/emp_load.msg
  REPLACE INTO THERESAX.EMPLOYEE (EMPNO, FIRSTNME, MIDINIT, LASTNAME,
  WORKDEPT, PHONENO, HIREDATE, JOB, EDLEVEL, SEX, BIRTHDATE, SALARY,
  BONUS, COMM) COPY NO INDEXING MODE AUTOSELECT ISOLATE_PART_ERRS
  LOAD_ERRS_ONLY MODE PARTITION_AND_LOAD' )
```

以下のセクションは、単一パーティション・データベースからの出力例です。

Result set 1

-----

```
ROWS_READ      ROWS_SKIPPED    ROWS_LOADED     ROWS_REJECTED   ...
-----
...            32              0                32              0 ...
```

1 record(s) selected.

Return Status = 0

単一パーティション・データベースからの出力 (続き)。

```

... ROWS_DELETED      ROWS_COMMITTED      MSG_RETRIEVAL
... -----
...                0                32 SELECT SQLCODE, MSG_TEXT FROM
...                                     TABLE(SYSPROC.ADMIN_GET_MSGS(
...                                     '2203498_thx')) AS MSG

```

単一パーティション・データベースからの出力 (続き)。

```

... MSG_REMOVAL
... -----
... CALL SYSPROC.ADMIN_REMOVE_MSGS('2203498_thx')
...

```

注: この結果セットでは ROWS\_PARTITIONED および NUM\_AGENTINFO\_ENTRIES 列も戻されますが、これらの列は、複数パーティション・データベースへのロード時のみデータが取り込まれるので、NULL に設定されます。

以下のセクションは、複数パーティション・データベースからの出力例です。

Result set 1

```

-----
ROWS_READ      ROWS_REJECTED      ROWS_PARTITIONED      NUM_AGENTINFO_ENTRIES ...
-----
                32                0                32                5 ...
...
...

```

1 record(s) selected.

複数パーティション・データベースからの出力 (続き)。

```

... MSG_RETRIEVAL      MSG_REMOVAL
... -----
... SELECT DBPARTITIONNUM, AGENT_TYPE,      CALL SYSPROC.ADMIN_REMOVE_MSGS
...   SQLCODE, MSG_TEXT FROM TABLE          ('2203498_thx')
...   (SYSPROC.ADMIN_GET_MSGS
...   ('2203498_thx')) AS MSG

```

注: この結果セットでは ROWS\_SKIPPED、ROWS\_LOADED、ROWS\_DELETED、および ROWS\_COMMITTED 列も戻されますが、これらの列は、単一パーティション・データベースへのロード時のみデータが取り込まれるので、NULL に設定されます。

複数パーティション・データベースからの出力 (続き)。

Result set 2

```

-----
DBPARTITIONNUM      SQLCODE      TABSTATE      AGENTTYPE
-----
                10                0 NORMAL      LOAD
                20                0 NORMAL      LOAD
                30                0 NORMAL      LOAD
                20                0 NORMAL      PARTITION

```

1 record(s) selected.

Return Status = 0

### 例：XML データのロード

ユーザーは、表に挿入される文書を記述するために、XDS フィールドを持つデータ・ファイルを構成しました。内容は以下のとおりです。

```
1, "<XDS FIL=""file1.xml"" />"
2, "<XDS FIL='file2.xml' OFF='23' LEN='45' />"
```

最初の行では、XML 文書は file1.xml というファイルで示されます。区切り文字は二重引用符であり、二重引用符は XDS の内側にあるため、XDS 内にある二重引用符は二重になります。2 番目の行では、XML 文書は file2.xml というファイルで示され、バイト・オフセット 23 で始まります。長さは 45 バイトです。

ユーザーは、XML 列の構文解析または妥当性検査オプションを指定しないでロード・コマンドを発行し、データは正常にロードされました。

```
LOAD
FROM data.del of DEL INSERT INTO mytable
```

### 例：カーソルからの XML データのロード

データをカーソルからロードすることは、正規のリレーショナル列タイプを使用する場合と同じです。ユーザーは 2 つの表、T1 および T2 を持っており、それぞれは C1 という単一の XML 列で構成されます。T1 から T2 にロードするには、ユーザーは最初に次のようにカーソルを宣言します。

```
DECLARE
X1 CURSOR FOR SELECT C1 FROM T1;
```

その後、ユーザーは次のようにカーソル・タイプを使用して **LOAD** を発行できます。

```
LOAD FROM X1 of
CURSOR INSERT INTO T2
```

カーソル・タイプに XML 固有の **LOAD** オプションを適用する操作は、ファイルからロードする場合と同じです。

### 使用上の注意

- データは、入力ファイル内に並んでいる順序でロードされます。特定の順序にしたい場合には、ロードが試行される前にデータをソートしてください。ソース・データの順序を保持する必要がなければ、**ANYORDER** ファイル・タイプ修飾子を使用することを考慮してください。この修飾子については、以下の『ロード・ユーティリティーのファイル・タイプ修飾子』セクションを参照してください。
- ロード・ユーティリティーは、既存の定義に基づいて索引を作成します。ユニーク・キーの重複を処理するのに、例外表が使用されます。ユーティリティーは、参照整合性を強制したり、制約検査を実行したり、ロードする表に従属するマテリアライズ照会表を更新したりすることはありません。参照制約またはチェック制約を含む表は、**SET INTEGRITY** ペンディング状態になります。**REFRESH IMMEDIATE** として定義されているサマリー表、およびロードする表に依存する

サマリー表もまた、SET INTEGRITY ペンディング状態になります。これらの表に関して、SET INTEGRITY ペンディング (Set Integrity Pending) 状態を解除するには、SET INTEGRITY ステートメントを発行してください。ロード操作は、複製されたマテリアライズ照会表では実行できません。

- クラスタリング索引が表に存在する場合、ロード前にクラスタリング索引でデータをソートしてください。ただし、データはマルチディメンション・クラスタリング (MDC) 表にロードする前にソートする必要はありません。
- 保護された表へのロード時に例外表を指定すると、無効なセキュリティー・ラベルで保護されている行がその表に送られます。そのため、例外表にアクセスできるユーザーは、通常はアクセス権限のないデータにアクセスできてしまう可能性があります。セキュリティー・レベルを上げるために、誰に例外表アクセス権限を付与するかに注意し、行が修復されてロードする表にコピーされたら直ちにそれぞれの行を削除するとともに、使い終えた例外表は直ちにドロップしてください。
- 内部形式のセキュリティー・ラベルには、改行文字が含まれている可能性があります。DEL ファイル形式を使用するファイルをロードする場合、この改行文字が区切り文字と間違われることがあります。この問題が起きた場合は、LOAD コマンドで **delprioritychar** ファイル・タイプ修飾子を指定することによって、区切り文字に以前のデフォルト優先順位を使用してください。
- **LOAD** ユーティリティーは、操作の開始時に COMMIT ステートメントを発行しますが、これによってタイプ 2 接続の場合に、プロシージャは SQL30090N、理由コード 1 を戻します。
- **LOAD** コマンドで使用されるすべてのパスは、サーバー・コーディネーター・パーティション上の有効な完全修飾パスでなければなりません。
- DECLARE CURSOR ステートメントの実行中に指定した DATABASE キーワードが CURSOR ファイル・タイプを使用してロードを実行する場合、現在接続されているデータベース (ロード用) の認証に使用されるユーザー ID およびパスワードが (DECLARE CURSOR ステートメントの DATABASE オプションによって指定された) ソース・データベースの認証に使用されます。ユーザー ID またはパスワードがロード・データベースの接続に指定されない場合、ソース・データベースのユーザー ID とパスワードは DECLARE CURSOR ステートメントの実行中に指定する必要があります。
- 個々のパートが Windows システムから AIX システムにコピーされる、複数パートの PC/IXF ファイルのロードがサポートされます。すべてのファイルの名前は、LOAD コマンドで指定する必要があります。例えば、LOAD FROM DATA.IXF, DATA.002 OF IXF INSERT INTO TABLE1 となります。論理分割された PC/IXF ファイルから Windows オペレーティング・システムにロードすることはサポートされていません。
- 失敗した **LOAD** を再始動する場合、その動作は、BUILD フェーズで索引用に REBUILD モードの使用が強制されるという点で、既存の動作に従います。
- **COPY YES** オプションが使用されている場合、ロード・ユーティリティーはイメージ・ファイルの非常に大きなコピーを生成する場合があります。この動作は、ロードされる LOB/LF 列値ごとに **LOAD** コマンドがデータのバッファー全体をコピー・イメージに書き出すときに行われることになっています。バッファーは内部オブジェクトであり、そのサイズは複数の内部および外部要因によって決まります。一般的に、バッファー・サイズは 68 KB から数百 KB の間です。

- データベース間での XML 文書のロードはサポートされておらず、エラー・メッセージ SQL1407N が戻されます。
- **LOAD** ユーティリティーでは、fenced プロシージャを参照する列を含む表へのロードはサポートされていません。このような表に対して **LOAD** コマンドを発行すると、エラー・メッセージ SQL1376N を受け取ります。この制約事項に対処するには、ルーチンを unfenced に再定義するか、インポート・ユーティリティーを使用することができます。
- 暗黙的な非表示列がデータベース表に含まれる場合には、ロード操作に非表示列のデータを含めるかどうかを指定する必要があります。
- **IMPORT** ユーティリティーは、表内の列数とデータ・ファイル内のフィールド数の突き合わせを行いません。このユーティリティーは、データ・ファイル内に十分な量のデータがあるかどうかをチェックします。データ・ファイル内の行に十分なデータ列が含まれていない場合、その行は拒否されて警告メッセージが表示される (表の対応するデータのない列が **NOT NULL** として定義されている場合) か、警告メッセージは表示されずに正常に挿入されます (表の対応する列が **NULL** として定義されている場合)。一方、行に必要な数より多くの列が含まれている場合、必要な数の列が処理され、残りのデータ列は省略されます。警告メッセージは表示されません。

#### LOAD TERMINATE および LOAD RESTART ディクショナリー管理の要約

次の図表は、**TERMINATE** ディレクティブの下での **LOAD** 処理に関するコンプレッション・ディクショナリー管理の動作を要約しています。

表 44. **LOAD TERMINATE** ディクショナリー管理

表の <b>COMPRESS</b> 属性	<b>LOAD</b> 前に表の行データ・ディクショナリーが存在したかどうか	<b>LOAD</b> 前に XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	<b>TERMINATE: LOAD REPLACE KEEPDICTIONARY</b> または <b>LOAD INSERT</b>	<b>TERMINATE: LOAD REPLACE RESETDICTIONARY</b>
YES	YES	YES	既存のディクショナリーを保持します。	どちらのディクショナリーも保持しません。 <sup>2</sup>
YES	YES	NO	既存のディクショナリーを保持します。	何も保持しません。 <sup>2</sup>
YES	NO	YES	既存のディクショナリーを保持します。	何も保持しません。
YES	NO	NO	何も保持しません。	何も保持しません。
NO	YES	YES	既存のディクショナリーを保持します。	何も保持しません。
NO	YES	NO	既存のディクショナリーを保持します。	何も保持しません。
NO	NO	YES	既存のディクショナリーを保持します。	何も保持しません。
NO	NO	NO	何も行われません。	何も行われません。

注:

1. コンプレッション・ディクショナリーは、XML 列が DB2 バージョン 9.7 以降の表に追加された場合または表が Online Table Move を使用してマイグレーションされた場合にのみ、表の XML ストレージ・オブジェクトに対して作成できます。
2. 表でデータ・キャプチャーが使用可能にされているという特殊な場合には、表の行データ・ディクショナリーが保持されます。

**LOAD RESTART** は、到達した最後の整合点まで表を切り捨てます。 **LOAD RESTART** 処理の一部として、最後の **LOAD** 整合点が取られたときにコンプレッション・ディクショナリーが表にあった場合、そのディクショナリーが表に存在します。その場合、**LOAD RESTART** では新規ディクショナリーは作成されません。考えられる条件の要約については、表 4 を参照してください。

表 45. **LOAD RESTART** ディクショナリー管理

表の COMPRESS 属性	LOAD 整合点の前に 表の行データ・ディクショナリーが存在するか <sup>1</sup>	最後の LOAD 前に XML ストレージ・オブジェクト・ディクショナリーが存在したか <sup>2</sup>	RESTART: LOAD REPLACE KEEPDICTIONARY または LOAD INSERT	RESTART: LOAD REPLACE RESETDICTIONARY
YES	YES	YES	既存のディクショナリーを保持します。	既存のディクショナリーを保持します。
YES	YES	NO	既存の表の行データ・ディクショナリーを保存し、ADC の制約を受ける XML ディクショナリーを作成します。	既存の表の行データ・ディクショナリーを保存し、XML ディクショナリーを作成します。
YES	NO	YES	ADC の制約を受ける表の行データ・ディクショナリーを作成します。既存の XML ディクショナリーを保持します。	表の行データ・ディクショナリーを作成します。既存の XML ディクショナリーを保持します。
YES	NO	NO	ADC の制約を受ける表の行データ・ディクショナリーおよび XML ディクショナリーを作成します。	表の行データ・ディクショナリーおよび XML ディクショナリーを作成します。
NO	YES	YES	既存のディクショナリーを保持します。	既存のディクショナリーを除去します。
NO	YES	NO	既存の表の行データ・ディクショナリーを保存します。	既存の表の行データ・ディクショナリーを除去します。
NO	NO	YES	既存の XML ディクショナリーを保持します。	既存の XML ディクショナリーを除去します。
NO	NO	NO	何も行われません。	何も行われません。

注:

1. XML データのロード時には **SAVECOUNT** オプションは使用できません。ロード・フェーズ中に失敗したロード操作は、操作の最初から再開されます。

2. コンプレッション・ディクショナリーは、XML 列が DB2 バージョン 9.7 以降の表に追加された場合または表が Online Table Move を使用してマイグレーションされた場合にのみ、表の XML ストレージ・オブジェクトに対して作成できます。

### 結果セット情報

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。実行が成功すると、コマンドは追加情報を戻します。単一パーティション・データベースは 1 つの結果セットを返し、複数パーティション・データベースは 2 つの結果セットを戻します。

- 表 46: ロード操作の結果セット。
- 138 ページの表 47: 結果セット 2 には、複数パーティションでのロード操作における、データベース・パーティションごとの情報が含まれます。

表 46. LOAD コマンドによって戻される結果セット

列名	データ・タイプ	説明
ROWS_READ	BIGINT	ロード操作時の読み取り行数。
ROWS_SKIPPED	BIGINT	ロード操作開始前のスキップ行数。この情報は、単一パーティション・データベースの場合のみ戻されます。
ROWS_LOADED	BIGINT	ターゲット表にロードされた行の数。この情報は、単一パーティション・データベースの場合のみ戻されます。
ROWS_REJECTED	BIGINT	ターゲット表にロードできなかった行数。
ROWS_DELETED	BIGINT	ターゲット表にロードされなかった重複行の数。この情報は、単一パーティション・データベースの場合のみ戻されます。
ROWS_COMMITTED	BIGINT	処理された行の総数。この数は、ターゲット表に正常にロードされた行数と、スキップおよびリジェクトされた行数の合計です。この情報は、単一パーティション・データベースの場合のみ戻されます。
ROWS_PARTITIONED	BIGINT	すべてのデータベース分散エージェントによって分散された行数。この情報は、複数パーティション・データベースの場合のみ戻されます。
NUM_AGENTINFO_ENTRIES	BIGINT	複数パーティション・データベースの 2 番目の結果セットで戻される項目数。これは、ロード操作によって生成されたエージェント情報項目の数です。この情報は、複数パーティション・データベースの場合のみ戻されます。

表 46. LOAD コマンドによって戻される結果セット (続き)

列名	データ・タイプ	説明
MSG_RETRIEVAL	VARCHAR(512)	<p>このユーティリティーによって作成されたメッセージを取り出すために使用する SQL ステートメント。例えば、次のようになります。</p> <pre>SELECT SQLCODE, MSG FROM TABLE (SYSPROC.ADMIN_GET_MSGS ('2203498_thx')) AS MSG</pre> <p>この情報は、<b>MESSAGES ON SERVER</b> 節が指定された場合のみ戻されます。</p>
MSG_REMOVAL	VARCHAR(512)	<p>このユーティリティーによって作成されたメッセージをクリーンアップするために使用する SQL ステートメント。以下に例を示します。</p> <pre>CALL SYSPROC.ADMIN_REMOVE_MSGS ('2203498_thx')</pre> <p>この情報は、<b>MESSAGES ON SERVER</b> 節が指定された場合のみ戻されます。</p>

表 47. LOAD コマンドによって戻される結果セット 2 (複数パーティション・データベースのデータベース・パーティションごと)

列名	データ・タイプ	説明
DBPARTITIONNUM	SMALLINT	エージェントがロード操作を実行したデータベース・パーティション番号。
SQLCODE	INTEGER	ロード処理の結果の最終 SQLCODE。



表 47. LOAD コマンドによって戻される結果セット 2 (複数パーティション・データベースのデータベース・パーティションごと) (続き)

列名	データ・タイプ	説明
TABSTATE	VARCHAR(20)	<p>ロード操作完了後の表の状態。以下のいずれかです。</p> <ul style="list-style-type: none"> <li>• <b>LOADPENDING</b>: ロードは完了していないが、パーティション上の表は <b>LOAD PENDING</b> 状態のままになっていることを示します。このデータベース・パーティションに対して、ロードの再始動または終了操作を行う必要があります。</li> <li>• <b>NORMAL</b>: このデータベース・パーティションでのロードが正常に完了し、表の <b>LOAD IN PROGRESS</b> (または <b>LOAD PENDING</b>) 状態が解除されたことを示します。さらに制約処理が必要な場合に表はまだ <b>SET INTEGRITY PENDING</b> 状態である可能性があります。このインターフェースではその状態は報告されないため、注意してください。</li> <li>• <b>UNCHANGED</b>: エラーのためにロードは完了していないが、表の状態はまだ変わっていないことを示します。このデータベース・パーティションに対してロードの再始動または終了操作を行う必要はありません。</li> </ul> <p><b>注:</b> 起こりうるすべての表の状態がこのインターフェースで戻されるわけではありません。</p>
AGENTTYPE	VARCHAR(20)	<p>エージェント・タイプで、以下のいずれかです。</p> <ul style="list-style-type: none"> <li>• <b>FILE_TRANSFER</b></li> <li>• <b>LOAD</b></li> <li>• <b>LOAD_TO_FILE</b></li> <li>• <b>PARTITIONING</b></li> <li>• <b>PRE_PARTITIONING</b></li> </ul>

### ロード・ユーティリティー用のファイル・タイプ修飾子

表 48. ロード・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット

修飾子	説明
<b>anyorder</b>	<p>この修飾子は、 <b>cpu_parallelism</b> パラメーターと共に使用されます。ソース・データの順序を保持する必要がないことを指定します。そのため、SMP システムでは、パフォーマンスがかなり向上します。 <b>cpu_parallelism</b> の値が 1 になっていると、このオプションは無視されます。このオプションは、 <b>SAVECOUNT &gt; 0</b> の場合はサポートされません。整合点の後のクラッシュ・リカバリーでは、順序のとおりデータを読み込む必要があるからです。</p>

表 48. ロード・ユーティリティで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
<b>generatedignore</b>	この修飾子を指定すると、ロード・ユーティリティは、データ・ファイルに入っている、すべての生成済み列のデータを無視するようになります。この結果、すべての生成列の値はユーティリティによって生成されます。この修飾子は、 <b>generatedmissing</b> または <b>generatedoverride</b> 修飾子とともに使用することはできません。
<b>generatedmissing</b>	この修飾子が指定されている場合、ユーティリティは、生成列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものを見なします。この結果、すべての生成列の値はユーティリティによって生成されます。この修飾子は、 <b>generatedignore</b> または <b>generatedoverride</b> 修飾子とともに使用することはできません。
<b>generatedoverride</b>	<p>この修飾子は、(こうした列のタイプの通常の規則に反して) 表内のすべての生成列で、ユーザーのデータを受け入れるようにロード・ユーティリティに指示します。別のデータベース・システムからデータをマイグレーションする場合や、<b>ROLLFORWARD DATABASE</b> コマンドの <b>RECOVER DROPPED TABLE</b> オプションを使用してリカバリーしたデータから表をロードする場合は、この修飾子を使用すると便利です。この修飾子を使用した場合、NULL 不可の生成列でデータまたは NULL データの入っていない行はリジェクトされます (SQL3116W)。この修飾子が使用される場合、表は SET INTEGRITY PENDING 状態になります。ユーザー提供の値をチェックせずに表を SET INTEGRITY PENDING 状態から解放するには、ロード操作後に以下のコマンドを発行します。</p> <pre>SET INTEGRITY FOR <i>table-name</i> GENERATED COLUMN IMMEDIATE UNCHECKED</pre> <p>表の SET INTEGRITY PENDING 状態を解除し、ユーザー定義の値の検査を強制するには、ロード操作の後以下のコマンドを発行してください。</p> <pre>SET INTEGRITY FOR <i>table-name</i> IMMEDIATE CHECKED.</pre> <p>この修飾子が指定され、パーティション・キー、ディメンション・キー、または分散キーのいずれかに生成された列がある場合、<b>LOAD</b> コマンドが修飾子を <b>generatedignore</b> 自動的に変換し、ロードを進めます。この影響で、生成された列の値すべてが再生成されます。</p> <p>この修飾子は、<b>generatedmissing</b> または <b>generatedignore</b> 修飾子と共に使用することはできません。</p>
<b>identityignore</b>	この修飾子はロード・ユーティリティに対して、ID 列のデータがデータ・ファイル内に存在するが、それらのデータは無視するべきものであることを通知します。この結果として、すべて ID 値はこのユーティリティによって生成されます。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT のどちらの ID 列の場合も同じです。つまり、GENERATED ALWAYS 列の場合には、リジェクトされる行はありません。この修飾子は、 <b>identitymissing</b> または <b>identityoverride</b> 修飾子とともに使用することはできません。
<b>identitymissing</b>	この修飾子を指定すると、ユーティリティは、ID 列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものを見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS および GENERATED BY DEFAULT のどちらの ID 列の場合も同じです。この修飾子は、 <b>identityignore</b> または <b>identityoverride</b> 修飾子とともに使用することはできません。

表 48. ロード・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
<b>identityoverride</b>	<p>この修飾子は、GENERATED ALWAYS として定義した ID 列が、ロードする表に存在している場合にのみ使用するべきです。この修飾子はユーティリティーに対し、そのような列に関して、明示的な非 NULL データを受け入れる (これらのタイプの ID 列に関する通常の規則に反する) ように指示します。表を GENERATED ALWAYS として定義しなければならない状態で別のデータベース・システムからデータをマイグレーションする場合や、ROLLFORWARD DATABASE コマンドの DROPPED TABLE RECOVERY オプションを使用してリカバリーしたデータから表をロードする場合は、この修飾子を使用すると便利です。この修飾子を使用すると、ID 列にデータのない行や NULL データが入っている行は、リジェクトされます (SQL3116W)。この修飾子は、identitymissing または identityignore 修飾子とともに使用することはできません。このオプションが使用されていると、ロード・ユーティリティーは、表の ID 列内の値の固有性の保守または検証を行いません。</p>
<b>implicitlyhiddeninclude</b>	<p>この修飾子が指定されている場合、ユーティリティーは、入力データ・ファイルに暗黙非表示列へのデータが含まれると見なし、そのデータもロードします。この修飾子は、implicitlyhiddenmissing 修飾子と共に使用することはできません。複数の修飾子が指定されている場合の優先順位については、注: セクションを参照してください。</p>
<b>implicitlyhiddenmissing</b>	<p>この修飾子を指定すると、ユーティリティーは入力データ・ファイルに暗黙的な非表示列のデータが含まれないと見なし、ユーティリティーがそれらの暗黙的な非表示列の値を生成します。この修飾子は、implicitlyhiddeninclude 修飾子と共に使用することはできません。複数の修飾子が指定されている場合の優先順位については、注: セクションを参照してください。</p>
<b>indexfreespace=x</b>	<p><math>x</math> は、0 から 99 までの整数です。この値は、ロード操作で索引を再作成するときに各索引ページに残すフリー・スペースのパーセンテージとして解釈されます。INDEXING MODE INCREMENTAL を指定したロード操作では、このオプションが無視されます。ページの最初の項目は、制限なしで追加されます。それより後の項目は、フリー・スペースのパーセントしきい値内である場合に追加されます。デフォルト値は、CREATE INDEX の実行時に使用した値です。</p> <p>この値は、CREATE INDEX ステートメントに指定された PCTFREE 値よりも優先して使用されます。indexfreespace オプションの対象になるのは、索引のリーフ・ページだけです。</p>

表 48. ロード・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
<b>lobsinfile</b>	<p><i>lob-path</i> には、LOB データの入ったファイルへのパスを指定します。ASC、DEL、または IXF ロード入力ファイルには、LOB 列に LOB データが入っているファイルの名前が入っています。</p> <p>ファイル・タイプが CURSOR の場合、このオプションはサポートされていません。</p> <p><b>lobsinfile</b> 修飾子を使用するときには、LOB ファイルの配置場所を <b>LOBS FROM</b> 節で指定します。<b>LOBS FROM</b> 節を指定すると、<b>lobsinfile</b> の動作が暗黙的にアクティブになります。<b>LOAD</b> ユーティリティーは、データをロードするときに、LOB ファイルを検索するためのパスのリストを <b>LOBS FROM</b> 節から受け取ります。</p> <p>各パスには、データ・ファイル内で LOB ロケーション指定子 (LLS) によって示される 1 つ以上の LOB の入った、少なくとも 1 つのファイルが組み込まれます。LLS は、LOB ファイル・パスに保管されるファイル内の LOB のロケーションのストリング表現です。LLS の形式は、<i>filename.ext.nnn.mmm/</i> です。ここで、<i>filename.ext</i> は、LOB が含まれているファイルの名前、<i>nnn</i> は、そのファイルに入っている LOB のオフセット (バイト単位)、<i>mmm</i> は、その LOB の長さ (バイト単位) です。例えば、ストリング <i>db2exp.001.123.456/</i> がデータ・ファイルに保管される場合、LOB はファイル <i>db2exp.001</i> のオフセット 123 に位置し、456 バイト長です。</p> <p>NULL LOB を指定するには、サイズに -1 と入力します。サイズを 0 と指定すると、長さが 0 の LOB として扱われます。長さが -1 の NULL LOB の場合、オフセットとファイル名は無視されます。例えば、NULL LOB の LLS は、<i>db2exp.001.7.-1/</i> のようになります。</p>
<b>noheader</b>	<p>ヘッダー検査コードをスキップします (単一パーティション・データベースのパーティション・グループに存在する表へのロード操作にのみ適用します)。</p> <p>単一パーティションのデータベース・パーティション・グループに存在する表に対してデフォルトの MPP ロード (モード PARTITION_AND_LOAD) が使用される場合、ファイルにはヘッダーが組み込まれないと想定されます。したがって、<b>noheader</b> 修飾子を指定する必要はありません。LOAD_ONLY モードが使用される場合、ファイルにはヘッダーが付いていると想定されます。<b>noheader</b> 修飾子が必要になるのは、ヘッダーのないファイルを使用して LOAD_ONLY 操作を実行する場合に限られます。</p>
<b>norowwarnings</b>	<p>リジェクトされた行についてのすべての警告を抑止します。</p>
<b>pagefreespace=x</b>	<p><i>x</i> は、0 から 100 までの整数です。この値は、各データ・ページ内でフリー・スペースとして残される部分のパーセンテージとして解釈されます。最小の行サイズのために指定値が無効な場合 (例えば、最低でも 3,000 バイトの長さが必要な行で、<i>x</i> 値が 50 になっている場合など) は、行が新しいページに配置されます。値として 100 を指定すると、各行が新しいページに配置されます。表の PCTFREE 値は、ページごとに指定されたフリー・スペースの量を決定します。ロード操作の <b>pagefreespace</b> 値または表の PCTFREE 値が設定されていないと、ユーティリティーはそれぞれのページで可能なかぎり多くのスペースを満たします。<b>pagefreespace</b> に設定されている値は、表で指定されている PCTFREE 値をオーバーライドします。</p>

表 48. ロード・ユーティリティで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
<b>periodignore</b>	この修飾子はロード・ユーティリティに対して、期間列のデータがデータ・ファイル内に存在するが、それらのデータは無視するべきものであることを通知します。この修飾子が指定された場合、すべての期間列の値はユーティリティによって生成されます。この修飾子は、 <b>periodmissing</b> 修飾子および <b>periodoverride</b> 修飾子のどちらとも、共に使用することはできません。
<b>periodmissing</b>	この修飾子が指定されている場合、ユーティリティは、期間列のデータが入力データ・ファイルに入っていないものと見なします。この修飾子が指定された場合、すべての期間列の値はユーティリティによって生成されます。この修飾子は、 <b>periodignore</b> 修飾子および <b>periodoverride</b> 修飾子のどちらとも、共に使用することはできません。
<b>periodoverride</b>	この修飾子は、システム期間テンポラル表の GENERATED ALWAYS AS ROW BEGIN 列および GENERATED ALWAYS AS ROW END 列で、ユーザーのデータを受け入れるようにロード・ユーティリティに指示します。この動作は、このようなタイプの列に対する通常の規則とは異なっています。この修飾子は履歴データを維持する場合、およびシステム期間テンポラル表にタイム・スタンプを含むデータをロードする場合に、役立てることができます。この修飾子を使用すると、データが入っていない行や、ROW BEGIN 列または ROW END 列に対する NULL データはリジェクトされます。
<b>rowchangetimestampignore</b>	この修飾子はロード・ユーティリティに対して、ROW CHANGE TIMESTAMP 列のデータがデータ・ファイル内に存在するが、それらのデータは無視するべきものであることを通知します。この結果、すべての ROW CHANGE TIMESTAMP 列がユーティリティによって生成されます。この動作は、GENERATED ALWAYS 列でも GENERATED BY DEFAULT 列でも同じです。つまり、GENERATED ALWAYS 列の場合には、リジェクトされる行はありません。この修飾子は、 <b>rowchangetimestampmissing</b> または <b>rowchangetimestampoverride</b> 修飾子とともに使用することはできません。
<b>rowchangetimestampmissing</b>	この修飾子を指定すると、ユーティリティは、行変更タイム・スタンプ列のデータが入力データ・ファイルに入っていない (NULL も入っていない) ものと見なし、行ごとに値を生成します。この動作は、GENERATED ALWAYS 列でも GENERATED BY DEFAULT 列でも同じです。この修飾子は、 <b>rowchangetimestampignore</b> または <b>rowchangetimestampoverride</b> 修飾子とともに使用することはできません。
<b>rowchangetimestampoverride</b>	この修飾子は、GENERATED ALWAYS として定義した ROW CHANGE TIMESTAMP 列が、ロードする表に存在している場合のみ使用するべきです。この修飾子はユーティリティに対し、そのような列に関して、明示的な非 NULL データを受け入れる (これらのタイプの ROW CHANGE TIMESTAMP 列に関する通常の規則に反する) ように指示します。表を GENERATED ALWAYS として定義しなければならない状況で別のデータベース・システムからデータをマイグレーションする場合や、 <b>ROLLFORWARD DATABASE</b> コマンドの <b>DROPPED TABLE RECOVERY</b> オプションを使用してリカバリーしたデータから表をロードする場合は、この修飾子を使用すると便利です。この修飾子を使用すると、データが入っていない行や ROW CHANGE TIMESTAMP 列に対する NULL データはすべてリジェクトされます (SQL3116W)。この修飾子は、 <b>rowchangetimestampmissing</b> または <b>rowchangetimestampignore</b> 修飾子とともに使用することはできません。このオプションが使用されていると、ロード・ユーティリティは、表の ROW CHANGE TIMESTAMP 列内の値の固有性の保守または検証を行いません。

表 48. ロード・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
<p><b>seclabelchar</b></p>	<p>入力ソース・ファイル内のセキュリティ・ラベルが、デフォルトのエンコードされた数値形式ではなく、セキュリティ・ラベル値の文字列形式であることを示します。 <b>LOAD</b> は、ロード時に各セキュリティ・ラベルを内部形式に変換しません。文字列が適切な形式ではない場合、行はロードされず、警告 (SQLSTATE 01H53、SQLCODE SQL3242W) が戻されます。文字列が表を保護するセキュリティ・ポリシーの一部である有効なセキュリティ・ラベルを表していない場合、行はロードされず、警告 (SQLSTATE 01H53、SQLCODE SQL3243W) が戻されます。</p> <p><b>seclabelname</b> 修飾子を指定した場合は、この修飾子を指定できません。同時に指定すると、ロードは失敗し、エラー (SQLCODE SQL3525N) が戻されます。</p> <p>単一の DB2SECURITYLABEL 列で構成される表がある場合、データ・ファイルは例えば次のようになります。</p> <pre>"CONFIDENTIAL:ALPHA:G2" "CONFIDENTIAL;SIGMA:G2" "TOP SECRET:ALPHA:G2"</pre> <p>このデータのロードまたはインポートでは、以下のように <b>seclabelchar</b> ファイル・タイプ修飾子を使用する必要があります。</p> <pre>LOAD FROM input.del OF DEL MODIFIED BY SECLABELCHAR INSERT INTO t1</pre>
<p><b>seclabelname</b></p>	<p>入力ソース・ファイル内のセキュリティ・ラベルが、デフォルトのエンコードされた数値形式ではなく、名前によって指定されることを示します。 <b>LOAD</b> は、その名前に対応する適切なセキュリティ・ラベルがあれば、その名前をそのセキュリティ・ラベルに変換します。表を保護するセキュリティ・ポリシーで、指定された名前のセキュリティ・ラベルが存在しない場合、行はロードされず、警告 (SQLSTATE 01H53、SQLCODE SQL3244W) が戻されます。</p> <p><b>seclabelchar</b> 修飾子を指定した場合は、この修飾子を指定できません。同時に指定すると、ロードは失敗し、エラー (SQLCODE SQL3525N) が戻されます。</p> <p>単一の DB2SECURITYLABEL 列で構成される表がある場合、データ・ファイルは以下のようなセキュリティ・ラベル名で構成される可能性があります。</p> <pre>"LABEL1" "LABEL1" "LABEL2"</pre> <p>このデータのロードまたはインポートでは、以下のように <b>seclabelname</b> ファイル・タイプ修飾子を使用する必要があります。</p> <pre>LOAD FROM input.del OF DEL MODIFIED BY SECLABELNAME INSERT INTO t1</pre> <p><b>注:</b> ファイル・タイプが ASC の場合、セキュリティ・ラベルの名前の後にスペースがあれば、それも名前の一部として解釈されます。これを回避するには、<b>striptblanks</b> ファイル・タイプ修飾子を使用してスペースが除去されるようにします。</p>

表 48. ロード・ユーティリティーで有効なファイル・タイプ修飾子: すべてのファイル・フォーマット (続き)

修飾子	説明
<b>total freespace=x</b>	<p><math>x</math> は、0 以上の整数です。この値は表内の合計ページのうち、表の終わりにフリー・スペースとして追加される部分のパーセンテージと解釈されます。例えば、<math>x</math> が 20 で、データのロード後に表に 100 個のデータ・ページがある場合は、20 個の空ページが追加されます。その表のデータ・ページの合計数は 120 になります。データ・ページの総数は、表の索引ページの数には影響を与えません。このオプションは、索引オブジェクトには影響を与えません。このオプションを指定して 2 つのロードが行われる場合、2 番目のロードは、最初のロードによって最後に付加された余分のスペースを再利用しません。</p>
<b>transactionidignore</b>	<p>この修飾子はロード・ユーティリティーに対して、TRANSACTION START ID 列のデータがデータ・ファイル内に存在するが、それらのデータは無視するべきものであることを通知します。この修飾子が指定された場合、TRANSACTION START ID 列の値はユーティリティーによって生成されます。この修飾子は、<b>transactionidmissing</b> 修飾子および <b>transactionidoverride</b> 修飾子のどちらとも、共に使用することはできません。</p>
<b>transactionidmissing</b>	<p>この修飾子が指定されている場合、ユーティリティーは、TRANSACTION START ID 列のデータが入力データ・ファイルに入っていないものと見なします。この修飾子が指定された場合、TRANSACTION START ID 列の値はユーティリティーによって生成されます。この修飾子は、<b>transactionidignore</b> 修飾子および <b>transactionidoverride</b> 修飾子のどちらとも、共に使用することはできません。</p>
<b>transactionidoverride</b>	<p>この修飾子は、システム期間テンポラル表の GENERATED ALWAYS AS TRANSACTION START ID 列で、ユーザーのデータを受け入れるようにロード・ユーティリティーに指示します。この動作は、このタイプの列に対する通常の規則とは異なっています。この修飾子を使用すると、データが入っていない行や、TRANSACTION START ID 列に対する NULL データはリジェクトされます。</p>
<b>usedefaults</b>	<p>ターゲット表の列のソース列が指定されているが、1 つ以上の行インスタンスのデータが入っていない場合は、デフォルト値がロードされます。欠落データの例を以下に示します。</p> <ul style="list-style-type: none"> <li>• DEL ファイルの場合: 列の値として、2 つの連続した列区切り (,) や、任意の数のスペースで分離した 2 つの連続する列区切り (, ) が指定されている。</li> <li>• DEL/ASC ファイルの場合: 列が不足している行、または元の指定には十分な長さでない行。ASC ファイルの場合: NULL 列値は明示的に欠落しているとは見なされず、NULL 列値にはデフォルトが置換されません。数値、日付、時刻、タイム・スタンプの列では、全桁スペース文字で NULL 列値を表記します。また、どのタイプの列でも、NULL INDICATOR を使用すれば、その列が NULL であることを示せます。</li> </ul> <p>このオプションが指定されていない場合、行インスタンスのソース列にデータがないと、以下のいずれかの処理が行われます。</p> <ul style="list-style-type: none"> <li>• DEL/ASC ファイルの場合: 列が NULL 可能であれば、NULL がロードされます。列が NULL 可能でない場合、ユーティリティーはその行をリジェクトします。</li> </ul>

表 49. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL)

修飾子	説明
<b>codepage=x</b>	<p>x は、ASCII 文字ストリングです。この値は、入力データ・セット内のデータのコード・ページとして解釈されます。ロード操作時に、文字データ (および文字内で指定された数値データ) は、このコード・ページからデータベースのコード・ページへ変換されます。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>• DBCS のみ (GRAPHIC)、混合 DBCS、および EUC の場合、区切り文字の範囲は x00 から x3F に制限されます。</li> <li>• EBCDIC コード・ページで指定された DEL データの場合、区切り文字は DBCS のシフトイン文字およびシフトアウト文字と一致しない場合があります。</li> <li>• nullindchar では、標準の ASCII セットのコード・ポイント x20 から x7F の範囲に含まれているシンボルを指定する必要があります。これは、ASCII 記号およびコード・ポイントを示します。EBCDIC データでは、コード・ポイントが異なるとしても、対応する記号を使用できます。</li> </ul> <p>ファイル・タイプが CURSOR の場合、このオプションはサポートされていません。</p>
<b>dateformat="x"</b>	<p>x はソース・ファイルの日付のフォーマットです。<sup>1</sup> 有効な日付エレメントは次のとおりです。</p> <p>YYYY - 年 (0000 から 9999 の範囲の 4 桁の数)  M - 月 (1 から 12 の範囲の 1 桁または 2 桁の数)  MM - 月 (01 から 12 の 2 桁の数。  M とは相互に排他的)  D - 日 (1 から 31 の範囲の 1 桁または 2 桁の数)  DD - 日 (01 から 31 の範囲の 2 桁の数。  D とは相互に排他的)  DDD - 元日から数えた日数 (001 から 366 の範囲の 3 桁の数。  他の日または月エレメントとは相互に排他的)</p> <p>デフォルト値の 1 が、指定されない各エレメントに割り当てられます。日付形式の例を以下に示します。</p> <p>"D-M-YYYY"  "MM.DD.YYYY"  "YYYYDDD"</p>



表 49. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
<b>dumpfile = x</b>	<p>x は、リジェクトされた行を書き込む例外ファイルの (サーバー・データベース・パーティションによる) 完全修飾名です。1 レコードにつき、最大で 32 KB のデータが書き込まれます。以下に、ダンプ・ファイルの指定方法の例を示します。</p> <pre>db2 load from data of del   modified by dumpfile = /u/user/filename   insert into table_name</pre> <p>ファイルは、インスタンスの所有者によって作成されて所有されます。デフォルトのファイル権限をオーバーライドするには、<b>dumpfileaccessall</b> ファイル・タイプ修飾子を使用します。</p> <p><b>注:</b></p> <ol style="list-style-type: none"> <li>パーティション・データベース環境の場合、パスはロードを実行するデータベース・パーティションにローカルなものでなければなりません。それによって、並行して実行される複数のロード操作が同じファイルに書き込むことを防ぐことができます。</li> <li>ファイルの内容は、非同期バッファー・モードでディスクに書き込まれます。ロード操作が失敗した場合や割り込みが発生した場合は、ディスクにコミットされたレコードの数を確実に把握する方法がありません。<b>LOAD RESTART</b> 後の整合性も保証できません。ファイルが完全であるとされるのは、1 回のパスの中で開始して完了するロード操作の場合だけです。</li> <li>指定されたファイルが既に存在する場合は、再作成されずに切り捨てられます。</li> </ol>
<b>dumpfileaccessall</b>	<p>ダンプ・ファイルの作成時に、読み取りアクセスを OTHERS に付与します。</p> <p>このファイル・タイプ修飾子が有効なのは、以下の場合のみです。</p> <ol style="list-style-type: none"> <li><b>dumpfile</b> ファイル・タイプ修飾子と一緒に使用された場合。</li> <li>ロード・ターゲット表に対してユーザーが <b>SELECT</b> 特権をもっている場合。</li> <li>UNIX オペレーティング・システムに置かれている DB2 サーバー・データベース・パーティション上で発行された場合。</li> </ol> <p>指定されたファイルが既に存在する場合、その権限は変更されません。</p>
<b>fastparse</b>	<p>使用に際しては、注意が必要です。ユーザー指定の列値の構文検査が削減されるので、パフォーマンスは向上します。表は、体系的な正確さが確保されます (セグメント化違反またはトラップを防ぐための十分なデータ・チェックがユーティリティーで実行されます) が、データの一貫性の妥当性検査は行われません。データに一貫性があり、正確であることが確実な場合のみ、このオプションを使用してください。例えば、ユーザー指定のデータに無効なタイム・スタンプ列値 :1&gt;0-00-20-07.11.12.000000 が含まれている場合でも、<b>fastparse</b> が指定されていれば、その値は表に挿入されてしまいますが、<b>fastparse</b> が指定されていなければ、その値はリジェクトされます。</p>
<b>implieddecimal</b>	<p>暗黙指定されている小数点の位置が列定義によって決定され、値の終わりにあるとは見なされなくなります。例えば、値 12345 は、12345.00 ではなく、123.45 として DECIMAL(8,2) 列にロードされます。</p> <p>この修飾子は、<b>packeddecimal</b> 修飾子と共に使用することはできません。</p>

表 49. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
<b>timeformat="x"</b>	<p>x はソース・ファイル内の時刻のフォーマットです。<sup>1</sup> 有効な時刻エレメントは以下のとおりです。</p> <p>H - 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の範囲の 1 桁または 2 桁の数)</p> <p>HH - 時 (12 時間制の場合は 00 から 12、24 時間制では 00 から 24 の範囲の 2 桁の数; H と相互に排他的)</p> <p>M - 分 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p>MM - 分 (00 から 59 の範囲の 2 桁の数。 M とは相互に排他的)</p> <p>S - 秒 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p>SS - 秒 (00 から 59 の範囲の 2 桁の数。 S と相互に排他的)</p> <p>SSSSS - 夜中の 12 時から数えた秒数 (00000 から 86400 の範囲の 5 桁の数。 他の時刻エレメントとは相互に排他的)</p> <p>TT - 午前/午後の指定子 (AM または PM)</p> <p>指定されない各エレメントには、デフォルト値の 0 が割り当てられます。時刻フォーマットの例を以下に示します。</p> <p>"HH:MM:SS" "HH.MM TT" "SSSSS"</p>
<b>timestampformat="x"</b>	<p>x はソース・ファイルのタイム・スタンプのフォーマットです。<sup>1</sup> 有効なタイム・スタンプ・エレメントは以下のとおりです。</p> <p>YYYY - 年 (0000 から 9999 の範囲の 4 桁の数)</p> <p>M - 月 (1 から 12 の範囲の 1 桁または 2 桁の数)</p> <p>MM - 月 (01 から 12 の 2 桁の数。 M および MMM とは相互に排他的)</p> <p>MMM - 月 (大文字小文字を区別しない月名の 3 文字の省略形。 M と MM とは相互に排他的)</p> <p>D - 日 (1 から 31 の範囲の 1 桁または 2 桁の数)</p> <p>DD - 日 (01 から 31 の範囲の 2 桁の数。D とは相互に排他的)</p> <p>DDD - 元日から数えた日数 (001 から 366 の範囲の 3 桁の数。 他の日または月のエレメントとは相互に排他的)</p> <p>H - 時 (12 時間制の場合は 0 から 12、24 時間制では 0 から 24 の範囲の 1 桁または 2 桁の数。)</p> <p>HH - 時 (12 時間制の場合は 00 から 12、24 時間制では 0 から 24 の範囲の 2 桁の数。 H と相互に排他的)</p> <p>M - 分 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p>MM - 分 (00 から 59 の範囲の 2 桁の数。 M (分) とは相互に排他的)</p> <p>S - 秒 (0 から 59 の範囲の 1 桁または 2 桁の数)</p> <p>SS - 秒 (00 から 59 の範囲の 2 桁の数。 S と相互に排他的)</p> <p>SSSSS - 夜中の 12 時から数えた秒数 (00000 から 86400 の範囲の 5 桁の数。 他の時刻エレメントとは相互に排他的)</p> <p>U (1 から 12 時) - 小数秒 (U のオカレンス数は、各桁を 0 から 9 の範囲として、桁数を表します)</p> <p>TT - 午前/午後の指定子 (AM または PM)</p>

表 49. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
<p><b>timestampformat="x"</b> (Continued)</p>	<p>YYYY、M、MM、D、DD、または DDD エlementが指定されていない場合、デフォルト値の 1 が割り当てられます。MMM Elementが指定されていない場合、デフォルト値の「Jan」が割り当てられます。他のElementが指定されていない場合には、デフォルト値の 0 が割り当てられます。タイム・スタンプ・フォーマットの例を以下のセクションに示します。</p> <p style="text-align: center;">"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>MMM Elementの有効な値は、「jan」、「feb」、「mar」、「apr」、「may」、「jun」、「jul」、「aug」、「sep」、「oct」、「nov」、および「dec」です。これらの値では、大/小文字は区別されません。</p> <p><b>timestampformat</b> 修飾子を指定しなかった場合、ロード・ユーティリティーは、タイム・スタンプ・フィールドで以下の 2 つの有効な形式のいずれかを使用します。</p> <p>YYYY-MM-DD-HH.MM.SS          YYYY-MM-DD HH:MM:SS</p> <p>ロード・ユーティリティーは、DD と HH の間の区切り記号を調べてフォーマットを選択します。ダッシュ「-」になっていれば、ロード・ユーティリティーは、通常のダッシュとドットの形式 (YYYY-MM-DD-HH.MM.SS) を使用します。区切り文字がブランク・スペースの場合、ロード・ユーティリティーはコロン「:」を使用して、HH、MM、および SS を区切ります。</p> <p>どちらのフォーマットでも、マイクロ秒フィールド (UUUUUU) が含まれる場合、ロード・ユーティリティーは区切り文字としてドット「.」を使用します。YYYY-MM-DD-HH.MM.SS.UUUUUU も YYYY-MM-DD HH:MM:SS.UUUUUU も有効です。</p> <p>次の例では、ユーザー定義の日時形式を指示するデータを、schedule という表にロードする方法を示します。</p> <pre>db2 load from delfile2 of del       modified by timestampformat="yyyy.mm.dd hh:mm tt"       insert into schedule</pre>

表 49. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASCII ファイル・フォーマット (ASC/DEL) (続き)

修飾子	説明
<p><b>usegraphiccodepage</b></p>	<p><b>usegraphiccodepage</b> が指定された場合、GRAPHIC または 2 バイト文字ラージ・オブジェクト (DBCLOB) データ・フィールドにロードされるデータは、GRAPHIC コード・ページであると見なされます。データの残りは、文字コード・ページであると見なされます。GRAPHIC コード・ページは、文字コード・ページと関連付けられます。<b>LOAD</b> は、<b>codepage</b> 修飾子が指定されている場合はその修飾子、または <b>codepage</b> 修飾子が指定されていない場合は、データベースのコード・ページを通じて、文字コード・ページを決定します。</p> <p>この修飾子は、リカバリーされている表に GRAPHIC データがある場合のみ、ドロップ済み表のリカバリーによって生成された区切りデータ・ファイルとともに使用される必要があります。</p> <p><b>制約事項</b></p> <p><b>EXPORT</b> ユーティリティーで作成された DEL ファイルは、1 つのコード・ページのみでエンコードされたデータを含んでいるため、これらのファイルで <b>usegraphiccodepage</b> 修飾子を指定することはできません。 <b>usegraphiccodepage</b> 修飾子はまた、ファイル内の 2 バイト文字ラージ・オブジェクト (DBCLOB) には無視されます。</p>
<p><b>xmlchar</b></p>	<p>XML 文書が文字コード・ページでエンコードされていることを示します。</p> <p>このオプションは、指定の文字コード・ページでエンコードされていてもエンコード宣言を含まない XML 文書を処理するために役立ちます。</p> <p>各文書で、宣言タグが存在してエンコード属性が含まれる場合、そのエンコード方式は文字コード・ページと一致する必要があります。一致しない場合、その文書を含む行はリジェクトされます。文字コード・ページは <b>codepage</b> ファイル・タイプ修飾子で指定されている値であるか、または指定がない場合はアプリケーションのコード・ページであることに注意してください。デフォルトでは、文書は Unicode でエンコードされているか、またはエンコード属性のある宣言タグを含んでいません。</p>
<p><b>xmlgraphic</b></p>	<p>XML 文書が指定された GRAPHIC コード・ページでエンコードされていることを示します。</p> <p>このオプションは、特定の GRAPHIC コード・ページでエンコードされていてもエンコード宣言を含まない XML 文書を処理するために役立ちます。</p> <p>各文書で、宣言タグが存在してエンコード属性が含まれる場合、そのエンコード方式は GRAPHIC コード・ページと一致する必要があります。一致しない場合、その文書を含む行はリジェクトされます。GRAPHIC コード・ページは <b>codepage</b> ファイル・タイプ修飾子で指定されている値のグラフィック・コンポーネントであるか、または指定がない場合はアプリケーションのコード・ページのグラフィック・コンポーネントであることに注意してください。デフォルトでは、文書は Unicode でエンコードされているか、またはエンコード属性のある宣言タグを含んでいます。</p>

表 50. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASC ファイル・フォーマット (区切り文字で区切られていない ASCII)

修飾子	説明
<b>binarynumerics</b>	<p>数値データ (DECIMAL 以外) は、文字表記ではなく、バイナリー形式でなければなりません。これによって、コストの大きい変換操作を避けることができます。</p> <p>このオプションがサポートされるのは、定位置 ASC において、<b>reclen</b> オプションによって固定長レコードが指定されている場合だけです。</p> <p>以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>• BIGINT、INTEGER、および SMALLINT を除き、データ・タイプ間の変換は実行されません。</li> <li>• データ長は、それぞれのターゲット列定義と一致している必要があります。</li> <li>• FLOAT は、IEEE 浮動小数点フォーマットでなければなりません。</li> <li>• ロード・ソース・ファイル中のバイナリー・データは、ロード操作を実行するプラットフォームに関係なく、ビッグ・エンディアンであると見なされます。</li> </ul> <p>この修飾子の影響を受ける列のデータに NULL があってはなりません。この修飾子を使用すると、ブランク (通常は NULL と解釈される) は、バイナリー値であると解釈されます。</p>
<b>nochecklengths</b>	<p><b>nochecklengths</b> を指定した場合は、ソース・データの中にターゲット表の列のサイズを超える列定義がある場合であっても、各行のロードが試みられます。コード・ページ変換によってソース・データが縮小されれば、そのような行であったとしても正常にロードすることができます。例えば、ソースに 4 バイトの EUC データがある場合、それがターゲットで 2 バイトの DBCS データに縮小されれば、必要なスペースは半分で済みます。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。</p>
<b>nullindchar=<i>x</i></b>	<p><i>x</i> は、単一文字です。NULL 値を示す文字を <i>x</i> に変更します。<i>x</i> のデフォルト値は <i>Y</i> です。<sup>2</sup></p> <p>文字が 1 つの英字である場合を除いて、この修飾子は EBCDIC データ・ファイルで大文字小文字を区別します。例えば、NULL 標識文字を文字 <i>N</i> に指定した場合、<i>n</i> も NULL 標識と認識されます。</p>

表 50. ロード・ユーティリティーで有効なファイル・タイプ修飾子: ASC ファイル・フォーマット (区切り文字で区切られていない ASCII) (続き)

修飾子	説明
<b>packeddecimal</b>	<p><b>binarynumerics</b> 修飾子は DECIMAL フィールド・タイプで構成されないため、パック 10 進数データを直接ロードします。</p> <p>このオプションがサポートされるのは、定位置 ASC において、<b>reclen</b> オプションによって固定長レコードが指定されている場合だけです。</p> <p>符号ニブル用にサポートされる値は以下のとおりです。</p> <pre> + = 0xC 0xA 0xE 0xF - = 0xD 0xB </pre> <p>この修飾子の影響を受ける列のデータに NULL があってはなりません。この修飾子を使用すると、ブランク (通常は NULL と解釈される) は、バイナリー値であると解釈されます。</p> <p>サーバーのプラットフォームには関係なく、ロードのソース・ファイルに入っているバイナリー・データのバイト順はビッグ・エンディアンであることが前提となっています。つまり、この修飾子を Windows オペレーティング・システムで使用する場合も、バイト順を逆にしてはなりません。</p> <p>この修飾子は、<b>implieddecimal</b> 修飾子と共に使用することはできません。</p>
<b>reclen=x</b>	<p><math>x</math> は、最大値 32767 の整数です。各行では <math>x</math> 個の文字が読み取られ、行の終わりを示す改行文字は使用されません。</p>
<b>striptblanks</b>	<p>データを可変長フィールドにロードする際に、後書きブランク・スペースを切り捨てます。このオプションを指定しない場合、ブランク・スペースはそのまま保持されます。</p> <p>このオプションは、<b>striptnulls</b> と一緒に指定することはできません。これらは、相互に排他的なオプションです。このオプションは、廃止された <b>t</b> オプション (下位互換性のためだけにサポートされる) に代わるものです。</p>
<b>striptnulls</b>	<p>データを可変長フィールドにロードする際に、後書き NULL (0x00 文字) を切り捨てます。このオプションを指定しない場合、NULL はそのまま保持されます。</p> <p>このオプションは、<b>striptblanks</b> と一緒に指定することはできません。これらは、相互に排他的なオプションです。このオプションは、廃止された <b>padwithzero</b> オプション (下位互換性のためだけにサポートされる) に代わるものです。</p>
<b>zoneddecimal</b>	<p>ゾーン 10 進数データをロードします。DECIMAL フィールド・タイプは、<b>binarynumerics</b> 修飾子の対象に含まれていません。このオプションがサポートされるのは、定位置 ASC において、<b>reclen</b> オプションによって固定長レコードが指定されている場合だけです。</p> <p>ハーフバイト符号値は、以下の値のいずれかになります。</p> <pre> + = 0xC 0xA 0xE 0xF 0x3 - = 0xD 0xB 0x7 </pre> <p>数字としてサポートされている値は、0x0 から 0x9 です。</p> <p>ゾーンとしてサポートされている値は、0x3 および 0xF です。</p>

表 51. ロード・ユーティリティーで有効なファイル・タイプ修飾子: DEL ファイル・フォーマット (区切り文字で区切られている ASCII)

修飾子	説明
<b>chardelx</b>	<p><i>x</i> は単一文字のストリング区切り文字です。デフォルト値は、二重引用符 (") です。指定した文字は、文字ストリングを囲むために、二重引用符の代わりに使用されます。<sup>23</sup> 文字ストリング区切り文字として明示的に二重引用符 (") を指定したい場合、以下のように指定します。</p> <pre>modified by chardel""</pre> <p>以下のように、文字ストリング区切りとして単一引用符 (') を指定することもできます。</p> <pre>modified by chardel''</pre>
<b>coldelx</b>	<p><i>x</i> は単一文字の列区切り文字です。デフォルト値はコンマ (,) です。指定した文字は、列の終わりを表すために、コンマの代わりに使用されます。<sup>23</sup></p>
<b>decplusblank</b>	<p>正符号文字。正の 10 進値の接頭部として、正符号 (+) ではなくブランク・スペースを使用します。デフォルトのアクションでは、正の 10 進数の前に正符号 (+) が付けられます。</p>
<b>decptx</b>	<p><i>x</i> は、小数点文字としてピリオドの代わりに使用される単一文字です。デフォルト値はピリオド (.) です。指定した文字は、小数点文字としてピリオドの代わりに使用されます。<sup>23</sup></p>
<b>delprioritychar</b>	<p>区切り文字の現在のデフォルト優先順位は、(1) レコード区切り文字、(2) 区切り文字、(3) 列区切り文字です。この修飾子を使用すると、区切り文字の優先順位が (1) 区切り文字、(2) レコード区切り文字、(3) 列区切り文字に戻り、以前の優先順位に依存している既存のアプリケーションが保護されます。構文:</p> <pre>db2 load ... modified by delprioritychar ...</pre> <p>例えば、以下のような DEL データ・ファイルがあるとします。</p> <pre>"Smith, Joshua",4000,34.98&lt;row delimiter&gt; "Vincent,&lt;row delimiter&gt;, is a manager", ... ... 4005,44.37&lt;row delimiter&gt;</pre> <p><b>delprioritychar</b> 修飾子を指定しているため、このデータ・ファイルは、2 行だけになります。2 番目の &lt;row delimiter&gt; は 2 番目の行の最初のデータ列の一部と解釈されますが、1 番目と 3 番目の &lt;row delimiter&gt; は実レコードの区切り文字と解釈されます。この修飾子が指定されていない場合、このデータ・ファイルでは 3 行になり、各行は &lt;row delimiter&gt; によって区切られます。</p>
<b>keepblanks</b>	<p>タイプが CHAR、VARCHAR、LONG VARCHAR、または CLOB の各フィールドの前後のブランクを保持します。このオプションを指定しないと、区切り文字で囲まれていないすべての前後のブランクは除去され、表のすべてのブランク・フィールドに NULL が挿入されます。</p> <p>以下の例では、データ・ファイルにある前後のブランクを保存しながら、TABLE1 という表にデータをロードする方法を示します。</p> <pre>db2 load from delfile3 of del modified by keepblanks insert into table1</pre>

表 51. ロード・ユーティリティーで有効なファイル・タイプ修飾子: DEL ファイル・フォーマット (区切り文字で区切られている ASCII) (続き)

修飾子	説明
<b>nocharde1</b>	<p>ロード・ユーティリティーは、列区切り文字と列区切り文字の間にあるすべてのバイトが列データの一部であると見なします。文字区切り文字は、列データの一部として構文解析されます。DB2 データベース・システムを使用してエクスポートしたデータについては、このオプションを指定しないでください (ただし、エクスポート時に <b>nocharde1</b> を指定していた場合は例外です)。これは、区切り文字を持たないベンダー・データ・ファイルをサポートするために用意されています。不適切に使用すると、データが損失または破壊される場合があります。</p> <p>このオプションを <b>chardelx</b>、<b>delprioritychar</b> または <b>nodoublede1</b> と一緒に指定することはできません。これらは、相互に排他的なオプションです。</p>
<b>nodoublede1</b>	二重文字区切りの認識を抑止します。

表 52. ロード・ユーティリティーで有効なファイル・タイプ修飾子: IXF ファイル・フォーマット

修飾子	説明
<b>forcein</b>	<p>コード・ページが不一致でもデータを受け入れ、コード・ページ間の変換を抑止するようにユーティリティーに指示します。</p> <p>固定長ターゲット・フィールドに、そのデータが入るだけの十分な大きさがあるかどうかチェックされます。<b>nochecklengths</b> を指定した場合、そのような検査は実行されず、各行のロードが試みられます。</p>
<b>nochecklengths</b>	<p><b>nochecklengths</b> を指定した場合は、ソース・データの中にターゲット表の列のサイズを超える列定義がある場合であっても、各行のロードが試みられます。コード・ページ変換によってソース・データが縮小されれば、そのような行であったとしても正常にロードすることができます。例えば、ソースに 4 バイトの EUC データがある場合、それがターゲットで 2 バイトの DBCS データに縮小されれば、必要なスペースは半分で済みます。このオプションが特に役立つのは、列の定義は不一致であるがソース・データが常に適合することが分かっている場合です。</p>

注:

1. 日付形式ストリングは必ず二重引用符で囲まなければなりません。フィールド区切り文字には、a から z、A から Z、および 0 から 9 の文字を使用することはできません。フィールド区切り文字として、DEL ファイル・フォーマットの文字区切りまたはフィールド区切りと同じ文字を使用することはできません。エレメントの開始および終了位置が明らかな場合、フィールド区切り文字は任意指定です。あいまいさが生じうるのは、項目の長さが一定でない D、H、M、または S などのエレメントが使用されている場合です (修飾の仕方によって異なります)。

タイム・スタンプ・フォーマットの場合、月の記述子と分の記述子のどちらも文字 M を使用するため、区別があいまいにならないように注意する必要があります。月のフィールドは、他の日付フィールドと隣接していなければなりません。分フィールドは、他の時刻フィールドに隣接していなければなりません。あいまいなタイム・スタンプ形式の例を以下に示します。



"M" (月または分のどちらにもとれる)  
"M:M" (月と分の区別がつかない)  
"M:YYYY:M" (両方とも月と解釈される)  
"S:M:YYYY" (時刻値と日付値の両方に隣接している)

あいまいな場合、ユーティリティーはエラー・メッセージを報告し、操作は失敗します。

以下に、明確なタイム・スタンプ・フォーマットを示します。

"M:YYYY" (M (月))  
"S:M" (M (分))  
"M:YYYY:S:M" (M (月)...M (分))  
"M:H:YYYY:M:D" (M (分)...M (月))

二重引用符や円記号などの文字の前には、エスケープ文字 (例えば、¥) を付けなければなりません。

2. **charde1**、**colde1**、または **decpt** ファイル・タイプ修飾子に提供される文字値は、ソース・データのコード・ページで指定する必要があります。

文字コード・ポイント (文字記号ではない) は、xJJ または 0xJJ という構文で指定することができます (JJ はコード・ポイントの 16 進表記)。例えば、列区切りとして # 文字を指定するには、以下のステートメントのいずれかを使用します。

```
... modified by colde1# ...  
... modified by colde10x23 ...  
... modified by colde1X23 ...
```

3. 『データ移動のための区切り文字の制約事項』に、区切り文字の指定変更として使用できる文字に適用される制限のリストが示されています。
4. サポートされていないファイル・タイプを **MODIFIED BY** オプションで使用しようとしても、ロード・ユーティリティーは警告を出しません。この場合、ロード操作が失敗し、エラー・コードが戻されます。
5. **ignore**、**include**、**missing**、**override** が接尾部に付された複数の修飾子が指定されると、それらの修飾子はリストされた順に適用されます。以下のステートメントの場合、入力データに ID 列ではない暗黙的な非表示列のデータが含まれません。一方、暗黙的に非表示かどうかに関係なく、ID 列のデータは含まれません。

```
db2 load from delfile1 of del modified by  
    implicitlyhiddeninclude identitymissing insert into table1
```

しかし、以下のステートメントにおいてファイル・タイプ修飾子の順序を変更すると、入力データにすべての暗黙的な非表示列のデータ (非表示の ID 列も含む) が含まれることとなります。一方、暗黙的に非表示でない ID 列のデータは含まれません。

```
db2 load from delfile1 of del modified by  
    identitymissing implicitlyhiddeninclude insert into table1
```

表 53. codepage および usegraphiccodepage 使用時の LOAD 動作

codepage=N	usegraphiccodepage	LOAD の動作
なし	なし	<b>CLIENT</b> オプションの指定があっても、ファイル内のすべてのデータは、アプリケーション・コード・ページではなく、データベース・コード・ページであると見なされます
あり	なし	ファイル内のすべてのデータは、コード・ページ N であると見なされます。  <b>警告:</b> N が 1 バイト・コード・ページの場合、GRAPHIC データをデータベースにロードすると壊れます。
なし	あり	<b>CLIENT</b> オプションの指定があっても、ファイル内の文字データは、データベース・コード・ページであると見なされます。 <b>CLIENT</b> オプションの指定があっても、 <b>GRAPHIC</b> データは、データベース GRAPHIC データのコード・ページであると見なされます。  データベース・コード・ページが 1 バイトの場合は、すべてのデータはデータベース・コード・ページであると見なされます。  <b>警告:</b> 1 バイト・データベースに GRAPHIC データをロードすると、壊れます。
あり	あり	文字データは、コード・ページ N であると見なされま す。 GRAPHIC データは、N の GRAPHIC コード・ペ ージであると見なされます。  N が 1 バイトまたは 2 バイト・コード・ページの場合 は、すべてのデータは、コード・ページ N であると見な されます。  <b>警告:</b> N が 1 バイト・コード・ページの場合、GRAPHIC データをデータベースにロードすると壊れます。

**PRUNE HISTORY/LOGFILE コマンド (ADMIN\_CMD プロシージャを使用):**

リカバリ履歴ファイルから項目を削除したり、現在接続されているデータベ  
ース・パーティションのアクティブ・ログ・ファイル・パスからログ・ファイルを削  
除したりするのに使用します。リカバリ履歴ファイルからの項目の削除は、ファ  
イルが非常に大きくなったり保持期間が長くなっている場合に必要になることがあ  
ります。

区画に分割された環境では、**PRUNE HISTORY** コマンドは、発行されたデータベ  
ース・パーティションに関してのみ実行されます。複数のパーティションに関して履  
歴を整理するには、**PRUNE HISTORY** コマンドをそれぞれ個々のデータベース・パー  
ティションから発行するか、または db2\_a11 接頭部を使用して **PRUNE HISTORY** コ  
マンドをすべてのデータベース・パーティションに対して実行します。

**重要:** **PRUNE LOGFILE** コマンドは非推奨になっており、将来のリリースで除去され  
る可能性があります。代わりに **PRUNE HISTORY** コマンドを使用してください。

## 許可

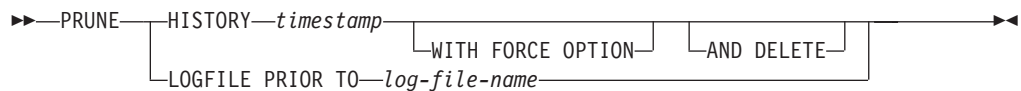
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT
- DBADM

## 必要な接続

データベース

## コマンド構文



## コマンド・パラメーター

### HISTORY *timestamp*

削除される、リカバリー履歴ファイルにある項目範囲を識別します。完全なタイム・スタンプ (書式 `yyyymmddhhmmss`)、または最初の接頭部 (最小値 `yyyy`) を指定できます。提供されているそのタイム・スタンプ以下のタイム・スタンプ付きのすべての項目は、リカバリー履歴ファイルから削除されます。最初の接頭部を指定した場合、指定されていないタイム・スタンプの構成要素は `yyyy0101000000` と解釈されます。

### WITH FORCE OPTION

最新のリストア・セットのいくつかの項目がファイルから削除されるとしても、指定したタイム・スタンプに従って項目を整理することを指定します。リストア・セットは、バックアップ・イメージのすべてのリストアを含む、最新の全データベース・バックアップです。このパラメーターを指定しない場合、バックアップ・イメージ転送からのすべての項目は履歴の中で保守されます。

### AND DELETE

履歴ファイルの項目を削除する際に、関連するログ・アーカイブを (ロケーション情報に基づいて) 物理的に削除することを指定します。このオプションは、ログ・アーカイブが不要になった場合に、アーカイブ・ストレージ・スペースがリカバリーされるようにする上で、特に有用です。ユーザー出口プログラムによりログをアーカイブしている場合は、このオプションを使用してそれらのログを削除することはできません。

**auto\_del\_rec\_obj** データベース構成パラメーターを ON に設定している場合、**AND DELETE** パラメーターを指定して **PRUNE HISTORY** を呼び出すと、バックアップ・イメージも物理的に削除され、その履歴ファイル項目が整理される場合にはコピー・イメージがロードされます。

### LOGFILE PRIOR TO *log-file-name*

ログ・ファイル名を表すストリング (例: `S0000100.LOG`) を指定します。指定したログ・ファイルより前のすべてのログ・ファイルは削除されます。指

定したログ・ファイルそのものは削除されません。 **logarchmeth1** データベース構成パラメーターは、**OFF** 以外の値に設定する必要があります。

注: この値は、DB2 pureScale 環境ではサポートされていません。

## 例

例 1 リカバリー履歴ファイルから、2003 年 12 月 31 日以前に書き込まれたすべての項目を除去します。

```
CALL SYSPROC.ADMIN_CMD ('prune history 20031231')
```

例 2: アクティブ・ログ・ファイル・パスから、S0000100.LOG より前のすべてのログ・ファイルを削除します (S0000100.LOG は削除しません)。

```
CALL SYSPROC.ADMIN_CMD('prune logfile prior to S0000100.LOG')
```

## 使用上の注意

**WITH FORCE OPTION** が使用されている場合、データベースの自動リストアに必要な項目を削除してしまう可能性があります。その場合でも手動リストアは正常に動作します。また、このコマンドを使用すると、**db2ckrst** ユーティリティーが、必要なバックアップ・イメージの完全なチェーンを正しく分析できなくなる可能性もあります。**PRUNE HISTORY** コマンドを **WITH FORCE OPTION** なしで使用した場合、必要な項目が削除されることはありません。

状況が **DB2HISTORY\_STATUS\_DO\_NOT\_DELETE** の項目は整理されません。**WITH FORCE OPTION** が使用されている場合、**DB2HISTORY\_STATUS\_DO\_NOT\_DELETE** というマークが付いたオブジェクトは、やはり整理または削除されます。**UPDATE HISTORY** コマンド、**UPDATE\_HISTORY** を指定した **ADMIN\_CMD**、または **db2HistoryUpdate** API を使用して、リカバリー履歴ファイルの項目の状況を **DB2HISTORY\_STATUS\_DO\_NOT\_DELETE** に設定できます。

**DB2HISTORY\_STATUS\_DO\_NOT\_DELETE** 状況を使用して、キー・リカバリー履歴ファイルの項目が整理されないようにしたり、関連するリカバリー・オブジェクトが削除されないようにすることができます。

**PRUNE HISTORY** コマンドを使用してスナップショット・データベース・バックアップ履歴ファイルの項目を整理できますが、**AND DELETE** パラメーターを使用して関連する物理リカバリー・オブジェクトを削除することはできません。スナップショット・バックアップ・オブジェクトを削除する唯一の方法は、**db2acsutil** コマンドを使用することです。

このコマンドは、アプリケーションが現在接続しているデータベース・パーティションだけに作用します。

## **QUIESCE DATABASE** コマンド (**ADMIN\_CMD** プロシージャーを使用):

指定したデータベースからすべてのユーザーを強制的に切断して、静止モードにします。

データベースが静止モードにある間、それに対して管理タスクを実行できます。管理タスクの完了後、**UNQUIESCE** コマンドを使用してデータベースをアクティブ化し

て、シャットダウンしたり他のデータベースの開始を実行したりすることなく、他のユーザーがデータベースに接続できるようにします。

このモードでは、この制限モード中に権限を持つユーザーだけがデータベースに接続することができます。SYSADM および DBADM 権限を持つユーザーは、データベースの静止中に常にそのデータベースにアクセスできます。

### 有効範囲

**QUIESCE DATABASE** を使用すると、データベース内のすべてのオブジェクトは静止モードに入ります。許可が与えられたユーザー/グループ、および SYSADM、SYSMAINT、DBADM、または SYSCTRL だけが、データベースまたはそのオブジェクトにアクセスできます。

データベースが SUSPEND\_WRITE 状態の場合、そのデータベースを静止モードにはできません。

### 許可

以下の権限のいずれか。

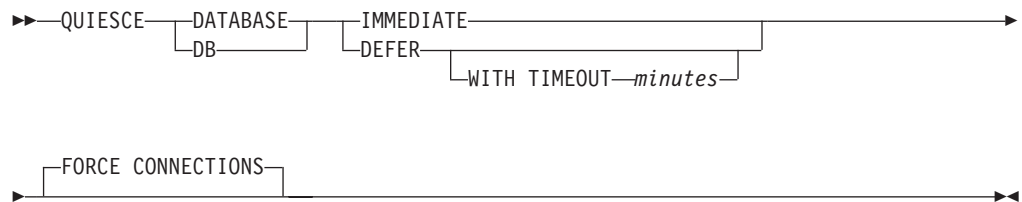
データベース・レベルの静止の場合:

- SYSADM
- DBADM

### 必要な接続

データベース

### コマンド構文



### コマンド・パラメーター

**DEFER** QUIESCE の実行をアプリケーションが現行の作業単位をコミットするまで待ちます。

#### **WITH TIMEOUT** *minutes*

アプリケーションが現在の作業単位をコミットするのを待機する時間を分単位で指定します。値を指定しない場合、単一パーティション・データベース環境では、デフォルト値が 10 分になります。パーティション・データベース環境では、データベース・マネージャ構成パラメーター **start\_stop\_time** によって指定された値が使用されます。

## IMMEDIATE

トランザクションがコミットされるのを待たず、即時にトランザクションをロールバックします。

## FORCE CONNECTIONS

接続を強制的にオフにします。

## DATABASE

データベースを静止します。データベース内のすべてのオブジェクトを静止モードにします。指定したグループのユーザーと、SYSADM、SYSMAINT、および SYSCTRL 権限を持つユーザーだけが、データベースまたはそのオブジェクトにアクセスすることができます。

## 例

データベースと接続しているすべてのユーザーを強制的に切断します。

```
CALL SYSPROC.ADMIN_CMD( 'quiesce db immediate' )
```

- このコマンドとともに **FORCE CONNECTIONS** オプションを指定すると、データベースからすべてのユーザーを強制的に切断します。 **FORCE CONNECTIONS** はデフォルトの動作です。コマンドのパラメーターは、互換性の理由により許容されています。
- コマンドは **FORCE CONNECTIONS** と同期され、**FORCE CONNECTIONS** が完了しないと完了しません。

## 使用上の注意

- **QUIESCE DATABASE** の後、SYSADM、SYSMAINT、SYSCTRL、または DBADM 権限、および GRANT または REVOKE 特権を持つユーザーは、接続可能なユーザーを指定できます。この情報は永続的にデータベース・カタログ表に保管されます。

以下に例を示します。

```
grant quiesce_connect on database to username/groupname  
revoke quiesce_connect on database from username/groupname
```

- コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。
- DB2 pureScale 環境では、データベースの静止およびインスタンスの再始動の後、データベースはすべてのメンバーにわたり静止状態のままになります。静止状態を解除するには、明示的な **UNQUIESCE DATABASE** コマンド呼び出しが必要です。

## QUIESCE TABLESPACES FOR TABLE コマンド (ADMIN\_CMD プロシージャーを使用):

特定の表の表スペースを静止させます。有効な静止モードは、共有、更新意図、および排他の 3 つです。

静止機能の結果として生じる状態には、次の 3 つの状態があります。

- 静止: SHARE
- 静止: UPDATE
- 静止: EXCLUSIVE

## 有効範囲

単一パーティション環境では、ロード操作中に排他モードのロード操作を起動すると、このコマンドは表スペースをすべて静止します。パーティション・データベース環境では、このコマンドはデータベース・パーティションでローカルに活動します。このコマンドは、ロード操作を実行しているデータベース・パーティションに属する表スペースの部分のみを静止します。パーティション表の場合、表に関連付けられている `SYSDATAPARTITIONS.TBSPACEID` および `SYSDATAPARTITIONS.LONG_TBSPACEID` の中のリストに含まれているすべての表スペースのうち、状況が「正常」、「アタッチ」、または「デタッチ」であるもの (例えば `SYSDATAPARTITIONS.STATUS` が `''`、`'A'`、または `'D'`) が静止されます。

## 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT
- DBADM
- LOAD

## 必要な接続

データベース

## コマンド構文

```
▶▶—QUIESCE TABLESPACES FOR TABLE tablename [ schema.tablename ] { SHARE | INTENT TO UPDATE | EXCLUSIVE | RESET }▶▶
```

## コマンド・パラメーター

### TABLE

*tablename*

非修飾表名を指定します。システム・カタログ表を指定することはできません。

*schema.tablename*

修飾表名を指定します。 *schema* が指定されない場合には、CURRENT SCHEMA が使用されます。システム・カタログ表を指定することはできません。

**SHARE** 静止が共有モードであることを指定します。

「静止モードでの共有」要求を行うと、トランザクションは、表スペースに対して意図的共有ロックを、および表に対して共有ロックを要求します。トランザクションがロックを獲得すると、表スペースの状態が QUIESCED SHARE に変更されます。この状態は、他のユーザーがこれと矛盾する状態

を保持していない場合に限り、静止者に付与されます。表スペースの状態は、その状態が持続されるように、許可 ID およびそのユーザーのデータベース・エージェント ID とともに、表スペースにある表に記録されます。ある表の表スペースが QUIESCED SHARE 状態である間は、その表を変更できません。表および表スペースに要求するその他の共有モードは、認められません。トランザクションがコミットまたはロールバックされる際、ロックは解除されますが、その表の表スペースはその状態が明示的にリセットされるまで、QUIESCED SHARE 状態のまま残ります。

#### INTENT TO UPDATE

静止モードが更新意図モードであることを指定します。

「静止モードでの更新意図」要求を行うと、表スペースは意図的排他 (IX) モードでロックされ、表は更新 (U) モードでロックされます。表スペースの状態は、表スペースの表に記録されます。

#### EXCLUSIVE

静止が排他モードであることを指定します。

「静止モードでの排他」要求を行うと、トランザクションは、表スペースに対する特別な排他ロックと、表に対する特別な排他ロックを要求します。トランザクションがロックを獲得すると、表スペースの状態が QUIESCED EXCLUSIVE に変更されます。表スペースの状態は、静止者の許可 ID およびデータベース・エージェント ID とともに、表スペース表に記録されます。表スペースは、スーパー排他モードで保留されているため、表スペースへの他のアクセスは認められません。静止プログラム機能呼び出すユーザー (静止プログラム) は、その表と表スペースへの排他的アクセスを行うことができます。

**RESET** 表スペースの状態が、「正常」にリセットされることを指定します。静止要求を発行した接続がまだアクティブである場合、静止状態をリセットすることはできません。

静止者がリセットを発行すると、その静止者に関する静止モードのみリセットされます。静止者が複数存在する場合は、表スペースの状態が変化していないように見えます。

システム期間テンポラル表および関連した履歴表を操作している場合、リセット操作は、最初に静止モードを設定したときに使用したのと同じ表に対して実行する必要があります。

#### 例

社員 (staff) 表が含まれている表スペースを静止します。

```
CALL SYSPROC.ADMIN_CMD( 'quiesce tablespaces for table staff share' )
```

#### 使用上の注意

このコマンドは、DB2 pureScale 環境ではサポートされていません。

静止は持続ロックです。その利点は、それがトランザクション障害、接続障害、およびシステム障害 (電源障害や、リブートなど) が生じても持続することです。



静止は接続によって所有されます。接続が失われた場合、静止は残りますが、非所有の状態に移り、ファントム静止 と呼ばれます。例えば、ロード操作の削除フェーズ中に停電が発生して中断すると、ロード中の表の表スペースは静止排他状態のままになります。データベースの再始動時に、この静止は非所有 (ファントム) の状態になります。ファントム静止を取り除くには、静止モードが設定されたときに使用されたのと同じユーザー ID による接続が必要です。

ファントム静止を取り除くには、次のようにします。

1. 静止モードが設定されたときに使用されたのと同じユーザー ID でデータベースと接続する。
2. **LIST TABLESPACES** コマンドを使用して、静止させる表スペースを決定する。
3. 現行の静止状態を使用して、表スペースを再静止させる。以下に例を示します。

```
CALL SYSPROC.ADMIN_CMD('quiesce tablespaces for table mytable exclusive' )
```

完了すると、新しい接続が静止を所有するようになり、ロード操作を再開できるようになります。

1 つの表スペースに対する静止者の限度は、常に 5 人です。

静止プログラムは表スペースの状態を、あまり制限的でない状態から、より制限的な状態へ (例えば、S から U へ、または U から X へ) 変更できます。しかし、ユーザーが既に保持している状態より低い状態を要求しても、元の状態が戻されません。つまり、状態がダウングレードされることはありません。

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

システム期間テンポラル表に対する静止操作を実行すると、そのシステム期間テンポラル表および履歴表に関連付けられたすべての表スペースが静止されます。履歴表に対する静止操作を実行すると、その履歴表に関連付けられたすべての表スペースと、関連付けられたシステム期間テンポラル表が静止されます。

#### **REDISTRIBUTE DATABASE PARTITION GROUP コマンド (ADMIN\_CMD プロシージャを使用):**

データベース・パーティション・グループ内のパーティション間でデータを再配分します。このコマンドはデータベース・パーティション・グループにあるすべてのオブジェクトに影響を及ぼし、1 つのオブジェクトだけに限定することはできません。

#### **有効範囲**

このコマンドは、データベース・パーティション・グループ内のすべてのデータベース・パーティションに影響を与えます。

#### **許可**

以下のいずれかの権限が必要です。

- SYSADM
- SYSCTRL

• DBADM

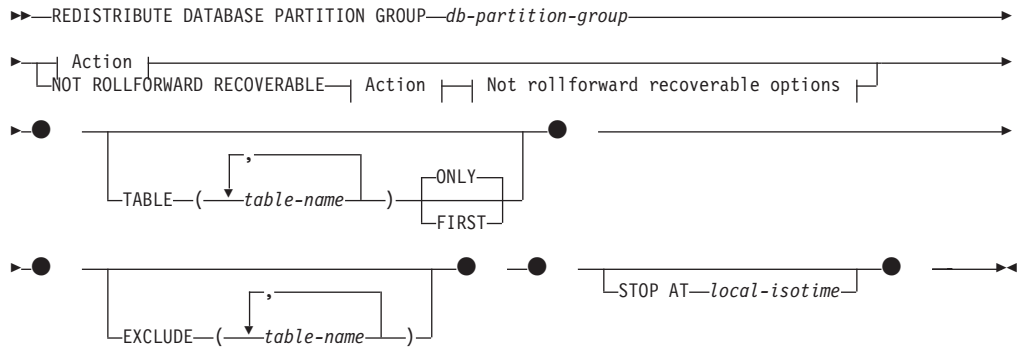
さらに、以下の権限のグループのいずれかも必要です。

- 再配分されるデータベース・パーティション・グループ内のすべての表に対する DELETE、INSERT、および SELECT 特権。
- DATAACCESS 権限

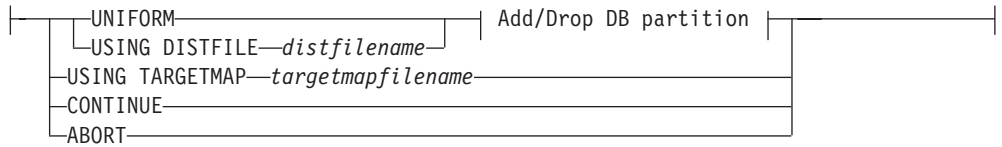
必要な接続

カタログ・パーティションとの接続。

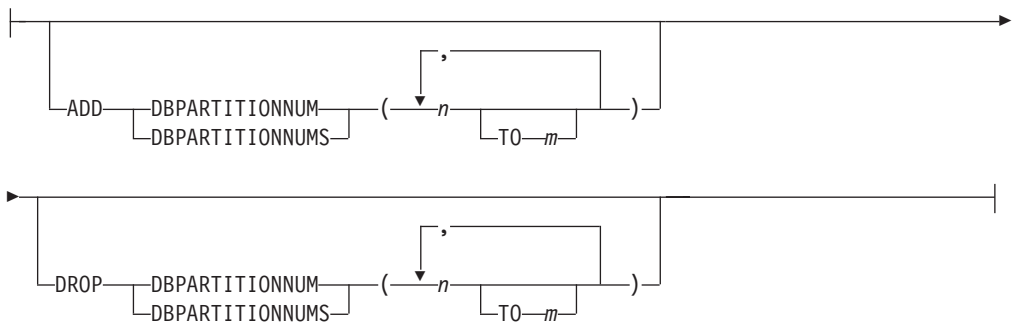
コマンド構文



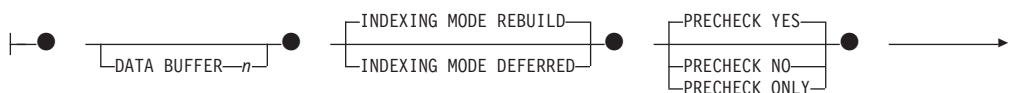
Action:

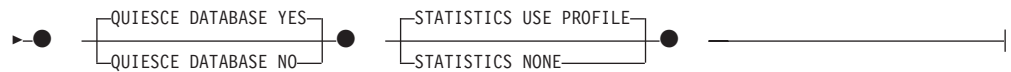


Add/Drop DB partition:



Not rollforward recoverable options:





## コマンド・パラメーター

### DATABASE PARTITION GROUP *db-partition-group*

データベース・パーティション・グループの名前。この 1 部構成の名前は、SYSCAT.DBPARTITIONGROUPS カタログ表に記述されたデータベース・パーティション・グループを識別します。データベース・パーティション・グループは、現在再配分を受けることはできません。

注: IBMCATGROUP および IBMTEMPGROUP データベース・パーティション・グループ内の表を再配分することはできません。

### NOT ROLLFORWARD RECOVERABLE

このオプションを使用すると、**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドはロールフォワード・リカバリー可能ではありません。

- データは、内部での挿入および削除操作によってではなく、一括して移動されます。これにより、表のスキャンおよびアクセスの回数が減り、パフォーマンスが向上します。
- 挿入および削除操作それぞれに対するログ・レコードは必要ではなくなりました。このため、データの再配分を実行するときに、システム内で大容量のアクティブ・ログ・スペースおよびログ・アーカイブ・スペースを管理する必要がなくなりました。
- **REDISTRIBUTE DATABASE PARTITION GROUP** コマンドを **NOT ROLLFORWARD RECOVERABLE** オプションとともに使用する場合、再配分操作は XML 列の入った表に対して **INDEXING MODE DEFERRED** オプションを使用します。表に XML 列が含まれていない場合、再配分操作はコマンドの発行時に指定された索引付けモードを使用します。

このオプションが使用されない 場合には、すべての行移動に関する詳細なロギングが実行されるため、中断やエラーが起きた場合、またはほかにビジネス上の必要が生じた場合に、データベースを後からリカバリーすることができます。

### UNIFORM

データがハッシュ・パーティション間で均等に配分されることを指定します (つまり、それぞれのハッシュ・パーティションが同じ数の行を持つことが想定されます)。しかし、それぞれのデータベース・パーティションに同じ数のハッシュ・パーティションはマップされません。再配分後、データベース・パーティション・グループのすべてのデータベース・パーティションは、ほぼ同じ数のハッシュ・パーティションを持っています。

### USING DISTFILE *distfilename*

分散キー値の分散に偏りがある場合、このオプションを使用して、データベース・パーティション・グループのデータベース・パーティション全体にわたるデータの均一な再分散を行います。

*distfilename* を使用して、32 768 個のハッシュ・パーティションにわたる現行のデータの配分を指示します。

行カウント、バイト・ボリューム、または他の任意の尺度を使用して、各ハッシュ・パーティションで表示されたデータ量を示します。ユーティリティーは、パ

パーティションに関連する整数値をそのパーティションの重みとして読み取ります。 *distfilename* を指定した場合、ユーティリティーはターゲット分散マップを生成します。このマップは、データベース・パーティション・グループのデータベース・パーティション全体においてデータをできるだけ均一に再配分するために使用されます。再配分した後は、データベース・パーティション・グループ内の各データベース・パーティションの重みが、ほぼ同じになります (データベース・パーティションの重みは、そのデータベース・パーティションにマップするすべてのハッシュ・パーティションの重みの合計です)。

例えば、入力配布ファイルに以下の項目があるとします。

```
10223
1345
112000
0
100
...
```

例の中で、ハッシュ・パーティション 2 は 112000 の重みを持ち、パーティション 3 (重さは 0) には、マッピングするデータがまったくありません。

*distfilename* には、32 768 の正整数値が文字形式で入っていなければなりません。値の合計は、4 294 967 295 以下である必要があります。

*distfilename* は、その完全なパス名を含めることが必要であり、また *distfilename* はサーバー上に存在していなければならず、接続されているパーティションからアクセス可能でなければなりません。

#### **USING TARGETMAP *targetmapfilename***

*targetmapfilename* で指定されたファイルは、ターゲット分散マップとして使用されます。データの再配分はこのファイルに従って行われます。

*targetmapfilename* は、その完全なパス名を含めることが必要であり、また *targetmapfilename* はサーバー上に存在していなければならず、接続されているパーティションからアクセス可能でなければなりません。

*targetmapfilename* には 32 768 個の整数を格納する必要があります、それぞれは有効なデータベース・パーティション番号を表します。各行の番号によって、ハッシュ値が特定のデータベース・パーティションにマップされます。つまり、行 *X* に値 *Y* が入っている場合、HASHEDVALUE() が *X* のすべてのレコードはデータベース・パーティション *Y* に配置されます。

ターゲット・マップに含まれるデータベース・パーティションがデータベース・パーティション・グループ中に存在しないと、エラーが戻されます。

**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドを実行する前に、ALTER DATABASE PARTITION GROUP ADD DBPARTITIONNUM ステートメントを実行してください。

ターゲット・マップから除外されたデータベース・パーティションが、データベース・パーティション・グループにある場合、そのデータベース・パーティションはパーティションの中に含まれていません。このようなデータベース・パーティションは、**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドの前か後に ALTER DATABASE PARTITION GROUP DROP DBPARTITIONNUM ステートメントを使用することによってドロップできます。

## CONTINUE

直前に失敗または停止した **REDISTRIBUTE DATABASE PARTITION GROUP** 操作を継続します。何も起こらなければ、エラーが戻されます。

## ABORT

直前に失敗または停止した **REDISTRIBUTE DATABASE PARTITION GROUP** 操作をアボートします。何も起こらなければ、エラーが戻されます。

## ADD

### DBPARTITIONNUM *n*

#### TO *m*

*n* または *n* TO *m* では、データベース・パーティション・グループに追加するデータベース・パーティション番号のリストを指定します。指定するパーティションは、データベース・パーティション・グループにすでに定義済みであってはなりません (SQLSTATE 42728)。ADD DBPARTITIONNUM 節を指定して ALTER DATABASE PARTITION GROUP ステートメントを実行する操作と等価です。

### DBPARTITIONNUMS *n*

#### TO *m*

*n* または *n* TO *m* では、データベース・パーティション・グループに追加するデータベース・パーティション番号のリストを指定します。指定するパーティションは、データベース・パーティション・グループにすでに定義済みであってはなりません (SQLSTATE 42728)。ADD DBPARTITIONNUM 節を指定して ALTER DATABASE PARTITION GROUP ステートメントを実行する操作と等価です。

## 注:

1. このオプションを使用してデータベース・パーティションを追加すると、表スペースのコンテナは、データベース・パーティション・グループ内で最も小さい番号を持つ既存のパーティション内の、対応する表スペースのコンテナに基づくこととなります。その結果、コンテナの名前の競合が発生する場合は、このオプションを使用しないでください (新しいパーティションが既存のコンテナと同じ物理マシンにあると、そのような名前の競合が発生する可能性があります)。そのような場合は、WITHOUT TABLESPACES オプションを指定して ALTER DATABASE PARTITION GROUP ステートメントを使用してから、**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドを実行してください。その後、適切な名前を指定して、表スペース・コンテナを手動で作成できます。
2. **ADD DBPARTITIONNUMS** パラメーターを指定すると、データ再配分によって、すべての新規データベース・パーティションに表スペースが作成される可能性があります。

## DROP

### DBPARTITIONNUM *n*

#### TO *m*

*n* または *n TO m* では、データベース・パーティション・グループからドロップするデータベース・パーティション番号のリストを指定します。指定するパーティションは、データベース・パーティション・グループにすでに定義されている必要があります (SQLSTATE 42729)。DROP DBPARTITIONNUM 節を指定して ALTER DATABASE PARTITION GROUP ステートメントを実行する操作と等価です。

#### DBPARTITIONNUMS *n*

**TO** *m*

*n* または *n TO m* では、データベース・パーティション・グループからドロップするデータベース・パーティション番号のリストを指定します。指定するパーティションは、データベース・パーティション・グループにすでに定義されている必要があります (SQLSTATE 42729)。DROP DBPARTITIONNUM 節を指定して ALTER DATABASE PARTITION GROUP ステートメントを実行する操作と等価です。

#### TABLE *tablename*

再配分処理する表の順番を指定します。

#### ONLY

表の順序の後に **ONLY** キーワード (デフォルト) を使用すると、指定した表だけが再配分の対象になります。残りの表は、**REDISTRIBUTE CONTINUE** コマンドによって後で処理できます。これはデフォルトです。

#### FIRST

表の順序の後に **FIRST** キーワードを使用すると、指定した表が指定の順序で再配分処理を受け、データベース・パーティション・グループ内の残りの表は、ランダムな順序で再配分処理を受けることになります。

#### EXCLUDE *tablename*

再配分処理をしない表を指定します。例えば、データ再配分の要件を表が満たすように構成できるようになるまで、その表を一時的に対象外にできます。対象から外した表は、**REDISTRIBUTE CONTINUE** コマンドを使用して後で処理できます。

#### STOP AT *local-isotime*

このオプションを指定すると、各表のデータ再配分を開始する前に、*local-isotime* と現在のローカル・タイム・スタンプが比較されます。指定した *local-isotime* が現在のローカル・タイム・スタンプと同じか、それよりも早いと、ユーティリティーは処理を停止して、警告メッセージを生成します。停止時に進行中であった表のデータ再配分の処理は中断されずに完了します。表の新規のデータ再配分の処理は開始されません。未処理の表の再配分を実行するには、**CONTINUE** オプションを使用します。この *local-isotime* 値は、日付と時刻の組み合わせを識別する 7 部構成の文字ストリングのタイム・スタンプとして指定します。形式は、現地時間の *yyyy-mm-dd-hh.mm.ss.nnnnnn* (年、月、日、時、分、秒、マイクロ秒) です。

#### DATA BUFFER *n*

ユーティリティー内でデータを転送するためのバッファ・スペースとして使用する 4 KB ページの数を指定します。このコマンド・パラメーターは、**NOT ROLLFORWARD RECOVERABLE** パラメーターも指定された場合のみ使用できます。

指定された値がサポートされている最小値よりも小さい場合には、最小値が使用され、警告は戻されません。 **DATA BUFFER** 値を指定しないと、実行時に各表の処理を開始する時点で、ユーティリティーによって適切なデフォルトが計算されます。具体的には、表の再配分の開始時点でユーティリティー・ヒープで使用可能になっているメモリーの 50% を使用することを基本にしながら、さまざまな表プロパティーを考慮に入れることによって、デフォルトを計算することになります。

このメモリーは、ユーティリティー・ヒープから直接に割り当てられ、そのサイズは **util\_heap\_sz** データベース構成パラメーターで修正可能です。システムにさらに使用可能なメモリーがある場合、**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドの **DATA BUFFER** パラメーターの値は、一時的に **util\_heap\_sz** を超える場合があります。

#### **INDEXING MODE**

再配分時の索引の保守方法を指定します。このコマンド・パラメーターは、**NOT ROLLFORWARD RECOVERABLE** パラメーターも指定された場合のみ使用できます。

有効な値は以下のとおりです。

#### **REBUILD**

索引が最初から再作成されます。このオプションを使用する場合は、索引が有効である必要はありません。このオプションを使用する結果として、「索引」ページがディスク上で一緒にクラスタ化されます。

#### **DEFERRED**

再配分で索引の維持を試行しません。リフレッシュが必要であることを示すマークが索引に付けられます。そのような索引に最初にアクセスした時点で再作成が強制実行されるか、データベースの再始動時に索引が再作成されることとなります。

注: 非 MDC 表および非 ITC 表の場合、表に無効な索引があると、

**INDEXING MODE DEFERRED** が指定されていないければ、**REDISTRIBUTE DATABASE PARTITION GROUP** コマンドによって自動的に索引が再作成されます。MDC 表または ITC 表の場合、**INDEXING MODE DEFERRED** が指定されている場合でも、無効な複合索引は表の再配分が始まる前に再作成されます。なぜなら、ユーティリティーは MDC 表または ITC 表を処理するために複合索引を必要とするからです。

#### **PRECHECK**

データベース・パーティション・グループを再配分できるかどうかを検査します。このコマンド・パラメーターは、**NOT ROLLFORWARD RECOVERABLE** パラメーターも指定された場合のみ使用できます。

#### **YES**

これはデフォルト値です。再配分操作が開始されるのは、検査が正常に終了した場合のみです。検査に失敗すると、このコマンドは終了して、失敗した最初の検査に関連したエラー・メッセージを返します。

**NO** 再配分操作はすぐに開始されます。検査は行われません。

#### **ONLY**

検査実行後に、コマンドは終了します。再配分は行われません。デフォルトでは、データベースは静止されません。**QUIESCE DATABASE** コマンド・パラ

メーターが YES に設定された場合、またはデフォルト値の YES の場合には、データベースは静止状態のままになります。データベースへの接続を復元するには、再配分操作を実行するか、**UNQUIESCE DATABASE** コマンドを発行します。

#### QUIESCE DATABASE

データベースからすべてのユーザーを強制的に切断して静止モードにするように指定します。このコマンド・パラメーターは、**NOT ROLLFORWARD RECOVERABLE** パラメーターも指定された場合のみ使用できます。

#### YES

これはデフォルト値です。SYSADM、SYSMAINT、または SYSCTRL の権限を持つユーザーと、QUIESCE\_CONNECT 権限を付与されたユーザーだけが、データベースとそのオブジェクトにアクセスできるようになります。再配分が正常に完了すると、データベースは静止解除されます。

**NO** 再配分操作によって、データベースが静止状態になることはありません。ユーザーがデータベースから強制的に切断されることもありません。

詳しくは、**QUIESCE DATABASE** コマンドを参照してください。

#### STATISTICS

ユーティリティが、統計プロファイルのある表の統計を収集するように指定します。このコマンド・パラメーターは、**NOT ROLLFORWARD RECOVERABLE** パラメーターも指定された場合のみ使用できます。

このオプションを指定するほうが、データ再配分の完了後に **RUNSTATS** コマンドを別途実行するよりも効率的です。

#### USE PROFILE

統計プロファイルのある表の統計を収集します。統計プロファイルのない表については、何も実行されません。これはデフォルトです。

#### NONE

表の統計を収集しません。

#### 例

データ分散ファイル `distfile_for_dbpg_1` によって現在のデータ分散を提供することにより、データベース・パーティション・グループ `DBPG_1` を再配分します。データを 2 つの新しいデータベース・パーティション 6 および 7 に移動します。

```
CALL SYSPROC.ADMIN_CMD('REDISTRIBUTE DATABASE PARTITION GROUP DBPG_1
  USING DISTFILE /home/user1/data/distfile_for_dbpg_1
  ADD DATABASE PARTITION (6 TO 7)')
```

データベース・パーティション・グループ `DBPG_2` に対して以下のような再配分を行います。

- 再配分はロールフォワード・リカバリー可能ではない。
- データは複数のハッシュ・パーティションにわたって一様に配分される。
- 索引は最初から再作成される。
- 統計は収集されない。
- データ転送のためのバッファ・スペースとして、180,000 個の 4 KB ページが使用される。



```
CALL SYSPROC.ADMIN_CMD('REDISTRIBUTE DATABASE PARTITION GROUP DBPG_2
NOT ROLLFORWARD RECOVERABLE
UNIFORM
INDEXING MODE REBUILD
DATA BUFFER 180000
STATISTICS NONE')
```

この再配分操作では、**QUIESCE DATABASE** および **PRECHECK** コマンド・パラメーターのデフォルト値により、データベースの静止および事前検査も実行されます。

### 使用上の注意

- 再配分操作を開始する前に、表が通常状態にあり、「ロード・ペンディング」状態でも「REORG ペンディング」状態でもないことを確認してください。表状態は **LOAD QUERY** コマンドを使って調べることができます。
- **NOT ROLLFORWARD RECOVERABLE** オプションが指定され、データベースがリカバリー可能である場合、ユーティリティが最初に表スペースにアクセスした時点で、表スペースは **BACKUP PENDING** 状態になります。表スペースのバックアップが作成されるまで、その表スペースに含まれているすべての表は読み取り専用になります。表スペースのバックアップは、その表スペース内のすべての表の再配分処理が完了したときにのみ可能になります。
- 再配分操作の実行中に、その再配分操作に関する一般情報、および各表の処理開始時刻と処理終了時刻などの情報を含むイベント・ログ・ファイルが作成されます。このイベント・ログ・ファイルは、以下のようにサーバーに書き込まれます。
  - Linux および UNIX オペレーティング・システムの場合は、  
homeinst/sqllib/redis ディレクトリー。サブディレクトリーとファイル名の形式は、*database-name.database-partition-group-name.timestamp.log* になります。
  - Windows オペレーティング・システムの場合は、  
**DB2INSTPROF**¥instance¥redis ディレクトリー (**DB2INSTPROF** は、**DB2INSTPROF** レジストリー変数の値です)。サブディレクトリーとファイル名の形式は、  
*database-name.database-partition-group-name.timestamp.log* になります。
  - タイム・スタンプ値は、コマンドが発行された時の時刻です。
- このユーティリティは、処理中に断続的な **COMMIT** を実行します。そのため、タイプ 2 接続では **SQL30090N** エラーを受け取ることになります。
- 再配分を受けた表と従属関係があるすべてのパッケージは無効になります。データベース・パーティション・グループの再配分操作が完了した後で、そのようなパッケージを明示的に再バインドすることをお勧めします。明示的な再バインドにより、無効パッケージに対する最初の **SQL** 要求の実行での初期遅延がなくなります。再配分メッセージ・ファイルには、再配分を受けたすべての表のリストが入ります。
- 統計プロファイルがある表については、再配分ユーティリティの実行時に、デフォルトで統計が更新されます。統計プロファイルがない表の場合は、表や索引の統計を別途更新することをお勧めします。そのためには、再配分操作の完了後に、db2Runstats API を呼び出すか、**RUNSTATS** コマンドを実行できます。
- 複製されたマテリアライズ照会表や **DATA CAPTURE CHANGES** を用いて定義された表を含むデータベース・パーティション・グループは、再配分することができません。

- データベース・パーティション・グループに、既存の宣言済み一時表または作成済み一時表を含む **USER TEMPORARY** 表スペースがある場合、再配分を行うことはできません。
- **INDEXING MODE** などのオプションは、適用対象にならない表では無視されます (警告も生成されません)。例えば、**INDEXING MODE** は、索引のない表では無視されます。
- コマンドの実行状況は、**CALL** ステートメントからの結果である **SQLCA** で戻されます。
- **USING DISTFILE** *distfilename* または **USING TARGETMAP** *targetmapfilename* で参照するファイルでは、サーバー上のファイルを参照する必要があります。
- **REDISTRIBUTE DATABASE PARTITION GROUP** コマンドは、データベース・パーティションの追加サーバー要求が保留中または進行中の場合は、失敗する可能性があります (**SQLSTATE 55071**)。さらに、新規データベース・パーティション・サーバーがオンラインでインスタンスに追加された場合、およびすべてのアプリケーションが新規データベース・パーティション・サーバーを認識しているわけではない場合、このコマンドは失敗する可能性があります (**SQLSTATE 55077**)。

### 互換性

DB2 バージョン 9.5 以前の XML レコード・フォーマットを使用する XML 列が入った表は、再配分できません。そうした表を新しいフォーマットに移行するには、**ADMIN\_MOVE\_TABLE** ストアド・プロシージャを使用します。

### **REORG INDEXES/TABLE** コマンド (**ADMIN\_CMD** プロシージャを使用):

索引または表を再編成します。

フラグメント化されていない物理的に連続したページに索引データを再構築することによって、表に定義されたすべての索引を再編成することができます。データ・パーティション表では、パーティション表で特定の非パーティション索引を再編成すること、もしくは特定のデータ・パーティションですべてのパーティション索引を再編成することができます。

索引節の **CLEANUP** オプションを指定すると、索引を再作成しないでクリーンアップが実行されます。このコマンドを宣言済み一時表または作成済み一時表の索引に対して使用することはできません (**SQLSTATE 42995**)。

表オプションは、フラグメント化されたデータを消去するために行を再作成、および情報を縮小化することによって、表を再編成します。パーティション表では、単一パーティションを再編成できます。

### 有効範囲

このコマンドは、データベース・パーティション・グループ内のすべてのデータベース・パーティションに影響を与えます。

### 許可

以下の権限のいずれか。

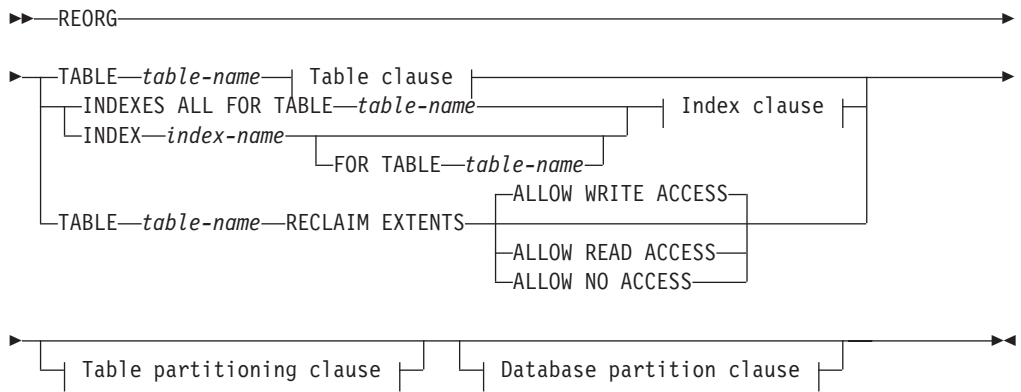
- **SYSADM**

- SYSCTRL
- SYSMANT
- DBADM
- SQLADM
- 表に対する CONTROL 特権

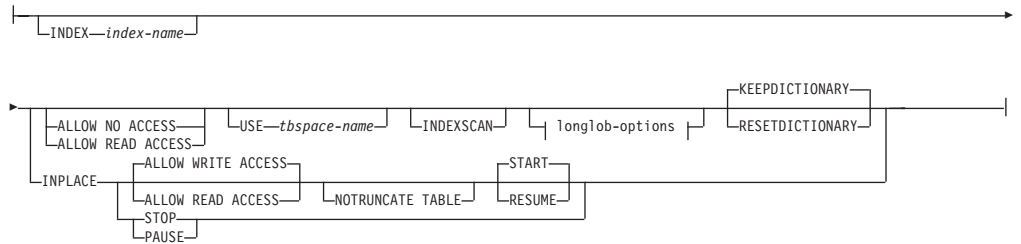
**必要な接続**

データベース

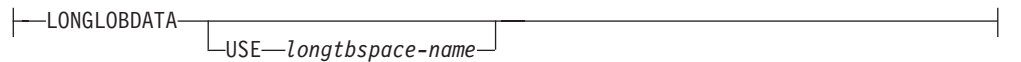
**コマンド構文**



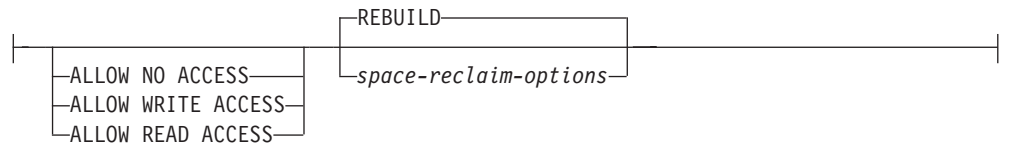
**Table clause:**



**longlob-options:**



**Index clause:**



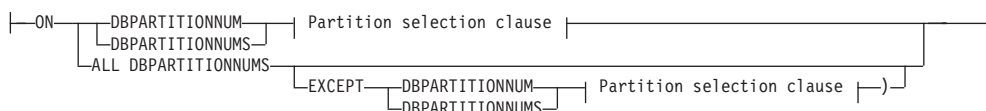
### space-reclaim-options:



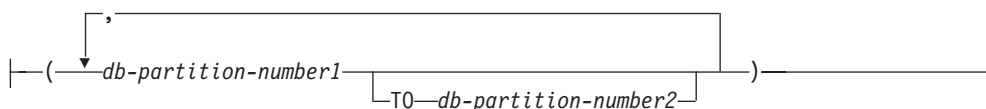
### Table partitioning clause:



### Database partition clause:



### Partition selection clause:



### コマンド・パラメーター

#### INDEXES ALL FOR TABLE *table-name*

索引を再編成する表を指定します。表は、ローカルまたはリモート・データベースにあるものです。

#### INDEX *index-name*

データ・パーティション表上の再編成する個別索引を指定します。個別索引の再編成は、パーティション表上の非パーティション索引についてのみサポートされています。このパラメーターは、ブロック索引に関してはサポートされていません。

#### FOR TABLE *table-name*

非パーティション索引 *index-name* が作成される表の名前を指定します。索引名がデータベースを通じて固有であるなら、このパラメーターはオプションです。

#### ALLOW NO ACCESS

**REORG INDEXES** の場合、索引が再編成される間に、他のユーザーがその表にアクセスできないことを指定します。パーティション表に対して **ON DATA PARTITION** 節が指定されている場合、その指定されたパーティションのみがこのアクセス・モード・レベルに制限されます。

**REORG INDEX** の場合、非パーティション索引が再編成される間に、他のユーザーがその表にアクセスできないことを指定します。

## ALLOW READ ACCESS

**REORG INDEXES** の場合、索引が再編成される間に、他のユーザーがその表に対して、読み取り専用でアクセスできることを指定します。**CLEANUP** オプションまたは **RECLAIM EXTENTS** オプション、あるいは **ON DATA PARTITION** 節が指定されていない限り、パーティション表の **REORG INDEXES** に対して **ALLOW READ ACCESS** モードはサポートされません。パーティション表に対して **ON DATA PARTITION** 節が指定されている場合、その指定されたパーティションのみがこのアクセス・モード・レベルに制限されます。

**REORG INDEX** の場合、非パーティション索引が再編成される間に、その表に読み取り専用でアクセスできることを指定します。

## ALLOW WRITE ACCESS

**REORG INDEXES** の場合、索引が再編成される間に、他のユーザーがその表から読み取ったりそこに書き込んだりできることを指定します。**CLEANUP** オプションまたは **RECLAIM EXTENTS** オプション、あるいは **ON DATA PARTITION** 節が指定されていない限り、パーティション表に対して **ALLOW WRITE ACCESS** モードはサポートされません。パーティション表に対して **ON DATA PARTITION** 節が指定されている場合、その指定されたパーティションのみがこのアクセス・モード・レベルに制限されます。

**REORG INDEX** の場合、非パーティション索引が再編成される間に、その表から読み取ったりそこに書き込んだりできることを指定します。

**CLEANUP** オプションまたは **RECLAIM EXTENTS** オプションが指定されている場合を除き、**ALLOW WRITE ACCESS** モードは、マルチディメンション・クラスタリング (MDC) 表、挿入時クラスタリング (ITC) 表、および拡張索引ではサポートされません。

**ON DATA PARTITION** 節が **REORG INDEXES ALL** コマンドと共に指定されている場合、以下の項目がデータ・パーティション表に適用されます。

- 指定されたデータ・パーティションのみがこのアクセス・モード・レベルに制限されます。指定されたパーティションのパーティション索引が再編成される間に、ユーザーは表の他のパーティションから読み取ったりそこに書き込んだりできます。

以下の表に、**ON DATA PARTITION** 節が指定された時にサポートされるアクセス・モードおよび表の他のパーティションで許可される並行アクセスを示します。

表 54. **REORG INDEXES ALL** と共に **ON DATA PARTITION** 節が指定された時にサポートされるアクセス・モードおよび許可される並行アクセス

アクセス・モード	指定されたパーティションで許可される並行アクセス	他のパーティションで許可される並行アクセス
<b>ALLOW NO ACCESS</b>	アクセスできません	読み取りおよび書き込みアクセス
<b>ALLOW READ ACCESS</b>	索引が更新されるまでのパーティションにおける読み取り	読み取りおよび書き込みアクセス

表 54. REORG INDEXES ALL と共に ON DATA PARTITION 節が指定された時にサポートされるアクセス・モードおよび許可される並行アクセス (続き)

アクセス・モード	指定されたパーティションで許可される並行アクセス	他のパーティションで許可される並行アクセス
ALLOW WRITE ACCESS	索引が更新されるまでのパーティションにおける読み取りおよび書き込みアクセス	読み取りおよび書き込みアクセス

- 指定されたパーティションのパーティション索引のみが再編成されます。非パーティション索引は再編成されません。

「無効」または「再作成」とマークされた表に非パーティション索引がある場合、これらの索引は再編成前に再作成されます。それ以外の場合、索引オブジェクトが「無効」または「再作成」とマークされているなら、指定されたパーティションのパーティション索引のみが再編成されるか、または再作成されます。

- CLEANUP** オプションまたは **RECLAIM EXTENTS** オプションも指定されている場合、指定されたパーティションのパーティション索引のみがクリーンにされます。

以下の表は、パーティションおよび非パーティション表における索引の再編成に対してサポートされるアクセス・モードを示します。

表 55. パーティションおよび非パーティション表における索引の再編成に対してサポートされるアクセス・モード

コマンド	表タイプ	表パーティション節	index (索引) 節に指定されたその他のパラメーター	サポートされるアクセス・モード
REORG INDEXES	非パーティション表	該当なし	任意	ALLOW NO ACCESS, ALLOW READ ACCESS <sup>1</sup> , ALLOW WRITE ACCESS
REORG INDEX	パーティション表	該当なし	任意	ALLOW NO ACCESS, ALLOW READ ACCESS <sup>1</sup> , ALLOW WRITE ACCESS
REORG INDEXES	パーティション表	なし	REBUILD (何も指定されていない場合のデフォルト)	ALLOW NO ACCESS <sup>1</sup>
REORG INDEXES	パーティション表	ON DATA PARTITION	REBUILD (何も指定されていない場合のデフォルト)	ALLOW NO ACCESS, ALLOW READ ACCESS <sup>1</sup> , ALLOW WRITE ACCESS
REORG INDEXES	パーティション表	ON DATA PARTITION 節ありまたはなし	CLEANUP または RECLAIM EXTENTS を指定	ALLOW NO ACCESS, ALLOW READ ACCESS <sup>1</sup> , ALLOW WRITE ACCESS

注:

1. アクセス節が指定されていない場合のデフォルト・モード。

## CLEANUP

**CLEANUP** が要求されると、**REBUILD** ではなくクリーンアップが実行されます。索引は再作成されません。解放されたページはこの表に定義された索引だけが再使用できます。

**ALL** コミット済み疑似削除済みキーおよびコミット済み疑似空白キーを除去することにより、索引をクリーンアップすることを指定します。

**CLEANUP ALL** オプションはコミット済み疑似空白ページを解放して、コミット済み疑似削除済みキーを疑似空白ではないページから除去します。このオプションは、隣接する複数のリーフ・ページをマージすると少なくとも **PCTFREE** のフリー・スペースを持つマージ済みリーフ・ページが生じる場合に、そのマージを試行します。**PCTFREE** は、索引作成時に索引に定義されたフリー・スペースのパーセントです。デフォルトの **PCTFREE** は 10% です。2つのページがマージ可能な場合、そのうちの1つのページが解放されます。疑似空白ページにあるものを除く、索引内の疑似削除済みキーの数は、**RUNSTATS** を実行してから **NUMRIDS DELETED** を **SYSCAT.INDEXES** から選択することによって判別できます。**ALL** オプションは、コミット済みと判別された場合に **NUMRIDS DELETED** および **NUM EMPTY LEAFS** を消去します。

**PAGES** コミット済み疑似空白ページを索引ツリーから除去することを指定します。これは、疑似空白ではないページ上の疑似削除済みキーはクリーンアップしません。これは疑似空白リーフ・ページだけをチェックするので、ほとんどの場合に **ALL** オプションを使用するよりも相当速くなります。

**CLEANUP PAGES** オプションは、コミット済み疑似空白ページを検索して解放します。コミット済み疑似空白ページとは、ページ上のすべてのキーに削除済みのマークが付いていて、それらすべての削除がコミット済みとして知られているページのことです。索引内の疑似空白ページの数、**RUNSTATS** を実行して **SYSCAT.INDEXES** の **NUM EMPTY LEAFS** 列を調べることにより判別できます。**PAGES** オプションは、コミット済みと判別された場合に **NUM EMPTY LEAFS** を消去します。

**ALLOW READ ACCESS** または **ALLOW WRITE ACCESS** オプションを使用して、索引が再編成されている間に、他のトランザクションに表に対する読み取り専用または読み取り/書き込みのいずれかのアクセス権限を許可することができます。索引の再編成されたコピーを使用できる期間中に索引を再作成する場合、表へのアクセスは許可されません。

## INDEX *index-name* REBUILD

**REBUILD** オプションがデフォルトです。このオプションは、**CLEANUP** 節および **CONVERT** 節が指定されていない場合の以前のリリースにおける索引再編成で提供されるのと同じ機能を表します。索引再編成の **REBUILD** オプションにより、物理的に連続したページに索引データが再作成されます。デフォルトのアクセス・モードは表タイプによって異なります。

## INDEX *index-name* RECLAIM EXTENTS

再編成する索引を指定し、使用されていないエクステントを再利用します。

このアクションによって、索引オブジェクト内の索引ページを移動して空エクステントが作成され、索引オブジェクトの排他的使用からこれらの空エクステントを解放され、表スペース内の他のデータベース・オブジェクトがスペースを使用できるようにします。エクステントは、索引オブジェクトから表スペースに戻されて再利用されます。**ALLOW READ ACCESS** はデフォルトですが、すべてのアクセス・モードがサポートされています。

#### **TABLE** *table-name* **RECLAIM EXTENTS**

再編成する表を指定し、使用されていないエクステントを再利用します。*table-name* 変数では、マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング (ITC) 表を指定する必要があります。*schema.table-name* 形式の名前あるいは別名を使用することができます。*schema* には、表作成時のユーザー名が入ります。スキーマ名を省略した場合、デフォルトのスキーマが想定されます。

**REORG TABLE RECLAIM EXTENTS** の場合、**ON DATA PARTITION** 節が指定されるときに、アクセス節は指定されたパーティションに対してのみ適用されます。指定されたパーティションのエクステントが再利用される間に、ユーザーは表の残りの部分から読み取ったりそこに書き込んだりできます。この状態も、デフォルトのアクセス・レベルに適用されます。

#### **ALLOW NO ACCESS**

**REORG TABLE RECLAIM EXTENTS** の場合、エクステントが再利用される間に、他のユーザーがその表にアクセスできないことを指定します。

#### **ALLOW READ ACCESS**

**REORG TABLE RECLAIM EXTENTS** の場合、エクステントが再利用される間に、他のユーザーがその表に読み取り専用アクセスができることを指定します。

#### **ALLOW WRITE ACCESS**

**REORG TABLE RECLAIM EXTENTS** の場合、エクステントが再利用される間に、他のユーザーがその表で読み取り/書き込みができることを指定します。

#### **TABLE** *table-name*

再編成する表を指定します。表は、ローカルまたはリモート・データベースにあるものです。*schema.table-name* 形式の名前あるいは別名を使用することができます。*schema* には、表作成時のユーザー名が入ります。スキーマ名を省略した場合、デフォルトのスキーマが想定されます。

型付き表の場合、指定する表名は階層のルート表の名前でなければなりません。

マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング (ITC) 表の再編成に対して索引を指定することはできません。表のインプレース再編成を MDC 表または ITC 表に対して使用することはできません。

データ・パーティション表における表の再編成に対して **ON DATA PARTITION** 節が指定されている場合、指定されたデータ・パーティションのみが再編成されます。



- 表に非パーティション索引が定義されていない場合 (システム生成された XML パス索引を除く)、アクセス・モードは指定されたパーティションにのみ適用され、ユーザーは表の他のパーティションから読み取ったりそこに書き込んだりできます。
- 表に非パーティション索引が定義されている場合 (システム生成された XML パス索引を除く)、**ALLOW NO ACCESS** モードがデフォルトとなり、サポートされる唯一のアクセス・モードになります。この場合、その表は **ALLOW NO ACCESS** モードになります。**ALLOW READ ACCESS** が指定されている場合、SQL1548N が戻されます (SQLSTATE 5U047)。

表 56. 非パーティションおよびパーティション表における表の再編成に対してサポートされるアクセス・モード

コマンド	表タイプ	表パーティション節	サポートされるアクセス・モード
REORG TABLE	非パーティション表	該当なし	ALLOW NO ACCESS, ALLOW READ ACCESS <sup>1</sup>
REORG TABLE	パーティション表	指定なし	ALLOW NO ACCESS <sup>1</sup>
REORG TABLE (索引がない、または表にパーティション索引のみが定義されている)	パーティション表	ON DATA PARTITION	ALLOW NO ACCESS, ALLOW READ ACCESS <sup>1</sup>
REORG TABLE (システム生成された XML パス索引を除き、表に非パーティション索引が定義されていない)	パーティション表	ON DATA PARTITION	ALLOW NO ACCESS <sup>1</sup>

注:

1. アクセス節が指定されていない場合のデフォルト・モード。

データ・パーティション表では、表を再編成すると、表の再編成後にその表の非パーティション索引およびパーティション索引が再作成されます。 **ON DATA PARTITION** 節を使用して、データ・パーティション表の特定のデータ・パーティションを再編成する場合、表の再編成によって、指定したパーティションのみで非パーティション索引およびパーティション索引が再作成されます。

**INDEX** *index-name*

表を再編成する際に使用する索引を指定します。 *schema.index-name* 形式の完全修飾名を指定しない場合、デフォルトのスキーマが想定されます。 *schema* は、その索引が作成された時のユーザー名です。データベース・マネージャは、再編成している表のレコードを物理的に再配列する索引を使用します。

表のインプレース再編成では、クラスタリング索引が表に定義されて、索引が指定されている場合、それはクラスタリング索引でなければなりません。インプレース・オプションが指定されない場合、指定された任意の索引が使用されます。索引名を指定しない場合には、そのレコードは順番に関係なく再編成されます。しかし、表にクラスタリング索引が定義されている場合、索引が指定されてい

ければ、クラスタリング索引が使用されて表がクラスタリングされます。MDC 表または ITC 表を再編成しているときには、索引を指定できません。

**INDEX** および **ON DATA PARTITION** 節の両方を使用して表が再編成される場合、指定したパーティションのみが、索引 *index-name* を使用して再編成されます。

#### **ALLOW NO ACCESS**

表が再編成される間に、他のユーザーがその表にアクセスできないことを指定します。

**ON DATA PARTITION** 節を使用せずにパーティション表を再編成する場合、**ALLOW NO ACCESS** モードがデフォルトになり、唯一のサポートされるアクセス・モードになります。

データ・パーティション表に対して **ON DATA PARTITION** 節が指定されている場合、指定されたデータ・パーティションのみが再編成されます。

- 表に非パーティション索引が定義されていない場合 (システム生成された XML パス索引を除く)、指定されたパーティションのみが **ALLOW NO ACCESS** モードに制限されます。ユーザーは表の他のパーティションから読み取ったりそこに書き込んだりできません。
- 表に非パーティション索引が定義されている場合 (システム生成された XML パス索引を除く)、**ALLOW NO ACCESS** モードがデフォルトとなり、サポートされる唯一のアクセス・モードになります。この場合、その表は **ALLOW NO ACCESS** モードになります。

#### **ALLOW READ ACCESS**

再編成の際に表に対する読み取りアクセスだけを許可します。

**ALLOW READ ACCESS** モードが、非パーティション表のデフォルト・モードになります。

データ・パーティション表に対して **ON DATA PARTITION** 節が指定されている場合、指定されたデータ・パーティションのみが再編成されます。

- 表に非パーティション索引が定義されていない場合 (システム生成された XML パス索引を除く)、**ALLOW READ ACCESS** モードがデフォルト・モードとなり、指定されたパーティションのみがそのアクセス・モード・レベルに制限されます。ユーザーは表の他のパーティションから読み取ったりそこに書き込んだりできません。
- 表に非パーティション索引が定義されている場合 (システム生成された XML パス索引を除く)、**ALLOW READ ACCESS** モードはサポートされません。この場合に **ALLOW READ ACCESS** が指定されると、SQL1548N が戻されます (SQLSTATE 5U047)。

#### **INPLACE**

ユーザー・アクセスを許可しながら、表を再編成します。

インプレース表再編成が可能なのは、拡張索引を持たず、表内の XML 列に対して索引が定義されていない非パーティション表、非 MDC 表、および非 ITC 表のみです。表のインプレース再編成は、サイズが少なくとも 3 ページである表に対してのみ実行できます。

表のインプレース再編成は非同期に発生するので、即時に有効にならないことがあります。

#### **ALLOW READ ACCESS**

再編成の際に表に対する読み取りアクセスだけを許可します。

#### **ALLOW WRITE ACCESS**

再編成の際に表に対する書き込みアクセスを許可します。これがデフォルトの動作です。

#### **NOTRUNCATE TABLE**

インプレース再編成の後に表を切り捨てないでください。切り捨ての際に、表は S ロックされます。

**START** インプレース **REORG** 処理を開始します。これがデフォルトなので、このキーワードはオプションです。

**STOP** インプレース **REORG** 処理を現時点で停止します。

**PAUSE** インプレース **REORG** を当面の間、中断または一時停止します。

**RESUME** 以前に一時停止した、表のインプレース再編成を継続または再開します。オンライン再編成が再開された時点で、再編成の一時停止時と同じオプションを指定したい場合は、再開時にそれらのオプションを再び指定する必要があります。

#### **USE *tblspace-name***

再編成されている表の一時コピーを保管する **SYSTEM TEMPORARY** 表スペースの名前を指定します。表スペースの名前を指定しない場合、データベース・マネージャは、再編成しようとする表を含む表スペースにその表の作業用コピーを保管します。

8 KB、16 KB、または 32 KB の表オブジェクトの場合、指定した **SYSTEM TEMPORARY** 表スペースのページ・サイズが、表データの存在する表スペースのページ・サイズと一致していなければ、**DB2** データベース製品は、**LONG/LOB** オブジェクトのサイズが正しい **TEMPORARY** 表スペースを検出しようとします。再編成が正常に実行されるためには、そのような表スペースが存在していなければなりません。

パーティション表の場合、表に含まれるデータ・パーティションの再編成において、**TEMPORARY** 表スペースが一時ストレージとして使用されます。パーティション表全体の再編成では、一度に 1 つのデータ・パーティションが再編成されます。**TEMPORARY** 表スペースは、表全体ではなく、表に含まれる最大のデータ・パーティションを保持する必要があります。**ON DATA PARTITION** 節が指定される場合、**TEMPORARY** 表スペースは指定されたパーティションを保持する必要があります。

パーティション表の表スペース名を指定しない場合、各データ・パーティションの存在する表スペースが、そのデータ・パーティションの一時ストレージとして使用されます。各データ・パーティションの表スペースには、そのデータ・パーティションのコピーが入るだけの十分なフリー・スペースがなければなりません。

#### INDEXSCAN

クラスタリング **REORG** では、索引スキャンが使用されて表レコードが再配列されます。索引を介して表にアクセスすることにより、表の行を再編成します。デフォルトの方法は、必要に応じて **TEMPORARY** 表スペースを使用しながら、表をスキャンして結果をソートし表を再編成することです。索引キーはソートの順序に配列していますが、スキャンおよびソートはまず索引から行 ID を読み取って行をフェッチするよりも通常は高速です。

#### LONGLOBDATA

長いフィールドおよび LOB データが再編成されます。

表に長い列または LOB 列が含まれる場合でも、これは必要ではありません。これは時間がかかり、クラスタリングを改善しないために、デフォルトではこれらのオブジェクトを再編成しません。しかし、XML 列を持つ表に対して **LONGLOBDATA** オプションを指定して再編成を実行すると、未使用のスペースが再利用されるため、XML ストレージ・オブジェクトのサイズが削減されます。

このパラメーターは、既存の LOB データをインライン化された LOB データに変換するときが必要です。

#### USE *longtbspace-name*

これはオプション・パラメーターであり、LONG データを再作成するために使用する **TEMPORARY** 表スペースの名前を指定するために使用できます。表オブジェクトについても LONG オブジェクトについても **TEMPORARY** 表スペースが指定されていない場合、現在それらのオブジェクトが存在している表スペース中にそれらのオブジェクトが構成されることとなります。表の **TEMPORARY** 表スペースが指定されているが、このパラメーターが指定されていない場合には、ページ・サイズが異なるのでない限り、基本再編成データのために使用される表スペースが使用されることとなります。ページ・サイズが異なる場合、DB2 データベース・システムは、LONG オブジェクトを作成するために適切なページ・サイズの一時コンテナを選択することを試みます。

**USE *longtbspace-name*** が指定されている場合、**USE *tbspace-name*** も指定する必要があります。そうでない場合、*longtbspace-name* 引数は無視されます。

#### KEEPDICTIONARY

表の **COMPRESS** 属性が **YES** であり、表にコンプレッション・ディクショナリーがある場合、新しいディクショナリーは作成されません。再編成中に処理されるすべての行は、既存のディクショナリーを使用して圧縮されます。 **COMPRESS** 属性が **YES** であり、表にコンプレッション・ディクショナリーが存在しない場合、ディクショナリーは、表が特定のサイズ (およそ 1 から 2 MB) で、この表に十

分のデータがある場合のみ、このシナリオで作成されます (さらに表が圧縮されます)。代わりに明示的に **REORG RESETDICTIONARY** を指定した場合には、表に少なくとも 1 行あれば、ディクショナリーが作成されます。表の **COMPRESS** 属性が **NO** であり、表にコンプレッション・ディクショナリーがある場合、再編成処理によりそのディクショナリーは保存され、新たに編成された表のすべての行は非圧縮形式になります。基本表の行に保管されない **LOB** などの一部のデータは圧縮できません。

**LONGLOBDATA** オプションを指定しない場合、表の行データだけが再編成されます。以下の表は、**LONGLOBDATA** オプションが指定されていないときの、**REORG** コマンドの **KEEPDICTIONARY** 構文の動作を説明しています。

表 57. **REORG KEEPDICTIONARY**

圧縮	ディクショナリーが存在するかどうか	結果と効果
Y	Y	ディクショナリーはそのまま、行圧縮
Y	N	ディクショナリーを作成、行圧縮
N	Y	ディクショナリーはそのまま、全行圧縮解除
N	N	影響なし、全行圧縮解除

以下の表は、**LONGLOBDATA** オプションが指定されているときの、**REORG** コマンドの **KEEPDICTIONARY** 構文の動作を説明しています。

表 58. **LONGLOBDATA** オプションが指定されているときの **REORG KEEPDICTIONARY**。

圧縮	表の行データ・ディクショナリーが存在する	XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	コンプレッション・ディクショナリー	データ圧縮
Y	Y	Y	ディクショナリーを保存します。	既存のデータは、圧縮されます。新規データは、圧縮されます。
Y	Y	N	表の行ディクショナリーを保存し、XML ストレージ・オブジェクト・ディクショナリーを作成します。	既存のデータは、圧縮されます。新規データは、圧縮されます。
Y	N	Y	表の行ディクショナリーを作成し、XML ディクショナリーを保存します。	既存のデータは、圧縮されます。新規データは、圧縮されます。
Y	N	N	表の行ディクショナリーおよび XML ディクショナリーを作成します。	既存のデータは、圧縮されます。新規データは、圧縮されます。

表 58. LONGLOBDATA オプションが指定されているときの REORG  
KEEPDICTIONARY。(続き)

圧縮	表の行データ・ディクショナリーが存在する	XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	コンプレッション・ディクショナリー	データ圧縮
N	Y	Y	表の行ディクショナリーおよび XML ディクショナリーを保存します。	表データは圧縮解除されます。新規データは、圧縮されません。
N	Y	N	表の行ディクショナリーを保存します。	表データは圧縮解除されます。新規データは、圧縮されません。
N	N	Y	XML ディクショナリーを保存します。	表データは圧縮解除されます。新規データは、圧縮されません。
N	N	N	影響なし。	表データは圧縮解除されます。新規データは、圧縮されません。

**注:**

1. コンプレッション・ディクショナリーは、XML 列が DB2 V9.7 以降の表に追加された場合または表が ONLINE\_TABLE\_MOVE ストアード・プロシージャを使用してマイグレーションされた場合にのみ、表の XML ストレージ・オブジェクトに対して作成できます。

表の圧縮属性が NO の場合、(置換操作などで) 表の再初期設定または切り捨てが発生したなら、ディクショナリーが存在していればそれは廃棄されます。逆に、表の圧縮属性が YES の場合にディクショナリーが存在しているなら、切り捨てではディクショナリーが保たれ、廃棄は実行されません。リカバリーのことを考慮して、また将来データ・キャプチャーの変更 (つまりレプリケーション) をサポートすることを考慮して、ディクショナリーは全体としてログ記録されます。

**RESETDICTIONARY**

表の COMPRESS 属性が YES の場合、新しい行コンプレッション・ディクショナリーが作成されます。再編成の際に処理されるすべての行は、この新しいディクショナリーによる圧縮の対象になります。前のすべてのディクショナリーは、このディクショナリーに置き換わります。表の COMPRESS 属性が NO であり、表にコンプレッション・ディクショナリーが存在している場合、再編成処理によりそのディクショナリーは除去され、新たに編成された表のすべての行は非圧縮形式になります。基本表の行に保管されない LOB などの一部のデータは圧縮できません。

**LONGLOBDATA** オプションを指定しない場合、表の行データだけが再編成されます。以下の表は、**LONGLOBDATA** オプションが指定されていないときの、**REORG** コマンドの **RESETDICTIONARY** 構文の動作を説明しています。

表 59. *REORG RESETDICTIONARY*

圧縮	ディクショナリーが存在するかどうか	結果と効果
Y	Y	ディクショナリー新規作成*、行圧縮。DATA CAPTURE CHANGES オプションが CREATE TABLE または ALTER TABLE ステートメントで指定されていると、現在のディクショナリーが保持されます (これは履歴コンプレッション・ディクショナリーと呼ばれます)。
Y	N	ディクショナリー新規作成、行圧縮
N	Y	ディクショナリー除去、全行圧縮解除。DATA CAPTURE NONE オプションが CREATE TABLE または ALTER TABLE ステートメントで指定されていると、指定された表の履歴コンプレッション・ディクショナリーも除去されます。
N	N	影響なし、全行圧縮解除

\* - ディクショナリーが存在し、圧縮属性が有効になっていても、現在は表にデータがない場合は、**RESETDICTIONARY** 操作でも既存のディクショナリーが維持されます。内部の最小レコード長よりサイズの小さい行と、圧縮が試みられたときにレコード長の節減が図られない行は、このような場合は「十分でない」とみなされます。

以下の表は、**LONGLOBDATA** オプションが指定されているときの、**REORG** コマンドの **RESETDICTIONARY** 構文の動作を説明しています。

表 60. *LONGLOBDATA* オプションが指定されているときの *REORG RESETDICTIONARY*。

圧縮	表の行データ・ディクショナリーが存在する	XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	データ・ディクショナリー	データ圧縮
Y	Y	Y	ディクショナリーの作成 <sup>2</sup> 。	既存のデータは、圧縮されます。新規データは、圧縮されます。
Y	Y	N	新規の表の行ディクショナリーを作成し、新規の XML ディクショナリーを作成します <sup>3</sup> 。	既存のデータは、圧縮されます。新規データは、圧縮されます。
Y	N	Y	表の行データ・ディクショナリーを作成し、新規 XML ディクショナリーを作成します。	既存のデータは、圧縮されます。新規データは、圧縮されます。

表 60. LONGLOBDATA オプションが指定されているときの REORG RESETDICTIONARY。 (続き)

圧縮	表の行データ・ディクショナリーが存在する	XML ストレージ・オブジェクト・ディクショナリーが存在する <sup>1</sup>	データ・ディクショナリー	データ圧縮
Y	N	N	ディクショナリーを作成します。	既存のデータは、圧縮されます。新規データは、圧縮されます。
N	Y	Y	ディクショナリーを除去します。既存および新規のデータは圧縮されません。	既存の表データは圧縮解除されます。新規データは、圧縮されません。
N	Y	N	表の行ディクショナリーを除去します。すべてのデータは圧縮解除されます。	既存の表データは圧縮解除されます。新規データは、圧縮されません。
N	N	Y	XML ストレージ・オブジェクト・ディクショナリーを除去します。	既存の表データは圧縮解除されます。新規データは、圧縮されません。
N	N	N	影響なし。	既存の表データは圧縮解除されます。新規データは、圧縮されません。

注:

1. コンプレッション・ディクショナリーは、XML 列が DB2 V9.7 以降の表に追加された場合または表が Online Table Move を使用してマイグレーションされた場合にのみ、表の XML ストレージ・オブジェクトに対して作成できます。
2. ディクショナリーが存在していて圧縮属性が有効であるが、現在のところ表中にデータが存在しない場合、**RESETDICTIONARY** 操作では既存のディクショナリーがそのまま保たれます。この場合、不適切とみなされるのは、内部最小レコード長よりもサイズが小さい行、および圧縮してもレコード長の節約にならない行です。
3. DATA CAPTURE CHANGES オプションが CREATE TABLE または ALTER TABLE ステートメントで指定されていると、現在のデータ・ディクショナリーが保持されます (これは履歴コンプレッション・ディクショナリーと呼ばれます)。

**ON DATA PARTITION** *partition-name*

データ・パーティション表では、再編成するデータ・パーティションを指定します。

DB2 V9.7 フィックスパック 1 以降のリリースでは、この節を **REORG INDEXES ALL** コマンドで使用して、特定のパーティションのパーティション



索引を再編成することができます。また、**REORG TABLE** コマンドで使用して、特定のパーティションのデータを再編成することもできます。

パーティション表で **REORG TABLE** または **REORG INDEXES ALL** コマンドでこの節を使用するときに、パーティション *partition-name* が指定した表にない場合、再編成は失敗し、理由コード 1 と共に SQL2222N を戻します。パーティション *partition-name* がアタッチまたはデタッチ状態の場合、再編成は失敗し、理由コード 3 と共に SQL2222N を戻します。

**ON DATA PARTITION** 節を指定して **REORG INDEX** コマンドが発行された場合、再編成は失敗し、理由コード 2 と共に SQL2222N を戻します。

パーティション表が **REORG** ペンディング状態にあり、かつ表に非パーティション索引が定義されている場合、**REORG TABLE** コマンドは失敗し、SQL1549N (SQLSTATE 5U047) を戻します。

#### **ALL DBPARTITIONNUMS**

*db2nodes.cfg* ファイルに指定されているすべてのデータベース・パーティションで操作が実行されることを指定します。データベース・パーティション節が指定されていない場合、これがデフォルトです。

**EXCEPT** データベース・パーティション・リストに指定されているものを除き、*db2nodes.cfg* ファイルに指定されているすべてのデータベース・パーティションで操作が実行されることを指定します。

#### **ON DBPARTITIONNUM | ON DBPARTITIONNUMS**

データベース・パーティションのセットに対して操作を実行します。

##### *db-partition-number1*

データベース・パーティション・リスト内のデータベース・パーティション番号を指定します。

##### *db-partition-number2*

2 番目のデータベース・パーティション番号を指定し、*db-partition-number1* から *db-partition-number2* までのすべてのデータベース・パーティションがデータベース・パーティション・リストに含まれるようにします。

#### **例**

データベース・パーティション 1、3、および 4 で構成されるデータベース・パーティション・グループ内の表を再編成します。

```
CALL SYSPROC.ADMIN_CMD ('REORG TABLE employee  
INDEX empid ON DBPARTITIONNUM (1,3,4)')
```

#### **使用上の注意**

制約事項:

- コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。
- **REORG** ユーティリティは、操作の開始時に COMMIT ステートメントを発行しますが、これによってタイプ 2 接続の場合に、プロシージャは SQL30090N、理由コード 2 を戻します。
- **REORG** ユーティリティでは、ニックネームの使用はサポートされません。

- **REORG TABLE** コマンドは、宣言済み一時表または作成済み一時表に対してサポートされていません。
- **REORG TABLE** コマンドは、ビューに対しては使用できません。
- 表の再編成は、範囲がクラスター化された表との互換性がありません。表の範囲領域は常に、クラスター化されているからです。
- DMS 表スペース中のパーティション表が属している表スペース (LOB や索引を含む) のオンライン・バックアップが実行されている間は、**REORG TABLE** をその表に対して使用することはできません。
- **REORG TABLE** は、索引拡張子に基づく索引を使用できません。
- 表が **REORG** ペンディング状態なら、その表に対してインプレース再編成を実行することはできません。
- 同じ **TEMPORARY DMS** 表スペースの共有による表の再編成の同時実行はサポートされていません。
- イベント・モニターの書き込み先である表に対して再編成を実行する場合は、その前にその表に関係するイベント・モニターを非活動化しておく必要があります。
- データ・パーティション表では、以下のようになります。
  - **SYSCAT.TABLES** の中でその表の **ACCESS\_MODE** はフル・アクセス権限でなければなりません。
  - 再編成では、アタッチまたはデタッチ操作のため、制限状態にあるデータ・パーティションがスキップされます。**ON DATA PARTITION** 節が指定されている場合、そのパーティションは完全にアクセス可能でなければなりません。
  - 表の再編成中にエラーが発生した場合、一部の索引または索引パーティションは無効のままとなる可能性があります。表の非パーティション索引は、再編成が最初のデータ・パーティションの置換フェーズに達するかそれをパスすると、無効のマークが付きます。置換フェーズに既に達したかまたはそれを通過したデータ・パーティションの索引パーティションには、無効のマークが付きます。索引は次回表またはデータ・パーティションにアクセスしたときに再作成されます。
  - **ALLOW NO ACCESS** モードが使用されている場合、索引の再編成中にエラーが発生すると、表の一部の索引が無効のままになる場合があります。表の非パーティション **RID** 索引については、障害発生時点で再編成中の索引のみが無効のままになります。非パーティション・ブロック索引がある **MDC** 表については、エラーが発生した場合、1 つ以上のブロック索引が無効のままになる可能性があります。パーティション索引を持つ **MDC** 表または **ITC** 表の場合、再編成中のデータ・パーティションの索引オブジェクトだけが無効のままになります。無効のマークが付いたすべての索引は、次回表またはデータ・パーティションにアクセスしたときに再作成されます。
  - 表に定義されたパーティション索引のみを使用するデータ・パーティション表が **REORG** ペンディング状態にある場合、**ON DATA PARTITION** 節を指定して **REORG TABLE** コマンドを発行すると、指定されたデータ・パーティションのみ **REORG** ペンディング状態ではなくなります。その表の残りのパーティションを **REORG** ペンディング状態から戻すためには、表全体に対して **REORG TABLE**

コマンド (**ON DATA PARTITION** 節を使用しないで) を発行するか、または残っているパーティションそれぞれに対して **ON DATA PARTITION** 節を指定して **REORG TABLE** コマンドを発行します。

表の再編成の現在の進行状態に関する情報は、データベース活動の履歴ファイルに書き込まれます。履歴ファイルには、再編成イベントごとの記録が含まれています。このファイルを表示するには、再編成している表を含むデータベースに対して **LIST HISTORY** コマンドを実行します。

さらに、表スナップショットを使用して表の再編成の進行状況をモニターすることもできます。表の再編成のモニター・データは、「データベース・モニター表スイッチ (Database Monitor Table Switch)」の設定値に関係なく記録されます。

エラーが生じた場合、SQLCA ダンプが履歴ファイルに書き込まれます。表のインプレース再編成の場合、状況が **PAUSED** として記録されます。

索引付き表が何回も変更されると、索引内のデータがフラグメント化されることがあります。表が索引に関してクラスター化されている場合、表および索引をクラスターの順序から取り出すことができます。これら両方の要素は索引を使用するスキヤンのパフォーマンスを低下させ、索引ページの事前取り出しの効果に影響を与えることがあります。 **REORG INDEX** または **REORG INDEXES** で **REBUILD** オプションを指定して、表の 1 つの索引またはすべての索引を再編成できます。索引を再編成して再作成すると、フラグメントが除去され、リーフ・ページに物理クラスタリングがリストアされます。 **REORGCHK** を使用すると、索引に再編成が必要かどうかを判断するために役立ちます。すべてのデータベース操作が完了し、すべてのロックを解放したことを確かめてから、索引の再編成を呼び出してください。これは、**WITH HOLD** でオープンされた、すべてのカーソルをクローズした後で **COMMIT** または **ROLLBACK** を発行することによって行われます。

従来の表の再編成 (オフライン再編成) では、再編成の最後のフェーズで索引が再作成されます。 **TEMPORARY** 表スペースが複数存在する場合は、表の再編成処理に伴って追加のソートを行う際に、**REORG TABLE** コマンドで指定した **TEMPORARY** 表スペースに加えて、それ以外の **TEMPORARY** 表スペースが 1 つ使用される可能性もあります。ただし、表のインプレース再編成 (オンライン再編成) では索引は再作成されません。インプレース表再編成の完了後に、**REORG INDEXES** コマンドを発行することをお勧めします。インプレース表再編成は非同期であるため、**REORG INDEXES** コマンドを発行する前にインプレース表再編成が完了していることを確認するように注意しなければなりません。インプレース表再編成が完了する前に **REORG INDEXES** コマンドを発行すると、再編成が失敗するおそれがあります (SQLCODE -2219)。

何回も修正されてデータがフラグメント化して、アクセス・パフォーマンスが大幅に低下した表も **REORG TABLE** コマンドの対象になります。構造化タイプ列のインラインの長さを変更後、このユーティリティーもまた呼び出して、変更の有用性を確認してください。 **REORGCHK** コマンドを使用して、表の再編成が必要であるかどうか判断してください。すべてのデータベース操作が完了し、すべてのロックが解放されていることを確かめてから、**REORG TABLE** を呼び出してください。これは、**WITH HOLD** でオープンされた、すべてのカーソルをクローズした後で **COMMIT** または **ROLLBACK** を発行することによって行われます。表の再編成の後で、

**RUNSTATS** を使用して表統計を更新し、**REBIND** を使用してこの表を使用するパッケージを再バインドします。再編成ユーティリティーは、暗黙的にすべてのカーソルをクローズします。

DB2 V9.7 フィックスパック 1 以降では、1 つのデータ・パーティション表に対して **REORG TABLE** コマンドおよび **REORG INDEXES ALL** コマンドを発行し、同時に複数のデータ・パーティション、または 1 つのパーティション上の複数のパーティション索引を再編成することができます。複数のデータ・パーティション、または 1 つのパーティション上の複数のパーティション索引を並行して再編成する場合、ユーザーは影響を受けないパーティションにはアクセスできますが、影響を受けるパーティションにはアクセスできません。同一表に対して同時に操作する複数の **REORG** コマンドを発行するには、以下の基準をすべて満たす必要があります。

- 各 **REORG** コマンドは、**ON DATA PARTITION** 節を指定して、それぞれ別個のパーティションを指定しなければならない。
- 各 **REORG** コマンドは、**ALLOW NO ACCESS** モードを使用して、データ・パーティションに対するアクセスを制限しなければならない。
- **REORG TABLE** コマンドを発行する場合、パーティション表にはパーティション索引のみが含まれていなければならない。非パーティション索引が表に定義されているはなりません (システム生成された XML パス索引を除く)。

非パーティション索引がなく (システムが生成した XML パス索引を除く)、パーティション P1、P2、P3、P4 を使用するパーティション表 T1 の場合、以下の複数の **REORG** コマンドを並行して実行することができます。

```
REORG INDEXES ALL FOR TABLE T1 ALLOW NO ACCESS ON DATA PARTITION P1
REORG TABLE T1 ALLOW NO ACCESS ON DATA PARTITION P2
REORG INDEXES ALL FOR TABLE T1 ALLOW NO ACCESS ON DATA PARTITION P3
```

並行 **REORG** コマンドを使用する場合、以下のような操作はサポートされません。

- **ON DATA PARTITION** 節を指定せずに、表に対して **REORG** コマンドを使用する。
- データ・パーティションを追加、アタッチ、またはデタッチするのに、表で **ALTER TABLE** ステートメントを使用する。
- データを表にロードする。
- 表を含むオンライン・バックアップを実行する。

表の値圧縮をアクティブ化または非アクティブ化したために表に混合した行形式が含まれている場合、オフラインで表を再編成することによって、既存の行すべてをターゲットの行形式に変換することができます。

表がいくつかのデータベース・パーティションに分散している場合、影響を受けるデータベース・パーティションのいずれかで表または索引の再編成が失敗すると、失敗したデータベース・パーティションでのみ表または索引の再編成がロールバックされます。

再編成が成功しなかった場合には、一時ファイルを削除すべきではありません。データベース・マネージャーは、これらのファイルを使用し、データベースをリカバリします。

索引の名前が指定されると、データベース・マネージャーはその索引の順番に従って、データを再編成します。パフォーマンスを最大にするため、SQL 照会で頻繁に

使用される索引を指定してください。索引の名前が指定されておらず、クラスタリング索引が存在する場合、データはクラスタリング索引に従って配列されます。

表の PCTFREE 値は、ページごとに指定されたフリー・スペースの量を決定します。この値が設定されていない場合、ユーティリティーは、それぞれのページにできるだけ大きなスペースを割り当てます。

表の再編成の後に表スペースのロールフォワード・リカバリーを完了させるには、通常の表スペースと LARGE 表スペースの両方で、ロールフォワード・リカバリーが有効になっていなければなりません。

その表が、**COMPACT** オプションを使用しない LOB 列を含む場合、LOB DATA ストレージ・オブジェクトは、表の再編成に従いかなり大きくなることができます。これは、行が再編成された順序、および使用される (SMS または DMS) 表スペースのタイプの結果になります。

**REORG INDEXES/TABLE** コマンドにより、XML データに対する索引が再作成される場合があります。詳細については、『XML データに対する索引の再作成』を参照してください。

**RESET ALERT CONFIGURATION** コマンド (**ADMIN\_CMD** プロシージャを使用):

特定のオブジェクトのヘルス・インディケータ設定を、そのオブジェクト・タイプの現行デフォルトにリセットするか、またはオブジェクト・タイプの現行のデフォルトのヘルス・インディケータ設定を、インストール時のデフォルトにリセットします。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、このコマンドおよび API は推奨されておらず、将来のリリースで除去される可能性があります。これは DB2 pureScale 環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

## 許可

以下の権限のいずれか。

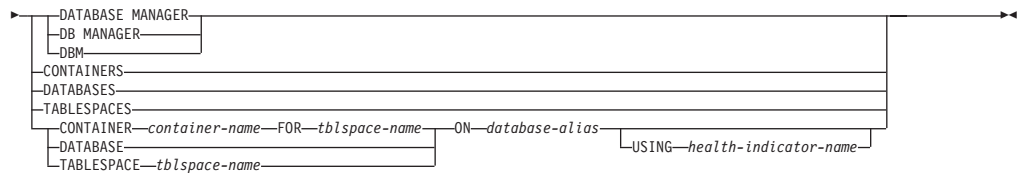
- SYSADM
- SYSMANT
- SYSCTRL

## 必要な接続

データベース

## コマンド構文

```
→→→RESET ALERT CONFIGURATION FOR →→→
      |-----|
      | CONFIG |
      |-----|
      |  CFG  |
```



## コマンド・パラメーター

### DATABASE MANAGER | DB MANAGER | DBM

データベース・マネージャーでアラート設定をリセットします。

### CONTAINERS

データベース・マネージャーが管理するすべての表スペース・コンテナのデフォルトのアラート設定を、インストール時のデフォルトにリセットします。これは、カスタム設定を持たないすべての表スペース・コンテナに適用される設定です。カスタム設定は、**CONTAINER container-name FOR tblspace-name ON database-alias** 節を使って定義されます。

### DATABASES

データベース・マネージャーが管理するすべてのデータベースのアラート設定をリセットします。これは、カスタム設定を持たないすべてのデータベースに適用される設定です。カスタム設定は、**DATABASE ON database-alias** 節を使って定義されます。

### TABLESPACES

データベース・マネージャーが管理するすべての表スペースのデフォルトのアラート設定を、インストール時のデフォルトにリセットします。これは、カスタム設定を持たないすべての表スペースに適用される設定です。カスタム設定は、**TABLESPACE tblspace-name ON database-alias** 節を使って定義されます。

### CONTAINER container-name FOR tblspace-name ON database-alias

**ON database-alias** 節を使って指定したデータベース上で、**FOR tblspace-name** 節を使って指定した表スペースの、*container-name* という名前の表スペース・コンテナのアラート設定をリセットします。この表スペース・コンテナにカスタム設定がある場合、これらの設定は除去され、現行の表スペース・コンテナのデフォルトが使用されます。

### DATABASE ON database-alias

**ON database-alias** 節を使って指定したデータベースのアラート設定をリセットします。このデータベースにカスタム設定がある場合、これらの設定は除去され、インストール時のデフォルトが使用されます。

### TABLESPACE tblspace-name ON database-alias

**ON database-alias** 節を使って指定したデータベース上で、*tblspace-name* という名前の表スペースのアラート設定をリセットします。この表スペースにカスタム設定がある場合、これらの設定は除去され、インストール時のデフォルトが使用されます。

### USING health-indicator-name

アラート構成がリセットされるヘルス・インディケーターの設定を指定し

ます。ヘルス・インディケータ名は 2 文字のオブジェクト ID で構成され、その後にインディケータの測定対象を説明する名前が続きます。以下に例を示します。

```
db.sort_privmem_util
```

このオプションを指定しない場合は、指定したオブジェクトまたはオブジェクト・タイプのすべてのヘルス・インディケータがリセットされます。

## 例

**ADMIN\_CMD** プロシージャが含まれているデータベースを所有するデータベース・マネージャのアラートの設定値をリセットします。

```
CALL SYSPROC.ADMIN_CMD( 'reset alert cfg for dbm' )
```

## 使用上の注意

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

**ADMIN\_CMD** プロシージャはサーバーでのみ実行されるため、*database-alias* は、サーバー上のカタログの中で定義されているローカル・データベースでなければなりません。

**RESET DATABASE CONFIGURATION** コマンド (**ADMIN\_CMD** プロシージャを使用):

特定データベースの構成をシステム・デフォルトにリセットします。

## 有効範囲

このコマンドは、アプリケーションの接続先であるデータベース・パーティションにのみ影響を与えます。

## 許可

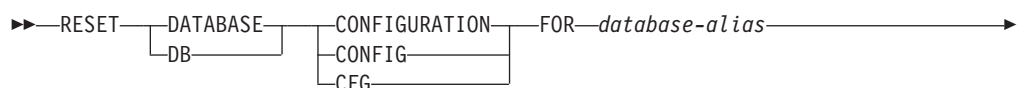
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMANT

## 必要な接続

データベース

## コマンド構文



MEMBER *member-number*

## コマンド・パラメーター

### **FOR** *database-alias*

構成がシステム・デフォルトにリセットされるそのデータベースの別名を指定します。データベース別名は、サーバー上のカタログの中で定義されているものでなければならず、サーバー上のローカル・データベースを指していなければなりません。

### **MEMBER** *member-number*

データベース構成のリセットが特定のメンバーに適用される場合、このパラメーターが使用されることがあります。このパラメーターが指定されていない場合、リセットはすべてのメンバーに対して有効です。

## 例

サーバー上で別名 **SAMPLE** でカタログされているデータベースの構成をリセットします。

```
CALL SYSPROC.ADMIN_CMD( 'reset db cfg for SAMPLE' )
```

## 使用上の注意

データベース構成パラメーターのリストを表示または印刷するには、**SYSIBMADM.DBCFG** 管理ビューを使用してください。

構成可能なパラメーターの値を変更するには、**UPDATE DATABASE CONFIGURATION** コマンドを使用してください。

データベース構成ファイルへの変更は、ファイルがメモリーにロードされた後のみ有効になります。これを行う前にすべてのアプリケーションはデータベースから切断されている必要があります。

エラーが発生した場合、データベース構成ファイルは変更されません。

チェックサムが無効である場合には、データベースの構成ファイルは、リセットできません。適当なコマンドを使用しないでデータベース構成ファイルを変更するとこれが発生することがあります。これが発生する場合、データベースをリストアしてデータベース構成ファイルをリセットする必要があります。

**RESET DATABASE CONFIGURATION** コマンドを実行すると、データベース構成パラメーターが文書化されているデフォルトの構成値にリセットされ、**auto\_runstats** は ON になります。**Self\_tuning\_mem** は、非パーティション・データベース環境の場合は ON にリセットされ、パーティション・データベース環境の場合は OFF にリセットされます。

コマンドの実行状況は、CALL ステートメントからの結果である **SQLCA** で戻されます。



ADMIN\_CMD プロシージャはサーバーでのみ実行されるため、*database-alias* は、サーバー上のカタログの中で定義されているローカル・データベースでなければなりません。

## 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** を **MEMBER** の代わりに使用できます。

## RESET DATABASE MANAGER CONFIGURATION コマンド (ADMIN\_CMD プロシージャを使用):

現在接続されているデータベースを含むインスタンスについて、データベース・マネージャーの構成ファイルのパラメーターをシステム・デフォルトにリセットします。この値は、ノード・タイプに基づいたデフォルト値にリセットされます。

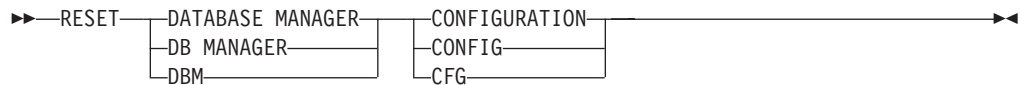
## 許可

SYSADM

## 必要な接続

データベース

## コマンド構文



## コマンド・パラメーター

なし

## 例

ADMIN\_CMD ストアード・プロシージャの属するデータベースを含むインスタンスの構成をリセットします。

```
CALL SYSPROC.ADMIN_CMD( 'reset dbm cfg' )
```

## 使用上の注意

このコマンドは、インストール・プログラムによって設定されたすべてのパラメーターをリセットします。パラメーターがリセットされると、DB2 を再始動するときにエラー・メッセージが戻される原因となる場合があります。例えば、**svcsname** パラメーターがリセットされると、DB2 を再始動しようとするとき、ユーザーは SQL5043N エラー・メッセージを受け取ります。

このコマンドを実行する前に、既存の設定値を参照できるようにするために、SYSIBMADM.DBMCFG 管理ビューからの出力をファイルに保存します。個々の設

定値は、ADMIN\_CMD プロシージャーを通じて **UPDATE DATABASE MANAGER CONFIGURATION** コマンドを使用して更新できます。

インストール・プログラムで設定される **svcename** パラメーターの、ユーザーによる修正は推奨されません。

データベース・マネージャー構成パラメーターのリストを表示または印刷するには、SYSIBMADM.DBMCFG 管理ビューを使用してください。構成可能なパラメーターの値を変更するには、ADMIN\_CMD プロシージャーを通じて **UPDATE DATABASE MANAGER CONFIGURATION** コマンドを使用してください。

これらのパラメーターについての詳細は、構成パラメーターおよび個々のパラメーターについてのサマリー・リストを参照してください。

データベース・マネージャー構成ファイルへの変更の一部は、ファイルがメモリーにロードされた後にのみ有効になります。オンラインで構成できるパラメーターと構成できないパラメーターについては、構成パラメーターの一覧をご覧ください。即時にリセットされないサーバー構成パラメーターは、**db2start** の実行中にリセットされます。クライアント構成パラメーターの場合、パラメーターは次にアプリケーションを開始するときにリセットされます。クライアントがコマンド行プロセッサである場合は、**TERMINATE** を呼び出す必要があります。

エラーが生じた場合には、データベース・マネージャー構成ファイルは変更されません。

データベース・マネージャー構成ファイルは、そのチェックサムが無効であると、リセットすることができません。このような状況は、適切なコマンドを使用せずに手動で構成ファイルが編集された場合などに発生します。チェックサムが無効な場合は、インスタンスを再作成する必要があります。

#### **REWIND TAPE コマンド (ADMIN\_CMD プロシージャーを使用):**

ストリーミング磁気テープ装置へのバックアップおよびリストア操作のためにテープを巻き戻します。このコマンドは Windows オペレーティング・システムでのみサポートされています。

#### **許可**

以下の権限のいずれか。

- SYSADM
- SYSCtrl
- SYSMaint

#### **必要な接続**

データベース

#### **コマンド構文**

▶▶—REWIND TAPE—┐  
└—ON—device—┘▶▶

## コマンド・パラメーター

### ON device

有効なテープ装置名を指定します。デフォルト値は ¥¥.¥TAPE0 です。装置の指定は、サーバーに対する相対指定でなければなりません。

### 例

¥¥.¥TAPE1' という装置のテープを巻き戻します。

```
CALL SYSPROC.ADMIN_CMD( 'rewind tape on ¥¥.¥TAPE1' )
```

### 使用上の注意

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

### RUNSTATS コマンド (ADMIN\_CMD プロシージャーを使用):

表やそれに関連した索引の特性、あるいは統計ビューの特性に関する統計情報を更新します。これらの特性には、レコード数、ページ数、および平均レコード長が含まれます。オプティマイザーは、データへのアクセス・パスを判別するとき、これらの統計を使用します。

表の場合、表で数多くの更新が行われたとき、または表の再編成後に、RUNSTATS コマンドを呼び出します。統計ビューの場合、基礎表に対する変更がそのビューによって返される行に対してかなりの影響を及ぼしている場合に、RUNSTATS コマンドを呼び出します。そのビューは、それ以前に ALTER VIEW ステートメントを使用して、照会最適化で使用できる状態になっていなければなりません。

### 有効範囲

RUNSTATS コマンドは、db2nodes.cfg ファイル内のいずれのデータベース・パーティションからでも発行できます。この API は、カタログ・データベース・パーティション上のカタログを更新するために使用できます。

表の場合、このコマンドは、呼び出し元のデータベース・パーティションの表の統計を収集します。表がそのデータベース・パーティションに存在しない場合、データベース・パーティション・グループの最初のデータベース・パーティションが選択されます。

ビューの場合、このコマンドは、関連するすべてのデータベース・パーティションに含まれる表のデータを使用して、統計情報を収集します。

### 許可

表の場合、以下の権限のいずれか 1 つです。

- SYSADM
- SYSCTRL
- SYSMANT
- DBADM
- SQLADM

- 表に対する CONTROL 特権
- LOAD authority

このコマンドを使用する際には、接続内に存在する宣言された一時表のいずれにおいても明示特権は必要ありません。

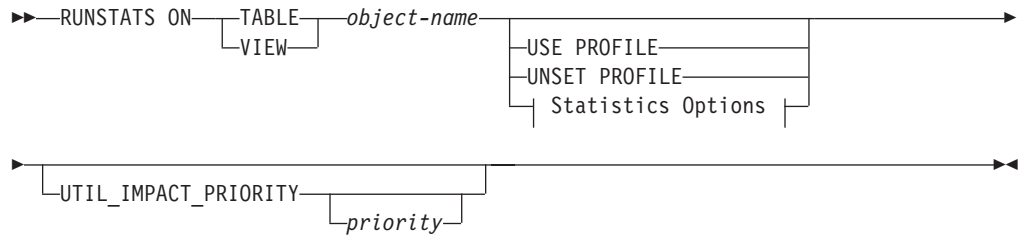
統計ビューの場合、以下の権限のいずれか 1 つです。

- SYSADM
- SYSCTRL
- SYSMANT
- DBADM
- SQLADM
- 統計ビューに対する CONTROL 特権

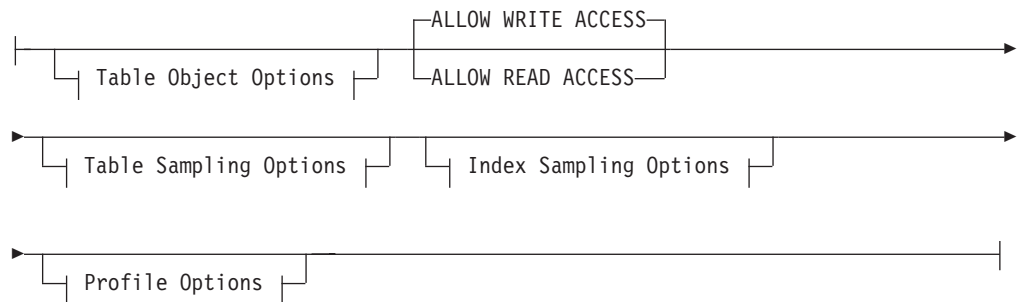
### 必要な接続

データベース

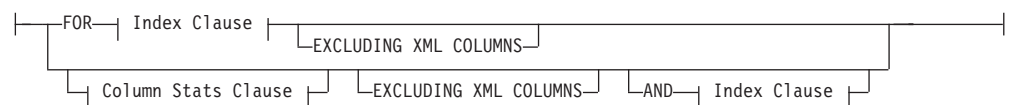
### コマンド構文



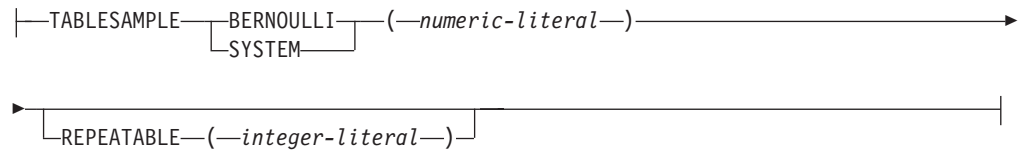
### Statistics Options:



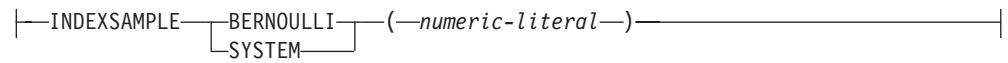
### Table Object Options:



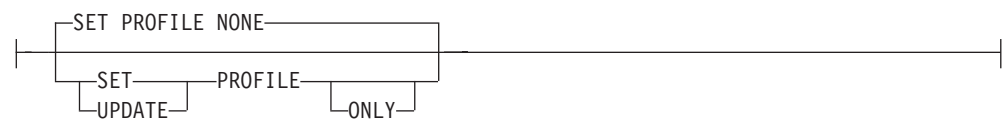
### Table Sampling Options:



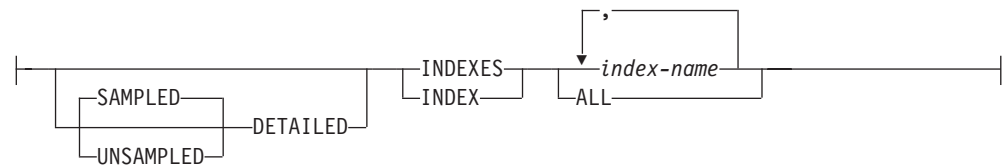
### Index Sampling Options:



### Profile Options:



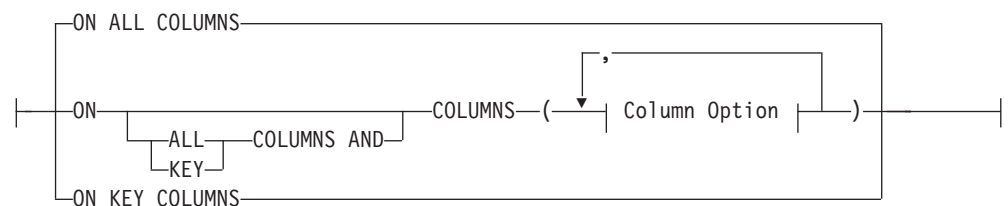
### Index Clause:



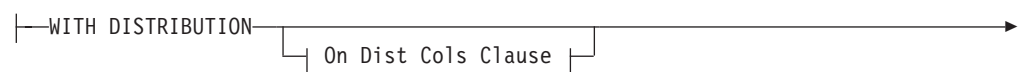
### Column Stats Clause:



### On Cols Clause:

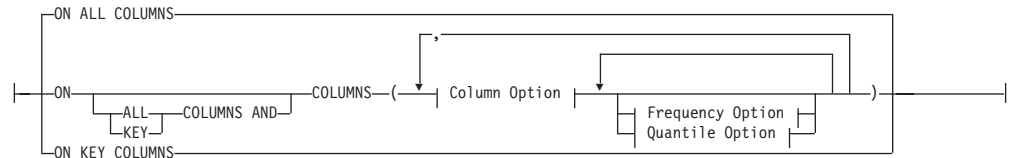


### Distribution Clause:

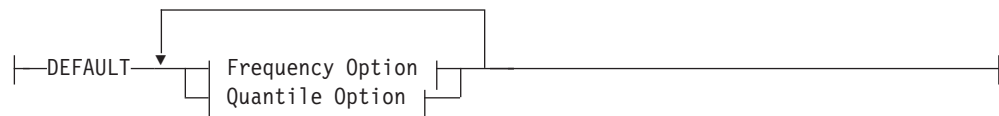




**On Dist Cols Clause:**



**Default Dist Option:**



**Frequency Option:**



**Quantile Option:**



**Column Option:**



**コマンド・パラメーター**

*object-name*

統計が収集される表または統計ビューを識別します。このパラメーターは、階層表であってはなりません。型付き表の場合は、*object-name* は表階層のルート表の名前でなければなりません。 *schema.object-name* という形式の完全修飾名または別名を使用しなければなりません。 *schema* には、表作成時のユーザー名が入ります。

**USE PROFILE**

このオプションを使用した **RUNSTATS** は、以前に保管した統計プロファイルを使用して表または統計ビューの統計を収集します。統計プロファイルの作成には、**SET PROFILE** オプションを使用し、更新には **UPDATE PROFILE** オプションを使用します。

## UNSET PROFILE

既存の統計プロファイルを除去する場合は、このオプションを指定します。以下に例を示します。

```
RUNSTATS ON tablemyschema.mytable UNSET PROFILE
```

## FOR INDEXES

索引のみの統計を収集および更新します。表に、以前に収集された表統計がない場合は、基本表統計も収集されます。これらの基本統計には、分散統計は一切含まれません。ビューの場合、このオプションは使用できません。

## SAMPLED

組み合わせて使用できるのは **DETAILED** パラメーターのみです。このオプションを指定しても、**DETAILED** によるデフォルトの機能は変更されません。このオプションは、以前のバージョンの **DB2** との互換性のために残されています。ビューの場合、このオプションは使用できません。

## UNSAMPLED

このオプションが **DETAILED** オプションと共に使用された場合、強制的に **RUNSTATS** が実行されることにより、索引内のすべての項目が調べられて拡張索引統計が計算されます。このオプションはビューに対しては使用できず、スキャン索引サンプリング (**INDEXSAMPLE** キーワード) と共に使用することもできません。このオプションを指定すると **RUNSTATS** のリソース使用量が大幅に増加させられます。一方、**DETAILED** オプションおよび **SAMPLED DETAILED** オプション (これらは等価です) では大幅な改善が得られることはほとんどありません。

## DETAILED

拡張された索引統計を計算します。拡張された索引統計とは、比較的大規模な索引の場合に収集される、**CLUSTERFACTOR** 統計および **PAGE\_FETCH\_PAIRS** 統計のことです。パフォーマンスを改善するため、索引項目のすべてを調べるのではなく、CPU サンプリング技法が採用されます。ビューの場合、このオプションは使用できません。

## *index-name*

表で定義されている既存の索引を識別します。 *schema.index-name* 形式の完全修飾名を指定しない場合、デフォルトのスキーマが想定されます。ビューの場合、このオプションは使用できません。

## EXCLUDING XML COLUMNS

この節を使用して、XML タイプのすべての列を、統計情報収集から除外します。XML データを含めた場合、必要となるシステム・リソースが増加することがあるため、この節を使用すれば、XML 以外の列に関する統計情報の収集が容易になります。**EXCLUDING XML COLUMNS** 節は、統計情報収集の対象として XML 列を指定する他の節よりも優先されます。例えば、**EXCLUDING XML COLUMNS** 節を使用する場合、**ON COLUMNS** 節で XML タイプの列を指定したり **ON ALL COLUMNS** 節を使用したりしても、統計情報収集において XML タイプの列はすべて無視されます。DB2 V9.7 フィックスパック 1 以降のリリースでは、この節が指定される場合、XML タイプの列に対する分散統計は収集されません。

## AND INDEXES

表と索引両方の統計を収集および更新します。ビューの場合、このオプションは使用できません。

## ON ALL COLUMNS

適格なすべての列で統計を収集するには、**ON ALL COLUMNS** 節を使用します。列は、基本統計の収集に指定するか (**On Cols** 節)、または **WITH DISTRIBUTION** 節と組み合わせて指定 (**On Dist Cols** 節) できます。これら列固有の節がどちらも指定されない場合は、デフォルト・オプションとして **ON ALL COLUMNS** が指定されます。

なお、**On Cols** 節でこれが指定された場合は、特定の列が **WITH DISTRIBUTION** 節の一部として選択されない限り、すべての列で基本列統計だけが収集されます。**WITH DISTRIBUTION** 節の一部として指定された列では、基本統計も分散統計も収集されます。

**WITH DISTRIBUTION ON ALL COLUMNS** が指定されている場合は、収集が行えるすべての列で基本統計と分散統計の両方が収集されます。**On Cols** 節での指定はすべて重複になるため、必要ありません。

## ON COLUMNS

特定の列、列グループ、またはその両方に関する統計を収集するには、**ON COLUMNS** を使用します。列グループとは、結合した統計を収集しようとする複数の列を括弧でくくったコンマ区切りリストです。

列および列グループは括弧でくくったコンマ区切りリストとして指定します。

索引統計情報を収集することなく表に対して **RUNSTATS** コマンドを実行し、統計情報収集の対象として列のサブセットを指定した場合、

- **RUNSTATS** コマンドに指定されていないが、索引の最初の列である列の統計情報は、リセットされません。
- **RUNSTATS** コマンドで指定されていない他のすべての列の統計情報は、リセットされます。

この節は、**On Cols** 節と **On Dist Cols** 節の中で使用できます。列のグループに関する分散統計の収集は、現在サポートされていません。

列グループの中に **XML** タイプの列が指定されている場合は、そのグループについて重複のない異なる値を収集するため、**XML** タイプのそれらの列は無視されます。しかし、その列グループ中の **XML** タイプ列について、**XML** 列の基本統計情報は収集されます。

## ON KEY COLUMNS

特定の列をリストする代わりに、表で定義されたすべての索引を構成する列の統計を収集することもできます。ここでは、照会に含まれる重要な列が、表での索引の作成にも使用されることが前提となっています。表に索引がない場合は、列がリストされず、列統計が収集されない場合と同様になります。これは、**On Cols** 節または **On Dist Cols** 節の中で使用できます。ただし、その両方の節で指定すると、**WITH DISTRIBUTION** 節で基本統計と分散統計の両方の収集が指定されているため、**On Cols** 節で重複が生じます。定義



により XML タイプの列はキー列ではなく、**ON KEY COLUMNS** 節による統計情報収集には含まれません。ビューの場合、このオプションは使用できません。

#### *column-name*

表または統計ビューの中の列の名前。存在しない列が指定された場合や列名の入力を誤った場合など、統計収集を行えない列の名前が指定された場合は、エラー (-205) が戻されます。一方は配分なし、一方は配分ありで、2つの列のリストを指定できます。**WITH DISTRIBUTION** 節が関連付けられていないリストで列を指定する場合は、基本列統計だけが収集されます。列が両方のリストに含まれている場合は、分散統計が収集されます (**NUM\_FREQVALUES** および **NUM\_QUANTILES** がゼロに設定されていない限り)。

#### **LIKE STATISTICS**

このオプションを指定すると、コード・ページ属性が 1 バイト文字セット (SBCS)、FOR BIT DATA、または UTF-8 であるタイプ CHAR および VARCHAR の列について、追加の列統計が収集される場合があります。これらの統計が収集されるのは、**runstats** ユーティリティが列値を分析した後に、これらの統計が有効であると判断した場合です。これらの統計とは、SYSSTAT.COLUMNS の SUB\_COUNT および SUB\_DELIM\_LENGTH 統計です。照会オプションは、これらを使用して "column LIKE '%xyz'" や "column LIKE '%xyz%'" のようなタイプの述部に関する選択度の見積もりを向上させます。

#### **WITH DISTRIBUTION**

この節は、指定された列で基本統計と分散統計の両方を収集することを指定します。**ON COLUMNS** 節が指定されていない場合は、表または統計ビューの中のすべての列 (CLOB や LONG VARCHAR といった、収集用には選択できない列を除く) で分散統計が収集されます。一方 **ON COLUMNS** 節が指定されている場合は、提供された列リストに対してのみ (統計収集用には選択できない列を除く) 分散統計が収集されます。なお、節が指定されなければ、基本統計だけが収集されます。

列のグループに関する分散統計の収集は、現在サポートされていません。

**WITH DISTRIBUTION ON COLUMNS** 節で列のグループが指定された場合は、分散統計は収集されません。

#### **DEFAULT**

**NUM\_FREQVALUES** または **NUM\_QUANTILES** を指定した場合、これらの値を **ON COLUMNS** 節で個々の列に対して指定していなければ、これらの値によって、すべての列に関して収集する度数と分位の統計の最大個数が決まります。**DEFAULT** 節が指定されない場合は、対応するデータベース構成パラメーターにある値が使用されます。

#### **NUM\_FREQVALUES**

収集の頻度を示す値の最大値を定義します。これは、**ON COLUMNS** 節の中で、個々の列ごとに指定できます。個々の列に対して値が指定されない場合は、**DEFAULT** 節で指定されている頻度のしきい値が選出されます。どちらも指定されていない場合は、**num\_freqvalues** データベース構成パラメーターで設定されている値が、収集の頻度を指定する値の最大値になります。

#### **NUM\_QUANTILES**

収集する分散変位値の最大値を定義します。これは、**ON COLUMNS** 節の中

で、個々の列ごとに指定できます。個々の列に対して値が指定されない場合は、**DEFAULT** 節で指定されている変位値のしきい値が選出されます。どちらも指定されていない場合は、**num\_quantiles** データベース構成パラメーターで設定されている値が、収集する分位値の最大個数になります。

DB2 V9.7 フィックスパック 1 以降のリリースでは、XML データに対する各索引の分散統計は、デフォルトで最大 250 の分位数を使用します。このデフォルトは、**ON COLUMNS** 節または **DEFAULT** 節の **NUM\_QUANTILES** パラメーターを指定して変更することができます。XML 分散統計を収集している間、**num\_quantiles** データベース構成パラメーターは無視されます。

#### **ALLOW WRITE ACCESS**

統計が計算される間に、他のユーザーが表から読み込んだりそこに書き込んだりできることを指定します。統計ビューの場合は、ビュー定義の中で参照されている基本表です。

多くの挿入、更新、または削除が同時に実行される表の場合、**ALLOW WRITE ACCESS** オプションは勧められていません。**RUNSTATS** コマンドは、まず表統計を実行した後、索引統計を実行します。表および索引の統計情報を収集している間に表の状態が変化した場合、不整合が発生することがあります。照会の最適化のために最新の統計を収集することは重要ですが、整合性のある統計を収集することも重要です。したがって統計情報の収集は、挿入、更新、または削除の操作が最小になる時間に実行してください。

#### **ALLOW READ ACCESS**

統計が計算される間に、他のユーザーが表に対して読み取り専用のアクセスを行えることを指定します。統計ビューの場合は、ビュー定義の中で参照されている基本表です。

#### **TABLESAMPLE BERNOULLI**

このオプションを使用した **RUNSTATS** は、表または統計ビューから取られた行のサンプルに関する統計を収集します。**BERNOULLI** (ベルヌーイ) サンプリングでは各行が個別に処理され、その際に  $P/100$  ( $P$  は数値リテラル値) の確率で行が含まれ、 $1-P/100$  の確率で行が除外されます。例えば、数値リテラルが値 10 (つまり、10 % のサンプル) と評価された場合は、0.1 の確率で各行が含まれ、0.9 の確率で行が除外されます。オプションの **REPEATABLE** 節を指定しない限り、**RUNSTATS** を実行するたびに、通常は異なった表のサンプルが作成されます。すべてのデータ・ページが表スキャンによって検索されますが、数値リテラル・パラメーターによって指定したパーセンテージの行だけが、統計収集に使用されます。

#### **TABLESAMPLE SYSTEM**

このオプションを使用した **RUNSTATS** は、表から取られたデータ・ページのサンプルに関する統計を収集します。**SYSTEM** (システム) サンプリングでは各ページが個別に処理され、その際に  $P/100$  ( $P$  は数値リテラル値) の確率でページが含まれ、 $1-P/100$  の確率でページが除外されます。オプションの **REPEATABLE** 節を指定しない限り、**RUNSTATS** を実行するたびに、通常は異なった表のサンプルが作成されます。サンプルのサイズは、括弧内の数値リテラル・パラメーターによって制御し、表の約  $P$  % を戻すように指定します。数値リテラル・パラメーターによって指定したパーセンテージのデータ・ページだけが、検索されて統計収集に使用されます。

統計ビューでは、SYSTEM サンプルングは、ビュー定義で参照される単一の基本表にのみ適用できます。ビューに複数の表が含まれている場合、統計ビュー内のすべての表から、ある単一の表をそのビュー内で使用されている他の表の主キーまたはユニーク索引列すべてによって結合されているものとして特定できるのであれば、SYSTEM サンプルングが可能です。統計ビューがこれらの条件を満たしていない場合、BERNOULLI サンプルングが代わりに使用され、警告が戻されます。

#### *numeric-literal*

**numeric-literal** パラメーターでは、入手するサンプルのサイズを指定します ( $P$  %)。この値は 100 以下の正数でなければならず、1 と 0 の間の数を指定することもできます。例えば、値 0.01 は、1 % の 100 分の 1 を表します。この場合は、平均して 10,000 行のうちの 1 行がサンプルとして取られます。0 または 100 の値を指定した場合、DB2 データベース・システムでは、TABLESAMPLE BERNOULLI と TABLESAMPLE SYSTEM のどちらが指定されているかにかかわらず、サンプルングが指定されていない場合と同じように処理されます。100 より大きい値または 0 より小さい値は、DB2 データベース・システムではエラー (SQL1197N) として処理されます。

#### **REPEATABLE** (*integer-literal*)

**REPEATABLE** 節を TABLESAMPLE 節に追加すれば、RUNSTATS の反復実行時に必ず同じサンプルが戻されるようになります。 *integer-literal* パラメーターは、サンプルングで使用するシードを表す負以外の整数です。負のシードを渡した場合、エラー (SQL1197N) が発生します。 TABLESAMPLE REPEATABLE の最後の実行以降に行われた表または統計ビューに対する活動によって表または統計ビューのデータが変更された場合には、反復可能な RUNSTATS 呼び出しでサンプル・セットが変化する可能性があります。また、必ず整合した結果を得るためには、BERNOULLI (バルヌーイ) または SYSTEM (システム) キーワードによって指定するサンプルの入手方法が同じでなければなりません。

#### **INDEXSAMPLE BERNOULLI**

このオプションを使用すると、索引内の行のサンプルに対する索引統計が収集されます。BERNOULLI (バルヌーイ) サンプルング では各行が個別に処理され、その際に  $P/100$  ( $P$  は数値リテラル値) の確率で行が含まれ、 $1-P/100$  の確率で行が除外されます。例えば、数値リテラルが値 10 (つまり、10 % のサンプル) と評価された場合は、0.1 の確率で各行が含まれ、0.9 の確率で行が除外されます。 RUNSTATS は実行されるたびに、索引から異なるサンプルを得る結果になると考えられます。すべての索引ページが索引スキャンによって検索されますが、数値リテラル・パラメーターによって指定したパーセンテージの行だけが、統計収集に使用されます。このオプションは統計ビューではサポートされていません。

#### **INDEXSAMPLE SYSTEM**

このオプションを使用すると、索引ページのサンプルに対する統計が収集されます。SYSTEM (システム) サンプルング では各ページが個別に処理され、その際に  $P/100$  ( $P$  は数値リテラル値) の確率でページが含まれ、 $1-P/100$  の確率でページが除外されます。通常は、RUNSTATS コマンドが実行されるたびに索引から異なるサンプルを得る結果になります。サンプルのサイズは、括弧内の数値リテラル・パラメーターによって制御し、索引の約  $P$  % を戻すように指定します。数値リテラル・パラメーターによって指定

したパーセントの索引ページだけが、検索されて統計収集に使用されます。  
このオプションは統計ビューではサポートされていません。

#### SET PROFILE NONE

この **RUNSTATS** 呼び出しには統計プロファイルを設定しないことを指定します。

#### SET PROFILE

**RUNSTATS** は、特定の統計プロファイルを生成してシステム・カタログ表に保管し、**RUNSTATS** コマンド・オプションを実行して統計を収集します。

#### SET PROFILE ONLY

**RUNSTATS** は、特定の統計プロファイルを生成してシステム・カタログ表に保管しますが、**RUNSTATS** のコマンド・オプションを実行しません。

#### UPDATE PROFILE

**RUNSTATS** は、システム・カタログ表内の既存の統計プロファイルを変更し、その更新済みの統計プロファイルの **RUNSTATS** コマンド・オプションを実行して統計を収集します。UPDATE PROFILE オプションを使用して、統計プロファイルにある節を削除することはできません。

#### UPDATE PROFILE ONLY

**RUNSTATS** は、システム・カタログ表内の既存の統計プロファイルを変更しますが、その更新済みの統計プロファイルの **RUNSTATS** コマンド・オプションを実行しません。UPDATE PROFILE ONLY オプションを使用して、統計プロファイルにある節を削除することはできません。

#### UTIL\_IMPACT\_PRIORITY *priority*

*priority* に指定されているレベルで、**RUNSTATS** をスロットルすることを指定します。*priority* は 1 から 100 の範囲の数であり、100 が最高の優先順位、1 が最低の優先順位を表します。優先順位によって、ユーティリティのスロットルの量が決まります。優先順位が同じユーティリティはすべて同じ量のスロットルになり、優先順位の低いユーティリティは、優先順位の高いユーティリティよりも量が絞られます。*priority* を指定しない場合、**RUNSTATS** はデフォルトの優先順位 50 を使用します。

**UTIL\_IMPACT\_PRIORITY** キーワードを省略すると、スロットルのサポートなしで **RUNSTATS** ユーティリティが呼び出されます。

**UTIL\_IMPACT\_PRIORITY** キーワードを指定した場合でも、**util\_impact\_lim** 構成パラメーターが 100 に設定されていれば、スロットルなしでユーティリティが実行されます。

パーティション・データベースでは、**RUNSTATS** コマンドが表に対して使用された場合、1 つのデータベース・パーティションでしか統計を収集できません。**RUNSTATS** コマンドが実行されたデータベース・パーティションに表のパーティションがある場合、コマンドは、そのデータベース・パーティションで実行されます。それ以外の場合は、表がパーティションに分けられているデータベース・パーティション・グループの最初のデータベース・パーティションで実行されます。

#### 例

索引で使用されるすべての列と、すべての索引の統計を収集します。

```
CALL SYSPROC.ADMIN_CMD ('RUNSTATS ON TABLE employee  
ON KEY COLUMNS and INDEXES ALL')
```

## 使用上の注意

1. パーティション表にデタッチされたパーティションが存在する場合、デタッチされたデータ・パーティションでクリーンアップの必要なものにまだ属している索引キーは、統計においてキーの一部としてカウントされません。それらのキーは、不可視であり表の一部ではなくなっているため、カウントされません。そのような索引キーは、最終的に非同期の索引クリーンアップによって索引から除去されます。その結果、非同期の索引クリーンアップを実行する前に収集された統計は誤ったものとなります。非同期索引クリーンアップの完了前に **RUNSTATS** コマンドが発行された場合、不正確な統計情報を基に、索引再編成または索引クリーンアップに対する誤ったアラームが生成されることがしばしばあります。非同期の索引クリーンアップの実行を開始すると、クリーンアップを必要とするデタッチされたデータ・パーティションにまだ属しているすべての索引キーが除去されるので、索引の再編成の必要がなくなることもあります。

パーティション表の場合、非同期索引クリーンアップ完了後に **RUNSTATS** コマンドを発行することをお勧めします。それは、デタッチされているデータ・パーティションの存在に関して正確な索引統計情報を生成するためです。表の中にデタッチされているデータ・パーティションがあるかどうかを調べるには、**SYSCAT.DATAPARTITIONS** カタログ・ビューの状況フィールドを確認して、値 **L** (論理的にデタッチ済み)、**I** (索引クリーンアップ)、または **D** (デタッチ済みで従属 **MQT** 付き) を探してください。

**RUNSTATS** コマンドは、パーティション索引のすべての索引パーティションの統計を収集します。パーティション索引の **SYSSTAT.INDEXES** ビュー内の統計は、索引パーティションを表します。ただし

**FIRSTKEYCARD**、**FIRST2KEYCARD**、**FIRST3KEYCARD**、**FIRST4KEYCARD**、および **FULLKEYCARD** 統計は除きます。これらの統計はカーディナリティーの見積もりで使用されるので、それらは索引全体に対するものであり、1 つの索引パーティションに対するものではありません。分散統計 (頻度および変位値) は、パーティション索引では収集されませんが、**RUNSTATS** が表上で実行される場合は収集されます。パーティション索引の先行列の統計は、非パーティション索引の先行列の統計ほど正確でない場合があります。

2. コマンドの実行状況は、**CALL** ステートメントからの結果である **SQLCA** で戻されます。
3. **RUNSTATS** コマンドは、次のような場合に実行することが勧められています。
  - 表が大幅に変更されている場合 (例えば、多くの変更が行われている場合や、大量のデータが挿入または削除されている場合、あるいは、**LOAD** 時に統計オプションを指定しないで **LOAD**が行われた場合など)。
  - 表が再編成されている場合 (**REORG**、**REDISTRIBUTE DATABASE PARTITION GROUP** を使用)。
  - 行圧縮が実行された表の場合。
  - 新しい索引が作成されている場合
  - パフォーマンスが重要な意味を持つアプリケーションのバインドの前。
  - プリフェッチ・サイズが変更された場合。

- 基礎表に実質的な変更が加えられたことにより、ビューによって戻される行が変更された統計ビューの場合。
  - **STATISTICS** オプションを指定した **LOAD** が実行された後には、**RUNSTATS** ユーティリティを使用して、XML 列に関する統計情報を収集してください。**LOAD** 実行時には、**LOAD** に **STATISTICS** オプションを指定して実行された場合であっても、XML 列に関する統計情報が収集されることは決してありません。**RUNSTATS** を使用することにより、XML 列についてのみの統計情報を収集する場合、XML 以外の列に関して **LOAD** またはそれ以前に実行された **RUNSTATS** ユーティリティによって収集された既存の統計情報は、そのまま保持されます。以前に一部の XML 列に関する統計情報が収集されていた場合、ある XML 列に関する統計情報が現在のコマンドでは収集されないのであれば、その XML 列に関して以前に収集された統計情報はドロップされます。あるいは、その XML 列に関する統計が現在のコマンドで収集されるのであれば、置き換えられます。
4. オプションの選択は、特定の表やアプリケーションに合わせて行う必要があります。一般的なヒントとして、以下の点を考慮してください。
- 重要な照会に使用される非常に重要な表、比較的小規模な表、またはあまり変化がなく、システムそのものでの活動があまりない表には、可能な限り詳細に統計を収集する努力を費やす価値があります。
  - 統計を収集する時間が限られている場合、表が比較的大規模な場合、または表が頻繁に更新される場合には、述部で使用される列セットに限って **RUNSTATS** を実行するのも良い方法かもしれません。このような方法を使用する場合には、より頻繁に **RUNSTATS** コマンドを実行できるでしょう。
  - 統計を収集する時間が極めて限られており、表ごとに表の **RUNSTATS** コマンドを調整するのが時間の面で大きな問題となっている場合は、"KEY" 列だけの統計を収集することも考慮してください。索引に含まれている列セットは、表にとって重要で、述部に使用される確立が最も高いと考えられます。
  - 統計を収集する時間が非常に限られている状況で表統計を収集する場合は、**TABLESAMPLE** オプションを使用して、表データのサブセットに関する統計を収集することを検討してください。
  - 統計を収集する時間が非常に限られている状況で索引統計を収集する場合は、**INDEXSAMPLE** オプションを使用して、索引データのサブセットに関する統計を収集することを検討してください。
  - 特定の列にスキューがあり、述部のタイプが "column = constant" である場合、その列にはより大きな **NUM\_FREQVALUES** 値を指定するほうが良い可能性があります。
  - 等式の述部で使用される列や、値の分散がスキューされる可能性のある列では、必ず分散統計を収集してください。
  - 範囲の述部を持つ列 (例えば、"column >= constant" や "column BETWEEN constant1 AND constant2" など) や、タイプ "column LIKE '%xyz'" の列では、**NUM\_QUANTILES** に指定する値を大きくしたほうが有益な場合があります。
  - ストレージ・スペースが関係している場合で、統計の収集にあまり時間をかけられない場合は、述部で使用されない列の **NUM\_FREQVALUES** 値や **NUM\_QUANTILES** 値をあまり高くしないでください。

- 索引統計を要求したときに、索引を含む表についての統計がそれまで実行されていなかった場合、表と索引の両方に関する統計が計算されます。
  - 表に含まれる XML 列に関する統計情報が必要ない場合は、**EXCLUDING XML COLUMNS** オプションを使用することによって、XML 列をすべて除外することができます。このオプションは、統計情報収集の対象として XML 列を指定する他のどの節よりも優先されます。
5. コマンドを実行した後は、以下の点に注意してください。
- ロックを解除するには、**COMMIT** を発行する必要があります。
  - 新しいアクセス・プランを生成できるようにするには、ターゲット表を参照するパッケージを再バインドする必要があります。
  - 表で部分的にコマンドを実行すると、コマンドが最後に実行されてからの表での活動の結果として、不整合が生じる可能性があります。このような場合には、警告メッセージが戻されます。表でだけ **RUNSTATS** が実行されると、表レベルの統計と索引レベルの統計に不整合が生じます。例えば、ある表に関して索引レベルの統計を収集した後で、その表からかなりの数の行を削除してしまったとします。このような場合に、その表でだけ **RUNSTATS** を発行すると、表のカーディナリティーが **FIRSTKEYCARD** よりも小さくなってしまふ可能性があります。これは不整合です。これと同様に、作成した新しい索引で統計を収集した場合にも、表レベルの統計に不整合が生じることがあります。
6. **RUNSTATS** コマンドは、表統計が要求したときに、以前に収集された分散統計をドロップします。例えば、**RUNSTATS ON TABLE** または **RUNSTATS ON TABLE ... AND INDEXES ALL** は、以前に収集された分散統計がドロップされる原因になります。コマンドが索引でのみ実行される場合、以前に収集された分散統計は保持されます。例えば、**RUNSTATS ON TABLE ... FOR INDEXES ALL** は、以前に収集された分散統計が保持される原因になります。**RUNSTATS** コマンドが XML 列に対してのみ実行される場合、それ以前に収集された基本列統計および分散統計はそのまま保持されます。以前に一部の XML 列に関する統計情報が収集されていた場合、ある XML 列に関する統計情報が現在のコマンドでは収集されないのであれば、その XML 列に関して以前に収集された統計情報はドロップされます。あるいは、その XML 列に関する統計が現在のコマンドで収集されるのであれば、置き換えられます。
7. DB2 バージョン 9.7 フィックスパック 1 以降のリリースでは、XML 列に定義された XML データに対する索引で、分散統計が収集されます。表に対して、**RUNSTATS** コマンドを **WITH DISTRIBUTION** 節付きで実行する場合、タイプ XML の列に関する分散統計の収集は以下のようになります。
- 分散統計は、XML 列で指定されている XML データに対する各索引に関して収集されます。
  - **RUNSTATS** コマンドでは、XML 列で定義された XML データに対する索引に関して分散統計を収集するため、分散統計と表統計の両方を収集する必要があります。XML 分散統計は表統計と共に保管されるので、分散統計を収集するためには表統計を収集しなければなりません。
- XML 分散統計を収集するために、索引節は必要ありません。索引節だけを指定しても、XML 分散統計は収集されません。

XML 分散統計はデフォルトで、XML データに対する各索引に関して最大 250 の分位数を使用します。XML 列に対する分散統計を収集する場合、**ON COLUMNS** 節または **DEFAULT** 節の **NUM\_QUANTILES** パラメーターに値を指定することによって分位の最大数を変更することができます。

- 分散統計は、タイプ **VARCHAR**、**DOUBLE**、**TIMESTAMP**、および **DATE** の XML データの索引に関して収集されます。タイプ **VARCHAR HASHED** の索引に関して、分散統計は収集されません。
  - 分散統計は、パーティション表で定義された XML データに対するパーティション索引に関しては収集されません。
8. 範囲がクラスター化された表の場合、範囲がクラスター化された表の範囲配列プロパティーを表す特殊なシステム生成索引がカタログ表内に存在します。この種の表の統計を収集するときに、統計収集の一部として表を組み込む場合は、システム生成索引用の統計も収集されます。この統計は、基本データ表と同じページ数を持つ 2 レベルの索引として索引を表現し、索引の順序に沿って完全に基本データをクラスター化することによって、範囲検索の高速アクセスを反映することになっています。
  9. コマンド構文の **On Dist Cols** 節では、列 **GROUPS** に対する **Frequency Option** および **Quantile Option** のパラメーターの使用は、現在サポートされていません。これらのオプションは、単一の列でのみサポートされています。
  10. **DMS** モードでの作業中に、計算できない 3 つのプリフェッチ統計があります。索引カタログ内の索引統計で、以下の統計の値は -1 になります。
    - **AVERAGE\_SEQUENCE\_FETCH\_PAGES**
    - **AVERAGE\_SEQUENCE\_FETCH\_GAP**
    - **AVERAGE\_RANDOM\_FETCH\_PAGES**
  11. プロファイルの設定またはプロファイルの更新のオプションを使用することによって、**RUNSTATS** コマンドで指定する表または統計ビューの統計プロファイルを設定または更新できます。統計プロファイルは、**STATISTICS\_PROFILE** システム・カタログ表の **SYSCAT.TABLES** 列に、可視ストリングのフォーマットで保管されます。これが **RUNSTATS** コマンドに相当します。
  12. XML タイプの列に関する統計情報収集は、**DB2\_XML\_RUNSTATS\_PATHID\_K** および **DB2\_XML\_RUNSTATS\_PATHVALUE\_K** の 2 つの **DB2** データベース・システム・レジストリー値によって制御されます。これらの 2 つのパラメーターは、収集する度数の数を指定するという点において **NUM\_FREQVALUES** パラメーターに似ています。設定されていない場合、どちらのパラメーターについてもデフォルトとして 200 が使用されます。
  13. **RUNSTATS** は、開始時に、**SYSTABLES** に対して **IX** 表ロックを取得し、統計収集の対象となる表の行に対して **U** ロックを取得します。**U** ロックのかかった行を含め、**SYSTABLES** からの読み取り操作は可能です。また、**U** ロックのかかった行でない限り、書き込み操作も可能です。しかし、**RUNSTATS** が **IX** ロックを取得しているため、別の読み取りプログラムまたは書き込みプログラムが **SYSTABLES** に対する **S** ロックを取得することはできません。
  14. 統計は、構造化タイプの列については収集されません。それらが指定されている場合、そのデータ・タイプの列は無視されます。
  15. **LOB** または **LONG** データ・タイプの列では、**AVGCOLLEN** および **NUMNULLS** のみが収集されます。



16. AVGCOLLEN は、列がデータベース・メモリーまたは一時表に保管される場合の、バイト単位の平均スペースを表します。この値は、LOB または LONG データ・タイプのデータ記述子の長さを表します。ただし LOB データがデータ・ページ上でインライン化されている場合を除きます。

注: ディスク上で列を保管するための平均スペース所要量は、この統計により表される値とは異なる場合があります。

17. 索引統計の収集方法を変更するため、**UNSAMPLED DETAILED** オプションを使用できます。ただし、これを使用するのは、デフォルトと **DETAILED** では適正に機能しないことが明らかな場合に限るべきです。

18. **INDEXSAMPLE** キーワードを使用するときは、単一のコマンド内で索引に対する索引サンプリング率を別々に指定することはできません。例:

```
runstats on table orders and index o_ck indexsample system(5),
index o_ok indexsample system(10)
```

この指定は無効です。必要な結果を得るには、以下の 2 つの **RUNSTATS** コマンドを使用できます。

```
runstats on table orders and index o_ck indexsample system(5)
runstats on table orders for index o_ok indexsample system(10)
```

#### SET TAPE POSITION コマンド (ADMIN\_CMD プロシージャーを使用):

ストリーミング磁気テープ装置へのバックアップおよびリストア操作のためにテープの位置を設定します。このコマンドは Windows オペレーティング・システムでのみサポートされています。

#### 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMAINT

#### 必要な接続

データベース

#### コマンド構文

```
▶▶ SET TAPE POSITION [ON device] TO position ▶▶
```

#### コマンド・パラメーター

##### ON device

有効なテープ装置名を指定します。デフォルト値は ¥\$.¥TAPE0 です。装置の指定は、サーバーに対する相対指定でなければなりません。

##### TO position

テープ位置のマークを指定します。DB2 (Windows 版) は、バックアップ・イメージの度にテープ・マークを書き込みます。値 1 は 1 番目の位

置、2 は 2 番目の位置、以下同じ手順で指定します。テープがテープ・マーク 1 に位置している場合、例えば、アーカイブ 2 がリストアされる位置に置かれます。

## 例

DB2 データベースは各バックアップ・イメージの後にテープ・マークを書き込むため、位置として 1 を指定すると、テープ上の 2 番目のアーカイブの開始位置にテープが移動します。

```
CALL SYSPROC.ADMIN_CMD( 'set tape position to 1' )
```

## 使用上の注意

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

## UNQUIESCE DATABASE コマンド (ADMIN\_CMD プロシージャを使用):

保守などの理由で静止状態になっていたデータベースに対するユーザー・アクセスを回復します。**UNQUIESCE** コマンドにより、シャットダウンおよびデータベースの再始動をしなくても、ユーザー・アクセスを回復できます。

## 有効範囲

**UNQUIESCE DB** は、静止データベース内のすべてのオブジェクトに対するユーザー・アクセスを回復します。

インスタンスを停止した後、そのインスタンスとそのすべてのデータベースの静止を解除するには、**db2stop** コマンドを発行します。DB2 を停止し、再開すると、すべてのインスタンスとデータベースの静止が解除されます。

## 許可

以下の権限のいずれか。

データベース・レベルの静止解除の場合:

- SYSADM
- DBADM

## コマンド構文

▶▶—UNQUIESCE—DB—▶▶

## 必要な接続

データベース

## コマンド・パラメーター

**DB** データベースの静止解除。データベース内のすべてのオブジェクトに対するユーザー・アクセスが回復されます。

## 例：データベースの静止解除

次のコマンドは、以前に静止されていたデータベースの静止を解除します。

```
CALL SYSPROC.ADMIN_CMD( 'unquiesce db' )
```

次のコマンドは、以前に静止されていたインスタンス `instA` の静止を解除します。

```
db2 unquiesce instance instA
```

### 使用上の注意

- コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。
- DB2 pureScale 環境では、データベースの静止およびインスタンスの再始動の後、データベースはすべてのメンバーにわたり静止状態のままになります。静止状態を解除するには、明示的な **UNQUIESCE DATABASE** コマンド呼び出しが必要です。

### UPDATE ALERT CONFIGURATION コマンド (ADMIN\_CMD プロシージャーを使用):

ヘルス・インディケーターのアラート構成設定を更新します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、このコマンドおよび API は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

### 許可

以下の権限のいずれか。

- SYSADM
- SYSMANT
- SYSCTRL

### 必要な接続

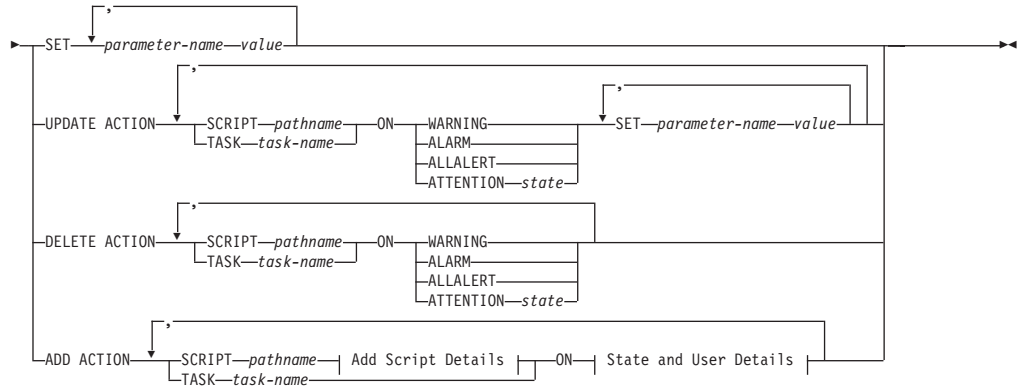
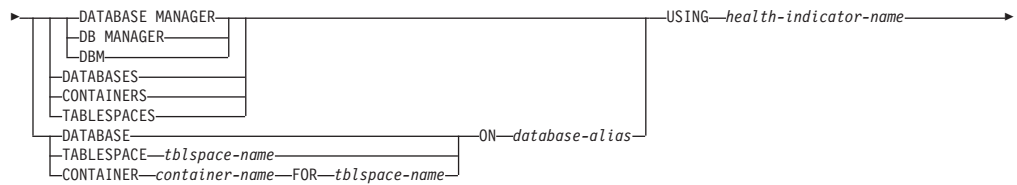
データベース

### コマンド構文

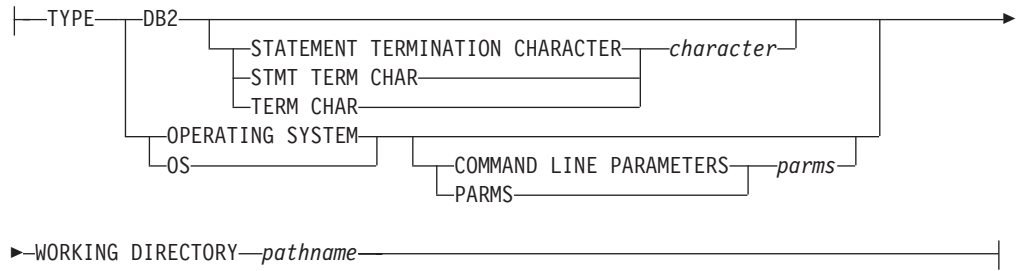
```
►► UPDATE ALERT CONFIGURATION FOR
```

CONFIG
CFG

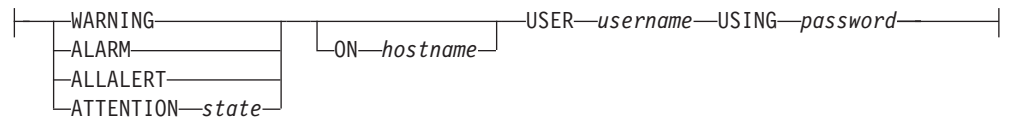
→



**Add Script Details:**



**State and User Details:**



**コマンド・パラメーター**

**DATABASE MANAGER**

データベース・マネージャーのアラート設定を更新します。

**DATABASES**

データベース・マネージャーによって管理されるすべてのデータベースのアラート設定を更新します。これは、カスタム設定を持たないすべてのデータベースに適用される設定です。カスタム設定は、**DATABASE ON database-alias** 節を使って定義されます。

**CONTAINERS**

データベース・マネージャーによって管理されるすべての表スペース・コンテナのアラート設定を更新します。これは、カスタム設定を持たないすべ

ての表スペース・コンテナに適用される設定です。カスタム設定は、**CONTAINER** *container-name* **ON** *database-alias* 節を使って定義されます。

#### **TABLESPACES**

データベース・マネージャーによって管理されるすべての表スペースのアラート設定を更新します。これは、カスタム設定を持たないすべての表スペースに適用される設定です。カスタム設定は、**TABLESPACE** *tblspace-name* **ON** *database-alias* 節を使って定義されます。

#### **DATABASE ON** *database-alias*

**ON** *database-alias* 節を使って指定したデータベースのアラート設定を更新します。このデータベースがカスタム設定を持つ場合、インスタンスの全データベースの設定をオーバーライドします。これは、**DATABASES** パラメーターを使って指定されます。

#### **CONTAINER** *container-name* **FOR** *tblspace-name* **ON** *database-alias*

**ON** *database-alias* 節を使って指定したデータベース上で、**FOR** *tblspace-name* 節を使って指定した表スペースの、*container-name* という名前の表スペース・コンテナのアラート設定を更新します。この表スペース・コンテナがカスタム設定を持つ場合、データベースの全表スペース・コンテナの設定をオーバーライドします。これは、**CONTAINERS** パラメーターを使って指定されます。

#### **TABLESPACE** *tblspace-name* **ON** *database-alias*

**ON** *database-alias* 節を使って指定したデータベース上で、*name* という名前の表スペースのアラート設定を更新します。この表スペースがカスタム設定を持つ場合、データベースの全表スペースの設定をオーバーライドします。これは、**TABLESPACES** パラメーターを使って指定されます。

#### **USING** *health-indicator-name*

アラート構成が更新されるヘルス・インディケーターのセットを指定します。ヘルス・インディケーター名は 2 文字のオブジェクト ID で構成され、その後にインディケーターの測定対象を説明する名前が続きます。以下に例を示します。

```
db.sort_privmem_util
```

#### **SET** *parameter-name* *value*

ヘルス・インディケーターのアラート構成エレメント *parameter-name* を、指定した値に更新します。 *parameter-name* は以下の値のどれかになります。

- **ALARM:** *value* はヘルス・インディケーター・ユニット。
- **WARNING:** *value* はヘルス・インディケーター・ユニット。
- **SENSITIVITY:** *value* は秒単位。
- **ACTIONSENABLED:** *value* は YES または NO。
- **THRESHOLDSCHECKED:** *value* は YES または NO。

特定の DB2 バージョンについて可能性のあるヘルス・インディケーター・ユニットのリストは、以下の照会を実行することにより収集できます。

```
SELECT SUBSTR(UNIT,1,80) AS UNIT  
FROM TABLE(HEALTH_GET_IND_DEFINITION('')) AS T GROUP BY UNIT
```

**UPDATE ACTION SCRIPT** *pathname* ON [WARNING | ALARM | ALLALERT | ATTENTION *state*] 絶対パス名 *pathname* を持つ定義済みスクリプトのスクリプト属性が以下の節に従って更新されるように指定します。

**SET** *parameter-name value*

スクリプト属性 *parameter-name* を、指定した値に更新します。  
*parameter-name* は以下の値のどれかになります。

- SCRIPTTYPE

有効なタイプは OS または DB2。

- WORKINGDIR

- TERMCHAR

- CMDLINEPARMS

オペレーティング・システム・スクリプトに対して指定するコマンド行パラメーターが、デフォルトで指定されるパラメーターに先行します。オペレーティング・システム・スクリプトに送られるパラメーターは、以下のとおりです。

- ユーザーの指定するパラメーターのリスト

- ヘルス・インディケーターの短縮名

- 完全修飾オブジェクト名

- ヘルス・インディケーターの値

- アラート状態

- USERID

- PASSWORD

- SYSTEM

**UPDATE ACTION TASK** *task-name* ON [WARNING | ALARM | ALLALERT | ATTENTION *state*] 名前 *name* を持つタスクのタスク属性が以下の節に従って更新されるように指定します。

**SET** *parameter-name value*

タスク属性 *parameter-name* を、指定した値に更新します。  
*parameter-name* は以下の値のどれかになります。

- USERID

- PASSWORD

- SYSTEM

**DELETE ACTION SCRIPT** *pathname* ON [WARNING | ALARM | ALLALERT | ATTENTION *state*] アラート・アクション・スクリプトから、絶対パス名 *pathname* を持つアクション・スクリプトを除去します。

**DELETE ACTION TASK** *task-name* ON [WARNING | ALARM | ALLALERT | ATTENTION *state*] アラート・アクション・タスクのリストから *name* という名前のアクション・タスクを除去します。

**ADD ACTION SCRIPT** *pathname* ON [WARNING | ALARM | ALLALERT | ATTENTION *state*] 絶対パス名 *pathname* を持つ新規アクション・スクリプトが追加されるように指定します。その属性は、以下のように指定されます。

**TYPE** アクション・スクリプトは、DB2 コマンド・スクリプトか、オペレーティング・システム・スクリプトのいずれかでなければなりません。

- DB2
- OPERATING SYSTEM

DB2 コマンド・スクリプトの場合、以下の節を使用することにより、オプションで文字 *character* を指定することができます。この文字は、ステートメントを終了するのにスクリプト内で使用されます。

STATEMENT TERMINATION CHARACTER ;

オペレーティング・システム・スクリプトの場合、以下の節を使用することにより、オプションでコマンド行パラメーター *parms* を指定することができます。これは、呼び出しの際にスクリプトに渡されます。 **COMMAND LINE PARAMETERS** *parms*

**WORKING DIRECTORY** *pathname*

スクリプトが実行されるディレクトリーの絶対パス名 *pathname* を指定します。

**USER** *username* **USING** *password*

スクリプトが実行される際のユーザー・アカウント *username*、およびそれに関連したパスワード *password* を指定します。 **ADD ACTION** オプションが使用されている場合、*username* および *password* が、ネットワーク (*username* および *password* が暗号化されないで送信される) において、**db2diag** ログ・ファイル、トレース・ファイル、ダンプ・ファイル、スナップショット・モニター (動的 SQL スナップショット)、システム・モニター・スナップショット、いくつかのイベント・モニター (ステートメントやデッドロックなど)、**Explain** 表、**db2pd** 出力 (パッケージ・キャッシュやロック・タイムアウト・メカニズムなど)、および DB2 監査レコードで露出する可能性があります。

**ADD ACTION TASK** *name* **ON** [**WARNING** | **ALARM** | **ALLALERT** | **ATTENTION** *state*]

指定した条件を **ON** にして *name* という新規タスクを追加し、実行することを指定します。

**ON** [**WARNING** | **ALARM** | **ALLALERT** | **ATTENTION** *state*]

アクションまたはタスクが実行される条件を指定します。しきい値ベースのヘルス・インディケーター (HI) の場合、これは **WARNING** または **ALARM** になります。状態ベースの HI の場合、これは、各状態ベースの HI ごとに記されている数値状態 (例えば、*ts.ts\_op\_status* ヘルス・インディケーターの場合は、表スペース状態の **tablespace\_state** モニター・エレメントを参照してください) か、この状態を表すテキストの ID となります。**ALLALERTS** は、しきい値ベースの HI と状態ベースの HI に関する状態の変更を処理します (例えば、警告から正常への状態変更)。

**ATTENTION** *state*

**ADD ACTION SCRIPT** **CLP** コマンド・オプションの例として、データベース・ヘルス・インディケーターのいくつかの状態に対応した有効な数値を以下のセクションにまとめます。

- 0 - アクティブ、正常 (ACTIVE)
- 1 - 静止ペンディング (QUIESCE\_PEND)
- 2 - 静止済み (QUIESCED)
- 3 - ロールフォワード (ROLLFWD)

その他の状態ベースのヘルス・インディケータは、ヘッダー・ファイル `sqlmon.h` と `sqlutil.h` に定義されています。

ADMIN\_CMD ストアード・プロシージャにより呼び出される **UPDATE ALERT CFG** コマンドは、*state* において数値またはテキスト ID をサポートします。表スペース操作状況のヘルス・インディケータ (`ts.ts_op_status`) の例として、ヘルス・インディケータの一部の追加の状態の有効な数値およびテキスト ID は、次のようになります。

- 0x1 - QUIESCED\_SHARE
- 0x2 - QUIESCED\_UPDATE
- 0x4 - QUIESCED\_EXCLUSIVE

**UPDATE ALERT CFG** コマンドと上記のヘルス・インディケータ値を使用して、以下のコマンド行を入力するとします。

```
ADD ACTION SCRIPT ... ON ATTENTION 2
```

これは、以下と同じ意味になります。

```
ADD ACTION SCRIPT ... ON ATTENTION QUIESCED_UPDATE
```

さらに、表スペース操作状況のヘルス・インディケータ (`ts.ts_op_status`) では、複数の状態を OR 操作することにより、単一の数値を使用して複数の状態を指定できます。例えば、状態 7 (= 0x1 + 0x2 + 0x4) を指定すると、表スペースが、静止: SHARE、静止: UPDATE、または静止: EXCLUSIVE のいずれかの状態になったときにアクションが実行されます。その代わりに、3 つの別個の **UPDATE ALERT CFG** コマンド実行で、QUIESCED\_SHARE、QUIESCED\_UPDATE、および QUIESCED\_EXCLUSIVE を指定することもできます。

## 例

ホスト名 'plato' のシステムでアラームがあった場合に、スクリプト `/home/test/scripts/logfsutilact` を実行するという、`db.log_fs_util` インディケータのアクションを追加します。

```
CALL SYSPROC.ADMIN_CMD( 'update alert cfg for databases using
db.log_fs_util add action script /home/test/scripts/logfsutilact
type os command line parameters "param1 param2" working
directory /tmp on alarm on plato user dricard using mypasswdv' )
```

アラート構成が設定された後、それをチェックするには、以下のようにして `HEALTH_GET_IND_DEFINITION` および `HEALTH_GET_ALERT_ACTION_CFG` の表関数を使用することができます。



```
SELECT OBJECTTYPE, ID, CONDITION, ACTIONTYPE,
       SUBSTR(ACTIONNAME,1,50) AS ACTION_NAME
FROM TABLE(SYSPROC.HEALTH_GET_ALERT_ACTION_CFG('DB','G','',''))
AS ALERT_ACTION_CFG
```

以下はこの照会の出力例です。

OBJECTTYPE	ID	CONDITION	ACTIONTYPE	ACTION_NAME
DB	1006	ALARM	S	/home/dricard/scripts/logfsutilact

1 record(s) selected.

### 使用上の注意

**ADD ACTION** オプションでは、入力した *username* および *password* は、SQL ステートメント・テキストがキャプチャーされる以下のさまざまな場所で公開される場合があります。

- ネットワーク (*username* と *password* は暗号化されずにワイヤーを經由して渡されます)
- **db2diag** ログ・ファイル
- トレース・ファイル
- ダンプ・ファイル
- スナップショット・モニター (動的 SQL スナップショット)
- システム・モニター・スナップショット
- いくつかのイベント・モニター (ステートメント、デッドロック)
- Explain 表
- **db2pd** 出力 (パッケージ・キャッシュおよびロック・タイムアウト機構など)
- DB2 監査レコード

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

*database-alias* は、サーバー上のカタログの中で定義されていなければならない、サーバーに対してローカルなものでなければなりません。

*pathname* は、完全修飾サーバー・パス名を使用して指定する必要があります。

### UPDATE CONTACT コマンド (ADMIN\_CMD プロシージャを使用):

ローカル・システムで定義される連絡先の属性を更新します。連絡先とは、スケジューラーおよびヘルス・モニターがメッセージを送信する先のユーザーです。

連絡先を作成するには、**ADD CONTACT** コマンドを使用します。Database Administration Server (DAS) の **contact\_host** 構成パラメーターの設定により、リストがローカルかグローバルかが決まります。

### 許可

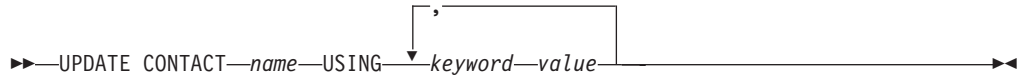
なし

## 必要な接続

データベース。 DAS が実行中でなければなりません。

## コマンド構文

```
►► UPDATE CONTACT name USING keyword value ◄◄
```



## コマンド・パラメーター

### UPDATE CONTACT *name*

更新される連絡先の名前。

### USING *keyword value*

更新される連絡先パラメーター (*keyword*) および設定される値 (*value*) を指定します。有効なキーワードのセットは次のとおりです。

#### ADDRESS

SMTP サーバーが通知を送信するのに使用する E メール・アドレス。

**TYPE** アドレスが E メール・アドレスか、ページャーかを指定します。

#### MAXPAGELEN

ページャーが受信できる最大文字数。

#### DESCRIPTION

連絡先のテキスト記述。長さは、最大 128 文字です。

## 例

ユーザー 'test' のアドレスを 'newaddress@test.com' に更新します。

```
CALL SYSPROC.ADMIN_CMD( 'update contact test using address newaddress@test.com' )
```

## 使用上の注意

DAS が作成されていて実行中でなければなりません。

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

### UPDATE CONTACTGROUP コマンド (ADMIN\_CMD プロシージャを使用):

ローカル・システムで定義される連絡先グループの属性を更新します。連絡先グループは、スケジューラーおよびヘルス・モニターから通知を受け取るユーザーのリストです。

Database Administration Server (DAS) の **contact\_host** 構成パラメーターの設定により、リストがローカルかグローバルかが決まります。

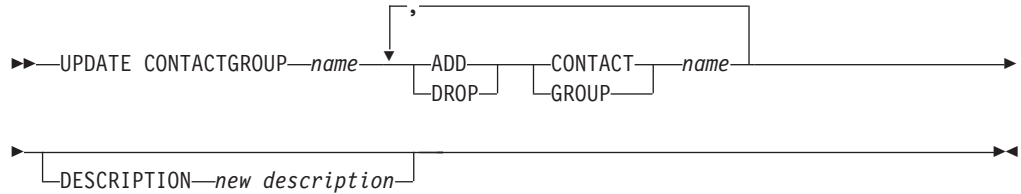
## 許可

なし

## 必要な接続

データベース。 DAS が実行中でなければなりません。

## コマンド構文



## コマンド・パラメーター

### CONTACTGROUP *name*

更新される連絡先グループの名前。

### ADD CONTACT *name*

グループに追加される新しい連絡先の名前を指定します。グループに追加された後、**ADD CONTACT** コマンドを使用して連絡先を定義できます。

### DROP CONTACT *name*

グループからドロップされる、グループ中の連絡先の名前を指定します。

### ADD GROUP *name*

グループに追加される新しい連絡先グループの名前を指定します。

### DROP GROUP *name*

グループからドロップされる、連絡先グループの名前を指定します。

### DESCRIPTION *new description*

オプション。連絡先グループの新しいテキスト記述。

## 例

gname1 という連絡先グループに cname2 という連絡先を追加します。

```
CALL SYSPROC.ADMIN_CMD( 'update contactgroup gname1 add contact cname2' )
```

## 使用上の注意

DAS が作成されていて実行中でなければなりません。

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

## UPDATE DATABASE CONFIGURATION コマンド (ADMIN\_CMD プロシージャーを使用):

特定のデータベース構成ファイルの中の個々の項目を修正します。データベース構成ファイルは、データベースが作成されたデータベース・パーティションすべてに存在しています。

## 有効範囲

このコマンドはデフォルトですべてのデータベース・パーティションまたはメンバーを更新します。ただし、以下のオプションの節が指定されている場合は除きます。

- **MEMBER** を指定して、DB2 pureScale 環境の 1 つのデータベース・メンバーのみを更新する場合、またはパーティション・データベース環境の 1 つのデータベース・パーティションのみを更新する場合。

## 許可

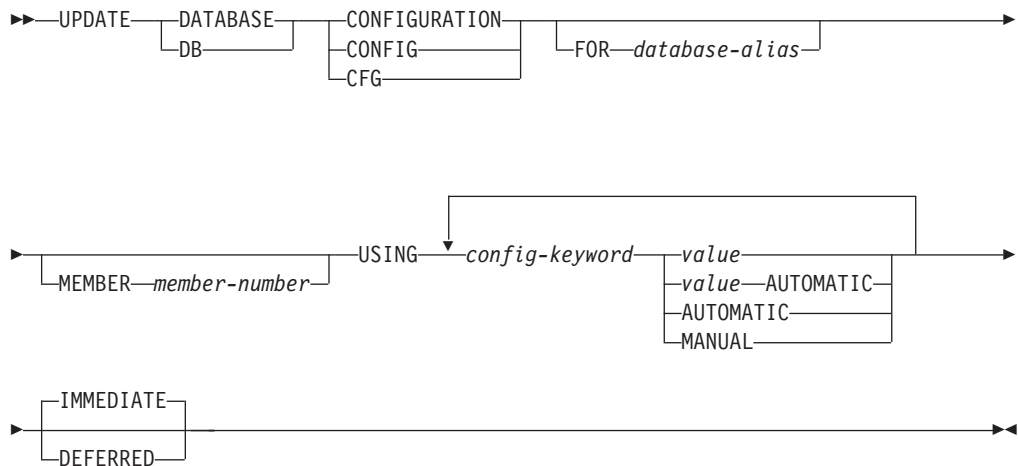
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMMAINT

## 必要な接続

データベース。データベース接続は、接続されているデータベースが含まれているインスタンスに対してローカルなものでなければなりません。

## コマンド構文



## コマンド・パラメーター

### **FOR** *database-alias*

構成を更新するデータベースの別名を指定します。データベース接続が既に確立されている場合は、データベース別名を指定する必要はありません。データベース別名は、サーバー上でローカルに定義されていなければなりません。同じデータベース・インスタンスにある別のデータベースについては、その構成ファイルを更新できます。例えば、データベース `db11` にのみ接続されている場合に `update db config for alias db22 using .... immediate` を発行すると、

- db22 上にアクティブな接続がない場合、更新の必要なのは構成ファイルだけであるため、更新は成功します。新しい接続 (それによりデータベースがアクティブになる) により、メモリー内で新しい変更が認識されるようになります。
- db22 上に他のアプリケーションからのアクティブな接続があるなら、更新はディスク上で動作しますが、メモリー内では動作しません。データベースを再始動する必要があることを示す警告を受け取るようになります。

#### **MEMBER** *member-number*

**MEMBER** 節では、変更適用の対象にするメンバーを指定します。この節を省略すると、結果としてすべてのメンバーに対して変更が適用されることになります。

#### **USING** *config-keyword value*

*config-keyword* は、更新するデータベース構成パラメーターを指定します。*value* は、パラメーターに割り当てる値を指定します。

#### **AUTOMATIC**

いくつかの構成パラメーターは **AUTOMATIC** に設定できます。それにより、DB2 データベース・システムが自動的にこれらのパラメーターを調整し、現行のリソース要件を反映します。 **AUTOMATIC** キーワードをサポートする構成パラメーターのリストは、構成パラメーターのサマリーを参照してください。 **AUTOMATIC** キーワードと一緒に値を指定すると、自動計算に影響する可能性があります。この動作の具体的な詳細については、構成パラメーターの資料を参照してください。

**注:** **appl\_memory**、**logindexbuild**、**max\_log**、**num\_log\_span** のデータベース構成パラメーターを **AUTOMATIC** に設定できるのは、コマンド行プロセッサを使用する場合に限られます。

**MANUAL** 構成パラメーターの自動チューニングを使用不可にします。パラメーターはその現行の内部値に設定され、自動的な更新は行われなくなります。

#### **IMMEDIATE**

データベースが稼働している場合に、即時に変更を行います。 **IMMEDIATE** はデフォルトのアクションです。 **ADMIN\_CMD** プロシージャにはデータベース接続が必要であるため、接続されているデータベースの動的構成可能パラメーターについては、即時に変更が有効になります。

これは、CLPPlus インターフェースで操作する場合のデフォルトの節でもあります。 **IMMEDIATE** は、CLPPlus プロセッサの使用時に呼び出す必要はありません。

#### **DEFERRED**

構成ファイルでのみ変更を行います。したがって、加えられた変更は、次にデータベースが再活動化されるときに有効になります。

#### **例**

データベース構成パラメーター **sortheap** を、アプリケーションが現在接続しているデータベース・パーティションに対して 1000 の値に設定します。

```
CALL SYSPROC.ADMIN_CMD ('UPDATE DB CFG USING sortheap 1000')
```

## 使用上の注意

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

*database-alias* は、サーバー上で定義されている別名でなければなりません。

このコマンドは、**MEMBER** が指定されていなければ、すべてのデータベース・パーティションに影響します。

データベース構成パラメーターのリストを表示または印刷するには、SYSIBMADM.DBCFG 管理ビューを使用してください。

すべてのデータベース構成パラメーターを推奨されているデフォルトにリセットするには、ADMIN\_CMD プロシージャーを使用し、**RESET DATABASE CONFIGURATION** コマンドを使用してください。

データベース構成パラメーターを変更するには、ADMIN\_CMD プロシージャーを通じて **UPDATE DATABASE CONFIGURATION** コマンドを使用してください。例えば、ロギング・モードを「archival logging」に、ZELLMART というデータベースを含む単一パーティション・データベース環境で変更する場合は、次を使用します。

```
CALL SYSPROC.ADMIN_CMD ('update db cfg for zellmart using logarchmeth1 logretain')
```

**logarchmeth1** 構成パラメーターが変更されたことをチェックするには、次を使用します。

```
SELECT * FROM SYSIBMADM.DBCFG WHERE NAME='logarchmeth1'
```

特定のデータベース・パーティションのデータベース構成パラメーターを更新するには、以下を行います。

1. **DB2NODE** 変数にデータベース・パーティション番号を設定します。
2. データベース・パーティションに接続する。
3. ADMIN\_CMD プロシージャーで **UPDATE DATABASE CONFIGURATION** コマンドを使用してデータベース構成パラメーターを更新する。
4. データベース・パーティションから切断する。

あるいは、**MEMBER** を使用することもできます。例えば、**MEMBER** を使用してロギング・モードを 1 つだけの特定のパーティション (30) に更新するには、次を使用します。

```
CALL SYSPROC.ADMIN_CMD ('update db cfg for zellmart member 30 using  
logarchmeth1 logretain')
```

DB2 データベース構成パラメーターと、各種データベース・ノードに使用できる値についての詳細は、個々の構成パラメーターの説明を参照してください。これらのパラメーターの値は、構成するデータベース・ノードの各タイプ (サーバー、クライアント、またはリモート・クライアントを持つサーバー) によって異なります。

すべてのパラメーターを更新できるわけではありません。

データベース構成ファイルへの変更の一部は、ファイルがメモリーにロードされた後にのみ有効になります。これを行う前にすべてのアプリケーションはデータベー

スから切断されている必要があります。オンラインで構成できるパラメーターと構成できないパラメーターについては、構成パラメーターの一覧をご覧ください。

エラーが発生した場合、データベース構成ファイルは変更されません。チェックサムが無効な場合、データベース構成ファイルは更新できません。適当なコマンドを使用しないでデータベース構成ファイルを変更するとこれが発生することがあります。これが発生する場合、データベースをリストアしてデータベース構成ファイルをリセットする必要があります。

## 互換性

以前のバージョンとの互換性:

- **DB2\_ENFORCE\_MEMBER\_SYNTAX** レジストリー変数が ON に設定されている場合を除き、**DBPARTITIONNUM** を **MEMBER** の代わりに使用できます。

## UPDATE DATABASE MANAGER CONFIGURATION コマンド (ADMIN\_CMD プロシージャを使用):

現在接続されているデータベースを含むインスタンスについて、データベース・マネージャーの構成ファイルの中の個々の項目を修正します。

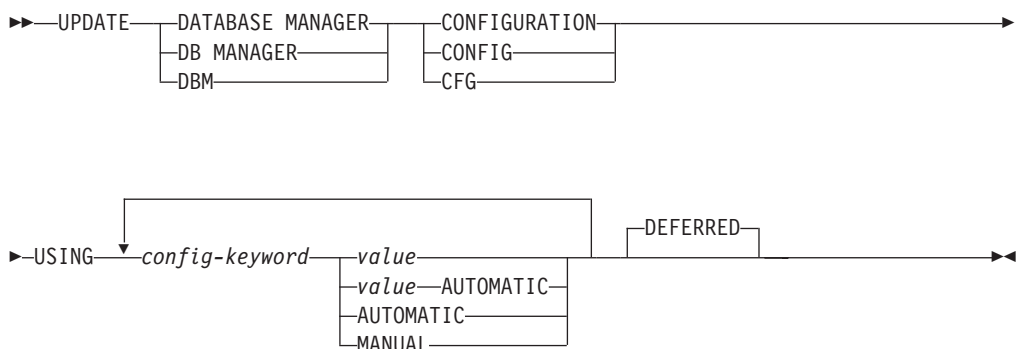
## 許可

SYSADM

## 必要な接続

データベース

## コマンド構文



## コマンド・パラメーター

### USING *config-keyword value*

更新するデータベース・マネージャー構成パラメーターを指定します。構成パラメーターのリストは、構成パラメーターのサマリーを参照してください。 *value* は、パラメーターに割り当てる値を指定します。

### AUTOMATIC

いくつかの構成パラメーターは **AUTOMATIC** に設定できます。それにより、DB2 が自動的にこれらのパラメーターを調整し、現行のリソース要件を反

映します。 **AUTOMATIC** キーワードをサポートする構成パラメーターのリストは、構成パラメーターのサマリーを参照してください。 **AUTOMATIC** キーワードと一緒に値を指定すると、自動計算に影響する可能性があります。この動作の具体的な詳細については、構成パラメーターの資料を参照してください。

**注: federated\_async** データベース・マネージャー構成パラメーターは、コマンド行プロセッサを使用することによってのみ **AUTOMATIC** に設定できます。

**MANUAL** 構成パラメーターの自動チューニングを使用不可にします。パラメーターはその現行の内部値に設定され、自動的な更新は行われなくなります。

#### **DEFERRED**

構成ファイルでのみ変更を行います。したがって、加えられた変更は、インスタンスの再始動時に有効になります。これはデフォルトです。

これは、CLPPlus インターフェースで操作する場合のデフォルトの節です。

**DEFERRED** は、CLPPlus プロセッサの使用時に呼び出す必要はありません。

#### **例**

データベース・マネージャー構成の診断レベルを 1 に更新します。

```
CALL SYSPROC.ADMIN_CMD('db2 update dbm cfg using DIAGLEVEL 1')
```

#### **使用上の注意**

データベース・マネージャー構成パラメーターのリストを表示または印刷するには、SYSIBMADM.DBMCFG 管理ビューを使用してください。データベース・マネージャー構成パラメーターを推奨されているデータベース・マネージャーのデフォルトにリセットするには、ADMIN\_CMD プロシージャを通じて **RESET DATABASE MANAGER CONFIGURATION** コマンドを使用してください。データベース・マネージャーの構成パラメーターと、構成されている各種データベース・ノード (サーバー、クライアント、またはリモート・クライアントを持つサーバー) に適したこれらのパラメーターの値については、個々の構成パラメーターの説明を参照してください。

すべてのパラメーターを更新できるわけではありません。

データベース・マネージャー構成ファイルへの変更の一部は、ファイルがメモリーにロードされた後にのみ有効になります。オンラインで構成できるパラメーターと構成できないパラメーターについては、構成パラメーターの一覧をご覧ください。即時にリセットされないサーバー構成パラメーターは、**db2start** の実行中にリセットされます。クライアント構成パラメーターの場合、パラメーターは次にアプリケーションを開始するときにリセットされます。クライアントがコマンド行プロセッサである場合は、**TERMINATE** を呼び出すことが必要です。

エラーが生じた場合には、データベース・マネージャー構成ファイルは変更されません。



データベース・マネージャー構成ファイルは、そのチェックサムが無効であると、更新することができません。このような状況は、データベース・マネージャー構成ファイルが変更されて、適切なコマンドが使用されていない場合に起こります。チェックサムが無効な場合は、データベース・マネージャーを再インストールして、データベース・マネージャー構成ファイルをリセットする必要があります。

現行のインスタンスの **SVCENAME**、または **TPNAME** データベース・マネージャー構成パラメーターを更新するとき、LDAP サポートを使用することができて、このインスタンスに LDAP サーバーが登録されている場合は、LDAP サーバーが新しい値に更新されます。

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

更新は、接続されているデータベースが含まれているデータベース・インスタンスに対してのみ可能です。

動的更新がサポートされているパラメーターの場合、**IMMEDIATE** キーワードが指定されていなくても、それを動的に更新することが試みられます。使用される許可は現在の SYSTEM\_USER ID です。

#### UPDATE HEALTH NOTIFICATION CONTACT LIST コマンド (ADMIN\_CMD プロシージャを使用):

インスタンスによって発行されるヘルス・アラートについての通知の連絡先リストを更新します。

#### 許可

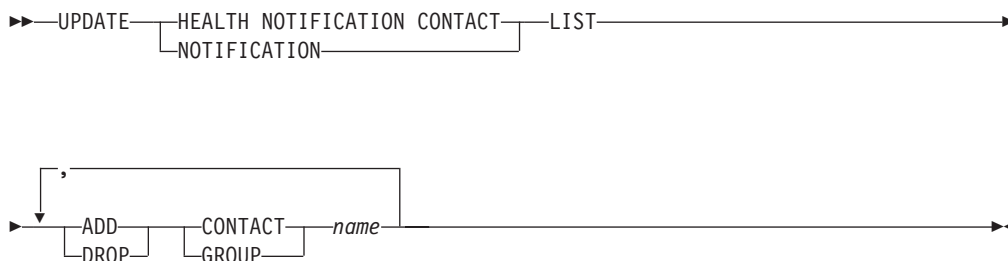
以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSM maint

#### 必要な接続

データベース

#### コマンド構文



## コマンド・パラメーター

### ADD GROUP *name*

インスタンスのヘルスの通知を受ける新しい連絡先グループを追加します。

### ADD CONTACT *name*

インスタンスのヘルスの通知を受ける新しい連絡先を追加します。

### DROP GROUP *name*

インスタンスのヘルスの通知を受ける連絡先のリストから、連絡先グループを除去します。

### DROP CONTACT *name*

インスタンスのヘルスの通知を受ける連絡先のリストから、連絡先を除去します。

## 例

ヘルス通知連絡先リストに連絡先グループ `gname1` を追加します。

```
CALL SYSPROC.ADMIN_CMD( 'update notification list add group gname1' )
```

## 使用上の注意

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

## UPDATE HISTORY コマンド (ADMIN\_CMD プロシージャーを使用):

現在接続されているデータベース・パーティションのデータベース履歴レコード項目にあるロケーション、装置タイプ、コメント、または状況を更新します。

## 許可

以下の権限のいずれか。

- SYSADM
- SYSCTRL
- SYSMOINT
- DBADM

## 必要な接続

データベース

## コマンド構文

```
▶▶ UPDATE HISTORY — FOR — object-part — WITH —————▶
                    └─ EID — eid ─┘

▶ ┌ LOCATION — new-location — DEVICE TYPE — new-device-type ─▶
  └ COMMENT — new-comment ─────────────────────────────────▶
    STATUS — new-status ─────────────────────────────────▶
```

## コマンド・パラメーター

### **FOR** *object-part*

更新される履歴項目の ID を指定します。この ID は、タイム・スタンプと 001 から 999 までのオプションのシーケンス番号で構成されます。項目の状況を更新するためにこのパラメーターを使用することはできません。項目の状況を更新するには、代わりに EID を指定してください。

### **EID** *eid*

履歴項目 ID を指定します。

### **LOCATION** *new-location*

バックアップ・イメージの新しい物理ロケーションを指定します。このパラメーターの解釈は装置タイプに依存します。

### **DEVICE TYPE** *new-device-type*

バックアップ・イメージを保管する新しい装置タイプを指定します。有効な装置タイプは次のとおりです。

<b>D</b>	ディスク
<b>K</b>	ディスケット
<b>T</b>	テープ
<b>A</b>	Tivoli Storage Manager
<b>F</b>	スナップショット・バックアップ
<b>U</b>	ユーザー出口
<b>P</b>	パイプ
<b>N</b>	Null 装置
<b>X</b>	XBSA
<b>Q</b>	SQL ステートメント
<b>O</b>	その他

### **COMMENT** *new-comment*

項目を記述する新しい注釈を指定します。

### **STATUS** *new-status*

項目の新しい状況を指定します。バックアップ項目のみがその状況を更新できます。有効な値は以下のとおりです。

<b>A</b>	アクティブ。バックアップ・イメージはアクティブ・ログ・チェーン上にあります。ほとんどの項目はアクティブです。
<b>I</b>	非アクティブ。現行のログ・シーケンス (現行のログ・チェーンとも言う) に対応しなくなったバックアップ・イメージには、非アクティブのフラグが立てられます。
<b>E</b>	期限切れ。アクティブ・イメージの数が NUM_DB_BACKUPS を超えたために不要になったバックアップ・イメージは、期限切れのフラグが立てられます。
<b>D</b>	削除済み。リカバリーに使用可能でなくなったバックアップ・イメージは、削除済みとしてマークされることとなります。

- X 削除しません。DB2HISTORY\_STATUS\_DO\_NOT\_DELETE のマークが付いたリカバリー・データベース履歴レコード項目は、**PRUNE HISTORY** コマンドの呼び出し、**PRUNE HISTORY** を指定した **ADMIN\_CMD** プロシージャの実行、db2Prune API の呼び出し、またはリカバリー・データベース履歴レコードファイルの自動ブルーニングにより整理されません。
- DB2HISTORY\_STATUS\_DO\_NOT\_DELETE 状況を使用すると、キー・リカバリー・ファイルの項目が整理されたり、それらに関連付けられたリカバリー・オブジェクトが削除されたりしないように保護できます。ログ・ファイル、バックアップ・イメージ、およびロード・コピー・イメージのみに
- DB2HISTORY\_STATUS\_DO\_NOT\_DELETE のマークを付けることができます。

## 例

1997 年 4 月 13 日午前 10 時 00 分にとった全データベース・バックアップのデータベース履歴レコード項目を更新するには、次のように入力します。

```
CALL SYSPROC.ADMIN_CMD('update history
for 199704131000000001 with location
/backup/dbbackup.1 device type D')
```

## 使用上の注意

データベース履歴レコードの主な用途は情報を記録することですが、履歴に含まれるデータは、自動リストア操作で直接使用されます。 **AUTOMATIC** オプションを指定したリストアにおいては、リストア・ユーティリティーによりバックアップ・イメージとそのロケーションの履歴が参照および使用されることにより、自動リストア要求が処理されます。自動リストア機能を使用する場合に、バックアップ・イメージが作成されてから再配置されているなら、現在のロケーションを反映するように、それらのイメージのデータベース履歴レコードを更新することをお勧めします。データベース履歴の中のバックアップ・イメージのロケーションが更新されない場合、自動リストア処理においてはバックアップ・イメージを見つけることができなくなりますが、手動リストア・コマンドは正常に使用できます。

コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。

*object-part* または *eid* が、接続されているデータベース・パーティションのログ履歴項目を指していなければなりません。

## **ADMIN\_CMD** プロシージャを使用する **UPDATE STMM TUNING** コマンド:

ユーザー設定のセルフチューニング・メモリー・マネージャー (STMM) の調整データベース・メンバー番号を更新します。

## 許可

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかの権限が含まれていなければなりません。

- DBADM

- DATAACCESS
- SQLADM

### 必要な接続

データベース

### コマンド構文

▶▶—UPDATE—STMM—TUNING—MEMBER—*member-number*—▶▶

### コマンド・パラメーター

#### *member-number*

*member-number* は整数です。パーティション・データベース環境では、-1 または存在しないメンバー番号が使用される場合、DB2 は STMM メモリー・チューナーを実行する適切なメンバーを自動的に選択します。DB2 pureScale 環境では、-1 または存在しないメンバー番号が使用される場合、DB2 は STMM メモリー・チューナーを実行する適切なメンバーをランダムに選択します。

### 例

パーティション・データベース環境で、ユーザー設定のセルフチューニング・メモリー・マネージャー (STMM) の調整データベース・パーティションを更新して、メンバー 3 にします。

```
CALL SYSPROC.ADMIN_CMD( 'update stmm tuning member 3' )
```

### 使用上の注意

- STMM 調整プロセスは、ユーザー設定の STMM 調整メンバー番号の値の変更を定期的に検査します。STMM 調整プロセスは、*member-number* が存在しており、それがアクティブなメンバーであれば、ユーザー設定の STMM 調整メンバーに移ります。このコマンドが STMM 調整メンバー番号を変更すると、現在の STMM 調整メンバー番号は即時に変更されます。
- コマンドの実行状況は、CALL ステートメントからの結果である SQLCA で戻されます。
- このコマンドは、その変更内容を ADMIN\_CMD プロシージャでコミットします。

### 互換性

以前のバージョンとの互換性:

- DB2\_ENFORCE\_MEMBER\_SYNTAX レジストリー変数が ON に設定されている場合を除き、DBPARTITIONNUM を MEMBER の代わりに使用できます。

### ADMIN\_COPY\_SCHEMA プロシージャ - 特定のスキーマとそのオブジェクトのコピー

ADMIN\_COPY\_SCHEMA プロシージャは、特定のスキーマと、その中に含まれているすべてのオブジェクトをコピーするために使用されます。新しいターゲット・

スキーマ・オブジェクトは、ソース・スキーマ内のオブジェクトと同じオブジェクト名を使って作成されますが、ターゲット・スキーマの修飾子が付きます。

ADMIN\_COPY\_SCHEMA プロシージャは表をコピーするために使用することができます。元の表のデータは、含めることも除くことも可能です。

## 構文

```
▶—ADMIN_COPY_SCHEMA—(—sourceschema—,—targetschema—,—copymode—,——————▶  
▶—objectowner—,—sourcetbsp—,—targettbsp—,—errortabschema—,—errortab—)——▶
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *sourceschema*

コピーされるオブジェクトが属しているスキーマの名前を指定する、タイプ VARCHAR(128) の入力引数。この名前は大文字小文字が区別されます。

### *targetschema*

コピーされたオブジェクトの作成先となる固有のスキーマ名を指定する、タイプ VARCHAR(128) の入力引数。この名前は大文字小文字が区別されます。そのスキーマ名が既に存在する場合、プロシージャ呼び出しは失敗し、プロシージャを呼び出す前にそのスキーマを削除する必要があることを示すメッセージが戻されます。

### *copymode*

コピー操作のモードを指定する、タイプ VARCHAR(128) の入力引数。有効なオプションは以下のとおりです。

- 'DDL': ソース・スキーマの、サポートされているすべてのオブジェクトの空のコピーを作成します。
- 'COPY': ソース・スキーマのすべてのオブジェクトの空のコピーを作成し、それから各ターゲット・スキーマ表にデータをロードします。ロードは 'NONRECOVERABLE' モードで行われます。ADMIN\_COPY\_SCHEMA を呼び出した後でバックアップを取る必要があります。そうしなければ、リカバリーの後、コピーされた表がアクセス不能になります。
- 'COPYNO': ソース・スキーマのすべてのオブジェクトの空のコピーを作成し、それから各ターゲット・スキーマ表にデータをロードします。ロードは 'COPYNO' モードで行われます。

注: *copymode* が「COPY」または「COPYNO」の場合、完全修飾ファイル名 (例えば「COPYNO /home/mckeough/loadoutput」) を *copymode* パラメーター値とともに指定できます。パスが渡されると、指定のファイルにロード・メッセージが記録されます。ファイル名は、インスタンスでの fenced ルーチン呼び出しで使用されるユーザー ID が書き込めるものでなければなりません。パスが指定されないと、ロード・メッセージ・ファイルは廃棄されます (デフォルトの振る舞い)。

### *objectowner*

コピーされたオブジェクトの所有者として使用される許可 ID を指定する、タ

イブ VARCHAR(128) の入力引数。NULL の場合、所有者は、コピー操作を実行するユーザーの許可 ID になります。

**sourcetbsp**

コピー用のソース表スペースのコンマ区切りリストを指定する、タイプ CLOB(2 M) の入力引数。区切り文字で区切られている表スペース名がサポートされません。作成される表ごとに、このリストに含まれているいずれかの表スペースと表定義が、*targettbsp* リストの n 番目の項目に変換されます。このパラメーターに NULL が指定されている場合、新規オブジェクトは、ソース・オブジェクトが使用するのと同じ表スペースを使って作成されます。

**targettbsp**

コピー用のターゲット表スペースのコンマ区切りリストを指定する、タイプ CLOB(2 M) の入力引数。区切り文字で区切られている表スペース名がサポートされます。表スペースのリスト *sourcetbsp* の各項目ごとに 1 つの表スペースが指定されていなければなりません。DDL の再生中、*sourcetbsp* リストの n 番目の表スペースは、*targettbsp* リストの n 番目の表スペースにマップされます。'SYS\_ANY' を最後の表スペース (ソース・リストのどの名前とも対応しない、追加の表スペース名) として指定することが可能です。'SYS\_ANY' が検出されると、オブジェクト作成時にデフォルトの表スペース選択アルゴリズムが使用されます (選択アルゴリズムについて詳しくは、CREATE TABLE ステートメントの資料の IN *tablespace-name1* オプションを参照してください)。このパラメーターに NULL が指定されている場合、新規オブジェクトは、ソース・オブジェクトが使用するのと同じ表スペースを使って作成されます。

**errortabschema**

コピーできなかったオブジェクトのエラー情報を入れる表のスキーマ名を指定する、タイプ VARCHAR(128) の入出力引数。ADMIN\_COPY\_SCHEMA プロシージャがユーザーのためにこの表を SYSTOOLSPACE 表スペースに作成します。エラーが生じなかった場合、このパラメーターの出力は NULL です。

**errortab**

コピーできなかったオブジェクトのエラー情報を入れる表の名前を指定する、タイプ VARCHAR(128) の入出力引数。ADMIN\_COPY\_SCHEMA プロシージャがユーザーのためにこの表を SYSTOOLSPACE 表スペースに作成します。この表は、プロシージャを呼び出したユーザー ID によって所有されます。エラーが生じなかった場合、このパラメーターの出力は NULL です。表が作成できなかった場合、または既に存在する場合には、プロシージャ操作は失敗し、エラー・メッセージが戻されます。ADMIN\_COPY\_SCHEMA プロシージャを呼び出した後、ユーザーは表をクリーンアップしなければなりません。つまり、SYSTOOLSPACE で表が占めているスペースを取り戻すためには、表をドロップする必要があります。

表 61. ADMIN\_COPY\_SCHEMA errortab の形式

列名	データ・タイプ	説明
OBJECT_SCHEMA	VARCHAR(128)	object_schema - オブジェクト・スキーマ・モニター・エレメント
OBJECT_NAME	VARCHAR(128)	object_name - オブジェクト名モニター・エレメント

表 61. ADMIN\_COPY\_SCHEMA errortab の形式 (続き)

列名	データ・タイプ	説明
OBJECT_TYPE	VARCHAR(30)	objtype - オブジェクト・タイプ・モニター・エレメント
SQLCODE	INTEGER	エラー SQLCODE。
SQLSTATE	CHAR(5)	エラー SQLSTATE。
ERROR_TIMESTAMP	TIMESTAMP	失敗した操作の失敗時刻。
STATEMENT	CLOB(2 M)	失敗したオブジェクトの DDL。ターゲット表へのデータのロード中に障害が生じた場合、このフィールドには失敗したロード・コマンドに対応するテキストが含まれません。
DIAGTEXT	CLOB(2 K)	失敗した操作のエラー・メッセージ・テキスト。

## 許可

スキーマが正常にコピーされるためには、ユーザーは CREATE\_SCHEMA 特権および DB2 オブジェクト固有の特権を持っている必要があります。

例: ADMIN\_COPY\_SCHEMA コマンドで表をコピーするために必要な CREATE\_TABLE 特権および索引をコピーするために必要な CREATE\_INDEX 特権が必要です。

ソース・スキーマ内の表がラベル・ベースのアクセス制御 (LBAC) で保護されている場合、ユーザー ID は、ターゲット表にもその同じ保護を作成できるようにするための LBAC 信用証明情報を必要とします。データもコピーする場合、ユーザー ID は、ソース表からのデータの読み取りとそのデータのターゲット表への書き込みの両方を許可する LBAC 信用証明情報を必要とします。

ADMIN\_COPY\_SCHEMA プロシージャに対する EXECUTE 特権も必要です。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
CALL SYSPROC.ADMIN_COPY_SCHEMA('SOURCE_SCHEMA', 'TARGET_SCHEMA',
    'COPY', NULL, 'SOURCETS1', SOURCETS2', 'TARGETTS1', TARGETTS2,
    SYS_ANY', 'ERRORSCHEMA', 'ERRORNAME')
```

## 制約事項

- HADR を構成するデータベースでは、DDL *copymode* のみがサポートされています。
- COPY または COPY NO を指定した XML はサポートされません。



- COPYNO オプションを指定した ADMIN\_COPY\_SCHEMA プロシージャを使用して、ターゲット・データベース・オブジェクトが常駐している表スペースを、バックアップ・ペンディング状態にします。ロード操作の完了後、ターゲット・スキーマ表は SET INTEGRITY ペンディング状態になります。その後、ADMIN\_COPY\_SCHEMA プロシージャは、SET INTEGRITY ステートメントを発行して、表をこの状態から解除します。表スペースは既にバックアップ・ペンディング状態にあるので、SET INTEGRITY ステートメントは失敗します。この問題の解決方法について詳しくは、『スキーマのコピー』を参照してください。

## 使用上の注意

- コピーされるオブジェクト内の完全修飾されたオブジェクトへの参照は変更されません。ADMIN\_COPY\_SCHEMA プロシージャは、作成されるオブジェクトの修飾スキーマのみを変更し、それらのオブジェクトの SQL 式内に現れるスキーマ名は変更しません。これには生成された列やトリガー本体などのオブジェクトが含まれます。
- このプロシージャは、以下のオブジェクトのコピーはサポートしません。
  - 索引拡張
  - ニックネーム
  - パッケージ
  - 型付き表
  - 配列タイプ
  - ユーザー定義の構造化タイプ (およびそれらのトランスフォーム関数)
  - 型付きビュー
  - JAR (Java ルーチン・アーカイブ)
  - ステージング表
  - 同じソース・スキーマに属さない基本オブジェクトの別名
- コピーされるスキーマ内にこれらのオブジェクトのいずれかが存在する場合、そのオブジェクトはコピーされず、オブジェクトがコピーされなかったことを示す項目がエラー表に追加されます。
- 複製された表をコピーする場合、表の新しいコピーではサブスクリプションが使用可能になりません。表は、基本表として再作成されるに過ぎません。
- このプロシージャを操作するには、SYSTOOLSPACE 表スペースが存在していることが必要です。この表スペースは、ADMIN\_COPY\_SCHEMA プロシージャが使用するメタデータと、このプロシージャによって戻されるエラー表を保管するために使用されます。表スペースが存在しない場合は、エラーが戻されません。
- ターゲット・スキーマ内のオブジェクトの統計はデフォルトに設定されています。
- 表に生成された ID 列があり、かつ *copymode* が 'COPY' または 'COPYNO' のいずれかである場合、ソース表からのデータ値はロード中に保存されます。
- 各外部ルーチンごとに、元のソース・ルーチンのバイナリーを参照する新しいカタログ項目が 1 つ作成されます。

- コピー操作の開始時に表が SET INTEGRITY ペンディング状態にあった場合、データはターゲット表にロードされず、その表に関してデータがロードされなかったことを示す項目が *errortab* に記録されます。
- ロードまたは DDL 操作が失敗した場合、作成されなかったオブジェクトに関する項目が *errortab* に記録されます。正常に作成されたオブジェクトすべてはそのまま残ります。リカバリーするには、手動ロードを開始するか、ADMIN\_DROP\_SCHEMA プロシージャを使用して新しいスキーマをドロップしてから、ADMIN\_COPY\_SCHEMA プロシージャを再び呼び出すことができます。
- ターゲット・スキーマがソース・スキーマと一致する場合、DDL 再生中に、デフォルト・スキーマはターゲット・スキーマにオーバーライドされます。
- トリガー、ビュー、または SQL 関数のコンパイルに使用される関数パスは、ソース・オブジェクトの作成に使用されたパスです。ただしこれには例外があり、オブジェクトの関数パスにソース・スキーマ名が含まれている場合、DDL 再生中にパス内のこの項目がターゲット・スキーマ名に変更されます。
- 複数の ADMIN\_COPY\_SCHEMA プロシージャを実行すると、デッドロックが生じます。一度に 1 つの ADMIN\_COPY\_SCHEMA プロシージャ呼び出しだけを発行する必要があります。コピー処理中にソース・スキーマ内で表を変更すると、ターゲット・スキーマ内のデータがコピー操作の後には等しくないことを示す場合があります。
- 単一パーティションのデータベース・パーティション・グループ内の表スペースから、複数パーティションのデータベース・パーティション・グループ内の表スペースに、表を備えたスキーマをコピーするときは、慎重に考慮する必要があります。分散キーの自動選択が設定済みでない限り、スキーマのコピー操作を実行する前に、分散キーを表で定義する必要があります。分散キーの変更は、表スペースが単一パーティション・データベース・パーティション・グループと関連している表に対してのみ行うことができます。

### トランザクションの考慮事項

- ADMIN\_COPY\_SCHEMA プロシージャが、その処理中にデッドロックまたはロックのタイムアウトのためにロールバックを強制された場合、ADMIN\_COPY\_SCHEMA プロシージャを呼び出した作業単位内で行われた作業もロールバックされます。
- コピーの DDL 段階で障害が生じた場合、ターゲット・スキーマになされた変更はすべてセーブポイントにロールバックされます。
- *copymode* が 'COPY' または 'COPYNO' に設定されている場合、いったんコピーの DDL 段階が完了すると ADMIN\_COPY\_SCHEMA プロシージャがコミットし、そのプロシージャを呼び出した作業単位でなされた作業もコミットします。

### ADMIN\_DROP\_SCHEMA プロシージャ - 特定のスキーマとそのオブジェクトのドロップ

ADMIN\_DROP\_SCHEMA プロシージャは、特定のスキーマと、その中に含まれているすべてのオブジェクトをドロップするために使用されます。

## 構文

```
▶▶ ADMIN_DROP_SCHEMA (—schema—, —dropmode—, —errortabschema—, —errortab—)
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *schema*

ドロップされるスキーマの名前を指定する、タイプ VARCHAR(128) の入力引数。名前は、大文字で指定しなければなりません。

### *dropmode*

将来の使用のために予約済みであり、NULL に設定する必要があります。

### *errortabschema*

ドロップできなかったオブジェクトのエラー情報を入れる表のスキーマ名を指定する、タイプ VARCHAR(128) の入出力引数。この名前は大文字小文字が区別されます。ADMIN\_DROP\_SCHEMA プロシージャがユーザーのためにこの表を SYSTOOLSPACE 表スペースに作成します。エラーが生じなかった場合、このパラメーターの出力は NULL です。

### *errortab*

ドロップできなかったオブジェクトのエラー情報を入れる表の名前を指定する、タイプ VARCHAR(128) の入出力引数。この名前は大文字小文字が区別されます。ADMIN\_DROP\_SCHEMA プロシージャがユーザーのためにこの表を SYSTOOLSPACE 表スペースに作成します。この表は、プロシージャを呼び出したユーザー ID によって所有されます。エラーが生じなかった場合、このパラメーターの出力は NULL です。表が作成できなかった場合、または既存する場合には、プロシージャ操作は失敗し、エラー・メッセージが戻されず。ADMIN\_DROP\_SCHEMA を呼び出した後、ユーザーは表をクリーンアップしなければなりません。つまり、SYSTOOLSPACE で表が占めているスペースを取り戻すためには、表をドロップする必要があります。

表 62. ADMIN\_DROP\_SCHEMA errortab の形式

列名	データ・タイプ	説明
OBJECT_SCHEMA	VARCHAR(128)	object_schema - オブジェクト・スキーマ・モニター・エレメント
OBJECT_NAME	VARCHAR(128)	object_name - オブジェクト名モニター・エレメント
OBJECT_TYPE	VARCHAR(30)	objtype - オブジェクト・タイプ・モニター・エレメント
SQLCODE	INTEGER	エラー SQLCODE。
SQLSTATE	CHAR(5)	エラー SQLSTATE。
ERROR_TIMESTAMP	TIMESTAMP	ドロップ・コマンドが失敗した時刻。

表 62. ADMIN\_DROP\_SCHEMA errortab の形式 (続き)

列名	データ・タイプ	説明
STATEMENT	CLOB(2 M)	失敗したオブジェクトの DDL。
DIAGTEXT	CLOB(2 K)	失敗したドロップ・コマンドのエラー・メッセージ・テキスト。

## 許可

以下のいずれかの権限が必要です。

- ADMIN\_DROP\_SCHEMA プロシージャに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、このプロシージャを呼び出すユーザーには、削除対象のすべてのオブジェクトに対するドロップ権限が必要です。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
CALL SYSPROC.ADMIN_DROP_SCHEMA('SCHNAME', NULL, 'ERRORSCHEMA', 'ERRORTABLE')
```

以下はこのプロシージャの出力例です。

```
Value of output parameters
-----
Parameter Name : ERRORTABSCHEMA
Parameter Value : ERRORSCHEMA <-- error!

Parameter Name : ERRORTAB
Parameter Value : ERRORTABLE <-- error!

Return Status = 0
```

戻り状況は、内部エラーが検出された場合 (SYSTOOLSPACE が存在しないなど) のみ、0 ではありません。

エラーは、エラー表の照会によってチェックできます。

```
SELECT * FROM ERRORSCHEMA.ERRORTABLE
```

## 使用上の注意

- ドロップするオブジェクトに別のスキーマ内のオブジェクトが従属している場合、デフォルトの DROP ステートメントのセマンティクスが適用されます。
- このプロシージャは、以下のオブジェクトのドロップはサポートしません。
  - 索引拡張
  - ニックネーム

- パッケージ
  - 型付き表
  - 配列タイプ
  - ユーザー定義の構造化タイプ (およびそれらのトランスフォーム関数)
  - 型付きビュー
  - JAR (Java ルーチン・アーカイブ)
  - ステージング表
  - XSR オブジェクト
- ドロップされるスキーマ内にこれらのオブジェクトのいずれかが存在する場合、そのオブジェクトもそのスキーマもドロップされず、オブジェクトがドロップされなかったことを示す項目がエラー表に追加されます。
  - このプロシージャを操作するには、SYSTOOLSPACE 表スペースが存在していることが必要です。この表スペースは、ADMIN\_DROP\_SCHEMA プロシージャが使用するメタデータと、このプロシージャによって戻されるエラー表を保管するために使用されます。表スペースが存在しない場合は、エラーが戻されません。

## ADMIN\_EST\_INLINE\_LENGTH 関数 - データのインライン化に必要な長さを見積もる

ADMIN\_EST\_INLINE\_LENGTH 関数は、XML 列、BLOB 列、CLOB 列、または DBCLOB 列に保管されているデータをインライン化するために必要なインライン長の見積もりを戻します。

データをインライン化できない場合、関数は負の値を戻します。

データがすでにインライン化されている場合、関数はインライン化データの実際の長さを戻します。

### 構文

▶▶—ADMIN\_EST\_INLINE\_LENGTH—(—column-name—)—————▶▶

スキーマは SYSIBM です。

### 戻り値

この関数は、データの見積インライン長 (バイト) を表す INTEGER 値か、あるいは以下のいずれかの値を戻します。

NULL 入力が NULL であることを示します。

- 1 列値をインライン化するのを可能にする有効なインライン長がないため、データをインライン化できないことを示します。
- 2 文書が DB2 for Linux, UNIX, and Windows バージョン 9.7 より前のリリースに挿入および保管されたため、文書の見積インライン長を判別できないことを示します。

## 関数のパラメーター

*column-name*

データ・タイプ XML、BLOB、CLOB、または DBCLOB を持つ基本表の列を識別します (SQLSTATE 42884)。この列は、式に基づいて生成されていない基本表の列を直接的または間接的に参照する必要があります (SQLSTATE 42815)。

### 例

例 1: 以下の例は、TAB1 表の XML 列 xml\_doc1 に含まれている 3 つの XML 文書の見積インライン長を戻します。

```
db2 => SELECT PK, ADMIN_IS_INLINED(xml_doc1) as IS_INLINED,  
          ADMIN_EST_INLINE_LENGTH(xml_doc1) as EST_INLINE_LENGTH  
        from TAB1
```

この照会では、以下の出力が結果として戻ります。

PK	IS_INLINED	EST_INLINE_LENGTH
1	1	292
2	0	450
3	0	454

3 record(s) selected.

この例では、ADMIN\_IS\_INLINED 関数は最初の文書がインライン化されていることを示しています。このため、ADMIN\_EST\_INLINE\_LENGTH 関数はインライン化 XML 文書の実際の長さを戻します。2 番目の文書はインライン化されていないため、ADMIN\_EST\_INLINE\_LENGTH 関数は、2 番目の XML 文書をインライン化するために必要な見積インライン長を戻します。

例 2: 以下の例は、TAB1 表の XML 列 xml\_doc1 に含まれている 1 つの XML 文書の見積インライン長を戻します。この例には、述部が含まれます。

```
db2 => SELECT PK, ADMIN_IS_INLINED(xml_doc1) as IS_INLINED,  
          ADMIN_EST_INLINE_LENGTH(xml_doc1) as EST_INLINE_LENGTH  
        from TAB1 where PK=2
```

この照会では、以下の出力が結果として戻ります。

PK	IS_INLINED	EST_INLINE_LENGTH
2	0	450

1 record(s) selected.

例 3: 以下の例は、TAB1 表の CLOB 列 clob\_1 に含まれている 3 つの CLOB データの見積インライン長を戻します。

```
db2 => SELECT PK, ADMIN_IS_INLINED(clob_1) as IS_INLINED,  
          ADMIN_EST_INLINE_LENGTH(clob_1) as EST_INLINE_LENGTH  
        from TAB1
```

この照会では、以下の出力が結果として戻ります。

PK	IS_INLINED	EST_INLINE_LENGTH
1	1	68

```

      2      0      3665
      3      0      -1

```

3 record(s) selected.

### 使用上の注意

- XML 列がサポートされるのは、XML 文書が DB2 for Linux, UNIX, and Windows バージョン 9.7 以降を使用して挿入された場合のみです。これより前のリリースで挿入された XML 文書の場合、保管形式は異なります。正しくない保管形式を検出すると、ADMIN\_EST\_INLINE\_LENGTH 関数は値 -2 を返します。
- 列のインライン長を増やすことを計画している場合、この長さは減らすことができないことに注意してください。
- インライン長を増やすと、行サイズの合計も増加し、バッファーク・プールのパフォーマンスにも影響を与える場合があります。行サイズの合計には、以下の制限があります。

表 63. 行サイズの制限

ページ・サイズ	行サイズの制限	インライン長の制限
4K	4005	4001
8K	8101	8097
16K	16 293	16 289
32K	32 677	32 673

- XML ストレージ・オブジェクトのページ・サイズが基本表のページ・サイズと同じでない場合は、見積インライン長は正確でない可能性があります。

### ADMIN\_GET\_INDEX\_COMPRESS\_INFO 表関数 - 圧縮索引情報を戻す

ADMIN\_GET\_INDEX\_COMPRESS\_INFO 表関数は、非圧縮索引について索引圧縮により見込まれる削減量を戻します。

#### 構文

```

▶▶—ADMIN_GET_INDEX_COMPRESS_INFO—(—objecttype—,—objectschema—,—objectname—,—
▶—member—,—datapartitionid—)—————▶▶

```

スキーマは SYSPROC です。

#### 表関数パラメーター

##### objecttype

オブジェクト・タイプを示す、タイプ VARCHAR(1) の入力引数。値は以下のいずれかでなければならず、大/小文字を区別します。

- 'T'、NULL、または空ストリング (表を示す場合)
- 'I' (索引の場合)

##### objectschema

オブジェクト・スキーマを指定する、タイプ VARCHAR (128) の大/小文字を区別する入力パラメーター。

*objecttype* が 'T'、NULL、または空ストリング (") である場合、*objectschema* は表スキーマを示します。

- *objectschema* が指定され、*objectname* が NULL または空ストリング (") の場合、指定されたスキーマのすべての表に関する情報が戻されます。
- *objectschema* と *objectname* の両方が指定される場合、指定された表のすべての索引に関する情報が戻されます。

*objecttype* が 'I' である場合、*objectschema* は索引スキーマを示します。

- *objectschema* が指定され、*objectname* が NULL または空ストリング (") の場合、指定されたスキーマのすべての索引に関する情報が戻されます。
- *objectschema* と *objectname* の両方が指定される場合、指定された索引に関する情報が戻されます。
- *objectschema* と *objectname* のどちらも指定されない場合、すべてのスキーマのすべての索引に関する情報が戻されます。

*objectname* が指定され、*objectschema* が指定されない場合、関数が SQL エラーを戻します。値が NULL または空ストリング (") である場合、パラメーター値は未指定であるといえます。

#### *objectname*

オブジェクト名を指定する、タイプ VARCHAR (128) の大/小文字を区別する入力パラメーター。 *objectschema* パラメーターの説明を参照してください。

#### *member*

データベース・メンバー番号を指定するタイプ INTEGER の入力パラメーター。指定されている場合、指定されたデータベース・メンバーに存在する索引に関する情報のみが戻されます。すべてのアクティブ・データベース・メンバーに関してデータが戻されるように指定するには、*member* パラメーター値を -2 または NULL に設定します。単一メンバー環境では、-2 または NULL を指定します。

#### *datapartitionid*

データ・パーティション ID を指定するタイプ INTEGER の入力パラメーター。指定されている場合、指定されたデータ・パーティションで定義された索引パーティションに関する情報のみが戻されます。データ・パーティション ID は、SYSCAT.DATAPARTITIONS ビューにある DATAPARTITIONID に対応します。すべてのデータ・パーティションに関してデータが戻されるように指定するには、*datapartitionid* パラメーター値を -2 または NULL に設定します。非パーティション索引の場合は、-2、0、または NULL を指定します。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限



## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 使用上の注意

- 既存の索引についての圧縮による実際の削減量を取得したい場合、この関数は使用しないでください。その場合は代わりに (統計が最新でなければ) RUNSTATS コマンドを実行し、次のカタログ値のいずれかまたは両方を照会します。
  - SYSCAT.INDEXES.PCTPAGESSAVED
  - SYSCAT.INDEXPARTITIONS.PCTPAGESSAVED

### 例

データベースのマイグレーション後は、すべての既存の索引は圧縮解除されます。データベース・パーティション番号 2 に置かれている、データ・パーティション ID が 3 の表「S.T1」について、既存の索引の索引圧縮による節約の可能性を見積もることもできます。この例では、S はスキーマ名、T1 は表名であり、T1 は圧縮されていません。

```
SELECT compress_attr, iid, dbpartitionnum, index_compressed,
       pct_pages_saved, num_leaf_pages_saved
FROM TABLE(sysproc.admin_get_index_compress_info('S', 'T1', 2, 3))
AS t
```

以下は、このステートメントからの出力例です。

COMPRESS_ATTR	IID	DBPARTITIONNUM	INDEX_COMPRESSED	...
N	1	2	N	...
N	2	2	N	...
...	PCT_PAGES_SAVED	NUM_LEAF_PAGES_SAVED		
...	50	200		
...	45	150		

圧縮による節約を行う価値があると判断し、索引圧縮を使用可能にしたいと考えます。

```
ALTER INDEX INDEX1 compress yes
ALTER INDEX INDEX2 compress yes
REORG INDEXES all FOR table S.T1
```

時間の経過とともに、表の新規索引を作成する必要について判断し、圧縮する前に索引圧縮によってこれらの索引についてどれほど節約されるかを見積もることもできます。さらに、すでに圧縮済みの索引から圧縮統計を参照することもできます。

```
SELECT compress_attr, iid, dbpartitionnum, index_compressed,
       pct_pages_saved, num_leaf_pages_saved
FROM TABLE(sysproc.admin_get_index_compress_info('S', 'T1', 2, 3))
AS t
```

以下は、このステートメントからの出力例です。

COMPRESS_ATTR	IID	DBPARTITIONNUM	INDEX_COMPRESSED	...
Y	1	2	Y	...

```

Y          2          2 Y          ...
N          3          2 N          ...
N          4          2 N          ...
... PCT_PAGES_SAVED NUM_LEAF_PAGES_SAVED
... -----
...          -1          -1
...          -1          -1
...          58          230
...          49          140

```

index\_compressed 列で示されているように、最初の 2 つの索引がすでに圧縮されているため、ステートメントはシステム・カタログから値が戻されます。この場合、カタログからの値は収集されませんでした。

表に対して RUNSTATS を実行した後、索引関数を次回実行するときに、訂正された結果が生成されます。

```

RUNSTATS ON TABLE S.T1 FOR INDEXES ALL
SELECT compress_attr, iid, dbpartitionnum, index_compressed,
       pct_pages_saved, num_leaf_pages_saved
FROM TABLE(sysproc.admin_get_index_compress_info(' ', 'S', 'T1', 2, 3))
AS t

```

以下は、このステートメントからの出力例です。

```

COMPRESS_ATTR      IID DBPARTITIONNUM INDEX_COMPRESSED ...
-----
Y          1          2 Y          ...
Y          2          2 Y          ...
N          3          2 N          ...
N          4          2 N          ...
... PCT_PAGES_SAVED NUM_LEAF_PAGES_SAVED
... -----
...          50          200
...          45          150
...          58          230
...          49          140

```

## 戻される情報

表 64. ADMIN\_GET\_INDEX\_COMPRESS\_INFO によって戻される情報

列名	データ・タイプ	説明
INDSCHEMA	VARCHAR(128)	index_schema - 索引スキーマ・モニター・エレメント
INDNAME	VARCHAR(128)	index_name - 索引名モニター・エレメント
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント
TABNAME	VARCHAR(128)	table_name - 表名モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
IID	SMALLINT	iid - 索引 ID モニター・エレメント
DATAPARTITIONID	INTEGER	データ・パーティション ID。
COMPRESS_ATTR	CHAR(1)	索引の COMPRESSION 属性の状態。 <ul style="list-style-type: none"> <li>• Y = 索引圧縮は使用可能</li> <li>• N = 索引圧縮は使用不可</li> </ul>

表 64. ADMIN\_GET\_INDEX\_COMPRESS\_INFO によって戻される情報 (続き)

列名	データ・タイプ	説明
INDEX_COMPRESSED	CHAR(1)	物理索引形式。 <ul style="list-style-type: none"> <li>• Y = 索引は圧縮形式である</li> <li>• N = 索引は非圧縮形式である</li> </ul> 物理索引形式が圧縮属性と一致しない場合、索引を定義済み形式に変換するには索引の再編成が必要です。この関数を実行したときに表または索引でエラーが発生した場合、この値は NULL です。
PCT_PAGES_SAVED	SMALLINT	索引が物理的に圧縮されていない (INDEX_COMPRESSED が N である) 場合、この値は、索引が実際に圧縮されているかのように、節約されるリーフ・ページの見積パーセントを表します。索引が物理的に圧縮されている (INDEX_COMPRESSED が Y である) 場合、この値は、システム・カタログ・ビュー (SYSCAT.INDEXES または SYSCAT.INDEXPARTITIONS のいずれか) からの PCTPAGESSAVED 値をレポートします。 <b>注:</b> この値は、パーティション・データベース環境のデータベース・パーティションごとの索引または索引パーティションの各項目で同じになります。この関数を実行したときに表または索引でエラーが発生した場合、この値は NULL です。
NUM_LEAF_PAGES_SAVED	BIGINT	索引が物理的に圧縮されていない (INDEX_COMPRESSED が N である) 場合、この値は、索引が実際に圧縮されているかのように、節約されるリーフ・ページ数の見積もりを表します。索引が物理的に圧縮されている (INDEX_COMPRESSED が Y である) 場合、この値は、システム・カタログ・ビュー (SYSCAT.INDEXES または SYSCAT.INDEXPARTITIONS のいずれか) から PCTPAGESSAVED および NLEAF 値に基づいて、節約される計算済みリーフ・ページ数をレポートします。 PCTPAGESSAVED または NLEAF のいずれかが無効値 (-1) である場合、この値も -1 に設定されます。 <b>注:</b> この値は、パーティション・データベース環境のデータベース・パーティションごとの索引または索引パーティションの各項目で同じになります。この関数を実行したときに表または索引でエラーが発生した場合、この値は NULL です。

## ADMIN\_GET\_INDEX\_INFO 表関数 - 索引情報を戻す

ADMIN\_GET\_INDEX\_INFO 表関数は、圧縮情報および索引の論理および物理サイズなど、カタログ・ビューで使用できない索引情報を戻します。

### 構文

▶▶—ADMIN\_GET\_INDEX\_INFO—(—objecttype—, —objectschema—, —objectname—)——▶▶

スキーマは SYSPROC です。

## 表関数パラメーター

### *objecttype*

オブジェクト・タイプを示す、タイプ VARCHAR(1) の入力引数。値は以下のいずれかでなければならず、大/小文字を区別します。

- 'T'、NULL、または空ストリング (") (表を示す場合)
- 'I' (索引の場合)

### *objectschema*

オブジェクト・スキーマを指定する、タイプ VARCHAR (128) の大/小文字を区別する入力パラメーター。

*objecttype* が 'T'、NULL、または空ストリング (") である場合、*objectschema* は表スキーマを示します。

- *objectschema* が指定され、*objectname* が NULL または空ストリング (") の場合、指定されたスキーマのすべての表に関する情報が戻されます。
- *objectschema* と *objectname* の両方が指定される場合、指定された表のすべての索引に関する情報が戻されます。

*objecttype* が 'I' である場合、*objectschema* は索引スキーマを示します。

- *objectschema* が指定され、*objectname* が NULL または空ストリング (") の場合、指定されたスキーマのすべての索引に関する情報が戻されます。
- *objectschema* と *objectname* の両方が指定される場合、指定された索引に関する情報が戻されます。
- *objectschema* と *objectname* のどちらも指定されない場合、すべてのスキーマのすべての索引に関する情報が戻されます。

*objectname* が指定され、*objectschema* が指定されない場合、関数が SQL エラーを戻します。値が NULL または空ストリング (") である場合、パラメーター値は未指定であるといえます。

### *objectname*

オブジェクト名を指定する、タイプ VARCHAR (128) の大/小文字を区別する入力パラメーター。 *objectschema* パラメーターの説明を参照してください。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

表のいくつかの索引に対して索引圧縮を有効にした後で、どの索引が圧縮されるのか、および圧縮するためにはどの索引を再作成する必要があるかを知りたいとします。この例では、S はスキーマ名、T1 は表名です。

```
db2 SELECT iid, compress_attr, index_compressed
      FROM TABLE(sysproc.admin_get_index_info('','S','T1')) AS t
```

以下はこの照会の出力例です。

IID	COMPRESS_ATTR	INDEX_COMPRESSED
1	Y	Y
2	Y	Y
3	Y	N
4	N	N

さらに、スキーマ S2 のすべての索引の他の索引情報を参照するとします。この例では、それぞれ次の意味を持つとします。

- T2 = 2 つのデータ・パーティションを持つパーティション表
- T3 = 非パーティション表
- IND\_1 = T2 上の非パーティション索引
- IND\_2 = T2 上のパーティション索引
- IND\_3 = T2 上のパーティション索引
- IND\_4 = T3 上の索引
- IND\_5 = T3 上の索引

```
db2 SELECT tablename, indname, iid, index_partitioning, datapartitionid,
      index_object_l_size, index_object_p_size, index_requires_rebuild,
      large_rids FROM TABLE(sysproc.admin_get_index_info('I','S2','')) AS t
```

以下はこの照会の出力例です。

TABNAME	INDNAME	IID	INDEX_PARTITIONING	DATAPARTITIONID
T2	IND_1	1	N	0
T2	IND_2	2	P	1
T2	IND_2	2	P	2
T2	IND_3	3	P	1
T2	IND_3	3	P	2
T3	IND_4	4		0
T3	IND_5	5		0

この手順からの出力 (続き):

INDEX_OBJECT_L_SIZE	INDEX_OBJECT_P_SIZE	INDEX_REQUIRES_REBUILD	LARGE_RIDS
50	51	N	Y
40	40	N	Y
45	45	N	Y
40	40	N	Y
45	45	N	Y
20	20	N	Y
20	20	N	Y

## 戻される情報

表 65. ADMIN\_GET\_INDEX\_INFO によって戻される情報

列名	データ・タイプ	説明
INDSCHEMA	VARCHAR(128)	index_schema - 索引スキーマ・モニター・エレメント
INDNAME	VARCHAR(128)	index_name - 索引名モニター・エレメント
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント
TABNAME	VARCHAR(128)	table_name - 表名モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
IID	SMALLINT	iid - 索引 ID モニター・エレメント
DATAPARTITIONID	INTEGER	データ・パーティション ID。
COMPRESS_ATTR	CHAR(1)	索引の COMPRESSION 属性の状態。 <ul style="list-style-type: none"> <li>• Y = 索引圧縮は使用可能</li> <li>• N = 索引圧縮は使用不可</li> </ul>
INDEX_COMPRESSED	CHAR(1)	物理索引形式。 <ul style="list-style-type: none"> <li>• Y = 索引は圧縮形式である</li> <li>• N = 索引は非圧縮形式である</li> </ul> 物理索引形式が圧縮属性と一致しない場合、索引を定義済み形式に変換するには索引の再編成が必要です。この関数を実行したときに表または索引でエラーが発生した場合、この値は NULL です。
INDEX_PARTITIONING	CHAR(1)	索引のパーティション特性を示します。 <ul style="list-style-type: none"> <li>• N = 非パーティション索引</li> <li>• P = パーティション索引</li> <li>• ブランク = 索引はパーティション表にはない</li> </ul>
INDEX_OBJECT_L_SIZE	BIGINT	索引オブジェクトの論理サイズ。非パーティション表の場合、これは、表に定義されるすべての索引に対して論理的に割り振られるディスク・スペースの量になります。パーティション表の非パーティション索引の場合、これは、索引に対して論理的に割り振られるディスク・スペースの量になります。パーティション表のパーティション索引の場合、これは、データ・パーティションに定義されるすべての索引パーティションに対して論理的に割り振られるディスク・スペースの量になります。すべてのサイズはキロバイト (KB) 単位でレポートされます。  論理サイズとは、表またはデータ・パーティションが認識するスペースの量のことです。このサイズは表またはデータ・パーティションの索引データを保持するために物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。戻されるサイズは、索引に対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される索引の場合は EMP エクステントの見積もりを考慮に入れます。この関数を実行したときに表または索引でエラーが発生した場合、この値は NULL です。

表 65. ADMIN\_GET\_INDEX\_INFO によって戻される情報 (続き)

列名	データ・タイプ	説明
INDEX_OBJECT_P_SIZE	BIGINT	<p>索引オブジェクトの物理サイズ。非パーティション表の場合、これは、表に定義されるすべての索引に対して物理的に割り振られるディスク・スペースの量になります。パーティション表の非パーティション索引の場合、これは、索引に対して物理的に割り振られるディスク・スペースの量になります。パーティション表のパーティション索引の場合、これは、データ・パーティションに定義されるすべての索引パーティションに対して物理的に割り振られるディスク・スペースの量になります。すべてのサイズはキロバイト (KB) 単位でレポートされます。</p> <p>戻されるサイズは、索引に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される索引の EMP エクステントが含まれます。この関数を実行したときに表または索引でエラーが発生した場合、この値は NULL です。</p>
INDEX_REQUIRES_REBUILD	CHAR(1)	<p>索引の再作成状況。</p> <ul style="list-style-type: none"> <li>表またはデータ・パーティションで定義される索引で再編成が必要な場合は Y</li> <li>それ以外の場合は N</li> </ul> <p>この関数を実行したときに表でエラーが発生した場合、この値は NULL です。</p>
LARGE_RIDS	CHAR(1)	<p>索引がラージ行 ID (RID) を使用しているかどうかを示します (4 バイトのページ番号と 2 バイトのスロット番号)。</p> <ul style="list-style-type: none"> <li>「Y」は、索引がラージ RID を使用していることを示します。</li> <li>「N」は、索引がラージ RID を使用していないことを示します。</li> <li>「P」(保留) は、索引が定義されている表がラージ RID をサポートしている (つまり表が LARGE 表スペースにある) もの、表またはデータ・パーティションの索引がまだ再編成されていないかまたは再作成されていないことを示します。したがって、表はまだ 4 バイト RID を使用しているので、表または索引をラージ RID に変換するアクションを取る必要があります。</li> </ul> <p>この関数を実行する表に誤りがある場合、この値は NULL になります。</p>
RECLAIMABLE_SPACE	BIGINT	<p>この値は、DMS 表スペース内の索引にのみ適用されます。この値は、RECLAIM EXTENTS オプションを指定した <b>REORG INDEXES</b> または <b>REORG INDEX</b> コマンドを実行することによって索引オブジェクト全体から再利用できるディスク・スペースの見積もりです (キロバイト単位)。DMS 表スペースに定義されていない索引については、値はゼロです。この関数を実行したときに表または索引でエラーが発生した場合、この値は NULL です。</p>

## ADMIN\_GET\_INTRA\_PARALLEL - パーティション内並列処理の状態の取得

ADMIN\_GET\_INTRA\_PARALLEL スカラー関数は、アプリケーションのパーティション内並列処理の現在の状態を戻します。これを使用して、現行ステートメントが並列化照会アクセス・プランで実行されているかどうかを検査できます。

### 構文

```
▶▶ ADMIN_GET_INTRA_PARALLEL (—state—) ◀◀
```

スキーマは SYSPROC です。

### 状態

データベース・アプリケーションのパーティション内並列処理の現在の状態を指定する、タイプ VARCHAR(3) の出力引数。引数の値は、以下のいずれかになります。

- YES** データベース・アプリケーションは、パーティション内並列処理が有効な状態で実行されます。
- NO** データベース・アプリケーションは、パーティション内並列処理が無効な状態で実行されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

DB2 CLP からパーティション内並列処理の現在の状態を検出します。

```
VALUES SYSPROC.ADMIN_GET_INTRA_PARALLEL
```

```
1  
---  
NO
```

```
1 record(s) selected.
```

### 使用上の注意

以下のケースでは、ADMIN\_GET\_INTRA\_PARALLEL によって戻される値が ADMIN\_SET\_INTRA\_PARALLEL で設定された状態と異なる場合があります。

- トランザクション中に ADMIN\_SET\_INTRA\_PARALLEL が呼び出され、そのトランザクションがまだコミット/ロールバックされていない。



- WITH HOLD カーソルを開くトランザクション中に ADMIN\_SET\_INTRA\_PARALLEL が呼び出されて、現行トランザクションがそのカーソルをまだ閉じていない。
- ADMIN\_SET\_INTRA\_PARALLEL 呼び出しにより指定される値とは異なる MAX DEGREE ワークロード属性値が適用された DB2 ワークロードに、アプリケーションが関連付けられている。

## ADMIN\_GET\_MEM\_USAGE 表関数 - インスタンスの合計メモリー消費量の取得

ADMIN\_GET\_MEM\_USAGE 表関数は、指定されたインスタンスの合計メモリー消費量を取得します。

### 構文

```

▶▶ ADMIN_GET_MEM_USAGE ( ( member ) )

```

スキーマは SYSPROC です。

### 表関数パラメーター

#### member

メモリー使用量統計が取得されるメンバーを指定する、整数タイプのオプション入力引数。 -1 または NULL 値を指定した場合は、現在接続されているメンバーのデータが戻されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 戻される情報

表 66. ADMIN\_GET\_MEM\_USAGE について戻される情報

列名	データ・タイプ	説明
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
MAX_MEMBER_MEM	BIGINT	メンバーが使用できる最大メモリー量 (バイト単位)。
CURRENT_MEMBER_MEM	BIGINT	メンバーによって現在使用されているメモリー量 (バイト単位)。

表 66. ADMIN\_GET\_MEM\_USAGE について戻される情報 (続き)

列名	データ・タイプ	説明
PEAK_MEMBER_MEM	BIGINT	インスタンスが開始されて以降、メンバーによって使用されたメモリーのピークつまり最高水準点 (バイト単位)。

## 例

例 1: すべてのメンバーのメモリー使用量を報告します。

```
SELECT MEMBER, MAX_MEMBER_MEM, CURRENT_MEMBER_MEM, PEAK_MEMBER_MEM
FROM TABLE(SYSPROC.ADMIN_GET_MEM_USAGE()) AS T
```

```
MEMBER MAX_MEMBER_MEM      CURRENT_MEMBER_MEM  PEAK_MEMBER_MEM
-----
0       7430103040       958169088          958300160
3       7430103040       951615488          951615488
1       7430103040       952664064          952664064
2       7430103040       951615488          951615488
```

4 record(s) selected.

## ADMIN\_GET\_MSGS 表関数 - ADMIN\_CMD プロシージャを通して実行するデータ移動ユーティリティによって生成されたメッセージの検索

ADMIN\_GET\_MSGS 表関数は、ADMIN\_CMD プロシージャによるデータ移動ユーティリティ・コマンドの単一の実行によって生成されたメッセージを検索するために使用します。

入力パラメーター *operation\_id* によってこの操作が識別されます。

## 構文

```
▶▶—ADMIN_GET_MSGS—(—operation_id—)—————▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

*operation\_id*

ADMIN\_CMD プロシージャを通して実行したデータ移動ユーティリティによって作成された、メッセージ・ファイルの操作 ID を指定する、タイプ VARCHAR(139) の入力引数。操作 ID は、ADMIN\_CMD プロシージャによって生成されます。

## 許可

ADMIN\_GET\_MSGS 表関数に対する EXECUTE 特権。fenced ユーザー ID には、レジストリー変数 DB2\_UTIL\_MSGPATH によって示されるディレクトリーの下にあるファイルへの読み取りアクセスが必要です。レジストリー変数が設定されていない場合、fenced ユーザー ID には、instance ディレクトリーの tmp サブディレクトリーにあるファイルへの読み取りアクセスが必要です。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

ADMIN\_CMD プロシージャを通して実行した EXPORT ユーティリティによって戻された、操作 ID '24523\_THERESAX' のすべてのメッセージを確認します。

```
SELECT * FROM TABLE(SYSPROC.ADMIN_GET_MSGS('24523_THERESAX')) AS MSG
```

以下は、この照会からの出力例です。

```
DBPARTITIONNUM AGENTTYPE SQLCODE  MSG
-----
-                -          SQL3104N  The Export utility is beginning to
                        export data to file
                        "/home/theresax/rtest/data/ac_load03.del".
-                -          SQL3105N  The Export utility has finished
                        exporting "8" rows.
```

2 record(s) selected.

### 使用上の注意

適切な *operation\_id* を指定してこの表関数を呼び出す照会ステートメントは、ADMIN\_CMD プロシージャが戻す最初の結果セットの MSG\_RETRIEVAL 列に示されます。

### 戻される情報

表 67. ADMIN\_GET\_MSGS 表関数によって戻される情報

列名	データ・タイプ	説明
DBPARTITIONNUM	INTEGER	dbpartitionnum - データベース・パーティション番号モニター・エレメント
AGENTTYPE	CHAR(4)	エージェント・タイプ。この値は、分散ロードの場合にのみ戻されます。可能な値は次のとおりです。 <ul style="list-style-type: none"><li>• 'LOAD': ロード・エージェント</li><li>• 'PART': パーティション・エージェント</li><li>• 'PREP': 事前パーティション・エージェント</li><li>• NULL: 使用可能なエージェント・タイプ情報なし</li></ul>
SQLCODE	VARCHAR(9)	戻されるメッセージの SQLCODE。
MSG	VARCHAR(1024)	SQLCODE に対応する簡略エラー・メッセージ。

## ADMIN\_GET\_STORAGE\_PATHS 表関数 - 自動ストレージ・パス情報の取得

ADMIN\_GET\_STORAGE\_PATHS 表関数は、各データベース・ストレージ・グループの自動ストレージ・パスのリストを戻します。これには、各ストレージ・パスのファイル・システム情報が含まれます。

戻される可能性のある情報の完全なリストは、255 ページの表 68 を参照してください。

### 構文

```
▶▶—ADMIN_GET_STORAGE_PATHS—(—storage_group_name—,—member—)————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *storage\_group\_name*

この関数が呼び出されるときに現在接続されているデータベースでの有効なストレージ・グループ名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのストレージ・グループの情報が戻されます。引数を指定した場合は、識別されたストレージ・グループの情報のみが戻されます。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーの場合は -1、すべてのデータベース・メンバーの場合は -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

ドロップ後も引き続き使用されているストレージ・パスを判別します。

```
SELECT VARCHAR(STORAGE_GROUP_NAME, 30) AS STOGROUP, VARCHAR(DB_STORAGE_PATH, 40)
      AS STORAGE_PATH FROM TABLE(ADMIN_GET_STORAGE_PATHS('',-1)) AS T
WHERE DB_STORAGE_PATH_STATE = 'DROP_PENDING'
```

以下はこの照会の出力例です。

```

STOGROUP          STORAGE_PATH
-----
HOTSTORAGE        /home/hotel55/hotpath1

```

1 record(s) selected.

現在接続されているデータベースのすべてのストレージ・パスをリストします。

```

SELECT VARCHAR(STORAGE_GROUP_NAME, 30) AS STOGROUP, VARCHAR(DB_STORAGE_PATH, 40)
      AS STORAGE_PATH FROM TABLE(ADMIN_GET_STORAGE_PATHS('','-1')) AS T

```

以下はこの照会の出力例です。

```

STOGROUP          STORAGE_PATH
-----
IBMSTOGROUP       /home/hotel55/instowner
HOTSTORAGE        /home/hotel55/hotpath1
COLDSTORAGE       /home/hotel55/colddpath1

```

3 record(s) selected.

## ADMIN\_GET\_STORAGE\_PATHS によって戻される情報

表 68. ADMIN\_GET\_STORAGE\_PATHS 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
STORAGE_GROUP_NAME	VARCHAR(128)	storage_group_name - ストレージ・グループ名
STORAGE_GROUP_ID	INTEGER	storage_group_id - ストレージ・グループ ID
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号
DB_STORAGE_PATH	VARCHAR(256)	db_storage_path - 自動ストレージ・パス
DB_STORAGE_PATH_WITH_DPE	VARCHAR(256)	db_storage_path_with_dpe - データベース・パーティション式を含むデータベース・ストレージ・パス
DB_STORAGE_PATH_STATE	VARCHAR(16)	db_storage_path_state - ストレージ・パスの状態  値は以下のいずれかです。  <ul style="list-style-type: none"> <li>• IN_USE</li> <li>• NOT_IN_USE</li> <li>• DROP_PENDING</li> </ul>
DB_STORAGE_PATH_ID	BIGINT	db_storage_path_id - ストレージ・パス ID
FS_ID	VARCHAR(22)	fs_id - 固有のファイル・システム識別番号
FS_TOTAL_SIZE	BIGINT	fs_total_size - ファイル・システムの合計サイズ
FS_USED_SIZE	BIGINT	fs_used_size - ファイル・システム上で使用されるスペースの量

表 68. ADMIN\_GET\_STORAGE\_PATHS 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
STO_PATH_FREE_SIZE	BIGINT	sto_path_free_size - 自動ストレージ・パスのフリー・スペース

## ADMIN\_GET\_TAB\_COMPRESS\_INFO

### 表関数 - 圧縮による節約の見積もり

ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数は、RESETDICTIONARY オプションを指定した REORG が実行されると仮定して、圧縮により得られる表の節約を見積もります。

この表関数は、以前のバージョンの DB2 for Linux, UNIX, and Windows における非推奨 ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数での「ESTIMATE」モードに直接置き代わるものです。

### 構文

```
▶▶—ADMIN_GET_TAB_COMPRESS_INFO—(—tabschema—,—tablename—)————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *tabschema*

スキーマ名を指定する、タイプ VARCHAR (128) の入力引数。

#### *tablename*

表の名前、マテリアライズ照会表の名前、または階層表の名前を指定するタイプ VARCHAR(128) の入力引数。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 戻される情報

表 69. ADMIN\_GET\_TAB\_COMPRESS\_INFO で戻される情報

列名	データ・タイプ	説明
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント

表 69. ADMIN\_GET\_TAB\_COMPRESS\_INFO で戻される情報 (続き)

列名	データ・タイプ	説明
TABNAME	VARCHAR(128)	table_name - 表名モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
DATAPARTITIONID	INTEGER	データ・パーティション番号。
OBJECT_TYPE	VARCHAR(4)	objtype - オブジェクト・タイプ・モニター・エレメント
ROWCOMPMode	CHAR(1)	オブジェクトの現在の行圧縮モード。戻されるメトリックは、以下のいずれかの値になります。 <ul style="list-style-type: none"> <li>クラシック行圧縮が使用可能な場合は「S」</li> <li>アダプティブ行圧縮が使用可能な場合は「A」</li> <li>行圧縮が使用可能でない場合はブランク</li> </ul>
PCTPAGESSAVED_CURRENT	SMALLINT	行圧縮により節約されるページの現在のパーセンテージ。
AVGROWSIZE_CURRENT	SMALLINT	現在の平均レコード長。
PCTPAGESSAVED_STATIC	SMALLINT	クラシック行圧縮により節約されるページのパーセンテージの見積もり。
AVGROWSIZE_STATIC	SMALLINT	クラシック行圧縮による平均レコード長の見積もり。
PCTPAGESSAVED_ADAPTIVE	SMALLINT	アダプティブ行圧縮により節約されるページのパーセンテージの見積もり。
AVGROWSIZE_ADAPTIVE	SMALLINT	アダプティブ行圧縮による平均レコード長の見積もり。

### 使用上の注意

- *tabschema* と *tablename* の両方が指定される場合、その特定の表の情報のみが戻されます。
- *tabschema* が指定され、*tablename* が空 (") または NULL の場合、指定したスキーマのすべての表に関する情報が戻されます。
- *tabschema* が空 (") または NULL で、*tablename* が指定される場合、エラーは戻されます。特定の表の情報を取り出すには、その表がスキーマと表名の両方によって識別されることが必要です。
- *tabschema* と *tablename* の両方が空 (") または NULL の場合、すべての表の情報が戻されます。
- *tabschema* または *tablename* が存在しないか、あるいは *tablename* が表名 (タイプ T) またはマテリアライズ照会表名 (タイプ S) と一致しない場合、空の結果セットが戻されます。
- ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数が指定の表のデータを検索するとき、この表関数は SYSTABLES の対応する行に対する共有ロックを獲得します。

これは、戻されるデータの整合性を確保するための動作です (例えば、情報の検索中に、検索されている表が変更されないようにするなど)。ロックが保持されるのは、表関数の呼び出し期間中ではなく、表の圧縮情報を検索する間だけです。

- 指定された表に XML 列が 1 つ以上ある場合、  
ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数はパーティションごとに 2 行返します。OBJECT\_TYPE として「DATA」を返す 1 行と、OBJECT\_TYPE として「XML」を返す 1 行です。指定された表に XML 列がない場合は、OBJECT\_TYPE が「DATA」である行がパーティションごとに 1 行のみ返されず。
- XML オブジェクト・タイプの場合、PCTPAGESSAVED\_ADAPTIVE および PCTPAGESSAVED\_STATIC として戻される見積もりは等しくなります。アダプティブ圧縮は表のデータ部分にのみ適用されるからです。

## 例

例 1: SCHEMA1 スキーマの TABLE1 表の現在の圧縮結果と、クラシック行圧縮およびアダプティブ圧縮の両方の情報の見積レポートを表示します。

```
SELECT SUBSTR(TABSCHEMA, 1, 10) AS TABSCHEMA, SUBSTR(TABNAME, 1, 10) AS TABNAME,
       DBPARTITIONNUM, DATAPARTITIONID, OBJECT_TYPE, ROWCOMPMODE,
       PCTPAGESSAVED_CURRENT, AVGROWSIZE_CURRENT,
       PCTPAGESSAVED_STATIC, AVGROWSIZE_STATIC,
       PCTPAGESSAVED_ADAPTIVE, AVGROWSIZE_ADAPTIVE
FROM TABLE(SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO('SCHEMA1', 'TABLE1'))
```

この照会からの出力:

TABSCHEMA	TABNAME	DBPARTITIONNUM	DATAPARTITIONID	OBJECT_TYPE	ROWCOMPMODE	...
SCHEMA1	TABLE1	0	0	DATA	A	...
SCHEMA1	TABLE1	0	0	XML	S	...

PCTPAGESSAVED_CURRENT	AVGROWSIZE_CURRENT	PCTPAGESSAVED_STATIC	AVGROWSIZE_STATIC	...
60	40	68	34	...
58	255	62	198	...

PCTPAGESSAVED_ADAPTIVE	AVGROWSIZE_ADAPTIVE
70	30
62	198

2 record(s) selected.

## ADMIN\_GET\_TAB\_DICTIONARY\_INFO

### 表関数 - 既存の表ディクショナリーのプロパティの報告

ADMIN\_GET\_TAB\_DICTIONARY\_INFO 表関数は、表ディクショナリーが作成されたときの、指定されたスキーマと表に関するクラシック行圧縮のディクショナリー情報を報告します。

これは、以前のバージョンの DB2 for Linux, UNIX, and Windows における非推奨 ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数の「REPORT」モードに直接置き代わるものです。



## 構文

▶▶—ADMIN\_GET\_TAB\_DICTIONARY\_INFO—(—*tabschema*—,—*tabname*—)————▶▶

スキーマは SYSPROC です。

## 表関数パラメーター

### *tabschema*

スキーマ名を指定する、タイプ VARCHAR (128) の入力引数。

### *tabname*

表の名前、マテリアライズ照会表の名前、または階層表の名前を指定するタイプ VARCHAR(128) の入力引数。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 使用上の注意

- *tabschema* と *tabname* の両方が指定される場合、その特定の表の情報のみが戻されます。
- *tabschema* が指定され、*tabname* が空 (") または NULL の場合、指定したスキーマのすべての表に関する情報が戻されます。
- *tabschema* が空 (") または NULL で、*tabname* が指定される場合、エラーは戻されます。特定の表の情報を取り出すには、その表がスキーマと表名の両方によって識別されることが必要です。
- *tabschema* と *tabname* の両方が空 (") または NULL の場合、すべての表の情報が戻されます。
- *tabschema* または *tabname* が存在しないか、あるいは *tabname* が表名 (タイプ T) またはマテリアライズ照会表名 (タイプ S) と一致しない場合、空の結果セットが戻されます。
- 指定された表に XML 列が 1 つ以上ある場合、ADMIN\_GET\_TAB\_DICTIONARY\_INFO 表関数はパーティションごとに 2 行返します。OBJECT\_TYPE として「DATA」を返す 1 行と、OBJECT\_TYPE として「XML」を返す 1 行です。指定された表に XML 列がない場合は、OBJECT\_TYPE が「DATA」である行がパーティションごとに 1 行のみ返されず。
- ADMIN\_GET\_TAB\_DICTIONARY\_INFO 表関数が指定の表のデータを検索するとき、この表関数は SYSTABLES の対応する行に対する共有ロックを獲得します。

これは、戻されるデータの整合性を確保するための動作です (例えば、情報の検索中に、検索されている表が変更されないようにするなど)。ロックが保持されるのは、表関数の呼び出し期間中ではなく、表の圧縮情報を検索する間だけです。

### 戻される情報

表 70. ADMIN\_GET\_TAB\_DICTIONARY\_INFO によって戻される情報

列名	データ・タイプ	説明
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント
TABNAME	VARCHAR(128)	table_name - 表名モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
DATAPARTITIONID	INTEGER	データ・パーティション番号。
OBJECT_TYPE	VARCHAR(4)	objtype - オブジェクト・タイプ・モニター・エレメント
ROWCOMPMODE	CHAR(1)	オブジェクトの現在の行圧縮モード。戻されるメトリックは、以下のいずれかの値になります。 <ul style="list-style-type: none"> <li>クラシック行圧縮が使用可能な場合は「S」</li> <li>アダプティブ行圧縮が使用可能な場合は「A」</li> <li>行圧縮が使用可能でない場合はブランク</li> </ul>
BUILDER	VARCHAR(30)	ディクショナリーを作成するために使われるコード・パス。以下の値のいずれかになります。 <ul style="list-style-type: none"> <li>'INSPECT' = INSPECT ROWCOMPESTIMATE</li> <li>'LOAD' = LOAD INSERT/REPLACE</li> <li>'NOT BUILT' = 使用可能なディクショナリーがありません</li> <li>'REDISTRIBUTE' = REDISTRIBUTE</li> <li>'REORG' = REORG RESETDICTIONARY</li> <li>'TABLE GROWTH' = INSERT</li> </ul>
BUILD_TIMESTAMP	TIMESTAMP	ディクショナリーが作成された時刻のタイム・スタンプ。タイム・スタンプの細分性は秒単位です。使用可能なディクショナリーがない場合、タイム・スタンプは NULL です。

表 70. ADMIN\_GET\_TAB\_DICTIONARY\_INFO によって戻される情報 (続き)

列名	データ・タイプ	説明
SIZE	BIGINT	バイト単位で測定されたエクспанション・ディクショナリーのサイズ。履歴ディクショナリーが存在する場合、この値は現在のディクショナリー・サイズと履歴ディクショナリー・サイズの合計になります。
HISTORICAL_DICTIONARY	CHAR(1)	履歴ディクショナリーの存在を示します。戻されるメトリックは、以下のいずれかの値になります。 <ul style="list-style-type: none"> <li>「N」 = 履歴ディクショナリーは存在しません</li> <li>「Y」 = 履歴ディクショナリーが存在します</li> </ul>
ROWS_SAMPLED	INTEGER	ディクショナリーの作成に寄与したレコードの数。
PCTPAGESSAVED	SMALLINT	圧縮により節約されるページのパーセンテージ。この情報は、ディクショナリーの作成に寄与するレコードに基づく予測です。
AVGCOMPRESSEDROWSIZE	SMALLINT	ディクショナリーの作成に役立っているレコードの平均圧縮レコード長。

## 例

例 1。PAGECOMP スキーマの ADMIN\_VIEW 表のディクショナリー情報レポートを表示します。

```
SELECT SUBSTR(TABSCHEMA, 1, 10) AS TABSCHEMA, SUBSTR(TABNAME, 1, 10) AS TABNAME,
       DBPARTITIONNUM, DATAPARTITIONID, OBJECT_TYPE, ROWCOMPMODE, BUILDER,
       BUILD_TIMESTAMP, SIZE, HISTORICAL_DICTIONARY, ROWS_SAMPLED,
       PCTPAGESSAVED, AVGCOMPRESSEDROWSIZE
FROM TABLE( SYSPROC.ADMIN_GET_TAB_DICTIONARY_INFO( 'PAGECOMP', 'ADMIN_VIEW' ) )
```

この照会からの出力:

```
TABSCHEMA TABNAME DBPARTITIONNUM DATAPARTITIONID OBJECT_TYPE ROWCOMPMODE ...
-----
PAGECOMP ADMIN_VIEW 0 0 DATA S ...

BUILDER BUILD_TIMESTAMP SIZE
-----
REORG 2010-09-03-01.10.33.000000 52736 ...

HISTORICAL_DICTIONARY ROWS_SAMPLED PCTPAGESSAVED AVGCOMPRESSEDROWSIZE
-----
N 300000 80 38
```

1 record(s) selected.

## ADMIN\_IS\_INLINED 関数 - データがインラインかどうかを判別する

ADMIN\_IS\_INLINED 関数は、XML 列、BLOB 列、CLOB 列、または DBCLOB 列のインライン・データに関する状態情報を取得します。

## 構文

▶▶—ADMIN\_IS\_INLINED—(—column-name—)————▶▶

スキーマは SYSIBM です。

## 戻り値

この関数は、タイプ SMALLINT の以下のいずれかの値、あるいは NULL 値を返します。

1 データがインライン化されることを示します。

0 データがインライン化されないことを示します。

NULL 入力が NULL であることを示します。

## 関数のパラメーター

*column-name*

データ・タイプ XML、BLOB、CLOB、または DBCLOB を持つ基本表の列を識別します (SQLSTATE 42884)。この列は、式に基づいて生成されていない基本表の列を直接的または間接的に参照する必要があります (SQLSTATE 42815)。

## 例

例 1: 以下の例は、TAB1 表の XML 列 xml\_doc1 の 3 つの XML 文書がインライン化されるかどうかを示します。

```
db2 => SELECT PK, ADMIN_IS_INLINED(xml_doc1) as IS_INLINED
        from TAB1
```

この照会では、以下の出力が結果として返ります。

PK	IS_INLINED
1	1
2	0
3	0

3 record(s) selected.

例 2: 以下の例は、TAB1 表の XML 列 xml\_doc1 の XML 文書のいずれかがインライン化されるかどうかを示します。

```
db2 => SELECT PK, ADMIN_IS_INLINED(xml_doc1) as IS_INLINED
        from TAB1 where PK=1
```

この照会では、以下の出力が結果として返ります。

PK	IS_INLINED
1	1

1 record(s) selected.

例 3: TAB1 表の CLOB 列に含まれている 3 つの CLOB データがインライン化されるかどうかを示します。

```
db2 => SELECT PK, ADMIN_IS_INLINED(clob_1) as IS_INLINED
       from TAB1
```

この照会では、以下の出力が結果として戻ります。

PK	IS_INLINED
1	0
2	0
3	1

3 record(s) selected.

## ADMIN\_MOVE\_TABLE プロシージャ - オンラインの表移動

ADMIN\_MOVE\_TABLE ストアード・プロシージャは、アクティブな表のデータを、同じ名前の新しい表オブジェクトに移動します。データはオンラインのままなので、引き続きアクセス可能です。このストアード・プロシージャは、移動対象の表に関連する状況情報と構成オプションを内容とする行から成るプロトコル表を作成します。このプロシージャからの戻りセットは、移動対象の表に関連するプロトコル表からの行のセットです。

このストアード・プロシージャでは、以下の用語を使用します。

### ソース表

ストアード・プロシージャにパラメーターとして渡される元の表の名前。移動対象の表です。

### ターゲット表

ストアード・プロシージャで渡された表定義を使用して、ストアード・プロシージャによって作成される表。この表にソース表のすべてのデータがコピーされ、ソース表と同じ名前に名前変更されます。

### ステージング表

ストアード・プロシージャによって作成される表。表移動の実行中にソース表に対して行われる更新、削除、または挿入による変更はすべて、ステージング表に保管されます。この表は、移動が完了するとドロップされます。

## 構文

ADMIN\_MOVE\_TABLE ストアード・プロシージャを開始するには、2 つの同等に有効な方法があります。

- 1 つ目の方法では、ターゲット表の表定義の特定の部分のみを変更します。例えば、表定義がかなり大きい (数 KB) 場合、表の表スペースのみを変更したいときは、ソース表の再作成に必要な CREATE TABLE ステートメント全体を決定しなくても、その変更を行うことができます。 **data\_tbsp**、**index\_tbsp**、および **lob\_tbsp** パラメーターを設定するだけでよく、他のオプション・パラメーターはブランクのままにしておくことができます。
- 2 つ目の方法は、制御できる要素が多く、柔軟性の高い方法です。ターゲット表をストアード・プロシージャに作成させるのではなく、事前にターゲット表を作成することができます。ターゲット表を事前に作成すると、1 つ目の方法を使用して作成できないようなターゲット表を作成できます。

方法 1:

▶▶ ADMIN\_MOVE\_TABLE (—*tabschema*—, —*tabname*—, —*data\_tbsp*—, —*index\_tbsp*—, —

▶—*lob\_tbsp*—, —*organize\_by\_clause*—, —*partkey\_cols*—, —*data\_part*—, —*coldef*—, —

▶—*options*—, —*operation*—) —————▶▶

方法 2:

▶▶ ADMIN\_MOVE\_TABLE (—*tabschema*—, —*tabname*—, —*target\_tabname*—, —

▶—*options*—, —*operation*—) —————▶▶

両方式ともスキーマは SYSPROC です。

## プロシージャ・パラメーター

### *tabschema*

この入力パラメーターは、移動対象の表が含まれるスキーマの名前を指定します。このパラメーターには大/小文字の区別があり、データ・タイプは VARCHAR(128) です。

### *tabname*

この入力パラメーターは、移動対象の表の名前を指定します。このパラメーターには大/小文字の区別があり、データ・タイプは VARCHAR(128) です。

### *data\_tbsp*

この入力パラメーターは、ターゲット表の新しいデータ表スペースを指定します。値を指定する場合は、*index\_tbsp* および *lob\_tbsp* パラメーターが必須となります。値を指定しない場合は、ソース表のデータ表スペースが使用されます。このパラメーターには大/小文字の区別があり、データ・タイプは VARCHAR(128) です。このパラメーターを NULL または空ストリングにすることができます。

### *index\_tbsp*

この入力パラメーターは、ターゲット表の新しい索引表スペースを指定します。値を指定する場合は、*data\_tbsp* および *lob\_tbsp* パラメーターが必須となります。値を指定しない場合は、ソース表の索引表スペースが使用されます。このパラメーターには大/小文字の区別があり、データ・タイプは VARCHAR(128) です。このパラメーターを NULL または空ストリングにすることができます。

### *lob\_tbsp*

この入力パラメーターは、ターゲット表の新しい LOB 表スペースを指定します。値を指定する場合は、*data\_tbsp* および *index\_tbsp* パラメーターが必須となります。値を指定しない場合は、ソース表の LOB 表スペースが使用されます。

このパラメーターには大/小文字の区別があり、データ・タイプは VARCHAR(128) です。このパラメーターを NULL または空ストリングにすることができます。

#### *organize\_by\_clause*

この入力パラメーターを使用して、表に関する ORGANIZE BY 節を指定できません。指定した値が「ORGANIZE BY」で始まっていなければ、ターゲット表はマルチディメンション・クラスタリング (MDC) 仕様になります。ターゲット表のデータをマルチディメンションに従ってクラスタ化するために使用する列を決定し、それらをコンマ区切りリストにして値を入力します。値として NULL または「-」を指定した場合、ORGANIZE BY 節は使用されません。空ストリングまたは単一ブランクを指定すると、プロシージャはソース表に MDC または ITC 仕様があるかどうかを検査し、見つかった場合はその仕様を使用します。引数を「ORGANIZE BY」で始めた場合、これを使用して CREATE TABLE ステートメントの ORGANIZE BY 節に関連した任意のオプションを指定できます。このパラメーターは VARCHAR(32672) データ・タイプで、CREATE TABLE ステートメントの ORGANIZE BY DIMENSIONS 節と同じ形式です。このパラメーターを NULL、空ストリング、または単一ブランクにすることができます。

例 1: 'C1, C4, (C3,C1), C2'

例 2: ORGANIZE BY INSERT TIME

#### *partkey\_cols*

この入力パラメーターは、ターゲット表のパーティション・キー列仕様を指定します。複数のデータベース・パーティションにデータをどのように分散させるかを指定するキー列を決定し、それらをコンマ区切りリストにして値を入力します。値として NULL または「-」を指定した場合、PARTITIONING KEY 節は使用されません。空ストリングまたは単一ブランクを指定すると、プロシージャはソース表にパーティション・キー列仕様があるかどうかを検査し、見つかった場合はその仕様を使用します。このパラメーターは VARCHAR(32672) データ・タイプで、CREATE TABLE ステートメントの DISTRIBUTE BY HASH 節と同じ形式です。

例: 'C1, C3'

#### *data\_part*

この入力パラメーターは、ターゲット表のデータ・パーティション仕様を指定します。このステートメントは、表データを複数のストレージ・オブジェクト (データ・パーティションと言う) にどのように分割するかを、表の 1 つ以上の列の値に基づいて定義します。値として NULL または「-」を指定した場合、PARTITION BY RANGE 節は使用されません。空ストリングまたは単一ブランクを指定すると、プロシージャはソース表にデータ・パーティション・スキームがあるかどうかを検査し、見つかった場合はその情報 (パーティション名を含む) を使用します。このパラメーターは VARCHAR(32672) データ・タイプで、CREATE TABLE ステートメントの PARTITION BY RANGE 節と同じ形式です。

例: '(C1) (STARTING FROM (1) EXCLUSIVE ENDING AT (1000) EVERY (100))'

### *coldef*

この入力パラメーターは、ターゲット表の新しい列定義を指定します。列タイプの変更は互換性のある限り可能ですが、列名は同じままでなければなりません。

新しい列を追加したり、既存の列をドロップしたりすることもできます。列を追加する場合は、NULL 可能と定義するか、またはデフォルト値を設定する必要があります。また、表にユニーク索引または 1 次索引があり、ドロップされる列がそのユニーク索引または 1 次索引の一部でない場合のみ、列をドロップできます。このパラメーターのデータ・タイプは VARCHAR(32672) です。このパラメーターを NULL または空ストリングにすることができます。

例: 'C1 INT, C2 INT DEFAULT 0'

### *target\_tabname*

この入力パラメーターは、移動中にターゲット表として使用する既存の表の名前を指定します。渡されるターゲット表に対して、以下の変更を加えることができます。

- データ、索引、LOB の表スペースを変更できます。
- マルチディメンション列 (MDC) 仕様を追加または変更できます。
- パーティション・キー列仕様を追加または変更できます。
- データ・パーティション仕様を追加または変更できます。
- データ圧縮を追加または除去できます。
- 新しい列定義を指定できます。ただし、*coldef* パラメーターを指定する場合と同じ制限が適用されます。

指定する表には、以下の制限が適用されます。

- ソース表と同じスキーマに表が存在しなければなりません。
- 表は空でなければなりません。
- 型付き表、マテリアライズ照会表 (MQT)、ステージング表、リモート表、クラスター表は許可されません。

このパラメーターを NULL または空ストリングに設定した場合、ストアード・プロシージャはソース表と同じ定義を使用します。このパラメーターには大/小文字の区別があり、データ・タイプは VARCHAR(128) です。

### *options*

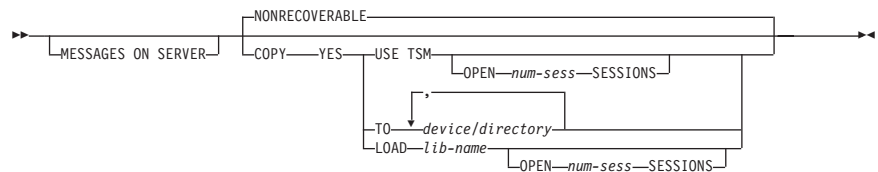
この入力パラメーターは、コンマ区切りの値で構成されたストリングで、ストアード・プロシージャによって使用されるオプションを定義します。このオプション・リストには大/小文字の区別がなく、データ・タイプは VARCHAR(32672) です。リスト値を NULL または空ストリングにすることができます。以下の値が有効です。

**KEEP** このオプションを設定すると、元のソース表のコピーが異なる名前でも保持されます。ソース表の名前が T1 の場合、移動後にこの表は自動的に T1AAAAVxo のような名前に変更されます。ソース表の正確な名前は、戻されるプロトコル表の ORIGINAL キーのフィールドに示されます。このオプションは、SWAP フェーズまでの (SWAP フェーズを含む) 任意の時点で設定できます。

**COPY\_USE\_LOAD “<load options>”**



## COPY\_USE\_LOAD の構文



COPY\_USE\_LOAD でロード・オプションを指定した場合、ADMIN\_MOVE\_TABLE は ADMIN\_CMD ロードを使用してソース表からターゲット表にデータをコピーします。COPY\_USE\_LOAD でどのオプションも指定しない場合は、ソース表からターゲット表へのデータのコピーに NONRECOVERABLE オプションの db2Load API が使用されます。DB2 バージョン 9.7 フィックスパック 2 より前のリリースでは、COPY\_USE\_LOAD を使用する場合は FORCE オプションを指定する必要があります。

### MESSAGES ON SERVER

**LOAD** コマンドによってサーバー上に作成されるメッセージ・ファイルを、ロード失敗に備えて保持することを指定します。プロトコル表の WARNINGS 項目には、ロード中に発生したすべての警告およびエラー・メッセージを取り出すのに必要なメッセージ検索 SQL ステートメントと、メッセージをクリーンアップするのに必要なメッセージ除去 SQL ステートメントが含まれます。この節を使用するしないにかかわらず、fenced ユーザー ID は、DB2\_UTIL\_MSGPATH レジストリー変数で示されるディレクトリー下にファイルを作成できる権限を持っていなければなりません。

### COPY YES

ロードするデータのコピーを保存することを指定します。フォワード・リカバリーが使用不可になっている場合、このオプションは無効です。

### USE TSM

Tivoli Storage Manager (TSM) を使ってコピーを保管することを指定します。

### OPEN num-sess SESSIONS

TSM またはベンダー製品とともに使用する入出力セッションの数です。デフォルト値は 1 です。

### TO device または directory

コピー・イメージを作成する先の装置またはディレクトリーを指定します。

### LOAD lib-name

使用するバックアップおよびリストア I/O ベンダー関数を含む共有ライブラリー (Windows オペレーティング・システムでは DLL) の名前。絶対パスで指定することができます。絶対パスを指定しない場合、デフォルトでユーザー出口プログラムの存在するパスになります。

### **NONRECOVERABLE**

ロード・トランザクションがリカバリー不能としてマークされており、それ以降のロールフォワード・アクションによってそれをリカバリーさせることは不可能であることを指定します。COPY YES が使用されない場合は NONRECOVERABLE がデフォルトです。

### **COPY\_WITH\_INDEXES**

このオプションを設定すると、ソース表のコピー前に索引が作成されます。一方デフォルトでは、ソース表のコピー後に索引が作成されます。このオプションの利点は、コピー後の索引作成の場合は索引ごとに表全体スキャンが必要になるということ、および索引作成はアクティブ・ログ・スペースを必要とするトランザクションであるということです。LOGINDEXREBUILD データベース構成パラメーターがオンの場合、索引を短い時間枠で作成するためには、かなりのログ・スペースが必要になります。このオプションの欠点の 1 つは、索引をターゲット表で維持する必要があるために、コピー・パフォーマンスが低下するという点です。また、結果として生成される索引は疑似削除済み のキーを多く含んでおり、コピー後に索引を作成した場合ほど索引のバランスが取れていません。COPY\_WITH\_INDEXES オプションは、COPY フェーズまでの (COPY フェーズを含む) 任意の時点で設定できます。

### **FORCE**

この強制オプションを設定すると、ソース表の表定義が変わったかどうか SWAP フェーズで検査されません。DB2 バージョン 9.7 フィックスパック 2 より前のリリースでは、COPY\_USE\_LOAD を使用する場合は FORCE オプションを指定しなければなりません。このオプションは、SWAP フェーズまでの (SWAP フェーズを含む) 任意の時点で設定できます。

### **NO\_STATS**

このオプションを設定すると、ターゲット表での RUNSTATS または統計コピーは開始されません。AUTO\_RUNSTATS または AUTO\_STMT\_STATS データベース構成パラメーターを使用すると、その後 DB2 が自動的に新しい統計を作成します。後方互換性のために STATS\_NO も受け入れます。NO\_STATS オプションは、SWAP フェーズまでの (SWAP フェーズを含む) 任意の時点で設定できます。

### **COPY\_STATS**

このオプションを設定すると、スワップ実行前にソース表からターゲット表に統計がコピーされます。これにより、ページ・サイズが変わる場合は特に、物理統計が不正確になる可能性があります。しかし、このオプションを設定すると、新しい統計を計算するために RUNSTATS が呼び出されることがないので、計算時間を節約できます。また、同じ統計になるので、オプティマイザーが同じアクセス・プランを選択する可能性があります。後方互換性のために STATS\_COPY も受け入れます。STATS\_COPY オプションは、SWAP フェーズまでの (SWAP フェーズを含む) 任意の時点で設定できます。

### **NO\_AUTO\_REVAL**

このオプションを設定すると、表に関する自動再有効化は行われませ

ん。代わりに、すべてのトリガーとビューが再作成されます。  
NO\_AUTO\_REVAL オプションは INIT フェーズでのみ設定できます。

### REORG

このオプションを設定すると、スワップ実行前にターゲット表に対する補足的なオフライン REORG がセットアップされます。このオプションを使用してコンプレッション・ディクショナリーを改良することも考えられますが、最適なコンプレッション・ディクショナリーを作成する方法としては、デフォルトのサンプリング・アプローチの方が適していると言えます。ただし、最適な XML コンプレッション・ディクショナリーを必要とする場合は、REORG が唯一の方法です。REORG オプションは、SWAP フェーズまでの (SWAP フェーズを含む) 任意の時点で設定できます。

### NO\_TARGET\_LOCKSIZE\_TABLE

このオプションは、COPY および SWAP フェーズ中にターゲット表で LOCKSIZE 表オプションを保持しません。デフォルトでは、ソース表でユニーク索引が指定されていない場合、ロック・オーバーヘッドを防ぐためにターゲット表で LOCKSIZE 表オプションを使用します。

### CLUSTER

このオプションは、ADMIN\_MOVE\_TABLE\_UTIL を使用してコピー索引が指定された場合で、ソース表にクラスタリング索引が存在するか、ソース表でユニーク索引または主キーが定義されている場合に、ORDER BY 節を使用してソース表からデータを読み取ります。

注: コピー索引はクラスタリング索引をオーバーライドします。クラスタリング索引は主キーより優先して使用されます。主キーはユニーク索引より優先して使用されます。

### NON\_CLUSTER

このオプションは、コピー索引が指定されたかどうか、ソース表にクラスタリング索引が存在するかどうか、ソース表でユニーク索引または主キーが定義されているかどうかにかかわらずに、ORDER BY 節を使用せずにソース表からデータを読み取ります。

注: CLUSTER オプションも NON\_CLUSTER オプションも指定しない場合、ADMIN\_MOVE\_TABLE によるソース表からのデータの読み取りで ORDER BY 節が使用されるのは、ソース表にクラスタリング索引が存在する場合のみです。

### LOAD\_MSGPATH <path>

このオプションは、ロード・メッセージ・ファイル・パスを定義するために使用できます。

LOAD\_MSGPATH は、以下の状況において、COPY\_USE\_LOAD と一緒に使用できます。

- COPY\_USE\_LOAD が追加のオプションなしで指定されている (つまり、デフォルトの COPY\_USE\_LOAD NONRECOVERABLE が想定されている)
- COPY\_USE\_LOAD NONRECOVERABLE が明示的に指定されている

LOAD\_MSGPATH は、COPY\_USE\_LOAD とともにオプションのキーワードも指定されている場合は、COPY\_USE\_LOAD と一緒に使用することはできません。

LOAD\_MSGPATH オプションが指定されていない場合、デフォルトのパスは、DB2\_UTIL\_MSGPATH レジストリー変数によって決定されます。

#### *operation*

この入力パラメーターは、ストアード・プロシージャが実行する操作を指定します。ストアード・プロシージャを呼び出す方法は 2 通りあります。MOVE コマンドを使用してすべての操作を一度に実行する方法と、個々のコマンドを使用して表移動を 1 ステップずつ実行する方法です。この 2 番目の方法の主な利点は、SWAP フェーズを実際にいつ行うかを制御できるので、表をいつ一時的にオフラインにするかを決定できるという点です。これにより、システム・アクティビティーが低い間に移動を行うことができます。個々のコマンドを使用する場合は、INIT、COPY、REPLAY、VERIFY (オプション)、SWAP の順に呼び出す必要があります。

- MOVE: 表移動全体 (INIT、COPY、REPLAY、および SWAP 操作) を 1 ステップで実行します。
- INIT: 表移動を行えることを検査し、表移動の過程で必要なすべてのデータ (ターゲット表、ステージング表、およびソース表のトリガー) を初期化します。
- COPY: ソース表の内容をターゲット表にコピーします。その間ソース表で行われる更新、削除、挿入はキャプチャーされ、ステージング表に保管されます。COPY\_WITH\_INDEXES オプションが選択されていなければ、COPY フェーズの最後に新しい索引が作成されます。パフォーマンスを上げるために必要であれば、REPLAY フェーズでソース表とターゲット表の副次索引も作成されます。COPY を使用できるのは、INIT フェーズが完了した後に限られます。
- REDIRECT: ステージング表で変更をキャプチャーするのではなく、直接ターゲット表に変更を転送します。

**注:** REDIRECT コマンドは、マルチパーティション・システムでは、ユニーク索引を持たない表に対して機能しません。

- REVERT: ステージング表で変更をキャプチャーするオリジナルの動作に戻します。
- REPLAY: COPY フェーズの開始以降ソース表で変更された行を、ターゲット表にコピーします。REPLAY を使用できるのは、COPY フェーズが完了した後に限られます。
- VERIFY: (オプション) ソース表とターゲット表の表の内容が等しいかどうかを検査します。このプロセスでは、まずソース表とターゲット表の共有ロックが獲得されます。次にソース表で行われた変更が再生された後に、比較が行われます。表にユニーク索引がある場合、このコマンドは両方の表にある列どうしですべての値を比較します。そうでない場合、このコマンドは両方の表にある列 (LONG、LOB、または XML 列を除く) どうしですべての値を比較します。これは高コストの操作になるので、移動に有用かどうかを慎重

に判断する必要があります。VERIFY を使用できるのは、COPY または REPLAY フェーズが完了した後に限られます。

- **SWAP:** ステージング表の最後のスキャン中に適用された変更の数が、プロトコル表に保管されている REPLAY\_THRESHOLD 値より小さくなるまで、REPLAY フェーズを実行します。その後ソース表が一時的にオフラインになって最終 REPLAY が完了します。次にこのコマンドはソース表とターゲット表をスワップし、表をオンラインに戻します。SWAP を使用できるのは COPY フェーズが完了した後ですが、理想的には REPLAY フェーズが呼び出された後です。
- **CLEANUP:** ステージング表と、ストアード・プロシージャによってソース表に対して作成された非ユニーク索引またはトリガーをドロップします。さらに、KEEP オプションが設定されていない場合は、ソース表もドロップします。CLEANUP を呼び出すことができるのは、SWAP フェーズでコマンドが失敗した場合です。
- **CANCEL:** 複数ステップの表移動をフェーズとフェーズの間で取り消します。または、失敗した表移動操作を取り消します。このコマンドを実行するためには、操作状況が COMPLETED 状態でも CLEANUP 状態でもないことが必要です。CANCEL は、すべての中間データ (索引、ステージング表、ターゲット表、およびソース表のトリガー) をクリアします。

このパラメーターには大/小文字の区別がなく、データ・タイプは VARCHAR(128) です。

## 許可

ADMIN\_MOVE\_TABLE ストアード・プロシージャを呼び出すためには、SQLADM 権限または DBADM 権限が必要です。また、SELECT ステートメントをソース表に発行したり、ターゲット表に INSERT ステートメントを発行したりする権限を含め、適切なオブジェクト作成権限もなければなりません。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

この例は、最初の方法でストアード・プロシージャを呼び出し、スキーマ SVALENTI にある T1 という名前の表を移動します。ここでは、ターゲット表がプロシージャ内で定義されます。

```
CALL SYSPROC.ADMIN_MOVE_TABLE(  
'SVALENTI',  
'T1',  
'ACCOUNTING',  
'ACCOUNT_IDX',  
'ACCOUNT_LONG',  
'',  
'',  
'',  
'CUSTOMER VARCHAR(80), REGION CHAR(5), YEAR INTEGER, CONTENTS CLOB',  
'',  
'MOVE')
```

以下はこの照会の出力例です

Result set 1

-----

KEY	VALUE
AUTHID	SVALENTI
CLEANUP_END	2009-02-13-11.34.07.609575
CLEANUP_START	2009-02-13-11.34.07.369331
COPY_END	2009-02-13-11.34.05.148018
COPY_OPTS	BY_KEY,OVER_INDEX
COPY_START	2009-02-13-11.34.04.841292
COPY_TOTAL_ROWS	100
INDEXNAME	T1_INDEX
INDEXSCHEMA	SVALENTI
INDEX_CREATION_TOTAL_TIME	0
INIT_END	2009-02-13-11.34.04.552875
INIT_START	2009-02-13-11.34.03.013563
PAR_COLDEF	CUSTOMER VARCHAR(80), REGION CHAR(5), YEAR INTEGER, CONTENTS CLOB
REPLAY_END	2009-02-13-11.34.06.198369
REPLAY_START	2009-02-13-11.34.05.164582
REPLAY_TOTAL_ROWS	100
REPLAY_TOTAL_TIME	5
STATUS	COMPLETE
SWAP_END	2009-02-12-11.34.07.214447
SWAP_RETRIES	0
SWAP_START	2009-02-13-11.34.06.244506
VERSION	09.07.0000

22 record(s) selected.

Return Status = 0

この例は、2 番目の方法でストアード・プロシージャを呼び出し、先ほどの例と同じ表を移動します。ここでは、ターゲット表がプロシージャ外で作成され、その後 *target\_tabname* パラメーター内の名前が付けられます。

最初のステップとして、手動で表を作成します。

```
CREATE TABLE SVALENTI.T1_TARGET (  
    CUSTOMER VARCHAR(80),  
    REGION CHAR(5),  
    YEAR INTEGER,  
    CONTENTS CLOB)  
IN ACCOUNTING  
INDEX IN ACCOUNT_IDX  
LONG IN ACCOUNT_LONG'
```

次に、ストアード・プロシージャを呼び出し、ターゲット表の名前を渡します。

```
CALL SYSPROC.ADMIN_MOVE_TABLE(  
'SVALENTI',  
'T1',  
'T1_TARGET',  
'',  
'MOVE')
```

以下はこの照会の出力例です

Result set 1

-----

KEY	VALUE
-----	-------

```

AUTHID                                SVALENTI
CLEANUP_END                           2009-02-13-11.37.49.283090
CLEANUP_START                         2009-02-13-11.37.49.125786
COPY_END                              2009-02-13-11.37.47.806060
COPY_OPTS                             BY_KEY,OVER_INDEX
COPY_START                            2009-02-13-11.37.47.446616
COPY_TOTAL_ROWS                       0
INDEXNAME                             T1_INDEX
INDEXSCHEMA                           SVALENTI
INDEX_CREATION_TOTAL_TIME             1
INIT_END                              2009-02-13-11.37.47.287703
INIT_START                            2009-02-13-11.37.46.052952
PAR_COLDEF                            using a supplied target table so COLDEF
                                        could be different
REPLAY_END                            2009-02-13-11.37.48.785503
REPLAY_START                          2009-02-13-11.37.47.822109
REPLAY_TOTAL_ROWS                     0
REPLAY_TOTAL_TIME                     0
STATUS                                COMPLETE
SWAP_END                              2009-02-13-11.37.48.977745
SWAP_RETRIES                          0
SWAP_START                            2009-02-13-11.37.48.825228
VERSION                               09.07.0000
22 record(s) selected.

```

Return Status = 0

## 使用上の注意

### このプロシージャを使用する場合に最良の結果を得るためのヒント

- 同一表スペースに複数の移動を同時に行うことは避けてください。ターゲット表スペースでのフラグメント化の妨げになります。
- 表に対するアクティビティーが低いときに、このプロシージャを実行してください。並列読み取りアクセスが問題にならないように、データの一括ロードや一括削除は避けてください。
- 複数ステップの移動操作を使用してください。INIT フェーズと COPY フェーズはいつでも呼び出すことができます。ステージング表のサイズを小さくしておくためには、REPLAY フェーズを複数回実行します。その後、表に対するアクティビティーが低いときに SWAP を発行してください。
- ユニーク索引のない表を検討する場合や索引のない表の場合は特に、表移動の方式としてオフライン方式を選択した方が良くないか確認してください。

### ソース表に対する制限された操作

ストアード・プロシージャは、ソース表の何らかの変更点をキャプチャーするためにトリガーに依存します。ソース表に影響を与える可能性がある操作の中には、トリガーを起動しないものもあります。その結果、ストアード・プロシージャによって簡単に検出できないような不整合がソース表とターゲット表の間に発生する可能性があります。次のような操作がこれに該当します。

- TRUNCATE TABLE (トリガー削除時の制限なし)
- IMPORT ... REPLACE INTO ...
- LOAD TABLE
- ALTER TABLE

- REORG (オンラインおよびオフライン)

ソース表に対するこれらの操作は、新しい表レベルの状態フラグを使って制限されます。このフラグは INIT フェーズ中に設定され、CLEANUP または CANCEL フェーズ中に消去されます。制限された操作は失敗して、SQL0668N 理由コード 10 (sqlstate 57016) が出されます。

#### 表の移動操作に影響を与える操作

移動操作中にストアード・プロシージャの失敗の原因となり得る操作があります。次のような操作がこれに該当します。

- **SYSTOOLSPACE** 表スペースのドロップ
- ソース表のドロップ/名前変更
- INIT フェーズで OTM によって作成されたいずれかの一時オブジェクトのドロップ/名前変更 (ターゲット表、ステージング表、ソース表のトリガー、プロトコル表)
- ユーザー構成可能としてリストされていない、プロトコル表内の値の変更

#### 一時オブジェクトの命名規則

一時オブジェクトを作成する際の名前の競合を防ぐために、以下のような命名規則が使用されます。

- 接尾部
  - "t" (ターゲットを示す)
  - "s" (ステージングを示す)
  - "o" (「元の」を示す)
  - "g" (「生成された」を示す)
  - "i" (挿入トリガーを示す)
  - "d" (削除トリガーを示す)
  - "u" (更新前トリガーを示す)
  - "v" (更新後トリガーを示す)
- 名前は、<オブジェクト名から取られた文字><オブジェクト名に対する Base64 エンコード・ハッシュ・キー><接尾部> という構成で作成されます。
- 名前の長さがオブジェクトの長さ (128 バイト) を超える場合には、<オブジェクト名から取られた文字> が短くなります。
- ハッシュ値はオブジェクト名から計算されて、Base64 エンコード方式と同様にエンコードされます。

サンプル:

```
Name of object: T1
Staging object: T1AAAVxs
Target object: T1AAAVxt
Original object: T1AAAVxo
Generated index: T1AAAVxg (if table has no index)
Insert trigger: T1AAAVxi
Delete trigger: T1AAAVxd
Before update trigger: T1AAAVxu
After update trigger: T1AAAVxv
```

#### 圧縮およびディクショナリー作成を伴うオンライン表移動



オンライン表移動を使ってデータ・コンプレッション・ディクショナリーを作成する方法は、いくつかあります。ソース表で圧縮が使用可能に設定されるか、または新しい表定義 (提供される場合) で圧縮がアクティブと指定される必要があります。

サンプリングを使用したディクショナリーの作成は、オンライン表移動を介したディクショナリー作成のデフォルト方式です。表の圧縮がオンに設定されている場合、**COPY** 操作の実行前に、ソース表のデータのベルヌーイ・サンプリングがターゲット表に挿入されます。サンプリングされるデータの量はプロトコル表の `DEEPCOMPRESSSION_SAMPLE` フィールドで指定されます。その後、このランダム・サンプルに基づいてコンプレッション・ディクショナリーが作成され、結果として最適なコンプレッション・ディクショナリーになります。

サンプリング方式では XML コンプレッション・ディクショナリーが作成されないことに注意してください。これはコンプレッション・ディクショナリーの作成に `db2Inspect` が使用されるためであり、現在、`db2Inspect` には XML コンプレッション・ディクショナリーを作成する機能がありません。XML コンプレッション・ディクショナリーは自動ディクショナリー作成 (ADC) を介して作成されます。

自動ディクショナリー作成 (ADC) によるディクショナリー作成は、DB2 の表でディクショナリーを作成する標準的な方法です。表の圧縮をオンに設定するだけで、データが表に挿入されるときに DB2 はディクショナリーを自動作成します。その結果として、非最適コンプレッション・ディクショナリーが生成されます。ストアード・プロシージャがより最適なコンプレッション・ディクショナリーの作成を試みるのを防止するには、プロトコル表の `DEEPCOMPRESSSION_SAMPLE` フィールドを 0 に設定する必要があることに注意してください。

ディクショナリー作成の `REORG` メソッドを使ってディクショナリーを作成すると、`COPY` フェーズ中にソース表で発生したすべてのアクティビティを反映するディクショナリーが結果として作成されます。これは、`RESETDICTIONARY` オプションを設定して `SWAP` フェーズの前に `REORG` を実行することによって行われます。最適なディクショナリーが作成されますが、表のサイズによっては `REORG` に長い時間がかかる可能性があります。さらに、最適な XML ディクショナリーが必要な場合、それを生成する唯一の方法は `REORG` です。サンプリング方式を使ってディクショナリーを作成することが推奨されます。

## オンライン表移動と表の統計

統計が収集される表に対して表移動を実行する場合、デフォルト動作として、表に対する `RUNSTATS` が `SWAP` フェーズ中に実行されます。統計プロファイルが見つかった場合、その統計プロファイルを使用して `RUNSTATS` が呼び出されます。そうでない場合は、`WITH DISTRIBUTION ON COLUMNS (...) AND SAMPLE DETAILED INDEXES ALL` オプションを使って `RUNSTATS` が呼び出されます。

`COPY_STATS` オプションが設定されている場合、スワップの実行前にソース表から統計がターゲット表にコピーされます。統計をコピーすると、特にページ・サイズが変更される場合には、物理統計が不正確になる可能性があります。ただし、新しい統計を計算するために `RUNSTATS` を呼び出す必要がな

いため、コンピューティング時間が節約されます。また、統計が同じであるため、オブティマイザーは同じアクセス・プランを選択することができます(プランの安定性)。コピーされる統計は、**SYSSTAT.TABLES**、**SYSSTAT.COLUMNS**、**SYSSTAT.COLDIST**、**SYSSTAT.INDEXES**、および**SYSSTAT.COLGROUPS** カタログ・ビューの中にあります。

**NO\_STATS** オプションが設定されている場合、ストアード・プロシージャは **RUNSTATS** や統計コピー操作をターゲット表に対して実行しません。**AUTO\_RUNSTATS** または **AUTO\_STMT\_STATS** を使用すると、DB2 は新しい統計を自動的に作成します。

### **COPY** での **LOAD** 使用を伴うオンライン表移動

**COPY\_USE\_LOAD** オプションを使用するときに、サブオプションを指定しない場合、または **NONRECOVERABLE** を選択する場合には、リカバリー可能性を確保するために、**SWAP** フェーズの前にターゲット表スペースのバックアップを実行する必要があります。次のようなステートメントを発行することで、バックアップを作成できます。

```
BACKUP DB dbname TABLESPACE targetDataTablesapce,  
        targetIndexTablesapce, targetLongTablesapce  
ONLINE TO <destination>
```

### **HADR** 環境でのリカバリー可能 **LOAD** によるオンライン表移動

**HADR** 環境におけるリカバリー可能 **LOAD** の宛先がスタンバイで見つからない場合、表スペースはアクセス不能になります。この状態は 1 次には通知されないで、スタンバイは最新であると想定されるかもしれませんが、スタンバイへのフェイルオーバーが発生した場合、表にはアクセスできません。

DB2 バージョン 9.7 フィックスパック 2 より前のリリースでは、**COPY\_USE\_LOAD** を使用する場合は **FORCE** オプションを指定しなければなりません。そうしない場合、**SWAP** フェーズは実行されずエラーを受け取ることになります。

### 生成済み列を伴うオンライン表移動

表移動ストアード・プロシージャは、ソース表の中の生成済み列を特別な方法で扱います。以下の段落で、さまざまな種類の生成済み列がどのように扱われるかを説明します。

**行変更タイム・スタンプ列**は、行が最後に変更された時間を表すタイム・スタンプを保持する列です。

ソース表で行変更タイム・スタンプ列が見つかった場合、表移動操作が完了した後のこの列の値は、表移動操作の前と同じではありません。表移動後のこの列の値は、新しい表オブジェクトで行が挿入/更新された時間を表すようになります。これは、行が実際に変更されようとしており、行変更タイム・スタンプ列の値はこれらの変更を反映する必要があるためです。

新しい表定義が提供される場合、ソース表で列が行変更タイム・スタンプ列と定義されていても、新しい表定義でそのように定義されていなければ、その列は行変更タイム・スタンプ列になりません。

**ID** 列は、表に行が挿入されるときに列に関する値を自動生成する列です。

ソース表で ID 列が見つかった場合、表移動操作が完了した後のこの列の値は、表移動操作の前に存在した値と同じになります。ただし、ソース表での ID 列の「前の/次の」値を判別する方法はありません。このため、ターゲット表で ID 列を作成する際、次の「未変更」値から値の生成が始まるように設定されます。これはデータベース再始動 (停止/開始) のときに発生する動作と同じです。この動作は、インフォメーション・センターの『ALTER TABLE』という項目の『identity-alteration』セクションの『SET NO CACHE または CACHE integer-constant』という見出しで説明されています。

列はターゲット表で最初に通常の列として作成された後、SWAP フェーズの短いオフライン期間中に ID 列に変更されます。こうする理由は、列が GENERATED ALWAYS として作成された可能性があるためです。そのような場合、ストアード・プロシージャはソース表の実際の値をターゲット表の列に正確に挿入できません。

新しい表定義が指定される場合、列が新しい表定義で ID 列として指定されていれば、ストアード・プロシージャは ID 列の定義がソース表の列の定義と一致するかどうか検査します。一致する場合には、ストアード・プロシージャは上記の説明のような処理を続けます。一致しない場合、ストアード・プロシージャは新しい ID 列定義を使用します。この場合、ID 列カウンタは指定された開始値から再び開始されることに注意してください。ただし、列にある行の現在の値は引き続き同じです。

新しい表定義が指定され、ソース表で ID 列と指定されている列が新しい表定義では ID 列と指定されていない場合には、ストアード・プロシージャはその列を引き続き ID 列としてターゲット表に作成します (その際、ソース表で見つかった仕様と同じものが使われます)。これにより、ユーザーは既存の ID 列の定義を検索して新しい表定義にそれを再び入力する必要がなくなります。ユーザーがこの列を ID 列として保持したくない場合には、ストアード・プロシージャの呼び出し後にターゲット表を変更して、列の ID 仕様を除去することができます。

**式列**は、表に行が挿入されるときに、式に基づいて列の値を自動生成する列です。

ソース表で式列が見つかった場合、表移動操作が完了した後のこの列の値は、表移動操作の前に存在した値と同じになります。

列はターゲット表で最初に通常の列として作成された後、SWAP フェーズの短いオフライン期間中に式列に変更されます。こうする理由は、式列が GENERATED ALWAYS として作成されて、その列への挿入が禁止されるためです。ただし、ターゲット表の列を式列に変更するために、ターゲット表の保全性の設定が一時的にオフになります。ALTER ステートメントが実行された後、GENERATED COLUMN IMMEDIATE UNCHECKED オプションを使って保全性が元どおりオンに設定されます。

表名を含む列式 (例えば表 T1 での式 (T1.C \*5)) は、ソース表とターゲット表のどちらの場合も、ストアード・プロシージャによってサポートされません。この解決策として、ユーザーは列を修正して、表名を含まないように式を変更することができます。

新しい表定義が指定される場合、列が新しい表定義で式列として指定されていれば、ストアード・プロシージャは基本的なストリング比較を実行する

ことにより、式列の定義がソース表の列の定義と一致するかどうか検査します。一致する場合には、ストアード・プロシージャは上記の説明のような処理を続けます。一致しない場合、ストアード・プロシージャは新しい式列の定義を使用します。列にある行の現在の値は引き続き同じであることに注意してください。

新しい表定義が指定され、ソース表で式列と指定されている列が新しい表定義では式列と指定されていない場合には、ストアード・プロシージャはその列を引き続き式列としてターゲット表に作成します (その際、ソース表で見つかった仕様と同じものが使われます)。これにより、ユーザーは既存の式列の定義を検索して新しい表定義にそれを再び入力する必要がなくなります。ユーザーがこの列を式列として保持したくない場合には、ストアード・プロシージャの呼び出し後にターゲット表を変更して、列の式仕様を除去することができます。

### オンライン表移動と、保持されるオブジェクトおよび特権

表移動が実行されるとき、ストアード・プロシージャは以下のオブジェクトを保持します。

**ビュー** SWAP フェーズでの短いオフライン期間中に、ビューはソース表からドロップされてターゲット表に再作成されます。

また、ビューの所有権を元の所有者に戻すために、所有権の転送も実行されます。

ビューに関して付与された特権は、**auto\_reval** 構成パラメーターが DEFERRED に設定されている場合にのみ保持されます。

#### トリガー

SWAP フェーズでの短いオフライン期間中に、トリガーはソース表からドロップされてターゲット表に再作成されます。

また、トリガーの所有権を元の所有者に戻すために、所有権の転送も実行されます。

**索引** 表移動の操作中には、ターゲット表に索引が何度か作成されます。まず、索引は COPY フェーズの終わりに作成されます。ただし **COPY\_WITH\_INDEXES** オプションが設定されている場合には、COPY フェーズの開始時に索引が最初に作成されます。さらに、その後の REPLAY および SWAP フェーズの開始時に、ストアード・プロシージャは新しく作成される索引があるかどうかを検索します。その際、索引名だけから判断します。新しい索引が見つかった場合、それが作成されます。ただし、ストアード・プロシージャはソース表からいずれかの索引が削除されたかどうかを確認しません。

ユーザー作成の索引の場合、索引名はソース表と同じになります。ただし、システム作成の索引が同じ名前になるかどうかは保証されません。

保持される索引のタイプは、'REG'、'CLUST'、および 'XVIL' です。

ターゲット表でドロップされる列を参照するユーザー作成の索引は、保持されません。

ソース・パーティション表からターゲット・パーティション表に移動する場合、索引のパーティション属性は保持されます。パーティション化されたソース表から非パーティション化されたターゲット表に移動する場合、またはその逆の場合には、データベースのデフォルト動作によってパーティション属性が決まります。

**制約** (参照制約を除く) 制約は、同じ制約名を使ってターゲット表で再作成されます。ただし、ユニーク制約と主制約の場合、基礎となる索引名はソース表の索引名と異なる可能性があります。

#### 表フラグ

ソース表の表フラグは、INIT フェーズでターゲット表が作成されると直ちにターゲット表に作成されます。該当するフラグは、'append\_mode'、'locksize'、'volatile'、'compression'、'datacapture'、'pctfree'、'logindexbuild'、'owner'、および 'droprule' です。その後、これらのフラグは COPY フェーズの終わりと SWAP フェーズ中に検査されます。フラグの内容が変更されている場合は、ターゲット表でそれが更新されます。

データベースをリカバリー可能にして、しかも HADR セットアップとの互換性を保つために、ADMIN\_MOVE\_TABLE はソースの NOT LOGGED INITIALLY 情報をターゲット表にコピーしません。

#### 付与/取り消し

SWAP フェーズ中に、ストアード・プロシージャは **SYSCAT.TABAUTH** 内の項目を調べて、表に関するユーザー/グループ/ロールへの特権付与を複製します。

ストアード・プロシージャの呼び出し元が ACCESSCTRL 権限または SECADM 権限をどちらも持っていない場合、CONTROL 特権を付与することはできません。CONTROL 特権を付与されなかったすべてのユーザー/グループ/ロールのリストは、プロトコル表の WARNINGS キーの部分にあります。

#### 使用量リスト

SWAP フェーズでの短いオフライン期間中に、ソース表またはソース表の索引に対して定義された使用量リストがドロップされ、ターゲット表に対して再作成されます。移動前にアクティブ状態だった使用量リストは、移動後にすべて再びアクティブ化されます。

データベースの auto\_revalidation が有効で、しかも **USE\_AUTO\_REVAL** オプション (auto\_revalidation が有効である場合のデフォルト) が設定されている場合、ビューは上記のようにドロップされないことに注意してください。代わりに、ビューは保持されて、auto\_revalidation によって再妥当性検査されます。現在、サブジェクトとして定義されたトリガーを持つ表の名前変更に関して制限があるため、トリガーがストアード・プロシージャによってドロップされて、再作成されます。

#### 索引によるクラスタリングを伴うオンライン表移動

ターゲット表を索引によってクラスタ化することが可能です。ソース表にクラスタ索引が存在する場合、デフォルトでは、その索引によってクラスタ化されます。INIT フェーズ後にこのデフォルトを変更することができ

ます (これはフェーズ全体にわたるオンライン表移動の実行を意味します)。クラスター索引が存在しない 1 つの MOVE フェーズでオンライン表移動を呼び出した場合、ストアード・プロシージャはユニーク/主索引を使ってターゲット表をクラスター化します。クラスター索引が存在する場合、ストアード・プロシージャはクラスター索引を使ってターゲット表をクラスター化します。

ソース表にクラスター索引が存在する場合、ターゲット表をクラスター索引でクラスター化しないことも可能です。そうするには、複数ステップから成る移動を実行して INIT フェーズ後にキー項目 COPY\_INDEXSCHEMA および COPY\_INDEXNAME をプロトコル表から削除します。

任意の副次索引によってターゲット表をクラスター化できます。そうするには、複数ステップから成る移動を実行し、適切な索引を使ってプロトコル表のキー項目 COPY\_INDEXSCHEMA および COPY\_INDEXNAME を挿入/更新することで、ターゲット表をクラスター化します。

### 索引属性の変更

既存のいずれかの属性を変更する必要がある場合 (例えば索引のクラスタリング、索引の圧縮、グローバル索引からローカル索引への変更、またはその逆への変更)、ユーザーは複数ステップから成る移動操作中に手動で属性を変更できます。

これを行うには、複数ステップから成る移動で INIT および COPY 移動フェーズを実行します。その後、ターゲット表の索引を手動で変更します。ターゲット表の名前はプロトコル表の中にあります。変更が完了した後、REPLAY および SWAP フェーズから再開します。

### 制約事項

ADMIN\_MOVE\_TABLE ストアード・プロシージャには、以下の制限が適用されます。

- ソース表としてサポートされるのは単純な表のみです。マテリアライズ照会表、型付き表、範囲がクラスター化された表、システム表、ビュー、ニックネーム、別名は許可されません。
- イベント・モニターが現在アクティブになっている表を移動することはできません。
- ユニーク索引のない表は、複雑で高コストになる可能性がある再生フェーズの対象となります。
- 表に LOB、XML、または LONG 列がある場合は、ユニーク索引が必要になります。
- 生成された列を MDC 仕様の一部にすることはできません。
- テキスト検索索引に対するサポートはありません。
- ディスク・スペース所要量が大きいことに気を付けてください。このプロシージャは表と索引の 2 種類のコピーを作成するほか、ステージング表とログ・スペースも作成します。
- データのほとんどが「選択から挿入」の形式を使用して新しい表に移動されるので、コピー・パフォーマンスが問題になる可能性があります。

- ユニーク索引のない表に対する VERIFY 操作は、LOB のある表では機能しません。
- DB2 バージョン 9.7 フィックスパック 2 より前のリリースでは、**DB2\_SKIPDELETED** レジストリー変数を ON に設定できません。
- SYSTOOLSPACE 表スペースが作成されていて、「PUBLIC」でアクセスできなければなりません。
- ソース表に対する長時間の実行トランザクションのために、COPY フェーズでロック・タイムアウトになる可能性があります。
- SWAP フェーズでデッドロックが発生する可能性があります。
- ソース表に非ユニーク索引があり、更新処理がいくつも行われると、デッドロックが発生する可能性があります。
- VARCHAR2 サポートが有効になっている場合、データベースは空ストリングと NULL を等価の値として扱いますが、単一ブランクは別個の値です。  
VARCHAR2 サポートが有効になっている場合、*mdc\_cols*、*partkey\_cols*、および *data\_part* パラメーターは、単一ブランクを空ストリングおよび NULL とは異なるものとして使用することになります。
- SET INTEGRITY PENDING 状態の表は、移動できません。
- XSR オブジェクトが従属している表は、移動できません。

## 戻される情報

表 71. ADMIN\_MOVE\_TABLE ストアド・プロシージャによって戻される情報

列名	データ・タイプ	説明
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント
TABNAME	VARCHAR(128)	table_name - 表名モニター・エレメント
KEY	VARCHAR(32)	属性の名前。
VALUE	CLOB(10M)	属性の値。

結果セットで戻されるキーと値のペアを、表 72 に示しています。結果セット内のユーザー構成可能キーを変更するには、ADMIN\_MOVE\_TABLE\_UTIL ストアド・プロシージャを使用します。

表 72. ADMIN\_MOVE\_TABLE ストアド・プロシージャによって戻されるキーと値のペア

キー	戻り値	ユーザー構成可能
AUTHID	ストアド・プロシージャを呼び出したユーザーの許可 ID を表示します。	いいえ
CLEANUP_END	CLEANUP フェーズの終了時刻を表示します。	いいえ
CLEANUP_START	CLEANUP フェーズの開始時刻を表示します。	いいえ
COMMIT_AFTER_N_ROWS	この数の行がコピーされると COPY フェーズでコミットが実行されます。0 は COPY 中にコミットが行われないことを意味します。デフォルト値は 10000 です。	はい

表 72. ADMIN\_MOVE\_TABLE ストアード・プロシージャーによって戻されるキーと値のペア (続き)

キー	戻り値	ユーザー構成可能
COPY_ARRAY_SIZE	COPY_ARRAY_INSERT の配列サイズを指定します。0 以下の値は、COPY_ARRAY_INSERT を使用しないことを意味します。デフォルト値は 100 です。	はい
COPY_END	COPY フェーズの終了時刻を表示します。	いいえ
COPY_INDEXNAME	COPY フェーズでターゲット表のデータのクラスタ化に使用される索引の名前。この値は、COPY フェーズの前に設定する必要があります。ソース表のクラスタ索引が存在する場合は、その名前がデフォルトの名前になります。存在しない場合は、ソース表のユニーク索引または 1 次索引の名前がデフォルトの名前になります。	はい
COPY_INDEXSCHEMA	COPY フェーズでターゲット表のデータのクラスタ化に使用される索引のスキーマ。この値は、COPY フェーズの前に設定する必要があります。ソース表のクラスタ索引が存在する場合は、そのスキーマ名がデフォルト・スキーマになります。存在しない場合は、ソース表のユニーク索引または 1 次索引のスキーマ名がデフォルト・スキーマになります。	はい
COPY_OPTS	COPY フェーズで使用されるコピー・オプション。	いいえ
COPY_START	COPY フェーズの開始時刻を表示します。	いいえ
COPY_TOTAL_ROWS	COPY フェーズでコピーされた行の総数を表示します。	いいえ
DEEPCOMPRESSSION_SAMPLE	このフィールドは、ソース表の圧縮が有効になっている場合に、圧縮用ディクショナリーの作成時にサンプリングするデータ量 (KB 単位) を指定します。0 はサンプリングが行われないことを意味します。デフォルト値は 20MB (20480 KB) です。	はい
INDEX_CREATION_TOTAL_TIME	副次索引の作成に要した合計時間を表示します。	いいえ
INDEXNAME	索引の名前を表示します。表に索引がない場合は空ストリングになります。	いいえ
INDEXSCHEMA	索引のスキーマを表示します。表に索引がない場合は空ストリングになります。	いいえ
INIT_END	INIT フェーズの終了時刻を表示します。	いいえ
INIT_START	INIT フェーズの開始時刻を表示します。	いいえ
LOCK	別のオンライン表移動ストアード・プロシージャー呼び出しがアクティブの場合に、ロック開始時刻を表示します。それ以外の場合は空です。	いいえ
ORIGINAL	スワップ後の元の表の名前を表示します。	いいえ
REORG_USE_TEMPSPACE	REORG オプションを呼び出す場合は、REORG コマンドの USE 節に TEMPORARY 表スペースを指定することもできます。この値が指定されない場合、REORG コマンドは再編成される表と同じ表スペースを使用します。	はい
REPLAY_END	REPLAY フェーズの終了時刻を表示します。	いいえ



表 72. ADMIN\_MOVE\_TABLE ストアード・プロシージャによって戻されるキーと値のペア (続き)

キー	戻り値	ユーザー構成可能
REPLAY_MAX_ERR_RETRIES	REPLAY フェーズでエラー (ロック・タイムアウトまたはデッドロック) が発生した場合の最大再試行回数を指定します。デフォルト値は 100 です。	はい
REPLAY_START	REPLAY フェーズの開始時刻を表示します。	いいえ
REPLAY_THRESHOLD	REPLAY フェーズの単一反復について、ステージング表に適用される行数がこの値より小さければ、REPLAY は停止します。この動作は、その間に新しい項目が作成されたとしても変わりません。デフォルト値は 100 です。	はい
REPLAY_TOTAL_ROWS	再生された行の累積数を表示します。	いいえ
REPLAY_TOTAL_TIME	行の再生に使用された累積時間 (秒単位) を表示します。	いいえ
STAGING	ステージング表の名前を表示します。	いいえ
STATUS	オンライン表移動の現在の状況を表示します。 <ul style="list-style-type: none"> <li>• INIT: INIT が進行中。</li> <li>• COPY: COPY が進行中であるか COPY が見込まれる。</li> <li>• REPLAY: REPLAY が進行中であるか REPLAY と SWAP が見込まれる。</li> <li>• CLEANUP: MOVE が完了しているが、CLEANUP が完了していないか CLEANUP が見込まれる。</li> <li>• COMPLETE: MOVE と CLEANUP が完了している。</li> <li>• COMPLETE_WITH_WARNINGS: MOVE と CLEANUP が完了しているが、警告がある (WARNINGS キーのフィールドにリストされる)。</li> </ul>	いいえ
SWAP_END	SWAP フェーズの終了時刻を表示します。	いいえ
SWAP_MAX_RETRIES	SWAP フェーズにおける (ロック・タイムアウトまたはデッドロックが発生した場合の) 再試行の許容最大回数を指定します。デフォルト値は 10 です。	はい
SWAP_RETRIES	SWAP フェーズで行われた再試行の回数を表示します。	いいえ
SWAP_START	SWAP フェーズの開始時刻を表示します。	いいえ
TARGET	ターゲット表の名前を表示します。	いいえ
UTILITY_INVOCATION_ID	表移動操作の固有 ID を表示します。	いいえ
VERIFY_END	検査の終了時刻を表示します。	いいえ
VERIFY_START	検査の開始時刻を表示します。	いいえ
VERSION	ストアード・プロシージャのバージョンを表示します。	いいえ

表 72. ADMIN\_MOVE\_TABLE ストアド・プロシージャーによって戻されるキーと値のペア (続き)

キー	戻り値	ユーザー構成可能
WARNINGS	<p>ユーザーに伝える警告を表示します。この警告には、以下の種類があります。</p> <ul style="list-style-type: none"> <li>失敗したすべてのオブジェクトの再確認。</li> <li>ユーザー、グループ、またはロールに制御権を付与できなかった。</li> <li>索引が参照する列がもう存在しないために、索引が作成されなかった。</li> </ul>	いいえ

## ADMIN\_MOVE\_TABLE\_UTIL プロシージャー - オンライン表移動の変更プロシージャー

ADMIN\_MOVE\_TABLE\_UTIL プロシージャーは、アクティブ表データの移動時に、SYSPROC.ADMIN\_MOVE\_TABLE ストアド・プロシージャーと連動します。このストアド・プロシージャーは、ADMIN\_MOVE\_TABLE プロシージャーによって作成および使用される ADMIN\_MOVE\_TABLE プロトコル表のユーザー定義可能値を変更するための手段として使用できます。

このプロシージャーが ADMIN\_MOVE\_TABLE プロトコル表の値を変更するのは、TABSCHEMA および TABNAME パラメーターが参照する表の表移動が既に進行中であり、プロシージャーの呼び出し元の許可 ID が表移動を実行しているユーザーと同じである場合のみです。

### 構文

```
▶▶ADMIN_MOVE_TABLE_UTIL(—tabschema—,—tablename—,—action—,—key—,—value—)▶▶
```

このストアド・プロシージャーのスキーマは SYSPROC です。

### プロシージャー・パラメーター

#### *tabschema*

この入力パラメーターは、移動中の表が含まれるスキーマの名前を指定します。この名前には、大/小文字の区別があります。データ・タイプは VARCHAR(128) です。

#### *tablename*

この入力パラメーターは、移動中の表の名前を指定します。このパラメーターには大/小文字の区別があり、データ・タイプは VARCHAR(128) です。

#### *action*

この入力パラメーターは、プロシージャーの実行アクションを指定します。

有効な値は以下のとおりです。

- UPSERT: 指定した TABSCHEMA.TABNAME.KEY が ADMIN\_MOVE\_TABLE プロトコル表に存在する場合は、対応する値が新しい *value* パラメーターで更新されます。その他の場合は、ADMIN\_MOVE\_TABLE プロトコル表にキーと値のペアが挿入されます。

- DELETE: 指定した TABSCHEMA.TABNAME.KEY が ADMIN\_MOVE\_TABLE プロトコル表に存在する場合は、指定したキーと値のペアが ADMIN\_MOVE\_TABLE プロトコル表から削除されます。

このパラメーターのデータ・タイプは VARCHAR(128) です。

#### key

この入力パラメーターは、ADMIN\_MOVE\_TABLE プロトコル表で「UPSERT」または削除を行うためのキーを指定します。

有効な値は以下のとおりです。

- COMMIT\_AFTER\_N\_ROWS: この数の行がコピーされると COPY フェーズでコミットが実行されます。値 0 は、COPY 中にコミットを実行しないことを意味します。
- DEEPCOMPRESSION\_SAMPLE: このフィールドは、ソース表の圧縮が有効になっている場合に、圧縮用ディクショナリーの作成時にサンプリングするデータ量 (KB 単位) を指定します。値 0 は、サンプリングを行わないことを意味します。
- COPY\_ARRAY\_SIZE: COPY\_ARRAY\_INSERT の配列サイズを指定します。0 以下の値は、COPY\_ARRAY\_INSERT を使用しないことを意味します。
- COPY\_INDEXSCHEMA: COPY フェーズでターゲット表のデータのクラスター化に使用する索引のスキーマ。
- COPY\_INDEXNAME: COPY フェーズでターゲット表のデータのクラスター化に使用する索引の名前。
- REPLAY\_MAX\_ERR\_RETRIES: REPLAY フェーズでエラー (ロック・タイムアウトまたはデッドロック) が発生した場合の最大再試行回数を指定します。
- REPLAY\_THRESHOLD: REPLAY フェーズの単一反復について、ステージング表に適用される行数がこの値より小さければ、REPLAY は停止します。この動作は、その間に新しい項目が作成されたとしても変わりません。
- REORG\_USE\_TEMPSPACE: 表移動で REORG オプションを呼び出す場合は、REORG コマンドの USE 節に TEMPORARY 表スペースを指定することもできます。この値が指定されない場合、REORG コマンドは再編成される表と同じ表スペースを使用します。
- SWAP\_MAX\_RETRIES: SWAP フェーズにおける (ロック・タイムアウトまたはデッドロックが発生した場合の) 再試行の許容最大回数を指定します。

このパラメーターのデータ・タイプは VARCHAR(128) です。

#### value

この入力パラメーターは、ADMIN\_MOVE\_TABLE プロトコル表に「UPSERT」する値を指定します。このパラメーターのデータ・タイプは CLOB(10M) です。このパラメーターを NULL または空ストリングにすることができます。

### 許可

このルーチンを使用するには、以下のいずれかの許可が必要です。

- ADMIN\_MOVE\_TABLE\_UTIL プロシージャに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限

- SQLADM 権限

さらに、使用する許可 ID は、ADMIN\_MOVE\_TABLE ストアド・プロシージャーの呼び出しに使用されたものと同じでなければなりません。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャーが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

この例では、このストアド・プロシージャーの基本的な呼び出しを取り上げています。ここでは、圧縮の値を更新し、ターゲット表のコピー処理に使用される特定の索引情報を除去するために呼び出します。

まず、ADMIN\_MOVE\_TABLE プロシージャーを呼び出して表移動プロセスを開始します。その後で、ADMIN\_MOVE\_TABLE プロトコル表の値を更新または削除するために、ADMIN\_MOVE\_TABLE\_UTIL プロシージャーを呼び出します。

```
CALL SYSPROC.ADMIN_MOVE_TABLE('SVALENTI','T1','','','','','','','','','INIT')
```

次に、DEEP\_COMPRESSION\_SAMPLE 値を 30720 KB に更新します。

```
CALL SYSPROC.ADMIN_MOVE_TABLE_UTIL('SVALENTI','T1','UPSERT',  
    'DEEPCOMPRESSON_SAMPLE','30720')
```

ここで、COPY\_INDEXSCHEMA と COPY\_INDEXNAME の値を削除します。

```
CALL SYSPROC.ADMIN_MOVE_TABLE_UTIL('SVALENTI','T1','DELETE','COPY_INDEXSCHEMA','')  
CALL SYSPROC.ADMIN_MOVE_TABLE_UTIL('SVALENTI','T1','DELETE','COPY_INDEXNAME','')
```

これらの変更後、メタ・テーブル内の新しい値を使用して ADMIN\_MOVE\_TABLE プロシージャーを続行します。

```
CALL SYSPROC.ADMIN_MOVE_TABLE('SVALENTI','T1','','','','','','','','','COPY')  
CALL SYSPROC.ADMIN_MOVE_TABLE('SVALENTI','T1','','','','','','','','','REPLAY')  
CALL SYSPROC.ADMIN_MOVE_TABLE('SVALENTI','T1','','','','','','','','','SWAP')
```

## 使用上の注意

ADMIN\_MOVE\_TABLE プロトコル表の変更可能キー値についての詳細が、ADMIN\_MOVE\_TABLE プロシージャーの『使用上の注意』のセクションに記載してあります。

## ADMIN\_REMOVE\_MSGS プロシージャー - ADMIN\_CMD プロシージャーを通して実行するデータ移動ユーティリティによって生成されたメッセージのクリーンアップ

ADMIN\_REMOVE\_MSGS プロシージャーは、ADMIN\_CMD プロシージャーによるデータ移動ユーティリティ・コマンドの単一の実行によって生成されたメッセージをクリーンアップするために使用します。

入力パラメーター *operation\_id* によって操作が識別されます。

## 構文

▶▶—ADMIN\_REMOVE\_MSGS—(—*operation\_id*—)————▶▶▶▶

スキーマは SYSPROC です。

### プロシージャ・パラメーター

*operation\_id*

ADMIN\_CMD プロシージャを通して実行したデータ移動ユーティリティによって作成された、メッセージ・ファイルの操作 ID を指定する、タイプ VARCHAR(139) の入力引数。操作 ID は、ADMIN\_CMD プロシージャによって生成されます。

### 許可

ADMIN\_REMOVE\_MSGS プロシージャに対する EXECUTE 特権。fenced ユーザー ID は、レジストリー変数 DB2\_UTIL\_MSGPATH によって示されるディレクトリーの下にあるファイルを削除できなければなりません。レジストリー変数が設定されていない場合、fenced ユーザー ID には、instance ディレクトリーの tmp サブディレクトリーにあるファイルを削除できなければなりません。

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

操作 ID '24523\_THERESAX' のメッセージをクリーンアップします。

```
CALL SYSPROC.ADMIN_REMOVE_MSGS('24523_THERESAX')
```

### 使用上の注意

適切な *operation\_id* を指定してこのプロシージャを呼び出す CALL ステートメントは、ADMIN\_CMD プロシージャが戻す最初の結果セットの MSG\_REMOVAL 列に示されます。

### ADMIN\_REVALIDATE\_DB\_OBJECTS プロシージャ - 無効なデータベース・オブジェクトを再検証する

ADMIN\_REVALIDATE\_DB\_OBJECTS プロシージャは、データベース・オブジェクトを再検証します。

このプロシージャは、実行する再検証のレベルを制御するさまざまな入力パラメーターを使用します。

- データベースの無効なオブジェクトをすべて再検証するには、すべてのパラメーターに NULL を指定するか、パラメーターを指定せずにプロシージャを呼び出します。

- 特定のスキーマの下の無効なデータベース・オブジェクトをすべて再検証するには、*object\_schema* に値を指定し、*object\_name* および *object\_type* に NULL を指定します。
- 特定の無効なデータベース・オブジェクトを再検証するには、最初のパラメーターに有効な値を指定します。
- 現在の状態が有効か無効かにかかわらず、コンパイル済み SQL PL または PL/SQL オブジェクトの再検証を強制的に実行するには、*object\_type*、*object\_schema*、*object\_name* に値を指定し、*force* に 'Y' を指定します。
- 再検証されているコンパイル済み SQL PL または PL/SQL オブジェクトのデバッグを使用可能または使用不可にするには、*debug\_flag* に 'Y' または 'N' を指定します。

## 構文

```
▶▶—ADMIN_REVALIDATE_DB_OBJECTS—(—object_type—,—object_schema—,——————▶
▶—object_name—,—force—,—debug_flag—)——————▶▶
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *object\_type*

データベース・オブジェクトのタイプを識別する、タイプ VARCHAR (30) の入力引数。以下のタイプが有効です。

- FUNCTION
- GLOBAL VARIABLE
- MASK
- METHOD
- MODULE
- PERMISSION
- PROCEDURE
- SPECIFIC
- TABLE
- TRIGGER
- TYPE
- USAGELIST
- VIEW

この値は大/小文字が区別されません。この値は NULL 値にすることができます。そうした場合、すべてのオブジェクト・タイプが考慮されます。

これらのタイプのいずれかが指定された場合は、プロシージャは、MODULE に属するオブジェクトを除いて、そのタイプの無効なオブジェクトをすべて再検証します。モジュール内のオブジェクトを再検証する場合は、特定のモジュールの名前を指定して MODULE タイプを使用すると、そのモジュール内のすべての無効なオブジェクトが再検証されます。

複数のパラメーター・シグニチャーを持つルーチンがあり、それらのうちの 1 つのみを再検証する場合は、再検証するルーチンの名前を指定して **SPECIFIC** タイプを使用します。

**TABLE** タイプを使用すると、指定された表が再編成され、その統計が収集されます。プロシージャーは、**REORG** ペンディング状態にある正規またはマテリアライズ照会表に対して、**REORG** ユーティリティ、**RUNSTATS** ユーティリティの順に呼び出します。プロシージャーは、**RUNSTATS** のユーザー・プロファイルが存在する場合は、これを使用することを試みます。存在しない場合は、デフォルトの **RUNSTATS** 操作が呼び出されます。

#### *object\_schema*

データベース・オブジェクト・リファレンスの修飾に使用されるスキーマ名を識別する、タイプ **VARCHAR (128)** の入力引数。この名前は大文字小文字が区別されます。この値を **NULL** にすることができます。

#### *object\_name*

データベース・オブジェクトを識別する、タイプ **VARCHAR(128)** の入力引数。この名前は大文字小文字が区別されます。この値は、型付き表または行数の値とすることはできません。これは、プロシージャーがこれらのタイプのオブジェクトをサポートしていないからです。そのようなオブジェクトの名前が指定されると、エラーが戻されます。この値を **NULL** にすることができます。

#### *force*

以下のオブジェクトについて (その現在の状態にかかわらず) 再検証を強制的に実行するかどうかを示す、タイプ **CHAR(1)** の入力引数。

- 呼び出し元のユーザーが **DBADM** 権限を持つ場合は、すべての **SQL PL** および **PL/SQL** オブジェクト。
- 呼び出し元のユーザーが所有している **SQL PL** および **PL/SQL** オブジェクト。

上記の基準を満たしていない **SQL PL** および **PL/SQL** オブジェクトと、**SQL PL** オブジェクトでも **PL/SQL** オブジェクトでもないオブジェクトについては、この引数は無視されます。

値は以下のとおりです。

- 'Y' または 'y' - すべてのコンパイル済み **SQL PL** および **PL/SQL** オブジェクトが再検証されます。
- 'N'、'n'、または **NULL** - デフォルトの動作 (無効なオブジェクトだけが再検証される) が実行されます。

#### *debug\_flag*

このプロシージャーが処理する以下のオブジェクトについてデバッグ・モードを使用可能または使用不可を示す、タイプ **CHAR(1)** の入力引数。

- 呼び出し元のユーザーが **DBADM** 権限を持っているか、組み込み役割 **SYSDEBUG** のメンバーである場合は、すべての **SQL PL** および **PL/SQL** オブジェクト。
- 呼び出し元のユーザーが所有している **SQL PL** および **PL/SQL** オブジェクト。

上記の基準を満たしていない SQL PL および PL/SQL オブジェクトと、SQL PL オブジェクトでも PL/SQL オブジェクトでもないオブジェクトについては、この引数は無視されます。

値は以下のとおりです。

- Y または y - コンパイル済み SQL PL および PL/SQL オブジェクトが、デバッグ・モードを有効にして再検証されます。
- N または n - コンパイル済み SQL PL および PL/SQL オブジェクトが、デバッグ・モードを無効にして再検証されます。
- NULL - コンパイル済み SQL PL および PL/SQL オブジェクトが、デバッグ・モードを変更せずに再検証されます。これがデフォルトの動作です。

## 許可

以下のいずれかの権限が必要です。

- ADMIN\_REVALIDATE\_DB\_OBJECTS プロシージャに対する EXECUTE 特権
- object\_type が MASK または PERMISSION である場合、SECADM 権限

*force* オプションの値が 'Y' である場合は、呼び出し元のユーザーが、再検証するオブジェクトの所有者であるか、DBADM 権限を持っている必要があります。

*debug\_flag* オプションの値が 'Y' である場合は、呼び出し元のユーザーが、オブジェクトの所有者である、DBADM 権限を持っている、または SYSDEBUG システム定義役割のメンバーである必要があります。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: 現在のデータベースのすべてを再検証します。

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS(NULL, NULL, NULL)
```

または、パラメーターを指定せずにプロシージャを呼び出します。

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS()
```

例 2: スキーマ MY\_SCHEMA によって修飾されるオブジェクトをすべて再検証します。

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS(NULL, 'MY_SCHEMA', NULL)
```

例 3: データベース内のトリガー・オブジェクトをすべて再検証します。

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS('trigger', NULL, NULL)
```

例 4: 特定のビュー・オブジェクトを再検証します。

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS('view', 'MY_SCHEMA', 'MY_VIEW')
```

例 5: MY\_SCHEMA の下のプロシージャをすべて再検証します。この例では、このスキーマの下に 3 つのプロシージャ (proc1、proc2、および proc3) があります。proc1 によって使用された参照されたオブジェクトは存在しません。以下の呼



び出しによって、proc2 および proc3 を再検証できますが、proc1 は無効のままです。この状態の場合、呼び出しによって警告が戻されます。

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS('procedure', 'MY_SCHEMA', NULL)
```

例 6: 存在しないオブジェクトを再検証します。この例では、エラーが戻されません。

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS('procedure', 'MY_SCHEMA', 'MY_VIEW')
```

例 7: 名前付きパラメーター表記を使用して、MY\_SCHEMA の下のプロシージャをすべて再検証します。

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS(  
  object_type=>'PROCEDURE',object_schema=>'MY_SCHEMA')
```

## 使用上の注意

ADMIN\_REVALIDATE\_DB\_OBJECTS プロシージャに渡される NULL 以外のパラメーター値は、すべて指定される必要があります。指定されない場合は、プロシージャは再検証が必要なオブジェクトを識別できません。例えば、ビュー名をトリガー・タイプで指定すると、タイプが一致しないので、プロシージャはビューを再検証しません。

このプロシージャは、無効なオブジェクト、および REORG ペンディング状態の正規またはマテリアライズ照会表のみを再検証します。すべての無効なオブジェクトは SYSCAT.INVALIDOBJECTS で検出できます。どの表が REORG ペンディング状態であるかを判別するには、ADMIN\_GET\_TAB\_INFO 表関数を使用します。

有効なオブジェクトが入力として指定された場合は、プロシージャはあらゆる操作を実行せず、成功コードを戻します。表の再検証中に障害が発生すると、プロシージャも失敗します。他のオブジェクトの再検証中に障害が発生すると、プロシージャは、障害を無視し、他のオブジェクトの再検証を続行します。1 つでも障害が発生すると、プロシージャは、警告 (SQLSTATE 0168B) を戻します。すべてのオブジェクトの再検証が失敗すると、プロシージャはエラーを戻します (SQLSTATE 429C4)。表を除くオブジェクトのすべての再検証の失敗について詳しくは、SYSCAT.INVALIDOBJECTS を参照してください。

無効なマスクまたは権限を再検証するためには、ADMIN\_REVALIDATE\_DB\_OBJECTS を実行するユーザーに SECADM 権限が必要です。1 つ以上の障害が発生して、最初の障害の原因が、マスクまたは権限の再検証を実行するときにユーザーに SECADM 権限がないためである場合、プロシージャは、警告 (SQLSTATE 0168B、SQLCODE +361) を戻し、msg-token2 には、CREATE PERMISSION または CREATE MASK が含まれます。すべてのオブジェクトの再検証が失敗して、最初の障害の原因が、マスクまたは権限の再検証を実行するときにユーザーに SECADM 権限がないためである場合、プロシージャがエラー (SQLSTATE 42501、SQLCODE -551) を戻します。

グローバル変数が再検証されると、現行セッションに対してもインスタンス化されます。

表の再検証の進行をモニターするには、関連する表の REORG 操作の進行をモニターできます。他のすべてのオブジェクトについては、SYSCAT.INVALIDOBJECTS

カタログ・ビューを照会します。オブジェクトは、正常に再検証されるとこのビューから削除され、再検証が失敗すると、項目は更新されます。

## ADMIN\_SET\_INTRA\_PARALLEL プロシージャ - パーティション内並列処理を有効または無効にする

ADMIN\_SET\_INTRA\_PARALLEL プロシージャは、データベース・アプリケーションのパーティション内並列処理を有効または無効にします。このプロシージャは現行トランザクション内で呼び出されますが、次のトランザクションから適用されます。

### 構文

```
▶▶—ADMIN_SET_INTRA_PARALLEL—(—state—)—————▶▶
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### 状態

データベース・アプリケーションのパーティション内並列処理の要求状態を指定する、タイプ VARCHAR(3) の入力引数。引数の値は、以下のいずれかになります。

#### YES または yes

データベース・アプリケーションは、次のトランザクションからパーティション内並列処理が有効になった状態で実行が開始されます。

#### NO、no

データベース・アプリケーションは、次のトランザクションからパーティション内並列処理が無効になった状態で実行が開始されます。

**NULL** 次のトランザクションから、**intra\_parallel** データベース・マネージャー構成パラメーターの値で決まるパーティション内並列処理の状態で、データベース・アプリケーションの実行が開始されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

例 1: 次のトランザクションからパーティション内並列処理を有効にした状態でデータベース・アプリケーションを実行するには、以下のようにします。

```
CALL ADMIN_SET_INTRA_PARALLEL('YES')
```

例 2 次のトランザクションからパーティション内並列処理を無効にした状態でデータベース・アプリケーションを実行するには、以下のようになります。

```
CALL ADMIN_SET_INTRA_PARALLEL('NO')
```

例 3: **intra\_parallel** データベース・マネージャー構成パラメーターの値で決まるパーティション内並列処理の状態でデータベース・アプリケーションを実行するには、次のようになります。

```
CALL ADMIN_SET_INTRA_PARALLEL(NULL)
```

## 使用上の注意

ADMIN\_SET\_INTRA\_PARALLEL で設定される並列処理の要求状態は、このプロシージャを呼び出すアプリケーションに対してのみ適用されます。

ADMIN\_SET\_INTRA\_PARALLEL を呼び出しても、**intra\_parallel** データベース・マネージャー構成パラメーターの設定値は変わりません。

MAX DEGREE ワークロード属性の値を指定したワークロードにデータベース・アプリケーションが関連付けられている場合は、ADMIN\_SET\_INTRA\_PARALLEL によって適用されるパーティション内並列処理の設定がオーバーライドされることがあります。

共有ソート・ヒープを使用できない場合、パーティション内並列処理をこのプロシージャによって使用可能にすることはできません (SQL5192W)。

## ADMIN\_SET\_MAINT\_MODE プロシージャ - SQL コンパイル用の保守モードの設定

ADMIN\_SET\_MAINT\_MODE プロシージャは、SQL の実行時の SQL コンパイル中にトリガーや制約を展開しないように、現行接続の内部フラグを設定します。

### 構文

```
►►—ADMIN_SET_MAINT_MODE—(—settings—)—————►►
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *settings*

この入力パラメーターを使用すると、クライアントから管理保守モードを設定できます。この設定は、クライアント SQL がどのように実行されるかに影響を与えます。値として、設定のリストを含む VARCHAR を指定します。設定のリストは以下のとおりです。

#### **INHIBIT\_TRIG\_RI\_MQT**

トリガー、制約、またはマテリアライズ照会表 (MQT) が照会グラフに追加されないように、照会コンパイル・フェーズを変更します。

#### **ALLOWGENERATED**

「常に生成される」フィールドの値をクライアントが指定できます。

NONE 現在の設定をオフにします。

注: 「NONE」を除く設定値は、AND キーワードで結合できます。

## 許可

以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限

## デフォルトの PUBLIC 特権

なし

## 例

1. このクライアントについて、トリガー、制約、および MQT の実行をオフにします。

```
CALL ADMIN_SET_MAINT_MODE('INHIBIT_TRIG_RI_MQT')
```

2. 通常は生成されるフィールドにこのクライアントが値を指定できるようにします。これは通常はエラーになりますが、この設定に値を指定することは可能になります。

```
CALL ADMIN_SET_MAINT_MODE('ALLOWGENERATED')
```

3. このクライアントについて両方のタイプの設定を指定します。

```
CALL ADMIN_SET_MAINT_MODE('INHIBIT_TRIG_RI_MQT AND ALLOWGENERATED')
```

4. すべての保守モード設定をオフにします。

```
CALL ADMIN_SET_MAINT_MODE('NONE')
```

## ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO 表関数 - 表のサイズおよび状態に関する情報の検索

ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO 表関数は、現在カタログ・ビューで使用できない表のサイズおよび状態に関する情報を検索するメソッドを提供します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『ADMINTABINFO 管理ビュー』
- 296 ページの『ADMIN\_GET\_TAB\_INFO 表関数』

## ADMINTABINFO 管理ビュー

ADMINTABINFO 管理ビューは、表、マテリアライズ照会表 (MQT)、および階層表だけのサイズおよび状態に関する情報を戻します。これらの表タイプは SYSCAT.TABLES カatalog・ビューで、T (表)、S (マテリアライズ照会表)、および H (階層表) として報告されます。情報は表のデータ・パーティション・レベルとデータベース・パーティション・レベルの両方のものが戻されます。

DB2 pureScale 環境では、すべてのメンバーが単一のデータ物理区画上で作動するため、表について報告される値はすべてのメンバー上で同じです。これは、各メン

パーが別々のデータ物理区画で作動し、異なる値を報告するパーティション・データベース環境とは異なる点です。すべてのメンバーの値が常に同じであるため、ADMINTABINFO ビューおよび ADMIN\_GET\_TAB\_INFO 表関数は、DB2 pureScale インスタンスで実行されるたびに、各表に単一行のみを戻します。

スキーマは SYSIBMADM です。

戻すことができる情報の全リストについては、ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO によって戻される情報の表を参照してください。

## 許可

以下のいずれかの権限が必要です。

- ADMINTABINFO 管理ビューに対する SELECT 特権
- ADMINTABINFO 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

例 1: すべての表のサイズおよび状態に関する情報を検索します。

```
SELECT * FROM SYSIBMADM.ADMINTABINFO
```

例 2: データ密度の低い多数の表が使用する物理スペースの量を判別します。

```
SELECT TABSCHEMA, TABNAME, SUM(DATA_OBJECT_P_SIZE),  
       SUM(INDEX_OBJECT_P_SIZE), SUM(LONG_OBJECT_P_SIZE),  
       SUM(LOB_OBJECT_P_SIZE), SUM(XML_OBJECT_P_SIZE)  
FROM SYSIBMADM.ADMINTABINFO GROUP BY TABSCHEMA, TABNAME
```

例 3: ラージ RID の使用に適してはいるものの、現在使用可能になっていない表を識別します。

```
SELECT TABSCHEMA, TABNAME FROM SYSIBMADM.ADMINTABINFO  
WHERE LARGE_RIDS = 'P'
```

例 4: タイプ 1 形式の XML データがあり、Online Table Move でタイプ 2 形式に変換する必要がある表を識別します。

```
SELECT TABSCHEMA, TABNAME FROM SYSIBMADM.ADMINTABINFO  
WHERE XML_RECORD_TYPE=1
```

例 5: 表 T1 について収集された統計情報の現在のタイプを調べます。

```
SELECT SUBSTR(TABSCHEMA, 1, 10) AS TBSCHM, SUBSTR(TABNAME, 1, 10)  
       AS TBNAME, STATSTYPE FROM SYSIBMADM.ADMINTABINFO WHERE TABNAME = 'T1';
```

```
TBSCHM    TBNAME    STATSTYPE
```

```
-----  
DB2USER1  T1          U
```

1 record(s) selected.

## ADMIN\_GET\_TAB\_INFO 表関数

ADMIN\_GET\_TAB\_INFO 表関数は ADMINTABINFO 管理ビューと同じ情報を戻しますが、この関数ではスキーマおよび表名を指定することが可能です。

戻すことができる情報の全リストについては、ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO によって戻される情報の表を参照してください。

## 構文

```
▶▶ ADMIN_GET_TAB_INFO (—tabschema—, —tablename—) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

*tabschema*

スキーマ名を指定する、タイプ VARCHAR(128) の入力引数。

*tablename*

表、マテリアライズ照会表、または階層表それぞれの名前を指定する、タイプ VARCHAR(128) の入力引数。

## 許可

ADMIN\_GET\_TAB\_INFO 表関数に対する EXECUTE 特権。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: 表 DBUSER1.EMPLOYEE のサイズおよび状態に関する情報を検索します。

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_INFO('DBUSER1', 'EMPLOYEE'))  
AS T
```

例 2: 非パーティション表 (DBUSER1.EMPLOYEE) が存在し、関連オブジェクト (例えば索引や LOB など) がすべて 1 つの表スペースに保管されていると仮定します。表が表スペース内のどのくらいの物理スペースを使用しているかを計算します。

```
SELECT (data_object_p_size + index_object_p_size + long_object_p_size +  
lob_object_p_size + xml_object_p_size) as total_p_size  
FROM TABLE( SYSPROC.ADMIN_GET_TAB_INFO( 'DBUSER1', 'EMPLOYEE' )) AS T
```

表が別の表スペースに移動されるときにどのくらいのスペースが必要になるかを計算します。ここでは、新規の表スペースには元の表スペースと同じページ・サイズおよびエクステンツ・サイズがあるとします。

```
SELECT (data_object_l_size + index_object_l_size + long_object_l_size +
lob_object_l_size + xml_object_l_size) as total_l_size
FROM TABLE( SYSPROC.ADMIN_GET_TAB_INFO( 'DBUSER1', 'EMPLOYEE' )) AS T
```

例 3: 表 DBUSER1.EMPLOYEE 用のコンプレッション・ディクショナリーの合計サイズを判別します。

```
SELECT SUBSTR(TABSCHEMA,1,10) AS TBSHEMA, SUBSTR(TABNAME,1,10) AS TBNAME,
DICTIONARY_SIZE + XML_DICTIONARY_SIZE AS TOTAL_DICTIONARY_SIZE
FROM TABLE(SYSPROC.ADMIN_GET_TAB_INFO('DBUSER1','EMPLOYEE'))
```

例 4: マルチディメンション・クラスタリング表 SAMPLE.STAFF から再利用可能なスペースの量を判別します。

```
SELECT RECLAIMABLE_SPACE
FROM TABLE(SYSPROC.ADMIN_GET_TAB_INFO('SAMPLE','STAFF'))
```

### 使用上の注意

- *tabschema* と *tablename* の両方が指定される場合、その特定の表の情報のみが戻されます。
- *tabschema* が指定され、*tablename* が NULL または空ストリング (") の場合、指定したスキーマのすべての表に関する情報が戻されます。
- *tabschema* が NULL または空ストリング (") で、*tablename* が指定される場合、エラーは戻されます。特定の表の情報を取り出すには、その表がスキーマと表名の両方によって識別されることが必要です。
- *tabschema* と *tablename* の両方が NULL または空ストリング (") の場合、すべての表の情報が戻されます。
- *tabschema* または *tablename* が存在しないか、あるいは *tablename* が表名 (タイプ T)、マテリアライズ照会表名 (タイプ S)、または階層表名 (タイプ H) と一致しない場合、空の結果セットが戻されます。
- ADMIN\_GET\_TAB\_INFO 表関数が指定の表のデータを検索するとき、この表関数は SYSTABLES の対応する行に対する共有ロックを獲得します。これは、戻されるデータの整合性を確保するための動作です (例えば、情報の検索中に、検索されている表がドロップされないようにするなど)。ロックが保持されるのは、表関数の呼び出し期間中ではなく、表のサイズおよび状態に関する情報を検索する間だけです。
- SMS 表スペースの表の物理サイズが報告されますが、このサイズは論理サイズと同じです。
- 表で INPLACE の REORG がアクティブになっていると、データ・オブジェクトの物理サイズ (DATA\_OBJECT\_P\_SIZE) は計算されません。論理サイズだけが戻されます。INPLACE の REORG が表でアクティブになっているかどうかは、INPLACE\_REORG\_STATUS 出力の列を見ると分かります。

### REDISTRIBUTING\_PENDING

1. 指定された表について再配分は実行されていません N
2. 再配分の実行がデータベース・パーティション・グループで開始されましたが、表では開始されませんでした N
3. データを移動する前のフェーズで再配分が失敗しました N
4. データの移動のフェーズで再配分が失敗しました Y
5. 表について再配分が正常に完了し、コミットされました N

## 戻される情報

表 73. ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO によって戻される情報

列名	データ・タイプ	説明
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント
TABNAME	VARCHAR(128)	table_name - 表名モニター・エレメント
TABTYPE	CHAR(1)	表タイプ: <ul style="list-style-type: none"> <li>「H」 = 階層表</li> <li>「S」 = マテリアライズ照会表</li> <li>「T」 = 表</li> </ul>
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID モニター・エレメント
AVAILABLE	CHAR(1)	表の状態: <ul style="list-style-type: none"> <li>「N」 = 表は使用不可。表を使用できない場合、サイズおよび状態に関する他の出力列はすべて NULL になります。</li> <li>「Y」 = 表は使用可能。</li> </ul> <b>注:</b> リカバリー不能ロードでのロールフォワードを行うと、表の状態が使用不可になります。
DATA_OBJECT_L_SIZE	BIGINT	データ・オブジェクトの論理サイズ。表に対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表に対して物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。マルチディメンション・クラスタリング (MDC) 表および挿入時クラスタリング (ITC) 表の場合、このサイズにはブロック・マップ・オブジェクトの論理サイズが含まれます。戻されるサイズは、表に対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成されるオブジェクトの場合は Extent Map Page (EMP) エクステントの見積もりを考慮に入れます。このサイズは、基本表のみの論理サイズを表します。LOB データ、長形式データ、索引、および XML オブジェクトが消費するスペースは別の列で報告されます。
DATA_OBJECT_P_SIZE	BIGINT	データ・オブジェクトの物理サイズ。表に対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。MDC 表および ITC 表の場合、このサイズにはブロック・マップ・オブジェクトのサイズが含まれます。戻されるサイズは、表に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成されるオブジェクトの EMP エクステントが含まれます。このサイズは、基本表のみの物理サイズを表します。LOB データ、長形式データ、索引、および XML オブジェクトが消費するスペースは別の列で報告されます。



表 73. ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO によって戻される情報 (続き)

列名	データ・タイプ	説明
INDEX_OBJECT_L_SIZE	BIGINT	<p>索引オブジェクトの論理サイズ。表で定義される索引に対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表の索引データを保持するために物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。戻されるサイズは、索引に対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される索引の場合は EMP エクステントの見積もりを考慮に入れます。</p> <p>パーティション表のパーティション索引の場合、これは DATA_PARTITION_ID によって識別されるデータ・パーティションの索引パーティションを含む索引オブジェクトの論理サイズです。この値には、パーティション表の非パーティション索引は考慮に入れられません。パーティション索引と非パーティション索引の両方の情報については、ADMIN_GET_INDEX_INFO 関数を使用できます。</p>
INDEX_OBJECT_P_SIZE	BIGINT	<p>索引オブジェクトの物理サイズ。表で定義される索引に対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。戻されるサイズは、索引に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される索引の EMP エクステントが含まれます。</p> <p>パーティション表のパーティション索引の場合、これは DATA_PARTITION_ID によって識別されるデータ・パーティションの索引パーティションを含む索引オブジェクトの物理サイズです。この値には、パーティション表の非パーティション索引は考慮に入れられません。パーティション索引と非パーティション索引の両方の情報については、ADMIN_GET_INDEX_INFO 関数を使用できます。</p>
LONG_OBJECT_L_SIZE	BIGINT	<p>長形式オブジェクトの論理サイズ。表の長形式フィールド・データに対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表の長形式フィールド・データを保持するために物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。戻されるサイズは、長形式フィールド・データに対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される長形式フィールド・データの場合は EMP エクステントの見積もりを考慮に入れます。</p>
LONG_OBJECT_P_SIZE	BIGINT	<p>長形式オブジェクトの物理サイズ。表の長形式フィールド・データに対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。戻されるサイズは、長形式フィールド・データに割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される長形式フィールド・データの EMP エクステントが含まれます。</p>

表 73. ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO によって戻される情報 (続き)

列名	データ・タイプ	説明
LOB_OBJECT_L_SIZE	BIGINT	LOB オブジェクトの論理サイズ。表の LOB データに対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表の LOB データを保持するために物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。サイズには LOB 割り振りオブジェクトに対して論理的に割り振られるスペースが含まれます。戻されるサイズは、LOB データに対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される LOB データの場合は EMP エクステントの見積もりを考慮に入れます。
LOB_OBJECT_P_SIZE	BIGINT	LOB オブジェクトの物理サイズ。表の LOB データに対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。サイズには LOB 割り振りオブジェクトに対して割り振られるスペースが含まれます。戻されるサイズは、LOB データに割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される LOB データの EMP エクステントが含まれます。
XML_OBJECT_L_SIZE	BIGINT	XML オブジェクトの論理サイズ。表の XML データに対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表の XML データを保持するために物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。戻されるサイズは、XML データに対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される XML データの場合は EMP エクステントの見積もりを考慮に入れます。
XML_OBJECT_P_SIZE	BIGINT	XML オブジェクトの物理サイズ。表の XML データに対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。戻されるサイズは、XML データに割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される XML データの EMP エクステントが含まれます。
INDEX_TYPE	SMALLINT	現在表で使用されている索引のタイプを示します。タイプ 2 索引が使用されているときは 2 を戻します。
REORG_PENDING	CHAR(1)	「Y」は、REORG 推奨の変更がすでに表に適用されており、クラシック (オフライン) REORG が必要であることを示しています。それ以外の場合は「N」が戻されます。

表 73. ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO によって戻される情報 (続き)

列名	データ・タイプ	説明
INPLACE_REORG_STATUS	VARCHAR(10)	表に対する表のインプレース再編成の現在の状況。状況値は、以下のいずれかになります。 <ul style="list-style-type: none"> <li>• ABORTED (PAUSED 状態にあるが、RESUME は不可。STOP は必須)</li> <li>• EXECUTING</li> <li>• NULL (表に対して INPLACE の REORG が実行されていない場合)</li> <li>• PAUSED</li> </ul>
LOAD_STATUS	VARCHAR(12)	表に対するロード操作の現在の状況。状況値は、以下のいずれかになります。 <ul style="list-style-type: none"> <li>• IN_PROGRESS</li> <li>• NULL (表でロードが進行しておらず、表がロード・ペンディング状態になっていない場合)</li> <li>• PENDING</li> </ul>
READ_ACCESS_ONLY	CHAR(1)	表が「読み取りアクセス専用」の状態になっていれば「Y」、そうでなければ「N」になります。「N」の値を、表が完全にアクセス可能であるという意味に解釈するべきではありません。ロードが進行中またはペンディング状態の場合、「Y」の値は表データが読み取りアクセス可能であることを意味し、「N」の値は表がアクセス不能であることを意味します。同様に、表の状況が SET INTEGRITY ペンディングである場合 (SYSCAT.TABLES STATUS 列を参照)、「N」の値は表がアクセス不能であることを意味します。
NO_LOAD_RESTART	CHAR(1)	「Y」の値は、表が部分的にロードされている状態になっていることを示します。この場合、ロードを再始動することができません。この状態になっていなければ「N」の値が戻されます。
NUM_REORG_REC_ALTERS	SMALLINT	最後に再編成が行われてからこの表に対して実行された REORG 推奨の変更操作 (例えば直後に再編成を必要とする変更操作) の回数。
INDEXES_REQUIRE_REBUILD	CHAR(1)	非パーティション表において、表に定義されている索引のいずれかが再ビルドを必要とする場合は「Y」、必要としない場合は「N」。パーティション表において、DATA_PARTITION_ID によって識別されるデータ・パーティションの索引パーティションのいずれかが再ビルドを必要とする場合は「Y」、必要としない場合は「N」。

表 73. ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO によって戻される情報 (続き)

列名	データ・タイプ	説明
LARGE_RIDS	CHAR(1)	表がラージ行 ID (RID) を使用しているかどうかを示します (4 バイトのページ番号と 2 バイトのスロット番号)。 「Y」の値は表がラージ RID を使用していることを示し、「N」は使用していないことを示します。表がラージ RID をサポートしている (つまり表が LARGE 表スペースにある) もの、少なくとも表の索引の 1 つがまだ再編成されていないかまたは再ビルドされていない場合、「P」(保留)の値が戻されます。これは表が 4 バイトの RID を使用しているためです (これは表または索引を変換するためのアクションを取る必要があることを意味します)。
LARGE_SLOTS	CHAR(1)	表がラージ・スロット (これは 1 ページにつき 255 を超える行が可能です) を使用しているかどうかを示します。 「Y」の値は表がラージ・スロットを使用していることを示し、「N」は使用していないことを示します。表がラージ・スロットをサポートしている (つまり表が LARGE 表スペースにある) もの、表に対してまだオフラインの表の再編成または表の切り捨て操作が実行されていない場合、「P」(保留)の値が戻されます。これは、表が 1 ページにつき最大 255 行のラージ・スロットを使用しているためです。
DICTIONARY_SIZE	BIGINT	表ディクショナリーのサイズ (バイト)。表に行コンプレッション・ディクショナリーが存在する場合に行の圧縮で使用されます。履歴ディクショナリーが存在する場合、この値は現在のディクショナリー・サイズと履歴ディクショナリー・サイズの合計になります。
BLOCKS_PENDING_CLEANUP	BIGINT	blocks_pending_cleanup - クリーンアップ保留中のロールアウト済みブロック・モニター・エレメント

表 73. ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO によって戻される情報 (続き)

列名	データ・タイプ	説明
STATSTYPE	CHAR(1)	<ul style="list-style-type: none"> <li>「F」 = 表または索引のスキャンを行わずにシステムが作り上げた統計。これらの統計はメモリーに保管され、システム・カタログに保管されているものとは異なります。これは一時状態であり、最終的に完全な統計が DB2 によって収集され、システム・カタログに保管されます。</li> <li>「A」 = システムが非同期に収集した統計。統計は DB2 によりバックグラウンド・プロセスで自動的に収集され、システム・カタログに保管されました。</li> <li>「S」 = システムが同期に収集した統計。統計は DB2 によって SQL ステートメントのコンパイル中に自動的に収集されました。これらの統計はメモリーに保管され、システム・カタログに保管されているものとは異なります。これは一時状態であり、最終的に DB2 が統計をシステム・カタログに保管します。</li> <li>「U」 = ユーザーが収集した統計。統計の収集は、RUNSTATS、CREATE INDEX、LOAD、REDISTRIBUTE などのユーティリティーを使用するか、システム・カタログ統計を手動で更新することによって、ユーザーが開始しました。</li> <li>NULL = 不明なタイプ</li> </ul>
XML_RECORD_TYPE	SMALLINT	<p>現在表で使用されている XML レコードのタイプを示します。</p> <ul style="list-style-type: none"> <li>タイプ 1 (シングル・ノード) XML レコード・フォーマットが使用されている場合には 1。</li> <li>タイプ 2 (マルチノード) XML レコード・フォーマットが使用されている場合には 2。</li> <li>表に XML 列がない場合には NULL。</li> </ul>
RECLAIMABLE_SPACE	BIGINT	<p>DMS 表スペースの MDC 表または ITC 表の場合、この値は、RECLAIM EXTENTS オプションを指定して <b>REORG TABLE</b> コマンドを実行することで再利用可能なディスク・スペース量を示します。ディスク・スペースは、K バイト (KB) で報告されます。その他の表の場合、この値はゼロです。</p>
XML_DICTIONARY_SIZE	BIGINT	<p>XML ストレージ・オブジェクトにデータ・コンプレッション・ディクショナリーが存在する場合、データ圧縮に使用される XML ディクショナリーのサイズ (バイト単位)。表に XML 列がない場合、またはコンプレッション・ディクショナリーが作成されていない場合、値は 0 になります。</p>
AMT_STATUS	VARCHAR(12)	<p>表に対する ADMIN_MOVE_TABLE ストアード・プロシージャ呼び出しの現在の状況。表に進行中の動きがない場合は、値 'IN_PROGRESS' または NULL 値が戻されます。</p>

## ADMINTEMPCOLUMNS 管理ビューおよび

### ADMIN\_GET\_TEMP\_TABLES 表関数 - 一時表の列情報を取得する

ADMINTEMPCOLUMNS 管理ビューおよび ADMIN\_GET\_TEMP\_COLUMNS 表関数は、作成済み一時表および宣言済み一時表の列属性に関する情報を取得する方法を提供します。

カタログ・ビューには作成済み一時表のインスタンスの列属性情報が含まれますが、宣言済み一時表の表属性情報は含まれません。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『ADMINTEMPCOLUMNS 管理ビュー』
- 305 ページの『ADMIN\_GET\_TEMP\_COLUMNS 表関数』

## ADMINTEMPCOLUMNS 管理ビュー

ADMINTEMPCOLUMNS 管理ビューは、作成済み一時表および宣言済み一時表のインスタンスの列属性に関する情報を戻します。

スキーマは SYSIBMADM です。

戻すことができる情報の全リストについては、ADMINTEMPCOLUMNS 管理ビューおよび ADMIN\_GET\_TEMP\_COLUMNS で戻される情報の表を参照してください。

## 許可

以下のいずれかの権限が必要です。

- ADMINTEMPCOLUMNS 管理ビューに対する SELECT 特権
- ADMINTEMPCOLUMNS 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- ADMIN\_GET\_TEMP\_COLUMNS 表関数に対する EXECUTE 特権
- DATAACCESS 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

例 1: データベースに現在存在する作成済み一時表およびすべての宣言済み一時表のすべてのインスタンスの列属性に関する情報を戻します。

```
SELECT * FROM SYSIBMADM.ADMINTEMPCOLUMNS
```

例 2: データベースでアクティブになっているどの一時表がユーザー定義のデータ・タイプ USMONEY を使用しているかを判別します。

```
SELECT APPLICATION_HANDLE, TABSCHEMA, TABNAME
FROM SYSIBMADM.ADMINTEMPCOLUMNS
WHERE TYPENAME = 'USMONEY'
```

例 3: SYSTEM\_USER で宣言されたすべての宣言済み一時表の表スキーマ、表名、および列名を取得します。

```
SELECT T.TABSCHEMA, T.TABNAME, C.COLNAME
FROM SYSIBMADM.ADMINTEMPCOLUMNS C, SYSIBMADM.ADMINTEMPTABLES T
WHERE T.EMPTABTYPE = 'D'
AND T.INSTANTIATOR = SYSTEM_USER
AND T.TABSCHEMA = C.TABSCHEMA
AND T.TABNAME = C.TABNAME
```

## ADMIN\_GET\_TEMP\_COLUMNS 表関数

ADMIN\_GET\_TEMP\_TABLES 表関数は ADMINTEMPCOLUMNS 管理ビューと同じ情報を戻しますが、この関数ではスキーマ名および表名を指定することが可能です。

戻ることができる情報の全リストについては、ADMINTEMPCOLUMNS 管理ビューおよび ADMIN\_GET\_TEMP\_COLUMNS で戻される情報の表を参照してください。

### 構文

```
▶▶—ADMIN_GET_TEMP_COLUMNS—(—application_handle—,—tabschema—,—tabname—)————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *application\_handle*

アプリケーション・ハンドルを指定する、タイプ BIGINT の入力引数。  
*application\_handle* が指定されている場合、指定された接続に関するデータのみが戻されます。*application\_handle* が NULL である場合、すべての接続に関するデータが戻されます。

#### *tabschema*

スキーマ名を指定する、タイプ VARCHAR(128) の入力引数。

#### *tabname*

作成済み一時表の名前または宣言済み一時表の名前を指定する、タイプ VARCHAR(128) の入力引数。

### 許可

以下のいずれかの権限が必要です。

- ADMIN\_GET\_TEMP\_COLUMNS 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

例 1: 現行接続の宣言済み一時表 TEMPEMPLYEE の列情報を取得します。

```
SELECT *
FROM TABLE (
  SYSPROC.ADMIN_GET_TEMP_COLUMNS(
    APPLICATION_ID(), 'SESSION', 'TEMPEMPLYEE')
AS T
```

### 使用上の注意

- *tabschema* と *tablename* の両方が指定される場合、その特定の一時表の情報のみが戻されます。
- *tabschema* が指定され、*tablename* が NULL または空ストリング (") の場合、指定したスキーマのすべての表に関する情報が戻されます。
- *tabschema* が NULL または空ストリング (") で、*tablename* が指定される場合、エラーは戻されます。特定の一時表の情報を取り出すには、その表がスキーマと表名の両方によって識別されることが必要です。
- *tabschema* と *tablename* の両方が NULL または空ストリング (") の場合、*application\_handle* の値に応じて、特定の接続 (またはすべての接続) のすべての一時表に関する情報が戻されます。
- *tabschema* または *tablename* が存在しないか、*tablename* が一時表名と一致しないか、あるいは識別された一時表のインスタンスがデータベースに存在しない場合、空の結果セットが戻されます。

### 戻される情報

表 74. ADMINTEMPCOLUMNS 管理ビューおよび ADMIN\_GET\_TEMP\_COLUMNS で戻される情報

列名	データ・タイプ	説明
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル・モニター・エレメント
APPLICATION_NAME	VARCHAR(256)	appl_name - アプリケーション名モニター・エレメント
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント
TABNAME	VARCHAR(128)	table_name - 表名モニター・エレメント
COLNAME	VARCHAR(128)	列の名前。
COLNO	SMALLINT	表の中のこの列の番号 (0 から始まる)。
TYPESCHEMA	VARCHAR(128)	列のデータ・タイプのスキーマ名。
TYPENAME	VARCHAR(128)	列のデータ・タイプの非修飾名。
LENGTH	INTEGER	データの最大長。特殊タイプの場合は 0。LENGTH 列は DECIMAL フィールドの精度を示し、10 進浮動小数点列に必要なストレージのバイト数を示します。つまり、DECFLOAT(16) の場合には 8 列、DECFLOAT(34) の場合には 16 列です。



表 74. ADMINTEMPCOLUMNS 管理ビューおよび ADMIN\_GET\_TEMP\_COLUMNS で戻される情報 (続き)

列名	データ・タイプ	説明
SCALE	SMALLINT	列タイプが DECIMAL の場合は位取りで、列タイプが TIMESTAMP の場合には秒の小数部分の桁数。その他の場合には 0。
DEFAULT	VARCHAR(254)	列のデータ・タイプに適した定数、特殊レジスター、または cast 関数で表された表の列のデフォルト値。キーワード NULL の場合もあります。値は、デフォルト値として指定された値から変換されることがあります。例えば、日時の定数は ISO フォーマットで表示され、cast 関数名はスキーマ名で修飾され、ID は区切り文字で区切られません。DEFAULT 節が指定されていないか、または列がビューの列の場合は、NULL 値になります。
NULLS	CHAR(1)	列の NULL 可能性属性。 <ul style="list-style-type: none"> <li>• Y = 列は NULL 可能。</li> <li>• N = 列は NULL 不可。</li> </ul> 式または関数から派生したビューの列の場合、値は N である可能性があります。それでも、このビューを使用するステートメントが処理され、算術計算エラーの警告を出された場合は、この列に NULL 値が入ります。
CODEPAGE	SMALLINT	この列のデータに使用されるコード・ページ。列が FOR BIT DATA として定義されている場合、またはストリング・タイプではない場合は 0。
LOGGED	CHAR(1)	LOB タイプまたは LOB に基づく特殊タイプの列だけに適用されます。それ以外はブランクです。 <ul style="list-style-type: none"> <li>• Y = 列のログ記録を取る。</li> <li>• N = 列のログ記録を取らない。</li> </ul>
COMPACT	CHAR(1)	LOB タイプまたは LOB に基づく特殊タイプの列だけに適用されます。それ以外はブランクです。 <ul style="list-style-type: none"> <li>• Y = ストレージ内で列を圧縮する。</li> <li>• N = 列を圧縮しない。</li> </ul>
INLINE_LENGTH	INTEGER	基本表に保管できる XML 文書または構造化タイプのインスタンスの内部表記の最大サイズ (バイト数)。適用されない場合には 0。
IDENTITY	CHAR(1)	<ul style="list-style-type: none"> <li>• Y = ID 列。</li> <li>• N = ID 列ではない。</li> </ul>
GENERATED	CHAR(1)	生成される列の型。 <ul style="list-style-type: none"> <li>• A = 列の値が常に生成される</li> <li>• D = 列の値がデフォルトで生成される</li> <li>• ブランク = 列は生成されない</li> </ul>

## ADMINTEMPTABLES 管理ビューおよび

### ADMIN\_GET\_TEMP\_TABLES 表関数 - 一時表の情報を取得する

ADMINTEMPTABLES 管理ビューおよび ADMIN\_GET\_TEMP\_TABLES 表関数は、作成済み一時表および宣言済み一時表のインスタンスの表属性およびインスタンス生成時間に関する情報を検索する方法を提供します。

カタログ・ビューには作成済み一時表の表属性情報が含まれますが、宣言済み一時表の表属性情報は含まれません。さらに、カタログ・ビューには、作成済み一時表または宣言済み一時表の表インスタンス生成時間に関する情報は含まれません。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『ADMINTEMPTABLES 管理ビュー』
- 309 ページの『ADMIN\_GET\_TEMP\_TABLES 表関数』

### ADMINTEMPTABLES 管理ビュー

ADMINTEMPTABLES 管理ビューは、作成済み一時表および宣言済み一時表のインスタンスの表属性およびインスタンス生成時間に関する情報を戻します。

スキーマは SYSIBMADM です。

戻すことができる情報の全リストについては、ADMINTEMPTABLES 管理ビューおよび ADMIN\_GET\_TEMP\_TABLES で戻される情報の表を参照してください。

### 許可

以下のいずれかの権限が必要です。

- ADMINTEMPTABLES 管理ビューに対する SELECT 特権
- ADMINTEMPTABLES 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- ADMIN\_GET\_TEMP\_TABLES 表関数に対する EXECUTE 特権
- DATAACCESS 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

例 1: データベースに現在存在する作成済み一時表および宣言済み一時表のすべてのインスタンスの表属性およびインスタンス生成時間に関する情報を取得します。

```
SELECT * FROM SYSIBMADM.ADMINTEMPTABLES
```

例 2: どの接続に作成済み一時表のインスタンスが含まれているかを判別します。

```
SELECT APPLICATION_HANDLE, TABSCHEMA, TABNAME
FROM SYSIBMADM.ADMINTEMPTABLES
WHERE TEMPTABTYPE = 'C'
```

例 3: データベースに接続したユーザーによってインスタンス生成されたすべての表に宣言された、すべての宣言済み一時表の表属性およびインスタンス生成時間に関する情報を検索します。

```
SELECT TABSCHEMA, TABNAME, ONCOMMIT, ONROLLBACK,
INSTANTIATION_TIME
FROM SYSIBMADM.ADMINTEMPTABLES
WHERE TEMPTABTYPE = 'D' AND INSTANTIATOR = SYSTEM_USER
```

## ADMIN\_GET\_TEMP\_TABLES 表関数

ADMIN\_GET\_TEMP\_TABLES 表関数は ADMINTABINFO 管理ビューと同じ情報を戻しますが、この関数ではスキーマ名および表名を指定することが可能です。

戻ることができる情報の全リストについては、ADMINTEMPTABLES 管理ビューおよび ADMIN\_GET\_TEMP\_TABLES で戻される情報の表を参照してください。

### 構文

```
►►—ADMIN_GET_TEMP_TABLES—(—application_handle—,—tabschema—,—tablename—)————►►
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *application\_handle*

アプリケーション・ハンドルを指定する、タイプ BIGINT の入力引数。

*application\_handle* が指定されている場合、指定された接続に関するデータのみが戻されます。*application\_handle* が NULL である場合、すべての接続に関するデータが戻されます。

#### *tabschema*

スキーマ名を指定する、タイプ VARCHAR(128) の入力引数。

#### *tablename*

作成済み一時表の名前または宣言済み一時表の名前を指定する、タイプ VARCHAR(128) の入力引数。

### 許可

以下のいずれかの権限が必要です。

- ADMIN\_GET\_TEMP\_TABLES 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: すべての接続の作成済み一時表 DBUSER1.EMPLOYEE のすべてのインスタンスの表属性およびインスタンス生成時間に関する情報を取得します。

```
SELECT TABSCHEMA, TABNAME, ONCOMMIT, ONROLLBACK, INSTANTIATION_TIME
FROM TABLE (SYSPROC.ADMIN_GET_TEMP_TABLES(NULL, 'DBUSER1', 'EMPLOYEE'))
AS T
```

例 2: 現行接続でのユーザー一時表のすべてのインスタンスのインスタンス生成時間および表スペース ID を取得します。

```
SELECT TABSCHEMA, TABNAME, INSTANTIATION_TIME, TBSP_ID
FROM TABLE (SYSPROC.ADMIN_GET_TEMP_TABLES(APPLICATION_ID(), '', ''))
AS T
```

## 使用上の注意

- *tabschema* と *tablename* の両方が指定される場合、その特定の一時表の情報のみが戻されます。
- *tabschema* が指定され、*tablename* が NULL または空ストリング (") の場合、指定したスキーマのすべての表に関する情報が戻されます。
- *tabschema* が NULL または空ストリング (") で、*tablename* が指定される場合、エラーは戻されます。特定の一時表の情報を取り出すには、その表がスキーマと表名の両方によって識別されることが必要です。
- *tabschema* と *tablename* の両方が NULL または空ストリング (") の場合、*application\_handle* の値に応じて、特定の接続 (またはすべての接続) のすべての一時表に関する情報が戻されます。
- *tabschema* または *tablename* が存在しないか、*tablename* が一時表名と一致しないか、あるいは識別された一時表のインスタンスがデータベースに存在しない場合、空の結果セットが戻されます。

## 戻される情報

表 75. ADMINTEMPTABLES 管理ビューおよび ADMIN\_GET\_TEMP\_TABLES で戻される情報

列名	データ・タイプ	説明
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル・モニター・エレメント
APPLICATION_NAME	VARCHAR(256)	appl_name - アプリケーション名モニター・エレメント
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント
TABNAME	VARCHAR(128)	table_name - 表名モニター・エレメント
INSTANTIATOR	VARCHAR(128)	作成済み一時表がインスタンス生成されたか、または宣言済み一時表が宣言された許可 ID。
INSTANTIATORTYPE	CHAR(1)	• U = インスタンス生成者は個々のユーザー
TEMPTABTYPE	CHAR(1)	一時表タイプ: • C = 作成済み一時表 • D = 宣言済み一時表
INSTANTIATION_TIME	TIMESTAMP	作成済み一時表がインスタンス生成されたか、または宣言済み一時表が宣言された時刻。
COLCOUNT	SMALLINT	列の数。継承された列がある場合は、それも含む。

表 75. ADMINTEMPTABLES 管理ビューおよび ADMIN\_GET\_TEMP\_TABLES で戻される情報 (続き)

列名	データ・タイプ	説明
TAB_FILE_ID	BIGINT	table_file_id - 表のファイル ID (FID)。
TBSP_ID	BIGINT	tablespace_id - 現行データベースで使用する表スペースを表す固有の整数。
PMAP_ID	SMALLINT	この表によって現在使用されている分散マップの ID。
PARTITION_MODE	CHAR(1)	パーティション・データベース・システム内のデータベース・パーティション間でどのようにデータが配分されるかを示します。 <ul style="list-style-type: none"> <li>• H = ハッシュ</li> <li>• ブランク = データベース・パーティションなし</li> </ul>
CODEPAGE	SMALLINT	オブジェクトのコード・ページ。これが、すべての文字の列および式で生成された列のデフォルトのコード・ページです。
ONCOMMIT	CHAR(1)	COMMIT 操作の実行時にこの表で行うアクションを指定します。 <ul style="list-style-type: none"> <li>• D = 行の削除</li> <li>• P = 行の保存</li> </ul>
ONROLLBACK	CHAR(1)	ROLLBACK 操作の実行時にこの表で行うアクションを指定します。 <ul style="list-style-type: none"> <li>• D = 行の削除</li> <li>• P = 行の保存</li> </ul>
LOGGED	CHAR(1)	この表をログ対象にするかどうかを指定します。 <ul style="list-style-type: none"> <li>• N = ログ対象にしません</li> <li>• Y = ログ対象です</li> </ul>

## 管理タスク・スケジューラー・ルーチンおよびビュー

### ADMIN\_TASK\_ADD プロシージャ - 新規タスクのスケジュール

ADMIN\_TASK\_ADD プロシージャは、管理タスク、つまりプロシージャの内部にカプセル化できる処理をスケジュールに入れます。

#### 構文

```
▶▶ADMIN_TASK_ADD(—name—,—begin_timestamp—,—end_timestamp—,——————▶
▶max_invocations—,—schedule—,—procedure_schema—,—procedure_name—,—————▶
▶—procedure_input—,—options—,—remarks—)—————▶▶▶
```

スキーマは SYSPROC です。

#### プロシージャ・パラメーター

##### name

タスクの名前を指定する、タイプ VARCHAR (128) の入力引数。この引数は NULL にできません。

### *begin\_timestamp*

タスクの実行を開始できる最も早い時刻を指定する、タイプ `TIMESTAMP` の入力引数。この引数の値は、過去の日付にしたり、*end\_timestamp* より後の日付にしたりすることはできません。

タスク実行が開始されるタイミングは、この引数と *schedule* 引数が定義されている方法によって異なります。

- *begin\_timestamp* 引数が `NULL` 以外の場合:
  - *schedule* 引数が `NULL` の場合、タスク実行は *begin\_timestamp* に開始されます。
  - *schedule* 引数が `NULL` 以外の場合、タスク実行は、スケジュールされている次の時刻または *begin\_timestamp* よりも後に開始されます。
- *begin\_timestamp* 引数が `NULL` の場合:
  - *schedule* 引数が `NULL` の場合、タスク実行は即時に開始されます。
  - *schedule* 引数が `NULL` 以外の場合、タスク実行は、スケジュールされている次の時刻に開始されます。

### *end\_timestamp*

タスクの実行を開始できる最後の時刻を指定する、タイプ `TIMESTAMP` の入力引数。この引数の値は、過去の日付にしたり、*begin\_timestamp* より前の日付にしたりすることはできません。引数が `NULL` の場合、タスクは、スケジュールされたとおりに無期限に実行されます。

実行中のタスクは、その *end\_timestamp* で中断されることはありません。

### *max\_invocations*

タスクで許容される最大実行数を指定する、タイプ `INTEGER` の入力引数。引数が `NULL` の場合、タスクを実行できる回数に制限はありません。引数が `0` の場合、タスクは実行されません。

*schedule* が `NULL` 以外の場合、この値がスケジュールに適用されます。

*end\_timestamp* と *max\_invocations* の両方が指定されている場合、*end\_timestamp* が優先されます。つまり、*end\_timestamp* タイム・スタンプに達すると、タスクの回数とその時点で *max\_invocations* の値に達していなくても、そのタスクが再び実行されることはありません。

### *schedule*

特定の時間にタスクが実行されるようにスケジュールを指定する、タイプ `VARCHAR(1024)` の入力引数。引数が `NULL` の場合、タスクは、特定の時間にスケジュールされません。

*schedule* ストリングは、UNIX `cron` 形式を使用して指定する必要があります。

複数のスケジュールはサポートされていません。

### *procedure\_schema*

このタスクが実行するプロシージャースキーマを指定する、タイプ `VARCHAR(128)` の入力引数。この引数は `NULL` にできません。

### *procedure\_name*

このタスクが実行するプロシージャーの名前を指定する、タイプ `VARCHAR(128)` の入力引数。この引数は `NULL` にできません。

### *procedure\_input*

このタスクが実行するプロシージャの入力引数を指定する、タイプ CLOB(2M) の入力引数。この引数には、1 行のデータを戻す SQL ステートメントが含まれている必要があります。戻り値は、引数としてプロシージャに渡されます。この引数が NULL の場合、引数はプロシージャに渡されません。

SQL ステートメントによって戻される列数は、プロシージャの引数の総数 (およびタイプ) と一致していなければならない、単一行を含んでいる必要があります。出力引数の場合、値自体は無視されますが、プロシージャが要求するのと同じ SQL データ・タイプでなければなりません。

この SQL ステートメントは、タスクが実行されるたびに実行されます。SQL ステートメントが失敗すると、タスクの状況は NOTRUN に設定され、特定の SQLCODE 情報が記録されます。ステートメントが結果セットや行を戻さず、複数の行または結果セットを戻す場合、タスクは実行されません。タスクの状況は NOTRUN に設定され、この引数が無効であることを示す SQLCODE SQL1465N が設定されます。

ステートメントの結果に直列化 XML パラメーターが含まれる場合、結合されるすべての XML パラメーターの合計サイズは 256 キロバイトに制限されます。結果がこのしきい値を超えた場合、タスクの状況は NOTRUN に設定されます。データ切り捨てが発生したことを示す、SQLCODE -302 および SQLSTATE 22001 が設定されます。

タスクの状況を表示するには、SYSTOOL.ADMIN\_TASK\_STATUS ビューを使用します。

### *options*

タイプ VARCHAR(512) の入力引数。この引数は NULL でなければなりません。

### *remarks*

タスクの説明を指定する、タイプ VARCHAR(254) の入力引数。この引数はオプションであり、NULL にすることができます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 使用上の注意

ADMIN\_TASK\_ADD プロシージャを呼び出す前に、SYSTOOLSPACE 表スペースが存在している必要があります。存在しない場合、プロシージャは SQL0204N エラー・メッセージを戻します。

タスクがスケジュールされている場合は、現行セッション・ユーザーの許可 ID が記録されます。スケジューラーは、タスクを実行するときこのセッション許可 ID に切り替えます。

管理タスク・スケジューラーは、指定したユーザー ID およびパスワードを使用せずにデータベース接続を実行するプロシージャの実行をサポートしていません。例えば、ADMIN\_CMD プロシージャを使用して、データベースから LOAD を実行できます。ソース・データベースへの接続は、現在接続中のデータベースに提供されているユーザー ID およびパスワードを使用して確立されます。このタイプの LOAD 操作は、タスク・スケジューラーでは実行できません。

無効な引数がプロシージャに渡されると、SQL0171N が戻されます。メッセージのトークンは、無効な引数およびプロシージャの名前を示します。

タスクは、作業単位がコミットされて、スケジューラーがタスク定義をフェッチするまでは、実行をスケジュールすることはできません。

スケジューラーは、新規および更新されたタスクがないかどうか 5 分おきに調べます。予期されるとおりにタスクが実行されるようにするには、*begin\_timestamp*、*end\_timestamp*、および *schedule* 引数で定義される最も早い開始時刻は、作業単位がコミットされてから少なくとも 5 分後でなければなりません。

タスクがスケジューラーによって実行できるようにするには、データベースはすべてのデータベース・パーティションでアクティブである必要があります。

パーティション・データベース環境では、ADMIN\_TASK\_ADD プロシージャは任意のデータベース・パーティションから呼び出すことができます。ただし、スケジューラーはカタログ・データベース・パーティションからすべてのタスクを実行します。

*begin\_timestamp*、*end\_timestamp*、および *schedule* はサーバーの時間帯に基づいています。夏時間調整 (DST) の移行期間にタスクをスケジュールする場合には、特に注意が必要です。タスクが午前 2 時 1 分に実行されるようスケジュールされている場合に、それが時間を早めるときに当たると、時間は午前 2 時から午前 3 時までスキップされるため、そのタスクは実行されません。DST の変更がデータベースがアクティブな状態で行われ、DST の変更時間中にスケジュールされたタスクがない場合、DST の変更後にスケジュールされている最初のタスクは実行されません。2 番目以降のタスクは正常に実行されます。例えば、午前 2 時から午前 3 時の間にスケジュールされているタスクはなく、午前 3 時 10 分にスケジュールされているタスクがあり、それが DST の変更後の最初のタスクである場合、このタスクは実行されません。これは、DST 変更のどれくらい後にタスクがスケジュールされているかとは関係がありません。例えば、DST より後の最初のタスクが DST の変更から 1 日後や 1 カ月後にスケジュールされていたとしても、そのタスクが DST 変更後の最初のタスクであり、データベースがその間ずっとアクティブな状態であった限り、そのタスクは実行されません。一方、時間が 1 時間戻されるときには、午前 2 時から 3 時の間にスケジュールされているタスクは 2 回実行されることになります。必要な動作が実行されるように、夏時間調整時間の調整はユーザーの責任で行う必要があります。



スケジューラーは常に、*procedure\_schema* および *procedure\_name* で指定されたプロシージャーを呼び出した後にコミットされます。トランザクションのロールバックが必要な場合、ロールバックはプロシージャーの内部で行われる必要があります。

タスク名が固有のものでない場合、プロシージャーは SQL0601N で失敗します。

## 例

例 1: 毎日深夜 12:00 にオンライン TSM バックアップを実行するタスクを作成し、すぐに有効にする。

```
CALL SYSPROC.ADMIN_TASK_ADD
( 'DAILY TSM BACKUP',
  CURRENT_TIMESTAMP,
  NULL,
  NULL,
  '0 0 * * *',
  'SYSPROC',
  'ADMIN_CMD',
  'VALUES(''BACKUP DATABASE SALES ONLINE USE TSM WITHOUT PROMPTING'')',
  NULL,
  NULL )
```

例 2: 1 時間ごとにイベント・モニターをフラッシュするタスクをスケジュールする:

1. "em" というイベント・モニターをフラッシュする SQL プロシージャーを PROD スキーマに作成します。

```
CREATE PROCEDURE FLUSH_EVENT_MONITOR()
SPECIFIC FLUSH_EVENT_MONITOR
LANGUAGE SQL
BEGIN
DECLARE stmt VARCHAR(100) ;
SET stmt = 'FLUSH EVENT MONITOR em' ;
EXECUTE IMMEDIATE stmt ;
END
```

注: FLUSH EVENT MONITOR SQL ステートメントをプロシージャー内で直接呼び出すことはできません。ただし、EXECUTE IMMEDIATE を使用できます。

2. ADMIN\_TASK\_ADD を呼び出してタスクをスケジュールする:

```
CALL SYSPROC.ADMIN_TASK_ADD
('FLUSH_EVENT_MONITOR EVERY HOUR',
 NULL,
 NULL,
 NULL,
 '0 0-23 * * *',
 'PROD',
 'FLUSH_EVENT_MONITOR',
 NULL,
 NULL,
 NULL )
```

## UNIX cron 形式:

UNIX cron 形式は、ADMIN\_TASK\_ADD および ADMIN\_TASK\_UPDATE プロシージャーの *schedule* パラメーターで時刻を指定するために使用されます。

cron 形式には、少なくとも 1 つの空白で区切られた 5 つの日時フィールドがあります。フィールド値を空白にすることはできません。スケジュールされたタスクは、分、時間、および月フィールドが現在の日時と一致し、2 つの日付フィールド (日、あるいは曜日) のうち少なくとも 1 つが現在日付と一致する場合に実行されます。

表 1 では、日時フィールドとその許容値が cron 形式でリストされています。

表 76. UNIX cron 形式のフィールド名および値

フィールド名	許容値
分	0-59
時	0-23
日	1-31
月	<ul style="list-style-type: none"> <li>1-12。1 は 1 月、2 は 2 月、以下 3 月、4 月と続きます。</li> <li>英語の月名に基づいた大文字、小文字、および大/小文字混合の 3 文字のストリング。例: jan、feb、mar、apr、may、jun、jul、aug、sep、oct、nov、または dec。</li> </ul>
曜日	<ul style="list-style-type: none"> <li>0-7。0 または 7 は日曜日、1 は月曜日、以降同様です。</li> <li>英語の曜日名に基づいた大文字、小文字、または大/小文字混合の 3 文字のストリング: mon、tue、wed、thu、fri、sat、または sun。</li> </ul>

### 範囲およびリスト

数値の範囲を使用できます。範囲は、ハイフンで区切られた 2 つの数値で指定されます。指定した範囲は包括的です。例えば、時間項目の範囲 8-11 は、8、9、10 および 11 時に実行されることを示します。

リストを使用できます。リストは、コンマで区切られた数値または範囲のセットです。例:

```
1,2,5,9
0-4,8-12
```

### 無制限の範囲

フィールドには、アスタリスク (\*) を含めることができます。これは、フィールド内のすべての可能な値を表します。

コマンドを実行する日は、日および曜日の 2 つのフィールドで指定できます。両方のフィールドがアスタリスク以外の値の使用で制限されている場合、コマンドはいずれかのフィールドが現在時刻と一致したときに実行されます。例えば、値 30 4 1,15 \* 5 の場合、コマンドは各月の 1 日、15 日、および毎週金曜日の午前 4 時 30 分に実行されます。

## ステップ値

ステップ値は、範囲とともに使用することができます。構文 *range/step* は、範囲および実行間隔を定義します。

*first-last/step* を指定した場合、実行は *first* で行われ、次に *first* から *step* ずつ離れた連続するすべての値で (*last* まで) 行われます。

例えば、2 時間おきにコマンド実行を指定するには、0-23/2 を使用します。この式は、値 0,2,4,6,8,10,12,14,16,18,20,22 と同等です。

*\*/step* を指定した場合、実行は *step* の間隔ごとに無制限の範囲で行われます。例えば、2 時間おきに実行する場合には、0-23/2 の代わりに *\*/2* を使用します。

## 例

表 2 には、さまざまなスケジュール・シナリオで ADMIN\_TASK\_ADD または ADMIN\_TASK\_UPDATE プロシージャの *schedule* 引数に使用できる値がリストされています。

表 77. タスク・スケジュールおよび該当する *schedule* 引数値の例

希望するタスク・スケジュール	<i>schedule</i> 値
毎月曜日の午後 2 時 10 分	10 14 * * 1
毎日午前 0 時	0 0 * * *
毎平日の午前 0 時	0 0 * * 1-5
月の 1 日および 15 日の午前 0 時	0 0 1,15 * *
各年の 11 月 17 日、21 日および 29 日 と、11 月の毎月曜日および毎水曜日の午後 6 時 32 分	32 18 17,21,29 11 mon,wed

## ADMIN\_TASK\_LIST 管理ビュー - スケジューラーのタスクに関する情報の取得

ADMIN\_TASK\_LIST 管理ビューは、管理タスク・スケジューラーで定義されている各タスクに関する情報を取得します。

スキーマは SYSTOOLS です。

このビューは、最初に ADMIN\_TASK\_ADD プロシージャが呼び出されるときに作成されます。

## 許可

ADMIN\_TASK\_LIST 管理ビューに対する SELECT または CONTROL 特権。データベースが **RESTRICTIVE** オプションで作成された場合を除き、デフォルトで SELECT 特権が PUBLIC に付与されます。

ADMIN\_TASK\_LIST ビューを照会した場合、セッション許可 ID を使用して作成されたタスクのみが戻されます。SYSADM、SYSCTRL、SYSMAINT、または DBADM の各権限を持っている場合、すべてのタスクが戻されます。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

スケジューラー内のタスクのリストの要求:

```
SELECT * from SYSTOOLS.ADMIN_TASK_LIST
```

### 戻される情報

表 78. ADMIN\_TASK\_LIST 管理ビューによって戻される情報

列名	データ・タイプ	説明
NAME	VARCHAR(128)	タスクの名前。
TASKID	INTEGER	タスク ID。
OWNER	VARCHAR(128)	タスクを作成したユーザーのセッション許可 ID。
OWNERTYPE	VARCHAR(1)	許可 ID タイプ。有効な値は以下のとおりです。 • U - ユーザー
BEGIN_TIME	TIMESTAMP	タスクが最初に実行可能になるときのタイム・スタンプ。 <sup>1</sup>
END_TIME	TIMESTAMP	タスクが最後に実行可能になるときのタイム・スタンプ。 <sup>1</sup>  この列が NULL である場合、MAX_INVOCATIONS が指定されていない限り、タスクは無限に実行できます。
MAX_INVOCATIONS	INTEGER	タスクで許容される最大実行数。この列が NULL である場合、END_TIME が指定されていない限り、タスクは無限に実行できます。
SCHEDULE	VARCHAR(1024)	タスクのスケジュール (UNIX cron 形式)
PROCEDURE_SCHEMA	VARCHAR(128)	このタスクが実行するプロシーチャーのスキーマ。
PROCEDURE_NAME	VARCHAR(128)	このタスクが実行するプロシーチャーの名前。
PROCEDURE_INPUT	CLOB(2M)	このタスクが実行するプロシーチャーの入力パラメーター。この列が NULL である場合、入力パラメーターは存在しません。
OPTIONS	VARCHAR(512)	タスクの動作に影響するオプション。
UPDATE_TIME	TIMESTAMP	update_time - 更新応答時間モニター・エレメント
REMARKS	VARCHAR(254)	タスクの説明。

注:

- <sup>1</sup> BEGIN\_TIME および END\_TIME はデータベース・サーバーの時間帯に基づいています。夏時間調整時間 (DST) の調整はユーザーの責任で行う必要があります。

## ADMIN\_TASK\_REMOVE プロシージャ - スケジュールされたタスクまたはタスク状況レコードの除去

ADMIN\_TASK\_REMOVE プロシージャは、スケジュールされた管理タスク、つまりプロシージャの内部にカプセル化できる作業を除去します。また、タスク状況レコードも除去します。

### 構文

```
►►ADMIN_TASK_REMOVE(—name—,—end_timestamp—)◄◄
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

*name*

タスクの名前を指定する、タイプ VARCHAR (128) の入力引数。

*end\_timestamp*

状況レコードの *end\_timestamp* タイム・スタンプを指定するタイプ TIMESTAMP の出力引数。

### 許可

以下のいずれかの権限が必要です。

- ADMIN\_TASK\_REMOVE プロシージャに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

ステートメント許可 ID を使用してプロシージャを実行することが可能な場合でも、タスクおよび状況レコードが正常に除去されるかどうかは現行のセッション許可 ID の値に依存します。現行のセッション許可 ID は、タスクの作成時に記録されたセッション許可 ID と一致する必要があります。タスクまたは状況レコードを除去できるのは、SYSADM、SYSCTRL、SYSMAINT、または DBADM の各権限を持ったユーザーです。許可を持たないユーザーがタスクまたは状況レコードを除去しようとする、SQL0551N が戻されます。

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 使用上の注意

作業単位がコミットされるまでタスクは除去されません。

タスク除去の動作は、*name* および *end\_timestamp* 引数が定義される方法によって異なります。

- *end\_timestamp* 引数が NULL の場合:
  - *name* 引数が NULL の場合、すべてのタスクおよび状況レコードが除去されます。1 つ以上のタスクが現在実行中の場合、タスクおよび関連した状況レコードは除去されません。この場合、SQL1464W が戻されます。
  - *name* 引数が NULL 以外の場合、*name* と一致するタスク・レコードは除去されます。指定されたタスクが現在実行中の場合、タスクは除去されず、SQL20453N が戻されます。指定されたタスクが除去されると、関連したすべての状況レコードが除去されます。
- *end\_timestamp* 引数が NULL 以外の場合:
  - *name* 引数が NULL の場合、*end\_timestamp* 以前の *end\_timestamp* タイム・スタンプを持つすべてのタスクおよび状況レコードが除去されます。タスク・レコードは除去されません。プロシージャは、状況値 RUNNING を持つ状況レコードを除去しません。
  - *name* 引数が NULL 以外の場合、タスクの *end\_timestamp* タイム・スタンプが *end\_timestamp* 以前であれば、*name* と一致するタスクの状況レコードが除去されます。タスク・レコードは除去されません。プロシージャは、状況値 RUNNING を持つ状況レコードを除去しません。

ユーザーが存在しないタスクを除去しようとする、SQL0204N が戻されます。

## 例

「DAILY TSM BACKUP」というバックアップ・タスクを除去する場合:

```
CALL SYSPROC.ADMIN_TASK_REMOVE('DAILY TSM BACKUP', NULL)
```

## ADMIN\_TASK\_STATUS 管理ビュー - タスク状況情報の取得

ADMIN\_TASK\_STATUS 管理ビューは、管理タスク・スケジューラーでのタスク実行の状況に関する情報を取得します。

スキーマは SYSTOOLS です。

このビューは、最初に ADMIN\_TASK\_ADD プロシージャが呼び出されるときに作成されます。

## 許可

ADMIN\_TASK\_STATUS 管理ビューに対する SELECT または CONTROL 特権。データベースが **RESTRICTIVE** オプションで作成された場合を除き、デフォルトで SELECT 特権が PUBLIC に付与されます。

ADMIN\_TASK\_STATUS ビューを照会した場合、セッション許可 ID によって作成されたタスク状況レコードのみが戻されます。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

例 1: スケジューラー内のタスクの状況の要求:

```
SELECT * from SYSTOOLS.ADMIN_TASK_STATUS
```

例 2: SQLERRM 関数を使用した SQLERRMC 列のデータのフォーマット:

```
SELECT TASKID, STATUS, SQLCODE, SQLSTATE, RC,  
       VARCHAR( SQLERRM( 'SQL' || CHAR( ABS(SQLCODE) ),  
                       SQLERRMC, x'FF', 'en_US', 1 ), 256) AS MSG_TXT  
FROM SYSTOOLS.ADMIN_TASK_STATUS
```

## 戻される情報

表 79. ADMIN\_TASK\_STATUS 管理ビューによって戻される情報

列名	データ・タイプ	説明
NAME	VARCHAR(128)	タスクの名前。
TASKID	INTEGER	タスク ID。
STATUS	VARCHAR(10)	タスクの状況。有効な値は以下のとおりです。 <ul style="list-style-type: none"><li>• RUNNING - タスクは現在実行中です。</li><li>• COMPLETED - タスクは実行を完了しました。</li><li>• NOTRUN - エラーが原因で、スケジューラーはタスクのプロシーチャーを呼び出すことができません。</li><li>• UNKNOWN - タスクの実行が開始されましたが、予期しない状態が原因で、スケジューラーはタスクの結果を記録できませんでした。これは、タスクの実行中にシステムが異常終了したり、電源障害が発生したりした場合に生じる可能性があります。</li></ul>
INVOCATION	INTEGER	現在の呼び出しカウント。
BEGIN_TIME	TIMESTAMP	タスクが開始された時刻。 <sup>1</sup>  STATUS が RUNNING、COMPLETED、または UNKNOWN である場合、この値はタスクの実行が開始された時刻を示します。  STATUS が NOTRUN である場合、これはタスクが開始されるはずだった時刻を示します。
END_TIME	TIMESTAMP	タスクの実行が完了した時刻。 <sup>1</sup>  STATUS が RUNNING である場合、この値は NULL になります。  STATUS が UNKNOWN である場合、この値は、タスク・スケジューラーがタスクはもはや実行されていないと判断し、状況表を更新した時刻になります。
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID) モニター・エレメント
SQLCODE	INTEGER	STATUS が COMPLETED である場合、この値は、プロシーチャーの呼び出しによって戻される SQLCODE を示します。  STATUS が NOTRUN である場合、この値は、タスクを実行できない原因となった SQLCODE のエラーを示します。  状況が RUNNING または UNKNOWN である場合、この値は NULL になります。

表 79. ADMIN\_TASK\_STATUS 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明
SQLSTATE	CHAR(5)	STATUS が COMPLETED である場合、この値は、プロシーチャーの呼び出しによって戻される SQLSTATE を示します。  STATUS が NOTRUN である場合、この値は、タスクを実行できない原因となった SQLSTATE のエラーを示します。  状況が RUNNING または UNKNOWN である場合、この値は NULL になります。
SQLERRMC	VARCHAR(70) FOR BIT DATA	SQLCA の SQLERRMC フィールドに現れるとおり、X'FF' で区切られた 1 つ以上のトークンが入ります。これらのトークンは、エラー条件の説明の中の変数を置き換えます。  STATUS が COMPLETED である場合、この値は、プロシーチャーの呼び出しによって戻される SQLERRMC を示します。  STATUS が NOTRUN である場合、この値は、タスクを実行できない原因となった SQLERRMC のエラーを示します。  状況が RUNNING または UNKNOWN である場合、この値は NULL になります。
RC	INTEGER	STATUS が COMPLETED である場合、プロシーチャーに戻りコードが含まれていれば、これにはプロシーチャーの呼び出しからの戻りコードが含まれます。それ以外の場合、これは NULL になります。

- <sup>1</sup> BEGIN\_TIME および END\_TIME はデータベース・サーバーの時間帯に基づいています。夏時間調整時間 (DST) の調整はユーザーの責任で行う必要があります。

### ADMIN\_TASK\_UPDATE プロシーチャー - 既存のタスクの更新

ADMIN\_TASK\_UPDATE プロシーチャーは、管理タスク、つまりプロシーチャーの内部にカプセル化できる処理を更新します。

#### 構文

```
▶▶ ADMIN_TASK_UPDATE ( ( name , begin_timestamp , end_timestamp ,
▶ max_invocations , schedule , options , remarks ) ) ▶▶
```

スキーマは SYSPROC です。

#### プロシーチャー・パラメーター

##### name

既存のタスクの名前を指定する、タイプ VARCHAR (128) の入力引数。この引数は NULL にできません。

##### begin\_timestamp

タスクの実行を開始できる最も早い時刻を指定する、タイプ TIMESTAMP の入力引数。この引数の値は、過去の日付にしたり、end\_timestamp より後の日付にしたりすることはできません。



タスク実行が開始されるタイミングは、このパラメーターと *schedule* パラメーターが定義されている方法によって異なります。

- *begin\_timestamp* 引数が NULL 以外の場合:
  - *schedule* 引数が NULL の場合、タスク実行は *begin\_timestamp* に開始されます。
  - *schedule* 引数が NULL 以外の場合、タスク実行は、スケジュールされている次の時刻または *begin\_timestamp* よりも後に開始されます。
- *begin\_timestamp* 引数が NULL の場合:
  - *schedule* 引数が NULL の場合、タスク実行は即時に開始されます。
  - *schedule* 引数が NULL 以外の場合、タスク実行は、スケジュールされている次の時刻に開始されます。

#### *end\_timestamp*

タスクの実行を開始できる最後の時刻を指定する、タイプ `TIMESTAMP` の入力引数。この引数の値は、過去の日付にしたり、*begin\_timestamp* より前の日付にしたりすることはできません。引数が NULL の場合、タスクは、スケジュールされたとおりに無期限に実行されます。

実行中のタスクは、その *end\_timestamp* で中断されることはありません。

#### *max\_invocations*

タスクで許容される最大実行数を指定する、タイプ `INTEGER` の入力引数。引数が NULL の場合、タスクを実行できる回数に制限はありません。引数が 0 の場合、タスクは実行されません。

*schedule* が NULL 以外の場合、この値がスケジュールに適用されます。

*end\_timestamp* と *max\_invocations* の両方が指定されている場合、*end\_timestamp* が優先されます。つまり、*end\_timestamp* タイム・スタンプに達すると、タスクの回数とその時点で *max\_invocations* の値に達していなくても、そのタスクが再び実行されることはありません。

#### *schedule*

特定の時間にタスクが実行されるようにスケジュールを指定する、タイプ `VARCHAR(1024)` の入力引数。引数が NULL の場合、タスクは、特定の時間にスケジュールされません。

*schedule* ストリングは、UNIX `cron` 形式を使用して指定する必要があります。

複数のスケジュールはサポートされていません。

#### *options*

タイプ `VARCHAR(512)` の入力引数。この引数は NULL でなければなりません。

#### *remarks*

タスクの説明を指定する、タイプ `VARCHAR(254)` の入力引数。これは、NULL に設定できるオプションの引数です。

## 許可

以下のいずれかの権限が必要です。

- `ADMIN_TASK_UPDATE` プロシージャに対する `EXECUTE` 特権

- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

ステートメント許可 ID を使用してプロシージャを実行することが可能な場合でも、現行のセッション許可 ID がタスクの作成時に記録されたセッション許可 ID と一致しない限り、タスクを更新することはできません。既存のタスクを更新できるのは、SYSADM、SYSCTRL、SYSMAINT、または DBADM の各権限を持ったユーザーです。別のユーザーによって追加されたタスクを更新しようとする、SQL0551N が戻されます。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 使用上の注意

無効な引数がプロシージャに渡されると、SQL0171N が戻されます。メッセージのトークンは、無効な引数およびプロシージャの名前を示します。

タスクに対する変更は、作業単位がコミットされて、スケジューラーが更新されたタスク定義をフェッチするまでは有効になりません。作業単位を非コミットのままにしておくと、既存のタスクの実行が遅延したり、実行されなかったりする場合があります。

スケジューラーは、更新されたタスクがないかどうか 5 分おきに調べます。想定どおりにタスクが実行されるようにするには、*begin\_timestamp*、*end\_timestamp*、および *schedule* パラメーターで定義される最も早い開始時刻は、作業単位がコミットされてから少なくとも 5 分後でなければなりません。

タスクがスケジューラーによって実行できるようにするには、データベースはすべてのデータベース・パーティションでアクティブである必要があります。

*begin\_timestamp*、*end\_timestamp*、および *schedule* はデータベース・サーバーの時間帯に基づいています。夏時間調整 (DST) の移行期間にタスクをスケジュールする場合には、特に注意が必要です。タスクが午前 2 時 1 分に実行されるようスケジュールされている場合に、それが時間を早めるときに当たると、時間は午前 2 時から午前 3 時までスキップされるため、そのタスクは実行されません。DST の変更でデータベースがアクティブな状態で行われ、DST の変更時間中にスケジュールされたタスクがない場合、DST の変更後にスケジュールされている最初のタスクは実行されません。2 番目以降のタスクは正常に実行されます。例えば、午前 2 時から午前 3 時の間にスケジュールされているタスクはなく、午前 3 時 10 分にスケジュールされているタスクがあり、それが DST の変更後の最初のタスクである場合、このタスクは実行されません。これは、DST 変更のどれくらい後にタスクがスケジュールされているかとは関係がありません。例えば、DST より後の最初のタスクが DST の変更から 1 日後や 1 カ月後にスケジュールされていたとしても、そのタスクが DST 変更後の最初のタスクであり、データベースがその間ずっとアクティブな状態であった限り、そのタスクは実行されません。一方、時間が 1 時間戻される

ときには、午前 2 時から 3 時の間にスケジュールされているタスクは 2 回実行されることとなります。必要な動作が実行されるように、夏時間調整時間の調整はユーザーの責任で行う必要があります。

タスクが更新されると、タスクの内部呼び出しカウンターはリセットされます。一例として、*max\_invocations* 値が 10 の循環タスクについて考えてみましょう。タスクが 3 回実行されると、ADMIN\_TASK\_STATUS 出力には 3 つの対応する状況レコードが存在することとなります。エントリーの INVOCATION 値はそれぞれ、1、2、3 となります。次に、タスク作成者がこのタスクを更新するとします。この更新によって、内部呼び出しカウンターはリセットされます。元の状況レコードはそのまま残されます。時間の経過とともに、1、2、3 といった INVOCATION 値を持つ新しい状況レコードが作成されます。BEGIN\_TIME を使用して、元のタスク実行と更新されたタスク実行を区別することができます。

## 監査ルーチンおよびプロシージャ

### AUDIT\_ARCHIVE プロシージャおよび表関数 - 監査ログ・ファイルのアーカイブ

AUDIT\_ARCHIVE プロシージャおよび表関数はどちらも、接続中のデータベースの監査ログ・ファイルをアーカイブします。

#### 構文

```
▶▶—AUDIT_ARCHIVE—(—directory—, —member—)—————▶▶
```

スキーマは SYSPROC です。

構文は、プロシージャと表関数のどちらでも同じです。

#### プロシージャおよび表関数パラメーター

##### *directory*

アーカイブ対象監査ファイルが書き込まれるディレクトリーを指定する、タイプ VARCHAR(1024) の入力引数。ディレクトリーがサーバー上に存在しており、インスタンス所有者がそのディレクトリーにファイルを作成できなければなりません。引数が NULL または空ストリングである場合、デフォルト・ディレクトリーが使用されます。

##### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのメンバーの集合の場合は NULL または -2 を指定します。

#### 許可

AUDIT\_ARCHIVE プロシージャまたは表関数に対する Execute 特権。

#### デフォルトの PUBLIC 特権

なし

## 例

例 1: プロシージャを使用して、すべてのメンバーの監査ログをデフォルト・ディレクトリーにアーカイブします。

```
CALL SYSPROC.AUDIT_ARCHIVE(NULL, NULL)
```

例 2: 表関数を使用して、すべてのメンバーの監査ログをデフォルト・ディレクトリーにアーカイブします。

```
SELECT * FROM TABLE(SYSPROC.AUDIT_ARCHIVE('', -2)) AS T1
```

## 戻される情報

表 80. AUDIT\_ARCHIVE 表関数によって戻される情報

列名	データ・タイプ	説明
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
PATH	VARCHAR(1024)	アーカイブ対象ファイルのディレクトリー位置。
FILE	VARCHAR(1024)	アーカイブ対象ファイルの名前。
SQLCODE	INTEGER	ファイルのアーカイブ試行中に受信した SQLCODE。
SQLSTATE	VARCHAR(5)	ファイルのアーカイブ試行中に受信した SQLSTATE。SQLSTATE が NULL の場合、値はゼロです。
SQLERRMC	VARCHAR(70) FOR BIT DATA	ファイルのアーカイブ試行中に受信した sqlerrmc。SQLSTATE が NULL の場合、値はゼロです。
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## AUDIT\_DELIM\_EXTRACT - 区切り文字付きファイルへの抽出の実行

AUDIT\_DELIM\_EXTRACT ストアード・プロシージャは、接続中のデータベースのアーカイブ対象監査ファイル上で区切り文字付きファイルへの抽出を実行します。特に、指定されたマスク・パターンに一致するファイル名を持つアーカイブ済み監査ファイルへの抽出を実行します。

### 構文

```
▶▶—AUDIT_DELIM_EXTRACT—(—delimiter—,—target_directory—,—source_directory—,—  
▶—file_mask—,—event_options—)————▶▶
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *delimiter*

区切り文字付きファイルで使用される区切り文字を指定する、タイプ VARCHAR(1) のオプション入力引数。引数が NULL または空ストリングである場合、二重引用符が区切り文字として使用されます。

#### *target\_directory*

区切り文字付きファイルが保管されるディレクトリーを指定する、タイプ VARCHAR(1024) のオプション入力引数。引数が NULL または空ストリングである場合、*source\_directory* と同じディレクトリーが使用されます。

#### *source\_directory*

アーカイブ対象監査ログ・ファイルが保管されるディレクトリーを指定する、タイプ VARCHAR(1024) のオプション入力引数。引数が NULL または空ストリングである場合、監査デフォルトが使用されます。

#### *file\_mask*

どのファイルを抽出するかについてのマスクである、タイプ VARCHAR(1024) のオプション入力引数。引数が NULL または空ストリングである場合、ソース・ディレクトリーのすべての監査ログ・ファイルから抽出されます。

#### *event\_options*

どのイベントを抽出するかを定義するストリングを指定する、タイプ VARCHAR(1024) のオプション入力引数。これは、db2audit ユーティリティ어의同じストリングと一致します。引数が NULL または空ストリングである場合、すべてのイベントが抽出されます。

### 許可

AUDIT\_DELIM\_EXTRACT 関数に対する実行特権。

### デフォルトの PUBLIC 特権

なし

### 例

注: 監査ログ・ファイルには、その命名規則の一部としてタイム・スタンプが含まれます。

例 1: 2007 年 6 月 18 日にデフォルト・アーカイブ・ディレクトリーにアーカイブされたすべての監査ログ・ファイルについて区切り付きの抽出を実行します。この例では、実行イベントのみを抽出し、二重引用符 (") 区切り文字を使用し、さらに生成された抽出ファイル (<category>.del) を \$HOME/audit\_delim\_extract ディレクトリーで作成または追加します。

```
CALL SYSPROC.AUDIT_DELIM_EXTRACT(NULL, '$HOME/AUDIT_DELIM_EXTRACT', NULL, '%20070618%', 'CATEGORY EXECUTE')
```

### AUDIT\_LIST\_LOGS 表関数 - アーカイブ対象監査ログ・ファイルのリスト

AUDIT\_LIST\_LOGS 表関数は、指定されたディレクトリーにある、データベースのアーカイブ対象監査ログ・ファイルをリストします。

### 構文

▶▶—AUDIT\_LIST\_LOGS—(—directory—)————▶▶

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *directory*

アーカイブ対象監査ファイルが書き込まれるディレクトリーを指定する、タイプ VARCHAR(1024) のオプション入力引数。ディレクトリーがサーバー上に存在しており、インスタンス所有者がそのディレクトリーにファイルを作成できなければなりません。引数が NULL または空ストリングである場合、検索デフォルト・ディレクトリーが使用されます。

### 許可

AUDIT\_LIST\_LOGS 表関数に対する EXECUTE 特権。

### デフォルトの PUBLIC 特権

なし

### 例

例 1: デフォルトの監査アーカイブ・ディレクトリーにあるアーカイブ対象監査ログをすべてリストします。

```
SELECT * FROM TABLE(SYSPROC.AUDIT_LIST_LOGS('')) AS T1
```

注: これは、照会が実行されるデータベースのディレクトリーにあるログのみリストします。アーカイブ対象ファイルのフォーマットは db2audit.db.<dbname>.log.<timestamp> です。

### 戻される情報

表 81. AUDIT\_LIST\_LOGS について戻される情報

列名	データ・タイプ	説明
PATH	VARCHAR(1024)	アーカイブ対象ファイルのパス位置。
FILE	VARCHAR(1024)	アーカイブ対象ファイルのファイル名。
SIZE	BIGINT	アーカイブ対象ファイルのファイル・サイズ。

## 自動保守ルーチン

### AUTOMAINT\_GET\_POLICY プロシージャ - 自動保守ポリシーの取得

AUTOMAINT\_GET\_POLICY システム・ストアード・プロシージャは、データベースの自動保守構成を取得します。このプロシージャは 2 つのパラメーターを取ります。1 つはどの情報を収集するかに関する自動保守のタイプで、もう 1 つは構成情報を戻す BLOB へのポインターです。構成情報は XML フォーマットで戻されます。

### 構文

```
▶▶—AUTOMAINT_GET_POLICY—(—policy_type—,—policy—)————▶▶
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *policy\_type*

取得する自動保守ポリシーのタイプを指定する、タイプ VARCHAR(128) の入力引数。引数の値は、以下のいずれかになります。

#### *AUTO\_BACKUP*

自動バックアップ

#### *AUTO\_REORG*

表および索引の自動再編成

#### *AUTO\_RUNSTATS*

表の自動 RUNSTATS 操作

#### *MAINTENANCE\_WINDOW*

保守ウィンドウ

### *policy*

指定のポリシー・タイプの自動保守設定を XML フォーマットで指定する、タイプ BLOB(2M) の出力引数。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

以下の例は、組み込み SQL C ソース・コード内からの AUTOMAINT\_GET\_POLICY プロシージャの呼び出しを示しています。

- プロシージャ出力パラメーター用に BLOB 変数が宣言されます。
- プロシージャが呼び出されます。このとき、自動保守ポリシーのタイプとして自動バックアップを指定し、このプロシージャが現在接続中のデータベースのバックアップ・ポリシーを戻す出力パラメーターとして BLOB 変数を指定します。

```
EXEC SQL BEGIN DECLARE SECTION;  
SQL TYPE IS BLOB(2M) backupPolicy;  
EXEC SQL END DECLARE SECTION;
```

```
EXEC SQL CALL AUTOMAINT_GET_POLICY( 'AUTO_BACKUP', :backupPolicy );
```

## AUTOMAINT\_GET\_POLICYFILE プロシージャ - 自動保守ポリシーの取得

AUTOMAINT\_GET\_POLICYFILE システム・ストアード・プロシージャは、データベースの自動保守構成を取得します。このプロシージャは 2 つのパラメータを取ります。1 つはどの情報を収集するかに関する自動保守のタイプで、もう 1 つは構成情報を戻すファイルの名前です。構成情報は XML フォーマットで戻されます。

### 構文

```
▶▶AUTOMAINT_GET_POLICYFILE(—policy_type—,—policy_file_name—)————▶▶
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *policy\_type*

取得する自動保守ポリシーのタイプを指定する、タイプ VARCHAR(128) の入力引数。引数の値は、以下のいずれかになります。

#### *AUTO\_BACKUP*

自動バックアップ

#### *AUTO\_REORG*

表および索引の自動再編成

#### *AUTO\_RUNSTATS*

表の自動 RUNSTATS 操作

#### *MAINTENANCE\_WINDOW*

保守ウィンドウ

#### *policy\_file\_name*

DB2 インスタンス・ディレクトリーの tmp サブディレクトリーに作成されるファイルの名前を指定する、タイプ VARCHAR(2048) の入力引数。

注: ファイル名には接頭部として tmp への相対パスが付けられることがあります。その場合、ディレクトリーが存在しており、ファイルを作成/上書きする権限を持っていないければなりません。また、DB2 サーバーの正しいパス区切り記号を使用する必要があります。

例:

インスタンス・ディレクトリーが \$HOME/sql1lib で定義されている場合は、UNIX。 'policy.xml' という名前のポリシー・ファイルの場合、ファイル名は '\$HOME/sql1lib/tmp/policy.xml' です。

Windows の場合、インスタンス・ディレクトリー名は **DB2INSTPROF** レジストリー変数および **DB2INSTANCE** 環境変数の値から決定できます。 'policy.xml' という名前のポリシー・ファイルの場合、**db2set** が **DB2INSTPROF=C:¥DB2PROF** および **%DB2INSTANCE%=db2** を指定すると、ファイル名は **C:¥DB2PROF¥db2¥tmp¥policy.xml** です。



## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

バックアップ操作の現在の自動保守設定を取得するには、次のようにします。

```
call sysproc.automaint_get_policyfile( 'AUTO_BACKUP', 'AutoBackup.xml' )
```

これで、AutoBackup.xml という名前の XML ファイルが DB2 インスタンス・ディレクトリーの下に tmp サブディレクトリーに作成されます。

## AUTOMAINT\_SET\_POLICY プロシージャ - 自動保守ポリシーの構成

AUTOMAINT\_SET\_POLICY システム・ストアード・プロシージャを使用して、データベースの自動保守を構成できます。このプロシージャは 2 つのパラメーターを取ります。1 つは構成する自動保守のタイプで、もう 1 つは構成を指定する XML を含む BLOB です。

マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング (ITC) 表での自動再編成操作中に RECLAIM EXTENTS オプションを有効にするには、XML 入力ファイルで、ReorgOptions エlementに「reclaimExtentSizeForTables」属性をしきい値とともに指定する必要があります。

索引での自動再編成操作中に RECLAIM EXTENTS オプションを有効にするには、XML 入力ファイルで、ReorgOptions エlementに「reclaimExtentsSizeForIndexObjects」属性をしきい値とともに指定する必要があります。

**注:** 「reclaimExtentsSizeForIndexObjects」に指定されたしきい値は、索引オブジェクト・レベルで適用されます。非パーティション表の場合は、結合された表のすべての索引に値が適用されます。パーティション表の場合、値は非パーティション索引のそれぞれに別々に適用されるとともに、各データ・パーティションの索引に別々に適用されます。

表または索引スペースの再利用に指定するしきい値は、スペースを再利用するためのオンライン再編成が起動される前の表または索引の再利用可能スペースの最小サイズです (キロバイト単位)。このしきい値は 0 以上でなければなりません。例えば、しきい値として 1024 KB を指定した場合、再利用可能スペースが 1 MB 以上のオブジェクトに対してのみ、スペースを再利用するための自動再編成が考慮されます。

## 構文

▶▶AUTOMAINT\_SET\_POLICY(—policy\_type—,—policy—)————▶▶

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *policy\_type*

構成する自動保守ポリシーのタイプを指定する、タイプ VARCHAR(128) の入力引数。値は以下のいずれかです。

#### **AUTO\_BACKUP**

自動バックアップ

#### **AUTO\_REORG**

表および索引の自動再編成

#### **AUTO\_RUNSTATS**

表の自動 RUNSTATS 操作

#### **MAINTENANCE\_WINDOW**

保守ウィンドウ

#### *policy*

自動保守ポリシーを XML フォーマットで指定する、タイプ BLOB(2M) の入力引数。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

**例 1:** RUNSTATS 操作の現在の自動保守設定を設定するには、次のようにします。

```
CALL SYSPROC.AUTOMAINT_SET_POLICY
( 'AUTO_RUNSTATS',
  BLOB(1 <?xml version="1.0" encoding="UTF-8"?>
    <DB2AutoRunstatsPolicy
      xmlns="http://www.ibm.com/xmlns/prod/db2/autonomic/config">
      <RunstatsTableScope><FilterCondition/></RunstatsTableScope>
    </DB2AutoRunstatsPolicy>')
)
```

これにより、現在の自動統計収集構成が、XML 文書に含まれる新規構成と置換され、それは 2 番目のパラメーターとしてプロシージャに渡されます。

**例 2:** DB2 の自動再編成フィーチャーは、新しい「RECLAIM EXTENTS」オプションを使用して、マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング (ITC) 表を再編成することができます。このフィーチャーを使用可能にするには、以下に示すように AUTO\_REORG ポリシー内に「reclaimExtentSizeForTables」値を設定します。

```
CALL SYSPROC.AUTOMAINT_SET_POLICY
('AUTO_REORG',
 BLOB(' '<?xml version="1.0" encoding="UTF-8"?>
 <DB2AutoReorgPolicy
   xmlns="http://www.ibm.com/xmlns/prod/db2/autonomic/config">
 <ReorgOptions dictionaryOption="Keep" indexReorgMode="Online"
   useSystemTempTableSpace="false" reclaimExtentSizeForTables ="1024" >
 <ReorgTableScope>
 <FilterClause>TABSCHEMA NOT LIKE 'EMP%'</FilterClause>
 </ReorgTableScope>
 </DB2AutoReorgPolicy>')
```

サンプル XML 入力ファイルが SQLLIB/samples/automaintcfg ディレクトリーにあります。これをご使用の要件に合わせて変更し、この例で示されるように、XML の内容を BLOB() スカラー関数に渡すことができます。

## AUTOMAINT\_SET\_POLICYFILE プロシージャ - 自動保守ポリシーの構成

AUTOMAINT\_SET\_POLICYFILE システム・ストアード・プロシージャを使用して、データベースの自動保守を構成できます。このプロシージャは 2 つのパラメータを取ります。1 つは構成する自動保守のタイプで、もう 1 つは構成を指定する XML 文書の名前です。

このプロシージャは、SQL 成功または SQL エラー・コードを戻します。

### 構文

```
►►AUTOMAINT_SET_POLICYFILE(—policy_type—,—policy_file_name—)◄◄
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *policy\_type*

構成する自動保守ポリシーのタイプを指定する、タイプ VARCHAR(128) の入力引数。引数の値は、以下のいずれかになります。

#### *AUTO\_BACKUP*

自動バックアップ

#### *AUTO\_REORG*

表および索引の自動再編成

#### *AUTO\_RUNSTATS*

表の自動 RUNSTATS 操作

#### *MAINTENANCE\_WINDOW*

保守ウィンドウ

### *policy\_file\_name*

DB2 インスタンス・ディレクトリーの tmp サブディレクトリーで使用可能なファイルの名前を指定する、タイプ VARCHAR(2048) の入力引数。

注: ファイル名が相対パスで指定される場合、DB2 サーバーの正しいパス区切り記号を使用する必要があり、ディレクトリーおよびファイルが読み取り権限付きで存在しなければなりません。

例:

インスタンス・ディレクトリーが \$HOME/sql1lib で定義されている場合は、UNIX。 'automaint/policy.xml' という名前のポリシー・ファイルの場合、ファイル名は '\$HOME/sql1lib/tmp/automaint/policy.xml' です。

Windows の場合、インスタンス・ディレクトリー名は **DB2INSTPROF** レジストリー変数および **DB2INSTANCE** 環境変数の値から決定できます。

'automaint%policy.xml' という名前のポリシー・ファイルの場合、**db2set** が **DB2INSTPROF=C:%DB2PROF** および **%DB2INSTANCE%=db2** を指定すると、ファイル名は **C:%DB2PROF%db2%tmp%automaint%policy.xml** です。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

自動バックアップの現在の自動保守設定を変更するには、次のようにします。

```
call sysproc.automaint_set_policyfile( 'AUTO_BACKUP', 'AutoBackup.xml' )
```

これにより、現在の自動バックアップ構成設定が、DB2 インスタンス・ディレクトリーの下の tmp ディレクトリーにある AutoBackup.xml ファイルに含まれる新規構成と置換されます。

サンプル XML 入力ファイルが **SQLLIB/samples/automaintcfg** ディレクトリーにあります。これをポリシー xml ファイルを作成するための参照として使用することができます。

## 共通 SQL API プロシージャ

共通 SQL API は、複数の IBM データ・サーバーで共通に使用できる、共通シグニチャーの集合およびシグニチャー安定型のストアード・プロシージャを提供します。これらのストアード・プロシージャを使用して、構成パラメーターの取得および設定、システム情報の取得などのさまざまな管理機能を実行するアプリケーションを作成できます。

ストアード・プロシージャは、あらゆるデータ・サーバーの上で共通に使用できる、構文的に同一の XML パラメーターとエラー処理を提供することにより、データ・サーバーのバージョン独立性を保ちます。シグニチャーの安定性と共通性は、共通 DTD を使用する簡単な XML 文書をパラメーターとして使用することにより実現されます。バージョン、プラットフォーム、およびテクノロジーの相違は、階層プロパティ・リスト内の異なるキーと値のペアで表現されます。

## 共通入出力パラメーター

共通 SQL API ストアード・プロシージャは、入出力パラメーターのセットを共有します。

次の表では、これらのパラメーターの要旨を示しています。さらに詳しい情報は、共通 SQL API ストアード・プロシージャについてのリファレンス・トピックを参照してください。

表 82. 共通 SQL API の共有入出力パラメーター

パラメーター	説明
<i>major_version</i>	プロシージャでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする文書タイプのメジャー・バージョンを示します。
<i>minor_version</i>	<p>プロシージャでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする文書タイプのマイナー・バージョンを示します。</p> <p>パラメーター <i>major_version</i> と <i>minor_version</i> は、呼び出し元が誤ったバージョンの XML 入力文書を使用しないように同時に使用されます。プロシージャは、指定した <i>major_version</i> および <i>minor_version</i> のすべての XML 文書を処理するか、またはバージョンが無効な場合はエラーを戻します。この設計では、新しい文書タイプのバージョンの追加を既存のアプリケーションに影響を与えることなく行えるため、将来のリリースにおける拡張性がサポートされます。</p>
<i>requested_locale</i>	<i>xml_output</i> および <i>xml_message</i> パラメーターで戻された XML 文書の変換された内容を戻すために使用するロケールを指定します。値のみが変換され、キー名は変換されません。
<i>xml_input</i>	プロシージャの入力値が含まれる XML 入力文書を指定します。
<i>xml_filter</i>	出力パラメーター文書から単一の値を検索するのに使用する有効な XPath 照会ストリングを指定します。

表 82. 共通 SQL API の共有入出力パラメーター (続き)

パラメーター	説明
<i>xml_output</i>	UTF-8 でエンコードされた完全な XML 出力文書を戻します。呼び出されるプロシージャに応じて、この文書には構成パラメーターとその値、システム情報、またはメッセージ・テキストが含まれる場合があります。プロシージャが <code>コンプリート・モード</code> で動作する場合、このパラメーターによって戻される XML 文書は、変更することができ、 <i>xml_input</i> パラメーターとしてプロシージャに再び渡すことができます。このアプローチにより、有効な XML 入力文書を作成するためのプログラマ的な方法が提供されます。
<i>xml_message</i>	SQL 警告状態についての詳細情報を提供する Data Server Message タイプの完全な XML 出力文書を UTF-8 で戻します。

## XML 文書のバージョン管理

将来のリリースにおける拡張性をサポートするために、共通 SQL API ストアード・プロシージャは、バージョン情報を含む XML 出力文書を戻します。

XML 出力文書の構造が変更される場合は常に (例えば、エレメントが追加または除去される場合)、バージョン・レベルが増分されます。したがって、プロシージャは複数バージョンの XML 出力文書をサポートする場合があります。

XML 文書のバージョン情報は、文書タイプのメジャー・バージョンおよび文書タイプのマイナー・バージョンに関するキーと値のペアで表現されます。例えば、XML 出力文書はディクショナリーのエレメントに次のキーと値を定義する場合があります。

```
<key>Document Type Name</key><string>Data Server Configuration Output</string>
<key>Document Type Major Version</key><integer>2</integer>
<key>Document Type Minor Version</key><integer>0</integer>
```

プロシージャを呼び出す際に、戻したい XML 文書のメジャー・バージョンとマイナー・バージョンを指定します。XML 出力文書の内容は、指定する値によって異なります。

例えば、GET\_CONFIG プロシージャは特定のインスタンスに対して設定されたデータベースおよびデータベース・マネージャー構成パラメーターを取得します。

*major\_version* が 2 に、*minor\_version* が 0 に設定された状態でこのプロシージャが呼び出されると、カテゴリ別にグループ化された構成パラメーターを含む XML 文書が戻されます。しかし、*major\_version* が 1 に、*minor\_version* が 0 に設定された状態で同じプロシージャが呼び出されると、構成パラメーターを含む XML 文書が戻されますが、これらのパラメーターはカテゴリ別にグループ化されてはいません。

同様に、GET\_MESSAGE プロシージャは指定した SQLCODE のメッセージ・テキストと SQLSTATE を取得します。*major\_version* を 2 に、*minor\_version* を 0 に設定してこのプロシージャが呼び出されると、対応する SQLCODE の短いテキス

ト・メッセージ、長いテキスト・メッセージ、および SQLSTATE を含む XML 文書が戻されます。ただし、*major\_version* を 1 に、*minor\_version* を 0 に設定して同じプロシージャが呼び出されると、短いテキスト・メッセージと SQLSTATE のみを含む XML 文書が戻されます。長いテキスト・メッセージは文書のバージョン 1 では使用できません。

プロシージャに対してサポートされる文書の最も高いバージョンを判別するには、*major\_version*、*minor\_version*、およびその他すべての入力パラメーターに NULL を指定します。プロシージャは、*major\_version* および *minor\_version* 出力パラメーターの値として、サポートされる文書の最も高いバージョンを戻し、*xml\_output* および *xml\_message* 出力パラメーターを NULL に設定します。

*major\_version* および *minor\_version* に NULL 以外の値を指定する場合、サポートされる文書のバージョンを指定しなければなりません。そうしない場合、サポートされていないバージョンがプロシージャによって検出されたことを示すエラー (-20457) がプロシージャによって起こされます。

XML 入力文書には、オプションで文書タイプのメジャー・バージョンとマイナー・バージョンの値を組み込むことができます。これらの値が XML 入力文書に指定された場合、プロシージャ呼び出しで *major\_version* および *minor\_version* に渡された値は、XML 文書に指定されている値と完全に一致していなければなりません。そうでない場合にプロシージャはエラー (+20458) を起こします。この動作により、サポートされていない XML 入力文書のバージョンを呼び出し元が指定することが防止されます。

## XML 入力文書

共通 SQL API ストアード・プロシージャへの入力として渡される XML 文書は、共通の DTD に基づく簡単な XML 形式を共有します。

XML 入力文書は、すべてのストアード・プロシージャに共通する項目のセット、およびそれぞれのストアード・プロシージャに固有の項目のセットから構成されています。XML 入力文書の一般的な構造は以下のとおりです。

```
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key><string>Data Server Message Input</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Document Locale</key><string>en_US</string>
  <key>Complete</key><false/>
  <dict>
    <!-- Document type specific data appears here. -->
  </dict>
</dict>
</plist>
```

**重要:** XML 入力文書は UTF-8 でエンコードされている必要があり、含めることができるのは英語の文字のみです。

**有効な XML 入力文書を戻すためのコンプリート・モード:**

コンプリート・モード を使用して、入力を受け入れる共通 SQL API ストアード・プロシージャに対して有効な XML 文書を作成できます。文書をカスタマイズしてプロシージャに戻すことができます。

コンプリート・モードでプロシージャを実行するには、入力 XML 文書の Complete キーに true を指定し、次の最小限の内容を渡します。

```
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Complete</key><true/>
</dict>
</plist>
```

必須でない XML エLEMENT はいずれも無視され、出力文書で戻されません。

プロシージャを実行したときに、完全な XML 入力文書がストアード・プロシージャの *xml\_output* パラメーターで戻されます。戻された XML 文書には、指定可能なすべての必須パラメーターとオプション・パラメーターについての文書タイプおよびセクションが含まれます。また、戻された XML 文書には必須ではない他の項目 (表示名、ヒント、および文書ロケールなど) が含まれますが、通常、これらはクライアント・アプリケーションで文書をレンダリングする場合に必要ななりません。

XML 文書をレンダリングし、プラットフォームに依存しない方式でそれを変更してから、同じストアード・プロシージャを実行し、変更した XML 文書を入力として渡すことができます。

## XML 出力文書

共通 SQL API ストアード・プロシージャから出力として戻される XML 文書には共通の項目のセットがあります。

*xml\_output* パラメーターで戻される XML 文書には、少なくとも次の必須のキーと値のペアが含まれます。

```
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key>
  <string>Data Server Configuration Output</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Data Server Product Name</key><string>DSN</string>
  <key>Data Server Product Version</key><string>8.1.0.356</string>
  <key>Data Server Major Version</key><integer>8</integer>
  <key>Data Server Minor Version</key><integer>1</integer>
  <key>Data Server Platform</key><string>z/OS</string>
  <key>Document Locale</key><string>en_US</string>

  <!-- Document type specific data appears here. -->
</dict>
</plist>
```

XML 出力文書内の項目はネストされたディクショナリーを使用してグループ化されることがあります。XML 出力文書内の各項目は、単一の情報について記述します。項目は、値、表示名、およびヒントから構成されます。オプションで、表示単位が指定される場合があります。表示名、ヒント、および表示単位は言語によって異なる



り、*requested\_locale* パラメーターの値に指定された言語 (または要求したロケールがサポートされていない場合はデフォルト) に変換されます。通常、項目には次に示されている例のような構造があります。

```
<key>Real Storage Size</key>
<dict>
  <key>Display Name</key><string>Real Storage Size</string>
  <key>Value</key><integer>2048</integer>
  <key>Display Unit</key><string>MB</string>
  <key>Hint</key><string>Size of actual real storage online</string>
</dict>
```

IBM データ・サーバーには、すべてのデータ・サーバーに適用されるキーワード、およびデータ・サーバーに固有のキーワードを含む共通のパラメーター文書があります。データ・サーバーが新たなキーワードを追加したり除去したりする場合は常に、(すべてのデータ・サーバーの) バージョン番号が増分されます。変更内容により、メジャー・バージョン番号が増加してマイナー・バージョン番号が 0 (ゼロ) に設定されるか、またはマイナー・バージョン番号のみが増分されることがあります。

XML 出力文書は UTF-8 で生成され、英語の文字のみが含まれます。

#### 出力のフィルタリングのための XPath 式:

XPath 式を使用して、共通 SQL API ストアード・プロシージャで戻される XML 出力をフィルタリングできます。

出力をフィルタリングするには、有効な XPath 照会ストリングをプロシージャの *xml\_filter* パラメーターに指定します。指定する XPath 式には次の制約事項が適用されます。

- XPath 式は単独の値を参照しなければなりません。
- XPath 式は常にルート・ノードからの絶対パスでなければなりません。例えば、*/*、*nodename*、*..*、および *..* のようなパス式は使用できますが、*//* および *@* のようなパス式は使用できません。
- 使用できる述部は [*path*='value'] および [*n*] のみです。
- 使用できる軸は *following-sibling* のみです。
- XPath 式の末尾は次のいずれかでなければならず、必要な場合には述部 [1] を追加しなければなりません。 *following-sibling::string*、*following-sibling::data*、*following-sibling::date*、*following-sibling::real*、または *following-sibling::integer*
- 軸が XPath 式の末尾にない場合、軸の後に *::dict*、*::string*、*::data*、*::date*、*::real*、または *::integer* を付け、必要な場合には述部 [1] を追加しなければなりません。
- サポートされる XPath 演算子は = のみです。
- XPath 式には、関数、名前空間、処理命令、またはコメントを含めることができません。

**ヒント:** ストアード・プロシージャが *コンプリート・モード* で動作する場合、フィルタリングを行わないでください。これを行うと、SQLCODE (+20458) が発生します。

*xml\_output* パラメーターで戻された XML 文書の処理を細かく制御するために、DB2 pureXML<sup>®</sup> で使用可能な XMLPARSE 関数を使用できます。

## 例

次の XPath 式は、XML 出力文書から Data Server Product Version キーの値を選択します。

```
/plist/dict/key[.='Data Server Product Version']following-sibling::string[1]
```

プロシージャは、*xml\_output* パラメーターでストリング 8.1.0.356 を戻します。したがって、プロシージャ呼び出しは、XML 文書ではなく単独の値を戻します。

## XML メッセージ文書

共通 SQL API ストアード・プロシージャで内部処理エラーまたは無効なパラメーターが検出された場合、データ・サーバーは SQLCODE と対応する SQL メッセージを呼び出し元に戻します。

この場合、プロシージャは、*xml\_message* パラメーターで警告状態についてのより詳細な情報を含む XML メッセージ文書に戻します。

XML メッセージ文書の一般的な構造は次のとおりです。

```
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key><string>Data Server Message</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Data Server Product Name</key><string>QDB2/AIX64</string>
  <key>Data Server Product Version</key><string>9.5.0.3</string>
  <key>Data Server Major Version</key><integer>9</integer>
  <key>Data Server Minor Version</key><integer>5</integer>
  <key>Data Server Platform</key><string>AIX 64BIT</string>
  <key>Document Locale</key><string>en_US</string>
  <key>Short Message Text</key>
  <dict>
    <key>Value</key><string>
      <!-- Additional description of warning appears here. --></string>
    <key>Hint</key><string></string>
  </dict>
</dict>
</plist>
```

XML メッセージ文書は UTF-8 で生成され、英語の文字のみが含まれます。

## 例

次の例では、GET\_MESSAGE プロシージャへの呼び出しにより SQL 警告が発生しています。

```
db2 "CALL SYSPROC.GET_MESSAGE(NULL,NULL,'en_US',NULL,NULL,?,?)"
```

```
SQL20458W The procedure "SYSPROC.GET_MESSAGE" has encountered
an internal parameter processing error in parameter "3".
The value for parameter "7" contains further information about
the error. SQLSTATE=01H54
```

パラメーター 7 (*xml\_message*) で戻された XML 文書には次の内容が含まれます。

```

<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key><string>Data Server Message</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Data Server Product Name</key><string>QDB2/AIX64</string>
  <key>Data Server Product Version</key><string>9.5.0.3</string>
  <key>Data Server Major Version</key><integer>9</integer>
  <key>Data Server Minor Version</key><integer>5</integer>
  <key>Data Server Platform</key><string>AIX 64BIT</string>
  <key>Document Locale</key><string>en_US</string>
  <key>Short Message Text</key>
  <dict>
    <key>Value</key><string>If parameters 1 and 2 are set to NULL, all
      other input parameters must be set to NULL as well, but the value
      of parameter "3" is not NULL. </string>
    <key>Hint</key><string></string>
  </dict>
</dict>
</plist>

```

Short Message Text キーの値は、警告についての追加の情報を提供します。

## CANCEL\_WORK プロシージャ - 作業の取り消し

CANCEL\_WORK ストアード・プロシージャは、特定のアクティビティー (例えば SQL ステートメントなど)、または接続されたアプリケーションに対するすべてのアクティビティーを取り消します。

特定のアクティビティーを取り消すには、取り消したいアクティビティーの、アプリケーション・ハンドル、作業単位 ID、およびアクティビティー ID を渡します。接続したアプリケーションに対するすべてのアクティビティーを取り消す場合は、アプリケーション・ハンドルを渡します。取り消したアクティビティーに関連付けられているすべての変更は、ロールバックされます。

### 構文

```

▶▶—CANCEL_WORK—(—major_version—,—minor_version—,—requested_locale—,—————▶
▶—xml_input—,—xml_filter—,—xml_output—,—xml_message—)—————▶▶

```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *major\_version*

メジャー文書バージョンを指す、タイプ INTEGER の入出力引数。入力において、この引数は、プロシージャでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする、メジャー文書バージョンを示します (*xml\_input*、*xml\_output*、および *xml\_message* のパラメーターの説明を参照してください)。プロシージャは、指定したバージョンのすべての XML 文書を処理するか、またはバージョンが無効な場合はエラー (+20458) を戻します。出力において、このパラメーターは、プロシージャによってサポートされている最も高いメジャー文書バージョンを指定します。サポートされる文書の最も高いバージョンを判別するには、この入力パラメーターおよびその他すべての必須パラメーターに NULL を指定します。

## サポートされているバージョン: 1

### *minor\_version*

マイナー文書バージョンを指す、タイプ INTEGER の入出力引数。入力において、この引数は、プロシージャーでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする、マイナー文書バージョンを示します (*xml\_input*、*xml\_output*、および *xml\_message* のパラメーターの説明を参照してください)。プロシージャーは、指定したバージョンのすべての XML 文書を処理するか、またはバージョンが無効な場合はエラーを戻します。出力において、このパラメーターは、サポートされている最も高いメジャー・バージョンでサポートされている、最も高いマイナー文書バージョンを指します。サポートされる文書の最も高いバージョンを判別するには、この入力パラメーターおよびその他すべての必須パラメーターに NULL を指定します。

## サポートされているバージョン: 0

### *requested\_locale*

ロケールを指定する、タイプ VARCHAR(33) の入力引数。指定した言語がサーバーでサポートされている場合は、変換された内容が *xml\_output* および *xml\_message* パラメーターに戻されます。サポートされていない場合は、内容はデフォルトの言語に戻されます。ロケールから使用されるのは、言語 (および場合によっては地域情報) のみです。ロケールは数字のフォーマット設定に使用されたり、文書エンコードに影響を与えたりすることはありません。例えば、キー名と値は変換されません。XML 出力および XML メッセージ文書で変換される部分は、各項目のヒントのテキスト、表示名、および表示単位のみです。呼び出し元は常に、要求された言語と XML 出力文書で使用されている言語を比較する必要があります (XML 出力文書の文書ロケール項目を参照してください)。

現時点でサポートされている *requested\_locale* の値は en\_US のみです。

### *xml\_input*

プロシージャーの入力値が含まれる XML 入力文書 (UTF-8 でエンコードされている) を指定する、タイプ BLOB(32MB) の入力引数。

このプロシージャーでは、XML 入力文書でアプリケーション・ハンドルを指定する必要があります。特定のアクティビティーを取り消す場合は、XML 入力文書で、作業単位 ID およびアクティビティー ID を識別するオプション・パラメーターも指定する必要があります。このストアード・プロシージャーの完全な XML 入力文書は、次の文書のようなものになります。

```
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key><string>Data Server Cancel Work Input</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Required Parameters</key>
  <dict>
    <key>Application Handle</key>
    <dict>
      <key>Display name</key><string>Application Handle</string>
      <key>Value</key><integer>10</integer>
      <key>Hint</key>
      <string>
        Numeric value equivalent to the application handle to be cancelled
      </string>
    </dict>
  </dict>
</dict>
```

```

</dict>
<key>Optional Parameters</key>
<dict>
  <key>Unit Of Work Id</key>
  <dict>
    <key>Display Name</key><string>Unit Of Work Id</string>
    <key>Value</key><integer>20</integer>
    <key>Hint</key>
    <string>
      Numeric value that specifies the unit of work id of the activity
      that is to be cancelled
    </string>
  </dict>
  <key>Activity Id</key>
  <dict>
    <key>Display Name</key><string>Activity Id</string>
    <key>Value</key><integer>10</integer>
    <key>Hint</key>
    <string>
      Numeric value equivalent to the activity id to be cancelled
    </string>
  </dict>
</dict>
</plist>

```

ストアード・プロシージャが実行されているアプリケーションのアプリケーション・ハンドルを指定した場合、プロシージャは警告 (SQL20458) を戻します。

#### *xml\_filter*

有効な XPath 照会ストリングを指定する、タイプ BLOB(4K) の入力引数。XML 出力文書から単一値を検索する場合、フィルターを使用します。詳しくは、XPath フィルター操作について記述しているトピックを参照してください。

次の例では、XML 出力文書から、データ・サーバー製品バージョンの値を選択しています (/plist/dict/key[.='Data Server Product Version']/following-sibling::string)。キーの後に兄弟が指定されていないと、エラーが戻されません。

#### *xml\_output*

UTF-8 の完全な XML 出力文書を戻す、タイプ BLOB(32MB) の出力パラメータを戻します。フィルターが指定されている場合、このパラメータはストリング値を戻します。ストアード・プロシージャが完全な出力文書を戻すことができない場合 (例えば、処理エラーが発生して SQL 警告やエラーが出される場合など)、このパラメータは NULL に設定されます。

XML 出力は、以下のように *major\_version* および *minor\_version* で指定した値によって決まります。

メジャー・バージョン	マイナー・バージョン	<i>xml_output</i> の値
NULL	NULL	NULL
1	0	プロシージャが取り消そうとしているアクティビティの状況。

プロシージャが コンプリート・モード で動作する場合、このパラメータによって戻される XML 文書は、変更して *xml\_input* パラメータとしてプロシ

ージャーに再び渡すことができます。このアプローチにより、有効な XML 入力文書を作成するためのプログラマ的な方法が提供されます。詳しくは、コンプリート・モードについてのトピックを参照してください。

#### *xml\_message*

SQL 警告状態についての詳細情報を提供する Data Server Message タイプの完全な XML 出力文書を UTF-8 で戻す、タイプ BLOB(64K) の出力パラメーター。プロシージャの呼び出しにより SQL 警告が出され、XML メッセージ出力文書に追加情報が戻されることを警告メッセージが示す場合に、この文書が戻されます。追加情報が戻されることを警告メッセージが示していない場合は、このパラメーターは NULL に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

例 1: プロシージャの、サポートされる最も高いバージョンを戻します。

```
db2 "call sysproc.cancel_work(null,null,null,null,null,?,?)"
```

以下はこの照会の出力例です。

```
Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 1

Parameter Name : MINOR_VERSION
Parameter Value : 0

Parameter Name : XML_OUTPUT
Parameter Value : -

Parameter Name : XML_MESSAGE
Parameter Value : -

Return Status = 0
```

例 2: 特定のアクティビティを取り消します。

```
db2 "call sysproc.cancel_work(1,0,'en_US',blob(
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key><string>Data Server Cancel Work Input</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Required Parameters</key>
```

```

<dict>
  <key>Application Handle</key>
  <dict>
    <key>Display name</key><string>Application Handle</string>
    <key>Value</key><integer>1</integer>
    <key>Hint</key>
    <string>
      Numeric value equivalent to the application handle to be cancelled
    </string>
  </dict>
</dict>
<key>Optional Parameters</key>
<dict>
  <key>Unit Of Work Id</key>
  <dict>
    <key>Display Name</key><string>Unit Of Work Id</string>
    <key>Value</key><integer>2</integer>
    <key>Hint</key>
    <string>
      Numeric value that specifies the unit of work id of the activity
      that is to be cancelled
    </string>
  </dict>
  <key>Activity Id</key>
  <dict>
    <key>Display Name</key><string>Activity Id</string>
    <key>Value</key><integer>3</integer>
    <key>Hint</key>
    <string>
      Numeric value equivalent to the activity id to be cancelled
    </string>
  </dict>
</dict>
</dict>
</plist> ) ,null,?,?)"

```

以下はこの照会の出力例です。

```

Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 1

Parameter Name : MINOR_VERSION
Parameter Value : 0

Parameter Name : XML_OUTPUT
Parameter Value : x'3C3F78...'

Parameter Name : XML_MESSAGE
Parameter Value : -

Return Status = 0

```

CANCEL\_WORK プロシージャでアクティビティーを取り消せる場合、XML 出力文書には次の内容が含まれます。

```

<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict><key>Document Type Name</key><string>Data Server Cancel Work Output</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Data Server Product Name</key><string>QDB2/AIX64</string>
  <key>Data Server Product Version</key><string>9.7.0.0</string>
  <key>Data Server Major Version</key><integer>9</integer>
  <key>Data Server Minor Version</key><integer>7</integer>
  <key>Data Server Platform</key><string>AIX 64BIT</string>

```

```

<key>Document Locale</key><string>en_US</string>
<key>Successful Cancel Work Message</key>
<dict>
  <key>Display Name</key><string>Successful Cancel Work Message</string>
  <key>Value</key><string>The activity has been cancelled successfully</string>
  <key>Hint</key><string></string>
</dict>
</dict>
</plist>

```

例 2: アプリケーションを取り消します。

```

db2 "call sysproc.cancel_work(1,0,'en_US,blob(
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key><string>Data Server Cancel Work Input</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Required Parameters</key>
  <dict>
    <key>Application Handle</key>
    <dict>
      <key>Display name</key><string>Application Handle</string>
      <key>Value</key><integer>101</integer>
      <key>Hint</key>
      <string>
        Numeric value equivalent to the application handle to be cancelled
      </string>
    </dict>
  </dict>
</dict>
</plist> ),null,?,?)"

```

以下はこの照会の出力例です。

```

Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 1

Parameter Name : MINOR_VERSION
Parameter Value : 0

Parameter Name : XML_OUTPUT
Parameter Value : x'3C3F78...'

Parameter Name : XML_MESSAGE
Parameter Value : -

Return Status = 0

```

CANCEL\_WORK プロシージャでアプリケーションを取り消せる場合、XML 出力文書には次の内容が含まれます。

```

<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key><string>Data Server Cancel Work Output</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Data Server Product Name</key><string>QDB2/AIX64</string>
  <key>Data Server Product Version</key><string>9.7.0.0</string>
  <key>Data Server Major Version</key><integer>9</integer>
  <key>Data Server Minor Version</key><integer>7</integer>
  <key>Data Server Platform</key><string>AIX 64BIT</string>
  <key>Document Locale</key><string>en_US</string>

```



```

    <key>Successful Cancel Work Message</key>
    <dict>
      <key>Display Name</key><string>Successful Cancel Work Message</string>
      <key>Value</key>
      <string>The application has been cancelled successfully</string>
      <key>Hint</key><string></string>
    </dict>
  </dict>
</plist>

```

例 3: フィルターを指定して、成功した取り消し処理メッセージの値を戻します。

```

db2 "call sysproc.cancel_work(1,0,'en_US,blob(
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key><string>Data Server Cancel Work Input</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Required Parameters</key>
  <dict>
    <key>Application Handle</key>
    <dict>
      <key>Display name</key><string>Application Handle</string>
      <key>Value</key><integer>101</integer>
      <key>Hint</key>
      <string>
        Numeric value equivalent to the application handle to be cancelled
      </string>
    </dict>
  </dict>
</dict>
</plist> ),blob('/plist/dict/key[.="Successful Cancel Work Message"]
/following-sibling::dict[1]/key[.="Value"]
/following-sibling::string[1]'),?,?)"

```

以下はこの照会の出力例です。

```

Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 1

Parameter Name : MINOR_VERSION
Parameter Value : 0

Parameter Name : XML_OUTPUT
Parameter Value : x'3C3F78...'

Parameter Name : XML_MESSAGE
Parameter Value : -

Return Status = 0

```

*xml\_output* に次の値が戻されます。

```
"The application has been cancelled successfully"
```

## DESIGN\_ADVISOR - 設計アドバイザーの推奨値の取得

DESIGN\_ADVISOR プロシージャは、IBM DB2 10.1 サーバーから設計アドバイザーの推奨値を取得します。

## 構文

注: ご使用のインスタンスにバージョン 9.7 フィックスパック 5 より前で作成されたデータベースが含まれている場合、**db2updv97** コマンドを実行して、システム・カタログにこの新しいプロシージャを追加する必要があります。

```
►—DESIGN_ADVISOR—(—major_version—,—minor_version—,—requested_locale—,——————►  
►—xml_input—,—xml_filter—,—xml_output—,—xml_message—)—————►
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *major\_version*

メジャー文書バージョンを指す、タイプ INTEGER の入出力引数。入力において、この引数は、プロシージャでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする、メジャー文書バージョンを示します (*xml\_input*、*xml\_output*、および *xml\_message* のパラメーターの説明を参照してください)。プロシージャは、指定したバージョンのすべての XML 文書进行处理するか、またはバージョンが無効な場合はエラー (+20458) を戻します。出力において、このパラメーターは、プロシージャによってサポートされている最も高いメジャー文書バージョンを指定します。サポートされる文書の最も高いバージョンを判別するには、この入力パラメーターおよびその他すべての必須パラメーターに NULL を指定します。

*xml\_input* パラメーターで文書タイプのメジャー・バージョンのキーが指定されている場合、*major\_version* パラメーターが NULL 以外であれば、*major\_version* パラメーターが優先されます。

サポートされているバージョン: 1

### *minor\_version*

マイナー文書バージョンを指す、タイプ INTEGER の入出力引数。入力において、この引数は、プロシージャでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする、マイナー文書バージョンを示します (*xml\_input*、*xml\_output*、および *xml\_message* のパラメーターの説明を参照してください)。プロシージャは、指定したバージョンのすべての XML 文書进行处理するか、またはバージョンが無効な場合はエラーを戻します。出力において、このパラメーターは、サポートされている最も高いメジャー・バージョンでサポートされている、最も高いマイナー文書バージョンを指します。サポートされる文書の最も高いバージョンを判別するには、この入力パラメーターおよびその他すべての必須パラメーターに NULL を指定します。

*xml\_input* パラメーターで文書タイプのマイナー・バージョンのキーが指定されている場合、*minor\_version* パラメーターが NULL 以外であれば、*minor\_version* パラメーターが優先されます。

サポートされているバージョン: 0

### *requested\_locale*

ロケールを指定する、タイプ VARCHAR(33) の入力引数。指定した言語がサーバーでサポートされている場合は、変換された内容が *xml\_output* および

`xml_message` パラメーターに戻されます。サポートされていない場合は、内容はデフォルトの言語で戻されます。ロケールから使用されるのは、言語 (および場合によっては地域情報) のみです。ロケールは数字のフォーマット設定に使用されたり、文書エンコードに影響を与えたりすることはありません。例えば、キー名と値は変換されません。XML 出力と XML メッセージ文書で翻訳される部分は、エラー・メッセージのテキストのみです。呼び出し元は常に、要求された言語と XML 出力文書で使用されている言語を比較する必要があります (XML 出力文書の文書ロケール項目を参照してください)。

現時点でサポートされている `requested_locale` の値は `en_US` のみです。

#### `xml_input`

PLIST XML 入力ストリングを指定する、タイプ BLOB(32M) の入力引数。

#### `xml_filter`

タイプ BLOB(4K) の入力引数。このパラメーターは、将来の利用のために予約されています。

#### `xml_output`

PLIST XML 出力ストリングを戻す、タイプ BLOB(12K) の出力パラメーター。

#### `xml_message`

Data Server Message タイプの完全な XML 出力文書を UTF-8 エンコード方式で戻す、タイプ BLOB(64K) の出力パラメーター。この文書には、SQL 警告条件に関する詳細が記されています。

## 許可

- データベースへの読み取りアクセス。
- 現在接続しているスキーマまたは SYSTOOLS スキーマの Explain 表への読み取りおよび書き込みアクセス。
- マテリアライズ照会表 (MQT) を使用する場合、CREATE TABLE 許可、および MQT に対する読み取り/書き込みアクセスが必要です。
- DESIGN\_ADVISOR 関数に対する EXECUTE 特権。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 戻される情報

この情報は、エラーが生成されない限り、必ず戻されます。

表 83. DESIGN\_ADVISOR - 設計アドバイザーの推奨値の取得表関数によって戻される情報

列名	データ・タイプ	説明
SCHEMA	VARCHAR(128)	この推奨値の適用先となるオブジェクトまたは提案オブジェクトのスキーマ名。
NAME	VARCHAR(128)	この推奨値の適用先となるオブジェクトまたは提案オブジェクトの名前。
EXISTS	CHAR(1)	オブジェクトが存在することを示します。

表 83. DESIGN\_ADVISOR - 設計アドバイザーの推奨値の取得表関数によって戻される情報  
(続き)

列名	データ・タイプ	説明
RECOMMENDATION	VARCHAR(8)	<p>勧告タイプを示します。有効な値は以下のとおりです。</p> <ul style="list-style-type: none"> <li>• MQT の場合には M</li> <li>• データベース・パーティショニングの場合には P</li> <li>• MDC の場合には C</li> <li>• 索引の場合には I</li> <li>• オブジェクトが指定のワークロードによって使用されていない場合で、削除対象と見なせる場合には D</li> </ul> <p>結果は、これらの値の任意の組み合わせとなる可能性があります。例えば「MC」は、表が MQT 表および MDC 表として推奨されることを示します。</p>
BENEFIT	DOUBLE	<p>提案オブジェクト、または照会の提案オブジェクトに対する変更に関する timeron 単位の見積もり上の利点。MDC またはパーティションに関する推奨値がある基本表または MQT の場合、この値は NULL です。</p>
OVERHEAD	DOUBLE	<p>提案オブジェクト、または提案オブジェクトへの変更のいずれかを保守するための timeron 単位の見積もりコスト。索引は、公式 <math>BENEFIT - OVERHEAD</math> によってランク付けされます。MQT は <math>BENEFIT - (0.5 * OVERHEAD)</math> によってランク付けされます。MDC またはパーティションに関する推奨値がある基本表または MQT の場合、この値は NULL です。</p>
STATEMENT_NO	INTEGER	<p>この推奨値によって参照されるステートメント番号。ADVISE_WORKLOAD 表のステートメント番号が反映されます。推奨値が複数のステートメントに適用される場合、ステートメントごとに 1 行のみが戻されます。</p>
DISKUSE	DOUBLE	<p>推奨オブジェクト、またはディスク上の現行オブジェクトに対する変更結果の見積もりサイズ (MB 単位)。</p>

## 使用上の注意

### XML\_INPUT オプション

表 84. XML\_INPUT オプション

キー名	Optional	データ・タイプ	説明
MAJOR_VERSION	Y	INTEGER	入力として、クライアントによってサポートされている XML_OUTPUT スキーマのメジャー・バージョン。同じ名前のプロシージャ・パラメーターが指定される場合、指定されたパラメーターが使用されます。それ以外の場合、XML_INPUT でこの値が検索され、それは必要とされます。
MINOR_VERSION	Y	INTEGER	入力として、クライアントによってサポートされている XML_OUTPUT スキーマのマイナー・バージョン。同じ名前のプロシージャ・パラメーターが指定される場合、指定されたパラメーターが使用されます。それ以外の場合、XML_INPUT でこの値が検索され、それは必要とされます。
REQUESTED_LOCALE	Y	STRING	ロケールがサーバーでサポートされている場合、要求されたロケールでエラー・メッセージが戻されます。ロケールがサポートされていない場合または無効な場合には、サーバーのデフォルト・ロケールでデータが戻されます。同じ名前のプロシージャ・パラメーターが指定される場合、指定されたパラメーターが使用されます。それ以外の場合、XML_INPUT でこの値が検索され、それは必要とされます。
CMD_OPTIONS	N	STRING	<b>db2adv</b> コマンドによって受け入れられる引数のリスト。 <b>db2adv</b> コマンドとこのプロシージャで受け入れられるコマンド行パラメーターの違いのリストについては、「使用上の注意」を参照してください。

表 84. XML\_INPUT オプション (続き)

キー名	Optional	データ・タイプ	説明
USER_TEMP_TSPACE	Y	STRING	宣言済みのグローバル一時表 (DGTT) を作成して結果セットを保持できる USER TEMPORARY TABLESPACE の名前。名前を指定しないと、fenced プロセス・メモリーが代わりに使用されます。指定の表スペースは存在し、呼び出し元により書き込み可能で、結果セット全体に対して十分なスペースがなければなりません。アプリケーションが切断するまで、DGTT は存在し、システム・リソースを使用します。スペースを節約するため、また出力は非決定的なので、コンテンツは毎回削除されます。

注: 「&」、「<」、「>」、「'」(単一引用符)、「"」(二重引用符)といった特殊文字は、対応する HTML エンティティの「&amp;」、「&lt;」、「&gt;」、「&apos;」、「&quot;」によって指定できます。

#### XML\_OUTPUT 記述

XML\_OUTPUT 文書は、常に UTF-8 コード・ページで戻されます。コード・ページ変換は、データベース ID、オブジェクトに実行され、その他の非 UTF-8 文字に実行される可能性もあります。特殊文字 (リストについては XML\_INPUT オプションを参照) も変換されます。

表 85. XML\_OUTPUT 記述

キー名	データ・タイプ	説明
文書タイプ名	STRING	「Data Server Message Output」というストリングを常に戻します。
MAJOR_VERSION	INTEGER	文書バージョン。現在、戻される値は 1 のみです。
MINOR_VERSION	INTEGER	文書のサブバージョン。現在、戻される値は 0 のみです。
NUM_OUTPUT_ROWS	INTEGER	結果セットで戻される行数。
ADVISE_START_TIME	STRING	アドバイザーが作業を開始した時点のタイム・スタンプ。ADVISE_INSTANCE.START_TIME 列に相当します。
WORKLOAD_NAME	STRING	アドバイザーが使用したワークロード名。ADVISE_WORKLOAD.WORKLOAD_NAME 列に相当します。
ADVISE_SCHEMA	STRING	Explain 表/アドバイザー表のスキーマ名。このスキーマは、ADVISE_WORKLOAD.ADVISE_INSTANCE と他の Explain 表/アドバイザー表との間で読み書きするために使用します。

表 85. XML\_OUTPUT 記述 (続き)

キー名	データ・タイプ	説明
TOTAL_DISK	STRING	すべての推奨オブジェクトを作成するのに必要な合計初期ディスク・スペース (MB 単位)。
TOTAL_DISK_UPPER_BOUND	INTEGER	解決オプションを評価する際に使用する合計ディスク・スペースの上限 (MB 単位)。
ORIG_TOTAL_COST	STRING	推奨値を使用しない場合の合計コスト (timeron 単位)。
NEW_TOTAL_COST	STRING	推奨値を使用する場合の合計コスト (timeron 単位)。
NUM_SOLUTIONS_EVAL	INTEGER	アドバイザーによって考慮されて評価された解決策の数。

### db2adviz コマンド行パラメーターと DESIGN\_ADVISOR の相違点

プロシーチャーが使用しているのは現在のデータベース接続のみなので、以下のオプションは許可されません。

**-[dbld]** データベース名。現行のデータベース接続が使用されます。

**-[userla]**

接続に使用するユーザー名 (オプションで、ユーザー・パスワードも)。DESIGN\_ADVISOR では、このオプションは SESSION\_USER 特殊レジスターによって置き換わります。

**-[passwordlx]**

このパラメーターは、パスワードが標準入力から読み取られることを示します。DESIGN\_ADVISOR では使用されません。

ファイルとディレクトリーの場所に関しては、異なるデータベース・サーバーのインストールにおいても一貫性のある動作を確保するために、可能な限り絶対パス名を指定してください。ファイルおよびディレクトリーは、Linux/UNIX の場合には fenced ユーザー ID、または Windows の場合には DB2USERS グループによって、読み取り可能 (-file) または書き込み可能 (-script) でなければなりません。

コマンド行オプション **-file** または **-script** を使用すると、後ほど固有のワークロード名で参照するために、ステートメントが ADVISE\_WORKLOAD 表に挿入されます。

### db2adviz によって使用される種々のスキーマの解説

#### Explain 表/アドバイザー表のスキーマ名

DESIGN\_ADVISOR プロシーチャーが使用する Explain 表/アドバイザー表のスキーマ名は、CURRENT\_USER 特殊レジスターによって定義されます。この特殊レジスターは、デフォルトで現在の接続済みユーザーに設定されます。Explain 表/アドバイザー表が前述の 2つのオプションで定義されているユーザー ID を使用しても見つからない場合、SYSTOOLS スキーマが使用されます。

#### 推奨されたオブジェクトのスキーマ名

推奨されたオブジェクトのスキーマ名は、オプション

で、**-[schema|n]** コマンド行オプションを使用して定義します。名前を指定しないと、デフォルトで **SESSION\_USER** 特殊レジスタの値が使用されます。

### デフォルトのワークロード・スキーマ名

オプションで、デフォルトのワークロードのスキーマ名を **-q** コマンド行オプションを使用して定義します。名前を指定しないと、デフォルトで **SESSION\_USER** 特殊レジスタの値が使用されます。

## 例

### 例 1: XML\_INPUT の例

```
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
  <dict>
    <key>MAJOR_VERSION</key><integer>1</integer>
    <key>MINOR_VERSION</key><integer>0</integer>
    <key>REQUESTED_LOCALE</key><string>en_US</string>
    <key>CMD_OPTIONS</key><string>-w "workload 1" -t 5</string>
    <key>USER_TEMP_TSPACE</key><string>MY_TEMP_TS</string>
  </dict>
</plist>
```

### 例 2: XML\_OUTPUT の例

```
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
  <dict>
    <key>Document Type Name</key><string>Data Server Message Output</string>
    <key>MAJOR_VERSION</key><integer>1</integer>
    <key>MINOR_VERSION</key><integer>0</integer>
    <key>NUM_OUTPUT_ROWS</key><integer>1</integer>
    <key>NUM_RESULT_SETS</key><integer>1</integer>
    <key>ADVISE_START_TIME</key><string>2011-03-10-14.22.51.707742</string>
    <key>WORKLOAD_NAME</key><string>MYWORKLOAD</string>
    <key>ADVISE_SCHEMA</key><string>MYSCHEMA</string>
    <key>TOTAL_DISK</key><string>0.0762</string>
    <key>TOTAL_DISK_UPPER_BOUND</key><string>33.3203</string>
    <key>ORIG_TOTAL_COST</key><string>28434.0000</string>
    <key>NEW_TOTAL_COST</key><string>11108.0000</string>
    <key>NUM_SOLUTIONS_EVAL</key><integer>31</integer>
  </dict>
</plist>
```

### 例 3: XML\_MESSAGE の例

```
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
  <dict>
    <key>db21uw20458E</key><string>SQLCODE: 20458, SQLSTATE: 01H54 ,
    <![CDATA[SQL20458W The procedure "SYSPROC.DESIGN_ADVISOR" has
    encountered an internal parameter processing error in parameter "4".
    The value for parameter "7" contains further information about
    the error.]]>(Extra message: <![CDATA[Error: Unsupported major
    version.]]></string>
  </dict>
</plist>
```

### 例 4: 完全初期化とプロシージャ呼び出し (結果の XML 出力付き)

ワークロードの初期化:





BENEFIT	OVERHEAD	STMTNO	DISKUSE
+0.000000000000000E+000	+0.000000000000000E+000	0	+0.000000000000000E+000
+1.720000000000000E+004	+0.000000000000000E+000	1	+0.063500000000000E+000
+1.260000000000000E+002	+0.000000000000000E+000	2	+1.271900000000000E-002

3 record(s) selected.

## GET\_CONFIG プロシージャ - 構成データの取得

GET\_CONFIG ストアード・プロシージャは、特定のインスタンスに対して設定されたデータベース構成、データベース・マネージャ構成、およびレジストリー変数を検索します。

パーティション・データベース環境では、このプロシージャはすべてのパーティションから、データベース構成およびレジストリー変数設定値を検索します。

### 構文

```
▶▶ GET_CONFIG (—major_version—, —minor_version—, —requested_locale—, —
—xml_input—, —xml_filter—, —xml_output—, —xml_message—)
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### major\_version

メジャー文書バージョンを指す、タイプ INTEGER の入出力引数。入力において、この引数は、プロシージャでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする、メジャー文書バージョンを示します (*xml\_input*、*xml\_output*、および *xml\_message* のパラメーターの説明を参照してください)。プロシージャは、指定したバージョンのすべての XML 文書进行处理するか、またはバージョンが無効な場合はエラー (+20458) を戻します。出力において、このパラメーターは、プロシージャによってサポートされている最も高いメジャー文書バージョンを指定します。サポートされる文書の最も高いバージョンを判別するには、この入力パラメーターおよびその他すべての必須パラメーターに NULL を指定します。

サポートされているバージョン: 1 および 2

#### minor\_version

マイナー文書バージョンを指す、タイプ INTEGER の入出力引数。入力において、この引数は、プロシージャでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする、マイナー文書バージョンを示します (*xml\_input*、*xml\_output*、および *xml\_message* のパラメーターの説明を参照してください)。プロシージャは、指定したバージョンのすべての XML 文書进行处理するか、またはバージョンが無効な場合はエラーを戻します。出力において、このパラメーターは、サポートされている最も高いメジャー・バージョンでサポートされている、最も高いマイナー文書バージョンを指します。サポートされる文書の最も高いバージョンを判別するには、この入力パラメーターおよびその他すべての必須パラメーターに NULL を指定します。

サポートされているバージョン: 0

### *requested\_locale*

ロケールを指定する、タイプ VARCHAR(33) の入力引数。指定した言語がサーバーでサポートされている場合は、変換された内容が *xml\_output* および *xml\_message* パラメーターに戻されます。サポートされていない場合は、内容はデフォルトの言語に戻されます。ロケールから使用されるのは、言語 (および場合によっては地域情報) のみです。ロケールは数字のフォーマット設定に使用されたり、文書エンコードに影響を与えたりすることはありません。例えば、キー名と値は変換されません。XML 出力および XML メッセージ文書で変換される部分は、各項目のヒントのテキスト、表示名、および表示単位のみです。呼び出し元は常に、要求された言語と XML 出力文書で使用されている言語を比較する必要があります (XML 出力文書の文書ロケール項目を参照してください)。

現時点でサポートされている *requested\_locale* の値は en\_US のみです。

### *xml\_input*

現時点で、このプロシージャは入力を受け入れません。このパラメーターに NULL を指定してください。NULL を指定しないとエラー (+20458) が発生し、入力が無効であることが示されます。

### *xml\_filter*

有効な XPath 照会ストリングを指定する、タイプ BLOB(4K) の入力引数。XML 出力文書から単一値を検索する場合、フィルターを使用します。詳しくは、XPath フィルター操作について記述しているトピックを参照してください。

次の例では、XML 出力文書から、データ・サーバー製品バージョンの値を選択しています (/plist/dict/key[.='Data Server Product Version']/following-sibling::string)。キーの後に兄弟が指定されていないと、エラーが戻されません。

### *xml\_output*

UTF-8 の完全な XML 出力文書に戻す、タイプ BLOB(32MB) の出力パラメーターに戻します。フィルターが指定されている場合、このパラメーターはストリング値に戻します。ストアド・プロシージャが完全な出力文書に戻すことができない場合 (例えば、処理エラーが発生して SQL 警告やエラーが出される場合など)、このパラメーターは NULL に設定されます。

XML 出力は、以下のように *major\_version* および *minor\_version* で指定した値によって決まります。

メジャー・バージョン	マイナー・バージョン	<i>xml_output</i> の値
NULL	NULL	NULL
1	0	データベース・マネージャー構成パラメーター、データベース構成パラメーター、およびレジストリー変数とそれぞれの値

メジャー・バージョン	マイナー・バージョン	xml_output の値
2	0	カテゴリーにグループ化された、データベース・マネージャーおよびデータベース・マネージャー構成パラメーター各パラメーターに関して、そのパラメーターが更新可能かどうかを示します。さらに、インスタンスに設定された、レジストリー変数および値も戻します。

プロシージャが `コンプリート・モード` で動作する場合、このパラメーターによって戻される XML 文書は、変更して `xml_input` パラメーターとしてプロシージャに再び渡すことができます。このアプローチにより、有効な XML 入力文書を作成するためのプログラムの方法が提供されます。詳しくは、`コンプリート・モード` についてのトピックを参照してください。

#### *xml\_message*

SQL 警告状態についての詳細情報を提供する Data Server Message タイプの完全な XML 出力文書を UTF-8 で戻す、タイプ BLOB(64K) の出力パラメーター。プロシージャの呼び出しにより SQL 警告が出され、XML メッセージ出力文書に追加情報が戻されることを警告メッセージが示す場合に、この文書が戻されます。追加情報が戻されることを警告メッセージが示していない場合は、このパラメーターは NULL に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: プロシージャの最新バージョンを戻します。

```
db2 "call sysproc.get_config(null,null,null,null,null,?,?)"
```

以下はこの照会の出力例です。

```
Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 2

Parameter Name : MINOR_VERSION
```

Parameter Value : 0

Parameter Name : XML\_OUTPUT  
Parameter Value : -

Parameter Name : XML\_MESSAGE  
Parameter Value : -

Return Status = 0

例 2: カテゴリにグループ化された、データベースおよびデータベース・マネージャー構成パラメーターを戻します。

```
db2 "call sysproc.get_config(2,0,'en_US',null, null, ?,?)"
```

以下はこの照会の出力例です。

```
Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 2

Parameter Name : MINOR_VERSION
Parameter Value : 0

Parameter Name : XML_OUTPUT
Parameter Value : x'3C3F78.....'

Parameter Name : XML_MESSAGE
Parameter Value : -

Return Status = 0
```

XML 出力文書には次の内容が含まれます。

```
<plist version="1.0">
<dict>
  <key>Document Type Name</key><string>Data Server Configuration Output</string>
  <key>Document Type Major Version</key><integer>2</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Data Server Product Name</key><string>QDB2/AIX64</string>
  <key>Data Server Product Version</key><string>9.7.0.0</string>
  <key>Data Server Major Version</key><integer>9</integer>
  <key>Data Server Minor Version</key><integer>7</integer>
  <key>Data Server Platform</key><string>AIX 64BIT</string>
  <key>Document Locale</key><string>en_US</string>
  <key>Database Manager Configuration Parameter Settings</key>
  <dict>
    <key>Display Name</key>
    <string>Database Manager Configuration Parameter Settings</string>
    <key>Application</key>
    <dict>
      <key>Display Name</key><string>Application</string>
      <key>agentpri</key>
      <dict>
        <key>Display Name</key><string>agentpri</string>
        <key>Parameter Value</key>
        <dict>
          <key>Display Name</key><string>Parameter Value</string>
          <key>Value</key><string>-1</string>
          <key>Updatable</key><string>No</string>
          <key>Hint</key><string></string>
        </dict>
      <key>Value Flags</key>
      <dict>
        <key>Display Name</key><string>Value Flags</string>
        <key>Value</key><string>NONE</string>
      </dict>
    </dict>
  </dict>
</plist>
```

```

        <key>Updatable</key><string>No</string>
        <key>Hint</key><string></string>
    </dict>
    <key>Deferred Value</key>
    <dict>
        <key>Display Name</key><string>Deferred Value</string>
        <key>Value</key><string>-1</string>
        <key>Updatable</key><string>Yes</string>
        <key>Hint</key><string></string>
    </dict>
    <key>Deferred Value Flags</key>
    <dict>
        <key>Display Name</key><string>Deferred Value Flags</string>
        <key>Value</key><string>INTEGER</string>
        <key>Updatable</key><string>Yes</string>
        <key>Hint</key><string></string>
    </dict>
    <key>Data Type</key>
    <dict>
        <key>Display Name</key><string>Data Type</string>
        <key>Value</key><string>NONE</string>
        <key>Hint</key><string></string>
    </dict>
    <key>Hint</key>
    <string>
        Specifies the priority given to an agent and other database manager
        instance processes and threads by the operating system scheduler.
        Consider rebinding applications after changing this parameter.
    </string>
    </dict>
    <key>Hint</key><string></string>
</dict>
</dict>
<key>Administration</key>
.
.
<key>Communication</key>
.
.
<key>Diagnostics</key>
.
.
<key>Environment</key>
.
.
<key>Miscellaneous</key>
.
.
<key>Monitor</key>
.
.
<key>Parallel</key>
.
.
<key>Performance</key>
.
.
</dict>
<key>Database Partition</key>

```

```

<dict>
  <key>Display Name</key><string>Database Partition</string>
  <key>0</key>
  <dict>
    <key>Display Name</key><string>0</string>
    <key>Database Configuration Parameter Settings</key>
    <dict>
      <key>Display Name</key>
      <string>Database Configuration Parameter Settings</string>
      <key>Application</key>
      .
      .
      .
    <key>Environment</key>
  <dict>
    <key>Display Name</key><string>Environment</string>
    <key>alt_collate</key>
    <dict>
      <key>Display Name</key><string>alt_collate</string>
      <key>Parameter Value</key>
      <dict>
        <key>Display Name</key><string>Parameter Value</string>
        <key>Value</key><string></string>
        <key>Updatable</key><string>No</string>
        <key>Hint</key><string></string>
      </dict>
      <key>Value Flags</key>
      <dict>
        <key>Display Name</key><string>Value Flags</string>
        <key>Value</key><string>NONE</string>
        <key>Updatable</key><string>No</string>
        <key>Hint</key><string></string>
      </dict>
      <key>Deferred Value</key>
      <dict>
        <key>Display Name</key><string>Deferred Value</string>
        <key>Value</key><string></string>
        <key>Updatable</key><string>Yes</string>
        <key>Hint</key><string></string>
      </dict>
      <key>Deferred Value Flags</key>
      <dict>
        <key>Display Name</key><string>Deferred Value Flags</string>
        <key>Value</key><string>INTEGER</string>
        <key>Updatable</key><string>Yes</string>
        <key>Hint</key><string></string>
      </dict>
      <key>Data Type</key>
      <dict>
        <key>Display Name</key><string>Data Type</string>
        <key>Value</key><string>NONE</string>
        <key>Hint</key><string></string>
      </dict>
      <key>Hint</key>
      <string>
        Specifies the collating sequence to be used for Unicode tables in a
        non-Unicode database. Until this parameter is set, Unicode tables and
        routines cannot be created in a non-Unicode database. When set, this
        parameter cannot be changed or reset. Default [range] :
        Null [IDENTITY_16BIT].
      </string>
    </dict>
    .
    .
    .
  </dict>
  <key>Logs</key>

```

```

.
.
.
<key>Maintenance</key>
.
.
.
<key>Performance</key>
.
.
.
<key>Recovery</key>
.
.
.
<key>Status</key>
.
.
.
</dict>
<key>Registry Variables Settings</key>
<dict>
  <key>Display Name</key><string>Registry Variables Settings</string>
  <key>DB2CODEPAGE</key>
  <dict>
    <key>Display Name</key><string>DB2CODEPAGE</string>
    <key>Parameter Value</key>
    <dict>
      <key>Display Name</key><string>Parameter Value</string>
      <key>Value</key><string>1208</string>
      <key>Hint</key><string></string>
    </dict>
    <key>Is Aggregate</key>
    <dict>
      <key>Display Name</key><string>Is Aggregate</string>
      <key>Value</key><integer>0</integer>
      <key>Hint</key><string></string>
    </dict>
    <key>Aggregate Name</key>
    <dict>
      <key>Display Name</key><string>Aggregate Name</string>
      <key>Value</key><string></string>
      <key>Hint</key><string></string>
    </dict>
    <key>Level</key>
    <dict>
      <key>Display Name</key><string>Level</string>
      <key>Value</key><string>I</string>
      <key>Hint</key><string></string>
    </dict>
    <key>Hint</key><string></string>
  </dict>
  .
  .
  .
</dict>
<key>Hint</key><string></string>
</dict>
</plist>

```

例 3: データベースおよびデータベース・マネージャー構成パラメーターを戻します。

```
db2 "call sysproc.get_config(1,0,'en_US',null, null, ?,?)"
```



以下はこの照会の出力例です。

```
Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 1

Parameter Name : MINOR_VERSION
Parameter Value : 0

Parameter Name : XML_OUTPUT
Parameter Value : x'3C3F78.....'

Parameter Name : XML_MESSAGE
Parameter Value : -

Return Status = 0
```

XML 出力文書には、例 2 のような内容が含まれますが、構成パラメーターはカテゴリーにグループ化されません。

例 4: 関数からプロシージャを呼び出します。

```
EXEC SQL BEGIN DECLARE SECTION;
  sqlint16 getconfigMaj;
  sqlint16 getconfigMin;

  SQL TYPE IS BLOB(2M) xmlOutput;
  SQL TYPE IS BLOB(2K) xmlOutMessage;
EXEC SQL END DECLARE SECTION;
  getconfigMaj = 2;
  getconfigMin = 0;

EXEC SQL CALL SYSPROC.GET_CONFIG(
  :getconfigMaj,
  :getconfigMin,
  'en_US',
  null,
  null,
  :xmlOutput,
  :xmlOutMessage );
```

## GET\_MESSAGE プロシージャ - メッセージ・テキストの取得

GET\_MESSAGE プロシージャは、SQLCODE の簡略メッセージ・テキスト、詳細メッセージ・テキスト、および SQLSTATE を戻します。

### 構文

```
▶▶GET_MESSAGE(—major_version—,—minor_version—,—requested_locale—,—▶▶
▶—xml_input—,—xml_filter—,—xml_output—,—xml_message—)▶▶
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### major\_version

メジャー文書バージョンを指す、タイプ INTEGER の入出力引数。入力において、この引数は、プロシージャでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする、メジャー文書バージョンを示します (*xml\_input*、*xml\_output*、および *xml\_message* のパラメーターの説明を参照して

ください)。プロシージャーは、指定したバージョンのすべての XML 文書処理するか、またはバージョンが無効な場合はエラー (+20458) を戻します。出力において、このパラメーターは、プロシージャーによってサポートされている最も高いメジャー文書バージョンを指定します。サポートされる文書の最も高いバージョンを判別するには、この入力パラメーターおよびその他すべての必須パラメーターに NULL を指定します。

*xml\_input* パラメーターの XML 文書で、文書タイプのメジャー・バージョンのキーが指定されている場合は、そのキーの値は *major\_version* パラメーターで指定される値と同じでなければなりません。同じでない場合、エラー (+20458) が発生します。

**サポートされているバージョン: 1 および 2**

#### *minor\_version*

マイナー文書バージョンを指す、タイプ INTEGER の入出力引数。入力において、この引数は、プロシージャーでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする、マイナー文書バージョンを示します (*xml\_input*、*xml\_output*、および *xml\_message* のパラメーターの説明を参照してください)。プロシージャーは、指定したバージョンのすべての XML 文書処理するか、またはバージョンが無効な場合はエラーを戻します。出力において、このパラメーターは、サポートされている最も高いメジャー・バージョンでサポートされている、最も高いマイナー文書バージョンを指します。サポートされる文書の最も高いバージョンを判別するには、この入力パラメーターおよびその他すべての必須パラメーターに NULL を指定します。

*xml\_input* パラメーターの XML 文書で、文書タイプのマイナー・バージョンのキーが指定されている場合は、そのキーの値は *minor\_version* パラメーターで指定される値と同じでなければなりません。同じでない場合、エラー (+20458) が発生します。

**サポートされているバージョン: 0**

#### *requested\_locale*

ロケールを指定する、タイプ VARCHAR(33) の入力引数。指定した言語がサーバーでサポートされている場合は、変換された内容が *xml\_output* および *xml\_message* パラメーターに戻されます。サポートされていない場合は、内容はデフォルトの言語で戻されます。ロケールから使用されるのは、言語 (および場合によっては地域情報) のみです。ロケールは数字のフォーマット設定に使用されたり、文書エンコードに影響を与えたりすることはありません。例えば、キー名と値は変換されません。XML 出力および XML メッセージ文書で変換される部分は、各項目のヒントのテキスト、表示名、および表示単位のみです。呼び出し元は常に、要求された言語と XML 出力文書で使用されている言語を比較する必要があります (XML 出力文書の文書ロケール項目を参照してください)。

現時点でサポートされている *requested\_locale* の値は en\_US のみです。

#### *xml\_input*

プロシージャーの入力値が含まれる XML 入力文書 (UTF-8 でエンコードされている) を指定する、タイプ BLOB(32MB) の入力引数。

このプロシージャーでは、XML 入力文書に SQLCODE が含まれ、次のフォーマットを使用します。

```

<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key><string>Data Server Message Input</string>
  <key>Required Parameters</key>
  <!-- Specify either SQLCODE or message identifier and message tokens
  for the key values below. -->
  <dict>
    <key>SQL Code</key><integer></integer>
    <key>Message Identifier</key><integer></integer>
    <key>Message Tokens</key><array><string>...</string></array>
  </dict>
  <key>Optional Parameters</key>
  <dict>
    <key>Message Token Delimiter<key><string>;</string>
  </key></key></dict>
</dict>
</plist>

```

### xml\_filter

有効な XPath 照会ストリングを指定する、タイプ BLOB(4K) の入力引数。XML 出力文書から単一値を検索する場合、フィルターを使用します。詳しくは、XPath フィルター操作について記述しているトピックを参照してください。

次の例では、XML 出力文書から SQLSTATE の値を選択しています (/plist/dict/key[.="SQLSTATE"]/following-sibling::dict[1]/key[.="Value"]/following-sibling::string[1])。キーの後に兄弟が指定されていないと、エラーが戻されます。

### xml\_output

UTF-8 の完全な XML 出力文書を戻す、タイプ BLOB(32MB) の出力パラメーターを戻します。フィルターが指定されている場合、このパラメーターはストリング値を戻します。ストアード・プロシージャが完全な出力文書を戻すことができない場合 (例えば、処理エラーが発生して SQL 警告やエラーが出される場合など)、このパラメーターは NULL に設定されます。

XML 出力は、以下のように *major\_version* および *minor\_version* で指定した値によって決まります。

メジャー・バージョン	マイナー・バージョン	xml_output の値
NULL	NULL	NULL
1	0	xml_input に渡された対応する SQLCODE の、簡略テキスト・メッセージおよび SQLSTATE を戻します。
2	0	xml_input に渡された対応する SQLCODE の、簡略テキスト・メッセージ、詳細テキスト・メッセージ、および SQLSTATE を戻します。

プロシージャが コンプリート・モード で動作する場合、このパラメーターによって戻される XML 文書は、変更して xml\_input パラメーターとしてプロシージャに再び渡すことができます。このアプローチにより、有効な XML 入力文書を作成するためのプログラムの方法が提供されます。詳しくは、コンプリート・モードについてのトピックを参照してください。

## xml\_message

SQL 警告状態についての詳細情報を提供する Data Server Message タイプの完全な XML 出力文書を UTF-8 で戻す、タイプ BLOB(64K) の出力パラメーター。プロシージャーの呼び出しにより SQL 警告が出され、XML メッセージ出力文書に追加情報が戻されることを警告メッセージが示す場合に、この文書が戻されます。追加情報が戻されることを警告メッセージが示していない場合は、このパラメーターは NULL に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャーが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: プロシージャーの、サポートされる最も高いバージョンを戻します。

```
db2 "call sysproc.get_message(null,null,null,null,null,?,?)"
```

以下はこの照会の出力例です。

```
Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 2

Parameter Name : MINOR_VERSION
Parameter Value : 0

Parameter Name : XML_OUTPUT
Parameter Value : -

Parameter Name : XML_MESSAGE
Parameter Value : -

Return Status = 0
```

例 2: getmsglong.sql というスクリプトを実行し、SQL1034 の簡略テキスト・メッセージおよび詳細テキスト・メッセージを戻します。

```
getmsglong.sql:
```

```
call sysproc.get_message(2,0, 'en_US', blob('
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key><string>Data Server Message Input</string>
  <key>Document Type Major Version</key><integer>2</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Required Parameters</key>
</dict>
```

```
    <key>SQLCODE</key><string>SQL1034</string>
  </dict>
</dict>
</plist>'), null, ? , ?)@
```

以下はこの照会の出力例です。

Value of output parameters

-----  
Parameter Name : MAJOR\_VERSION  
Parameter Value : 2

Parameter Name : MINOR\_VERSION  
Parameter Value : 0

Parameter Name : XML\_OUTPUT  
Parameter Value : x'3C3F786D6C20766572.....'

Parameter Name : XML\_MESSAGE  
Parameter Value : -

Return Status = 0

出力 XML 文書には次の内容が含まれます。

```
<plist version="1.0">
<dict>
  <key>Document Type Name</key>
  <string>Data Server Message Output</string>
  <key>Document Type Major Version</key>
  <integer>2</integer>
  <key>Document Type Minor Version</key>
  <integer>0</integer>
  <key>Data Server Product Name</key>
  <string>QDB2/AIX64</string>
  <key>Data Server Product Version</key>
  <string>9.7.0.0</string>
  <key>Data Server Major Version</key>
  <integer>9</integer>
  <key>Data Server Minor Version</key>
  <integer>7</integer>
  <key>Data Server Platform</key>
  <string>AIX 64BIT</string>
  <key>Document Locale</key>
  <string>en_US</string>
  <key>Short Message Text</key>
  <dict>
    <key>Display Name</key><string>Short Message Text</string>
    <key>Value</key>
    <string>
SQL1034C The database is damaged. All applications processing the database
have been stopped.
</string>
    <key>Hint</key><string></string>
  </dict>
</dict>
  <key>SQLSTATE</key>
  <dict>
    <key>Display Name</key><string>SQLSTATE</string>
    <key>Value</key><string> 58031</string>
    <key>Hint</key><string></string>
  </dict>
  <key>Long Message Text</key>
  <dict>
    <key>Display Name</key><string>Long Message Text</string>
    <key>Value</key>
    <array>
      <string>
```

```

SQL1034C The database is damaged. All applications
processing the
</string>
<string> database have been stopped.</string>
<string></string>
<string>Explanation: </string>
<string></string>
<string>
Damage has occurred to the database. It cannot be used until it is
</string>
<string>
recovered. All applications connected to the database have been
</string>
<string>
disconnected and all processes running applications on the
database have
</string>
<string>been stopped.</string>
<string></string>
<string>The command cannot be processed.</string>
<string></string>
<string>User response: </string>
<string></string>
<string>
Issue a RESTART DATABASE command to recover the database. If the RESTART
</string>
<string>
command consistently fails, you may want to restore the database from a
</string>
<string>
backup. In a partitioned database server environment, check the syslog
</string>
<string>
to find out if the RESTART command fails because of node or
</string>
<string>
communication failures before restoring the database from a backup. If
</string>
<string>
so, ensure the database manager is up and running and communication is
</string>
<string>
available among all the nodes, then resubmit the restart command.
</string>
<string></string>
<string>
If you encountered this error during roll-forward processing, you must
</string>
<string>
restore the database from a backup and perform roll-forward again.
</string>
<string></string>
<string>
Note that in a partitioned database environment, the RESTART database
</string>
<string>
command is run on a per-node basis. To ensure that the database is
</string>
<string>restarted on all nodes, use the command: </string>
<string></string>
<string>db2_all db2 restart database</string>
<string><database_name></string>
<string></string>
<string>
This command may have to be run several times to ensure that all
</string>
<string>in-doubt transactions have been resolved.</string>

```

```

    <string></string>
    <string>
    If you are installing the sample database, drop it and install the
    </string>
    <string>sample database again.</string>
    <string></string>
    <string> sqlcode: -1034</string>
    <string></string>
    <string> sqlstate: 58031</string>
    <string></string>
    <string></string>
    <string></string>
  </array>
  <key>Hint</key><string></string>
</dict>
</dict>
</plist>

```

例 3: getmsgshort.sql というスクリプトを実行し、SQL1034 の簡略テキスト・メッセージのみを戻します。

getmsgshort.sql:

```

call sysproc.get_message(1,0,'en_US', blob('
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key><string>Data Server Message Input</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Required Parameters</key>
  <dict>
    <key>SQLCODE</key><string>SQL1034</string>
  </dict>
</dict>
</plist>'), null, ? , ?)@

```

以下はこの照会の出力例です。

```

Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 2

Parameter Name : MINOR_VERSION
Parameter Value : 0

Parameter Name : XML_OUTPUT
Parameter Value : x'3C3F786D6C20766572.....'

Parameter Name : XML_MESSAGE
Parameter Value : -

Return Status = 0

```

SQL20460W The procedure "SYSPROC.GET\_MESSAGE" supports a higher version, "2", than the specified version, "1", for parameter "1".

XML 出力文書には次の内容が含まれます。

```

<plist version="1.0">
<dict><key>Document Type Name</key><string>Data Server Message Output</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Data Server Product Name</key><string>QDB2/AIX64</string>
  <key>Data Server Product Version</key><string>9.7.0.0</string>
  <key>Data Server Major Version</key><integer>9</integer>

```

```

<key>Data Server Minor Version</key><integer>7</integer>
<key>Data Server Platform</key><string>AIX 64BIT</string>
<key>Document Locale</key><string>en_US</string>
<key>Short Message Text</key>
<dict>
  <key>Display Name</key><string>Short Message Text</string>
  <key>Value</key>
  <string>
    SQL1034C The database is damaged. All applications processing the database
    have been stopped.
  </string>
  <key>Hint</key><string></string>
</dict>
<key>SQLSTATE</key>
<dict>
  <key>Display Name</key><string>SQLSTATE</string>
  <key>Value</key><string> 58031</string>
  <key>Hint</key><string></string>
</dict>
</dict>
</plist>

```

例 4: フィルターを指定して、SQL1034 の SQLSTATE を戻します。

```

db2 "call sysproc.get_message(2,0, 'en_US', blob('
<plist version="1.0">
<dict>
  <key>Document Type Name</key>
  <string>Data Server Message Input</string>
  <key>Required Parameters</key>
  <dict>
    <key>SQLCODE</key><string>SQL1034</string>
  </dict>
</dict>
</plist>'),
blob('/plist/dict/key[.="SQLSTATE"]/following-sibling::dict[1]/
key[.="Value"]/following-sibling::string[1]'), ? , ?)"

```

以下はこの照会の出力例です。

```

Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 2

Parameter Name : MINOR_VERSION
Parameter Value : 0

Parameter Name : XML_OUTPUT
Parameter Value : x'203538303331'

Parameter Name : XML_MESSAGE
Parameter Value : -

Return Status = 0

```

*xml\_output* に次の値が戻されます。

```
58031
```

例 5: 関数からプロシージャを呼び出します。

```

EXEC SQL BEGIN DECLARE SECTION;
  sqlint16 getMsgMaj;
  sqlint16 getMsgMin;

SQL TYPE IS BLOB(2M) xmlOutput;

```



```

SQL TYPE IS BLOB(2K) xmlOutMessage;
EXEC SQL END DECLARE SECTION;
getMsgMaj = 2;
getMsgMin = 0;

EXEC SQL CALL SYSPROC.GET_MESSAGE(
    :getMsgMaj,
    :getMsgMin,
    'en_US',
    BLOB('
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key>
  <string>
Data Server Message Input
</string>
  <key>Document Type Major Version</key><integer>2</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Required Parameters</key>
  <dict>
    <key>SQLCODE</key><string>SQL1034</string>
  </dict>
</dict>
</plist>'),
    null,
    :xmlOutput,
    :xmlOutMessage );

```

## GET\_SYSTEM\_INFO プロシージャ - システム情報の取得

GET\_SYSTEM\_INFO プロシージャは、データ・サーバーについての情報を戻します。その情報には、システムに関する情報、現行インスタンス、インストール済みデータ・サーバー製品、環境変数、使用可能な CPU、および他のシステム情報が含まれます。

### 構文

```

▶▶ GET_SYSTEM_INFO (—major_version—, —minor_version—, —requested_locale—, —
▶ —xml_input—, —xml_filter—, —xml_output—, —xml_message—)

```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### major\_version

メジャー文書バージョンを指す、タイプ INTEGER の入出力引数。入力において、この引数は、プロシージャでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする、メジャー文書バージョンを示します (*xml\_input*、*xml\_output*、および *xml\_message* のパラメーターの説明を参照してください)。プロシージャは、指定したバージョンのすべての XML 文書进行处理するか、またはバージョンが無効な場合はエラー (+20458) を戻します。出力において、このパラメーターは、プロシージャによってサポートされている最も高いメジャー文書バージョンを指定します。サポートされる文書の最も高いバージョンを判別するには、この入力パラメーターおよびその他すべての必須パラメーターに NULL を指定します。

*xml\_input* パラメーターの XML 文書で、文書タイプのメジャー・バージョンのキーが指定されている場合は、そのキーの値は *major\_version* パラメーターで指定される値と同じでなければなりません。同じでない場合、エラー (+20458) が発生します。

サポートされているバージョン: 1

#### *minor\_version*

マイナー文書バージョンを指す、タイプ INTEGER の入出力引数。入力において、この引数は、プロシージャーでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする、マイナー文書バージョンを示します (*xml\_input*、*xml\_output*、および *xml\_message* のパラメーターの説明を参照してください)。プロシージャーは、指定したバージョンのすべての XML 文書を処理するか、またはバージョンが無効な場合はエラーを戻します。出力において、このパラメーターは、サポートされている最も高いメジャー・バージョンでサポートされている、最も高いマイナー文書バージョンを指します。サポートされる文書の最も高いバージョンを判別するには、この入力パラメーターおよびその他すべての必須パラメーターに NULL を指定します。

サポートされているバージョン: 0

#### *requested\_locale*

ロケールを指定する、タイプ VARCHAR(33) の入力引数。指定した言語がサーバーでサポートされている場合は、変換された内容が *xml\_output* および *xml\_message* パラメーターに戻されます。サポートされていない場合は、内容はデフォルトの言語に戻されます。ロケールから使用されるのは、言語 (および場合によっては地域情報) のみです。ロケールは数字のフォーマット設定に使用されたり、文書エンコードに影響を与えたりすることはありません。例えば、キー名と値は変換されません。XML 出力および XML メッセージ文書で変換される部分は、各項目のヒントのテキスト、表示名、および表示単位のみです。呼び出し元は常に、要求された言語と XML 出力文書で使用されている言語を比較する必要があります (XML 出力文書の文書ロケール項目を参照してください)。

現時点でサポートされている *requested\_locale* の値は en\_US のみです。

#### *xml\_input*

現時点で、このプロシージャーは入力を受け入れません。このパラメーターに NULL を指定してください。NULL を指定しないとエラー (+20458) が発生し、入力が無効であることが示されます。

#### *xml\_filter*

有効な XPath 照会ストリングを指定する、タイプ BLOB(4K) の入力引数。XML 出力文書から単一値を検索する場合、フィルターを使用します。詳しくは、XPath フィルター操作について記述しているトピックを参照してください。

次の例では、XML 出力文書から、データ・サーバー製品バージョンの値を選択しています (/plist/dict/key[.='Data Server Product Version']/following-sibling::string)。キーの後に兄弟が指定されていないと、エラーが戻されます。

#### *xml\_output*

UTF-8 の完全な XML 出力文書に戻す、タイプ BLOB(32MB) の出力パラメーターに戻します。フィルターが指定されている場合、このパラメーターはストリ

ング値を戻します。ストアード・プロシージャが完全な出力文書を戻すことができない場合 (例えば、処理エラーが発生して SQL 警告やエラーが出される場合など)、このパラメーターは NULL に設定されます。

XML 出力文書には、フィックスパック・レベル、リリース、システム情報、および環境変数に関する情報などの、インスタンス情報が含まれます。

#### *xml\_message*

SQL 警告状態についての詳細情報を提供する Data Server Message タイプの完全な XML 出力文書を UTF-8 で戻す、タイプ BLOB(64K) の出力パラメーター。プロシージャの呼び出しにより SQL 警告が出され、XML メッセージ出力文書に追加情報が戻されることを警告メッセージが示す場合に、この文書が戻されます。追加情報が戻されることを警告メッセージが示していない場合は、このパラメーターは NULL に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: プロシージャの最も高いバージョンを戻します。

```
db2 "call sysproc.get_system_info(null,null,null,null,null,?,?)"
```

以下はこの照会の出力例です。

```
Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 1

Parameter Name : MINOR_VERSION
Parameter Value : 0

Parameter Name : XML_OUTPUT
Parameter Value : -

Parameter Name : XML_MESSAGE
Parameter Value : -

Return Status = 0
```

例 2: システム情報を戻します。

```
db2 "call sysproc.get_system_info(1,0,'en_US',null,null,?,?)"
```

以下はこの照会の出力例です。

```

Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 1

Parameter Name : MINOR_VERSION
Parameter Value : 0

Parameter Name : XML_OUTPUT
Parameter Value : x'3C3F786D6C20766572.....

Parameter Name : XML_MESSAGE
Parameter Value : -

Return Status = 0

```

XML 出力文書には次のような内容が含まれます。

```

<plist version="1.0">
<dict><key>Document Type Name</key><string>Data Server System Output</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Data Server Product Name</key><string>QDB2/AIX64</string>
  <key>Data Server Product Version</key><string>9.7.0.0</string>
  <key>Data Server Major Version</key><integer>9</integer>
  <key>Data Server Minor Version</key><integer>7</integer>
  <key>Data Server Platform</key><string>AIX 64BIT</string>
  <key>Document Locale</key><string>en_US</string>
  <key>Instance Information</key>
  <dict>
    <key>Display Name</key><string>Instance Information</string>
    <key>Instance Name</key>
    <dict>
      <key>Display Name</key><string>Instance Name</string>
      <key>Value</key><string>myinstance</string>
      <key>Hint</key><string></string>
    </dict>
    <key>Partitionable State</key>
    <dict>
      <key>Display Name</key><string>Partitionable State</string>
      <key>Value</key><integer>0</integer>
      <key>Hint</key><string></string>
    </dict>
    <key>Number of Database Partitions</key>
    <dict>
      <key>Display Name</key><string>Number of Database Partitions</string>
      <key>Value</key><integer>1</integer>
      <key>Hint</key><string></string>
    </dict>
    .
    .
    .
  </dict>
<key>Product Information</key>
<dict>
  <key>Display Name</key><string>Product Information</string>
  .
  .
  .
  <key>DB2_ENTERPRISE_SERVER_EDITION</key>
  <dict>
    <key>Display Name</key><string>DB2_ENTERPRISE_SERVER_EDITION</string>
    <key>Product short name</key>
    <dict>
      <key>Display Name</key><string>Product short name</string>
      <key>Value</key><string>ESE</string>
      <key>Hint</key><string></string>
    </dict>
  </dict>
</dict>
</plist>

```

```

</dict>
<key>Licence</key>
<dict>
  <key>Display Name</key><string>Licence</string>
  <key>Value</key><string>Y</string>
  <key>Hint</key><string></string>
</dict>
<key>Product Release</key>
<dict>
  <key>Display Name</key><string>Product Release</string>
  <key>Value</key><string>9.7</string>
  <key>Hint</key><string></string>
</dict>
<key>Licence type</key>
<dict>
  <key>Display Name</key><string>Licence type</string>
  <key>Value</key><string>DEVELOPER</string>
  <key>Hint</key><string></string>
</dict>
<key>Hint</key><string></string>
</dict>
.
.
.
<key>Operating System Information</key>
<dict>
  <key>Display Name</key><string>Operating System Information</string>
  <key>Name</key>
  <dict>
    <key>Display Name</key><string>Name</string>
    <key>Value</key><string>AIX</string>
    <key>Hint</key><string></string>
  </dict>
  <key>Version</key>
  <dict>
    <key>Display Name</key><string>Version</string>
    <key>Value</key><string>5</string>
    <key>Hint</key><string></string>
  </dict>
  <key>Release</key>
  <dict>
    <key>Display Name</key><string>Release</string>
    <key>Value</key><string>3</string>
    <key>Hint</key><string></string>
  </dict>
  <key>Hostname</key>
  <dict>
    <key>Display Name</key><string>Hostname</string>
    <key>Value</key><string>achilles</string>
    <key>Hint</key><string></string>
  </dict>
.
.
.
</dict>
<key>Workload Management Configuration</key>
<dict>
  <key>Display Name</key><string>Workload Management Configuration</string>
  <key>Service Class Information</key>
  <dict>
    <key>Display Name</key><string>Service Class Information</string>
    <key>1</key>
    <dict>
      <key>Display Name</key><string>1</string>
      <key>Service Class Name</key>
      <dict>
        <key>Display Name</key><string>Service Class Name</string>

```

```

        <key>Value</key><string>SYSDEFAULTSYSTEMCLASS</string>
        <key>Hint</key><string></string>
    </dict>
    <key>Parent Identifier</key>
    <dict>
        <key>Display Name</key><string>Parent Identifier</string>
        <key>Value</key><integer>0</integer>
        <key>Hint</key><string></string>
    </dict>
    <key>Parent Class Name</key>
    <dict>
        <key>Display Name</key><string>Parent Class Name</string>
        <key>Value</key><string></string>
        <key>Hint</key><string></string>
    </dict>
    <key>Creation Time</key>
    <dict>
        <key>Display Name</key><string>Creation Time</string>
        <key>Value</key><string>2008-04-21-15.14.32.956930</string>
        <key>Hint</key><string></string>
    </dict>
    <key>Alter Time</key>
    <dict>
        <key>Display Name</key><string>Alter Time</string>
        <key>Value</key><string>2008-04-21-15.14.32.956930</string>
        <key>Hint</key><string></string>
    </dict>
    <key>Enabled</key>
    <dict>
        <key>Display Name</key><string>Enabled</string>
        <key>Value</key><string>Y</string>
        <key>Hint</key><string></string>
    </dict>
    <key>Agent Priority</key>
    <dict>
        <key>Display Name</key><string>Agent Priority</string>
        <key>Value</key><integer>-32768</integer>
        <key>Hint</key><string></string>
    </dict>
    <key>Prefetcher Priority</key>
    <dict>
        <key>Display Name</key><string>Prefetcher Priority</string>
        <key>Value</key><string> </string>
        <key>Hint</key><string></string>
    </dict>
    .
    :
    .
</dict>
.
.
.
<key>Workload Information</key>
<dict>
    <key>Display Name</key><string>Workload Information</string>
    <key>1</key>
    <dict>
        <key>Display Name</key><string>1</string>
        <key>Workload Name</key>
        <dict>
            <key>Display Name</key><string>Workload Name</string>
            <key>Value</key><string>SYSDEFAULTUSERWORKLOAD</string>
            <key>Hint</key><string></string>
        </dict>
        <key>Evaluation Order</key>
    </dict>

```

```

        <key>Display Name</key><string>Evaluation Order</string>
        <key>Value</key><integer>1</integer>
        <key>Hint</key><string></string>
    </dict>
    <key>Creation Time</key>
    <dict>
        <key>Display Name</key><string>Creation Time</string>
        <key>Value</key><string>2008-04-21-15.14.32.955296</string>
        <key>Hint</key><string></string>
    </dict>
    <key>Alter Time</key>
    <dict>
        <key>Display Name</key><string>Alter Time</string>
        <key>Value</key><string>2008-04-21-15.14.32.955296</string>
        <key>Hint</key><string></string>
    </dict>
    <key>Enabled</key>
    <dict>
        <key>Display Name</key><string>Enabled</string>
        <key>Value</key><string>Y</string>
        <key>Hint</key><string></string>
    </dict>
    <key>Allow Access</key>
    <dict>
        <key>Display Name</key><string>Allow Access</string>
        <key>Value</key><string>Y</string>
        <key>Hint</key><string></string>
    </dict>
    <key>Service Class Name</key>
    <dict>
        <key>Display Name</key><string>Service Class Name</string>
        <key>Value</key><string>SYSDEFAULTSUBCLASS</string>
        <key>Hint</key><string></string>
    </dict>
    <key>Parent Service Class Name</key>
    <dict>
        <key>Display Name</key><string>Parent Service Class Name</string>
        <key>Value</key><string>SYSDEFAULTUSERCLASS</string>
        <key>Hint</key><string></string>
    </dict>
    .
    .
    .
    </dict>
    <key>Hint</key><string></string>
</dict>
</dict>
</dict></dict></dict></plist>

```

例 3: GET\_SYSTEM\_INFO プロシージャを呼び出し、サポートされないロケールを渡します。

```
db2 "call sysproc. get_system_info(1,0,'ja_JP',null,null,?,?)"
```

以下はこの照会の出力例です。

```

Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 1

Parameter Name : MINOR_VERSION
Parameter Value : 0

Parameter Name : XML_OUTPUT
Parameter Value : x'3C3F786D6C20766572.....

```

Parameter Name : XML\_MESSAGE  
Parameter Value : -

Return Status = 0

SQL20461W The procedure "SYSPROC.GET\_SYSTEM\_INFO" returned output in the alternate locale, "en\_US", instead of the locale, "ja\_JP", specified in parameter "3". SQLSTATE=01H57

XML 出力文書には、例 2 に示されているのと同じ内容が含まれます。

例 4: 関数からプロシージャを呼び出します。

```
EXEC SQL BEGIN DECLARE SECTION;
  sqlint16  getSysInfMaj;
  sqlint16  getSysInfMin;

  SQL TYPE IS BLOB(2M) xmlOutput;
  SQL TYPE IS BLOB(2K) xmlOutMessage;
EXEC SQL END DECLARE SECTION;
  getSysInfMaj = 1;
  getSysInfMin = 0;

EXEC SQL CALL SYSPROC.GET_SYSTEM_INFO(
  :getSysInfMaj,
  :getSysInfMin,
  'en_US',
  null,
  null,
  :xmlOutput,
  :xmlOutMessage );
```

## SET\_CONFIG プロシージャ - 構成パラメーターの設定

SET\_CONFIG ストアド・プロシージャは、GET\_CONFIG プロシージャで戻される、データベースおよびデータベース・マネージャー構成パラメーターを更新します。

SET\_CONFIG プロシージャは、構成パラメーターとその値が含まれる入力 XML 文書を受け入れ、その情報を使用して指定の構成パラメーターを更新します。さらに、各構成パラメーターの更新状況を示す出力 XML 文書を戻します。

### 構文

```
►►—SET_CONFIG—(—major_version—,—minor_version—,—requested_locale—,—►►
►—xml_input—,—xml_filter—,—xml_output—,—xml_message—)—►►
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *major\_version*

メジャー文書バージョンを指す、タイプ INTEGER の入出力引数。入力において、この引数は、プロシージャでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする、メジャー文書バージョンを示します (*xml\_input*、*xml\_output*、および *xml\_message* のパラメーターの説明を参照してください)。プロシージャは、指定したバージョンのすべての XML 文書を処理するか、またはバージョンが無効な場合はエラー (+20458) を戻します。出力



において、このパラメーターは、プロシージャーによってサポートされている最も高いメジャー文書バージョンを指定します。サポートされる文書の最も高いバージョンを判別するには、この入力パラメーターおよびその他すべての必須パラメーターに NULL を指定します。

*xml\_input* パラメーターの XML 文書で、文書タイプのメジャー・バージョンのキーが指定されている場合は、そのキーの値は *major\_version* パラメーターで指定される値と同じでなければなりません。同じでない場合、エラー (+20458) が発生します。

#### サポートされているバージョン: 1

##### *minor\_version*

マイナー文書バージョンを指す、タイプ INTEGER の入出力引数。入力において、この引数は、プロシージャーでパラメーターとして渡される XML 文書に対して呼び出し元がサポートする、マイナー文書バージョンを示します (*xml\_input*、*xml\_output*、および *xml\_message* のパラメーターの説明を参照してください)。プロシージャーは、指定したバージョンのすべての XML 文書进行处理するか、またはバージョンが無効な場合はエラーを戻します。出力において、このパラメーターは、サポートされている最も高いメジャー・バージョンでサポートされている、最も高いマイナー文書バージョンを指します。サポートされる文書の最も高いバージョンを判別するには、この入力パラメーターおよびその他すべての必須パラメーターに NULL を指定します。

*xml\_input* パラメーターの XML 文書で、文書タイプのマイナー・バージョンのキーが指定されている場合は、そのキーの値は *minor\_version* パラメーターで指定される値と同じでなければなりません。同じでない場合、エラー (+20458) が発生します。

#### サポートされているバージョン: 0

##### *requested\_locale*

ロケールを指定する、タイプ VARCHAR(33) の入力引数。指定した言語がサーバーでサポートされている場合は、変換された内容が *xml\_output* および *xml\_message* パラメーターに戻されます。サポートされていない場合は、内容はデフォルトの言語に戻されます。ロケールから使用されるのは、言語 (および場合によっては地域情報) のみです。ロケールは数字のフォーマット設定に使用されたり、文書エンコードに影響を与えたりすることはありません。例えば、キー名と値は変換されません。XML 出力および XML メッセージ文書で変換される部分は、各項目のヒントのテキスト、表示名、および表示単位のみです。呼び出し元は常に、要求された言語と XML 出力文書で使用されている言語を比較する必要があります (XML 出力文書の文書ロケール項目を参照してください)。

現時点でサポートされている *requested\_locale* の値は en\_US のみです。

##### *xml\_input*

プロシージャーの入力値が含まれる XML 入力文書 (UTF-8 でエンコードされている) を指定する、タイプ BLOB(32MB) の入力引数。

このプロシージャーでは、XML 入力文書にデータベースおよびデータベース・マネージャー構成設定が含まれます。

##### *xml\_filter*

有効な XPath 照会ストリングを指定する、タイプ BLOB(4K) の入力引数。

XML 出力文書から単一値を検索する場合、フィルターを使用します。詳しくは、XPath フィルター操作について記述しているトピックを参照してください。

次の例では、XML 出力文書から、特定の構成パラメーター設定の値を選択しています (`/plist/dict/key[.="Database Manager Configuration Parameter Settings"]/following-sibling::dict[1]/key[3]/following-sibling::dict[1]/dict[1]/key[.="Value"]/following-sibling::string[1]`)。キーの後に兄弟が指定されていないと、エラーが戻されます。

#### *xml\_output*

UTF-8 の完全な XML 出力文書を戻す、タイプ BLOB(32MB) の出力パラメーターを戻します。フィルターが指定されている場合、このパラメーターはストリング値を戻します。ストアド・プロシージャが完全な出力文書を戻すことができない場合 (例えば、処理エラーが発生して SQL 警告やエラーが出される場合など)、このパラメーターは NULL に設定されます。

このプロシージャが *complete* モードで動作する場合、このパラメーターによって戻される XML 文書には、サーバーで設定されている現行の構成値が含まれます。この文書を変更して、*xml\_input* パラメーターとしてプロシージャに戻すことができます。このアプローチにより、有効な XML 入力文書を作成するためのプログラマ的な方法が提供されます。

#### *xml\_message*

SQL 警告状態についての詳細情報を提供する Data Server Message タイプの完全な XML 出力文書を UTF-8 で戻す、タイプ BLOB(64K) の出力パラメーター。プロシージャの呼び出しにより SQL 警告が出され、XML メッセージ出力文書に追加情報が戻されることを警告メッセージが示す場合に、この文書が戻されます。追加情報が戻されることを警告メッセージが示していない場合は、このパラメーターは NULL に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: プロシージャの最新バージョンを戻します。

```
db2 "call sysproc.set_config (null,null,null,null,null,?,?)"
```

以下はこの照会の出力例です。

```
Value of output parameters
-----
Parameter Name : MAJOR_VERSION
```

Parameter Value : 1

Parameter Name : MINOR\_VERSION  
Parameter Value : 0

Parameter Name : XML\_OUTPUT  
Parameter Value : -

Parameter Name : XML\_MESSAGE  
Parameter Value : -

Return Status = 0

例 2: いくつかのデータベースおよびデータベース・マネージャー構成パラメーターを更新する `setconfig.sql` というスクリプトを実行します。

`setconfig.sql`:

```
call sysproc.set_config(1,0,'en_US',blob('
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key><string>Data Server Set Configuration Input</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Document Locale</key><string>en_US</string>
  <key>Database Manager Configuration Parameter Settings</key>
  <dict>
    <key>diaglevel</key><dict><key>Parameter Value</key>
      <dict>
        <key>Value</key><string>4</string>
      </dict>
    </dict>
  <key>fcm_num_buffers</key>
  <dict>
    <key>Parameter Value</key>
    <dict>
      <key>Value</key><string>4096</string>
    </dict>
    <key>Value Flags</key>
    <dict>
      <key>Value</key><string>MANUAL</string>
    </dict>
  </dict>
  <key>instance_memory</key>
  <dict>
    <key>Deferred Value</key>
    <dict>
      <key>Value</key><string>7424</string>
    </dict>
    <key>Deferred Value Flags</key>
    <dict>
      <key>Value</key><string>AUTOMATIC</string>
    </dict>
  </dict>
</dict>
</dict>
<key>Database Partition</key>
<dict>
  <key>All</key>
  <dict>
    <key>Database Configuration Parameter Settings</key>
    <dict>
      <key>avg_appls</key>
      <dict>
        <key>Parameter Value</key>
        <dict>
          <key>Value</key><string>2</string>
        </dict>
      </dict>
    </dict>
  </dict>
</dict>
</plist>
')
```

```

        </dict>
        <key>Value Flags</key>
        <dict>
          <key>Value</key><string>AUTOMATIC</string>
        </dict>
      </dict>
    <key>database_memory</key>
    <dict>
      <key>Deferred Value</key>
      <dict>
        <key>Value</key><string>2</string>
      </dict>
      <key>Deferred Value Flags</key>
      <dict>
        <key>Value</key><string>MANUAL</string>
      </dict>
    </dict>
  </dict>
</dict>
</plist>'), null, ?,?)@

```

以下はこの照会の出力例です。

```

Value of output parameters
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 1

Parameter Name : MINOR_VERSION
Parameter Value : 0

Parameter Name : XML_OUTPUT
Parameter Value : x'3C3F78...'

Parameter Name : XML_MESSAGE
Parameter Value : -

Return Status = 0

```

出力 XML 文書には、次のような内容が含まれます。

```

<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key>
  <string>Data Server Set Configuration Output</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Data Server Product Name</key><string>QDB2/AIX64</string>
  <key>Data Server Product Version</key><string>9.7.0.0</string>
  <key>Data Server Major Version</key><integer>9</integer>
  <key>Data Server Minor Version</key><integer>7</integer>
  <key>Data Server Platform</key><string>AIX 64BIT</string>
  <key>Document Locale</key><string>en_US</string>
  <key>Database Manager Configuration Parameter Settings</key>
</dict>
  <key>Display Name</key>
  <string>Database Manager Configuration Parameter Settings</string>
  <key>diaglevel</key>
  <dict>
    <key>Display Name</key><string>diaglevel</string>
    <key>Parameter Value</key>
    <dict>
      <key>Display Name</key><string>Parameter Value</string>
      <key>Value</key><string>4</string>
    </dict>
  </dict>
</plist>

```

```

</dict>
<key>Parameter Update Status</key>
<dict>
  <key>Display Name</key><string>Parameter Update Status</string>
  <key>SQLCODE</key>
  <dict>
    <key>Display Name</key><string>SQLCODE</string>
    <key>Value</key><integer>0</integer>
  </dict>
<key>Message Tokens</key>
<dict>
  <key>Display Name</key><string>Message Tokens</string>
  <key>Value</key><array><string></string></array>
</dict>
<key>SQLSTATE</key>
<dict>
  <key>Display Name</key><string>SQLSTATE</string>
  <key>Value</key><string></string>
</dict>
</dict>
</dict>
<key>fcm_num_buffers</key>
<dict>
  <key>Display Name</key><string>fcm_num_buffers</string>
  <key>Parameter Value</key>
  <dict>
    <key>Display Name</key><string>Parameter Value</string>
    <key>Value</key><string>4096</string>
  </dict>
  <key>Value Flags</key>
  <dict>
    <key>Display Name</key><string>Value Flags</string>
    <key>Value</key><string>MANUAL</string>
  </dict>
<key>Parameter Update Status</key>
<dict>
  <key>Display Name</key><string>Parameter Update Status</string>
  <key>SQLCODE</key><dict>
    <key>Display Name</key><string>SQLCODE</string>
    <key>Value</key> <integer>0</integer>
  </dict>
  <key>Message Tokens</key>
  <dict>
    <key>Display Name</key><string>Message Tokens</string>
    <key>Value</key><array><string></string></array>
  </dict>
  <key>SQLSTATE</key>
  <dict>
    <key>Display Name</key><string>SQLSTATE</string>
    <key>Value</key><string></string>
  </dict>
</dict>
</dict>
<key>instance_memory</key>
<dict>
  <key>Display Name</key><string>instance_memory</string>
  <key>Deferred Value</key>
  <dict>
    <key>Display Name</key><string>Deferred Value</string>
    <key>Value</key><string>7424</string>
  </dict>
  <key>Deferred Value Flags</key>
  <dict>
    <key>Display Name</key><string>Deferred Value Flags</string>
    <key>Value</key><string>AUTOMATIC</string>
  </dict>
<key>Parameter Update Status</key>

```

```

<dict>
  <key>Display Name</key><string>Parameter Update Status</string>
  <key>SQLCODE</key>
  <dict>
    <key>Display Name</key><string>SQLCODE</string>
    <key>Value</key><integer>0</integer>
  </dict>
</dict>
  <key>Message Tokens</key>
  <dict>
    <key>Display Name</key><string>Message Tokens</string>
    <key>Value</key><array><string></string></array>
  </dict>
  <key>SQLSTATE</key>
  <dict>
    <key>Display Name</key><string>SQLSTATE</string>
    <key>Value</key><string></string>
  </dict>
</dict>
</dict>
</dict>
<key>Database Partition</key>
<dict>
  <key>Display Name</key><string>Database Partition</string>
  <key>All</key>
  <dict>
    <key>Display Name</key><string>All</string>
  <key>Database Configuration Parameter Settings</key>
  <dict>
    <key>Display Name</key>
    <string>Database Configuration Parameter Settings</string>
    <key>avg_appls</key>
    <dict>
      <key>Display Name</key><string>avg_appls</string>
      <key>Parameter Value</key>
      <dict>
        <key>Display Name</key><string>Parameter Value</string>
        <key>Value</key><string>2</string>
      </dict>
    </dict>
    <key>Value Flags</key>
    <dict>
      <key>Display Name</key><string>Value Flags</string>
      <key>Value</key><string>AUTOMATIC</string>
    </dict>
  <key>Parameter Update Status</key>
  <dict>
    <key>Display Name</key><string>Parameter Update Status</string>
    <key>Update Coverage</key>
    <dict>
      <key>Display Name</key><string>Update Coverage</string>
      <key>Value</key><string>Complete</string>
    </dict>
    <key>SQLCODE</key>
    <dict>
      <key>Display Name</key><string>SQLCODE</string>
      <key>Value</key><integer>0</integer>
    </dict>
    <key>Message Tokens</key>
    <dict>
      <key>Display Name</key><string>Message Tokens</string>
      <key>Value</key><array><string></string> </array>
    </dict>
    <key>SQLSTATE</key>
    <dict>
      <key>Display Name</key><string>SQLSTATE</string>
      <key>Value</key><string></string>
    </dict>
  </dict>
</dict>

```



```

        <key>Value</key><string>4096</string>
      </dict>
    <key>Value Flags</key>
    <dict>
      <key>Value</key><string>MANUAL</string>
    </dict>
  </dict>
<key>instance_memory</key>
<dict>
  <key>Deferred Value</key>
  <dict>
    <key>Value</key><string>7424</string>
  </dict>
  <key>Deferred Value Flags</key>
  <dict>
    <key>Value</key><string>AUTOMATIC</string>
  </dict>
</dict>
</dict>
<key>Database Partition</key>
<dict>
  <key>All</key>
  <dict>
    <key>Database Configuration Parameter Settings</key>
    <dict>
      <key>avg_appls</key>
      <dict>
        <key>Parameter Value</key>
        <dict>
          <key>Value</key><string>2</string>
        </dict>
        <key>Value Flags</key>
        <dict>
          <key>Value</key><string>AUTOMATIC</string>
        </dict>
      </dict>
      <key>database_memory</key>
      <dict>
        <key>Deferred Value</key>
        <dict>
          <key>Value</key><string>2</string>
        </dict>
        <key>Deferred Value Flags</key>
        <dict>
          <key>Value</key><string>MANUAL</string>
        </dict>
      </dict>
    </dict>
  </dict>
</dict>
</dict>
</plist>'),
blob('/plist/dict/key[.="Database Manager Configuration Parameter Settings"]
/following-sibling::dict[1]/key[3]
/following-sibling::dict[1]/dict[1]/key[.="Value"]
/following-sibling::string[1]'),?,?)"
```

以下はこの照会の出力例です。

Value of output parameters

```
-----
Parameter Name : MAJOR_VERSION
Parameter Value : 1
```

```
Parameter Name : MINOR_VERSION
Parameter Value : 0
```



Parameter Name : XML\_OUTPUT  
Parameter Value : x'34303936'

Parameter Name : XML\_MESSAGE  
Parameter Value : -

Return Status = 0

*xml\_output* に次の値が戻されます。

4096

例 4: 関数からプロシーチャーを呼び出します。

```
EXEC SQL BEGIN DECLARE SECTION;
sqlint16 getconfigMaj;
sqlint16 getconfigMin;

SQL TYPE IS BLOB(2M) xmlOutput;
SQL TYPE IS BLOB(2K) xmlOutMessage;
EXEC SQL END DECLARE SECTION;
getconfigMaj = 1;
getconfigMin = 0;

EXEC SQL CALL SYSPROC.SET_CONFIG(
: getconfigMaj,
: getconfigMin,
' en_US',
BLOB(' blob('
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Document Type Name</key>
  <string>Data Server Set Configuration Input</string>
  <key>Document Type Major Version</key><integer>1</integer>
  <key>Document Type Minor Version</key><integer>0</integer>
  <key>Document Locale</key><string>en_US</string>
  <key>Database Manager Configuration Parameter Settings</key>
  <dict>
    <key>diaglevel</key><dict><key>Parameter Value</key>
    <dict>
      <key>Value</key><string>4</string>
    </dict>
  </dict>
  <key>fcm_num_buffers</key>
  <dict>
    <key>Parameter Value</key>
    <dict>
      <key>Value</key><string>4096</string>
    </dict>
    <key>Value Flags</key>
    <dict>
      <key>Value</key><string>MANUAL</string>
    </dict>
  </dict>
  <key>instance_memory</key>
  <dict>
    <key>Deferred Value</key>
    <dict>
      <key>Value</key><string>7424</string>
    </dict>
    <key>Deferred Value Flags</key>
    <dict>
      <key>Value</key><string>AUTOMATIC</string>
    </dict>
  </dict>
</dict>
</dict>
```

```

<key>Database Partition</key>
<dict>
  <key>All</key>
  <dict>
    <key>Database Configuration Parameter Settings</key>
    <dict>
      <key>avg_appls</key>
      <dict>
        <key>Parameter Value</key>
        <dict>
          <key>Value</key><string>2</string>
        </dict>
        <key>Value Flags</key>
        <dict>
          <key>Value</key><string>AUTOMATIC</string>
        </dict>
      </dict>
    <key>database_memory</key>
    <dict>
      <key>Deferred Value</key>
      <dict>
        <key>Value</key><string>2</string>
      </dict>
      <key>Deferred Value Flags</key>
      <dict>
        <key>Value</key><string>MANUAL</string>
      </dict>
    </dict>
  </dict>
</dict>
</dict>
</dict>
</plist>'),
null,
:xmlOutput,
:xmlOutMessage );

```

## 構成ルーチンおよびビュー

### DBCFCG 管理ビューおよび DB\_GET\_CFG 表関数 - データベース構成パラメーター情報の取得

DBCFCG 管理ビューおよび DB\_GET\_CFG 表関数は、現在接続中のデータベースのデータベース構成パラメーター情報を、すべてのデータベース・メンバーについて取得します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『DBCFCG 管理ビュー』
- 389 ページの『DB\_GET\_CFG 表関数』

#### DBCFCG 管理ビュー

スキーマは SYSIBMADM です。

#### 許可

以下のいずれかの権限が必要です。

- DBCFCG 管理ビューに対する SELECT 特権

- DBCFG 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

例 1: メモリーに保管されている、すべてのデータベース・メンバーに関するデータベース構成内の自動保守の設定を検索します。

```
SELECT DBPARTITIONNUM, NAME, VALUE FROM SYSIBMADM.DBCFG WHERE NAME LIKE 'auto_%'
```

以下はこの照会の出力例です。

DBPARTITIONNUM	NAME	VALUE
0	auto_maint	OFF
0	auto_db_backup	OFF
0	auto_tbl_maint	OFF
0	auto_runstats	OFF
0	auto_stats_prof	OFF
0	auto_prof_upd	OFF
0	auto_reorg	OFF
0	autorestart	ON

8 record(s) selected.

例 2: ディスクに保管されている、すべてのデータベース・メンバーのデータベース構成パラメーター値すべてを検索します。

```
SELECT NAME, DEFERRED_VALUE, DBPARTITIONNUM FROM SYSIBMADM.DBCFG
```

以下はこの照会の出力例です。

NAME	DEFERRED_VALUE	DBPARTITIONNUM
app_ctl_heap_sz	128	0
appgroup_mem_sz	30000	0
applheapsz	256	0
archretrydelay	20	0
...		
autorestart	ON	0
avg_appls	1	0
blk_log_dsk_ful	NO	0
catalogcache_sz	-1	0
...		

## DB\_GET\_CFG 表関数

### 構文

```
DB_GET_CFG ( member )
```

スキーマは SYSPROC です。

## 表関数パラメーター

### member

現在接続されているデータベースと同じインスタンス内のメンバーの番号を指定する、タイプ INTEGER のオプション入力引数。現行のメンバーには -1、すべてのメンバーには -2 を指定します。 NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

以下のいずれかの権限が必要です。

- DB\_GET\_CFG 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

DB2 pureScale 環境で、メモリーに保管されているデータベース構成内の自動保守の設定を、すべてのメンバーについて取得します。

```
SELECT NAME, VARCHAR(VALUE, 20) AS VALUE, MEMBER,
       DBPARTITIONNUM FROM TABLE(SYSPROC.DB_GET_CFG(-2))
WHERE NAME LIKE 'auto_%' ORDER BY NAME, MEMBER
```

以下はこの照会の出力例です。

NAME	VALUE	MEMBER	DBPARTITIONNUM
auto_db_backup	OFF	0	0
auto_db_backup	OFF	1	0
auto_db_backup	OFF	2	0
auto_del_rec_obj	OFF	0	0
auto_del_rec_obj	OFF	1	0
auto_del_rec_obj	OFF	2	0
auto_maint	ON	0	0
auto_maint	ON	1	0
auto_maint	ON	2	0
auto_prof_upd	OFF	0	0
auto_prof_upd	OFF	1	0
auto_prof_upd	OFF	2	0
auto_reorg	OFF	0	0
auto_reorg	OFF	1	0
auto_reorg	OFF	2	0
auto_reval	DEFERRED	0	0
auto_reval	DEFERRED	1	0
auto_reval	DEFERRED	2	0
auto_runstats	ON	0	0
auto_runstats	ON	1	0
auto_runstats	ON	2	0
auto_stats_prof	OFF	0	0
auto_stats_prof	OFF	1	0
auto_stats_prof	OFF	2	0
auto_stats_views	OFF	0	0
auto_stats_views	OFF	1	0

```

auto_stats_views          OFF          2          0
auto_stmt_stats           ON           0          0
auto_stmt_stats           ON           1          0
auto_stmt_stats           ON           2          0
auto_tbl_maint            ON           0          0
auto_tbl_maint            ON           1          0
auto_tbl_maint            ON           2          0
autorestart               ON           0          0
autorestart               ON           1          0
autorestart               ON           2          0

```

36 record(s) selected.

## 戻される情報

表 86. DBCFG 管理ビューおよび DB\_GET\_CFG 表関数によって戻される情報

列名	データ・タイプ	説明
NAME	VARCHAR(32)	構成パラメーター名。
VALUE	VARCHAR(1024)	メモリー内に保管されている構成パラメーターの現行値。
VALUE_FLAGS	VARCHAR(10)	構成パラメーターの現行値に固有の情報を指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> <li>• NONE - 追加情報なし</li> <li>• AUTOMATIC - 構成パラメーターが自動的に設定されている</li> </ul>
DEFERRED_VALUE	VARCHAR(1024)	ディスク上の構成パラメーター値。データベース構成パラメーターによっては、データベースを再活動化しないと変更が有効にならないことがあります。その場合に、すべてのアプリケーションをまずデータベースから切断する必要があります。(データベースがアクティブ化されていた場合には、非アクティブにしてから再アクティブ化しなければなりません。)変更は次にデータベースに接続したときに有効になります。
DEFERRED_VALUE_FLAGS	VARCHAR(10)	構成パラメーターの据え置き値に固有の情報を指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> <li>• NONE - 追加情報なし</li> <li>• AUTOMATIC - 構成パラメーターが自動的に設定されている</li> </ul>
DATATYPE	VARCHAR(128)	構成パラメーター・データ・タイプ。

表 86. DBCFG 管理ビューおよび DB\_GET\_CFG 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## DBMCFG 管理ビュー - データベース・マネージャー構成パラメーター情報の検索

DBMCFG 管理ビューは、メモリー内の値およびディスクに保管された値を含む、データベース・マネージャー構成パラメーター情報を検索します。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- DBMCFG 管理ビューに対する SELECT 特権
- DBMCFG 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

例 1: ディスクに保管されているすべてのデータベース・マネージャー構成パラメーターの値を検索します。

```
SELECT NAME, DEFERRED_VALUE FROM SYSIBMADM.DBMCFG
```

以下はこの照会の出力例です。

```

NAME                                DEFERRED_VALUE
-----
agent_stack_sz                       0
agentpri                              -1
alternate_auth_enc                   AES_ONLY
aslheapsz                             15
audit_buf_sz                          0
authentication                       SERVER
catalog_noauth                       YES
clnt_krb_plugin
...
comm_bandwidth                       0.000000e+00
conn_elapse                           0
cpuspeed                              4.000000e-05
dft_account_str

```

```

dft_mon_bufpool          OFF
...
dft_mon_timestamp       ON
dft_mon_uow             OFF
...
jdk_path                 /wsdb/v91/bldsupp/AIX5L
...
ssl_svcname             22711
ssl_svr_keydb           /GSKit/Keystore/key.kdb
ssl_svr_label
ssl_svr_stash           /GSKit/Keystore/key.sth

```

例 2: データベース・マネージャ構成パラメータ値すべてを検索します。

```
SELECT * FROM SYSIBMADM.DBMCFG
```

以下はこの照会の出力例です。

NAME	VALUE	VALUE_FLAGS	...
agent_stack_sz	0	NONE	...
agentpri	-1	NONE	...
alternate_auth_enc	NOT_SPECIFIED	NONE	...
aslheapsz	15	NONE	...
audit_buf_sz	0	NONE	...
authentication	SERVER	NONE	...
catalog_noauth	YES	NONE	...
clnt_krb_plugin		NONE	...
clnt_pw_plugin		NONE	...
comm_bandwidth	0.000000e+00	NONE	...
conn_elapse	0	NONE	...
cpuspeed	4.000000e-05	NONE	...
dft_account_str		NONE	...
dft_mon_bufpool	OFF	NONE	...
dft_mon_lock	OFF	NONE	...
dft_mon_sort	OFF	NONE	...
dft_mon_stmt	OFF	NONE	...
dft_mon_table	OFF	NONE	...
dft_mon_timestamp	ON	NONE	...
dft_mon_uow	OFF	NONE	...
dftdbpath	/home/userb	NONE	...
diaglevel	3	NONE	...
diagpath	/home/userc/tmp/ \$m	NONE	...
diagpath_resolved	/home/userc/tmp/DIAG000	NONE	...
dir_cache	YES	NONE	...
discover	SEARCH	NONE	...
discover_inst	ENABLE	NONE	...
fcm_num_anchors	0	AUTOMATIC	...
fcm_num_buffers	0	AUTOMATIC	...
fcm_num_connect	0	AUTOMATIC	...

この照会の出力 (続き)。

... DEFERRED_VALUE	DEFERRED_VALUE_FLAGS	DATATYPE
...	...	...
... 0	NONE	INTEGER
... -1	NONE	INTEGER
... AES_ONLY	NONE	VARCHAR(32)
... 15	NONE	BIGINT
... 0	NONE	BIGINT
... SERVER	NONE	VARCHAR(32)
... YES	NONE	VARCHAR(3)
...	NONE	VARCHAR(32)
...	NONE	VARCHAR(32)
... 0.000000e+00	NONE	REAL
... 0	NONE	INTEGER
... 4.000000e-05	NONE	REAL

```

...                               NONE                               VARCHAR(25)
... OFF                           NONE                               VARCHAR(3)
... OFF                           NONE                               VARCHAR(3)
... OFF                           NONE                               VARCHAR(3)
... OFF                           NONE                               VARCHAR(3)
... OFF                           NONE                               VARCHAR(3)
... ON                             NONE                               VARCHAR(3)
... OFF                           NONE                               VARCHAR(3)
... /home/userb                   NONE                               VARCHAR(215)
... 3                              NONE                               INTEGER
... /home/userc/tmp $m            NONE                               VARCHAR(215)
... /home/userc/tmp/DIAG000      NONE                               VARCHAR(215)
... YES                           NONE                               VARCHAR(3)
... SEARCH                        NONE                               VARCHAR(8)
... ENABLE                        NONE                               VARCHAR(8)
... 0                              AUTOMATIC                         BIGINT
... 512                           AUTOMATIC                         BIGINT
... 0                              AUTOMATIC                         BIGINT
...

```

## 戻される情報

表 87. DBMCFG 管理ビューによって戻される情報

列名	データ・タイプ	説明
NAME	VARCHAR(32)	構成パラメーター名。
VALUE	VARCHAR(256)	メモリー内に保管されている構成パラメーターの現行値。
VALUE_FLAGS	VARCHAR(10)	構成パラメーターの現行値に固有の情報を指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> <li>• NONE - 追加情報なし</li> <li>• AUTOMATIC - 構成パラメーターが自動的に設定されている</li> </ul>
DEFERRED_VALUE	VARCHAR(256)	ディスク上の構成パラメーター値。データベース・マネージャ構成パラメーターによっては、データベース・マネージャを停止 ( <b>db2stop</b> ) してから再始動 ( <b>db2start</b> ) しないと、この値が有効になりません。
DEFERRED_VALUE_FLAGS	VARCHAR(10)	構成パラメーターの据え置き値に固有の情報を指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> <li>• NONE - 追加情報なし</li> <li>• AUTOMATIC - 構成パラメーターが自動的に設定されている</li> </ul>
DATATYPE	VARCHAR(128)	構成パラメーター・データ・タイプ。



## DB2 pureScale インスタンス情報ルーチンおよびビュー

DB2 pureScale 環境では、特定のルーチンが、DB2 pureScale インスタンスやそのメンバーの状態に関する情報を提供します。

### DB\_MEMBERS 表関数

DB\_MEMBERS 表関数は、DB2 pureScale インスタンスに関する基本的なメンバー情報を返します。

### 構文

▶▶—DB\_MEMBERS—(—)—————▶▶

スキーマは SYSPROC です。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 表関数パラメーター

関数には入力パラメーターはありません。

### 例

インスタンス情報の取得:

```
$ db2 "select * from table(SYSPROC.DB_MEMBERS()) as members"
```

以下はこの照会の出力例です。

MEMBER_NUMBER	HOST_NAME	PARTITION_NUMBER	MEMBER_TYPE
0	member1.mycompany.com	0	D
1	member2.mycompany.com	0	D
2	member3.mycompany.com	0	D
128	ca1.mycompany.com	0	C
129	ca2.mycompany.com	0	C

PORT_NUMBER	SWITCH_NAME	STATUS
0	-	-
0	-	-
0	-	-

```

0 - -
0 - -

```

5 record(s) selected.

## 戻される情報

表 88. *DB\_MEMBERS* 表関数によって戻される情報

列名	データ・タイプ	説明
MEMBER_NUMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
HOST_NAME	VARCHAR(256)	host_name - ホスト名モニター・エレメント
PARTITION_NUMBER	SMALLINT	partition_number - パーティション番号モニター・エレメント
MEMBER_TYPE	CHAR(1)	メンバーのタイプ: <ul style="list-style-type: none"> <li>• C - CF メンバー</li> <li>• D - データベース・メンバー</li> </ul>
PORT_NUMBER	SMALLINT	port_number - ポート番号
SWITCH_NAME	VARCHAR(128)	ホスト上のメンバーのスイッチ名
STATUS	SMALLINT	将来の利用のために予約済み

## DB2\_CLUSTER\_HOST\_STATE 管理ビューおよび DB2\_GET\_CLUSTER\_HOST\_STATE 表関数 - ホストについての情報の取得

DB2\_CLUSTER\_HOST\_STATE 管理ビューおよび

DB2\_GET\_CLUSTER\_HOST\_STATE 表関数は、DB2 pureScale インスタンスに含まれるホストについての情報を取得します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『DB2\_CLUSTER\_HOST\_STATE 管理ビュー』
- 397 ページの『DB2\_GET\_CLUSTER\_HOST\_STATE 表関数』

### DB2\_CLUSTER\_HOST\_STATE 管理ビュー

DB2\_CLUSTER\_HOST\_STATE 管理ビューは、DB2 pureScale インスタンスに含まれるホストを戻します。固有のホストのリストに加えて、それらに関連したインスタンス状態情報を取得できます。

スキーマは SYSIBMADM です。

戻すことができる情報の全リストについては、DB2\_CLUSTER\_HOST\_STATE 管理ビューおよび DB2\_GET\_CLUSTER\_HOST\_STATE で戻される情報の表を参照してください。

## 許可

以下のいずれかの権限が必要です。

- DB2\_CLUSTER\_HOST\_STATE 管理ビューに対する SELECT 特権
- DB2\_CLUSTER\_HOST\_STATE 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## 例

次の例は、DB2\_CLUSTER\_HOST\_STATE 管理ビューを使用して、4 つのメンバーと 2 つの CF が含まれる DB2 pureScale インスタンスに関する情報を取得します。

```
SELECT * FROM SYSIBMADM.DB2_CLUSTER_HOST_STATE
```

この照会からの出力は、次のようになります。

HOSTNAME	STATE	INSTANCE_STOPPED	ALERT
so1	ACTIVE	NO	NO
so2	ACTIVE	NO	NO
so3	ACTIVE	NO	NO
so4	ACTIVE	NO	NO
so5	ACTIVE	NO	NO
so6	ACTIVE	NO	NO

6 record(s) selected.

## DB2\_GET\_CLUSTER\_HOST\_STATE 表関数

DB2\_GET\_CLUSTER\_HOST\_STATE 表関数は DB2\_CLUSTER\_HOST\_STATE 管理ビューと同じ情報を戻しますが、ホスト名の指定が可能です。

戻すことができる情報の全リストについては、DB2\_CLUSTER\_HOST\_STATE 管理ビューおよび DB2\_GET\_CLUSTER\_HOST\_STATE で戻される情報の表を参照してください。

## 構文

►►—DB2\_GET\_CLUSTER\_HOST\_STATE—(—*hostname*—)—————►►

スキーマは SYSPROC です。

## 表関数パラメーター

*hostname*

レコードが戻されるホスト名を短形式または長形式で指定する、タイプ VARCHAR(255) のオプション入力引数。代わりに IP アドレスを指定した場合、レコードは戻されません。このパラメーターが NULL または空ストリング (" ) である場合、すべてのレコードが戻されます。

## 許可

以下のいずれかの権限が必要です。

- DB2\_GET\_CLUSTER\_HOST\_STATE 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

次の例は、DB2\_GET\_CLUSTER\_HOST\_STATE 表関数を使用して、4 つのメンバーと 2 つの CF が含まれる DB2 pureScale インスタンス内の so3 というホストについての情報を取得します。

```
SELECT * FROM TABLE(DB2_GET_CLUSTER_HOST_STATE('so3')) as T
```

以下はこの照会の出力例です。

```
HOSTNAME STATE                               INSTANCE_STOPPED ALERT
-----
so3      ACTIVE                               NO                NO
```

1 record(s) selected.

## 戻される情報

表 89. DB2\_CLUSTER\_HOST\_STATE 管理ビューおよび DB2\_GET\_CLUSTER\_HOST\_STATE で戻される情報

列名	データ・タイプ	説明
HOSTNAME	VARCHAR(255)	hostname - ホスト名 : モニター・エレメント
STATE	VARCHAR(32)	ホストの状態情報。 <ul style="list-style-type: none"><li>• ACTIVE - ホストは DB2 データベース・マネージャーによって使用可能です。</li><li>• INACTIVE - ホストは DB2 データベース・マネージャーによって使用可能ではありません。</li><li>• NULL - ホスト状態は該当しません (例えば、パーティション・データベース環境にある場合)。</li></ul>
INSTANCE_STOPPED	VARCHAR(8)	インスタンスがホストで停止しているかどうか。 <ul style="list-style-type: none"><li>• YES - インスタンスはホストで停止しています</li><li>• NO - インスタンスはホストで停止していません</li><li>• NULL - 該当しません (例えば、パーティション・データベース環境にある場合)</li></ul>
ALERT	VARCHAR(8)	ホストに関するアラートの情報。 <ul style="list-style-type: none"><li>• YES - ホストに関するアラートがあります</li><li>• NO - ホストに関するアラートはありません</li><li>• NULL - 該当しません (例えば、パーティション・データベース環境にある場合)</li></ul>

## DB2\_INSTANCE\_ALERTS ビュー - アラートに関する情報の取得

The DB2\_INSTANCE\_ALERTS 管理ビューは、DB2 pureScale インスタンス内のアラートについての情報を提供します。

スキーマは SYSIBMADM です。

### 許可

以下の許可を必要とします。

- DB2\_INSTANCE\_ALERTS 管理ビューに対する SELECT 特権

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

次の例は、DB2\_INSTANCE\_ALERTS 管理ビューを使用して取得できる情報の種類を示しています。

```
SELECT * FROM SYSIBMADM.DB2_INSTANCE_ALERTS
```

以下はこの照会の出力例です。読みやすくするために、列を別々の行に配置しています。

MESSAGE

```
-----  
Restart light failed for member 0 on host(s) hostC. Check the db2diag.log for  
messages concerning a restart light or database crash recovery failure on these  
hosts for member 0. See the DB2 Information Center for more details.
```

ACTION

```
-----  
This alert must be cleared manually with the command:  
db2cluster -clear -alert -member 0
```

IMPACT

```
-----  
Restart light will not succeed for member 0 on the hosts listed until this  
alert has been cleared.
```

### 戻される情報

表 90. DB2\_INSTANCE\_ALERTS 管理ビューによって戻される情報

列名	データ・タイプ	説明またはモニター・エレメント
MESSAGE	VARCHAR(32672)	message - コントロール表メッセージ・モニター・エレメント
ACTION	VARCHAR(32672)	アラートをクリアするために必要な処置。
IMPACT	VARCHAR(32672)	アラートがクリアされない場合の DB2 pureScale インスタンスへの影響。

## DB2\_MEMBER および DB2\_CF 管理ビューと DB2\_GET\_INSTANCE\_INFO 表関数

DB2\_MEMBER および DB2\_CF 管理ビューと DB2\_GET\_INSTANCE\_INFO 表関数は、DB2 pureScale インスタンスのメンバーおよび CF の状態情報 (該当する場合) などの情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『DB2\_MEMBER および DB2\_CF 管理ビュー』
- 401 ページの『DB2\_GET\_INSTANCE\_INFO 表関数』

### DB2\_MEMBER および DB2\_CF 管理ビュー

DB2\_MEMBER 管理ビューは、メンバーが現在実行されているマシン名、その状態、アラートが存在するかどうか、および内部データベース通信用の高速相互接続の名前などの情報を戻します。

DB2\_CF 管理ビューは、クラスター・キャッシング・ファシリティー (CF と呼ばれる) について同様の情報を戻します。

スキーマは SYSIBMADM です。

戻すことができる情報の全リストについては、DB2\_MEMBER 管理ビューおよび DB2\_CF 管理ビューと DB2\_GET\_INSTANCE\_INFO で戻される情報の表を参照してください。

### 許可

以下のいずれかの権限が必要です。

- 管理ビューに対する SELECT 特権
- 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### 例

次の例は、DB2\_MEMBER 管理ビューを使用して、DB2 インスタンス内のすべてのメンバーの状況を表示します。これは、以下の種類のインスタンスに使用できません。

- DB2 pureScale インスタンス
- DB2 高可用性災害時リカバリー (HADR) フィーチャーを使用するパーティションまたは非パーティション・データベース・インスタンス。

DB2 インスタンス内のすべてのメンバーの状況を表示するには、次の照会を実行します。

```
SELECT * FROM SYSIBMADM.DB2_MEMBER
```

以下はこの照会の出力例です。

ID	HOME_HOST	CURRENT_HOST	STATE	ALERT
0	so1	so1	STARTED	NO
2	so2	so2	STARTED	NO
4	so3	so3	STARTED	NO

3 record(s) selected.

次の例は、DB2\_CF 管理ビューを使用して、DB2 インスタンス内のすべての CF の状況を表示します。これは、DB2 pureScale インスタンスにのみ使用できます。

DB2 インスタンス内のすべてのクラスター・キャッシング・ファシリティの状況を表示するには、次の照会を実行します。

```
SELECT * FROM SYSIBMADM.DB2_CF
```

以下はこの照会の出力例です。

ID	CURRENT_HOST	STATE	ALERT
128	so5	PRIMARY	NO
129	so6	PEER	NO

2 record(s) selected.

## DB2\_GET\_INSTANCE\_INFO 表関数

DB2\_GET\_INSTANCE\_INFO 表関数は DB2\_MEMBER および DB2\_CF 管理ビューと同じ情報を戻しますが、現行ホストなどの入力パラメーターを渡すことにより、戻される情報をフィルターに掛けることが可能です。現行ホストとは、メンバーが現在実行されているホストのことです。

### 構文

```
DB2_GET_INSTANCE_INFO(—id—,—home_host—,—current_host—,—type—,—db_partition_num—)
```

スキーマは SYSPROC です。

### 表関数パラメーター

*id* メンバーまたはクラスター・キャッシング・ファシリティの有効な ID を指定する、タイプ INTEGER のオプション入力引数。入力に一致するすべてのデータベース・メンバーまたはクラスター・キャッシング・ファシリティの項目が戻されます。このパラメーターが NULL または -2 である場合、すべてのレコードが戻されます。このパラメーターが -1 の場合は、現在接続されているメンバーの情報が戻されます。

#### *home\_host*

レコードが戻される対象のホーム・ホストを指定する、タイプ VARCHAR(255) のオプション入力引数。ホーム・ホストとは、特定のメンバーが作成されたときにそのメンバーに関連付けられたオリジナルのホスト名のことです。 *home\_host* 引数には短形式のホスト名を使用します (IP アドレスは使用しないでください)。このパラメーターが NULL または空ストリングである場合、すべてのレコードが戻されます。

#### *current\_host*

レコードが戻される対象の現行ホストを指定する、タイプ VARCHAR(255) の

オプション入力引数。現行ホストとは、メンバーが現在実行されているホストのことであり、ホーム・ホストと同じでない場合があります (例えば restart light 操作の一部として別のホストでメンバーを開始する必要があった場合)。  
*current\_host* 引数には短形式のホスト名を使用します (IP アドレスは使用しないでください)。

*type*

メンバーの情報を取得するか、それともクラスター・キャッシング・ファシリティーの情報を取得するかを指定する、タイプ VARCHAR(32) のオプション入力引数。可能な値は次のとおりです。

- CF – 現行 DB2 pureScale インスタンス内のクラスター・キャッシング・ファシリティーに関するすべてのレコードを戻します。
- MEMBER – 現行 DB2 インスタンス内のメンバーに関するすべてのレコードを戻します。

このパラメーターが NULL または空ストリングである場合、すべてのレコードが戻されます。

*db\_partition\_num*

有効なデータベース・パーティション番号を指定する、タイプ INTEGER のオプション入力引数。入力に一致するすべてのデータベース・パーティション番号についての情報が戻されます。このパラメーターが NULL または -2 である場合、すべてのレコードが戻されます。このパラメーターが -1 の場合は、現在接続されているメンバーの情報が戻されます。DB2 pureScale インスタンスの場合、有効な入力は 0 または NULL のみです。他のすべての入力値では、データは戻されません。

**許可**

DB2\_GET\_INSTANCE\_INFO 表関数に対する EXECUTE 特権。

**デフォルトの PUBLIC 特権**

なし

**例**

次の例は、DB2\_GET\_INSTANCE\_INFO 表関数を使用して、4 つのメンバーと 2 つのクラスター・キャッシング・ファシリティーが含まれる DB2 pureScale インスタンスから情報を取得します。この例では、メンバー番号 2 は、so1 というホストで再始動されています。

```
SELECT * FROM TABLE(DB2_GET_INSTANCE_INFO(null,','',',null)) as T
```

以下はこの照会の出力例です。

ID	HOME_HOST	CURRENT_HOST	TYPE	STATE	ALERT	DB_P..._NUM	LOGICAL_PORT	NETNAME
0	so1	so1	MEMBER	STARTED	NO	0	0	so1-ib0
1	so2	so1	MEMBER	RESTARTING	NO	0	0	so2-ib0
2	so3	so3	MEMBER	STARTED	NO	0	0	so3-ib0
3	so4	so4	MEMBER	STARTED	NO	0	0	so4-ib0
128	so5	so5	CF	PRIMARY	NO	-	0	so5-ib0
129	so6	so6	CF	PEER	NO	-	0	so6-ib0

6 record(s) selected.



次の例は、DB2\_GET\_INSTANCE\_INFO 表関数を使用して、DB2 高可用性災害時リカバリー (HADR) フィーチャーを使用する、4 つのメンバーが含まれるパーティション・データベース・インスタンスから情報を取得します。

```
SELECT * FROM TABLE(DB2_GET_INSTANCE_INFO(null,'','','null')) as T
```

以下はこの照会の出力例です。

```
ID HOME_HOST CURRENT_HOST TYPE STATE ALERT DB_PARTITION_NUM LOGICAL_PORT NETNAME
-----
0 so1 so1 MEMBER STARTED NO 0 0 so1-ib0
2 so2 so2 MEMBER STARTED NO 2 0 so2-ib0
4 so3 so3 MEMBER STARTED NO 4 0 so3-ib0
7 so4 so4 MEMBER STARTED NO 7 0 so4-ib0
4 record(s) selected.
```

## 戻される情報

表 91. DB2\_MEMBER 管理ビューおよび DB2\_CF 管理ビューと DB2\_GET\_INSTANCE\_INFO で戻される情報

列名	データ・タイプ	説明
ID	SMALLINT	id - クラスター・キャッシング・ファシリティー ID モニター・エレメント
HOME_HOST	VARCHAR(255)	インスタンスに最初に追加されたときにメンバーに関連付けられたマシン。
CURRENT_HOST	VARCHAR(255)	メンバーが現在実行されているマシン名。
TYPE	VARCHAR(32)	「MEMBER」または「CF」のどちらか。
STATE	VARCHAR(32)	<p>メンバーまたはクラスター・キャッシング・ファシリティーの状態。</p> <p>メンバーの状態を表す値には、STARTED、STOPPED、RESTARTING、WAITING_FOR_FAILBACK、ERROR、および UNKNOWN があります。</p> <p>クラスター・キャッシング・ファシリティーの状態を表す値には、STOPPED、RESTARTING、BECOMING_PRIMARY、PRIMARY、CATCHUP<sup>1</sup>、PEER、ERROR、および UNKNOWN があります。</p> <p>メンバーまたはクラスター・キャッシング・ファシリティーが NULL 状態である場合、その状態が適用できないことを示します。例えば、パーティション・データベース環境の場合などです。</p> <p>詳しくは、『メンバーおよびクラスター・キャッシング・ファシリティーの状態とアラートの値』を参照してください。</p>

表 91. DB2\_MEMBER 管理ビューおよび DB2\_CF 管理ビューと DB2\_GET\_INSTANCE\_INFO で戻される情報 (続き)

列名	データ・タイプ	説明
ALERT	VARCHAR(8)	<p>インスタンスに関するアラートの情報。</p> <ul style="list-style-type: none"> <li>• YES - アラートがあります</li> <li>• NO - アラートはありません</li> <li>• メンバーまたはクラスター・キャッシング・ファシリティーが NULL 状態である場合、その状態が適用できないことを示します。例えば、パーティション・データベース環境の場合などです。</li> </ul> <p>アラートについての詳細情報を取得するには、DB2_INSTANCE_ALERTS 管理ビューを使用します。</p>
DB_PARTITION_NUM	SMALLINT	このメンバーのデータベース・パーティション番号。
LOGICAL_PORT	SMALLINT	メンバーまたはクラスター・キャッシング・ファシリティーの論理ポート番号。
NETNAME	VARCHAR(255)	内部データベース通信用の高速相互接続の名前。

**注:**

1. CATCHUP の場合、戻される状態の一部としてパーセンテージ値が含まれます。このパーセンテージ値は、2 次クラスター・キャッシング・ファシリティーが 1 次キャッシング・ファシリティーの現在の状態にキャッチアップした量を表します。

## 環境ルーチンおよびビュー

### ENV\_CF\_SYS\_RESOURCES 管理ビュー - クラスター・キャッシング・ファシリティーのシステム・リソース情報の取得

ENV\_CF\_SYS\_RESOURCES 管理ビューは、システム上のクラスター・キャッシング・ファシリティー (CF と呼ばれる) によって使用されるシステム・リソースのリストを戻します。

スキーマは SYSIBMADM です。

#### 許可

以下のいずれかの権限が必要です。

- ビューに対する SELECT 特権
- ビューに対する CONTROL 特権
- DATAACCESS 権限

#### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

次の例は、ENV\_CF\_SYS\_RESOURCES 管理ビューを使用して取得できる情報を示しています。

```
SELECT * FROM SYSIBMADM.ENV_CF_SYS_RESOURCES
```

以下はこの照会の出力例です。

## 戻される情報

表 92. ENV\_CF\_SYS\_RESOURCES 管理ビューによって戻される情報

列名	データ・タイプ	説明
NAME	VARCHAR(128)	システム・リソースの名前 (使用される値については、次の表を参照)。
VALUE	VARCHAR(1024)	システム・リソースの値。
DATATYPE	VARCHAR(128)	値のデータ・タイプ (使用される値については、次の表を参照)。
UNIT	VARCHAR(128)	値で表される単位 (該当する場合)。それ以外の場合は NULL。
ID	SMALLINT	id - クラスタ・キャッシング・ファシリティ ID モニター・エレメント

表 93. NAME 列および DATATYPE 列に使用される値

情報タイプ	名前	データ・タイプ	説明
オペレーティング・システム	HOST_NAME	VARCHAR(256)	クラスタ・キャッシング・ファシリティが実行されているホストの名前。
物理メモリー	MEMORY_TOTAL	BIGINT	物理メモリーの合計サイズ。
	MEMORY_FREE	BIGINT	空き物理メモリーの量。
	MEMORY_SWAP_TOTAL	BIGINT	スワップ・スペースの合計量。
	MEMORY_SWAP_FREE	BIGINT	空きスワップ・スペースの量。
仮想メモリー	VIRTUAL_MEM_TOTAL	BIGINT	システム上の仮想メモリーの合計量。
	VIRTUAL_MEM_RESERVED	BIGINT	予約済み仮想メモリーの量。
CPU ロード	CPU_USAGE_TOTAL	DECIMAL	マシンの全 CPU 使用量のパーセント。

## ENV\_FEATURE\_INFO 管理ビュー - DB2 フィーチャーのライセンス情報を戻す

ENV\_FEATURE\_INFO 管理ビューは、ライセンスが必要とされる使用可能なすべてのフィーチャーに関する情報を戻します。フィーチャーごとに、そのフィーチャーの有効なライセンスがインストールされているかどうかに関する情報があります。

スキーマは SYSIBMADM です。

## 許可

以下のいずれかの権限が必要です。

- ENV\_FEATURE\_INFO 管理ビューに対する SELECT 特権
- ENV\_FEATURE\_INFO 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

インストール済みの DB2 フィーチャー・ライセンス情報の要求

```
SELECT * FROM SYSIBMADM.ENV_FEATURE_INFO
```

以下はこの照会の出力例です。

```
FEATURE_NAME  FEATURE_FULLNAME  ...
-----
DPF           DB2_DATABASE_PARTITIONING_FEATURE  ...
POESE        DB2_PERFORMANCE_OPTIMIZATION_FEATURE_FOR_ESE  ...
SO           DB2_STORAGE_OPTIMIZATION_FEATURE  ...
AAC          DB2_ADVANCED_ACCESS_CONTROL_FEATURE  ...
HFESE       IBM_HOMOGENEOUS_FEDERATION_FEATURE_FOR_ESE  ...
XMLESE      DB2_PUREXML_FEATURE_FOR_ESE  ...
```

この照会からの出力 (続き)。

```
... LICENSE_INSTALLED  PRODUCT_NAME  FEATURE_USE_STATUS
... -----
... Y                  ESE          IN_COMPLIANCE
... Y                  ESE          IN_COMPLIANCE
... Y                  ESE          IN_COMPLIANCE
... Y                  ESE          NOT_USED
... Y                  ESE          NOT_USED
... N                  ESE          IN_VIOLATION
```

## ENV\_FEATURE\_INFO 管理ビューのメタデータ

表 94. ENV\_FEATURE\_INFO 管理ビューのメタデータ

列名	データ・タイプ	説明
FEATURE_NAME	VARCHAR(26)	ライセンスがある DB2 サーバー上で使用可能な DB2 フィーチャーの短縮名。
FEATURE_FULLNAME	VARCHAR(100)	DB2 フィーチャーのフルネーム。列値は英語の大文字で表示されます。語はスペース文字ではなく下線文字で区切られます。

表 94. ENV\_FEATURE\_INFO 管理ビューのメタデータ (続き)

列名	データ・タイプ	説明
LICENSE_INSTALLED	CHAR(1)	フィーチャーにライセンスがあるかどうかを示します。値が 'N' の場合、フィーチャーにはライセンスがありません。値が 'Y' の場合、フィーチャーにはライセンスがあります。
PRODUCT_NAME	VARCHAR(26)	product_name - 製品名モニター・エレメント
FEATURE_USE_STATUS	VARCHAR(30)	ライセンス準拠状況を示します。この値はフィーチャーの使用状況を示します。使用できる値は次の 3 つです。 <ul style="list-style-type: none"> <li>• IN_COMPLIANCE: フィーチャーは少なくとも 1 回使用されており、フィーチャーの有効なライセンスがあります。</li> <li>• IN_VIOLATION: フィーチャーは少なくとも 1 回使用されていますが、フィーチャーの有効なライセンスがありません。</li> <li>• NOT_USED: フィーチャーは使用されていません。</li> </ul>

## ENV\_GET\_DB2\_SYSTEM\_RESOURCES 表関数 - DB2 システム情報を戻す

ENV\_GET\_DB2\_SYSTEM\_RESOURCES 表関数は、現行インスタンス内の指定したメンバーの CPU 使用量と DB2 プロセスに関する情報を戻します。

### 構文

▶—ENV\_GET\_DB2\_SYSTEM\_RESOURCES—(—member—)————▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### member

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーの場合は -1、すべてのデータベース・メンバーの場合は -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

以下のいずれかの権限が必要です。

- 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

### 例

```
SELECT VARCHAR(db2_process_name, 20) AS NAME, CPU_USER, CPU_SYSTEM
  FROM TABLE(ENV_GET_DB2_SYSTEM_RESOURCES(-2))
```

この照会は、以下の出力を戻します。

NAME	CPU_USER	CPU_SYSTEM
db2fmp	14	9
db2sysc	11752	541
db2syscr	13	8
db2fmp	24	14

4 record(s) selected.

### 戻される情報

表 95. ENV\_GET\_DB2\_SYSTEM\_RESOURCES 表関数によって戻される情報

列名	データ・タイプ	説明
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
DB2_PROCESS_NAME	VARCHAR(128)	db2_process_name - DB2 プロセス名のモニター・エレメント
DB2_PROCESS_ID	BIGINT	db2_process_id - DB2 プロセス ID のモニター・エレメント
CPU_USER	BIGINT	cpu_user - 非カーネル処理時間 : モニター・エレメント
CPU_SYSTEM	BIGINT	cpu_system - カーネル時間のモニター・エレメント

## ENV\_GET\_NETWORK\_RESOURCES 表関数 - ネットワーク情報を戻す

ENV\_GET\_NETWORK\_RESOURCES 表関数は、DB2 を実行するホスト・マシン上でアクティブなすべてのネットワーク・アダプターに関する情報を戻します。

### 許可

以下のいずれかの権限が必要です。

- 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

スキーマは SYSPROC です。

## デフォルトの PUBLIC 特権

なし

### 例

```
SELECT varchar(adapter_name, 20) as name,  
       packets_received,  
       packets_sent,  
       total_bytes_received,  
       total_bytes_sent  
FROM TABLE(ENV_GET_NETWORK_RESOURCES())
```

この照会は、以下の入力に戻します。

NAME	PACKETS_RECEIVED	PACKETS_SENT	TOTAL_BYTES_RECEIVED
lo	467182039	467182039	528451011980
eth0	426287355	431398744	351656704796
eth1	0	0	0

TOTAL_BYTES_SENT
528451011980
272061746005
0

### 戻される情報

表 96. ENV\_GET\_NETWORK\_RESOURCES 表関数によって戻される情報

列名	データ・タイプ	説明
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
HOST_NAME	VARCHAR(255)	host_name - ホスト名モニター・エレメント
ADAPTER_NAME	VARCHAR(255)	adapter_name - アダプター名のモニター・エレメント
PACKETS_RECEIVED	BIGINT	packets_received - 受信パケット数のモニター・エレメント
PACKETS_SENT	BIGINT	packets_sent - 送信パケット数のモニター・エレメント
PACKET_RECEIVE_ERRORS	BIGINT	packet_receive_errors - パケット受信エラーのモニター・エレメント
PACKET_SEND_ERRORS	BIGINT	packet_send_errors - パケット送信エラーのモニター・エレメント
TOTAL_BYTES_RECEIVED	BIGINT	total_bytes_received - 受信バイト数の合計モニター・エレメント
TOTAL_BYTES_SENT	BIGINT	total_bytes_sent - 送信バイト数の合計モニター・エレメント

## ENV\_GET\_REG\_VARIABLES 表関数 - 使用中の DB2 レジストリー設定の取得

ENV\_GET\_REG\_VARIABLES 表関数は、1 つまたはすべてのデータベース・メンバーの DB2 レジストリー設定値を戻します。

注: ENV\_GET\_REG\_VARIABLES 表関数は、使用すべきでない REG\_VARIABLES 管理ビューに代わる機能です。 ENV\_GET\_REG\_VARIABLES 関数は、照会する特定のメンバーを単一パラメーター値で示せる点、およびディスクに現在格納されているレジストリー設定値の追加結果を戻す点が異なります。

### 構文

```
▶▶—ENV_GET_REG_VARIABLES—(—member—)—————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

*member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

以下のいずれかの権限が必要です。

- ENV\_GET\_REG\_VARIABLES 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、以下のいずれかの特権または権限も必要です。

- ENV\_GET\_REG\_VARIABLES 表関数に対する EXECUTE 特権

### デフォルトの PUBLIC 特権

なし

### 例

この例では、暗黙接続に使用するデータベース別名を指定するレジストリー変数 DB2DBDFT を、CORP\_1 に設定します。これを DB2 インスタンスの開始前に行います。

```
db2set db2dbdft=CORP_1
db2start
```

この後で、そのレジストリー変数設定値を示すために次の照会を発行できます。



```
select substr(reg_var_value,1,20) as VALUE,
       substr(reg_var_on_disk_value,1,20) as ON_DISK_VALUE
from table(env_get_reg_variables(-1))
where reg_var_name = 'DB2DBDFT'
```

この照会は、以下の出力を戻します。

```
VALUE                ON_DISK_VALUE
-----
CORP_1              CORP_1

      1 record(s) selected.
```

メモリー内とディスク上のレジストリー設定値の違いを示すために、DB2DBDFT レジストリー変数を今度は次のように変更します。

```
db2set db2dbdft=DEPT_MAJOR
```

先ほどと同じ照会を実行すると、この新しい値が ENV\_GET\_REG\_VARIABLES 関数によって直ちに取り込まれることが分かります。

```
VALUE                ON_DISK_VALUE
-----
CORP_1              DEPT_MAJOR

      1 record(s) selected.
```

ただし、メモリー内の値は DB2 インスタンスが再始動されるまで新しい値に変わらないことに注意してください。

## 戻される情報

表 97. ENV\_GET\_REG\_VARIABLES 表関数によって戻される情報

列名	データ・タイプ	説明
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
REG_VAR_NAME	VARCHAR(256)	DB2 レジストリー変数の名前。
REG_VAR_VALUE	VARCHAR(2048)	DB2 レジストリー変数のメモリー内の現行設定値。
REG_VAR_ON_DISK_VALUE	VARCHAR(2048)	DB2 レジストリー変数のディスク上の現行設定値。
IS_AGGREGATE	SMALLINT	DB2 レジストリー変数が集約変数であるかどうかを示します。考えられる戻り値は、集約変数でない場合は 0、集約変数である場合は 1 です。

表 97. ENV\_GET\_REG\_VARIABLES 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
AGGREGATE_NAME	VARCHAR(256)	現在 DB2 レジストリー変数が、構成済みの集約から値を取得している場合には、集約の名前。レジストリー変数が集約から設定されているのではない場合、または集約から設定されたもののオーバーライドされている場合、AGGREGATE_NAME の値は NULL になります。
LEVEL	CHAR(1)	DB2 レジストリー変数がその値を獲得するレベルを示します。考えられる戻り値と、それが表す対応するレベルは以下のとおりです。 <ul style="list-style-type: none"> <li>• I = インスタンス</li> <li>• G = グローバル</li> <li>• N = データベース・パーティション</li> <li>• E = 環境</li> </ul>

### ENV\_GET\_SYSTEM\_RESOURCES 表関数 - システム情報を戻す

ENV\_GET\_SYSTEM\_RESOURCES 表関数は、オペレーティング・システム、CPU、メモリーなど、システム上のメンバーに関連した情報を戻します。アクティブなデータベースは、システム上の 1 つ以上のメンバーに存在することがあります。コマンドを発行したデータベースがアクティブ状態になっているメンバーからのみ、この表関数がデータを戻します。

注: ENV\_GET\_SYSTEM\_RESOURCES 表関数は、ENV\_GET\_SYS\_RESOURCES 表関数とそれに関連する ENV\_SYS\_RESOURCES 管理ビューに代わる機能です。

ENV\_GET\_SYSTEM\_RESOURCES 表関数は、メンバーごとに複数の行を戻すのではなく、メンバーごとに複数の列から成る 1 行で、すべてのデータを戻します。また、ENV\_GET\_SYS\_RESOURCES 表関数で戻される DBPARTITIONNUM 列のデータは、ENV\_GET\_SYSTEM\_RESOURCES 表関数では MEMBER 列で戻されます。

スキーマは SYSPROC です。

### 許可

以下のいずれかの権限が必要です。

- 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

### 使用上の注意

- 仮想化環境 (VMWare ESX など) では、DB2 が CPU\_HMT\_DEGREE または CPU\_CORES\_PER\_SOCKET などの CPU トポロジを判別できない場合があります。これは、ハイパーバイザーが仮想化オペレーティング・システムに対して提供する情報が限られているために生じます。このような場合は、CPU\_CORES\_PER\_SOCKET および CPU\_HMT\_DEGREE の値が 1 になります。

### 戻される情報

表 98. ENV\_GET\_SYSTEM\_RESOURCES で戻される情報

列名	データ・タイプ	説明
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
OS_NAME	VARCHAR(256)	os_name - オペレーティング・システム名 : モニター・エレメント
HOST_NAME	VARCHAR(255)	host_name - ホスト名モニター・エレメント
OS_VERSION	VARCHAR(256)	os_version - オペレーティング・システム・バージョン : モニター・エレメント
OS_RELEASE	VARCHAR(256)	os_release - オペレーティング・システム・リリース : モニター・エレメント
MACHINE_IDENTIFICATION	VARCHAR(256)	machine_identification - ホスト・ハードウェア ID のモニター・エレメント
OS_LEVEL	VARCHAR(256)	os_level - オペレーティング・システムのレベル : モニター・エレメント
CPU_TOTAL	BIGINT	cpu_total - CPU 数モニター・エレメント
CPU_ONLINE	BIGINT	cpu_online - オンライン CPU 数のモニター・エレメント
CPU_CONFIGURED	BIGINT	cpu_configured - 構成されている CPU 数のモニター・エレメント
CPU_SPEED	BIGINT	cpu_speed - CPU クロック速度のモニター・エレメント
CPU_TIMEBASE	BIGINT	cpu_timebase - 時間基準のレジスター増分の周波数のモニター・エレメント
CPU_HMT_DEGREE	BIGINT	cpu_hmt_degree - 論理 CPU 数のモニター・エレメント
CPU_CORES_PER_SOCKET	BIGINT	cpu_cores_per_socket - ソケットあたりの CPU コア数のモニター・エレメント
MEMORY_TOTAL	BIGINT	memory_total - 合計物理メモリーのモニター・エレメント
MEMORY_FREE	BIGINT	memory_free - 物理メモリーの空き容量のモニター・エレメント
MEMORY_SWAP_TOTAL	BIGINT	memory_swap_total - スワップ・スペース合計 : モニター・エレメント
MEMORY_SWAP_FREE	BIGINT	memory_swap_free - フリーのスワップ・スペース合計 : モニター・エレメント
VIRTUAL_MEM_TOTAL	BIGINT	virtual_mem_total - 合計仮想メモリーのモニター・エレメント

表 98. ENV\_GET\_SYSTEM\_RESOURCES で戻される情報 (続き)

列名	データ・タイプ	説明
VIRTUAL_MEM_RESERVED	BIGINT	virtual_mem_reserved - 予約済み仮想メモリーのモニター・エレメント
VIRTUAL_MEM_FREE	BIGINT	virtual_mem_free - 空き仮想メモリーのモニター・エレメント
CPU_LOAD_SHORT	DOUBLE	cpu_load_short - プロセッサ負荷 (短い時間フレーム) のモニター・エレメント
CPU_LOAD_MEDIUM	DOUBLE	cpu_load_medium - プロセッサ負荷 (中程度の時間フレーム) のモニター・エレメント
CPU_LOAD_LONG	DOUBLE	cpu_load_long - プロセッサ負荷 (長い時間フレーム) のモニター・エレメント
CPU_USAGE_TOTAL	SMALLINT	cpu_usage_total - プロセッサ使用率のモニター・エレメント
CPU_USER <sup>1</sup>	BIGINT	cpu_user - 非カーネル処理時間：モニター・エレメント
CPU_IDLE <sup>1</sup>	BIGINT	cpu_idle - プロセッサ・アイドル時間のモニター・エレメント
CPU_IOWAIT <sup>1</sup>	BIGINT	cpu_iowait - 入出力待機時間モニター・エレメント
CPU_SYSTEM <sup>1</sup>	BIGINT	cpu_system - カーネル時間のモニター・エレメント
SWAP_PAGE_SIZE	BIGINT	swap_page_size - スワップ・ページ・サイズのモニター・エレメント
SWAP_PAGES_IN	BIGINT	swap_pages_in - ディスクからスワップインされたページ数のモニター・エレメント
SWAP_PAGES_OUT	BIGINT	swap_pages_out - ディスクにスワップアウトされたページ数のモニター・エレメント

注:

- 1 これらのメトリックは、システム上のすべての論理プロセッサを対象とした総計です。  
 AIX オペレーティング・システムでは、DB2 サーバーが実行されているワークロード・パーティション (WPAR) および論理パーティション (LPAR) のメトリックです。

例

例 1: 3 つのメンバーから成る DB2 pureScale がセットアップされた環境で、メンバーごとのオペレーティング・システム情報を取得します。

```
select MEMBER, varchar(HOST_NAME,12) as HOST_NAME, varchar(OS_NAME,8) as OS_NAME,
       varchar(OS_VERSION,8) as OS_VERSION, varchar(OS_RELEASE,8)
       as OS_RELEASE from table(SYSPROC.ENV_GET_SYSTEM_RESOURCES()) order by MEMBER
```

出力例を以下に示します。

MEMBER	HOST_NAME	OS_NAME	OS_VERSION	OS_RELEASE
0	hote175	Linux	2	6
1	hote178	Linux	2	6
2	hote190	Linux	2	6

3 record(s) selected.

例 2: DB2 Enterprise Server Edition システム上の、このデータベースのホストとなっているシステムのメモリー情報を取得します。

```
select varchar(HOST_NAME,12) as HOST_NAME, MEMORY_TOTAL, MEMORY_FREE,
       MEMORY_SWAP_TOTAL, MEMORY_SWAP_FREE, VIRTUAL_MEM_TOTAL, VIRTUAL_MEM_FREE
       from table(SYSPROC.ENV_GET_SYSTEM_RESOURCES())
```

出力例を以下に示します。スペースの制約上、出力の続きを 2 番目の行で示します。

HOST_NAME	MEMORY_TOTAL	MEMORY_FREE	MEMORY_SWAP_TOTAL	...
hotel75	32189	4370	8198	...

1 record(s) selected

...	MEMORY_SWAP_FREE	VIRTUAL_MEM_TOTAL	VIRTUAL_MEM_FREE
...	7316	40387	11686

例 3: 4 つのメンバーから成るパーティション・データベース環境で、データベースのホストとなっているすべてのシステムでのロード回数を検出します。

```
select MEMBER, varchar(HOST_NAME,12) as HOST_NAME,
       CPU_LOAD_SHORT, CPU_LOAD_MEDIUM, CPU_LOAD_LONG
       from table(SYSPROC.ENV_GET_SYSTEM_RESOURCES()) order by MEMBER
```

出力例を以下に示します。

MEMBER	HOST_NAME	CPU_LOAD_SHORT	CPU_LOAD_MEDIUM	CPU_LOAD_LONG
0	hotel75	+5.21000000000000E+000	+5.08000000000000E+000	+4.67000000000000E+000
1	hotel78	+1.33000000000000E+000	+2.18000000000000E+000	+3.66000000000000E+000
2	hotel90	+9.02000000000000E+000	+9.08000000000000E+000	+9.47000000000000E+000
3	hotel32	+1.09000000000000E+000	+1.38000000000000E+000	+1.27000000000000E+000

4 record(s) selected.

例 4:

```
select MEMBER, varchar(HOST_NAME,12) as HOST_NAME, CPU_TOTAL,
       MEMORY_TOTAL, CPU_LOAD_SHORT
       from table(SYSPROC.ENV_GET_SYSTEM_RESOURCES())
       order by MEMBER
```

出力例を以下に示します。

MEMBER	HOST_NAME	CPU_TOTAL	MEMORY_TOTAL	CPU_LOAD_SHORT
0	coralpi23	24	81920	+1.23696899414062E+000
1	coralpi23	24	81920	+1.23696899414062E+000
2	coralpi23	24	81920	+1.23696899414062E+000
3	coralpi23	24	81920	+1.23696899414062E+000

4 record(s) selected.

## ENV\_INST\_INFO 管理ビュー - 現在のインスタンスに関する情報の検索

ENV\_INST\_INFO 管理ビューは、現在のインスタンスについての情報を戻します。

スキーマは SYSIBMADM です。

## 許可

以下のいずれかの権限が必要です。

- ENV\_INST\_INFO 管理ビューに対する SELECT 特権
- ENV\_INST\_INFO 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在のインスタンスについての情報を要求します。

```
SELECT * FROM SYSIBMADM.ENV_INST_INFO
```

以下はこの照会の出力例です。

```
INST_NAME          IS_INST_PARTITIONABLE NUM_DBPARTITIONS INST_PTR_SIZE ...
-----
DB2                  0                      1                32 ...
```

1 record(s) selected.

この照会の出力 (続き)。

```
... RELEASE_NUM      SERVICE_LEVEL          BLD_LEVEL             PTF                   FIXPACK_NUM
... -----
... 01010107         DB2 v9.1.0.115        n051106                .                      0
```

## 戻される情報

表 99. ENV\_INST\_INFO 管理ビューによって戻される情報

列名	データ・タイプ	説明
INST_NAME	VARCHAR(128)	現在のインスタンスの名前。
IS_INST_PARTITIONABLE	SMALLINT	現在のインスタンスがパーティション化が可能なデータベース・サーバー・インスタンスであるかどうかを示します。考えられる戻り値は、パーティション化可能データベース・サーバー・インスタンスでない場合は 0、パーティション化可能データベース・サーバー・インスタンスである場合は 1 です。
NUM_DBPARTITIONS	INTEGER	データベース・パーティションの数。パーティション・データベース環境ではない場合、値 1 を戻します。
INST_PTR_SIZE	INTEGER	現在のインスタンスのビット・サイズ (32 または 64)。

表 99. ENV\_INST\_INFO 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明
RELEASE_NUM	VARCHAR(128)	db2level コマンドによって戻される、内部リリース番号。例えば 03030106 など。
SERVICE_LEVEL	VARCHAR(128)	service_level - サービス・レベル・モニター・エレメント
BLD_LEVEL	VARCHAR(128)	db2level コマンドによって戻される、ビルド・レベル。例えば n041021 など。
PTF	VARCHAR(128)	db2level コマンドによって戻される、プログラム一時修正 (PTF) ID。例えば U498350 など。
FIXPACK_NUM	INTEGER	db2level コマンドによって戻される、フィックスパック番号。例えば 9 など。
NUM_MEMBERS	INTEGER	このインスタンス上のメンバーの数。

## ENV\_PROD\_INFO 管理ビュー - インストール済みの DB2 製品に関する情報の検索

ENV\_PROD\_INFO 管理ビューは、インストール済みの DB2 データベース製品についての情報を戻します。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- ENV\_PROD\_INFO 管理ビューに対する SELECT 特権
- ENV\_PROD\_INFO 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

インストール済みの DB2 製品の情報を要求します。

```
SELECT * FROM SYSIBMADM.ENV_PROD_INFO
```

以下はこの照会の出力例です。

```

INSTALLED_PROD  INSTALLED_PROD_FULLNAME  ...
-----
ESE              DB2_ENTERPRISE_SERVER_EDITION  ...
WSE              DB2_WORKGROUP_SERVER_EDITION  ...
EXP              DB2_EXPRESS_EDITION            ...

```

この照会からの出力 (続き)。

```

... LICENSE_INSTALLED  PROD_RELEASE  LICENSE_TYPE
... -----
... Y                  9.5          AUTHORIZED_USER_OPTION
... N                  9.5          LICENSE_NOT_REGISTERED
... Y                  9.5          RESTRICTED

```

## ENV\_PROD\_INFO 管理ビューのメタデータ

表 100. ENV\_PROD\_INFO 管理ビューのメタデータ

列名	データ・タイプ	説明
INSTALLED_PROD	VARCHAR(26)	システムにインストール済みの DB2 製品の ID。
INSTALLED_PROD_FULLNAME	VARCHAR(100)	インストール済みの DB2 製品のフルネーム。列値は英語の大文字で表示されます。語は下線文字で区切られます。
LICENSE_INSTALLED	CHAR(1)	製品にライセンスがあるかどうかを示します。値が N の場合、製品にはライセンスがありません。値が Y の場合、製品にはライセンスがあります。LICENSE_TYPE 値が TRIAL および LICENSE_NOT_REGISTERED の場合、値は常に N になります。
PROD_RELEASE	VARCHAR(26)	製品リリース番号。
LICENSE_TYPE	VARCHAR(50)	製品についてインストールされるライセンスのタイプの名前。使用できる戻り値は次のとおりです。 <ul style="list-style-type: none"> <li>• 12_MONTHS_LICENSE_AND_SUBSCRIPTION</li> <li>• AUTHORIZED_USER</li> <li>• AUTHORIZED_USER_OPTION</li> <li>• CLIENT_DEVICE</li> <li>• CPU</li> <li>• CPU_OPTION</li> <li>• HOST_SERVER_AND_MSU</li> <li>• LICENSE_NOT_REGISTERED</li> <li>• MANAGED_PROCESSOR</li> <li>• N/A</li> <li>• RESTRICTED</li> <li>• TRIAL</li> <li>• UNWARRANTED</li> <li>• USER</li> </ul>

## ENV\_SYS\_INFO 管理ビュー - システムに関する情報の検索

ENV\_SYS\_INFO 管理ビューは、システムについての情報を戻します。

スキーマは SYSIBMADM です。



## 許可

以下のいずれかの権限が必要です。

- ENV\_SYS\_INFO 管理ビューに対する SELECT 特権
- ENV\_SYS\_INFO 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

システムについての情報を要求します。

```
SELECT * from SYSIBMADM.ENV_SYS_INFO
```

以下はこの照会の出力例です。

```
OS_NAME      OS_VERSION  OS_RELEASE      HOST_NAME
-----
WIN32_NT     5.1         Service Pack 1  D570
```

1 record(s) selected.

この照会からの出力 (続き)。

```
... TOTAL_CPUS  CONFIGURED_CPUS  TOTAL_MEMORY
... -----
...           1             2           1527
```

## 戻される情報

表 101. ENV\_SYS\_INFO 管理ビューによって戻される情報

列名	データ・タイプ	説明
OS_NAME	VARCHAR(256)	オペレーティング・システムの名前。
OS_VERSION	VARCHAR(256)	オペレーティング・システムのバージョン番号。
OS_RELEASE	VARCHAR(256)	オペレーティング・システムのリリース番号。
HOST_NAME	VARCHAR(256)	host_name - ホスト名モニター・エレメント
TOTAL_CPUS	INTEGER	システム上の物理 CPU の総数。
CONFIGURED_CPUS	INTEGER	システム上の構成済み物理 CPU の数。
TOTAL_MEMORY	INTEGER	システム上のメモリーの合計量 (MB 単位)。

## Explain ルーチン

### EXPLAIN\_GET\_MSGS

EXPLAIN\_GET\_MSGS 表関数は、EXPLAIN\_DIAGNOSTIC および EXPLAIN\_DIAGNOSTIC\_DATA Explain 表を照会し、定様式メッセージを戻します。

#### 構文

```
►►—EXPLAIN_GET_MSGS—(—explain-requester—,—explain-time—,—source-name—,—►►  
►—source-schema—,—source-version—,—explain-level—,—stmtno—,—sectno—,—►►  
►—locale—)—►►
```

スキーマは Explain 表スキーマと同じです。

#### 表関数パラメーター

以下の入力引数はいずれもヌルにすることができます。引数がヌルの場合、その引数は照会を制限するためには使用されません。

##### *explain-requester*

この Explain 要求のイニシエーターの許可 ID を指定する、タイプ VARCHAR(128) の入力引数。NULL 値であれば、このパラメーターは照会の検索条件から除外されます。

##### *explain-time*

Explain 要求の開始時刻を指定する、タイプ TIMESTAMP の入力引数。NULL 値であれば、このパラメーターは照会の検索条件から除外されます。

##### *source-name*

動的ステートメントの Explain 実行時に実行されるパッケージの名前、または静的 SQL ステートメントの Explain 実行時のソース・ファイルの名前を指定する、タイプ VARCHAR(128) の入力引数。NULL 値であれば、このパラメーターは照会の検索条件から除外されます。

##### *source-schema*

Explain 要求のソースのスキーマ、または修飾子を指定する、タイプ VARCHAR(128) の入力引数。NULL 値であれば、このパラメーターは照会の検索条件から除外されます。

##### *source-version*

Explain 要求のソースのバージョンを指定する、タイプ VARCHAR(64) の入力引数。NULL 値であれば、このパラメーターは照会の検索条件から除外されません。

##### *explain-level*

この行が関係する Explain 情報のレベルを指定する、タイプ CHAR(1) 入力引数。NULL 値であれば、このパラメーターは照会の検索条件から除外されません。

### *stmtno*

この Explain 情報が関連付けられるパッケージ内のステートメント番号を指定する、タイプ INTEGER の入力引数。 NULL 値であれば、このパラメーターは照会の検索条件から除外されます。

### *sectno*

この Explain 情報が関連付けられるパッケージ内のセクション番号を指定する、タイプ INTEGER の入力引数。 NULL 値であれば、このパラメーターは照会の検索条件から除外されます。

### *locale*

戻されるメッセージのロケールを指定する、タイプ VARCHAR(33) の入力引数。指定したロケールが DB2 サーバーにインストールされていない場合、値は無視されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- EXPLAIN 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 戻される情報

表 102. EXPLAIN\_GET\_MSGS 表関数によって戻される情報

列名	データ・タイプ	説明
EXPLAIN_REQUESTER	VARCHAR(128)	この Explain 要求のイニシエーターの許可 ID。
EXPLAIN_TIME	TIMESTAMP	Explain 要求の開始時刻。
SOURCE_NAME	VARCHAR(128)	動的ステートメントの Explain 実行時に実行されるパッケージの名前、または静的 SQL ステートメントの Explain 実行時のソース・ファイルの名前。
SOURCE_SCHEMA	VARCHAR(128)	Explain 要求のソースのスキーマまたは修飾子。
SOURCE_VERSION	VARCHAR(64)	Explain 要求のソースのバージョン。
EXPLAIN_LEVEL	CHAR(1)	この行に関連する Explain 情報のレベル。
STMTNO	INTEGER	この Explain 情報が関連付けられるパッケージ内のステートメント番号。

表 102. EXPLAIN\_GET\_MSGS 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
SECTNO	INTEGER	この Explain 情報が関連付けられるパッケージ内のセクション番号。
DIAGNOSTIC_ID	INTEGER	EXPLAIN_STATEMENT 表内の特定のステートメントのインスタンスの診断 ID。
LOCALE	VARCHAR(33)	戻されるメッセージのロケール。このロケールは、指定のロケールが DB2 サーバーにインストールされていない場合、指定のロケールとは一致しません。
MSG	VARCHAR(4096)	定様式メッセージ・テキスト。

## 例

過去 1 時間に生成された、リクエスター SIMMEN のデフォルト・スキーマの Explain 表から、定様式の英語のメッセージを要求します。SQLC2E03 のソース名を指定します。

```
SELECT MSG
  FROM TABLE(EXPLAIN_GET_MSGS(
    'SIMMEN',
    CAST(NULL AS TIMESTAMP),
    'SQLC2E03',
    CAST(NULL AS VARCHAR(128)),
    CAST(NULL AS VARCHAR(64)),
    CAST(NULL AS CHAR(1)),
    CAST(NULL AS INTEGER),
    CAST(NULL AS INTEGER),
    'en_US'))
 AS REGISTRYINFO
 WHERE EXPLAIN_TIME >= (CURRENT_TIMESTAMP - 1 HOUR)
 ORDER BY DIAGNOSTIC_ID
```

以下はこの照会の出力例です。

```
MSG
-----
EXP0012W Invalid access request. The index "index1" could not be found.
  Line number "554", character number "20".
EXP0012W Invalid access request. The index "index2" could not be found.
  Line number "573", character number "20".
EXP0015W Invalid join request. Join refers to tables that are not in
  the same FROM clause. Line number "573", character number "20".
```

## EXPLAIN\_FORMAT\_STATS

このスカラー関数は、定様式の統計情報を表示するために使用されます。この情報は構文解析され、特定の照会についてキャプチャーされた Explain スナップショットから抽出されます。結果のデータ・タイプは CLOB(50M) です。

## 構文

▶▶—EXPLAIN\_FORMAT\_STATS—(—snapshot—)————▶▶

スキーマは SYSPROC です。

## 関数のパラメーター

### *snapshot*

指定された照会についてキャプチャーされた Explain スナップショットである、タイプ BLOB(10M) の入力引数。これは Explain 表 *EXPLAIN\_STATEMENT* のスナップショット列として保管されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- EXPLAIN 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
SELECT EXPLAIN_FORMAT_STATS(SNAPSHOT)
FROM EXPLAIN_STATEMENT
WHERE EXPLAIN_REQUESTER = 'DB2USER1' AND
      EXPLAIN_TIME = timestamp('2006-05-12-14.38.11.109432') AND
      SOURCE_NAME = 'SQLC2F0A' AND
      SOURCE_SCHEMA = 'NULLID' AND
      SOURCE_VERSION = '' AND
      EXPLAIN_LEVEL = '0' AND
      STMTNO = 1 AND
      SECTNO = 201
```

以下はこの関数の出力例です。

Tablespace Context:

```
-----
Name:                                USERSPACE1
Overhead:                             7.500000
Transfer Rate:                         0.060000
Prefetch Size:                          32
Extent Size:                            32
Type:                                    Database managed
Partition Group Name:                   NULLP
Buffer Pool Identifier:                  0
```

Base Table Statistics:

```
-----
Name:                                  T1
Schema:                                DB2USER2
Number of Columns:                      3
Number of Pages with Rows:              1
Number of Pages:                        1
Number of Rows:                         5
Table Overflow Record Count:            0
Width of Rows:                          26
Time of Creation:                       2006-06-16-11.46.53.041085
```

```

Last Statistics Update:          2006-06-26-12.23.44.814201
Statistics Type:                Fabrication
Primary Tablespace:            USERSPACE1
Tablespace for Indexes:        USERSPACE1
Tablespace for Long Data:      NULLP
Number of Referenced Columns:   2
Number of Indexes:             1
Volatile Table:                No
Table Active Blocks:           1
Number of Column Groups:       0
Number of Data Partitions:     1
Average Row Compression Ratio:  -9.000000
Percent Rows Compressed:       -9.000000
Average Compressed Row Size:    -9
Statistics Type:                U

```

Column Information:

-----

```

Number:                1
Name:                  C1
Statistics Available:  Yes

```

Column Statistics:

-----

```

Schema name of the column type:  SYSIBM
Name of column type:            INTEGER
Maximum column length:          4
Scale for decimal column:       0
Number of distinct column values: 4
Average column length:          5
Number of most frequent values:  1
Number of quantiles:           5
Second highest data value:      3
Second lowest data value:       2
Column sequence in partition key: 0
Average number of sub-elements: -1
Average length of delimiters:   -1

```

Column Distribution Statistics:

-----

Frequency Statistics:

Valcount	Value
2	1

Quantile Statistics:

Valcount	Distcount	Value
0	1	1
2	1	1
3	2	2
4	3	3
5	4	4

Column Information:

-----

```

Number:                2
Name:                  C2
Statistics Available:  Yes

```

Column Statistics:

-----

```

Schema name of the column type:  SYSIBM
Name of column type:            INTEGER
Maximum column length:          4
Scale for decimal column:       0
Number of distinct column values: 4

```

Average column length: 5  
 Number of most frequent values: 1  
 Number of quantiles: 5  
 Second highest data value: 3  
 Second lowest data value: 2  
 Column sequence in partition key: 0  
 Average number of sub-elements: -1  
 Average length of delimiters: -1

Column Distribution Statistics:

-----  
 Frequency Statistics:

Valcount	Value
2	1

Quantile Statistics:

Valcount	Distcount	Value
0	0	1
2	0	1
3	0	2
4	0	4
5	0	4

Indexes defined on the table:

-----

Name:	IDX_T1C1C2
Schema:	DB2USER2
Unique Rule:	Duplicate index
Used in Operator:	Yes
Page Fetch Pairs:	Not Available
Number of Columns:	2
Index Leaf Pages:	1
Index Tree Levels:	1
Index First Key Cardinality:	4
Index Full Key Cardinality:	4
Index Cluster Ratio:	100
Index Cluster Factor:	-1.000000
Time of Creation:	2006-06-16-11.46.53.596717
Last Statistics Update:	2006-06-26-12.23.44.814201
Index Sequential Pages:	0
Index First 2 Keys Cardinality:	4
Index First 3 Keys Cardinality:	-1
Index First 4 Keys Cardinality:	-1
Index Avg Gap between Sequences:	0.000000
Fetch Avg Gap between Sequences:	-1.000000
Index Avg Sequential Pages:	0.000000
Fetch Avg Sequential Pages:	-1.000000
Index Avg Random Pages:	1.000000
Fetch Avg Random Pages:	-1.000000
Index RID Count:	5
Index Deleted RID Count:	0
Index Empty Leaf Pages:	0
Avg Partition Cluster Ratio:	-1
Avg Partition Cluster Factor:	-1.000000
Data Partition Cluster Factor:	1.000000
Data Partition Page Fetch Pairs:	Not Available

Base Table Statistics:

-----

Name:	T2
Schema:	DB2USER2
Number of Columns:	3
Number of Pages with Rows:	1
Number of Pages:	1
Number of Rows:	2

```

Table Overflow Record Count:          0
Width of Rows:                       26
Time of Creation:                    2006-06-16-11.46.53.398092
Last Statistics Update:              2006-06-26-12.23.45.157028
Statistics Type:                     Synchronous
Primary Tablespace:                 USERSPACE1
Tablespace for Indexes:             USERSPACE1
Tablespace for Long Data:           NULLP
Number of Referenced Columns:       2
Number of Indexes:                  1
Volatile Table:                     No
Table Active Blocks:                -1
Number of Column Groups:            0
Number of Data Partitions:          1

```

Column Information:

-----

```

Number:                               1
Name:                                  C1
Statistics Available:                 Yes

```

Column Statistics:

-----

```

Schema name of the column type:      SYSIBM
Name of column type:                 INTEGER
Maximum column length:               4
Scale for decimal column:            0
Number of distinct column values:    2
Average column length:               5
Number of most frequent values:      -1
Number of quantiles:                 2
Second highest data value:           2
Second lowest data value:            1
Column sequence in partition key:    0
Average number of sub-elements:      -1
Average length of delimiters:        -1

```

Column Distribution Statistics:

-----

Quantile Statistics:

	Valcount	Distcount	Value
1	1	1	1
2	2	2	2

Column Information:

-----

```

Number:                               2
Name:                                  C2
Statistics Available:                 Yes

```

Column Statistics:

-----

```

Schema name of the column type:      SYSIBM
Name of column type:                 INTEGER
Maximum column length:               4
Scale for decimal column:            0
Number of distinct column values:    2
Average column length:               5
Number of most frequent values:      -1
Number of quantiles:                 2
Second highest data value:           2
Second lowest data value:            1
Column sequence in partition key:    0
Average number of sub-elements:      -1
Average length of delimiters:        -1

```



Column Distribution Statistics:

Quantile Statistics:

Valcount	Distcount	Value
1	0	1
2	0	2

Indexes defined on the table:

```

Name : IDX_T2C1
Schema: DB2USER2
Unique Rule: Duplicate index
Used in Operator: No
Page Fetch Pairs: Not Available
Number of Columns: 1
Index Leaf Pages: 1
Index Tree Levels: 1
Index First Key Cardinality: 2
Index Full Key Cardinality: 2
Index Cluster Ratio: 100
Index Cluster Factor: -1.000000
Time of Creation: 2006-06-16-11.46.53.857520
Last Statistics Update: 2006-06-26-12.23.45.157028
Index Sequential Pages: 0
Index First 2 Keys Cardinality: -1
Index First 3 Keys Cardinality: -1
Index First 4 Keys Cardinality: -1
Index Avg Gap between Sequences: 0.000000
Fetch Avg Gap between Sequences: -1.000000
Index Avg Sequential Pages: 0.000000
Fetch Avg Sequential Pages: -1.000000
Index Avg Random Pages: 1.000000
Fetch Avg Random Pages: -1.000000
Index RID Count: 2
Index Deleted RID Count: 0
Index Empty Leaf Pages: 0
Avg Partition Cluster Ratio: -1
Avg Partition Cluster Factor: -1.000000
Data Partition Cluster Factor: 1.000000
Data Partition Page Fetch Pairs: Not Available
    
```

## EXPLAIN\_FROM\_ACTIVITY プロシージャ - アクティビティ・イベント・モニター情報を使用したステートメントの Explain

EXPLAIN\_FROM\_ACTIVITY プロシージャは、アクティビティ・イベント・モニターから得られるセクションの内容を使用して、ステートメントの特定の実行を Explain します。

Explain の出力は Explain 表に格納され、任意の既存の Explain ツール (例えば **db2exfmt**) を使ってこれを処理できます。 Explain の出力には、存在する場合、アクセス・プランとセクション実行時統計(アクセス・プランの演算子に関するランタイム統計) がどちらも含まれます。

```

▶—EXPLAIN_FROM_ACTIVITY—▶
▶(—appl_id—,—uow_id—,—activity_id—,—activity_evmon_name—,—explain_schema—▶
▶,—explain_requester—,—explain_time—,—source_name—,—source_schema—,—source_version—)▶
    
```

スキーマは SYSPROC です。

## 許可

このルーチンを実行するには、以下のいずれかの権限または特権が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- EXPLAIN 権限

さらに、以下のすべての特権が必要です。

- 指定されたスキーマ内の Explain 表に対する INSERT 特権
- ソース・アクティビティ・イベント・モニターに関するイベント・モニター表に対する SELECT 特権

## デフォルトの PUBLIC 特権

なし

## 表関数パラメーター

### *appl\_id*

タイプ VARCHAR(64) の入力引数。セクションを Explain する対象となるアクティビティを発行したアプリケーションを一意的に識別します。 *appl\_id* が NULL または空ストリングの場合、SQL2032N が戻されます。

### *uow\_id*

タイプ INTEGER の入力引数。セクションを Explain する対象となるアクティビティの作業単位 ID を指定します。作業単位 ID は、特定のアプリケーション内でのみ固有です。 *uow\_id* が NULL である場合、SQL2032N が戻されます。

### *activity\_id*

タイプ INTEGER の入力引数。セクションを Explain する対象となるアクティビティの ID を指定します。アクティビティ ID は、作業単位の中でのみ固有です。 *activity\_id* が NULL である場合、SQL2032N が戻されます。

### *activity\_evmon\_name*

VARCHAR(128) の入力引数。セクションを Explain する対象となるアクティビティを含む、表への書き込みアクティビティ・イベント・モニターの名前を指定します。イベント・モニターが存在しない場合、またはアクティビティ・イベント・モニターでない場合には、SQL0204N が戻されます。イベント・モニターが表への書き込みイベント・モニターでない場合、SQL20502N が戻されます。 *activity\_evmon\_name* が指定されない場合、SQL2032N が戻されます。呼び出し元がアクティビティ・イベント・モニター表に対する SELECT 特権を持っていない場合、SQL0551N が戻されます。

### *explain\_schema*

タイプ VARCHAR(128) のオプションの入力または出力引数。Explain 情報が書き込まれる Explain 表を含むスキーマを指定します。空ストリングまたは NULL を指定した場合、セッション許可 ID のもとで Explain 表が検索され、その後、SYSTOOLS スキーマで検索されます。 Explain 表が見つからない場

合、SQL0219N が戻されます。呼び出し元が Explain 表に対する INSERT 特権を持っていない場合、SQL0551N が戻されます。出力の場合、このパラメーターは、情報が書き込まれた Explain 表を含んでいるスキーマに設定されます。

#### *explain\_requester*

タイプ VARCHAR(128) の出力引数。このルーチンが呼び出された接続のセッション許可 ID が格納されます。

#### *explain\_time*

Explain 要求の開始時刻を格納する、タイプ TIMESTAMP の出力引数。

#### *source\_name*

タイプ VARCHAR (128) の出力引数。ステートメントの準備時またはコンパイル時に実行されていたパッケージの名前を格納します。

#### *source\_schema*

ソース Explain 要求のスキーマまたは修飾子を格納する、タイプ VARCHAR(128) の出力引数。

#### *source\_version*

Explain 要求のソースのバージョンを格納する、タイプ VARCHAR(64) の出力引数。

## 例

以下の例では、一定期間にわたってアクティビティ・イベント・モニターに収集されたデータに対してマイニングを実行していると想定します。以下の照会を使用することで、CPU コストが非常に高い SQL ステートメントの存在に気がきます。

```
SELECT APPL_ID,
       UOW_ID,
       ACTIVITY_ID,
       USER_CPU_TIME
FROM ACTIVITY_A
ORDER BY USER_CPU_TIME
```

以下は、この照会の出力例を示しています。 N2.DB2INST1.0B5A12222841 という ID を持つアプリケーションが非常に多くの CPU 時間を消費しています。

APPL_ID	UOW_ID	ACTIVITY_ID	USER_CPU_TIME
*N2.DB2INST1.0B5A12222841	1	1	92782334234
*N2.DB2INST1.0B5A12725841	2	7	326

2 record(s) selected.

このアクティビティのアクセス・プランを調べる EXPLAIN\_FROM\_ACTIVITY プロシージャを使用して、このアクティビティのチューニング (例えば索引の追加) に効果があるかどうかを判別できます。

```
CALL EXPLAIN_FROM_ACTIVITY( '*N2.DB2INST1.0B5A12222841', 1, 1, 'A', 'MYSHEMA',
?, ?, ?, ?, ? )
```

## 使用上の注意

アクティビティのセクションに対して Explain を実行するには、アクティビティ・データの収集を有効にするときに COLLECT ACTIVITY DATA WITH SECTION 節を指定する必要があります。こうすると、アクティビティ情報と共に

セクションが収集されます。識別されたアクティビティー項目と共にセクションが保管されていない場合には、SQL20501 が戻されます。

アクティビティーに関するセクション実行時統計が収集されなかった場合、セクションの Explain は成功しますが、Explain 出力には actuals 情報が含まれません。以下のようなケースでは、セクション実行時統計が収集されません。

- 入力として指定されたアクティビティーが WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS ストアード・プロシージャを使ってキャプチャーされた。この場合、アクティビティー論理グループの *partial\_record* エレメントの値は 1 です。
- アクティビティー・イベント・モニターの ACTIVITY 表で SECTION\_ACTUALS エレメントが欠落している。
- 実行されるセクションが静的セクションであり、DB2 バージョン 9.7 フィックスパック 1 の適用以降、このセクションがまだ再バインドされていない。
- キャプチャー対象のセクションに関するセクション実行時統計が有効になっていなかった。セクション実行時統計を有効にするには、**section\_actuals** データベース構成パラメーターを使用します。または特定のアプリケーションを対象に、**WLM\_SET\_CONN\_ENV** プロシージャを使用します。デフォルトではセクション実行時統計が無効です。

注: アクティビティーに関するセクション実行時統計が収集されたことを検証するには、ACTIVITY 表の SECTION\_ACTUALS エレメントの長さが 0 より大きいかどうか確認してください。

注: アプリケーション用に **WLM\_SET\_CONN\_ENV** プロシージャによって指定された **section\_actuals** 設定は、直ちに有効になります。セクション実行時統計は、アプリケーションが次に発行するステートメントについて収集されます。

注: パーティション・データベース環境では、アクティビティー・データが収集されるメンバーでのみセクション実行時統計が収集されます。すべてのメンバーで実行時統計を収集するには、必ず COLLECT ACTIVITY DATA ON ALL MEMBERS WITH DETAILS, SECTION 節を使用してアクティビティーを収集してください。特定のアプリケーションを対象にすべてのメンバーでの収集を有効にするには、**WLM\_SET\_CONN\_ENV** プロシージャを呼び出すときに、2 番目の引数に値を「ALL」に設定した *<collectactpartition>* タグを含めることができます。

入力した *appl\_id*、*uow\_id*、および *activity\_id* に一致するアクティビティーが見つからない場合、SQL20501 が戻されます。実行中に

WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS ストアード・プロシージャを使ってアクティビティーが複数回にわたって収集された場合には、複数のアクティビティーがこれらに一致する可能性があります。そのような場合、セクションのキャプチャー対象となった最新の項目が Explain に使用されます。

出力パラメーター *explain\_requester*、*explain\_time*、*source\_name*、*source\_schema*、および *source\_version* はキーを構成し、これを使って Explain 表内のセクションに関する Explain 情報を検索します。セクションから取得された Explain 情報をフォーマット設定するために、既存の任意の Explain ツール (例えば **db2exfmt**) でこれらのパラメーターを使用できます。

EXPLAIN\_FROM\_ACTIVITY プロシージャは、Explain 表への挿入後に COMMIT ステートメントを発行しません。プロシージャの呼び出し元が COMMIT を発行する必要があります。

ACTIVITY\_STMT 論理グループの中には、エレメント STMT\_TEXT、ORIGINAL\_STMT\_TEXT、SECTION\_ENV、EXECUTABLE\_ID、APPL\_ID、ACTIVITY\_ID、UOW\_ID が含まれる必要があります。これらのいずれかのエレメントが欠落している場合、ストアード・プロシージャは SQL206 を戻します。

## EXPLAIN\_FROM\_CATALOG プロシージャ - カタログからのセクション情報を使用したステートメントの Explain

EXPLAIN\_FROM\_CATALOG プロシージャは、カタログから得られるセクションの内容を使用してステートメントを Explain します。Explain の出力は Explain 表に格納され、任意の既存の Explain ツール (例えば `db2exfmt`) を使ってこれを処理できます。

```
▶▶EXPLAIN_FROM_CATALOG(—pkgschema—,—pkgname—,—pkgversion—,—sectno—,—explain_schema—————▶▶
▶,—explain_requester—,—explain_time—,—source_name—,—source_schema—,—source_version—)————▶▶
```

スキーマは SYSPROC です。

### 許可

このルーチンを実行するには、以下のいずれかの権限または特権が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- EXPLAIN 権限

さらに、次の特権が必要です。

- 指定されたスキーマ内の Explain 表に対する INSERT 特権

### デフォルトの PUBLIC 特権

なし

### プロシージャ・パラメーター

#### *pkgschema*

Explain されるセクションを含むパッケージのスキーマを指定する、タイプ VARCHAR(128) の入力引数。 *pkgschema* が NULL または空ストリングの場合、SQL2032N が戻されます。

#### *pkgname*

Explain されるセクションを含むパッケージを指定する、タイプ VARCHAR(128) の入力引数。 *pkgname* が NULL または空ストリングの場合、SQL2032N が戻されます。

### *pkgversion*

Explain されるセクションを含むパッケージのバージョン ID を指定する、タイプ VARCHAR(64) の入力引数。パッケージにバージョンがない場合は、空ストリングを指定します (VARCHAR2 互換モードが有効になっている場合はブランク ' ' 文字)。 *pkgversion* が NULL である場合、SQL2032N が戻されます。

### *sectno*

Explain されるセクションを指定する、タイプ SMALLINT の入力引数。 *sectno* が NULL である場合、SQL2032N が戻されます。

### *explain\_schema*

タイプ VARCHAR(128) のオプションの入力または出力引数。Explain 情報が書き込まれる Explain 表を含むスキーマを指定します。空ストリングまたは NULL を指定した場合、セッション許可 ID のもとで Explain 表が検索され、その後、SYSTOOLS スキーマで検索されます。 Explain 表が見つからない場合、SQL0219N が戻されます。呼び出し元が Explain 表に対する INSERT 特権を持っていない場合、SQL0551N が戻されます。出力の場合、このパラメータは、情報が書き込まれた Explain 表を含んでいるスキーマに設定されます。

### *explain\_requester*

タイプ VARCHAR(128) の出力引数。このルーチンが呼び出された接続のセッション許可 ID が格納されます。

### *explain\_time*

Explain 要求の開始時刻を格納する、タイプ TIMESTAMP の出力引数。

### *source\_name*

タイプ VARCHAR (128) の出力引数。ステートメントの準備時またはコンパイル時に実行されていたパッケージの名前を格納します。

### *source\_schema*

ソース Explain 要求のスキーマまたは修飾子を格納する、タイプ VARCHAR(128) の出力引数。

### *source\_version*

Explain 要求のソースのバージョンを格納する、タイプ VARCHAR(64) の出力引数。

## 例

以下の例は、コンパイル済みでカタログ内に存在する静的ステートメントを Explain する方法を示しています。まず、例えば次のようにして SYSCAT.STATEMENTS カタログ・ビューから選択することにより、セクションを識別できます。

```
SELECT pkgschema,
       pkgname,
       version,
       Sectno
FROM SYSCAT.STATEMENTS
WHERE TEXT = 'select count(*) from syscat.tables'
```

この照会により、以下の例のような出力が戻されます。

PKGSHEMA	PKGNAME	VERSION	SECTNO
NULLID	SQL2G0S		1
NULLID	SQL2G0S	VERSION1	1

2 record(s) selected.

次に、例えば以下のようにして、識別情報 *pkgschema*、*pkgname*、*version* および *sectno* を EXPLAIN\_FROM\_CATALOG プロシージャに渡します。

```
CALL EXPLAIN_FROM_CATALOG( 'NULLID', 'SQL2G0S', ' ', 1, 'MYSHEMA', ?, ?, ?, ?, ? )
```

## 使用上の注意

入力パラメーターに一致するセクションが見つからない場合は SQL20501 が戻されます。

出力パラメーター *explain\_requester*、*explain\_time*、*source\_name*、*source\_schema*、*source\_version* はキーを構成し、これを使って Explain 表内のセクションに関する Explain 情報を検索します。セクションから取得された Explain 情報をフォーマット設定するために、既存の任意の Explain ツール (例えば **db2exfmt**) でこれらのパラメーターを使用できます。

このプロシージャは、Explain 表への挿入後に COMMIT ステートメントを発行しません。プロシージャの呼び出し元が COMMIT を発行する必要があります。

## EXPLAIN\_FROM\_DATA プロシージャ - 入力セクションを使用したステートメントの Explain

EXPLAIN\_FROM\_DATA プロシージャは、入力セクションの内容を使用してステートメントを Explain します。Explain の出力は Explain 表に格納され、任意の既存の Explain ツール (例えば **db2exfmt**) を使ってこれを処理できます。

```
►►EXPLAIN_FROM_DATA(—section—,—stmt_text—,—executable_id—,—explain_schema—►►
►,—explain_requester—,—explain_time—,—source_name—,—source_schema—,—source_version—)►►
```

スキーマは SYSPROC です。

## 許可

このルーチンを実行するには、以下のいずれかの権限または特権が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- EXPLAIN 権限

さらに、次の特権が必要です。

- 指定されたスキーマ内の Explain 表に対する INSERT 特権

## デフォルトの PUBLIC 特権

なし

## プロシージャー・パラメーター

### *section*

Explain 対象のセクションが含まれる、タイプ BLOB(134M) の入力引数。イベント・モニター表、カタログ表など、さまざまなソースからセクションを取得できます。入力セクションが有効なセクションでない場合、SQL20503N が戻されます。

### *stmt\_text*

オプションの、タイプ CLOB(2M) の入力引数。入力セクションに対応するステートメントのテキストが含まれます。 *stmt\_text* が NULL の場合、フォーマット済み Explain 出力にはステートメント・テキストが含まれません。

### *executable\_id*

オプションの、タイプ VARCHAR(32) FOR BIT DATA の入力引数。セクションの識別に使用される実行可能 ID が含まれます。 *executable\_id* が NULL の場合、フォーマット済み Explain 出力には実行可能 ID が含まれません。

### *explain\_schema*

タイプ VARCHAR(128) のオプションの入力または出力引数。Explain 情報が書き込まれる Explain 表を含むスキーマを指定します。空ストリングまたは NULL を指定した場合、セッション許可 ID のもとで Explain 表が検索され、その後、SYSTOOLS スキーマで検索されます。 Explain 表が見つからない場合、SQL0219N が戻されます。呼び出し元が Explain 表に対する INSERT 特権を持っていない場合、SQL0551N が戻されます。出力の場合、このパラメーターは、情報が書き込まれた Explain 表を含んでいるスキーマに設定されます。

### *explain\_requester*

タイプ VARCHAR(128) の出力引数。このルーチンが呼び出された接続のセッション許可 ID が格納されます。

### *explain\_time*

Explain 要求の開始時刻を格納する、タイプ TIMESTAMP の出力引数。

### *source\_name*

タイプ VARCHAR (128) の出力引数。ステートメントの準備時またはコンパイル時に実行されていたパッケージの名前を格納します。

### *source\_schema*

ソース Explain 要求のスキーマまたは修飾子を格納する、タイプ VARCHAR(128) の出力引数。

### *source\_version*

Explain 要求のソースのバージョンを格納する、タイプ VARCHAR(64) の出力引数。

## 例

パッケージ・キャッシュ・イベント・モニターを使って多数のステートメントをキャプチャーした後、(EVMON\_FORMAT\_UE\_TO\_TABLE ストアード・プロシージャーを使って) PKGCACHE という表にイベント・モニター・データを抽出したとします。表のデータを調べてみると、非常にコストが高い、実行可能 ID 「x'0100000000000000070000000000000000000000000000000000200200811261904103698」 のステートメントが見つかりました。



このステートメントのアクセス・プランを把握するために EXPLAIN\_FROM\_DATA プロシージャを発行し、PKG\_CACHE 表の項目からのセクションを入力として渡します。MYSCHEMA スキーマの Explain 表の中に Explain 出力を書き込みます。

```
SET SERVEROUTPUT ON;

BEGIN
  DECLARE EXECUTABLE_ID VARCHAR(32) FOR BIT DATA; --
  DECLARE SECTION BLOB(134M); --
  DECLARE STMT_TEXT CLOB(2M); --
  DECLARE EXPLAIN_SCHEMA VARCHAR(128); --

  DECLARE EXPLAIN_REQUESTER VARCHAR(128); --
  DECLARE EXPLAIN_TIME TIMESTAMP; --
  DECLARE SOURCE_NAME VARCHAR(128); --
  DECLARE SOURCE_SCHEMA VARCHAR(128); --
  DECLARE SOURCE_VERSION VARCHAR(128); --

  SET EXPLAIN_SCHEMA = 'MYSCHEMA'; --

  SELECT P.SECTION, P.STMT_TEXT, P.EXECUTABLE_ID INTO
    SECTION, STMT_TEXT, EXECUTABLE_ID
  FROM PKG_CACHE WHERE EXECUTABLE_ID =
    x'0100000000000000000000700000000000000000000000000000000200200811261904103698'; --

  CALL EXPLAIN_FROM_DATA( SECTION,
    STMT_TEXT,
    EXECUTABLE_ID,
    EXPLAIN_SCHEMA,
    EXPLAIN_REQUESTER,
    EXPLAIN_TIME,
    SOURCE_NAME,
    SOURCE_SCHEMA,
    SOURCE_VERSION ); --

  CALL DBMS_OUTPUT.PUT( 'EXPLAIN_REQUESTER = ' ); --
  CALL DBMS_OUTPUT.PUT_LINE( EXPLAIN_REQUESTER ); --
  CALL DBMS_OUTPUT.PUT( 'EXPLAIN_TIME = ' ); --
  CALL DBMS_OUTPUT.PUT_LINE( EXPLAIN_TIME ); --
  CALL DBMS_OUTPUT.PUT( 'SOURCE_NAME = ' ); --
  CALL DBMS_OUTPUT.PUT_LINE( SOURCE_NAME ); --
  CALL DBMS_OUTPUT.PUT( 'SOURCE_SCHEMA = ' ); --
  CALL DBMS_OUTPUT.PUT_LINE( SOURCE_SCHEMA ); --
  CALL DBMS_OUTPUT.PUT( 'SOURCE_VERSION = ' ); --
  CALL DBMS_OUTPUT.PUT_LINE( SOURCE_VERSION ); --
END;

SET SERVEROUTPUT OFF;
```

### 使用上の注意

入力セクションは、以下のようなさまざまなソースから入手可能です。

- アクティビティ・イベント・モニター
- パッケージ・キャッシュ・イベント・モニター
- カタログ表
- 上記のいずれかの場所からセクションをコピーした、任意のユーザー表または入力ソース。

出力パラメーター *explain\_requester*、*explain\_time*、*source\_name*、*source\_schema*、*source\_version* はキーを構成し、これを使って Explain 表内のセクションに関する

Explain 情報を検索します。セクションから取得された Explain 情報をフォーマット設定するために、既存の任意の Explain ツール (例えば **db2exfmt**) でこれらのパラメーターを使用できます。

このプロシージャは、Explain 表への挿入後に COMMIT を発行しません。プロシージャの呼び出し元が COMMIT を発行する必要があります。

### EXPLAIN\_FROM\_SECTION プロシージャ - パッケージ・キャッシュまたはパッケージ・キャッシュ・イベント・モニター情報を使用したステートメントの Explain

EXPLAIN\_FROM\_SECTION プロシージャは、パッケージ・キャッシュまたはパッケージ・キャッシュ・イベント・モニターから得られるセクションの内容を使ってステートメントを Explain します。Explain の出力は Explain 表に格納され、任意の既存の Explain ツール (例えば **db2exfmt**) を使ってこれを処理できます。

```

▶▶—EXPLAIN_FROM_SECTION—————▶▶
▶—(—executable_id—,—section_source_type—,—section_source_name—,—member—,—explain_schema—————▶
▶,—,—explain_requester—,—explain_time—,—source_name—,—source_schema—,—source_version—)—▶▶

```

スキーマは SYSPROC です。

#### 許可

このルーチンを実行するには、以下のいずれかの権限または特権が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- EXPLAIN 権限

さらに、以下のすべての特権が必要です。

- 指定されたスキーマ内の Explain 表に対する INSERT 特権
- セクション・ソース名がパッケージ・キャッシュ・イベント・モニターを識別する場合には、パッケージ・キャッシュ・イベント・モニター表に対する SELECT 特権

#### デフォルトの PUBLIC 特権

なし

#### プロシージャ・パラメーター

*executable\_id*

Explain されるセクションを一意的に識別する、タイプ VARCHAR(32) FOR BIT DATA の入力引数。この引数が NULL または空ストリングである場合、SQL2032 が戻されます。

### *section\_source\_type*

Explain されるセクションのソースを指定する、タイプ CHAR(1) の入力引数。有効な値は以下のとおりです。

- M - メモリー内のパッケージ・キャッシュからセクションが取得されます
- P - パッケージ・キャッシュ・イベント・モニターからセクションが取得されます。通常の表、または未フォーマット・イベント表のどちらかをイベント・モニターの出力タイプとして使用できます。

静的 SQL では、*section\_source\_type* が M でセクションがパッケージ・キャッシュに存在しない場合、カタログ表の中でセクションが検索されます。

### *section\_source\_name*

VARCHAR(128) の入力引数。*section\_source\_type* が P の場合はパッケージ・キャッシュ・イベント・モニターの名前を指定します。*section\_source\_type* が M の場合は、オプションで、パッケージ・キャッシュ・イベント・モニターの名前を指定できます。パッケージ・キャッシュ内にセクションが見つからない場合 (例えば EXPLAIN\_FROM\_SECTION ストアド・プロシージャの呼び出し前にセクションがパッケージ・キャッシュからフラッシュされた場合)、イベント・モニターの中でセクションが検索されます。ソース入力イベント・モニターが、COLLECT DETAILED DATA オプションを使って作成されたパッケージ・キャッシュ・イベント・モニターではない場合、SQL0204N が戻されます。呼び出し元がパッケージ・キャッシュ・イベント・モニター表に対する SELECT 特権を持っていない場合、SQL0551N が戻されます。

### *member*

タイプ INTEGER の入力引数。*section\_source\_type* が M の場合、Explain 対象のセクションがメモリー内で格納されているメンバーを指定します。-1 を指定した場合、プロシージャは現在のコーディネーター・メンバーおよびセクション・コンパイル・メンバーの中でセクションを検索します。*section\_source\_type* が M 以外の場合には、この引数は無視されます。

### *explain\_schema*

タイプ VARCHAR(128) のオプションの入力または出力引数。Explain 情報が書き込まれる Explain 表を含むスキーマを指定します。空ストリングまたは NULL を指定した場合、セッション許可 ID のもとで Explain 表が検索され、その後、SYSTOOLS スキーマで検索されます。Explain 表が見つからない場合、SQL0219N が戻されます。呼び出し元が Explain 表に対する INSERT 特権を持っていない場合、SQL0551N が戻されます。出力の場合、このパラメータは、情報が書き込まれた Explain 表を含んでいるスキーマに設定されます。

### *explain\_requester*

タイプ VARCHAR(128) の出力引数。このルーチンが呼び出された接続のセッション許可 ID が格納されます。

### *explain\_time*

Explain 要求の開始時刻を格納する、タイプ TIMESTAMP の出力引数。

### *source\_name*

タイプ VARCHAR (128) の出力引数。ステートメントの準備時またはコンパイル時に実行されていたパッケージの名前を格納します。

### source\_schema

ソース Explain 要求のスキーマまたは修飾子を格納する、タイプ VARCHAR(128) の出力引数。

### source\_version

Explain 要求のソースのバージョンを格納する、タイプ VARCHAR(64) の出力引数。

## 例

この例は、セクションごとに入手可能なモニタリング・メトリックを調べること  
で、パッケージ・キャッシュ内の非常にコストが高いステートメントを識別して分  
析する方法を示しています。まず、セクションによる CPU 時間の使用状況を判別  
するために、以下の SELECT ステートメントのような照会を発行します。

```
SELECT executable_id,  
       Total_cpu_time,  
       Varchar(stmt_text, 100) as stmt_text  
FROM TABLE(MON_GET_PKG_CACHE_STMT (NULL, NULL, NULL, -1)) AS T
```

以下は、この照会の出力例を示しています。

EXECUTABLE_ID	TOTAL_CPU_TIME	...
x'0100000000000000012...200200811261904103698'	91875622	...
x'0100000000000000007...200200811261904103238'	300	...

2 record(s) selected.

以下は、この照会の出力例の続きを示しています。

```
...STMT_TEXT  
.....  
...SELECT * FROM SYSCAT.TABLES  
...INSERT INTO T1 VALUES(123)
```

2 record(s) selected.

高コストの SELECT ステートメントのアクセス・プランを調べるには、その  
*executable\_id* を EXPLAIN\_FROM\_SECTION プロシージャに渡します。出力を  
MYSCHEMA スキーマの Explain 表に入れます。

```
CALL EXPLAIN_FROM_SECTION  
 ( x'0100000000000000012000000000000000000000000000000000200200811261904103698',  
   'M', NULL, 0, 'MYSCHEMA', '?', '?', '?', '?', ? )
```

## 使用上の注意

入力の実行可能 ID に一致するセクションが見つからない場合、SQL20501 が戻さ  
れます。入力 *executable\_id* は、以下のソースから入手可能です。

- アクティビティ・イベント・モニター
- パッケージ・キャッシュ・イベント・モニター
- MON\_GET\_ACTIVITY\_DETAILS 表関数
- MON\_GET\_PKG\_CACHE\_STMT 表関数
- WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 表関数
- WLM\_GET\_SERVICE\_CLASS\_AGENTS 表関数
- MON\_GET\_PKG\_CACHE\_STMT\_DETAILS 表関数

- MON\_GET\_APPL\_LOCKWAIT 表関数

出力パラメーター *explain\_requester*、*explain\_time*、*source\_name*、*source\_schema*、*source\_version* はキーを構成し、これを使って Explain 表内のセクションに関する情報を検索します。セクションから取得された Explain 情報をフォーマット設定するために、既存の任意の Explain ツール (例えば **db2exfmt**) でこれらのパラメーターを使用できます。

このプロシージャは、Explain 表への挿入後に COMMIT ステートメントを発行しません。プロシージャの呼び出し元が COMMIT を発行する必要があります。

## モニター・ルーチン

モニター・ルーチンは、「MON」で始まる名前 (例えば MON\_GET\_SERVICE\_SUBCLASS) または「EVMON」で始まる名前 (例えば EVMON\_FORMAT\_UE\_TO\_TABLES) を持つルーチンです。

これらのルーチンは、システムの状況照会やモニター・データ操作に関連したさまざまな機能を実行します。一部のルーチンは、表形式のデータを返す表関数です。例えば、DB2 バージョン 9.7 で導入されたモニター・インフラストラクチャーで使用できるモニター・エレメントに対するアクセスを提供する表関数がいくつかあります。ビューのルーチンもあります。ビューは、表関数と非常に似ており、表形式のデータを返しますが、表関数と違って入力パラメーターを必要としません。

「EVMON」で始まる一部のモニター・ルーチンは、データの形式を変換することに注意してください。スナップショット関数など他の特定のルーチンも、モニター情報を戻します。

「MON」ルーチンは戦略的に重要であるため、このルーチンの名前は将来のリリースでも変更されません。ただし、将来のリリースで機能が拡張される場合には、新しい出力列が追加されます。したがって、組み込みルーチンまたはビューを使って情報を取得するために照会を発行するとき、SELECT \* ... という形式のステートメントを使用しないでください。その代わりに、SELECT ステートメントで結果列を指定します。こうすることで、アプリケーションは結果列の数と、それらが戻される順序を制御することができます。

## モニター (MON) 表関数

すべての表関数には、共通のモニター・エレメントの集合が組み込まれます。これらのエレメントは、アプリケーション応答時間に影響する可能性のあるさまざまなシステム・パフォーマンス標識の集合に関する情報を提供します。関心あるワークロードのサブセットに関するモニター・データを入手することもできます。

例えば、次のようなモニター表関数は、システム・ワークロード全体のさまざまな局面に関して報告します。

- MON\_GET\_CONNECTION および MON\_GET\_CONNECTION\_DETAILS
- MON\_GET\_SERVICE\_SUBCLASS および MON\_GET\_SERVICE\_SUBCLASS\_DETAILS
- MON\_GET\_UNIT\_OF\_WORK および MON\_GET\_UNIT\_OF\_WORK\_DETAILS
- MON\_GET\_WORKLOAD および MON\_GET\_WORKLOAD\_DETAILS

- MON\_GET\_PKG\_CACHE\_STMT および  
MON\_GET\_PKG\_CACHE\_STMT\_DETAILS

これらの表関数には 2 つのバージョンがあります。その 1 つの接尾部は `_DETAILS` になります。 `_DETAILS` 接尾部のないバージョンは、最も一般的に使用されるデータを戻すリレーショナル SQL インターフェースを提供します。 `_DETAILS` 接尾部のあるバージョンは、モニター・データに対する XML ベースのアクセスを提供し、より包括的なデータ・セットを戻します。

このほかに、例えば次のような表関数は、特定のタイプのデータ・オブジェクトに関するデータを戻します。

- MON\_GET\_APPL\_LOCKWAIT
- MON\_GET\_BUFFERPOOL
- MON\_GET\_CONTAINER
- MON\_GET\_EXTENDED\_LATCH\_WAIT
- MON\_GET\_INDEX
- MON\_GET\_LOCKS
- MON\_GET\_PAGE\_ACCESS\_INFO
- MON\_GET\_TABLE
- MON\_GET\_TABLESPACE

これらの表関数は、特定のデータ・オブジェクトに関連したパフォーマンスの問題を調査するのに使用します。

サブシステムのモニターには、以下のその他の表関数が役立ちます。

- MON\_GET\_FCM
- MON\_GET\_FCM\_CONNECTION\_LIST
- MON\_GET\_HADR
- MON\_GET\_SERVERLIST
- MON\_GET\_TRANSACTION\_LOG

さらに別の表関数は、個々のアクティビティおよびステートメントの詳細を調べるのに役立ちます。

- `MON_GET_ACTIVITY_DETAILS` は、システム上で現在実行中の特定のアクティビティに関する詳細を戻します。これらの詳細には、一般的なアクティビティ情報 (ステートメント・テキストなど) や一連のメトリックが含まれます。
- `MON_GET_INDEX_USAGE_LIST` は、索引に定義されている使用量リストから情報を戻します。
- `MON_GET_TABLE_USAGE_LIST` は、表に定義されている使用量リストから情報を戻します。

さらに、以下の表関数は進捗モニターの役割を果たします。

- `MON_GET_AUTO_MAINT_QUEUE` は、オートノミック・コンピューティング・デーモン (`db2acd`) で実行するために現在キューに入れられている自動保守ジョブすべてに関する情報を戻します。

- `MON_GET_AUTO_RUNSTATS_QUEUE` は、現在接続しているデータベースでの自動統計収集によって評価するために現在キューに入れられているオブジェクトすべてに関する情報を戻します。
- `MON_GET_EXTENT_MOVEMENT_STATUS` はエクステント移動操作の状況を戻します。
- `MON_GET_REBALANCE_STATUS` は、表スペースに対する再平衡化操作の状況を戻します。
- `MON_GET_RTS_RQST` は、システム内で保留中のすべてのリアルタイム統計要求と、リアルタイム統計デーモンによって現在処理されている要求のセットに関する情報を戻します。
- `MON_GET_USAGE_LIST_STATUS` は、使用量リストの現在の状況を戻します。

`MON_FORMAT_` で始まる表関数は、読みやすい行ベースの形式で情報を戻します。 `MON_FORMAT_LOCK_NAME` はロックの内部バイナリー名を入力として、そのロックに関する詳細情報を戻します。 `MON_FORMAT_XML_` で始まる表関数は、いずれかの `MON_GET_*_DETAILS` 表関数によって (あるいは統計、アクティビティ、作業単位、またはパッケージ・キャッシュ・イベント・モニターの出力から) 戻される XML メトリック文書を入力として受け入れ、フォーマット設定された行ベースの出力を戻します。

- `MON_FORMAT_XML_COMPONENT_TIMES_BY_ROW` はコンポーネント時間に関するフォーマット設定された行ベースの出力を戻します。
- `MON_FORMAT_XML_METRICS_BY_ROW` はすべてのメトリックに関するフォーマット設定された行ベースの出力を戻します。
- `MON_FORMAT_XML_TIMES_BY_ROW` は、待機時間と処理時間の結合された階層に関するフォーマット設定された行ベースの出力を戻します。
- `MON_FORMAT_XML_WAIT_TIMES_BY_ROW` 表関数は、待機時間に関するフォーマット設定された行ベースの出力を戻します。

## モニター (MON) 表関数の特性

- モニター表関数によって戻されるメトリックがリセットされることはありません。データベースがアクティブにされる時点で 0 で開始され、データベースが非アクティブ化されるまで継続的に累算されます。
- ほとんどの表関数と同様に、受信データの対象を単一のオブジェクト (サービス・クラス「A」など) にするか、すべてのオブジェクトにするかを選択できます。
- ほとんどの表関数と同様に、パーティション・データベース環境でこれらの表関数を使用する際には、受信データの対象を単一のパーティションにするか、すべてのパーティションにするかを選択できます。すべてのパーティションに関するデータの受信を選択すると、表関数はパーティションごとに 1 行ずつ戻します。パーティション全体の値を加算して、パーティション全体にわたるモニター・エレメントの値を取得できます。

## モニター (MON) ビュー

モニター・ビューは、例えば次のような、さまざまなデータベース・アクティビティに関するメトリックを戻します。

- `MON_CURRENT_SQL` は、データベースの全メンバーに対してサブミットされた、まだ未完了であるすべてのアクティビティに関するメトリックを戻します (現在実行中の SQL ステートメントのポイント・イン・タイム・ビューを含む)。
- `MON_DB_SUMMARY` はすべてのサービス・クラスにわたって集約されたメトリックを戻します。
- `MON_LOCKWAITS` は、現在接続しているデータベースでのロック取得を待機しているアプリケーションのために機能するエージェントについての情報を戻します。
- `MON_SERVICE_SUBCLASS_SUMMARY` はすべてのサービス・サブクラスに関するメトリックを戻し、サービス・クラスごとに実行された作業を示します。
- `MON_CURRENT_UOW` はすべての作業単位に関するメトリックを戻します。
- `MON_WORKLOAD_SUMMARY` はすべてのワークロードに関するメトリックを戻し、ワークロードごとに受信される作業を示します。

## イベント・モニター (EVMON) ルーチン

イベント・モニター・ルーチンは、イベントを未フォーマット・イベント表に書き込むイベント・モニターからデータを抽出してフォーマットします。LOCKING および UNIT OF WORK イベント・モニター・タイプは、不定形式のイベント表を使用します。ルーチン名は、次のとおりです。

- `EVMON_FORMAT_UE_TO_XML` 表関数
- `EVMON_FORMAT_UE_TO_TABLES` プロシージャ

イベント・モニター・ルーチンを使用すると、`EVMON_FORMAT_UE_TO_XML` 表関数を使用して XML 文書を介するか、または `EVMON_FORMAT_UE_TO_TABLES` プロシージャを使用してリレーショナル表を介することにより、イベント・モニター・データにアクセスできます。

## `EVMON_FORMAT_UE_TO_TABLES` プロシージャ - XML 文書をリレーショナル表へ移動する

`EVMON_FORMAT_UE_TO_TABLES` プロシージャは、イベント・モニターによって作成された未フォーマット・イベント (UE) 表に保管されたデータを取り出し、一連のリレーショナル表に変換します。

リレーショナル表の作成処理には 2 つのステップがあります。最初は、`EVMON_FORMAT_UE_TO_XML` 表関数を使用して、UE 表にあるデータを XML 形式に変換します。この表関数は、`EVMON_FORMAT_UE_TO_TABLES` プロシージャの実行の一部として、自動的に実行されます。次に、イベント・モニター・データを含む XML 文書を、XML 分解を使用してリレーショナル表に入れます。

### 構文

```

▶▶—EVMON_FORMAT_UE_TO_TABLES—(—evmon_type—,—xsrschema—,——————▶
▶—xsrobjectname—,—xmlschemafilename—,—tabschema—,——————▶
▶—tbsp_name—,—options—,—commit_count—,—fullselect—)—————▶▶

```

スキーマは `SYSPROC` です。



## プロシージャー・パラメーター

### *evmon\_type*

フォーマットされていないイベント表に保管されているデータのタイプを表す、タイプ VARCHAR(128) の入力パラメーター。可能な値は、以下のとおりです。

#### *LOCKING*

未フォーマット・イベント表に保管されているデータは、ロック・イベント・モニターからのものです。

#### *PKGCACHE*

フォーマットされていないイベント表に保管されているデータは、PACKAGE CACHE イベント・モニターからのものです。

*UOW* フォーマットされていないイベント表に保管されているデータは、UOW イベント・モニターからのものです。

### *xsrschema*

UE ファイルからのデータがどのように表の列に対応するかを示す XSR オブジェクトの名前の最初の部分を指定する VARCHAR (128) タイプの入力パラメーター。XSR オブジェクト名の 2 番目の部分は *xsubjectname* パラメーターから導出されます。完全な XSR オブジェクト名は、*xsrschema.xsubjectname* と定義されます。この値が NULL の場合は、現行セッションのユーザーの許可 ID が使用されます。

### *xsubjectname*

UE ファイルからのデータがどのように表の列に対応するかを示す XSR オブジェクトの名前の 2 番目の部分を指定する VARCHAR (128) タイプの入力パラメーター。XSR オブジェクト名の最初の部分は、*xsrschema* パラメーターから導出されます。完全な XSR オブジェクト名は *xsrschema.xsubjectname* と定義され、XSR 内のすべてのオブジェクト間で固有です。この値が NULL の場合、*xsubjectname* は *EVMON\_<evmon\_type>\_SCHEMA\_<SQL release level>* として導出されます。例えば、DB2 バージョン 9.7 のロック・イベント・モニターは、*EVMON\_LOCKING\_SCHEMA\_SQL09070* という導出された *xsrname* を持ちます。

XSR オブジェクトは、イベント・モニターの出力を説明する XML スキーマ・ファイルのコピーです。これは、XML スキーマ・リポジトリ (XSR) に格納され、*EVMON\_FORMAT\_UE\_TO\_TABLES* 処理の最初の段階で作成される一時 XML 文書の要素と、最終的にプロシージャーが作成する表および列の間のリレーションシップを定義します。XSR オブジェクトは、作成されるすべての表とそれらの表を派生させる XML スキーマとの間の相互依存関係の管理を行うためにも使用されます。XSR オブジェクトがドロップされるか、プロシージャーにより作成されたいずれかの表がドロップされたり列が変更されたりした場合、これら 2 つのものの間の依存関係は「壊れた」と言います。

*EVMON\_FORMAT\_UE\_TO\_TABLES* (または *EVMON\_FORMAT\_UE\_TO\_XML* 表関数) が特定のタイプのイベント・モニターの UE ファイルに対してまだ実行されていない場合、イベント・モニター出力を説明する XSR オブジェクトはまだ存在していません。この場合、イベント・モニター用の XML スキーマ・ファイルが使用されて、システム・カタログ表に XSR オブジェクトを作成し登録します。

データベースを新しいリリースにアップグレードした場合、リレーショナル表と XML スキーマの間の従属関係を維持するためには、元の *xsobjectname* を明示的に指定する必要があります。

#### *xmlschemafile*

イベント・モニターにより作成された出力を説明するディスク上の XML スキーマ文書の絶対パスを表わす、タイプ VARCHAR (1024) の入力パラメーター。XML スキーマ文書要素には、XML 要素と属性をリレーショナル表とその列にマップする情報でアノテーションが付けられています。

このパラメーターは、XSR オブジェクトを登録するために使用されます。*evmon\_type* で指定されているイベント・モニターのタイプに関して登録されていて使用可能になっている XSR オブジェクトが存在しない場合、XSR オブジェクトは以下のように登録されます。

- *xmlschemafile* が NULL の場合、*evmon\_type* に指定されている値に応じて、以下のように、プロシーチャーはディスク上の XML スキーマ・ファイルを使用します。

#### *LOCKING*

sqllib/misc/DB2EvmonLocking.xsd

#### *PKGCACHE*

sqllib/misc/DB2EvmonPkgCache.xsd

#### *UOW* sqllib/misc/DB2EvmonUOW.xsd

- XML スキーマ・ファイル名を指定した場合、分解のための XSR オブジェクトを登録し使用可能にするためにそのファイルが使用されます。
- *xrschema* および *xsobjectname* パラメーターの値を指定した場合、これらの名前を使用して XSR オブジェクトが作成されます。指定されていない場合、XSR オブジェクトの名前は、以前 *xsobjectname* について説明したように、デフォルトを使用して付けられます。

**重要:** 以前、XSR オブジェクトが分解のために登録されて使用可能になっている場合、このパラメーターは無視されます。別の XML スキーマ・ファイルを使用する XSR オブジェクトを登録した場合、まず、既存の XSR オブジェクトをドロップする必要があります。

#### *tabschema*

イベント・モニターのリレーショナル表が作成される SQL スキーマ名を表す、タイプ VARCHAR (128) の入力パラメーター。この値が NULL の場合は、現行セッションのユーザーの許可 ID が使用されます。表が作成される SQL スキーマは、次のように判別されます。

- <db2-xdb:SQLSchema> が指定される場合には、このスキーマを使用します。
- <db2-xdb:defaultSchema> が指定される場合には、このスキーマを使用します。
- これらのどちらの値も指定されない場合には、*sqlschema* 入力パラメーターからの値を使用します。

**注:** XML スキーマが分解用に登録されると、XSR スキーマ・リポジトリは、スキーマ内で参照される各表とこのスキーマに対応する XSR オブジェクトの従属関係を作成します。これは、データベース内のリレーショナル表の固有のセッ

トに XSR オブジェクト名がリンクされることを意味します。既存の XSR オブジェクトを参照すると、そのデータは必ず分解され、XSR オブジェクトがリンクされた表に挿入されます。

#### *tblsp\_name*

リレーショナル表が作成される表スペースを示す、タイプ VARCHAR(128) の入力パラメーター。このパラメーターのデフォルト値は NULL です。なお、XML スキーマ・ファイル内の CREATE TABLE ステートメントで指定された表スペース名は、この入力パラメーターよりも優先されることに注意してください。

#### *options*

この表関数でサポートされるキーワード・オプションのリストを表す、タイプ VARCHAR (1024) の入力パラメーター。各オプションはセミコロン (;) 文字で区切られている必要があります。使用できる値は次のとおりです。

#### **RECREATE\_FORCE**

分解の前にリレーショナル表をドロップして再作成します。

#### **RECREATE\_ONERROR**

以下の状況では、リレーショナル表をドロップして再作成します。

1. XSR オブジェクトは登録されていないが、表が存在する場合。
2. 最初に失敗した分解の試行の際。その後の失敗は戻され、表を再作成する試みは行われません。

例えば表スペースの満杯エラーや許可エラーなど、エラーが発生した場合にプロシージャは、分解プロシージャから戻される SQLCODE をフィルターに掛けません。プロシージャは負の SQLCODE すべてを等しく扱い、表の再作成を試みます。

#### **PRUNE\_UE\_TABLE**

リレーショナル表に正常に挿入されたバイナリー・イベントをすべて UE 表からプルーニングする (つまり削除する) ことを示します。プルーニングは、リレーショナル表への挿入が実行される同じ作業単位の中で行われます。

#### **UPGRADE\_TABLES**

このプロシージャによって生成されたリレーショナル表を変更して、現行リリースの XSR オブジェクト *xsobjectname* で定義されている表定義と一致するようにすることを示します。以前のリリースで作成されたリレーショナル表をアップグレードして、現行リリースで加えた変更を反映する場合に、このパラメーターを指定します。リリース間では、以下のタイプの変更が生じる可能性があります。

- 新しい列が表に追加される
- 新しい表がイベント・モニターの出力に追加される
- 列定義 (例えば、データ・タイプや長さ) が変更される

UPGRADE\_TABLES オプションを使用しない場合、既存の表定義が保持されます。現在のリリースで追加された新しい列または表のデータは、リレーショナル表にまったく書き込まれません。

UPGRADE\_TABLES を指定する場合は、元の *xsobjectname* も明示的に指定する必要があります。

### *commit\_count*

タイプ INTEGER の入力パラメーター。可能な値は、以下のとおりです。

- 1 100 文書が正常に分解されるごとにコミットします。-1 がデフォルト値です。
- 0 コミットなし。
- n* *n* 個の文書が正常に分解されるごとにコミットします。

### *fullselect*

フォーマットされていないイベント表からの全選択ステートメントを表す、タイプ CLOB(2M) の入力パラメーター。全選択ステートメントは、SELECT ステートメントの規則に準拠する照会です。この照会は、以下の規則に従っていなければなりません。

- この照会は、"\*" 節を使用するか、未フォーマット・イベント表のすべての列を指定する必要があります。それ以外の場合、エラーが戻されます。列は、未フォーマット・イベント表の DESCRIBE ステートメントで戻されたのと同じ順序で指定されている必要があります。
- この照会は、フォーマットされていないイベント表からのみ選択を行う必要があります。
- WHERE 節は、フォーマットされていないイベント表の非 LOB 列のいずれかを使用して、イベントをフィルターで除外することができます。

## 許可

EVMON\_FORMAT\_UE\_TO\_TABLES ストアード・プロシージャに対する EXECUTE 特権。

未フォーマット・イベント表に対する SELECT 特権 (表を作成していない場合)。

指定した SQL スキーマのリレーショナル表を作成するための CREATE 特権。

リレーショナル表に挿入するための INSERT 特権 (それらの表を作成していない場合)。

XDB\_DECOMP\_XMP\_FROM\_QUERY プロシージャが必要とするすべての特権。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 使用上の注意

### UE 表のレコードと EVMON\_FORMAT\_UE\_TO\_TABLES 表関数の出力の関係

UE 表に書き込まれるレコードと EVMON\_FORMAT\_UE\_TO\_TABLES プロシージャの出力の間のマッピングは、1 対 1 ではありません。UE 表に複数のレコードを生成するイベントもありますし、1 つのレコードだけが追加されるものもあります。リレーショナル表にデータを書き込むとき、EVMON\_FORMAT\_UE\_TO\_TABLES プロシージャは、複数の UE 表レコードにある情報を単一のリレーショナル表に結合させる場合もありますし、複数の行を異なる出力表に生成する場合もあります。

## 表の作成

分解が行われるためには、一連のリレーショナル表が存在しなければなりません。EVMON\_FORMAT\_UE\_TO\_TABLES プロシージャは、以下のよう  
にリレーショナル表を自動的に作成します。

- プロシージャは、<db2-mon:createStmt> エレメントを見つけるために、イベント・モニターの XML スキーマ・ファイルを解析します。各エレメントには、完全な CREATE TABLE ステートメントが含まれています。
- プロシージャは CREATE TABLE ステートメントを抽出して実行します。

<db2-mon:createStmt> は、既存の <db2-xdb:table> エレメントの子エレメントです。この子エレメントを認識して使用できるのは、EVMON\_FORMAT\_UE\_TO\_TABLES プロシージャだけです。XML スキーマ・ファイルを解析する XSR オブジェクトなどの他のプロシージャはすべて、このエレメントを無視します。

<db2-mon:createStmt> 内の表名を修飾してはなりません。

## 各リリースに対応した XML スキーマ・ファイル

イベント・モニターごとに用意されているデフォルトの XML スキーマ・ファイルは、現行リリース用の XML スキーマを常に反映しています。それで、EVMON\_FORMAT\_UE\_TO\_TABLES (または EVMON\_FORMAT\_UE\_TO\_XML) を実行する際には、出力はそのリリースのイベント・モニター用に定義されているモニター要素を反映します。次のセクションでは、イベント・モニター用のスキーマ・ファイルが時間と共に変更された場合に何が起るかを説明します。

EVMON\_FORMAT\_UE\_TO\_TABLES プロシージャを使用して表を作成してから、フィックスパックを適用したり新規リリースにアップグレードしたりする場合、これらの変更の影響について理解することが重要です。

## EVMON\_FORMAT\_UE\_TO\_TABLES により作成される表へのスキーマ更新の影響

将来のフィックスパックまたはリリースでは、新規モニター要素がイベント・モニターに追加されるものと思われます。これらの新規モニター要素により、新しい列また新規表を EVMON\_FORMAT\_UE\_TO\_TABLES プロシージャが作成することになるかもしれません。しかし、フィックスパックを適用したり新規リリースにアップグレードしたりする前に既にこのプロシージャで作成した表がある場合、以下のことを行って新規リレーショナル列または表を作成する必要があります。

### フィックスパック更新の場合

最新のフィックスパックのインストール前に

EVMON\_FORMAT\_UE\_TO\_TABLES により作成されたりリレーショナル表がまだ存在する場合、リレーショナル形式で新規モニター要素を表示したいならば、フィックスパックで出荷された新規スキーマに基づいて一連の新しい表を作成させなければなりません。

EVMON\_FORMAT\_UE\_TO\_TABLES プロシージャーに、フィックスバックで出荷された新規スキーマを強制的に使用させて新規表を作成させるには、以下のステップを行います。

1. 現登録バージョンの XML スキーマ (スキーマの登録について詳しくは、EVMON\_FORMAT\_UE\_TO\_TABLES プロシージャーの *tabschema* パラメーターの注を参照してください) と既存の表の間の依存関係を、以下のいずれかを行って切断してください。
  - EVMON\_FORMAT\_UE\_TO\_TABLES によって作成された既存の表の 1 つをドロップする
  - DROP XSROBJECT ステートメントを使用して、既存の表に関連する登録済み XML スキーマ・オブジェクトをドロップする。例えば、DB2 V9.7 のロッキング・イベント・モニター用に EVMON\_FORMAT\_UE\_TO\_TABLES により作成された表に関連した登録済み XML スキーマ・オブジェクトをドロップするには、次のコマンドを使用します: DROP XSROBJECT EVMON\_LOCKING\_SCHEMA\_SQL09070。
  - 現行の登録済み XML スキーマ・オブジェクトのアノテーションの付いたモニター要素に対応する既存の列を変更する。
2. FORCE オプションを使用して、EVMON\_FORMAT\_UE\_TO\_TABLES プロシージャーを実行する。このオプションにより、以前の表はドロップされ、一連の新しい表が作成されます。このオプションを指定しないと、SQL0601N エラーが戻ります。

このプロセスは、451 ページの『例 5: フィックスバック更新に含まれる新規エレメントを取り出す』に例示されています。

上記のステップを実行しない場合、既存の表は以前に登録されたスキーマ・ファイルに基づいて更新されます。フィックスバックに追加されているかもしれない新しい列または表は、EVMON\_FORMAT\_UE\_TO\_TABLES プロシージャーの出力に反映されません。

### リリース・アップグレードの場合

他の指定をしない場合、EVMON\_FORMAT\_UE\_TO\_TABLES プロシージャーを呼び出す際には、現行リリース用の XML スキーマ・ファイルのデフォルト・バージョンが使用されます。それで、DB2 製品の新規リリースにアップグレードする場合は、このプロシージャーを実行する際には、デフォルトで、スキーマ・ファイルの新規バージョンが使用されます。

以前のリリースからの表が存在しない場合、EVMON\_FORMAT\_UE\_TO\_TABLES は最新のスキーマを使用して表を作成します。しかし、以前のリリースからの表が存在する場合、FORCE または RECREATE\_ONERROR オプションを使用して以前の表を新規表に置き換えなければなりません。こうしないと、SQL0601N エラーが戻ります。452 ページの『例 6: リリース更新に含まれる新規エレメントを取り出す』は、新規リリースでデフォルト・スキーマを使用して表を再作成する例を示しています。

または、最新のリリースで導入されるかもしれないどの新しい列または表も追加せずに、既存の表の使用を続けることもできます。既存の表を更新するには、表を作成する際に使用された登録済み XML スキーマ・ファイルの名前を `EVMON_FORMAT_UE_TO_TABLES` プロシージャの `xsobjectname` パラメーターとして指定しなければなりません。452 ページの『例 7: リリース更新でそれまでのリレーショナル表を使用する』は、以前のリリースからのスキーマを使用する例を示しています。

**注:** `EVMON_FORMAT_UE_TO_TABLES` により作成されたりレーショナル表に以前存在していたデータを保持している限り、フィックスパックまたは新規リリースで導入された新しい列または表を取り出すことはできません。どの新しい列を取り出す場合でも、表の再作成が必要です。

### 部分的なイベント

部分的な (不完全な) イベントが UE 表に存在する場合、`EVMON_FORMAT_UE_TO_TABLES` を実行すると、メッセージ (SQL443N) が戻されます。不完全なイベントは、イベント・レコード全体を UE 表に挿入できるようになる前にエージェントが処理を完了すると発生することがあります。この状況は、特にパーティション・データベース環境で、ロッキングが関係する場合に発生する可能性があります。例えば、`LOCKWAIT` しきい値を超えたとき、ロックの保有者に関する詳細が UE 表に書き込まれます。しかし、同じオブジェクトに対するロックを待機しているエージェントに関する詳細は、ロックがタイムアウトになるか、またはウェイターがロックを獲得するまで取り込まれません。ロックを待機しているエージェントがその情報を書き込む前に `EVMON_FORMAT_UE_TO_TABLES` が実行された場合、ロックに関する情報の一部しか UE 表に存在しない可能性があります。

不完全なイベントに関する詳細を参照するには、`EVMON_FORMAT_UE_TO_XML` に `LOG_PARTIAL_EVENTS` オプションを付けて実行します。

### 例

- 450 ページの『例 1: デフォルト・パラメーターを使用する』
- 450 ページの『例 2: 異なるスキーマ下の表の使用を試みる』
- 450 ページの『例 3: 異なるスキーマ下の表の使用を試みる』
- 451 ページの『例 4: `RECREATE_FORCE` オプションを使用する』
- 451 ページの『例 5: フィックスパック更新に含まれる新規エレメントを取り出す』
- 452 ページの『例 6: リリース更新に含まれる新規エレメントを取り出す』
- 452 ページの『例 7: リリース更新でそれまでのリレーショナル表を使用する』
- 452 ページの『例 8: `UPGRADE_TABLES` および `PRUNE_UE_TABLE` オプションを使用する』

## 例 1: デフォルト・パラメーターを使用する

Paul という名前のユーザーが、デフォルト・パラメーターを使用して、このプロシージャを呼び出します。サービス・クラス STUDENTS に含まれるすべてのイベントをリレーショナル表に挿入するためです。

```
EVMON_FORMAT_UE_TO_TABLES (  
  'UOW', NULL, NULL, NULL, NULL, NULL, NULL, -1,  
  'SELECT * FROM UOWUE  
    WHERE service_subclass_name = 'STUDENTS'  
    ORDER BY event_id, event_timestamp')
```

呼び出しの結果は、以下のようになります。

1. プロシージャは、デフォルト XML スキーマ・ファイルである DB2EvmonUOW.xsd ファイルを解析して、作成するリレーショナル表のセットを特定します。
2. SQL スキーマ Paul の下にリレーショナル表が作成されます。
3. XML スキーマが XSR オブジェクト名 PAUL.EVMON\_UOW\_SCHEMA\_SQL09070 で登録されます。
4. XSR オブジェクトが分解可能になります。
5. データが分解されて SQL スキーマ Paul 下の表に挿入されます。

## 例 2: 異なるスキーマ下の表の使用を試みる

前の例に続いて、Dave という名前のユーザーがこのストアード・プロシージャを呼び出します。 *tabschema* パラメーターが Paul に設定されています。

```
EVMON_FORMAT_UE_TO_TABLES (  
  'UOW', NULL, NULL, NULL, 'Paul', NULL, NULL, -1,  
  'SELECT * FROM UOWTBLE  
    ORDER BY event_timestamp')
```

呼び出しの結果は、以下のようになります。

1. プロシージャは、デフォルト XML スキーマ・ファイルである DB2EvmonUOW.xsd ファイルを解析して、作成するリレーショナル表のセットを特定します。
2. プロシージャは、スキーマ Paul の下に表を作成しようとします。しかし、リレーショナル表は現在、SQL スキーマ Paul の下に存在するため、エラーが戻されます。新規 XSR オブジェクトが登録されているときには、既存の表は使用できません。

## 例 3: 異なるスキーマ下の表の使用を試みる

前の例に続いて、Greg という名前のユーザーがこのストアード・プロシージャを呼び出します。入力パラメーター *xrschema* が Paul に設定されています。

```
EVMON_FORMAT_UE_TO_TABLES (  
  'UOW', 'Paul', NULL, NULL, NULL, NULL, NULL, -1,  
  'SELECT * FROM UOWTBL  
    ORDER BY event_timestamp')
```

呼び出しの結果は、以下のようになります。

1. XSR オブジェクト Paul.EVMON\_UOW\_SCHEMA\_SQL09070 が存在し、分解可能になっています。



2. Greg が表に対する INSERT 特権を持っている場合には、データが分解され、SQL スキーマ Paul 下のリレーショナル表に挿入されます。既存の XSR オブジェクト Paul.EVMON\_UOW\_SCHEMA\_SQL09070 が使用されるので、プロシージャへの入力パラメーターとして提供されているものではなく XSR オブジェクトから、リレーショナル表用の SQL スキーマが取得されます。

#### 例 4: RECREATE\_FORCE オプションを使用する

前の例に続いて、Paul が表を再び作成しますが、今度は表スペース MYSPACE に作成します。Paul は RECREATE\_FORCE オプションと *tbsp\_name* パラメーターを指定してプロシージャを呼び出します。

```
EVMON_FORMAT_UE_TO_TABLES (  
  'UOW', NULL, NULL, NULL, NULL, 'MYSPACE', 'RECREATE_FORCE', -1,  
  'SELECT * FROM UOWTBL  
  ORDER BY event_timestamp')
```

呼び出しの結果は、以下のようになります。

1. XSR オブジェクト Paul.EVMON\_UOW\_SCHEMA\_SQL09070 が存在し、分解可能になっています。
2. RECREATE\_FORCE オプションが設定されます。
3. リレーショナル・ファイルのセットを特定するために、XML スキーマ・ファイルがスキーマ・リポジトリから取り出され、解析されます。
4. 現行の表がドロップされ、MYSPACE 表スペースに再び作成されます。
5. データが分解され、新しい表に挿入されます。

#### 例 5: フィックスバック更新に含まれる新規エレメントを取り出す

最新のフィックスバックで、ロック・イベント・モニターの XML スキーマ・ファイルに、「db2EventNew」という新規 XML エレメントが追加されました。Paul は、XML ファイルの分解に使用する新規エレメントを取り出したいと考えています。そのために、Paul は以下の手順に従います。

1. Paul は元のリリースで作成された XSR オブジェクトをドロップします。  

```
DROP XSROBJECT EVMON_LOCKING_SCHEMA_SQL09070
```
2. プロシージャを RECREATE\_ONERROR オプションを指定して呼び出します。

```
EVMON_FORMAT_UE_TO_TABLES (  
  'LOCKING', NULL, NULL, NULL, NULL, 'RECREATE_ONERROR', -1,  
  'SELECT * FROM LOCK  
  ORDER BY event_timestamp')
```

呼び出しの結果は、以下のようになります。

- a. XSR オブジェクトが存在しないので、リレーショナル表のセットを特定するために、デフォルトの DB2EvmonLocking.xsd スキーマ・ファイルが解析されます。
- b. RECREATE\_ONERROR オプションが指定されたので、既存の表がドロップされて再作成されます。

## 例 6: リリース更新に含まれる新規エレメントを取り出す

Paul は新規 DB2 リリースにアップグレードしており、イベント・モニターの XML スキーマ・ファイルに含まれる新規変更を取り出したいと考えています。Paul は RECREATE\_ONERROR オプションを指定してプロシーチャーを呼び出します。

```
EVMON_FORMAT_UE_TO_TABLES (  
  'LOCKING', NULL, NULL, NULL, NULL, NULL, 'RECREATE_ONERROR', -1,  
  'SELECT * FROM LOCK  
  ORDER BY event_timestamp')
```

呼び出しの結果は、以下のようになります。

1. XSR オブジェクトの Paul.EVMON\_LOCKING\_SCHEMA\_SQL1000 が存在しません。
2. RECREATE\_ONERROR オプションが指定されたので、表がドロップされて再作成されます。

## 例 7: リリース更新でそれまでのリレーショナル表を使用する

Greg は新しい DB2 リリースにアップグレードしました。Greg は、前のリリースからの *xsobjectname* 値を指定して、プロシーチャーを呼び出します。

```
EVMON_FORMAT_UE_TO_TABLES (  
  'LOCKING', NULL, 'EVMON_LOCKING_SCHEMA_SQL09070', NULL, NULL, NULL, NULL, -1,  
  'SELECT * FROM LOCK  
  ORDER BY event_timestamp')
```

## 例 8: UPGRADE\_TABLES および PRUNE\_UE\_TABLE オプションを使用する

Paul は、UOWTABLE という UE 表に出力を書き込む作業単位イベント・モニターを V9.7 で作成しました。そこから V10.1 にアップグレードしますが、前のリリースで EVMON\_FORMAT\_UE\_TO\_TABLES によって生成されたリレーショナル表を、新しいデータの処理前に UPGRADE\_TABLES オプションを使ってアップグレードしたいと考えています。さらに、UOWTABLE のレコードが処理された後、PRUNE\_UE\_TABLE オプションを使ってそれらのレコードを削除したいと考えています。

```
EVMON_FORMAT_UE_TO_TABLES (  
  'UOW', NULL, 'EVMON_UOW_SCHEMA_SQL09070', NULL, NULL, NULL,  
  'UPGRADE_TABLES;PRUNE_UE_TABLE', -1,  
  'SELECT * FROM UOWTABLE  
  ORDER BY event_timestamp')
```

注: この例では、*xsobjectname* パラメーターに値

「EVMON\_UOW\_SCHEMA\_SQL09070」を指定する必要があります。

「EVMON\_UOW\_SCHEMA\_SQL09070」は、(UE 表からリレーショナル表を作成するために EVMON\_FORMAT\_UE\_TO\_TABLES を実行したときの) 直前のリリースで使われた XSR オブジェクトの名前だからです。

## 戻される情報

SQLCA を除いて、このプロシーチャーからの出力はありません。SQLCA は完了状況を示します。使用される SQLCODES は次のとおりです。

- 0       すべてのイベントがリレーショナル表に正常に挿入されました。

16278 1 つ以上のイベントがリレーショナル表に挿入されませんでした。SQLCA 内のトークンに、試みられた文書の合計数と分解に成功した文書の合計数が含まれています。

診断ファイルも作成されます。その診断ファイルの名前とロケーションは、DB2 診断パスにある db2diag ログ・ファイルに保管されます。

#### 負の SQLCODE

エラーが発生しました。SQLCODE メッセージを調べることにより、失敗に関する追加の詳細情報が得られます。追加の診断メッセージについては、DB2 診断パスにある db2diag ログ・ファイルを参照してください。

## **EVMON\_FORMAT\_UE\_TO\_XML 表関数 - 不定形式イベントを XML に変換する**

EVMON\_FORMAT\_UE\_TO\_XML 表関数は、バイナリー・イベントを未フォーマット・イベント表から抽出し、XML 文書にフォーマットします。

### 構文

```
▶▶—EVMON_FORMAT_UE_TO_XML—(—options—, —————→  
▶—FOR EACH ROW OF—(—fullselect-statement—)—————→▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### オプション

この表関数でサポートされるキーワード・オプションのリストを表す、タイプ VARCHAR (1024) の入力引数。

#### LOG\_TO\_FILE

XML 文書が 100 MB より大きい場合、表関数が XML 文書をファイルに書き込むことを示します。この表関数によって行ごとに戻される各文書の最大サイズは 100 MB です。ファイルは、  
<xml\_document\_id>.xml ファイルに書き込まれます。ここで、  
<xml\_document\_id> は文書ごとに生成されるユニーク ID です。出力ファイルは、DB2 診断パス・ディレクトリーに書き込まれます。

#### LOG\_PARTIAL\_EVENTS

表関数がすべての部分的な (不完全な) イベントをファイルに書き込むことを示します。部分的なイベントについて詳しくは、このトピックの 455 ページの『使用上の注意』のセクションを参照してください。

NULL オプションが選択されていません。

#### fullselect-statement

全選択ステートメントは、SELECT ステートメントの規則に準拠する照会です。この照会は、以下の規則に従っていなければなりません。

- この照会は、"\*" 節を使用するか、未フォーマット・イベント表のすべての列を指定する必要があります。それ以外の場合、エラーが戻されます。列は、未フォーマット・イベント表の DESCRIBE ステートメントで戻されたのと同じ順序で指定されている必要があります。

- この照会は、フォーマットされていないイベント表からのみ選択を行う必要があります。
- WHERE 節は、フォーマットされていないイベント表の非 LOB 列のいずれかを使用して、イベントをフィルターで除外することができます。
- SELECT ステートメントは、大括弧で囲まれたキーワード FOR EACH ROWS OF で指定されている必要があります。

## 許可

EVMON\_FORMAT\_UE\_TO\_XML 関数に対する EXECUTE 特権。

フォーマットされていないイベント表に対する SELECT 特権。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: 未フォーマット・イベント表「MYLOCKS」からすべてのイベントを照会します。

```
SELECT evmon.* FROM TABLE (
  EVMON_FORMAT_UE_TO_XML (
    NULL,
    FOR EACH ROW OF (
      select * from MYLOCKS
      order by EVENT_TIMESTAMP )))
AS evmon;
```

例 2: 未フォーマット・イベント表「LOCK」から、過去 5 時間に発生したタイプ「LOCKWAIT」のすべてのイベントを照会します。

```
SELECT evmon.* FROM TABLE (
  EVMON_FORMAT_UE_TO_XML (
    NULL,
    FOR EACH ROW OF (
      select * from LOCK order by EVENT_TIMESTAMP
      where EVENT_TYPE = 'LOCKWAIT'
      and EVENT_TIMESTAMP >= CURRENT_TIMESTAMP - 5 hours )))
AS evmon;
```

例 3: 未フォーマット・イベント表「UOW」から、過去 32 時間に発生したワークロード「PAYROLL」に属するすべてのイベントを取得します。任意の文書が 100 MB より大きい場合、結果をファイルに書き込みます。

```
SELECT evmon.* FROM TABLE (
  EVMON_FORMAT_UE_TO_XML(
    'LOG TO FILE',
    FOR EACH ROW OF (
      select * from UOW order by EVENT_TIMESTAMP
      where WORKLOAD_NAME = 'PAYROLL'
      and EVENT_TIMESTAMP = CURRENT_TIMESTAMP - 32 hours )))
AS evmon;
```

例 4: 「UOWEVMON」表からすべての作業単位イベントを照会し、XMLTABLE 表関数を使用して、UOW ID、UOW の開始時刻と停止時刻、および作業単位を発行したユーザーのユーザー ID を提示します。

```

SELECT EVENT.UOW_ID, EVENT.APPLICATION_ID, EVENT.SESSION_AUTHID,
EVENT.START_TIME, EVENT.STOP_TIME
FROM TABLE(
  EVMON_FORMAT_UE_TO_XML(
    'LOG TO FILE',
    FOR EACH ROW OF(
      select * from UOWEVMON )))
AS UEXML,
XMLTABLE(
  XMLNAMESPACES( DEFAULT 'http://www.ibm.com/xmlns/prod/db2/mon' ),
  '$uowevent/db2_uow_event'
  PASSING XMLPARSE( DOCUMENT UEXML.XMLREPORT ) as "uowevent"
  COLUMNS UOW_ID INTEGER PATH 'uow_id',
  MEMBER SMALLINT PATH '@member',
  APPLICATION_ID VARCHAR(128) PATH 'application_id',
  SESSION_AUTHID VARCHAR(128) PATH 'session_authid',
  START_TIME TIMESTAMP PATH 'start_time',
  STOP_TIME TIMESTAMP PATH 'stop_time'
)
AS EVENT

```

## 使用上の注意

### メモリー使用量に対する EVMON\_FORMAT\_UE\_TO\_XML 表関数の影響

UE 表を生成したイベント・モニターのタイプに応じて、EVMON\_FORMAT\_UE\_TO\_XML 表関数は未フォーマット・イベント表の複数のレコードを単一イベントにマップする場合があります。この場合、イベントを構成するすべてのレコードを受け取るまで、レコードはメモリー内にキャッシュされます。表関数に渡されるレコードが、作成されて表に挿入された順序になっていない場合、大量のメモリーが必要になる可能性があります。レコードがそのようにソートされていない場合、表関数は複数のイベントのレコードをキャッシュに入れる必要があります。この問題を回避するには、列 (EVENT\_ID、EVENT\_TIMESTAMP、EVENT\_TYPE、および MEMBER) が含まれる ORDER BY 節で *fullselect-statement* パラメーターを修飾します。表関数は、常に単一イベントからのレコードのみを処理およびキャッシュするため、メモリー消費量は削減されます。

### UE 表のレコードと EVMON\_FORMAT\_UE\_TO\_XML 表関数の出力の関係

UE 表に書き込まれるレコードと EVMON\_FORMAT\_UE\_TO\_XML 表関数の出力の間のマッピングは、1 対 1 ではありません。UE 表に複数のレコードを生成するイベントもありますし、1 つのレコードだけが追加されるものもあります。EVMON\_FORMAT\_UE\_XML 表関数は、単一イベントを記述する UE 表のすべてのレコードを常に 1 つの XML 文書に結合します。

### 部分的なイベント

部分的な (不完全な) イベントが UE 表に存在する場合、LOG\_PARTIAL\_EVENTS オプションの指定の有無に関係なく、EVMON\_FORMAT\_UE\_TO\_XML を実行すると、メッセージ (SQL443N) が戻されます。不完全なイベントは、イベント・レコード全体を UE 表に挿入できるようになる前にエージェントが処理を完了すると発生することがあります。この状況は、特にパーティション・データベース環境で、ロックングが関係する場合に発生する可能性があります。例えば、LOCKWAIT しきい値を超えたとき、ロックの保有者に関する詳細が UE 表に書き込まれます。しかし、同じオブジェクトに対するロックを待機しているエージェントに関する詳細は、ロックがタイムアウトになるか、またはウェイターがロッ

クを獲得するまで取り込まれません。ロックを待機しているエージェントがその情報を書き込む前に `EVMON_FORMAT_UE_TO_XML` が実行された場合、ロックに関する情報の一部しか UE 表に存在しない可能性があります。

`LOG_PARTIAL_EVENTS` オプションを指定すると、UE 表の中の不完全なイベントは別個の XML 文書に書き込まれます。また、不完全なイベントが発生したことを示すメッセージが `db2diag` ログ・ファイルに書き込まれます。そのメッセージには、不完全なイベントに関する詳細を記述する XML 文書のファイル名が明記されます。生成される XML 文書は、`db2evmonfmt` ツールを使用してフォーマット設定できます。

## 戻される情報

表 103. `EVMON_FORMAT_UE_TO_XML` について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
XMLID	VARCHAR(1024)	xmlid - XML ID モニター・エレメント
XMLREPORT	BLOB(100M)	単一の完全イベントを含む XML 文書。各文書の最大サイズは 100 MB です。

## EVMON\_UPGRADE\_TABLES プロシージャ - イベント・モニターのターゲット表のアップグレード

`EVMON_UPGRADE_TABLES` プロシージャは、イベント・モニターの作成後に追加された新規モニター・エレメントまたは変更されたモニター・エレメントに対応するように、イベント・モニターのターゲットの SQL イベント表または未フォーマット・イベント表を変更します。

### 構文

```

▶▶—EVMON_UPGRADE_TABLES—(—evmon_name—, —evmon_type—
▶,—options—, —num_evmons_evaluated—
▶,—num_evmons_to_upgrade—, —num_evmons_upgraded—)▶▶

```

スキーマは `SYSPROC` です。

### プロシージャ・パラメーター

#### `evmon_name`

既存の表定義をアップグレードする対象となるイベント・モニターの名前を指定する、タイプ `VARCHAR (128)` の入力引数。イベント・モニターのターゲットのタイプは、表または未フォーマット・イベント表でなければなりません。`evmon_name` を指定した場合、`evmon_type` 引数は無視されます。

この名前には、`LIKE` 述部に使用するようなパターン式を使用できます。つまり、下線文字 (`_`) またはパーセント (`%`) 文字をワイルドカード文字として含めることができます。`LIKE` 述部について詳しくは、`LIKE` 述部を参照してください。パターン式に下線文字またはパーセント文字を使用する必要がある場合は、エスケープ・オプションを使用してパターン中の下線文字またはパーセント文字の先行文字を指定します。

*evmon\_name* を指定しない場合、または NULL に設定した場合は、これ以外の入力パラメーターに指定した条件に従って、表または未フォーマット・イベント表の出力ターゲットを持つすべてのイベント・モニターが処理されます。

#### *evmon\_type*

既存の表定義をアップグレードする対象となるイベント・モニターのタイプを指定する、タイプ VARCHAR (128) の入力引数。指定するタイプは、SYSCAT.EVENTS の TYPE 列にある値のいずれかでなければなりません。詳しくは、SYSCAT.EVENTS カタログ・ビューを参照してください。このイベント・タイプのイベント・モニターで、ターゲット・タイプが表または未フォーマット・イベント表であるイベント・モニターは、すべてアップグレード対象として評価されます。*evmon\_type* を指定する場合、*evmon\_name* は NULL でなければなりません。*evmon\_type* を指定しない場合、または NULL に設定した場合は、これ以外の入力パラメーターに指定した条件に従って、すべてのタイプのイベント・モニターが評価されます。

#### *options*

イベント・モニターのアップグレード設定を 1 つ以上指定できる、タイプ CLOB(8K) の入力引数。このパラメーターのデフォルトは NULL です。設定は、次の形式の名前と値の対として指定されます。

```
<setting name tag>value</setting name tag>
```

設定は、それぞれ最大で 1 回指定できます。設定名には、大/小文字の区別があります。各設定値は、特に注記がない限り、大/小文字を区別しません。使用可能な設定名タグは次のとおりです。

- '`<tbodyName>value</tbodyName>`'

新規グループを作成する表スペースの名前を指定します。この値には、大/小文字の区別があります。これを指定しない場合、イベント・モニターの現行のターゲット表がすべて同じ表スペースに存在するのであれば、新規表もその同じ表スペースに作成されます (従って、すべての表が同じ表スペースに存在することになります)。そうでない場合は、「CREATE EVENT MONITOR」の「IN」節のところに記載している表スペース選出アルゴリズムが使用されます (CREATE EVENT MONITOR ステートメントを参照)。

- '`<createNewGroups>value</createNewGroups>`'

新規グループを作成するかどうかを指定します。指定できる値は、「yes」または「no」のどちらかです。指定しない場合、デフォルトの「yes」となり、アップグレードするイベント・モニターに新規グループが作成されます。

- '`<verbose>value</verbose>`'

ストアード・プロシージャから結果セットとして診断情報を返すかどうかを指定します。指定できる値は、「yes」または「no」のどちらかです。指定しない場合、デフォルトの「yes」となり、診断情報が結果セット内に返されます。「no」を指定すると、結果セットは返されません。ストアード・プロシージャから結果セットを返すためには、ユーザー TEMPORARY 表スペースが存在しなければならないことに注意してください。結果セットについては、表 1 を参照してください。

- '<force>value</force>'

アップグレードが必要かどうか判断する際にイベント・モニターの検査を強制するかどうかを指定します。指定しない場合、デフォルトの「no」となります。「no」を指定した場合は、SYSCAT.EVENTMONITORS の VERSIONNUMBER 列が検査され、その値が現行バージョンと等しければ、イベント・モニターは最新であると見なされます。「yes」を指定した場合は、SYSCAT.EVENTMONITORS の VERSIONNUMBER 列は無視され、アップグレードが必要かどうか、または新規表を追加すべきかどうかを調べるために、イベント・モニターの各表が再検査されます。force を yes に設定すると有用なのは、新規イベント・グループが追加されたイベント・モニターのアップグレードを、createNewGroups オプションに値 NO を指定して既に一度済ませており、新規イベント・グループ用の新規表を作成するためにそのイベント・モニターをもう一度強制的にアップグレードしたい場合です。

- '<escape>char</escape>'

evmon\_name のパターン式の下線 ( \_ ) 文字とパーセント ( % ) 文字の特殊な意味を変更するために使用する文字を指定します。これにより、evmon\_name を使用して、実際にパーセント文字および下線文字を含んでいる値を突き合わせる事が可能になります。このオプションを指定しない場合、エスケープ文字は使用できません。

#### *num\_evmons\_evaluated*

タイプ INTEGER のこの出力パラメーターには、表をアップグレードする必要があるかどうかストアード・プロシージャが検査したイベント・モニターの数が返されます。

#### *num\_evmons\_to\_upgrade*

タイプ INTEGER のこの出力パラメーターには、実際にアップグレードを必要とするターゲット表を少なくとも 1 つ持つイベント・モニターの総数が返されます。これには、新規または変更されたエレメント、あるいは新規イベント・グループを持つイベント・モニターが含まれます。

#### *num\_evmons\_upgraded*

タイプ INTEGER のこの出力パラメーターには、ターゲット表が正常にアップグレードまたは作成されたイベント・モニターの総数が返されます。

### 許可

この関数を実行するには、DBADM 権限が必要です。

### デフォルトの PUBLIC 特権

なし

### 使用上の注意

- evmon\_name および evmon\_type の両方が NULL の場合、ストアード・プロシージャは SYSCAT.EVENTTABLES カタログ表をスキャンし、TABLE イベント・モニターおよび UE 表イベント・モニターごとにターゲット表をアップグレードする必要があるかどうか検査します。



- SYSCAT.EVENTMONITORS の VERSIONNUMBER 列には、イベント・モニターが作成されたとき、または最後にアップグレードされたときのバージョン、リリース、および修正レベルが入っています。VERSIONNUMBER と現行リリースとの間に列の追加や変更があった場合に、イベント・モニターのターゲット表はアップグレードが必要であると見なされます。つまり、これらのリリース間の変更のみが考慮されます。
  - データベースのアップグレード時に、VERSIONNUMBER 列に値が入っていないければ、イベント・モニターのアップグレード元のリリース番号が設定されます。
  - evmon\_upgrade\_tables がイベント・モニターを処理し、そのイベント・モニターが変更を必要としない場合、VERSIONNUMBER は現行レベルに更新されます。これによって、どのイベント・モニターが処理済みかが分かりやすくなります。
- evmon\_type を指定すると、ストアド・プロシージャは SYSCAT.EVENTTABLES カタログ表をスキャンし、そのイベント・タイプが記録されているイベント・モニターごとに、ターゲット表をアップグレードする必要があるかどうか検査します。1 つのイベント・モニターで複数のイベント・タイプを記録できる場合に、そのうちの 1 つのタイプがアップグレード対象として選択されると、そのタイプに関する表のみが検査され、必要に応じてアップグレードされます。例えば、イベント・モニターが CREATE EVENT MONITOR SAMPLE\_EVMON FOR CONNECTIONS, STATEMENTS WRITE TO TABLE で作成されている場合に、evmon\_type に CONNECTIONS を指定すると、CONTROL、CONNHEADER、および CONN のグループのみが検査され、STMT および SUBSECTION はスキップされます。しかし、それにもかかわらず、このイベント・モニターのバージョン番号は更新されます。後でステートメント・イベント表をアップグレードする必要がある場合は、force オプションを使用する必要があります。
- 表はアップグレードに関して検査される前に、まず、イベント・モニターのアクティブ化の際に使用されるのと同じ規則で妥当性検査されます。この妥当性検査により、各列の名前がモニター・エレメント名と一致すること、各列のデータ・タイプがそのエレメントと互換性があること、およびパーティション・データベース環境では、表の最初の列が PARTITION\_KEY であることが確認されます。
- アップグレードが必要な表を含むイベント・モニターの場合:
  - イベント・モニターがアクティブである場合は、非アクティブ化され (まず、プロシージャはキューに入れられたイベントが排出されるまで待機します)、ターゲット表に対する排他ロックが取得されます。3 秒待機しても表をロックできない場合は、次のイベント・モニターに処理が移ります。
  - イベント・モニターの表をアップグレードしている間、そのイベント・モニターに関する新規イベントは失われます。
  - 列をドロップする、あるいは列を追加または変更して最新の状態にするときには、表が変更されます。 '<createNewGroups>yes</createNewGroups>' オプションを指定すると、新規表が作成されます。
  - 表の列が変更またはドロップされると、すべての従属オブジェクトが無効になります。それらの従属オブジェクトの再検査がいつ行われるかは、AUTO\_REVAL データベース構成パラメーターの設定によって決まります。auto\_reval について詳しくは、資料を参照してください。

- 必要に応じて、表を使用可能にするために表の再編成が実行されます。
- 排他表ロックが解放されます。
- イベント・モニターの表がすべて正常にアップグレードされると、SYSCAT.EVENTMONITORS の VERSIONNUMBER 列は現行のバージョン番号に更新され、COMMIT が発行されます。そうでない場合、この作業単位はロールバックされます。
- ストアード・プロシージャーによってイベント・モニターが非アクティブ化された場合、ストアード・プロシージャーが処理を完了した後に再アクティブ化されます。
- イベント・モニター表の作成時に CREATE EVENT MONITOR ステートメントに INCLUDES 節が指定されていた場合、その表はスキップされます。INCLUDES 節が指定されていたかどうかを調べるには、SYSCAT.EVENTTTABLES の TABOPTIONS 列を確認してください。バージョン 10.1 より前に INCLUDES を使用して作成されたイベント・モニターの場合、この情報は失われ、TABOPTIONS 列にはブランクが入っていることに注意してください。
- イベント・モニター表の作成時に CREATE EVENT MONITOR ステートメントに EXCLUDES 節が指定されていた場合、その表はアップグレードの対象となります。イベント・モニターが作成された後の変更に対応するように表が変更されます。EXCLUDES 節が指定されていたかどうかを調べるには、SYSCAT.EVENTTTABLES の TABOPTIONS 列を確認してください。バージョン 10.1 より前に EXCLUDES を使用して作成されたイベント・モニターの場合、この情報は失われ、TABOPTIONS 列にはブランクが入っていることに注意してください。
- プロシージャーは、イベント・モニターの個々のターゲット表を処理している時に検出したエラーを返しません。
- 結果セットが戻される場合 (456 ページの『プロシージャー・パラメーター』を参照)、以下の情報に注意してください。
  - EVMON\_UPGRADE\_TABLES プロシージャーを実行する前に、ユーザー一時表スペースが存在している必要があります。
  - 結果セットに使用される表の名前は SESSION.EVMON\_UPGRADE\_TABLES\_RESULTSET です。
  - 同じセッションで以前に EVMON\_UPGRADE\_TABLES を実行した結果 SESSION.EVMON\_UPGRADE\_TABLES\_RESULTSET が既に存在する場合、新しい結果が既存の表に追加されます。既存の表が作成された後に、何らかの方法で表定義が変更された場合には、ドロップされ、表の新規バージョンが作成されます。
  - EVMON\_UPGRADE\_TABLES が実行されている接続がドロップされると、SESSION.EVMON\_UPGRADE\_TABLES\_RESULTSET も自動的にドロップされます。この表に使用されるユーザー TEMPORARY 表スペースは、SESSION.EVMON\_UPGRADE\_TABLES\_RESULTSET 表がドロップされるまでドロップできません。

## 例

DB2 バージョン 9.7 でユーザーが以下のイベント・モニターを作成していたとします。

```

create event monitor lock for locking write to unformatted event table
create event monitor act for activities write to table control (in
actspace), activity (in actspace), activitystmt (in actspace), activityvals
(in actspace)
create event monitor stat for statistics write to table
create event monitor conn for connections write to table

```

データベースを現行リリースにアップグレードした後に、すべてのイベント・モニター表をアップグレードするには、以下のコマンドを使用します。

```
call evmon_upgrade_tables(null, null, null, ?, ?, ?)
```

そうではなく、act のアップグレードのみを必要とする場合は、以下のコマンドを使用します。

```
call evmon_upgrade_tables('ACT', null, null, ?, ?)
```

また、アクティビティ・イベント・モニターのみをアップグレードすることを選択した場合は、以下のコマンドを使用します。

```
call evmon_upgrade_tables(null, 'ACTIVITIES', null, ?, ?, ?)
```

## 戻される情報

表 104 では、結果セットを作成することを選択した場合にそこに戻される情報が説明されています。

表 104. *EVMON\_UPGRADE\_TABLES* によって表 *SESSION.EVMON\_UPGRADE\_TABLES\_RESULTSET* に戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
EVMON_NAME	VARCHAR(128)	この診断メッセージが該当するイベント・モニターの名前。
EVMON_TYPE	VARCHAR(128)	この診断メッセージが該当するイベント・モニターのイベント・タイプ。
MESSAGE_TIME	TIMESTAMP	message_time - タイム・スタンプ・コントロール表メッセージ・モニター・エレメント
MESSAGE_TEXT	VARCHAR(1024)	診断メッセージ。

## MON\_BP\_UTILIZATION 管理ビュー - バッファース・プールに関するメトリックの取得

MON\_BP\_UTILIZATION 管理ビューは、現在接続されているデータベース内のすべてのバッファース・プールとすべてのデータベース・パーティションに関するヒット率、平均読み取り/書き込み時間などの主要なモニタリング・メトリックを戻します。

これはバッファース・プールの使用効率の確認に役立つため、パフォーマンスをモニターするうえで重要な情報が提供されるといえます。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- MON\_BP\_UTILIZATION 管理ビューに対する SELECT 特権

- MON\_BP\_UTILIZATION 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 戻される情報

表 105. MON\_BP\_UTILIZATION 管理ビューによって戻される情報

列名	データ・タイプ	説明またはモニター・エレメント
BP_NAME	VARCHAR(128)	bp_name - バッファ・プール名
MEMBER	SMALLINT	member - データベース・メンバー
DATA_PHYSICAL_READS	BIGINT	TEMPORARY、REGULAR、および LARGE 表スペースに対して、表スペース・コンテナ (物理) から読み取られたデータ・ページの数を示します。これは ( <i>pool_data_p_reads</i> + <i>pool_temp_data_p_reads</i> ) として計算されます。ここで <i>pool_data_p_reads</i> と <i>pool_temp_data_p_reads</i> は以下のモニター・エレメントを表しています。 <ul style="list-style-type: none"> <li>• <i>pool_data_p_reads</i> - バッファ・プール・データの物理読み取り</li> <li>• <i>pool_temp_data_p_reads</i> - バッファ・プール一時データの物理読み取り</li> </ul>
DATA_HIT_RATIO_PERCENT	DECIMAL(5,2)	データのヒット率。つまり、データ・ページ要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要のなかった時間のパーセンテージ。DB2 pureScale 環境では、この値は、データベース・マネージャーがローカル・バッファ・プールでデータ・ページを見つけた時間のパーセンテージです。

表 105. MON\_BP\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
INDEX_PHYSICAL_READS	BIGINT	<p>TEMPORARY、REGULAR、および LARGE 表スペースに対して、表スペース・コンテナ (物理) から読み取られた索引ページの数を示します。これは (<i>pool_index_p_reads</i> + <i>pool_temp_index_p_reads</i>) として計算されます。ここで <i>pool_index_p_reads</i> + <i>pool_temp_index_p_reads</i> は以下のモニター・エレメントを表しています。</p> <ul style="list-style-type: none"> <li>• <i>pool_index_p_reads</i> - バッファーク・プール索引の物理読み取り</li> <li>• <i>pool_temp_index_p_reads</i> - バッファーク・プールの一時索引の物理読み取り</li> </ul>
INDEX_HIT_RATIO_PERCENT	DECIMAL(5,2)	<p>索引のヒット率。つまり、索引データ・ページ要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要のなかった時間のパーセンテージ。DB2 pureScale 環境では、この値は、データベース・マネージャーがローカル・バッファーク・プールでデータ・ページを見つけた時間のパーセンテージです。</p>
XDA_PHYSICAL_READS	BIGINT	<p>TEMPORARY、REGULAR、および LARGE 表スペースに対して、表スペース・コンテナ (物理) から読み取られた XML ストレージ・オブジェクト (XDA) 用データ・ページの数を示します。これは (<i>pool_xda_p_reads</i> + <i>pool_temp_xda_p_reads</i>) として計算されます。ここで <i>pool_xda_p_reads</i> と <i>pool_temp_xda_p_reads</i> は以下のモニター・エレメントを表しています。</p> <ul style="list-style-type: none"> <li>• <i>pool_xda_p_reads</i> - バッファーク・プール XDA データの物理読み取り</li> <li>• <i>pool_temp_xda_p_reads</i> - バッファーク・プールの一時 XDA データの物理読み取り</li> </ul>

表 105. MON\_BP\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
XDA_HIT_RATIO_PERCENT	DECIMAL(5,2)	<p>補助ストレージ・オブジェクトのヒット率。つまり、XML ストレージ・オブジェクト (XDA) に関するデータ・ページ要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要のなかった時間のパーセンテージ。</p> <p>DB2 pureScale システムでは、この値は、データベース・マネージャーがローカル・バッファ・プールで XDA のデータ・ページを見つけるために使った時間のパーセンテージです。</p>

表 105. MON\_BP\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
TOTAL_PHYSICAL_READS	BIGINT	<p>TEMPORARY、REGULAR、および LARGE 表スペースに対して、表スペース・コンテナ (物理) から読み取られたデータ・ページ、索引ページ、および XML ストレージ・オブジェクト (XDA) 用データ・ページの数を示します。</p> <p>これは (<i>pool_data_p_reads</i> + <i>pool_temp_data_p_reads</i> + <i>pool_index_p_reads</i> + <i>pool_temp_index_p_reads</i> + <i>pool_xda_p_reads</i> + <i>pool_temp_xda_p_reads</i>) として計算されます。ここで <i>pool_data_p_reads</i>、<i>pool_temp_data_p_reads</i>、<i>pool_index_p_reads</i>、<i>pool_temp_index_p_reads</i>、<i>pool_xda_p_reads</i>、および <i>pool_temp_xda_p_reads</i> は以下のモニター・エレメントを表しています。</p> <ul style="list-style-type: none"> <li>• <i>pool_data_p_reads</i> - バッファークール・データの物理読み取り</li> <li>• <i>pool_temp_data_p_reads</i> - バッファークール一時データの物理読み取り</li> <li>• <i>pool_index_p_reads</i> - バッファークール索引の物理読み取り</li> <li>• <i>pool_temp_index_p_reads</i> - バッファークール一時索引の物理読み取り</li> <li>• <i>pool_xda_p_reads</i> - バッファークール XDA データの物理読み取り</li> <li>• <i>pool_temp_xda_p_reads</i> - バッファークール一時 XDA データの物理読み取り</li> </ul>

表 105. MON\_BP\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
AVG_PHYSICAL_READ_TIME	BIGINT	<p>すべてのタイプの表スペースに関して、表スペース・コンテナ (物理) からページを読み取るために費やされた平均時間 (ミリ秒)。</p> <p>物理的読み取りの合計が 0 より大きい場合、これは <math>pool\_read\_time / (pool\_data\_p\_reads + pool\_temp\_data\_p\_reads + pool\_index\_p\_reads + pool\_temp\_index\_p\_reads + pool\_xda\_p\_reads + pool\_temp\_xda\_p\_reads)</math> として計算されます。ここで <math>pool\_read\_time</math>、<math>pool\_data\_p\_reads</math>、<math>pool\_temp\_data\_p\_reads</math>、<math>pool\_index\_p\_reads</math>、<math>pool\_temp\_index\_p\_reads</math>、<math>pool\_xda\_p\_reads</math>、および <math>pool\_temp\_xda\_p\_reads</math> は以下のモニター・エレメントを表しています。</p> <ul style="list-style-type: none"> <li>• <math>pool\_read\_time</math> - バッファークール物理読み取り時間の合計</li> <li>• <math>pool\_data\_p\_reads</math> - バッファークールデータの物理読み取り</li> <li>• <math>pool\_temp\_data\_p\_reads</math> - バッファークール一時データの物理読み取り</li> <li>• <math>pool\_index\_p\_reads</math> - バッファークール索引の物理読み取り</li> <li>• <math>pool\_temp\_index\_p\_reads</math> - バッファークール一時索引の物理読み取り</li> <li>• <math>pool\_xda\_p\_reads</math> - バッファークール XDA データの物理読み取り</li> <li>• <math>pool\_temp\_xda\_p\_reads</math> - バッファークール一時 XDA データの物理読み取り</li> </ul> <p>物理的読み取りの合計が 0 より大きくない場合は、NULL が戻されます。</p>



表 105. MON\_BP\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
PREFETCH_RATIO_ PERCENT	DECIMAL(5,2)	(プリフェッチを使用して) 非同期的に読み取られたページのパーセンテージ。多数のアプリケーションがプリフェッチなしで同期的にデータを読み取っている場合は、システムが最適に調整されていない可能性があります。
ASYNC_NOT_READ_ PERCENT	DECIMAL(5,2)	<p>ディスクから非同期で読み取られ、照会でアクセスされなかったページのパーセンテージ。ディスクからバッファ・プールに非同期的に読み取られて、照会で一度もアクセスされないようなページが非常に多く存在する場合は、プリフェッチによってパフォーマンスが低下する可能性があります。</p> <p>非同期読み取りの合計が 0 より大きい場合、これは</p> $\frac{\text{unread\_prefetch\_pages}}{(\text{pool\_async\_data\_reads} + \text{pool\_async\_index\_reads} + \text{pool\_async\_xda\_reads})}$ <p>として計算されます。ここで</p> <p><i>unread_prefetch_pages</i>、 <i>pool_async_data_reads</i>、 <i>pool_async_index_reads</i>、および <i>pool_async_xda_reads</i> は以下のモニター・エレメントを表しています。</p> <ul style="list-style-type: none"> <li>• <i>unread_prefetch_pages</i> - 読み取り不能プリフェッチ・ページ</li> <li>• <i>pool_async_data_reads</i> - バッファ・プール非同期データ読み取り</li> <li>• <i>pool_async_index_reads</i> - バッファ・プール非同期索引読み取り</li> <li>• <i>pool_async_xda_reads</i> - バッファ・プール非同期 XDA データ読み取り</li> </ul> <p>非同期読み取りの合計が 0 より大きくない場合は、NULL が戻されます。</p>

表 105. MON\_BP\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
TOTAL_WRITES	BIGINT	<p>データ、索引、または XML ストレージ・オブジェクト (XDA) 用データのページが物理的にディスクに書き込まれた回数。</p> <p>これは (<i>pool_data_writes</i> + <i>pool_index_writes</i> + <i>pool_xda_writes</i>) として計算されます。ここで <i>pool_data_writes</i>、<i>pool_index_writes</i>、および <i>pool_xda_writes</i> は以下のモニター・エレメントを表しています。</p> <ul style="list-style-type: none"> <li>• <i>pool_data_writes</i> - バッファークールへのデータの書き込み</li> <li>• <i>pool_index_writes</i> - バッファークール索引の書き込み</li> <li>• <i>pool_xda_writes</i> - バッファークール XDA データの書き込み</li> </ul>
AVG_WRITE_TIME	BIGINT	<p>バッファークールからディスクにページを物理的に書き込むために費やされた平均時間 (ミリ秒)。</p> <p>書き込み操作の合計が 0 より大きい場合、これは <math>pool\_write\_time / (pool\_data\_writes + pool\_index\_writes + pool\_xda\_writes)</math> として計算されます。ここで <i>pool_write_time</i>、<i>pool_data_writes</i>、<i>pool_index_writes</i>、および <i>pool_xda_writes</i> は以下のモニター・エレメントを表しています。</p> <ul style="list-style-type: none"> <li>• <i>pool_write_time</i> - バッファークール物理書き込み時間の合計</li> <li>• <i>pool_data_writes</i> - バッファークールへのデータの書き込み</li> <li>• <i>pool_index_writes</i> - バッファークール索引の書き込み</li> <li>• <i>pool_xda_writes</i> - バッファークール XDA データの書き込み</li> </ul> <p>書き込み操作の合計が 0 より大きくない場合は、NULL が戻されます。</p>
SYNC_WRITES_PERCENT	DECIMAL(5,2)	<p>同期的な書き込み操作のパーセンテージ。</p>

表 105. MON\_BP\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
GBP_DATA_HIT_RATIO _PERCENT	DECIMAL(5,2)	グループ・バッファ・プール・データのヒット率。グループ・バッファ・プールに既にページが存在しているため、データ・ページ要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要がなかった時間のパーセンテージ。この値は、DB2 pureScale 環境の外では NULL です。
GBP_INDEX_HIT_RATIO _PERCENT	DECIMAL(5,2)	グループ・バッファ・プール索引のヒット率。グループ・バッファ・プールに既にページが存在しているため、索引ページ要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要がなかった時間のパーセンテージ。この値は、DB2 pureScale 環境の外では NULL です。
GBP_XDA_HIT_RATIO _PERCENT	DECIMAL(5,2)	グループ・バッファ・プールの補助ストレージ・オブジェクトのヒット率。グループ・バッファ・プールに既にページが存在しているため、XML ストレージ・オブジェクト (XDA) に関するデータ・ページ要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要がなかった時間のパーセンテージ。この値は、DB2 pureScale 環境の外では NULL です。
AVG_SYNC_READ_TIME	BIGINT	バッファ・プールからの同期読み取りに費やされた平均時間 (ミリ秒単位)。
AVG_ASYNC_READ_TIME	BIGINT	バッファ・プールからの非同期読み取りに費やされた平均時間 (ミリ秒単位)。
AVG_SYNC_WRITE_TIME	BIGINT	バッファ・プールへの同期書き込みに費やされた平均時間 (ミリ秒単位)。
AVG_ASYNC_WRITE_TIME	BIGINT	バッファ・プールへの非同期書き込みに費やされた平均時間 (ミリ秒単位)。

## MON\_CONNECTION\_SUMMARY - すべての接続に関するメトリックの取得

MON\_CONNECTION\_SUMMARY 管理ビューは、現在接続されているデータベース内のすべての接続に関する主要なメトリックを戻します。これは各接続が受け取る処理を示すため、システムの概要をモニターするのに役立ちます。

戻されるメトリックは、データベースの全メンバーにおける、識別された接続によってサブミットされた要求に関するすべてのメトリックの累計を表します。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- MON\_CONNECTION\_SUMMARY 管理ビューに対する SELECT 特権
- MON\_CONNECTION\_SUMMARY 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 戻される情報

表 106. MON\_CONNECTION\_SUMMARY 管理ビューによって戻される情報

列名	データ・タイプ	説明またはモニター・エレメント
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル
APPLICATION_NAME	VARCHAR(128)	appl_name - アプリケーション名
APPLICATION_ID	VARCHAR(128)	appl_id - アプリケーション ID
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - セッション許可 ID
TOTAL_APP_COMMITS	BIGINT	total_app_commits - アプリケーション・コミットの合計回数モニター・エレメント
TOTAL_APP_ROLLBACKS	BIGINT	total_app_rollback - アプリケーション・ロールバックの合計回数モニター・エレメント
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 完了したアクティビティの合計数モニター・エレメント
APP_RQSTS_COMPLETED_TOTAL	BIGINT	app_rqsts_completed_total - 完了したアプリケーション要求の合計数モニター・エレメント

表 106. MON\_CONNECTION\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
AVG_RQST_CPU_TIME	BIGINT	正常に完了したすべての外部要求によって費やされた平均 CPU 時間 (マイクロ秒)。これは、ユーザーとシステムの両方の CPU 時間の合計を表します。比率を計算する数式: TOTAL_CPU_TIME / APP_RQSTS_COMPLETED_TOTAL
ROUTINE_TIME_RQST_PERCENT	DECIMAL(5,2)	データベース・サーバーが要求を処理するのに費やした時間のうち、ユーザー・ルーチンの実行に費やした時間のパーセンテージ。比率を計算する数式: TOTAL_ROUTINE_TIME / TOTAL_RQST_TIME
RQST_WAIT_TIME_PERCENT	DECIMAL(5,2)	要求を処理するのに費やされた時間のうち、DB2 データベース・サーバー内部での待機に費やされた時間のパーセンテージ。比率を計算する数式: TOTAL_WAIT_TIME / TOTAL_RQST_TIME
ACT_WAIT_TIME_PERCENT	DECIMAL(5,2)	アクティビティーの実行に費やされた時間のうち、DB2 データベース・サーバー内部での待機に費やされた時間のパーセンテージ。比率を計算する数式: TOTAL_ACT_WAIT_TIME / TOTAL_ACT_TIME
IO_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、入出力操作に費やされた時間のパーセンテージ。これには、直接読み取り/直接書き込みの実行に費やされた時間に加えて、表スペースからバッファ・プールへのデータおよび索引ページの読み取りや元のディスクへの書き込みに費やされた時間が含まれます。比率を計算する数式: (POOL_READ_TIME + POOL_WRITE_TIME + DIRECT_READ_TIME + DIRECT_WRITE_TIME) / TOTAL_WAIT_TIME
LOCK_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、ロック待機に費やされた時間のパーセンテージ。比率を計算する数式: LOCK_WAIT_TIME / TOTAL_WAIT_TIME

表 106. MON\_CONNECTION\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
AGENT_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、コンソントレーター構成のもとでエージェントを待機するためにキューに入れられたアプリケーションによって費やされた時間のパーセンテージ。比率を計算する数式: AGENT_WAIT_TIME / TOTAL_WAIT_TIME
NETWORK_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、クライアント/サーバー通信のために費やされた時間のパーセンテージ。これには、TCP/IP または IPC プロトコルを使ってデータを送受信するのに費やされた時間が含まれます。比率を計算する数式: (TCPIP_SEND_WAIT_TIME + TCPIP_RECV_WAIT_TIME + IPC_SEND_WAIT_TIME + IPC_RECV_WAIT_TIME) / TOTAL_WAIT_TIME
SECTION_PROC_TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、セクションの実行に費やした時間のパーセンテージ。これには、ソートの実行に費やされた時間が含まれます。比率を計算する数式: TOTAL_SECTION_PROC_TIME / (TOTAL_RQST_TIME - TOTAL_WAIT_TIME)
SECTION_SORT_PROC_TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、セクションの実行中にソートの実行に費やした時間のパーセンテージ。比率を計算する数式: TOTAL_SECTION_SORT_PROC_TIME / (TOTAL_RQST_TIME - TOTAL_WAIT_TIME)
COMPILE_PROC_TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、SQL ステートメントのコンパイルに費やした時間のパーセンテージ。これには、明示的および暗黙的コンパイル時間が含まれます。比率を計算する数式: (TOTAL_COMPILE_PROC_TIME + TOTAL_IMPLICIT_COMPILE_PROC_TIME) / (TOTAL_RQST_TIME - TOTAL_WAIT_TIME)

表 106. MON\_CONNECTION\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
TRANSACT_END_PROC_TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、コミット処理の実行またはトランザクションのロールバックに費やした時間のパーセンテージ。比率を計算する数式： $\frac{(\text{TOTAL\_COMMIT\_PROC\_TIME} + \text{TOTAL\_ROLLBACK\_PROC\_TIME})}{(\text{TOTAL\_RQST\_TIME} - \text{TOTAL\_WAIT\_TIME})}$
UTILS_PROC_TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、ユーティリティの実行に費やした時間のパーセンテージ。これには、 <b>runstats</b> 、再編成、およびロード操作の実行が含まれます。比率を計算する数式： $\frac{(\text{TOTAL\_RUNSTATS\_PROC\_TIME} + \text{TOTAL\_REORG\_PROC\_TIME} + \text{TOTAL\_LOAD\_PROC\_TIME})}{(\text{TOTAL\_RQST\_TIME} - \text{TOTAL\_WAIT\_TIME})}$
AVG_LOCK_WAITS_PER_ACT	BIGINT	各コーディネーター・アクティビティ（成功およびアボート）に対して、アプリケーションまたは接続がロックを待機した平均回数。比率を計算する数式： $\frac{\text{LOCK\_WAITS}}{(\text{ACT\_COMPLETED\_TOTAL} + \text{ACT\_ABORTED\_TOTAL})}$
AVG_LOCK_TIMEOUTS_PER_ACT	BIGINT	各コーディネーター・アクティビティ（成功およびアボート）に対して、オブジェクトのロック要求がタイムアウトになった平均回数。比率を計算する数式： $\frac{\text{LOCK\_TIMEOUTS}}{(\text{ACT\_COMPLETED\_TOTAL} + \text{ACT\_ABORTED\_TOTAL})}$
AVG_DEADLOCKS_PER_ACT	BIGINT	コーディネーター・アクティビティ（成功およびアボート）ごとのデッドロックの平均数。比率を計算する数式： $\frac{\text{DEADLOCKS}}{(\text{ACT\_COMPLETED\_TOTAL} + \text{ACT\_ABORTED\_TOTAL})}$
AVG_LOCK_ESCALATIONS_PER_ACT	BIGINT	各コーディネーター・アクティビティ（成功およびアボート）に対して、いくつかの行ロックから表ロックにロックがエスカレートした平均回数。比率を計算する数式： $\frac{\text{LOCK\_ESCALATIONS}}{(\text{ACT\_COMPLETED\_TOTAL} + \text{ACT\_ABORTED\_TOTAL})}$

表 106. MON\_CONNECTION\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
ROWS_READ_PER_ ROWS_RETURNED	BIGINT	アプリケーションに戻された各行に対して、表から読み取られた平均行数。比率を計算する数式: ROWS_READ / ROWS_RETURNED
TOTAL_BP_HIT_ RATIO_PERCENT	DECIMAL(5,2)	XML ストレージ・オブジェクト (XDA) の要求を含めて、データまたは索引ページの要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要のなかった時間のパーセンテージ。DB2 pureScale 環境では、この値はローカル・バッファ・プールの合計ヒット率を表します。比率を計算する数式: (POOL_DATA_LBP_PAGES_FOUND + POOL_INDEX_LBP_PAGES_FOUND + POOL_XDA_LBP_PAGES_FOUND) / (POOL_DATA_L_READS + POOL_TEMP_DATA_L_READS + POOL_INDEX_L_READS + POOL_TEMP_INDEX_L_READS + POOL_XDA_L_READS + POOL_TEMP_XDA_L_READS)
TOTAL_GBP_HIT_ RATIO_PERCENT	DECIMAL(5,2)	DB2 pureScale システムでは、ページがグループ・バッファ・プールに置かれていたため、データ・ページ要求、索引ページ要求、または XML ストレージ・オブジェクト (XDA) ページ要求を処理するためにデータベース・マネージャーがディスクからローカル・バッファ・プールにページをロードする必要がなかった時間のパーセンテージ。この値は、DB2 pureScale 以外のシステムでは必ず 0 になります。
CF_WAIT_TIME_ PERCENT	DECIMAL(5,2)	DB2 pureScale システムでは、キャッシング・ファシリティー通信の待機に費やされた合計待機時間のパーセンテージ。この値は、DB2 pureScale 以外のシステムでは必ず 0 になります。
RECLAIM_WAIT_TIME_ PERCENT	DECIMAL(5,2)	DB2 pureScale システムでは、ページ再利用の待機に費やされた合計待機時間のパーセンテージ。この値は、DB2 pureScale 以外のシステムでは必ず 0 になります。



表 106. MON\_CONNECTION\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
SPACEMAPPAGE_RECLAIM_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 pureScale システムでは、スペース・マップ・ページ再利用の待機に費やされた合計待機時間のパーセンテージ。この値は、DB2 pureScale 以外のシステムでは必ず 0 になります。

## MON\_CURRENT\_SQL - 全メンバーの全アクティビティに関する主要なメトリックの取得

MON\_CURRENT\_SQL 管理ビューは、データベースの全メンバーに対してサブミットされた、まだ未完了であるすべてのアクティビティに関する主要なメトリックを戻します。これには、現在接続されているデータベースで現在実行中の SQL ステートメント (静的および動的) のポイント・イン・タイム・ビューが含まれます。

MON\_CURRENT\_SQL 管理ビューを使用すると、長く実行されているアクティビティを識別し、パフォーマンス上の問題を防ぐことができます。

このビューは、すべてのメンバーで集約されたメトリックを戻します。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- MON\_CURRENT\_SQL 管理ビューに対する SELECT 特権
- MON\_CURRENT\_SQL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 戻される情報

表 107. MON\_CURRENT\_SQL 管理ビューによって戻される情報

列名	データ・タイプ	説明またはモニター・エレメント
COORD_MEMBER	SMALLINT	coord_member - コーディネーター・メンバー
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル
APPLICATION_NAME	VARCHAR(128)	appl_name - アプリケーション名
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - セッション許可 ID

表 107. MON\_CURRENT\_SQL 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
CLIENT_APPLNAME	VARCHAR(255)	CURRENT CLIENT_APPLNAME 特殊レジスター
ELAPSED_TIME_SEC	INTEGER	このアクティビティーが開始してから経過した時間 (秒)。アクティビティーがシステムに入ったが、キューに入っているためまだ実行開始済みでない場合は、この列の値が NULL です。
ACTIVITY_STATE	VARCHAR(32)	activity_state - アクティビティー状態
ACTIVITY_TYPE	VARCHAR(32)	activity_type - アクティビティー・タイプ
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_RETURNED	BIGINT	rows_returned - 戻り行数
QUERY_COST_ESTIMATE	BIGINT	query_cost_estimate - 照会コストの見積もり
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
STMT_TEXT	CLOB(2MB)	stmt_text - SQL 動的ステートメント・テキスト

## MON\_CURRENT\_UOW - すべての作業単位に関するメトリックの取得

MON\_CURRENT\_UOW 管理ビューは、データベースの全メンバーに対してサブミットされたすべての作業単位に関する主要なメトリックを戻します。長く実行されている作業単位が識別されるため、これを使用してパフォーマンス上の問題を防ぐことができます。

MON\_CURRENT\_UOW ビューは、個々のメンバーではなく、コーディネーターの観点を表します。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- MON\_CURRENT\_UOW 管理ビューに対する SELECT 特権
- MON\_CURRENT\_UOW 管理ビューに対する CONTROL 特権

- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

以下の例では、1 分を超えて実行されたすべての作業単位に関するアプリケーション・ハンドル、作業単位 ID、経過時間、読み取られた行と戻された行の総数が取得されます。

```
SELECT APPLICATION_HANDLE AS APPL_HANDLE,
       UOW_ID, ELAPSED_TIME_SEC,
       TOTAL_ROWS_MODIFIED AS TOTAL_READ,
       TOTAL_ROWS_MODIFIED AS TOTAL_MODIFIED
FROM MON_CURRENT_UOW
WHERE ELAPSED_TIME_SEC > 60
ORDER BY ELAPSED_TIME_SEC DESC
```

以下はこの照会の出力例です。

```
APPL_HANDLE UOW_ID ELAPSED_TIME_SEC TOTAL_READ TOTAL_MODIFIED
-----
          254      1           750      87460           0
           61      1           194       108           0
          145      4            82         0           34
```

3 record(s) selected.

## 戻される情報

表 108. MON\_CURRENT\_UOW 管理ビューによって戻される情報

列名	データ・タイプ	説明またはモニター・エレメント
COORD_MEMBER	SMALLINT	coord_member - コーディネーター・メンバー
UOW_ID	INTEGER	uow_id - 作業単位 ID
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル
APPLICATION_NAME	VARCHAR(128)	appl_name - アプリケーション名
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - セッション許可 ID
CLIENT_APPLNAME	VARCHAR(255)	CURRENT CLIENT_APPLNAME 特殊レジスター
ELAPSED_TIME_SEC	INTEGER	この作業単位が開始してから経過した時間 (秒)。
WORKLOAD_OCCURRENCE_STATE	VARCHAR(32)	workload_occurrence_state - ワークロード・オカレンスの状態

表 108. MON\_CURRENT\_UOW 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間
TOTAL_ROWS_MODIFIED	BIGINT	挿入、更新、または削除された行の総数。
TOTAL_ROWS_READ	BIGINT	表から読み取られた行の総数。
TOTAL_ROWS_RETURNED	BIGINT	選択されてアプリケーションに戻された行の総数。

## MON\_DB\_SUMMARY - データベースの全メンバーにわたる累計メトリックの取得

MON\_DB\_SUMMARY 管理ビューは、現在接続されているデータベース内のすべてのサービス・クラスにわたって集計された主要なメトリックを戻します。データベースの簡潔な要約が示されるため、システムの概要をモニターするのに役立ちます。

戻されるメトリックは、データベースの全メンバーにわたるメトリックの累計を表します。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- MON\_DB\_SUMMARY 管理ビューに対する SELECT 特権
- MON\_DB\_SUMMARY 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 戻される情報

表 109. MON\_DB\_SUMMARY 管理ビューによって戻される情報

列名	データ・タイプ	説明またはモニター・エレメント
TOTAL_APP_COMMITS	BIGINT	total_app_commits - アプリケーション・コミットの合計回数モニター・エレメント
TOTAL_APP_ROLLBACKS	BIGINT	total_app_rollback - アプリケーション・ロールバックの合計回数モニター・エレメント

表 109. MON\_DB\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 完了したアクティビティーの合計数モニター・エレメント
APP_RQSTS_COMPLETED_TOTAL	BIGINT	app_rqsts_completed_total - 完了したアプリケーション要求の合計数モニター・エレメント
AVG_RQST_CPU_TIME	BIGINT	正常に完了したすべての外部要求によって費やされた平均 CPU 時間 (マイクロ秒)。これは、ユーザーとシステムの両方の CPU 時間の合計を表します。
ROUTINE_TIME_RQST_PERCENT	DECIMAL(5,2)	データベース・サーバーが要求を処理するのに費やした時間のうち、ユーザー・ルーチンの実行に費やした時間のパーセンテージ。
RQST_WAIT_TIME_PERCENT	DECIMAL(5,2)	要求を処理するのに費やされた時間のうち、DB2 データベース・サーバー内部での待機に費やされた時間のパーセンテージ。
ACT_WAIT_TIME_PERCENT	DECIMAL(5,2)	アクティビティーの実行に費やされた時間のうち、DB2 データベース・サーバー内部での待機に費やされた時間のパーセンテージ。
IO_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、入出力操作に費やされた時間のパーセンテージ。これには、直接読み取り/直接書き込みの実行に費やされた時間に加えて、表スペースからバッファー・プールへのデータおよび索引ページの読み取りや元のディスクへの書き込みに費やされた時間が含まれます。
LOCK_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、ロック待機に費やされた時間のパーセンテージ。
AGENT_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、コンセントレーター構成のもとでエージェントを待機するためにキューに入れられたアプリケーションによって費やされた時間のパーセンテージ。

表 109. MON\_DB\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
NETWORK_WAIT_ TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、クライアント/サーバー通信のために費やされた時間のパーセンテージ。これには、TCP/IP または IPC プロトコルを使ってデータを送受信するのに費やされた時間が含まれます。
SECTION_PROC_ TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、セクションの実行に費やした時間のパーセンテージ。これには、ソートの実行に費やされた時間が含まれます。
SECTION_SORT_ PROC_TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、セクションの実行中にソートの実行に費やした時間のパーセンテージ。
COMPILE_PROC_ TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、SQL ステートメントのコンパイルに費やした時間のパーセンテージ。これには、明示的および暗黙的コンパイル時間が含まれます。
TRANSACT_END_PROC_ TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、コミット処理の実行またはトランザクションのロールバックに費やした時間のパーセンテージ。
UTILS_PROC_ TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、ユーティリティの実行に費やした時間のパーセンテージ。これには、 <b>runstats</b> 、再編成、およびロード操作の実行が含まれます。
AVG_LOCK_WAITS_ _PER_ACT	BIGINT	各コーディネーター・アクティビティ (成功およびアボート) に対して、アプリケーションまたは接続がロックを待機した平均回数。
AVG_LOCK_TIMEOUTS_ _PER_ACT	BIGINT	各コーディネーター・アクティビティ (成功およびアボート) に対して、オブジェクトのロック要求がタイムアウトになった平均回数。

表 109. MON\_DB\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
AVG_DEADLOCKS_ PER_ACT	BIGINT	コーディネーター・アクティビティ (成功およびアボート) ごとのデッドロックの平均数。
AVG_LOCK_ESCALS_ _PER_ACT	BIGINT	各コーディネーター・アクティビティ (成功およびアボート) に対して、いくつかの行ロックから表ロックにロックがエスカレートした平均回数。
ROWS_READ_PER_ ROWS_RETURNED	BIGINT	アプリケーションに戻された各行に対して、表から読み取られた平均行数。
TOTAL_BP_HIT_ RATIO_PERCENT	DECIMAL(5,2)	XML ストレージ・オブジェクト (XDA) の要求を含めて、データまたは索引ページの要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要のなかった時間のパーセンテージ。DB2 pureScale 環境では、この値はローカル・バッファ・プールの合計ヒット率を表します。
TOTAL_GBP_HIT_ RATIO_PERCENT	DECIMAL(5,2)	DB2 pureScale 環境では、ページがグループ・バッファ・プールに置かれていたため、データ・ページ要求、索引ページ要求、または XML ストレージ・オブジェクト (XDA) ページ要求を処理するためにデータベース・マネージャーがディスクからローカル・バッファ・プールにページをロードする必要がなかった時間のパーセンテージ。この値は、DB2 pureScale 環境の外では常に NULL です。
CF_WAIT_TIME_ PERCENT	DECIMAL(5,2)	DB2 pureScale 環境では、キャッシング・ファシリティ通信の待機に費やされた合計待機時間のパーセンテージ。この値は、DB2 pureScale 環境の外では常に NULL です。
RECLAIM_WAIT_TIME_ PERCENT	DECIMAL(5,2)	DB2 pureScale 環境では、ページ再利用の待機に費やされた合計待機時間のパーセンテージ。この値は、DB2 pureScale 環境の外では常に NULL です。

表 109. MON\_DB\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
SPACEMAPPAGE_RECLAIM_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 pureScale 環境では、スペース・マップ・ページ再利用の待機に費やされた合計待機時間のパーセンテージ。この値は、DB2 pureScale 環境の外では常に NULL です。

## MON\_FORMAT\_LOCK\_NAME - 内部ロック名のフォーマット設定と詳細の出力

MON\_FORMAT\_LOCK\_NAME 表関数は、内部ロック名をフォーマット設定し、ロックについての詳細情報を行ベースの形式で戻します。戻される各行は、その特定のロックに関する *key-value* の組で構成されます。

ロックについての情報を取得するには MON\_FORMAT\_LOCK\_NAME、MON\_GET\_LOCKS、および MON\_GET\_APPL\_LOCKWAIT 表関数を使用してください。SNAPLOCKWAIT 管理ビューと SNAP\_GET\_LOCKWAIT 表関数、および SNAPLOCK 管理ビューと SNAP\_GET\_LOCK 表関数はバージョン 9.7 フィックスパック 1 で非推奨になりました。

▶—MON\_FORMAT\_LOCK\_NAME—(—lockname—)————▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### *lockname*

フォーマット設定されるロックの内部バイナリー名を指定する、タイプ VARCHAR (32) の入力引数。NULL 値を指定した場合、結果としてエラー SQL0171N が戻されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

内部ロック名は、**db2diag** ログ・ファイルへの書き込み、**lock\_name** モニター・エレメントの値などのさまざまな状況で戻されます。以下の例は、



MON\_FORMAT\_LOCK\_NAME 表関数を使ってロックについての詳細情報を検出する方法を示しています。この場合はロック名 0000000E00000000000B00C152 を使用します。

```
SELECT SUBSTR(NAME,1,20) AS NAME,
       SUBSTR(VALUE,1,50) AS VALUE
FROM
TABLE( MON_FORMAT_LOCK_NAME('0000000E00000000000B00C152')) as LOCK
```

以下の出力が戻されます。

```
NAME                VALUE
-----
LOCK_OBJECT_TYPE    ROW
ROWID                0
DATA_PARTITION_ID   49408
PAGEID              184549376
TBSP_NAME            SYSCATSPACE
```

5 record(s) selected.

## 戻される情報

表 110. MON\_FORMAT\_LOCK\_NAME 表関数によって戻される情報

列名	データ・タイプ	説明
NAME	VARCHAR(256)	ロック名のエレメント。詳しくは、下記の表を参照してください。
VALUE	VARCHAR(1024)	エレメントの値。

指定されたロック名を構成するすべてのエレメントが戻されるわけではありません。関連のある *key-value* の組だけが戻されます。

戻される可能性のあるエレメントは次のとおりです。

表 111. 戻される可能性のあるモニター・エレメント

エレメント名	説明	可能な値またはモニター・エレメント
LOCK_OBJECT_TYPE	lock_object_type - 待機中のロック対象タイプ・モニター・エレメント	lock_object_type - ロック対象タイプ  可能な値については、『lock_object_type - 待機中のロック対象タイプ』モニター・エレメントを参照してください。
DATA_PARTITION_ID	data_partition_id - データ・パーティション ID モニター・エレメント	data_partition_id - データ・パーティション ID
TBSP_NAME	表スペースの名前	tablespace_name - 表スペース名
TABSCHEMA	table_schema - 表スキーマ名 モニター・エレメント	table_schema - 表スキーマ名

表 111. 戻される可能性のあるモニター・エレメント (続き)

エレメント名	説明	可能な値またはモニター・エレメント
<b>TABNAME</b>	table_name - 表名モニター・エレメント	table_name - 表名
<b>ROWID</b>	表の行 ID	-
<b>PAGEID</b>	ページ ID	-
<b>WORKLOAD_NAME</b>	workload_name - ワークロード名 : モニター・エレメント	workload_name - ワークロード名
<b>STORAGE_GRP_ID</b>	ストレージ・グループ ID	-
<b>BUFFERPOOL_NAME</b>	バッファ・プールの名前	-
<b>FED_SERVER_NAME</b>	フェデレーション・サーバーの名前	-
<b>FED_USER_NAME</b>	フェデレーション・ユーザー・マッピングの名前	-
<b>SEQ_OPERATION</b>	シーケンス・ロックを要求している操作	可能な値は以下のとおりです。 <ul style="list-style-type: none"> <li>• AUTONOMIC_POLICIES</li> <li>• CATALOG_ARRAY</li> <li>• DESCRIBE</li> <li>• INIT_EVMON</li> <li>• INIT_PACKAGE</li> <li>• INIT_AUDIT</li> <li>• PACKAGE_CREATION</li> <li>• INIT_ROUTINE_ID</li> <li>• INIT_ROLE_ID</li> <li>• TEMP_TBSPACE</li> <li>• AUDIT_DDL</li> <li>• VERSION_TIMES</li> <li>• WLM</li> <li>• TRUSTED_CTX</li> <li>• INIT_TRUSTED_CTX</li> <li>• STATIC_STMT</li> <li>• USER_TEMP_TBSPACE</li> </ul>
<b>CONTAINER_ID</b>	container_id - コンテナ ID モニター・エレメント	-
<b>STMT_UID</b>	ステートメント ID	-
<b>PACKAGE_TOKEN</b>	パッケージ・トークン	-
<b>INTERNAL</b>	内部使用のために予約済み	-

## MON\_FORMAT\_XML\_COMPONENT

### \_TIMES\_BY\_ROW - フォーマット設定された行ベースのコンポーネント時間の取得

MON\_FORMAT\_XML\_COMPONENT\_TIMES\_BY\_ROW 表関数は、XML メトリック文書に含まれるコンポーネント時間に関するフォーマット設定された行ベースの出力を戻します。

#### 構文

▶▶MON\_FORMAT\_XML\_COMPONENT\_TIMES\_BY\_ROW(—*xml doc*—)▶▶

スキーマは SYSPROC です。

#### 表関数パラメーター

*xml doc*

system\_metrics または activity\_metrics エレメントを持つ XML 文書が含まれる、タイプ BLOB(100M) の入力引数。これらのエレメントを持つ XML 文書は、以下のソースから入手可能です。

- いずれかの MON\_GET\_\*\_DETAILS 表関数からの戻り。
- 統計およびアクティビティ・イベント・モニターからのメトリック列出力。
- 作業単位またはパッケージ・キャッシュ・イベント・モニターからのフォーマット設定された出力。

#### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

#### デフォルトの PUBLIC 特権

なし

#### 例

以下の例は、サービス・サブクラスに関する DB2 データベース・マネージャー内のコンポーネント時間の明細を戻します。特定のコンポーネントで費やされた合計時間、およびコンポーネント内で (待機ではなく) 実際の処理に費やされた時間の合計が示されます。

```
SELECT SUBSTR(T.SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS,  
       SUBSTR(T.SERVICE_SUBCLASS_NAME,1,19) AS SUBCLASS,  
       T.MEMBER,  
       SUBSTR(COMP.METRIC_NAME,1,15) AS METRIC_NAME  
       SUBSTR(COMP.PARENT_METRIC_NAME,1,15) AS PARENT_NAME  
       COMP.TOTAL_TIME_VALUE AS TOTAL_TIME,  
       COMP.PROC_TIME_VALUE AS TOTAL_PROC_TIME,
```

```

COMP.COUNT
FROM TABLE (MON_GET_SERVICE_SUBCLASS_DETAILS(NULL,
NULL,-2)) AS T,
TABLE(MON_FORMAT_XML_COMPONENT_TIMES_BY_ROW(
T.DETAILS
)) AS COMP

WHERE COMP.PARENT_METRIC_NAME IS NOT NULL;

```

以下はこの照会の出力例です。

```

SUPERCLASS      SUBCLASS      MEMBER METRIC_NAME      PARENT_NAME      ...
-----
MYSC             MYSSC             0 TOTAL_COMPILE_T TOTAL_RQST_TIME...
MYSC             MYSSC             0 TOTAL_IMPLICIT_ TOTAL_RQST_TIME...
MYSC             MYSSC             0 TOTAL_SECTION_T TOTAL_RQST_TIME...
MYSC             MYSSC             0 TOTAL_COMMIT_TI TOTAL_RQST_TIME...
MYSC             MYSSC             0 TOTAL_ROLLBACK_ TOTAL_RQST_TIME...
MYSC             MYSSC             0 TOTAL_RUNSTATS_ TOTAL_RQST_TIME...
MYSC             MYSSC             0 TOTAL_REORG_TIM TOTAL_RQST_TIME...
MYSC             MYSSC             0 TOTAL_LOAD_TIME TOTAL_RQST_TIME...
MYSC             MYSSC             0 TOTAL_SECTION_S TOTAL_SECTION_T...

```

9 record(s) selected.

以下は、この照会の出力例の続きです。

```

...TOTAL_TIME      TOTAL_PROC_TIME      COUNT
-----
...             100             100             1
...             0             0             0
...             1253            953             0
...             213            153             0
...             0             0             0
...             0             0             0
...             0             0             0
...             0             0             0
...             0             0             0

```

9 record(s) selected.

## 戻される情報

表 112. MON\_FORMAT\_XML\_COMPONENT\_TIMES\_BY\_ROW で戻される情報

列名	データ・タイプ	説明
METRIC_NAME	VARCHAR(128)	合計時間メトリック値のユニーク ID。
PROC_METRIC_NAME	VARCHAR(128)	処理時間メトリックのユニーク ID。
TOTAL_TIME_VALUE	BIGINT	metric_name に対応する合計時間値 (ミリ秒)。
PROC_TIME_VALUE	BIGINT	proc_metric_name に対応する処理時間値 (ミリ秒)
COUNT	BIGINT	count - イベント・モニター・オーバーフロー数モニター・エレメント
PARENT_METRIC_NAME	VARCHAR(128)	親である合計時間メトリックの ID。そのメトリックの値には total_time_value がサブセットとして含まれます。
PARENT_PROC_METRIC_NAME	VARCHAR(128)	親である処理時間メトリックの ID。そのメトリックの値には proc_time_value がサブセットとして含まれます。

タイプ *system\_metrics* のエレメントを含む XML 文書は、以下のインターフェースから生成されます。

- MON\_GET\_CONNECTION\_DETAILS
- MON\_GET\_SERVICE\_SUBCLASS\_DETAILS
- MON\_GET\_UNIT\_OF\_WORK\_DETAILS
- MON\_GET\_WORKLOAD\_DETAILS
- STATISTICS イベント・モニターからの DETAILS\_XML 列
- UNIT OF WORK イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_TABLES によって生成される METRICS 列
- UNIT OF WORK イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_XML の XMLREPORT 列

この場合に XML 文書から戻されるメトリックの種類とその親メトリックについては、表 113 を参照してください。

表 113. *system\_metrics* エレメント・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_COMPONENT\_TIMES\_BY\_ROW* によって戻されるメトリック名

メトリック名	処理メトリック名	親メトリック名	親処理メトリック名	メトリックについての説明またはモニター・エレメント
TOTAL_RQST_TIME	NULL	NULL	NULL	total_rqst_time - 合計要求時間
TOTAL_COMPILE_TIME	TOTAL_COMPILE_PROC_TIME	TOTAL_RQST_TIME	TOTAL_RQST_TIME	total_compile_time - コンパイル時間の合計
TOTAL_IMPLICIT_COMPILE_TIME	TOTAL_IMPLICIT_COMPILE_PROC_TIME	TOTAL_RQST_TIME	TOTAL_RQST_TIME	total_implicit_compile_time - 暗黙的コンパイル時間の合計
TOTAL_SECTION_TIME	TOTAL_SECTION_PROC_TIME	TOTAL_RQST_TIME	TOTAL_RQST_TIME	total_section_time - セクション時間の合計
TOTAL_COMMIT_TIME	TOTAL_COMMIT_PROC_TIME	TOTAL_RQST_TIME	TOTAL_RQST_TIME	total_commit_time - コミット時間の合計
TOTAL_ROLLBACK_TIME	TOTAL_ROLLBACK_PROC_TIME	TOTAL_RQST_TIME	TOTAL_RQST_TIME	total_rollback_time - ロールバック時間の合計
TOTAL_ROUTINE_USER_CODE_TIME	TOTAL_ROUTINE_USER_CODE_PROC_TIME	TOTAL_RQST_TIME	TOTAL_RQST_TIME	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
TOTAL_RUNSTATS_TIME	TOTAL_RUNSTATS_PROC_TIME	TOTAL_RQST_TIME	TOTAL_RQST_TIME	total_runstats_time - ランタイム統計時間の合計
TOTAL_REORG_TIME	TOTAL_REORG_PROC_TIME	TOTAL_RQST_TIME	TOTAL_RQST_TIME	total_reorg_time - 再編成時間の合計
TOTAL_LOAD_TIME	TOTAL_LOAD_PROC_TIME	TOTAL_RQST_TIME	TOTAL_RQST_TIME	total_load_time - ロード時間の合計
TOTAL_SECTION_SORT_TIME	TOTAL_SECTION_SORT_PROC_TIME	TOTAL_SECTION_TIME	TOTAL_SECTION_PROC_TIME	total_section_sort_time - セクション・ソート時間合計
TOTAL_STATS_FABRICATION_TIME	TOTAL_STATS_FABRICATION_PROC_TIME	TOTAL_COMPILE_TIME	TOTAL_COMPILE_PROC_TIME	total_stats_fabrication_time - 統計作成の合計時間

表 113. *system\_metrics* エlement・タイプを含む XML 文書に対する  
*MON\_FORMAT\_XML\_COMPONENT\_TIMES\_BY\_ROW* によって戻されるメトリック名 (続き)

メトリック名	処理メトリック名	親メトリック名	親処理メトリック名	メトリックについての説明またはモニター・Element
TOTAL_SYNC_RUNSTATS_TIME	TOTAL_SYNC_RUNSTATS_PROC_TIME	TOTAL_COMPILE_TIME	TOTAL_COMPILE_PROC_TIME	total_sync_runstats_time - 同期 RUNSTATS の合計時間
TOTAL_CONNECT_REQUEST_TIME	TOTAL_CONNECT_REQUEST_PROC_TIME	TOTAL_RQST_TIME	TOTAL_RQST_TIME	total_connect_request_time - 接続要求またはユーザー切り替え要求の合計時間
TOTAL_CONNECT_AUTHENTICATION_TIME	TOTAL_CONNECT_AUTHENTICATION_PROC_TIME	TOTAL_CONNECT_REQUEST_TIME	TOTAL_CONNECT_REQUEST_PROC_TIME	total_connect_authentication_time - 接続またはユーザー切り替えの認証要求の合計時間

タイプ *activity\_metrics* のElementを含む XML 文書は、以下のインターフェースから生成されます。

- MON\_GET\_ACTIVITY\_DETAILS
- MON\_GET\_PKG\_CACHE\_STMT\_DETAILS
- ACTIVITY イベント・モニターからの DETAILS\_XML 列
- PACKAGE CACHE イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_TABLES によって生成される METRICS 列
- PACKAGE CACHE イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_XML の XMLREPORT 列

この場合に XML 文書から戻されるメトリックの種類とその親メトリックについては、表 114 を参照してください。

表 114. *activity\_metrics* Element・タイプを含む XML 文書に対する  
*MON\_FORMAT\_XML\_COMPONENT\_TIMES\_BY\_ROW* によって戻されるメトリック名

メトリック名	処理メトリック名	親メトリック名	親処理メトリック名	説明またはモニター・Element
STMT_EXEC_TIME	NULL	NULL	NULL	stmt_exec_time - ステートメントの実行時間
TOTAL_ROUTINE_TIME	NULL	STMT_EXEC_TIME	NULL	total_routine_time - ルーチン時間の合計
TOTAL_ROUTINE_NON_SECT_TIME	TOTAL_ROUTINE_NON_SECT_PROC_TIME	TOTAL_ROUTINE_TIME	STMT_EXEC_TIME	total_routine_non_sect_time - 非セクション・ルーチン実行時間
TOTAL_ROUTINE_USER_CODE_TIME	TOTAL_ROUTINE_USER_CODE_PROC_TIME	TOTAL_ROUTINE_NON_SECT_TIME	TOTAL_ROUTINE_NON_SECT_PROC_TIME	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
TOTAL_SECTION_TIME	TOTAL_SECTION_PROC_TIME	STMT_EXEC_TIME	STMT_EXEC_TIME	total_section_time - セクション時間の合計
TOTAL_SECTION_SORT_TIME	TOTAL_SECTION_SORT_PROC_TIME	TOTAL_SECTION_TIME	TOTAL_SECTION_PROC_TIME	total_section_sort_time - セクション・ソート時間合計

## MON\_FORMAT\_XML\_METRICS\_BY\_ROW - すべてのメトリックに関するフォーマット設定された行ベースの出力の取得

MON\_FORMAT\_XML\_METRICS\_BY\_ROW 表関数は、XML メトリック文書に含まれるすべてのメトリックに関するフォーマット設定された行ベースの出力を戻します。

### 構文

```
►►—MON_FORMAT_XML_METRICS_BY_ROW—(—xml doc—)—————►►
```

スキーマは SYSPROC です。

### 表関数パラメーター

*xml doc*

system\_metrics または activity\_metrics エレメントを持つ XML 文書が含まれる、タイプ BLOB(100M) の入力引数。これらのエレメントを持つ XML 文書は、以下のソースから入手可能です。

- いずれかの MON\_GET\_\*\_DETAILS 表関数からの戻り。
- 統計およびアクティビティ・イベント・モニターからのメトリック列出力。
- 作業単位またはパッケージ・キャッシュ・イベント・モニターからのフォーマット設定された出力。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

この例は、MON\_GET\_WORKLOAD\_DETAILS 表関数によって生成された XML 文書から行ベースのフォーマット設定された情報を戻すために、

MON\_FORMAT\_XML\_METRICS\_BY\_ROW 表関数を呼び出す方法を示しています。

```
SELECT SUBSTR(TFXML.WORKLOAD_NAME, 1, 13) AS WORKLOAD_NAME,  
       SUBSTR(METRICS.METRIC_NAME, 1, 25) AS METRIC_NAME,  
       METRICS.VALUE  
FROM  
       TABLE( MON_GET_WORKLOAD_DETAILS( NULL, -2 ) ) AS TFXML,  
       TABLE( MON_FORMAT_XML_METRICS_BY_ROW( TFXML.DETAILS ) ) AS METRICS  
ORDER BY METRICS.VALUE DESC
```

以下は、この照会によって出力されるリストの一部です。

WORKLOAD_NAME	METRIC_NAME	VALUE
PAYROLL	ACT_COMPLETED_TOTAL	15
FINANCE	ACT_COMPLETED_TOTAL	12
PAYROLL	LOCK_WAITS	8
FINANCE	LOCK_WAITS	5
FINANCE	DEADLOCKS	3
PAYROLL	DEADLOCKS	0

## 戻される情報

表 115. *MON\_FORMAT\_XML\_METRICS\_BY\_ROW* で戻される情報

列名	データ・タイプ	説明
METRIC_NAME	VARCHAR(128)	合計時間メトリック値のユニーク ID。
VALUE	BIGINT	メトリックの現行値。

タイプ *system\_metrics* のエレメントを含む XML 文書は、以下のインターフェースから生成されます。

- MON\_GET\_CONNECTION\_DETAILS
- MON\_GET\_SERVICE\_SUBCLASS\_DETAILS
- MON\_GET\_UNIT\_OF\_WORK\_DETAILS
- MON\_GET\_WORKLOAD\_DETAILS
- STATISTICS イベント・モニターからの DETAILS\_XML 列
- UNIT OF WORK イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_TABLES によって生成される METRICS 列
- UNIT OF WORK イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_XML の XMLREPORT 列

この場合に XML 文書から戻されるメトリックについては、表 116 を参照してください。

表 116. *system\_metrics* エレメント・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_METRICS\_BY\_ROW* によって戻されるメトリック名

メトリック名	メトリックについての説明またはモニター・エレメント
WLM_QUEUE_TIME_TOTAL	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間
WLM_QUEUE_ASSIGNMENTS_TOTAL	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
FCM_TQ_RECV_WAIT_TIME	fcm_tq_recv_wait_time - FCM 表キュー受信待機時間
FCM_MESSAGE_RECV_WAIT_TIME	fcm_message_recv_wait_time - FCM メッセージ受信待機時間
FCM_TQ_SEND_WAIT_TIME	fcm_tq_send_wait_time - FCM 表キュー送信待機時間
FCM_MESSAGE_SEND_WAIT_TIME	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
AGENT_WAIT_TIME	agent_wait_time - エージェント待機時間
AGENT_WAITS_TOTAL	agent_waits_total - エージェント待機の合計
LOCK_WAIT_TIME	lock_wait_time - ロック待機中の時間
LOCK_WAITS	lock_waits - ロック待機数
DIRECT_READ_TIME	direct_read_time - 直接読み取り時間
DIRECT_READ_REQS	direct_read_reqs - 直接読み取り要求



表 116. system\_metrics エlement・タイプを含む XML 文書に対する MON\_FORMAT\_XML\_METRICS\_BY\_ROW によって戻されるメトリック名 (続き)

メトリック名	メトリックについての説明またはモニター・Element
DIRECT_WRITE_TIME	direct_write_time - 直接書き込み時間
DIRECT_WRITE_REQS	direct_write_reqs - 直接書き込み要求
LOG_BUFFER_WAIT_TIME	log_buffer_wait_time - ログ・バッファ待ち時間
NUM_LOG_BUFFER_FULL	num_log_buffer_full - エージェントの待機が発生したフル・ログ・バッファの回数
LOG_DISK_WAIT_TIME	log_disk_wait_time - ログ・ディスク待機時間
LOG_DISK_WAITS_TOTAL	log_disk_waits_total - ログ・ディスク待機の合計
TCPIP_RECV_WAIT_TIME	tcPIP_recv_wait_time - TCP/IP 受信待機時間
TCPIP_RECVS_TOTAL	tcPIP_recvs_total - TCP/IP 受信の合計
CLIENT_IDLE_WAIT_TIME	client_idle_wait_time - クライアントのアイドル待機時間
IPC_RECV_WAIT_TIME	ipc_recv_wait_time - プロセス間通信受信待機時間
IPC_RECVS_TOTAL	ipc_recvs_total - プロセス間通信受信の合計
IPC_SEND_WAIT_TIME	ipc_send_wait_time - プロセス間通信送信待ち時間
IPC_SENDS_TOTAL	ipc_sends_total - プロセス間通信合計送信数
TCPIP_SEND_WAIT_TIME	tcPIP_send_wait_time - TCP/IP 送信待ち時間
TCPIP_SENDS_TOTAL	tcPIP_sends_total - TCP/IP 合計送信数
POOL_WRITE_TIME	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_READ_TIME	pool_read_time - バッファ・プール物理読み取り時間の合計
AUDIT_FILE_WRITE_WAIT_TIME	audit_file_write_wait_time - 監査ファイル書き込み待機時間
AUDIT_FILE_WRITES_TOTAL	audit_file_writes_total - 書き込まれた監査ファイルの合計
AUDIT_SUBSYSTEM_WAIT_TIME	audit_subsystem_wait_time - 監査サブシステム待機時間
AUDIT_SUBSYSTEM_WAITS_TOTAL	audit_subsystem_waits_total - 監査サブシステム待機の合計
DIAGLOG_WRITE_WAIT_TIME	diaglog_write_wait_time - 診断ログ・ファイル書き込み待機時間
DIAGLOG_WRITES_TOTAL	diaglog_writes_total - 診断ログ・ファイル書き込みの合計
FCM_SEND_WAIT_TIME	fcm_send_wait_time - FCM 送信待ち時間
FCM_RECV_WAIT_TIME	fcm_recv_wait_time - FCM 受信待機時間
TOTAL_ACT_WAIT_TIME	total_act_wait_time - 合計アクティビティ待機時間
TOTAL_WAIT_TIME	total_wait_time - 合計待ち時間
LOCK_WAIT_TIME_GLOBAL	lock_wait_time_global - グローバル・ロック待機時間
LOCK_WAITS_GLOBAL	lock_waits_global - グローバル・ロック待機
RECLAIM_WAIT_TIME	reclaim_wait_time - 再利用の待機時間
SPACEMAPPAGE_RECLAIM_WAIT_TIME	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間
CF_WAIT_TIME	cf_wait_time - クラスタ・キャッシング・ファシリティー待機時間
CF_WAITS	cf_waits - クラスタ・キャッシング・ファシリティー待機回数
EVMON_WAIT_TIME	evmon_wait_time - イベント・モニターの待機時間
EVMON_WAITS_TOTAL	evmon_waits_total - イベント・モニターの待機の合計回数
TOTAL_EXTENDED_LATCH_WAIT_TIME	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
TOTAL_EXTENDED_LATCH_WAITS	total_extended_latch_waits - 拡張ラッチの合計待機回数

表 116. system\_metrics エlement・タイプを含む XML 文書に対する MON\_FORMAT\_XML\_METRICS\_BY\_ROW によって戻されるメトリック名 (続き)

メトリック名	メトリックについての説明またはモニター・Element
PREFETCH_WAIT_TIME	prefetch_wait_time - プリフェッチ待ち時間
PREFETCH_WAITS	prefetch_waits - プリフェッチャーの待機カウント
COMM_EXIT_WAIT_TIME	comm_exit_wait_time - 通信バッファ出口待機時間モニター・Element
COMM_EXIT_WAITS	comm_exit_waits - 通信バッファ出口待機回数モニター・Element
TOTAL_RQST_TIME	total_rqst_time - 合計要求時間
RQSTS_COMPLETED_TOTAL	rqsts_completed_total - 完了した要求の合計数
TOTAL_APP_RQST_TIME	total_app_rqst_time - アプリケーション要求合計時間
APP_RQSTS_COMPLETED_TOTAL	app_rqsts_completed_total - 完了したアプリケーション要求の合計数
TOTAL_SECTION_SORT_PROC_TIME	total_section_sort_proc_time - セクション・ソート処理時間合計
TOTAL_SECTION_SORT_TIME	total_section_sort_time - セクション・ソート時間合計
TOTAL_SECTION_SORTS	total_section_sorts - セクション・ソート合計
TOTAL_ACT_TIME	total_act_time - 合計アクティビティー時間
TOTAL_ROUTINE_TIME	total_routine_time - ルーチン時間の合計
TOTAL_COMPILE_PROC_TIME	total_compile_proc_time - コンパイル処理時間の合計
TOTAL_COMPILE_TIME	total_compile_time - コンパイル時間の合計
TOTAL_COMPILATIONS	total_compilations - コンパイルの合計回数
TOTAL_IMPLICIT_COMPILE_PROC_TIME	total_implicit_compile_proc_time - 暗黙的コンパイルの処理時間の合計
TOTAL_IMPLICIT_COMPILE_TIME	total_implicit_compile_time - 暗黙的コンパイル時間の合計
TOTAL_IMPLICIT_COMPILATIONS	total_implicit_compilations - 暗黙的コンパイルの合計回数
TOTAL_RUNSTATS_PROC_TIME	total_runstats_proc_time - ランタイム統計の処理時間の合計
TOTAL_RUNSTATS_TIME	total_runstats_time - ランタイム統計時間の合計
TOTAL_RUNSTATS	total_runstats - ランタイム統計の合計回数
TOTAL_REORG_PROC_TIME	total_reorg_proc_time - 再編成の処理時間の合計
TOTAL_REORG_TIME	total_reorg_time - 再編成時間の合計
TOTAL_REORGS	total_reorgs - 再編成の合計回数
TOTAL_LOAD_PROC_TIME	total_load_proc_time - ロード処理時間の合計
TOTAL_LOAD_TIME	total_load_time - ロード時間の合計
TOTAL_LOADS	total_loads - ロードの合計回数
TOTAL_SECTION_PROC_TIME	total_section_proc_time - セクション処理時間の合計
TOTAL_SECTION_TIME	total_section_time - セクション時間の合計
TOTAL_APP_SECTION_EXECUTIONS	total_app_section_executions - アプリケーションのセクション実行数の合計
TOTAL_COMMIT_PROC_TIME	total_commit_proc_time - コミット処理時間の合計
TOTAL_COMMIT_TIME	total_commit_time - コミット時間の合計
TOTAL_APP_COMMITS	total_app_commits - アプリケーション・コミットの合計回数
TOTAL_ROLLBACK_PROC_TIME	total_rollback_proc_time - ロールバック処理時間の合計
TOTAL_ROLLBACK_TIME	total_rollback_time - ロールバック時間の合計

表 116. *system\_metrics* エlement・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_METRICS\_BY\_ROW* によって戻されるメトリック名 (続き)

メトリック名	メトリックについての説明またはモニター・Element
TOTAL_APP_ROLLBACKS	total_app_rollback - アプリケーション・ロールバックの合計回数
TOTAL_ROUTINE_USER_CODE_PROC_TIME	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
TOTAL_ROUTINE_USER_CODE_TIME	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
TOTAL_STATS_FABRICATION_PROC_TIME	total_stats_fabrication_proc_time - 統計作成の合計処理時間
TOTAL_STATS_FABRICATION_TIME	total_stats_fabrication_time - 統計作成の合計時間
TOTAL_STATS_FABRICATIONS	total_stats_fabrications - 統計作成の合計回数
TOTAL_SYNC_RUNSTATS_PROC_TIME	total_sync_runstats_proc_time - 同期 RUNSTATS 処理時間の合計
TOTAL_SYNC_RUNSTATS_TIME	total_sync_runstats_time - 同期 RUNSTATS 時間の合計
TOTAL_SYNC_RUNSTATS	total_sync_runstats - 同期 RUNSTATS アクティビティの合計
TOTAL_CONNECT_REQUEST_PROC_TIME	total_connect_request_proc_time - 接続要求またはユーザー切り替え要求の合計処理時間
TOTAL_CONNECT_REQUEST_TIME	total_connect_request_time - 接続要求またはユーザー切り替え要求の合計時間
TOTAL_CONNECT_REQUESTS	total_connect_requests - 接続要求またはユーザー切り替え要求
TOTAL_CONNECT_AUTHENTICATION_PROC_TIME	total_connect_authentication_proc_time - 接続認証の合計処理時間
TOTAL_CONNECT_AUTHENTICATION_TIME	total_connect_authentication_time - 接続またはユーザー切り替えの認証要求の合計時間
TOTAL_CONNECT_AUTHENTIFICATIONS	total_connect_authentications - 接続またはユーザー切り替えの認証の実行回数
ROWS_READ	rows_read - 読み取り行数
ROWS_MODIFIED	rows_modified - 変更された行数
POOL_DATA_L_READS	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_INDEX_L_READS	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_TEMP_DATA_L_READS	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_INDEX_L_READS	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_XDA_L_READS	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_TEMP_XDA_L_READS	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
TOTAL_CPU_TIME	total_cpu_time - 合計 CPU 時間
ACT_COMPLETED_TOTAL	act_completed_total - 完了したアクティビティの合計数
CAT_CACHE_LOOKUPS	cat_cache_lookups - カタログ・キャッシュ参照数
PKG_CACHE_LOOKUPS	pkg_cache_lookups - パッケージ・キャッシュ参照
POOL_DATA_P_READS	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_TEMP_DATA_P_READS	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_XDA_P_READS	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り

表 116. *system\_metrics* エlement・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_METRICS\_BY\_ROW* によって戻されるメトリック名 (続き)

メトリック名	メトリックについての説明またはモニター・Element
POOL_TEMP_XDA_P_READS	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り
POOL_INDEX_P_READS	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_INDEX_P_READS	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_DATA_WRITES	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_XDA_WRITES	pool_xda_writes - バッファ・プール XDA データの書き込み
POOL_INDEX_WRITES	pool_index_writes - バッファ・プール索引の書き込み
DIRECT_READS	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	direct_writes - データベースへの直接書き込み
ROWS_RETURNED	rows_returned - 戻り行数
DEADLOCKS	deadlocks - デッドロック検出数
LOCK_TIMEOUTS	lock_timeouts - ロック・タイムアウト数
LOCK_ESCALS	lock_escalations - ロック・エスカレーション数
FCM_SENDS_TOTAL	fcm_sends_total - FCM 合計送信数
FCM_RECVS_TOTAL	fcm_recvs_total - FCM 受信の合計
FCM_SEND_VOLUME	fcm_send_volume - FCM 送信ボリューム
FCM_RECV_VOLUME	fcm_recv_volume - FCM 受信ボリューム
FCM_MESSAGE_SENDS_TOTAL	fcm_message_sends_total - FCM メッセージ送信の合計
FCM_MESSAGE_RECVS_TOTAL	fcm_message_recvs_total - FCM メッセージ受信の合計
FCM_MESSAGE_SEND_VOLUME	fcm_message_send_volume - FCM メッセージ送信ボリューム
FCM_MESSAGE_RECV_VOLUME	fcm_message_recv_volume - FCM メッセージ受信ボリューム
FCM_TQ_SENDS_TOTAL	fcm_tq_sends_total - FCM 表キュー送信の合計
FCM_TQ_RECVS_TOTAL	fcm_tq_recvs_total - FCM 表キュー受信の合計
FCM_TQ_SEND_VOLUME	fcm_tq_send_volume - FCM 表キューの送信ボリューム
FCM_TQ_RECV_VOLUME	fcm_tq_recv_volume - FCM 表キューの受信ボリューム
TQ_TOT_SEND_SPILLS	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
TCPIP_SEND_VOLUME	tcpip_send_volume - TCP/IP 送信ボリューム
TCPIP_RECV_VOLUME	tcpip_recv_volume - TCP/IP 受信ボリューム
IPC_SEND_VOLUME	ipc_send_volume - プロセス間通信送信ボリューム
IPC_RECV_VOLUME	ipc_recv_volume - プロセス間通信の受信ボリューム
POST_THRESHOLD_SORTS	post_threshold_sorts - ポストしきい値ソート
POST_SHRTHRESHOLD_SORTS	post_shrthreshold_sorts - ポスト共有しきい値ソート
SORT_OVERFLOWS	sort_overflows - ソート・オーバーフロー
AUDIT_EVENTS_TOTAL	audit_events_total - 監査イベント合計数
ACT_REJECTED_TOTAL	act_rejected_total - リジェクトされたアクティビティの合計数
ACT_ABORTED_TOTAL	act_aborted_total - 打ち切られたアクティビティの合計数
TOTAL_SORTS	total_sorts - ソート合計
LOCK_TIMEOUTS_GLOBAL	lock_timeouts_global - グローバル・ロック・タイムアウト

表 116. *system\_metrics* エlement・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_METRICS\_BY\_ROW* によって戻されるメトリック名 (続き)

メトリック名	メトリックについての説明またはモニター・Element
LOCK_ESCALS_MAXLOCKS	lock_escals_maxlocks - maxlocks ロック・エスカレーション数
LOCK_ESCALS_LOCKLIST	lock_escals_locklist - locklist ロック・エスカレーション数
LOCK_ESCALS_GLOBAL	lock_escals_global - グローバル・ロック・エスカレーション数
POOL_DATA_GBP_L_READS	pool_data_gbp_l_reads - グループ・バッファ・プールのデータの論理読み取り
POOL_DATA_GBP_P_READS	pool_data_gbp_p_reads - グループ・バッファ・プールのデータの物理読み取り
POOL_DATA_LBP_PAGES_FOUND	pool_data_lbp_pages_found - ローカル・バッファ・プールで検出されたデータ・ページ
POOL_DATA_GBP_INVALID_PAGES	pool_data_gbp_invalid_pages - グループ・バッファ・プールの無効なデータ・ページ
POOL_INDEX_GBP_L_READS	pool_index_gbp_l_reads - グループ・バッファ・プールの索引の論理読み取り
POOL_INDEX_GBP_P_READS	pool_index_gbp_p_reads - グループ・バッファ・プールの索引の物理読み取り
POOL_INDEX_LBP_PAGES_FOUND	pool_index_lbp_pages_found - 検出されたローカル・バッファ・プールの索引ページ
POOL_INDEX_GBP_INVALID_PAGES	pool_index_gbp_invalid_pages - グループ・バッファ・プールの無効な索引ページ
POOL_XDA_GBP_L_READS	pool_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの論理読み取りの要求
POOL_XDA_GBP_P_READS	pool_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの物理読み取りの要求
POOL_XDA_LBP_PAGES_FOUND	pool_xda_lbp_pages_found - 検出されたローカル・バッファ・プールの XDA データ・ページ
POOL_XDA_GBP_INVALID_PAGES	pool_xda_gbp_invalid_pages - グループ・バッファ・プールの無効な XDA データ・ページ
THRESH_VIOLATIONS	thresh_violations - しきい値違反の回数
NUM_LW_THRESH_EXCEEDED	num_lw_thresh_exceeded - ロック待機の超過されたしきい値の数
TOTAL_ROUTINE_INVOCATIONS	total_routine_invocations - ルーチン呼び出しの合計回数
INT_COMMITS	int_commits - 内部コミット数
INT_ROLLBACKS	int_rollback - 内部ロールバック数
CAT_CACHE_INSERTS	cat_cache_inserts - カタログ・キャッシュ挿入数
PKG_CACHE_INSERTS	pkg_cache_inserts - パッケージ・キャッシュ挿入
ACT_RQSTS_TOTAL	act_rqsts_total - アクティビティ要求の合計
TOTAL_DISP_RUN_QUEUE_TIME	total_disp_run_queue_time - ディスパッチャー実行キュー時間の合計
POOL_QUEUED_ASYNC_DATA_REQS	pool_queued_async_data_reqs - データ・プリフェッチ要求
POOL_QUEUED_ASYNC_INDEX_REQS	pool_queued_async_index_reqs - 索引プリフェッチ要求
POOL_QUEUED_ASYNC_XDA_REQS	pool_queued_async_xda_reqs - XDA プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_DATA_REQS	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求

表 116. *system\_metrics* エlement・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_METRICS\_BY\_ROW* によって戻されるメトリック名 (続き)

メトリック名	メトリックについての説明またはモニター・Element
POOL_QUEUED_ASYNC_TEMP_INDEX_REQS	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_XDA_REQS	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求
POOL_QUEUED_ASYNC_OTHER_REQS	pool_queued_async_other_reqs - 非プリフェッチの要求
POOL_QUEUED_ASYNC_DATA_PAGES	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_INDEX_PAGES	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_XDA_PAGES	pool_queued_async_xda_pages - XDA ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_DATA_PAGES	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_INDEX_PAGES	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_XDA_PAGES	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求
POOL_FAILED_ASYNC_DATA_REQS	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求
POOL_FAILED_ASYNC_INDEX_REQS	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求
POOL_FAILED_ASYNC_XDA_REQS	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_DATA_REQS	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_INDEX_REQS	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_XDA_REQS	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求
POOL_FAILED_ASYNC_OTHER_REQS	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求
APP_ACT_COMPLETED_TOTAL	app_act_completed_total - 成功した外部コーディネーター・アクティビティの総数
APP_ACT_ABORTED_TOTAL	app_act_aborted_total - 失敗した外部コーディネーター・アクティビティの総数
APP_ACT_REJECTED_TOTAL	app_act_rejected_total - リジェクトされた外部コーディネーター・アクティビティの総数
TOTAL_PEDS	total_peds - partial early distinct の合計回数
DISABLED_PEDS	disabled_peds - 無効化された partial early distinct
POST_THRESHOLD_PEDS	post_threshold_peds - partial early distinct しきい値
TOTAL_PEAS	total_peas - partial early aggregation の合計回数
POST_THRESHOLD_PEAS	post_threshold_peas - partial early aggregation しきい値
TQ_SORT_HEAP_REQUESTS	tq_sort_heap_requests - 表キューのソート・ヒープの要求
TQ_SORT_HEAP_REJECTIONS	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否

表 116. *system\_metrics* エlement・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_METRICS\_BY\_ROW* によって戻されるメトリック名 (続き)

メトリック名	メトリックについての説明またはモニター・Element
POOL_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ
POOL_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ
POOL_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ
IDA_SEND_WAIT_TIME	インデータベース分析処理にデータを送信するために待機した合計時間。
IDA_SENDS_TOTAL	インデータベース分析処理にデータが送信された合計回数。
IDA_SEND_VOLUME	インデータベース分析処理にデータベース・サーバーから送信されたデータの合計量。値はバイト数で報告されます。
IDA_RECV_WAIT_TIME	インデータベース分析処理からデータを受信するために待機した合計時間。
IDA_RECVS_TOTAL	インデータベース分析処理からデータを受け取った合計回数。
IDA_RECV_VOLUME	インデータベース分析処理からデータベース・サーバーが受け取ったデータの合計量。値はバイト数で報告されます。

タイプ *activity\_metrics* のElementを含む XML 文書は、以下のインターフェースから生成されます。

- MON\_GET\_ACTIVITY\_DETAILS
- MON\_GET\_PKG\_CACHE\_STMT\_DETAILS
- ACTIVITY イベント・モニターからの DETAILS\_XML 列
- PACKAGE CACHE イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_TABLES によって生成される METRICS 列
- PACKAGE CACHE イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_XML の XMLREPORT 列

この場合に XML 文書から戻されるメトリックについては、表 117 を参照してください。

表 117. *activity\_metrics* Element・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_METRICS\_BY\_ROW* によって戻されるメトリック名

メトリック名	説明またはモニター・Element
FCM_TQ_RECV_WAIT_TIME	fcm_tq_recv_wait_time - FCM 表キュー受信待機時間
FCM_MESSAGE_RECV_WAIT_TIME	fcm_message_recv_wait_time - FCM メッセージ受信待機時間
FCM_TQ_SEND_WAIT_TIME	fcm_tq_send_wait_time - FCM 表キュー送信待機時間
FCM_MESSAGE_SEND_WAIT_TIME	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
LOCK_WAIT_TIME	lock_wait_time - ロック待機中の時間
LOCK_WAITS	lock_waits - ロック待機数
DIRECT_READ_TIME	direct_read_time - 直接読み取り時間
DIRECT_READ_REQS	direct_read_reqs - 直接読み取り要求

表 117. activity\_metrics エlement・タイプを含む XML 文書に対する MON\_FORMAT\_XML\_METRICS\_BY\_ROW によって戻されるメトリック名 (続き)

メトリック名	説明またはモニター・Element
DIRECT_WRITE_TIME	direct_write_time - 直接書き込み時間
DIRECT_WRITE_REQS	direct_write_reqs - 直接書き込み要求
LOG_BUFFER_WAIT_TIME	log_buffer_wait_time - ログ・バッファ待ち時間
NUM_LOG_BUFFER_FULL	num_log_buffer_full - エージェントの待機が発生したフル・ログ・バッファの回数
LOG_DISK_WAIT_TIME	log_disk_wait_time - ログ・ディスク待機時間
LOG_DISK_WAITS_TOTAL	log_disk_waits_total - ログ・ディスク待機の合計
POOL_WRITE_TIME	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_READ_TIME	pool_read_time - バッファ・プール物理読み取り時間の合計
AUDIT_FILE_WRITE_WAIT_TIME	audit_file_write_wait_time - 監査ファイル書き込み待機時間
AUDIT_FILE_WRITES_TOTAL	audit_file_writes_total - 書き込まれた監査ファイルの合計
AUDIT_SUBSYSTEM_WAIT_TIME	audit_subsystem_wait_time - 監査サブシステム待機時間
AUDIT_SUBSYSTEM_WAITS_TOTAL	audit_subsystem_waits_total - 監査サブシステム待機の合計
DIAGLOG_WRITE_WAIT_TIME	diaglog_write_wait_time - 診断ログ・ファイル書き込み待機時間
DIAGLOG_WRITES_TOTAL	diaglog_writes_total - 診断ログ・ファイル書き込みの合計
FCM_SEND_WAIT_TIME	fcm_send_wait_time - FCM 送信待ち時間
FCM_RECV_WAIT_TIME	fcm_recv_wait_time - FCM 受信待機時間
TOTAL_ACT_WAIT_TIME	total_act_wait_time - 合計アクティビティ待機時間
LOCK_WAIT_TIME_GLOBAL	lock_wait_time_global - グローバル・ロック待機時間
LOCK_WAITS_GLOBAL	lock_waits_global - グローバル・ロック待機
RECLAIM_WAIT_TIME	reclaim_wait_time - 再利用の待機時間
SPACEMAPPAGE_RECLAIM_WAIT_TIME	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間
CF_WAIT_TIME	cf_wait_time - クラスタ・キャッシング・ファシリティー待機時間
CF_WAITS	cf_waits - クラスタ・キャッシング・ファシリティー待機回数
EVMON_WAIT_TIME	evmon_wait_time - イベント・モニターの待機時間
EVMON_WAITS_TOTAL	evmon_waits_total - イベント・モニターの待機の合計回数
TOTAL_EXTENDED_LATCH_WAIT_TIME	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
TOTAL_EXTENDED_LATCH_WAITS	total_extended_latch_waits - 拡張ラッチの合計待機回数
PREFETCH_WAIT_TIME	prefetch_wait_time - プリフェッチ待ち時間
PREFETCH_WAITS	prefetch_waits - プリフェッチャーの待機カウント
WLM_QUEUE_TIME_TOTAL	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間
WLM_QUEUE_ASSIGNMENTS_TOTAL	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
TOTAL_SECTION_SORT_PROC_TIME	total_section_sort_proc_time - セクション・ソート処理時間合計
TOTAL_SECTION_SORT_TIME	total_section_sort_time - セクション・ソート時間合計
TOTAL_SECTION_SORTS	total_section_sorts - セクション・ソート合計
TOTAL_ACT_TIME	total_act_time - 合計アクティビティ時間
STMT_EXEC_TIME	stmt_exec_time - ステートメントの実行時間



表 117. activity\_metrics エlement・タイプを含む XML 文書に対する MON\_FORMAT\_XML\_METRICS\_BY\_ROW によって戻されるメトリック名 (続き)

メトリック名	説明またはモニター・Element
COORD_STMT_EXEC_TIME	coord_stmt_exec_time - コーディネーター・エージェントによるステートメントの実行時間
TOTAL_ROUTINE_NON_SECT_PROC_TIME	total_routine_non_sect_proc_time - 非セクション処理時間
TOTAL_ROUTINE_NON_SECT_TIME	total_routine_non_sect_time - 非セクション・ルーチン実行時間
TOTAL_SECTION_PROC_TIME	total_section_proc_time - セクション処理時間の合計
TOTAL_SECTION_TIME	total_section_time - セクション時間の合計
TOTAL_APP_SECTION_EXECUTIONS	total_app_section_executions - アプリケーションのセクション実行数の合計
TOTAL_ROUTINE_TIME	total_routine_time - ルーチン時間の合計
TOTAL_ROUTINE_USER_CODE_PROC_TIME	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
TOTAL_ROUTINE_USER_CODE_TIME	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
ROWS_READ	rows_read - 読み取り行数
ROWS_MODIFIED	rows_modified - 変更された行数
POOL_DATA_L_READS	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_INDEX_L_READS	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_TEMP_DATA_L_READS	pool_temp_data_l_reads - バッファ・プールの一時データの論理読み取り
POOL_TEMP_INDEX_L_READS	pool_temp_index_l_reads - バッファ・プールの一時索引の論理読み取り
POOL_XDA_L_READS	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_TEMP_XDA_L_READS	pool_temp_xda_l_reads - バッファ・プールの一時 XDA データの論理読み取り
TOTAL_CPU_TIME	total_cpu_time - 合計 CPU 時間
POOL_DATA_P_READS	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_TEMP_DATA_P_READS	pool_temp_data_p_reads - バッファ・プールの一時データの物理読み取り
POOL_XDA_P_READS	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_TEMP_XDA_P_READS	pool_temp_xda_p_reads - バッファ・プールの一時 XDA データの物理読み取り
POOL_INDEX_P_READS	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_INDEX_P_READS	pool_temp_index_p_reads - バッファ・プールの一時索引の物理読み取り
POOL_DATA_WRITES	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_XDA_WRITES	pool_xda_writes - バッファ・プール XDA データの書き込み
POOL_INDEX_WRITES	pool_index_writes - バッファ・プール索引の書き込み
DIRECT_READS	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	direct_writes - データベースへの直接書き込み
ROWS_RETURNED	rows_returned - 戻り行数
DEADLOCKS	deadlocks - デッドロック検出数
LOCK_TIMEOUTS	lock_timeouts - ロック・タイムアウト数
LOCK_ESCALS	lock_escalations - ロック・エスカレーション数
FCM_SENDS_TOTAL	fcm_sends_total - FCM 合計送信数

表 117. *activity\_metrics* エレメント・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_METRICS\_BY\_ROW* によって戻されるメトリック名 (続き)

メトリック名	説明またはモニター・エレメント
FCM_RECVS_TOTAL	fcm_recvs_total - FCM 受信の合計
FCM_SEND_VOLUME	fcm_send_volume - FCM 送信ボリューム
FCM_RECV_VOLUME	fcm_recv_volume - FCM 受信ボリューム
FCM_MESSAGE_SENDS_TOTAL	fcm_message_sends_total - FCM メッセージ送信の合計
FCM_MESSAGE_RECVS_TOTAL	fcm_message_recvs_total - FCM メッセージ受信の合計
FCM_MESSAGE_SEND_VOLUME	fcm_message_send_volume - FCM メッセージ送信ボリューム
FCM_MESSAGE_RECV_VOLUME	fcm_message_recv_volume - FCM メッセージ受信ボリューム
FCM_TQ_SENDS_TOTAL	fcm_tq_sends_total - FCM 表キュー送信の合計
FCM_TQ_RECVS_TOTAL	fcm_tq_recvs_total - FCM 表キュー受信の合計
FCM_TQ_SEND_VOLUME	fcm_tq_send_volume - FCM 表キューの送信ボリューム
FCM_TQ_RECV_VOLUME	fcm_tq_recv_volume - FCM 表キューの受信ボリューム
TQ_TOT_SEND_SPILLS	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
POST_THRESHOLD_SORTS	post_threshold_sorts - ポストしきい値ソート
POST_SHRTHRESHOLD_SORTS	post_shrthreshold_sorts - ポスト共有しきい値ソート
SORT_OVERFLOWS	sort_overflows - ソート・オーバーフロー
AUDIT_EVENTS_TOTAL	audit_events_total - 監査イベント合計数
TOTAL_SORTS	total_sorts - ソート合計
LOCK_TIMEOUTS_GLOBAL	lock_timeouts_global - グローバル・ロック・タイムアウト
LOCK_ESCALS_MAXLOCKS	lock_escal_maxlocks - maxlocks ロック・エスカレーション数
LOCK_ESCALS_LOCKLIST	lock_escal_locklist - locklist ロック・エスカレーション数
LOCK_ESCALS_GLOBAL	lock_escal_global - グローバル・ロック・エスカレーション数
POOL_DATA_GBP_L_READS	pool_data_gbp_l_reads - グループ・バッファ・プールのデータの論理読み取り
POOL_DATA_GBP_P_READS	pool_data_gbp_p_reads - グループ・バッファ・プールのデータの物理読み取り
POOL_DATA_LBP_PAGES_FOUND	pool_data_lbp_pages_found - ローカル・バッファ・プールで検出されたデータ・ページ
POOL_DATA_GBP_INVALID_PAGES	pool_data_gbp_invalid_pages - グループ・バッファ・プールの無効なデータ・ページ
POOL_INDEX_GBP_L_READS	pool_index_gbp_l_reads - グループ・バッファ・プールの索引の論理読み取り
POOL_INDEX_GBP_P_READS	pool_index_gbp_p_reads - グループ・バッファ・プールの索引の物理読み取り
POOL_INDEX_LBP_PAGES_FOUND	pool_index_lbp_pages_found - 検出されたローカル・バッファ・プールの索引ページ
POOL_INDEX_GBP_INVALID_PAGES	pool_index_gbp_invalid_pages - グループ・バッファ・プールの無効な索引ページ
POOL_XDA_GBP_L_READS	pool_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの論理読み取りの要求
POOL_XDA_GBP_P_READS	pool_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの物理読み取りの要求

表 117. activity\_metrics エlement・タイプを含む XML 文書に対する MON\_FORMAT\_XML\_METRICS\_BY\_ROW によって戻されるメトリック名 (続き)

メトリック名	説明またはモニター・Element
POOL_XDA_LBP_PAGES_FOUND	pool_xda_lbp_pages_found - 検出されたローカル・バッファ・プールの XDA データ・ページ
POOL_XDA_GBP_INVALID_PAGES	pool_xda_gbp_invalid_pages - グループ・バッファ・プールの無効な XDA データ・ページ
THRESH_VIOLATIONS	thresh_violations - しきい値違反の回数
NUM_LW_THRESH_EXCEEDED	num_lw_thresh_exceeded - ロック待機の超過されたしきい値の数
TOTAL_ROUTINE_INVOCATIONS	total_routine_invocations - ルーチン呼び出しの合計回数
TOTAL_DISP_RUN_QUEUE_TIME	total_disp_run_queue_time - ディスパッチャー実行キュー時間の合計
POOL_QUEUED_ASYNC_DATA_REQS	pool_queued_async_data_reqs - データ・プリフェッチ要求
POOL_QUEUED_ASYNC_INDEX_REQS	pool_queued_async_index_reqs - 索引プリフェッチ要求
POOL_QUEUED_ASYNC_XDA_REQS	pool_queued_async_xda_reqs - XDA プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_DATA_REQS	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_INDEX_REQS	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_XDA_REQS	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求
POOL_QUEUED_ASYNC_OTHER_REQS	pool_queued_async_other_reqs - 非プリフェッチの要求
POOL_QUEUED_ASYNC_DATA_PAGES	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_INDEX_PAGES	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_XDA_PAGES	pool_queued_async_xda_pages - XDA ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_DATA_PAGES	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_INDEX_PAGES	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_XDA_PAGES	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求
POOL_FAILED_ASYNC_DATA_REQS	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求
POOL_FAILED_ASYNC_INDEX_REQS	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求
POOL_FAILED_ASYNC_XDA_REQS	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_DATA_REQS	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_INDEX_REQS	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_XDA_REQS	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求
POOL_FAILED_ASYNC_OTHER_REQS	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求
TOTAL_PEDS	total_peds - partial early distinct の合計回数
DISABLED_PEDS	disabled_peds - 無効化された partial early distinct
POST_THRESHOLD_PEDS	post_threshold_peds - partial early distinct しきい値
TOTAL_PEAS	total_peas - partial early aggregation の合計回数

表 117. *activity\_metrics* エlement・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_METRICS\_BY\_ROW* によって戻されるメトリック名 (続き)

メトリック名	説明またはモニター・Element
POST_THRESHOLD_PEAS	post_threshold_peas - partial early aggregation しきい値
TQ_SORT_HEAP_REQUESTS	tq_sort_heap_requests - 表キューのソート・ヒープの要求
TQ_SORT_HEAP_REJECTIONS	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否
POOL_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ
POOL_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ
POOL_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ
IDA_SEND_WAIT_TIME	インデータベース分析処理にデータを送信するために待機した合計時間。
IDA_SENDS_TOTAL	インデータベース分析処理にデータが送信された合計回数。
IDA_SEND_VOLUME	インデータベース分析処理にデータベース・サーバーから送信されたデータの合計量。値はバイト数で報告されます。
IDA_RECV_WAIT_TIME	インデータベース分析処理からデータを受信するために待機した合計時間。
IDA_RECVS_TOTAL	インデータベース分析処理からデータを受け取った合計回数。
IDA_RECV_VOLUME	インデータベース分析処理からデータベース・サーバーが受け取ったデータの合計量。値はバイト数で報告されます。

## MON\_FORMAT\_XML\_TIMES\_BY\_ROW - フォーマット設定された行ベースの待機/処理時間の結合された階層を取得する

*MON\_FORMAT\_XML\_TIMES\_BY\_ROW* 表関数は、XML メトリック文書に含まれる待機時間と処理時間の結合された階層に関するフォーマット設定された行ベースの出力を戻します。

### 構文

►►—*MON\_FORMAT\_XML\_TIMES\_BY\_ROW*—(—*xml doc*—)—————►►

スキーマは *SYSPROC* です。

### 表関数パラメーター

*xml doc*

*system\_metrics* または *activity\_metrics* Elementを持つ XML 文書が含まれる、タイプ *BLOB(100M)* の入力引数。これらのElementを持つ XML 文書は、以下のソースから入手可能です。

- いずれかの *MON\_GET\_\*\_DETAILS* 表関数からの戻り。
- 統計およびアクティビティ・イベント・モニターからのメトリック列出力。
- 作業単位またはパッケージ・キャッシュ・イベント・モニターからのフォーマット設定された出力。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

DB2 データベース・マネージャーの中で、アプリケーションによって時間が費やされている場所を判別するには、以下の照会を実行することにより、待機/処理時間メトリックの組み合わせをメトリック階層で表示することができます。

```
SELECT SUBSTR(T.SERVICE_SUPERCLASS_NAME,1,15) as SUPERCLASS,
       SUBSTR(T.SERVICE_SUBCLASS_NAME,1,15) as SUBCLASS,
       T.MEMBER,
       SUBSTR(U.METRIC_NAME, 1,15) AS METRIC_NAME,
       SUBSTR(U.PARENT_METRIC_NAME,1,15) AS PARENT_NAME,
       U.TOTAL_TIME_VALUE,
       U.COUNT
FROM
TABLE(MON_GET_SERVICE_SUBCLASS_DETAILS(NULL, NULL, -2)) AS T,
TABLE(MON_FORMAT_XML_TIMES_BY_ROW(T.DETAILS)) AS U
```

以下はこの照会の出力例です。

SUPERCLASS	SUBCLASS	MEMBER	METRIC_NAME	PARENT_NAME	T..._VALUE	COUNT
MYSC	MYSSC	0	FCM_MESSAGE_REC	FCM_RECV_WAIT_T	0	0
MYSC	MYSSC	0	FCM_TQ_RECV_WAI	FCM_RECV_WAIT_T	0	0
MYSC	MYSSC	0	FCM_MESSAGE_SEN	FCM_SEND_WAIT_T	0	0
MYSC	MYSSC	0	FCM_TQ_SEND_WAI	FCM_SEND_WAIT_T	0	0
MYSC	MYSSC	0	TOTAL_COMMIT_PR	TOTAL_RQST_TIME	300	1
MYSC	MYSSC	0	TOTAL_COMPILE_P	TOTAL_RQST_TIME	700	1
MYSC	MYSSC	0	TOTAL_IMPLICIT	TOTAL_RQST_TIME	0	0
MYSC	MYSSC	0	TOTAL_LOAD_PROG	TOTAL_RQST_TIME	0	0
MYSC	MYSSC	0	TOTAL_REORG_PRO	TOTAL_RQST_TIME	0	0
MYSC	MYSSC	0	TOTAL_ROLLBACK	TOTAL_RQST_TIME	0	0
MYSC	MYSSC	0	TOTAL_RUNSTATS	TOTAL_RQST_TIME	0	0
MYSC	MYSSC	0	TOTAL_SECTION_P	TOTAL_RQST_TIME	7322	1
MYSC	MYSSC	0	TOTAL_WAIT_TIME	TOTAL_RQST_TIME	0	0
MYSC	MYSSC	0	TOTAL_SECTION_S	TOTAL_SECTION_P	0	0
MYSC	MYSSC	0	AGENT_WAIT_TIME	TOTAL_WAIT_TIME	0	0
MYSC	MYSSC	0	AUDIT_FILE_WRIT	TOTAL_WAIT_TIME	0	0
MYSC	MYSSC	0	AUDIT_SUBSYSTEM	TOTAL_WAIT_TIME	0	0
MYSC	MYSSC	0	DIAGLOG_WRITE_W	TOTAL_WAIT_TIME	0	0
MYSC	MYSSC	0	DIRECT_READ_TIM	TOTAL_WAIT_TIME	1204	17
MYSC	MYSSC	0	DIRECT_WRITE_TI	TOTAL_WAIT_TIME	0	0
MYSC	MYSSC	0	FCM_RECV_WAIT_T	TOTAL_WAIT_TIME	0	0
MYSC	MYSSC	0	FCM_SEND_WAIT_T	TOTAL_WAIT_TIME	0	0
MYSC	MYSSC	0	IPC_RECV_WAIT_T	TOTAL_WAIT_TIME	0	0
MYSC	MYSSC	0	IPC_SEND_WAIT_T	TOTAL_WAIT_TIME	0	0
MYSC	MYSSC	0	LOCK_WAIT_TIME	TOTAL_WAIT_TIME	0	0
MYSC	MYSSC	0	LOG_BUFFER_WAIT	TOTAL_WAIT_TIME	0	0
MYSC	MYSSC	0	LOG_DISK_WAIT_T	TOTAL_WAIT_TIME	523	2

```

MYSC      MYSSC      0 POOL_READ_TIME TOTAL_WAIT_TIME      2432      7
MYSC      MYSSC      0 POOL_WRITE_TIME TOTAL_WAIT_TIME          0          0
MYSC      MYSSC      0 TCPIP_RECV_WAIT TOTAL_WAIT_TIME       523        1
MYSC      MYSSC      0 TCPIP_SEND_WAIT TOTAL_WAIT_TIME       241        1
MYSC      MYSSC      0 WLM_QUEUE_TIME  TOTAL_WAIT_TIME          0          0
MYSC      MYSSC      0 CLIENT_IDLE_WAI -          234        -
MYSC      MYSSC      0 TOTAL_RQST_TIME -          13245      1
34 record(s) selected.

```

## 戻される情報

表 118. *MON\_FORMAT\_XML\_TIMES\_BY\_ROW* で戻される情報

列名	データ・タイプ	説明
METRIC_NAME	VARCHAR(128)	合計時間メトリック値のユニーク ID。
TOTAL_TIME_VALUE	BIGINT	metric_name に対応する合計時間値 (ミリ秒)。
COUNT	BIGINT	count - イベント・モニター・オーバーフロー数モニター・エレメント
PARENT_METRIC_NAME	VARCHAR(128)	親である合計時間メトリックの ID。そのメトリックの値には total_time_value がサブセットとして含まれます。

タイプ *system\_metrics* のエレメントを含む XML 文書は、以下のインターフェースから生成されます。

- MON\_GET\_CONNECTION\_DETAILS
- MON\_GET\_SERVICE\_SUBCLASS\_DETAILS
- MON\_GET\_UNIT\_OF\_WORK\_DETAILS
- MON\_GET\_WORKLOAD\_DETAILS
- STATISTICS イベント・モニターからの DETAILS\_XML 列
- UNIT OF WORK イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_TABLES によって生成される METRICS 列
- UNIT OF WORK イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_XML の XMLREPORT 列

この場合に XML 文書から戻されるメトリックの種類とその親メトリックについては、表 119 を参照してください。

表 119. *system\_metrics* エレメント・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_TIMES\_BY\_ROW* によって戻されるメトリック名

メトリック名	親メトリック名	メトリックについての説明またはモニター・エレメント
WLM_QUEUE_TIME_TOTAL	TOTAL_WAIT_TIME	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間
FCM_TQ_RECV_WAIT_TIME	FCM_RECV_WAIT_TIME	fcm_tq_recv_wait_time - FCM 表キュー受信待機時間
FCM_MESSAGE_RECV_WAIT_TIME	FCM_RECV_WAIT_TIME	fcm_message_recv_wait_time - FCM メッセージ受信待機時間
FCM_TQ_SEND_WAIT_TIME	FCM_SEND_WAIT_TIME	fcm_tq_send_wait_time - FCM 表キュー送信待機時間
FCM_MESSAGE_SEND_WAIT_TIME	FCM_SEND_WAIT_TIME	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
AGENT_WAIT_TIME	TOTAL_WAIT_TIME	agent_wait_time - エージェント待機時間

表 119. system\_metrics エlement・タイプを含む XML 文書に対する MON\_FORMAT\_XML\_TIMES\_BY\_ROW によって戻されるメトリック名 (続き)

メトリック名	親メトリック名	メトリックについての説明またはモニター・エレメント
LOCK_WAIT_TIME	TOTAL_WAIT_TIME	lock_wait_time - ロック待機中の時間
DIRECT_READ_TIME	TOTAL_WAIT_TIME	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	TOTAL_WAIT_TIME	direct_write_time - 直接書き込み時間
LOG_BUFFER_WAIT_TIME	TOTAL_WAIT_TIME	log_buffer_wait_time - ログ・バッファ待ち時間
LOG_DISK_WAIT_TIME	TOTAL_WAIT_TIME	log_disk_wait_time - ログ・ディスク待機時間
TCPIP_RECV_WAIT_TIME	TOTAL_WAIT_TIME	tcPIP_recv_wait_time - TCP/IP 受信待機時間
CLIENT_IDLE_WAIT_TIME	NULL	client_idle_wait_time - クライアントのアイドル待機時間
IPC_RECV_WAIT_TIME	TOTAL_WAIT_TIME	ipc_recv_wait_time - プロセス間通信受信待機時間
IPC_SEND_WAIT_TIME	TOTAL_WAIT_TIME	ipc_send_wait_time - プロセス間通信送信待ち時間
TCPIP_SEND_WAIT_TIME	TOTAL_WAIT_TIME	tcPIP_send_wait_time - TCP/IP 送信待ち時間
POOL_WRITE_TIME	TOTAL_WAIT_TIME	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_READ_TIME	TOTAL_WAIT_TIME	pool_read_time - バッファ・プール物理読み取り時間の合計
AUDIT_FILE_WRITE_WAIT_TIME	TOTAL_WAIT_TIME	audit_file_write_wait_time - 監査ファイル書き込み待機時間
AUDIT_SUBSYSTEM_WAIT_TIME	TOTAL_WAIT_TIME	audit_subsystem_wait_time - 監査サブシステム待機時間
DIAGLOG_WRITE_WAIT_TIME	TOTAL_WAIT_TIME	diaglog_write_wait_time - 診断ログ・ファイル書き込み待機時間
FCM_SEND_WAIT_TIME	TOTAL_WAIT_TIME	fcm_send_wait_time - FCM 送信待ち時間
FCM_RECV_WAIT_TIME	TOTAL_WAIT_TIME	fcm_recv_wait_time - FCM 受信待機時間
TOTAL_WAIT_TIME	TOTAL_RQST_TIME	total_wait_time - 合計待ち時間
LOCK_WAIT_TIME_GLOBAL	LOCK_WAIT_TIME	lock_wait_time_global - グローバル・ロック待機時間
RECLAIM_WAIT_TIME	TOTAL_WAIT_TIME	reclaim_wait_time - 再利用の待機時間
SPACEMAPPAGE_RECLAIM_WAIT_TIME	TOTAL_WAIT_TIME	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間
CF_WAIT_TIME	TOTAL_WAIT_TIME	cf_wait_time - クラスタ・キャッシング・ファシリティー待機時間
EVMON_WAIT_TIME	TOTAL_WAIT_TIME	evmon_wait_time - イベント・モニターの待機時間
TOTAL_EXTENDED_LATCH_WAIT_TIME	TOTAL_WAIT_TIME	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
PREFETCH_WAIT_TIME	TOTAL_WAIT_TIME	prefetch_wait_time - プリフェッチ待ち時間
COMM_EXIT_WAIT_TIME	TOTAL_WAIT_TIME	comm_exit_wait_time - 通信バッファ出口待機時間モニター・エレメント
TOTAL_SECTION_SORT_PROC_TIME	TOTAL_SECTION_PROC_TIME	total_section_sort_proc_time - セクション・ソート処理時間合計
TOTAL_COMPILE_PROC_TIME	TOTAL_RQST_TIME	total_compile_proc_time - コンパイル処理時間の合計

表 119. *system\_metrics* エlement・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_TIMES\_BY\_ROW* によって戻されるメトリック名 (続き)

メトリック名	親メトリック名	メトリックについての説明またはモニター・エlement
TOTAL_IMPLICIT_COMPILE_PROC_TIME	TOTAL_RQST_TIME	total_implicit_compile_proc_time - 暗黙的コンパイルの処理時間の合計
TOTAL_RUNSTATS_PROC_TIME	TOTAL_RQST_TIME	total_runstats_proc_time - ランタイム統計の処理時間の合計
TOTAL_REORG_PROC_TIME	TOTAL_RQST_TIME	total_reorg_proc_time - 再編成の処理時間の合計
TOTAL_LOAD_PROC_TIME	TOTAL_RQST_TIME	total_load_proc_time - ロード処理時間の合計
TOTAL_SECTION_PROC_TIME	TOTAL_RQST_TIME	total_section_proc_time - セクション処理時間の合計
TOTAL_COMMIT_PROC_TIME	TOTAL_RQST_TIME	total_commit_proc_time - コミット処理時間の合計
TOTAL_ROLLBACK_PROC_TIME	TOTAL_RQST_TIME	total_rollback_proc_time - ロールバック処理時間の合計
TOTAL_ROUTINE_USER_CODE_PROC_TIME	TOTAL_RQST_TIME	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
TOTAL_STATS_FABRICATION_PROC_TIME	TOTAL_COMPILE_PROC_TIME	total_stats_fabrication_proc_time - 統計作成の合計処理時間
TOTAL_SYNC_RUNSTATS_PROC_TIME	TOTAL_COMPILE_PROC_TIME	total_sync_runstats_proc_time - 同期 RUNSTATS 処理時間の合計
TOTAL_CONNECT_REQUEST_PROC_TIME	TOTAL_RQST_TIME	total_connect_request_proc_time - 接続要求またはユーザー切り替え要求の合計処理時間
TOTAL_CONNECT_AUTHENTICATION_PROC_TIME	TOTAL_CONNECT_REQUEST_PROC_TIME	total_connect_authentication_proc_time - 接続認証の合計処理時間
TOTAL_RQST_TIME	NULL	total_rqst_time - 合計要求時間
IDA_SEND_WAIT_TIME	TOTAL_WAIT_TIME	インデータベース分析処理にデータを送信するために待機した合計時間。
IDA_RECV_WAIT_TIME	TOTAL_WAIT_TIME	インデータベース分析処理からデータを受信するために待機した合計時間。

タイプ *activity\_metrics* のエlementを含む XML 文書は、以下のインターフェースから生成されます。

- MON\_GET\_ACTIVITY\_DETAILS
- MON\_GET\_PKG\_CACHE\_STMT\_DETAILS
- ACTIVITY イベント・モニターからの DETAILS\_XML 列
- PACKAGE CACHE イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_TABLES によって生成される METRICS 列
- PACKAGE CACHE イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_XML の XMLREPORT 列

この場合に XML 文書から戻されるメトリックの種類とその親メトリックについては、507 ページの表 120 を参照してください。



表 120. activity\_metrics エlement・タイプを含む XML 文書に対する MON\_FORMAT\_XML\_TIMES\_BY\_ROW によって戻されるメトリック名

メトリック名	親メトリック名	説明またはモニター・Element
FCM_TQ_RECV_WAIT_TIME	FCM_RECV_WAIT_TIME	fcm_tq_recv_wait_time - FCM 表キュー受信待機時間
FCM_MESSAGE_RECV_WAIT_TIME	FCM_RECV_WAIT_TIME	fcm_message_recv_wait_time - FCM メッセージ受信待機時間
FCM_TQ_SEND_WAIT_TIME	FCM_SEND_WAIT_TIME	fcm_tq_send_wait_time - FCM 表キュー送信待機時間
FCM_MESSAGE_SEND_WAIT_TIME	FCM_SEND_WAIT_TIME	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
LOCK_WAIT_TIME	TOTAL_ACT_WAIT_TIME	lock_wait_time - ロック待機中の時間
DIRECT_READ_TIME	TOTAL_ACT_WAIT_TIME	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	TOTAL_ACT_WAIT_TIME	direct_write_time - 直接書き込み時間
LOG_BUFFER_WAIT_TIME	TOTAL_ACT_WAIT_TIME	log_buffer_wait_time - ログ・バッファ待ち時間
LOG_DISK_WAIT_TIME	TOTAL_ACT_WAIT_TIME	log_disk_wait_time - ログ・ディスク待機時間
POOL_WRITE_TIME	TOTAL_ACT_WAIT_TIME	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_READ_TIME	TOTAL_ACT_WAIT_TIME	pool_read_time - バッファ・プール物理読み取り時間の合計
AUDIT_FILE_WRITE_WAIT_TIME	TOTAL_ACT_WAIT_TIME	audit_file_write_wait_time - 監査ファイル書き込み待機時間
AUDIT_SUBSYSTEM_WAIT_TIME	TOTAL_ACT_WAIT_TIME	audit_subsystem_wait_time - 監査サブシステム待機時間
DIAGLOG_WRITE_WAIT_TIME	TOTAL_ACT_WAIT_TIME	diaglog_write_wait_time - 診断ログ・ファイル書き込み待機時間
FCM_SEND_WAIT_TIME	TOTAL_ACT_WAIT_TIME	fcm_send_wait_time - FCM 送信待ち時間
FCM_RECV_WAIT_TIME	TOTAL_ACT_WAIT_TIME	fcm_recv_wait_time - FCM 受信待機時間
TOTAL_ACT_WAIT_TIME	STMT_EXEC_TIME	total_act_wait_time - 合計アクティビティー待機時間
LOCK_WAIT_TIME_GLOBAL	LOCK_WAIT_TIME	lock_wait_time_global - グローバル・ロック待機時間
RECLAIM_WAIT_TIME	TOTAL_ACT_WAIT_TIME	reclaim_wait_time - 再利用の待機時間
SPACEMAPPAGE_RECLAIM_WAIT_TIME	TOTAL_ACT_WAIT_TIME	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間
CF_WAIT_TIME	TOTAL_ACT_WAIT_TIME	cf_wait_time - クラスタ・キャッシング・ファシリティー待機時間
EVMON_WAIT_TIME	TOTAL_ACT_WAIT_TIME	evmon_wait_time - イベント・モニターの待機時間
TOTAL_EXTENDED_LATCH_WAIT_TIME	TOTAL_ACT_WAIT_TIME	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
PREFETCH_WAIT_TIME	TOTAL_ACT_WAIT_TIME	prefetch_wait_time - プリフェッチ待ち時間
WLM_QUEUE_TIME_TOTAL	NULL	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間
TOTAL_SECTION_SORT_PROC_TIME	TOTAL_SECTION_PROC_TIME	total_section_sort_proc_time - セクション・ソート処理時間合計
TOTAL_ROUTINE_NON_SECT_PROC_TIME	STMT_EXEC_TIME	total_routine_non_sect_proc_time - 非セクション処理時間
TOTAL_SECTION_PROC_TIME	STMT_EXEC_TIME	total_section_proc_time - セクション処理時間の合計

表 120. *activity\_metrics* エlement・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_TIMES\_BY\_ROW* によって戻されるメトリック名 (続き)

メトリック名	親メトリック名	説明またはモニター・Element
TOTAL_ROUTINE_USER_CODE_PROC_TIME	TOTAL_ROUTINE_NON_SECT_PROC_TIME	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
STMT_EXEC_TIME	NULL	stmt_exec_time - ステートメントの実行時間
IDA_SEND_WAIT_TIME	TOTAL_WAIT_TIME	インデータベース分析処理にデータを送信するために待機した合計時間。
IDA_RECV_WAIT_TIME	TOTAL_WAIT_TIME	インデータベース分析処理からデータを受信するために待機した合計時間。

## MON\_FORMAT\_XML\_WAIT\_TIMES\_BY\_ROW - 待機時間に関するフォーマット設定された行ベースの出力の取得

*MON\_FORMAT\_XML\_WAIT\_TIMES\_BY\_ROW* 表関数は、XML メトリック文書に含まれる待機時間に関するフォーマット設定された行ベースの出力を戻します。

### 構文

►►—*MON\_FORMAT\_XML\_WAIT\_TIMES\_BY\_ROW*—(—*xml doc*—)—————►►

スキーマは *SYSPROC* です。

### 表関数パラメーター

#### *xml doc*

*system\_metrics* または *activity\_metrics* Elementを持つ XML 文書が含まれる、タイプ *BLOB(100M)* の入力引数。これらのElementを持つ XML 文書は、以下のソースから入手可能です。

- いずれかの *MON\_GET\_\*\_DETAILS* 表関数からの戻り。
- 統計およびアクティビティ・イベント・モニターからのメトリック列出力。
- 作業単位またはパッケージ・キャッシュ・イベント・モニターからのフォーマット設定された出力。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する *EXECUTE* 特権
- *DATAACCESS* 権限
- *DBADM* 権限
- *SQLADM* 権限

### デフォルトの *PUBLIC* 特権

なし

## 例

この例は、MON\_GET\_WORKLOAD\_DETAILS 表関数によって生成された XML 文書から行ベースのフォーマット設定された出力を戻すために、MON\_FORMAT\_XML\_WAIT\_TIMES\_BY\_ROW 表関数を呼び出す方法を示しています。各ワークロードのメトリックとその値が出力に示されます。

```
SELECT SUBSTR(TFXML.WORKLOAD_NAME, 1, 13) AS WORKLOAD_NAME,
       SUBSTR(WAITS.METRIC_NAME, 1, 25) AS METRIC_NAME,
       WAITS.TOTAL_TIME_VALUE,
       WAITS.COUNT
FROM
  TABLE( MON_GET_WORKLOAD_DETAILS( NULL, -2 ) ) AS TFXML,
  TABLE( MON_FORMAT_XML_WAIT_TIMES_BY_ROW(
                                               TFXML.DETAILS
                                               )) AS WAITS
ORDER BY WAITS.TOTAL_TIME_VALUE DESC
```

以下は、この照会によって出力されるリストの一部です。

WORKLOAD_NAME	METRIC_NAME	TOTAL_TIME_VALUE	COUNT
PAYROLL	CLIENT_IDLE_WAIT_TIME	2193672	174
FINANCE	CLIENT_IDLE_WAIT_TIME	738290	16
PAYROLL	DIRECT_READ_TIME	67892	81
FINANCE	DIRECT_READ_TIME	32343	8
FINANCE	LOCK_WAIT_TIME	8463	3
PAYROLL	LOCK_WAIT_TIME	55	1

## 戻される情報

表 121. MON\_FORMAT\_XML\_WAIT\_TIMES\_BY\_ROW で戻される情報

列名	データ・タイプ	説明
METRIC_NAME	VARCHAR(128)	合計時間メトリック値のユニーク ID。
TOTAL_TIME_VALUE	BIGINT	metric_name に対応する合計時間値 (ミリ秒)。
COUNT	BIGINT	count - イベント・モニター・オーバーフロー数モニター・エレメント
PARENT_METRIC_NAME	VARCHAR(128)	親である合計時間メトリックの ID。そのメトリックの値には total_time_value がサブセットとして含まれます。

タイプ *system\_metrics* のエレメントを含む XML 文書は、以下のインターフェースから生成されます。

- MON\_GET\_CONNECTION\_DETAILS
- MON\_GET\_SERVICE\_SUBCLASS\_DETAILS
- MON\_GET\_UNIT\_OF\_WORK\_DETAILS
- MON\_GET\_WORKLOAD\_DETAILS
- STATISTICS イベント・モニターからの DETAILS\_XML 列
- UNIT OF WORK イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_TABLES によって生成される METRICS 列
- UNIT OF WORK イベント・モニターに対する EVMON\_FORMAT\_UE\_TO\_XML の XMLREPORT 列

この場合に XML 文書から戻されるメトリックの種類とその親メトリックについては、表 122 を参照してください。

表 122. *system\_metrics* エレメント・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_WAIT\_TIMES\_BY\_ROW* によって戻されるメトリック名

メトリック名	親メトリック名	メトリックについての説明またはモニター・エレメント
WLM_QUEUE_TIME_TOTAL	TOTAL_WAIT_TIME	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間
FCM_TQ_RECV_WAIT_TIME	FCM_RECV_WAIT_TIME	fcm_tq_recv_wait_time - FCM 表キュー受信待機時間
FCM_MESSAGE_RECV_WAIT_TIME	FCM_RECV_WAIT_TIME	fcm_message_recv_wait_time - FCM メッセージ受信待機時間
FCM_TQ_SEND_WAIT_TIME	FCM_SEND_WAIT_TIME	fcm_tq_send_wait_time - FCM 表キュー送信待機時間
FCM_MESSAGE_SEND_WAIT_TIME	FCM_SEND_WAIT_TIME	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
AGENT_WAIT_TIME	TOTAL_WAIT_TIME	agent_wait_time - エージェント待機時間
LOCK_WAIT_TIME	TOTAL_WAIT_TIME	lock_wait_time - ロック待機中の時間
DIRECT_READ_TIME	TOTAL_WAIT_TIME	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	TOTAL_WAIT_TIME	direct_write_time - 直接書き込み時間
LOG_BUFFER_WAIT_TIME	TOTAL_WAIT_TIME	log_buffer_wait_time - ログ・バッファ待ち時間
LOG_DISK_WAIT_TIME	TOTAL_WAIT_TIME	log_disk_wait_time - ログ・ディスク待機時間
TCPIP_RECV_WAIT_TIME	TOTAL_WAIT_TIME	tcPIP_recv_wait_time - TCP/IP 受信待機時間
CLIENT_IDLE_WAIT_TIME	NULL	client_idle_wait_time - クライアントのアイドル待機時間
IPC_RECV_WAIT_TIME	TOTAL_WAIT_TIME	ipc_recv_wait_time - プロセス間通信受信待機時間
IPC_SEND_WAIT_TIME	TOTAL_WAIT_TIME	ipc_send_wait_time - プロセス間通信送信待ち時間
TCPIP_SEND_WAIT_TIME	TOTAL_WAIT_TIME	tcPIP_send_wait_time - TCP/IP 送信待ち時間
POOL_WRITE_TIME	TOTAL_WAIT_TIME	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_READ_TIME	TOTAL_WAIT_TIME	pool_read_time - バッファ・プール物理読み取り時間の合計
AUDIT_FILE_WRITE_WAIT_TIME	TOTAL_WAIT_TIME	audit_file_write_wait_time - 監査ファイル書き込み待機時間
AUDIT_SUBSYSTEM_WAIT_TIME	TOTAL_WAIT_TIME	audit_subsystem_wait_time - 監査サブシステム待機時間
DIAGLOG_WRITE_WAIT_TIME	TOTAL_WAIT_TIME	diaglog_write_wait_time - 診断ログ・ファイル書き込み待機時間
FCM_SEND_WAIT_TIME	TOTAL_WAIT_TIME	fcm_send_wait_time - FCM 送信待ち時間
FCM_RECV_WAIT_TIME	TOTAL_WAIT_TIME	fcm_recv_wait_time - FCM 受信待機時間
TOTAL_WAIT_TIME	TOTAL_RQST_TIME	total_wait_time - 合計待ち時間
LOCK_WAIT_TIME_GLOBAL	LOCK_WAIT_TIME	lock_wait_time_global - グローバル・ロック待機時間
RECLAIM_WAIT_TIME	TOTAL_WAIT_TIME	reclaim_wait_time - 再利用の待機時間
SPACEMAPPAGE_RECLAIM_WAIT_TIME	TOTAL_WAIT_TIME	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間

表 122. *system\_metrics* エlement・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_WAIT\_TIMES\_BY\_ROW* によって戻されるメトリック名 (続き)

メトリック名	親メトリック名	メトリックについての説明またはモニター・エlement
CF_WAIT_TIME	TOTAL_WAIT_TIME	cf_wait_time - クラスター・キャッシング・ファシリティー待機時間
EVMON_WAIT_TIME	TOTAL_WAIT_TIME	evmon_wait_time - イベント・モニターの待機時間
TOTAL_EXTENDED_LATCH_WAIT_TIME	TOTAL_WAIT_TIME	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
PREFETCH_WAIT_TIME	TOTAL_WAIT_TIME	prefetch_wait_time - プリフェッチ待ち時間
COMM_EXIT_WAIT_TIME	TOTAL_WAIT_TIME	comm_exit_wait_time - 通信バッファ出口待機時間 モニター・エlement
IDA_SEND_WAIT_TIME	TOTAL_WAIT_TIME	インデータベース分析処理にデータを送信するために待機した合計時間。
IDA_RECV_WAIT_TIME	TOTAL_WAIT_TIME	インデータベース分析処理からデータを受信するために待機した合計時間。

タイプ *activity\_metrics* のエlementを含む XML 文書は、以下のインターフェースから生成されます。

- MON\_GET\_ACTIVITY\_DETAILS
- MON\_GET\_PKG\_CACHE\_STMT\_DETAILS
- ACTIVITY イベント・モニターからの DETAILS\_XML 列
- PACKAGE CACHE イベント・モニターに対する  
EVMON\_FORMAT\_UE\_TO\_TABLES によって生成される METRICS 列
- PACKAGE CACHE イベント・モニターに対する  
EVMON\_FORMAT\_UE\_TO\_XML の XMLREPORT 列

この場合に XML 文書から戻されるメトリックの種類とその親メトリックについては、表 123 を参照してください。

表 123. *activity\_metrics* エlement・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_WAIT\_TIMES\_BY\_ROW* によって戻されるメトリック名

メトリック名	親メトリック名	説明またはモニター・エlement
FCM_TQ_RECV_WAIT_TIME	FCM_RECV_WAIT_TIME	fcm_tq_recv_wait_time - FCM 表キュー受信待機時間
FCM_MESSAGE_RECV_WAIT_TIME	FCM_RECV_WAIT_TIME	fcm_message_recv_wait_time - FCM メッセージ受信待機時間
FCM_TQ_SEND_WAIT_TIME	FCM_SEND_WAIT_TIME	fcm_tq_send_wait_time - FCM 表キュー送信待機時間
FCM_MESSAGE_SEND_WAIT_TIME	FCM_SEND_WAIT_TIME	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
LOCK_WAIT_TIME	TOTAL_ACT_WAIT_TIME	lock_wait_time - ロック待機中の時間
DIRECT_READ_TIME	TOTAL_ACT_WAIT_TIME	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	TOTAL_ACT_WAIT_TIME	direct_write_time - 直接書き込み時間
LOG_BUFFER_WAIT_TIME	TOTAL_ACT_WAIT_TIME	log_buffer_wait_time - ログ・バッファ待ち時間
LOG_DISK_WAIT_TIME	TOTAL_ACT_WAIT_TIME	log_disk_wait_time - ログ・ディスク待機時間

表 123. *activity\_metrics* エlement・タイプを含む XML 文書に対する *MON\_FORMAT\_XML\_WAIT\_TIMES\_BY\_ROW* によって戻されるメトリック名 (続き)

メトリック名	親メトリック名	説明またはモニター・Element
POOL_WRITE_TIME	TOTAL_ACT_WAIT_TIME	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_READ_TIME	TOTAL_ACT_WAIT_TIME	pool_read_time - バッファ・プール物理読み取り時間の合計
AUDIT_FILE_WRITE_WAIT_TIME	TOTAL_ACT_WAIT_TIME	audit_file_write_wait_time - 監査ファイル書き込み待機時間
AUDIT_SUBSYSTEM_WAIT_TIME	TOTAL_ACT_WAIT_TIME	audit_subsystem_wait_time - 監査サブシステム待機時間
DIAGLOG_WRITE_WAIT_TIME	TOTAL_ACT_WAIT_TIME	diaglog_write_wait_time - 診断ログ・ファイル書き込み待機時間
FCM_SEND_WAIT_TIME	TOTAL_ACT_WAIT_TIME	fcm_send_wait_time - FCM 送信待ち時間
FCM_RECV_WAIT_TIME	TOTAL_ACT_WAIT_TIME	fcm_recv_wait_time - FCM 受信待機時間
TOTAL_ACT_WAIT_TIME	STMT_EXEC_TIME	total_act_wait_time - 合計アクティビティ待機時間
LOCK_WAIT_TIME_GLOBAL	LOCK_WAIT_TIME	lock_wait_time_global - グローバル・ロック待機時間
RECLAIM_WAIT_TIME	TOTAL_ACT_WAIT_TIME	reclaim_wait_time - 再利用の待機時間
SPACEMAPPAGE_RECLAIM_WAIT_TIME	TOTAL_ACT_WAIT_TIME	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間
CF_WAIT_TIME	TOTAL_ACT_WAIT_TIME	cf_wait_time - クラスタ・キャッシング・ファシリティー待機時間
EVMON_WAIT_TIME	TOTAL_ACT_WAIT_TIME	evmon_wait_time - イベント・モニターの待機時間
TOTAL_EXTENDED_LATCH_WAIT_TIME	TOTAL_ACT_WAIT_TIME	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
PREFETCH_WAIT_TIME	TOTAL_ACT_WAIT_TIME	prefetch_wait_time - プリフェッチ待ち時間
WLM_QUEUE_TIME_TOTAL	NULL	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間
IDA_SEND_WAIT_TIME	TOTAL_WAIT_TIME	インデータベース分析処理にデータを送信するために待機した合計時間。
IDA_RECV_WAIT_TIME	TOTAL_WAIT_TIME	インデータベース分析処理からデータを受信するために待機した合計時間。

## MON\_GET\_ACTIVITY\_DETAILS 表関数 - 完全なアクティビティ詳細の取得

MON\_GET\_ACTIVITY\_DETAILS 表関数は、一般的なアクティビティ情報 (ステートメント・テキストなど) やアクティビティの一連のメトリックを含む、アクティビティについての詳細を戻します。

### 構文

```

▶▶ MON_GET_ACTIVITY_DETAILS (—application_handle—, —uow_id—, —
▶—activity_id—, —member—)

```

スキーマは SYSPROC です。

## 表関数パラメーター

### *application\_handle*

有効なアプリケーション・ハンドルを指定する、タイプ BIGINT の入力引数。引数が NULL の場合、行はこの関数から戻されず、SQL0171N エラーが戻されます。

### *uow\_id*

アプリケーション内で固有の有効な作業単位 ID を指定する、タイプ INTEGER の入力引数。引数が NULL の場合、行はこの関数から戻されず、SQL0171N エラーが戻されます。

### *activity\_id*

作業単位内で固有の有効なアクティビティ ID を指定する、タイプ INTEGER の入力引数。引数が NULL の場合、行はこの関数から戻されず、SQL0171N エラーが戻されます。

### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバー番号を指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

長時間実行される照会を確認し、時間を要しているのが実行または待機のどちらであるかを判別します (例えば、ロックまたは入出力のどちらでブロックされているかなど)。

注: 以下の照会を結合して 1 つのステートメントにすることができますが、分かりやすくするために 2 ステップで示されています。さらに、テキスト全体を検索したい場合には、実行可能 ID を使って MON\_GET\_PKG\_CACHE\_STMT 表関数からステートメント・テキストを取得できます。

1. まず、WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 表関数を使用して、アクティビティとそれらの開始時刻をリストします。

```
SELECT application_handle,  
       activity_id,  
       uow_id,  
       local_start_time
```

```
FROM TABLE(
  WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES(
    cast(NULL as bigint), -1)
) AS T
```

以下はこの照会の出力例です。

APPLICATION_HANDLE	ACTIVITY_ID	UOW_ID	LOCAL_START_TIME
7	1	1	2008-06-10-10.06.55.675668
16	1	1	2008-06-10-10.08.38.613610

2 record(s) selected.

- 次に、MON\_GET\_ACTIVITY\_DETAILS 表関数を使用してアクティビティーが待機に要した時間のパーセントを表示します。

```
SELECT actmetrics.application_handle,
  actmetrics.activity_id,
  actmetrics.uow_id,
  varchar(actmetrics.stmt_text, 50) as stmt_text,
  actmetrics.total_act_time,
  actmetrics.total_act_wait_time,
  CASE WHEN actmetrics.total_act_time > 0
    THEN DEC(
      FLOAT(actmetrics.total_act_wait_time) /
      FLOAT(actmetrics.total_act_time)) * 100, 5, 2)
    ELSE NULL
  END AS PERCENTAGE_WAIT_TIME
FROM TABLE(MON_GET_ACTIVITY_DETAILS(7, 2, 1, -2)) AS ACTDETAILS,
XMLTABLE (XMLNAMESPACES( DEFAULT 'http://www.ibm.com/xmlns/prod/db2/mon'),
'$actmetrics/db2_activity_details'
PASSING XMLPARSE(DOCUMENT ACTDETAILS.DETAILS) as "actmetrics"
COLUMNS "APPLICATION_HANDLE" INTEGER PATH 'application_handle',
"ACTIVITY_ID" INTEGER PATH 'activity_id',
"UOW_ID" INTEGER PATH 'uow_id',
"STMT_TEXT" VARCHAR(1024) PATH 'stmt_text',
"TOTAL_ACT_TIME" INTEGER PATH 'activity_metrics/total_act_time',
"TOTAL_ACT_WAIT_TIME" INTEGER PATH 'activity_metrics/total_act_wait_time'
) AS ACTMETRICS;
```

以下はこの照会の出力例です。

APPLICATION_HANDLE	ACTIVITY_ID	UOW_ID	...
7	1	2	...

1 record(s) selected.

照会の出力 (続き)。

```
... STMT_TEXT ...
... ----- ...
... select * from syscat.tables optimize for 1 row ...
```

照会の出力 (続き)。

TOTAL_ACT_TIME	TOTAL_ACT_WAIT_TIME	PERCENTAGE_WAIT_TIME
459	0	0.00

MON\_GET\_ACTIVITY\_DETAILS 表関数を使用することにより、システムで現在実行中のすべてのアクティビティーについての情報をキャプチャーする照会を作成します。



- 例 1: DB2 コマンド行プロセッサ (CLP) を使って以下のコマンドを実行します。

```
WITH A1 AS
  (SELECT * FROM TABLE(wlm_get_workload_occurrence_activities(null, -1))
   WHERE activity_id > 0 )
SELECT A1.application_handle,
       A1.activity_id,
       A1.uow_id,
       total_act_time,
       total_act_wait_time,
       varchar(actmetrics.stmt_text, 50) AS stmt_text FROM A1,
TABLE(MON_GET_ACTIVITY_DETAILS(A1.application_handle, A1.uow_id,A1.activity_id, -1))
  AS ACTDETAILS,
XMLTABLE (XMLNAMESPACES( DEFAULT 'http://www.ibm.com/xmlns/prod/db2/mon'),
 '$actmetrics/db2_activity_details'
 PASSING XMLPARSE(DOCUMENT ACTDETAILS.DETAILS) AS "actmetrics"
 COLUMNS "STMT_TEXT" VARCHAR(1024) PATH 'stmt_text',
 "TOTAL_ACT_TIME" INTEGER PATH 'activity_metrics/total_act_time',
 "TOTAL_ACT_WAIT_TIME" INTEGER PATH 'activity_metrics/total_act_wait_time' )
 AS ACTMETRICS
```

以下はこの照会の出力例です。

```
APP...HANDLE  A..._ID  UOW_ID  T...ACT_TIME  T...WAIT_TIME
-----
15             1         5         16             5
15             1         3         17             5
7              1        49          0             0
SQL0445W Value "with A1 as (select * from table(wlm_get_workload
3 record(s) selected with 1 warning messages printed.
```

以下は、この照会の出力例の続きを示しています。

```
... STMT_TEXT
... -----
... select name from sysibm.systables
... select * from sysibm.systables
... with A1 as (select * from table(wlm_get_workload_o
_occurrence_" has been truncated. SQLSTATE=01004

3 record(s) selected with 1 warning messages printed.
```

## 使用上の注意

MON\_GET\_ACTIVITY\_DETAILS 関数は、単一のアクティビティの詳細情報を 1 つの XML 文書として戻すため、出力フォーマットを最大限柔軟性のあるものにします。XML 出力には、記述情報 (例えば、ステートメント・テキスト) とメトリックの両方が含まれます。出力は XML パーサーで直接解析でき、例に示すように XMLTABLE 関数でリレーショナル形式に変換することもできます。

この関数で報告されるメトリック (例えば、CPU 使用率) は、アクティビティの存続期間中、定期的にアクティビティにロールアップされます。したがって、この表関数で報告される値は、直前の ROLLUP 時のシステムの現行状態を反映しています。

アクティビティ・メトリックは、ワークロードの COLLECT ACTIVITY METRICS 節、またはデータベース・レベルの mon\_act\_metrics データベース構成パラメーターで制御されます。メトリックは、アクティビティをサブミットする接続が、アクティビティ・メトリックが使用可能なワークロードまたはデータベースに関連付けられている場合に収集されます。アクティビティ・メトリックが、アクティビティについて収集されない場合、すべてのメトリックが 0 として報告されます。

MON\_GET\_ACTIVITY\_DETAILS 表関数は、アクティビティーが存在するメンバーごとに 1 行のデータを戻します。メトリックについて、メンバー全体の集約は実行されません。しかし、集約は SQL 照会を使用して実行できます。

DETAILS 列に戻される XML 文書のスキーマは、ファイル `sqllib/misc/DB2MonRoutines.xsd` で入手できます。詳細は、ファイル `sqllib/misc/DB2MonCommon.xsd` 内にあります。

## 戻される情報

表 124. MON\_GET\_ACTIVITY\_DETAILS について戻される情報

列名	データ・タイプ	説明
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル
UOW_ID	INTEGER	uow_id - 作業単位 ID
ACTIVITY_ID	INTEGER	activity_id - アクティビティー ID
MEMBER	SMALLINT	member - データベース・メンバー
DETAILS	BLOB(8M)	アクティビティーの詳細を含む XML 文書。エレメントの説明については、本書の表 125を参照してください。

以下の例は、DETAILS 列で戻される XML 文書の構造を示しています。

```
<db2_activity_details xmlns="http://www.ibm.com/xmlns/prod/db2/mon" release="90700000">
  <member>0</member>
  <application_handle>70</application_handle>
  <activity_id>1</activity_id>
  <activity_state>IDLE</activity_state>
  <activity_type>READ_DML</activity_type>
  <uow_id>1</uow_id>
  ...
  <activity_metrics release="90700000">
    <lock_wait_time>2000</lock_wait_time>
    ...
  </activity_metrics>
</db2_activity_details>
```

完全スキーマについては、`sqllib/misc/DB2MonRoutines.xsd` を参照してください。本書では、以下のような非プリミティブ XML 型定義を使用しています。

```
<xs:simpleType name = "executable_id_type" >
  <xs:annotation>
    <xs:documentation>
      The binary Executable ID
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base = "xs:hexBinary" >
    <xs:maxLength value = "32" />
  </xs:restriction>
</xs:simpleType>
```

## 戻される詳細メトリック

表 125. MON\_GET\_ACTIVITY\_DETAILS について戻される詳細メトリック

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
activity_id	xs:nonNegativeInteger	activity_id - アクティビティー ID
activity_state	xs:string	activity_state - アクティビティー状態
activity_type	xs:string	activity_type - アクティビティー・タイプ

表 125. MON\_GET\_ACTIVITY\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
activitytotaltime_threshold_id	xs:int	activitytotaltime_threshold_id - アクティビティー合計時間しきい値 ID
activitytotaltime_threshold_value	xs:dateTime	activitytotaltime_threshold_value - アクティビティー合計時間しきい値
activitytotaltime_threshold_violated	xs:short (1 = はい、0 = いいえ)	activitytotaltime_threshold_violated - アクティビティー合計時間しきい値の違反
aggsqltempespace_threshold_id	xs:int	aggsqltempespace_threshold_id - AggSQL 一時スペースしきい値 ID。
aggsqltempespace_threshold_value	xs:long	aggsqltempespace_threshold_value - AggSQL 一時スペースしきい値
aggsqltempespace_threshold_violated	xs:short (1 = はい、0 = いいえ)	aggsqltempespace_threshold_violated - AggSQL 一時スペースしきい値の違反
application_handle	xs:nonNegativeInteger	application_handle - アプリケーション・ハンドル
audit_events_total	xs:nonNegativeInteger	audit_events_total - 監査イベント合計数
audit_file_write_wait_time	xs:nonNegativeInteger	audit_file_write_wait_time - 監査ファイル書き込み待機時間
audit_file_writes_total	xs:nonNegativeInteger	audit_file_writes_total - 書き込まれた監査ファイルの合計数
audit_subsystem_wait_time	xs:nonNegativeInteger	audit_subsystem_wait_time - 監査サブシステム待機時間
audit_subsystem_waits_total	xs:nonNegativeInteger	audit_subsystem_waits_total - 監査サブシステム待機の合計
client_acctng	xs:string(255)	CURRENT CLIENT_ACCTNG 特殊レジスター
client_applname	xs:string(255)	CURRENT CLIENT_APPLNAME 特殊レジスター
client_userid	xs:string(255)	CURRENT CLIENT_USERID 特殊レジスター
client_wrkstnname	xs:string(255)	CURRENT CLIENT_WRKSTNNAME 特殊レジスター
concurrentdbcoordactivities_db_threshold_id	xs:int	concurrentdbcoordactivities_db_threshold_id - 並行データベース・コーディネーター・アクティビティーしきい値 ID
concurrentdbcoordactivities_db_threshold_queued	xs:short (1 = はい、0 = いいえ)	concurrentdbcoordactivities_db_threshold_queued - 並行データベース・コーディネーター・アクティビティーしきい値によるキュー待機
concurrentdbcoordactivities_db_threshold_value	xs:long	concurrentdbcoordactivities_db_threshold_value - 並行データベース・コーディネーター・アクティビティーしきい値
concurrentdbcoordactivities_db_threshold_violated	xs:short (1 = はい、0 = いいえ)	concurrentdbcoordactivities_db_threshold_violated - 並行データベース・コーディネーター・アクティビティーしきい値の違反
concurrentdbcoordactivities_subclass_threshold_id	xs:int	concurrentdbcoordactivities_subclass_threshold_id - 並行データベース・コーディネーター・アクティビティーのサブクラスしきい値 ID
concurrentdbcoordactivities_subclass_threshold_queued	xs:short (1 = はい、0 = いいえ)	concurrentdbcoordactivities_subclass_threshold_queued - 並行データベース・コーディネーター・アクティビティーのサブクラスしきい値によるキュー待機
concurrentdbcoordactivities_subclass_threshold_value	xs:long	concurrentdbcoordactivities_subclass_threshold_value - 並行データベース・コーディネーター・アクティビティーのサブクラスしきい値

表 125. MON\_GET\_ACTIVITY\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
concurrentdbcoordactivities_subclass_threshold_violated	xs:short (1 = はい, 0 = いいえ)	concurrentdbcoordactivities_subclass_threshold_violated - 並行データベース・コーディネーター・アクティビティのサブクラスしきい値の違反
concurrentdbcoordactivities_superclass_threshold_id	xs:int	concurrentdbcoordactivities_superclass_threshold_id - 並行データベース・コーディネーター・アクティビティのスーパークラスしきい値 ID
concurrentdbcoordactivities_superclass_threshold_queued	xs:short (1 = はい, 0 = いいえ)	concurrentdbcoordactivities_superclass_threshold_queued - 並行データベース・コーディネーター・アクティビティのスーパークラスしきい値によるキュー待機
concurrentdbcoordactivities_superclass_threshold_value	xs:long	concurrentdbcoordactivities_superclass_threshold_value - 並行データベース・コーディネーター・アクティビティのスーパークラスしきい値
concurrentdbcoordactivities_superclass_threshold_violated	xs:short (1 = はい, 0 = いいえ)	concurrentdbcoordactivities_superclass_threshold_violated - 並行データベース・コーディネーター・アクティビティのスーパークラスしきい値の違反
concurrentdbcoordactivities_wl_was_threshold_id	xs:int	concurrentdbcoordactivities_wl_was_threshold_id - 並行データベース・コーディネーター・アクティビティのワークロード作業アクション・セットしきい値 ID
concurrentdbcoordactivities_wl_was_threshold_queued	xs:short (1 = はい, 0 = いいえ)	concurrentdbcoordactivities_wl_was_threshold_queued - 並行データベース・コーディネーター・アクティビティのワークロード作業アクション・セットしきい値によるキュー待機
concurrentdbcoordactivities_wl_was_threshold_value	xs:long	concurrentdbcoordactivities_wl_was_threshold_value - 並行データベース・コーディネーター・アクティビティのワークロード作業アクション・セットしきい値
concurrentdbcoordactivities_wl_was_threshold_violated	xs:short (1 = はい, 0 = いいえ)	concurrentdbcoordactivities_wl_was_threshold_violated - 並行データベース・コーディネーター・アクティビティのワークロード作業アクション・セットしきい値の違反
concurrentdbcoordactivities_work_action_set_threshold_id	xs:int	concurrentdbcoordactivities_work_action_set_threshold_id - 並行データベース・コーディネーター・アクティビティの作業アクション・セットしきい値 ID
concurrentdbcoordactivities_work_action_set_threshold_queued	xs:short (1 = はい, 0 = いいえ)	concurrentdbcoordactivities_work_action_set_threshold_queued - 並行データベース・コーディネーター・アクティビティの作業アクション・セットしきい値によるキュー待機
concurrentdbcoordactivities_work_action_set_threshold_value	xs:long	concurrentdbcoordactivities_work_action_set_threshold_value - 並行データベース・コーディネーター・アクティビティの作業アクション・セットしきい値
concurrentdbcoordactivities_work_action_set_threshold_violated	xs:short (1 = はい, 0 = いいえ)	concurrentdbcoordactivities_work_action_set_threshold_violated - 並行データベース・コーディネーター・アクティビティの作業アクション・セットしきい値の違反
coord_member	xs:nonNegativeInteger	coord_member - コーディネーター・メンバー
coord_stmt_exec_time		coord_stmt_exec_time - コーディネーター・エージェントによるステートメントの実行時間
cputime_threshold_id	xs:int	cputime_threshold_id - CPU 時間しきい値 ID
cputime_threshold_value	xs:long	cputime_threshold_value - CPU 時間しきい値

表 125. MON\_GET\_ACTIVITY\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
cputime_threshold_violated	xs:short (1 = はい、0 = いいえ)	cputime_threshold_violated - CPU 時間しきい値の違反
cputimeinsec_threshold_id	xs:int	cputimeinsec_threshold_id - サービス・クラス内の CPU 時間しきい値 ID
cputimeinsec_threshold_value	xs:long	cputimeinsec_threshold_value - サービス・クラス内の CPU 時間しきい値
cputimeinsec_threshold_violated	xs:short (1 = はい、0 = いいえ)	cputimeinsec_threshold_violated - サービス・クラス内の CPU 時間しきい値の違反
database_work_action_set_id	xs:nonNegativeInteger	db_work_action_set_id - データベース作業アクション・セット ID
database_work_class_id	xs:nonNegativeInteger	db_work_class_id - データベース作業クラス ID
datataginsc_threshold_id	xs:int	datataginsc_threshold_id - DATATAGINSC IN しきい値 ID
datataginsc_threshold_value	xs:string(32)	datataginsc_threshold_value - DATATAGINSC IN しきい値
datataginsc_threshold_violated	xs:short (1 = はい、0 = いいえ)	datataginsc_threshold_violated - DATATAGINSC IN しきい値違反
datatagnotinsc_threshold_id	xs:int	datatagnotinsc_threshold_id - DATATAGNOTINSC NOT IN しきい値 ID
datatagnotinsc_threshold_value	xs:string(32)	datatagnotinsc_threshold_value - DATATAGNOTINSC NOT IN しきい値
datatagnotinsc_threshold_violated	xs:short (1 = はい、0 = いいえ)	datatagnotinsc_threshold_violated - DATATAGNOTINSC NOT IN しきい値違反
deadlocks	xs:nonNegativeInteger	deadlocks - デッドロック検出数
diaglog_write_wait_time	xs:nonNegativeInteger	diaglog_write_wait_time - 診断ログ・ファイルの書き込み待機時間
diaglog_writes_total	xs:nonNegativeInteger	diaglog_writes_total - 診断ログ・ファイル書き込みの合計
direct_read_reqs	xs:nonNegativeInteger	direct_read_reqs - 直接読み取り要求
direct_read_time	xs:nonNegativeInteger	direct_read_time - 直接読み取り時間
direct_reads	xs:nonNegativeInteger	direct_reads - データベースからの直接読み取り
direct_write_reqs	xs:nonNegativeInteger	direct_write_reqs - 直接書き込み要求
direct_write_time	xs:nonNegativeInteger	direct_write_time - 直接書き込み時間
direct_writes	xs:nonNegativeInteger	direct_writes - データベースへの直接書き込み
disabled_peds	xs:long	disabled_peds - 無効な partial early distinct の数モニター・エレメント
eff_stmt_text	xs:string	eff_stmt_text - 有効なステートメント・テキスト。ステートメント・コンソレーターによって実行されたリテラル置換の後の、対象ステートメント・テキストの最初の 1024 文字です。ステートメント・コンソレーターが使用可能で、このステートメントがステートメント・コンソレーターによって変更された場合のみ存在します。
effective_isolation	xs:string	effective_isolation - 有効な分離
effective_lock_timeout	xs:integer	effective_lock_timeout - 有効なロック・タイムアウト
effective_query_degree	xs:integer	effective_query_degree - 有効な照会の度合い

表 125. MON\_GET\_ACTIVITY\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
entry_time	xs:dateTime	entry_time - 入力時刻このアクティビティがシステムに到達した時刻。
estimatedsqlcost_threshold_id	xs:int	estimatedsqlcost_threshold_id - 見積もり SQL コストしきい値 ID
estimatedsqlcost_threshold_value	xs:long	estimatedsqlcost_threshold_value - 見積もり SQL コストしきい値
estimatedsqlcost_threshold_violated	xs:short (1 = はい、0 = いいえ)	estimatedsqlcost_threshold_violated - 見積もり SQL コストしきい値の違反
evmon_wait_time	xs:nonNegativeInteger	evmon_wait_time - イベント・モニターの待機時間
evmon_waits_total	xs:nonNegativeInteger	evmon_waits_total - イベント・モニターの待機の合計回数
executable_id	executable_id_type	executable_id - 実行可能 ID
fcm_message_recv_volume	xs:nonNegativeInteger	fcm_message_recv_volume - FCM メッセージ受信ボリューム
fcm_message_recv_wait_time	xs:nonNegativeInteger	fcm_message_recv_wait_time - FCM メッセージの受信待機時間
fcm_message_recvs_total	xs:nonNegativeInteger	fcm_message_recvs_total - FCM メッセージ合計受信数
fcm_message_send_volume	xs:nonNegativeInteger	fcm_message_send_volume - FCM メッセージ送信ボリューム
fcm_message_send_wait_time	xs:nonNegativeInteger	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
fcm_message_sends_total	xs:nonNegativeInteger	fcm_message_sends_total - FCM メッセージ合計送信数
fcm_recv_volume	xs:nonNegativeInteger	fcm_recv_volume - FCM 受信ボリューム
fcm_recv_wait_time	xs:nonNegativeInteger	fcm_recv_wait_time - FCM 受信待機時間
fcm_recvs_total	xs:nonNegativeInteger	fcm_recvs_total - FCM 合計受信数
fcm_send_volume	xs:nonNegativeInteger	fcm_send_volume - FCM 送信ボリューム
fcm_send_wait_time	xs:nonNegativeInteger	fcm_send_wait_time - FCM 送信待ち時間
fcm_sends_total	xs:nonNegativeInteger	fcm_sends_total - FCM 合計送信数
fcm_tq_recv_volume	xs:nonNegativeInteger	fcm_tq_recv_volume - FCM 表キューの受信ボリューム
fcm_tq_recv_wait_time	xs:nonNegativeInteger	fcm_tq_recv_wait_time - FCM 表キューの受信待ち時間
fcm_tq_recvs_total	xs:nonNegativeInteger	fcm_tq_recvs_total - FCM 表キューの合計受信数
fcm_tq_send_volume	xs:nonNegativeInteger	fcm_tq_send_volume - FCM 表キューの送信ボリューム
fcm_tq_send_wait_time	xs:nonNegativeInteger	fcm_tq_send_wait_time - FCM 表キューの送信待ち時間
fcm_tq_sends_total	xs:nonNegativeInteger	fcm_tq_sends_total - FCM 表キューの合計送信数
ida_recv_wait_time	xs:nonNegativeInteger	インデータベース分析処理からデータを受信するために待機した合計時間。
ida_recv_volume	xs:nonNegativeInteger	インデータベース分析処理からデータベース・サーバーが受け取ったデータの合計量。値はバイト数で報告されます。
ida_recvs_total	xs:nonNegativeInteger	インデータベース分析処理からデータを受け取った合計回数。
ida_send_wait_time	xs:nonNegativeInteger	インデータベース分析処理にデータを送信するために待機した合計時間。

表 125. MON\_GET\_ACTIVITY\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
ida_send_volume	xs:nonNegativeInteger	インデータベース分析処理にデータベース・サーバーから送信されたデータの合計量。値はバイト数で報告されます。
ida_sends_total	xs:nonNegativeInteger	インデータベース分析処理にデータが送信された合計回数。
intra_parallel_state	xs:string	intra_parallel_state - パーティション内並列処理の現行状態 モニター・エレメント
invocation_id	xs:nonNegativeInteger	stmt_invocation_id - ステートメント呼び出し ID
last_reference_time	xs:dateTime	last_reference_time - 最終参照時刻。要求がこのアクティビティで発生するたびに、このフィールドは更新されます。
local_start_time	xs:dateTime	local_start_time - ローカル開始時刻。
lock_escals	xs:nonNegativeInteger	lock_escals - ロック・エスカレーション数
lock_timeouts	xs:nonNegativeInteger	lock_timeouts - ロック・タイムアウト数
lock_wait_time	xs:nonNegativeInteger	lock_wait_time - ロック待機中の時間
lock_waits	xs:nonNegativeInteger	lock_waits - ロック待機数
log_buffer_wait_time	xs:nonNegativeInteger	log_buffer_wait_time - ログ・バッファ待ち時間
log_disk_wait_time	xs:nonNegativeInteger	log_disk_wait_time - ログ・ディスク待機時間
log_disk_waits_total	xs:nonNegativeInteger	log_disk_waits_total - ログ・ディスク待機の合計
member	xs:nonNegativeInteger	member - データベース・メンバー
nesting_level	xs:nonNegativeInteger	stmt_nest_level - ステートメント・ネスティング・レベル
num_log_buffer_full	xs:nonNegativeInteger	num_log_buffer_full - フル・ログ・バッファの回数
num_lw_thresh_exceeded	xs:nonNegativeInteger	num_lw_thresh_exceeded - しきい値を超えた回数
num_routines	xs:int	num_routines - ルーチンの数
package_name	xs:string (128)	package_name - パッケージ名
package_schema	xs:string (128)	package_schema - パッケージ・スキーマ
package_version_id	xs:string (128)	package_version_id - パッケージ・バージョン
parent_activity_id	xs:nonNegativeInteger	parent_activity_id - 親アクティビティ ID
parent_uow_id	xs:nonNegativeInteger	parent_uow_id - 親作業単位 ID
pool_data_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ・モニター・エレメント
pool_data_l_reads	xs:nonNegativeInteger	pool_data_l_reads - バッファ・プール・データの論理読み取り
pool_data_p_reads	xs:nonNegativeInteger	pool_data_p_reads - バッファ・プール・データの物理読み取り
pool_data_writes	xs:nonNegativeInteger	pool_data_writes - バッファ・プールへのデータの書き込み
pool_failed_async_data_reqs	xs:nonNegativeInteger	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求モニター・エレメント
pool_failed_async_index_reqs	xs:nonNegativeInteger	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求モニター・エレメント

表 125. MON\_GET\_ACTIVITY\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
pool_failed_async_other_reqs	xs:nonNegativeInteger	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求モニター・エレメント
pool_failed_async_temp_data_reqs	xs:nonNegativeInteger	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
pool_failed_async_temp_index_reqs	xs:nonNegativeInteger	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
pool_failed_async_temp_xda_reqs	xs:nonNegativeInteger	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求モニター・エレメント
pool_failed_async_xda_reqs	xs:nonNegativeInteger	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求モニター・エレメント
pool_index_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ・モニター・エレメント
pool_index_l_reads	xs:nonNegativeInteger	pool_index_l_reads - バッファ・プール索引の論理読み取り
pool_index_p_reads	xs:nonNegativeInteger	pool_index_p_reads - バッファ・プール索引の物理読み取り
pool_index_writes	xs:nonNegativeInteger	pool_index_writes - バッファ・プール索引の書き込み
pool_queued_async_data_pages	xs:nonNegativeInteger	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_data_reqs	xs:nonNegativeInteger	pool_queued_async_data_reqs - データ・プリフェッチ要求モニター・エレメント
pool_queued_async_index_pages	xs:nonNegativeInteger	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_index_reqs	xs:nonNegativeInteger	pool_queued_async_index_reqs - 索引プリフェッチ要求モニター・エレメント
pool_queued_async_other_reqs	xs:nonNegativeInteger	pool_queued_async_other_reqs - 非プリフェッチの要求モニター・エレメント
pool_queued_async_temp_data_pages	xs:nonNegativeInteger	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_data_reqs	xs:nonNegativeInteger	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_index_pages	xs:nonNegativeInteger	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_index_reqs	xs:nonNegativeInteger	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
pool_queued_async_temp_xda_pages	xs:nonNegativeInteger	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求モニター・エレメント



表 125. MON\_GET\_ACTIVITY\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
pool_queued_async_temp_xda_reqs	xs:nonNegativeInteger	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求モニター・エレメント
pool_queued_async_xda_pages	xs:nonNegativeInteger	pool_queued_async_xda_pages - XDA ページ・プリフェッチ要求
pool_queued_async_xda_reqs	xs:nonNegativeInteger	pool_queued_async_xda_reqs - XDA プリフェッチ要求モニター・エレメント
pool_read_time	xs:nonNegativeInteger	pool_read_time - バッファ・プール物理読み取り時間の合計
pool_temp_data_l_reads	xs:nonNegativeInteger	pool_temp_data_l_reads - バッファ・プールの時データの論理読み取り
pool_temp_data_p_reads	xs:nonNegativeInteger	pool_temp_data_p_reads - バッファ・プールの時データの物理読み取り
pool_temp_index_l_reads	xs:nonNegativeInteger	pool_temp_index_l_reads - バッファ・プールの時索引の論理読み取り
pool_temp_index_p_reads	xs:nonNegativeInteger	pool_temp_index_p_reads - バッファ・プールの時索引の物理読み取り
pool_temp_xda_l_reads	xs:nonNegativeInteger	pool_temp_xda_l_reads - バッファ・プールの時 XDA データの論理読み取り
pool_temp_xda_p_reads	xs:nonNegativeInteger	pool_temp_xda_p_reads - バッファ・プールの時 XDA データの物理読み取り
pool_write_time	xs:nonNegativeInteger	pool_write_time - バッファ・プール物理書き込み時間の合計
pool_xda_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ・モニター・エレメント
pool_xda_gbp_invalid_pages	xs:nonNegativeInteger	pool_xda_gbp_invalid_pages - グループ・バッファ・プールの無効な XDA データ・ページ
pool_xda_gbp_l_reads	xs:nonNegativeInteger	pool_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの論理読み取りの要求
pool_xda_gbp_p_reads	xs:nonNegativeInteger	pool_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの物理読み取りの要求
pool_xda_l_reads	xs:nonNegativeInteger	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
pool_xda_lbp_pages_found	xs:nonNegativeInteger	pool_xda_lbp_pages_found - 検出されたローカル・バッファ・プールの XDA データ・ページ
pool_xda_p_reads	xs:nonNegativeInteger	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
pool_xda_writes	xs:nonNegativeInteger	pool_xda_writes - バッファ・プール XDA データの書き込み
post_shrthreshold_sorts	xs:nonNegativeInteger	post_shrthreshold_sorts - ポスト共有しきい値ソート
post_threshold_peas	xs:long	post_threshold_peas - partial early aggregation のしきい値モニター・エレメント

表 125. MON\_GET\_ACTIVITY\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
post_threshold_peds	xs:long	post_threshold_peds - partial early distinct のしきい値モニター・エレメント
post_threshold_sorts	xs:nonNegativeInteger	post_threshold_sorts - ポストしきい値ソート
prefetch_wait_time	xs:nonNegativeInteger	prefetch_wait_time - プリフェッチ待ち時間
prefetch_waits	xs:nonNegativeInteger	prefetch_waits - プリフェッチャーの待機カウント・モニター・エレメント
query_actual_degree	xs:int	query_actual_degree - パーティション内並列処理の実際の実行時の多重度のモニター・エレメント
query_cost_estimate	xs:integer	query_cost_estimate - 照会コストの見積もり
query_data_tag_list	xs:string(32)	query_data_tag_list - 照会データ・タグ・リスト
routine_id	xs:nonNegativeInteger	routine_id - ルーチン ID
rows_modified	xs:nonNegativeInteger	rows_modified - 変更された行数
rows_read	xs:nonNegativeInteger	rows_read - 読み取り行数
rows_returned	xs:nonNegativeInteger	rows_returned - 戻り行数
section_number	xs:integer	section_number - セクション番号
service_class_id	xs:integer	service_class_id - サービス・クラス
service_class_work_action_set_id	xs:nonNegativeInteger	sc_work_action_set_id - サービス・クラス作業アクション・セット ID
service_class_work_class_id	xs:nonNegativeInteger	sc_work_class_id - サービス・クラス作業クラス ID
sort_overflows	xs:nonNegativeInteger	sort_overflows - ソート・オーバーフロー
sqlrowsread_threshold_id	xs:int	sqlrowsread_threshold_ID - SQL 読み取り行数しきい値 ID
sqlrowsread_threshold_value	xs:long	sqlrowsread_threshold_value - SQL 読み取り行数しきい値
sqlrowsread_threshold_violated	xs:short (1 = はい, 0 = いいえ)	sqlrowsread_threshold_violated - SQL SQL 読み取り行数しきい値の違反
sqlrowsreadinsc_threshold_id	xs:int	sqlrowsreadinsc_threshold_id - サービス・クラス内の SQL 読み取り行数しきい値 ID
sqlrowsreadinsc_threshold_value	xs:long	sqlrowsreadinsc_threshold_value - サービス・クラス内の SQL 読み取り行数しきい値
sqlrowsreadinsc_threshold_violated	xs:short (1 = はい, 0 = いいえ)	sqlrowsreadinsc_threshold_violated - サービス・クラス内の SQL 読み取り行数しきい値の違反
sqlrowsreturned_threshold_id	xs:int	sqlrowsreturned_threshold_id - 戻される SQL 読み取り行数しきい値 ID
sqlrowsreturned_threshold_value	xs:long	sqlrowsreturned_threshold_value - 戻される SQL 読み取り行数しきい値
sqlrowsreturned_threshold_violated	xs:short (1 = はい, 0 = いいえ)	sqlrowsreturned_threshold_violated - 戻される SQL 読み取り行数しきい値の違反
sqltempespace_threshold_id	xs:int	sqltempespace_threshold_id - SQL 一時スペースしきい値 ID
sqltempespace_threshold_value	xs:long	sqltempespace_threshold_value - SQL 一時スペースしきい値
sqltempespace_threshold_violated	xs:short (1 = はい, 0 = いいえ)	sqltempespace_threshold_violated - SQL 一時スペースしきい値の違反
stmt_exec_time	xs:nonNegativeInteger	stmt_exec_time - ステートメントの実行時間

表 125. MON\_GET\_ACTIVITY\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
stmt_pkg_cache_id	xs:nonNegativeInteger	stmt_pkgs_id - ステートメント・パッケージ・キャッシュ ID
stmt_text	xs:string	stmt_text - SQL 動的ステートメント・テキスト。アクティビティが動的 SQL であるか、またはステートメント・テキストが使用可能になっている静的 SQL である場合、このフィールドにはそのステートメント・テキストの最初の 1024 文字が入っています。そうでない場合、空ストリングが入っています。
stmtno	xs:int	stmtno - ステートメント番号モニター・エレメント
thresh_violations	xs:nonNegativeInteger	thresh_violations - しきい値違反の回数
total_act_time	xs:nonNegativeInteger	total_act_time - 合計アクティビティ時間
total_act_wait_time	xs:nonNegativeInteger	total_act_wait_time - 合計アクティビティ待機時間
total_app_section_executions	xs:nonNegativeInteger	total_app_section_executions - セクション実行の合計回数
total_cpu_time	xs:nonNegativeInteger	total_cpu_time - 合計 CPU 時間
total_disp_run_queue_time	xs:long	total_disp_run_queue_time - ディスパッチャーの合計実行キュー時間 : モニター・エレメント (マイクロ秒単位)
total_extended_latch_wait_time	xs:nonNegativeInteger	total_extended_latch_wait_time - 拡張ラッチの合計待機時間のモニター・エレメント
total_extended_latch_waits	xs:nonNegativeInteger	total_extended_latch_waits - 拡張ラッチの合計待機回数のモニター・エレメント
total_peas	xs:long	total_peas - partial early aggregation の合計モニター・エレメント
total_peds	xs:long	total_peds - partial early distinct の合計モニター・エレメント
total_routine_invocations	xs:nonNegativeInteger	total_routine_invocations - ルーチン呼び出しの合計回数
total_routine_non_sect_proc_time	xs:nonNegativeInteger	total_routine_non_sect_proc_time - 非セクション処理時間
total_routine_non_sect_time	xs:nonNegativeInteger	total_routine_non_sect_time - 非セクション・ルーチン実行時間
total_routine_time	xs:nonNegativeInteger	total_routine_time - ルーチン時間の合計
total_routine_user_code_proc_time	xs:nonNegativeInteger	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
total_routine_user_code_time	xs:nonNegativeInteger	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
total_section_proc_time	xs:nonNegativeInteger	total_section_proc_time - セクション処理時間の合計
total_section_sort_proc_time	xs:nonNegativeInteger	total_section_sort_proc_time - セクション・ソート処理時間合計
total_section_sort_time	xs:nonNegativeInteger	total_section_sort_time - セクション・ソート時間合計
total_section_sorts	xs:nonNegativeInteger	total_section_sorts - セクション・ソート合計
total_section_time	xs:nonNegativeInteger	total_section_time - セクション時間の合計
total_sorts	xs:nonNegativeInteger	total_sorts - ソート合計
tq_sort_heap_rejections	xs:long	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否モニター・エレメント

表 125. MON\_GET\_ACTIVITY\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
tq_sort_heap_requests	xs:long	tq_sort_heap_requests - 表キューのソート・ヒープの要求モニター・エレメント
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
uow_id	xs:nonNegativeInteger	uow_id - 作業単位 ID
utility_id	xs:nonNegativeInteger	utility_id - ユーティリティ ID
wl_work_action_set_id	xs:nonNegativeInteger	wl_work_action_set_id - ワークロード作業アクション・セット ID モニター・エレメント
wl_work_class_id	xs:nonNegativeInteger	wl_work_class_id - ワークロード作業クラス ID
wlm_queue_assignments_total	xs:nonNegativeInteger	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
wlm_queue_time_total	xs:nonNegativeInteger	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間

## MON\_GET\_APPL\_LOCKWAIT - アプリケーションが待機しているロックについての情報の取得

MON\_GET\_APPL\_LOCKWAIT 表関数は、(現在のデータベースに接続されている) 各アプリケーションのエージェントが取得を待機しているすべてのロックについての情報を戻します。

ロックについての情報を取得するには MON\_GET\_APPL\_LOCKWAIT、MON\_FORMAT\_LOCK\_NAME、および MON\_GET\_LOCKS 表関数を使用してください。SNAPLOCKWAIT 管理ビューと SNAP\_GET\_LOCKWAIT 表関数、および SNAPLOCK 管理ビューと SNAP\_GET\_LOCK 表関数はバージョン 9.7 フィックスパック 1 で非推奨になりました。

►►MON\_GET\_APPL\_LOCKWAIT(—application\_handle—,—member—)◄◄

スキーマは SYSPROC です。

### 表関数パラメーター

#### application\_handle

現在接続されているのと同じデータベース内の有効なアプリケーション・ハンドルを指定する、オプションのタイプ BIGINT の入力パラメーター。この引数が NULL である場合、ロックの取得を現在待機しているすべてのアプリケーションに関するロックが検索されます。

#### member

現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力パラメーター。現在のデータベース・メンバーの場合は -1、すべてのアクティブ・メンバーの場合は -2 を指定します。NULL 値を指定すると、-1 が設定されます。

## 許可

以下のいずれかの権限または特権が必要です。

- SYSADM 権限
- SYSMON 権限
- MON\_GET\_APPL\_LOCKWAIT 表関数に対する EXECUTE 特権。

## デフォルトの PUBLIC 特権

なし

## 例

この例のシナリオでは MON\_GET\_APPL\_LOCKWAIT 表関数を使用して、セッション許可 ID USER1 に関するハングしているアプリケーションを調べます。

1. 次のように WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES 表関数を使用して、SESSION\_USER 値が USER1 であるすべての接続のアプリケーション・ハンドルを調べます。

```
SELECT COORD_PARTITION_NUM,  
       APPLICATION_HANDLE  
FROM TABLE(WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES('','',-2))  
WHERE SESSION_USER = 'USER1'
```

この照会は、以下の出力を戻します。

COORD_PARTITION_NUM	APPLICATION_HANDLE
2	131130

1 record(s) selected.

2. 次のように WLM\_GET\_SERVICE\_CLASS\_AGENTS 表関数を使用して、すべてのデータベース・パーティションについて、この接続のための処理を行っているすべてのエージェントに関する現行情報を取得します。

```
SELECT SUBSTR(CHAR(DBPARTITIONNUM),1,3) AS DBPART,  
       SUBSTR(CHAR(APPLICATION_HANDLE),1,7) AS APP_ID,  
       SUBSTR(CHAR(WORKLOAD_OCCURRENCE_ID),1,7) AS WLO_ID,  
       SUBSTR(CHAR(AGENT_TID),1,7) AS AGENT_ID,  
       SUBSTR(CHAR(AGENT_TYPE),1,12) AS AGENT_TYPE,  
       SUBSTR(AGENT_STATE,1, 8) AS STATE,  
       SUBSTR(EVENT_TYPE,1, 8) AS EV_TYPE,  
       SUBSTR(EVENT_OBJECT,1,12) AS EV_OBJECT  
FROM TABLE(WLM_GET_SERVICE_CLASS_AGENTS('','',131130,-2))  
ORDER BY AGENT_TYPE, DBPART
```

この照会は、以下の出力を戻します。

DBPART	APP_ID	WLO_ID	AGENT_ID	AGENT_TYPE	STATE	EV_TYPE	EV_OBJECT
2	131130	1	3110	COORDINATOR	ACTIVE	WAIT	REQUEST
0	131130	1	7054	PDBSUBAGENT	ACTIVE	ACQUIRE	LOCK
1	131130	1	5709	PDBSUBAGENT	ACTIVE	ACQUIRE	LOCK
2	131130	1	5960	PDBSUBAGENT	ACTIVE	ACQUIRE	LOCK

4 record(s) selected.

タイプ LOCK のイベント・オブジェクトに対するタイプ ACQUIRE のイベントはロック待機シナリオを示しているため、待機の対象となっているオブジェクトは何か、および何がそのロックを保持しているかを調べる必要があります。

- アプリケーションが待機しているすべてのロックを判別するには、アプリケーション・ハンドル 131130 およびメンバー -2 を入力パラメーターとして指定して MON\_GET\_APPL\_LOCKWAIT 表関数を呼び出します。

```
SELECT lock_name,
       hld_member AS member,
       hld_agent_tid as TID,
       hld_application_handle AS HLD_APP FROM
TABLE (MON_GET_APPL_LOCKWAIT(131130, -2))
```

この照会は、以下の出力を戻します。

LOCK_NAME	MEMBER	TID	HLD_APP
00030005000000000280000452	0	1234	65564
00030005000000000280000452	1	5478	65564
00030005000000000280000452	2	4678	65564

3 record(s) selected.

- WLM\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES 表関数を呼び出して、ロックを保持しているアプリケーションについてさらに調べます (このアプリケーションのアプリケーション・ハンドルは 65564)。

```
SELECT SYSTEM_AUTH_ID,
       APPLICATION_NAME AS APP_NAME,
       WORKLOAD_NAME AS WORKLOAD,
       WORKLOAD_OCCURRENCE_STATE AS WL_STATE
FROM TABLE(WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES('','',-2))
WHERE APPLICATION_HANDLE = 65564
```

この照会は、以下の出力を戻します。

SYSTEM_AUTH_ID	APP_NAME	WORKLOAD	WL_STATE
ZURBIE	db2bp	SYSDEFAULTUSERWORKLOAD	UOWWAIT

1 record(s) selected

## 戻される情報

戻される列は、次のような領域の情報を提供します。

- 以下の列は、アプリケーションが取得を現在待機しているロックについての詳細を表します。

LOCK\_WAIT\_START\_TIME、 LOCK\_NAME、 LOCK\_OBJECT\_TYPE、  
 LOCK\_MODE、 LOCK\_CURRENT\_MODE、 LOCK\_MODE\_REQUESTED、  
 LOCK\_STATUS、 LOCK\_ESCALATION、 LOCK\_ATTRIBUTES、  
 LOCK\_RRIID、 LOCK\_COUNT、 TBSP\_ID、 TAB\_FILE\_ID、  
 SUBSECTION\_NUMBER。

- 以下の列は、このロックの取得を待機しているアプリケーションについての詳細を表します。

REQ\_APPLICATION\_HANDLE、 REQ\_AGENT\_TID、 REQ\_MEMBER、  
 REQ\_EXECUTABLE\_ID

- 以下の列は、ロックを現在保持しているアプリケーションについての詳細を表します。

HLD\_APPLICATION\_HANDLE、HLD\_MEMBER、ADDITIONAL\_DETAILS

表 126. MON\_GET\_APPL\_LOCKWAIT 表関数によって戻される情報

列名	データ・タイプ	説明またはモニター・エレメント
LOCK_WAIT_START_TIME	TIMESTAMP	lock_wait_start_time - ロック待機開始タイム・スタンプ
LOCK_NAME	VARCHAR(32)	lock_name - ロック名  ロックについての詳細情報を得るために、 MON_FORMAT_LOCK_NAME 表関数を使用して内部名をフォーマット設定できます。例えば、表ロックの場合には、ロックが参照している表および表スペースを検出できます。
LOCK_OBJECT_TYPE_ID	CHAR(1) FOR BIT DATA	将来の利用のために予約済み
LOCK_OBJECT_TYPE	VARCHAR(32)	lock_object_type - 待機中のロック対象タイプ  可能な値については、 『lock_object_type - 待機中のロック対象タイプ』モニター・エレメントを参照してください
LOCK_MODE	VARCHAR(3)	lock_mode - ロック・モード  このロックを保持しているアプリケーションが見つからない場合、NULL 値が戻されます。  グローバル・ロック待機の場合、この値は NULL です。
LOCK_CURRENT_MODE	VARCHAR(3)	lock_current_mode - 移行前の元のロック・モード  移行が行われなかった場合は、NULL 値が戻されます。
LOCK_MODE_REQUESTED	VARCHAR(3)	lock_mode_requested - 要求されているロック・モード
LOCK_STATUS	CHAR(1)	lock_status - ロック状況
LOCK_ESCALATION	CHAR(1)	lock_escalation - ロック・エスカレーション
LOCK_ATTRIBUTES	CHAR(16)	lock_attributes - ロック属性
LOCK_RRIID	BIGINT	lock_count - ロック・カウント・モニター・エレメント
LOCK_COUNT	BIGINT	lock_count - ロック・カウント・モニター・エレメント

表 126. MON\_GET\_APPL\_LOCKWAIT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
TAB_FILE_ID	BIGINT	table_file_id - 表ファイル ID
SUBSECTION_NUMBER	BIGINT	ss_number - サブセクション番号  サブセクション番号を入手できない場合には NULL 値が戻されます。
REQ_APPLICATION_HANDLE	BIGINT	req_application_handle - 要求元のアプリケーション・ハンドル
REQ_AGENT_TID	BIGINT	req_agent_tid - 要求元のエージェント TID
REQ_MEMBER	SMALLINT	req_member - 要求元のメンバー
REQ_EXECUTABLE_ID	VARCHAR (32) FOR BIT DATA	req_executable_id - 要求元の実行可能 ID
HLD_APPLICATION_HANDLE	BIGINT	hld_application_handle - 保持しているアプリケーション・ハンドル  このロックを保持しているアプリケーションが不明または見つからない場合には、NULL 値が戻されます。  グローバル・ロック待機の場合、この値は NULL です。
HLD_MEMBER	SMALLINT	hld_member - 保持しているメンバー

## MON\_GET\_APPLICATION\_HANDLE - 接続アプリケーション・ハンドルの取得

MON\_GET\_APPLICATION\_HANDLE スカラー関数は、呼び出しアプリケーションのアプリケーション・ハンドルを返します。結果のデータ・タイプは BIGINT です。

### 構文

▶▶—MON\_GET\_APPLICATION\_HANDLE—(—)—▶▶

スキーマは SYSPROC です。

### 許可

以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限



## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

MON\_GET\_APPLICATION\_HANDLE スカラー関数を使用して現行セッションのアプリケーション・ハンドルをモニター関数に受け渡すと、アプリケーション・ハンドルによるフィルタリングによって、そのセッションのモニター情報のみを取得できます。例:

```
select application_handle, application_name, application_id, member, rows_read
from table(sysproc.mon_get_connection(sysproc.mon_get_application_handle(), -1))
as conn
```

以下はこの照会の出力例です。

APPLICATION_HANDLE	APPLICATION_NAME	APPLICATION_ID	MEMBER
644	db2bp	*LOCAL.amurchis.110831180720	0

ROWS_READ
0

1 record(s) selected.

## MON\_GET\_APPLICATION\_ID - 接続アプリケーション ID の取得

MON\_GET\_APPLICATION\_ID スカラー関数は、呼び出しアプリケーションのアプリケーション ID を返します。この結果のデータ・タイプは VARCHAR(128) です。

この関数から戻される値は 100 年のインターバル内で固有であり、関数を呼び出す前に確立された接続の期間中だけ有効です。

▶▶MON\_GET\_APPLICATION\_ID(—)◀◀

スキーマは SYSPROC です。

### 許可

以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
values sysproc.mon_get_application_id()
```

以下はこの照会の出力例です。

```
1
-----
*LOCAL.amurchis.110831180720
```

## MON\_GET\_AUTO\_MAINT\_QUEUE 表関数 - 自動保守ジョブに関する情報の取得

MON\_GET\_AUTO\_MAINT\_QUEUE 表関数は、オートノミック・コンピューティング・デーモン (db2acd) による実行のキューに現在入っているすべての自動保守ジョブ (自動保守キューでのジョブのサブミットを行わないリアルタイム統計を除く) に関する情報を戻します。

## 構文

```
▶▶—MON_GET_AUTO_MAINT_QUEUE—( )—————▶▶
```

スキーマは SYSPROC です。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

各パーティションの自動保守キューにある現在のジョブを表示します。

```
SELECT MEMBER
       QUEUE_POSITION,
       JOB_STATUS,
       JOB_TYPE,
       VARCHAR(DB_NAME, 10) AS DB_NAME,
       OBJECT_TYPE,
       VARCHAR(OBJECT_SCHEMA, 10) AS OBJECT_SCHEMA
       VARCHAR(OBJECT_NAME, 10) AS OBJECT_NAME
FROM TABLE(MON_GET_AUTO_MAINT_QUEUE()) AS T
ORDER BY MEMBER, QUEUE_POSITION ASC
```

以下はこの照会の出力例です。

```
MEMBER QUEUE_POSITION JOB_STATUS JOB_TYPE DB_NAME OBJECT_TYPE OBJECT_SCHEMA
-----
0          1 EXECUTING  RUNSTATS SAMPLE TABLE TEST
0          2 QUEUED    REORG   SAMPLE TABLE TEST
0          3 QUEUED    REORG   SAMPLE TABLE TEST
```

```
OBJECT_NAME
-----
EMPLOYEE
T1
BLAH
```

3 record(s) selected.

自動保守キューにある現在のジョブを表示します。

```
SELECT JOB_STATUS,
       JOB_TYPE,
       OBJECT_TYPE,
       VARCHAR(OBJECT_NAME, 10) AS OBJECT_NAME,
       VARCHAR(JOB_DETAILS,60) AS JOB_DETAILS
FROM TABLE(MON_GET_AUTO_MAINT_QUEUE()) AS T
ORDER BY MEMBER, QUEUE_POSITION ASC
```

以下はこの照会の出力例です。

```
JOB_STATUS JOB_TYPE    OBJECT_TYPE OBJECT_NAME
-----
EXECUTING  REORG             TABLE      TP3
```

```
JOB_DETAILS
-----
REORG INDEXES ALLOW WRITE CLEANUP ALL; RECLAIM EXTENTS
```

1 record(s) selected.

## 使用上の注意

MON\_GET\_AUTO\_MAINT\_QUEUE 表関数は、すべてのメンバーに関して、キューに入っている自動保守ジョブと実行中の自動保守ジョブのすべてについて情報を戻します。メンバーごとに別個の自動保守キューが存在することに注意してください。

**注:** 自動保守キュー内のジョブは、まず最も早い開始時刻 (ジョブを実行できる次の保守ウィンドウの始まり)、次に優先順位 (最も早い開始時刻が同じ複数の項目間で)、そしてキュー項目時刻 (最も早い開始時刻と優先順位がどちらも同じ複数の項目間で) の順に並べられます。

MON\_GET\_AUTO\_MAINT\_QUEUE から戻される情報は、HEALTH\_DB\_HIC インターフェースからの情報を補います。HEALTH\_DB\_HIC は、自動保守が適用される各表の自動保守状況に加えて、状況が最後に更新された時刻 (保守が必要かどうかを調べるために表が最後に検査された時刻) を示します。

MON\_GET\_AUTO\_MAINT\_QUEUE インターフェースは、状態がいつ AUTOMATED になるか (保守ジョブが自動保守キューのどこにあるかの詳細)、およびキューの中でジョブの前に他にどんなジョブがあるかについてのドリルダウンを提供します。

MON\_GET\_AUTO\_MAINT\_QUEUE 表関数は、自動保守が使用可能になっていない場合、自動保守ジョブをまったく報告しません。

## 戻される情報

表 127. MON\_GET\_AUTO\_MAINT\_QUEUE で戻される情報

列名	データ・タイプ	説明
DB_NAME	VARCHAR(128)	db_name - データベース名 : モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
OBJECT_TYPE	VARCHAR(8)	objtype - オブジェクト・タイプ・モニター・エレメント
OBJECT_SCHEMA	VARCHAR(128)	object_schema - オブジェクト・スキーマ・モニター・エレメント
OBJECT_NAME	VARCHAR(128)	object_name - オブジェクト名モニター・エレメント
JOB_TYPE	VARCHAR(12)	自動保守ジョブのタイプ。以下のいずれか  RUNSTATS  REORG  BACKUP  STATSPROFILE
JOB_DETAILS	VARCHAR(256)	保守ジョブについての詳細 (タイプが RUNSTATS または REORG の場合)。RUNSTATS の場合は、RUNSTATS ジョブが完全 RUNSTATS を行うのか、それともサンプリングのみを行うのかを示します。REORG の場合は、REORG ユーティリティーの動作を変更するために適用されるキーワード (例えば INDEXES や CLEANUP など) をリストします。索引 REORG の場合、キーワードとして CLEANUP と RECLAIM EXTENTS の両方がセミコロンで区切られて指定されていれば、索引の REORG CLEANUP の後に索引 RECLAIM EXTENTS の評価が行われ、実行される可能性もあります (RECLAIM EXTENTS が実行されるのは、再利用可能スペースを reclaimExtentsSizeForIndexObjects 設定値と比較して評価し、再利用が必要とされる場合です)。
JOB_STATUS	VARCHAR(10)	現在のジョブの状況。以下のいずれか  QUEUED  EXECUTING
JOB_PRIORITY	INTEGER	キュー内のジョブの優先順位を示す数値。優先順位が重要になるのは、複数のジョブ間で EARLIEST_START_TIME の値が同じ場合のみです。
MAINT_WINDOW_TYPE	VARCHAR(8)	ジョブに使用される保守ウィンドウのタイプを示します。以下のいずれか  ONLINE  OFFLINE
QUEUE_POSITION	INTEGER	自動保守キュー内のジョブの位置を示します。
QUEUE_ENTRY_TIME	TIMESTAMP	自動保守キューにジョブが追加された時刻。

表 127. MON\_GET\_AUTO\_MAINT\_QUEUE で戻される情報 (続き)

列名	データ・タイプ	説明
EXECUTION_START_TIME	TIMESTAMP	状況が EXECUTING の場合、ジョブが実行を開始した時刻。その他の場合は NULL。
EARLIEST_START_TIME	TIMESTAMP	ジョブを実行できる次の保守ウィンドウの開始時刻。

## MON\_GET\_AUTO\_RUNSTATS\_QUEUE 表関数 - 評価のためにキューに入っているオブジェクトに関する情報の取得

MON\_GET\_AUTO\_RUNSTATS\_QUEUE 表関数は、現在接続されているデータベースで自動統計収集による評価のためにキューに現在入っているすべてのオブジェクトについての情報を戻します。

### 構文

```
▶▶ MON_GET_AUTO_RUNSTATS_QUEUE—() —————▶▶
```

スキーマは SYSPROC です。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

現在接続されているデータベースにおいて現時点で自動統計収集による評価段階にあるすべてのオブジェクトを表示します。

```
SELECT QUEUE_POSITION,
       OBJECT_TYPE,
       OBJECT_STATUS,
       VARCHAR(OBJECT_SCHEMA, 10) AS OBJECT_SCHEMA
       VARCHAR(OBJECT_NAME, 10) AS OBJECT_NAME
FROM TABLE(MON_GET_AUTO_RUNSTATS_QUEUE()) AS T
ORDER BY QUEUE_POSITION ASC
```

以下はこの照会の出力例です。

```
QUEUE_POSITION OBJECT_TYPE OBJECT_STATUS      OBJECT_SCHEMA OBJECT_NAME
-----
          1 TABLE      JOB_SUBMITTED      TEST      EMPLOYEE
          2 TABLE      EVALUATION_PENDING TEST      T1
          3 TABLE      EVALUATION_PENDING TEST      BLAH
```

3 record(s) selected.

## 使用上の注意

MON\_GET\_AUTO\_RUNSTATS\_QUEUE 表関数は、現在接続されているデータベースで自動統計収集による評価のためにキューに入っているすべてのオブジェクトについての情報を戻します。評価用のキューに入っているオブジェクトは、その後、自動統計収集によって検査され、統計更新が必要かどうか判別されます。オブジェクトが統計更新を必要としていることが分かると、自動保守スケジューラーにジョブがサブミットされ、そのジョブの実行が完了するまで評価が一時的にブロックされます。統計収集ジョブの進行状況をモニターするには、

MON\_GET\_AUTO\_MAINT\_QUEUE インターフェースを使用します。自動統計収集によって評価されたオブジェクトが評価キューから除去されると、次の自動統計収集評価インターバルになるまで、そのオブジェクトは再びキューには追加されません。オブジェクトごとの評価履歴については、HEALTH\_DB\_HIC インターフェースを参照してください。

データベースごとに別々の自動統計収集評価キューがあります。

MON\_GET\_AUTO\_RUNSTATS\_QUEUE は、現在接続されているデータベースの評価キューを報告します。

現時点で、自動統計収集による評価のためにキューに入っているオブジェクトがなければ、MON\_GET\_AUTO\_RUNSTATS\_QUEUE インターフェースによって報告されるリストが空になることがあります。例えば、現在の評価インターバルで評価用のキューに入っていたすべての表が既に自動統計収集により処理された場合は、次の評価インターバルになるまで、評価用のキューに入っているオブジェクトのリストが空になります。自動統計収集の評価インターバルは、ほぼ 2 時間おきに発生します。

自動統計収集が使用可能になっていない場合には、

MON\_GET\_AUTO\_RUNSTATS\_QUEUE インターフェースによって報告されるリストも空になります。

## 戻される情報

表 128. MON\_GET\_AUTO\_RUNSTATS\_QUEUE で戻される情報

列名	データ・タイプ	説明
OBJECT_TYPE	VARCHAR(8)	objtype - オブジェクト・タイプ・モニター・エレメント
OBJECT_SCHEMA	VARCHAR(128)	object_schema - オブジェクト・スキーマ・モニター・エレメント
OBJECT_NAME	VARCHAR(128)	object_name - オブジェクト名モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

表 128. MON\_GET\_AUTO\_RUNSTATS\_QUEUE で戻される情報 (続き)

列名	データ・タイプ	説明
OBJECT_STATUS	VARCHAR(18)	キューに入っているオブジェクトの状況。以下のいずれか  EVALUATION_PENDING - 自動統計収集は、オブジェクトを調べて統計更新が必要かどうかを判別する必要があります。  JOB_SUBMITTED - 自動統計収集は、オブジェクトの統計更新が必要であると判別したので、自動保守スケジューラーにジョブをサブミットしました。自動統計収集はそのジョブが完了するのを待機しています。
JOB_SUBMIT_TIME	TIMESTAMP	状況が JOB_SUBMITTED の場合、自動統計収集ジョブが自動保守スケジューラーにサブミットされた時刻。その他の場合は NULL。
QUEUE_POSITION	INTEGER	自動統計収集評価キュー内のジョブの位置を示します。
QUEUE_ENTRY_TIME	TIMESTAMP	ジョブが自動統計収集評価キューに追加された時刻。

## MON\_GET\_BUFFERPOOL 表関数 - バッファ・プール・メトリックの取得

MON\_GET\_BUFFERPOOL 表関数は、1 つ以上のバッファ・プールのモニター・メトリックを戻します。

### 構文

►►—MON\_GET\_BUFFERPOOL—(—bp\_name—, —member—)—————►►

スキーマは SYSPROC です。

### 表関数パラメーター

#### bp\_name

この関数を呼び出すときに現在接続されているデータベース内の有効なバッファ・プール名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのバッファ・プールについてメトリックが取得されます。

#### member

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限

- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

バッファース・プール・ヒット率を計算します。

```
WITH BPMETRICS AS (
  SELECT bp_name,
         pool_data_l_reads + pool_temp_data_l_reads +
         pool_index_l_reads + pool_temp_index_l_reads +
         pool_xda_l_reads + pool_temp_xda_l_reads as logical_reads,
         pool_data_p_reads + pool_temp_data_p_reads +
         pool_index_p_reads + pool_temp_index_p_reads +
         pool_xda_p_reads + pool_temp_xda_p_reads as physical_reads,
         member
  FROM TABLE(MON_GET_BUFFERPOOL(' ', -2)) AS METRICS)
SELECT
  VARCHAR(bp_name, 20) AS bp_name,
  logical_reads,
  physical_reads,
  CASE WHEN logical_reads > 0
        THEN DEC((1 - (FLOAT(physical_reads) / FLOAT(logical_reads))) * 100, 5, 2)
        ELSE NULL
  END AS HIT_RATIO,
  member
FROM BPMETRICS;
```

以下はこの照会の出力例です。

BP_NAME	LOGICAL_READS	PHYSICAL_READS	HIT_RATIO	MEMBER
IBMDEFAULTBP	619	385	37.80	0
IBMSYSTEMBP4K	0	0	-	0
IBMSYSTEMBP8K	0	0	-	0
IBMSYSTEMBP16K	0	0	-	0
IBMSYSTEMBP32K	0	0	-	0

5 record(s) selected.

照会の出力 (続き)。

```
... HIT_RATIO MEMBER
... -----
...      37.80      0
...      -        0
...      -        0
...      -        0
...      -        0
```

## 使用上の注意

MON\_GET\_BUFFERPOOL 表関数は、データベース・バッファース・プールごとおよびデータベース・メンバーごとに 1 行のデータを戻します。データベース・メンバー全体からの集約は実行されません。ただし、集約は例に示されるように SQL 照会を使用して実行できます。

この関数によって収集されるメトリックは、mon\_obj\_metrics 構成パラメーターを使用してデータベース・レベルで制御されます。デフォルトでは、メトリック収集は有効になります。



## 戻される情報

表 129. MON\_GET\_BUFFERPOOL について戻される情報

列名	データ・タイプ	説明
BP_NAME	VARCHAR(128)	
MEMBER	SMALLINT	member - データベース・メンバー
AUTOMATIC	SMALLINT	automatic - バッファ・プール自動
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDA データの書き込み
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込み
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計

表 129. MON\_GET\_BUFFERPOOL について戻される情報 (続き)

列名	データ・タイプ	説明
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - バッファ・プール非同期データ読み取り
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - バッファ・プール非同期読み取り要求
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - バッファ・プール非同期データ書き込み
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - バッファ・プール非同期索引読み取り
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - バッファ・プール非同期索引読み取り要求
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - バッファ・プール非同期索引書き込み
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - バッファ・プール非同期 XDA データ読み取り
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - バッファ・プール非同期 XDA 読み取り要求
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - バッファ・プール非同期 XDA データ書き込み
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - バッファ・プールの非ビクティム・バッファ数
POOL_LSN_GAP_CLNS	BIGINT	pool_lsn_gap_clns - 起動されたバッファ・プール・ログ・スペース・クリーナー
POOL_DRDY_PG_STEAL_CLNS	BIGINT	pool_drty_pg_steal_clns - 起動されたバッファ・プール・ビクティム・ページ・クリーナー
POOL_DRDY_PG_THRSH_CLNS	BIGINT	pool_drty_pg_thrsh_clns - 起動されたバッファ・プールしきい値クリーナー
VECTORED_IOS	BIGINT	vectored_ios - ベクトル化入出力要求数
PAGES_FROM_VECTORED_IOS	BIGINT	pages_from_vectored_ios - ベクトル化入出力によって読み取られたページ数の合計
BLOCK_IOS	BIGINT	block_ios - ブロック入出力要求数
PAGES_FROM_BLOCK_IOS	BIGINT	pages_from_block_ios - ブロック入出力によって読み取られたページ数の合計
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不能プリフェッチ・ページ
FILES_CLOSED	BIGINT	files_closed - 閉じられたデータベース・ファイル
POOL_DATA_GBP_L_READS	BIGINT	pool_data_gbp_l_reads - グループ・バッファ・プールのデータの論理読み取りモニター・エレメント
POOL_DATA_GBP_P_READS	BIGINT	pool_data_gbp_p_reads - グループ・バッファ・プールのデータの物理読み取りモニター・エレメント

表 129. MON\_GET\_BUFFERPOOL について戻される情報 (続き)

列名	データ・タイプ	説明
POOL_DATA_LBP_PAGES_FOUND	BIGINT	pool_data_lbp_pages_found - ローカル・バッファ ー・プールで検出されたデータ・ページ・モニタ ー・エレメント
POOL_DATA_GBP_INVALID_PAGES	BIGINT	pool_data_gbp_invalid_pages - グループ・バッファ ー・プールの無効なデータ・ページ・モニター・ エレメント
POOL_INDEX_GBP_L_READS	BIGINT	pool_index_gbp_l_reads - グループ・バッファ ー・プールの索引の論理読み取りモニター・エレメン ト
POOL_INDEX_GBP_P_READS	BIGINT	pool_index_gbp_p_reads - グループ・バッファ ー・プールの索引の物理読み取りモニター・エレメン ト
POOL_INDEX_LBP_PAGES_FOUND	BIGINT	pool_index_lbp_pages_found - 検出されたローカ ル・バッファ ー・プールの索引ページ・モニタ ー・エレメント
POOL_INDEX_GBP_INVALID_PAGES	BIGINT	pool_index_gbp_invalid_pages - グループ・バッ ファ ー・プールの無効な索引ページ
POOL_ASYNC_DATA_GBP_L_READS	BIGINT	pool_async_data_gbp_l_reads - グループ・バッ ファ ー・プールの非同期データ論理読み取り
POOL_ASYNC_DATA_GBP_P_READS	BIGINT	pool_async_data_gbp_p_reads - グループ・バッ ファ ー・プールの非同期データ物理読み取り
POOL_ASYNC_DATA_LBP_PAGES_FOUND	BIGINT	pool_async_data_lbp_pages_found - ローカル・バ ッ ファ ー・プールの非同期データ・ページ検出
POOL_ASYNC_DATA_GBP_INVALID _PAGES	BIGINT	pool_async_data_gbp_invalid_pages - グループ・バ ッ ファ ー・プールの非同期無効データ・ページ
POOL_ASYNC_INDEX_GBP_L_READS	BIGINT	pool_async_index_gbp_l_reads - グループ・バッ ファ ー・プールの非同期索引論理読み取り
POOL_ASYNC_INDEX_GBP_P_READS	BIGINT	pool_async_index_gbp_p_reads - グループ・バッ ファ ー・プールの非同期索引物理読み取り
POOL_ASYNC_INDEX_LBP_PAGES _FOUND	BIGINT	pool_async_index_lbp_pages_found - ローカル・バ ッ ファ ー・プールの非同期索引ページ検出
POOL_ASYNC_INDEX_GBP_INVALID _PAGES	BIGINT	pool_async_index_gbp_invalid_pages - グループ・ バ ッ ファ ー・プールの非同期無効索引ページ
POOL_XDA_GBP_L_READS	BIGINT	pool_xda_gbp_l_reads - グループ・バッファ ー・プールの XDA データの論理読み取りの要求
POOL_XDA_GBP_P_READS	BIGINT	pool_xda_gbp_p_reads - グループ・バッファ ー・プールの XDA データの物理読み取りの要求
POOL_XDA_LBP_PAGES_FOUND	BIGINT	pool_xda_lbp_pages_found - 検出されたローカル・ バ ッ ファ ー・プールの XDA データ・ページ
POOL_XDA_GBP_INVALID_PAGES	BIGINT	pool_xda_gbp_invalid_pages - グループ・バッ ファ ー・プールの無効な XDA データ・ページ
POOL_ASYNC_XDA_GBP_L_READS	BIGINT	pool_async_xda_gbp_l_reads - グループ・バッ ファ ー・プールの XDA データの非同期論理読み取り の要求

表 129. MON\_GET\_BUFFERPOOL について戻される情報 (続き)

列名	データ・タイプ	説明
POOL_ASYNC_XDA_GBP_P_READS	BIGINT	pool_async_xda_gbp_p_reads - グループ・バッファ ー・プールの XDA データの非同期物理読み取り の要求
POOL_ASYNC_XDA_LBP_PAGES_FOUND	BIGINT	pool_async_xda_lbp_pages_found - 検出された非同 期ローカル・バッファ・プールの XDA デー タ・ページ
POOL_ASYNC_XDA_GBP_INVALID_PAGES	BIGINT	pool_async_xda_gbp_invalid_pages - 非同期グルー プ・バッファ・プールの無効な XDA データ・ ページ
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time バッファ・プール非同期 読み取り時間 :
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - バッファ・プール非同 期書き込み時間
BP_CUR_BUFFSZ	BIGINT	bp_cur_buffsz - バッファ・プールの現行のサイ ズ
POOL_QUEUED_ASYNC_DATA_REQS	BIGINT	pool_queued_async_data_reqs - データ・プリフェ ッチ要求
POOL_QUEUED_ASYNC_INDEX_REQS	BIGINT	pool_queued_async_index_reqs - 索引プリフェッチ 要求
POOL_QUEUED_ASYNC_XDA_REQS	BIGINT	pool_queued_async_xda_reqs - XDA プリフェッチ 要求
POOL_QUEUED_ASYNC_TEMP_DATA _REQS	BIGINT	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_INDEX _REQS	BIGINT	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要 求
POOL_QUEUED_ASYNC_TEMP_XDA _REQS	BIGINT	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求
POOL_QUEUED_ASYNC_OTHER_REQS	BIGINT	pool_queued_async_other_reqs - 非プリフェッチの 要求
POOL_QUEUED_ASYNC_DATA_PAGES	BIGINT	pool_queued_async_data_pages - データ・ページ・ プリフェッチ要求
POOL_QUEUED_ASYNC_INDEX_PAGES	BIGINT	pool_queued_async_index_pages - 索引ページ・プ リフェッチ要求
POOL_QUEUED_ASYNC_XDA_PAGES	BIGINT	pool_queued_async_xda_pages - XDA ページ・プ リフェッチ要求
POOL_QUEUED_ASYNC_TEMP_DATA _PAGES	BIGINT	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プ リフェッチ要求
POOL_QUEUED_ASYNC_TEMP_INDEX _PAGES	BIGINT	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフ ェッチ要求
POOL_QUEUED_ASYNC_TEMP_XDA _PAGES	BIGINT	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ペー ジ・プリフェッチ要求

表 129. MON\_GET\_BUFFERPOOL について戻される情報 (続き)

列名	データ・タイプ	説明
POOL_FAILED_ASYNC_DATA_REQS	BIGINT	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求
POOL_FAILED_ASYNC_INDEX_REQS	BIGINT	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求
POOL_FAILED_ASYNC_XDA_REQS	BIGINT	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求
POOL_FAILED_ASYNC_OTHER_REQS	BIGINT	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求
SKIPPED_PREFETCH_DATA_P_READS	BIGINT	skipped_prefetch_data_p_reads - スキップされたプリフェッチ (データ物理読み取り)
SKIPPED_PREFETCH_INDEX_P_READS	BIGINT	skipped_prefetch_index_p_reads - スキップされたプリフェッチ (索引物理読み取り)
SKIPPED_PREFETCH_XDA_P_READS	BIGINT	skipped_prefetch_xda_p_reads - スキップされたプリフェッチ (XDA 物理読み取り)
SKIPPED_PREFETCH_TEMP_DATA_P_READS	BIGINT	skipped_prefetch_temp_data_p_reads - スキップされたプリフェッチ (一時データ物理読み取り)
SKIPPED_PREFETCH_TEMP_INDEX_P_READS	BIGINT	skipped_prefetch_temp_index_p_reads - スキップされたプリフェッチ (一時索引物理読み取り)
SKIPPED_PREFETCH_TEMP_XDA_P_READS	BIGINT	skipped_prefetch_temp_xda_p_reads - スキップされたプリフェッチ (一時 XDA データ物理読み取り)
SKIPPED_PREFETCH_UOW_DATA_P_READS	BIGINT	skipped_prefetch_uow_data_p_reads - スキップされたプリフェッチ (作業単位データ物理読み取り)
SKIPPED_PREFETCH_UOW_INDEX_P_READS	BIGINT	skipped_prefetch_uow_index_p_reads - スキップされたプリフェッチ (作業単位索引物理読み取り)
SKIPPED_PREFETCH_UOW_XDA_P_READS	BIGINT	skipped_prefetch_uow_xda_p_reads - スキップされたプリフェッチ (作業単位 XDA データ物理読み取り)
SKIPPED_PREFETCH_UOW_TEMP_DATA_P_READS	BIGINT	skipped_prefetch_uow_temp_data_p_reads - スキップされたプリフェッチ (作業単位一時データ物理読み取り)
SKIPPED_PREFETCH_UOW_TEMP_INDEX_P_READS	BIGINT	skipped_prefetch_uow_temp_index_p_reads - スキップされたプリフェッチ (作業単位一時索引物理読み取り)
SKIPPED_PREFETCH_UOW_TEMP_XDA_P_READS	BIGINT	skipped_prefetch_uow_temp_xda_p_reads - スキップされたプリフェッチ (作業単位一時 XDA データ物理読み取り)

表 129. MON\_GET\_BUFFERPOOL について戻される情報 (続き)

列名	データ・タイプ	説明
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間
PREFETCH_WAITS	BIGINT	prefetch_waits - プリフェッチャーの待機カウント
POOL_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ・モニター・エレメント
POOL_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ・モニター・エレメント
POOL_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ・モニター・エレメント
POOL_ASYNC_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_async_data_gbp_indep_pages_found_in_lbp - 非同期 EDU によってローカル・バッファ・プールで検出されたグループ・バッファ・プール非従属データ・ページのモニター・エレメント
POOL_ASYNC_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_async_index_gbp_indep_pages_found_in_lbp - 非同期 EDU によってローカル・バッファ・プールで検出されたグループ・バッファ・プール非従属索引ページのモニター・エレメント
POOL_ASYNC_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_async_xda_gbp_indep_pages_found_in_lbp - 非同期 EDU によってローカル・バッファ・プールで検出されたグループ・バッファ・プール非従属 XML ストレージ・オブジェクト (XDA) ページのモニター・エレメント

## MON\_GET\_CF 表関数 - クラスタ・キャッシング・ファシリティー・メトリックの取得

MON\_GET\_CF 表関数は、システム上の 1 つ以上のクラスタ・キャッシング・ファシリティー (CF と呼ばれる) についての状況情報を戻します。

### 構文

▶▶—MON\_GET\_CF—(—id—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

*id* データが戻される対象となるクラスタ・キャッシング・ファシリティーの ID を指定する、タイプ INTEGER の入力引数。DB2\_CF 管理ビューを使用して、システム上のクラスタ・キャッシング・ファシリティーの ID を取得できます。このパラメーターが NULL の場合、システム上のすべてのクラスタ・キャッシング・ファシリティーの情報が戻されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

次の照会は、システム上のすべてのクラスター・キャッシング・ファシリティーのグループ・バッファ・プール・サイズに関する情報を表示します。進行中のグループ・バッファ・プールのサイズ変更を示します。CURRENT\_CF\_GBP\_SIZE パラメーターは、現在使用中のグループ・バッファ・プール・メモリーを示しています。CONFIGURED\_CF\_GBP\_SIZE パラメーターは、現在割り振られて予約されているグループ・バッファ・プール・メモリーを示しています。TARGET\_CF\_GBP\_SIZE パラメーターは、グループ・バッファ・プールの動的サイズ変更のターゲットを示しています。

```
SELECT SUBSTR(HOST_NAME,1,8) AS HOST,
       SUBSTR(DB_NAME, 1,8) AS DBNAME,
       CURRENT_CF_GBP_SIZE,
       CONFIGURED_CF_GBP_SIZE,
       TARGET_CF_GBP_SIZE
FROM TABLE( MON_GET_CF( cast(NULL as integer) ) ) AS CAMETRICS
ORDER BY HOST;
```

以下はこの照会の出力例です。

HOST	DBNAME	CURRENT_CF_GBP_SIZE	CONFIGURED_CF_GBP_SIZE	TARGET_CF_GBP_SIZE
cf15	SAMPLE	2402	3000	5000
cf16	SAMPLE	1276	3000	5000

次の例は、ID が 128 であるクラスター・キャッシング・ファシリティーのグループ・バッファ・プールのサイズに関する情報を戻します。

```
SELECT SUBSTR(HOST_NAME,1,8) AS HOST,
       SUBSTR(DB_NAME, 1,8) AS DBNAME,
       CURRENT_CF_GBP_SIZE,
       CONFIGURED_CF_GBP_SIZE,
       TARGET_CF_GBP_SIZE
FROM TABLE( MON_GET_CF( 128 ) ) AS CAMETRICS;
```

以下はこの照会の出力例です。

HOST	DBNAME	CURRENT_CF_GBP_SIZE	CONFIGURED_CF_GBP_SIZE	TARGET_CF_GBP_SIZE
cf16	SAMPLE	1276	2000	2000

## 使用上の注意

MON\_GET\_CF 表関数は、インスタンスで定義されたクラスター・キャッシング・ファシリティーとデータベースの対ごとに、1 行のデータを戻します。複数のクラ

スター・キャッシング・ファシリティーおよびデータベースにわたる集約は行われません。

## 戻される情報

表 130. MON\_GET\_CF で戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
HOST_NAME	VARCHAR(128)	host_name - ホスト名
ID	SMALLINT	id - ID
DB_NAME	VARCHAR(128)	db_name - データベース名
CURRENT_CF_GBP_SIZE	INTEGER	current_cf_gbp_size - 現行 CF のグループ・バッファ・プール・サイズ
CONFIGURED_CF_GBP_SIZE	INTEGER	configured_cf_gbp_size - 構成済み CF のグループ・バッファ・プール・サイズ
TARGET_CF_GBP_SIZE	INTEGER	target_cf_gbp_size - ターゲット CF のグループ・バッファ・プール・サイズ
CURRENT_CF_LOCK_SIZE	INTEGER	current_cf_lock_size - 現行 CF のロック・サイズ
CONFIGURED_CF_LOCK_SIZE	INTEGER	configured_cf_lock_size - 構成済み CF のロック・サイズ
TARGET_CF_LOCK_SIZE	INTEGER	target_cf_lock_size - ターゲット CF のロック・サイズ
CURRENT_CF_SCA_SIZE	INTEGER	current_cf_sca_size - 現行 CF の共用通信域サイズ
CONFIGURED_CF_SCA_SIZE	INTEGER	configured_cf_sca_size - 構成済み CF の共用通信域サイズ
TARGET_CF_SCA_SIZE	INTEGER	target_cf_sca_size - ターゲット CF の共用通信域サイズ
CURRENT_CF_MEM_SIZE	INTEGER	current_cf_mem_size - 現行 CF のメモリー・サイズ
CONFIGURED_CF_MEM_SIZE	INTEGER	configured_cf_mem_size - 構成済み CF のメモリー・サイズ

## MON\_GET\_CF\_CMD - クラスタ・キャッシング・ファシリティー・コマンドの処理時間の取得

MON\_GET\_CF\_CMD 表関数は、クラスタ・キャッシング・ファシリティー (CF) コマンドの処理時間に関する情報を戻します。

### 構文

▶▶—MON\_GET\_CF\_CMD—(—id—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

*id* データが戻される対象となるCFの ID を指定する、タイプ INTEGER の入力引数。DB2\_CF 管理ビューを使用して、DB2 pureScale 環境内の CF の ID を取得できます。このパラメーターが NULL の場合は、DB2 pureScale 環境内のすべての CF について情報が戻されます。



## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

複数のメンバーと 2 つのクラスター・キャッシング・ファシリティ (ID は CF15 と CF16) を持つ DB2 pureScale インスタンスに対して、次のステートメントを発行します。

```
SELECT SUBSTR(HOSTNAME,1,8) AS HOST,  
       SUBSTR(CF_CMD_NAME,1,20) AS CF_CMD_NAME,  
       TOTAL_CF_REQUESTS AS REQUESTS  
FROM TABLE(MON_GET_CF_CMD(NULL) ) AS CFMETRICS  
WHERE CF_CMD_NAME = 'SetLockState' OR CF_CMD_NAME = 'ReadAndRegister'
```

この照会の出力例を以下に示します。

HOST	CF_CMD_NAME	REQUESTS
CF15	SetLockState	4
CF15	ReadAndRegister	7
CF16	SetLockState	1
CF16	ReadAndRegister	1

## 使用上の注意

データが戻されるのは、この関数を DB2 pureScale 環境で発行した場合のみです。

報告される値は CF によって提供されます。メンバーのインスタンス・レベルやメンバーに対してアクティブなデータベースの数とは無関係です。

**TOTAL\_CF\_REQUESTS** 列および **TOTAL\_CF\_CMD\_TIME\_MICRO** 列で戻される値は、クラスターの開始以降に計算されます。値を再設定することはできません。これらは読み取り専用です。

**TOTAL\_CF\_CMD\_TIME\_MICRO** 列の戻り値を **TOTAL\_CF\_REQUESTS** 列で戻される数値で除算することにより、平均時間を計算できます。

## 戻される情報

表 131. MON\_GET\_CF\_CMD 関数で戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
HOSTNAME	VARCHAR(255)	hostname - ホスト名 : モニター・エレメント
ID	SMALLINT	id - ID
CF_CMD_NAME	VARCHAR(128)	CF コマンドの名前。

表 131. MON\_GET\_CF\_CMD 関数で戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_CF_REQUESTS	BIGINT	このコマンドに関する CF 内の要求の総数。
TOTAL_CF_CMD_TIME_MICRO	BIGINT	CF でのこのコマンドの合計処理時間 (マイクロ秒単位)。

表 132. MON\_GET\_CF\_CMD 関数によってモニターされる CF コマンドのリスト

CF コマンドまたは別名	説明
<b>AllocationUnitRecovery</b>	ローカル・ロック・マネージャー (LLM) がDETACHした後にグローバル・ロック・マネージャー (GLM) をリカバリーし、すべての非保持ロックを解放するとともに保持ロックをダウングレードします。
<b>AttachLocalCache</b>	グループ・バッファ・プール (GBP) にアタッチします。
<b>CrossInvalidate   XI</b>	単一の相互無効化メッセージを送信します。
<b>ExtendedMessageResponseBlock</b>	大規模なメッセージ応答ブロック送信と終端からの応答送信のための確認応答に関するメトリックを戻します。これは、通信媒体を介した測定です。
<b>ExtendedMessageResponseBlockAsync</b>	通信媒体を介した大規模なメッセージ応答ブロック送信のみに関するメトリックを戻します。通信媒体を介した終端からの応答送信のための確認応答は待機しません。送信の完了は別に検査されます。
<b>LockNotification</b>	ロック通知を送信します。
<b>MessageResponseBlock</b>	メッセージ応答ブロック送信と終端からの応答送信のための確認応答に関するメトリックを戻します。これは、通信媒体を介した測定です。
<b>MessageResponseBlockAsync</b>	通信媒体を介したメッセージ応答ブロック送信のみに関するメトリックを戻します。通信媒体を介した終端からの応答送信のための確認応答は待機しません。送信の完了は別に検査されます。
<b>ProcessSetLockStateExistingClient</b>	既存のクライアントからのロック状態設定を処理します。
<b>ProcessSetLockStateExistingLock</b>	既存のロックのロック状態設定の時間を処理します。
<b>ProcessSetLockStateNewClient</b>	新しいクライアントからのロック状態設定を処理します。
<b>ProcessSetLockStateNewLock</b>	新しいロックのロック状態設定の時間を処理します。
<b>ProcessSetLockStateSingleLock</b>	単一ロックのロック状態設定の内部時間を処理します。
<b>ReadAndRegister   RAR</b>	CF から単一ページを読み取ります。
<b>ReadCacheInfo</b>	GBP キャッシュ情報を読み取ります。

表 132. MON\_GET\_CF\_CMD 関数によってモニターされる CF コマンドのリスト (続き)

CF コマンドまたは別名	説明
<b>ReadCastoutClass</b>	CF からキャストアウトできるキャストアウト・クラス内のページのリストを読み取ります。
<b>ReadCCInfo</b>	CF からキャストアウト・クラスの範囲のキャストアウト情報を読み取ります。
<b>ReadForCastout</b>	キャストアウトできるように CF からデータを読み取ります。
<b>ReadForCastoutMultiple</b>	CF 内のキャストアウト・キューで適格項目をスキャンします。これは、ReadCastoutClass コマンドと ReadForCastout コマンドの組み合わせです。
<b>ReadForCastoutMultipleList</b>	CF 内のキャストアウト・キューで適格項目をスキャンします。これは、指定されたリストに対する ReadCastoutClass コマンドと ReadForCastout コマンドの組み合わせです。
<b>ReadLocks</b>	CF 内の GLM 構造で定義されたロックを読み取ります。
<b>ReadSA</b>	CF 内のスマート配列 (SA) から集約値を読み取ります。
<b>ReadSetLFS</b>	CF 内のログ・フラッシュ・シーケンス (LFS) 番号を取得または設定します。
<b>RecordLockState   RLS</b>	2 次 CF にロック状態を記録します。
<b>RegisterPageList</b>	ページのリストを CF に登録します。
<b>ReleaseCastoutLocks</b>	CF 内のページに対するキャストアウト・ロックを解放します。
<b>SetLockState   SLS</b>	CF のロック状態を設定します。
<b>TestPageValidity</b>	CF 内のローカル・ページが有効かどうかをテストします。
<b>TryInstant</b>	CF からインスタント・ロックを取得するために発行されたコマンドを戻します。
<b>WriteAndRegister   WAR</b>	単一ページのイメージを CF に送信します。
<b>WriteAndRegisterMultiple   WARM</b>	複数ページのイメージを CF に送信します。
<b>WriteAndRegisterMultipleSubOperation</b>	CF への複数ページのイメージの送信の副操作。このコマンドは、WARM 操作のためのデータが通信媒体から読み取られた後の、CF での各ページの処理時間を戻します。
<b>WriteSA</b>	SA に値を書き込み、CF 内の集約値を取り出します。

## MON\_GET\_CF\_WAIT\_TIME - クラスタ・キャッシング・ファシリティー・コマンドの待ち時間の取得

MON\_GET\_CF\_WAIT\_TIME 表関数は、クラスタ・キャッシング・ファシリティー (CF) コマンドの合計時間の測定を戻します。この時間には、CF との間のネットワーク・トランスポート時間、および CF 内部でのコマンドの実行時間が含まれます。

## 構文

►►—MON\_GET\_CF\_WAIT\_TIME—(—member—)————→

### 表関数パラメーター

スキーマは SYSPROC です。

#### member

現在接続しているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブ・メンバーの場合は -2 を指定します。NULL 値を指定した場合は、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

db2nodes.cfg ファイルで番号 128 とされている特定の CF で行われるページ登録解除の速度を調べるために、特定の時間の長さにおける値を比較します。開始するには、次のステートメントを発行します。

```
SELECT TOTAL_CF_WAIT_TIME_MICRO AS Dereg_T1,  
       TOTAL_CF_REQUESTS AS Dereg_RQ1  
FROM TABLE (MON_GET_CF_WAIT_TIME( -1 ))  
WHERE CF_CMD_NAME = 'DeregisterPage' and ID = '128'
```

この照会は、比較の初期値を戻します。次に、CF が追加ページを登録解除するのを 10 秒待ちます。

更新された値を取得するには、次のステートメントを発行します。

```
SELECT TOTAL_CF_WAIT_TIME_MICRO AS Dereg_T2,  
       TOTAL_CF_REQUESTS AS Dereg_RQ2  
FROM TABLE (MON_GET_CF_WAIT_TIME( -1 ))  
WHERE CF_CMD_NAME = 'DeregisterPage' and ID = '128'
```

最後に、次の公式を使用して、CF がページを登録解除する速度を計算します。

$$\text{Dereg\_PAGE\_RATE} = \frac{(\text{Dereg\_RQ2} - \text{Dereg\_RQ1})}{(\text{Dereg\_T2} - \text{Dereg\_T1})}$$

## 使用上の注意

データが戻されるのは、この関数を DB2 pureScale 環境で発行した場合のみです。

この表関数は、コマンドごとおよびメンバーごとに 1 行のデータを戻します。

値はメンバーのインスタンス・レベルで提供され、メンバーに対してアクティブなデータベースの数とは無関係です。メンバーはインスタンス・レベルで CF に接続されるからです。

戻り列の **TOTAL\_CF\_REQUESTS** と **TOTAL\_CF\_WAIT\_TIME\_MICRO** で戻される値は、クラスターの開始以降に計算されます。値を再設定することはできません。これらは読み取り専用です。

**TOTAL\_CF\_WAIT\_TIME\_MICRO** 列の戻り値を **TOTAL\_CF\_REQUESTS** 列で戻される数値で除算することにより、平均時間を計算できます。

## 戻される情報

表 133. MON\_GET\_CF\_WAIT\_TIME 関数で戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー
HOSTNAME	VARCHAR(255)	hostname - ホスト名 : モニター・エレメント
ID	SMALLINT	id - ID
CF_CMD_NAME	VARCHAR(128)	CF コマンドの名前。
TOTAL_CF_REQUESTS	BIGINT	このコマンドに関する CF 内の要求の総数。
TOTAL_CF_WAIT_TIME_MICRO	BIGINT	CF でのこのコマンドの合計待ち時間 (マイクロ秒単位)。

表 134. MON\_GET\_CF\_WAIT\_TIME 関数によってモニターされる CF コマンドのリスト

CF コマンドまたは別名	説明
DeletePage	CF からページを削除します。
DeletePageList	CF からページのリストを削除します。
DeleteSA	CF 内のスマート配列 (SA) から項目を削除します。
DeleteSAList	CF 内の SA から項目のリストを削除します。
DeregisterPage	CF からページを登録解除します。
DeregisterPageList	CF からページのリストを登録解除します。
GetAndIncLFS	CF 内のログ・フラッシュ・シーケンス (LFS) 番号を取得およびインクリメントします。
GetLFS	CF から LFS を取得します。
GetLSN	CF からログ・シーケンス番号 (LSN) を取得します。
GetNotification	CF 内のグローバル・ロック・マネージャー (GLM) からの通知メッセージを取得するために発行されたコマンドを戻します。
GetStatus	CF からユーザー制御構造フィールドを取得します。
GetUDF	ユーザー・データ・フィールドを使用して、CF から最も古いページを取得します。

表 134. MON\_GET\_CF\_WAIT\_TIME 関数によってモニターされる CF コマンドのリスト (続き)

CF コマンドまたは別名	説明
<b>GLMDump</b>	CF に GLM 構造のダンプを要求します。
<b>InitNotify</b>	CF 内の GLM 構造からのメッセージ通知を初期化するために発行されたコマンドを戻します。
<b>MemberUD</b>	CF 内のメンバーごとのユーザー・データ・フィールドにアクセスします。
<b>ReadAndRegister   RAR</b>	CF から単一ページを読み取ります。
<b>ReadAndRegisterMultiple   RARM</b>	CF から複数ページを読み取ります。
<b>ReadCastoutClass</b>	CF からキャストアウトできるキャストアウト・クラス内のページのリストを読み取ります。
<b>ReadCCInfo</b>	CF からキャストアウト・クラスの範囲のキャストアウト情報を読み取ります。
<b>ReadForCastout</b>	キャストアウトできるように CF からデータを読み取ります。
<b>ReadForCastoutMultiple</b>	CF 内のキャストアウト・キューで適格項目をスキャンします。このコマンドは、ReadCastoutClass コマンドと ReadForCastout コマンドの組み合わせです。
<b>ReadForCastoutMultipleList</b>	CF 内のキャストアウト・キューで適格項目をスキャンします。このコマンドは、指定されたリストに対する ReadCastoutClass コマンドと ReadForCastout コマンドの組み合わせです。
<b>ReadLocks</b>	CF 内の GLM 構造で定義されたロックを読み取ります。
<b>ReadSA</b>	CF 内の SA から集約値を読み取ります。
<b>ReadSAList</b>	CF 内のすべての SA 変数または特定の SA 変数のすべての値を読み取ります。
<b>ReleaseCastoutLocks</b>	CF 内のページに対するキャストアウト・ロックを解放します。
<b>RecordLockState   RLS</b>	2 次 CF にロック状態を記録します。
<b>RecordLockStateAsync</b>	2 次 CF にロック状態を非同期で記録します。
<b>RecordLockStateMultiple   RLSN</b>	2 次 CF に複数のロック状態を記録します。
<b>RegisterPageList</b>	ページのリストを CF に登録します。
<b>SetLFS</b>	CF 内の LFS 番号を設定します。
<b>SetLSN</b>	CF 内の LSN を設定します。
<b>SetLockState   SLS</b>	CF にロック状態を設定します。
<b>SetLockStateAsync</b>	CF にロック状態を非同期で設定します。
<b>SetLockStateMultiple   SLSN</b>	CF に複数のロック状態を設定します。
<b>SetLockStateMultipleAsync</b>	CF に複数のロック状態を非同期で設定します。
<b>SetStatus</b>	CF 内のユーザー制御構造フィールドを設定します。
<b>TestPageValidity</b>	CF 内のローカル・ページが有効かどうかをテストします。
<b>TryInstant</b>	CF からインスタント・ロックを取得するために発行されたコマンドを戻します。

表 134. MON\_GET\_CF\_WAIT\_TIME 関数によってモニターされる CF コマンドのリスト (続き)

CF コマンドまたは別名	説明
WriteAndRegister   WAR	単一ページのイメージを CF に送信します。
WriteAndRegisterMultiple   WARM	複数ページのイメージを CF に送信します。
WriteSA	SA に値を書き込み、CF 内の集約値を取り出します。
WriteSAList	CF 内の 1 つ以上の SA 変数を書き込みます。

## MON\_GET\_CONNECTION 表関数 - 接続メトリックの取得

MON\_GET\_CONNECTION 表関数は、1 つ以上の接続のメトリックを戻します。

### 構文

```
►► MON_GET_CONNECTION (—application_handle—, —member—) ◀◀
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### application\_handle

メトリックが戻される特定のアプリケーション・ハンドル (接続を識別する) を指定する、タイプ BIGINT の入力引数。引数が NULL の場合、すべての接続のメトリックが戻されます。

#### member

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

戻される行で順序付けられた、最も大きなデータ・ボリュームをクライアントに戻す接続を表示します。

```

SELECT application_handle,
       rows_returned,
       tcpip_send_volume
FROM TABLE(MON_GET_CONNECTION(cast(NULL as bigint), -2)) AS t
ORDER BY rows_returned DESC

```

以下はこの照会の出力例です。

```

APPLICATION_HANDLE  ROWS_RETURNED      TCPIP_SEND_VOLUME
-----
                    55                          6                          0

```

1 record(s) selected.

## 使用上の注意

MON\_GET\_CONNECTION 表関数で戻されるメトリックは、接続によってサブミットされた要求のすべてのメトリックの累計を表します。メトリックは、作業単位境界でロールアップされ、要求の実行中には定期的にロールアップされます。したがって、この表関数で報告される値は、直前の ROLLUP 時のシステムの現行状態を反映しています。メトリックの値は確実に増加します。ある時間間隔に対する指定されたメトリックの値を判別するには、MON\_GET\_CONNECTION 表関数を使用してその間隔の始めと終わりのメトリックを照会し、差異を計算します。

要求メトリックは、サービス・スーパークラスに対する COLLECT REQUEST METRICS 節、およびデータベース・レベルの *mon\_req\_metrics* データベース構成パラメーターを介して制御されます。親サービス・スーパークラスで要求メトリックを使用可能にしているサービス・サブクラスのエージェントが要求を処理する場合にのみ、またはデータベース全体で要求メトリック・コレクションが有効な場合にのみ、その要求に対しメトリックが収集されます。デフォルトでは、要求メトリックはデータベース・レベルで使用可能です。要求メトリックがデータベース・レベルでもサービス・スーパークラスに対しても使用不可になっている場合、そのサービス・スーパークラスにマッピングされた接続ごとに報告されるメトリックは増加を停止します (または、要求メトリックがデータベースのアクティブ化時に無効であった場合には 0 のままになります)。

**ヒント:** 接続は存続時間中に複数のサービス・スーパークラスにマッピングできるので、MON\_GET\_CONNECTION 表関数によって報告されるメトリックは、接続によってサブミットされたすべての要求のメトリックのサブセットを表す場合があります。これはメトリックの集合が、接続によってマッピングされるスーパークラスの一部に対して使用不可である場合に生じる可能性があります。

MON\_GET\_CONNECTION 表関数は、接続ごとおよびメンバーごとに 1 行のデータを戻します。(1 つ以上のサービス・クラスの) メンバー全体の集約は実行されません。しかし、集約は SQL 照会を使用して実行できます。

## 戻される情報

表 135. MON\_GET\_CONNECTION について戻される情報

列名	データ・タイプ	説明
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル
APPLICATION_NAME	VARCHAR(128)	appl_name - アプリケーション名
APPLICATION_ID	VARCHAR(128)	appl_id - アプリケーション ID



表 135. MON\_GET\_CONNECTION について戻される情報 (続き)

列名	データ・タイプ	説明
MEMBER	SMALLINT	member - データベース・メンバー
CLIENT_WRKSTNNAME	VARCHAR(255)	CURRENT CLIENT_WRKSTNNAME 特殊レジスター
CLIENT_ACCTNG	VARCHAR(255)	CURRENT CLIENT_ACCTNG 特殊レジスター
CLIENT_USERID	VARCHAR(255)	CURRENT CLIENT_USERID 特殊レジスター
CLIENT_APPLNAME	VARCHAR(255)	CURRENT CLIENT_APPLNAME 特殊レジスター
CLIENT_PID	BIGINT	client_pid - クライアント・プロセス ID
CLIENT_PRDID	VARCHAR(128)	client_prdid - クライアント製品およびバージョン ID
CLIENT_PLATFORM	VARCHAR(12)	client_platform - クライアント・プラットフォーム
CLIENT_PROTOCOL	VARCHAR(10)	client_protocol - クライアント通信プロトコル
SYSTEM_AUTH_ID	VARCHAR(128)	system_auth_id - システム許可 ID
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - セッション許可 ID
COORD_MEMBER	SMALLINT	coord_member - コーディネーター・メンバー
CONNECTION_START_TIME	TIMESTAMP	connection_start_time - 接続開始時刻
ACT_ABORTED_TOTAL	BIGINT	act_aborted_total - 打ち切られたアクティビティの合計数
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 完了したアクティビティの合計数
ACT_REJECTED_TOTAL	BIGINT	act_rejected_total - リジェクトされたアクティビティの合計数
AGENT_WAIT_TIME	BIGINT	agent_wait_time - エージェント待機時間
AGENT_WAITS_TOTAL	BIGINT	agent_waits_total - エージェント待機の合計
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り

表 135. MON\_GET\_CONNECTION について戻される情報 (続き)

列名	データ・タイプ	説明
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込み
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDA データの書き込み
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
CLIENT_IDLE_WAIT_TIME	BIGINT	client_idle_wait_time - クライアントのアイドル待機時間
DEADLOCKS	BIGINT	deadlocks - デッドロック検出数
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
FCM_RECV_VOLUME	BIGINT	fcm_recv_volume - FCM 受信ボリューム
FCM_RECVS_TOTAL	BIGINT	fcm_recvs_total - FCM 合計受信数
FCM_SEND_VOLUME	BIGINT	fcm_send_volume - FCM 送信ボリューム
FCM_SENDS_TOTAL	BIGINT	fcm_sends_total - FCM 合計送信数
FCM_RECV_WAIT_TIME	BIGINT	fcm_recv_wait_time - FCM 受信待機時間
FCM_SEND_WAIT_TIME	BIGINT	fcm_send_wait_time - FCM 送信待ち時間
IPC_RECV_VOLUME	BIGINT	ipc_recv_volume - プロセス間通信受信ボリューム
IPC_RECV_WAIT_TIME	BIGINT	ipc_recv_wait_time - プロセス間通信受信待ち時間
IPC_RECVS_TOTAL	BIGINT	ipc_recvs_total - プロセス間通信合計受信数
IPC_SEND_VOLUME	BIGINT	ipc_send_volume - プロセス間通信送信ボリューム
IPC_SEND_WAIT_TIME	BIGINT	ipc_send_wait_time - プロセス間通信送信待ち時間
IPC_SENDS_TOTAL	BIGINT	ipc_sends_total - プロセス間通信合計送信数
LOCK_ESCALS	BIGINT	lock_escals - ロック・エスカレーション数
LOCK_TIMEOUTS	BIGINT	lock_timeouts - ロック・タイムアウト数
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
LOG_BUFFER_WAIT_TIME	BIGINT	log_buffer_wait_time - ログ・バッファ待ち時間

表 135. MON\_GET\_CONNECTION について戻される情報 (続き)

列名	データ・タイプ	説明
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - フル・ログ・バッファの回数
LOG_DISK_WAIT_TIME	BIGINT	log_disk_wait_time - ログ・ディスク待機時間
LOG_DISK_WAITS_TOTAL	BIGINT	log_disk_waits_total - ログ・ディスク待機の合計
NUM_LOCKS_HELD	BIGINT	locks_held - ロック保持数
RQSTS_COMPLETED_TOTAL	BIGINT	rqsts_completed_total - 完了した要求の合計数
ROWS_MODIFIED	BIGINT	rows_modified - 変更された行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_RETURNED	BIGINT	rows_returned - 戻り行数
TCPIP_RECV_VOLUME	BIGINT	tcPIP_recv_volume - TCP/IP 受信ボリューム
TCPIP_SEND_VOLUME	BIGINT	tcPIP_send_volume - TCP/IP 送信ボリューム
TCPIP_RECV_WAIT_TIME	BIGINT	tcPIP_recv_wait_time - TCP/IP 受信待ち時間
TCPIP_RECVS_TOTAL	BIGINT	tcPIP_recvs_total - TCP/IP 合計受信数
TCPIP_SEND_WAIT_TIME	BIGINT	tcPIP_send_wait_time - TCP/IP 送信待ち時間
TCPIP_SENDS_TOTAL	BIGINT	tcPIP_sends_total - TCP/IP 合計送信数
TOTAL_APP_RQST_TIME	BIGINT	total_app_rqst_time - アプリケーション要求合計時間
TOTAL_RQST_TIME	BIGINT	total_rqst_time - 合計要求時間
WLM_QUEUE_TIME_TOTAL	BIGINT	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間
WLM_QUEUE_ASSIGNMENTS_TOTAL	BIGINT	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間
TOTAL_WAIT_TIME	BIGINT	total_wait_time - 合計待ち時間
APP_RQSTS_COMPLETED_TOTAL	BIGINT	app_rqsts_completed_total - 完了したアプリケーション要求の合計数
TOTAL_SECTION_SORT_TIME	BIGINT	total_section_sort_time - セクション・ソート時間合計
TOTAL_SECTION_SORT_PROC_TIME	BIGINT	total_section_sort_proc_time - セクション・ソート処理時間合計
TOTAL_SECTION_SORTS	BIGINT	total_section_sorts - セクション・ソート合計
TOTAL_SORTS	BIGINT	total_sorts - ソート合計
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - ポストしきい値ソート
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - ポスト共有しきい値ソート
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
TOTAL_COMPILE_TIME	BIGINT	total_compile_time - コンパイル時間の合計
TOTAL_COMPILE_PROC_TIME	BIGINT	total_compile_proc_time - コンパイル処理時間の合計
TOTAL_COMPILATIONS	BIGINT	total_compilations - コンパイルの合計回数
TOTAL_IMPLICIT_COMPILE_TIME	BIGINT	total_implicit_compile_time - 暗黙的コンパイル時間の合計
TOTAL_IMPLICIT_COMPILE_PROC_TIME	BIGINT	total_implicit_compile_proc_time - 暗黙的コンパイルの処理時間の合計

表 135. MON\_GET\_CONNECTION について戻される情報 (続き)

列名	データ・タイプ	説明
TOTAL_IMPLICIT_COMPILATIONS	BIGINT	total_implicit_compilations - 暗黙的コンパイルの合計回数
TOTAL_SECTION_TIME	BIGINT	total_section_time - セクション時間の合計
TOTAL_SECTION_PROC_TIME	BIGINT	total_section_proc_time - セクション処理時間の合計
TOTAL_APP_SECTION_EXECUTIONS	BIGINT	total_app_section_executions - セクション実行の合計回数
TOTAL_ACT_TIME	BIGINT	total_act_time - 合計アクティビティー時間
TOTAL_ACT_WAIT_TIME	BIGINT	total_act_wait_time - 合計アクティビティー待機時間
ACT_RQSTS_TOTAL	BIGINT	act_rqsts_total - アクティビティー要求の合計
TOTAL_ROUTINE_TIME	BIGINT	total_routine_time - ルーチン時間の合計
TOTAL_ROUTINE_INVOCATIONS	BIGINT	total_routine_invocations - ルーチン呼び出しの合計回数
TOTAL_COMMIT_TIME	BIGINT	total_commit_time - コミット時間の合計
TOTAL_COMMIT_PROC_TIME	BIGINT	total_commit_proc_time - コミット処理時間の合計
TOTAL_APP_COMMITS	BIGINT	total_app_commits - アプリケーション・コミットの合計回数
INT_COMMITS	BIGINT	int_commits - 内部コミット数
TOTAL_ROLLBACK_TIME	BIGINT	total_rollback_time - ロールバック時間の合計
TOTAL_ROLLBACK_PROC_TIME	BIGINT	total_rollback_proc_time - ロールバック処理時間の合計
TOTAL_APP_ROLLBACKS	BIGINT	total_app_rollbacks - アプリケーション・ロールバックの合計回数
INT_ROLLBACKS	BIGINT	int_rollbacks - 内部ロールバック数
TOTAL_RUNSTATS_TIME	BIGINT	total_runstats_time - ランタイム統計時間の合計
TOTAL_RUNSTATS_PROC_TIME	BIGINT	total_runstats_proc_time - ランタイム統計の処理時間の合計
TOTAL_RUNSTATS	BIGINT	total_runstats - ランタイム統計の合計回数
TOTAL_REORG_TIME	BIGINT	total_reorg_time - 再編成時間の合計
TOTAL_REORG_PROC_TIME	BIGINT	total_reorg_proc_time - 再編成の処理時間の合計
TOTAL_REORGS	BIGINT	total_reorgs - 再編成の合計回数
TOTAL_LOAD_TIME	BIGINT	total_load_time - ロード時間の合計
TOTAL_LOAD_PROC_TIME	BIGINT	total_load_proc_time - ロード処理時間の合計
TOTAL_LOADS	BIGINT	total_loads - ロードの合計回数
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - カタログ・キャッシュ挿入数
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - カタログ・キャッシュ参照数
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - パッケージ・キャッシュ挿入
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - パッケージ・キャッシュ参照
THRESH_VIOLATIONS	BIGINT	thresh_violations - しきい値違反の回数
NUM_LW_THRESH_EXCEEDED	BIGINT	num_lw_thresh_exceeded - しきい値を超えた回数
LOCK_WAITS_GLOBAL	BIGINT	lock_waits_global - グローバル・ロック待機
LOCK_WAIT_TIME_GLOBAL	BIGINT	lock_wait_time_global - グローバル・ロック待機時間

表 135. MON\_GET\_CONNECTION について戻される情報 (続き)

列名	データ・タイプ	説明
LOCK_TIMEOUTS_GLOBAL	BIGINT	lock_timeouts_global - グローバル・ロック・タイムアウト
LOCK_ESCALS_MAXLOCKS	BIGINT	lock_escal_maxlocks - maxlocks ロック・エスカレーション数
LOCK_ESCALS_LOCKLIST	BIGINT	lock_escal_locklist - locklist ロック・エスカレーション数
LOCK_ESCALS_GLOBAL	BIGINT	lock_escal_global - グローバル・ロック・エスカレーション数
RECLAIM_WAIT_TIME	BIGINT	reclaim_wait_time - 再利用の待機時間
SPACEMAPPAGE_RECLAIM_WAIT_TIME	BIGINT	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間
CF_WAITS	BIGINT	cf_waits - クラスタ・キャッシング・ファシリティー待機回数
CF_WAIT_TIME	BIGINT	cf_wait_time - クラスタ・キャッシング・ファシリティーの待機時間
POOL_DATA_GBP_L_READS	BIGINT	pool_data_gbp_l_reads - グループ・バッファ・プールのデータの論理読み取り
POOL_DATA_GBP_P_READS	BIGINT	pool_data_gbp_p_reads - グループ・バッファ・プールのデータの物理読み取り
POOL_DATA_LBP_PAGES_FOUND	BIGINT	pool_data_lbp_pages_found - ローカル・バッファ・プールで検出されたデータ・ページ
POOL_DATA_GBP_INVALID_PAGES	BIGINT	pool_data_gbp_invalid_pages - グループ・バッファ・プールの無効なデータ・ページ
POOL_INDEX_GBP_L_READS	BIGINT	pool_index_gbp_l_reads - グループ・バッファ・プールの索引の論理読み取り
POOL_INDEX_GBP_P_READS	BIGINT	pool_index_gbp_p_reads - グループ・バッファ・プールの索引の物理読み取り
POOL_INDEX_LBP_PAGES_FOUND	BIGINT	pool_index_lbp_pages_found - 検出されたローカル・バッファ・プールの索引ページ
POOL_INDEX_GBP_INVALID_PAGES	BIGINT	pool_index_gbp_invalid_pages - グループ・バッファ・プールの無効な索引ページ
POOL_XDA_GBP_L_READS	BIGINT	pool_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの論理読み取りの要求
POOL_XDA_GBP_P_READS	BIGINT	pool_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの物理読み取りの要求
POOL_XDA_LBP_PAGES_FOUND	BIGINT	pool_xda_lbp_pages_found - 検出されたローカル・バッファ・プールの XDA データ・ページ
POOL_XDA_GBP_INVALID_PAGES	BIGINT	pool_xda_gbp_invalid_pages - グループ・バッファ・プールの無効な XDA データ・ページ
AUDIT_EVENTS_TOTAL	BIGINT	audit_events_total - 監査イベント合計数
AUDIT_FILE_WRITES_TOTAL	BIGINT	audit_file_writes_total - 書き込まれた監査ファイルの合計数
AUDIT_FILE_WRITE_WAIT_TIME	BIGINT	audit_file_write_wait_time - 監査ファイル書き込み待機時間

表 135. MON\_GET\_CONNECTION について戻される情報 (続き)

列名	データ・タイプ	説明
AUDIT_SUBSYSTEM_WAITS_TOTAL	BIGINT	audit_subsystem_waits_total - 監査サブシステム待機の合計
AUDIT_SUBSYSTEM_WAIT_TIME	BIGINT	audit_subsystem_wait_time - 監査サブシステム待機時間
CLIENT_HOSTNAME	VARCHAR(255)	client_hostname - クライアント・ホスト名のモニター・エレメント
CLIENT_PORT_NUMBER	INTEGER	client_port_number - クライアント・ポート番号のモニター・エレメント
DIAGLOG_WRITES_TOTAL	BIGINT	diaglog_writes_total - 診断ログ・ファイル書き込みの合計
DIAGLOG_WRITE_WAIT_TIME	BIGINT	diaglog_write_wait_time - 診断ログ・ファイルの書き込み待機時間
FCM_MESSAGE_RECVS_TOTAL	BIGINT	fcm_message_recvs_total - FCM メッセージ合計受信数
FCM_MESSAGE_RECV_VOLUME	BIGINT	fcm_message_recv_volume - FCM メッセージ受信ボリューム
FCM_MESSAGE_RECV_WAIT_TIME	BIGINT	fcm_message_recv_wait_time - FCM メッセージの受信待機時間
FCM_MESSAGE_SENDS_TOTAL	BIGINT	fcm_message_sends_total - FCM メッセージ合計送信数
FCM_MESSAGE_SEND_VOLUME	BIGINT	fcm_message_send_volume - FCM メッセージ送信ボリューム
FCM_MESSAGE_SEND_WAIT_TIME	BIGINT	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
FCM_TQ_RECVS_TOTAL	BIGINT	fcm_tq_recvs_total - FCM 表キューの合計受信数
FCM_TQ_RECV_VOLUME	BIGINT	fcm_tq_recv_volume - FCM 表キューの受信ボリューム
FCM_TQ_RECV_WAIT_TIME	BIGINT	fcm_tq_recv_wait_time - FCM 表キューの受信待ち時間
FCM_TQ_SENDS_TOTAL	BIGINT	fcm_tq_sends_total - FCM 表キューの合計送信数
FCM_TQ_SEND_VOLUME	BIGINT	fcm_tq_send_volume - FCM 表キューの送信ボリューム
FCM_TQ_SEND_WAIT_TIME	BIGINT	fcm_tq_send_wait_time - FCM 表キューの送信待ち時間
LAST_EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	last_executable_id - 最後の実行可能 ID
LAST_REQUEST_TYPE	VARCHAR(32)	last_request_type - 最後の要求タイプ
TOTAL_ROUTINE_USER_CODE_PROC_TIME	BIGINT	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
TOTAL_ROUTINE_USER_CODE_TIME	BIGINT	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
TQ_TOT_SEND_SPILLS	BIGINT	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数

表 135. MON\_GET\_CONNECTION について戻される情報 (続き)

列名	データ・タイプ	説明
EVMON_WAIT_TIME	BIGINT	evmon_wait_time - イベント・モニターの待機時間
EVMON_WAITS_TOTAL	BIGINT	evmon_waits_total - イベント・モニターの待機の合計回数
TOTAL_EXTENDED_LATCH_WAIT_TIME	BIGINT	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
TOTAL_EXTENDED_LATCH_WAITS	BIGINT	total_extended_latch_waits - 拡張ラッチの合計待機回数
INTRA_PARALLEL_STATE	VARCHAR(3)	intra_parallel_state - パーティション内並列処理の現在の状態
TOTAL_STATS_FABRICATION_TIME	BIGINT	total_stats_fabrication_time - 統計作成の合計時間
TOTAL_STATS_FABRICATION_PROC_TIME	BIGINT	total_stats_fabrication_proc_time - 統計作成の合計処理時間
TOTAL_STATS_FABRICATIONS	BIGINT	total_stats_fabrications - 統計作成の合計回数
TOTAL_SYNC_RUNSTATS_TIME	BIGINT	total_sync_runstats_time - 同期 RUNSTATS 時間の合計
TOTAL_SYNC_RUNSTATS_PROC_TIME	BIGINT	total_sync_runstats_proc_time - 同期 RUNSTATS 処理時間の合計
TOTAL_SYNC_RUNSTATS	BIGINT	total_sync_runstats - 同期 RUNSTATS アクティビティの合計
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - ディスパッチャー実行キュー時間の合計
TOTAL_PEDS	BIGINT	total_peds - partial early distinct の合計回数
DISABLED_PEDS	BIGINT	disabled_peds - 無効化された partial early distinct
POST_THRESHOLD_PEDS	BIGINT	post_threshold_peds - partial early distinct しきい値
TOTAL_PEAS	BIGINT	total_peas - partial early aggregation の合計回数
POST_THRESHOLD_PEAS	BIGINT	post_threshold_peas - partial early aggregation しきい値
TQ_SORT_HEAP_REQUESTS	BIGINT	tq_sort_heap_requests - 表キューのソート・ヒープの要求
TQ_SORT_HEAP_REJECTIONS	BIGINT	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否
POOL_QUEUED_ASYNC_DATA_REQS	BIGINT	pool_queued_async_data_reqs - データ・プリフェッチ要求
POOL_QUEUED_ASYNC_INDEX_REQS	BIGINT	pool_queued_async_index_reqs - 索引プリフェッチ要求
POOL_QUEUED_ASYNC_XDA_REQS	BIGINT	pool_queued_async_xda_reqs - XDA プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求

表 135. MON\_GET\_CONNECTION について戻される情報 (続き)

列名	データ・タイプ	説明
POOL_QUEUED_ASYNC_OTHER_REQS	BIGINT	pool_queued_async_other_reqs - 非プリフェッチの要求
POOL_QUEUED_ASYNC_DATA_PAGES	BIGINT	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_INDEX_PAGES	BIGINT	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_XDA_PAGES	BIGINT	pool_queued_async_xda_pages - XDA ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_DATA_PAGES	BIGINT	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_INDEX_PAGES	BIGINT	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_XDA_PAGES	BIGINT	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求
POOL_FAILED_ASYNC_DATA_REQS	BIGINT	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求
POOL_FAILED_ASYNC_INDEX_REQS	BIGINT	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求
POOL_FAILED_ASYNC_XDA_REQS	BIGINT	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求
POOL_FAILED_ASYNC_OTHER_REQS	BIGINT	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間
PREFETCH_WAITS	BIGINT	prefetch_waits - プリフェッチャーの待機カウント
APP_ACT_COMPLETED_TOTAL	BIGINT	app_act_completed_total - 成功した外部コーディネーター・アクティビティの総数
APP_ACT_ABORTED_TOTAL	BIGINT	app_act_aborted_total - 失敗した外部コーディネーター・アクティビティの総数
APP_ACT_REJECTED_TOTAL	BIGINT	app_act_rejected_total - リジェクトされた外部コーディネーター・アクティビティの総数
TOTAL_CONNECT_REQUEST_TIME	BIGINT	total_connect_request_time - 接続要求またはユーザー切り替え要求の合計時間
TOTAL_CONNECT_REQUEST_PROC_TIME	BIGINT	total_connect_request_proc_time - 接続要求またはユーザー切り替え要求の合計処理時間



表 135. MON\_GET\_CONNECTION について戻される情報 (続き)

列名	データ・タイプ	説明
TOTAL_CONNECT_REQUESTS	BIGINT	total_connect_requests - 接続要求またはユーザー切り替え要求
TOTAL_CONNECT_AUTHENTICATION_TIME	BIGINT	total_connect_authentication_time - 接続またはユーザー切り替えの認証要求の合計時間
TOTAL_CONNECT_AUTHENTICATION_PROC_TIME	BIGINT	total_connect_authentication_proc_time - 接続認証の合計処理時間
TOTAL_CONNECT_AUTHENTICATIONS	BIGINT	total_connect_authentications - 接続またはユーザー切り替えの認証の実行回数
POOL_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	エージェントによってローカル・バッファ・プール (LBP) で検出されたグループ・バッファ・プール (GBP) 非依存のデータ・ページの数。
POOL_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	エージェントにより LBP で検出された GBP 非依存の索引ページの数。
POOL_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	エージェントにより LBP で検出された、GBP 非依存の XML ストレージ・オブジェクト (XDA) のデータ・ページの数。
COMM_EXIT_WAIT_TIME	BIGINT	comm_exit_wait_time - 通信バッファ出口待機時間モニター・エレメント
COMM_EXIT_WAITS	BIGINT	comm_exit_waits - 通信バッファ出口待機回数モニター・エレメント
IDA_SEND_WAIT_TIME	BIGINT	インデータベース分析処理にデータを送信するために待機した合計時間。
IDA_SENDS_TOTAL	BIGINT	インデータベース分析処理にデータが送信された合計回数。
IDA_SEND_VOLUME	BIGINT	インデータベース分析処理にデータベース・サーバーから送信されたデータの合計量。値はバイト数で報告されます。
IDA_RECV_WAIT_TIME	BIGINT	インデータベース分析処理からデータを受信するために待機した合計時間。
IDA_RECVS_TOTAL	BIGINT	インデータベース分析処理からデータを受け取った合計回数。
IDA_RECV_VOLUME	BIGINT	インデータベース分析処理からデータベース・サーバーが受け取ったデータの合計量。値はバイト数で報告されます。

## MON\_GET\_CONNECTION\_DETAILS 表関数 - 詳細な接続メトリックの取得

MON\_GET\_CONNECTION\_DETAILS 表関数は、1 つ以上の接続の詳細メトリックを戻します。

### 構文

▶▶—MON\_GET\_CONNECTION\_DETAILS—(—application\_handle—,—member—)—▶▶

スキーマは SYSPROC です。

## 表関数パラメーター

### *application\_handle*

メトリックが戻される特定のアプリケーション・ハンドル (接続を識別する) を指定する、タイプ **BIGINT** の入力引数。引数が **NULL** の場合、すべての接続についてメトリックが戻されます。

### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ **INTEGER** の入力引数。現行のデータベース・メンバーには **-1**、すべてのデータベース・メンバーには **-2** を指定します。NULL 値を指定すると、**-1** が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する **EXECUTE** 特権
- **DATAACCESS** 権限
- **DBADM** 権限
- **SQLADM** 権限

## デフォルトの **PUBLIC** 特権

なし

## 例

戻される行で順序付けられた、最も大きなボリュームのデータをクライアントに戻す接続を表示します。

```
SELECT detmetrics.application_handle,
       detmetrics.rows_returned,
       detmetrics.tcpip_send_volume
FROM TABLE(MON_GET_CONNECTION_DETAILS(CAST(NULL as bigint), -2))
AS CONNMETRICS,
XMLTABLE (XMLNAMESPACES( DEFAULT 'http://www.ibm.com/xmlns/prod/db2/mon'),
 '$detmetric/db2_connection' PASSING XMLPARSE(DOCUMENT CONNMETRICS.DETAILS)
 as "detmetric"
COLUMNS "APPLICATION_HANDLE" INTEGER PATH 'application_handle',
 "ROWS_RETURNED" BIGINT PATH 'system_metrics/rows_returned',
 "TCPIP_SEND_VOLUME" BIGINT PATH 'system_metrics/tcpip_send_volume'
) AS DETMETRICS
ORDER BY rows_returned DESC
```

以下はこの照会の出力例です。

APPLICATION_HANDLE	ROWS_RETURNED	TCPIP_SEND_VOLUME
-----	-----	-----
21	4	0

1 record(s) selected.

## 使用上の注意

MON\_GET\_CONNECTION\_DETAILS 表関数で戻されるメトリックは、接続によってサブミットされた要求のすべてのメトリックの累計を表します。この関数は、以下の点で MON\_GET\_CONNECTION 表関数と似ています。

- MON\_GET\_CONNECTION 表関数は、最も一般的に使用されるメトリックを列ベースの形式で戻すため、メトリックの取得において最も効率的な方法です。
- MON\_GET\_CONNECTION\_DETAILS 表関数は、使用可能なすべてのメトリック一式を XML 文書形式で戻すため、出力フォーマットを最大限柔軟性のあるものにします。XML ベースの出力は、XML パーサーで直接解析でき、XMLTABLE 関数でリレーショナル形式に変換することもできます (例を参照)。

メトリックは、作業単位境界でロールアップされ、要求の実行中には定期的にロールアップされます。したがって、この表関数で報告される値は、直前の ROLLUP 時のシステムの現行状態を反映しています。メトリックの値は確実に増加します。ある時間間隔に対する指定されたメトリックの値を判別するには、MON\_GET\_CONNECTION\_DETAILS 表関数を使用してその間隔の始めと終わりのメトリックを照会し、差異を計算します。

要求メトリックは、サービス・スーパークラスに対する COLLECT REQUEST METRICS 節、およびデータベース・レベルの `mon_req_metrics` データベース構成パラメーターを介して制御されます。親サービス・スーパークラスで要求メトリックを使用可能にしているサービス・サブクラスのエージェントが要求を処理する場合にのみ、またはデータベース全体で要求メトリック・コレクションが有効な場合にのみ、その要求に対しメトリックが収集されます。デフォルトでは、要求メトリックはデータベース・レベルで使用可能です。要求メトリックがデータベース・レベルでもサービス・スーパークラスに対しても使用不可になっている場合、そのサービス・スーパークラスにマッピングされた接続ごとに報告されるメトリックは増加を停止します (または、要求メトリックがデータベースのアクティブ化時に無効であった場合には 0 のままになります)。

**ヒント:** 接続はその存続時間中に複数のサービス・スーパークラスにマッピングされる場合があるので、収集がデータベース・レベルで無効になっていると、MON\_GET\_CONNECTION\_DETAILS 表関数によって報告されるメトリックは、接続を介してサブミットされたすべての要求のメトリックのサブセットを表していることがあります。これはメトリックの集合が、接続がマッピングするスーパークラスの一部に対して使用不可である場合に生じる可能性があります。

MON\_GET\_CONNECTION\_DETAILS 表関数は、接続ごとおよびメンバーごとに 1 行のデータを戻します。(1 つ以上のサービス・クラスの) メンバー全体の集約は実行されません。しかし、集約は SQL 照会を使用して実行できます。

DETAILS 列に戻される XML 文書のスキーマは、ファイル `sqllib/misc/DB2MonRoutines.xsd` で入手できます。詳細は、ファイル `sqllib/misc/DB2MonCommon.xsd` 内にあります。

## 戻される情報

表 136. MON\_GET\_CONNECTION\_DETAILS について戻される情報

列名	データ・タイプ	説明
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル
MEMBER	SMALLINT	member - データベース・メンバー
DETAILS	BLOB(1M)	作業単位についての詳細メトリックを含む XML 文書。エレメントの説明については、本書の表 137を参照してください。

以下の例は、DETAILS 列で戻される XML 文書の構造を示しています。

```
<db2_connection xmlns="http://www.ibm.com/xmlns/prod/db2/mon" release="90700000">
  <application_handle>21</application_handle>
  <member>0</member>
  <system_metrics release="90700000">
    <act_aborted_total>5</act_aborted_total>
    ...
    <wlm_queue_assignments_total>3</wlm_queue_assignments_total>
  </system_metrics>
</db2_connection>
```

完全スキーマについては、sqllib/misc/DB2MonRoutines.xsd を参照してください。

本書では、以下のような非プリミティブ XML 型定義を使用しています。

```
<xs:simpleType name="db2DbObjectString">
  <xs:restriction base="xs:string">
    <xs:maxLength value="128"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="db2PartitionNum">
  <xs:restriction base="xs:nonNegativeInteger">
    <xs:maxInclusive value="999"/>
  </xs:restriction>
</xs:simpleType>
```

表 137. MON\_GET\_CONNECTION\_DETAILS について戻される詳細メトリック

エレメント名	データ・タイプ	説明
act_aborted_total	xs:nonNegativeInteger	act_aborted_total - 打ち切られたアクティビティーの合計数
act_completed_total	xs:nonNegativeInteger	act_completed_total - 完了したアクティビティーの合計数
act_rejected_total	xs:nonNegativeInteger	act_rejected_total - リジェクトされたアクティビティーの合計数
act_rqsts_total	xs:nonNegativeInteger	act_rqsts_total - アクティビティー要求の合計
agent_wait_time	xs:nonNegativeInteger	agent_wait_time - エージェント待機時間
agent_waits_total	xs:nonNegativeInteger	agent_waits_total - エージェント待機の合計
app_act_aborted_total	xs:nonNegativeInteger	app_act_aborted_total - 失敗した外部コーディネーター・アクティビティーの総数
app_act_completed_total	xs:nonNegativeInteger	app_act_completed_total - 成功した外部コーディネーター・アクティビティーの総数

表 137. MON\_GET\_CONNECTION\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
app_act_rejected_total	xs:nonNegativeInteger	app_act_rejected_total - リジェクトされた外部コーディネーター・アクティビティの総数
app_rqsts_completed_total	xs:nonNegativeInteger	app_rqsts_completed_total - 完了したアプリケーション要求の合計数
application_handle	xs:nonNegativeInteger	application_handle - アプリケーション・ハンドル
application_id	xs:string	appl_id - アプリケーション ID
application_name	xs:string	appl_name - アプリケーション名
audit_events_total	xs:nonNegativeInteger	audit_events_total - 監査イベント合計数
audit_file_write_wait_time	xs:nonNegativeInteger	audit_file_write_wait_time - 監査ファイル書き込み待機時間
audit_file_writes_total	xs:nonNegativeInteger	audit_file_writes_total - 書き込まれた監査ファイルの合計数
audit_subsystem_wait_time	xs:nonNegativeInteger	audit_subsystem_wait_time - 監査サブシステム待機時間
audit_subsystem_waits_total	xs:nonNegativeInteger	audit_subsystem_waits_total - 監査サブシステム待機の合計
cat_cache_inserts	xs:nonNegativeInteger	cat_cache_inserts - カタログ・キャッシュ挿入数
cat_cache_lookups	xs:nonNegativeInteger	cat_cache_lookups - カタログ・キャッシュ参照数
client_acctng	xs:string (255)	CURRENT_CLIENT_ACCTNG 特殊レジスター
client_applname	xs:string (255)	CURRENT_CLIENT_APPLNAME 特殊レジスター
client_hostname	xs:string	client_hostname - クライアント・ホスト名
client_idle_wait_time	xs:nonNegativeInteger	client_idle_wait_time - クライアントのアイドル待機時間
client_pid	xs:nonNegativeInteger	client_pid - クライアント・プロセス ID
client_platform	xs:string	client_platform - クライアント・プラットフォーム
client_port_number	xs:nonNegativeInteger	client_port_number - クライアント・ポート番号
client_prdid	xs:string	client_prdid - クライアント製品およびバージョン ID
client_protocol	xs:string	client_protocol - クライアント通信プロトコル
client_userid	xs:string (255)	CURRENT_CLIENT_USERID 特殊レジスター
client_wrkstnname	xs:string (255)	CURRENT_CLIENT_WRKSTNNAME 特殊レジスター
comm_exit_wait_time	xs:nonNegativeInteger	comm_exit_wait_time - 通信バッファ出口待機時間モニター・エレメント
comm_exit_waits	xs:nonNegativeInteger	comm_exit_waits - 通信バッファ出口待機回数モニター・エレメント
connection_start_time	xs:dateTime	connection_start_time - 接続開始時刻
coord_member	xs:short	coord_member - コーディネーター・メンバー
deadlocks	xs:nonNegativeInteger	deadlocks - デッドロック検出数
diaglog_write_wait_time	xs:nonNegativeInteger	diaglog_write_wait_time - 診断ログ・ファイルの書き込み待機時間
diaglog_writes_total	xs:nonNegativeInteger	diaglog_writes_total - 診断ログ・ファイル書き込みの合計
direct_read_reqs	xs:nonNegativeInteger	direct_read_reqs - 直接読み取り要求
direct_read_time	xs:nonNegativeInteger	direct_read_time - 直接読み取り時間

表 137. MON\_GET\_CONNECTION\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
direct_reads	xs:nonNegativeInteger	direct_reads - データベースからの直接読み取り
direct_write_reqs	xs:nonNegativeInteger	direct_write_reqs - 直接書き込み要求
direct_write_time	xs:nonNegativeInteger	direct_write_time - 直接書き込み時間
direct_writes	xs:nonNegativeInteger	direct_writes - データベースへの直接書き込み
disabled_peds	xs:long	disabled_peds - 無効化された partial early distinct
evmon_wait_time	xs:nonNegativeInteger	evmon_wait_time - イベント・モニターの待機時間
evmon_waits_total	xs:nonNegativeInteger	evmon_waits_total - イベント・モニターの待機の合計回数
fcm_message_rcv_volume	xs:nonNegativeInteger	fcm_message_rcv_volume - FCM メッセージ受信ボリューム
fcm_message_rcv_wait_time	xs:nonNegativeInteger	fcm_message_rcv_wait_time - FCM メッセージの受信待機時間
fcm_message_rcvs_total	xs:nonNegativeInteger	fcm_message_rcvs_total - FCM メッセージ合計受信数
fcm_message_send_volume	xs:nonNegativeInteger	fcm_message_send_volume - FCM メッセージ送信ボリューム
fcm_message_send_wait_time	xs:nonNegativeInteger	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
fcm_message_sends_total	xs:nonNegativeInteger	fcm_message_sends_total - FCM メッセージ合計送信数
fcm_rcv_volume	xs:nonNegativeInteger	fcm_rcv_volume - FCM 受信ボリューム
fcm_rcv_wait_time	xs:nonNegativeInteger	fcm_rcv_wait_time - FCM 受信待機時間
fcm_rcvs_total	xs:nonNegativeInteger	fcm_rcvs_total - FCM 合計受信数
fcm_send_volume	xs:nonNegativeInteger	fcm_send_volume - FCM 送信ボリューム
fcm_send_wait_time	xs:nonNegativeInteger	fcm_send_wait_time - FCM 送信待ち時間
fcm_sends_total	xs:nonNegativeInteger	fcm_sends_total - FCM 合計送信数
fcm_tq_rcv_volume	xs:nonNegativeInteger	fcm_tq_rcv_volume - FCM 表キューの受信ボリューム
fcm_tq_rcv_wait_time	xs:nonNegativeInteger	fcm_tq_rcv_wait_time - FCM 表キューの受信待ち時間
fcm_tq_rcvs_total	xs:nonNegativeInteger	fcm_tq_rcvs_total - FCM 表キューの合計受信数
fcm_tq_send_volume	xs:nonNegativeInteger	fcm_tq_send_volume - FCM 表キューの送信ボリューム
fcm_tq_send_wait_time	xs:nonNegativeInteger	fcm_tq_send_wait_time - FCM 表キューの送信待ち時間
fcm_tq_sends_total	xs:nonNegativeInteger	fcm_tq_sends_total - FCM 表キューの合計送信数
ida_rcv_wait_time	xs:nonNegativeInteger	インデータベース分析処理からデータを受信するために待機した合計時間。
ida_rcv_volume	xs:nonNegativeInteger	インデータベース分析処理からデータベース・サーバーが受け取ったデータの合計量。値はバイト数で報告されます。
ida_rcvs_total	xs:nonNegativeInteger	インデータベース分析処理からデータを受け取った合計回数。
ida_send_wait_time	xs:nonNegativeInteger	インデータベース分析処理にデータを送信するために待機した合計時間。
ida_send_volume	xs:nonNegativeInteger	インデータベース分析処理にデータベース・サーバーから送信されたデータの合計量。値はバイト数で報告されます。

表 137. MON\_GET\_CONNECTION\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
ida_sends_total	xs:nonNegativeInteger	インデータベース分析処理にデータが送信された合計回数。
int_commits	xs:nonNegativeInteger	int_commits - 内部コミット数
int_rollback	xs:nonNegativeInteger	int_rollback - 内部ロールバック数
intra_parallel_state	xs:string	intra_parallel_state - パーティション内並列処理の現行状態モニター・エレメント
ipc_recv_volume	xs:nonNegativeInteger	ipc_recv_volume - プロセス間通信受信ボリューム
ipc_recv_wait_time	xs:nonNegativeInteger	ipc_recv_wait_time - プロセス間通信受信待ち時間
ipc_recvs_total	xs:nonNegativeInteger	ipc_recvs_total - プロセス間通信合計受信数
ipc_send_volume	xs:nonNegativeInteger	ipc_send_volume - プロセス間通信送信ボリューム
ipc_send_wait_time	xs:nonNegativeInteger	ipc_send_wait_time - プロセス間通信送信待ち時間
ipc_sends_total	xs:nonNegativeInteger	ipc_sends_total - プロセス間通信合計送信数
last_executable_id	xs:hexBinary(32)	last_executable_id - 最後の実行可能 ID
last_request_type	xs:string(32)	last_request_type - 最後の要求タイプ
lock_escals	xs:nonNegativeInteger	lock_escals - ロック・エスカレーション数
lock_timeouts	xs:nonNegativeInteger	lock_timeouts - ロック・タイムアウト数
lock_wait_time	xs:nonNegativeInteger	lock_wait_time - ロック待機中の時間
lock_waits	xs:nonNegativeInteger	lock_waits - ロック待機数
log_buffer_wait_time	xs:nonNegativeInteger	log_buffer_wait_time - ログ・バッファ待ち時間
log_disk_wait_time	xs:nonNegativeInteger	log_disk_wait_time - ログ・ディスク待機時間
log_disk_waits_total	xs:nonNegativeInteger	log_disk_waits_total - ログ・ディスク待機の合計
member	xs:nonNegativeInteger	member - データベース・メンバー
num_locks_held	xs:nonNegativeInteger	locks_held - ロック保持数
num_log_buffer_full	xs:nonNegativeInteger	num_log_buffer_full - フル・ログ・バッファの回数
num_lw_thresh_exceeded	xs:nonNegativeInteger	num_lw_thresh_exceeded - しきい値を超えた回数
pkg_cache_inserts	xs:nonNegativeInteger	pkg_cache_inserts - パッケージ・キャッシュ挿入
pkg_cache_lookups	xs:nonNegativeInteger	pkg_cache_lookups - パッケージ・キャッシュ参照
pool_data_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ・モニター・エレメント
pool_data_l_reads	xs:nonNegativeInteger	pool_data_l_reads - バッファ・プール・データの論理読み取り
pool_data_p_reads	xs:nonNegativeInteger	pool_data_p_reads - バッファ・プール・データの物理読み取り
pool_data_writes	xs:nonNegativeInteger	pool_data_writes - バッファ・プールへのデータの書き込み
pool_failed_async_data_reqs	xs:nonNegativeInteger	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求
pool_failed_async_index_reqs	xs:nonNegativeInteger	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求
pool_failed_async_other_reqs	xs:nonNegativeInteger	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求

表 137. MON\_GET\_CONNECTION\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
pool_failed_async_temp_data_reqs	xs:nonNegativeInteger	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求
pool_failed_async_temp_index_reqs	xs:nonNegativeInteger	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求
pool_failed_async_temp_xda_reqs	xs:nonNegativeInteger	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求
pool_failed_async_xda_reqs	xs:nonNegativeInteger	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求
pool_index_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ・モニター・エレメント
pool_index_l_reads	xs:nonNegativeInteger	pool_index_l_reads - バッファ・プール索引の論理読み取り
pool_index_p_reads	xs:nonNegativeInteger	pool_index_p_reads - バッファ・プール索引の物理読み取り
pool_index_writes	xs:nonNegativeInteger	pool_index_writes - バッファ・プール索引の書き込み
pool_queued_async_data_pages	xs:nonNegativeInteger	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求
pool_queued_async_data_reqs	xs:nonNegativeInteger	pool_queued_async_data_reqs - データ・プリフェッチ要求
pool_queued_async_index_pages	xs:nonNegativeInteger	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求
pool_queued_async_index_reqs	xs:nonNegativeInteger	pool_queued_async_index_reqs - 索引プリフェッチ要求
pool_queued_async_other_reqs	xs:nonNegativeInteger	pool_queued_async_other_reqs - 非プリフェッチの要求
pool_queued_async_temp_data_pages	xs:nonNegativeInteger	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求
pool_queued_async_temp_data_reqs	xs:nonNegativeInteger	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求
pool_queued_async_temp_index_pages	xs:nonNegativeInteger	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求
pool_queued_async_temp_index_reqs	xs:nonNegativeInteger	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求
pool_queued_async_temp_xda_pages	xs:nonNegativeInteger	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求
pool_queued_async_temp_xda_reqs	xs:nonNegativeInteger	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求
pool_queued_async_xda_reqs	xs:nonNegativeInteger	pool_queued_async_xda_reqs - XDA プリフェッチ要求
pool_read_time	xs:nonNegativeInteger	pool_read_time - バッファ・プール物理読み取り時間の合計
pool_temp_data_l_reads	xs:nonNegativeInteger	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
pool_temp_data_p_reads	xs:nonNegativeInteger	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り



表 137. MON\_GET\_CONNECTION\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
pool_temp_index_l_reads	xs:nonNegativeInteger	pool_temp_index_l_reads - バッファ・プールの論理読み取り
pool_temp_index_p_reads	xs:nonNegativeInteger	pool_temp_index_p_reads - バッファ・プールの物理読み取り
pool_temp_xda_l_reads	xs:nonNegativeInteger	pool_temp_xda_l_reads - バッファ・プールの XDA データの論理読み取り
pool_temp_xda_p_reads	xs:nonNegativeInteger	pool_temp_xda_p_reads - バッファ・プールの XDA データの物理読み取り
pool_write_time	xs:nonNegativeInteger	pool_write_time - バッファ・プールの物理書き込み時間の合計
pool_xda_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プールの XDA 非依存ページ・モニター・エレメント
pool_xda_gbp_invalid_pages	xs:nonNegativeInteger	pool_xda_gbp_invalid_pages - グループ・バッファ・プールの無効な XDA データ・ページ
pool_xda_gbp_l_reads	xs:nonNegativeInteger	pool_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの論理読み取りの要求
pool_xda_gbp_p_reads	xs:nonNegativeInteger	pool_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの物理読み取りの要求
pool_xda_l_reads	xs:nonNegativeInteger	pool_xda_l_reads - バッファ・プールの XDA データの論理読み取り
pool_xda_lbp_pages_found	xs:nonNegativeInteger	pool_xda_lbp_pages_found - 検出されたローカル・バッファ・プールの XDA データ・ページ
pool_xda_p_reads	xs:nonNegativeInteger	pool_xda_p_reads - バッファ・プールの XDA データの物理読み取り
pool_xda_writes	xs:nonNegativeInteger	pool_xda_writes - バッファ・プールの XDA データの書き込み
post_shrthreshold_sorts	xs:nonNegativeInteger	post_shrthreshold_sorts - ポスト共有しきい値ソート
post_threshold_peas	xs:long	post_threshold_peas - partial early aggregation しきい値
post_threshold_peds	xs:long	post_threshold_peds - partial early distinct しきい値
post_threshold_sorts	xs:nonNegativeInteger	post_threshold_sorts - ポストしきい値ソート
prefetch_wait_time	xs:nonNegativeInteger	prefetch_wait_time - プリフェッチ待ち時間
prefetch_waits	xs:nonNegativeInteger	prefetch_waits - プリフェッチャーの待機カウント
rows_modified	xs:nonNegativeInteger	rows_modified - 変更された行数
rows_read	xs:nonNegativeInteger	rows_read - 読み取り行数
rows_returned	xs:nonNegativeInteger	rows_returned - 戻り行数
rqsts_completed_total	xs:nonNegativeInteger	rqsts_completed_total - 完了した要求の合計数
session_auth_id	xs:string	session_auth_id - セッション許可 ID
sort_overflows	xs:nonNegativeInteger	sort_overflows - ソート・オーバーフロー
system_auth_id	xs:string	system_auth_id - システム許可 ID
tcpip_recv_volume	xs:nonNegativeInteger	tcpip_recv_volume - TCP/IP 受信ボリューム
tcpip_recv_wait_time	xs:nonNegativeInteger	tcpip_recv_wait_time - TCP/IP 受信待ち時間

表 137. MON\_GET\_CONNECTION\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
tcpip_recvs_total	xs:nonNegativeInteger	tcpip_recvs_total - TCP/IP 合計受信数
tcpip_send_volume	xs:nonNegativeInteger	tcpip_send_volume - TCP/IP 送信ボリューム
tcpip_send_wait_time	xs:nonNegativeInteger	tcpip_send_wait_time - TCP/IP 送信待ち時間
tcpip_sends_total	xs:nonNegativeInteger	tcpip_sends_total - TCP/IP 合計送信数
thresh_violations	xs:nonNegativeInteger	thresh_violations - しきい値違反の回数
total_act_time	xs:nonNegativeInteger	total_act_time - 合計アクティビティー時間
total_act_wait_time	xs:nonNegativeInteger	total_act_wait_time - 合計アクティビティー待機時間
total_app_commits	xs:nonNegativeInteger	total_app_commits - アプリケーション・コミットの合計回数
total_app_rollback	xs:nonNegativeInteger	total_app_rollback - アプリケーション・ロールバックの合計回数
total_app_rqst_time	xs:nonNegativeInteger	total_app_rqst_time - アプリケーション要求合計時間
total_app_section_executions	xs:nonNegativeInteger	total_app_section_executions - セクション実行の合計回数
total_commit_proc_time	xs:nonNegativeInteger	total_commit_proc_time - コミット処理時間の合計
total_commit_time	xs:nonNegativeInteger	total_commit_time - コミット時間の合計
total_compilations	xs:nonNegativeInteger	total_compilations - コンパイルの合計回数
total_compile_proc_time	xs:nonNegativeInteger	total_compile_proc_time - コンパイル処理時間の合計
total_compile_time	xs:nonNegativeInteger	total_compile_time - コンパイル時間の合計
total_connect_authentication_proc_time	xs:nonNegativeInteger	total_connect_authentication_proc_time - 接続認証の合計処理時間のモニター・エレメント
total_connect_authentication_time	xs:nonNegativeInteger	total_connect_authentication_time - 接続またはユーザー切り替えの認証要求の合計時間のモニター・エレメント
total_connect_authentications	xs:nonNegativeInteger	total_connect_authentications - 接続またはユーザー切り替えの認証の実行回数
total_connect_request_proc_time	xs:nonNegativeInteger	total_connect_request_proc_time - 接続要求またはユーザー切り替え要求の合計処理時間のモニター・エレメント
total_connect_request_time	xs:nonNegativeInteger	total_connect_request_time - 接続要求またはユーザー切り替え要求の合計時間のモニター・エレメント
total_connect_requests	xs:nonNegativeInteger	total_connect_requests - 接続要求またはユーザー切り替え要求
total_cpu_time	xs:nonNegativeInteger	total_cpu_time - 合計 CPU 時間
total_disp_run_queue_time	xs:long	total_disp_run_queue_time - ディスパッチャー実行キュー時間の合計
total_extended_latch_wait_time	xs:nonNegativeInteger	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
total_extended_latch_waits	xs:nonNegativeInteger	total_extended_latch_waits - 拡張ラッチの合計待機回数
total_implicit_compilations	xs:nonNegativeInteger	total_implicit_compilations - 暗黙的コンパイルの合計回数
total_implicit_compile_proc_time	xs:nonNegativeInteger	total_implicit_compile_proc_time - 暗黙的コンパイルの処理時間の合計

表 137. MON\_GET\_CONNECTION\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
total_implicit_compile_time	xs:nonNegativeInteger	total_implicit_compile_time - 暗黙的コンパイル時間の合計
total_load_proc_time	xs:nonNegativeInteger	total_load_proc_time - ロード処理時間の合計
total_load_time	xs:nonNegativeInteger	total_load_time - ロード時間の合計
total_loads	xs:nonNegativeInteger	total_loads - ロードの合計回数
total_peas	xs:long	total_peas - partial early aggregation の合計回数
total_peds	xs:long	total_peds - partial early distinct の合計回数
total_reorg_proc_time	xs:nonNegativeInteger	total_reorg_proc_time - 再編成の処理時間の合計
total_reorg_time	xs:nonNegativeInteger	total_reorg_time - 再編成時間の合計
total_reorgs	xs:nonNegativeInteger	total_reorgs - 再編成の合計回数
total_rollback_proc_time	xs:nonNegativeInteger	total_rollback_proc_time - ロールバック処理時間の合計
total_rollback_time	xs:nonNegativeInteger	total_rollback_time - ロールバック時間の合計
total_routine_invocations	xs:nonNegativeInteger	total_routine_invocations - ルーチン呼び出しの合計回数
total_routine_time	xs:nonNegativeInteger	total_routine_time - ルーチン時間の合計
total_routine_user_code_proc_time	xs:nonNegativeInteger	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
total_routine_user_code_time	xs:nonNegativeInteger	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
total_rqst_time	xs:nonNegativeInteger	total_rqst_time - 合計要求時間
total_runstats	xs:nonNegativeInteger	total_runstats - ランタイム統計の合計回数
total_runstats_proc_time	xs:nonNegativeInteger	total_runstats_proc_time - ランタイム統計の処理時間の合計
total_runstats_time	xs:nonNegativeInteger	total_runstats_time - ランタイム統計時間の合計
total_section_proc_time	xs:nonNegativeInteger	total_section_proc_time - セクション処理時間の合計
total_section_sort_proc_time	xs:nonNegativeInteger	total_section_sort_proc_time - セクション・ソート処理時間合計
total_section_sort_time	xs:nonNegativeInteger	total_section_sort_time - セクション・ソート時間合計
total_section_sorts	xs:nonNegativeInteger	total_section_sorts - セクション・ソート合計
total_section_time	xs:nonNegativeInteger	total_section_time - セクション時間の合計
total_sorts	xs:nonNegativeInteger	total_sorts - ソート合計
total_stats_fabrication_proc_time	xs:nonNegativeInteger	total_stats_fabrication_proc_time - 統計作成の合計処理時間
total_stats_fabrication_time	xs:nonNegativeInteger	total_stats_fabrication_time - 統計作成の合計時間
total_stats_fabrications	xs:nonNegativeInteger	total_stats_fabrications - 統計作成の合計回数
total_sync_runstats	xs:nonNegativeInteger	total_sync_runstats - 同期 RUNSTATS アクティビティの合計
total_sync_runstats_proc_time	xs:nonNegativeInteger	total_sync_runstats_proc_time - 同期 RUNSTATS 処理時間の合計
total_sync_runstats_time	xs:nonNegativeInteger	total_sync_runstats_time - 同期 RUNSTATS 時間の合計
total_wait_time	xs:nonNegativeInteger	total_wait_time - 合計待ち時間

表 137. MON\_GET\_CONNECTION\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
tq_sort_heap_rejections	xs:long	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否
tq_sort_heap_requests	xs:long	tq_sort_heap_requests - 表キューのソート・ヒープの要求
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
wlm_queue_assignments_total	xs:nonNegativeInteger	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
wlm_queue_time_total	xs:nonNegativeInteger	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間

## MON\_GET\_CONTAINER 表関数 - 表スペース・コンテナ・メトリックの取得

MON\_GET\_CONTAINER 表関数は、1 つ以上の表スペース・コンテナのモニター・メトリックを戻します。

### 構文

►►MON\_GET\_CONTAINER(—*tbsp\_name*—,—*member*—)◄◄

スキーマは SYSPROC です。

### 表関数パラメーター

#### *tbsp\_name*

この関数を呼び出すときに現在接続されているデータベースと同じデータベース内の有効な表スペース名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべての表スペース内のすべてのコンテナについてメトリックが戻されます。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

### 例

例 1: 読み取り時間が最長のすべてのデータベース・メンバーのコンテナをリストします。

```
SELECT varchar(container_name,70) as container_name,
       varchar(tbsp_name,20) as tbsp_name,
       pool_read_time
FROM TABLE(MON_GET_CONTAINER('','-2)) AS t
ORDER BY pool_read_time DESC
```

以下はこの照会の出力例です。

```
CONTAINER_NAME
-----
/home/hotel55/swalkty/swalkty/NODE0000/TEST/T0000000/C0000000.CAT ...
/home/hotel55/swalkty/swalkty/NODE0000/TEST/T0000002/C0000000.LRG ...
/home/hotel55/swalkty/swalkty/NODE0000/TEST/T0000001/C0000000.TMP ...
```

3 record(s) selected.

照会の出力 (続き)。

```
... TBSP_NAME          POOL_READ_TIME
... -----
... SYSCATSPACE              597
... USERSPACE1                42
... TEMPSPACE1                 0
```

例 2: アクセスできないすべてのコンテナをリストします。

```
SELECT varchar(container_name, 70) as container_name
FROM TABLE(MON_GET_CONTAINER('','-1)) AS t
WHERE accessible = 0
```

以下はこの照会の出力例です。

```
CONTAINER_NAME
-----
0 record(s) selected.
```

例 3: 使用率の高い順にコンテナ・ファイル・システムの使用率をリストします。

```
SELECT varchar(container_name, 65) as container_name,
       fs_id,
       fs_used_size,
       fs_total_size,
       CASE WHEN fs_total_size > 0
            THEN DEC(100*(FLOAT(fs_used_size)/FLOAT(fs_total_size)),5,2)
            ELSE DEC(-1,5,2)
       END as utilization
FROM TABLE(MON_GET_CONTAINER('','-1)) AS t
ORDER BY utilization DESC
```

以下はこの照会の出力例です。

```
CONTAINER_NAME
-----
/home/hotel55/swalkty/swalkty/NODE0000/TEST/T0000000/C0000000.CAT ...
```

```
/home/hotel55/swalkty/swalkty/NODE0000/TEST/T0000001/C0000000.TMP ...
/home/hotel55/swalkty/swalkty/NODE0000/TEST/T0000002/C0000000.LRG ...
```

3 record(s) selected.

照会の出力 (続き)。

FS_ID	FS_USED_SIZE	FS_TOTAL_SIZE	UTILIZATION
64768	106879311872	317068410880	33.70
64768	106879311872	317068410880	33.70
64768	106879311872	317068410880	33.70

### 使用上の注意

MON\_GET\_CONTAINER 表関数は、コンテナごとおよびデータベース・メンバーごとに 1 行のデータを戻します。データは、指定の表スペース内のすべてのコンテナ、またはデータベース内のすべてのコンテナについて戻すことができます。データベース・パーティション全体からの集約は実行されません。しかし、集約は SQL 照会を使用して実行できます。

この関数によって収集されるメトリックは、mon\_obj\_metrics 構成パラメーターを使用してデータベース・レベルで制御されます。デフォルトでは、メトリック収集は有効になります。

### 戻される情報

表 138. MON\_GET\_CONTAINER について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
CONTAINER_NAME	VARCHAR(256)	container_name - コンテナ名
CONTAINER_ID	BIGINT	container_id - コンテナ ID
MEMBER	SMALLINT	member - データベース・メンバー
CONTAINER_TYPE	VARCHAR(16)	container_type - コンテナ・タイプ。これは、sqlutil.h での定義を基にしたテキスト ID です。以下のいずれかとなります。 <ul style="list-style-type: none"> <li>• DISK_EXTENT_TAG</li> <li>• DISK_PAGE_TAG</li> <li>• FILE_EXTENT_TAG</li> <li>• FILE_PAGE_TAG</li> <li>• PATH</li> </ul>
STRIPE_SET	BIGINT	container_stripe_set - ストライプ・セット
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
PAGES_READ	BIGINT	pages_read - 読み取られたページの数
PAGES_WRITTEN	BIGINT	pages_written - 書き込まれたページの数
VECTORED_IOS	BIGINT	vectored_ios - ベクトル化入出力要求数

表 138. MON\_GET\_CONTAINER について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
PAGES_FROM_VECTORED_IOS	BIGINT	pages_from_vectored_ios - ベクトル化入出力によって読み取られたページ数の合計
BLOCK_IOS	BIGINT	block_ios - ブロック入出力要求数
PAGES_FROM_BLOCK_IOS	BIGINT	pages_from_block_ios - ブロック入出力によって読み取られたページ数の合計
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
TOTAL_PAGES	BIGINT	container_total_pages - コンテナ内の合計ページ数
USABLE_PAGES	BIGINT	container_usable_pages - コンテナ内の使用可能なページ数
ACCESSIBLE	SMALLINT	container_accessible - コンテナのアクセス可能性
FS_ID	VARCHAR(22)	fs_id - 固有のファイル・システム識別番号
FS_TOTAL_SIZE	BIGINT	fs_total_size - ファイル・システムの合計サイズ
FS_USED_SIZE	BIGINT	fs_used_size - ファイル・システム上で使用されるスペースの量
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント  パーティション・データベース環境の場合、これは MEMBER 列の値と同じになります。DB2 Enterprise Server Edition および DB2 pureScale では、この値は 0 になります。 <b>注:</b> DBPARTITIONNUM は data_partition_id とは異なります。後者は、値に基づいて表内のデータを細分割することにより作成されたデータ・パーティションを識別するために使用されます。
DB_STORAGE_PATH_ID	BIGINT	db_storage_path_id - ストレージ・パス ID : モニター・エレメント

## MON\_GET\_EXTENDED\_LATCH\_WAIT 表関数 - ラッチに関する情報を戻す

この関数は、拡張ラッチ待機で扱われたラッチに関する情報を戻します。この情報には、メンバー、ラッチ名、拡張待機の回数、拡張待機に費やされた時間が含まれます。

### 構文

▶▶—MON\_GET\_EXTENDED\_LATCH\_WAIT—(—member—)————▶▶

スキーマは SYSPROC です。

## 表関数パラメーター

### *member*

データが戻されるメンバー ID。 NULL または -1 を指定した場合、現在接続されているメンバーのデータが戻されます。 -2 を指定した場合は、すべてのメンバーのデータが戻されます。その他の場合は、指定したメンバーのデータだけが戻されます。無効、未定義、またはオフラインのメンバーを指定した場合は、データが戻されません。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 使用上の注意

MON\_GET\_EXTENDED\_LATCH\_WAIT は、指定したメンバーのどのラッチが拡張ラッチ待機で扱われたかについての情報を戻します。この情報には、特定のラッチが拡張ラッチ待機で扱われた時間と回数が含まれます。この情報は、基本モニター・メトリックの一部として収集されます。

## 戻される情報

表 139. MON\_GET\_EXTENDED\_LATCH\_WAIT で戻される情報

列名	データ・タイプ	説明
MEMBER	SMALLINT	member - データベース・メンバー
LATCH_NAME	VARCHAR(256)	ラッチの名前。この値は、db2diag ログ・ファイルやスタック・ファイルなどの DB2 診断ファイルで報告されるラッチ名に対応します。
TOTAL_EXTENDED_LATCH_WAITS	BIGINT	total_extended_latch_waits - 拡張ラッチの合計待機回数
TOTAL_EXTENDED_LATCH_WAIT_TIME	BIGINT	total_extended_latch_wait_time - 拡張ラッチの合計待機時間

## MON\_GET\_EXTENT\_MOVEMENT\_STATUS - エクステンツ移動進行状況の取得

MON\_GET\_EXTENT\_MOVEMENT\_STATUS 表関数は、エクステンツ移動操作の状況を戻します。

## 構文



スキーマは SYSPROC です。

### 表関数パラメーター

#### *tbsp\_name*

照会する表スペースを指定する、タイプ VARCHAR(128) の入力引数。引数値が NULL の場合、この関数はすべての表スペースについて情報を戻します。

#### *member*

現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。引数値が NULL の場合は、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

すべての表スペースについて、現在のエクステント進行状況に関するすべての情報を取得します。

```
SELECT * FROM TABLE(SYSPROC.MON_GET_EXTENT_MOVEMENT_STATUS('', -1))
```

以下は、この照会の出力例です。

TBSP_NAME	TBSP_ID	MEMBER	CURRENT_EXTENT	LAST_EXTENT	NUM_EXTENTS_MOVED
SYSCATSPACE	0	0	-1	-1	-1
TEMPSPACE1	1	0	-1	-1	-1
USERSPACE1	2	0	-1	-1	-1
TS1	3	0	1	2	3
SYSTOOLSPACE	4	0	-1	-1	-1

5 record(s) selected.

この照会からの出力 (続き):

...	NUM_EXTENTS_LEFT	TOTAL_MOVE_TIME	ADDITIONAL_DETAILS
...	-1	-1	-
...	-1	-1	-
...	-1	-1	-
...	4	0	-
...	-1	-1	-

## 戻される情報

表 140. MON\_GET\_EXTENT\_MOVEMENT\_STATUS について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
MEMBER	SMALLINT	member - この情報が収集されたメンバー
CURRENT_EXTENT	INTEGER	current_extent - 移動中の現在のエクステント
LAST_EXTENT	INTEGER	last_extent - 最後に移動されたエクステント
NUM_EXTENTS_MOVED	INTEGER	num_extents_moved - このエクステント移動操作中にこれまで移動したエクステントの数
NUM_EXTENTS_LEFT	INTEGER	num_extents_left - このエクステント移動操作中にまだ移動されていないエクステントの数
TOTAL_MOVE_TIME	BIGINT	total_move_time - 移動されたすべてのエクステントの合計移動時間 (ミリ秒単位)

## MON\_GET\_FCM - FCM メトリックの取得

MON\_GET\_FCM 表関数は、高速コミュニケーション・マネージャー (FCM) に関するメトリックを戻します。

### 構文

▶▶ MON\_GET\_FCM (—member—) ◀◀

スキーマは SYSPROC です。

### 表関数パラメーター

#### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現行データベース・メンバーの場合は -1、すべてのアクティブ・データベース・メンバーからの情報の場合は -2 を指定します。アクティブ・データベース・メンバーとは、アプリケーションがデータベースに接続して使用できるところのことです。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限

- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

### 例

すべてのメンバーにおける高速コミュニケーション・マネージャーのメッセージ・バッファについての情報を検索するには、次のようにします。

```
SELECT member, buff_free, buff_free_bottom
FROM TABLE (MON_GET_FCM (-2))
```

この照会は、以下を戻します。

MEMBER	BUFF_FREE	BUFF_FREE_BOTTOM
2	13425	13416
10	13425	13416
1	13425	13416

3 record(s) selected.

### 戻される情報

表 141. MON\_GET\_FCM について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
HOSTNAME	VARCHAR(128)	hostname - ホスト名
MEMBER	SMALLINT	member - データベース・メンバー
BUFF_MAX	BIGINT	buff_max - FCM バッファの最大可能数
BUFF_TOTAL	BIGINT	buff_total - 現在割り振られている FCM バッファ数
BUFF_FREE	BIGINT	buff_free - 現在空いている FCM バッファ
BUFF_FREE_BOTTOM	BIGINT	buff_free_bottom - 空き FCM バッファの最小数
BUFF_AUTO_TUNING	SMALLINT	buff_auto_tuning - FCM バッファ自動チューニング標識
CH_MAX	BIGINT	ch_max - FCM チャンルの最大可能数
CH_TOTAL	BIGINT	ch_total - 現在割り振られている FCM チャンル数
CH_FREE	BIGINT	ch_free - 現在空いているチャンネル
CH_FREE_BOTTOM	BIGINT	ch_free_bottom - 空いているチャンネルの最小
CH_AUTO_TUNING	SMALLINT	ch_auto_tuning - FCM チャンル自動チューニング標識

注: この表関数が提供するメトリックは、ある特定のホスト・マシン上のすべてのメンバーに適用されます。ある特定のホスト・マシン上のすべてのメンバーが、同一セットのバッファとチャンネルを共有します。これは、ある特定のホスト・マシン上の各メンバーで個々のメトリックが通常は同じになることを意味します。ただし、各メンバーは個別に実行し、異なるメンバーのサンプリング間ではリソース番号が異なるため、メトリックが若干異なる可能性があります。

## MON\_GET\_FCM\_CONNECTION\_LIST - すべての FCM 接続の詳細の取得

MON\_GET\_FCM\_CONNECTION\_LIST 表関数は、指定されたメンバーのすべての高速コミュニケーション・マネージャー (FCM) 接続に関するモニター・メトリックを戻します。

### 構文

```
▶▶—MON_GET_FCM_CONNECTION_LIST—(—member—)————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現行データベース・メンバーの場合は -1、すべてのアクティブ・データベース・メンバーからの情報の場合は -2 を指定します。アクティブ・データベース・メンバーとは、アプリケーションがデータベースに接続して使用できるところのことです。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 戻される情報

表 142. MON\_GET\_FCM\_CONNECTION\_LIST について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー
REMOTE_MEMBER	SMALLINT	remote_member - リモート・メンバー
CONNECTION_STATUS	VARCHAR(16)	connection_status - 接続状況。
TOTAL_BUFFERS_SENT	BIGINT	total_buffers_sent - 送信された FCM バッファの合計
TOTAL_BUFFERS_RCVD	BIGINT	total_buffers_rcvd - 受信された FCM バッファの合計
FCM_CONGESTION_TIME	BIGINT	fcm_congestion_time - FCM 輻輳時間
FCM_CONGESTED_SENDS	BIGINT	fcm_congested_sends - FCM 輻輳送信
FCM_NUM_CONGESTION_TIMEOUTS	BIGINT	fcm_num_congestion_timeouts - FCM 輻輳タイムアウト
FCM_SEND_VOLUME	BIGINT	fcm_send_volume - FCM 送信ボリューム
FCM_RECV_VOLUME	BIGINT	fcm_recv_volume - FCM 受信ボリューム

表 142. MON\_GET\_FCM\_CONNECTION\_LIST について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
FCM_MESSAGE_SEND_VOLUME	BIGINT	fcm_message_send_volume - FCM メッセージ送信ボリューム
FCM_MESSAGE_RECV_VOLUME	BIGINT	fcm_message_recv_volume - FCM メッセージ受信ボリューム
FCM_TQ_SEND_VOLUME	BIGINT	fcm_tq_send_volume - FCM 表キューの送信ボリューム
FCM_TQ_RECV_VOLUME	BIGINT	fcm_tq_recv_volume - FCM 表キューの受信ボリューム
FCM_NUM_CONN_LOST	BIGINT	fcm_num_conn_lost - FCM 接続損失数
FCM_NUM_CONN_TIMEOUTS	BIGINT	fcm_num_conn_timeouts - FCM 接続タイムアウト数

## MON\_GET\_GROUP\_BUFFERPOOL

MON\_GET\_GROUP\_BUFFERPOOL 表関数は、グループ・バッファ・プール (GBP) に関する統計を戻します。

### 構文

►►MON\_GET\_GROUP\_BUFFERPOOL(—member—)◄◄

### 表関数パラメーター

スキーマは SYSPROC です。

#### member

現在接続しているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブ・メンバーの場合は -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

グループ・バッファ・プール (GBP) に十分なスペースがない場合に GBP へのページ登録やページ書き込みを試行すると、GBP\_FULL エラーが発生します。

以下の例では、GBP\_FULL エラーの検出回数がすべてのメンバーから戻されます。

```
SELECT SUM(T.NUM_GBP_FULL) AS NUM_GBP_FULL
FROM TABLE(MON_GET_GROUP_BUFFERPOOL(-2)) AS T
```

以下はこの照会の出力例です。

```
NUM_GBP_FULL
-----
          123
```

1 record(s) selected.

NUM\_GBP\_FULL の値が 1 分当たり 2 つ以上増える場合は、GBP の現行サイズがニーズを満たしていないことが考えられます。その場合は、次のコマンドで GBP のサイズを大きくしてください。

```
UPDATE DB CFG USING CF_GBP_SIZE <new_size>
```

このコマンドの <new\_size> の値により、GBP\_FULL エラーの回数の増加が遅くなるまたは止まるのに十分なサイズまでグループ・バッファ・プールが大きくなります。

## 戻される情報

表 143. MON\_GET\_GROUP\_BUFFERPOOL について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー
NUM_GBP_FULL	BIGINT	GBP_FULL エラーの発生回数。

## MON\_GET\_HADR 表関数 - 高可用性災害時リカバリー (HADR) モニター情報を戻す

この関数は、高可用性災害時リカバリー (HADR) モニター情報を戻します。

### 構文

```
▶▶ MON_GET_HADR (—member—) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### member

メンバー番号を指定する、タイプ INTEGER の入力引数。戻される行は、このメンバーによって処理されているログ・ストリームを表します。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限

- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

### 例

- 例 1:

```
db2 SELECT HADR_ROLE, STANDBY_ID, HADR_STATE, varchar(PRIMARY_MEMBER_HOST ,20)
as PRIMARY_MEMBER_HOST, varchar(STANDBY_MEMBER_HOST ,20)
as STANDBY_MEMBER_HOST from table(MON_GET_HADR(NULL))
```

以下はこの照会の出力例です。

HADR_ROLE	STANDBY_ID	HADR_STATE	PRIMARY_MEMBER_HOST
PRIMARY	1	PEER	hostP.ibm.com
PRIMARY	2	REMOTE_CATCHUP	hostP.ibm.com
PRIMARY	3	REMOTE_CATCHUP	hostP.ibm.com

STANDBY_MEMBER_HOST
hostS1.ibm.com
hostS2.ibm.com
hostS3.ibm.com

3 record(s) selected.

3 つのスタンバイを持つ 1 次データベースに対して照会が発行され、3 つの行が戻されます。各行は、1 次とスタンバイとのログ配送チャネルを表します。HADR\_ROLE 列は、照会の発行先であるデータベースの役割を表します。したがって、すべての行で PRIMARY になります。

- 例 2:

```
db2 SELECT HADR_ROLE, STANDBY_ID, HADR_STATE, varchar(PRIMARY_MEMBER_HOST ,20)
as PRIMARY_MEMBER_HOST, varchar(STANDBY_MEMBER_HOST ,20)
as STANDBY_MEMBER_HOST from table(MON_GET_HADR(NULL))
```

以下はこの照会の出力例です。

HADR_ROLE	STANDBY_ID	HADR_STATE	PRIMARY_MEMBER_HOST
STANDBY	0	PEER	hostP.ibm.com

STANDBY_MEMBER_HOST
hostS1.ibm.com

1 record(s) selected.

スタンバイ中の読み取りが有効になっているスタンバイ・データベースに対して照会が発行されます。スタンバイは自分自身の 1 次だけを認識しています。スタンバイが複数スタンバイ・システムの一部であっても、戻されるのは 1 行だけです。スタンバイに対して照会が発行された場合、STANDBY\_ID は常にゼロです。

## 使用上の注意

### HADR ペア・ビュー

一部のフィールドは、1 次またはスタンバイにのみ適用されます。例えば、PEER\_WAIT\_LIMIT は 1 次にのみ適用され、STANDBY\_RECV\_BUF\_SIZE、STANDBY\_SPOOL\_LIMIT、READS\_ON\_STANDBY\_ENABLED はスタンバイにのみ適用されます。この種の情報が報告される時、ローカル・データベースではなく、現在その役割にあるデータベースが使用されます (リモート・データベースの場合があります)。例えば、スタンバイ・データベースで見られる PEER\_WAIT\_LIMIT は、(スタンバイが 1 次になるときだけ使用される) スタンバイ・データベースのローカル構成ではなく、1 次データベースで構成された値です。

### リモート・データベースについての情報

1 次とスタンバイは、ハートビート・メッセージを介してモニター情報を交換します。そのため、リモート・データベースについての情報が少し古くなる可能性があります。情報の適時性を推定するには、(表関数で報告される) ハートビート・インターバルを調べてください (ネットワーク待ち時間のためにさらに遅延が加わることがあります)。アクティブ化以降、データベースが一度もそのパートナー・データベースに接続されていない場合、リモート・データベースについての情報は、「不明」を示す SQL NULL として戻されることがあります。

### ログ配送チャネルのエンドポイント

ログ配送チャネルのエンドポイントは、以下のように、ホスト、インスタンス、およびメンバーで一意的に識別されます。

- 1 次側: PRIMARY\_MEMBER\_HOST、PRIMARY\_INSTANCE、PRIMARY\_MEMBER
- スタンバイ側: STANDBY\_MEMBER\_HOST、STANDBY\_INSTANCE、STANDBY\_MEMBER

接続が確立されるまでは、リモート・エンドのエンドポイント情報が使用不可になる場合があります。情報が使用不可のときは、ホストおよびインスタンスの名前として空ストリングが戻され、メンバー ID としてゼロが戻されます。また、DB2 Enterprise Server Edition 環境の場合は、メンバー ID として必ず 0 が返されます。

### 時刻期間の単位に関する注記

モニター表関数の規則に従い、MON\_GET\_HADR の時刻期間フィールドではすべて、ミリ秒が単位として使われます。構成の単位が秒である構成パラメーター (HADR\_TIMEOUT、HADR\_PEER\_WINDOW など) を反映するフィールドの場合、MON\_GET\_HADR 表関数で戻される数値は、**db2 get/update db cfg** コマンドで使われる数値および SYSIBMADM.DBCFG 管理ビューや SYSPROC.DB\_GET\_CFG() 表関数で戻される数値とは異なります。例えば HADR\_TIMEOUT 値が 60 秒である場合、MON\_GET\_HADR は 60000 を戻しますが、構成指向のインターフェースでは 60 が戻されます。ミリ秒の数値を秒に変換するには、照会で column\_name/1000 を使用してください。

### テークオーバー中の使用



テークオーバー中は、クライアントが 1 次データベースにもスタンバイ・データベースにも接続できない期間が発生する可能性があります。テークオーバー中の推奨されるモニター方式は **db2pd -hadr** です。

#### 列の順序とグループ:

1. クラスタ・レベルのサマリー:  
HADR\_ROLE、REPLAY\_TYPE、HADR\_SYNCMODE
2. ログ・ストリーム・レベルのサマリー:  
STANDBY\_ID、LOG\_STREAM\_ID、HADR\_STATE
3. ログ配送チャネルのエンドポイント:
  - a. 1 次側: PRIMARY\_MEMBER\_HOST、 PRIMARY\_INSTANCE、  
PRIMARY\_MEMBER
  - b. スタンバイ側: STANDBY\_MEMBER\_HOST、 STANDBY\_INSTANCE、  
STANDBY\_MEMBER

エンドポイントは、すべてのシナリオで HADR ログ配送チャネルを一意的に識別します。ホスト、インスタンス、または MEMBER\_ID により、メンバーが一意的に識別されます。

4. 接続の詳細:
  - a. 状況: HADR\_CONNECT\_STATUS、HADR\_CONNECT\_STATUS\_TIME
  - b. ネットワーク・タイミグ: HEARTBEAT\_INTERVAL、 HADR\_TIMEOUT、  
TIME\_SINCE\_LAST\_RECV
  - c. ロガー待機タイミグ: PEER\_WAIT\_LIMIT、 LOG\_HADR\_WAIT\_CUR、  
LOG\_HADR\_WAIT\_TIME、 LOG\_HADR\_WAITS\_TOTAL
  - d. TCP バッファ・サイズ: SOCK\_SEND\_BUF\_REQUESTED、  
SOCK\_SEND\_BUF\_ACTUAL、 SOCK\_RECV\_BUF\_REQUESTED、  
SOCK\_RECV\_BUF\_ACTUAL
5. ログ位置の詳細:
  - a. 1 次ログ位置: PRIMARY\_LOG\_FILE、 PRIMARY\_LOG\_PAGE、  
PRIMARY\_LOG\_POS、 PRIMARY\_LOG\_TIME
  - b. スタンバイ・ログ受信位置: STANDBY\_LOG\_FILE、  
STANDBY\_LOG\_PAGE、 STANDBY\_LOG\_POS、 STANDBY\_LOG\_TIME
  - c. 1 次とスタンバイのログのギャップ: HADR\_LOG\_GAP
  - d. スタンバイ・ログの適用位置: STANDBY\_REPLAY\_LOG\_FILE、  
STANDBY\_REPLAY\_LOG\_PAGE、 STANDBY\_REPLAY\_LOG\_POS、  
STANDBY\_REPLAY\_LOG\_TIME
  - e. スタンバイの受信と適用のギャップ: STANDBY\_RECV\_REPLAY\_GAP
  - f. 適用の遅延: STANDBY\_REPLAY\_DELAY
6. ログ・バッファおよびスプーリング: STANDBY\_RECV\_BUF\_SIZE、  
STANDBY\_RECV\_BUF\_PERCENT、 STANDBY\_SPOOL\_LIMIT
7. ピア・ウィンドウ: PEER\_WINDOW、 PEER\_WINDOW\_END
8. テークオーバー: TAKEOVER\_APP\_REMAINING\_PRIMARY、  
TAKEOVER\_APP\_REMAINING\_STANDBY

9. スタンバイ中の読み取り: READS\_ON\_STANDBY\_ENABLED、  
STANDBY\_REPLAY\_ONLY\_WINDOW\_ACTIVE、  
STANDBY\_REPLAY\_ONLY\_WINDOW\_START、  
STANDBY\_REPLAY\_ONLY\_WINDOW\_TRAN\_COUNT

### 戻される情報

表 144. MON\_GET\_HADR で戻される情報

列名	データ・タイプ	説明
HADR_ROLE	VARCHAR(13)	<p>hadr_role - HADR の役割モニター・エレメント</p> <p>このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• PRIMARY</li> <li>• STANDBY</li> </ul>
REPLAY_TYPE	VARCHAR(9)	<p>HADR レプリケーションのタイプ。</p> <ul style="list-style-type: none"> <li>• PHYSICAL</li> </ul>
HADR_SYNCMODE	VARCHAR(10)	<p>hadr_syncmode - HADR 同期モード・モニター・エレメント</p> <p>このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• ASYNC</li> <li>• STANDBY</li> <li>• SYNC</li> <li>• SUPERASYNC</li> </ul>
STANDBY_ID	SMALLINT	<p>スタンバイ ID。この ID はシステムによって生成されず。ID とデータベースとのマッピングは照会によって変わる場合があります。照会が 1 次に対して発行される場合、戻されるデータでは、そのすべてのスタンバイが表されます。この ID は、各スタンバイを区別するのに使用されます。これはシステムにより生成され、ID とスタンバイのマッピングは照会によって変わる場合があります。ただし、ID 「1」は常にプリンシパル・スタンバイ (または単一スタンバイ・システムでは唯一のスタンバイ) に割り当てられます。照会がスタンバイ・データベースに対して発行される場合、他のスタンバイは不可視であるため、常に 0 が戻されます。</p>
LOG_STREAM_ID	INTEGER	<p>配送されているログ・ストリームを識別します。ソース・データベース上のストリーム ID が戻されます。</p>

表 144. MON\_GET\_HADR で戻される情報 (続き)

列名	データ・タイプ	説明
HADR_STATE	VARCHAR(23)	<p>hadr_state - HADR の状態モニター・エレメント</p> <p>このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• DISCONNECTED</li> <li>• LOCAL_CATCHUP</li> <li>• REMOTE_CATCHUP_PENDING</li> <li>• REMOTE_CATCHUP</li> <li>• PEER</li> <li>• DISCONNECTED_PEER</li> </ul>
PRIMARY_MEMBER_HOST	VARCHAR(255)	ログ・ストリームを処理している 1 次メンバーの HADR_LOCAL_HOST。
PRIMARY_INSTANCE	VARCHAR(128)	ログ・ストリームを処理している 1 次メンバーのインスタンス名。
PRIMARY_MEMBER	SMALLINT	ログ・ストリームを処理している 1 次メンバー。
STANDBY_MEMBER_HOST	VARCHAR(255)	ログ・ストリームを処理しているスタンバイ・メンバーの HADR_LOCAL_HOST。
STANDBY_INSTANCE	VARCHAR(128)	ログ・ストリームを処理しているスタンバイ・メンバーのインスタンス名。
STANDBY_MEMBER	SMALLINT	ログ・ストリームを処理しているスタンバイ・メンバー。
HADR_CONNECT_STATUS	VARCHAR(12)	hadr_connect_status - HADR 接続状況モニター・エレメント
HADR_CONNECT_STATUS_TIME	TIMESTAMP	CONNECT_STATUS に応じて、接続開始時刻、輻輳開始時刻、切断時刻のいずれかの値を示します。
HEARTBEAT_INTERVAL	BIGINT	ハートビート・インターバル。インターバルは、HADR_TIMEOUT、PEER_WINDOW などのさまざまな要因から計算されます。このエレメントは、1 次とスタンバイがモニター情報を交換する頻度を示しています。単位はミリ秒です。
HADR_TIMEOUT	BIGINT	hadr_timeout - HADR タイムアウト・モニター・エレメント
TIME_SINCE_LAST_RECV	BIGINT	最後のメッセージが受信された以降の経過時間。この数値は通常、HEARTBEAT_INTERVAL 以下です。チャンネルがアイドル状態であっても、HADR データベースはハートビート・インターバルでハートビート・メッセージを送信するからです。これよりも大きい数値は、メッセージ配信の遅延を示しています。この数値が HADR_TIMEOUT に達すると、接続は閉じられます。単位はミリ秒です。
PEER_WAIT_LIMIT	BIGINT	(レジストリー変数 DB2_HADR_PEER_WAIT_LIMIT を介して設定される) ピア待機限度構成を反映します。単位はミリ秒です。

表 144. MON\_GET\_HADR で戻される情報 (続き)

列名	データ・タイプ	説明
LOG_HADR_WAIT_CUR	BIGINT	HADR ログ配送要求での現在のロガー待機時間。ロガーが待機していない場合は 0 を戻します。待機時間がピア待機限度に達すると、HADR はピア状態から抜け出して 1 次データベースのブロックを解除します。ロガーが完全にブロックされている場合、LOG_HADR_WAIT_CUR と LOG_HADR_WAIT_TIME はリアルタイムで増えていきますが、LOG_HADR_WAITS_TOTAL は同じままです。単位はミリ秒です。
LOG_HADR_WAIT_TIME	BIGINT	HADR によるログの配送をロガーが待機するのに費やした累積時間。LOG_HADR_WAIT_TIME と LOG_HADR_WAITS_TOTAL を使用すると、任意のインターバルにおけるログ・フラッシュ当たりの平均 HADR 待機時間を計算できます。また、この 2 つのフィールドは表関数 MON_GET_TRANSACTION_LOG によっても報告されます。単位はミリ秒です。
LOG_HADR_WAITS_TOTAL	BIGINT	ロガーでの HADR 待機イベントの合計カウント。ロガーが HADR ログ配送の待機を開始するたびに (待機が直ちに帰ったとしても)、カウントが増えます。従って、このカウントは事実上、ピア状態でのログ・フラッシュの数です。LOG_HADR_WAIT_TIME と LOG_HADR_WAITS_TOTAL を使用すると、任意のインターバルにおけるログ・フラッシュ当たりの平均 HADR 待機時間を計算できます。また、この 2 つのフィールドは表関数 MON_GET_TRANSACTION_LOG によっても報告されます。
SOCK_SEND_BUF_REQUESTED	BIGINT	要求されたソケット送信バッファ・サイズのバイト数 (レジストリー変数 DB2_HADR_SOSNDBUF)。要求がない場合、値は 0 です (システム・デフォルトを使用)。
SOCK_SEND_BUF_ACTUAL	BIGINT	実際のソケット送信バッファ・サイズのバイト数。要求されたサイズとは異なる場合があります。
SOCK_RECV_BUF_REQUESTED	BIGINT	要求されたソケット受信バッファ・サイズのバイト数 (レジストリー変数 DB2_HADR_SORCVBUF)。要求がない場合、値は 0 です (システム・デフォルトを使用)。
SOCK_RECV_BUF_ACTUAL	BIGINT	実際のソケット受信バッファ・サイズのバイト数。要求されたサイズとは異なる場合があります。
PRIMARY_LOG_FILE	VARCHAR(12)	1 次 HADR データベース上のこのログ・ストリームでの現行ログ・ファイルの名前。
PRIMARY_LOG_PAGE	BIGINT	1 次 HADR データベース上の現在のログ位置に対応する、現行ログ・ファイル内のページ番号。ページ番号はログ・ファイルと相対的です。例えば、ページ・ゼロはファイルの先頭です。
PRIMARY_LOG_POS	BIGINT	1 次 HADR データベース上のこのログ・ストリームでの現在のログ位置。これはバイト・オフセットです。
PRIMARY_LOG_TIME	TIMESTAMP	1 次 HADR データベース上のこのログ・ストリームでの最新のトランザクション・タイム・スタンプ。

表 144. MON\_GET\_HADR で戻される情報 (続き)

列名	データ・タイプ	説明
STANDBY_LOG_FILE	VARCHAR(12)	このログ・ストリームでのスタンバイ受信ログ位置に対応するログ・ファイルの名前。
STANDBY_LOG_PAGE	BIGINT	スタンバイ受信ログ位置に対応する STANDBY_LOG_FILE 内のページ番号。ページ番号はログ・ファイルと相対的です。例えば、ページ・ゼロはファイルの先頭です。
STANDBY_LOG_POS	BIGINT	このログ・ストリームでのスタンバイ受信ログ位置。これはバイト・オフセットです。より詳細なスタンバイ状況が得られるように、受信位置と適用位置が別々に報告されます。スプーリングにより、受信位置と適用位置が大きく異なることが可能になります。STANDBY_LOG_POS は受信位置を示します。PRIMARY_LOG_POS と比較することで、STANDBY_LOG_POS はフェイルオーバー時のデータ損失リスクを示します。 STANDBY_REPLAY_LOG_POS は、テークオーバー (強制および非強制) に要する時間の長さに影響します。テークオーバーでは、受信したすべてのログの適用を完了する必要がありますがあるからです。また、STANDBY_REPLAY_LOG_POS は、スタンバイ上で読み取られるデータがどの程度新しいものであるかも示します。バージョン 9.7 以前では、報告されるスタンバイ・ログ位置が適用位置です。
STANDBY_LOG_TIME	TIMESTAMP	スタンバイ HADR データベース上のこのログ・ストリームでの受信ログの最新のトランザクション・タイム・スタンプ。
HADR_LOG_GAP	BIGINT	hadr_log_gap - HADR ログ・ギャップ・モニター・エレメント
STANDBY_REPLAY_LOG_FILE	VARCHAR(12)	このログ・ストリームでのスタンバイ適用ログ位置に対応するログ・ファイルの名前。
STANDBY_REPLAY_LOG_PAGE	BIGINT	スタンバイ適用ログ位置に対応する STANDBY_REPLAY_LOG_FILE 内のページ番号。ページ番号は、このログ・ファイル内での位置を示します。例えば、ページ・ゼロはファイルの先頭です。
STANDBY_REPLAY_LOG_POS	BIGINT	このログ・ストリームでのスタンバイ適用ログ位置。これはバイト・オフセットです。
STANDBY_REPLAY_LOG_TIME	TIMESTAMP	スタンバイ HADR データベースで適用されているログのトランザクション・タイム・スタンプ。

表 144. MON\_GET\_HADR で戻される情報 (続き)

列名	データ・タイプ	説明
STANDBY_RECV_REPLAY_GAP	BIGINT	このエレメントは、スタンバイ・ログ受信位置とスタンバイ・ログ適用位置とのギャップの最近の平均を示します。この差はバイト数で測定されます。一般に、これは STANDBY_RECV_BUF_SIZE と STANDBY_SPOOL_LIMIT の合計を超えません。柔軟なバッファ管理およびスプール管理により、その合計を少し超える可能性があります。バッファとスプールを合わせた限界にギャップが達すると、スタンバイはログの受信を停止し、ピア状態の 1 次がブロックされます。また、報告された受信/適用ギャップがバッファとスプールの合計より小さいときには、スタンバイでバッファとスプールのスペースが不足している可能性があります。部分ページが複数回にわたって送信され、バッファ内で複数ページのスペースを占有する可能性があるためです (ただしスプールでは常に 1 ページ)。しかし、ログ・ギャップ計算では複数の送信が考慮に入れられません。
STANDBY_REPLAY_DELAY	BIGINT	このエレメントは、スタンバイ・データベースでの HADR_REPLAY_DELAY 構成を反映します。単位はミリ秒です。
STANDBY_RECV_BUF_SIZE	BIGINT	スタンバイ受信バッファ・サイズ (ページ単位)。
STANDBY_RECV_BUF_PERCENT	DOUBLE	使用中のスタンバイ受信バッファのパーセンテージ。スプーリングが使用可能な場合は、受信バッファが満杯 (100% 使用) であっても、スタンバイはログを受信し続けることができます。
STANDBY_SPOOL_LIMIT	BIGINT	スプールする最大ページ数。スプーリングが使用不可の場合は 0、無制限の場合は -1。このエレメントは、スタンバイ・データベースでの HADR_SPOOL_LIMIT 構成を反映します。
PEER_WINDOW	BIGINT	<b>HADR_PEER_WINDOW</b> データベース構成パラメーター。単位はミリ秒です。
PEER_WINDOW_END	TIMESTAMP	現行ピア・ウィンドウの終了時刻。ピア・ウィンドウが使用不可の場合は NULL。
TAKEOVER_APP_REMAINING_PRIMARY	BIGINT	非強制テークオーバー時に、1 次上でまだ強制的に除外されていないアプリケーションの数。テークオーバー中でない場合は NULL を戻します。
TAKEOVER_APP_REMAINING_STANDBY	BIGINT	(強制および非強制) テークオーバー時に、スタンバイ上でまだ強制的に除外されていない、スタンバイ中の読み取りアプリケーションの数。テークオーバー中でない場合は NULL を戻します。
READS_ON_STANDBY_ENABLED	CHAR(1)	スタンバイ中の読み取り機能が使用可能になっているかどうかを検査します。レジストリー変数 DB2_HADR_ROS によって制御されます。以下のいずれか <ul style="list-style-type: none"> <li>• Y (はい)</li> <li>• N (いいえ)</li> </ul>

表 144. MON\_GET\_HADR で戻される情報 (続き)

列名	データ・タイプ	説明
STANDBY_REPLAY_ONLY_WINDOW_ACTIVE	CHAR(1)	適用専用ウィンドウの状況。値は以下のとおりです。 <ul style="list-style-type: none"> <li>• Y: アクティブ</li> <li>• N: 非アクティブ</li> </ul>
STANDBY_REPLAY_ONLY_WINDOW_START	TIMESTAMP	適用専用ウィンドウの開始時刻。
STANDBY_REPLAY_ONLY_WINDOW_TRAN_COUNT	BIGINT	現行の適用専用ウィンドウで現在までに実行された非コミット DDL または保守トランザクションの総数。

## MON\_GET\_INDEX 表関数 - 索引メトリックの取得

MON\_GET\_INDEX 表関数は、1 つ以上の索引のメトリックを戻します。

### 構文

```
▶▶ MON_GET_INDEX ( (—tabschema—, —tablename—, —member—) ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *tabschema*

この関数を呼び出すときに現在接続されているデータベースと同じデータベース内の有効な表スキーマ名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内の全スキーマの表の索引についてメトリックが取得されます。引数が指定されている場合、指定したスキーマ内の表の索引についてのメトリックのみが戻されます。

#### *tablename*

この関数を呼び出すときに現在接続されているデータベースと同じデータベース内の有効な表名を指定する、タイプ VARCHAR(128) の入力引数。指定される表のすべての索引についてのメトリックが戻されます。引数が NULL または空ストリングである場合、データベース内のすべての表のすべての索引についてメトリックが取得されます。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

### 例

最後にデータベースがアクティブ化されてから、DMEXT002.TABLE1 表の索引で最も頻繁に使用された索引を識別します。

```
SELECT VARCHAR(S.INDSCHEMA, 10) AS INDSCHEMA,  
       VARCHAR(S.INDNAME, 10) AS INDNAME,  
       T.DATA_PARTITION_ID,  
       T.MEMBER,  
       T.INDEX_SCANS,  
       T.INDEX_ONLY_SCANS  
FROM TABLE(MON_GET_INDEX('DMEXT002','TABLE1', -2)) as T, SYSCAT.INDEXES AS S  
WHERE T.TABSCHEMA = S.TABSCHEMA AND  
       T.TABNAME = S.TABNAME AND  
       T.IID = S.IID  
ORDER BY INDEX_SCANS DESC
```

以下はこの照会の出力例です。

INDSCHEMA	INDNAME	DATA_PARTITION_ID	MEMBER	INDEX_SCANS	INDEX_ONLY_SCANS
DMEXT002	INDEX3	-	-	0	1
DMEXT002	INDEX4	-	-	0	1
DMEXT002	INDEX1	-	-	0	0
DMEXT002	INDEX2	-	-	0	0
DMEXT002	INDEX5	-	-	0	0
DMEXT002	INDEX6	-	-	0	0

6 record(s) selected.

### 使用上の注意

MON\_GET\_INDEX 表関数は、索引およびデータベース・メンバーごとに 1 行のデータを戻します。パーティション化索引が使用されている場合、データベース・メンバーの索引パーティションごとに 1 つの行が戻されます。データベース・メンバー全体からの集約は実行されません。ただし、集約は上記の例に示されるように SQL 照会を使用して実行できます。

データベースがアクティブにされた後にアクセスのあった表の索引についてのみ、メトリックが戻されます。すべてのカウンターは、現在のデータベースがアクティブにされた後のデータを表します。例えば、*pseudo\_empty\_pages* カウンターは、データベースがアクティブにされた後に疑似空白として識別されたページ数です。これは、索引内の現在の疑似空ページ数ではありません。

この関数によって収集されるメトリックは、*mon\_obj\_metrics* 構成パラメーターを使用してデータベース・レベルで制御されます。デフォルトでは、メトリック収集は有効になります。

### 戻される情報

表 145. MON\_GET\_INDEX について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TABNAME	VARCHAR(128)	table_name - 表名



表 145. MON\_GET\_INDEX について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
IID	SMALLINT	iid - 索引 ID
MEMBER	SMALLINT	member - データベース・メンバー
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID
NLEAF	BIGINT	nleaf - リーフ・ページ数
NLEVELS	SMALLINT	nlevels - 索引レベルの数
INDEX_SCANS	BIGINT	index_scans - 索引スキャン
INDEX_ONLY_SCANS	BIGINT	index_only_scans - 索引のみのスキャン
KEY_UPDATES	BIGINT	key_updates - キー更新
INCLUDE_COL_UPDATES	BIGINT	include_col_updates - 列更新の組み込み
PSEUDO_DELETES	BIGINT	pseudo_deletes - 疑似削除
DEL_KEYS_CLEANED	BIGINT	del_keys_cleaned - 疑似削除されたキーのクリーン
ROOT_NODE_SPLITS	BIGINT	root_node_splits - ルート・ノード分割
INT_NODE_SPLITS	BIGINT	int_node_splits - 中間ノード分割
BOUNDARY_LEAF_NODE_SPLITS	BIGINT	boundary_leaf_node_splits - 境界リーフ・ノード分割
NONBOUNDARY_LEAF_NODE_SPLITS	BIGINT	nonboundary_leaf_node_splits - 非境界リーフ・ノード分割
PAGE_ALLOCATIONS	BIGINT	page_allocations - ページ割り振り
PSEUDO_EMPTY_PAGES	BIGINT	pseudo_empty_pages - 疑似空ページ
EMPTY_PAGES_REUSED	BIGINT	empty_pages_reused - 再利用された空ページ
EMPTY_PAGES_DELETED	BIGINT	empty_pages_deleted - 削除された空ページ
PAGES_MERGED	BIGINT	pages_merged - マージされたページ
OBJECT_INDEX_L_READS	BIGINT	object_index_l_reads - 索引のバッファ・プール索引論理読み取り
OBJECT_INDEX_P_READS	BIGINT	object_index_p_reads - 索引のバッファ・プール索引物理読み取り
OBJECT_INDEX_GBP_L_READS	BIGINT	object_index_gbp_l_reads - 索引のグループ・バッファ・プール索引論理読み取り
OBJECT_INDEX_GBP_P_READS	BIGINT	object_index_gbp_p_reads - 索引のグループ・バッファ・プール索引物理読み取り
OBJECT_INDEX_GBP_INVALID_PAGES	BIGINT	object_index_gbp_invalid_pages - 索引に関するグループ・バッファ・プールの無効な索引ページ
OBJECT_INDEX_LBP_PAGES_FOUND	BIGINT	object_index_lbp_pages_found - 索引に関して検出されたローカル・バッファ・プールの索引ページ
OBJECT_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	object_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ
INDEX_JUMP_SCANS	BIGINT	index_jump_scans - 索引ジャンプ・スキャン

## MON\_GET\_INDEX\_USAGE\_LIST 表関数 - 索引用使用量リストからの情報を戻す

MON\_GET\_INDEX\_USAGE\_LIST 表関数は、索引に対して定義された使用量リストからの情報を戻します。

### 構文

```
▶▶MON_GET_INDEX_USAGE_LIST(—usagelistschema—,—usagelistname—,—member—)▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *usagelistschema*

この関数を呼び出すときに現在接続されているデータベース内の有効なスキーマ名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのスキーマの使用量リストが取得されます。引数を指定した場合は、指定したスキーマの使用量リストのみが戻されます。デフォルトは NULL です。

#### *usagelistname*

この関数を呼び出すときに現在接続されているデータベース内の索引に対して定義された使用量リストを指定する、タイプ VARCHAR(128) の入力引数。

*usagelistname* が NULL または空ストリングである場合、*usagelistschema* で識別されるスキーマにある索引に対して定義された、存在するすべての使用量リストが取得されます。これを指定した場合は、*usagelistschema* で識別されるスキーマの、指定した使用量リストのみが戻されます。デフォルトは NULL です。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

すべてのデータベース・メンバーから使用量リスト USL\_MON\_IND1 を戻します。

```

SELECT * FROM TABLE(
MON_GET_INDEX_USAGE_LIST(NULL, 'USL_MON_IND1', -2))

USAGELISTSHEMA      USAGELISTNAME      INDSHEMA      INDNAME      MEMBER
-----
ISAYYID              USL_MON_IND1      ISAYYID      I1            0
ISAYYID              USL_MON_IND1      ISAYYID      I1            0

DATA_PARTITION_ID
-----
-
-

EXECUTABLE_ID
-----
x'0100000000000000490000000000000000000000000020020110706093802577065'
x'01000000000000004B0000000000000000000000000020020110706093825981548'

MON_INTERVAL_ID      LAST_UPDATED      NUM_REFERENCES
-----
02011-07-06-09.38.15.881668      1
02011-07-06-09.38.25.984147      1

NUM_REF_WITH_METRICS OBJECT_INDEX_L_READS OBJECT_INDEX_P_READS
-----
1 1 0
1 1 0

OBJECT_INDEX_GBP_L_READS OBJECT_INDEX_GBP_P_READS
-----
0 0
0 0

OBJECT_INDEX_GBP_INVALID_PAGES OBJECT_INDEX_LBP_PAGES_FOUND
-----
0 0
0 0

```

2 record(s) selected.

## 使用上の注意

この関数によって戻される各行は、リストに追加されて以降、特定の時間間隔 (モニター・インターバル ID) 中に固有のセクション (DML ステートメントのみ、実行可能 ID) が特定のオブジェクトを参照した合計回数 (num\_references) を表します。この行に関して収集された統計は、その時間間隔中のこれらの実行にわたる総計値を表します。

平均を計算するときには、num\_references 列ではなく num\_ref\_with\_metrics 列を使用してください。報告されるメトリックにセクション実行が寄与したかどうかにかかわらず、num\_references 列ではセクションのすべての実行がカウントされるためです。

この関数によって収集されるメトリックは、mon\_obj\_metrics 構成パラメーターを使用してデータベース・レベルで制御されます。デフォルトでは、メトリック収集は有効になります。

## 戻される情報

表 146. MON\_GET\_INDEX\_USAGE\_LIST で戻される情報

列名	データ・タイプ	説明
USAGELISTSHEMA	VARCHAR (128)	usage_list_schema - 使用量リスト・スキーマ
USAGELISTNAME	VARCHAR (128)	usage_list_name - 使用量リスト名
INDSCHEMA	VARCHAR (128)	index_schema - 索引スキーマ
INDNAME	VARCHAR (128)	index_name - 索引名
MEMBER	SMALLINT	member - データベース・メンバー
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID
EXECUTABLE_ID	VARCHAR (32) FOR BIT DATA	executable_id - 実行可能 ID
MON_INTERVAL_ID	BIGINT	mon_interval_id - モニター・インターバル ID
LAST_UPDATED	TIMESTAMP	last_updated - 最終更新タイム・スタンプ
NUM_REFERENCES	BIGINT	num_references - 参照の数
NUM_REF_WITH_METRICS	BIGINT	num_ref_with_metrics - メトリックに関連した参照の数
OBJECT_INDEX_L_READS	BIGINT	object_index_l_reads - 索引のバッファ・プール索引論理読み取り
OBJECT_INDEX_P_READS	BIGINT	object_index_p_reads - 索引のバッファ・プール索引物理読み取り
OBJECT_INDEX_GBP_L_READS	BIGINT	object_index_gbp_l_reads - 索引のグループ・バッファ・プール索引論理読み取り
OBJECT_INDEX_GBP_P_READS	BIGINT	object_index_gbp_p_reads - 索引のグループ・バッファ・プール索引物理読み取り
OBJECT_INDEX_GBP_INVALID_PAGES	BIGINT	object_index_gbp_invalid_pages - 索引に関するグループ・バッファ・プールの無効な索引ページ
OBJECT_INDEX_LBP_PAGES_FOUND	BIGINT	object_index_lbp_pages_found - 索引に関して検出されたローカル・バッファ・プールの索引ページ
OBJECT_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	object_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ

## MON\_GET\_LOCKS - 現在接続されているデータベース内のすべてのロックのリスト

MON\_GET\_LOCKS 表関数は、現在接続されているデータベース内のすべてのロックのリストを戻します。

ロックについての情報を取得するには

MON\_GET\_LOCKS、MON\_FORMAT\_LOCK\_NAME、MON\_GET\_APPL\_LOCKWAIT 表関数と MON\_LOCKWAIT 管理ビューを使用してください。SNAPLOCKWAIT 管理ビューと SNAP\_GET\_LOCKWAIT 表関数、SNAPLOCK 管理ビューと SNAP\_GET\_LOCK 表関数、および LOCKS\_HELD 管理ビューはバージョン 9.7 フィックスパック 1 で非推奨になりました。

▶▶MON\_GET\_LOCKS(—search\_args—,—member—)◀◀

スキーマは SYSPROC です。

## 表関数パラメーター

### *search\_args*

*key-value* の組のリストを表すタイプ CLOB(1K) の入力パラメーター。リストが空または NULL の場合には、現在接続されているデータベース内のすべてのロックが戻されます。それ以外の場合は、*key-value* の組のリストによって表されるすべての条件に一致するすべてのロックが戻されます。*key-value* の組は以下の形式に従う必要があります。

- *key* は、開始タグ、値、終了タグの順序で構成される文字列です。
- 開始タグは、タグの始まりを示す不等号括弧、キー名、タグの終わりを示す不等号括弧の順序で構成されます。スペースは使用できません。
- 終了タグは、タグの始まりを示す不等号括弧、スラッシュ、キー名、タグの終わりを示す不等号括弧の順序で構成されます。スペースは使用できません。
- すべてのキーは大文字と小文字を区別し、*search\_args* パラメーター内で一度だけ指定できます。
- 複数のキーの順序は重要ではありません。

無効な *key-value* の組に対しては SQLCODE -171 が戻されます。

表が存在しない場合には SQLCODE -204 が戻されます。

複数の異なるキーの間で AND 演算が実行されます。同じキーの複数の値の間では OR 演算が実行されます。例えば以下のように *search\_args* パラメーターを使用した場合に戻されるリストには、共有モードまたは排他モードでハンドル 123 のアプリケーションによって保持されている (または取得待機中の) 表タイプまたは行タイプのすべてのロックが含まれます。

```
CLOB('<application_handle>123</application_handle>  
      <lock_object_type>Table:Row</lock_object_type>  
      <lock_mode>S:X</lock_mode>')
```

MON\_GET\_LOCKS 表関数で使用できるキーは次のとおりです。

- *application\_handle*

指定されたアプリケーション・ハンドルによって現在保持されている、または取得されようとしているすべてのロックのリストを戻します。キー値の単一オカレンスのみ指定できます。値は INTEGER として指定します。例:

```
CLOB('<application_handle>145</application_handle>')
```

- *lock\_name*

指定されたロック名に一致するすべてのロックのリストを戻します。キー値の単一オカレンスのみ指定できます。値は、最大長 32 の文字列として指定します。例:

```
CLOB('<lock_name>0003000500000000280000452</lock_name>')
```

- *lock\_object\_type*

指定されたロック対象タイプに一致するすべてのロックのリストを戻します。キー値の複数のオカレンスを指定できます (最大で 5 回まで)。それぞれ

の値 (大/小文字を区別しない) をコロン (:) で区切る必要があります。値は最大長 32 文字のストリングとして指定します。例:

```
CLOB('<lock_object_type>Table:Chunk:Plan</lock_object_type>')
```

可能な入力値のリストについては、『lock\_object\_type - 待機中のロック対象タイプ』モニター・エレメントを参照してください。

- lock\_mode

指定されたロック・モードに一致するすべてのロックのリストを戻します。キー値の複数のオカレンスを指定できます (最大で 5 回まで)。それぞれの値 (大/小文字を区別しない) をコロン (:) で区切ります。値は最大長 3 のストリングとして指定します。例えば、

```
CLOB('<lock_mode>IS:IN:U</lock_mode>')
```

可能な入力値のリストについては、『lock\_mode - ロック・モード・モニター・エレメント』を参照してください。

- lock\_status

指定された状況にあるすべてのロックのリストを戻します。キー値の単一オカレンスのみ指定できます。値は文字として指定します。

```
CLOB('<lock_status>W</lock_status>')
```

可能な入力値のリストについては、『lock\_status - ロック状況モニター・エレメント』を参照してください。

- table\_schema

指定されたスキーマ名によって限定されるすべてのロックのリストを戻します。さらに table\_name キーも指定する必要があります。キー値の単一オカレンスのみ指定できます。値は、最大長 128 のストリングとして指定します。

- table\_name

指定された表を参照するすべてのロックのリストを戻します。さらに table\_schema キーも指定する必要があります。キー値の単一オカレンスのみ指定できます。値は、最大長 128 のストリングとして指定します。例:

```
CLOB('<table_schema>USER1</table_schema>  
<table_name>INVENTORY</table_name>')
```

以下の例は、key-value の組を search\_args パラメーターで使用方法を示しています。

1. すべての ROW および TABLE ロックを検索するには、次のようにします。

```
CLOB('<lock_object_type>Table:Row</lock_object_type>')
```

2. 表 T1 を参照し、ユーザー USER1 によって作成された、アプリケーション・ハンドル 123 が保持している (または取得待機中の) すべてのロックを検索するには、次のようにします。

```
CLOB('<application_handle>123</application_handle>  
<table_schema>USER1</table_schema>  
<table_name>T1</table_name>')
```

- 共有モードで現在保持されているすべての TABLE、ROW、および BUFFERPOOL ロックを検索するには、次のようにします。

```
CLOB('<lock_mode>S</lock_mode>
      <lock_status>G</lock_status>
      <lock_object_type>Table:Row:Bufferpool</lock_object_type>')
```

#### member

どのメンバーのデータが戻されるかを指定する、タイプ INTEGER の入力引数。現在のメンバーの場合は -1、すべてのアクティブ・メンバーの場合は -2 を指定します。

## 許可

以下のいずれかの権限または特権が必要です。

- SYSADM 権限
- SYSMON 権限

## デフォルトの PUBLIC 特権

なし

## 例

この例のシナリオでは表関数 MON\_GET\_LOCKS および MON\_GET\_APPL\_LOCKWAIT を使用して、現在接続されているデータベース内の、すべてのメンバーに関するロック状態を調べます。

- 次のように MON\_GET\_APPL\_LOCKWAIT 表関数を呼び出して、現在接続されているデータベース内の、すべてのメンバーに関する取得待機中のすべてのロックを判別します。

```
SELECT lock_name,
       hld_member,
       lock_status,
       hld_application_handle FROM
TABLE (MON_GET_APPL_LOCKWAIT(NULL, -2))
```

この照会は、以下の出力を戻します。

LOCK_NAME	HLD_MEMBER	LOCK_STATUS	HLD_APPLICATION_HANDLE
00030005000000000280000452	-2	W	
00030005000000000280000452	-2	W	
00030005000000000280000452	-2	W	

3 record(s) selected.

HLD\_MEMBER が -2 であることを示すレコードは、ロック 0x00030005000000000280000452 がリモート・メンバーで保持されていることを示しています。

- 次のように MON\_GET\_LOCKS 表関数を呼び出してロックのホルダーを判別します。その際、検索引数としてロック名 0x00030005000000000280000452 を指定します。

```
SELECT lock_name,
       member,
       lock_status,
```

```

        application_handle FROM
TABLE (MON_GET_LOCKS(
        CLOB('<lock_name>0003000500000000280000452</lock_name>'),
        -2))

```

この照会は、以下の出力を戻します。

```

LOCK_NAME                MEMBER LOCK_STATUS APPLICATION_HANDLE
-----
0003000500000000280000452 0      W              12562
0003000500000000280000452 1      W              12562
0003000500000000280000452 2      G              65545
0003000500000000280000452 3      W              12562

```

4 record(s) selected.

ロックを保持しているアプリケーションについてさらに詳しく調べるには、  
WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES または  
WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 表関数を呼び出すことができます。

## 戻される情報

表 147. MON\_GET\_LOCKS 表関数によって戻される情報

列名	データ・タイプ	説明またはモニター・エレメント
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル  LOCK_STATUS 列が G である場合、これはロックを現在保持しているアプリケーションを表します。  LOCK_STATUS 列が W または C である場合、これはロックの取得を現在待機しているアプリケーションを表します。
MEMBER	SMALLINT	member - この行のデータの取得元となったデータベース・メンバー。
LOCK_NAME	VARCHAR(32)	lock_name - ロック名
LOCK_OBJECT_TYPE_ID	CHAR(1) FOR BIT DATA	将来の利用のために予約済み



表 147. MON\_GET\_LOCKS 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
LOCK_OBJECT_TYPE	VARCHAR(32)	lock_object_type - ロック対象タイプ  LOCK_STATUS 列が G である場合、これはアプリケーションが現在保持しているオブジェクトのタイプを表します。  LOCK_STATUS 列が W または C である場合、これはアプリケーションが取得を現在待機しているオブジェクトのタイプを表します。  可能な入力値については、『lock_object_type - 待機中のロック対象タイプ』モニター・エレメントを参照してください。
LOCK_MODE	VARCHAR(3)	lock_mode - ロック・モード  LOCK_STATUS 列が G である場合、これはアプリケーションがロックを現在保持しているモードを表します。  LOCK_STATUS 列が W または C である場合、これはアプリケーションがロックの取得を現在待機しているモードを表します。  モードが不明な場合、NULL 値がこの列に戻されます。
LOCK_CURRENT_MODE	VARCHAR(3)	lock_current_mode - 移行前の元のロック・モード  モードが不明な場合、NULL 値がこの列に戻されます。
LOCK_STATUS	CHAR(1)	lock_status - ロック状況
LOCK_ATTRIBUTES	CHAR(16)	lock_attributes - ロック属性
LOCK_RELEASE_FLAGS	CHAR(16)	lock_release_flags - ロック解放フラグ・モニター・エレメント
LOCK_RRIID	BIGINT	内部使用のために予約済み 内部使用のために予約済み

表 147. MON\_GET\_LOCKS 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
LOCK_COUNT	BIGINT	Lock_count モニター・エレメント
LOCK_HOLD_COUNT	BIGINT	lock_hold_count モニター・エレメント
TBSP_ID	BIGINT	tablespace_id - 表スペース ID  表スペースを参照しないロックの場合、NULL 値が戻されます。
TAB_FILE_ID	BIGINT	table_file_id - 表ファイル ID

## MON\_GET\_MEMORY\_POOL - メモリー・プール情報の取得

MON\_GET\_MEMORY\_POOL 表関数は、メモリー・セット内に含まれるメモリー・プールのメトリックを取得します。

### 構文

►►MON\_GET\_MEMORY\_POOL(—memory\_set\_type—,—db\_name—,—member—)◀◀

スキーマは SYSPROC です。

### 表関数パラメーター

#### memory\_set\_type

この関数を呼び出すときのメモリー・セットのタイプを指定する、タイプ VARCHAR(32) の入力引数。この引数が NULL または空ストリングである場合、インスタンス・レベルとデータベース・レベルのすべてのメモリー・セットについてメトリックが取得されます。その他の場合には、指定のメモリー・セットのメトリックが取得されます。

以下のパラメーター値が受け入れられます。

値	有効範囲	説明
DBMS	インスタンス	DB2 データベース・マネージャー (DBM) メモリー・セット
FMP	インスタンス	fenced モード・プロセス (FMP) メモリー・セット
PRIVATE	インスタンス	専用メモリー・セット
DATABASE	データベース	データベース・メモリー・セット
APPLICATION	データベース	アプリケーション・メモリー・セット

値	有効範囲	説明
FCM	ホスト - 1 つのインスタンスにつき、1 台のマシンに割り振られる FCM メモリー・セットは 1 つのみです。	高速コミュニケーション・マネージャー (FCM) メモリー・セット
NULL	すべて	インスタンス・レベルとデータベース・レベルのすべてのメモリー・セット

#### *db\_name*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。

データベースのディレクトリー項目のタイプは、LIST DATABASE DIRECTORY コマンドによって戻されるとおり、『INDIRECT』または『HOME』のいずれかでなければなりません。データベースはアクティブでなければなりません。また、現在接続されているデータベースからメトリックを取得するために、CURRENT\_SERVER 特殊レジスターを指定することもできます。このレジスター値には、別名ではなく、データベースの実際の名前が含まれます。

引数が NULL または空ストリングの場合、インスタンス内のアクティブなデータベースすべてからメトリックが取得されます。この入力引数が適用されるのは、データベース・レベルのメモリー・セットのみです。

#### *member*

どのメンバーのデータが戻されるかを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのアクティブ・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

例 1: 現在のインスタンスと現在接続されているデータベースに関するメモリー・セットのメトリックを取得します。

```
SELECT varchar(memory_set_type, 20) AS set_type,
       varchar(memory_pool_type, 20) AS pool_type,
       varchar(db_name, 20) AS dbname,
       memory_pool_used,
```

```

memory_pool_used_hwm
FROM TABLE(
MON_GET_MEMORY_POOL(NULL, CURRENT_SERVER, -2))

```

以下は、この照会の出力の例です。

SET_TYPE	POOL_TYPE	DBNAME	MEMORY_POOL_USED	MEMORY_POOL_HWM_USED
DBMS	FCM_LOCAL	-	0	0
DBMS	FCM_SESSION	-	2359296	2359296
DBMS	FCM_CHANNEL	-	589824	589824
DBMS	FCMBP	-	983040	983040
DBMS	FCM_CHANNEL	-	35520512	35520512
DBMS	MONITOR	-	458752	589824
DBMS	RESYNC	-	262144	262144
DBMS	OSS_TRACKER	-	7667712	7667712
DBMS	APM	-	13041664	13238272
DBMS	BSU	-	3932160	4390912
DBMS	KERNEL_CONTROL	-	3932160	4390912
DBMS	EDU	-	655360	655360
FMP	MISC	-	655360	655360
DATABASE	UTILITY	TESTDB	65536	65536
DATABASE	PACKAGE_CACHE	TESTDB	983040	983040
DATABASE	XMLCACHE	TESTDB	196608	196608
DATABASE	CAT_CACHE	TESTDB	458752	458752
DATABASE	BP	TESTDB	850132992	850132992
DATABASE	BP	TESTDB	655360	655360
APPLICATION	APPLICATION	TESTDB	393216	393216
APPLICATION	APPLICATION	TESTDB	262144	262144

21 record(s) selected

## 使用上の注意

パーティション・データベース環境の場合、高速コミュニケーション・マネージャー (FCM) メモリー・セットはホストごとに割り振られます。つまり、1 つホスト・マシン上のすべてのメンバーが、1 つのセットを共有します。

MON\_GET\_MEMORY\_POOL 関数は、それぞれのメンバーからデータを取得します。FCM メモリー・セットは 1 つのホスト上のすべてのメンバーで共有されるので、そのホスト上の各メンバーの FCM メモリーに関して報告されるメトリックは、同じ共有メモリー・セットについての情報を示します。このため、FCM メモリーのメトリックを調べる場合は、それぞれ固有のホストに関するデータを調べてください。複数のメンバーで使用するホストの場合、そのホスト上の 1 人のメンバーからのデータのみを使用してください。FCM メモリーのメトリックは、特定のホスト上のすべてのメンバーに関して集約した合計を表すためです。

## 戻される情報

表 148. MON\_GET\_MEMORY\_POOL について戻される情報

列名	データ・タイプ	説明
MEMBER	SMALLINT	member - データベース・メンバー
HOST_NAME	VARCHAR(255)	host_name - ホスト名
DB_NAME	VARCHAR(128)	db_name - データベース名
MEMORY_SET_TYPE	VARCHAR(32)	memory_set_type - メモリー・セット・タイプ。可能なタイプのリストについては、memory_set_type 入力パラメーターを参照してください。
MEMORY_POOL_TYPE	VARCHAR(32)	memory_pool_type - メモリー・プール・タイプ。

表 148. MON\_GET\_MEMORY\_POOL について戻される情報 (続き)

列名	データ・タイプ	説明
MEMORY_POOL_ID	BIGINT	memory_pool_id - メモリー・プール ID
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル。適用可能なのは、APPLICATION、STATISTICS、STATEMENT、SORT_PRIVATE の各メモリー・プール・タイプのみです。それ以外の場合、値は NULL です。
EDU_ID	BIGINT	edu_id - エンジン・ディスパッチ可能単位 ID。PRIVATE メモリー・セット・タイプから割り振られたメモリー・プールに対してのみ適用可能です。それ以外の場合、値は NULL です。
MEMORY_POOL_USED	BIGINT	memory_pool_used - 使用中のメモリー・プール量。値は KB 単位です。
MEMORY_POOL_USED_HWM	BIGINT	memory_pool_used_hwm - メモリー・プール最高水準点。値は KB 単位です。

## MON\_GET\_MEMORY\_SET - メモリー・セット情報の取得

MON\_GET\_MEMORY\_SET 表関数は、インスタンス・レベルと、インスタンス内のすべてのアクティブなデータベースの両方について、割り振り済みメモリー・セットからメトリックを取り出します。

### 構文

►►—MON\_GET\_MEMORY\_SET—(—memory\_set\_type—, —db\_name—, —member—)——►►

スキーマは SYSPROC です。

### 表関数パラメーター

#### memory\_set\_type

この関数を呼び出すときのメモリー・セットのタイプを指定する、タイプ VARCHAR(32) の入力引数。この引数が NULL または空ストリングである場合、インスタンス・レベルとデータベース・レベルのすべてのメモリー・セットについてメトリックが取得されます。その他の場合には、指定のメモリー・セットのメトリックが取得されます。

以下のパラメーター値が受け入れられます。

値	有効範囲	説明
DBMS	インスタンス	DB2 データベース・マネージャー (DBM) メモリー・セット
FMP	インスタンス	fenced モード・プロセス (FMP) メモリー・セット
PRIVATE	インスタンス	専用メモリー・セット
DATABASE	データベース	データベース・メモリー・セット
APPLICATION	データベース	アプリケーション・メモリー・セット

値	有効範囲	説明
FCM	ホスト - 1 つのインスタンスにつき、1 台のマシンに割り振られる FCM メモリー・セットは 1 つのみです。	高速コミュニケーション・マネージャー (FCM) メモリー・セット
NULL	すべて	インスタンス・レベルとデータベース・レベルのすべてのメモリー・セット

#### *db\_name*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。

データベースのディレクトリー項目のタイプは、LIST DATABASE DIRECTORY コマンドによって戻されるとおり、『INDIRECT』または『HOME』のいずれかでなければなりません。データベースはアクティブでなければなりません。また、現在接続されているデータベースからメトリックを取得するために、CURRENT\_SERVER 特殊レジスターを指定することもできます。このレジスター値には、別名ではなく、データベースの実際の名前が含まれます。

引数が NULL または空ストリングの場合、インスタンス内のアクティブなデータベースすべてからメトリックが取得されます。この入力引数が適用されるのは、データベース・レベルのメモリー・セットのみです。

#### *member*

どのメンバーのデータが戻されるかを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのアクティブ・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

例 1: 現在のインスタンスと現在接続されているデータベースに関するメモリー・セットのメトリックを取得します。

```
SELECT varchar(memory_set_type, 20) as set_type,
       varchar(db_name, 20) as dbname,
       memory_set_used,
```

```

memory_set_used_hwm
FROM TABLE(
MON_GET_MEMORY_SET(NULL, CURRENT_SERVER, -2))

```

以下は、この照会の出力の例です。

SET_TYPE	DBNAME	MEMORY_SET_USED	MEMORY_SET_HWM_USED
DBMS	-	86080	87360
FMP	-	0	704
PRIVATE	-	10624	16256
DATABASE	TESTDB	928000	928000
APPLICATION	TESTDB	1472	2752

5 record(s) selected

## 使用上の注意

パーティション・データベース環境の場合、高速コミュニケーション・マネージャー (FCM) メモリー・セットはホストごとに割り振られます。つまり、1 つホスト・マシン上のすべてのメンバーが、1 つのセットを共有します。

MON\_GET\_MEMORY\_SET 関数は、それぞれのメンバーからデータを取得します。FCM メモリー・セットは 1 つのホスト上のすべてのメンバーで共有されるので、そのホスト上の各メンバーの FCM メモリーに関して報告されるメトリックは、同じ共有メモリー・セットについての情報を示します。このため、FCM メモリーのメトリックを調べる場合は、それぞれ固有のホストに関するデータを調べてください。複数のメンバーで使用するホストの場合、そのホスト上の 1 人のメンバーからのデータのみを使用してください。FCM メモリーのメトリックは、特定のホスト上のすべてのメンバーに関して集約した合計を表すためです。

## 戻される情報

表 149. MON\_GET\_MEMORY\_SET について戻される情報

列名	データ・タイプ	説明
MEMBER	SMALLINT	member - データベース・メンバー
HOST_NAME	VARCHAR(255)	host_name - Host name
DB_NAME	VARCHAR(128)	db_name - データベース名
MEMORY_SET_TYPE	VARCHAR(32)	memory_set_type - メモリー・セット・タイプ。可能なタイプのリストについては、 <i>memory_set_type</i> 入力パラメーターを参照してください。
MEMORY_SET_ID	BIGINT	memory_set_id - メモリー・セット ID。
MEMORY_SET_SIZE	BIGINT	memory_set_size - メモリー・セット・サイズ。値は KB 単位です。
MEMORY_SET_COMMITTED	BIGINT	memory_set_committed - 現在コミットされているメモリー。値は KB 単位です。
MEMORY_SET_USED	BIGINT	memory_set_used - このセットで使用しているメモリー。値は KB 単位です。
MEMORY_SET_USED_HWM	BIGINT	memory_set_used_hwm - メモリー・セット最高水準点。値は KB 単位です。

## MON\_GET\_PAGE\_ACCESS\_INFO 表関数 - バッファ・プール・ページ待機情報の取得

MON\_GET\_PAGE\_ACCESS\_INFO 表関数は、指定された表のために待機されているバッファ・プール・ページについての情報を戻します。これはデータ共有インスタンスにのみ適用できます。

### 構文

```
▶▶ MON_GET_PAGE_ACCESS_INFO (—tabschema—, —tablename—, —member—) ▶▶▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *tabschema*

照会するデータベース・スキーマを指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、すべてのスキーマの情報が戻されます。

#### *tablename*

照会する表名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、すべての表の情報が戻されます。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバー番号を指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

以下のいずれかの権限が必要です。

- 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

この例は、現在接続されているメンバー上のスキーマ BASETAB に含まれるすべての表のページ再利用カウントを戻します。表 TABLE1 のページをアプリケーションが待機していることを示しています (この状態の原因として、例えば 2 つの異なるメンバーから同一ページ上の異なる行を更新する場合があります)。

```
SELECT SUBSTR(TABNAME,1,8) AS NAME,  
       SUBSTR(OBJTYPE,1,5) AS TYPE,  
       PAGE_RECLAIMS_X AS PGRCX,  
       PAGE_RECLAIMS_S AS PGRCS,  
       SPACEMAPPAGE_PAGE_RECLAIMS_X AS SMPPGRCX,
```



```

SPACEMAPPAGE_PAGE_RECLAIMS_S AS SMPPGRCS
FROM TABLE( MON_GET_PAGE_ACCESS_INFO('BASETAB', NULL, NULL) ) AS WAITMETRICS

```

```
ORDER BY NAME;
```

以下はこの照会の出力例です。

NAME	TYPE	PGRXCX	PGRCS	SMPPGRXCX	SMPPGRCS
TABLE1	TABLE	12641	320	72	17
TABLE1	INDEX	5042	78	7	0
TABLE2	TABLE	420	12	9	0
TABLE2	INDEX	7	0	0	0

### 使用上の注意

MON\_GET\_PAGE\_ACCESS\_INFO 表関数は、データが収集されるメンバー当たり、表ごとに 1 行または 2 行のデータを戻します。1 行のデータは、索引ページの情報を示しています。2 番目の行は、データ・ページの情報を示しています。複数のメンバーにわたる集約は行われません。

### 戻される情報

表 150. MON\_GET\_PAGE\_ACCESS\_INFO で戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TABNAME	VARCHAR(128)	table_name - 表名
OBJTYPE	VARCHAR(128)	objtype - オブジェクト・タイプ
MEMBER	SMALLINT	member - データベース・メンバー
PAGE_RECLAIMS_X	BIGINT	page_reclaims_x - ページ再利用の排他的アクセス
PAGE_RECLAIMS_S	BIGINT	page_reclaims_s - ページ再利用の共有アクセス
SPACEMAPPAGE_PAGE_RECLAIMS_X	BIGINT	spacemappage_page_reclaims_x - スペース・マップ・ページ再利用の排他的アクセス
SPACEMAPPAGE_PAGE_RECLAIMS_S	BIGINT	spacemappage_page_reclaims_s - スペース・マップ・ページ再利用の共有アクセス
PAGE_RECLAIMS_INITIATED_X	BIGINT	page_reclaims_initiated_x - 排他的アクセスで開始されたページ再利用
PAGE_RECLAIMS_INITIATED_S	BIGINT	page_reclaims_initiated_s - 共有アクセスで開始されたページ再利用
SPACEMAPPAGE_PAGE_RECLAIMS_INITIATED_X	BIGINT	spacemappage_reclaims_initiated_x - 排他的アクセスで開始されたスペース・マップ・ページ再利用
SPACEMAPPAGE_PAGE_RECLAIMS_INITIATED_S	BIGINT	spacemappage_reclaims_initiated_s - 共有アクセスで開始されたスペース・マップ・ページ再利用
RECLAIM_WAIT_TIME	BIGINT	reclaim_wait_time - 再利用の待機時間
SPACEMAPPAGE_RECLAIM_WAIT_TIME	BIGINT	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID

表 150. MON\_GET\_PAGE\_ACCESS\_INFO で戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
IID	SMALLINT	iid - 索引 ID。このモニター・エレメントが値を戻すのは、OBJTYPE が INDEX で、パーティション表の非パーティション・タイプの索引である場合だけです。それ以外の場合、このモニター・エレメントの値は NULL になります。

## MON\_GET\_PKG\_CACHE\_STMT 表関数 - パッケージ・キャッシュ内の SQL ステートメント・アクティビティ・メトリックの取得

MON\_GET\_PKG\_CACHE\_STMT 表関数は、データベース・パッケージ・キャッシュ内の静的 SQL ステートメントと動的 SQL ステートメントの両方のポイント・イン・タイム・ビューを戻します。

### 構文

```

▶▶MON_GET_PKG_CACHE_STMT(—section_type—, —————▶
▶—executable_id—, —search_args—, —member—)————▶

```

スキーマは SYSPROC です。

### 表関数パラメーター

#### section\_type

オプションの入力引数 (D または S)、タイプは CHAR(1)。戻されるステートメントの情報の種類を指定します。引数が NULL または空ストリングである場合、すべての SQL ステートメントについての情報が戻されます。大/小文字は区別されません。"D" は動的を表し、"S" は静的を表します。

#### executable\_id

オプションの入力引数、タイプは VARCHAR(32)。データベース・パッケージ・キャッシュの固有のセクションを指定するビット・データを表します。NULL 値を指定すると、すべての SQL ステートメントについての情報が戻されます。executable\_id が指定されている場合、section\_type 引数は無視されません。例えば、動的ステートメントで executable\_id が指定されている場合、section\_type が静的 ("S") として指定されていても、この表関数によって動的ステートメントの詳細が戻されます。

#### search\_args

オプションのタイプ CLOB(1K) の入力パラメーター。これを使用して 1 つ以上の検索引数ストリングをオプションで指定できます。例:

```
'<modified_within>5</modified_within><update_boundary_time>myPkgEvmon
</update_boundary_time>'
```

使用可能な検索引数タグは次のとおりです。

- '<modified\_within>X</modified\_within>'

現在までの X 分間にキャッシュに挿入された、または実行されたステートメント項目だけを戻します (X は正の整数値)。引数が指定されない場合、キャッシュ内のすべての項目が戻されます。

- '<update\_boundary\_time>evmon\_name</update\_boundary\_time>'

*evmon\_name* で指定されたパッケージ・キャッシュ・イベント・モニターに関して、イベント・モニター境界タイム・スタンプを現在時刻に更新します。このイベント・モニターの WHERE 節で出力基準として *where updated\_since\_boundary\_time* が指定される場合、今後にメトリックが更新されるパッケージ・キャッシュ項目だけが、パッケージ・キャッシュからの退去時にキャプチャーされます。指定されたパッケージ・キャッシュ・イベント・モニターがコマンド発行時にアクティブである場合にのみ、この操作が効果を及ぼします。

それぞれの入力引数は一度だけ指定できます。検索引数タグは小文字で指定する必要があります。

#### member

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER のオプション入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が設定されます。

#### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

#### デフォルトの PUBLIC 特権

なし

#### 例

平均 CPU 時間ごとに配列されているデータベース・パッケージ・キャッシュから、すべての動的 SQL ステートメントをリストします。

```
db2 SELECT MEMBER,
SECTION_TYPE ,
TOTAL_CPU_TIME/NUM_EXEC_WITH_METRICS as
AVG_CPU_TIME,EXECUTABLE_ID
FROM TABLE(MON_GET_PKG_CACHE_STMT ( 'D', NULL, NULL, -2)) as T
WHERE T.NUM_EXEC_WITH_METRICS <> 0 ORDER BY AVG_CPU_TIME
```

以下はこの照会の出力例です。

MEMBER	SECTION_TYPE	AVG_CPU_TIME	EXECUTABLE_ID
0 D		754	x'010000000000000007A000000000000000000000000020020081126171554951791'
0 D		2964	x'0100000000000000079000000000000000000000000020020081126171533551120'
0 D		5664	x'010000000000000007C000000000000000000000000020020081126171720728997'

```
0 D 5723 x'01000000000000007B0000000000000000000000000020020081126171657272914'
0 D 9762 x'01000000000000007D0000000000000000000000000020020081126172409987719'
```

5 record(s) selected.

注: コンパイル環境を作成して、ステートメント・テキスト (2 MB の大きさになることもある) をメンバー間で転送するには、長い時間がかかります。パッケージ・キャッシュからすべてのステートメントのリストを取り出すときのパフォーマンスを改善するため、`STMT_TEXT` および `COMP_ENV_DESC` 列を選択しないでください。

上記の出力例では、`executable_id` 列を使用して最もコストの高い (平均 CPU 時間の観点で) ステートメントに関する詳細情報を取得できます。

```
db2 SELECT STMT_TEXT FROM TABLE(MON_GET_PKG_CACHE_STMT
      (null, x'01000000000000007D0000000000000000000000000020020081126172409987719', null, -2))
```

```
STMT_TEXT
-----
SELECT * FROM EMPLOYEE
```

別の例として、ワークロード A に関連付けられた接続を持つ Alex という名前のユーザーを想定します。ワークロード A には `COLLECT ACTIVITY METRICS` セットがあります。別のユーザー Brent はワークロード B に関連付けられています。ワークロード B には、`NONE` に設定された `COLLECT ACTIVITY METRICS` があります。さらに、データベースの `mon_act_metrics` 構成パラメーターが `NONE` に設定されています。Brent が次の照会を実行すると、

```
SELECT count(*) FROM syscat.tables
```

すべてのメトリックが 0 として返され、`num_exec_with_metrics` の値も 0 になります。その後、Alex が同じステートメントを実行しますが、ステートメントの実行に対して今回はメトリックが収集され、`num_exec_with_metrics` が増分します。そのため、Brent と Alex がそのステートメントを実行した後、以下の照会の結果は、

```
SELECT num_executions, num_exec_with_metrics, SUBSTR(stmt_text,1,50) AS stmt_text
      FROM TABLE (MON_GET_PKG_CACHE_STMT('d', null, null, -1)) AS tf
      WHERE stmt_text LIKE 'SELECT count%'
```

`SELECT` ステートメントが 2 回実行され、そのうちの 1 回でアクティビティ・メトリックが収集されたことを示します。

```
NUM_EXECUTIONS NUM_EXEC_WITH_METRICS STMT_TEXT
-----
                2                      1 SELECT count(*) FROM syscat.tables
```

1 record(s) selected.

### 使用上の注意

`MON_GET_PKG_CACHE_STMT` 表関数は、データベース・パッケージ・キャッシュ内の静的 SQL ステートメントと動的 SQL ステートメントの両方のポイント・イン・タイム・ビューを戻します。これにより、特定の SQL ステートメントに対して集約されたメトリックを検討し、照会パフォーマンスが低下している理由を迅速に判別することができます。戻されるメトリックは、ステートメントのそれぞれの実行中に収集されたメトリックの集約になります。

さらに、他のステートメントとの相対関係で、個々のキャッシュ付きセクションの振る舞いを比較し、(実行コストの観点から) 最もコストの高いセクションまたはステートメントの識別に役立てることができます。

この関数でレポートされたアクティビティ・メトリックは、アクティビティの実行の終わりにデータベース・キャッシュにロールアップされます。

あらゆるステートメントの実行のメトリック収集は、ワークロードの `COLLECT ACTIVITY METRICS` 節を介して、またはデータベース・レベルでの `mon_act_metrics` データベース構成パラメーターを介して制御されます。アクティビティ・メトリックが使用可能になっているワークロードまたはデータベースに関連付けられた接続でステートメントがサブミットされた場合、そのステートメントの実行のメトリックのみが収集されます。 `MON_GET_PKG_CACHE_STMT` 関数によって戻される `num_exec_with_metrics` エレメントは、収集されたメトリックを持っていて、報告された集約メトリックに寄与したステートメントの実行数を示します。ステートメントのどの実行に対してもメトリックが収集されていない場合には、`num_exec_with_metrics` エレメントは 0 であり、すべてのメトリック値は 0 として戻されます。

### 戻される情報

表 151. `MON_GET_PKG_CACHE_STMT` について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー
SECTION_TYPE	CHAR(1)	section_type - セクション・タイプ標識
INSERT_TIMESTAMP	TIMESTAMP	insert_timestamp - ステートメント挿入タイム・スタンプ
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	executable_id - 実行可能 ID
PACKAGE_NAME	VARCHAR(128)	package_name - パッケージ名。この出力は、静的 SQL ステートメントでのみ有効です。ステートメントが動的である場合、NULL 値が戻されます。
PACKAGE_SCHEMA	VARCHAR(128)	package_schema - パッケージ・スキーマ。この出力は、静的 SQL ステートメントでのみ有効です。ステートメントが動的である場合、NULL 値が戻されます。
PACKAGE_VERSION_ID	VARCHAR(64)	package_version_id - パッケージ・バージョン。この出力は、静的 SQL ステートメントでのみ有効です。ステートメントが動的であるか、または静的ステートメントのパッケージ・バージョンを指定しなかった場合、NULL 値が戻されます。パッケージの作成時にパッケージのバージョン ID が指定されなかった場合、静的ステートメントに対して空ストリングが戻されます。
SECTION_NUMBER	BIGINT	section_number - セクション番号。ステートメントが動的である場合、NULL 値が戻されます。
EFFECTIVE_ISOLATION	CHAR(2)	effective_isolation - 有効な分離。これは、セクションで有効な分離値です。コンパイル時に最初に要求されたものとは異なる可能性があります。

表 151. MON\_GET\_PKG\_CACHE\_STMT について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
NUM_EXECUTIONS	BIGINT	num_executions - ステートメント実行回数
NUM_EXEC_WITH_METRICS	BIGINT	num_exec_with_metrics - メトリック収集を伴う実行数
PREP_TIME	BIGINT	prep_time - 準備時間。PREP_TIME は、動的 SQL ステートメントでのみ有効であることに注意してください。静的 SQL ステートメントの場合、PREP_TIME は 0 としてレポートされます。
TOTAL_ACT_TIME	BIGINT	total_act_time - 合計アクティビティー時間
TOTAL_ACT_WAIT_TIME	BIGINT	total_act_wait_time - 合計アクティビティー待機時間
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間
TOTAL_SECTION_SORT_TIME	BIGINT	total_section_sort_time - セクション・ソート時間合計
TOTAL_SECTION_SORT_PROC_TIME	BIGINT	total_section_sort_proc_time - セクション・ソート処理時間合計
TOTAL_SECTION_SORTS	BIGINT	total_section_sorts - セクション・ソート合計
LOCK_ESCALS	BIGINT	lock_escalations - ロック・エスカレーション数
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
ROWS_MODIFIED	BIGINT	rows_modified - 変更された行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_RETURNED	BIGINT	rows_returned - 戻り行数
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り

表 151. MON\_GET\_PKG\_CACHE\_STMT について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDA データの書き込み
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込み
TOTAL_SORTS	BIGINT	total_sorts - ソート合計
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - ポストしきい値ソート
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - ポスト共有しきい値ソート
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
WLM_QUEUE_TIME_TOTAL	BIGINT	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間
WLM_QUEUE_ASSIGNMENTS_TOTAL	BIGINT	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
DEADLOCKS	BIGINT	deadlocks - デッドロック検出数
FCM_RECV_VOLUME	BIGINT	fcm_recv_volume - FCM 受信ボリューム
FCM_RECVS_TOTAL	BIGINT	fcm_recvs_total - FCM 合計受信数
FCM_SEND_VOLUME	BIGINT	fcm_send_volume - FCM 送信ボリューム
FCM_SENDS_TOTAL	BIGINT	fcm_sends_total - FCM 合計送信数
FCM_RECV_WAIT_TIME	BIGINT	fcm_recv_wait_time - FCM 受信待機時間
FCM_SEND_WAIT_TIME	BIGINT	fcm_send_wait_time - FCM 送信待ち時間
LOCK_TIMEOUTS	BIGINT	lock_timeouts - ロック・タイムアウト数
LOG_BUFFER_WAIT_TIME	BIGINT	log_buffer_wait_time - ログ・バッファ待ち時間
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - フル・ログ・バッファの回数
LOG_DISK_WAIT_TIME	BIGINT	log_disk_wait_time - ログ・ディスク待機時間
LOG_DISK_WAITS_TOTAL	BIGINT	log_disk_waits_total - ログ・ディスク待機の合計
LAST_METRICS_UPDATE	TIMESTAMP	last_metrics_update - メトリックの最終更新タイム・スタンプ
NUM_COORD_EXEC	BIGINT	num_coord_exec - コーディネーター・エージェントによる実行数

表 151. MON\_GET\_PKG\_CACHE\_STMT について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
NUM_COORD_EXEC_WITH_METRICS	BIGINT	num_coord_exec_with_metrics - コーディネーター・エージェントによる実行数
VALID	CHAR(1)	valid - セクション妥当性検査標識
TOTAL_ROUTINE_TIME	BIGINT	total_routine_time - ルーチン時間の合計
TOTAL_ROUTINE_INVOCATIONS	BIGINT	total_routine_invocations - ルーチン呼び出しの合計回数
ROUTINE_ID	BIGINT	routine_id - ルーチン ID モニター・エレメント
STMT_TYPE_ID	VARCHAR(32)	stmt_type_id - ステートメント・タイプ ID
QUERY_COST_ESTIMATE	BIGINT	query_cost_estimate - 照会コストの見積もり
STMT_PKG_CACHE_ID	BIGINT	stmt_pkgcache_id - ステートメント・パッケージ・キャッシュ ID
COORD_STMT_EXEC_TIME	BIGINT	coord_stmt_exec_time - コーディネーター・エージェントによるステートメントの実行時間
STMT_EXEC_TIME	BIGINT	stmt_exec_time - ステートメントの実行時間
TOTAL_SECTION_TIME	BIGINT	total_section_time - セクション時間の合計
TOTAL_SECTION_PROC_TIME	BIGINT	total_section_proc_time - セクション処理時間の合計
TOTAL_ROUTINE_NON_SECT_TIME	BIGINT	total_routine_non_sect_time - 非セクション・ルーチン実行時間
TOTAL_ROUTINE_NON_SECT_PROC_TIME	BIGINT	total_routine_non_sect_proc_time - 非セクション処理時間
LOCK_WAITS_GLOBAL	BIGINT	lock_waits_global - グローバル・ロック待機
LOCK_WAIT_TIME_GLOBAL	BIGINT	lock_wait_time_global - グローバル・ロック待機時間
LOCK_TIMEOUTS_GLOBAL	BIGINT	lock_timeouts_global - グローバル・ロック・タイムアウト
LOCK_ESCALS_MAXLOCKS	BIGINT	lock_escals_maxlocks - maxlocks ロック・エスカレーション数
LOCK_ESCALS_LOCKLIST	BIGINT	lock_escals_locklist - locklist ロック・エスカレーション数
LOCK_ESCALS_GLOBAL	BIGINT	lock_escals_global - グローバル・ロック・エスカレーション数
RECLAIM_WAIT_TIME	BIGINT	reclaim_wait_time - 再利用の待機時間
SPACEMAPPAGE_RECLAIM_WAIT_TIME	BIGINT	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間
CF_WAITS	BIGINT	cf_waits - クラスタ・キャッシング・ファシリティ待機回数
CF_WAIT_TIME	BIGINT	cf_wait_time - クラスタ・キャッシング・ファシリティの待機時間
POOL_DATA_GBP_L_READS	BIGINT	pool_data_gbp_l_reads - グループ・バッファ・プールのデータの論理読み取り
POOL_DATA_GBP_P_READS	BIGINT	pool_data_gbp_p_reads - グループ・バッファ・プールのデータの物理読み取り
POOL_DATA_LBP_PAGES_FOUND	BIGINT	pool_data_lbp_pages_found - ローカル・バッファ・プールで検出されたデータ・ページ



表 151. MON\_GET\_PKG\_CACHE\_STMT について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_DATA_GBP_INVALID_PAGES	BIGINT	pool_data_gbp_invalid_pages - グループ・バッファ ー・プールの無効なデータ・ページ
POOL_INDEX_GBP_L_READS	BIGINT	pool_index_gbp_l_reads - グループ・バッファ ー・プールの索引の論理読み取り
POOL_INDEX_GBP_P_READS	BIGINT	pool_index_gbp_p_reads - グループ・バッファ ー・プールの索引の物理読み取り
POOL_INDEX_LBP_PAGES_FOUND	BIGINT	pool_index_lbp_pages_found - 検出されたローカル・ バッファ ー・プールの索引ページ
POOL_INDEX_GBP_INVALID_PAGES	BIGINT	pool_index_gbp_invalid_pages - グループ・バッファ ー・プールの無効な索引ページ
POOL_XDA_GBP_L_READS	BIGINT	pool_xda_gbp_l_reads - グループ・バッファ ー・プールの XDA データの論理読み取りの要求
POOL_XDA_GBP_P_READS	BIGINT	pool_xda_gbp_p_reads - グループ・バッファ ー・プールの XDA データの物理読み取りの要求
POOL_XDA_LBP_PAGES_FOUND	BIGINT	pool_xda_lbp_pages_found - 検出されたローカル・バ ッファ ー・プールの XDA データ・ページ
POOL_XDA_GBP_INVALID_PAGES	BIGINT	pool_xda_gbp_invalid_pages - グループ・バッファ ー・プールの無効な XDA データ・ページ
AUDIT_EVENTS_TOTAL	BIGINT	audit_events_total - 監査イベント合計数
AUDIT_FILE_WRITES_TOTAL	BIGINT	audit_file_writes_total - 書き込まれた監査ファイルの 合計数
AUDIT_FILE_WRITE_WAIT_TIME	BIGINT	audit_file_write_wait_time - 監査ファイル書き込み待 機時間
AUDIT_SUBSYSTEM_WAITS_TOTAL	BIGINT	audit_subsystem_waits_total - 監査サブシステム待機 の合計
AUDIT_SUBSYSTEM_WAIT_TIME	BIGINT	audit_subsystem_wait_time - 監査サブシステム待機時 間
DIAGLOG_WRITES_TOTAL	BIGINT	diaglog_writes_total - 診断ログ・ファイル書き込みの 合計
DIAGLOG_WRITE_WAIT_TIME	BIGINT	diaglog_write_wait_time - 診断ログ・ファイルの書き 込み待機時間
FCM_MESSAGE_RECVS_TOTAL	BIGINT	fcm_message_recvs_total - FCM メッセージ合計受信 数
FCM_MESSAGE_RECV_VOLUME	BIGINT	fcm_message_recv_volume - FCM メッセージ受信ボ リューム
FCM_MESSAGE_RECV_WAIT_TIME	BIGINT	fcm_message_recv_wait_time - FCM メッセージの受 信待機時間
FCM_MESSAGE_SENDS_TOTAL	BIGINT	fcm_message_sends_total - FCM メッセージ合計送信 数
FCM_MESSAGE_SEND_VOLUME	BIGINT	fcm_message_send_volume - FCM メッセージ送信ボ リューム
FCM_MESSAGE_SEND_WAIT_TIME	BIGINT	fcm_message_send_wait_time - FCM メッセージ送信 待ち時間
FCM_TQ_RECVS_TOTAL	BIGINT	fcm_tq_recvs_total - FCM 表キューの合計受信数

表 151. MON\_GET\_PKG\_CACHE\_STMT について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
FCM_TQ_RECV_VOLUME	BIGINT	fcm_tq_recv_volume - FCM 表キューの受信ボリューム
FCM_TQ_RECV_WAIT_TIME	BIGINT	fcm_tq_recv_wait_time - FCM 表キューの受信待ち時間
FCM_TQ_SENDS_TOTAL	BIGINT	fcm_tq_sends_total - FCM 表キューの合計送信数
FCM_TQ_SEND_VOLUME	BIGINT	fcm_tq_send_volume - FCM 表キューの送信ボリューム
FCM_TQ_SEND_WAIT_TIME	BIGINT	fcm_tq_send_wait_time - FCM 表キューの送信待ち時間
NUM_LW_THRESH_EXCEEDED	BIGINT	num_lw_thresh_exceeded - しきい値を超えた回数
THRESH_VIOLATIONS	BIGINT	thresh_violations - しきい値違反の回数
TOTAL_APP_SECTION_EXECUTIONS	BIGINT	total_app_section_executions - セクション実行の合計回数
TOTAL_ROUTINE_USER_CODE_PROC_TIME	BIGINT	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
TOTAL_ROUTINE_USER_CODE_TIME	BIGINT	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
TQ_TOT_SEND_SPILLS	BIGINT	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
EVMON_WAIT_TIME	BIGINT	evmon_wait_time - イベント・モニターの待機時間
EVMON_WAITS_TOTAL	BIGINT	evmon_waits_total - イベント・モニターの待機の合計回数
TOTAL_EXTENDED_LATCH_WAIT_TIME	BIGINT	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
TOTAL_EXTENDED_LATCH_WAITS	BIGINT	total_extended_latch_waits - 拡張ラッチの合計待機回数
MAX_COORD_STMT_EXEC_TIME	BIGINT	max_coord_stmt_exec_time - コーディネーターの最大ステートメント実行時間
MAX_COORD_STMT_EXEC_TIMESTAMP	TIMESTAMP	max_coord_stmt_exec_timestamp - コーディネーターの最大ステートメント実行のタイム・スタンプ・モニター・エレメント
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - ディスパッチャー実行キュー時間の合計
QUERY_DATA_TAG_LIST	VARCHAR(32)	query_data_tag_list - 照会データ・タグ・リスト
TOTAL_STATS_FABRICATION_TIME	BIGINT	total_stats_fabrication_time - 統計作成の合計時間
TOTAL_STATS_FABRICATIONS	BIGINT	total_stats_fabrications - 統計作成の合計回数
TOTAL_SYNC_RUNSTATS_TIME	BIGINT	total_sync_runstats_time - 同期 RUNSTATS 時間の合計
TOTAL_SYNC_RUNSTATS	BIGINT	total_sync_runstats - 同期 RUNSTATS アクティビティの合計
TOTAL_PEDS	BIGINT	total_peds - partial early distinct の合計回数
DISABLED_PEDS	BIGINT	disabled_peds - 無効化された partial early distinct
POST_THRESHOLD_PEDS	BIGINT	post_threshold_peds - partial early distinct しきい値

表 151. MON\_GET\_PKG\_CACHE\_STMT について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_PEAS	BIGINT	total_peas - partial early aggregation の合計回数
POST_THRESHOLD_PEAS	BIGINT	post_threshold_peas - partial early aggregation しきい値
TQ_SORT_HEAP_REQUESTS	BIGINT	tq_sort_heap_requests - 表キューのソート・ヒープの要求
TQ_SORT_HEAP_REJECTIONS	BIGINT	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否
POOL_QUEUED_ASYNC_DATA_REQS	BIGINT	pool_queued_async_data_reqs - データ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_INDEX_REQS	BIGINT	pool_queued_async_index_reqs - 索引プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_XDA_REQS	BIGINT	pool_queued_async_xda_reqs - XDA プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_OTHER_REQS	BIGINT	pool_queued_async_other_reqs - 非プリフェッチの要求モニター・エレメント
POOL_QUEUED_ASYNC_DATA_PAGES	BIGINT	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_INDEX_PAGES	BIGINT	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_XDA_PAGES	BIGINT	pool_queued_async_xda_pages - XDA ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_DATA_PAGES	BIGINT	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_INDEX_PAGES	BIGINT	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_XDA_PAGES	BIGINT	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_DATA_REQS	BIGINT	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_INDEX_REQS	BIGINT	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求モニター・エレメント

表 151. MON\_GET\_PKG\_CACHE\_STMT について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_FAILED_ASYNC_XDA_REQS	BIGINT	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_OTHER_REQS	BIGINT	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求モニター・エレメント
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間
PREFETCH_WAITS	BIGINT	prefetch_waits - プリフェッチャーの待機カウント・モニター・エレメント
POOL_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ・モニター・エレメント
POOL_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ・モニター・エレメント
POOL_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ・モニター・エレメント
NUM_WORKING_COPIES	BIGINT	作業用コピーの数。
IDA_SEND_WAIT_TIME	BIGINT	インデータベース分析処理にデータを送信するために待機した合計時間。
IDA_SENDS_TOTAL	BIGINT	インデータベース分析処理にデータが送信された合計回数。
IDA_SEND_VOLUME	BIGINT	インデータベース分析処理にデータベース・サーバーから送信されたデータの合計量。値はバイト数で報告されます。
IDA_RECV_WAIT_TIME	BIGINT	インデータベース分析処理からデータを受信するために待機した合計時間。
IDA_RECVS_TOTAL	BIGINT	インデータベース分析処理からデータを受け取った合計回数。
IDA_RECV_VOLUME	BIGINT	インデータベース分析処理からデータベース・サーバーが受け取ったデータの合計量。値はバイト数で報告されます。
STMTNO	INTEGER	stmtno - ステートメント番号モニター・エレメント
NUM_ROUTINES	INTEGER	num_routines - ルーチンの数

表 151. MON\_GET\_PKG\_CACHE\_STMT について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
STMT_TEXT	CLOB(2MB)	stmt_text - SQL 動的ステートメント・テキスト
COMP_ENV_DESC	BLOB(10K)	comp_env_desc - コンパイル環境ハンドル。既存の COMPILATION_ENV 表関数を使用して、必要に応じて特定のステートメントの詳細なコンパイル環境を取得できます。
MAX_COORD_STMT_EXEC_TIME_ARGS	BLOB(10M)	max_coord_stmt_exec_time_args - コーディネーターの最大ステートメント実行時間の引数

## MON\_GET\_PKG\_CACHE\_STMT\_DETAILS - パッケージ・キャッシュ項目に関する詳細メトリックの取得

MON\_GET\_PKG\_CACHE\_STMT\_DETAILS 表関数は、1 つ以上のパッケージ・キャッシュ項目に関する詳細メトリックを戻します。

MON\_GET\_PKG\_CACHE\_STMT\_DETAILS 表関数で戻されるメトリックは、パッケージ・キャッシュ内のステートメントに関するすべてのメトリックの累計を表します。ステートメント・メトリックは、アクティビティ完了時にパッケージ・キャッシュにロールアップされます。

### 構文

```

▶▶—MON_GET_PKG_CACHE_STMT_DETAILS—(—section_type—, —————▶
▶—executable_id—, —search_args—, —member—)—————▶▶

```

スキーマは SYSPROC です。

### 表関数パラメーター

#### section\_type

オプションの入力引数 (D または S)、タイプは CHAR(1)。戻されるステートメントの情報の種類を指定します。引数が NULL または空ストリングである場合、すべての SQL ステートメントについての情報が戻されます。大/小文字は区別されません。D は動的を表し、S は静的を表します。

#### executable\_id

オプションの入力引数、タイプは VARCHAR(32)。データベース・パッケージ・キャッシュの固有のセクションを指定するビット・データを表します。NULL 値を指定すると、すべての SQL ステートメントについての情報が戻されます。executable\_id が指定されている場合、section\_type 引数は無視されます。例えば、動的ステートメントで executable\_id が指定されている場合、section\_type が静的 ("S") として指定されていても、この表関数によって動的ステートメントの詳細が戻されます。

#### search\_args

オプションのタイプ CLOB(1K) の入力パラメーター。これを使用して 1 つ以上の検索引数ストリングをオプションで指定できます。例:

```
'<modified_within>5</modified_within><update_boundary_time>myPkgEvmon
</update_boundary_time>'
```

使用可能な検索引数タグは次のとおりです。

- '<modified\_within>X</modified\_within>'

現在までの *X* 分間にキャッシュに挿入された、または実行されたステートメント項目だけを戻します (*X* は正の整数値)。引数が指定されない場合、キャッシュ内のすべての項目が戻されます。

- '<update\_boundary\_time>evmon\_name</update\_boundary\_time>'

*evmon\_name* で指定されたパッケージ・キャッシュ・イベント・モニターに関して、イベント・モニター境界タイム・スタンプを現在時刻に更新します。このイベント・モニターの WHERE 節で出力基準として *where updated\_since\_boundary\_time* が指定される場合、今後メトリックが更新されるパッケージ・キャッシュ項目だけが、パッケージ・キャッシュからの退去時にキャプチャーされます。指定されたパッケージ・キャッシュ・イベント・モニターがコマンド発行時にアクティブである場合にのみ、この操作が効果を及ぼします。

- '<stmt\_details>>true</stmt\_details>' または '<stmt\_details>>false</stmt\_details>'

結果の XML 文書に *stmt\_text* および *comp\_env\_desc* データを含めるか、または除外します。これを使用すると、文書のこのような比較的大きい部分が不要な場合に、それらを除外できます (例えば、フォーマット設定された行ベースの出力を戻す *MON\_FORMAT\_XML\_\** 表関数に入力を提供するために XML 文書を使用する場合)。この引数タグを指定しない場合、デフォルトでは *stmt\_text* および *comp\_env\_desc* データが含まれます。

それぞれの入力引数は一度だけ指定できます。検索引数タグは小文字で指定する必要があります。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER のオプション入力引数。現在のデータベース・メンバーの場合は -1、すべてのデータベース・メンバーの場合は -2 を指定します。NULL 値を指定すると、-1 が設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

最初の例は、パッケージ・キャッシュを調査して、既に読み取った (および戻した) 行数の多い上位 10 個のステートメントを選ぶ方法を示しています。さらに、これらの各ステートメントの実行に費やされた時間の累計が結果に示されます (STMT\_EXEC\_TIME 出力列)。

```
SELECT SUBSTR(DETMETRICS.STMT_TEXT, 1, 40) STMT_TEXT,
       DETMETRICS.ROWS_RETURNED,
       DETMETRICS.STMT_EXEC_TIME
FROM TABLE(MON_GET_PKG_CACHE_STMT_DETAILS(CAST(NULL AS CHAR(1)),
      CAST(NULL AS VARCHAR(32) FOR BIT DATA),
      CAST(NULL AS CLOB(1K)), -1)) AS STMT_METRICS,
XMLTABLE (XMLNAMESPACES( DEFAULT 'http://www.ibm.com/xmlns/prod/db2/mon'),
      '$DETMETRICS/db2_pkg_cache_stmt_details' PASSING
      XMLPARSE(DOCUMENT STMT_METRICS.DETAILS) as "DETMETRICS"
      COLUMNS "STMT_TEXT" CLOB PATH 'stmt_text',
      "ROWS_RETURNED" BIGINT PATH 'activity_metrics/rows_returned',
      "STMT_EXEC_TIME" BIGINT PATH 'activity_metrics/stmt_exec_time'
      ) AS DETMETRICS
ORDER BY rows_returned DESC
FETCH FIRST 10 ROWS ONLY
```

以下はこの照会の出力例です。

STMT_TEXT	ROWS_RETURNED	STMT_EXEC_TIME
-----	-----	-----
SELECT CREATOR, NAME, CTIME FROM SYSIBM.	134	38
SELECT SUBSTR(DETMETRICS.STMT_TEXT, 1, 4	44	336
SELECT SUBSTR(DETMETRICS.STMT_TEXT, 1, 4	10	333
SELECT COLNAME, TYPENAME FROM SYSCAT.CO	10	6
SELECT SUBSTR(DETMETRICS.STMT_TEXT, 1, 4	10	334
SELECT TRIGNAME FROM SYSCAT.TRIGGERS WH	8	1
SELECT COUNT(*) FROM SYSCAT.TABLESPACES	2	0
SELECT POLICY FROM SYSTOOLS.POLICY WHERE	1	0
CALL SYSPROC.POLICY_INSTALL ('I','DB2Tab	1	62
CALL SYSPROC.POLICY_INSTALL ('I','DB2Tab	1	64

10 record(s) selected.

2 番目の例は、実行中にロックを待機した動的 SQL ステートメントに関して、実行の数、ロック待機の数、およびロック待機ごとに費やされた平均時間を示します。出力にはパッケージ・キャッシュ項目の存続期間にわたって累算された値が示されますが、(modified\_within 引数タグを 1 に設定することで) 最近 1 分以内に実行されたステートメントの情報に限定します。ステートメントの詳細情報 (stmt\_text および comp\_env\_desc データ) は不要であり、レポートの生成処理でのコストが高いため、(stmt\_details 引数タグを false に設定することにより) 照会ではこれらが除外されます。

```
SELECT NUM_EXEC_WITH_METRICS, LOCK_WAITS,
       (LOCK_WAIT_TIME / LOCK_WAITS) AVG_LOCK_WAIT_TIME
FROM TABLE(MON_GET_PKG_CACHE_STMT_DETAILS('D', CAST(NULL
      AS VARCHAR(32) FOR BIT DATA),
      CLOB(
      '<modified_within>1</modified_within><stmt_details>>false</stmt_details>'
      , -1))
      AS STMT_METRICS,
XMLTABLE (XMLNAMESPACES( DEFAULT 'http://www.ibm.com/xmlns/prod/db2/mon'),
      '$DETMETRICS/db2_pkg_cache_stmt_details' PASSING
      XMLPARSE(DOCUMENT STMT_METRICS.DETAILS) as "DETMETRICS"
      COLUMNS "NUM_EXEC_WITH_METRICS" BIGINT PATH 'num_exec_with_metrics',
      "LOCK_WAITS" BIGINT PATH 'lock_waits',
```

```

        "LOCK_WAIT_TIME" BIGINT PATH 'activity_metrics/lock_wait_time'
    ) AS DETMETRICS
WHERE LOCK_WAITS <> 0
ORDER BY AVG_LOCK_WAIT_TIME DESC

```

以下はこの照会の出力例です。

NUM_EXEC_WITH_METRICS	LOCK_WAITS	AVG_LOCK_WAIT_TIME
4	2	139
9	3	90

## 使用上の注意

この関数によって戻されるメトリックは、パッケージ・キャッシュ内のステートメントに関するすべてのメトリックの累計を表します。ステートメント・メトリックは、アクティビティ完了時にパッケージ・キャッシュにロールアップされます。

DETAILS 列に戻される XML 文書のスキーマは、ファイル `sqllib/misc/DB2MonRoutines.xsd` で入手できます。詳細は、ファイル `sqllib/misc/DB2MonCommon.xsd` 内にあります。

## 戻される情報

表 152. `MON_GET_PKG_CACHE_STMT_DETAILS` で戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー
SECTION_TYPE	CHAR(1)	section_type - セクション・タイプ標識
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	executable_id - 実行可能 ID
DETAILS	BLOB(20M)	作業単位についての詳細メトリックを含む XML 文書。エレメントの説明については、本書の表 153を参照してください。

表 153. `MON_GET_PKG_CACHE_STMT_DETAILS` で戻される詳細メトリック

エレメント名	データ・タイプ	説明
audit_events_total	xs:long	audit_events_total - 監査イベント合計数
audit_file_write_wait_time	xs:long	audit_file_write_wait_time - 監査ファイル書き込み待機時間
audit_file_writes_total	xs:long	audit_file_writes_total - 書き込まれた監査ファイルの合計数
audit_subsystem_wait_time	xs:long	audit_subsystem_wait_time - 監査サブシステム待機時間
audit_subsystem_waits_total	xs:long	audit_subsystem_waits_total - 監査サブシステム待機の合計
comp_env_desc	xs:hexBinary(10240)	comp_env_desc - コンパイル環境ハンドル。既存の <code>COMPILATION_ENV</code> 表関数を使用して、必要に応じて特定のステートメントの詳細なコンパイル環境を取得できます。
coord_stmt_exec_time	xs:long	coord_stmt_exec_time - コーディネーター・エージェントによるステートメントの実行時間
deadlocks	xs:long	deadlocks - デッドロック検出数



表 153. MON\_GET\_PKG\_CACHE\_STMT\_DETAILS で戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
diaglog_write_wait_time	xs:long	diaglog_write_wait_time - 診断ログ・ファイルの書き込み待機時間
diaglog_writes_total	xs:long	diaglog_writes_total - 診断ログ・ファイル書き込みの合計
direct_read_reqs	xs:long	direct_read_reqs - 直接読み取り要求
direct_read_time	xs:long	direct_read_time - 直接読み取り時間
direct_reads	xs:long	direct_reads - データベースからの直接読み取り
direct_write_reqs	xs:long	direct_write_reqs - 直接書き込み要求
direct_write_time	xs:long	direct_write_time - 直接書き込み時間
direct_writes	xs:long	direct_writes - データベースへの直接書き込み
disabled_peds	xs:long	disabled_peds - 無効化された partial early distinct
effective_isolation	xs:string(2)	effective_isolation - 有効な分離。これは、セクションで有効な分離値です。コンパイル時に最初に要求されたものとは異なる可能性があります。
evmon_wait_time	xs:nonNegativeInteger	evmon_wait_time - イベント・モニターの待機時間
evmon_waits_total	xs:nonNegativeInteger	evmon_waits_total - イベント・モニターの待機の合計回数
executable_id	xs:hexBinary(32)	executable_id - 実行可能 ID
fcm_message_recv_volume	xs:long	fcm_message_recv_volume - FCM メッセージ受信ボリューム
fcm_message_recv_wait_time	xs:long	fcm_message_recv_wait_time - FCM メッセージの受信待機時間
fcm_message_recvs_total	xs:long	fcm_message_recvs_total - FCM メッセージ合計受信数
fcm_message_send_volume	xs:long	fcm_message_send_volume - FCM メッセージ送信ボリューム
fcm_message_send_wait_time	xs:long	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
fcm_message_sends_total	xs:long	fcm_message_sends_total - FCM メッセージ合計送信数
fcm_recv_volume	xs:long	fcm_recv_volume - FCM 受信ボリューム
fcm_recv_wait_time	xs:long	fcm_recv_wait_time - FCM 受信待機時間
fcm_recvs_total	xs:long	fcm_recvs_total - FCM 合計受信数
fcm_send_volume	xs:long	fcm_send_volume - FCM 送信ボリューム
fcm_send_wait_time	xs:long	fcm_send_wait_time - FCM 送信待ち時間
fcm_sends_total	xs:long	fcm_sends_total - FCM 合計送信数
fcm_tq_recv_volume	xs:long	fcm_tq_recv_volume - FCM 表キューの受信ボリューム
fcm_tq_recv_wait_time	xs:long	fcm_tq_recv_wait_time - FCM 表キューの受信待ち時間
fcm_tq_recvs_total	xs:long	fcm_tq_recvs_total - FCM 表キューの合計受信数
fcm_tq_send_volume	xs:long	fcm_tq_send_volume - FCM 表キューの送信ボリューム
fcm_tq_send_wait_time	xs:long	fcm_tq_send_wait_time - FCM 表キューの送信待ち時間
fcm_tq_sends_total	xs:long	fcm_tq_sends_total - FCM 表キューの合計送信数
ida_recv_wait_time	xs:nonNegativeInteger	インデータベース分析処理からデータを受信するために待機した合計時間。

表 153. MON\_GET\_PKG\_CACHE\_STMT\_DETAILS で戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
ida_recv_volume	xs:nonNegativeInteger	インデータベース分析処理からデータベース・サーバーが受け取ったデータの合計量。値はバイト数で報告されます。
ida_recv_total	xs:nonNegativeInteger	インデータベース分析処理からデータを受け取った合計回数。
ida_send_wait_time	xs:nonNegativeInteger	インデータベース分析処理にデータを送信するために待機した合計時間。
ida_send_volume	xs:nonNegativeInteger	インデータベース分析処理にデータベース・サーバーから送信されたデータの合計量。値はバイト数で報告されます。
ida_sends_total	xs:nonNegativeInteger	インデータベース分析処理にデータが送信された合計回数。
insert_timestamp	xs:dateTime	insert_timestamp - ステートメント挿入タイム・スタンプ
last_metrics_update	xs:dateTime	last_metrics_update - メトリックの最終更新タイム・スタンプ
lock_escals	xs:long	lock_escals - ロック・エスカレーション数
lock_timeouts	xs:long	lock_timeouts - ロック・タイムアウト数
lock_wait_time	xs:long	lock_wait_time - ロック待機中の時間
lock_waits	xs:long	lock_waits - ロック待機数
log_buffer_wait_time	xs:long	log_buffer_wait_time - ログ・バッファ待ち時間
log_disk_wait_time	xs:long	log_disk_wait_time - ログ・ディスク待機時間
log_disk_waits_total	xs:long	log_disk_waits_total - ログ・ディスク待機の合計
max_coord_stmt_exec_time	xs:nonNegativeInteger	max_coord_stmt_exec_time - コーディネーターの最大ステートメント実行時間
max_coord_stmt_exec_time_args	logical-grouping	max_coord_stmt_exec_time_args - コーディネーターの最大ステートメント実行時間の引数
max_coord_stmt_exec_timestamp	xs:dateTime	max_coord_stmt_exec_timestamp - コーディネーターの最大ステートメント実行のタイム・スタンプ
member	xs:short	member - データベース・メンバー
num_coord_exec	xs:long	num_coord_exec - コーディネーター・エージェントによる実行数
num_coord_exec_with_metrics	xs:long	num_coord_exec_with_metrics - メトリックを伴うコーディネーター・エージェントによる実行数
num_exec_with_metrics	xs:nonNegativeInteger	num_exec_with_metrics - メトリック収集を伴う実行数
num_executions	xs:nonNegativeInteger	num_executions - ステートメント実行回数
num_log_buffer_full	xs:long	num_log_buffer_full - フル・ログ・バッファの回数
num_lw_thresh_exceeded	xs:long	num_lw_thresh_exceeded - しきい値を超えた回数
num_routines	xs:int	num_routines - ルーチンの数
num_working_copies	xs:long	作業用コピーの数。
package_name	xs:string(128)	package_name - パッケージ名。この出力は、静的 SQL ステートメントでのみ有効です。ステートメントが動的である場合、NULL 値が戻されます。

表 153. MON\_GET\_PKG\_CACHE\_STMT\_DETAILS で戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
package_schema	xs:string(128)	package_schema - パッケージ・スキーマ。この出力は、静的 SQL ステートメントでのみ有効です。ステートメントが動的である場合、NULL 値が戻されます。
package_version_id	xs:string(64)	package_version_id - パッケージ・バージョン。この出力は、静的 SQL ステートメントでのみ有効です。ステートメントが動的であるか、または静的ステートメントのパッケージ・バージョンを指定しなかった場合には、このエレメントは生成されません。パッケージの作成時にパッケージのバージョン ID を指定しなかった場合には、静的ステートメントに対して空ストリングが戻されます。
pool_data_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ・モニター・エレメント
pool_data_l_reads	xs:long	pool_data_l_reads - バッファ・プール・データの論理読み取り
pool_data_p_reads	xs:long	pool_data_p_reads - バッファ・プール・データの物理読み取り
pool_data_writes	xs:long	pool_data_writes - バッファ・プールへのデータの書き込み
pool_failed_async_data_reqs	xs:nonNegativeInteger	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求モニター・エレメント
pool_failed_async_index_reqs	xs:nonNegativeInteger	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求モニター・エレメント
pool_failed_async_other_reqs	xs:nonNegativeInteger	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求モニター・エレメント
pool_failed_async_temp_data_reqs	xs:nonNegativeInteger	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
pool_failed_async_temp_index_reqs	xs:nonNegativeInteger	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
pool_failed_async_temp_xda_reqs	xs:nonNegativeInteger	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求モニター・エレメント
pool_failed_async_xda_reqs	xs:nonNegativeInteger	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求モニター・エレメント
pool_index_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ・モニター・エレメント
pool_index_l_reads	xs:long	pool_index_l_reads - バッファ・プール索引の論理読み取り
pool_index_p_reads	xs:long	pool_index_p_reads - バッファ・プール索引の物理読み取り
pool_index_writes	xs:long	pool_index_writes - バッファ・プール索引の書き込み

表 153. MON\_GET\_PKG\_CACHE\_STMT\_DETAILS で戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
pool_queued_async_data_pages	xs:nonNegativeInteger	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_data_reqs	xs:nonNegativeInteger	pool_queued_async_data_reqs - データ・プリフェッチ要求モニター・エレメント
pool_queued_async_index_pages	xs:nonNegativeInteger	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_index_reqs	xs:nonNegativeInteger	pool_queued_async_index_reqs - 索引プリフェッチ要求モニター・エレメント
pool_queued_async_other_reqs	xs:nonNegativeInteger	pool_queued_async_other_reqs - 非プリフェッチの要求モニター・エレメント
pool_queued_async_temp_data_pages	xs:nonNegativeInteger	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_data_reqs	xs:nonNegativeInteger	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_index_pages	xs:nonNegativeInteger	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_index_reqs	xs:nonNegativeInteger	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
pool_queued_async_temp_xda_pages	xs:nonNegativeInteger	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_xda_reqs	xs:nonNegativeInteger	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求モニター・エレメント
pool_queued_async_xda_reqs	xs:nonNegativeInteger	pool_queued_async_xda_reqs - XDA プリフェッチ要求モニター・エレメント
pool_read_time	xs:long	pool_read_time - バッファ・プール物理読み取り時間の合計
pool_temp_data_l_reads	xs:long	pool_temp_data_l_reads - バッファ・プルー時データの論理読み取り
pool_temp_data_p_reads	xs:long	pool_temp_data_p_reads - バッファ・プルー時データの物理読み取り
pool_temp_index_l_reads	xs:long	pool_temp_index_l_reads - バッファ・プルー時索引の論理読み取り
pool_temp_index_p_reads	xs:long	pool_temp_index_p_reads - バッファ・プルー時索引の物理読み取り
pool_temp_xda_p_reads	xs:long	pool_temp_xda_p_reads - バッファ・プルー時 XDA データの物理読み取り
pool_write_time	xs:long	pool_write_time - バッファ・プール物理書き込み時間の合計

表 153. MON\_GET\_PKG\_CACHE\_STMT\_DETAILS で戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
pool_xda_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ・モニター・エレメント
pool_xda_gbp_invalid_pages	xs:nonNegativeInteger	pool_xda_gbp_invalid_pages - グループ・バッファ・プールの無効な XDA データ・ページ
pool_xda_gbp_l_reads	xs:nonNegativeInteger	pool_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの論理読み取りの要求
pool_xda_gbp_p_reads	xs:nonNegativeInteger	pool_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの物理読み取りの要求
pool_xda_lbp_pages_found	xs:nonNegativeInteger	pool_xda_lbp_pages_found - 検出されたローカル・バッファ・プールの XDA データ・ページ
pool_xda_p_reads	xs:long	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
pool_xda_writes	xs:long	pool_xda_writes - バッファ・プール XDA データの書き込み
post_shrthreshold_sorts	xs:long	post_shrthreshold_sorts - ポスト共有しきい値ソート
post_threshold_peas	xs:long	post_threshold_peas - partial early aggregation しきい値
post_threshold_peds	xs:long	post_threshold_peds - partial early distinct しきい値
post_threshold_sorts	xs:long	post_threshold_sorts - ポストしきい値ソート
prefetch_wait_time	xs:nonNegativeInteger	prefetch_wait_time - プリフェッチ待ち時間
prefetch_waits	xs:nonNegativeInteger	prefetch_waits - プリフェッチャーの待機カウント・モニター・エレメント
prep_time	xs:nonNegativeInteger	prep_time - 準備時間。PREP_TIME は、動的 SQL ステートメントでのみ有効であることに注意してください。静的 SQL ステートメントの場合、PREP_TIME は 0 としてレポートされます。
query_cost_estimate	xs:long	query_cost_estimate - 照会コストの見積もり
query_data_tag_list	xs:string(32)	query_data_tag_list - 照会データ・タグ・リスト
routine_id	xs:long	routine_id - ルーチン ID。CALL ステートメントの場合、ターゲット・プロシージャのルーチン ID を示します。他のすべてのタイプのステートメントの場合、値は 0 です。
rows_modified	xs:long	rows_modified - 変更された行数
rows_read	xs:long	rows_read - 読み取り行数
rows_returned	xs:long	rows_returned - 戻り行数
section_number	xs:short	section_number - セクション番号。ステートメントが動的である場合には、このエレメントは生成されません。
section_type	xs:string(1)	section_type - セクション・タイプ標識
sort_overflows	xs:long	sort_overflows - ソート・オーバーフロー
stmt_exec_time	xs:long	stmt_exec_time - ステートメントの実行時間
stmt_pkgcache_id	xs:long	stmt_pkgcache_id - ステートメント・パッケージ・キャッシュ ID
stmt_text	xs:string(2097152)	stmt_text - SQL 動的ステートメント・テキスト

表 153. MON\_GET\_PKG\_CACHE\_STMT\_DETAILS で戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
stmt_type_id	xs:string	stmt_type_id - ステートメント・タイプ ID
stmt_value_data	xs:string(32768)	stmt_value_data - 値データ
stmt_value_index	xs:nonNegativeInteger	stmt_value_index - 値索引
stmt_value_isnull	xs:string(20)	stmt_value_isnull - NULL 値の値
stmt_value_isreopt	xs:string(20)	stmt_value_isreopt - ステートメント再最適化に使用される変数
stmt_value_type	xs:string(255)	stmt_value_type - 値タイプ
stmtno	xs:int	stmtno - ステートメント番号モニター・エレメント
thresh_violations	xs:long	thresh_violations - しきい値違反の回数
total_act_time	xs:long	total_act_time - 合計アクティビティー時間
total_act_wait_time	xs:long	total_act_wait_time - 合計アクティビティー待機時間
total_app_section_executions	xs:long	total_app_section_executions - セクション実行の合計回数
total_cpu_time	xs:long	total_cpu_time - 合計 CPU 時間
total_disp_run_queue_time	xs:long	total_disp_run_queue_time - ディスパッチャー実行キュー時間の合計
total_extended_latch_wait_time	xs:nonNegativeInteger	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
total_extended_latch_waits	xs:nonNegativeInteger	total_extended_latch_waits - 拡張ラッチの合計待機回数
total_peas	xs:long	total_peas - partial early aggregation の合計回数
total_peds	xs:long	total_peds - partial early distinct の合計回数
total_routine_invocations	xs:long	total_routine_invocations - ルーチン呼び出しの合計回数
total_routine_non_sect_proc_time	xs:long	total_routine_non_sect_proc_time - 非セクション処理時間
total_routine_non_sect_time	xs:long	total_routine_non_sect_time - 非セクション・ルーチン実行時間
total_routine_time	xs:long	total_routine_time - ルーチン時間の合計
total_routine_user_code_proc_time	xs:long	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
total_routine_user_code_time	xs:long	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
total_section_proc_time	xs:long	total_section_proc_time - セクション処理時間の合計
total_section_sort_proc_time	xs:long	total_section_sort_proc_time - セクション・ソート処理時間合計
total_section_sort_time	xs:long	total_section_sort_time - セクション・ソート時間合計
total_section_sorts	xs:long	total_section_sorts - セクション・ソート合計
total_section_time	xs:long	total_section_time - セクション時間の合計
total_sorts	xs:long	total_sorts - ソート合計
total_stats_fabrication_time	xs:nonNegativeInteger	total_stats_fabrication_time - 統計作成の合計時間
total_stats_fabrications	xs:nonNegativeInteger	total_stats_fabrications - 統計作成の合計回数
total_sync_runstats	xs:nonNegativeInteger	total_sync_runstats - 同期 RUNSTATS アクティビティーの合計

表 153. MON\_GET\_PKG\_CACHE\_STMT\_DETAILS で戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明
total_sync_runstats_time	xs:nonNegativeInteger	total_sync_runstats_time - 同期 RUNSTATS 時間の合計
tq_sort_heap_rejections	xs:long	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否
tq_sort_heap_requests	xs:long	tq_sort_heap_requests - 表キューのソート・ヒープの要求
tq_tot_send_spills	xs:long	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
valid	xs:string(1)	valid - セクション妥当性検査標識
wlm_queue_assignments_total	xs:long	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
wlm_queue_time_total	xs:long	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間

## MON\_GET\_REBALANCE\_STATUS 表関数 - 表スペースのリバランス進行状況の取得

MON\_GET\_REBALANCE\_STATUS 表関数は、表スペースに対するリバランス操作の状況に戻します。

### 構文

▶▶—MON\_GET\_REBALANCE\_STATUS—(—*tbsp\_name*—,—*dbpartitionnum*—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### *tbsp\_name*

照会する表スペースの名前を指定する、タイプ VARCHAR(128) の入力引数。引数値が NULL の場合、この関数はすべての表スペースについて情報を戻します。

#### *dbpartitionnum*

この関数が呼び出されるときに現在接続されているデータベースと同じインスタンス内の有効なデータベース・パーティションを指定する、タイプ INTEGER の入力引数。現行のデータベース・パーティションには -1、すべてのデータベース・パーティションには -2 を指定します。NULL 値を指定すると、-1 が自動的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

### 例

データベースに対するアクティブなリバランス操作をすべてリストし、それらの現在の状態と進行状況を表示します。

```
select
  varchar(tbsp_name, 30) as tbsp_name,
  dbpartitionnum,
  member,
  rebalancer_mode,
  rebalancer_status,
  rebalancer_extents_remaining,
  rebalancer_extents_processed,
  rebalancer_start_time
from table(mon_get_rebalance_status(NULL,-2)) as t
```

```
TBSP_NAME                DBPARTITIONNUM MEMBER REBALANCER_MODE
-----
SYSCATSPACE                0      0 REV_REBAL

REBALANCER_STATUS REBALANCER_EXTENTS_REMAINING REBALANCER_EXTENTS_PROCESSED
-----
ACTIVE                6517                        4

REBALANCER_START_TIME
-----
2011-12-01-12.08.16.000000
```

1 record(s) selected.

### 使用上の注意

MON\_GET\_REBALANCE\_STATUS 表関数は、リバランスが進行中の場合のみ、表スペースのデータを戻します。それ以外の場合、データは戻されません。

### 戻される情報

表 154. MON\_GET\_REBALANCE\_STATUS で戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名モニター・エレメント
TBSP_ID	BIGINT	tablespace_id - 表スペース ID : モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号。リバランスされているデータを示します。
MEMBER	SMALLINT	member - データベース・メンバー。リバランスを行っている処理メンバーを示します。 DB2 pureScale ではリバランスはサポートされていません。



表 154. MON\_GET\_REBALANCE\_STATUS で戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
REBALANCER_MODE	VARCHAR(30)	rebalancer_mode - リバランサー・モード  リバランサー・モードは以下のいずれかです。 <ul style="list-style-type: none"> <li>• FWD_REBAL</li> <li>• REV_REBAL</li> <li>• FWD_REBAL_OF_2PASS</li> <li>• REV_REBAL_OF_2PASS</li> </ul>
REBALANCER_STATUS	VARCHAR(10)	rebalancer_status - リバランサー状況 rebalancer_status - リバランサー状況
REBALANCER_EXTENTS_REMAINING	BIGINT	rebalancer_extents_remaining - リバランサーで処理されるエクステントの合計数
REBALANCER_EXTENTS_PROCESSED	BIGINT	rebalancer_extents_processed - リバランサーで処理されたエクステントの数
REBALANCER_PRIORITY	BIGINT	rebalancer_priority - 現行のリバランサー優先順位
REBALANCER_LAST_EXTENT_MOVED	BIGINT	rebalancer_last_extent_moved - リバランサーによって最後に移動されたエクステント
REBALANCER_START_TIME	TIMESTAMP	rebalancer_start_time - リバランサー開始時刻
REBALANCER_RESTART_TIME	TIMESTAMP	rebalancer_restart_time - リバランサー再始動時刻
REBALANCER_SOURCE_STORAGE_GROUP_NAME	VARCHAR(128)	rebalancer_source_storage_group_name - リバランサー・ソース・ストレージ・グループ名
REBALANCER_SOURCE_STORAGE_GROUP_ID	INTEGER	rebalancer_source_storage_group_id - リバランサー・ソース・ストレージ・グループ ID
REBALANCER_TARGET_STORAGE_GROUP_NAME	VARCHAR(128)	rebalancer_target_storage_group_name - リバランサー・ターゲット・ストレージ・グループ名
REBALANCER_TARGET_STORAGE_GROUP_ID	INTEGER	tablespace_rebalancer_target_storage_group_id - リバランサー・ターゲット・ストレージ・グループ ID

## MON\_GET\_ROUTINE 表関数 - ルーチンの集約された実行メトリックの取得

MON\_GET\_ROUTINE 表関数は、データベースの活動化以降に呼び出されたプロシージャ、外部プロシージャ、コンパイル済み関数、外部関数、コンパイル済みトリガー、および無名ブロックの集約された実行メトリックを返します。

### 許可

以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- SQLADM 権限
- DBADM 権限

## デフォルトの PUBLIC 特権

なし

## 構文

```
▶▶—MON_GET_ROUTINE—(—routine_type—, —routines_chema—, —————→  
▶—routine_module—, —routine_name—, —member—)————→▶▶
```

スキーマは SYSPROC です。

## ルーチン・パラメーター

### *routine\_type*

どのタイプのルーチンまたはコンパイル済みトリガーについてデータを返すかを指定する、タイプ CHAR(2) の入力パラメーター。

- "P" - プロシージャ
- "SP" - プロシージャの特定名
- "F" - コンパイル済み関数
- "SF" - コンパイル済み関数の特定名
- "T" - コンパイル済みトリガー
- "C" - 動的に準備されたコンパウンド SQL ステートメント、または PL/SQL 内の無名ブロック。

ルーチンのタイプとして "P"、"F"、"T"、または "C" を指定した場合に、入力名を指定しないと、指定したタイプのすべてのルーチンが返されます。ルーチンのタイプが "SP" または "SF" の場合は、特定のルーチン名を指定する必要があります。

ルーチンのタイプが、空ストリング、NULL、またはブランクの場合は、すべてのタイプのすべてのルーチンが返されます。

### *routine\_schema*

ルーチンまたはトリガーのスキーマを指定する、タイプ VARCHAR(128) の入力パラメーター。動的に準備されたコンパウンド SQL ステートメントまたは無名ブロックについては、MON\_GET\_SECTION\_ROUTINES 表関数を使用してスキーマを判別できます。すべてのスキーマのルーチンおよびトリガーを返すには、NULL または空ストリングを使用します。このパラメーターには、大文字と小文字の区別があります。

### *routine\_module\_name*

入力ルーチンのモジュール名を指定する (該当する場合)、タイプ VARCHAR(128) の入力パラメーター。すべてのモジュールのルーチンを返すには、NULL または空ストリングを使用します。このパラメーターには、大文字と小文字の区別があります。

### *routine\_name*

ルーチンの名前を指定する、タイプ VARCHAR (128) の入力パラメーター。入力パラメーターが "SP" または "SF" の場合は、特定のルーチン名を指定する必要があります。動的に準備されたコンパウンド SQL ステートメントまたは無名ブロックについては、MON\_GET\_SECTION\_ROUTINES 表関数を使用して名前

を判別できます。その他の入力パラメーターに一致するすべてのルーチンを返すには、NULL または空ストリングを使用します。このパラメーターには、大文字と小文字の区別があります。

*member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力パラメーター。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 戻される情報

表 155. MON\_GET\_ROUTINE について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
ROUTINE_TYPE	CHAR(1)	routine_type - ルーチン・タイプ・モニター・エレメント
ROUTINE_SCHEMA	VARCHAR(128)	routine_schema - ルーチン・スキーマ・モニター・エレメント
ROUTINE_MODULE_NAME	VARCHAR(128)	routine_module_name - ルーチン・モジュール名モニター・エレメント
ROUTINE_NAME	VARCHAR(128)	routine_name - ルーチン名モニター・エレメント
SPECIFIC_NAME	VARCHAR(128)	specific_name - 特定名モニター・エレメント
DYN_COMPOUND_EXEC_ID	VARCHAR(32) FOR BIT DATA	dyn_compound_exec_id - 動的コンパウンド・ステートメント実行可能 ID モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
ROUTINE_ID	INTEGER	routine_id - ルーチン ID モニター・エレメント
SUBROUTINE_ID	INTEGER	subroutine_id - サブルーチン ID モニター・エレメント
LIB_ID	BIGINT	lib_id - ライブラリー ID モニター・エレメント
EXEC_LIST_MEM_EXCEEDED	CHAR(1)	exec_list_mem_exceeded - 実行リスト・メモリー超過モニター・エレメント
EXEC_LIST_CLEANUP_TIME	TIMESTAMP	exec_list_cleanup_time - 実行リスト・クリーンアップ時間モニター・エレメント
TOTAL_TIMES_ROUTINE_INVOKED	BIGINT	total_times_routine_invoked - ルーチン呼び出しの合計数モニター・エレメント
TOTAL_NESTED_INVOCATIONS	BIGINT	total_nested_invocations - ネストされた呼び出しの合計数モニター・エレメント
TOTAL_ROUTINE_COORD_TIME	BIGINT	total_routine_coord_time - ルーチン・コーディネーターの合計時間モニター・エレメント
ACT_ABORTED_TOTAL	BIGINT	act_aborted_total - 打ち切られたアクティビティの合計数
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 完了したアクティビティの合計数

表 155. MON\_GET\_ROUTINE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
ACT_REJECTED_TOTAL	BIGINT	act_rejected_total - リジェクトされたアクティビティの合計数
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込み
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDA データの書き込み
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
DEADLOCKS	BIGINT	deadlocks - デッドロック検出数
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求

表 155. MON\_GET\_ROUTINE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
FCM_RECV_VOLUME	BIGINT	fcm_recv_volume - FCM 受信ボリューム
FCM_RECVS_TOTAL	BIGINT	fcm_recvs_total - FCM 受信の合計
FCM_SEND_VOLUME	BIGINT	fcm_send_volume - FCM 送信ボリューム
FCM_SENDS_TOTAL	BIGINT	fcm_sends_total - FCM 合計送信数
FCM_RECV_WAIT_TIME	BIGINT	fcm_recv_wait_time - FCM 受信待機時間
FCM_SEND_WAIT_TIME	BIGINT	fcm_send_wait_time - FCM 送信待ち時間
LOCK_ESCALS	BIGINT	lock_escalations - ロック・エスカレーション数
LOCK_TIMEOUTS	BIGINT	lock_timeouts - ロック・タイムアウト数
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
LOG_BUFFER_WAIT_TIME	BIGINT	log_buffer_wait_time - ログ・バッファ待ち時間
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - エージェントの待機が発生したフル・ログ・バッファの回数
LOG_DISK_WAIT_TIME	BIGINT	log_disk_wait_time - ログ・ディスク待機時間
LOG_DISK_WAITS_TOTAL	BIGINT	log_disk_waits_total - ログ・ディスク待機の合計
RQSTS_COMPLETED_TOTAL	BIGINT	rqsts_completed_total - 完了した要求の合計数
ROWS_MODIFIED	BIGINT	rows_modified - 変更された行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_RETURNED	BIGINT	rows_returned - 戻り行数
TOTAL_APP_RQST_TIME	BIGINT	total_app_rqst_time - アプリケーション要求合計時間
TOTAL_RQST_TIME	BIGINT	total_rqst_time - 合計要求時間
WLM_QUEUE_TIME_TOTAL	BIGINT	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間
WLM_QUEUE_ASSIGNMENTS_TOTAL	BIGINT	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間
TOTAL_WAIT_TIME	BIGINT	total_wait_time - 合計待ち時間
APP_RQSTS_COMPLETED_TOTAL	BIGINT	app_rqsts_completed_total - 完了したアプリケーション要求の合計数
TOTAL_SECTION_SORT_TIME	BIGINT	total_section_sort_time - セクション・ソート時間合計
TOTAL_SECTION_SORT_PROC_TIME	BIGINT	total_section_sort_proc_time - セクション・ソート処理時間合計
TOTAL_SECTION_SORTS	BIGINT	total_section_sorts - セクション・ソート合計
TOTAL_SORTS	BIGINT	total_sorts - ソート合計
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - ポストしきい値ソート
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - ポスト共有しきい値ソート

表 155. MON\_GET\_ROUTINE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
TOTAL_COMPILE_TIME	BIGINT	total_compile_time - コンパイル時間の合計
TOTAL_COMPILE_PROC_TIME	BIGINT	total_compile_proc_time - コンパイル処理時間の合計
TOTAL_COMPILATIONS	BIGINT	total_compilations - コンパイルの合計回数
TOTAL_IMPLICIT_COMPILE_TIME	BIGINT	total_implicit_compile_time - 暗黙的コンパイル時間の合計
TOTAL_IMPLICIT_COMPILE_PROC_TIME	BIGINT	total_implicit_compile_proc_time - 暗黙的コンパイルの処理時間の合計
TOTAL_IMPLICIT_COMPILATIONS	BIGINT	total_implicit_compilations - 暗黙的コンパイルの合計回数
TOTAL_SECTION_TIME	BIGINT	total_section_time - セクション時間の合計
TOTAL_SECTION_PROC_TIME	BIGINT	total_section_proc_time - セクション処理時間の合計
TOTAL_APP_SECTION_EXECUTIONS	BIGINT	total_app_section_executions - アプリケーションのセクション実行数の合計
TOTAL_ACT_TIME	BIGINT	total_act_time - 合計アクティビティー時間
TOTAL_ACT_WAIT_TIME	BIGINT	total_act_wait_time - 合計アクティビティー待機時間
ACT_RQSTS_TOTAL	BIGINT	act_rqsts_total - アクティビティー要求の合計
TOTAL_ROUTINE_TIME	BIGINT	total_routine_time - ルーチン時間の合計
TOTAL_ROUTINE_INVOCATIONS	BIGINT	total_routine_invocations - ルーチン呼び出しの合計回数 ルーチンがこのルーチンによって呼び出された回数の合計。
TOTAL_COMMIT_TIME	BIGINT	total_commit_time - コミット時間の合計
TOTAL_COMMIT_PROC_TIME	BIGINT	total_commit_proc_time - コミット処理時間の合計
TOTAL_APP_COMMITS	BIGINT	total_app_commits - アプリケーション・コミットの合計回数
INT_COMMITS	BIGINT	int_commits - 内部コミット数
TOTAL_ROLLBACK_TIME	BIGINT	total_rollback_time - ロールバック時間の合計
TOTAL_ROLLBACK_PROC_TIME	BIGINT	total_rollback_proc_time - ロールバック処理時間の合計
TOTAL_APP_ROLLBACKS	BIGINT	total_app_rollback - アプリケーション・ロールバックの合計回数
INT_ROLLBACKS	BIGINT	int_rollback - 内部ロールバック数
TOTAL_RUNSTATS_TIME	BIGINT	total_runstats_time - ランタイム統計時間の合計
TOTAL_RUNSTATS_PROC_TIME	BIGINT	total_runstats_proc_time - ランタイム統計の処理時間の合計
TOTAL_RUNSTATS	BIGINT	total_runstats - ランタイム統計の合計回数
TOTAL_REORG_TIME	BIGINT	total_reorg_time - 再編成時間の合計

表 155. MON\_GET\_ROUTINE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_REORG_PROC_TIME	BIGINT	total_reorg_proc_time - 再編成の処理時間の合計
TOTAL_REORGS	BIGINT	total_reorgs - 再編成の合計回数
TOTAL_LOAD_TIME	BIGINT	total_load_time - ロード時間の合計
TOTAL_LOAD_PROC_TIME	BIGINT	total_load_proc_time - ロード処理時間の合計
TOTAL_LOADS	BIGINT	total_loads - ロードの合計回数
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - カタログ・キャッシュ挿入数
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - カタログ・キャッシュ参照数
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - パッケージ・キャッシュ挿入
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - パッケージ・キャッシュ参照
THRESH_VIOLATIONS	BIGINT	thresh_violations - しきい値違反の回数
NUM_LW_THRESH_EXCEEDED	BIGINT	num_lw_thresh_exceeded - ロック待機の超過されたしきい値の数
LOCK_WAITS_GLOBAL	BIGINT	lock_waits_global - グローバル・ロック待機
LOCK_WAIT_TIME_GLOBAL	BIGINT	lock_wait_time_global - グローバル・ロック待機時間
LOCK_TIMEOUTS_GLOBAL	BIGINT	lock_timeouts_global - グローバル・ロック・タイムアウト
LOCK_ESCALS_MAXLOCKS	BIGINT	lock_escals_maxlocks - maxlocks ロック・エスカレーション数
LOCK_ESCALS_LOCKLIST	BIGINT	lock_escals_locklist - locklist ロック・エスカレーション数
LOCK_ESCALS_GLOBAL	BIGINT	lock_escals_global - グローバル・ロック・エスカレーション数
RECLAIM_WAIT_TIME	BIGINT	reclaim_wait_time - 再利用の待機時間
SPACEMAPPAGE_RECLAIM_WAIT_TIME	BIGINT	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間
CF_WAITS	BIGINT	cf_waits - クラスタ・キャッシング・ファシリティ待機回数
CF_WAIT_TIME	BIGINT	cf_wait_time - クラスタ・キャッシング・ファシリティ待機時間
POOL_DATA_GBP_L_READS	BIGINT	pool_data_gbp_l_reads - グループ・バッファ・プールのデータの論理読み取り
POOL_DATA_GBP_P_READS	BIGINT	pool_data_gbp_p_reads - グループ・バッファ・プールのデータの物理読み取り
POOL_DATA_LBP_PAGES_FOUND	BIGINT	pool_data_lbp_pages_found - ローカル・バッファ・プールで検出されたデータ・ページ
POOL_DATA_GBP_INVALID_PAGES	BIGINT	pool_data_gbp_invalid_pages - グループ・バッファ・プールの無効なデータ・ページ

表 155. MON\_GET\_ROUTINE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_INDEX_GBP_L_READS	BIGINT	pool_index_gbp_l_reads - グループ・バッファ ー・プールの索引の論理読み取り
POOL_INDEX_GBP_P_READS	BIGINT	pool_index_gbp_p_reads - グループ・バッファ ー・プールの索引の物理読み取り
POOL_INDEX_LBP_PAGES_FOUND	BIGINT	pool_index_lbp_pages_found - 検出されたロー カル・バッファー・プールの索引ページ
POOL_INDEX_GBP_INVALID_PAGES	BIGINT	pool_index_gbp_invalid_pages - グループ・バ ッファー・プールの無効な索引ページ
POOL_XDA_GBP_L_READS	BIGINT	pool_xda_gbp_l_reads - グループ・バッファ ー・プールの XDA データの論理読み取りの 要求
POOL_XDA_GBP_P_READS	BIGINT	pool_xda_gbp_p_reads - グループ・バッファ ー・プールの XDA データの物理読み取りの 要求
POOL_XDA_LBP_PAGES_FOUND	BIGINT	pool_xda_lbp_pages_found - 検出されたローカ ル・バッファー・プールの XDA データ・ペ ージ
POOL_XDA_GBP_INVALID_PAGES	BIGINT	pool_xda_gbp_invalid_pages - グループ・バッ ッファー・プールの無効な XDA データ・ペ ージ
AUDIT_EVENTS_TOTAL	BIGINT	audit_events_total - 監査イベント合計数
AUDIT_FILE_WRITES_TOTAL	BIGINT	audit_file_writes_total - 書き込まれた監査ファ イルの合計数
AUDIT_FILE_WRITE_WAIT_TIME	BIGINT	audit_file_write_wait_time - 監査ファイル書き 込み待機時間
AUDIT_SUBSYSTEM_WAITS_TOTAL	BIGINT	audit_subsystem_waits_total - 監査サブシステ ム待機の合計
AUDIT_SUBSYSTEM_WAIT_TIME	BIGINT	audit_subsystem_wait_time - 監査サブシステム 待機時間
DIAGLOG_WRITES_TOTAL	BIGINT	diaglog_writes_total - 診断ログ・ファイル書き 込みの合計
DIAGLOG_WRITE_WAIT_TIME	BIGINT	diaglog_write_wait_time - 診断ログ・ファイル 書き込み待機時間
FCM_MESSAGE_RECVS_TOTAL	BIGINT	fcm_message_recvs_total - FCM メッセージ受 信の合計
FCM_MESSAGE_RECV_VOLUME	BIGINT	fcm_message_recv_volume - FCM メッセージ 受信ボリューム
FCM_MESSAGE_RECV_WAIT_TIME	BIGINT	fcm_message_recv_wait_time - FCM メッセー ジ受信待機時間
FCM_MESSAGE_SENDS_TOTAL	BIGINT	fcm_message_sends_total - FCM メッセージ送 信の合計
FCM_MESSAGE_SEND_VOLUME	BIGINT	fcm_message_send_volume - FCM メッセー ジ送信ボリューム



表 155. MON\_GET\_ROUTINE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
FCM_MESSAGE_SEND_WAIT_TIME	BIGINT	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
FCM_TQ_RECVS_TOTAL	BIGINT	fcm_tq_recvs_total - FCM 表キュー受信の合計
FCM_TQ_RECV_VOLUME	BIGINT	fcm_tq_recv_volume - FCM 表キューの受信ボリューム
FCM_TQ_RECV_WAIT_TIME	BIGINT	fcm_tq_recv_wait_time - FCM 表キュー受信待機時間
FCM_TQ_SENDS_TOTAL	BIGINT	fcm_tq_sends_total - FCM 表キュー送信の合計
FCM_TQ_SEND_VOLUME	BIGINT	fcm_tq_send_volume - FCM 表キューの送信ボリューム
FCM_TQ_SEND_WAIT_TIME	BIGINT	fcm_tq_send_wait_time - FCM 表キュー送信待機時間
TOTAL_ROUTINE_USER_CODE_PROC_TIME	BIGINT	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
TOTAL_ROUTINE_USER_CODE_TIME	BIGINT	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
TQ_TOT_SEND_SPILLS	BIGINT	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
EVMON_WAIT_TIME	BIGINT	evmon_wait_time - イベント・モニターの待機時間
EVMON_WAITS_TOTAL	BIGINT	evmon_waits_total - イベント・モニターの待機の合計回数
TOTAL_EXTENDED_LATCH_WAIT_TIME	BIGINT	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
TOTAL_EXTENDED_LATCH_WAITS	BIGINT	total_extended_latch_waits - 拡張ラッチの合計待機回数
TOTAL_STATS_FABRICATION_TIME	BIGINT	total_stats_fabrication_time - 統計作成の合計時間
TOTAL_STATS_FABRICATION_PROC_TIME	BIGINT	total_stats_fabrication_proc_time - 統計作成の合計処理時間
TOTAL_STATS_FABRICATIONS	BIGINT	total_stats_fabrications - 統計作成の合計回数
TOTAL_SYNC_RUNSTATS_TIME	BIGINT	total_sync_runstats_time - 同期 RUNSTATS 時間の合計
TOTAL_SYNC_RUNSTATS_PROC_TIME	BIGINT	total_sync_runstats_proc_time - 同期 RUNSTATS 処理時間の合計
TOTAL_SYNC_RUNSTATS	BIGINT	total_sync_runstats - 同期 RUNSTATS アクティビティの合計
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - ディスパッチャー実行キュー時間の合計
TOTAL_PEDS	BIGINT	total_peds - partial early distinct の合計回数
DISABLED_PEDS	BIGINT	disabled_peds - 無効化された partial early distinct

表 155. MON\_GET\_ROUTINE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POST_THRESHOLD_PEDS	BIGINT	post_threshold_peds - partial early distinct しきい値
TOTAL_PEAS	BIGINT	total_peas - partial early aggregation の合計回数
POST_THRESHOLD_PEAS	BIGINT	post_threshold_peas - partial early aggregation しきい値
TQ_SORT_HEAP_REQUESTS	BIGINT	tq_sort_heap_requests - 表キューのソート・ヒープの要求
TQ_SORT_HEAP_REJECTIONS	BIGINT	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否
POOL_QUEUED_ASYNC_DATA_REQS	BIGINT	pool_queued_async_data_reqs - データ・プリフェッチ要求
POOL_QUEUED_ASYNC_INDEX_REQS	BIGINT	pool_queued_async_index_reqs - 索引プリフェッチ要求
POOL_QUEUED_ASYNC_XDA_REQS	BIGINT	pool_queued_async_xda_reqs - XDA プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求
POOL_QUEUED_ASYNC_OTHER_REQS	BIGINT	pool_queued_async_other_reqs - プリフェッチャーによって処理されたその他の要求
POOL_QUEUED_ASYNC_DATA_PAGES	BIGINT	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_INDEX_PAGES	BIGINT	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_XDA_PAGES	BIGINT	pool_queued_async_xda_pages - XDA ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_DATA_PAGES	BIGINT	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_INDEX_PAGES	BIGINT	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_XDA_PAGES	BIGINT	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求
POOL_FAILED_ASYNC_DATA_REQS	BIGINT	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求

表 155. MON\_GET\_ROUTINE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_FAILED_ASYNC_INDEX_REQS	BIGINT	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求
POOL_FAILED_ASYNC_XDA_REQS	BIGINT	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求
POOL_FAILED_ASYNC_OTHER_REQS	BIGINT	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間
PREFETCH_WAITS	BIGINT	prefetch_waits - プリフェッチャーの待機カウント
POOL_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ
POOL_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ
POOL_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ
COMM_EXIT_WAIT_TIME	BIGINT	comm_exit_wait_time - 通信バッファ出口待機時間
COMM_EXIT_WAITS	BIGINT	comm_exit_waits - 通信バッファ出口待機回数
IDA_SEND_WAIT_TIME	BIGINT	ida_send_wait_time - データ送信の待機に費やされた時間
IDA_SENDS_TOTAL	BIGINT	ida_sends_total - データの送信回数
IDA_SEND_VOLUME	BIGINT	ida_send_volume - 送信したデータの合計量
IDA_RECV_WAIT_TIME	BIGINT	ida_recv_wait_time - データ受信の待機に費やされた時間
IDA_RECVS_TOTAL	BIGINT	ida_recv_total - データの受信回数
IDA_RECV_VOLUME	BIGINT	ida_recv_volume - 受信したデータの合計量

## 使用上の注意

MON\_GET\_ROUTINE 表関数は、データベース・サーバー上で最もコストの高いルーチンを識別するために使用します。

この表関数は、入力引数に一致する各ルーチンまたはトリガー、および各メンバーについてメトリックを 1 行ずつ返します。入力引数の値は補完的です。返されるメトリックは、各メンバーでのルーチンのすべての実行を集約したものです。メンバー全体からの集約は実行されません。しかし、メンバー全体の集約は、SQL 照会を使用して実行できます（「例」セクションを参照）。ルーチンがサブルーチンを実行する場合、サブルーチンによる処理は親ルーチンのメトリックに組み込まれます。

この表関数が返すカウンターおよび消費時間モニター・エレメントは、サービス・クラスの COLLECT REQUEST METRICS 節によって制御され、データベース・レベルでは mon\_req\_metrics データベース構成パラメーターによって制御されます。どちらの制御も無効になっている場合、報告されるカウンターおよび消費時間モニター・エレメントは 0 です。

動的に準備されたコンパウンド SQL ステートメントのパッケージがパッケージ・キャッシュから除去されると、このルーチンに関する情報は MON\_GET\_ROUTINE 関数によって報告されなくなります。同様に、ルーチンまたはトリガーがドロップされると、そのルーチンまたはトリガーに関する情報は報告されなくなります。

直前の 24 時間の期間に実行されなかったすべてのルーチンはメモリーから除去され、情報が返されなくなります。

## 例

1. モジュール MOD1 内の任意のタイプのルーチンで、CPU 消費量が最も多いルーチンをリストします。

```
SELECT ROUTINE_TYPE, ROUTINE_SCHEMA, ROUTINE_NAME, SPECIFIC_NAME, TOTAL_CPU_TIME
FROM TABLE(MON_GET_ROUTINE(NULL, NULL, 'MOD1', NULL, NULL))
AS T ORDER BY TOTAL_CPU_TIME DESC
```

戻り値

ROUTINE_TYPE	ROUTINE_SCHEMA	ROUTINE_NAME	SPECIFIC_NAME	TOTAL_CPU_TIME
F	DRICARD	FUNCTION3	FUNCTION3	19425
F	DRICARD	FUNCTION4	FUNCTION4	5780
P	DRICARD	P1	SQL120801142627900	4685
C	SYSIBMINTERNAL	COMPILED_ANON_BLOCK_INVOKE	SQL120801153841490	3471
P	SYSPROC	SYSINSTALLOBJECTS	SYSINSTALLOBJECTS	1158
F	DRICARD	FUNCTION1	FUNCTION1	2632
F	DRICARD	FUNCTION2	FUNCTION2	2029

7 record(s) selected.

2. すべてのストアード・プロシージャーについて集約したモニター・データをリストします。

```
SELECT * FROM TABLE(MON_GET_ROUTINE('P', NULL, NULL, NULL, -2)) AS T
```

3. すべてのタイプのルーチンについて集約したモニター・データをリストします。

```
SELECT * FROM TABLE(MON_GET_ROUTINE(NULL, NULL, NULL, NULL, -2)) AS T
```

4. スキーマ TEST 内の PROC1 という名前のすべてのプロシージャーについて集約したモニター・データをリストします。

```
SELECT * FROM TABLE(MON_GET_ROUTINE('P', 'TEST', NULL, 'PROC1', -2)) AS T
```

5. スキーマ TEST 内の多重定義プロシージャ PROC1 のうち、特定名 PROC1\_OVERLOAD のプロシージャについて集約したモニター・データをリストします。

```
SELECT * FROM TABLE(MON_GET_ROUTINE('SP', 'TEST', NULL, 'PROC1_OVERLOAD', -2))
AS T
```

6. 実行可能 ID

x'010000000000000052010000000000001000000010020120822205618607103' を持つ無名ブロックについて集約したモニター・データをリストします。

```
SELECT * FROM TABLE(MON_GET_ROUTINE
('A', 'MYSCHEMA', NULL, 'SQL181500027522310', -1)) AS T
WHERE
DYN_COMPOUND_EXEC_ID = x'010000000000000052010000000000001000000010020120822205618607103'
```

## MON\_GET\_ROUTINE\_DETAILS 表関数 - ルーチンの集約された実行メトリック詳細の取得

MON\_GET\_ROUTINE\_DETAILS 表関数は、データベースの活動化以降に呼び出されたプロシージャ、外部関数、コンパイル済み関数、外部関数、コンパイル済みトリガー、および無名ブロックの集約された実行メトリックを返します。メトリックは XML 文書で返されます。

### 許可

以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- SQLADM 権限
- DBADM 権限

### デフォルトの PUBLIC 特権

なし

### 構文

```
►►MON_GET_ROUTINE_DETAILS(—routine_type—,—routine_schema—,——————►
►—routine_module_name—,—routine_name—,—member—)—————►►
```

スキーマは SYSPROC です。

### ルーチン・パラメーター

#### *routine\_type*

どのタイプのルーチンまたはコンパイル済みトリガーについてデータを返すかを指定する、タイプ CHAR(2) の入力パラメーター。

- "P" - プロシージャ
- "SP" - プロシージャの特定名
- "F" - コンパイル済み関数
- "SF" - コンパイル済み関数の特定名
- "T" - コンパイル済みトリガー

- "C" - 無名ブロック

ルーチンのタイプとして "P"、"F"、"T"、または "C" を指定した場合に、入力名を指定しないと、指定したタイプのすべてのルーチンが返されます。ルーチンのタイプが "SP" または "SF" の場合は、特定のルーチン名を指定する必要があります。

ルーチンのタイプが NULL、空ストリング、またはブランクの場合は、すべてのタイプのすべてのルーチンが返されます。

#### *routine\_schema*

ルーチンまたはトリガーのスキーマを指定する、タイプ VARCHAR(128) の入力パラメーター。動的に準備されたコンパウンド SQL ステートメントおよび PL/SQL 無名ブロックについては、MON\_GET\_SECTION\_ROUTINES 表関数を使用してスキーマを判別できます。すべてのスキーマのルーチンおよびトリガーを返すには、NULL または空ストリングを使用します。このパラメーターには、大文字と小文字の区別があります。

#### *routine\_module\_name*

入力ルーチンのモジュール名を指定する (該当する場合)、タイプ VARCHAR(128) の入力パラメーター。すべてのモジュールのルーチンを返すには、NULL または空ストリングを使用します。このパラメーターには、大文字と小文字の区別があります。

#### *routine\_name*

ルーチンの名前を指定する、タイプ VARCHAR (128) の入力パラメーター。入力パラメーターが "SP" または "SF" の場合は、特定のルーチン名を指定する必要があります。動的に準備されたコンパウンド SQL ステートメントおよび PL/SQL 無名ブロックについては、MON\_GET\_SECTION\_ROUTINES 表関数を使用して名前を判別できます。その他の入力パラメーターに一致するすべてのルーチンを返すには、NULL または空ストリングを使用します。このパラメーターには、大文字と小文字の区別があります。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力パラメーター。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 戻される情報

表 156. MON\_GET\_ROUTINE\_DETAILS について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
ROUTINE_TYPE	CHAR(1)	routine_type - ルーチン・タイプ・モニター・エレメント
ROUTINE_SCHEMA	VARCHAR(128)	routine_schema - ルーチン・スキーマ・モニター・エレメント
ROUTINE_MODULE_NAME	VARCHAR(128)	routine_module_name - ルーチン・モジュール名モニター・エレメント
ROUTINE_NAME	VARCHAR(128)	routine_name - ルーチン名モニター・エレメント

表 156. MON\_GET\_ROUTINE\_DETAILS について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SPECIFIC_NAME	VARCHAR(128)	specific_name - 特定名モニター・エレメント
DYN_COMPOUND_EXEC_ID	VARCHAR(32) FOR BIT DATA	dyn_compound_exec_id - 動的コンパウンド・ステートメント実行可能 ID モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
ROUTINE_ID	INTEGER	routine_id - ルーチン ID モニター・エレメント
SUBROUTINE_ID	INTEGER	subroutine_id - サブルーチン ID モニター・エレメント
LIB_ID	INTEGER	lib_id - ライブラリー ID モニター・エレメント
DETAILS	BLOB(1M)	ルーチンのモニター情報が入った XML 文書。この文書内のエレメントの説明は、表 2 を参照してください。

以下の例は、DETAILS 列で戻される XML 文書の構造を示しています。

```
<db2_routine xmlns="http://www.ibm.com/xmlns/prod/db2/mon" release="100102000">
  <routine_type>P</routine_type>
  <routine_schema>TEST</routine_schema>
  <routine_module></routine_module>
  <routine_name>PROC1</routine_name>
  ...
  <system_metrics release="100102000">
    <act_aborted_total>5</act_aborted_total>
  </system_metrics>
</db2_routine>
```

スキーマの全体については、sql1lib/misc/DB2MonRoutines.xsd を参照してください。

表 157. MON\_GET\_ROUTINE\_DETAILS について戻される詳細メトリック

列名	データ・タイプ	説明または対応するモニター・エレメント
ACT_ABORTED_TOTAL	xs:long	act_aborted_total - 打ち切られたアクティビティーの合計数
ACT_COMPLETED_TOTAL	xs:long	act_completed_total - 完了したアクティビティーの合計数
ACT_REJECTED_TOTAL	xs:long	act_rejected_total - リジェクトされたアクティビティーの合計数
ACT_RQSTS_TOTAL	xs:long	act_rqsts_total - アクティビティー要求の合計
APP_RQSTS_COMPLETED_TOTAL	xs:long	app_rqsts_completed_total - 完了したアプリケーション要求の合計数
AUDIT_EVENTS_TOTAL	xs:long	audit_events_total - 監査イベント合計数

表 157. MON\_GET\_ROUTINE\_DETAILS について戻される詳細メトリック (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
AUDIT_FILE_WRITE_WAIT_TIME	xs:long	audit_file_write_wait_time - 監査ファイル書き込み待機時間
AUDIT_FILE_WRITES_TOTAL	xs:long	audit_file_writes_total - 書き込まれた監査ファイルの合計数
AUDIT_SUBSYSTEM_WAIT_TIME	xs:long	audit_subsystem_wait_time - 監査サブシステム待機時間
AUDIT_SUBSYSTEM_WAITS_TOTAL	xs:long	audit_subsystem_waits_total - 監査サブシステム待機の合計
CAT_CACHE_INSERTS	xs:long	cat_cache_inserts - カタログ・キャッシュ挿入数
CAT_CACHE_LOOKUPS	xs:long	cat_cache_lookups - カタログ・キャッシュ参照数
CF_WAIT_TIME	xs:long	cf_wait_time - クラスタ・キャッシング・ファシリティー待機時間
CF_WAITS	xs:long	cf_waits - クラスタ・キャッシング・ファシリティー待機回数
COMM_EXIT_WAIT_TIME	xs:long	comm_exit_wait_time - 通信バッファ出口待機時間
COMM_EXIT_WAITS	xs:long	comm_exit_waits - 通信バッファ出口待機回数
DEADLOCKS	xs:long	deadlocks - デッドロック検出数
DIAGLOG_WRITE_WAIT_TIME	xs:long	diaglog_write_wait_time - 診断ログ・ファイル書き込み待機時間
DIAGLOG_WRITES_TOTAL	xs:long	diaglog_writes_total - 診断ログ・ファイル書き込みの合計
DIRECT_READ_REQS	xs:long	direct_read_reqs - 直接読み取り要求
DIRECT_READ_TIME	xs:long	direct_read_time - 直接読み取り時間
DIRECT_READS	xs:long	direct_reads - データベースからの直接読み取り
DIRECT_WRITE_REQS	xs:long	direct_write_reqs - 直接書き込み要求
DIRECT_WRITE_TIME	xs:long	direct_write_time - 直接書き込み時間
DIRECT_WRITES	xs:long	direct_writes - データベースへの直接書き込み
DISABLED_PEDS	xs:long	disabled_peds - 無効化された partial early distinct
DYN_COMPOUND_EXEC_ID	xs:hexBinary(32)	dyn_compound_exec_id - 動的コンパウンド・ステートメント実行可能 ID モニター・エレメント
EVMON_WAIT_TIME	xs:long	evmon_wait_time - イベント・モニターの待機時間
EVMON_WAITS_TOTAL	xs:long	evmon_waits_total - イベント・モニターの待機の合計回数
EXEC_LIST_CLEANUP_TIME	xs:dateTime	exec_list_cleanup_time - 実行リスト・クリーンアップ時間モニター・エレメント



表 157. MON\_GET\_ROUTINE\_DETAILS について戻される詳細メトリック (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
EXEC_LIST_MEM_EXCEEDED	xs:string(1)	exec_list_mem_exceeded - 実行リスト・メモリー超過モニター・エレメント
FCM_MESSAGE_RECV_VOLUME	xs:long	fcm_message_recv_volume - FCM メッセージ受信ボリューム
FCM_MESSAGE_RECV_WAIT_TIME	xs:long	fcm_message_recv_wait_time - FCM メッセージ受信待機時間
FCM_MESSAGE_RECVS_TOTAL	xs:long	fcm_message_recvs_total - FCM メッセージ受信の合計
FCM_MESSAGE_SEND_VOLUME	xs:long	fcm_message_send_volume - FCM メッセージ送信ボリューム
FCM_MESSAGE_SEND_WAIT_TIME	xs:long	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
FCM_MESSAGE_SENDS_TOTAL	xs:long	fcm_message_sends_total - FCM メッセージ送信の合計
FCM_RECV_VOLUME	xs:long	fcm_recv_volume - FCM 受信ボリューム
FCM_RECV_WAIT_TIME	xs:long	fcm_recv_wait_time - FCM 受信待機時間
FCM_RECVS_TOTAL	xs:long	fcm_recvs_total - FCM 受信の合計
FCM_SEND_VOLUME	xs:long	fcm_send_volume - FCM 送信ボリューム
FCM_SEND_WAIT_TIME	xs:long	fcm_send_wait_time - FCM 送信待ち時間
FCM_SENDS_TOTAL	xs:long	fcm_sends_total - FCM 合計送信数
FCM_TQ_RECV_VOLUME	xs:long	fcm_tq_recv_volume - FCM 表キューの受信ボリューム
FCM_TQ_RECV_WAIT_TIME	xs:long	fcm_tq_recv_wait_time - FCM 表キュー受信待機時間
FCM_TQ_RECVS_TOTAL	xs:long	fcm_tq_recvs_total - FCM 表キュー受信の合計
FCM_TQ_SEND_VOLUME	xs:long	fcm_tq_send_volume - FCM 表キューの送信ボリューム
FCM_TQ_SEND_WAIT_TIME	xs:long	fcm_tq_send_wait_time - FCM 表キュー送信待機時間
FCM_TQ_SENDS_TOTAL	xs:long	fcm_tq_sends_total - FCM 表キュー送信の合計
IDA_RECV_VOLUME	xs:long	ida_recv_volume - 受信したデータの合計量
IDA_RECV_WAIT_TIME	xs:long	ida_recv_wait_time - データ受信の待機に費やされた時間
IDA_RECVS_TOTAL	xs:long	ida_recvs_total - データの受信回数
IDA_SEND_VOLUME	xs:long	ida_send_volume - 送信したデータの合計量
IDA_SEND_WAIT_TIME	xs:long	ida_send_wait_time - データ送信の待機に費やされた時間
IDA_SENDS_TOTAL	xs:long	ida_sends_total - データの送信回数
INT_COMMITS	xs:long	int_commits - 内部コミット数
INT_ROLLBACKS	xs:long	int_rollbacks - 内部ロールバック数

表 157. MON\_GET\_ROUTINE\_DETAILS について戻される詳細メトリック (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LIB_ID	xs:int	lib_id - ライブラリー ID モニター・エレメント
LOCK_ESCALS_GLOBAL	xs:long	lock_escals_global - グローバル・ロック・エスカレーション数
LOCK_ESCALS_LOCKLIST	xs:long	lock_escals_locklist - locklist ロック・エスカレーション数
LOCK_ESCALS_MAXLOCKS	xs:long	lock_escals_maxlocks - maxlocks ロック・エスカレーション数
LOCK_ESCALS	xs:long	lock_escals - ロック・エスカレーション数
LOCK_TIMEOUTS_GLOBAL	xs:long	lock_timeouts_global - グローバル・ロック・タイムアウト
LOCK_TIMEOUTS	xs:long	lock_timeouts - ロック・タイムアウト数
LOCK_WAIT_TIME_GLOBAL	xs:long	lock_wait_time_global - グローバル・ロック待機時間
LOCK_WAIT_TIME	xs:long	lock_wait_time - ロック待機中の時間
LOCK_WAITS_GLOBAL	xs:long	lock_waits_global - グローバル・ロック待機
LOCK_WAITS	xs:long	lock_waits - ロック待機数
LOG_BUFFER_WAIT_TIME	xs:long	log_buffer_wait_time - ログ・バッファ待ち時間
LOG_DISK_WAIT_TIME	xs:long	log_disk_wait_time - ログ・ディスク待機時間
LOG_DISK_WAITS_TOTAL	xs:long	log_disk_waits_total - ログ・ディスク待機の合計
MEMBER	xs:short	member - データベース・メンバー・モニター・エレメント
NUM_LOG_BUFFER_FULL	xs:long	num_log_buffer_full - エージェントの待機が発生したフル・ログ・バッファの回数
NUM_LW_THRESH_EXCEEDED	xs:long	num_lw_thresh_exceeded - ロック待機の超過されたしきい値の数
PKG_CACHE_INSERTS	xs:long	pkg_cache_inserts - パッケージ・キャッシュ挿入
PKG_CACHE_LOOKUPS	xs:long	pkg_cache_lookups - パッケージ・キャッシュ参照
POOL_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	xs:long	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ
POOL_DATA_GBP_INVALID_PAGES	xs:long	pool_data_gbp_invalid_pages - グループ・バッファ・プールの無効なデータ・ページ
POOL_DATA_GBP_L_READS	xs:long	pool_data_gbp_l_reads - グループ・バッファ・プールのデータの論理読み取り
POOL_DATA_GBP_P_READS	xs:long	pool_data_gbp_p_reads - グループ・バッファ・プールのデータの物理読み取り

表 157. MON\_GET\_ROUTINE\_DETAILS について戻される詳細メトリック (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_DATA_L_READS	xs:long	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_DATA_LBP_PAGES_FOUND	xs:long	pool_data_lbp_pages_found - ローカル・バッファ・プールで検出されたデータ・ページ
POOL_DATA_P_READS	xs:long	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_DATA_WRITES	xs:long	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_FAILED_ASYNC_DATA_REQS	xs:long	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求
POOL_FAILED_ASYNC_INDEX_REQS	xs:long	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求
POOL_FAILED_ASYNC_OTHER_REQS	xs:long	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求
POOL_FAILED_ASYNC_TEMP_DATA_REQS	xs:long	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_INDEX_REQS	xs:long	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求
POOL_FAILED_ASYNC_TEMP_XDA_REQS	xs:long	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求
POOL_FAILED_ASYNC_XDA_REQS	xs:long	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求
POOL_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	xs:long	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ
POOL_INDEX_GBP_INVALID_PAGES	xs:long	pool_index_gbp_invalid_pages - グループ・バッファ・プールの無効な索引ページ
POOL_INDEX_GBP_L_READS	xs:long	pool_index_gbp_l_reads - グループ・バッファ・プールの索引の論理読み取り
POOL_INDEX_GBP_P_READS	xs:long	pool_index_gbp_p_reads - グループ・バッファ・プールの索引の物理読み取り
POOL_INDEX_L_READS	xs:long	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_INDEX_LBP_PAGES_FOUND	xs:long	pool_index_lbp_pages_found - 検出されたローカル・バッファ・プールの索引ページ
POOL_INDEX_P_READS	xs:long	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_INDEX_WRITES	xs:long	pool_index_writes - バッファ・プール索引の書き込み

表 157. MON\_GET\_ROUTINE\_DETAILS について戻される詳細メトリック (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_QUEUED_ASYNC_DATA_PAGES	xs:long	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_DATA_REQS	xs:long	pool_queued_async_data_reqs - データ・プリフェッチ要求
POOL_QUEUED_ASYNC_INDEX_PAGES	xs:long	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_INDEX_REQS	xs:long	pool_queued_async_index_reqs - 索引プリフェッチ要求
POOL_QUEUED_ASYNC_OTHER_REQS	xs:long	pool_queued_async_other_reqs - プリフェッチャーによって処理されたその他の要求
POOL_QUEUED_ASYNC_TEMP_DATA_PAGES	xs:long	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_DATA_REQS	xs:long	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_INDEX_PAGES	xs:long	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_INDEX_REQS	xs:long	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_XDA_PAGES	xs:long	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_TEMP_XDA_REQS	xs:long	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求
POOL_QUEUED_ASYNC_XDA_PAGES	xs:long	pool_queued_async_xda_pages - XDA ページ・プリフェッチ要求
POOL_QUEUED_ASYNC_XDA_REQS	xs:long	pool_queued_async_xda_reqs - XDA プリフェッチ要求
POOL_READ_TIME	xs:long	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_TEMP_DATA_L_READS	xs:long	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	xs:long	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_TEMP_INDEX_L_READS	xs:long	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	xs:long	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_TEMP_XDA_L_READS	xs:long	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り

表 157. MON\_GET\_ROUTINE\_DETAILS について戻される詳細メトリック (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_TEMP_XDA_P_READS	xs:long	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り
POOL_WRITE_TIME	xs:long	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	xs:long	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ
POOL_XDA_GBP_INVALID_PAGES	xs:long	pool_xda_gbp_invalid_pages - グループ・バッファ・プールの無効な XDA データ・ページ
POOL_XDA_GBP_L_READS	xs:long	pool_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの論理読み取りの要求
POOL_XDA_GBP_P_READS	xs:long	pool_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの物理読み取りの要求
POOL_XDA_L_READS	xs:long	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_XDA_LBP_PAGES_FOUND	xs:long	pool_xda_lbp_pages_found - 検出されたローカル・バッファ・プールの XDA データ・ページ
POOL_XDA_P_READS	xs:long	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_XDA_WRITES	xs:long	pool_xda_writes - バッファ・プール XDA データの書き込み
POST_SHRTHRESHOLD_SORTS	xs:long	post_shrthreshold_sorts - ポスト共有しきい値ソート
POST_THRESHOLD_PEAS	xs:long	post_threshold_peas - partial early aggregation しきい値
POST_THRESHOLD_PEDS	xs:long	post_threshold_peds - partial early distinct しきい値
POST_THRESHOLD_SORTS	xs:long	post_threshold_sorts - ポストしきい値ソート
PREFETCH_WAIT_TIME	xs:long	prefetch_wait_time - プリフェッチ待ち時間
PREFETCH_WAITS	xs:long	prefetch_waits - プリフェッチャーの待機カウント
RECLAIM_WAIT_TIME	xs:long	reclaim_wait_time - 再利用の待機時間
ROUTINE_ID	xs:int	routine_id - ルーチン ID モニター・エレメント
ROUTINE_MODULE_NAME	xs:string(128)	routine_module_name - ルーチン・モジュール名モニター・エレメント
ROUTINE_NAME	xs:string(128)	routine_name - ルーチン名モニター・エレメント

表 157. MON\_GET\_ROUTINE\_DETAILS について戻される詳細メトリック (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
ROUTINE_SCHEMA	xs:string(128)	routine_schema - ルーチン・スキーマ・モニター・エレメント
ROUTINE_TYPE	xs:string(1)	routine_type - ルーチン・タイプ・モニター・エレメント
ROWS_MODIFIED	xs:long	rows_modified - 変更された行数
ROWS_READ	xs:long	rows_read - 読み取り行数
ROWS_RETURNED	xs:long	rows_returned - 戻り行数
RQSTS_COMPLETED_TOTAL	xs:long	rqsts_completed_total - 完了した要求の合計数
SORT_OVERFLOW	xs:long	sort_overflows - ソート・オーバーフロー
SPACEMAPPAGE_RECLAIM_WAIT_TIME	xs:long	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間
SPECIFIC_NAME	xs:string(128)	specific_name - 特定名モニター・エレメント
SUBROUTINE_ID	xs:int	subroutine_id - サブルーチン ID モニター・エレメント
THRESH_VIOLATIONS	xs:long	thresh_violations - しきい値違反の回数
TOTAL_ACT_TIME	xs:long	total_act_time - 合計アクティビティー時間
TOTAL_ACT_WAIT_TIME	xs:long	total_act_wait_time - 合計アクティビティー待機時間
TOTAL_APP_COMMITS	xs:long	total_app_commits - アプリケーション・コミットの合計回数
TOTAL_APP_ROLLBACKS	xs:long	total_app_rollback - アプリケーション・ロールバックの合計回数
TOTAL_APP_RQST_TIME	xs:long	total_app_rqst_time - アプリケーション要求合計時間
TOTAL_APP_SECTION_EXECUTIONS	xs:long	total_app_section_executions - アプリケーションのセクション実行数の合計
TOTAL_COMMIT_PROC_TIME	xs:long	total_commit_proc_time - コミット処理時間の合計
TOTAL_COMMIT_TIME	xs:long	total_commit_time - コミット時間の合計
TOTAL_COMPILE	xs:long	total_compilations - コンパイルの合計回数
TOTAL_COMPILE_PROC_TIME	xs:long	total_compile_proc_time - コンパイル処理時間の合計
TOTAL_COMPILE_TIME	xs:long	total_compile_time - コンパイル時間の合計
TOTAL_CPU_TIME	xs:long	total_cpu_time - 合計 CPU 時間
TOTAL_DISP_RUN_QUEUE_TIME	xs:long	total_disp_run_queue_time - ディスパッチャー実行キュー時間の合計
TOTAL_EXTENDED_LATCH_WAIT_TIME	xs:long	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
TOTAL_EXTENDED_LATCH_WAITS	xs:long	total_extended_latch_waits - 拡張ラッチの合計待機回数
TOTAL_IMPLICIT_COMPILE	xs:long	total_implicit_compilations - 暗黙的コンパイルの合計回数

表 157. MON\_GET\_ROUTINE\_DETAILS について戻される詳細メトリック (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_IMPLICIT_COMPILE_PROC_TIME	xs:long	total_implicit_compile_proc_time - 暗黙的コンパイルの処理時間の合計
TOTAL_IMPLICIT_COMPILE_TIME	xs:long	total_implicit_compile_time - 暗黙的コンパイル時間の合計
TOTAL_LOAD_PROC_TIME	xs:long	total_load_proc_time - ロード処理時間の合計
TOTAL_LOAD_TIME	xs:long	total_load_time - ロード時間の合計
TOTAL_LOADS	xs:long	total_loads - ロードの合計回数
TOTAL_NESTED_INVOCATIONS	xs:long	total_nested_invocations - ネストされた呼び出しの合計数モニター・エレメント
TOTAL_PEAS	xs:long	total_peas - partial early aggregation の合計回数
TOTAL_PEDS	xs:long	total_peds - partial early distinct の合計回数
TOTAL_REORG_PROC_TIME	xs:long	total_reorg_proc_time - 再編成の処理時間の合計
TOTAL_REORG_TIME	xs:long	total_reorg_time - 再編成時間の合計
TOTAL_REORGS	xs:long	total_reorgs - 再編成の合計回数
TOTAL_ROLLBACK_PROC_TIME	xs:long	total_rollback_proc_time - ロールバック処理時間の合計
TOTAL_ROLLBACK_TIME	xs:long	total_rollback_time - ロールバック時間の合計
TOTAL_ROUTINE_COORD_TIME	xs:long	total_routine_coord_time - ルーチン・コーディネーターの合計時間モニター・エレメント
TOTAL_ROUTINE_INVOCATIONS	xs:long	total_routine_invocations - ルーチン呼び出しの合計回数ルーチンがこのルーチンによって呼び出された回数の合計。
TOTAL_ROUTINE_TIME	xs:long	total_routine_time - ルーチン時間の合計
TOTAL_ROUTINE_USER_CODE_PROC_TIME	xs:long	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
TOTAL_ROUTINE_USER_CODE_TIME	xs:long	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
TOTAL_RQST_TIME	xs:long	total_rqst_time - 合計要求時間
TOTAL_RUNSTATS_PROC_TIME	xs:long	total_runstats_proc_time - ランタイム統計の処理時間の合計
TOTAL_RUNSTATS_TIME	xs:long	total_runstats_time - ランタイム統計時間の合計
TOTAL_RUNSTATS	xs:long	total_runstats - ランタイム統計の合計回数
TOTAL_SECTION_PROC_TIME	xs:long	total_section_proc_time - セクション処理時間の合計
TOTAL_SECTION_SORT_PROC_TIME	xs:long	total_section_sort_proc_time - セクション・ソート処理時間合計
TOTAL_SECTION_SORT_TIME	xs:long	total_section_sort_time - セクション・ソート時間合計
TOTAL_SECTION_SORTS	xs:long	total_section_sorts - セクション・ソート合計
TOTAL_SECTION_TIME	xs:long	total_section_time - セクション時間の合計

表 157. MON\_GET\_ROUTINE\_DETAILS について戻される詳細メトリック (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_SORTS	xs:long	total_sorts - ソート合計
TOTAL_STATS_FABRICATION_PROC_TIME	xs:long	total_stats_fabrication_proc_time - 統計作成の合計処理時間
TOTAL_STATS_FABRICATION_TIME	xs:long	total_stats_fabrication_time - 統計作成の合計時間
TOTAL_STATS_FABRICATIONS	xs:long	total_stats_fabrications - 統計作成の合計回数
TOTAL_SYNC_RUNSTATS_PROC_TIME	xs:long	total_sync_runstats_proc_time - 同期 RUNSTATS 処理時間の合計
TOTAL_SYNC_RUNSTATS_TIME	xs:long	total_sync_runstats_time - 同期 RUNSTATS 時間の合計
TOTAL_SYNC_RUNSTATS	xs:long	total_sync_runstats - 同期 RUNSTATS アクティビティの合計
TOTAL_TIMES_ROUTINE_INVOKED	xs:long	total_times_routine_invoked - ルーチン呼び出しの合計数モニター・エレメント
TOTAL_WAIT_TIME	xs:long	total_wait_time - 合計待ち時間
TQ_SORT_HEAP_REJECTIONS	xs:long	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否
TQ_SORT_HEAP_REQUESTS	xs:long	tq_sort_heap_requests - 表キューのソート・ヒープの要求
TQ_TOT_SEND_SPILLS	xs:long	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
WLM_QUEUE_ASSIGNMENTS_TOTAL	xs:long	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
WLM_QUEUE_TIME_TOTAL	xs:long	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間

### 使用上の注意

MON\_GET\_ROUTINE\_DETAILS 表関数は、データベース・サーバー上で最もコストの高いルーチンを識別するために使用します。

この表関数は、入力引数に一致する各ルーチンまたはトリガー、および各メンバーについてメトリックを 1 行ずつ返します。入力引数の値は補完的です。返されるメトリックは、各メンバーでのルーチンのすべての実行を集約したものです。メンバー全体からの集約は実行されません。しかし、メンバー全体の集約は、SQL 照会を使用して実行できます (「例」セクションを参照)。ルーチンがサブルーチンを実行する場合、サブルーチンによる処理は親ルーチンのメトリックに組み込まれます。

この表関数が返すカウンターおよび消費時間モニター・エレメントは、サービス・クラスの COLLECT REQUEST METRICS 節によって制御され、データベース・レベルでは mon\_req\_metrics データベース構成パラメーターによって制御されます。どちらの制御も無効になっている場合、報告されるカウンターおよび消費時間モニター・エレメントは 0 です。



動的に準備されたコンパウンド SQL ステートメントのパッケージがパッケージ・キャッシュから除去されると、このルーチンに関する情報は `MON_GET_ROUTINE_DETAILS` 関数によって報告されなくなります。同様に、ルーチンがドロップされると、そのルーチンに関する情報は報告されなくなります。

直前の 24 時間の期間に実行されなかったすべてのルーチンはメモリーから除去され、情報が返されなくなります。

DETAILS 列に戻される XML 文書のスキーマは、ファイル `sqllib/misc/DB2MonRoutines.xsd` で入手できます。詳細は、ファイル `sqllib/misc/DB2MonCommon.xsd` 内にあります。

## 例

TEST.PROC1 ストアド・プロシージャの処理中にどこで時間が消費されたかをリストし、費やされた時間の上位 5 領域を識別します。

```
SELECT B.* FROM TABLE(MON_GET_ROUTINE_DETAILS('P','TEST',NULL,'PROC1',-1))
AS A, TABLE(MON_FORMAT_XML_TIMES_BY_ROW(A.DETAILS)) AS B
ORDER BY TOTAL_TIME_VALUE DESC FETCH FIRST 5 ROWS ONLY
```

戻り値

METRIC_NAME	TOTAL_TIME_VALUE	COUNT	PARENT_METRIC_NAME
TOTAL_RQST_TIME	1055	0	-
TOTAL_ROUTINE_USER_CODE_PROC_TIME	673	0	TOTAL_RQST_TIME
TOTAL_WAIT_TIME	317	-	TOTAL_RQST_TIME
POOL_READ_TIME	236	142	TOTAL_WAIT_TIME
DIRECT_READ_TIME	81	39	TOTAL_WAIT_TIME

5 record(s) selected.

## MON\_GET\_ROUTINE\_EXEC\_LIST 表関数 - ルーチンによって実行されるステートメントのリストの取得

`MON_GET_ROUTINE_EXEC_LIST` 表関数は、データベースの活動化以降に、それぞれのプロシージャ、外部関数、コンパイル済み関数、コンパイル済みトリガー、および無名ブロックによって呼び出されて実行されたすべてのステートメント(セクション) のリストを返します。

## 許可

以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- SQLADM 権限
- DBADM 権限

## デフォルトの PUBLIC 特権

なし

## 構文

```
►►—MON_GET_ROUTINE_EXEC_LIST—(—routine_type—, —routine_schema—, —————→  
►—routine_module_name—, —routine_name—, —member—)—————→◀◀
```

スキーマは SYSPROC です。

## ルーチン・パラメーター

### *routine\_type*

どのタイプのルーチンまたはコンパイル済みトリガーについてデータを返すかを指定する、タイプ CHAR(2) の入力パラメーター。

- "P" - プロシージャ
- "SP" - プロシージャの特定名
- "F" - コンパイル済み関数
- "SF" - コンパイル済み関数の特定名
- "T" - コンパイル済みトリガー
- "C" - SQL PL で動的に準備されたコンパウンド SQL ステートメント、または PL/SQL の無名ブロック

ルーチンのタイプとして "P"、"F"、"T"、または "C" を指定した場合に、入力名を指定しないと、指定したタイプのすべてのルーチンが返されます。ルーチンのタイプが "SP" または "SF" の場合は、特定のルーチン名を指定する必要があります。

ルーチンのタイプが、空ストリング、NULL、またはブランクの場合は、すべてのタイプのすべてのルーチンが返されます。

### *routine\_schema*

ルーチンまたはトリガーのスキーマを指定する、タイプ VARCHAR(128) の入力パラメーター。動的に準備されたコンパウンド SQL ステートメントまたは無名ブロックについては、MON\_GET\_SECTION\_ROUTINES 表関数を使用してスキーマを判別できます。すべてのスキーマのルーチンおよびトリガーを返すには、NULL または空ストリングを使用します。このパラメーターには、大文字と小文字の区別があります。

### *routine\_module\_name*

入力ルーチンのモジュール名を指定する (該当する場合)、タイプ VARCHAR(128) の入力パラメーター。すべてのモジュールのルーチンを返すには、NULL または空ストリングを使用します。このパラメーターには、大文字と小文字の区別があります。

### *routine\_name*

ルーチンの名前を指定する、タイプ VARCHAR (128) の入力パラメーター。入力パラメーターが "SP" または "SF" の場合は、特定のルーチン名を指定する必要があります。動的に準備されたコンパウンド SQL ステートメントまたは無名ブロックについては、MON\_GET\_SECTION\_ROUTINES 表関数を使用して名前を判別できます。その他の入力パラメーターに一致するすべてのルーチンを返すには、NULL または空ストリングを使用します。このパラメーターには、大文字と小文字の区別があります。

### member

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力パラメーター。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 戻される情報

表 158. MON\_GET\_ROUTINE\_EXEC\_LIST について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
ROUTINE_TYPE	CHAR(1)	routine_type - ルーチン・タイプ・モニター・エレメント
ROUTINE_SCHEMA	VARCHAR(128)	routine_schema - ルーチン・スキーマ・モニター・エレメント
ROUTINE_MODULE_NAME	VARCHAR(128)	routine_module_name - ルーチン・モジュール名モニター・エレメント
ROUTINE_NAME	VARCHAR(128)	routine_name - ルーチン名モニター・エレメント
SPECIFIC_NAME	VARCHAR(128)	specific_name - 特定名モニター・エレメント
DYN_COMPOUND_EXEC_ID	VARCHAR(32) FOR BIT DATA	dyn_compound_exec_id - 動的コンパウンド・ステートメント実行可能 ID モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
ROUTINE_ID	INTEGER	routine_id - ルーチン ID モニター・エレメント
SUBROUTINE_ID	INTEGER	subroutine_id - サブルーチン ID モニター・エレメント
LIB_ID	BIGINT	lib_id - ライブラリー ID モニター・エレメント
SECTION_TYPE	CHAR(1)	section_type - セクション・タイプ標識
PACKAGE_SCHEMA	VARCHAR(128)	package_schema - パッケージ・スキーマ。
PACKAGE_NAME	VARCHAR(128)	package_name - パッケージ名。
PACKAGE_VERSION_ID	VARCHAR(64)	package_version_id - パッケージ・バージョン ID。
SECTION_NUMBER	BIGINT	section_number - セクション番号。
STMTNO	INTEGER	stmtno - ステートメント番号モニター・エレメント
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	executable_id - 実行可能 ID
NUM_ROUTINES	INTEGER	num_routines - ルーチンの数
CALL_STMT_ROUTINE_ID	INTEGER	call_stmt_routine_id - Call ステートメント・ルーチン ID モニター・エレメント
CALL_STMT_SUBROUTINE_ID	INTEGER	call_stmt_subroutine_id - Call ステートメント・サブルーチン ID モニター・エレメント

表 158. MON\_GET\_ROUTINE\_EXEC\_LIST について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
NUM_EXECUTIONS	BIGINT	num_executions - ステートメント実行回数
NUM_EXEC_WITH_METRICS	BIGINT	num_exec_with_metrics - メトリック収集を伴う実行数
NUM_COORD_EXEC	BIGINT	num_coord_exec - コーディネーター・エージェントによる実行数
NUM_COORD_EXEC_WITH_METRICS	BIGINT	num_coord_exec_with_metrics - メトリックを伴うコーディネーター・エージェントによる実行数
COORD_STMT_EXEC_TIME	BIGINT	coord_stmt_exec_time - コーディネーター・エージェントによるステートメントの実行時間
TOTAL_ACT_TIME	BIGINT	total_act_time - 合計アクティビティー時間
TOTAL_ACT_WAIT_TIME	BIGINT	total_act_wait_time - 合計アクティビティー待機時間
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - ポストしきい値ソート
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - ポスト共有しきい値ソート
ROWS_READ	BIGINT	rows_read - 読み取り行数
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間
TOTAL_SORTS	BIGINT	total_sorts - ソート合計

### 使用上の注意

MON\_GET\_ROUTINE\_EXEC\_LIST 表関数は、ルーチンまたはトリガーによって実行されるユニークな各ステートメント (セクション) について 1 行ずつ情報を返します。また、この関数は、ルーチンまたはトリガー内でのそのステートメントのすべての実行に渡って集約したメトリックのセットを返します。メンバー全体からの集約は実行されません。しかし、メンバー全体の集約は、SQL 照会を使用して実行できます (「例」セクションを参照)。

集約されるメトリックには、子ステートメントのメトリックは含まれません。例えば、CALL ステートメントがルーチンによって実行された場合、その CALL ステートメントについて MON\_GET\_ROUTINE\_EXEC\_LIST から返されるメトリックには、その CALL で呼び出された他のステートメントが実行する作業は含まれません。唯一の例外はステートメントの経過時間を返す coord\_stmt\_exec\_time モニター・エレメントで、子ステートメントで費やされた時間も暗黙的に収集されます。

ルーチンのステートメント・モニターは、mon\_rtn\_execlist データベース構成パラメーターを使用して明示的に有効にする必要があります。この構成パラメーターを OFF に設定した場合、情報は返されません。

この表関数から返されるカウンターおよび消費時間モニター・エレメントは、ワークロードの COLLECT ACTIVITY METRICS 節によって制御され、データベース・レベルでは mon\_act\_metrics データベース構成パラメーターによって制御されます。どちらの制御も無効になっている場合、報告されるカウンターおよび消費時間モニター・エレメントは 0 です。

動的に準備されたコンパウンド SQL ステートメントまたは無名ブロックのパッケージがパッケージ・キャッシュからクリーンアップされると、このステートメントに関する情報が MON\_GET\_ROUTINE\_EXEC\_LIST 関数によって報告されなくなります。同様に、ルーチンまたはトリガーがドロップされると、そのルーチンまたはトリガーに関する情報は報告されなくなります。

ルーチンによって実行された後にドロップされたステートメントや、直前の 24 時間の期間に実行されなかったステートメントはメモリーから除去され、情報が返されなくなります。

## 例

1. 関数 TEST.X によって実行されたすべてのステートメントをリストします。

```
SELECT SUBSTR(P.STMT_TEXT,1,45) AS TEXT
      FROM TABLE(MON_GET_ROUTINE_EXEC_LIST('F', 'TEST', NULL, 'X', -1))
      AS T, TABLE(MON_GET_PKG_CACHE_STMT(NULL,NULL,-1)) AS P
      WHERE T.EXECUTABLE_ID = P.EXECUTABLE_ID
```

戻り値

```
TEXT
-----
WITH PID_VALUES (WIDEBAND,ENGN_TEMP) AS SEL
insert into GSO_LOGS values(8.7, 145, 1.406)
call SYSIBMSUBROUTINE.TEST_66613_1157394573()
```

3 record(s) selected.

2. プロシージャ TEST.PROC1 内のすべてのステートメントを、それらが消費したコーディネーター・ステートメント実行時間のパーセンテージの順にリストします。

```
SELECT 100*B.COORD_STMT_EXEC_TIME / A.TOTAL_ROUTINE_COORD_EXEC_TIME
      AS PERCENT_EXEC_TIME, SUBSTR(C.STMT_TEXT,1,45)
      AS STMT_TEXT FROM TABLE(MON_GET_ROUTINE('SP',NULL,NULL,'PROC1', -1)) AS A,
      TABLE(MON_GET_ROUTINE_EXEC_LIST('SP',NULL,NULL,'PROC1', -1)) AS B,
      TABLE(MON_GET_PKG_CACHE_STMT(NULL,NULL,-1)) AS C
      WHERE B.EXECUTABLE_ID = C.EXECUTABLE_ID ORDER BY B.COORD_STMT_EXEC_TIME DESC
```

戻り値

```
PERCENT_EXEC_TIME STMT_TEXT
-----
10 SELECT WAFR, MPHX64, ENGN_RPM FROM KB28_LOGS
3 SELECT ( ENGN_TQ * ENGN_RPM )/5252 AS HP FRO
```

2 record(s) selected.

## MON\_GET\_RTS\_RQST 表関数 - リアルタイム統計要求に関する情報の取得

MON\_GET\_RTS\_RQST 表関数は、システム内で保留になっているすべてのリアルタイム統計要求、およびリアルタイム統計デーモンにより現在処理されている (例えば、リアルタイム統計処理キューにある) 要求のセットについての情報を戻します。

### 構文

▶▶—MON\_GET\_RTS\_RQST—( )—▶▶

スキーマは SYSPROC です。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

保留中および現在キューに入っているリアルタイム統計要求、または実行中のリアルタイム統計要求をすべて表示します。

```
SELECT MEMBER
       QUEUE_POSITION,
       REQUEST_STATUS,
       REQUEST_TYPE,
       OBJECT_TYPE,
       VARCHAR(OBJECT_SCHEMA, 10) AS SCHEMA,
       VARCHAR(OBJECT_NAME, 10) AS NAME
FROM TABLE(MON_GET_RTS_REQUEST()) AS T
ORDER BY MEMBER, QUEUE_POSITION ASC
```

以下はこの照会の出力例です。

MEMBER	QUEUE_POSITION	REQUEST_STATUS	REQUEST_TYPE	OBJECT_TYPE	SCHEMA	NAME
0	1	EXECUTING	COLLECT_STATS	TABLE	TEST	EMPLOYEE
0	2	QUEUED	COLLECT_STATS	TABLE	TEST	T1
0	3	QUEUED	WRITE_STATS	TABLE	TEST	T3
0	-	PENDING	WRITE_STATS	TABLE	TEST	BLAH
1	-	PENDING	COLLECT_STATS	TABLE	TEST	DEPT
1	-	PENDING	WRITE_STATS	TABLE	TEST	SALES
2	-	PENDING	WRITE_STATS	TABLE	TEST	SALARY

7 record(s) selected.

## 使用上の注意

MON\_GET\_RTS\_REQUEST 表関数は、すべてのメンバーに関して、保留中のすべてのリアルタイム統計要求についての情報を戻します。リアルタイム統計要求を処理するためのキューは、単一メンバーにのみ存在します。

MON\_GET\_RTS\_REQUEST 表関数は、リアルタイム統計収集が使用可能になっていない場合、リアルタイム統計要求はまったく報告しません。

## 戻される情報

表 159. MON\_GET\_RTS\_REQUEST で戻される情報

列名	データ・タイプ	説明
OBJECT_TYPE	VARCHAR(8)	objtype - オブジェクト・タイプ・モニター・エレメント
OBJECT_SCHEMA	VARCHAR(128)	object_schema - オブジェクト・スキーマ・モニター・エレメント
OBJECT_NAME	VARCHAR(128)	object_name - オブジェクト名モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
REQUEST_TYPE	VARCHAR(14)	以下のいずれか  WRITE_STATS - 既に収集された統計をカタログに書き込む要求。  COLLECT_STATS - 完全統計収集を実行する要求。
REQUEST_STATUS	VARCHAR(10)	以下のいずれか  PENDING - 要求はリアルタイム統計デーモンによってピックアップされるのを待っています。  QUEUED - 要求はリアルタイム統計デーモンによって既に収集され、処理を待っています。  EXECUTING - 要求はリアルタイム統計デーモンによって現在処理されています。
REQUEST_TIME	TIMESTAMP	要求がサブミットされた時刻。
QUEUE_POSITION	INTEGER	REQUEST_STATUS が QUEUED の場合、リアルタイム統計デーモン処理キュー内の要求の位置。
QUEUE_ENTRY_TIME	TIMESTAMP	REQUEST_STATUS が QUEUED の場合、リアルタイム統計デーモン処理キューに要求が追加された時刻。
EXECUTION_START_TIME	TIMESTAMP	REQUEST_STATUS が EXECUTING の場合、要求が実行を開始した時刻。その他の場合は NULL。

## MON\_GET\_SECTION\_ROUTINE 表関数 - 入力セクションのルーチンのリストの取得

MON\_GET\_SECTION\_ROUTINE 表関数は、入力セクションの実行中に呼び出される可能性のあるすべてのプロシージャ、外部関数、コンパイル済み関数、およびコンパイル済みトリガーのリストを返します。

## 許可

以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- SQLADM 権限
- DBADM 権限

## デフォルトの PUBLIC 特権

なし

## 構文

```
►►—MON_GET_SECTION_ROUTINE—(—executable_id—)—————►►
```

スキーマは SYSPROC です。

## ルーチン・パラメーター

*executable\_id*

どのセクションについてルーチンのリストを返すかをユニークに識別する、タイプ VARCHAR(32) FOR BIT DATA の入力パラメーター。この引数が NULL または空ストリングである場合、あるいはセクションがメモリー内に見つからない場合は、SQL0171 が返されます。

## 戻される情報

表 160. MON\_GET\_SECTION\_ROUTINE について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
ROUTINE_TYPE	CHAR(1)	routine_type - ルーチン・タイプ・モニター・エレメント
ROUTINE_SCHEMA	VARCHAR(128)	routine_schema - ルーチン・スキーマ・モニター・エレメント
ROUTINE_MODULE_NAME	VARCHAR(128)	routine_module_name - ルーチン・モジュール名モニター・エレメント
ROUTINE_NAME	VARCHAR(128)	routine_name - ルーチン名モニター・エレメント
SPECIFIC_NAME	VARCHAR(128)	specific_name - 特定名モニター・エレメント
ROUTINE_ID	INTEGER	routine_id - ルーチン ID モニター・エレメント
SUBROUTINE_ID	INTEGER	subroutine_id - サブルーチン ID モニター・エレメント
LIB_ID	BIGINT	lib_id - ライブラリー ID モニター・エレメント



## 使用上の注意

入力セクションは、パッケージ・キャッシュ内になければなりません。

この関数によって返されるオブジェクトは、必ずしも入力セクションの実行時に呼び出されるとは限りません。例えば、次の SELECT ステートメントで、ステートメント FUNCTION1 はコンパイル済み関数です。

```
SELECT CASE WHEN COL1 = 0 THEN FUNCTION1() ELSE NULL END AS T1 FROM TABLE1
```

MON\_GET\_SECTION\_ROUTINE 表関数は、引数として渡された executable\_id がこのステートメントのセクションを識別している場合に、FUNCTION1 関数の行を返します。しかし、このステートメントの実行時に FUNCTION1 が呼び出されるのは、TABLE1 のいずれかの行で COL1 の値が 0 になっている場合のみです。

## 例

### 1. 実行可能 ID

x'0100000000000005201000000000001000000010020120822205618607103' を持つパッケージ・キャッシュ内のセクションによって実行される可能性のあるすべてのルーチンをリストします。

```
SELECT * FROM
  TABLE(MON_GET_SECTION_ROUTINE(
    x'0100000000000005201000000000001000000010020120822205618607103'))
  AS T
```

戻り値

ROUTINE_TYPE	ROUTINE_SCHEMA	ROUTINE_MODULE_NAME	ROUTINE_NAME	...
F	SPEEDY	TANKS	ANALYZE_LOG1	...
P	SPEEDY	TANKS	POST_PROCESS	...

2 record(s) selected.

### 2. 実行可能 ID

x'0100000000000005201000000000001000000010020120822205618607103' を持つ INSERT ステートメントが長時間に渡って実行されていることにユーザーが気付きます。次のステートメントを実行して、その INSERT の実行中にコンパイル済みトリガーが呼び出されたかどうかを確認します。

```
SELECT * FROM
  TABLE(MON_GET_SECTION_ROUTINE(
    x'0100000000000005201000000000001000000010020120822205618607103'))
  AS T WHERE ROUTINETYPE = 'T'
```

戻り値

ROUTINE_TYPE	ROUTINE_SCHEMA	ROUTINE_MODULE_NAME	ROUTINE_NAME	...
T	MIKE		INVALID_WELD_VALUE	...
P	MIKE		INSERT_WELD_SPECS	...

2 record(s) selected.

### 3. 実行可能 ID

x'0100000000000005201000000000001000000010020120822205618607103' を持つ無名ブロックについて内部スキーマとプロシージャ名を調べます。

```
SELECT ROUTINESCHEMA, ROUTINENAME, SPECIFICNAME
  FROM TABLE(MON_GET_SECTION_ROUTINE(
    x'0100000000000005201000000000001000000010020120822205618607103'))
  AS T
```

戻り値

```
ROUTINE_SCHEMA ROUTINE_NAME SPECIFIC_NAME ...
-----
GSOSchema PING_RUBBLE SQL120801135351900 ...
1 record(s) selected.
```

## MON\_GET\_SERVERLIST 表関数 - メンバー優先順位の詳細の取得

MON\_GET\_SERVERLIST 表関数は、1 つ以上のメンバーにキャッシュされている、現在接続されているデータベースのサーバー・リストに関するメトリックを戻します。

アクティブ・メンバーごとに、接続情報と優先順位情報がサーバー・リストに含まれます。これらの情報により、リモート・クライアントはワークロード・バランシング (WLB) と自動クライアント・リルート (ACR) を実行できます。member パラメーターは、サーバー・リストがキャッシュされているメンバーに該当します。

### 構文

```
►► MON_GET_SERVERLIST ( (member) ) ◀◀
```

スキーマは SYSPROC です。

### 表関数パラメーター

*member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーの場合は -1、すべてのデータベース・メンバーの場合は -2 を指定します。つまり、-1 は現行メンバーのみに対して db2pd -serverlist -db dbname (dbname は現在の接続先データベースの名前) を発行することと同等であり、-2 はすべてのデータベース・メンバーに対して同じコマンドを発行することと同等です。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

例 1: メンバー 0 上のデータベース SAMPLE への接続を想定し、このデータベースがリモート・クライアントによってアクセスされたとします。このデータベースに関する、このメンバーにキャッシュされているサーバー・リストのすべてのメトリックをリストします。

```
SELECT MEMBER, CACHED_TIMESTAMP, VARCHAR(HOSTNAME, 30)
AS HOSTNAME, PORT_NUMBER, SSL_PORT_NUMBER, PRIORITY
FROM TABLE (MON_GET_SERVERLIST (-1))
```

メンバー 0 上のデータベース SAMPLE への接続に関する出力結果。

```
MEMBER CACHED_TIMESTAMP          HOSTNAME          ...
-----
0 2011-02-19-17.39.33.000000 coralxib23.torolab.ibm.com ...
0 2011-02-19-17.39.33.000000 coralxib24.torolab.ibm.com ...
```

照会の出力 (続き)。

```
... PORT_NUMBER SSL_PORT_NUMBER  PRIORITY
... -----
...      49712          0              67
...      49712          0              32
```

例 2: メンバー 0 上のデータベース SAMPLE への接続を想定し、メンバー 0 とメンバー 1 の両方でこのデータベースがリモート・クライアントによってアクセスされたとします。このデータベースに関する、両方のメンバーでキャッシュされているサーバー・リストのすべてのメトリックをリストします。

```
SELECT MEMBER, CACHED_TIMESTAMP, VARCHAR(HOSTNAME, 30)
AS HOSTNAME, PORT_NUMBER, SSL_PORT_NUMBER, PRIORITY
FROM TABLE (MON_GET_SERVERLIST (-2))
```

(メンバー 0 とメンバー 1 で現在アクティブな) メンバー 0 上のデータベース SAMPLE への接続に関する出力結果。

```
MEMBER CACHED_TIMESTAMP          HOSTNAME          ...
-----
0 2011-02-19-17.39.33.000000 coralxib23.torolab.ibm.com ...
0 2011-02-19-17.39.33.000000 coralxib24.torolab.ibm.com ...
1 2011-02-19-17.39.33.000000 coralxib24.torolab.ibm.com ...
1 2011-02-19-17.39.33.000000 coralxib23.torolab.ibm.com ...
```

照会の出力 (続き)。

```
... PORT_NUMBER SSL_PORT_NUMBER  PRIORITY
... -----
...      49712          0              67
...      49712          0              32
...      49712          0              32
...      49712          0              67
```

## 戻される情報

表 161. MON\_GET\_SERVERLIST で戻される情報

列名	データ・タイプ	説明
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
CACHED_TIMESTAMP	TIMESTAMP	cached_timestamp - キャッシュ・タイム・スタンプ
HOSTNAME	VARCHAR(255)	hostname - ホスト名 : モニター・エレメント
PORT_NUMBER	INTEGER	port_number - ポート番号
SSL_PORT_NUMBER	INTEGER	ssl_port_number - SSL ポート番号
PRIORITY	SMALLINT	priority - 優先順位の値

## MON\_GET\_SERVICE\_SUBCLASS 表関数 - サービス・サブクラス のメトリックの取得

MON\_GET\_SERVICE\_SUBCLASS 表関数は、1 つ以上のサービス・サブクラスのメトリックを戻します。

### 構文

```
▶▶—MON_GET_SERVICE_SUBCLASS—(—service_superclass_name—,—————▶▶  
▶—service_subclass_name—,—member—)—————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *service\_superclass\_name*

この関数を呼び出すときに現在接続されているデータベースで有効なサービス・スーパークラス名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのスーパークラスについてメトリックが取得されます。

#### *service\_subclass\_name*

この関数を呼び出すときに現在接続されているデータベースで有効なサービス・サブクラス名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのサブクラスについてメトリックが取得されます。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

CPU 使用率で順序付けられる、使用された合計 CPU 時間と、各サービス・クラスの処理された要求の合計数を表示します。

```

SELECT varchar(service_superclass_name,30) as service_superclass,
       varchar(service_subclass_name,30) as service_subclass,
       sum(total_cpu_time) as total_cpu,
       sum(app_rqsts_completed_total) as total_rqsts
FROM TABLE(MON_GET_SERVICE_SUBCLASS('','",-2)) AS t
GROUP BY service_superclass_name, service_subclass_name
ORDER BY total_cpu desc

```

以下はこの照会の出力例です。

```

SERVICE_SUPERCLASS      SERVICE_SUBCLASS      ...
-----
SYSDEFAULTUSERCLASS      SYSDEFAULTSUBCLASS    ...
SYSDEFAULTMAINTENANCECLASS  SYSDEFAULTSUBCLASS    ...
SYSDEFAULTSYSTEMCLASS      SYSDEFAULTSUBCLASS    ...

```

3 record(s) selected.

照会の出力 (続き)。

```

... TOTAL_CPU      TOTAL_RQSTS
... -----
...          967673          100
...          0              0
...          0              0

```

### 使用上の注意

MON\_GET\_SERVICE\_SUBCLASS 表関数で戻されるメトリックは、指示されたサービス・サブクラスの下で実行された要求についてのすべてのメトリックの累計を表します。メトリックは、作業単位境界でサービス・クラスにロールアップされ、要求の実行中には定期的にロールアップされます。したがって、この表関数で報告される値は、直前の ROLLUP 時のシステムの現行状態を反映しています。メトリックの値は確実に増加します。ある時間間隔に対する指定されたメトリックの値を判別するには、MON\_GET\_SERVICE\_SUBCLASS 表関数を使用してその間隔の始めと終わりのメトリックを照会し、差異を計算します。

要求メトリックは、サービス・スーパークラスに対する COLLECT REQUEST METRICS 節、およびデータベース・レベルの *mon\_req\_metrics* データベース構成パラメーターを介して制御されます。親サービス・スーパークラスで要求メトリックを使用可能にしているサービス・サブクラスのエージェントが要求を処理する場合にのみ、またはデータベース全体で要求メトリック・コレクションが有効な場合にのみ、その要求に対しメトリックが収集されます。デフォルトでは、要求メトリックはデータベース・レベルで使用可能です。要求メトリックがデータベース・レベルでもサービス・スーパークラスに対しても使用不可になっている場合、そのサービス・スーパークラスにマッピングされた接続ごとに報告されるメトリックは増加を停止します (または、要求メトリックがデータベースのアクティブ化時に無効であった場合には 0 のままになります)。

MON\_GET\_SERVICE\_SUBCLASS 表関数は、サービス・サブクラスごとおよびメンバーごとに 1 行のデータを戻します。(メンバー上の) サービス・クラス全体または (1 つ以上のサービス・クラスの) メンバー全体の集約は実行されません。ただし、集約は例に示されるように SQL 照会を使用して実行できます。この入力パラメーターの影響として、ANDing されます。したがって、競合する入力パラメーター (例えば、スーパークラス名 SUPA と SUPA のサブクラスではないサブクラス名 SUBB など) を指定する場合、行は戻されません。

**ヒント:** 要求は、複数のサービス・サブクラスで実行される場合があります。例えば、REMAP ACTIVITY アクションによってワークロード・マネージャー (WLM) しきい値を使用して要求を 1 つのサービス・サブクラスから別のサービス・サブクラスへマッピングする場合、この状況が生じる可能性があります。メトリックに使用された時間は、要求が実行されるサービス・サブクラスごとに更新されますが、要求カウンターは、要求が完了したサービス・サブクラスに対して増分します。したがって、単一のサブクラスの要求時間の平均を分析する必要はありません。アクティビティをマッピングできるすべてのサブクラスは相互に関連付けて分析する必要があります。例えば、サービス・サブクラス A からサービス・クラス B にアクティビティをマッピングできるしきい値があり、要求の平均を計算する場合、サービス・サブサービス A および B についてのカウンターおよびメトリックを集約し、その集約を使用して平均を計算する必要があります。

### 戻される情報

**注:** スーパークラス SYSDEFAULTSYSTEMCLASS の下のデフォルト・サブクラス SYSDEFAULTSUBCLASS を記述する、この関数によって戻される行では、メトリックを報告するすべての列の値が 0 になります。

表 162. MON\_GET\_SERVICE\_SUBCLASS について戻される情報

列名	データ・タイプ	説明
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - サービス・スーパークラス名モニター・エレメント
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - サービス・スーパークラス名モニター・エレメント
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - サービス・サブクラス名モニター・エレメント
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - サービス・サブクラス名モニター・エレメント
SERVICE_CLASS_ID	INTEGER	service_class_id - サービス・クラス ID モニター・エレメント
MEMBER	INTEGER	member - データベース・メンバー・モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
ACT_ABORTED_TOTAL	BIGINT	act_aborted_total - 打ち切られたアクティビティの合計数モニター・エレメント
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 完了したアクティビティの合計数モニター・エレメント
ACT_REJECTED_TOTAL	BIGINT	act_rejected_total - リジェクトされたアクティビティの合計数モニター・エレメント
AGENT_WAIT_TIME	BIGINT	agent_wait_time - エージェント待ち時間モニター・エレメント
AGENT_WAITS_TOTAL	BIGINT	agent_waits_total - エージェント待機の合計モニター・エレメント
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファー・プール・データの論理読み取りモニター・エレメント

表 162. MON\_GET\_SERVICE\_SUBCLASS について戻される情報 (続き)

列名	データ・タイプ	説明
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取りモニター・エレメント
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取りモニター・エレメント
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取りモニター・エレメント
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取りモニター・エレメント
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取りモニター・エレメント
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取りモニター・エレメント
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取りモニター・エレメント
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取りモニター・エレメント
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取りモニター・エレメント
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取りモニター・エレメント
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取りモニター・エレメント
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込みモニター・エレメント
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込みモニター・エレメント
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDA データ書き込みモニター・エレメント
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計モニター・エレメント
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計モニター・エレメント
CLIENT_IDLE_WAIT_TIME	BIGINT	client_idle_wait_time - クライアント・アイドル待ち時間モニター・エレメント
DEADLOCKS	BIGINT	deadlocks - デッドロック検出数モニター・エレメント
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取りモニター・エレメント
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間モニター・エレメント
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込みモニター・エレメント
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間モニター・エレメント

表 162. MON\_GET\_SERVICE\_SUBCLASS について戻される情報 (続き)

列名	データ・タイプ	説明
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求モニター・エレメント
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求モニター・エレメント
FCM_RECV_VOLUME	BIGINT	fcm_recv_volume - FCM 受信ボリューム・モニター・エレメント
FCM_RECVS_TOTAL	BIGINT	fcm_recvs_total - FCM 合計受信数モニター・エレメント
FCM_SEND_VOLUME	BIGINT	fcm_send_volume - FCM 送信ボリューム・モニター・エレメント
FCM_SENDS_TOTAL	BIGINT	fcm_sends_total - FCM 合計送信数モニター・エレメント
FCM_RECV_WAIT_TIME	BIGINT	fcm_recv_wait_time - FCM 受信待ち時間モニター・エレメント
FCM_SEND_WAIT_TIME	BIGINT	fcm_send_wait_time - FCM 送信待ち時間モニター・エレメント
IPC_RECV_VOLUME	BIGINT	ipc_recv_volume - プロセス間通信受信ボリューム・モニター・エレメント
IPC_RECV_WAIT_TIME	BIGINT	ipc_recv_wait_time - プロセス間通信受信待ち時間モニター・エレメント
IPC_RECVS_TOTAL	BIGINT	ipc_recvs_total - プロセス間通信合計受信数モニター・エレメント
IPC_SEND_VOLUME	BIGINT	ipc_send_volume - プロセス間通信送信ボリューム・モニター・エレメント
IPC_SEND_WAIT_TIME	BIGINT	ipc_send_wait_time - プロセス間通信送信待ち時間モニター・エレメント
IPC_SENDS_TOTAL	BIGINT	ipc_sends_total - プロセス間通信合計送信数モニター・エレメント
LOCK_ESCALS	BIGINT	lock_escals - ロック・エスカレーション数モニター・エレメント
LOCK_TIMEOUTS	BIGINT	lock_timeouts - ロック・タイムアウト数モニター・エレメント
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間モニター・エレメント
LOCK_WAITS	BIGINT	lock_waits - ロック待機数モニター・エレメント
LOG_BUFFER_WAIT_TIME	BIGINT	log_buffer_wait_time - ログ・バッファ待ち時間モニター・エレメント
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - エージェントの待機が発生したフル・ログ・バッファの回数モニター・エレメント
LOG_DISK_WAIT_TIME	BIGINT	log_disk_wait_time - ログ・ディスク待ち時間モニター・エレメント
LOG_DISK_WAITS_TOTAL	BIGINT	log_disk_waits_total - ログ・ディスク待機の合計モニター・エレメント



表 162. MON\_GET\_SERVICE\_SUBCLASS について戻される情報 (続き)

列名	データ・タイプ	説明
RQSTS_COMPLETED_TOTAL	BIGINT	rqsts_completed_total - 完了した要求の合計数モニター・エレメント
ROWS_MODIFIED	BIGINT	rows_modified - 変更された行数モニター・エレメント
ROWS_READ	BIGINT	rows_read - 読み取り行数モニター・エレメント
ROWS_RETURNED	BIGINT	rows_returned - 戻り行数モニター・エレメント
TCPIP_RECV_VOLUME	BIGINT	tcPIP_recv_volume - TCP/IP 受信ボリューム・モニター・エレメント
TCPIP_SEND_VOLUME	BIGINT	tcPIP_send_volume - TCP/IP 送信ボリューム・モニター・エレメント
TCPIP_RECV_WAIT_TIME	BIGINT	tcPIP_recv_wait_time - TCP/IP 受信待ち時間モニター・エレメント
TCPIP_RECVS_TOTAL	BIGINT	tcPIP_recvs_total - TCP/IP 合計受信数モニター・エレメント
TCPIP_SEND_WAIT_TIME	BIGINT	tcPIP_send_wait_time - TCP/IP 送信待ち時間モニター・エレメント
TCPIP_SENDS_TOTAL	BIGINT	tcPIP_sends_total - TCP/IP 合計送信数モニター・エレメント
TOTAL_APP_RQST_TIME	BIGINT	total_app_rqst_time - アプリケーション要求合計時間モニター・エレメント
TOTAL_RQST_TIME	BIGINT	total_rqst_time - 要求合計時間モニター・エレメント
WLM_QUEUE_TIME_TOTAL	BIGINT	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間モニター・エレメント
WLM_QUEUE_ASSIGNMENTS_TOTAL	BIGINT	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数モニター・エレメント
TOTAL_RQST_MAPPED_IN	BIGINT	total_rqst_mapped_in - マップで含められた要求の合計モニター・エレメント
TOTAL_RQST_MAPPED_OUT	BIGINT	total_rqst_mapped_out - マップで除外された要求の合計モニター・エレメント
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間 : モニター・エレメント
TOTAL_WAIT_TIME	BIGINT	total_wait_time - 合計待ち時間モニター・エレメント
APP_RQSTS_COMPLETED_TOTAL	BIGINT	app_rqsts_completed_total - 完了したアプリケーション要求の合計数モニター・エレメント
TOTAL_SECTION_SORT_TIME	BIGINT	total_section_sort_time - セクション・ソート時間合計モニター・エレメント
TOTAL_SECTION_SORT_PROC_TIME	BIGINT	total_section_sort_proc_time - セクション・ソート処理時間合計モニター・エレメント
TOTAL_SECTION_SORTS	BIGINT	total_section_sorts - セクション・ソート合計モニター・エレメント
TOTAL_SORTS	BIGINT	total_sorts - ソート合計モニター・エレメント

表 162. MON\_GET\_SERVICE\_SUBCLASS について戻される情報 (続き)

列名	データ・タイプ	説明
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - ポストしきい値ソート・モニター・エレメント
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - ポスト共有しきい値ソート・モニター・エレメント
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー・モニター・エレメント
TOTAL_COMPILE_TIME	BIGINT	total_compile_time - コンパイル時間の合計モニター・エレメント
TOTAL_COMPILE_PROC_TIME	BIGINT	total_compile_proc_time - コンパイル処理時間の合計モニター・エレメント
TOTAL_COMPILATIONS	BIGINT	total_compilations - コンパイルの合計回数モニター・エレメント
TOTAL_IMPLICIT_COMPILE_TIME	BIGINT	total_implicit_compile_time - 暗黙的コンパイル時間の合計モニター・エレメント
TOTAL_IMPLICIT_COMPILE_PROC_TIME	BIGINT	total_implicit_compile_proc_time - 暗黙的コンパイルの処理時間の合計モニター・エレメント
TOTAL_IMPLICIT_COMPILATIONS	BIGINT	total_implicit_compilations - 暗黙的コンパイルの合計回数モニター・エレメント
TOTAL_SECTION_TIME	BIGINT	total_section_time - セクション時間の合計モニター・エレメント
TOTAL_SECTION_PROC_TIME	BIGINT	total_section_proc_time - セクション処理時間の合計モニター・エレメント
TOTAL_APP_SECTION_EXECUTIONS	BIGINT	total_app_section_executions - アプリケーションのセクション実行の合計回数モニター・エレメント
TOTAL_ACT_TIME	BIGINT	total_act_time - 合計アクティビティー時間モニター・エレメント
TOTAL_ACT_WAIT_TIME	BIGINT	total_act_wait_time - 合計アクティビティー待機時間モニター・エレメント
ACT_RQSTS_TOTAL	BIGINT	act_rqsts_total - アクティビティー要求の合計モニター・エレメント
TOTAL_ROUTINE_TIME	BIGINT	total_routine_time - ルーチン時間の合計モニター・エレメント
TOTAL_ROUTINE_INVOCATIONS	BIGINT	total_routine_invocations - ルーチン呼び出しの合計回数モニター・エレメント
TOTAL_COMMIT_TIME	BIGINT	total_commit_time - コミット時間の合計モニター・エレメント
TOTAL_COMMIT_PROC_TIME	BIGINT	total_commit_proc_time - コミット処理時間の合計モニター・エレメント
TOTAL_APP_COMMITS	BIGINT	total_app_commits - アプリケーション・コミットの合計回数モニター・エレメント
INT_COMMITS	BIGINT	int_commits - 内部コミット数モニター・エレメント
TOTAL_ROLLBACK_TIME	BIGINT	total_rollback_time - ロールバック時間の合計モニター・エレメント

表 162. MON\_GET\_SERVICE\_SUBCLASS について戻される情報 (続き)

列名	データ・タイプ	説明
TOTAL_ROLLBACK_PROC_TIME	BIGINT	total_rollback_proc_time - ロールバック処理時間の合計モニター・エレメント
TOTAL_APP_ROLLBACKS	BIGINT	total_app_rollback - アプリケーション・ロールバックの合計回数モニター・エレメント
INT_ROLLBACKS	BIGINT	int_rollback - 内部ロールバック数モニター・エレメント
TOTAL_RUNSTATS_TIME	BIGINT	total_runstats_time - ランタイム統計の時間の合計モニター・エレメント
TOTAL_RUNSTATS_PROC_TIME	BIGINT	total_runstats_proc_time - ランタイム統計の処理時間の合計モニター・エレメント
TOTAL_RUNSTATS	BIGINT	total_runstats - ランタイム統計の合計回数モニター・エレメント
TOTAL_REORG_TIME	BIGINT	total_reorg_time - 再編成時間の合計モニター・エレメント
TOTAL_REORG_PROC_TIME	BIGINT	total_reorg_proc_time - 再編成の処理時間の合計モニター・エレメント
TOTAL_REORGS	BIGINT	total_reorgs - 再編成の合計回数モニター・エレメント
TOTAL_LOAD_TIME	BIGINT	total_load_time - ロード時間の合計モニター・エレメント
TOTAL_LOAD_PROC_TIME	BIGINT	total_load_proc_time - ロード処理時間の合計モニター・エレメント
TOTAL_LOADS	BIGINT	total_loads - ロードの合計回数モニター・エレメント
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - カタログ・キャッシュ挿入数モニター・エレメント
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - カタログ・キャッシュ参照数モニター・エレメント
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - パッケージ・キャッシュ挿入モニター・エレメント
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - パッケージ・キャッシュ参照モニター・エレメント
THRESH_VIOLATIONS	BIGINT	thresh_violations - しきい値違反の回数モニター・エレメント
NUM_LW_THRESH_EXCEEDED	BIGINT	num_lw_thresh_exceeded - ロック待機しきい値超過回数モニター・エレメント
LOCK_WAITS_GLOBAL	BIGINT	lock_waits_global - ロック待機数 (グローバル) モニター・エレメント
LOCK_WAIT_TIME_GLOBAL	BIGINT	lock_wait_time_global - ロックの待機時間 (グローバル) モニター・エレメント
LOCK_TIMEOUTS_GLOBAL	BIGINT	lock_timeouts_global - ロックのタイムアウト数 (グローバル) モニター・エレメント
LOCK_ESCALS_MAXLOCKS	BIGINT	lock_escal_maxlocks - maxlocks ロック・エスカレーション数モニター・エレメント

表 162. MON\_GET\_SERVICE\_SUBCLASS について戻される情報 (続き)

列名	データ・タイプ	説明
LOCK_ESCALS_LOCKLIST	BIGINT	lock_escals_locklist - locklist ロック・エスカレーション数モニター・エレメント
LOCK_ESCALS_GLOBAL	BIGINT	lock_escals_global - グローバル・ロック・エスカレーション数モニター・エレメント
RECLAIM_WAIT_TIME	BIGINT	reclaim_wait_time - 再利用の待機時間モニター・エレメント
SPACEMAPPAGE_RECLAIM_WAIT_TIME	BIGINT	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間モニター・エレメント
CF_WAITS	BIGINT	cf_waits - クラスタ・キャッシング・ファシリティー待機回数モニター・エレメント
CF_WAIT_TIME	BIGINT	cf_wait_time - クラスタ・キャッシング・ファシリティー待機時間モニター・エレメント
POOL_DATA_GBP_L_READS	BIGINT	pool_data_gbp_l_reads - グループ・バッファ・プールのデータの論理読み取りモニター・エレメント
POOL_DATA_GBP_P_READS	BIGINT	pool_data_gbp_p_reads - グループ・バッファ・プールのデータの物理読み取りモニター・エレメント
POOL_DATA_LBP_PAGES_FOUND	BIGINT	pool_data_lbp_pages_found - ローカル・バッファ・プールで検出されたデータ・ページ・モニター・エレメント
POOL_DATA_GBP_INVALID_PAGES	BIGINT	pool_data_gbp_invalid_pages - グループ・バッファ・プールの無効なデータ・ページ・モニター・エレメント
POOL_INDEX_GBP_L_READS	BIGINT	pool_index_gbp_l_reads - グループ・バッファ・プールの索引の論理読み取りモニター・エレメント
POOL_INDEX_GBP_P_READS	BIGINT	pool_index_gbp_p_reads - グループ・バッファ・プールの索引の物理読み取りモニター・エレメント
POOL_INDEX_LBP_PAGES_FOUND	BIGINT	pool_index_lbp_pages_found - 検出されたローカル・バッファ・プールの索引ページ・モニター・エレメント
POOL_INDEX_GBP_INVALID_PAGES	BIGINT	pool_index_gbp_invalid_pages - グループ・バッファ・プールの無効な索引ページ・モニター・エレメント
POOL_XDA_GBP_L_READS	BIGINT	pool_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの論理読み取りの要求モニター・エレメント
POOL_XDA_GBP_P_READS	BIGINT	pool_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの物理読み取りの要求モニター・エレメント
POOL_XDA_LBP_PAGES_FOUND	BIGINT	pool_xda_lbp_pages_found - 検出されたローカル・バッファ・プールの XDA データ・ページ・モニター・エレメント
POOL_XDA_GBP_INVALID_PAGES	BIGINT	pool_xda_gbp_invalid_pages - グループ・バッファ・プールの無効な XDA データ・ページ・モニター・エレメント

表 162. MON\_GET\_SERVICE\_SUBCLASS について戻される情報 (続き)

列名	データ・タイプ	説明
AUDIT_EVENTS_TOTAL	BIGINT	audit_events_total - 監査イベント合計数モニター・エレメント
AUDIT_FILE_WRITES_TOTAL	BIGINT	audit_file_writes_total - 監査ファイル書き込み合計回数モニター・エレメント
AUDIT_FILE_WRITE_WAIT_TIME	BIGINT	audit_file_write_wait_time - 監査ファイル書き込み待機時間モニター・エレメント
AUDIT_SUBSYSTEM_WAITS_TOTAL	BIGINT	audit_subsystem_waits_total - 監査サブシステム待機の合計モニター・エレメント
AUDIT_SUBSYSTEM_WAIT_TIME	BIGINT	audit_subsystem_wait_time - 監査サブシステム待機時間モニター・エレメント
DIAGLOG_WRITES_TOTAL	BIGINT	diaglog_writes_total - 診断ログ・ファイル書き込み合計回数モニター・エレメント
DIAGLOG_WRITE_WAIT_TIME	BIGINT	diaglog_write_wait_time - 診断ログ・ファイル書き込み待機時間モニター・エレメント
FCM_MESSAGE_RECVS_TOTAL	BIGINT	fcm_message_recvs_total - FCM メッセージ受信の合計モニター・エレメント
FCM_MESSAGE_RECV_VOLUME	BIGINT	fcm_message_recv_volume - FCM メッセージ受信ボリューム・モニター・エレメント
FCM_MESSAGE_RECV_WAIT_TIME	BIGINT	fcm_message_recv_wait_time - FCM メッセージ受信待ち時間モニター・エレメント
FCM_MESSAGE_SENDS_TOTAL	BIGINT	fcm_message_sends_total - FCM メッセージ送信の合計モニター・エレメント
FCM_MESSAGE_SEND_VOLUME	BIGINT	fcm_message_send_volume - FCM メッセージ送信ボリューム・モニター・エレメント
FCM_MESSAGE_SEND_WAIT_TIME	BIGINT	fcm_message_send_wait_time - FCM メッセージ送信待ち時間モニター・エレメント
FCM_TQ_RECVS_TOTAL	BIGINT	fcm_tq_recvs_total - FCM 表キュー受信の合計モニター・エレメント
FCM_TQ_RECV_VOLUME	BIGINT	fcm_tq_recv_volume - FCM 表キュー受信ボリュームのモニター・エレメント
FCM_TQ_RECV_WAIT_TIME	BIGINT	fcm_tq_recv_wait_time - FCM 表キュー受信待機時間モニター・エレメント
FCM_TQ_SENDS_TOTAL	BIGINT	fcm_tq_sends_total - FCM 表キュー送信の合計モニター・エレメント
FCM_TQ_SEND_VOLUME	BIGINT	fcm_tq_send_volume - FCM 表キュー送信ボリュームのモニター・エレメント
FCM_TQ_SEND_WAIT_TIME	BIGINT	fcm_tq_send_wait_time - FCM 表キュー送信待機時間モニター・エレメント
TOTAL_ROUTINE_USER_CODE_PROC_TIME	BIGINT	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計モニター・エレメント
TOTAL_ROUTINE_USER_CODE_TIME	BIGINT	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計モニター・エレメント
TQ_TOT_SEND_SPILLS	BIGINT	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数モニター・エレメント

表 162. MON\_GET\_SERVICE\_SUBCLASS について戻される情報 (続き)

列名	データ・タイプ	説明
EVMON_WAIT_TIME	BIGINT	evmon_wait_time - イベント・モニターの待機時間モニター・エレメント
EVMON_WAITS_TOTAL	BIGINT	evmon_waits_total - イベント・モニターの待機の合計回数モニター・エレメント
TOTAL_EXTENDED_LATCH_WAIT_TIME	BIGINT	total_extended_latch_wait_time - 拡張ラッチの合計待機時間のモニター・エレメント
TOTAL_EXTENDED_LATCH_WAITS	BIGINT	total_extended_latch_waits - 拡張ラッチの合計待機回数のモニター・エレメント
TOTAL_STATS_FABRICATION_TIME	BIGINT	total_stats_fabrication_time - 統計作成の合計時間モニター・エレメント
TOTAL_STATS_FABRICATION_PROC_TIME	BIGINT	total_stats_fabrication_proc_time - 統計作成処理時間の合計モニター・エレメント
TOTAL_STATS_FABRICATIONS	BIGINT	total_stats_fabrications - 統計作成の合計回数モニター・エレメント
TOTAL_SYNC_RUNSTATS_TIME	BIGINT	total_sync_runstats_time - 同期 RUNSTATS 時間の合計モニター・エレメント
TOTAL_SYNC_RUNSTATS_PROC_TIME	BIGINT	total_sync_runstats_proc_time - 同期 RUNSTATS 処理時間の合計モニター・エレメント
TOTAL_SYNC_RUNSTATS	BIGINT	total_sync_runstats - 同期 RUNSTATS アクティビティの合計モニター・エレメント
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - ディスパッチャーの合計実行キュー時間 : モニター・エレメント
TOTAL_PEDS	BIGINT	total_peds - partial early distinct の合計モニター・エレメント
DISABLED_PEDS	BIGINT	disabled_peds - 無効な partial early distinct の数モニター・エレメント
POST_THRESHOLD_PEDS	BIGINT	post_threshold_peds - partial early distinct のしきい値モニター・エレメント
TOTAL_PEAS	BIGINT	total_peas - partial early aggregation の合計モニター・エレメント
POST_THRESHOLD_PEAS	BIGINT	post_threshold_peas - partial early aggregation のしきい値モニター・エレメント
TQ_SORT_HEAP_REQUESTS	BIGINT	tq_sort_heap_requests - 表キューのソート・ヒープの要求モニター・エレメント
TQ_SORT_HEAP_REJECTIONS	BIGINT	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否モニター・エレメント
POOL_QUEUED_ASYNC_DATA_REQS	BIGINT	pool_queued_async_data_reqs - データ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_INDEX_REQS	BIGINT	pool_queued_async_index_reqs - 索引プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_XDA_REQS	BIGINT	pool_queued_async_xda_reqs - XDA プリフェッチ要求モニター・エレメント

表 162. MON\_GET\_SERVICE\_SUBCLASS について戻される情報 (続き)

列名	データ・タイプ	説明
POOL_QUEUED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_OTHER_REQS	BIGINT	pool_queued_async_other_reqs - 非プリフェッチの要求モニター・エレメント
POOL_QUEUED_ASYNC_DATA_PAGES	BIGINT	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_INDEX_PAGES	BIGINT	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_XDA_PAGES	BIGINT	pool_queued_async_xda_pages - XDA ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_DATA_PAGES	BIGINT	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_INDEX_PAGES	BIGINT	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_XDA_PAGES	BIGINT	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_DATA_REQS	BIGINT	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_INDEX_REQS	BIGINT	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_XDA_REQS	BIGINT	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_OTHER_REQS	BIGINT	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求モニター・エレメント
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間モニター・エレメント

表 162. MON\_GET\_SERVICE\_SUBCLASS について戻される情報 (続き)

列名	データ・タイプ	説明
PREFETCH_WAITS	BIGINT	prefetch_waits - プリフェッチャーの待機カウント・モニター・エレメント
APP_ACT_COMPLETED_TOTAL	BIGINT	app_act_completed_total - 成功した外部コーディネーター・アクティビティの総数モニター・エレメント
APP_ACT_ABORTED_TOTAL	BIGINT	app_act_aborted_total - 失敗した外部コーディネーター・アクティビティの総数モニター・エレメント
APP_ACT_REJECTED_TOTAL	BIGINT	app_act_rejected_total - リジェクトされた外部コーディネーター・アクティビティの総数モニター・エレメント
TOTAL_CONNECT_REQUEST_TIME	BIGINT	total_connect_request_time - 接続またはユーザー切り替えの要求の合計時間モニター・エレメント
TOTAL_CONNECT_REQUEST_PROC_TIME	BIGINT	total_connect_request_proc_time - 接続要求またはユーザー切り替え要求の合計処理時間モニター・エレメント
TOTAL_CONNECT_REQUESTS	BIGINT	total_connect_requests - 接続またはユーザー切り替えの要求の数モニター・エレメント
TOTAL_CONNECT_AUTHENTICATION_TIME	BIGINT	total_connect_authentication_time - 接続またはユーザー切り替えの認証要求の合計時間モニター・エレメント
TOTAL_CONNECT_AUTHENTICATION_PROC_TIME	BIGINT	total_connect_authentication_proc_time - 接続認証処理の合計時間モニター・エレメント
TOTAL_CONNECT_AUTHENTICATIONS	BIGINT	total_connect_authentications - 接続またはユーザー切り替えの認証の実行回数モニター・エレメント
POOL_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ・モニター・エレメント
POOL_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ・モニター・エレメント
POOL_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ・モニター・エレメント
COMM_EXIT_WAIT_TIME	BIGINT	comm_exit_wait_time - 通信バッファ出口待機時間モニター・エレメント
COMM_EXIT_WAITS	BIGINT	comm_exit_waits - 通信バッファ出口待機回数モニター・エレメント
IDA_SEND_WAIT_TIME	BIGINT	インデータベース分析処理にデータを送信するために待機した合計時間。
IDA_SENDS_TOTAL	BIGINT	インデータベース分析処理にデータが送信された合計回数。



表 162. MON\_GET\_SERVICE\_SUBCLASS について戻される情報 (続き)

列名	データ・タイプ	説明
IDA_SEND_VOLUME	BIGINT	インデータベース分析処理にデータベース・サーバーから送信されたデータの合計量。値はバイト数で報告されます。
IDA_RECV_WAIT_TIME	BIGINT	インデータベース分析処理からデータを受信するために待機した合計時間。
IDA_RECVS_TOTAL	BIGINT	インデータベース分析処理からデータを受け取った合計回数。
IDA_RECV_VOLUME	BIGINT	インデータベース分析処理からデータベース・サーバーが受け取ったデータの合計量。値はバイト数で報告されます。

## MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 表関数 - 詳細サービス・サブクラスのメトリックの取得

MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 表関数は、1 つ以上のサービス・サブクラスの詳細メトリックを戻します。

### 構文

```

▶▶—MON_GET_SERVICE_SUBCLASS_DETAILS—(—service_superclass_name—, —————▶
▶—service_subclass_name—, —member—)—————▶▶
    
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *service\_superclass\_name*

この関数を呼び出すときに現在接続されているデータベースで有効なサービス・スーパークラス名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのスーパークラスについてメトリックが取得されます。

#### *service\_subclass\_name*

この関数を呼び出すときに現在接続されているデータベースで有効なサービス・サブクラス名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのサブクラスについてメトリックが取得されます。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権

- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

サービス・スーパークラスごとに使用された合計 CPU 時間、および処理された要求の合計数を CPU 使用時間順に配列し、リレーショナル形式で表示します (XMLTABLE を使用)。

```
SELECT varchar(scmetrics.service_superclass_name,30) as service_superclass,
       sum(detmetrics.total_cpu_time) as total_cpu,
       sum(detmetrics.app_rqsts_completed_total) as total_rqsts
FROM TABLE(MON_GET_SERVICE_SUBCLASS_DETAILS('','",-2)) AS SCMETRICS,
XMLTABLE (XMLNAMESPACES( DEFAULT 'http://www.ibm.com/xmlns/prod/db2/mon'),
          '$detmetric/db2_service_subclass'
          PASSING XMLPARSE(DOCUMENT SCMETRICS.DETAILS)
          as "detmetric"
COLUMNS "TOTAL_CPU_TIME" INTEGER PATH 'system_metrics/total_cpu_time',
          "APP_RQSTS_COMPLETED_TOTAL" INTEGER PATH
          'system_metrics/app_rqsts_completed_total')
AS DETMETRICS
GROUP BY service_superclass_name
ORDER BY total_cpu desc
```

以下はこの照会の出力例です。

SERVICE_SUPERCLASS	TOTAL_CPU	TOTAL_RQSTS
SYSDEFAULTUSERCLASS	2428188	26
SYSDEFAULTMAINTENANCECLASS	0	0
SYSDEFAULTSYSTEMCLASS	0	0

3 record(s) selected.

## 使用上の注意

MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 表関数で戻されるメトリックは、指示されたサービス・サブクラスの下で実行された要求についてのすべてのメトリックの累計を表します。この関数は、以下の点で MON\_GET\_SERVICE\_SUBCLASS 表関数と似ています。

- MON\_GET\_SERVICE\_SUBCLASS 表関数は、列ベース形式の最も一般的に使用されるメトリックを戻し、メトリックを取得する際に最もパフォーマンス効率が高い方法です。
- MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 表関数は、使用可能なすべてのメトリック一式を XML 文書形式で戻すため、出力フォーマットを最大限柔軟性のあるものにします。XML ベースの出力は、XML パーサーで直接解析でき、XMLTABLE 関数でリレーショナル形式に変換することもできます (例を参照)。

メトリックは、作業単位境界でサービス・クラスにロールアップされ、要求の実行中には定期的にロールアップされます。したがって、この表関数で報告される値は、直前の ROLLUP 時のシステムの現行状態を反映しています。メトリックの値は確実に増加します。ある時間間隔に対する指定されたメトリックの値を判別する

には、MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 表関数を使用してその間隔の始めと終わりのメトリックを照会し、差異を計算します。

要求メトリックは、サービス・スーパークラスに対する COLLECT REQUEST METRICS 節、およびデータベース・レベルの mon\_req\_metrics データベース構成パラメーターを介して制御されます。親サービス・スーパークラスで要求メトリックを使用可能にしているサービス・サブクラスのエージェントが要求を処理する場合にのみ、またはデータベース全体で要求メトリック・コレクションが有効な場合にのみ、その要求に対しメトリックが収集されます。デフォルトでは、要求メトリックはデータベース・レベルで使用可能です。要求メトリックがデータベース・レベルでもサービス・スーパークラスに対しても使用不可になっている場合、そのサービス・スーパークラスにマッピングされた接続ごとに報告されるメトリックは増加を停止します (または、要求メトリックがデータベースのアクティブ化時に無効であった場合には 0 のままになります)。

MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 表関数は、サービス・サブクラスごとおよびメンバーごとに 1 行のデータを戻します。(メンバー上の) サービス・クラス全体または (1 つ以上のサービス・クラスの) メンバー全体の集約は実行されません。しかし、集約は例に示されるように SQL 照会を使用して実行できます (例を参照)。この入力パラメーターの影響として、ANDing されます。したがって、競合する入力パラメーター (例えば、スーパークラス名 SUPA と SUPA のサブクラスではないサブクラス名 SUBB など) を指定する場合、行は戻されません。

スーパークラス SYSDEFAULTSYSTEMCLASS の下のデフォルト・サブクラス SYSDEFAULTSUBCLASS を記述する、この関数によって戻されるデータでは、すべてのメトリックの値が 0 になります。

**ヒント:** 要求は、複数のサービス・サブクラスで実行される場合があります。例えば、REMAP ACTIVITY アクションによってワークロード・マネージャー (WLM) しきい値を使用して要求を 1 つのサービス・サブクラスから別のサービス・サブクラスへマッピングする場合、この状況が生じる可能性があります。メトリックに使用された時間は、要求が実行されるサービス・サブクラスごとに更新されますが、要求カウンターは、要求が完了したサービス・サブクラスに対して増分します。したがって、単一のサブクラスの要求時間の平均を分析する必要はありません。アクティビティーをマッピングできるすべてのサブクラスは相互に関連付けて分析する必要があります。例えば、サービス・サブクラス A からサービス・クラス B にアクティビティーをマッピングできるしきい値があり、要求の平均を計算する場合、サービス・サブサービス A および B についてのカウンターおよびメトリックを集約し、その集約を使用して平均を計算する必要があります。

DETAILS 列に戻される XML 文書のスキーマは、ファイル sqllib/misc/DB2MonRoutines.xsd で入手できます。詳細は、ファイル sqllib/misc/DB2MonCommon.xsd 内にあります。

## 戻される情報

表 163. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS について戻される情報

列名	データ・タイプ	説明
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - サービス・スーパークラス名

表 163. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS について戻される情報 (続き)

列名	データ・タイプ	説明
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - サービス・サブクラス名
SERVICE_CLASS_ID	INTEGER	service_class_id - サービス・クラス ID
MEMBER	SMALLINT	member - データベース・メンバー
DETAILS	BLOB(1M)	サービス・クラスについての詳細メトリックを含む XML 文書。エレメントの説明については、本書の表 164を参照してください。

以下の例は、DETAILS 列で戻される XML 文書の構造を示しています。

```
<db2_service_subclass xmlns="http://www.ibm.com/xmlns/prod/db2/mon" release="90700000">
  <service_superclass_name>SYSDEFAULTSYSTEMCLASS</service_superclass_name>
  <service_subclass_name>SYSDEFAULTSUBCLASS</service_subclass_name>
  <service_subclass_id>11</service_subclass_id>
  <member>0</member>
  <system_metrics release="90700000">
    <act_aborted_total>5</act_aborted_total>
    ...
    <wlm_queue_assignments_total>3</wlm_queue_assignments_total>
  </system_metrics>
</db2_service_subclass>
```

完全スキーマについては、sqllib/misc/DB2MonRoutines.xsd を参照してください。

表 164. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS について戻される詳細メトリック

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
act_aborted_total	xs:nonNegativeInteger	act_aborted_total - 打ち切られたアクティビティーの合計数
act_completed_total	xs:nonNegativeInteger	act_completed_total - 完了したアクティビティーの合計数
act_rejected_total	xs:nonNegativeInteger	act_rejected_total - リジェクトされたアクティビティーの合計数
act_rqsts_total	xs:nonNegativeInteger	act_rqsts_total - アクティビティー要求の合計
agent_wait_time	xs:nonNegativeInteger	agent_wait_time - エージェント待機時間
agent_waits_total	xs:nonNegativeInteger	agent_waits_total - エージェント待機の合計
app_act_aborted_total	xs:nonNegativeInteger	app_act_aborted_total - 失敗した外部コーディネーター・アクティビティーの総数モニター・エレメント
app_act_completed_total	xs:nonNegativeInteger	app_act_completed_total - 成功した外部コーディネーター・アクティビティーの総数モニター・エレメント
app_act_rejected_total	xs:nonNegativeInteger	app_act_rejected_total - リジェクトされた外部コーディネーター・アクティビティーの総数モニター・エレメント
app_rqsts_completed_total	xs:nonNegativeInteger	app_rqsts_completed_total - 完了したアプリケーション要求の合計数
audit_events_total	xs:nonNegativeInteger	audit_events_total - 監査イベント合計数
audit_file_write_wait_time	xs:nonNegativeInteger	audit_file_write_wait_time - 監査ファイル書き込み待機時間
audit_file_writes_total	xs:nonNegativeInteger	audit_file_writes_total - 書き込まれた監査ファイルの合計数
audit_subsystem_wait_time	xs:nonNegativeInteger	audit_subsystem_wait_time - 監査サブシステム待機時間
audit_subsystem_waits_total	xs:nonNegativeInteger	audit_subsystem_waits_total - 監査サブシステム待機の合計
cat_cache_inserts	xs:nonNegativeInteger	cat_cache_inserts - カタログ・キャッシュ挿入数
cat_cache_lookups	xs:nonNegativeInteger	cat_cache_lookups - カタログ・キャッシュ参照数

表 164. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
client_idle_wait_time	xs:nonNegativeInteger	client_idle_wait_time - クライアントのアイドル待機時間
comm_exit_wait_time	xs:nonNegativeInteger	comm_exit_wait_time - 通信バッファ出口待機時間モニター・エレメント
comm_exit_waits	xs:nonNegativeInteger	comm_exit_waits - 通信バッファ出口待機回数モニター・エレメント
deadlocks	xs:nonNegativeInteger	deadlocks - デッドロック検出数
diaglog_write_wait_time	xs:nonNegativeInteger	diaglog_write_wait_time - 診断ログ・ファイルの書き込み待機時間
diaglog_writes_total	xs:nonNegativeInteger	diaglog_writes_total - 診断ログ・ファイル書き込みの合計
direct_read_reqs	xs:nonNegativeInteger	direct_read_reqs - 直接読み取り要求
direct_read_time	xs:nonNegativeInteger	direct_read_time - 直接読み取り時間
direct_reads	xs:nonNegativeInteger	direct_reads - データベースからの直接読み取り
direct_write_reqs	xs:nonNegativeInteger	direct_write_reqs - 直接書き込み要求
direct_write_time	xs:nonNegativeInteger	direct_write_time - 直接書き込み時間
direct_writes	xs:nonNegativeInteger	direct_writes - データベースへの直接書き込み
disabled_peds	xs:long	disabled_peds - 無効化された partial early distinct
evmon_wait_time	xs:nonNegativeInteger	evmon_wait_time - イベント・モニターの待機時間
evmon_waits_total	xs:nonNegativeInteger	evmon_waits_total - イベント・モニターの待機の合計回数
fcm_message_rcv_volume	xs:nonNegativeInteger	fcm_message_rcv_volume - FCM メッセージ受信ボリューム
fcm_message_rcv_wait_time	xs:nonNegativeInteger	fcm_message_rcv_wait_time - FCM メッセージの受信待機時間
fcm_message_rcvs_total	xs:nonNegativeInteger	fcm_message_rcvs_total - FCM メッセージ合計受信数
fcm_message_send_volume	xs:nonNegativeInteger	fcm_message_send_volume - FCM メッセージ送信ボリューム
fcm_message_send_wait_time	xs:nonNegativeInteger	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
fcm_message_sends_total	xs:nonNegativeInteger	fcm_message_sends_total - FCM メッセージ合計送信数
fcm_rcv_volume	xs:nonNegativeInteger	fcm_rcv_volume - FCM 受信ボリューム
fcm_rcv_wait_time	xs:nonNegativeInteger	fcm_rcv_wait_time - FCM 受信待機時間
fcm_rcvs_total	xs:nonNegativeInteger	fcm_rcvs_total - FCM 合計受信数
fcm_send_volume	xs:nonNegativeInteger	fcm_send_volume - FCM 送信ボリューム
fcm_send_wait_time	xs:nonNegativeInteger	fcm_send_wait_time - FCM 送信待ち時間
fcm_sends_total	xs:nonNegativeInteger	fcm_sends_total - FCM 合計送信数
fcm_tq_rcv_volume	xs:nonNegativeInteger	fcm_tq_rcv_volume - FCM 表キューの受信ボリューム
fcm_tq_rcv_wait_time	xs:nonNegativeInteger	fcm_tq_rcv_wait_time - FCM 表キューの受信待ち時間
fcm_tq_rcvs_total	xs:nonNegativeInteger	fcm_tq_rcvs_total - FCM 表キューの合計受信数
fcm_tq_send_volume	xs:nonNegativeInteger	fcm_tq_send_volume - FCM 表キューの送信ボリューム
fcm_tq_send_wait_time	xs:nonNegativeInteger	fcm_tq_send_wait_time - FCM 表キューの送信待ち時間
fcm_tq_sends_total	xs:nonNegativeInteger	fcm_tq_sends_total - FCM 表キューの合計送信数

表 164. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
ida_recv_wait_time	xs:nonNegativeInteger	インデータベース分析処理からデータを受信するために待機した合計時間。
ida_recv_volume	xs:nonNegativeInteger	インデータベース分析処理からデータベース・サーバーが受け取ったデータの合計量。値はバイト数で報告されます。
ida_recvs_total	xs:nonNegativeInteger	インデータベース分析処理からデータを受け取った合計回数。
ida_send_wait_time	xs:nonNegativeInteger	インデータベース分析処理にデータを送信するために待機した合計時間。
ida_send_volume	xs:nonNegativeInteger	インデータベース分析処理にデータベース・サーバーから送信されたデータの合計量。値はバイト数で報告されます。
ida_sends_total	xs:nonNegativeInteger	インデータベース分析処理にデータが送信された合計回数。
int_commits	xs:nonNegativeInteger	int_commits - 内部コミット数
int_rollback	xs:nonNegativeInteger	int_rollback - 内部ロールバック数
ipc_recv_volume	xs:nonNegativeInteger	ipc_recv_volume - プロセス間通信受信ボリューム
ipc_recv_wait_time	xs:nonNegativeInteger	ipc_recv_wait_time - プロセス間通信受信待ち時間
ipc_recvs_total	xs:nonNegativeInteger	ipc_recvs_total - プロセス間通信合計受信数
ipc_send_volume	xs:nonNegativeInteger	ipc_send_volume - プロセス間通信送信ボリューム
ipc_send_wait_time	xs:nonNegativeInteger	ipc_send_wait_time - プロセス間通信送信待ち時間
ipc_sends_total	xs:nonNegativeInteger	ipc_sends_total - プロセス間通信合計送信数
lock_escals	xs:nonNegativeInteger	lock_escals - ロック・エスカレーション数
lock_timeouts	xs:nonNegativeInteger	lock_timeouts - ロック・タイムアウト数
lock_wait_time	xs:nonNegativeInteger	lock_wait_time - ロック待機中の時間
lock_waits	xs:nonNegativeInteger	lock_waits - ロック待機数
log_buffer_wait_time	xs:nonNegativeInteger	log_buffer_wait_time - ログ・バッファ待ち時間
log_disk_wait_time	xs:nonNegativeInteger	log_disk_wait_time - ログ・ディスク待機時間
log_disk_waits_total	xs:nonNegativeInteger	log_disk_waits_total - ログ・ディスク待機の合計
member	xs:nonNegativeInteger	member - データベース・メンバー
num_log_buffer_full	xs:nonNegativeInteger	num_log_buffer_full - フル・ログ・バッファの回数
num_lw_thresh_exceeded	xs:nonNegativeInteger	num_lw_thresh_exceeded - しきい値を超えた回数
pkg_cache_inserts	xs:nonNegativeInteger	pkg_cache_inserts - パッケージ・キャッシュ挿入
pkg_cache_lookups	xs:nonNegativeInteger	pkg_cache_lookups - パッケージ・キャッシュ参照
pool_data_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ・モニター・エレメント
pool_data_l_reads	xs:nonNegativeInteger	pool_data_l_reads - バッファ・プール・データの論理読み取り
pool_data_p_reads	xs:nonNegativeInteger	pool_data_p_reads - バッファ・プール・データの物理読み取り

表 164. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
pool_data_writes	xs:nonNegativeInteger	pool_data_writes - バッファー・プールへのデータの書き込み
pool_failed_async_data_reqs	xs:nonNegativeInteger	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求モニター・エレメント
pool_failed_async_index_reqs	xs:nonNegativeInteger	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求モニター・エレメント
pool_failed_async_other_reqs	xs:nonNegativeInteger	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求モニター・エレメント
pool_failed_async_temp_data_reqs	xs:nonNegativeInteger	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
pool_failed_async_temp_index_reqs	xs:nonNegativeInteger	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
pool_failed_async_temp_xda_reqs	xs:nonNegativeInteger	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求モニター・エレメント
pool_failed_async_xda_reqs	xs:nonNegativeInteger	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求モニター・エレメント
pool_index_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファー・プールで検出されたグループ・バッファー・プール非依存索引ページ・モニター・エレメント
pool_index_l_reads	xs:nonNegativeInteger	pool_index_l_reads - バッファー・プール索引の論理読み取り
pool_index_p_reads	xs:nonNegativeInteger	pool_index_p_reads - バッファー・プール索引の物理読み取り
pool_index_writes	xs:nonNegativeInteger	pool_index_writes - バッファー・プール索引の書き込み
pool_queued_async_data_pages	xs:nonNegativeInteger	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_data_reqs	xs:nonNegativeInteger	pool_queued_async_data_reqs - データ・プリフェッチ要求モニター・エレメント
pool_queued_async_index_pages	xs:nonNegativeInteger	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_index_reqs	xs:nonNegativeInteger	pool_queued_async_index_reqs - 索引プリフェッチ要求モニター・エレメント
pool_queued_async_other_reqs	xs:nonNegativeInteger	pool_queued_async_other_reqs - 非プリフェッチの要求モニター・エレメント
pool_queued_async_temp_data_pages	xs:nonNegativeInteger	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_data_reqs	xs:nonNegativeInteger	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント

表 164. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
pool_queued_async_temp_index_pages	xs:nonNegativeInteger	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_index_reqs	xs:nonNegativeInteger	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
pool_queued_async_temp_xda_pages	xs:nonNegativeInteger	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_xda_reqs	xs:nonNegativeInteger	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求モニター・エレメント
pool_queued_async_xda_reqs	xs:nonNegativeInteger	pool_queued_async_xda_reqs - XDA プリフェッチ要求モニター・エレメント
pool_read_time	xs:nonNegativeInteger	pool_read_time - バッファ・プール物理読み取り時間の合計
pool_temp_data_l_reads	xs:nonNegativeInteger	pool_temp_data_l_reads - バッファ・プールの時データの論理読み取り
pool_temp_data_p_reads	xs:nonNegativeInteger	pool_temp_data_p_reads - バッファ・プールの時データの物理読み取り
pool_temp_index_l_reads	xs:nonNegativeInteger	pool_temp_index_l_reads - バッファ・プールの時索引の論理読み取り
pool_temp_index_p_reads	xs:nonNegativeInteger	pool_temp_index_p_reads - バッファ・プールの時索引の物理読み取り
pool_temp_xda_l_reads	xs:nonNegativeInteger	pool_temp_xda_l_reads - バッファ・プールの時 XDA データの論理読み取り
pool_temp_xda_p_reads	xs:nonNegativeInteger	pool_temp_xda_p_reads - バッファ・プールの時 XDA データの物理読み取り
pool_write_time	xs:nonNegativeInteger	pool_write_time - バッファ・プール物理書き込み時間の合計
pool_xda_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ・モニター・エレメント
pool_xda_gbp_invalid_pages	xs:nonNegativeInteger	pool_xda_gbp_invalid_pages - グループ・バッファ・プールの無効な XDA データ・ページ
pool_xda_gbp_l_reads	xs:nonNegativeInteger	pool_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの論理読み取りの要求
pool_xda_gbp_p_reads	xs:nonNegativeInteger	pool_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの物理読み取りの要求
pool_xda_l_reads	xs:nonNegativeInteger	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
pool_xda_lbp_pages_found	xs:nonNegativeInteger	pool_xda_lbp_pages_found - 検出されたローカル・バッファ・プールの XDA データ・ページ
pool_xda_p_reads	xs:nonNegativeInteger	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り



表 164. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
pool_xda_writes	xs:nonNegativeInteger	pool_xda_writes - バッファ・プール XDA データの書き込み
post_shrthreshold_sorts	xs:nonNegativeInteger	post_shrthreshold_sorts - ポスト共有しきい値ソート
post_threshold_peas	xs:long	post_threshold_peas - partial early aggregation しきい値
post_threshold_peds	xs:long	post_threshold_peds - partial early distinct しきい値
post_threshold_sorts	xs:nonNegativeInteger	post_shrthreshold_sorts - ポスト共有しきい値ソート
prefetch_wait_time	xs:nonNegativeInteger	prefetch_wait_time - プリフェッチ待ち時間
prefetch_waits	xs:nonNegativeInteger	prefetch_waits - プリフェッチャーの待機カウント・モニター・エレメント
rows_modified	xs:nonNegativeInteger	rows_modified - 変更された行数
rows_read	xs:nonNegativeInteger	rows_read - 読み取り行数
rows_returned	xs:nonNegativeInteger	rows_returned - 戻り行数
rqsts_completed_total	xs:nonNegativeInteger	rqsts_completed_total - 完了した要求の合計数
service_class_id	xs:nonNegativeInteger	service_class_id - サービス・クラス ID
service_subclass_name	xs:string(128)	service_subclass_name - サービス・サブクラス名
service_superclass_name	xs:string(128)	service_superclass_name - サービス・スーパークラス名
sort_overflows	xs:nonNegativeInteger	sort_overflows - ソート・オーバーフロー
tcpip_rcv_volume	xs:nonNegativeInteger	tcpip_rcv_volume - TCP/IP 受信ボリューム
tcpip_rcv_wait_time	xs:nonNegativeInteger	tcpip_rcv_wait_time - TCP/IP 受信待ち時間
tcpip_rcvs_total	xs:nonNegativeInteger	tcpip_rcvs_total - TCP/IP 合計受信数
tcpip_send_volume	xs:nonNegativeInteger	tcpip_send_volume - TCP/IP 送信ボリューム
tcpip_send_wait_time	xs:nonNegativeInteger	tcpip_send_wait_time - TCP/IP 送信待ち時間
tcpip_sends_total	xs:nonNegativeInteger	tcpip_sends_total - TCP/IP 合計送信数
thresh_violations	xs:nonNegativeInteger	thresh_violations - しきい値違反の回数
total_act_time	xs:nonNegativeInteger	total_act_time - 合計アクティビティー時間
total_act_wait_time	xs:nonNegativeInteger	total_act_wait_time - 合計アクティビティー待機時間
total_app_commits	xs:nonNegativeInteger	total_app_commits - アプリケーション・コミットの合計回数
total_app_rollback	xs:nonNegativeInteger	total_app_rollback - アプリケーション・ロールバックの合計回数
total_app_rqst_time	xs:nonNegativeInteger	total_app_rqst_time - アプリケーション要求合計時間
total_app_section_executions	xs:nonNegativeInteger	total_app_section_executions - セクション実行の合計回数
total_commit_proc_time	xs:nonNegativeInteger	total_commit_proc_time - コミット処理時間の合計
total_commit_time	xs:nonNegativeInteger	total_commit_time - コミット時間の合計
total_compilations	xs:nonNegativeInteger	total_compilations - コンパイルの合計回数
total_compile_proc_time	xs:nonNegativeInteger	total_compile_proc_time - コンパイル処理時間の合計
total_compile_time	xs:nonNegativeInteger	total_compile_time - コンパイル時間の合計
total_connect_authentication_proc_time	xs:nonNegativeInteger	total_connect_authentication_proc_time - 接続認証の合計処理時間
total_connect_authentication_time	xs:nonNegativeInteger	total_connect_authentication_time - 接続またはユーザー切り替えの認証要求の合計時間

表 164. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
total_connect_authentications	xs:nonNegativeInteger	total_connect_authentications - 接続またはユーザー切り替えの認証の実行回数
total_connect_request_proc_time	xs:nonNegativeInteger	total_connect_request_proc_time - 接続要求またはユーザー切り替え要求の合計処理時間
total_connect_request_time	xs:nonNegativeInteger	total_connect_request_time - 接続要求またはユーザー切り替え要求の合計時間
total_connect_requests	xs:nonNegativeInteger	total_connect_requests - 接続要求またはユーザー切り替え要求
total_cpu_time	xs:nonNegativeInteger	total_cpu_time - 合計 CPU 時間
total_disp_run_queue_time	xs:long	total_disp_run_queue_time - ディスバッチャー実行キュー時間の合計
total_extended_latch_wait_time	xs:nonNegativeInteger	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
total_extended_latch_waits	xs:nonNegativeInteger	total_extended_latch_waits - 拡張ラッチの合計待機回数
total_implicit_compilations	xs:nonNegativeInteger	total_implicit_compilations - 暗黙的コンパイルの合計回数
total_implicit_compile_proc_time	xs:nonNegativeInteger	total_implicit_compile_proc_time - 暗黙的コンパイルの処理時間の合計
total_implicit_compile_time	xs:nonNegativeInteger	total_implicit_compile_time - 暗黙的コンパイル時間の合計
total_load_proc_time	xs:nonNegativeInteger	total_load_proc_time - ロード処理時間の合計
total_load_time	xs:nonNegativeInteger	total_load_time - ロード時間の合計
total_loads	xs:nonNegativeInteger	total_loads - ロードの合計回数
total_peas	xs:long	total_peas - partial early aggregation の合計回数
total_peds	xs:long	total_peds - partial early distinct の合計回数
total_reorg_proc_time	xs:nonNegativeInteger	total_reorg_proc_time - 再編成の処理時間の合計
total_reorg_time	xs:nonNegativeInteger	total_reorg_time - 再編成時間の合計
total_reorgs	xs:nonNegativeInteger	total_reorgs - 再編成の合計回数
total_rollback_proc_time	xs:nonNegativeInteger	total_rollback_proc_time - ロールバック処理時間の合計
total_rollback_time	xs:nonNegativeInteger	total_rollback_time - ロールバック時間の合計
total_routine_invocations	xs:nonNegativeInteger	total_routine_invocations - ルーチン呼び出しの合計回数
total_routine_time	xs:nonNegativeInteger	total_routine_time - ルーチン時間の合計
total_routine_user_code_proc_time	xs:nonNegativeInteger	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
total_routine_user_code_time	xs:nonNegativeInteger	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
total_rqst_mapped_in	xs:nonNegativeInteger	total_rqst_mapped_in - マップで含められた要求の合計
total_rqst_mapped_out	xs:nonNegativeInteger	total_rqst_mapped_out - マップで除外された要求の合計
total_rqst_time	xs:nonNegativeInteger	total_rqst_time - 合計要求時間
total_runstats	xs:nonNegativeInteger	total_runstats - ランタイム統計の合計回数
total_runstats_proc_time	xs:nonNegativeInteger	total_runstats_proc_time - ランタイム統計の処理時間の合計
total_runstats_time	xs:nonNegativeInteger	total_runstats_time - ランタイム統計時間の合計
total_section_proc_time	xs:nonNegativeInteger	total_section_proc_time - セクション処理時間の合計

表 164. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
total_section_sort_proc_time	xs:nonNegativeInteger	total_section_sort_proc_time - セクション・ソート処理時間合計
total_section_sort_time	xs:nonNegativeInteger	total_section_sort_time - セクション・ソート時間合計
total_section_sorts	xs:nonNegativeInteger	total_section_sorts - セクション・ソート合計
total_section_time	xs:nonNegativeInteger	total_section_time - セクション時間の合計
total_sorts	xs:nonNegativeInteger	total_sorts - ソート合計
total_stats_fabrication_proc_time	xs:nonNegativeInteger	total_stats_fabrication_proc_time - 統計作成の合計処理時間
total_stats_fabrication_time	xs:nonNegativeInteger	total_stats_fabrication_time - 統計作成の合計時間
total_stats_fabrications	xs:nonNegativeInteger	total_stats_fabrications - 統計作成の合計回数
total_sync_runstats	xs:nonNegativeInteger	total_sync_runstats - 同期 RUNSTATS アクティビティの合計
total_sync_runstats_proc_time	xs:nonNegativeInteger	total_sync_runstats_proc_time - 同期 RUNSTATS 処理時間の合計
total_sync_runstats_time	xs:nonNegativeInteger	total_sync_runstats_time - 同期 RUNSTATS 時間の合計
total_wait_time	xs:nonNegativeInteger	total_wait_time - 合計待ち時間
tq_sort_heap_rejections	xs:long	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否
tq_sort_heap_requests	xs:long	tq_sort_heap_requests - 表キューのソート・ヒープの要求
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
wlm_queue_assignments_total	xs:nonNegativeInteger	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
wlm_queue_time_total	xs:nonNegativeInteger	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間

## MON\_GET\_TABLE 表関数 - 表メトリックの取得

MON\_GET\_TABLE 表関数は、1 つ以上の表のモニター・メトリックを戻します。

### 構文

►►—MON\_GET\_TABLE—(—*tabschema*—, —*tabname*—, —*member*—)————►►

スキーマは SYSPROC です。

### 表関数パラメーター

#### *tabschema*

この関数を呼び出すときに現在接続されているデータベースで有効な表スキーマ名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのスキーマのすべての表についてメトリックが取得されます。引数が指定される場合、メトリックは指定したスキーマ内の表についてのみ戻されます。

#### *tabname*

この関数を呼び出すときに現在接続されているデータベースで有効な表名を指定

する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべての表についてメトリックが取得されます。

*member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

**許可**

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

**デフォルトの PUBLIC 特権**

なし

**例**

すべてのデータベース・メンバー全体で集約し、読み取り数の高い順に、データベースのアクティブ化後にアクセスされたすべての表のアクティビティーをリストします。

```
SELECT varchar(tabschema,20) as tabschema,
       varchar(tabname,20) as tabname,
       sum(rows_read) as total_rows_read,
       sum(rows_inserted) as total_rows_inserted,
       sum(rows_updated) as total_rows_updated,
       sum(rows_deleted) as total_rows_deleted
FROM TABLE(MON_GET_TABLE('','-2')) AS t
GROUP BY tabschema, tabname
ORDER BY total_rows_read DESC
```

以下はこの照会の出力例です。

TABSHEMA	TABNAME	TOTAL_ROWS_READ	...
SYSIBM	SYSHISTO	113	...
SYSIBM	SYSWORKL	22	...
SYSIBM	SYSROUTI	13	...
SYSIBM	SYSSERVI	13	...
SYSIBM	SYSTHRES	6	...
SYSIBM	SYSTABLE	3	...
SYSIBM	SYSCONTE	2	...
SYSIBM	SYSDBAUT	2	...
SYSIBM	SYSEVENT	2	...
SYSIBM	SYSPLAN	1	...
SYSIBM	SYSSURRO	1	...
SYSIBM	SYSVERSI	1	...
SYSIBM	SYSXMLST	1	...
SYSIBM	SYSAUDIT	0	...
SYSIBM	SYSROLEA	0	...
SYSIBM	SYSROLES	0	...
SYSIBM	SYSTASKS	0	...

```

SYSIBM          SYSWORKA          0 ...
SYSIBM          SYSXMLPA          0 ...

```

19 record(s) selected.

照会の出力 (続き)。

```

... TOTAL_ROWS_INSERTED TOTAL_ROWS_UPDATED TOTAL_ROWS_DELETED
... -----
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0
...                0                0                0

```

### 使用上の注意

MON\_GET\_TABLE 表関数は、データベース表ごとおよびデータベース・メンバーごとに 1 行のデータを戻します。範囲パーティション表が使用される場合、データベース・メンバーごとに、各表パーティションについて 1 行が戻されます。データベース・メンバー全体からの集約は実行されません。ただし、集約は例に示されるように SQL 照会を使用して実行できます。

メトリックは、データベースのアクティブ化後にアクセスされた表についてのみ戻されます。

この関数によって収集されるメトリックは、mon\_obj\_metrics 構成パラメーターを使用してデータベース・レベルで制御されます。デフォルトでは、メトリック収集は有効になります。

### 戻される情報

表 165. MON\_GET\_TABLE について戻される情報

列名	データ・タイプ	説明
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TABNAME	VARCHAR(128)	table_name - 表名
MEMBER	SMALLINT	member - データベース・メンバー
TAB_TYPE	VARCHAR(14)	table_type - 表タイプ
TAB_FILE_ID	BIGINT	table_file_id - 表ファイル ID
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
INDEX_TBSP_ID	BIGINT	index_tbsp_id - 索引表スペース ID

表 165. MON\_GET\_TABLE について戻される情報 (続き)

列名	データ・タイプ	説明
LONG_TBSP_ID	BIGINT	long_tbsp_id - 長い表スペース ID
TABLE_SCANS	BIGINT	table_scans - 表スキャン
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_INSERTED	BIGINT	rows_inserted - 挿入行数
ROWS_UPDATED	BIGINT	rows_updated - 更新行数
ROWS_DELETED	BIGINT	rows_deleted - 削除行数
OVERFLOW_ACCESSES	BIGINT	overflow_accesses - オーバーフロー・レコードへのアクセス
OVERFLOW_CREATES	BIGINT	overflow_creates - オーバーフローの作成
PAGE_REORGS	BIGINT	page_reorgs - ページ再編成
DATA_OBJECT_L_PAGES	BIGINT	data_object_l_pages - 論理データ・オブジェクト・ページ数
LOB_OBJECT_L_PAGES	BIGINT	lob_object_l_pages - 論理 LOB オブジェクト・ページ数
LONG_OBJECT_L_PAGES	BIGINT	long_object_l_pages - 論理 LONG オブジェクト・ページ数
INDEX_OBJECT_L_PAGES	BIGINT	index_object_l_pages - 論理索引オブジェクト・ページ数
XDA_OBJECT_L_PAGES	BIGINT	xda_object_l_pages - 論理 XDA オブジェクト・ページ数
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号
NO_CHANGE_UPDATES	BIGINT	no_change_updates - 無変更の行更新の数
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間
LOCK_WAIT_TIME_GLOBAL	BIGINT	lock_wait_time_global - グローバル・ロック待機時間
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
LOCK_WAITS_GLOBAL	BIGINT	lock_waits_global - グローバル・ロック待機
LOCK_ESCALS	BIGINT	lock_escals - ロック・エスカレーション数
LOCK_ESCALS_GLOBAL	BIGINT	lock_escals_global - グローバル・ロック・エスカレーション数
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
OBJECT_DATA_L_READS	BIGINT	object_data_l_reads - 表のバッファ・プール・データの論理読み取り
OBJECT_DATA_P_READS	BIGINT	object_data_p_reads - 表のバッファ・プール・データの物理読み取り
OBJECT_DATA_GBP_L_READS	BIGINT	object_data_gbp_l_reads - 表のグループ・バッファ・プール・データの論理読み取り
OBJECT_DATA_GBP_P_READS	BIGINT	object_data_gbp_p_reads - 表のグループ・バッファ・プール・データの物理読み取り
OBJECT_DATA_GBP_INVALID_PAGES	BIGINT	object_data_gbp_invalid_pages - 表に関するグループ・バッファ・プールの無効なデータ・ページ
OBJECT_DATA_LBP_PAGES_FOUND	BIGINT	object_data_lbp_pages_found - 表に関して検出されたローカル・バッファ・プールのデータ・ページ

表 165. MON\_GET\_TABLE について戻される情報 (続き)

列名	データ・タイプ	説明
OBJECT_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	object_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ
OBJECT_XDA_L_READS	BIGINT	object_xda_l_reads - 表のバッファ・プール XDA データの論理読み取り
OBJECT_XDA_P_READS	BIGINT	object_xda_p_reads - 表のバッファ・プール XDA データの物理読み取り
OBJECT_XDA_GBP_L_READS	BIGINT	object_xda_gbp_l_reads - 表のグループ・バッファ・プール XDA データの論理読み取り要求
OBJECT_XDA_GBP_P_READS	BIGINT	object_xda_gbp_p_reads - 表のグループ・バッファ・プール XDA データの物理読み取り要求
OBJECT_XDA_GBP_INVALID_PAGES	BIGINT	object_xda_gbp_invalid_pages - 表に関するグループ・バッファ・プールの無効な XDA データ・ページ
OBJECT_XDA_LBP_PAGES_FOUND	BIGINT	object_xda_lbp_pages_found - 表に関して検出されたローカル・バッファ・プールの XDA データ・ページ
OBJECT_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	object_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ
NUM_PAGE_DICT_BUILT	BIGINT	num_page_dict_built - 作成または再作成されたページ・レベルのコンプレッション・ディクショナリーの数

## MON\_GET\_TABLESPACE 表関数 - 表スペース・メトリックの取得

MON\_GET\_TABLESPACE 表関数は、1 つ以上の表スペースのモニター・メトリックを戻します。

### 構文

▶▶MON\_GET\_TABLESPACE(—*tbsp\_name*—,—*member*—)▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### *tbsp\_name*

この関数を呼び出すときに現在接続されているデータベースで有効な表スペース名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべての表スペースについてメトリックが取得されます。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

表スペース・コンテナからの物理的な読み取りの数で順序付けられた表スペースをリストします。

```
SELECT varchar(tbsp_name, 30) as tbsp_name,
        member,
        tbsp_type,
        pool_data_p_reads
FROM TABLE(MON_GET_TABLESPACE(' ', -2)) AS t
ORDER BY pool_data_p_reads DESC
```

以下はこの照会の出力例です。

TBSP_NAME	MEMBER	TBSP_TYPE	POOL_DATA_P_READS
SYSCATSPACE	0	DMS	79
USERSPACE1	0	DMS	34
TEMPSPACE1	0	SMS	0

3 record(s) selected.

## 使用上の注意

MON\_GET\_TABLESPACE 表関数は、データベース表スペースごとおよびデータベース・メンバーごとに 1 行のデータを戻します。データベース・メンバー全体からの集約は実行されません。しかし、集約は SQL 照会を使用して実行できます。

この関数によって収集されるメトリックは、mon\_obj\_metrics 構成パラメーターを使用してデータベース・レベルで制御されます。デフォルトでは、メトリック収集は有効になります。

## 戻される情報

表 166. MON\_GET\_TABLESPACE について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
MEMBER	SMALLINT	member - データベース・メンバー



表 166. MON\_GET\_TABLESPACE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_TYPE	VARCHAR(10)	tablespace_type - 表スペース・タイプ。このインターフェースは、sqlutil.h の定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• DMS</li> <li>• SMS</li> </ul>
TBSP_CONTENT_TYPE	VARCHAR(10)	tablespace_content_type - 表スペースのコンテンツ・タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• ANY</li> <li>• LARGE</li> <li>• SYSTEMP</li> <li>• USRTEMP</li> </ul>
TBSP_PAGE_SIZE	BIGINT	tablespace_page_size - 表スペースのページ・サイズ
TBSP_EXTENT_SIZE	BIGINT	tablespace_extent_size - 表スペースのエクステント・サイズ
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 表スペースのプリフェッチ・サイズ
TBSP_CUR_POOL_ID	BIGINT	tablespace_cur_pool_id - 現在使用中のバッファーク・プール
TBSP_NEXT_POOL_ID	BIGINT	tablespace_next_pool_id - 次の始動時に使用されるバッファーク・プール
FS_CACHING	SMALLINT	fs_caching - ファイル・システム・キャッシング
TBSP_REBALANCER_MODE	VARCHAR(30)	tablespace_rebalancer_mode - リバランサー・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• NO_REBAL</li> <li>• FWD_REBAL</li> <li>• REV_REBAL</li> <li>• FWD_REBAL_OF_2PASS</li> <li>• REV_REBAL_OF_2PASS</li> </ul>
TBSP_USING_AUTO_STORAGE	SMALLINT	tablespace_using_auto_storage - 自動ストレージが使用可能な表スペース
TBSP_AUTO_RESIZE_ENABLED	SMALLINT	tablespace_auto_resize_enabled - 表スペースの自動サイズ変更可能
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファーク・プール・データの論理読み取り

表 166. MON\_GET\_TABLESPACE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDA データの書き込み
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込み
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - バッファ・プール非同期データ読み取り
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - バッファ・プール非同期読み取り要求
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - バッファ・プール非同期データ書き込み
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - バッファ・プール非同期索引読み取り
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - バッファ・プール非同期索引読み取り要求

表 166. MON\_GET\_TABLESPACE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - バッファ・プール非同期索引書き込み
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - バッファ・プール非同期 XDA データ読み取り
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - バッファ・プール非同期 XDA 読み取り要求
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - バッファ・プール非同期 XDA データ書き込み
VECTORED_IOS	BIGINT	vectored_ios - ベクトル化入出力要求数
PAGES_FROM_VECTORED_IOS	BIGINT	pages_from_vectored_ios - ベクトル化入出力によって読み取られたページ数の合計
BLOCK_IOS	BIGINT	block_ios - ブロック入出力要求数
PAGES_FROM_BLOCK_IOS	BIGINT	pages_from_block_ios - ブロック入出力によって読み取られたページ数の合計
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不能プリフェッチ・ページ
FILES_CLOSED	BIGINT	files_closed - 閉じられたデータベース・ファイル
TBSP_STATE	VARCHAR(256)	tablespace_state - 表スペースの状態
TBSP_USED_PAGES	BIGINT	tablespace_used_pages - 表スペース内の使用されているページ数
TBSP_FREE_PAGES	BIGINT	tablespace_free_pages - 表スペース内のフリー・ページ数
TBSP_USABLE_PAGES	BIGINT	tablespace_usable_pages - 表スペース内の使用可能ページ数
TBSP_TOTAL_PAGES	BIGINT	tablespace_total_pages - 表スペース内の合計ページ数
TBSP_PENDING_FREE_PAGES	BIGINT	tablespace_pending_free_pages - 表スペース内のペンディング・フリー・ページ数
TBSP_PAGE_TOP	BIGINT	tablespace_page_top - 表スペース最高水準点
TBSP_MAX_PAGE_TOP	BIGINT	tblsp_max_page_top - 最大表スペース・ページの最高水準点
RECLAIMABLE_SPACE_ENABLED	SMALLINT	reclaimable_space_enabled - 有効な利用可能スペース標識
AUTO_STORAGE_HYBRID	SMALLINT	auto_storage_hybrid - ハイブリッド自動ストレージ表スペース標識
TBSP_PATHS_DROPPED	SMALLINT	tablespace_paths_dropped - ドロップされるパスを使用する表スペース
POOL_DATA_GBP_L_READS	BIGINT	pool_data_gbp_l_reads - グループ・バッファ・プールのデータの論理読み取りモニター・エレメント
POOL_DATA_GBP_P_READS	BIGINT	pool_data_gbp_p_reads - グループ・バッファ・プールのデータの物理読み取りモニター・エレメント
POOL_DATA_LBP_PAGES_FOUND	BIGINT	pool_data_lbp_pages_found - ローカル・バッファ・プールで検出されたデータ・ページ・モニター・エレメント

表 166. MON\_GET\_TABLESPACE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_DATA_GBP_INVALID_PAGES	BIGINT	pool_data_gbp_invalid_pages - グループ・バッファ ー・プールの無効なデータ・ページ・モニター・エ レメント
POOL_INDEX_GBP_L_READS	BIGINT	pool_index_gbp_l_reads - グループ・バッファ ー・プールの索引の論理読み取りモニター・エレメント
POOL_INDEX_GBP_P_READS	BIGINT	pool_index_gbp_p_reads - グループ・バッファ ー・プールの索引の物理読み取りモニター・エレメント
POOL_INDEX_LBP_PAGES_FOUND	BIGINT	pool_index_lbp_pages_found - 検出されたローカル・ バッファ ー・プールの索引ページ・モニター・エレ メント
POOL_INDEX_GBP_INVALID_PAGES	BIGINT	pool_index_gbp_invalid_pages - グループ・バッファ ー・プールの無効な索引ページ・モニター・エレメ ント
POOL_ASYNC_DATA_GBP_L_READS	BIGINT	pool_async_data_gbp_l_reads - グループ・バッファ ー・プールの非同期データ論理読み取りモニター・ エレメント
POOL_ASYNC_DATA_GBP_P_READS	BIGINT	pool_async_data_gbp_p_reads - グループ・バッファ ー・プールの非同期データ物理読み取りモニター・ エレメント
POOL_ASYNC_DATA_ LBP_PAGES_FOUND	BIGINT	pool_async_data_lbp_pages_found - ローカル・バッ ファ ー・プールで検出された非同期データ・ページ・ モニター・エレメント
POOL_ASYNC_DATA_ GBP_INVALID_PAGES	BIGINT	pool_async_data_gbp_invalid_pages - グループ・バッ ファ ー・プールの無効な非同期データ・ページ・モ ニター・エレメント
POOL_ASYNC_INDEX_ GBP_L_READS	BIGINT	pool_async_index_gbp_l_reads - グループ・バッファ ー・プールの非同期索引論理読み取りモニター・エ レメント
POOL_ASYNC_INDEX_ GBP_P_READS	BIGINT	pool_async_index_gbp_p_reads - グループ・バッファ ー・プールの非同期索引物理読み取りモニター・エ レメント
POOL_ASYNC_INDEX_ LBP_PAGES_FOUND	BIGINT	pool_async_index_lbp_pages_found - ローカル・バッ ファ ー・プールで検出された非同期索引ページ・モ ニター・エレメント
POOL_ASYNC_INDEX_ GBP_INVALID_PAGES	BIGINT	pool_async_index_gbp_invalid_pages - グループ・バッ ファ ー・プールの無効な非同期索引ページ・モニタ ー・エレメント
TABLESPACE_MIN_RECOVERY_TIME	TIMESTAMP	tablespace_min_recovery_time - ロールフォワードの最 小リカバリー時間モニター・エレメント

表 166. MON\_GET\_TABLESPACE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号 モニター・エレメント  パーティション・データベース環境の場合、これは MEMBER 列と同じ値になります。DB2 Enterprise Server Edition および DB2 pureScale では、この値は 0 になります。 <b>注:</b> DBPARTITIONNUM は、(値に基づいて表のデータを細分化することにより作成されたデータ・パーティションを識別するのに使われる) <b>data_partition_id</b> とは異なります。
POOL_XDA_GBP_L_READS	BIGINT	pool_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの論理読み取りの要求
POOL_XDA_GBP_P_READS	BIGINT	pool_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの物理読み取りの要求
POOL_XDA_LBP_PAGES_FOUND	BIGINT	pool_xda_lbp_pages_found - 検出されたローカル・バッファ・プールの XDA データ・ページ
POOL_XDA_GBP_INVALID_PAGES	BIGINT	pool_xda_gbp_invalid_pages - グループ・バッファ・プールの無効な XDA データ・ページ
POOL_ASYNC_XDA_GBP_L_READS	BIGINT	pool_async_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの非同期論理読み取りの要求
POOL_ASYNC_XDA_GBP_P_READS	BIGINT	pool_async_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの非同期物理読み取りの要求
POOL_ASYNC_XDA_LBP_PAGES_FOUND	BIGINT	pool_async_xda_lbp_pages_found - 検出された非同期ローカル・バッファ・プールの XDA データ・ページ
POOL_ASYNC_XDA_GBP_INVALID_PAGES	BIGINT	pool_async_xda_gbp_invalid_pages - 非同期グループ・バッファ・プールの無効な XDA データ・ページ
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time バッファ・プール非同期読み取り時間 : モニター・エレメント
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - バッファ・プール非同期書き込み時間 : モニター・エレメント
TBSP_TRACKMOD_STATE	VARCHAR(32)	tbsp_trackmod_state - 表スペースの TRACKMOD 状態。このインターフェースはテキスト ID を戻します。以下のいずれかとなります。 <ul style="list-style-type: none"><li>• CLEAN</li><li>• DIRTY</li><li>• ININCREMENTAL</li><li>• READFULL</li><li>• READINCREMENTAL</li><li>• UNAVAILABLE</li></ul>

表 166. MON\_GET\_TABLESPACE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
STORAGE_GROUP_NAME	VARCHAR(128)	storage_group_name - ストレージ・グループ名モニター・エレメント
STORAGE_GROUP_ID	INTEGER	storage_group_id - ストレージ・グループ ID モニター・エレメント
TBSP_DATATAG	SMALLINT	tbsp_datatag - 表スペース・データ・タグ・モニター・エレメント
TBSP_LAST_CONSEC_PAGE	BIGINT	tbsp_last_consec_page - 最後の連続したオブジェクト表ページ・モニター・エレメント
POOL_QUEUED_ASYNC_DATA_REQS	BIGINT	pool_queued_async_data_reqs - データ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_INDEX_REQS	BIGINT	pool_queued_async_index_reqs - 索引プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_OTHER_REQS	BIGINT	pool_queued_async_other_reqs - 非プリフェッチの要求モニター・エレメント
POOL_QUEUED_ASYNC_DATA_PAGES	BIGINT	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_INDEX_PAGES	BIGINT	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_XDA_REQS	BIGINT	pool_queued_async_xda_reqs - XDA プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_DATA_PAGES	BIGINT	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_INDEX_PAGES	BIGINT	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_XDA_PAGES	BIGINT	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_DATA_REQS	BIGINT	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_INDEX_REQS	BIGINT	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_XDA_REQS	BIGINT	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求モニター・エレメント

表 166. MON\_GET\_TABLESPACE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_FAILED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_OTHER_REQS	BIGINT	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求モニター・エレメント
SKIPPED_PREFETCH_DATA_P_READS	BIGINT	skipped_prefetch_data_p_reads - スキップされたプリフェッチ (データ物理読み取り) モニター・エレメント
SKIPPED_PREFETCH_INDEX_P_READS	BIGINT	skipped_prefetch_index_p_reads - スキップされたプリフェッチ (索引物理読み取り) モニター・エレメント
SKIPPED_PREFETCH_XDA_P_READS	BIGINT	skipped_prefetch_xda_p_reads - スキップされたプリフェッチ (XDA 物理読み取り) モニター・エレメント
SKIPPED_PREFETCH_TEMP_DATA_P_READS	BIGINT	skipped_prefetch_temp_data_p_reads - スキップされたプリフェッチ (一時データ物理読み取り) モニター・エレメント
SKIPPED_PREFETCH_TEMP_INDEX_P_READS	BIGINT	skipped_prefetch_temp_index_p_reads - スキップされたプリフェッチ (一時索引物理読み取り) モニター・エレメント
SKIPPED_PREFETCH_TEMP_XDA_P_READS	BIGINT	skipped_prefetch_temp_xda_p_reads - スキップされたプリフェッチ (一時 XDA データ物理読み取り) モニター・エレメント
SKIPPED_PREFETCH_UOW_DATA_P_READS	BIGINT	skipped_prefetch_uow_data_p_reads - スキップされたプリフェッチ (作業単位データ物理読み取り) モニター・エレメント
SKIPPED_PREFETCH_UOW_INDEX_P_READS	BIGINT	skipped_prefetch_uow_index_p_reads - スキップされたプリフェッチ (作業単位索引物理読み取り) モニター・エレメント
SKIPPED_PREFETCH_UOW_XDA_P_READS	BIGINT	skipped_prefetch_uow_xda_p_reads - スキップされたプリフェッチ (作業単位 XDA データ物理読み取り) モニター・エレメント
SKIPPED_PREFETCH_UOW_TEMP_DATA_P_READS	BIGINT	skipped_prefetch_uow_temp_data_p_reads - スキップされたプリフェッチ (作業単位一時データ物理読み取り) モニター・エレメント
SKIPPED_PREFETCH_UOW_TEMP_INDEX_P_READS	BIGINT	skipped_prefetch_uow_temp_index_p_reads - スキップされたプリフェッチ (作業単位一時索引物理読み取り) モニター・エレメント
SKIPPED_PREFETCH_UOW_TEMP_XDA_P_READS	BIGINT	skipped_prefetch_uow_temp_xda_p_reads - スキップされたプリフェッチ (作業単位一時 XDA データ物理読み取り) モニター・エレメント

表 166. MON\_GET\_TABLESPACE について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間
PREFETCH_WAITS	BIGINT	prefetch_waits - プリフェッチャーの待機カウント・モニター・エレメント
POOL_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ・モニター・エレメント
POOL_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ・モニター・エレメント
POOL_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ・モニター・エレメント
POOL_ASYNC_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_async_data_gbp_indep_pages_found_in_lbp - 非同期 EDU によってローカル・バッファ・プールで検出されたグループ・バッファ・プール非従属データ・ページのモニター・エレメント
POOL_ASYNC_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_async_index_gbp_indep_pages_found_in_lbp - 非同期 EDU によってローカル・バッファ・プールで検出されたグループ・バッファ・プール非従属索引ページのモニター・エレメント
POOL_ASYNC_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_async_xda_gbp_indep_pages_found_in_lbp - 非同期 EDU によってローカル・バッファ・プールで検出されたグループ・バッファ・プール非従属 XML ストレージ・オブジェクト (XDA) ページのモニター・エレメント

## MON\_GET\_TABLE\_USAGE\_LIST 表関数 - 表の使用量リストからの情報を戻す

MON\_GET\_TABLE\_USAGE\_LIST 表関数は、表に関して定義された使用量リストからの情報を戻します。

### 構文

▶▶—MON\_GET\_TABLE\_USAGE\_LIST—(—usagelistschema—,—usagelistname—,—member—)—▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

*usagelistschema*

この関数を呼び出すときに現在接続されているデータベース内の有効なスキーマ名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ス



トリングである場合、データベース内のすべてのスキーマの使用量リストが取得されます。引数を指定した場合は、指定したスキーマの使用量リストのみが戻されます。

#### *usagelistname*

この関数を呼び出すときに現在接続されているデータベース内の表に対して定義された使用量リストを指定する、タイプ VARCHAR(128) の入力引数。

*usagelistname* が NULL または空ストリングである場合、*usagelistschema* で識別されるスキーマにある表に対して定義された、存在するすべての使用量リストが取得されます。これを指定した場合は、*usagelistschema* で識別されるスキーマの、指定した使用量リストのみが戻されます。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

メンバー 3 から使用量リスト USL\_MON\_PAYROLL を取得します。

```
SELECT * FROM TABLE(  
  MON_GET_TABLE_USAGE_LIST(NULL, 'USL_MON_PAYROLL', 3))
```

USAGELISTSHEMA	USAGELISTNAME	TABSCHEMA	TABNAME
ISAYYID	USL_MON_PAYROLL	ISAYYID	T1
ISAYYID	USL_MON_PAYROLL	ISAYYID	T1
MEMBER	...	LAST_UPDATED	...
3	...	2011-07-06-10.20.22.727803	...
3	...	2011-07-06-10.20.58.202161	...

2 record(s) selected.

### 使用上の注意

この関数によって戻される各行は、リストに追加されて以降、特定の時間間隔 (モニター・インターバル ID) 中に固有のセクション (DML ステートメントのみ、実

行可能 ID) が特定のオブジェクトを参照した合計回数 (num\_references) を表します。この行に関して収集された統計は、その時間間隔中のこれらの実行にわたる総計値を表します。

平均を計算するときには、num\_references 列ではなく num\_ref\_with\_metrics 列を使用してください。報告されるメトリックにセクション実行が寄与したかどうかにかかわらず、num\_references 列ではセクションのすべての実行がカウントされるためです。

この関数によって収集されるメトリックは、mon\_obj\_metrics 構成パラメーターを使用してデータベース・レベルで制御されます。デフォルトでは、メトリック収集は有効になります。

## 戻される情報

表 167. MON\_GET\_TABLE\_USAGE\_LIST で戻される情報

列名	データ・タイプ	説明
USAGELISTSHEMA	VARCHAR (128)	usage_list_schema - 使用量リスト・スキーマ
USAGELISTNAME	VARCHAR (128)	usage_list_name - 使用量リスト名
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TABNAME	VARCHAR(128)	table_name - 表名
MEMBER	SMALLINT	member - データベース・メンバー
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	executable_id - 実行可能 ID
MON_INTERVAL_ID	BIGINT	mon_interval_id - モニター・インターバル ID
LAST_UPDATED	TIMESTAMP	last_updated - 最終更新タイム・スタンプ
NUM_REFERENCES	BIGINT	num_references - 参照の数
NUM_REF_WITH_METRICS	BIGINT	num_ref_with_metrics - メトリックに関係した参照の数
ROWS_INSERTED	BIGINT	rows_inserted - 挿入行数
ROWS_DELETED	BIGINT	rows_deleted - 削除行数
ROWS_UPDATED	BIGINT	rows_updated - 更新行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
OVERFLOW_CREATES	BIGINT	overflow_creates - オーバーフローの作成
OVERFLOW_ACCESSES	BIGINT	overflow_accesses - オーバーフロー・レコードへのアクセス
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間
LOCK_WAIT_TIME_GLOBAL	BIGINT	lock_wait_time_global - グローバル・ロック待機時間
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
LOCK_WAITS_GLOBAL	BIGINT	lock_waits_global - グローバル・ロック待機
LOCK_ESCALS	BIGINT	lock_escalations - ロック・エスカレーション数

表 167. MON\_GET\_TABLE\_USAGE\_LIST で戻される情報 (続き)

列名	データ・タイプ	説明
LOCK_ESCALS_GLOBAL	BIGINT	lock_escals_global - グローバル・ロック・エスケーション数
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
OBJECT_DATA_L_READS	BIGINT	object_data_l_reads - 表のバッファ・プール・データの論理読み取り
OBJECT_DATA_P_READS	BIGINT	object_data_p_reads - 表のバッファ・プール・データの物理読み取り
OBJECT_DATA_GBP_L_READS	BIGINT	object_data_gbp_l_reads - 表の GBP データの論理読み取り
OBJECT_DATA_GBP_P_READS	BIGINT	object_data_gbp_p_reads - 表の GBP データの物理読み取り
OBJECT_DATA_GBP_INVALID_PAGES	BIGINT	object_data_gbp_invalid_pages - 表に関する GBP の無効なデータ・ページ
OBJECT_DATA_LBP_PAGES_FOUND	BIGINT	object_data_lbp_pages_found - 表に関して検出された LBP のデータ・ページ
OBJECT_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	object_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ
OBJECT_XDA_L_READS	BIGINT	object_xda_l_reads - 表のバッファ・プール XDA データの論理読み取り
OBJECT_XDA_P_READS	BIGINT	object_xda_p_reads - 表のバッファ・プール XDA データの物理読み取り
OBJECT_XDA_GBP_L_READS	BIGINT	object_xda_gbp_l_reads - 表の GBP XDA データの論理読み取り要求
OBJECT_XDA_GBP_P_READS	BIGINT	object_xda_gbp_p_reads - 表の GBP XDA データの物理読み取り要求
OBJECT_XDA_GBP_INVALID_PAGES	BIGINT	object_xda_gbp_invalid_pages - 表に関する GBP の無効な XDA データ・ページ
OBJECT_XDA_LBP_PAGES_FOUND	BIGINT	object_xda_lbp_pages_found - 表に関して検出された LBP の XDA データ・ページ
OBJECT_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	object_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ

## MON\_GET\_TRANSACTION\_LOG 表関数 - ログ情報の取得

MON\_GET\_TRANSACTION\_LOG 表関数は、現在接続されているデータベースのトランザクション・ロギング・サブシステムに関する情報を戻します。

## 構文

▶▶—MON\_GET\_TRANSACTION\_LOG—(—member—)————▶▶

スキーマは SYSPROC です。

## 表関数パラメーター

### member

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

```
Select MEMBER, CUR_COMMIT_DISK_LOG_READS, CURRENT_ACTIVE_LOG,  
APPLID_HOLDING_OLDEST_TX from table(mon_get_transaction_log(-1)) as t  
order by member asc
```

```
MEMBER  CUR_COMMIT_DISK_LOG_READS  CURRENT_ACTIVE_LOG  APPLID_HOLDING_OLDEST_TX  
-----  
0          9999              1              7
```

## 戻される情報

表 168. MON\_GET\_TRANSACTION\_LOG で戻される情報： DB2 pureScale およびパーティション・データベース環境のどちらも、メンバーごとに 1 行になります。

列名	データ・タイプ	説明
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
TOTAL_LOG_AVAILABLE	BIGINT	total_log_available - 使用可能なログの合計モニター・エレメント
TOTAL_LOG_USED	BIGINT	total_log_used - 使用されているログ・スペースの合計モニター・エレメント
SEC_LOG_USED_TOP	BIGINT	sec_log_used_top - 使用された最大 2 次ログ・スペース・モニター・エレメント
TOT_LOG_USED_TOP	BIGINT	tot_log_used_top - 使用された最大合計ログ・スペース・モニター・エレメント

表 168. MON\_GET\_TRANSACTION\_LOG で戻される情報 (続き): DB2 pureScale およびパーティション・データベース環境のどちらも、メンバーごとに 1 行になります。

列名	データ・タイプ	説明
SEC_LOGS_ALLOCATED	BIGINT	sec_logs_allocated - 現在割り振られている 2 次ログ・モニター・エレメント
LOG_READS	BIGINT	log_reads - 読み取られたログ・ページの数モニター・エレメント
LOG_READ_TIME	BIGINT	log_read_time - ログ読み取り時間モニター・エレメント
LOG_WRITES	BIGINT	log_writes - 書き込まれたログ・ページの数モニター・エレメント
LOG_WRITE_TIME	BIGINT	log_write_time - ログ書き込み時間モニター・エレメント
NUM_LOG_WRITE_IO	BIGINT	num_log_write_io - ログ書き込み数モニター・エレメント
NUM_LOG_READ_IO	BIGINT	num_log_read_io - ログ読み取り数モニター・エレメント
NUM_LOG_PART_PAGE_IO	BIGINT	num_log_part_page_io - 部分ログ・ページ書き込み数モニター・エレメント
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - エージェントの待機が発生したフル・ログ・バッファの回数モニター・エレメント
NUM_LOG_DATA_FOUND_IN_BUFFER	BIGINT	num_log_data_found_in_buffer - ログ・データがバッファにある回数モニター・エレメント
APPLID_HOLDING_OLDEST_XACT	BIGINT	最も古いトランザクションを保持しているアプリケーション・ハンドル。
LOG_TO_REDO_FOR_RECOVERY	BIGINT	log_to_redo_for_recovery - リカバリーの場合に再実行されるログの量モニター・エレメント
LOG_HELD_BY_DIRTY_PAGES	BIGINT	log_held_by_dirty_pages - ダーティー・ページによって占有されるログ・スペースの量モニター・エレメント
FIRST_ACTIVE_LOG	BIGINT	first_active_log - 先頭アクティブ・ログ・ファイル番号モニター・エレメント
LAST_ACTIVE_LOG	BIGINT	last_active_log - 最終アクティブ・ログ・ファイル番号モニター・エレメント
CURRENT_ACTIVE_LOG	BIGINT	current_active_log - 現行アクティブ・ログ・ファイル番号モニター・エレメント
CURRENT_ARCHIVE_LOG	BIGINT	current_archive_log - 現行アーカイブ・ログ・ファイル番号モニター・エレメント
CUR_COMMIT_DISK_LOG_READS	BIGINT	(ログ・バッファではなく) ディスクからのログ読み取りにより、現在コミット済みバージョンの行が検索された回数。
CUR_COMMIT_TOTAL_LOG_READS	BIGINT	現在コミット済みバージョンの行が、ログ (ログ・バッファおよびディスク) から検索された合計回数。
CUR_COMMIT_LOG_BUFF_LOG_READS	BIGINT	これは CUR_COMMIT_TOTAL_LOG_READS - CUR_COMMIT_DISK_LOG_READS として計算されます。
ARCHIVE_METHOD1_STATUS	SMALLINT	最新のログ・アーカイブの試行結果。使用される値は、1 (成功)、0 (失敗)、あるいは NULL (該当なし) です。
METHOD1_NEXT_LOG_TO_ARCHIVE	BIGINT	次にアーカイブされるログ・ファイル。
METHOD1_FIRST_FAILURE	BIGINT	アーカイブが失敗した最初のログ・ファイル。

表 168. MON\_GET\_TRANSACTION\_LOG で戻される情報 (続き): DB2 pureScale およびパーティション・データベース環境のどちらも、メンバーごとに 1 行になります。

列名	データ・タイプ	説明
ARCHIVE_METHOD2_STATUS	SMALLINT	最新の 2 次ログ・アーカイブ試行の結果。使用される値は、1 (成功)、0 (失敗)、あるいは NULL (該当なし) です。
METHOD2_NEXT_LOG_TO_ARCHIVE	BIGINT	次にアーカイブされる 2 次ログ・ファイル。
METHOD2_FIRST_FAILURE	BIGINT	アーカイブに失敗した最初の 2 次ログ・ファイル。
LOG_CHAIN_ID	BIGINT	ログ・チェーン番号の ID。
CURRENT_LSO	BIGINT	現在のログ・シーケンス・オフセット。
CURRENT_LSN	BIGINT	現在のログ・シーケンス番号。
OLDEST_TX_LSN	BIGINT	最も古いトランザクション実行に関連付けられたログ・シーケンス番号。
NUM_LOGS_AVAIL_FOR_RENAME	INTEGER	ログ・アーカイブを使用している場合、これは再使用の目的でログ・パスに維持されているログ・ファイルの数です。それ以外の場合、この値は 0 です。
NUM_INDOUBT_TRANS	BIGINT	num_indoubt_trans - 未確定トランザクション数モニター・エレメント
LOG_HADR_WAIT_TIME	BIGINT	HADR 処理の待機に費やされた時間。
LOG_HADR_WAITS_TOTAL	BIGINT	トランザクション・ログ・サブシステムが HADR 処理を待機した合計回数。

## MON\_GET\_UNIT\_OF\_WORK 表関数 - 作業単位メトリックの取得

MON\_GET\_UNIT\_OF\_WORK 表関数は、1 つ以上の作業単位のメトリックを戻します。

### 構文

▶▶ MON\_GET\_UNIT\_OF\_WORK (—application\_handle—, —member—) ◀◀

スキーマは SYSPROC です。

### 表関数パラメーター

#### application\_handle

この関数を呼び出すときに現在接続されているデータベースと同じデータベース内の有効なアプリケーション・ハンドルを指定する、タイプ BIGINT のオプション入力引数。引数が NULL である場合、データベース内のすべてのスーパークラスで実行される作業単位についてメトリックが取得されます。

#### member

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER のオプション入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

システムで最長の CPU 時間を消費する作業単位を識別します。

```
SELECT application_handle,  
       uow_id,  
       total_cpu_time,  
       app_rqsts_completed_total,  
       rqsts_completed_total  
FROM TABLE(MON_GET_UNIT_OF_WORK(NULL,-1)) AS t  
ORDER BY total_cpu_time DESC
```

以下はこの照会の出力例です。

APPLICATION_HANDLE	UOW_ID	TOTAL_CPU_TIME	...
-----	-----	-----	-----
	46	5	27959 ...

1 record(s) selected.

照会の出力 (続き)。

...	APP_RQSTS_COMPLETED_TOTAL	RQSTS_COMPLETED_TOTAL
...	-----	-----
...	72	48

## 使用上の注意

MON\_GET\_UNIT\_OF\_WORK 表関数で戻されるメトリックは、作業単位の間サブミットされた要求についてのすべてのメトリックの累計を表します。メトリックは作業単位の間定期的にロールアップされます。したがって、この表関数で報告される値は、直前の ROLLUP 時のシステムの現行状態を反映しています。メトリックの値は確実に増加します。ある時間間隔に対する指定されたメトリックの値を判別するには、この関数を使用してその間隔の始めと終わりのメトリックを照会し、差異を計算します。

要求メトリックは、サービス・スーパークラスに対する COLLECT REQUEST METRICS 節、およびデータベース・レベルの *mon\_req\_metrics* データベース構成パラメーターを介して制御されます。親サービス・スーパークラスで要求メトリックを使用可能にしているサービス・サブクラスのエージェントが要求を処理する場合にのみ、またはデータベース全体で要求メトリック・コレクションが有効な場合にのみ、その要求に対しメトリックが収集されます。デフォルトでは、要求メトリックはデータベース・レベルで使用可能です。要求メトリックがデータベース・レベルでもサービス・スーパークラスに対しても使用不可になっている場合、そのサ

サービス・スーパークラスにマッピングされた作業単位ごとに報告されるメトリックは増加を停止します (または、要求メトリックがデータベースのアクティブ化時に無効であった場合には 0 のままになります)。

MON\_GET\_UNIT\_OF\_WORK 表関数は、作業単位ごとおよびメンバーごとに 1 行のデータを戻します。(メンバー上の) 作業単位全体または (1 つ以上のサービス・クラスの) メンバー全体の集約は実行されません。しかし、集約は SQL 照会を使用して実行できます。この入力パラメーターの影響として、ANDing されます。

## 戻される情報

表 169. MON\_GET\_UNIT\_OF\_WORK について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - サービス・スーパークラス名
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - サービス・サブクラス名
SERVICE_CLASS_ID	INTEGER	service_class_id - サービス・クラス ID
MEMBER	SMALLINT	member - データベース・メンバー
COORD_MEMBER	SMALLINT	coord_member - コーディネーター・メンバー
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル
APPLICATION_ID	VARCHAR(128)	appl_id - アプリケーション ID
WORKLOAD_NAME	VARCHAR(128)	workload_name - ワークロード名
WORKLOAD_OCCURRENCE_ID	INTEGER	workload_occurrence_id - ワークロード・オカレンス ID。ワークロード・オカレンスがコーディネーター・メンバーおよびワークロード名と結合されていない場合は、この ID はワークロード・オカレンスを一意的に識別しません。
UOW_ID	INTEGER	uow_id - 作業単位 ID
WORKLOAD_OCCURRENCE_STATE	VARCHAR(32)	workload_occurrence_state - ワークロード・オカレンスの状態
CLIENT_WRKSTNNAME	VARCHAR(255)	CURRENT CLIENT_WRKSTNNAME 特殊レジスター
CLIENT_ACCTNG	VARCHAR(255)	CURRENT CLIENT_ACCTNG 特殊レジスター
CLIENT_USERID	VARCHAR(255)	CURRENT CLIENT_USERID 特殊レジスター
CLIENT_APPLNAME	VARCHAR(255)	CURRENT CLIENT_APPLNAME 特殊レジスター
UOW_START_TIME	TIMESTAMP	uow_start_time - 作業単位開始タイム・スタンプ
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - セッション許可 ID
ACT_ABORTED_TOTAL	BIGINT	act_aborted_total - 打ち切られたアクティビティーの合計数
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 完了したアクティビティーの合計数
ACT_REJECTED_TOTAL	BIGINT	act_rejected_total - リジェクトされたアクティビティーの合計数
AGENT_WAIT_TIME	BIGINT	agent_wait_time - エージェント待ち時間
AGENT_WAITS_TOTAL	BIGINT	agent_waits_total - エージェント待機の合計



表 169. MON\_GET\_UNIT\_OF\_WORK について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込み
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDA データ書き込み
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
CLIENT_IDLE_WAIT_TIME	BIGINT	client_idle_wait_time - クライアントのアイドル待機時間
DEADLOCKS	BIGINT	deadlocks - デッドロック検出数
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求

表 169. MON\_GET\_UNIT\_OF\_WORK について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
FCM_RECV_VOLUME	BIGINT	fcm_recv_volume - FCM 受信ボリューム
FCM_RECVS_TOTAL	BIGINT	fcm_recvs_total - FCM 合計受信数
FCM_SEND_VOLUME	BIGINT	fcm_send_volume - FCM 送信ボリューム
FCM_SENDS_TOTAL	BIGINT	fcm_sends_total - FCM 合計送信数
FCM_RECV_WAIT_TIME	BIGINT	fcm_recv_wait_time - FCM 受信待ち時間
FCM_SEND_WAIT_TIME	BIGINT	fcm_send_wait_time - FCM 送信待ち時間
IPC_RECV_VOLUME	BIGINT	ipc_recv_volume - プロセス間通信受信ボリューム
IPC_RECV_WAIT_TIME	BIGINT	ipc_recv_wait_time - プロセス間通信受信待ち時間
IPC_RECVS_TOTAL	BIGINT	ipc_recvs_total - プロセス間通信合計受信数
IPC_SEND_VOLUME	BIGINT	ipc_send_volume - プロセス間通信送信ボリューム
IPC_SEND_WAIT_TIME	BIGINT	ipc_send_wait_time - プロセス間通信送信待ち時間
IPC_SENDS_TOTAL	BIGINT	ipc_sends_total - プロセス間通信合計送信数
LOCK_ESCALS	BIGINT	lock_escalations - ロック・エスカレーション数
LOCK_TIMEOUTS	BIGINT	lock_timeouts - ロック・タイムアウト数
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
LOG_BUFFER_WAIT_TIME	BIGINT	log_buffer_wait_time - ログ・バッファ待ち時間
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - フル・ログ・バッファの回数
LOG_DISK_WAIT_TIME	BIGINT	log_disk_wait_time - ログ・ディスク待ち時間
LOG_DISK_WAITS_TOTAL	BIGINT	log_disk_waits_total - ログ・ディスク待機の合計
NUM_LOCKS_HELD	BIGINT	locks_held - ロック保持数
RQSTS_COMPLETED_TOTAL	BIGINT	rqsts_completed_total - 完了した要求の合計数
ROWS_MODIFIED	BIGINT	rows_modified - 変更された行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_RETURNED	BIGINT	rows_returned - 戻り行数
TCPIP_RECV_VOLUME	BIGINT	tcPIP_recv_volume - TCP/IP 受信ボリューム
TCPIP_SEND_VOLUME	BIGINT	tcPIP_send_volume - TCP/IP 送信ボリューム
TCPIP_RECV_WAIT_TIME	BIGINT	tcPIP_recv_wait_time - TCP/IP 受信待ち時間
TCPIP_RECVS_TOTAL	BIGINT	tcPIP_recvs_total - TCP/IP 合計受信数
TCPIP_SEND_WAIT_TIME	BIGINT	tcPIP_send_wait_time - TCP/IP 送信待ち時間
TCPIP_SENDS_TOTAL	BIGINT	tcPIP_sends_total - TCP/IP 合計送信数
TOTAL_APP_RQST_TIME	BIGINT	total_app_rqst_time - アプリケーション要求合計時間
TOTAL_RQST_TIME	BIGINT	total_rqst_time - 合計要求時間
WLM_QUEUE_TIME_TOTAL	BIGINT	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間
WLM_QUEUE_ASSIGNMENTS_TOTAL	BIGINT	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間

表 169. MON\_GET\_UNIT\_OF\_WORK について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_WAIT_TIME	BIGINT	total_wait_time - 合計待ち時間
APP_RQSTS_COMPLETED_TOTAL	BIGINT	app_rqsts_completed_total - 完了したアプリケーション要求の合計数
TOTAL_SECTION_SORT_TIME	BIGINT	total_section_sort_time - セクション・ソート時間合計
TOTAL_SECTION_SORT_PROC_TIME	BIGINT	total_section_sort_proc_time - セクション・ソート処理時間合計
TOTAL_SECTION_SORTS	BIGINT	total_section_sorts - セクション・ソート合計
TOTAL_SORTS	BIGINT	total_sorts - ソート合計
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - ポストしきい値ソート
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - ポスト共有しきい値ソート
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
TOTAL_COMPILE_TIME	BIGINT	total_compile_time - コンパイル時間の合計
TOTAL_COMPILE_PROC_TIME	BIGINT	total_compile_proc_time - コンパイル処理時間の合計
TOTAL_COMPILATIONS	BIGINT	total_compilations - コンパイルの合計回数
TOTAL_IMPLICIT_COMPILE_TIME	BIGINT	total_implicit_compile_time - 暗黙的コンパイル時間の合計
TOTAL_IMPLICIT_COMPILE_PROC_TIME	BIGINT	total_implicit_compile_proc_time - 暗黙的コンパイルの処理時間の合計
TOTAL_IMPLICIT_COMPILATIONS	BIGINT	total_implicit_compilations - 暗黙的コンパイルの合計回数
TOTAL_SECTION_TIME	BIGINT	total_section_time - セクション時間の合計
TOTAL_SECTION_PROC_TIME	BIGINT	total_section_proc_time - セクション処理時間の合計
TOTAL_APP_SECTION_EXECUTIONS	BIGINT	total_app_section_executions - セクション実行の合計回数
TOTAL_ACT_TIME	BIGINT	total_act_time - 合計アクティビティー時間
TOTAL_ACT_WAIT_TIME	BIGINT	total_act_wait_time - 合計アクティビティー待機時間
ACT_RQSTS_TOTAL	BIGINT	act_rqsts_total - アクティビティー要求の合計
TOTAL_ROUTINE_TIME	BIGINT	total_routine_time - ルーチン時間の合計
TOTAL_ROUTINE_INVOCATIONS	BIGINT	total_routine_invocations - ルーチン呼び出しの合計回数
TOTAL_COMMIT_TIME	BIGINT	total_commit_time - コミット時間の合計
TOTAL_COMMIT_PROC_TIME	BIGINT	total_commit_proc_time - コミット処理時間の合計
TOTAL_APP_COMMITS	BIGINT	total_app_commits - アプリケーション・コミットの合計回数
INT_COMMITS	BIGINT	int_commits - 内部コミット数
TOTAL_ROLLBACK_TIME	BIGINT	total_rollback_time - ロールバック時間の合計
TOTAL_ROLLBACK_PROC_TIME	BIGINT	total_rollback_proc_time - ロールバック処理時間の合計

表 169. MON\_GET\_UNIT\_OF\_WORK について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_APP_ROLLBACKS	BIGINT	total_app_rollback - アプリケーション・ロールバックの合計回数
INT_ROLLBACKS	BIGINT	int_rollback - 内部ロールバック数
TOTAL_RUNSTATS_TIME	BIGINT	total_runstats_time - ランタイム統計時間の合計
TOTAL_RUNSTATS_PROC_TIME	BIGINT	total_runstats_proc_time - ランタイム統計の処理時間の合計
TOTAL_RUNSTATS	BIGINT	total_runstats - ランタイム統計の合計回数
TOTAL_REORG_TIME	BIGINT	total_reorg_time - 再編成時間の合計
TOTAL_REORG_PROC_TIME	BIGINT	total_reorg_proc_time - 再編成の処理時間の合計
TOTAL_REORGS	BIGINT	total_reorgs - 再編成の合計回数
TOTAL_LOAD_TIME	BIGINT	total_load_time - ロード時間の合計
TOTAL_LOAD_PROC_TIME	BIGINT	total_load_proc_time - ロード処理時間の合計
TOTAL_LOADS	BIGINT	total_loads - ロードの合計回数
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - カタログ・キャッシュ挿入数
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - カタログ・キャッシュ参照数
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - パッケージ・キャッシュ挿入
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - パッケージ・キャッシュ参照
THRESH_VIOLATIONS	BIGINT	thresh_violations - しきい値違反の回数
NUM_LW_THRESH_EXCEEDED	BIGINT	num_lw_thresh_exceeded - しきい値を超えた回数
UOW_LOG_SPACE_USED	BIGINT	uow_log_space_used - 作業単位ログ・スペース
LOCK_WAITS_GLOBAL	BIGINT	lock_waits_global - グローバル・ロック待機
LOCK_WAIT_TIME_GLOBAL	BIGINT	lock_wait_time_global - グローバル・ロック待機時間
LOCK_TIMEOUTS_GLOBAL	BIGINT	lock_timeouts_global - グローバル・ロック・タイムアウト
LOCK_ESCALS_MAXLOCKS	BIGINT	lock_escals_maxlocks - maxlocks ロック・エスカレーション数
LOCK_ESCALS_LOCKLIST	BIGINT	lock_escals_locklist - locklist ロック・エスカレーション数
LOCK_ESCALS_GLOBAL	BIGINT	lock_escals_global - グローバル・ロック・エスカレーション数
RECLAIM_WAIT_TIME	BIGINT	reclaim_wait_time - 再利用の待機時間
SPACEMAPPAGE_RECLAIM_WAIT_TIME	BIGINT	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間
CF_WAITS	BIGINT	cf_waits - クラスター・キャッシング・ファシリテーター待機回数
CF_WAIT_TIME	BIGINT	cf_wait_time - クラスター・キャッシング・ファシリテーターの待機時間
POOL_DATA_GBP_L_READS	BIGINT	pool_data_gbp_l_reads - グループ・バッファ・プールのデータの論理読み取り
POOL_DATA_GBP_P_READS	BIGINT	pool_data_gbp_p_reads - グループ・バッファ・プールのデータの物理読み取り

表 169. MON\_GET\_UNIT\_OF\_WORK について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_DATA_LBP_PAGES_FOUND	BIGINT	pool_data_lbp_pages_found - ローカル・バッファ ー・プールで検出されたデータ・ページ
POOL_DATA_GBP_INVALID_PAGES	BIGINT	pool_data_gbp_invalid_pages - グループ・バッファ ー・プールの無効なデータ・ページ
POOL_INDEX_GBP_L_READS	BIGINT	pool_index_gbp_l_reads - グループ・バッファ ー・プールの索引の論理読み取り
POOL_INDEX_GBP_P_READS	BIGINT	pool_index_gbp_p_reads - グループ・バッファ ー・プールの索引の物理読み取り
POOL_INDEX_LBP_PAGES_FOUND	BIGINT	pool_index_lbp_pages_found - 検出されたローカル ・バッファ・プールの索引ページ
POOL_INDEX_GBP_INVALID_PAGES	BIGINT	pool_index_gbp_invalid_pages - グループ・バッファ ー・プールの無効な索引ページ
POOL_XDA_GBP_L_READS	BIGINT	pool_xda_gbp_l_reads - グループ・バッファ ー・プールの XDA データの論理読み取りの要求
POOL_XDA_GBP_P_READS	BIGINT	pool_xda_gbp_p_reads - グループ・バッファ ー・プールの XDA データの物理読み取りの要求
POOL_XDA_LBP_PAGES_FOUND	BIGINT	pool_xda_lbp_pages_found - 検出されたローカル ・バッファ・プールの XDA データ・ページ
POOL_XDA_GBP_INVALID_PAGES	BIGINT	pool_xda_gbp_invalid_pages - グループ・バッファ ー・プールの無効な XDA データ・ページ
AUDIT_EVENTS_TOTAL	BIGINT	audit_events_total - 監査イベント合計数
AUDIT_FILE_WRITES_TOTAL	BIGINT	audit_file_writes_total - 書き込まれた監査ファイル の合計数
AUDIT_FILE_WRITE_WAIT_TIME	BIGINT	audit_file_write_wait_time - 監査ファイル書き込み待 機時間
AUDIT_SUBSYSTEM_WAITS_TOTAL	BIGINT	audit_subsystem_waits_total - 監査サブシステム待機 の合計
AUDIT_SUBSYSTEM_WAIT_TIME	BIGINT	audit_subsystem_wait_time - 監査サブシステム待機 時間
CLIENT_HOSTNAME	VARCHAR(255)	client_hostname - クライアント・ホスト名のモニタ ー・エレメント
CLIENT_PORT_NUMBER	INTEGER	client_port_number - クライアント・ポート番号のモ ニター・エレメント
DIAGLOG_WRITES_TOTAL	BIGINT	diaglog_writes_total - 診断ログ・ファイル書き込み の合計
DIAGLOG_WRITE_WAIT_TIME	BIGINT	diaglog_write_wait_time - 診断ログ・ファイルの書 き込み待機時間
FCM_MESSAGE_RECVS_TOTAL	BIGINT	fcm_message_recvs_total - FCM メッセージ合計受 信数
FCM_MESSAGE_RECV_VOLUME	BIGINT	fcm_message_recv_volume - FCM メッセージ受信ボ リューム
FCM_MESSAGE_RECV_WAIT_TIME	BIGINT	fcm_message_recv_wait_time - FCM メッセージの受 信待機時間

表 169. MON\_GET\_UNIT\_OF\_WORK について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
FCM_MESSAGE_SENDS_TOTAL	BIGINT	fcm_message_sends_total - FCM メッセージ合計送信数
FCM_MESSAGE_SEND_VOLUME	BIGINT	fcm_message_send_volume - FCM メッセージ送信ボリューム
FCM_MESSAGE_SEND_WAIT_TIME	BIGINT	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
FCM_TQ_RECVS_TOTAL	BIGINT	fcm_tq_recvs_total - FCM 表キューの合計受信数
FCM_TQ_RECV_VOLUME	BIGINT	fcm_tq_recv_volume - FCM 表キューの受信ボリューム
FCM_TQ_RECV_WAIT_TIME	BIGINT	fcm_tq_recv_wait_time - FCM 表キューの受信待ち時間
FCM_TQ_SENDS_TOTAL	BIGINT	fcm_tq_sends_total - FCM 表キューの合計送信数
FCM_TQ_SEND_VOLUME	BIGINT	fcm_tq_send_volume - FCM 表キューの送信ボリューム
FCM_TQ_SEND_WAIT_TIME	BIGINT	fcm_tq_send_wait_time - FCM 表キューの送信待ち時間
LAST_EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	last_executable_id - 最後の実行可能 ID
LAST_REQUEST_TYPE	VARCHAR(32)	last_request_type - 最後の要求タイプ
TOTAL_ROUTINE_USER_CODE_PROC_TIME	BIGINT	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
TOTAL_ROUTINE_USER_CODE_TIME	BIGINT	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
TQ_TOT_SEND_SPILLS	BIGINT	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
EVMON_WAIT_TIME	BIGINT	evmon_wait_time - イベント・モニターの待機時間
EVMON_WAITS_TOTAL	BIGINT	evmon_waits_total - イベント・モニターの待機の合計回数
TOTAL_EXTENDED_LATCH_WAIT_TIME	BIGINT	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
TOTAL_EXTENDED_LATCH_WAITS	BIGINT	total_extended_latch_waits - 拡張ラッチの合計待機回数
INTRA_PARALLEL_STATE	VARCHAR(3)	intra_parallel_state - パーティション内並列処理の現在の状態
TOTAL_STATS_FABRICATION_TIME	BIGINT	total_stats_fabrication_time - 統計作成の合計時間
TOTAL_STATS_FABRICATION_PROC_TIME	BIGINT	total_stats_fabrication_proc_time - 統計作成の合計処理時間
TOTAL_STATS_FABRICATIONS	BIGINT	total_stats_fabrications - 統計作成の合計回数
TOTAL_SYNC_RUNSTATS_TIME	BIGINT	total_sync_runstats_time - 同期 RUNSTATS 時間の合計
TOTAL_SYNC_RUNSTATS_PROC_TIME	BIGINT	total_sync_runstats_proc_time - 同期 RUNSTATS 処理時間の合計

表 169. MON\_GET\_UNIT\_OF\_WORK について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_SYNC_RUNSTATS	BIGINT	total_sync_runstats - 同期 RUNSTATS アクティビティの合計
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - ディスパッチャー実行キュー時間の合計
DISABLED_PEDS	BIGINT	disabled_peds - 無効化された partial early distinct
POOL_QUEUED_ASYNC_DATA_REQS	BIGINT	pool_queued_async_data_reqs - データ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_INDEX_REQS	BIGINT	pool_queued_async_index_reqs - 索引プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_XDA_REQS	BIGINT	pool_queued_async_xda_reqs - XDA プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_OTHER_REQS	BIGINT	pool_queued_async_other_reqs - 非プリフェッチの要求モニター・エレメント
POOL_QUEUED_ASYNC_DATA_PAGES	BIGINT	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_INDEX_PAGES	BIGINT	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_XDA_PAGES	BIGINT	pool_queued_async_xda_pages - XDA ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_DATA_PAGES	BIGINT	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_INDEX_PAGES	BIGINT	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_XDA_PAGES	BIGINT	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_DATA_REQS	BIGINT	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_INDEX_REQS	BIGINT	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_XDA_REQS	BIGINT	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求モニター・エレメント

表 169. MON\_GET\_UNIT\_OF\_WORK について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_FAILED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_OTHER_REQS	BIGINT	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求モニター・エレメント
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間
PREFETCH_WAITS	BIGINT	prefetch_waits - プリフェッチャーの待機カウント・モニター・エレメント
POST_THRESHOLD_PEAS	BIGINT	post_threshold_peas - partial early aggregation しきい値
POST_THRESHOLD_PEDS	BIGINT	post_threshold_peds - partial early distinct しきい値
TOTAL_PEAS	BIGINT	total_peas - partial early aggregation の合計回数
TOTAL_PEDS	BIGINT	total_peds - partial early distinct の合計回数
TQ_SORT_HEAP_REJECTIONS	BIGINT	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否
TQ_SORT_HEAP_REQUESTS	BIGINT	tq_sort_heap_requests - 表キューのソート・ヒープの要求
APP_ACT_ABORTED_TOTAL	BIGINT	app_act_aborted_total - 失敗した外部コーディネーター・アクティビティの総数モニター・エレメント
APP_ACT_COMPLETED_TOTAL	BIGINT	app_act_completed_total - 成功した外部コーディネーター・アクティビティの総数モニター・エレメント
APP_ACT_REJECTED_TOTAL	BIGINT	app_act_rejected_total - リジェクトされた外部コーディネーター・アクティビティの総数モニター・エレメント
TOTAL_CONNECT_REQUEST_TIME	BIGINT	total_connect_request_time - 接続またはユーザー切り替えの要求の合計時間モニター・エレメント
TOTAL_CONNECT_REQUEST_PROC_TIME	BIGINT	total_connect_request_proc_time - 接続要求またはユーザー切り替え要求の合計処理時間モニター・エレメント
TOTAL_CONNECT_REQUESTS	BIGINT	total_connect_requests - 接続またはユーザー切り替えの要求の数モニター・エレメント
TOTAL_CONNECT_AUTHENTICATION_TIME	BIGINT	total_connect_authentication_time - 接続またはユーザー切り替えの認証要求の合計時間モニター・エレメント
TOTAL_CONNECT_AUTHENTICATION_PROC_TIME	BIGINT	total_connect_authentication_proc_time - 接続認証処理の合計時間モニター・エレメント



表 169. MON\_GET\_UNIT\_OF\_WORK について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_CONNECT_AUTHENTICATIONS	BIGINT	total_connect_authentications - 接続またはユーザー切り替えの認証の実行回数モニター・エレメント
POOL_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ・モニター・エレメント
POOL_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ・モニター・エレメント
POOL_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ・モニター・エレメント
COMM_EXIT_WAIT_TIME	BIGINT	comm_exit_wait_time - 通信バッファ出口待機時間モニター・エレメント
COMM_EXIT_WAITS	BIGINT	comm_exit_waits - 通信バッファ出口待機回数モニター・エレメント
IDA_SEND_WAIT_TIME	BIGINT	インデータベース分析処理にデータを送信するために待機した合計時間。
IDA_SENDS_TOTAL	BIGINT	インデータベース分析処理にデータが送信された合計回数。
IDA_SEND_VOLUME	BIGINT	インデータベース分析処理にデータベース・サーバーから送信されたデータの合計量。値はバイト数で報告されます。
IDA_RECV_WAIT_TIME	BIGINT	インデータベース分析処理からデータを受信するために待機した合計時間。
IDA_RECVS_TOTAL	BIGINT	インデータベース分析処理からデータを受け取った合計回数。
IDA_RECV_VOLUME	BIGINT	インデータベース分析処理からデータベース・サーバーが受け取ったデータの合計量。値はバイト数で報告されます。

## MON\_GET\_UNIT\_OF\_WORK\_DETAILS 表関数 - 作業単位の詳細メトリックの取得

MON\_GET\_UNIT\_OF\_WORK\_DETAILS 表関数は、1 つ以上の作業単位の詳細メトリックを戻します。

### 構文

►►MON\_GET\_UNIT\_OF\_WORK\_DETAILS(—application\_handle—, —member—)◄◄

スキーマは SYSPROC です。

## 表関数パラメーター

### *application\_handle*

この関数を呼び出すときに現在接続されているデータベースと同じデータベース内の有効なアプリケーション・ハンドルを指定する、タイプ **BIGINT** の入力引数。引数が **NULL** である場合、データベース内のすべてのスーパークラスで実行される作業単位についてメトリックが取得されます。

### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ **INTEGER** の入力引数。現行のデータベース・メンバーには **-1**、すべてのデータベース・メンバーには **-2** を指定します。**NULL** 値を指定すると、**-1** が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する **EXECUTE** 特権
- **DATAACCESS** 権限
- **DBADM** 権限
- **SQLADM** 権限

## デフォルトの **PUBLIC** 特権

なし

## 例

システムで最長の CPU 時間を消費する作業単位を識別します。

```
SELECT detmetrics.application_handle,
       detmetrics.uow_id,
       detmetrics.total_cpu_time,
       detmetrics.app_rqsts_completed_total,
       detmetrics.rqsts_completed_total
FROM TABLE(MON_GET_UNIT_OF_WORK_DETAILS(NULL,-2)) AS UOWMETRICS,
XMLTABLE (
  XMLNAMESPACES( DEFAULT 'http://www.ibm.com/xmlns/prod/db2/mon'),
  '$detmetric/db2_unit_of_work' PASSING
  XMLPARSE(DOCUMENT UOWMETRICS.DETAILS)
  as "detmetric"
) AS DETMETRICS
ORDER BY total_cpu_time DESC
```

以下はこの照会の出力例です。

```
APPLICATION_HANDLE  UOW_ID  TOTAL_CPU_TIME  ...
-----
                    46          5          27959  ...
```

1 record(s) selected.

照会の出力 (続き)。

```
... APP_RQSTS_COMPLETED_TOTAL RQSTS_COMPLETED_TOTAL
... -----
...                               72                               48
```

## 使用上の注意

MON\_GET\_UNIT\_OF\_WORK 関数で戻されるメトリックは、作業単位の間サブミットされた要求についてのすべてのメトリックの累計を表します。この関数は、以下の点で MON\_GET\_UNIT\_OF\_WORK 表関数と似ています。

- MON\_GET\_UNIT\_OF\_WORK 表関数は、最も一般的に使用されるメトリックを列ベースの形式で戻すため、メトリックの取得において最も効率的な方法です。
- MON\_GET\_UNIT\_OF\_WORK\_DETAILS 表関数は、使用可能なすべてのメトリック一式を XML 文書形式で戻すため、出力フォーマットを最大限柔軟性のあるものにします。XML ベースの出力は、XML パーサーで直接解析でき、XMLTABLE 関数でリレーショナル形式に変換することもできます (例を参照)。

メトリックは作業単位の間定期的にロールアップされます。したがって、この表関数で報告される値は、直前の ROLLUP 時のシステムの現行状態を反映しています。メトリックの値は確実に増加します。ある時間間隔に対する指定されたメトリックの値を判別するには、MON\_GET\_UNIT\_OF\_WORK\_DETAILS 表関数を使用してその間隔の始めと終わりのメトリックを照会し、差異を計算します。

要求メトリックは、サービス・スーパークラスに対する COLLECT REQUEST METRICS 節、およびデータベース・レベルの **mon\_req\_metrics** データベース構成パラメーターを介して制御されます。親サービス・スーパークラスで要求メトリックを使用可能にしているサービス・サブクラスのエージェントが要求を処理する場合にのみ、またはデータベース全体で要求メトリック・コレクションが有効な場合にのみ、その要求に対しメトリックが収集されます。デフォルトでは、要求メトリックはデータベース・レベルで使用可能です。要求メトリックがデータベース・レベルでもサービス・スーパークラスに対しても使用不可になっている場合、そのサービス・スーパークラスにマッピングされた作業単位ごとに報告されるメトリックは増加を停止します (または、要求メトリックがデータベースのアクティブ化時に無効であった場合には 0 のままになります)。

MON\_GET\_UNIT\_OF\_WORK\_DETAILS 表関数は、作業単位ごとおよびメンバーごとに 1 行のデータを戻します。(メンバー上の) 作業単位全体または (1 つ以上のサービス・クラスの) メンバー全体の集約は実行されません。しかし、集約は SQL 照会を使用して実行できます。この入力パラメーターの影響として、ANDing されます。

DETAILS 列に戻される XML 文書のスキーマは、ファイル `sqllib/misc/DB2MonRoutines.xsd` で入手できます。詳細は、ファイル `sqllib/misc/DB2MonCommon.xsd` 内にあります。

## 戻される情報

表 170. MON\_GET\_UNIT\_OF\_WORK\_DETAILS について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - サービス・スーパークラス名
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - サービス・サブクラス名
SERVICE_CLASS_ID	INTEGER	service_class_id - サービス・クラス ID
MEMBER	SMALLINT	member - データベース・メンバー
COORD_MEMBER	SMALLINT	coord_member - コーディネーター・メンバー。指定された作業単位のコーディネーター・パーティションのデータベース・メンバー。
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル
WORKLOAD_NAME	VARCHAR(128)	workload_name - ワークロード名
WORKLOAD_OCCURRENCE_ID	INTEGER	workload_occurrence_id - ワークロード・オカレンス ID。ワークロード・オカレンスがコーディネーター・データベース・パーティション番号およびワークロード名と結合されていない場合は、この ID はワークロード・オカレンスを一意的に識別しません。
UOW_ID	INTEGER	uow_id - 作業単位 ID
DETAILS	BLOB(1M)	作業単位についての詳細メトリックを含む XML 文書。エレメントの説明については、本書の表 171を参照してください。

以下の例は、DETAILS 列で戻される XML 文書の構造を示しています。

```
<db2_unit_of_work xmlns="http://www.ibm.com/xmlns/prod/db2/mon" release="90700000">
  <service_superclass_name>SYSDEFAULTUSERCLASS</service_superclass_name>
  <service_subclass_name>SYSDEFAULTSUBCLASS</service_subclass_name>
  <service_class_id>13</service_class_id>
  <workload_name>SYSDEFAULTUSERWORKLOAD</workload_name>
  <member>0</member>
  <coord_member>0</coord_member>
  <application_handle>21</application_handle>
  <workload_occurrence_id>1</workload_occurrence_id>
  <uow_id>2</uow_id>
  <workload_occurrence_state>UOWEXEC</workload_occurrence_state>
  <system_metrics>
    <act_aborted_total>5</act_aborted_total>
    ...
    <wlm_queue_assignments_total>3</wlm_queue_assignments_total>
  </system_metrics>
</db2_unit_of_work_metrics>
```

完全スキーマについては、sqllib/misc/DB2MonRoutines.xsd を参照してください。

表 171. MON\_GET\_UNIT\_OF\_WORK\_DETAILS について戻される詳細メトリック

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
act_aborted_total	xs:nonNegativeInteger	act_aborted_total - 打ち切られたアクティビティーの合計数
act_completed_total	xs:nonNegativeInteger	act_completed_total - 完了したアクティビティーの合計数

表 171. MON\_GET\_UNIT\_OF\_WORK\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
act_rejected_total	xs:nonNegativeInteger	act_rejected_total - リジェクトされたアクティビティの合計数
act_rqsts_total	xs:nonNegativeInteger	act_rqsts_total - アクティビティ要求の合計
agent_wait_time	xs:nonNegativeInteger	agent_wait_time - エージェント待機時間
agent_waits_total	xs:nonNegativeInteger	agent_waits_total - エージェント待機の合計
app_act_aborted_total	xs:nonNegativeInteger	app_act_aborted_total - 失敗した外部コーディネーター・アクティビティの総数モニター・エレメント
app_act_completed_total	xs:nonNegativeInteger	app_act_completed_total - 成功した外部コーディネーター・アクティビティの総数モニター・エレメント
app_act_rejected_total	xs:nonNegativeInteger	app_act_rejected_total - リジェクトされた外部コーディネーター・アクティビティの総数モニター・エレメント
app_rqsts_completed_total	xs:nonNegativeInteger	app_rqsts_completed_total - 完了したアプリケーション要求の合計数
application_handle	xs:nonNegativeInteger	application_handle - アプリケーション・ハンドル
application_id	xs:string	appl_id - アプリケーション ID
audit_events_total	xs:nonNegativeInteger	audit_events_total - 監査イベント合計数
audit_file_write_wait_time	xs:nonNegativeInteger	audit_file_write_wait_time - 監査ファイル書き込み待機時間
audit_file_writes_total	xs:nonNegativeInteger	audit_file_writes_total - 書き込まれた監査ファイルの合計数
audit_subsystem_wait_time	xs:nonNegativeInteger	audit_subsystem_wait_time - 監査サブシステム待機時間
audit_subsystem_waits_total	xs:nonNegativeInteger	audit_subsystem_waits_total - 監査サブシステム待機の合計
cat_cache_inserts	xs:nonNegativeInteger	cat_cache_inserts - カタログ・キャッシュ挿入数
cat_cache_lookups	xs:nonNegativeInteger	cat_cache_lookups - カタログ・キャッシュ参照数
client_acctng	xs:string	CURRENT_CLIENT_ACCTNG 特殊レジスター
client_applname	xs:string	CURRENT_CLIENT_APPLNAME 特殊レジスター
client_hostname	xs:string	client_hostname - クライアント・ホスト名
client_idle_wait_time	xs:nonNegativeInteger	client_idle_wait_time - クライアントのアイドル待機時間
client_port_number	xs:nonNegativeInteger	client_port_number - クライアント・ポート番号
client_userid	xs:string	CURRENT_CLIENT_USERID 特殊レジスター
client_wrkstnname	xs:string	CURRENT_CLIENT_WRKSTNNAME 特殊レジスター
comm_exit_wait_time	xs:nonNegativeInteger	comm_exit_wait_time - 通信バッファ出口待機時間モニター・エレメント
comm_exit_waits	xs:nonNegativeInteger	comm_exit_waits - 通信バッファ出口待機回数モニター・エレメント
coord_member	xs:nonNegativeInteger	coord_member - コーディネーター・メンバー
deadlocks	xs:nonNegativeInteger	deadlocks - デッドロック検出数

表 171. MON\_GET\_UNIT\_OF\_WORK\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
diaglog_write_wait_time	xs:nonNegativeInteger	diaglog_write_wait_time - 診断ログ・ファイルの書き込み待機時間
diaglog_writes_total	xs:nonNegativeInteger	diaglog_writes_total - 診断ログ・ファイル書き込みの合計
direct_read_reqs	xs:nonNegativeInteger	direct_read_reqs - 直接読み取り要求
direct_read_time	xs:nonNegativeInteger	direct_read_time - 直接読み取り時間
direct_reads	xs:nonNegativeInteger	direct_reads - データベースからの直接読み取り
direct_write_reqs	xs:nonNegativeInteger	direct_write_reqs - 直接書き込み要求
direct_write_time	xs:nonNegativeInteger	direct_write_time - 直接書き込み時間
direct_writes	xs:nonNegativeInteger	direct_writes - データベースへの直接書き込み
disabled_peds	xs:long	disabled_peds - 無効化された partial early distinct
evmon_wait_time	xs:nonNegativeInteger	evmon_wait_time - イベント・モニターの待機時間
evmon_waits_total	xs:nonNegativeInteger	evmon_waits_total - イベント・モニターの待機の合計回数
fcm_message_recv_volume	xs:nonNegativeInteger	fcm_message_recv_volume - FCM メッセージ受信ボリューム
fcm_message_recv_wait_time	xs:nonNegativeInteger	fcm_message_recv_wait_time - FCM メッセージの受信待機時間
fcm_message_recvs_total	xs:nonNegativeInteger	fcm_message_recvs_total - FCM メッセージ合計受信数
fcm_message_send_volume	xs:nonNegativeInteger	fcm_message_send_volume - FCM メッセージ送信ボリューム
fcm_message_send_wait_time	xs:nonNegativeInteger	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
fcm_message_sends_total	xs:nonNegativeInteger	fcm_message_sends_total - FCM メッセージ合計送信数
fcm_recv_volume	xs:nonNegativeInteger	fcm_recv_volume - FCM 受信ボリューム
fcm_recv_wait_time	xs:nonNegativeInteger	fcm_recv_wait_time - FCM 受信待機時間
fcm_recvs_total	xs:nonNegativeInteger	fcm_recvs_total - FCM 合計受信数
fcm_send_volume	xs:nonNegativeInteger	fcm_send_volume - FCM 送信ボリューム
fcm_send_wait_time	xs:nonNegativeInteger	fcm_send_wait_time - FCM 送信待ち時間
fcm_sends_total	xs:nonNegativeInteger	fcm_sends_total - FCM 合計送信数
fcm_tq_recv_volume	xs:nonNegativeInteger	fcm_tq_recv_volume - FCM 表キューの受信ボリューム
fcm_tq_recv_wait_time	xs:nonNegativeInteger	fcm_tq_recv_wait_time - FCM 表キューの受信待ち時間
fcm_tq_recvs_total	xs:nonNegativeInteger	fcm_tq_recvs_total - FCM 表キューの合計受信数
fcm_tq_send_volume	xs:nonNegativeInteger	fcm_tq_send_volume - FCM 表キューの送信ボリューム
fcm_tq_send_wait_time	xs:nonNegativeInteger	fcm_tq_send_wait_time - FCM 表キューの送信待ち時間
fcm_tq_sends_total	xs:nonNegativeInteger	fcm_tq_sends_total - FCM 表キューの合計送信数

表 171. MON\_GET\_UNIT\_OF\_WORK\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
ida_recv_wait_time	xs:nonNegativeInteger	インデータベース分析処理からデータを受信するために待機した合計時間。
ida_recv_volume	xs:nonNegativeInteger	インデータベース分析処理からデータベース・サーバーが受け取ったデータの合計量。値はバイト数で報告されます。
ida_recvs_total	xs:nonNegativeInteger	インデータベース分析処理からデータを受け取った合計回数。
ida_send_wait_time	xs:nonNegativeInteger	インデータベース分析処理にデータを送信するために待機した合計時間。
ida_send_volume	xs:nonNegativeInteger	インデータベース分析処理にデータベース・サーバーから送信されたデータの合計量。値はバイト数で報告されます。
ida_sends_total	xs:nonNegativeInteger	インデータベース分析処理にデータが送信された合計回数。
int_commits	xs:nonNegativeInteger	int_commits - 内部コミット数
int_rollbacks	xs:nonNegativeInteger	int_rollbacks - 内部ロールバック数
intra_parallel_state	xs:string	intra_parallel_state - パーティション内並列処理の現行状態モニター・エレメント
ipc_recv_volume	xs:nonNegativeInteger	ipc_recv_volume - プロセス間通信受信ボリューム
ipc_recv_wait_time	xs:nonNegativeInteger	ipc_recv_wait_time - プロセス間通信受信待ち時間
ipc_recvs_total	xs:nonNegativeInteger	ipc_recvs_total - プロセス間通信合計受信数
ipc_send_volume	xs:nonNegativeInteger	ipc_send_volume - プロセス間通信送信ボリューム
ipc_send_wait_time	xs:nonNegativeInteger	ipc_send_wait_time - プロセス間通信送信待ち時間
ipc_sends_total	xs:nonNegativeInteger	ipc_sends_total - プロセス間通信合計送信数
last_executable_id	xs:hexBinary(32)	last_executable_id - 最後の実行可能 ID
last_request_type	xs:string(32)	last_request_type - 最後の要求タイプ
lock_escals	xs:nonNegativeInteger	lock_escals - ロック・エスカレーション数
lock_timeouts	xs:nonNegativeInteger	lock_timeouts - ロック・タイムアウト数
lock_wait_time	xs:nonNegativeInteger	lock_wait_time - ロック待機中の時間
lock_waits	xs:nonNegativeInteger	lock_waits - ロック待機数
log_buffer_wait_time	xs:nonNegativeInteger	log_buffer_wait_time - ログ・バッファ待ち時間
log_disk_wait_time	xs:nonNegativeInteger	log_disk_wait_time - ログ・ディスク待機時間
log_disk_waits_total	xs:nonNegativeInteger	log_disk_waits_total - ログ・ディスク待機の合計
member	xs:nonNegativeInteger	member - データベース・メンバー
num_locks_held	xs:nonNegativeInteger	locks_held - ロック保持数
num_log_buffer_full	xs:nonNegativeInteger	num_log_buffer_full - フル・ログ・バッファの回数
num_lw_thresh_exceeded	xs:nonNegativeInteger	num_lw_thresh_exceeded - しきい値を超えた回数
pkg_cache_inserts	xs:nonNegativeInteger	pkg_cache_inserts - パッケージ・キャッシュ挿入
pkg_cache_lookups	xs:nonNegativeInteger	pkg_cache_lookups - パッケージ・キャッシュ参照

表 171. MON\_GET\_UNIT\_OF\_WORK\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
pool_data_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ・モニター・エレメント
pool_data_l_reads	xs:nonNegativeInteger	pool_data_l_reads - バッファ・プール・データの論理読み取り
pool_data_p_reads	xs:nonNegativeInteger	pool_data_p_reads - バッファ・プール・データの物理読み取り
pool_data_writes	xs:nonNegativeInteger	pool_data_writes - バッファ・プールへのデータの書き込み
pool_failed_async_data_reqs	xs:nonNegativeInteger	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求モニター・エレメント
pool_failed_async_index_reqs	xs:nonNegativeInteger	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求モニター・エレメント
pool_failed_async_other_reqs	xs:nonNegativeInteger	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求モニター・エレメント
pool_failed_async_temp_data_reqs	xs:nonNegativeInteger	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
pool_failed_async_temp_index_reqs	xs:nonNegativeInteger	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
pool_failed_async_temp_xda_reqs	xs:nonNegativeInteger	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求モニター・エレメント
pool_failed_async_xda_reqs	xs:nonNegativeInteger	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求モニター・エレメント
pool_index_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ・モニター・エレメント
pool_index_l_reads	xs:nonNegativeInteger	pool_index_l_reads - バッファ・プール索引の論理読み取り
pool_index_p_reads	xs:nonNegativeInteger	pool_index_p_reads - バッファ・プール索引の物理読み取り
pool_index_writes	xs:nonNegativeInteger	pool_index_writes - バッファ・プール索引の書き込み
pool_queued_async_data_pages	xs:nonNegativeInteger	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_data_reqs	xs:nonNegativeInteger	pool_queued_async_data_reqs - データ・プリフェッチ要求モニター・エレメント
pool_queued_async_index_pages	xs:nonNegativeInteger	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_index_reqs	xs:nonNegativeInteger	pool_queued_async_index_reqs - 索引プリフェッチ要求モニター・エレメント



表 171. MON\_GET\_UNIT\_OF\_WORK\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
pool_queued_async_other_reqs	xs:nonNegativeInteger	pool_queued_async_other_reqs - 非プリフェッチの要求モニター・エレメント
pool_queued_async_temp_data_pages	xs:nonNegativeInteger	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_data_reqs	xs:nonNegativeInteger	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_index_pages	xs:nonNegativeInteger	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_index_reqs	xs:nonNegativeInteger	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
pool_queued_async_temp_xda_pages	xs:nonNegativeInteger	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_xda_reqs	xs:nonNegativeInteger	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求モニター・エレメント
pool_queued_async_xda_pages	xs:nonNegativeInteger	プリフェッチが正常に要求された、XML ストレージ・オブジェクト (XDA) のデータ・ページの数。
pool_queued_async_xda_reqs	xs:nonNegativeInteger	pool_queued_async_xda_reqs - XDA プリフェッチ要求モニター・エレメント
pool_read_time	xs:nonNegativeInteger	pool_read_time - バッファ・プール物理読み取り時間の合計
pool_temp_data_l_reads	xs:nonNegativeInteger	pool_temp_data_l_reads - バッファ・プールのデータ論理読み取り
pool_temp_data_p_reads	xs:nonNegativeInteger	pool_temp_data_p_reads - バッファ・プールのデータ物理読み取り
pool_temp_index_l_reads	xs:nonNegativeInteger	pool_temp_index_l_reads - バッファ・プールの索引論理読み取り
pool_temp_index_p_reads	xs:nonNegativeInteger	pool_temp_index_p_reads - バッファ・プールの索引物理読み取り
pool_temp_xda_l_reads	xs:nonNegativeInteger	pool_temp_xda_l_reads - バッファ・プールの XDA データ論理読み取り
pool_temp_xda_p_reads	xs:nonNegativeInteger	pool_temp_xda_p_reads - バッファ・プールの XDA データ物理読み取り
pool_write_time	xs:nonNegativeInteger	pool_write_time - バッファ・プール物理書き込み時間の合計
pool_xda_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ・モニター・エレメント

表 171. MON\_GET\_UNIT\_OF\_WORK\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
pool_xda_gbp_invalid_pages	xs:nonNegativeInteger	pool_xda_gbp_invalid_pages - グループ・バッファ・プールの無効な XDA データ・ページ
pool_xda_gbp_l_reads	xs:nonNegativeInteger	pool_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの論理読み取りの要求
pool_xda_gbp_p_reads	xs:nonNegativeInteger	pool_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの物理読み取りの要求
pool_xda_l_reads	xs:nonNegativeInteger	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
pool_xda_lbp_pages_found	xs:nonNegativeInteger	pool_xda_lbp_pages_found - 検出されたローカル・バッファ・プールの XDA データ・ページ
pool_xda_p_reads	xs:nonNegativeInteger	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
pool_xda_writes	xs:nonNegativeInteger	pool_xda_writes - バッファ・プール XDA データの書き込み
post_shrthreshold_sorts	xs:nonNegativeInteger	post_shrthreshold_sorts - ポスト共有しきい値ソート
post_threshold_peas	xs:long	post_threshold_peas - partial early aggregation しきい値
post_threshold_peds	xs:long	post_threshold_peds - partial early distinct しきい値
post_threshold_sorts	xs:nonNegativeInteger	post_threshold_sorts - ポストしきい値ソート
prefetch_wait_time	xs:nonNegativeInteger	prefetch_wait_time - プリフェッチ待ち時間
prefetch_waits	xs:nonNegativeInteger	prefetch_waits - プリフェッチャーの待機カウント・モニター・エレメント
rows_modified	xs:nonNegativeInteger	rows_modified - 変更された行数
rows_read	xs:nonNegativeInteger	rows_read - 読み取り行数
rows_returned	xs:nonNegativeInteger	rows_returned - 戻り行数
rqsts_completed_total	xs:nonNegativeInteger	rqsts_completed_total - 完了した要求の合計数
service_class_id	xs:nonNegativeInteger	service_class_id - サービス・クラス ID
service_subclass_name	xs:string (128)	service_subclass_name - サービス・サブクラス名
service_superclass_name	xs:string (128)	service_superclass_name - サービス・スーパークラス名
session_auth_id	xs:string	session_auth_id - セッション許可 ID
sort_overflows	xs:nonNegativeInteger	sort_overflows - ソート・オーバーフロー
tcpip_recv_volume	xs:nonNegativeInteger	tcpip_recv_volume - TCP/IP 受信ボリューム
tcpip_recv_wait_time	xs:nonNegativeInteger	tcpip_recv_wait_time - TCP/IP 受信待ち時間
tcpip_recvs_total	xs:nonNegativeInteger	tcpip_recvs_total - TCP/IP 合計受信数
tcpip_send_volume	xs:nonNegativeInteger	tcpip_send_volume - TCP/IP 送信ボリューム
tcpip_send_wait_time	xs:nonNegativeInteger	tcpip_send_wait_time - TCP/IP 送信待ち時間
tcpip_sends_total	xs:nonNegativeInteger	tcpip_sends_total - TCP/IP 合計送信数
thresh_violations	xs:nonNegativeInteger	thresh_violations - しきい値違反の回数
total_act_time	xs:nonNegativeInteger	total_act_time - 合計アクティビティー時間
total_act_wait_time	xs:nonNegativeInteger	total_act_wait_time - 合計アクティビティー待機時間
total_app_commits	xs:nonNegativeInteger	total_app_commits - アプリケーション・コミットの合計回数

表 171. MON\_GET\_UNIT\_OF\_WORK\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
total_app_rollbacks	xs:nonNegativeInteger	total_app_rollbacks - アプリケーション・ロールバックの合計回数
total_app_rqst_time	xs:nonNegativeInteger	total_app_rqst_time - アプリケーション要求合計時間
total_app_section_executions	xs:nonNegativeInteger	total_app_section_executions - セクション実行の合計回数
total_commit_proc_time	xs:nonNegativeInteger	total_commit_proc_time - コミット処理時間の合計
total_commit_time	xs:nonNegativeInteger	total_commit_time - コミット時間の合計
total_compilations	xs:nonNegativeInteger	total_compilations - コンパイルの合計回数
total_compile_proc_time	xs:nonNegativeInteger	total_compile_proc_time - コンパイル処理時間の合計
total_compile_time	xs:nonNegativeInteger	total_compile_time - コンパイル時間の合計
total_connect_authentication_proc_time	xs:nonNegativeInteger	total_connect_authentication_proc_time - 接続認証の合計処理時間
total_connect_authentication_time	xs:nonNegativeInteger	total_connect_authentication_time - 接続またはユーザー切り替えの認証要求の合計時間
total_connect_authentications	xs:nonNegativeInteger	total_connect_authentications - 接続またはユーザー切り替えの認証の実行回数
total_connect_request_proc_time	xs:nonNegativeInteger	total_connect_request_proc_time - 接続要求またはユーザー切り替え要求の合計処理時間
total_connect_request_time	xs:nonNegativeInteger	total_connect_request_time - 接続要求またはユーザー切り替え要求の合計時間
total_connect_requests	xs:nonNegativeInteger	total_connect_requests - 接続要求またはユーザー切り替え要求
total_cpu_time	xs:nonNegativeInteger	total_cpu_time - 合計 CPU 時間
total_disp_run_queue_time	xs:long	total_disp_run_queue_time - ディスパッチャー実行キュー時間の合計
total_extended_latch_wait_time	xs:nonNegativeInteger	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
total_extended_latch_waits	xs:nonNegativeInteger	total_extended_latch_waits - 拡張ラッチの合計待機回数
total_implicit_compilations	xs:nonNegativeInteger	total_implicit_compilations - 暗黙的コンパイルの合計回数
total_implicit_compile_proc_time	xs:nonNegativeInteger	total_implicit_compile_proc_time - 暗黙的コンパイルの処理時間の合計
total_implicit_compile_time	xs:nonNegativeInteger	total_implicit_compile_time - 暗黙的コンパイル時間の合計
total_load_proc_time	xs:nonNegativeInteger	total_load_proc_time - ロード処理時間の合計
total_load_time	xs:nonNegativeInteger	total_load_time - ロード時間の合計
total_loads	xs:nonNegativeInteger	total_loads - ロードの合計回数
total_peas	xs:long	total_peas - partial early aggregation の合計回数
total_peds	xs:long	total_peds - partial early distinct の合計回数
total_reorg_proc_time	xs:nonNegativeInteger	total_reorg_proc_time - 再編成の処理時間の合計
total_reorg_time	xs:nonNegativeInteger	total_reorg_time - 再編成時間の合計
total_reorgs	xs:nonNegativeInteger	total_reorgs - 再編成の合計回数

表 171. MON\_GET\_UNIT\_OF\_WORK\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
total_rollback_proc_time	xs:nonNegativeInteger	total_rollback_proc_time - ロールバック処理時間の合計
total_rollback_time	xs:nonNegativeInteger	total_rollback_time - ロールバック時間の合計
total_routine_invocations	xs:nonNegativeInteger	total_routine_invocations - ルーチン呼び出しの合計回数
total_routine_time	xs:nonNegativeInteger	total_routine_time - ルーチン時間の合計
total_routine_user_code_proc_time	xs:nonNegativeInteger	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
total_routine_user_code_time	xs:nonNegativeInteger	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
total_rqst_time	xs:nonNegativeInteger	total_rqst_time - 合計要求時間
total_runstats	xs:nonNegativeInteger	total_runstats - ランタイム統計の合計回数
total_runstats_proc_time	xs:nonNegativeInteger	total_runstats_proc_time - ランタイム統計の処理時間の合計
total_runstats_time	xs:nonNegativeInteger	total_runstats_time - ランタイム統計時間の合計
total_section_proc_time	xs:nonNegativeInteger	total_section_proc_time - セクション処理時間の合計
total_section_sort_proc_time	xs:nonNegativeInteger	total_section_sort_proc_time - セクション・ソート処理時間合計
total_section_sort_time	xs:nonNegativeInteger	total_section_sort_time - セクション・ソート時間合計
total_section_sorts	xs:nonNegativeInteger	total_section_sorts - セクション・ソート合計
total_section_time	xs:nonNegativeInteger	total_section_time - セクション時間の合計
total_sorts	xs:nonNegativeInteger	total_sorts - ソート合計
total_stats_fabrication_proc_time	xs:nonNegativeInteger	total_stats_fabrication_proc_time - 統計作成の合計処理時間
total_stats_fabrication_time	xs:nonNegativeInteger	total_stats_fabrication_time - 統計作成の合計時間
total_stats_fabrications	xs:nonNegativeInteger	total_stats_fabrications - 統計作成の合計回数
total_sync_runstats	xs:nonNegativeInteger	total_sync_runstats - 同期 RUNSTATS アクティビティの合計
total_sync_runstats_proc_time	xs:nonNegativeInteger	total_sync_runstats_proc_time - 同期 RUNSTATS 処理時間の合計
total_sync_runstats_time	xs:nonNegativeInteger	total_sync_runstats_time - 同期 RUNSTATS 時間の合計
total_wait_time	xs:nonNegativeInteger	total_wait_time - 合計待ち時間
tq_sort_heap_rejections	xs:long	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否
tq_sort_heap_requests	xs:long	tq_sort_heap_requests - 表キューのソート・ヒープの要求
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
uow_id	xs:nonNegativeInteger	uow_id - 作業単位 ID
uow_log_space_used	xs:nonNegativeInteger	uow_log_space_used - 作業単位ログ・スペース
uow_start_time	xs:dateTime	uow_start_time - 作業単位開始タイム・スタンプ

表 171. MON\_GET\_UNIT\_OF\_WORK\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
wlm_queue_assignments_total	xs:nonNegativeInteger	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
wlm_queue_time_total	xs:nonNegativeInteger	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間
workload_name	xs:string (128)	workload_name - ワークロード名
workload_occurrence_id	xs:nonNegativeInteger	workload_occurrence_id - ワークロード・オカレンス ID。ワークロード・オカレンスがコーディネーター・メンバーおよびワークロード名と結合されていなければ、この ID はワークロード・オカレンスを一意的に識別しません。
workload_occurrence_state	xs:string	workload_occurrence_state - ワークロード・オカレンスの状態

## MON\_GET\_USAGE\_LIST\_STATUS 表関数 - 使用量リストの状況を戻す

MON\_GET\_USAGE\_LIST\_STATUS 表関数は、使用量リストの現在の状況を戻します。

### 構文

```

▶▶—MON_GET_USAGE_LIST_STATUS—(—usagelistschema—, —————▶
▶—usagelistname—, —member—)————▶▶▶

```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *usagelistschema*

この関数を呼び出すときに現在接続されているデータベース内の有効なスキーマ名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのスキーマの使用量リストが取得されます。引数を指定した場合は、指定したスキーマの使用量リストのみが戻されます。

#### *usagelistname*

この関数を呼び出すときに現在接続されているデータベース内にある使用量リストを指定する、タイプ VARCHAR(128) の入力引数。usagelistname が NULL または空ストリングである場合、usagelistschema で識別されるスキーマの、存在するすべての使用量リストの状況が取得されます。これを指定した場合は、usagelistschema で識別されるスキーマの、指定した使用量リストの状況のみが戻されます。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータ

ベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 使用上の注意

MON\_GET\_USAGE\_LIST\_STATUS 関数は、解放状態にある使用量リストについての情報は報告しません。使用量リストが解放状態にあると見なされるのは、それが定義されているが (明示的にも自動的に) まだアクティブ化されていない場合、または SET USAGE LIST STATE ステートメントを使って解放された場合です。

使用量リストが (明示的または自動的に) アクティブ化されると、使用量リストの定義された表または索引がセクションで最初に参照される時点までは、使用量リストの状態が「アクティブ化処理中」に設定され、使用量リストのメモリー割り振りが据え置かれます。参照された時点で、使用量リストの状態がアクティブに設定されます。使用量リストのメモリーを割り振れない場合は、使用量リストの状態が「失敗」に設定され、SET USAGE LIST STATE ステートメントを使ってこれを明示的にアクティブ化する必要があります。

## 戻される情報

表 172. MON\_GET\_USAGE\_LIST\_STATUS で戻される情報

列名	データ・タイプ	説明
USAGELISTSHEMA	VARCHAR(128)	usage_list_schema - 使用量リスト・スキーマ
USAGELISTNAME	VARCHAR(128)	usage_list_name - 使用量リスト名
OBJECTSCHEMA	VARCHAR(128)	object_schema - オブジェクト・スキーマ
OBJECTNAME	VARCHAR(128)	object_name - オブジェクト名
OBJECTTYPE	CHAR	objtype - オブジェクト・タイプ
MEMBER	SMALLINT	member - データベース・メンバー
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID
STATE	CHAR(1)	usage_list_state - 使用量リストの状態
LAST_STATE_CHANGE	TIMESTAMP	usage_list_last_state_change - 最後の状態変更
LIST_SIZE	INTEGER	usage_list_size - 使用量リストのサイズ
USED_ENTRIES	INTEGER	usage_list_used_entries - 使用量リストで使用されている項目
LIST_MEM_SIZE	INTEGER	usage_list_mem_size - 使用量リストのメモリー・サイズ
WRAPPED	CHAR	usage_list_wrapped - 使用量リストの循環標識

## MON\_GET\_WORKLOAD 表関数 - ワークロード・メトリックの取得

MON\_GET\_WORKLOAD 表関数は、1 つ以上のワークロードのメトリックを戻します。

### 構文

```
▶▶ MON_GET_WORKLOAD(—workload_name—, —member—)————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *workload\_name*

メトリックが戻される特定のワークロードを指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、すべてのワークロードについてメトリックが戻されます。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

メンバー全体で集約し、ロック待機時間の長い順に、ワークロードごとのロック情報を表示します。

```
SELECT varchar(workload_name,30) as workload_name,  
       sum(lock_wait_time) as total_lock_wait_time,  
       sum(lock_waits) as total_lock_waits,  
       sum(lock_timeouts) as total_lock_timeouts,  
       sum(lock_escals) as total_lock_escals  
FROM TABLE(MON_GET_WORKLOAD(' ', -2)) AS t  
GROUP BY workload_name  
ORDER BY total_lock_wait_time DESC
```

以下はこの照会の出力例です。

```

WORKLOAD_NAME                TOTAL_LOCK_WAIT_TIME TOTAL_LOCK_WAITS    ...
-----
SYSDEFAULTADMWORKLOAD                0                0 ...
SYSDEFAULTUSERWORKLOAD                0                0 ...

```

2 record(s) selected.

照会の出力 (続き)。

```

... TOTAL_LOCK_TIMEOUTS TOTAL_LOCK_ESCALS
... -----
...                0                0
...                0                0

```

## 使用上の注意

`MON_GET_WORKLOAD` 表関数で戻されるメトリックは、識別されたワークロード・オブジェクトにマッピングされた接続によってサブミットされた要求のすべてのメトリックの累計を表します。メトリックは、作業単位境界でワークロードにロールアップされ、要求の実行中には定期的にロールアップされます。したがって、この表関数で報告される値は、直前の `ROLLUP` 時のシステムの現行状態を反映しています。メトリックの値は確実に増加します。ある時間間隔に対する指定されたメトリックの値を判別するには、`MON_GET_WORKLOAD` 表関数を使用してその間隔の始めと終わりのメトリックを照会し、差異を計算します。

要求メトリックは、サービス・スーパークラスに対する `COLLECT REQUEST METRICS` 節、およびデータベース・レベルの `mon_req_metrics` データベース構成パラメーターを介して制御されます。親サービス・スーパークラスで要求メトリックを使用可能にしているサービス・サブクラスのエージェントが要求を処理する場合にのみ、またはデータベース全体で要求メトリック・コレクションが有効な場合にのみ、その要求に対しメトリックが収集されます。デフォルトでは、要求メトリックはデータベース・レベルで使用可能です。要求メトリックがデータベース・レベルでもサービス・スーパークラスに対しても使用不可になっている場合、そのサービス・スーパークラスにマッピングされたワークロードごとに報告されるメトリックは増加を停止します (または、要求メトリックがデータベースのアクティブ化時に無効であった場合には 0 のままになります)。

`MON_GET_WORKLOAD` 表関数は、ワークロードおよびメンバーごとに 1 行のデータを戻します。(メンバー上の) ワークロード全体または (1 つ以上のサービス・クラスの) メンバー全体の集約は実行されません。しかし、集約は例に示されるように SQL 照会を使用して実行できます (例を参照)。

## 戻される情報

表 173. `MON_GET_WORKLOAD` について戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
<code>WORKLOAD_NAME</code>	<code>VARCHAR(128)</code>	<code>workload_name</code> - ワークロード名
<code>WORKLOAD_ID</code>	<code>INTEGER</code>	<code>workload_id</code> - ワークロード ID
<code>MEMBER</code>	<code>SMALLINT</code>	<code>member</code> - データベース・メンバー
<code>ACT_ABORTED_TOTAL</code>	<code>BIGINT</code>	<code>act_aborted_total</code> - 打ち切られたアクティビティの合計数



表 173. MON\_GET\_WORKLOAD について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 完了したアクティビティーの合計数
ACT_REJECTED_TOTAL	BIGINT	act_rejected_total - リジェクトされたアクティビティーの合計数
AGENT_WAIT_TIME	BIGINT	agent_wait_time - エージェント待機時間
AGENT_WAITS_TOTAL	BIGINT	agent_waits_total - エージェント待機の合計
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込み
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDA データの書き込み
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
CLIENT_IDLE_WAIT_TIME	BIGINT	client_idle_wait_time - クライアントのアイドル待機時間
DEADLOCKS	BIGINT	deadlocks - デッドロック検出数

表 173. MON\_GET\_WORKLOAD について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
FCM_RECV_VOLUME	BIGINT	fcm_recv_volume - FCM 受信ボリューム
FCM_RECVS_TOTAL	BIGINT	fcm_recvs_total - FCM 合計受信数
FCM_SEND_VOLUME	BIGINT	fcm_send_volume - FCM 送信ボリューム
FCM_SENDS_TOTAL	BIGINT	fcm_sends_total - FCM 合計送信数
FCM_RECV_WAIT_TIME	BIGINT	fcm_recv_wait_time - FCM 受信待機時間
FCM_SEND_WAIT_TIME	BIGINT	fcm_send_wait_time - FCM 送信待ち時間
IPC_RECV_VOLUME	BIGINT	ipc_recv_volume - プロセス間通信受信ボリューム
IPC_RECV_WAIT_TIME	BIGINT	ipc_recv_wait_time - プロセス間通信受信待ち時間
IPC_RECVS_TOTAL	BIGINT	ipc_recvs_total - プロセス間通信合計受信数
IPC_SEND_VOLUME	BIGINT	ipc_send_volume - プロセス間通信送信ボリューム
IPC_SEND_WAIT_TIME	BIGINT	ipc_send_wait_time - プロセス間通信送信待ち時間
IPC_SENDS_TOTAL	BIGINT	ipc_sends_total - プロセス間通信合計送信数
LOCK_ESCALS	BIGINT	lock_escalations - ロック・エスカレーション数
LOCK_TIMEOUTS	BIGINT	lock_timeouts - ロック・タイムアウト数
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
LOG_BUFFER_WAIT_TIME	BIGINT	log_buffer_wait_time - ログ・バッファ待ち時間
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - フル・ログ・バッファの回数
LOG_DISK_WAIT_TIME	BIGINT	log_disk_wait_time - ログ・ディスク待機時間
LOG_DISK_WAITS_TOTAL	BIGINT	log_disk_waits_total - ログ・ディスク待機の合計
RQSTS_COMPLETED_TOTAL	BIGINT	rqsts_completed_total - 完了した要求の合計数
ROWS_MODIFIED	BIGINT	rows_modified - 変更された行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_RETURNED	BIGINT	rows_returned - 戻り行数
TCPIP_RECV_VOLUME	BIGINT	tcPIP_recv_volume - TCP/IP 受信ボリューム
TCPIP_SEND_VOLUME	BIGINT	tcPIP_send_volume - TCP/IP 送信ボリューム
TCPIP_RECV_WAIT_TIME	BIGINT	tcPIP_recv_wait_time - TCP/IP 受信待ち時間
TCPIP_RECVS_TOTAL	BIGINT	tcPIP_recvs_total - TCP/IP 合計受信数
TCPIP_SEND_WAIT_TIME	BIGINT	tcPIP_send_wait_time - TCP/IP 送信待ち時間
TCPIP_SENDS_TOTAL	BIGINT	tcPIP_sends_total - TCP/IP 合計送信数
TOTAL_APP_RQST_TIME	BIGINT	total_app_rqst_time - アプリケーション要求合計時間
TOTAL_RQST_TIME	BIGINT	total_rqst_time - 合計要求時間
WLM_QUEUE_TIME_TOTAL	BIGINT	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間

表 173. MON\_GET\_WORKLOAD について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
WLM_QUEUE_ASSIGNMENTS_TOTAL	BIGINT	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間
TOTAL_WAIT_TIME	BIGINT	total_wait_time - 合計待ち時間
APP_RQSTS_COMPLETED_TOTAL	BIGINT	app_rqsts_completed_total - 完了したアプリケーション要求の合計数
TOTAL_SECTION_SORT_TIME	BIGINT	total_section_sort_time - セクション・ソート時間合計
TOTAL_SECTION_SORT_PROC_TIME	BIGINT	total_section_sort_proc_time - セクション・ソート処理時間合計
TOTAL_SECTION_SORTS	BIGINT	total_section_sorts - セクション・ソート合計
TOTAL_SORTS	BIGINT	total_sorts - ソート合計
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - ポストしきい値ソート
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - ポスト共有しきい値ソート
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
TOTAL_COMPILE_TIME	BIGINT	total_compile_time - コンパイル時間の合計
TOTAL_COMPILE_PROC_TIME	BIGINT	total_compile_proc_time - コンパイル処理時間の合計
TOTAL_COMPILATIONS	BIGINT	total_compilations - コンパイルの合計回数
TOTAL_IMPLICIT_COMPILE_TIME	BIGINT	total_implicit_compile_time - 暗黙的コンパイル時間の合計
TOTAL_IMPLICIT_COMPILE_PROC_TIME	BIGINT	total_implicit_compile_proc_time - 暗黙的コンパイルの処理時間の合計
TOTAL_IMPLICIT_COMPILATIONS	BIGINT	total_implicit_compilations - 暗黙的コンパイルの合計回数
TOTAL_SECTION_TIME	BIGINT	total_section_time - セクション時間の合計
TOTAL_SECTION_PROC_TIME	BIGINT	total_section_proc_time - セクション処理時間の合計
TOTAL_APP_SECTION_EXECUTIONS	BIGINT	total_app_section_executions - セクション実行の合計回数
TOTAL_ACT_TIME	BIGINT	total_activity_time - 合計アクティビティー時間
TOTAL_ACT_WAIT_TIME	BIGINT	total_activity_wait_time - 合計アクティビティー待機時間
ACT_RQSTS_TOTAL	BIGINT	act_rqsts_total - アクティビティー要求の合計
TOTAL_ROUTINE_TIME	BIGINT	total_routine_time - ルーチン時間の合計
TOTAL_ROUTINE_INVOCATIONS	BIGINT	total_routine_invocations - ルーチン呼び出しの合計回数
TOTAL_COMMIT_TIME	BIGINT	total_commit_time - コミット時間の合計
TOTAL_COMMIT_PROC_TIME	BIGINT	total_commit_proc_time - コミット処理時間の合計
TOTAL_APP_COMMITS	BIGINT	total_app_commits - アプリケーション・コミットの合計回数
INT_COMMITS	BIGINT	int_commits - 内部コミット数
TOTAL_ROLLBACK_TIME	BIGINT	total_rollback_time - ロールバック時間の合計
TOTAL_ROLLBACK_PROC_TIME	BIGINT	total_rollback_proc_time - ロールバック処理時間の合計

表 173. MON\_GET\_WORKLOAD について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_APP_ROLLBACKS	BIGINT	total_app_rollback - アプリケーション・ロールバックの合計回数
INT_ROLLBACKS	BIGINT	int_rollback - 内部ロールバック数
TOTAL_RUNSTATS_TIME	BIGINT	total_runstats_time - ランタイム統計時間の合計
TOTAL_RUNSTATS_PROC_TIME	BIGINT	total_runstats_proc_time - ランタイム統計の処理時間の合計
TOTAL_RUNSTATS	BIGINT	total_runstats - ランタイム統計の合計回数
TOTAL_REORG_TIME	BIGINT	total_reorg_time - 再編成時間の合計
TOTAL_REORG_PROC_TIME	BIGINT	total_reorg_proc_time - 再編成の処理時間の合計
TOTAL_REORGS	BIGINT	total_reorgs - 再編成の合計回数
TOTAL_LOAD_TIME	BIGINT	total_load_time - ロード時間の合計
TOTAL_LOAD_PROC_TIME	BIGINT	total_load_proc_time - ロード処理時間の合計
TOTAL_LOADS	BIGINT	total_loads - ロードの合計回数
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - カタログ・キャッシュ挿入数
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - カタログ・キャッシュ参照数
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - パッケージ・キャッシュ挿入
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - パッケージ・キャッシュ参照
THRESH_VIOLATIONS	BIGINT	hresh_violations - しきい値違反の回数
NUM_LW_THRESH_EXCEEDED	BIGINT	num_lw_thresh_exceeded - しきい値を超えた回数
LOCK_WAITS_GLOBAL	BIGINT	lock_waits_global - グローバル・ロック待機
LOCK_WAIT_TIME_GLOBAL	BIGINT	lock_wait_time_global - グローバル・ロック待機時間
LOCK_TIMEOUTS_GLOBAL	BIGINT	lock_timeouts_global - グローバル・ロック・タイムアウト
LOCK_ESCALS_MAXLOCKS	BIGINT	lock_escals_maxlocks - maxlocks ロック・エスカレーション数
LOCK_ESCALS_LOCKLIST	BIGINT	lock_escals_locklist - locklist ロック・エスカレーション数
LOCK_ESCALS_GLOBAL	BIGINT	lock_escals_global - グローバル・ロック・エスカレーション数
RECLAIM_WAIT_TIME	BIGINT	reclaim_wait_time - 再利用の待機時間
SPACEMAPPAGE_RECLAIM_WAIT_TIME	BIGINT	spacemappage_reclaim_wait_time - スペース・マップ・ページの再利用の待機時間
CF_WAITS	BIGINT	cf_waits - クラスター・キャッシング・ファシリティー待機回数
CF_WAIT_TIME	BIGINT	cf_wait_time - クラスター・キャッシング・ファシリティーの待機時間
POOL_DATA_GBP_L_READS	BIGINT	pool_data_gbp_l_reads - グループ・バッファ・プールのデータの論理読み取り
POOL_DATA_GBP_P_READS	BIGINT	pool_data_gbp_p_reads - グループ・バッファ・プールのデータの物理読み取り
POOL_DATA_LBP_PAGES_FOUND	BIGINT	pool_data_lbp_pages_found - ローカル・バッファ・プールで検出されたデータ・ページ

表 173. MON\_GET\_WORKLOAD について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_DATA_GBP_INVALID_PAGES	BIGINT	pool_data_gbp_invalid_pages - グループ・バッファ・プールの無効なデータ・ページ
POOL_INDEX_GBP_L_READS	BIGINT	pool_index_gbp_l_reads - グループ・バッファ・プールの索引の論理読み取り
POOL_INDEX_GBP_P_READS	BIGINT	pool_index_gbp_p_reads - グループ・バッファ・プールの索引の物理読み取り
POOL_INDEX_LBP_PAGES_FOUND	BIGINT	pool_index_lbp_pages_found - 検出されたローカル・バッファ・プールの索引ページ
POOL_INDEX_GBP_INVALID_PAGES	BIGINT	pool_index_gbp_invalid_pages - グループ・バッファ・プールの無効な索引ページ
POOL_XDA_GBP_L_READS	BIGINT	pool_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの論理読み取りの要求
POOL_XDA_GBP_P_READS	BIGINT	pool_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの物理読み取りの要求
POOL_XDA_LBP_PAGES_FOUND	BIGINT	pool_xda_lbp_pages_found - 検出されたローカル・バッファ・プールの XDA データ・ページ
POOL_XDA_GBP_INVALID_PAGES	BIGINT	pool_xda_gbp_invalid_pages - グループ・バッファ・プールの無効な XDA データ・ページ
AUDIT_EVENTS_TOTAL	BIGINT	audit_events_total - 監査イベント合計数
AUDIT_FILE_WRITES_TOTAL	BIGINT	audit_file_writes_total - 書き込まれた監査ファイルの合計数
AUDIT_FILE_WRITE_WAIT_TIME	BIGINT	audit_file_write_wait_time - 監査ファイル書き込み待機時間
AUDIT_SUBSYSTEM_WAITS_TOTAL	BIGINT	audit_subsystem_waits_total - 監査サブシステム待機の合計
AUDIT_SUBSYSTEM_WAIT_TIME	BIGINT	audit_subsystem_wait_time - 監査サブシステム待機時間
DIAGLOG_WRITES_TOTAL	BIGINT	diaglog_writes_total - 診断ログ・ファイル書き込みの合計
DIAGLOG_WRITE_WAIT_TIME	BIGINT	diaglog_write_wait_time - 診断ログ・ファイルの書き込み待機時間
FCM_MESSAGE_RECVS_TOTAL	BIGINT	fcm_message_recvs_total - FCM メッセージ合計受信数
FCM_MESSAGE_RECV_VOLUME	BIGINT	fcm_message_recv_volume - FCM メッセージ受信ボリューム
FCM_MESSAGE_RECV_WAIT_TIME	BIGINT	fcm_message_recv_wait_time - FCM メッセージの受信待機時間
FCM_MESSAGE_SENDS_TOTAL	BIGINT	fcm_message_sends_total - FCM メッセージ合計送信数
FCM_MESSAGE_SEND_VOLUME	BIGINT	fcm_message_send_volume - FCM メッセージ送信ボリューム
FCM_MESSAGE_SEND_WAIT_TIME	BIGINT	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
FCM_TQ_RECVS_TOTAL	BIGINT	fcm_tq_recvs_total - FCM 表キューの合計受信数
FCM_TQ_RECV_VOLUME	BIGINT	fcm_tq_recv_volume - FCM 表キューの受信ボリューム
FCM_TQ_RECV_WAIT_TIME	BIGINT	fcm_tq_recv_wait_time - FCM 表キューの受信待ち時間

表 173. MON\_GET\_WORKLOAD について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
FCM_TQ_SENDS_TOTAL	BIGINT	fcm_tq_sends_total - FCM 表キューの合計送信数
FCM_TQ_SEND_VOLUME	BIGINT	fcm_tq_send_volume - FCM 表キューの送信ボリューム
FCM_TQ_SEND_WAIT_TIME	BIGINT	fcm_tq_send_wait_time - FCM 表キューの送信待ち時間
TOTAL_ROUTINE_USER_CODE_PROC_TIME	BIGINT	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
TOTAL_ROUTINE_USER_CODE_TIME	BIGINT	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
TQ_TOT_SEND_SPILLS	BIGINT	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
EVMON_WAIT_TIME	BIGINT	evmon_wait_time - イベント・モニターの待機時間
EVMON_WAITS_TOTAL	BIGINT	evmon_waits_total - イベント・モニターの待機の合計回数
TOTAL_CONNECT_REQUEST_TIME	BIGINT	total_connect_request_time - 接続またはユーザー切り替えの要求の合計時間モニター・エレメント
TOTAL_CONNECT_REQUEST_PROC_TIME	BIGINT	total_connect_request_proc_time - 接続要求またはユーザー切り替え要求の合計処理時間モニター・エレメント
TOTAL_CONNECT_REQUESTS	BIGINT	total_connect_requests - 接続またはユーザー切り替えの要求の数モニター・エレメント
TOTAL_CONNECT_AUTHENTICATION_TIME	BIGINT	total_connect_authentication_time - 接続またはユーザー切り替えの認証要求の合計時間モニター・エレメント
TOTAL_CONNECT_AUTHENTICATION_PROC_TIME	BIGINT	total_connect_authentication_proc_time - 接続認証処理の合計時間モニター・エレメント
TOTAL_CONNECT_AUTHENTICATIONS	BIGINT	total_connect_authentications - 接続またはユーザー切り替えの認証の実行回数モニター・エレメント
TOTAL_EXTENDED_LATCH_WAIT_TIME	BIGINT	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
TOTAL_EXTENDED_LATCH_WAITS	BIGINT	total_extended_latch_waits - 拡張ラッチの合計待機回数
TOTAL_STATS_FABRICATION_TIME	BIGINT	total_stats_fabrication_time - 統計作成の合計時間
TOTAL_STATS_FABRICATION_PROC_TIME	BIGINT	total_stats_fabrication_proc_time - 統計作成の合計処理時間
TOTAL_STATS_FABRICATIONS	BIGINT	total_stats_fabrications - 統計作成の合計回数
TOTAL_SYNC_RUNSTATS_TIME	BIGINT	total_sync_runstats_time - 同期 RUNSTATS 時間の合計
TOTAL_SYNC_RUNSTATS_PROC_TIME	BIGINT	total_sync_runstats_proc_time - 同期 RUNSTATS 処理時間の合計
TOTAL_SYNC_RUNSTATS	BIGINT	total_sync_runstats - 同期 RUNSTATS アクティビティの合計
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - ディスパッチャー実行キュー時間の合計
DISABLED_PEDS	BIGINT	disabled_peds - 無効化された partial early distinct
POST_THRESHOLD_PEAS	BIGINT	post_threshold_peas - partial early aggregation しきい値

表 173. MON\_GET\_WORKLOAD について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POST_THRESHOLD_PEDS	BIGINT	post_threshold_peds - partial early distinct しきい値
TOTAL_PEAS	BIGINT	total_peas - partial early aggregation の合計回数
TOTAL_PEDS	BIGINT	total_peds - partial early distinct の合計回数
TQ_SORT_HEAP_REJECTIONS	BIGINT	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否
TQ_SORT_HEAP_REQUESTS	BIGINT	tq_sort_heap_requests - 表キューのソート・ヒープの要求
APP_ACT_ABORTED_TOTAL	BIGINT	app_act_aborted_total - 失敗した外部コーディネーター・アクティビティの総数モニター・エレメント
APP_ACT_COMPLETED_TOTAL	BIGINT	app_act_completed_total - 成功した外部コーディネーター・アクティビティの総数モニター・エレメント
APP_ACT_REJECTED_TOTAL	BIGINT	app_act_rejected_total - リジェクトされた外部コーディネーター・アクティビティの総数モニター・エレメント
POOL_QUEUED_ASYNC_DATA_REQS	BIGINT	pool_queued_async_data_reqs - データ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_INDEX_REQS	BIGINT	pool_queued_async_index_reqs - 索引プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_XDA_REQS	BIGINT	pool_queued_async_xda_reqs - XDA プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_OTHER_REQS	BIGINT	pool_queued_async_other_reqs - 非プリフェッチの要求モニター・エレメント
POOL_QUEUED_ASYNC_DATA_PAGES	BIGINT	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_INDEX_PAGES	BIGINT	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_XDA_PAGES	BIGINT	pool_queued_async_xda_pages - XDA ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_DATA_PAGES	BIGINT	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求モニター・エレメント
POOL_QUEUED_ASYNC_TEMP_INDEX_PAGES	BIGINT	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求モニター・エレメント

表 173. MON\_GET\_WORKLOAD について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_QUEUED_ASYNC_TEMP_XDA_PAGES	BIGINT	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_DATA_REQS	BIGINT	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_INDEX_REQS	BIGINT	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_XDA_REQS	BIGINT	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_TEMP_DATA_REQS	BIGINT	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_TEMP_INDEX_REQS	BIGINT	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_TEMP_XDA_REQS	BIGINT	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求モニター・エレメント
POOL_FAILED_ASYNC_OTHER_REQS	BIGINT	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求モニター・エレメント
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間
PREFETCH_WAITS	BIGINT	prefetch_waits - プリフェッチャーの待機カウント・モニター・エレメント
POOL_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ・モニター・エレメント
POOL_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ・モニター・エレメント
POOL_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP	BIGINT	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ・モニター・エレメント
COMM_EXIT_WAIT_TIME	BIGINT	comm_exit_wait_time - 通信バッファ出口待機時間モニター・エレメント
COMM_EXIT_WAITS	BIGINT	comm_exit_waits - 通信バッファ出口待機回数モニター・エレメント
IDA_SEND_WAIT_TIME	BIGINT	インデータベース分析処理にデータを送信するために待機した合計時間。
IDA_SENDS_TOTAL	BIGINT	インデータベース分析処理にデータが送信された合計回数。
IDA_SEND_VOLUME	BIGINT	インデータベース分析処理にデータベース・サーバーから送信されたデータの合計量。値はバイト数で報告されます。



表 173. MON\_GET\_WORKLOAD について戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
IDA_RECV_WAIT_TIME	BIGINT	インデータベース分析処理からデータを受信するために待機した合計時間。
IDA_RECVS_TOTAL	BIGINT	インデータベース分析処理からデータを受け取った合計回数。
IDA_RECV_VOLUME	BIGINT	インデータベース分析処理からデータベース・サーバーが受け取ったデータの合計量。値はバイト数で報告されます。

## MON\_GET\_WORKLOAD\_DETAILS 表関数 - 詳細ワークロード・メトリックの取得

MON\_GET\_WORKLOAD\_DETAILS 表関数は、1 つ以上のワークロードの詳細メトリックを戻します。

### 構文

```
▶▶—MON_GET_WORKLOAD_DETAILS—(—workload_name—,—member—)————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *workload\_name*

メトリックが戻される特定のワークロードを指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、すべてのワークロードについてメトリックが戻されます。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

## 例

メンバー全体で集約し、ロック待機時間の長い順に、ワークロードごとのロック情報を表示します。

```
SELECT varchar(wlmetrics.workload_name,30) as workload_name,
       sum(detmetrics.lock_wait_time) as total_lock_wait_time,
       sum(detmetrics.lock_waits) as total_lock_waits,
       sum(detmetrics.lock_timeouts) as total_lock_timeouts,
       sum(detmetrics.lock_escals) as total_lock_escals
FROM TABLE(MON_GET_WORKLOAD_DETAILS('-',-2)) AS WLMETRICS,
XMLTABLE (XMLNAMESPACES( DEFAULT 'http://www.ibm.com/xmlns/prod/db2/mon'),
         '$detmetric/db2_workload' PASSING
         XMLPARSE(DOCUMENT WLMETRICS.DETAILS)
         as "detmetric"
COLUMNS "LOCK_WAIT_TIME" INTEGER PATH 'system_metrics/lock_wait_time',
         "LOCK_WAITS" INTEGER PATH 'system_metrics/lock_waits',
         "LOCK_TIMEOUTS" INTEGER PATH 'system_metrics/lock_timeouts',
         "LOCK_ESCALS" INTEGER PATH 'system_metrics/lock_escals'
) AS DETMETRICS
GROUP BY workload_name
ORDER BY total_lock_wait_time desc;
```

以下はこの照会の出力例です。

WORKLOAD_NAME	TOTAL_LOCK_WAIT_TIME	TOTAL_LOCK_WAITS	...
SYSDEFAULTADMWORKLOAD	0	0	...
SYSDEFAULTUSERWORKLOAD	0	0	...

2 record(s) selected.

照会の出力 (続き)。

...	TOTAL_LOCK_TIMEOUTS	TOTAL_LOCK_ESCALS
...	0	0
...	0	0

## 使用上の注意

MON\_GET\_WORKLOAD 関数で戻されるメトリックは、識別されたワークロード・オブジェクトにマッピングされた接続によってサブミットされた要求のすべてのメトリックの累計を表します。この関数は、以下の点で MON\_GET\_WORKLOAD 表関数と似ています。

- MON\_GET\_WORKLOAD 表関数は、最も一般的に使用されるメトリックを列ベースの形式で戻すため、メトリックの取得において最も効率的な方法です。
- MON\_GET\_WORKLOAD\_DETAILS 表関数は、使用可能なすべてのメトリック形式を XML 文書形式で戻すため、出力フォーマットを最大限柔軟性のあるものにします。XML ベースの出力は、XML パーサーで直接解析でき、XMLTABLE 関数でリレーショナル形式に変換することもできます (例を参照)。

メトリックは、作業単位境界でワークロードにロールアップされ、要求の実行中には定期的にロールアップされます。したがって、この表関数で報告される値は、直前の ROLLUP 時のシステムの現行状態を反映しています。メトリックの値は確実に増加します。ある時間間隔に対する指定されたメトリックの値を判別するには、MON\_GET\_WORKLOAD\_DETAILS 表関数を使用してその間隔の始めと終わりのメトリックを照会し、差異を計算します。

要求メトリックは、サービス・スーパークラスに対する COLLECT REQUEST METRICS 節、およびデータベース・レベルの `mon_req_metrics` データベース構成パラメーターを介して制御されます。親サービス・スーパークラスで要求メトリックを使用可能にしているサービス・サブクラスのエージェントが要求を処理する場合にのみ、またはデータベース全体で要求メトリック・コレクションが有効な場合にのみ、その要求に対しメトリックが収集されます。デフォルトでは、要求メトリックはデータベース・レベルで使用可能です。要求メトリックがデータベース・レベルでもサービス・スーパークラスに対しても使用不可になっている場合、そのサービス・スーパークラスにマッピングされたワークロードごとに報告されるメトリックは増加を停止します (または、要求メトリックがデータベースのアクティブ化時に無効であった場合には 0 のままになります)。

MON\_GET\_WORKLOAD\_DETAILS 表関数は、ワークロードおよびメンバーごとに 1 行のデータを戻します。(メンバー上の) ワークロード全体または (1 つ以上のサービス・クラスの) メンバー全体の集約は実行されません。ただし、集約は例に示されるように SQL 照会を使用して実行できます。

DETAILS 列に戻される XML 文書のスキーマは、ファイル `sqllib/misc/DB2MonRoutines.xsd` で入手できます。詳細は、ファイル `sqllib/misc/DB2MonCommon.xsd` 内にあります。

## 戻される情報

表 174. MON\_GET\_WORKLOAD\_DETAILS について戻される情報

列名	データ・タイプ	説明
WORKLOAD_NAME	VARCHAR(128)	workload_name - ワークロード名
WORKLOAD_ID	INTEGER	workload_id - ワークロード ID
MEMBER	SMALLINT	member - データベース・メンバー
DETAILS	BLOB(1M)	ワークロードについての詳細メトリックを含む XML 文書。エレメントの説明については、本書の表 175を参照してください。

以下の例は、DETAILS 列で戻される XML 文書の構造を示しています。

```
<db2_workload xmlns="http://www.ibm.com/xmlns/prod/db2/mon" release="90700000">
  <workload_name>SYSDEFAULTADMWORKLOAD</workload_name>
  <workload_id>11</workload_id>
  <member>0</member>
  <system_metrics release="90700000">
    <act_aborted_total>5</act_aborted_total>
    ...
    <wlm_queue_assignments_total>3</wlm_queue_assignments_total>
  </system_metrics>
</db2_workload>
```

完全スキーマについては、`sqllib/misc/DB2MonRoutines.xsd` を参照してください。

表 175. MON\_GET\_WORKLOAD\_DETAILS について戻される詳細メトリック

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
act_aborted_total	xs:nonNegativeInteger	act_aborted_total - 打ち切られたアクティビティーの合計数
act_completed_total	xs:nonNegativeInteger	act_completed_total - 完了したアクティビティーの合計数

表 175. MON\_GET\_WORKLOAD\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
act_rejected_total	xs:nonNegativeInteger	act_rejected_total - リジェクトされたアクティビティーの合計数
act_rqsts_total	xs:nonNegativeInteger	act_rqsts_total - アクティビティー要求の合計
agent_wait_time	xs:nonNegativeInteger	agent_wait_time - エージェント待機時間
agent_waits_total	xs:nonNegativeInteger	agent_waits_total - エージェント待機の合計
app_act_aborted_total	xs:nonNegativeInteger	app_act_aborted_total - 失敗した外部コーディネーター・アクティビティーの総数モニター・エレメント
app_act_completed_total	xs:nonNegativeInteger	app_act_completed_total - 成功した外部コーディネーター・アクティビティーの総数モニター・エレメント
app_act_rejected_total	xs:nonNegativeInteger	app_act_rejected_total - リジェクトされた外部コーディネーター・アクティビティーの総数モニター・エレメント
app_rqsts_completed_total	xs:nonNegativeInteger	app_rqsts_completed_total - 完了したアプリケーション要求の合計数
audit_events_total	xs:nonNegativeInteger	audit_events_total - 監査イベント合計数
audit_file_write_wait_time	xs:nonNegativeInteger	audit_file_write_wait_time - 監査ファイル書き込み待機時間
audit_file_writes_total	xs:nonNegativeInteger	audit_file_writes_total - 書き込まれた監査ファイルの合計数
audit_subsystem_wait_time	xs:nonNegativeInteger	audit_subsystem_wait_time - 監査サブシステム待機時間
audit_subsystem_waits_total	xs:nonNegativeInteger	audit_subsystem_waits_total - 監査サブシステム待機の合計
cat_cache_inserts	xs:nonNegativeInteger	cat_cache_inserts - カタログ・キャッシュ挿入数
cat_cache_lookups	xs:nonNegativeInteger	cat_cache_lookups - カタログ・キャッシュ参照数
client_idle_wait_time	xs:nonNegativeInteger	client_idle_wait_time - クライアントのアイドル待機時間
comm_exit_wait_time	xs:nonNegativeInteger	comm_exit_wait_time - 通信バッファ出口待機時間モニター・エレメント
comm_exit_waits	xs:nonNegativeInteger	comm_exit_waits - 通信バッファ出口待機回数モニター・エレメント
deadlocks	xs:nonNegativeInteger	deadlocks - デッドロック検出数
diaglog_write_wait_time	xs:nonNegativeInteger	diaglog_write_wait_time - 診断ログ・ファイルの書き込み待機時間
diaglog_writes_total	xs:nonNegativeInteger	diaglog_writes_total - 診断ログ・ファイル書き込みの合計
direct_read_reqs	xs:nonNegativeInteger	direct_read_reqs - 直接読み取り要求
direct_read_time	xs:nonNegativeInteger	direct_read_time - 直接読み取り時間
direct_reads	xs:nonNegativeInteger	direct_reads - データベースからの直接読み取り
direct_write_reqs	xs:nonNegativeInteger	direct_write_reqs - 直接書き込み要求
direct_write_time	xs:nonNegativeInteger	direct_write_time - 直接書き込み時間
direct_writes	xs:nonNegativeInteger	direct_writes - データベースへの直接書き込み
disabled_peds	xs:long	disabled_peds - 無効化された partial early distinct
evmon_wait_time	xs:nonNegativeInteger	evmon_wait_time - イベント・モニターの待機時間
evmon_waits_total	xs:nonNegativeInteger	evmon_waits_total - イベント・モニターの待機の合計回数

表 175. MON\_GET\_WORKLOAD\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
fcm_message_recv_volume	xs:nonNegativeInteger	fcm_message_recv_volume - FCM メッセージ受信ボリューム
fcm_message_recv_wait_time	xs:nonNegativeInteger	fcm_message_recv_wait_time - FCM メッセージの受信待機時間
fcm_message_recvs_total	xs:nonNegativeInteger	fcm_message_recvs_total - FCM メッセージ合計受信数
fcm_message_send_volume	xs:nonNegativeInteger	fcm_message_send_volume - FCM メッセージ送信ボリューム
fcm_message_send_wait_time	xs:nonNegativeInteger	fcm_message_send_wait_time - FCM メッセージ送信待ち時間
fcm_message_sends_total	xs:nonNegativeInteger	fcm_message_sends_total - FCM メッセージ合計送信数
fcm_recv_volume	xs:nonNegativeInteger	fcm_recv_volume - FCM 受信ボリューム
fcm_recv_wait_time	xs:nonNegativeInteger	fcm_recv_wait_time - FCM 受信待機時間
fcm_recvs_total	xs:nonNegativeInteger	fcm_recvs_total - FCM 合計受信数
fcm_send_volume	xs:nonNegativeInteger	fcm_send_volume - FCM 送信ボリューム
fcm_send_wait_time	xs:nonNegativeInteger	fcm_send_wait_time - FCM 送信待ち時間
fcm_sends_total	xs:nonNegativeInteger	fcm_sends_total - FCM 合計送信数
fcm_tq_recv_volume	xs:nonNegativeInteger	fcm_tq_recv_volume - FCM 表キューの受信ボリューム
fcm_tq_recv_wait_time	xs:nonNegativeInteger	fcm_tq_recv_wait_time - FCM 表キューの受信待ち時間
fcm_tq_recvs_total	xs:nonNegativeInteger	fcm_tq_recvs_total - FCM 表キューの合計受信数
fcm_tq_send_volume	xs:nonNegativeInteger	fcm_tq_send_volume - FCM 表キューの送信ボリューム
fcm_tq_send_wait_time	xs:nonNegativeInteger	fcm_tq_send_wait_time - FCM 表キューの送信待ち時間
fcm_tq_sends_total	xs:nonNegativeInteger	fcm_tq_sends_total - FCM 表キューの合計送信数
ida_recv_wait_time	xs:nonNegativeInteger	インデータベース分析処理からデータを受信するために待機した合計時間。
ida_recv_volume	xs:nonNegativeInteger	インデータベース分析処理からデータベース・サーバーが受け取ったデータの合計量。値はバイト数で報告されます。
ida_recvs_total	xs:nonNegativeInteger	インデータベース分析処理からデータを受け取った合計回数。
ida_send_wait_time	xs:nonNegativeInteger	インデータベース分析処理にデータを送信するために待機した合計時間。
ida_send_volume	xs:nonNegativeInteger	インデータベース分析処理にデータベース・サーバーから送信されたデータの合計量。値はバイト数で報告されます。
ida_sends_total	xs:nonNegativeInteger	インデータベース分析処理にデータが送信された合計回数。
int_commits	xs:nonNegativeInteger	int_commits - 内部コミット数
int_rollbacks	xs:nonNegativeInteger	int_rollbacks - 内部ロールバック数
ipc_recv_volume	xs:nonNegativeInteger	ipc_recv_volume - プロセス間通信受信ボリューム
ipc_recv_wait_time	xs:nonNegativeInteger	ipc_recv_wait_time - プロセス間通信受信待ち時間
ipc_recvs_total	xs:nonNegativeInteger	ipc_recvs_total - プロセス間通信合計受信数
ipc_send_volume	xs:nonNegativeInteger	ipc_send_volume - プロセス間通信送信ボリューム

表 175. MON\_GET\_WORKLOAD\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
ipc_send_wait_time	xs:nonNegativeInteger	ipc_send_wait_time - プロセス間通信送信待ち時間
ipc_sends_total	xs:nonNegativeInteger	ipc_sends_total - プロセス間通信合計送信数
lock_escals	xs:nonNegativeInteger	lock_escals - ロック・エスカレーション数
lock_timeouts	xs:nonNegativeInteger	lock_timeouts - ロック・タイムアウト数
lock_wait_time	xs:nonNegativeInteger	lock_wait_time - ロック待機中の時間
lock_waits	xs:nonNegativeInteger	lock_waits - ロック待機数
log_buffer_wait_time	xs:nonNegativeInteger	log_buffer_wait_time - ログ・バッファ待ち時間
log_disk_wait_time	xs:nonNegativeInteger	log_disk_wait_time - ログ・ディスク待機時間
log_disk_waits_total	xs:nonNegativeInteger	log_disk_waits_total - ログ・ディスク待機の合計
member	xs:nonNegativeInteger	member - データベース・メンバー
num_log_buffer_full	xs:nonNegativeInteger	num_log_buffer_full - フル・ログ・バッファの回数
num_lw_thresh_exceeded	xs:nonNegativeInteger	num_lw_thresh_exceeded - しきい値を超えた回数
pkg_cache_inserts	xs:nonNegativeInteger	pkg_cache_inserts - パッケージ・キャッシュ挿入
pkg_cache_lookups	xs:nonNegativeInteger	pkg_cache_lookups - パッケージ・キャッシュ参照
pool_data_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_data_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存データ・ページ・モニター・エレメント
pool_data_l_reads	xs:nonNegativeInteger	pool_data_l_reads - バッファ・プール・データの論理読み取り
pool_data_p_reads	xs:nonNegativeInteger	pool_data_p_reads - バッファ・プール・データの物理読み取り
pool_data_writes	xs:nonNegativeInteger	pool_data_writes - バッファ・プールへのデータの書き込み
pool_failed_async_data_reqs	xs:nonNegativeInteger	pool_failed_async_data_reqs - 失敗したデータ・プリフェッチ要求モニター・エレメント
pool_failed_async_index_reqs	xs:nonNegativeInteger	pool_failed_async_index_reqs - 失敗した索引プリフェッチ要求モニター・エレメント
pool_failed_async_other_reqs	xs:nonNegativeInteger	pool_failed_async_other_reqs - 失敗した非プリフェッチの要求モニター・エレメント
pool_failed_async_temp_data_reqs	xs:nonNegativeInteger	pool_failed_async_temp_data_reqs - 失敗した TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
pool_failed_async_temp_index_reqs	xs:nonNegativeInteger	pool_failed_async_temp_index_reqs - 失敗した TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
pool_failed_async_temp_xda_reqs	xs:nonNegativeInteger	pool_failed_async_temp_xda_reqs - 失敗した TEMPORARY 表スペースの XDA プリフェッチ要求モニター・エレメント
pool_failed_async_xda_reqs	xs:nonNegativeInteger	pool_failed_async_xda_reqs - 失敗した XDA プリフェッチ要求モニター・エレメント
pool_index_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_index_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール非依存索引ページ・モニター・エレメント

表 175. MON\_GET\_WORKLOAD\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
pool_index_l_reads	xs:nonNegativeInteger	pool_index_l_reads - バッファ・プール索引の論理読み取り
pool_index_p_reads	xs:nonNegativeInteger	pool_index_p_reads - バッファ・プール索引の物理読み取り
pool_index_writes	xs:nonNegativeInteger	pool_index_writes - バッファ・プール索引の書き込み
pool_queued_async_data_pages	xs:nonNegativeInteger	pool_queued_async_data_pages - データ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_data_reqs	xs:nonNegativeInteger	pool_queued_async_data_reqs - データ・プリフェッチ要求モニター・エレメント
pool_queued_async_index_pages	xs:nonNegativeInteger	pool_queued_async_index_pages - 索引ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_index_reqs	xs:nonNegativeInteger	pool_queued_async_index_reqs - 索引プリフェッチ要求モニター・エレメント
pool_queued_async_other_reqs	xs:nonNegativeInteger	pool_queued_async_other_reqs - 非プリフェッチの要求モニター・エレメント
pool_queued_async_temp_data_pages	xs:nonNegativeInteger	pool_queued_async_temp_data_pages - TEMPORARY 表スペースのデータ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_data_reqs	xs:nonNegativeInteger	pool_queued_async_temp_data_reqs - TEMPORARY 表スペースのデータ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_index_pages	xs:nonNegativeInteger	pool_queued_async_temp_index_pages - TEMPORARY 表スペースの索引ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_index_reqs	xs:nonNegativeInteger	pool_queued_async_temp_index_reqs - TEMPORARY 表スペースの索引プリフェッチ要求モニター・エレメント
pool_queued_async_temp_xda_pages	xs:nonNegativeInteger	pool_queued_async_temp_xda_pages - TEMPORARY 表スペースの XDA データ・ページ・プリフェッチ要求モニター・エレメント
pool_queued_async_temp_xda_reqs	xs:nonNegativeInteger	pool_queued_async_temp_xda_reqs - TEMPORARY 表スペースの XDA データ・プリフェッチ要求モニター・エレメント
pool_queued_async_xda_reqs	xs:nonNegativeInteger	pool_queued_async_xda_reqs - XDA プリフェッチ要求モニター・エレメント
pool_read_time	xs:nonNegativeInteger	pool_read_time - バッファ・プール物理読み取り時間の合計
pool_temp_data_l_reads	xs:nonNegativeInteger	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
pool_temp_data_p_reads	xs:nonNegativeInteger	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
pool_temp_index_l_reads	xs:nonNegativeInteger	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
pool_temp_index_p_reads	xs:nonNegativeInteger	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り

表 175. MON\_GET\_WORKLOAD\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
pool_temp_xda_l_reads	xs:nonNegativeInteger	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
pool_temp_xda_p_reads	xs:nonNegativeInteger	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り
pool_write_time	xs:nonNegativeInteger	pool_write_time - バッファ・プール物理書き込み時間の合計
pool_xda_gbp_indep_pages_found_in_lbp	xs:nonNegativeInteger	pool_xda_gbp_indep_pages_found_in_lbp - ローカル・バッファ・プールで検出されたグループ・バッファ・プール XDA 非依存ページ・モニター・エレメント
pool_xda_gbp_invalid_pages	xs:nonNegativeInteger	pool_xda_gbp_invalid_pages - グループ・バッファ・プールの無効な XDA データ・ページ
pool_xda_gbp_l_reads	xs:nonNegativeInteger	pool_xda_gbp_l_reads - グループ・バッファ・プールの XDA データの論理読み取りの要求
pool_xda_gbp_p_reads	xs:nonNegativeInteger	pool_xda_gbp_p_reads - グループ・バッファ・プールの XDA データの物理読み取りの要求
pool_xda_l_reads	xs:nonNegativeInteger	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
pool_xda_lbp_pages_found	xs:nonNegativeInteger	pool_xda_lbp_pages_found - 検出されたローカル・バッファ・プールの XDA データ・ページ
pool_xda_p_reads	xs:nonNegativeInteger	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
pool_xda_writes	xs:nonNegativeInteger	pool_xda_writes - バッファ・プール XDA データの書き込み
post_shrthreshold_sorts	xs:nonNegativeInteger	post_shrthreshold_sorts - ポスト共有しきい値ソート
post_threshold_peas	xs:long	post_threshold_peas - partial early aggregation しきい値
post_threshold_peds	xs:long	post_threshold_peds - partial early distinct しきい値
post_threshold_sorts	xs:nonNegativeInteger	post_threshold_sorts - ポストしきい値ソート
prefetch_wait_time	xs:nonNegativeInteger	prefetch_wait_time - プリフェッチ待ち時間
prefetch_waits	xs:nonNegativeInteger	prefetch_waits - プリフェッチャーの待機カウント・モニター・エレメント
rows_modified	xs:nonNegativeInteger	rows_modified - 変更された行数
rows_read	xs:nonNegativeInteger	rows_read - 読み取り行数
rows_returned	xs:nonNegativeInteger	rows_returned - 戻り行数
rqsts_completed_total	xs:nonNegativeInteger	rqsts_completed_total - 完了した要求の合計数
sort_overflows	xs:nonNegativeInteger	sort_overflows - ソート・オーバーフロー
tcpip_rcv_volume	xs:nonNegativeInteger	tcpip_rcv_volume - TCP/IP 受信ボリューム
tcpip_rcv_wait_time	xs:nonNegativeInteger	tcpip_rcv_wait_time - TCP/IP 受信待ち時間
tcpip_recvs_total	xs:nonNegativeInteger	tcpip_recvs_total - TCP/IP 合計受信数
tcpip_send_volume	xs:nonNegativeInteger	tcpip_send_volume - TCP/IP 送信ボリューム
tcpip_send_wait_time	xs:nonNegativeInteger	tcpip_send_wait_time - TCP/IP 送信待ち時間
tcpip_sends_total	xs:nonNegativeInteger	tcpip_sends_total - TCP/IP 合計送信数
thresh_violations	xs:nonNegativeInteger	thresh_violations - しきい値違反の回数



表 175. MON\_GET\_WORKLOAD\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
total_act_time	xs:nonNegativeInteger	total_act_time - 合計アクティビティー時間
total_act_wait_time	xs:nonNegativeInteger	total_act_wait_time - 合計アクティビティー待機時間
total_app_commits	xs:nonNegativeInteger	total_app_commits - アプリケーション・コミットの合計回数
total_app_rollback	xs:nonNegativeInteger	total_app_rollback - アプリケーション・ロールバックの合計回数
total_app_rqst_time	xs:nonNegativeInteger	total_app_rqst_time - アプリケーション要求合計時間
total_app_section_executions	xs:nonNegativeInteger	total_app_section_executions - セクション実行の合計回数
total_commit_proc_time	xs:nonNegativeInteger	total_commit_proc_time - コミット処理時間の合計
total_commit_time	xs:nonNegativeInteger	total_commit_time - コミット時間の合計
total_compilations	xs:nonNegativeInteger	total_compilations - コンパイルの合計回数
total_compile_proc_time	xs:nonNegativeInteger	total_compile_proc_time - コンパイル処理時間の合計
total_compile_time	xs:nonNegativeInteger	total_compile_time - コンパイル時間の合計
total_connect_authentication_proc_time	xs:nonNegativeInteger	total_connect_authentication_proc_time - 接続認証処理の合計時間
total_connect_authentication_time	xs:nonNegativeInteger	total_connect_authentication_time - 接続またはユーザー切り替えの認証要求の合計時間
total_connect_authentications	xs:nonNegativeInteger	total_connect_authentications - 接続またはユーザー切り替えの認証の実行回数
total_connect_request_proc_time	xs:nonNegativeInteger	total_connect_request_proc_time - 接続要求またはユーザー切り替え要求の合計処理時間
total_connect_request_time	xs:nonNegativeInteger	total_connect_request_time - 接続またはユーザー切り替えの要求の合計時間
total_connect_requests	xs:nonNegativeInteger	total_connect_requests - 接続要求またはユーザー切り替え要求
total_cpu_time	xs:nonNegativeInteger	total_cpu_time - 合計 CPU 時間
total_disp_run_queue_time	xs:long	total_disp_run_queue_time - ディスパッチャー実行キュー時間の合計
total_extended_latch_wait_time	xs:nonNegativeInteger	total_extended_latch_wait_time - 拡張ラッチの合計待機時間
total_extended_latch_waits	xs:nonNegativeInteger	total_extended_latch_waits - 拡張ラッチの合計待機回数
total_implicit_compilations	xs:nonNegativeInteger	total_implicit_compilations - 暗黙的コンパイルの合計回数
total_implicit_compile_proc_time	xs:nonNegativeInteger	total_implicit_compile_proc_time - 暗黙的コンパイルの処理時間の合計
total_implicit_compile_time	xs:nonNegativeInteger	total_implicit_compile_time - 暗黙的コンパイル時間の合計
total_load_proc_time	xs:nonNegativeInteger	total_load_proc_time - ロード処理時間の合計
total_load_time	xs:nonNegativeInteger	total_load_time - ロード時間の合計
total_loads	xs:nonNegativeInteger	total_loads - ロードの合計回数
total_peas	xs:long	total_peas - partial early aggregation の合計
total_peds	xs:long	total_peds - partial early distinct の合計
total_reorg_proc_time	xs:nonNegativeInteger	total_reorg_proc_time - 再編成の処理時間の合計

表 175. MON\_GET\_WORKLOAD\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
total_reorg_time	xs:nonNegativeInteger	total_reorg_time - 再編成時間の合計
total_reorgs	xs:nonNegativeInteger	total_reorgs - 再編成の合計回数
total_rollback_proc_time	xs:nonNegativeInteger	total_rollback_proc_time - ロールバック処理時間の合計
total_rollback_time	xs:nonNegativeInteger	total_rollback_time - ロールバック時間の合計
total_routine_invocations	xs:nonNegativeInteger	total_routine_invocations - ルーチン呼び出しの合計回数
total_routine_time	xs:nonNegativeInteger	total_routine_time - ルーチン時間の合計
total_routine_user_code_proc_time	xs:nonNegativeInteger	total_routine_user_code_proc_time - ルーチン・ユーザー・コード処理時間の合計
total_routine_user_code_time	xs:nonNegativeInteger	total_routine_user_code_time - ルーチン・ユーザー・コード時間の合計
total_rqst_time	xs:nonNegativeInteger	total_rqst_time - 合計要求時間
total_runstats	xs:nonNegativeInteger	total_runstats - ランタイム統計の合計回数
total_runstats_proc_time	xs:nonNegativeInteger	total_runstats_proc_time - ランタイム統計の処理時間の合計
total_runstats_time	xs:nonNegativeInteger	total_runstats_time - ランタイム統計時間の合計
total_section_proc_time	xs:nonNegativeInteger	total_section_proc_time - セクション処理時間の合計
total_section_sort_proc_time	xs:nonNegativeInteger	total_section_sort_proc_time - セクション・ソート処理時間合計
total_section_sort_time	xs:nonNegativeInteger	total_section_sort_time - セクション・ソート時間合計
total_section_sorts	xs:nonNegativeInteger	total_section_sorts - セクション・ソート合計
total_section_time	xs:nonNegativeInteger	total_section_time - セクション時間の合計
total_sorts	xs:nonNegativeInteger	total_sorts - ソート合計
total_stats_fabrication_proc_time	xs:nonNegativeInteger	total_stats_fabrication_proc_time - 統計作成の合計処理時間
total_stats_fabrication_time	xs:nonNegativeInteger	total_stats_fabrication_time - 統計作成の合計時間
total_stats_fabrications	xs:nonNegativeInteger	total_stats_fabrications - 統計作成の合計
total_sync_runstats	xs:nonNegativeInteger	total_sync_runstats - 同期 RUNSTATS アクティビティの合計
total_sync_runstats_proc_time	xs:nonNegativeInteger	total_sync_runstats_proc_time - 同期 RUNSTATS 処理時間の合計
total_sync_runstats_time	xs:nonNegativeInteger	total_sync_runstats_time - 同期 RUNSTATS 時間の合計
total_wait_time	xs:nonNegativeInteger	total_wait_time - 合計待ち時間
tq_sort_heap_rejections	xs:long	tq_sort_heap_rejections - 表キューのソート・ヒープの拒否
tq_sort_heap_requests	xs:long	tq_sort_heap_requests - 表キューのソート・ヒープの要求
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
wlm_queue_assignments_total	xs:nonNegativeInteger	wlm_queue_assignments_total - ワークロード・マネージャー合計キュー割り当て数
wlm_queue_time_total	xs:nonNegativeInteger	wlm_queue_time_total - ワークロード・マネージャー合計キュー時間
workload_id	xs:nonNegativeInteger	workload_id - ワークロード ID

表 175. MON\_GET\_WORKLOAD\_DETAILS について戻される詳細メトリック (続き)

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
workload_name	xs:string (128)	workload_name - ワークロード名

## MON\_INCREMENT\_INTERVAL\_ID プロシージャ - モニター・インターバルのインクリメント

MON\_INCREMENT\_INTERVAL\_ID プロシージャは、モニター・インターバルを 1 つ増やし、新しい値を出力引数で戻します。現行モニター・インターバルは MON\_INTERVAL\_ID データベース・グローバル変数で示されます。

### 構文

```
►►—MON_INTERVAL_INCREMENT_ID—(—new_id—)—————►►
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

*new\_id*

新しいモニター・インターバルを戻す、タイプ BIGINT の出力引数。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

なし

### 使用上の注意

自律型プロシージャである MON\_INCREMENT\_INTERVAL\_ID は、独自のトランザクション・スコープ内で実行されます。

MON\_INTERVAL\_ID の値が最大 BIGINT 値に達すると、循環して 1 に戻ります。

### 例

インターバル ID をインクリメントし、新しい値を引数で戻します。この例で、MY\_ID は既に作成されているセッション・グローバル変数であることを前提としています。

```
CALL SYSPROC.MON_INCREMENT_INTERVAL_ID(MY_ID)
```

## MON\_LOCKWAITS 管理ビュー - ロックの取得を待機しているアプリケーションに関するメトリックの取得

MON\_LOCKWAITS 管理ビューは、現在接続しているデータベースでのロック取得を待機しているアプリケーションのために機能するエージェントについての情報を戻します。この照会はロックの問題を識別するうえで役立ちます。

この管理ビューは SNAPLOCKWAIT 管理ビューに置き換わるものです。SNAPLOCKWAIT は DB2 バージョン 9.7 フィックスパック 1 で非推奨になり、将来のリリースでは廃止される可能性があります。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- MON\_LOCKWAITS 管理ビューに対する SELECT 特権
- MON\_LOCKWAITS 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

### デフォルトの PUBLIC 特権

なし

### 戻される情報

表 176. MON\_LOCKWAITS 管理ビューによって戻される情報

列名	データ・タイプ	説明またはモニター・エレメント
LOCK_NAME	VARCHAR(32)	lock_name - ロック名MON_FORMAT_LOCK_NAME ルーチンを使用すると、この内部バイナリー・ロック名のフォーマットを設定したり、ロックに関する詳細情報 (例えば表ロックが参照する表と表スペースの情報) を取得したりできます。
LOCK_OBJECT_TYPE	VARCHAR(32)	lock_object_type - 待機中のロック対象タイプ
LOCK_WAIT_ELAPSED_TIME	INTEGER	エージェントがロックの取得を待機し始めてから経過した時間。この値は秒単位です。
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名 表を参照しないロックの場合、NULL が戻されます。
TABNAME	VARCHAR(128)	table_name - 表名 表を参照しないロックの場合、NULL が戻されます。

表 176. MON\_LOCKWAITS 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID  このエレメントは、パーティション表およびパーティション化索引にのみ該当します。ロック・レベル情報が戻されると、値 -1 は表全体へのアクセスを制御するロックを表します。
LOCK_MODE	VARCHAR(10)	lock_mode - ロック・モード
LOCK_CURRENT_MODE	VARCHAR(10)	lock_current_mode - 移行前の元のロック・モード  LOCK_STATUS が "C" (移行) でない場合、NULL 値が戻されます。
LOCK_MODE_REQUESTED	VARCHAR(10)	lock_mode_requested - 要求されているロック・モード
REQ_APPLICATION_HANDLE	BIGINT	req_application_handle - 要求元のアプリケーション・ハンドル
REQ_AGENT_TID	BIGINT	req_agent_tid - 要求元のエージェント TID
REQ_MEMBER	SMALLINT	req_member - 要求元のメンバー
REQ_APPLICATION_NAME	VARCHAR(128)	このロックの取得を待機しているクライアントで実行中のアプリケーションの名前。
REQ_USERID	VARCHAR(128)	このロックの取得を待機しているアプリケーションによって使用されているセッションの現在の許可 ID。
REQ_STMT_TEXT	CLOB(2MB)	ロックの取得を待機しているアプリケーションで実行されている SQL ステートメント・セクション。  非 SQL アクティビティーの場合、長さ 0 のストリング値が戻されます。
HLD_APPLICATION_HANDLE	BIGINT	hld_application_handle - 保持しているアプリケーション・ハンドル  このロックを保持しているアプリケーションが不明または見つからない場合には、NULL 値が戻されます。
HLD_MEMBER	SMALLINT	hld_member - 保持しているメンバー
HLD_APPLICATION_NAME	VARCHAR(128)	このロックを保持しているクライアントで実行中のアプリケーションの名前。  このロックを保持しているアプリケーションが不明または見つからない場合には、長さ 0 のストリング値が戻されます。
HLD_USERID	VARCHAR(128)	このロックを保持しているアプリケーションによって使われているセッションの現在の許可 ID。
HLD_CURRENT_STMT_TEXT	CLOB(2MB)	ロックを保持しているアプリケーションに現在関連付けられている SQL ステートメント・テキスト。これは、ロックの原因となっているステートメントとは限らないことに注意してください。

## MON\_PKG\_CACHE\_SUMMARY - データベース・パッケージ・キャッシュの概要情報の取得

MON\_PKG\_CACHE\_SUMMARY 管理ビューは、キャッシュ内の静的 SQL ステートメントと動的 SQL ステートメントの両方に関する主要なメトリックを戻し、データベース・パッケージ・キャッシュの概要情報を提供します。戻されるメトリックは、データベースの全メンバーのすべてのステートメント実行にわたって集計されます。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- MON\_PKG\_CACHE\_SUMMARY 管理ビューに対する SELECT 特権
- MON\_PKG\_CACHE\_SUMMARY 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

### デフォルトの PUBLIC 特権

なし

### 戻される情報

表 177. MON\_PKG\_CACHE\_SUMMARY 管理ビューによって戻される情報

列名	データ・タイプ	説明またはモニター・エレメント
SECTION_TYPE	CHAR(1)	section_type - セクション・タイプ標識
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	executable_id - 実行可能 ID
NUM_COORD_EXEC	BIGINT	num_coord_exec - コーディネーター・エージェントによる実行数
NUM_COORD_EXEC_WITH_METRICS	BIGINT	num_coord_exec_with_metrics - メトリックを伴うコーディネーター・エージェントによる実行数
TOTAL_STMT_EXEC_TIME	BIGINT	メトリックの収集対象となるすべてのステートメント実行にわたって、(ネストされたアクティビティーを含む) そのステートメントの実行に費やされた時間の合計 (ミリ秒)。
AVG_STMT_EXEC_TIME	BIGINT	メトリックの収集対象となるすべてのステートメント実行にわたって、(ネストされたアクティビティーを含む) そのステートメントの実行に費やされた時間の平均 (ミリ秒)。

表 177. MON\_PKG\_CACHE\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
TOTAL_CPU_TIME	BIGINT	DB2 データベース・マネージャー内で費やされた CPU 時間の合計 (マイクロ秒)。この値は、ユーザーとシステムの両方の CPU 時間の合計を表します。これは、そのステートメントのすべての total_cpu_time - 合計 CPU 時間値の累積として計算されます。
AVG_CPU_TIME	BIGINT	メトリックの収集対象となるすべてのステートメント実行にわたって、DB2 データベース・マネージャー内で費やされた CPU 時間の平均 (マイクロ秒)。
TOTAL_LOCK_WAIT_TIME	BIGINT	ロックの待機に費やされた経過時間の合計 (ミリ秒)。この値は、そのステートメントに関するすべての lock_wait_time - ロック待機中の時間値の合計として計算されます。
AVG_LOCK_WAIT_TIME	BIGINT	メトリックの収集対象となるすべてのステートメント実行にわたって、ロックの待機に費やされた経過時間の平均 (ミリ秒)。
TOTAL_IO_WAIT_TIME	BIGINT	入出力操作に費やされた経過時間の合計 (ミリ秒)。この値は、直接読み取りまたは直接書き込みの実行に必要とされた経過時間と、表スペース・コンテナーとの間のデータおよび索引ページの物理的読み取り/書き込みに費やされた経過時間の合計として計算されます。
AVG_IO_WAIT_TIME	BIGINT	メトリックの収集対象となるすべてのステートメント実行にわたって、入出力操作に費やされた経過時間の平均 (ミリ秒)。
PREP_TIME	BIGINT	prep_time - 準備時間
ROWS_READ_PER_ROWS_RETURNED	BIGINT	メトリックの収集対象となるすべてのステートメント実行にわたって、戻された各行に対する読み取られた行数の平均。
AVG_ACT_WAIT_TIME	BIGINT	データベース・アクティビティーの待機に費やされたステートメント実行当たり平均時間。
AVG_LOCK_ESCALS	BIGINT	ステートメント実行当たりのロック・エスカレーションの平均回数。

表 177. MON\_PKG\_CACHE\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
AVG_RECLAIM_WAIT_TIME	BIGINT	ページ再利用の待機に費やされたステートメント実行当たり平均時間。この値は、DB2 pureScale 環境の外では常に NULL です。
AVG_SPACEMAPPAGE_RECLAIM_WAIT_TIME	BIGINT	スペース・マップ・ページ再利用の待機に費やされたステートメント実行当たり平均時間。この値は、DB2 pureScale 環境の外では常に NULL です。
STMT_TEXT	CLOB(2MB)	stmt_text - SQL 動的ステートメント・テキスト

## MON\_SAMPLE\_SERVICE\_CLASS\_METRICS - サービス・クラス・メトリックのサンプルの取得

MON\_SAMPLE\_SERVICE\_CLASS\_METRICS 表関数は、1 つ以上のデータベースにおいて、1 つ以上のサービス・クラスのシステム・メトリックを 2 つの時点 (この関数が呼び出された時点と、指定された時間の経過後) で読み取ります。これらのメトリックからさまざまな統計を計算します。

### 構文

```

▶▶ MON_SAMPLE_SERVICE_CLASS_METRICS (—hostname—, —db_name—, —
▶—service_superclass_name—, —service_subclass_name—, —
▶—sample_time—, —member—)

```

スキーマは SYSPROC です。

### 表関数パラメーター

#### hostname

レポートの生成元となるノードの完全修飾ホスト名または IP アドレスを指定する、タイプ VARCHAR(255) の入力引数。引数が NULL または空ストリングである場合、インスタンス内のすべてのアクティブ・データベースからメトリックが取得されます。

#### db\_name

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。データベースのディレクトリー項目タイプは、(LIST DATABASE DIRECTORY コマンドによって戻される) 「Indirect」または「Home」のどちらかでなければなりません。引数が NULL または空ストリングである場合、インスタンス内のすべてのデータベースからメトリックが取得されます。

#### service\_superclass\_name

この関数を呼び出すときに現在接続されているデータベースで有効なサービス・スーパークラス名を指定する、タイプ VARCHAR(128) の入力引数。引数が



NULL または空ストリングである場合、db\_name パラメーターにより課される制約を満たすデータベース名を持つインスタンス内のすべてのスーパークラスについて、パフォーマンス・メトリックが取得されます。

#### *service\_subclass\_name*

この関数を呼び出すときに現在接続されているデータベースで有効なサービス・サブクラス名を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、db\_name パラメーターと service\_superclass\_name パラメーターにより課される制約をそれぞれ満たすデータベース名とサービス・スーパークラス名を持つインスタンス内のすべてのサブクラスについて、パフォーマンス・メトリックが取得されます。

#### *sample\_time*

データに対して結果を計算する前に、関数とそのデータを収集する時間。この値の尺度は秒単位であり、1 以上でなければなりません。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例 1

すべてのパーティションにおける各サービス・サブクラスの 30 秒間のアクティビティ・スループットと CPU 速度を示します。

```
SELECT varchar(service_superclass_name,30) AS service_superclass,
       varchar(service_subclass_name,30) AS service_subclass,
       decimal(sum(act_throughput),10,2) AS act_throughput,
       decimal(sum(total_cpu_time) /
              (sum(total_cpu_time) +
               sum(total_disp_run_queue_time)),3,2) AS cpu_velocity
FROM TABLE(MON_SAMPLE_SERVICE_CLASS_METRICS
            (null, current server, '', '', 30, -2)) AS t
WHERE service_subclass_name IS NOT NULL
GROUP BY service_superclass_name, service_subclass_name
ORDER BY service_superclass_name, service_subclass_name
```

以下はこの照会の出力例です。

```
SERVICE_SUPERCLASS      SERVICE_SUBCLASS      ...
-----
SYSDEFAULTUSERCLASS     SYSDEFAULTSUBCLASS    ...
```

```

SYSDEFAULTMAINTENANCECLASS SYSDEFAULTSUBCLASS ...
SYSDEFAULTSYSTEMCLASS     SYSDEFAULTSUBCLASS ...

```

3 record(s) selected.

照会の出力 (続き)。

```

... ACT_THROUGHPUT CPU_VELOCITY
... -----
...          214.76          0.72
...           0             0
...           0             0

```

## 例 2

システム上のすべてのデータベースとパーティションにおける各サービス・クラスの 5 分間の構成済み共有、見積 CPU 割り当て率、および実際の CPU 使用率を示します。

```

SELECT varchar(db_name,18) AS db_name,
       varchar(service_superclass_name,30) AS service_superclass,
       varchar(service_subclass_name,30) AS service_subclass,
       cpu_shares,
       cpu_limit,
       decimal(estimated_cpu_entitlement, 9, 2) AS estimated_cpu_entitlement,
       decimal(cpu_utilization, 3, 2) AS cpu_utilization
FROM TABLE(MON_SAMPLE_SERVICE_CLASS_METRICS
            (null, null, '', '', 300, -2)) AS t
ORDER BY db_name, service_superclass_name, service_subclass_name, member

```

以下はこの照会の出力例です。

```

DB_NAME          SERVICE_SUPERCLASS      ...
-----
SAMPLE           SYSDEFAULTUSERCLASS     ...
SAMPLE           SYSDEFAULTUSERCLASS     ...
SAMPLE           SYSDEFAULTMAINTENANCECLASS ...
SAMPLE           SYSDEFAULTMAINTENANCECLASS ...
SAMPLE           SYSDEFAULTSYSTEMCLASS   ...
SAMPLE           SYSDEFAULTSYSTEMCLASS   ...
OTHER            SYSDEFAULTUSERCLASS     ...
OTHER            SYSDEFAULTUSERCLASS     ...
OTHER            SYSDEFAULTMAINTENANCECLASS ...
OTHER            SYSDEFAULTMAINTENANCECLASS ...
OTHER            SYSDEFAULTSYSTEMCLASS   ...
OTHER            SYSDEFAULTSYSTEMCLASS   ...

```

12 record(s) selected.

照会の出力 (続き)。

```

SERVICE_SUBCLASS CPU_SHARES CPU_LIMIT ...
-----
SYSDEFAULTSUBCLASS 1000      - ...
-                  2000      70 ...
SYSDEFAULTSUBCLASS 1000      - ...
-                  1000      - ...
SYSDEFAULTSUBCLASS -          - ...
-                  -          - ...
SYSDEFAULTSUBCLASS 1000      - ...
-                  5000      70 ...
SYSDEFAULTSUBCLASS 1000      - ...
-                  2000      - ...
SYSDEFAULTSUBCLASS -          - ...
-                  -          - ...

```

照会の出力 (続き)。

```

ESTIMATED_CPU_ENTITLEMENT CPU_UTILIZATION
-----
0.20 0.16
0.20 0.16
0.10 0.09
0.10 0.09
- 0.10
- 0.10
0.50 0.45
0.50 0.45
0.20 0.11
0.20 0.11
- 0.09
- 0.09

```

### 使用上の注意

MON\_SAMPLE\_SERVICE\_CLASS\_METRICS 表関数は、サービス・クラスごとおよび各データベースのメンバーごとに、1 行のデータを戻します。この表関数は、(1 つのメンバー上の) 複数のサービス・クラスにわたる集約や、(1 つ以上のサービス・クラスの) 複数メンバーにわたる集約を行いません。しかし、SQL 照会を使用して集約を実行できます。入力パラメーターは、一緒に「ANDing される」こととなります。したがって、矛盾する複数の入力パラメーター (例えば SUPA というスーパークラスと、SUPA のサブクラスではない SUBB というサブクラス) を指定すると、行は戻されません。

### 戻される情報

表 178. MON\_SAMPLE\_SERVICE\_CLASS\_METRICS で戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
HOSTNAME	VARCHAR(255)	hostname - ホスト名 : モニター・エレメント
DB_NAME	VARCHAR(128)	db_name - データベース名 : モニター・エレメント
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - サービス・スーパークラス名
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - サービス・サブクラス名
SERVICE_CLASS_ID	INTEGER	service_class_id - サービス・クラス ID
MEMBER	SMALLINT	member - データベース・メンバー
UOW_THROUGHPUT	DOUBLE	uow_throughput - 作業単位スループット : モニター・エレメント
UOW_LIFETIME_AVG	DOUBLE	uow_lifetime_avg - 作業単位の平均存続期間 : モニター・エレメント

表 178. MON\_SAMPLE\_SERVICE\_CLASS\_METRICS で戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
UOW_COMPLETED_TOTAL	BIGINT	uow_completed_total - 完了済みの合計作業単位 : モニター・エレメント
ACT_THROUGHPUT	DOUBLE	act_throughput - アクティビティ・スループット : モニター・エレメント
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 完了したアクティビティの合計数
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間 : モニター・エレメント
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - ディスパッチャーの合計実行キュー時間 : モニター・エレメント
CPU_SHARES	INTEGER	cpu_shares - WLM ディスパッチャー CPU シェア : モニター・エレメント
CPU_SHARE_TYPE	VARCHAR(4)	cpu_share_type - WLM ディスパッチャー CPU シェア・タイプのモニター・エレメント
CPU_LIMIT	SMALLINT	cpu_limit - WLM ディスパッチャーの CPU リミット : モニター・エレメント
ESTIMATED_CPU_ENTITLEMENT	DOUBLE	estimated_cpu_entitlement - 見積もりの CPU 割り当て率のモニター・エレメント
CPU_UTILIZATION	DOUBLE	cpu_utilization - CPU 使用率 : モニター・エレメント
CPU_VELOCITY	DOUBLE	cpu_velocity - CPU 速度モニター・エレメント

## MON\_SAMPLE\_WORKLOAD\_METRICS - サンプルの取得

MON\_SAMPLE\_WORKLOAD\_METRICS 表関数は、1 つ以上のデータベースにおいて、1 つ以上のワークロードのシステム・メトリックを 2 つの時点 (この関数が呼び出された時点と、指定された時間の経過後) で読み取ります。これらのメトリックからさまざまな統計を計算します。

### 構文

```

▶▶ MON_SAMPLE_SERVICE_CLASS_METRICS (—hostname—, —db_name—, —————▶
▶ workload_name—, —sample_time—, —member—) —————▶▶

```

スキーマは SYSPROC です。

## 表関数パラメーター

### *hostname*

レポートの生成元となるノードの完全修飾ホスト名または IP アドレスを指定する、タイプ VARCHAR(255) の入力引数。引数が NULL または空ストリングである場合、インスタンス内のすべてのノードからメトリックが取得されます。

### *db\_name*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。データベースのディレクトリー項目タイプは、(LIST DATABASE DIRECTORY コマンドによって戻される) 「Indirect」 または 「Home」 のどちらかでなければなりません。引数が NULL または空ストリングである場合、インスタンス内のすべてのアクティブ・データベースからメトリックが取得されます。

### *workload\_name*

メトリックが戻される特定のワークロードを指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、db\_name パラメーターにより課される制約を満たすデータベース名を持つインスタンス内のすべてのワークロードについて、メトリックが戻されます。

### *sample\_time*

データに対して結果を計算する前に、関数とそのデータを収集する時間。この値の尺度は秒単位であり、1 以上でなければなりません。

### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバーを指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

すべてのパーティションにおける各ワークロードの 30 秒間の作業単位 (UOW) スループット、アクティビティー・スループット、および平均 CPU 使用率を表示します。

```
SELECT varchar(workload_name,30) AS workload_name,  
       decimal(sum(uow_throughput),10,2) AS uow_throughput,  
       decimal(sum(act_throughput),10,2) AS act_throughput,
```

```

        decimal(sum(cpu_utilization),3,2) AS cpu_utilization
FROM TABLE(MON_SAMPLE_WORKLOAD_METRICS
            (null, current server, '',30 ,-2)) AS t
GROUP BY workload_name
ORDER BY workload_name

```

以下はこの照会の出力例です。

```

WORKLOAD_NAME          UOW_THROUGHPUT          ...
-----
SYSDEFAULTUSERWORKLOAD          124.43 ...
SYSDEFAULTADMWORKLOAD          0 ...

2 record(s) selected.

```

照会の出力 (続き)。

```

... ACT_THROUGHPUT  CPU_UTILIZATION
... -----
...          214.76          0.89
...          0          0

```

## 使用上の注意

MON\_SAMPLE\_WORKLOAD\_METRICS 表関数は、ワークロードごと、および各データベースのメンバーごとに 1 行のデータを戻します。この表関数は、(1 つのメンバー上の) 複数のワークロードにわたる集約や、(1 つ以上のワークロードの) 複数メンバーにわたる集約を行いません。しかし、集約は SQL 照会を使用して実行できます。

## 戻される情報

表 179. MON\_SAMPLE\_WORKLOAD\_METRICS から戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
HOSTNAME	VARCHAR(255)	hostname - ホスト名 : モニター・エレメント
DB_NAME	VARCHAR(128)	db_name - データベース名 : モニター・エレメント
WORKLOAD_NAME	VARCHAR(128)	workload_name - ワークロード名 : モニター・エレメント
WORKLOAD_ID	INTEGER	workload_id - ワークロード ID : モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー
UOW_THROUGHPUT	DOUBLE	uow_throughput - 作業単位スループット : モニター・エレメント
UOW_LIFETIME_AVG	DOUBLE	uow_lifetime_avg - 作業単位の平均存続期間 : モニター・エレメント
UOW_COMPLETED_TOTAL	BIGINT	uow_completed_total - 完了済みの合計作業単位 : モニター・エレメント
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間 : モニター・エレメント

表 179. MON\_SAMPLE\_WORKLOAD\_METRICS から戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - ディスパッチャーの合計実行キュー時間 : モニター・エレメント
ACT_THROUGHPUT	DOUBLE	act_throughput - アクティビティー・スループット : モニター・エレメント
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 完了したアクティビティーの合計数
CPU_UTILIZATION	DOUBLE	cpu_utilization - CPU 使用率 : モニター・エレメント
CPU_VELOCITY	DOUBLE	cpu_velocity - CPU 速度モニター・エレメント

## MON\_SERVICE\_SUBCLASS\_SUMMARY - すべてのサービス・サブクラスに関するメトリックの取得

MON\_SERVICE\_SUBCLASS\_SUMMARY 管理ビューは、現在接続されているデータベース内のすべてのサービス・サブクラスに関する主要なメトリックを戻します。これはサービス・クラスごとに実行された処理を示すため、システムの概要をモニターするのに役立ちます。

戻されるメトリックは、データベースの全メンバーにおいて、指定されたサービス・サブクラスのもとで実行された要求に関するすべてのメトリックの累計を表します。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- MON\_SERVICE\_SUBCLASS\_SUMMARY 管理ビューに対する SELECT 特権
- MON\_SERVICE\_SUBCLASS\_SUMMARY 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

### デフォルトの PUBLIC 特権

なし

### 戻される情報

表 180. MON\_SERVICE\_SUBCLASS\_SUMMARY 管理ビューによって戻される情報

列名	データ・タイプ	説明またはモニター・エレメント
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - サービス・スーパークラス名
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - サービス・サブクラス名

表 180. MON\_SERVICE\_SUBCLASS\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
SERVICE_CLASS_ID	INTEGER	service_class_id - サービス・クラス ID
TOTAL_APP_COMMITS	BIGINT	total_app_commits - アプリケーション・コミットの合計回数モニター・エレメント
TOTAL_APP_ROLLBACKS	BIGINT	total_app_rollback - アプリケーション・ロールバックの合計回数モニター・エレメント
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 完了したアクティビティーの合計数モニター・エレメント
APP_RQSTS_COMPLETED_TOTAL	BIGINT	app_rqsts_completed_total - 完了したアプリケーション要求の合計数モニター・エレメント
AVG_RQST_CPU_TIME	BIGINT	正常に完了したすべての外部要求によって費やされた平均 CPU 時間 (マイクロ秒)。これは、ユーザーとシステムの両方の CPU 時間の合計を表します。
ROUTINE_TIME_RQST_PERCENT	DECIMAL(5,2)	データベース・サーバーが要求を処理するのに費やした時間のうち、ユーザー・ルーチンの実行に費やした時間のパーセンテージ。
RQST_WAIT_TIME_PERCENT	DECIMAL(5,2)	要求を処理するのに費やされた時間のうち、DB2 データベース・サーバー内部での待機に費やされた時間のパーセンテージ。
ACT_WAIT_TIME_PERCENT	DECIMAL(5,2)	アクティビティーの実行に費やされた時間のうち、DB2 データベース・サーバー内部での待機に費やされた時間のパーセンテージ。
IO_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、入出力操作に費やされた時間のパーセンテージ。これには、直接読み取り/直接書き込みの実行に費やされた時間に加えて、表スペースからバッファ・プールへのデータおよび索引ページの読み取りや元のディスクへの書き込みに費やされた時間が含まれます。
LOCK_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、ロック待機に費やされた時間のパーセンテージ。



表 180. MON\_SERVICE\_SUBCLASS\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
AGENT_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、コンセントレーター構成のもとでエージェントを待機するためにキューに入れられたアプリケーションによって費やされた時間のパーセンテージ。
NETWORK_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、クライアント/サーバー通信のために費やされた時間のパーセンテージ。これには、TCP/IP または IPC プロトコルを使ってデータを送受信するのに費やされた時間が含まれます。
SECTION_PROC_TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、セクションの実行に費やした時間のパーセンテージ。これには、ソートの実行に費やされた時間が含まれます。
SECTION_SORT_PROC_TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、セクションの実行中にソートの実行に費やした時間のパーセンテージ。
COMPILE_PROC_TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、SQL ステートメントのコンパイルに費やした時間のパーセンテージ。これには、明示的および暗黙的コンパイル時間が含まれます。
TRANSACT_END_PROC_TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、コミット処理の実行またはトランザクションのロールバックに費やした時間のパーセンテージ。
UTILS_PROC_TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、ユーティリティーの実行に費やした時間のパーセンテージ。これには、 <b>runstats</b> 、再編成、およびロード操作の実行が含まれます。
AVG_LOCK_WAITS_PER_ACT	BIGINT	各コーディネーター・アクティビティ (成功およびアボート) に対して、アプリケーションまたは接続がロックを待機した平均回数。

表 180. MON\_SERVICE\_SUBCLASS\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
AVG_LOCK_TIMEOUTS_PER_ACT	BIGINT	各コーディネーター・アクティビティ (成功およびアボート) に対して、オブジェクトのロック要求がタイムアウトになった平均回数。
AVG_DEADLOCKS_PER_ACT	BIGINT	コーディネーター・アクティビティ (成功およびアボート) ごとのデッドロックの平均数。
AVG_LOCK_ESCALS_PER_ACT	BIGINT	各コーディネーター・アクティビティ (成功およびアボート) に対して、いくつかの行ロックから表ロックにロックがエスカレートした平均回数。
ROWS_READ_PER_ROWS_RETURNED	BIGINT	アプリケーションに戻された各行に対して、表から読み取られた平均行数。
TOTAL_BP_HIT_RATIO_PERCENT	DECIMAL(5,2)	XML ストレージ・オブジェクト (XDA) の要求を含めて、データまたは索引ページの要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要のなかった時間のパーセンテージ。DB2 pureScale 環境では、この値はローカル・バッファ・プールの合計ヒット率を表します。
TOTAL_GBP_HIT_RATIO_PERCENT	DECIMAL(5,2)	DB2 pureScale 環境では、ページがグループ・バッファ・プールに置かれていたため、データ・ページ要求、索引ページ要求、または XML ストレージ・オブジェクト (XDA) ページ要求を処理するためにデータベース・マネージャーがディスクからローカル・バッファ・プールにページをロードする必要がなかった時間のパーセンテージ。この値は、DB2 pureScale 環境の外では常に NULL です。
CF_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 pureScale 環境では、キャッシング・ファシリティ通信の待機に費やされた合計待機時間のパーセンテージ。この値は、DB2 pureScale 環境の外では常に NULL です。

表 180. MON\_SERVICE\_SUBCLASS\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
RECLAIM_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 pureScale 環境 では、ページ再利用の待機に費やされた合計待機時間のパーセンテージ。この値は、DB2 pureScale 環境の外では常に NULL です。
SPACEMAPPAGE_RECLAIM_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 pureScale 環境 では、スペース・マップ・ページ再利用の待機に費やされた合計待機時間のパーセンテージ。この値は、DB2 pureScale 環境の外では常に NULL です。

## MON\_TBSP\_UTILIZATION - すべての表スペースとすべてのデータベース・パーティションに関するモニタリング・メトリックの取得

MON\_TBSP\_UTILIZATION 管理ビューは、現在接続されているデータベース内のすべての表スペースとすべてのデータベース・パーティションに関する、ヒット率や使用率パーセンテージなどの主要なモニタリング・メトリックを戻します。これにより、パフォーマンスとスペース使用状況をモニターするうえで重要な情報が提供されます。この管理ビューは、TBSP\_UTILIZATION 管理ビューに置き換わるものです。

### 許可

スキーマは SYSIBMADM です。

以下のいずれかの権限が必要です。

- MON\_TBSP\_UTILIZATION 管理ビューに対する SELECT 特権
- MON\_TBSP\_UTILIZATION 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

### デフォルトの PUBLIC 特権

なし

### 戻される情報

表 181. MON\_TBSP\_UTILIZATION 管理ビューによって戻される情報

列名	データ・タイプ	説明またはモニター・エレメント
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
MEMBER	SMALLINT	member - データベース・メンバー

表 181. MON\_TBSP\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
TBSP_TYPE	VARCHAR(10)	<p>tablespace_type - 表スペース・タイプ。このインターフェースは、sqlutil.h の定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• DMS</li> <li>• SMS</li> </ul>
TBSP_CONTENT_TYPE	VARCHAR(10)	<p>tablespace_content_type - 表スペースのコンテンツ・タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• ANY</li> <li>• LARGE</li> <li>• SYSTEMP</li> <li>• USRTEMP</li> </ul>
TBSP_STATE	VARCHAR(256)	tablespace_state - 表スペースの状態
TBSP_PAGE_SIZE	BIGINT	tablespace_page_size - 表スペースのページ・サイズ
TBSP_EXTENT_SIZE	BIGINT	tablespace_extent_size - 表スペースのエクステント・サイズ
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 表スペースのプリフェッチ・サイズ
TBSP_USING_AUTO_STORAGE	SMALLINT	tablespace_using_auto_storage - 自動ストレージが使用可能な表スペース
TBSP_AUTO_RESIZE_ENABLED	SMALLINT	tablespace_auto_resize_enabled - 表スペースの自動サイズ変更可能
TBSP_TOTAL_SIZE_KB	BIGINT	<p>表スペースの合計サイズ (キロバイト)。これは <math>(tablespace\_total\_pages * tablespace\_page\_size) / 1024</math> として計算されます。ここで <i>tablespace_total_pages</i> と <i>tablespace_page_size</i> は以下のモニター・エレメントを表しています。</p> <ul style="list-style-type: none"> <li>• tablespace_total_pages - 表スペース内の合計ページ数</li> <li>• tablespace_page_size - 表スペースのページ・サイズ</li> </ul>

表 181. MON\_TBSP\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
TBSP_USABLE_SIZE_KB	BIGINT	<p>使用可能な表スペースの合計サイズ (キロバイト)。これは、表スペースの合計サイズからオーバーヘッド・ページに使われるスペースを差し引いた数値に等しくなります。これは</p> $(\text{tablespace\_usable\_pages} * \text{tablespace\_page\_size}) / 1024$ <p>として計算されます。ここで <i>tablespace_usable_pages</i> と <i>tablespace_page_size</i> は以下のモニター・エレメントを表しています。</p> <ul style="list-style-type: none"> <li>• <i>tablespace_usable_pages</i> - 表スペース内の使用可能ページ数</li> <li>• <i>tablespace_page_size</i> - 表スペースのページ・サイズ</li> </ul>
TBSP_UTILIZATION_PERCENT	DECIMAL(5,2)	<p>表スペースの使用率 (パーセンテージ)。</p> <p><i>tablespace_usable_pages</i> がゼロより大きい場合、これは</p> $(\text{tablespace\_used\_pages} / \text{tablespace\_usable\_pages}) * 100$ <p>として計算されます。ここで <i>tablespace_used_pages</i> と <i>tablespace_usable_pages</i> は以下のモニター・エレメントを表しています。</p> <ul style="list-style-type: none"> <li>• <i>tablespace_used_pages</i> - 表スペース内の使用されているページ数</li> <li>• <i>tablespace_usable_pages</i> - 表スペース内の使用可能ページ数</li> </ul> <p><i>tablespace_usable_pages</i> がゼロより大きくない場合は NULL が戻されます。</p>
TBSP_PAGE_TOP	BIGINT	<p><i>tablespace_page_top</i> - 表スペース最高水準点</p>

表 181. MON\_TBSP\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
DATA_PHYSICAL_READS	BIGINT	<p>TEMPORARY、REGULAR、および LARGE 表スペースに対して、表スペース・コンテナ(物理) から読み取られたデータ・ページの数を示します。これは (<i>pool_data_p_reads</i> + <i>pool_temp_data_p_reads</i>) として計算されます。ここで <i>pool_data_p_reads</i> と <i>pool_temp_data_p_reads</i> は以下のモニター・エレメントを表しています。</p> <ul style="list-style-type: none"> <li>• <i>pool_data_p_reads</i> - バッファ・プール・データの物理読み取り</li> <li>• <i>pool_temp_data_p_reads</i> - バッファ・プール一時データの物理読み取り</li> </ul>
DATA_HIT_RATIO_PERCENT	DECIMAL(5,2)	<p>データのヒット率。つまり、データ・ページ要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要のなかった時間のパーセンテージ。</p>
INDEX_PHYSICAL_READS	BIGINT	<p>TEMPORARY、REGULAR、および LARGE 表スペースに対して、表スペース・コンテナ(物理) から読み取られた索引ページの数を示します。これは (<i>pool_index_p_reads</i> + <i>pool_temp_index_p_reads</i>) として計算されます。ここで <i>pool_index_p_reads</i> と <i>pool_temp_index_p_reads</i> は以下のモニター・エレメントを表しています。</p> <ul style="list-style-type: none"> <li>• <i>pool_index_p_reads</i> - バッファ・プール索引の物理読み取り</li> <li>• <i>pool_temp_index_p_reads</i> - バッファ・プール一時索引の物理読み取り</li> </ul>

表 181. MON\_TBSP\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
INDEX_HIT_RATIO _PERCENT	DECIMAL(5,2)	索引のヒット率。つまり、索引データ・ページ要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要のなかった時間のパーセンテージ。
XDA_PHYSICAL_READS	BIGINT	<p>TEMPORARY、REGULAR、および LARGE 表スペースに対して、表スペース・コンテナ (物理) から読み取られた XML ストレージ・オブジェクト (XDA) 用データ・ページの数を示します。これは <math>(pool\_xda\_p\_reads + pool\_temp\_xda\_p\_reads)</math> として計算されます。ここで <math>pool\_xda\_p\_reads</math> と <math>pool\_temp\_xda\_p\_reads</math> は以下のモニター・エレメントを表しています。</p> <ul style="list-style-type: none"> <li>• <math>pool\_xda\_p\_reads</math> - バッファ・プール XDA データの物理読み取り</li> <li>• <math>pool\_temp\_xda\_p\_reads</math> - バッファ・プール一時 XDA データの物理読み取り</li> </ul>
XDA_HIT_RATIO_PERCENT	DECIMAL(5,2)	補助ストレージ・オブジェクトのヒット率。つまり、XML ストレージ・オブジェクト (XDA) に関するデータ・ページ要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要のなかった時間のパーセンテージ。DB2 pureScale 環境では、この値は、データベース・マネージャーがローカル・バッファ・プールで XDA のデータ・ページを見つけるために使った時間のパーセンテージです。

表 181. MON\_TBSP\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
GBP_DATA_HIT_RATIO_PERCENT	DECIMAL(5,2)	グループ・バッファ・プール・データのヒット率。グループ・バッファ・プールに既にページが存在しているため、データ・ページ要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要がなかった時間のパーセンテージ。この値は、DB2 pureScale 環境の外では NULL です。
GBP_INDEX_HIT_RATIO_PERCENT	DECIMAL(5,2)	グループ・バッファ・プール索引のヒット率。グループ・バッファ・プールに既にページが存在しているため、索引ページ要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要がなかった時間のパーセンテージ。この値は、DB2 pureScale 環境の外では NULL です。
GBP_XDA_HIT_RATIO_PERCENT	DECIMAL(5,2)	グループ・バッファ・プールの補助ストレージ・オブジェクトのヒット率。つまり、グループ・バッファ・プールにページが存在するため、XML ストレージ・オブジェクト (XDA) に関するデータ・ページ要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要がなかった時間のパーセンテージ。この値は、DB2 pureScale 環境の外では常に NULL です。

## MON\_WORKLOAD\_SUMMARY - すべてのワークロードに関するメトリックの取得

MON\_WORKLOAD\_SUMMARY 管理ビューは、現在接続されているデータベース内のすべてのワークロードに関する主要なメトリックを戻します。これはワークロードごとに受け取る処理を示すため、システムの概要をモニターするのに役立ちます。

戻されるメトリックは、データベースの全メンバーにおいて、識別されたワークロード・オブジェクトにマップされる接続によってサブミットされた要求についてのすべてのメトリックの累計を表します。



スキーマは SYSIBMADM です。

## 許可

以下のいずれかの権限が必要です。

- MON\_WORKLOAD\_SUMMARY 管理ビューに対する SELECT 特権
- MON\_WORKLOAD\_SUMMARY 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

## デフォルトの PUBLIC 特権

なし

## 戻される情報

表 182. MON\_WORKLOAD\_SUMMARY 管理ビューによって戻される情報

列名	データ・タイプ	説明またはモニター・エレメント
WORKLOAD_NAME	VARCHAR(128)	workload_name - ワークロード名
WORKLOAD_ID	INTEGER	workload_id - ワークロード ID
TOTAL_APP_COMMITS	BIGINT	total_app_commits - アプリケーション・コミットの合計回数モニター・エレメント
TOTAL_APP_ROLLBACKS	BIGINT	total_app_rollbacks - アプリケーション・ロールバックの合計回数モニター・エレメント
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 完了したアクティビティの合計数モニター・エレメント
APP_RQSTS_COMPLETED_TOTAL	BIGINT	指定されたサービス・サブクラスに関する、データベースの全メンバーで正常に完了した外部(アプリケーション) 要求の総数。
AVG_RQST_CPU_TIME	BIGINT	正常に完了したすべての外部要求によって費やされた平均 CPU 時間 (マイクロ秒)。これは、ユーザーとシステムの両方の CPU 時間の合計を表します。
ROUTINE_TIME_RQST_PERCENT	DECIMAL(5,2)	データベース・サーバーが要求を処理するのに費やした時間のうち、ユーザー・ルーチンの実行に費やした時間のパーセンテージ。

表 182. MON\_WORKLOAD\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
RQST_WAIT_TIME_PERCENT	DECIMAL(5,2)	要求を処理するのに費やされた時間のうち、DB2 データベース・サーバー内部での待機に費やされた時間のパーセンテージ。
ACT_WAIT_TIME_PERCENT	DECIMAL(5,2)	アクティビティーの実行に費やされた時間のうち、DB2 データベース・サーバー内部での待機に費やされた時間のパーセンテージ。
IO_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、入出力操作に費やされた時間のパーセンテージ。これには、直接読み取り/直接書き込みの実行に費やされた時間に加えて、表スペースからバッファ・プールへのデータおよび索引ページの読み取りや元のディスクへの書き込みに費やされた時間が含まれます。
LOCK_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、ロック待機に費やされた時間のパーセンテージ。
AGENT_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、コンセントレーター構成のもとでエージェントを待機するためにキューに入れられたアプリケーションによって費やされた時間のパーセンテージ。
NETWORK_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 データベース・サーバー内部での待機に費やされた時間のうち、クライアント/サーバー通信のために費やされた時間のパーセンテージ。これには、TCP/IP または IPC プロトコルを使ってデータを送受信するのに費やされた時間が含まれます。

表 182. MON\_WORKLOAD\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
SECTION_PROC_ TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、セクションの実行に費やした時間のパーセンテージ。これには、ソートの実行に費やされた時間が含まれます。
SECTION_SORT_ PROC_TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、セクションの実行中にソートの実行に費やした時間のパーセンテージ。
COMPILE_PROC_ TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、SQL ステートメントのコンパイルに費やした時間のパーセンテージ。これには、明示的および暗黙的コンパイル時間が含まれます。
TRANSACT_END_PROC_ TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、コミット処理の実行またはトランザクションのロールバックに費やした時間のパーセンテージ。
UTILS_PROC_ TIME_PERCENT	DECIMAL(5,2)	要求をアクティブに処理するデータベース・サーバーが、ユーティリティの実行に費やした時間のパーセンテージ。これには、 <b>runstats</b> 、再編成、およびロード操作の実行が含まれます。
AVG_LOCK_WAITS_ _PER_ACT	BIGINT	各コーディネーター・アクティビティ (成功およびアボート) に対して、アプリケーションまたは接続がロックを待機した平均回数。
AVG_LOCK_TIMEOUTS_ _PER_ACT	BIGINT	各コーディネーター・アクティビティ (成功およびアボート) に対して、オブジェクトのロック要求がタイムアウトになった平均回数。
AVG_DEADLOCKS_ PER_ACT	BIGINT	コーディネーター・アクティビティ (成功およびアボート) ごとのデッドロックの平均数。

表 182. MON\_WORKLOAD\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
AVG_LOCK_ESCALATIONS_PER_ACT	BIGINT	各コーディネーター・アクティビティ (成功およびアボート) に対して、いくつかの行ロックから表ロックにロックがエスカレートした平均回数。
ROWS_READ_PER_ROWS_RETURNED	BIGINT	アプリケーションに戻された各行に対して、表から読み取られた平均行数。
TOTAL_BP_HIT_RATIO_PERCENT	DECIMAL(5,2)	XML ストレージ・オブジェクト (XDA) の要求を含めて、データまたは索引ページの要求を処理するためにデータベース・マネージャーがディスクからページをロードする必要のなかった時間のパーセンテージ。DB2 pureScale 環境では、この値はローカル・バッファ・プールの合計ヒット率を表します。
TOTAL_GBP_HIT_RATIO_PERCENT	DECIMAL(5,2)	DB2 pureScale 環境では、ページがグループ・バッファ・プールに置かれていたため、データ・ページ要求、索引ページ要求、または XML ストレージ・オブジェクト (XDA) ページ要求を処理するためにデータベース・マネージャーがディスクからローカル・バッファ・プールにページをロードする必要がなかった時間のパーセンテージ。この値は、DB2 pureScale 環境の外では常に NULL です。
CF_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 pureScale 環境では、キャッシング・ファシリティ通信の待機に費やされた合計待機時間のパーセンテージ。この値は、DB2 pureScale 環境の外では常に NULL です。
RECLAIM_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 pureScale 環境では、ページ再利用の待機に費やされた合計待機時間のパーセンテージ。この値は、DB2 pureScale 環境の外では常に NULL です。

表 182. MON\_WORKLOAD\_SUMMARY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明またはモニター・エレメント
SPACEMAPPAGE_RECLAIM_WAIT_TIME_PERCENT	DECIMAL(5,2)	DB2 pureScale 環境 では、スペース・マップ・ページ再利用の待機に費やされた合計待機時間のパーセンテージ。この値は、DB2 pureScale 環境の外では常に NULL です。

## MQSeries ルーチン

### MQPUBLISH

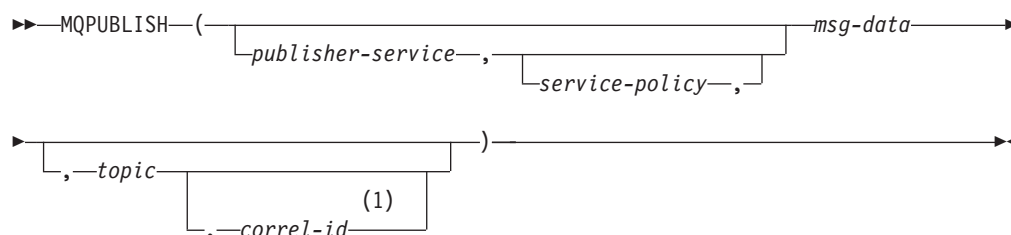
MQPUBLISH 関数はデータを MQSeries にパブリッシュします。

詳しくは、<http://www.ibm.com/software/mqseries/> を参照してください。

MQPUBLISH 関数は、*msg-data* に含まれるデータを、*publisher-service* に指定された MQSeries パブリッシャーにパブリッシュします。その際、*service-policy* に定義されたサービス・ポリシーの品質を使用します。オプションで、メッセージ・トピックおよびユーザー定義のメッセージ相関 ID を指定することができます。

この結果のデータ・タイプは VARCHAR(1) です。関数の結果は、成功した場合は '1' で失敗した場合は '0' です。

### 構文



### 注:

- 1 *service* および *policy* が指定されていない限り、*correl-id* を指定することはできません。

スキーマは、非トランザクション・メッセージ・キューイング関数の場合は DB2MQ、1 フェーズ・コミット・トランザクション MQ 関数の場合は DB2MQ1C です。

### 関数のパラメーター

#### *publisher-service*

メッセージの送信先である MQSeries 論理宛先を含むストリング。

*publisher-service* を指定する場合、パブリッシャー・サービスのタイプ値が「P」の DB2MQ.MQPUBSUB 表に定義されているパブリッシャー・サービス・ポイ

ントを参照する必要があります。 `publisher-service` を指定しない場合、`DB2.DEFAULT.PUBLISHER` が使用されます。 `publisher-service` の最大サイズは 48 バイトです。

#### *service-policy*

このメッセージの処理に使われる MQSeries サービス・ポリシーを含むストリング。 `service-policy` を指定する場合、`DB2MQ.MQPOLICY` 表に定義されたポリシーを参照する必要があります。 サービス・ポリシーは、このメッセージング操作に適用されるサービス・オプションの一連の品質を定義します。これらのオプションには、メッセージ優先順位やメッセージ・パーシスタンスが含まれません。 `service-policy` を指定しない場合、デフォルトの `DB2.DEFAULT.POLICY` が使用されます。 `service-policy` の最大サイズは 48 バイトです。

#### *msg-data*

MQSeries を介して送られるデータを含むストリング式。 `VARCHAR` ストリング式の最大サイズは 32 000 バイトで、`CLOB` ストリング式の最大サイズは 1M バイトです。

#### *topic*

メッセージ公表のトピックを含むストリング式。トピックを指定しない場合、メッセージにはトピックが関連付けられません。 `topic` の最大サイズは 40 バイトです。 1 つのストリングに複数のトピックを指定することができます (最大で 40 文字の長さ)。それぞれのトピックは、コロンで区切ります。例えば "t1:t2:the third topic" は、3 つのトピック t1、 t2、 および "the third topic" がメッセージに関連付けられていることを示します。

#### *correl-id*

このメッセージに関連した相関 ID を含むオプションのストリング式。 `correl-id` は、要求/応答シナリオで要求と応答を関連付けるためにしばしば使用されます。これを指定しない場合、メッセージには相関 ID が追加されません。 `correl-id` の最大サイズは 24 バイトです。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: この例では、ストリング "Testing 123" をデフォルト・パブリッシャー・サービス (`DB2.DEFAULT.PUBLISHER`) にパブリッシュします。その際、デフォルト・ポリシー (`DB2.DEFAULT.POLICY`) を使用します。メッセージには相関 ID とトピックが指定されません。

```
VALUES MQPUBLISH('Testing 123')
```

例 2: この例では、トピック "TESTS" を使って、ストリング "Testing 345" をパブリッシャー・サービス "MYPUBLISHER" にパブリッシュします。デフォルト・ポリシーを使用し、相関 ID は指定しません。

```
VALUES MQPUBLISH('MYPUBLISHER','Testing 345', 'TESTS')
```

例 3: この例では、ストリング "Testing 678" をパブリッシャー・サービス "MYPUBLISHER" にパブリッシュします。その際、ポリシー "MYPOLICY" と相関 ID "TEST1" を使用します。メッセージは、トピック "TESTS" とともにパブリッシュされます。

```
VALUES MQPUBLISH('MYPUBLISHER','MYPOLICY','Testing 678','TESTS','TEST1')
```

例 4: この例では、ストリング "Testing 901" をパブリッシャー・サービス "MYPUBLISHER" にパブリッシュします。トピック "TESTS" とデフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用し、相関 ID は使用しません。

```
VALUES MQPUBLISH('Testing 901','TESTS')
```

## MQREAD

MQREAD 関数は、*receive-service* で指定された MQSeries のロケーションからメッセージを戻します。その際、サービス・ポリシー *service-policy* で定義された品質を使用します。

この操作を実行しても、*receive-service* に関連付けられたキューからメッセージは除去されません。ただし、キューの先頭にあるメッセージのみを戻します。

この結果のデータ・タイプは VARCHAR(32000) です。戻すメッセージが存在しない場合、結果は NULL 値になります。

## 構文

```
MQREAD ( ( receive-service [, service-policy ] ) )
```

スキーマは、非トランザクション・メッセージ・キューイング関数の場合は DB2MQ、1 フェーズ・コミット・トランザクション MQ 関数の場合は DB2MQ1C です。

## 関数のパラメーター

### *receive-service*

受信するメッセージの送信元である MQSeries 論理宛先を含むストリング。  
*receive-service* を指定する場合、DB2MQ.MQSERVICE 表に定義されたサービス・ポイントを参照する必要があります。サービス・ポイントとは、メッセージが送受信される論理上のエンドポイントです。サービス・ポイントの定義には、MQSeries キュー・マネージャーおよびキューの名前が含まれます。  
*receive-service* を指定しない場合、DB2.DEFAULT.SERVICE が使用されます。  
*receive-service* の最大サイズは 48 バイトです。

### *service-policy*

このメッセージの処理に使われる MQSeries サービス・ポリシーを含むストリング。 *service-policy* を指定する場合、DB2MQ.MQPOLICY 表に定義されたポリシーを参照する必要があります。 サービス・ポリシーは、このメッセージング操作に適用されるサービス・オプションの一連の品質を定義します。これらのオプションには、メッセージ優先順位やメッセージ・パーシスタンスが含まれます。 *service-policy* を指定しない場合、デフォルトの DB2.DEFAULT.POLICY が使用されます。 *service-policy* の最大サイズは 48 バイトです。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューの先頭にあるメッセージを読み取ります。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。

```
VALUES MQREAD()
```

例 2: この例では、サービス "MYSERVICE" に指定されたキューの先頭にあるメッセージを、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用して読み取ります。

```
VALUES MQREAD('MYSERVICE')
```

例 3: この例では、サービス "MYSERVICE" に指定されたキューの先頭にあるメッセージを、ポリシー "MYPOLICY" を使用して読み取ります。

```
VALUES MQREAD('MYSERVICE', 'MYPOLICY')
```

## MQREADALL

MQREADALL 表関数は *receive-service* で指定された MQSeries のロケーションから、メッセージおよびメッセージ・メタデータを含む表を戻します。その際、サービス・ポリシー *service-policy* の品質を使用します。

この操作を実行しても、*receive-service* に関連付けられたキューからメッセージは除去されません。



## 構文

```
MQREADALL ( ( receive-service [, service-policy ] num-rows ) )
```

スキーマは、非トランザクション・メッセージ・キューイング関数の場合は DB2MQ、1 フェーズ・コミット・トランザクション MQ 関数の場合は DB2MQ1C です。

## 表関数パラメーター

### *receive-service*

読み取る対象のメッセージが格納されている MQSeries 論理宛先を含むストリング。 *receive-service* を指定する場合、DB2MQ.MQSERVICE 表に定義されたサービス・ポイントを参照する必要があります。サービス・ポイントとは、メッセージが送受信される論理上のエンドポイントです。サービス・ポイントの定義には、MQSeries キュー・マネージャーおよびキューの名前が含まれます。 *receive-service* を指定しない場合、DB2.DEFAULT.SERVICE が使用されます。 *receive-service* の最大サイズは 48 バイトです。

### *service-policy*

このメッセージの処理に使われる MQSeries サービス・ポリシーを含むストリング。 *service-policy* を指定する場合、DB2MQ.MQPOLICY 表に定義されたポリシーを参照させます。サービス・ポリシーは、このメッセージング操作に適用されるサービス・オプションの一連の品質を定義します。これらのオプションには、メッセージ優先順位やメッセージ・パーシスタンスが含まれます。 *service-policy* を指定しない場合、デフォルトの DB2.DEFAULT.POLICY が使用されます。 *service-policy* の最大サイズは 48 バイトです。

### *num-rows*

関数によって戻されるメッセージの最大数を示す正の整数。

*num-rows* を指定すると、最大で *num-rows* の数だけメッセージが戻されます。 *num-rows* を指定しない場合、入手可能なすべてのメッセージが戻されます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューからすべてのメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。メッセージおよびすべてのメタデータが、1 つの表として戻されます。

```
SELECT * FROM table (MQREADALL()) AS T
```

例 2: この例では、サービス MYSERVICE に指定されたキューの先頭からすべてのメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。MSG 列と CORRELID 列のみが戻されます。

```
SELECT T.MSG, T.CORRELID FROM table (MQREADALL('MYSERVICE')) AS T
```

例 3: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューの先頭を読み取ります。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。CORRELID が '1234' のメッセージのみが戻されます。すべての列が戻されます。

```
SELECT * FROM table (MQREADALL()) AS T WHERE T.CORRELID = '1234'
```

例 4: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューの先頭から数えて最初の 10 個のメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。すべての列が戻されます。

```
SELECT * FROM table (MQREADALL(10)) AS T
```

## 戻される情報

表 183. MQREADALL 表関数によって戻される情報

列名	データ・タイプ	説明
MSG	VARCHAR(32000)	MQSeries メッセージの内容を含みます。
CORRELID	VARCHAR(24)	メッセージの識別に使用できる相関 ID を含みます。この ID を使用して、キューから特定メッセージを選択できます。要求と応答のシナリオでは、相関 ID を使用すると、応答を特定の要求と関連付けられるようになります。
TOPIC	VARCHAR(40)	メッセージがパブリッシュされたときのトピックを含みます (使用可能な場合)。
QNAME	VARCHAR(48)	メッセージが受信されたキューの名前を含みます。
MSGID	CHAR(24)	このメッセージの割り当てられた固有の MQSeries ID を含みます。

表 183. MQREADALL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
MSGFORMAT	VARCHAR(8)	MQSeries によって定義された、メッセージの形式を含みます。通常、ストリングの形式は MQSTR です。

## MQREADALLCLOB

MQREADALLCLOB 表関数は *receive-service* で指定された MQSeries のロケーションから、メッセージおよびメッセージ・メタデータを含む表を戻します。その際、サービス・ポリシー *service-policy* の品質を使用します。

この操作を実行しても、*receive-service* に関連付けられたキューからメッセージは除去されません。

### 構文



スキーマは DB2MQ です。

### 表関数パラメーター

#### *receive-service*

読み取る対象のメッセージが格納されている MQSeries 論理宛先を含むストリング。 *receive-service* を指定する場合、DB2MQ.MQSERVICE 表に定義されたサービス・ポイントを参照する必要があります。サービス・ポイントとは、メッセージが送受信される論理上のエンドポイントです。サービス・ポイントの定義には、MQSeries キュー・マネージャーおよびキューの名前が含まれます。 *receive-service* を指定しない場合、DB2.DEFAULT.SERVICE が使用されます。 *receive-service* の最大サイズは 48 バイトです。

#### *service-policy*

このメッセージの処理に使われる MQSeries サービス・ポリシーを含むストリング。 *service-policy* を指定する場合、DB2MQ.MQPOLICY 表に定義されたポリシーを参照させます。サービス・ポリシーは、このメッセージング操作に適用されるサービス・オプションの一連の品質を定義します。これらのオプションには、メッセージ優先順位やメッセージ・パーシスタンスが含まれます。 *service-policy* を指定しない場合、デフォルトの DB2.DEFAULT.POLICY が使用されます。 *service-policy* の最大サイズは 48 バイトです。

#### *num-rows*

関数によって戻されるメッセージの最大数を示す正の整数。

*num-rows* を指定すると、最大で *num-rows* の数だけメッセージが戻されます。 *num-rows* を指定しない場合、入手可能なすべてのメッセージが戻されます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューからすべてのメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。メッセージおよびすべてのメタデータが、1 つの表として戻されます。

```
SELECT * FROM table (MQREADALLCLOB()) AS T
```

例 2: この例では、サービス MYSERVICE に指定されたキューの先頭からすべてのメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。MSG 列と CORRELID 列のみが戻されます。

```
SELECT T.MSG, T.CORRELID FROM table (MQREADALLCLOB('MYSERVICE')) AS T
```

例 3: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューの先頭を読み取ります。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。CORRELID が '1234' のメッセージのみが戻されます。すべての列が戻されます。

```
SELECT * FROM table (MQREADALLCLOB()) AS T WHERE T.CORRELID = '1234'
```

例 4: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューの先頭から数えて最初の 10 個のメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。すべての列が戻されます。

```
SELECT * FROM table (MQREADALLCLOB(10)) AS T
```

## 戻される情報

表 184. MQREADALLCLOB 表関数によって戻される情報

列名	データ・タイプ	説明
MSG	CLOB(1M)	MQSeries メッセージの内容を含みます。

表 184. MQREADALLCLOB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
CORRELID	VARCHAR(24)	メッセージの識別に使用できる関連 ID を含みます。この ID を使用して、キューから特定メッセージを選択できます。要求と応答のシナリオでは、関連 ID を使用すると、応答を特定の要求と関連付けられるようになります。
TOPIC	VARCHAR(40)	メッセージがパブリッシュされたときのトピックを含みます (使用可能な場合)。
QNAME	VARCHAR(48)	メッセージが受信されたキューの名前を含みます。
MSGID	CHAR(24)	このメッセージの割り当てられた固有の MQSeries ID を含みます。
MSGFORMAT	VARCHAR(8)	MQSeries によって定義された、メッセージの形式を含みます。通常、ストリングの形式は MQSTR です。

## MQREADCLOB

MQREADCLOB 関数は、*receive-service* で指定された MQSeries のロケーションからメッセージを戻します。その際、サービス・ポリシー *service-policy* で定義された品質を使用します。

この操作を実行しても、*receive-service* に関連付けられたキューからメッセージは除去されません。ただし、キューの先頭にあるメッセージのみを戻します。

結果のデータ・タイプは CLOB(1M) です。戻すメッセージが存在しない場合、結果は NULL 値になります。

### 構文

```

MQREADCLOB ( ( receive-service [, service-policy] ) )

```

スキーマは DB2MQ です。

### 関数のパラメーター

#### *receive-service*

受信するメッセージの送信元である MQSeries 論理宛先を含むストリング。  
*receive-service* を指定する場合、DB2MQ.MQSERVICE 表に定義されたサービス・ポイントを参照する必要があります。サービス・ポイントとは、メッセージが送受信される論理上のエンドポイントです。サービス・ポイントの定義に

は、MQSeries キュー・マネージャーおよびキューの名前が含まれます。  
*receive-service* を指定しない場合、DB2.DEFAULT.SERVICE が使用されます。  
*receive-service* の最大サイズは 48 バイトです。

#### *service-policy*

このメッセージの処理に使われる MQSeries サービス・ポリシーを含むストリング。  
*service-policy* を指定する場合、DB2MQ.MQPOLICY 表に定義されたポリシーを参照する必要があります。サービス・ポリシーは、このメッセージング操作に適用されるサービス・オプションの一連の品質を定義します。これらのオプションには、メッセージ優先順位やメッセージ・パーシスタンスが含まれます。  
*service-policy* を指定しない場合、デフォルトの DB2.DEFAULT.POLICY が使用されます。  
*service-policy* の最大サイズは 48 バイトです。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューの先頭にあるメッセージを読み取ります。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。

```
VALUES MQREADCLOB()
```

例 2: この例では、サービス "MYSERVICE" に指定されたキューの先頭にあるメッセージを、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用して読み取ります。

```
VALUES MQREADCLOB('MYSERVICE')
```

例 3: この例では、サービス "MYSERVICE" に指定されたキューの先頭にあるメッセージを、ポリシー "MYPOLICY" を使用して読み取ります。

```
VALUES MQREADCLOB('MYSERVICE', 'MYPOLICY')
```

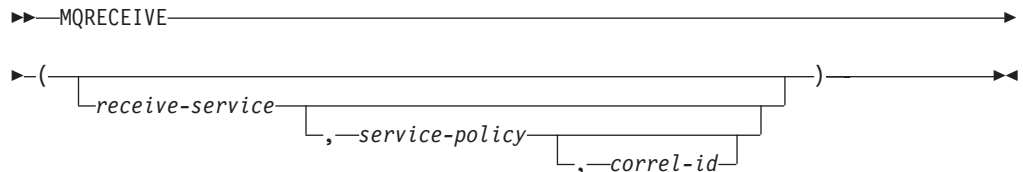
## MQRECEIVE

MQRECEIVE 関数は、*receive-service* で指定された MQSeries のロケーションからメッセージを戻します。その際、サービス・ポリシー *service-policy* の品質を使用します。

この操作を実行すると、*receive-service* に関連付けられたキューからメッセージが除去されます。*correl-id* を指定すると、相関 ID が一致する最初のメッセージが戻されます。*correl-id* を指定しない場合、キューの先頭にあるメッセージが戻されます。

この結果のデータ・タイプは VARCHAR(32000) です。戻すメッセージが存在しない場合、結果は NULL 値になります。

## 構文



スキーマは、非トランザクション・メッセージ・キューイング関数の場合は DB2MQ、1 フェーズ・コミット・トランザクション MQ 関数の場合は DB2MQ1C です。

## 関数のパラメーター

### *receive-service*

受信するメッセージの送信元である MQSeries 論理宛先を含むストリング。*receive-service* を指定する場合、DB2MQ.MQSERVICE 表に定義されたサービス・ポイントを参照する必要があります。サービス・ポイントとは、メッセージが送受信される論理上のエンドポイントです。サービス・ポイントの定義には、MQSeries キュー・マネージャーおよびキューの名前が含まれます。*receive-service* を指定しない場合、DB2.DEFAULT.SERVICE が使用されます。*receive-service* の最大サイズは 48 バイトです。

### *service-policy*

このメッセージの処理に使われる MQSeries サービス・ポリシーを含むストリング。*service-policy* を指定する場合、DB2MQ.MQPOLICY 表に定義されたポリシーを参照する必要があります。サービス・ポリシーは、このメッセージング操作に適用されるサービス・オプションの一連の品質を定義します。これらのオプションには、メッセージ優先順位やメッセージ・パーシスタンスが含まれます。*service-policy* を指定しない場合、デフォルトの DB2.DEFAULT.POLICY が使用されます。*service-policy* の最大サイズは 48 バイトです。

### *correl-id*

このメッセージに関連付けられた、オプションの相関 ID を含むストリング。*correl-id* は、要求/応答シナリオで要求と応答を関連付けるためにしばしば使用されます。これを指定しない場合、相関 ID は指定されません。*correl-id* の最大サイズは 24 バイトです。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限

- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

例 1: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューの先頭にあるメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。

```
VALUES MQRECEIVE()
```

例 2: この例では、サービス "MYSERVICE" に指定されたキューの先頭にあるメッセージを、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使って受信します。

```
VALUES MQRECEIVE('MYSERVICE')
```

例 3: この例では、サービス "MYSERVICE" に指定されたキューの先頭にあるメッセージを、ポリシー "MYPOLICY" を使って受信します。

```
VALUES MQRECEIVE('MYSERVICE', 'MYPOLICY')
```

例 4: この例では、相関 ID '1234' に一致する最初のメッセージを、サービス "MYSERVICE" に指定されたキューの先頭から検索して受信します。その際、ポリシー "MYPOLICY" を使用します。

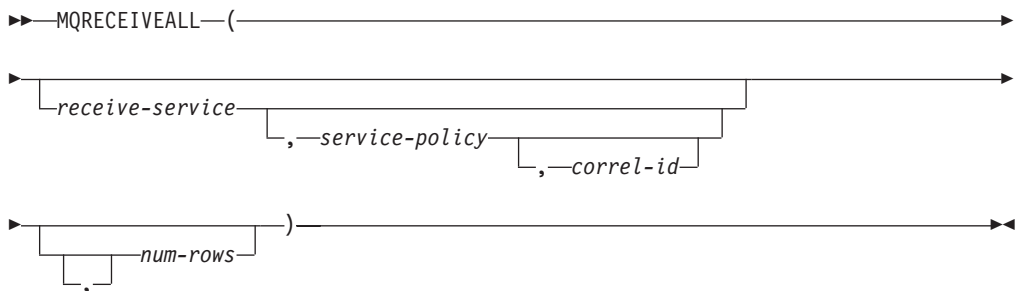
```
VALUES MQRECEIVE('MYSERVICE', 'MYPOLICY', '1234')
```

## MQRECEIVEALL

MQRECEIVEALL 表関数は *receive-service* で指定された MQSeries のロケーションから、メッセージおよびメッセージ・メタデータを含む表を戻します。その際、サービス・ポリシー *service-policy* の品質を使用します。

この操作を実行すると、*receive-service* に関連付けられたキューからメッセージが除去されます。

### 構文





スキーマは、非トランザクション・メッセージ・キューイング関数の場合は DB2MQ、1 フェーズ・コミット・トランザクション MQ 関数の場合は DB2MQ1C です。

## 表関数パラメーター

### *receive-service*

受信するメッセージの送信元である MQSeries 論理宛先を含むストリング。  
*receive-service* を指定する場合、DB2MQ.MQSERVICE 表に定義されたサービス・ポイントを参照する必要があります。サービス・ポイントとは、メッセージが送受信される論理上のエンドポイントです。サービス・ポイントの定義には、MQSeries キュー・マネージャーおよびキューの名前が含まれます。  
*receive-service* を指定しない場合、DB2.DEFAULT.SERVICE が使用されます。  
*receive-service* の最大サイズは 48 バイトです。

### *service-policy*

このメッセージの処理に使われる MQSeries サービス・ポリシーを含むストリング。  
*service-policy* を指定する場合、DB2MQ.MQPOLICY 表に定義されたポリシーを参照させます。サービス・ポリシーは、このメッセージング操作に適用されるサービス・オプションの一連の品質を定義します。これらのオプションには、メッセージ優先順位やメッセージ・パーシスタンスが含まれます。  
*service-policy* を指定しない場合、デフォルトの DB2.DEFAULT.POLICY が使用されます。  
*service-policy* の最大サイズは 48 バイトです。

### *correl-id*

このメッセージに関連した相関 ID を含むオプション・ストリング。  
*correl-id* は、要求/応答シナリオで要求と応答を関連付けるためにしばしば使用されます。これを指定しない場合、相関 ID は指定されません。  
*correl-id* の最大サイズは 24 バイトです。

*correl-id* を指定すると、相関 ID が一致するすべてのメッセージが戻され、キューから除去されます。  
*correl-id* を指定しない場合、キューの先頭にあるメッセージが戻されます。

### *num-rows*

関数によって戻されるメッセージの最大数を示す正の整数。

*num-rows* を指定すると、最大で *num-rows* の数だけメッセージが戻されます。  
*num-rows* を指定しない場合、入手可能なすべてのメッセージが戻されます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューからすべてのメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。メッセージおよびすべてのメタデータが、1 つの表として戻されます。

```
SELECT * FROM table (MQRECEIVEALL()) AS T
```

例 2: この例では、サービス MYSERVICE に指定されたキューの先頭からすべてのメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。MSG 列と CORRELID 列のみが戻されます。

```
SELECT T.MSG, T.CORRELID FROM table (MQRECEIVEALL('MYSERVICE')) AS T
```

例 3: この例では、ポリシー "MYPOLICY" を使用して、サービス "MYSERVICE" に指定されたキューの先頭からすべてのメッセージを受信します。CORRELID が '1234' のメッセージのみが戻されます。MSG 列と CORRELID 列のみが戻されます。

```
SELECT T.MSG, T.CORRELID FROM table  
(MQRECEIVEALL('MYSERVICE','MYPOLICY','1234')) AS T
```

例 4: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューの先頭から数えて最初の 10 個のメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。すべての列が戻されます。

```
SELECT * FROM table (MQRECEIVEALL(10)) AS T
```

## 戻される情報

表 185. MQRECEIVEALL 表関数によって戻される情報

列名	データ・タイプ	説明
MSG	VARCHAR(32000)	MQSeries メッセージの内容を含みます。
CORRELID	VARCHAR(24)	メッセージの識別に使用できる相関 ID を含みます。この ID を使用して、キューから特定メッセージを選択できます。要求と応答のシナリオでは、相関 ID を使用すると、応答を特定の要求と関連付けられるようになります。
TOPIC	VARCHAR(40)	メッセージがパブリッシュされたときのトピックを含みます (使用可能な場合)。
QNAME	VARCHAR(48)	メッセージが受信されたキューの名前を含みます。
MSGID	CHAR(24)	このメッセージの割り当てられた固有の MQSeries ID を含みます。

表 185. MQRECEIVEALL 表関数によって戻される情報 (続き)

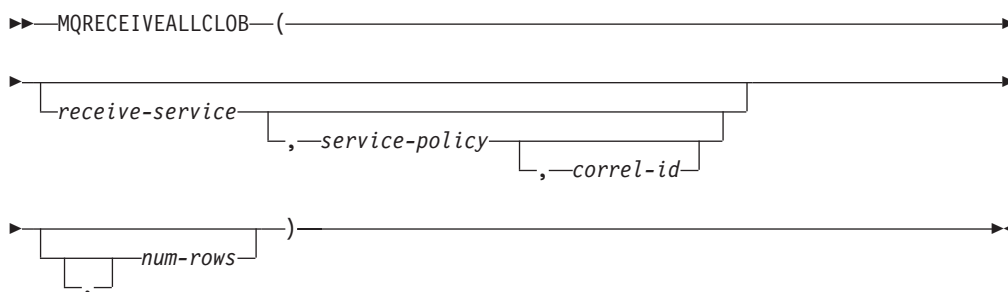
列名	データ・タイプ	説明
MSGFORMAT	VARCHAR(8)	MQSeries によって定義された、メッセージの形式を含みます。通常、ストリングの形式は MQSTR です。

## MQRECEIVEALLCLOB

MQRECEIVEALLCLOB 表関数は *receive-service* で指定された MQSeries のロケーションから、メッセージおよびメッセージ・メタデータを含む表を戻します。その際、サービス・ポリシー *service-policy* の品質を使用します。

この操作を実行すると、*receive-service* に関連付けられたキューからメッセージが除去されます。

### 構文



スキーマは DB2MQ です。

### 表関数パラメーター

#### *receive-service*

受信するメッセージの送信元である MQSeries 論理宛先を含むストリング。  
*receive-service* を指定する場合、DB2MQ.MQSERVICE 表に定義されたサービス・ポイントを参照する必要があります。サービス・ポイントとは、メッセージが送受信される論理上のエンドポイントです。サービス・ポイントの定義には、MQSeries キュー・マネージャーおよびキューの名前が含まれます。  
*receive-service* を指定しない場合、DB2.DEFAULT.SERVICE が使用されます。  
*receive-service* の最大サイズは 48 バイトです。

#### *service-policy*

このメッセージの処理に使われる MQSeries サービス・ポリシーを含むストリング。  
*service-policy* を指定する場合、DB2MQ.MQPOLICY 表に定義されたポリシーを参照させます。サービス・ポリシーは、このメッセージング操作に適用されるサービス・オプションの一連の品質を定義します。これらのオプションには、メッセージ優先順位やメッセージ・パーシスタンスが含まれます。  
*service-policy* を指定しない場合、デフォルトの DB2.DEFAULT.POLICY が使用されます。  
*service-policy* の最大サイズは 48 バイトです。

### *correl-id*

このメッセージに関連した相関 ID を含むオプション・ストリング。 *correl-id* は、要求/応答シナリオで要求と応答を関連付けるためにしばしば使用されます。これを指定しない場合、相関 ID は指定されません。 *correl-id* の最大サイズは 24 バイトです。

*correl-id* を指定すると、相関 ID が一致するメッセージのみが戻されます。  
*correl-id* を指定しない場合、キューの先頭にあるメッセージが戻されます。

### *num-rows*

関数によって戻されるメッセージの最大数を示す正の整数。

*num-rows* を指定すると、最大で *num-rows* の数だけメッセージが戻されます。  
*num-rows* を指定しない場合、入手可能なすべてのメッセージが戻されます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューからすべてのメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。メッセージおよびすべてのメタデータが、1 つの表として戻されます。

```
SELECT * FROM table (MQRECEIVEALLCLOB()) AS T
```

例 2: この例では、サービス MYSERVICE に指定されたキューの先頭からすべてのメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。MSG 列と CORRELID 列のみが戻されます。

```
SELECT T.MSG, T.CORRELID  
FROM table (MQRECEIVEALLCLOB('MYSERVICE')) AS T
```

例 3: この例では、ポリシー "MYPOLICY" を使用して、サービス "MYSERVICE" に指定されたキューの先頭からすべてのメッセージを受信します。CORRELID が '1234' のメッセージのみが戻されます。MSG 列と CORRELID 列のみが戻されます。

```
SELECT T.MSG, T.CORRELID  
FROM table (MQRECEIVEALLCLOB('MYSERVICE','MYPOLICY','1234')) AS T
```

例 4: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューの先頭から数えて最初の 10 個のメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。すべての列が戻されません。

```
SELECT * FROM table (MQRECEIVEALLCLOB(10)) AS T
```

## 戻される情報

表 186. MQRECEIVEALLCLOB 表関数によって戻される情報

列名	データ・タイプ	説明
MSG	CLOB(1M)	MQSeries メッセージの内容を含みます。
CORRELID	VARCHAR(24)	メッセージの識別に使用できる相関 ID を含みます。この ID を使用して、キューから特定メッセージを選択できます。要求と応答のシナリオでは、相関 ID を使用すると、応答を特定の要求と関連付けられるようになります。
TOPIC	VARCHAR(40)	メッセージがパブリッシュされたときのトピックを含みます (使用可能な場合)。
QNAME	VARCHAR(48)	メッセージが受信されたキューの名前を含みます。
MSGID	CHAR(24)	このメッセージの割り当てられた固有の MQSeries ID を含みます。
MSGFORMAT	VARCHAR(8)	MQSeries によって定義された、メッセージの形式を含みます。通常、ストリングの形式は MQSTR です。

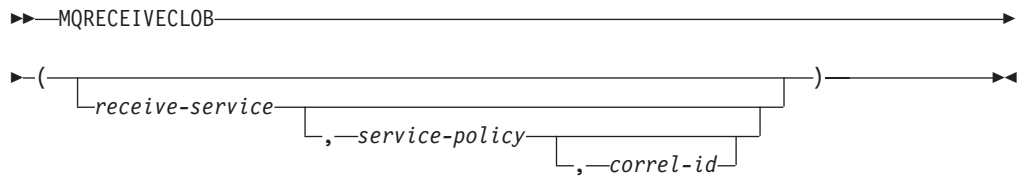
## MQRECEIVECLOB

MQRECEIVECLOB 関数は、*receive-service* で指定された MQSeries のロケーションからメッセージを戻します。その際、サービス・ポリシー *service-policy* の品質を使用します。

この操作を実行すると、*receive-service* に関連付けられたキューからメッセージが除去されます。 *correl-id* を指定すると、相関 ID が一致する最初のメッセージが戻されます。 *correl-id* を指定しない場合、キューの先頭にあるメッセージが戻されます。

結果のデータ・タイプは CLOB(1M) です。戻すメッセージが存在しない場合、結果は NULL 値になります。

## 構文



スキーマは DB2MQ です。

## 関数のパラメーター

### *receive-service*

受信するメッセージの送信元である MQSeries 論理宛先を含むストリング。  
*receive-service* を指定する場合、DB2MQ.MQSERVICE 表に定義されたサービス・ポイントを参照する必要があります。サービス・ポイントとは、メッセージが送受信される論理上のエンドポイントです。サービス・ポイントの定義には、MQSeries キュー・マネージャーおよびキューの名前が含まれます。  
*receive-service* を指定しない場合、DB2.DEFAULT.SERVICE が使用されます。  
*receive-service* の最大サイズは 48 バイトです。

### *service-policy*

このメッセージの処理に使われる MQSeries サービス・ポリシーを含むストリング。  
*service-policy* を指定する場合、DB2MQ.MQPOLICY 表に定義されたポリシーを参照する必要があります。サービス・ポリシーは、このメッセージング操作に適用されるサービス・オプションの一連の品質を定義します。これらのオプションには、メッセージ優先順位やメッセージ・パーシスタンスが含まれます。  
*service-policy* を指定しない場合、デフォルトの DB2.DEFAULT.POLICY が使用されます。  
*service-policy* の最大サイズは 48 バイトです。

### *correl-id*

このメッセージに関連付けられた、オプションの相関 ID を含むストリング。  
*correl-id* は、要求/応答シナリオで要求と応答を関連付けるためにしばしば使用されます。これを指定しない場合、相関 ID は使用されません。  
*correl-id* の最大サイズは 24 バイトです。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: この例では、デフォルト・サービス (DB2.DEFAULT.SERVICE) に指定されたキューの先頭にあるメッセージを受信します。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用します。

```
VALUES MQRECEIVECLOB()
```

例 2: この例では、サービス "MYSERVICE" に指定されたキューの先頭にあるメッセージを、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使って受信します。

```
VALUES MQRECEIVECLOB('MYSERVICE')
```

例 3: この例では、サービス "MYSERVICE" に指定されたキューの先頭にあるメッセージを、ポリシー "MYPOLICY" を使って受信します。

```
VALUES MQRECEIVECLOB('MYSERVICE','MYPOLICY')
```

例 4: この例では、相関 ID '1234' に一致する最初のメッセージを、サービス "MYSERVICE" に指定されたキューの先頭から検索して受信します。その際、ポリシー "MYPOLICY" を使用します。

```
VALUES MQRECEIVECLOB('MYSERVICE','MYPOLICY','1234')
```

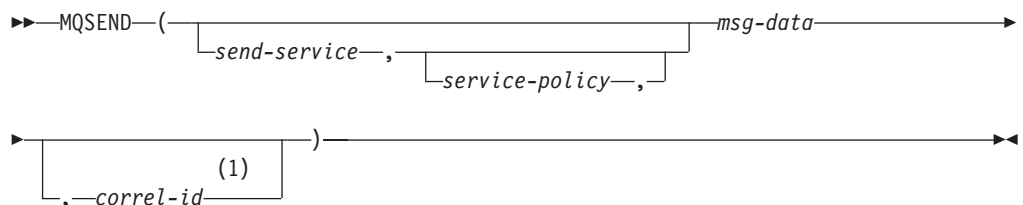
## MQSEND

MQSEND 関数は *msg-data* に含まれているデータを、*send-service* に指定された MQSeries のロケーションに送ります。その際、*service-policy* に定義されたサービス・ポリシーの品質を使用します。

オプションのユーザー定義メッセージ相関 ID は、*correl-id* を使用して指定できます。

この結果のデータ・タイプは VARCHAR(1) です。関数の結果は、成功した場合は '1' で失敗した場合は '0' です。

## 構文



## 注:

- 1 *service* および *policy* が指定されていない限り、*correl-id* を指定することはできません。

スキーマは、非トランザクション・メッセージ・キューイング関数の場合は DB2MQ、1 フェーズ・コミット・トランザクション MQ 関数の場合は DB2MQ1C です。

## 関数のパラメーター

### *msg-data*

MQSeries を介して送られるデータを含むストリング式。 VARCHAR ストリング式の最大サイズは 32 000 バイトで、CLOB ストリング式の最大サイズは 1M バイトです。

### *send-service*

メッセージの送信先である MQSeries 論理宛先を含むストリング。 *send-service* を指定する場合、DB2MQ.MQSERVICE 表に定義されたサービス・ポイントを参照させます。 サービス・ポイントとは、メッセージが送受信される論理上のエンドポイントです。サービス・ポイントの定義には、MQSeries キュー・マネージャーおよびキューの名前が含まれます。 *send-service* を指定しない場合、DB2.DEFAULT.SERVICE の値が使用されます。 *send-service* の最大サイズは 48 バイトです。

### *service-policy*

このメッセージの処理に使われる MQSeries サービス・ポリシーを含むストリング。 *service-policy* を指定する場合、DB2MQ.MQPOLICY 表に定義されたサービス・ポリシーを参照する必要があります。 サービス・ポリシーは、このメッセージング操作に適用されるサービス・オプションの一連の品質を定義します。これらのオプションには、メッセージ優先順位やメッセージ・パーシスタンスが含まれます。 *service-policy* を指定しない場合、デフォルト値 DB2.DEFAULT.POLICY が使用されます。 *service-policy* の最大サイズは 48 バイトです。

### *correl-id*

このメッセージに関連した相関 ID を含むオプション・ストリング。 *correl-id* は、要求/応答シナリオで要求と応答を関連付けるためにしばしば使用されます。これを指定しない場合、相関 ID は送信されません。 *correl-id* の最大サイズは 24 バイトです。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: この例では、ストリング "Testing 123" をデフォルト・サービス (DB2.DEFAULT.SERVICE) に送ります。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用し、相関 ID は使用しません。

```
VALUES MQSEND('Testing 123')
```



例 2: この例では、ストリング "Testing 345" をサービス "MYSERVICE" に送ります。その際、ポリシー "MYPOLICY" を使用し、相関 ID は使用しません。

```
VALUES MQSEND('MYSERVICE','MYPOLICY','Testing 345')
```

例 3: この例では、ストリング "Testing 678" をサービス "MYSERVICE" に送ります。その際、ポリシー "MYPOLICY" と相関 ID "TEST3" を使用します。

```
VALUES MQSEND('MYSERVICE','MYPOLICY','Testing 678','TEST3')
```

例 4: この例では、ストリング "Testing 901" をサービス "MYSERVICE" に送ります。その際、デフォルト・ポリシー (DB2.DEFAULT.POLICY) を使用し、相関 ID は使用しません。

```
VALUES MQSEND('MYSERVICE','Testing 901')
```

## MQSUBSCRIBE

MQSUBSCRIBE 関数は、指定されたトピックで公表された MQSeries メッセージに対する関心事を登録するために使用されます。この関数が正常に実行されると、パブリッシュ/サブスクライブ・サーバーは、トピックに一致するメッセージを、*subscriber-service* に定義されたサービス・ポイントに転送するようになります。

*subscriber-service* は、指定されたトピックに一致するメッセージの論理宛先を指定します。*topic* に一致するメッセージは *subscriber-service* に定義されたキューの中に入れられて、その後 MQREAD、MQRECEIVE、MQREADALL、または MQRECEIVEALL の呼び出し時に読み取りまたは受信することができます。詳しくは、<http://www.ibm.com/software/mqseries/> を参照してください。

この結果のデータ・タイプは VARCHAR(1) です。関数の結果は、成功した場合は '1' で失敗した場合は '0' です。

### 構文

```
MQSUBSCRIBE ( ( subscriber-service , service-policy ) topic )
```

スキーマは、非トランザクション・メッセージ・キューイング関数の場合は DB2MQ、1 フェーズ・コミット・トランザクション MQ 関数の場合は DB2MQ1C です。

### 関数のパラメーター

#### *subscriber-service*

*topic* に一致するメッセージの送信先である MQSeries 論理サブスクリプション・ポイントを含むストリング。*subscriber-service* を指定する場合、パブリッシャー・サービスのタイプ値が「S」の DB2MQ.MQPUBSUB 表に定義されているサブスクライバー・サービス・ポイントを参照する必要があります。

*subscriber-service* を指定しない場合、代わりに DB2.DEFAULT.SUBSCRIBER が使用されます。*subscriber-service* の最大サイズは 48 バイトです。

#### *service-policy*

メッセージの処理に使われる MQSeries サービス・ポリシーを含むストリング。*service-policy* を指定する場合、DB2MQ.MQPOLICY 表に定義されたポリ

シーを参照する必要があります。 サービス・ポリシーは、このメッセージング操作に適用されるサービス・オプションの一連の品質を定義します。これらのオプションには、メッセージ優先順位やメッセージ・パーシスタンスが含まれます。 *service-policy* を指定しない場合、代わりにデフォルト DB2.DEFAULT.POLICY が使用されます。 *service-policy* の最大サイズは 48 バイトです。

### *topic*

受信するメッセージのタイプを定義するストリング。指定されたトピックでパブリッシュされたメッセージのみが、このサブスクリプションによって受信されます。複数のサブスクリプションを同時に存在させることができます。 *topic* の最大サイズは 40 バイトです。1つのストリングに複数のトピックを指定することができます (最大で 40 バイトの長さ)。それぞれのトピックは、コロンで区切ります。例えば "t1:t2:the third topic" は、3つのトピック t1、t2、および "the third topic" がメッセージに関連付けられていることを示します。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: この例では、トピック "Weather" を含んでいるメッセージに対する関心事を登録します。デフォルト・サブスクライバー・サービス (DB2.DEFAULT.SUBSCRIBER) がサブスクライバーとして登録され、デフォルト・サービス・ポリシー (DB2.DEFAULT.POLICY) でサービスの品質を指定します。

```
VALUES MQSUBSCRIBE('Weather')
```

例 2: この例では、"Stocks" を含むメッセージに対する関心事をサブスクライバーが登録します。サブスクライバーはポリシー "BASIC-POLICY" を使用し、"PORTFOLIO-UPDATES" として登録します。

```
VALUES MQSUBSCRIBE('PORTFOLIO-UPDATES','BASIC-POLICY','Stocks')
```

## MQUNSUBSCRIBE

MQUNSUBSCRIBE 関数は、既存のメッセージ・サブスクリプションを登録抹消するために使用されます。

*subscriber-service*、*service-policy*、および *topic* を使って、取り消す予定のサブスクリプションを識別します。この関数が正常に実行されると、パブリッシュ/サブスクライブ・サーバーは、指定したサブスクリプションを除去するようになります。指

定された *topic* を持つメッセージは、*subscriber-service* に定義された論理宛先にもはや送信されなくなります。詳しくは、<http://www.ibm.com/software/mqseries/> を参照してください。

この結果のデータ・タイプは VARCHAR(1) です。関数の結果は、成功した場合は '1' で失敗した場合は '0' です。

## 構文

```
▶▶—MQUNSUBSCRIBE—▶▶
▶—(—subscriber-service—, —service-policy—, —topic—)—▶▶
```

スキーマは、非トランザクション・メッセージ・キューイング関数の場合は DB2MQ、1 フェーズ・コミット・トランザクション MQ 関数の場合は DB2MQ1C です。

## 関数のパラメーター

### *subscriber-service*

*subscriber-service* を指定する場合、パブリッシャー・サービスのタイプ値が「S」の DB2MQ.MQPUBSUB 表に定義されているサブスクライバー・サービス・ポイントを参照する必要があります。*subscriber-service* を指定しない場合、代わりに DB2.DEFAULT.SUBSCRIBER が使用されます。*subscriber-service* の最大サイズは 48 バイトです。

### *service-policy*

*service-policy* を指定する場合、DB2MQ.MQPOLICY 表に定義されたポリシーを参照する必要があります。サービス・ポリシーは、このメッセージング操作に適用されるサービス・オプションの一連の品質を定義します。*service-policy* を指定しない場合、デフォルトの DB2.DEFAULT.POLICY が使用されます。*service-policy* の最大サイズは 48 バイトです。

### *topic*

もはや受信しないメッセージの主題を指定するストリング。*topic* の最大サイズは 40 バイトです。1 つのストリングに複数のトピックを指定することができます (最大で 40 バイトの長さ)。それぞれのトピックは、コロンで区切ります。例えば "t1:t2:the third topic" は、3 つのトピック t1、t2、および "the third topic" がメッセージに関連付けられていることを示します。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

例 1: この例では、トピック "Weather" を含んでいるメッセージに対する関心を取り消します。デフォルト・サブスクライバー・サービス (DB2.DEFAULT.SUBSCRIBER) がアンサブスクライバーとして登録され、デフォルト・サービス・ポリシー (DB2.DEFAULT.POLICY) でサービスの品質を指定します。

```
VALUES MQUNSUBSCRIBE('Weather')
```

例 2: この例では、"Stocks" を含むメッセージに対する関心をサブスクライバーが取り消します。サブスクライバーはポリシー "BASIC-POLICY" を使用し、"PORTFOLIO-UPDATES" として登録されています。

```
VALUES MQUNSUBSCRIBE('PORTFOLIO-UPDATES','BASIC-POLICY','Stocks')
```

## セキュリティ・ルーチンおよびビュー

### AUTH\_GET\_INSTANCE\_AUTHID - インスタンス所有者の許可 ID の取得

AUTH\_GET\_INSTANCE\_AUTHID スカラー関数はインスタンス所有者の許可 ID を戻します

#### 構文

▶▶—AUTH\_GET\_INSTANCE\_AUTHID—(—)—▶▶

スキーマは SYSPROC です。

#### 許可

AUTH\_GET\_INSTANCE\_AUTHID スカラー関数に対する EXECUTE 特権。

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

次の例は、DB2 コマンド行プロセッサ (CLP) を使用してインスタンス所有者の許可 ID を取得する方法を示しています。

```
db2 "values SYSPROC.AUTH_GET_INSTANCE_AUTHID()"
```

このコマンドの出力例は、次のとおりです。

1

-----  
ZURBIE

1 record(s) selected.

### 使用上の注意

一般的な構成では、インスタンス所有者アカウントが SYSADM グループのメンバーとして含まれます。このため DB2 バージョン 9.7 より前では、インスタンス所有者アカウントのもとで実行されるアプリケーションはデータベースに対する無制限の権限を持っていました。DB2 バージョン 9.7 以降は、SYSADM 権限を持つユーザーは暗黙的な DBADM 権限を持たなくなりました。その結果、インスタンス所有者アカウントのもとで実行されるアプリケーションでは、SYSADM 権限のスコープに含まれなくなった操作を実行すると SQL1092N、SQL0551N、SQL0552N などの許可エラーが発生する可能性があります。

**UPGRADE DATABASE** コマンドと **RESTORE DATABASE** コマンド (以前のデータベース用) は、DBADM 権限を SYSADM グループに付与しますが、新しいバージョン 9.7 データベースではこれが該当しません。

インスタンス所有者の許可 ID が保持している権限のリストを取得するには、次の手順に従います。

1. SYSPROC.AUTH\_GET\_INSTANCE\_AUTHID() スカラー関数を使用して、インスタンス所有者の許可 ID を判別します。以下に例を示します。

```
db2 "VALUES SYSPROC.AUTH_GET_INSTANCE_AUTHID()"
```

このコマンドにより、以下が戻されます。

```
1
-----
BOB
```

1 record(s) selected.

2. この許可 ID が持つ権限のリストを取得します。以下に例を示します。

```
SELECT * FROM
TABLE (SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID ('BOB', 'U') ) AS T
ORDER BY AUTHORITY
```

3. 必要に応じて、欠落している権限を付与します。以下に例を示します。

```
GRANT DBADM ON DATABASE TO USER BOB
```

### 戻される情報

表 187. AUTH\_GET\_INSTANCE\_AUTHID スカラー関数によって戻される情報

列名	データ・タイプ	説明
InstanceAuthId	VARCHAR(128)	インスタンス所有者の許可 ID。

### AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID

AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID 表関数は、データベース構成ファイルにあるか、許可 ID に直接付与されたか、あるいはグループまたはロールを介して間接的に付与された許可 ID によって保持されているすべての権限を戻します。

## 構文

```
▶▶—AUTH_LIST_AUTHORITIES_FOR_AUTHID—(—authid—,—authidtype—)————▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *authid*

照会対象の許可 ID を指定する、タイプ VARCHAR(128) の入力引数。許可 ID はユーザー、グループ、またはロールです。*authid* が NULL または空ストリングである場合、空の結果表が戻されます。

### *authidtype*

照会対象の許可 ID を指定する、タイプ VARCHAR(1) の入力引数。*authidtype* が存在しないか、NULL または空ストリングである場合、空の結果表が戻されます。*authidtype* に考えられる値は以下のとおりです。

- G: グループ
- R: ロール
- U: ユーザー

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 戻される情報

表 188. AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID について戻される情報

列名	データ・タイプ	説明
AUTHORITY	VARCHAR(128)	許可 ID によって保持される権限。
D_USER	CHAR(1)	<i>authidtype</i> がユーザー (U) の場合に、 <i>authid</i> に直接付与される権限。 <i>authidtype</i> がグループ (G) またはロール (R) の場合、値は適用外 (*) です。 <ul style="list-style-type: none"><li>• N = 保有しない</li><li>• Y= 保有する</li><li>• * = 適用外</li></ul>

表 188. AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID について戻される情報 (続き)

列名	データ・タイプ	説明
D_GROUP	CHAR(1)	<p><i>authidtype</i> がグループ (G) の場合に、<i>authid</i> に直接付与される権限、または <i>authidtype</i> がユーザー (U) の場合に <i>authid</i> が属するグループに付与される権限。 <i>authidtype</i> がロール (R) の場合、値は適用外 (*) です。</p> <ul style="list-style-type: none"> <li>• N = 保有しない</li> <li>• Y= 保有する</li> <li>• * = 適用外</li> </ul>
D_PUBLIC	CHAR(1)	<p><i>authidtype</i> がユーザー (U) またはグループ (G) の場合に、PUBLIC という <i>authid</i> に直接付与される権限。 <i>authidtype</i> がロール (R) の場合、値は適用外 (*) です。</p> <ul style="list-style-type: none"> <li>• N = 保有しない</li> <li>• Y= 保有する</li> <li>• * = 適用外</li> </ul>
ROLE_USER	CHAR(1)	<p><i>authidtype</i> がユーザー (U) の場合に、<i>authid</i> に付与されたロールに直接付与される権限。 <i>authidtype</i> がグループ (G) またはロール (R) の場合、値は適用外 (*) です。ロールはロール階層の一部である可能性があります。</p> <ul style="list-style-type: none"> <li>• N = 保有しない</li> <li>• Y= 保有する</li> <li>• * = 適用外</li> </ul>
ROLE_GROUP	CHAR(1)	<p><i>authidtype</i> がグループ (G) の場合に、<i>authid</i> に付与されたロールに直接付与される権限。 <i>authidtype</i> がユーザー (U) またはロール (R) の場合、値は適用外 (*) です。ロールはロール階層の一部である可能性があります。</p> <ul style="list-style-type: none"> <li>• N = 保有しない</li> <li>• Y= 保有する</li> <li>• * = 適用外</li> </ul>
ROLE_PUBLIC	CHAR(1)	<p><i>authidtype</i> がユーザー (U) またはグループ (G) の場合に、PUBLIC という <i>authid</i> に付与されたロールに直接付与される権限。 <i>authidtype</i> がロール (R) の場合、値は適用外 (*) です。ロールはロール階層の一部である可能性があります。</p> <ul style="list-style-type: none"> <li>• N = 保有しない</li> <li>• Y= 保有する</li> <li>• * = 適用外</li> </ul>
D_ROLE	CHAR(1)	<p>ロールに付与された権限、またはそのロールに付与されているロールに付与された権限。 <i>authidtype</i> がユーザー (U) またはグループ (G) の場合、値は適用外 (*) です。ロールはロール階層の一部である可能性があります。</p> <ul style="list-style-type: none"> <li>• N = 保有しない</li> <li>• Y= 保有する</li> <li>• * = 適用外</li> </ul>

## 例

特殊グループ PUBLIC を介してデフォルトで BIND、CONNECT、CREATETAB、および IMPLICIT\_SCHEMA 特権を保持するユーザー ALICE について考慮します。ALICE は、SYSADM、SYSCTRL、および SYSMOINT というシステム権限を持つグループ ADMIN1 のメンバーです。また、DBADM 権限を持つグループ ADMIN2 のメンバーでもあります。さらに、DBADM および SECADM データベース権限も付与されています。ロール R1 が ALICE に付与されました。LOAD 権限がロール R1 に付与されました。ロール R2 がグループ ADMIN1 に付与されました。CREATE\_NOT\_FENCED\_ROUTINE 権限がロール R2 に付与されました。

例 1: ユーザー ALICE が自分自身に直接、またはグループ PUBLIC またはロールを介して間接に付与したすべての権限を取得します。

```
SELECT AUTHORITY, D_USER, D_GROUP, D_PUBLIC, ROLE_USER, ROLE_GROUP, ROLE_PUBLIC, D_ROLE
FROM TABLE (SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID ('ALICE', 'U') ) AS T
ORDER BY AUTHORITY
```

AUTHORITY	D_USER	D_GROUP	D_PUBLIC	ROLE_USER	ROLE_GROUP	ROLE_PUBLIC	D_ROLE
ACCESSCTRL	N	N	N	N	N	N	*
BINDADD	N	N	Y	N	N	N	*
CONNECT	N	N	Y	N	N	N	*
CREATE_EXTERNAL_ROUTINE	N	N	N	N	N	N	*
CREATE_NOT_FENCED_ROUTINE	N	N	N	N	Y	N	*
CREATETAB	N	N	Y	N	N	N	*
DATAACCESS	N	N	N	N	N	N	*
DBADM	Y	Y	N	N	N	N	*
EXPLAIN	N	N	N	N	N	N	*
IMPLICIT_SCHEMA	N	N	Y	N	N	N	*
LOAD	N	N	N	Y	N	N	*
QUIESCE_CONNECT	N	N	N	N	N	N	*
SECADM	Y	N	N	N	N	N	*
SQLADM	N	N	N	N	N	N	*
SYSADM	*	Y	*	*	*	*	*
SYSCTRL	*	Y	*	*	*	*	*
SYSMOINT	*	Y	*	*	*	*	*
SYSMON	*	N	*	*	*	*	*
WLMADM	N	N	N	N	N	N	*

例 2: グループ ADMIN1 がそれ自身に直接、または PUBLIC またはロールを介して間接に付与したすべての権限を取得します。

```
SELECT AUTHORITY, D_USER, D_GROUP, D_PUBLIC, ROLE_USER, ROLE_GROUP, ROLE_PUBLIC, D_ROLE
FROM TABLE (SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID ('ADMIN1', 'G') ) AS T
ORDER BY AUTHORITY
```

AUTHORITY	D_USER	D_GROUP	D_PUBLIC	ROLE_USER	ROLE_GROUP	ROLE_PUBLIC	D_ROLE
ACCESSCTRL	*	N	*	*	N	*	*
BINDADD	*	N	*	*	N	*	*
CONNECT	*	N	*	*	N	*	*
CREATE_EXTERNAL_ROUTINE	*	N	*	*	N	*	*
CREATE_NOT_FENCED_ROUTINE	*	N	*	*	Y	*	*
CREATETAB	*	N	*	*	N	*	*
DATAACCESS	*	N	*	*	N	*	*
DBADM	*	N	*	*	N	*	*
EXPLAIN	*	N	*	*	N	*	*
IMPLICIT_SCHEMA	*	N	*	*	N	*	*
LOAD	*	N	*	*	N	*	*
QUIESCE_CONNECT	*	N	*	*	N	*	*
SECADM	*	N	*	*	N	*	*
SQLADM	*	N	*	*	N	*	*
SYSADM	*	Y	*	*	*	*	*
SYSCTRL	*	Y	*	*	*	*	*
SYSMOINT	*	Y	*	*	*	*	*
SYSMON	*	N	*	*	*	*	*
WLMADM	*	N	*	*	N	*	*



例 3: 特殊グループ PUBLIC がそれ自身に直接、またはロールを介して間接に付与したすべての権限を取得します。

```
SELECT AUTHORITY, D_USER, D_GROUP, D_PUBLIC, ROLE_USER, ROLE_GROUP, ROLE_PUBLIC, D_ROLE
FROM TABLE (SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID ('PUBLIC', 'G')) AS T
ORDER BY AUTHORITY
```

1	D_USER	D_GROUP	D_PUBLIC	ROLE_USER	ROLE_GROUP	ROLE_PUBLIC	D_ROLE
ACCESSCTRL	*	*	N	*	*	N	*
BINDADD	*	*	Y	*	*	N	*
CONNECT	*	*	Y	*	*	N	*
CREATE_EXTERNAL_ROUTINE	*	*	N	*	*	N	*
CREATE_NOT_FENCED_ROUTINE	*	*	N	*	*	N	*
CREATETAB	*	*	Y	*	*	N	*
DATAACCESS	*	*	N	*	*	N	*
DBADM	*	*	N	*	*	N	*
EXPLAIN	*	*	N	*	*	N	*
IMPLICIT_SCHEMA	*	*	Y	*	*	N	*
LOAD	*	*	N	*	*	N	*
QUIESCE_CONNECT	*	*	N	*	*	N	*
SECADM	*	*	N	*	*	N	*
SQLADM	*	*	N	*	*	N	*
SYSADM	*	*	*	*	*	*	*
SYSCTRL	*	*	*	*	*	*	*
SYSMAINT	*	*	*	*	*	*	*
SYSMON	*	*	*	*	*	*	*
WLMADM	*	*	N	*	*	N	*

例 4: ロール R1 がそれ自身に直接、またはロールを介して間接に付与したすべての権限を取得します。この場合、ロール R2 もロール R1 に付与されたことを考慮してください。

```
SELECT AUTHORITY, D_USER, D_GROUP, D_PUBLIC, ROLE_USER, ROLE_GROUP, ROLE_PUBLIC, D_ROLE
FROM TABLE (SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID ('R1', 'R')) AS T
ORDER BY AUTHORITY
```

AUTHORITY	D_USER	D_GROUP	D_PUBLIC	ROLE_USER	ROLE_GROUP	ROLE_PUBLIC	D_ROLE
ACCESSCTRL	*	*	*	*	*	*	N
BINDADD	*	*	*	*	*	*	N
CONNECT	*	*	*	*	*	*	N
CREATE_EXTERNAL_ROUTINE	*	*	*	*	*	*	N
CREATE_NOT_FENCED_ROUTINE	*	*	*	*	*	*	Y
CREATETAB	*	*	*	*	*	*	N
DATAACCESS	*	*	*	*	*	*	N
DBADM	*	*	*	*	*	*	N
EXPLAIN	*	*	*	*	*	*	N
IMPLICIT_SCHEMA	*	*	*	*	*	*	N
LOAD	*	*	*	*	*	*	Y
QUIESCE_CONNECT	*	*	*	*	*	*	N
SECADM	*	*	*	*	*	*	N
SYSADM	*	*	*	*	*	*	*
SQLADM	*	*	*	*	*	*	N
SYSCTRL	*	*	*	*	*	*	*
SYSMAINT	*	*	*	*	*	*	*
SYSMON	*	*	*	*	*	*	*
WLMADM	*	*	*	*	*	*	N

### 使用上の注意

AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID 表関数の出力は *authidtype* に応じて異なります。例えば、USER の *authidtype* の場合、*authid* が以下の手段で保持しているすべての権限を戻します。

- *authid* に直接付与された
- *authid* が属するグループ (またはそのグループに付与されたロール) に付与された
- *authid* に付与されたロール (またはそのロールに付与されたロール) に付与された

- PUBLIC (または PUBLIC に付与されたロール) に付与された

## AUTH\_LIST\_GROUPS\_FOR\_AUTHID 表関数 - 指定の許可 ID のグループ・メンバーシップ・リストの検索

AUTH\_LIST\_GROUPS\_FOR\_AUTHID 表関数は、指定の許可 ID がメンバーとして所属するグループのリストを戻します。

### 構文

```
▶▶—AUTH_LIST_GROUPS_FOR_AUTHID—(—authid—)—————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

*authid*

照会対象の許可 ID を指定する、タイプ VARCHAR(128) の入力引数。許可 ID は、ユーザーのみを表すことができます。 *authid* が存在しない、NULL である、または空ストリングの場合、空の結果表が戻されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

AMY が属するすべてのグループを検索します。

```
SELECT * FROM TABLE (SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID('AMY')) AS T
```

以下はこの照会の出力例です。

```
GROUP
-----
BUILD
PDXDB2
```

2 record(s) selected.

### 使用上の注意

以下の理由により、戻されるグループ情報は予想と異なる場合があります。

- Windows Active Directory 環境の場合、データベース・マネージャーについて次のことが言えます。

- ローカル・グループ内での 1 レベルのグループ・ネストはサポートしているものの、ローカル・グループ内でのドメイン・ローカル・グループのネストは例外となります。例えば、*authid* がグローバル・グループ G1 に属し、G1 がローカル・グループ L1 に属する場合、*authid* のグループとしてローカル・グループ L1 が戻されます。しかし、*authid* がドメイン・ローカル・グループ DL1 に属し、DL1 がローカル・グループ L1 に属する場合、*authid* のグループ情報は戻されません。
- グローバル・グループのネストはサポートしていません。例えば、*authid* がグローバル G2 に属し、G2 がグローバル G3 に属する場合、G2 のみが *authid* のグループとして戻されます。
- ユーザーの属するグループを列挙するために使用される Windows セキュリティ機能は、レジストリー変数 **DB2\_GRP\_LOOKUP** によって指定されます。
- 特定のドメインに属する許可 ID に関し、*authid* の一部としてドメインを指定せず、かつローカルとドメインの両方に同じ名前の *authid* が存在している場合には、ローカル許可 ID のグループが戻されます。
- AUTH\_LIST\_GROUPS\_FOR\_AUTHID への呼び出しが接続済みユーザーと同じ *authid* に対するものである場合、接続済みユーザーのグループが戻されます。例えば、AMY がローカル・ユーザーおよびドメイン・ユーザーとして存在し、ドメイン・ユーザー AMY がデータベースに接続済みである場合、AUTH\_LIST\_GROUPS\_FOR\_AUTHID はドメイン AMY が所属するグループを戻します。

## 戻される情報

表 189. AUTH\_LIST\_GROUPS\_FOR\_AUTHID 表関数によって戻される情報

列名	データ・タイプ	説明
GROUP	VARCHAR(128)	許可 ID を保有するグループ

## AUTH\_LIST\_ROLES\_FOR\_AUTHID 関数 - ロールのリストを戻す

AUTH\_LIST\_ROLES\_FOR\_AUTHID 関数は、指定の許可 ID がメンバーになっているロールのリストを戻します。

### 構文

▶▶—AUTH\_LIST\_ROLES\_FOR\_AUTHID—(—*authid*—,—*authidtype*—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### *authid*

照会対象の許可 ID を指定する、タイプ VARCHAR(128) の入力引数。許可 ID はユーザー、グループ、またはロールです。*authid* が NULL または空ストリングである場合、空の結果表が戻されます。

#### *authidtype*

照会対象の許可 ID を指定する、タイプ VARCHAR(1) の入力引数。*authidtype*

が存在しないか、NULL または空ストリングである場合、空の結果表が戻されます。 *authidtype* に考えられる値は以下のとおりです。

- G: グループ
- R: ロール
- U: ユーザー

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 戻される情報

表 190. AUTH\_LIST\_ROLES\_FOR\_AUTHID の結果セット

列名	データ・タイプ	説明
GRANTOR	VARCHAR(128)	ロールの認可者。
GRANTORTYPE	CHAR(1)	認可者のタイプ: • U = 認可者は個々のユーザー
GRANTEE	VARCHAR(128)	ユーザーがロールを付与しました。
GRANTEETYPE	CHAR(1)	被認可者のタイプ: • G = GRANTEE はグループ。 • R = GRANTEE はロール。 • U= 被認可者はユーザー。
ROLENAME	VARCHAR(128)	グループまたは別のロールを介して許可 ID に直接または間接に付与されたロールの名前。
CREATE_TIME	TIMESTAMP	ロールが作成された時刻。
ADMIN	CHAR(1)	ロールを付与したり、ロールを取り消したり、またはロールに関するコメントを付けたりする特権。 • N = 保有しない • Y= 保有する

## 例

ロール INTERN をロール DOCTOR に、さらにロール DOCTOR をロール SPECIALIST に付与してから、ロール SPECIALIST をユーザー ALICE に付与することについて考慮します。 ALICE はグループ HOSPITAL に属しており、ロール

EMPLOYEE はグループ HOSPITAL に付与されます。また、ALICE は特殊グループ PUBLIC にも属しており、ロール PATIENTS が PUBLIC に付与されています。

例 1: ユーザー ALICE に付与されたすべてのロールを取得します。

```
SELECT GRANTOR, GRANTORTYPE, GRANTEE, GRANTEETYPE, ROLENAME,
       CREATE_TIME, ADMIN
FROM TABLE (SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID ('ALICE', 'U') ) AS T
```

以下はこの照会の出力例です。

GRANTOR	GRANTORTYPE	GRANTEE	GRANTEETYPE	ROLENAME	CREATE_TIME	ADMIN
ZURBIE	U	DOCTOR	R	INTERN	2006-08-01-15.09.58.537399	N
ZURBIE	U	SPECIALIST	R	DOCTOR	2006-08-01-15.10.04.540660	N
ZURBIE	U	ALICE	U	SPECIALIST	2006-08-01-15.10.08.776218	N
ZURBIE	U	HOSPITAL	G	EMPLOYEE	2006-08-01-15.10.14.277576	N
ZURBIE	U	PUBLIC	G	PATIENTS	2006-08-01-15.10.18.878609	N

5 record(s) selected.

例 2: グループ HOSPITAL に付与されたすべてのロールを取得します。

```
SELECT GRANTOR, GRANTORTYPE, GRANTEE, GRANTEETYPE, ROLENAME,
       CREATE_TIME, ADMIN
FROM TABLE (SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID ('HOSPITAL', 'G') ) AS T
```

以下はこの照会の出力例です。

GRANTOR	GRANTORTYPE	GRANTEE	GRANTEETYPE	ROLENAME	CREATE_TIME	ADMIN
ZURBIE	U	HOSPITAL	G	EMPLOYEE	2006-08-01-15.10.14.277576	N

1 record(s) selected.

例 3: ロール SPECIALIST に付与されたすべてのロールを取得します。

```
SELECT GRANTOR, GRANTORTYPE, GRANTEE, GRANTEETYPE, ROLENAME,
       CREATE_TIME, ADMIN
FROM TABLE (SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID ('SPECIALIST', 'R') ) AS T
```

以下はこの照会の出力例です。

GRANTOR	GRANTORTYPE	GRANTEE	GRANTEETYPE	ROLENAME	CREATE_TIME	ADMIN
ZURBIE	U	DOCTOR	R	INTERN	2006-08-01-15.09.58.537399	N
ZURBIE	U	SPECIALIST	R	DOCTOR	2006-08-01-15.10.04.540660	N

2 record(s) selected.

例 4: グループ PUBLIC に付与されたすべてのロールを取得します。

```
SELECT GRANTOR, GRANTORTYPE, GRANTEE, GRANTEETYPE, ROLENAME,
       CREATE_TIME, ADMIN
FROM TABLE (SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID ('PUBLIC', 'G') ) AS T
```

以下はこの照会の出力例です。

GRANTOR	GRANTORTYPE	GRANTEE	GRANTEETYPE	ROLENAME	CREATE_TIME	ADMIN
ZURBIE	U	PUBLIC	G	PATIENTS	2006-08-01-15.10.18.878609	N

1 record(s) selected.

## 使用上の注意

AUTH\_LIST\_ROLES\_FOR\_AUTHID 表関数の出力は AUTHIDTYPE に応じて異なります。

- ユーザーの場合、別のロールまたはそのユーザーが属する別のグループ (または PUBLIC) を介して直接または間接にユーザーに付与されたロールを戻します。
- グループの場合、別のロールを介して直接または間接にグループに付与されたロールを戻します。
- ロールの場合、別のロールを介して直接または間接にロールに付与されたロールを戻します。

## AUTHORIZATIONIDS 管理ビュー - 許可 ID およびタイプの検索

AUTHORIZATIONIDS 管理ビューは、現在接続しているサーバーのデータベース・カタログに GRANT ステートメントの結果として存在するすべてのユーザー、ロール、グループの一覧を返します。

ビューにより返されるそれぞれの許可 ID と関連タイプは、最低 1 つの特権、権限、またはロール・メンバーシップを有しています。現在、いずれかのグループのメンバーになっているユーザーは、結果に含まれません。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- AUTHORIZATIONIDS 管理ビューに対する SELECT 特権
- AUTHORIZATIONIDS 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

特権または権限を付与された許可 ID すべてを、それらのタイプとともに検索します。

```
SELECT * FROM SYSIBMADM.AUTHORIZATIONIDS
```

以下はこの照会の出力例です。

AUTHID	AUTHIDTYPE
PUBLIC	G
JESSICAE	U
DOCTOR	R

3 record(s) selected.

## 戻される情報

表 191. AUTHORIZATIONIDS 管理ビューによって戻される情報

列名	データ・タイプ	説明
AUTHID	VARCHAR(128)	特権または権限を明示的に付与された許可 ID。
AUTHIDTYPE	CHAR(1)	許可 ID タイプ。 • U: ユーザー • R: ロール • G: グループ

## OBJECTOWNERS 管理ビュー - オブジェクト所有権情報の検索

OBJECTOWNERS 管理ビューは、現在接続中のデータベースのシステム・カタログに定義されているオブジェクトを所有するタイプ USER のすべての許可 ID のすべてのオブジェクト所有権情報を戻します。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- OBJECTOWNERS 管理ビューに対する SELECT 特権
- OBJECTOWNERS 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

オブジェクト・スキーマ 'THERESAX' のすべてのオブジェクト所有権情報を検索します。

```
SELECT SUBSTR(OWNER,1,10) AS OWNER, OWNERTYPE,  
       SUBSTR(OBJECTNAME,1,30) AS OBJECTNAME,  
       SUBSTR(OBJECTSCHEMA,1,10) AS OBJECTSCHEMA, OBJECTTYPE  
FROM SYSIBMADM.OBJECTOWNERS WHERE OBJECTSCHEMA='THERESAX'
```

以下はこの照会の出力例です。

OWNER	OWNERTYPE	OBJECTNAME	OBJECTSCHEMA	OBJECTTYPE
THERESAX	U	MIN_SALARY	THERESAX	TRIGGER
THERESAX	U	POLICY_IR	SYSTOOLS	TRIGGER
THERESAX	U	CUSTOMER	THERESAX	XML SCHEMA
THERESAX	U	DB2DETAILDEADLOCK		EVENTMONITORS
THERESAX	U	SAMPSEQUENCE	THERESAX	SEQUENCE
THERESAX	U	SQLQ0F00	NULLID	PACKAGE

```
...
THERESAX U HI_OBJ_UNIQ SYSTOOLS TABLE CONSTRAINT
```

257 record(s) selected.

## 戻される情報

表 192. OBJECTOWNERS 管理ビューによって戻される情報

列名	データ・タイプ	説明
OWNER	VARCHAR(128)	このオブジェクトを所有する許可 ID。
OWNERTYPE	VARCHAR(1)	許可 ID タイプ。 <ul style="list-style-type: none"> <li>• U: ユーザー</li> <li>• S: システム</li> </ul>
OBJECTNAME	VARCHAR(128)	object_name - オブジェクト名 モニター・エレメント
OBJECTSCHEMA	VARCHAR(128)	object_schema - オブジェクト・スキーマ・モニター・エレメント
OBJECTTYPE	VARCHAR(24)	データベース・オブジェクト・タイプ。

## PRIVILEGES 管理ビュー - 特権情報の検索

PRIVILEGES 管理ビューは、現在接続中のデータベースのシステム・カタログに定義されているすべての許可 ID に関するすべての明示的な特権を戻します。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- PRIVILEGES 管理ビューに対する SELECT 特権
- PRIVILEGES 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

すべての許可 ID に対して付与された特権を、オブジェクト名、スキーマ、およびタイプと共に検索します。

```
SELECT AUTHID, PRIVILEGE, OBJECTNAME, OBJECTSCHEMA, OBJECTTYPE
FROM SYSIBMADM.PRIVILEGES
```

以下はこの照会の出力例です。



```

AUTHID      PRIVILEGE  OBJECTNAME      OBJECTSCHEMA    OBJECTTYPE
-----
JESSICAE    EXECUTE    SQLEOF00        NULLID          PACKAGE
PUBLIC      EXECUTE    SYSSH201        NULLID          PACKAGE
JESSICAE    EXECUTE    SYSSH202        NULLID          PACKAGE
PUBLIC      EXECUTE    SYSSH202        NULLID          PACKAGE
DOCTOR      EXECUTE    PKG0123         NULLID          PACKAGE
...
PUBLIC      EXECUTE    SQL051109185227800  SYSPROC        FUNCTION
JESSICAE    EXECUTE    SQL051109185227801  SYSPROC        FUNCTION
PUBLIC      EXECUTE    SQL051109185227801  SYSPROC        FUNCTION
JESSICAE    EXECUTE    SQL051109185227838  SYSPROC        FUNCTION
PUBLIC      EXECUTE    SQL051109185227838  SYSPROC        FUNCTION
...
PUBLIC      EXECUTE    LIST_SRVR_TYPES    SYSPROC        PROCEDURE
PUBLIC      EXECUTE    LIST_SRVR_VERSIONS  SYSPROC        PROCEDURE
PUBLIC      EXECUTE    LIST_WRAP_OPTIONS   SYSPROC        PROCEDURE
PUBLIC      EXECUTE    LIST_SRVR_OPTIONS   SYSPROC        PROCEDURE
...
SYSTEM      CREATEIN   POLICY_UNQ        SYSTOOLS       INDEX
PUBLIC      UPDATE    COLUMNS          NULLID          SCHEMA
PUBLIC      UPDATE    COLGROUPS         SYSSTAT        VIEW
PUBLIC      UPDATE    COLGROUPS         SYSSTAT        VIEW
...

```

## 戻される情報

表 193. PRIVILEGES 管理ビューによって戻される情報

列名	データ・タイプ	説明
AUTHID	VARCHAR(128)	この特権を明示的に付与された許可 ID。
AUTHIDTYPE	CHAR(1)	許可 ID タイプ。 <ul style="list-style-type: none"> <li>U: ユーザー</li> <li>R: ロール</li> <li>G: グループ</li> </ul>
PRIVILEGE	VARCHAR(11)	この許可 ID に明示的に認可された特権。
GRANTABLE	VARCHAR(1)	特権が付与可能か否かを示します。 <ul style="list-style-type: none"> <li>Y: 付与可能</li> <li>N: 付与不能</li> </ul>
OBJECTNAME	VARCHAR(128)	object_name - オブジェクト名モニター・エレメント
OBJECTSCHEMA	VARCHAR(128)	object_schema - オブジェクト・スキーマ・モニター・エレメント
OBJECTTYPE	VARCHAR(24)	データベース・オブジェクト・タイプ。

## スナップショット・ルーチンおよびビュー

### APPL\_PERFORMANCE 管理ビュー - アプリケーションで選択される行のパーセンテージの検索

APPL\_PERFORMANCE 管理ビューは、アプリケーションによって選択される行のパーセンテージに関する情報を表示します。

戻されるのは現在接続されているデータベースのすべてのデータベース・パーティションに関する情報です。このビューは、大量の表スキャンを実行しているアプリケーションや、問題が発生する可能性のある照会を探すのに使用できます。

スキーマは SYSIBMADM です。

#### 許可

以下のいずれかの権限が必要です。

- APPL\_PERFORMANCE 管理ビューに対する SELECT 特権
- APPL\_PERFORMANCE 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

#### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

#### 例

アプリケーション・パフォーマンスのレポートを検索します。

```
SELECT SNAPSHOT_TIMESTAMP, SUBSTR(AUTHID,1,10) AS AUTHID,  
       SUBSTR(APPL_NAME,1,10) AS APPL_NAME, AGENT_ID,  
       PERCENT_ROWS_SELECTED, DBPARTITIONNUM  
FROM SYSIBMADM.APPL_PERFORMANCE
```

以下はこの照会の出力例です。

```
SNAPSHOT_TIMESTAMP      AUTHID      APPL_NAME ...  
-----  
2006-01-07-17.01.15.966668 JESSICAE  db2bp.exe ...  
2006-01-07-17.01.15.980278 JESSICAE  db2taskd ...  
2006-01-07-17.01.15.980278 JESSICAE  db2bp.exe ...  
...  
3 record(s) selected.  ...
```

この照会の出力 (続き)。

```

... AGENT_ID          PERCENT_ROWS_SELECTED DBPARTITIONNUM
... -----
...              67              -              1
...              68              -              0
...              67              57.14          0
...

```

## 戻される情報

表 194. APPL\_PERFORMANCE 管理ビューによって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
AUTHID	VARCHAR(128)	auth_id - 許可 ID
APPL_NAME	VARCHAR(256)	appl_name - アプリケーション名
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
PERCENT_ROWS_SELECTED	DECIMAL(5,2)	ディスクから読み取られ、実際にアプリケーションに戻された行のパーセント。 注: 表示されるパーセンテージは、100.00 % より大きくなることはありません。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## APPLICATIONS 管理ビュー - 接続されているデータベース・アプリケーション情報の検索

APPLICATIONS 管理ビューは、接続されているデータベース・アプリケーションに関する情報を戻します。

ビューは **LIST APPLICATIONS SHOW DETAIL CLP** コマンドの SQL インターフェースです。ただし、現在接続されているデータベースに限ります。その情報は、SNAPAPPL\_INFO 管理ビューに基づきます。

スキーマは SYSIBMADM です。

## 許可

以下のいずれかの権限が必要です。

- APPLICATIONS 管理ビューに対する SELECT 特権
- APPLICATIONS 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

例 1: 単一パーティション・データベース SAMPLE 内のすべてのアクティブ・アプリケーションに関する情報をリストします。

```
SELECT AGENT_ID, SUBSTR(APPL_NAME,1,10) AS APPL_NAME, AUTHID,
       APPL_STATUS FROM SYSIBADM.APPLICATIONS WHERE DB_NAME = 'SAMPLE'
```

以下はこの照会の出力例です。

```
AGENT_ID          APPL_NAME  AUTHID    APPL_STATUS
-----
                23 db2bp.exe JESSICAE  UOWEXEC
```

1 record(s) selected.

例 2: 複数パーティション・データベース SAMPLE の、データベース・パーティション 0 上のアプリケーションごとのエージェントの数をリストします。

```
SELECT SUBSTR(APPL_NAME, 1, 10) AS APPL_NAME, COUNT(*) AS NUM
       FROM SYSIBADM.APPLICATIONS WHERE DBPARTITIONNUM = 0
       AND DB_NAME = 'SAMPLE' GROUP BY APPL_NAME
```

以下はこの照会の出力例です。

```
APPL_NAME  NUM
-----
db2bp.exe    3
javaw.exe   1
```

2 record(s) selected.

## 使用上の注意

ビューは **GLOBAL** 構文 (CLP で使用可能な構文) をサポートしていません。しかし、データベース・パーティションのデータはすべてビューから戻されるので、集約は SQL 集約関数を使うことによって行えます。

## 戻される情報

表 195. APPLICATIONS 管理ビューによって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。

表 195. APPLICATIONS 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
CLIENT_DB_ALIAS	VARCHAR(128)	client_db_alias - アプリケーションで使用するデータベース別名
DB_NAME	VARCHAR(128)	db_name - データベース名
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
APPL_NAME	VARCHAR(256)	appl_name - アプリケーション名
AUTHID	VARCHAR(128)	auth_id - 許可 ID
APPL_ID	VARCHAR(128)	appl_id - アプリケーション ID
APPL_STATUS	VARCHAR(22)	<p>appl_status - アプリケーション状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• BACKUP</li> <li>• COMMIT_ACT</li> <li>• COMP</li> <li>• CONNECTED</li> <li>• CONNECTPEND</li> <li>• CREATE_DB</li> <li>• DECOUPLED</li> <li>• DISCONNECTPEND</li> <li>• INTR</li> <li>• IOERROR_WAIT</li> <li>• LOAD</li> <li>• LOCKWAIT</li> <li>• QUIESCE_TABLESPACE</li> <li>• RECOMP</li> <li>• REMOTE_RQST</li> <li>• RESTART</li> <li>• RESTORE</li> <li>• ROLLBACK_ACT</li> <li>• ROLLBACK_TO_SAVEPOINT</li> <li>• TEND</li> <li>• THABRT</li> <li>• THCOMT</li> <li>• TPREP</li> <li>• UNLOAD</li> <li>• UOWEXEC</li> <li>• UOWWAIT</li> <li>• WAITFOR_REMOTE</li> </ul>

表 195. APPLICATIONS 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
STATUS_CHANGE_TIME	TIMESTAMP	status_change_time - アプリケーション状況変更時刻
SEQUENCE_NO	VARCHAR(4)	sequence_no - シーケンス番号
CLIENT_PRDID	VARCHAR(128)	client_prdid - クライアント製品/バージョン ID
CLIENT_PID	BIGINT	client_pid - クライアント・プロセス ID

表 195. APPLICATIONS 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
CLIENT_PLATFORM	VARCHAR(12)	<p>client_platform - クライアント・オペレーティング・プラットフォーム。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• AIX</li> <li>• AIX64</li> <li>• AS400_DRDA</li> <li>• DOS</li> <li>• DYNIX</li> <li>• HP</li> <li>• HP64</li> <li>• HPIA</li> <li>• HPIA64</li> <li>• LINUX</li> <li>• LINUX390</li> <li>• LINUXIA64</li> <li>• LINUXPPC</li> <li>• LINUXPPC64</li> <li>• LINUXX8664</li> <li>• LINUXZ64</li> <li>• MAC</li> <li>• MVS_DRDA</li> <li>• NT</li> <li>• NT64</li> <li>• OS2</li> <li>• OS390</li> <li>• SCO</li> <li>• SGI</li> <li>• SNI</li> <li>• SUN</li> <li>• SUN64</li> <li>• 不明 (UNKNOWN)</li> <li>• UNKNOWN_DRDA</li> <li>• VM_DRDA</li> <li>• VSE_DRDA</li> <li>• WINDOWS</li> <li>• WINDOWS95</li> </ul>

表 195. APPLICATIONS 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
CLIENT_PROTOCOL	VARCHAR(10)	client_protocol - クライアント通信プロトコル。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。 <ul style="list-style-type: none"> <li>• CPIC</li> <li>• LOCAL</li> <li>• NPIPE</li> <li>• TCPIP</li> <li>• TCPIP4</li> <li>• TCPIP6</li> </ul>
CLIENT_NNAME	VARCHAR(128)	client_nname - クライアント名モニター・エレメント
COORD_NODE_NUM	SMALLINT	coord_node - コーディネーター・ノード
COORD_AGENT_PID	BIGINT	coord_agent_pid - コーディネーター・エージェント
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 関連したエージェント数
TPMON_CLIENT_USERID	VARCHAR(256)	tpmon_client_userid - TP モニター・クライアント・ユーザー ID
TPMON_CLIENT_WKSTN	VARCHAR(256)	tpmon_client_wkstn - TP モニター・クライアント・ワークステーション名
TPMON_CLIENT_APP	VARCHAR(256)	tpmon_client_app - TP モニター・クライアント・アプリケーション名
TPMON_ACC_STR	VARCHAR(200)	tpmon_acc_str - TP モニター・クライアント・アカウント・アカウント・ストリング
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
COORD_MEMBER	SMALLINT	coord_member - コーディネーター・メンバー・モニター・エレメント
COORD_DBPARTITIONNUM	SMALLINT	コーディネーター・データベース・パーティション番号。

### BP\_HITRATIO 管理ビュー - バッファース・プールのヒット率に関する情報の検索

BP\_HITRATIO 管理ビューは、バッファース・プールのヒット率を戻します。これには、現在接続されているデータベースのすべてのバッファース・プールおよびすべて



のデータベース・パーティションに関する、全体のヒット率、データのヒット率、XDA のヒット率、および索引のヒット率が含まれます。

**注:** この管理ビューは、IBM DB2 pureScale Featureのない DB2 環境のみで作動します。DB2 pureScale 環境でのヒット率の計算に関する情報については、「データベースのモニタリング ガイドおよびリファレンス」の『DB2 pureScale 環境でのバッファ・プール・ヒット率の計算』を参照してください。

スキーマは SYSIBMADM です。

## 許可

以下のいずれかの権限が必要です。

- BP\_HITRATIO 管理ビューに対する SELECT 特権
- BP\_HITRATIO 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

接続されているデータベースのすべてのバッファ・プールに関するレポートを検索します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, SUBSTR(BP_NAME,1,14) AS BP_NAME,  
       TOTAL_HIT_RATIO_PERCENT, DATA_HIT_RATIO_PERCENT,  
       INDEX_HIT_RATIO_PERCENT, XDA_HIT_RATIO_PERCENT, DBPARTITIONNUM  
FROM SYSIBMADM.BP_HITRATIO ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

DB_NAME	BP_NAME	TOTAL_HIT_RATIO_PERCENT	DATA_HIT_RATIO_PERCENT	...
TEST	IBMDEFAULTBP	63.09	68.94	...
TEST	IBMSYSTEMBP4K	-	-	...
TEST	IBMSYSTEMBP8K	-	-	...
TEST	IBMSYSTEMBP16K	-	-	...
TEST	IBMSYSTEMBP32K	-	-	...

この照会の出力 (続き)。

```

... INDEX_HIT_RATIO_PERCENT XDA_HIT_RATIO_PERCENT DBPARTITIONNUM
... -----
...                43.20                -                0
...                -                -                0
...                -                -                0
...                -                -                0
...                -                -                0
...                -                -                0

```

## 使用上の注意

バッファーク・プールのヒット率は、読み取り合計に対する物理読み取りの率から算出されます。ヒット率が低いと、それだけキャッシュ・バッファーク・プールよりもディスクから読み取られるデータの量が多くなり、その分だけ操作にコストがかかることとなります。

## 戻される情報

表 196. BP\_HITRATIO 管理ビューによって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	レポートが要求された時刻のタイム・スタンプ。
DB_NAME	VARCHAR(128)	db_name - データベース名
BP_NAME	VARCHAR(128)	bp_name - バッファーク・プール名
TOTAL_LOGICAL_READS	BIGINT	バッファーク・プール内の論理読み取り (索引、XDA、およびデータ) の合計。
TOTAL_PHYSICAL_READS	BIGINT	バッファーク・プール内の物理読み取り (索引、XDA、およびデータ) の合計。
TOTAL_HIT_RATIO_PERCENT	DECIMAL(5,2)	ヒット率の合計 (索引、XDA、およびデータの読み取り)。
DATA_LOGICAL_READS	BIGINT	pool_data_l_reads - バッファーク・プール・データの論理読み取り
DATA_PHYSICAL_READS	BIGINT	pool_data_p_reads - バッファーク・プール・データの物理読み取り
DATA_HIT_RATIO_PERCENT	DECIMAL(5,2)	データのヒット率。
INDEX_LOGICAL_READS	BIGINT	pool_index_l_reads - バッファーク・プール索引の論理読み取り
INDEX_PHYSICAL_READS	BIGINT	pool_index_p_reads - バッファーク・プール索引の物理読み取り
INDEX_HIT_RATIO_PERCENT	DECIMAL(5,2)	索引のヒット率。
XDA_LOGICAL_READS	BIGINT	pool_xda_l_reads - バッファーク・プール XDA データの論理読み取り
XDA_PHYSICAL_READS	BIGINT	pool_xda_p_reads - バッファーク・プール XDA データの物理読み取り
XDA_HIT_RATIO_PERCENT	DECIMAL(5,2)	補助ストレージ・オブジェクトのヒット率。

表 196. BP\_HITRATIO 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## BP\_READ\_IO 管理ビュー - バッファ・プール読み取りパフォーマンス情報の検索

BP\_READ\_IO 管理ビューは、バッファ・プール読み取りパフォーマンス情報を戻します。このビューは、プリフェッチャーの効果を調べるために各バッファ・プールを表示するのに使用できます。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- BP\_READ\_IO 管理ビューに対する SELECT 特権
- BP\_READ\_IO 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベースの全パーティション上のすべてのバッファ・プールに関する、物理読み取りの合計と平均読み取り時間を検索します。

```
SELECT SUBSTR(BP_NAME, 1, 15) AS BP_NAME, TOTAL_PHYSICAL_READS,
       AVERAGE_READ_TIME_MS, DBPARTITIONNUM
FROM SYSIBMADM.BP_READ_IO ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

BP_NAME	TOTAL_PHYSICAL_READS	AVERAGE_READ_TIME_MS	DBPARTITIONNUM
IBMDEFAULTBP	811	4	0
IBMSYSTEMBP4K	0	-	0
IBMSYSTEMBP8K	0	-	0
IBMSYSTEMBP16K	0	-	0
IBMDEFAULTBP	34	0	1
IBMSYSTEMBP4K	0	-	1
IBMSYSTEMBP8K	0	-	1
IBMDEFAULTBP	34	0	2
IBMSYSTEMBP4K	0	-	2
IBMSYSTEMBP8K	0	-	2

10 record(s) selected.

## 戻される情報

表 197. BP\_READ\_IO 管理ビューによって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	レポートの生成日時。
BP_NAME	VARCHAR(128)	bp_name - バッファ ー・プール名
TOTAL_PHYSICAL_READS	BIGINT	物理読み取りの合計。
AVERAGE_READ_TIME_MS	BIGINT	平均読み取り時間 (ミリ秒)。
TOTAL_ASYNC_READS	BIGINT	非同期読み取りの合計。
AVERAGE_ASYNC_READ_TIME_MS	BIGINT	平均非同期読み取り時間 (ミリ秒)。
TOTAL_SYNC_READS	BIGINT	同期読み取りの合計。
AVERAGE_SYNC_READ_TIME_MS	BIGINT	平均同期読み取り時間 (ミリ秒)。
PERCENT_SYNC_READS	DECIMAL(5,2)	プリフェッチなしで同期的に読み取られるページのパーセンテージ。多数のアプリケーションがプリフェッチなしで同期的にデータの読み取りを行うと、システムが最適に調整されない可能性があります。

表 197. BP\_READ\_IO 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
ASYNC_NOT_READ_PERCENT	DECIMAL(5,2)	ディスクから非同期で読み取られ、照会でアクセスされなかったページのパーセンテージ。ディスクからバッファ・プールに非同期で読み取られ、照会でアクセスされないページが多すぎると、プリフェッチによってパフォーマンスが低下する可能性があります。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## BP\_WRITE\_IO 管理ビュー - バッファ・プール書き込みパフォーマンス情報の検索

BP\_WRITE\_IO 管理ビューは、各バッファ・プールごとのバッファ・プール書き込みパフォーマンス情報を戻します。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- BP\_WRITE\_IO 管理ビューに対する SELECT 特権
- BP\_WRITE\_IO 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベースの全データベース・パーティション上のすべてのバッファ・プールに関する、書き込みの合計と平均書き込み時間を検索します。

```
SELECT SUBSTR(BP_NAME, 1, 15) AS BP_NAME, TOTAL_WRITES,
       AVERAGE_WRITE_TIME_MS, DBPARTITIONNUM
FROM SYSIBMADM.BP_WRITE_IO ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

BP_NAME	TOTAL_WRITES	AVERAGE_WRITE_TIME_MS	DBPARTITIONNUM
IBMDEFAULTBP	11	5	0
IBMSYSTEMBP4K	0	-	0
IBMSYSTEMBP8K	0	-	0
IBMSYSTEMBP16K	0	-	0
IBMSYSTEMBP32K	0	-	0
IBMDEFAULTBP	0	-	1
IBMSYSTEMBP4K	0	-	1
IBMSYSTEMBP8K	0	-	1
IBMDEFAULTBP	0	-	2
IBMSYSTEMBP4K	0	-	2
IBMSYSTEMBP8K	0	-	2

11 record(s) selected.

## 戻される情報

表 198. BP\_WRITE\_IO 管理ビューによって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	レポートの生成日時。
BP_NAME	VARCHAR(128)	bp_name - バッファ・プール名
TOTAL_WRITES	BIGINT	書き込みの合計。
AVERAGE_WRITE_TIME_MS	BIGINT	平均書き込み時間 (ミリ秒)。
TOTAL_ASYNC_WRITES	BIGINT	非同期書き込みの合計。
PERCENT_WRITES_ASYNC	BIGINT	非同期書き込みのパーセント。
AVERAGE_ASYNC_WRITE_TIME_MS	BIGINT	平均非同期書き込み時間 (ミリ秒)。
TOTAL_SYNC_WRITES	BIGINT	同期書き込みの合計。
AVERAGE_SYNC_WRITE_TIME_MS	BIGINT	平均同期書き込み時間 (ミリ秒)。

表 198. BP\_WRITE\_IO 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## CONTAINER\_UTILIZATION 管理ビュー - 表スペース・コンテナーおよび使用率に関する情報の検索

CONTAINER\_UTILIZATION 管理ビューは、表スペース・コンテナーと使用率に関する情報を戻します。

これは、単一パーティション・データベースに対する LIST TABLESPACES コマンドと同様のレポートを取得します。その情報は、SNAPCONTAINER 管理ビューに基づきます。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- CONTAINER\_UTILIZATION 管理ビューに対する SELECT 特権
- CONTAINER\_UTILIZATION 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

接続されている単一パーティション・データベース内のすべての表スペース・コンテナーのリスト (ページの合計、使用可能ページ、およびそのアクセス可能性の状況に関する情報を含む) を検索します。

```

SELECT SUBSTR(TBSP_NAME,1,20) AS TBSP_NAME, INT(TBSP_ID) AS TBSP_ID,
       SUBSTR(CONTAINER_NAME,1,45) AS CONTAINER_NAME, INT(CONTAINER_ID)
       AS CONTAINER_ID, CONTAINER_TYPE, INT(TOTAL_PAGES) AS TOTAL_PAGES,
       INT(USABLE_PAGES) AS USABLE_PAGES, ACCESSIBLE
FROM SYSIBMADM.CONTAINER_UTILIZATION

```

以下はこの照会の出力例です。

```

TBSP_NAME          TBSP_ID    CONTAINER_NAME          ...
-----
SYSCATSPACE              0 D:¥DB2¥NODE0000¥SQL00001¥SQLT0000.0 ...
TEMPSPACE1              1 D:¥DB2¥NODE0000¥SQL00001¥SQLT0001.0 ...
USERSPACE1              2 D:¥DB2¥NODE0000¥SQL00001¥SQLT0002.0 ...
SYSTOOLSPACE           3 D:¥DB2¥NODE0000¥SQL00001¥SYSTOOLSPACE ...
SYSTOOLSTMPSPACE      4 D:¥DB2¥NODE0000¥SQL00001¥SYSTOOLSTMPSPACE ...

```

5 record(s) selected.

この照会の出力 (続き)。

```

... CONTAINER_ID CONTAINER_TYPE TOTAL_PAGES USABLE_PAGES ACCESSIBLE
... -----
...          0 PATH                0           0           1
...          0 PATH                0           0           1
...          0 PATH                0           0           1
...          0 PATH                0           0           1
...          0 PATH                0           0           1

```

## 戻される情報

ファイル・システムの情報を戻すためには、データベース・マネージャ構成で BUFFERPOOL スナップショット・モニター・スイッチを有効にする必要があります。

表 199. CONTAINER\_UTILIZATION 管理ビューによって戻される情報

列名	データ・タイプ	説明または対応するモニター・エリメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
CONTAINER_NAME	VARCHAR(256)	container_name - コンテナ名
CONTAINER_ID	BIGINT	container_id - コンテナ ID
CONTAINER_TYPE	VARCHAR(16)	container_type - コンテナ・タイプ  これは、sqlutil.h での定義を基にしたテキスト ID です。以下のいずれかとなります。  <ul style="list-style-type: none"> <li>• DISK_EXTENT_TAG</li> <li>• DISK_PAGE_TAG</li> <li>• FILE_EXTENT_TAG</li> <li>• FILE_PAGE_TAG</li> <li>• PATH</li> </ul>
TOTAL_PAGES	BIGINT	container_total_pages - コンテナ内の合計ページ数



表 199. CONTAINER\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
USABLE_PAGES	BIGINT	container_usable_pages - コンテナ内の使用可能なページ数
ACCESSIBLE	SMALLINT	container_accessible - コンテナのアクセス可能性
STRIPE_SET	BIGINT	container_stripe_set - ストライプ・セット
FS_ID	VARCHAR(22)	fs_id - 固有のファイル・システム識別番号
FS_TOTAL_SIZE_KB	BIGINT	fs_total_size - ファイル・システムの合計サイズ。このインターフェースは値を KB 単位で戻します。
FS_USED_SIZE_KB	BIGINT	fs_used_size - ファイル・システム上で使用されるスペースの量。このインターフェースは値を KB 単位で戻します。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

## LOG\_UTILIZATION 管理ビュー - ログ使用率に関する情報の検索

LOG\_UTILIZATION 管理ビューは、現在接続されているデータベースのログ使用率に関する情報を戻します。各データベース・パーティションごとに 1 つの行が戻されます。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- LOG\_UTILIZATION 管理ビューに対する SELECT 特権
- LOG\_UTILIZATION 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベース SAMPLE のログ使用率をリストします。

```
SELECT * FROM SYSIBMADM.LOG_UTILIZATION
```

以下はこの照会の出力例です。

```
DB_NAME    ... LOG_UTILIZATION_PERCENT TOTAL_LOG_USED_KB    ...
-----    ... -----
SAMPLE     ...                9.75                1989 ...
          ...
          1 record(s) selected.          ...
```

この照会の出力 (続き)。

```
... TOTAL_LOG_AVAILABLE_KB TOTAL_LOG_USED_TOP_KB DBPARTITIONNUM
... -----
...                18411                1990                0
...
...
...
...
```

### 使用上の注意

無限ロギングが構成されているデータベースの場合、LOG\_UTILIZATION\_PERCENT および TOTAL\_LOG\_AVAILABLE\_KB は NULL になります。

### 戻される情報

表 200. LOG\_UTILIZATION 管理ビューによって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
DB_NAME	VARCHAR(128)	db_name - データベース名
LOG_UTILIZATION_PERCENT	DECIMAL(5,2)	合計ログ・スペースの使用率パーセント。
TOTAL_LOG_USED_KB	BIGINT	total_log_used - 使用されているログ・スペースの合計。このインターフェースは値を KB 単位で戻します。
TOTAL_LOG_AVAILABLE_KB	BIGINT	total_log_available - 使用可能なログ合計。このインターフェースは値を KB 単位で戻します。
TOTAL_LOG_USED_TOP_KB	BIGINT	tot_log_used_top - 使用された最大合計ログ・スペース。このインターフェースは値を KB 単位で戻します。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

表 200. LOG\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## LONG\_RUNNING\_SQL 管理ビュー

LONG\_RUNNING\_SQL 管理ビューは、現在接続されているデータベースで実行されている SQL ステートメントを戻します。このビューを使用することにより、データベース内の、実行時間の長い SQL ステートメントを識別することができます。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- LONG\_RUNNING\_SQL 管理ビューに対する SELECT 特権
- LONG\_RUNNING\_SQL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベースで実行時間が最も長い SQL ステートメントに関するレポートを検索します。

```
SELECT SUBSTR(STMT_TEXT, 1, 50) AS STMT_TEXT, AGENT_ID,
       ELAPSED_TIME_MIN, APPL_STATUS, DBPARTITIONNUM
FROM SYSIBMADM.LOG_RUNNING_SQL ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

```
STMT_TEXT                                AGENT_ID    ...
-----
select * from dbuser.employee            228 ...
select * from dbuser.employee            228 ...
select * from dbuser.employee            228 ...
...
3 record(s) selected.                    ...
```

この照会の出力 (続き)。

```
... ELAPSED_TIME_MIN APPL_STATUS      DBPARTITIONNUM
... -----
...                2 UOWWAIT                0
...                0 CONNECTED              1
...                0 CONNECTED              2
```

## 使用上の注意

このビューを使用することにより、データベース内の、実行時間の長い SQL ステートメントを識別することができます。現在実行中の照会を見て、実行時間が最も長いステートメント、および照会の現行状況を調べることができます。エージェント ID をユニーク ID として使用することにより、その SQL ステートメントを含むアプリケーションをさらに詳しく調査することができます。長時間実行していてロック待機している場合は、LOCKWAITS または LOCKS\_HELD 管理ビューを使用してさらに深く掘り下げることもできます。「waiting on User (ユーザー待機中)」という表示は、DB2 サーバーが何の動作も行っておらず、アプリケーションによる何らかの動作 (例えば次のフェッチの発行または次の SQL ステートメントのサブミット) を待機していることを意味します。

## 戻される情報

表 201. LONG\_RUNNING\_SQL 管理ビューによって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	レポートの生成時刻。
ELAPSED_TIME_MIN	INTEGER	ステートメントの経過時間 (分)。
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
APPL_NAME	VARCHAR(256)	appl_name - アプリケーション名

表 201. LONG\_RUNNING\_SQL 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
APPL_STATUS	VARCHAR(22)	<p>appl_status - アプリケーション状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• BACKUP</li> <li>• COMMIT_ACT</li> <li>• COMP</li> <li>• CONNECTED</li> <li>• CONNECTPEND</li> <li>• CREATE_DB</li> <li>• DECOUPLED</li> <li>• DISCONNECTPEND</li> <li>• INTR</li> <li>• IOERROR_WAIT</li> <li>• LOAD</li> <li>• LOCKWAIT</li> <li>• QUIESCE_TABLESPACE</li> <li>• RECOMP</li> <li>• REMOTE_RQST</li> <li>• RESTART</li> <li>• RESTORE</li> <li>• ROLLBACK_ACT</li> <li>• ROLLBACK_TO_SAVEPOINT</li> <li>• TEND</li> <li>• THABRT</li> <li>• THCOMT</li> <li>• TPREP</li> <li>• UNLOAD</li> <li>• UOWEXEC</li> <li>• UOWWAI</li> <li>• WAITFOR_REMOTE</li> </ul>
AUTHID	VARCHAR(128)	auth_id - 許可 ID
INBOUND_COMM_ADDRESS	VARCHAR(32)	inbound_comm_address - インバウンド通信アドレス
STMT_TEXT	CLOB(16 M)	stmt_text - SQL 動的ステートメント・テキスト
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

表 201. LONG\_RUNNING\_SQL 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## QUERY\_PREP\_COST 管理ビュー - ステートメント準備時間に関する情報の検索

QUERY\_PREP\_COST 管理ビューは、ステートメントのリストおよびステートメントの準備に必要な時間に関する情報を戻します。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- QUERY\_PREP\_COST 管理ビューに対する SELECT 特権
- QUERY\_PREP\_COST 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

準備に最も高い割合の時間が費やされた照会に関するレポートを検索します。

```
SELECT NUM_EXECUTIONS, AVERAGE_EXECUTION_TIME_S, PREP_TIME_PERCENT,
       SUBSTR(STMT_TEXT, 1, 30) AS STMT_TEXT, DBPARTITIONNUM
FROM SYSIBMADM.QUERY_PREP_COST ORDER BY PREP_TIME_PERCENT
```

以下はこの照会の出力例です。

```
NUM_EXECUTIONS      AVERAGE_EXECUTION_TIME_S ...
-----...-----
                1                          25 ...
```

1 record(s) selected.

この照会の出力 (続き)。

```
... PREP_TIME_PERCENT STMT_TEXT                      DBPARTITIONNUM
... -----
...                0.0 select * from dbuser.employee          0
```

### 使用上の注意

ビューから選択するとき ORDER BY 節を使用することにより、最も準備コストの高い照会を識別することができます。このビューで、照会の実行頻度および各照会の平均実行時間を調べることができます。照会をコンパイルして最適化する時間

と照会の実行にかかる時間にほとんど差がない場合には、使用している最適化クラスを調べたほうが良いかもしれません。最適化クラスを下げることによって照会での完全な最適化がより短時間で実行されるようになるため、結果がより早く戻される可能性があります。しかし、照会での準備に多大な時間がかかり、しかもそれが (再準備されることなく) 何度も実行されるとなると、最適化クラスの問題ではない可能性があります。

## 戻される情報

表 202. QUERY\_PREP\_COST 管理ビューによって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	レポートの生成日時。
NUM_EXECUTIONS	BIGINT	num_executions - ステートメント実行回数
AVERAGE_EXECUTION_TIME_S	BIGINT	平均実行時間 (秒単位)。
AVERAGE_EXECUTION_TIME_MS	BIGINT	平均実行時間 (小数のマイクロ秒単位)。
PREP_TIME_MS	BIGINT	prep_time_worst - ステートメント最長準備時間 (ミリ秒単位)
PREP_TIME_PERCENT	DECIMAL(8,2)	準備に費やされる実行時間のパーセント。準備時間を合計実行時間で除算して計算されます。
STMT_TEXT	CLOB(2 M)	stmt_text - SQL 動的ステートメント・テキスト
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPAGENT 管理ビューおよび SNAP\_GET\_AGENT 表関数 - agent 論理データ・グループのアプリケーション・スナップショット情報の検索

SNAPAGENT 管理ビューおよび SNAP\_GET\_AGENT 表関数は、アプリケーション・スナップショットから、特に agent 論理データ・グループのエージェント情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 842 ページの『SNAPAGENT 管理ビュー』
- 843 ページの『SNAP\_GET\_AGENT 表関数』

## SNAPAGENT 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関する agent 論理データ・グループのアプリケーション・スナップショット情報を取得することができます。

SNAPAGENT 管理ビューを SNAPAGENT\_MEMORY\_POOL、SNAPAPPL、SNAPAPPL\_INFO、SNAPSTMT、および SNAPSUBSECTION 管理ビューとともに使用すると、**GET SNAPSHOT FOR APPLICATIONS ON database-alias** CLP コマンドに相当する情報が提供されます。ただし、すべてのデータベース・メンバーからデータを取得します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、845 ページの表 203 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPAGENT 管理ビューに対する SELECT 特権
- SNAPAGENT 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_AGENT 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続中のデータベースに関するすべてのアプリケーション・スナップショット情報を、agent 論理データ・グループから取得します。

```
SELECT * FROM SYSIBMADM.SNAPAGENT
```



以下はこの照会の出力例です。

```
SNAPSHOT_TIMESTAMP      DB_NAME      AGENT_ID      ...
-----
2005-07-19-11.03.26.740423 SAMPLE      101 ...
2005-07-19-11.03.26.740423 SAMPLE      49 ...
...
2 record(s) selected.      ...
```

この照会からの出力 (続き)。

```
... AGENT_PID      LOCK_TIMEOUT_VAL      DBPARTITIONNUM
... -----
...      11980      -1      0
...      15940      -1      0
...
...
...
```

## SNAP\_GET\_AGENT 表関数

SNAP\_GET\_AGENT 表関数は SNAPAGENT 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_AGENT 表関数を SNAP\_GET\_AGENT\_MEMORY\_POOL、SNAP\_GET\_APPL、SNAP\_GET\_APPL\_INFO、SNAP\_GET\_STMT、および SNAP\_GET\_SUBSECTION 表関数とともに使用すると、**GET SNAPSHOT FOR ALL APPLICATIONS CLP** コマンドに相当する情報が提供されます。ただし、すべてのデータベース・メンバーからデータを取得します。

戻される可能性のある情報の完全なリストは、845 ページの表 203 を参照してください。

## 構文

```
▶▶ SNAP_GET_AGENT ( ( dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブ

なデータベース・メンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり dbname のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されま  
す。アクティブなデータベース・メンバーとは、アプリケーションによるデータ  
ベースへの接続およびデータベースの使用が可能なメンバーのことです。

dbname と member の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシ  
ジャーにより作成されるファイルからのデータの読み取りが試行されます。このファ  
イルはいつでも作成される可能性があるため、データは現行のものであるとは限ら  
ないことに注意してください。対応するスナップショット API 要求タイプを持つフ  
ァイルが存在しない場合、SNAP\_GET\_AGENT 表関数は、現在接続中のデータベー  
スおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_AGENT 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいづれ  
かの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権  
が PUBLIC に付与されます。

## 例

すべてのアクティブ・データベース内のすべてのアプリケーションに関する、すべ  
てのアプリケーション・スナップショット情報を取得します。

```
SELECT * FROM TABLE(SNAP_GET_AGENT(CAST(NULL AS VARCHAR(128)), -1)) AS T
```

以下はこの照会の出力例です。

SNAPSHOT_TIMESTAMP	DB_NAME	AGENT_ID	...
2006-01-03-17.21.38.530785	SAMPLE	48	...
2006-01-03-17.21.38.530785	SAMPLE	47	...
2006-01-03-17.21.38.530785	SAMPLE	46	...
2006-01-03-17.21.38.530785	TESTDB	30	...
2006-01-03-17.21.38.530785	TESTDB	29	...
2006-01-03-17.21.38.530785	TESTDB	28	...

6 record(s) selected.

この照会からの出力 (続き)。

```

... AGENT_PID          LOCK_TIMEOUT_VAL    DBPARTITIONNUM
... -----
...          7696             -1                0
...          8536             -1                0
...          6672             -1                0
...          2332             -1                0
...          8360             -1                0
...          6736             -1                0
...

```

## 戻される情報

表 203. SNAPAGENT 管理ビューおよび SNAP\_GET\_AGENT 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
AGENT_PID	BIGINT	agent_pid - エンジン・ディスパッチ可能単位 (EDU)
LOCK_TIMEOUT_VAL	BIGINT	lock_timeout_val - ロック・タイムアウト (秒)
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数 - appl\_info 論理データ・グループのスナップショット情報の検索

SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数は、アプリケーション・スナップショットから、特に appl\_info 論理データ・グループのアプリケーション情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPAPPL\_INFO 管理ビュー』
- 846 ページの 『SNAP\_GET\_APPL\_INFO 表関数』

### SNAPAPPL\_INFO 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関する appl\_info 論理データ・グループのスナップショット情報を取得することができます。

SNAPAPPL\_INFO 管理ビューを

SNAPAGENT、SNAPAGENT\_MEMORY\_POOL、SNAPAPPL、SNAPSTMT、および SNAPSUBSECTION 管理ビューとともに使用すると、**GET SNAPSHOT FOR APPLICATIONS ON database-alias** CLP コマンドに相当する情報が提供されます。ただし、すべてのデータベース・メンバーからデータを取得します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、849 ページの表 204 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPAPPL\_INFO 管理ビューに対する SELECT 特権
- SNAPAPPL\_INFO 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_APPL\_INFO 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

また、以下のいずれかの権限が必要です。

- SYSMON
- SYSMAINT
- SYSCTRL
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在のデータベースに接続中のアプリケーションの状況を取得します。

```
SELECT AGENT_ID, SUBSTR(APPL_NAME,1,10) AS APPL_NAME, APPL_STATUS
FROM SYSIBMADM.SNAPAPPL_INFO
```

以下はこの照会の出力例です。

AGENT_ID	APPL_NAME	APPL_STATUS
101	db2bp.exe	UOWEXEC
49	db2bp.exe	CONNECTED

2 record(s) selected.

## SNAP\_GET\_APPL\_INFO 表関数

SNAP\_GET\_APPL\_INFO 表関数は SNAPAPPL\_INFO 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集

約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_APPL\_INFO 表関数を

SNAP\_GET\_AGENT、SNAP\_GET\_AGENT\_MEMORY\_POOL、

SNAP\_GET\_APPL、SNAP\_GET\_STMT、および SNAP\_GET\_SUBSECTION 表関数とともに使用すると、**GET SNAPSHOT FOR ALL APPLICATIONS CLP** コマンドに相当する情報が提供されます。ただし、すべてのデータベース・メンバーからデータを取得します。

戻される可能性のある情報の完全なリストは、849 ページの表 204 を参照してください。

## 構文

```
▶▶ SNAP_GET_APPL_INFO ( ( dbname ) [ , member ] ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されず、アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_APPL\_INFO 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_APPL\_INFO 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

接続中のデータベース・メンバー上のすべてのアプリケーションの状況を取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AGENT_ID,  
       SUBSTR(APPL_NAME,1,10) AS APPL_NAME, APPL_STATUS  
FROM TABLE(SNAP_GET_APPL_INFO(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	AGENT_ID	APPL_NAME	APPL_STATUS
TOOLSDB	14	db2bp.exe	CONNECTED
SAMPLE	15	db2bp.exe	UOWEXEC
SAMPLE	8	javaw.exe	CONNECTED
SAMPLE	7	db2bp.exe	UOWWAIT

4 record(s) selected.

以下は、表関数の結果からの SELECT の実行時に入手できる内容について示しています。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AUTHORITY_LVL  
FROM TABLE(SNAP_GET_APPL_INFO(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	AUTHORITY_LVL
TESTDB	SYSDM(GROUP) + DBADM(USER) + CREATETAB(USER, GROUP) + BINDADD(USER, GROUP) + CONNECT(USER, GROUP) + CREATE_NOT_FENC(USER) + IMPLICIT_SCHEMA(USER, GROUP) + LOAD(USER) + CREATE_EXT_RT(USER) + QUIESCE_CONN(USER)
TESTDB	SYSDM(GROUP) + DBADM(USER) + CREATETAB(USER, GROUP) + BINDADD(USER, GROUP) + CONNECT(USER, GROUP) + CREATE_NOT_FENC(USER) + IMPLICIT_SCHEMA(USER, GROUP) + LOAD(USER) + CREATE_EXT_RT(USER) + QUIESCE_CONN(USER)
TESTDB	SYSDM(GROUP) + DBADM(USER) + CREATETAB(USER, GROUP) + BINDADD(USER, GROUP) + CONNECT(USER, GROUP) +

CREATE\_NOT\_FENC(USER) + IMPLICIT\_SCHEMA(USER, GROUP) +  
LOAD(USER) + CREATE\_EXT\_RT(USER) + QUIESCE\_CONN(USER)

3 record(s) selected.

### 戻される情報

表 204. SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
APPL_STATUS	VARCHAR(22)	<p>appl_status - アプリケーション状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• BACKUP</li> <li>• COMMIT_ACT</li> <li>• COMP</li> <li>• CONNECTED</li> <li>• CONNECTPEND</li> <li>• CREATE_DB</li> <li>• DECOUPLED</li> <li>• DISCONNECTPEND</li> <li>• INTR</li> <li>• IOERROR_WAIT</li> <li>• LOAD</li> <li>• LOCKWAIT</li> <li>• QUIESCE_TABLESPACE</li> <li>• RECOMP</li> <li>• REMOTE_RQST</li> <li>• RESTART</li> <li>• RESTORE</li> <li>• ROLLBACK_ACT</li> <li>• ROLLBACK_TO_SAVEPOINT</li> <li>• TEND</li> <li>• THABRT</li> <li>• THCOMT</li> <li>• TPREP</li> <li>• UNLOAD</li> <li>• UOWEXEC</li> <li>• UOWWAIT</li> <li>• WAITFOR_REMOTE</li> </ul>
CODEPAGE_ID	BIGINT	codepage_id - アプリケーションで使用するコード・ページ ID

表 204. SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 関連したエージェント数
COORD_NODE_NUM	SMALLINT	COORD_NODE_NUM は使用が推奨されないため、MEMBER で置き換えられます。
AUTHORITY_LVL	VARCHAR(512)	<p>authority_bitmap - ユーザー許可レベル。</p> <p>このインターフェースは、sql.h で定義されたデータベース権限およびそれらのソースに基づくテキスト ID を戻し、その形式は次のとおりです。authority(source, ...) + authority(source, ...) + ... 権限のソースは複数でも構いません。USER、GROUP、または USER と GROUP のいずれかです。</p> <p>"authority" に使用できる値</p> <ul style="list-style-type: none"> <li>• ACCESSCTRL</li> <li>• BINDADD</li> <li>• CONNECT</li> <li>• CREATE_EXT_RT</li> <li>• CREATE_NOT_FENC</li> <li>• CREATETAB</li> <li>• DATAACCESS</li> <li>• DBADM</li> <li>• EXPLAIN</li> <li>• IMPLICIT_SCHEMA</li> <li>• LOAD</li> <li>• LIBADM</li> <li>• QUIESCE_CONN</li> <li>• SECADM</li> <li>• SQLADM</li> <li>• SYSADM</li> <li>• SYSCTRL</li> <li>• SYSMOINT</li> <li>• SYSMON</li> <li>• SYSQUIESCE</li> <li>• WLMADM</li> </ul> <p>"source" に使用できる値</p> <ul style="list-style-type: none"> <li>• USER - ユーザーに付与された権限、またはそのユーザーに付与されているロールに付与された権限。</li> <li>• GROUP - ユーザーが属するグループに付与された権限、またはユーザーが属するグループに付与されるロールに付与された権限。</li> </ul>



表 204. SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
CLIENT_PID	BIGINT	client_pid - クライアント・プロセス ID
COORD_AGENT_PID	BIGINT	coord_agent_pid - コーディネーター・エージェント
STATUS_CHANGE_TIME	TIMESTAMP	status_change_time - アプリケーション状況変更時刻
CLIENT_PLATFORM	VARCHAR(12)	<p>client_platform - クライアント・オペレーティング・プラットフォーム。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。</p> <ul style="list-style-type: none"> <li>• AIX</li> <li>• AIX64</li> <li>• AS400_DRDA</li> <li>• DOS</li> <li>• DYNIX</li> <li>• HP</li> <li>• HP64</li> <li>• HPIA</li> <li>• HPIA64</li> <li>• LINUX</li> <li>• LINUX390</li> <li>• LINUXIA64</li> <li>• LINUXPPC</li> <li>• LINUXPPC64</li> <li>• LINUXX8664</li> <li>• LINUXZ64</li> <li>• MAC</li> <li>• MVS_DRDA</li> <li>• NT</li> <li>• NT64</li> <li>• OS2</li> <li>• OS390</li> <li>• SCO</li> <li>• SGI</li> <li>• SNI</li> <li>• SUN</li> <li>• SUN64</li> <li>• UNKNOWN</li> <li>• UNKNOWN_DRDA</li> <li>• VM_DRDA</li> <li>• VSE_DRDA</li> <li>• WINDOWS</li> </ul>

表 204. SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
CLIENT_PROTOCOL	VARCHAR(10)	client_protocol - クライアント通信プロトコル。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。 <ul style="list-style-type: none"><li>• CPIC</li><li>• LOCAL</li><li>• NPIPE</li><li>• TCPIP (DB2 UDB の場合)</li><li>• TCPIP4</li><li>• TCPIP6</li></ul>
TERRITORY_CODE	SMALLINT	territory_code - データベース・テリトリリー・コード
APPL_NAME	VARCHAR(256)	appl_name - アプリケーション名
APPL_ID	VARCHAR(128)	appl_id - アプリケーション ID
SEQUENCE_NO	VARCHAR(4)	sequence_no - シーケンス番号
PRIMARY_AUTH_ID	VARCHAR(128)	auth_id - 許可 ID
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - セッション許可 ID
CLIENT_NNAME	VARCHAR(128)	client_nname - クライアント名モニター・エレメント
CLIENT_PRDID	VARCHAR(128)	client_prdid - クライアント製品/バージョン ID
INPUT_DB_ALIAS	VARCHAR(128)	input_db_alias - 入力データベース別名
CLIENT_DB_ALIAS	VARCHAR(128)	client_db_alias - アプリケーションで使用されるデータベース別名
DB_NAME	VARCHAR(128)	db_name - データベース名
DB_PATH	VARCHAR(1024)	db_path - データベース・パス
EXECUTION_ID	VARCHAR(128)	execution_id - ユーザー・ログイン ID
CORR_TOKEN	VARCHAR(128)	corr_token - DRDA 関連トークン
TPMON_CLIENT_USERID	VARCHAR(256)	tpmon_client_userid - TP モニター・クライアント・ユーザー ID
TPMON_CLIENT_WKSTN	VARCHAR(256)	tpmon_client_wkstn - TP モニター・クライアント・ワークステーション名
TPMON_CLIENT_APP	VARCHAR(256)	tpmon_client_app - TP モニター・クライアント・アプリケーション名
TPMON_ACC_STR	VARCHAR(200)	tpmon_acc_str - TP モニター・クライアント・アカウント・アカウンティング・ストリング
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
WORKLOAD_ID	INTEGER	workload_id - ワークロード ID : モニター・エレメント
IS_SYSTEM_APPL	SMALLINT	is_system_appl - システム・アプリケーション・モニター・エレメント

表 204. SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
COORD_MEMBER	SMALLINT	coord_member - コーディネーター・メンバー・モニター・エレメント
COORD_DBPARTITIONNUM	SMALLINT	調整データベース・パーティション番号。

## SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数 - appl 論理データ・グループのスナップショット情報の検索

SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数は、アプリケーション・スナップショットから、特に appl 論理データ・グループのアプリケーション情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPAPPL 管理ビュー』
- 854 ページの『SNAP\_GET\_APPL 表関数』

### SNAPAPPL 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関する appl 論理データ・グループのスナップショット情報を取得することができます。

SNAPAPPL 管理ビューを

SNAPAGENT、SNAPAGENT\_MEMORY\_POOL、SNAPAPPL\_INFO、SNAPSTMT、および SNAPSUBSECTION 管理ビューとともに使用すると、**GET SNAPSHOT FOR APPLICATIONS ON database-alias** CLP コマンドに相当する情報が提供されます。ただし、すべてのデータベース・メンバーからデータを取得します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、856 ページの表 205 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPAPPL 管理ビューに対する SELECT 特権
- SNAPAPPL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかの特権または権限が必要です。

- SNAP\_GET\_APPL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

接続中のデータベース内の各アプリケーションについて読み取りおよび書き込みが行われた行の詳細を取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AGENT_ID, ROWS_READ, ROWS_WRITTEN
FROM SYSIBMADM.SNAPAPPL
```

以下はこの照会の出力例です。

DB_NAME	AGENT_ID	ROWS_READ	ROWS_WRITTEN
SAMPLE		7	25

1 record(s) selected.

## SNAP\_GET\_APPL 表関数

SNAP\_GET\_APPL 表関数は SNAPAPPL 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_APPL 表関数を SNAP\_GET\_AGENT、SNAP\_GET\_AGENT\_MEMORY\_POOL、SNAP\_GET\_APPL\_INFO、SNAP\_GET\_STMT、および SNAP\_GET\_SUBSECTION 表関数とともに使用すると、**GET SNAPSHOT FOR ALL APPLICATIONS** CLP コマンドに相当する情報が提供されます。ただし、すべてのデータベース・メンバーからデータを取得します。

戻される可能性のある情報の完全なリストは、856 ページの表 205 を参照してください。

### 構文

```
▶▶ SNAP_GET_APPL ( ( dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_APPL 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_APPL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべてのアクティブ・データベースの各アプリケーションについて読み取りおよび書き込みが行われた行の詳細を取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AGENT_ID, ROWS_READ, ROWS_WRITTEN
FROM TABLE (SNAP_GET_APPL(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	AGENT_ID	ROWS_READ	ROWS_WRITTEN
WSDB	679	0	0
WSDB	461	3	0
WSDB	460	4	0
TEST	680	4	0
TEST	455	6	0
TEST	454	0	0
TEST	453	50	0

## 戻される情報

表 205. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エリメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
UOW_LOG_SPACE_USED	BIGINT	uow_log_space_used - 作業単位ログ・スペース
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
STMT_HISTORY_LIST_SIZE	BIGINT	stmt_history_list_size - ステートメント履歴リストのサイズ
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファークール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファークール・データの物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファークールへのデータの書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファークール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファークール索引の物理読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファークール索引の書き込み
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファークール一時データの論理読み取り

表 205. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファーク・プールの時データの物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファーク・プールの時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファーク・プールの時索引の物理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファーク・プールの時 XDA データの論理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファーク・プールの時 XDA データの物理読み取り : モニター・エレメント
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファーク・プール XDA データの論理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファーク・プール XDA データの物理読み取り
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファーク・プール XDA データの書き込み
POOL_READ_TIME	BIGINT	pool_read_time - バッファーク・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファーク・プール物理書き込み時間の合計
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不能プリフェッチ・ページ
LOCKS_HELD	BIGINT	locks_held - ロック保持数
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間

表 205. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_ESCALS	BIGINT	lock_escalations - ロック・エスカレーション数
X_LOCK_ESCALS	BIGINT	x_lock_escalations - 排他ロック・エスカレーション数
DEADLOCKS	BIGINT	deadlocks - デッドロック検出数
TOTAL_SORTS	BIGINT	total_sorts - ソート合計
TOTAL_SORT_TIME	BIGINT	total_sort_time - ソート時間合計
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
COMMIT_SQL_STMTS	BIGINT	commit_sql_stmts - 試行されたコミット・ステートメント
ROLLBACK_SQL_STMTS	BIGINT	rollback_sql_stmts - 試行されたロールバック・ステートメント
DYNAMIC_SQL_STMTS	BIGINT	dynamic_sql_stmts - 試行された動的 SQL ステートメント
STATIC_SQL_STMTS	BIGINT	static_sql_stmts - 試行された静的 SQL ステートメント
FAILED_SQL_STMTS	BIGINT	failed_sql_stmts - 失敗したステートメント操作
SELECT_SQL_STMTS	BIGINT	select_sql_stmts - 実行された選択 SQL ステートメント
DDL_SQL_STMTS	BIGINT	ddl_sql_stmts - データ定義言語 (DDL) SQL ステートメント
UID_SQL_STMTS	BIGINT	uid_sql_stmts - 実行された更新/挿入/削除 SQL ステートメント
INT_AUTO_REBINDS	BIGINT	int_auto_rebinds - 内部自動再バインド
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 削除された内部行数
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 更新された内部行数
INT_COMMITS	BIGINT	int_commits - 内部コミット数
INT_ROLLBACKS	BIGINT	int_rollback - 内部ロールバック数
INT_DEADLOCK_ROLLBACKS	BIGINT	int_deadlock_rollback - デッドロックによる内部ロールバック数
ROWS_DELETED	BIGINT	rows_deleted - 削除行数
ROWS_INSERTED	BIGINT	rows_inserted - 挿入行数
ROWS_UPDATED	BIGINT	rows_updated - 更新行数
ROWS_SELECTED	BIGINT	rows_selected - 選択行数
BINDS_PRECOMPILES	BIGINT	binds_precompiles - 試行されたバインド/プリコンパイル



表 205. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
OPEN_REM_CURS	BIGINT	open_rem_curs - 開かれているリモート・カーソル
OPEN_REM_CURS_BLK	BIGINT	open_rem_curs_blk - 開かれているリモート・ブロック・カーソル
REJ_CURS_BLK	BIGINT	rej_curs_blk - リジェクトされたブロック・カーソル要求
ACC_CURS_BLK	BIGINT	acc_curs_blk - 受け入れられたブロック・カーソル要求
SQL_REQS_SINCE_COMMIT	BIGINT	sql_reqs_since_commit - 最終コミット後の SQL 要求数
LOCK_TIMEOUTS	BIGINT	lock_timeouts - ロック・タイムアウト数
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 挿入された内部行数
OPEN_LOC_CURS	BIGINT	open_loc_curs - 開かれているローカル・カーソル
OPEN_LOC_CURS_BLK	BIGINT	open_loc_curs_blk - 開かれているローカル・ブロック・カーソル
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - パッケージ・キャッシュ参照
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - パッケージ・キャッシュ挿入
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - カタログ・キャッシュ参照数
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - カタログ・キャッシュ挿入数
CAT_CACHE_OVERFLOWS	BIGINT	cat_cache_overflows - カタログ・キャッシュ・オーバーフロー数
NUM_AGENTS	BIGINT	num_agents - ステートメントで作動しているエージェントの数
AGENTS_STOLEN	BIGINT	agents_stolen - スチールされたエージェント
ASSOCIATED_AGENTS_TOP	BIGINT	associated_agents_top - 関連エージェント最大数
APPL_PRIORITY	BIGINT	appl_priority - アプリケーション・エージェント優先順位

表 205. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
APPL_PRIORITY_TYPE	VARCHAR(16)	appl_priority_type - アプリケーション優先順位タイプ。このインターフェースは、sqlmon.h 内の定義に基づいてテキスト ID を戻します。それは、次のうちの 1 つです。 <ul style="list-style-type: none"> <li>• DYNAMIC_PRIORITY</li> <li>• FIXED_PRIORITY</li> </ul>
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間
APPL_SECTION_LOOKUPS	BIGINT	appl_section_lookups - セクションの参照回数
APPL_SECTION_INSERTS	BIGINT	appl_section_inserts - セクション挿入数
LOCKS_WAITING	BIGINT	locks_waiting - ロックで待機中の現行エージェント
TOTAL_HASH_JOINS	BIGINT	total_hash_joins - ハッシュ結合の合計
TOTAL_HASH_LOOPS	BIGINT	total_hash_loops - ハッシュ・ループの合計
HASH_JOIN_OVERFLOWS	BIGINT	hash_join_overflows - ハッシュ結合のオーバーフロー
HASH_JOIN_SMALL_OVERFLOWS	BIGINT	hash_join_small_overflows - ハッシュ結合の短精度オーバーフロー
APPL_IDLE_TIME	BIGINT	appl_idle_time - アプリケーション・アイドル時間
UOW_LOCK_WAIT_TIME	BIGINT	uow_lock_wait_time - ロック待機中の作業単位の合計時間
UOW_COMP_STATUS	VARCHAR(14)	uow_comp_status - 作業単位完了状況。このインターフェースは、sqlmon.h 内の定義に基づいてテキスト ID を戻します。それは、次のうちの 1 つです。 <ul style="list-style-type: none"> <li>• APPL_END</li> <li>• UOWABEND</li> <li>• UOWCOMMIT</li> <li>• UOWDEADLOCK</li> <li>• UOWLOCKTIMEOUT</li> <li>• UOWROLLBACK</li> <li>• UOWUNKNOWN</li> </ul>
AGENT_USR_CPU_TIME_S	BIGINT	agent_usr_cpu_time - エージェントが使用したユーザー CPU 時間 (秒単位)*

表 205. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
AGENT_USR_CPU_TIME_MS	BIGINT	agent_usr_cpu_time - エージェントが使用したユーザー CPU 時間 (小数部、マイクロ秒単位)*
AGENT_SYS_CPU_TIME_S	BIGINT	agent_sys_cpu_time - エージェントが使用したシステム CPU 時間 (秒単位)*
AGENT_SYS_CPU_TIME_MS	BIGINT	agent_sys_cpu_time - エージェントが使用したシステム CPU 時間 (小数部、マイクロ秒単位)*
APPL_CON_TIME	TIMESTAMP	appl_con_time - 接続要求開始タイム・スタンプ
CONN_COMPLETE_TIME	TIMESTAMP	conn_complete_time - 接続要求完了タイム・スタンプ
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ
UOW_START_TIME	TIMESTAMP	uow_start_time - 作業単位開始タイム・スタンプ
UOW_STOP_TIME	TIMESTAMP	uow_stop_time - 作業単位停止タイム・スタンプ
PREV_UOW_STOP_TIME	TIMESTAMP	prev_uow_stop_time - 直前の作業単位完了タイム・スタンプ
UOW_ELAPSED_TIME_S	BIGINT	uow_elapsed_time - 最新の作業単位の経過時間 (秒単位)*
UOW_ELAPSED_TIME_MS	BIGINT	uow_elapsed_time - 最新の作業単位の経過時間 (小数部、マイクロ秒単位)*
ELAPSED_EXEC_TIME_S	BIGINT	elapsed_exec_time - ステートメント実行経過時間 (秒単位)*
ELAPSED_EXEC_TIME_MS	BIGINT	elapsed_exec_time - ステートメント実行経過時間 (小数部、マイクロ秒単位)*
INBOUND_COMM_ADDRESS	VARCHAR(32)	inbound_comm_address - インバウンド通信アドレス
LOCK_TIMEOUT_VAL	BIGINT	lock_timeout_val - ロック・タイムアウト (秒)
PRIV_WORKSPACE_NUM_OVERFLOWS	BIGINT	priv_workspace_num_overflows - 専用ワークスペースのオーバーフロー回数
PRIV_WORKSPACE_SECTION_INSERTS	BIGINT	priv_workspace_section_inserts - 専用ワークスペース・セクション挿入
PRIV_WORKSPACE_SECTION_LOOKUPS	BIGINT	priv_workspace_section_lookups - 専用ワークスペース・セクションの参照

表 205. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
PRIV_WORKSPACE_SIZE_TOP	BIGINT	priv_workspace_size_top - 専用ワークスペースの最大サイズ
SHR_WORKSPACE_NUM_OVERFLOWS	BIGINT	shr_workspace_num_overflows - 共有ワークスペースのオーバーフロー回数
SHR_WORKSPACE_SECTION_INSERTS	BIGINT	shr_workspace_section_inserts - 共有ワークスペース・セクション挿入数
SHR_WORKSPACE_SECTION_LOOKUPS	BIGINT	shr_workspace_section_lookups - 共有ワークスペース・セクションの参照回数
SHR_WORKSPACE_SIZE_TOP	BIGINT	shr_workspace_size_top - 最大共有ワークスペース・サイズ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
CAT_CACHE_SIZE_TOP	BIGINT	cat_cache_size_top - カタログ・キャッシュ最高水準点
TOTAL_OLAP_FUNCS	BIGINT	total_olap_funcs - OLAP 関数の合計数
OLAP_FUNC_OVERFLOWS	BIGINT	olap_func_overflows - OLAP 関数のオーバーフロー
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
<p>* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する <code>_S</code> で終わっている列で報告されている整数秒と、このモニター・エレメントに関する <code>_MS</code> で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません: <math>(\text{monitor-element-name}_S \times 1,000,000 + \text{monitor-element-name}_MS) \div 1,000,000</math>. 例えば、<math>(\text{ELAPSED\_EXEC\_TIME}_S \times 1,000,000 + \text{ELAPSED\_EXEC\_TIME}_MS) \div 1,000,000</math>.</p>		

## SNAPBP 管理ビューおよび SNAP\_GET\_BP 表関数 - bufferpool 論理グループのスナップショット情報の検索

SNAPBP 管理ビューおよび SNAP\_GET\_BP 表関数は、bufferpool スナップショットから、特に bufferpool 論理データ・グループのバッファ・プール情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 863 ページの『SNAPBP 管理ビュー』
- 864 ページの『SNAP\_GET\_BP 表関数』

## SNAPBP 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関する bufferpool 論理グループのスナップショット情報を取得することができます。

SNAPBP 管理ビューを SNAPBP\_PART 管理ビューとともに使用すると、**GET SNAPSHOT FOR BUFFERPOOLS ON database-alias** CLP コマンドに相当するデータが提供されます。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、866 ページの表 206 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPBP 管理ビューに対する SELECT 特権
- SNAPBP 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_BP 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続中のデータベースのすべてのバッファ・プールについて、データおよび索引の書き込みを取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME,SUBSTR(BP_NAME,1,15)  
      AS BP_NAME,POOL_DATA_WRITES,POOL_INDEX_WRITES  
FROM SYSIBMADM.SNAPBP
```

以下はこの照会の出力例です。

DB_NAME	BP_NAME	POOL_DATA_WRITES	POOL_INDEX_WRITES
TEST	IBMDEFAULTBP	0	0
TEST	IBMSYSTEMBP4K	0	0
TEST	IBMSYSTEMBP8K	0	0
TEST	IBMSYSTEMBP16K	0	0
TEST	IBMSYSTEMBP32K	0	0

5 record(s) selected

## SNAP\_GET\_BP 表関数

SNAP\_GET\_BP 表関数は SNAPBP 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_BP 表関数を SNAP\_GET\_BP\_PART 表関数とともに使用すると、**GET SNAPSHOT FOR ALL BUFFERPOOLS CLP** コマンドに相当するデータが提供されます。

戻される可能性のある情報の完全なリストは、866 ページの表 206 を参照してください。

## 構文

```
▶▶ SNAP_GET_BP ( (dbname [ , member ] ) )
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求を持つファイルが存在しない場合、SNAP\_GET\_BP 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_BP 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベース・メンバーのすべてのアクティブ・データベースのすべてのバッファ・プールについて、物理および論理読み取りの合計を取得します。

```
SELECT SUBSTR(T.DB_NAME,1,10) AS DB_NAME,
       SUBSTR(T.BP_NAME,1,20) AS BP_NAME,
       (T.POOL_DATA_L_READS+T.POOL_INDEX_L_READS) AS TOTAL_LOGICAL_READS,
       (T.POOL_DATA_P_READS+T.POOL_INDEX_P_READS) AS TOTAL_PHYSICAL_READS,
       T.DBPARTITIONNUM
FROM TABLE(SNAP_GET_BP(CAST(NULL AS VARCHAR(128)), -1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	BP_NAME	TOTAL_LOGICAL_READS	...
SAMPLE	IBMDEFAULTBP	0	...
TOOLSDB	IBMDEFAULTBP	0	...
TOOLSDB	BP32K0000	0	...

3 record(s) selected.

この照会からの出力 (続き)。

...	TOTAL_PHYSICAL_READS	DBPARTITIONNUM
...	0	0
...	0	0
...	0	0

## 戻される情報

表 206. SNAPBP 管理ビューおよび SNAP\_GET\_BP 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
BP_NAME	VARCHAR(128)	bp_name - バッファ・プール名
DB_NAME	VARCHAR(128)	db_name - データベース名
DB_PATH	VARCHAR(1024)	db_path - データベース・パス
INPUT_DB_ALIAS	VARCHAR(128)	input_db_alias - 入力データベース別名
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込み
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDA データの書き込み
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - バッファ・プール非同期データ読み取り
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - バッファ・プール非同期データ書き込み
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - バッファ・プール非同期索引読み取り
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - バッファ・プール非同期索引書き込み
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - バッファ・プール非同期 XDA データ読み取り



表 206. SNAPBP 管理ビューおよび SNAP\_GET\_BP 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - バッファ ー・プール非同期 XDA データ書 き込み
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - バッファ ー・プール非同期読み取り時間
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - バッファ ー・プール非同期書き込み時間
POOL_ASYNC_DATA_ READ_REQS	BIGINT	pool_async_data_read_reqs - バッファ ー・プール非同期読み取り要求
POOL_ASYNC_INDEX_ READ_REQS	BIGINT	pool_async_index_read_reqs - バッファ ー・プール非同期索引読み取り要 求
POOL_ASYNC_XDA_ READ_REQS	BIGINT	pool_async_xda_read_reqs - バッファ ー・プール非同期 XDA 読み取 り要求
DIRECT_READS	BIGINT	direct_reads - データベースからの 直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直 接書き込み
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要 求
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時 間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時 間
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不 能プリフェッチ・ページ
FILES_CLOSED	BIGINT	files_closed - 閉じられたデータベ ース・ファイル
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ ー・プール一時データの論理読み取 り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ ー・プール一時データの物理読み取 り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ ー・プール一時索引の論理読み取 り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ ー・プール一時索引の物理読み取 り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ ー・プール一時 XDA データの論 理読み取り

表 206. SNAPBP 管理ビューおよび SNAP\_GET\_BP 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ ー・プール一時 XDA データの物 理読み取り : モニター・エレメン ト
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - バッファ ー・プールの非ビクティム・バッフ ァー数
PAGES_FROM_BLOCK_IOS	BIGINT	pages_from_block_ios - ブロック入 出力によって読み取られたページ数 の合計
PAGES_FROM_VECTORED_IOS	BIGINT	pages_from_vectored_ios - ベクトル 化入出力によって読み取られたペ ージ数の合計
VECTORED_IOS	BIGINT	vectored_ios - ベクトル化入出力要 求数
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パ ーティション番号モニター・エレメ ント
MEMBER	SMALLINT	member - データベース・メンバ ー・モニター・エレメント

### SNAPBP\_PART 管理ビューおよび SNAP\_GET\_BP\_PART 表関数 - bufferpool\_nodeinfo 論理データ・グループのスナップショット情報の 検索

SNAPBP\_PART 管理ビューおよび SNAP\_GET\_BP\_PART 表関数は、バッファ  
ー・プール・スナップショットから、特に bufferpool\_nodeinfo 論理データ・グループの  
バッファ  
ー・プール情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセ  
クションを参照してください。

- 『SNAPBP\_PART 管理ビュー』
- 870 ページの 『SNAP\_GET\_BP\_PART 表関数』

#### SNAPBP\_PART 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関する bufferpool\_nodeinfo  
論理データ・グループのスナップショット情報を取得することができます。

SNAPBP\_PART 管理ビューを SNAPBP 管理ビューとともに使用すると、**GET  
SNAPSHOT FOR BUFFERPOOLS ON database-alias** CLP コマンドに相当するデータが  
提供されます。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、872 ページの表 207 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPBP\_PART 管理ビューに対する SELECT 特権
- SNAPBP\_PART 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_BP\_PART 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

SAMPLE データベースへの接続中に、すべてのバッファ・プールのデータを取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, SUBSTR(BP_NAME,1,15) AS BP_NAME,  
       BP_CUR_BUFFSZ, BP_NEW_BUFFSZ, BP_PAGES_LEFT_TO_REMOVE, BP_TBSP_USE_COUNT  
FROM SYSIBMADM.SNAPBP_PART
```

以下はこの照会の出力例です。

DB_NAME	BP_NAME	BP_CUR_BUFFSZ	BP_NEW_BUFFSZ	...
SAMPLE	IBMDEFAULTBP	1000	1000	...
SAMPLE	IBMSYSTEMBP4K	16	16	...
SAMPLE	IBMSYSTEMBP8K	16	16	...
SAMPLE	IBMSYSTEMBP16K	16	16	...

4 record(s) selected.

この照会からの出力 (続き)。

```

... BP_PAGES_LEFT_TO_REMOVE BP_TBSP_USE_COUNT
... -----
...                0                3
...                0                0
...                0                0
...                0                0
...

```

## SNAP\_GET\_BP\_PART 表関数

SNAP\_GET\_BP\_PART 表関数は SNAPBP\_PART 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_BP\_PART 表関数を SNAP\_GET\_BP 表関数とともに使用すると、**GET SNAPSHOT FOR ALL BUFFERPOOLS** CLP コマンドに相当するデータが提供されます。

戻される可能性のある情報の完全なリストは、872 ページの表 207 を参照してください。

## 構文

```

▶▶ SNAP_GET_BP_PART ( (dbname [ , member ] ) )

```

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースにある、すべてのバッファー・プールのスナップショットを取得するには、NULL 値を指定します。

### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり dbname のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

dbname と member の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限ら

ないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_BP\_PART 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_BP\_PART 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

SAMPLE データベースへの接続中に、すべてのアクティブ・データベースのすべてのバッファ・プールのデータを取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, SUBSTR(BP_NAME,1,15) AS BP_NAME,
       BP_CUR_BUFFSZ, BP_NEW_BUFFSZ, BP_PAGES_LEFT_TO_REMOVE, BP_TBSP_USE_COUNT
FROM TABLE(SNAP_GET_BP_PART(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	BP_NAME	BP_CUR_BUFFSZ	BP_NEW_BUFFSZ	...
SAMPLE	IBMDEFAULTBP	250	250	...
SAMPLE	IBMSYSTEMBP4K	16	16	...
SAMPLE	IBMSYSTEMBP8K	16	16	...
SAMPLE	IBMSYSTEMBP16K	16	16	...
SAMPLE	IBMSYSTEMBP32K	16	16	...
TESTDB	IBMDEFAULTBP	250	250	...
TESTDB	IBMSYSTEMBP4K	16	16	...
TESTDB	IBMSYSTEMBP8K	16	16	...
TESTDB	IBMSYSTEMBP16K	16	16	...
TESTDB	IBMSYSTEMBP32K	16	16	...

...

この照会からの出力 (続き)。

...	BP_PAGES_LEFT_TO_REMOVE	BP_TBSP_USE_COUNT
...	0	3
...	0	0
...	0	0
...	0	0
...	0	0
...	0	3

```

...           0           0
...           0           0
...           0           0
...           0           0
...

```

## 戻される情報

表 207. SNAPBP\_PART 管理ビューおよび SNAP\_GET\_BP\_PART 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
BP_NAME	VARCHAR(128)	bp_name - バッファ・プール名
DB_NAME	VARCHAR(128)	db_name - データベース名
BP_CUR_BUFFSZ	BIGINT	bp_cur_buffsz - バッファ・プールの現行サイズ
BP_NEW_BUFFSZ	BIGINT	bp_new_buffsz - 新規バッファ・プール・サイズ
BP_PAGES_LEFT_TO_REMOVE	BIGINT	bp_pages_left_to_remove - 除去残ページ数
BP_TBSP_USE_COUNT	BIGINT	bp_tbsp_use_count - バッファ・プールにマップされている表スペースの数
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPCONTAINER 管理ビューおよび SNAP\_GET\_CONTAINER 表関数 - tablespace\_container 論理データ・グループのスナップショット情報の検索

SNAPCONTAINER 管理ビューおよび SNAP\_GET\_CONTAINER 表関数は、tablespace\_container 論理データ・グループからの表スペース・スナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPCONTAINER 管理ビュー』
- 874 ページの『SNAP\_GET\_CONTAINER 表関数』

## SNAPCONTAINER 管理ビュー

この管理ビューでは、現在接続されているデータベースの tablespace\_container 論理データ・グループ・スナップショット情報を検索できます。

SNAPTBSP、SNAPTBSP\_PART、SNAPTBSP\_QUIESCER、および SNAPTBSP\_RANGE 管理ビューと共に使用すると、SNAPCONTAINER 管理ビューは、**GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等のデータを戻します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、876 ページの表 208 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPCONTAINER 管理ビューに対する SELECT 特権
- SNAPCONTAINER 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_CONTAINER 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースのすべてのデータベース・メンバーの表スペース・コンテナの詳細を検索します。

```
SELECT SNAPSHOT_TIMESTAMP, SUBSTR(TBSP_NAME, 1, 15) AS TBSP_NAME,  
       TBSP_ID, SUBSTR(CONTAINER_NAME, 1, 20) AS CONTAINER_NAME,  
       CONTAINER_ID, CONTAINER_TYPE, ACCESSIBLE, DBPARTITIONNUM  
FROM SYSIBMADM.SNAPCONTAINER ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

```
SNAPSHOT_TIMESTAMP      TBSP_NAME      TBSP_ID  ...  
-----  
2006-01-08-16.49.24.639945 SYSCATSPACE      0  ...
```

```

2006-01-08-16.49.24.639945 TEMPSPACE1      1 ...
2006-01-08-16.49.24.639945 USERSPACE1      2 ...
2006-01-08-16.49.24.639945 SYSTOOLSPACE  3 ...
2006-01-08-16.49.24.640747 TEMPSPACE1      1 ...
2006-01-08-16.49.24.640747 USERSPACE1      2 ...
2006-01-08-16.49.24.639981 TEMPSPACE1      1 ...
2006-01-08-16.49.24.639981 USERSPACE1      2 ...
...

```

8 record(s) selected.

この照会からの出力 (続き)。

```

... CONTAINER_NAME      CONTAINER_ID  CONTAINER_TYPE  ...
... -----
... /home/swalkty/swalkt      0  FILE_EXTENT_TAG  ...
... /home/swalkty/swalkt      0  PATH              ...
... /home/swalkty/swalkt      0  FILE_EXTENT_TAG  ...
... /home/swalkty/swalkt      0  FILE_EXTENT_TAG  ...
... /home/swalkty/swalkt      0  PATH              ...
... /home/swalkty/swalkt      0  FILE_EXTENT_TAG  ...
... /home/swalkty/swalkt      0  PATH              ...
... /home/swalkty/swalkt      0  FILE_EXTENT_TAG  ...

```

この照会からの出力 (続き)。

```

... ACCESSIBLE DBPARTITIONNUM
... -----
...          1          0
...          1          0
...          1          0
...          1          0
...          1          1
...          1          1
...          1          2
...          1          2

```

## SNAP\_GET\_CONTAINER 表関数

SNAP\_GET\_CONTAINER 表関数は SNAPCONTAINER 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_TBSP、SNAP\_GET\_TBSP\_PART、SNAP\_GET\_TBSP\_QUIESCER、および SNAP\_GET\_TBSP\_RANGE 表関数と共に使用すると、SNAP\_GET\_CONTAINER 表関数は、**GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等のデータを戻します。

戻される可能性のある情報の完全なリストは、876 ページの表 208 を参照してください。

## 構文

```

▶▶ SNAP_GET_CONTAINER ( ( dbname [ , member ] ) )

```

スキーマは SYSPROC です。



## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはデータベースがアクティブになっているすべてのメンバーから戻されます。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_CONTAINER 表関数は、現在接続中のデータベースおよびデータベース・メンバーのスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_CONTAINER 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベース・メンバー上で現在接続されているデータベースの表スペース・コンテナの詳細を検索します。

```
SELECT SNAPSHOT_TIMESTAMP, TBSP_NAME, TBSP_ID, CONTAINER_NAME,  
CONTAINER_ID, CONTAINER_TYPE, ACCESSIBLE  
FROM TABLE(SNAP_GET_CONTAINER('',-1)) AS T
```

以下はこの照会の出力例です。

```

SNAPSHOT_TIMESTAMP          TBSP_NAME          TBSP_ID ...
-----
2005-04-25-14.42.10.899253 SYSCATSPACE          0 ...
2005-04-25-14.42.10.899253 TEMPSPACE1           1 ...
2005-04-25-14.42.10.899253 USERSPACE1           2 ...
2005-04-25-14.42.10.899253 SYSTOOLSPACE        3 ...
2005-04-25-14.42.10.899253 MYTEMP               4 ...
2005-04-25-14.42.10.899253 WHATSNEWTMPSPACE    5 ...

```

この照会からの出力 (続き)。

```

... CONTAINER_NAME          CONTAINER_ID ...
-----
... D:¥DB2¥NODE0000¥SQL00002¥SFLT0000.0 0 ...
... D:¥DB2¥NODE0000¥SQL00002¥SFLT0001.0 0 ...
... D:¥DB2¥NODE0000¥SQL00002¥SFLT0002.0 0 ...
... D:¥DB2¥NODE0000¥SQL00002¥SYSTOOLSPACE 0 ...
... D:¥DB2¥NODE0000¥SQL003                0 ...
... d:¥DGTsWhatsNewContainer             0 ...

```

この照会からの出力 (続き)。

```

... CONTAINER_TYPE ACCESSIBLE
... -----
... CONT_PATH              1
... CONT_PATH              1
... CONT_PATH              1
... CONT_PATH              1
... CONT_PATH              1
... CONT_PATH              1
... CONT_PATH              1

```

## 戻される情報

注: ファイル・システム情報を戻すためには、BUFFERPOOL データベース・マネージャのモニター・スイッチをオンにする必要があります。

表 208. SNAPCONTAINER 管理ビューおよび SNAP\_GET\_CONTAINER 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
CONTAINER_NAME	VARCHAR(256)	container_name - コンテナ名
CONTAINER_ID	BIGINT	container_id - コンテナ ID
CONTAINER_TYPE	VARCHAR(16)	container_type - コンテナ・タイプ。これは、sqlutil.h での定義を基にしたテキスト ID です。以下のいずれかとなります。 <ul style="list-style-type: none"> <li>• DISK_EXTENT_TAG</li> <li>• DISK_PAGE_TAG</li> <li>• FILE_EXTENT_TAG</li> <li>• FILE_PAGE_TAG</li> <li>• PATH</li> </ul>

表 208. SNAPCONTAINER 管理ビューおよび SNAP\_GET\_CONTAINER 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_PAGES	BIGINT	container_total_pages - コンテナ内の合計ページ数
USABLE_PAGES	BIGINT	container_usable_pages - コンテナ内の使用可能なページ数
ACCESSIBLE	SMALLINT	container_accessible - コンテナのアクセス可能性
STRIPE_SET	BIGINT	container_stripe_set - ストライプ・セット
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
FS_ID	VARCHAR(22)	fs_id - 固有のファイル・システム識別番号
FS_TOTAL_SIZE	BIGINT	fs_total_size - ファイル・システムの合計サイズ
FS_USED_SIZE	BIGINT	fs_used_size - ファイル・システム上で使用されるスペースの量

## SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数 - dbase 論理グループからのスナップショット情報の取得

SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数は、データベース (dbase) 論理グループからのスナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPDB 管理ビュー』
- 879 ページの 『SNAP\_GET\_DB 表関数』

### SNAPDB 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関するスナップショット情報を dbase 論理グループから検索できます。

SNAPDB 管理ビューを ADMIN\_GET\_STORAGE\_PATHS 表関数、MON\_GET\_MEMORY\_POOL、MON\_GET\_TRANSACTION\_LOG、および MON\_GET\_HADR と併せて使用することにより、**GET SNAPSHOT FOR DATABASE on database-alias** CLP コマンドと同等の情報が得られます。

スキーマは SYSIBMADM です。

戻される情報の完全なリストは、882 ページの表 209 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPDB 管理ビューに対する SELECT 特権
- SNAPDB 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_DB 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースのすべてのデータベース・メンバーの状況、プラットフォーム、ロケーション、および接続時間を取り出します。

```
SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,  
       DB_LOCATION, DB_CONN_TIME, DBPARTITIONNUM  
FROM SYSIBMADM.SNAPDB ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

DB_NAME	DB_STATUS	SERVER_PLATFORM	DB_LOCATION	...
TEST	ACTIVE	AIX64	LOCAL	...
TEST	ACTIVE	AIX64	LOCAL	...
TEST	ACTIVE	AIX64	LOCAL	...

3 record(s) selected.

この照会からの出力 (続き)。

...	DB_CONN_TIME	DBPARTITIONNUM
...	2006-01-08-16.48.30.665477	0
...	2006-01-08-16.48.34.005328	1
...	2006-01-08-16.48.34.007937	2

このルーチンは、コマンド行で以下を呼び出すことにより使用できます。

```
SELECT TOTAL_OLAP_FUNCS, OLAP_FUNC_OVERFLOWS, ACTIVE_OLAP_FUNCS
FROM SYSIBMADM.SNAPDB
```

```
TOTAL_OLAP_FUNCS    OLAP_FUNC_OVERFLOWS  ACTIVE_OLAP_FUNCS
-----
                      7                      2                      1
```

1 record(s) selected.

ワークロードの実行後に、ユーザーは次の照会を使用できます。

```
SELECT STATS_CACHE_SIZE, STATS_FABRICATIONS, SYNC_RUNSTATS,
ASYNC_RUNSTATS, STATS_FABRICATE_TIME, SYNC_RUNSTATS_TIME
FROM SYSIBMADM.SNAPDB
```

```
STATS_CACHE_SIZE  STATS_FABRICATIONS  SYNC_RUNSTATS  ASYNC_RUNSTATS  ...
-----
                128                2                1                0  ...
```

```
... STATS_FABRICATE_TIME  SYNC_RUNSTATS_TIME
... -----
...                10                100
```

1 record(s) selected.

## SNAP\_GET\_DB 表関数

SNAP\_GET\_DB 表関数は、SNAPDB 管理ビューと同じ情報を戻します。

SNAP\_GET\_DB 表関数を ADMIN\_GET\_STORAGE\_PATHS 表関数、MON\_GET\_MEMORY\_POOL、MON\_GET\_TRANSACTION\_LOG、および MON\_GET\_HADR 表関数と併せて使用することにより、**GET SNAPSHOT FOR ALL DATABASES** CLP コマンドと同等の情報が得られます。

戻される情報の完全なリストは、882 ページの表 209 を参照してください。

## 構文

```
▶▶ SNAP_GET_DB ( (dbname [ , member ] ) )
```

スキーマは SYSPROC です。

## 表関数パラメーター

*dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

## member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり dbname のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されず。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

dbname と member の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DB 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DB 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: 現在接続されているデータベースのすべてのデータベース・メンバーを対象とした集約ビューとして、状況、プラットフォーム、ロケーション、および接続時間を取り出します。

```
SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,  
       DB_LOCATION, DB_CONN_TIME FROM TABLE(SNAP_GET_DB('', -2)) AS T
```

以下はこの照会の出力例です。

DB_NAME	DB_STATUS	SERVER_PLATFORM	...
-----	-----	-----	...
SAMPLE	ACTIVE	AIX64	...

1 record(s) selected.

この照会からの出力 (続き)。

```

... DB_LOCATION DB_CONN_TIME
... -----
... LOCAL      2005-07-24-22.09.22.013196

```

例 2: 現在接続されているデータベースが含まれる同じインスタンス内にあるすべてのアクティブ・データベースのすべてのデータベース・メンバーを対象とした集約ビューとして、状況、プラットフォーム、ロケーション、および接続時間を取り出します。

```

SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,
       DB_LOCATION, DB_CONN_TIME
FROM TABLE(SNAP_GET_DB(CAST (NULL AS VARCHAR(128)), -2)) AS T

```

以下はこの照会の出力例です。

```

DB_NAME      DB_STATUS  SERVER_PLATFORM ...
-----
TOOLSDB     ACTIVE     AIX64           ...
SAMPLE      ACTIVE     AIX64           ...

```

この照会からの出力 (続き)。

```

... DB_LOCATION DB_CONN_TIME
... -----
... LOCAL      2005-07-24-22.26.54.396335
... LOCAL      2005-07-24-22.09.22.013196

```

例 3: このルーチンは、データベースへの接続時にコマンド行で以下を呼び出すことにより使用できます。

```

SELECT TOTAL_OLAP_FUNCS, OLAP_FUNC_OVERFLOWS, ACTIVE_OLAP_FUNCS
FROM TABLE (SNAP_GET_DB('', 0)) AS T

```

出力は次のようになります。

```

TOTAL_OLAP_FUNCS  OLAP_FUNC_OVERFLOWS  ACTIVE_OLAP_FUNCS
-----
                7                2                1

```

1 record(s) selected.

例 4: ワークロードの実行後に、ユーザーは次の照会を表関数とともに使用できます。

```

SELECT STATS_CACHE_SIZE, STATS_FABRICATIONS, SYNC_RUNSTATS,
       ASYNC_RUNSTATS, STATS_FABRICATE_TIME, SYNC_RUNSTATS_TIME
FROM TABLE (SNAP_GET_DB('mytestdb', -1)) AS SNAPDB

```

```

STATS_CACHE_SIZE  STATS_FABRICATIONS  SYNC_RUNSTATS  ASYNC_RUNSTATS ...
-----
                200                1                2                0 ...

```

Continued

```

...STATS_FABRICATE_TIME  SYNC_RUNSTATS_TIME
...-----
...                2                32

```

1 record(s) selected.

例 5: 以下の例は、SNAP\_GET\_DB 表関数を使用してデータベースの状況を判別する方法を示しています。

```
SELECT SUBSTR
(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS
FROM table(SNAP_GET_DB('hadrdbr', 0))
```

```
DB_NAME          DB_STATUS
-----
HADRDDB          ACTIVE_STANDBY
```

### 戻される情報

表 209. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
DB_PATH	VARCHAR(1024)	db_path - データベース・パス
INPUT_DB_ALIAS	VARCHAR(128)	input_db_alias - 入力データベース別名
DB_STATUS	VARCHAR(16)	db_status - データベース状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• ACTIVE</li> <li>• QUIESCE_PEND</li> <li>• QUIESCED</li> <li>• ROLLFWD</li> <li>• ACTIVE_STANDBY - HADR データベースはスタンバイ・モードで、スタンバイ中の読み取りが有効になっています。</li> <li>• STANDBY - HADR データベースはスタンバイ・モードです (スタンバイ中の読み取りは有効になっていません)。</li> </ul>
CATALOG_PARTITION	SMALLINT	catalog_node - カタログ・データベース・パーティション番号
CATALOG_PARTITION_NAME	VARCHAR(128)	catalog_node_name - カタログ・データベース・パーティション・ネットワーク名



表 209. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SERVER_PLATFORM	VARCHAR(12)	<p>server_platform - サーバーのオペレーティング・システム。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• AIX</li> <li>• AIX64</li> <li>• AS400_DRDA</li> <li>• DOS</li> <li>• DYNIX</li> <li>• HP</li> <li>• HP64</li> <li>• HPIA</li> <li>• HPIA64</li> <li>• LINUX</li> <li>• LINUX390</li> <li>• LINUXIA64</li> <li>• LINUXPPC</li> <li>• LINUXPPC64</li> <li>• LINUXX8664</li> <li>• LINUXZ64</li> <li>• MAC</li> <li>• MVS_DRDA</li> <li>• NT</li> <li>• NT64</li> <li>• OS2</li> <li>• OS390</li> <li>• SCO</li> <li>• SGI</li> <li>• SNI</li> <li>• SUN</li> <li>• SUN64</li> <li>• UNKNOWN</li> <li>• UNKNOWN_DRDA</li> <li>• VM_DRDA</li> <li>• VSE_DRDA</li> <li>• WINDOWS</li> </ul>

表 209. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DB_LOCATION	VARCHAR(12)	db_location - データベース・ロケーション。 このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。  • LOCAL • REMOTE
DB_CONN_TIME	TIMESTAMP	db_conn_time- データベース・アクティブ化タイム・スタンプ
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ
LAST_BACKUP	TIMESTAMP	last_backup - 最終バックアップ・タイム・スタンプ
CONNECTIONS_TOP	BIGINT	connections_top - 同時接続の最大数
TOTAL_CONS	BIGINT	total_cons - データベース・アクティブ化以降の接続
TOTAL_SEC_CONS	BIGINT	total_sec_cons - 2 次接続
APPLS_CUR_CONS	BIGINT	appls_cur_cons - 現在接続されているアプリケーション
APPLS_IN_DB2	BIGINT	appls_in_db2 - データベースで現在実行中のアプリケーション
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 関連したエージェント数
AGENTS_TOP	BIGINT	agents_top - 作成されたエージェントの数
COORD_AGENTS_TOP	BIGINT	coord_agents_top - コーディネーター・エージェント最大数
LOCKS_HELD	BIGINT	locks_held - ロック保持数
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間
LOCK_LIST_IN_USE	BIGINT	lock_list_in_use - 使用中のロック・リスト・メモリーの合計
DEADLOCKS	BIGINT	deadlocks - デッドロック検出数
LOCK_ESCALS	BIGINT	lock_escalations - ロック・エスカレーション数
X_LOCK_ESCALS	BIGINT	x_lock_escalations - 排他ロック・エスカレーション数
LOCKS_WAITING	BIGINT	locks_waiting - ロックで待機中の現行エージェント
LOCK_TIMEOUTS	BIGINT	lock_timeouts - ロック・タイムアウト数
NUM_INDOUBT_TRANS	BIGINT	num_indoubt_trans - 未確定トランザクション数
SORT_HEAP_ALLOCATED	BIGINT	sort_heap_allocated - 割り振られたソート・ヒープの合計
SORT_SHRHEAP_ALLOCATED	BIGINT	sort_shrheap_allocated - 現在割り振られているソート共有ヒープ

表 209. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SORT_SHRHEAP_TOP	BIGINT	sort_shrheap_top - ソート共有ヒープの最高水準点
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - ポスト共有しきい値ソート
TOTAL_SORTS	BIGINT	total_sorts - ソート合計
TOTAL_SORT_TIME	BIGINT	total_sort_time - ソート時間合計
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
ACTIVE_SORTS	BIGINT	active_sorts - アクティブ・ソート
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - バッファ・プール非同期データ読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - バッファ・プール非同期データ書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - バッファ・プール非同期索引読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込み
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - バッファ・プール非同期索引書き込み
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDAデータの物理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDAデータの論理読み取り
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDAデータの書き込み

表 209. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - バッファ・プール非同期 XDA データ読み取り
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - バッファ・プール非同期 XDA データ書き込み
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り : モニター・エレメント
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - バッファ・プール非同期読み取り時間
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - バッファ・プール非同期書き込み時間
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - バッファ・プール非同期読み取り要求
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - バッファ・プール非同期索引読み取り要求
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - バッファ・プール非同期 XDA 読み取り要求
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - バッファ・プールの非ビクティム・バッファ数
POOL_LSN_GAP_CLNS	BIGINT	pool_lsn_gap_clns - 起動されたバッファ・プール・ログ・スペース・クリーナー
POOL_DRTY_PG_STEAL_CLNS	BIGINT	pool_drty_pg_steal_clns - 起動されたバッファ・プール・ビクティム・ページ・クリーナー
POOL_DRTY_PG_THRSH_CLNS	BIGINT	pool_drty_pg_thrsh_clns - 起動されたバッファ・プールしきい値クリーナー
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不能プリフェッチ・ページ
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間

表 209. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
FILES_CLOSED	BIGINT	files_closed - 閉じられたデータベース・ファイル
ELAPSED_EXEC_TIME_S	BIGINT	elapsed_exec_time - ステートメント実行経過時間
ELAPSED_EXEC_TIME_MS	BIGINT	elapsed_exec_time - ステートメント実行経過時間
COMMIT_SQL_STMTS	BIGINT	commit_sql_stmts - 試行されたコミット・ステートメント
ROLLBACK_SQL_STMTS	BIGINT	rollback_sql_stmts - 試行されたロールバック・ステートメント
DYNAMIC_SQL_STMTS	BIGINT	dynamic_sql_stmts - 試行された動的 SQL ステートメント
STATIC_SQL_STMTS	BIGINT	static_sql_stmts - 試行された静的 SQL ステートメント
FAILED_SQL_STMTS	BIGINT	failed_sql_stmts - 失敗したステートメント操作
SELECT_SQL_STMTS	BIGINT	select_sql_stmts - 実行された選択 SQL ステートメント
UID_SQL_STMTS	BIGINT	uid_sql_stmts - 実行された更新/挿入/削除 SQL ステートメント
DDL_SQL_STMTS	BIGINT	ddl_sql_stmts - データ定義言語 (DDL) SQL ステートメント
INT_AUTO_REBINDS	BIGINT	int_auto_rebinds - 内部自動再バインド
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 削除された内部行数
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 挿入された内部行数
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 更新された内部行数
INT_COMMITS	BIGINT	int_commits - 内部コミット数
INT_ROLLBACKS	BIGINT	int_rollback - 内部ロールバック数
INT_DEADLOCK_ROLLBACKS	BIGINT	int_deadlock_rollback - デッドロックによる内部ロールバック数
ROWS_DELETED	BIGINT	rows_deleted - 削除行数
ROWS_INSERTED	BIGINT	rows_inserted - 挿入行数
ROWS_UPDATED	BIGINT	rows_updated - 更新行数
ROWS_SELECTED	BIGINT	rows_selected - 選択行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
BINDS_PRECOMPILES	BIGINT	binds_precompiles - 試行されたバインド/プリコンパイル
TOTAL_LOG_AVAILABLE	BIGINT	total_log_available - 使用可能なログ合計
TOTAL_LOG_USED	BIGINT	total_log_used - 使用されているログ・スペースの合計
SEC_LOG_USED_TOP	BIGINT	sec_log_used_top - 使用された最大 2 次ログ・スペース

表 209. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOT_LOG_USED_TOP	BIGINT	tot_log_used_top - 使用された最大合計ログ・スペース
SEC_LOGS_ALLOCATED	BIGINT	sec_logs_allocated - 現在割り振られている 2 次ログ
LOG_READS	BIGINT	log_reads - 読み取られたログ・ページの数
LOG_READ_TIME_S	BIGINT	log_read_time - ログ読み取り時間
LOG_READ_TIME_NS	BIGINT	log_read_time - ログ読み取り時間
LOG_WRITES	BIGINT	log_writes - 書き込まれたログ・ページの数
LOG_WRITE_TIME_S	BIGINT	log_write_time - ログ書き込み時間
LOG_WRITE_TIME_NS	BIGINT	log_write_time - ログ書き込み時間
NUM_LOG_WRITE_IO	BIGINT	num_log_write_io - ログ書き込み数
NUM_LOG_READ_IO	BIGINT	num_log_read_io - ログ読み取り数
NUM_LOG_PART_PAGE_IO	BIGINT	num_log_part_page_io - 部分ログ・ページ書き込み数
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - フル・ログ・バッファの回数
NUM_LOG_DATA_FOUND_IN_BUFFER	BIGINT	num_log_data_found_in_buffer - ログ・データがバッファにある回数
APPL_ID_OLDEST_XACT	BIGINT	appl_id_oldest_xact - 最も古いトランザクションを使用するアプリケーション
LOG_TO_REDO_FOR_RECOVERY	BIGINT	log_to_redo_for_recovery - リカバリーの場合に再実行されるログの量
LOG_HELD_BY_DIRTY_PAGES	BIGINT	log_held_by_dirty_pages - ダーティー・ページ別に計算されるログ・スペースの量
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - パッケージ・キャッシュ参照
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - パッケージ・キャッシュ挿入
PKG_CACHE_NUM_OVERFLOWES	BIGINT	pkg_cache_num_overflows - パッケージ・キャッシュ・オーバーフロー数
PKG_CACHE_SIZE_TOP	BIGINT	pkg_cache_size_top - パッケージ・キャッシュ最高水準点
APPL_SECTION_LOOKUPS	BIGINT	appl_section_lookups - セクションの参照回数
APPL_SECTION_INSERTS	BIGINT	appl_section_inserts - セクション挿入数
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - カタログ・キャッシュ参照数
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - カタログ・キャッシュ挿入数
CAT_CACHE_OVERFLOWES	BIGINT	cat_cache_overflows - カタログ・キャッシュ・オーバーフロー数
CAT_CACHE_SIZE_TOP	BIGINT	cat_cache_size_top - カタログ・キャッシュ最高水準点

表 209. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
PRIV_WORKSPACE_SIZE_TOP	BIGINT	priv_workspace_size_top - 専用ワークスペースの最大サイズ
PRIV_WORKSPACE_NUM_OVERFLOWS	BIGINT	priv_workspace_num_overflows - 専用ワークスペースのオーバーフロー回数
PRIV_WORKSPACE_SECTION_INSERTS	BIGINT	priv_workspace_section_inserts - 専用ワークスペース・セクション挿入
PRIV_WORKSPACE_SECTION_LOOKUPS	BIGINT	priv_workspace_section_lookups - 専用ワークスペース・セクションの参照
SHR_WORKSPACE_SIZE_TOP	BIGINT	shr_workspace_size_top - 最大共有ワークスペース・サイズ
SHR_WORKSPACE_NUM_OVERFLOWS	BIGINT	shr_workspace_num_overflows - 共有ワークスペースのオーバーフロー回数
SHR_WORKSPACE_SECTION_INSERTS	BIGINT	shr_workspace_section_inserts - 共有ワークスペース・セクション挿入数
SHR_WORKSPACE_SECTION_LOOKUPS	BIGINT	shr_workspace_section_lookups - 共有ワークスペース・セクションの参照回数
TOTAL_HASH_JOINS	BIGINT	total_hash_joins - ハッシュ結合の合計
TOTAL_HASH_LOOPS	BIGINT	total_hash_loops - ハッシュ・ループの合計
HASH_JOIN_OVERFLOWS	BIGINT	hash_join_overflows - ハッシュ結合のオーバーフロー
HASH_JOIN_SMALL_OVERFLOWS	BIGINT	hash_join_small_overflows - ハッシュ結合の短精度オーバーフロー
POST_SHRTHRESHOLD_HASH_JOINS	BIGINT	post_shrthreshold_hash_joins - ポストしきい値ハッシュ結合
ACTIVE_HASH_JOINS	BIGINT	active_hash_joins - アクティブ・ハッシュ結合
NUM_DB_STORAGE_PATHS	BIGINT	num_db_storage_paths - 自動ストレージ・パスの数
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
SMALLEST_LOG_AVAIL_NODE	INTEGER	smallest_log_avail_node - 使用可能なログ・スペースが最小のノード
TOTAL_OLAP_FUNCS	BIGINT	total_olap_funcs - OLAP 関数の合計数
OLAP_FUNC_OVERFLOWS	BIGINT	olap_func_overflows - OLAP 関数のオーバーフロー
ACTIVE_OLAP_FUNCS	BIGINT	active_olap_funcs - アクティブ OLAP 関数
STATS_CACHE_SIZE	BIGINT	stats_cache_size - 統計キャッシュのサイズ
STATS_FABRICATIONS	BIGINT	stats_fabrications - 統計作成の合計数
SYNC_RUNSTATS	BIGINT	sync_runstats - 同期 RUNSTATS アクティビティの合計数
ASYNC_RUNSTATS	BIGINT	async_runstats - 非同期 RUNSTATS 要求の合計数
STATS_FABRICATE_TIME	BIGINT	stats_fabricate_time - 統計作成アクティビティに費やされた合計時間

表 209. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SYNC_RUNSTATS_TIME	BIGINT	sync_runstats_time - 同期 RUNSTATS アクティビティに費やされた合計時間
NUM_THRESHOLD_VIOLATIONS	BIGINT	num_threshold_violations - しきい値違反の回数
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPDBM 管理ビューおよび SNAP\_GET\_DBM 表関数 - dbm 論理グループ・スナップショット情報の検索

SNAPDBM 管理ビューおよび SNAP\_GET\_DBM 表関数は、スナップショット・モニターに、DB2 database データベース・マネージャー (dbm) 論理グループ情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPDBM 管理ビュー』
- 891 ページの 『SNAP\_GET\_DBM 表関数』

### SNAPDBM 管理ビュー

SNAPDBM\_MEMORY\_POOL、SNAPFCM、SNAPFCM\_PART、および SNAPSWITCHES 管理ビューと共に使用すると、SNAPDBM 管理ビューは、**GET SNAPSHOT FOR DBM** コマンドと同等のデータを提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、893 ページの表 210 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPDBM 管理ビューに対する SELECT 特権
- SNAPDBM 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_DBM 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限



さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

すべてのデータベース・メンバーにおけるデータベース・マネージャーの状況と接続情報を検索します。

```
SELECT DB2_STATUS, DB2START_TIME, LAST_RESET, LOCAL_CONS, REM_CONS_IN,
       (AGENTS_CREATED_EMPTY_POOL/AGENTS_FROM_POOL) AS AGENT_USAGE,
       DBPARTITIONNUM FROM SYSIBMADM.SNAPDBM ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

DB2_STATUS	DB2START_TIME	LAST_RESET	...
ACTIVE	2006-01-06-14.59.59.059879	-	...
ACTIVE	2006-01-06-14.59.59.097605	-	...
ACTIVE	2006-01-06-14.59.59.062798	-	...

3 record(s) selected. ...

この照会からの出力 (続き)。

...	LOCAL_CONS	REM_CONS_IN	AGENT_USAGE	DBPARTITIONNUM
...	1	1	0	0
...	0	0	0	1
...	0	0	0	2

### SNAP\_GET\_DBM 表関数

SNAP\_GET\_DBM 表関数は SNAPDBM 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーを対象とした情報を検索することができます。

SNAP\_GET\_DBM\_MEMORY\_POOL、SNAP\_GET\_FCM、SNAP\_GET\_FCM\_PART、および SNAP\_GET\_SWITCHES 表関数と共に使用すると、SNAP\_GET\_DBM 表関数は、GET SNAPSHOT FOR DBM コマンドと同等のデータを提供します。

戻される可能性のある情報の完全なリストは、893 ページの表 210 を参照してください。

### 構文

```
▶▶ SNAP_GET_DBM ( [ member ] ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DBM 表関数はメモリーからスナップショットを呼び出します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DBM 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

データベース・メンバー番号 2 の開始時刻と現行状況を検索します。

```
SELECT DB2START_TIME, DB2_STATUS FROM TABLE(SNAP_GET_DBM(2)) AS T
```

以下はこの照会の出力例です。

```
DB2START_TIME          DB2_STATUS
-----
2006-01-06-14.59.59.062798 ACTIVE
```

## 戻される情報

表 210. SNAPDBM 管理ビューおよび SNAP\_GET\_DBM 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
SORT_HEAP_ALLOCATED	BIGINT	sort_heap_allocated - 割り振られたソート・ヒープの合計
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - ポストしきい値ソート
PIPED_SORTS_REQUESTED	BIGINT	piped_sorts_requested - 要求されたパイプ・ソート数
PIPED_SORTS_ACCEPTED	BIGINT	piped_sorts_accepted - 受け入れられたパイプ・ソート
REM_CONS_IN	BIGINT	rem_cons_in - データベース・マネージャーへのリモート接続
REM_CONS_IN_EXEC	BIGINT	rem_cons_in_exec - データベース・マネージャーで実行中のリモート接続：モニター・エレメント
LOCAL_CONS	BIGINT	local_cons - ローカル接続
LOCAL_CONS_IN_EXEC	BIGINT	local_cons_in_exec - データベース・マネージャーで実行中のローカル接続：モニター・エレメント
CON_LOCAL_DBASES	BIGINT	con_local_dbases - 現行接続を使用したローカル・データベース
AGENTS_REGISTERED	BIGINT	agents_registered - 登録済みエージェント
AGENTS_WAITING_ON_TOKEN	BIGINT	agents_waiting_on_token - トークン待ちエージェント
DB2_STATUS	VARCHAR(12)	db2_status - DB2 インスタンス状況。  このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。  <ul style="list-style-type: none"> <li>• ACTIVE</li> <li>• QUIESCE_PEND</li> <li>• QUIESCED</li> </ul>
AGENTS_REGISTERED_TOP	BIGINT	agents_registered_top - エージェント最大登録数
AGENTS_WAITING_TOP	BIGINT	agents_waiting_top - エージェント最大待機数
COMM_PRIVATE_MEM	BIGINT	comm_private_mem - コミット済み専用メモリー
IDLE_AGENTS	BIGINT	idle_agents - アイドル・エージェント数
AGENTS_FROM_POOL	BIGINT	agents_from_pool - プールから割り当てられたエージェント
AGENTS_CREATED_EMPTY_POOL	BIGINT	agents_created_empty_pool - エージェント・プールが空のために作成されたエージェント

表 210. SNAPDBM 管理ビューおよび SNAP\_GET\_DBM 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
COORD_AGENTS_TOP	BIGINT	coord_agents_top - コーディネーター・エージェント最大数
MAX_AGENT_OVERFLOW	BIGINT	max_agent_overflows - 最大エージェント・オーバーフロー回数
AGENTS_STOLEN	BIGINT	agents_stolen - スチールされたエージェント
GW_TOTAL_CONS	BIGINT	gw_total_cons - DB2 Connect の接続試行合計回数
GW_CUR_CONS	BIGINT	gw_cur_cons - DB2 Connect の現在の接続数
GW_CONS_WAIT_HOST	BIGINT	gw_cons_wait_host - ホストの応答を待機している接続の数
GW_CONS_WAIT_CLIENT	BIGINT	gw_cons_wait_client - クライアントの要求送信を待機している接続の数
POST_THRESHOLD_HASH_JOINS	BIGINT	post_threshold_hash_joins - ハッシュ結合のしきい値
NUM_GW_CONN_SWITCHES	BIGINT	num_gw_conn_switches - 接続切り替え回数
DB2START_TIME	TIMESTAMP	db2start_time - データベース・マネージャー開始タイム・スタンプ
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ
NUM_NODES_IN_DB2_INSTANCE	INTEGER	num_nodes_in_db2_instance - データベース・パーティション内のノード数
PRODUCT_NAME	VARCHAR(32)	product_name - 製品名
SERVICE_LEVEL	VARCHAR(18)	service_level - サービス・レベル
SORT_HEAP_TOP	BIGINT	sort_heap_top - ソート専用ヒープの最高水準点
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
POST_THRESHOLD_OLAP_FUNCS	BIGINT	post_threshold_olap_funcs - OLAP 関数のしきい値
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

### SNAPDETAILLOG 管理ビューおよび SNAP\_GET\_DETAILLOG 表関数 - detail\_log 論理データ・グループからのスナップショット情報の検索

SNAPDETAILLOG 管理ビューおよび SNAP\_GET\_DETAILLOG 表関数は、detail\_log 論理データ・グループからのスナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 895 ページの『SNAPDETAILLOG 管理ビュー』
- 896 ページの『SNAP\_GET\_DETAILLOG 表関数』

## SNAPDETAILLOG 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関するスナップショット情報を detail\_log 論理データ・グループから検索できます。

SNAPDETAILLOG 管理ビューを ADMIN\_GET\_STORAGE\_PATHS、MON\_GET\_HADR、MON\_GET\_MEMORY\_POOL、および SNAPDB と併せて使用することにより、**GET SNAPSHOT FOR DATABASE on database-alias** CLP コマンドと同等の情報が得られます。

スキーマは SYSIBMADM です。

戻される情報の完全なリストは、898 ページの表 211 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPDETAILLOG 管理ビューに対する SELECT 特権
- SNAPDETAILLOG 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_DETAILLOG 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

すべてのデータベース・メンバーに関して、現在接続されているデータベースのログ情報を検索します。

```
SELECT SUBSTR(DB_NAME, 1, 8) AS DB_NAME, FIRST_ACTIVE_LOG,  
       LAST_ACTIVE_LOG, CURRENT_ACTIVE_LOG, CURRENT_ARCHIVE_LOG,  
       DBPARTITIONNUM  
FROM SYSIBMADM.SNAPDETAILLOG ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

```
DB_NAME FIRST_ACTIVE_LOG LAST_ACTIVE_LOG ...
-----
TEST          0          8 ...
TEST          0          8 ...
TEST          0          8 ...
```

3 record(s) selected.

この照会からの出力 (続き)。

```
... CURRENT_ACTIVE_LOG CURRENT_ARCHIVE_LOG DBPARTITIONNUM
... -----
...          0          -          0
...          0          -          1
...          0          -          2
```

## SNAP\_GET\_DETAILLOG 表関数

SNAP\_GET\_DETAILLOG 表関数は、SNAPDETAILLOG 管理ビューと同じ情報を戻します。

SNAPDETAILLOG 管理ビューを ADMIN\_GET\_STORAGE\_PATHS、MON\_GET\_HADR、MON\_GET\_MEMORY\_POOL、および SNAP\_GET\_DB と併せて使用することにより、GET SNAPSHOT FOR ALL DATABASES CLP コマンドと同等の情報が得られます。

戻される情報の完全なリストは、898 ページの表 211 を参照してください。

## 構文

```
▶▶ SNAP_GET_DETAILLOG ( (dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、LIST DATABASE DIRECTORY コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり dbname のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されま

す。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DETAILLOG 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DETAILLOG 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

データベース・メンバー 1 に関して、現在接続されているデータベースのログ情報を検索します。

```
SELECT SUBSTR(DB_NAME, 1, 8) AS DB_NAME, FIRST_ACTIVE_LOG,
       LAST_ACTIVE_LOG, CURRENT_ACTIVE_LOG, CURRENT_ARCHIVE_LOG
FROM TABLE(SNAP_GET_DETAILLOG('1', 1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	FIRST_ACTIVE_LOG	LAST_ACTIVE_LOG	...
TEST	0	8	...

1 record(s) selected.

この照会からの出力 (続き)。

...	CURRENT_ACTIVE_LOG	CURRENT_ARCHIVE_LOG
...	0	-

## 戻される情報

表 211. SNAPDETAILLOG 管理ビューおよび SNAP\_GET\_DETAILLOG 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
FIRST_ACTIVE_LOG	BIGINT	first_active_log - 先頭アクティブ・ログ・ファイル番号
LAST_ACTIVE_LOG	BIGINT	last_active_log - 最終アクティブ・ログ・ファイル番号
CURRENT_ACTIVE_LOG	BIGINT	current_active_log - 現行アクティブ・ログ・ファイル番号
CURRENT_ARCHIVE_LOG	BIGINT	current_archive_log - 現行アーカイブ・ログ・ファイル番号
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPDYN\_SQL 管理ビューおよび SNAP\_GET\_DYN\_SQL 表関数 - dynsql 論理グループのスナップショット情報の検索

SNAPDYN\_SQL 管理ビューおよび SNAP\_GET\_DYN\_SQL 表関数は、dynsql 論理データ・グループからのスナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPDYN\_SQL 管理ビュー』
- 900 ページの『SNAP\_GET\_DYN\_SQL 表関数』

### SNAPDYN\_SQL 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関する dynsql 論理グループのスナップショット情報を検索できます。

このビューは、**GET SNAPSHOT FOR DYNAMIC SQL ON database-alias CLP** コマンドと同等の情報を戻します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、903 ページの表 212 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPDYN\_SQL 管理ビューに対する SELECT 特権



- SNAPDYN\_SQL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかの権限が必要です。

- SNAP\_GET\_DYN\_SQL 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベース上で実行された動的 SQL のリストを、読み取られた行数の順序で取得します。

```
SELECT PREP_TIME_WORST, NUM_COMPILATIONS, SUBSTR(STMT_TEXT, 1, 60)
       AS STMT_TEXT, DBPARTITIONNUM
FROM SYSIBMADM.SNAPDYN_SQL ORDER BY ROWS_READ
```

以下はこの照会の出力例です。

```
PREP_TIME_WORST      NUM_COMPILATIONS      ...
-----
          98              1 ...
           9              1 ...
           0              0 ...
           0              1 ...
           0              1 ...
           0              1 ...
           0              1 ...
           0              1 ...
           0              1 ...
          40              1 ...
```

9 record(s) selected.

この照会からの出力 (続き)。

```
... STMT_TEXT
... -----
... select prep_time_worst, num_compilations, substr(stmt_text,
... select * from dbuser.employee
... SET CURRENT LOCALE LC_CTYPE = 'en_US'
... select prep_time_worst, num_compilations, substr(stmt_text, ...
```

```

... select prep_time_worst, num_compilations, substr(stmt_text, ...
... select * from dbuser.employee ...
... insert into dbuser.employee values(1) ...
... select * from dbuser.employee ...
... insert into dbuser.employee values(1) ...

```

この照会からの出力 (続き)。

```

... DBPARTITIONNUM
... -----
...          0
...          0
...          0
...          2
...          1
...          2
...          2
...          1
...          0

```

## SNAP\_GET\_DYN\_SQL 表関数

SNAP\_GET\_DYN\_SQL 表関数は SNAPDYN\_SQL 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

この表関数は、**GET SNAPSHOT FOR DYNAMIC SQL ON database-alias** CLP コマンドと同等の情報を戻します。

戻される可能性のある情報の完全なリストは、903 ページの表 212 を参照してください。

## 構文

```

▶▶ SNAP_GET_DYN_SQL ( (dbname [ , member ] ) )

```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のメンバーの場合は -1、すべてのアクティブなメンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプション

ョンを使用しない場合、つまり *dbname* のみを指定する場合、データベースがアクティブなメンバーからデータが戻されます。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DYN\_SQL 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DYN\_SQL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベース上で実行された動的 SQL のリストを、読み取られた行数の順序で取得します。

```
SELECT PREP_TIME_WORST, NUM_COMPILATIONS, SUBSTR(STMT_TEXT, 1, 60)
      AS STMT_TEXT FROM TABLE(SNAP_GET_DYN_SQL('',-1)) as T
      ORDER BY ROWS_READ
```

以下はこの照会の出力例です。

PREP_TIME_WORST	NUM_COMPILATIONS	...
0	0	...
49	1	...
0	0	...
46	1	...
0	0	...
0	0	...
0	0	...
29	1	...
0	0	...
0	0	...
10	1	...
0	0	...
4	0	...
53	0	...
0	0	...

```

        6          1 ...
       334        0 ...
        0         0 ...
        5         0 ...
       10         0 ...
       599        0 ...
        15        1 ...
         7         0 ...

```

23 record(s) selected.

この照会からの出力 (続き)。

```

... STMT_TEXT
...
-----
... SET :HV00017 :HI00017 = RPAD(VARCHAR(:HV00035 :HI00035 ),
... SELECT COLNAME, TYPENAME FROM SYSCAT.COLUMNS WHERE TABNAME=
... DECLARE RES CURSOR WITH RETURN TO CALLER FOR SELECT R.TEXT F
... SELECT PREP_TIME_WORST, NUM_COMPILATIONS, SUBSTR(STMT_TEXT,
... VALUES (:HV00026 :HI00026 + 1, :HV00024 :HI00024 + 1) IN
... VALUES (:HV00035 :HI00035 + 1, :HV00024 :HI00024 + 1) IN
... VALUES (1) INTO :HV00035 :HI00035
... SELECT TRIGNAME FROM SYSCAT.TRIGGERS WHERE TABNAME='POLICY'
... VALUES (:HV00024 :HI00024 +1, :HV00022 :HI00022 +1) INTO :
... VALUES (1, CARDINALITY(CAST(:HV00040 :HI00040 AS "SYSIBMADM
... CALL SYSPROC.SYSINSTALLOBJECTS('POLICY','V',' ',' '))
... SET :HV00017 :HI00017 = RPAD(VARCHAR(:HV00035 :HI00035 ),
... drop event monitor act
... SELECT TABSCHEMA, TABNAME, TYPE, STATUS, TBSpaceID, PROPERTY
... CALL SAVE_EXEC_INFO (CAST(:HV00040 :HI00040 AS "SYSIBMADM"
... SET CURRENT LOCK TIMEOUT 5
... SELECT TABNAME FROM SYSCAT.PERIODS WHERE PERIODNAME = 'SYSTE
... SELECT ARRAY_AGG(P.EXECUTABLE_ID ORDER BY M.IO_WAIT_TIME DES
... SET CURRENT ISOLATION RESET
... CALL monreport.pkgcache()
... SELECT A.SPECIFICNAME FROM SYSCAT.ROUTINES A WHERE (A.FENCED
... SELECT POLICY FROM SYSTOOLS.POLICY WHERE MED='DB2CommonMED'
... VALUES 0

```

23 record(s) selected.

ワークロードの実行後に、ユーザーは次の照会を表関数とともに使用できます。

```

SELECT STATS_FABRICATE_TIME,SYNC_RUNSTATS_TIME
FROM TABLE (SNAP_GET_DYN_SQL('mytestdb', -1))
AS SNAPDB

```

STATS_FABRICATE_TIME	SYNC_RUNSTATS_TIME
2	12
1	30

この表関数に基づくビューの場合:

```

SELECT STATS_FABRICATE_TIME,SYNC_RUNSTATS_TIME
FROM SYSIBMADM.SNAPDYN_SQL

```

STATS_FABRICATE_TIME	SYNC_RUNSTATS_TIME
5	10
3	20

2 record(s) selected.

## 戻される情報

表 212. SNAPDYN\_SQL 管理ビューおよび SNAP\_GET\_DYN\_SQL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
NUM_EXECUTIONS	BIGINT	num_executions - ステートメント実行回数
NUM_COMPILATIONS	BIGINT	num_compilations - ステートメント・コンパイル数
PREP_TIME_WORST	BIGINT	prep_time_worst - ステートメント最長準備時間
PREP_TIME_BEST	BIGINT	prep_time_best - ステートメント最短準備時間
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 削除された内部行数
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 挿入された内部行数
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 更新された内部行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
STMT_SORTS	BIGINT	stmt_sorts - ステートメント・ソート回数
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
TOTAL_SORT_TIME	BIGINT	total_sort_time - ソート時間合計
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り : モニター・エレメント

表 212. SNAPDYN\_SQL 管理ビューおよび SNAP\_GET\_DYN\_SQL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_EXEC_TIME	BIGINT	total_exec_time - ステートメント実行の経過時間 (秒単位)*
TOTAL_EXEC_TIME_MS	BIGINT	total_exec_time - ステートメント実行の経過時間 (小数部、マイクロ秒単位)*
TOTAL_USR_CPU_TIME	BIGINT	total_usr_cpu_time - ステートメントのユーザー CPU の合計 (秒単位)*
TOTAL_USR_CPU_TIME_MS	BIGINT	total_usr_cpu_time - ステートメントのユーザー CPU の合計 (小数部、マイクロ秒単位)*
TOTAL_SYS_CPU_TIME	BIGINT	total_sys_cpu_time - ステートメントのシステム CPU の合計 (秒単位)*
TOTAL_SYS_CPU_TIME_MS	BIGINT	total_sys_cpu_time - ステートメントのシステム CPU の合計 (小数部、マイクロ秒単位)*
STMT_TEXT	CLOB(2 M)	stmt_text - SQL 動的ステートメント・テキスト
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
STATS_FABRICATE_TIME	BIGINT	動的ステートメントの照会コンパイル中に表または索引のスキャンを実行しないで、システムが必要とされる統計を作成するのに費やす合計時間 (ミリ秒)。
SYNC_RUNSTATS_TIME	BIGINT	動的ステートメントの照会コンパイル中に同期 statistics-collect アクティビティーに費やされる合計時間 (ミリ秒)。
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
<p>* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する <code>_S</code> で終わっている列で報告されている整数秒と、このモニター・エレメントに関する <code>_MS</code> で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません: <math>(\text{monitor-element-name}_S \times 1,000,000 + \text{monitor-element-name}_{MS}) \div 1,000,000</math>. 例えば、<math>(\text{ELAPSED\_EXEC\_TIME}_S \times 1,000,000 + \text{ELAPSED\_EXEC\_TIME}_{MS}) \div 1,000,000</math>.</p>		

## SNAPFCM 管理ビューおよび SNAP\_GET\_FCM 表関数 - fcm 論理データ・グループ・スナップショット情報の検索

SNAPFCM 管理ビューおよび SNAP\_GET\_FCM 表関数は、データベース・マネージャー・スナップショットから、特に fcm 論理データ・グループの高速コミュニケーション・マネージャー情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 905 ページの『SNAPFCM 管理ビュー』
- 906 ページの『SNAP\_GET\_FCM 表関数』

## SNAPFCM 管理ビュー

SNAPDBM、SNAPDBM\_MEMORY\_POOL、SNAPFCM\_PART、および SNAPSWITCHES 管理ビューと共に使用すると、SNAPFCM 管理ビューは、**GET SNAPSHOT FOR DBM** コマンドと同等のデータを提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、907 ページの表 213 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPFCM 管理ビューに対する SELECT 特権
- SNAPFCM 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_FCM 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

すべてのメンバーにおける高速コミュニケーション・マネージャーのメッセージ・バッファについて情報を検索します。

```
SELECT BUFF_FREE, BUFF_FREE_BOTTOM, MEMBER  
FROM SYSIBMADM.SNAPFCM ORDER BY MEMBER
```

以下はこの照会の出力例です。

BUFF_FREE	BUFF_FREE_BOTTOM	MEMBER
5120	5100	0
5120	5100	1
5120	5100	2

## SNAP\_GET\_FCM 表関数

SNAP\_GET\_FCM 表関数は SNAPFCM 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーを対象とした情報を検索することができます。

SNAP\_GET\_DBM、SNAP\_GET\_DBM\_MEMORY\_POOL、SNAP\_GET\_FCM\_PART、および SNAP\_GET\_SWITCHES 表関数と共に使用すると、SNAP\_GET\_FCM 表関数は、**GET SNAPSHOT FOR DBM** コマンドと同等のデータを提供します。

戻される可能性のある情報の完全なリストは、907 ページの表 213 を参照してください。

## 構文

```

▶▶ SNAP_GET_FCM ( [ member ] )

```

スキーマは SYSPROC です。

## 表関数パラメーター

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のメンバーの場合は -1、すべてのアクティブなメンバーの集合の場合は -2 を指定します。 *dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_FCM 表関数は、現在接続中のデータベースおよびデータベース・メンバーのスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_FCM 表関数に対する EXECUTE 特権
- DATAACCESS 権限



さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

データベース・メンバー 1 における高速コミュニケーション・マネージャーのメッセージ・バッファーについての情報を検索します。

```
SELECT BUFF_FREE, BUFF_FREE_BOTTOM, MEMBER
FROM TABLE(SYSPROC.SNAP_GET_FCM( 1 )) AS T
```

以下はこの照会の出力例です。

```

BUFF_FREE          BUFF_FREE_BOTTOM    MEMBER
-----
                5120                5100         1

```

### 戻される情報

表 213. SNAPFCM 管理ビューおよび SNAP\_GET\_FCM 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
BUFF_FREE	BIGINT	buff_free - 現在空いている FCM バッファー
BUFF_FREE_BOTTOM	BIGINT	buff_free_bottom - 空き FCM バッファーの最小数
CH_FREE	BIGINT	ch_free - 現在空いているチャンネル
CH_FREE_BOTTOM	BIGINT	ch_free_bottom - 空いているチャンネルの最小
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPFCM\_PART 管理ビューおよび SNAP\_GET\_FCM\_PART 表関数 - fcm\_node 論理データ・グループ・スナップショット情報の検索

SNAPFCM\_PART 管理ビューおよび SNAP\_GET\_FCM\_PART 表関数は、データベース・マネージャー・スナップショットから、特に fcm\_node 論理データ・グループの高速コミュニケーション・マネージャー情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPFCM\_PART 管理ビュー』
- 909 ページの『SNAP\_GET\_FCM\_PART 表関数』

### SNAPFCM\_PART 管理ビュー

SNAPDBM、SNAPDBM\_MEMORY\_POOL、SNAPFCM、および SNAPSWITCHES 管理ビューと共に使用すると、SNAPFCM\_PART 管理ビューは、**GET SNAPSHOT FOR DBM** コマンドと同等のデータを提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、910 ページの表 214 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPFCM\_PART 管理ビューに対する SELECT 特権
- SNAPFCM\_PART 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_FCM\_PART 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

高速コミュニケーション・マネージャーのバッファ送受信情報を検索します。

```
SELECT CONNECTION_STATUS, TOTAL_BUFFERS_SENT, TOTAL_BUFFERS_RECEIVED  
FROM SYSIBADM.SNAPFCM_PART WHERE MEMBER = 0
```

以下はこの照会の出力例です。

CONNECTION_STATUS	TOTAL_BUFFERS_SENT	TOTAL_BUFFERS_RCVD
INACTIVE	2	1

1 record(s) selected.

## SNAP\_GET\_FCM\_PART 表関数

SNAP\_GET\_FCM\_PART 表関数は SNAPFCM\_PART 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーを対象とした情報を検索することができます。

SNAP\_GET\_DBM、SNAP\_GET\_DBM\_MEMORY\_POOL、SNAP\_GET\_FCM、および SNAP\_GET\_SWITCHES 表関数と共に使用すると、SNAP\_GET\_FCM\_PART 表関数は、GET SNAPSHOT FOR DBM コマンドと同等のデータを提供します。

戻される可能性のある情報の完全なリストは、910 ページの表 214 を参照してください。

### 構文

```
▶▶ SNAP_GET_FCM_PART ( (member) )
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のメンバーの場合は -1 を指定します。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプが含まれるファイルが存在

しない場合、SNAP\_GET\_FCM\_PART 表関数は、現在接続されているデータベースとメンバーのスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_FCM\_PART 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべてのメンバーにおける高速コミュニケーション・マネージャーのバッファ送受信情報を検索します。

```
SELECT FCM_MEMBER, TOTAL_BUFFERS_SENT, TOTAL_BUFFERS_RCVD,  
       MEMBER FROM TABLE(SNAP_GET_FCM_PART()) AS T  
ORDER BY MEMBER
```

以下はこの照会の出力例です。

FCM_MEMBER	TOTAL_BUFFERS_SENT	TOTAL_BUFFERS_RCVD	MEMBER
0	305	305	0
1	5647	1664	0
2	5661	1688	0
0	19	19	1
1	305	301	1
2	1688	5661	1
0	1664	5647	2
1	10	10	2
2	301	305	2

## 戻される情報

表 214. SNAP\_FCM\_PART 管理ビューおよび SNAP\_GET\_FCM\_PART 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。

表 214. SNAPFCM\_PART 管理ビューおよび SNAP\_GET\_FCM\_PART 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
CONNECTION_STATUS	VARCHAR(10)	connection_status - 接続状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• INACTIVE</li> <li>• ACTIVE</li> <li>• CONGESTED</li> </ul>
TOTAL_BUFFERS_SENT	BIGINT	total_buffers_sent - 送信された FCM バッファの合計
TOTAL_BUFFERS_RCVD	BIGINT	total_buffers_rcvd - 受信された FCM バッファの合計
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
FCM_DBPARTITIONNUM	SMALLINT	データの送信先または受信元のデータベース・パーティション番号 (TOTAL_BUFFERS_SENT および TOTAL_BUFFERS_RCVD 列ごとに)。
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
FCM_MEMBER	SMALLINT	(TOTAL_BUFFERS_SENT 列および TOTAL_BUFFERS_RCVD 列のとおり) データの送信または受信が行われたメンバー。

## SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数 - ステートメント・スナップショット情報の検索

SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数は、アプリケーション・スナップショットから SQL または XQuery ステートメントについての情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPSTMT 管理ビュー』
- 913 ページの『SNAP\_GET\_STMT 表関数』

## SNAPSTMT 管理ビュー

この管理ビューでは、現在接続されているデータベースのステートメント・スナップショット情報を検索できます。

SNAPAGENT、SNAPAGENT\_MEMORY\_POOL、SNAPAPPL、SNAPAPPL\_INFO、および SNAPSUBSECTION 管理ビューと共に使用すると、SNAPSTMT 管理ビューは、**GET SNAPSHOT FOR APPLICATIONS on database-alias** CLP コマンドと同等の情報提供しますが、すべてのデータベース・メンバーからデータを検索します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、915 ページの表 215 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPSTMT 管理ビューに対する SELECT 特権
- SNAPSTMT 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_STMT 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されている単一メンバー・データベース上で実行されたステートメントのために読み取り、書き込み、および操作の実行が行われた行を検索します。

```
SELECT SUBSTR(STMT_TEXT,1,30) AS STMT_TEXT, ROWS_READ, ROWS_WRITTEN,  
       STMT_OPERATION FROM SYSIBMADM.SNAPSTMT
```

以下はこの照会の出力例です。

STMT_TEXT	ROWS_READ	ROWS_WRITTEN	STMT_OPERATION
-		0	0 FETCH
-		0	0 STATIC_COMMIT

2 record(s) selected.

## SNAP\_GET\_STMT 表関数

SNAP\_GET\_STMT 表関数は SNAPSTMT 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_AGENT、SNAP\_GET\_AGENT\_MEMORY\_POOL、SNAP\_GET\_APPL、SNAP\_GET\_APPL\_INFO、および SNAP\_GET\_SUBSECTION 表関数と共に使用すると、SNAP\_GET\_STMT 表関数は、**GET SNAPSHOT FOR ALL APPLICATIONS CLP** コマンドと同等の情報を提供しますが、すべてのデータベース・パーティションからデータを検索します。

戻される可能性のある情報の完全なリストは、915 ページの表 215 を参照してください。

## 構文

```

▶▶ SNAP_GET_STMT ( (dbname [ , member ] ) )

```

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり dbname のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されず、アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャーにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_STMT 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_STMT 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースの現行データベース・メンバー上で実行されたステートメントのために読み取り、書き込み、および操作の実行が行われた行を検索します。

```
SELECT SUBSTR(STMT_TEXT,1,30) AS STMT_TEXT, ROWS_READ,
       ROWS_WRITTEN, STMT_OPERATION FROM TABLE(SNAP_GET_STMT('','-1)) AS T
```

以下はこの照会の出力例です。

```
STMT_TEXT                                ROWS_READ    ...
-----
update t set a=3                          0 ...
SELECT SUBSTR(STMT_TEXT,1,30)             0 ...
-                                           0 ...
-                                           0 ...
update t set a=2                          9 ...
5 record(s) selected.                      ...
```

この照会からの出力 (続き)。

```
... ROWS_WRITTEN    STMT_OPERATION
... -----
...                0 EXECUTE_IMMEDIATE
...                0 FETCH
...                0 NONE
...                0 NONE
...                1 EXECUTE_IMMEDIATE
...                ...
```



## 戻される情報

表 215. SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
NUM_AGENTS	BIGINT	num_agents - ステートメントで作動しているエージェントの数
AGENTS_TOP	BIGINT	agents_top - 作成されたエージェントの数
STMT_TYPE	VARCHAR(20)	stmt_type - ステートメント・タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"><li>• DYNAMIC</li><li>• NON_STMT</li><li>• STATIC</li><li>• STMT_TYPE_UNKNOWN</li></ul>

表 215. SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
STMT_OPERATION	VARCHAR(20)	stmt_operation/operation - ステートメント操作。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• CALL</li> <li>• CLOSE</li> <li>• COMPILE</li> <li>• DESCRIBE</li> <li>• EXECUTE</li> <li>• EXECUTE_IMMEDIATE</li> <li>• FETCH</li> <li>• FREE_LOCATOR</li> <li>• GETAA</li> <li>• GETNEXTCHUNK</li> <li>• GETTA</li> <li>• NONE</li> <li>• OPEN</li> <li>• PREP_COMMIT</li> <li>• PREP_EXEC</li> <li>• PREP_OPEN</li> <li>• PREPARE</li> <li>• REBIND</li> <li>• REDIST</li> <li>• REORG</li> <li>• RUNSTATS</li> <li>• SELECT</li> <li>• SET</li> <li>• STATIC_COMMIT</li> <li>• STATIC_ROLLBACK</li> </ul>
SECTION_NUMBER	BIGINT	section_number - セクション番号
QUERY_COST_ESTIMATE	BIGINT	query_cost_estimate - 照会コストの見積もり
QUERY_CARD_ESTIMATE	BIGINT	query_card_estimate - 照会行数の見積もり
DEGREE_PARALLELISM	BIGINT	degree_parallelism - 並列処理の度合い
STMT_SORTS	BIGINT	stmt_sorts - ステートメント・ソート回数
TOTAL_SORT_TIME	BIGINT	total_sort_time - ソート時間合計

表 215. SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 削除された内部行数
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 更新された内部行数
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 挿入された内部行数
FETCH_COUNT	BIGINT	fetch_count - 成功した取り出しの数
STMT_START	TIMESTAMP	stmt_start - ステートメント操作開始タイム・スタンプ
STMT_STOP	TIMESTAMP	stmt_stop - ステートメント操作停止タイム・スタンプ
STMT_USR_CPU_TIME_S	BIGINT	stmt_usr_cpu_time - ステートメントに使用されたユーザー CPU 時間 (秒単位)*
STMT_USR_CPU_TIME_MS	BIGINT	stmt_usr_cpu_time - ステートメントに使用されたユーザー CPU 時間 (小数部、マイクロ秒単位)*
STMT_SYS_CPU_TIME_S	BIGINT	stmt_sys_cpu_time - ステートメントが使用したシステム CPU 時間 (秒単位)*
STMT_SYS_CPU_TIME_MS	BIGINT	stmt_sys_cpu_time - ステートメントが使用したシステム CPU 時間 (小数部、マイクロ秒単位)*
STMT_ELAPSED_TIME_S	BIGINT	stmt_elapsed_time - 最新のステートメント経過時間 (秒単位)*
STMT_ELAPSED_TIME_MS	BIGINT	stmt_elapsed_time - 最新のステートメント経過時間 (小数部、マイクロ秒単位)*
BLOCKING_CURSOR	SMALLINT	blocking_cursor - ブロック・カーソル
STMT_NODE_NUMBER	SMALLINT	stmt_node_number - ステートメント・ノード
CURSOR_NAME	VARCHAR(128)	cursor_name - カーソル名
CREATOR	VARCHAR(128)	creator - アプリケーション作成者
PACKAGE_NAME	VARCHAR(128)	package_name - パッケージ名
STMT_TEXT	CLOB(16 M)	stmt_text - SQL 動的ステートメント・テキスト
CONSISTENCY_TOKEN	VARCHAR(128)	consistency_token - パッケージ整合性トークン

表 215. SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
PACKAGE_VERSION_ID	VARCHAR(128)	package_version_id - パッケージ・バージョン
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り : モニター・エレメント
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り : モニター・エレメント
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り : モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

表 215. SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
<p>* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する _S で終わっている列で報告されている整数秒と、このモニター・エレメントに関する _MS で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません: <math>(\text{monitor-element-name}_S \times 1,000,000 + \text{monitor-element-name}_{MS}) \div 1,000,000</math>. 例えば、<math>(\text{ELAPSED\_EXEC\_TIME}_S \times 1,000,000 + \text{ELAPSED\_EXEC\_TIME}_{MS}) \div 1,000,000</math>.</p>		

## SNAPSUBSECTION 管理ビューおよび SNAP\_GET\_SUBSECTION 表関数 - subsection 論理モニター・グループ・スナップショット情報の検索

SNAPSUBSECTION 管理ビューおよび SNAP\_GET\_SUBSECTION 表関数は、アプリケーション・サブセクション情報として、subsection 論理モニター・グループの情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPSUBSECTION 管理ビュー』
- 920 ページの 『SNAP\_GET\_SUBSECTION 表関数』

### SNAPSUBSECTION 管理ビュー

この管理ビューでは、現在接続されているデータベースの subsection 論理モニター・グループ・スナップショット情報を検索できます。

SNAPAGENT、SNAPAGENT\_MEMORY\_POOL、SNAPAPPL、SNAPAPPL\_INFO、および SNAPSTMT 管理ビューと共に使用すると、SNAPSUBSECTION 管理ビューは、**GET SNAPSHOT FOR APPLICATIONS on database-alias** CLP コマンドと同等の情報を提供しますが、すべてのデータベース・メンバーからデータを検索します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、922 ページの表 216 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPSUBSECTION 管理ビューに対する SELECT 特権
- SNAPSUBSECTION 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_SUBSECTION 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

すべてのデータベース・メンバーで実行するサブセクションの状況を取得します。

```
SELECT DB_NAME, STMT_TEXT, SS_STATUS, DBPARTITIONNUM
FROM SYSIBMADM.SNAPSUBSECTION
ORDER BY DB_NAME, SS_STATUS, DBPARTITIONNUM
```

以下はこの照会の出力例です。

DB_NAME	STMT_TEXT	SS_STATUS	DBPARTITIONNUM
SAMPLE	select * from EMPLOYEE	EXEC	0
SAMPLE	select * from EMPLOYEE	EXEC	1

### SNAP\_GET\_SUBSECTION 表関数

SNAP\_GET\_SUBSECTION 表関数は SNAPSUBSECTION 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

戻される可能性のある情報の完全なリストは、922 ページの表 216 を参照してください。

SNAP\_GET\_AGENT、SNAP\_GET\_AGENT\_MEMORY\_POOL、SNAP\_GET\_APPL、SNAP\_GET\_APPL\_INFO、および SNAP\_GET\_STMT 表関数と共に使用すると、SNAP\_GET\_SUBSECTION 表関数は、**GET SNAPSHOT FOR ALL APPLICATIONS CLP** コマンドと同等の情報を提供しますが、すべてのデータベース・メンバーからデータを検索します。

### 構文

```
▶▶—SNAP_GET_SUBSECTION—(—dbname— [ , member— ] )—▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されません。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_SUBSECTION 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_SUBSECTION 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべてのデータベース・メンバーで実行するサブセクションの状況を取得します。

```
SELECT DB_NAME, STMT_TEXT, SS_STATUS, DBPARTITIONNUM
FROM TABLE(SYSPROC.SNAP_GET_SUBSECTION( ' ', 0 )) as T
ORDER BY DB_NAME, SS_STATUS, DBPARTITIONNUM
```

以下はこの照会の出力例です。

DB_NAME	STMT_TEXT	SS_STATUS	DBPARTITIONNUM
SAMPLE	select * from EMPLOYEE	EXEC	0
SAMPLE	select * from EMPLOYEE	EXEC	1

## 戻される情報

表 216. SNAPSUBSECTION 管理ビューおよび SNAP\_GET\_SUBSECTION 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
STMT_TEXT	CLOB(16 M)	stmt_text - SQL 動的ステートメント・テキスト
SS_EXEC_TIME	BIGINT	ss_exec_time - サブセクション実行経過時間
TQ_TOT_SEND_SPILLS	BIGINT	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
TQ_CUR_SEND_SPILLS	BIGINT	tq_cur_send_spills - オーバーフローした表キュー・バッファの現在数
TQ_MAX_SEND_SPILLS	BIGINT	tq_max_send_spills - 表キュー・バッファ・オーバーフローの最大数
TQ_ROWS_READ	BIGINT	tq_rows_read - 表キューから読み取られた行数
TQ_ROWS_WRITTEN	BIGINT	tq_rows_written - 表キューに書き込まれた行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
SS_USR_CPU_TIME_S	BIGINT	ss_usr_cpu_time - サブセクションに使用されたユーザー CPU 時間 (秒単位)*
SS_USR_CPU_TIME_MS	BIGINT	ss_usr_cpu_time - サブセクションに使用されたユーザー CPU 時間 (小数部、マイクロ秒単位)*
SS_SYS_CPU_TIME_S	BIGINT	ss_sys_cpu_time - サブセクションに使用されたシステム CPU 時間 (秒単位)*
SS_SYS_CPU_TIME_MS	BIGINT	ss_sys_cpu_time - サブセクションに使用されたシステム CPU 時間 (小数部、マイクロ秒単位)*



表 216. SNAPSUBSECTION 管理ビューおよび SNAP\_GET\_SUBSECTION 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SS_NUMBER	INTEGER	ss_number - サブセクション番号
SS_STATUS	VARCHAR(20)	ss_status - サブセクションの状況。 このインターフェースは、sqlmon.h での定義に基づくテキスト ID を 戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• EXEC</li> <li>• TQ_WAIT_TO_RCV</li> <li>• TQ_WAIT_TO_SEND</li> <li>• COMPLETED</li> </ul>
SS_NODE_NUMBER	SMALLINT	ss_node_number - サブセクション・ノード番号
TQ_NODE_WAITED_FOR	SMALLINT	tq_node_waited_for - 表キュー上のノード待機
TQ_WAIT_FOR_ANY	INTEGER	tq_wait_for_any - 表キュー上のノード送信待機
TQ_ID_WAITING_ON	INTEGER	tq_id_waiting_on - ノード上の表キュー待機
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
<p>* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する <code>_S</code> で終わっている列で報告されている整数秒と、このモニター・エレメントに関する <code>_MS</code> で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません: <math>(\text{monitor-element-name}_S \times 1,000,000 + \text{monitor-element-name}_MS) \div 1,000,000</math>。例えば、<math>(\text{ELAPSED\_EXEC\_TIME}_S \times 1,000,000 + \text{ELAPSED\_EXEC\_TIME}_MS) \div 1,000,000</math>。</p>		

## SNAPSWITCHES 管理ビューおよび SNAP\_GET\_SWITCHES 表関数 - データベース・スナップショットのスイッチ状態情報の検索

SNAPSWITCHES 管理ビューおよび SNAP\_GET\_SWITCHES 表関数は、データベース・スナップショットのスイッチ状態に関する情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 924 ページの『SNAPSWITCHES 管理ビュー』
- 925 ページの『SNAP\_GET\_SWITCHES 表関数』

## SNAPSWITCHES 管理ビュー

このビューは、`GET DBM MONITOR SWITCHES CLP` コマンドと同等のデータを提供します。

スキーマは `SYSIBMADM` です。

戻される可能性のある情報の完全なリストは、926 ページの表 217 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPSWITCHES 管理ビューに対する `SELECT` 特権
- SNAPSWITCHES 管理ビューに対する `CONTROL` 特権
- `DATAACCESS` 権限
- `DBADM` 権限
- `SQLADM` 権限

この表関数を使用するには、以下のいずれかが必要です。

- `SNAP_GET_SWITCHES` 表関数に対する `EXECUTE` 特権
- `DATAACCESS` 権限
- `DBADM` 権限
- `SQLADM` 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- `SYSMON`
- `SYSCTRL`
- `SYSMAINT`
- `SYSADM`

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、`SELECT` 特権が `PUBLIC` に付与されます。

### 例

すべてのデータベース・メンバーにおける `DBM` モニターのスイッチ状態情報を検索します。

```
SELECT UOW_SW_STATE, STATEMENT_SW_STATE, TABLE_SW_STATE, BUFFPOOL_SW_STATE,  
       LOCK_SW_STATE, SORT_SW_STATE, TIMESTAMP_SW_STATE,  
       DBPARTITIONNUM FROM SYSIBMADM.SNAPSWITCHES
```

以下はこの照会の出力例です。

```
UOW_SW_STATE STATEMENT_SW_STATE TABLE_SW_STATE BUFFPOOL_SW_STATE ...  
-----  
0 0 0 0 ...
```

```

          0          0          0          0 ...
          0          0          0          0 ...
                                     ...

```

3 record selected.

この照会からの出力 (続き)。

```

... LOCK_SW_STATE SORT_SW_STATE TIMESTAMP_SW_STATE DBPARTITIONNUM
... -----
...          1          0          1          0
...          1          0          1          1
...          1          0          1          2

```

## SNAP\_GET\_SWITCHES 表関数

SNAP\_GET\_SWITCHES 表関数は SNAPSWITCHES 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーを対象とした情報を検索することができます。

この表関数は、**GET DBM MONITOR SWITCHES CLP** コマンドと同等のデータを提供します。

戻される可能性のある情報の完全なリストは、926 ページの表 217 を参照してください。

## 構文

```

▶▶ SNAP_GET_SWITCHES ( [member] )

```

スキーマは SYSPROC です。

## 表関数パラメーター

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_SWITCHES 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_SWITCHES 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現行データベース・メンバーにおける DBM モニターのスイッチ状態情報を検索します。

```
SELECT UOW_SW_STATE, STATEMENT_SW_STATE, TABLE_SW_STATE,  
       BUFFPOOL_SW_STATE, LOCK_SW_STATE, SORT_SW_STATE, TIMESTAMP_SW_STATE  
FROM TABLE(SNAP_GET_SWITCHES(-1)) AS T
```

以下はこの照会の出力例です。

```
UOW_SW_STATE STATEMENT_SW_STATE TABLE_SW_STATE...  
-----  
1 1 1...  
...  
1 record(s) selected. ...
```

この照会からの出力 (続き)。

```
... BUFFPOOL_SW_STATE LOCK_SW_STATE SORT_SW_STATE TIMESTAMP_SW_STATE  
... -----  
... 1 1 0 1
```

## 戻される情報

表 217. SNAPSWITCHES 管理ビューおよび SNAP\_GET\_SWITCHES 表関数によって戻される情報

列名	データ・タイプ	説明
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
UOW_SW_STATE	SMALLINT	作業単位モニター記録スイッチの状態 (0 または 1)。
UOW_SW_TIME	TIMESTAMP	作業単位モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
STATEMENT_SW_STATE	SMALLINT	SQL ステートメント・モニター記録スイッチの状態 (0 または 1)。

表 217. SNAPSWITCHES 管理ビューおよび SNAP\_GET\_SWITCHES 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
STATEMENT_SW_TIME	TIMESTAMP	SQL ステートメント・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
TABLE_SW_STATE	SMALLINT	表アクティビティ・モニター記録スイッチの状態 (0 または 1)。
TABLE_SW_TIME	TIMESTAMP	表アクティビティ・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
BUFFPOOL_SW_STATE	SMALLINT	バッファ・プール・アクティビティ・モニター記録スイッチの状態 (0 または 1)。
BUFFPOOL_SW_TIME	TIMESTAMP	バッファ・プール・アクティビティ・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
LOCK_SW_STATE	SMALLINT	ロック・モニター記録スイッチの状態 (0 または 1)。
LOCK_SW_TIME	TIMESTAMP	ロック・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
SORT_SW_STATE	SMALLINT	ソート・モニター記録スイッチの状態 (0 または 1)。
SORT_SW_TIME	TIMESTAMP	ソート・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
TIMESTAMP_SW_STATE	SMALLINT	タイム・スタンプのモニター記録スイッチの状態 (0 または 1)。
TIMESTAMP_SW_TIME	TIMESTAMP	タイム・スタンプのモニター記録スイッチがオンの場合、このスイッチがオンになった日付と時刻。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

### SNAPTAB\_REORG 管理ビューおよび SNAP\_GET\_TAB\_REORG 表関数 - 表再編成スナップショット情報の検索

SNAPTAB\_REORG 管理ビューおよび SNAP\_GET\_TAB\_REORG 表関数は、表再編成情報を戻します。

再編成された表がない場合は、0 行が戻されます。データ・パーティション表が再編成される時、各データ・パーティションに対して 1 つのレコードが戻されま

す。データ・パーティション表の特定の 1 つのデータ・パーティションだけが再編成される場合、そのパーティションのレコードだけが戻されます。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPTAB\_REORG 管理ビュー』
- 929 ページの『SNAP\_GET\_TAB\_REORG 表関数』

## SNAPTAB\_REORG 管理ビュー

この管理ビューでは、現在接続されているデータベースの表再編成スナップショット情報を検索できます。

SNAPTAB 管理ビューと共に使用すると、SNAPTAB\_REORG 管理ビューは、**GET SNAPSHOT FOR TABLES ON database-alias CLP** コマンドと同等のデータを提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、931 ページの表 218 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPTAB\_REORG 管理ビューに対する SELECT 特権
- SNAPTAB\_REORG 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_TAB\_REORG 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベース上のすべてのデータベース・メンバーでの再編成操作の詳細を選択します。

```
SELECT SUBSTR(TABNAME, 1, 15) AS TAB_NAME, SUBSTR(TABSCHEMA, 1, 15)
      AS TAB_SCHEMA, REORG_PHASE, SUBSTR(REORG_TYPE, 1, 20) AS REORG_TYPE,
      REORG_STATUS, REORG_COMPLETION, DBPARTITIONNUM
FROM SYSIBMADM.SNAPTAB_REORG ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

TAB_NAME	TAB_SCHEMA	REORG_PHASE	...
EMPLOYEE	DBUSER	REPLACE	...
EMPLOYEE	DBUSER	REPLACE	...
EMPLOYEE	DBUSER	REPLACE	...

3 record(s) selected.

この照会からの出力 (続き)。

...	REORG_TYPE	REORG_STATUS	REORG_COMPLETION	DBPARTITIONNUM
...	RECLAIM+OFFLINE+ALLO	COMPLETED	SUCCESS	0
...	RECLAIM+OFFLINE+ALLO	COMPLETED	SUCCESS	1
...	RECLAIM+OFFLINE+ALLO	COMPLETED	SUCCESS	2

マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング (ITC) 表からエクステントを再利用するための再編成操作に関するすべての情報を、SNAPTAB\_REORG 管理ビューから選択します。

```
db2 -v "select * from sysibmadm.snaptab_reorg"
```

TABNAME	REORG_PHASE	REORG_MAX_PHASE	REORG_TYPE
T1	RELEASE	3	RECLAIM_EXTENTS+ALLOW_WRITE

REORG_STATUS	REORG_COMPLETION	REORG_START	REORG_END
COMPLETED	SUCCESS	2008-09-24-14.35.30.734741	2008-09-24-14.35.31.460674

## SNAP\_GET\_TAB\_REORG 表関数

SNAP\_GET\_TAB\_REORG 表関数は SNAPTAB\_REORG 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_TAB 表関数と共に使用すると、SNAP\_GET\_TAB\_REORG 表関数は、**GET SNAPSHOT FOR TABLES ON database-alias** CLP コマンドと同等のデータを提供します。

戻される可能性のある情報の完全なリストは、931 ページの表 218 を参照してください。

## 構文

```
▶▶ SNAP_GET_TAB_REORG ( ( dbname [ , member ] ) )
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されません。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_TAB\_REORG 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_TAB\_REORG 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベース上のデータベース・メンバー 1 での再編成操作の詳細を選択します。



```
SELECT SUBSTR(TABNAME, 1, 15) AS TAB_NAME, SUBSTR(TABSCHEMA, 1, 15)
AS TAB_SCHEMA, REORG_PHASE, SUBSTR(REORG_TYPE, 1, 20) AS REORG_TYPE,
REORG_STATUS, REORG_COMPLETION, DBPARTITIONNUM
FROM TABLE( SNAP_GET_TAB_REORG(' ', 1)) AS T
```

以下はこの照会の出力例です。

```
TAB_NAME      TAB_SCHEMA    REORG_PHASE    REORG_TYPE      ...
-----
EMPLOYEE      DBUSER        REPLACE        RECLAIM+OFFLINE+ALLO ...
1 record(s) selected.      ...
```

この照会からの出力 (続き)。

```
... REORG_STATUS REORG_COMPLETION DBPARTITIONNUM
... -----
... COMPLETED   SUCCESS                               1
...
```

SNAP\_GET\_TAB\_REORG 表関数を使用して、マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング (ITC) 表からエクステントを再利用するための再編成操作に関するすべての情報を選択します。

```
db2 -v "select * from table(snap_get_tab_reorg(' '))"
```

```
TABNAME  REORG_PHASE    REORG_MAX_PHASE  REORG_TYPE
-----
T1       RELEASE        3                 RECLAIM_EXTENTS+ALLOW_WRITE

REORG_STATUS REORG_COMPLETION REORG_START                REORG_END
-----
COMPLETED   SUCCESS          2008-09-24-14.35.30.734741 2008-09-24-14.35.31.460674
```

## 戻される情報

表 218. SNAPTAB\_REORG 管理ビューおよび SNAP\_GET\_TAB\_REORG 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TABNAME	VARCHAR (128)	table_name - 表名
TABSCHEMA	VARCHAR (128)	table_schema - 表スキーマ名
PAGE_REORGS	BIGINT	page_reorgs - ページ再編成
REORG_PHASE	VARCHAR (16)	reorg_phase - 表再編成フェーズ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• BUILD</li> <li>• DICT_SAMPLE</li> <li>• INDEX_RECREATE</li> <li>• REPLACE</li> <li>• SORT</li> <li>• SCAN</li> <li>• DRAIN</li> <li>• RELEASE</li> </ul> または SORT+DICT_SAMPLE.
REORG_MAX_PHASE	INTEGER	reorg_max_phase - 表再編成の最大フェーズ数

表 218. SNAPTAB\_REORG 管理ビューおよび SNAP\_GET\_TAB\_REORG 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
REORG_CURRENT_COUNTER	BIGINT	reorg_current_counter - 表再編成の進行状況
REORG_MAX_COUNTER	BIGINT	reorg_max_counter - 表再編成の合計量

表 218. SNAPTAB\_REORG 管理ビューおよび SNAP\_GET\_TAB\_REORG 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
REORG_TYPE	VARCHAR (128)	<p>reorg_type - 表再編成の属性。このインターフェースは、以下の ID の組み合わせを '+' 記号で区切ったものを使用してテキスト ID を戻します。</p> <p>以下のいずれかが使用されます。</p> <ul style="list-style-type: none"> <li>• RECLAIM</li> <li>• RECLUSTER</li> <li>• RECLAIM_EXTS</li> </ul> <p>さらに以下のいずれかが使用されます。</p> <ul style="list-style-type: none"> <li>• +OFFLINE</li> <li>• +ONLINE</li> </ul> <p>アクセス・モードが指定されている場合、以下のいずれかが使用されます。</p> <ul style="list-style-type: none"> <li>• +ALLOW_NONE</li> <li>• +ALLOW_READ</li> <li>• +ALLOW_WRITE</li> </ul> <p>オフラインで RECLUSTER オプションが指定されている場合、以下のいずれかが使用されます。</p> <ul style="list-style-type: none"> <li>• +INDEXSCAN</li> <li>• +TABLESCAN</li> </ul> <p>オフラインの場合、以下のいずれかが使用されます。</p> <ul style="list-style-type: none"> <li>• +LONGLOB</li> <li>• +DATAONLY</li> </ul> <p>オフラインで、オプションが指定されている場合、以下の任意のものが使用されます。</p> <ul style="list-style-type: none"> <li>• +CHOOSE_TEMP</li> <li>• +KEEPDICTIONARY</li> <li>• +RESETDICTIONARY</li> </ul> <p>オンラインで、オプションが指定されている場合、以下が使用されます。</p> <ul style="list-style-type: none"> <li>• +NOTRUNCATE</li> </ul> <p>例 1: REORG TABLE TEST.EMPLOYEE が実行された場合、以下のように表示されます。</p> <pre>RECLAIM+OFFLINE+ALLOW_READ+DATAONLY +KEEPDICTIONARY</pre> <p>例 2: REORG TABLE TEST.EMPLOYEE INDEX EMPIDX INDEXSCAN が実行された場合、以下のように表示されます。</p> <pre>RECLUSTER+OFFLINE+ALLOW_READ+INDEXSCAN +DATAONLY+KEEPDICTIONARY</pre>

表 218. SNAPTAB\_REORG 管理ビューおよび SNAP\_GET\_TAB\_REORG 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
REORG_STATUS	VARCHAR (10)	reorg_status - 表再編成の状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• COMPLETED</li> <li>• PAUSED</li> <li>• STARTED</li> <li>• STOPPED</li> <li>• TRUNCATE</li> </ul>
REORG_COMPLETION	VARCHAR (10)	reorg_completion - 表再編成完了フラグ。このインターフェースは、sqlmon.h での定義を基にしてテキスト ID を戻します。以下のいずれかとなります。 <ul style="list-style-type: none"> <li>• FAIL</li> <li>• SUCCESS</li> </ul>
REORG_START	TIMESTAMP	reorg_start - 表再編成開始時刻
REORG_END	TIMESTAMP	reorg_end - 表再編成終了時刻
REORG_PHASE_START	TIMESTAMP	reorg_phase_start - 表再編成フェーズ開始時刻
REORG_INDEX_ID	BIGINT	reorg_index_id - 表の再編成に使用される索引
REORG_TBSPC_ID	BIGINT	reorg_tbspc_id - 表が再編成される表スペース
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID。非パーティション表では、このエレメントは NULL になります。
REORG_ROWSCOMPRESSED	BIGINT	reorg_rows_compressed - 圧縮行数
REORG_ROWSREJECTED	BIGINT	reorg_rows_rejected_for_compression - 圧縮がリジェクトされる行
REORG_LONG_TBSPC_ID	BIGINT	reorg_long_tbspc_id - 長いオブジェクトが再編成される表スペース
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

### SNAPTbsp 管理ビューおよび SNAP\_GET\_TBSP 表関数 - table space 論理データ・グループのスナップショット情報の検索

SNAPTbsp 管理ビューおよび SNAP\_GET\_TBSP 表関数は、table space 論理データ・グループからのスナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 935 ページの『SNAPTbsp 管理ビュー』
- 936 ページの『SNAP\_GET\_TBSP 表関数』

## SNAPTbsp 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関する table space 論理データ・グループのスナップショット情報を検索できます。

SNAPTbsp\_PART、SNAPTbsp\_QUIESCER、SNAPTbsp\_RANGE、SNAPCONTAINER 管理ビューと併せて使用することにより、SNAPTbsp 管理ビューは **GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を戻します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、937 ページの表 219 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPTbsp 管理ビューに対する SELECT 特権
- SNAPTbsp 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_TBSP 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベースの、カタログ・データベース・メンバーの表スペースのリストを検索します。

```
SELECT SUBSTR(TBSP_NAME,1,30) AS TBSP_NAME, TBSP_ID, TBSP_TYPE,  
       TBSP_CONTENT_TYPE FROM SYSIBMADM.SNAPTbsp WHERE DBPARTITIONNUM = 1
```

以下はこの照会の出力例です。

TBSP_NAME	TBSP_ID	TBSP_TYPE	TBSP_CONTENT_TYPE
TEMPSPACE1	1	SMS	SYSTEMP
USERSPACE1	2	DMS	LONG

2 record(s) selected.

## SNAP\_GET\_TBSP 表関数

SNAP\_GET\_TBSP 表関数は SNAPTBSP 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_TBSP\_PART、SNAP\_GET\_TBSP QUIESCER、SNAP\_GET\_TBSP\_RANGE、SNAP\_GET\_CONTAINER 表関数と併せて使用することにより、SNAP\_GET\_TBSP 表関数は **GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を戻します。

戻される可能性のある情報の完全なリストは、937 ページの表 219 を参照してください。

## 構文

```
▶▶ SNAP_GET_TBSP ( (dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

### member

有効なメンバー番号を指定する、タイプ INTEGER の入力引数。現在のメンバーの場合は -1 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

dbname と member の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_TBSP 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_TBSP 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースの、すべてのデータベース・メンバーの表スペースのリストを検索します。

```
SELECT SUBSTR(TBSP_NAME,1,10) AS TBSP_NAME, TBSP_ID, TBSP_TYPE,  
       TBSP_CONTENT_TYPE, DBPARTITIONNUM FROM TABLE(SNAP_GET_TBSP('')) AS T
```

以下はこの照会の出力例です。

TBSP_NAME	TBSP_ID	TBSP_TYPE	TBSP_CONTENT_TYPE	DBPARTITIONNUM
TEMPSPACE1	1	SMS	SYSTEMP	1
USERSPACE1	2	DMS	LONG	1
SYSCATSPAC	0	DMS	ANY	0
TEMPSPACE1	1	SMS	SYSTEMP	0
USERSPACE1	2	DMS	LONG	0
SYSTOOLSPA	3	DMS	LONG	0
TEMPSPACE1	1	SMS	SYSTEMP	2
USERSPACE1	2	DMS	LONG	2

8 record(s) selected.

## 戻される情報

表 219. SNAPTbsp 管理ビューおよび SNAP\_GET\_TBSP 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TBSP_ID	BIGINT	tablespace_id - 表スペース ID

表 219. SNAPTBSP 管理ビューおよび SNAP\_GET\_TBSP 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_TYPE	VARCHAR(10)	tablespace_type - 表スペース・タイプ。このインターフェースは、sqlutil.h の定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• DMS</li> <li>• SMS</li> </ul>
TBSP_CONTENT_TYPE	VARCHAR(10)	tablespace_content_type - 表スペースのコンテンツ・タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• ANY</li> <li>• LARGE</li> <li>• SYSTEMP</li> <li>• USRTEMP</li> </ul>
TBSP_PAGE_SIZE	BIGINT	tablespace_page_size - 表スペースのページ・サイズ
TBSP_EXTENT_SIZE	BIGINT	tablespace_extent_size - 表スペースのエクステント・サイズ
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 表スペースのプリフェッチ・サイズ
TBSP_CUR_POOL_ID	BIGINT	tablespace_cur_pool_id - 現在使用中のバッファーク・プール
TBSP_NEXT_POOL_ID	BIGINT	tablespace_next_pool_id - 次の始動時に使用されるバッファーク・プール
FS_CACHING	SMALLINT	fs_caching - ファイル・システム・キャッシング
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファーク・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファーク・プール・データの物理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファーク・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファーク・プール一時データの物理読み取り
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - バッファーク・プール非同期データ読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファーク・プールへのデータの書き込み



表 219. SNAPTBSP 管理ビューおよび SNAP\_GET\_TBSP 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - バッファ ー・プール非同期データ書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ ー・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ ー・プール索引の物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ ー・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ ー・プール一時索引の物理読み取り
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - バッファ ー・プール非同期索引読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ ー・プール索引の書き込み
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - バッファ ー・プール非同期索引書き込み
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ ー・プール XDA データの論理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ ー・プール XDA データの物理読み取り
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ ー・プール XDA データの書き込み
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - バッファ ー・プール非同期 XDA データ読 み取り
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - バッファ ー・プール非同期 XDA データ書 き込み
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ ー・プール一時 XDA データの論 理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ ー・プール一時 XDA データの物 理読み取り : モニター・エレメン ト
POOL_READ_TIME	BIGINT	pool_read_time - バッファ ー・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ ー・プール物理書き込み時間の合計
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - バッファ ー・プール非同期読み取り時間
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - バッファ ー・プール非同期書き込み時間

表 219. SNAP\_TBSP 管理ビューおよび SNAP\_GET\_TBSP 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - バッファ・プール非同期読み取り要求
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - バッファ・プール非同期索引読み取り要求
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - バッファ・プール非同期 XDA 読み取り要求
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - バッファ・プールの非ビクティム・バッファ数
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
FILES_CLOSED	BIGINT	files_closed - 閉じられたデータベース・ファイル
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不能プリフェッチ・ページ
TBSP_REBALANCER_MODE	VARCHAR(10)	tablespace_rebalancer_mode - リバランサー・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• NO_REBAL</li> <li>• FWD_REBAL</li> <li>• REV_REBAL</li> </ul>
TBSP_USING_AUTO_STORAGE	SMALLINT	tablespace_using_auto_storage - 自動ストレージが使用可能な表スペース
TBSP_AUTO_RESIZE_ENABLED	SMALLINT	tablespace_auto_resize_enabled - 表スペースの自動サイズ変更可能
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPTbsp\_part 管理ビューおよび SNAP\_GET\_TBSP\_PART 表関数 - tablespace\_nodeinfo 論理データ・グループのスナップショット情報の取得

SNAPTbsp\_part 管理ビューおよび SNAP\_GET\_TBSP\_PART 表関数は、tablespace\_nodeinfo 論理データ・グループからのスナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPTbsp\_part 管理ビュー』
- 942 ページの 『SNAP\_GET\_TBSP\_PART 表関数』

### SNAPTbsp\_part 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関する tablespace\_nodeinfo 論理データ・グループのスナップショット情報を検索することができます。

SNAPTbsp、SNAPTbsp\_quiescer、SNAPTbsp\_range、SNAPCONTAINER 管理ビューと併せて使用することにより、SNAPTbsp\_part 管理ビューは **GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を戻します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、944 ページの表 220 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPTbsp\_range 管理ビューに対する SELECT 特権
- SNAPTbsp\_range 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかの権限が必要です。

- SNAP\_GET\_TBSP\_PART 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT

- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベースのすべてのデータベース・パーティションの表スペースとその状態のリストを検索します。

```
SELECT SUBSTR(TBSP_NAME,1,30) AS TBSP_NAME, TBSP_ID,
       SUBSTR(TBSP_STATE,1,30) AS TBSP_STATE, DBPARTITIONNUM
FROM SYSIBMADM.SNAPTbsp_PART
```

以下はこの照会の出力例です。

TBSP_NAME	TBSP_ID	TBSP_STATE	DBPARTITIONNUM
SYSCATSPACE	0	NORMAL	0
TEMPSPACE1	1	NORMAL	0
USERSPACE1	2	NORMAL	0
TEMPSPACE1	1	NORMAL	1
USERSPACE1	2	NORMAL	1

5 record(s) selected.

## SNAP\_GET\_TBSP\_PART 表関数

SNAP\_GET\_TBSP\_PART 表関数は SNAPTbsp\_PART 管理ビューと同じ情報を戻します。ただし、SNAP\_GET\_TBSP\_PART 表関数の場合は、特定のデータベース・パーティション、すべてのデータベース・パーティションの集約、またはすべてのデータベース・パーティションについて、特定のデータベースに関する情報を取得できます。

表関数 SNAP\_GET\_TBSP、SNAP\_GET\_TBSP QUIESCER、SNAP\_GET\_TBSP\_RANGE、SNAP\_GET\_CONTAINER と併せて使用することにより、SNAP\_GET\_TBSP\_PART 表関数は CLP コマンド **GET SNAPSHOT FOR TABLESPACES ON database-alias** と同等の情報を戻します。

戻される可能性のある情報の完全なリストは、944 ページの表 220 を参照してください。

### 構文

```
▶▶ SNAP_GET_TBSP_PART ( ( dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

*dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home"

のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### *member*

有効なメンバー番号を指定する、タイプ INTEGER のオプション入力引数。現在のメンバーの場合は -1、すべてのアクティブなメンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプションが使用されない場合、つまり dbname だけが指定される場合は、データベースがアクティブになっているすべてのメンバーからデータが戻されます。

dbname と member の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_TBSP\_PART 表関数は、現在接続されているデータベースおよびメンバーのスナップショットを取得します。

#### 許可

- SYSMON 権限
- SNAP\_GET\_TBSP\_PART 表関数に対する EXECUTE 特権

#### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

#### 例

接続されているデータベースの接続されているデータベース・パーティションの表スペースとその状態のリストを検索します。

```
SELECT SUBSTR(TBSP_NAME,1,30) AS TBSP_NAME, TBSP_ID,
       SUBSTR(TBSP_STATE,1,30) AS TBSP_STATE
FROM TABLE(SNAP_GET_TBSP_PART(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下はこの照会の出力例です。

TBSP_NAME	TBSP_ID	TBSP_STATE
SYSCATSPACE		0 NORMAL
TEMPSPACE1		1 NORMAL
USERSPACE1		2 NORMAL
SYSTOOLSPACE		3 NORMAL
SYSTOOLSTMPSPACE		4 NORMAL

5 record(s) selected.

## 戻される情報

表 220. *SNAPTbsp\_Part* 管理ビューおよび *SNAP\_Get\_Tbsp\_Part* 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_NAME	VARCHAR (128)	tablespace_name - 表スペース名
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
TBSP_STATE	VARCHAR (256)	<p>tablespace_state - 表スペースの状態。このインターフェースは、<code>sqlutil.h</code> での定義に基づくテキスト ID を戻します。これは次のものを「+」符号で分離して組み合わせたものになります。</p> <ul style="list-style-type: none"> <li>• BACKUP_IN_PROGRESS</li> <li>• BACKUP_PENDING</li> <li>• DELETE_PENDING</li> <li>• DISABLE_PENDING</li> <li>• DROP_PENDING</li> <li>• LOAD_IN_PROGRESS</li> <li>• LOAD_PENDING</li> <li>• NORMAL</li> <li>• OFFLINE</li> <li>• PSTAT_CREATION</li> <li>• PSTAT_DELETION</li> <li>• QUIESCED_EXCLUSIVE</li> <li>• QUIESCED_SHARE</li> <li>• QUIESCED_UPDATE</li> <li>• REBAL_IN_PROGRESS</li> <li>• REORG_IN_PROGRESS</li> <li>• RESTORE_IN_PROGRESS</li> <li>• RESTORE_PENDING</li> <li>• ROLLFORWARD_IN_PROGRESS</li> <li>• ROLLFORWARD_PENDING</li> <li>• STORDEF_ALLOWED</li> <li>• STORDEF_CHANGED</li> <li>• STORDEF_FINAL_VERSION</li> <li>• STORDEF_PENDING</li> <li>• SUSPEND_WRITE</li> </ul>
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 表スペースのプリフェッチ・サイズ
TBSP_NUM QUIESCERS	BIGINT	tablespace_num_quiescers - 静止プログラム数

表 220. SNAPTBSP\_PART 管理ビューおよび SNAP\_GET\_TBSP\_PART 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_STATE_CHANGE_OBJECT_ID	BIGINT	tablespace_state_change_object_id - 状態変更オブジェクト ID
TBSP_STATE_CHANGE_TBSP_ID	BIGINT	tablespace_state_change_ts_id - 状態変更表スペース ID
TBSP_MIN_RECOVERY_TIME	TIMESTAMP	tablespace_min_recovery_time - ロールフォワードの最小リカバリー時間
TBSP_TOTAL_PAGES	BIGINT	tablespace_total_pages - 表スペース内の合計ページ数
TBSP_USABLE_PAGES	BIGINT	tablespace_usable_pages - 表スペース内の使用可能ページ数
TBSP_USED_PAGES	BIGINT	tablespace_used_pages - 表スペース内の使用されているページ数
TBSP_FREE_PAGES	BIGINT	tablespace_free_pages - 表スペース内のフリー・ページ数
TBSP_PENDING_FREE_PAGES	BIGINT	tablespace_pending_free_pages - 表スペース内のペンディング・フリー・ページ数
TBSP_PAGE_TOP	BIGINT	tablespace_page_top - 表スペース最高水準点
REBALANCER_MODE	VARCHAR (30)	tablespace_rebalancer_mode - リバランサー・モード。このインターフェースは、sqlmon.hでの定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• FWD_REBAL</li> <li>• NO_REBAL</li> <li>• REV_REBAL</li> <li>• FWD_REBAL_OF_2PASS</li> <li>• REV_REBAL_OF_2PASS</li> </ul>
REBALANCER_EXTENTS_REMAINING	BIGINT	tablespace_rebalancer_extents_remaining - リバランサーで処理されるエクステントの合計数
REBALANCER_EXTENTS_PROCESSED	BIGINT	tablespace_rebalancer_extents_processed - リバランサーで処理されたエクステントの数
REBALANCER_PRIORITY	BIGINT	tablespace_rebalancer_priority - 現行のリバランサー優先順位
REBALANCER_START_TIME	TIMESTAMP	tablespace_rebalancer_start_time - リバランサー開始時刻
REBALANCER_RESTART_TIME	TIMESTAMP	tablespace_rebalancer_restart_time - リバランサー再始動時刻
REBALANCER_LAST_EXTENT_MOVED	BIGINT	tablespace_rebalancer_last_extent_moved - リバランサーによって最後に移動されたエクステント
TBSP_NUM_RANGES	BIGINT	tablespace_num_ranges - 表スペース・マップ内の範囲数

表 220. *SNAPTbsp\_Part* 管理ビューおよび *SNAP\_Get\_Tbsp\_Part* 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_NUM_CONTAINERS	BIGINT	tablespace_num_containers - 表スペース内のコンテナ数
TBSP_INITIAL_SIZE	BIGINT	tablespace_initial_size - 表スペースの初期サイズ
TBSP_CURRENT_SIZE	BIGINT	tablespace_current_size - 表スペースの現行サイズ
TBSP_MAX_SIZE	BIGINT	tablespace_max_size - 表スペースの最大サイズ
TBSP_INCREASE_SIZE	BIGINT	tablespace_increase_size - バイト単位のサイズの増加
TBSP_INCREASE_SIZE_PERCENT	SMALLINT	tablespace_increase_size_percent - パーセント単位のサイズの増加
TBSP_LAST_RESIZE_TIME	TIMESTAMP	tablespace_last_resize_time - 最後にサイズ変更が正常に行われた時刻
TBSP_LAST_RESIZE_FAILED	SMALLINT	tablespace_last_resize_failed - 失敗した最後のサイズ変更
TBSP_PATHS_DROPPED	SMALLINT	表スペースが、ドロップされた 1 つ以上のストレージ・パスにあることを示します (0 - いいえ、1 - はい)
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

## SNAPTbsp\_Quiescer 管理ビューおよび SNAP\_Get\_Tbsp\_Quiescer 表関数 - quiescer 表スペース・スナップショット情報の検索

SNAPTbsp\_Quiescer 管理ビューおよび SNAP\_Get\_Tbsp\_Quiescer 表関数は、表スペース・スナップショットから、静止プログラムに関する情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPTbsp\_Quiescer 管理ビュー』
- 949 ページの 『SNAP\_Get\_Tbsp\_Quiescer 表関数』

### SNAPTbsp\_Quiescer 管理ビュー

この管理ビューでは、現在接続されているデータベースの静止プログラム表スペース・スナップショット情報を検索できます。

SNAPTbsp、SNAPTbsp\_Part、SNAPTbsp\_Range、SNAPContainer 管理ビューと共に使用すると、SNAPTbsp\_Quiescer 管理ビューは、**GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を提供します。



スキーマは SYSIBMADM です。

戻ることができる情報の全リストについては、950 ページの表 221を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPTbsp\_QUIESCER 管理ビューに対する SELECT 特権
- SNAPTbsp\_QUIESCER 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_TBSP\_QUIESCER 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースのすべてのデータベース・メンバーの静止した表スペースの情報を検索します。

```
SELECT SUBSTR(TBSP_NAME, 1, 10) AS TBSP_NAME, QUIESCER_TS_ID,  
       QUIESCER_OBJ_ID, QUIESCER_AUTH_ID, QUIESCER_AGENT_ID,  
       QUIESCER_STATE, DBPARTITIONNUM  
FROM SYSIBMADM.SNAPTbsp_QUIESCER ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

TBSP_NAME	QUIESCER_TS_ID	QUIESCER_OBJ_ID	QUIESCER_AUTH_ID	..
USERSPACE1	2		5 SWALKTY	..
USERSPACE1	2		5 SWALKTY	..

2 record(s) selected.

この照会からの出力 (続き)。

```

... QUIESCER_AGENT_ID    QUIESCER_STATE DBPARTITIONNUM
... -----
...                0 EXCLUSIVE                0
...                65983 EXCLUSIVE            1

```

### 例: パーティション表の名前の判別

表がパーティション化されていて、静止状態に保たれている場合、表スペース ID および表 ID の値は、SYSCAT.TABLES 内のものとは異なる表記となります。これらの ID は、符号なしの短い表記となります。静止した表の名前を検索するには、QUIESCER\_TS\_ID から 65536 (最大値) を減算した表スペース ID を計算することによって、まず符号付きの短い表記を検索する必要があります。それから、この表スペース ID を使用して、静止した表を特定します。(実際の表スペース ID は、表内の各データ・パーティションの SYSCAT.DATAPARTITIONS にあります。)

```

SELECT SUBSTR(TBSP_NAME, 1, 10) AS TBSP_NAME,
       CASE WHEN QUIESCER_TS_ID = 65530
            THEN QUIESCER_TS_ID - 65536
            ELSE QUIESCER_TS_ID END as tbspaceid,
       CASE WHEN QUIESCER_TS_ID = 65530
            THEN QUIESCER_OBJ_ID - 65536
            ELSE QUIESCER_OBJ_ID END as tableid
FROM SYSIBMADM.SNAPTbsp_QUIESCER
ORDER BY DBPARTITIONNUM

```

以下はこの照会の出力例です。

```

TBSP_NAME    TSPACEID    TABLEID
-----
TABDATA      -6           -32768
DATAMART     -6           -32765
SMALL        5            17

```

3 record(s) selected.

この照会で得られた特定の TSPACEID および TABLEID を使用して、SYSCAT.TABLES から表スキーマおよび表名を検索します。

```

SELECT CHAR(tbschema, 10)tbschema, CHAR(tabname,15)tabname
FROM SYSCAT.TABLES
WHERE tspaceid = -6 AND tableid in (-32768,-32765)

```

以下はこの照会の出力例です。

```

TBSHEMA     TABNAME
-----
SAMP        ORDERS_RP
SAMP        ORDERS_DMART

```

2 record(s) selected.

```

SELECT CHAR(tbschema, 10)tbschema, CHAR(tabname,15)tabname
FROM SYSCAT.TABLES
WHERE tspaceid = 5 AND tableid = 17

```

以下はこの照会の出力例です。

```

TBSHEMA     TABNAME
-----
SAMP        NATION

```

1 record(s) selected.

## SNAP\_GET\_TBSP QUIESCER 表関数

SNAP\_GET\_TBSP QUIESCER 表関数は SNAPTBSP QUIESCER 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_TBSP、SNAP\_GET\_TBSP\_PART、SNAP\_GET\_TBSP\_RANGE、SNAP\_GET\_CONTAINER 表関数と共に使用すると、SNAP\_GET\_TBSP QUIESCER 表関数は、**GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を提供します。

戻ることができる情報の全リストについては、950 ページの表 221を参照してください。

### 構文

```
▶▶ SNAP_GET_TBSP QUIESCER (—dbname——)——▶▶  
└──, member──┘
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_TBSP QUIESCER 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_TBSP\_QUIESCER 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースのデータベース・メンバー 1 の静止した表スペースの情報を検索します。

```
SELECT SUBSTR(TBSP_NAME, 1, 10) AS TBSP_NAME, QUIESCER_TS_ID,  
       QUIESCER_OBJ_ID, QUIESCER_AUTH_ID, QUIESCER_AGENT_ID,  
       QUIESCER_STATE, DBPARTITIONNUM  
FROM TABLE( SYSPROC.SNAP_GET_TBSP_QUIESCER( ' ', 1)) AS T
```

以下はこの照会の出力例です。

```
TBSP_NAME  QUIESCER_TS_ID    QUIESCER_OBJ_ID    QUIESCER_AUTH_ID    ...  
-----  
USERSPACE1          2                5 SWALKTY            ...
```

1 record(s) selected.

この照会からの出力 (続き)。

```
... QUIESCER_AGENT_ID  QUIESCER_STATE DBPARTITIONNUM  
... -----  
...                65983 EXCLUSIVE                1
```

## 戻される情報

表 221. SNAPTbsp\_QUIESCER 管理ビューおよび SNAP\_GET\_TBSP\_QUIESCER 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
QUIESCER_TS_ID	BIGINT	quiescer_ts_id - 静止プログラム表スペース ID
QUIESCER_OBJ_ID	BIGINT	quiescer_obj_id - 静止プログラム・オブジェクト ID

表 221. *SNAPTbsp\_QUIESCER* 管理ビューおよび *SNAP\_GET\_TBSP\_QUIESCER* 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
QUIESCER_AUTH_ID	VARCHAR(128)	quiescer_auth_id - 静止プログラム・ユーザー許可 ID
QUIESCER_AGENT_ID	BIGINT	quiescer_agent_id - 静止プログラム・エージェント ID
QUIESCER_STATE	VARCHAR(14)	quiescer_state - 静止プログラムの状態。このインターフェースは、 <code>sqlutil.h</code> での定義を基にしてテキスト ID を戻します。以下のいずれかとなります。 <ul style="list-style-type: none"> <li>• EXCLUSIVE</li> <li>• UPDATE</li> <li>• SHARE</li> </ul>
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPTbsp\_RANGE 管理ビューおよび SNAP\_GET\_TBSP\_RANGE 表関数 - 範囲スナップショット情報の検索

*SNAPTbsp\_RANGE* 管理ビューおよび *SNAP\_GET\_TBSP\_RANGE* 表関数は、範囲スナップショットから情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『*SNAPTbsp\_RANGE* 管理ビュー』
- 953 ページの 『*SNAP\_GET\_TBSP\_RANGE* 表関数』

### SNAPTbsp\_RANGE 管理ビュー

この管理ビューでは、現在接続されているデータベースの範囲スナップショット情報を検索できます。

*SNAPTbsp*、*SNAPTbsp\_PART*、*SNAPTbsp\_QUIESCER*、および *SNAPCONTAINER* 管理ビューと共に使用すると、*SNAPTbsp\_RANGE* 管理ビューは、**GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を提供します。

スキーマは *SYSIBMADM* です。

戻される可能性のある情報の完全なリストは、955 ページの表 222 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPTbsp\_RANGE 管理ビューに対する SELECT 特権
- SNAPTbsp\_RANGE 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_Tbsp\_RANGE 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースのすべてのデータベース・メンバーの表スペース範囲についての情報を選択します。

```
SELECT Tbsp_ID, SUBSTR(Tbsp_NAME, 1, 15) AS Tbsp_NAME, RANGE_NUMBER,
       RANGE_STRIPE_SET_NUMBER, RANGE_OFFSET, RANGE_MAX_PAGE,
       RANGE_MAX_EXTENT, RANGE_START_STRIPE, RANGE_END_STRIPE,
       RANGE_ADJUSTMENT, RANGE_NUM_CONTAINER, RANGE_CONTAINER_ID,
       DBPARTITIONNUM FROM SYSIBMADM.SNAPTbsp_RANGE
ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

Tbsp_ID	Tbsp_NAME	RANGE_NUMBER	RANGE_STRIPE_SET_NUMBER	...
0	SYSCATSPACE	0	0	...
2	USERSPACE1	0	0	...
3	SYSTOOLSPACE	0	0	...
2	USERSPACE1	0	0	...
2	USERSPACE1	0	0	...

5 record(s) selected.

この照会からの出力 (続き)。

...	RANGE_OFFSET	RANGE_MAX_PAGE	RANGE_MAX_EXTENT	...
...	0	11515	2878	...
...	0	479	14	...
...	0	251	62	...
...	0	479	14	...
...	0	479	14	...

この照会からの出力 (続き)。

...	RANGE_START_STRIPE	RANGE_END_STRIPE	RANGE_ADJUSTMENT	...
...	0	2878	0	...
...	0	14	0	...
...	0	62	0	...
...	0	14	0	...
...	0	14	0	...

この照会からの出力 (続き)。

...	RANGE_NUM_CONTAINER	RANGE_CONTAINER_ID	DBPARTITIONNUM
...	1	0	0
...	1	0	0
...	1	0	0
...	1	0	1
...	1	0	2

## SNAP\_GET\_TBSP\_RANGE 表関数

SNAP\_GET\_TBSP\_RANGE 表関数は SNAPTbsp\_RANGE 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_TBSP、SNAP\_GET\_TBSP\_PART、SNAP\_GET\_TBSP\_QUIESCER、および SNAP\_GET\_CONTAINER 表関数と共に使用すると、SNAP\_GET\_TBSP\_RANGE 表関数は、**GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を提供します。

戻される可能性のある情報の完全なリストは、955 ページの表 222 を参照してください。

### 構文

```

▶▶ SNAP_GET_TBSP_RANGE ( ( dbname [ , member ] ) )

```

スキーマは SYSPROC です。

### 表関数パラメーター

*dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレク

トリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ **INTEGER** のオプションの入力引数。現在のデータベース・メンバーの場合は **-1**、すべてアクティブなデータベース・メンバーの集合の場合は **-2** を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に **-1** に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されず、アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、**SNAP\_WRITE\_FILE** プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、**SNAP\_GET\_TBSP\_RANGE** 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

### 許可

以下のいずれかの権限が必要です。

- **SNAP\_GET\_TBSP\_RANGE** 表関数に対する **EXECUTE** 特権
- **DATAACCESS** 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- **SYSMON**
- **SYSCTRL**
- **SYSMAINT**
- **SYSADM**

### デフォルトの **PUBLIC** 特権

制限のないデータベースでは、この関数が自動的に作成されると、**EXECUTE** 特権が **PUBLIC** に付与されます。

### 例

現在接続されているデータベース・メンバー上で *tbasp\_id* = 2 である表スペースの表スペース範囲の情報を選択します。

```
SELECT TBSP_ID, SUBSTR(TBSP_NAME, 1, 15) AS TBSP_NAME, RANGE_NUMBER,
       RANGE_STRIPE_SET_NUMBER, RANGE_OFFSET, RANGE_MAX_PAGE, RANGE_MAX_EXTENT,
       RANGE_START_STRIPE, RANGE_END_STRIPE, RANGE_ADJUSTMENT,
       RANGE_NUM_CONTAINER, RANGE_CONTAINER_ID
FROM TABLE(SNAP_GET_TBSP_RANGE('',-1)) AS T WHERE TBSP_ID = 2
```

以下はこの照会の出力例です。



```

TBSP_ID      TBSP_NAME      RANGE_NUMBER      ...
-----
2 USERSPACE1      0 ...

```

1 record(s) selected.

この照会からの出力 (続き)。

```

... RANGE_STRIPE_SET_NUMBER RANGE_OFFSET      RANGE_MAX_PAGE      ...
... -----
...              0              0              3967 ...

```

この照会からの出力 (続き)。

```

... RANGE_MAX_EXTENT      RANGE_START_STRIPE      RANGE_END_STRIPE      ...
... -----
...              123              0              123 ...

```

この照会からの出力 (続き)。

```

... RANGE_ADJUSTMENT      RANGE_NUM_CONTAINER      RANGE_CONTAINER_ID
... -----
...              0              1              0

```

## 戻される情報

表 222. *SNAPTBSP\_RANGE* 管理ビューおよび *SNAP\_GET\_TBSP\_RANGE* 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
RANGE_NUMBER	BIGINT	range_number - 範囲番号
RANGE_STRIPE_SET_NUMBER	BIGINT	range_stripe_set_number - ストライプ・セット番号
RANGE_OFFSET	BIGINT	range_offset - 範囲オフセット
RANGE_MAX_PAGE	BIGINT	range_max_page_number - 範囲内の最大ページ
RANGE_MAX_EXTENT	BIGINT	range_max_extent - 範囲内の最大エクステンツ
RANGE_START_STRIPE	BIGINT	range_start_stripe - 開始ストライプ
RANGE_END_STRIPE	BIGINT	range_end_stripe - 終了ストライプ
RANGE_ADJUSTMENT	BIGINT	range_adjustment - 範囲調整
RANGE_NUM_CONTAINER	BIGINT	range_num_containers - 範囲内コンテナの数
RANGE_CONTAINER_ID	BIGINT	range_container_id - 範囲コンテナ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

## SNAPUTIL 管理ビューおよび SNAP\_GET\_UTIL 表関数 - utility\_info 論理データ・グループ・スナップショット情報の検索

SNAPUTIL 管理ビューおよび SNAP\_GET\_UTIL 表関数は、utility\_info 論理データ・グループからのユーティリティー・スナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPUTIL 管理ビュー』
- 957 ページの『SNAP\_GET\_UTIL 表関数』

### SNAPUTIL 管理ビュー

SNAPUTIL\_PROGRESS 管理ビューと組み合わせて使用すると、SNAPUTIL 管理ビューは、**LIST UTILITIES SHOW DETAIL CLP** コマンドと同じ情報を提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、959 ページの表 223 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPUTIL 管理ビューに対する SELECT 特権
- SNAPUTIL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_UTIL 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

接続されているデータベースが含まれるインスタンス内のすべてのアクティブ・データベースのすべてのデータベース・メンバー上にある、ユーティリティとその状態のリストを検索します。

```
SELECT UTILITY_TYPE, UTILITY_PRIORITY, SUBSTR(UTILITY_DESCRIPTION, 1, 72)
      AS UTILITY_DESCRIPTION, SUBSTR(UTILITY_DBNAME, 1, 17) AS
      UTILITY_DBNAME, UTILITY_STATE, UTILITY_INVOKER_TYPE, DBPARTITIONNUM
FROM SYSIBMADM.SNAPUTIL ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

```
UTILITY_TYPE      UTILITY_PRIORITY ...
-----
LOAD              - ...
LOAD              - ...
LOAD              - ...
```

3 record(s) selected.

この照会からの出力 (続き)。

```
... UTILITY_DESCRIPTION ...
... -----
... ONLINE LOAD DEL AUTOMATIC INDEXING INSERT COPY NO TEST .LOADTEST ...
... ONLINE LOAD DEL AUTOMATIC INDEXING INSERT COPY NO TEST .LOADTEST ...
... ONLINE LOAD DEL AUTOMATIC INDEXING INSERT COPY NO TEST .LOADTEST ...
```

この照会からの出力 (続き)。

```
... UTILITY_DBNAME  UTILITY_STATE UTILITY_INVOKER_TYPE DBPARTITIONNUM
... -----
... SAMPLE          EXECUTE      USER                0
... SAMPLE          EXECUTE      USER                1
... SAMPLE          EXECUTE      USER                2
```

## SNAP\_GET\_UTIL 表関数

SNAP\_GET\_UTIL 表関数は SNAPUTIL 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーを対象とした情報を検索することができます。

SNAP\_GET\_UTIL\_PROGRESS 表関数と組み合わせて使用すると、SNAP\_GET\_UTIL 表関数は、**LIST UTILITIES SHOW DETAIL CLP** コマンドと同じ情報を提供します。

戻される可能性のある情報の完全なリストは、959 ページの表 223 を参照してください。

## 構文

```
▶▶ SNAP_GET_UTIL ( [member] ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

*member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプション

ンの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_UTIL 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_UTIL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: データベース SAMPLE 上の現在接続されているデータベース・メンバーのユーティリティー ID (そのタイプと状態を含む) のリストを検索します。

```
SELECT UTILITY_ID, UTILITY_TYPE, UTILITY_STATE
FROM TABLE(SNAP_GET_UTIL(-1)) AS T WHERE UTILITY_DBNAME='SAMPLE'
```

以下はこの照会の出力例です。

UTILITY_ID	UTILITY_TYPE	STATE
-----	-----	-----
	1 BACKUP	EXECUTE

1 record(s) selected.

例 2: 現在接続されているデータベース・メンバーのユーティリティー ID とそのタイプ、メンバー番号、およびデータベース・パーティション番号のリストを検索します。

```
SELECT UTILITY_ID, UTILITY_TYPE, MEMBER, DBPARTITIONNUM
FROM TABLE(SNAP_GET_UTIL(-1)) AS T
```

以下はこの照会の出力例です。

```

UTILITY_ID  UTILITY_TYPE                MEMBER DBPARTITIONNUM
-----
          2 BACKUP                2                2
  
```

## 戻される情報

表 223. SNAPUTIL 管理ビューおよび SNAP\_GET\_UTIL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
UTILITY_ID	INTEGER	utility_id - ユーティリティー ID。データベース・パーティションに固有。
UTILITY_TYPE	VARCHAR(26)	utility_type - ユーティリティー・タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• ASYNC_INDEX_CLEANUP</li> <li>• BACKUP</li> <li>• CRASH_RECOVERY</li> <li>• LOAD</li> <li>• REBALANCE</li> <li>• REDISTRIBUTE</li> <li>• RESTART_RECREATE_INDEX</li> <li>• RESTORE</li> <li>• ROLLFORWARD_RECOVERY</li> <li>• RUNSTATS</li> <li>• GROUP_CRASH_RECOVERY</li> <li>• MEMBER_CRASH_RECOVERY</li> </ul>
UTILITY_PRIORITY	INTEGER	utility_priority - ユーティリティー優先度。ユーティリティーがスロットルをサポートする場合には優先順位、それ以外の場合は NULL。
UTILITY_DESCRIPTION	VARCHAR(2048)	utility_description - ユーティリティー記述。NULL にすることもできます。
UTILITY_DBNAME	VARCHAR(128)	utility_dbname - ユーティリティーで操作されるデータベース
UTILITY_START_TIME	TIMESTAMP	utility_start_time - ユーティリティー開始時刻

表 223. SNAPUTIL 管理ビューおよび SNAP\_GET\_UTIL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
UTILITY_STATE	VARCHAR(10)	utility_state - ユーティリティー状態。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• ERROR</li> <li>• EXECUTE</li> <li>• WAIT</li> </ul>
UTILITY_INVOKER_TYPE	VARCHAR(10)	utility_invoker_type - ユーティリティー呼び出し側タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• AUTO</li> <li>• USER</li> </ul>
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
PROGRESS_LIST_ATTR	VARCHAR(10)	progress_list_attr - 現在の進行リストの属性
PROGRESS_LIST_CUR_SEQ_NUM	INTEGER	progress_list_current_seq_num - 現在の進行リストのシーケンス番号
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPUTIL\_PROGRESS 管理ビューおよび SNAP\_GET\_UTIL\_PROGRESS 表関数 - progress 論理データ・グループ・スナップショット情報の検索

SNAPUTIL\_PROGRESS 管理ビューおよび SNAP\_GET\_UTIL\_PROGRESS 表関数は、特に progress 論理データ・グループのユーティリティー進行状況スナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPUTIL\_PROGRESS 管理ビュー』
- 961 ページの『SNAP\_GET\_UTIL\_PROGRESS 表関数』

### SNAPUTIL\_PROGRESS 管理ビュー

SNAPUTIL 管理ビューと組み合わせて使用すると、SNAPUTIL\_PROGRESS 管理ビューは、LIST UTILITIES SHOW DETAIL CLP コマンドと同じ情報を提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、963 ページの表 224 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPUTIL\_PROGRESS 管理ビューに対する SELECT 特権
- SNAPUTIL\_PROGRESS 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_UTIL\_PROGRESS 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

ユーティリティー ID ごとの合計進行単位および完了した進行単位の詳細を検索します。

```
SELECT UTILITY_ID, PROGRESS_TOTAL_UNITS, PROGRESS_COMPLETED_UNITS,  
       DBPARTITIONNUM FROM SYSIBMADM.SNAPUTIL_PROGRESS
```

以下はこの照会の出力例です。

UTILITY_ID	PROGRESS_TOTAL_UNITS	PROGRESS_COMPLETED_UNITS	DBPARTITIONNUM
7	10	5	0
9	10	5	1

1 record(s) selected.

## SNAP\_GET\_UTIL\_PROGRESS 表関数

SNAP\_GET\_UTIL\_PROGRESS 表関数は SNAPUTIL\_PROGRESS 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メン

パーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_UTIL 表関数と組み合わせて使用すると、SNAP\_GET\_UTIL\_PROGRESS 表関数は、**LIST UTILITIES SHOW DETAIL CLP** コマンドと同じ情報を提供します。

戻される可能性のある情報の完全なリストは、963 ページの表 224 を参照してください。

## 構文

```
▶▶—SNAP_GET_UTIL_PROGRESS—(—member—)
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_UTIL\_PROGRESS 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_UTIL\_PROGRESS 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM



## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

現在接続されているメンバー上のユーティリティの進行状況の詳細を検索します。

```
SELECT UTILITY_ID, PROGRESS_TOTAL_UNITS, PROGRESS_COMPLETED_UNITS,  
       DBPARTITIONNUM FROM TABLE(SNAP_GET_UTIL_PROGRESS(-1)) as T
```

以下はこの照会の出力例です。

```
UTILITY_ID PROGRESS_TOTAL_UNITS PROGRESS_COMPLETED_UNITS DBPARTITIONNUM  
-----  
          7                10                5                0
```

1 record(s) selected.

### 戻される情報

表 224. SNAPUTIL\_PROGRESS 管理ビューおよび SNAP\_GET\_UTIL\_PROGRESS 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
UTILITY_ID	INTEGER	utility_id - ユーティリティ ID。 データベース・パーティションに固有。
PROGRESS_SEQ_NUM	INTEGER	progress_seq_num - 進行シーケンス番号。逐次の場合、フェーズの数。並行の場合、NULL の場合があります。
UTILITY_STATE	VARCHAR(16)	utility_state - ユーティリティ状態。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"><li>• ERROR</li><li>• EXECUTE</li><li>• WAIT</li></ul>
PROGRESS_DESCRIPTION	VARCHAR(2048)	progress_description - 進行の記述
PROGRESS_START_TIME	TIMESTAMP	progress_start_time - 進行開始時刻。フェーズが開始済みの場合には開始時刻、それ以外の場合は NULL。

表 224. SNAPUTIL\_PROGRESS 管理ビューおよび SNAP\_GET\_UTIL\_PROGRESS 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
PROGRESS_WORK_METRIC	VARCHAR(16)	progress_work_metric - 進行作業メトリック。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• NOT_SUPPORT</li> <li>• BYTES</li> <li>• EXTENTS</li> <li>• INDEXES</li> <li>• PAGES</li> <li>• ROWS</li> <li>• TABLES</li> </ul>
PROGRESS_TOTAL_UNITS	BIGINT	progress_total_units - 合計進行作業単位
PROGRESS_COMPLETED_UNITS	BIGINT	progress_completed_units - 完了した進行作業単位
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAP\_WRITE\_FILE プロシージャ

SNAP\_WRITE\_FILE プロシージャはシステム・スナップショット・データを、インスタンス・ディレクトリーの tmp サブディレクトリーにあるファイルに書き込みます。

### 構文

```
▶▶—SNAP_WRITE_FILE—(—requestType—,—dbname—,—member—)————▶▶
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### requestType

有効なスナップショット要求タイプを指定する、タイプ VARCHAR(32) の入力引数。可能な要求タイプは、sqlmon.h での定義を基にしたテキスト ID です。以下のいずれかとなります。

- APPL\_ALL
- BUFFERPOOLS\_ALL
- DB2

- DBASE\_ALL
- DBASE\_LOCKS
- DBASE\_TABLES
- DBASE\_TABLESPACES
- DYNAMIC\_SQL

#### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### *member*

有効なメンバー番号を指定する、タイプ INTEGER の入力引数。現在のメンバーの場合は -1、すべてのアクティブなメンバーの集合の場合は -2 を指定します。アクティブなメンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

プロシージャを実行するには、ユーザーに SYSADM、SYSCTRL、SYSMON、または SYSMON 権限が必要です。保存されたスナップショットは、スナップショット表関数への入力として NULL 値を渡すことにより、SYSADM、SYSCTRL、SYSMAINT、または SYSMON 権限のないユーザーでも読み取れます。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

'DB2' の要求タイプ (SQLMA\_DB2 に相当) を指定し、現在接続されているデータベースおよび現行データベース・パーティションをデフォルトにすることで、データベース・マネージャー情報のスナップショットをとります。

```
CALL SYSPROC.SNAP_WRITE_FILE ('DB2', '', -1)
```

この場合、スナップショット・データは、インスタンス一時ディレクトリー (UNIX オペレーティング・システムでは sqllib/tmp/SQLMA\_DB2.dat、Windows オペレーティング・システムでは sqllib¥DB2¥tmp¥SQLMA\_DB2.dat) に書き込まれます。

## 使用上の注意

未認識の入力パラメーターが指定された場合、「SQL2032N "REQUEST\_TYPE" パラメーターが無効です」エラーが戻されます。

## SNAPAGENT 管理ビューおよび SNAP\_GET\_AGENT 表関数 - agent 論理データ・グループのアプリケーション・スナップショット情報の検索

SNAPAGENT 管理ビューおよび SNAP\_GET\_AGENT 表関数は、アプリケーション・スナップショットから、特に agent 論理データ・グループのエージェント情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 842 ページの『SNAPAGENT 管理ビュー』
- 843 ページの『SNAP\_GET\_AGENT 表関数』

### SNAPAGENT 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関する agent 論理データ・グループのアプリケーション・スナップショット情報を取得することができます。

SNAPAGENT 管理ビューを SNAPAGENT\_MEMORY\_POOL、SNAPAPPL、SNAPAPPL\_INFO、SNAPSTMT、および SNAPSUBSECTION 管理ビューとともに使用すると、**GET SNAPSHOT FOR APPLICATIONS ON database-alias** CLP コマンドに相当する情報が提供されます。ただし、すべてのデータベース・メンバーからデータを取得します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、845 ページの表 203 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPAGENT 管理ビューに対する SELECT 特権
- SNAPAGENT 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_AGENT 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL

- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続中のデータベースに関するすべてのアプリケーション・スナップショット情報を、agent 論理データ・グループから取得します。

```
SELECT * FROM SYSIBMADM.SNAPAGENT
```

以下はこの照会の出力例です。

SNAPSHOT_TIMESTAMP	DB_NAME	AGENT_ID	...
2005-07-19-11.03.26.740423	SAMPLE	101	...
2005-07-19-11.03.26.740423	SAMPLE	49	...

2 record(s) selected.

この照会からの出力 (続き)。

AGENT_PID	LOCK_TIMEOUT_VAL	DBPARTITIONNUM
11980	-1	0
15940	-1	0

## SNAP\_GET\_AGENT 表関数

SNAP\_GET\_AGENT 表関数は SNAPAGENT 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_AGENT 表関数を SNAP\_GET\_AGENT\_MEMORY\_POOL、SNAP\_GET\_APPL、SNAP\_GET\_APPL\_INFO、SNAP\_GET\_STMT、および SNAP\_GET\_SUBSECTION 表関数とともに使用すると、**GET SNAPSHOT FOR ALL APPLICATIONS CLP** コマンドに相当する情報が提供されます。ただし、すべてのデータベース・メンバーからデータを取得します。

戻される可能性のある情報の完全なリストは、845 ページの表 203 を参照してください。

### 構文

```
▶▶ SNAP_GET_AGENT ( ( dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されません。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_AGENT 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_AGENT 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべてのアクティブ・データベース内のすべてのアプリケーションに関する、すべてのアプリケーション・スナップショット情報を取得します。

```
SELECT * FROM TABLE(SNAP_GET_AGENT(CAST(NULL AS VARCHAR(128)), -1)) AS T
```

以下はこの照会の出力例です。

```
SNAPSHOT_TIMESTAMP      DB_NAME      AGENT_ID      ...
-----
2006-01-03-17.21.38.530785 SAMPLE      48 ...
2006-01-03-17.21.38.530785 SAMPLE      47 ...
2006-01-03-17.21.38.530785 SAMPLE      46 ...
2006-01-03-17.21.38.530785 TESTDB      30 ...
2006-01-03-17.21.38.530785 TESTDB      29 ...
2006-01-03-17.21.38.530785 TESTDB      28 ...
```

6 record(s) selected.

この照会からの出力 (続き)。

```
... AGENT_PID      LOCK_TIMEOUT_VAL      DBPARTITIONNUM
... -----
...      7696      -1      0
...      8536      -1      0
...      6672      -1      0
...      2332      -1      0
...      8360      -1      0
...      6736      -1      0
...
```

## 戻される情報

表 225. SNAPAGENT 管理ビューおよび SNAP\_GET\_AGENT 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
AGENT_PID	BIGINT	agent_pid - エンジン・ディスパッチ可能単位 (EDU)
LOCK_TIMEOUT_VAL	BIGINT	lock_timeout_val - ロック・タイムアウト (秒)
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数 - appl 論理データ・グループのスナップショット情報の検索

SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数は、アプリケーション・スナップショットから、特に appl 論理データ・グループのアプリケーション情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 853 ページの『SNAPAPPL 管理ビュー』
- 854 ページの『SNAP\_GET\_APPL 表関数』

## SNAPAPPL 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関する appl 論理データ・グループのスナップショット情報を取得することができます。

SNAPAPPL 管理ビューを

SNAPAGENT、SNAPAGENT\_MEMORY\_POOL、SNAPAPPL\_INFO、SNAPSTMT、および SNAPSUBSECTION 管理ビューとともに使用すると、**GET SNAPSHOT FOR APPLICATIONS ON database-alias** CLP コマンドに相当する情報が提供されます。ただし、すべてのデータベース・メンバーからデータを取得します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、856 ページの表 205 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPAPPL 管理ビューに対する SELECT 特権
- SNAPAPPL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかの特権または権限が必要です。

- SNAP\_GET\_APPL 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。



## 例

接続中のデータベース内の各アプリケーションについて読み取りおよび書き込みが行われた行の詳細を取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AGENT_ID, ROWS_READ, ROWS_WRITTEN
FROM SYSIBMADM.SNAPAPPL
```

以下はこの照会の出力例です。

DB_NAME	AGENT_ID	ROWS_READ	ROWS_WRITTEN
SAMPLE		7	25

1 record(s) selected.

## SNAP\_GET\_APPL 表関数

SNAP\_GET\_APPL 表関数は SNAPAPPL 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_APPL 表関数を SNAP\_GET\_AGENT、SNAP\_GET\_AGENT\_MEMORY\_POOL、SNAP\_GET\_APPL\_INFO、SNAP\_GET\_STMT、および SNAP\_GET\_SUBSECTION 表関数とともに使用すると、**GET SNAPSHOT FOR ALL APPLICATIONS CLP** コマンドに相当する情報が提供されます。ただし、すべてのデータベース・メンバーからデータを取得します。

戻される可能性のある情報の完全なリストは、856 ページの表 205 を参照してください。

## 構文

```
▶▶—SNAP_GET_APPL—(—dbname— [ , member ] )▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。dbname を NULL

に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_APPL 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_APPL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべてのアクティブ・データベースの各アプリケーションについて読み取りおよび書き込みが行われた行の詳細を取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AGENT_ID, ROWS_READ, ROWS_WRITTEN
FROM TABLE (SNAP_GET_APPL(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	AGENT_ID	ROWS_READ	ROWS_WRITTEN
WSDB	679	0	0
WSDB	461	3	0
WSDB	460	4	0
TEST	680	4	0
TEST	455	6	0
TEST	454	0	0
TEST	453	50	0

## 戻される情報

表 226. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドドル (エージェント ID)
UOW_LOG_SPACE_USED	BIGINT	uow_log_space_used - 作業単位ログ・スペース
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
STMT_HISTORY_LIST_SIZE	BIGINT	stmt_history_list_size - ステートメント履歴リストのサイズ
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファークール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファークール・データの物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファークールへのデータの書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファークール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファークール索引の物理読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファークール索引の書き込み
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファークール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファークール一時データの物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファークール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファークール一時索引の物理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファークール一時 XDA データの論理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファークール一時 XDA データの物理読み取り : モニター・エレメント
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファークール XDA データの論理読み取り

表 226. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDA データの書き込み
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不能プリフェッチ・ページ
LOCKS_HELD	BIGINT	locks_held - ロック保持数
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間
LOCK_ESCALS	BIGINT	lock_escals - ロック・エスカレーション数
X_LOCK_ESCALS	BIGINT	x_lock_escals - 排他ロック・エスカレーション数
DEADLOCKS	BIGINT	deadlocks - デッドロック検出数
TOTAL_SORTS	BIGINT	total_sorts - ソート合計
TOTAL_SORT_TIME	BIGINT	total_sort_time - ソート時間合計
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
COMMIT_SQL_STMTS	BIGINT	commit_sql_stmts - 試行されたコミット・ステートメント
ROLLBACK_SQL_STMTS	BIGINT	rollback_sql_stmts - 試行されたロールバック・ステートメント
DYNAMIC_SQL_STMTS	BIGINT	dynamic_sql_stmts - 試行された動的 SQL ステートメント
STATIC_SQL_STMTS	BIGINT	static_sql_stmts - 試行された静的 SQL ステートメント

表 226. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
FAILED_SQL_STMTS	BIGINT	failed_sql_stmts - 失敗したステートメント操作
SELECT_SQL_STMTS	BIGINT	select_sql_stmts - 実行された選択 SQL ステートメント
DDL_SQL_STMTS	BIGINT	ddl_sql_stmts - データ定義言語 (DDL) SQL ステートメント
UID_SQL_STMTS	BIGINT	uid_sql_stmts - 実行された更新/挿入/削除 SQL ステートメント
INT_AUTO_REBINDS	BIGINT	int_auto_rebinds - 内部自動再バインド
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 削除された内部行数
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 更新された内部行数
INT_COMMITS	BIGINT	int_commits - 内部コミット数
INT_ROLLBACKS	BIGINT	int_rollback - 内部ロールバック数
INT_DEADLOCK_ROLLBACKS	BIGINT	int_deadlock_rollback - デッドロックによる内部ロールバック数
ROWS_DELETED	BIGINT	rows_deleted - 削除行数
ROWS_INSERTED	BIGINT	rows_inserted - 挿入行数
ROWS_UPDATED	BIGINT	rows_updated - 更新行数
ROWS_SELECTED	BIGINT	rows_selected - 選択行数
BINDS_PRECOMPILES	BIGINT	binds_precompiles - 試行されたバインド/プリコンパイル
OPEN_REM_CURS	BIGINT	open_rem_curs - 開かれているリモート・カーソル
OPEN_REM_CURS_BLK	BIGINT	open_rem_curs_blk - 開かれているリモート・ブロック・カーソル
REJ_CURS_BLK	BIGINT	rej_curs_blk - リジェクトされたブロック・カーソル要求
ACC_CURS_BLK	BIGINT	acc_curs_blk - 受け入れられたブロック・カーソル要求
SQL_REQS_SINCE_COMMIT	BIGINT	sql_reqs_since_commit - 最終コミット後の SQL 要求数
LOCK_TIMEOUTS	BIGINT	lock_timeouts - ロック・タイムアウト数
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 挿入された内部行数
OPEN_LOC_CURS	BIGINT	open_loc_curs - 開かれているローカル・カーソル
OPEN_LOC_CURS_BLK	BIGINT	open_loc_curs_blk - 開かれているローカル・ブロック・カーソル

表 226. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - パッケージ・キャッシュ参照
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - パッケージ・キャッシュ挿入
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - カタログ・キャッシュ参照数
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - カタログ・キャッシュ挿入数
CAT_CACHE_OVERFLOWS	BIGINT	cat_cache_overflows - カタログ・キャッシュ・オーバーフロー数
NUM_AGENTS	BIGINT	num_agents - ステートメントで作動しているエージェントの数
AGENTS_STOLEN	BIGINT	agents_stolen - スチールされたエージェント
ASSOCIATED_AGENTS_TOP	BIGINT	associated_agents_top - 関連エージェント最大数
APPL_PRIORITY	BIGINT	appl_priority - アプリケーション・エージェント優先順位
APPL_PRIORITY_TYPE	VARCHAR(16)	appl_priority_type - アプリケーション優先順位タイプ。このインターフェースは、sqlmon.h 内の定義に基づいてテキスト ID を戻します。それは、次のうちの 1 つです。 <ul style="list-style-type: none"> <li>• DYNAMIC_PRIORITY</li> <li>• FIXED_PRIORITY</li> </ul>
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間
APPL_SECTION_LOOKUPS	BIGINT	appl_section_lookups - セクションの参照回数
APPL_SECTION_INSERTS	BIGINT	appl_section_inserts - セクション挿入数
LOCKS_WAITING	BIGINT	locks_waiting - ロックで待機中の現行エージェント
TOTAL_HASH_JOINS	BIGINT	total_hash_joins - ハッシュ結合の合計
TOTAL_HASH_LOOPS	BIGINT	total_hash_loops - ハッシュ・ループの合計
HASH_JOIN_OVERFLOWS	BIGINT	hash_join_overflows - ハッシュ結合のオーバーフロー
HASH_JOIN_SMALL_OVERFLOWS	BIGINT	hash_join_small_overflows - ハッシュ結合の短精度オーバーフロー

表 226. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
APPL_IDLE_TIME	BIGINT	appl_idle_time - アプリケーション・アイドル時間
UOW_LOCK_WAIT_TIME	BIGINT	uow_lock_wait_time - ロック待機中の作業単位の合計時間
UOW_COMP_STATUS	VARCHAR(14)	uow_comp_status - 作業単位完了状況。このインターフェースは、sqlmon.h 内の定義に基づいてテキスト ID を戻します。それは、次のうちの 1 つです。 <ul style="list-style-type: none"> <li>• APPL_END</li> <li>• UOWABEND</li> <li>• UOWCOMMIT</li> <li>• UOWDEADLOCK</li> <li>• UOWLOCKTIMEOUT</li> <li>• UOWROLLBACK</li> <li>• UOWUNKNOWN</li> </ul>
AGENT_USR_CPU_TIME_S	BIGINT	agent_usr_cpu_time - エージェントが使用したユーザー CPU 時間 (秒単位)*
AGENT_USR_CPU_TIME_MS	BIGINT	agent_usr_cpu_time - エージェントが使用したユーザー CPU 時間 (小数部、マイクロ秒単位)*
AGENT_SYS_CPU_TIME_S	BIGINT	agent_sys_cpu_time - エージェントが使用したシステム CPU 時間 (秒単位)*
AGENT_SYS_CPU_TIME_MS	BIGINT	agent_sys_cpu_time - エージェントが使用したシステム CPU 時間 (小数部、マイクロ秒単位)*
APPL_CON_TIME	TIMESTAMP	appl_con_time - 接続要求開始タイム・スタンプ
CONN_COMPLETE_TIME	TIMESTAMP	conn_complete_time - 接続要求完了タイム・スタンプ
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ
UOW_START_TIME	TIMESTAMP	uow_start_time - 作業単位開始タイム・スタンプ
UOW_STOP_TIME	TIMESTAMP	uow_stop_time - 作業単位停止タイム・スタンプ
PREV_UOW_STOP_TIME	TIMESTAMP	prev_uow_stop_time - 直前の作業単位完了タイム・スタンプ
UOW_ELAPSED_TIME_S	BIGINT	uow_elapsed_time - 最新の作業単位の経過時間 (秒単位)*

表 226. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
UOW_ELAPSED_TIME_MS	BIGINT	uow_elapsed_time - 最新の作業単位の経過時間 (小数部、マイクロ秒単位)*
ELAPSED_EXEC_TIME_S	BIGINT	elapsed_exec_time - ステートメント実行経過時間 (秒単位)*
ELAPSED_EXEC_TIME_MS	BIGINT	elapsed_exec_time - ステートメント実行経過時間 (小数部、マイクロ秒単位)*
INBOUND_COMM_ADDRESS	VARCHAR(32)	inbound_comm_address - インバウンド通信アドレス
LOCK_TIMEOUT_VAL	BIGINT	lock_timeout_val - ロック・タイムアウト (秒)
PRIV_WORKSPACE_NUM_OVERFLOWS	BIGINT	priv_workspace_num_overflows - 専用ワークスペースのオーバーフロー回数
PRIV_WORKSPACE_SECTION_INSERTS	BIGINT	priv_workspace_section_inserts - 専用ワークスペース・セクション挿入
PRIV_WORKSPACE_SECTION_LOOKUPS	BIGINT	priv_workspace_section_lookups - 専用ワークスペース・セクションの参照
PRIV_WORKSPACE_SIZE_TOP	BIGINT	priv_workspace_size_top - 専用ワークスペースの最大サイズ
SHR_WORKSPACE_NUM_OVERFLOWS	BIGINT	shr_workspace_num_overflows - 共有ワークスペースのオーバーフロー回数
SHR_WORKSPACE_SECTION_INSERTS	BIGINT	shr_workspace_section_inserts - 共有ワークスペース・セクション挿入数
SHR_WORKSPACE_SECTION_LOOKUPS	BIGINT	shr_workspace_section_lookups - 共有ワークスペース・セクションの参照回数
SHR_WORKSPACE_SIZE_TOP	BIGINT	shr_workspace_size_top - 最大共有ワークスペース・サイズ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
CAT_CACHE_SIZE_TOP	BIGINT	cat_cache_size_top - カタログ・キャッシュ最高水準点
TOTAL_OLAP_FUNCS	BIGINT	total_olap_funcs - OLAP 関数の合計数
OLAP_FUNC_OVERFLOWS	BIGINT	olap_func_overflows - OLAP 関数のオーバーフロー
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント



表 226. SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
<p>* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する _S で終わっている列で報告されている整数秒と、このモニター・エレメントに関する _MS で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません: <math>(\text{monitor-element-name}_S \times 1,000,000 + \text{monitor-element-name}_{MS}) \div 1,000,000</math>. 例えば、<math>(\text{ELAPSED\_EXEC\_TIME}_S \times 1,000,000 + \text{ELAPSED\_EXEC\_TIME}_{MS}) \div 1,000,000</math>.</p>		

## SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数 - appl\_info 論理データ・グループのスナップショット情報の検索

SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数は、アプリケーション・スナップショットから、特に appl\_info 論理データ・グループのアプリケーション情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 845 ページの『SNAPAPPL\_INFO 管理ビュー』
- 846 ページの『SNAP\_GET\_APPL\_INFO 表関数』

### SNAPAPPL\_INFO 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関する appl\_info 論理データ・グループのスナップショット情報を取得することができます。

SNAPAPPL\_INFO 管理ビューを SNAPAGENT、SNAPAGENT\_MEMORY\_POOL、SNAPAPPL、SNAPSTMT、および SNAPSUBSECTION 管理ビューとともに使用すると、**GET SNAPSHOT FOR APPLICATIONS ON database-alias** CLP コマンドに相当する情報が提供されます。ただし、すべてのデータベース・メンバーからデータを取得します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、849 ページの表 204 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPAPPL\_INFO 管理ビューに対する SELECT 特権
- SNAPAPPL\_INFO 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_APPL\_INFO 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

また、以下のいずれかの権限が必要です。

- SYSMON
- SYSMOINT
- SYSCTRL
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在のデータベースに接続中のアプリケーションの状況を取得します。

```
SELECT AGENT_ID, SUBSTR(APPL_NAME,1,10) AS APPL_NAME, APPL_STATUS
FROM SYSIBMADM.SNAPAPPL_INFO
```

以下はこの照会の出力例です。

AGENT_ID	APPL_NAME	APPL_STATUS
101	db2bp.exe	UOWEXEC
49	db2bp.exe	CONNECTED

2 record(s) selected.

## SNAP\_GET\_APPL\_INFO 表関数

SNAP\_GET\_APPL\_INFO 表関数は SNAPAPPL\_INFO 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_APPL\_INFO 表関数を SNAP\_GET\_AGENT、SNAP\_GET\_AGENT\_MEMORY\_POOL、SNAP\_GET\_APPL、SNAP\_GET\_STMT、および SNAP\_GET\_SUBSECTION 表関数とともに使用すると、**GET SNAPSHOT FOR ALL APPLICATIONS CLP** コマンドに相当する情報が提供されます。ただし、すべてのデータベース・メンバーからデータを取得します。

戻される可能性のある情報の完全なリストは、849 ページの表 204 を参照してください。

## 構文

```
▶▶ SNAP_GET_APPL_INFO ( ( dbname ) [ , member ] ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されません。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_APPL\_INFO 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_APPL\_INFO 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

接続中のデータベース・メンバー上のすべてのアプリケーションの状況を取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AGENT_ID,  
       SUBSTR(APPL_NAME,1,10) AS APPL_NAME, APPL_STATUS  
FROM TABLE(SNAP_GET_APPL_INFO(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下はこの照会の出力例です。

```
DB_NAME  AGENT_ID          APPL_NAME  APPL_STATUS  
-----  
TOOLSDB          14 db2bp.exe  CONNECTED  
SAMPLE           15 db2bp.exe  UOWEXEC  
SAMPLE            8 javaw.exe  CONNECTED  
SAMPLE            7 db2bp.exe  UOWWAIT
```

4 record(s) selected.

以下は、表関数の結果からの SELECT の実行時に入手できる内容について示しています。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AUTHORITY_LVL  
FROM TABLE(SNAP_GET_APPL_INFO(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下はこの照会の出力例です。

```
DB_NAME  AUTHORITY_LVL  
-----  
TESTDB   SYSADM(GROUP) + DBADM(USER) + CREATETAB(USER, GROUP) +  
         BINDADD(USER, GROUP) + CONNECT(USER, GROUP) +  
         CREATE_NOT_FENC(USER) + IMPLICIT_SCHEMA(USER, GROUP) +  
         LOAD(USER) + CREATE_EXT_RT(USER) + QUIESCE_CONN(USER)  
TESTDB   SYSADM(GROUP) + DBADM(USER) + CREATETAB(USER, GROUP) +  
         BINDADD(USER, GROUP) + CONNECT(USER, GROUP) +  
         CREATE_NOT_FENC(USER) + IMPLICIT_SCHEMA(USER, GROUP) +  
         LOAD(USER) + CREATE_EXT_RT(USER) + QUIESCE_CONN(USER)  
TESTDB   SYSADM(GROUP) + DBADM(USER) + CREATETAB(USER, GROUP) +  
         BINDADD(USER, GROUP) + CONNECT(USER, GROUP) +  
         CREATE_NOT_FENC(USER) + IMPLICIT_SCHEMA(USER, GROUP) +  
         LOAD(USER) + CREATE_EXT_RT(USER) + QUIESCE_CONN(USER)
```

3 record(s) selected.

## 戻される情報

表 227. SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)

表 227. SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
APPL_STATUS	VARCHAR(22)	<p>appl_status - アプリケーション状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• BACKUP</li> <li>• COMMIT_ACT</li> <li>• COMP</li> <li>• CONNECTED</li> <li>• CONNECTPEND</li> <li>• CREATE_DB</li> <li>• DECOUPLED</li> <li>• DISCONNECTPEND</li> <li>• INTR</li> <li>• IOERROR_WAIT</li> <li>• LOAD</li> <li>• LOCKWAIT</li> <li>• QUIESCE_TABLESPACE</li> <li>• RECOMP</li> <li>• REMOTE_RQST</li> <li>• RESTART</li> <li>• RESTORE</li> <li>• ROLLBACK_ACT</li> <li>• ROLLBACK_TO_SAVEPOINT</li> <li>• TEND</li> <li>• THABRT</li> <li>• THCOMT</li> <li>• TPREP</li> <li>• UNLOAD</li> <li>• UOWEXEC</li> <li>• UOWWAIT</li> <li>• WAITFOR_REMOTE</li> </ul>
CODEPAGE_ID	BIGINT	codepage_id - アプリケーションで使用するコード・ページ ID
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 関連したエージェント数
COORD_NODE_NUM	SMALLINT	COORD_NODE_NUM は使用が推奨されないため、MEMBER で置き換えられます。

表 227. SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
AUTHORITY_LVL	VARCHAR(512)	<p>authority_bitmap - ユーザー許可レベル。</p> <p>このインターフェースは、sql.h で定義されたデータベース権限およびそれらのソースに基づくテキスト ID を戻し、その形式は次のとおりです。authority(source, ...) + authority(source, ...) + ... 権限のソースは複数でも構いません。USER、GROUP、または USER と GROUP のいずれかです。</p> <p>"authority" に使用できる値</p> <ul style="list-style-type: none"> <li>• ACCESSCTRL</li> <li>• BINDADD</li> <li>• CONNECT</li> <li>• CREATE_EXT_RT</li> <li>• CREATE_NOT_FENC</li> <li>• CREATETAB</li> <li>• DATAACCESS</li> <li>• DBADM</li> <li>• EXPLAIN</li> <li>• IMPLICIT_SCHEMA</li> <li>• LOAD</li> <li>• LIBADM</li> <li>• QUIESCE_CONN</li> <li>• SECADM</li> <li>• SQLADM</li> <li>• SYSADM</li> <li>• SYSCTRL</li> <li>• SYSMANT</li> <li>• SYSMON</li> <li>• SYSQUIESCE</li> <li>• WLMADM</li> </ul> <p>"source" に使用できる値</p> <ul style="list-style-type: none"> <li>• USER - ユーザーに付与された権限、またはそのユーザーに付与されているロールに付与された権限。</li> <li>• GROUP - ユーザーが属するグループに付与された権限、またはユーザーが属するグループに付与されるロールに付与された権限。</li> </ul>
CLIENT_PID	BIGINT	client_pid - クライアント・プロセス ID
COORD_AGENT_PID	BIGINT	coord_agent_pid - コーディネーター・エージェント

表 227. SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
STATUS_CHANGE_TIME	TIMESTAMP	status_change_time - アプリケーション状況変更時刻
CLIENT_PLATFORM	VARCHAR(12)	<p>client_platform - クライアント・オペレーティング・プラットフォーム。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。</p> <ul style="list-style-type: none"> <li>• AIX</li> <li>• AIX64</li> <li>• AS400_DRDA</li> <li>• DOS</li> <li>• DYNIX</li> <li>• HP</li> <li>• HP64</li> <li>• HPIA</li> <li>• HPIA64</li> <li>• LINUX</li> <li>• LINUX390</li> <li>• LINUXIA64</li> <li>• LINUXPPC</li> <li>• LINUXPPC64</li> <li>• LINUXX8664</li> <li>• LINUXZ64</li> <li>• MAC</li> <li>• MVS_DRDA</li> <li>• NT</li> <li>• NT64</li> <li>• OS2</li> <li>• OS390</li> <li>• SCO</li> <li>• SGI</li> <li>• SNI</li> <li>• SUN</li> <li>• SUN64</li> <li>• UNKNOWN</li> <li>• UNKNOWN_DRDA</li> <li>• VM_DRDA</li> <li>• VSE_DRDA</li> <li>• WINDOWS</li> </ul>

表 227. SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
CLIENT_PROTOCOL	VARCHAR(10)	client_protocol - クライアント通信プロトコル。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。  <ul style="list-style-type: none"> <li>• CPIC</li> <li>• LOCAL</li> <li>• NPIPE</li> <li>• TCPIP (DB2 UDB の場合)</li> <li>• TCPIP4</li> <li>• TCPIP6</li> </ul>
TERRITORY_CODE	SMALLINT	territory_code - データベース・テリトリリー・コード
APPL_NAME	VARCHAR(256)	appl_name - アプリケーション名
APPL_ID	VARCHAR(128)	appl_id - アプリケーション ID
SEQUENCE_NO	VARCHAR(4)	sequence_no - シーケンス番号
PRIMARY_AUTH_ID	VARCHAR(128)	auth_id - 許可 ID
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - セッション許可 ID
CLIENT_NNAME	VARCHAR(128)	client_nname - クライアント名モニター・エレメント
CLIENT_PRDID	VARCHAR(128)	client_prdid - クライアント製品/バージョン ID
INPUT_DB_ALIAS	VARCHAR(128)	input_db_alias - 入力データベース別名
CLIENT_DB_ALIAS	VARCHAR(128)	client_db_alias - アプリケーションで使用するデータベース別名
DB_NAME	VARCHAR(128)	db_name - データベース名
DB_PATH	VARCHAR(1024)	db_path - データベース・パス
EXECUTION_ID	VARCHAR(128)	execution_id - ユーザー・ログイン ID
CORR_TOKEN	VARCHAR(128)	corr_token - DRDA 相関トークン
TPMON_CLIENT_USERID	VARCHAR(256)	tpmon_client_userid - TP モニター・クライアント・ユーザー ID
TPMON_CLIENT_WKSTN	VARCHAR(256)	tpmon_client_wkstn - TP モニター・クライアント・ワークステーション名
TPMON_CLIENT_APP	VARCHAR(256)	tpmon_client_app - TP モニター・クライアント・アプリケーション名
TPMON_ACC_STR	VARCHAR(200)	tpmon_acc_str - TP モニター・クライアント・アカウント・アカウント・アカウンティング・ストリング
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
WORKLOAD_ID	INTEGER	workload_id - ワークロード ID : モニター・エレメント
IS_SYSTEM_APPL	SMALLINT	is_system_appl - システム・アプリケーション・モニター・エレメント



表 227. SNAPAPPL\_INFO 管理ビューおよび SNAP\_GET\_APPL\_INFO 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
COORD_MEMBER	SMALLINT	coord_member - コーディネーター・メンバー・モニター・エレメント
COORD_DBPARTITIONNUM	SMALLINT	調整データベース・パーティション番号。

## SNAPBP 管理ビューおよび SNAP\_GET\_BP 表関数 - bufferpool 論理グループのスナップショット情報の検索

SNAPBP 管理ビューおよび SNAP\_GET\_BP 表関数は、bufferpool スナップショットから、特に bufferpool 論理データ・グループのバッファ・プール情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 863 ページの『SNAPBP 管理ビュー』
- 864 ページの『SNAP\_GET\_BP 表関数』

### SNAPBP 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関する bufferpool 論理グループのスナップショット情報を取得することができます。

SNAPBP 管理ビューを SNAPBP\_PART 管理ビューとともに使用すると、**GET SNAPSHOT FOR BUFFERPOOLS ON database-alias** CLP コマンドに相当するデータが提供されます。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、866 ページの表 206 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPBP 管理ビューに対する SELECT 特権
- SNAPBP 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_BP 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続中のデータベースのすべてのバッファ・プールについて、データおよび索引の書き込みを取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME,SUBSTR(BP_NAME,1,15)
       AS BP_NAME,POOL_DATA_WRITES,POOL_INDEX_WRITES
FROM SYSIBMADM.SNAPBP
```

以下はこの照会の出力例です。

DB_NAME	BP_NAME	POOL_DATA_WRITES	POOL_INDEX_WRITES
TEST	IBMDEFAULTBP	0	0
TEST	IBMSYSTEMBP4K	0	0
TEST	IBMSYSTEMBP8K	0	0
TEST	IBMSYSTEMBP16K	0	0
TEST	IBMSYSTEMBP32K	0	0

5 record(s) selected

### SNAP\_GET\_BP 表関数

SNAP\_GET\_BP 表関数は SNAPBP 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_BP 表関数を SNAP\_GET\_BP\_PART 表関数とともに使用すると、GET SNAPSHOT FOR ALL BUFFERPOOLS CLP コマンドに相当するデータが提供されます。

戻される可能性のある情報の完全なリストは、866 ページの表 206 を参照してください。

### 構文

```
→ SNAP_GET_BP ( ( dbname [ , member ] ) ) →
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求を持つファイルが存在しない場合、SNAP\_GET\_BP 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_BP 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベース・メンバーのすべてのアクティブ・データベースのすべてのバッファ・プールについて、物理および論理読み取りの合計を取得します。

```
SELECT SUBSTR(T.DB_NAME,1,10) AS DB_NAME,
       SUBSTR(T.BP_NAME,1,20) AS BP_NAME,
       (T.POOL_DATA_L_READS+T.POOL_INDEX_L_READS) AS TOTAL_LOGICAL_READS,
       (T.POOL_DATA_P_READS+T.POOL_INDEX_P_READS) AS TOTAL_PHYSICAL_READS,
       T.DBPARTITIONNUM
FROM TABLE(SNAP_GET_BP(CAST(NULL AS VARCHAR(128)), -1)) AS T
```

以下はこの照会の出力例です。

```
DB_NAME    BP_NAME          TOTAL_LOGICAL_READS  ...
-----
SAMPLE     IBMDEFAULTBP          0 ...
TOOLSDB    IBMDEFAULTBP          0 ...
TOOLSDB    BP32K0000             0 ...
```

3 record(s) selected.

この照会からの出力 (続き)。

```
... TOTAL_PHYSICAL_READS DBPARTITIONNUM
... -----
...                0                0
...                0                0
...                0                0
```

## 戻される情報

表 228. SNAPBP 管理ビューおよび SNAP\_GET\_BP 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
BP_NAME	VARCHAR(128)	bp_name - バッファ・プール名
DB_NAME	VARCHAR(128)	db_name - データベース名
DB_PATH	VARCHAR(1024)	db_path - データベース・パス
INPUT_DB_ALIAS	VARCHAR(128)	input_db_alias - 入力データベース別名
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込み

表 228. SNAPBP 管理ビューおよび SNAP\_GET\_BP 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファークール XDA データの論理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファークール XDA データの物理読み取り
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファークール XDA データの書き込み
POOL_READ_TIME	BIGINT	pool_read_time - バッファークール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファークール物理書き込み時間の合計
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - バッファークール非同期データ読み取り
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - バッファークール非同期データ書き込み
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - バッファークール非同期索引読み取り
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - バッファークール非同期索引書き込み
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - バッファークール非同期 XDA データ読み取り
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - バッファークール非同期 XDA データ書き込み
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - バッファークール非同期読み取り時間
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - バッファークール非同期書き込み時間
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - バッファークール非同期読み取り要求
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - バッファークール非同期索引読み取り要求
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - バッファークール非同期 XDA 読み取り要求
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求

表 228. SNAPBP 管理ビューおよび SNAP\_GET\_BP 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不能プリフェッチ・ページ
FILES_CLOSED	BIGINT	files_closed - 閉じられたデータベース・ファイル
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り : モニター・エレメント
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - バッファ・プールの非ビクティム・バッファ数
PAGES_FROM_BLOCK_IOS	BIGINT	pages_from_block_ios - ブロック入出力によって読み取られたページ数の合計
PAGES_FROM_VECTORED_IOS	BIGINT	pages_from_vectored_ios - ベクトル化入出力によって読み取られたページ数の合計
VECTORED_IOS	BIGINT	vectored_ios - ベクトル化入出力要求数
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPBP\_PART 管理ビューおよび SNAP\_GET\_BP\_PART 表関数 - bufferpool\_nodeinfo 論理データ・グループのスナップショット情報の 検索

SNAPBP\_PART 管理ビューおよび SNAP\_GET\_BP\_PART 表関数は、バッファーク・  
プール・スナップショットから、特に bufferpool\_nodeinfo 論理データ・グループの  
バッファーク・プール情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセ  
クションを参照してください。

- 868 ページの『SNAPBP\_PART 管理ビュー』
- 870 ページの『SNAP\_GET\_BP\_PART 表関数』

### SNAPBP\_PART 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関する bufferpool\_nodeinfo  
論理データ・グループのスナップショット情報を取得することができます。

SNAPBP\_PART 管理ビューを SNAPBP 管理ビューとともに使用すると、**GET  
SNAPSHOT FOR BUFFERPOOLS ON database-alias** CLP コマンドに相当するデータが  
提供されます。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、872 ページの表 207 を参照してくだ  
さい。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPBP\_PART 管理ビューに対する SELECT 特権
- SNAPBP\_PART 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_BP\_PART 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれ  
かの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

SAMPLE データベースへの接続中に、すべてのバッファーク・プールのデータを取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, SUBSTR(BP_NAME,1,15) AS BP_NAME,  
       BP_CUR_BUFFSZ, BP_NEW_BUFFSZ, BP_PAGES_LEFT_TO_REMOVE, BP_TBSP_USE_COUNT  
FROM SYSIBMADM.SNAPBP_PART
```

以下はこの照会の出力例です。

DB_NAME	BP_NAME	BP_CUR_BUFFSZ	BP_NEW_BUFFSZ	...
SAMPLE	IBMDEFAULTBP	1000	1000	...
SAMPLE	IBMSYSTEMBP4K	16	16	...
SAMPLE	IBMSYSTEMBP8K	16	16	...
SAMPLE	IBMSYSTEMBP16K	16	16	...

4 record(s) selected.

この照会からの出力 (続き)。

...	BP_PAGES_LEFT_TO_REMOVE	BP_TBSP_USE_COUNT
...	0	3
...	0	0
...	0	0
...	0	0

## SNAP\_GET\_BP\_PART 表関数

SNAP\_GET\_BP\_PART 表関数は SNAPBP\_PART 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_BP\_PART 表関数を SNAP\_GET\_BP 表関数とともに使用すると、GET SNAPSHOT FOR ALL BUFFERPOOLS CLP コマンドに相当するデータが提供されます。

戻される可能性のある情報の完全なリストは、872 ページの表 207 を参照してください。

### 構文

```
▶▶ SNAP_GET_BP_PART ( (dbname) [ , member ] ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

*dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名



を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースにある、すべてのバッファー・プールのスナップショットを取得するには、NULL 値を指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されず、アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_BP\_PART 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_BP\_PART 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

SAMPLE データベースへの接続中に、すべてのアクティブ・データベースのすべてのバッファー・プールのデータを取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, SUBSTR(BP_NAME,1,15) AS BP_NAME,  
       BP_CUR_BUFFSZ, BP_NEW_BUFFSZ, BP_PAGES_LEFT_TO_REMOVE, BP_TBSP_USE_COUNT  
FROM TABLE(SNAP_GET_BP_PART(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下はこの照会の出力例です。

```

DB_NAME  BP_NAME          BP_CUR_BUFFSZ    BP_NEW_BUFFSZ    ...
-----
SAMPLE  IBMDEFAULTBP      250              250              ...
SAMPLE  IBMSYSTEMBP4K     16               16               ...
SAMPLE  IBMSYSTEMBP8K     16               16               ...
SAMPLE  IBMSYSTEMBP16K    16               16               ...
SAMPLE  IBMSYSTEMBP32K    16               16               ...
TESTDB  IBMDEFAULTBP      250              250              ...
TESTDB  IBMSYSTEMBP4K     16               16               ...
TESTDB  IBMSYSTEMBP8K     16               16               ...
TESTDB  IBMSYSTEMBP16K    16               16               ...
TESTDB  IBMSYSTEMBP32K    16               16               ...

```

...

この照会からの出力 (続き)。

```

... BP_PAGES_LEFT_TO_REMOVE BP_TBSP_USE_COUNT
... -----
...                0                3
...                0                0
...                0                0
...                0                0
...                0                0
...                0                3
...                0                0
...                0                0
...                0                0
...                0                0

```

...

## 戻される情報

表 229. SNAPBP\_PART 管理ビューおよび SNAP\_GET\_BP\_PART 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
BP_NAME	VARCHAR(128)	bp_name - バッファ・プール名
DB_NAME	VARCHAR(128)	db_name - データベース名
BP_CUR_BUFFSZ	BIGINT	bp_cur_buffsz - バッファ・プールの現行サイズ
BP_NEW_BUFFSZ	BIGINT	bp_new_buffsz - 新規バッファ・プール・サイズ
BP_PAGES_LEFT_TO_REMOVE	BIGINT	bp_pages_left_to_remove - 除去残ページ数
BP_TBSP_USE_COUNT	BIGINT	bp_tbsp_use_count - バッファ・プールにマップされている表スペースの数
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPCONTAINER 管理ビューおよび SNAP\_GET\_CONTAINER 表関数 - tablespace\_container 論理データ・グループのスナップショット情報の検索

SNAPCONTAINER 管理ビューおよび SNAP\_GET\_CONTAINER 表関数は、tablespace\_container 論理データ・グループからの表スペース・スナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 872 ページの『SNAPCONTAINER 管理ビュー』
- 874 ページの『SNAP\_GET\_CONTAINER 表関数』

### SNAPCONTAINER 管理ビュー

この管理ビューでは、現在接続されているデータベースの tablespace\_container 論理データ・グループ・スナップショット情報を検索できます。

SNAPTBSP、SNAPTBSP\_PART、SNAPTBSP\_QUIESCER、および SNAPTBSP\_RANGE 管理ビューと共に使用すると、SNAPCONTAINER 管理ビューは、**GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等のデータを戻します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、876 ページの表 208 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPCONTAINER 管理ビューに対する SELECT 特権
- SNAPCONTAINER 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_CONTAINER 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT

- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベースのすべてのデータベース・メンバーの表スペース・コンテナの詳細を検索します。

```
SELECT SNAPSHOT_TIMESTAMP, SUBSTR(TBSP_NAME, 1, 15) AS TBSP_NAME,
       TBSP_ID, SUBSTR(CONTAINER_NAME, 1, 20) AS CONTAINER_NAME,
       CONTAINER_ID, CONTAINER_TYPE, ACCESSIBLE, DBPARTITIONNUM
FROM SYSIBMADM.SNAPCONTAINER ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

```
SNAPSHOT_TIMESTAMP      TBSP_NAME      TBSP_ID  ...
-----
2006-01-08-16.49.24.639945 SYSCATSPACE    0  ...
2006-01-08-16.49.24.639945 TEMPSPACE1     1  ...
2006-01-08-16.49.24.639945 USERSPACE1     2  ...
2006-01-08-16.49.24.639945 SYSTOOLSPACE  3  ...
2006-01-08-16.49.24.640747 TEMPSPACE1     1  ...
2006-01-08-16.49.24.640747 USERSPACE1     2  ...
2006-01-08-16.49.24.639981 TEMPSPACE1     1  ...
2006-01-08-16.49.24.639981 USERSPACE1     2  ...
...
```

8 record(s) selected.

この照会からの出力 (続き)。

```
... CONTAINER_NAME      CONTAINER_ID  CONTAINER_TYPE  ...
... -----
... /home/swalkty/swalkt    0  FILE_EXTENT_TAG  ...
... /home/swalkty/swalkt    0  PATH             ...
... /home/swalkty/swalkt    0  FILE_EXTENT_TAG  ...
... /home/swalkty/swalkt    0  FILE_EXTENT_TAG  ...
... /home/swalkty/swalkt    0  PATH             ...
... /home/swalkty/swalkt    0  FILE_EXTENT_TAG  ...
... /home/swalkty/swalkt    0  PATH             ...
... /home/swalkty/swalkt    0  FILE_EXTENT_TAG  ...
```

この照会からの出力 (続き)。

```
... ACCESSIBLE DBPARTITIONNUM
... -----
...          1          0
...          1          0
...          1          0
...          1          0
...          1          1
...          1          1
...          1          2
...          1          2
```

## SNAP\_GET\_CONTAINER 表関数

SNAP\_GET\_CONTAINER 表関数は SNAPCONTAINER 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集

約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_TBSP、SNAP\_GET\_TBSP\_PART、SNAP\_GET\_TBSP\_QUIESCER、および SNAP\_GET\_TBSP\_RANGE 表関数と共に使用すると、SNAP\_GET\_CONTAINER 表関数は、**GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等のデータを戻します。

戻される可能性のある情報の完全なリストは、876 ページの表 208 を参照してください。

## 構文

```
▶▶ SNAP_GET_CONTAINER ( ( dbname [ , member ] ) )
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはデータベースがアクティブになっているすべてのメンバーから戻されます。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_CONTAINER 表関数は、現在接続中のデータベースおよびデータベース・メンバーのスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_CONTAINER 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベース・メンバー上で現在接続されているデータベースの表スペース・コンテナの詳細を検索します。

```
SELECT SNAPSHOT_TIMESTAMP, TBSP_NAME, TBSP_ID, CONTAINER_NAME,
       CONTAINER_ID, CONTAINER_TYPE, ACCESSIBLE
FROM TABLE(SNAP_GET_CONTAINER('','-1)) AS T
```

以下はこの照会の出力例です。

SNAPSHOT_TIMESTAMP	TBSP_NAME	TBSP_ID	...
2005-04-25-14.42.10.899253	SYSCATSPACE	0	...
2005-04-25-14.42.10.899253	TEMPSPACE1	1	...
2005-04-25-14.42.10.899253	USERSPACE1	2	...
2005-04-25-14.42.10.899253	SYSTOOLSPACE	3	...
2005-04-25-14.42.10.899253	MYTEMP	4	...
2005-04-25-14.42.10.899253	WHATSNEWTMPSPACE	5	...

この照会からの出力 (続き)。

...	CONTAINER_NAME	CONTAINER_ID	...
...	D:¥DB2¥NODE0000¥SQL00002¥SQLT0000.0	0	...
...	D:¥DB2¥NODE0000¥SQL00002¥SQLT0001.0	0	...
...	D:¥DB2¥NODE0000¥SQL00002¥SQLT0002.0	0	...
...	D:¥DB2¥NODE0000¥SQL00002¥SYSTOOLSPACE	0	...
...	D:¥DB2¥NODE0000¥SQL003	0	...
...	d:¥DGTsWhatsNewContainer	0	...

この照会からの出力 (続き)。

...	CONTAINER_TYPE	ACCESSIBLE
...	CONT_PATH	1
...	CONT_PATH	1
...	CONT_PATH	1
...	CONT_PATH	1
...	CONT_PATH	1
...	CONT_PATH	1

## 戻される情報

注: ファイル・システム情報を戻すためには、BUFFERPOOL データベース・マネージャーのモニター・スイッチをオンにする必要があります。

表 230. SNAPCONTAINER 管理ビューおよび SNAP\_GET\_CONTAINER 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
CONTAINER_NAME	VARCHAR(256)	container_name - コンテナ名
CONTAINER_ID	BIGINT	container_id - コンテナ ID
CONTAINER_TYPE	VARCHAR(16)	container_type - コンテナ・タイプ。これは、sqlutil.h での定義を基にしたテキスト ID です。以下のいずれかとなります。 <ul style="list-style-type: none"> <li>• DISK_EXTENT_TAG</li> <li>• DISK_PAGE_TAG</li> <li>• FILE_EXTENT_TAG</li> <li>• FILE_PAGE_TAG</li> <li>• PATH</li> </ul>
TOTAL_PAGES	BIGINT	container_total_pages - コンテナ内の合計ページ数
USABLE_PAGES	BIGINT	container_usable_pages - コンテナ内の使用可能なページ数
ACCESSIBLE	SMALLINT	container_accessible - コンテナのアクセス可能性
STRIPE_SET	BIGINT	container_stripe_set - ストライプ・セット
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
FS_ID	VARCHAR(22)	fs_id - 固有のファイル・システム識別番号
FS_TOTAL_SIZE	BIGINT	fs_total_size - ファイル・システムの合計サイズ
FS_USED_SIZE	BIGINT	fs_used_size - ファイル・システム上で使用されるスペースの量

## SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数 - dbase 論理グループからのスナップショット情報の取得

SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数は、データベース (dbase) 論理グループからのスナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 877 ページの『SNAPDB 管理ビュー』
- 879 ページの『SNAP\_GET\_DB 表関数』

## SNAPDB 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関するスナップショット情報を `dbase` 論理グループから検索できます。

SNAPDB 管理ビューを `ADMIN_GET_STORAGE_PATHS` 表関数、`MON_GET_MEMORY_POOL`、`MON_GET_TRANSACTION_LOG`、および `MON_GET_HADR` と併せて使用することにより、**GET SNAPSHOT FOR DATABASE on database-alias** CLP コマンドと同等の情報が得られます。

スキーマは `SYSIBMADM` です。

戻される情報の完全なリストは、882 ページの表 209 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPDB 管理ビューに対する `SELECT` 特権
- SNAPDB 管理ビューに対する `CONTROL` 特権
- `DATAACCESS` 権限
- `DBADM` 権限
- `SQLADM` 権限

この表関数を使用するには、以下のいずれかが必要です。

- `SNAP_GET_DB` 表関数に対する `EXECUTE` 特権
- `DATAACCESS` 権限
- `DBADM` 権限
- `SQLADM` 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- `SYSMON`
- `SYSCTRL`
- `SYSMAINT`
- `SYSADM`

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、`SELECT` 特権が `PUBLIC` に付与されます。

### 例

現在接続されているデータベースのすべてのデータベース・メンバーの状況、プラットフォーム、ロケーション、および接続時間を取り出します。

```
SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,  
       DB_LOCATION, DB_CONN_TIME, DBPARTITIONNUM  
FROM SYSIBMADM.SNAPDB ORDER BY DBPARTITIONNUM
```



以下はこの照会の出力例です。

```
DB_NAME      DB_STATUS  SERVER_PLATFORM DB_LOCATION ...
-----
TEST         ACTIVE    AIX64           LOCAL      ...
TEST         ACTIVE    AIX64           LOCAL      ...
TEST         ACTIVE    AIX64           LOCAL      ...
```

3 record(s) selected.

この照会からの出力 (続き)。

```
... DB_CONN_TIME          DBPARTITIONNUM
... -----
... 2006-01-08-16.48.30.665477      0
... 2006-01-08-16.48.34.005328      1
... 2006-01-08-16.48.34.007937      2
```

このルーチンは、コマンド行で以下を呼び出すことにより使用できます。

```
SELECT TOTAL_OLAP_FUNCS, OLAP_FUNC_OVERFLOWS, ACTIVE_OLAP_FUNCS
FROM SYSIBMADM.SNAPDB
```

```
TOTAL_OLAP_FUNCS  OLAP_FUNC_OVERFLOWS  ACTIVE_OLAP_FUNCS
-----
                      7                      2                      1
```

1 record(s) selected.

ワークロードの実行後に、ユーザーは次の照会を使用できます。

```
SELECT STATS_CACHE_SIZE, STATS_FABRICATIONS, SYNC_RUNSTATS,
ASYNC_RUNSTATS, STATS_FABRICATE_TIME, SYNC_RUNSTATS_TIME
FROM SYSIBMADM.SNAPDB
```

```
STATS_CACHE_SIZE  STATS_FABRICATIONS  SYNC_RUNSTATS  ASYNC_RUNSTATS ...
-----
                128                2                1                0 ...
```

```
... STATS_FABRICATE_TIME  SYNC_RUNSTATS_TIME
... -----
...                10                100
```

1 record(s) selected.

## SNAP\_GET\_DB 表関数

SNAP\_GET\_DB 表関数は、SNAPDB 管理ビューと同じ情報を戻します。

SNAP\_GET\_DB 表関数を ADMIN\_GET\_STORAGE\_PATHS 表関数、MON\_GET\_MEMORY\_POOL、MON\_GET\_TRANSACTION\_LOG、および MON\_GET\_HADR 表関数と併せて使用することにより、**GET SNAPSHOT FOR ALL DATABASES** CLP コマンドと同等の情報が得られます。

戻される情報の完全なリストは、882 ページの表 209 を参照してください。

## 構文

```
▶▶ SNAP_GET_DB ( ( dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DB 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DB 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: 現在接続されているデータベースのすべてのデータベース・メンバーを対象とした集約ビューとして、状況、プラットフォーム、ロケーション、および接続時間を取り出します。

```
SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,
       DB_LOCATION, DB_CONN_TIME FROM TABLE(SNAP_GET_DB('', -2)) AS T
```

以下はこの照会の出力例です。

DB_NAME	DB_STATUS	SERVER_PLATFORM	...
.....	.....	.....	...
SAMPLE	ACTIVE	AIX64	...

1 record(s) selected.

この照会からの出力 (続き)。

...	DB_LOCATION	DB_CONN_TIME
...	.....	.....
...	LOCAL	2005-07-24-22.09.22.013196

例 2: 現在接続されているデータベースが含まれる同じインスタンス内にあるすべてのアクティブ・データベースのすべてのデータベース・メンバーを対象とした集約ビューとして、状況、プラットフォーム、ロケーション、および接続時間を取り出します。

```
SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,
       DB_LOCATION, DB_CONN_TIME
FROM TABLE(SNAP_GET_DB(CAST (NULL AS VARCHAR(128)), -2)) AS T
```

以下はこの照会の出力例です。

DB_NAME	DB_STATUS	SERVER_PLATFORM	...
.....	.....	.....	...
TOOLSDB	ACTIVE	AIX64	...
SAMPLE	ACTIVE	AIX64	...

この照会からの出力 (続き)。

...	DB_LOCATION	DB_CONN_TIME
...	.....	.....
...	LOCAL	2005-07-24-22.26.54.396335
...	LOCAL	2005-07-24-22.09.22.013196

例 3: このルーチンは、データベースへの接続時にコマンド行で以下を呼び出すことにより使用できます。

```
SELECT TOTAL_OLAP_FUNCS, OLAP_FUNC_OVERFLOWS, ACTIVE_OLAP_FUNCS
FROM TABLE (SNAP_GET_DB('', 0)) AS T
```

出力は次のようになります。

TOTAL_OLAP_FUNCS	OLAP_FUNC_OVERFLOWS	ACTIVE_OLAP_FUNCS
-----	-----	-----
7	2	1

1 record(s) selected.

例 4: ワークロードの実行後に、ユーザーは次の照会を表関数とともに使用できません。

```
SELECT STATS_CACHE_SIZE, STATS_FABRICATIONS, SYNC_RUNSTATS,
ASYNC_RUNSTATS, STATS_FABRICATE_TIME, SYNC_RUNSTATS_TIME
FROM TABLE (SNAP_GET_DB('mytestdb', -1)) AS SNAPDB
```

```
STATS_CACHE_SIZE STATS_FABRICATIONS SYNC_RUNSTATS ASYNC_RUNSTATS ...
-----
                200                1                2                0 ...
```

Continued

```
...STATS_FABRICATE_TIME SYNC_RUNSTATS_TIME
-----
...                2                32
```

1 record(s) selected.

例 5: 以下の例は、SNAP\_GET\_DB 表関数を使用してデータベースの状況を判別する方法を示しています。

```
SELECT SUBSTR
(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS
FROM table(SNAP_GET_DB('hadrdbr', 0))
```

```
DB_NAME                DB_STATUS
-----
HADRDB                ACTIVE_STANDBY
```

## 戻される情報

表 231. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
DB_PATH	VARCHAR(1024)	db_path - データベース・パス
INPUT_DB_ALIAS	VARCHAR(128)	input_db_alias - 入力データベース別名
DB_STATUS	VARCHAR(16)	db_status - データベース状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>ACTIVE</li> <li>QUIESCE_PEND</li> <li>QUIESCED</li> <li>ROLLFWD</li> <li>ACTIVE_STANDBY - HADR データベースはスタンバイ・モードで、スタンバイ中の読み取りが有効になっています。</li> <li>STANDBY - HADR データベースはスタンバイ・モードです (スタンバイ中の読み取りは有効になっていません)。</li> </ul>
CATALOG_PARTITION	SMALLINT	catalog_node - カタログ・データベース・パーティション番号
CATALOG_PARTITION_NAME	VARCHAR(128)	catalog_node_name - カタログ・データベース・パーティション・ネットワーク名

表 231. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SERVER_PLATFORM	VARCHAR(12)	<p>server_platform - サーバーのオペレーティング・システム。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• AIX</li> <li>• AIX64</li> <li>• AS400_DRDA</li> <li>• DOS</li> <li>• DYNIX</li> <li>• HP</li> <li>• HP64</li> <li>• HPIA</li> <li>• HPIA64</li> <li>• LINUX</li> <li>• LINUX390</li> <li>• LINUXIA64</li> <li>• LINUXPPC</li> <li>• LINUXPPC64</li> <li>• LINUXX8664</li> <li>• LINUXZ64</li> <li>• MAC</li> <li>• MVS_DRDA</li> <li>• NT</li> <li>• NT64</li> <li>• OS2</li> <li>• OS390</li> <li>• SCO</li> <li>• SGI</li> <li>• SNI</li> <li>• SUN</li> <li>• SUN64</li> <li>• UNKNOWN</li> <li>• UNKNOWN_DRDA</li> <li>• VM_DRDA</li> <li>• VSE_DRDA</li> <li>• WINDOWS</li> </ul>

表 231. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DB_LOCATION	VARCHAR(12)	db_location - データベース・ロケーション。 このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。  <ul style="list-style-type: none"> <li>• LOCAL</li> <li>• REMOTE</li> </ul>
DB_CONN_TIME	TIMESTAMP	db_conn_time- データベース・アクティブ化タイム・スタンプ
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ
LAST_BACKUP	TIMESTAMP	last_backup - 最終バックアップ・タイム・スタンプ
CONNECTIONS_TOP	BIGINT	connections_top - 同時接続の最大数
TOTAL_CONS	BIGINT	total_cons - データベース・アクティブ化以降の接続
TOTAL_SEC_CONS	BIGINT	total_sec_cons - 2 次接続
APPLS_CUR_CONS	BIGINT	appls_cur_cons - 現在接続されているアプリケーション
APPLS_IN_DB2	BIGINT	appls_in_db2 - データベースで現在実行中のアプリケーション
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 関連したエージェント数
AGENTS_TOP	BIGINT	agents_top - 作成されたエージェントの数
COORD_AGENTS_TOP	BIGINT	coord_agents_top - コーディネーター・エージェント最大数
LOCKS_HELD	BIGINT	locks_held - ロック保持数
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間
LOCK_LIST_IN_USE	BIGINT	lock_list_in_use - 使用中のロック・リスト・メモリーの合計
DEADLOCKS	BIGINT	deadlocks - デッドロック検出数
LOCK_ESCALS	BIGINT	lock_escalations - ロック・エスカレーション数
X_LOCK_ESCALS	BIGINT	x_lock_escalations - 排他ロック・エスカレーション数
LOCKS_WAITING	BIGINT	locks_waiting - ロックで待機中の現行エージェント
LOCK_TIMEOUTS	BIGINT	lock_timeouts - ロック・タイムアウト数
NUM_INDOUBT_TRANS	BIGINT	num_indoubt_trans - 未確定トランザクション数
SORT_HEAP_ALLOCATED	BIGINT	sort_heap_allocated - 割り振られたソート・ヒープの合計
SORT_SHRHEAP_ALLOCATED	BIGINT	sort_shrheap_allocated - 現在割り振られているソート共有ヒープ

表 231. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SORT_SHRHEAP_TOP	BIGINT	sort_shrheap_top - ソート共有ヒープの最高水準点
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - ポスト共有しきい値ソート
TOTAL_SORTS	BIGINT	total_sorts - ソート合計
TOTAL_SORT_TIME	BIGINT	total_sort_time - ソート時間合計
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
ACTIVE_SORTS	BIGINT	active_sorts - アクティブ・ソート
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - バッファ・プール非同期データ読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - バッファ・プール非同期データ書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - バッファ・プール非同期索引読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込み
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - バッファ・プール非同期索引書き込み
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDAデータの物理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDAデータの論理読み取り
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDAデータの書き込み

表 231. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - バッファ・プール非同期 XDA データ読み取り
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - バッファ・プール非同期 XDA データ書き込み
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り : モニター・エレメント
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - バッファ・プール非同期読み取り時間
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - バッファ・プール非同期書き込み時間
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - バッファ・プール非同期読み取り要求
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - バッファ・プール非同期索引読み取り要求
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - バッファ・プール非同期 XDA 読み取り要求
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - バッファ・プールの非ビクティム・バッファ数
POOL_LSN_GAP_CLNS	BIGINT	pool_lsn_gap_clns - 起動されたバッファ・プール・ログ・スペース・クリーナー
POOL_DRTY_PG_STEAL_CLNS	BIGINT	pool_drty_pg_steal_clns - 起動されたバッファ・プール・ビクティム・ページ・クリーナー
POOL_DRTY_PG_THRSH_CLNS	BIGINT	pool_drty_pg_thrsh_clns - 起動されたバッファ・プールしきい値クリーナー
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不能プリフェッチ・ページ
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間



表 231. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
FILES_CLOSED	BIGINT	files_closed - 閉じられたデータベース・ファイル
ELAPSED_EXEC_TIME_S	BIGINT	elapsed_exec_time - ステートメント実行経過時間
ELAPSED_EXEC_TIME_MS	BIGINT	elapsed_exec_time - ステートメント実行経過時間
COMMIT_SQL_STMTS	BIGINT	commit_sql_stmts - 試行されたコミット・ステートメント
ROLLBACK_SQL_STMTS	BIGINT	rollback_sql_stmts - 試行されたロールバック・ステートメント
DYNAMIC_SQL_STMTS	BIGINT	dynamic_sql_stmts - 試行された動的 SQL ステートメント
STATIC_SQL_STMTS	BIGINT	static_sql_stmts - 試行された静的 SQL ステートメント
FAILED_SQL_STMTS	BIGINT	failed_sql_stmts - 失敗したステートメント操作
SELECT_SQL_STMTS	BIGINT	select_sql_stmts - 実行された選択 SQL ステートメント
UID_SQL_STMTS	BIGINT	uid_sql_stmts - 実行された更新/挿入/削除 SQL ステートメント
DDL_SQL_STMTS	BIGINT	ddl_sql_stmts - データ定義言語 (DDL) SQL ステートメント
INT_AUTO_REBINDS	BIGINT	int_auto_rebinds - 内部自動再バインド
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 削除された内部行数
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 挿入された内部行数
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 更新された内部行数
INT_COMMITS	BIGINT	int_commits - 内部コミット数
INT_ROLLBACKS	BIGINT	int_rollback - 内部ロールバック数
INT_DEADLOCK_ROLLBACKS	BIGINT	int_deadlock_rollback - デッドロックによる内部ロールバック数
ROWS_DELETED	BIGINT	rows_deleted - 削除行数
ROWS_INSERTED	BIGINT	rows_inserted - 挿入行数
ROWS_UPDATED	BIGINT	rows_updated - 更新行数
ROWS_SELECTED	BIGINT	rows_selected - 選択行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
BINDS_PRECOMPILES	BIGINT	binds_precompiles - 試行されたバインド/プリコンパイル
TOTAL_LOG_AVAILABLE	BIGINT	total_log_available - 使用可能なログ合計
TOTAL_LOG_USED	BIGINT	total_log_used - 使用されているログ・スペースの合計
SEC_LOG_USED_TOP	BIGINT	sec_log_used_top - 使用された最大 2 次ログ・スペース

表 231. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOT_LOG_USED_TOP	BIGINT	tot_log_used_top - 使用された最大合計ログ・スペース
SEC_LOGS_ALLOCATED	BIGINT	sec_logs_allocated - 現在割り振られている 2 次ログ
LOG_READS	BIGINT	log_reads - 読み取られたログ・ページの数
LOG_READ_TIME_S	BIGINT	log_read_time - ログ読み取り時間
LOG_READ_TIME_NS	BIGINT	log_read_time - ログ読み取り時間
LOG_WRITES	BIGINT	log_writes - 書き込まれたログ・ページの数
LOG_WRITE_TIME_S	BIGINT	log_write_time - ログ書き込み時間
LOG_WRITE_TIME_NS	BIGINT	log_write_time - ログ書き込み時間
NUM_LOG_WRITE_IO	BIGINT	num_log_write_io - ログ書き込み数
NUM_LOG_READ_IO	BIGINT	num_log_read_io - ログ読み取り数
NUM_LOG_PART_PAGE_IO	BIGINT	num_log_part_page_io - 部分ログ・ページ書き込み数
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - フル・ログ・バッファの回数
NUM_LOG_DATA_FOUND_IN_BUFFER	BIGINT	num_log_data_found_in_buffer - ログ・データがバッファにある回数
APPL_ID_OLDEST_XACT	BIGINT	appl_id_oldest_xact - 最も古いトランザクションを使用するアプリケーション
LOG_TO_REDO_FOR_RECOVERY	BIGINT	log_to_redo_for_recovery - リカバリーの場合に再実行されるログの量
LOG_HELD_BY_DIRTY_PAGES	BIGINT	log_held_by_dirty_pages - ダーティー・ページ別に計算されるログ・スペースの量
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - パッケージ・キャッシュ参照
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - パッケージ・キャッシュ挿入
PKG_CACHE_NUM_OVERFLOWES	BIGINT	pkg_cache_num_overflows - パッケージ・キャッシュ・オーバーフロー数
PKG_CACHE_SIZE_TOP	BIGINT	pkg_cache_size_top - パッケージ・キャッシュ最高水準点
APPL_SECTION_LOOKUPS	BIGINT	appl_section_lookups - セクションの参照回数
APPL_SECTION_INSERTS	BIGINT	appl_section_inserts - セクション挿入数
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - カタログ・キャッシュ参照数
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - カタログ・キャッシュ挿入数
CAT_CACHE_OVERFLOWES	BIGINT	cat_cache_overflows - カタログ・キャッシュ・オーバーフロー数
CAT_CACHE_SIZE_TOP	BIGINT	cat_cache_size_top - カタログ・キャッシュ最高水準点

表 231. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
PRIV_WORKSPACE_SIZE_TOP	BIGINT	priv_workspace_size_top - 専用ワークスペースの最大サイズ
PRIV_WORKSPACE_NUM_OVERFLOWS	BIGINT	priv_workspace_num_overflows - 専用ワークスペースのオーバーフロー回数
PRIV_WORKSPACE_SECTION_INSERTS	BIGINT	priv_workspace_section_inserts - 専用ワークスペース・セクション挿入
PRIV_WORKSPACE_SECTION_LOOKUPS	BIGINT	priv_workspace_section_lookups - 専用ワークスペース・セクションの参照
SHR_WORKSPACE_SIZE_TOP	BIGINT	shr_workspace_size_top - 最大共有ワークスペース・サイズ
SHR_WORKSPACE_NUM_OVERFLOWS	BIGINT	shr_workspace_num_overflows - 共有ワークスペースのオーバーフロー回数
SHR_WORKSPACE_SECTION_INSERTS	BIGINT	shr_workspace_section_inserts - 共有ワークスペース・セクション挿入数
SHR_WORKSPACE_SECTION_LOOKUPS	BIGINT	shr_workspace_section_lookups - 共有ワークスペース・セクションの参照回数
TOTAL_HASH_JOINS	BIGINT	total_hash_joins - ハッシュ結合の合計
TOTAL_HASH_LOOPS	BIGINT	total_hash_loops - ハッシュ・ループの合計
HASH_JOIN_OVERFLOWS	BIGINT	hash_join_overflows - ハッシュ結合のオーバーフロー
HASH_JOIN_SMALL_OVERFLOWS	BIGINT	hash_join_small_overflows - ハッシュ結合の短精度オーバーフロー
POST_SHRTHRESHOLD_HASH_JOINS	BIGINT	post_shrthreshold_hash_joins - ポストしきい値ハッシュ結合
ACTIVE_HASH_JOINS	BIGINT	active_hash_joins - アクティブ・ハッシュ結合
NUM_DB_STORAGE_PATHS	BIGINT	num_db_storage_paths - 自動ストレージ・パスの数
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
SMALLEST_LOG_AVAIL_NODE	INTEGER	smallest_log_avail_node - 使用可能なログ・スペースが最小のノード
TOTAL_OLAP_FUNCS	BIGINT	total_olap_funcs - OLAP 関数の合計数
OLAP_FUNC_OVERFLOWS	BIGINT	olap_func_overflows - OLAP 関数のオーバーフロー
ACTIVE_OLAP_FUNCS	BIGINT	active_olap_funcs - アクティブ OLAP 関数
STATS_CACHE_SIZE	BIGINT	stats_cache_size - 統計キャッシュのサイズ
STATS_FABRICATIONS	BIGINT	stats_fabrications - 統計作成の合計数
SYNC_RUNSTATS	BIGINT	sync_runstats - 同期 RUNSTATS アクティビティの合計数
ASYNC_RUNSTATS	BIGINT	async_runstats - 非同期 RUNSTATS 要求の合計数
STATS_FABRICATE_TIME	BIGINT	stats_fabricate_time - 統計作成アクティビティに費やされた合計時間

表 231. SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SYNC_RUNSTATS_TIME	BIGINT	sync_runstats_time - 同期 RUNSTATS アクティビティに費やされた合計時間
NUM_THRESHOLD_VIOLATIONS	BIGINT	num_threshold_violations - しきい値違反の回数
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPDBM 管理ビューおよび SNAP\_GET\_DBM 表関数 - dbm 論理グループ・スナップショット情報の検索

SNAPDBM 管理ビューおよび SNAP\_GET\_DBM 表関数は、スナップショット・モニターに、DB2 database データベース・マネージャー (dbm) 論理グループ情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 890 ページの『SNAPDBM 管理ビュー』
- 891 ページの『SNAP\_GET\_DBM 表関数』

### SNAPDBM 管理ビュー

SNAPDBM\_MEMORY\_POOL、SNAPFCM、SNAPFCM\_PART、および SNAPSWITCHES 管理ビューと共に使用すると、SNAPDBM 管理ビューは、**GET SNAPSHOT FOR DBM** コマンドと同等のデータを提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、893 ページの表 210 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPDBM 管理ビューに対する SELECT 特権
- SNAPDBM 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_DBM 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

すべてのデータベース・メンバーにおけるデータベース・マネージャーの状況と接続情報を検索します。

```
SELECT DB2_STATUS, DB2START_TIME, LAST_RESET, LOCAL_CONS, REM_CONS_IN,
       (AGENTS_CREATED_EMPTY_POOL/AGENTS_FROM_POOL) AS AGENT_USAGE,
       DBPARTITIONNUM FROM SYSIBMADM.SNAPDBM ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

DB2_STATUS	DB2START_TIME	LAST_RESET	...
ACTIVE	2006-01-06-14.59.59.059879	-	...
ACTIVE	2006-01-06-14.59.59.097605	-	...
ACTIVE	2006-01-06-14.59.59.062798	-	...

3 record(s) selected. ...

この照会からの出力 (続き)。

...	LOCAL_CONS	REM_CONS_IN	AGENT_USAGE	DBPARTITIONNUM
...	1	1	0	0
...	0	0	0	1
...	0	0	0	2

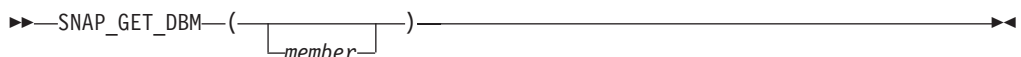
### SNAP\_GET\_DBM 表関数

SNAP\_GET\_DBM 表関数は SNAPDBM 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーを対象とした情報を検索することができます。

SNAP\_GET\_DBM\_MEMORY\_POOL、SNAP\_GET\_FCM、SNAP\_GET\_FCM\_PART、および SNAP\_GET\_SWITCHES 表関数と共に使用すると、SNAP\_GET\_DBM 表関数は、GET SNAPSHOT FOR DBM コマンドと同等のデータを提供します。

戻される可能性のある情報の完全なリストは、893 ページの表 210 を参照してください。

### 構文



スキーマは SYSPROC です。

## 表関数パラメーター

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DBM 表関数はメモリーからスナップショットを呼び出します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DBM 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

データベース・メンバー番号 2 の開始時刻と現行状況を検索します。

```
SELECT DB2START_TIME, DB2_STATUS FROM TABLE(SNAP_GET_DBM(2)) AS T
```

以下はこの照会の出力例です。

```
DB2START_TIME          DB2_STATUS
-----
2006-01-06-14.59.59.062798 ACTIVE
```

## 戻される情報

表 232. SNAPDBM 管理ビューおよび SNAP\_GET\_DBM 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
SORT_HEAP_ALLOCATED	BIGINT	sort_heap_allocated - 割り振られたソート・ヒープの合計
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - ポストしきい値ソート
PIPED_SORTS_REQUESTED	BIGINT	piped_sorts_requested - 要求されたパイプ・ソート数
PIPED_SORTS_ACCEPTED	BIGINT	piped_sorts_accepted - 受け入れられたパイプ・ソート
REM_CONS_IN	BIGINT	rem_cons_in - データベース・マネージャーへのリモート接続
REM_CONS_IN_EXEC	BIGINT	rem_cons_in_exec - データベース・マネージャーで実行中のリモート接続：モニター・エレメント
LOCAL_CONS	BIGINT	local_cons - ローカル接続
LOCAL_CONS_IN_EXEC	BIGINT	local_cons_in_exec - データベース・マネージャーで実行中のローカル接続：モニター・エレメント
CON_LOCAL_DBASES	BIGINT	con_local_dbases - 現行接続を使用したローカル・データベース
AGENTS_REGISTERED	BIGINT	agents_registered - 登録済みエージェント
AGENTS_WAITING_ON_TOKEN	BIGINT	agents_waiting_on_token - トークン待ちエージェント
DB2_STATUS	VARCHAR(12)	db2_status - DB2 インスタンス状況。  このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。  <ul style="list-style-type: none"> <li>• ACTIVE</li> <li>• QUIESCE_PEND</li> <li>• QUIESCED</li> </ul>
AGENTS_REGISTERED_TOP	BIGINT	agents_registered_top - エージェント最大登録数
AGENTS_WAITING_TOP	BIGINT	agents_waiting_top - エージェント最大待機数
COMM_PRIVATE_MEM	BIGINT	comm_private_mem - コミット済み専用メモリー
IDLE_AGENTS	BIGINT	idle_agents - アイドル・エージェント数
AGENTS_FROM_POOL	BIGINT	agents_from_pool - プールから割り当てられたエージェント
AGENTS_CREATED_EMPTY_POOL	BIGINT	agents_created_empty_pool - エージェント・プールが空のために作成されたエージェント

表 232. SNAPDBM 管理ビューおよび SNAP\_GET\_DBM 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
COORD_AGENTS_TOP	BIGINT	coord_agents_top - コーディネーター・エージェント最大数
MAX_AGENT_OVERFLOW	BIGINT	max_agent_overflows - 最大エージェント・オーバーフロー回数
AGENTS_STOLEN	BIGINT	agents_stolen - スチールされたエージェント
GW_TOTAL_CONS	BIGINT	gw_total_cons - DB2 Connect の接続試行合計回数
GW_CUR_CONS	BIGINT	gw_cur_cons - DB2 Connect の現在の接続数
GW_CONS_WAIT_HOST	BIGINT	gw_cons_wait_host - ホストの応答を待機している接続の数
GW_CONS_WAIT_CLIENT	BIGINT	gw_cons_wait_client - クライアントの要求送信を待機している接続の数
POST_THRESHOLD_HASH_JOINS	BIGINT	post_threshold_hash_joins - ハッシュ結合のしきい値
NUM_GW_CONN_SWITCHES	BIGINT	num_gw_conn_switches - 接続切り替え回数
DB2START_TIME	TIMESTAMP	db2start_time - データベース・マネージャー開始タイム・スタンプ
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ
NUM_NODES_IN_DB2_INSTANCE	INTEGER	num_nodes_in_db2_instance - データベース・パーティション内のノード数
PRODUCT_NAME	VARCHAR(32)	product_name - 製品名
SERVICE_LEVEL	VARCHAR(18)	service_level - サービス・レベル
SORT_HEAP_TOP	BIGINT	sort_heap_top - ソート専用ヒープの最高水準点
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
POST_THRESHOLD_OLAP_FUNCS	BIGINT	post_threshold_olap_funcs - OLAP 関数のしきい値
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

### SNAPDETAILLOG 管理ビューおよび SNAP\_GET\_DETAILLOG 表関数 - detail\_log 論理データ・グループからのスナップショット情報の検索

SNAPDETAILLOG 管理ビューおよび SNAP\_GET\_DETAILLOG 表関数は、detail\_log 論理データ・グループからのスナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 895 ページの『SNAPDETAILLOG 管理ビュー』
- 896 ページの『SNAP\_GET\_DETAILLOG 表関数』



## SNAPDETAILLOG 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関するスナップショット情報を detail\_log 論理データ・グループから検索できます。

SNAPDETAILLOG 管理ビューを ADMIN\_GET\_STORAGE\_PATHS、MON\_GET\_HADR、MON\_GET\_MEMORY\_POOL、および SNAPDB と併せて使用することにより、**GET SNAPSHOT FOR DATABASE on database-alias** CLP コマンドと同等の情報が得られます。

スキーマは SYSIBMADM です。

戻される情報の完全なリストは、898 ページの表 211 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPDETAILLOG 管理ビューに対する SELECT 特権
- SNAPDETAILLOG 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_DETAILLOG 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

すべてのデータベース・メンバーに関して、現在接続されているデータベースのログ情報を検索します。

```
SELECT SUBSTR(DB_NAME, 1, 8) AS DB_NAME, FIRST_ACTIVE_LOG,  
       LAST_ACTIVE_LOG, CURRENT_ACTIVE_LOG, CURRENT_ARCHIVE_LOG,  
       DBPARTITIONNUM  
FROM SYSIBMADM.SNAPDETAILLOG ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

```
DB_NAME FIRST_ACTIVE_LOG LAST_ACTIVE_LOG ...
-----
TEST          0          8 ...
TEST          0          8 ...
TEST          0          8 ...
```

3 record(s) selected.

この照会からの出力 (続き)。

```
... CURRENT_ACTIVE_LOG CURRENT_ARCHIVE_LOG DBPARTITIONNUM
... -----
...          0          -          0
...          0          -          1
...          0          -          2
```

## SNAP\_GET\_DETAILLOG 表関数

SNAP\_GET\_DETAILLOG 表関数は、SNAPDETAILLOG 管理ビューと同じ情報を戻します。

SNAPDETAILLOG 管理ビューを ADMIN\_GET\_STORAGE\_PATHS、MON\_GET\_HADR、MON\_GET\_MEMORY\_POOL、および SNAP\_GET\_DB と併せて使用することにより、GET SNAPSHOT FOR ALL DATABASES CLP コマンドと同等の情報が得られます。

戻される情報の完全なリストは、898 ページの表 211 を参照してください。

## 構文

```
▶▶ SNAP_GET_DETAILLOG ( (dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、LIST DATABASE DIRECTORY コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり dbname のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されま

す。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DETAILLOG 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DETAILLOG 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

データベース・メンバー 1 に関して、現在接続されているデータベースのログ情報を検索します。

```
SELECT SUBSTR(DB_NAME, 1, 8) AS DB_NAME, FIRST_ACTIVE_LOG,
       LAST_ACTIVE_LOG, CURRENT_ACTIVE_LOG, CURRENT_ARCHIVE_LOG
FROM TABLE(SNAP_GET_DETAILLOG('1', 1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	FIRST_ACTIVE_LOG	LAST_ACTIVE_LOG	...
TEST	0	8	...

1 record(s) selected.

この照会からの出力 (続き)。

...	CURRENT_ACTIVE_LOG	CURRENT_ARCHIVE_LOG
...	0	-

## 戻される情報

表 233. SNAPDETAILLOG 管理ビューおよび SNAP\_GET\_DETAILLOG 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
FIRST_ACTIVE_LOG	BIGINT	first_active_log - 先頭アクティブ・ログ・ファイル番号
LAST_ACTIVE_LOG	BIGINT	last_active_log - 最終アクティブ・ログ・ファイル番号
CURRENT_ACTIVE_LOG	BIGINT	current_active_log - 現行アクティブ・ログ・ファイル番号
CURRENT_ARCHIVE_LOG	BIGINT	current_archive_log - 現行アーカイブ・ログ・ファイル番号
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPDYN\_SQL 管理ビューおよび SNAP\_GET\_DYN\_SQL 表関数 - dynsql 論理グループのスナップショット情報の検索

SNAPDYN\_SQL 管理ビューおよび SNAP\_GET\_DYN\_SQL 表関数は、dynsql 論理データ・グループからのスナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 898 ページの『SNAPDYN\_SQL 管理ビュー』
- 900 ページの『SNAP\_GET\_DYN\_SQL 表関数』

### SNAPDYN\_SQL 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関する dynsql 論理グループのスナップショット情報を検索できます。

このビューは、**GET SNAPSHOT FOR DYNAMIC SQL ON database-alias CLP** コマンドと同等の情報を戻します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、903 ページの表 212 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPDYN\_SQL 管理ビューに対する SELECT 特権

- SNAPDYN\_SQL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかの権限が必要です。

- SNAP\_GET\_DYN\_SQL 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベース上で実行された動的 SQL のリストを、読み取られた行数の順序で取得します。

```
SELECT PREP_TIME_WORST, NUM_COMPILATIONS, SUBSTR(STMT_TEXT, 1, 60)
       AS STMT_TEXT, DBPARTITIONNUM
FROM SYSIBMADM.SNAPDYN_SQL ORDER BY ROWS_READ
```

以下はこの照会の出力例です。

PREP_TIME_WORST	NUM_COMPILATIONS	...
98	1	...
9	1	...
0	0	...
0	1	...
0	1	...
0	1	...
0	1	...
0	1	...
0	1	...
40	1	...

9 record(s) selected.

この照会からの出力 (続き)。

```
... STMT_TEXT
... -----
... select prep_time_worst, num_compilations, substr(stmt_text,
... select * from dbuser.employee
... SET CURRENT LOCALE LC_CTYPE = 'en_US'
... select prep_time_worst, num_compilations, substr(stmt_text, ...
```

```

... select prep_time_worst, num_compilations, substr(stmt_text, ...
... select * from dbuser.employee ...
... insert into dbuser.employee values(1) ...
... select * from dbuser.employee ...
... insert into dbuser.employee values(1) ...

```

この照会からの出力 (続き)。

```

... DBPARTITIONNUM
... -----
...          0
...          0
...          0
...          2
...          1
...          2
...          2
...          1
...          0

```

## SNAP\_GET\_DYN\_SQL 表関数

SNAP\_GET\_DYN\_SQL 表関数は SNAPDYN\_SQL 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

この表関数は、**GET SNAPSHOT FOR DYNAMIC SQL ON database-alias** CLP コマンドと同等の情報を戻します。

戻される可能性のある情報の完全なリストは、903 ページの表 212 を参照してください。

## 構文

```

▶▶ SNAP_GET_DYN_SQL ( ( dbname [ , member ] ) )

```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のメンバーの場合は -1、すべてのアクティブなメンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプション

ョンを使用しない場合、つまり *dbname* のみを指定する場合、データベースがアクティブなメンバーからデータが戻されます。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DYN\_SQL 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DYN\_SQL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベース上で実行された動的 SQL のリストを、読み取られた行数の順序で取得します。

```
SELECT PREP_TIME_WORST, NUM_COMPILATIONS, SUBSTR(STMT_TEXT, 1, 60)
      AS STMT_TEXT FROM TABLE(SNAP_GET_DYN_SQL(' ', -1)) as T
      ORDER BY ROWS_READ
```

以下はこの照会の出力例です。

PREP_TIME_WORST	NUM_COMPILATIONS	...
0	0	...
49	1	...
0	0	...
46	1	...
0	0	...
0	0	...
0	0	...
29	1	...
0	0	...
0	0	...
10	1	...
0	0	...
4	0	...
53	0	...
0	0	...

```

        6                1 ...
       334              0 ...
        0                0 ...
        5                0 ...
       10                0 ...
       599              0 ...
       15                1 ...
        7                0 ...

```

23 record(s) selected.

この照会からの出力 (続き)。

```

... STMT_TEXT
-----
... SET :HV00017 :HI00017 = RPAD(VARCHAR(:HV00035 :HI00035 ),
... SELECT COLNAME, TYPENAME FROM SYSCAT.COLUMNS WHERE TABNAME=
... DECLARE RES CURSOR WITH RETURN TO CALLER FOR SELECT R.TEXT F
... SELECT PREP_TIME_WORST, NUM_COMPILATIONS, SUBSTR(STMT_TEXT,
... VALUES (:HV00026 :HI00026 + 1, :HV00024 :HI00024 + 1) IN
... VALUES (:HV00035 :HI00035 + 1, :HV00024 :HI00024 + 1) IN
... VALUES (1) INTO :HV00035 :HI00035
... SELECT TRIGNAME FROM SYSCAT.TRIGGERS WHERE TABNAME='POLICY'
... VALUES (:HV00024 :HI00024 +1, :HV00022 :HI00022 +1) INTO :
... VALUES (1, CARDINALITY(CAST(:HV00040 :HI00040 AS "SYSIBMADM
... CALL SYSPROC.SYSINSTALLOBJECTS('POLICY','V',' ',' '))
... SET :HV00017 :HI00017 = RPAD(VARCHAR(:HV00035 :HI00035 ),
... drop event monitor act
... SELECT TABSCHEMA, TABNAME, TYPE, STATUS, TBSpaceID, PROPERTY
... CALL SAVE_EXEC_INFO (CAST(:HV00040 :HI00040 AS "SYSIBMADM"
... SET CURRENT LOCK TIMEOUT 5
... SELECT TABNAME FROM SYSCAT.PERIODS WHERE PERIODNAME = 'SYSTE
... SELECT ARRAY_AGG(P.EXECUTABLE_ID ORDER BY M.IO_WAIT_TIME DES
... SET CURRENT ISOLATION RESET
... CALL monreport.pkgcache()
... SELECT A.SPECIFICNAME FROM SYSCAT.ROUTINES A WHERE (A.FENCED
... SELECT POLICY FROM SYSTOOLS.POLICY WHERE MED='DB2CommonMED'
... VALUES 0

```

23 record(s) selected.

ワークロードの実行後に、ユーザーは次の照会を表関数とともに使用できます。

```

SELECT STATS_FABRICATE_TIME,SYNC_RUNSTATS_TIME
FROM TABLE (SNAP_GET_DYN_SQL('mytestdb', -1))
AS SNAPDB

```

STATS_FABRICATE_TIME	SYNC_RUNSTATS_TIME
2	12
1	30

この表関数に基づくビューの場合:

```

SELECT STATS_FABRICATE_TIME,SYNC_RUNSTATS_TIME
FROM SYSIBMADM.SNAPDYN_SQL

```

STATS_FABRICATE_TIME	SYNC_RUNSTATS_TIME
5	10
3	20

2 record(s) selected.



## 戻される情報

表 234. SNAPDYN\_SQL 管理ビューおよび SNAP\_GET\_DYN\_SQL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
NUM_EXECUTIONS	BIGINT	num_executions - ステートメント実行回数
NUM_COMPILATIONS	BIGINT	num_compilations - ステートメント・コンパイル数
PREP_TIME_WORST	BIGINT	prep_time_worst - ステートメント最長準備時間
PREP_TIME_BEST	BIGINT	prep_time_best - ステートメント最短準備時間
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 削除された内部行数
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 挿入された内部行数
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 更新された内部行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
STMT_SORTS	BIGINT	stmt_sorts - ステートメント・ソート回数
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
TOTAL_SORT_TIME	BIGINT	total_sort_time - ソート時間合計
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り : モニター・エレメント

表 234. SNAPDYN\_SQL 管理ビューおよび SNAP\_GET\_DYN\_SQL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_EXEC_TIME	BIGINT	total_exec_time - ステートメント実行の経過時間 (秒単位)*
TOTAL_EXEC_TIME_MS	BIGINT	total_exec_time - ステートメント実行の経過時間 (小数部、マイクロ秒単位)*
TOTAL_USR_CPU_TIME	BIGINT	total_usr_cpu_time - ステートメントのユーザー CPU の合計 (秒単位)*
TOTAL_USR_CPU_TIME_MS	BIGINT	total_usr_cpu_time - ステートメントのユーザー CPU の合計 (小数部、マイクロ秒単位)*
TOTAL_SYS_CPU_TIME	BIGINT	total_sys_cpu_time - ステートメントのシステム CPU の合計 (秒単位)*
TOTAL_SYS_CPU_TIME_MS	BIGINT	total_sys_cpu_time - ステートメントのシステム CPU の合計 (小数部、マイクロ秒単位)*
STMT_TEXT	CLOB(2 M)	stmt_text - SQL 動的ステートメント・テキスト
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
STATS_FABRICATE_TIME	BIGINT	動的ステートメントの照会コンパイル中に表または索引のスキャンを実行しないで、システムが必要とされる統計を作成するのに費やす合計時間 (ミリ秒)。
SYNC_RUNSTATS_TIME	BIGINT	動的ステートメントの照会コンパイル中に同期 statistics-collect アクティビティーに費やされる合計時間 (ミリ秒)。
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
<p>* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する _S で終わっている列で報告されている整数秒と、このモニター・エレメントに関する _MS で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません: (monitor-element-name_S × 1,000,000 + monitor-element-name_MS) ÷ 1,000,000. 例えば、(ELAPSED_EXEC_TIME_S × 1,000,000 + ELAPSED_EXEC_TIME_MS) ÷ 1,000,000。</p>		

## SNAPFCM 管理ビューおよび SNAP\_GET\_FCM 表関数 - fcm 論理データ・グループ・スナップショット情報の検索

SNAPFCM 管理ビューおよび SNAP\_GET\_FCM 表関数は、データベース・マネージャー・スナップショットから、特に fcm 論理データ・グループの高速コミュニケーション・マネージャー情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 905 ページの『SNAPFCM 管理ビュー』
- 906 ページの『SNAP\_GET\_FCM 表関数』

## SNAPFCM 管理ビュー

SNAPDBM、SNAPDBM\_MEMORY\_POOL、SNAPFCM\_PART、および SNAPSWITCHES 管理ビューと共に使用すると、SNAPFCM 管理ビューは、**GET SNAPSHOT FOR DBM** コマンドと同等のデータを提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、907 ページの表 213 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPFCM 管理ビューに対する SELECT 特権
- SNAPFCM 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_FCM 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

すべてのメンバーにおける高速コミュニケーション・マネージャーのメッセージ・バッファについて情報を検索します。

```
SELECT BUFF_FREE, BUFF_FREE_BOTTOM, MEMBER  
FROM SYSIBMADM.SNAPFCM ORDER BY MEMBER
```

以下はこの照会の出力例です。

BUFF_FREE	BUFF_FREE_BOTTOM	MEMBER
5120	5100	0
5120	5100	1
5120	5100	2

## SNAP\_GET\_FCM 表関数

SNAP\_GET\_FCM 表関数は SNAPFCM 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーを対象とした情報を検索することができます。

SNAP\_GET\_DBM、SNAP\_GET\_DBM\_MEMORY\_POOL、SNAP\_GET\_FCM\_PART、および SNAP\_GET\_SWITCHES 表関数と共に使用すると、SNAP\_GET\_FCM 表関数は、**GET SNAPSHOT FOR DBM** コマンドと同等のデータを提供します。

戻される可能性のある情報の完全なリストは、907 ページの表 213 を参照してください。

## 構文

```

▶▶ SNAP_GET_FCM ( [ member ] )

```

スキーマは SYSPROC です。

## 表関数パラメーター

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のメンバーの場合は -1、すべてのアクティブなメンバーの集合の場合は -2 を指定します。 *dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_FCM 表関数は、現在接続中のデータベースおよびデータベース・メンバーのスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_FCM 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

データベース・メンバー 1 における高速コミュニケーション・マネージャーのメッセージ・バッファーについての情報を検索します。

```
SELECT BUFF_FREE, BUFF_FREE_BOTTOM, MEMBER
FROM TABLE(SYSPROC.SNAP_GET_FCM( 1 )) AS T
```

以下はこの照会の出力例です。

```

BUFF_FREE          BUFF_FREE_BOTTOM    MEMBER
-----
                5120                5100        1

```

### 戻される情報

表 235. SNAPFCM 管理ビューおよび SNAP\_GET\_FCM 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
BUFF_FREE	BIGINT	buff_free - 現在空いている FCM バッファー
BUFF_FREE_BOTTOM	BIGINT	buff_free_bottom - 空き FCM バッファーの最小数
CH_FREE	BIGINT	ch_free - 現在空いているチャンネル
CH_FREE_BOTTOM	BIGINT	ch_free_bottom - 空いているチャンネルの最小
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPFCM\_PART 管理ビューおよび SNAP\_GET\_FCM\_PART 表関数 - fcm\_node 論理データ・グループ・スナップショット情報の検索

SNAPFCM\_PART 管理ビューおよび SNAP\_GET\_FCM\_PART 表関数は、データベース・マネージャー・スナップショットから、特に fcm\_node 論理データ・グループの高速コミュニケーション・マネージャー情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 908 ページの『SNAPFCM\_PART 管理ビュー』
- 909 ページの『SNAP\_GET\_FCM\_PART 表関数』

### SNAPFCM\_PART 管理ビュー

SNAPDBM、SNAPDBM\_MEMORY\_POOL、SNAPFCM、および SNAPSWITCHES 管理ビューと共に使用すると、SNAPFCM\_PART 管理ビューは、**GET SNAPSHOT FOR DBM** コマンドと同等のデータを提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、910 ページの表 214 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPFCM\_PART 管理ビューに対する SELECT 特権
- SNAPFCM\_PART 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_FCM\_PART 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

高速コミュニケーション・マネージャーのバッファ送受信情報を検索します。

```
SELECT CONNECTION_STATUS, TOTAL_BUFFERS_SENT, TOTAL_BUFFERS_RECEIVED
FROM SYSIBADM.SNAPFCM_PART WHERE MEMBER = 0
```

以下はこの照会の出力例です。

CONNECTION_STATUS	TOTAL_BUFFERS_SENT	TOTAL_BUFFERS_RCVD
INACTIVE	2	1

1 record(s) selected.

## SNAP\_GET\_FCM\_PART 表関数

SNAP\_GET\_FCM\_PART 表関数は SNAPFCM\_PART 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーを対象とした情報を検索することができます。

SNAP\_GET\_DBM、SNAP\_GET\_DBM\_MEMORY\_POOL、SNAP\_GET\_FCM、および SNAP\_GET\_SWITCHES 表関数と共に使用すると、SNAP\_GET\_FCM\_PART 表関数は、GET SNAPSHOT FOR DBM コマンドと同等のデータを提供します。

戻される可能性のある情報の完全なリストは、910 ページの表 214 を参照してください。

### 構文

```
▶▶ SNAP_GET_FCM_PART ( (member) )
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のメンバーの場合は -1 を指定します。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプが含まれるファイルが存在

しない場合、SNAP\_GET\_FCM\_PART 表関数は、現在接続されているデータベースとメンバーのスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_FCM\_PART 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべてのメンバーにおける高速コミュニケーション・マネージャーのバッファ送受信情報を検索します。

```
SELECT FCM_MEMBER, TOTAL_BUFFERS_SENT, TOTAL_BUFFERS_RCVD,
       MEMBER FROM TABLE(SNAP_GET_FCM_PART()) AS T
ORDER BY MEMBER
```

以下はこの照会の出力例です。

FCM_MEMBER	TOTAL_BUFFERS_SENT	TOTAL_BUFFERS_RCVD	MEMBER
0	305	305	0
1	5647	1664	0
2	5661	1688	0
0	19	19	1
1	305	301	1
2	1688	5661	1
0	1664	5647	2
1	10	10	2
2	301	305	2

## 戻される情報

表 236. SNAP\_FCM\_PART 管理ビューおよび SNAP\_GET\_FCM\_PART 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。



表 236. SNAPFCM\_PART 管理ビューおよび SNAP\_GET\_FCM\_PART 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
CONNECTION_STATUS	VARCHAR(10)	connection_status - 接続状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• INACTIVE</li> <li>• ACTIVE</li> <li>• CONGESTED</li> </ul>
TOTAL_BUFFERS_SENT	BIGINT	total_buffers_sent - 送信された FCM バッファの合計
TOTAL_BUFFERS_RCVD	BIGINT	total_buffers_rcvd - 受信された FCM バッファの合計
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
FCM_DBPARTITIONNUM	SMALLINT	データの送信先または受信元のデータベース・パーティション番号 (TOTAL_BUFFERS_SENT および TOTAL_BUFFERS_RCVD 列ごとに)。
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
FCM_MEMBER	SMALLINT	(TOTAL_BUFFERS_SENT 列および TOTAL_BUFFERS_RCVD 列のとおり) データの送信または受信が行われたメンバー。

## SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数 - ステートメント・スナップショット情報の検索

SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数は、アプリケーション・スナップショットから SQL または XQuery ステートメントについての情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 911 ページの『SNAPSTMT 管理ビュー』
- 913 ページの『SNAP\_GET\_STMT 表関数』

### SNAPSTMT 管理ビュー

この管理ビューでは、現在接続されているデータベースのステートメント・スナップショット情報を検索できます。

SNAPAGENT、SNAPAGENT\_MEMORY\_POOL、SNAPAPPL、SNAPAPPL\_INFO、および SNAPSUBSECTION 管理ビューと共に使用すると、SNAPSTMT 管理ビューは、**GET SNAPSHOT FOR APPLICATIONS on database-alias** CLP コマンドと同等の情報提供しますが、すべてのデータベース・メンバーからデータを検索します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、915 ページの表 215 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPSTMT 管理ビューに対する SELECT 特権
- SNAPSTMT 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_STMT 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されている単一メンバー・データベース上で実行されたステートメントのために読み取り、書き込み、および操作の実行が行われた行を検索します。

```
SELECT SUBSTR(STMT_TEXT,1,30) AS STMT_TEXT, ROWS_READ, ROWS_WRITTEN,  
       STMT_OPERATION FROM SYSIBMADM.SNAPSTMT
```

以下はこの照会の出力例です。

STMT_TEXT	ROWS_READ	ROWS_WRITTEN	STMT_OPERATION
-		0	0 FETCH
-		0	0 STATIC_COMMIT

2 record(s) selected.

## SNAP\_GET\_STMT 表関数

SNAP\_GET\_STMT 表関数は SNAPSTMT 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_AGENT、SNAP\_GET\_AGENT\_MEMORY\_POOL、SNAP\_GET\_APPL、SNAP\_GET\_APPL\_INFO、および SNAP\_GET\_SUBSECTION 表関数と共に使用すると、SNAP\_GET\_STMT 表関数は、**GET SNAPSHOT FOR ALL APPLICATIONS CLP** コマンドと同等の情報を提供しますが、すべてのデータベース・パーティションからデータを検索します。

戻される可能性のある情報の完全なリストは、915 ページの表 215 を参照してください。

## 構文

```

▶▶ SNAP_GET_STMT ( (dbname [ , member ] ) )

```

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり dbname のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャーにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_STMT 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_STMT 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースの現行データベース・メンバー上で実行されたステートメントのために読み取り、書き込み、および操作の実行が行われた行を検索します。

```
SELECT SUBSTR(STMT_TEXT,1,30) AS STMT_TEXT, ROWS_READ,
       ROWS_WRITTEN, STMT_OPERATION FROM TABLE(SNAP_GET_STMT(' ', -1)) AS T
```

以下はこの照会の出力例です。

```
STMT_TEXT                                ROWS_READ    ...
-----
update t set a=3                          0 ...
SELECT SUBSTR(STMT_TEXT,1,30)             0 ...
-                                           0 ...
-                                           0 ...
update t set a=2                          9 ...
5 record(s) selected.                      ...
```

この照会からの出力 (続き)。

```
... ROWS_WRITTEN    STMT_OPERATION
... -----
...                0 EXECUTE_IMMEDIATE
...                0 FETCH
...                0 NONE
...                0 NONE
...                1 EXECUTE_IMMEDIATE
...                ...
```

## 戻される情報

表 237. SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
NUM_AGENTS	BIGINT	num_agents - ステートメントで作動しているエージェントの数
AGENTS_TOP	BIGINT	agents_top - 作成されたエージェントの数
STMT_TYPE	VARCHAR(20)	stmt_type - ステートメント・タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"><li>• DYNAMIC</li><li>• NON_STMT</li><li>• STATIC</li><li>• STMT_TYPE_UNKNOWN</li></ul>

表 237. SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
STMT_OPERATION	VARCHAR(20)	stmt_operation/operation - ステートメント操作。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• CALL</li> <li>• CLOSE</li> <li>• COMPILE</li> <li>• DESCRIBE</li> <li>• EXECUTE</li> <li>• EXECUTE_IMMEDIATE</li> <li>• FETCH</li> <li>• FREE_LOCATOR</li> <li>• GETAA</li> <li>• GETNEXTCHUNK</li> <li>• GETTA</li> <li>• NONE</li> <li>• OPEN</li> <li>• PREP_COMMIT</li> <li>• PREP_EXEC</li> <li>• PREP_OPEN</li> <li>• PREPARE</li> <li>• REBIND</li> <li>• REDIST</li> <li>• REORG</li> <li>• RUNSTATS</li> <li>• SELECT</li> <li>• SET</li> <li>• STATIC_COMMIT</li> <li>• STATIC_ROLLBACK</li> </ul>
SECTION_NUMBER	BIGINT	section_number - セクション番号
QUERY_COST_ESTIMATE	BIGINT	query_cost_estimate - 照会コストの見積もり
QUERY_CARD_ESTIMATE	BIGINT	query_card_estimate - 照会行数の見積もり
DEGREE_PARALLELISM	BIGINT	degree_parallelism - 並列処理の度合い
STMT_SORTS	BIGINT	stmt_sorts - ステートメント・ソート回数
TOTAL_SORT_TIME	BIGINT	total_sort_time - ソート時間合計

表 237. SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 削除された内部行数
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 更新された内部行数
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 挿入された内部行数
FETCH_COUNT	BIGINT	fetch_count - 成功した取り出しの数
STMT_START	TIMESTAMP	stmt_start - ステートメント操作開始タイム・スタンプ
STMT_STOP	TIMESTAMP	stmt_stop - ステートメント操作停止タイム・スタンプ
STMT_USR_CPU_TIME_S	BIGINT	stmt_usr_cpu_time - ステートメントに使用されたユーザー CPU 時間 (秒単位)*
STMT_USR_CPU_TIME_MS	BIGINT	stmt_usr_cpu_time - ステートメントに使用されたユーザー CPU 時間 (小数部、マイクロ秒単位)*
STMT_SYS_CPU_TIME_S	BIGINT	stmt_sys_cpu_time - ステートメントが使用したシステム CPU 時間 (秒単位)*
STMT_SYS_CPU_TIME_MS	BIGINT	stmt_sys_cpu_time - ステートメントが使用したシステム CPU 時間 (小数部、マイクロ秒単位)*
STMT_ELAPSED_TIME_S	BIGINT	stmt_elapsed_time - 最新のステートメント経過時間 (秒単位)*
STMT_ELAPSED_TIME_MS	BIGINT	stmt_elapsed_time - 最新のステートメント経過時間 (小数部、マイクロ秒単位)*
BLOCKING_CURSOR	SMALLINT	blocking_cursor - ブロック・カーソル
STMT_NODE_NUMBER	SMALLINT	stmt_node_number - ステートメント・ノード
CURSOR_NAME	VARCHAR(128)	cursor_name - カーソル名
CREATOR	VARCHAR(128)	creator - アプリケーション作成者
PACKAGE_NAME	VARCHAR(128)	package_name - パッケージ名
STMT_TEXT	CLOB(16 M)	stmt_text - SQL 動的ステートメント・テキスト
CONSISTENCY_TOKEN	VARCHAR(128)	consistency_token - パッケージ整合性トークン

表 237. SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
PACKAGE_VERSION_ID	VARCHAR(128)	package_version_id - パッケージ・バージョン
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り : モニター・エレメント
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り : モニター・エレメント
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り : モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント



表 237. SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
<p>* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する _S で終わっている列で報告されている整数秒と、このモニター・エレメントに関する _MS で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません: <math>(\text{monitor-element-name}_S \times 1,000,000 + \text{monitor-element-name}_{MS}) \div 1,000,000</math>. 例えば、<math>(\text{ELAPSED\_EXEC\_TIME}_S \times 1,000,000 + \text{ELAPSED\_EXEC\_TIME}_{MS}) \div 1,000,000</math>.</p>		

## SNAPSUBSECTION 管理ビューおよび SNAP\_GET\_SUBSECTION 表関数 - subsection 論理モニター・グループ・スナップショット情報の検索

SNAPSUBSECTION 管理ビューおよび SNAP\_GET\_SUBSECTION 表関数は、アプリケーション・サブセクション情報として、subsection 論理モニター・グループの情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 919 ページの『SNAPSUBSECTION 管理ビュー』
- 920 ページの『SNAP\_GET\_SUBSECTION 表関数』

### SNAPSUBSECTION 管理ビュー

この管理ビューでは、現在接続されているデータベースの subsection 論理モニター・グループ・スナップショット情報を検索できます。

SNAPAGENT、SNAPAGENT\_MEMORY\_POOL、SNAPAPPL、SNAPAPPL\_INFO、および SNAPSTMT 管理ビューと共に使用すると、SNAPSUBSECTION 管理ビューは、**GET SNAPSHOT FOR APPLICATIONS on database-alias** CLP コマンドと同等の情報を提供しますが、すべてのデータベース・メンバーからデータを検索します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、922 ページの表 216 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPSUBSECTION 管理ビューに対する SELECT 特権
- SNAPSUBSECTION 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_SUBSECTION 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

すべてのデータベース・メンバーで実行するサブセクションの状況を取得します。

```
SELECT DB_NAME, STMT_TEXT, SS_STATUS, DBPARTITIONNUM
FROM SYSIBMADM.SNAPSUBSECTION
ORDER BY DB_NAME, SS_STATUS, DBPARTITIONNUM
```

以下はこの照会の出力例です。

DB_NAME	STMT_TEXT	SS_STATUS	DBPARTITIONNUM
SAMPLE	select * from EMPLOYEE	EXEC	0
SAMPLE	select * from EMPLOYEE	EXEC	1

### SNAP\_GET\_SUBSECTION 表関数

SNAP\_GET\_SUBSECTION 表関数は SNAPSUBSECTION 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

戻される可能性のある情報の完全なリストは、922 ページの表 216 を参照してください。

SNAP\_GET\_AGENT、SNAP\_GET\_AGENT\_MEMORY\_POOL、SNAP\_GET\_APPL、SNAP\_GET\_APPL\_INFO、および SNAP\_GET\_STMT 表関数と共に使用すると、SNAP\_GET\_SUBSECTION 表関数は、**GET SNAPSHOT FOR ALL APPLICATIONS CLP** コマンドと同等の情報を提供しますが、すべてのデータベース・メンバーからデータを検索します。

### 構文

```
▶▶—SNAP_GET_SUBSECTION—(—dbname— [ , member— ] )—▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されません。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_SUBSECTION 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_SUBSECTION 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべてのデータベース・メンバーで実行するサブセクションの状況を取得します。

```
SELECT DB_NAME, STMT_TEXT, SS_STATUS, DBPARTITIONNUM
FROM TABLE(SYSPROC.SNAP_GET_SUBSECTION( ' ', 0 )) as T
ORDER BY DB_NAME, SS_STATUS, DBPARTITIONNUM
```

以下はこの照会の出力例です。

DB_NAME	STMT_TEXT	SS_STATUS	DBPARTITIONNUM
SAMPLE	select * from EMPLOYEE	EXEC	0
SAMPLE	select * from EMPLOYEE	EXEC	1

## 戻される情報

表 238. SNAPSUBSECTION 管理ビューおよび SNAP\_GET\_SUBSECTION 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
STMT_TEXT	CLOB(16 M)	stmt_text - SQL 動的ステートメント・テキスト
SS_EXEC_TIME	BIGINT	ss_exec_time - サブセクション実行経過時間
TQ_TOT_SEND_SPILLS	BIGINT	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
TQ_CUR_SEND_SPILLS	BIGINT	tq_cur_send_spills - オーバーフローした表キュー・バッファの現在数
TQ_MAX_SEND_SPILLS	BIGINT	tq_max_send_spills - 表キュー・バッファ・オーバーフローの最大数
TQ_ROWS_READ	BIGINT	tq_rows_read - 表キューから読み取られた行数
TQ_ROWS_WRITTEN	BIGINT	tq_rows_written - 表キューに書き込まれた行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
SS_USR_CPU_TIME_S	BIGINT	ss_usr_cpu_time - サブセクションに使用されたユーザー CPU 時間 (秒単位)*
SS_USR_CPU_TIME_MS	BIGINT	ss_usr_cpu_time - サブセクションに使用されたユーザー CPU 時間 (小数部、マイクロ秒単位)*
SS_SYS_CPU_TIME_S	BIGINT	ss_sys_cpu_time - サブセクションに使用されたシステム CPU 時間 (秒単位)*
SS_SYS_CPU_TIME_MS	BIGINT	ss_sys_cpu_time - サブセクションに使用されたシステム CPU 時間 (小数部、マイクロ秒単位)*

表 238. SNAPSUBSECTION 管理ビューおよび SNAP\_GET\_SUBSECTION 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SS_NUMBER	INTEGER	ss_number - サブセクション番号
SS_STATUS	VARCHAR(20)	ss_status - サブセクションの状況。 このインターフェースは、sqlmon.h での定義に基づくテキスト ID を 戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• EXEC</li> <li>• TQ_WAIT_TO_RCV</li> <li>• TQ_WAIT_TO_SEND</li> <li>• COMPLETED</li> </ul>
SS_NODE_NUMBER	SMALLINT	ss_node_number - サブセクション・ノード番号
TQ_NODE_WAITED_FOR	SMALLINT	tq_node_waited_for - 表キュー上のノード待機
TQ_WAIT_FOR_ANY	INTEGER	tq_wait_for_any - 表キュー上のノード送信待機
TQ_ID_WAITING_ON	INTEGER	tq_id_waiting_on - ノード上の表キュー待機
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
<p>* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する <code>_S</code> で終わっている列で報告されている整数秒と、このモニター・エレメントに関する <code>_MS</code> で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません: <math>(\text{monitor-element-name}_S \times 1,000,000 + \text{monitor-element-name}_MS) \div 1,000,000</math>。例えば、<math>(\text{ELAPSED\_EXEC\_TIME}_S \times 1,000,000 + \text{ELAPSED\_EXEC\_TIME}_MS) \div 1,000,000</math>。</p>		

## SNAPSWITCHES 管理ビューおよび SNAP\_GET\_SWITCHES 表関数 - データベース・スナップショットのスイッチ状態情報の検索

SNAPSWITCHES 管理ビューおよび SNAP\_GET\_SWITCHES 表関数は、データベース・スナップショットのスイッチ状態に関する情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 924 ページの『SNAPSWITCHES 管理ビュー』
- 925 ページの『SNAP\_GET\_SWITCHES 表関数』

## SNAPSWITCHES 管理ビュー

このビューは、`GET DBM MONITOR SWITCHES CLP` コマンドと同等のデータを提供します。

スキーマは `SYSIBMADM` です。

戻される可能性のある情報の完全なリストは、926 ページの表 217 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPSWITCHES 管理ビューに対する `SELECT` 特権
- SNAPSWITCHES 管理ビューに対する `CONTROL` 特権
- `DATAACCESS` 権限
- `DBADM` 権限
- `SQLADM` 権限

この表関数を使用するには、以下のいずれかが必要です。

- `SNAP_GET_SWITCHES` 表関数に対する `EXECUTE` 特権
- `DATAACCESS` 権限
- `DBADM` 権限
- `SQLADM` 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- `SYSMON`
- `SYSCTRL`
- `SYSMAINT`
- `SYSADM`

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、`SELECT` 特権が `PUBLIC` に付与されます。

### 例

すべてのデータベース・メンバーにおける `DBM` モニターのスイッチ状態情報を検索します。

```
SELECT UOW_SW_STATE, STATEMENT_SW_STATE, TABLE_SW_STATE, BUFFPOOL_SW_STATE,  
       LOCK_SW_STATE, SORT_SW_STATE, TIMESTAMP_SW_STATE,  
       DBPARTITIONNUM FROM SYSIBMADM.SNAPSWITCHES
```

以下はこの照会の出力例です。

```
UOW_SW_STATE STATEMENT_SW_STATE TABLE_SW_STATE BUFFPOOL_SW_STATE ...  
-----  
0 0 0 0 ...
```

```

          0          0          0          0 ...
          0          0          0          0 ...
                                     ...

```

3 record selected.

この照会からの出力 (続き)。

```

... LOCK_SW_STATE SORT_SW_STATE TIMESTAMP_SW_STATE DBPARTITIONNUM
... -----
...          1          0          1          0
...          1          0          1          1
...          1          0          1          2

```

## SNAP\_GET\_SWITCHES 表関数

SNAP\_GET\_SWITCHES 表関数は SNAPSWITCHES 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーを対象とした情報を検索することができます。

この表関数は、GET DBM MONITOR SWITCHES CLP コマンドと同等のデータを提供します。

戻される可能性のある情報の完全なリストは、926 ページの表 217 を参照してください。

## 構文

```

▶▶ SNAP_GET_SWITCHES ( [member] )

```

スキーマは SYSPROC です。

## 表関数パラメーター

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャーにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_SWITCHES 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_SWITCHES 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現行データベース・メンバーにおける DBM モニターのスイッチ状態情報を検索します。

```
SELECT UOW_SW_STATE, STATEMENT_SW_STATE, TABLE_SW_STATE,  
       BUFFPOOL_SW_STATE, LOCK_SW_STATE, SORT_SW_STATE, TIMESTAMP_SW_STATE  
FROM TABLE(SNAP_GET_SWITCHES(-1)) AS T
```

以下はこの照会の出力例です。

```
UOW_SW_STATE STATEMENT_SW_STATE TABLE_SW_STATE...  
-----  
1 1 1...  
...  
1 record(s) selected. ...
```

この照会からの出力 (続き)。

```
... BUFFPOOL_SW_STATE LOCK_SW_STATE SORT_SW_STATE TIMESTAMP_SW_STATE  
... -----  
... 1 1 0 1
```

## 戻される情報

表 239. SNAPSWITCHES 管理ビューおよび SNAP\_GET\_SWITCHES 表関数によって戻される情報

列名	データ・タイプ	説明
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
UOW_SW_STATE	SMALLINT	作業単位モニター記録スイッチの状態 (0 または 1)。
UOW_SW_TIME	TIMESTAMP	作業単位モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
STATEMENT_SW_STATE	SMALLINT	SQL ステートメント・モニター記録スイッチの状態 (0 または 1)。



表 239. SNAPSWITCHES 管理ビューおよび SNAP\_GET\_SWITCHES 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
STATEMENT_SW_TIME	TIMESTAMP	SQL ステートメント・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
TABLE_SW_STATE	SMALLINT	表アクティビティ・モニター記録スイッチの状態 (0 または 1)。
TABLE_SW_TIME	TIMESTAMP	表アクティビティ・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
BUFFPOOL_SW_STATE	SMALLINT	バッファ・プール・アクティビティ・モニター記録スイッチの状態 (0 または 1)。
BUFFPOOL_SW_TIME	TIMESTAMP	バッファ・プール・アクティビティ・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
LOCK_SW_STATE	SMALLINT	ロック・モニター記録スイッチの状態 (0 または 1)。
LOCK_SW_TIME	TIMESTAMP	ロック・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
SORT_SW_STATE	SMALLINT	ソート・モニター記録スイッチの状態 (0 または 1)。
SORT_SW_TIME	TIMESTAMP	ソート・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
TIMESTAMP_SW_STATE	SMALLINT	タイム・スタンプのモニター記録スイッチの状態 (0 または 1)。
TIMESTAMP_SW_TIME	TIMESTAMP	タイム・スタンプのモニター記録スイッチがオンの場合、このスイッチがオンになった日付と時刻。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

### SNAPTAB\_REORG 管理ビューおよび SNAP\_GET\_TAB\_REORG 表関数 - 表再編成スナップショット情報の検索

SNAPTAB\_REORG 管理ビューおよび SNAP\_GET\_TAB\_REORG 表関数は、表再編成情報を戻します。

再編成された表がない場合は、0 行が戻されます。データ・パーティション表が再編成される時、各データ・パーティションに対して 1 つのレコードが戻されま

す。データ・パーティション表の特定の 1 つのデータ・パーティションだけが再編成される場合、そのパーティションのレコードだけが戻されます。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 928 ページの『SNAPTAB\_REORG 管理ビュー』
- 929 ページの『SNAP\_GET\_TAB\_REORG 表関数』

## SNAPTAB\_REORG 管理ビュー

この管理ビューでは、現在接続されているデータベースの表再編成スナップショット情報を検索できます。

SNAPTAB 管理ビューと共に使用すると、SNAPTAB\_REORG 管理ビューは、**GET SNAPSHOT FOR TABLES ON database-alias CLP** コマンドと同等のデータを提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、931 ページの表 218 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPTAB\_REORG 管理ビューに対する SELECT 特権
- SNAPTAB\_REORG 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_TAB\_REORG 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベース上のすべてのデータベース・メンバーでの再編成操作の詳細を選択します。

```
SELECT SUBSTR(TABNAME, 1, 15) AS TAB_NAME, SUBSTR(TABSCHEMA, 1, 15)
      AS TAB_SCHEMA, REORG_PHASE, SUBSTR(REORG_TYPE, 1, 20) AS REORG_TYPE,
      REORG_STATUS, REORG_COMPLETION, DBPARTITIONNUM
FROM SYSIBMADM.SNAPTAB_REORG ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

TAB_NAME	TAB_SCHEMA	REORG_PHASE	...
EMPLOYEE	DBUSER	REPLACE	...
EMPLOYEE	DBUSER	REPLACE	...
EMPLOYEE	DBUSER	REPLACE	...

3 record(s) selected.

この照会からの出力 (続き)。

...	REORG_TYPE	REORG_STATUS	REORG_COMPLETION	DBPARTITIONNUM
...	RECLAIM+OFFLINE+ALLO	COMPLETED	SUCCESS	0
...	RECLAIM+OFFLINE+ALLO	COMPLETED	SUCCESS	1
...	RECLAIM+OFFLINE+ALLO	COMPLETED	SUCCESS	2

マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング (ITC) 表からエクステントを再利用するための再編成操作に関するすべての情報を、SNAPTAB\_REORG 管理ビューから選択します。

```
db2 -v "select * from sysibmadm.snaptab_reorg"
```

TABNAME	REORG_PHASE	REORG_MAX_PHASE	REORG_TYPE
T1	RELEASE	3	RECLAIM_EXTENTS+ALLOW_WRITE

REORG_STATUS	REORG_COMPLETION	REORG_START	REORG_END
COMPLETED	SUCCESS	2008-09-24-14.35.30.734741	2008-09-24-14.35.31.460674

## SNAP\_GET\_TAB\_REORG 表関数

SNAP\_GET\_TAB\_REORG 表関数は SNAPTAB\_REORG 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_TAB 表関数と共に使用すると、SNAP\_GET\_TAB\_REORG 表関数は、**GET SNAPSHOT FOR TABLES ON database-alias** CLP コマンドと同等のデータを提供します。

戻される可能性のある情報の完全なリストは、931 ページの表 218 を参照してください。

## 構文

```
▶▶ SNAP_GET_TAB_REORG ( ( dbname [ , member ] ) )
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されません。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_TAB\_REORG 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_TAB\_REORG 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベース上のデータベース・メンバー 1 での再編成操作の詳細を選択します。

```
SELECT SUBSTR(TABNAME, 1, 15) AS TAB_NAME, SUBSTR(TABSHEMA, 1, 15)
AS TAB_SCHEMA, REORG_PHASE, SUBSTR(REORG_TYPE, 1, 20) AS REORG_TYPE,
REORG_STATUS, REORG_COMPLETION, DBPARTITIONNUM
FROM TABLE( SNAP_GET_TAB_REORG(' ', 1)) AS T
```

以下はこの照会の出力例です。

```
TAB_NAME      TAB_SCHEMA    REORG_PHASE    REORG_TYPE      ...
-----
EMPLOYEE      DBUSER        REPLACE        RECLAIM+OFFLINE+ALLO ...
1 record(s) selected.      ...
```

この照会からの出力 (続き)。

```
... REORG_STATUS REORG_COMPLETION DBPARTITIONNUM
... -----
... COMPLETED   SUCCESS                               1
...
```

SNAP\_GET\_TAB\_REORG 表関数を使用して、マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング (ITC) 表からエクステントを再利用するための再編成操作に関するすべての情報を選択します。

```
db2 -v "select * from table(snap_get_tab_reorg(' '))"
```

```
TABNAME  REORG_PHASE    REORG_MAX_PHASE  REORG_TYPE
-----
T1       RELEASE        3                 RECLAIM_EXTENTS+ALLOW_WRITE

REORG_STATUS REORG_COMPLETION REORG_START                REORG_END
-----
COMPLETED   SUCCESS          2008-09-24-14.35.30.734741 2008-09-24-14.35.31.460674
```

## 戻される情報

表 240. SNAPTAB\_REORG 管理ビューおよび SNAP\_GET\_TAB\_REORG 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TABNAME	VARCHAR (128)	table_name - 表名
TABSHEMA	VARCHAR (128)	table_schema - 表スキーマ名
PAGE_REORGS	BIGINT	page_reorgs - ページ再編成
REORG_PHASE	VARCHAR (16)	reorg_phase - 表再編成フェーズ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• BUILD</li> <li>• DICT_SAMPLE</li> <li>• INDEX_RECREATE</li> <li>• REPLACE</li> <li>• SORT</li> <li>• SCAN</li> <li>• DRAIN</li> <li>• RELEASE</li> </ul> または SORT+DICT_SAMPLE.
REORG_MAX_PHASE	INTEGER	reorg_max_phase - 表再編成の最大フェーズ数

表 240. SNAPTAB\_REORG 管理ビューおよび SNAP\_GET\_TAB\_REORG 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
REORG_CURRENT_COUNTER	BIGINT	reorg_current_counter - 表再編成の進行状況
REORG_MAX_COUNTER	BIGINT	reorg_max_counter - 表再編成の合計量

表 240. SNAPTAB\_REORG 管理ビューおよび SNAP\_GET\_TAB\_REORG 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
REORG_TYPE	VARCHAR (128)	<p>reorg_type - 表再編成の属性。このインターフェースは、以下の ID の組み合わせを '+' 記号で区切ったものを使用してテキスト ID を戻します。</p> <p>以下のいずれかが使用されます。</p> <ul style="list-style-type: none"> <li>• RECLAIM</li> <li>• RECLUSTER</li> <li>• RECLAIM_EXTS</li> </ul> <p>さらに以下のいずれかが使用されます。</p> <ul style="list-style-type: none"> <li>• +OFFLINE</li> <li>• +ONLINE</li> </ul> <p>アクセス・モードが指定されている場合、以下のいずれかが使用されます。</p> <ul style="list-style-type: none"> <li>• +ALLOW_NONE</li> <li>• +ALLOW_READ</li> <li>• +ALLOW_WRITE</li> </ul> <p>オフラインで RECLUSTER オプションが指定されている場合、以下のいずれかが使用されます。</p> <ul style="list-style-type: none"> <li>• +INDEXSCAN</li> <li>• +TABLESCAN</li> </ul> <p>オフラインの場合、以下のいずれかが使用されます。</p> <ul style="list-style-type: none"> <li>• +LONGLOB</li> <li>• +DATAONLY</li> </ul> <p>オフラインで、オプションが指定されている場合、以下の任意のものが使用されます。</p> <ul style="list-style-type: none"> <li>• +CHOOSE_TEMP</li> <li>• +KEEPDICTIONARY</li> <li>• +RESETDICTIONARY</li> </ul> <p>オンラインで、オプションが指定されている場合、以下が使用されます。</p> <ul style="list-style-type: none"> <li>• +NOTRUNCATE</li> </ul> <p>例 1: REORG TABLE TEST.EMPLOYEE が実行された場合、以下のように表示されます。</p> <pre>RECLAIM+OFFLINE+ALLOW_READ+DATAONLY +KEEPDICTIONARY</pre> <p>例 2: REORG TABLE TEST.EMPLOYEE INDEX EMPIDX INDEXSCAN が実行された場合、以下のように表示されます。</p> <pre>RECLUSTER+OFFLINE+ALLOW_READ+INDEXSCAN +DATAONLY+KEEPDICTIONARY</pre>

表 240. SNAPTAB\_REORG 管理ビューおよび SNAP\_GET\_TAB\_REORG 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
REORG_STATUS	VARCHAR (10)	reorg_status - 表再編成の状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• COMPLETED</li> <li>• PAUSED</li> <li>• STARTED</li> <li>• STOPPED</li> <li>• TRUNCATE</li> </ul>
REORG_COMPLETION	VARCHAR (10)	reorg_completion - 表再編成完了フラグ。このインターフェースは、sqlmon.h での定義を基にしてテキスト ID を戻します。以下のいずれかとなります。 <ul style="list-style-type: none"> <li>• FAIL</li> <li>• SUCCESS</li> </ul>
REORG_START	TIMESTAMP	reorg_start - 表再編成開始時刻
REORG_END	TIMESTAMP	reorg_end - 表再編成終了時刻
REORG_PHASE_START	TIMESTAMP	reorg_phase_start - 表再編成フェーズ開始時刻
REORG_INDEX_ID	BIGINT	reorg_index_id - 表の再編成に使用される索引
REORG_TBSPC_ID	BIGINT	reorg_tbspc_id - 表が再編成される表スペース
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID。非パーティション表では、このエレメントは NULL になります。
REORG_ROWSCOMPRESSED	BIGINT	reorg_rows_compressed - 圧縮行数
REORG_ROWSREJECTED	BIGINT	reorg_rows_rejected_for_compression - 圧縮がリジェクトされる行
REORG_LONG_TBSPC_ID	BIGINT	reorg_long_tbspc_id - 長いオブジェクトが再編成される表スペース
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

### SNAPTbsp 管理ビューおよび SNAP\_GET\_TBSP 表関数 - table space 論理データ・グループのスナップショット情報の検索

SNAPTbsp 管理ビューおよび SNAP\_GET\_TBSP 表関数は、table space 論理データ・グループからのスナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 935 ページの『SNAPTbsp 管理ビュー』
- 936 ページの『SNAP\_GET\_TBSP 表関数』



## SNAPTbsp 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関する table space 論理データ・グループのスナップショット情報を検索できます。

SNAPTbsp\_PART、SNAPTbsp\_QUIESCER、SNAPTbsp\_RANGE、SNAPCONTAINER 管理ビューと併せて使用することにより、SNAPTbsp 管理ビューは **GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を戻します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、937 ページの表 219 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPTbsp 管理ビューに対する SELECT 特権
- SNAPTbsp 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_TBSP 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベースの、カタログ・データベース・メンバーの表スペースのリストを検索します。

```
SELECT SUBSTR(TBSP_NAME,1,30) AS TBSP_NAME, TBSP_ID, TBSP_TYPE,  
       TBSP_CONTENT_TYPE FROM SYSIBMADM.SNAPTbsp WHERE DBPARTITIONNUM = 1
```

以下はこの照会の出力例です。

TBSP_NAME	TBSP_ID	TBSP_TYPE	TBSP_CONTENT_TYPE
TEMPSPACE1	1	SMS	SYSTEMP
USERSPACE1	2	DMS	LONG

2 record(s) selected.

## SNAP\_GET\_TBSP 表関数

SNAP\_GET\_TBSP 表関数は SNAPTBSP 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_TBSP\_PART、SNAP\_GET\_TBSP QUIESCER、SNAP\_GET\_TBSP\_RANGE、SNAP\_GET\_CONTAINER 表関数と併せて使用することにより、SNAP\_GET\_TBSP 表関数は **GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を戻します。

戻される可能性のある情報の完全なリストは、937 ページの表 219 を参照してください。

## 構文

```
▶▶ SNAP_GET_TBSP ( (dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

### member

有効なメンバー番号を指定する、タイプ INTEGER の入力引数。現在のメンバーの場合は -1 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

dbname と member の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_TBSP 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_TBSP 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースの、すべてのデータベース・メンバーの表スペースのリストを検索します。

```
SELECT SUBSTR(TBSP_NAME,1,10) AS TBSP_NAME, TBSP_ID, TBSP_TYPE,  
       TBSP_CONTENT_TYPE, DBPARTITIONNUM FROM TABLE(SNAP_GET_TBSP('')) AS T
```

以下はこの照会の出力例です。

TBSP_NAME	TBSP_ID	TBSP_TYPE	TBSP_CONTENT_TYPE	DBPARTITIONNUM
TEMPSPACE1	1	SMS	SYSTEMP	1
USERSPACE1	2	DMS	LONG	1
SYSCATSPAC	0	DMS	ANY	0
TEMPSPACE1	1	SMS	SYSTEMP	0
USERSPACE1	2	DMS	LONG	0
SYSTOOLSPA	3	DMS	LONG	0
TEMPSPACE1	1	SMS	SYSTEMP	2
USERSPACE1	2	DMS	LONG	2

8 record(s) selected.

## 戻される情報

表 241. SNAPTbsp 管理ビューおよび SNAP\_GET\_TBSP 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TBSP_ID	BIGINT	tablespace_id - 表スペース ID

表 241. SNAPTBSP 管理ビューおよび SNAP\_GET\_TBSP 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_TYPE	VARCHAR(10)	tablespace_type - 表スペース・タイプ。このインターフェースは、sqlutil.h の定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• DMS</li> <li>• SMS</li> </ul>
TBSP_CONTENT_TYPE	VARCHAR(10)	tablespace_content_type - 表スペースのコンテンツ・タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• ANY</li> <li>• LARGE</li> <li>• SYSTEMP</li> <li>• USRTEMP</li> </ul>
TBSP_PAGE_SIZE	BIGINT	tablespace_page_size - 表スペースのページ・サイズ
TBSP_EXTENT_SIZE	BIGINT	tablespace_extent_size - 表スペースのエクステント・サイズ
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 表スペースのプリフェッチ・サイズ
TBSP_CUR_POOL_ID	BIGINT	tablespace_cur_pool_id - 現在使用中のバッファ・プール
TBSP_NEXT_POOL_ID	BIGINT	tablespace_next_pool_id - 次の始動時に使用されるバッファ・プール
FS_CACHING	SMALLINT	fs_caching - ファイル・システム・キャッシング
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - バッファ・プール非同期データ読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み

表 241. SNAPTBSP 管理ビューおよび SNAP\_GET\_TBSP 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - バッファ ー・プール非同期データ書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ ー・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ ー・プール索引の物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ ー・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ ー・プール一時索引の物理読み取り
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - バッファ ー・プール非同期索引読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ ー・プール索引の書き込み
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - バッファ ー・プール非同期索引書き込み
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ ー・プール XDA データの論理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ ー・プール XDA データの物理読み取り
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ ー・プール XDA データの書き込み
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - バッファ ー・プール非同期 XDA データ読 み取り
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - バッファ ー・プール非同期 XDA データ書 き込み
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ ー・プール一時 XDA データの論 理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ ー・プール一時 XDA データの物 理読み取り : モニター・エレメン ト
POOL_READ_TIME	BIGINT	pool_read_time - バッファ ー・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ ー・プール物理書き込み時間の合計
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - バッファ ー・プール非同期読み取り時間
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - バッファ ー・プール非同期書き込み時間

表 241. SNAPTbsp 管理ビューおよび SNAP\_GET\_Tbsp 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - バッファ・プール非同期読み取り要求
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - バッファ・プール非同期索引読み取り要求
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - バッファ・プール非同期 XDA 読み取り要求
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - バッファ・プールの非ビクティム・バッファ数
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
FILES_CLOSED	BIGINT	files_closed - 閉じられたデータベース・ファイル
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不能プリフェッチ・ページ
Tbsp_REBALANCER_MODE	VARCHAR(10)	tablespace_rebalancer_mode - リバランサー・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• NO_REBAL</li> <li>• FWD_REBAL</li> <li>• REV_REBAL</li> </ul>
Tbsp_USING_AUTO_STORAGE	SMALLINT	tablespace_using_auto_storage - 自動ストレージが使用可能な表スペース
Tbsp_AUTO_RESIZE_ENABLED	SMALLINT	tablespace_auto_resize_enabled - 表スペースの自動サイズ変更可能
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPTbsp\_part 管理ビューおよび SNAP\_GET\_TBSP\_PART 表関数 - tablespace\_nodeinfo 論理データ・グループのスナップショット情報の取得

SNAPTbsp\_part 管理ビューおよび SNAP\_GET\_TBSP\_PART 表関数は、tablespace\_nodeinfo 論理データ・グループからのスナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 941 ページの『SNAPTbsp\_part 管理ビュー』
- 942 ページの『SNAP\_GET\_TBSP\_PART 表関数』

### SNAPTbsp\_part 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関する tablespace\_nodeinfo 論理データ・グループのスナップショット情報を検索することができます。

SNAPTbsp、SNAPTbsp\_quiescer、SNAPTbsp\_range、SNAPCONTAINER 管理ビューと併せて使用することにより、SNAPTbsp\_part 管理ビューは **GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を戻します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、944 ページの表 220 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPTbsp\_range 管理ビューに対する SELECT 特権
- SNAPTbsp\_range 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかの権限が必要です。

- SNAP\_GET\_TBSP\_PART 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT

- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベースのすべてのデータベース・パーティションの表スペースとその状態のリストを検索します。

```
SELECT SUBSTR(TBSP_NAME,1,30) AS TBSP_NAME, TBSP_ID,
       SUBSTR(TBSP_STATE,1,30) AS TBSP_STATE, DBPARTITIONNUM
FROM SYSIBMADM.SNAPTbsp_PART
```

以下はこの照会の出力例です。

TBSP_NAME	TBSP_ID	TBSP_STATE	DBPARTITIONNUM
SYSCATSPACE	0	NORMAL	0
TEMPSPACE1	1	NORMAL	0
USERSPACE1	2	NORMAL	0
TEMPSPACE1	1	NORMAL	1
USERSPACE1	2	NORMAL	1

5 record(s) selected.

## SNAP\_GET\_TBSP\_PART 表関数

SNAP\_GET\_TBSP\_PART 表関数は SNAPTbsp\_PART 管理ビューと同じ情報を戻します。ただし、SNAP\_GET\_TBSP\_PART 表関数の場合は、特定のデータベース・パーティション、すべてのデータベース・パーティションの集約、またはすべてのデータベース・パーティションについて、特定のデータベースに関する情報を取得できます。

表関数 SNAP\_GET\_TBSP、SNAP\_GET\_TBSP QUIESCER、SNAP\_GET\_TBSP\_RANGE、SNAP\_GET\_CONTAINER と併せて使用することにより、SNAP\_GET\_TBSP\_PART 表関数は CLP コマンド **GET SNAPSHOT FOR TABLESPACES ON database-alias** と同等の情報を戻します。

戻される可能性のある情報の完全なリストは、944 ページの表 220 を参照してください。

### 構文

```
▶▶ SNAP_GET_TBSP_PART ( ( dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

*dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home"



のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### *member*

有効なメンバー番号を指定する、タイプ INTEGER のオプション入力引数。現在のメンバーの場合は -1、すべてのアクティブなメンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプションが使用されない場合、つまり dbname だけが指定される場合は、データベースがアクティブになっているすべてのメンバーからデータが戻されます。

dbname と member の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_TBSP\_PART 表関数は、現在接続されているデータベースおよびメンバーのスナップショットを取得します。

#### 許可

- SYSMON 権限
- SNAP\_GET\_TBSP\_PART 表関数に対する EXECUTE 特権

#### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

#### 例

接続されているデータベースの接続されているデータベース・パーティションの表スペースとその状態のリストを検索します。

```
SELECT SUBSTR(TBSP_NAME,1,30) AS TBSP_NAME, TBSP_ID,  
       SUBSTR(TBSP_STATE,1,30) AS TBSP_STATE  
FROM TABLE(SNAP_GET_TBSP_PART(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下はこの照会の出力例です。

TBSP_NAME	TBSP_ID	TBSP_STATE
SYSCATSPACE		0 NORMAL
TEMPSPACE1		1 NORMAL
USERSPACE1		2 NORMAL
SYSTOOLSPACE		3 NORMAL
SYSTOOLSTMPSPACE		4 NORMAL

5 record(s) selected.

## 戻される情報

表 242. *SNAPTbsp\_Part* 管理ビューおよび *SNAP\_Get\_Tbsp\_Part* 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_NAME	VARCHAR (128)	tablespace_name - 表スペース名
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
TBSP_STATE	VARCHAR (256)	<p>tablespace_state - 表スペースの状態。このインターフェースは、<code>sqlutil.h</code> での定義に基づくテキスト ID を戻します。これは次のものを「+」符号で分離して組み合わせたものになります。</p> <ul style="list-style-type: none"> <li>• BACKUP_IN_PROGRESS</li> <li>• BACKUP_PENDING</li> <li>• DELETE_PENDING</li> <li>• DISABLE_PENDING</li> <li>• DROP_PENDING</li> <li>• LOAD_IN_PROGRESS</li> <li>• LOAD_PENDING</li> <li>• NORMAL</li> <li>• OFFLINE</li> <li>• PSTAT_CREATION</li> <li>• PSTAT_DELETION</li> <li>• QUIESCED_EXCLUSIVE</li> <li>• QUIESCED_SHARE</li> <li>• QUIESCED_UPDATE</li> <li>• REBAL_IN_PROGRESS</li> <li>• REORG_IN_PROGRESS</li> <li>• RESTORE_IN_PROGRESS</li> <li>• RESTORE_PENDING</li> <li>• ROLLFORWARD_IN_PROGRESS</li> <li>• ROLLFORWARD_PENDING</li> <li>• STORDEF_ALLOWED</li> <li>• STORDEF_CHANGED</li> <li>• STORDEF_FINAL_VERSION</li> <li>• STORDEF_PENDING</li> <li>• SUSPEND_WRITE</li> </ul>
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 表スペースのプリフェッチ・サイズ
TBSP_NUM QUIESCERS	BIGINT	tablespace_num_quiescers - 静止プログラム数

表 242. SNAPTBSP\_PART 管理ビューおよび SNAP\_GET\_TBSP\_PART 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_STATE_CHANGE_OBJECT_ID	BIGINT	tablespace_state_change_object_id - 状態変更オブジェクト ID
TBSP_STATE_CHANGE_TBSP_ID	BIGINT	tablespace_state_change_ts_id - 状態変更表スペース ID
TBSP_MIN_RECOVERY_TIME	TIMESTAMP	tablespace_min_recovery_time - ロールフォワードの最小リカバリー時間
TBSP_TOTAL_PAGES	BIGINT	tablespace_total_pages - 表スペース内の合計ページ数
TBSP_USABLE_PAGES	BIGINT	tablespace_usable_pages - 表スペース内の使用可能ページ数
TBSP_USED_PAGES	BIGINT	tablespace_used_pages - 表スペース内の使用されているページ数
TBSP_FREE_PAGES	BIGINT	tablespace_free_pages - 表スペース内のフリー・ページ数
TBSP_PENDING_FREE_PAGES	BIGINT	tablespace_pending_free_pages - 表スペース内のペンディング・フリー・ページ数
TBSP_PAGE_TOP	BIGINT	tablespace_page_top - 表スペース最高水準点
REBALANCER_MODE	VARCHAR (30)	tablespace_rebalancer_mode - リバランサー・モード。このインターフェースは、sqlmon.hでの定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• FWD_REBAL</li> <li>• NO_REBAL</li> <li>• REV_REBAL</li> <li>• FWD_REBAL_OF_2PASS</li> <li>• REV_REBAL_OF_2PASS</li> </ul>
REBALANCER_EXTENTS_REMAINING	BIGINT	tablespace_rebalancer_extents_remaining - リバランサーで処理されるエクステントの合計数
REBALANCER_EXTENTS_PROCESSED	BIGINT	tablespace_rebalancer_extents_processed - リバランサーで処理されたエクステントの数
REBALANCER_PRIORITY	BIGINT	tablespace_rebalancer_priority - 現行のリバランサー優先順位
REBALANCER_START_TIME	TIMESTAMP	tablespace_rebalancer_start_time - リバランサー開始時刻
REBALANCER_RESTART_TIME	TIMESTAMP	tablespace_rebalancer_restart_time - リバランサー再始動時刻
REBALANCER_LAST_EXTENT_MOVED	BIGINT	tablespace_rebalancer_last_extent_moved - リバランサーによって最後に移動されたエクステント
TBSP_NUM_RANGES	BIGINT	tablespace_num_ranges - 表スペース・マップ内の範囲数

表 242. *SNAPTbsp\_Part* 管理ビューおよび *SNAP\_Get\_Tbsp\_Part* 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_NUM_CONTAINERS	BIGINT	tablespace_num_containers - 表スペース内のコンテナ数
TBSP_INITIAL_SIZE	BIGINT	tablespace_initial_size - 表スペースの初期サイズ
TBSP_CURRENT_SIZE	BIGINT	tablespace_current_size - 表スペースの現行サイズ
TBSP_MAX_SIZE	BIGINT	tablespace_max_size - 表スペースの最大サイズ
TBSP_INCREASE_SIZE	BIGINT	tablespace_increase_size - バイト単位のサイズの増加
TBSP_INCREASE_SIZE_PERCENT	SMALLINT	tablespace_increase_size_percent - パーセント単位のサイズの増加
TBSP_LAST_RESIZE_TIME	TIMESTAMP	tablespace_last_resize_time - 最後にサイズ変更が正常に行われた時刻
TBSP_LAST_RESIZE_FAILED	SMALLINT	tablespace_last_resize_failed - 失敗した最後のサイズ変更
TBSP_PATHS_DROPPED	SMALLINT	表スペースが、ドロップされた 1 つ以上のストレージ・パスにあることを示します (0 - いいえ、1 - はい)
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

## SNAPTbsp\_Quiescer 管理ビューおよび SNAP\_Get\_Tbsp\_Quiescer 表関数 - quiescer 表スペース・スナップショット情報の検索

SNAPTbsp\_Quiescer 管理ビューおよび SNAP\_Get\_Tbsp\_Quiescer 表関数は、表スペース・スナップショットから、静止プログラムに関する情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 946 ページの『SNAPTbsp\_Quiescer 管理ビュー』
- 949 ページの『SNAP\_Get\_Tbsp\_Quiescer 表関数』

### SNAPTbsp\_Quiescer 管理ビュー

この管理ビューでは、現在接続されているデータベースの静止プログラム表スペース・スナップショット情報を検索できます。

SNAPTbsp、SNAPTbsp\_Part、SNAPTbsp\_Range、SNAPContainer 管理ビューと共に使用すると、SNAPTbsp\_Quiescer 管理ビューは、**GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を提供します。

スキーマは SYSIBMADM です。

戻すことができる情報の全リストについては、950 ページの表 221を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPTbsp\_QUIESCER 管理ビューに対する SELECT 特権
- SNAPTbsp\_QUIESCER 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_TBSP\_QUIESCER 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースのすべてのデータベース・メンバーの静止した表スペースの情報を検索します。

```
SELECT SUBSTR(TBSP_NAME, 1, 10) AS TBSP_NAME, QUIESCER_TS_ID,
       QUIESCER_OBJ_ID, QUIESCER_AUTH_ID, QUIESCER_AGENT_ID,
       QUIESCER_STATE, DBPARTITIONNUM
FROM SYSIBMADM.SNAPTbsp_QUIESCER ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

TBSP_NAME	QUIESCER_TS_ID	QUIESCER_OBJ_ID	QUIESCER_AUTH_ID	..
-----	-----	-----	-----	-----
USERSPACE1	2		5 SWALKTY	..
USERSPACE1	2		5 SWALKTY	..

2 record(s) selected.

この照会からの出力 (続き)。

```

... QUIESCER_AGENT_ID    QUIESCER_STATE DBPARTITIONNUM
... -----
...                0 EXCLUSIVE                0
...                65983 EXCLUSIVE            1

```

### 例: パーティション表の名前の判別

表がパーティション化されていて、静止状態に保たれている場合、表スペース ID および表 ID の値は、SYSCAT.TABLES 内のものとは異なる表記となります。これらの ID は、符号なしの短い表記となります。静止した表の名前を検索するには、QUIESCER\_TS\_ID から 65536 (最大値) を減算した表スペース ID を計算することによって、まず符号付きの短い表記を検索する必要があります。それから、この表スペース ID を使用して、静止した表を特定します。(実際の表スペース ID は、表内の各データ・パーティションの SYSCAT.DATAPARTITIONS にあります。)

```

SELECT SUBSTR(TBSP_NAME, 1, 10) AS TBSP_NAME,
       CASE WHEN QUIESCER_TS_ID = 65530
            THEN QUIESCER_TS_ID - 65536
            ELSE QUIESCER_TS_ID END as tbspaceid,
       CASE WHEN QUIESCER_TS_ID = 65530
            THEN QUIESCER_OBJ_ID - 65536
            ELSE QUIESCER_OBJ_ID END as tableid
FROM SYSIBMADM.SNAPTbsp_QUIESCER
ORDER BY DBPARTITIONNUM

```

以下はこの照会の出力例です。

```

TBSP_NAME    TBSPACEID    TABLEID
-----
TABDATA      -6            -32768
DATAMART     -6            -32765
SMALL        5             17

```

3 record(s) selected.

この照会で得られた特定の TBSPACEID および TABLEID を使用して、SYSCAT.TABLES から表スキーマおよび表名を検索します。

```

SELECT CHAR(tabschema, 10)tabschema, CHAR(tabname,15)tabname
FROM SYSCAT.TABLES
WHERE tbspaceid = -6 AND tableid in (-32768,-32765)

```

以下はこの照会の出力例です。

```

TABSCHEMA    TABNAME
-----
SAMP         ORDERS_RP
SAMP         ORDERS_DMART

```

2 record(s) selected.

```

SELECT CHAR(tabschema, 10)tabschema, CHAR(tabname,15)tabname
FROM SYSCAT.TABLES
WHERE tbspaceid = 5 AND tableid = 17

```

以下はこの照会の出力例です。

```

TABSCHEMA    TABNAME
-----
SAMP         NATION

```

1 record(s) selected.

## SNAP\_GET\_TBSP QUIESCER 表関数

SNAP\_GET\_TBSP QUIESCER 表関数は SNAPTBSP QUIESCER 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_TBSP、SNAP\_GET\_TBSP\_PART、SNAP\_GET\_TBSP\_RANGE、SNAP\_GET\_CONTAINER 表関数と共に使用すると、SNAP\_GET\_TBSP QUIESCER 表関数は、**GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を提供します。

戻ることができる情報の全リストについては、950 ページの表 221を参照してください。

### 構文

```
▶▶ SNAP_GET_TBSP QUIESCER ( ( dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_TBSP QUIESCER 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_TBSP QUIESCER 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースのデータベース・メンバー 1 の静止した表スペースの情報を検索します。

```
SELECT SUBSTR(TBSP_NAME, 1, 10) AS TBSP_NAME, QUIESCER_TS_ID,  
       QUIESCER_OBJ_ID, QUIESCER_AUTH_ID, QUIESCER_AGENT_ID,  
       QUIESCER_STATE, DBPARTITIONNUM  
FROM TABLE( SYSPROC.SNAP_GET_TBSP QUIESCER( ' ', 1)) AS T
```

以下はこの照会の出力例です。

```
TBSP_NAME  QUIESCER_TS_ID    QUIESCER_OBJ_ID    QUIESCER_AUTH_ID    ...  
-----  
USERSPACE1          2          5 SWALKTY          ...
```

1 record(s) selected.

この照会からの出力 (続き)。

```
... QUIESCER_AGENT_ID  QUIESCER_STATE DBPARTITIONNUM  
... -----  
...          65983 EXCLUSIVE          1
```

## 戻される情報

表 243. SNAPTbsp\_QUIESCER 管理ビューおよび SNAP\_GET\_TBSP\_QUIESCER 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
QUIESCER_TS_ID	BIGINT	quiescer_ts_id - 静止プログラム表スペース ID
QUIESCER_OBJ_ID	BIGINT	quiescer_obj_id - 静止プログラム・オブジェクト ID



表 243. *SNAPTbsp\_QUIESCER* 管理ビューおよび *SNAP\_GET\_TBSP\_QUIESCER* 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
QUIESCER_AUTH_ID	VARCHAR(128)	quiescer_auth_id - 静止プログラム・ユーザー許可 ID
QUIESCER_AGENT_ID	BIGINT	quiescer_agent_id - 静止プログラム・エージェント ID
QUIESCER_STATE	VARCHAR(14)	quiescer_state - 静止プログラムの状態。このインターフェースは、 <code>sqlutil.h</code> での定義を基にしてテキスト ID を戻します。以下のいずれかとなります。 <ul style="list-style-type: none"> <li>• EXCLUSIVE</li> <li>• UPDATE</li> <li>• SHARE</li> </ul>
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPTbsp\_RANGE 管理ビューおよび SNAP\_GET\_TBSP\_RANGE 表関数 - 範囲スナップショット情報の検索

*SNAPTbsp\_RANGE* 管理ビューおよび *SNAP\_GET\_TBSP\_RANGE* 表関数は、範囲スナップショットから情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 951 ページの『*SNAPTbsp\_RANGE* 管理ビュー』
- 953 ページの『*SNAP\_GET\_TBSP\_RANGE* 表関数』

### SNAPTbsp\_RANGE 管理ビュー

この管理ビューでは、現在接続されているデータベースの範囲スナップショット情報を検索できます。

*SNAPTbsp*、*SNAPTbsp\_PART*、*SNAPTbsp\_QUIESCER*、および *SNAPCONTAINER* 管理ビューと共に使用すると、*SNAPTbsp\_RANGE* 管理ビューは、**GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を提供します。

スキーマは *SYSIBMADM* です。

戻される可能性のある情報の完全なリストは、955 ページの表 222 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPTbsp\_RANGE 管理ビューに対する SELECT 特権
- SNAPTbsp\_RANGE 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_Tbsp\_RANGE 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースのすべてのデータベース・メンバーの表スペース範囲についての情報を選択します。

```
SELECT Tbsp_ID, SUBSTR(Tbsp_NAME, 1, 15) AS Tbsp_NAME, RANGE_NUMBER,
       RANGE_STRIPE_SET_NUMBER, RANGE_OFFSET, RANGE_MAX_PAGE,
       RANGE_MAX_EXTENT, RANGE_START_STRIPE, RANGE_END_STRIPE,
       RANGE_ADJUSTMENT, RANGE_NUM_CONTAINER, RANGE_CONTAINER_ID,
       DBPARTITIONNUM FROM SYSIBMADM.SNAPTbsp_RANGE
ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

Tbsp_ID	Tbsp_NAME	RANGE_NUMBER	RANGE_STRIPE_SET_NUMBER	...
0	SYSCATSPACE	0	0	...
2	USERSPACE1	0	0	...
3	SYSTOOLSPACE	0	0	...
2	USERSPACE1	0	0	...
2	USERSPACE1	0	0	...

5 record(s) selected.

この照会からの出力 (続き)。

...	RANGE_OFFSET	RANGE_MAX_PAGE	RANGE_MAX_EXTENT	...
...	0	11515	2878	...
...	0	479	14	...
...	0	251	62	...
...	0	479	14	...
...	0	479	14	...

この照会からの出力 (続き)。

...	RANGE_START_STRIPE	RANGE_END_STRIPE	RANGE_ADJUSTMENT	...
...	0	2878	0	...
...	0	14	0	...
...	0	62	0	...
...	0	14	0	...
...	0	14	0	...

この照会からの出力 (続き)。

...	RANGE_NUM_CONTAINER	RANGE_CONTAINER_ID	DBPARTITIONNUM
...	1	0	0
...	1	0	0
...	1	0	0
...	1	0	1
...	1	0	2

## SNAP\_GET\_TBSP\_RANGE 表関数

SNAP\_GET\_TBSP\_RANGE 表関数は SNAPTbsp\_RANGE 管理ビューと同じ情報を返しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_TBSP、SNAP\_GET\_TBSP\_PART、SNAP\_GET\_TBSP\_QUIESCER、および SNAP\_GET\_CONTAINER 表関数と共に使用すると、SNAP\_GET\_TBSP\_RANGE 表関数は、**GET SNAPSHOT FOR TABLESPACES ON database-alias** CLP コマンドと同等の情報を提供します。

戻される可能性のある情報の完全なリストは、955 ページの表 222 を参照してください。

### 構文

```

▶▶ SNAP_GET_TBSP_RANGE ( ( dbname [ , member ] ) )

```

スキーマは SYSPROC です。

### 表関数パラメーター

*dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレク

トリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ **INTEGER** のオプションの入力引数。現在のデータベース・メンバーの場合は **-1**、すべてアクティブなデータベース・メンバーの集合の場合は **-2** を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に **-1** に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されず、アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、**SNAP\_WRITE\_FILE** プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、**SNAP\_GET\_TBSP\_RANGE** 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

### 許可

以下のいずれかの権限が必要です。

- **SNAP\_GET\_TBSP\_RANGE** 表関数に対する **EXECUTE** 特権
- **DATAACCESS** 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- **SYSMON**
- **SYSCTRL**
- **SYSMAINT**
- **SYSADM**

### デフォルトの **PUBLIC** 特権

制限のないデータベースでは、この関数が自動的に作成されると、**EXECUTE** 特権が **PUBLIC** に付与されます。

### 例

現在接続されているデータベース・メンバー上で *tbasp\_id* = 2 である表スペースの表スペース範囲の情報を選択します。

```
SELECT TBSP_ID, SUBSTR(TBSP_NAME, 1, 15) AS TBSP_NAME, RANGE_NUMBER,
       RANGE_STRIPE_SET_NUMBER, RANGE_OFFSET, RANGE_MAX_PAGE, RANGE_MAX_EXTENT,
       RANGE_START_STRIPE, RANGE_END_STRIPE, RANGE_ADJUSTMENT,
       RANGE_NUM_CONTAINER, RANGE_CONTAINER_ID
FROM TABLE(SNAP_GET_TBSP_RANGE('',-1)) AS T WHERE TBSP_ID = 2
```

以下はこの照会の出力例です。

```

TBSP_ID      TBSP_NAME      RANGE_NUMBER      ...
-----
2 USERSPACE1      0 ...

```

1 record(s) selected.

この照会からの出力 (続き)。

```

... RANGE_STRIPE_SET_NUMBER RANGE_OFFSET      RANGE_MAX_PAGE      ...
... -----
...                0                0                3967 ...

```

この照会からの出力 (続き)。

```

... RANGE_MAX_EXTENT      RANGE_START_STRIPE      RANGE_END_STRIPE      ...
... -----
...                123                0                123 ...

```

この照会からの出力 (続き)。

```

... RANGE_ADJUSTMENT      RANGE_NUM_CONTAINER      RANGE_CONTAINER_ID
... -----
...                0                1                0

```

## 戻される情報

表 244. *SNAPTbsp\_Range* 管理ビューおよび *SNAP\_Get\_Tbsp\_Range* 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
RANGE_NUMBER	BIGINT	range_number - 範囲番号
RANGE_STRIPE_SET_NUMBER	BIGINT	range_stripe_set_number - ストライプ・セット番号
RANGE_OFFSET	BIGINT	range_offset - 範囲オフセット
RANGE_MAX_PAGE	BIGINT	range_max_page_number - 範囲内の最大ページ
RANGE_MAX_EXTENT	BIGINT	range_max_extent - 範囲内の最大エクステンツ
RANGE_START_STRIPE	BIGINT	range_start_stripe - 開始ストライプ
RANGE_END_STRIPE	BIGINT	range_end_stripe - 終了ストライプ
RANGE_ADJUSTMENT	BIGINT	range_adjustment - 範囲調整
RANGE_NUM_CONTAINER	BIGINT	range_num_containers - 範囲内コンテナの数
RANGE_CONTAINER_ID	BIGINT	range_container_id - 範囲コンテナ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

## SNAPUTIL 管理ビューおよび SNAP\_GET\_UTIL 表関数 - utility\_info 論理データ・グループ・スナップショット情報の検索

SNAPUTIL 管理ビューおよび SNAP\_GET\_UTIL 表関数は、utility\_info 論理データ・グループからのユーティリティー・スナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 956 ページの『SNAPUTIL 管理ビュー』
- 957 ページの『SNAP\_GET\_UTIL 表関数』

### SNAPUTIL 管理ビュー

SNAPUTIL\_PROGRESS 管理ビューと組み合わせて使用すると、SNAPUTIL 管理ビューは、**LIST UTILITIES SHOW DETAIL CLP** コマンドと同じ情報を提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、959 ページの表 223 を参照してください。

### 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPUTIL 管理ビューに対する SELECT 特権
- SNAPUTIL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_UTIL 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

接続されているデータベースが含まれるインスタンス内のすべてのアクティブ・データベースのすべてのデータベース・メンバー上にある、ユーティリティとその状態のリストを検索します。

```
SELECT UTILITY_TYPE, UTILITY_PRIORITY, SUBSTR(UTILITY_DESCRIPTION, 1, 72)
      AS UTILITY_DESCRIPTION, SUBSTR(UTILITY_DBNAME, 1, 17) AS
      UTILITY_DBNAME, UTILITY_STATE, UTILITY_INVOKER_TYPE, DBPARTITIONNUM
FROM SYSIBMADM.SNAPUTIL ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

```
UTILITY_TYPE      UTILITY_PRIORITY ...
-----
LOAD              - ...
LOAD              - ...
LOAD              - ...
```

3 record(s) selected.

この照会からの出力 (続き)。

```
... UTILITY_DESCRIPTION ...
... -----
... ONLINE LOAD DEL AUTOMATIC INDEXING INSERT COPY NO TEST .LOADTEST ...
... ONLINE LOAD DEL AUTOMATIC INDEXING INSERT COPY NO TEST .LOADTEST ...
... ONLINE LOAD DEL AUTOMATIC INDEXING INSERT COPY NO TEST .LOADTEST ...
```

この照会からの出力 (続き)。

```
... UTILITY_DBNAME  UTILITY_STATE UTILITY_INVOKER_TYPE DBPARTITIONNUM
... -----
... SAMPLE          EXECUTE      USER                0
... SAMPLE          EXECUTE      USER                1
... SAMPLE          EXECUTE      USER                2
```

## SNAP\_GET\_UTIL 表関数

SNAP\_GET\_UTIL 表関数は SNAPUTIL 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーを対象とした情報を検索することができます。

SNAP\_GET\_UTIL\_PROGRESS 表関数と組み合わせて使用すると、SNAP\_GET\_UTIL 表関数は、**LIST UTILITIES SHOW DETAIL CLP** コマンドと同じ情報を提供します。

戻される可能性のある情報の完全なリストは、959 ページの表 223 を参照してください。

## 構文

```
▶▶ SNAP_GET_UTIL ( [member] ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

*member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプション

ンの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_UTIL 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_UTIL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: データベース SAMPLE 上の現在接続されているデータベース・メンバーのユーティリティー ID (そのタイプと状態を含む) のリストを検索します。

```
SELECT UTILITY_ID, UTILITY_TYPE, UTILITY_STATE
FROM TABLE(SNAP_GET_UTIL(-1)) AS T WHERE UTILITY_DBNAME='SAMPLE'
```

以下はこの照会の出力例です。

UTILITY_ID	UTILITY_TYPE	STATE
-----	-----	-----
	1 BACKUP	EXECUTE

1 record(s) selected.

例 2: 現在接続されているデータベース・メンバーのユーティリティー ID とそのタイプ、メンバー番号、およびデータベース・パーティション番号のリストを検索します。

```
SELECT UTILITY_ID, UTILITY_TYPE, MEMBER, DBPARTITIONNUM
FROM TABLE(SNAP_GET_UTIL(-1)) AS T
```



以下はこの照会の出力例です。

```

UTILITY_ID  UTILITY_TYPE                MEMBER DBPARTITIONNUM
-----
          2 BACKUP                2                2
  
```

## 戻される情報

表 245. SNAPUTIL 管理ビューおよび SNAP\_GET\_UTIL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
UTILITY_ID	INTEGER	utility_id - ユーティリティー ID。データベース・パーティションに固有。
UTILITY_TYPE	VARCHAR(26)	utility_type - ユーティリティー・タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• ASYNC_INDEX_CLEANUP</li> <li>• BACKUP</li> <li>• CRASH_RECOVERY</li> <li>• LOAD</li> <li>• REBALANCE</li> <li>• REDISTRIBUTE</li> <li>• RESTART_RECREATE_INDEX</li> <li>• RESTORE</li> <li>• ROLLFORWARD_RECOVERY</li> <li>• RUNSTATS</li> <li>• GROUP_CRASH_RECOVERY</li> <li>• MEMBER_CRASH_RECOVERY</li> </ul>
UTILITY_PRIORITY	INTEGER	utility_priority - ユーティリティー優先度。ユーティリティーがスロットルをサポートする場合には優先順位、それ以外の場合は NULL。
UTILITY_DESCRIPTION	VARCHAR(2048)	utility_description - ユーティリティー記述。NULL にすることもできます。
UTILITY_DBNAME	VARCHAR(128)	utility_dbname - ユーティリティーで操作されるデータベース
UTILITY_START_TIME	TIMESTAMP	utility_start_time - ユーティリティー開始時刻

表 245. SNAPUTIL 管理ビューおよび SNAP\_GET\_UTIL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
UTILITY_STATE	VARCHAR(10)	utility_state - ユーティリティー状態。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• ERROR</li> <li>• EXECUTE</li> <li>• WAIT</li> </ul>
UTILITY_INVOKER_TYPE	VARCHAR(10)	utility_invoker_type - ユーティリティー呼び出し側タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• AUTO</li> <li>• USER</li> </ul>
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
PROGRESS_LIST_ATTR	VARCHAR(10)	progress_list_attr - 現在の進行リストの属性
PROGRESS_LIST_CUR_SEQ_NUM	INTEGER	progress_list_current_seq_num - 現在の進行リストのシーケンス番号
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPUTIL\_PROGRESS 管理ビューおよび SNAP\_GET\_UTIL\_PROGRESS 表関数 - progress 論理データ・グループ・スナップショット情報の検索

SNAPUTIL\_PROGRESS 管理ビューおよび SNAP\_GET\_UTIL\_PROGRESS 表関数は、特に progress 論理データ・グループのユーティリティー進行状況スナップショット情報を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 960 ページの『SNAPUTIL\_PROGRESS 管理ビュー』
- 961 ページの『SNAP\_GET\_UTIL\_PROGRESS 表関数』

### SNAPUTIL\_PROGRESS 管理ビュー

SNAPUTIL 管理ビューと組み合わせて使用すると、SNAPUTIL\_PROGRESS 管理ビューは、LIST UTILITIES SHOW DETAIL CLP コマンドと同じ情報を提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、963 ページの表 224 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPUTIL\_PROGRESS 管理ビューに対する SELECT 特権
- SNAPUTIL\_PROGRESS 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_UTIL\_PROGRESS 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

ユーティリティー ID ごとの合計進行単位および完了した進行単位の詳細を検索します。

```
SELECT UTILITY_ID, PROGRESS_TOTAL_UNITS, PROGRESS_COMPLETED_UNITS,  
       DBPARTITIONNUM FROM SYSIBMADM.SNAPUTIL_PROGRESS
```

以下はこの照会の出力例です。

UTILITY_ID	PROGRESS_TOTAL_UNITS	PROGRESS_COMPLETED_UNITS	DBPARTITIONNUM
7	10	5	0
9	10	5	1

1 record(s) selected.

## SNAP\_GET\_UTIL\_PROGRESS 表関数

SNAP\_GET\_UTIL\_PROGRESS 表関数は SNAPUTIL\_PROGRESS 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メン

パーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_UTIL 表関数と組み合わせて使用すると、SNAP\_GET\_UTIL\_PROGRESS 表関数は、**LIST UTILITIES SHOW DETAIL CLP** コマンドと同じ情報を提供します。

戻される可能性のある情報の完全なリストは、963 ページの表 224 を参照してください。

## 構文

```
▶▶—SNAP_GET_UTIL_PROGRESS—(—member—)
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_UTIL\_PROGRESS 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_UTIL\_PROGRESS 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

現在接続されているメンバー上のユーティリティの進行状況の詳細を検索します。

```
SELECT UTILITY_ID, PROGRESS_TOTAL_UNITS, PROGRESS_COMPLETED_UNITS,  
       DBPARTITIONNUM FROM TABLE(SNAP_GET_UTIL_PROGRESS(-1)) as T
```

以下はこの照会の出力例です。

```
UTILITY_ID PROGRESS_TOTAL_UNITS PROGRESS_COMPLETED_UNITS DBPARTITIONNUM  
-----  
          7                10                5                0
```

1 record(s) selected.

### 戻される情報

表 246. SNAPUTIL\_PROGRESS 管理ビューおよび SNAP\_GET\_UTIL\_PROGRESS 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
UTILITY_ID	INTEGER	utility_id - ユーティリティ ID。データベース・パーティションに固有。
PROGRESS_SEQ_NUM	INTEGER	progress_seq_num - 進行シーケンス番号。逐次の場合、フェーズの数。並行の場合、NULL の場合があります。
UTILITY_STATE	VARCHAR(16)	utility_state - ユーティリティ状態。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"><li>• ERROR</li><li>• EXECUTE</li><li>• WAIT</li></ul>
PROGRESS_DESCRIPTION	VARCHAR(2048)	progress_description - 進行の記述
PROGRESS_START_TIME	TIMESTAMP	progress_start_time - 進行開始時刻。フェーズが開始済みの場合には開始時刻、それ以外の場合は NULL。

表 246. SNAPUTIL\_PROGRESS 管理ビューおよび SNAP\_GET\_UTIL\_PROGRESS 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
PROGRESS_WORK_METRIC	VARCHAR(16)	progress_work_metric - 進行作業メトリック。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• NOT_SUPPORT</li> <li>• BYTES</li> <li>• EXTENTS</li> <li>• INDEXES</li> <li>• PAGES</li> <li>• ROWS</li> <li>• TABLES</li> </ul>
PROGRESS_TOTAL_UNITS	BIGINT	progress_total_units - 合計進行作業単位
PROGRESS_COMPLETED_UNITS	BIGINT	progress_completed_units - 完了した進行作業単位
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAP\_WRITE\_FILE プロシージャ

SNAP\_WRITE\_FILE プロシージャはシステム・スナップショット・データを、インスタンス・ディレクトリーの tmp サブディレクトリーにあるファイルに書き込みます。

### 構文

```
▶▶—SNAP_WRITE_FILE—(—requestType—,—dbname—,—member—)————▶▶
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### requestType

有効なスナップショット要求タイプを指定する、タイプ VARCHAR(32) の入力引数。可能な要求タイプは、sqlmon.h での定義を基にしたテキスト ID です。以下のいずれかとなります。

- APPL\_ALL
- BUFFERPOOLS\_ALL
- DB2

- DBASE\_ALL
- DBASE\_LOCKS
- DBASE\_TABLES
- DBASE\_TABLESPACES
- DYNAMIC\_SQL

#### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### *member*

有効なメンバー番号を指定する、タイプ INTEGER の入力引数。現在のメンバーの場合は -1、すべてのアクティブなメンバーの集合の場合は -2 を指定します。アクティブなメンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

プロシージャーを実行するには、ユーザーに SYSADM、SYSCTRL、SYSMON、または SYSMON 権限が必要です。保存されたスナップショットは、スナップショット表関数への入力として NULL 値を渡すことにより、SYSADM、SYSCTRL、SYSMAINT、または SYSMON 権限のないユーザーでも読み取れます。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャーが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

'DB2' の要求タイプ (SQLMA\_DB2 に相当) を指定し、現在接続されているデータベースおよび現行データベース・パーティションをデフォルトにすることで、データベース・マネージャー情報のスナップショットをとります。

```
CALL SYSPROC.SNAP_WRITE_FILE ('DB2', '', -1)
```

この場合、スナップショット・データは、インスタンス一時ディレクトリー (UNIX オペレーティング・システムでは sqllib/tmp/SQLMA\_DB2.dat、Windows オペレーティング・システムでは sqllib¥DB2¥tmp¥SQLMA\_DB2.dat) に書き込まれます。

## 使用上の注意

未認識の入力パラメーターが指定された場合、「SQL2032N "REQUEST\_TYPE" パラメーターが無効です」エラーが戻されます。

## TBSP\_UTILIZATION 管理ビュー - 表スペースの構成および使用率に関する情報の検索

TBSP\_UTILIZATION 管理ビューは、表スペースの構成および使用率に関する情報を戻します。

このビューは、単一パーティション・データベースに対する **LIST TABLESPACES** コマンドと同様のレポートを取得します。その情報は、SNAPTbsp、SNAPTbsp\_Part 管理ビューと TABLESPACES カタログ・ビューに基づきます。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- TBSP\_UTILIZATION 管理ビューに対する SELECT 特権
- TBSP\_UTILIZATION 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCtrl
- SYSMAINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

単一パーティション・データベース上での **LIST TABLESPACES** コマンドと同じレポートを検索します。

```
SELECT TBSP_ID, SUBSTR(TBSP_NAME,1,20) as TBSP_NAME, TBSP_TYPE,
       TBSP_CONTENT_TYPE, TBSP_STATE FROM SYSIBMADM.TBSP_UTILIZATION
```

以下はこの照会の出力例です。

TBSP_ID	TBSP_NAME	TBSP_TYPE	...
0	SYSCATSPACE	DMS	...
1	TEMPSPACE1	DMS	...
2	USERSPACE1	DMS	...
3	SYSTOOLSPACE	DMS	...
4	SYSTOOLSTMPSPACE	DMS	...

この照会の出力 (続き)。



```

... TBSP_CONTENT_TYPE TBSP_STATE
... -----
... ANY                NORMAL
... SYSTEMP           NORMAL
... ANY                NORMAL
... ANY                NORMAL
... USRTEMP           NORMAL

```

## 戻される情報

表 247. TBSP\_UTILIZATION 管理ビューによって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TBSP_TYPE	VARCHAR(10)	tablespace_type - 表スペース・タイプ
TBSP_CONTENT_TYPE	VARCHAR(10)	tablespace_content_type - 表スペースのコンテンツ・タイプ
TBSP_CREATE_TIME	TIMESTAMP	表スペースの作成時刻。
TBSP_STATE	VARCHAR(256)	tablespace_state - 表スペースの状態
TBSP_TOTAL_SIZE_KB	BIGINT	表スペースの合計サイズ (KB)。計算式は total_pages*pagesize/1024。
TBSP_USABLE_SIZE_KB	BIGINT	表スペースの合計使用可能サイズ (KB)。計算式は usable_pages*pagesize/1024。
TBSP_USED_SIZE_KB	BIGINT	表スペースの合計使用済みサイズ (KB)。計算式は used_pages*pagesize/1024。
TBSP_FREE_SIZE_KB	BIGINT	表スペースの合計利用可能サイズ (KB)。計算式は free_pages*pagesize/1024。
TBSP_UTILIZATION_PERCENT	BIGINT	表スペースの使用率 (パーセンテージ)。 usable_pages が使用可能な場合、計算式は (used_pages/usable_pages)*100。使用可能でない場合には -1 が表示されます。
TBSP_TOTAL_PAGES	BIGINT	tablespace_total_pages - 表スペース内の合計ページ数
TBSP_USABLE_PAGES	BIGINT	tablespace_usable_pages - 表スペース内の使用可能ページ数
TBSP_USED_PAGES	BIGINT	tablespace_used_pages - 表スペース内の使用されているページ数
TBSP_FREE_PAGES	BIGINT	tablespace_free_pages - 表スペース内のフリー・ページ数
TBSP_PAGE_TOP	BIGINT	tablespace_page_top - 表スペース最高水準点

表 247. TBSP\_UTILIZATION 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_PAGE_SIZE	INTEGER	tablespace_page_size - 表スペースのページ・サイズ
TBSP_EXTENT_SIZE	INTEGER	tablespace_extent_size - 表スペースのエクス Tent・サイズ
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 表スペースのプリフェッチ・サイズ
TBSP_MAX_SIZE	BIGINT	tablespace_max_size - 表スペースの最大サイズ
TBSP_INCREASE_SIZE	BIGINT	tablespace_increase_size - バイト単位のサイズの増加
TBSP_INCREASE_SIZE_PERCENT	SMALLINT	tablespace_increase_size_percent - パーセント単位のサイズの増加
TBSP_LAST_RESIZE_TIME	TIMESTAMP	tablespace_last_resize_time - 最後にサイズ変更が正常に行われた時刻
TBSP_LAST_RESIZE_FAILED	SMALLINT	tablespace_last_resize_failed - 失敗した最後のサイズ変更
TBSP_USING_AUTO_STORAGE	SMALLINT	tablespace_using_auto_storage - 自動ストレージが使用可能な表スペース
TBSP_AUTO_RESIZE_ENABLED	SMALLINT	tablespace_auto_resize_enabled - 表スペースの自動サイズ変更可能
DBPGNAME	VARCHAR(128)	表スペースのデータベース・パーティション・グループの名前。
TBSP_NUM_CONTAINERS	BIGINT	tablespace_num_containers - 表スペース内のコンテナ数
REMARKS	VARCHAR(254)	ユーザーが入力したコメント。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

## TOP\_DYNAMIC\_SQL 管理ビュー - 上位動的 SQL ステートメントに関する情報の検索

TOP\_DYNAMIC\_SQL 管理ビューは、実行数、平均実行時間、ソート数、またはステートメントあたりのソートによってソートできる動的 SQL ステートメントのうち、上位のものを戻します。これは適切な調整を行うために注目すべき照会です。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- TOP\_DYNAMIC\_SQL 管理ビューに対する SELECT 特権
- TOP\_DYNAMIC\_SQL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

実行頻度の高い上位 5 つの SQL を識別します。

```
SELECT NUM_EXECUTIONS, AVERAGE_EXECUTION_TIME_S, STMT_SORTS,
       SORTS_PER_EXECUTION, SUBSTR(STMT_TEXT,1,60) AS STMT_TEXT
FROM SYSIBMADM.TOP_DYNAMIC_SQL
ORDER BY NUM_EXECUTIONS DESC FETCH FIRST 5 ROWS ONLY
```

以下はこの照会の出力例です。

NUM_EXECUTIONS	AVERAGE_EXECUTION_TIME_S	STMT_SORTS	...
148	0	0	...
123	0	0	...
2	0	0	...
1	0	0	...
1	0	0	...

5 record(s) selected.

この照会の出力 (続き)。

```
... SORTS_PER_EXECUTION ...
... ----- ...
... 0 ...
... 0 ...
... 0 ...
... 0 ...
... 0 ...
```

この照会の出力 (続き)。

```
... STMT_TEXT
... -----
... SELECT A.ID, B.EMPNO, B.FIRSTNME, B.LASTNAME, A.DEPT FROM E
... SELECT A.EMPNO, A.FIRSTNME, A.LASTNAME, B.LOCATION, B.MGRNO
... SELECT A.EMPNO, A.FIRSTNME, A.LASTNAME, B.DEPTNAME FROM EMP
... SELECT ATM.SCHEMA, ATM.NAME, ATM.CREATE_TIME, ATM.LAST_WAIT,
... SELECT * FROM JESSICAE.EMP_RESUME
```

## 戻される情報

表 248. TOP\_DYNAMIC\_SQL 管理ビューによって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	レポートのタイム・スタンプ。
NUM_EXECUTIONS	BIGINT	num_executions - ステートメント実行回数
AVERAGE_EXECUTION_TIME_S	BIGINT	平均実行時間。
STMT_SORTS	BIGINT	stmt_sorts - ステートメント・ソート回数
SORTS_PER_EXECUTION	BIGINT	ステートメントの実行ごとに行われるソートの数。
STMT_TEXT	CLOB(2 M)	stmt_text - SQL 動的ステートメント・テキスト
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SQL プロシージャ・ルーチン

### ALTER\_ROUTINE\_PACKAGE プロシージャ

このプロシージャは、コンパイル済み SQL ルーチンまたはコンパイル済みトリガーに関連付けられたパッケージに使用される値を、再バインドを必要とせずに変更します。

機能的には ALTER PACKAGE ステートメントと同等ですが、パッケージ名の代わりにオブジェクト名を引数として使用します。 ALTER\_ROUTINE\_PACKAGE プロシージャは、コマンド行またはアプリケーションから呼び出すことができます。

▶▶ALTER\_ROUTINE\_PACKAGE(—type—,—schema—,—module—,—name—,—options—)————▶▶

スキーマは SYSPROC です。

### プロシージャ・パラメーター

*type*

以下のいずれかの値を使用してルーチンまたはコンパイル・トリガーのタイプを指定する、タイプ CHAR(2) の入力引数。

- 'P' - プロシージャ
- 'SP' - プロシージャの特定名
- 'F' - コンパイル関数
- 'SF' - コンパイル関数の特定名
- 'T' - コンパイル・トリガー

### *schema*

ルーチンまたはトリガーのスキーマを指定する、タイプ VARCHAR(128) のオプションの入力引数。スキーマが指定されない場合、値はデフォルトで CURRENT SCHEMA 特殊レジスターの値になります。このパラメーターには、大文字と小文字の区別があります。

### *module*

ルーチンがあるモジュールの名前を指定する、タイプ VARCHAR(128) のオプションの入力引数。トリガーの場合、このパラメーターを指定することはできません。このパラメーターを指定しない場合、モジュール・ルーチンは無視されます。このパラメーターには、大文字と小文字の区別があります。

### *name*

ルーチンまたはトリガーの名前を指定する、タイプ VARCHAR(128) の入力引数。このパラメーターには、大文字と小文字の区別があります。

### *options*

ALTER PACKAGE ステートメントがサポートしているオプションのリストを指定する、タイプ VARCHAR(1024) の入力引数。 *options* パラメーター内に、ALTER PACKAGE 節を少なくとも 1 つ指定する必要があります。

## 許可

このプロシージャを実行するには、以下のいずれかの権限が必要です。

- プロシージャでの EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

UPDATE\_EMPLOYEE という名前の既存のストアード・プロシージャの基礎パッケージを変更します。

```
CALL SYSPROC.ALTER_ROUTINE_PACKAGE ('P','','','UPDATE_EMPLOYEE',  
  'ACCESS PLAN REUSE YES OPTIMIZATION PROFILE AYYANG.INDEXHINTS')
```

DRICARD スキーマの MIN\_SALARY というコンパイル済みトリガーのパッケージを変更します。

```
CALL SYSPROC.ALTER_ROUTINE_PACKAGE ('T','DRICARD','','MIN_SALARY',  
  'OPTIMIZATION PROFILE AYYANG.INDEXHINTS')
```

3 部構成の名前を使用して、コンパイル済み関数のパッケージを変更します。

```
CALL SYSPROC.ALTER_ROUTINE_PACKAGE ('F','DRICARD','MODULE','FUNCTION','APREUSE YES')
```

## GET\_ROUTINE\_NAME プロシージャ - SQL PL オブジェクト名の取得

GET\_ROUTINE\_NAME プロシージャは、オブジェクトの LIB\_ID が指定された場合に、その SQL PL オブジェクトの名前を返します。

### 許可

以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- SQLADM 権限
- DBADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 構文

```
▶ GET_ROUTINE_NAME (—lib_id—, —type—, —schema—, —————▶  
▶ —module—, —name—, —specific_name—) —————▶
```

スキーマは SYSPROC です。

### ルーチン・パラメーター

#### *lib\_id*

コンパイル済み SQL PL ルーチンまたはトリガーに対応する LIB\_ID を指定する、タイプ INTEGER の入力引数。これは、ランタイム SQL PL エラーの SQLCA にある SQLERRD(4) フィールドの値です。

#### *type*

エラーを引き起こしたオブジェクトのタイプを指定する、タイプ CHAR(1) の出力引数。以下のいずれかの値を使用します。

- 'P' - プロシージャ
- 'F' - 関数
- 'T' - トリガー

#### *schema*

オブジェクトが属するスキーマを指定する、タイプ VARCHAR(128) の出力引数。

#### *module*

オブジェクトが属するモジュールを指定する、タイプ VARCHAR(128) の出力引数。オブジェクトがトリガーである場合、またはモジュールに属していない場合は、NULL になります。

#### *name*

オブジェクトの名前を指定する、タイプ VARCHAR(128) の出力引数。

### *specific\_name*

オブジェクトの特定名を指定する、タイプ VARCHAR(128) の出力引数。オブジェクトがトリガーの場合は、NULL になります。

### 使用上の注意

- SQLCA 構造に、SQL PL ルーチンまたはトリガーの実行時に発生したエラーが入っている場合、SQLCAID フィールドには値 'SQLCAM' が入り、SQLERRM(4) にはエラーを引き起こしたオブジェクトを一意的に識別する数値が入ります。このユニーク ID は、SYSCAT.ROUTINES および SYSCAT.TRIGGERS カタログ・ビューの LIB\_ID 列に対応します。GET\_ROUTINE\_NAME ルーチンは、指定された LIB\_ID 値を、対応するオブジェクト名にマップします。
- SYSPROC.GET\_ROUTINE\_NAME に指定した最初の引数の値が SQL PL のルーチンまたはトリガーに対応していない場合、このプロシージャはすべての出力パラメーターに NULL を返し、ルーチンまたはトリガーが見つからなかったことを示します。

### 例

アプリケーションの実行がエラー終了し、SQLCA 構造に以下の情報が記述されているとします。

```
SQLCA Information
  sqlcaid : SQLCAM      sqlcabc: 136   sqlcode: -801
  ...
  sqlerrd : (1) 0        (2) 0        (3) 3
            (4) 13152254 (5) 0        (6) 0
  ...
  sqlstate: 22012
```

SQLCAID フィールドに SQLCAM とあることから、SQL PL オブジェクトの実行中にエラーが発生したことが分かります。SQLERRD(3) および SQLERRD(4) フィールドに 3 および 13152254 という値があることは、値 13152254 によって識別される SQL PL オブジェクトの 3 行目にあるステートメントがエラーの発生原因であることを示しています。その SQL PL オブジェクトが何であるかを判別するには、次のような方法で SYSPROC.GET\_ROUTINE\_NAME を呼び出します。

```
db2 call "GET_ROUTINE_NAME(13152254, ?, ?, ?, ?, ?)"
```

```
Value of output parameters
-----
Parameter Name : OBJTYPE
Parameter Value : P

Parameter Name : OBJSCHEMA
Parameter Value : APPDEV

Parameter Name : OBJMODULE
Parameter Value : -

Parameter Name : OBJNAME
Parameter Value : PROC2

Parameter Name : OBJSPECIFIC
Parameter Value : APPDEV_PROC2

Return Status = 0
```

SYSPROC.GET\_ROUTINE\_NAME への呼び出しから得られる OUT パラメーターの値は、ストアード・プロシージャ APPDEV.PROC2 からエラーが発生したことを示しています。

## GET\_ROUTINE\_OPTS

GET\_ROUTINE\_OPTS 関数は、現行のセッションで SQL プロシージャを作成するのに使用されるオプションの文字ストリング値を返します。

▶▶—GET\_ROUTINE\_OPTS—(—)————→

スキーマは SYSPROC です。

### 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数の結果は、長さ属性が 1024 である、可変長文字ストリング (VARCHAR) 値です。

### 例

照会の結果として SQL プロシージャを作成するのに使用されるオプションを返します。

```
SELECT GET_ROUTINE_OPTS()  
FROM SYSTBM.SYSDUMMY1
```

## GET\_ROUTINE\_SAR

GET\_ROUTINE\_SAR プロシージャは、同じオペレーティング・システムで同じレベルを実行している別のデータベース・サーバーで、同じルーチンをインストールするために必要な情報を検索します。情報は、SQL アーカイブ・ファイルを表す単一 BLOB ストリングへ取り出されます。

▶▶—GET\_ROUTINE\_SAR————→

▶(—sarblob—,—type—,—routine-name-string—)————→  
  [,—hide-body-flag—]

スキーマは SYSFUN です。



## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## プロシージャ・パラメーター

### *sarblob*

ルーチン SAR ファイル・コンテンツを含む、タイプ BLOB(3M) の出力引数。

### *type*

以下のいずれかの値を使用してルーチンのタイプを指定する、タイプ CHAR(2) の入力引数。

- 'P' - プロシージャ
- 'SP' - プロシージャの特定名

### *routine-name-string*

ルーチンの修飾名を指定する、タイプ VARCHAR(257) の入力引数。スキーマ名が指定されていない場合、ルーチンが処理される時デフォルトは CURRENT SCHEMA になります。 *routine-name-string* に二重引用符 (") を組み込むことはできません。

### *hide-body-flag*

カタログからルーチンのテキストを抽出する際にルーチンの本文を隠すかどうかを指定する、タイプ INTEGER の入力引数 (次のいずれかの値を使用します)。有効な値は以下のとおりです。

0       ルーチンのテキストをそのまま残す。これはデフォルト値です。

1       カタログからルーチンのテキストを抽出する際に、ルーチンの本文を空の本文に置き換える。

ルーチンの修飾名は、検索するルーチンを決定するために使用されます。検出されるルーチンは、SQL ルーチンでなければなりません。また、特定の名前を使用せずに検索を行うと、複数のルーチンが検出されて、エラー (SQLSTATE 42725) になる場合があります。このような場合は、必ず、検索したいルーチンの固有の名前を使用してください。

SAR ファイルは、サーバーで使用可能ではない可能性のあるバインド・ファイルを組み込む必要があります。バインド・ファイルが見つからず、SAR ファイルに保管できない場合、エラーが起きます (SQLSTATE 55045)。

## PUT\_ROUTINE\_SAR

PUT\_ROUTINE\_SAR プロシージャは、サーバーで SQL ルーチンを作成するために必要なファイルを渡し、ルーチンを定義します。

### 許可

このプロシージャを実行するには、以下のいずれかの権限が必要です。

- プロシージャでの EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

```
▶▶ PUT_ROUTINE_SAR (—sarblob [—new-owner—, —use-register-flag—]) ▶▶
```

スキーマは SYSFUN です。

### プロシージャ・パラメーター

#### *sarblob*

ルーチン SAR ファイル・コンテンツを含む、タイプ BLOB(3M) の入力引数。

#### *new-owner*

ルーチンの許可検査に使用される許可名を含む、タイプ VARCHAR(128) の入力引数。 *new-owner* は、定義されるルーチンに必要な権限を持っていない限りなりません。 *new-owner* が指定されない場合、オリジナル・ルーチン定義者の許可名が使用されます。

#### *use-register-flag*

CURRENT SCHEMA および CURRENT PATH 特殊レジスターがルーチンの定義に使用されるかどうかを指示する、タイプ INTEGER の入力引数。特殊レジスターが使用されない場合、ルーチンがはじめて定義される場合はデフォルト・スキーマと SQL パスの設定が使用されます。 *use-register-flag* に考えられる値は以下のとおりです。

- 0 現行環境の特殊レジスターを使用しません。
- 1 CURRENT SCHEMA および CURRENT PATH 特殊レジスターを使用します。

値が 1 の場合、ルーチン定義 (ルーチンの名前を含む) の非修飾オブジェクト名に CURRENT SCHEMA が使用され、ルーチン定義の非修飾ルーチンとデータ・タイプを解決するために CURRENT PATH が使用されます。

*use-registers-flag* が指定されない場合、振る舞いは 0 が指定されたときと同じです。

*sarblob* に含まれる識別情報がチェックされ、入力が環境に対して適切であるかどうかを確認されます。入力が適切でない場合は、エラーが起きます (SQLSTATE 55046)。PUT\_ROUTINE\_SAR プロシージャは次に、*sarblob* の内容を使用してサーバーでルーチンを定義します。

*sarblob* 引数の内容は SQL アーカイブ・ファイルを構成する個々のファイルへ抽出されます。共有ライブラリーとバインド・ファイルは、一時ディレクトリーのファイルに書き込まれます。環境は、コンパイルおよびリンクが不要であること、共有ライブラリーとバインド・ファイルのロケーションが使用可能であることを、ルーチン定義ステートメント処理に認識させるように設定されます。DDL ファイルの内容が、ルーチン定義ステートメントを動的に実行するために使用されます。

指定のスキーマの下で、複数のプロシージャが並行してインストールされることはありません。

このステートメントの処理は、他のインターフェースを使用してルーチン定義ステートメントを実行するのと同じエラーを起こす可能性があります。ルーチン定義処理中に、共有ライブラリーとバインド・ファイルの存在が通知され、プリコンパイル、コンパイル、およびリンクのステップがスキップされます。バインド・ファイルはバインド処理中に使用され、両方のファイルの内容が SQL ルーチンの通常ディレクトリーにコピーされます。

GET ROUTINE または PUT ROUTINE 操作 (またはそれに対応するプロシージャ) が正常に実行できない場合、エラー (SQLSTATE 38000)、および失敗の原因に関する情報を示す診断テキストを毎回戻します。例えば、GET ROUTINE に指定されたプロシージャ名が SQL プロシージャを識別しない場合、"-204, 42704" という診断テキストが戻ります。ここで、"-204" は SQLCODE、"42704" は SQLSTATE で、それぞれ問題の原因を示します。この例の SQLCODE と SQLSTATE は、GET ROUTINE コマンドで指定されたプロシージャ名が未定義であることを示します。

## REBIND\_ROUTINE\_PACKAGE プロシージャ - パッケージの再バインド

REBIND\_ROUTINE\_PACKAGE プロシージャは、SQL プロシージャ、ルーチン、コンパイル済み関数、またはトリガーに関連付けられたパッケージを再バインドします。これは機能的には REBIND コマンドと同じですが、これは、パッケージ名の代わりにプロシージャ名を引数として使用します。

REBIND\_ROUTINE\_PACKAGE プロシージャは、コマンド行またはアプリケーションから呼び出すことができます。

### 構文

REBIND\_ROUTINE\_PACKAGE を呼び出すには、いずれも同じように有効な 2 つの方法があります。2 つの呼び出しの違いは、ルーチン名を指定する方法だけです。1 つ目の例では、*routine-name-string* 変数はピリオドで区切られた ID 名で構成されます。2 つ目の方法では、ルーチンは *schema*、*module*、および *name* 値という別個の値で識別されます。

方法 1:

▶▶—REBIND\_ROUTINE\_PACKAGE—(—*type*—, —*routine-name-string*—, —*options*—)——▶▶

方法 2:

▶▶—REBIND\_ROUTINE\_PACKAGE—(—*type*—, —————▶▶

▶—*schema*—, —*module*—, —*name*—, —*options*—)——▶▶

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *type*

以下のいずれかの値を使用してルーチンまたはコンパイル・トリガーのタイプを指定する、タイプ CHAR(2) の入力引数。

- 'P' - プロシージャ
- 'SP' - プロシージャの特定名
- 'F' - コンパイル関数
- 'SF' - コンパイル関数の特定名
- 'T' - コンパイル・トリガー

### *routine-name-string* (方法 1 のみ)

ルーチンまたはトリガーの名前を指定する、タイプ VARCHAR(386) の入力引数。トリガー名はピリオドで区切られた 2 つの部分で構成されていて、*schema.trigger* という形式です。スキーマはオプションです。ルーチン名はピリオドで区切られた 3 つの名前部分で構成されていて、*schema.module.routine* という形式です。スキーマとモジュールはオプションです。スキーマを指定しないと、この値はデフォルトの CURRENT SCHEMA 特殊レジスターの値になります。2 部構成の名前が指定されると、最初の部分はスキーマ名として当初解釈されます。そのスキーマ名の下にルーチンが見つからないと、最初の部分はモジュール名として解釈され、CURRENT SCHEMA の下のその名前のモジュール内でルーチンを検出しようとしています。スキーマ、モジュール、またはオブジェクト名に、二重引用符 (") またはピリオド (.) を含めることはできません。

### *schema* (方法 2 のみ)

ルーチンまたはトリガーのスキーマを指定する、タイプ VARCHAR(128) のオプションの入力引数。スキーマが指定されない場合、値はデフォルトで CURRENT SCHEMA 特殊レジスターの値になります。このパラメーターには、大文字と小文字の区別があります。

### *module* (方法 2 のみ)

ルーチンがあるモジュールの名前を指定する、タイプ VARCHAR (128) のオプションの入力引数。このパラメーターをトリガーに指定しないでください。このパラメーターを指定しない場合は、モジュール・ルーチンは無視されます。このパラメーターには、大文字と小文字の区別があります。

### *name* (方法 2 のみ)

ルーチンまたはトリガーの名前を指定する、タイプ VARCHAR (128) の入力引数。このパラメーターには、大文字と小文字の区別があります。

### options

REBIND コマンド構文に従うすべての再バインド・オプションのリストを指定する、タイプ VARCHAR(1024) のオプションの入力引数。後方互換性の目的で「ANY」または「CONSERVATIVE」という単一値もサポートされており、RESOLVE 再バインド・オプションの値として解釈されます。

ルーチンの修飾名は、検索するルーチンを決定するために使用されます。検出されるルーチンは、SQL ルーチンでなければなりません。それ以外は、エラーが返されます。(SQLSTATE 428F7) 特定名が使用されない場合は、複数のルーチンが検出される可能性があり、エラーが返されます。(SQLSTATE 42725) このような場合は、必ず、検索したいルーチンの固有の名前を使用してください。

### 許可

このプロシージャを実行するには、以下のいずれかの権限が必要です。

- プロシージャでの EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

例 1: RESOLVE、REOPT、および APREUSE オプションを使用して、ルーチン UPDATE\_EMPLOYEE のパッケージを再バインドします。

方法 1:  
CALL SYSPROC.REBIND\_ROUTINE\_PACKAGE (  
 'P','UPDATE\_EMPLOYEE','RESOLVE ANY REOPT ONCE APREUSE YES')

方法 2:  
CALL SYSPROC.REBIND\_ROUTINE\_PACKAGE (  
 'P','','','UPDATE\_EMPLOYEE','RESOLVE ANY REOPT ONCE APREUSE YES')

例 2: オプションを使用せずに、ルーチン UPDATE\_EMPLOYEE のパッケージを再バインドします。

方法 1:  
CALL SYSPROC.REBIND\_ROUTINE\_PACKAGE (  
 'P','UPDATE\_EMPLOYEE','')

方法 2:  
CALL SYSPROC.REBIND\_ROUTINE\_PACKAGE (  
 'P','','','UPDATE\_EMPLOYEE','')

例 3: コンパイル済みトリガーのパッケージを再バインドします。

方法 1:  
CALL SYSPROC.REBIND\_ROUTINE\_PACKAGE (  
 'T','DRICARD.MIN\_SALARY','REOPT ALWAYS')

方法 2:  
CALL SYSPROC.REBIND\_ROUTINE\_PACKAGE (  
 'T','DRICARD','','MIN\_SALARY','REOPT ALWAYS')

例 4: 3 つの部分に分かれた名前を使用して、コンパイル済み関数のパッケージを再バインドします。

方法 1

```
CALL SYSPROC.REBIND_ROUTINE_PACKAGE (  
    'F','DRICARD.MODULE.FUNCTION','REOPT ALWAYS')
```

方法 2

```
CALL SYSPROC.REBIND_ROUTINE_PACKAGE (  
    'F','DRICARD','MODULE','FUNCTION','REOPT ALWAYS')
```

## SET\_ROUTINE\_OPTS

SET\_ROUTINE\_OPTS プロシージャは、現行セッションで SQL プロシージャを作成するのに使用されるオプションを設定します。この設定は、

**DB2\_SQLROUTINE\_PREOPTS** レジストリー変数で指定された、インスタンス全体の設定をオーバーライドします。

▶▶—SET\_ROUTINE\_OPTS—(—*character-expression*—)—————▶▶

スキーマは SYSPROC です。

### プロシージャ・パラメーター

*character-expression*

現行セッションのオプション設定を指定する、タイプ VARCHAR(1024) の入力引数。

指定されたオプションは、セッション期間に有効です。引数として NULL 値が指定される場合、**DB2\_SQLROUTINE\_PREOPTS** レジストリー変数の値は、現行セッションのデフォルトのオプション設定としてリストアされます。許可されたオプションのリストは、『照会コンパイラ変数』の下にある、**DB2\_SQLROUTINE\_PREOPTS** レジストリー変数の説明を参照してください。

### 許可

このプロシージャを実行するには、以下の権限のいずれかが必要です。

- プロシージャでの EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

例 1: 現行セッションのオプション設定を NULL に設定します。

```
CALL SYSPROC.SET_ROUTINE_OPTS(CAST (NULL AS VARCHAR(1)))
```

例 2: 現行セッションのオプション設定を EXPLAIN YES に設定します。

```
CALL SET_ROUTINE_OPTS('EXPLAIN YES')
```

例 3: 現行セッションのオプション設定を EXPLAIN YES および BLOCKING NO に設定します。

```
CALL SET_ROUTINE_OPTS('EXPLAIN YES BLOCKING NO')
```

## 段階的な再配分ルーチン

### ANALYZE\_LOG\_SPACE プロシージャ - ログ・スペース分析情報の検索

ANALYZE\_LOG\_SPACE プロシージャは、指定されたデータベース・パーティション・グループのデータベース・パーティションそれぞれのログ・スペース分析結果を戻します。

#### 構文

```
►—ANALYZE_LOG_SPACE—(—inDBPGroup—,—inMainTbSchema—,—inMainTable—,————►  
►—analysisType—,—inStmgTime—,—addDropOption—,—addDropList—,—pNumber—,———►  
►—pWeight—)——————►
```

スキーマは SYSPROC です。

#### プロシージャ・パラメーター

##### *inDBPGroup*

データベース・パーティション・グループ名を指定する、タイプ VARCHAR (128) の入力引数。

##### *inMainTbSchema*

メイン表のスキーマを指定する、タイプ VARCHAR (128) の入力引数。

##### *inMainTable*

データベース・パーティション・グループ内のメイン表を指定する、タイプ VARCHAR (128) の入力引数。この表は通常、データベース・パーティション・グループ内で最大の表です。

##### *analysisType*

分析タイプの標識を指定する、タイプ SMALLINT の入力引数。以下の標識があります。

- SWRD\_USE\_STMG\_TABLE (1): データベース・パーティションあたりの表の行数を検出するのに、ストレージ管理表内の情報が使用されることを表します。このタイプは、ストレージ管理表がセットアップされていて、再配分の対象のデータベース・パーティション・グループのストレージ・スナップショットが少なくとも 1 つ取られている場合のみ、使用してください。
- SWRD\_USE\_REALTIME\_ANALYSIS (2): データベース・パーティションあたりの表の行数を検出するのに、SELECT 照会が使用されることを表します。

##### *inStmgTime*

ストレージ管理レコードのタイム・スタンプを指定する、タイプ VARCHAR (26) の入力引数。 *analysisType* を SWRD\_USE\_REALTIME\_ANALYSIS に設定した場合、このパラメーターは無視されます。

### *addDropOption*

追加またはドロップされるデータベース・パーティションを指定する、タイプ CHAR (1) の入力引数。以下の値を使用できます。

- 'A': データベース・パーティションを追加
- 'D': データベース・パーティションをドロップ
- 'N': 追加もドロップもしない

### *addDropList*

追加またはドロップされるデータベース・パーティションを指定する、タイプ VARCHAR (6000) の入力引数。このデータベース・パーティション番号は、コンマで区切られたストリング・フォーマットで指定します。ストリング内でスペースは使用できません。

### *pNumber*

すべてのデータベース・パーティション番号を指定するタイプ VARCHAR (6000) の入力引数で、データベース・パーティションの重みに対応します。各データベース・パーティション番号は、0 から 999 までの範囲の数値です。また、データベース・パーティション番号は、コンマで区切られたストリングで指定し、ストリング内にスペースは使用できません。

### *pWeight*

すべてのデータベース・パーティションのユーザー指定の重みを指定するタイプ VARCHAR (6000) の入力引数で、*pNumber* ストリング内のデータベース・パーティション番号に対応します。各データベース・パーティションの重みは、0 から 32767 までの範囲の数値です。また、データベース・パーティションの重みは、コンマで区切られたストリングで指定し、ストリング内にスペースは使用できません。

## 許可

- SYSADM, SYSMON, SYSCTRL, または SYSMAINT
- ANALYZE\_LOG\_SPACE プロシージャに対する EXECUTE 特権

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

データベース・パーティションを追加した場合の影響を、変更を適用しないで分析します。ここでは、データベース・パーティション・グループにデータベース・パーティション 40、50、60 を追加し、データベース・パーティション 10、20、30、40、50、60 のターゲット率を 1:2:1:2:1:2 にした場合を仮説とします。この例では、データベース・パーティション・グループに実際に存在するのは、パーティション 10、20、30 だけであることを注意してください。

```
CALL SYSPROC.ANALYZE_LOG_SPACE('IBMDEFAULTGROUP', 'TEST',  
    'EMP', 2, ' ', 'A', '40,50,60', '10,20,30,40,50,60',  
    '1,2,1,2,1,2')
```

データベース・パーティションをドロップした場合の影響を、変更を適用しないで分析します。ここでは、データベース・パーティション・グループからデータベー



ス・パーティション 30 をドロップし、データベース・パーティション 10 および 20 に、ターゲット率 1 : 1 でデータを再配分した場合を仮説とします。この例では、データベース・パーティション 10、20、30 がすべてデータベース・パーティション・グループに存在していなければならないことに注意してください。

```
CALL SYSPROC.ANALYZE_LOG_SPACE('IBMDEFAULTGROUP', 'TEST',
    'EMP', 2, ' ', 'D', '30', '10,20', '1,1')
```

## 使用上の注意

パラメーターの値を取得できない場合は、その出力値として -1 が使用されます。

再配分に関するストアード・プロシージャおよび関数は、各表の分散キーが定義されているパーティション・データベース環境でのみ動作します。

## 戻される情報

ANALYZE\_LOG\_SPACE プロシージャは、ログ・スペース分析結果の結果セット (オープン・カーソル) を戻します。この結果セットには、指定されたデータベース・パーティション・グループのデータベース・パーティションごとに、以下のフィールドが含まれます。

表 249. ANALYZE\_LOG\_SPACE プロシージャによって戻される情報

列名	列タイプ	説明
PARTITION_NUM	SMALLINT	ログ・スペース分析のデータベース・パーティション番号。
TOTAL_LOG_SIZE	BIGINT	割り振られるログ・スペースの合計 (バイト単位)。-1 はサイズの制限がないことを表します。
AVAIL_LOG_SPACE	BIGINT	フリーであり再配分処理で使用できるログ・スペース量 (バイト単位)。
DATA_SKEW	BIGINT	ターゲット・レベルから逸脱するデータのサイズの絶対値 (バイト単位)。
REQ_LOG_SPACE	BIGINT	望ましいデータ配分に到達するのに必要なスペース量 (バイト単位)。
NUM_OF_STEPS	SMALLINT	データ・スキューをゼロまで減らすのに必要なステップ数。
MAX_STEP_SIZE	BIGINT	ログ・フル・エラーにならずに一度に移動できるデータ最大量 (バイト単位)。

## GENERATE\_DISTFILE プロシージャ - データ配分ファイルの生成

GENERATE\_DISTFILE プロシージャは、指定された表のデータ配分ファイルを生成し、指定されたファイル名で保存します。

### 構文

```
►►—GENERATE_DISTFILE—(—inTbSchema—, —inTbName—, —fileName—)————►►
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

*inTbSchema*

表スキーマ名を指定する、タイプ VARCHAR (128) の入力引数。

*inTbName*

表名を指定する、タイプ VARCHAR (128) の入力引数。

*fileName*

データ配分ファイル名を指定する、タイプ VARCHAR (255) の入力または出力引数。指定したファイル名が単なるファイル名だった場合、ファイルはインスタンス・ディレクトリー下の tmp サブディレクトリーに保存され、このパラメーターに完全ファイル・パス名が戻されます。

## 許可

- GENERATE\_DISTFILE プロシージャに対する EXECUTE 特権。
- SYSCAT.TABLES、SYSCAT.COLUMNS、および指定した表に対する SELECT 特権。

加えて、fenced ユーザー ID は、インスタンス・ディレクトリー下の tmp サブディレクトリーにファイルを作成できなければなりません。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

再配分処理で使用されるデータ配分ファイルを生成します。

```
CALL SYSPROC.GENERATE_DISTFILE('TEST', 'EMP',  
    '$HOME/sql1lib/function/SAMPLE.IBMDEFAULTGROUP_swrData.dst')
```

## 使用上の注意

再配分に関するストアード・プロシージャおよび関数は、各表の分散キーが定義されているパーティション・データベース環境でのみ動作します。

## GET\_SWRD\_SETTINGS プロシージャ - 再配分情報の検索

GET\_SWRD\_SETTINGS プロシージャは、指定されたデータベース・パーティション・グループの既存の再配分レジストリー・レコードを読み取ります。

## 構文

```
▶▶ GET_SWRD_SETTINGS ( ( dbpgName —, matchingSpec —, redistMethod —, —————▶  
▶ pMapFile —, distFile —, stepSize —, totalSteps —, stageSize —, —————▶  
▶ nextStep —, processState —, pNumber —, pWeight —) —————▶▶
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *dbpgName*

再配分処理の実行の対象となるデータベース・パーティション・グループ名を指定する、タイプ VARCHAR(128) の入力引数。

### *matchingSpec*

表 250 に示すビット単位フィールド ID を指定する、タイプ SMALLINT の入力引数。出力パラメーターによって戻されるターゲット・フィールドを、これで表します。必須でない出力パラメーターは NULL 設定できます。

例えば *matchingSpec* を (REDIST\_STAGE\_SIZE | REDIST\_NEXT\_STEP) の整数値である 96 に設定した場合、この関数の呼び出し側は、値を受け取るためには *stageSize* と *nextStep* を指定するだけでよく、残りの出力パラメーターは NULL にすることができます。

表 250. ビット単位フィールド ID

フィールド名	16 進値	10 進値
REDIST_METHOD	0x0001<<0	1
REDIST_PMAP_FILE	0x0001<<1	2
REDIST_DIST_FILE	0x0001<<2	4
REDIST_STEP_SIZE	0x0001<<3	8
REDIST_NUM_STEPS	0x0001<<4	16
REDIST_STAGE_SIZE	0x0001<<5	32
REDIST_NEXT_STEP	0x0001<<6	64
REDIST_PROCESS_STATE	0x0001<<7	128
REDIST_PWEIGHT_START_NODE	0x0001<<8	256
REDIST_PWEIGHT	0x0001<<9	512

### *redistMethod*

再配分が実行される際に、データ分散ファイルが使用されるか、またはターゲット分散マップが使用されるかを指定する、タイプ SMALLINT の出力引数。戻り値として次の 2 つがあります。

- 2: 再配分処理の入力としてデータ配分ファイルが使用されることを表します。
- 3: 再配分処理の入力としてターゲット分散マップが使用されることを表します。

### *pMapFile*

データベース・サーバー上のターゲット分散マップの絶対パス・ファイル名を指定する、タイプ VARCHAR (255) の出力引数。

### *distFile*

データベース・サーバー上のデータ配分ファイルの絶対パス・ファイル名を指定する、タイプ VARCHAR (255) の出力引数。

### *stepSize*

ログ・フル状態になるのを防止するためのコミット呼び出しの前に移動可能な最大行数を指定する、タイプ BIGINT の出力引数。この数は、各再配分ステップ内で変わることがあります。

### *totalSteps*

このデータベース・パーティション・グループを完全に再配分するために必要なステップ数を指定する、タイプ SMALLINT の出力引数。

### *stageSize*

連続して実行されるステップ数を指定する、タイプ SMALLINT の出力引数。

### *nextStep*

どのステップが完了していて、まだ何を実行する必要があるかを区分する索引を指定する、タイプ SMALLINT の出力引数。

### *processState*

再配分処理が次のチェックポイントで停止するかどうかを指定する、タイプ SMALLINT の出力引数。チェックポイントは、各再配分ステップの先頭に置かれています。この引数が 1 に設定されている場合、ステップは開始されません。値が 0 の場合、ステップは進行します。

### *pNumber*

データベース・パーティション番号がコンマで区切られたストリング・フォーマットのリスト (該当する場合) を戻す、タイプ VARCHAR (6000) の出力引数。これらのパーティション番号は、データベース・パーティション・グループが現在使用しているデータベース・パーティションか、あるいは追加またはドロップされるデータベース・パーティションです。これらのパーティション番号の順序と数は、*pWeight* 変数によって戻されるターゲット・パーティションの重みに対応します。

### *pWeight*

ターゲット・データベース・パーティションの重み数値がコンマで区切られたリスト (該当する場合) を戻す、タイプ VARCHAR (6000) の出力引数。これらのパーティション重みの順序と数は、*pNumber* 変数によって戻されるパーティション番号に対応します。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

指定されたデータベース・パーティション・グループのステップ単位の再配分プランの内容を報告します。

```
CALL SYSPROC.GET_SWRD_SETTINGS  
('IBMDEFAULTGROUP', 255, ?, ?, ?, ?, ?, ?, ?, ?)
```

## 使用上の注意

再配分に関するストアード・プロシージャおよび関数は、各表の分散キーが定義されているパーティション・データベース環境でのみ動作します。

## SET\_SWRD\_SETTINGS プロシージャ - 再配分レジストリーの作成または変更

SET\_SWRD\_SETTINGS プロシージャは、再配分レジストリーを作成または変更します。

レジストリーが存在しない場合は、レジストリーを作成してそれにレコードを追加します。レジストリーがすでに存在する場合は、どのフィールド値に上書きする必要があるかを *overwriteSpec* を使用して識別します。 *overwriteSpec* フィールドを指定することにより、更新する必要のないフィールドについては、この関数の入力に NULL を使用できます。

## 構文

```
▶▶SET_SWRD_SETTINGS—(—dbpgName—,—overwriteSpec—,—redistMethod—,—————▶  
▶—pMapFile—,—distFile—,—stepSize—,—totalSteps—,—stageSize—,—————▶  
▶—nextStep—,—processState—,—pNumber—,—pWeight—)—————▶▶
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *dbpgName*

再配分処理の実行の対象となるデータベース・パーティション・グループ名を指定する、タイプ VARCHAR(128) の入力引数。

### *overwriteSpec*

表 251 に示すビット単位フィールド ID。これで、再配分設定レジストリーに書き込むか上書きするターゲット・フィールドを表します。

表 251. ビット単位フィールド ID

フィールド名	16 進値	10 進値
REDIST_METHOD	0x0001<<0	1
REDIST_PMAP_FILE	0x0001<<1	2
REDIST_DIST_FILE	0x0001<<2	4
REDIST_STEP_SIZE	0x0001<<3	8
REDIST_NUM_STEPS	0x0001<<4	16
REDIST_STAGE_SIZE	0x0001<<5	32
REDIST_NEXT_STEP	0x0001<<6	64
REDIST_PROCESS_STATE	0x0001<<7	128
REDIST_PWEIGHT_START_NODE	0x0001<<8	256
REDIST_PWEIGHT	0x0001<<9	512

### *redistMethod*

再配分を実行する際に、データ分散ファイルを使用するか、またはターゲット分散マップを使用するかを指定する、タイプ `SMALLINT` の入力引数。有効な入力値は、次の 2 つです。

- 2: 再配分処理の入力としてデータ配分ファイルを使用することを表します。
- 3: 再配分処理の入力としてターゲット分散マップを使用することを表します。

### *pMapFile*

データベース・サーバー上のターゲット分散マップの絶対パス・ファイル名を指定する、タイプ `VARCHAR (255)` の入力引数。

### *distFile*

データベース・サーバー上のデータ配分ファイルの絶対パス・ファイル名を指定する、タイプ `VARCHAR (255)` の入力引数。

### *stepSize*

ログ・フル状態になるのを防止するためのコミット呼び出しの前に移動可能な最大行数を指定する、タイプ `BIGINT` の入力引数。この数は、各再配分ステップ内で変わることがあります。 *stepSize* の値に `-2` を使用することにより、この数が無制限であることを表すことができます。

### *totalSteps*

このデータベース・パーティション・グループを完全に再配分するために必要なステップ数を指定する、タイプ `SMALLINT` の入力引数。 *totalSteps* の値に `-2` を使用することにより、この数が無制限であることを表すことができます。

### *stageSize*

連続して実行するステップ数を指定する、タイプ `SMALLINT` の入力引数。

### *nextStep*

どのステップが完了していて、まだ何を実行する必要があるかを区分する索引を指定する、タイプ `SMALLINT` の入力引数。

### *processState*

再配分処理を次のチェックポイントで停止させるかどうかを指定する、タイプ `SMALLINT` の入力引数。チェックポイントは、各再配分ステップの先頭に置かれています。この引数が `1` に設定されている場合、ステップは開始されません。値が `0` の場合、ステップは進行します。

### *pNumber*

データベース・パーティション番号がコンマで区切られたストリング・フォーマットのリストを含めることのできる、タイプ `VARCHAR (6000)` の入力引数。これらのパーティション番号は、データベース・パーティション・グループが現在使用しているデータベース・パーティションか、あるいは追加またはドロップされるデータベース・パーティションです。これらのパーティション番号の順序と数は、*pWeight* 変数によって戻されるターゲット・パーティションの重みに対応します。各データベース・パーティション番号は、`0` から `999` までの範囲の数値です。ストリング内でスペースは使用できません。

### *pWeight*

すべてのデータベース・パーティションのユーザー指定の重みがコンマで区切られたストリングを含めることのできるタイプ `VARCHAR (6000)` の入力引数

で、*pNumber* ストリング内のデータベース・パーティション番号に対応します。各データベース・パーティションの重みは、0 から 32767 までの範囲の数値です。ストリング内でスペースは使用できません。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

ステップ単位の再配分プランをレジストリーに書き込みます。 *processState* を 1 に設定すると、現在実行中のステップ単位の再配分ストアード・プロシージャが現行ステップを完了して停止する場合があります。その場合は、このパラメーターを 0 にリセットして再配分ストアード・プロシージャを再度呼び出すまで停止したままになります。

```
CALL SYSPROC.SET_SWRD_SETTINGS('IBMDEFAULTGROUP', 255, 0, ' ',
    '$HOME/sql1lib/function/TEST.IBMDEFAULTGROUP_swrData.dst', 1000,
    12, 2, 1, 0, '10,20,30', '50,50,50')
```

## 使用上の注意

再配分に関するストアード・プロシージャおよび関数は、各表の分散キーが定義されているパーティション・データベース環境でのみ動作します。

## STEPWISE\_REDISTRIBUTE\_DBPG プロシージャ - データベース・パーティション・グループの一部の再配分

STEPWISE\_REDISTRIBUTE\_DBPG プロシージャは、このプロシージャに指定された入力と、SET\_SWRD\_SETTINGS プロシージャによって作成または更新された設定ファイルに従って、データベース・パーティション・グループの一部を再配分します。

## 構文

```
►►STEPWISE_REDISTRIBUTE_DBPG(—inDBPGGroup—,—inStartingPoint—,——————►  
►—inNumSteps—)—————►►
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *inDBPGroup*

ターゲット・データベース・パーティション・グループの名前を指定する、タイプ VARCHAR (128) の入力引数。

### *inStartingPoint*

使用する開始点を指定する、タイプ SMALLINT の入力引数。このパラメーターが NULL 以外の正の整数に設定されると、STEPWISE\_REDISTRIBUTE\_DBPG プロシージャは、設定ファイルで指定された *nextStep* 値を使用しないで、この値を使用します。このオプションは、STEPWISE\_REDISTRIBUTE\_DBPG プロシージャを特定のステップから再実行するときに役立ちます。このパラメーターを NULL に設定した場合は、*nextStep* 値が使用されます。

### *inNumSteps*

実行するステップ数を指定する、タイプ SMALLINT の入力引数。このパラメーターが NULL 以外の正の整数に設定されると、STEPWISE\_REDISTRIBUTE\_DBPG プロシージャは、設定ファイルで指定された *stageSize* 値を使用しないで、この値を使用します。このオプションは、設定で指定された内容とは異なるステップ数を指定して STEPWISE\_REDISTRIBUTE\_DBPG プロシージャを再実行するときに役立ちます。例えば、スケジュール済みステージに 5 つのステップがあり、再配分処理がステップ 3 で失敗した場合、エラー状態が訂正されたら STEPWISE\_REDISTRIBUTE\_DBPG プロシージャを呼び出すことによって、残りの 3 つのステップを実行することができます。このパラメーターを NULL に設定した場合は、*stageSize* 値が使用されます。このプロシージャに値 -2 を使用することにより、この数が無制限であることを表すことができます。

**注:** REDISTRIBUTE DATABASE PARTITION GROUP コマンドに NOT ROLLFORWARD RECOVERABLE オプションと同等のものを指定するパラメーターはありません。STEPWISE\_REDISTRIBUTE\_DBPG プロシージャの使用時には、行データ再配布に対してロギングが必ず実行されます。

## 許可

- STEPWISE\_REDISTRIBUTE\_DBPG プロシージャに対する EXECUTE 特権
- SYSADM、SYSCTRL、または DBADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

SET\_SWRD\_SETTINGS プロシージャによってレジストリーに保管された再配分プランに従って、データベース・パーティション・グループ「IBMDEFAULTGROUP」を再配分します。データの再配分をステップ 3 から開始して、再配分プランの 2 つのステップを完了します。

```
CALL SYSPROC.STEPWISE_REDISTRIBUTE_DBPG('IBMDEFAULTGROUP', 3, 2)
```



ステップ単位の再配分プロシージャの詳細な使用例については、「パーティションおよびクラスタリングのガイド」の『STEPWISE\_REDISTRIBUTE\_DBPG プロシージャを使用したデータベース・パーティション・グループの再配分』を参照してください。

### 使用上の注意

STEPWISE\_REDISTRIBUTE\_DBPG プロシージャの実行開始後に SET\_SWRD\_SETTINGS プロシージャを使用して *processState* のレジストリー値を 1 に更新すると、処理は次のステップの先頭で停止し、警告メッセージが戻されます。

再配分処理で SQL COMMIT ステートメントが呼び出されるため、タイプ 2 接続での再配分処理の実行はサポートされていません。

## ストレージ管理ツール・ルーチン

### CAPTURE\_STORAGEMGMT\_INFO プロシージャ - 特定ルート・オブジェクトのストレージ関連情報の検索

CAPTURE\_STORAGEMGMT\_INFO プロシージャは、指定されたルート・オブジェクト、およびその有効範囲内の定義済みストレージ・オブジェクトについて、ストレージ関連情報の収集を試みます。すべてのストレージ・オブジェクトは SYSTOOLS.STMG\_OBJECT\_TYPE 表で特定されています。

表 252. STMG\_OBJECT\_TYPE 表

列名	データ・タイプ	NULL 可能	説明
OBJ_TYPE	INTEGER	N	ストレージ・オブジェクトのタイプに対応する整数値 <ul style="list-style-type: none"> <li>• 0 - データベース</li> <li>• 1 - データベース・パーティション・グループ</li> <li>• 2 - 表スペース</li> <li>• 3 - 表スペース・コンテナ</li> <li>• 4 - 表</li> <li>• 5 - 索引</li> </ul>
TYPE_NAME	VARCHAR	N	ストレージ・オブジェクト・タイプの記述名 <ul style="list-style-type: none"> <li>• STMG_DATABASE</li> <li>• STMG_DBPGROUP</li> <li>• STMG_TABLESPACE</li> <li>• STMG_CONTAINER</li> <li>• STMG_TABLE</li> <li>• STMG_INDEX</li> </ul>

## 構文

```
▶▶—CAPTURE_STORAGEMGMT_INFO—(—in_rootType—,—in_rootSchema—,——————▶▶  
▶—in_rootName—)—————▶▶
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *in\_rootType*

タイプ SMALLINT の入力引数。有効なオプション・タイプは、以下のとおりです。

- 0 - データベース
- 1 - データベース・パーティション・グループ
- 2 - 表スペース
- 4 - 表
- 5 - 索引

この入力引数を NULL にすることはできません。NULL 値を指定すると、SQL0443 エラー (SQLSTATE 38553) と、トークン DBA7617 が戻されます。

### *in\_rootSchema*

ストレージ・スナップショットのルート・オブジェクトのスキーマ名を指定する、タイプ VARCHAR (128) の入力引数。 *in\_rootType* がデータベース、データベース・パーティション・グループ、または表スペースである場合は、NULL 値を指定できます。

### *in\_rootName*

ルート・オブジェクトの名前を指定する、タイプ VARCHAR (128) の入力引数。この入力引数を NULL にすることはできません。NULL 値を指定すると、SQL0443 エラー (SQLSTATE 38553) と、トークン DBA7617 が戻されます。

## 許可

- CAPTURE\_STORAGEMGMT\_INFO プロシージャに対する EXECUTE 特権。
- SYSPROC.DB\_PARTITIONS、 SYSPROC.SNAP\_GET\_CONTAINER、 SYSPROC.SNAPSHOT\_CNTRFS 表関数に対する EXECUTE 特権。
- SYSCAT.TABLES、 SYSCAT.TABLESPACES、 SYSCAT.NODEGROUPDEF、 SYSCAT.DATABASEPARTITIONS、 SYSCAT.DATAPARTITIONEXPRESSION、 SYSCAT.INDEXES、 SYSCAT.COLUMNS に対する SELECT 特権。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 使用上の注意:

1. 以下のストアド・プロシージャを使用してストレージ管理表を作成する必要があります。

`create_storagemgmt_tables(TABLESPACE_NAME)` (この 'TABLESPACE' は、ストレージ管理表の作成場所となる表スペースの名前です)。

既存のストレージ管理表で問題が発生した場合は、そのストレージ管理表を下記のストアード・プロシージャを使用してドロップした後、上記のストアード・プロシージャを使用して再作成できます。

`drop_storagemgmt_tables(0 または 1)` ここで '0' はエラー検出時に「停止」、'1' は「続行」をそれぞれ示します。

2. 以下のコマンドを使用して、詳細を取得する対象となるストレージ・オブジェクトに関する統計を実行する必要があります。

`RUNSTATS ON TABLE (TABLESCHEMA.TABLENAME) ON KEY COLUMNS AND INDEXES ALL`

3. 以下のコマンドを使用して、ストレージ管理表にデータを入れる必要があります。

ストレージ管理表にデータを入れるために 'capture\_storagemgmt\_info()' ストアード・プロシージャを実行します。場合によっては、`CAPTURE_STORAGEMGMT_INFO` プロシージャを 2 度実行する必要があることがあります。最初の実行では、ストレージ表に表スペースの詳細を取り込むために `CAPTURE_STORAGEMGMT_INFO` プロシージャを使用します。以下に例を示します。

```
db2 "call capture_storagemgmt_info(0,<SCHEMA_NAME>,<DATABASE_NAME>)"
```

2 度目は、実際のオブジェクトのストレージに関する詳細をストレージ表に追加するために `CAPTURE_STORAGEMGMT_INFO` プロシージャを使用します。例えば、以下の例は、索引タイプのオブジェクトの詳細を追加します (`in_rootType` 引数は 5 に設定されています)。

```
db2 "call capture_storagemgmt_info  
(5,<SCHEMA_NAME>,<SCHEMA_NAME.INDEX_NAME>)"
```

4. 必要なストレージ管理表に対する `SELECT` 照会を実行して、ストレージ・オブジェクトの詳細を表示します。例えば `INDEX` オブジェクトの場合には次のようになります。

```
db2 "SELECT * FROM SYSTOOLS.STMG_INDEX"
```

## CREATE\_STORAGEMGMT\_TABLES プロシージャ - ストレージ管理表の作成

`CREATE_STORAGEMGMT_TABLES` プロシージャは、入力で指定された表スペースに、「DB2TOOLS」固定スキーマ下のすべてのストレージ管理表を作成します。

### 構文

```
►►—CREATE_STORAGEMGMT_TABLES—(—in_tbspace—)—————►►
```

スキーマは `SYSPROC` です。

## プロシージャ・パラメーター

*in\_tbspace*

表スペース名を指定する、タイプ VARCHAR(128) の入力引数。この入力引数を NULL にすることはできません。NULL 値を指定すると、SQL0443 エラー (SQLSTATE 38553) と、トークン DBA7617 が戻されます。

## 許可

CREATE\_STORAGEMGMT\_TABLES プロシージャに対する EXECUTE 特権。

この他に、データベースに対する CREATETAB 特権および表スペースに対する USE 特権と、以下のいずれかが必要です。

- データベースに対する IMPLICIT\_SCHEMA 権限 (暗黙または明示のスキーマ名 DB2TOOLS が存在しない場合)
- スキーマに対する CREATEIN 特権 (表のスキーマ名が存在する場合)
- DBADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 使用上の注意

以下の表が DB2TOOLS スキーマで作成されます。

- STMG\_CONTAINER
- STMG\_CURR\_THRESHOLD
- STMG\_DATABASE
- STMG\_DBPARTITION
- STMG\_DBPGROUP
- STMG\_HIST\_THRESHOLD
- STMG\_INDEX
- STMG\_OBJECT
- STMG\_OBJECT\_TYPE
- STMG\_ROOT\_OBJECT
- STMG\_TABLE
- STMG\_TABLESPACE
- STMG\_TBPARTITION
- STMG\_THRESHOLD\_REGISTRY

## DROP\_STORAGEMGMT\_TABLES プロシージャ - すべてのストレージ管理表のドロップ

DROP\_STORAGEMGMT\_TABLES プロシージャは、すべてのストレージ管理表のドロップを試みます。

## 構文

▶▶—DROP\_STORAGEMGMT\_TABLES—(—*dropSpec*—)—————▶▶

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *dropSpec*

タイプ SMALLINT の入力引数。 *dropSpec* を 0 に設定した場合、エラーが検出されると処理を停止します。 *dropSpec* を 1 に設定した場合、検出したエラーを無視して処理を続行します。この入力引数を NULL にすることはできません。 NULL 値を指定すると、SQL0443 エラー (SQLSTATE 38553) と、トークン DBA7617 が戻されます。

### 許可

DROP\_STORAGEMGMT\_TABLES プロシージャに対する EXECUTE 特権。

データベース接続を確立するユーザー ID は、ストレージ管理表の定義者 (SYSCAT.TABLES の DEFINER 列に記録されている) であるか、または以下の特権の少なくとも 1 つを持っていないければなりません。

- DBADM 権限
- これらの表のスキーマに対する DROPIN 特権
- これらの表に対する CONTROL 特権

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## テキスト検索ルーチン

### SYSTS\_ADMIN\_CMD プロシージャ - テキスト検索管理コマンドの実行

SYSTS\_ADMIN\_CMD プロシージャは、SQL CALL ステートメントを使用してテキスト検索管理コマンドを実行するアプリケーションで使用されます。

### 許可

SYSTS\_ADMIN\_CMD プロシージャに対する EXECUTE 特権、および要求される操作作用にリストされている必要な権限。

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 構文

→SYSYS\_ADMIN\_CMD(—command-string—, —message-locale—, —message—)←

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *command-string*

実行される単一の DB2 Text Search 索引管理コマンドを指定する、タイプ VARCHAR (32 K) の入力引数。コマンド構文は、このプロシージャではサポートされていない接続オプションを除き、DB2 Text Search コマンドと同じです。このプロシージャから発行されるコマンドは、現在接続を使用します。

このプロシージャは、以下の DB2 Text Search コマンドをサポートします。

- ALTER INDEX
- CLEAR COMMAND LOCKS
- CLEAR EVENTS
- CREATE INDEX
- DISABLE DATABASE
- DROP INDEX
- ENABLE DATABASE
- RESET PENDING
- UPDATE INDEX

#### *message-locale*

戻されるすべてのエラー・メッセージ・テキスト用に使用される言語を指定する、タイプ VARCHAR(33) の入力引数。引数が NULL または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、'en\_US' が使用されます。

#### *message*

正常と見なされる操作に対する警告または通知メッセージを指定する、タイプ VARCHAR(32K) の出力引数。

## 例

以下の例は、スキーマ DB2TS の MYTEXTINDEX テキスト検索索引を更新し、すべてのエラー・メッセージを英語で戻します。

```
CALL SYSPROC.SYSYS_ADMIN_CMD  
('UPDATE INDEX DB2TS.MYTEXTINDEX FOR TEXT','en_US', ?)";
```

出力例を以下に示します。

```
Value of output parameters  
-----  
Parameter Name : MESSAGE  
Parameter Value : C1E00001 Operation completed successfully.  
  
Return Status = 0
```

## 使用上の注意

コマンドが正常に実行されない場合、SQLCODE -20427 および SQLSTATE 38H14 がテキスト検索特有のエラー・メッセージとともに戻されます。例えば、索引 MYTEXTINDEX が存在しており、以下のステートメントが発行された場合:

```
CALL SYSPROC.SYSTS_ADMIN_CMD ('CREATE INDEX MYTEXTINDEX FOR TEXT
ON DB2TS.TEXTBOOKS (STORY)', 'en_US', ?)
```

索引の作成が、次のエラー・メッセージで失敗します。

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message is "CIE00201 Text search
index "DB2TS ". "MYTEXTINDEX" already exists. ". SQLSTATE=38H14
```

SQLCODE がプロシージャにより戻される場合、メッセージは切り捨てられる可能性があります。詳細なメッセージ情報は **db2diag** ログ・ファイル内にあります。

## SYSTS\_ALTER プロシージャ - 索引の更新特性の変更

このプロシージャは、索引の更新特性を変更します。

このプロシージャは **ALTER INDEX** テキスト検索管理コマンドをデータベース・サーバー上で発行します。

### 許可

ステートメントの許可 ID が保持する特権には、SYSTS\_MGR ロールと以下の少なくとも 1 つの権限が含まれている必要があります。

- DBADM 権限
- 基本スキーマに対する ALTERIN 特権
- そのテキスト検索索引が定義されている基本表に対する CONTROL または ALTER 特権

既存のスケジュールを変更するには、許可 ID が索引の作成者と同じであるか、DBADM 権限が含まれている必要があります。

### デフォルトの PUBLIC 特権

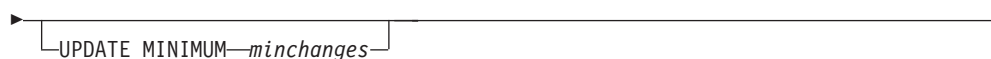
なし

### 構文

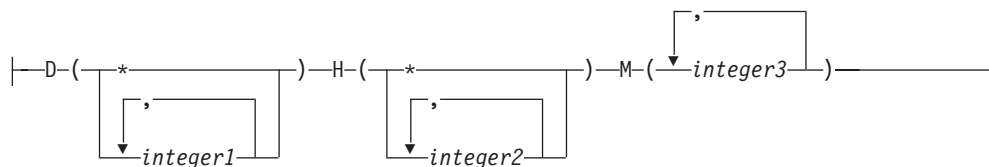
```
►► SYSTS_ALTER (—index_schema—, —index_name—, — update characteristics |—►
► | options |, —message_locale—, —message—) ◀◀
```

#### update characteristics:

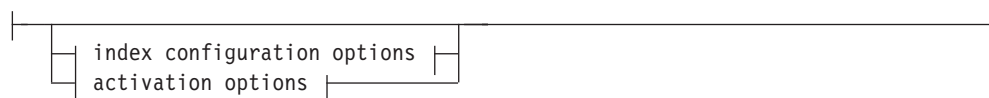
```
|—►
| | UPDATE FREQUENCY |—►
| | | NONE |—►
| | | update frequency |—►
```



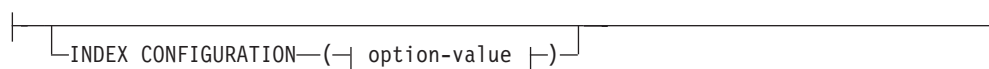
#### update frequency:



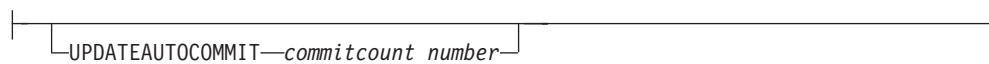
#### options:



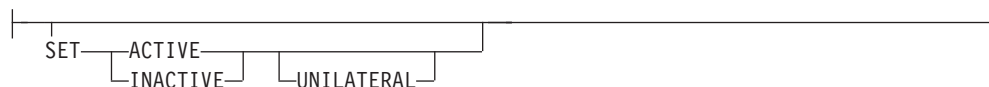
#### index configuration options:



#### option-value:



#### activation options:



スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *index\_schema*

テキスト検索索引のスキーマを指定する、タイプ VARCHAR(128) の入力引数。 *index\_schema* は、DB2 スキーマ名の命名上の制約に従う必要があります。引数が NULL または空ストリングである場合、値 CURRENT SCHEMA が使用されます。 *index\_schema* には、大文字小文字の区別があります。

#### *index\_name*

索引の名前を指定する、タイプ VARCHAR (128) の入力引数。 *index\_schema* とともに使用すると、データベースのテキスト検索索引を一意的に識別します。 *index\_name* には、大文字小文字の区別があります。



## update characteristics

変更オプションを指定する、タイプ VARCHAR(32K) の入力引数。指定可能な変更オプションは以下のとおりです。

### UPDATE FREQUENCY

索引の更新が行われる頻度を指定します。変更回数が **UPDATE MINIMUM** に設定された値以上である場合、索引は更新されます。更新頻度 **NONE** は、索引の更新はそれ以降行われないことを意味します。これは、変更されないデータを持つ表のテキスト列に役立ちます。さらに、これはユーザーが (**UPDATE INDEX** コマンドを使用して) 手動で索引を更新する予定の場合にも役立ちます。自動更新を行えるのは、DB2\_ATS\_ENABLE レジストリー変数が設定されていて、**START FOR TEXT** コマンドが発行される場合のみです。

デフォルトの頻度値はビュー SYSIBMTS.TSDEFAULTS から取られ、DEFAULTNAME='UPDATEFREQUENCY' となります。

### NONE

自動更新は、テキスト索引に適用されません。以降のすべての索引更新は手動で開始する必要があります。

### D 索引が更新される曜日。

\* 毎日。

*integer1*

日曜日から土曜日の特定の曜日: 0 から 6

### H 索引が更新される特定の日の時。

\* 毎時。

*integer2*

午前 0 時から午後 11 時までの特定の時: 0 から 23

### M 索引が更新される特定の時刻の分。

*integer3*

**UPDATE FREQUENCY** オプションを指定しない場合、頻度の設定は変更されないままとなります。

### UPDATE MINIMUM *minchanges*

索引が増分的に更新される前に生じなければならない、テキスト文書への変更の最小数を指定します。同じテキスト文書への複数の変更は、別個の変更として扱われます。**UPDATE MINIMUM** オプションを指定しない場合、設定は変更されないままとなります。

### INDEX CONFIGURATION (*option-value*)

バージョン 9.7 フィックスパック 3 以降のフィックスパックでは、これは、テキスト索引構成設定を変更できる VARCHAR(32K) タイプのオプションの入力引数です。以下のオプションがサポートされています。

表 253. オプション値の仕様

オプション	値	データ・タイプ	説明
SERIALUPDATE	<i>updatemode</i>	整数	<p>パーティション化されたテキスト検索索引の更新処理を並列モードと直列モードのどちらで行うかを指定します。並列モードの場合、実行が複数のデータベース・パーティションに分散され、それぞれのノードで独立して行われます。直列モードでは、実行は分散されずに行われ、障害が検出されると停止します。通常、直列モード実行の方が時間はかかりますが、必要とするリソースがかなり少なく済みます。</p> <ul style="list-style-type: none"> <li>• 0 = 並列モード</li> <li>• 1 = 直列モード</li> </ul>

表 253. オプション値の仕様 (続き)

オプション	値	データ・タイプ	説明
UPDATEAUTO COMMIT	<i>commitcount</i> <i>_number</i>	整数	<p>ここで指定された回数の索引更新が行われると、コミットが実行され、イニシャル更新またはインクリメンタル更新のためにそれまでの作業が自動的に保存されます。</p> <ul style="list-style-type: none"> <li>イニシャル更新では、データ更新をキャプチャーするトリガーがアクティブ化された後の基本表からの文書のバッチを、索引更新が処理します。文書更新の回数が COMMITCOUNT に達すると、サーバーは暫定コミットを実行します。処理されていない文書により生成されたログ項目は、ステージング表から除去されます。イニシャル・テキスト索引更新に UPDATEAUTOCOMMIT オプションを使用すると、実行時間が大幅に増加します。</li> <li>インクリメンタル更新では、処理されているログ項目は、各暫定コミットで対応するステージング表から除去されます。COMMITCOUNT は更新される文書の数であり、ステージング表項目の数ではありません。</li> </ul>

*activation options*

バージョン 9.7 フィックスパック 3 以降のフィックスパックでは、この整数タイプの入力引数はテキスト索引の状況を設定します。

**ACTIVE**

テキスト索引状況をアクティブに設定します

**INACTIVE**

テキスト索引状況を非アクティブに設定します

## UNILATERAL

DB2 Text Search 索引の状況にのみ影響する UNILATERAL 変更を指定します。この引数が指定された場合、DB2 Text Search 索引の状況だけがアクティブまたは非アクティブに変更されます。UNILATERAL 引数がない場合、DB2 Text Search 索引および DB2 Net Search Extender 索引のアクティブ化の状況はともに切り替えられ、テキスト索引の一方だけがアクティブ化されます。

### *message\_locale*

戻されるすべてのエラー・メッセージに使用されるロケールを指定する、タイプ VARCHAR(33) の入力引数。引数が NULL または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、'en\_US' が使用されます。

### *message*

正常に完了した操作の警告または通知メッセージを指定する、タイプ VARCHAR(32K) の出力引数。

## 例

例 1: 以下の例では、テキスト検索索引の更新特性が変更されます。この索引は、最初は *index\_schema* 'db2ts' および *index\_name* 'myTextIndex' で作成されています。関連する表列に変更が加えられないことが予想されるため、テキスト検索索引にそれ以上の更新をしないという意図で UPDATE FREQUENCY NONE を使用します。すべてのエラー・メッセージは英語で戻すことが要求されます。このプロシージャが成功すると、操作の成功を示す出力パラメーター・メッセージが呼び出し元に戻されます。

```
CALL SYSPROC.SYSTS_ALTER('db2ts', 'myTextIndex',  
'UPDATE FREQUENCY NONE', 'en_US', ?)
```

以下はこの照会の出力例です。

```
Value of output parameters  
-----  
Parameter Name : MESSAGE  
Parameter Value : Operation completed successfully.  
  
Return Status = 0
```

例 2: 以下の例では、SYSTS\_ALTER ストアード・プロシージャが呼び出され、*index\_schema* 'db2ts' および *index\_name* 'myTextIndex2' があるテキスト検索索引の更新特性が変更されます。この意図は、索引への更新が毎時行われるということです。ただし、この索引は存在せず、結果としてエラーを戻します。

```
CALL SYSPROC.SYSTS_ALTER('db2ts', 'myTextIndex2',  
'update frequency D(*) H(*) M(0)', 'en_US', ?)
```

以下はこの照会の出力例です。

```
SQL20427N An error occurred during a text search administration  
procedure or command. The error message is "CIE00316 Text search  
index "db2ts"."myTextIndex2" does not exist. ". SQLSTATE 38H14
```

## 使用上の注意

- テキスト検索管理プロシージャは、データベースへの既存の接続を使用します。テキスト検索管理プロシージャ内のコミットまたはロールバックからの予

期しない影響を避けるため、プロシージャーを実行する前に、すべてのトランザクション変更をコミットすることをお勧めします。これを行う 1 つの方法は、AUTOCOMMIT をオンにすることです。

- 複数のプロシージャーまたはコマンドは、競合する可能性がある場合は、テキスト検索索引に対して同時に実行することはできません。競合するプロシージャーおよびコマンドの一部は以下のとおりです。
  - SYSTS\_ALTER プロシージャーまたは ALTER INDEX db2ts コマンド
  - SYSTS\_CLEAR\_EVENTS プロシージャーまたは CLEAR EVENTS FOR INDEX db2ts コマンド
  - SYSTS\_DISABLE プロシージャーまたは DISABLE DATABASE FOR TEXT db2ts コマンド
  - SYSTS\_DROP プロシージャーまたは DROP INDEX db2ts コマンド
  - SYSTS\_UPDATE プロシージャーまたは UPDATE INDEX db2ts コマンド
  - SYSTS\_CONFIGURE プロシージャー

競合がある場合、プロシージャーは SQLCODE -20426 および SQLSTATE 38H13 を戻します。

- 頻度を変更するためにこのプロシージャーを実行する場合、テキスト索引用にスケジュール・タスクが作成、変更、またはドロップされます。
- 索引をアクティブにした結果は、元の索引の状況により異なります。次の表は、結果を示したものです。

表 254. 無効索引無し of 状況変更:

イニシャルの DB2 Text Search または Net Search Extender の状況	ACTIVE 要求	ACTIVE UNILATERAL 要求	INACTIVE 要求	INACTIVE UNILATERAL 要求
アクティブ / 非アクティブ	変更なし	変更なし	非アクティブ / アクティブ	非アクティブ / 非アクティブ
非アクティブ / アクティブ	アクティブ / 非アクティブ	エラー	変更なし	変更なし
非アクティブ / 非アクティブ	アクティブ / 非アクティブ	アクティブ / 非アクティブ	非アクティブ / アクティブ	変更なし

SQL20427N および CIE0379E エラー・メッセージが、索引のアクティブ化競合の場合に戻されます。

### SYSTS\_CLEANUP プロシージャー - 無効なテキスト検索索引の削除

このプロシージャーは、無効なテキスト検索索引とそれに関連するコレクションをデータベースから削除します。表のすべての内容に影響するデータベース操作 (例えば、切り捨て) が実行されると、索引は無効になります。

このプロシージャーは、データベースを有効範囲とする場合に **db2ts CLEANUP FOR TEXT** コマンドの代替として機能します。インスタンス・レベルでの廃止コレクションを削除するには、コマンドを使用する必要があります。

## 許可

このステートメントの許可 ID が持つ特権に、DBADM および DATAACCESS 権限のある SYSTS\_ADM ロールが含まれている必要があります。

## デフォルトの PUBLIC 特権

なし

## 構文

```
▶▶ SYSTS_CLEANUP(—options—, —message_locale—, —message—) ▶▶
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *options*

タイプ VARCHAR(32K) の入力引数。 NULL または空パラメーターのみが受け入れられます。 IBM の内部使用のために予約済みです。

### *message\_locale*

戻されるすべてのエラー・メッセージに使用されるロケールを指定する、タイプ VARCHAR(33) の入力引数。引数が NULL または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、'en\_US' が使用されます。

### *message*

正常に完了した操作の警告または通知メッセージを指定する、タイプ VARCHAR(32K) の出力引数。

## 例

例 1: 次の例では、現行データベース内の無効なテキスト検索索引がすべてドロップされます。メッセージ言語が英語に設定され、プロシージャが成功すると、操作の成功を示す出力パラメーター・メッセージが呼び出し元に戻されます。

```
CALL SYSPROC.SYSTS_CLEANUP('', 'en_US', ?)
```

Value of output parameters

```
-----  
Parameter Name : MESSAGE  
Parameter Value : Operation completed successfully.
```

Return Status = 0

例 2: エラーが戻される出力:

```
SQL0462W Command or routine "SYSTS_CLEANUP" (specific name "*N") has returned a  
warning SQLSTATE, with diagnostic text "CIE00212W 2 of 20 collections could not be  
deleted. Check db2diag.log for details.". SQLSTATE=01H14
```

## SYSTS\_CLEAR\_COMMANDLOCKS プロシージャ - テキスト検索索引のコマンド・ロックの削除

このプロシージャは、データベース内の特定のテキスト検索索引またはすべてのテキスト検索索引のすべてのコマンド・ロックを解除します。

## 許可

索引のロックをクリアするために使用されるステートメントの許可 ID が保持する特権には、以下の両方の権限が含まれている必要があります。

- SYSTS\_MGR ロール
- DBADM 権限またはその索引が定義されている基本表に対する CONTROL 特権

データベース接続のロックをクリアするために使用されるステートメントの許可 ID が保持する特権には、SYSTS\_ADM ロールが含まれている必要があります。

## デフォルトの PUBLIC 特権

なし

コマンド・ロックはテキスト検索索引コマンドの開始時に作成され、コマンドの完了時に破棄されます。それにより、異なるコマンド間の望ましくない競合が避けられます。まれなケースとして、予期しないシステム動作によってロックがそのまま残り、明示的にクリーンアップする必要があるときに、このプロシージャの使用が必要とされます。

このプロシージャは、データベース・サーバーで **CLEAR COMMAND LOCKS** テキスト検索管理コマンドを発行します。

## 構文

```
►►—SYSTS_CLEAR_COMMANDLOCKS—(—index_schema—,—index_name—,——————►  
►—message_locale—,—message—)——————►◄
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *index\_schema*

テキスト索引のスキーマを指定する、タイプ VARCHAR(128) の入力引数。

*index\_schema* は、DB2 スキーマ名の命名上の制約に従う必要があります。引数が NULL または空ストリングである場合、値 CURRENT SCHEMA が使用されます。 *index\_schema* には、大文字小文字の区別があります。

### *index\_name*

索引の名前を指定する、タイプ VARCHAR (128) の入力引数。 *index\_schema*

とともに使用すると、データベースのテキスト検索索引を一意的に識別します。引数が NULL または空ストリングである場合、このプロシージャはデータベース内のすべてのテキスト検索索引のコマンド・ロックを削除します。

*index\_name* には、大文字小文字の区別があります。

### *message\_locale*

戻されるすべてのエラー・メッセージに使用されるロケールを指定する、タイプ VARCHAR(33) の入力引数。引数が NULL または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、'en\_US' が使用されます。

## message

正常に完了した操作の警告または通知メッセージを指定する、タイプ VARCHAR(32K) の出力引数。

## 例

例 1: 以下の例では、SYSTS\_CLEAR\_COMMANDLOCKS が、*index\_schema* 'db2ts' および *index\_name* 'myTextIndex' があるテキスト検索索引に対して発行されます。エラー・メッセージは英語で戻ることが要求されます。このプロシージャが成功すると、操作の成功を示す出力パラメーター・メッセージが呼び出し元に戻されます。

```
CALL SYSPROC.SYSTS_CLEAR_COMMANDLOCKS('db2ts', 'myTextIndex', 'en_US', ?)
```

以下はこの照会の出力例です。

```
Value of output parameters
-----
Parameter Name : MESSAGE
Parameter Value : Operation completed successfully.

Return Status = 0
```

例 2: 以下の例では、*index\_schema* 'db2ts' および *index\_name* 'myTextIndex2' があるテキスト検索索引のコマンド・ロックをクリアするために、SYSTS\_CLEAR\_COMMANDLOCKS が呼び出されます。この索引は存在せず、プロシージャはエラー・メッセージを戻します。

```
CALL SYSPROC.SYSTS_CLEAR_COMMANDLOCKS('db2ts', 'myTextIndex2', 'en_US', ?)
```

以下はこの照会の出力例です。

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message is "CIE00316 Text search
index "db2ts"."myTextIndex2" does not exist. ". SQLSTATE 38H14
```

## 使用上の注意

- テキスト検索管理プロシージャは、データベースへの既存の接続を使用します。テキスト検索管理プロシージャにおけるコミットまたはロールバックからの予期しない影響を避けるために、そのプロシージャを実行する前に、すべてのトランザクション変更をコミットすることをお勧めします。これを行う 1 つの方法は、AUTOCOMMIT をオンにすることです。
- コマンド・ロックを所有するプロセスが非活動であるため、このプロシージャを呼び出す場合があります。このケースでは、(ロックにより表される) コマンドが完了していない場合や、索引が操作できない場合があります。適切なアクションを取る必要があります。例えば、DROP INDEX コマンドを実行している処理が突然停止したとします。いくつかの索引データは削除されましたが、すべてのカタログおよびコレクション情報が削除された訳ではありません。コマンド・ロックは、そのまま残されています。DROP INDEX コマンド・ロックをクリアした後に、SYSTS\_DROP プロシージャを再実行することができます。別の例として、UPDATE INDEX コマンドを実行しているプロセスが中断したとします。一部の文書は処理済みであるものの、すべてが処理されたわけではなく、依然としてコマンド・ロックが有効になっています。テキスト検索索引の状況を検討し、UPDATE INDEX コマンド・ロックをクリアした後に、UPDATE INDEX コマンドを再実行できます。



- このプロシージャーを実行すると、DB2 テキスト検索ビュー SYSIBMTS.TSLOCKS の内容は更新されます。

## SYSTS\_CLEAR\_EVENTS プロシージャー - 索引のイベント表からの索引付けイベントの削除

このプロシージャーは、管理に使用される索引のイベント表から、索引付けイベントを削除します。

イベント表の名前は、列 EVENTVIEWNAME のビュー SYSIBMTS.TSINDEXES にあります。少なくとも 1 つの文書処理する索引の更新操作が行われるごとに、通知項目と、場合によってはエラー項目がイベント表に生成されます。自動更新の場合、イベント表は定期的に検査する必要があります。文書固有のエラーは、文書内容を変更することで訂正する必要があります。エラーを訂正した後、イベントをクリアすることができます (スペースを消費しすぎないために、クリアすべきです)。

このプロシージャーは **CLEAR EVENTS FOR INDEX** テキスト検索管理コマンドをデータベース・サーバー上で発行します。

### 許可

このステートメントの許可 ID には、以下の両方の特権が含まれている必要があります。

- SYSTS\_MGR ロール
- その索引が定義されている表に対する DATAACCESS と DBADM を併せ持つ権限または CONTROL 特権

### デフォルトの PUBLIC 特権

なし

### 構文

```
►►—SYSTS_CLEAR_EVENTS—(—index_schema—,—index_name—,——————►
►—message_locale—,—message—)—————►►
```

スキーマは SYSPROC です。

### プロシージャー・パラメーター

#### *index\_schema*

テキスト検索索引のスキーマを指定する、タイプ VARCHAR(128) の入力引数。 *index\_schema* は、DB2 スキーマ名の命名上の制約に従う必要があります。引数が NULL または空ストリングである場合、値 CURRENT SCHEMA が使用されます。 *index\_schema* には、大文字小文字の区別があります。

#### *index\_name*

索引の名前を指定する、タイプ VARCHAR (128) の入力引数。 *index\_schema* とともに使用すると、データベースのテキスト検索索引を一意的に識別します。 *index\_name* には、大文字小文字の区別があります。

### *message\_locale*

戻されるすべてのエラー・メッセージに使用されるロケールを指定する、タイプ VARCHAR(33) の入力引数。引数が NULL または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、'en\_US' が使用されます。

### *message*

正常に完了した操作の警告または通知メッセージを指定する、タイプ VARCHAR(32K) の出力引数。

## 例

例 1: 以下の例では、*index\_schema* 'db2ts' および *index\_name* 'myTextIndex' で作成されたテキスト検索索引に対して SYSTS\_CLEAR\_EVENTS が呼び出されます。すべてのエラー・メッセージは英語で戻すことが要求されます。このプロシージャが成功すると、操作の成功を示す出力パラメーター・メッセージが呼び出し元に戻されます。

```
CALL SYSPROC.SYSTS_CLEAR_EVENTS('db2ts', 'myTextIndex', 'en_US', ?)
```

以下はこの照会の出力例です。

```
Value of output parameters
-----
Parameter Name   : MESSAGE
Parameter Value  : Operation completed successfully.

Return Status = 0
```

例 2: 以下の例では、SYSTS\_CLEAR\_EVENTS が呼び出され、*index\_schema* 'db2ts' および *index\_name* 'myTextIndex2' があるテキスト検索索引のイベント表項目がクリアされます。この索引は存在せず、結果としてエラーを戻します。

```
CALL SYSPROC.SYSTS_CLEAR_EVENTS('db2ts', 'myTextIndex2', 'en_US', ?)
```

以下はこの照会の出力例です。

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message is "CIE00316 Text search
index "db2ts"."myTextIndex2" does not exist. ". SQLSTATE 38H14
```

## 使用上の注意

- テキスト検索管理プロシージャは、データベースへの既存の接続を使用します。現行トランザクションは、プロシージャの完了結果に応じてコミットされるかまたはロールバックされる場合があります。そのため、そのようなコミットまたはロールバックからの予期しない影響を避けるために、すべてのトランザクション変更をコミットすることができます。これを行う 1 つの方法は、AUTOCOMMIT をオンにすることです。
- 複数のプロシージャまたはコマンドは、競合する可能性がある場合は、テキスト検索索引に対して同時に実行することはできません。競合するプロシージャおよびコマンドの一部は以下のとおりです。
  - SYSTS\_ALTER プロシージャまたは ALTER INDEX db2ts コマンド
  - SYSTS\_DISABLE プロシージャまたは DISABLE DATABASE FOR TEXT db2ts コマンド
  - SYSTS\_DROP プロシージャまたは DROP INDEX db2ts コマンド

- SYSTS\_CONFIGURE プロシージャ
  - SYSTS\_UPDATE プロシージャまたは UPDATE INDEX db2ts コマンド
- 競合がある場合、プロシージャは SQLCODE -20426 および SQLSTATE 38H13 を戻します。
- 定期的な更新がスケジュールされている場合 (SYSTS\_CREATE または SYSTS\_ALTER プロシージャの UPDATE FREQUENCY オプションを参照)、イベント表を定期的に検査する必要があります。
  - テキスト検索索引の DB2 テキスト検索イベントをクリーンアップするには、イベントの理由を検査し、エラーの原因を除去した後に、SYSTS\_CLEAR\_EVENTS プロシージャまたは **CLEAR EVENTS FOR INDEX db2ts** コマンドを使用します。
  - イベント表で参照されるすべての行に変更が加えられていることを確認します。ユーザー表内の行を変更することで、SYSTS\_UPDATE プロシージャまたは **UPDATE INDEX db2ts** コマンドを再度実行するときに、エラーがある文書に索引付けが再試行されます。
  - このコマンドの発行時に、イベント表はクリアされます。

## SYSTS\_CONFIGURE プロシージャ - 現行データベースをテキスト検索用に構成する

**SYSTS\_CONFIGURE** プロシージャは、テキスト検索サーバー接続情報をテキスト検索カタログに適用します。

テキスト検索サーバーの一部のプロパティを、テキスト検索管理表 (テキスト検索カタログ) に反映させる必要があります。これらのプロパティは、索引の作成時にテキスト索引に関連付けられます。トークンなどのパラメーターが定期的に更新される場合は、変更をデータベースに反映させて、すべての索引のプロパティを更新する必要もあります。

このプロシージャは、以下の場合に最初に必要となります。

- 不完全な使用可能化
- スタンドアロン・テキスト検索サーバーのセットアップ
- パーティション・データベース
- および後続のテキスト検索サーバー接続情報の更新

後続の更新の場合には、アクティブなテキスト検索管理操作がないことを確認して、現在構成されているテキスト検索サーバーをシャットダウンしておく必要があります。

データベースの使用可能化のときに、統合テキスト検索サーバーの接続情報を使って SYSIBMTS.TSSERVER 管理ビューが更新されます。SYSIBMTS.TSSERVER ビューのテキスト・サーバー情報を確認し、該当するテキスト検索サーバー・データを使って更新した後、SYSTS\_CONFIGURE プロシージャを実行して更新済み情報を適用してください。インスタンス内にデータベースが複数ある場合は、同じテキスト検索サーバーの情報を使って各データベースを構成します。

一般的に、操作の順序は以下のようになります。

1. テキスト検索サーバーを構成します。インストール時または DB2 インスタンスの作成時に、統合テキスト検索サーバーを構成できます。スタンドアロン・テキスト検索サーバーは、別個に構成されます。
2. **db2ts ENABLE** コマンド、あるいは **ENABLE** オプションを指定した **SYSTS\_ENABLE** または **SYSTS\_ADMIN\_CMD** プロシージャを使用して、データベースをテキスト検索可能にします。
3. **SYSIBMTS.TSSERVERS** ビューでの必要に応じて、テキスト検索サーバーのパラメーターを決定します。
4. **SYSIBMTS.TSSERVERS** 管理ビューをテキスト検索サーバーのパラメーターで更新します。
  - ビューが空の場合は、**INSERT** ステートメントを使用します。以下に例を示します。
 

```
INSERT INTO SYSIBMTS.TSSERVERS (HOST, PORT, TOKEN, SERVERSTATUS)
VALUES ('localhost', 55000, '9kfsjg48=', 0);
```
  - ビューに既に行が含まれている場合は、**SQL UPDATE** ステートメントを使用します。以下に例を示します。
 

```
UPDATE SYSIBMTS.TSSERVERS SET (HOST, PORT, TOKEN) =
('tsmach1.ibm.com', 55002, 'k3j4fjk9u=')
```
5. **SYSTS\_CONFIGURE** プロシージャを実行します。

## 許可

このステートメントの許可 ID が持つ特権に、**SYSTS\_ADM** ロールが含まれている必要があります。

## デフォルトの PUBLIC 特権

なし

## 構文

▶▶—SYSTS\_CONFIGURE—(—options—,—message-locale—,—message—)————▶▶

スキーマは **SYSPROC** です。

## プロシージャ・パラメーター

### *options*

使用するオプションを指定する、タイプ **VARCHAR(32K)** の入力引数。オプションを必要としない場合は、引数を **NULL** にするか、または空ストリングにすることができます。

### *message-locale*

戻されるすべてのエラー・メッセージに使用されるロケールを指定する、タイプ **VARCHAR(33)** の入力引数。引数が **NULL** または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、**en\_US** が使用されます。

### message

成功したと見なされる操作に対する警告または通知メッセージを指定する、タイプ VARCHAR(32K) の出力引数。

### 例

例 1: データベースをテキスト検索サービスのために準備し、すべてのメッセージを英語で戻します。

```
CALL SYSTS_ENABLE('', 'en_US', ?)"

INSERT INTO SYSIBMTS.TSSERVERS
(HOST,PORT,TOKEN,KEY,LOCALE,SERVERTYPE,SERVERSTATUS)
VALUES ('tsmach1.ibm.com', 55000, '9kfsjg48=', 'en_US', 0);

CALL SYSPROC.SYSTS_CONFIGURE('', 'en_US', ?)
```

この照会の出力例:

```
Value of output parameters
-----
Parameter Name : MESSAGE
Parameter Value : Operation completed successfully.

Return Status = 0
```

### 使用上の注意

- データベースがテキスト検索可能に設定されるとき、SYSIBMTS.TSSERVERS ビューが作成されます。このビューは、データベースの使用可能化時に統合テキスト検索サーバーについての情報で更新されます。テキスト検索サーバーの構成を取得できない場合、使用可能化の操作は「不完全な使用可能化」警告と共に終了します。
- SYSIBMTS.TSSERVERS で行の挿入や更新が行われた場合は、必ず SYSTS\_CONFIGURE プロシージャを発行する必要があります。SYSIBMTS.TSSERVERS を更新する前に、アクティブなテキスト検索管理操作がないことを確認して、テキスト検索サーバーをシャットダウンする必要があります。
- データベース内の SYSIBMTS.TSSERVERS を更新するときは、すべてのテキスト検索可能データベースを同じパラメーターで更新する必要があります。1 つの DB2 インスタンスでサポートされるテキスト検索サーバーは 1 つだけです。
- データベースに対して SYSTS\_CONFIGURE プロシージャを実行すると、インスタンスでの構成済みテキスト検索サーバーの使用が登録されます。このプロシージャを実行しない場合、重大エラーにはなりませんが、いくつかのコマンドに対して予期しない応答が発生することがあります。
- 複数のプロシージャまたはコマンドは、競合する可能性がある場合は、テキスト検索索引に対して同時に実行することはできません。競合するプロシージャおよびコマンドの一部は以下のとおりです。
  - SYSTS\_ALTER プロシージャ
  - SYSTS\_DISABLE プロシージャ
  - SYSTS\_CONFIGURE プロシージャ
  - SYSTS\_UPDATE プロシージャ

競合がある場合、プロシージャは SQLCODE -20426 および SQLSTATE 38H13 を戻します。

- テキスト検索のインストールと DB2 インスタンスのテキスト検索構成に関係する一部の設定を更新する必要があります。これには、以下が含まれます。
  - DB2 インスタンスによって使われる検索サーバーが統合される (DB2 によって DB2 インスタンスの一部として構成される) のか、あるいは ECMTS サーバーの別個のスタンドアロン・インストールであるかの指示。
  - テキスト検索のセットアップでリッチ・テキスト・サポートが可能であるかどうかの指示。

## SYSTS\_CREATE プロシージャ - 列のテキスト検索索引の作成

SYSTS\_CREATE プロシージャは、データベース・サーバー上で DB2 Text Search **CREATE INDEX** コマンドを発行して、テキスト列のテキスト検索索引を作成します。テキスト検索索引を作成して更新した後、テキスト検索関数を使用して列データを検索できます。

索引更新操作が処理されるまでは、索引には何のデータも入れられません。ストアード・プロシージャ・インターフェースまたはコマンド・ライン・インターフェースのいずれかを使用して、更新操作を開始できます。後者の場合、DB2 Text Search の **UPDATE INDEX** コマンドを明示的に発行することができます。そうしない場合、索引に定義されている更新頻度に従って、DB2 管理タスク・スケジューラーによって、コマンドは暗黙的に発行されます。

### 許可

ステートメントの許可 ID には、SYSTS\_MGR ロールおよびデータベースに対する CREATETAB 権限と、以下のいずれかの権限または特権が含まれていなければなりません。

- その索引の定義されている表に対する CONTROL 特権。
- 索引が定義される表に対する INDEX 特権と、以下の権限のいずれか。
  - データベースに対する IMPLICIT\_SCHEMA 権限 (索引の暗黙または明示のスキーマ名が存在しない場合)
  - スキーマに対する CREATEIN 特権 (索引のスキーマ名が存在する場合)
- DBADM 権限

### デフォルトの PUBLIC 特権

なし

### 構文

```
►►SYSTS_CREATE(—index_schema—,—index_name—,— | text source |—,—————►  
► | options |—,—message_locale—,—message—)—————►►
```

**text source:**

|-----|  
 |-----| *table-name* (-----| text column name |-----)|  
 |-----|  
 |-----| *table-schema* |-----|

**text column name:**

|-----|  
 |-----| *column-name* |-----|  
 |-----| *function-name (column-name)* |-----|

**options:**

-----
-----
-----
-----
-----
-----
-----

**text default information:**

|-----|  
 |-----| *CODEPAGE code-page* |-----| *LANGUAGE language* |-----| *FORMAT format* |-----|  
 |-----|

**update characteristics:**

|-----|  
 |-----| *UPDATE FREQUENCY NONE* |-----|  
 |-----| update frequency |-----|  
 |-----|  
 |-----| incremental update characteristics |-----|

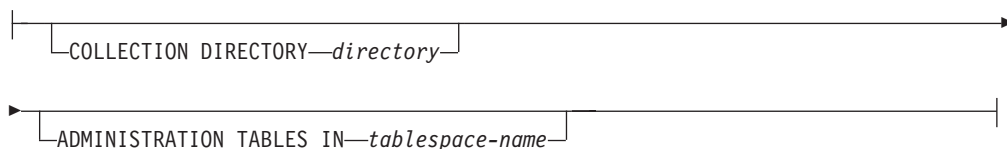
**update frequency:**

|-----|  
 |-----| *D* (-----| \* |-----|) |-----| *H* (-----| \* |-----|) |-----| *M* (-----| *integer3* |-----|) |-----|  
 |-----|  
 |-----| *integer1* |-----|  
 |-----| *integer2* |-----|

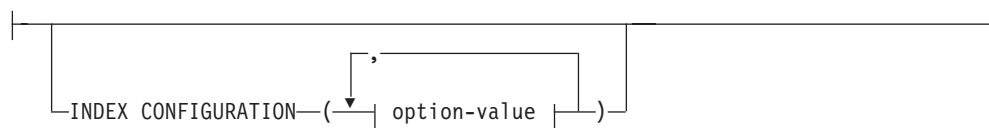
**incremental update characteristics:**

|-----|  
 |-----| *UPDATE MINIMUM minchanges* |-----|  
 |-----|

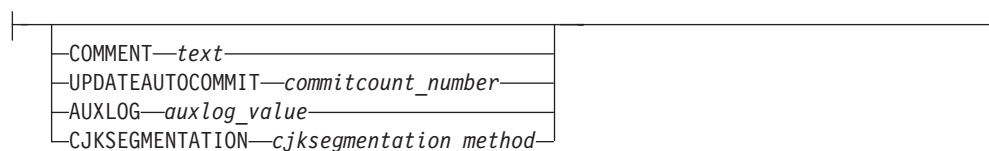
### storage options:



### index configuration options:



### option-value:



スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *index\_schema*

テキスト検索索引のスキーマを指定する、タイプ VARCHAR(128) の入力引数。 *index\_schema* は、DB2 スキーマ名の命名上の制約に準拠する必要があります。引数が NULL または空ストリングである場合、スキーマのデフォルト値が使用されます。 *index\_schema* には、大文字小文字の区別があります。

#### *index\_name*

索引の名前を指定する、タイプ VARCHAR (128) の入力引数。 *index\_name* を *index\_schema* と共に使用すると、データベースのテキスト検索索引を一意的に識別します。 *index\_name* には、大文字小文字の区別があります。

#### text source

索引付けされた列の名前を指定する、タイプ VARCHAR (1024) の入力引数。オプションは以下のとおりです。

#### *table-schema*

テキスト検索索引が作成される表のスキーマ。

#### *table-name*

テキスト列がある表の名前を指定します。フェデレーテッド表、マテリアライズ照会表、またはビューに対するテキスト検索索引を作成することはできません。 *table-name* には、大文字小文字の区別があります。

#### text column name

索引を付ける列の名前を指定します。



### *column-name*

索引を付ける列の名前を指定します。列は、CHAR、 VARCHAR、 CLOB、 DBCLOB、 BLOB、 GRAPHIC、 VARGRAPHIC、 または XML のデータ・タイプのいずれかでなければなりません。列のデータ・タイプがこれらのいずれのデータ・タイプでもない場合、 *function-schema.function-name* という名前の変換関数を使用して、列タイプを有効なタイプのいずれかに変換できます。(詳しくは、 *function-name (column-name)* パラメーターを参照してください。)あるいは、索引付けされるテキスト文書にアクセスするユーザー定義の外部関数を指定することができます。1つの列に作成できるテキスト検索索引は1つのみです。 *column-name* には、大文字小文字の区別があります。

### *function-name (column-name)*

テキスト検索用にサポートされていない列データ・タイプを、テキスト検索用にサポートされているデータ・タイプに変換する、外部スカラー関数のスキーマ修飾名を指定します。この関数は、1つのパラメーターのみを取り、1つの値のみを戻す必要があります。関数の名前は、DB2命名規則に準拠していなければなりません。 *function-name(column-name)* パラメーターには大文字小文字の区別があります。

### **options**

使用するオプションを指定する、タイプ VARCHAR(32) の入力引数。オプションを必要としない場合は、このパラメーターを NULL にするか、または空ストリングにすることができます。オプションは以下のとおりです。

### **CODEPAGE** *code-page*

テキスト文書を索引付けするときに使用される DB2 コード・ページを指定します。デフォルトのコード・ページ値は、SYSIBMTS.TSDEFAULTS ビューの、DEFAULTNAME 列値が CODEPAGE の行から取られます。新規索引の初期デフォルトのコード・ページは、データベース・コード・ページです。このパラメーターはバイナリー・データ・タイプにのみ適用されます。つまり列タイプや変換関数からの戻りの型は、BLOB または文字タイプの FOR BIT DATA でなければなりません。

### **LANGUAGE** *language*

索引付け中に、文書の言語固有の処理のために DB2 Text Search によって使用される言語を指定します。ロケールを指定しない場合、データベース・テリトリーを使用して **LANGUAGE** パラメーターのデフォルト設定が判別されます。ロケールを判別するために文書をスキャンさせる場合は、 *locale* を AUTO と指定してください。

### **FORMAT** *format*

列内のテキスト文書のフォーマットを指定します。サポートされているフォーマットは TEXT、XML、HTML、および INSO です。列のデータ・タイプが XML でない場合は、デフォルトのフォーマットは、SYSIBMTS.TSDEFAULTS ビューの、DEFAULTNAME 列値が **FORMAT** の行から取られます。列のデータ・タイプが XML の場合は、デフォルトのフォーマットは常に XML です。INSO フォーマットを使用する場合は、リッチ・テキスト・サポートをインストールしなければなりません。

### **UPDATE FREQUENCY**

索引の更新頻度を指定します。変更の数が **UPDATE MINIMUM** パラメーターに

設定された値以上になると、索引が更新されます。自動更新を行えるのは、DB2\_ATS\_ENABLE レジストリー変数が設定されていて、**START FOR TEXT** コマンドが発行される場合のみです。

デフォルトのフォーマットは、SYSIBMTS.TSDEFAULTS ビューの、DEFAULTNAME 列値が UPDATEFREQUENCY の行から取られます。

#### **NONE**

これ以上の索引更新は行われません。この値は、変更されないデータを持つ表のテキスト列に役立ちます。さらに、**UPDATE INDEX** コマンドを使用して手動で索引を更新する予定になっている場合にも役立ちます。

#### **D** 索引が更新される曜日。

\* 毎日。

*integer1*

日曜日から土曜日の特定の曜日: 0 から 6。

#### **H** 索引が更新される指定日の時。

\* 毎時。

*integer2*

午前 0 時から午後 11 時までの特定の時: 0 から 23。

#### **M** 索引が更新される指定時の分。

*integer3*

#### **UPDATE MINIMUM** *minchanges*

**UPDATE FREQUENCY** パラメーターで指定された時間に索引が増分的に更新される前に、テキスト文書に加えられている必要のある変更の最小数を指定します。正の整数値のみが許可されます。デフォルト値は、SYSIBMTS.TSDEFAULTS ビューの、DEFAULTNAME 列値が UPDATEMINIMUM の行から取られます。値が小さい場合は、表列とテキスト検索索引の間の整合性が増しますが、システムの負荷も増えます。**UPDATE INDEX** コマンドは、**USING UPDATE MINIMUM** パラメーターが指定されない限りこのパラメーターの値を無視します。

#### **COLLECTION DIRECTORY** *directory*

テキスト検索索引のコレクションが保管されるディレクトリーを指定します。絶対パスを指定する必要があり、絶対パス名の最大長は 215 文字です。Text Search サーバー・インスタンス・サービスのプロセス所有者は、このディレクトリーに対する読み取りおよび書き込み権限を持っている必要があります。

**COLLECTION DIRECTORY** パラメーターは、統合テキスト検索サーバーのセットアップでのみサポートされています。コレクションの場所に関する追加情報については、使用上の注意を参照してください。

#### **ADMINISTRATION TABLES IN** *table-space-name*

索引用に作成された管理表の既存の表スペース (TEMPORARY 表スペース以外) の名前を指定します。デフォルトでは、索引を作成する基本表の表スペースが使用されます。

パーティション表またはパーティション・データベース内の表に対してテキスト索引を作成する場合、この引数は必須です。パーティション・データベースの場合、この表スペースは基本表の表スペースと同じパーティション・グループでなければなりません。テキスト索引の管理表は、対応する基本表と同じように配分されます。

**INDEX CONFIGURATION (option-value)**

追加の索引関連オプションを、オプションと値のペアとして指定します。以下のオプションがサポートされています。

表 255. INDEX CONFIGURATION パラメーターのオプションと値のペア

オプション	値	データ・タイプ	説明
COMMENT	<i>text</i>	512 バイト未満のストリング値	DB2 Text Search の SYSIBMTS.TSINDEXES カタログ・ビューにある REMARKS 列にストリング・コメントを追加します。コメントをコレクションの説明としても使用します。
UPDATEAUTOCOMMIT	<i>commitcount _number</i>	整数	<p>ここで指定された回数の索引更新が行われると、コミットが実行され、イニシャル更新またはインクリメンタル更新としてそれまでの作業が保存されます。</p> <ul style="list-style-type: none"> <li>イニシャル更新では、データ更新をキャプチャーするトリガーがアクティブ化された後の基本表からの文書のバッチを、索引更新が処理します。更新された文書の数が <i>commitcount_number</i> の数に達すると、サーバーは暫定コミットを実行します。処理されていない文書により生成されたログ項目は、ステージング表から除去されます。イニシャル・テキスト索引更新に UPDATEAUTOCOMMIT オプションを使用すると、実行時間が大幅に増加します。</li> <li>インクリメンタル更新では、処理されたログ項目は、各暫定コミットでステージング表から除去されます。 <i>commitcount_number</i> は更新された文書の数であり、ステージング表項目の数ではありません。</li> </ul>

表 255. INDEX CONFIGURATION パラメーターのオプションと値のペア (続き)

オプション	値	データ・タイプ	説明
AUXLOG	<i>auxlog</i> <i>_value</i>	ストリング	トリガーに認識されない変更をキャプチャーするための、追加のログ・インフラストラクチャーの作成を制御します。範囲パーティション表のデフォルト設定は ON です。非範囲パーティション表の場合は <i>AuxLogNorm</i> を設定し、範囲パーティション表の場合は <i>AuxLogPart</i> を設定して、デフォルト表内のデフォルト値を変更できます。  索引の作成後に、テキスト索引の補助ログ・インフラストラクチャー・プロパティを変更することはできません。
CJKSEGMENTATION	<i>CJKSEGMENTATION</i> <i>_method</i>	ストリング	中国語、日本語、および韓国語 (zh_CN、zh_TW、ja_JP、ko_KR ロケール・セット) の文書に関するセグメンテーション方式を適用できます。自動言語検出が使用可能になっている (LANGUAGE AUTO) 場合も行われます。オプションを指定しないと、デフォルト表内の CJKSEGMENTATION の値が適用されます。サポートされる値は、以下のとおりです。 <ul style="list-style-type: none"> <li>• MORPHOLOGICAL</li> <li>• NGRAM</li> </ul> 指定されたセグメンテーション方式は SYSIBMTS.TSCONFIGURATION 管理ビューに追加され、テキスト索引の作成後に変更することはできません。

**要確認:** コメントなどの非数値は、単一引用符で囲む必要があります。ストリング値内の単一引用符の文字は、2 つの連続した単一引用符によって表記される必要があります。

**例:** INDEX CONFIGURATION (COMMENT 'Index on User''s Guide column')

*message\_locale*

戻されるすべてのエラー・メッセージに使用されるロケールを指定する、タイプ VARCHAR(33) の入力引数。引数が NULL または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、'en\_US' が使用されます。

*message*

正常に完了した操作の警告メッセージまたは通知メッセージを指定する、タイプ VARCHAR(32) の出力引数。

## 例

例 1: 以下の例では、myTextIndex と呼ばれるテキスト検索索引を作成するために、SYSTS\_CREATE プロシージャを呼び出します。UPDATE MINIMUM 10 パラメータは、索引に関連付けられたテキスト文書に 10 の変更が加えられてから、索引のインクリメンタル更新が実行されることを指定します。すべてのエラー・メッセージは英語で戻されます。基本となるテキスト検索コマンドが正常に実行された場合は、message 出力パラメーターがコマンド実行の状況を示すように設定されます。

```
CALL SYSPROC.SYSTS_CREATE('db2ts', 'myTextIndex',
  'myUserSchema.myBaseTable (myTextColumn)', 'UPDATE MINIMUM 10',
  'en_US', ?)
```

出力例を以下に示します。

```
Value of output parameters
-----
Parameter Name   : MESSAGE
Parameter Value  : Operation completed successfully.
Return Status    = 0
```

例 2: 以下の例では、myTextIndex2 と呼ばれるテキスト検索索引を作成するために、SYSTS\_CREATE プロシージャを呼び出します。オプションが指定されていません。この例では、索引が存在しており、結果としてエラー・メッセージが呼び出し元に戻されます。

```
CALL SYSPROC.SYSTS_CREATE('db2ts', 'myTextIndex2',
  'myUserSchema.myBaseTable (myTextColumn)', '', 'en_US', ?)
```

出力例を以下に示します。

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message is "CIE00201 Text search
index "db2ts"."myTextIndex2" already exists. "
```

## 使用上の注意

- テキスト検索管理プロシージャは、データベースへの既存の接続を使用します。テキスト検索管理操作を実行する前に、直前のトランザクションがコミットされなかった場合、デッドロックになる場合があります。これは、同じデータベース・オブジェクトに影響が及ぶので、直前のステップがコミットされるまで操作待機するためです。例えば、AUTOCOMMIT がオフになっている場合、表を作成してからテキスト索引を作成する際に 2 つのトランザクションの間で明示的なコミットをしないと、このデッドロックが生じる可能性があります。
- COLLECTION DIRECTORY 節を指定しないと、Text Server 構成内で defaultDataDirectory パラメーターを使用して定義したパス内の、システム生成の索引 ID にちなんで命名されたサブディレクトリー内にコレクションが入れられます。configTool ユーティリティを使用して、このパラメーターを明示的に構成することができます。defaultDataDirectory パラメーターが明示的に構成されていない場合には、コレクションのサブディレクトリーは、<configPath>/config/collections ディレクトリー内に配置されます。ホーム・ディレクトリーやデータベース・インスタンス・パス以外の場所を defaultDataDirectory や COLLECTION DIRECTORY として使用することを強くお勧めします。この場所に適切なストレージ・スペースがあり、しかもこの場所がローカルである (NFS にマウントされていない) ことを確認してください。

- **COLLECTION DIRECTORY** パラメーターの値を構成すると、ストライピングされた RAID デバイスの使用が可能になります。

キーに関連した以下の規則が適用されます。

- 表の主キーを定義する必要があります。
- 主キー列の数は、他の DB2 環境で許可されている主キー列の数より 2 列少ない数に制限されています。
- DB2 Text Search 索引のある表のすべての主キー列の合計長は、他の DB2 環境で許可されている主キーの最大合計長より 15 バイト少ない長さに制限されています。DB2 CREATE INDEX ステートメントに関する制約事項を参照してください。

**LANGUAGE** パラメーターを AUTO オプションに設定して索引を作成すると、CJKSEGMENTATION 指定がオプションとして許可されます。指定されたセグメンテーション方式は、中国語、日本語、および韓国語の文書に適用されます。いったん索引作成が完了したら、CJKSEGMENTATION\_method 値の値セットを変更することはできません。

特定のプロシージャまたはコマンドは、操作が競合するタイミングによってはエラー・メッセージが出される可能性があるため、テキスト検索索引に対して同時に実行することはできません。

- SYSTS\_DISABLE プロシージャまたは **db2ts DISABLE DATABASE FOR TEXT** コマンド
- SYSTS\_CONFIGURE プロシージャ

**CREATE INDEX** コマンドが正常に実行されると、以下の効果が得られます。

- DB2 Text Search のサーバー・データが更新されます。以下の例のように、*instance\_database-name\_index-identifier\_number* という名前のコレクションが作成されます。

```
tigertail_MYTSDB_TS250517_0000
```

パーティション・データベースの場合、コレクションはパーティションごとに作成されます。SYSIBMTS.TSCOLLECTIONNAMES ビューの COLLECTIONNAME 列からコレクション名を取得できます。

- 索引のイベント表が、指定された表スペースに作成されます。さらに、索引のステージング表が、適切な DB2 索引のある指定された表スペースに作成されます。**INDEX CONFIGURATION** パラメーターの AUXLOG ON オプションを設定した場合には、保全性処理による変更をキャプチャーするために、2 つ目のステージング表が作成されます。
- DB2 Text Search のカタログ情報が更新されます。しかし、新しく作成されたテキスト検索索引には、自動的にデータが追加されません。テキスト検索索引にデータを追加するには、**SYSTS\_UPDATE** プロシージャまたは **UPDATE INDEX** コマンドを手動で、または (**UPDATE FREQUENCY** パラメーターを使用して索引の更新スケジュールを定義した結果として) 自動で発行する必要があります。
- DB2 Text Search と DB2 Net Search Extender が共に存在し、表列のアクティブ Net Search Extender 索引が存在している場合、新しい DB2 Text Search 索引は非アクティブに設定されます。

- **LANGUAGE** パラメーターを **AUTO** に設定し、**CJKSEGMENTATION** を **MORPHOLOGICAL** に設定して、テキスト検索索引を作成した場合、形態素索引で有効なストリングを検索すると、ゼロ行が戻されます。結果を取得するには、以下の照会のように **CONTAINS** 関数に **QUERYLANGUAGE** オプションを追加します。

```
select bookname from morphobooks
where contains (story, 'QUERYLANGUAGE=zh_CN') = 1
```

## SYSTS\_DISABLE プロシージャ - テキスト検索の現行データベースを使用不可にする

このプロシージャは、現行データベースの DB2 テキスト検索を使用不可にします。

テキスト検索機能を使用不可にすると、テキスト検索索引およびコマンドは、データベースで使用できなくなります。

このプロシージャは **DISABLE DATABASE FOR TEXT** テキスト検索管理コマンドをデータベース・サーバー上で発行します。

### 許可

このステートメントの許可 ID には、以下の特権が含まれている必要があります。

- DBADM と DATAACCESS を併せ持つ権限。
- SYSTS\_ADM ロール。

### デフォルトの PUBLIC 特権

なし

### 構文

```
►►SYSTS_DISABLE(—options—, —message_locale—, —message—)◄◄
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### options

データベースが使用不可の場合に使用するオプションを指定する、タイプ **VARCHAR(128)** の入力引数。引数は **FORCE** に設定できます。この値を指定した場合、すべての索引はドロップされ、テキスト検索機能は強制的に使用不可にされます。テキスト検索索引は保存されず、エラー・メッセージまたは警告は戻されません。引数が **NULL** または空ストリングである場合、データベースのテキスト検索機能を使用不可にすることが試行されます。

#### message\_locale

戻されるすべてのエラー・メッセージに使用されるロケールを指定する、タイプ **VARCHAR(33)** の入力引数。引数が **NULL** または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、**'en\_US'** が使用されます。

## message

正常に完了した操作の警告または通知メッセージを指定する、タイプ VARCHAR(32K) の出力引数。

## 例

例 1: 以下の例では、テキスト検索は、SYSTS\_DISABLE プロシージャを使用してデータベースに対して使用不可にされます。FORCE オプションを指定すると、テキスト検索索引が引き続きデータベース内の表に存在しているとしても、この機能を実際に使用不可にできます。エラー・メッセージは英語で戻ることが指定されます。message 出力パラメーターは、情報メッセージ・ストリングに設定されません。

```
CALL SYSPROC.SYSTS_DISABLE('FORCE', 'en_US', ?)
```

以下は、この照会からの出力例です。

```
Value of output parameters
-----
Parameter Name   : MESSAGE
Parameter Value  : Operation completed successfully.

Return Status = 0
```

例 2: 以下の例では、FORCE オプションを指定せずに SYSTS\_DISABLE プロシージャを使用して、テキスト検索を既存のテキスト検索索引があるデータベースに対して使用不可にします。これは結果としてエラー・メッセージを呼び出し元に戻します。テキスト検索機能を使用不可にするか、またはその代わりに options 入力パラメーター値に FORCE オプションを指定する前に、すべての既存のテキスト検索索引をドロップすることをお勧めします。

```
CALL SYSPROC.SYSTS_DISABLE('', 'en_US', ?)
```

以下は、この照会からの出力例です。

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message is "CIE00326 Text search
index active in specified or default database. ". SQLSTATE 38H14
```

## 使用上の注意

- テキスト検索管理プロシージャは、データベースへの既存の接続を使用します。テキスト検索管理プロシージャにおけるコミットまたはロールバックからの予期しない影響を避けるために、そのプロシージャを実行する前に、すべてのトランザクション変更をコミットすることをお勧めします。これを行う 1 つの方法は、AUTOCOMMIT をオンにすることです。
- 複数のプロシージャまたはコマンドは、競合する可能性がある場合は、テキスト検索索引に対して同時に実行することはできません。競合するプロシージャのいくつかは、以下のとおりです。
  - SYSTS\_ALTER プロシージャ
  - SYSTS\_CLEAR\_EVENTS プロシージャ
  - SYSTS\_DISABLE プロシージャ
  - SYSTS\_CONFIGURE プロシージャ
  - SYSTS\_UPDATE プロシージャ



競合がある場合、プロシージャは SQLCODE -20426 および SQLSTATE 38H13 を戻します。

**注: SYSTS\_DISABLE** プロシージャに関するロックが、データベース・レベルで設定されます。

- このプロシージャが実行されると、
  - DB2 Text Search のカタログ情報が更新されます。索引ログおよびイベント表がドロップされます。ユーザー・テキスト表のトリガーが削除されます。
  - **FORCE** オプションを指定した場合、すべてのテキスト索引情報はデータベースから削除され、すべての関連する集合も削除されます。詳しくは、『db2ts DROP INDEX コマンド』または『SYSTS\_DROP プロシージャ』を参照してください。
- このプロシージャは、データベースの DB2 Net Search Extender 使用可能化状況には影響を与えません。これは、SYSTS\_ENABLE プロシージャまたは **ENABLE FOR TEXT** コマンドにより作成された DB2 テキスト検索カタログ表およびビューを削除します。
- テキスト検索索引定義がある DB2 データベースをドロップする前に、このプロシージャを実行して、テキスト索引および集合が正常に削除されていることを確認してください。
- **FORCE** オプションを使用して、いくつかの索引が削除できなかった場合、コレクション名が **db2diag** ログ・ファイルに書き込まれます。

**注:** 結果がオーファン集合 (つまり残された集合がテキスト検索サーバー上に定義された状態であるが、DB2 には使用されない) となる使用法は推奨されていません。以下は、孤立したコレクションの原因となる可能性のあるいくつかのケースです。

- **DISABLE DATABASE FOR TEXT** コマンドを実行せずに、**DROP DATABASE CLP** コマンドが実行される場合。
- **FORCE** オプションを使用して、**SYSTS\_DISABLE** プロシージャまたは **DISABLE DATABASE FOR TEXT** コマンドが実行される場合。

## SYSTS\_DROP プロシージャ - テキスト検索索引のドロップ

SYSTS\_DROP プロシージャは、表列と関連付けられた既存のテキスト検索索引をドロップします。

このプロシージャが正常に実行した後は、テキスト検索索引がドロップされた列に対してテキスト検索照会を実行することはできません。

このプロシージャは **DROP INDEX** テキスト検索管理コマンドをデータベース・サーバー上で発行します。

### 許可

ステートメントの許可 ID によって保持されている特権には、SYSTS\_MGR ロールと、少なくとも以下のいずれかの特権または権限が含まれていなければなりません。

- その索引の定義されている表に対する **CONTROL** 特権。
- その索引の定義されているスキーマに対する **DROPIN** 特権。

- テキスト検索索引に既存のスケジュールがある場合、許可 ID は索引の作成者と同じであるか、DBADM 権限が含まれている必要があります。

## デフォルトの PUBLIC 特権

なし

## 構文

```
►►—SYSTS_DROP—(—index_schema—,—index_name—,—options——————►
►—message_locale—,—message—)—————►►
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *index\_schema*

テキスト検索索引のスキーマを指定する、タイプ VARCHAR(128) の入力引数。 *index\_schema* は、DB2 スキーマ名の命名上の制約に従う必要があります。引数が NULL または空ストリングである場合、値 CURRENT SCHEMA が使用されます。 *index\_schema* には、大文字小文字の区別があります。

### *index\_name*

索引の名前を指定する、タイプ VARCHAR (128) の入力引数。 *index\_schema* とともに使用すると、データベースのテキスト検索索引を一意的に識別します。 *index\_name* には、大文字小文字の区別があります。

### *options*

使用するオプションを指定する、タイプ VARCHAR(32000) の入力引数。オプションを必要としない場合は、引数を NULL にするか、または空ストリングにすることができます。

### *message\_locale*

戻されるすべてのエラー・メッセージに使用されるロケールを指定する、タイプ VARCHAR(33) の入力引数。引数が NULL または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、'en\_US' が使用されます。

### *message*

正常に完了した操作の警告または通知メッセージを指定する、タイプ VARCHAR(32K) の出力引数。

## 例

例 1: 以下の例では、*index\_schema* 'db2ts' および *index\_name* 'myTextIndex' で作成されたテキスト検索索引がドロップされます。すべてのエラー・メッセージは英語で戻すことが要求されます。このプロシージャが成功すると、操作の成功を示す出力パラメーター・メッセージが呼び出し元に戻されます。

```
CALL SYSPROC.SYSTS_DROP('db2ts', 'myTextIndex', '', 'en_US', ?)
```

前のリリースと同様に SYSTS\_DROP プロシージャは、 **options** 引数を指定しなくてもサポートされます。以下に例を示します。

```
CALL SYSPROC.SYSTS_DROP('db2ts', 'myTextIndex', 'en_US', ?)
```

以下はこの照会の出力例です。

```
Value of output parameters
-----
Parameter Name : MESSAGE
Parameter Value : Operation completed successfully.

Return Status = 0
```

例 2: 以下の例では、SYSTS\_DROP が呼び出され、*index\_schema* 'db2ts' および *index\_name* 'myTextIndex2' があるテキスト検索索引がドロップされます。この索引は存在せず、結果としてエラーを戻します。

```
CALL SYSPROC.SYSTS_DROP('db2ts', 'myTextIndex2', 'en_US', ?)
```

以下はこの照会の出力例です。

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message is "CIE00316 Text search
index "db2ts"."myTextIndex2" does not exist. ". SQLSTATE 38H14
```

### 使用上の注意

- 複数のプロシージャまたはコマンドは、競合する可能性がある場合は、テキスト検索索引に対して同時に実行することはできません。以下に、競合するプロシージャをいくつか示します。

- SYSTS\_ALTER プロシージャ
- SYSTS\_CLEAR\_EVENTS プロシージャ
- SYSTS\_DISABLE プロシージャ
- SYSTS\_DROP プロシージャ
- SYSTS\_UPDATE プロシージャ
- SYSTS\_CONFIGURE プロシージャ

DROP 操作と並行して実行する STOP FOR TEXT コマンドでは、競合するコマンド・メッセージは発生しませんが、代わりに、DROP がコレクションを除去する前にテキスト検索サーバーがシャットダウンされる場合は、テキスト検索サーバーが使用不可であるというエラーが戻されます。

- テキスト検索索引がドロップされた後、テキスト検索は対応するテキスト列で使用できなくなります。同じテキスト列に新規のテキスト検索を作成する予定の場合、まずデータベースから切断し、再接続してから、新規のテキスト検索索引を作成する必要があります。
- db2ts DROP INDEX コマンドは、データベースに以下の変更を行います。
  - DB2 Text Search のカタログ情報を更新します。
  - 索引ステージング表およびイベント表をドロップします。
  - ユーザー・テキスト表のトリガーを削除します。
  - DB2 Text Search 索引定義に関連付けられたコレクションを破棄します。

### SYSTS\_ENABLE プロシージャ - テキスト検索の現行データベースを使用可能にする

SYSTS\_ENABLE プロシージャは、現行データベースの DB2 Text Search を使用可能にします。

このプロシージャは、データベース内の列に対してテキスト検索索引を作成する前に、正常に実行する必要があります。

このプロシージャは **ENABLE DATABASE FOR TEXT** テキスト検索管理コマンドをデータベース・サーバー上で発行します。

## 許可

プロシージャの許可 ID によって保持されている特権には、SYSTS\_ADM ロールと DBADM 権限が含まれていなければなりません。

## デフォルトの PUBLIC 特権

なし

## 構文

```
▶▶SYSTS_ENABLE—(—options—,—message_locale—,—message—)————▶▶
```

### options:

```
▶▶————▶▶  
└── ADMINISTRATION TABLES IN table-space-name ───┘
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### options

使用するオプションを指定する、タイプ VARCHAR(32K) の入力引数。オプションを必要としない場合は、引数を NULL にするか、または空ストリングにすることができます。サポートされる値は以下のとおりです。

#### ADMINISTRATION TABLES IN *table-space-name*

データベースを DB2 Text Search に使用可能にする際に作成される管理表のための、既存の REGULAR 表スペースの名前を指定します。この節が指定されない場合には、SYSTOOLSPACE が表スペースとして使用されます。SYSTOOLSPACE を使用するのか、明示的に指定された表スペース名を使用するのにかかわらず、表スペースは特定の要件を満たしている必要があります。要件は次のとおりです。

- REGULAR 表スペース
- パーティション・データベースの場合、バッファ・プールおよびページ・サイズが 32KB の表スペースを定義することをお勧めします。

大/小文字の区別のある表スペース名を使用するには、名前を二重引用符で囲んで、区切り ID を使用します。デフォルトでは、通常 ID として扱われ、大文字に変換されます。

#### *message\_locale*

戻されるすべてのエラー・メッセージに使用されるロケールを指定する、タイプ

VARCHAR(33) の入力引数。引数が NULL または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、'en\_US' が使用されます。

#### *message*

正常に完了した操作の警告または通知メッセージを指定する、タイプ VARCHAR(32K) の出力引数。

### 例

例 1: 表スペース内に管理表と英語の出力メッセージを作成して、データベースをテキスト検索に使用できるようにします。

```
CALL SYSPROC.SYSTS_ENABLE ('ADMINISTRATION TABLES IN TSSPACE', 'en_US', ?)
```

表スペース内に管理表とフランス語の出力メッセージを作成して、データベースをテキスト検索に使用できるようにします。

```
CALL SYSPROC.SYSTS_ENABLE ('ADMINISTRATION TABLES IN "tbs32k" ', 'fr_FR', ?)
```

前のリリースと同様に SYSTS\_ENABLE プロシージャは、**options** 引数を指定しなくてもサポートされます。以下に例を示します。

```
CALL SYSPROC.SYSTS_ENABLE ('en_US', ?)
```

この照会の出力例:

```
Value of output parameters
-----
Parameter Name : MESSAGE
Parameter Value : Operation completed successfully.

Return Status = 0
```

例 2: 以下の例では、SYSTS\_ENABLE がテキスト検索用に既に使用可能にされているデータベース上で呼び出されます。これは結果としてエラー・メッセージを呼び出し元に戻します。

```
CALL SYSPROC.SYSTS_ENABLE('en_US', ?)
```

この照会の出力例:

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message from the text search
product is "CIE00322 Specified or default database already
enabled for text. ". SQLSTATE 38H14
```

### 使用上の注意

- テキスト検索管理プロシージャは、データベースへの既存の接続を使用します。テキスト検索管理プロシージャ内のコミットまたはロールバックからの予期しない影響を避けるため、プロシージャを実行する前に、すべてのトランザクション変更をコミットすることをお勧めします。これを行う 1 つの方法は、AUTOCOMMIT をオンにすることです。
- このプロシージャが実行されると、次のイベントが発生します。
  - このプロシージャは、スキーマ SYSIBMTS にテキスト検索管理カタログ表およびビューなどのデータベース・オブジェクトを作成します。
  - テキスト検索索引の設定済みデータベース・デフォルトは、ビュー SYSIBMTS.TSDEFAULTS で入手可能です。

- コマンドが正常に完了したら、テキスト検索カタログ表およびビューが作成され、使用可能になります。
- このプロシージャを実行する場合、Text Search サーバー構成を取得するための十分な実行およびファイルアクセス特権がなければ、プロシージャはテキスト検索カタログを「不完全な使用可能化」警告とともに作成します。この場合は、Text Search サーバー接続情報を手動で更新する必要があります。

## SYSTS\_UPDATE プロシージャ - テキスト検索索引の更新

**SYSTS\_UPDATE** プロシージャは、索引が関連付けられるテキスト列の現在の内容を反映するテキスト検索索引を更新します。

更新が実行されている間も、検索は可能です。更新が完了するまでは、検索は部分的に更新された索引に対して実行されます。

このプロシージャは **UPDATE INDEX** テキスト検索管理コマンドをデータベース・サーバー上で発行します。

### 許可

ステートメントの許可 ID が保持する特権には、SYSTS\_MGR ロールと以下の少なくとも 1 つの権限が含まれている必要があります。

- DATAACCESS 権限
- そのテキスト索引の定義されている表に対する CONTROL 特権。
- テキスト索引が定義されている基本表に対する、INDEX 特権および SELECT 特権。

さらに、イニシャル更新では、許可要件が **CREATE TRIGGER** ステートメントで概説されているように適用されます。

### デフォルトの PUBLIC 特権

なし

### 構文

```
►►SYSTS_UPDATE(—index_schema—,—index_name—,——————►
►—update_options—,—message_locale—,—message—)—————►
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *index\_schema*

テキスト検索索引のスキーマを指定する、タイプ VARCHAR(128) の入力引数。 *index\_schema* は、DB2 スキーマ名の命名上の制約に従う必要があります。引数が NULL または空ストリングである場合、値 CURRENT SCHEMA が使用されます。 *index\_schema* には、大文字小文字の区別があります。

#### *index\_name*

索引の名前を指定する、タイプ VARCHAR (128) の入力引数。 *index\_schema*

とともに使用すると、データベースのテキスト検索索引を一意的に識別します。  
*index\_name* には、大文字小文字の区別があります。

#### *update\_options*

更新オプションを指定する、タイプ VARCHAR(32K) の入力引数。オプションを必要としない場合は、引数を NULL にするか、または空ストリングにすることができます。可能な値は次のとおりです。

- USING UPDATE MINIMUM: この値は、**CREATE INDEX** テキスト検索管理コマンドおよび SYSTS\_CREATE プロシージャからの UPDATE MINIMUM 設定を受け入れます。
- FOR DATA REDISTRIBUTION: データ・パーティションが追加または削除され、後続のデータ再配分操作が完了した後に、パーティション・データベース内のテキスト検索索引をリフレッシュする必要があることを指定します。検索結果は、テキスト検索索引が FOR DATA REDISTRIBUTION オプションで更新されるまでは、不整合である場合があります。
- ALLROWS: イニシャル更新を無条件に試行することを指定します。
- NULL または空ストリング (""): プロシージャの呼び出し時に更新が無条件に開始されることを指定します。

#### *message\_locale*

戻されるすべてのエラー・メッセージに使用されるロケールを指定する、タイプ VARCHAR(33) の入力引数。引数が NULL または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、'en\_US' が使用されます。

#### *message*

正常に完了した操作の警告または通知メッセージを指定する、タイプ VARCHAR(32K) の出力引数。

## 例

例 1: 以下の例では、*index\_schema* 'db2ts' および *index\_name* 'myTextIndex' で作成されたテキスト検索索引が更新されます。 *update\_options* が NULL 値であることは、ストアード・プロシージャの呼び出し時に更新が無条件で開始されることを意味します。すべてのエラー・メッセージは英語で戻ることが要求されます。このプロシージャが成功すると、操作の成功を示す出力パラメーター・メッセージが呼び出し元に戻されます。

```
CALL SYSPROC.SYSTS_UPDATE  
( 'db2ts', 'myTextIndex', '', 'en_US', ? )
```

この照会からの出力例:

```
Value of output parameters  
-----  
Parameter Name : MESSAGE  
Parameter Value : Operation completed successfully.
```

例 2: 基本表に関連付けられているパーティション・グループ上で操作が実行された後、テキスト索引を更新し、エラー・メッセージを英語で戻します。

```
CALL SYSPROC.SYSTS_UPDATE  
( 'db2ts', 'myTextIndex2', 'FOR DATA REDISTRIBUTION', 'en_US', ? )
```

Value of output parameters

-----  
Parameter Name : MESSAGE  
Parameter Value : Operation completed successfully.

Return Status = 0

例 3: 以下の例では、SYSTS\_UPDATE が呼び出され、*index\_schema* 'db2ts' および *index\_name* 'myTextIndex3' があるテキスト検索索引が更新されます。この索引は存在せず、結果としてエラーを戻します。

```
CALL SYSPROC.SYSTS_UPDATE('db2ts', 'myTextIndex3', 'USING UPDATE MINIMUM',  
'en_US', ?)
```

```
SQL20427N An error occurred during a text search administration  
procedure or command. The error message is "CIE00316 Text search  
index "db2ts"."myTextIndex3" does not exist. ". SQLSTATE 38H14
```

### 使用上の注意

- テキスト検索管理プロシージャは、データベースへの既存の接続を使用します。現行トランザクションは、プロシージャの完了結果に応じてコミットされるかまたはロールバックされる場合があります。そのようなコミットまたはロールバックからの予期しない影響を避けるために、すべてのトランザクション変更をコミットすることができます。すべてのトランザクション変更をコミットする方法の 1 つとして、AUTOCOMMIT をオンにする方法があります。
- 特定のプロシージャまたはコマンドは、テキスト検索索引に対して同時に実行することはできません。これは、競合する操作のタイミングによってエラーが発生する可能性があるためです。競合するプロシージャおよびコマンドの一部は以下のとおりです。
  - SYSTS\_ALTER プロシージャまたは **db2ts ALTER INDEX** コマンド
  - SYSTS\_CLEAR\_EVENTS プロシージャまたは **db2ts CLEAR EVENTS FOR INDEX** コマンド
  - SYSTS\_DISABLE プロシージャまたは **db2ts DISABLE DATABASE FOR TEXT** コマンド
  - SYSTS\_UPDATE プロシージャまたは **db2ts UPDATE INDEX** コマンド

競合がある場合、プロシージャは SQLCODE -20426 および SQLSTATE 38H13 を戻します。

- このプロシージャは、すべての索引更新処理が完了するまで戻されません。この期間は、これから索引付けされる文書の数および既に索引付けされている文書の数に応じて異なります。索引のコレクション名は、SYSIBMTS.TSCOLLECTIONNAMES ビューの COLLECTIONNAME 列から取られます。
- 個々の文書にエラーがある場合、文書を訂正する必要があります。エラーがある文書の主キーは、索引のイベント表で参照できます。ユーザー表内の対応する行を変更することで、SYSTS\_UPDATE への次の呼び出しはこれらの文書を再処理します。
- SYSTS\_UPDATE プロシージャが実行すると、次のイベントが発生します。



- 行がイベント表に挿入されます (パーサー・エラー情報を含む)。インクリメンタル更新の場合、情報は索引ステーキング表から削除されます。最初の更新の前に、SYSTS\_UPDATE プロシージャはユーザー表にトリガーを作成します。
- コレクションが更新されます。
  - 新規または変更された文書が構文解析および索引付けされます。
  - 削除された文書が索引から破棄されます。
- シノニム・ディクショナリーがテキスト索引に関連付けられている場合、ALLROWS オプションまたは FOR DATA REDISTRIBUTION オプションを指定して更新を実行すると、テキスト索引のコレクションがドロップおよび再作成されて、関連が除去されます。シノニム・ディクショナリーは、新しいテキスト索引コレクションに関連付ける必要があります。

## SYSTS\_UPGRADE\_CATALOG プロシージャ - テキスト検索カタログのアップグレード

このプロシージャは、管理表および管理ビューを含む DB2 Text Search カタログを、製品の最新バージョンにアップグレードします。

このプロシージャは、製品の最新バージョンで使用される新しいカタログ表およびビューを作成するとともに、既存のカタログ表およびビューを更新します。廃止されたカタログ表およびビューは削除されます。

### 許可

このプロシージャの許可 ID が持つ特権に、SYSTS\_ADM ロールおよび DBADM 権限が含まれている必要があります。

### デフォルトの PUBLIC 特権

なし

### 構文

```
►►—SYSTS_UPGRADE_CATALOG—(—message_locale—,—message—)—————►►
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *message\_locale*

戻されるすべてのエラー・メッセージに使用されるロケールを指定する、タイプ VARCHAR(33) の入力引数。引数が NULL または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、'en\_US' が使用されます。

#### *message*

正常に完了した操作の警告または通知メッセージを指定する、タイプ VARCHAR(32K) の出力引数。

## 例

例 1: 次の例では、以前のリリースのテキスト検索がデータベースで使用可能になっています。 **SYSTS\_UPGRADE\_CATALOG** プロシージャを呼び出すと、テキスト検索カタログ表がアップグレードされます。プロシージャが成功すると、操作の成功を示す出力パラメーター・メッセージが呼び出し元に戻されます。

```
CALL SYSPROC.SYSTS_UPGRADE_CATALOG('en_US',?)
```

Value of output parameters

```
-----  
Parameter Name : MESSAGE  
Parameter Value : CIE0213W The DB2 Text Search Catalog has been upgraded to  
the current version. You will now need to update the text search index using  
the SYSPROC.SYSTS_UPGRADE_INDEX stored procedure.  
index also needs to be updated by calling "SYSPROC.SYSTS_UPGRADE_INDEX".
```

Return Status = 0

例 2: 次の例では、プロシージャを呼び出す前は、旧リリースの DB2 Text Search がデータベースで使用可能になっていません。 *message\_locale* の位置にある NULL 値は、デフォルト・ロケール 'en\_US' が使用されることを意味します。エラー・メッセージはすべて英語で戻されます。

```
CALL SYSPROC.SYSTS_UPGRADE_CATALOG('','?')
```

```
SQL20427N An error occurred during a text search administration procedure  
or command. The error message is "CIE0323E Specified or default database  
not enabled for text. ".  
SQLSTATE=38H14
```

例 3: 次の例では、テキスト検索カタログとテキスト検索索引が現行バージョンに既にアップグレードされています。

```
CALL SYSPROC.SYSTS_UPGRADE_CATALOG('en_US',?)
```

Value of output parameters

```
-----  
Parameter Name : MESSAGE  
Parameter Value : CIE0002I The DB2 Text Search release level is current  
for the database. The system has not been upgraded.
```

Return Status = 0

## 使用上の注意

テキスト検索カタログ固有のアップグレードを実行するため、

**SYSTS\_UPGRADE\_CATALOG** プロシージャが **DB2 UPGRADE DATABASE** コマンドに組み込まれています。データベース・アップグレードがテキスト検索カタログのアップグレードに失敗した場合、ユーザーが **SYSTS\_UPGRADE\_CATALOG** プロシージャを別個に実行し、データベース・アップグレードを完了させる必要があります。

テキスト検索索引カタログをアップグレードするには、以下の手順を実行します。

1. DB2 Text Search インスタンス・サービスが停止していることを確認します。
2. **SYSTS\_UPGRADE\_CATALOG** プロシージャを実行します。

注: DB2 Text Search 管理プロシージャは、データベースへの既存の接続を使用します。現行トランザクションは、プロシージャの完了結果に応じてコミットされるかまたはロールバックされる場合があります。そのため、そのようなコ

ミットまたはロールバックからの予期しない影響を避けるために、すべてのトランザクション変更をコミットすることができます。これを行う 1 つの方法は、**AUTOCOMMIT** をオンにすることです。

3. **SYSTS\_UPGRADE\_CATALOG** プロシージャが、カタログに Text Search サーバー情報の設定を試行します。 **SYSIBMTS.TSSERVER** の内容を検討して、必要に応じて、Text Search サーバー情報を更新してください。
4. **SYSTS\_UPGRADE\_INDEX** プロシージャについて示されている手順に従って、テキスト検索索引をアップグレードします。 **SYSTS\_UPGRADE\_INDEX** プロシージャの実行が正常に完了するまでは、**SYSIBMTS.TSDEFAULTS** 管理ビュー内のバージョン値が更新されません。

## SYSTS\_UPGRADE\_INDEX - テキスト検索索引のアップグレード

このプロシージャは、DB2 カタログ表およびテキスト検索カタログ表での DB2 Text Search 索引情報を更新します。

テキスト検索索引コレクションは、Text Search サーバーによって管理されます。詳しくは、DB2 Text Search のアップグレードを参照してください。

### 許可

このプロシージャの許可 ID が持つ特権に、**SYSTS\_ADM** ロールおよび **DBADM** 権限が含まれている必要があります。

### デフォルトの PUBLIC 特権

なし

### 構文

```
►►SYSTS_UPGRADE_INDEX(—message_locale—,—message—)◄◄
```

スキーマは **SYSPROC** です。

### プロシージャ・パラメーター

#### *message\_locale*

戻されるすべてのエラー・メッセージに使用されるロケールを指定する、タイプ **VARCHAR(33)** の入力引数。引数が **NULL** または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、**'en\_US'** が使用されます。

#### *message*

正常に完了した操作の警告または通知メッセージを指定する、タイプ **VARCHAR(32K)** の出力引数。

### 例

例 1: 次の例では、旧リリースのテキスト検索がデータベースで使用可能になっていましたが、プロシージャ **SYSTS\_UPGRADE\_CATALOG** が既に正常に完了しています。プロシージャ **SYSTS\_UPGRADE\_INDEX** は、データベース・カタログでのテキスト索

引メタデータのアップグレードを完了します。このプロシージャが成功すると、操作の成功を示す出力パラメーター・メッセージが呼び出し元に戻されます。

```
CALL SYSPROC.SYSTS_UPGRADE_INDEX('en_US', ?)
```

Parameter Name : MESSAGE

Parameter Value : CIE00001 Operation completed successfully.

Return Status = 0

例 2: 次の例では、旧リリースの製品のテキスト検索がデータベースで使用可能になっていません。 *message\_locale* に NULL 値を設定した場合、システム・ロケールが使用されることとなります。システム・ロケールが使用不可の場合は、デフォルト・ロケール 'en\_US' が使用されます。

```
CALL SYSPROC.SYSTS_UPGRADE_INDEX('', ?)
```

SQL20427N An error occurred during a text search administration procedure or command. The error message is "CIE0323E Specified or default database not enabled for text. ".

SQLSTATE=38H14

例 3: 次の例では、DB2 Text Search カタログとテキスト検索の索引が現行バージョンに既にアップグレードされています。

```
CALL SYSPROC.SYSTS_UPGRADE_INDEX('en_US', ?)
```

Value of output parameters

-----

Parameter Name : MESSAGE

Parameter Value : CIE0002I The DB2 Text Search release level is current for the database. The system has not been upgraded.

Return Status = 0

例 4: 次の例では、DB2 Text Search カタログがアップグレードされていません。

```
CALL SYSPROC.SYSTS_UPGRADE_INDEX('en_US', ?)
```

CIE0409E The DB2 Text Search catalog has not been upgraded to the current version.

## 使用上の注意

- テキスト検索索引のアップグレードを実行するため、**SYSTS\_UPGRADE\_INDEX** プロシージャが **DB2 UPGRADE DATABASE** コマンドに組み込まれています。 **DB2 UPGRADE DATABASE** コマンドを発行すると、このプロシージャもまた実行されます。テキスト検索索引のアップグレードに失敗した場合、手順を手動で実行する必要があります。
- **SYSTS\_UPGRADE\_INDEX** プロシージャの前に **SYSTS\_UPGRADE\_CATALOG** プロシージャを実行する必要があります。 **SYSTS\_UPGRADE\_INDEX** プロシージャの実行が正常に完了するまでは、**SYSIBMTS.TSDEFAULTS** 管理ビュー内のバージョン値が更新されません。
- DB2 Text Search 管理プロシージャは、データベースへの既存の接続を使用します。現行トランザクションは、プロシージャの完了結果に応じてコミットされるかまたはロールバックされる場合があります。そのため、そのようなコミットまたはロールバックからの予期しない影響を避けるために、すべてのトランザクション変更をコミットすることができます。これを行う 1 つの方法は、**AUTOCOMMIT** をオンにすることです。

## ワークロード管理ルーチン

### WLM\_CANCEL\_ACTIVITY - アクティビティのキャンセル

このプロシージャは、指定されたアクティビティをキャンセルします。キャンセルが行われる場合、キャンセルされたアクティビティをサブミットしたアプリケーションにエラー・メッセージが戻されます。

#### 構文

```
▶▶—WLM_CANCEL_ACTIVITY—(—application_handle—,—uow_id—,—activity_id—)————▶▶
```

スキーマは SYSPROC です。

#### プロシージャ・パラメーター

##### *application\_handle*

アクティビティがキャンセルされるアプリケーション・ハンドルを指定する、タイプ BIGINT の入力引数。引数が NULL の場合、アクティビティは検出されず、SQLSTATE 5U035 の SQL4702N が戻されます。

##### *uow\_id*

キャンセルされるアクティビティの作業単位 ID を指定する、タイプ INTEGER の入力引数。引数が NULL の場合、アクティビティは検出されず、SQLSTATE 5U035 の SQL4702N が戻されます。

##### *activity\_id*

キャンセルされる作業単位内のアクティビティを一意的に識別するアクティビティ ID を指定する、タイプ INTEGER の入力引数。引数が NULL の場合、アクティビティは検出されず、SQLSTATE 5U035 の SQL4702N が戻されます。

#### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

#### デフォルトの PUBLIC 特権

なし

#### 例

管理者は WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 表関数を使用して、アクティビティのアプリケーション・ハンドル、作業単位 ID、およびアクティビティ ID を検索できます。アプリケーション・ハンドル 1、作業単位 ID 2、およびアクティビティ ID 3 のアクティビティをキャンセルするには、次のようにします。

CALL WLM\_CANCEL\_ACTIVITY(1, 2, 3)

### 使用上の注意

- アクティビティーが見つからない場合、SQLSTATE 5U035 の SQL4702N が戻されます。
- アクティビティーが正しい状態でない (初期化されていない) ためにキャンセルできない場合、SQLSTATE 5U016 の SQL4703N (理由コード 1) が戻されます。
- アクティビティーが正常にキャンセルされた場合、SQLSTATE 57014 の SQL4725N がキャンセルされたアプリケーションに戻されます。
- キャンセル時に、コーディネーターが別のアクティビティーの要求を処理しているかまたはアイドル状態である場合、アクティビティーは CANCEL\_PENDING 状態になり、コーディネーターが次の要求を処理するとキャンセルされます。

### WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS - アクティビティー・イベント・モニターのアクティビティー情報の収集

WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS プロシージャは、指定されたアクティビティーに関する情報を収集し、その情報をアクティブなアクティビティー・イベント・モニターに書き込みます。

子アクティビティーを持つアクティビティーにこのプロシージャを適用する場合、プロシージャはそれぞれの子アクティビティーのレコードを再帰的に生成します。この情報は、プロシージャを呼び出すときに収集されて送信されます。プロシージャは、親アクティビティーによる実行の完了を待機しません。イベント・モニター内のアクティビティーのレコードは部分レコードとしてマークが付けられます。

### 構文

```
▶▶—WLM_CAPTURE_ACTIVITY_IN_PROGRESS—(—application_handle—,—————▶▶  
▶—uow_id—, —activity_id—)—————▶▶
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

以下のパラメーターがすべて指定されないと、アクティビティーは検出されず、SQL4702N が SQLSTATE 5U035 とともに返されます。

#### *application\_handle*

そのアクティビティー情報がキャプチャーされるアプリケーションのハンドルを指定する、タイプ BIGINT の入力引数。

#### *uow\_id*

その情報がキャプチャーされるアクティビティーの作業単位 ID を指定する、タイプ INTEGER の入力引数。

#### *activity\_id*

その情報がキャプチャーされる作業単位内のアクティビティーを一意的に識別するアクティビティー ID を指定する、タイプ INTEGER の入力引数。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

ストアド・プロシージャ `MYSCHEMA.MYSLOWSTP` の実行がいつもより遅いように感じる、とユーザーが苦情を言うとします。管理者はスローダウンの原因の調査に乗り出します。ストアド・プロシージャを実行しながらの調査は実際的とは言えないので、管理者はストアド・プロシージャ・アクティビティーおよびその中にネストされたすべてのアクティビティーに関する情報をキャプチャーすることにします。

`DB2ACTIVITIES` という名前の `DB2` アクティビティーのイベント・モニターがアクティブ化されています。管理者は

`WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES` 関数を使用して、このストアド・プロシージャの呼び出しに関するアプリケーション・ハンドル、作業単位 ID、およびアクティビティー ID を取得します。ここで、管理者がアクティビティーがアプリケーション・ハンドル 1、作業単位 ID 2、およびアクティビティー ID 3 で識別されていると想定し、`WLM_CAPTURE_ACTIVITY_IN_PROGRESS` への呼び出しを次のように発行できます。

```
CALL WLM_CAPTURE_ACTIVITY_IN_PROGRESS(1,2,3)
```

プロシージャが完了した後、管理者は次の表関数を使用してアクティビティーが時間を要した場所を発見することができます。関数は、`DB2ACTIVITIES` イベント・モニターから情報を取得します。

```
CREATE FUNCTION SHOWCAPTUREDACTIVITY (APPHNDL BIGINT,  
                                       UOWID INTEGER,  
                                       ACTIVITYID INTEGER)  
  RETURNS TABLE (UOW_ID INTEGER, ACTIVITY_ID INTEGER, STMT_TEXT VARCHAR(40),  
                 LIFE_TIME DOUBLE)  
  LANGUAGE SQL  
  READS SQL DATA  
  NO EXTERNAL ACTION  
  DETERMINISTIC  
  RETURN WITH RAH (LEVEL, APPL_ID, PARENT_UOW_ID, PARENT_ACTIVITY_ID,  
                  UOW_ID, ACTIVITY_ID, STMT_TEXT, ACT_EXEC_TIME) AS  
  (SELECT 1, ROOT.APPL_ID, ROOT.PARENT_UOW_ID,  
          ROOT.PARENT_ACTIVITY_ID, ROOT.UOW_ID, ROOT.ACTIVITY_ID,  
          ROOTSTMT.STMT_TEXT, ACT_EXEC_TIME  
   FROM ACTIVITY_DB2ACTIVITIES ROOT, ACTIVITYSTMT_DB2ACTIVITIES ROOTSTMT  
   WHERE ROOT.APPL_ID = ROOTSTMT.APPL_ID AND ROOT.AGENT_ID = APPHNDL  
         AND ROOT.UOW_ID = ROOTSTMT.UOW_ID AND ROOT.UOW_ID = UOWID  
         AND ROOT.ACTIVITY_ID = ROOTSTMT.ACTIVITY_ID AND ROOT.ACTIVITY_ID = ACTIVITYID  
   UNION ALL
```

```

SELECT PARENT.LEVEL +1, CHILD.APPL_ID, CHILD.PARENT_UOW_ID,
       CHILD.PARENT_ACTIVITY_ID, CHILD.UOW_ID,
       CHILD.ACTIVITY_ID, CHILDSTMT.STMT_TEXT, CHILD.ACT_EXEC_TIME
FROM RAH PARENT, ACTIVITY_DB2ACTIVITIES CHILD,
     ACTIVITYSTMT_DB2ACTIVITIES CHILDSTMT
WHERE PARENT.APPL_ID = CHILD.APPL_ID AND
      CHILD.APPL_ID = CHILDSTMT.APPL_ID AND
      PARENT.UOW_ID = CHILD.PARENT_UOW_ID AND
      CHILD.UOW_ID = CHILDSTMT.UOW_ID AND
      PARENT.ACTIVITY_ID = CHILD.PARENT_ACTIVITY_ID AND
      CHILD.ACTIVITY_ID = CHILDSTMT.ACTIVITY_ID AND
      PARENT.LEVEL < 64
)
SELECT UOW_ID, ACTIVITY_ID, SUBSTR(STMT_TEXT,1,40),
       ACT_EXEC_TIME AS
       LIFE_TIME
FROM RAH

```

以下のサンプル照会では、表関数を使用します。

```

SELECT * FROM TABLE(SHOWCAPTUREDACTIVITY(1, 2, 3))
AS ACTS ORDER BY UOW_ID, ACTIVITY_ID

```

### 使用上の注意

アクティブなアクティビティ・イベント・モニターがない場合、SQLSTATE 01H53 の SQL1633W が戻されます。

アクティビティ情報は、アクティビティのコーディネーター・メンバーでのみ収集されます。

### WLM\_COLLECT\_STATS - ワークロード管理統計の収集およびリセット

WLM\_COLLECT\_STATS プロシージャは、サービス・クラス、ワークロード、作業クラス、およびしきい値キューの統計を収集し、統計イベント・モニターに書き込みます。また、このプロシージャは、サービス・クラス、ワークロード、作業クラス、およびしきい値キューの統計のリセットも行います。

アクティブな統計イベント・モニターがない場合、プロシージャは統計のリセットのみを行います。

### 構文

```

▶▶ WLM_COLLECT_STATS ( ( wait [, statistics_timestamp ] ) ) ▶▶

```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *wait*

このプロシージャが統計収集とリセットを開始した直後に戻るかどうかを指定する、タイプ CHAR のオプション入力引数。 'Y' を指定した場合、プロシージャは、すべての統計が統計イベント・モニター表に書き込まれてフラッシュされるまで戻りません。それ以外の場合、プロシージャは統計収集およびリセットを開始した直後に戻ります。



*statistics\_timestamp*

統計収集の開始のタイム・スタンプ値を戻すタイプ **TIMESTAMP** のオプション  
出力引数。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

例 1: **WLM\_COLLECT\_STATS** を呼び出して統計収集とリセットを開始します。

```
CALL WLM_COLLECT_STATS()
```

以下はこの照会の出力例です。

```
Return Status = 0
```

例 2: **WLM\_COLLECT\_STATS** を呼び出して、統計を収集およびリセットしますが、データが統計イベント・モニター表に書き込まれるまで戻らないようにします。

```
CALL WLM_COLLECT_STATS('Y', ::collect_timestamp)
```

以下はこの照会の出力例です。

```
Return Status = 0
```

例 3: 別の呼び出しが進行中に、**WLM\_COLLECT\_STATS** を呼び出して、統計を収集およびリセットする。

```
CALL WLM_COLLECT_STATS()
```

以下はこの照会の出力例です。

```
SQL1632W The collect and reset statistics request was ignored because  
another collect and reset statistics request is already in progress.
```

## 使用上の注意

**WLM\_COLLECT\_STATS** プロシージャは、**wlm\_collect\_int** データベース構成パラメーターで定義された間隔で自動的に行われるものと同じ収集操作 (アクティブな統計イベント・モニターへの統計の送信) およびリセット操作を実行します。

別の収集およびリセット要求の進行中 (例えばプロシージャの別の呼び出しの実行中または自動収集の発生中) にプロシージャを呼び出すと、**SQL1632W** が **SQLSTATE 01H53** とともに返され、新しい要求は無視されます。

非同期モードでは、WLM\_COLLECT\_STATS プロシージャは、収集およびリセット・プロセスのみ開始します。このプロシージャは、すべての統計がアクティブな統計イベント・モニターに書き込まれる前に呼び出し元に戻る場合があります。統計の収集およびリセットが発生する頻度に応じて、WLM\_COLLECT\_STATS プロシージャの呼び出し（これ自体がアクティビティ）は統計において、前の収集間隔または直前に開始された新規の収集間隔のいずれかでカウントされます。

同期モードでは、WLM\_COLLECT\_STATS プロシージャは、統計収集が完了し、すべての統計がアクティブな統計イベント・モニターの表に書き込まれるまで戻りません。統計収集が開始した時のタイム・スタンプは、*statistics\_timestamp* 出力パラメーターを介して戻されます。

## WLM\_GET\_CONN\_ENV - 接続のアクティビティ・データ収集の設定の取得

WLM\_GET\_CONN\_ENV 表関数は、特定の接続の、アクティビティ・データおよびセクション実行時統計の収集を制御する設定の値を戻します。この表関数を使用することで、WLM\_SET\_CONN\_ENV ストアード・プロシージャによって適用される設定の現行値を検査できます。

### 構文

```
▶▶ WLM_GET_CONN_ENV (—application_handle—) ▶▶
```

### パラメーター

*application\_handle*

情報が戻される接続のアプリケーション・ハンドルを指定するタイプ BIGINT の入力引数。 NULL 値を使用することで、プロシージャが呼び出された接続を示すことができます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

以下の照会は、現行接続のアクティビティが収集されているかどうかを検査します。

```

SELECT application_handle,
       xmlparse(document details preserve whitespace)
FROM TABLE (
  WLM_GET_CONN_ENV(
    cast(NULL as bigint))
  ) connenv

```

以下はこの照会の出力例です。

```

APPLICATION_HANDLE  DETAILS
-----
7 <wlm_conn_env
  xmlns=http://www.ibm.com/xmlns/prod/db2/mon
  release="9070100">
  <collectactdata>NONE</collectactdata>
  <collectactpartition>COORDINATOR</collectactpartition>
  <collectsectionactuals>NONE</collectsectionactuals>
</wlm_conn_env>

```

### 使用上の注意

WLM\_GET\_CONN\_ENV 表関数は、接続の WLM 環境情報を 1 つの XML 文書として戻すため、出力フォーマットを最大限柔軟性のあるものにします。出力は XML パーサーで直接解析でき、XMLTABLE 関数でリレーショナル形式に変換することもできます。

DETAILS 列に戻される XML 文書のスキーマは、ファイル sqllib/misc/DB2MonRoutines.xsd で入手できます。詳細は、ファイル sqllib/misc/DB2MonCommon.xsd 内にあります。

### 戻される情報

表 256. WLM\_GET\_CONN\_ENV について戻される情報

列名	データ・タイプ	説明
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル
DETAILS	BLOB(8M)	接続環境の詳細を含む XML 文書。エレメントの説明については、本書の 1166 ページの表 257を参照してください。

## 戻される詳細設定

表 257. WLM\_GET\_CONN\_ENV について戻される詳細メトリック

エレメント名	データ・タイプ	説明または対応するモニター・エレメント
collectactdata	xs:string(255)	<p>収集されているアクティビティ・データの種別を指定します (存在する場合)。可能な値は以下のとおりです。</p> <ul style="list-style-type: none"> <li>• NONE</li> <li>• WITHOUT DETAILS</li> <li>• WITH DETAILS</li> <li>• WITH DETAILS, SECTION</li> <li>• WITH DETAILS, SECTION AND VALUES</li> <li>• WITH DETAILS AND VALUES</li> </ul> <p>これらのオプションについて詳しくは、WLM_SET_CONN_ENV プロシージャーに関する情報を参照してください。</p>
collectactpartition	xs:string(255)	<p>どこでアクティビティ・データを収集するかを指定します。可能な値は以下のとおりです。</p> <ul style="list-style-type: none"> <li>• COORDINATOR</li> <li>• ALL</li> </ul> <p>これらのオプションについて詳しくは、WLM_SET_CONN_ENV プロシージャーに関する情報を参照してください。</p>
collectsectionactuals	xs:string(255)	<p>セクション実行時統計を収集するかどうかを指定します。使用できる値は以下のとおりです。</p> <ul style="list-style-type: none"> <li>• NONE</li> <li>• BASE</li> </ul> <p>これらのオプションについて詳しくは、WLM_SET_CONN_ENV プロシージャーに関する情報を参照してください。</p>

### WLM\_GET\_QUEUE\_STATS 表関数 - しきい値キュー統計を戻す

WLM\_GET\_QUEUE\_STATS 関数は、すべてのアクティブ・メンバー上の 1 つ以上のしきい値キューの基本統計を戻します。この関数は、しきい値キューごとに 1 行の統計を戻します。

#### 構文

```

▶▶—WLM_GET_QUEUE_STATS—(—threshold_predicate—,—threshold_domain—,—
▶—threshold_name—,—threshold_id—)—▶▶

```

スキーマは SYSPROC です。

## 表関数パラメーター

### *threshold\_predicate*

しきい値述部を指定する、タイプ VARCHAR(27) の入力引数。可能な値は、以下のとおりです。

#### *CONCDBC*

並行データベース・コーディネーター・アクティビティしきい値

#### *DBCINN*

データベース・メンバー接続合計しきい値

#### *SCCINN*

サービス・クラス・メンバー接続合計しきい値

引数が NULL または空ストリングである場合、他の基準を満たすすべてのしきい値についてデータが戻されます。

*threshold\_predicate* の値は、SYSCAT.THRESHOLDS ビューの THRESHOLDPREDICATE 列の値と一致します。

### *threshold\_domain*

しきい値ドメインを指定する、タイプ VARCHAR(18) の入力引数。可能な値は、以下のとおりです。

*DB* データベース

*SB* サービス・サブクラス

*SP* サービス・スーパークラス

*WA* 作業アクション・セット

引数が NULL または空ストリングである場合、他の基準を満たすすべてのしきい値についてデータが戻されます。

*threshold\_domain* の値は、SYSCAT.THRESHOLDS ビューの DOMAIN 列の値と一致します。

### *threshold\_name*

しきい値の名前を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、他の基準を満たすすべてのしきい値についてデータが戻されます。 *threshold\_name* の値は、SYSCAT.THRESHOLDS ビューの THRESHOLDNAME 列の値と一致します。

### *threshold\_id*

しきい値 ID を指定する、タイプ INTEGER の入力引数。引数が NULL または -1 である場合、他の基準を満たすすべてのしきい値についてデータが戻されず。 *threshold\_id* の値は、SYSCAT.THRESHOLDS ビューの THRESHOLDID 列の値と一致します。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限

- SQLADM 権限
- WLMADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

以下の照会は、システム上のすべてのキューの基本統計を、すべてのメンバーにわたって表示します。

```
SELECT substr(THRESHOLD_NAME, 1, 6) THRESHNAME,
       THRESHOLD_PREDICATE,
       THRESHOLD_DOMAIN,
       MEMBER MEMB,
       QUEUE_SIZE_TOP,
       QUEUE_TIME_TOTAL,
       QUEUE_ASSIGNMENTS_TOTAL QUEUE_ASSIGN
FROM table(WLM_GET_QUEUE_STATS('',' ', '-1')) as QSTATS
```

出力例を以下に示します。

```
THRESHNAME THRESHOLD_PREDICATE THRESHOLD_DOMAIN ...
-----
LIMIT1     CONCDBC                      DB                ...
LIMIT2     SCCONN                       SP                ...
LIMIT3     DBCONN                       DB                ...

... MEMB QUEUE_SIZE_TOP QUEUE_TIME_TOTAL QUEUE_ASSIGN
... -----
... 0      12          1238540          734
... 0      4           741249          24
... 0      7           412785          128
```

## 使用上の注意

この関数は、(1 つのメンバー上の) キュー全体や (1 つ以上のキューの) メンバー全体のデータ集約は行いません。ただし、上の例で示された SQL 照会を使用すると、データを集約することができます。

## 戻される情報

表 258. WLM\_GET\_QUEUE\_STATS について戻される情報

列名	データ・タイプ	説明
THRESHOLD_PREDICATE	VARCHAR(27)	threshold_predicate - しきい値述部モニター・エレメント
THRESHOLD_DOMAIN	VARCHAR(18)	threshold_domain - しきい値ドメイン・モニター・エレメント
THRESHOLD_NAME	VARCHAR(128)	threshold_name - しきい値名モニター・エレメント
THRESHOLD_ID	INTEGER	thresholdid - しきい値 ID モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - サービス・スーパークラス名モニター・エレメント

表 258. WLM\_GET\_QUEUE\_STATS について戻される情報 (続き)

列名	データ・タイプ	説明
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - サービス・サブクラス名モニター・エレメント
WORK_ACTION_SET_NAME	VARCHAR(128)	work_action_set_name - 作業アクション・セット名モニター・エレメント
WORK_CLASS_NAME	VARCHAR(128)	work_class_name - 作業クラス名モニター・エレメント
WORKLOAD_NAME	VARCHAR(128)	workload_name - ワークロード名 : モニター・エレメント
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ・モニター・エレメント
QUEUE_SIZE_TOP	INTEGER	queue_size_top - キュー・サイズの最上位モニター・エレメント
QUEUE_TIME_TOTAL	BIGINT	queue_time_total - キュー時間の合計モニター・エレメント
QUEUE_ASSIGNMENTS_TOTAL	BIGINT	queue_assignments_total - キュー割り当ての合計モニター・エレメント
QUEUE_SIZE_CURRENT	INTEGER	キュー内の接続またはアクティビティーの数。
QUEUE_TIME_LATEST	BIGINT	最後の接続またはアクティビティーがキューをそのままにしておくためにキューで費やした時間。単位はミリ秒です。
QUEUE_EXIT_TIME_LATEST	TIMESTAMP	最後の接続またはアクティビティーがキューをそのままにしておいた時間。
THRESHOLD_CURRENT_CONCURRENCY	INTEGER	しきい値に従って現在実行中の接続またはアクティビティーの数。
THRESHOLD_MAX_CONCURRENCY	INTEGER	しきい値によって現在実行中の接続またはアクティビティーの最大数。
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

### WLM\_GET\_SERVICE\_CLASS\_AGENTS 表関数 - サービス・クラスで実行中のエージェントのリスト

WLM\_GET\_SERVICE\_CLASS\_AGENTS 関数は、指定されたサービス・クラスで (または指定されたアプリケーションの代わりに) 稼働している指定されたメンバー上のエージェント、fenced モード・プロセス (db2fmp プロセス)、およびシステム・エンティティーのリストを戻します。システム・エンティティーは、非エージェント・スレッドおよびプロセス (ページ・クリーナーおよびプリフェッチャーなど) です。

戻される可能性のある情報の完全なリストは、1172 ページの表 259 を参照してください。

## 構文

```
►—WLM_GET_SERVICE_CLASS_AGENTS—(—service_superclass_name—,—————►  
►—service_subclass_name—,—application_handle—,—member—)—————►
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *service\_superclass\_name*

現在接続されているデータベースのサービス・スーパークラスの名前を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのスーパークラスについてデータが取得されます。

### *service\_subclass\_name*

スーパークラス内の特定のサブクラスを参照する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのサブクラスについてデータが取得されます。

### *application\_handle*

エージェント情報が戻されるアプリケーション・ハンドルを指定する、タイプ BIGINT の入力引数。引数が NULL である場合、データベース内のすべてのアプリケーションについてデータが取得されます。アプリケーション・ハンドルが 0 の場合、システム・エンティティーのみ戻されます。

### *member*

現在接続されているデータベースと同じインスタンス内のメンバー番号を指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

## デフォルトの PUBLIC 特権

なし

## 例 1

以下の照会は、すべてのデータベース・メンバーについてアプリケーション・ハンドル 1 に関連付けられたエージェントのリストを戻します。 **LIST APPLICATIONS** コマンドまたは **WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES** 表関数を使用して、アプリケーション・ハンドルを判別することができます。



```

SELECT SUBSTR(CHAR(APPLICATION_HANDLE),1,7) AS APPHANDLE,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       SUBSTR(CHAR(AGENT_TID),1,9) AS AGENT_TID,
       SUBSTR(AGENT_TYPE,1,11) AS AGENTTYPE,
       SUBSTR(AGENT_STATE,1,10) AS AGENTSTATE,
       SUBSTR(REQUEST_TYPE,1,12) AS REQTYPE,
       SUBSTR(CHAR(UOW_ID),1,6) AS UOW_ID,
       SUBSTR(CHAR(ACTIVITY_ID),1,6) AS ACT_ID
FROM TABLE(WLM_GET_SERVICE_CLASS_AGENTS(CAST(NULL AS VARCHAR(128)),
     CAST(NULL AS VARCHAR(128)), 1, -2)) AS SCDETAILS
ORDER BY APPHANDLE, MEMB, AGENT_TID

```

出力例を以下に示します。

APPHANDLE	MEMB	AGENT_TID	AGENTTYPE	AGENTSTATE	REQTYPE	UOW_ID	ACT_ID
1	0	3	COORDINATOR	ACTIVE	FETCH	1	5
1	0	4	SUBAGENT	ACTIVE	SUBSECTION:1	1	5
1	1	2	SUBAGENT	ACTIVE	SUBSECTION:2	1	5

この出力は、UOW ID 1 およびアクティビティ ID 5 のアクティビティの代わりに作動している、メンバー 0 上のコーディネーター・エージェントとサブエージェント、およびメンバー 1 上のサブエージェントを示しています。AGENTTYPE 列の値 COORDINATOR に対しては、REQTYPE 列に FETCH の値があります (これは、メインまたは初期要求タイプを示しています)。これは、要求のタイプがコーディネーター・エージェントに対するフェッチ要求であることを意味しています。

## 例 2

以下の照会は、エージェントがどのロックを待機しているかを判別します。

```

select event_object, event_type, event_state, varchar(event_object_name, 30)
as event_object_name
from table(WLM_GET_SERVICE_CLASS_AGENTS('','cast(NULL as bigint), -1)) as t

```

出力例を以下に示します。

EVENT_OBJECT	EVENT_TYPE	EVENT_STATE	EVENT_OBJECT_NAME
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	WAIT	IDLE	-
LOCK	ACQUIRE	IDLE	02000500000000000000000000000054
ROUTINE	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-

21 record(s) selected.

後で同じ照会を使用すると、WLM しきい値によってエージェントがキューに入れられたことが示されます。

EVENT_OBJECT	EVENT_TYPE	EVENT_STATE	EVENT_OBJECT_NAME
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
WLM_QUEUE	WAIT	IDLE	MYCONCDBCOORDTH
ROUTINE	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-

21 record(s) selected.

### 使用上の注意

これらのパラメーターの作用については、論理積 (AND) が取られます。つまり、矛盾する入力パラメーターを指定する (例えば、サービス・スーパークラス SUP\_A とサブクラス SUB\_B を、SUB\_B が SUP\_A のサブクラスにならないように指定する) 場合、行は戻されません。

### 戻される情報

表 259. WLM\_GET\_SERVICE\_CLASS\_AGENTS によって戻される情報

列名	データ・タイプ	説明
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	service_superclass_name - サービス・スーパークラス名モニター・エレメント
SERVICE_SUBCLASS_NAME	VARCHAR (128)	service_subclass_name - サービス・サブクラス名モニター・エレメント
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル・モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
ENTITY	VARCHAR (32)	以下の値のいずれか。 <ul style="list-style-type: none"> <li>エンティティーのタイプがエージェントである場合、値は db2agent です。</li> <li>エンティティーのタイプが fenced モード・プロセスである場合、値は db2fmp (pid) です。ここで、pid は fenced モード・プロセスのプロセス ID です。</li> <li>それ以外の場合、値はシステム・エンティティーの名前です。</li> </ul>

表 259. WLM\_GET\_SERVICE\_CLASS\_AGENTS によって戻される情報 (続き)

列名	データ・タイプ	説明
WORKLOAD_NAME	VARCHAR (128)	workload_name - ワークロード名 : モニター・エレメント
WORKLOAD_OCCURRENCE_ID	INTEGER	workload_occurrence_id - ワークロード・オカレンス ID モニター・エレメント
UOW_ID	INTEGER	uow_id - 作業単位 ID モニター・エレメント
ACTIVITY_ID	INTEGER	activity_id - アクティビティ ID モニター・エレメント
PARENT_UOW_ID	INTEGER	parent_uow_id - 親作業単位 ID モニター・エレメント
PARENT_ACTIVITY_ID	INTEGER	parent_activity_id - 親アクティビティ ID モニター・エレメント
AGENT_TID	BIGINT	agent_tid - エージェント・スレッド ID モニター・エレメント
AGENT_TYPE	VARCHAR (32)	<p>エージェント・タイプ。エージェント・タイプは以下のとおりです。</p> <ul style="list-style-type: none"> <li>• COORDINATOR</li> <li>• OTHER</li> <li>• PDBSUBAGENT</li> <li>• SMPSUBAGENT</li> </ul> <p>値が COORDINATOR である場合、エージェント ID はコンソレータ環境で変わることがあります。</p>
SMP_COORDINATOR	INTEGER	エージェントが SMP コーディネーターかどうかを示します。「はい」の場合は 1、「いいえ」の場合は 0。
AGENT_SUBTYPE	VARCHAR (32)	<p>エージェント・サブタイプ。可能なサブタイプは以下のとおりです。</p> <ul style="list-style-type: none"> <li>• DSS</li> <li>• OTHER</li> <li>• RPC</li> <li>• SMP</li> </ul>
AGENT_STATE	VARCHAR (32)	<p>エージェントが関連付けられているか、アクティブであるかを示します。可能な値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• ASSOCIATED</li> <li>• ACTIVE</li> </ul>
EVENT_TYPE	VARCHAR (32)	<p>このエージェントによって最後に処理されたイベントのタイプ。可能な値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• ACQUIRE</li> <li>• PROCESS</li> <li>• WAIT</li> </ul> <p>この列に使用できる値についての詳細は、1177 ページの表 260 を参照してください。</p>

表 259. WLM\_GET\_SERVICE\_CLASS\_AGENTS によって戻される情報 (続き)

列名	データ・タイプ	説明
EVENT_OBJECT	VARCHAR (32)	<p>このエージェントによって最後に処理されたイベントのオブジェクト。可能な値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• COMPRESSION_DICTIONARY_BUILD</li> <li>• IMPLICIT_REBIND</li> <li>• INDEX_RECREATE</li> <li>• LOCK</li> <li>• LOCK_ESCALATION</li> <li>• QP_QUEUE</li> <li>• REMOTE_REQUEST</li> <li>• REQUEST</li> <li>• ROUTINE</li> <li>• WLM_QUEUE</li> </ul> <p>この列に使用できる値についての詳細は、1177 ページの表 260 を参照してください。</p>
EVENT_STATE	VARCHAR (32)	<p>このエージェントによって最後に処理されたイベントの状態。可能な値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• EXECUTING</li> <li>• IDLE</li> </ul> <p>この列に使用できる値についての詳細は、1177 ページの表 260 を参照してください。</p>
REQUEST_ID	VARCHAR (64)	<p>要求 ID。この値は、<i>application_handle</i> の値と組み合わせて指定される場合のみ固有です。この組み合わせを使用して、長い時間を要する 1 つの要求と複数の要求とを区別することができます。例えば、1 つの長いフェッチと複数のフェッチを区別するなどです。</p>

表 259. WLM\_GET\_SERVICE\_CLASS\_AGENTS によって戻される情報 (続き)

列名	データ・タイプ	説明
REQUEST_TYPE	VARCHAR (32)	<p>要求のタイプ。可能な値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• コーディネーター・エージェントの場合: <ul style="list-style-type: none"> <li>- CLOSE</li> <li>- COMMIT</li> <li>- COMPILE</li> <li>- DESCRIBE</li> <li>- EXCSQLSET</li> <li>- EXECIMMD</li> <li>- EXECUTE</li> <li>- FETCH</li> <li>- INTERNAL <i>number</i> (<i>number</i> は内部定数の値)</li> <li>- OPEN</li> <li>- PREPARE</li> <li>- REBIND</li> <li>- REDISTRIBUTE</li> <li>- REORG</li> <li>- ROLLBACK</li> <li>- RUNSTATS</li> </ul> </li> <li>• AGENT_SUBTYPE が DSS または SMP であるサブエージェントの場合: <ul style="list-style-type: none"> <li>- サブセクション番号がゼロ以外の場合は、 SUBSECTION:<i>subsection number</i> の形式のサブセクション番号。そうでない場合は NULL を戻します。</li> </ul> </li> </ul>

表 259. WLM\_GET\_SERVICE\_CLASS\_AGENTS によって戻される情報 (続き)

列名	データ・タイプ	説明
REQUEST_TYPE (続く)	VARCHAR (32)	<ul style="list-style-type: none"> <li>• AGENT_SUBTYPE が RPC であるサブエージェントの場合: <ul style="list-style-type: none"> <li>- ABP</li> <li>- CATALOG</li> <li>- INTERNAL</li> <li>- REORG</li> <li>- RUNSTATS</li> <li>- WLM</li> </ul> </li> <li>• SUBTYPE が OTHER であるサブエージェントの場合: <ul style="list-style-type: none"> <li>- ABP</li> <li>- APP_RBSVPT</li> <li>- APP_RELSVPT</li> <li>- BACKUP</li> <li>- CLOSE</li> <li>- EXTERNAL_RBSVPT</li> <li>- EVMON</li> <li>- FORCE</li> <li>- FORCE_ALL</li> <li>- INTERNAL <i>number</i> (<i>number</i> は内部定数の値)</li> <li>- INTERRUPT</li> <li>- NOOP (要求がない場合)</li> <li>- QP</li> <li>- REDISTRIBUTE</li> <li>- STMT_RBSVPT</li> <li>- STOP_USING</li> <li>- UPDATE_DBM_CFG</li> <li>- WLM</li> </ul> </li> </ul>
NESTING_LEVEL	INTEGER	nesting_level - ネスティング・レベル・モニター・エレメント
INVOCATION_ID	INTEGER	invocation_id - 呼び出し ID モニター・エレメント
ROUTINE_ID	INTEGER	routine_id - ルーチン ID モニター・エレメント
EVENT_OBJECT_NAME	VARCHAR (1024)	イベント・オブジェクト名。 EVENT_OBJECT の値が LOCK である場合、この列の値は、エージェントが待機するロックの名前です。 EVENT_OBJECT の値が WLM_QUEUE である場合、この列の値は、エージェントがキューに入れている WLM しきい値の名前です。それ以外の場合、値は NULL です。
APPLICATION_NAME	VARCHAR (128)	appl_name - アプリケーション名
APPLICATION_ID	VARCHAR (128)	appl_id - アプリケーション ID

表 259. WLM\_GET\_SERVICE\_CLASS\_AGENTS によって戻される情報 (続き)

列名	データ・タイプ	説明
CLIENT_PID	BIGINT	client_pid - クライアント・プロセス ID
SESSION_AUTH_ID	VARCHAR (128)	session_auth_id - セッション許可 ID
REQUEST_START_TIME	TIMESTAMP	エージェントが現在処理中の要求の処理を開始した時刻
AGENT_STATE_LAST_UPDATE_TIME	TIMESTAMP	エージェントによって処理されているイベントが最後に変更された時刻。エージェントによって現在処理されているイベントは、EVENT_TYPE、EVENT_OBJECT、および EVENT_STATE 列で示されます。
EXECUTABLE_ID	VARCHAR (32) FOR BIT DATA	executable_id - 実行可能 ID モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

注: EVENT\_STATE、EVENT\_TYPE、EVENT\_OBJECT および EVENT\_OBJECT\_NAME 列値の可能な組み合わせを、以下の表にリストします。

表 260. EVENT\_STATE、EVENT\_TYPE、EVENT\_OBJECT および EVENT\_OBJECT\_NAME 列値の可能な組み合わせ

イベント記述	EVENT_STATE 値	EVENT_TYPE 値	EVENT_OBJECT 値	EVENT_OBJECT_NAME 値
ロックの獲得	IDLE	ACQUIRE	LOCK	ロック名
ロックのエスカレート	EXECUTING	PROCESS	LOCK_ESCALATION	NULL
要求の処理	EXECUTING	PROCESS	REQUEST	NULL
新規要求の待機	IDLE	WAIT	REQUEST	NULL
リモート・メンバーで処理される要求の待機	IDLE	WAIT	REMOTE_REQUEST	NULL
WLM threshold キューの待機	IDLE	WAIT	WLM_QUEUE	しきい値名
ルーチンの処理	EXECUTING	PROCESS	ROUTINE	NULL
索引の再作成	EXECUTING	PROCESS	INDEX_RECREATE	NULL
コンプレッション・ディクショナリーの作成	EXECUTING	PROCESS	COMP_DICT_BLD	NULL
暗黙的な再バインド	EXECUTING	PROCESS	IMPLICIT_REBIND	NULL

## WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES - ワークロード・オカレンスのリスト

WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES 関数は、特定のメンバー上の指定されたサービス・クラスで実行されているすべてのワークロード・オカレンスのリストを戻します。ワークロード・オカレンスとは、属性がワークロードの定義と一致しており、そのためにワークロードに関連付けられた、またはワークロードに割り当てられた特定のデータベース接続です。

戻される可能性のある情報の完全なリストは、1179 ページの表 261 を参照してください。

## 構文

```
▶▶—WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES—(—service_superclass_name—,————▶▶  
▶—service_subclass_name—, —member—)————▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *service\_superclass\_name*

現在接続されているデータベースのサービス・スーパークラスの名前を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、他のパラメーターの値と一致する、データベース内のすべてのスーパークラスについてデータが取得されます。

### *service\_subclass\_name*

ワークロード・オカレンスのターゲット・サービス・サブクラス。このワークロード・オカレンスによってサブミットされる作業は、別のサブクラスにマップまたは再マップされるアクティビティーを除いて、すべてターゲット・サービス・スーパークラスの下でのこのサービス・サブクラスで実行されます。

### *member*

現在接続されているデータベースと同じインスタンス内のメンバーの番号を指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

システム全体で実行されているワークロード・オカレンスを管理者が調べるには、*service\_superclass\_name* および *service\_subclass\_name* に NULL 値または空ストリングを、*member* に -2 をそれぞれ指定して、WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES 関数を呼び出すことができます。

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,  
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,  
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,  
       SUBSTR(CHAR(COORD_MEMBER),1,4) AS COORDMEMB,
```



```

SUBSTR(CHAR(APPLICATION_HANDLE),1,7) AS APPHNDL,
SUBSTR(WORKLOAD_NAME,1,22) AS WORKLOAD_NAME,
SUBSTR(CHAR(WORKLOAD_OCCURRENCE_ID),1,6) AS WLO_ID
FROM TABLE(WLM_GET_SERVICE_CLASS WORKLOAD_OCCURRENCES
(CAST(NULL AS VARCHAR(128)), CAST(NULL AS VARCHAR(128)), -2))
AS SCINFO
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, MEMB, APPHNDL,
WORKLOAD_NAME, WLO_ID

```

システムに 4 つのデータベース・メンバーがあり、現時点で 2 つのワークロードを実行していると想定すると、上記の照会には以下のような結果を生成します。

```

SUPERCLASS_NAME  SUBCLASS_NAME  MEMB COORDMEMB ...
-----
SYSDEFAULTMAINTENAN  SYSDEFAULTSUBCLASS  0    0    ...
SYSDEFAULTSYSTEMCLA  SYSDEFAULTSUBCLASS  0    0    ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS  0    0    ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS  0    0    ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS  1    0    ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS  1    0    ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS  2    0    ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS  2    0    ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS  3    0    ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS  3    0    ...

... APPHNDL WORKLOAD_NAME          WLO_ID
... -----
... -          -                    -
... -          -                    -
... 1          SYSDEFAULTUSERWORKLOAD 1
... 2          SYSDEFAULTUSERWORKLOAD 2
... 1          SYSDEFAULTUSERWORKLOAD 1
... 2          SYSDEFAULTUSERWORKLOAD 2
... 1          SYSDEFAULTUSERWORKLOAD 1
... 2          SYSDEFAULTUSERWORKLOAD 2
... 1          SYSDEFAULTUSERWORKLOAD 1
... 2          SYSDEFAULTUSERWORKLOAD 2
... 1          SYSDEFAULTUSERWORKLOAD 1
... 2          SYSDEFAULTUSERWORKLOAD 2

```

### 使用上の注意

これらのパラメーターの作用については、論理積 (AND) が取られます。つまり、矛盾する入力パラメーターを指定する (例えば、サービス・スーパークラス SUP\_A とサブクラス SUB\_B を、SUB\_B が SUP\_A のサブクラスにならないように指定する) 場合、行は戻されません。

**注:** ワークロード・オカレンスについて報告される統計 (例えば、coord\_act\_completed\_total) が、対応するワークロード統計と結合されると、ワークロード・オカレンスについて報告される統計が各作業単位の初めにリセットされます。

### 戻される情報

表 261. WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES について戻される情報

列名	データ・タイプ	説明
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - サービス・スーパークラス名モニター・エレメント
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - サービス・サブクラス名モニター・エレメント

表 261. WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES について戻される情報 (続き)

列名	データ・タイプ	説明
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
COORD_PARTITION_NUM	SMALLINT	coord_partition_num - コーディネーター・パーティション番号モニター・エレメント
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル・モニター・エレメント
WORKLOAD_NAME	VARCHAR(128)	workload_name - ワークロード名 : モニター・エレメント
WORKLOAD_OCCURRENCE_ID	INTEGER	workload_occurrence_id - ワークロード・オカレンス ID モニター・エレメント
UOW_ID	INTEGER	uow_id - 作業単位 ID モニター・エレメント
WORKLOAD_OCCURRENCE_STATE	VARCHAR(32)	workload_occurrence_state - ワークロード・オカレンスの状態モニター・エレメント
SYSTEM_AUTH_ID	VARCHAR(128)	system_auth_id - システム許可 ID モニター・エレメント
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - セッション許可 ID モニター・エレメント
APPLICATION_NAME	VARCHAR(128)	appl_name - アプリケーション名モニター・エレメント
CLIENT_WRKSTNNAME	VARCHAR(255)	client_wrkstnname - クライアント・ワークステーション名モニター・エレメント
CLIENT_ACCTNG	VARCHAR(255)	client_acctng - クライアント・アカウントティング・ストリング・モニター・エレメント
CLIENT_USER	VARCHAR(255)	このワークロード・オカレンスの CLIENT_USERID 特殊レジスターの現行値。
CLIENT_APPLNAME	VARCHAR(255)	client_applname - クライアント・アプリケーション名モニター・エレメント
COORD_ACT_COMPLETED_TOTAL	INTEGER	coord_act_completed_total - コーディネーター・アクティビティー完了総数モニター・エレメント
COORD_ACT_ABORTED_TOTAL	INTEGER	coord_act_aborted_total - コーディネーター・アクティビティー打ち切り総数モニター・エレメント
COORD_ACT_REJECTED_TOTAL	INTEGER	coord_act_rejected_total - コーディネーター・アクティビティー・リジェクト総数モニター・エレメント
CONCURRENT_ACT_TOP	INTEGER	concurrent_act_top - 並行アクティビティーの最上位モニター・エレメント
ADDRESS	VARCHAR(255)	address - 接続の開始元となった IP アドレス
APPL_ID	VARCHAR(128)	appl_id - アプリケーション ID
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
COORD_MEMBER	SMALLINT	coord_member - コーディネーター・メンバー・モニター・エレメント

## WLM\_GET\_SERVICE\_SUBCLASS\_STATS 表関数 - サービス・サブクラスの統計を戻す

WLM\_GET\_SERVICE\_SUBCLASS\_STATS 関数は、1 つ以上のサービス・サブクラスの基本統計を戻します。

戻すことができる情報の全リストについては、1183 ページの表 262を参照してください。

### 構文

```
▶▶—WLM_GET_SERVICE_SUBCLASS_STATS—(—service_superclass_name—,—————▶▶  
▶—service_subclass_name—, —member—)—————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *service\_superclass\_name*

現在接続されているデータベースのサービス・スーパークラスの名前を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのスーパークラスについてデータが取得されます。

#### *service\_subclass\_name*

現在接続されているデータベースのサービス・サブクラスの名前を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのサブクラスについてデータが取得されます。

#### *member*

現在接続されているデータベースと同じインスタンス内の有効なメンバー番号を指定する、タイプ INTEGER の入力引数。現行のメンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が自動的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

### デフォルトの PUBLIC 特権

なし

## 例

例 1: すべてのアクティビティーは実行前に DB2 サービス・クラスに対してマップされる必要があるため、サービス・クラス統計表関数を使用し、すべてのメンバー上のすべてのサービス・クラスを照会して、システムの全体的な状態をモニターできます。以下の例では、NULL 値が *service\_superclass\_name* と *service\_subclass\_name* に渡されてすべてのサービス・クラスの統計を戻し、*member* には値 -2 が指定されてすべてのメンバーの統計を戻します。

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       CAST(COORD_ACT_LIFETIME_AVG / 1000 AS DECIMAL(9,3))
       AS AVGLIFETIME,
       CAST(COORD_ACT_LIFETIME_STDDEV / 1000 AS DECIMAL(9,3))
       AS STDDEVLIFETIME,
       SUBSTR(CAST(LAST_RESET AS VARCHAR(30)),1,16) AS LAST_RESET
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS(CAST(NULL AS VARCHAR(128)),
      CAST(NULL AS VARCHAR(128)), -2)) AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, MEMB
```

ステートメントは、以下の出力例で示されているように、アクティビティー存続期間の平均および標準偏差などのサービス・クラス統計を秒単位で戻します。

SUPERCLASS_NAME	SUBCLASS_NAME	MEMB	...
-----	-----	-----	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	1	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	2	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	3	...
...	AVGLIFETIME	STDDEVLIFETIME	LAST_RESET
...	-----	-----	-----
...	691.242	34.322	2006-07-24-11.44
...	644.740	22.124	2006-07-24-11.44
...	612.431	43.347	2006-07-24-11.44
...	593.451	28.329	2006-07-24-11.44

例 2: また、同じ表関数が、各メンバー上のサービス・クラスで実行しているコーディネーター・アクティビティーの平均並行性の最高値を示すこともできます。

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       CONCURRENT_ACT_TOP AS ACTTOP,
       CONCURRENT_WLO_TOP AS CONNTOP
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS(CAST(NULL AS VARCHAR(128)),
      CAST(NULL AS VARCHAR(128)), -2)) AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, MEMB
```

出力例を以下に示します。

SUPERCLASS_NAME	SUBCLASS_NAME	MEMB	ACTTOP	CONNTOP
-----	-----	-----	-----	-----
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	10	7
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	1	0	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	2	0	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	3	0	0

この表関数の出力内のアクティビティーの平均実行時間および回数を調べることで、特定のデータベースの各メンバーの負荷の概要を、的確に知ることができます。この表関数によって戻される高水準ゲージが大きく変わった場合は、システムの負荷の変化を示していることがあります。

例 3: アクティビティーが REMAP ACTIVITY TO アクションでしきい値を使用する場合、そのアクティビティーは、その存続時間中に複数のサービス・クラスで時間を費やすことがあります。以下の例に示すように、ACT\_REMAPPED\_IN 列と ACT\_REMAPPED\_OUT 列を調べると、一連のサービス・クラスを通過したアクティビティーの数を判別できます。

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       ACT_REMAPPED_IN AS MAPPED_IN,
       ACT_REMAPPED_OUT AS MAPPED_OUT
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS(CAST(NULL AS VARCHAR(128)),
      CAST(NULL AS VARCHAR(128)), -2)) AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME
```

出力例を以下に示します。

SUPERCLASS_NAME	SUBCLASS_NAME	MAPPED_IN	MAPPED_OUT
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	0
SUPERCLASS1	SYSDEFAULTSUBCLASS	0	0
SUPERCLASS1	SUBCLASS1	0	7
SUPERCLASS1	SUBCLASS2	7	0

### 使用上の注意

一部の統計は、対応するサービス・サブクラスの COLLECT AGGREGATE ACTIVITY DATA および COLLECT AGGREGATE REQUEST DATA パラメーターを NONE 以外の値に設定した場合のみ、戻されます。

WLM\_GET\_SERVICE\_SUBCLASS\_STATS 表関数は、サービス・サブクラスごとおよびメンバーごとに 1 行のデータを戻します。この関数は、(パーティション上の) サービス・クラス全体または (1 つ以上のサービス・クラスの) パーティション全体のデータ集約は行いません。ただし、SQL 照会を使用すると、データを集約することができます。

これらのパラメーターの作用については、論理積 (AND) が取られます。つまり、例えばスーパークラス SUPA と、SUPA のサブクラスでないサブクラス SUBB などの競合する入力パラメーターを指定した場合、行は戻されません。

### 戻される情報

表 262. WLM\_GET\_SERVICE\_SUBCLASS\_STATS について戻される情報

列名	データ・タイプ	説明
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - サービス・スーパークラス名モニター・エレメント
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - サービス・サブクラス名モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ・モニター・エレメント
COORD_ACT_COMPLETED_TOTAL	BIGINT	coord_act_completed_total - コーディネーター・アクティビティー完了総数モニター・エレメント

表 262. WLM\_GET\_SERVICE\_SUBCLASS\_STATS について戻される情報 (続き)

列名	データ・タイプ	説明
COORD_ACT_ABORTED_TOTAL	BIGINT	coord_act_aborted_total - コーディネーター・アクティビティー打ち切り総数モニター・エレメント
COORD_ACT_REJECTED_TOTAL	BIGINT	coord_act_rejected_total - コーディネーター・アクティビティー・リジェクト総数モニター・エレメント
CONCURRENT_ACT_TOP	INTEGER	concurrent_act_top - 並行アクティビティーの最上位モニター・エレメント
COORD_ACT_LIFETIME_TOP	BIGINT	coord_act_lifetime_top - コーディネーター・アクティビティー存続時間の最上位モニター・エレメント
COORD_ACT_LIFETIME_AVG	DOUBLE	coord_act_lifetime_avg - コーディネーター・アクティビティー平均存続期間モニター・エレメント
COORD_ACT_LIFETIME_STDDEV	DOUBLE	<p>最後のリセット以降、このサービス・サブクラスに関連付けられたネスト・レベル 0 のコーディネーター・アクティビティーの存続期間の標準偏差。サービス・クラスの COLLECT AGGREGATE ACTIVITY DATA パラメーターが NONE に設定される場合、列の値は NULL です。単位はミリ秒です。</p> <p>この標準偏差はコーディネーター・アクティビティーの存続期間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。</p> <p>サービス・サブクラスの COORD_ACT_LIFETIME_STDDEV 値は、完了前にサービス・サブクラスを通過したものの別のサブクラスに再マップされるアクティビティーの影響は受けません。</p>
COORD_ACT_EXEC_TIME_AVG	DOUBLE	coord_act_exec_time_avg - コーディネーター・アクティビティー平均実行時間モニター・エレメント

表 262. WLM\_GET\_SERVICE\_SUBCLASS\_STATS について戻される情報 (続き)

列名	データ・タイプ	説明
COORD_ACT_EXEC_TIME_STDDEV	DOUBLE	<p>最後のリセット以降、このサービス・サブクラスに関連付けられたネスト・レベル 0 のコーディネーター・アクティビティの実行時間の標準偏差。単位はミリ秒です。</p> <p>この標準偏差はコーディネーター・アクティビティの実行時間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。</p> <p>サービス・サブクラスの実行時間標準偏差は、完了前にサブクラスを通過したものの別のサブクラスに再マップされるアクティビティの影響は受けません。</p>
COORD_ACT_QUEUE_TIME_AVG	DOUBLE	<p>coord_act_queue_time_avg - コーディネーター・アクティビティ平均キュー時間モニター・エレメント</p>
COORD_ACT_QUEUE_TIME_STDDEV	DOUBLE	<p>最後のリセット以降、このサービス・サブクラスに関連付けられたネスト・レベル 0 のコーディネーター・アクティビティのキュー時間の標準偏差。サービス・クラスの COLLECT AGGREGATE ACTIVITY DATA パラメーターが NONE に設定される場合、列の値は NULL です。単位はミリ秒です。</p> <p>この標準偏差はコーディネーター・アクティビティのキュー時間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。</p> <p>キュー時間標準偏差のカウント対象は、アクティビティがキューに入れられたサービス・サブクラスだけです。</p>
NUM_REQUESTS_ACTIVE	BIGINT	<p>この表関数の実行時にサービス・サブクラスで実行している要求の数。</p>

表 262. WLM\_GET\_SERVICE\_SUBCLASS\_STATS について戻される情報 (続き)

列名	データ・タイプ	説明
NUM_REQUESTS_TOTAL	BIGINT	<p>最後のリセット以降、このサービス・サブクラスで実行を終了する要求の数。この終了状態は、アクティビティ内の要求のメンバーシップに関係なく、任意の要求に適用されます。サービス・クラスの COLLECT AGGREGATE ACTIVITY DATA パラメーターが NONE に設定される場合、列の値は NULL です。</p> <p>サービス・サブクラスの NUM_REQUESTS_TOTAL 値は、サービス・サブクラスを通過したものの、その中で完了しない要求の影響は受けません。</p>
REQUEST_EXEC_TIME_AVG	DOUBLE	<p>request_exec_time_avg - 要求の平均実行時間 モニター・エレメント</p>
REQUEST_EXEC_TIME_STDDEV	DOUBLE	<p>最後のリセット以降、このサービス・サブクラスに関連付けられた要求の実行時間の標準偏差。単位はミリ秒です。サービス・クラスの COLLECT AGGREGATE REQUEST DATA パラメーターが NONE に設定される場合、この列の値は NULL です。</p> <p>この標準偏差は要求実行時間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。</p> <p>サービス・サブクラスの実行時間標準偏差は、サブクラスを通過したものの、その中で完了しなかった要求の影響は受けません。</p>



表 262. WLM\_GET\_SERVICE\_SUBCLASS\_STATS について戻される情報 (続き)

列名	データ・タイプ	説明
REQUEST_EXEC_TIME_TOTAL	BIGINT	<p>最後のリセット以降、このサービス・サブクラスに関連付けられた要求の実行時間の合計。単位はミリ秒です。サービス・クラスの COLLECT AGGREGATE REQUEST DATA パラメーターが NONE に設定される場合、この列の値は NULL です。</p> <p>この合計は要求実行時間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。</p> <p>サービス・サブクラスの実行時間合計は、サブクラスを通過したものの、その中で完了しない要求の影響は受けません。</p>
ACT_REMAPPED_IN	BIGINT	最後のリセット以降にしい値 REMAP ACTIVITY アクションによってこのサービス・サブクラスに再マップされたアクティビティー数。
ACT_REMAPPED_OUT	BIGINT	最後のリセット以降にしい値 REMAP ACTIVITY アクションによってこのサービス・サブクラスから出て再マップされたアクティビティー数。
CONCURRENT_WLO_TOP	INTEGER	concurrent_wlo_top - 並行ワークロード・オカレンスの最上位モニター・エレメント
UOW_TOTAL_TIME_TOP	BIGINT	uow_total_time_top - 作業単位合計時間トップ・モニター・エレメント
UOW_THROUGHPUT	DOUBLE	<p>uow_throughput - 作業単位スループット : モニター・エレメント</p> <p>統計を最後にリセットして以降の作業単位のスループット。</p>
UOW_LIFETIME_AVG	DOUBLE	uow_lifetime_avg - 作業単位の平均存続期間 : モニター・エレメント
UOW_COMPLETED_TOTAL	BIGINT	uow_completed_total - 完了済みの合計作業単位 : モニター・エレメント
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間 : モニター・エレメント
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - ディスパッチャーの合計実行キュー時間 : モニター・エレメント
ACT_THROUGHPUT	DOUBLE	act_throughput - アクティビティー・スループット : モニター・エレメント
CPU_UTILIZATION	DOUBLE	cpu_utilization - CPU 使用率 : モニター・エレメント

表 262. WLM\_GET\_SERVICE\_SUBCLASS\_STATS について戻される情報 (続き)

列名	データ・タイプ	説明
APP_ACT_COMPLETED_TOTAL	BIGINT	app_act_completed_total - 成功した外部コーディネーター・アクティビティの総数モニター・エレメント
APP_ACT_ABORTED_TOTAL	BIGINT	app_act_aborted_total - 失敗した外部コーディネーター・アクティビティの総数モニター・エレメント
APP_ACT_REJECTED_TOTAL	BIGINT	app_act_rejected_total - リジェクトされた外部コーディネーター・アクティビティの総数モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## WLM\_GET\_SERVICE\_SUPERCLASS\_STATS - サービス・スーパークラスの統計を戻す

WLM\_GET\_SERVICE\_SUPERCLASS\_STATS 関数は、1 つ以上のサービス・スーパークラスの基本統計を戻します。

### 構文

```
▶▶—WLM_GET_SERVICE_SUPERCLASS_STATS—(—service_superclass_name—,—member—)——▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *service\_superclass\_name*

現在接続されているデータベースのサービス・スーパークラスの名前を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのスーパークラスについてデータが取得されます。

#### *member*

現在接続されているデータベースと同じインスタンス内の有効なメンバー番号を指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

## デフォルトの PUBLIC 特権

なし

### 例

以下の照会は、システム上のすべてのサービス・スーパークラスの基本統計を、すべてのデータベース・メンバーにわたって表示します。

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME, 1, 26) SERVICE_SUPERCLASS_NAME,  
       MEMBER,  
       LAST_RESET,  
       CONCURRENT_CONNECTION_TOP CONCURRENT_CONN_TOP  
FROM TABLE(WLM_GET_SERVICE_SUPERCLASS_STATS(' ', -2)) as SCSTATS
```

出力例を以下に示します。

```
SERVICE_SUPERCLASS_NAME  MEMBER ...  
-----  
SYSDEFAULTSYSTEMCLASS           0 ...  
SYSDEFAULTMAINTENANCECLASS      0 ...  
SYSDEFAULTUSERCLASS             0 ...  
  
... LAST_RESET                  CONCURRENT_CONN_TOP  
... -----  
... 2006-09-05-09.38.44.396788           0  
... 2006-09-05-09.38.44.396795           0  
... 2006-09-05-09.38.44.396796           1
```

### 使用上の注意

WLM\_GET\_SERVICE\_SUPERCLASS\_STATS 表関数は、サービス・スーパークラスおよびメンバーごとに 1 行のデータを戻します。この関数は、(メンバー上の) サービス・スーパークラス全体または (1 つ以上のサービス・スーパークラスの) メンバー全体のデータ集約は行いません。ただし、上の例で示された SQL 照会を使用すると、データを集約することができます。

### 戻される情報

表 263. WLM\_GET\_SERVICE\_SUPERCLASS\_STATS について戻される情報

列名	データ・タイプ	説明
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - サービス・スーパークラス名モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ・モニター・エレメント
CONCURRENT_CONNECTION_TOP	INTEGER	concurrent_connection_top - 並行接続の最上位モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## WLM\_GET\_WORK\_ACTION\_SET\_STATS - 作業アクション・セット統計を戻す

WLM\_GET\_WORK\_ACTION\_SET\_STATS 関数は、作業アクション・セットの統計を戻します。

### 構文

```
▶—WLM_GET_WORK_ACTION_SET_STATS—(—work_action_set_name—,—member—)————▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *work\_action\_set\_name*

統計を戻す作業アクション・セットを指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、すべての作業アクション・セットについて統計が戻されます。

#### *member*

現在接続されているデータベースと同じインスタンス内の有効なメンバー番号を指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

### デフォルトの PUBLIC 特権

なし

### 例

3 つの作業クラス、ReadClass、WriteClass、および LoadClass があると想定します。ReadClass に関連した作業アクションと LoadClass に関連した作業アクションはありますが、WriteClass に関連した作業アクションはありません。メンバー 0 上で現在実行されている、またはキューに入れられているアクティビティの数は次のとおりです。

- ReadClass クラス: 8
- WriteClass クラス: 4
- LoadClass クラス: 2
- 未割り当て: 3

```

SELECT SUBSTR(WORK_ACTION_SET_NAME,1,18) AS WORK_ACTION_SET_NAME,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       SUBSTR(WORK_CLASS_NAME,1,15) AS WORK_CLASS_NAME,
       LAST_RESET,
       SUBSTR(CHAR(ACT_TOTAL),1,14) AS ACT_TOTAL
FROM TABLE(WLM_GET_WORK_ACTION_SET_STATS
            (CAST(NULL AS VARCHAR(128)), -2)) AS WASSTATS
ORDER BY WORK_ACTION_SET_NAME, WORK_CLASS_NAME, MEMB

```

出力例を以下に示します。 WriteClass 作業クラスに関連した作業アクションはないため、その作業クラスが該当する 4 つのアクティビティーは、出力でアスタリスク (\*) によって示される人工的なクラスでカウントされます。どの作業クラスにも割り当てられなかった 3 つのアクティビティーも、人工的なクラスに組み込まれています。

WORK_ACTION_SET_NAME	MEMB	WORK_CLASS_NAME	LAST_RESET	ACT_TOTAL
AdminActionSet	0	ReadClass	2005-11-25-18.52.49.343000	8
AdminActionSet	1	ReadClass	2005-11-25-18.52.50.478000	0
AdminActionSet	0	LoadClass	2005-11-25-18.52.49.343000	2
AdminActionSet	1	LoadClass	2005-11-25-18.52.50.478000	0
AdminActionSet	0	*	2005-11-25-18.52.49.343000	7
AdminActionSet	1	*	2005-11-25-18.52.50.478000	0

## 戻される情報

表 264. WLM\_GET\_WORK\_ACTION\_SET\_STATS について戻される情報

列名	データ・タイプ	説明
WORK_ACTION_SET_NAME	VARCHAR(128)	work_action_set_name - 作業アクション・セット名モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ・モニター・エレメント
WORK_CLASS_NAME	VARCHAR(128)	work_class_name - 作業クラス名モニター・エレメント
ACT_TOTAL	BIGINT	act_total - アクティビティーの合計モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## WLM\_GET\_WORKLOAD

### \_OCCURRENCE \_ACTIVITIES - アクティビティーのリストを戻す

WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 関数は、指定されたメンバー上の指定されたアプリケーションによってサブミットされ、まだ完了していないすべてのアクティビティーのリストを戻します。

戻される可能性のある情報の完全なリストは、1193 ページの表 265 を参照してください。

## 構文

```

▶▶—WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES—(—application_handle—,—————▶

```

スキーマは SYSPROC です。

## 表関数パラメーター

### *application\_handle*

アクティビティのリストが戻されるアプリケーション・ハンドルを指定する、タイプ BIGINT の入力引数。引数が NULL である場合、データベース内のすべてのアプリケーションについてデータが取得されます。

### *member*

現在接続されているデータベースと同じインスタンス内の有効なメンバー番号を指定する、タイプ INTEGER の入力引数。現行メンバーであれば -1、すべてのメンバーであれば -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

### 認識済みのアプリケーション・ハンドルで現在実行中のアクティビティ

アプリケーション・ハンドルを識別した後、このアプリケーションで現在実行中のすべてのアクティビティを検索できます。 **LIST APPLICATIONS** コマンドを使用して判別されるアプリケーション・ハンドルが 1 であるアプリケーションのアクティビティをリストすることを管理者が望んでいるという場面を、例として考えてみましょう。管理者は以下の照会を実行します。

```
SELECT SUBSTR(CHAR(COORD_MEMBER),1,5) AS COORD,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       SUBSTR(CHAR(UOW_ID),1,5) AS UOWID,
       SUBSTR(CHAR(ACTIVITY_ID),1,5) AS ACTID,
       SUBSTR(CHAR(PARENT_UOW_ID),1,8) AS PARUOWID,
       SUBSTR(CHAR(PARENT_ACTIVITY_ID),1,8) AS PARACTID,
       ACTIVITY_TYPE AS ACTTYPE,
       SUBSTR(CHAR(NESTING_LEVEL),1,7) AS NESTING
FROM TABLE(WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES(1, -2)) AS WLOACTS
ORDER BY MEMB, UOWID, ACTID
```

照会からの出力例は、次のようになります。

COORD	MEMB	UOWID	ACTID	PARUOWID	PARACTID	ACTTYPE	NESTING
0	0	2	3	-	-	CALL	0
0	0	2	5	2	3	READ_DML	1

### システムで現在実行中のアクティビティ

以下の照会では、EXECUTABLE\_ID で WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 出力と MON\_GET\_PKG\_CACHE\_STMT 出力を結合し、システムで現在実行中のすべての SQL アクティビティに関するステートメント・テキストを提供します。

```
SELECT t.application_handle,
       t.uow_id,
       t.activity_id,
       varchar(p.stmt_text, 256) as stmt_text
FROM table(wlm_get_workload_occurrence_ACTIVITIES(NULL, -1)) as t,
     table(mon_get_pkg_cache_stmt(NULL, NULL, NULL, -1)) as p
WHERE t.executable_id = p.executable_id
```

出力例を以下に示します。

APPLICATION_HANDLE	UOW_ID	ACTIVITY_ID	STMT_TEXT
1	1	1	SELECT * FROM SYSCAT.TABLES
47	1	36	INSERT INTO T1 VALUES(123)

### 戻される情報

表 265. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES によって戻される情報

列名	データ・タイプ	説明
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル・モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
COORD_PARTITION_NUM	SMALLINT	coord_partition_num - コーディネーター・パーティション番号モニター・エレメント
LOCAL_START_TIME	TIMESTAMP	local_start_time - ローカル開始時刻モニター・エレメント
UOW_ID	INTEGER	uow_id - 作業単位 ID モニター・エレメント
ACTIVITY_ID	INTEGER	activity_id - アクティビティ ID モニター・エレメント
PARENT_UOW_ID	INTEGER	parent_uow_id - 親作業単位 ID モニター・エレメント
PARENT_ACTIVITY_ID	INTEGER	parent_activity_id - 親アクティビティ ID モニター・エレメント
ACTIVITY_STATE	VARCHAR(32)	activity_state - アクティビティの状態モニター・エレメント

表 265. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES によって戻される情報 (続き)

列名	データ・タイプ	説明
ACTIVITY_STATE (続き)	VARCHAR(32)	<p>アクティビティーの状態。可能な値は、以下のとおりです。</p> <p><b>QUEUED</b></p> <p>アクティビティーが、ワークロード管理キューイングしきい値によってキューに入れられています。パーティション・データベース環境では、この状態は、コーディネーター・エージェントがカタログ・メンバーに対する RPC を行ってしきい値チケットを取得したものの、まだ応答を受け取っていないことを示す場合があります。この状態は、アクティビティーがワークロード管理キューイングしきい値によってキューに入れているか、それほど時間が経過していない場合はアクティビティーがそのチケットを取得する処理中であることを示すことがあります。アクティビティーがキューに入れられているかどうかについての詳細を知るために、どのエージェントがアクティビティーで作業しているかを判別し、カタログ・メンバーにあるオブジェクトの EVENT_OBJECT の値が WLM_QUEUE の値であるかどうかを調べてください。</p> <p><b>TERMINATING</b></p> <p>アクティビティーは実行を完了し、システムから除去されています。</p>
ACTIVITY_TYPE	VARCHAR(32)	<p>アクティビティー・タイプ。可能な値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• CALL</li> <li>• DDL</li> <li>• LOAD</li> <li>• OTHER</li> <li>• READ_DML</li> <li>• WRITE_DML</li> </ul>
NESTING_LEVEL	INTEGER	nesting_level - ネスティング・レベル・モニター・エレメント
INVOCATION_ID	INTEGER	invocation_id - 呼び出し ID モニター・エレメント
ROUTINE_ID	INTEGER	routine_id - ルーチン ID モニター・エレメント



表 265. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES によって戻される情報 (続き)

列名	データ・タイプ	説明
UTILITY_ID	INTEGER	utility_id - ユーティリティ ID モニター・エレメント
SERVICE_CLASS_ID	INTEGER	service_class_id - サービス・クラス ID モニター・エレメント
DATABASE_WORK_ACTION_SET_ID	INTEGER	以下の値のいずれか。 <ul style="list-style-type: none"> <li>このアクティビティがデータベース有効範囲の作業クラスに分類されている場合、値はこの作業クラスがメンバーとなっている作業クラス・セットの ID です。</li> <li>このアクティビティがデータベース有効範囲の作業クラスに分類されていない場合、値は NULL です。</li> </ul>
DATABASE_WORK_CLASS_ID	INTEGER	以下の値のいずれか。 <ul style="list-style-type: none"> <li>このアクティビティがデータベース有効範囲の作業クラスに分類されている場合、値は作業クラスの ID です。</li> <li>このアクティビティがデータベース有効範囲の作業クラスに分類されていない場合、値は NULL です。</li> </ul>
SERVICE_CLASS_WORK_ACTION_SET_ID	INTEGER	以下の値のいずれか。 <ul style="list-style-type: none"> <li>このアクティビティがサービス・クラス有効範囲の作業クラスに分類されている場合、値は作業クラスが属する作業クラス・セットに関連付けられた作業アクション・セットの ID です。</li> <li>このアクティビティがサービス・クラス有効範囲の作業クラスに分類されていない場合、値は NULL です。</li> </ul>
SERVICE_CLASS_WORK_CLASS_ID	INTEGER	以下の値のいずれか。 <ul style="list-style-type: none"> <li>このアクティビティがサービス・クラス有効範囲の作業クラスに分類されている場合、この値はこのアクティビティに割り当てられた作業クラスの ID です。</li> <li>このアクティビティがサービス・クラス有効範囲の作業クラスに分類されていない場合、値は NULL です。</li> </ul>
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	executable_id - 実行可能 ID モニター・エレメント
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_RETURNED	BIGINT	rows_returned - 戻り行数
QUERY_COST_ESTIMATE	BIGINT	query_cost_estimate - 照会コストの見積もり
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り

表 265. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES によって戻される情報 (続き)

列名	データ・タイプ	説明
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
ENTRY_TIME	TIMESTAMP	entry_time - エントリー時間
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
COORD_MEMBER	SMALLINT	coord_member - コーディネーター・メンバー・モニター・エレメント
PACKAGE_SCHEMA	VARCHAR(128)	package_schema - パッケージ・スキーマ。
PACKAGE_NAME	VARCHAR(128)	package_name - パッケージ名。
PACKAGE_VERSION_ID	VARCHAR(64)	package_version_id - パッケージ・バージョン。
SECTION_NUMBER	BIGINT	section_number - セクション番号。
STMTNO	INTEGER	stmtno - ステートメント番号モニター・エレメント

## WLM\_GET\_WORKLOAD\_STATS 表関数 - ワークロード統計を戻す

WLM\_GET\_WORKLOAD\_STATS 関数は、ワークロード名とデータベース・メンバー番号のすべての組み合わせについてワークロード統計の 1 行を返します。

戻すことができる情報の全リストについては、1197 ページの表 266 を参照してください。

### 構文

```
▶▶—WLM_GET_WORKLOAD_STATS—(—workload_name—,—member—)————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *workload\_name*

統計が戻されるワークロードを指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、すべてのワークロードについて統計が戻されます。

#### *member*

現在接続されているデータベースと同じインスタンス内のメンバーの番号を指定する、タイプ INTEGER の入力引数。現行のメンバーには -1、すべてのメンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限

- DBADM 権限
- SQLADM 権限
- WLMADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

以下の照会は、ワークロードの統計を表示します。

```
SELECT SUBSTR(WORKLOAD_NAME,1,18) AS WL_DEF_NAME,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       COORD_ACT_LIFETIME_TOP,
       COORD_ACT_LIFETIME_AVG,
       COORD_ACT_LIFETIME_STDDEV
FROM TABLE(WLM_GET_WORKLOAD_STATS(CAST(NULL AS VARCHAR(128)), -2)) AS WLSTATS
ORDER BY WL_DEF_NAME, MEMB
```

照会からの出力例は、次のようになります。

```
WL_DEF_NAME      MEMB COORD_ACT_LIFETIME_TOP ...
-----
SYSDEFAULTADMWKL 0                -1 ...
SYSDEFAULTUSERW 0                -1 ...
WL1              0                 2 ...

... COORD_ACT_LIFETIME_AVG  COORD_ACT_LIFETIME_STDDEV
... -----
... -1.000000000000000E+000  -1.000000000000000E+000
... -1.000000000000000E+000  -1.000000000000000E+000
... +2.560000000000000E+000   +6.00000000000001E-002
```

## 使用上の注意

この関数は、ワークロード、メンバー、またはサービス・クラスを越えたデータ集約は行いません。ただし、SQL 照会を使用すると、データを集約することができます。

## 戻される情報

表 266. WLM\_GET\_WORKLOAD\_STATS によって戻される情報

列名	データ・タイプ	説明
WORKLOAD_NAME	VARCHAR(128)	workload_name - ワークロード名 : モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号 モニター・エレメント
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ・モニター・エレメント
CONCURRENT_WLO_TOP	INTEGER	concurrent_wlo_top - 並行ワークロード・オカレンスの最上位モニター・エレメント
CONCURRENT_WLO_ACT_TOP	INTEGER	concurrent_wlo_act_top - 並行 WLO アクティビティの最上位モニター・エレメント
COORD_ACT_COMPLETED_TOTAL	BIGINT	coord_act_completed_total - コーディネーター・アクティビティ完了総数モニター・エレメント

表 266. WLM\_GET\_WORKLOAD\_STATS によって戻される情報 (続き)

列名	データ・タイプ	説明
COORD_ACT_ABORTED_TOTAL	BIGINT	coord_act_aborted_total - コーディネーター・アクティビティ打ち切り総数モニター・エレメント
COORD_ACT_REJECTED_TOTAL	BIGINT	coord_act_rejected_total - コーディネーター・アクティビティ・リジェクト総数モニター・エレメント
WLO_COMPLETED_TOTAL	BIGINT	wlo_completed_total - 完了したワークロード・オカレンスの合計モニター・エレメント
COORD_ACT_LIFETIME_TOP	BIGINT	coord_act_lifetime_top - コーディネーター・アクティビティ存続時間の最上位モニター・エレメント
COORD_ACT_LIFETIME_AVG	DOUBLE	coord_act_lifetime_avg - コーディネーター・アクティビティ平均存続期間モニター・エレメント
COORD_ACT_LIFETIME_STDDEV	DOUBLE	このワークロードに関連付けられたネスト・レベル 0 の完了または打ち切られたコーディネーター・アクティビティの存続期間の標準偏差。単位はミリ秒です。ワークロードの COLLECT AGGREGATE ACTIVITY DATA パラメーターが NONE に設定される場合、列の値は NULL です。この標準偏差はコーディネーター・アクティビティの存続期間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。
COORD_ACT_EXEC_TIME_AVG	DOUBLE	coord_act_exec_time_avg - コーディネーター・アクティビティ平均実行時間モニター・エレメント
COORD_ACT_EXEC_TIME_STDDEV	DOUBLE	このワークロードに関連付けられたネスト・レベル 0 の完了または打ち切られたコーディネーター・アクティビティの実行時間の標準偏差。単位はミリ秒です。この標準偏差はコーディネーター・アクティビティの実行時間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。ワークロードの COLLECT AGGREGATE ACTIVITY DATA パラメーターが NONE に設定される場合、列の値は NULL です。
COORD_ACT_QUEUE_TIME_AVG	DOUBLE	coord_act_queue_time_avg - コーディネーター・アクティビティ平均キュー時間モニター・エレメント

表 266. WLM\_GET\_WORKLOAD\_STATS によって戻される情報 (続き)

列名	データ・タイプ	説明
COORD_ACT_QUEUE_TIME_STDDEV	DOUBLE	このワークロードに関連付けられたネスト・レベル 0 の完了または打ち切られたコーディネーター・アクティビティのキュー時間の標準偏差。単位はミリ秒です。ワークロードの COLLECT AGGREGATE ACTIVITY DATA パラメーターが NONE に設定される場合、列の値は NULL です。この標準偏差はコーディネーター・アクティビティのキュー時間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻れます。
UOW_TOTAL_TIME_TOP	BIGINT	uow_total_time_top - 作業単位合計時間トップ・モニター・エレメント
UOW_THROUGHPUT	DOUBLE	uow_throughput - 作業単位スループット : モニター・エレメント
UOW_LIFETIME_AVG	DOUBLE	uow_lifetime_avg - 作業単位の平均存続期間 : モニター・エレメント
UOW_COMPLETED_TOTAL	BIGINT	uow_completed_total - 完了済みの合計作業単位 : モニター・エレメント
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間 : モニター・エレメント
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - ディスパッチャーの合計実行キュー時間 : モニター・エレメント
ACT_THROUGHPUT	DOUBLE	act_throughput - アクティビティ・スループット : モニター・エレメント
CPU_UTILIZATION	DOUBLE	cpu_utilization - CPU 使用率 : モニター・エレメント
APP_ACT_COMPLETED_TOTAL	BIGINT	app_act_completed_total - 成功した外部コーディネーター・アクティビティの総数モニター・エレメント
APP_ACT_ABORTED_TOTAL	BIGINT	app_act_aborted_total - 失敗した外部コーディネーター・アクティビティの総数モニター・エレメント
APP_ACT_REJECTED_TOTAL	BIGINT	app_act_rejected_total - リジェクトされた外部コーディネーター・アクティビティの総数モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## WLM\_SET\_CLIENT\_INFO プロシージャ - クライアント情報設定

WLM\_SET\_CLIENT\_INFO プロシージャは、DB2 サーバーでの現行接続に関連付けられたクライアント情報を設定します。

このプロシージャを使用することにより、クライアントのユーザー ID、アプリケーション名、ワークステーション名、アカウント情報、またはワークロード

情報を DB2 サーバーで設定できます。このプロシージャーを呼び出すと、この接続に関する、関連するトランザクション・プロセッサ (TP) モニターのクライアント情報フィールドおよび特殊レジスター設定の保管された値が変更されます。

クライアント情報フィールドは、現在接続を使用しているアプリケーションまたはユーザーの ID を判別するために、DB2 サーバーで使用されます。接続用のクライアント情報フィールドは、DB2 のワークロード評価中に考慮され、この接続用に生成される DB2 監査レコードまたはアプリケーション・スナップショットにも表示されます。

sqleseti API とは異なり、このプロシージャーは、クライアントでクライアント情報を設定するのではなく、対応するクライアント属性を DB2 サーバーで設定します。そのため、このプロシージャーを使用して DB2 サーバーに設定されたクライアント情報を照会するために sqleqry API を使用することはできません。アプリケーションが sqleseti API を使用してクライアント情報を変更する場合、新しい値によって DB2 サーバーでの設定値が変更されます。sqleseti API を使用してアカウント情報を変更することなくユーザー ID またはアプリケーション名のいずれかを変更すると、DB2 サーバーのアカウント情報もクライアントのアカウント情報の値にリセットされます。

このプロシージャーで提供されたデータ値は、関連した TP モニターのフィールドまたは特殊レジスターに保管される前に、適切なデータベースのコード・ページに変換されます。データベースのコード・ページへの変換後に最大サポート・サイズを超えるデータ値は、サーバーに保管される前に切り捨てられます。切り捨てられた値は、それら保管された値が照会されるときに TP モニターのフィールドおよび特殊レジスターの両方によって返されます。

WLM\_SET\_CLIENT\_INFO プロシージャーはトランザクションの制御下にはなく、このプロシージャーにより加えられたクライアント情報変更は、作業単位のコミットまたはロールバックとは無関係です。ただし、各アプリケーションの次の作業単位の開始時にワークロードの再評価が行われるので、クライアント情報変更を有効にするには、COMMIT または ROLLBACK のいずれかのステートメントを発行する必要があります。

## 構文

```
►►—WLM_SET_CLIENT_INFO—(—client_userid—,—client_wrkstnname—,——————►  
►—client_applname—,—client_acctstr—,—client_workload—)——————►◄
```

スキーマは SYSPROC です。

## プロシージャー・パラメーター

### *client\_userid*

クライアントのユーザー ID を指定する、タイプ VARCHAR(255) の入力引数。NULL を指定する場合、値は変わりません。空ストリング (デフォルト値) を指定する場合、クライアントのユーザー ID はデフォルト値であるブランクにリセットされます。

#### *client\_wrkstnname*

クライアントのワークステーション名を指定する、タイプ VARCHAR(255) の入力引数。 NULL を指定する場合、値は変わりません。空ストリング (デフォルト値) を指定する場合、クライアントのワークステーション名はデフォルト値であるブランクにリセットされます。

#### *client\_applname*

クライアントのアプリケーション名を指定する、タイプ VARCHAR(255) の入力引数。 NULL を指定する場合、値は変わりません。空ストリング (デフォルト値) を指定する場合、クライアントのアプリケーション名はデフォルト値であるブランクにリセットされます。

#### *client\_acctstr*

クライアント・アカウント・ストリングを指定する、タイプ VARCHAR(200) の入力引数。 NULL を指定する場合、値は変わりません。空ストリング (デフォルト値) を指定する場合、クライアント・アカウント・ストリングはデフォルト値であるブランクにリセットされます。

#### *client\_workload*

クライアントのワークロードの割り当てを指定する、タイプ VARCHAR(255) の入力引数。 NULL を指定する場合、値は変わりません。値は、以下のとおりです。

#### **SYSDEFAULTADMWORKLOAD**

データベース接続が SYSDEFAULTADMWORKLOAD に割り当てられることを指定します。これにより、ACCESSCTRL、DATAACCESS、DBADM、SECADM、または WLMADM 権限を持つユーザーは通常のワークロード評価を迂回することができます。

#### **AUTOMATIC**

サーバーが自動的に実行するワークロード計算に選ばれているワークロードに、データベース接続を割り当てるよう指定します。

注: *client\_workload* 引数は大文字小文字を区別します。

### **許可**

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

### **デフォルトの PUBLIC 特権**

なし

## 例

以下のプロシージャ呼び出しは、クライアントのユーザー ID、ワークステーション名、アプリケーション名、アカウントिंग・ストリング、およびワークロード割り当てモードを設定します。

```
CALL SYSPROC.WLM_SET_CLIENT_INFO('db2user', 'machine.torolab.ibm.com',  
    'auditor', 'Accounting department', 'AUTOMATIC')
```

以下のプロシージャ呼び出しは、クライアントのユーザー ID を db2user2 に設定し、他のクライアント属性は設定しません。

```
CALL SYSPROC.WLM_SET_CLIENT_INFO('db2user2', NULL, NULL, NULL, NULL)
```

以下のプロシージャ呼び出しは、クライアントのユーザー ID をブランクにリセットし、他のクライアント属性の値を変更しません。

```
CALL SYSPROC.WLM_SET_CLIENT_INFO('', NULL, NULL, NULL, NULL)
```

## 使用上の注意

プロシージャ・パラメーターに指定した入力がある指定フィールド長を超える場合、入力フィールドは切り捨てられ、プロシージャは切り捨てられた入力を使用して実行されます。

単一引用符を含む入力フィールドはサポートされておらず、エラーになります。

## WLM\_SET\_CONN\_ENV - アクティビティ・データの収集とセクション実行時統計の測定の有効化

WLM\_SET\_CONN\_ENV プロシージャは、特定の接続でのアクティビティ・データの収集とセクション実行時統計 (セクションの実行中に測定されるランタイム統計) の測定を有効にします。

WLM\_SET\_CONN\_ENV プロシージャによる設定が適用されると、別の WLM\_SET\_CONN\_ENV プロシージャの呼び出しによって明示的に上書きされるか、接続が閉じられるまで適用され続けます。接続が閉じられた後に同じアプリケーション・ハンドルを再利用する新規接続がある場合、その接続でそのアプリケーション・ハンドルを使用するために、前の接続の設定が継承されることはありません。

▶—WLM\_SET\_CONN\_ENV—(—application\_handle—, —settings—)————▶

スキーマは SYSPROC です。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限



## デフォルトの PUBLIC 特権

なし

## パラメーター

### *application\_handle*

接続環境が変更されるアプリケーション・ハンドルを指定する、タイプ BIGINT の入力引数。指定されるアプリケーション・ハンドルは、既存のアプリケーションを参照する必要があります (参照しない場合、SQLSTATE 5U002 が戻されます)。NULL 値を使用することで、変更される環境を持つ接続が、プロシージャが呼び出された接続であることを示すことができます。

### *settings*

1 つ以上のモニター設定の指定を可能にする、タイプ CLOB(8K) の入力引数。設定は、次の形式の名前と値の対として指定されます。

```
<setting name tag>value</setting name tag>
```

設定は、それぞれ最大で 1 回指定できます。設定名には、大/小文字の区別があります。設定の変更は、次回ステートメントを実行するときに有効になります。既に進行中のステートメントに対しては有効になりません。

使用可能な設定名タグは次のとおりです。

- '`<collectactdata>value</collectactdata>`'

アクティビティー・イベント・モニターによって収集するアクティビティー・データを指定します。有効な値は以下のとおりです (単語の間のスペースのバリエーションはサポートされています)。

値	説明
NONE	アクティビティー・データを収集しません。
WITHOUT DETAILS	各アクティビティーに関するデータを、アクティビティーの実行が完了したときに、任意のアクティブなアクティビティー・イベント・モニターに送信します。ステートメント、コンパイル環境、およびセクション環境のデータに関する詳細は送信されません。
WITH DETAILS	任意のアクティブなアクティビティーにステートメントおよびコンパイル環境のデータが含まれる場合、それを該当するアクティビティーのイベント・モニターへ送信します。セクションの環境データは送信されません。

値	説明
WITH DETAILS, SECTION	<p>ステートメント、コンパイル環境、セクション環境データ、セクション実行時統計を、それらが含まれるアクティビティー用のアクティブなアクティビティー・イベント・モニターに送信します。</p> <p>収集されるセクション実行時統計で、<i>collectsectionactuals</i> を BASE に設定するか、<b>section_actuals</b> データベース構成パラメーターを BASE に設定する必要があります。セクション実行時統計は、アクティビティー・データが収集されるすべてのパーティションで収集されます。</p>
WITH DETAILS, SECTION AND VALUES	<p>ステートメント、コンパイル環境、セクション環境データ、セクション実行時統計、および入力データ値を、それらが含まれるアクティビティー用のアクティブなアクティビティー・イベント・モニターに送信します。</p> <p>収集されるセクション実行時統計で、<i>collectsectionactuals</i> を BASE に設定するか、<b>section_actuals</b> データベース構成パラメーターを BASE に設定する必要があります。セクション実行時統計は、アクティビティー・データが収集されるすべてのパーティションで収集されます。</p>
WITH DETAILS AND VALUES	<p>ステートメント、コンパイル環境、および入力データ値を、それらが含まれるアクティビティー用のアクティブなアクティビティー・イベント・モニターに送信します。セクションの環境データは送信されません。</p>

- '`<collectactpartition>COORDINATOR</collectactpartition>`' または '`<collectactpartition>ALL</collectactpartition>`'

アクティビティー・データが収集される場所 (コーディネーター・パーティションのみか、またはすべてのパーティションか) を指定します。

*collectactpartition* を指定しない場合、接続は *collectactpartition* のこれまでの値 (デフォルトは COORDINATOR) を維持します。

- '`<collectsectionactuals>NONE</collectsectionactuals>`' または '`<collectsectionactuals>BASE</collectsectionactuals>`'

*collectsectionactuals* が BASE に設定される場合、セクション実行時統計は収集されます。

## 例

以下の例はどちらも、コーディネーター・パーティションで、現行接続の、詳細を伴わないアクティビティーの収集を有効にします。

```
CALL WLM_SET_CONN_ENV(NULL, '<collectactdata>WITHOUT DETAILS</collectactdata>')
```

```
CALL WLM_SET_CONN_ENV(NULL, '<collectactdata>WITHOUT
DETAILS</collectactdata><collectactpartition>COORDINATOR
</collectactpartition>')
```

次の例は、すべてのパーティションで、現行接続の、セクションの環境データとセクション実行時統計は伴うがデータ値は伴わないアクティビティー・データの収集を有効にします。

```
CALL WLM_SET_CONN_ENV(NULL, '<collectactdata>WITH DETAILS, SECTION
</collectactdata><collectactpartition>ALL</collectactpartition>')
```

以下の例は、現行接続のアクティビティー・データの収集を無効にします。

```
CALL WLM_SET_CONN_ENV(NULL, '<collectactdata>NONE</collectactdata>')
```

## 使用上の注意

*collectactdata* 設定は、アクティビティー・データの収集を接続レベルでのみ制御します。アクティビティーには、複数のアクティビティー・データの収集制御が適用されていることがあります。例えば、接続は COLLECT ACTIVITY DATA 節が適用されたサービス・クラスにマップされる可能性があります。複数のアクティビティー・データの収集制御が適用されている場合、すべての設定を組み合わせたものが実際の設定になります。以下に例を示します。

1. 接続レベルの制御は、詳細を伴わない (WITHOUT DETAILS) アクティビティー・データである。
2. ワークロードの制御は、なし (NONE) である。
3. サービス・クラスの制御は、詳細および値を伴う (WITH DETAILS AND VALUES) アクティビティー・データである。
4. アクティビティーが実行を完了したときに、アクティビティーおよびデータ値に関する詳細な情報がすべてのアクティブなイベント・モニターに送信される。

設定が WLM\_SET\_CONN\_ENV プロシージャの入力で指定されない場合、接続環境で変更されません。

セクション実行時統計の収集の有効な設定は、*collectsectionactuals* 設定と **section\_actuals** データベース構成パラメーターを組み合わせたものです。例えば、*collectsectionactuals* が BASE に設定され、**section\_actuals** データベース構成パラメーターの値が NONE に設定された場合、セクション実行時統計の収集の有効な設定は、BASE になります (逆も当てはまります)。*collectsectionactuals* が BASE に設定される場合には、自動統計プロファイル (**auto\_stats\_prof** データベース構成パラメーターを使って有効にされる) を使用しないでください (使用すると、警告 SQLSTATE 01HN2 が返されます)。

アクティビティー・データおよびセクション実行時統計が収集されているとき (*collectactdata* が NONE 以外の値に設定されているとき) に、接続に対して自動クライアント・リポートを実行することはできません。

## その他のルーチンおよびビュー

### ALTOBJ

ALTOBJ プロシージャは、変更される既存の表のターゲット・データ定義言語 (DDL) として機能する、入力 CREATE TABLE ステートメントを解析します。

このプロシージャは、変更される表のデータのバックアップを取ってから、元の表をドロップし、DDL ステートメントを使用して新しいバージョンを作成します。最終ステップでは、保管データを新しい表にロードして戻します。

このプロシージャでは、次の表変更操作がサポートされていて、リカバリー可能な従属関係が維持されています。

- 列の名前変更
- 列のサイズの増減
- 列タイプの変更と DB2 スカラー関数を使用した既存データのトランスフォーム
- 10 進数値の精度または位取りの変更
- 列のデフォルト値の変更
- 列の NULL 可能属性の NULL 可能への変更
- 列のドロップ

## 構文

```
▶▶—ALTOBJ—(—exec-mode—,—sql-stmt—,—alter-id—,—msg—)————▶▶
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *exec-mode*

次の実行モードのいずれかを指定する、タイプ VARCHAR(30) の入力引数。

#### 'GENERATE'

VALIDATE、APPLY、および UNDO モードに必要なすべてのスクリプトを生成することを指定します。

#### 'VALIDATE'

ステートメント構文が検査されることを指定します。このオプションにより、変更される表の関連オブジェクトおよびリレーションシップの処理を管理するスクリプトも生成されます。

#### 'APPLY\_CONTINUE\_ON\_ERROR' または 'APPLY\_STOP\_ON\_ERROR'

変更される表の関連オブジェクトおよびリレーションシップの処理を管理するスクリプトを生成することを指定します。元の表のデータは、新しい表を取り込むために、エクスポートされ、トランスフォームされて使用されません。

#### 'UNDO'

生じた可能性のあるエラーをロールバック操作でリカバリーできない場合に、表変更操作によって加えられた変更を取り消すことを指定します。このモードは、元の表および生成されたスクリプトが削除されていない場合のみ有効です。

#### 'FINISH'

名前変更された元の表をドロップすることを指定します。

### *sql-stmt*

既存の表を変更するときのテンプレートとして使用される CREATE TABLE ステートメントを指定する、タイプ VARCHAR(2048) の入力引数。 *exec-mode* が

'GENERATE' である場合、*sql-stmt* は NULL 値であってはなりません。それ以外の場合、*sql-stmt* を NULL 値にすることはできますが、*alter-id* が -1 ではない場合だけです。

#### *alter-id*

この呼び出しによって生成されるすべてのステートメントを識別する、タイプ INTEGER の入力および出力引数。 -1 が指定される場合、新しい ID が生成されて呼び出し元に戻されます。指定した整数で識別される既存のステートメントがあれば、それらは上書きされます。

#### *msg*

指定した実行モード下で、表変更プロセス用に生成されるか表変更プロセスで使用されるすべての SQL ステートメントを表示するために実行できる SQL 照会を含む、タイプ VARCHAR(2048) の出力引数。

## 許可

この関数を実行するには、DBADM 権限が必要です。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: ALTOBJ プロシージャを実行して、表 T1 の列 CL2 をタイプ INTEGER からタイプ BIGINT に変更します。表 T1 の元のデータ定義言語は、次のとおりです。

```
CREATE TABLE T1 (CL1 VARCHAR(5), CL2 INTEGER)
```

列データを変更するための ALTOBJ プロシージャ呼び出しは、次のとおりです。

```
CALL SYSPROC.ALTOBJ('APPLY_CONTINUE_ON_ERROR',  
  'CREATE TABLE T1 (CL1 VARCHAR(5), CL2 BIGINT)', -1, ?)
```

注: 次のエラーが表示された場合は、**applheapsz** パラメーターの値を増やしてください。

SQL0443N ルーチン "SYSPROC.ALTOBJ" (特定名 "ALTOBJ") が、診断テキスト "SQL0954" とともにエラー SQLSTATE を返しました。SQLSTATE=38553

例 2: ALTOBJ プロシージャを、*alter-id* 入力と共に VALIDATE モードで実行します。

```
CALL SYSPROC.ALTOBJ('VALIDATE', CAST (NULL AS VARCHAR(2048)), 123, ?)
```

## 使用上の注意

このプロシージャは表をドロップして再作成しますが、元の表を作成したユーザーは表の定義者として残ります。しかし、監査では、このプロシージャを実行しているユーザーが表をドロップして再作成したと示されます。

このプロシージャは、以下の表変更操作はサポートしません。

- マテリアライズ照会表 (MQT) の変更はサポートされていません。MQT の入っている表の変更はサポートされます。
- 型付き表の変更はサポートされていません。
- ニックネームを使用したりリモート表の変更は、サポートされていません。
- 列のシーケンスを再配列することはできません。
- 列の追加と除去、または名前変更と除去を、プロシーチャーの 1 回の呼び出しで行うことはサポートされていませんが、列の追加と名前変更はサポートされています。その理由は、表の変更方法を指示する唯一の方法が、列一致情報ではなく、ターゲット DDL を使用する方法であるからです。既存の表から変更後の表ヘデータを変換するときには、ALTOBJ プロシーチャーは次のような規則に従います。
  1. 既存の表中の列数が、変更後の表と同じである場合、列の追加または除去は行われないとみなされます。この場合、列の名前変更のみが可能であり、列索引によって突き合わせられます。
  2. 既存の表中の列数が、変更後の表より少ない場合、列の追加が行われるとみなされます。列の名前変更が可能であり、新規の列が末尾に追加されます。既存の列は、索引によって突き合わせられます。
  3. 既存の表中の列数が、変更後の表より多い場合、列の除去が行われるとみなされます。列は、名前変更不能になり、名前で突き合わせられます。ドロップする対象の列は、表中の任意の既存の列でかまいません。
- 構造化タイプ UDT および参照タイプの UDT は、サポートされていません。
- 変更される基本表上に定義された MQT が、表の変更プロセス中にデータを追加されることはありません。

ALTOBJ プロシーチャーを使用して表を変更した場合に、その表に MQT が定義されていると、MQT は作成されますが、そこにはデータは取り込まれません。

ALTOBJ プロシーチャーを使用して表を変更した場合に、その表に MQT が定義されていると、変更された表での選択結果に属さないすべての列は失われてしまいます。なぜなら、MQT の内容は、新規の基本表から再作成されるからです。

オブジェクトの定義は、ALTOBJ プロシーチャーを次に呼び出したときには変わっていることがあります。セッションからセッションへ持続するオブジェクト・ロックはないからです。

表に関連付けられた表プロファイル (RUNSTATS プロファイルなど) は、このような広範囲にわたるプロセスを実行する過程で失われてしまいます。

SYSTOOLSPACE をルーチンの操作表に使用してメタデータを保管します。つまり、データベース・オブジェクトとその操作の記述に使用されるデータです。

表の行または列のアクセス制御がアクティブになっている場合は、その表に対して ALTOBJ を実行するとエラー (DBA7903) になります。

## COMPILATION\_ENV 表関数 - コンパイル環境の要素の検索

COMPILATION\_ENV 表関数は、コンパイル環境の要素を戻します。

## 構文

▶▶—COMPILATION\_ENV—(—*compilation-env*—)————▶▶

スキーマは SYSPROC です。

## 表関数パラメーター

*compilation-env*

**comp\_env\_desc** (コンパイル環境) モニター・エレメントから得られる、コンパイル環境が入った、タイプ BLOB(2M) の入力引数。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

関数は、NAME VARCHAR(256) および VALUE VARCHAR(1024) という 2 つの列 (1210 ページの表 267を参照) がある表を戻します。コンパイル環境のエレメント名として可能な値を、1210 ページの表 268で説明します。

エレメント値の提供元は主に、SQL ステートメントが動的に発行されるか、またはパッケージの一部としてバインドされるかに応じて異なります。

コンパイル環境の項目の数とタイプは、DB2 データベース・マネージャーの機能の追加に応じ、時の経過と共に変わる可能性があります。コンパイル環境が、この関数が実行されているのとは異なる DB2 データベース・マネージャー・レベルのものである場合、そのレベルの関数によって認識されるエレメントだけが戻されます。エレメントの記述は、リリースごとに異なる場合もあります。

## 例

**例 1:** デッドロック・イベント・モニターによって以前にキャプチャーされた特定のコンパイル環境のすべてのエレメントを要求します。WITH DETAILS HISTORY オプションを指定して作成されたデッドロック・イベント・モニターは、動的 SQL ステートメントのコンパイル環境をキャプチャーします。このキャプチャー環境は、表関数への入力として受け入れられるものです。

```
SELECT NAME, VALUE
FROM TABLE(SYSPROC.COMPILATION_ENV(:hv1)) AS t
```

**例 2:** コンパイル環境の特定のエレメント (デフォルトのスキーマ) を要求します。

```

SELECT NAME, VALUE
FROM TABLE(SYSPROC.COMPILOATION_ENV(:hv1)) AS t
WHERE NAME = 'SCHEMA'

```

例 3: パッケージ・キャッシュ内の特定のステートメントのコンパイル環境を表示します。

1. 実行可能 ID を取得します。これは、以下のステートメントを使って、対象のステートメントを識別するために使用されます。

```

SELECT EXECUTABLE_ID, VARCHAR{STMT_TEXT, 100}
FROM TABLE(MON_GET_PKG_CACHE_STMT(NULL,NULL,NULL,-1)) AS t

```

以下は、上記ステートメントを実行した後の出力例です。

```

EXECUTABLE_ID                    2
-----
x'01000000000000000000000000000000000000000000000020020090914151405241700' select count(*) from syscat.tables
...

```

2. ステートメント (実行可能 ID を使って識別される) のコンパイル環境を調べ、COMPILOATION\_ENV 表関数を使ってコンパイル環境をフォーマットします。以下のステートメントは、これを行う方法を示す例です。

```

SELECT VARCHAR(NAME, 30), VARCHAR(VALUE, 50)
FROM TABLE(COMPILOATION_ENV((SELECT COMP_ENV_DESC FROM TABLE
(MON_GET_PKG_CACHE_STMT(NULL,
x'01000000000000000000000000000000000000000000000020020090914151405241700',
NULL, -1)) AS t))) AS s

```

以下は、上記ステートメントを実行した後の出力例です。

```

1                                2
-----
ISOLATION                        CS
QUERY_OPTIMIZATION              5
MIN_DEC_DIV_3                   NO
DEGREE                           1
SQLRULES                         DB2
REFRESH_AGE                     +0000000000000000.000000
RESOLUTION_TIMESTAMP            2009-09-14-15.14.05.000000
FEDERATED_ASYNCHRONY           0
PATH                            "SYSIBM","SYSFUN","SYSPROC","SYSIBMADM","SWALKTY"
MAINTAINED_TABLE_TYPE          SYSTEM

```

10 record(s) selected.

## 戻される情報

表 267. COMPILOATION 表関数によって戻される情報

列名	データ・タイプ	説明
NAME	VARCHAR(256)	コンパイル環境のエレメント。詳しくは、表 268を参照してください。
VALUE	VARCHAR(1024)	エレメントの値。

表 268. COMPILOATION\_ENV 表関数によって戻されるコンパイル環境のエレメント

エレメント名	説明
ISOLATION	SQL コンパイラに渡される分離レベル。この値は、現行のパッケージの CURRENT ISOLATION 特殊レジスタまたは ISOLATION BIND オプションから取得されます。



表 268. COMPILATION\_ENV 表関数によって戻されるコンパイル環境の要素 (続き)

要素名	説明
QUERY_OPTIMIZATION	SQL コンパイラに渡される照会最適化レベル。この値は、現在のパッケージの CURRENT QUERY OPTIMIZATION 特殊レジスタまたは QUERYOPT BIND オプションから取得されます。
MIN_DEC_DIV_3	SQL コンパイラに渡される、要求された 10 進計算スケール。この値は、min_dec_div_3 データベース構成パラメータから取得されます。
DEGREE	SQL コンパイラに渡される、要求された内部並列処理の度合い。この値は、現在のパッケージの CURRENT DEGREE 特殊レジスタまたは DEGREE BIND オプションから取得されます。
SQLRULES	SQL コンパイラに渡される、要求された SQL ステートメントの動作。この値は、現在のパッケージの LANGLVL BIND オプションの設定から導出されます。可能な値は、'DB2' または 'SQL92' です。
REFRESH_AGE	SQL コンパイラに渡される、許容データ待ち時間。この値は、現在のパッケージの CURRENT REFRESH AGE 特殊レジスタまたは REFRESHAGE BIND オプションから取得されます。
SCHEMA	SQL コンパイラに渡されるデフォルトのスキーマ。この値は、現在のパッケージの CURRENT SCHEMA 特殊レジスタまたは QUALIFIER BIND オプションから取得されます。
PATH	SQL コンパイラに渡される関数パス。この値は、現在のパッケージの CURRENT PATH 特殊レジスタまたは FUNC_PATH BIND オプションから取得されます。
TRANSFORM_GROUP	SQL コンパイラに渡されるトランスフォーム・グループ情報。この値は、CURRENT DEFAULT TRANSFORM GROUP 特殊レジスタまたは TRANSFORMGROUP パッケージ BIND オプションから取得されます。
MAINTAINED_TABLE_TYPE	SQL コンパイラに渡される、どの表タイプを最適化の検討対象にできるかを示すインディケータ。この値は、CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION 特殊レジスタから取得されます。
RESOLUTION_TIMESTAMP	SQL ステートメント内の関数やデータ・タイプ参照などの項目の解決のために、SQL コンパイラによって使用されるタイム・スタンプ。このタイム・スタンプは、現在のタイム・スタンプか、または現在のパッケージの最新の明示的なバインド操作のタイム・スタンプです。
FEDERATED_ASYNCRONY	SQL コンパイラに渡される、要求されたフェデレーテッド非同期並列処理の度合い。この値は、現在のパッケージの CURRENT FEDERATED ASYNCHRONY 特殊レジスタまたは FEDERATED_ASYNCRONY BIND オプションから取得されます。

## CONTACTGROUPS 管理ビュー - 連絡先グループのリストの検索

CONTACTGROUPS 管理ビューは連絡先グループのリストを戻します。これはシステム上にローカルに定義されている場合も、グローバルなリストに定義されている場合もあります。

Database Administration Server (DAS) の `contact_host` 構成パラメーターの設定により、リストがローカルかグローバルかが決まります。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- CONTACTGROUPS 管理ビューに対する SELECT 特権
- CONTACTGROUPS 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

すべての連絡先グループのリストを検索します。

```
SELECT * FROM SYSIBMADM.CONTACTGROUPS
```

以下はこの照会の出力例です。

NAME	DESCRIPTION	MEMBERNAME	MEMBERTYPE
group1	DBA Group1 Contact List	name1	CONTACT
group1	DBA Group1 Contact List	name9	CONTACT
group2	DBA Group2 List	name2	CONTACT
group3		group2	GROUP
group5	DBA Group5	group2	GROUP
group6	DBA Group6	group3	GROUP
group7		name1	CONTACT

7 record(s) selected.

### 使用上の注意

DAS が作成されていて実行中でなければなりません。

### 戻される情報

表 269. CONTACTGROUPS 管理ビューによって戻される情報

列名	データ・タイプ	説明
NAME	VARCHAR(128)	連絡先グループの名前。
DESCRIPTION	VARCHAR(128)	連絡先グループの説明。

表 269. CONTACTGROUPS 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明
MEMBERNAME	VARCHAR(128)	連絡先グループのメンバーの名前。この名前は 1 つの連絡先を指す場合もあれば、別の連絡先グループを指す場合もあります。
MEMBERTYPE	VARCHAR(7)	連絡先グループのメンバーのタイプ。タイプは CONTACT または GROUP のいずれかです。

## CONTACTS 管理ビュー - 連絡先のリストの検索

CONTACTS 管理ビューは、データベース・サーバーで定義されている連絡先のリストを戻します。

Database Administration Server (DAS) の **contact\_host** 構成パラメーターの設定により、リストがローカルかグローバルかが決まります。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- CONTACTS 管理ビューに対する SELECT 特権
- CONTACTS 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

すべての連絡先を検索します。

```
SELECT * FROM SYSIBMADM.CONTACTS
```

以下はこの照会の出力例です。

```

NAME      TYPE  ADDRESS                MAX_PAGE_LENGTH  DESCRIPTION
-----
user1     EMAIL user3@ca.ibm.com      - DBA Extraordinaire
user2     EMAIL user2@ca.ibm.com - DBA on Email
user3     PAGE  user3@ca.ibm.com      128 DBA on Page
user5     EMAIL user2@ca.ibm.com - DBA Extraordinaire

```

4 record(s) selected.

## 使用上の注意

DAS が作成されていて実行中でなければなりません。

## 戻される情報

表 270. CONTACTS 管理ビューによって戻される情報

列名	データ・タイプ	説明
NAME	VARCHAR(128)	連絡先の名前。
TYPE	VARCHAR(5)	連絡先のタイプ • 'EMAIL' • 'PAGE'
ADDRESS	VARCHAR(128)	address - 接続の開始元となった IP アドレス
MAX_PAGE_LENGTH	INTEGER	メッセージの最大長。例えば、ポケットベル呼び出しサービスにメッセージ長の制限がある場合に使用されます。
DESCRIPTION	VARCHAR(128)	連絡先の説明。

## DB\_HISTORY 管理ビュー - 履歴ファイル情報の検索

DB\_HISTORY 管理ビューは、すべてのデータベース・パーティションの履歴ファイルからの情報を戻します。

スキーマは SYSIBMADM です。

## 許可

以下のいずれかの権限が必要です。

- DB\_HISTORY 管理ビューに対する SELECT 特権
- DB\_HISTORY 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 使用上の注意

データ・パーティション表が再編成される時、再編成対象のそれぞれのデータ・パーティションに対して 1 つのレコードが戻されます。データ・パーティション表の特定の 1 つのデータ・パーティションだけが再編成される場合、そのパーティションのレコードだけが戻されます。

## 例

データベース・パーティション番号、項目 ID、操作、開始時刻、および状況情報を、クライアントが現在接続しているデータベース・パーティションすべてのデータベース履歴ファイルから選択します。

```
SELECT DBPARTITIONNUM, EID, OPERATION, START_TIME, ENTRY_STATUS
FROM SYSIBMADM.DB_HISTORY
```

以下はこの照会の出力例です。

```
DBPARTITIONNUM EID                OPERATION START_TIME      ENTRY_STATUS
-----
                0                1 A          20051109185510 A
```

1 record(s) selected.

## 戻される情報

表 271. DB\_HISTORY 管理ビューによって戻される情報

列名	データ・タイプ	説明
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
EID	BIGINT	履歴ファイル内の項目を一意的に識別する番号。
START_TIME	VARCHAR(14)	start_time - イベント開始時刻モニター・エレメント
SEQNUM	SMALLINT	シーケンス番号。
END_TIME	VARCHAR(14)	ログに記録されたイベントの終了を示すタイム・スタンプ。
FIRSTLOG	VARCHAR(254)	イベントに関連した最も初期のトランザクション・ログの名前。
LASTLOG	VARCHAR(254)	イベントに関連した最新のトランザクション・ログの名前。
BACKUP_ID	VARCHAR(24)	バックアップ ID または固有の表 ID。
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント
TABNAME	VARCHAR(128)	table_name - 表名モニター・エレメント
COMMENT	VARCHAR(254)	ログに記録されたイベントに関連する、システム生成のコメント・テキスト。
CMD_TEXT	CLOB(2 M)	ログに記録されたイベントに関連するデータ定義言語。
NUM_TBSPS	INTEGER	num_tbsps - 表スペースの数モニター・エレメント

表 271. DB\_HISTORY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明
TBSPNAMES	CLOB(5 M)	ログに記録されたイベントに関連する表スペースの名前。
OPERATION	CHAR(1)	操作 ID。考えられる値については、1218 ページの表 272 を参照してください。
OPERATIONTYPE	CHAR(1)	操作のアクション ID。考えられる値については、1218 ページの表 272 を参照してください。
OBJECTTYPE	CHAR(1)	操作のターゲット・オブジェクトの ID。可能な値は、D (全データベース)、I (索引)、P (表スペース)、R (パーティション表)、および T (表) です。
LOCATION	VARCHAR(255)	ログに記録されたイベントに関連する、バックアップ・イメージやロード入カファイルなどのファイルの絶対パス名。
DEVICETYPE	CHAR(1)	ログに記録されたイベントに関連するデバイス・タイプ ID。このフィールドにより、LOCATION フィールドを解釈する方法が決まります。可能な値は、A (TSM)、C (クライアント)、D (ディスク)、F (スナップショット・バックアップ)、K (ディスケット)、L (ローカル)、N (DB2 により内部生成される)、O (その他)(他のペンダー・デバイスのサポート用)、P (パイプ)、Q (カーソル)、R (リモート・フェッチ・データ)、S (サーバー)、T (テープ)、U (ユーザー出口)、および X (X/Open XBSA インターフェース) です。
ENTRY_STATUS	CHAR(1)	履歴ファイル内の項目の状況 ID。可能な値は、A (アクティブ)、D (削除済み (将来の利用))、E (期限切れ)、I (非アクティブ)、N (まだコミットされていない)、Y (コミット済みまたはアクティブ) です。

表 271. DB\_HISTORY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明
SQLCAID	VARCHAR(8)	SQL 連絡域 (SQLCA) の SQLCAID フィールドに現れるとおりの、'SQLCA' の入ったストレージ・ダンプの「目印」。
SQLCABC	INTEGER	SQLCA の SQLCABC フィールドに現れるとおりの SQLCA の長さ。
SQLCODE	INTEGER	SQLCA の SQLCODE フィールドに現れるとおりの SQL 戻りコード。
SQLERRML	SMALLINT	SQLCA の SQLERRML フィールドに現れるとおりの SQLERRMC の長さ標識。
SQLERRMC	VARCHAR(70)	SQLCA の SQLERRMC フィールドに現れるとおりの、X'FF' で区切られた 1 つ以上のトークンが入ります。これらのトークンは、エラー条件の説明の中の変数を置き換えます。
SQLERRP	VARCHAR(8)	SQLCA の SQLERRP フィールドに現れるとおりの、製品を示す 3 文字の ID の後に、製品のバージョン、リリース、および修正レベルを示す 5 文字の英数字が続きます。
SQLERRD1	INTEGER	「SQL リファレンス 第 1 巻」の『SQLCA (SQL 連絡域)』を参照。
SQLERRD2	INTEGER	「SQL リファレンス 第 1 巻」の『SQLCA (SQL 連絡域)』を参照。
SQLERRD3	INTEGER	「SQL リファレンス 第 1 巻」の『SQLCA (SQL 連絡域)』を参照。
SQLERRD4	INTEGER	「SQL リファレンス 第 1 巻」の『SQLCA (SQL 連絡域)』を参照。
SQLERRD5	INTEGER	「SQL リファレンス 第 1 巻」の『SQLCA (SQL 連絡域)』を参照。
SQLERRD6	INTEGER	「SQL リファレンス 第 1 巻」の『SQLCA (SQL 連絡域)』を参照。

表 271. DB\_HISTORY 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明
SQLWARN	VARCHAR(11)	一連の警告標識で、それぞれの標識はブランクか 'W' です。「SQL リファレンス 第 1 巻」の『SQLCA (SQL 連絡域)』を参照。
SQLSTATE	VARCHAR(5)	SQLCA の SQLSTATE フィールドに現れるとおりの、直前に実行された SQL ステートメントの結果を示す戻りコード。

表 272. OPERATION および OPERATIONTYPE 値

操作値	操作値の説明	操作タイプ
A	表スペースの追加	なし
B	バックアップ	操作タイプは、以下のとおりです。 <ul style="list-style-type: none"> <li>• D = 差分オフライン</li> <li>• E = 差分オンライン</li> <li>• F = オフライン</li> <li>• I = 増分オフライン</li> <li>• N = オンライン</li> <li>• O = 増分オンライン</li> </ul>
C	ロード・コピー	なし
D	ドロップ済み表	なし
F	ロールフォワード	操作タイプは、以下のとおりです。 <ul style="list-style-type: none"> <li>• E = ログの終わり</li> <li>• P = ポイント・イン・タイム</li> </ul>
G	表の再編成	操作タイプは、以下のとおりです。 <ul style="list-style-type: none"> <li>• F = オフライン</li> <li>• N = オンライン</li> </ul>
L	ロード	操作タイプは、以下のとおりです。 <ul style="list-style-type: none"> <li>• I = 挿入</li> <li>• R = 置換</li> </ul>
N	表スペースの名前変更	なし
O	表スペースのドロップ	なし



表 272. OPERATION および OPERATIONTYPE 値 (続き)

操作値	操作値の説明	操作タイプ
Q	静止	操作タイプは、以下のとおりです。 <ul style="list-style-type: none"> <li>• S = 静止共有</li> <li>• U = 静止更新</li> <li>• X = 静止排他</li> <li>• Z = 静止リセット</li> </ul>
R	リストア	操作タイプは、以下のとおりです。 <ul style="list-style-type: none"> <li>• F = オフライン</li> <li>• I = 増分オフライン</li> <li>• N = オンライン</li> <li>• O = 増分オンライン</li> <li>• R = 再ビルド</li> </ul>
T	表スペースの変更	操作タイプは、以下のとおりです。 <ul style="list-style-type: none"> <li>• C = コンテナの追加</li> <li>• R = リバランス</li> </ul>
U	アンロード	なし
X	アーカイブ・ログ	操作タイプは、以下のとおりです。 <ul style="list-style-type: none"> <li>• F = 障害アーカイブ・パス</li> <li>• M = ミラー・ログ・パス</li> <li>• N = ARCHIVE LOG コマンドを介しての強制切り捨て</li> <li>• P = 1 次ログ・パス</li> <li>• 1 = 第 1 ログ・アーカイブ方法</li> <li>• 2 = 第 2 ログ・アーカイブ方法</li> </ul>

## DBPATHS 管理ビューおよび ADMIN\_LIST\_DB\_PATHS 表関数 - データベース・パスの検索

DBPATHS 管理ビューと ADMIN\_LIST\_DB\_PATHS 表関数は、スプリット・ミラー・バックアップ作成などのタスクに必要なデータベース・パスの値を戻します。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『DBPATHS 管理ビュー』
- 1220 ページの『ADMIN\_LIST\_DB\_PATHS 表関数』

### DBPATHS 管理ビュー

スキーマは SYSIBMADM です。

## 許可

以下のいずれかの権限が必要です。

- DBPATHS 管理ビューに対する SELECT 特権
- DBPATHS 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

以下のステートメントは、すべてのデータベース・パスを検索します。

```
select dbpartitionnum, substr(type,1,20) as type, path from sysibmadm.dbpaths
```

以下はこの照会の出力例です。

DBPARTITIONNUM	TYPE	PATH
0	LOGPATH	/home/sun/sun/NODE0000/SQL00001/LOGSTREAM0000/
0	DB_STORAGE_PATH	/home/sun/
0	LOCAL_DB_DIRECTORY	/home/sun/sun/NODE0000/sqlbdir/
0	DBPATH	/home/sun/sun/NODE0000/SQL00001/
0	DBPATH	/home/sun/sun/NODE0000/SQL00001/MEMBER0000/

5 record(s) selected.

E

## ADMIN\_LIST\_DB\_PATHS 表関数

ADMIN\_LIST\_DB\_PATHS 表関数は、スプリット・ミラー・バックアップなどのバックアップ・メカニズムに必要なファイルのリストを戻します。

## 構文

```
▶▶—ADMIN_LIST_DB_PATHS—(—)——▶▶
```

スキーマは SYSPROC です。

## 許可

以下のいずれかの権限が必要です。

- ADMIN\_LIST\_DB\_PATHS 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

次のようにして ADMIN\_LIST\_DB\_PATHS 表関数を呼び出すことができます。

```
SELECT DBPARTITIONNUM, TYPE, PATH FROM TABLE(ADMIN_LIST_DB_PATHS()) AS FILES
```

以下はこの照会の出力例です。

```
DBPARTITIONNUM TYPE
-----
                2 LOGPATH
                2 DB_STORAGE_PATH
                2 TBSP_DIRECTORY
                2 TBSP_CONTAINER
                2 LOCAL_DB_DIRECTORY
                2 DBPATH
                2 DBPATH
                0 LOGPATH
                0 DB_STORAGE_PATH
                0 TBSP_DIRECTORY
                0 TBSP_CONTAINER
                0 LOCAL_DB_DIRECTORY
                0 DBPATH
                0 DBPATH
                1 LOGPATH
                1 DB_STORAGE_PATH
                1 TBSP_DIRECTORY
                1 TBSP_CONTAINER
                1 LOCAL_DB_DIRECTORY
                1 DBPATH
                1 DBPATH

PATH
-----
/home/sun/sun/NODE0002/SQL00001/LOGSTREAM0002/
/home/sun/
/home/sun/tablespace/sms/sms2/
/home/sun/tablespace/dms/dms2
/home/sun/sun/NODE0002/sqlbdbir/
/home/sun/sun/NODE0002/SQL00001/
/home/sun/sun/NODE0002/SQL00001/MEMBER0002/
/home/sun/sun/NODE0000/SQL00001/LOGSTREAM0000/
```

```

/home/sun/
/home/sun/tablespace/sms/sms0/
/home/sun/tablespace/dms/dms0
/home/sun/sun/NODE0000/sqlbdir/
/home/sun/sun/NODE0000/SQL00001/
/home/sun/sun/NODE0000/SQL00001/MEMBER0000/
/home/sun/sun/NODE0001/SQL00001/LOGSTREAM0001/
/home/sun/
/home/sun/tablespace/sms/sms1/
/home/sun/tablespace/dms/dms1
/home/sun/sun/NODE0001/sqlbdir/
/home/sun/sun/NODE0001/SQL00001/
/home/sun/sun/NODE0001/SQL00001/MEMBER0001/

```

21 record(s) selected.

スプリット・ミラー操作を実行するストレージ・ライブラリーが、通常のファイル・システムとは異なる方法でロー・デバイス上のファイルとディレクトリーを扱う場合、以下の照会を使用して、ロー・デバイス上のすべての場所のリストを取得することができます。

```

SELECT DBPARTITIONNUM, TYPE, PATH
FROM TABLE(ADMIN_LIST_DB_PATHS()) AS FILES
WHERE TYPE LIKE '%_DEVICE%'

```

次に、通常のファイル・システム上のファイルとディレクトリーのリストを以下のようにして取得します。

```

SELECT DBPARTITIONNUM, TYPE, PATH
FROM TABLE(ADMIN_LIST_DB_PATHS()) AS FILES
WHERE TYPE NOT LIKE '%_DEVICE%'

```

## 戻される情報

表 273. DBPATHS 管理ビューおよび ADMIN\_LIST\_DB\_PATHS 表関数によって戻される情報

列名	データ・タイプ	説明
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
TYPE	VARCHAR(64)	パスが属するデータベース・オブジェクトのタイプを記述します。例えば、LOGPATH データベース構成パラメーターが示すログ・ディレクトリーへのパスは、この列で LOGPATH と表示されます。考えられる戻り値のリストについては、1223 ページの表 274を参照してください。

表 273. DBPATHS 管理ビューおよび ADMIN\_LIST\_DB\_PATHS 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
PATH	VARCHAR(5000)	データベース・マネージャーのファイルまたはディレクトリーが見つかった場所へのパス。パスがファイル・システム区切り文字 (UNIX 環境では /、Windows 環境では \) で終わる場合、パスはディレクトリーを指します。

表 274. TYPE 列の値

タイプ値	説明
TBSP_DEVICE	データベース管理スペース (DMS) 表スペースのロー・デバイス。
TBSP_CONTAINER	DMS 表スペースのファイル・コンテナ。
TBSP_DIRECTORY	システム管理スペース (SMS) 表スペースのディレクトリー。
LOGPATH	1 次ログ・パス。
LOGPATH_DEVICE	1 次ログ・パスのロー・デバイス。
MIRRORLOGPATH	データベース構成ミラー・ログ・パス。
DB_STORAGE_PATH	自動ストレージ・パス。
DBPATH	データベース・ディレクトリー・パス。
LOCAL_DB_DIRECTORY	ローカル・データベース・ディレクトリーへのパス。

- 自動ストレージを使用する表スペースの場合、使用済みおよび未使用のストレージ・パスの両方が戻されます。スプリット・ミラー・バックアップを使用してリストアする場合には、未使用の自動ストレージ・パスが必要です。

以下の例について考慮します。実動システムでスプリット・ミラー・バックアップを取ります。バックアップ完了後に、バックアップ前は未使用だった自動ストレージ・パスが実動で使用されるようになります。ここで、スプリット・ミラー・バックアップを使用してデータベースをリストアする必要が生じるとします。この時点で、実動データベースからログをロールフォワードする必要があります。ログをロールフォワードするには、自動ストレージ・パスがすべて必要です。現在、すべての自動ストレージ・パスが使用中だからです。

- 自動ストレージの管理下にある表スペース・コンテナは、個別には戻されません。それらは自動ストレージ・パス列に反映されます。
- 自動ストレージ・パスはデータベース・パーティションごとに 1 回ずつ戻されません。
- **logpath** および **mirrorlogpath** 構成パラメーターの戻される値は、メモリー内に保管されている値です。ディスクに保管されている、変更された値 (データベースの再始動後に初めて適用される) は、戻されません。

- `SELECT * FROM SYSIBMADM.DBPATHS` 照会からの出力を使用して `db2relocatedb` コマンド構成ファイル (データベースの再配置に必要な構成情報を記載したファイル) を作成する場合、それに応じて `DBPATH` 出力を修正してから、構成ファイル内で使用する必要があります。

例えば、以下の `DBPATH` 出力について考えてみます。

```
/storage/svtdbm3/svtdbm3/NODE0000/SQL00001/
```

以下のように、この出力を使用して、`db2relocatedb` コマンド構成ファイル内で `DB_PATH` パラメーターの値を指定できます。

```
DB_PATH=/storage/svtdbm3,/storage_copy2/svtdbm3
```

- `LOCAL_DB_DIRECTORY` パスには、複数のデータベースに属する情報を入れることができます。同じディレクトリー内に作成した各データベースには独自の `sqlbdir` ファイルがないので、ファイルのコピー先のターゲット・システムのそのパスに、既に存在するデータベースがないことを確認してください。
- 複数のデータベースが、少なくとも 1 つの自動ストレージ・パスを共有する場合、それらのデータベースのいずれかに対してスプリット・ミラー操作を実行すると、複数のデータベースがその影響を受け、それによって、分割の予定のなかったデータベースで入出力上の問題が起きることがあります。
- `DB_STORAGE_PATH` タイプには、定義されたストレージ・グループすべてからのストレージ・パスがすべて含まれます。ストレージ・パスが複数のストレージ・グループで使用されていたり、同じストレージ・グループで複数回指定されていたりする場合は、ストレージ・グループ内でのそのパスのオカレンスにつき 1 つずつレコードが戻されます。

## 制約事項

データベースが `WRITE SUSPEND` モードのときにはこの管理ビューを呼び出せません。また、ビューを呼び出すときと `WRITE SUSPEND` モードでアクティブ化するときにデータベースの物理レイアウトが変わらないようにしなければなりません。これはスプリット・ミラー操作を実行するために必要です。例えば、その時に表スペースのレイアウトが変わっていると、スプリット・ミラー・バックアップ・イメージからリストアできない可能性があります。

## GET\_DBSIZE\_INFO

`GET_DBSIZE_INFO` プロシージャは、データベース・サイズと最大容量を計算します。

## 構文

```
▶▶ GET_DBSIZE_INFO (—snapshot-timestamp—, —dbsize—, —dbcapacity—, —————▶
▶—refresh-window—) —————▶▶
```

スキーマは `SYSPROC` です。

## プロシージャ・パラメーター

*snapshot-timestamp*

*dbsize* および *dbcapacity* が計算された時刻を戻す、タイプ `TIMESTAMP` の出

力パラメーター。このタイム・スタンプは、*refresh-window* の値と共に、SYSTOOLS.STMG\_DBSIZE\_INFO 表のキャッシュ値を更新しなければならない時刻を判別するのに使用されます。

#### *dbsize*

データベースのサイズを (バイト単位で) 戻すタイプ BIGINT の出力パラメーター。データベースのサイズは、各表スペース (SMS および DMS) ごとに、次のように計算されます。 $dbsize = \text{sum}(\text{使用されているページ数} * \text{ページ・サイズ})$ 。

#### *dbcapacity*

データベース容量を (バイト単位で) 戻すタイプ BIGINT の出力パラメーター。この値は、パーティション・データベース・システムでは使用できません。データベースの容量は、次のように計算されます。 $dbcapacity = \text{SUM}(\text{DMS 使用可能ページ数} * \text{ページ・サイズ}) + \text{SUM}(\text{SMS コンテナ・サイズ} + \text{コンテナあたりのファイル・システムの空きサイズ})$ 。同じファイル・システムに複数の SMS コンテナが定義されている場合、容量計算にファイル・システムの空きサイズが含まれるのは 1 回だけです。

#### *refresh-window*

データベース・サイズおよび容量のキャッシュ値を更新するまでの分数を指定する、タイプ INTEGER の入力引数。30 分のデフォルト更新枠の場合は -1 を指定します。更新枠が 0 であると、キャッシュ値の更新が即時に強制実行されます。

### 許可

- SYSMON 権限
- GET\_DBSIZE\_INFO プロシージャに対する EXECUTE 特権。

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

例 1: 30 分のデフォルト更新枠を使用して、データベース・サイズおよび容量を入手します。データベース・サイズおよび容量は、キャッシュ・データが 30 分よりも前のものであると再計算されます。

```
CALL GET_DBSIZE_INFO(?, ?, ?, -1)
```

プロシージャは、次を戻します。

Value of output parameters

```
-----  
Parameter Name : SNAPSHOTTIMESTAMP  
Parameter Value : 2004-02-29-18.31.55.178000
```

```
Parameter Name : DATABASESIZE  
Parameter Value : 22302720
```

```
Parameter Name : DATABASECAPACITY  
Parameter Value : 4684793856
```

```
Return Status = 0
```

例 2: 0 分の更新枠を使用して、データベース・サイズおよび容量を入手します。データベース・サイズおよび容量がすぐに再計算されます。

```
CALL GET_DBSIZE_INFO(?, ?, ?, 0)
```

プロシージャは、次を戻します。

Value of output parameters

```
-----  
Parameter Name : SNAPSHOTTIMESTAMP  
Parameter Value : 2004-02-29-18.33.34.561000
```

```
Parameter Name : DATABASESIZE  
Parameter Value : 22302720
```

```
Parameter Name : DATABASECAPACITY  
Parameter Value : 4684859392
```

Return Status = 0

例 3: 24 時間の更新枠を使用して、データベース・サイズおよび容量を入手します。データベース・サイズおよび容量は、キャッシュ・データが 1440 分よりも前のものであると再計算されます。

```
CALL GET_DBSIZE_INFO(?, ?, ?, 1440)
```

プロシージャは、次を戻します。

Value of output parameters

```
-----  
Parameter Name : SNAPSHOTTIMESTAMP  
Parameter Value : 2004-02-29-18.33.34.561000
```

```
Parameter Name : DATABASESIZE  
Parameter Value : 22302720
```

```
Parameter Name : DATABASECAPACITY  
Parameter Value : 4684859392
```

Return Status = 0

## 使用上の注意

計算値は、プロシージャ出力パラメーターとして戻され、SYSTOOLS.STMG\_DBSIZE\_INFO 表にキャッシュされます。プロシージャがこの値をキャッシュに入れるのは、計算にコストがかかるためです。SYSTOOLS.STMG\_DBSIZE\_INFO 表は、初めてプロシージャを実行すると自動的に作成されます。SYSTOOLS.STMG\_DBSIZE\_INFO 表にキャッシュされた値が存在し、それが最新であれば、*snapshot-timestamp* および *refresh-window* 値で決定されたように、それらのキャッシュに入れられた値が戻されます。キャッシュに入れられた値が最新でない場合、キャッシュに入れられた新しい値が計算され、SYSTOOLS.STMG\_DBSIZE\_INFO 表に挿入されて戻され、*snapshot-timestamp* 値が更新されます。

グローバル表スペース・スナップショットのデータがすべてのパーティションから必ず戻されるようにするには、データベースをアクティブにしておく必要があります。

SYSTOOLSPACE をルーチンの操作表に使用してメタデータを保管します。つまり、データベース・オブジェクトとその操作の記述に使用されるデータです。



## NOTIFICATIONLIST 管理ビュー - ヘルス通知の連絡先リストの検索

NOTIFICATIONLIST 管理ビューは、インスタンスの状況が通知される連絡先および連絡先グループのリストを戻します。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- NOTIFICATIONLIST 管理ビューに対する SELECT 特権
- NOTIFICATIONLIST 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

NOTIFICATIONLIST 管理ビューは、インスタンスの状況が通知される連絡先および連絡先グループのリストを戻します。

以下はこの照会の出力例です。

```
NAME                TYPE
-----
group3              GROUP
user4              CONTACT
group3              GROUP
```

3 record(s) selected.

### 戻される情報

表 275. NOTIFICATIONLIST 管理ビューによって戻される情報

列名	データ・タイプ	説明
NAME	VARCHAR(128)	連絡先の名前。
TYPE	VARCHAR(7)	連絡先のタイプ • 'CONTACT' • 'GROUP'

## PD\_GET\_DIAG\_HIST - 指定された機能からレコードを戻す

PD\_GET\_DIAG\_HIST 関数は、指定された機能からログ・レコード、イベント・レコード、および通知レコードを戻します。また、レコードのタイプ、レコードの顧客影響値、および from-until タイム・スタンプに基づいてフィルター操作を行うためのオプションもサポートされています。

## 構文

```
▶▶—PD_GET_DIAG_HIST—(—facility—,—rectype—,—impact—,—start_time—,—end_time—,—member—)————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *facility*

レコードが戻される機能を指定する、タイプ VARCHAR(20) のオプション入力引数。機能とは、レコードが関連付けられる論理グループです。可能な値は次のとおりです。

- ALL: すべての機能からレコードを戻します。
- MAIN: DB2 一般診断ログからレコードを戻します。現時点では、**db2diag** ログ・ファイル、管理通知ログ、および循環イベント・ログのことです。
- OPTSTATS: オプティマイザー統計に関連したレコードを戻します。

このパラメーターが NULL または空ストリング (" ) の場合、「ALL」がデフォルトです。

#### *rectype*

戻すレコード・タイプを指定する、タイプ VARCHAR(30) のオプション入力引数。 '+' で区切られた組み合わせのタイプがサポートされます。例えば、'D + EI' などです。可能な値は次のとおりです。

- 「ALL」: すべてのレコード・タイプを戻します。
- 「D」: すべての診断レコードを戻します。
- 「E」: すべてのイベント・レコードを戻します。
- 「DI」: 内部診断レコード。これらは、診断状況で IBM サポートによって使用される翻訳されていない診断レコードです。
- 「DX」: 外部診断レコード。これらは、ユーザーに役立つ翻訳済みの診断です。これらのレコードは通知レコードです。
- 「EI」: 内部イベント・レコード。これらは、診断状況で IBM サポートによって使用されるイベント・レコードです。
- 「EX」: 外部イベント・レコード。これらは、ユーザーに役立つ診断レコードです。

このパラメーターが NULL または空ストリング (" ) の場合、すべてのレコードが戻されます。

#### *impact*

戻されるレコードの最小の顧客影響レベルを指定する、タイプ VARCHAR(18) のオプション入力引数。可能な値は次のとおりです。

- 'NONE'
- 'UNLIKELY'
- 'POTENTIAL'
- 'IMMEDIATE'
- 'CRITICAL'

このパラメーターが NULL または空ストリング (") の場合、すべてのレコードが戻されます。

#### *start\_time*

有効なタイム・スタンプを指定する、タイプ **TIMESTAMP** のオプション入力引数。項目のタイム・スタンプがこの値より新しい場合は、それらの項目が戻されます。このパラメーターが NULL の場合は、項目の古さに関係なくレコードが戻されます。

#### *end\_time*

有効なタイム・スタンプを指定する、タイプ **TIMESTAMP** のオプション入力引数。項目のタイム・スタンプがこの値より古い場合は、それらの項目が戻されます。このパラメーターが NULL の場合は、項目の新しさに関係なくレコードが戻されます。

#### *member*

レコードの取り出し元である有効なデータベース・メンバーを指定する、タイプ **INTEGER** のオプションの入力引数。現行メンバーの場合は -1 または NULL、すべてのアクティブ・データベース・メンバーからの情報の場合は -2 を指定します。アクティブ・データベース・メンバーとは、アプリケーションから接続して使用できるデータベースのメンバーのことです。このパラメーターが指定されない場合は、デフォルトの値がすべてのアクティブ・データベース・メンバーになります。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: 指定された機能からレコードを取得します。

```
SELECT FACILITY, RECTYPE, TIMESTAMP, IMPACT, SUBSTR(MSG,1, 50) AS MSG
FROM TABLE (PD_GET_DIAG_HIST( 'MAIN', 'E', '', NULL,
NULL ) ) AS T
WHERE T.PROCESS_NAME = 'db2star2' OR T.PROCESS_NAME = 'db2stop2'
```

以下はこの照会の出力例です。

FACILITY	RECTYPE	TIMESTAMP	...
-----	-----	-----	-----
MAIN	EX	2007-06-25-11.34.05.756171	...
MAIN	EX	2007-06-25-11.34.25.946646	...

2 record(s) selected.

この照会からの出力 (続き)。

```
... IMPACT          MSG
... -----
... -                ADM7514W Database manager has stopped.
... -                ADM7513W Database manager has started.
```

例 2: 特定のメンバーのレコードを取得します。

```
SELECT MEMBER,DBPARTITIONNUM, FACILITY, RECTYPE, TIMESTAMP, IMPACT,
SUBSTR(MSG,1, 50) AS MSG FROM TABLE (PD_GET_DIAG_HIST('MAIN', 'E', '',
CAST (NULL AS TIMESTAMP), CAST (NULL AS_TIMESTAMP), NULL ) ) AS T
WHERE T.PROCESS_NAME = 'db2star2' OR T.PROCESS_NAME = 'db2stop2' ORDER BY MEMBER
```

以下はこの照会の出力例です。

MEMBER	DBPARTITIONNUM	FACILITY	RECTYPE	TIMESTAMP	...
0	0	MAIN	EI	2011-04-28-09.44.57.720041	...
0	0	MAIN	EI	2011-04-28-09.44.57.723015	...
0	0	MAIN	EI	2011-04-28-09.44.57.723736	...
0	0	MAIN	EI	2011-04-28-09.44.59.409586	...
0	0	MAIN	EX	2011-04-28-09.45.01.554096	...
0	0	MAIN	EI	2011-04-28-09.45.01.605231	...
0	0	MAIN	EI	2011-04-28-12.34.20.571551	...
0	0	MAIN	EI	2011-04-28-12.34.20.574612	...
0	0	MAIN	EI	2011-04-28-12.34.20.575323	...
0	0	MAIN	EI	2011-04-28-12.34.20.602452	...
0	0	MAIN	EI	2011-04-28-12.34.20.665227	...
0	0	MAIN	EI	2011-04-28-09.44.57.715392	...

12 record(s) selected.

この照会からの出力 (続き)。

```
... IMPACT          MSG
... -----
... -                -
... -                -
... -                Obtained exclusive mode lock on the file:
... -                -
... -                ADM7513W Database manager has started.
... -                Released lock on the file:
... -                -
... -                -
... -                Obtained exclusive mode lock on the file:
... -                ZRC=0xFFFFFBFE=-1026
... -                Released lock on the file:
... -                -
```

## 使用上の注意

PD\_GET\_DIAG\_HIST 表関数では、関連データベースに最低 8K のページ・サイズを持つ TEMPORARY 表スペースがあることが必要です。ページ・サイズが 8K 未満である場合にこの関数を使用すると、SQL1585N エラー・メッセージが戻されます。

## 戻される情報

表 276. PD\_GET\_DIAG\_HIST 表関数によって戻される情報

列名	データ・タイプ	説明
FACILITY	VARCHAR(20)	機能とは、レコードが関連付けられる論理グループです。可能な値は次のとおりです。 <ul style="list-style-type: none"> <li>• ALL: すべての機能からレコードを戻します。</li> <li>• MAIN: DB2 一般診断ログからレコードを戻します。現時点では、<b>db2diag</b> ログ・ファイル、管理通知ログ、および循環イベント・ログのことです。</li> <li>• OPTSTATS: オプティマイザ統計に関連したレコードを戻します。</li> </ul>
RECTYPE	VARCHAR(3)	レコードのタイプ。可能な値は次のとおりです。 <ul style="list-style-type: none"> <li>• 「DI」: 内部診断レコード</li> <li>• 「DX」: 外部診断レコード</li> <li>• 「EI」: 内部イベント・レコード</li> <li>• 「EX」: 外部イベント・レコード</li> </ul>
TIMESTAMP	TIMESTAMP	メッセージが作成された時刻。
TIMEZONE	INTEGER	協定世界時 (UCT) との時差 (分単位)。例えば、-300 は米東部標準時です。
INSTANCENAME	VARCHAR(128)	メッセージが作成されたインスタンスの名前。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
LEVEL	CHAR(1)	レコードの重大度レベル。可能な値は次のとおりです。 <ul style="list-style-type: none"> <li>• 「C」: クリティカル</li> <li>• 「E」: エラー</li> <li>• 「I」: 通知</li> <li>• 「S」: 重大</li> <li>• 「W」: 警告</li> </ul>
IMPACT	VARCHAR(18)	ユーザーの視点からこのメッセージの影響を限定します。これは、DB2 が構成要素となっているビジネス・プロセスに関するメッセージの影響を明確にします。可能な値は次のとおりです。 <ul style="list-style-type: none"> <li>• 'CRITICAL'</li> <li>• 'IMMEDIATE'</li> <li>• 'NONE'</li> <li>• 'POTENTIAL'</li> <li>• 'UNLIKELY'</li> </ul>
DBNAME	VARCHAR(128)	このメッセージの作成中にアクセスされているデータベースの名前。
EDU_ID	BIGINT	edu_ID - エンジン・ディスパッチ可能単位 ID モニター・エレメント
EDUNAME	VARCHAR(64)	このメッセージを作成したエンジン・ディスパッチ可能単位の名前。
PID	BIGINT	このメッセージを作成したオペレーティング・システム・プロセス ID。
PROCESS_NAME	VARCHAR(255)	このメッセージを作成したオペレーティング・システム・プロセス名。
TID	BIGINT	このメッセージを作成したスレッド数値 ID。

表 276. PD\_GET\_DIAG\_HIST 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
APPLNAME	VARCHAR(255)	接続を開始したクライアント・アプリケーションの名前 (使用可能な場合)。
APPL_ID	VARCHAR(64)	appl_id - アプリケーション ID モニター・エレメント
APPLHANDLE	VARCHAR(9)	使用可能な場合に接続を開始したアプリケーションの、システム全体におけるユニーク ID。これはエージェント ID と同義です。ID は、調整メンバー番号と 16 ビットのカウンターを '!' で区切ったもので構成されます。形式は、次のとおりです。 'nnn-xxxxx'
AUTH_ID	VARCHAR(30)	auth_id - 許可 ID モニター・エレメント
PRODUCT	VARCHAR(50)	メッセージを作成したプロダクトの名前。例えば、「DB2 Common」などです。
COMPONENT	VARCHAR(255)	メッセージを作成したコンポーネントの名前。
FUNCTION	VARCHAR(255)	メッセージを生成した関数の名前。
PROBE	INTEGER	メッセージが関数で生成された場所を識別するために使用されるプローブ・ポイント番号。
CALLEDPRODUCT	VARCHAR(50)	エラーのソースにあるプロダクトの名前。エラーのソースがメッセージが作成された場所ではない場合にこれが使用されます。
CALLEDCOMPONENT	VARCHAR(255)	エラーのソースにあるコンポーネントの名前。エラーのソースがメッセージが作成された場所ではない場合にこれが使用されます。
CALLEDFUNCTION	VARCHAR(255)	エラーのソースにある関数の名前。エラーのソースがメッセージが作成された場所ではない場合にこれが使用されます。
OSERR	INTEGER	オペレーティング・システム・エラー番号。
RETCODE	INTEGER	プロダクト固有の戻りコード。
MSGNUM	INTEGER	関連メッセージの数値メッセージ番号 (使用可能な場合)。例えば、これは ADM7513W の数値部分です。
MSGTYPE	CHAR(3)	メッセージ ID に関連したタイプ (使用可能な場合)。例えば、ADM は管理通知ログ・メッセージに使用されます。
MSG	CLOB(16KB)	このレコードの簡略説明テキスト。これは、翻訳済みメッセージの MSGNUM、および MSGTYPE に対応する翻訳済みメッセージ・テキストです。翻訳済みでないメッセージの場合、これは簡略説明です。例えば、「Bringing down all db2fmp processes as part of db2stop」のようになります。
OBJTYPE	VARCHAR(64)	objtype - オブジェクト・タイプ・モニター・エレメント
OBJNAME	VARCHAR(255)	イベントが関連するオブジェクトの名前 (使用可能な場合)。
OBJNAME_QUALIFIER	VARCHAR(255)	オブジェクトの追加情報 (使用可能な場合)。

表 276. PD\_GET\_DIAG\_HIST 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
EVENTTYPE	VARCHAR(24)	<p>イベント・タイプは、このイベントに関連したアクションまたは verb です。可能な値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• 'ACCEPT'</li> <li>• 'ACCESS'</li> <li>• 'ADD'</li> <li>• 'ALTER'</li> <li>• 'ASSOCIATE'</li> <li>• 'AVAILABLE'</li> <li>• 'BRINGDOWN'</li> <li>• 'CHANGE'</li> <li>• 'CHANGECFG'</li> <li>• 'CLOSE'</li> <li>• 'COLLECT'</li> <li>• 'CONNECT'</li> <li>• 'CREATE'</li> <li>• 'DEPENDENCY'</li> <li>• 'DESTROY'</li> <li>• 'DISASSOCIATE'</li> <li>• 'DISCONNECT'</li> <li>• 'DISPATCH'</li> <li>• 'DROP'</li> <li>• 'FINI'</li> <li>• 'FREE'</li> <li>• 'GET'</li> <li>• 'INIT'</li> <li>• 'INTERRUPT'</li> <li>• 'OPEN','READ'</li> <li>• 'RECV'</li> <li>• 'REPLY'</li> <li>• 'REPORT'</li> <li>• 'REQUEST'</li> <li>• 'RESET'</li> <li>• 'SEND'</li> <li>• 'START'</li> <li>• 'STARTUP'</li> <li>• 'STOP'</li> <li>• 'SWITCH'</li> <li>• 'TERMINATE'</li> <li>• 'TRANSFER'</li> <li>• 'WAIT'</li> <li>• 'WORK'</li> <li>• 'WRITE'</li> </ul>
EVENTDESC	VARCHAR(256)	このイベントのキー・フィールドの簡略表現。

表 276. PD\_GET\_DIAG\_HIST 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
FIRST_EVENTQUALIFIERTYPE	VARCHAR(64)	最初のイベント修飾子のタイプ。イベント修飾子は、イベントの影響を受けたものを説明するために使用されます。可能な値は次のとおりです。 <ul style="list-style-type: none"> <li>• 'AT'</li> <li>• 'BY'</li> <li>• 'CONTEXT'</li> <li>• 'DUE TO'</li> <li>• 'FOR'</li> <li>• 'FROM'</li> <li>• 'ON'</li> <li>• 'TO'</li> </ul> <i>facility</i> が OPTSTATS の場合、値は 'AT' のみです。
FIRST_EVENTQUALIFIER	CLOB(16K)	イベントの最初の修飾子。 <i>facility</i> が OPTSTATS の場合、これは統計収集が行われた時を示すタイム・スタンプになります。
SECOND_EVENTQUALIFIERTYPE	VARCHAR(64)	2 番目のイベント修飾子のタイプ。 <i>facility</i> が OPTSTATS の場合、値は 'BY' です。
SECOND_EVENTQUALIFIER	CLOB(16K)	イベントの 2 番目の修飾子。 <i>facility</i> が OPTSTATS の場合、使用できる値は次のとおりです。 <ul style="list-style-type: none"> <li>• Asynchronous</li> <li>• FABRICATE</li> <li>• FABRICATE PARTIAL</li> <li>• SYNCHRONOUS</li> <li>• SYNCHRONOUS SAMPLED</li> <li>• USER</li> </ul>
THIRD_EVENTQUALIFIERTYPE	VARCHAR(64)	3 番目のイベント修飾子のタイプ。 <i>facility</i> が OPTSTATS の場合、値は 'DUE TO' です。
THIRD_EVENTQUALIFIER	CLOB(16K)	イベントの 3 番目の修飾子。 <i>facility</i> が OPTSTATS の場合、使用できる値は次のとおりです。 <ul style="list-style-type: none"> <li>• 競合</li> <li>• エラー</li> <li>• 使用不可のオブジェクト</li> <li>• RUNSTATS エラー</li> <li>• タイムアウト</li> </ul>
EVENTSTATE	VARCHAR(255)	イベントの結果としてのオブジェクトまたはアクションの状態。また、これにはイベントの進捗を示すパーセントも含めることができます。



表 276. PD\_GET\_DIAG\_HIST 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
EVENTATTRIBUTE	VARCHAR(255)	イベント属性。これはイベントに関連した属性のリストです。複数の属性が使用される場合、リストは '+' 文字で区切られます。例えば、'CACHED + LOGICAL + AUTO' などです。可能な値は次のとおりです。 <ul style="list-style-type: none"> <li>• 'ASYNC'</li> <li>• 'AUTO'</li> <li>• 'CACHED'</li> <li>• 'DIRECT'</li> <li>• 'EXTERNAL'</li> <li>• 'INDIRECT'</li> <li>• 'INTERNAL'</li> <li>• 'LOGICAL'</li> <li>• 'PERMANENT'</li> <li>• 'PHYSICAL'</li> <li>• 'SYNC'</li> <li>• 'TEMPORARY'</li> </ul>
EVENTSTACK	CLOB(16K)	該当する場合に、レコードが記録された地点の論理イベント・スタック。
CALLSTACK	CLOB(16K)	該当する場合に、このレコードを生成したスレッドのオペレーティング・システム・スタック・ダンプ。
DUMPFIL	CLOB(5000)	該当する場合に、ログ・レコードに関連した 2 次ダンプ・ファイルの名前。これはファイルへの絶対パス、またはメッセージに関連した追加情報を検索できるディレクトリーです。
FULLREC	CLOB(16K)	レコード全体のフォーマット済みテキスト・バージョン。このセクションには追加 DATA フィールドも含まれます。
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
HOSTNAME	VARCHAR(255)	hostname - ホスト名

## PDLOGMSG\_LAST24HOURS 管理ビューおよび PD\_GET\_LOG\_MSGS 表関数 - 問題判別メッセージの検索

PDLOGMSG\_LAST24HOURS 管理ビューおよび PD\_GET\_LOG\_MSGS 表関数は、DB2 通知ログに記録された問題判別ログ・メッセージを戻します。この情報は、データベース管理者とシステム管理者が使用するためのものです。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『PDLOGMSG\_LAST24HOURS 管理ビュー』
- 1238 ページの『PD\_GET\_LOG\_MSGS 表関数』

### PDLOGMSG\_LAST24HOURS 管理ビュー

PDLOGMSG\_LAST24HOURS 管理ビューは、DB2 通知ログに過去 24 時間に記録された問題判別ログ・メッセージを戻します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1242 ページの表 277 を参照してください。

## 許可

以下のいずれかの権限が必要です。

- PDLOGMSG\_LAST24HOURS 管理ビューに対する SELECT 特権
- PDLOGMSG\_LAST24HOURS 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

過去 24 時間に記録されたすべての重大ログ・メッセージを、最新のものから順番に取得します。

```
SELECT * FROM SYSIBMADM.PDLOGMSG_LAST24HOURS
      WHERE MSGSEVERITY = 'C' ORDER BY TIMESTAMP DESC
```

以下はこの照会の出力例です。

TIMESTAMP	TIMEZONE	INSTANCENAME	MEMBER	...
2005-11-23-21.56.41.240066	-300	svtdbm4	0	...
				...
				...
				...
				...
				...
2005-11-23-21.56.39.150597	-300	svtdbm4	0	...
2005-11-23-21.56.37.363384	-300	svtdbm4	0	...
				...
				...
				...
2005-11-23-21.56.35.880314	-300	svtdbm4	0	...
				...

4 record(s) selected.

この照会からの出力 (続き)。

...	DBPARTITIONNUM	DBNAME	PID	PROCESSNAME	...
...					...
...	0	CAPTAIN	4239374	db2agent (CAPTAIN)	0 ...
...					...
...					...
...					...
...					...
...					...
...	0	CAPTAIN	4239374	db2agent (CAPTAIN)	0 ...
...	0	CAPTAIN	4239374	db2agent (CAPTAIN)	0 ...

```

...
...
...
...          0 CAPTAIN          4239374 db2agent (CAPTAIN) 0 ...
...
...

```

この照会からの出力 (続き)。

```

...TID APPL_ID          COMPONENT          ...
-----
...  1 9.26.15.148.36942.051124025612 oper system services ...
...
...
...
...
...
...  1 9.26.15.148.36942.051124025612 base sys utilities ...
...  1 9.26.15.148.36942.051124025612 relation data serv ...
...
...
...
...  1 9.26.15.148.36942.051124025612 relation data serv ...
...
...

```

この照会からの出力 (続き)。

```

... FUNCTION          PROBE  MSGNUM      MSGTYPE ...
-----
... sqloSleepInstance    38      504 ADM      ...
...
...
...
...
...
... sqlMarkDBad          10      7518 ADM     ...
... sqlrr_dump_ffdc      10      1 ADM       ...
...
...
...
... sqlrr_dump_ffdc      10      1 ADM       ...
...

```

この照会からの出力 (続き)。

```

... MSGSEVERITY MSG
-----
... C          ADM0504C An unexpected internal
...           processing error has occurred. ALL
...           DB2 PROCESSES ASSOCIATED WITH THIS
...           INSTANCE HAVE BEEN SUSPENDED.
...           Diagnostic information has been
...           recorded. Contact IBM Support
...           for further assistance.
... C          ADM7518C "CAPTAIN " marked bad.
... C          ADM0001C A severe error has occurred.
...           Examine the administration notification
...           log and contact IBM Support if
...           necessary.
... C          ADM0001C A severe error has occurred.
...           Examine the administration notification
...           log and contact IBM Support if necessary.

```

## PD\_GET\_LOG\_MSGS 表関数

PD\_GET\_LOG\_MSGS 表関数は PDLOGMSG\_LAST24HOURS 管理ビューと同じ情報を戻しますが、この関数では過去 24 時間に限らず、特定の時間枠を指定することが可能です。

戻される可能性のある情報の完全なリストは、1242 ページの表 277 を参照してください。

### 構文

```
▶▶ PD_GET_LOG_MSGS (—oldest_timestamp—, —member—) ◀◀
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *oldest\_timestamp*

有効なタイム・スタンプを指定する、タイプ `TIMESTAMP` の入力引数。最新のタイム・スタンプの項目を先頭に、この入力引数に指定したタイム・スタンプまでのログ項目が戻されます。NULL 値が指定された場合、すべてのログ項目が戻されます。

#### *member*

レコードの取り出し元である有効なデータベース・メンバーを指定する、タイプ `INTEGER` のオプションの入力引数。現行メンバーの場合は `-1` または `NULL`、すべてのアクティブ・データベース・メンバーからの情報の場合は `-2` を指定します。アクティブ・データベース・メンバーとは、アプリケーションから接続して使用できるデータベースのメンバーのことです。クラスター・キャッシング・ファシリティ (CF) を指定すると、このデータを要求するためにアクティブ・メンバーが使用されます。通知ログにアクセスできない場合は、エラーが返されます。

このパラメーターが指定されない場合は、デフォルトの値がすべてのアクティブ・データベース・メンバーになります。

### 許可

以下のいずれかの権限が必要です。

- PD\_GET\_LOG\_MSGS 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: DB2 というインスタンス上のデータベース SAMPLE のすべてのデータベース・パーティションに関する、過去 1 週間にログに記録されたすべての通知メッセージを検索します。メッセージを日時順にレポートします。

```
SELECT TIMESTAMP, APPL_ID, DBPARTITIONNUM, MSG
FROM TABLE ( PD_GET_LOG_MSGS( CURRENT_TIMESTAMP - 7 DAYS)) AS T
WHERE INSTANCENAME = 'DB2' AND DBNAME = 'SAMPLE'
ORDER BY TIMESTAMP ASC
```

以下はこの照会の出力例です。

TIMESTAMP	APPL_ID	DBPARTITIONNUM	...
2005-11-13-12.51.37.772000	*LOCAL.DB2.050324175005	0	...
2005-11-13-12.51.37.772001	*LOCAL.DB2.050324175005	0	...
2005-11-13-12.51.37.781000	*LOCAL.DB2.050324175005	0	...
2005-11-13-12.51.37.781001	*LOCAL.DB2.050324175005	0	...
2005-11-17-14.12.39.036001	*LOCAL.DB2.041117191249	0	...
2005-11-17-14.12.39.056000	*LOCAL.DB2.041117191249	0	...
2005-11-17-14.13.04.450000	*LOCAL.DB2.041117191307	0	...
2005-11-17-14.13.04.460000	*LOCAL.DB2.041117191307	0	...
2005-11-17-14.18.29.042000	*LOCAL.DB2.041117190824	0	...
...			
...			
...			

この照会からの出力 (続き)。

```
... MSG
... -----
... ADM5502W The escalation of "143" locks on table
... "SYSIBM .SYSINDEXAUTH" to lock intent "X" was successful.
... ADM5502W The escalation of "144" locks on table
... "SYSIBM .SYSINDEXES" to lock intent "X" was successful.
... ADM5502W The escalation of "416" locks on table
... "SYSIBM .SYSINDEXCOLUSE" to lock intent "X" was successful.
... ADM5500W DB2 is performing lock escalation. The total
... number of locks currently held is "1129", and the target
... number of locks to hold is "564".
... ADM7506W Database quiesce has been requested.
... ADM7507W Database quiesce request has completed successfully.
... ADM7510W Database unquiesce has been requested.
... ADM7509W Database unquiesce request has completed successfully.
... ADM4500W A package cache overflow condition has occurred. There
... is no error but this indicates that the package cache has
... exceeded the configured maximum size. If this condition persists,
... you may want to adjust the PCKCACHESZ DB configuration parameter.
```

例 2: DB2 というインスタンスのデータベース・パーティション 0 に関する、過去 1 日間にログに記録されたすべてのクリティカル・エラーを、最新のものから順にソートして検索します。

```
SELECT TIMESTAMP, DBNAME, MSG
FROM TABLE (PD_GET_LOG_MSGS(CURRENT_TIMESTAMP - 1 DAYS)) AS T
WHERE MSGSEVERITY = 'C' AND INSTANCENAME = 'DB2' AND
DBPARTITIONNUM = 0
ORDER BY TIMESTAMP DESC
```

以下はこの照会の出力例です。

TIMESTAMP	DBNAME	MSG
2004-11-04-13.49.17.022000	TESTSBCS	ADM0503C An unexpected internal processing error has occurred. ALL DB2 PROCESSES ASSOCIATED WITH THIS INSTANCE HAVE BEEN SHUTDOWN. Diagnostic information has been recorded. Contact IBM Support for further assistance.
2004-11-04-11.32.26.760000	SAMPLE	ADM0503C An unexpected internal processing error has occurred. ALL DB2 PROCESSES ASSOCIATED WITH THIS INSTANCE HAVE BEEN SHUTDOWN. Diagnostic information has been recorded. Contact IBM Support for further assistance.

2 record(s) selected.

例 3: \*LOCAL.DB2.050927195337 というアプリケーション ID の DB2 プロセスによって過去 1 日の間に書き込まれたメッセージを検索します。

```
SELECT TIMESTAMP, MSG
FROM TABLE (PD_GET_LOG_MSGS(CURRENT_TIMESTAMP - 1 DAYS)) AS T
WHERE APPL_ID = '*LOCAL.DB2.050927195337'
```

以下はこの照会の出力例です。

TIMESTAMP	MSG
2005-06-27-21.17.12.389000	ADM4500W A package cache overflow condition has occurred. There is no error but this indicates that the package cache has exceeded the configured maximum size. If this condition persists, you may want to adjust the PCKCACHESZ DB configuration parameter.
2005-06-27-18.41.22.248000	ADM4500W A package cache overflow condition has occurred. There is no error but this indicates that the package cache has exceeded the configured maximum size. If this condition persists, you may want to adjust the PCKCACHESZ DB configuration parameter.
2005-06-27-12.51.37.772001	ADM5502W The escalation of "143" locks on table "SYSIBM .SYSINDEXAUTH" to lock intent "X" was successful.
2005-06-27-12.51.37.772000	ADM5502W The escalation of "144" locks on table "SYSIBM .SYSINDEXES" to lock intent "X" was successful.
2005-06-27-12.51.37.761001	ADM5502W The escalation of "416" locks on table "SYSIBM .SYSINDEXCOLUSE" to lock intent "X" was successful.

...



```

... ADM7513W Database manager has started.
...
... ADM7513W Database manager has started.

```

## 戻される情報

注: 複数メンバー環境では、ログ・メッセージが戻される順序は保証できません。ログ・レコードの順序が重要である場合は、結果をタイム・スタンプでソートする必要があります。

表 277. PDLOGMSG\$LAST24HOURS 管理ビューおよび PD\_GET\_LOG\_MSGS 表関数によって戻される情報

列名	データ・タイプ	説明
TIMESTAMP	TIMESTAMP	項目がログに記録された時刻。
TIMEZONE	INTEGER	協定世界時 (UCT) との時差 (分単位)。例えば、-300 は米東部標準時です。
INSTANCENAME	VARCHAR(128)	メッセージを生成したインスタンスの名前。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
DBNAME	VARCHAR(128)	エラーまたはイベントが発生したデータベース。
PID	BIGINT	メッセージを生成したプロセスのプロセス ID。
PROCESSNAME	VARCHAR(255)	メッセージを生成したプロセスの名前。
TID	BIGINT	メッセージを生成したプロセス内のスレッドの ID。
APPL_ID	VARCHAR(64)	appl_id - アプリケーション ID モニター・エレメント
COMPONENT	VARCHAR(255)	メッセージを提供している DB2 コンポーネントの名前。ユーザー・アプリケーションが db2AdminMsgWrite API を使用して書き込んだメッセージの場合、「User Application」が戻されます。
FUNCTION	VARCHAR(255)	メッセージを提供している DB2 関数の名前。ユーザー・アプリケーションが db2AdminMsgWrite API を使用して書き込んだメッセージの場合、「User Function」が戻されます。



表 277. PDLOGMSGs\_LAST24HOURS 管理ビューおよび PD\_GET\_LOG\_MSGS 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
PROBE	INTEGER	DB2 お客様サポートや開発部門がメッセージの生成された DB2 ソース・コード内のポイントを見つけられるようにするための、固有の内部 ID。
MSGNUM	INTEGER	エラーまたはイベントの数値のメッセージ番号。
MSGTYPE	CHAR(3)	メッセージ・タイプを示します。ADM (管理通知ログに書き込まれたメッセージの場合)、または NULL (メッセージ・タイプが判別できない場合)。
MSGSEVERITY	CHAR(1)	メッセージ重大度: C (重大)、E (エラー)、W (警告)、I (通知)、または NULL (メッセージ重大度が判別できない場合)。
MSG	CLOB(16K)	通知ログ・メッセージ・テキスト。
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## REORGCHK\_IX\_STATS プロシージャ - 再編成の評価用の索引統計の検索

REORGCHK\_IX\_STATS プロシージャは、再編成が必要かどうかを示す索引統計を含む、結果セットを戻します。

### 構文

```
▶▶ REORGCHK_IX_STATS (—scope—, —criteria—) ◀◀
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### scope

評価する表の有効範囲を指定するタイプ CHAR(1) の入力引数であり、以下のいずれかの値を使用します。

'T'

表

'S'

スキーマ

### criteria

タイプ VARCHAR(259) の入力引数。 *scope* の値が 'T' である場合、完全修飾表名を指定するか、ALL、USER、または SYSTEM のいずれかの値を受け入れます。 *scope* の値が 'S' である場合、スキーマ名を指定します。

### 許可

- カタログ表に対する SELECT 特権。
- REORGCHK\_IX\_STATS プロシージャに対する EXECUTE 特権。

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

```
CALL SYSPROC.REORGCHK_IX_STATS('T','JESCOTT.EMPLOYEE')
```

### 使用上の注意

プロシージャは、SYSTOOLSTMPSPACE 表スペースを使用します。SYSTOOLSTMPSPACE がまだ存在しない場合、このプロシージャは、この表スペースを作成します。

### 戻される情報

表 278. REORGCHK\_IX\_STATS プロシージャによって戻される情報

列名	データ・タイプ	説明
TABLE_SCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント
TABLE_NAME	VARCHAR(128)	table_name - 表名モニター・エレメント
INDEX_SCHEMA	VARCHAR(128)	index_schema - 索引スキーマ・モニター・エレメント
INDEX_NAME	VARCHAR(128)	index_name - 索引名モニター・エレメント
DATAPARTITIONNAME	VARCHAR(128)	データ・パーティションの名前。非パーティション表の場合は NULL。
INDCARD	BIGINT	索引内の索引項目の数。一部の索引の場合、これは、表のカーディナリティーとは異なることがあります。例えば、XML 列に対する索引カーディナリティーは、表のカーディナリティーより大きいことがあります。
NLEAF	BIGINT	nleaf - リーフ・ページ数モニター・エレメント
NUM_EMPTY_LEAFS	BIGINT	疑似的な空の索引リーフ・ページの数。

表 278. REORGCHK\_IX\_STATS プロシージャによって戻される情報 (続き)

列名	データ・タイプ	説明
NLEVELS	INTEGER	nlevels - 索引レベルの数モニター・エレメント
NUMRIDS_DELETED	BIGINT	疑似的な削除済み RID の数。
FULLKEYCARD	BIGINT	削除マークの付いていないユニーク索引項目の数。
LEAF_RECSIZE	BIGINT	リーフ・ページ上の索引項目のレコード・サイズ。これは、すべてのオーバーヘッドを除外した索引項目の平均サイズであり、索引に関与するすべての列から得た列の平均の長さから計算されます。
NONLEAF_RECSIZE	BIGINT	非リーフ・ページ上の索引項目のレコード・サイズ。これは、すべてのオーバーヘッドを除外した索引項目の平均サイズであり、索引に関与するすべての列 (INCLUDE 列を除く) から得た列の平均の長さから計算されます。
LEAF_PAGE_OVERHEAD	BIGINT	内部使用に備えて予約されている索引リーフ・ページ上のスペース。
NONLEAF_PAGE_OVERHEAD	BIGINT	内部使用に備えて索引の非リーフ・ページ上で予約されているスペース。
PCT_PAGES_SAVED	SMALLINT	索引圧縮を使用して節約されるページのパーセント。ゼロ以外の数値は、索引が圧縮されることを意味します。
F4	INTEGER	F4 公式値。
F5	INTEGER	F5 公式値。
F6	INTEGER	F6 公式値。
F7	INTEGER	F7 公式値。
F8	INTEGER	F8 公式値。
REORG	CHAR(5)	5 文字のフィールド。各文字は、F4、F5、F6、F7、および F8 の 5 つの公式のいずれかにマッピングされます。ダッシュが示される場合、公式値が推奨範囲に収まっていることを示します。アスタリスクが示される場合、公式値が推奨範囲から出ているので、再編成が必要であることを示します。

## REORGCHK\_TB\_STATS プロシージャ - 再編成の評価用の表統計の検索

REORGCHK\_TB\_STATS プロシージャは、再編成が必要かどうかを示す表統計を含む、結果セットを戻します。

### 構文

```
▶▶—REORGCHK_TB_STATS—(—scope—, —criteria—)————▶▶
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### scope

評価する表の有効範囲を指定するタイプ CHAR(1) の入力引数であり、以下のいずれかの値を使用します。

'T'

表

'S'

スキーマ

#### criteria

タイプ VARCHAR(259) の入力引数。scope の値が 'T' である場合、完全修飾表名を指定するか、ALL、USER、または SYSTEM のいずれかの値を受け入れます。scope の値が 'S' である場合、スキーマ名を指定します。

### 許可

- カタログ表に対する SELECT 特権。
- REORGCHK\_TB\_STATS プロシージャに対する EXECUTE 特権。

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

```
CALL SYSPROC.REORGCHK_TB_STATS('T','JESCOTT.EMPLOYEE')
```

### 使用上の注意

プロシージャは、SYSTOOLSTMPSPACE 表スペースを使用します。SYSTOOLSTMPSPACE がまだ存在しない場合、このプロシージャは、この表スペースを作成します。

### 戻される情報

表 279. REORGCHK\_TB\_STATS プロシージャによって戻される情報

列名	データ・タイプ	説明
TABLE_SCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント

表 279. REORGCHK\_TB\_STATS プロシージャによって戻される情報 (続き)

列名	データ・タイプ	説明
TABLE_NAME	VARCHAR(128)	table_name - 表名モニター・エレメント
DATAPARTITIONNAME	VARCHAR(128)	データ・パーティションの名前。非パーティション表の場合は NULL。
CARD	BIGINT	カーディナリティー (表の中の行数)。
OVERFLOW	BIGINT	オーバーフローした行数。
NPAGES	BIGINT	表の行が存在しているページの総数。ビューまたは別名の場合、または統計が収集されていない場合は -1。副表および階層表の場合は -2。
FPAGES	BIGINT	ページの総数。ビューまたは別名の場合、または統計が収集されていない場合は -1。副表および階層表の場合は -2。
ACTIVE_BLOCKS	BIGINT	マルチディメンション・クラスタリング (MDC) 表または挿入時クラスタリング (ITC) 表のアクティブ・ブロックの合計数。このフィールドは、ORGANIZE BY 節を使用して定義された表に対してのみ適用できます。これは、データを収めた表のブロック数を示します。
TSIZE	BIGINT	表のサイズ。
F1	INTEGER	F1 公式値。
F2	INTEGER	F2 公式値。
F3	INTEGER	F3 公式値。
REORG	CHAR(3)	3 文字のフィールド。各文字は、F1、F2、および F3 の 3 つの公式のいずれかにマッピングされます。ダッシュが示される場合、公式値が推奨範囲に収まっていることを示し、アスタリスクが示される場合、公式値が推奨範囲から出ているので、再編成が必要なことを示します。

## SET\_MAINT\_MODE\_RECORD\_NO\_TEMPORALHISTORY プロシージャ - テンポラル履歴の記録の無効化

SET\_MAINT\_MODE\_RECORD\_NO\_TEMPORALHISTORY プロシージャは、システム期間テンポラル表のテンポラル履歴の記録を使用不可にし、特殊テンポラル列に値を指定できるように、内部オプションを設定します。

## 許可

以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限

## デフォルトの PUBLIC 特権

なし

## 構文

```
▶▶—SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY—(—)—————▶▶
```

スキーマは SYSPROC です。

## 使用上の注意

- システム期間テンポラル表に対するテンポラル履歴の記録を SET\_MAINT\_MODE\_RECORD\_NO\_TEMPORALHISTORY プロシージャを使用し、使用不可にした場合、その設定は現行セッションの期間中存続します。

## 例

システム期間テンポラル表のテンポラル履歴の記録を使用不可にし、特殊テンポラル列に値を指定できるように、内部オプションを設定します。

```
call SYSPROC.SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY()
```

## SQLERRM スカラー関数 - エラー・メッセージ情報の検索

SQLERRM スカラー関数には 2 つのバージョンがあります。1 つは、メッセージ・トークンの使用や言語選択などを含む、十分に柔軟性をもたせたメッセージ検索を提供します。もう 1 つは、SQLCODE のみを入力パラメーターとし、簡略メッセージを英語で戻します。

## SQLERRM スカラー関数

この SQLERRM スカラー関数は、メッセージ ID、ロケール、およびトークンの入力を取り、タイプ VARCHAR(32672) の簡略メッセージまたは詳細メッセージを、指定のロケールで戻します。入力ロケールをサーバーがサポートしていない場合、メッセージは英語で戻されます。

## 構文

```
▶▶—SQLERRM—(—msgid—,—tokens—,—token_delimiter—,—locale—,—shortmsg—)——▶▶  
▶—)—————▶▶
```

スキーマは SYSPROC です。

## スカラー関数パラメーター

### *msgid*

情報を検索するメッセージ番号を表す、タイプ VARCHAR(9) の入力引数。メッセージ番号は、アプリケーション戻りコードに、接頭部 'SQL'、'DBA'、または 'CLI' を付けたものです。例えば、'SQL551'、'CLI0001' などです。メッセージ番号は、SQLSTATE である場合もあります。例えば、'42829' などです。

### *tokens*

エラー・メッセージのトークン・リストを表す、タイプ VARCHAR(70) の入力引数。トークンのないメッセージもあります。このパラメーターが NULL の場合は、戻されるメッセージでトークンの置き換えはなされません。トークンの置き換えは、デフォルトの簡略メッセージを戻す場合にのみ生じます。詳細メッセージのオプションが選択されている場合、トークンの置き換えは生じません。

### *token\_delimiter*

トークンの区切り文字を表す、タイプ VARCHAR(1) の入力引数。この区切り文字は固有でなければならず、スカラー関数に渡されるトークンに含まれてはなりません。区切り文字が指定されない場合、使用されるデフォルトの区切り文字はセミコロンです。

### *locale*

ロケールを表すタイプ VARCHAR(33) の入力引数。その言語のエラー・メッセージを検索するためにサーバーに渡します。ロケールが指定されていない場合、またはサーバーがそのロケールをサポートしない場合、メッセージは英語で戻され、警告が戻されます。

### *shortmsg*

デフォルトの簡略メッセージの代わりに詳細メッセージを戻すかどうかを示すために使用される、タイプ INTEGER の入力引数。詳細メッセージを戻すには、この値は 0 または CAST(NULL as INTEGER) に設定しなければなりません。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: SQL0551N の英語の簡略メッセージを、トークン "AYYANG"、"UPDATE"、および "SYSCAT.TABLES" の含まれた状態で検索します。

```
VALUES (SYSPROC.SQLERRM
        ('SQL551', 'AYYANG;UPDATE;SYSCAT.TABLES', ';', 'en_US', 1))
```

以下は戻される出力の例です。

```
1
-----
SQL0551N "AYYANG" does not have the privilege to perform operation
"UPDATE" on object "SYSCAT.TABLES"
```

例 2: SQLSTATE 42501 に関連した英語のエラー・メッセージを検索します。

```
VALUES (SYSPROC.SQLERRM ('42501', '', '', 'en_US', 1))
```

以下は戻される出力の例です。

```
1
-----
SQLSTATE 42501: The authorization ID does not have the privilege to
perform the specified operation on the identified object.
```

例 3: SQL1001N の英語の長いエラー・メッセージを検索します。

```
VALUES (SYSPROC.SQLERRM ('SQL1001', '', '', 'en_US', 0))
```

以下は戻される出力の例です。

```
1
-----
SQL1001N "<name>" is not a valid database name.
```

Explanation:

The syntax of the database name specified in the command is not valid. The database name must contain 1 to 8 characters and all the characters must be from the database manager base character set.

The command cannot be processed.

User Response:

Resubmit the command with the correct database name.

sqlcode : -1001

sqlstate : 2E000

## SQLERRM スカラー関数

この SQLERRM スカラー関数は SQLCODE を唯一の入力データとして取り、指定の SQLCODE に対するタイプ VARCHAR(32672) の簡略メッセージを英語で戻します。

### 構文

```
►►SQLERRM(—sqlcode—)◄◄
```

スキーマは SYSPROC です。

### スカラー関数パラメーター

*sqlcode*

SQLCODE を表す、タイプ INTEGER の入力引数。



## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

SQLCODE SQL0551N の簡略メッセージを検索します。

```
VALUES (SYSPROC.SQLERRM (551))
```

以下は戻される出力の例です。

```
1
-----...--
SQL0551N  "" does not have the privilege to perform operation
         "" on object "".
```

## SYSINSTALLOBJECTS

SYSINSTALLOBJECTS プロシージャは、特定のツールで必要なデータベース・オブジェクトを作成またはドロップします。

## 構文

```
▶▶—SYSINSTALLOBJECTS—(—tool-name—,—action—,—tablespace-name—,——————▶
▶—schema-name—)—————▶▶
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

*tool-name*

以下のいずれかの値を使用して、ロードされるツールの名前を指定する、タイプ VARCHAR(128) の入力引数。

- 'DB2AC' (自立走行式コンピューティングの場合 (ヘルス・モニター))
- 'STMG\_DBSIZE\_INFO' (ストレージ管理の場合)
- 'OPT\_PROFILES' (最適化プロファイル表を作成する場合)
- 'POLICY' (ポリシーの場合 (表およびトリガー))
- 'EXPLAIN' (EXPLAIN 表を作成またはマイグレーションする場合)
- 'INGEST' (INGEST ユーティリティで使用する再始動表を作成する場合)
- 'ASP' (統計プロファイルを自動的に生成する場合)

**重要:** ASP 値は、推奨されていません。バージョン 10.1 の自動統計プロファイルは推奨されておらず、今後のリリースで除去される可能性があります。詳しくは、「DB2 バージョン 10.1 の新機能」の『自動統計プロファイルは非推奨』を参照してください。

#### *action*

実行予定のアクションを指定する、タイプ CHAR(1) 入力引数。有効な値は以下のとおりです。

- C** オブジェクトを作成します。
- D** オブジェクトをドロップします。
- V** オブジェクトを検証します。
- M** オブジェクトをマイグレーションします。M オプションは、ツール名 EXPLAIN と共に使用する場合にのみ有効です。このオプションは、バージョン 7 以降で作成された Explain 表をマイグレーションして、現行バージョンとの互換性を持たせます。

#### *tablespace-name*

オブジェクトを作成するときの表スペースの名前を指定する、タイプ VARCHAR(128) の入力引数。値を指定しないか、値が空またはブランク・ストリングである場合には、ツール名が AM であればデフォルトのユーザー・スペースが使用されます。ツール名が EXPLAIN でアクションが M である場合には、入力表スペース名は無視されて、マイグレーション対象の Explain 表が作成された表スペースが使用されます。それ以外の場合は、SYSTOOLSPACE 表スペースが使用されます。SYSTOOLSPACE がまだ存在しない場合は、作成されます。

#### *schema-name*

'EXPLAIN' tool-name オプションを除く、入力パラメーターとして渡される schema-name に関係なく、SYSTOOLS が必ずスキーマとして使用されます。

'EXPLAIN' tool-name オプションに対しては、入力の schema-name を渡すことができ、指定されたその schema-name で表が作成されます。schema-name が入力パラメーターとして渡されない場合は、SYSTOOLS スキーマが使用されます。

## 許可

このプロシージャを実行するには、以下のいずれかの権限が必要です。

- プロシージャでの EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべての Explain 表をマイグレーションする。

```
CALL SYSPROC.SYSINSTALLOBJECTS('EXPLAIN', 'M', CAST (NULL AS VARCHAR(128)),  
CAST (NULL AS VARCHAR(128)))
```

## 使用すべきでない SQL 管理ルーチンおよびビュー

既存の管理ルーチンと管理ビューを新しいより包括的なルーチンとビューに置き換えることで、拡張サポートが提供されます。

バージョン 10.1 以降、名前にバージョンの接尾部があるルーチンは使用すべきでなくなりました。ルーチン名にバージョンの接尾部がなくなったので、リリース間で名前の一貫性が保たれます。置き換え後のバージョン接尾部がないルーチンには、列の追加や除去、既存の列の新規データ・タイプ、既存の列の新規値などの変更が加えられている可能性があります。SQL 管理ルーチンで照会を発行する際には、これらのルーチンに対する変更の影響が最小限になる方法で行ってください。

次の表には、現在使用すべきでない SQL 管理ルーチンや管理ビューがすべてリストされています。

表 280. DB2 バージョン 10.1 の使用すべきでない SQL 管理ルーチンまたはビュー、およびそれらに代わるルーチンまたはビュー

使用すべきでないルーチンまたはビュー	初めて使用すべきでなくなったバージョン	新しいルーチンまたはビュー	置き換え後のものが初めて使用可能になったバージョン
1259 ページの『ADMIN_GET_DBP_MEM_USAGE 表関数 - インスタンスの合計メモリー消費量の取得』	バージョン 10.1	251 ページの『ADMIN_GET_MEM_USAGE 表関数 - インスタンスの合計メモリー消費量の取得』	バージョン 10.1
1261 ページの『ADMINTABCOMPRESSINFO 管理ビューおよび ADMIN_GET_TAB_COMPRESS_INFO 表関数 (使用すべきでない) - 圧縮された情報を戻す』	バージョン 10.1	256 ページの『ADMIN_GET_TAB_COMPRESS_INFO 表関数 - 圧縮による節約の見積もり』 258 ページの『ADMIN_GET_TAB_DICTIONARY_INFO 表関数 - 既存の表ディクショナリーのプロパティの報告』	バージョン 10.1
1266 ページの『ADMIN_GET_TAB_COMPRESS_INFO_V97』	バージョン 10.1	256 ページの『ADMIN_GET_TAB_COMPRESS_INFO 表関数 - 圧縮による節約の見積もり』 258 ページの『ADMIN_GET_TAB_DICTIONARY_INFO 表関数 - 既存の表ディクショナリーのプロパティの報告』	バージョン 10.1
1273 ページの『ADMIN_GET_TAB_INFO_V95 表関数 - 表のサイズおよび状態に関する情報の検索』	バージョン 10.1	296 ページの『ADMIN_GET_TAB_INFO 表関数』	バージョン 10.1
1280 ページの『ADMIN_GET_TAB_INFO_V97 表関数 - 表のサイズおよび状態に関する情報の取得』	バージョン 10.1	296 ページの『ADMIN_GET_TAB_INFO 表関数』	バージョン 10.1
1288 ページの『AM_BASE_RPT_RECOMS - アクティビティ・レポートに関する推奨事項』	バージョン 10.1		バージョン 10.1

表 280. DB2 パージョン 10.1 の使用すべきでない SQL 管理ルーチンまたはビュー、およびそれらに代わるルーチンまたはビュー (続き)

使用すべきでないルーチンまたはビュー	初めて使用すべきでなくなったバージョン	新しいルーチンまたはビュー	置き換え後のものが初めて使用可能になったバージョン
1290 ページの『AM_BASE_RPTS - アクティビティ・イベント・モニター・レポート』	バージョン 10.1		バージョン 10.1
1291 ページの『AM_DROP_TASK - モニター・タスクの削除』	バージョン 10.1		バージョン 10.1
1292 ページの『AM_GET_LOCK_CHN_TB - 表形式のアプリケーション・ロック・チェーン・データの検索』	バージョン 10.1		バージョン 10.1
1293 ページの『AM_GET_LOCK_CHNS - 特定のアプリケーションに関するロック・チェーン情報の検索』	バージョン 10.1		バージョン 10.1
1294 ページの『AM_GET_LOCK_RPT - アプリケーション・ロックに関する詳細の検索』	バージョン 10.1		バージョン 10.1
1302 ページの『AM_GET_RPT - アクティビティ・モニター・データの検索』	バージョン 10.1		バージョン 10.1
1304 ページの『AM_SAVE_TASK - モニター・タスクの作成または変更』	バージョン 10.1		バージョン 10.1
1305 ページの『APPLICATION_ID』	バージョン 10.1	531 ページの『MON_GET_APPLICATION_ID - 接続アプリケーション ID の取得』	バージョン 10.1
1368 ページの『REG_VARIABLES 管理ビュー - 使用中の DB2 レジストリー設定の検索』	バージョン 10.1	410 ページの『ENV_GET_REG_VARIABLES 表関数 - 使用中の DB2 レジストリー設定の取得』	バージョン 10.1
1370 ページの『SNAPAGENT_MEMORY_POOL 管理ビュー および SNAP_GET_AGENT_MEMORY_POOL 表関数 - memory_pool 論理データ・グループのスナップショット情報の検索』	バージョン 10.1	604 ページの『MON_GET_MEMORY_POOL - メモリー・プール情報の取得』および 607 ページの『MON_GET_MEMORY_SET - メモリー・セット情報の取得』	バージョン 10.1
1375 ページの『SNAP_GET_APPL_INFO_V95 表関数 - appl_info 論理データ・グループのスナップショット情報の検索』	バージョン 10.1	846 ページの『SNAP_GET_APPL_INFO 表関数』	バージョン 10.1
1382 ページの『SNAP_GET_APPL_V95 表関数 - appl 論理データ・グループのスナップショット情報の検索』	バージョン 10.1	854 ページの『SNAP_GET_APPL 表関数』	バージョン 10.1
1390 ページの『SNAP_GET_BP_V95 表関数 - bufferpool 論理グループのスナップショット情報の検索』	バージョン 10.1	864 ページの『SNAP_GET_BP 表関数』	バージョン 10.1
1394 ページの『SNAP_GET_CONTAINER_V91 表関数 - tablespace_container 論理データ・グループ・スナップショット情報の検索』	バージョン 10.1	574 ページの『MON_GET_CONTAINER 表関数 - 表スペース・コンテナ・メトリックの取得』	バージョン 10.1
1409 ページの『SNAP_GET_DB_V97 表関数 - dbase 論理グループからのスナップショット情報の検索』	バージョン 10.1	879 ページの『SNAP_GET_DB 表関数』	バージョン 10.1
1402 ページの『SNAP_GET_DBM_V95 表関数 - dbm 論理グループ・スナップショット情報の検索』	バージョン 10.1	891 ページの『SNAP_GET_DBM 表関数』	バージョン 10.1
1420 ページの『SNAP_GET_DETAILLOG_V91 表関数 - detail_log 論理データ・グループからのスナップショット情報の取得』	バージョン 10.1	MON_GET_TRANSACTION_LOG 表関数	バージョン 10.1

表 280. DB2 バージョン 10.1 の使用すべきでない SQL 管理ルーチンまたはビュー、およびそれらに代わるルーチンまたはビュー (続き)

使用すべきでないルーチンまたはビュー	初めて使用すべきでなくなったバージョン	新しいルーチンまたはビュー	置き換え後のものが初めて使用可能になったバージョン
1422 ページの『SNAP_GET_DYN_SQL_V95 表関数 - dynsql 論理グループのスナップショット情報の検索』	バージョン 10.1	900 ページの『SNAP_GET_DYN_SQL 表関数』	バージョン 10.1
1447 ページの『SNAPSTORAGE_PATHS 管理ビューおよび SNAP_GET_STORAGE_PATHS_V97 表関数 - 自動ストレージ・パスの情報の検索』	バージョン 10.1	254 ページの『ADMIN_GET_STORAGE_PATHS 表関数 - 自動ストレージ・パス情報の取得』	バージョン 10.1
1454 ページの『SNAP_GET_TAB_V91』	バージョン 10.1	693 ページの『MON_GET_TABLE 表関数 - 表メトリックの取得』	バージョン 10.1
1457 ページの『SNAP_GET_TBSP_PART_V97 表関数 - tablespace_nodeinfo 論理データ・グループのスナップショット情報の検索』	バージョン 10.1	697 ページの『MON_GET_TABLESPACE 表関数 - 表スペース・メトリックの取得』	バージョン 10.1
1461 ページの『SNAP_GET_TBSP_V91』	バージョン 10.1	697 ページの『MON_GET_TABLESPACE 表関数 - 表スペース・メトリックの取得』	バージョン 10.1
1426 ページの『SNAPHADR 管理ビューおよび SNAP_GET_HADR 表関数 - hadr 論理データ・グループのスナップショット情報の検索』	バージョン 10.1	584 ページの『MON_GET_HADR 表関数 - 高可用性災害時リカバリー (HADR) モニター情報を戻す』	バージョン 10.1
1564 ページの『WLM_GET_SERVICE_CLASS_AGENTS_V97 - サービス・クラスで実行中のエージェントのリスト』	バージョン 10.1	1169 ページの『WLM_GET_SERVICE_CLASS_AGENTS 表関数 - サービス・クラスで実行中のエージェントのリスト』	バージョン 10.1
1573 ページの『WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES_V97 - ワークロード・オカレンスのリスト』	バージョン 10.1	1177 ページの『WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES - ワークロード・オカレンスのリスト』	バージョン 10.1
1577 ページの『WLM_GET_SERVICE_SUBCLASS_STATS_V97 - サービス・サブクラスの統計を戻す』	バージョン 10.1	1181 ページの『WLM_GET_SERVICE_SUBCLASS_STATS 表関数 - サービス・サブクラスの統計を戻す』	バージョン 10.1
1583 ページの『WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97 - アクティビティのリストを戻す』	バージョン 10.1	1191 ページの『WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES - アクティビティのリストを戻す』	バージョン 10.1
1588 ページの『WLM_GET_WORKLOAD_STATS_V97 - ワークロード統計を戻す』	バージョン 10.1	1196 ページの『WLM_GET_WORKLOAD_STATS 表関数 - ワークロード統計を戻す』	バージョン 10.1
1306 ページの『DB_PARTITIONS』	バージョン 9.8	400 ページの『DB2_MEMBER および DB2_CF 管理ビューと DB2_GET_INSTANCE_INFO 表関数』	バージョン 9.8
1397 ページの『SNAPDB_MEMORY_POOL 管理ビューおよび SNAP_GET_DB_MEMORY_POOL 表関数 - データベース・レベルのメモリー使用量情報の検索』	バージョン 9.7	607 ページの『MON_GET_MEMORY_SET - メモリー・セット情報の取得』 604 ページの『MON_GET_MEMORY_POOL - メモリー・プール情報の取得』	バージョン 9.7 フィックスバック 5
1405 ページの『SNAPDBM_MEMORY_POOL 管理ビューおよび SNAP_GET_DBM_MEMORY_POOL 表関数 - データベース・マネージャー・レベルのメモリー使用量情報の検索』	バージョン 9.7	607 ページの『MON_GET_MEMORY_SET - メモリー・セット情報の取得』 604 ページの『MON_GET_MEMORY_POOL - メモリー・プール情報の取得』	バージョン 9.7 フィックスバック 5

表 280. DB2 バージョン 10.1 の使用すべきでない SQL 管理ルーチンまたはビュー、およびそれらに代わるルーチンまたはビュー (続き)

使用すべきでないルーチンまたはビュー	初めて使用すべきでなくなったバージョン	新しいルーチンまたはビュー	置き換え後のものが初めて使用可能になったバージョン
1313 ページの『LOCKS_HELD 管理ビュー - 保持されているロックに関する情報の検索』	バージョン 9.7	526 ページの『MON_GET_APPL_LOCKWAIT - アプリケーションが待機しているロックについての情報の取得』 598 ページの『MON_GET_LOCKS - 現在接続されているデータベース内のすべてのロックのリスト』 482 ページの『MON_FORMAT_LOCK_NAME - 内部ロック名のフォーマット設定と詳細の出力』 758 ページの『MON_LOCKWAITS 管理ビュー - ロックの取得を待機しているアプリケーションに関するメトリックの取得』	バージョン 9.7 フィックスパック 1
1316 ページの『LOCKWAITS 管理ビュー - 現行のロック待機情報の検索』	バージョン 9.7	526 ページの『MON_GET_APPL_LOCKWAIT - アプリケーションが待機しているロックについての情報の取得』 598 ページの『MON_GET_LOCKS - 現在接続されているデータベース内のすべてのロックのリスト』 482 ページの『MON_FORMAT_LOCK_NAME - 内部ロック名のフォーマット設定と詳細の出力』 758 ページの『MON_LOCKWAITS 管理ビュー - ロックの取得を待機しているアプリケーションに関するメトリックの取得』	バージョン 9.7 フィックスパック 1
1431 ページの『SNAPLOCK 管理ビューおよび SNAP_GET_LOCK 表関数 - lock 論理データ・グループのスナップショット情報の検索』	バージョン 9.7	526 ページの『MON_GET_APPL_LOCKWAIT - アプリケーションが待機しているロックについての情報の取得』 598 ページの『MON_GET_LOCKS - 現在接続されているデータベース内のすべてのロックのリスト』 482 ページの『MON_FORMAT_LOCK_NAME - 内部ロック名のフォーマット設定と詳細の出力』 758 ページの『MON_LOCKWAITS 管理ビュー - ロックの取得を待機しているアプリケーションに関するメトリックの取得』	バージョン 9.7 フィックスパック 1
1438 ページの『SNAPLOCKWAIT 管理ビューおよび SNAP_GET_LOCKWAIT 表関数 - lockwait 論理データ・グループのスナップショット情報の検索』	バージョン 9.7	526 ページの『MON_GET_APPL_LOCKWAIT - アプリケーションが待機しているロックについての情報の取得』 598 ページの『MON_GET_LOCKS - 現在接続されているデータベース内のすべてのロックのリスト』 482 ページの『MON_FORMAT_LOCK_NAME - 内部ロック名のフォーマット設定と詳細の出力』 758 ページの『MON_LOCKWAITS 管理ビュー - ロックの取得を待機しているアプリケーションに関するメトリックの取得』	バージョン 9.7 フィックスパック 1

表 280. DB2 バージョン 10.1 の使用すべきでない SQL 管理ルーチンまたはビュー、およびそれらに代わるルーチンまたはビュー (続き)

使用すべきでないルーチンまたはビュー	初めて使用すべきでなくなったバージョン	新しいルーチンまたはビュー	置き換え後のものが初めて使用可能になったバージョン
1558 ページの『WLM_GET_ACTIVITY_DETAILS - 特定のアクティビティに関する詳細情報を戻す』	バージョン 9.5	512 ページの『MON_GET_ACTIVITY_DETAILS 表関数 - 完全なアクティビティ詳細の取得』	バージョン 9.7
1307 ページの『GET_DB_CONFIG』	バージョン 9.1	388 ページの『DBCFCG 管理ビューおよび DB_GET_CFG 表関数 - データベース構成パラメーター情報の取得』	バージョン 9.1
1309 ページの『GET_DBM_CONFIG』	バージョン 9.1	392 ページの『DBMCFG 管理ビュー - データベース・マネージャー構成パラメーター情報の検索』	バージョン 9.1
1445 ページの『SNAP_GET_STO_PATHS』	バージョン 9.1	254 ページの『ADMIN_GET_STORAGE_PATHS 表関数 - 自動ストレージ・パス情報の取得』	バージョン 10.1
1498 ページの『SNAPSHOT_AGENT』	バージョン 9.1	843 ページの『SNAP_GET_AGENT 表関数』	バージョン 9.1
1500 ページの『SNAPSHOT_APPL』	バージョン 9.1	854 ページの『SNAP_GET_APPL 表関数』	バージョン 10.1
1505 ページの『SNAPSHOT_APPL_INFO』	バージョン 9.1	846 ページの『SNAP_GET_APPL_INFO 表関数』	バージョン 10.1
1508 ページの『SNAPSHOT_BP』	バージョン 9.1	864 ページの『SNAP_GET_BP 表関数』	バージョン 10.1
1511 ページの『SNAPSHOT_CONTAINER』	バージョン 9.1	874 ページの『SNAP_GET_CONTAINER 表関数』	バージョン 10.1
1512 ページの『SNAPSHOT_DATABASE』	バージョン 9.1	879 ページの『SNAP_GET_DB 表関数』	バージョン 10.1
1519 ページの『SNAPSHOT_DBM』	バージョン 9.1	891 ページの『SNAP_GET_DBM 表関数』	バージョン 10.1
1521 ページの『SNAPSHOT_DYN_SQL』	バージョン 9.1	900 ページの『SNAP_GET_DYN_SQL 表関数』	バージョン 10.1
1523 ページの『SNAPSHOT_FCM』	バージョン 9.1	906 ページの『SNAP_GET_FCM 表関数』	バージョン 9.1
1525 ページの『SNAPSHOT_FCMNODE』	バージョン 9.1	909 ページの『SNAP_GET_FCM_PART 表関数』	バージョン 9.1
1526 ページの『SNAPSHOT_FILEW』	バージョン 9.1	964 ページの『SNAP_WRITE_FILE プロシージャ』	バージョン 9.1
1527 ページの『SNAPSHOT_LOCK』	バージョン 9.1	526 ページの『MON_GET_APPL_LOCKWAIT - アプリケーションが待機しているロックについての情報の取得』 598 ページの『MON_GET_LOCKS - 現在接続されているデータベース内のすべてのロックのリスト』 482 ページの『MON_FORMAT_LOCK_NAME - 内部ロック名のフォーマット設定と詳細の出力』	バージョン 9.7 フィックスパック 1

表 280. DB2 バージョン 10.1 の使用すべきでない SQL 管理ルーチンまたはビュー、およびそれらに代わるルーチンまたはビュー (続き)

使用すべきでないルーチンまたはビュー	初めて使用すべきでなくなったバージョン	新しいルーチンまたはビュー	置き換え後のものが初めて使用可能になったバージョン
1529 ページの『SNAPSHOT_LOCKWAIT』	バージョン 9.1	526 ページの『MON_GET_APPL_LOCKWAIT - アプリケーションが待機しているロックについての情報の取得』 598 ページの『MON_GET_LOCKS - 現在接続されているデータベース内のすべてのロックのリスト』 482 ページの『MON_FORMAT_LOCK_NAME - 内部ロック名のフォーマット設定と詳細の出力』	バージョン 9.7 フィックスパック 1
1530 ページの『SNAPSHOT QUIESCERS』	バージョン 9.1	949 ページの『SNAP_GET_TBSP QUIESCER 表関数』	バージョン 9.1
1532 ページの『SNAPSHOT_RANGES』	バージョン 9.1	953 ページの『SNAP_GET_TBSP_RANGE 表関数』	バージョン 9.1
1533 ページの『SNAPSHOT_STATEMENT』	バージョン 9.1	913 ページの『SNAP_GET_STMT 表関数』	バージョン 9.1
1536 ページの『SNAPSHOT_SUBSECT』	バージョン 9.1	920 ページの『SNAP_GET_SUBSECTION 表関数』	バージョン 9.1
1538 ページの『SNAPSHOT_SWITCHES』	バージョン 9.1	925 ページの『SNAP_GET_SWITCHES 表関数』	バージョン 9.1
1540 ページの『SNAPSHOT_TABLE』	バージョン 9.1	1451 ページの『SNAP_GET_TAB 表関数』	バージョン 9.1
1541 ページの『SNAPSHOT_TBREORG』	バージョン 9.1	929 ページの『SNAP_GET_TAB_REORG 表関数』	バージョン 9.1
1543 ページの『SNAPSHOT_TBS』	バージョン 9.1	936 ページの『SNAP_GET_TBSP 表関数』	バージョン 9.1
1546 ページの『SNAPSHOT_TBS_CFG』	バージョン 9.1	942 ページの『SNAP_GET_TBSP_PART 表関数』	バージョン 10.1
1555 ページの『SQLCACHE_SNAPSHOT』	バージョン 9.1	900 ページの『SNAP_GET_DYN_SQL 表関数』	バージョン 10.1
1557 ページの『SYSINSTALLROUTINES』	バージョン 9.1	使用不可	

DB2 for Linux, UNIX, and Windows バージョン 9.7 では、ヘルス・モニターは使用すべきでなくなりました。ヘルス・モニター・ルーチンも使用すべきでなく、将来のリリースで除去される可能性があります。DB2 for Linux, UNIX, and Windows のデータおよびデータ中心アプリケーションの管理用に新しい GUI ツールが一式用意されており、廃止されたコントロール・センター・ツールの代わりにこれらを使用できます。詳しくは、データベース管理およびアプリケーション開発ツールを参照してください。

以下のセクションは、使用すべきでないヘルス・モニター・ルーチンのリストです。

- 1320 ページの『HEALTH\_CONT\_HI』
- 1322 ページの『HEALTH\_CONT\_HI\_HIS』
- 1324 ページの『HEALTH\_CONT\_INFO』
- 1326 ページの『HEALTH\_DB\_HI』
- 1330 ページの『HEALTH\_DB\_HI\_HIS』



- 1334 ページの『HEALTH\_DB\_HIC』
- 1336 ページの『HEALTH\_DB\_HIC\_HIS』
- 1339 ページの『HEALTH\_DB\_INFO』
- 1341 ページの『HEALTH\_DBM\_HI』
- 1343 ページの『HEALTH\_DBM\_HI\_HIS』
- 1345 ページの『HEALTH\_DBM\_INFO』
- 1347 ページの『HEALTH\_GET\_ALERT\_ACTION\_CFG』
- 1351 ページの『HEALTH\_GET\_ALERT\_CFG』
- 1355 ページの『HEALTH\_GET\_IND\_DEFINITION』
- 1357 ページの『HEALTH\_HI\_REC』
- 1359 ページの『HEALTH\_TBS\_HI』
- 1362 ページの『HEALTH\_TBS\_HI\_HIS』
- 1366 ページの『HEALTH\_TBS\_INFO』

## ADMIN\_GET\_DBP\_MEM\_USAGE 表関数 - インスタンスの合計メモリー消費量の取得

ADMIN\_GET\_DBP\_MEM\_USAGE 表関数は、指定されたインスタンスの合計メモリー消費量を取得します。

注: この表関数は使用すべきではなく、251 ページの『ADMIN\_GET\_MEM\_USAGE 表関数 - インスタンスの合計メモリー消費量の取得』に置き換えられました。

ADMIN\_GET\_DBP\_MEM\_USAGE 表関数は、オプション入力引数 *member* (INTEGER タイプ) を取ります。これは、有効なデータベース・メンバー番号を指定し、その単一データベース・メンバーの統計のみを戻します。引数が省略される場合、すべてのアクティブなデータベース・メンバーの統計が戻されます。複数メンバー環境では、*member* に -1 または NULL 値を指定すると、現在接続されているメンバーからデータが戻されます。

### 構文

```
▶▶ ADMIN_GET_DBP_MEM_USAGE ( [member] ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

*member*

メモリー使用量統計が取得されるデータベース・メンバーを指定する、タイプ integer のオプション入力引数。-1 または NULL 値が指定される場合、データは現在接続中のメンバーから戻されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権

- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 戻される情報

表 281. ADMIN\_GET\_DBP\_MEM\_USAGE の結果セット

列名	データ・タイプ	説明
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MAX_PARTITION_MEM	BIGINT	インスタンス・メモリー限度が実施される場合にデータベース・パーティションで消費することが許可されるインスタンス・メモリーの最大量 (バイト)。
CURRENT_PARTITION_MEM	BIGINT	データベース・パーティションで現在消費されているインスタンス・メモリーの量 (バイト)。
PEAK_PARTITION_MEM	BIGINT	データベース・パーティション内のインスタンス・メモリーの消費量のピーク水準点または最高水準点 (バイト)。

## 例

例 1: メモリー使用量統計をデータベース・パーティション 3 から取得します。

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_DBP_MEM_USAGE(3)) AS T
```

DBPARTITIONNUM	MAX_PARTITION_MEM	CURRENT_PARTITION_MEM	PEAK_PARTITION_MEM
3	500000000	381000000	481000000

1 record(s) selected.

例 2: 現在接続中のメンバーからメモリー使用量統計を取得します (ユーザーがメンバー 2 でデータベースに接続されていると想定)。

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_DBP_MEM_USAGE(-1)) AS T
```

DBPARTITIONNUM	MAX_PARTITION_MEM	CURRENT_PARTITION_MEM	PEAK_PARTITION_MEM
2	500000000	381000000	481000000

1 record(s) selected.

例 3: メモリー使用量統計をすべてのメンバーから取得します。

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_DBP_MEM_USAGE()) AS T
```

DBPARTITIONNUM	MAX_PARTITION_MEM	CURRENT_PARTITION_MEM	PEAK_PARTITION_MEM
0	500000000	381000000	481000000
1	500000000	381000000	481000000

2	500000000	381000000	481000000
3	500000000	381000000	481000000

4 record(s) selected.

例 4: メモリー使用量統計をメガバイト (MB) 単位の値で取得します。

```
SELECT DBPARTITIONNUM, MAX_PARTITION_MEM/1048576 AS MAX_MEM_MB,
       CURRENT_PARTITION_MEM/1048576 AS CURRENT_MEM_MB, PEAK_PARTITION_MEM/1048576
       AS PEAK_MEM_MB FROM TABLE (SYSPROC.ADMIN_GET_DBP_MEM_USAGE()) AS T
```

DBPARTITIONNUM	MAX_MEM_MB	CURRENT_MEM_MB	PEAK_MEM_MB
0	4590	1107	1107
1	4590	1108	1108
2	4590	1106	1106

3 record(s) selected.

## ADMINTABCOMPRESSINFO 管理ビューおよび ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数 (使用すべきでない) - 圧縮された情報を戻す

ADMINTABCOMPRESSINFO 管理ビューおよび  
ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数は、表、マテリアライズ照会表 (MQT)、階層表に関する圧縮された情報を戻します。

注: この管理ビューおよび関連する表関数は非推奨になり、256 ページの

『ADMIN\_GET\_TAB\_COMPRESS\_INFO  
表関数 - 圧縮による節約の見積もり』および 258 ページの

『ADMIN\_GET\_TAB\_DICTIONARY\_INFO  
表関数 - 既存の表ディクショナリーのプロパティの報告』に置き換えられました。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『ADMINTABCOMPRESSINFO 管理ビュー』
- 1263 ページの『ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数』

### ADMINTABCOMPRESSINFO 管理ビュー

ADMINTABCOMPRESSINFO 管理ビューは、表、マテリアライズ照会表 (MQT)、および階層表だけの圧縮された情報を戻します。これらの表タイプは SYSCAT.TABLES カタログ・ビューで、T (表)、S (マテリアライズ照会表)、および H (階層表) として報告されます。情報は表のデータ・パーティション・レベルとデータベース・パーティション・レベルの両方のものが戻されます。

スキーマは SYSIBMADM です。

戻すことができる情報の全リストについては、ADMINTABCOMPRESSINFO 管理ビューおよび ADMIN\_GET\_TAB\_COMPRESS\_INFO によって戻される情報の表を参照してください。

## 許可

以下のいずれかの権限が必要です。

- ADMINTABCOMPRESSINFO 管理ビューに対する SELECT 特権
- ADMINTABCOMPRESSINFO 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

さらに、以下のいずれかの特権または権限も必要です。

- ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数に対する EXECUTE 特権
- DATAACCESS 権限

## デフォルトの PUBLIC 特権

なし

## 例

例 1: すべての表に関するすべての圧縮された情報を検索します。

```
SELECT * FROM SYSIBMADM.ADMINTABCOMPRESSINFO
```

以下はこの照会の出力例です。

TABSCHEMA	TABNAME	DBPARTITIONNUM	DATA_PARTITION_ID	COMPRESS_ATTR	DICT_BUILDER	DICT_BUILD_TIMESTAMP
.....	.....	.....	.....	.....	.....	.....
SYSIBM	SYSTABLES	0	0	N	NOT BUILT	-
SYSIBM	SYSCOLUMNS	0	0	N	NOT BUILT	-
...						
SIMAP2	STAFF	0	0	Y	REORG	2006-08-27-19.07.36.000000
SIMAP2	PARTTAB	0	0	Y	REORG	2006-08-27-22.07.17.000000
...						

156 record(s) selected.

この照会からの出力 (続き):

COMPRESS_DICT_SIZE	EXPAND_DICT_SIZE	ROWS_SAMPLED	PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH
.....	.....	.....	.....	.....	.....
0	0	0	0	0	0
0	0	0	0	0	0
...					
13312	5312	35	65	84	100
5760	4248	45	76	79	98
...					

例 2: すべての表に関するディクショナリー作成アクションおよびディクショナリー作成時間を決定します。

```
SELECT TABSCHEMA, TABNAME, DBPARTITIONNUM, DATA_PARTITION_ID, DICT_BUILDER, DICT_BUILD_TIMESTAMP  
FROM SYSIBMADM.ADMINTABCOMPRESSINFO
```

以下はこの照会の出力例です。

TABSCHEMA	TABNAME	DBPARTITIONNUM	DATA_PARTITION_ID	DICT_BUILDER	DICT_BUILD_TIMESTAMP
.....	.....	.....	.....	.....	.....
SYSIBM	SYSTABLES	0	0	NOT BUILT	-
SYSIBM	SYSCOLUMNS	0	0	NOT BUILT	-
...					
SIMAP2	STAFF	0	0	REORG	2006-08-27-19.07.36.000000
SIMAP2	SALES	0	0	NOT BUILT	-
SIMAP2	CATALOG	0	0	NOT BUILT	-
...					

156 record(s) selected.

## ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数

ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数は ADMINTABCOMPRESSINFO 管理ビューと同じ情報を戻しますが、この関数ではスキーマ、表名、および実行モードを指定することが可能です。

戻ることができる情報の全リストについては、ADMINTABCOMPRESSINFO 管理ビューおよび ADMIN\_GET\_TAB\_COMPRESS\_INFO によって戻される情報の表を参照してください。

### 構文

```
▶▶—ADMIN_GET_TAB_COMPRESS_INFO—(—tabschema—,—tabname—,—execmode—)————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *tabschema*

スキーマ名を指定する、タイプ VARCHAR(128) の入力引数。

#### *tabname*

表、マテリアライズ照会表、または階層表それぞれの名前を指定する、タイプ VARCHAR(128) の入力引数。

#### *execmode*

実行モードを指定する、タイプ VARCHAR(30) の入力引数。実行モードは以下のいずれかのモードになります。

- 'REPORT'-- 最後の生成の時点での圧縮情報を報告します。これはデフォルト値です。
- 'ESTIMATE'-- 現在の表に基づいて新しい圧縮情報を生成します。

### 許可

ADMIN\_GET\_TAB\_COMPRESS\_INFO 関数に対する EXECUTE 特権。

### デフォルトの PUBLIC 特権

なし

### 例

例 1: 表 SIMAP2.STAFF に関する既存の圧縮情報を検索します。

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO('SIMAP2', 'STAFF', 'REPORT'))
AS T
```

以下はこの照会の出力からの例です。

TABSCHEMA	TABNAME	DBPARTITIONNUM	DATA_PARTITION_ID	COMPRESS_ATTR	DICT_BUILDER	DICT_BUILD_TIMESTAMP
SIMAP2	STAFF	0	0 Y	REORG		2006-08-27-19.07.36.000000

1 record(s) selected.

この照会からの出力 (続き):

COMPRESS_DICT_SIZE	EXPAND_DICT_SIZE	ROWS_SAMPLED	PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH
13312	5312	35		65	84
					100

例 2: 現時点での表 SIMAP2.STAFF に関する推定圧縮圧縮情報を検索します。

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO('SIMAP2', 'STAFF', 'ESTIMATE')) AS T
```

以下はこの照会の出力からの例です。

TABSCHEMA	TABNAME	DBPARTITIONNUM	DATA_PARTITION_ID	COMPRESS_ATTR	DICT_BUILDER	DICT_BUILD_TIMESTAMP
SIMAP2	STAFF	0	0 Y		TABLE FUNCTION	2006-08-28-19.18.13.000000

1 record(s) selected.

この照会からの出力 (続き):

COMPRESS_DICT_SIZE	EXPAND_DICT_SIZE	ROWS_SAMPLED	PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH
13508	6314	68		72	89
					98

例 3: スキーマ SIMAP2 でのすべての表に関する合計ディクショナリー・サイズを決定します。

```
SELECT TABSCHEMA, TABNAME, DICT_BUILDER,
       (COMPRESS_DICT_SIZE+EXPAND_DICT_SIZE) AS TOTAL_DICT_SIZE,
       DBPARTITIONNUM, DATA_PARTITION_ID
FROM TABLE (SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO('SIMAP2', '', 'REPORT')) AS T
```

この照会からの出力:

TABSCHEMA	TABNAME	DICT_BUILDER	TOTAL_DICT_SIZE	DBPARTITIONNUM	DATA_PARTITION_ID
SIMAP2	ACT	NOT BUILT	0	0	0
SIMAP2	ADEFUSR	NOT BUILT	0	0	0
...					
SIMAP2	INVENTORY	NOT BUILT	0	0	0
SIMAP2	ORG	NOT BUILT	0	0	0
SIMAP2	PARTTAB	REORG	10008	0	0
SIMAP2	PARTTAB	REORG	5464	0	1
SIMAP2	PARTTAB	REORG	8456	0	2
SIMAP2	PARTTAB	REORG	6960	0	3
SIMAP2	PARTTAB	REORG	7136	0	4
...					
SIMAP2	STAFF	REORG	18624	0	0
SIMAP2	SUPPLIERS	NOT BUILT	0	0	0
SIMAP2	TESTTABLE	NOT BUILT	0	0	0

28 record(s) selected.

例 4: SIMAP2 スキーマの表のディクショナリー情報のレポートを表示します。

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO('SIMAP2', '', 'REPORT')) AS T
```

この照会からの出力:

TABSCHEMA	TABNAME	DBPARTITIONNUM	DATA_PARTITION_ID	COMPRESS_ATTR	DICT_BUILDER	DICT_BUILD_TIMESTAMP
SIMAP2	T1	0	0 Y		NOT BUILT	-
SIMAP2	T2	0	0 N		REORG	2007-02-03-17.35.28.000000
SIMAP2	T3	0	0 Y		INSPECT	2007-02-03-17.35.44.000000
SIMAP2	T4	0	0 N		NOT BUILT	-

4 record(s) selected.

この照会からの出力 (続き):

COMPRESS_DICT_SIZE	EXPAND_DICT_SIZE	ROWS_SAMPLED	PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH
0	0	0		0	0

1280	2562	-	-	-	-
1340	2232	-	-	-	-
0	0	0	0	0	0

## 使用上の注意

- *tabschema* と *tablename* の両方が指定される場合、その特定の表の情報のみが戻されます。
- *tabschema* が指定され、*tablename* が空 (") または NULL の場合、指定したスキーマのすべての表に関する情報が戻されます。
- *tabschema* が空 (") または NULL で、*tablename* が指定される場合、エラーは戻されます。特定の表の情報を取り出すには、その表がスキーマと表名の両方によって識別されることが必要です。
- *tabschema* と *tablename* の両方が空 (") または NULL の場合、すべての表の情報が戻されます。
- *tabschema* または *tablename* が存在しないか、あるいは *tablename* が表名 (タイプ T)、マテリアライズ照会表名 (タイプ S)、または階層表名 (タイプ H) と一致しない場合、空の結果セットが戻されます。
- ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数が指定の表のデータを検索するとき、この表関数は SYSTABLES の対応する行に対する共有ロックを獲得します。これは、戻されるデータの整合性を確保するための動作です (例えば、情報の検索中に、検索されている表が変更されないようにするなど)。ロックが保持されるのは、表関数の呼び出し期間中ではなく、表の圧縮情報を検索する間だけです。
- 照会された表が非 XML 表である場合は、XML ストレージ・オブジェクト (XDA) についての行が戻されます。

## 戻される情報

表 282. ADMIN\_TAB\_COMPRESS\_INFO 管理ビューおよび ADMIN\_GET\_TAB\_COMPRESS\_INFO によって戻される情報

列名	データ・タイプ	説明
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント
TABNAME	VARCHAR(128)	table_name - 表名モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID モニター・エレメント
COMPRESS_ATTR	CHAR(1)	表の COMPRESS 属性の状態。以下の値のいずれかになります。 <ul style="list-style-type: none"> <li>• 「Y」 = 行圧縮は yes に設定されます。</li> <li>• 「N」 = 行圧縮は no に設定されます。</li> </ul>

表 282. ADMIN\_TAB\_COMPRESS\_INFO 管理ビューおよび ADMIN\_GET\_TAB\_COMPRESS\_INFO によって戻される情報 (続き)

列名	データ・タイプ	説明
DICT_BUILDER	VARCHAR(30)	ディクショナリーを作成するために取られるコード・パス。以下の値のいずれかになります。 <ul style="list-style-type: none"> <li>• 'INSPECT' = INSPECT ROWCOMPESTIMATE</li> <li>• 'LOAD' = LOAD INSERT/REPLACE</li> <li>• 'NOT BUILT' = 使用可能なディクショナリーがありません</li> <li>• 'REDISTRIBUTE' = REDISTRIBUTE</li> <li>• 'REORG' = REORG RESETDICTIONARY</li> <li>• 'TABLE GROWTH' = INSERT</li> <li>• 'TABLE FUNCTION' = 'ESTIMATE' オプションのための表関数によって作成されています</li> </ul>
DICT_BUILD_TIMESTAMP	TIMESTAMP	ディクショナリーが作成された時刻のタイム・スタンプ。タイム・スタンプの細分性は秒単位です。使用可能なディクショナリーがない場合、タイム・スタンプは NULL です。
COMPRESS_DICT_SIZE	BIGINT	バイト単位で測定されたコンプレッション・ディクショナリーのサイズ。
EXPAND_DICT_SIZE	BIGINT	バイト単位で測定されたエクспанション・ディクショナリーのサイズ。履歴ディクショナリーが存在する場合、この値は現在のディクショナリー・サイズと履歴ディクショナリー・サイズの合計になります。
ROWS_SAMPLED	INTEGER	ディクショナリーの作成に役立つレコードの数。コンプレッション・ディクショナリーを含むマイグレーション済みの表はこの列で NULL を戻します。
PAGES_SAVED_PERCENT	SMALLINT	圧縮により削減されるページのパーセンテージこの情報は、サンプル・バッファ内のレコード・データのみに基づきます。コンプレッション・ディクショナリーを含むマイグレーション済みの表はこの列で NULL を戻します。
BYTES_SAVED_PERCENT	SMALLINT	圧縮により削減されるバイトのパーセンテージこの情報は、サンプル・バッファ内のレコード・データのみに基づきます。コンプレッション・ディクショナリーを含むマイグレーション済みの表はこの列で NULL を戻します。
AVG_COMPRESS_REC_LENGTH	SMALLINT	ディクショナリーの作成に役立っているレコードの平均圧縮レコード長。コンプレッション・ディクショナリーを含むマイグレーション済みの表はこの列で NULL を戻します。

## ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97

ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 表関数は、表、マテリアライズ照会表 (MQT)、階層表の圧縮された情報を戻します。



注: この表関数は推奨されないため、ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数および ADMIN\_GET\_TAB\_DICTIONARY\_INFO 表関数で置き換えられます。

戻される可能性のある情報の完全なリストは、1272 ページの表 283 の表を参照してください。

## 構文

```
►►—ADMIN_GET_TAB_COMPRESS_INFO_V97—(—tabschema—,—tablename—,—execmode—)——►►
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *tabschema*

スキーマ名を指定する、タイプ VARCHAR(128) の入力引数。

### *tablename*

表、マテリアライズ照会表、または階層表それぞれの名前を指定する、タイプ VARCHAR(128) の入力引数。

### *execmode*

実行モードを指定する、タイプ VARCHAR(30) の入力引数。実行モードは以下のいずれかのモードになります。

- 'REPORT' -- 最後の生成の時点での圧縮情報を報告します。これはデフォルト値です。
- 'ESTIMATE' -- 現在の表に基づいて新規の圧縮情報を生成します。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

例 1: 表 SIMAP2.STAFF に関する既存の圧縮情報を検索します。

```
SELECT *  
FROM TABLE (  
    SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO_V97('SIMAP2', 'STAFF', 'REPORT'))  
AS T
```

以下はこの照会の出力からの例です。

TABSCHEMA	TABNAME	DBPARTITIONNUM	DATA_PARTITION_ID	COMPRESS_ATTR	DICT_BUILDER	...
SIMAP2	STAFF	0	4	Y	REORG	...
SIMAP2	STAFF	0	4	Y	NOT BUILT	...

2 record(s) selected.

この照会からの出力 (続き):

DICT_BUILD_TIMESTAMP	COMPRESS_DICT_SIZE	EXPAND_DICT_SIZE	ROWS_SAMPLED	...
2009-03-31-12.19.30.000000	13312	5296	35	...
-	0	0	0	...

この照会からの出力 (続き):

PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH	OBJECT_TYPE
38	38	27	DATA
0	0	0	XML

例 2: 現時点での表 SIMAP2.STAFF に関する推定圧縮圧縮情報を検索します。

```
SELECT *
FROM TABLE (
  SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO_V97('SIMAP2', 'STAFF', 'ESTIMATE'))
AS T
```

以下はこの照会の出力からの例です。

TABSCHEMA	TABNAME	DBPARTITIONNUM	DATA_PARTITION_ID	COMPRESS_ATTR	DICT_BUILDER	...
SIMAP2	STAFF	0	4	Y	TABLE FUNCTION	...
SIMAP2	STAFF	0	4	Y	TABLE FUNCTION	...

2 record(s) selected.

この照会からの出力 (続き):

DICT_BUILD_TIMESTAMP	COMPRESS_DICT_SIZE	EXPAND_DICT_SIZE	ROWS_SAMPLED	...
2009-03-31-12.27.06.000000	13312	5296	35	...
2009-03-31-12.27.06.000000	13312	9544	8	...

この照会からの出力 (続き):

PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH	OBJECT_TYPE
38	38	27	DATA
75	75	95	XML

例 3: スキーマ SIMAP2 でのすべての表のオブジェクトに関する合計ディクショナリー・サイズを決定します。

```
SELECT TABSCHEMA, TABNAME, OBJECT_TYPE, DICT_BUILDER, (
  COMPRESS_DICT_SIZE+EXPAND_DICT_SIZE)
AS TOTAL_DICT_SIZE, DBPARTITIONNUM, DATA_PARTITION_ID
FROM TABLE (
  SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO_V97('SIMAP2', '', 'REPORT'))
AS T
```

この照会からの出力:

TABSCHEMA	TABNAME	OBJECT_TYPE	DICT_BUILDER	...
SIMAP2	ACT	DATA	NOT BUILT	...
SIMAP2	ACT	XML	NOT BUILT	...

```

SIMAP2      ADEFUSR      DATA      INSPECT      ...
SIMAP2      ADEFUSR      XML        NOT BUILT     ...
...
SIMAP2      CUSTOMER     DATA      REORG         ...
SIMAP2      CUSTOMER     XML        REORG         ...
SIMAP2      DEPARTMENT   DATA      NOT BUILT     ...
SIMAP2      DEPARTMENT   XML        NOT BUILT     ...
...
SIMAP2      STAFF        DATA      REORG         ...
SIMAP2      STAFF        XML        NOT BUILT     ...
SIMAP2      SUPPLIERS    DATA      TABLE GROWTH ...
SIMAP2      SUPPLIERS    XML        NOT BUILT     ...

```

44 record(s) selected.

この照会からの出力 (続き):

```

TOTAL_DICT_SIZE DBPARTITIONNUM DATA_PARTITION_ID
-----
          0          0          0
          0          0          0
        1890          0          0
          0          0          0
...
        6968          0          1
       24256          0          1
          0          1          0
          0          1          0
...
       18608          0          4
          0          0          4
        6960          0          2
          0          0          2

```

例 4: SIMAP2 スキーマの表のディクショナリー情報のレポートを表示します。

```

SELECT * FROM TABLE (
  SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO_V97('SIMAP2', '', 'REPORT'))
AS T

```

この照会からの出力:

```

TABSCHEMA TABNAME  DBPARTITIONNUM DATA_PARTITION_ID COMPRESS_ATTR DICT_BUILDER ...
-----
SIMAP2     ACT             0              0 N             NOT BUILT     ...
SIMAP2     ACT             0              0 N             NOT BUILT     ...
SIMAP2     ADEFUSR        0              0 N             INSPECT      ...
SIMAP2     ADEFUSR        0              0 N             NOT BUILT     ...
...
SIMAP2     CUSTOMER       0              1 Y             REORG        ...
SIMAP2     CUSTOMER       0              1 Y             REORG        ...
...
SIMAP2     STAFF          0              4 Y             REORG        ...
SIMAP2     STAFF          0              4 Y             NOT BUILT     ...
SIMAP2     SUPPLIERS     0              2 N             NOT BUILT     ...
SIMAP2     SUPPLIERS     0              2 N             NOT BUILT     ...

```

44 record(s) selected.

この照会からの出力 (続き):

```

DICT_BUILD_TIMESTAMP      COMPRESS_DICT_SIZE EXPAND_DICT_SIZE ROWS_SAMPLED ...
-----
-                          0                  0                0 ...
-                          0                  0                0 ...
2009-03-31-12.11.02.000000 290                1890             22 ...
-                          0                  0                0 ...

```

```

...
2009-03-31-11.08.18.000000          3968          3000          6 ...
2009-03-31-11.08.18.000000          13312         10944          6 ...
...
2009-03-31-12.19.30.000000          13312          5296          35 ...
-                                     0              0              0 ...
-                                     0              0              0 ...
-                                     0              0              0 ...

```

この照会からの出力 (続き):

```

PAGES_SAVED_PERCENT BYTES_SAVED_PERCENT AVG_COMPRESS_REC_LENGTH OBJECT_TYPE
-----
          0              0              0 DATA
          0              0              0 XML
         20              25              21 DATA
          0              0              0 XML
...
         70              70              31 DATA
         66              66              235 XML
...
         38              38              27 DATA
          0              0              0 XML
          0              0              0 DATA
          0              0              0 XML

```

例 5: SIMAP2 スキーマの表の DATA オブジェクトのディクショナリー情報のレポートを表示します。

```

SELECT * FROM TABLE (
    SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO_V97('SIMAP2',' ','REPORT')
    WHERE OBJECT_TYPE='DATA')

```

この照会からの出力:

```

TABSCHEMA TABNAME DBPARTITIONNUM DATA_PARTITION_ID COMPRESS_ATTR DICT_BUILDER ...
-----
SIMAP2     ACT              0              0 N              NOT BUILT ...
SIMAP2     ADEFUSR         0              0 N              INSPECT ...
...
SIMAP2     CUSTOMER        0              1 Y              REORG ...
SIMAP2     DEPARTMENT      1              0 N              NOT BUILT ...
...
SIMAP2     STAFF           0              4 Y              REORG ...
SIMAP2     SUPPLIERS       0              2 N              NOT BUILT ...

```

22 record(s) selected.

この照会からの出力 (続き):

```

DICT_BUILD_TIMESTAMP COMPRESS_DICT_SIZE EXPAND_DICT_SIZE ROWS_SAMPLED...
-----
-                                     0              0              0 ...
2009-03-31-12.11.02.000000          290            1890            22 ...
...
2009-03-31-11.08.18.000000          3968           3000            6 ...
-                                     0              0              0 ...
...
2009-03-31-12.19.30.000000          13312          5296            35 ...
-                                     0              0              0 ...

```

この照会からの出力 (続き):

```

PAGES_SAVED_PERCENT BYTES_SAVED_PERCENT AVG_COMPRESS_REC_LENGTH OBJECT_TYPE
-----
          0              0              0 DATA
         20              25              21 DATA

```

70	70	31 DATA
0	0	0 DATA
38	38	27 DATA
0	0	0 DATA

例 6: SIMAP2 スキーマの CUSTOMER 表の XML オブジェクトのディクショナリー情報のレポートを表示します。

```
SELECT * FROM TABLE (
  SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO_V97('SIMAP2', 'CUSTOMER', 'REPORT'))
WHERE OBJECT_TYPE='XML'
```

この照会からの出力:

TABSCHHEMA	TABNAME	DBPARTITIONNUM	DATA_PARTITION_ID	COMPRESS_ATTR	DICT_BUILDER	...
SIMAP2	CUSTOMER	0	1	Y	REORG	...

この照会からの出力 (続き):

DICT_BUILD_TIMESTAMP	COMPRESS_DICT_SIZE	EXPAND_DICT_SIZE	ROWS_SAMPLED	...
2009-03-31-11.08.18.000000	13312	10944	6	...

この照会からの出力 (続き):

PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH	OBJECT_TYPE
66	66	235	XML

## 使用上の注意

- *tabschema* と *tablename* の両方が指定される場合、その特定の表の情報のみが戻されます。
- *tabschema* が指定され、*tablename* が空 (") または NULL の場合、指定したスキーマのすべての表に関する情報が戻されます。
- *tabschema* が空 (") または NULL で、*tablename* が指定される場合、エラーは戻されます。特定の表の情報を取り出すには、その表がスキーマと表名の両方によって識別されることが必要です。
- *tabschema* と *tablename* の両方が空 (") または NULL の場合、すべての表の情報が戻されます。
- *tabschema* または *tablename* が存在しないか、あるいは *tablename* が表名 (タイプ T)、マテリアライズ照会表名 (タイプ S)、または階層表名 (タイプ H) と一致しない場合、空の結果セットが戻されます。
- ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 表関数が指定の表のデータを検索するとき、この表関数は SYSTABLES の対応する行に対する共有ロックを獲得します。これは、戻されるデータの整合性を確保するための動作です (例えば、情報の検索中に、検索されている表が変更されないようにするなど)。ロックが保持されるのは、表関数の呼び出し期間中ではなく、表の圧縮情報を検索する間だけです。
- 照会された表が非 XML 表である場合は、XML ストレージ・オブジェクト (XDA) についての行が戻されます。

## 戻される情報

表 283. ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 によって戻される情報

列名	データ・タイプ	説明
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント
TABNAME	VARCHAR(128)	table_name - 表名モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID モニター・エレメント
COMPRESS_ATTR	CHAR(1)	表の COMPRESS 属性の状態。以下の値のいずれかになります。 <ul style="list-style-type: none"> <li>「Y」 = 行圧縮は yes に設定されます。</li> <li>「N」 = 行圧縮は no に設定されます。</li> </ul>
DICTIONARY_BUILDER	VARCHAR(30)	ディクショナリーを作成するために取られるコード・パス。以下の値のいずれかになります。 <ul style="list-style-type: none"> <li>'INSPECT' = INSPECT ROWCOMPESTIMATE</li> <li>'LOAD' = LOAD INSERT/REPLACE</li> <li>'NOT BUILT' = 使用可能なディクショナリーがありません</li> <li>'REDISTRIBUTE' = REDISTRIBUTE</li> <li>'REORG' = REORG RESETDICTIONARY</li> <li>'TABLE GROWTH' = INSERT</li> <li>'TABLE FUNCTION' = 'ESTIMATE' オプションのための表関数によって作成されています</li> </ul>
DICTIONARY_BUILD_TIMESTAMP	TIMESTAMP	ディクショナリーが作成された時刻のタイム・スタンプ。タイム・スタンプの細分性は秒単位です。使用可能なディクショナリーがない場合、タイム・スタンプは NULL です。
COMPRESS_DICT_SIZE	BIGINT	バイト単位で測定されたコンプレッション・ディクショナリーのサイズ。
EXPAND_DICT_SIZE	BIGINT	バイト単位で測定されたエクспанション・ディクショナリーのサイズ。履歴ディクショナリーが存在する場合、この値は現在のディクショナリー・サイズと履歴ディクショナリー・サイズの合計になります。
ROWS_SAMPLED	INTEGER	ディクショナリーの作成に役立つレコードの数。コンプレッション・ディクショナリーを含むマイグレーション済みの表はこの列で NULL を戻します。
PAGES_SAVED_PERCENT	SMALLINT	圧縮により削減されるページのパーセンテージこの情報は、サンプル・バッファ内のレコード・データのみに基づきます。コンプレッション・ディクショナリーを含むマイグレーション済みの表はこの列で NULL を戻します。
BYTES_SAVED_PERCENT	SMALLINT	圧縮により削減されるバイトのパーセンテージこの情報は、サンプル・バッファ内のレコード・データのみに基づきます。コンプレッション・ディクショナリーを含むマイグレーション済みの表はこの列で NULL を戻します。

表 283. ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
AVG_COMPRESS_REC_LENGTH	SMALLINT	ディクショナリーの作成に役立っているレコードの平均圧縮レコード長。コンプレッション・ディクショナリーを含むマイグレーション済みの表はこの列で NULL を戻します。
OBJECT_TYPE	VARCHAR(4)	objtype - オブジェクト・タイプ・モニター・エレメント

## ADMIN\_GET\_TAB\_INFO\_V95 表関数 - 表のサイズおよび状態に関する情報の検索

ADMIN\_GET\_TAB\_INFO\_V95 表関数は、現在カタログ・ビューで使用できない表のサイズおよび状態に関する情報を検索する方式を提供します。

注: この表関数は使用すべきではなく、294 ページの『ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO 表関数 - 表のサイズおよび状態に関する情報の検索』に置き換えられました。

戻される可能性のある情報の完全なリストは、1275 ページの表 284 の表を参照してください。

### 構文

```
▶▶—ADMIN_GET_TAB_INFO_V95—(—tabschema—,—tabname—)————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

*tabschema*

スキーマ名を指定する、タイプ VARCHAR(128) の入力引数。

*tabname*

表、マテリアライズ照会表、または階層表それぞれの名前を指定する、タイプ VARCHAR(128) の入力引数。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: 表 DBUSER1.EMPLOYEE のサイズおよび状態に関する情報を検索します。

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_INFO_V95('DBUSER1', 'EMPLOYEE'))
AS T
```

例 2: 非パーティション表 (DBUSER1.EMPLOYEE) が存在し、関連オブジェクト (例えば索引や LOB など) がすべて 1 つの表スペースに保管されていると仮定します。表が表スペース内のどのくらいの物理スペースを使用しているかを計算します。

```
SELECT (data_object_p_size + index_object_p_size + long_object_p_size +
lob_object_p_size + xml_object_p_size) as total_p_size
FROM TABLE( SYSPROC.ADMIN_GET_TAB_INFO_V95( 'DBUSER1', 'EMPLOYEE' )) AS T
```

表が別の表スペースに移動されるときにどのくらいのスペースが必要になるかを計算します。ここでは、新規の表スペースには元の表スペースと同じページ・サイズおよびエクステント・サイズがあるとします。

```
SELECT (data_object_l_size + index_object_l_size + long_object_l_size +
lob_object_l_size + xml_object_l_size) as total_l_size
FROM TABLE( SYSPROC.ADMIN_GET_TAB_INFO_V95( 'DBUSER1', 'EMPLOYEE' )) AS T
```

例 3: 表 T1 について収集された統計情報の現在のタイプを調べます。

```
db2 => select substr(tabschema, 1, 10) as tbschema, substr(tabname, 1, 10)
as tbnname, statstype from SYSIBMADM.ADMINTABINFO where tabname = 'T1';
```

TBSHEMA	TBNNAME	STATSTYPE
DB2USER1	T1	U

1 record(s) selected.

## 使用上の注意

- *tabschema* と *tabname* の両方が指定される場合、その特定の表の情報のみが戻されます。
- *tabschema* が指定され、*tabname* が空 (") または NULL の場合、指定したスキーマのすべての表に関する情報が戻されます。
- *tabschema* が空 (") または NULL で、*tabname* が指定される場合、エラーは戻されます。特定の表の情報を取り出すには、その表がスキーマと表名の両方によって識別されることが必要です。
- *tabschema* と *tabname* の両方が空 (") または NULL の場合、すべての表の情報が戻されます。
- *tabschema* または *tabname* が存在しないか、あるいは *tabname* が表名 (タイプ T)、マテリアライズ照会表名 (タイプ S)、または階層表名 (タイプ H) と一致しない場合、空の結果セットが戻されます。
- ADMIN\_GET\_TAB\_INFO\_V95 表関数が指定の表のデータを検索するとき、この表関数は SYSTABLES の対応する行に対する共有ロックを獲得します。これは、戻されるデータの整合性を確保するための動作です (例えば、情報の検索中に、検索されている表がドロップされないようにするなど)。ロックが保持されるのは、表関数の呼び出し期間中ではなく、表のサイズおよび状態に関する情報を検索する間だけです。



- SMS 表スペースの表の物理サイズが報告されますが、このサイズは論理サイズと同じです。
- 表で INPLACE の REORG がアクティブになっていると、データ・オブジェクトの物理サイズ (DATA\_OBJECT\_P\_SIZE) は計算されません。論理サイズだけが戻されます。INPLACE の REORG が表でアクティブになっているかどうかは、INPLACE\_REORG\_STATUS 出力の列を見ると分かります。

#### REDISTRIBUTING\_PENDING

1. 指定された表について再配分は実行されていません N
2. 再配分の実行がデータベース・パーティション・グループで開始されましたが、表では開始されませんでした N
3. データを移動する前のフェーズで再配分が失敗しました N
4. データの移動のフェーズで再配分が失敗しました Y
5. 表について再配分が正常に完了し、コミットされました N

### 戻される情報

表 284. ADMIN\_GET\_TAB\_INFO\_V95 表関数によって戻される情報

列名	データ・タイプ	説明
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント
TABNAME	VARCHAR(128)	table_name - 表名モニター・エレメント
TABTYPE	CHAR(1)	表タイプ: <ul style="list-style-type: none"> <li>• 「H」 = 階層表</li> <li>• 「S」 = マテリアライズ照会表</li> <li>• 「T」 = 表</li> </ul>
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID モニター・エレメント
AVAILABLE	CHAR(1)	表の状態: <ul style="list-style-type: none"> <li>• 「N」 = 表は使用不可。表を使用できない場合、サイズおよび状態に関する他の出力列はすべて NULL になります。</li> <li>• 「Y」 = 表は使用可能。</li> </ul> <b>注:</b> リカバリー不能ロードでのロールフォワードを行うと、表の状態が使用不可になります。

表 284. ADMIN\_GET\_TAB\_INFO\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
DATA_OBJECT_L_SIZE	BIGINT	データ・オブジェクトの論理サイズ。表に対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表に対して物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。多次元クラスタリング (MDC) 表の場合、このサイズにはブロック・マップ・オブジェクトの論理サイズが含まれます。戻されるサイズは、表に対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成されるオブジェクトの場合は Extent Map Page (EMP) エクステントの見積もりを考慮に入れます。このサイズは、基本表のみの論理サイズを表します。LOB データ、長形式データ、索引、および XML オブジェクトが消費するスペースは別の列で報告されます。
DATA_OBJECT_P_SIZE	BIGINT	データ・オブジェクトの物理サイズ。表に対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。MDC 表の場合、このサイズにはブロック・マップ・オブジェクトのサイズが含まれます。戻されるサイズは、表に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成されるオブジェクトの EMP エクステントが含まれます。このサイズは、基本表のみの物理サイズを表します。LOB データ、長形式データ、索引、および XML オブジェクトが消費するスペースは別の列で報告されます。
INDEX_OBJECT_L_SIZE	BIGINT	索引オブジェクトの論理サイズ。表で定義される索引に対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表の索引データを保持するために物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。戻されるサイズは、索引に対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される索引の場合は EMP エクステントの見積もりを考慮に入れます。この値は非パーティション表の場合にのみ報告されます。パーティション表の場合、この値は 0 になります。
INDEX_OBJECT_P_SIZE	BIGINT	索引オブジェクトの物理サイズ。表で定義される索引に対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。戻されるサイズは、索引に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される索引の EMP エクステントが含まれます。この値は非パーティション表の場合にのみ報告されます。パーティション表の場合、この値は 0 になります。

表 284. ADMIN\_GET\_TAB\_INFO\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
LONG_OBJECT_L_SIZE	BIGINT	長形式オブジェクトの論理サイズ。表の長形式フィールド・データに対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表の長形式フィールド・データを保持するために物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。戻されるサイズは、長形式フィールド・データに対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される長形式フィールド・データの場合は EMP エクステントの見積もりを考慮に入れます。
LONG_OBJECT_P_SIZE	BIGINT	長形式オブジェクトの物理サイズ。表の長形式フィールド・データに対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。戻されるサイズは、長形式フィールド・データに割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される長形式フィールド・データの EMP エクステントが含まれます。
LOB_OBJECT_L_SIZE	BIGINT	LOB オブジェクトの論理サイズ。表の LOB データに対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表の LOB データを保持するために物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。サイズには LOB 割り振りオブジェクトに対して論理的に割り振られるスペースが含まれます。戻されるサイズは、LOB データに対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される LOB データの場合は EMP エクステントの見積もりを考慮に入れます。
LOB_OBJECT_P_SIZE	BIGINT	LOB オブジェクトの物理サイズ。表の LOB データに対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。サイズには LOB 割り振りオブジェクトに対して割り振られるスペースが含まれます。戻されるサイズは、LOB データに割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される LOB データの EMP エクステントが含まれます。
XML_OBJECT_L_SIZE	BIGINT	XML オブジェクトの論理サイズ。表の XML データに対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表の XML データを保持するために物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。戻されるサイズは、XML データに対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される XML データの場合は EMP エクステントの見積もりを考慮に入れます。

表 284. ADMIN\_GET\_TAB\_INFO\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
XML_OBJECT_P_SIZE	BIGINT	XML オブジェクトの物理サイズ。表の XML データに対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。戻されるサイズは、XML データに割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される XML データの EMP エクステントが含まれます。
INDEX_TYPE	SMALLINT	現在表で使用されている索引のタイプを示します。これは、以下のものを戻します。 <ul style="list-style-type: none"> <li>• タイプ 1 索引が使用されている場合は 1。</li> <li>• タイプ 2 索引が使用されている場合は 2。</li> </ul>
REORG_PENDING	CHAR(1)	「Y」は、REORG 推奨の変更がすでに表に適用されており、クラシック (オフライン) REORG が必要であることを示しています。それ以外の場合は「N」が戻されます。
INPLACE_REORG_STATUS	VARCHAR(10)	表に対する表のインプレース再編成の現在の状況。状況値は、以下のいずれかになります。 <ul style="list-style-type: none"> <li>• ABORTED (PAUSED 状態にあるが、RESUME は不可。STOP は必須)</li> <li>• EXECUTING</li> <li>• NULL (表に対して INPLACE の REORG が実行されていない場合)</li> <li>• PAUSED</li> </ul>
LOAD_STATUS	VARCHAR(12)	表に対するロード操作の現在の状況。状況値は、以下のいずれかになります。 <ul style="list-style-type: none"> <li>• IN_PROGRESS</li> <li>• NULL (表でロードが進行しておらず、表がロード・ペンディング状態になっていない場合)</li> <li>• PENDING</li> </ul>
READ_ACCESS_ONLY	CHAR(1)	表が「読み取りアクセス専用」の状態になっていれば「Y」、そうでなければ「N」になります。「N」の値を、表が完全にアクセス可能であるという意味に解釈するべきではありません。ロードが進行中またはペンディング状態の場合、「Y」の値は表データが読み取りアクセス可能であることを意味し、「N」の値は表がアクセス不能であることを意味します。同様に、表の状況が SET INTEGRITY ペンディングである場合 (SYSCAT.TABLES STATUS 列を参照)、「N」の値は表がアクセス不能であることを意味します。
NO_LOAD_RESTART	CHAR(1)	「Y」の値は、表が部分的にロードされている状態になっていることを示します。この場合、ロードを再始動することができません。この状態になっていなければ「N」の値が戻されます。
NUM_REORG_REC_ALTERS	SMALLINT	最後に再編成が行われてからこの表に対して実行された REORG 推奨の変更操作 (例えば直後に再編成を必要とする変更操作) の回数。

表 284. ADMIN\_GET\_TAB\_INFO\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
INDEXES_REQUIRE_REBUILD	CHAR(1)	表で定義される索引のいずれかが再ビルドを必要とする場合は「Y」、必要としない場合は「N」。
LARGE_RIDS	CHAR(1)	表がラージ行 ID (RID) を使用しているかどうかを示します (4 バイトのページ番号と 2 バイトのスロット番号)。 「Y」の値は表がラージ RID を使用していることを示し、「N」は使用していないことを示します。表がラージ RID をサポートしている (つまり表が LARGE 表スペースにある) もの、少なくとも表の索引の 1 つがまだ再編成されていないかまたは再ビルドされていない場合、「P」(保留)の値が戻されます。これは表が 4 バイトの RID を使用しているためです (これは表または索引を変換するためのアクションを取る必要があることを意味します)。
LARGE_SLOTS	CHAR(1)	表がラージ・スロット (これは 1 ページにつき 255 を超える行が可能です) を使用しているかどうかを示します。 「Y」の値は表がラージ・スロットを使用していることを示し、「N」は使用していないことを示します。表がラージ・スロットをサポートしている (つまり表が LARGE 表スペースにある) もの、表に対してまだオフラインの表の再編成または表の切り捨て操作が実行されていない場合、「P」(保留)の値が戻されます。これは、表が 1 ページにつき最大 255 行のラージ・スロットを使用しているためです。
DICTIONARY_SIZE	BIGINT	ディクショナリーのサイズ (バイト)。表に行コンプレッション・ディクショナリーが存在する場合に行の圧縮で使用されます。
BLOCKS_PENDING_CLEANUP	BIGINT	blocks_pending_cleanup - クリーンアップ保留中のロールアウト済みブロック・モニター・エレメント

表 284. ADMIN\_GET\_TAB\_INFO\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
STATSTYPE	CHAR(1)	<ul style="list-style-type: none"> <li>「F」 = 表または索引のスキャンを行わずにシステムが作り上げた統計。これらの統計はメモリーに保管され、システム・カタログに保管されているものとは異なります。これは一時状態であり、最終的に完全な統計が DB2 によって収集され、システム・カタログに保管されます。</li> <li>「A」 = システムが非同期に収集した統計。統計は DB2 によりバックグラウンド・プロセスで自動的に収集され、システム・カタログに保管されました。</li> <li>「S」 = システムが同期に収集した統計。統計は DB2 によって SQL ステートメントのコンパイル中に自動的に収集されました。これらの統計はメモリーに保管され、システム・カタログに保管されているものとは異なります。これは一時状態であり、最終的に DB2 が統計をシステム・カタログに保管します。</li> <li>「U」 = ユーザーが収集した統計。統計の収集は、RUNSTATS、CREATE INDEX、LOAD、REDISTRIBUTE などのユーティリティーを使用するか、システム・カタログ統計を手動で更新することによって、ユーザーが開始しました。</li> <li>NULL = 不明なタイプ</li> </ul>

## ADMIN\_GET\_TAB\_INFO\_V97 表関数 - 表のサイズおよび状態に関する情報の取得

ADMIN\_GET\_TAB\_INFO\_V97 表関数は、カタログ・ビューで現在使用できない表のサイズおよび状態に関する情報を取得するためのメソッドを提供します。

注: この表関数は使用すべきではなく、294 ページの『ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO 表関数 - 表のサイズおよび状態に関する情報の検索』に置き換えられました。

戻される可能性のある情報の完全なリストは、1282 ページの表 285 の表を参照してください。

### 構文

▶▶—ADMIN\_GET\_TAB\_INFO\_V97—(—*tabschema*—,—*tabname*—)—▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

*tabschema*

スキーマ名を指定する、タイプ VARCHAR(128) の入力引数。

*tabname*

表、マテリアライズ照会表、または階層表それぞれの名前を指定する、タイプ VARCHAR(128) の入力引数。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: 表 DBUSER1.EMPLOYEE のサイズおよび状態に関する情報を検索します。

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_INFO_V97('DBUSER1', 'EMPLOYEE'))
AS T
```

例 2: 非パーティション表 (DBUSER1.EMPLOYEE) が存在し、関連オブジェクト (例えば索引や LOB など) がすべて 1 つの表スペースに保管されていると仮定します。表が表スペース内のどのくらいの物理スペースを使用しているかを計算します。

```
SELECT (data_object_p_size + index_object_p_size + long_object_p_size +
lob_object_p_size + xml_object_p_size) as total_p_size
FROM TABLE( SYSPROC.ADMIN_GET_TAB_INFO_V97( 'DBUSER1', 'EMPLOYEE' )) AS T
```

表が別の表スペースに移動されるときにどのくらいのスペースが必要になるかを計算します。ここでは、新規の表スペースには元の表スペースと同じページ・サイズおよびエクステンツ・サイズがあるとします。

```
SELECT (data_object_l_size + index_object_l_size + long_object_l_size +
lob_object_l_size + xml_object_l_size) as total_l_size
FROM TABLE( SYSPROC.ADMIN_GET_TAB_INFO_V97( 'DBUSER1', 'EMPLOYEE' )) AS T
```

例 3: 表 DBUSER1.EMPLOYEE 用のコンプレッション・ディクショナリーの合計サイズを判別します。

```
SELECT SUBSTR(TABSCHEMA,1,10) AS TBSHEMA, SUBSTR(TABNAME,1,10) AS TBNAME,
DICTIONARY_SIZE + XML_DICTIONARY_SIZE AS TOTAL_DICTIONARY_SIZE
FROM TABLE(SYSPROC.ADMIN_GET_TAB_INFO_V97('DBUSER1','EMPLOYEE'))
```

例 4: マルチディメンション・クラスタリング表 SAMPLE.STAFF から再利用可能なスペースの量を判別します。

```
SELECT RECLAIMABLE_SPACE
FROM TABLE(SYSPROC.ADMIN_GET_TAB_INFO_V97('SAMPLE','STAFF'))
```

## 使用上の注意

- *tabschema* と *tabname* の両方が指定される場合、その特定の表の情報のみが戻されます。

- *tabschema* が指定され、*tablename* が NULL または空ストリング (") の場合、指定したスキーマのすべての表に関する情報が戻されます。
- *tabschema* が NULL または空ストリング (") で、*tablename* が指定される場合、エラーは戻されます。特定の表の情報を取り出すには、その表がスキーマと表名の両方によって識別されることが必要です。
- *tabschema* と *tablename* の両方が NULL または空ストリング (") の場合、すべての表の情報が戻されます。
- *tabschema* または *tablename* が存在しないか、あるいは *tablename* が表名 (タイプ T)、マテリアライズ照会表名 (タイプ S)、または階層表名 (タイプ H) と一致しない場合、空の結果セットが戻されます。
- ADMIN\_GET\_TAB\_INFO\_V97 表関数が指定の表のデータを検索するとき、この表関数は SYSTABLES の対応する行に対する共有ロックを獲得します。これは、戻されるデータの整合性を確保するための動作です (例えば、情報の検索中に、検索されている表がドロップされないようにするなど)。ロックが保持されるのは、表関数の呼び出し期間中ではなく、表のサイズおよび状態に関する情報を検索する間だけです。
- SMS 表スペースの表の物理サイズが報告されますが、このサイズは論理サイズと同じです。
- 表で INPLACE の REORG がアクティブになっていると、データ・オブジェクトの物理サイズ (DATA\_OBJECT\_P\_SIZE) は計算されません。論理サイズだけが戻されます。INPLACE の REORG が表でアクティブになっているかどうかは、INPLACE\_REORG\_STATUS 出力の列を見ると分かります。

#### REDISTRIBUTING\_PENDING

1. 指定された表について再配分は実行されていません N
2. 再配分の実行がデータベース・パーティション・グループで開始されましたが、表では開始されませんでした N
3. データを移動する前のフェーズで再配分が失敗しました N
4. データの移動のフェーズで再配分が失敗しました Y
5. 表について再配分が正常に完了し、コミットされました N

### 戻される情報

表 285. ADMIN\_TABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO\_V97 によって戻される情報

列名	データ・タイプ	説明
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名モニター・エレメント
TABNAME	VARCHAR(128)	table_name - 表名モニター・エレメント
TABTYPE	CHAR(1)	表タイプ: <ul style="list-style-type: none"> <li>• 「H」 = 階層表</li> <li>• 「S」 = マテリアライズ照会表</li> <li>• 「T」 = 表</li> </ul>
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID モニター・エレメント



表 285. ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
AVAILABLE	CHAR(1)	<p>表の状態:</p> <ul style="list-style-type: none"> <li>「N」 = 表は使用不可。表を使用できない場合、サイズおよび状態に関する他の出力列はすべて NULL になります。</li> <li>「Y」 = 表は使用可能。</li> </ul> <p><b>注:</b> リカバリー不能ロードでのロールフォワードを行うと、表の状態が使用不可になります。</p>
DATA_OBJECT_L_SIZE	BIGINT	<p>データ・オブジェクトの論理サイズ。表に対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表に対して物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。多次元クラスタリング (MDC) 表の場合、このサイズにはブロック・マップ・オブジェクトの論理サイズが含まれます。戻されるサイズは、表に対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成されるオブジェクトの場合は Extent Map Page (EMP) エクステントの見積もりを考慮に入れます。このサイズは、基本表のみの論理サイズを表します。LOB データ、長形式データ、索引、および XML オブジェクトが消費するスペースは別の列で報告されます。</p>
DATA_OBJECT_P_SIZE	BIGINT	<p>データ・オブジェクトの物理サイズ。表に対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。MDC 表の場合、このサイズにはブロック・マップ・オブジェクトのサイズが含まれます。戻されるサイズは、表に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成されるオブジェクトの EMP エクステントが含まれます。このサイズは、基本表のみの物理サイズを表します。LOB データ、長形式データ、索引、および XML オブジェクトが消費するスペースは別の列で報告されます。</p>

表 285. ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
INDEX_OBJECT_L_SIZE	BIGINT	<p>索引オブジェクトの論理サイズ。表で定義される索引に対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表の索引データを保持するために物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。戻されるサイズは、索引に対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される索引の場合は EMP エクステントの見積もりを考慮に入れます。</p> <p>パーティション表のパーティション索引の場合、これは DATA_PARTITION_ID によって識別されるデータ・パーティションの索引パーティションを含む索引オブジェクトの論理サイズです。この値には、パーティション表の非パーティション索引は考慮に入れられません。パーティション索引と非パーティション索引の両方の情報については、ADMIN_GET_INDEX_INFO 関数を使用できます。</p>
INDEX_OBJECT_P_SIZE	BIGINT	<p>索引オブジェクトの物理サイズ。表で定義される索引に対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。戻されるサイズは、索引に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される索引の EMP エクステントが含まれます。</p> <p>パーティション表のパーティション索引の場合、これは DATA_PARTITION_ID によって識別されるデータ・パーティションの索引パーティションを含む索引オブジェクトの物理サイズです。この値には、パーティション表の非パーティション索引は考慮に入れられません。パーティション索引と非パーティション索引の両方の情報については、ADMIN_GET_INDEX_INFO 関数を使用できます。</p>
LONG_OBJECT_L_SIZE	BIGINT	<p>長形式オブジェクトの論理サイズ。表の長形式フィールド・データに対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表の長形式フィールド・データを保持するために物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。戻されるサイズは、長形式フィールド・データに対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される長形式フィールド・データの場合は EMP エクステントの見積もりを考慮に入れます。</p>
LONG_OBJECT_P_SIZE	BIGINT	<p>長形式オブジェクトの物理サイズ。表の長形式フィールド・データに対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。戻されるサイズは、長形式フィールド・データに割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される長形式フィールド・データの EMP エクステントが含まれます。</p>

表 285. ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
LOB_OBJECT_L_SIZE	BIGINT	LOB オブジェクトの論理サイズ。表の LOB データに対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表の LOB データを保持するために物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。サイズには LOB 割り振りオブジェクトに対して論理的に割り振られるスペースが含まれます。戻されるサイズは、LOB データに対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される LOB データの場合は EMP エクステントの見積もりを考慮に入れます。
LOB_OBJECT_P_SIZE	BIGINT	LOB オブジェクトの物理サイズ。表の LOB データに対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。サイズには LOB 割り振りオブジェクトに対して割り振られるスペースが含まれます。戻されるサイズは、LOB データに割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される LOB データの EMP エクステントが含まれます。
XML_OBJECT_L_SIZE	BIGINT	XML オブジェクトの論理サイズ。表の XML データに対して論理的に割り振られるディスク・スペースの量 (KB 単位で報告)。論理サイズとは、表が認識するスペースの量のことです。このサイズは表の XML データを保持するために物理的に割り振られるスペースの量より小さくなることもあります (例えば論理表の切り捨ての場合)。戻されるサイズは、XML データに対して論理的に割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される XML データの場合は EMP エクステントの見積もりを考慮に入れます。
XML_OBJECT_P_SIZE	BIGINT	XML オブジェクトの物理サイズ。表の XML データに対して物理的に割り振られるディスク・スペースの量 (KB 単位で報告)。戻されるサイズは、XML データに割り振られるすべてのエクステントを考慮に入れ、DMS 表スペースで作成される XML データの EMP エクステントが含まれます。
INDEX_TYPE	SMALLINT	現在表で使用されている索引のタイプを示します。これは、以下のものを戻します。 <ul style="list-style-type: none"> <li>• タイプ 1 索引が使用されている場合は 1。</li> <li>• タイプ 2 索引が使用されている場合は 2。</li> </ul>
REORG_PENDING	CHAR(1)	「Y」は、REORG 推奨の変更がすでに表に適用されており、クラシック (オフライン) REORG が必要であることを示しています。それ以外の場合は「N」が戻されます。

表 285. ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
INPLACE_REORG_STATUS	VARCHAR(10)	表に対する表のインプレース再編成の現在の状況。状況値は、以下のいずれかになります。 <ul style="list-style-type: none"> <li>• ABORTED (PAUSED 状態にあるが、RESUME は不可。STOP は必須)</li> <li>• EXECUTING</li> <li>• NULL (表に対して INPLACE の REORG が実行されていない場合)</li> <li>• PAUSED</li> </ul>
LOAD_STATUS	VARCHAR(12)	表に対するロード操作の現在の状況。状況値は、以下のいずれかになります。 <ul style="list-style-type: none"> <li>• IN_PROGRESS</li> <li>• NULL (表でロードが進行しておらず、表がロード・ペンディング状態になっていない場合)</li> <li>• PENDING</li> </ul>
READ_ACCESS_ONLY	CHAR(1)	表が「読み取りアクセス専用」の状態になっていれば「Y」、そうでなければ「N」になります。「N」の値を、表が完全にアクセス可能であるという意味に解釈するべきではありません。ロードが進行中またはペンディング状態の場合、「Y」の値は表データが読み取りアクセス可能であることを意味し、「N」の値は表がアクセス不能であることを意味します。同様に、表の状況が SET INTEGRITY ペンディングである場合 (SYSCAT.TABLES STATUS 列を参照)、「N」の値は表がアクセス不能であることを意味します。
NO_LOAD_RESTART	CHAR(1)	「Y」の値は、表が部分的にロードされている状態になっていることを示します。この場合、ロードを再始動することができません。この状態になっていなければ「N」の値が戻されます。
NUM_REORG_REC_ALTERS	SMALLINT	最後に再編成が行われてからこの表に対して実行された REORG 推奨の変更操作 (例えば直後に再編成を必要とする変更操作) の回数。
INDEXES_REQUIRE_REBUILD	CHAR(1)	非パーティション表において、表に定義されている索引のいずれかが再ビルドを必要とする場合は「Y」、必要としない場合は「N」。パーティション表において、DATA_PARTITION_ID によって識別されるデータ・パーティションの索引パーティションのいずれかが再ビルドを必要とする場合は「Y」、必要としない場合は「N」。

表 285. ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
LARGE_RIDS	CHAR(1)	表がラージ行 ID (RID) を使用しているかどうかを示します (4 バイトのページ番号と 2 バイトのスロット番号)。 「Y」の値は表がラージ RID を使用していることを示し、「N」は使用していないことを示します。表がラージ RID をサポートしている (つまり表が LARGE 表スペースにある) もの、少なくとも表の索引の 1 つがまだ再編成されていないかまたは再ビルドされていない場合、「P」(保留)の値が戻されます。これは表が 4 バイトの RID を使用しているためです (これは表または索引を変換するためのアクションを取る必要があることを意味します)。
LARGE_SLOTS	CHAR(1)	表がラージ・スロット (これは 1 ページにつき 255 を超える行が可能です) を使用しているかどうかを示します。 「Y」の値は表がラージ・スロットを使用していることを示し、「N」は使用していないことを示します。表がラージ・スロットをサポートしている (つまり表が LARGE 表スペースにある) もの、表に対してまだオフラインの表の再編成または表の切り捨て操作が実行されていない場合、「P」(保留)の値が戻されます。これは、表が 1 ページにつき最大 255 行のラージ・スロットを使用しているためです。
DICTIONARY_SIZE	BIGINT	表ディクショナリーのサイズ (バイト)。表に行コンプレッション・ディクショナリーが存在する場合に行の圧縮で使用されます。履歴ディクショナリーが存在する場合、この値は現在のディクショナリー・サイズと履歴ディクショナリー・サイズの合計になります。
BLOCKS_PENDING_CLEANUP	BIGINT	blocks_pending_cleanup - クリーンアップ保留中のロールアウト済みブロック・モニター・エレメント

表 285. ADMINTABINFO 管理ビューおよび ADMIN\_GET\_TAB\_INFO\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
STATSTYPE	CHAR(1)	<ul style="list-style-type: none"> <li>「F」 = 表または索引のスキャンを行わずにシステムが作り上げた統計。これらの統計はメモリーに保管され、システム・カタログに保管されているものとは異なります。これは一時状態であり、最終的に完全な統計が DB2 によって収集され、システム・カタログに保管されます。</li> <li>「A」 = システムが非同期に収集した統計。統計は DB2 によりバックグラウンド・プロセスで自動的に収集され、システム・カタログに保管されました。</li> <li>「S」 = システムが同期に収集した統計。統計は DB2 によって SQL ステートメントのコンパイル中に自動的に収集されました。これらの統計はメモリーに保管され、システム・カタログに保管されているものとは異なります。これは一時状態であり、最終的に DB2 が統計をシステム・カタログに保管します。</li> <li>「U」 = ユーザーが収集した統計。統計の収集は、RUNSTATS、CREATE INDEX、LOAD、REDISTRIBUTE などのユーティリティーを使用するか、システム・カタログ統計を手動で更新することによって、ユーザーが開始しました。</li> <li>NULL = 不明なタイプ</li> </ul>
XML_RECORD_TYPE	SMALLINT	<p>現在表で使用されている XML レコードのタイプを示します。</p> <ul style="list-style-type: none"> <li>タイプ 1 (シングル・ノード) XML レコード・フォーマットが使用されている場合には 1。</li> <li>タイプ 2 (マルチノード) XML レコード・フォーマットが使用されている場合には 2。</li> <li>表に XML 列がない場合には NULL。</li> </ul>
RECLAIMABLE_SPACE	BIGINT	<p>DMS 表スペースの MDC 表の場合、この値は、RECLAIM オプションを指定して REORG コマンドを実行することで再利用可能なディスク・スペース量を示します。ディスク・スペースは、K バイト (KB) で報告されます。その他の表の場合、この値はゼロです。</p>
XML_DICTIONARY_SIZE	BIGINT	<p>XML ストレージ・オブジェクトにデータ・コンプレッション・ディクショナリーが存在する場合、データ圧縮に使用される XML ディクショナリーのサイズ (バイト単位)。表に XML 列がない場合、またはコンプレッション・ディクショナリーが作成されていない場合、値は 0 になります。</p>

## AM\_BASE\_RPT\_RECOMS - アクティビティ・レポートに関する推奨事項

AM\_BASE\_RPT\_RECOMS 表関数は、アクティビティ・イベント・モニターで 사용되는アクティビティ・レポートに関する推奨事項を戻します。

## 構文

**重要:** 関連するアクティビティ・モニター・ルーチンは、バージョン 10.1 で非推奨となっており、将来のリリースで除去される可能性があります。詳しくは、「DB2 バージョン 10.1 の新機能」の『アクティビティ・モニター・ルーチンが推奨されなくなった』を参照してください。

```
▶▶—AM_BASE_RPT_RECOMS—(—report_id—,—client_locale—)————▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *report\_id*

レポート ID を指定する、タイプ INTEGER の入力引数。引数が NULL の場合、すべての使用可能なレポートの推奨事項が戻されます。

### *client\_locale*

クライアントの言語 ID を指定する、タイプ VARCHAR(33) の入力引数。引数が NULL または空ストリングである場合、デフォルト値は 'En\_US' (英語) です。指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、'En\_US' が使用されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

**例 1:** ID が 1 のアクティビティ・イベント・モニター・レポート用に推奨事項を要求します (英語)。デフォルトのクライアント言語 ID である 'En\_US' が前提となります。

```
SELECT *  
  FROM TABLE(SYSPROC.AM_BASE_RPT_RECOMS(1, CAST(NULL AS VARCHAR(33))))  
  AS RECOMS
```

**例 2:** ID が 12 のアクティビティ・イベント・モニター・レポート用に推奨事項を要求します (フランス語)。

```
SELECT *  
  FROM TABLE(SYSPROC.AM_BASE_RPT_RECOMS(12, CAST('Fr_FR' AS VARCHAR(33))))  
  AS RECOMS
```

## 戻される情報

表 286. AM\_BASE\_RPT\_RECOMS 表関数によって戻される情報

列名	データ・タイプ	説明
REPORT_ID	INTEGER	レポート ID。
RECOM_NAME	VARCHAR(256)	推奨事項の名前または簡略説明。
RECOM_DESCRIPTION	CLOB(32K)	推奨事項の詳細記述。

## AM\_BASE\_RPTS - アクティビティ・イベント・モニター・レポート

AM\_BASE\_RPTS 表関数は、アクティビティ・イベント・モニターで 사용되는アクティビティ・レポートを戻します。

### 構文

**重要:** 関連するアクティビティ・モニター・ルーチンは、バージョン 10.1 で非推奨となっており、将来のリリースで除去される可能性があります。詳しくは、「DB2 バージョン 10.1 の新機能」の『アクティビティ・モニター・ルーチンが推奨されなくなった』を参照してください。

▶▶ AM\_BASE\_RPTS(—report\_id—,—type—,—client\_locale—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### report\_id

固有のレポート ID を指定する、タイプ INTEGER の入力引数。引数が NULL の場合、レポート ID が示されたレポートが戻されます。

#### type

レポート・タイプを指定する、タイプ CHAR(4) の入力引数。有効な値は以下のとおりです。

APPL アプリケーション

STMT SQL ステートメント

TRAN トランザクション

CACH 動的 SQL ステートメント・キャッシュ

値は、大文字または小文字で指定できます。引数が NULL または空ストリングである場合、すべてのタイプのレポートが戻されます。

#### client\_locale

クライアントの言語 ID を指定する、タイプ VARCHAR(33) の入力引数。引数が NULL または空ストリングであるか、指定したロケールのメッセージ・ファイルがサーバーで使用できない場合には、'En\_US' が使用されます。



## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1:

```
SELECT * FROM TABLE(SYSPROC.AM_BASE_RPTS(CAST(NULL AS INTEGER),
      CAST(NULL AS CHAR(4)), CAST(NULL AS VARCHAR(33)))) AS REPORTS
```

例 2:

```
SELECT ID, NAME FROM TABLE(SYSPROC.AM_BASE_RPTS(
      CAST(NULL AS INTEGER), CAST('STMT' AS CHAR(4)), 'En_US'))
      AS REPORTS WHERE TYPE = 'STMT'
```

## 戻される情報

表 287. AM\_BASE\_RPTS 表関数によって戻される情報

列名	データ・タイプ	説明
ID	INTEGER	id - クラスター・キャッシング・ファシリティー ID モニター・エレメント
TYPE	CHAR(4)	レポート・タイプ。有効な値は、APPL、STMT、TRAN、CACH です。
NAME	VARCHAR(256)	レポートの名前または簡略説明。
DESCRIPTION	VARCHAR(16384)	レポートの詳細記述。
SWITCHES	VARCHAR(100)	このレポートに必要なモニター・スイッチ。

## AM\_DROP\_TASK - モニター・タスクの削除

AM\_DROP\_TASK プロシージャは、モニター・タスクを削除します。戻されるデータはありません。

## 構文

**重要:** 関連するアクティビティー・モニター・ルーチンは、バージョン 10.1 で非推奨となっており、将来のリリースで除去される可能性があります。詳しくは、

「DB2 バージョン 10.1 の新機能」の『アクティビティ・モニター・ルーチンが推奨されなくなった』を参照してください。

▶▶—AM\_DROP\_TASK—(—task\_id—)—————▶▶

スキーマは SYSPROC です。

## プロシージャ・パラメーター

*task\_id*

固有のモニター・タスク ID を指定する、タイプ INTEGER の入力引数。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

ID 5 のモニター・タスクをドロップします。

```
CALL SYSPROC.AM_DROP_TASK(5)
```

## AM\_GET\_LOCK\_CHN\_TB - 表形式のアプリケーション・ロック・チェーン・データの検索

AM\_GET\_LOCK\_CHN\_TB プロシージャは、アプリケーション・ロック・チェーン・データを、タブ区切りフォーマットで戻します。ロック・チェーンは、現行のアプリケーションが、直接的または間接的に、保留しているまたは待機しているすべてのアプリケーションで構成されます。

## 構文

**重要:** 関連するアクティビティ・モニター・ルーチンは、バージョン 10.1 で非推奨となっており、将来のリリースで除去される可能性があります。詳しくは、

「DB2 バージョン 10.1 の新機能」の『アクティビティ・モニター・ルーチンが推奨されなくなった』を参照してください。

▶▶—AM\_GET\_LOCK\_CHN\_TB—(—agent\_id—)—————▶▶

スキーマは SYSPROC です。

## プロシージャ・パラメーター

*agent\_id*

ロック・チェーン・データを検索する対象のアプリケーションのエージェント ID を指定する、タイプ BIGINT の入力引数。

### 許可

- SYSMON 権限
- AM\_GET\_LOCK\_CHN\_TB プロシージャに対する EXECUTE 特権。

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

エージェント ID 68 に関するロック・チェーン情報を検索します。

```
CALL SYSPROC.AM_GET_LOCK_CHN_TB(68)
```

### 戻される情報

このプロシージャは、以下の表を戻します。この表のそれぞれの行は、ロック待機の関係を表します。さらに、結果セットには、保留のみのアプリケーションごとに 1 つの行が示されます。ここでは、HOLDING\_AGENT\_ID 列が NULL で、他の 4 つの列は保留のみのアプリケーション用です。

表 288. AM\_GET\_LOCK\_CHN\_TB プロシージャによって戻される情報

列名	データ・タイプ	説明
HOLDING_AGENT_ID	BIGINT	ロックを保留しているアプリケーションのエージェント ID。
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID) モニター・エレメント
APPL_NAME	VARCHAR(255)	appl_name - アプリケーション名モニター・エレメント
AUTH_ID	VARCHAR(128)	auth_id - 許可 ID モニター・エレメント
APPL_ID	VARCHAR(64)	appl_id - アプリケーション ID モニター・エレメント

## AM\_GET\_LOCK\_CHNS - 特定のアプリケーションに関するロック・チェーン情報の検索

AM\_GET\_LOCK\_CHNS プロシージャは、指定したアプリケーションのロック・チェーンを、フォーマット済みストリングとして戻します。ロック・チェーンは、現行のアプリケーションが、直接的または間接的に、保留しているまたは待機しているすべてのアプリケーションで構成されます。

## 構文

**重要:** 関連するアクティビティ・モニター・ルーチンは、バージョン 10.1 で非推奨となっており、将来のリリースで除去される可能性があります。詳しくは、「DB2 バージョン 10.1 の新機能」の『アクティビティ・モニター・ルーチンが推奨されなくなった』を参照してください。

```
▶▶—AM_GET_LOCK_CHNS—(—agent_id—,—lock_chains—)————▶▶
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *agent\_id*

ロック・チェーンが表示されるアプリケーションのエージェント ID を指定する、タイプ BIGINT の入力引数。

### *lock\_chains*

指定したアプリケーションの全ロック・チェーンを表示する、タイプ CLOB(2M) の出力引数。

## 許可

- SYSMON 権限
- AM\_GET\_LOCK\_CHNS プロシージャに対する EXECUTE 特権。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
CALL SYSPROC.AM_GET_LOCK_CHNS(17,?)
```

```
Value of output parameters
```

```
-----
```

```
Parameter Name : LOCK_CHAINS
```

```
Parameter Value : >db2bp.exe (Agent ID: 17) (Auth ID: AMUSERC )
```

```
<db2bp.exe (Agent ID: 17) (Auth ID: AMUSERC )
```

```
<db2bp.exe (Agent ID: 18) (Auth ID: AMUSERB )
```

```
<db2bp.exe (Agent ID: 16) (Auth ID: AMUSERA )
```

```
Return Status = 0
```

## AM\_GET\_LOCK\_RPT - アプリケーション・ロックに関する詳細の検索

AM\_GET\_LOCK\_RPT プロシージャは、アプリケーションのロック詳細を、3 つの出力結果セットで戻します。

## 構文

**重要:** 関連するアクティビティ・モニター・ルーチンは、バージョン 10.1 で非推奨となっており、将来のリリースで除去される可能性があります。詳しくは、

「DB2 バージョン 10.1 の新機能」の『アクティビティ・モニター・ルーチンが推奨されなくなった』を参照してください。

▶—AM\_GET\_LOCK\_RPT—(—agent\_id—)————▶

スキーマは SYSPROC です。

## プロシージャ・パラメーター

*agent\_id*

ロック詳細が戻されるアプリケーションのエージェント ID を指定する、タイプ BIGINT の入力引数。

## 許可

- SYSMON 権限
- AM\_GET\_LOCK\_RPT プロシージャに対する EXECUTE 特権。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
CALL SYSPROC.AM_GET_LOCK_RPT(68)
```

## 使用上の注意

このプロシージャから情報が戻されるようにするには、DFT\_MON\_LOCK モニター・スイッチをオンにしておかなければなりません。

## 戻される情報

このプロシージャは、3 つの結果セット (アプリケーションの一般情報用、アプリケーションが保持するロック用、およびアプリケーションが待機中のロック用) を戻します。

表 289. AM\_GET\_LOCK\_RPT プロシージャから戻される一般アプリケーション情報

列名	データ・タイプ	説明
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
APPL_NAME	VARCHAR(256)	appl_name - アプリケーション名
PRIMARY_AUTH_ID	VARCHAR(128)	auth_id - 許可 ID
APPL_ID	VARCHAR(128)	appl_id - アプリケーション ID

表 289. AM\_GET\_LOCK\_RPT プロシージャから戻される一般アプリケーション情報 (続き)

列名	データ・タイプ	説明
APPL_STATUS	VARCHAR(22)	<p>appl_status - アプリケーション状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• BACKUP</li> <li>• COMMIT_ACT</li> <li>• COMP</li> <li>• CONNECTED</li> <li>• CONNECTPEND</li> <li>• CREATE_DB</li> <li>• DECOUPLED</li> <li>• DISCONNECTPEND</li> <li>• INTR</li> <li>• IOERROR_WAIT</li> <li>• LOAD</li> <li>• LOCKWAIT</li> <li>• QUIESCE_TABLESPACE</li> <li>• RECOMP</li> <li>• REMOTE_RQST</li> <li>• RESTART</li> <li>• RESTORE</li> <li>• ROLLBACK_ACT</li> <li>• ROLLBACK_TO_SAVEPOINT</li> <li>• TEND</li> <li>• THABRT</li> <li>• THCOMT</li> <li>• TPREP</li> <li>• UNLOAD</li> <li>• UOWEXEC</li> <li>• UOWWAIT</li> <li>• WAITFOR_REMOTE</li> </ul>
COORD_NODE_NUM	SMALLINT	coord_node - コーディネーター・ノード
SEQUENCE_NO	VARCHAR(4)	sequence_no - シーケンス番号
CLIENT_PRDID	VARCHAR(128)	client_prdid - クライアント製品/バージョン ID
CLIENT_PID	BIGINT	client_pid - クライアント・プロセス ID

表 289. AM\_GET\_LOCK\_RPT プロシージャから戻される一般アプリケーション情報 (続き)

列名	データ・タイプ	説明
CLIENT_PLATFORM	VARCHAR(12)	<p>client_platform - クライアント・オペレーティング・プラットフォーム。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。</p> <ul style="list-style-type: none"> <li>• AIX</li> <li>• AIX64</li> <li>• AS400_DRDA</li> <li>• DOS</li> <li>• DYNIX</li> <li>• HP</li> <li>• HP64</li> <li>• HPIA</li> <li>• HPIA64</li> <li>• LINUX</li> <li>• LINUX390</li> <li>• LINUXIA64</li> <li>• LINUXPPC</li> <li>• LINUXPPC64</li> <li>• LINUXX8664</li> <li>• LINUXZ64</li> <li>• MAC</li> <li>• MVS_DRDA</li> <li>• NT</li> <li>• NT64</li> <li>• OS2</li> <li>• OS390</li> <li>• SCO</li> <li>• SGI</li> <li>• SNI</li> <li>• SUN</li> <li>• SUN64</li> <li>• 不明 (UNKNOWN)</li> <li>• UNKNOWN_DRDA</li> <li>• VM_DRDA</li> <li>• VSE_DRDA</li> <li>• WINDOWS</li> <li>• WINDOWS95</li> </ul>

表 289. AM\_GET\_LOCK\_RPT プロシージャから戻される一般アプリケーション情報 (続き)

列名	データ・タイプ	説明
CLIENT_PROTOCOL	VARCHAR(10)	client_protocol - クライアント通信 プロトコル。このインターフェース は、sqlmon.h での定義に基づくテ キスト ID を戻します。  <ul style="list-style-type: none"> <li>• CPIC</li> <li>• LOCAL</li> <li>• NPIPE</li> <li>• TCPIP (DB2 Universal Database、 または DB2 UDB の場合)</li> <li>• TCPIP4</li> <li>• TCPIP6</li> </ul>
CLIENT_NNAME	VARCHAR(128)	client_nname - クライアント名モニ ター・エレメント
LOCKS_HELD	BIGINT	locks_held - ロック保持数
LOCK_WAIT_START_TIME	TIMESTAMP	lock_wait_start_time - ロック待機開 始タイム・スタンプ
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時 間
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
LOCK_TIMEOUTS	BIGINT	lock_timeouts - ロック・タイムアウト数
LOCK_ESCALS	BIGINT	lock_escals - ロック・エスカレーシ ョン数
X_LOCK_ESCALS	BIGINT	x_lock_escals - 排他ロック・エスカ レーション数
DEADLOCKS	BIGINT	deadlocks - デッドロック検出数

表 290. AM\_GET\_LOCK\_RPT プロシージャから戻されるロック保持情報

列名	データ・タイプ	説明
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TABNAME	VARCHAR(128)	table_name - 表名



表 290. AM\_GET\_LOCK\_RPT プロシージャから戻されるロック保持情報 (続き)

列名	データ・タイプ	説明
LOCK_OBJECT_TYPE	VARCHAR(18)	<p>lock_object_type - 待機中のロック対象タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• AUTORESIZE_LOCK</li> <li>• AUTOSTORAGE_LOCK</li> <li>• BLOCK_LOCK</li> <li>• EOT_LOCK</li> <li>• INPLACE_REORG_LOCK</li> <li>• INTERNAL_LOCK</li> <li>• INTERNALB_LOCK</li> <li>• INTERNALC_LOCK</li> <li>• INTERNALJ_LOCK</li> <li>• INTERNALL_LOCK</li> <li>• INTERNALO_LOCK</li> <li>• INTERNALQ_LOCK</li> <li>• INTERNALP_LOCK</li> <li>• INTERNALS_LOCK</li> <li>• INTERNALT_LOCK</li> <li>• INTERNALV_LOCK</li> <li>• KEYVALUE_LOCK</li> <li>• ROW_LOCK</li> <li>• SYSBOOT_LOCK</li> <li>• TABLE_LOCK</li> <li>• TABLE_PART_LOCK</li> <li>• TABLESPACE_LOCK</li> <li>• XML_PATH_LOCK</li> </ul>

表 290. AM\_GET\_LOCK\_RPT プロシージャから戻されるロック保持情報 (続き)

列名	データ・タイプ	説明
LOCK_MODE	VARCHAR(10)	lock_mode - ロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
LOCK_STATUS	VARCHAR(10)	lock_status - ロック状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• CONV</li> <li>• GRNT</li> </ul>
LOCK_ESCALATION	SMALLINT	lock_escalation - ロック・エスカレーション
LOCK_NAME	VARCHAR(32)	lock_name - ロック名
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

表 291. AM\_GET\_LOCK\_RPT プロシージャから戻されるロック待機情報

列名	データ・タイプ	説明
AGENT_ID_HOLDING_LK	BIGINT	agent_id_holding_lock - ロックを保持しているエージェント ID
APPL_ID_HOLDING_LK	VARCHAR(128)	appl_id_holding_lk - ロックを保持しているアプリケーション ID
LOCK_WAIT_START_TIME	TIMESTAMP	lock_wait_start_time - ロック待機開始タイム・スタンプ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TABSHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TABNAME	VARCHAR(128)	table_name - 表名

表 291. AM\_GET\_LOCK\_RPT プロシージャから戻されるロック待機情報 (続き)

列名	データ・タイプ	説明
LOCK_OBJECT_TYPE	VARCHAR(18)	<p>lock_object_type - 待機中のロック対象タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• AUTORESIZE_LOCK</li> <li>• AUTOSTORAGE_LOCK</li> <li>• BLOCK_LOCK</li> <li>• EOT_LOCK</li> <li>• INPLACE_REORG_LOCK</li> <li>• INTERNAL_LOCK</li> <li>• INTERNALB_LOCK</li> <li>• INTERNALC_LOCK</li> <li>• INTERNALJ_LOCK</li> <li>• INTERNALL_LOCK</li> <li>• INTERNALO_LOCK</li> <li>• INTERNALQ_LOCK</li> <li>• INTERNALP_LOCK</li> <li>• INTERNALS_LOCK</li> <li>• INTERNALT_LOCK</li> <li>• INTERNALV_LOCK</li> <li>• KEYVALUE_LOCK</li> <li>• ROW_LOCK</li> <li>• SYSBOOT_LOCK</li> <li>• TABLE_LOCK</li> <li>• TABLE_PART_LOCK</li> <li>• TABLESPACE_LOCK</li> <li>• XML_PATH_LOCK</li> </ul>

表 291. AM\_GET\_LOCK\_RPT プロシージャから戻されるロック待機情報 (続き)

列名	データ・タイプ	説明
LOCK_MODE	VARCHAR(10)	lock_mode - ロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
LOCK_MODE_REQUESTED	VARCHAR(10)	lock_mode_requested - 要求されているロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
LOCK_ESCALATION	SMALLINT	lock_escalation - ロック・エスカレーション

## AM\_GET\_RPT - アクティビティ・モニター・データの検索

AM\_GET\_RPT プロシージャは、レポートのアクティビティ・モニター・データを戻します。

### 構文

**重要:** 関連するアクティビティ・モニター・ルーチンは、バージョン 10.1 で非推奨となっており、将来のリリースで除去される可能性があります。詳しくは、

「DB2 バージョン 10.1 の新機能」の『アクティビティ・モニター・ルーチンが推奨されなくなった』を参照してください。

```
▶▶—AM_GET_RPT—(—partition—,—report_id—,—appl_filter—,—max_number—)————▶▶
```

スキーマは SYSPROC です。

## プロシージャ・パラメーター

### *partition*

データベース・パーティション番号を指定するタイプ INTEGER の入力引数。  
有効な値は、-2 (すべてのデータベース・パーティションの表示) と、既存のデータベース・パーティションのデータベース・パーティション番号です。

### *report\_id*

固有のレポート ID を指定する、タイプ INTEGER の入力引数。

### *appl\_filter*

アプリケーション・フィルターを指定する、タイプ CLOB(32K) の入力引数。  
アプリケーション・フィルターとは、3 つの列 (AGENT\_ID、APPL\_NAME、および PRIMARY\_AUTH\_ID) の一部またはすべてが関係する検索条件のことです。ここで、AGENT\_ID と PRIMARY\_AUTH\_ID は整数で、APPL\_NAME は文字ストリングです。引数が NULL または空ストリングである場合、フィルター操作は実行されません。

### *max\_number*

表示するアプリケーション、ステートメント、またはトランザクションの最大数を指定する、タイプ INTEGER の入力引数。引数が NULL の場合、すべてのアプリケーション、ステートメント、およびトランザクションが表示されます。

## 許可

- SYSMON 権限
- AM\_GET\_RPT プロシージャに対する EXECUTE 特権。

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
CALL SYSPROC.AM_GET_RPT(-2, 18,  
  CAST('AGENT_ID=29 AND PRIMARY_AUTH_ID <> 'dbuser' AND APPL_NAME LIKE 'db2%''  
  AS CLOB(32K)), 100)
```

## 使用上の注意

戻される結果セットは、各レポート ID ごとに異なります。このプロシージャは、廃止されたアクティビティ・モニター・グラフィック・ツールをサポートするためのプロシージャです。解析できるレポートを作成するには、このプロシージャではなく、スナップショット管理 SQL ルーチンおよびビューを使用する必要があります。このプロシージャを使用するには、DFT\_MON\_LOCK モニター・スイッチをオンにしておかなければなりません。

## AM\_SAVE\_TASK - モニター・タスクの作成または変更

AM\_SAVE\_TASK プロシージャは、モニター・タスクを作成または変更します。

### 構文

**重要:** 関連するアクティビティ・モニター・ルーチンは、バージョン 10.1 で非推奨となっており、将来のリリースで除去される可能性があります。詳しくは、「DB2 バージョン 10.1 の新機能」の『アクティビティ・モニター・ルーチンが推奨されなくなった』を参照してください。

```
▶▶ AM_SAVE_TASK (—mode—, —task_id—, —task_name—, —appl_filter—, —————▶▶
▶—show_lock_chains—, —report_ids—) —————▶▶
```

スキーマは SYSPROC です。

### プロシージャ・パラメーター

#### *mode*

新規モニター・タスクを作成する ('C') か既存のモニター・タスクを変更する ('M') かを指定する、タイプ CHAR(1) の入力引数。

#### *task\_id*

固有のモニター・タスク ID を指定する、タイプ INTEGER の入力引数。 *mode* が 'C' である場合、*task\_id* の指定された入力は無視されます。新規モニター・タスクの ID は、プロシージャによって生成されて出力に戻されます。 *mode* が 'M' である場合、変更されるモニター・タスクの ID が指定されます。

#### *task\_name*

モニター・タスクの名前または簡略説明を指定する、タイプ VARCHAR(128) の入力引数。

#### *appl\_filter*

アプリケーション・フィルターを指定する、タイプ CLOB(32K) の入力引数。アプリケーション・フィルターとは、3 つの列 (AGENT\_ID、APPL\_NAME、および AUTH\_ID) の一部またはすべてが関係する検索条件のことです。ここで、AGENT\_ID と AUTH\_ID は整数で、APPL\_NAME は文字ストリングです。引数が NULL または空ストリングである場合、フィルター操作は実行されません。

#### *show\_lock\_chains*

ロック・チェーンを表示するかどうかを指定する、タイプ CHAR(1) の入力引数。有効な値は 'Y' および 'N' です。引数が NULL の場合、ロック・チェーンは表示されません。

#### *report\_ids*

1 つ以上のレポート ID をコンマで区切って指定する、タイプ VARCHAR(3893) の入力引数。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権

- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

#### 例

例:

```
CALL SYSPROC.AM_SAVE_TASK('M',11,'Task ABC',CAST (NULL AS CLOB(32K)),  
    'N','1,2,4,8,9,12')
```

## APPLICATION\_ID

APPLICATION\_ID 関数は、現行接続のアプリケーション ID を戻します。

この結果のデータ・タイプは VARCHAR(128) です。

この関数から戻される値は 100 年のインターバル内で固有であり、関数を呼び出す前に確立された接続の期間中だけ有効です。

**注:** このスカラー関数は使用すべきではなく、MON\_GET\_APPLICATION\_ID スカラー関数に置き換えられました。

#### 構文

▶▶ APPLICATION\_ID (—) ◀◀

スキーマは SYSFUN です。

#### 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

#### 例

```
SELECT APPLICATION_ID() AS APPL_ID FROM SYSIBM.SYSDUMMY1
```

## DB\_PARTITIONS

DB\_PARTITIONS 表関数は、表形式の db2nodes.cfg ファイルの内容を戻します。

注: この表関数は使用すべきではなく、DB2\_MEMBER 管理ビュー、DB2\_CF 管理ビュー、および DB2\_GET\_INSTANCE\_INFO 表関数に置き換えられました。

### 構文

▶▶—DB\_PARTITIONS—(—)—————▶▶

スキーマは SYSPROC です。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 表関数パラメーター

関数には入力パラメーターはありません。

### 例

4 つのメンバーを持つパーティション・データベース・インスタンスから情報を取り出します。

```
SELECT * FROM TABLE(DB_PARTITIONS()) as T
```

以下はこの照会の出力例です。

```
PARTITION_NUMBER HOST_NAME PORT_NUMBER SWITCHNAME
-----
          0 so1          0 so1-ib0
          1 so2          0 so2-ib0
          2 so3          0 so3-ib0
          3 so4          0 so4-ib0
```

4 record(s) selected.

DB2 pureScale 環境では、3 つのメンバーおよび 1 つのクラスター・キャッシング・ファシリティー DB2 pureScale インスタンスから情報を取り出します。

```
SELECT * FROM TABLE(DB_PARTITIONS()) as T
```

以下はこの照会の出力例です。



```

PARTITION_NUMBER HOST_NAME  PORT_NUMBER SWITCHNAME
-----
                0 so1          0 so1-ib0
                0 so2          0 so2-ib0
                0 so3          0 so3-ib0

```

3 record(s) selected.

## 使用上の注意

DB2 Enterprise Server Edition およびパーティション・データベース環境では、DB\_PARTITIONS 表関数は、db2nodes.cfg ファイル内の項目ごとに 1 行ずつ戻します。

DB2 pureScale 環境では、DB\_PARTITIONS 表関数は、以下の情報が列に入った複数の行を返します。

- PARTITION\_NUMBER 列は常に 0 です。
- 残りの列は、db2nodes.cfg ファイル内の、現在のメンバーの項目に関する情報を表示します。

## 戻される情報

表 292. DB\_PARTITIONS 表関数によって戻される情報

列名	データ・タイプ	説明
PARTITION_NUMBER	SMALLINT	partition_number - パーティション番号モニター・エレメント
HOST_NAME	VARCHAR(256)	host_name - ホスト名モニター・エレメント
PORT_NUMBER	SMALLINT	データベース・パーティション・サーバーのポート番号。
SWITCH_NAME	VARCHAR(128)	データベース・パーティション通信用の高速相互接続 (スイッチ) の名前。

## GET\_DB\_CONFIG

GET\_DB\_CONFIG プロシージャは、データベース構成情報を戻します。

このプロシージャは引数を取りません。

注: このプロシージャは使用すべきではなく、388 ページの『DBCFCG 管理ビュー および DB\_GET\_CFG 表関数 - データベース構成パラメーター情報の取得』によって置き換えられました。

▶▶ GET\_DB\_CONFIG(—) ◀◀

スキーマは SYSPROC です。

このプロシージャは、パラメーターごとに 1 つの列を備えた 2 つの行を示す単一の結果セットを戻します。最初の列は、以下の表に示すように DBCONFIG\_TYPE という名前です。

表 293. GET\_DB\_CONFIG プロシージャによって戻される情報

列名	データ・タイプ	説明
DBCONFIG_TYPE	INTEGER	この列の値が 0 の行には、ディスクに保管されたデータベース構成パラメーターの値が入ります。この列の値が 1 の行には、メモリーに保管されたデータベース構成パラメーターの現行値が入ります。

このプロシージャは、結果セットを保管する DB\_CONFIG という名前のグローバル一時表を作成するために使用する、USER TEMPORARY 表スペースを必要とします。

## 許可

このプロシージャを実行するには、以下の権限のいずれかが必要です。

- プロシージャでの EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

コマンド行プロセッサ (CLP) を使用して、**logarchmeth1** データベース構成パラメーターの値を変更します。オリジナルの (ディスク上の) 値と、更新済みの (メモリー内の) 値を GET\_DB\_CONFIG プロシージャを呼び出して検索し、次いで結果のグローバル一時表 (DB\_CONFIG) を照会します。

```
CONNECT TO SAMPLE

CREATE BUFFERPOOL MY8KPOOL SIZE 250 PAGESIZE 8K

CREATE USER TEMPORARY TABLESPACE MYTSP2 PAGESIZE
      8K MANAGED BY SYSTEM USING ( 'TSC2' ) BUFFERPOOL MY8KPOOL

UPDATE DB CFG USING LOGARCHMETH1 USEREXIT

CALL SYSPROC.GET_DB_CONFIG()

SELECT DBCONFIG_TYPE, LOGARCHMETH1
      FROM SESSION.DB_CONFIG

CONNECT RESET
```

以下はこの照会の出力例です。

```

DBCONFIG_TYPE LOGARCHMETH1
-----
              0          1
              1          0

```

2 record(s) selected.

## GET\_DBM\_CONFIG

GET\_DBM\_CONFIG 関数は、データベース・マネージャー構成情報を戻します。

この関数は引数を取りません。

注: この表関数は使用すべきではなく、392 ページの『DBMCFG 管理ビュー - データベース・マネージャー構成パラメーター情報の検索』に置き換えられました。

▶▶ GET\_DBM\_CONFIG (—) ◀◀

スキーマは SYSFUN です。

### 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、パラメーターごとに 1 つの列を備えた 2 つの行を示す表を戻します。最初の列は、以下の表に示すように DBMCONFIG\_TYPE という名前です。

表 294. GET\_DBM\_CONFIG 表関数によって戻される情報

列名	データ・タイプ	説明
DBMCONFIG_TYPE	INTEGER	この列の値が 0 の行には、ディスクに保管されたデータベース・マネージャー構成パラメーターの値が入ります。この列の値が 1 の行には、メモリーに保管されたデータベース・マネージャー構成パラメーターの現行値が入ります。

### 例

コマンド行プロセッサ (CLP) を使用し、numdb および diaglevel データベース・マネージャー構成パラメーターの値を変更し、次いでオリジナルの (ディスク上の) 値と、更新された (メモリー内の) 値を検索します。

```

UPDATE DBM CFG USING NUMDB 32 DIAGLEVEL 4

CONNECT TO SAMPLE

SELECT DBMCONFIG_TYPE, NUMDB, DIAGLEVEL
FROM TABLE(SYSFUN.GET_DBM_CONFIG()) AS DBMCFG

CONNECT RESET

```

以下はこの照会の出力例です。

DBMCONFIG_TYPE	NUMDB	DIAGLEVEL
0	32	4
1	8	3

2 record(s) selected.

## ENV\_SYS\_RESOURCES 管理ビュー - システム情報を戻す

ENV\_SYS\_RESOURCES 管理ビューは、オペレーティング・システム、CPU、メモリー、およびその他のシステム関連情報を戻します。

**注:** DB2 バージョン 9.8 フィックスパック 4 の時点では、ENV\_SYS\_RESOURCES 管理ビューおよび関連する ENV\_GET\_SYS\_RESOURCES 表関数は推奨されないため、ENV\_GET\_SYSTEM\_RESOURCES 表関数に置き換えられています。

ENV\_SYS\_RESOURCES 管理ビューは、オペレーティング・システム、CPU、メモリー、およびその他のシステム関連情報を戻します。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- ENV\_SYS\_RESOURCES 管理ビューに対する SELECT 特権
- ENV\_SYS\_RESOURCES 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

```

SELECT SUBSTR(NAME,1,20) AS NAME, SUBSTR(VALUE,1,10) AS VALUE,
SUBSTR(DATATYPE,1,10) AS DATATYPE, DBPARTITIONNUM
FROM SYSIBMADM.ENV_SYS_RESOURCES
WHERE SUBSTR(NAME,1,8)='CPU_LOAD' OR NAME='CPU_USAGE_TOTAL'

```

以下はこの照会の出力例です。

NAME	VALUE	DATATYPE	DBPARTITIONNUM
CPU_LOAD_SHORT	0.044052	DECIMAL	0
CPU_LOAD_MEDIUM	0.087250	DECIMAL	0

CPU_LOAD_LONG	0.142059	DECIMAL	0
CPU_USAGE_TOTAL	7	SMALLINT	0

4 record(s) selected.

## ENV\_SYS\_RESOURCES 管理ビューのメタデータ

表 295. ENV\_SYS\_RESOURCES 管理ビューのメタデータ

列名	データ・タイプ	説明
NAME	VARCHAR(128)	属性の名前。考えられる値については、表 296を参照してください。 注: サーバーでのオペレーティング・システムおよびハードウェア構成によっては、一部の属性が使用できない場合があります。
VALUE	VARCHAR(1024)	属性の値。
DATATYPE	VARCHAR(128)	属性データ・タイプ。
UNIT	VARCHAR(128)	VALUE 列に使用される単位 (適用される場合)。適用外の場合は NULL が戻されます。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

表 296. NAME 列に使用できる値

情報タイプ	名前	データ・タイプ	説明	この情報を戻すプラットフォーム	UNIT
オペレーティング・システム	OS_NAME	VARCHAR(256)	オペレーティング・システム・ソフトウェアの名前。	すべて	NULL
	HOST_NAME	VARCHAR(256)	システムのホスト名。	すべて	NULL
	OS_VERSION	VARCHAR(256)	オペレーティング・システムのバージョン。例えば、AIX: 4.3 version = 4 のようになります。	すべて	NULL
	OS_RELEASE	VARCHAR(256)	オペレーティング・システムのリリース。例えば、AIX: 4.3 release = 3 のようになります。	すべて	NULL
	MACHINE_IDENTIFICATION	VARCHAR(256)	マシン・ハードウェア ID。	すべて	NULL
	OS_LEVEL	VARCHAR(256)	現行バージョンおよびリリースの保守レベル。例えば、LINUX: 2.4.9, level = 9 のようになります。	Linux	NULL

表 296. NAME 列に使用できる値 (続き)

情報タイプ	名前	データ・タイプ	説明	この情報を戻すプラットフォーム	UNIT
CPU	CPU_TOTAL	BIGINT	CPU の総数。	すべて	NULL
	CPU_ONLINE	BIGINT	オンラインの CPU 数。	すべて	NULL
	CPU_CONFIGURED	BIGINT	構成済みの CPU 数。	すべて	NULL
	CPU_SPEED	BIGINT	CPU の速度。	すべて	MHz
	CPU_TIMEBASE	BIGINT	時間基準のレジスタ増分の周波数。	Linux PowerPC®	Hz
	CPU_HMT_DEGREE	BIGINT	ハードウェア・マルチスレッド化 (HMT) をサポートするシステムでは、これは物理プロセッサがオペレーティング・システムに存在すると見なすプロセッサの数です。非 HMT システムでは、この値は 1 です。HMT システムでは、「合計」は論理 CPU の数を反映します。物理 CPU の数を取得するには、「合計」を「threadingDegree」で除算します。	すべて	NULL
	CPU_CORES_PER_SOCKET	BIGINT	ソケットごとの CPU コアの数。単一コア・システムでは、この値は 1 です。	すべて	NULL
物理メモリー	MEMORY_TOTAL	BIGINT	物理メモリーの合計サイズ。	すべて	MB
	MEMORY_FREE	BIGINT	空き物理メモリーの量。	すべて	MB
	MEMORY_SWAP_TOTAL	BIGINT	スワップ・スペースの合計量。	すべて	MB
	MEMORY_SWAP_FREE	BIGINT	空きスワップ・スペースの量。	すべて	MB

表 296. NAME 列に使用できる値 (続き)

情報タイプ	名前	データ・タイプ	説明	この情報を戻すプラットフォーム	UNIT
仮想メモリー	VIRTUAL_MEM_TOTAL	BIGINT	システム上の仮想メモリーの合計量。	すべて	MB
	VIRTUAL_MEM_RESERVED	BIGINT	予約済み仮想メモリーの量。	すべて	MB
	VIRTUAL_MEM_FREE	BIGINT	空き仮想メモリーの量。	すべて	MB
CPU ロード	CPU_LOAD_SHORT	DECIMAL	最短期間。例えば、最後の 5 分間でサンプルをロードします。	Windows オペレーティング・システムを除くすべて。	NULL
	CPU_LOAD_MEDIUM	DECIMAL	中間の期間。例えば、最後の 10 分間でサンプルをロードします。	Windows オペレーティング・システムを除くすべて。	NULL
	CPU_LOAD_LONG	DECIMAL	長期間。例えば、最後の 15 分間でサンプルをロードします。	Windows オペレーティング・システムを除くすべて。	NULL
	CPU_USAGE_TOTAL	SMALLINT	マシンの全 CPU 使用量のパーセント。	すべて	パーセント

## LOCKS\_HELD 管理ビュー - 保持されているロックに関する情報の検索

LOCKS\_HELD 管理ビューは、現在保持されているロックに関する情報を戻します。

注: この管理ビューは非推奨になり、以下のものに置き換えられました。すなわち、526 ページの『MON\_GET\_APPL\_LOCKWAIT - アプリケーションが待機しているロックについての情報の取得』、598 ページの『MON\_GET\_LOCKS - 現在接続されているデータベース内のすべてのロックのリスト』、および 482 ページの『MON\_FORMAT\_LOCK\_NAME - 内部ロック名のフォーマット設定と詳細の出力』。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- LOCKS\_HELD 管理ビューに対する SELECT 特権
- LOCKS\_HELD 管理ビューに対する CONTROL 特権

- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

例 1: データベース SAMPLE の各表で保持されているロックの総数をリストします。

```
SELECT TABSCHEMA, TABNAME, COUNT(*) AS NUMBER_OF_LOCKS_HELD
FROM SYSIBMADM.LOCKS_HELD WHERE DB_NAME = 'SAMPLE'
GROUP BY DBPARTITIONNUM, TABSCHEMA, TABNAME
```

以下はこの照会の出力例です。

TABSCHEMA	TABNAME	NUMBER_OF_LOCKS_HELD
JESSICAE	EMPLOYEE	5
JESSICAE	EMP_RESUME	1
JESSICAE	ORG	3

例 2: 現在接続されているデータベース SAMPLE 内の、エスカレートされていないすべてのロックをリストします。

```
SELECT AGENT_ID, TABSCHEMA, TABNAME, LOCK_OBJECT_TYPE, LOCK_MODE,
LOCK_STATUS FROM SYSIBMADM.LOCKS_HELD WHERE LOCK_ESCALATION = 0
AND DBPARTITIONNUM = 0
```

以下はこの照会の出力例です。

AGENT_ID	TABSCHEMA	TABNAME	LOCK_OBJECT_TYPE	LOCK_MODE	LOCK_STATUS
680	JESSICAE	EMPLOYEE	INTERNALV_LOCK	S	GRNT
680	JESSICAE	EMPLOYEE	INTERNALP_LOCK	S	GRNT

例 3: エージェント ID 310 を持つアプリケーションで現在保持されているロックに関するロック情報をリストします。

```
SELECT TABSCHEMA, TABNAME, LOCK_OBJECT_TYPE, LOCK_MODE, LOCK_STATUS,
LOCK_ESCALATION FROM SYSIBMADM.LOCKS_HELD WHERE AGENT_ID = 310
```

以下はこの照会の出力例です。

TABSCHEMA	TABNAME	LOCK_OBJECT_TYPE	LOCK_MODE	LOCK_STATUS
JESSICAE	EMP_RESUME	TABLE_LOCK	S	GRNT
JESSICAE	EMPLOYEE	ROW_LOCK	S	GRNT



## 戻される情報

表 297. LOCKS\_HELD 管理ビューによって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	レポートの生成日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
APPL_NAME	VARCHAR(256)	appl_name - アプリケーション名
AUTHID	VARCHAR(128)	auth_id - 許可 ID
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TABSHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TABNAME	VARCHAR(128)	table_name - 表名
TAB_FILE_ID	BIGINT	table_file_id - 表ファイル ID
LOCK_OBJECT_TYPE	VARCHAR(18)	lock_object_type - 待機中のロック対象タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• AUTORESIZE_LOCK</li> <li>• AUTOSTORAGE_LOCK</li> <li>• BLOCK_LOCK</li> <li>• EOT_LOCK</li> <li>• INPLACE_REORG_LOCK</li> <li>• INTERNAL_LOCK</li> <li>• INTERNALB_LOCK</li> <li>• INTERNALC_LOCK</li> <li>• INTERNALJ_LOCK</li> <li>• INTERNALL_LOCK</li> <li>• INTERNALO_LOCK</li> <li>• INTERNALQ_LOCK</li> <li>• INTERNALP_LOCK</li> <li>• INTERNALS_LOCK</li> <li>• INTERNALT_LOCK</li> <li>• INTERNALV_LOCK</li> <li>• KEYVALUE_LOCK</li> <li>• ROW_LOCK</li> <li>• SYSBOOT_LOCK</li> <li>• TABLE_LOCK</li> <li>• TABLE_PART_LOCK</li> <li>• TABLESPACE_LOCK</li> <li>• XML_PATH_LOCK</li> </ul>
LOCK_NAME	VARCHAR(32)	lock_name - ロック名

表 297. LOCKS\_HELD 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_MODE	VARCHAR(10)	lock_mode - ロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
LOCK_STATUS	VARCHAR(10)	lock_status - ロック状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• CONV</li> <li>• GRNT</li> </ul>
LOCK_ESCALATION	SMALLINT	lock_escalation - ロック・エスカレーション
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## LOCKWAITS 管理ビュー - 現行のロック待機情報の検索

LOCKWAITS 管理ビューは、ロック取得待機中のアプリケーションのために作業を代行している DB2 エージェントに関する情報を戻します。

注: この管理ビューは非推奨になり、以下のものに置き換えられました。すなわち、758 ページの『MON\_LOCKWAITS 管理ビュー - ロックの取得を待機しているアプリケーションに関するメトリックの取得』。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- LOCKWAITS 管理ビューに対する SELECT 特権

- LOCKWAITS 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

例 1: エージェント ID 89 を持つアプリケーションのすべてのロック待機に関する情報をリストします。

```
SELECT SUBSTR(TABSCHEMA,1,8) AS TABSCHEMA, SUBSTR(TABNAME,1,15) AS TABNAME,
       LOCK_OBJECT_TYPE, LOCK_MODE, LOCK_MODE_REQUESTED, AGENT_ID_HOLDING_LK
FROM SYSIBMADM.LOCKWAITS WHERE AGENT_ID = 89
```

以下はこの照会の出力例です。

TABSCHEMA	TABNAME	LOCK_OBJECT_TYPE	LOCK_MODE	...
JESSICAE	T1	ROW_LOCK	X	...

1 record(s) selected.

この照会の出力 (続き)。

...	LOCK_MODE_REQUESTED	AGENT_ID_HOLDING_LK
...	NS	7

例 2: データベース SAMPLE における、表ごとの未解決ロック要求の総数をリストします。要求の数で出力をソートすると、競合が最も多い表を識別できます。

```
SELECT SUBSTR(TABSCHEMA,1,8) AS TABSCHEMA, SUBSTR(TABNAME, 1, 15)
       AS TABNAME, COUNT(*) AS NUM_OF_LOCK_REQUESTS_WAITING,
       DBPARTITIONNUM
FROM SYSIBMADM.LOCKWAITS WHERE DB_NAME = 'SAMPLE'
GROUP BY TABSCHEMA, TABNAME, DBPARTITIONNUM
ORDER BY NUM_OF_LOCK_REQUESTS_WAITING DESC
```

以下はこの照会の出力例です。

TABSCHEMA	TABNAME	NUM_OF_LOCK_REQUESTS_WAITING	DBPARTITIONNUM
JESSICAE	T3	2	0
JESSICAE	T1	1	0
JESSICAE	T2	1	0

3 record(s) selected.

## 戻される情報

表 298. LOCKWAITS 管理ビューによって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	レポートの生成日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
APPL_NAME	VARCHAR(256)	appl_name - アプリケーション名
AUTHID	VARCHAR(128)	auth_id - 許可 ID
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TABSHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TABNAME	VARCHAR(128)	table_name - 表名
SUBSECTION_NUMBER	BIGINT	ss_number - サブセクション番号
LOCK_OBJECT_TYPE	VARCHAR(18)	lock_object_type - 待機中のロック対象タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• AUTORESIZE_LOCK</li> <li>• AUTOSTORAGE_LOCK</li> <li>• BLOCK_LOCK</li> <li>• EOT_LOCK</li> <li>• INPLACE_REORG_LOCK</li> <li>• INTERNAL_LOCK</li> <li>• INTERNALB_LOCK</li> <li>• INTERNALC_LOCK</li> <li>• INTERNALJ_LOCK</li> <li>• INTERNALL_LOCK</li> <li>• INTERNALO_LOCK</li> <li>• INTERNALQ_LOCK</li> <li>• INTERNALP_LOCK</li> <li>• INTERNALS_LOCK</li> <li>• INTERNALT_LOCK</li> <li>• INTERNALV_LOCK</li> <li>• KEYVALUE_LOCK</li> <li>• ROW_LOCK</li> <li>• SYSBOOT_LOCK</li> <li>• TABLE_LOCK</li> <li>• TABLE_PART_LOCK</li> <li>• TABLESPACE_LOCK</li> <li>• XML_PATH_LOCK</li> </ul>

表 298. LOCKWAITS 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_WAIT_START_TIME	TIMESTAMP	lock_wait_start_time - ロック待機開始タイム・スタンプ
LOCK_NAME	VARCHAR(32)	lock_name - ロック名
LOCK_MODE	VARCHAR(10)	lock_mode - ロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
LOCK_MODE_REQUESTED	VARCHAR(10)	lock_mode_requested - 要求されているロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
AGENT_ID_HOLDING_LK	BIGINT	agent_id_holding_lock - ロックを保持しているエージェント ID
APPL_ID_HOLDING_LK	VARCHAR(128)	appl_id_holding_lk - ロックを保持しているアプリケーション ID
LOCK_ESCALATION	SMALLINT	lock_escalation - ロック・エスカレーション

表 298. LOCKWAITS 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## ヘルス・スナップショット・ルーチン

### HEALTH\_CONT\_HI

HEALTH\_CONT\_HI 表関数は、表スペース・コンテナのヘルス・インディケーター情報を、データベースの表スペースのヘルス・スナップショットから戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

### 構文

```
▶▶—HEALTH_CONT_HI—(—dbname—, —member—)————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### dbname

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権

- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

```
SELECT * FROM TABLE(HEALTH_CONT_HI(' ', -1)) AS T
```

以下はこの照会の出力例です。

```
SNAPSHOT_TIMESTAMP      CONTAINER_NAME      ...
-----
2006-02-13-12.30.40.759542 D:¥DB2¥NODE0000¥SAMPLE¥T0000000¥C0000000.CAT ...
2006-02-13-12.30.40.759542 D:¥DB2¥NODE0000¥SAMPLE¥T0000003¥C0000000.LRG ...
2006-02-13-12.30.40.759542 D:¥DB2¥NODE0000¥SAMPLE¥T0000004¥C0000000.UTM ...
2006-02-13-12.30.40.759542 D:¥DB2¥NODE0000¥SAMPLE¥T0000001¥C0000000.TMP ...
2006-02-13-12.30.40.759542 D:¥DB2¥NODE0000¥SAMPLE¥T0000002¥C0000000.LRG ...
```

5 record(s) selected.

この照会からの出力 (続き)。

```
... NODE_NUMBER HI_ID      HI_VALUE HI_TIMESTAMP      ...
... -----
...      -      3001      1 2006-02-13-12.26.26.158000 ...
...      -      3001      1 2006-02-13-12.26.26.158000 ...
...      -      3001      1 2006-02-13-12.26.26.158000 ...
...      -      3001      1 2006-02-13-12.26.26.158000 ...
...      -      3001      1 2006-02-13-12.26.26.158000 ...
```

この照会からの出力 (続き)。

```
... HI_ALERT_STATE      HI_ALERT_STATE_DETAIL HI_FORMULA      HI_ADDITIONAL_INFO
... -----
...      1 Normal      1      -
...      1 Normal      1      -
...      1 Normal      1      -
...      1 Normal      1      -
...      1 Normal      1      -
```

## 戻される情報

表 299. HEALTH\_CONT\_HI 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
CONTAINER_NAME	VARCHAR(256)	container_name - コンテナ名
NODE_NUMBER	INTEGER	node_number - ノード番号

表 299. HEALTH\_CONT\_HI 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
HI_ID	BIGINT	スナップショット・データ・ストリーム内のヘルス・インディケーターを固有に識別する番号。
HI_VALUE	SMALLINT	ヘルス・インディケーターの値。
HI_TIMESTAMP	TIMESTAMP	アラートが生成された日時。
HI_ALERT_STATE	BIGINT	アラートの重大度。
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 列のテキスト記述。
HI_FORMULA	VARCHAR(2048)	ヘルス・インディケーターを計算するのに使用する公式。
HI_ADDITIONAL_INFO	VARCHAR(4096)	ヘルス・インディケーターに関する追加情報。

## HEALTH\_CONT\_HI\_HIS

データベースのヘルス・スナップショットからコンテナーに関するヘルス・インディケーター履歴情報を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

## 構文

▶▶—HEALTH\_CONT\_HI\_HIS—(—dbname—, —member—)————▶▶

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。アクティブなデータベース・



メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
SELECT * FROM TABLE(HEALTH_CONT_HI_HIS(' ', -1)) AS T
```

以下はこの照会の出力例です。

```
SNAPSHOT_TIMESTAMP          CONTAINER_NAME          ...
-----
2006-02-13-12.30.41.915646 D:¥DB2¥NODE0000¥SAMPLE¥T0000000¥C0000000.CAT ...
2006-02-13-12.30.41.915646 D:¥DB2¥NODE0000¥SAMPLE¥T0000000¥C0000000.CAT ...
2006-02-13-12.30.41.915646 D:¥DB2¥NODE0000¥SAMPLE¥T0000003¥C0000000.LRG ...
2006-02-13-12.30.41.915646 D:¥DB2¥NODE0000¥SAMPLE¥T0000003¥C0000000.LRG ...
2006-02-13-12.30.41.915646 D:¥DB2¥NODE0000¥SAMPLE¥T0000004¥C0000000.UTM ...
2006-02-13-12.30.41.915646 D:¥DB2¥NODE0000¥SAMPLE¥T0000004¥C0000000.UTM ...
2006-02-13-12.30.41.915646 D:¥DB2¥NODE0000¥SAMPLE¥T0000001¥C0000000.TMP ...
2006-02-13-12.30.41.915646 D:¥DB2¥NODE0000¥SAMPLE¥T0000001¥C0000000.TMP ...
2006-02-13-12.30.41.915646 D:¥DB2¥NODE0000¥SAMPLE¥T0000002¥C0000000.LRG ...
2006-02-13-12.30.41.915646 D:¥DB2¥NODE0000¥SAMPLE¥T0000002¥C0000000.LRG ...
```

10 record(s) selected.

この照会からの出力 (続き)。

```
... NODE_NUMBER HI_ID      HI_TIMESTAMP          HI_VALUE HI_ALERT_STATE ...
... -----
...          -          3001 2006-02-13-12.16.25.911000          1          1 ...
...          -          3001 2006-02-13-12.06.26.168000          1          1 ...
...          -          3001 2006-02-13-12.16.25.911000          1          1 ...
...          -          3001 2006-02-13-12.06.26.168000          1          1 ...
...          -          3001 2006-02-13-12.16.25.911000          1          1 ...
...          -          3001 2006-02-13-12.06.26.168000          1          1 ...
...          -          3001 2006-02-13-12.16.25.911000          1          1 ...
...          -          3001 2006-02-13-12.06.26.168000          1          1 ...
...          -          3001 2006-02-13-12.16.25.911000          1          1 ...
...          -          3001 2006-02-13-12.06.26.168000          1          1 ...
```

この照会からの出力 (続き)。

```
... HI_ALERT_STATE_DETAIL HI_FORMULA      HI_ADDITIONAL_INFO
... -----
... Normal              1              -
... Normal              1              -
... Normal              1              -
... Normal              1              -
```

... Normal	1	-
... Normal	1	-
... Normal	1	-
... Normal	1	-
... Normal	1	-
... Normal	1	-

## 戻される情報

表 300. HEALTH\_CONT\_HI\_HIS 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
CONTAINER_NAME	VARCHAR(256)	container_name - コンテナ名
NODE_NUMBER	INTEGER	node_number - ノード番号
HI_ID	BIGINT	スナップショット・データ・ストリーム内のヘルス・インディケーターを固有に識別する番号。
HI_TIMESTAMP	TIMESTAMP	アラートが生成された日時。
HI_VALUE	SMALLINT	ヘルス・インディケーターの値。
HI_ALERT_STATE	BIGINT	アラートの重大度。
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 列のテキスト記述。
HI_FORMULA	VARCHAR(2048)	ヘルス・インディケーターを計算するのに使用する公式。
HI_ADDITIONAL_INFO	VARCHAR(4096)	ヘルス・インディケーターに関する追加情報。

## HEALTH\_CONT\_INFO

HEALTH\_CONT\_INFO 表関数は、コンテナ情報をデータベースのヘルス・スナップショットから戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

## 構文

▶▶—HEALTH\_CONT\_INFO—(—dbname—, —member—)————▶▶

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
SELECT * FROM TABLE(HEALTH_CONT_INFO('',-1)) AS T
```

以下はこの照会の出力例です。

```
SNAPSHOT_TIMESTAMP      CONTAINER_NAME          ...
-----
2006-02-13-12.30.40.541209 D:¥DB2¥NODE0000¥SAMPLE¥T0000000¥C0000000.CAT ...
2006-02-13-12.30.40.541209 D:¥DB2¥NODE0000¥SAMPLE¥T0000003¥C0000000.LRG ...
2006-02-13-12.30.40.541209 D:¥DB2¥NODE0000¥SAMPLE¥T0000004¥C0000000.UTM ...
2006-02-13-12.30.40.541209 D:¥DB2¥NODE0000¥SAMPLE¥T0000001¥C0000000.TMP ...
2006-02-13-12.30.40.541209 D:¥DB2¥NODE0000¥SAMPLE¥T0000002¥C0000000.LRG ...
```

5 record(s) selected.

この照会からの出力 (続き)。

```
... TABLESPACE_NAME      NODE_NUMBER ...
... -----
... SYSCATSPACE           - ...
... SYSTOOLSPACE          - ...
... SYSTOOLSTMPSPACE      - ...
... TEMPSPACE1            - ...
... USERSPACE1            - ...
```

この照会からの出力 (続き)。

```
... ROLLED_UP_ALERT_STATE ROLLED_UP_ALERT_STATE_DETAIL
-----
...                1 Normal
...                1 Normal
...                1 Normal
...                1 Normal
...                1 Normal
```

## 戻される情報

表 301. HEALTH\_CONT\_INFO 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
CONTAINER_NAME	VARCHAR(256)	container_name - コンテナ名
TABLESPACE_NAME	VARCHAR(128)	tablespace_name - 表スペース名
NODE_NUMBER	INTEGER	node_number - ノード番号
ROLLED_UP_ALERT_STATE	BIGINT	このスナップショットでキャプチャーされた最も重大なアラート状態。
ROLLED_UP_ALERT_STATE_DETAIL	VARCHAR(20)	ROLLED_UP_ALERT_STATE 列のテキスト記述。

## HEALTH\_DB\_HI

HEALTH\_DB\_HI 表関数は、データベースのヘルス・スナップショットからヘルス・インディケータの情報を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

## 構文

```
▶▶ HEALTH_DB_HI (—dbname—, —member—) ◀◀
```

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を

指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。データベース・インスタンス下のすべてのデータベースからスナップショットを取る場合は、**NULL** 値を指定します。

*member*

有効なデータベース・メンバー番号を指定する、タイプ **INTEGER** の入力引数。現在のデータベース・メンバーの場合は **-1**、すべてアクティブなデータベース・メンバーの集合の場合は **-2** を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

**NULL** 値を指定すると、**-1** が暗黙的に設定されます。

**許可**

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する **EXECUTE** 特権
- **DATAACCESS** 権限
- **DBADM** 権限
- **SQLADM** 権限

**デフォルトの PUBLIC 特権**

制限のないデータベースでは、この関数が自動的に作成されると、**EXECUTE** 特権が **PUBLIC** に付与されます。

**例**

```
SELECT * FROM TABLE(HEALTH_DB_HI(' ', -1)) AS T
```

以下はこの照会の出力例です。

SNAPSHOT_TIMESTAMP	HI_ID	DB_NAME	HI_VALUE	...
2006-02-13-12.30.23.949888	1001	SAMPLE	0	...
2006-02-13-12.30.23.949888	1002	SAMPLE	0	...
2006-02-13-12.30.23.949888	1003	SAMPLE	0	...
2006-02-13-12.30.23.949888	1005	SAMPLE	6	...
2006-02-13-12.30.23.949888	1006	SAMPLE	53	...
2006-02-13-12.30.23.949888	1008	SAMPLE	3	...
2006-02-13-12.30.23.949888	1010	SAMPLE	0	...
2006-02-13-12.30.23.949888	1014	SAMPLE	74	...
2006-02-13-12.30.23.949888	1015	SAMPLE	1	...
2006-02-13-12.30.23.949888	1018	SAMPLE	1	...
2006-02-13-12.30.23.949888	1022	SAMPLE	1	...

11 record(s) selected.

この照会からの出力 (続き)。

...	HI_TIMESTAMP	HI_ALERT_STATE	HI_ALERT_STATE_DETAIL	...
...	2006-02-13-12.26.26.158000	1	Normal	...
...	2006-02-13-12.26.26.158000	1	Normal	...
...	2006-02-13-12.26.26.158000	1	Normal	...
...	2006-02-13-12.26.26.158000	1	Normal	...
...	2006-02-13-12.26.26.158000	1	Normal	...
...	2006-02-13-12.26.26.158000	1	Normal	...
...	2006-02-13-12.26.26.158000	1	Normal	...
...	2006-02-13-12.26.26.158000	1	Normal	...

```
... 2006-02-13-12.30.25.640000      2 Attention      ...
... 2006-02-13-12.30.25.640000      2 Attention      ...
... 2006-02-13-12.29.25.281000      2 Attention      ...
```

この照会からの出力 (続き)。

```
... HI_FORMULA                      ...
... -----                        ...
... 0                                ...
... ((0 / 5000) * 100)               ...
...                                  ...
...                                  ...
...                                  ...
...                                  ...
...                                  ...
...                                  ...
...                                  ...
... ((0 - 0) / ((118 - 0) + 1)) * 100 ...
...                                  ...
...                                  ...
...                                  ...
...                                  ...
... ((1170384 / (1170384 + 19229616)) * 100) ...
...                                  ...
...                                  ...
...                                  ...
...                                  ...
...                                  ...
... ((11155116032 / 21138935808) * 100) ...
...                                  ...
...                                  ...
...                                  ...
...                                  ...
...                                  ...
... ((5264 / (50 * 4096)) * 100)      ...
... ((0 / 5) * 100)                  ...
... ((4587520 / 6160384) * 100)     ...
... -                                 ...
...                                  ...
...                                  ...
...                                  ...
...                                  ...
...                                  ...
... -                                 ...
...                                  ...
...                                  ...
...                                  ...
...                                  ...
... -                                 ...
...                                  ...
...                                  ...
```

この照会からの出力 (続き)。

```
... HI_ADDITIONAL_INFO              ...
... -----                        ...
... -                                 ...
... The high watermark for shared sort ...
... memory is "57". "99"% of the time ...
... the sort heap allocation is less ...
... than or equal to "246". The sort ...
```

```

... heap (sortheap) database
... configuration parameter is set
... to "256". The high watermark for
... private sort memory is "0".
... The sort heap (sortheap) database
... configuration parameter is set to
... "256". The high watermark for
... private sort memory is "57". The
... high watermark for shared sort
... memory is "0"
... The following are the related
... database configuration parameter
... settings: logprimary is "3",
... logsecond is "2", and logfilsiz
... is "1000". The application with
... the oldest transaction is "712".
... The following are the related
... database configuration parameter
... settings: logprimary is "3",
... logsecond is "2", and logfilsiz
... is "1000", blk_log_dsk_ful is
... "NO", logarchmeth1 is "OFF" and
... logarchmeth2 is "OFF".
... -
... -
... -
... The scope setting in the reorganization
... policy is "TABSCHEMA NOT LIKE 'SYS%'".
... Automatic reorganization (AUTO_REORG)
... for this database is set to "OFF".
... The longest estimated reorganization
... time is "N/A".
... The last successful backup was taken
... at "N/A". The log space consumed since
... this last backup has been "N/A" 4KB
... pages. Automation for database backup
... is set to "OFF". The last automated
... backup returned with SQLCODE = "N/A".
... The longest estimated backup time
... is "N/A".
... The scope is "N/A". Automatic
... statistics collection (AUTO_RUNSTATS)
... is set to "OFF".

```

## 戻される情報

表 302. HEALTH\_DB\_HI 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
HI_ID	BIGINT	スナップショット・データ・ストリーム内のヘルス・インディケーターを固有に識別する番号。
DB_NAME	VARCHAR(128)	db_name - データベース名
HI_VALUE	SMALLINT	ヘルス・インディケーターの値。
HI_TIMESTAMP	TIMESTAMP	アラートが生成された日時。

表 302. HEALTH\_DB\_HI 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
HI_ALERT_STATE	BIGINT	アラートの重大度。
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 列のテキスト記述。
HI_FORMULA	VARCHAR(2048)	ヘルス・インディケータを計算するのに使用する公式。
HI_ADDITIONAL_INFO	VARCHAR(4096)	ヘルス・インディケータに関する追加情報。

## HEALTH\_DB\_HI\_HIS

HEALTH\_DB\_HI\_HIS 表関数は、データベースのヘルス・スナップショットからヘルス・インディケータの履歴情報を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

### 構文

▶▶ HEALTH\_DB\_HI\_HIS (—dbname—, —member—) ▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### dbname

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。データベース・インスタンス下のすべてのデータベースからスナップショットを取る場合は、NULL 値を指定します。

#### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権



- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

```
SELECT * FROM TABLE(HEALTH_DB_HI_HIS(' ', -1)) AS T
```

以下はこの照会の出力例です。

SNAPSHOT_TIMESTAMP	HI_ID	DB_NAME	HI_VALUE	...
2006-02-13-12.30.26.325627	1001	SAMPLE	0	...
...	...	...	...	...
2006-02-13-12.30.26.325627	1002	SAMPLE	0	...
...	...	...	...	...
2006-02-13-12.30.26.325627	1003	SAMPLE	0	...
...	...	...	...	...
2006-02-13-12.30.26.325627	1005	SAMPLE	3	...
...	...	...	...	...
2006-02-13-12.30.26.325627	1008	SAMPLE	2	...
...	...	...	...	...
2006-02-13-12.30.26.325627	1010	SAMPLE	0	...
...	...	...	...	...
2006-02-13-12.30.26.325627	1014	SAMPLE	73	...
...	...	...	...	...
2006-02-13-12.30.26.325627	1015	SAMPLE	1	...
...	...	...	...	...
2006-02-13-12.30.26.325627	1018	SAMPLE	1	...
...	...	...	...	...
2006-02-13-12.30.26.325627	1022	SAMPLE	1	...
...	...	...	...	...

この照会からの出力 (続き)。

HI_TIMESTAMP	HI_ALERT_STATE	HI_ALERT_STATE_DETAIL	...
...	...	...	...
2006-02-13-12.21.25.649000	1	Normal	...
...	...	...	...
2006-02-13-12.21.25.649000	1	Normal	...
...	...	...	...
2006-02-13-12.20.25.182000	1	Normal	...
...	...	...	...
2006-02-13-12.16.25.911000	1	Normal	...
...	...	...	...
2006-02-13-12.16.25.911000	1	Normal	...
...	...	...	...
2006-02-13-12.16.25.911000	1	Normal	...
...	...	...	...
2006-02-13-12.21.25.649000	1	Normal	...
...	...	...	...
2006-02-13-12.29.55.461000	2	Attention	...
...	...	...	...
2006-02-13-12.29.25.281000	2	Attention	...
...	...	...	...
2006-02-13-12.27.55.743000	2	Attention	...
...	...	...	...

この照会からの出力 (続き)。



```

... configuration parameter is set
... to "256". The high watermark
... for private sort memory is "0".
...
... The sort heap (sortheap) database
... configuration parameter is set
... to "256". The high watermark for
... private sort memory is "15". The
... high watermark for shared sort
... memory is "0"
...
... The following are the related
... database configuration parameter
... settings: logprimary is "3",
... logsecond is "2", and logfilsiz
... is "1000". The application with
... the oldest transaction is "712".
...
... -
... -
... -
... -
...
... The scope setting in the
... reorganization policy is
... "TABSCHEMA NOT LIKE 'SYS%'".
... Automatic reorganization
... (AUTO_REORG) for this database
... is set to "OFF". The longest
... estimated reorganization time
... is "N/A".
...
... The last successful backup was taken
... at "N/A". The log space consumed
... since this last backup has been
... "N/A" 4KB pages. Automation for
... database backup is set to "OFF". The
... last automated backup returned with
... SQLCODE = "N/A". The longest
... estimated backup time is "N/A".
...
... The scope is "N/A". Automatic
... statistics collection
... (AUTO_RUNSTATS) is set to "OFF".
...

```

## 戻される情報

表 303. HEALTH\_DB\_HI\_HIS 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
HI_ID	BIGINT	スナップショット・データ・ストリーム内のヘルス・インディケータを固有に識別する番号。
DB_NAME	VARCHAR(128)	db_name - データベース名

表 303. HEALTH\_DB\_HI\_HIS 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
HI_VALUE	SMALLINT	ヘルス・インディケーターの値。
HI_TIMESTAMP	TIMESTAMP	アラートが生成された日時。
HI_ALERT_STATE	BIGINT	アラートの重大度。
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 列のテキスト記述。
HI_FORMULA	VARCHAR(2048)	ヘルス・インディケーターを計算するのに使用する公式。
HI_ADDITIONAL_INFO	VARCHAR(4096)	ヘルス・インディケーターに関する追加情報。

## HEALTH\_DB\_HIC

HEALTH\_DB\_HIC 関数は、データベースのヘルス・スナップショットからコレクション・ヘルス・インディケーターの情報を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

### 構文

►►—HEALTH\_DB\_HIC—(—dbname—, —member—)—————►►

スキーマは SYSPROC です。

### 表関数パラメーター

#### dbname

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。データベース・インスタンス下のすべてのデータベースからスナップショットを取る場合は、NULL 値を指定します。

#### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
SELECT * FROM TABLE(HEALTH_DB_HIC('',-1)) AS T
```

以下はこの照会の出力例です。

```
SNAPSHOT_TIMESTAMP      HI_ID      DB_NAME      ...
-----
2006-02-13-12.30.33.870959      1015 SAMPLE      ...
2006-02-13-12.30.33.870959      1022 SAMPLE      ...
```

2 record(s) selected.

この照会からの出力 (続き)。

```
... HI_OBJ_NAME                      HI_OBJ_DETAIL      ...
... -----
... "JESSICAE"."EMPLOYEE"              REORG TABLE      ...
... "SYSIBM"."SYSDATAPARTITIONEXPRESSION" RUNSTATS          ...
```

この照会からの出力 (続き)。

```
... HI_OBJ_STATE HI_OBJ_STATE_DETAIL HI_TIMESTAMP
... -----
...          2 Attention              2006-02-13-12.24.27.000000
...          2 Attention              2006-02-13-12.29.26.000000
```

## 戻される情報

表 304. HEALTH\_DB\_HIC 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
HI_ID	BIGINT	スナップショット・データ・ストリーム内のヘルス・インディケーターを固有に識別する番号。
DB_NAME	VARCHAR(128)	db_name - データベース名
HI_OBJ_NAME	VARCHAR(512)	コレクション内のオブジェクトを固有に識別する名前。

表 304. HEALTH\_DB\_HIC 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
HI_OBJ_DETAIL	VARCHAR(512)	オブジェクトがコレクションに追加された理由を記述したテキスト。
HI_OBJ_STATE	SMALLINT	<p>オブジェクトの状態。有効な状態 (sqlmon.h に定義されています) は次のとおりです。</p> <ul style="list-style-type: none"> <li>• <b>NORMAL (1)</b>。このオブジェクトに対してアクションは必要ありません。</li> <li>• <b>ATTENTION (2)</b>。このヘルス・インディケーターに対して自動化は有効になっていません。手動でアクションをとる必要があります。</li> <li>• <b>AUTOMATED (5)</b>。このヘルス・インディケーターに対して自動化が有効になっています。アクションは自動的に開始します。</li> <li>• <b>AUTOMATE_FAILED (6)</b>。このヘルス・インディケーターに対して自動化が有効になっています。アクションは開始しましたが、正常に完了できませんでした。現時点で、手操作による介入が必要です。</li> </ul>
HI_OBJ_STATE_DETAIL	VARCHAR(20)	HI_OBJ_STATE 列の値を変換したストリングのバージョン。
HI_TIMESTAMP	TIMESTAMP	アラートが生成された日時。

## HEALTH\_DB\_HIC\_HIS

データベースのヘルス・スナップショットからコレクション・ヘルス・インディケーターの履歴情報を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

### 構文

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。データベース・インスタンス下のすべてのデータベースからスナップショットを取る場合は、NULL 値を指定します。

### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現行のデータベース・パーティションの場合は -1、またはすべてのアクティブなデータベース・パーティションの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
SELECT * FROM TABLE(HEALTH_DB_HIC_HIS('',-1)) AS T
```

以下はこの照会の出力例です。

HI_HIS_ENTRY_NUM	SNAPSHOT_TIMESTAMP	HI_ID	...
1	2006-02-13-12.30.34.496720	1015	...
2	2006-02-13-12.30.34.496720	1022	...
3	2006-02-13-12.30.34.496720	1022	...
4	2006-02-13-12.30.34.496720	1022	...
5	2006-02-13-12.30.34.496720	1022	...
6	2006-02-13-12.30.34.496720	1022	...
7	2006-02-13-12.30.34.496720	1022	...
8	2006-02-13-12.30.34.496720	1022	...
9	2006-02-13-12.30.34.496720	1022	...
10	2006-02-13-12.30.34.496720	1022	...

10 record(s) selected.

この照会からの出力 (続き)。

```

... DB_NAME      HI_OBJ_NAME      HI_OBJ_STATE ...
-----
... SAMPLE      "JESSICAE"."EMPLOYEE"      2 ...
... SAMPLE      "SYSIBM"."SYSDATAPARTITIONEXPRESSION"      2 ...
... SAMPLE      "SYSIBM"."SYSDATAPARTITIONEXPRESSION"      2 ...
... SAMPLE      "SYSIBM"."SYSDATAPARTITIONEXPRESSION"      2 ...
... SAMPLE      "SYSIBM"."SYSDATAPARTITIONEXPRESSION"      1 ...
... SAMPLE      "SYSIBM"."SYSDATAPARTITIONEXPRESSION"      1 ...
... SAMPLE      "SYSIBM"."SYSDATAPARTITIONEXPRESSION"      1 ...
... SAMPLE      "SYSIBM"."SYSDATAPARTITIONEXPRESSION"      1 ...
... SAMPLE      "SYSIBM"."SYSDATAPARTITIONEXPRESSION"      1 ...
... SAMPLE      "SYSIBM"."SYSDATAPARTITIONEXPRESSION"      1 ...

```

この照会からの出力 (続き)。

```

... HI_OBJ_STATE_DETAIL  HI_TIMESTAMP
-----
... Attention            2006-02-10-09.04.57.000000
... Attention            2006-02-13-12.27.56.000000
... Attention            2006-02-13-12.26.27.000000
... Attention            2006-02-13-12.24.56.000000
... Normal                2006-02-13-12.23.28.000000
... Normal                2006-02-13-12.21.56.000000
... Normal                2006-02-13-12.20.26.000000
... Normal                2006-02-13-12.18.57.000000
... Normal                2006-02-13-12.17.27.000000
... Normal                2006-02-13-12.15.56.000000

```

## 戻される情報

表 305. HEALTH\_DB\_HIC\_HIS 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
HI_HIS_ENTRY_NUM	SMALLINT	履歴項目を固有に識別する番号。
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
HI_ID	BIGINT	スナップショット・データ・ストリーム内のヘルス・インディケーターを固有に識別する番号。
DB_NAME	VARCHAR(128)	db_name - データベース名
HI_OBJ_NAME	VARCHAR(512)	コレクション内のオブジェクトを固有に識別する名前。



表 305. HEALTH\_DB\_HIC\_HIS 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
HI_OBJ_STATE	SMALLINT	<p>オブジェクトの状態。有効な状態 (sqlmon.h に定義されています) は次のとおりです。</p> <ul style="list-style-type: none"> <li>• NORMAL (1)。このオブジェクトに対してアクションは必要ありません。</li> <li>• ATTENTION (2)。このヘルス・インディケーターに対して自動化は有効になっていません。手動でアクションをとる必要があります。</li> <li>• AUTOMATED (5)。このヘルス・インディケーターに対して自動化が有効になっています。アクションは自動的に開始します。</li> <li>• AUTOMATE_FAILED (6)。このヘルス・インディケーターに対して自動化が有効になっています。アクションは開始しましたが、正常に完了できませんでした。現時点で、手操作による介入が必要です。</li> </ul>
HI_OBJ_STATE_DETAIL	VARCHAR(20)	HI_OBJ_STATE 列の値を変換したストリングのバージョン。
HI_TIMESTAMP	TIMESTAMP	アラートが生成された日時。

## HEALTH\_DB\_INFO

HEALTH\_DB\_INFO 表関数は、データベースのヘルス・スナップショットからの情報を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

## 構文

▶▶—HEALTH\_DB\_INFO—(—dbname—,—member—)————▶▶

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。データベース・インスタンス下のすべてのデータベースからスナップショットを取る場合は、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
SELECT * FROM TABLE(HEALTH_DB_INFO('',-1)) AS T
```

以下はこの照会の出力例です。

```
SNAPSHOT_TIMESTAMP      DB_NAME      INPUT_DB_ALIAS      ...
-----
2006-02-13-12.30.23.340081 SAMPLE      SAMPLE      ...
```

1 record(s) selected.

この照会からの出力 (続き)。

```
... DB_PATH      DB_LOCATION SERVER_PLATFORM ...
... -----
... D:¥DB2¥NODE0000¥SQL00003¥      1      5 ...
```

この照会からの出力 (続き)。

```
... ROLLED_UP_ALERT_STATE ROLLED_UP_ALERT_STATE_DETAIL
... -----
...      4 Alarm
```

## 戻される情報

表 306. HEALTH\_DB\_INFO 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
DB_NAME	VARCHAR(128)	db_name - データベース名
INPUT_DB_ALIAS	VARCHAR(128)	input_db_alias - 入力データベース別名
DB_PATH	VARCHAR(1024)	db_path - データベース・パス
DB_LOCATION	INTEGER	db_location - データベース・ロケーション
SERVER_PLATFORM	INTEGER	server_platform - サーバーのオペレーティング・システム
ROLLED_UP_ALERT_STATE	BIGINT	このスナップショットでキャプチャーされた最も重大なアラート状態。
ROLLED_UP_ALERT_STATE_DETAIL	VARCHAR(20)	ROLLED_UP_ALERT_STATE 列のテキスト記述。

## HEALTH\_DBM\_HI

HEALTH\_DBM\_HI 表関数は、DB2 データベース・マネージャーのヘルス・スナップショットからヘルス・インディケーターの情報を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

## 構文

▶▶—HEALTH\_DBM\_HI—(—member—)————▶▶

スキーマは SYSPROC です。

## 表関数パラメーター

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
SELECT * FROM TABLE(HEALTH_DBM_HI(-1)) AS T
```

以下はこの照会の出力例です。

```
SNAPSHOT_TIMESTAMP      HI_ID      SERVER_INSTANCE_NAME    ...
-----
2006-02-13-12.30.19.773632      1 DB2      ...
2006-02-13-12.30.19.773632      4 DB2      ...
```

2 record(s) selected.

この照会からの出力 (続き)。

```
... HI_VALUE HI_TIMESTAMP      HI_ALERT_STATE HI_ALERT_STATE_DETAIL ...
... -----
...          0 2006-02-13-12.26.26.158000      1 Normal      ...
...          4 2006-02-13-12.26.26.158000      4 Alarm       ...
```

この照会からの出力 (続き)。

```
... HI_FORMULA      HI_ADDITIONAL_INFO
... -----
... 0 -
... ((327680 / 327680) * 100) -
```

表 307. HEALTH\_DBM\_HI 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
HI_ID	BIGINT	スナップショット・データ・ストリーム内のヘルス・インディケーターを固有に識別する番号。
SERVER_INSTANCE_NAME	VARCHAR(128)	server_instance_name - サーバー・インスタンス名
HI_VALUE	SMALLINT	ヘルス・インディケーターの値。
HI_TIMESTAMP	TIMESTAMP	アラートが生成された日時。

表 307. HEALTH\_DBM\_HI 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
HI_ALERT_STATE	BIGINT	アラートの重大度。
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 列のテキスト記述。
HI_FORMULA	VARCHAR(2048)	ヘルス・インディケータを計算するのに使用する公式。
HI_ADDITIONAL_INFO	VARCHAR(4096)	ヘルス・インディケータに関する追加情報。

## HEALTH\_DBM\_HI\_HIS

HEALTH\_DBM\_HI\_HIS 表関数は、DB2 データベース・マネージャーのヘルス・スナップショットからヘルス・インディケータの履歴情報を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

### 構文

▶▶—HEALTH\_DBM\_HI\_HIS—(—*member*—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

```
SELECT * FROM TABLE(HEALTH_DBM_HI_HIS(-1)) AS T
```

以下はこの照会の出力例です。

SNAPSHOT_TIMESTAMP	HI_ID	SERVER_INSTANCE_NAME	HI_VALUE ...
2006-02-13-12.30.20.460905	1	DB2	0 ...
2006-02-13-12.30.20.460905	1	DB2	0 ...
2006-02-13-12.30.20.460905	1	DB2	0 ...
2006-02-13-12.30.20.460905	1	DB2	0 ...
2006-02-13-12.30.20.460905	1	DB2	0 ...
2006-02-13-12.30.20.460905	1	DB2	0 ...
2006-02-13-12.30.20.460905	1	DB2	0 ...
2006-02-13-12.30.20.460905	1	DB2	0 ...
2006-02-13-12.30.20.460905	4	DB2	100 ...
2006-02-13-12.30.20.460905	4	DB2	100 ...
2006-02-13-12.30.20.460905	4	DB2	100 ...
2006-02-13-12.30.20.460905	4	DB2	100 ...
2006-02-13-12.30.20.460905	4	DB2	60 ...
2006-02-13-12.30.20.460905	4	DB2	60 ...
2006-02-13-12.30.20.460905	4	DB2	60 ...
2006-02-13-12.30.20.460905	4	DB2	60 ...

18 record(s) selected.

この照会の出力 (続き)。

HI_TIMESTAMP	HI_ALERT_STATE	HI_ALERT_STATE_DETAIL ...
2006-02-13-12.21.25.649000	1	Normal
2006-02-13-12.16.25.911000	1	Normal
2006-02-13-12.11.25.377000	1	Normal
2006-02-13-12.06.26.168000	1	Normal
2006-02-13-12.01.25.165000	1	Normal
2006-02-13-11.56.25.927000	1	Normal
2006-02-13-11.51.25.452000	1	Normal
2006-02-13-11.46.25.211000	1	Normal
2006-02-13-11.41.25.972000	1	Normal
2006-02-13-12.21.25.649000	4	Alarm
2006-02-13-12.16.25.911000	4	Alarm
2006-02-13-12.11.25.377000	4	Alarm
2006-02-13-12.06.26.168000	4	Alarm
2006-02-13-12.01.25.165000	1	Normal
2006-02-13-11.56.25.927000	1	Normal
2006-02-13-11.51.25.452000	1	Normal
2006-02-13-11.46.25.211000	1	Normal
2006-02-13-11.41.25.972000	1	Normal

この照会の出力 (続き)。

HI_FORMULA	HI_ADDITIONAL_INFO
0	-
0	-
0	-
0	-
0	-

```

... 0 -
... 0 -
... 0 -
... 0 -
... ((327680 / 327680) * 100) -
... ((327680 / 327680) * 100) -
... ((327680 / 327680) * 100) -
... ((327680 / 327680) * 100) -
... ((196608 / 327680) * 100) -
... ((196608 / 327680) * 100) -
... ((196608 / 327680) * 100) -
... ((196608 / 327680) * 100) -
... ((196608 / 327680) * 100) -
... ((196608 / 327680) * 100) -

```

## 戻される情報

表 308. HEALTH\_DBM\_HI\_HIS 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
HI_ID	BIGINT	スナップショット・データ・ストリーム内のヘルス・インディケーターを固有に識別する番号。
SERVER_INSTANCE_NAME	VARCHAR(128)	server_instance_name - サーバー・インスタンス名
HI_VALUE	SMALLINT	ヘルス・インディケーターの値。
HI_TIMESTAMP	TIMESTAMP	アラートが生成された日時。
HI_ALERT_STATE	BIGINT	アラートの重大度。
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 列のテキスト記述。
HI_FORMULA	VARCHAR(2048)	ヘルス・インディケーターを計算するのに使用する公式。
HI_ADDITIONAL_INFO	VARCHAR(4096)	ヘルス・インディケーターに関する追加情報。

## HEALTH\_DBM\_INFO

HEALTH\_DBM\_INFO 関数は、DB2 データベース・マネージャーのヘルス・スナップショットからの情報を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

## 構文

▶▶—HEALTH\_DBM\_INFO—(—member—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

```
SELECT * FROM TABLE(HEALTH_DBM_INFO(-1)) AS T
```

以下はこの照会の出力例です。

```
SNAPSHOT_TIMESTAMP          SERVER_INSTANCE_NAME          ROLLED_UP_ALERT_STATE  ...
-----
2006-02-13-12.30.19.663924  DB2                             4                      ...
```

1 record(s) selected.

この照会からの出力 (続き)。

```
... ROLLED_UP_ALERT_STATE_DETAIL DB2START_TIME          ...
... Alarm                          2006-02-09-10.56.18.126182 ...
```

この照会からの出力 (続き)。

```
... LAST_RESET          NUM_NODES_IN_DB2_INSTANCE
... -                      1
```



## 戻される情報

表 309. HEALTH\_DBM\_INFO 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
SERVER_INSTANCE_NAME	VARCHAR(128)	server_instance_name - サーバー・インスタンス名
ROLLED_UP_ALERT_STATE	BIGINT	このスナップショットでキャプチャーされた最も重大なアラート状態。
ROLLED_UP_ALERT_STATE_DETAIL	VARCHAR(20)	ROLLED_UP_ALERT_STATE 列のテキスト記述。
DB2START_TIME	TIMESTAMP	db2start_time - データベース・マネージャー開始タイム・スタンプ
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ
NUM_NODES_IN_DB2_INSTANCE	INTEGER	num_nodes_in_db2_instance - データベース・パーティション内のノード数

## HEALTH\_GET\_ALERT\_ACTION\_CFG

さまざまなオブジェクト・タイプ (データベース・マネージャー、データベース、表スペース、表スペース・コンテナ)、およびさまざまな構成レベル (インストール・デフォルト、インスタンス、グローバル、およびオブジェクト) のヘルス・アラート・アクション構成設定値を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』 (<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

### 構文

```
▶▶HEALTH_GET_ALERT_ACTION_CFG(—objecttype—,—cfg_level—,—dbname—,—  
▶—objectname—)————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### objecttype

オブジェクト・タイプを示す、タイプ VARCHAR(3) の入力引数。値は以下のいずれかでなければならず、大/小文字を区別しません。

- データベース・マネージャーは、'DBM'
- データベースは、'DB'
- 表スペースは、'TS'
- 表スペース・コンテナは、'TSC'

注: 前後のスペースがあっても、それは無視されます。

#### *cfg\_level*

構成レベルを表す、タイプ VARCHAR(1) の入力引数。値は以下のいずれかでなければならず、大/小文字を区別しません。

- *objecttype* が 'DBM' の場合: インストール・デフォルトは 'D'、インスタンス・レベルは 'G' または 'O'。
- *objecttype* が 'DBM' でない場合: インストール・デフォルトは 'D'、グローバル・レベルは 'G'、オブジェクト・レベルは 'O'。

#### *dbname*

データベース名を示す、タイプ VARCHAR(128) の入力引数。*objecttype* が 'DB'、'TS'、または 'TSC' で、かつ *cfg\_level* が 'O' である場合、データベース名を指定しなければなりません。*objecttype* と *cfg\_level* の組み合わせがそれ以外のものである場合、*dbname* パラメーターは NULL (または空ストリング) である必要があります。

#### *objectname*

オブジェクト名を示す、タイプ VARCHAR(1024) の入力引数。例えば、<table space name> または <table space name>.<container name> など。*objecttype* が 'TS' または 'TSC' で、かつ *cfg\_level* が 'O' である場合、オブジェクト名を指定しなければなりません。*objecttype* と *cfg\_level* の組み合わせがそれ以外のものである場合、*objectname* パラメーターは NULL (または空ストリング) である必要があります。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

例 1: データベース SAMPLE に関する、ヘルス・インディケータ ID 1004 のオブジェクト・レベルのアラート・アクション構成設定値を検索します。

```
SELECT OBJECTTYPE, CFG_LEVEL, SUBSTR(DBNAME,1,8) AS DBNAME,
       SUBSTR(OBJECTNAME,1,8) AS OBJECTNAME, ID, IS_DEFAULT,
       SUBSTR(CONDITION,1,10) AS CONDITION, ACTIONTYPE,
       SUBSTR(ACTIONNAME,1,30) AS ACTIONNAME, SUBSTR(USERID,1,8) AS USERID,
```

```

SUBSTR(HOSTNAME,1,10) AS HOSTNAME, SCRIPT_TYPE,
SUBSTR(WORKING_DIR,1,10) AS WORKING_DIR, TERMINATION_CHAR,
SUBSTR(PARAMETERS,1,10) AS PARAMETERS
FROM TABLE(HEALTH_GET_ALERT_ACTION_CFG('DB','0','SAMPLE','')) AS ACTION_CFG
WHERE ID = 1004

```

以下はこの照会の出力例です。

OBJECTTYPE	CFG_LEVEL	DBNAME	OBJECTNAME	ID	IS_DEFAULT	CONDITION
DB	0	SAMPLE		1004	1	ALARM
DB	0	SAMPLE		1004	1	ALARM

2 record(s) selected.

この照会の出力 (続き)。

...	ACTIONTYPE	ACTIONNAME	USERID	HOSTNAME
...	S	~/health_center/script/scrpn6	uid1	-
...	T	00.0005	uid1	HOST3

この照会の出力 (続き)。

...	SCRIPT_TYPE	WORKING_DIR	TERMINATION_CHAR	PARAMETERS
...	0	~/health_c	-	-
...	-	-	-	-

例 2: ヘルス・インディケータ ID 1004 の、データベース SAMPLE に関する状態、アクション・タイプ、アクション名、ホスト名、およびスクリプト・タイプを検索します。

```

SELECT CONDITION, ACTIONTYPE, SUBSTR(ACTIONNAME,1,35) AS ACTIONNAME,
SUBSTR(USERID,1,8) AS USERID, SUBSTR(HOSTNAME,1,10) AS HOSTNAME, SCRIPT_TYPE
FROM TABLE(HEALTH_GET_ALERT_ACTION_CFG('DB','0','SAMPLE','')) AS ALERT_ACTION_CFG
WHERE ID=1004

```

以下はこの照会の出力例です。

CONDITION	ACTIONTYPE	ACTIONNAME	...
ALARM	S	~/health_center/script/scrpn6	...
ALARM	T	00.0005	...

2 record(s) selected.

この照会の出力 (続き)。

...	USERID	HOSTNAME	SCRIPT_TYPE
...	uid1	-	0
...	uid1	HOST3	-

## 使用上の注意

HEALTH\_GET\_IND\_DEFINITION 表関数は、ヘルス・インディケータ ID をヘルス・インディケータ名にマップするために使用できます。

## 戻される情報

表 310. HEALTH\_GET\_ALERT\_ACTION\_CFG 表関数によって戻される情報

列名	データ・タイプ	説明
OBJECTTYPE	VARCHAR(3)	オブジェクト・タイプ。
CFG_LEVEL	CHAR(1)	構成レベル。
DBNAME	VARCHAR(128)	データベース名
OBJECTNAME	VARCHAR(512)	object_name - オブジェクト名 モニター・エレメント
ID	BIGINT	id - クラスター・キャッシング・ファシリティ ID モニター・エレメント
IS_DEFAULT	SMALLINT	設定がデフォルトであるかどうか。デフォルトである場合は 1、デフォルトでない場合は 0、適用外である場合は NULL。
CONDITION	VARCHAR(32)	アクションが起動するアラート状態。
ACTIONTYPE	CHAR(1)	アクション・タイプ。スクリプト・アクションは 'S'、タスク・アクションは 'T'。
ACTIONNAME	VARCHAR(5000)	ACTIONTYPE が 'S' である場合、これはスクリプト・パス名です。ACTIONTYPE が 'T' である場合、これはタスク ID です。
USERID	VARCHAR(1024)	その下でアクションが実行されるユーザー名。
HOSTNAME	VARCHAR(255)	hostname - ホスト名 : モニター・エレメント
SCRIPT_TYPE	CHAR(1)	スクリプト・タイプ。 ACTIONTYPE が 'S' の場合、オペレーティング・システムのコマンド・スクリプトは 'O'、DB2 コマンド・スクリプトは 'D'。ACTIONTYPE が 'T' の場合、NULL。
WORKING_DIR	VARCHAR(5000)	ACTIONTYPE が 'S' の場合、スクリプトの作業ディレクトリー。ACTIONTYPE が 'T' の場合、NULL。
TERMINATION_CHAR	VARCHAR(4)	DB2 コマンド・スクリプト・アクションである場合、ステートメント終了文字。その他の場合は NULL。





...	ID	IS_DEFAULT	WARNING_THRESHOLD	...
...	1001	0	0	...
...	1018	0	0	...
...	1015	0	0	...
...	1022	0	0	...
...	1002	1	95	...
...	1003	1	30	...
...	1004	1	60	...
...	1005	1	75	...
...	1006	1	75	...
...	1007	1	5	...
...	1008	1	75	...
...	1009	1	5	...
...	1010	1	50	...
...	1011	1	80	...

この照会の出力 (続き)。

...	ALARM_THRESHOLD	SENSITIVITY	EVALUATE	ACTION_ENABLED	...
...	0	0	0	0	0
...	0	0	1	0	0
...	0	0	1	0	0
...	0	0	1	0	0
...	100	0	0	0	0
...	50	0	1	0	0
...	30	0	1	0	0
...	85	0	1	0	0
...	85	0	1	0	0
...	10	0	1	0	0
...	85	0	1	0	0
...	10	0	1	0	0
...	70	0	1	0	0
...	70	0	0	0	0

例 2: データベース SAMPLE の表スペース USERSPACE1 に関する、ヘルス・インディケーター ID '2002' の警告およびアラームのしきい値を検索します。

```
SELECT WARNING_THRESHOLD, ALARM_THRESHOLD
FROM TABLE(SYSPROC.HEALTH_GET_ALERT_CFG('TS','0','SAMPLE','USERSPACE1'))
AS T WHERE ID = 2002
```

以下はこの照会の出力例です。

WARNING_THRESHOLD	ALARM_THRESHOLD
80	90

SQL22004N Cannot find the requested configuration for the given object.  
Returning default configuration for "tablespaces".

1 record(s) selected with 1 warning messages printed.

### 使用上の注意

HEALTH\_GET\_IND\_DEFINITION 表関数は、ヘルス・インディケーター ID をヘルス・インディケーター名にマップするために使用できます。

例: データベース SAMPLE の表スペース USERSPACE1 に関する、ヘルス・インディケーター「表スペース使用率」(ts.ts\_util) の警告およびアラームのしきい値を検索します。

```

WITH HINAME(ID) AS (SELECT ID FROM TABLE(SYSPROC.HEALTH_GET_IND_DEFINITION('')) AS W
WHERE NAME = 'ts.ts_util')
SELECT WARNING_THRESHOLD, ALARM_THRESHOLD
FROM TABLE(SYSPROC.HEALTH_GET_ALERT_CFG('TS','0','SAMPLE','USERSPACE1')) AS T,
HINAME AS H
WHERE T.ID = H.ID

```

以下はこの照会の出力例です。

```

WARNING_THRESHOLD    ALARM_THRESHOLD
-----
                        80                90
SQL22004N  Cannot find the requested configuration for the given object.
Returning default configuration for "tablespaces".

```

1 record(s) selected with 1 warning messages printed.

## 戻される情報

表 311. HEALTH\_GET\_ALERT\_CFG 表関数によって戻される情報

列名	データ・タイプ	説明
OBJECTTYPE	VARCHAR(3)	オブジェクト・タイプ。
CFG_LEVEL	CHAR(1)	構成レベル。
DBNAME	VARCHAR(128)	データベース名
OBJECTNAME	VARCHAR(512)	object_name - オブジェクト名モニター・エレメント
ID	BIGINT	id - クラスター・キャッシング・ファシリティ ID モニター・エレメント
IS_DEFAULT	SMALLINT	設定がデフォルトであるかどうか。デフォルトである場合は 1、デフォルトでない場合は 0、適用外である場合は NULL。
WARNING_THRESHOLD	BIGINT	警告しきい値。適用外の場合は NULL 値。
ALARM_THRESHOLD	BIGINT	アラームしきい値。適用外の場合は NULL 値。
SENSITIVITY	BIGINT	ヘルス・インディケータの感度。
EVALUATE	SMALLINT	このヘルス・インディケータが評価されている場合は 1、評価されていない場合は 0。
ACTION_ENABLED	SMALLINT	アラートの発生時にアクションの実行が可能な場合は 1、アラートの発生時にアクションの実行が可能でない場合は 0。



## HEALTH\_GET\_IND\_DEFINITION

ヘルス・インディケータ定義を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

### 構文

▶▶—HEALTH\_GET\_IND\_DEFINITION—(—*locale*—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### *locale*

戻される翻訳可能出力のロケールを示す、タイプ VARCHAR(33) の入力引数。入力ロケールをデータベース・サーバーがサポートしていない場合、SQL 警告メッセージが発行され、デフォルト言語 (英語) が使用されます。入力ロケールが指定されていない場合、つまりその値が NULL (または空ストリング) の場合、デフォルト言語が使用されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

**例 1:** ヘルス・インディケータ db.db\_op\_status のタイプと簡略説明をフランス語で検索します。

```
SELECT TYPE, SHORT_DESCRIPTION
  FROM TABLE(SYSPROC.HEALTH_GET_IND_DEFINITION('fr_FR'))
  AS IND_DEFINITION WHERE NAME = 'db.db_op_status'
```

以下はこの照会の出力例です。

TYPE	SHORT_DESCRIPTION
STATE	Etat opérationnel de la base de données

1 record(s) selected.

例 2: ヘルス・インディケータ ID 1001 の簡略説明を英語で検索します。

```
SELECT SHORT_DESCRIPTION FROM TABLE(SYSPROC.HEALTH_GET_IND_DEFINITION('en_US')
AS IND_DEFINITION WHERE ID = 1001
```

以下はこの照会の出力例です。

```
SHORT_DESCRIPTION
-----
Database Operational State
```

例 3: すべてのヘルス・インディケータ ID および名前を検索します。

```
SELECT ID, NAME FROM TABLE(HEALTH_GET_IND_DEFINITION('')) AS T
```

以下はこの照会の出力例です。

```
ID              NAME
-----
1 db2.db2_op_status
2 db2.sort_privmem_util
4 db2.mon_heap_util
1001 db.db_op_status
1002 db.sort_shrmem_util
...
2001 ts.ts_op_status
2002 ts.ts_util
...
3002 tsc.tscont_util
1015 db.tb_reorg_req
...
```

## 戻される情報

表 312. HEALTH\_GET\_IND\_DEFINITION 表関数によって戻される情報

列名	データ・タイプ	説明
ID	BIGINT	id - クラスタ・キャッシング・ファシリティ ID モニター・エレメント
NAME	VARCHAR(128)	ヘルス・インディケータ名。
SHORT_DESCRIPTION	VARCHAR(1024)	ヘルス・インディケータ簡略説明。
LONG_DESCRIPTION	VARCHAR(32672)	ヘルス・インディケータ詳細説明。

表 312. HEALTH\_GET\_IND\_DEFINITION 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明
TYPE	VARCHAR(16)	ヘルス・インディケータータイプ。可能な値は以下のとおりです。 <ul style="list-style-type: none"> <li>'THRESHOLD_UPPER': 上限のしきい値ベースのヘルス・インディケータータイプ。</li> <li>'THRESHOLD_LOWER': 下限のしきい値ベースのヘルス・インディケータータイプ。</li> <li>'STATE': 状態ベースのヘルス・インディケータータイプ。</li> <li>'COLLECTION_STATE': コレクション状態ベースのヘルス・インディケータータイプ。</li> </ul>
UNIT	VARCHAR(1024)	ヘルス・インディケータータイプの値およびしきい値の単位。または適用外の場合は NULL。
CATEGORY	VARCHAR(1024)	ヘルス・インディケータータイプのカテゴリ。
FORMULA	VARCHAR(512)	ヘルス・インディケータータイプの式。
REFRESH_INTERVAL	BIGINT	ヘルス・インディケータータイプの評価の間隔 (秒単位)。

## HEALTH\_HI\_REC

特定の DB2 オブジェクト上のアラート状態にあるヘルス・インディケータータイプに関連した一連の推奨事項を取り出します。推奨事項は、アラート状態を解決するために実行できるアクション (例えば、実行可能なスクリプト) についての情報を収めた XML 文書の中に戻されます。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、このプロシージャは非推奨になっており、今後のリリースで除去される可能性があります。

### 構文

```

▶▶HEALTH_HI_REC(—schema-version—,—indicator-id—,—dbname—,—
▶—object-type—,—object-name—,—dbpartitionnum—,—client-locale—,—
▶—recommendation-doc—)

```

スキーマは SYSPROC です。

このプロシージャで戻されるスクリプトがある場合、ヘルス・インディケータータイプがアラート状態になったインスタンスから呼び出す必要があります。

識別されたオブジェクト上の指定ヘルス・インディケーターがアラート状態ではない場合、エラーが戻されます (SQLSTATE 5U0ZZ)。

## プロシージャ・パラメーター

### *schema-version*

XML 文書を表すのに使用されるスキーマのバージョン番号を指定する、タイプ INTEGER の入力引数。推奨事項文書には、該当スキーマ・バージョンに定義されたエレメントと属性だけが含まれています。有効なスキーマ・バージョンは、`sqllib` ディレクトリーの `include` サブディレクトリーにある、`db2ApiDf.h` で定義されます。

### *indicator-id*

推奨事項が要求されているヘルス・インディケーターの数値 ID を指定する、タイプ INTEGER の入力引数。有効なヘルス・インディケーター ID は、`sqllib` ディレクトリーの `include` サブディレクトリーにある、`sqlmon.h` で定義されます。

### *dbname*

ヘルス・インディケーターがアラート状態となった対象のデータベースで、オブジェクト・タイプが `DB2HEALTH_OBJTYPE_TS_CONTAINER`、`DB2HEALTH_OBJTYPE_TABLESPACE`、または `DB2HEALTH_OBJTYPE_DATABASE` のいずれかであるとき、そのデータベースの別名を指定する、タイプ `VARCHAR(255)` の入力引数。それ以外の場合、`NULL` を指定します。

### *object-type*

ヘルス・インディケーターがアラート状態になったオブジェクトのタイプを指定する、タイプ INTEGER の入力引数。有効なオブジェクト・タイプは、`sqllib` ディレクトリーの `include` サブディレクトリーにある、`sqlmon.h` で定義されま

### *object-name*

オブジェクト・タイプが `DB2HEALTH_OBJTYPE_TABLESPACE` か `DB2HEALTH_OBJTYPE_TS_CONTAINER` に設定されるときの、表スペースまたは表スペース・コンテナの名前を指定する、タイプ `VARCHAR(255)` の入力引数。オブジェクト・タイプが `DB2HEALTH_OBJTYPE_DATABASE` または `DB2HEALTH_OBJTYPE_DATABASE_MANAGER` である場合は、`NULL` を指定してください。表スペース・コンテナの場合、オブジェクト名は、`table_space_name.container_name` と指定されます。

### *dbpartitionnum*

ヘルス・インディケーターがアラート状態になったデータベース・パーティション数を指定する INTEGER のタイプの入力引数。有効な値は、0 から 999、-1 (現在接続されているデータベース・パーティションを指定)、および -2 (すべてのデータベース・パーティションを指定) です。アクティブなデータベース・パーティションとは、アプリケーションからデータベースに接続して、これを使用することが可能なパーティションのことです。

### *client-locale*

クライアントの言語 ID を指定する、タイプ `VARCHAR(33)` の入力引数。このパラメーターは、推奨事項が戻される言語を指定するときに使用します。値が指定されていない場合は、`'En_US'` (英語) が使用されます。指定したロケールの

メッセージ・ファイルがサーバーで使用できない場合には、'En\_US' がデフォルトとして使用されるので注意してください。

#### *recommendation-doc*

DB2 Health Recommendation スキーマ定義 (sqllib ディレクトリーの misc サブディレクトリーにある、XML スキーマ DB2RecommendationSchema.xsd を参照) に従ってフォーマットされた推奨事項文書 (XML) を含む、タイプ BLOB(2M) の出力引数。メッセージがターゲット・インスタンスでの呼び出し元のロケールで使用できない場合、XML 文書は UTF-8 でエンコードされ、その文書のテキストは呼び出し元のロケールまたは英語になります。

## 許可

このプロシージャを実行するには、以下のいずれかの権限が必要です。

- プロシージャでの EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## HEALTH\_TBS\_HI

データベースの表スペースのヘルス・スナップショットから、表スペースのヘルス・インディケータの情報を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

## 構文

►►—HEALTH\_TBS\_HI—(—dbname—,—member—)—————►►

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。

"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

## member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
SELECT * FROM TABLE(HEALTH_TBS_HI('',-1)) AS T
```

以下はこの照会の出力例です。

SNAPSHOT_TIMESTAMP	TABLESPACE_NAME	HI_ID	HI_VALUE	...
2006-02-13-12.30.35.229196	SYSCATSPACE	2001	0	...
2006-02-13-12.30.35.229196	SYSCATSPACE	2002	99	...
2006-02-13-12.30.35.229196	SYSCATSPACE	2003	0	...
2006-02-13-12.30.35.229196	SYSTOOLSPACE	2001	0	...
2006-02-13-12.30.35.229196	SYSTOOLSPACE	2002	62	...
2006-02-13-12.30.35.229196	SYSTOOLSPACE	2003	0	...
2006-02-13-12.30.35.229196	SYSTOOLSTMPSPACE	2001	0	...
2006-02-13-12.30.35.229196	TEMPSPACE1	2001	0	...
2006-02-13-12.30.35.229196	USERSPACE1	2001	0	...
2006-02-13-12.30.35.229196	USERSPACE1	2002	100	...
2006-02-13-12.30.35.229196	USERSPACE1	2003	0	...

11 record(s) selected.

この照会からの出力 (続き)。

HI_TIMESTAMP	HI_ALERT_STATE	HI_ALERT_STATE_DETAIL	...
2006-02-13-12.26.26.158000	1	Normal	...
2006-02-13-12.26.26.158000	4	Alarm	...
2006-02-13-12.26.26.158000	1	Normal	...
2006-02-13-12.26.26.158000	1	Normal	...
2006-02-13-12.26.26.158000	1	Normal	...
2006-02-13-12.26.26.158000	1	Normal	...
2006-02-13-12.26.26.158000	1	Normal	...
2006-02-13-12.26.26.158000	1	Normal	...
2006-02-13-12.26.26.158000	1	Normal	...
2006-02-13-12.26.26.158000	4	Alarm	...
2006-02-13-12.26.26.158000	1	Normal	...

この照会からの出力 (続き)。

```
... HI_FORMULA          HI_ADDITIONAL_INFO
... -----
... 0                  -
... ((9376 / 9468) * 100) The short term table space growth rate
                        from "02/13/2006 11:26:26.000158" to
                        "02/13/2006 12:26:26.000158" is "N/A"
                        bytes per second and the long term growth
                        rate from "02/12/2006 12:26:26.000158"
                        to "02/13/2006 12:26:26.000158" is "N/A"
                        bytes per second. Time to fullness is
                        projected to be "N/A" and "N/A".
                        The table space is defined
                        with automatic storage set to "YES" and
                        automatic resize enabled set to "YES".
... 0                  The table space is defined with automatic
                        storage set to "YES" and automatic resize
                        enabled set to "YES". The following are
                        the automatic resize settings: increase
                        size (bytes) "-1", increase size (percent)
                        "N/A", maximum size (bytes) "-1". The
                        current table space size (bytes) is
                        "38797312".
... 0                  -
... ((156 / 252) * 100) The short term table space growth rate
                        from "02/13/2006 11:26:26.000158" to
                        "02/13/2006 12:26:26.000158" is "N/A"
                        bytes per second and the long term growth
                        rate from "02/12/2006 12:26:26.000158"
                        to "02/13/2006 12:26:26.000158" is "N/A"
                        bytes per second. Time to fullness is
                        projected to be "N/A" and "N/A".
                        The table space is defined
                        with automatic storage set to "YES" and
                        automatic resize enabled set to "YES".
... 0                  The table space is defined with automatic
                        storage set to "YES" and automatic resize
                        enabled set to "YES". The following are
                        the automatic resize settings: increase
                        size (bytes) "-1", increase size (percent)
                        "N/A", maximum size (bytes) "-1". The
                        current table space size (bytes) is
                        "1048576".
... 0                  -
... 0                  -
... 0                  -
... ((1504 / 1504) * 100) The short term table space growth rate from
                        "02/13/2006 11:26:26.000158" to
                        "02/13/2006 12:26:26.000158" is "N/A"
                        bytes per second and the long term growth
                        rate from "02/12/2006 12:26:26.000158" to
                        "02/13/2006 12:26:26.000158" is "N/A" bytes
                        per second. Time to fullness is projected
                        to be "N/A" and "N/A". The
                        table space is defined with automatic storage
                        set to "YES" and automatic resize enabled
                        set to "YES".
... 0                  The table space is defined with automatic
                        storage set to "YES" and automatic resize
                        enabled set to "YES". The following are
                        the automatic resize settings: increase
                        size (bytes) "-1", increase size (percent)
                        "N/A", maximum size (bytes) "-1". The
                        current table space size (bytes) is
                        "6291456".
```

## 戻される情報

表 313. HEALTH\_TBS\_HI 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
TABLESPACE_NAME	VARCHAR(128)	tablespace_name - 表スペース名
HI_ID	BIGINT	スナップショット・データ・ストリーム内のヘルス・インディケーターを固有に識別する番号。
HI_VALUE	SMALLINT	ヘルス・インディケーターの値。
HI_TIMESTAMP	TIMESTAMP	アラートが生成された日時。
HI_ALERT_STATE	BIGINT	アラートの重大度。
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 列のテキスト記述。
HI_FORMULA	VARCHAR(2048)	ヘルス・インディケーターを計算するのに使用する公式。
HI_ADDITIONAL_INFO	VARCHAR(4096)	ヘルス・インディケーターに関する追加情報。

## HEALTH\_TBS\_HI\_HIS

HEALTH\_TBS\_HI\_HIS 表関数は、データベースのヘルス・スナップショットから表スペースのヘルス・インディケーターの履歴情報を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

### 構文

▶▶—HEALTH\_TBS\_HI\_HIS—(—dbname—, —member—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

*dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマ



ンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

```
SELECT * FROM TABLE(HEALTH_TBS_HI_HIS('',-1)) AS T
```

以下はこの照会の出力例です。

SNAPSHOT_TIMESTAMP	TABLESPACE_NAME	HI_ID	...
2006-02-13-12.30.37.181478	SYSCATSPACE	2001	...
2006-02-13-12.30.37.181478	SYSCATSPACE	2001	...
2006-02-13-12.30.37.181478	SYSCATSPACE	2002	...
2006-02-13-12.30.37.181478	SYSCATSPACE	2002	...
2006-02-13-12.30.37.181478	SYSCATSPACE	2003	...
2006-02-13-12.30.37.181478	SYSCATSPACE	2003	...
2006-02-13-12.30.37.181478	SYSTOOLSPACE	2001	...
2006-02-13-12.30.37.181478	SYSTOOLSPACE	2001	...
2006-02-13-12.30.37.181478	SYSTOOLSPACE	2002	...
2006-02-13-12.30.37.181478	SYSTOOLSPACE	2002	...
2006-02-13-12.30.37.181478	SYSTOOLSPACE	2003	...
2006-02-13-12.30.37.181478	SYSTOOLSPACE	2003	...
2006-02-13-12.30.37.181478	SYSTOOLSTMPSPACE	2001	...
2006-02-13-12.30.37.181478	SYSTOOLSTMPSPACE	2001	...
2006-02-13-12.30.37.181478	TEMPSPACE1	2001	...
2006-02-13-12.30.37.181478	TEMPSPACE1	2001	...
2006-02-13-12.30.37.181478	USERSPACE1	2001	...
2006-02-13-12.30.37.181478	USERSPACE1	2001	...
2006-02-13-12.30.37.181478	USERSPACE1	2002	...
2006-02-13-12.30.37.181478	USERSPACE1	2002	...
2006-02-13-12.30.37.181478	USERSPACE1	2003	...
2006-02-13-12.30.37.181478	USERSPACE1	2003	...

22 record(s) selected.

この照会からの出力 (続き)。

HI_TIMESTAMP	HI_VALUE	HI_ALERT_STATE	HI_ALERT_STATE_DETAIL
2006-02-13-12.16.25.911000	0	1	Normal
2006-02-13-12.06.26.168000	0	1	Normal
2006-02-13-12.16.25.911000	99	4	Alarm
2006-02-13-12.06.26.168000	99	4	Alarm
2006-02-13-12.16.25.911000	0	1	Normal
2006-02-13-12.06.26.168000	0	1	Normal
2006-02-13-12.16.25.911000	0	1	Normal
2006-02-13-12.06.26.168000	0	1	Normal
2006-02-13-12.16.25.911000	62	1	Normal
2006-02-13-12.06.26.168000	62	1	Normal
2006-02-13-12.16.25.911000	0	1	Normal
2006-02-13-12.06.26.168000	0	1	Normal
2006-02-13-12.16.25.911000	0	1	Normal
2006-02-13-12.06.26.168000	0	1	Normal
2006-02-13-12.16.25.911000	0	1	Normal
2006-02-13-12.06.26.168000	0	1	Normal
2006-02-13-12.16.25.911000	0	1	Normal
2006-02-13-12.06.26.168000	0	1	Normal
2006-02-13-12.16.25.911000	100	4	Alarm
2006-02-13-12.06.26.168000	100	4	Alarm
2006-02-13-12.16.25.911000	0	1	Normal
2006-02-13-12.06.26.168000	0	1	Normal

この照会からの出力 (続き)。

HI_FORMULA	HI_ADDITIONAL_INFO
0	-
0	-
((9376 / 9468) * 100)	The short term table space growth rate from "02/13/2006 11:16:25.000911" to "02/13/2006 12:16:25.000911" is "N/A" bytes per second and the long term growth rate from "02/12/2006 12:16:25.000911" to "02/13/2006 12:16:25.000911" is "N/A" bytes per second. Time to fullness is projected to be "N/A" and "N/A". The table space is defined with automatic storage set to "YES" and automatic resize enabled set to "YES".
((9376 / 9468) * 100)	The short term table space growth rate from "02/13/2006 11:06:26.000168" to "02/13/2006 12:06:26.000168" is "N/A" bytes per second and the long term growth rate from "02/12/2006 12:06:26.000168" to "02/13/2006 12:06:26.000168" is "N/A" bytes per second. Time to fullness is projected to be "N/A" and "N/A". The table space is defined with automatic storage set to "YES" and automatic resize enabled set to "YES".
0	The table space is defined with automatic storage set to "YES" and automatic resize enabled set to "YES". The following are the automatic resize settings: increase size (bytes) "-1", increase size (percent) "N/A", maximum size (bytes) "-1". The current table space size (bytes) is "38797312".
0	The table space is defined with automatic storage set to "YES" and automatic resize enabled set to "YES". The following are the automatic resize settings: increase size (bytes) "-1", increase size (percent) "N/A", maximum size (bytes) "-1". The

```

... 0
... 0
... ((156 / 252) * 100)
... ((156 / 252) * 100)
... 0
... 0
... 0
... 0
... 0
... 0
... ((1504 / 1504) * 100)
... ((1504 / 1504) * 100)

```

current table space size (bytes) is "38797312".

The short term table space growth rate from "02/13/2006 11:16:25.000911" to "02/13/2006 12:16:25.000911" is "N/A" bytes per second and the long term growth rate from "02/12/2006 12:16:25.000911" to "02/13/2006 12:16:25.000911" is "N/A" bytes per second. Time to fullness is projected to be "N/A" and "N/A". The table space is defined with automatic storage set to "YES" and automatic resize enabled set to "YES".

The short term table space growth rate from "02/13/2006 11:06:26.000168" to "02/13/2006 12:06:26.000168" is "N/A" bytes per second and the long term growth rate from "02/12/2006 12:06:26.000168" to "02/13/2006 12:06:26.000168" is "N/A" bytes per second. Time to fullness is projected to be "N/A" and "N/A". The table space is defined with automatic storage set to "YES" and automatic resize enabled set to "YES".

The table space is defined with automatic storage set to "YES" and automatic resize enabled set to "YES". The following are the automatic resize settings: increase size (bytes) "-1", increase size (percent) "N/A", maximum size (bytes) "-1". The current table space size (bytes) is "1048576".

The table space is defined with automatic storage set to "YES" and automatic resize enabled set to "YES". The following are the automatic resize settings: increase size (bytes) "-1", increase size (percent) "N/A", maximum size (bytes) "-1". The current table space size (bytes) is "1048576".

The short term table space growth rate from "02/13/2006 11:16:25.000911" to "02/13/2006 12:16:25.000911" is "N/A" bytes per second and the long term growth rate from "02/12/2006 12:16:25.000911" to "02/13/2006 12:16:25.000911" is "N/A" bytes per second. Time to fullness is projected to be "N/A" and "N/A". The table space is defined with automatic storage set to "YES" and automatic resize enabled set to "YES".

The short term table space growth rate from "02/13/2006 11:06:26.000168" to "02/13/2006 12:06:26.000168" is "N/A" bytes per second and the long term growth rate from "02/12/2006 12:06:26.000168" to "02/13/2006 12:06:26.000168" is "N/A" bytes per second. Time to fullness is projected to be "N/A" and "N/A".

```

... 0 The table space is defined
with automatic storage set to "YES" and
automatic resize enabled set to "YES".
The table space is defined with automatic
storage set to "YES" and automatic
resize enabled set to "YES". The
following are the automatic resize
settings: increase size (bytes) "-1",
increase size (percent) "N/A", maximum
size (bytes) "-1". The current table
space size (bytes) is "6291456".
... 0 The table space is defined with automatic
storage set to "YES" and automatic
resize enabled set to "YES". The
following are the automatic resize
settings: increase size (bytes) "-1",
increase size (percent) "N/A", maximum
size (bytes) "-1". The current table
space size (bytes) is "6291456".

```

## 戻される情報

表 314. HEALTH\_TBS\_HI\_HIS 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
TABLESPACE_NAME	VARCHAR(128)	tablespace_name - 表スペース名
HI_ID	BIGINT	スナップショット・データ・ストリーム内のヘルス・インディケーターを固有に識別する番号。
HI_TIMESTAMP	TIMESTAMP	アラートが生成された日時。
HI_VALUE	SMALLINT	ヘルス・インディケーターの値。
HI_ALERT_STATE	BIGINT	アラートの重大度。
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 列のテキスト記述。
HI_FORMULA	VARCHAR(2048)	ヘルス・インディケーターを計算するのに使用する公式。
HI_ADDITIONAL_INFO	VARCHAR(4096)	ヘルス・インディケーターに関する追加情報。

## HEALTH\_TBS\_INFO

データベースのヘルス・スナップショットから表スペースの情報を戻します。

**重要:** バージョン 9.7 でヘルス・モニターが非推奨になったため、この表関数は推奨されておらず、将来のリリースで除去される可能性があります。これはDB2 pureScale環境ではサポートされていません。詳しくは、『ヘルス・モニターが推奨されなくなった』(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html>) を参照してください。

## 構文

```
▶▶—HEALTH_TBS_INFO—(—dbname—, —member—)—————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

```
SELECT * FROM TABLE(HEALTH_TBS_INFO(' ', -1)) AS T
```

以下はこの照会の出力例です。

```
SNAPSHOT_TIMESTAMP      TABLESPACE_NAME      ...
-----
2006-02-13-12.30.35.027383 SYSCATSPACE           ...
2006-02-13-12.30.35.027383 SYSTOOLSPACE          ...
2006-02-13-12.30.35.027383 SYSTOOLSTMPSPACE     ...
2006-02-13-12.30.35.027383 TEMPSPACE1           ...
2006-02-13-12.30.35.027383 USERSPACE1           ...
```

5 record(s) selected.

この照会からの出力 (続き)。

```

... ROLLED_UP_ALERT_STATE ROLLED_UP_ALERT_STATE_DETAIL
... -----
...                4 Alarm
...                1 Normal
...                1 Normal
...                1 Normal
...                4 Alarm

```

## 戻される情報

表 315. HEALTH\_TBS\_INFO 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
TABLESPACE_NAME	VARCHAR(128)	tablespace_name - 表スペース名
ROLLED_UP_ALERT_STATE	BIGINT	このスナップショットでキャプチャーされた最も重大なアラート状態。
ROLLED_UP_ALERT_STATE_DETAIL	VARCHAR(20)	ROLLED_UP_ALERT_STATE 列のテキスト記述。

## REG\_VARIABLES 管理ビュー - 使用中の DB2 レジストリー設定の検索

REG\_VARIABLES 管理ビューは、すべてのデータベース・パーティションから DB2 レジストリー設定値を戻します。

インスタンス開始後に **db2set** コマンドを使用して構成された DB2 レジストリー変数がある場合、REG\_VARIABLES 管理ビューを照会して戻される DB2 レジストリー変数値と、**db2set** コマンドによって戻される値が異なることがあります。REG\_VARIABLES はインスタンスの開始時に有効だった値を戻すに過ぎないため、この違いが生じます。

注: この管理ビューは非推奨になり、ENV\_GET\_REG\_VARIABLES 表関数に置き換えられました。

スキーマは SYSIBMADM です。

### 許可

以下のいずれかの権限が必要です。

- REG\_VARIABLES 管理ビューに対する SELECT 特権
- REG\_VARIABLES 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在使用されている DB2 レジストリー設定を要求します。

```
SELECT * from SYSIBMADM.REG_VARIABLES
```

以下はこの照会の出力例です。

DBPARTITIONNUM	REG_VAR_NAME	REG_VAR_VALUE	IS_AGGREGATE	AGGREGATE_NAME
0	DB2ADMINSERVER	DB2DAS00	0	-
0	DB2INSTPROF	D:%SQLLIB	0	-
0	DB2PATH	D:%SQLLIB	0	-
0	DB2SYSTEM	D570	0	-
0	DB2TEMPDIR	D:%SQLLIB%	0	-
0	DB2_EXTSECURITY	YES	0	-

6 record(s) selected.

### 戻される情報

表 316. REG\_VARIABLES 管理ビューによって戻される情報

列名	データ・タイプ	説明
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
REG_VAR_NAME	VARCHAR(256)	DB2 レジストリー変数の名前。
REG_VAR_VALUE	VARCHAR(2048)	DB2 レジストリー変数の現行設定。
IS_AGGREGATE	SMALLINT	DB2 レジストリー変数が集約変数であるかどうかを示します。考えられる戻り値は、集約変数でない場合は 0、集約変数である場合は 1 です。
AGGREGATE_NAME	VARCHAR(256)	現在 DB2 レジストリー変数が、構成済みの集約から値を取得している場合には、集約の名前。レジストリー変数が集約から設定されているのではない場合、または集約から設定されたもののオーバーライドされている場合、AGGREGATE_NAME の値は NULL になります。

表 316. REG\_VARIABLES 管理ビューによって戻される情報 (続き)

列名	データ・タイプ	説明
LEVEL	CHAR(1)	DB2 レジストリー変数がその値を獲得するレベルを示します。考えられる戻り値と、それが表す対応するレベルは以下のとおりです。 <ul style="list-style-type: none"> <li>• I = インスタンス</li> <li>• G = グローバル</li> <li>• N = データベース・パーティション</li> <li>• E = 環境</li> </ul>

## SNAPAGENT\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数 - memory\_pool 論理データ・グループのスナップショット情報の検索

SNAPAGENT\_MEMORY\_POOL 管理ビューおよび  
SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数は、エージェント・レベルでのメモリ一使用量についての情報を戻します。

**重要:** SNAPAGENT\_MEMORY\_POOL 管理ビューと  
SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数は非推奨になり、604 ページの『MON\_GET\_MEMORY\_POOL - メモリー・プール情報の取得』と 607 ページの『MON\_GET\_MEMORY\_SET - メモリー・セット情報の取得』に置き換えられました。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPAGENT\_MEMORY\_POOL 管理ビュー』
- 1371 ページの『SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数』

### SNAPAGENT\_MEMORY\_POOL 管理ビュー

この管理ビューを使用して、現在接続中のデータベースのエージェント・レベルでのメモリ使用量に関する memory\_pool 論理データ・グループのスナップショット情報を取得することができます。

SNAPAGENT\_MEMORY\_POOL 管理ビューを  
SNAPAGENT、SNAPAPPL、SNAPAPPL\_INFO、SNAPSTMT、および  
SNAPSUBSECTION 管理ビューとともに使用すると、**GET SNAPSHOT FOR APPLICATIONS ON database-alias** CLP コマンドに相当する情報が提供されます。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1374 ページの表 317 を参照してください。



## 許可

このビューを使用するには、以下のいずれかが必要です。

- SNAPAGENT\_MEMORY\_POOL 管理ビューに対する SELECT 特権
- SNAPAGENT\_MEMORY\_POOL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

メモリー・プールおよびそれらの現在のサイズのリストを取得します。

```
SELECT AGENT_ID, POOL_ID, POOL_CUR_SIZE FROM SYSIBMADM.SNAPAGENT_MEMORY_POOL
```

以下はこの照会の出力例です。

AGENT_ID	POOL_ID	POOL_CUR_SIZE
48	APPLICATION	65536
48	OTHER	65536
48	APPL_CONTROL	65536
47	APPLICATION	65536
47	OTHER	131072
47	APPL_CONTROL	65536
46	OTHER	327680
46	APPLICATION	262144
46	APPL_CONTROL	65536

9 record(s) selected.

## SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数

SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数は SNAPAGENT\_MEMORY\_POOL 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデー

データベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数を SNAP\_GET\_AGENT、SNAP\_GET\_APPL、SNAP\_GET\_APPL\_INFO、SNAP\_GET\_STMT、および SNAP\_GET\_SUBSECTION 表関数とともに使用すると、**GET SNAPSHOT FOR ALL APPLICATIONS** CLP コマンドに相当する情報が提供されます。

戻される可能性のある情報の完全なリストは、1374 ページの表 317 を参照してください。

## 構文

```
▶▶ SNAP_GET_AGENT_MEMORY_POOL ( ( dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべてのデータベースのメモリー・プールおよびそれらの現在のサイズのリストを取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AGENT_ID, POOL_ID, POOL_CUR_SIZE
FROM TABLE(SNAP_GET_AGENT_MEMORY_POOL(CAST (NULL AS VARCHAR(128)), -1))
AS T
```

以下はこの照会の出力例です。

DB_NAME	AGENT_ID	POOL_ID	POOL_CUR_SIZE
SAMPLE	48	APPLICATION	65536
SAMPLE	48	OTHER	65536
SAMPLE	48	APPL_CONTROL	65536
SAMPLE	47	APPLICATION	65536
SAMPLE	47	OTHER	131072
SAMPLE	47	APPL_CONTROL	65536
SAMPLE	46	OTHER	327680
SAMPLE	46	APPLICATION	262144
SAMPLE	46	APPL_CONTROL	65536
TESTDB	30	APPLICATION	65536
TESTDB	30	OTHER	65536
TESTDB	30	APPL_CONTROL	65536
TESTDB	29	APPLICATION	65536
TESTDB	29	OTHER	131072
TESTDB	29	APPL_CONTROL	65536
TESTDB	28	OTHER	327680
TESTDB	28	APPLICATION	65536
TESTDB	28	APPL_CONTROL	65536

18 record(s) selected.

## 戻される情報

表 317. SNAPAGENT\_MEMORY\_POOL 管理ビューおよび  
SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
AGENT_PID	BIGINT	agent_pid - エンジン・ディスパッチ可能単位 (EDU)
POOL_ID	VARCHAR(14)	pool_id - メモリー・プール ID。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• APP_GROUP</li> <li>• APPL_CONTROL</li> <li>• APPLICATION</li> <li>• BP</li> <li>• CAT_CACHE</li> <li>• DATABASE</li> <li>• DFM</li> <li>• FCMBP</li> <li>• IMPORT_POOL</li> <li>• LOCK_MGR</li> <li>• MONITOR</li> <li>• OTHER</li> <li>• PACKAGE_CACHE</li> <li>• QUERY</li> <li>• SHARED_SORT</li> <li>• SORT</li> <li>• STATEMENT</li> <li>• STATISTICS</li> <li>• UTILITY</li> </ul>
POOL_CUR_SIZE	BIGINT	pool_cur_size - メモリー・プールの現行サイズ
POOL_WATERMARK	BIGINT	pool_watermark - メモリー・プール水準点
POOL_CONFIG_SIZE	BIGINT	pool_config_size - メモリー・プールの構成済みサイズ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

表 317. SNAPAGENT\_MEMORY\_POOL 管理ビューおよび  
SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAP\_GET\_APPL\_INFO\_V95 表関数 - appl\_info 論理データ・グループのスナップショット情報の検索

SNAP\_GET\_APPL\_INFO\_V95 は、アプリケーション・スナップショットから、特に appl\_info 論理データ・グループのアプリケーション情報を戻します。

注: この表関数は使用すべきではなく、846 ページの『SNAP\_GET\_APPL\_INFO 表関数』に置き換えられました。

戻される可能性のある情報の完全なリストは、1377 ページの表 318 を参照してください。

### 構文

```

▶▶ SNAP_GET_APPL_INFO_V95 (—dbname— [ , member ] )

```

スキーマは SYSPROC です。

### 表関数パラメーター

#### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり dbname のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されません。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

dbname と member の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファ

イルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_APPL\_INFO\_V95 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_APPL\_INFO\_V95 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

接続中のデータベース・メンバー上のすべてのアプリケーションの状況を取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AGENT_ID,
       SUBSTR(APPL_NAME,1,10) AS APPL_NAME, APPL_STATUS
FROM TABLE(SNAP_GET_APPL_INFO_V95(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	AGENT_ID	APPL_NAME	APPL_STATUS
TOOLSDB	14	db2bp.exe	CONNECTED
SAMPLE	15	db2bp.exe	UOWEXEC
SAMPLE	8	javaw.exe	CONNECTED
SAMPLE	7	db2bp.exe	UOWWAIT

4 record(s) selected.

以下は、表関数の結果からの SELECT の実行時に入手できる内容について示しています。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AUTHORITY_LVL
FROM TABLE(SNAP_GET_APPL_INFO_V95(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	AUTHORITY_LVL
TESTDB	SYSADM(GROUP) + DBADM(USER) + CREATETAB(USER, GROUP) + BINDADD(USER, GROUP) + CONNECT(USER, GROUP) + CREATE_NOT_FENC(USER) + IMPLICIT_SCHEMA(USER, GROUP) + LOAD(USER) + CREATE_EXT_RT(USER) + QUIESCE_CONN(USER)
TESTDB	SYSADM(GROUP) + DBADM(USER) + CREATETAB(USER, GROUP) +

```

        BINDADD(USER, GROUP) + CONNECT(USER, GROUP) +
        CREATE_NOT_FENC(USER) + IMPLICIT_SCHEMA(USER, GROUP) +
        LOAD(USER) + CREATE_EXT_RT(USER) + QUIESCE_CONN(USER)
TESTDB  SYSADM(GROUP) + DBADM(USER) + CREATETAB(USER, GROUP) +
        BINDADD(USER, GROUP) + CONNECT(USER, GROUP) +
        CREATE_NOT_FENC(USER) + IMPLICIT_SCHEMA(USER, GROUP) +
        LOAD(USER) + CREATE_EXT_RT(USER) + QUIESCE_CONN(USER)

```

3 record(s) selected.

## 戻される情報

表 318. SNAP\_GET\_APPL\_INFO\_V95 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)

表 318. SNAP\_GET\_APPL\_INFO\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
APPL_STATUS	VARCHAR(22)	<p>appl_status - アプリケーション状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• BACKUP</li> <li>• COMMIT_ACT</li> <li>• COMP</li> <li>• CONNECTED</li> <li>• CONNECTPEND</li> <li>• CREATE_DB</li> <li>• DECOUPLED</li> <li>• DISCONNECTPEND</li> <li>• INTR</li> <li>• IOERROR_WAIT</li> <li>• LOAD</li> <li>• LOCKWAIT</li> <li>• QUIESCE_TABLESPACE</li> <li>• RECOMP</li> <li>• REMOTE_RQST</li> <li>• RESTART</li> <li>• RESTORE</li> <li>• ROLLBACK_ACT</li> <li>• ROLLBACK_TO_SAVEPOINT</li> <li>• TEND</li> <li>• THABRT</li> <li>• THCOMT</li> <li>• TPREP</li> <li>• UNLOAD</li> <li>• UOWEXEC</li> <li>• UOWWAIT</li> <li>• WAITFOR_REMOTE</li> </ul>
CODEPAGE_ID	BIGINT	codepage_id - アプリケーションで使用するコード・ページ ID
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 関連したエージェント数
COORD_NODE_NUM	SMALLINT	coord_node - コーディネーター・ノード



表 318. SNAP\_GET\_APPL\_INFO\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
AUTHORITY_LVL	VARCHAR(512)	<p>authority_bitmap - ユーザー許可レベル。</p> <p>このインターフェースは、sql.h で定義されたデータベース権限およびそれらのソースに基づくテキスト ID を戻し、その形式は次のとおりです。authority(source, ...) + authority(source, ...) + ...。権限のソースは複数でも構いません。USER、GROUP、または USER と GROUP のいずれかです。</p> <p>"authority" に指定可能な値は以下のとおりです。</p> <ul style="list-style-type: none"> <li>• ACCESSCTRL</li> <li>• BINDADD</li> <li>• CONNECT</li> <li>• CREATE_EXT_RT</li> <li>• CREATE_NOT_FENC</li> <li>• CREATETAB</li> <li>• DATAACCESS</li> <li>• DBADM</li> <li>• EXPLAIN</li> <li>• IMPLICIT_SCHEMA</li> <li>• LOAD</li> <li>• LIBADM</li> <li>• QUIESCE_CONN</li> <li>• SECADM</li> <li>• SQLADM</li> <li>• SYSADM</li> <li>• SYSCTRL</li> <li>• SYSMAINT</li> <li>• SYSMON</li> <li>• SYSQUIESCE</li> <li>• WLMADM</li> </ul> <p>"source" に指定可能な値は以下のとおりです。</p> <ul style="list-style-type: none"> <li>• USER - ユーザーに付与された権限、またはそのユーザーに付与されているロールに付与された権限。</li> <li>• GROUP - ユーザーが属するグループに付与された権限、またはユーザーが属するグループに付与されるロールに付与された権限。</li> </ul>
CLIENT_PID	BIGINT	client_pid - クライアント・プロセス ID

表 318. SNAP\_GET\_APPL\_INFO\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
COORD_AGENT_PID	BIGINT	coord_agent_pid - コーディネーター・エージェント
STATUS_CHANGE_TIME	TIMESTAMP	status_change_time - アプリケーション状況変更時刻
CLIENT_PLATFORM	VARCHAR(12)	<p>client_platform - クライアント・オペレーティング・プラットフォーム。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。</p> <ul style="list-style-type: none"> <li>• AIX</li> <li>• AIX64</li> <li>• AS400_DRDA</li> <li>• DOS</li> <li>• DYNIX</li> <li>• HP</li> <li>• HP64</li> <li>• HPIA</li> <li>• HPIA64</li> <li>• LINUX</li> <li>• LINUX390</li> <li>• LINUXIA64</li> <li>• LINUXPPC</li> <li>• LINUXPPC64</li> <li>• LINUXX8664</li> <li>• LINUXZ64</li> <li>• MAC</li> <li>• MVS_DRDA</li> <li>• NT</li> <li>• NT64</li> <li>• OS2</li> <li>• OS390</li> <li>• SCO</li> <li>• SGI</li> <li>• SNI</li> <li>• SUN</li> <li>• SUN64</li> <li>• 不明 (UNKNOWN)</li> <li>• UNKNOWN_DRDA</li> <li>• VM_DRDA</li> <li>• VSE_DRDA</li> <li>• WINDOWS</li> </ul>



## SNAP\_GET\_APPL\_V95 表関数 - appl 論理データ・グループのスナップショット情報の検索

SNAP\_GET\_APPL\_V95 は、アプリケーション・スナップショットから、特に appl 論理データ・グループのアプリケーション情報を戻します。

注: この表関数は使用すべきではなく、SNAPAPPL 管理ビューおよび SNAP\_GET\_APPL 表関数に置き換えられました。

戻される可能性のある情報の完全なリストは、1383 ページの表 319 を参照してください。

### 構文

```
▶▶ SNAP_GET_APPL_V95 ( (dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されません。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_APPL\_V95 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_APPL\_V95 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべてのアクティブ・データベースの各アプリケーションについて読み取りおよび書き込みが行われた行の詳細を取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AGENT_ID, ROWS_READ, ROWS_WRITTEN
FROM TABLE (SNAP_GET_APPL_V95(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	AGENT_ID	ROWS_READ	ROWS_WRITTEN
WSDB	679	0	0
WSDB	461	3	0
WSDB	460	4	0
TEST	680	4	0
TEST	455	6	0
TEST	454	0	0
TEST	453	50	0

## 戻される情報

表 319. SNAP\_GET\_APPL\_V95 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
UOW_LOG_SPACE_USED	BIGINT	uow_log_space_used - 作業単位ログ・スペース
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
STMT_HISTORY_LIST_SIZE	BIGINT	stmt_history_list_size - ステートメント履歴リストのサイズ

表 319. SNAP\_GET\_APPL\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファークール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファークール・データの物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファークールへのデータの書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファークール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファークール索引の物理読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファークール索引の書き込み
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファークール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファークール一時データの物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファークール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファークール一時索引の物理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファークール一時 XDA データの論理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファークール一時 XDA データの物理読み取り : モニター・エレメント
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファークール XDA データの論理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファークール XDA データの物理読み取り
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファークール XDA データの書き込み
POOL_READ_TIME	BIGINT	pool_read_time - バッファークール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファークール物理書き込み時間の合計
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み

表 319. SNAP\_GET\_APPL\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不能プリフェッチ・ページ
LOCKS_HELD	BIGINT	locks_held - ロック保持数
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間
LOCK_ESCALS	BIGINT	lock_escalations - ロック・エスカレーション数
X_LOCK_ESCALS	BIGINT	x_lock_escalations - 排他ロック・エスカレーション数
DEADLOCKS	BIGINT	deadlocks - デッドロック検出数
TOTAL_SORTS	BIGINT	total_sorts - ソート合計
TOTAL_SORT_TIME	BIGINT	total_sort_time - ソート時間合計
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
COMMIT_SQL_STMTS	BIGINT	commit_sql_stmts - 試行されたコミット・ステートメント
ROLLBACK_SQL_STMTS	BIGINT	rollback_sql_stmts - 試行されたロールバック・ステートメント
DYNAMIC_SQL_STMTS	BIGINT	dynamic_sql_stmts - 試行された動的 SQL ステートメント
STATIC_SQL_STMTS	BIGINT	static_sql_stmts - 試行された静的 SQL ステートメント
FAILED_SQL_STMTS	BIGINT	failed_sql_stmts - 失敗したステートメント操作
SELECT_SQL_STMTS	BIGINT	select_sql_stmts - 実行された選択 SQL ステートメント
DDL_SQL_STMTS	BIGINT	ddl_sql_stmts - データ定義言語 (DDL) SQL ステートメント
UID_SQL_STMTS	BIGINT	uid_sql_stmts - 実行された更新/挿入/削除 SQL ステートメント
INT_AUTO_REBINDS	BIGINT	int_auto_rebinds - 内部自動再バインド
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 削除された内部行数

表 319. SNAP\_GET\_APPL\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 更新された内部行数
INT_COMMITS	BIGINT	int_commits - 内部コミット数
INT_ROLLBACKS	BIGINT	int_rollback - 内部ロールバック数
INT_DEADLOCK_ROLLBACKS	BIGINT	int_deadlock_rollback - デッドロックによる内部ロールバック数
ROWS_DELETED	BIGINT	rows_deleted - 削除行数
ROWS_INSERTED	BIGINT	rows_inserted - 挿入行数
ROWS_UPDATED	BIGINT	rows_updated - 更新行数
ROWS_SELECTED	BIGINT	rows_selected - 選択行数
BINDS_PRECOMPILES	BIGINT	binds_precompiles - 試行されたバインド/プリコンパイル
OPEN_REM_CURS	BIGINT	open_rem_curs - 開かれているリモート・カーソル
OPEN_REM_CURS_BLK	BIGINT	open_rem_curs_blk - 開かれているリモート・ブロック・カーソル
REJ_CURS_BLK	BIGINT	rej_curs_blk - リジェクトされたブロック・カーソル要求
ACC_CURS_BLK	BIGINT	acc_curs_blk - 受け入れられたブロック・カーソル要求
SQL_REQS_SINCE_COMMIT	BIGINT	sql_reqs_since_commit - 最終コミット後の SQL 要求数
LOCK_TIMEOUTS	BIGINT	lock_timeouts - ロック・タイムアウト数
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 挿入された内部行数
OPEN_LOC_CURS	BIGINT	open_loc_curs - 開かれているローカル・カーソル
OPEN_LOC_CURS_BLK	BIGINT	open_loc_curs_blk - 開かれているローカル・ブロック・カーソル
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - パッケージ・キャッシュ参照
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - パッケージ・キャッシュ挿入
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - カタログ・キャッシュ参照数
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - カタログ・キャッシュ挿入数
CAT_CACHE_OVERFLOW	BIGINT	cat_cache_overflows - カタログ・キャッシュ・オーバーフロー数
NUM_AGENTS	BIGINT	num_agents - ステートメントで作動しているエージェントの数



表 319. SNAP\_GET\_APPL\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
AGENTS_STOLEN	BIGINT	agents_stolen - スチールされたエージェント
ASSOCIATED_AGENTS_TOP	BIGINT	associated_agents_top - 関連エージェント最大数
APPL_PRIORITY	BIGINT	appl_priority - アプリケーション・エージェント優先順位
APPL_PRIORITY_TYPE	VARCHAR(16)	appl_priority_type - アプリケーション優先順位タイプ。このインターフェースは、sqlmon.h 内の定義に基づいてテキスト ID を戻します。それは、次のうちの 1 つです。 <ul style="list-style-type: none"> <li>• DYNAMIC_PRIORITY</li> <li>• FIXED_PRIORITY</li> </ul>
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間
APPL_SECTION_LOOKUPS	BIGINT	appl_section_lookups - セクションの参照回数
APPL_SECTION_INSERTS	BIGINT	appl_section_inserts - セクション挿入数
LOCKS_WAITING	BIGINT	locks_waiting - ロックで待機中の現行エージェント
TOTAL_HASH_JOINS	BIGINT	total_hash_joins - ハッシュ結合の合計
TOTAL_HASH_LOOPS	BIGINT	total_hash_loops - ハッシュ・ループの合計
HASH_JOIN_OVERFLOW	BIGINT	hash_join_overflows - ハッシュ結合のオーバーフロー
HASH_JOIN_SMALL_OVERFLOW	BIGINT	hash_join_small_overflows - ハッシュ結合の短精度オーバーフロー
APPL_IDLE_TIME	BIGINT	appl_idle_time - アプリケーション・アイドル時間
UOW_LOCK_WAIT_TIME	BIGINT	uow_lock_wait_time - ロック待機中の作業単位の合計時間

表 319. SNAP\_GET\_APPL\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
UOW_COMP_STATUS	VARCHAR(14)	uow_comp_status - 作業単位完了状況。このインターフェースは、sqlmon.h 内の定義に基づいてテキスト ID を戻します。それは、次のうちの 1 つです。 <ul style="list-style-type: none"> <li>• APPL_END</li> <li>• UOWABEND</li> <li>• UOWCOMMIT</li> <li>• UOWDEADLOCK</li> <li>• UOWLOCKTIMEOUT</li> <li>• UOWROLLBACK</li> <li>• UOWUNKNOWN</li> </ul>
AGENT_USR_CPU_TIME_S	BIGINT	agent_usr_cpu_time - エージェントが使用したユーザー CPU 時間 (秒単位)*
AGENT_USR_CPU_TIME_MS	BIGINT	agent_usr_cpu_time - エージェントが使用したユーザー CPU 時間 (小数部、マイクロ秒単位)*
AGENT_SYS_CPU_TIME_S	BIGINT	agent_sys_cpu_time - エージェントが使用したシステム CPU 時間 (秒単位)*
AGENT_SYS_CPU_TIME_MS	BIGINT	agent_sys_cpu_time - エージェントが使用したシステム CPU 時間 (小数部、マイクロ秒単位)*
APPL_CON_TIME	TIMESTAMP	appl_con_time - 接続要求開始タイム・スタンプ
CONN_COMPLETE_TIME	TIMESTAMP	conn_complete_time - 接続要求完了タイム・スタンプ
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ
UOW_START_TIME	TIMESTAMP	uow_start_time - 作業単位開始タイム・スタンプ
UOW_STOP_TIME	TIMESTAMP	uow_stop_time - 作業単位停止タイム・スタンプ
PREV_UOW_STOP_TIME	TIMESTAMP	prev_uow_stop_time - 直前の作業単位完了タイム・スタンプ
UOW_ELAPSED_TIME_S	BIGINT	uow_elapsed_time - 最新の作業単位の経過時間 (秒単位)*
UOW_ELAPSED_TIME_MS	BIGINT	uow_elapsed_time - 最新の作業単位の経過時間 (小数部、マイクロ秒単位)*
ELAPSED_EXEC_TIME_S	BIGINT	elapsed_exec_time - ステートメント実行経過時間 (秒単位)*

表 319. SNAP\_GET\_APPL\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
ELAPSED_EXEC_TIME_MS	BIGINT	elapsed_exec_time - ステートメント実行経過時間 (小数部、マイクロ秒単位)*
INBOUND_COMM_ADDRESS	VARCHAR(32)	inbound_comm_address - インバウンド通信アドレス
LOCK_TIMEOUT_VAL	BIGINT	lock_timeout_val - ロック・タイムアウト (秒)
PRIV_WORKSPACE_NUM_OVERFLOWS	BIGINT	priv_workspace_num_overflows - 専用ワークスペースのオーバーフロー回数
PRIV_WORKSPACE_SECTION_INSERTS	BIGINT	priv_workspace_section_inserts - 専用ワークスペース・セクション挿入
PRIV_WORKSPACE_SECTION_LOOKUPS	BIGINT	priv_workspace_section_lookups - 専用ワークスペース・セクションの参照
PRIV_WORKSPACE_SIZE_TOP	BIGINT	priv_workspace_size_top - 専用ワークスペースの最大サイズ
SHR_WORKSPACE_NUM_OVERFLOWS	BIGINT	shr_workspace_num_overflows - 共有ワークスペースのオーバーフロー回数
SHR_WORKSPACE_SECTION_INSERTS	BIGINT	shr_workspace_section_inserts - 共有ワークスペース・セクション挿入数
SHR_WORKSPACE_SECTION_LOOKUPS	BIGINT	shr_workspace_section_lookups - 共有ワークスペース・セクションの参照回数
SHR_WORKSPACE_SIZE_TOP	BIGINT	shr_workspace_size_top - 最大共有ワークスペース・サイズ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
CAT_CACHE_SIZE_TOP	BIGINT	cat_cache_size_top - カタログ・キャッシュ最高水準点
TOTAL_OLAP_FUNCS	BIGINT	total_olap_funcs - OLAP 関数の合計数モニター・エレメント
OLAP_FUNC_OVERFLOWS	BIGINT	olap_func_overflows - OLAP 関数のオーバーフロー・モニター・エレメント
<p>* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する <code>_S</code> で終わっている列で報告されている整数秒と、このモニター・エレメントに関する <code>_MS</code> で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません: <math>(\text{monitor-element-name}_S \times 1,000,000 + \text{monitor-element-name}_MS) \div 1,000,000</math>. 例えば、<math>(\text{ELAPSED\_EXEC\_TIME}_S \times 1,000,000 + \text{ELAPSED\_EXEC\_TIME}_MS) \div 1,000,000</math>.</p>		

## SNAP\_GET\_BP\_V95 表関数 - bufferpool 論理グループのスナップショット情報の検索

SNAP\_GET\_BP\_V95 は、バッファ・プール・スナップショットから、特に bufferpool 論理データ・グループのバッファ・プール情報を戻します。

注: この表関数は使用すべきではなく、864 ページの『SNAP\_GET\_BP 表関数』に置き換えられました。

戻される可能性のある情報の完全なリストは、1391 ページの表 320 を参照してください。

### 構文

```
▶▶ SNAP_GET_BP_V95 ( (dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されません。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_BP\_V95 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_BP\_V95 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベース・メンバーのすべてのアクティブ・データベースのすべてのバッファ・プールについて、物理および論理読み取りの合計を取得します。

```
SELECT SUBSTR(T.DB_NAME,1,10) AS DB_NAME,
SUBSTR(T.BP_NAME,1,20) AS BP_NAME,
(T.POOL_DATA_L_READS+T.POOL_INDEX_L_READS) AS TOTAL_LOGICAL_READS,
(T.POOL_DATA_P_READS+T.POOL_INDEX_P_READS) AS TOTAL_PHYSICAL_READS,
T.DBPARTITIONNUM
FROM TABLE(SNAP_GET_BP_V95(CAST(NULL AS VARCHAR(128))), -1)) AS T
```

以下はこの照会の出力例です。

```
DB_NAME      BP_NAME      TOTAL_LOGICAL_READS  ...
-----
SAMPLE      IBMDEFAULTBP      0 ...
TOOLSDB     IBMDEFAULTBP      0 ...
TOOLSDB     BP32K0000         0 ...
3 record(s) selected.
```

この照会からの出力 (続き)。

```
... TOTAL_PHYSICAL_READS DBPARTITIONNUM
... -----
...                0                0
...                0                0
...                0                0
```

## 戻される情報

表 320. SNAP\_GET\_BP\_V95 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
BP_NAME	VARCHAR(128)	bp_name - バッファ・プール名
DB_NAME	VARCHAR(128)	db_name - データベース名

表 320. SNAP\_GET\_BP\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DB_PATH	VARCHAR(1024)	db_path - データベース・パス
INPUT_DB_ALIAS	VARCHAR(128)	input_db_alias - 入力データベース別名
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込み
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDA データの論理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDA データの物理読み取り
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDA データの書き込み
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - バッファ・プール非同期データ読み取り
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - バッファ・プール非同期データ書き込み
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - バッファ・プール非同期索引読み取り
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - バッファ・プール非同期索引書き込み
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - バッファ・プール非同期 XDA データ読み取り
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - バッファ・プール非同期 XDA データ書き込み
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - バッファ・プール非同期読み取り時間
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - バッファ・プール非同期書き込み時間

表 320. SNAP\_GET\_BP\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - バッファ・プール非同期読み取り要求
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - バッファ・プール非同期索引読み取り要求
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - バッファ・プール非同期 XDA 読み取り要求
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不能プリフェッチ・ページ
FILES_CLOSED	BIGINT	files_closed - 閉じられたデータベース・ファイル
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り : モニター・エレメント
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - バッファ・プールの非ビクティム・バッファ数

表 320. SNAP\_GET\_BP\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
PAGES_FROM_BLOCK_IOS	BIGINT	pages_from_block_ios - ブロック入出力によって読み取られたページ数の合計
PAGES_FROM_VECTORED_IOS	BIGINT	pages_from_vectored_ios - ベクトル化入出力によって読み取られたページ数の合計
VECTORED_IOS	BIGINT	vectored_ios - ベクトル化入出力要求数
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

## SNAP\_GET\_CONTAINER\_V91 表関数 - tablespace\_container 論理データ・グループ・スナップショット情報の検索

SNAP\_GET\_CONTAINER\_V91 は、tablespace\_container 論理データ・グループからの表スペース・スナップショット情報を戻します。

注: この表関数は使用すべきではなく、SNAPCONTAINER 管理ビューおよび SNAP\_GET\_CONTAINER 表関数に置き換えられました。

戻される可能性のある情報の完全なリストは、1396 ページの表 321 を参照してください。

### 構文

```

▶▶ SNAP_GET_CONTAINER_V91 ( ( dbname ) [ , dbpartitionnum ] )

```

スキーマは SYSPROC です。

### 表関数パラメーター

#### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### dbpartitionnum

有効なデータベース・パーティション番号を指定する、タイプ INTEGER のオプション入力引数。現行のデータベース・パーティションには -1、すべてのアクティブなデータベース・パーティションの集約には -2 を指定します。

dbname が NULL に設定されておらず、dbpartitionnum が NULL に設定されている場合、dbpartitionnum には暗黙的に -1 が設定されます。この入力オプション



が使用されない場合、つまり、*dbname* のみが指定されている場合、データはすべてのアクティブなデータベース・パーティションから戻されます。アクティブなデータベース・パーティションとは、アプリケーションからデータベースに接続して、これを使用することが可能なパーティションのことです。

*dbname* および *dbpartitionnum* の両方が NULL に設定された場合、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りを試行します。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合には、現在接続されているデータベースのスナップショットとデータベース・パーティション番号を SNAP\_GET\_CONTAINER\_V91 表関数が取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_CONTAINER\_V91 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベース・パーティション上で現在接続されているデータベースの表スペース・コンテナの詳細を検索します。

```
SELECT SNAPSHOT_TIMESTAMP, TBSP_NAME, TBSP_ID, CONTAINER_NAME,  
       CONTAINER_ID, CONTAINER_TYPE, ACCESSIBLE  
FROM TABLE(SNAP_GET_CONTAINER_V91(' ', -1)) AS T
```

以下はこの照会の出力例です。

SNAPSHOT_TIMESTAMP	TBSP_NAME	TBSP_ID	...
2005-04-25-14.42.10.899253	SYSCATSPACE	0	...
2005-04-25-14.42.10.899253	TEMPSPACE1	1	...
2005-04-25-14.42.10.899253	USERSPACE1	2	...
2005-04-25-14.42.10.899253	SYSTOOLSPACE	3	...
2005-04-25-14.42.10.899253	MYTEMP	4	...
2005-04-25-14.42.10.899253	WHATSNEWTMPSPACE	5	...

この照会からの出力 (続き)。

```

... CONTAINER_NAME                                CONTAINER_ID ...
... -----
... D:¥DB2¥NODE0000¥SQL00002¥SQLT0000.0          0 ...
... D:¥DB2¥NODE0000¥SQL00002¥SQLT0001.0          0 ...
... D:¥DB2¥NODE0000¥SQL00002¥SQLT0002.0          0 ...
... D:¥DB2¥NODE0000¥SQL00002¥SYSTOOLSPACE         0 ...
... D:¥DB2¥NODE0000¥SQL003                        0 ...
... d:¥DGTsWhatsNewContainer                      0 ...

```

この照会からの出力 (続き)。

```

... CONTAINER_TYPE ACCESSIBLE
... -----
... CONT_PATH                                     1
... CONT_PATH                                     1
... CONT_PATH                                     1
... CONT_PATH                                     1
... CONT_PATH                                     1
... CONT_PATH                                     1

```

## 戻される情報

注: ファイル・システム情報を戻すためには、**bufferpool** データベース・マネージャーのモニター・スイッチをオンにする必要があります。

表 321. *SNAP\_GET\_CONTAINER\_V91* 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
CONTAINER_NAME	VARCHAR(256)	container_name - コンテナ名
CONTAINER_ID	BIGINT	container_id - コンテナ ID
CONTAINER_TYPE	VARCHAR(16)	container_type - コンテナ・タイプ。これは、 <code>sqlutil.h</code> での定義を基にしたテキスト ID です。以下のいずれかとなります。 <ul style="list-style-type: none"> <li>• DISK_EXTENT_TAG</li> <li>• DISK_PAGE_TAG</li> <li>• FILE_EXTENT_TAG</li> <li>• FILE_PAGE_TAG</li> <li>• PATH</li> </ul>
TOTAL_PAGES	BIGINT	container_total_pages - コンテナ内の合計ページ数
USABLE_PAGES	BIGINT	container_usable_pages - コンテナ内の使用可能なページ数
ACCESSIBLE	SMALLINT	container_accessible - コンテナのアクセス可能性
STRIPE_SET	BIGINT	container_stripe_set - ストライプ・セット

表 321. SNAP\_GET\_CONTAINER\_V91 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
FS_ID	VARCHAR(22)	fs_id - 固有のファイル・システム識別番号
FS_TOTAL_SIZE	BIGINT	fs_total_size - ファイル・システムの合計サイズ
FS_USED_SIZE	BIGINT	fs_used_size - ファイル・システム上で使用されるスペースの量

## SNAPDB\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DB\_MEMORY\_POOL 表関数 - データベース・レベルのメモリー使用量情報の検索

SNAPDB\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DB\_MEMORY\_POOL 表関数は、データベース・レベルでのメモリー使用量についての情報を戻します (UNIX プラットフォームの場合のみ)。

SNAPDB\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DB\_MEMORY\_POOL 表関数は、データベース・レベルでのメモリー使用量についての情報を戻します (UNIX プラットフォームの場合のみ)。

**重要:** バージョン 9.7 フィックスパック 5 以降、SNAPDB\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DB\_MEMORY\_POOL 表関数は使用すべきでなくなり、604 ページの『MON\_GET\_MEMORY\_POOL - メモリー・プール情報の取得』および 607 ページの『MON\_GET\_MEMORY\_SET - メモリー・セット情報の取得』に置き換えられました。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPDB\_MEMORY\_POOL 管理ビュー』
- 1399 ページの『SNAP\_GET\_DB\_MEMORY\_POOL 表関数』

### SNAPDB\_MEMORY\_POOL 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関するデータベース・レベルのメモリー使用量情報を取得することができます。

SNAPDB\_MEMORY\_POOL 管理ビューを SNAPDB、SNAPDETAILLOG、SNAPHADR、ADMIN\_GET\_STORAGE\_PATHS、および MON\_GET\_HADR とともに使用することにより、**GET SNAPSHOT FOR DATABASE ON database-alias** CLP コマンドと同等の情報が得られます。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1401 ページの表 322 を参照してください。

## 許可

以下のいずれかの権限が必要です。

- SNAPDB\_MEMORY\_POOL 管理ビューに対する SELECT 特権
- SNAPDB\_MEMORY\_POOL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_DB\_MEMORY\_POOL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続中のデータベースである SAMPLE のメモリー・プールおよびそれらの現在のサイズのリストを取得します。

```
SELECT POOL_ID, POOL_CUR_SIZE FROM SYSIBMADM.SNAPDB_MEMORY_POOL
```

以下はこの照会の出力例です。

```
POOL_ID          POOL_CUR_SIZE
-----
UTILITY                32768
PACKAGE_CACHE        475136
CAT_CACHE             65536
BP                    2097152
BP                    1081344
BP                    540672
BP                    278528
BP                    147456
BP                    81920
LOCK_MGR              294912
DATABASE              3833856
OTHER                  0
```

12 record(s) selected.

## SNAP\_GET\_DB\_MEMORY\_POOL 表関数

SNAP\_GET\_DB\_MEMORY\_POOL 表関数は SNAPDB\_MEMORY\_POOL 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_DB\_MEMORY\_POOL 表関数を SNAP\_GET\_DB、SNAP\_GET\_DETAILLOG、SNAP\_GET\_HADR、および ADMIN\_GET\_STORAGE\_PATHS 表関数とともに使用すると、**GET SNAPSHOT FOR ALL DATABASES CLP** コマンドに相当する情報が提供されます。

戻される可能性のある情報の完全なリストは、1401 ページの表 322 を参照してください。

### 構文

```
▶▶ SNAP_GET_DB_MEMORY_POOL ( ( dbname ) [ , member ] ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されません。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DB\_MEMORY\_POOL 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得しません。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DB\_MEMORY\_POOL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべてのデータベースのメモリー・プールおよびそれらの現在のサイズのリストを取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, POOL_ID, POOL_CUR_SIZE
FROM TABLE(SNAPSHOT_GET_DB_MEMORY_POOL
(CAST(NULL AS VARCHAR(128))), -1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	POOL_ID	POOL_CUR_SIZE
TESTDB	UTILITY	65536
TESTDB	PACKAGE_CACHE	851968
TESTDB	CAT_CACHE	65536
TESTDB	BP	35913728
TESTDB	BP	589824
TESTDB	BP	327680
TESTDB	BP	196608
TESTDB	BP	131072
TESTDB	SHARED_SORT	65536
TESTDB	LOCK_MGR	10092544
TESTDB	DATABASE	4980736
TESTDB	OTHER	196608
SAMPLE	UTILITY	65536
SAMPLE	PACKAGE_CACHE	655360
SAMPLE	CAT_CACHE	131072
SAMPLE	BP	4325376
SAMPLE	BP	589824
SAMPLE	BP	327680
SAMPLE	BP	196608
SAMPLE	BP	131072
SAMPLE	SHARED_SORT	0
SAMPLE	LOCK_MGR	655360
SAMPLE	DATABASE	4653056
SAMPLE	OTHER	196608

24 record(s) selected.

## 戻される情報

表 322. SNAPDB\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DB\_MEMORY\_POOL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
POOL_ID	VARCHAR(14)	pool_id - メモリー・プール ID。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• APP_GROUP</li> <li>• APPL_CONTROL</li> <li>• APPLICATION</li> <li>• BP</li> <li>• CAT_CACHE</li> <li>• DATABASE</li> <li>• DFM</li> <li>• FCMBP</li> <li>• IMPORT_POOL</li> <li>• LOCK_MGR</li> <li>• MONITOR</li> <li>• OTHER</li> <li>• PACKAGE_CACHE</li> <li>• QUERY</li> <li>• SHARED_SORT</li> <li>• SORT</li> <li>• STATEMENT</li> <li>• STATISTICS</li> <li>• UTILITY</li> </ul>
POOL_SECONDARY_ID	VARCHAR(32)	pool_secondary_id - メモリー・プール 2 次 ID
POOL_CUR_SIZE	BIGINT	pool_cur_size - メモリー・プールの現行サイズ
POOL_WATERMARK	BIGINT	pool_watermark - メモリー・プール水準点
POOL_CONFIG_SIZE	BIGINT	pool_config_size - メモリー・プールの構成済みサイズ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAP\_GET\_DBM\_V95 表関数 - dbm 論理グループ・スナップショット情報の検索

SNAP\_GET\_DBM\_V95 は、DB2 データベース・マネージャー (dbm) 論理グループのスナップショット・モニター情報を戻します。

注: この表関数は使用すべきではなく、SNAPDBM 管理ビューおよび SNAP\_GET\_DBM 表関数に置き換えられました。

戻される可能性のある情報の完全なリストは、1403 ページの表 323 を参照してください。

### 構文

```
▶▶ SNAP_GET_DBM_V95 ( [member] ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャーにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DBM\_V95 表関数はメモリーからスナップショットを呼び出します。

### 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DBM\_V95 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM



## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

データベース・メンバー番号 2 の開始時刻と現行状況を検索します。

```
SELECT DB2START_TIME, DB2_STATUS FROM TABLE(SNAP_GET_DBM_V95(2)) AS T
```

以下はこの照会の出力例です。

```
DB2START_TIME          DB2_STATUS
-----
2006-01-06-14.59.59.062798 ACTIVE
```

### 戻される情報

表 323. SNAP\_GET\_DBM\_V95 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
SORT_HEAP_ALLOCATED	BIGINT	sort_heap_allocated - 割り振られたソート・ヒープの合計
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - ポストしきい値ソート
PIPED_SORTS_REQUESTED	BIGINT	piped_sorts_requested - 要求されたパイプ・ソート数
PIPED_SORTS_ACCEPTED	BIGINT	piped_sorts_accepted - 受け入れられたパイプ・ソート
REM_CONS_IN	BIGINT	rem_cons_in - データベース・マネージャーへのリモート接続
REM_CONS_IN_EXEC	BIGINT	rem_cons_in_exec - データベース・マネージャーで実行中のリモート接続：モニター・エレメント
LOCAL_CONS	BIGINT	local_cons - ローカル接続
LOCAL_CONS_IN_EXEC	BIGINT	local_cons_in_exec - データベース・マネージャーで実行中のローカル接続：モニター・エレメント
CON_LOCAL_DBASES	BIGINT	con_local_dbases - 現行接続を使用したローカル・データベース
AGENTS_REGISTERED	BIGINT	agents_registered - 登録済みエージェント
AGENTS_WAITING_ON_TOKEN	BIGINT	agents_waiting_on_token - トークン待ちエージェント
DB2_STATUS	VARCHAR(12)	db2_status - DB2 インスタンス状況。  このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。  <ul style="list-style-type: none"> <li>• ACTIVE</li> <li>• QUIESCE_PEND</li> <li>• QUIESCED</li> </ul>

表 323. SNAP\_GET\_DBM\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
AGENTS_REGISTERED_TOP	BIGINT	agents_registered_top - エージェント最大登録数
AGENTS_WAITING_TOP	BIGINT	agents_waiting_top - エージェント最大待機数
COMM_PRIVATE_MEM	BIGINT	comm_private_mem - コミット済み専用メモリー
IDLE_AGENTS	BIGINT	idle_agents - アイドル・エージェント数
AGENTS_FROM_POOL	BIGINT	agents_from_pool - プールから割り当てられたエージェント
AGENTS_CREATED_EMPTY_POOL	BIGINT	agents_created_empty_pool - エージェント・プールが空のために作成されたエージェント
COORD_AGENTS_TOP	BIGINT	coord_agents_top - コーディネーター・エージェント最大数
MAX_AGENT_OVERFLOWES	BIGINT	max_agent_overflows - 最大エージェント・オーバーフロー回数
AGENTS_STOLEN	BIGINT	agents_stolen - スチールされたエージェント
GW_TOTAL_CONS	BIGINT	gw_total_cons - DB2 Connect の接続試行合計回数
GW_CUR_CONS	BIGINT	gw_cur_cons - DB2 Connect の現在の接続数
GW_CONS_WAIT_HOST	BIGINT	gw_cons_wait_host - ホストの応答を待機している接続の数
GW_CONS_WAIT_CLIENT	BIGINT	gw_cons_wait_client - クライアントの要求送信を待機している接続の数
POST_THRESHOLD_HASH_JOINS	BIGINT	post_threshold_hash_joins - ハッシュ結合のしきい値
NUM_GW_CONN_SWITCHES	BIGINT	num_gw_conn_switches - 接続切り替え回数
DB2START_TIME	TIMESTAMP	db2start_time - データベース・マネージャー開始タイム・スタンプ
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ
NUM_NODES_IN_DB2_INSTANCE	INTEGER	num_nodes_in_db2_instance - データベース・パーティション内のノード数
PRODUCT_NAME	VARCHAR(32)	product_name - 製品名
SERVICE_LEVEL	VARCHAR(18)	service_level - サービス・レベル
SORT_HEAP_TOP	BIGINT	sort_heap_top - ソート専用ヒープの最高水準点
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
POST_THRESHOLD_OLAP_FUNCS	BIGINT	post_threshold_olap_funcs - OLAP 関数しきい値

## SNAPDBM\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DBM\_MEMORY\_POOL 表関数 - データベース・マ ネージャー・レベルのメモリー使用量情報の検索

SNAPDBM\_MEMORY\_POOL 管理ビューおよび

SNAP\_GET\_DBM\_MEMORY\_POOL 表関数は、データベース・マネージャーでのメモリー使用量についての情報を戻します。

**重要:** バージョン 9.7 フィックスパック 5 以降、SNAPDBM\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DBM\_MEMORY\_POOL 表関数は使用すべきでなくなり、604 ページの『MON\_GET\_MEMORY\_POOL - メモリー・プール情報の取得』および 607 ページの『MON\_GET\_MEMORY\_SET - メモリー・セット情報の取得』に置き換えられました。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPDBM\_MEMORY\_POOL 管理ビュー』
- 1406 ページの『SNAP\_GET\_DBM\_MEMORY\_POOL 表関数』

### SNAPDBM\_MEMORY\_POOL 管理ビュー

SNAPDBM\_MEMORY\_POOL 管理ビューを

SNAPDBM、SNAPFCM、SNAPFCM\_PART、および SNAPSWITCHES 管理ビューとともに使用すると、GET SNAPSHOT FOR DBM コマンドに相当するデータが提供されます。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1408 ページの表 324 を参照してください。

### 許可

以下のいずれかの権限が必要です。

- SNAPDBM\_MEMORY\_POOL 管理ビューに対する SELECT 特権
- SNAPDBM\_MEMORY\_POOL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_DBM\_MEMORY\_POOL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

接続中のデータベースのデータベース・マネージャーのメモリー・プールおよびそれらの現在のサイズのリストを取得します。

```
SELECT POOL_ID, POOL_CUR_SIZE FROM SNAPDBM_MEMORY_POOL
```

以下はこの照会の出力例です。

POOL_ID	POOL_CUR_SIZE
MONITOR	65536
OTHER	29622272
FCMBP	57606144
...	

## SNAP\_GET\_DBM\_MEMORY\_POOL 表関数

SNAP\_GET\_DBM\_MEMORY\_POOL 表関数は SNAPDBM\_MEMORY\_POOL 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーを対象とした情報を検索することができます。

SNAP\_GET\_DBM\_MEMORY\_POOL 表関数を SNAP\_GET\_DBM、SNAP\_GET\_FCM、SNAP\_GET\_FCM\_PART、および SNAP\_GET\_SWITCHES 表関数とともに使用すると、GET SNAPSHOT FOR DBM コマンドに相当するデータが提供されます。

戻される可能性のある情報の完全なリストは、1408 ページの表 324 を参照してください。

### 構文

```
▶▶ SNAP_GET_DBM_MEMORY_POOL ( [member] ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

*member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャーにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DBM\_MEMORY\_POOL 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DBM\_MEMORY\_POOL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

接続中のデータベースのデータベース・マネージャーのすべてのデータベース・パーティションのメモリー・プールおよびそれらの現在のサイズのリストを取得します。

```
SELECT POOL_ID, POOL_CUR_SIZE, DBPARTITIONNUM
FROM TABLE(SYSPROC.SNAP_GET_DBM_MEMORY_POOL())
AS T ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

POOL_ID	POOL_CUR_SIZE	DBPARTITIONNUM
MONITOR	65536	0
OTHER	29622272	0
FCMBP	57606144	0
MONITOR	65536	1
OTHER	29425664	1
FCMBP	57606144	1
MONITOR	65536	2
OTHER	29425664	2
FCMBP	57606144	2

## 戻される情報

表 324. SNAPDBM\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DBM\_MEMORY\_POOL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
POOL_ID	VARCHAR(14)	pool_id - メモリー・プール ID。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• APP_GROUP</li> <li>• APPL_CONTROL</li> <li>• APPLICATION</li> <li>• BP</li> <li>• CAT_CACHE</li> <li>• DATABASE</li> <li>• DFM</li> <li>• FCMBP</li> <li>• IMPORT_POOL</li> <li>• LOCK_MGR</li> <li>• MONITOR</li> <li>• OTHER</li> <li>• PACKAGE_CACHE</li> <li>• QUERY</li> <li>• SHARED_SORT</li> <li>• SORT</li> <li>• STATEMENT</li> <li>• STATISTICS</li> <li>• UTILITY</li> </ul>
POOL_CUR_SIZE	BIGINT	pool_cur_size - メモリー・プールの現行サイズ
POOL_WATERMARK	BIGINT	pool_watermark - メモリー・プール水準点
POOL_CONFIG_SIZE	BIGINT	pool_config_size - メモリー・プールの構成済みサイズ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAP\_GET\_DB\_V97 表関数 - dbase 論理グループからのスナップショット情報の検索

SNAP\_GET\_DB\_V97 表関数は、データベース (dbase) 論理グループからのスナップショット情報を戻します。

注: SNAP\_GET\_DB\_V97 表関数は使用すべきでなく、SNAPDB 管理ビューおよび SNAP\_GET\_DB 表関数に置き換えられました。

戻される情報の完全なリストは、1411 ページの表 325 を参照してください。

### 構文

```
▶▶ SNAP_GET_DB_V97 ( ( -dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されず、アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DB\_V97 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

### 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DB\_V97 表関数に対する EXECUTE 特権

- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

例 1: 現在接続されているデータベースのすべてのデータベース・メンバーに渡る集約ビューとして、状況、プラットフォーム、ロケーション、および接続時間を取り出します。

```
SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,
       DB_LOCATION, DB_CONN_TIME FROM TABLE(SNAP_GET_DB_V97(' ', -2)) AS T
```

以下はこの照会の出力例です。

```
DB_NAME      DB_STATUS    SERVER_PLATFORM ...
-----
SAMPLE      ACTIVE       AIX64           ...
```

1 record(s) selected.

この照会からの出力 (続き)。

```
... DB_LOCATION DB_CONN_TIME
... -----
... LOCAL      2005-07-24-22.09.22.013196
```

例 2: 現在接続されているデータベースを含む同じインスタンス内にあるすべてのアクティブ・データベースのすべてのデータベース・メンバーに渡る集約ビューとして、状況、プラットフォーム、ロケーション、および接続時間を取り出します。

```
SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,
       DB_LOCATION, DB_CONN_TIME
       FROM TABLE(SNAP_GET_DB_V97(CAST (NULL AS VARCHAR(128)), -2)) AS T
```

以下はこの照会の出力例です。

```
DB_NAME      DB_STATUS    SERVER_PLATFORM ...
-----
TOOLSDB     ACTIVE       AIX64           ...
SAMPLE      ACTIVE       AIX64           ...
```

この照会からの出力 (続き)。

```
... DB_LOCATION DB_CONN_TIME
... -----
... LOCAL      2005-07-24-22.26.54.396335
... LOCAL      2005-07-24-22.09.22.013196
```



例 3: このルーチンは、データベースへの接続時にコマンド行で以下を呼び出すことにより使用できます。

```
SELECT TOTAL_OLAP_FUNCS, OLAP_FUNC_OVERFLOWS, ACTIVE_OLAP_FUNCS
FROM TABLE (SNAP_GET_DB_V97(' ', 0)) AS T
```

出力は次のようになります。

```
TOTAL_OLAP_FUNCS  OLAP_FUNC_OVERFLOWS  ACTIVE_OLAP_FUNCS
-----
                      7                      2                      1
```

1 record(s) selected.

例 4: ワークロードの実行後に、ユーザーは次の照会を表関数とともに使用できます。

```
SELECT STATS_CACHE_SIZE, STATS_FABRICATIONS, SYNC_RUNSTATS,
ASYNC_RUNSTATS, STATS_FABRICATE_TIME, SYNC_RUNSTATS_TIME
FROM TABLE (SNAP_GET_DB_V97('mytestdb', -1)) AS SNAPDB
```

```
STATS_CACHE_SIZE  STATS_FABRICATIONS  SYNC_RUNSTATS  ASYNC_RUNSTATS  ...
-----
                200                1                2                0  ...
```

Continued

```
...STATS_FABRICATE_TIME  SYNC_RUNSTATS_TIME
-----
...                2                32
```

1 record(s) selected.

例 5: 以下の例は、SNAP\_GET\_DB\_V97 表関数を使ってデータベースの状況を判別する方法を示しています。

```
SELECT SUBSTR
(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS
FROM table(SNAP_GET_DB_V97('hadrdb', 0))
```

```
DB_NAME          DB_STATUS
-----
HADRDB          ACTIVE_STANDBY
```

## 戻される情報

表 325. SNAP\_GET\_DB\_V97 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
DB_PATH	VARCHAR(1024)	db_path - データベース・パス
INPUT_DB_ALIAS	VARCHAR(128)	input_db_alias - 入力データベース別名

表 325. SNAP\_GET\_DB\_V97 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DB_STATUS	VARCHAR(16)	<p>db_status - データベース状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• ACTIVE</li> <li>• QUIESCE_PEND</li> <li>• QUIESCED</li> <li>• ROLLFWD</li> <li>• ACTIVE_STANDBY - HADR データベースはスタンバイ・モードで、スタンバイ中の読み取りが有効になっています。</li> <li>• STANDBY - HADR データベースはスタンバイ・モードです (スタンバイ中の読み取りは有効になっていません)。</li> </ul>
CATALOG_PARTITION	SMALLINT	catalog_node - カタログ・ノード番号
CATALOG_PARTITION_NAME	VARCHAR(128)	catalog_node_name - カタログ・ノード・ネットワーク名

表 325. SNAP\_GET\_DB\_V97 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SERVER_PLATFORM	VARCHAR(12)	<p>server_platform - サーバーのオペレーティング・システム。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• AIX</li> <li>• AIX64</li> <li>• AS400_DRDA</li> <li>• DOS</li> <li>• DYNIX</li> <li>• HP</li> <li>• HP64</li> <li>• HPIA</li> <li>• HPIA64</li> <li>• LINUX</li> <li>• LINUX390</li> <li>• LINUXIA64</li> <li>• LINUXPPC</li> <li>• LINUXPPC64</li> <li>• LINUXX8664</li> <li>• LINUXZ64</li> <li>• MAC</li> <li>• MVS_DRDA</li> <li>• NT</li> <li>• NT64</li> <li>• OS2</li> <li>• OS390</li> <li>• SCO</li> <li>• SGI</li> <li>• SNI</li> <li>• SUN</li> <li>• SUN64</li> <li>• UNKNOWN</li> <li>• UNKNOWN_DRDA</li> <li>• VM_DRDA</li> <li>• VSE_DRDA</li> <li>• WINDOWS</li> </ul>

表 325. SNAP\_GET\_DB\_V97 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DB_LOCATION	VARCHAR(12)	db_location - データベース・ロケーション。 このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。  <ul style="list-style-type: none"> <li>• LOCAL</li> <li>• REMOTE</li> </ul>
DB_CONN_TIME	TIMESTAMP	db_conn_time- データベース・アクティブ化タイム・スタンプ
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ
LAST_BACKUP	TIMESTAMP	last_backup - 最終バックアップ・タイム・スタンプ
CONNECTIONS_TOP	BIGINT	connections_top - 同時接続の最大数
TOTAL_CONS	BIGINT	total_cons - データベース・アクティブ化以降の接続
TOTAL_SEC_CONS	BIGINT	total_sec_cons - 2 次接続
APPLS_CUR_CONS	BIGINT	appls_cur_cons - 現在接続されているアプリケーション
APPLS_IN_DB2	BIGINT	appls_in_db2 - データベースで現在実行中のアプリケーション
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 関連したエージェント数
AGENTS_TOP	BIGINT	agents_top - 作成されたエージェントの数
COORD_AGENTS_TOP	BIGINT	coord_agents_top - コーディネーター・エージェント最大数
LOCKS_HELD	BIGINT	locks_held - ロック保持数
LOCK_WAITS	BIGINT	lock_waits - ロック待機数
LOCK_WAIT_TIME	BIGINT	lock_wait_time - ロック待機中の時間
LOCK_LIST_IN_USE	BIGINT	lock_list_in_use - 使用中のロック・リスト・メモリーの合計
DEADLOCKS	BIGINT	deadlocks - デッドロック検出数
LOCK_ESCALS	BIGINT	lock_escals - ロック・エスカレーション数
X_LOCK_ESCALS	BIGINT	x_lock_escals - 排他ロック・エスカレーション数
LOCKS_WAITING	BIGINT	locks_waiting - ロックで待機中の現行エージェント
LOCK_TIMEOUTS	BIGINT	lock_timeouts - ロック・タイムアウト数
NUM_INDOUBT_TRANS	BIGINT	num_indoubt_trans - 未確定トランザクション数
SORT_HEAP_ALLOCATED	BIGINT	sort_heap_allocated - 割り振られたソート・ヒープの合計
SORT_SHRHEAP_ALLOCATED	BIGINT	sort_shrheap_allocated - 現在割り振られているソート共有ヒープ

表 325. SNAP\_GET\_DB\_V97 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SORT_SHRHEAP_TOP	BIGINT	sort_shrheap_top - ソート共有ヒープの最高水準点
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - ポスト共有しきい値ソート
TOTAL_SORTS	BIGINT	total_sorts - ソート合計
TOTAL_SORT_TIME	BIGINT	total_sort_time - ソート時間合計
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
ACTIVE_SORTS	BIGINT	active_sorts - アクティブ・ソート
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - バッファ・プール非同期データ読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ・プールへのデータの書き込み
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - バッファ・プール非同期データ書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - バッファ・プール非同期索引読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ・プール索引の書き込み
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - バッファ・プール非同期索引書き込み
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDAデータの物理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDAデータの論理読み取り
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ・プール XDAデータの書き込み

表 325. SNAP\_GET\_DB\_V97 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - バッファ・プール非同期 XDA データ読み取り
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - バッファ・プール非同期 XDA データ書き込み
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDA データの物理読み取り : モニター・エレメント
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDA データの論理読み取り
POOL_READ_TIME	BIGINT	pool_read_time - バッファ・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ・プール物理書き込み時間の合計
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - バッファ・プール非同期読み取り時間
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - バッファ・プール非同期書き込み時間
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - バッファ・プール非同期読み取り要求
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - バッファ・プール非同期索引読み取り要求
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - バッファ・プール非同期 XDA 読み取り要求
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - バッファ・プールの非ビクティム・バッファ数
POOL_LSN_GAP_CLNS	BIGINT	pool_lsn_gap_clns - 起動されたバッファ・プール・ログ・スペース・クリーナー
POOL_DRTY_PG_STEAL_CLNS	BIGINT	pool_drty_pg_steal_clns - 起動されたバッファ・プール・ビクティム・ページ・クリーナー
POOL_DRTY_PG_THRSH_CLNS	BIGINT	pool_drty_pg_thrsh_clns - 起動されたバッファ・プールしきい値クリーナー
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - プリフェッチ待ち時間
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不能プリフェッチ・ページ
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間

表 325. SNAP\_GET\_DB\_V97 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
FILES_CLOSED	BIGINT	files_closed - 閉じられたデータベース・ファイル
ELAPSED_EXEC_TIME_S	BIGINT	elapsed_exec_time - ステートメント実行経過時間
ELAPSED_EXEC_TIME_MS	BIGINT	elapsed_exec_time - ステートメント実行経過時間
COMMIT_SQL_STMTS	BIGINT	commit_sql_stmts - 試行されたコミット・ステートメント
ROLLBACK_SQL_STMTS	BIGINT	rollback_sql_stmts - 試行されたロールバック・ステートメント
DYNAMIC_SQL_STMTS	BIGINT	dynamic_sql_stmts - 試行された動的 SQL ステートメント
STATIC_SQL_STMTS	BIGINT	static_sql_stmts - 試行された静的 SQL ステートメント
FAILED_SQL_STMTS	BIGINT	failed_sql_stmts - 失敗したステートメント操作
SELECT_SQL_STMTS	BIGINT	select_sql_stmts - 実行された選択 SQL ステートメント
UID_SQL_STMTS	BIGINT	uid_sql_stmts - 実行された更新/挿入/削除 SQL ステートメント
DDL_SQL_STMTS	BIGINT	ddl_sql_stmts - データ定義言語 (DDL) SQL ステートメント
INT_AUTO_REBINDS	BIGINT	int_auto_rebinds - 内部自動再バインド
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 削除された内部行数
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 挿入された内部行数
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 更新された内部行数
INT_COMMITS	BIGINT	int_commits - 内部コミット数
INT_ROLLBACKS	BIGINT	int_rollback - 内部ロールバック数
INT_DEADLOCK_ROLLBACKS	BIGINT	int_deadlock_rollback - デッドロックによる内部ロールバック数
ROWS_DELETED	BIGINT	rows_deleted - 削除行数
ROWS_INSERTED	BIGINT	rows_inserted - 挿入行数
ROWS_UPDATED	BIGINT	rows_updated - 更新行数
ROWS_SELECTED	BIGINT	rows_selected - 選択行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
BINDS_PRECOMPILES	BIGINT	binds_precompiles - 試行されたバインド/プリコンパイル
TOTAL_LOG_AVAILABLE	BIGINT	total_log_available - 使用可能なログ合計
TOTAL_LOG_USED	BIGINT	total_log_used - 使用されているログ・スペースの合計
SEC_LOG_USED_TOP	BIGINT	sec_log_used_top - 使用された最大 2 次ログ・スペース

表 325. SNAP\_GET\_DB\_V97 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOT_LOG_USED_TOP	BIGINT	tot_log_used_top - 使用された最大合計ログ・スペース
SEC_LOGS_ALLOCATED	BIGINT	sec_logs_allocated - 現在割り振られている 2 次ログ
LOG_READS	BIGINT	log_reads - 読み取られたログ・ページの数
LOG_READ_TIME_S	BIGINT	log_read_time - ログ読み取り時間
LOG_READ_TIME_NS	BIGINT	log_read_time - ログ読み取り時間
LOG_WRITES	BIGINT	log_writes - 書き込まれたログ・ページの数
LOG_WRITE_TIME_S	BIGINT	log_write_time - ログ書き込み時間
LOG_WRITE_TIME_NS	BIGINT	log_write_time - ログ書き込み時間
NUM_LOG_WRITE_IO	BIGINT	num_log_write_io - ログ書き込み数
NUM_LOG_READ_IO	BIGINT	num_log_read_io - ログ読み取り数
NUM_LOG_PART_PAGE_IO	BIGINT	num_log_part_page_io - 部分ログ・ページ書き込み数
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - フル・ログ・バッファの回数
NUM_LOG_DATA_FOUND_IN_BUFFER	BIGINT	num_log_data_found_in_buffer - ログ・データがバッファにある回数
APPL_ID_OLDEST_XACT	BIGINT	appl_id_oldest_xact - 最も古いトランザクションを使用するアプリケーション
LOG_TO_REDO_FOR_RECOVERY	BIGINT	log_to_redo_for_recovery - リカバリーの場合に再実行されるログの量
LOG_HELD_BY_DIRTY_PAGES	BIGINT	log_held_by_dirty_pages - ダーティー・ページ別に計算されるログ・スペースの量
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - パッケージ・キャッシュ参照
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - パッケージ・キャッシュ挿入
PKG_CACHE_NUM_OVERFLOWS	BIGINT	pkg_cache_num_overflows - パッケージ・キャッシュ・オーバーフロー数
PKG_CACHE_SIZE_TOP	BIGINT	pkg_cache_size_top - パッケージ・キャッシュ最高水準点
APPL_SECTION_LOOKUPS	BIGINT	appl_section_lookups - セクションの参照回数
APPL_SECTION_INSERTS	BIGINT	appl_section_inserts - セクション挿入数
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - カタログ・キャッシュ参照数
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - カタログ・キャッシュ挿入数
CAT_CACHE_OVERFLOWS	BIGINT	cat_cache_overflows - カタログ・キャッシュ・オーバーフロー数
CAT_CACHE_SIZE_TOP	BIGINT	cat_cache_size_top - カタログ・キャッシュ最高水準点



表 325. SNAP\_GET\_DB\_V97 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
PRIV_WORKSPACE_SIZE_TOP	BIGINT	priv_workspace_size_top - 専用ワークスペースの最大サイズ
PRIV_WORKSPACE_NUM_OVERFLOWS	BIGINT	priv_workspace_num_overflows - 専用ワークスペースのオーバーフロー回数
PRIV_WORKSPACE_SECTION_INSERTS	BIGINT	priv_workspace_section_inserts - 専用ワークスペース・セクション挿入
PRIV_WORKSPACE_SECTION_LOOKUPS	BIGINT	priv_workspace_section_lookups - 専用ワークスペース・セクションの参照
SHR_WORKSPACE_SIZE_TOP	BIGINT	shr_workspace_size_top - 最大共有ワークスペース・サイズ
SHR_WORKSPACE_NUM_OVERFLOWS	BIGINT	shr_workspace_num_overflows - 共有ワークスペースのオーバーフロー回数
SHR_WORKSPACE_SECTION_INSERTS	BIGINT	shr_workspace_section_inserts - 共有ワークスペース・セクション挿入数
SHR_WORKSPACE_SECTION_LOOKUPS	BIGINT	shr_workspace_section_lookups - 共有ワークスペース・セクションの参照回数
TOTAL_HASH_JOINS	BIGINT	total_hash_joins - ハッシュ結合の合計
TOTAL_HASH_LOOPS	BIGINT	total_hash_loops - ハッシュ・ループの合計
HASH_JOIN_OVERFLOWS	BIGINT	hash_join_overflows - ハッシュ結合のオーバーフロー
HASH_JOIN_SMALL_OVERFLOWS	BIGINT	hash_join_small_overflows - ハッシュ結合の短精度オーバーフロー
POST_SHRTHRESHOLD_HASH_JOINS	BIGINT	post_shrthreshold_hash_joins - ポストしきい値ハッシュ結合
ACTIVE_HASH_JOINS	BIGINT	active_hash_joins - アクティブ・ハッシュ結合
NUM_DB_STORAGE_PATHS	BIGINT	num_db_storage_paths - 自動ストレージ・パスの数
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
SMALLEST_LOG_AVAIL_NODE	INTEGER	smallest_log_avail_node - 使用可能なログ・スペースが最小のノード
TOTAL_OLAP_FUNCS	BIGINT	total_olap_funcs - OLAP 関数の合計数
OLAP_FUNC_OVERFLOWS	BIGINT	olap_func_overflows - OLAP 関数のオーバーフロー
ACTIVE_OLAP_FUNCS	BIGINT	active_olap_funcs - アクティブな OLAP 関数
STATS_CACHE_SIZE	BIGINT	stats_cache_size 統計キャッシュのサイズ
STATS_FABRICATIONS	BIGINT	stats_fabrications 統計作成の総数
SYNC_RUNSTATS	BIGINT	sync_runstats 同期的 RUNSTATS アクティビティの総数
ASYNC_RUNSTATS	BIGINT	async_runstats 非同期的 RUNSTATS 要求の総数
STATS_FABRICATE_TIME	BIGINT	stats_fabricate_time 統計作成アクティビティに費やされた合計時間

表 325. SNAP\_GET\_DB\_V97 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
SYNC_RUNSTATS_TIME	BIGINT	sync_runstats_time 同期的 RUNSTATS アクティビティーに費やされた合計時間
NUM_THRESHOLD_VIOLATIONS	BIGINT	num_threshold_violations - しきい値違反の回数

## SNAP\_GET\_DETAILLOG\_V91 表関数 - detail\_log 論理データ・グループからのスナップショット情報の取得

SNAP\_GET\_DETAILLOG\_V91 表関数は、detail\_log 論理データ・グループからのスナップショット情報を戻します。

注: この表関数は使用すべきではなく、709 ページの『MON\_GET\_TRANSACTION\_LOG 表関数 - ログ情報の取得』に置き換えられました。

戻される情報の完全なリストは、1422 ページの表 326 を参照してください。

### 構文

```

--SNAP_GET_DETAILLOG_V91--(—dbname——, dbpartitionnum——)——

```

スキーマは SYSPROC です。

### 表関数パラメーター

#### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、LIST DATABASE DIRECTORY コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### dbpartitionnum

有効なデータベース・パーティション番号を指定する、タイプ INTEGER のオプション入力引数。現行のデータベース・パーティションには -1、すべてのアクティブなデータベース・パーティションの集約には -2 を指定します。  
 dbname が NULL に設定されておらず、dbpartitionnum が NULL に設定されている場合、dbpartitionnum には暗黙的に -1 が設定されます。この入力オプションが使用されない場合、つまり、dbname のみが指定されている場合、データはすべてのアクティブなデータベース・パーティションから戻されます。アクティブなデータベース・パーティションとは、アプリケーションからデータベースに接続して、これを使用することが可能なパーティションのことです。

*dbname* および *dbpartitionnum* の両方が NULL に設定された場合、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りを試行します。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合には、現在接続されているデータベースのスナップショットとデータベース・パーティション番号を SNAP\_GET\_DETAILLOG\_V91 表関数が取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DETAILLOG\_V91 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

データベース・パーティション 1 に関して、現在接続されているデータベースのログ情報を検索します。

```
SELECT SUBSTR(DB_NAME, 1, 8) AS DB_NAME, FIRST_ACTIVE_LOG,
       LAST_ACTIVE_LOG, CURRENT_ACTIVE_LOG, CURRENT_ARCHIVE_LOG
FROM TABLE(SNAP_GET_DETAILLOG_V91(' ', 1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	FIRST_ACTIVE_LOG	LAST_ACTIVE_LOG	...
TEST	0	8	...

1 record(s) selected.

この照会からの出力 (続き)。

...	CURRENT_ACTIVE_LOG	CURRENT_ARCHIVE_LOG
...	0	-

## 戻される情報

表 326. SNAP\_GET\_DETAILLOG\_V91 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
FIRST_ACTIVE_LOG	BIGINT	first_active_log - 先頭アクティブ・ログ・ファイル番号
LAST_ACTIVE_LOG	BIGINT	last_active_log - 最終アクティブ・ログ・ファイル番号
CURRENT_ACTIVE_LOG	BIGINT	current_active_log - 現行アクティブ・ログ・ファイル番号
CURRENT_ARCHIVE_LOG	BIGINT	current_archive_log - 現行アーカイブ・ログ・ファイル番号
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

## SNAP\_GET\_DYN\_SQL\_V95 表関数 - dynsql 論理グループのスナップショット情報の検索

SNAP\_GET\_DYN\_SQL\_V95 表関数は、dynsql 論理データ・グループからのスナップショット情報を戻します。

注: この表関数は使用すべきではなく、900 ページの『SNAP\_GET\_DYN\_SQL 表関数』に置き換えられました。

戻される可能性のある情報の完全なリストは、1424 ページの表 327 を参照してください。

### 構文

```
▶▶ SNAP_GET_DYN_SQL_V95 (—dbname— [ , member ] ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプション

ンの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり dbname のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されま  
す。アクティブなデータベース・メンバーとは、アプリケーションによるデータ  
ベースへの接続およびデータベースの使用が可能なメンバーのことです。

dbname と member の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシ  
ジャーにより作成されるファイルからのデータの読み取りが試行されます。このファ  
イルはいつでも作成される可能性があるため、データは現行のものであるとは限ら  
ないことに注意してください。対応するスナップショット API 要求タイプを持つフ  
ァイルが存在しない場合、SNAP\_GET\_DYN\_SQL\_V95 表関数は、現在接続中のデー  
タベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DYN\_SQL\_V95 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のい  
ずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特  
権が PUBLIC に付与されます。

## 例

現在接続されているデータベースの現在接続されているデータベース・メン  
バーで実行される動的 SQL のリスト (読み取られる行の番号順に並んでいるもの) を検索  
します。

```
SELECT PREP_TIME_WORST, NUM_COMPILATIONS, SUBSTR(STMT_TEXT, 1, 60)
AS STMT_TEXT FROM TABLE(SNAP_GET_DYN_SQL_V95(' ', -1)) as T
ORDER BY ROWS_READ
```

以下はこの照会の出力例です。

```
PREP_TIME_WORST    ...
-----
0 ...
3 ...
...
4 ...
...
4 ...
...
...
...
```

```

4 ...
...
3 ...
...
4 ...
...

```

この照会からの出力 (続き)。

```

... NUM_COMPILATIONS   STMT_TEXT
... -----
... 0 SET CURRENT LOCALE LC_CTYPE = 'en_US'
... 1 select rows_read, rows_written,
...   substr(stmt_text, 1, 40) as
... 1 select * from table
...   (snap_get_dyn_sqlv9(',-1)) as t
... 1 select * from table
...   (snap_getdetaillog9(',-1)) as t
... 1 select * from table
...   (snap_get_hadr(',-1)) as t
... 1 select prep_time_worst, num_compilations,
...   substr(stmt_text,
... 1 select prep_time_worst, num_compilations,
...   substr(stmt_text,

```

ワークロードの実行後に、ユーザーは次の照会を表関数とともに使用できます。

```

SELECT STATS_FABRICATE_TIME,SYNC_RUNSTATS_TIME
FROM TABLE (SNAP_GET_DYN_SQL_V95('mytestdb', -1))
AS SNAPDB

```

```

STATS_FABRICATE_TIME   SYNC_RUNSTATS_TIME
-----
                2                12
                1                30

```

この表関数に基づくビューの場合:

```

SELECT STATS_FABRICATE_TIME,SYNC_RUNSTATS_TIME
FROM SYSIBMADM.SNAPDYN_SQL

```

```

STATS_FABRICATE_TIME   SYNC_RUNSTATS_TIME
-----
                5                10
                3                20

```

2 record(s) selected.

## 戻される情報

表 327. SNAP\_GET\_DYN\_SQL\_V95 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
NUM_EXECUTIONS	BIGINT	num_executions - ステートメント実行回数
NUM_COMPILATIONS	BIGINT	num_compilations - ステートメント・コンパイル数
PREP_TIME_WORST	BIGINT	prep_time_worst - ステートメント最長準備時間
PREP_TIME_BEST	BIGINT	prep_time_best - ステートメント最短準備時間
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 削除された内部行数
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 挿入された内部行数

表 327. SNAP\_GET\_DYN\_SQL\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 更新された内部行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
STMT_SORTS	BIGINT	stmt_sorts - ステートメント・ソート回数
SORT_OVERFLOWS	BIGINT	sort_overflows - ソート・オーバーフロー
TOTAL_SORT_TIME	BIGINT	total_sort_time - ソート時間合計
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファ・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファ・プール・データの物理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファ・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファ・プール一時データの物理読み取り
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ・プール索引の物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ・プール一時索引の物理読み取り
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ・プール XDAデータの論理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ・プール XDAデータの物理読み取り
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ・プール一時 XDAデータの論理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ・プール一時 XDAデータの物理読み取り : モニター・エレメント
TOTAL_EXEC_TIME	BIGINT	total_exec_time - ステートメント実行の経過時間 (秒単位)*
TOTAL_EXEC_TIME_MS	BIGINT	total_exec_time - ステートメント実行の経過時間 (小数部、マイクロ秒単位)*
TOTAL_USR_CPU_TIME	BIGINT	total_usr_cpu_time - ステートメントのユーザー CPU の合計 (秒単位)*
TOTAL_USR_CPU_TIME_MS	BIGINT	total_usr_cpu_time - ステートメントのユーザー CPU の合計 (小数部、マイクロ秒単位)*
TOTAL_SYS_CPU_TIME	BIGINT	total_sys_cpu_time - ステートメントのシステム CPU の合計 (秒単位)*
TOTAL_SYS_CPU_TIME_MS	BIGINT	total_sys_cpu_time - ステートメントのシステム CPU の合計 (小数部、マイクロ秒単位)*

表 327. SNAP\_GET\_DYN\_SQL\_V95 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
STMT_TEXT	CLOB(2 M)	stmt_text - SQL 動的ステートメント・テキスト
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
STATS_FABRICATE_TIME	BIGINT	動的ステートメントの照会コンパイル中に表または索引のスキャンを実行しないで、システムが必要とされる統計を作成するのに費やす合計時間 (ミリ秒)。
SYNC_RUNSTATS_TIME	BIGINT	動的ステートメントの照会コンパイル中に同期 statistics-collect アクティビティに費やされる合計時間 (ミリ秒)。

\* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する `_S` で終わっている列で報告されている整数秒と、このモニター・エレメントに関する `_MS` で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません:  $(\text{monitor-element-name}_S \times 1,000,000 + \text{monitor-element-name}_{MS}) \div 1,000,000$ 。例えば、 $(\text{ELAPSED\_EXEC\_TIME}_S \times 1,000,000 + \text{ELAPSED\_EXEC\_TIME}_{MS}) \div 1,000,000$ 。

## SNAPHADR 管理ビューおよび SNAP\_GET\_HADR 表関数 - hadr 論理データ・グループのスナップショット情報の検索

SNAPHADR 管理ビューおよび SNAP\_GET\_HADR 表関数は、データベース・スナップショットから、特に `hadr` 論理データ・グループの高可用性災害時リカバリー情報を戻します。

SNAPHADR 管理ビューおよび SNAP\_GET\_HADR 表関数は使用すべきではなく、584 ページの『MON\_GET\_HADR 表関数 - 高可用性災害時リカバリー (HADR) モニター情報を戻す』に置き換えられました。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPHADR 管理ビュー』
- 1427 ページの『SNAP\_GET\_HADR 表関数』

### SNAPHADR 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関する `hadr` 論理データ・グループのスナップショット情報を取得することができます。データベースが 1 次またはスタンバイ高可用性災害時リカバリー (HADR) データベースの場合にのみ、このビューによってデータが戻されます。

SNAPHADR 管理ビューを `SNAPDB`、`SNAPDB_MEMORY_POOL`、`SNAPDETAILLOG`、および `ADMIN_GET_STORAGE_PATHS` 表関数とともに使用すると、`GET SNAPSHOT FOR DATABASE ON database-alias` CLP コマンドに相当する情報が提供されます。

スキーマは `SYSIBMADM` です。



戻される可能性のある情報の完全なリストは、1429 ページの表 328 を参照してください。

## 許可

以下のいずれかの権限が必要です。

- SNAPHADR 管理ビューに対する SELECT 特権
- SNAPHADR 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_HADR 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

1 次 HADR データベース上の HADR に関する構成および状況情報を取得します。

```
SELECT SUBSTR(DB_NAME, 1, 8) AS DBNAME, HADR_ROLE, HADR_STATE,  
       HADR_SYNCMODE, HADR_CONNECT_STATUS  
FROM SYSIBMADM.SNAPHADR
```

以下はこの照会の出力例です。

DBNAME	HADR_ROLE	HADR_STATE	HADR_SYNCMODE	HADR_CONNECT_STATUS
SAMPLE	PRIMARY	PEER	SYNC	CONNECTED

1 record(s) selected.

## SNAP\_GET\_HADR 表関数

SNAP\_GET\_HADR 表関数は SNAPHADR 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_HADR 表関数を SNAP\_GET\_DB、SNAP\_GET\_DB\_MEMORY\_POOL、SNAP\_GET\_DETAILLOG、および ADMIN\_GET\_STORAGE\_PATHS 表関数とともに使用すると、GET SNAPSHOT FOR ALL DATABASES CLP コマンドに相当する情報が提供されます。

戻される可能性のある情報の完全なリストは、1429 ページの表 328 を参照してください。

## 構文

```
▶▶ SNAP_GET_HADR ( ( dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されません。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_HADR 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_HADR 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT

- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

すべてのデータベースの HADR に関する構成および状況情報を取得します。

```
SELECT SUBSTR(DB_NAME, 1, 8) AS DBNAME, HADR_ROLE, HADR_STATE,
       HADR_SYNCMODE, HADR_CONNECT_STATUS
FROM TABLE (SNAP_GET_HADR (CAST (NULL as VARCHAR(128)), 0)) as T
```

以下はこの照会の出力例です。

DBNAME	HADR_ROLE	HADR_STATE	HADR_SYNCMODE	HADR_CONNECT_STATUS
SAMPLE	PRIMARY	PEER	SYNC	CONNECTED
TESTDB	PRIMARY	DISCONNECTED	NEARSYNC	DISCONNECTED

2 record(s) selected.

### 戻される情報

表 328. SNAPHADR 管理ビューおよび SNAP\_GET\_HADR 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
HADR_ROLE	VARCHAR(10)	<p>hadr_role - HADR のロール。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• PRIMARY</li> <li>• STANDARD</li> <li>• STANDBY</li> </ul>
HADR_STATE	VARCHAR(14)	<p>hadr_state - HADR の状態。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• DISCONNECTED</li> <li>• LOCAL_CATCHUP</li> <li>• PEER</li> <li>• REM_CATCH_PEN</li> <li>• REM_CATCHUP</li> </ul>

表 328. SNAPHADR 管理ビューおよび SNAP\_GET\_HADR 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
HADR_SYNCMODE	VARCHAR(10)	<p>hadr_syncmode - HADR 同期モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• ASYNC</li> <li>• NEARSYNC</li> <li>• SUPERASYNC</li> <li>• SYNC</li> </ul>
HADR_CONNECT_STATUS	VARCHAR(12)	<p>hadr_connect_status - HADR 接続状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• CONGESTED</li> <li>• CONNECTED</li> <li>• DISCONNECTED</li> </ul>
HADR_CONNECT_TIME	TIMESTAMP	hadr_connect_time - HADR 接続時刻
HADR_HEARTBEAT	INTEGER	hadr_heartbeat - HADR ハートビート
HADR_LOCAL_HOST	VARCHAR(255)	hadr_local_host - HADR ローカル・ホスト
HADR_LOCAL_SERVICE	VARCHAR(40)	hadr_local_service - HADR ローカル・サービス
HADR_REMOTE_HOST	VARCHAR(255)	hadr_remote_host - HADR リモート・ホスト
HADR_REMOTE_SERVICE	VARCHAR(40)	hadr_remote_service - HADR リモート・サービス
HADR_REMOTE_INSTANCE	VARCHAR(128)	hadr_remote_instance - HADR リモート・インスタンス
HADR_TIMEOUT	BIGINT	hadr_timeout - HADR タイムアウト
HADR_PRIMARY_LOG_FILE	VARCHAR(255)	hadr_primary_log_file - HADR 1 次ログ・ファイル
HADR_PRIMARY_LOG_PAGE	BIGINT	hadr_primary_log_page - HADR 1 次ログ・ページ
HADR_PRIMARY_LOG_LSN	BIGINT	hadr_primary_log_lsn - HADR 1 次ログ LSN
HADR_STANDBY_LOG_FILE	VARCHAR(255)	hadr_standby_log_file - HADR スタンバイ・ログ・ファイル
HADR_STANDBY_LOG_PAGE	BIGINT	hadr_standby_log_page - HADR スタンバイ・ログ・ページ

表 328. SNAPHADR 管理ビューおよび SNAP\_GET\_HADR 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
HADR_STANDBY_LOG_LSN	BIGINT	hadr_standby_log_lsn - HADR スタンバイ・ログ LSN
HADR_LOG_GAP	BIGINT	hadr_log_gap - HADR ログ・ギャップ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数 - lock 論理データ・グループのスナップショット情報の検索

SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数は、特に lock 論理データ・グループのロック・スナップショット情報を戻します。

注: この管理ビューおよび表関数は非推奨になり、以下のものに置き換えられました。すなわち、526 ページの『MON\_GET\_APPL\_LOCKWAIT - アプリケーションが待機しているロックについての情報の取得』、598 ページの『MON\_GET\_LOCKS - 現在接続されているデータベース内のすべてのロックのリスト』、および 482 ページの『MON\_FORMAT\_LOCK\_NAME - 内部ロック名のフォーマット設定と詳細の出力』。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPLOCK 管理ビュー』
- 1432 ページの『SNAP\_GET\_LOCK 表関数』

### SNAPLOCK 管理ビュー

この管理ビューでは、現在接続されているデータベースの lock 論理データ・グループ・スナップショット情報を検索できます。

SNAPLOCKWAIT 管理ビューと共に使用すると、SNAPLOCK 管理ビューは、**GET SNAPSHOT FOR LOCKS ON database-alias** CLP コマンドと同等の情報を提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1434 ページの表 329 を参照してください。

## 許可

以下のいずれかの権限が必要です。

- SNAPLOCK 管理ビューに対する SELECT 特権
- SNAPLOCK 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_LOCK 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースのデータベース・メンバー 0 のロック情報を検索します。

```
SELECT AGENT_ID, LOCK_OBJECT_TYPE, LOCK_MODE, LOCK_STATUS
FROM SYSIBMADM.SNAPLOCK WHERE DBPARTITIONNUM = 0
```

以下はこの照会の出力例です。

AGENT_ID	LOCK_OBJECT_TYPE	LOCK_MODE	LOCK_STATUS
7	TABLE	IX	GRNT

1 record(s) selected.

## SNAP\_GET\_LOCK 表関数

SNAP\_GET\_LOCK 表関数は SNAPLOCK 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_LOCKWAIT 表関数と共に使用すると、SNAP\_GET\_LOCK 表関数は、**GET SNAPSHOT FOR LOCKS ON database-alias** CLP コマンドと同等の情報を提供します。

戻される可能性のある情報の完全なリストは、1434 ページの表 329 を参照してください。

## 構文

```
▶▶—SNAP_GET_LOCK—(—dbname— [ , member ] )▶▶▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値または空ストリングを指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されません。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_LOCK 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_LOCK 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベースの現行データベース・メンバーのロック情報を検索します。

```
SELECT AGENT_ID, LOCK_OBJECT_TYPE, LOCK_MODE, LOCK_STATUS
FROM TABLE(SNAP_GET_LOCK('1',-1)) as T
```

以下はこの照会の出力例です。

```
AGENT_ID      LOCK_OBJECT_TYPE  LOCK_MODE  LOCK_STATUS
-----
          680 INTERNALV_LOCK      S          GRNT
          680 INTERNALP_LOCK      S          GRNT
```

2 record(s) selected.

### 戻される情報

表 329. SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
TAB_FILE_ID	BIGINT	table_file_id - 表ファイル ID



表 329. SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_OBJECT_TYPE	VARCHAR(18)	<p>lock_object_type - 待機中のロック対象タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• AUTORESIZE_LOCK</li> <li>• AUTOSTORAGE_LOCK</li> <li>• BLOCK_LOCK</li> <li>• EOT_LOCK</li> <li>• INPLACE_REORG_LOCK</li> <li>• INTERNAL_LOCK</li> <li>• INTERNALB_LOCK</li> <li>• INTERNALC_LOCK</li> <li>• INTERNALJ_LOCK</li> <li>• INTERNALL_LOCK</li> <li>• INTERNALO_LOCK</li> <li>• INTERNALQ_LOCK</li> <li>• INTERNALP_LOCK</li> <li>• INTERNALS_LOCK</li> <li>• INTERNALT_LOCK</li> <li>• INTERNALV_LOCK</li> <li>• KEYVALUE_LOCK</li> <li>• ROW_LOCK</li> <li>• SYSBOOT_LOCK</li> <li>• TABLE_LOCK</li> <li>• TABLE_PART_LOCK</li> <li>• TABLESPACE_LOCK</li> <li>• XML_PATH_LOCK</li> </ul>

表 329. SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_MODE	VARCHAR(10)	lock_mode - ロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
LOCK_STATUS	VARCHAR(10)	lock_status - ロック状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• CONV</li> <li>• GRNT</li> </ul>
LOCK_ESCALATION	SMALLINT	lock_escalation - ロック・エスカレーション
TABNAME	VARCHAR(128)	table_name - 表名
TABSHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名

表 329. SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_ATTRIBUTES	VARCHAR(128)	lock_attributes - ロック属性。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。ロックがない場合、テキスト ID は NONE となり、それ以外の場合、以下のいずれかの組み合わせを '+' 記号で区切ったものとなります。 <ul style="list-style-type: none"> <li>• ALLOW_NEW</li> <li>• DELETE_IN_BLOCK</li> <li>• ESCALATED</li> <li>• INSERT</li> <li>• NEW_REQUEST</li> <li>• RR</li> <li>• RR_IN_BLOCK</li> <li>• UPDATE_DELETE</li> <li>• WAIT_FOR_AVAIL</li> </ul>
LOCK_COUNT	BIGINT	lock_count - ロック・カウント
LOCK_CURRENT_MODE	VARCHAR(10)	lock_current_mode - 移行前の元のロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
LOCK_HOLD_COUNT	BIGINT	lock_hold_count - ロック保留カウント
LOCK_NAME	VARCHAR(32)	lock_name - ロック名
LOCK_RELEASE_FLAGS	BIGINT	lock_release_flags - ロック保留解除フラグ

表 329. SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID。非パーティション表では、このエレメントは NULL です。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数 - lockwait 論理データ・グループのスナップショット情報の検索

SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数は、特に lockwait 論理データ・グループのロック待機スナップショット情報を戻します。

注: この管理ビューおよび表関数は非推奨になり、以下のものに置き換えられました。すなわち、758 ページの『MON\_LOCKWAITS 管理ビュー - ロックの取得を待機しているアプリケーションに関するメトリックの取得』、および 526 ページの『MON\_GET\_APPL\_LOCKWAIT - アプリケーションが待機しているロックについての情報の取得』、598 ページの『MON\_GET\_LOCKS - 現在接続されているデータベース内のすべてのロックのリスト』、482 ページの『MON\_FORMAT\_LOCK\_NAME - 内部ロック名のフォーマット設定と詳細の出力』。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPLOCKWAIT 管理ビュー』
- 1439 ページの『SNAP\_GET\_LOCKWAIT 表関数』

### SNAPLOCKWAIT 管理ビュー

この管理ビューでは、現在接続されているデータベースの lockwait 論理データ・グループ・スナップショット情報を検索できます。

SNAPLOCK 管理ビューと共に使用すると、SNAPLOCKWAIT 管理ビューは、**GET SNAPSHOT FOR LOCKS ON database-alias** CLP コマンドと同等の情報を提供しません。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1441 ページの表 330 を参照してください。

## 許可

以下のいずれかの権限が必要です。

- SNAPLOCKWAIT 管理ビューに対する SELECT 特権
- SNAPLOCKWAIT 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_LOCKWAIT 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースのデータベース・メンバー 0 のロック待機情報を検索します。

```
SELECT AGENT_ID, LOCK_MODE, LOCK_OBJECT_TYPE, AGENT_ID_HOLDING_LK,  
       LOCK_MODE_REQUESTED FROM SYSIBMADM.SNAPLOCKWAIT  
WHERE DBPARTITIONNUM = 0
```

以下はこの照会の出力例です。

```
AGENT_ID      LOCK_MODE LOCK_OBJECT_TYPE ...  
-----  
          7 IX          TABLE          ...
```

1 record(s) selected.

この照会からの出力 (続き)。

```
... AGENT_ID_HOLDING_LK  LOCK_MODE_REQUESTED  
... -----  
...                   12 IS
```

## SNAP\_GET\_LOCKWAIT 表関数

SNAP\_GET\_LOCKWAIT 表関数は SNAPLOCKWAIT 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_LOCK 表関数と共に使用すると、SNAP\_GET\_LOCKWAIT 表関数は、**GET SNAPSHOT FOR LOCKS ON database-alias** CLP コマンドと同等の情報を提供します。

戻される可能性のある情報の完全なリストは、1441 ページの表 330 を参照してください。

## 構文

```
▶▶ SNAP_GET_LOCKWAIT (—dbname— [ , member ] )▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値または空ストリングを指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されず、アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_LOCKWAIT 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_LOCKWAIT 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON

- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベースの現行データベース・メンバーのロック待機情報を検索します。

```
SELECT AGENT_ID, LOCK_MODE, LOCK_OBJECT_TYPE, AGENT_ID_HOLDING_LK,
       LOCK_MODE_REQUESTED FROM TABLE(SNAP_GET_LOCKWAIT(' ', -1)) AS T
```

以下はこの照会の出力例です。

```
AGENT_ID      LOCK_MODE  LOCK_OBJECT_TYPE  ...
-----
          12 X          ROW_LOCK          ...
```

1 record(s) selected.

この照会からの出力 (続き)。

```
... AGENT_ID_HOLDING_LK  LOCK_MODE_REQUESTED
... -----
...                   7 X
```

### 使用上の注意

ロック待機情報を表示するには、まずデータベース・マネージャー構成でデフォルトの LOCK モニター・スイッチをオンにする必要があります。変更を即時に有効にするには、CLP を使用してインスタンスに明示的にアタッチし、次いで以下の CLP コマンドを発行します。

```
UPDATE DATABASE MANAGER CONFIGURATION CLP USING DFT_MON_LOCK ON
```

デフォルトの設定値も、ADMIN\_CMD ストアード・プロシージャによりオンにすることができます。例:

```
CALL SYSPROC.ADMIN_CMD('update dbm cfg using DFT_MON_LOCK ON')
```

ADMIN\_CMD ストアード・プロシージャを使用する場合、または事前にインスタンスにアタッチせずに CLP コマンドを使用する場合、インスタンスをリサイクルしなければ変更は有効になりません。

### 戻される情報

表 330. SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。

表 330. SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
SUBSECTION_NUMBER	BIGINT	ss_number - サブセクション番号
LOCK_MODE	VARCHAR(10)	lock_mode - ロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>



表 330. SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_OBJECT_TYPE	VARCHAR(18)	lock_object_type - 待機中のロック対象タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• AUTORESIZE_LOCK</li> <li>• AUTOSTORAGE_LOCK</li> <li>• BLOCK_LOCK</li> <li>• EOT_LOCK</li> <li>• INPLACE_REORG_LOCK</li> <li>• INTERNAL_LOCK</li> <li>• INTERNALB_LOCK</li> <li>• INTERNALC_LOCK</li> <li>• INTERNALJ_LOCK</li> <li>• INTERNALL_LOCK</li> <li>• INTERNALO_LOCK</li> <li>• INTERNALQ_LOCK</li> <li>• INTERNALP_LOCK</li> <li>• INTERNALS_LOCK</li> <li>• INTERNALT_LOCK</li> <li>• INTERNALV_LOCK</li> <li>• KEYVALUE_LOCK</li> <li>• ROW_LOCK</li> <li>• SYSBOOT_LOCK</li> <li>• TABLE_LOCK</li> <li>• TABLE_PART_LOCK</li> <li>• TABLESPACE_LOCK</li> <li>• XML_PATH_LOCK</li> </ul>
AGENT_ID_HOLDING_LK	BIGINT	agent_id_holding_lock - ロックを保持しているエージェント ID
LOCK_WAIT_START_TIME	TIMESTAMP	lock_wait_start_time - ロック待機開始タイム・スタンプ

表 330. SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_MODE_REQUESTED	VARCHAR(10)	lock_mode_requested - 要求されているロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
LOCK_ESCALATION	SMALLINT	lock_escalation - ロック・エスカレーション
TABNAME	VARCHAR(128)	table_name - 表名
TABSHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
APPL_ID_HOLDING_LK	VARCHAR(128)	appl_id_holding_lk - ロックを保持しているアプリケーション ID
LOCK_ATTRIBUTES	VARCHAR(128)	lock_attributes - ロック属性。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。ロックがない場合、テキスト ID は NONE となり、それ以外の場合、以下のいずれかの組み合わせを '+' 記号で区切ったものとなります。 <ul style="list-style-type: none"> <li>• ALLOW_NEW</li> <li>• DELETE_IN_BLOCK</li> <li>• ESCALATED</li> <li>• INSERT</li> <li>• NEW_REQUEST</li> <li>• RR</li> <li>• RR_IN_BLOCK</li> <li>• UPDATE_DELETE</li> <li>• WAIT_FOR_AVAIL</li> </ul>

表 330. SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_CURRENT_MODE	VARCHAR(10)	lock_current_mode - 移行前の元のロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
LOCK_NAME	VARCHAR(32)	lock_name - ロック名
LOCK_RELEASE_FLAGS	BIGINT	lock_release_flags - ロック保留解除フラグ。
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID。非パーティション表では、このエレメントは NULL です。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAP\_GET\_STO\_PATHS

SNAP\_GET\_STO\_PATHS 表関数は、storage\_paths 論理データ・グループからのスナップショット情報を戻します。

注: この表関数は使用すべきではなく、254 ページの

『ADMIN\_GET\_STORAGE\_PATHS 表関数 - 自動ストレージ・パス情報の取得』に置き換えられました。

▶▶—SNAP\_GET\_STO\_PATHS—(—dbname—, —member—)—▶▶

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現行のデータベース・パーティションの場合は -1、またはすべてのアクティブなデータベース・パーティションの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアード・プロシージャーによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 331. SNAP\_GET\_STO\_PATHS 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
DB_NAME	VARCHAR(128)	db_name - データベース名
DB_STORAGE_PATH	VARCHAR(256)	db_storage_path - 自動ストレージ・パス

## SNAPSTORAGE\_PATHS 管理ビューおよび SNAP\_GET\_STORAGE\_PATHS\_V97 表関数 - 自動ストレージ・パスの情報の検索

SNAPSTORAGE\_PATHS 管理ビューおよび SNAP\_GET\_STORAGE\_PATHS\_V97 は、特に db\_storage\_group 論理データ・グループから、データベースの自動ストレージ・パスのリストを、ストレージ・パスごとのファイル・システム情報を含めて戻します。

注: 『SNAPSTORAGE\_PATHS 管理ビュー』 および 1448 ページの 『SNAP\_GET\_STORAGE\_PATHS\_V97』 は使用すべきではなく、254 ページの 『ADMIN\_GET\_STORAGE\_PATHS 表関数 - 自動ストレージ・パス情報の取得』 に置き換えられました。この関数とビューは、将来のリリースで除去される可能性があります。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPSTORAGE\_PATHS 管理ビュー』
- 1448 ページの 『SNAP\_GET\_STORAGE\_PATHS\_V97』

### SNAPSTORAGE\_PATHS 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関する自動ストレージ・パスの情報を検索できます。

SNAPSTORAGE\_PATHS 管理ビューを SNAPDB、SNAPDETAILLOG、SNAPHADR、および SNAPDB\_MEMORY\_POOL 管理ビューと併せて使用することにより、**GET SNAPSHOT FOR DATABASE ON database-alias** CLP コマンドと同等の情報を戻します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1449 ページの表 332 を参照してください。

### 許可

- SYSMON 権限
- SNAPSTORAGE\_PATHS 管理ビューに対する SELECT または CONTROL 特権、および ADMIN\_GET\_STORAGE\_PATHS 表関数に対する EXECUTE 特権

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続されている単一メンバー・データベースのストレージ・パスを検索します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, SUBSTR(DB_STORAGE_PATH,1,8)
      AS DB_STORAGE_PATH, SUBSTR(HOSTNAME,1,10) AS HOSTNAME
FROM SYSIBMADM.SNAPSTORAGE_PATHS
```

以下はこの照会の出力例です。

```
DB_NAME  DB_STORAGE_PATH  HOSTNAME
-----  -
STOPATH  d:                      JESSICAE
```

1 record(s) selected.

## SNAP\_GET\_STORAGE\_PATHS\_V97

SNAP\_GET\_STORAGE\_PATHS\_V97 表関数は、SNAPSTORAGE\_PATHS 管理ビューと同様の情報を戻します。この表関数を使用すると、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_STORAGE\_PATHS\_V97 表関数を SNAP\_GET\_DB、SNAP\_GET\_DETAILLOG、SNAP\_GET\_HADR、および SNAP\_GET\_DB\_MEMORY\_POOL 表関数と併せて使用することにより、GET SNAPSHOT FOR ALL DATABASES CLP コマンドと同等の情報を戻します。

戻される可能性のある情報の完全なリストは、1449 ページの表 332 を参照してください。

## 構文

```
▶▶—SNAP_GET_STORAGE_PATHS_V97—(—dbname— [ , member ] )—▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、LIST DATABASE DIRECTORY コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### member

有効なメンバー番号を指定する、タイプ INTEGER のオプション入力引数。現在のメンバーの場合は -1、すべてのアクティブなメンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり dbname のみを指定する場合、データはデータベースがアクティブになっているすべてのメンバーから戻されます。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_STORAGE\_PATHS\_V97 表関数は、現在接続中のデータベースおよびデータベース・メンバーのスナップショットを取得します。

## 許可

- SYSMON 権限
- SNAP\_GET\_STORAGE\_PATHS\_V97 表関数に対する EXECUTE 特権

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべてのアクティブ・データベースに関するストレージ・パスの情報を取り出します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, DB_STORAGE_PATH
FROM TABLE(SNAP_GET_STORAGE_PATHS_V97(CAST (NULL AS VARCHAR(128)), -1)) AS T
```

以下はこの照会の出力例です。

```
DB_NAME  DB_STORAGE_PATH
-----  -
STOPATH  /home/jessicae/sdb
MYDB     /home/jessicae/mdb

      2 record(s) selected
```

## 戻される情報

ファイル・システムの情報を戻すためには、BUFFERPOOL モニター・スイッチをオンにする必要があります。

表 332. SNAPSTORAGE\_PATHS 管理ビューおよび SNAP\_GET\_STORAGE\_PATHS\_V97 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
DB_STORAGE_PATH	VARCHAR(256)	db_storage_path - 自動ストレージ・パス
DB_STORAGE_PATH_WITH_DPE	VARCHAR(256)	db_storage_path_with_dpe - データベース・パーティション式を含むストレージ・パス・モニター・エレメント

表 332. SNAPSTORAGE\_PATHS 管理ビューおよび SNAP\_GET\_STORAGE\_PATHS\_V97 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
DB_STORAGE_PATH_STATE	VARCHAR(16)	db_storage_path_state - ストレージ・パスの状態モニター・エレメント
FS_ID	VARCHAR(22)	fs_id - 固有のファイル・システム識別番号
FS_TOTAL_SIZE	BIGINT	fs_total_size - ファイル・システムの合計サイズ
FS_USED_SIZE	BIGINT	fs_used_size - ファイル・システム上で使用されるスペースの量
STO_PATH_FREE_SIZE	BIGINT	sto_path_free_size - 自動ストレージ・パスのフリー・スペース

## SNAPTAB 管理ビューおよび SNAP\_GET\_TAB 表関数 - table 論理データ・グループのスナップショット情報の検索

SNAPTAB 管理ビューおよび SNAP\_GET\_TAB 表関数は、table 論理データ・グループからのスナップショット情報を戻します。

**重要:** SNAPTAB 管理ビューおよび SNAP\_GET\_TAB 表関数は使用すべきではありません。これらの非推奨になったインターフェースによって返される情報を取得するには、537 ページの『MON\_GET\_BUFFERPOOL 表関数 - バッファ・プール・メトリックの取得』、693 ページの『MON\_GET\_TABLE 表関数 - 表メトリックの取得』、697 ページの『MON\_GET\_TABLESPACE 表関数 - 表スペース・メトリックの取得』、および 461 ページの『MON\_BP\_UTILIZATION 管理ビュー - バッファ・プールに関するメトリックの取得』を使用してください。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 『SNAPTAB 管理ビュー』
- 1451 ページの『SNAP\_GET\_TAB 表関数』

### SNAPTAB 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関する table 論理データ・グループのスナップショット情報を検索できます。

SNAPTAB\_REORG 管理ビューと併せて使用することにより、SNAPTAB 管理ビューは **GET SNAPSHOT FOR TABLES ON database-alias CLP** コマンドと同等の情報を戻します。

スキーマは SYSIBMADM です。



戻される可能性のある情報の完全なリストは、1453 ページの表 333 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPTAB 管理ビューに対する SELECT 特権
- SNAPTAB 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_TAB 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

すべてのアクティブな表のスキーマと名前を取り出します。

```
SELECT SUBSTR(TABSCHEMA,1,8), SUBSTR(TABNAME,1,15) AS TABNAME, TAB_TYPE,  
       DBPARTITIONNUM FROM SYSIBMADM.SNAPTAB
```

以下はこの照会の出力例です。

TABSCHEMA	TABNAME	TAB_TYPE	DBPARTITIONNUM
SYSTOOLS	HMON_ATM_INFO	USER_TABLE	0

1 record selected.

## SNAP\_GET\_TAB 表関数

SNAP\_GET\_TAB 表関数は SNAPTAB 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_TAB\_REORG 表関数と併せて使用することにより、SNAP\_GET\_TAB 表関数は **GET SNAPSHOT FOR TABLES ON database-alias** CLP コマンドと同等の情報を戻します。

戻される可能性のある情報の完全なリストは、1453 ページの表 333 を参照してください。

## 構文

```
▶▶ SNAP_GET_TAB ( ( dbname [ , member ] ) )
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のメンバーの場合は -1、すべてのアクティブなメンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはデータベースがアクティブになっているすべてのメンバーから戻されます。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_TAB 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_TAB 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT

- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベースの集約ビューとして、アクティブな表のリストを検索します。

```
SELECT SUBSTR(TABSCHEMA,1,8) AS TABSCHEMA, SUBSTR(TABNAME,1,15) AS TABNAME,
       TAB_TYPE, DBPARTITIONNUM FROM TABLE(SNAP_GET_TAB('','-2')) AS T
```

以下はこの照会の出力例です。

TABSCHEMA	TABNAME	TAB_TYPE	DBPARTITIONNUM
SYSTOOLS	HMON_ATM_INFO	USER_TABLE	-
JESSICAE	EMPLOYEE	USER_TABLE	-

## 戻される情報

表 333. SNAPTAB 管理ビューおよび SNAP\_GET\_TAB 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TABNAME	VARCHAR(128)	table_name - 表名
TAB_FILE_ID	BIGINT	table_file_id - 表ファイル ID
TAB_TYPE	VARCHAR(14)	table_type - 表タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• USER_TABLE</li> <li>• DROPPED_TABLE</li> <li>• TEMP_TABLE</li> <li>• CATALOG_TABLE</li> <li>• REORG_TABLE</li> </ul>
DATA_OBJECT_PAGES	BIGINT	data_object_pages - データ・オブジェクト・ページ数
INDEX_OBJECT_PAGES	BIGINT	index_object_pages - 索引オブジェクト・ページ数
LOB_OBJECT_PAGES	BIGINT	lob_object_pages - LOB オブジェクト・ページ数
LONG_OBJECT_PAGES	BIGINT	long_object_pages - 長オブジェクト・ページ数
XDA_OBJECT_PAGES	BIGINT	xda_object_pages - XDA オブジェクト・ページ数
ROWS_READ	BIGINT	rows_read - 読み取り行数

表 333. SNAPTAB 管理ビューおよび SNAP\_GET\_TAB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
OVERFLOW_ACCESSES	BIGINT	overflow_accesses - オーバーフロー・レコードへのアクセス
PAGE_REORGS	BIGINT	page_reorgs - ページ再編成
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID。非パーティション表では、このエレメントは NULL になります。
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAP\_GET\_TAB\_V91

SNAP\_GET\_TAB\_V91 表関数は、table 論理データ・グループからのスナップショット情報を戻します。

注: この表関数は使用すべきではなく、SNAPTAB 管理ビューおよび SNAP\_GET\_TAB 表関数に置き換えられました。

戻される可能性のある情報の完全なリストは、1456 ページの表 334 を参照してください。

### 構文

```

▶▶ SNAP_GET_TAB_V91( (dbname [ , dbpartitionnum ] ) )

```

スキーマは SYSPROC です。

### 表関数パラメーター

#### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### dbpartitionnum

有効なデータベース・パーティション番号を指定する、タイプ INTEGER のオプション入力引数。現行のデータベース・パーティションには -1、すべてのアクティブなデータベース・パーティションの集約には -2 を指定します。

*dbname* が NULL に設定されておらず、*dbpartitionnum* が NULL に設定されている場合、*dbpartitionnum* には暗黙的に -1 が設定されます。この入力オプションが使用されない場合、つまり、*dbname* のみが指定されている場合、データはすべてのアクティブなデータベース・パーティションから戻されます。アクティブなデータベース・パーティションとは、アプリケーションからデータベースに接続して、これを使用することが可能なパーティションのことです。

*dbname* および *dbpartitionnum* の両方が NULL に設定された場合、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りを試行します。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合には、現在接続されているデータベースのスナップショットとデータベース・パーティション番号を SNAP\_GET\_TAB\_V91 表関数が取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_TAB\_V91 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースの集約ビューとして、アクティブな表のリストを検索します。

```
SELECT SUBSTR(TABSCHEMA,1,8) AS TABSCHEMA, SUBSTR(TABNAME,1,15) AS TABNAME,  
       TAB_TYPE, DBPARTITIONNUM FROM TABLE(SNAP_GET_TAB('',-2)) AS T
```

以下はこの照会の出力例です。

TABSCHEMA	TABNAME	TAB_TYPE	DBPARTITIONNUM
SYSTOOLS	HMON_ATM_INFO	USER_TABLE	-
JESSICAE	EMPLOYEE	USER_TABLE	-

## 戻される情報

表 334. SNAP\_GET\_TAB\_V9I 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TABSHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TABNAME	VARCHAR(128)	table_name - 表名
TAB_FILE_ID	BIGINT	table_file_id - 表ファイル ID
TAB_TYPE	VARCHAR(14)	table_type - 表タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• USER_TABLE</li> <li>• DROPPED_TABLE</li> <li>• TEMP_TABLE</li> <li>• CATALOG_TABLE</li> <li>• REORG_TABLE</li> </ul>
DATA_OBJECT_PAGES	BIGINT	data_object_pages - データ・オブジェクト・ページ数
INDEX_OBJECT_PAGES	BIGINT	index_object_pages - 索引オブジェクト・ページ数
LOB_OBJECT_PAGES	BIGINT	lob_object_pages - LOB オブジェクト・ページ数
LONG_OBJECT_PAGES	BIGINT	long_object_pages - 長オブジェクト・ページ数
XDA_OBJECT_PAGES	BIGINT	xda_object_pages - XDA オブジェクト・ページ数
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
OVERFLOW_ACCESSES	BIGINT	overflow_accesses - オーバーフロー・レコードへのアクセス
PAGE_REORGS	BIGINT	page_reorgs - ページ再編成
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID。非パーティション表では、このエレメントは NULL になります。

## SNAP\_GET\_TBSP\_PART\_V97 表関数 - tablespace\_nodeinfo 論理データ・グループのスナップショット情報の検索

SNAP\_GET\_TBSP\_PART\_V97 表関数は、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象として、tablespace\_nodeinfo 論理データ・グループのスナップショット情報を戻します。

注: この表関数は使用すべきではなく、942 ページの『SNAP\_GET\_TBSP\_PART 表関数』に置き換えられました。

戻される可能性のある情報の完全なリストは、1458 ページの表 335 を参照してください。

### 構文

```
→ SNAP_GET_TBSP_PART_V97 ( (dbname [ , member ] ) ) →
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されません。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_TBSP\_PART\_V97 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

### 許可

- SYSMON 権限
- SNAP\_GET\_TBSP\_PART\_V97 表関数に対する EXECUTE 特権

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

接続されているデータベースの接続されているデータベース・メンバーの表スペースとその状態のリストを検索します。

```
SELECT SUBSTR(TBSP_NAME,1,30) AS TBSP_NAME, TBSP_ID,  
SUBSTR(TBSP_STATE,1,30) AS TBSP_STATE  
FROM TABLE(SNAP_GET_TBSP_PART_V97(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

以下は、この照会からの出力例です。

TBSP_NAME	TBSP_ID	TBSP_STATE
SYSCATSPACE		0 NORMAL
TEMPSPACE1		1 NORMAL
USERSPACE1		2 NORMAL
SYSTOOLSPACE		3 NORMAL
SYSTOOLSTMPSPACE		4 NORMAL

5 record(s) selected.

### 戻される情報

表 335. SNAP\_GET\_TBSP\_PART\_V97 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_NAME	VARCHAR (128)	tablespace_name - 表スペース名
TBSP_ID	BIGINT	tablespace_id - 表スペース ID



表 335. SNAP\_GET\_TBSP\_PART\_V97 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_STATE	VARCHAR (256)	<p>tablespace_state - 表スペースの状態。このインターフェースは、sqlutil.h での定義に基づくテキスト ID を戻します。これは次のものを「+」符号で分離して組み合わせたものになります。</p> <ul style="list-style-type: none"> <li>• BACKUP_IN_PROGRESS</li> <li>• BACKUP_PENDING</li> <li>• DELETE_PENDING</li> <li>• DISABLE_PENDING</li> <li>• DROP_PENDING</li> <li>• LOAD_IN_PROGRESS</li> <li>• LOAD_PENDING</li> <li>• NORMAL</li> <li>• OFFLINE</li> <li>• PSTAT_CREATION</li> <li>• PSTAT_DELETION</li> <li>• QUIESCED_EXCLUSIVE</li> <li>• QUIESCED_SHARE</li> <li>• QUIESCED_UPDATE</li> <li>• REBAL_IN_PROGRESS</li> <li>• REORG_IN_PROGRESS</li> <li>• RESTORE_IN_PROGRESS</li> <li>• RESTORE_PENDING</li> <li>• ROLLFORWARD_IN_PROGRESS</li> <li>• ROLLFORWARD_PENDING</li> <li>• STORDEF_ALLOWED</li> <li>• STORDEF_CHANGED</li> <li>• STORDEF_FINAL_VERSION</li> <li>• STORDEF_PENDING</li> <li>• SUSPEND_WRITE</li> </ul>
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 表スペースのプリフェッチ・サイズ
TBSP_NUM_QUIESCERS	BIGINT	tablespace_num_quiescers - 静止プログラム数
TBSP_STATE_CHANGE_OBJECT_ID	BIGINT	tablespace_state_change_object_id - 状態変更オブジェクト ID
TBSP_STATE_CHANGE_TBSP_ID	BIGINT	tablespace_state_change_ts_id - 状態変更表スペース ID
TBSP_MIN_RECOVERY_TIME	TIMESTAMP	tablespace_min_recovery_time - ロールフォワードの最小リカバリー時間
TBSP_TOTAL_PAGES	BIGINT	tablespace_total_pages - 表スペース内の合計ページ数

表 335. SNAP\_GET\_TBSP\_PART\_V97 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_USABLE_PAGES	BIGINT	tablespace_usable_pages - 表スペース内の使用可能ページ数
TBSP_USED_PAGES	BIGINT	tablespace_used_pages - 表スペース内の使用されているページ数
TBSP_FREE_PAGES	BIGINT	tablespace_free_pages - 表スペース内のフリー・ページ数
TBSP_PENDING_FREE_PAGES	BIGINT	tablespace_pending_free_pages - 表スペース内のペンディング・フリー・ページ数
TBSP_PAGE_TOP	BIGINT	tablespace_page_top - 表スペース最高水準点
REBALANCER_MODE	VARCHAR (30)	tablespace_rebalancer_mode - リバランサー・モード。このインターフェースは、sqlmon.hでの定義に基づくテキスト ID を戻します。これは次のいずれかです。  <ul style="list-style-type: none"> <li>• FWD_REBAL</li> <li>• NO_REBAL</li> <li>• REV_REBAL</li> <li>• FWD_REBAL_OF_2PASS</li> <li>• REV_REBAL_OF_2PASS</li> </ul>
REBALANCER_EXTENTS_REMAINING	BIGINT	tablespace_rebalancer_extents_remaining - リバランサーで処理されるエクステントの合計数
REBALANCER_EXTENTS_PROCESSED	BIGINT	tablespace_rebalancer_extents_processed - リバランサーで処理されたエクステントの数
REBALANCER_PRIORITY	BIGINT	tablespace_rebalancer_priority - 現行のリバランサー優先順位
REBALANCER_START_TIME	TIMESTAMP	tablespace_rebalancer_start_time - リバランサー開始時刻
REBALANCER_RESTART_TIME	TIMESTAMP	tablespace_rebalancer_restart_time - リバランサー再始動時刻
REBALANCER_LAST_EXTENT_MOVED	BIGINT	tablespace_rebalancer_last_extent_moved - リバランサーによって最後に移動されたエクステント
TBSP_NUM_RANGES	BIGINT	tablespace_num_ranges - 表スペース・マップ内の範囲数
TBSP_NUM_CONTAINERS	BIGINT	tablespace_num_containers - 表スペース内のコンテナ数
TBSP_INITIAL_SIZE	BIGINT	tablespace_initial_size - 表スペースの初期サイズ
TBSP_CURRENT_SIZE	BIGINT	tablespace_current_size - 表スペースの現行サイズ
TBSP_MAX_SIZE	BIGINT	tablespace_max_size - 表スペースの最大サイズ
TBSP_INCREASE_SIZE	BIGINT	tablespace_increase_size - バイト単位のサイズの増加

表 335. SNAP\_GET\_TBSP\_PART\_V97 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TBSP_INCREASE_SIZE_PERCENT	SMALLINT	tablespace_increase_size_percent - パーセント単位のサイズの増加
TBSP_LAST_RESIZE_TIME	TIMESTAMP	tablespace_last_resize_time - 最後にサイズ変更が正常に行われた時刻
TBSP_LAST_RESIZE_FAILED	SMALLINT	tablespace_last_resize_failed - 失敗した最後のサイズ変更
TBSP_PATHS_DROPPED	SMALLINT	表スペースが、ドロップされた 1 つ以上のストレージ・パスにあることを示します (0 - いいえ、1 - はい)
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

## SNAP\_GET\_TBSP\_V91

SNAP\_GET\_TBSP\_V91 表関数は、table space 論理データ・グループからのスナップショット情報を戻します。

注: この表関数は使用すべきではなく、936 ページの『SNAP\_GET\_TBSP 表関数』に置き換えられました。

戻される可能性のある情報の完全なリストは、1463 ページの表 336 を参照してください。

### 構文

```

▶▶ SNAP_GET_TBSP_V91 ( ( dbname [ , dbpartitionnum ] ) )

```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### *dbpartitionnum*

有効なデータベース・パーティション番号を指定する、タイプ INTEGER のオプション入力引数。現行のデータベース・パーティションには -1、すべてのアクティブなデータベース・パーティションの集約には -2 を指定します。

*dbname* が NULL に設定されておらず、*dbpartitionnum* が NULL に設定されている場合、*dbpartitionnum* には暗黙的に -1 が設定されます。この入力オプションが使用されない場合、つまり、*dbname* のみが指定されている場合、データはす

すべてのアクティブなデータベース・パーティションから戻されます。アクティブなデータベース・パーティションとは、アプリケーションからデータベースに接続して、これを使用することが可能なパーティションのことです。

*dbname* および *dbpartitionnum* の両方が NULL に設定された場合、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りを試行します。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合には、現在接続されているデータベースのスナップショットとデータベース・パーティション番号を SNAP\_GET\_TBSP\_V91 表関数が取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_TBSP\_V91 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースの、すべてのデータベース・パーティションの表スペースのリストを検索します。

```
SELECT SUBSTR(TBSP_NAME,1,10) AS TBSP_NAME, TBSP_ID, TBSP_TYPE,  
       TBSP_CONTENT_TYPE, DBPARTITIONNUM FROM TABLE(SNAP_GET_TBSP_V91('')) AS T
```

以下はこの照会の出力例です。

TBSP_NAME	TBSP_ID	TBSP_TYPE	TBSP_CONTENT_TYPE	DBPARTITIONNUM
TEMPSPACE1	1	SMS	SYSTEMP	1
USERSPACE1	2	DMS	LONG	1
SYSCATSPAC	0	DMS	ANY	0
TEMPSPACE1	1	SMS	SYSTEMP	0
USERSPACE1	2	DMS	LONG	0
SYSTOOLSPA	3	DMS	LONG	0
TEMPSPACE1	1	SMS	SYSTEMP	2
USERSPACE1	2	DMS	LONG	2

8 record(s) selected.

## 戻される情報

表 336. SNAP\_GET\_TBSP\_V91 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
TBSP_TYPE	VARCHAR(10)	tablespace_type - 表スペース・タイプ。このインターフェースは、sqlutil.h の定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• DMS</li> <li>• SMS</li> </ul>
TBSP_CONTENT_TYPE	VARCHAR(10)	tablespace_content_type - 表スペースのコンテンツ・タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• ANY</li> <li>• LARGE</li> <li>• SYSTEMP</li> <li>• USRTEMP</li> </ul>
TBSP_PAGE_SIZE	BIGINT	tablespace_page_size - 表スペースのページ・サイズ
TBSP_EXTENT_SIZE	BIGINT	tablespace_extent_size - 表スペースのエクステンツ・サイズ
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 表スペースのプリフェッチ・サイズ
TBSP_CUR_POOL_ID	BIGINT	tablespace_cur_pool_id - 現在使用中のバッファーク・プール
TBSP_NEXT_POOL_ID	BIGINT	tablespace_next_pool_id - 次の始動時に使用されるバッファーク・プール
FS_CACHING	SMALLINT	fs_caching - ファイル・システム・キャッシング
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファーク・プールのデータの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファーク・プールのデータの物理読み取り
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - バッファーク・プール一時データの論理読み取り
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - バッファーク・プール一時データの物理読み取り

表 336. SNAP\_GET\_TBSP\_V91 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - バッファ ー・プール非同期データ読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファ ー・プールへのデータの書き込み
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - バッファ ー・プール非同期データ書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファ ー・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファ ー・プール索引の物理読み取り
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - バッファ ー・プール一時索引の論理読み取り
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - バッファ ー・プール一時索引の物理読み取り
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - バッファ ー・プール非同期索引読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファ ー・プール索引の書き込み
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - バッファ ー・プール非同期索引書き込み
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - バッファ ー・プール XDA データの論理読み取り
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - バッファ ー・プール XDA データの物理読み取り
POOL_XDA_WRITES	BIGINT	pool_xda_writes - バッファ ー・プール XDA データの書き込み
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - バッファ ー・プール非同期 XDA データ読 み取り
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - バッファ ー・プール非同期 XDA データ書 き込み
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - バッファ ー・プール一時 XDA データの論 理読み取り
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - バッファ ー・プール一時 XDA データの物 理読み取り : モニター・エレメン ト
POOL_READ_TIME	BIGINT	pool_read_time - バッファ ー・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファ ー・プール物理書き込み時間の合計

表 336. SNAP\_GET\_TBSP\_V91 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - バッファ ー・プール非同期読み取り時間
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - バッファ ー・プール非同期書き込み時間
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - バッファ ー・プール非同期読み取り要求
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - バッファ ー・プール非同期索引読み取り要求
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - バッファ ー・プール非同期 XDA 読み取り要求
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - バッファ ー・プールの非ビクティム・バッファ ー数
DIRECT_READS	BIGINT	direct_reads - データベースからの 直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直 接書き込み
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要 求
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時 間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時 間
FILES_CLOSED	BIGINT	files_closed - 閉じられたデータベ ース・ファイル
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 読み取り不 能プリフェッチ・ページ
TBSP_REBALANCER_MODE	VARCHAR(10)	tablespace_rebalancer_mode - リバラ ンサー・モード。このインターフェ ースは、sqlmon.h での定義に基づ くテキスト ID を戻します。これ は次のいずれかです。  <ul style="list-style-type: none"> <li>• NO_REBAL</li> <li>• FWD_REBAL</li> <li>• REV_REBAL</li> </ul>
TBSP_USING_AUTO_STORAGE	SMALLINT	tablespace_using_auto_storage - 自動 ストレージが使用可能な表スペース
TBSP_AUTO_RESIZE_ENABLED	SMALLINT	tablespace_auto_resize_enabled - 表 スペースの自動サイズ変更可能

表 336. SNAP\_GET\_TBSP\_V91 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント

## SNAPAGENT\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数 - memory\_pool 論理データ・グループのスナップショット情報の検索

SNAPAGENT\_MEMORY\_POOL 管理ビューおよび

SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数は、エージェント・レベルでのメモリー使用量についての情報を戻します。

**重要:** SNAPAGENT\_MEMORY\_POOL 管理ビューと

SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数は非推奨になり、604 ページの

『MON\_GET\_MEMORY\_POOL - メモリー・プール情報の取得』と 607 ページの『MON\_GET\_MEMORY\_SET - メモリー・セット情報の取得』に置き換えられました。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 1370 ページの『SNAPAGENT\_MEMORY\_POOL 管理ビュー』
- 1371 ページの『SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数』

### SNAPAGENT\_MEMORY\_POOL 管理ビュー

この管理ビューを使用して、現在接続中のデータベースのエージェント・レベルでのメモリー使用量に関する memory\_pool 論理データ・グループのスナップショット情報を取得することができます。

SNAPAGENT\_MEMORY\_POOL 管理ビューを

SNAPAGENT、SNAPAPPL、SNAPAPPL\_INFO、SNAPSTMT、および

SNAPSUBSECTION 管理ビューとともに使用すると、**GET SNAPSHOT FOR**

**APPLICATIONS ON database-alias** CLP コマンドに相当する情報が提供されます。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1374 ページの表 317 を参照してください。

### 許可

このビューを使用するには、以下のいずれかが必要です。

- SNAPAGENT\_MEMORY\_POOL 管理ビューに対する SELECT 特権
- SNAPAGENT\_MEMORY\_POOL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限



- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

メモリー・プールおよびそれらの現在のサイズのリストを取得します。

```
SELECT AGENT_ID, POOL_ID, POOL_CUR_SIZE FROM SYSIBMADM.SNAPAGENT_MEMORY_POOL
```

以下はこの照会の出力例です。

AGENT_ID	POOL_ID	POOL_CUR_SIZE
48	APPLICATION	65536
48	OTHER	65536
48	APPL_CONTROL	65536
47	APPLICATION	65536
47	OTHER	131072
47	APPL_CONTROL	65536
46	OTHER	327680
46	APPLICATION	262144
46	APPL_CONTROL	65536

9 record(s) selected.

## SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数

SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数は SNAPAGENT\_MEMORY\_POOL 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数を SNAP\_GET\_AGENT、SNAP\_GET\_APPL、SNAP\_GET\_APPL\_INFO、SNAP\_GET\_STMT、および SNAP\_GET\_SUBSECTION 表関数とともに使用すると、**GET SNAPSHOT FOR ALL APPLICATIONS** CLP コマンドに相当する情報が提供されます。

戻される可能性のある情報の完全なリストは、1374 ページの表 317 を参照してください。

## 構文

```
▶▶ SNAP_GET_AGENT_MEMORY_POOL ( ( dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されず、アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL

- SYSMANT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

すべてのデータベースのメモリー・プールおよびそれらの現在のサイズのリストを取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AGENT_ID, POOL_ID, POOL_CUR_SIZE
FROM TABLE(SNAP_GET_AGENT_MEMORY_POOL(CAST (NULL AS VARCHAR(128)), -1))
AS T
```

以下はこの照会の出力例です。

DB_NAME	AGENT_ID	POOL_ID	POOL_CUR_SIZE
SAMPLE	48	APPLICATION	65536
SAMPLE	48	OTHER	65536
SAMPLE	48	APPL_CONTROL	65536
SAMPLE	47	APPLICATION	65536
SAMPLE	47	OTHER	131072
SAMPLE	47	APPL_CONTROL	65536
SAMPLE	46	OTHER	327680
SAMPLE	46	APPLICATION	262144
SAMPLE	46	APPL_CONTROL	65536
TESTDB	30	APPLICATION	65536
TESTDB	30	OTHER	65536
TESTDB	30	APPL_CONTROL	65536
TESTDB	29	APPLICATION	65536
TESTDB	29	OTHER	131072
TESTDB	29	APPL_CONTROL	65536
TESTDB	28	OTHER	327680
TESTDB	28	APPLICATION	65536
TESTDB	28	APPL_CONTROL	65536

18 record(s) selected.

## 戻される情報

表 337. SNAPAGENT\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
AGENT_PID	BIGINT	agent_pid - エンジン・ディスパッチ可能単位 (EDU)

表 337. SNAPAGENT\_MEMORY\_POOL 管理ビューおよび  
SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ID	VARCHAR(14)	pool_id - メモリー・プール ID。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• APP_GROUP</li> <li>• APPL_CONTROL</li> <li>• APPLICATION</li> <li>• BP</li> <li>• CAT_CACHE</li> <li>• DATABASE</li> <li>• DFM</li> <li>• FCMBP</li> <li>• IMPORT_POOL</li> <li>• LOCK_MGR</li> <li>• MONITOR</li> <li>• OTHER</li> <li>• PACKAGE_CACHE</li> <li>• QUERY</li> <li>• SHARED_SORT</li> <li>• SORT</li> <li>• STATEMENT</li> <li>• STATISTICS</li> <li>• UTILITY</li> </ul>
POOL_CUR_SIZE	BIGINT	pool_cur_size - メモリー・プールの現行サイズ
POOL_WATERMARK	BIGINT	pool_watermark - メモリー・プール水準点
POOL_CONFIG_SIZE	BIGINT	pool_config_size - メモリー・プールの構成済みサイズ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPDB\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DB\_MEMORY\_POOL 表関数 - データベース・レベルのメモリー使用量情報の検索

SNAPDB\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DB\_MEMORY\_POOL 表関数は、データベース・レベルでのメモリー使用量についての情報を戻します (UNIX プラットフォームの場合のみ)。

SNAPDB\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DB\_MEMORY\_POOL 表関数は、データベース・レベルでのメモリー使用量についての情報を戻します (UNIX プラットフォームの場合のみ)。

**重要:** バージョン 9.7 フィックスパック 5 以降、SNAPDB\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DB\_MEMORY\_POOL 表関数は使用すべきでなくなり、604 ページの『MON\_GET\_MEMORY\_POOL - メモリー・プール情報の取得』および 607 ページの『MON\_GET\_MEMORY\_SET - メモリー・セット情報の取得』に置き換えられました。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 1397 ページの『SNAPDB\_MEMORY\_POOL 管理ビュー』
- 1399 ページの『SNAP\_GET\_DB\_MEMORY\_POOL 表関数』

### SNAPDB\_MEMORY\_POOL 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関するデータベース・レベルのメモリー使用量情報を取得することができます。

SNAPDB\_MEMORY\_POOL 管理ビューを SNAPDB、SNAPDETAILLOG、SNAPHADR、ADMIN\_GET\_STORAGE\_PATHS、および MON\_GET\_HADR とともに使用することにより、**GET SNAPSHOT FOR DATABASE ON database-alias** CLP コマンドと同等の情報が得られます。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1401 ページの表 322 を参照してください。

### 許可

以下のいずれかの権限が必要です。

- SNAPDB\_MEMORY\_POOL 管理ビューに対する SELECT 特権
- SNAPDB\_MEMORY\_POOL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_DB\_MEMORY\_POOL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続中のデータベースである SAMPLE のメモリー・プールおよびそれらの現在のサイズのリストを取得します。

```
SELECT POOL_ID, POOL_CUR_SIZE FROM SYSIBMADM.SNAPDB_MEMORY_POOL
```

以下はこの照会の出力例です。

POOL_ID	POOL_CUR_SIZE
UTILITY	32768
PACKAGE_CACHE	475136
CAT_CACHE	65536
BP	2097152
BP	1081344
BP	540672
BP	278528
BP	147456
BP	81920
LOCK_MGR	294912
DATABASE	3833856
OTHER	0

12 record(s) selected.

## SNAP\_GET\_DB\_MEMORY\_POOL 表関数

SNAP\_GET\_DB\_MEMORY\_POOL 表関数は SNAPDB\_MEMORY\_POOL 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_DB\_MEMORY\_POOL 表関数を SNAP\_GET\_DB、SNAP\_GET\_DETAILLOG、SNAP\_GET\_HADR、および ADMIN\_GET\_STORAGE\_PATHS 表関数とともに使用すると、**GET SNAPSHOT FOR ALL DATABASES CLP** コマンドに相当する情報が提供されます。

戻される可能性のある情報の完全なリストは、1401 ページの表 322 を参照してください。

## 構文

```
▶▶—SNAP_GET_DB_MEMORY_POOL—(—dbname—  
└──, member──┘)
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DB\_MEMORY\_POOL 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DB\_MEMORY\_POOL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

すべてのデータベースのメモリー・プールおよびそれらの現在のサイズのリストを取得します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, POOL_ID, POOL_CUR_SIZE
FROM TABLE(SNAPSHOT_GET_DB_MEMORY_POOL
(CAST(NULL AS VARCHAR(128)), -1)) AS T
```

以下はこの照会の出力例です。

DB_NAME	POOL_ID	POOL_CUR_SIZE
TESTDB	UTILITY	65536
TESTDB	PACKAGE_CACHE	851968
TESTDB	CAT_CACHE	65536
TESTDB	BP	35913728
TESTDB	BP	589824
TESTDB	BP	327680
TESTDB	BP	196608
TESTDB	BP	131072
TESTDB	SHARED_SORT	65536
TESTDB	LOCK_MGR	10092544
TESTDB	DATABASE	4980736
TESTDB	OTHER	196608
SAMPLE	UTILITY	65536
SAMPLE	PACKAGE_CACHE	655360
SAMPLE	CAT_CACHE	131072
SAMPLE	BP	4325376
SAMPLE	BP	589824
SAMPLE	BP	327680
SAMPLE	BP	196608
SAMPLE	BP	131072
SAMPLE	SHARED_SORT	0
SAMPLE	LOCK_MGR	655360
SAMPLE	DATABASE	4653056
SAMPLE	OTHER	196608

24 record(s) selected.

### 戻される情報

表 338. SNAPDB\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DB\_MEMORY\_POOL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名



表 338. SNAPDB\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DB\_MEMORY\_POOL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ID	VARCHAR(14)	pool_id - メモリー・プール ID。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• APP_GROUP</li> <li>• APPL_CONTROL</li> <li>• APPLICATION</li> <li>• BP</li> <li>• CAT_CACHE</li> <li>• DATABASE</li> <li>• DFM</li> <li>• FCMBP</li> <li>• IMPORT_POOL</li> <li>• LOCK_MGR</li> <li>• MONITOR</li> <li>• OTHER</li> <li>• PACKAGE_CACHE</li> <li>• QUERY</li> <li>• SHARED_SORT</li> <li>• SORT</li> <li>• STATEMENT</li> <li>• STATISTICS</li> <li>• UTILITY</li> </ul>
POOL_SECONDARY_ID	VARCHAR(32)	pool_secondary_id - メモリー・プール 2 次 ID
POOL_CUR_SIZE	BIGINT	pool_cur_size - メモリー・プールの現行サイズ
POOL_WATERMARK	BIGINT	pool_watermark - メモリー・プール水準点
POOL_CONFIG_SIZE	BIGINT	pool_config_size - メモリー・プールの構成済みサイズ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPDBM\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DBM\_MEMORY\_POOL 表関数 - データベース・マ ネージャー・レベルのメモリー使用量情報の検索

SNAPDBM\_MEMORY\_POOL 管理ビューおよび

SNAP\_GET\_DBM\_MEMORY\_POOL 表関数は、データベース・マネージャーでのメモリー使用量についての情報を戻します。

**重要:** バージョン 9.7 フィックスパック 5 以降、SNAPDBM\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DBM\_MEMORY\_POOL 表関数は使用すべきでなくなり、604 ページの『MON\_GET\_MEMORY\_POOL - メモリー・プール情報の取得』および 607 ページの『MON\_GET\_MEMORY\_SET - メモリー・セット情報の取得』に置き換えられました。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 1405 ページの『SNAPDBM\_MEMORY\_POOL 管理ビュー』
- 1406 ページの『SNAP\_GET\_DBM\_MEMORY\_POOL 表関数』

### SNAPDBM\_MEMORY\_POOL 管理ビュー

SNAPDBM\_MEMORY\_POOL 管理ビューを

SNAPDBM、SNAPFCM、SNAPFCM\_PART、および SNAPSWITCHES 管理ビューとともに使用すると、**GET SNAPSHOT FOR DBM** コマンドに相当するデータが提供されます。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1408 ページの表 324 を参照してください。

### 許可

以下のいずれかの権限が必要です。

- SNAPDBM\_MEMORY\_POOL 管理ビューに対する SELECT 特権
- SNAPDBM\_MEMORY\_POOL 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_DBM\_MEMORY\_POOL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

接続中のデータベースのデータベース・マネージャーのメモリー・プールおよびそれらの現在のサイズのリストを取得します。

```
SELECT POOL_ID, POOL_CUR_SIZE FROM SNAPDBM_MEMORY_POOL
```

以下はこの照会の出力例です。

POOL_ID	POOL_CUR_SIZE
MONITOR	65536
OTHER	29622272
FCMBP	57606144
...	

## SNAP\_GET\_DBM\_MEMORY\_POOL 表関数

SNAP\_GET\_DBM\_MEMORY\_POOL 表関数は SNAPDBM\_MEMORY\_POOL 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーを対象とした情報を検索することができます。

SNAP\_GET\_DBM\_MEMORY\_POOL 表関数を SNAP\_GET\_DBM、SNAP\_GET\_FCM、SNAP\_GET\_FCM\_PART、および SNAP\_GET\_SWITCHES 表関数とともに使用すると、GET SNAPSHOT FOR DBM コマンドに相当するデータが提供されます。

戻される可能性のある情報の完全なリストは、1408 ページの表 324 を参照してください。

### 構文

```
▶▶ SNAP_GET_DBM_MEMORY_POOL ( (member) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

*member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。この入力オプションを使用しない場合、データはすべてのアクティブなデータベース・メンバーから戻されます。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*member* を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャーにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_DBM\_MEMORY\_POOL 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_DBM\_MEMORY\_POOL 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

接続中のデータベースのデータベース・マネージャーのすべてのデータベース・パーティションのメモリー・プールおよびそれらの現在のサイズのリストを取得します。

```
SELECT POOL_ID, POOL_CUR_SIZE, DBPARTITIONNUM
FROM TABLE(SYSPROC.SNAP_GET_DBM_MEMORY_POOL())
AS T ORDER BY DBPARTITIONNUM
```

以下はこの照会の出力例です。

POOL_ID	POOL_CUR_SIZE	DBPARTITIONNUM
MONITOR	65536	0
OTHER	29622272	0
FCMBP	57606144	0
MONITOR	65536	1
OTHER	29425664	1
FCMBP	57606144	1
MONITOR	65536	2
OTHER	29425664	2
FCMBP	57606144	2

## 戻される情報

表 339. SNAPDBM\_MEMORY\_POOL 管理ビューおよび SNAP\_GET\_DBM\_MEMORY\_POOL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
POOL_ID	VARCHAR(14)	pool_id - メモリー・プール ID。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• APP_GROUP</li> <li>• APPL_CONTROL</li> <li>• APPLICATION</li> <li>• BP</li> <li>• CAT_CACHE</li> <li>• DATABASE</li> <li>• DFM</li> <li>• FCMBP</li> <li>• IMPORT_POOL</li> <li>• LOCK_MGR</li> <li>• MONITOR</li> <li>• OTHER</li> <li>• PACKAGE_CACHE</li> <li>• QUERY</li> <li>• SHARED_SORT</li> <li>• SORT</li> <li>• STATEMENT</li> <li>• STATISTICS</li> <li>• UTILITY</li> </ul>
POOL_CUR_SIZE	BIGINT	pool_cur_size - メモリー・プールの現行サイズ
POOL_WATERMARK	BIGINT	pool_watermark - メモリー・プール水準点
POOL_CONFIG_SIZE	BIGINT	pool_config_size - メモリー・プールの構成済みサイズ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPHADR 管理ビューおよび SNAP\_GET\_HADR 表関数 - hadr 論理データ・グループのスナップショット情報の検索

SNAPHADR 管理ビューおよび SNAP\_GET\_HADR 表関数は、データベース・スナップショットから、特に hadr 論理データ・グループの高可用性災害時リカバリー情報を戻します。

SNAPHADR 管理ビューおよび SNAP\_GET\_HADR 表関数は使用すべきではなく、584 ページの『MON\_GET\_HADR 表関数 - 高可用性災害時リカバリー (HADR) モニター情報を戻す』に置き換えられました。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 1426 ページの『SNAPHADR 管理ビュー』
- 1427 ページの『SNAP\_GET\_HADR 表関数』

### SNAPHADR 管理ビュー

この管理ビューを使用して、現在接続中のデータベースに関する hadr 論理データ・グループのスナップショット情報を取得することができます。データベースが 1 次またはスタンバイ高可用性災害時リカバリー (HADR) データベースの場合にのみ、このビューによってデータが戻されます。

SNAPHADR 管理ビューを SNAPDB、SNAPDB\_MEMORY\_POOL、SNAPDETAILLOG、および ADMIN\_GET\_STORAGE\_PATHS 表関数とともに使用すると、**GET SNAPSHOT FOR DATABASE ON database-alias** CLP コマンドに相当する情報が提供されます。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1429 ページの表 328 を参照してください。

### 許可

以下のいずれかの権限が必要です。

- SNAPHADR 管理ビューに対する SELECT 特権
- SNAPHADR 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_HADR 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT

- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

1 次 HADR データベース上の HADR に関する構成および状況情報を取得します。

```
SELECT SUBSTR(DB_NAME, 1, 8) AS DBNAME, HADR_ROLE, HADR_STATE,
       HADR_SYNCMODE, HADR_CONNECT_STATUS
FROM SYSIBMADM.SNAPHADR
```

以下はこの照会の出力例です。

DBNAME	HADR_ROLE	HADR_STATE	HADR_SYNCMODE	HADR_CONNECT_STATUS
SAMPLE	PRIMARY	PEER	SYNC	CONNECTED

1 record(s) selected.

## SNAP\_GET\_HADR 表関数

SNAP\_GET\_HADR 表関数は SNAPHADR 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_HADR 表関数を SNAP\_GET\_DB、SNAP\_GET\_DB\_MEMORY\_POOL、SNAP\_GET\_DETAILLOG、および ADMIN\_GET\_STORAGE\_PATHS 表関数とともに使用すると、GET SNAPSHOT FOR ALL DATABASES CLP コマンドに相当する情報が提供されます。

戻される可能性のある情報の完全なリストは、1429 ページの表 328 を参照してください。

### 構文

```
▶▶ SNAP_GET_HADR ( ( dbname [ , member ] ) )
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、LIST DATABASE DIRECTORY コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリ

ングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されず、アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_HADR 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_HADR 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべてのデータベースの HADR に関する構成および状況情報を取得します。

```
SELECT SUBSTR(DB_NAME, 1, 8) AS DBNAME, HADR_ROLE, HADR_STATE,  
       HADR_SYNCMODE, HADR_CONNECT_STATUS  
FROM TABLE (SNAP_GET_HADR (CAST (NULL as VARCHAR(128)), 0)) as T
```

以下はこの照会の出力例です。



DBNAME	HADR_ROLE	HADR_STATE	HADR_SYNCMODE	HADR_CONNECT_STATUS
SAMPLE	PRIMARY	PEER	SYNC	CONNECTED
TESTDB	PRIMARY	DISCONNECTED	NEARSYNC	DISCONNECTED

2 record(s) selected.

## 戻される情報

表 340. SNAPHADR 管理ビューおよび SNAP\_GET\_HADR 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
HADR_ROLE	VARCHAR(10)	<p>hadr_role - HADR のロール。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• PRIMARY</li> <li>• STANDARD</li> <li>• STANDBY</li> </ul>
HADR_STATE	VARCHAR(14)	<p>hadr_state - HADR の状態。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• DISCONNECTED</li> <li>• LOCAL_CATCHUP</li> <li>• PEER</li> <li>• REM_CATCH_PEN</li> <li>• REM_CATCHUP</li> </ul>
HADR_SYNCMODE	VARCHAR(10)	<p>hadr_syncmode - HADR 同期モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• ASYNC</li> <li>• NEARSYNC</li> <li>• SUPERASYNC</li> <li>• SYNC</li> </ul>
HADR_CONNECT_STATUS	VARCHAR(12)	<p>hadr_connect_status - HADR 接続状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。</p> <ul style="list-style-type: none"> <li>• CONGESTED</li> <li>• CONNECTED</li> <li>• DISCONNECTED</li> </ul>

表 340. SNAPHADR 管理ビューおよび SNAP\_GET\_HADR 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
HADR_CONNECT_TIME	TIMESTAMP	hadr_connect_time - HADR 接続時刻
HADR_HEARTBEAT	INTEGER	hadr_heartbeat - HADR ハートビート
HADR_LOCAL_HOST	VARCHAR(255)	hadr_local_host - HADR ローカル・ホスト
HADR_LOCAL_SERVICE	VARCHAR(40)	hadr_local_service - HADR ローカル・サービス
HADR_REMOTE_HOST	VARCHAR(255)	hadr_remote_host - HADR リモート・ホスト
HADR_REMOTE_SERVICE	VARCHAR(40)	hadr_remote_service - HADR リモート・サービス
HADR_REMOTE_INSTANCE	VARCHAR(128)	hadr_remote_instance - HADR リモート・インスタンス
HADR_TIMEOUT	BIGINT	hadr_timeout - HADR タイムアウト
HADR_PRIMARY_LOG_FILE	VARCHAR(255)	hadr_primary_log_file - HADR 1 次ログ・ファイル
HADR_PRIMARY_LOG_PAGE	BIGINT	hadr_primary_log_page - HADR 1 次ログ・ページ
HADR_PRIMARY_LOG_LSN	BIGINT	hadr_primary_log_lsn - HADR 1 次ログ LSN
HADR_STANDBY_LOG_FILE	VARCHAR(255)	hadr_standby_log_file - HADR スタンバイ・ログ・ファイル
HADR_STANDBY_LOG_PAGE	BIGINT	hadr_standby_log_page - HADR スタンバイ・ログ・ページ
HADR_STANDBY_LOG_LSN	BIGINT	hadr_standby_log_lsn - HADR スタンバイ・ログ LSN
HADR_LOG_GAP	BIGINT	hadr_log_gap - HADR ログ・ギャップ
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数 - lock 論理データ・グループのスナップショット情報の検索

SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数は、特に lock 論理データ・グループのロック・スナップショット情報を戻します。

注: この管理ビューおよび表関数は非推奨になり、以下のものに置き換えられました。すなわち、526 ページの『MON\_GET\_APPL\_LOCKWAIT - アプリケーション

が待機しているロックについての情報の取得』、598ページの『MON\_GET\_LOCKS - 現在接続されているデータベース内のすべてのロックのリスト』、および482ページの『MON\_FORMAT\_LOCK\_NAME - 内部ロック名のフォーマット設定と詳細の出力』。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 1431ページの『SNAPLOCK 管理ビュー』
- 1432ページの『SNAP\_GET\_LOCK 表関数』

## SNAPLOCK 管理ビュー

この管理ビューでは、現在接続されているデータベースの lock 論理データ・グループ・スナップショット情報を検索できます。

SNAPLOCKWAIT 管理ビューと共に使用すると、SNAPLOCK 管理ビューは、**GET SNAPSHOT FOR LOCKS ON database-alias** CLP コマンドと同等の情報を提供します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1434ページの表 329 を参照してください。

## 許可

以下のいずれかの権限が必要です。

- SNAPLOCK 管理ビューに対する SELECT 特権
- SNAPLOCK 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_LOCK 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースのデータベース・メンバー 0 のロック情報を検索します。

```
SELECT AGENT_ID, LOCK_OBJECT_TYPE, LOCK_MODE, LOCK_STATUS
FROM SYSIBMADM.SNAPLOCK WHERE DBPARTITIONNUM = 0
```

以下はこの照会の出力例です。

```
AGENT_ID          LOCK_OBJECT_TYPE LOCK_MODE LOCK_STATUS
-----
              7 TABLE                IX      GRNT
```

1 record(s) selected.

## SNAP\_GET\_LOCK 表関数

SNAP\_GET\_LOCK 表関数は SNAPLOCK 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_LOCKWAIT 表関数と共に使用すると、SNAP\_GET\_LOCK 表関数は、**GET SNAPSHOT FOR LOCKS ON database-alias** CLP コマンドと同等の情報を提供します。

戻される可能性のある情報の完全なリストは、1434 ページの表 329 を参照してください。

## 構文

```
→ SNAP_GET_LOCK ( ( dbname [ , member ] ) ) →
```

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値または空ストリングを指定します。

### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を NULL に設定すると、member は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり dbname のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されま

す。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_LOCK 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_LOCK 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースの現行データベース・メンバーのロック情報を検索します。

```
SELECT AGENT_ID, LOCK_OBJECT_TYPE, LOCK_MODE, LOCK_STATUS
FROM TABLE(SNAP_GET_LOCK(' ', -1)) as T
```

以下はこの照会の出力例です。

AGENT_ID	LOCK_OBJECT_TYPE	LOCK_MODE	LOCK_STATUS
680	INTERNALV_LOCK	S	GRNT
680	INTERNALP_LOCK	S	GRNT

2 record(s) selected.

## 戻される情報

表 341. SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)

表 341. SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TAB_FILE_ID	BIGINT	table_file_id - 表ファイル ID
LOCK_OBJECT_TYPE	VARCHAR(18)	lock_object_type - 待機中のロック対象タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• AUTORESIZE_LOCK</li> <li>• AUTOSTORAGE_LOCK</li> <li>• BLOCK_LOCK</li> <li>• EOT_LOCK</li> <li>• INPLACE_REORG_LOCK</li> <li>• INTERNAL_LOCK</li> <li>• INTERNALB_LOCK</li> <li>• INTERNALC_LOCK</li> <li>• INTERNALJ_LOCK</li> <li>• INTERNALL_LOCK</li> <li>• INTERNALO_LOCK</li> <li>• INTERNALQ_LOCK</li> <li>• INTERNALP_LOCK</li> <li>• INTERNALS_LOCK</li> <li>• INTERNALT_LOCK</li> <li>• INTERNALV_LOCK</li> <li>• KEYVALUE_LOCK</li> <li>• ROW_LOCK</li> <li>• SYSBOOT_LOCK</li> <li>• TABLE_LOCK</li> <li>• TABLE_PART_LOCK</li> <li>• TABLESPACE_LOCK</li> <li>• XML_PATH_LOCK</li> </ul>

表 341. SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_MODE	VARCHAR(10)	lock_mode - ロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
LOCK_STATUS	VARCHAR(10)	lock_status - ロック状況。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• CONV</li> <li>• GRNT</li> </ul>
LOCK_ESCALATION	SMALLINT	lock_escalation - ロック・エスカレーション
TABNAME	VARCHAR(128)	table_name - 表名
TABSHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名

表 341. SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_ATTRIBUTES	VARCHAR(128)	lock_attributes - ロック属性。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。ロックがない場合、テキスト ID は NONE となり、それ以外の場合、以下のいずれかの組み合わせを '+' 記号で区切ったものとなります。 <ul style="list-style-type: none"> <li>• ALLOW_NEW</li> <li>• DELETE_IN_BLOCK</li> <li>• ESCALATED</li> <li>• INSERT</li> <li>• NEW_REQUEST</li> <li>• RR</li> <li>• RR_IN_BLOCK</li> <li>• UPDATE_DELETE</li> <li>• WAIT_FOR_AVAIL</li> </ul>
LOCK_COUNT	BIGINT	lock_count - ロック・カウント
LOCK_CURRENT_MODE	VARCHAR(10)	lock_current_mode - 移行前の元のロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
LOCK_HOLD_COUNT	BIGINT	lock_hold_count - ロック保留カウント
LOCK_NAME	VARCHAR(32)	lock_name - ロック名
LOCK_RELEASE_FLAGS	BIGINT	lock_release_flags - ロック保留解除フラグ



表 341. SNAPLOCK 管理ビューおよび SNAP\_GET\_LOCK 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID。非パーティション表では、このエレメントは NULL です。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数 - lockwait 論理データ・グループのスナップショット情報の検索

SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数は、特に lockwait 論理データ・グループのロック待機スナップショット情報を戻します。

注: この管理ビューおよび表関数は非推奨になり、以下のものに置き換えられました。すなわち、758 ページの『MON\_LOCKWAITS 管理ビュー - ロックの取得を待機しているアプリケーションに関するメトリックの取得』、および 526 ページの『MON\_GET\_APPL\_LOCKWAIT - アプリケーションが待機しているロックについての情報の取得』、598 ページの『MON\_GET\_LOCKS - 現在接続されているデータベース内のすべてのロックのリスト』、482 ページの『MON\_FORMAT\_LOCK\_NAME - 内部ロック名のフォーマット設定と詳細の出力』。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 1438 ページの『SNAPLOCKWAIT 管理ビュー』
- 1439 ページの『SNAP\_GET\_LOCKWAIT 表関数』

### SNAPLOCKWAIT 管理ビュー

この管理ビューでは、現在接続されているデータベースの lockwait 論理データ・グループ・スナップショット情報を検索できます。

SNAPLOCK 管理ビューと共に使用すると、SNAPLOCKWAIT 管理ビューは、**GET SNAPSHOT FOR LOCKS ON database-alias** CLP コマンドと同等の情報を提供しません。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1441 ページの表 330 を参照してください。

## 許可

以下のいずれかの権限が必要です。

- SNAPLOCKWAIT 管理ビューに対する SELECT 特権
- SNAPLOCKWAIT 管理ビューに対する CONTROL 特権
- DATAACCESS 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_LOCKWAIT 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースのデータベース・メンバー 0 のロック待機情報を検索します。

```
SELECT AGENT_ID, LOCK_MODE, LOCK_OBJECT_TYPE, AGENT_ID_HOLDING_LK,  
       LOCK_MODE_REQUESTED FROM SYSIBMADM.SNAPLOCKWAIT  
WHERE DBPARTITIONNUM = 0
```

以下はこの照会の出力例です。

```
AGENT_ID    LOCK_MODE LOCK_OBJECT_TYPE ...  
-----  
          7 IX          TABLE          ...
```

1 record(s) selected.

この照会からの出力 (続き)。

```
... AGENT_ID_HOLDING_LK LOCK_MODE_REQUESTED  
... -----  
...                      12 IS
```

## SNAP\_GET\_LOCKWAIT 表関数

SNAP\_GET\_LOCKWAIT 表関数は SNAPLOCKWAIT 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_LOCK 表関数と共に使用すると、SNAP\_GET\_LOCKWAIT 表関数は、**GET SNAPSHOT FOR LOCKS ON database-alias** CLP コマンドと同等の情報を提供します。

戻される可能性のある情報の完全なリストは、1441 ページの表 330 を参照してください。

## 構文

```
▶▶ SNAP_GET_LOCKWAIT ( ( dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値または空ストリングを指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のデータベース・メンバーの場合は -1、すべてアクティブなデータベース・メンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはすべてのアクティブなデータベース・メンバーから戻されず、アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_LOCKWAIT 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_LOCKWAIT 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON

- SYSCtrl
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

### 例

現在接続されているデータベースの現行データベース・メンバーのロック待機情報を検索します。

```
SELECT AGENT_ID, LOCK_MODE, LOCK_OBJECT_TYPE, AGENT_ID_HOLDING_LK,
       LOCK_MODE_REQUESTED FROM TABLE(SNAP_GET_LOCKWAIT(' ', -1)) AS T
```

以下はこの照会の出力例です。

```
AGENT_ID      LOCK_MODE  LOCK_OBJECT_TYPE  ...
-----
          12 X          ROW_LOCK          ...
```

1 record(s) selected.

この照会からの出力 (続き)。

```
... AGENT_ID_HOLDING_LK  LOCK_MODE_REQUESTED
... -----
...                   7 X
```

### 使用上の注意

ロック待機情報を表示するには、まずデータベース・マネージャー構成でデフォルトの LOCK モニター・スイッチをオンにする必要があります。変更を即時に有効にするには、CLP を使用してインスタンスに明示的にアタッチし、次いで以下の CLP コマンドを発行します。

```
UPDATE DATABASE MANAGER CONFIGURATION CLP USING DFT_MON_LOCK ON
```

デフォルトの設定値も、ADMIN\_CMD ストアード・プロシージャによりオンにすることができます。例:

```
CALL SYSPROC.ADMIN_CMD('update dbm cfg using DFT_MON_LOCK ON')
```

ADMIN\_CMD ストアード・プロシージャを使用する場合、または事前にインスタンスにアタッチせずに CLP コマンドを使用する場合、インスタンスをリサイクルしなければ変更は有効になりません。

### 戻される情報

表 342. SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。

表 342. SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
SUBSECTION_NUMBER	BIGINT	ss_number - サブセクション番号
LOCK_MODE	VARCHAR(10)	lock_mode - ロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>

表 342. SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_OBJECT_TYPE	VARCHAR(18)	lock_object_type - 待機中のロック対象タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• AUTORESIZE_LOCK</li> <li>• AUTOSTORAGE_LOCK</li> <li>• BLOCK_LOCK</li> <li>• EOT_LOCK</li> <li>• INPLACE_REORG_LOCK</li> <li>• INTERNAL_LOCK</li> <li>• INTERNALB_LOCK</li> <li>• INTERNALC_LOCK</li> <li>• INTERNALJ_LOCK</li> <li>• INTERNALL_LOCK</li> <li>• INTERNALO_LOCK</li> <li>• INTERNALQ_LOCK</li> <li>• INTERNALP_LOCK</li> <li>• INTERNALS_LOCK</li> <li>• INTERNALT_LOCK</li> <li>• INTERNALV_LOCK</li> <li>• KEYVALUE_LOCK</li> <li>• ROW_LOCK</li> <li>• SYSBOOT_LOCK</li> <li>• TABLE_LOCK</li> <li>• TABLE_PART_LOCK</li> <li>• TABLESPACE_LOCK</li> <li>• XML_PATH_LOCK</li> </ul>
AGENT_ID_HOLDING_LK	BIGINT	agent_id_holding_lock - ロックを保持しているエージェント ID
LOCK_WAIT_START_TIME	TIMESTAMP	lock_wait_start_time - ロック待機開始タイム・スタンプ

表 342. SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_MODE_REQUESTED	VARCHAR(10)	lock_mode_requested - 要求されているロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
LOCK_ESCALATION	SMALLINT	lock_escalation - ロック・エスカレーション
TABNAME	VARCHAR(128)	table_name - 表名
TABSHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TBSP_NAME	VARCHAR(128)	tablespace_name - 表スペース名
APPL_ID_HOLDING_LK	VARCHAR(128)	appl_id_holding_lk - ロックを保持しているアプリケーション ID
LOCK_ATTRIBUTES	VARCHAR(128)	lock_attributes - ロック属性。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。ロックがない場合、テキスト ID は NONE となり、それ以外の場合、以下のいずれかの組み合わせを '+' 記号で区切ったものとなります。 <ul style="list-style-type: none"> <li>• ALLOW_NEW</li> <li>• DELETE_IN_BLOCK</li> <li>• ESCALATED</li> <li>• INSERT</li> <li>• NEW_REQUEST</li> <li>• RR</li> <li>• RR_IN_BLOCK</li> <li>• UPDATE_DELETE</li> <li>• WAIT_FOR_AVAIL</li> </ul>

表 342. SNAPLOCKWAIT 管理ビューおよび SNAP\_GET\_LOCKWAIT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_CURRENT_MODE	VARCHAR(10)	lock_current_mode - 移行前の元のロック・モード。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (ロックなしの場合)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>
LOCK_NAME	VARCHAR(32)	lock_name - ロック名
LOCK_RELEASE_FLAGS	BIGINT	lock_release_flags - ロック保留解除フラグ。
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID。非パーティション表では、このエレメントは NULL です。
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SNAPSHOT\_AGENT

SNAPSHOT\_AGENT 関数は、アプリケーション・スナップショットからのエージェントに関する情報を戻します。

注: この表関数は使用すべきではなく、841 ページの『SNAPAGENT 管理ビューおよび SNAP\_GET\_AGENT 表関数 - agent 論理データ・グループのアプリケーション・スナップショット情報の検索』に置き換えられました。

▶▶—SNAPSHOT\_AGENT—(—dbname—, —member—)——▶▶

スキーマは SYSPROC です。



## 表関数パラメーター

### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。データベース・インスタンス下のすべてのデータベースからスナップショットを取る場合は、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの **SNAPSHOT\_FILEW** ストアード・プロシージャーによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 343. *SNAPSHOT\_AGENT* 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
AGENT_PID	BIGINT	agent_pid - エンジン・ディスパッチ可能単位 (EDU)

## SNAPSHOT\_APPL

アプリケーション・スナップショットからの一般情報を戻します。

注: この表関数は使用すべきではなく、854 ページの『SNAP\_GET\_APPL 表関数』に置き換えられました。

▶▶—SNAPSHOT\_APPL—(—dbname—,—member—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。データベース・インスタンス下のすべてのデータベースからスナップショットを取る場合は、NULL 値を指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアード・プロシージャーによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

### 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 344. SNAPSHOT\_APPL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
UOW_LOG_SPACE_USED	BIGINT	uow_log_space_used - 作業単位ログ・スペース
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファークール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファークール・データの物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファークールへのデータの書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファークール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファークール索引の物理読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファークール索引の書き込み
POOL_READ_TIME	BIGINT	pool_read_time - バッファークール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファークール物理書き込み時間の合計
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 直接書き込み要求
DIRECT_READ_TIME	BIGINT	direct_read_time - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 直接書き込み時間
POOL_DATA_TO_ESTORE	BIGINT	pool_data_to_estore ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。

表 344. SNAPSHOT\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_INDEX_TO_ESTORE	BIGINT	<b>pool_index_to_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
POOL_INDEX_FROM_ESTORE	BIGINT	<b>pool_index_from_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
POOL_DATA_FROM_ESTORE	BIGINT	<b>pool_data_from_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
UNREAD_PREFETCH_PAGES	BIGINT	<b>unread_prefetch_pages</b> - 読み取り不能プリフェッチ・ページ
LOCKS_HELD	BIGINT	<b>locks_held</b> - ロック保持数
LOCK_WAITS	BIGINT	<b>lock_waits</b> - ロック待機数
LOCK_WAIT_TIME	BIGINT	<b>lock_wait_time</b> - ロック待機中の時間
LOCK_ESCALS	BIGINT	<b>lock_escalations</b> - ロック・エスカレーション数
X_LOCK_ESCALS	BIGINT	<b>x_lock_escalations</b> - 排他ロック・エスカレーション数
DEADLOCKS	BIGINT	<b>deadlocks</b> - デッドロック検出数
TOTAL_SORTS	BIGINT	<b>total_sorts</b> - ソート合計
TOTAL_SORT_TIME	BIGINT	<b>total_sort_time</b> - ソート時間合計
SORT_OVERFLOWS	BIGINT	<b>sort_overflows</b> - ソート・オーバーフロー
COMMIT_SQL_STMTS	BIGINT	<b>commit_sql_stmts</b> - 試行されたコミット・ステートメント
ROLLBACK_SQL_STMTS	BIGINT	<b>rollback_sql_stmts</b> - 試行されたロールバック・ステートメント
DYNAMIC_SQL_STMTS	BIGINT	<b>dynamic_sql_stmts</b> - 試行された動的 SQL ステートメント
STATIC_SQL_STMTS	BIGINT	<b>static_sql_stmts</b> - 試行された静的 SQL ステートメント
FAILED_SQL_STMTS	BIGINT	<b>failed_sql_stmts</b> - 失敗したステートメント操作
SELECT_SQL_STMTS	BIGINT	<b>select_sql_stmts</b> - 実行された選択 SQL ステートメント
DDL_SQL_STMTS	BIGINT	<b>ddl_sql_stmts</b> - データ定義言語 (DDL) SQL ステートメント

表 344. SNAPSHOT\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
UID_SQL_STMTS	BIGINT	<b>uid_sql_stmts</b> - 実行された更新/挿入/削除 SQL ステートメント
INT_AUTO_REBINDS	BIGINT	<b>int_auto_rebinds</b> - 内部自動再バインド
INT_ROWS_DELETED	BIGINT	<b>int_rows_deleted</b> - 削除された内部行数
INT_ROWS_UPDATED	BIGINT	<b>int_rows_updated</b> - 更新された内部行数
INT_COMMITS	BIGINT	<b>int_commits</b> - 内部コミット数
INT_ROLLBACKS	BIGINT	<b>int_rollback</b> s - 内部ロールバック数
INT_DEADLOCK_ROLLBACKS	BIGINT	<b>int_deadlock_rollback</b> s - デッドロックによる内部ロールバック数
ROWS_DELETED	BIGINT	<b>rows_deleted</b> - 削除行数
ROWS_INSERTED	BIGINT	<b>rows_inserted</b> - 挿入行数
ROWS_UPDATED	BIGINT	<b>rows_updated</b> - 更新行数
ROWS_SELECTED	BIGINT	<b>rows_selected</b> - 選択行数
BINDS_PRECOMPILES	BIGINT	<b>binds_precompiles</b> - 試行されたバインド/プリコンパイル
OPEN_REM_CURS	BIGINT	<b>open_rem_curs</b> - 開かれているリモート・カーソル
OPEN_REM_CURS_BLK	BIGINT	<b>open_rem_curs_blk</b> - 開かれているリモート・ブロック・カーソル
REJ_CURS_BLK	BIGINT	<b>rej_curs_blk</b> - リジェクトされたブロック・カーソル要求
ACC_CURS_BLK	BIGINT	<b>acc_curs_blk</b> - 受け入れられたブロック・カーソル要求
SQL_REQS_SINCE_COMMIT	BIGINT	<b>sql_reqs_since_commit</b> - 最終コミット後の SQL 要求数
LOCK_TIMEOUTS	BIGINT	<b>lock_timeouts</b> - ロック・タイムアウト数
INT_ROWS_INSERTED	BIGINT	<b>int_rows_inserted</b> - 挿入された内部行数
OPEN_LOC_CURS	BIGINT	<b>open_loc_curs</b> - 開かれているローカル・カーソル
OPEN_LOC_CURS_BLK	BIGINT	<b>open_loc_curs_blk</b> - 開かれているローカル・ブロック・カーソル
PKG_CACHE_LOOKUPS	BIGINT	<b>pkg_cache_lookups</b> - パッケージ・キャッシュ参照
PKG_CACHE_INSERTS	BIGINT	<b>pkg_cache_inserts</b> - パッケージ・キャッシュ挿入

表 344. SNAPSHOT\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
CAT_CACHE_LOOKUPS	BIGINT	<b>cat_cache_lookups</b> - カタログ・キャッシュ参照数
CAT_CACHE_INSERTS	BIGINT	<b>cat_cache_inserts</b> - カタログ・キャッシュ挿入数
CAT_CACHE_OVERFLOWS	BIGINT	<b>cat_cache_overflows</b> - カタログ・キャッシュ・オーバーフロー数
CAT_CACHE_HEAP_FULL	BIGINT	<b>cat_cache_heap_full</b> - カタログ・キャッシュ・ヒープ・フル
NUM_AGENTS	BIGINT	<b>num_agents</b> - ステートメントで作動しているエージェントの数
AGENTS_STOLEN	BIGINT	<b>agents_stolen</b> - スチールされたエージェント
ASSOCIATED_AGENTS_TOP	BIGINT	<b>associated_agents_top</b> - 関連エージェント最大数
APPL_PRIORITY	BIGINT	<b>appl_priority</b> - アプリケーション・エージェント優先順位
APPL_PRIORITY_TYPE	BIGINT	<b>appl_priority_type</b> - アプリケーション優先順位タイプ
PREFETCH_WAIT_TIME	BIGINT	<b>prefetch_wait_time</b> - プリフェッチ待ち時間
APPL_SECTION_LOOKUPS	BIGINT	<b>appl_section_lookups</b> - セクションの参照回数
APPL_SECTION_INSERTS	BIGINT	<b>appl_section_inserts</b> - セクション挿入数
LOCKS_WAITING	BIGINT	<b>locks_waiting</b> - ロックで待機中の現行エージェント
TOTAL_HASH_JOINS	BIGINT	<b>total_hash_joins</b> - ハッシュ結合の合計
TOTAL_HASH_LOOPS	BIGINT	<b>total_hash_loops</b> - ハッシュ・ループの合計
HASH_JOIN_OVERFLOWS	BIGINT	<b>hash_join_overflows</b> - ハッシュ結合のオーバーフロー
HASH_JOIN_SMALL_OVERFLOWS	BIGINT	<b>hash_join_small_overflows</b> - ハッシュ結合の短精度オーバーフロー
APPL_IDLE_TIME	BIGINT	<b>appl_idle_time</b> - アプリケーション・アイドル時間
UOW_LOCK_WAIT_TIME	BIGINT	<b>uow_lock_wait_time</b> - ロック待機中の作業単位の合計時間
UOW_COMP_STATUS	BIGINT	<b>uow_comp_status</b> - 作業単位完了状況
AGENT_USR_CPU_TIME_S	BIGINT	<b>agent_usr_cpu_time</b> - エージェントが使用したユーザー CPU 時間 (秒単位)*

表 344. SNAPSHOT\_APPL 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
AGENT_USR_CPU_TIME_MS	BIGINT	<b>agent_usr_cpu_time</b> - エージェントが使用したユーザー CPU 時間 (小数部、マイクロ秒単位)*
AGENT_SYS_CPU_TIME_S	BIGINT	<b>agent_sys_cpu_time</b> - エージェントが使用したシステム CPU 時間 (秒単位)*
AGENT_SYS_CPU_TIME_MS	BIGINT	<b>agent_sys_cpu_time</b> - エージェントが使用したシステム CPU 時間 (小数部、マイクロ秒単位)*
APPL_CON_TIME	TIMESTAMP	<b>appl_con_time</b> - 接続要求開始タイム・スタンプ
CONN_COMPLETE_TIME	TIMESTAMP	<b>conn_complete_time</b> - 接続要求完了タイム・スタンプ
LAST_RESET	TIMESTAMP	<b>last_reset</b> - 最後のリセット・タイム・スタンプ
UOW_START_TIME	TIMESTAMP	<b>uow_start_time</b> - 作業単位開始タイム・スタンプ
UOW_STOP_TIME	TIMESTAMP	<b>uow_stop_time</b> - 作業単位停止タイム・スタンプ
PREV_UOW_STOP_TIME	TIMESTAMP	<b>prev_uow_stop_time</b> - 直前の作業単位完了タイム・スタンプ
UOW_ELAPSED_TIME_S	BIGINT	<b>uow_elapsed_time</b> - 最新の作業単位の経過時間 (秒単位)*
UOW_ELAPSED_TIME_MS	BIGINT	<b>uow_elapsed_time</b> - 最新の作業単位の経過時間 (小数部、マイクロ秒単位)*
ELAPSED_EXEC_TIME_S	BIGINT	<b>elapsed_exec_time</b> - ステートメント実行経過時間 (秒単位)*
ELAPSED_EXEC_TIME_MS	BIGINT	<b>elapsed_exec_time</b> - ステートメント実行経過時間 (小数部、マイクロ秒単位)*
INBOUND_COMM_ADDRESS	VARCHAR(32)	<b>inbound_comm_address</b> - インバウンド通信アドレス
<p>* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する <b>_S</b> で終わっている列で報告されている整数秒と、このモニター・エレメントに関する <b>_MS</b> で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません: <math>(\text{monitor-element-name}_S \times 1,000,000 + \text{monitor-element-name}_{MS}) \div 1,000,000</math>. 例えば、<math>(\text{ELAPSED\_EXEC\_TIME}_S \times 1,000,000 + \text{ELAPSED\_EXEC\_TIME}_{MS}) \div 1,000,000</math>.</p>		

## SNAPSHOT\_APPL\_INFO

アプリケーション・スナップショットからの一般情報を戻します。

注: この表関数は使用すべきではなく、846 ページの『SNAP\_GET\_APPL\_INFO 表関数』に置き換えられました。

▶▶—SNAPSHOT\_APPL\_INFO—(—dbname—, —member—)————▶▶

スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。データベース・インスタンス下のすべてのデータベースからスナップショットを取る場合は、NULL 値を指定します。

### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアード・プロシージャーによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 345. SNAPSHOT\_APPL\_INFO 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ



表 345. SNAPSHOT\_APPL\_INFO 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
AGENT_ID	BIGINT	<b>agent_id</b> - アプリケーション・ハンドル (エージェント ID)
APPL_STATUS	BIGINT	<b>appl_status</b> - アプリケーション状況
CODEPAGE_ID	BIGINT	<b>codepage_id</b> - アプリケーションで使用するコード・ページ ID
NUM_ASSOC_AGENTS	BIGINT	<b>num_assoc_agents</b> - 関連したエージェント数
COORD_PARTITION_NUM	SMALLINT	<b>coord_node</b> - コーディネーター・ノード
AUTHORITY_LVL	BIGINT	<b>authority_lvl</b> - ユーザー許可レベル
CLIENT_PID	BIGINT	<b>client_pid</b> - クライアント・プロセス ID
COORD_AGENT_PID	BIGINT	<b>coord_agent_pid</b> - コーディネーター・エージェント
STATUS_CHANGE_TIME	TIMESTAMP	<b>status_change_time</b> - アプリケーション状況変更時刻
CLIENT_PLATFORM	SMALLINT	<b>client_platform</b> - クライアント・オペレーティング・プラットフォーム
CLIENT_PROTOCOL	SMALLINT	<b>client_protocol</b> - クライアント通信プロトコル
COUNTRY_CODE	SMALLINT	<b>territory_code</b> - データベース・テリトリー・コード
APPL_NAME	VARCHAR(256)	<b>appl_name</b> - アプリケーション名
APPL_ID	VARCHAR(128)	<b>appl_id</b> - アプリケーション ID
SEQUENCE_NO	VARCHAR(4)	<b>sequence_no</b> - シーケンス番号
AUTH_ID	VARCHAR(128)	<b>auth_id</b> - 許可 ID
CLIENT_NNAME	VARCHAR(128)	<b>client_nname</b> - クライアント名モニター・エレメント
CLIENT_PRDID	VARCHAR(128)	<b>client_prdid</b> - クライアント製品/バージョン ID
INPUT_DB_ALIAS	VARCHAR(128)	<b>input_db_alias</b> - 入力データベース別名
CLIENT_DB_ALIAS	VARCHAR(128)	<b>client_db_alias</b> - アプリケーションで使用するデータベース別名

表 345. SNAPSHOT\_APPL\_INFO 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
DB_NAME	VARCHAR(128)	<b>db_name</b> - データベース名
DB_PATH	VARCHAR(1024)	<b>db_path</b> - データベース・パス
EXECUTION_ID	VARCHAR(128)	<b>execution_id</b> - ユーザー・ログイン ID
CORR_TOKEN	VARCHAR(128)	<b>corr_token</b> - DRDA 相関トークン
TPMON_CLIENT_USERID	VARCHAR(256)	<b>tpmon_client_userid</b> - TP モニター・クライアント・ユーザー ID
TPMON_CLIENT_WKSTN	VARCHAR(256)	<b>tpmon_client_wkstn</b> - TP モニター・クライアント・ワークステーション名
TPMON_CLIENT_APP	VARCHAR(256)	<b>tpmon_client_app</b> - TP モニター・クライアント・アプリケーション名
TPMON_ACC_STR	VARCHAR(200)	<b>tpmon_acc_str</b> - TP モニター・クライアント・アカウント・リング・ディング・ストリング

## SNAPSHOT\_BP

バッファ・プール・スナップショットからの情報を戻します。

注: この表関数は使用すべきではなく、864 ページの『SNAP\_GET\_BP 表関数』に置き換えられました。

→→SNAPSHOT\_BP(—dbname—, —member—)←←

スキーマは SYSPROC です。

### 表関数パラメーター

#### dbname

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。データベース・インスタンス下のすべてのデータベースからスナップショットを取る場合は、NULL 値を指定します。

#### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータ

ベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアード・プロシージャーによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 346. SNAPSHOT\_BP 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファークール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファークール・データの物理読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファークールへのデータの書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファークール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファークール索引の物理読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファークール索引の書き込み
POOL_READ_TIME	BIGINT	pool_read_time - バッファークール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	pool_write_time - バッファークール物理書き込み時間の合計
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - バッファークール非同期データ読み取り

表 346. SNAPSHOT\_BP 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_DATA_WRITES	BIGINT	<b>pool_async_data_writes</b> - バッファ・プール非同期データ書き込み
POOL_ASYNC_INDEX_WRITES	BIGINT	<b>pool_async_index_writes</b> - バッファ・プール非同期索引書き込み
POOL_ASYNC_READ_TIME	BIGINT	<b>pool_async_read_time</b> - バッファ・プール非同期読み取り時間
POOL_ASYNC_WRITE_TIME	BIGINT	<b>pool_async_write_time</b> - バッファ・プール非同期書き込み時間
POOL_ASYNC_DATA_READ_REQS	BIGINT	<b>pool_async_data_read_reqs</b> - バッファ・プール非同期読み取り要求
DIRECT_READS	BIGINT	<b>direct_reads</b> - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	<b>direct_writes</b> - データベースへの直接書き込み
DIRECT_READ_REQS	BIGINT	<b>direct_read_reqs</b> - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	<b>direct_write_reqs</b> - 直接書き込み要求
DIRECT_READ_TIME	BIGINT	<b>direct_read_time</b> - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	<b>direct_write_time</b> - 直接書き込み時間
POOL_ASYNC_INDEX_READS	BIGINT	<b>pool_async_index_reads</b> - バッファ・プール非同期索引読み取り
POOL_DATA_TO_ESTORE	BIGINT	<b>pool_data_to_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
POOL_INDEX_TO_ESTORE	BIGINT	<b>pool_index_to_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
POOL_INDEX_FROM_ESTORE	BIGINT	<b>pool_index_from_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
POOL_DATA_FROM_ESTORE	BIGINT	<b>pool_data_from_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
UNREAD_PREFETCH_PAGES	BIGINT	<b>unread_prefetch_pages</b> - 読み取り不能プリフェッチ・ページ

表 346. SNAPSHOT\_BP 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
FILES_CLOSED	BIGINT	<b>files_closed</b> - 閉じられたデータベース・ファイル
BP_NAME	VARCHAR(128)	<b>bp_name</b> - バッファ・プール名
DB_NAME	VARCHAR(128)	<b>db_name</b> - データベース名
DB_PATH	VARCHAR(1024)	<b>db_path</b> - データベース・パス
INPUT_DB_ALIAS	VARCHAR(128)	<b>input_db_alias</b> - 入力データベース別名

## SNAPSHOT\_CONTAINER

表スペース・スナップショットからのコンテナ構成情報を戻します。

注: この表関数は使用すべきではなく、874 ページの『SNAP\_GET\_CONTAINER 表関数』に置き換えられました。

▶▶—SNAPSHOT\_CONTAINER—(—dbname—,—member—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### dbname

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアード・プロシージャーによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

### 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限

- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 347. *SNAPSHOT\_CONTAINER* 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
TABLESPACE_ID	BIGINT	tablespace_id - 表スペース ID
TABLESPACE_NAME	VARCHAR(128)	tablespace_name - 表スペース名
CONTAINER_ID	BIGINT	container_id - コンテナ ID
CONTAINER_NAME	VARCHAR(256)	container_name - コンテナ名
CONTAINER_TYPE	SMALLINT	container_type - コンテナ・タイプ
TOTAL_PAGES	BIGINT	container_total_pages - コンテナ内の合計ページ数
USABLE_PAGES	BIGINT	container_usable_pages - コンテナ内の使用可能なページ数
ACCESSIBLE	BIGINT	container_accessible - コンテナのアクセス可能性
STRIPE_SET	BIGINT	container_stripe_set - ストライプ・セット

## SNAPSHOT\_DATABASE

データベース・スナップショットからの情報を戻します。

注: この表関数は使用すべきではなく、879 ページの『SNAP\_GET\_DB 表関数』に置き換えられました。

▶▶—SNAPSHOT\_DATABASE—(—dbname—, —member—)————▶▶

スキーマは SYSPROC です。

*dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス

内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
 "Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を  
 指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマ  
 ンドで確認できます。データベース・インスタンス下のすべてのデータベースか  
 らスナップショットを取る場合は、NULL 値を指定します。

*member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引  
 数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータ  
 ベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メン  
 バーとは、アプリケーションによるデータベースへの接続およびデータベースの  
 使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット  
 API 要求タイプの **SNAPSHOT\_FILEW** ストアード・プロシージャーによって以前  
 にファイルが作成されていない場合にのみ、スナップショットがとられます。

**許可**

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

**デフォルトの PUBLIC 特権**

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権  
 が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 348. *SNAPSHOT\_DATABASE* 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレ メント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップシヨ ットのタイム・スタンプ
SEC_LOG_USED_TOP	BIGINT	sec_log_used_top - 使用された最 大 2 次ログ・スペース
TOT_LOG_USED_TOP	BIGINT	tot_log_used_top - 使用された最 大合計ログ・スペース
TOTAL_LOG_USED	BIGINT	total_log_used - 使用されている ログ・スペースの合計
TOTAL_LOG_AVAILABLE	BIGINT	total_log_available - 使用可能な ログ合計
ROWS_READ	BIGINT	rows_read - 読み取り行数
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファー・ プール・データの論理読み取り

表 348. SNAPSHOT\_DATABASE 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_DATA_P_READS	BIGINT	<b>pool_data_p_reads</b> - バッファーク・プール・データの物理読み取り
POOL_DATA_WRITES	BIGINT	<b>pool_data_writes</b> - バッファーク・プールへのデータの書き込み
POOL_INDEX_L_READS	BIGINT	<b>pool_index_l_reads</b> - バッファーク・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	<b>pool_index_p_reads</b> - バッファーク・プール索引の物理読み取り
POOL_INDEX_WRITES	BIGINT	<b>pool_index_writes</b> - バッファーク・プール索引の書き込み
POOL_READ_TIME	BIGINT	<b>pool_read_time</b> - バッファーク・プール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	<b>pool_write_time</b> - バッファーク・プール物理書き込み時間の合計
POOL_ASYNC_INDEX_READS	BIGINT	<b>pool_async_index_reads</b> - バッファーク・プール非同期索引読み取り
POOL_DATA_TO_ESTORE	BIGINT	<b>pool_data_to_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
POOL_INDEX_TO_ESTORE	BIGINT	<b>pool_index_to_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
POOL_INDEX_FROM_ESTORE	BIGINT	<b>pool_index_from_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
POOL_DATA_FROM_ESTORE	BIGINT	<b>pool_data_from_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
POOL_ASYNC_DATA_READS	BIGINT	<b>pool_async_data_reads</b> - バッファーク・プール非同期データ読み取り
POOL_ASYNC_DATA_WRITES	BIGINT	<b>pool_async_data_writes</b> - バッファーク・プール非同期データ書き込み
POOL_ASYNC_INDEX_WRITES	BIGINT	<b>pool_async_index_writes</b> - バッファーク・プール非同期索引書き込み
POOL_ASYNC_READ_TIME	BIGINT	<b>pool_async_read_time</b> - バッファーク・プール非同期読み取り時間
POOL_ASYNC_WRITE_TIME	BIGINT	<b>pool_async_write_time</b> - バッファーク・プール非同期書き込み時間



表 348. SNAPSHOT\_DATABASE 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_ASYNC_DATA_READ_REQS	BIGINT	<b>pool_async_data_read_reqs</b> - バッファ・プール非同期読み取り要求
DIRECT_READS	BIGINT	<b>direct_reads</b> - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	<b>direct_writes</b> - データベースへの直接書き込み
DIRECT_READ_REQS	BIGINT	<b>direct_read_reqs</b> - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	<b>direct_write_reqs</b> - 直接書き込み要求
DIRECT_READ_TIME	BIGINT	<b>direct_read_time</b> - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	<b>direct_write_time</b> - 直接書き込み時間
UNREAD_PREFETCH_PAGES	BIGINT	<b>unread_prefetch_pages</b> - 読み取り不能プリフェッチ・ページ
FILES_CLOSED	BIGINT	<b>files_closed</b> - 閉じられたデータベース・ファイル
POOL_LSN_GAP_CLNS	BIGINT	<b>pool_lsn_gap_clns</b> - 起動されたバッファ・プール・ログ・スペース・クリーナー
POOL_DRTY_PG_STEAL_CLNS	BIGINT	<b>pool_drty_pg_steal_clns</b> - 起動されたバッファ・プール・ビクティム・ページ・クリーナー
POOL_DRTY_PG_THRSH_CLNS	BIGINT	<b>pool_drty_pg_thrsh_clns</b> - 起動されたバッファ・プールしきい値クリーナー
LOCKS_HELD	BIGINT	<b>locks_held</b> - ロック保持数
LOCK_WAITS	BIGINT	<b>lock_waits</b> - ロック待機数
LOCK_WAIT_TIME	BIGINT	<b>lock_wait_time</b> - ロック待機中の時間
LOCK_LIST_IN_USE	BIGINT	<b>lock_list_in_use</b> - 使用中のロック・リスト・メモリーの合計
DEADLOCKS	BIGINT	<b>deadlocks</b> - デッドロック検出数
LOCK_ESCALS	BIGINT	<b>lock_escalations</b> - ロック・エスカレーション数
X_LOCK_ESCALS	BIGINT	<b>x_lock_escalations</b> - 排他ロック・エスカレーション数
LOCKS_WAITING	BIGINT	<b>locks_waiting</b> - ロックで待機中の現行エージェント
SORT_HEAP_ALLOCATED	BIGINT	<b>sort_heap_allocated</b> - 割り振られたソート・ヒープの合計
TOTAL_SORTS	BIGINT	<b>total_sorts</b> - ソート合計

表 348. SNAPSHOT\_DATABASE 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エリメント
TOTAL_SORT_TIME	BIGINT	<b>total_sort_time</b> - ソート時間合計
SORT_OVERFLOWS	BIGINT	<b>sort_overflows</b> - ソート・オーバーフロー
ACTIVE_SORTS	BIGINT	<b>active_sorts</b> - アクティブ・ソート
COMMIT_SQL_STMTS	BIGINT	<b>commit_sql_stmts</b> - 試行されたコミット・ステートメント
ROLLBACK_SQL_STMTS	BIGINT	<b>rollback_sql_stmts</b> - 試行されたロールバック・ステートメント
DYNAMIC_SQL_STMTS	BIGINT	<b>dynamic_sql_stmts</b> - 試行された動的 SQL ステートメント
STATIC_SQL_STMTS	BIGINT	<b>static_sql_stmts</b> - 試行された静的 SQL ステートメント
FAILED_SQL_STMTS	BIGINT	<b>failed_sql_stmts</b> - 失敗したステートメント操作
SELECT_SQL_STMTS	BIGINT	<b>select_sql_stmts</b> - 実行された選択 SQL ステートメント
DDL_SQL_STMTS	BIGINT	<b>ddl_sql_stmts</b> - データ定義言語 (DDL) SQL ステートメント
UID_SQL_STMTS	BIGINT	<b>uid_sql_stmts</b> - 実行された更新/挿入/削除 SQL ステートメント
INT_AUTO_REBINDS	BIGINT	<b>int_auto_rebinds</b> - 内部自動再バインド
INT_ROWS_DELETED	BIGINT	<b>int_rows_deleted</b> - 削除された内部行数
INT_ROWS_UPDATED	BIGINT	<b>int_rows_updated</b> - 更新された内部行数
INT_COMMITS	BIGINT	<b>int_commits</b> - 内部コミット数
INT_ROLLBACKS	BIGINT	<b>int_rollback</b> - 内部ロールバック数
INT_DEADLOCK_ROLLBACKS	BIGINT	<b>int_deadlock_rollback</b> - デッドロックによる内部ロールバック数
ROWS_DELETED	BIGINT	<b>rows_deleted</b> - 削除行数
ROWS_INSERTED	BIGINT	<b>rows_inserted</b> - 挿入行数
ROWS_UPDATED	BIGINT	<b>rows_updated</b> - 更新行数
ROWS_SELECTED	BIGINT	<b>rows_selected</b> - 選択行数
BINDS_PRECOMPILES	BIGINT	<b>binds_precompiles</b> - 試行されたバインド/プリコンパイル
TOTAL_CONS	BIGINT	<b>total_cons</b> - データベース・アクティブ化以降の接続
APPLS_CUR_CONS	BIGINT	<b>appls_cur_cons</b> - 現在接続されているアプリケーション

表 348. SNAPSHOT\_DATABASE 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
APPLS_IN_DB2	BIGINT	<b>appls_in_db2</b> - データベースで現在実行中のアプリケーション
SEC_LOGS_ALLOCATED	BIGINT	<b>sec_logs_allocated</b> - 現在割り振られている 2 次ログ
DB_STATUS	BIGINT	<b>db_status</b> - データベース状況
LOCK_TIMEOUTS	BIGINT	<b>lock_timeouts</b> - ロック・タイムアウト数
CONNECTIONS_TOP	BIGINT	<b>connections_top</b> - 同時接続の最大数
DB_HEAP_TOP	BIGINT	<b>db_heap_top</b> - 割り振られた最大データベース・ヒープ
INT_ROWS_INSERTED	BIGINT	<b>int_rows_inserted</b> - 挿入された内部行数
LOG_READS	BIGINT	<b>log_reads</b> - 読み取られたログ・ページの数
LOG_WRITES	BIGINT	<b>log_writes</b> - 書き込まれたログ・ページの数
PKG_CACHE_LOOKUPS	BIGINT	<b>pkg_cache_lookups</b> - パッケージ・キャッシュ参照
PKG_CACHE_INSERTS	BIGINT	<b>pkg_cache_inserts</b> - パッケージ・キャッシュ挿入
CAT_CACHE_LOOKUPS	BIGINT	<b>cat_cache_lookups</b> - カタログ・キャッシュ参照数
CAT_CACHE_INSERTS	BIGINT	<b>cat_cache_inserts</b> - カタログ・キャッシュ挿入数
CAT_CACHE_OVERFLOWS	BIGINT	<b>cat_cache_overflows</b> - カタログ・キャッシュ・オーバーフロー数
CAT_CACHE_HEAP_FULL	BIGINT	<b>cat_cache_heap_full</b> - カタログ・キャッシュ・ヒープ・フル
CATALOG_PARTITION	SMALLINT	<b>catalog_node</b> - カタログ・ノード番号
TOTAL_SEC_CONS	BIGINT	<b>total_sec_cons</b> - 2 次接続
NUM_ASSOC_AGENTS	BIGINT	<b>num_assoc_agents</b> - 関連したエージェント数
AGENTS_TOP	BIGINT	<b>agents_top</b> - 作成されたエージェントの数
COORD_AGENTS_TOP	BIGINT	<b>coord_agents_top</b> - コーディネーター・エージェント最大数
PREFETCH_WAIT_TIME	BIGINT	<b>prefetch_wait_time</b> - プリフェッチ待ち時間
APPL_SECTION_LOOKUPS	BIGINT	<b>appl_section_lookups</b> - セクションの参照回数

表 348. SNAPSHOT\_DATABASE 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
APPL_SECTION_INSERTS	BIGINT	<b>appl_section_inserts</b> - セクション挿入数
TOTAL_HASH_JOINS	BIGINT	<b>total_hash_joins</b> - ハッシュ結合の合計
TOTAL_HASH_LOOPS	BIGINT	<b>total_hash_loops</b> - ハッシュ・ループの合計
HASH_JOIN_OVERFLOWS	BIGINT	<b>hash_join_overflows</b> - ハッシュ結合のオーバーフロー
HASH_JOIN_SMALL_OVERFLOWS	BIGINT	<b>hash_join_small_overflows</b> - ハッシュ結合の短精度オーバーフロー
PKG_CACHE_NUM_OVERFLOWS	BIGINT	<b>pkg_cache_num_overflows</b> - パッケージ・キャッシュ・オーバーフロー数
PKG_CACHE_SIZE_TOP	BIGINT	<b>pkg_cache_size_top</b> - パッケージ・キャッシュ最高水準点
DB_CONN_TIME	TIMESTAMP	<b>db_conn_time</b> - データベース・アクティビ化タイム・スタンプ
SQLM_ELM_LAST_RESET	TIMESTAMP	<b>last_reset</b> - 最後のリセット・タイム・スタンプ
SQLM_ELM_LAST_BACKUP	TIMESTAMP	<b>last_backup</b> - 最終バックアップ・タイム・スタンプ
APPL_CON_TIME	TIMESTAMP	<b>appl_con_time</b> - 接続要求開始タイム・スタンプ
ELAPSED_EXEC_TIME_S	BIGINT	<b>elapsed_exec_time</b> - ステートメント実行経過時間 (秒単位)*
ELAPSED_EXEC_TIME_MS	BIGINT	<b>elapsed_exec_time</b> - ステートメント実行経過時間 (小数のマイクロ秒単位)*
DB_LOCATION	INTEGER	<b>db_location</b> - データベース・ロケーション
SERVER_PLATFORM	INTEGER	<b>server_platform</b> - サーバーのオペレーティング・システム
APPL_ID_OLDEST_XACT	BIGINT	<b>appl_id_oldest_xact</b> - 最も古いトランザクションを使用するアプリケーション
CATALOG_PARTITION_NAME	VARCHAR(128)	<b>catalog_node_name</b> - カタログ・ノード・ネットワーク名
INPUT_DB_ALIAS	VARCHAR(128)	<b>input_db_alias</b> - 入力データベース別名
DB_NAME	VARCHAR(128)	<b>db_name</b> - データベース名
DB_PATH	VARCHAR(1024)	<b>db_path</b> - データベース・パス

表 348. SNAPSHOT\_DATABASE 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
<p>* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する <code>_S</code> で終わっている列で報告されている整数秒と、このモニター・エレメントに関する <code>_MS</code> で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません: <math>(\text{monitor-element-name}_S \times 1,000,000 + \text{monitor-element-name}_{MS}) \div 1,000,000</math>. 例えば、<math>(\text{ELAPSED\_EXEC\_TIME}_S \times 1,000,000 + \text{ELAPSED\_EXEC\_TIME}_{MS}) \div 1,000,000</math>.</p>		

## SNAPSHOT\_DBM

DB2 データベース・マネージャーのスナップショットからの情報を戻します。

注: この表関数は使用すべきではなく、891 ページの『SNAP\_GET\_DBM 表関数』に置き換えられました。

▶—SNAPSHOT\_DBM—(—member—)————▶

スキーマは SYSPROC です。

### 表関数パラメーター

*member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

NULL 値が指定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアード・プロシージャによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

### 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 349. SNAPSHOT\_DBM 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
SORT_HEAP_ALLOCATED	BIGINT	sort_heap_allocated - 割り振られたソート・ヒープの合計
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - ポストしきい値ソート
PIPED_SORTS_REQUESTED	BIGINT	pipedsorts_requested - 要求されたパイプ・ソート数
PIPED_SORTS_ACCEPTED	BIGINT	pipedsorts_accepted - 受け入れられたパイプ・ソート
REM_CONS_IN	BIGINT	rem_cons_in - データベース・マネージャーへのリモート接続
REM_CONS_IN_EXEC	BIGINT	rem_cons_in_exec - データベース・マネージャーで実行中のリモート接続 : モニター・エレメント
LOCAL_CONS	BIGINT	local_cons - ローカル接続
LOCAL_CONS_IN_EXEC	BIGINT	local_cons_in_exec - データベース・マネージャーで実行中のローカル接続 : モニター・エレメント
CON_LOCAL_DBASES	BIGINT	con_local_dbases - 現行接続を使用したローカル・データベース
AGENTS_REGISTERED	BIGINT	agents_registered - 登録済みエージェント
AGENTS_WAITING_ON_TOKEN	BIGINT	agents_waiting_on_token - トークン待ちエージェント
DB2_STATUS	BIGINT	db_status - データベース状況
AGENTS_REGISTERED_TOP	BIGINT	agents_registered_top - エージェント最大登録数
AGENTS_WAITING_TOP	BIGINT	agents_waiting_top - エージェント最大待機数
COMM_PRIVATE_MEM	BIGINT	comm_private_mem - コミット済み専用メモリー
IDLE_AGENTS	BIGINT	idle_agents - アイドル・エージェント数
AGENTS_FROM_POOL	BIGINT	agents_from_pool - プールから割り当てられたエージェント
AGENTS_CREATED_EMPTY_POOL	BIGINT	agents_created_empty_pool - エージェント・プールが空のために作成されたエージェント
COORD_AGENTS_TOP	BIGINT	coord_agents_top - コーディネーター・エージェント最大数
MAX_AGENT_OVERFLOW	BIGINT	max_agent_overflows - 最大エージェント・オーバーフロー回数

表 349. SNAPSHOT\_DBM 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
AGENTS_STOLEN	BIGINT	<b>agents_stolen</b> - スチールされたエージェント
GW_TOTAL_CONS	BIGINT	<b>gw_total_cons</b> - DB2 Connect の接続試行合計回数
GW_CUR_CONS	BIGINT	<b>gw_cur_cons</b> - DB2 Connect の現在の接続数
GW_CONS_WAIT_HOST	BIGINT	<b>gw_cons_wait_host</b> - ホストの応答を待機している接続の数
GW_CONS_WAIT_CLIENT	BIGINT	<b>gw_cons_wait_client</b> - クライアントの要求送信を待機している接続の数
POST_THRESHOLD_HASH_JOINS	BIGINT	<b>post_threshold_hash_joins</b> - ハッシュ結合のしきい値
INACTIVE_GW_AGENTS	BIGINT	<b>idle_agents</b> - アイドル・エージェント数
NUM_GW_CONN_SWITCHES	BIGINT	<b>num_gw_conn_switches</b> - 接続切り替え回数
DB2START_TIME	TIMESTAMP	<b>db2start_time</b> - データベース・マネージャー開始タイム・スタンプ
LAST_RESET	TIMESTAMP	<b>last_reset</b> - 最後のリセット・タイム・スタンプ

## SNAPSHOT\_DYN\_SQL

動的 SQL スナップショットからの情報を戻します。この関数は SQLCACHE\_SNAPSHOT 関数に代わるものですが、互換性の理由から、SQLCACHE\_SNAPSHOT も依然として使用可能です。

注: この表関数は使用すべきではなく、900 ページの『SNAP\_GET\_DYN\_SQL 表関数』に置き換えられました。

▶▶—SNAPSHOT\_DYN\_SQL—(—dbname—, —dbpartitionnum—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

*dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。

"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *dbpartitionnum*

有効なデータベース・パーティション番号を指定する、タイプ INTEGER の入力引数。現行のデータベース・パーティションに -1、またはすべてのアクティブなデータベース・パーティションに -2 を指定します。アクティブなデータベース・パーティションとは、アプリケーションからデータベースに接続して、これを使用することが可能なパーティションのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアド・プロシージャによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 350. SNAPSHOT\_DYN\_SQL 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
NUM_EXECUTIONS	BIGINT	num_executions - ステートメント実行回数
NUM_COMPILATIONS	BIGINT	num_compilations - ステートメント・コンパイル数
PREP_TIME_WORST	BIGINT	prep_time_worst - ステートメント最長準備時間
PREP_TIME_BEST	BIGINT	prep_time_best - ステートメント最短準備時間
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 削除された内部行数
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 挿入された内部行数



表 350. *SNAPSHOT\_DYN\_SQL* 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
INT_ROWS_UPDATED	BIGINT	<b>int_rows_updated</b> - 更新された内部行数
STMT_SORTS	BIGINT	<b>stmt_sorts</b> - ステートメント・ソート回数
TOTAL_EXEC_TIME	BIGINT	<b>total_exec_time</b> - ステートメント実行の経過時間
TOTAL_SYS_CPU_TIME	BIGINT	<b>total_sys_cpu_time</b> - ステートメントのシステム CPU の合計
TOTAL_USR_CPU_TIME	BIGINT	<b>total_usr_cpu_time</b> - ステートメントのユーザー CPU の合計
STMT_TEXT	CLOB(16M) <sup>1</sup>	<b>stmt_text</b> - SQL 動的ステートメント・テキスト

<sup>1</sup> STMT\_TEXT は CLOB(16M) として定義されていますが、これは単に将来の拡張に備えるために過ぎません。ステートメント・テキストの実際の出力は 64K で切り捨てられます。

## SNAPSHOT\_FCM

SNAPSHOT\_FCM 関数は、高速コミュニケーション・マネージャー (FCM) に関するデータベース・マネージャー・レベル情報を戻します。

注: この表関数は使用すべきではなく、904 ページの『SNAPFCM 管理ビューおよび SNAP\_GET\_FCM 表関数 - fcm 論理データ・グループ・スナップショット情報の検索』に置き換えられました。

▶▶—SNAPSHOT\_FCM—(—member—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限

- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 351. SNAPSHOT\_FCM 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
BUFF_FREE	BIGINT	buff_free - 現在空いている FCM バッファ
BUFF_FREE_BOTTOM	BIGINT	buff_free_bottom - 空き FCM バッファの最小値
MA_FREE	BIGINT	ma_free モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
MA_FREE_BOTTOM	BIGINT	ma_free_bottom モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
CE_FREE	BIGINT	ce_free monitor モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
CE_FREE_BOTTOM	BIGINT	ce_free_bottom モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
RB_FREE	BIGINT	rb_free monitor モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。

表 351. *SNAPSHOT\_FCM* 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
RB_FREE_BOTTOM	BIGINT	<b>rb_free_bottom</b> モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
PARTITION_NUMBER	SMALLINT	<b>node_number</b> - ノード番号

## SNAPSHOT\_FCMNODE

データベース・マネージャーの高速コミュニケーション・マネージャーのスナップショットから情報を戻します。

注: この表関数は使用すべきではなく、908 ページの『SNAPFCM\_PART 管理ビューおよび SNAP\_GET\_FCM\_PART 表関数 - fcm\_node 論理データ・グループ・スナップショット情報の検索』に置き換えられました。

▶▶—SNAPSHOT\_FCMNODE—(—member—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

NULL 値が指定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアード・プロシージャーによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

### 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

### デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 352. *SNAPSHOT\_FCMNODE* 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
CONNECTION_STATUS	BIGINT	connection_status - 接続状況
TOTAL_BUFFERS_SENT	BIGINT	total_buffers_sent - 送信された FCM バッファの合計
TOTAL_BUFFERS_RCVD	BIGINT	total_buffers_rcvd - 受信された FCM バッファの合計
PARTITION_NUMBER	SMALLINT	node_number - ノード番号

## SNAPSHOT\_FILEW

SNAPSHOT\_FILEW プロシージャはシステム・スナップショット・データを、インスタンス・ディレクトリーの tmp サブディレクトリーにあるファイルに書き込みます。

注: このプロシージャは使用すべきではなく、964 ページの『SNAP\_WRITE\_FILE プロシージャ』によって置き換えられました。

▶▶—SNAPSHOT\_FILEW—(—requestType—,—dbname—,—dbpartitionnum—)————▶▶

スキーマは SYSPROC です。

SNAPSHOT\_FILEW プロシージャを実行するには、ユーザーに SYSADM、SYSCTRL、または SYSMANT 権限が必要です。保存されたスナップショットは、スナップショット関数への入力として NULL 値を渡すことにより、SYSADM、SYSCTRL、または SYSMANT 権限のないユーザーでも読み取れます。

### プロシージャ・パラメーター

#### *requestType*

有効なスナップショット要求タイプ (sqlmon.h で定義されている) を指定する、タイプ SMALLINT の入力引数。

#### *dbname*

このプロシージャを呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### *dbpartitionnum*

有効なデータベース・パーティション番号を指定する、タイプ SMALLINT の入力引数。現行のデータベース・パーティションに -1、またはすべてのアクティブなデータベース・パーティションに -2 を指定します。アクティブなデータ

ベース・パーティションとは、アプリケーションからデータベースに接続して、これを使用することが可能なパーティションのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このプロシージャを実行するには、以下のいずれかの権限が必要です。

- プロシージャでの EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

例: 1 の要求タイプ (SQLMA\_DB2 に相当) を指定し、現在接続されているデータベースおよび現行のデータベース・パーティションをデフォルトにすることで、データベース・マネージャー情報のスナップショットをとりま

```
CALL SNAPSHOT_FILEW (1, CAST (NULL AS VARCHAR(128)), CAST (NULL AS SMALLINT))
```

この場合、スナップショット・データは、UNIX オペレーティング・システムでは、インスタンス・ディレクトリーの /tmp/SQLMA\_DB2.dat に、Windows オペレーティング・システムでは、インスタンス・ディレクトリーの %tmp%SQLMA\_DB2.dat に書き込まれます。

## SNAPSHOT\_LOCK

ロック・スナップショットからの情報を戻します。

注: この表関数は非推奨になり、以下のものに置き換えられました。すなわち、MON\_GET\_APPL\_LOCKWAIT 表関数、MON\_GET\_LOCKS 表関数、および MON\_FORMAT\_LOCK\_NAME 表関数。

▶—SNAPSHOT\_LOCK—(—dbname—, —member—)—————▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### dbname

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、LIST DATABASE DIRECTORY コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引

数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアード・プロシージャーによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 353. SNAPSHOT\_LOCK 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
TABLE_FILE_ID	BIGINT	table_file_id - 表ファイル ID
LOCK_OBJECT_TYPE	BIGINT	lock_object_type - 待機中のロック対象タイプ
LOCK_MODE	BIGINT	lock_mode - ロック・モード
LOCK_STATUS	BIGINT	lock_status - ロック状況
LOCK_OBJECT_NAME	BIGINT	lock_object_name - ロック対象名
PARTITION_NUMBER	SMALLINT	node_number - ノード番号
LOCK_ESCALATION	SMALLINT	lock_escalation - ロック・エスカレーション
TABLE_NAME	VARCHAR(128)	table_name - 表名
TABLE_SCHEMA	VARCHAR(128)	table_schema - 表スキーマ名

表 353. `SNAPSHOT_LOCK` 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
<code>TABLESPACE_NAME</code>	<code>VARCHAR(128)</code>	<code>tablespace_name</code> - 表スペース名

## SNAPSHOT\_LOCKWAIT

アプリケーション・スナップショットからロック待機情報を戻します。

注: この表関数は非推奨になり、以下のものに置き換えられました。すなわち、`MON_GET_APPL_LOCKWAIT` 表関数、`MON_GET_LOCKS` 表関数、および `MON_FORMAT_LOCK_NAME` 表関数。

▶▶—`SNAPSHOT_LOCKWAIT`—(`—dbname—`, `—member—`)————▶▶

スキーマは `SYSPROC` です。

### 表関数パラメーター

#### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ `VARCHAR(255)` の入力引数。

"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、`LIST DATABASE DIRECTORY` コマンドで確認できます。データベース・インスタンス下のすべてのデータベースからスナップショットを取る場合は、`NULL` 値を指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ `INTEGER` の入力引数。現在のデータベース・メンバーの場合は `-1`、すべてのアクティブなデータベース・メンバーの場合は `-2` を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

`NULL` 値を指定すると、`-1` が暗黙的に設定されます。

どちらのパラメーターも `NULL` に設定された場合は、対応するスナップショット API 要求タイプの `SNAPSHOT_FILEW` ストアード・プロシージャーによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

### 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する `EXECUTE` 特権
- `DATAACCESS` 権限
- `DBADM` 権限
- `SQLADM` 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 354. *SNAPSHOT\_LOCKWAIT* 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	<code>snapshot_timestamp</code> - スナップショットのタイム・スタンプ
AGENT_ID	BIGINT	<code>agent_id</code> - アプリケーション・ハンドル (エージェント ID)
SUBSECTION_NUMBER	BIGINT	<code>ss_number</code> - サブセクション番号
LOCK_MODE	BIGINT	<code>lock_mode</code> - ロック・モード
LOCK_OBJECT_TYPE	BIGINT	<code>lock_object_type</code> - 待機中のロック対象タイプ
AGENT_ID_HOLDING_LK	BIGINT	<code>agent_id_holding_lock</code> - ロックを保持しているエージェント ID
LOCK_WAIT_START_TIME	TIMESTAMP	<code>lock_wait_start_time</code> - ロック待機開始タイム・スタンプ
LOCK_MODE_REQUESTED	BIGINT	<code>lock_mode_requested</code> - 要求されているロック・モード
PARTITION_NUMBER	SMALLINT	<code>node_number</code> - ノード番号
LOCK_ESCALATION	SMALLINT	<code>lock_escalation</code> - ロック・エスカレーション
TABLE_NAME	VARCHAR(128)	<code>table_name</code> - 表名
TABLE_SCHEMA	VARCHAR(128)	<code>table_schema</code> - 表スキーマ名
TABLESPACE_NAME	VARCHAR(128)	<code>tablespace_name</code> - 表スペース名
APPL_ID_HOLDING_LK	VARCHAR(128)	<code>appl_id_holding_lk</code> - ロックを保持しているアプリケーション ID

## SNAPSHOT QUIESCERS

SNAPSHOT QUIESCERS 関数は、静止プログラムに関する情報を表スペース・スナップショットから戻します。

注: この表関数は使用すべきではなく、946 ページの『SNAPTbsp QUIESCER 管理ビューおよび SNAP\_GET\_TBSP QUIESCER 表関数 - quiescer 表スペース・スナップショット情報の検索』に置き換えられました。



スキーマは SYSPROC です。

## 表関数パラメーター

### dbname

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 355. SNAPSHOT QUIESCERS 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
TABLESPACE_NAME	VARCHAR(128)	tablespace_name - 表スペース名
QUIESCER_TBS_ID	BIGINT	quiescer_ts_id - 静止プログラム表スペース ID
QUIESCER_OBJ_ID	BIGINT	quiescer_obj_id - 静止プログラム・オブジェクト ID

表 355. *SNAPSHOT\_QUIESCERS* 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
QUIESCER_AUTH_ID	BIGINT	<b>quiescer_auth_id</b> - 静止プログラム・ユーザー許可 ID
QUIESCER_AGENT_ID	BIGINT	<b>quiescer_agent_id</b> - 静止プログラム・エージェント ID
QUIESCER_STATE	BIGINT	<b>quiescer_state</b> - 静止プログラムの状態

## SNAPSHOT\_RANGES

*SNAPSHOT\_RANGES* 関数は、範囲スナップショットから情報を戻します。

注: この表関数は使用すべきではなく、951 ページの『*SNAPTbsp\_RANGE* 管理ビューおよび *SNAP\_GET\_Tbsp\_RANGE* 表関数 - 範囲スナップショット情報の検索』に置き換えられました。

▶—*SNAPSHOT\_RANGES*—(—*dbname*—, —*member*—)————▶

スキーマは *SYSPROC* です。

### 表関数パラメーター

#### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ *VARCHAR(255)* の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、*NULL* 値を指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ *INTEGER* の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

*NULL* 値を指定すると、-1 が暗黙的に設定されます。

### 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する *EXECUTE* 特権
- *DATAACCESS* 権限
- *DBADM* 権限
- *SQLADM* 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 356. SNAPSHOT\_RANGES 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
TABLESPACE_ID	BIGINT	tablespace_id - 表スペース ID
TABLESPACE_NAME	VARCHAR(128)	tablespace_name - 表スペース名
RANGE_NUMBER	BIGINT	range_number - 範囲番号
RANGE_STRIPE_SET_NUMBER	BIGINT	range_stripe_set_number - ストライプ・セット番号
RANGE_OFFSET	BIGINT	range_offset - 範囲オフセット
RANGE_MAX_PAGE	BIGINT	range_max_page_number - 範囲内の最大ページ
RANGE_MAX_EXTENT	BIGINT	range_max_extent - 範囲内の最大エクステント
RANGE_START_STRIPE	BIGINT	range_start_stripe - 開始ストライプ
RANGE_END_STRIPE	BIGINT	range_end_stripe - 終了ストライプ
RANGE_ADJUSTMENT	BIGINT	range_adjustment - 範囲調整
RANGE_NUM_CONTAINER	BIGINT	range_num_containers - 範囲内コンテナの数
RANGE_CONTAINER_ID	BIGINT	range_container_id - 範囲コンテナ

## SNAPSHOT\_STATEMENT

アプリケーション・スナップショットからステートメントに関する情報を戻します。

注: この表関数は使用すべきではなく、911 ページの『SNAPSTMT 管理ビューおよび SNAP\_GET\_STMT 表関数 - ステートメント・スナップショット情報の検索』に置き換えられました。

▶▶—SNAPSHOT\_STATEMENT—(—dbname—, —member—)————▶▶

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。データベース・インスタンス下のすべてのデータベースからスナップショットを取る場合は、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアード・プロシージャーによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 357. SNAPSHOT\_STATEMENT 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
AGENT_ID	BIGINT	agent_id - アプリケーション・ハンドル (エージェント ID)
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
NUM_AGENTS	BIGINT	num_agents - ステートメントで作動しているエージェントの数

表 357. SNAPSHOT\_STATEMENT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エリメント
AGENTS_TOP	BIGINT	<b>agents_top</b> - 作成されたエージェントの数
STMT_TYPE	BIGINT	<b>stmt_type</b> - ステートメント・タイプ
STMT_OPERATION	BIGINT	<b>stmt_operation/operation</b> - ステートメント操作
SECTION_NUMBER	BIGINT	<b>section_number</b> - セクション番号
QUERY_COST_ESTIMATE	BIGINT	<b>query_cost_estimate</b> - 照会コストの見積もり
QUERY_CARD_ESTIMATE	BIGINT	<b>query_card_estimate</b> - 照会行数の見積もり
DEGREE_PARALLELISM	BIGINT	<b>degree_parallelism</b> - 並列処理の度合い
STMT_SORTS	BIGINT	<b>stmt_sorts</b> - ステートメント・ソート回数
TOTAL_SORT_TIME	BIGINT	<b>total_sort_time</b> - ソート時間合計
SORT_OVERFLOW	BIGINT	<b>sort_overflows</b> - ソート・オーバーフロー
INT_ROWS_DELETED	BIGINT	<b>int_rows_deleted</b> - 削除された内部行数
INT_ROWS_UPDATED	BIGINT	<b>int_rows_updated</b> - 更新された内部行数
INT_ROWS_INSERTED	BIGINT	<b>int_rows_inserted</b> - 挿入された内部行数
FETCH_COUNT	BIGINT	<b>fetch_count</b> - 成功した取り出しの数
STMT_START	TIMESTAMP	<b>stmt_start</b> - ステートメント操作開始タイム・スタンプ
STMT_STOP	TIMESTAMP	<b>stmt_stop</b> - ステートメント操作停止タイム・スタンプ
STMT_USR_CPU_TIME_S	BIGINT	<b>stmt_usr_cpu_time</b> - ステートメントに使用されたユーザー CPU 時間 (秒単位)*
STMT_USR_CPU_TIME_MS	BIGINT	<b>stmt_usr_cpu_time</b> - ステートメントに使用されたユーザー CPU 時間 (小数部、マイクロ秒単位)*
STMT_SYS_CPU_TIME_S	BIGINT	<b>stmt_sys_cpu_time</b> - ステートメントが使用したシステム CPU 時間 (秒単位)*
STMT_SYS_CPU_TIME_MS	BIGINT	<b>stmt_sys_cpu_time</b> - ステートメントが使用したシステム CPU 時間 (小数部、マイクロ秒単位)*

表 357. SNAPSHOT\_STATEMENT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
STMT_ELAPSED_TIME_S	BIGINT	<b>stmt_elapsed_time</b> - 最新のステートメント経過時間 (秒単位)*
STMT_ELAPSED_TIME_MS	BIGINT	<b>stmt_elapsed_time</b> - 最新のステートメント経過時間 (小数部、マイクロ秒単位)*
BLOCKING_CURSOR	SMALLINT	<b>blocking_cursor</b> - ブロック・カーソル
STMT_PARTITION_NUMBER	SMALLINT	<b>stmt_node_number</b> - ステートメント・ノード
CURSOR_NAME	VARCHAR(128)	<b>cursor_name</b> - カーソル名
CREATOR	VARCHAR(128)	<b>creator</b> - アプリケーション作成者
PACKAGE_NAME	VARCHAR(128)	<b>package_name</b> - パッケージ名
STMT_TEXT	CLOB(16M) <sup>1</sup>	<b>stmt_text</b> - SQL 動的ステートメント・テキスト

<sup>1</sup> STMT\_TEXT は CLOB(16M) として定義されていますが、これは単に将来の拡張に備えるために過ぎません。ステートメント・テキストの実際の出力は 64K で切り捨てられます。

\* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する **\_S** で終わっている列で報告されている整数秒と、このモニター・エレメントに関する **\_MS** で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません:  $(\text{monitor-element-name}_S \times 1,000,000 + \text{monitor-element-name}_{MS}) \div 1,000,000$ . 例えば、 $(\text{ELAPSED\_EXEC\_TIME}_S \times 1,000,000 + \text{ELAPSED\_EXEC\_TIME}_{MS}) \div 1,000,000$ .

## SNAPSHOT\_SUBJECT

アプリケーション・スナップショットからアクセス・プランのサブセクションに関する情報を戻します。

注: この表関数は使用すべきではなく、919 ページの『SNAPSUBSECTION 管理ビューおよび SNAP\_GET\_SUBSECTION 表関数 - subsection 論理モニター・グループ・スナップショット情報の検索』に置き換えられました。

▶▶—SNAPSHOT\_SUBJECT—(—dbname—, —member—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

*dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
 "Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。データベース・インスタンス下のすべてのデータベースからスナップショットを取る場合は、NULL 値を指定します。

## member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアド・プロシージャによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 358. SNAPSHOT\_SUBSECT 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
STMT_TEXT	CLOB(16M) <sup>1</sup>	stmt_text - SQL 動的ステートメント・テキスト
SS_EXEC_TIME	BIGINT	ss_exec_time - サブセクション実行経過時間
TQ_TOT_SEND_SPILLS	BIGINT	tq_tot_send_spills - オーバーフローした表キュー・バッファの合計数
TQ_CUR_SEND_SPILLS	BIGINT	tq_cur_send_spills - オーバーフローした表キュー・バッファの現在数
TQ_MAX_SEND_SPILLS	BIGINT	tq_max_send_spills - 表キュー・バッファ・オーバーフローの最大数
TQ_ROWS_READ	BIGINT	tq_rows_read - 表キューから読み取られた行数

表 358. SNAPSHOT\_SUBSECT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TQ_ROWS_WRITTEN	BIGINT	<b>tq_rows_written</b> - 表キューに書き込まれた行数
ROWS_READ	BIGINT	<b>rows_read</b> - 読み取り行数
ROWS_WRITTEN	BIGINT	<b>rows_written</b> - 書き込み行数
SS_USR_CPU_TIME	BIGINT	<b>ss_usr_cpu_time</b> - サブセクションに使用されたユーザー CPU 時間
SS_SYS_CPU_TIME	BIGINT	<b>ss_sys_cpu_time</b> - サブセクションに使用されたシステム CPU 時間
SS_NUMBER	INTEGER	<b>ss_number</b> - サブセクション番号
SS_STATUS	INTEGER	<b>ss_status</b> - サブセクションの状況
SS_PARTITION_NUMBER	SMALLINT	<b>ss_node_number</b> - サブセクション・ノード番号
TQ_PARTITION_WAITED_FOR	SMALLINT	<b>tq_node_waited_for</b> - 表キュー上のノード待機
TQ_WAIT_FOR_ANY	INTEGER	<b>tq_wait_for_any</b> - 表キュー上のノード送信待機
TQ_ID_WAITING_ON	INTEGER	<b>tq_id_waiting_on</b> - ノード上の表キュー待機

<sup>1</sup> STMT\_TEXT は CLOB(16M) として定義されていますが、これは単に将来の拡張に備えるために過ぎません。ステートメント・テキストの実際の出力は 64K で切り捨てられます。

## SNAPSHOT\_SWITCHES

データベース・スナップショットのスイッチ状態に関する情報を戻します。

注: この表関数は使用すべきではなく、923 ページの『SNAPSWITCHES 管理ビューおよび SNAP\_GET\_SWITCHES 表関数 - データベース・スナップショットのスイッチ状態情報の検索』に置き換えられました。

▶▶—SNAPSHOT\_SWITCHES—(—*member*—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

*member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。



## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 359. SNAPSHOT\_SWITCHES 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
UOW_SW_STATE	SMALLINT	作業単位モニター記録スイッチの状態 (0 または 1)。
UOW_SW_TIME	TIMESTAMP	作業単位モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
STATEMENT_SW_STATE	SMALLINT	SQL ステートメント・モニター記録スイッチの状態 (0 または 1)。
STATEMENT_SW_TIME	TIMESTAMP	SQL ステートメント・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
TABLE_SW_STATE	SMALLINT	表アクティビティ・モニター記録スイッチの状態 (0 または 1)。
TABLE_SW_TIME	TIMESTAMP	表アクティビティ・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
BUFFPOOL_SW_STATE	SMALLINT	バッファ・プール・アクティビティ・モニター記録スイッチの状態 (0 または 1)。
BUFFPOOL_SW_TIME	TIMESTAMP	バッファ・プール・アクティビティ・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。

表 359. *SNAPSHOT\_SWITCHES* 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
LOCK_SW_STATE	SMALLINT	ロック・モニター記録スイッチの状態 (0 または 1)。
LOCK_SW_TIME	TIMESTAMP	ロック・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
SORT_SW_STATE	SMALLINT	ソート・モニター記録スイッチの状態 (0 または 1)。
SORT_SW_TIME	TIMESTAMP	ソート・モニター記録スイッチがオンの場合、このスイッチがオンになった日時。
PARTITION_NUMBER	SMALLINT	<b>node_number</b> - ノード番号

## SNAPSHOT\_TABLE

表スナップショットからアクティビティー情報を戻します。

注: この表関数は使用すべきではなく、1450 ページの『SNAPTAB 管理ビューおよび SNAP\_GET\_TAB 表関数 - table 論理データ・グループのスナップショット情報の検索』に置き換えられました。

▶—SNAPSHOT\_TABLE—(—dbname—, —member—)————▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアード・プロシージャによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 360. SNAPSHOT\_TABLE 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
ROWS_READ	BIGINT	rows_read - 読み取り行数
OVERFLOW_ACCESSES	BIGINT	overflow_accesses - オーバーフロー・レコードへのアクセス
TABLE_FILE_ID	BIGINT	table_file_id - 表ファイル ID
TABLE_TYPE	BIGINT	table_type - 表タイプ
PAGE_REORGS	BIGINT	page_reorgs - ページ再編成
TABLE_NAME	VARCHAR(128)	table_name - 表名
TABLE_SCHEMA	VARCHAR(128)	table_schema - 表スキーマ名

## SNAPSHOT\_TBREORG

SNAPSHOT\_TBREORG 関数は、表の再編成に関する情報を結果セットの形式で戻します。再編成された表がない場合は、0 行が戻されます。リアルタイム・スナップショット情報を入手するには、SYSADM、SYSCTRL、または SYSMANT 権限がなければなりません。

注: この表関数は使用すべきではなく、927 ページの『SNAPTAB\_REORG 管理ビューおよび SNAP\_GET\_TAB\_REORG 表関数 - 表再編成スナップショット情報の検索』に置き換えられました。

►►SNAPSHOT\_TBREORG(—dbname—, —member—)◄◄

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。  
"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアード・プロシージャーによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 361. SNAPSHOT\_TBREORG 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
TABLE_NAME	VARCHAR(128)	table_name - 表名
TABLE_SCHEMA	VARCHAR(128)	table_schema - 表スキーマ名
PAGE_REORGS	BIGINT	page_reorgs - ページ再編成
REORG_PHASE	BIGINT	reorg_phase - 表再編成フェーズ
REORG_MAX_PHASE	INTEGER	reorg_max_phase - 表再編成の最大フェーズ数

表 361. SNAPSHOT\_TBREORG 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
REORG_CURRENT_COUNTER	BIGINT	<b>reorg_current_counter</b> - 表再編成の進行状況
REORG_MAX_COUNTER	BIGINT	<b>reorg_max_counter</b> - 表再編成の合計量
REORG_TYPE	INTEGER	<b>reorg_type</b> - 表再編成の属性
REORG_STATUS	SMALLINT	<b>reorg_status</b> - 表再編成の状況
REORG_COMPLETION	INTEGER	<b>reorg_completion</b> - 表再編成完了フラグ
REORG_START	TIMESTAMP	<b>reorg_start</b> - 表再編成開始時刻
REORG_END	TIMESTAMP	<b>reorg_end</b> - 表再編成終了時刻
REORG_PHASE_START	TIMESTAMP	<b>reorg_phase_start</b> - 表再編成フェーズ開始時刻
REORG_INDEX_ID	BIGINT	<b>reorg_index_id</b> - 表の再編成に使用される索引
REORG_TBSPC_ID	BIGINT	<b>reorg_tbspc_id</b> - 表が再編成される表スペース
PARTITION_NUMBER	SMALLINT	<b>node_number</b> - ノード番号

## SNAPSHOT\_TBS

表スペース・スナップショットからアクティビティー情報を戻します。

注: この表関数は使用すべきではなく、934 ページの『SNAPTbsp 管理ビューおよび SNAP\_GET\_TBSP 表関数 - table space 論理データ・グループのスナップショット情報の検索』に置き換えられました。

▶▶—SNAPSHOT\_TBS—(—dbname—, —member—)——▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、LIST DATABASE DIRECTORY コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータ

ベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアード・プロシージャーによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 362. SNAPSHOT\_TBS 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - バッファーク・プール・データの論理読み取り
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - バッファーク・プール・データの物理読み取り
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - バッファーク・プール非同期データ読み取り
POOL_DATA_WRITES	BIGINT	pool_data_writes - バッファーク・プールへのデータの書き込み
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - バッファーク・プール非同期データ書き込み
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - バッファーク・プール索引の論理読み取り
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - バッファーク・プール索引の物理読み取り
POOL_INDEX_WRITES	BIGINT	pool_index_writes - バッファーク・プール索引の書き込み
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - バッファーク・プール非同期索引書き込み

表 362. SNAPSHOT\_TBS 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
POOL_READ_TIME	BIGINT	<b>pool_read_time</b> - バッファークール物理読み取り時間の合計
POOL_WRITE_TIME	BIGINT	<b>pool_write_time</b> - バッファークール物理書き込み時間の合計
POOL_ASYNC_READ_TIME	BIGINT	<b>pool_async_read_time</b> - バッファークール非同期読み取り時間
POOL_ASYNC_WRITE_TIME	BIGINT	<b>pool_async_write_time</b> - バッファークール非同期書き込み時間
POOL_ASYNC_DATA_READ_REQS	BIGINT	<b>pool_async_data_read_reqs</b> - バッファークール非同期読み取り要求
DIRECT_READS	BIGINT	<b>direct_reads</b> - データベースからの直接読み取り
DIRECT_WRITES	BIGINT	<b>direct_writes</b> - データベースへの直接書き込み
DIRECT_READ_REQS	BIGINT	<b>direct_read_reqs</b> - 直接読み取り要求
DIRECT_WRITE_REQS	BIGINT	<b>direct_write_reqs</b> - 直接書き込み要求
DIRECT_READ_TIME	BIGINT	<b>direct_read_time</b> - 直接読み取り時間
DIRECT_WRITE_TIME	BIGINT	<b>direct_write_time</b> - 直接書き込み時間
UNREAD_PREFETCH_PAGES	BIGINT	<b>unread_prefetch_pages</b> - 読み取り不能プリフェッチ・ページ
POOL_ASYNC_INDEX_READS	BIGINT	<b>pool_async_index_reads</b> - バッファークール非同期索引読み取り
POOL_DATA_TO_ESTORE	BIGINT	<b>pool_data_to_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
POOL_INDEX_TO_ESTORE	BIGINT	<b>pool_index_to_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されます。
POOL_INDEX_FROM_ESTORE	BIGINT	<b>pool_index_from_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されま
POOL_DATA_FROM_ESTORE	BIGINT	<b>pool_data_from_estore</b> ESTORE モニター・エレメントは廃止されています。廃止されたモニター・エレメントには NULL 値が戻されま

表 362. SNAPSHOT\_TBS 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
FILES_CLOSED	BIGINT	<b>files_closed</b> - 閉じられたデータベース・ファイル
TABLESPACE_NAME	VARCHAR(128)	<b>tablespace_name</b> - 表スペース名

## SNAPSHOT\_TBS\_CFG

SNAPSHOT\_TBS\_CFG 関数は、表スペース・スナップショットから構成情報を戻します。

注: この表関数は使用すべきではなく、942 ページの『SNAP\_GET\_TBSP\_PART 表関数』に置き換えられました。

▶▶—SNAPSHOT\_TBS\_CFG—(—dbname—, —member—)————▶▶

スキーマは SYSPROC です。

### 表関数パラメーター

#### dbname

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(255) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL 値を指定します。

#### member

有効なデータベース・メンバー番号を指定する、タイプ INTEGER の入力引数。現在のデータベース・メンバーの場合は -1、すべてのアクティブなデータベース・メンバーの場合は -2 を指定します。アクティブなデータベース・メンバーとは、アプリケーションによるデータベースへの接続およびデータベースの使用が可能なメンバーのことです。

NULL 値を指定すると、-1 が暗黙的に設定されます。

どちらのパラメーターも NULL に設定された場合は、対応するスナップショット API 要求タイプの SNAPSHOT\_FILEW ストアード・プロシージャーによって以前にファイルが作成されていない場合にのみ、スナップショットがとられます。

### 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限



## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は、以下の表を戻します。

表 363. SNAPSHOT\_TBS\_CFG 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	snapshot_timestamp - スナップショットのタイム・スタンプ
TABLESPACE_ID	BIGINT	tablespace_id - 表スペース ID
TABLESPACE_NAME	VARCHAR (128)	tablespace_name - 表スペース名
TABLESPACE_TYPE	SMALLINT	tablespace_type - 表スペース・タイプ
TABLESPACE_STATE	BIGINT	tablespace_state - 表スペースの状態
NUM_QUIESCERS	BIGINT	tablespace_num_quiescers - 静止プログラム数
STATE_CHANGE_OBJ_ID	BIGINT	tablespace_state_change_object_id - 状態変更オブジェクト ID
STATE_CHANGE_TBS_ID	BIGINT	tablespace_state_change_ts_id - 状態変更表スペース ID
MIN_RECOVERY_TIME	TIMESTAMP	tablespace_min_recovery_time - ロールフォワードの最小リカバリー時間
TBS_CONTENTS_TYPE	SMALLINT	tablespace_content_type - 表スペースのコンテンツ・タイプ
BUFFERPOOL_ID	BIGINT	tablespace_cur_pool_id - 現在使用中のバッファ・プール
NEXT_BUFFERPOOL_ID	BIGINT	tablespace_next_pool_id - 次の始動時に使用されるバッファ・プール
PAGE_SIZE	BIGINT	tablespace_page_size - 表スペースのページ・サイズ
EXTENT_SIZE	BIGINT	tablespace_extent_size - 表スペースのエクステント・サイズ
PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 表スペースのプリフェッチ・サイズ
TOTAL_PAGES	BIGINT	tablespace_total_pages - 表スペース内の合計ページ数
USABLE_PAGES	BIGINT	tablespace_usable_pages - 表スペース内の使用可能ページ数
USED_PAGES	BIGINT	tablespace_used_pages - 表スペース内の使用されているページ数
FREE_PAGES	BIGINT	tablespace_free_pages - 表スペース内のフリー・ページ数
PENDING_FREE_PAGES	BIGINT	tablespace_pending_free_pages - 表スペース内のペンディング・フリー・ページ数

表 363. `SNAPSHOT_TBS_CFG` 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
<code>HIGH_WATER_MARK</code>	BIGINT	<code>pool_watermark</code> - メモリー・プール水準点
<code>REBALANCER_MODE</code>	BIGINT	<code>tablespace_rebalancer_mode</code> - リバランサー・モード
<code>REBALANCER_EXTENTS_REMAINING</code>	BIGINT	<code>tablespace_rebalancer_extents_remaining</code> - リバランサーで処理されるエクステントの合計数
<code>REBALANCER_EXTENTS_PROCESSED</code>	BIGINT	<code>tablespace_rebalancer_extents_processed</code> - リバランサーで処理されたエクステントの数
<code>REBALANCER_PRIORITY</code>	BIGINT	<code>tablespace_rebalancer_priority</code> - 現行のリバランサー優先順位
<code>REBALANCER_START_TIME</code>	TIMESTAMP	<code>tablespace_rebalancer_start_time</code> - リバランサー開始時刻
<code>REBALANCER_RESTART_TIME</code>	TIMESTAMP	<code>tablespace_rebalancer_restart_time</code> - リバランサー再始動時刻
<code>LAST_EXTENT_MOVED</code>	BIGINT	<code>tablespace_rebalancer_last_extent_moved</code> - リバランサーによって最後に移動されたエクステント
<code>NUM_RANGES</code>	BIGINT	<code>tablespace_num_ranges</code> - 表スペース・マップ内の範囲数
<code>NUM_CONTAINERS</code>	BIGINT	<code>tablespace_num_containers</code> - 表スペース内のコンテナ数

## SNAPSTORAGE\_PATHS 管理ビューおよび SNAP\_GET\_STORAGE\_PATHS\_V97 表関数 - 自動ストレージ・パスの情報の検索

`SNAPSTORAGE_PATHS` 管理ビューおよび `SNAP_GET_STORAGE_PATHS_V97` は、特に `db_storage_group` 論理データ・グループから、データベースの自動ストレージ・パスのリストを、ストレージ・パスごとのファイル・システム情報を含めて戻します。

注: 1447 ページの『`SNAPSTORAGE_PATHS` 管理ビュー』および 1448 ページの『`SNAP_GET_STORAGE_PATHS_V97`』は使用すべきではなく、254 ページの『`ADMIN_GET_STORAGE_PATHS` 表関数 - 自動ストレージ・パス情報の取得』に置き換えられました。この関数とビューは、将来のリリースで除去される可能性があります。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 1447 ページの『`SNAPSTORAGE_PATHS` 管理ビュー』
- 1448 ページの『`SNAP_GET_STORAGE_PATHS_V97`』

## SNAPSTORAGE\_PATHS 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関する自動ストレージ・パスの情報を検索できます。

SNAPSTORAGE\_PATHS 管理ビューを SNAPDB、SNAPDETAILLOG、SNAPHADR、および SNAPDB\_MEMORY\_POOL 管理ビューと併せて使用することにより、**GET SNAPSHOT FOR DATABASE ON database-alias** CLP コマンドと同等の情報を戻します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1449 ページの表 332 を参照してください。

### 許可

- SYSMON 権限
- SNAPSTORAGE\_PATHS 管理ビューに対する SELECT または CONTROL 特権、および ADMIN\_GET\_STORAGE\_PATHS 表関数に対する EXECUTE 特権

### デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

現在接続されている単一メンバー・データベースのストレージ・パスを検索します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, SUBSTR(DB_STORAGE_PATH,1,8)
       AS DB_STORAGE_PATH, SUBSTR(HOSTNAME,1,10) AS HOSTNAME
FROM SYSIBMADM.SNAPSTORAGE_PATHS
```

以下はこの照会の出力例です。

```
DB_NAME  DB_STORAGE_PATH  HOSTNAME
-----  -
STOPATH  d:                      JESSICAE
```

1 record(s) selected.

## SNAP\_GET\_STORAGE\_PATHS\_V97

SNAP\_GET\_STORAGE\_PATHS\_V97 表関数は、SNAPSTORAGE\_PATHS 管理ビューと同様の情報を戻します。この表関数を使用すると、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_STORAGE\_PATHS\_V97 表関数を SNAP\_GET\_DB、SNAP\_GET\_DETAILLOG、SNAP\_GET\_HADR、および SNAP\_GET\_DB\_MEMORY\_POOL 表関数と併せて使用することにより、**GET SNAPSHOT FOR ALL DATABASES** CLP コマンドと同等の情報を戻します。

戻される可能性のある情報の完全なリストは、1449 ページの表 332 を参照してください。

## 構文

```
▶▶ SNAP_GET_STORAGE_PATHS_V97 ( ( dbname [ , member ] ) ) ▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、空ストリングを指定します。現在接続されているデータベースと同じインスタンス内のすべてのデータベースからのスナップショットを取得するには、NULL 値を指定します。

### *member*

有効なメンバー番号を指定する、タイプ INTEGER のオプション入力引数。現在のメンバーの場合は -1、すべてのアクティブなメンバーの集合の場合は -2 を指定します。*dbname* を NULL に設定せずに、*member* を NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはデータベースがアクティブになっているすべてのメンバーから戻されます。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_STORAGE\_PATHS\_V97 表関数は、現在接続中のデータベースおよびデータベース・メンバーのスナップショットを取得します。

## 許可

- SYSMON 権限
- SNAP\_GET\_STORAGE\_PATHS\_V97 表関数に対する EXECUTE 特権

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

すべてのアクティブ・データベースに関するストレージ・パスの情報を取り出します。

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, DB_STORAGE_PATH
FROM TABLE(SNAP_GET_STORAGE_PATHS_V97(CAST (NULL AS VARCHAR(128)), -1)) AS T
```

以下はこの照会の出力例です。

```
DB_NAME  DB_STORAGE_PATH
-----  -----
STOPATH  /home/jessicae/sdb
MYDB     /home/jessicae/mdb
```

2 record(s) selected

## 戻される情報

ファイル・システムの情報を戻すためには、BUFFERPOOL モニター・スイッチをオンにする必要があります。

表 364. SNAPSTORAGE\_PATHS 管理ビューおよび SNAP\_GET\_STORAGE\_PATHS\_V97 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
DB_NAME	VARCHAR(128)	db_name - データベース名
DB_STORAGE_PATH	VARCHAR(256)	db_storage_path - 自動ストレージ・パス
DB_STORAGE_PATH_WITH_DPE	VARCHAR(256)	db_storage_path_with_dpe - データベース・パーティション式を含むストレージ・パス・モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
DB_STORAGE_PATH_STATE	VARCHAR(16)	db_storage_path_state - ストレージ・パスの状態モニター・エレメント
FS_ID	VARCHAR(22)	fs_id - 固有のファイル・システム識別番号
FS_TOTAL_SIZE	BIGINT	fs_total_size - ファイル・システムの合計サイズ
FS_USED_SIZE	BIGINT	fs_used_size - ファイル・システム上で使用されるスペースの量
STO_PATH_FREE_SIZE	BIGINT	sto_path_free_size - 自動ストレージ・パスのフリー・スペース

## SNAPTAB 管理ビューおよび SNAP\_GET\_TAB 表関数 - table 論理データ・グループのスナップショット情報の検索

SNAPTAB 管理ビューおよび SNAP\_GET\_TAB 表関数は、table 論理データ・グループからのスナップショット情報を戻します。

**重要:** SNAPTAB 管理ビューおよび SNAP\_GET\_TAB 表関数は使用すべきではありません。これらの非推奨になったインターフェースによって返される情報を取得するには、537 ページの『MON\_GET\_BUFFERPOOL 表関数 - バッファ・プール・メトリックの取得』、693 ページの『MON\_GET\_TABLE 表関数 - 表メトリックの取得』、697 ページの『MON\_GET\_TABLESPACE 表関数 - 表スペース・メトリックの取得』、および 461 ページの『MON\_BP\_UTILIZATION 管理ビュー - バッファ・プールに関するメトリックの取得』を使用してください。

管理ビューまたは表関数のどちらを使用しているかによって、以下のいずれかのセクションを参照してください。

- 1450 ページの『SNAPTAB 管理ビュー』
- 1451 ページの『SNAP\_GET\_TAB 表関数』

## SNAPTAB 管理ビュー

この管理ビューを使用すると、現在接続されているデータベースに関する table 論理データ・グループのスナップショット情報を検索できます。

SNAPTAB\_REORG 管理ビューと併せて使用することにより、SNAPTAB 管理ビューは **GET SNAPSHOT FOR TABLES ON database-alias** CLP コマンドと同等の情報を戻します。

スキーマは SYSIBMADM です。

戻される可能性のある情報の完全なリストは、1453 ページの表 333 を参照してください。

## 許可

このビューを使用するには、以下のいずれかの許可が必要です。

- SNAPTAB 管理ビューに対する SELECT 特権
- SNAPTAB 管理ビューに対する CONTROL 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

この表関数を使用するには、以下のいずれかが必要です。

- SNAP\_GET\_TAB 表関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このビューが自動的に作成されると、SELECT 特権が PUBLIC に付与されます。

### 例

すべてのアクティブな表のスキーマと名前を取り出します。

```
SELECT SUBSTR(TABSCHEMA,1,8), SUBSTR(TABNAME,1,15) AS TABNAME, TAB_TYPE,
       DBPARTITIONNUM FROM SYSIBMADM.SNAPTAB
```

以下はこの照会の出力例です。

TABSCHEMA	TABNAME	TAB_TYPE	DBPARTITIONNUM
SYSTOOLS	HMON_ATM_INFO	USER_TABLE	0

1 record selected.

## SNAP\_GET\_TAB 表関数

SNAP\_GET\_TAB 表関数は SNAPTAB 管理ビューと同じ情報を戻しますが、特定のデータベース・メンバー、すべてのデータベース・メンバーの集約、またはすべてのデータベース・メンバーのいずれかの特定のデータベースを対象とした情報を検索することができます。

SNAP\_GET\_TAB\_REORG 表関数と併せて使用することにより、SNAP\_GET\_TAB 表関数は **GET SNAPSHOT FOR TABLES ON database-alias** CLP コマンドと同等の情報を戻します。

戻される可能性のある情報の完全なリストは、1453 ページの表 333 を参照してください。

### 構文

```
→ SNAP_GET_TAB ( ( dbname ) [ , member ] )
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *dbname*

現在接続されているデータベースと同じインスタンス内の有効なデータベース名を指定する、タイプ VARCHAR(128) の入力引数。"Indirect" または "Home" のディレクトリー項目タイプを持つデータベース名を指定します。このディレクトリー項目タイプは、**LIST DATABASE DIRECTORY** コマンドで確認できます。現在接続されているデータベースからのスナップショットを取得するには、NULL または空ストリングを指定します。

#### *member*

有効なデータベース・メンバー番号を指定する、タイプ INTEGER のオプションの入力引数。現在のメンバーの場合は -1、すべてのアクティブなメンバーの集合の場合は -2 を指定します。dbname を NULL に設定せずに、member を

NULL に設定すると、*member* は暗黙的に -1 に設定されます。この入力オプションを使用しない場合、つまり *dbname* のみを指定する場合、データはデータベースがアクティブになっているすべてのメンバーから戻されます。

*dbname* と *member* の両方を NULL に設定すると、SNAP\_WRITE\_FILE プロシージャにより作成されるファイルからのデータの読み取りが試行されます。このファイルはいつでも作成される可能性があるため、データは現行のものであるとは限らないことに注意してください。対応するスナップショット API 要求タイプを持つファイルが存在しない場合、SNAP\_GET\_TAB 表関数は、現在接続中のデータベースおよびデータベース・メンバー番号のスナップショットを取得します。

## 許可

以下のいずれかの権限が必要です。

- SNAP\_GET\_TAB 表関数に対する EXECUTE 特権
- DATAACCESS 権限

さらに、スナップショット・モニター・データにアクセスするには、以下のいずれかの権限も必要です。

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## 例

現在接続されているデータベースの集約ビューとして、アクティブな表のリストを検索します。

```
SELECT SUBSTR(TABSCHEMA,1,8) AS TABSCHEMA, SUBSTR(TABNAME,1,15) AS TABNAME,
       TAB_TYPE, DBPARTITIONNUM FROM TABLE(SNAP_GET_TAB('','-2')) AS T
```

以下はこの照会の出力例です。

TABSCHEMA	TABNAME	TAB_TYPE	DBPARTITIONNUM
SYSTOOLS	HMON_ATM_INFO	USER_TABLE	-
JESSICAE	EMPLOYEE	USER_TABLE	-

## 戻される情報

表 365. SNAPTAB 管理ビューおよび SNAP\_GET\_TAB 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
SNAPSHOT_TIMESTAMP	TIMESTAMP	スナップショットがとられた日時。
TABSCHEMA	VARCHAR(128)	table_schema - 表スキーマ名
TABNAME	VARCHAR(128)	table_name - 表名



表 365. SNAPTAB 管理ビューおよび SNAP\_GET\_TAB 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TAB_FILE_ID	BIGINT	table_file_id - 表ファイル ID
TAB_TYPE	VARCHAR(14)	table_type - 表タイプ。このインターフェースは、sqlmon.h での定義に基づくテキスト ID を戻します。これは次のいずれかです。 <ul style="list-style-type: none"> <li>• USER_TABLE</li> <li>• DROPPED_TABLE</li> <li>• TEMP_TABLE</li> <li>• CATALOG_TABLE</li> <li>• REORG_TABLE</li> </ul>
DATA_OBJECT_PAGES	BIGINT	data_object_pages - データ・オブジェクト・ページ数
INDEX_OBJECT_PAGES	BIGINT	index_object_pages - 索引オブジェクト・ページ数
LOB_OBJECT_PAGES	BIGINT	lob_object_pages - LOB オブジェクト・ページ数
LONG_OBJECT_PAGES	BIGINT	long_object_pages - 長オブジェクト・ページ数
XDA_OBJECT_PAGES	BIGINT	xda_object_pages - XDA オブジェクト・ページ数
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_WRITTEN	BIGINT	rows_written - 書き込み行数
OVERFLOW_ACCESSES	BIGINT	overflow_accesses - オーバーフロー・レコードへのアクセス
PAGE_REORGS	BIGINT	page_reorgs - ページ再編成
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
TBSP_ID	BIGINT	tablespace_id - 表スペース ID
DATA_PARTITION_ID	INTEGER	data_partition_id - データ・パーティション ID。非パーティション表では、このエレメントは NULL になります。
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント

## SQLCACHE\_SNAPSHOT

SQLCACHE\_SNAPSHOT 関数は、DB2 動的 SQL ステートメント・キャッシュのスナップショットの結果を戻します。

注: この表関数は使用すべきではなく、1422 ページの『SNAP\_GET\_DYN\_SQL\_V95 表関数 - dynsql 論理グループのスナップショット情報の検索』に置き換えられました。

▶—SQLCACHE\_SNAPSHOT—(—)————▶

スキーマは SYSFUN です。

## 許可

この関数を実行するには、以下のいずれかの権限が必要です。

- 関数に対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、この関数が自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

この関数は引数を取りません。この関数は、以下の表を戻します。

表 366. SQLCACHE\_SNAPSHOT 表関数によって戻される情報

列名	データ・タイプ	説明または対応するモニター・エレメント
NUM_EXECUTIONS	INTEGER	num_executions - ステートメント実行回数
NUM_COMPILATIONS	INTEGER	num_compilations - ステートメント・コンパイル数
PREP_TIME_WORST	INTEGER	prep_time_worst - ステートメント最長準備時間
PREP_TIME_BEST	INTEGER	prep_time_best - ステートメント最短準備時間
INT_ROWS_DELETED	INTEGER	int_rows_deleted - 削除された内部行数
INT_ROWS_INSERTED	INTEGER	int_rows_inserted - 挿入された内部行数
ROWS_READ	INTEGER	rows_read - 読み取り行数
INT_ROWS_UPDATED	INTEGER	int_rows_updated - 更新された内部行数
ROWS_WRITTEN	INTEGER	rows_written - 書き込み行数
STMT_SORTS	INTEGER	stmt_sorts - ステートメント・ソート回数
TOTAL_EXEC_TIME_S	INTEGER	total_exec_time - ステートメント実行の経過時間 (秒単位)*

表 366. SQLCACHE\_SNAPSHOT 表関数によって戻される情報 (続き)

列名	データ・タイプ	説明または対応するモニター・エレメント
TOTAL_EXEC_TIME_MS	INTEGER	total_exec_time - ステートメント実行の経過時間 (小数部、マイクロ秒単位)*
TOT_U_CPU_TIME_S	INTEGER	total_usr_cpu_time - ステートメントのユーザー CPU の合計 (秒単位)*
TOT_U_CPU_TIME_MS	INTEGER	total_usr_cpu_time - ステートメントのユーザー CPU の合計 (小数部、マイクロ秒単位)*
TOT_S_CPU_TIME_S	INTEGER	total_sys_cpu_time - ステートメントのシステム CPU の合計 (秒単位)*
TOT_S_CPU_TIME_MS	INTEGER	total_sys_cpu_time - ステートメントのシステム CPU の合計 (小数部、マイクロ秒単位)*
DB_NAME	VARCHAR(128)	db_name - データベース名
STMT_TEXT	CLOB(16M) <sup>1</sup>	stmt_text - SQL 動的ステートメント・テキスト

<sup>1</sup> STMT\_TEXT は CLOB(16M) として定義されていますが、これは単に将来の拡張に備えるために過ぎません。ステートメント・テキストの実際の出力は 64K で切り捨てられます。

\* この列の元になるモニター・エレメントの合計消費時間を計算するには、このモニター・エレメントに関する \_S で終わっている列で報告されている整数秒と、このモニター・エレメントに関する \_MS で終わっている列に報告されている小数秒を、次の式を使用して加算しなければなりません: (monitor-element-name\_S × 1,000,000 + monitor-element-name\_MS) ÷ 1,000,000. 例えば、(ELAPSED\_EXEC\_TIME\_S × 1,000,000 + ELAPSED\_EXEC\_TIME\_MS) ÷ 1,000,000.

## SYSINSTALLROUTINES

このプロシージャは使用すべきではありません。

▶—SYSINSTALLROUTINES—(—)——▶

スキーマは SYSPROC です。

### 許可

このプロシージャを実行するには、以下のいずれかの権限が必要です。

- プロシージャでの EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限

## デフォルトの PUBLIC 特権

制限のないデータベースでは、このプロシージャが自動的に作成されると、EXECUTE 特権が PUBLIC に付与されます。

## WLM\_GET\_ACTIVITY\_DETAILS - 特定のアクティビティに関する詳細情報を戻す

この関数は、そのアプリケーション・ハンドル、作業単位 ID、およびアクティビティ ID によって識別される特定のアクティビティに関する詳細情報を戻します。この情報には、アクティビティが違反したしきい値に関する詳細が含まれます。

注: この表関数は使用すべきではなく、MON\_GET\_ACTIVITY\_DETAILS 表関数 に置き換えられました。

この関数は、1 つ以上のサービス・サブクラスの基本統計を戻します。

### 構文

```
▶▶ WLM_GET_ACTIVITY_DETAILS ( ( application_handle , uow_id ,  
▶ activity_id , member ) ) ▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *application\_handle*

有効なアプリケーション・ハンドルを指定する、タイプ BIGINT の入力引数。引数が NULL の場合、行はこの関数から戻されません。引数が NULL の場合、SQL171N エラーが戻されます。

#### *uow\_id*

アプリケーション内で固有の有効な作業単位 ID を指定する、タイプ INTEGER の入力引数。引数が NULL の場合、行はこの関数から戻されません。引数が NULL の場合、SQL171N エラーが戻されます。

#### *activity\_id*

作業単位内で固有の有効なアクティビティ ID を指定する、タイプ INTEGER の入力引数。引数が NULL の場合、行はこの関数から戻されません。引数が NULL の場合、SQL171N エラーが戻されます。

#### *member*

この関数を呼び出すときに現在接続されているデータベースと同じインスタンス内の有効なメンバー番号を指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーの場合は -1、すべてのデータベース・メンバーの場合は -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権

- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

## デフォルトの PUBLIC 特権

なし

### 例

個々のアクティビティに関する詳細情報は、WLM\_GET\_ACTIVITY\_DETAILS 表関数を使用して取得できます。この表関数は、メンバーごとにアクティビティ情報を、名前と値のペアで戻します。この例は、アプリケーション・ハンドル 1、作業単位 ID 1、アクティビティ ID 5 で識別されるアクティビティのメンバーごとに、名前と値のペアの 11 個のメンバー・サブセットのみを示すことに限定しています。名前と値のペアの完全なリストについては、1560 ページの表 368 および 1562 ページの表 369 を参照してください。

```
SELECT SUBSTR(CHAR(DBPARTITIONNUM),1,4) AS PART,
       SUBSTR(NAME, 1, 20) AS NAME,
       SUBSTR(VALUE, 1, 30) AS VALUE
FROM TABLE(WLM_GET_ACTIVITY_DETAILS(1, 1, 5, -2)) AS ACTDETAIL
WHERE NAME IN ('APPLICATION_HANDLE',
              'COORD_PARTITION_NUM',
              'LOCAL_START_TIME',
              'UOW_ID',
              'ACTIVITY_ID',
              'PARENT_UOW_ID',
              'PARENT_ACTIVITY_ID',
              'ACTIVITY_TYPE',
              'NESTING_LEVEL',
              'INVOCATION_ID',
              'ROUTINE_ID')
ORDER BY PART
```

以下はこの照会の出力例です。

PART	NAME	VALUE
0	APPLICATION_HANDLE	1
0	COORD_PARTITION_NUM	0
0	LOCAL_START_TIME	2005-11-25-18.52.49.343000
0	UOW_ID	1
0	ACTIVITY_ID	5
0	PARENT_UOW_ID	1
0	PARENT_ACTIVITY_ID	3
0	ACTIVITY_TYPE	READ_DML
0	NESTING_LEVEL	0
0	INVOCATION_ID	1
0	ROUTINE_ID	0
1	APPLICATION_HANDLE	1
1	COORD_PARTITION_NUM	0
1	LOCAL_START_TIME	2005-11-25-18.52.49.598000
1	UOW_ID	1
1	ACTIVITY_ID	5
1	PARENT_UOW_ID	
1	PARENT_ACTIVITY_ID	

```

1  ACTIVITY_TYPE      READ_DML
1  NESTING_LEVEL      0
1  INVOCATION_ID      1
1  ROUTINE_ID         0

```

## 使用上の注意

ACTIVITY\_STATE が QUEUED である場合、コーディネーター・アクティビティがカタログ・メンバーに対する RPC を行ってしきい値チケットを取得したが、まだ応答を受け取っていないことを意味します。この状態が表示されることは、アクティビティが WLM によってキューに入れられていることを示すか、または短期間にわたって、アクティビティがそのチケットを取得する処理中であることを示すことがあります。アクティビティが本当にキューに入れられているかどうかについてもっと正確な実態を把握するために、どのエージェントが (WLM\_GET\_SERVICE\_CLASS\_AGENTS 表関数を使用して) アクティビティで作業しているかを判別し、カタログ・メンバーにあるこのエージェントの event\_object の値が WLM\_QUEUE であるかどうかを検出することができます。

## 戻される情報

表 367. WLM\_GET\_ACTIVITY\_DETAILS について戻される情報

列名	データ・タイプ	説明
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
NAME	VARCHAR(256)	エレメント名。考えられる値については、表 368 および 1562 ページの表 369 を参照してください。
VALUE	VARCHAR(1024)	エレメントの値。考えられる値については、表 368 および 1562 ページの表 369 を参照してください。

表 368. 戻されるエレメント

エレメント名	説明
ACTIVITY_ID	activity_id - アクティビティ ID モニター・エレメント
ACTIVITY_STATE	activity_state - アクティビティの状態モニター・エレメント
ACTIVITY_TYPE	activity_type - アクティビティ・タイプ・モニター・エレメント
APPLICATION_HANDLE	application_handle - アプリケーション・ハンドル・モニター・エレメント
COORD_PARTITION_NUM	coord_partition_num - コーディネーター・パーティション番号モニター・エレメント
DATABASE_WORK_ACTION_SET_ID	このアクティビティがデータベースに適用されている作業アクション・セットにマップされている場合、この列にはその作業アクション・セットの ID が入っています。アクティビティが、データベースに適用されている作業アクション・セットにマップされていない場合、この列には 0 が入っています。

表 368. 戻されるエレメント (続き)

エレメント名	説明
DATABASE_WORK_CLASS_ID	このアクティビティがデータベースに適用されている作業アクション・セットにマップされている場合、この列にはこのアクティビティの作業クラスの ID が入っています。アクティビティが、データベースに適用されている作業アクション・セットにマップされていない場合、この列には 0 が入っています。
EFFECTIVE_ISOLATION	effective_isolation - 有効な分離モニター・エレメント
EFFECTIVE_LOCK_TIMEOUT	effective_lock_timeout - 有効なロック・タイムアウト・モニター・エレメント
EFFECTIVE_QUERY_DEGREE	effective_query_degree - 有効な照会の度合いモニター・エレメント
ENTRY_TIME	entry_time - エントリー時間モニター・エレメント
INVOCATION_ID	invocation_id - 呼び出し ID モニター・エレメント
LAST_REFERENCE_TIME	last_reference_time - 最終参照時刻モニター・エレメント
LOCAL_START_TIME	local_start_time - ローカル開始時刻モニター・エレメント
NESTING_LEVEL	nesting_level - ネスティング・レベル・モニター・エレメント
PACKAGE_NAME	package_name - パッケージ名モニター・エレメント
PACKAGE_SCHEMA	package_schema - パッケージ・スキーマ・モニター・エレメント
PACKAGE_VERSION_ID	package_version_id - パッケージ・バージョン・モニター・エレメント
PARENT_ACTIVITY_ID	parent_activity_id - 親アクティビティ ID モニター・エレメント
PARENT_UOW_ID	アプリケーション内の固有の作業単位 ID。このアクティビティの親アクティビティが開始された元の作業単位を表します。アクティビティに親アクティビティがない場合、またはそれがリモート・メンバーにある場合は、空ストリングを戻します。
QUERY_COST_ESTIMATE	query_cost_estimate - 照会コストの見積もりモニター・エレメント
ROUTINE_ID	routine_id - ルーチン ID モニター・エレメント
ROWS_FETCHED	rows_fetched - フェッチ行数モニター・エレメント
ROWS_MODIFIED	rows_modified - 変更された行数モニター・エレメント
SECTION_NUMBER	section_number - セクション番号モニター・エレメント
SERVICE_CLASS_ID	service_class_id - サービス・クラス ID モニター・エレメント
SERVICE_CLASS_WORK_ACTION_SET_ID	このアクティビティがサービス・クラスに適用されている作業アクション・セットにマップされている場合、この列にはその作業アクション・セットの ID が入っています。アクティビティがサービス・クラスに適用されている作業アクション・セットにマップされていない場合、この列には 0 が入っています。

表 368. 戻されるエレメント (続き)

エレメント名	説明
SERVICE_CLASS_WORK_CLASS_ID	このアクティビティがサービス・クラスに適用されている作業アクション・セットにマップされている場合、この列にはこのアクティビティの作業クラスの ID が入っています。アクティビティがサービス・クラスに適用されている作業アクション・セットにマップされていない場合、この列には 0 が入っています。
STMT_PKG_CACHE_ID	stmt_pkgcache_id - ステートメント・パッケージ・キャッシュ ID モニター・エレメント
STMT_TEXT	stmt_text - SQL ステートメント・テキスト・モニター・エレメント
SYSTEM_CPU_TIME	system_cpu_time - システム CPU 時間モニター・エレメント
UOW_ID	uow_id - 作業単位 ID モニター・エレメント
USER_CPU_TIME	user_cpu_time - ユーザー CPU 時間モニター・エレメント
UTILITY_ID	utility_id - ユーティリティ ID モニター・エレメント

**重要:** WLM\_GET\_ACTIVITY\_DETAILS 表関数は、現在アクティビティに適用されているしきい値のみを示します。

以下のエレメントは、対応するしきい値がアクティビティに適用される場合にのみ戻されます。

表 369. 適用される場合に戻されるエレメント

エレメント名	説明
ACTIVITYTOTALTIME_THRESHOLD_ID	activitytotaltime_threshold_id - アクティビティ合計時間しきい値 ID モニター・エレメント
ACTIVITYTOTALTIME_THRESHOLD_VALUE	activitytotaltime_threshold_value - アクティビティ合計時間しきい値モニター・エレメント
ACTIVITYTOTALTIME_THRESHOLD_VIOLATED	activitytotaltime_threshold_violated - アクティビティ合計時間しきい値の違反モニター・エレメント
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_ID	concurrentdbcoordactivities_db_threshold_id - 並行データベース・コーディネーター・アクティビティのデータベースしきい値 ID モニター・エレメント
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_QUEUED	concurrentdbcoordactivities_db_threshold_queued - 並行データベース・コーディネーター・アクティビティのデータベースしきい値によるキュー待機モニター・エレメント
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_VALUE	concurrentdbcoordactivities_db_threshold_value - 並行データベース・コーディネーター・アクティビティのデータベースしきい値モニター・エレメント
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_VIOLATED	concurrentdbcoordactivities_db_threshold_violated - 並行データベース・コーディネーター・アクティビティのデータベースしきい値の違反モニター・エレメント



表 369. 適用される場合に戻されるエレメント (続き)

エレメント名	説明
CONCURRENTDBCOORDACTIVITIES_ SUBCLASS_THRESHOLD_ID	concurrentdbcoordactivities_subclass_threshold_id - 並行データベース・コーディネーター・アクティビティのサービス・サブクラスしきい値 ID モニター・エレメント
CONCURRENTDBCOORDACTIVITIES_ SUBCLASS_THRESHOLD_QUEUED	「Yes」は、アクティビティが CONCURRENTDBCOORDACTIVITIES_SUBCLASS しきい値 によってキューに入れられたことを示します。「No」は、アク ティビティがキューに入れられなかったことを示します。
CONCURRENTDBCOORDACTIVITIES_ SUBCLASS_THRESHOLD_VALUE	concurrentdbcoordactivities_subclass_threshold_value - 並行デー タベース・コーディネーター・アクティビティのサービス・ サブクラスしきい値 : モニター・エレメント
CONCURRENTDBCOORDACTIVITIES_ SUBCLASS_THRESHOLD_VIOLATED	「Yes」は、アクティビティが CONCURRENTDBCOORDACTIVITIES_SUBCLASS しきい値 に違反したことを示します。「No」は、アクティビティがま だしきい値に違反していないことを示します。
CONCURRENTDBCOORDACTIVITIES_ SUPERCLASS_THRESHOLD_ID	アクティビティに適用された CONCURRENTDBCOORDACTIVITIES _SUPERCLASS しきい値の ID。
CONCURRENTDBCOORDACTIVITIES_ SUPERCLASS_THRESHOLD_QUEUED	「Yes」は、アクティビティが CONCURRENTDBCOORDACTIVITIES _SUPERCLASS しきい値によってキューに入れられたことを 示します。「No」は、 アクティビティがキューに入れられなかったことを示しま す。
CONCURRENTDBCOORDACTIVITIES_ SUPERCLASS_THRESHOLD_VALUE	アクティビティに適用された CONCURRENTDBCOORDACTIVITIES _SUPERCLASS しきい値の 上限。
CONCURRENTDBCOORDACTIVITIES_ SUPERCLASS_THRESHOLD_VIOLATED	「Yes」は、アクティビティが CONCURRENTDBCOORDACTIVITIES _SUPERCLASS しきい値に違反したことを示します。「No」 は、 アクティビティがまだしきい値に違反していないことを示し ます。
CONCURRENTDBCOORDACTIVITIES_ WORK_ACTION_SET_THRESHOLD_ID	アクティビティに適用された CONCURRENTDBCOORDACTIVITIES_WORK _ACTION_SET しきい値の ID。
CONCURRENTDBCOORDACTIVITIES_ WORK_ACTION_SET_THRESHOLD_QUEUED	「Yes」は、アクティビティが CONCURRENTDBCOORDACTIVITIES _WORK_ACTION_SET しきい値によってキューに入れられた ことを示します。「No」は、 アクティビティがキューに入れられなかったことを示しま す。

表 369. 適用される場合に戻されるエレメント (続き)

エレメント名	説明
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_VALUE	アクティビティーに適用された CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET しきい値の上限。
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_VIOLATED	「Yes」は、アクティビティーが CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET しきい値に違反したことを示します。「No」は、アクティビティーがまだしきい値に違反していないことを示します。
CONCURRENTWORKLOADACTIVITIES_THRESHOLD_ID	アクティビティーに適用された CONCURRENTWORKLOADACTIVITIES しきい値の ID。
CONCURRENTWORKLOADACTIVITIES_THRESHOLD_VALUE	アクティビティーに適用された CONCURRENTWORKLOADACTIVITIES しきい値の上限。
CONCURRENTWORKLOADACTIVITIES_THRESHOLD_VIOLATED	「Yes」は、アクティビティーが CONCURRENTWORKLOADACTIVITIES しきい値に違反したことを示します。「No」は、アクティビティーがまだしきい値に違反していないことを示します。
ESTIMATEDSQLCOST_THRESHOLD_ID	estimatedsqlcost_threshold_id - 見積もり SQL コストしきい値 ID モニター・エレメント
ESTIMATEDSQLCOST_THRESHOLD_VALUE	estimatedsqlcost_threshold_value - 見積もり SQL コストしきい値モニター・エレメント
ESTIMATEDSQLCOST_THRESHOLD_VIOLATED	estimatedsqlcost_threshold_violated - 見積もり SQL コストしきい値の違反モニター・エレメント
SQLROWSRETURNED_THRESHOLD_ID	sqlrowsreturned_threshold_id - 戻される SQL 読み取り行数しきい値 ID モニター・エレメント
SQLROWSRETURNED_THRESHOLD_VALUE	sqlrowsreturned_threshold_value - 戻される SQL 読み取り行数しきい値モニター・エレメント
SQLROWSRETURNED_THRESHOLD_VIOLATED	sqlrowsreturned_threshold_violated - 戻される SQL 読み取り行数しきい値の違反モニター・エレメント
SQLTEMPSPACE_THRESHOLD_ID	sqltempstorage_threshold_id - SQL 一時スペースしきい値 ID モニター・エレメント
SQLTEMPSPACE_THRESHOLD_VALUE	sqltempstorage_threshold_value - SQL 一時スペースしきい値モニター・エレメント
SQLTEMPSPACE_THRESHOLD_VIOLATED	sqltempstorage_threshold_violated - SQL 一時スペースしきい値の違反モニター・エレメント

## WLM\_GET\_SERVICE\_CLASS\_AGENTS

### \_V97 - サービス・クラスで実行中のエージェントのリスト

WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 表関数は、指定されたサービス・クラスで実行しているか、または指定されたアプリケーションの代わりに実行している、指定されたメンバーにあるエージェント、fenced モード・プロセス (db2fmp プ

ロセス)、およびシステム・エンティティのリストを戻します。システム・エンティティは、非エージェント・スレッドおよびプロセス (ページ・クリーナーおよびプリフェッチャーなど) です。

注: この表関数は使用すべきではなく、WLM\_GET\_SERVICE\_CLASS\_AGENTS 表関数に置き換えられました。

## 構文

```
▶▶—WLM_GET_SERVICE_CLASS_AGENTS_V97—(—service_superclass_name—, —————▶▶  
▶—service_subclass_name—, —application_handle—, —member—)————▶▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *service\_superclass\_name*

現在接続されているデータベースのサービス・スーパークラスの名前を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのスーパークラスについてデータが取得されます。

### *service\_subclass\_name*

スーパークラス内の特定のサブクラスを参照する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのサブクラスについてデータが取得されます。

### *application\_handle*

エージェント情報が戻されるアプリケーション・ハンドルを指定する、タイプ BIGINT の入力引数。引数が NULL である場合、データベース内のすべてのアプリケーションについてデータが取得されます。アプリケーション・ハンドルが 0 の場合、システム・エンティティのみ戻されます。

### *member*

現在接続されているデータベースと同じインスタンス内のメンバー番号を指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

### 例 1

以下の照会は、すべてのデータベース・メンバーについてアプリケーション・ハンドル 1 に関連付けられたエージェントのリストを戻します。 **LIST APPLICATIONS** コマンドまたは **WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97** 表関数を使用して、アプリケーション・ハンドルを判別することができます。

```
SELECT SUBSTR(CHAR(APPLICATION_HANDLE),1,7) AS APPHANDLE,
       SUBSTR(CHAR(DBPARTITIONNUM),1,4) AS PART,
       SUBSTR(CHAR(AGENT_TID),1,9) AS AGENT_TID,
       SUBSTR(AGENT_TYPE,1,11) AS AGENTTYPE,
       SUBSTR(AGENT_STATE,1,10) AS AGENTSTATE,
       SUBSTR(REQUEST_TYPE,1,12) AS REQTYPE,
       SUBSTR(CHAR(UOW_ID),1,6) AS UOW_ID,
       SUBSTR(CHAR(ACTIVITY_ID),1,6) AS ACT_ID
FROM TABLE(WLM_GET_SERVICE_CLASS_AGENTS_V97(CAST(NULL AS VARCHAR(128)),
      CAST(NULL AS VARCHAR(128)), 1, -2)) AS SCDETAILS
ORDER BY APPHANDLE, PART, AGENT_TID
```

出力例を以下に示します。

APPHANDLE	PART	AGENT_TID	AGENTTYPE	AGENTSTATE	REQTYPE	UOW_ID	ACT_ID
1	0	3	COORDINATOR	ACTIVE	FETCH	1	5
1	0	4	SUBAGENT	ACTIVE	SUBSECTION:1	1	5
1	1	2	SUBAGENT	ACTIVE	SUBSECTION:2	1	5

この出力は、UOW ID 1 およびアクティビティ ID 5 のアクティビティの代わりに作動している、メンバー 0 上のコーディネーター・エージェントとサブエージェント、およびメンバー 1 上のサブエージェントを示しています。AGENTTYPE 列の値 COORDINATOR に対しては、REQTYPE 列に FETCH の値があります (これは、メインまたは初期要求タイプを示しています)。これは、要求のタイプがコーディネーター・エージェントに対するフェッチ要求であることを意味しています。

### 例 2

以下の照会は、エージェントがどのロックを待機しているかを判別します。

```
select event_object, event_type, event_state, varchar
(event_object_name, 30)
as event_object_name
from table
(wlm_get_service_class_agents_v97('',' ',cast(NULL as bigint), -1)) as t
```

出力例を以下に示します。

EVENT_OBJECT	EVENT_TYPE	EVENT_STATE
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	WAIT	IDLE
LOCK	ACQUIRE	IDLE
ROUTINE	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING

REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING

```

EVENT_OBJECT_NAME
-----
-
-
-
-
-
-
-
-
-
-
-
-
-
020005000000000000000000054
-
-
-
-
-
-
-
-
-
-
-
-
-
-

```

21 record(s) selected.

後で同じ照会を使用すると、WLM しきい値によってエージェントがキューに入れられたことが示されます。

EVENT_OBJECT	EVENT_TYPE	EVENT_STATE
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
WLM_QUEUE	WAIT	IDLE
ROUTINE	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING
REQUEST	PROCESS	EXECUTING

```

EVENT_OBJECT_NAME
-----

```

```

-
-
-
-
-
-
-
-
-
-
-
-
MYCONCDBCkoordth
-
-
-
-
-
-
-
-
-
-
-
-

```

21 record(s) selected.

### 使用上の注意

これらのパラメーターの作用については、論理積 (AND) が取られます。つまり、矛盾する入力パラメーターを指定する (例えば、サービス・スーパークラス SUP\_A とサブクラス SUB\_B を、SUB\_B が SUP\_A のサブクラスにならないように指定する) 場合、行は戻されません。

### 戻される情報

表 370. WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 によって戻される情報

列名	データ・タイプ	説明
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	service_superclass_name - サービス・スーパークラス名モニター・エレメント
SERVICE_SUBCLASS_NAME	VARCHAR (128)	service_subclass_name - サービス・サブクラス名モニター・エレメント
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル・モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
ENTITY	VARCHAR (32)	以下の値のいずれか。 <ul style="list-style-type: none"> <li>エンティティーのタイプがエージェントである場合、値は db2agent です。</li> <li>エンティティーのタイプが fenced モード・プロセスである場合、値は db2fmp (pid) です。ここで、pid は fenced モード・プロセスのプロセス ID です。</li> <li>それ以外の場合、値はシステム・エンティティーの名前です。</li> </ul>
WORKLOAD_NAME	VARCHAR (128)	workload_name - ワークロード名：モニター・エレメント

表 370. WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
WORKLOAD_OCCURRENCE_ID	INTEGER	workload_occurrence_id - ワークロード・オカレンス ID モニター・エレメント
UOW_ID	INTEGER	uow_id - 作業単位 ID モニター・エレメント
ACTIVITY_ID	INTEGER	activity_id - アクティビティ ID モニター・エレメント
PARENT_UOW_ID	INTEGER	parent_uow_id - 親作業単位 ID モニター・エレメント
PARENT_ACTIVITY_ID	INTEGER	parent_activity_id - 親アクティビティ ID モニター・エレメント
AGENT_TID	BIGINT	agent_tid - エージェント・スレッド ID モニター・エレメント
AGENT_TYPE	VARCHAR (32)	<p>エージェント・タイプ。エージェント・タイプは以下のとおりです。</p> <ul style="list-style-type: none"> <li>• COORDINATOR</li> <li>• OTHER</li> <li>• PDBSUBAGENT</li> <li>• SMPSUBAGENT</li> </ul> <p>値が COORDINATOR である場合、エージェント ID はコンソントレーター環境で変わることがあります。</p>
SMP_COORDINATOR	INTEGER	エージェントが SMP コーディネーターかどうかを示します。「はい」の場合は 1、「いいえ」の場合は 0。
AGENT_SUBTYPE	VARCHAR (32)	<p>エージェント・サブタイプ。可能なサブタイプは以下のとおりです。</p> <ul style="list-style-type: none"> <li>• DSS</li> <li>• OTHER</li> <li>• RPC</li> <li>• SMP</li> </ul>
AGENT_STATE	VARCHAR (32)	<p>エージェントが関連付けられているか、アクティブであるかを示します。可能な値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• ASSOCIATED</li> <li>• ACTIVE</li> </ul>
EVENT_TYPE	VARCHAR (32)	<p>このエージェントによって最後に処理されたイベントのタイプ。可能な値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• ACQUIRE</li> <li>• PROCESS</li> <li>• WAIT</li> </ul> <p>この列に使用できる値についての詳細は、1177 ページの表 260 を参照してください。</p>

表 370. WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
EVENT_OBJECT	VARCHAR (32)	<p>このエージェントによって最後に処理されたイベントのオブジェクト。可能な値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• COMPRESSION_DICTIONARY_BUILD</li> <li>• IMPLICIT_REBIND</li> <li>• INDEX_RECREATE</li> <li>• LOCK</li> <li>• LOCK_ESCALATION</li> <li>• QP_QUEUE</li> <li>• REMOTE_REQUEST</li> <li>• REQUEST</li> <li>• ROUTINE</li> <li>• WLM_QUEUE</li> </ul> <p>この列に使用できる値についての詳細は、1177 ページの表 260 を参照してください。</p>
EVENT_STATE	VARCHAR (32)	<p>このエージェントによって最後に処理されたイベントの状態。可能な値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• EXECUTING</li> <li>• IDLE</li> </ul> <p>この列に使用できる値についての詳細は、1177 ページの表 260 を参照してください。</p>
REQUEST_ID	VARCHAR (64)	<p>要求 ID。この値は、<i>application_handle</i> の値と組み合わせて指定される場合のみ固有です。この組み合わせを使用して、長い時間を要する 1 つの要求と複数の要求とを区別することができます。例えば、1 つの長いフェッチと複数のフェッチを区別するなどです。</p>



表 370. WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
REQUEST_TYPE	VARCHAR (32)	<p>要求のタイプ。可能な値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• コーディネーター・エージェントの場合: <ul style="list-style-type: none"> <li>- CLOSE</li> <li>- COMMIT</li> <li>- COMPILE</li> <li>- DESCRIBE</li> <li>- EXCSQLSET</li> <li>- EXECIMMD</li> <li>- EXECUTE</li> <li>- FETCH</li> <li>- INTERNAL <i>number</i> (<i>number</i> は内部定数の値)</li> <li>- OPEN</li> <li>- PREPARE</li> <li>- REBIND</li> <li>- REDISTRIBUTE</li> <li>- REORG</li> <li>- ROLLBACK</li> <li>- RUNSTATS</li> </ul> </li> <li>• AGENT_SUBTYPE が DSS または SMP であるサブエージェントの場合: <ul style="list-style-type: none"> <li>- サブセクション番号がゼロ以外の場合は、 SUBSECTION:<i>subsection number</i> の形式のサブセクション番号。そうでない場合は NULL を戻します。</li> </ul> </li> </ul>

表 370. WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
REQUEST_TYPE (続く)	VARCHAR (32)	<ul style="list-style-type: none"> <li>• AGENT_SUBTYPE が RPC であるサブエージェントの場合: <ul style="list-style-type: none"> <li>- ABP</li> <li>- CATALOG</li> <li>- INTERNAL</li> <li>- REORG</li> <li>- RUNSTATS</li> <li>- WLM</li> </ul> </li> <li>• SUBTYPE が OTHER であるサブエージェントの場合: <ul style="list-style-type: none"> <li>- ABP</li> <li>- APP_RBSVPT</li> <li>- APP_RELSVPT</li> <li>- BACKUP</li> <li>- CLOSE</li> <li>- EXTERNAL_RBSVPT</li> <li>- EVMON</li> <li>- FORCE</li> <li>- FORCE_ALL</li> <li>- INTERNAL <i>number</i> (<i>number</i> は内部定数の値)</li> <li>- INTERRUPT</li> <li>- NOOP (要求がない場合)</li> <li>- QP</li> <li>- REDISTRIBUTE</li> <li>- STMT_RBSVPT</li> <li>- STOP_USING</li> <li>- UPDATE_DBM_CFG</li> <li>- WLM</li> </ul> </li> </ul>
NESTING_LEVEL	INTEGER	nesting_level - ネスティング・レベル・モニター・エレメント
INVOCATION_ID	INTEGER	invocation_id - 呼び出し ID モニター・エレメント
ROUTINE_ID	INTEGER	routine_id - ルーチン ID モニター・エレメント
EVENT_OBJECT_NAME	VARCHAR (1024)	イベント・オブジェクト名。 EVENT_OBJECT の値が LOCK である場合、この列の値は、エージェントが待機するロックの名前です。 EVENT_OBJECT の値が WLM_QUEUE である場合、この列の値は、エージェントがキューに入れている WLM しきい値の名前です。それ以外の場合、値は NULL です。
APPLICATION_NAME	VARCHAR (128)	appl_name - アプリケーション名
APPLICATION_ID	VARCHAR (128)	appl_id - アプリケーション ID

表 370. WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
CLIENT_PID	BIGINT	client_pid - クライアント・プロセス ID
SESSION_AUTH_ID	VARCHAR (128)	session_auth_id - セッション許可 ID
REQUEST_START_TIME	TIMESTAMP	エージェントが現在処理中の要求の処理を開始した時刻
AGENT_STATE_LAST_UPDATE_TIME	TIMESTAMP	エージェントによって処理されているイベントが最後に変更された時刻。エージェントによって現在処理されているイベントは、EVENT_TYPE、EVENT_OBJECT、および EVENT_STATE 列で示されます。
EXECUTABLE_ID	VARCHAR (32) FOR BIT DATA	executable_id - 実行可能 ID モニター・エレメント

注: EVENT\_STATE、EVENT\_TYPE、EVENT\_OBJECT および EVENT\_OBJECT\_NAME 列値の可能な組み合わせを、以下の表にリストします。

表 371. EVENT\_STATE、EVENT\_TYPE、EVENT\_OBJECT および EVENT\_OBJECT\_NAME 列値の可能な組み合わせ

イベント記述	EVENT_STATE 値	EVENT_TYPE 値	EVENT_OBJECT 値	EVENT_OBJECT_NAME 値
ロックの獲得	IDLE	ACQUIRE	LOCK	ロック名
ロックのエスカレート	EXECUTING	PROCESS	LOCK_ESCALATION	NULL
要求の処理	EXECUTING	PROCESS	REQUEST	NULL
新規要求の待機	IDLE	WAIT	REQUEST	NULL
リモート・パーティションで処理される要求の待機	IDLE	WAIT	REMOTE_REQUEST	NULL
WLM threshold キューの待機	IDLE	WAIT	WLM_QUEUE	しきい値名
ルーチンの処理	EXECUTING	PROCESS	ROUTINE	NULL
索引の再作成	EXECUTING	PROCESS	INDEX_RECREATE	NULL
コンプレッション・ディクショナリーの作成	EXECUTING	PROCESS	COMP_DICT_BLD	NULL
暗黙的な再バインド	EXECUTING	PROCESS	IMPLICIT_REBIND	NULL

## WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97 - ワークロード・オカレンスのリスト

WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97 関数は、特定のメンバー上の指定されたサービス・クラスで実行しているすべてのワークロード・オカレンスのリストを戻します。ワークロード・オカレンスとは、属性がワークロ

ードの定義と一致しており、そのためにワークロードに関連付けられた、またはワークロードに割り当てられた特定のデータベース接続です。

**注:** この表関数は使用すべきではなく、  
WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES 表関数に置き換えられました。

ワークロード・オカレンスとは、属性がワークロードの定義と一致しており、そのためにワークロードに関連付けられた、またはワークロードに割り当てられた特定のデータベース接続です。

## 構文

```
▶—WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES_V97—(—service_superclass_name—,—————▶  
▶—service_subclass_name—, —member—)—————▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *service\_superclass\_name*

現在接続されているデータベースのサービス・スーパークラスの名前を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、他のパラメーターの値と一致する、データベース内のすべてのスーパークラスについてデータが取得されます。

### *service\_subclass\_name*

ワークロード・オカレンスのターゲット・サービス・サブクラス。このワークロード・オカレンスによってサブミットされる作業は、別のサブクラスにマップまたは再マップされるアクティビティーを除いて、すべてターゲット・サービス・スーパークラスの下のこのサービス・サブクラスで実行されます。

### *member*

現在接続されているデータベースと同じインスタンス内のメンバーの番号を指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーには -1、すべてのデータベース・メンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

## デフォルトの PUBLIC 特権

なし

## 例

システム全体で実行中のワークロード・オカレンスを調べる場合、管理者は *service\_superclass\_name* および *service\_subclass\_name* に NULL 値または空ストリングを指定し、*member* に -2 を指定した

WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97 関数を呼び出すことができます。

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       SUBSTR(CHAR(DBPARTITIONNUM),1,4) AS PART,
       SUBSTR(CHAR(COORD_PARTITION_NUM),1,4) AS COORDPART,
       SUBSTR(CHAR(APPLICATION_HANDLE),1,7) AS APPHNDL,
       SUBSTR(WORKLOAD_NAME,1,22) AS WORKLOAD_NAME,
       SUBSTR(CHAR(WORKLOAD_OCCURRENCE_ID),1,6) AS WLO_ID
FROM TABLE(WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES_V97
            (CAST(NULL AS VARCHAR(128)), CAST(NULL AS VARCHAR(128)), -2))
AS SCINFO
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, PART, APPHNDL,
        WORKLOAD_NAME, WLO_ID
```

システムに 4 つのデータベース・メンバーがあり、現時点で 2 つのワークロードを実行していると想定すると、上記の照会は以下のような結果を生成します。

SUPERCLASS_NAME	SUBCLASS_NAME	PART	COORDPART	...
----	-----	----	-----	----
SYSDEFAULTMAINTENAN	SYSDEFAULTSUBCLASS	0	0	...
SYSDEFAULTSYSTEMCLA	SYSDEFAULTSUBCLASS	0	0	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	0	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	0	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	1	0	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	1	0	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	2	0	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	2	0	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	3	0	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	3	0	...
...	APPHNDL WORKLOAD_NAME		WLO_ID	
...	-----		-----	
...	-	-	-	-
...	-	-	-	-
...	1	SYSDEFAULTUSERWORKLOAD	1	
...	2	SYSDEFAULTUSERWORKLOAD	2	
...	1	SYSDEFAULTUSERWORKLOAD	1	
...	2	SYSDEFAULTUSERWORKLOAD	2	
...	1	SYSDEFAULTUSERWORKLOAD	1	
...	2	SYSDEFAULTUSERWORKLOAD	2	
...	1	SYSDEFAULTUSERWORKLOAD	1	
...	2	SYSDEFAULTUSERWORKLOAD	2	

## 使用上の注意

これらのパラメーターの作用については、論理積 (AND) が取られます。つまり、矛盾する入力パラメーターを指定する (例えば、サービス・スーパークラス SUP\_A とサブクラス SUB\_B を、SUB\_B が SUP\_A のサブクラスにならないように指定する) 場合、行は戻されません。

**注:** ワークロード・オカレンスについて報告される統計 (例えば、*coord\_act\_completed\_total*) が、対応するワークロード統計と結合されると、ワークロード・オカレンスについて報告される統計が各作業単位の初めにリセットされます。

## 戻される情報

表 372. WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97 について戻される情報

列名	データ・タイプ	説明
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - サービス・スーパークラス名モニター・エレメント
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - サービス・サブクラス名モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
COORD_PARTITION_NUM	SMALLINT	coord_partition_num - コーディネーター・パーティション番号モニター・エレメント
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル・モニター・エレメント
WORKLOAD_NAME	VARCHAR(128)	workload_name - ワークロード名 : モニター・エレメント
WORKLOAD_OCCURRENCE_ID	INTEGER	workload_occurrence_id - ワークロード・オカレンス ID モニター・エレメント
UOW_ID	INTEGER	uow_id - 作業単位 ID モニター・エレメント
WORKLOAD_OCCURRENCE_STATE	VARCHAR(32)	workload_occurrence_state - ワークロード・オカレンスの状態モニター・エレメント
SYSTEM_AUTH_ID	VARCHAR(128)	system_auth_id - システム許可 ID モニター・エレメント
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - セッション許可 ID モニター・エレメント
APPLICATION_NAME	VARCHAR(128)	appl_name - アプリケーション名モニター・エレメント
CLIENT_WRKSTNNAME	VARCHAR(255)	client_wrkstnname - クライアント・ワークステーション名モニター・エレメント
CLIENT_ACCTNG	VARCHAR(255)	client_acctng - クライアント・アカウントティング・ストリング・モニター・エレメント
CLIENT_USER	VARCHAR(255)	このワークロード・オカレンスの CLIENT_USERID 特殊レジスターの現行値。
CLIENT_APPLNAME	VARCHAR(255)	client_applname - クライアント・アプリケーション名モニター・エレメント
COORD_ACT_COMPLETED_TOTAL	INTEGER	coord_act_completed_total - コーディネーター・アクティビティー完了総数モニター・エレメント
COORD_ACT_ABORTED_TOTAL	INTEGER	coord_act_aborted_total - コーディネーター・アクティビティー打ち切り総数モニター・エレメント
COORD_ACT_REJECTED_TOTAL	INTEGER	coord_act_rejected_total - コーディネーター・アクティビティー・リジェクト総数モニター・エレメント
CONCURRENT_ACT_TOP	INTEGER	concurrent_act_top - 並行アクティビティーの最上位モニター・エレメント
ADDRESS	VARCHAR(255)	address - 接続の開始元となった IP アドレス

## WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 - サービス・サブクラスの統計を戻す

WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 表関数は、1 つ以上のサービス・サブクラスの基本統計を戻します。

注: この表関数は使用すべきではなく、WLM\_GET\_SERVICE\_SUBCLASS\_STATS 表関数に置き換えられました。

戻される可能性のある情報の完全なリストは、1579 ページの表 373 を参照してください。

### 構文

```
▶▶ WLM_GET_SERVICE_SUBCLASS_STATS_V97 (—service_superclass_name—, —————▶  
▶ —service_subclass_name—, —member—) —————▶▶
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *service\_superclass\_name*

現在接続されているデータベースのサービス・スーパークラスの名前を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのスーパークラスについてデータが取得されます。

#### *service\_subclass\_name*

現在接続されているデータベースのサービス・サブクラスの名前を指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、データベース内のすべてのサブクラスについてデータが取得されます。

#### *member*

現在接続されているデータベースと同じインスタンス内の有効なメンバー番号を指定する、タイプ INTEGER の入力引数。現行のデータベース・メンバーの場合は -1、すべてのデータベース・メンバーの場合は -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

### デフォルトの PUBLIC 特権

なし

## 例

例 1: すべてのアクティビティーは実行前に DB2 サービス・クラスに対してマップされる必要があるため、サービス・クラス統計表関数を使用し、すべてのメンバー上のすべてのサービス・クラスを照会して、システムの全体的な状態をモニターできます。以下の例では、NULL 値が *service\_superclass\_name* と *service\_subclass\_name* に渡されてすべてのサービス・クラスの統計を戻し、*dbpartitionnum* には値 -2 が指定されてすべてのパーティションの統計を戻します。

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       SUBSTR(CHAR(DBPARTITIONNUM),1,4) AS PART,
       CAST(COORD_ACT_LIFETIME_AVG / 1000 AS DECIMAL(9,3))
       AS AVGLIFETIME,
       CAST(COORD_ACT_LIFETIME_STDDEV / 1000 AS DECIMAL(9,3))
       AS STDDEVLIFETIME,
       SUBSTR(CAST(LAST_RESET AS VARCHAR(30)),1,16) AS LAST_RESET
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS_V97(CAST(NULL AS VARCHAR(128)),
        CAST(NULL AS VARCHAR(128)), -2)) AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, PART
```

ステートメントは、以下の出力例で示されているように、アクティビティー存続期間の平均および標準偏差などのサービス・クラス統計を秒単位で戻します。

SUPERCLASS_NAME	SUBCLASS_NAME	PART	...
-----	-----	-----	---
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	1	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	2	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	3	...
...	AVGLIFETIME	STDDEVLIFETIME	LAST_RESET
...	-----	-----	-----
...	691.242	34.322	2006-07-24-11.44
...	644.740	22.124	2006-07-24-11.44
...	612.431	43.347	2006-07-24-11.44
...	593.451	28.329	2006-07-24-11.44

例 2: また、同じ表関数が、各パーティション上のサービス・クラスで実行しているコーディネーター・アクティビティーの平均並行性の最高値を示すこともできます。

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       SUBSTR(CHAR(DBPARTITIONNUM),1,4) AS PART,
       CONCURRENT_ACT_TOP AS ACTTOP,
       CONCURRENT_WLO_TOP AS CONNTOP
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS_V97(CAST(NULL AS VARCHAR(128)),
        CAST(NULL AS VARCHAR(128)), -2)) AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, PART
```

出力例を以下に示します。

SUPERCLASS_NAME	SUBCLASS_NAME	PART	ACTTOP	CONNTOP
-----	-----	-----	-----	-----
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	10	7
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	1	0	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	2	0	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	3	0	0

この表関数の出力内のアクティビティーの平均実行時間および回数を調べること  
で、特定のデータベースの各パーティションの負荷の概要を、的確に知ることがで



きます。この表関数によって戻される高水準ゲージが大きく変わった場合は、システムの負荷の変化を示していることがあります。

例 3: アクティビティが REMAP ACTIVITY TO アクションでしきい値を使用する場合、そのアクティビティは、その存続時間中に複数のサービス・クラスで時間を費やすことがあります。以下の例で示されているように、ACTIVITIES\_MAPPED\_IN 列と ACTIVITIES\_MAPPED\_OUT 列を調べると、一連のサービス・クラスを通過したアクティビティの数を判別できます。

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       ACTIVITIES_MAPPED_IN AS MAPPED_IN,
       ACTIVITIES_MAPPED_OUT AS MAPPED_OUT
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS_V97(CAST(NULL AS VARCHAR(128)),
        CAST(NULL AS VARCHAR(128)), -2)) AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME
```

出力例を以下に示します。

SUPERCLASS_NAME	SUBCLASS_NAME	MAPPED_IN	MAPPED_OUT
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	0
SUPERCLASS1	SYSDEFAULTSUBCLASS	0	0
SUPERCLASS1	SUBCLASS1	0	7
SUPERCLASS1	SUBCLASS2	7	0

## 使用上の注意

一部の統計は、対応するサービス・サブクラスの COLLECT AGGREGATE ACTIVITY DATA および COLLECT AGGREGATE REQUEST DATA パラメーターを NONE 以外の値に設定した場合のみ、戻されます。

WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 表関数は、サービス・サブクラスごとおよびパーティションごとに 1 行のデータを戻します。この関数は、(パーティション上の) サービス・クラス全体または (1 つ以上のサービス・クラスの) パーティション全体のデータ集約は行いません。ただし、SQL 照会を使用すると、データを集約することができます。

これらのパラメーターの作用については、論理積 (AND) が取られます。つまり、例えばスーパークラス SUPA と、SUPA のサブクラスでないサブクラス SUBB などの競合する入力パラメーターを指定した場合、行は戻されません。

## 戻される情報

表 373. WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 について戻される情報

列名	データ・タイプ	説明
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - サービス・スーパークラス名モニター・エレメント
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - サービス・サブクラス名モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号モニター・エレメント
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ・モニター・エレメント

表 373. WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 について戻される情報 (続き)

列名	データ・タイプ	説明
COORD_ACT_COMPLETED_TOTAL	BIGINT	coord_act_completed_total - コーディネーター・アクティビティー完了総数モニター・エレメント
COORD_ACT_ABORTED_TOTAL	BIGINT	coord_act_aborted_total - コーディネーター・アクティビティー打ち切り総数モニター・エレメント
COORD_ACT_REJECTED_TOTAL	BIGINT	coord_act_rejected_total - コーディネーター・アクティビティー・リジェクト総数モニター・エレメント
CONCURRENT_ACT_TOP	INTEGER	concurrent_act_top - 並行アクティビティーの最上位モニター・エレメント
COORD_ACT_LIFETIME_TOP	BIGINT	coord_act_lifetime_top - コーディネーター・アクティビティー存続時間の最上位モニター・エレメント
COORD_ACT_LIFETIME_AVG	DOUBLE	coord_act_lifetime_avg - コーディネーター・アクティビティー平均存続期間モニター・エレメント
COORD_ACT_LIFETIME_STDDEV	DOUBLE	<p>最後のリセット以降、このサービス・サブクラスに関連付けられたネスト・レベル 0 のコーディネーター・アクティビティーの存続期間の標準偏差。サービス・クラスの COLLECT AGGREGATE ACTIVITY DATA パラメーターが NONE に設定される場合、列の値は NULL です。単位はミリ秒です。</p> <p>この標準偏差はコーディネーター・アクティビティーの存続期間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。</p> <p>サービス・サブクラスの COORD_ACT_LIFETIME_STDDEV 値は、完了前にサービス・サブクラスを通過したものの別のサブクラスに再マップされるアクティビティーの影響は受けません。</p>
COORD_ACT_EXEC_TIME_AVG	DOUBLE	coord_act_exec_time_avg - コーディネーター・アクティビティー平均実行時間モニター・エレメント

表 373. WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 について戻される情報 (続き)

列名	データ・タイプ	説明
COORD_ACT_EXEC_TIME_STDDEV	DOUBLE	<p>最後のリセット以降、このサービス・サブクラスに関連付けられたネスト・レベル 0 のコーディネーター・アクティビティの実行時間の標準偏差。単位はミリ秒です。</p> <p>この標準偏差はコーディネーター・アクティビティの実行時間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。</p> <p>サービス・サブクラスの実行時間標準偏差は、完了前にサブクラスを通過したものの別のサブクラスに再マップされるアクティビティの影響は受けません。</p>
COORD_ACT_QUEUE_TIME_AVG	DOUBLE	<p>coord_act_queue_time_avg - コーディネーター・アクティビティ平均キュー時間モニター・エレメント</p>
COORD_ACT_QUEUE_TIME_STDDEV	DOUBLE	<p>最後のリセット以降、このサービス・サブクラスに関連付けられたネスト・レベル 0 のコーディネーター・アクティビティのキュー時間の標準偏差。サービス・クラスの COLLECT AGGREGATE ACTIVITY DATA パラメーターが NONE に設定される場合、列の値は NULL です。単位はミリ秒です。</p> <p>この標準偏差はコーディネーター・アクティビティのキュー時間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。</p> <p>キュー時間標準偏差のカウント対象は、アクティビティがキューに入れられたサービス・サブクラスだけです。</p>
NUM_REQUESTS_ACTIVE	BIGINT	<p>この表関数の実行時にサービス・サブクラスで実行している要求の数。</p>

表 373. WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 について戻される情報 (続き)

列名	データ・タイプ	説明
NUM_REQUESTS_TOTAL	BIGINT	<p>最後のリセット以降、このサービス・サブクラスで実行を終了する要求の数。この終了状態は、アクティビティ内の要求のメンバーシップに関係なく、任意の要求に適用されます。サービス・クラスの COLLECT AGGREGATE ACTIVITY DATA パラメーターが NONE に設定される場合、列の値は NULL です。</p> <p>サービス・サブクラスの NUM_REQUESTS_TOTAL 値は、サービス・サブクラスを通過したものの、その中で完了しない要求の影響は受けません。</p>
REQUEST_EXEC_TIME_AVG	DOUBLE	<p>request_exec_time_avg - 要求の平均実行時間モニター・エレメント</p>
REQUEST_EXEC_TIME_STDDEV	DOUBLE	<p>最後のリセット以降、このサービス・サブクラスに関連付けられた要求の実行時間の標準偏差。単位はミリ秒です。サービス・クラスの COLLECT AGGREGATE REQUEST DATA パラメーターが NONE に設定される場合、この列の値は NULL です。</p> <p>この標準偏差は要求実行時間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。</p> <p>サービス・サブクラスの実行時間標準偏差は、サブクラスを通過したものの、その中で完了しなかった要求の影響は受けません。</p>

表 373. WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 について戻される情報 (続き)

列名	データ・タイプ	説明
REQUEST_EXEC_TIME_TOTAL	BIGINT	最後のリセット以降、このサービス・サブクラスに関連付けられた要求の実行時間の合計。単位はミリ秒です。サービス・クラスの COLLECT AGGREGATE REQUEST DATA パラメーターが NONE に設定される場合、この列の値は NULL です。  この合計は要求実行時間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。  サービス・サブクラスの実行時間合計は、サブクラスを通過したものの、その中で完了しない要求の影響は受けません。
ACT_REMAPPED_IN	BIGINT	最後のリセット以降にしい値 REMAP ACTIVITY アクションによってこのサービス・サブクラスに再マップされたアクティビティー数。
ACT_REMAPPED_OUT	BIGINT	最後のリセット以降にしい値 REMAP ACTIVITY アクションによってこのサービス・サブクラスから出て再マップされたアクティビティー数。
CONCURRENT_WLO_TOP	INTEGER	concurrent_wlo_top - 並行ワークロード・オカレンスの最上位モニター・エレメント
UOW_TOTAL_TIME_TOP	BIGINT	uow_total_time_top - 作業単位合計時間トップ・モニター・エレメント

## WLM\_GET\_WORKLOAD \_OCCURRENCE\_ACTIVITIES \_V97 - アクティビティーのリストを戻す

WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97 表関数は、指定されたメンバー上の指定されたアプリケーションからサブミットされ、まだ完了していないすべてのアクティビティーのリストを戻します。

注: この表関数は使用すべきではなく、WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 表関数に置き換えられました。

戻される可能性のある情報の完全なリストは、1585 ページの表 374 を参照してください。

## 構文

```
▶—WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97—(—application_handle—,—————▶  
▶—member—)—————▶
```

スキーマは SYSPROC です。

## 表関数パラメーター

### *application\_handle*

アクティビティのリストが戻されるアプリケーション・ハンドルを指定する、タイプ BIGINT の入力引数。引数が NULL である場合、データベース内のすべてのアプリケーションについてデータが取得されます。

### *member*

現在接続されているデータベースと同じインスタンス内の有効なメンバー番号を指定する、タイプ INTEGER の入力引数。現行のメンバーには -1、すべてのメンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

## 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

## デフォルトの PUBLIC 特権

なし

## 例 1: 認識済みのアプリケーション・ハンドルで現在実行中のアクティビティ

アプリケーション・ハンドルを識別した後、このアプリケーションで現在実行中のすべてのアクティビティを検索できます。 **LIST APPLICATIONS** コマンドを使用して判別されるアプリケーション・ハンドルが 1 であるアプリケーションのアクティビティをリストすることを管理者が望んでいるという場面を、例として考えてみましょう。管理者は以下の照会を実行します。

```
SELECT SUBSTR(CHAR(COORD_PARTITION_NUM),1,5) AS COORD,  
       SUBSTR(CHAR(DBPARTITIONNUM),1,4) AS PART,  
       SUBSTR(CHAR(UOW_ID),1,5) AS UOWID,  
       SUBSTR(CHAR(ACTIVITY_ID),1,5) AS ACTID,  
       SUBSTR(CHAR(PARENT_UOW_ID),1,8) AS PARUOWID,  
       SUBSTR(CHAR(PARENT_ACTIVITY_ID),1,8) AS PARACTID,  
       ACTIVITY_TYPE AS ACTTYPE,  
       SUBSTR(CHAR(NESTING_LEVEL),1,7) AS NESTING  
FROM TABLE(WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97(1, -2)) AS WLOACTS  
ORDER BY PART, UOWID, ACTID
```

照会からの出力例は、次のようになります。

```
COORD PART UOWID ACTID PARUOWID PARACTID ACTTYPE  NESTING
-----
0      0    2    3    -        -        CALL      0
0      0    2    5    2        3        READ_DML  1
```

## 例 2: システムで現在実行中のアクティビティ

以下の照会では、EXECUTABLE\_ID で WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97 出力と MON\_GET\_PKG\_CACHE\_STMT 出力を結合し、システム上で現在実行中のすべての SQL アクティビティにステートメント・テキストを提供します。

```
SELECT t.application_handle,
       t.uow_id,
       t.activity_id,
       varchar(p.stmt_text, 256) as stmt_text
FROM table(wlm_get_workload_occurrence_activities_v97(NULL, -1)) as t,
     table(mon_get_pkg_cache_stmt(NULL, NULL, NULL, -1)) as p
WHERE t.executable_id = p.executable_id
```

出力例を以下に示します。

```
APPLICATION_HANDLE  UOW_ID    ACTIVITY_ID  STMT_TEXT
-----
1                   1          1            SELECT * FROM SYSCAT.TABLES
47                  1          36           INSERT INTO T1 VALUES(123)
```

## 戻される情報

表 374. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97 によって戻される情報

列名	データ・タイプ	説明
APPLICATION_HANDLE	BIGINT	application_handle - アプリケーション・ハンドル・モニター・エレメント
MEMBER	SMALLINT	member - データベース・メンバー・モニター・エレメント
COORD_PARTITION_NUM	SMALLINT	coord_partition_num - コーディネーター・パーティション番号モニター・エレメント
LOCAL_START_TIME	TIMESTAMP	local_start_time - ローカル開始時刻モニター・エレメント
UOW_ID	INTEGER	uow_id - 作業単位 ID モニター・エレメント
ACTIVITY_ID	INTEGER	activity_id - アクティビティ ID モニター・エレメント
PARENT_UOW_ID	INTEGER	parent_uow_id - 親作業単位 ID モニター・エレメント
PARENT_ACTIVITY_ID	INTEGER	parent_activity_id - 親アクティビティ ID モニター・エレメント
ACTIVITY_STATE	VARCHAR(32)	activity_state - アクティビティの状態モニター・エレメント

表 374. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
ACTIVITY_STATE (続き)	VARCHAR(32)	<p>アクティビティーの状態。可能な値は、以下のとおりです。</p> <p><b>QUEUED</b></p> <p>アクティビティーが、ワークロード管理キューイングしきい値によってキューに入れられています。パーティション・データベース環境では、この状態は、コーディネーター・エージェントがカタログ・パーティションに対する RPC を行ってしきい値チケットを取得したものの、まだ応答を受け取っていないことを示す場合があります。この状態は、アクティビティーがワークロード管理キューイングしきい値によってキューに入れられているか、それほど時間が経過していない場合はアクティビティーがそのチケットを取得する処理中であることを示すことがあります。アクティビティーがキューに入れられているかどうかについての詳細を知るために、どのエージェントがアクティビティーで作業しているかを判別し、カタログ・パーティションにあるオブジェクトの <b>EVENT_OBJECT</b> の値が <b>WLM_QUEUE</b> の値であるかどうかを調べてください。</p> <p><b>TERMINATING</b></p> <p>アクティビティーは実行を完了し、システムから除去されています。</p>
ACTIVITY_TYPE	VARCHAR(32)	<p>アクティビティー・タイプ。可能な値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• CALL</li> <li>• DDL</li> <li>• LOAD</li> <li>• OTHER</li> <li>• READ_DML</li> <li>• WRITE_DML</li> </ul>
NESTING_LEVEL	INTEGER	nesting_level - ネスティング・レベル・モニター・エレメント
INVOCATION_ID	INTEGER	invocation_id - 呼び出し ID モニター・エレメント
ROUTINE_ID	INTEGER	routine_id - ルーチン ID モニター・エレメント



表 374. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
UTILITY_ID	INTEGER	utility_id - ユーティリティ ID モニター・エレメント
SERVICE_CLASS_ID	INTEGER	service_class_id - サービス・クラス ID モニター・エレメント
DATABASE_WORK_ACTION_SET_ID	INTEGER	以下の値のいずれか。 <ul style="list-style-type: none"> <li>このアクティビティがデータベース有効範囲の作業クラスに分類されている場合、値はこの作業クラスがメンバーとなっている作業クラス・セットの ID です。</li> <li>このアクティビティがデータベース有効範囲の作業クラスに分類されていない場合、値は NULL です。</li> </ul>
DATABASE_WORK_CLASS_ID	INTEGER	以下の値のいずれか。 <ul style="list-style-type: none"> <li>このアクティビティがデータベース有効範囲の作業クラスに分類されている場合、値は作業クラスの ID です。</li> <li>このアクティビティがデータベース有効範囲の作業クラスに分類されていない場合、値は NULL です。</li> </ul>
SERVICE_CLASS_WORK_ACTION_SET_ID	INTEGER	以下の値のいずれか。 <ul style="list-style-type: none"> <li>このアクティビティがサービス・クラス有効範囲の作業クラスに分類されている場合、値は作業クラスが属する作業クラス・セットに関連付けられた作業アクション・セットの ID です。</li> <li>このアクティビティがサービス・クラス有効範囲の作業クラスに分類されていない場合、値は NULL です。</li> </ul>
SERVICE_CLASS_WORK_CLASS_ID	INTEGER	以下の値のいずれか。 <ul style="list-style-type: none"> <li>このアクティビティがサービス・クラス有効範囲の作業クラスに分類されている場合、この値はこのアクティビティに割り当てられた作業クラスの ID です。</li> <li>このアクティビティがサービス・クラス有効範囲の作業クラスに分類されていない場合、値は NULL です。</li> </ul>
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	executable_id - 実行可能 ID モニター・エレメント
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 合計 CPU 時間
ROWS_READ	BIGINT	rows_read - 読み取り行数
ROWS_RETURNED	BIGINT	rows_returned - 戻り行数
QUERY_COST_ESTIMATE	BIGINT	query_cost_estimate - 照会コストの見積もり
DIRECT_READS	BIGINT	direct_reads - データベースからの直接読み取り

表 374. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
DIRECT_WRITES	BIGINT	direct_writes - データベースへの直接書き込み

## WLM\_GET\_WORKLOAD\_STATS\_V97 - ワークロード統計を戻す

WLM\_GET\_WORKLOAD\_STATS\_V97 表関数は、ワークロード名とデータベース・メンバー番号のすべての組み合わせについてワークロード統計を返します。

**注:** この表関数は使用すべきではなく、WLM\_GET\_WORKLOAD\_STATS 表関数に置き換えられました。

戻される可能性のある情報の完全なリストは、1589 ページの表 375 を参照してください。

### 構文

```
►►—WLM_GET_WORKLOAD_STATS_V97—(—workload_name—,—member—)—————►►
```

スキーマは SYSPROC です。

### 表関数パラメーター

#### *workload\_name*

統計が戻されるワークロードを指定する、タイプ VARCHAR(128) の入力引数。引数が NULL または空ストリングである場合、すべてのワークロードについて統計が戻されます。

#### *member*

現在接続されているデータベースと同じインスタンス内のメンバーの番号を指定する、タイプ INTEGER の入力引数。現行のメンバーには -1、すべてのメンバーには -2 を指定します。NULL 値を指定すると、-1 が暗黙的に設定されます。

### 許可

このルーチンを実行するには、以下のいずれかの権限が必要です。

- ルーチンに対する EXECUTE 特権
- DATAACCESS 権限
- DBADM 権限
- SQLADM 権限
- WLMADM 権限

### デフォルトの PUBLIC 特権

なし

## 例

以下の照会は、ワークロードの統計を表示します。

```
SELECT SUBSTR(WORKLOAD_NAME,1,18) AS WL_DEF_NAME,
       SUBSTR(CHAR(DBPARTITIONNUM),1,4) AS PART,
       COORD_ACT_LIFETIME_TOP,
       COORD_ACT_LIFETIME_AVG,
       COORD_ACT_LIFETIME_STDDEV
FROM TABLE(WLM_GET_WORKLOAD_STATS_V97(CAST(NULL AS VARCHAR(128)), -2)) AS WLSTATS
ORDER BY WL_DEF_NAME, PART
```

照会からの出力例は、次のようになります。

```
WL_DEF_NAME      PART COORD_ACT_LIFETIME_TOP ...
-----
SYSDEFAULTADWORKL 0                -1 ...
SYSDEFAULTUSERWORK 0                -1 ...
WL1                0                 2 ...

... COORD_ACT_LIFETIME_AVG  COORD_ACT_LIFETIME_STDDEV
... -----
... -1.000000000000000E+000  -1.000000000000000E+000
... -1.000000000000000E+000  -1.000000000000000E+000
... +2.560000000000000E+000  +6.00000000000001E-002
```

## 使用上の注意

この関数は、ワークロード、メンバー、またはサービス・クラスを越えたデータ集約は行いません。ただし、SQL 照会を使用すると、データを集約することができます。

## 戻される情報

表 375. WLM\_GET\_WORKLOAD\_STATS\_V97 によって戻される情報

列名	データ・タイプ	説明
WORKLOAD_NAME	VARCHAR(128)	workload_name - ワークロード名 : モニター・エレメント
DBPARTITIONNUM	SMALLINT	dbpartitionnum - データベース・パーティション番号 モニター・エレメント
LAST_RESET	TIMESTAMP	last_reset - 最後のリセット・タイム・スタンプ・モニター・エレメント
CONCURRENT_WLO_TOP	INTEGER	concurrent_wlo_top - 並行ワークロード・オカレンスの最上位モニター・エレメント
CONCURRENT_WLO_ACT_TOP	INTEGER	concurrent_wlo_act_top - 並行 WLO アクティビティの最上位モニター・エレメント
COORD_ACT_COMPLETED_TOTAL	BIGINT	coord_act_completed_total - コーディネーター・アクティビティ完了総数モニター・エレメント
COORD_ACT_ABORTED_TOTAL	BIGINT	coord_act_aborted_total - コーディネーター・アクティビティ打ち切り総数モニター・エレメント
COORD_ACT_REJECTED_TOTAL	BIGINT	coord_act_rejected_total - コーディネーター・アクティビティ・リジェクト総数モニター・エレメント
WLO_COMPLETED_TOTAL	BIGINT	wlo_completed_total - 完了したワークロード・オカレンスの合計モニター・エレメント

表 375. WLM\_GET\_WORKLOAD\_STATS\_V97 によって戻される情報 (続き)

列名	データ・タイプ	説明
COORD_ACT_LIFETIME_TOP	BIGINT	coord_act_lifetime_top - コーディネーター・アクティビティ存続時間の最上位モニター・エレメント
COORD_ACT_LIFETIME_AVG	DOUBLE	coord_act_lifetime_avg - コーディネーター・アクティビティ平均存続期間モニター・エレメント
COORD_ACT_LIFETIME_STDDEV	DOUBLE	このワークロードに関連付けられたネスト・レベル 0 の完了または打ち切られたコーディネーター・アクティビティの存続期間の標準偏差。単位はミリ秒です。ワークロードの COLLECT AGGREGATE ACTIVITY DATA パラメーターが NONE に設定される場合、列の値は NULL です。この標準偏差はコーディネーター・アクティビティの存続期間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。
COORD_ACT_EXEC_TIME_AVG	DOUBLE	coord_act_exec_time_avg - コーディネーター・アクティビティ平均実行時間モニター・エレメント
COORD_ACT_EXEC_TIME_STDDEV	DOUBLE	このワークロードに関連付けられたネスト・レベル 0 の完了または打ち切られたコーディネーター・アクティビティの実行時間の標準偏差。単位はミリ秒です。この標準偏差はコーディネーター・アクティビティの実行時間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。ワークロードの COLLECT AGGREGATE ACTIVITY DATA パラメーターが NONE に設定される場合、列の値は NULL です。
COORD_ACT_QUEUE_TIME_AVG	DOUBLE	coord_act_queue_time_avg - コーディネーター・アクティビティ平均キュー時間モニター・エレメント
COORD_ACT_QUEUE_TIME_STDDEV	DOUBLE	このワークロードに関連付けられたネスト・レベル 0 の完了または打ち切られたコーディネーター・アクティビティのキュー時間の標準偏差。単位はミリ秒です。ワークロードの COLLECT AGGREGATE ACTIVITY DATA パラメーターが NONE に設定される場合、列の値は NULL です。この標準偏差はコーディネーター・アクティビティのキュー時間ヒストグラムから計算され、ヒストグラムのサイズがデータに合わせて正しく設定されていない場合は不正確になることがあります。値が最後のヒストグラム bin に入る場合、値 -1 が戻されます。
UOW_TOTAL_TIME_TOP	BIGINT	uow_total_time_top - 作業単位合計時間トップ・モニター・エレメント

---

## 付録 A. DB2 技術情報の概説

DB2 技術情報は、さまざまな方法でアクセスすることが可能な、各種形式で入手できます。

DB2 技術情報は、以下のツールと方法を介して利用できます。

- DB2インフォメーション・センター
  - トピック (タスク、概念、およびリファレンス・トピック)
  - サンプル・プログラム
  - チュートリアル
- DB2 資料
  - PDF ファイル (ダウンロード可能)
  - PDF ファイル (DB2 PDF DVD に含まれる)
  - 印刷資料
- コマンド行ヘルプ
  - コマンド・ヘルプ
  - メッセージ・ヘルプ

**注:** DB2 インフォメーション・センターのトピックは、PDF やハードコピー資料よりも頻繁に更新されます。最新の情報を入手するには、資料の更新が発行されたときにそれをインストールするか、[ibm.com](http://ibm.com) にある DB2 インフォメーション・センターを参照してください。

技術資料、ホワイト・ペーパー、IBM Redbooks® 資料などのその他の DB2 技術情報には、オンライン ([ibm.com](http://ibm.com)) でアクセスできます。DB2 Information Management ソフトウェア・ライブラリー・サイト (<http://www.ibm.com/software/data/sw-library/>) にアクセスしてください。

### 資料についてのフィードバック

DB2 の資料についてのお客様からの貴重なご意見をお待ちしています。DB2 の資料を改善するための提案については、[db2docs@ca.ibm.com](mailto:db2docs@ca.ibm.com) まで E メールを送信してください。DB2 の資料チームは、お客様からのフィードバックすべてに目を通しますが、直接お客様に返答することはありません。お客様が関心をお持ちの内容について、可能な限り具体的な例を提供してください。特定のトピックまたはヘルプ・ファイルについてのフィードバックを提供する場合は、そのトピック・タイトルおよび URL を含めてください。

DB2 お客様サポートに連絡する場合には、この E メール・アドレスを使用しないでください。資料を参照しても、DB2 の技術的な問題が解決しない場合は、お近くの IBM サービス・センターにお問い合わせください。

## DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)

以下の表は、IBM Publications Center ([www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss](http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss)) から利用できる DB2 ライブラリーについて説明しています。英語および翻訳された DB2 バージョン 10.1 のマニュアル (PDF 形式) は、[www.ibm.com/support/docview.wss?rs=71&uid=swg27009474](http://www.ibm.com/support/docview.wss?rs=71&uid=swg27009474) からダウンロードできます。

この表には印刷資料が入手可能かどうかを示されていますが、国または地域によっては入手できない場合があります。

資料番号は、資料が更新される度に大きくなります。資料を参照する際は、以下にリストされている最新版であることを確認してください。

注: DB2 インフォメーション・センターは、PDF やハードコピー資料よりも頻繁に更新されます。

表 376. DB2 の技術情報

資料名	資料番号	印刷資料が入手可能かどうか	最終更新
管理 API リファレンス	SA88-4671-00	入手可能	2012 年 4 月
管理ルーチンおよびビュー	SA88-4672-01	入手不可	2013 年 1 月
コール・レベル・イン ターフェース ガイドお よびリファレンス 第 1 巻	SA88-4676-01	入手可能	2013 年 1 月
コール・レベル・イン ターフェース ガイドお よびリファレンス 第 2 巻	SA88-4677-01	入手可能	2013 年 1 月
コマンド・リファレン ス	SA88-4673-01	入手可能	2013 年 1 月
データベース: 管理の 概念および構成リファ レンス	SA88-4662-01	入手可能	2013 年 1 月
データ移動ユーティリ ティー: ガイドおよび リファレンス	SA88-4693-01	入手可能	2013 年 1 月
データベースのモニタ リング ガイドおよびリ ファレンス	SA88-4663-01	入手可能	2013 年 1 月
データ・リカバリーと 高可用性 ガイドおよび リファレンス	SA88-4694-01	入手可能	2013 年 1 月
データベース・セキュ リティ・ガイド	SA88-4695-01	入手可能	2013 年 1 月

表 376. DB2 の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能かどうか	最終更新
DB2 ワークロード管理ガイドおよびリファレンス	SA88-4685-01	入手可能	2013 年 1 月
ADO.NET および OLE DB アプリケーションの開発	SA88-4665-01	入手可能	2013 年 1 月
組み込み SQL アプリケーションの開発	SA88-4666-01	入手可能	2013 年 1 月
Java アプリケーションの開発	SA88-4669-01	入手可能	2013 年 1 月
Perl、PHP、Python および Ruby on Rails アプリケーションの開発	SA88-4670-00	入手不可	2012 年 4 月
IBM データ・サーバー用の RDF アプリケーション開発	SA88-5083-00	入手可能	2013 年 1 月
SQL および外部ルーチンの開発	SA88-4667-01	入手可能	2013 年 1 月
データベース・アプリケーション開発の基礎	GI88-4279-01	入手可能	2013 年 1 月
DB2 インストールおよび管理 概説 (Linux および Windows 版)	GI88-4280-00	入手可能	2012 年 4 月
グローバリゼーション・ガイド	SA88-4696-00	入手可能	2012 年 4 月
DB2 サーバー機能 インストール	GA88-4679-01	入手可能	2013 年 1 月
IBM データ・サーバー・クライアント機能インストール	GA88-4680-00	入手不可	2012 年 4 月
メッセージ・リファレンス 第 1 巻	SA88-4688-01	入手不可	2013 年 1 月
メッセージ・リファレンス 第 2 巻	SA88-4689-01	入手不可	2013 年 1 月
Net Search Extender 管理およびユーザズ・ガイド	SA88-4691-01	入手不可	2013 年 1 月
パーティションおよびクラスタリングのガイド	SA88-4697-01	入手可能	2013 年 1 月
Preparation Guide for DB2 10.1 Fundamentals Exam 610	SC27-4540-00	入手不可	2013 年 1 月

表 376. DB2 の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能 かどうか	最終更新
<i>Preparation Guide for DB2 10.1 DBA for Linux, UNIX, and Windows Exam 611</i>	SC27-4541-00	入手不可	2013 年 1 月
<i>pureXML ガイド</i>	SA88-4686-01	入手可能	2013 年 1 月
<i>Spatial Extender ユーザーズ・ガイドおよびリファレンス</i>	SA88-4690-00	入手不可	2012 年 4 月
<i>SQL プロシージャ言語: アプリケーションのイネーブルメントおよびサポート</i>	SA88-4668-01	入手可能	2013 年 1 月
<i>SQL リファレンス 第 1 巻</i>	SA88-4674-01	入手可能	2013 年 1 月
<i>SQL リファレンス 第 2 巻</i>	SA88-4675-01	入手可能	2013 年 1 月
<i>Text Search ガイド</i>	SA88-4692-01	入手可能	2013 年 1 月
<i>問題判別およびデータベース・パフォーマンスのチューニング</i>	SA88-4664-01	入手可能	2013 年 1 月
<i>DB2 バージョン 10.1 へのアップグレード</i>	SA88-4678-01	入手可能	2013 年 1 月
<i>DB2 バージョン 10.1 の新機能</i>	SA88-4684-01	入手可能	2013 年 1 月
<i>XQuery リファレンス</i>	SA88-4687-01	入手不可	2013 年 1 月

表 377. DB2 Connect 固有の技術情報

資料名	資料番号	印刷資料が入手可能 かどうか	最終更新
<i>DB2 Connect Personal Edition</i> インストールおよび構成	SA88-4681-00	入手可能	2012 年 4 月
<i>DB2 Connect サーバー機能</i> インストールおよび構成	SA88-4682-01	入手可能	2013 年 1 月
<i>DB2 Connect ユーザーズ・ガイド</i>	SA88-4683-01	入手可能	2013 年 1 月

## コマンド行プロセッサから SQL 状態ヘルプを表示する

DB2 製品は、SQL ステートメントの結果として生じる可能性がある状態に対応した SQLSTATE 値を戻します。SQLSTATE ヘルプは、SQL 状態および SQL 状態クラス・コードの意味を説明します。



## 手順

SQL 状態ヘルプを開始するには、コマンド行プロセッサを開いて以下のように入力します。

```
? sqlstate または ? class code
```

ここで、*sqlstate* は有効な 5 桁の SQL 状態を、*class code* は SQL 状態の最初の 2 桁を表します。

例えば、? 08003 を指定すると SQL 状態 08003 のヘルプが表示され、? 08 を指定するとクラス・コード 08 のヘルプが表示されます。

---

## 異なるバージョンの DB2 インフォメーション・センターへのアクセス

他のバージョンの DB2 製品の資料は、ibm.com<sup>®</sup> のそれぞれのインフォメーション・センターにあります。

### このタスクについて

DB2 バージョン 10.1 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v10r1> です。

DB2 バージョン 9.8 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/> です。

DB2 バージョン 9.7 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/> です。

DB2 バージョン 9.5 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/> です。

DB2 バージョン 9.1 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9/> です。

DB2 バージョン 8 のトピックについては、DB2 インフォメーション・センターの URL (<http://publib.boulder.ibm.com/infocenter/db2luw/v8/>) を参照してください。

---

## コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの更新

ローカルにインストールした DB2 インフォメーション・センターは、定期的に更新する必要があります。

### 始める前に

DB2 バージョン 10.1 インフォメーション・センターが既にインストール済みである必要があります。詳しくは、「DB2 サーバー機能 インストール」の『DB2 セットアップ・ウィザードによる DB2 インフォメーション・センターのインストール』のトピックを参照してください。インフォメーション・センターのインストールに適用されるすべての前提条件と制約事項は、インフォメーション・センターの更新にも適用されます。

## このタスクについて

既存の DB2 インフォメーション・センターは、自動で更新することも手動で更新することもできます。

- 自動更新は、既存のインフォメーション・センターのフィーチャーと言語を更新します。自動更新を使用すると、手動更新と比べて、更新中にインフォメーション・センターが使用できなくなる時間が短くなるというメリットがあります。さらに、自動更新は、定期的に行う他のバッチ・ジョブの一部として実行されるように設定することができます。
- 手動更新は、既存のインフォメーション・センターのフィーチャーと言語の更新に使用できます。自動更新は更新処理中のダウン時間を減らすことができますが、フィーチャーまたは言語を追加する場合は手動処理を使用する必要があります。例えば、ローカルのインフォメーション・センターが最初は英語とフランス語でインストールされており、その後ドイツ語もインストールすることにした場合、手動更新でドイツ語をインストールし、同時に、既存のインフォメーション・センターのフィーチャーおよび言語を更新できます。しかし、手動更新ではインフォメーション・センターを手動で停止、更新、再始動する必要があります。更新処理の間はずっと、インフォメーション・センターは使用できなくなります。自動更新処理では、インフォメーション・センターは、更新を行った後に、インフォメーション・センターを再始動するための停止が発生するだけで済みます。

このトピックでは、自動更新のプロセスを詳しく説明しています。手動更新の手順については、『コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新』のトピックを参照してください。

## 手順

コンピューターまたはイントラネット・サーバーにインストールされている DB2 インフォメーション・センターを自動更新する手順を以下に示します。

1. Linux オペレーティング・システムの場合、次のようにします。
  - a. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは、`/opt/ibm/db2ic/V10.1` ディレクトリーにインストールされています。
  - b. インストール・ディレクトリーから `doc/bin` ディレクトリーにナビゲートします。
  - c. 次のように `update-ic` スクリプトを実行します。

```
update-ic
```
2. Windows オペレーティング・システムの場合、次のようにします。
  - a. コマンド・ウィンドウを開きます。
  - b. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは、`<Program Files>%IBM%DB2 Information Center%バージョン 10.1` ディレクトリーにインストールされています (`<Program Files>` は「Program Files」ディレクトリーのロケーション)。

- c. インストール・ディレクトリーから doc¥bin ディレクトリーにナビゲートします。
- d. 次のように update-ic.bat ファイルを実行します。

```
update-ic.bat
```

## タスクの結果

DB2 インフォメーション・センターが自動的に再始動します。更新が入手可能な場合、インフォメーション・センターに、更新された新しいトピックが表示されます。インフォメーション・センターの更新が入手可能でなかった場合、メッセージがログに追加されます。ログ・ファイルは、doc¥eclipse¥configuration ディレクトリーにあります。ログ・ファイル名はランダムに生成された名前です。例えば、1239053440785.log のようになります。

---

## コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新

DB2 インフォメーション・センターをローカルにインストールしている場合は、IBM から資料の更新を入手してインストールすることができます。

### このタスクについて

ローカルにインストールされた DB2 インフォメーション・センター を手動で更新するには、以下のことを行う必要があります。

1. コンピューター上の DB2 インフォメーション・センター を停止し、インフォメーション・センターをスタンドアロン・モードで再始動します。インフォメーション・センターをスタンドアロン・モードで実行すると、ネットワーク上の他のユーザーがそのインフォメーション・センターにアクセスできなくなります。これで、更新を適用できるようになります。DB2 インフォメーション・センターのワークステーション・バージョンは、常にスタンドアロン・モードで実行されます。を参照してください。
2. 「更新」機能を使用することにより、どんな更新が利用できるかを確認します。インストールしなければならない更新がある場合は、「更新」機能を使用してそれを入手およびインストールできます。

**注:** ご使用の環境において、インターネットに接続されていないマシンに DB2 インフォメーション・センター の更新をインストールする必要がある場合、インターネットに接続されていて DB2 インフォメーション・センター がインストールされているマシンを使用して、更新サイトをローカル・ファイル・システムにミラーリングしてください。ネットワーク上の多数のユーザーが資料の更新をインストールする場合にも、更新サイトをローカルにミラーリングして、更新サイト用のプロキシを作成することにより、個々のユーザーが更新を実行するのに要する時間を短縮できます。

更新パッケージが入手可能な場合、「更新」機能を使用してパッケージを入手します。ただし、「更新」機能は、スタンドアロン・モードでのみ使用できます。

3. スタンドアロンのインフォメーション・センターを停止し、コンピューター上の DB2 インフォメーション・センター を再開します。

注: Windows 2008、Windows Vista (およびそれ以上) では、このセクションの後の部分でリストされているコマンドは管理者として実行する必要があります。完全な管理者特権でコマンド・プロンプトまたはグラフィカル・ツールを開くには、ショートカットを右クリックしてから、「管理者として実行」を選択します。

## 手順

コンピューターまたはイントラネット・サーバーにインストール済みの *DB2* インフォメーション・センター を更新するには、以下のようになります。

1. *DB2* インフォメーション・センター を停止します。
    - Windows では、「スタート」 > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックします。次に、「**DB2** インフォメーション・センター」サービスを右クリックして「停止」を選択します。
    - Linux では、以下のコマンドを入力します。

```
/etc/init.d/db2icdv10 stop
```
  2. インフォメーション・センターをスタンドアロン・モードで開始します。
    - Windows の場合:
      - a. コマンド・ウィンドウを開きます。
      - b. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、*DB2* インフォメーション・センター は、`Program_Files¥IBM¥DB2 Information Center¥バージョン 10.1` ディレクトリーにインストールされています (`Program_Files` は Program Files ディレクトリーのロケーション)。
      - c. インストール・ディレクトリーから `doc¥bin` ディレクトリーにナビゲートします。
      - d. 次のように `help_start.bat` ファイルを実行します。

```
help_start.bat
```
    - Linux の場合:
      - a. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、*DB2* インフォメーション・センター は、`/opt/ibm/db2ic/V10.1` ディレクトリーにインストールされています。
      - b. インストール・ディレクトリーから `doc/bin` ディレクトリーにナビゲートします。
      - c. 次のように `help_start` スクリプトを実行します。

```
help_start
```
- システムのデフォルト Web ブラウザーが開き、スタンドアロンのインフォメーション・センターが表示されます。
3. 「更新」ボタン (🔄) をクリックします。(ブラウザーで JavaScript が有効になっている必要があります。) インフォメーション・センターの右側のパネルで、「更新の検索」をクリックします。既存の文書に対する更新のリストが表示されます。
  4. インストール・プロセスを開始するには、インストールする更新をチェックして選択し、「更新のインストール」をクリックします。
  5. インストール・プロセスが完了したら、「完了」をクリックします。

6. 次のようにして、スタンドアロンのインフォメーション・センターを停止します。

- Windows の場合は、インストール・ディレクトリーの `doc\bin` ディレクトリーにナビゲートしてから、次のように `help_end.bat` ファイルを実行します。

```
help_end.bat
```

注: `help_end` バッチ・ファイルには、`help_start` バッチ・ファイルを使用して開始したプロセスを安全に停止するのに必要なコマンドが含まれています。`help_start.bat` は、Ctrl-C や他の方法を使用して停止しないでください。

- Linux の場合は、インストール・ディレクトリーの `doc/bin` ディレクトリーにナビゲートしてから、次のように `help_end` スクリプトを実行します。

```
help_end
```

注: `help_end` スクリプトには、`help_start` スクリプトを使用して開始したプロセスを安全に停止するのに必要なコマンドが含まれています。他の方法を使用して、`help_start` スクリプトを停止しないでください。

7. **DB2** インフォメーション・センター を再開します。

- Windows では、「スタート」 > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックします。次に、「**DB2** インフォメーション・センター」サービスを右クリックして「開始」を選択します。

- Linux では、以下のコマンドを入力します。

```
/etc/init.d/db2icdv10 start
```

## タスクの結果

更新された **DB2** インフォメーション・センター に、更新された新しいトピックが表示されます。

---

## DB2 チュートリアル

DB2 チュートリアルは、DB2 データベース製品のさまざまな機能について学習するための支援となります。この演習をとおして段階的に学習することができます。

### はじめに

インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/>) から、このチュートリアルの XHTML 版を表示できます。

演習の中で、サンプル・データまたはサンプル・コードを使用する場合があります。個々のタスクの前提条件については、チュートリアルを参照してください。

### DB2 チュートリアル

チュートリアルを表示するには、タイトルをクリックします。

「*pureXML* ガイド」の『**pureXML**』

XML データを保管し、ネイティブ XML データ・ストアに対して基本的な操作を実行できるように、DB2 データベースをセットアップします。

---

## DB2 トラブルシューティング情報

DB2 データベース製品を使用する際に役立つ、トラブルシューティングおよび問題判別に関する広範囲な情報を利用できます。

### DB2 の資料

トラブルシューティング情報は、「問題判別およびデータベース・パフォーマンスのチューニング」または *DB2* インフォメーション・センター の『データベースの基本』セクションにあります。ここには、以下の情報が記載されています。

- DB2 診断ツールおよびユーティリティーを使用した、問題の切り分け方法および識別方法に関する情報。
- 最も一般的な問題のうち、いくつかの解決方法。
- DB2 データベース製品で発生する可能性のある、その他の問題の解決に役立つアドバイス。

### IBM サポート・ポータル

現在問題が発生していて、考えられる原因とソリューションを見つけるには、IBM サポート・ポータルを参照してください。Technical Support サイトには、最新の DB2 資料、TechNotes、プログラム診断依頼書 (APAR またはバグ修正)、フィックスパック、およびその他のリソースへのリンクが用意されています。この知識ベースを活用して、問題に対する有効なソリューションを探し出すことができます。

IBM サポート・ポータル ([http://www.ibm.com/support/entry/portal/Overview/Software/Information\\_Management/DB2\\_for\\_Linux,\\_UNIX\\_and\\_Windows](http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_for_Linux,_UNIX_and_Windows)) にアクセスしてください。

---

## ご利用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

**適用度:** これらのご利用条件は、IBM Web サイトのあらゆるご利用条件に追加で適用されるものです。

**個人使用:** これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

**商業的使用:** これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

**権利:** ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

**IBM の商標:** IBM、IBM ロゴおよび [ibm.com](http://ibm.com) は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。





---

## 付録 B. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。IBM 以外の製品に関する情報は、本書の最初の発行時点で入手可能な情報に基づいており、変更される場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510  
東京都中央区日本橋箱崎町19番21号  
日本アイ・ビー・エム株式会社  
法務・知的財産  
知的財産権ライセンス渉外

**以下の保証は、国または地域の法律に沿わない場合は、適用されません。** IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Canada Limited  
U59/3600  
3600 Steeles Avenue East  
Markham, Ontario L3R 9Z7  
CANADA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、

利便性もしくは機能性があることをほのめかしたり、保証することはできません。サンプル・プログラムは、現存するままの状態を提供されるものであり、いかなる種類の保証も提供されません。IBM は、これらのサンプル・プログラムの使用から生ずるいかなる損害に対しても責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. \_年を入れる\_. All rights reserved.

## 商標

IBM、IBM ロゴおよび [ibm.com](http://ibm.com) は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

以下は、それぞれ各社の商標または登録商標です。

- Linux は、Linus Torvalds の米国およびその他の国における商標です。
- Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。
- UNIX は The Open Group の米国およびその他の国における登録商標です。
- インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Celeron、Intel SpeedStep、Itanium、Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。
- Microsoft、Windows、Windows NT、および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。



# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

インスタンス

現在の

情報の検索 415

インスタンス所有者許可 ID

取得 806

インストール

ライセンス情報 406, 1310

DB2 製品情報の取得 417

エクステント

移動状況 578

エラー・メッセージ

検索

SQLERRM スカラー関数 1248

オブジェクト

所有権の検索 817

オンライン表移動

ADMIN\_MOVE\_TABLE プロシージャ

詳細 263

ADMIN\_MOVE\_TABLE\_UTIL プロシージャ 284

## [カ行]

関数

スカラー

ADMIN\_GET\_INTRA\_PARALLEL 250

APPLICATION\_ID 1305

AUTH\_GET\_INSTANCE\_AUTHID 806

EXPLAIN\_FORMAT\_STATS 422

GET\_ROUTINE\_OPTS 1098

MON\_GET\_APPLICATION\_HANDLE 530

MON\_GET\_APPLICATION\_ID 531

MQPUBLISH 783

MQREAD 785

MQREADCLOB 791

MQRECEIVE 793

MQRECEIVECLOB 799

MQSEND 801

MQSUBSCRIBE 803

MQUNSUBSCRIBE 805

SQLERRM 1248

ストアド・プロシージャ

SYSTS\_ALTER 1121

SYSTS\_CLEANUP 1127

SYSTS\_CLEAR\_COMMANDLOCKS 1129

関数 (続き)

ストアド・プロシージャ (続き)

SYSTS\_CLEAR\_EVENTS 1131

SYSTS\_DROP 1147

SYSTS\_ENABLE 1150

SYSTS\_UPDATE 1152

SYSTS\_UPGRADE\_CATALOG 1155

SYSTS\_UPGRADE\_INDEX 1157

表

概要 1

組み込みビューの比較 3

使用すべきでない 1253

要約 4

ADMIN\_GET\_MSGS 252

ADMIN\_GET\_STORAGE\_PATHS 254

ADMIN\_GET\_TAB\_COMPRESS\_INFO (使用すべきでない) 1261

ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 1267

ADMIN\_GET\_TAB\_INFO 294

ADMIN\_GET\_TAB\_INFO\_V95 1273

ADMIN\_GET\_TAB\_INFO\_V97 1280

ADMIN\_GET\_TEMP\_COLUMNS 304

ADMIN\_GET\_TEMP\_TABLES 308

ADMIN\_IS\_INLINED 262

ADMIN\_IS\_INLINED\_LENGTH 239

ADMIN\_LIST\_DB\_PATHS 1219

AM\_BASE\_RPTS 1290

AM\_BASE\_RPT\_RECOMS 1289

AUDIT\_ARCHIVE 325

AUDIT\_LIST\_LOGS 327

AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID 808

AUTH\_LIST\_GROUPS\_FOR\_AUTHID 812

AUTH\_LIST\_ROLES\_FOR\_AUTHID 813

COMPILATION\_ENV 1209

DB\_GET\_CFG 388

DB\_MEMBERS 395

DB\_PARTITIONS 1306

ENV\_GET\_DB2\_SYSTEM\_RESOURCES 407

ENV\_GET\_NETWORK\_RESOURCES 408

ENV\_GET\_REG\_VARIABLES 410

ENV\_GET\_SYSTEM\_RESOURCES 412

EVMON\_FORMAT\_UE\_TO\_XML 453

EXPLAIN\_GET\_MSGS 420

GET\_DBM\_CONFIG 1309

GET\_DB\_CONFIG 1307

HEALTH\_CONT\_HI 1320

HEALTH\_CONT\_HI\_HIS 1322

HEALTH\_CONT\_INFO 1324

HEALTH\_DBM\_HI 1341

HEALTH\_DBM\_HI\_HIS 1343

HEALTH\_DBM\_INFO 1345

## 関数 (続き)

## 表 (続き)

HEALTH\_DB\_HI 1326  
 HEALTH\_DB\_HIC 1334  
 HEALTH\_DB\_HIC\_HIS 1336  
 HEALTH\_DB\_HI\_HIS 1330  
 HEALTH\_DB\_INFO 1339  
 HEALTH\_GET\_ALERT\_ACTION\_CFG 1347  
 HEALTH\_GET\_ALERT\_CFG 1351  
 HEALTH\_GET\_IND\_DEFINITION 1355  
 HEALTH\_TBS\_HI 1359  
 HEALTH\_TBS\_HI\_HIS 1362  
 HEALTH\_TBS\_INFO 1366  
 MON\_GET\_ACTIVITY\_DETAILS 512  
 MON\_GET\_AUTO\_MAINT\_QUEUE 532  
 MON\_GET\_AUTO\_RUNSTATS\_QUEUE 535  
 MON\_GET\_BUFFERPOOL 537  
 MON\_GET\_CF 544  
 MON\_GET\_CF\_CMD 546  
 MON\_GET\_CF\_WAIT\_TIME 550  
 MON\_GET\_CONNECTION 553  
 MON\_GET\_CONNECTION\_DETAILS 563  
 MON\_GET\_CONTAINER 574  
 MON\_GET\_EXTENDED\_LATCH\_WAIT 577  
 MON\_GET\_EXTENT\_MOVEMENT\_STATUS 578  
 MON\_GET\_FCM 580  
 MON\_GET\_FCM\_CONNECTION\_LIST 582  
 MON\_GET\_GROUP\_BUFFERPOOL 583  
 MON\_GET\_HADR 584  
 MON\_GET\_INDEX 593  
 MON\_GET\_INDEX\_USAGE\_LIST 596  
 MON\_GET\_PAGE\_ACCESS 610  
 MON\_GET\_PKG\_CACHE\_STMT 612  
 MON\_GET\_RTS\_RQST 664  
 MON\_GET\_SECTION\_ROUTINE 635, 647, 659, 666  
 MON\_GET\_SERVERLIST 668  
 MON\_GET\_SERVICE\_SUBCLASS 670  
 MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 683  
 MON\_GET\_TABLE 693  
 MON\_GET\_TABLESPACE 697  
 MON\_GET\_TABLE\_USAGE\_LIST 706  
 MON\_GET\_TRANSACTION\_LOG 710  
 MON\_GET\_UNIT\_OF\_WORK 712  
 MON\_GET\_UNIT\_OF\_WORK\_DETAILS 723  
 MON\_GET\_USAGE\_LIST\_STATUS 735  
 MON\_GET\_WORKLOAD 737  
 MON\_GET\_WORKLOAD\_DETAILS 747  
 MON\_SAMPLE\_SERVICE\_CLASS\_METRICS 762  
 MON\_SAMPLE\_WORKLOAD\_METRICS 766  
 MQREADALL 787  
 MQREADALLCLOB 789  
 MQRECEIVEALL 794  
 MQRECEIVEALLCLOB 797  
 PD\_GET\_DIAG\_HIST 1228  
 PD\_GET\_LOG\_MSGS 1235  
 SNAPSHOT\_AGENT (使用すべきでない) 1498

## 関数 (続き)

## 表 (続き)

SNAPSHOT\_APPL (使用すべきでない) 1500  
 SNAPSHOT\_APPL\_INFO (使用すべきでない) 1506  
 SNAPSHOT\_BP (使用すべきでない) 1508  
 SNAPSHOT\_CONTAINER (使用すべきでない) 1511  
 SNAPSHOT\_DATABASE (使用すべきでない) 1512  
 SNAPSHOT\_DBM (使用すべきでない) 1519  
 SNAPSHOT\_DYN\_SQL (使用すべきでない) 1521  
 SNAPSHOT\_FCM (使用すべきでない) 1523  
 SNAPSHOT\_FCMNODE (使用すべきでない) 1525  
 SNAPSHOT\_LOCK (使用すべきでない) 1527  
 SNAPSHOT\_LOCKWAIT (使用すべきでない) 1529  
 SNAPSHOT QUIESCERS (使用すべきでない) 1530  
 SNAPSHOT\_RANGES (使用すべきでない) 1532  
 SNAPSHOT\_STATEMENT (使用すべきでない) 1533  
 SNAPSHOT\_SUBSECT (使用すべきでない) 1536  
 SNAPSHOT\_SWITCHES (使用すべきでない) 1538  
 SNAPSHOT\_TABLE (使用すべきでない) 1540  
 SNAPSHOT\_TBREORG (使用すべきでない) 1541  
 SNAPSHOT\_TBS (使用すべきでない) 1543  
 SNAPSHOT\_TBS\_CFG (使用すべきでない) 1546  
 SNAP\_GET\_AGENT 841, 966  
 SNAP\_GET\_AGENT\_MEMORY\_POOL 1370, 1466  
 SNAP\_GET\_APPL 853, 970  
 SNAP\_GET\_APPL\_INFO 845, 979  
 SNAP\_GET\_APPL\_INFO\_V95 1375  
 SNAP\_GET\_APPL\_V95 1382  
 SNAP\_GET\_BP 862, 987  
 SNAP\_GET\_BP\_PART 868, 993  
 SNAP\_GET\_BP\_V95 1390  
 SNAP\_GET\_CONTAINER 872, 997  
 SNAP\_GET\_CONTAINER\_V91 (使用すべきでない)  
 1394  
 SNAP\_GET\_DBM 890, 1014  
 SNAP\_GET\_DBM\_MEMORY\_POOL 1405, 1476  
 SNAP\_GET\_DBM\_V95 1402  
 SNAP\_GET\_DB\_MEMORY\_POOL 1397, 1471  
 SNAP\_GET\_DB\_V97 1409  
 SNAP\_GET\_DETAIL\_LOG 894, 1018  
 SNAP\_GET\_DETAIL\_LOG\_V91 1420  
 SNAP\_GET\_DYN\_SQL 898, 1022  
 SNAP\_GET\_DYN\_SQL\_V95 1422  
 SNAP\_GET\_FCM 904, 1028  
 SNAP\_GET\_FCM\_PART 908, 1032  
 SNAP\_GET\_HADR 1426, 1480  
 SNAP\_GET\_LOCK 1431, 1484  
 SNAP\_GET\_LOCKWAIT 1438, 1491  
 SNAP\_GET\_STMT 911, 1035  
 SNAP\_GET\_STORAGE\_PATHS\_V97 1447, 1548  
 SNAP\_GET\_STO\_PATHS (使用すべきでない) 1445  
 SNAP\_GET\_SUBSECTION 919, 1043  
 SNAP\_GET\_SWITCHES 923, 1047  
 SNAP\_GET\_TAB 1450, 1552  
 SNAP\_GET\_TAB\_REORG 928, 1052  
 SNAP\_GET\_TAB\_V91 (使用すべきでない) 1454

関数 (続き)

表 (続き)

SNAP\_GET\_TBSP 934, 1058  
SNAP\_GET\_TBSP\_PART (使用すべきでない) 941, 1065  
SNAP\_GET\_TBSP\_PART\_V97 1457  
SNAP\_GET\_TBSP QUIESCER 946, 1070  
SNAP\_GET\_TBSP\_RANGE 951, 1075  
SNAP\_GET\_TBSP\_V91 (使用すべきでない) 1461  
SNAP\_GET\_UTIL 956, 1080  
SNAP\_GET\_UTIL\_PROGRESS 960, 1084  
SQLCACHE\_SNAPSHOT (使用すべきでない) 1556  
WLM\_GET\_ACTIVITY\_DETAILS 1558  
WLM\_GET\_CONN\_ENV 1164  
WLM\_GET\_QUEUE\_STATS 1166  
WLM\_GET\_SERVICE\_CLASS\_AGENTS 1169  
WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 1565  
WLM\_GET\_SERVICE\_CLASS\_WORKLOAD  
\_OCCURRENCES 1178  
WLM\_GET\_SERVICE\_CLASS\_WORKLOAD  
\_OCCURRENCES\_V97 1574  
WLM\_GET\_SERVICE\_SUBCLASS\_STATS 1181  
WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 1577  
WLM\_GET\_SERVICE\_SUPERCLASS\_STATS 1188  
WLM\_GET\_WORKLOAD\_OCCURRENCE  
\_ACTIVITIES 1191  
WLM\_GET\_WORKLOAD\_OCCURRENCE  
\_ACTIVITIES\_V97 1583  
WLM\_GET\_WORKLOAD\_STATS 1196  
WLM\_GET\_WORKLOAD\_STATS\_V97 1588  
WLM\_GET\_WORK\_ACTION\_SET\_STATS 1190

表関数

MON\_FORMAT\_XML\_COMPONENT\_TIMES  
\_BY\_ROW 485  
MON\_FORMAT\_XML\_METRICS\_BY\_ROW 489  
MON\_FORMAT\_XML\_TIMES\_BY\_ROW 502  
MON\_FORMAT\_XML\_WAIT\_TIMES\_BY\_ROW 508  
MON\_GET\_PKG\_CACHE\_STMT\_DETAILS 623  
MON\_GET\_REBALANCE\_STATUS 633  
SNAP\_GET\_DB 877, 1001

要約 4

管理タスク・スケジューラー

タスク・スケジュールの定義 316

共通 SQL API ストアード・プロシージャ

概要 335  
コンプリート・モード 338  
シグニチャー 335  
出力のフィルタリング 339  
ストアード・プロシージャ 336  
XML 出力ファイル 338  
XML 入力文書 337  
XML メッセージ文書 340

許可 ID

インスタンス所有者 806  
組み込みビュー 2  
組み込みルーチン 2  
グループのメンバーシップ・リスト 812

許可 ID (続き)

検索 816

組み込みビュー

概要 1  
権限 2  
コーディングの手法 1  
表関数の比較 3  
要約 4

ADMINTABCOMPRESSINFO (使用すべきでない) 1261

ADMINTABINFO 294

ADMINTEMPCOLUMNS 304

ADMINTEMPTABLES 308

ADMIN\_TASK\_LIST 317

ADMIN\_TASK\_STATUS 320

APPLICATIONS 821

APPL\_PERFORMANCE

詳細 820

AUTHORIZATIONIDS

詳細 816

BP\_HITRATIO

詳細 827

BP\_READ\_IO

詳細 829

BP\_WRITE\_IO

詳細 831

CONTACTGROUPS 1212

CONTACTS 1213

CONTAINER\_UTILIZATION 833

DB2\_CF

詳細 400

DB2\_CLUSTER\_HOST\_STATE

詳細 396

DB2\_INSTANCE\_ALERTS

詳細 399

DB2\_MEMBER

詳細 400

DBCFCG 388

DBMCFG 392

DBPATHS 1219

DB\_HISTORY

詳細 1214

ENV\_CF\_SYS\_RESOURCES 404

ENV\_FEATURE\_INFO 406

ENV\_INST\_INFO 415

ENV\_PROD\_INFO 417

ENV\_SYS\_INFO 418

ENV\_SYS\_RESOURCES 1310

LOCKS\_HELD 1313

LOCKWAIT 1316

LOG\_UTILIZATION 835

LONG\_RUNNING\_SQL

詳細 837

MON\_BP\_UTILIZATION 461

MON\_CONNECTION\_SUMMARY 470

MON\_CURRENT\_SQL 475

MON\_CURRENT\_UOW 476

## 組み込みビュー (続き)

MON\_DB\_SUMMARY 478  
MON\_LOCKWAITS 758  
MON\_PKG\_CACHE\_SUMMARY 760  
MON\_SERVICE\_SUBCLASS\_SUMMARY 769  
MON\_TBSP\_UTILIZATION 773  
MON\_WORKLOAD\_SUMMARY 778  
NOTIFICATIONLIST 1227  
OBJECTOWNERS  
    詳細 817  
PDLOGMSGS\_LAST24HOURS 1235  
PRIVILEGES  
    詳細 818  
QUERY\_PREP\_COST  
    詳細 840  
REG\_VARIABLES 1368  
SNAPAGENT 841, 966  
SNAPAGENT\_MEMORY\_POOL 1370, 1466  
SNAPAPPL 853, 970  
SNAPAPPL\_INFO 845, 979  
SNAPBP 862, 987  
SNAPBP\_PART 868, 993  
SNAPCONTAINER 872, 997  
SNAPDB 877, 1001  
SNAPDBM 890, 1014  
SNAPDBM\_MEMORY\_POOL 1405, 1476  
SNAPDB\_MEMORY\_POOL 1397, 1471  
SNAPDETAILLOG 894, 1018  
SNAPDYN\_SQL 898, 1022  
SNAPFCM 904, 1028  
SNAPFCM\_PART 908, 1032  
SNAPHADR 1426, 1480  
SNAPLOCK 1431, 1484  
SNAPLOCKWAIT 1438, 1491  
SNAPSTMT 911, 1035  
SNAPSTORAGE\_PATHS 1447, 1548  
SNAPSUBSECTION 919, 1043  
SNAPSWITCHES 923, 1047  
SNAPTAB 1450, 1552  
SNAPTAB\_REORG 928, 1052  
SNAPTbsp 934, 1058  
SNAPTbsp\_PART 941, 1065  
SNAPTbsp\_QUIESCER 946, 1070  
SNAPTbsp\_RANGE 951, 1075  
SNAPUTIL 956, 1080  
SNAPUTIL\_PROGRESS 960, 1084  
TBSP\_UTILIZATION 1090  
TOP\_DYNAMIC\_SQL  
    詳細 1092

## 組み込みルーチン

権限 2  
コーディングの手法 1  
要約 4

## グループ

グループ・メンバーシップの取得 812

## 更新

DB2 インフォメーション・センター 1595, 1597  
構成パラメーター  
    値の取得 356  
    データベース  
        値の取り出し 388  
SET\_CONFIG プロシーチャーの設定 378  
コマンド  
    プロシーチャーからの呼び出し 25, 1119  
ADD CONTACT 27  
ADD CONTACTGROUP 29  
AUTOCONFIGURE 30  
BACKUP DATABASE 34  
DESCRIBE  
    詳細 44  
DROP CONTACT 58  
DROP CONTACTGROUP 59  
EXPORT 60  
FORCE APPLICATION 73  
GET STMM TUNING 74  
IMPORT 76  
INITIALIZE TAPE 106  
LOAD 108  
PRUNE HISTORY/LOGFILE 156  
PRUNE LOGFILE  
    詳細 156  
QUIESCE DATABASE 158  
QUIESCE TABLESPACES FOR TABLE 160  
REDISTRIBUTE DATABASE PARTITION GROUP 163  
REORG INDEXES/TABLE 172  
RESET ALERT CONFIGURATION 191  
RESET DATABASE CONFIGURATION 193  
RESET DATABASE MANAGER CONFIGURATION 195  
REWIND TAPE 196  
RUNSTATS 197  
SET TAPE POSITION 211  
UNQUIESCE DATABASE 212  
UPDATE ALERT CONFIGURATION 213  
UPDATE CONTACT 219  
UPDATE CONTACTGROUP 220  
UPDATE DATABASE CONFIGURATION 222  
UPDATE DATABASE MANAGER CONFIGURATION 225  
UPDATE HEALTH NOTIFICATION CONTACT LIST 227  
UPDATE HISTORY 228  
UPDATE STMM TUNING 230

## ご利用条件

資料 1600  
コンプリート・モード 338

## [サ行]

### 再検証

プロシーチャー 287

### システム情報

検索 371, 418

使用すべきでない SNAPSHOT\_AGENT 表関数 1498



使用すべきでない SNAPSHOT\_APPL 表関数 1500  
 使用すべきでない SNAPSHOT\_APPL\_INFO 表関数 1506  
 使用すべきでない SNAPSHOT\_BP 表関数 1508  
 使用すべきでない SNAPSHOT\_CONTAINER 表関数 1511  
 使用すべきでない SNAPSHOT\_DATABASE 表関数 1512  
 使用すべきでない SNAPSHOT\_DBM 表関数 1519  
 使用すべきでない SNAPSHOT\_DYN\_SQL 表関数 1521  
 使用すべきでない SNAPSHOT\_FCM 表関数 1523  
 使用すべきでない SNAPSHOT\_FCMNODE 表関数 1525  
 使用すべきでない SNAPSHOT\_FILEW プロシーチャー 1526  
 使用すべきでない SNAPSHOT\_LOCK 表関数 1527  
 使用すべきでない SNAPSHOT\_LOCKWAIT 表関数 1529  
 使用すべきでない SNAPSHOT\_QUIESCERS 表関数 1530  
 使用すべきでない SNAPSHOT\_RANGES 表関数 1532  
 使用すべきでない SNAPSHOT\_STATEMENT 表関数 1533  
 使用すべきでない SNAPSHOT\_SUBSECT 表関数 1536  
 使用すべきでない SNAPSHOT\_SWITCHES 表関数 1538  
 使用すべきでない SNAPSHOT\_TABLE 表関数 1540  
 使用すべきでない SNAPSHOT\_TBREORG 表関数 1541  
 使用すべきでない SNAPSHOT\_TBS 表関数 1543  
 使用すべきでない SNAPSHOT\_TBS\_CFG 表関数 1546  
 使用すべきでない SNAP\_GET\_CONTAINER\_V91 表関数 1394  
 使用すべきでない SNAP\_GET\_STO\_PATHS 表関数 1445  
 使用すべきでない SNAP\_GET\_TAB\_V91 表関数 1454  
 使用すべきでない SNAP\_GET\_TBSP\_V91 表関数 1461  
 使用すべきでない SQLCACHE\_SNAPSHOT 表関数 1556  
 使用すべきでない SYSINSTALLROUTINES プロシーチャー  
 1557

使用すべきでない機能

管理ビュー

SNAPDB\_MEMORY\_POOL 1397, 1471  
 SNAPSTORAGE\_PATHS 1447, 1548  
 SNAPTAB 1450, 1552

表関数

ADMIN\_GET\_TAB\_COMPRESS\_INFO 1261  
 ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 1267  
 GET\_DBM\_CONFIG 1309  
 SNAPSHOT\_AGENT 1498  
 SNAPSHOT\_APPL 1500  
 SNAPSHOT\_APPL\_INFO 1506  
 SNAPSHOT\_BP 1508  
 SNAPSHOT\_CONTAINER 1511  
 SNAPSHOT\_DATABASE 1512  
 SNAPSHOT\_DBM 1519  
 SNAPSHOT\_DYN\_SQL 1521  
 SNAPSHOT\_FCM 1523  
 SNAPSHOT\_FCMNODE 1525  
 SNAPSHOT\_LOCK 1527  
 SNAPSHOT\_LOCKWAIT 1529  
 SNAPSHOT\_QUIESCERS 1530  
 SNAPSHOT\_RANGES 1532  
 SNAPSHOT\_STATEMENT 1533  
 SNAPSHOT\_SUBSECT 1536  
 SNAPSHOT\_SWITCHES 1538  
 SNAPSHOT\_TABLE 1540  
 SNAPSHOT\_TBREORG 1541

使用すべきでない機能 (続き)

表関数 (続き)

SNAPSHOT\_TBS 1543  
 SNAPSHOT\_TBS\_CFG 1546  
 SNAP\_GET\_APPL\_V95 1382  
 SNAP\_GET\_BP\_V95 1390  
 SNAP\_GET\_CONTAINER\_V91 1394  
 SNAP\_GET\_DBM\_V95 1402  
 SNAP\_GET\_DB\_MEMORY\_POOL 1397, 1471  
 SNAP\_GET\_DB\_V97 1409  
 SNAP\_GET\_DETAILLOG\_V91 1420  
 SNAP\_GET\_DYN\_SQL\_V95 1422  
 SNAP\_GET\_STORAGE\_PATHS\_V97 1447, 1548  
 SNAP\_GET\_STO\_PATHS 1445  
 SNAP\_GET\_TAB 1450, 1552  
 SNAP\_GET\_TAB\_V91 1454  
 SNAP\_GET\_TBSP\_PART\_V97 1457  
 SNAP\_GET\_TBSP\_V91 1461  
 SQLCACHE\_SNAPSHOT 1556  
 WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 1565  
 WLM\_GET\_SERVICE\_CLASS\_WORKLOAD  
 \_OCCURRENCES\_V97 1574  
 WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 1577

プロシーチャー

GET\_DB\_CONFIG 1307  
 HEALTH\_CONT\_HI 1320  
 HEALTH\_CONT\_HI\_HIS 1322  
 HEALTH\_CONT\_INFO 1324  
 HEALTH\_DBM\_HI 1341  
 HEALTH\_DBM\_HI\_HIS 1343  
 HEALTH\_DBM\_INFO 1345  
 HEALTH\_DB\_HI 1326  
 HEALTH\_DB\_HIC 1334  
 HEALTH\_DB\_HIC\_HIS 1336  
 HEALTH\_DB\_HI\_HIS 1330  
 HEALTH\_DB\_INFO 1339  
 HEALTH\_GET\_ALERT\_ACTION\_CFG 1347  
 HEALTH\_GET\_ALERT\_CFG 1351  
 HEALTH\_GET\_IND\_DEFINITION 1355  
 HEALTH\_HI\_REC 1357  
 HEALTH\_TBS\_HI 1359  
 HEALTH\_TBS\_HI\_HIS 1362  
 HEALTH\_TBS\_INFO 1366  
 SNAPSHOT\_FILEW 1526  
 SYSINSTALLROUTINES 1557  
 WLM\_GET\_WORKLOAD\_OCCURRENCE  
 \_ACTIVITIES\_V97 1583  
 WLM\_GET\_WORKLOAD\_STATS\_V97 1588

SQL 管理ルーチン 1253

資料

印刷 1592  
 概要 1591  
 使用に関するご利用条件 1600  
 PDF ファイル 1592

スカラー関数

AUTH\_GET\_INSTANCE\_AUTHID 806

スカラー関数 (続き)

SQLERRM 1248

スキーマ

オブジェクト 232

コピー 232

ドロップ 237

ストアード・プロシージャー

AUDIT\_ARCHIVE 325

AUDIT\_DELIM\_EXTRACT 326

GET\_ROUTINE\_NAME プロシージャー 1096

SET\_MAINT\_MODE\_RECORD\_NO\_TEMPORALHISTORY 1248

ストレージ管理ツール

ストアード・プロシージャー 1115, 1117, 1119

スプリット・ミラー

データベース・パスの検索 1219

設計アドバイザー

DESIGN\_ADVISOR ストアード・プロシージャー 348

## [夕行]

チュートリアル

トラブルシューティング 1600

問題判別 1600

リスト 1599

pureXML 1599

通知リスト

連絡先リストの検索 1227

通知ログ・メッセージ

検索 1235

データ再配分

プロシージャー 1105, 1107, 1108, 1111, 1113

データベース・パス

検索 1219

データベース・マネージャ構成パラメーター

値の取り出し 392

特記事項 1603

特権

付与されたものに関する情報

PRIVILEGES 管理ビュー 818

トラブルシューティング

オンライン情報 1600

チュートリアル 1600

## [ハ行]

パッケージ

再バインド

REBIND\_ROUTINE\_PACKAGE プロシージャー 1101

ビュー

管理ビュー

ADMINTABCOMPRESSINFO (使用すべきでない) 1261

ADMINTABINFO 294

ADMINTEMPCOLUMNS 304

ADMINTEMPTABLES 308

ADMIN\_TASK\_LIST 317

ビュー (続き)

管理ビュー (続き)

ADMIN\_TASK\_STATUS 320

APPLICATIONS 821

APPL\_PERFORMANCE 820

AUTHORIZATIONIDS 816

BP\_HITRATIO 827

BP\_READ\_IO 829

BP\_WRITE\_IO 831

CONTACTGROUPS 1212

CONTACTS 1213

CONTAINER\_UTILIZATION 833

DB2\_CF 400

DB2\_CLUSTER\_HOST\_STATE 396

DB2\_INSTANCE\_ALERTS 399

DB2\_MEMBER 400

DBCFCG 388

DBMCFG 392

DBPATHS 1219

DB\_HISTORY 1214

ENV\_FEATURE\_INFO 406

ENV\_INST\_INFO 415

ENV\_PROD\_INFO 417

ENV\_SYS\_INFO 418

ENV\_SYS\_RESOURCES 1310

LOCKS\_HELD 1313

LOCKWAIT 1316

LOG\_UTILIZATION 835

LONG\_RUNNING\_SQL 837

MON\_BP\_UTILIZATION 461

MON\_CONNECTION\_SUMMARY 470

MON\_CURRENT\_SQL 475

MON\_CURRENT\_UOW 476

MON\_DB\_SUMMARY 478

MON\_LOCKWAITS 758

MON\_PKG\_CACHE\_SUMMARY 760

MON\_SERVICE\_SUBCLASS\_SUMMARY 769

MON\_TBSP\_UTILIZATION 773

MON\_WORKLOAD\_SUMMARY 778

NOTIFICATIONLIST 1227

OBJECTOWNERS 817

PDLOGMSG\_LAST24HOURS 1235

PRIVILEGES 818

QUERY\_PREP\_COST 840

REG\_VARIABLES 1368

SNAPAGENT 841, 966

SNAPAGENT\_MEMORY\_POOL 1370, 1466

SNAPAPPL 853, 970

SNAPAPPL\_INFO 845, 979

SNAPBP 862, 987

SNAPBP\_PART 868, 993

SNAPCONTAINER 872, 997

SNAPDB 877, 1001

SNAPDBM 890, 1014

SNAPDBM\_MEMORY\_POOL 1405, 1476

SNAPDB\_MEMORY\_POOL 1397, 1471

## ビュー (続き)

## 管理ビュー (続き)

SNAPDETAILLOG 894, 1018  
 SNAPDYN\_SQL 898, 1022  
 SNAPFCM 904, 1028  
 SNAPFCM\_PART 908, 1032  
 SNAPHADR 1426, 1480  
 SNAPLOCK 1431, 1484  
 SNAPLOCKWAIT 1438, 1491  
 SNAPSTMT 911, 1035  
 SNAPSTORAGE\_PATHS 1447, 1548  
 SNAPSUBSECTION 919, 1043  
 SNAPSWITCHES 923, 1047  
 SNAPTAB 1450, 1552  
 SNAPTAB\_REORG 928, 1052  
 SNAPTbsp 934, 1058  
 SNAPTbsp\_PART 941, 1065  
 SNAPTbsp\_QUIESCER 946, 1070  
 SNAPTbsp\_RANGE 951, 1075  
 SNAPUTIL 956, 1080  
 SNAPUTIL\_PROGRESS 960, 1084  
 TBSP\_UTILIZATION 1090  
 TOP\_DYNAMIC\_SQL 1092

## 表

## 移動、オンライン

ADMIN\_MOVE\_TABLE プロシージャ 263  
 ADMIN\_MOVE\_TABLE\_UTIL プロシージャ 284

## 情報の検索

一時表 308  
 一時表の列情報 304  
 サイズ 294, 1280  
 状態 294, 1280

## 表関数

組み込みビューの比較 3  
 組み込みルーチン 4  
 使用すべきでない機能  
 要約 1253  
 ADMIN\_GET\_TAB\_COMPRESS\_INFO 1261  
 ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 1267  
 ADMIN\_GET\_TAB\_INFO\_V95 1273  
 ADMIN\_GET\_TAB\_INFO\_V97 1280  
 SNAP\_GET\_APPL\_INFO\_V95 1375  
 SNAP\_GET\_BP\_V95 1390  
 SNAP\_GET\_DBM\_V95 1402  
 SNAP\_GET\_DETAIL\_LOG\_V91 1420  
 SNAP\_GET\_DYN\_SQL\_V95 1422  
 admin\_get\_dbp\_mem\_usage 1259  
 ADMIN\_GET\_INDEX\_COMPRESS\_INFO 241  
 ADMIN\_GET\_INDEX\_INFO 245  
 admin\_get\_mem\_usage 251  
 ADMIN\_GET\_MSGS 252  
 ADMIN\_GET\_STORAGE\_PATHS 254  
 ADMIN\_GET\_TAB\_COMPRESS\_INFO 256  
 ADMIN\_GET\_TAB\_DICTIONARY\_INFO 258  
 ADMIN\_GET\_TAB\_INFO 294  
 ADMIN\_GET\_TEMP\_COLUMNS 304

## 表関数 (続き)

ADMIN\_GET\_TEMP\_TABLES 308  
 ADMIN\_LIST\_DB\_PATHS 1219  
 AUDIT\_ARCHIVE 325  
 AUTH\_LIST\_GROUPS\_FOR\_AUTHID 812  
 DB2\_GET\_CLUSTER\_HOST\_STATE  
 詳細 396  
 DB2\_GET\_INSTANCE\_INFO  
 詳細 400  
 DB\_GET\_CFG 388  
 ENV\_GET\_DB2\_SYSTEM\_RESOURCES 407  
 ENV\_GET\_NETWORK\_RESOURCES 408  
 ENV\_GET\_REG\_VARIABLES 410  
 ENV\_GET\_SYSTEM\_RESOURCES 412  
 HEALTH\_GET\_ALERT\_ACTION\_CFG 1347  
 HEALTH\_GET\_ALERT\_CFG 1351  
 HEALTH\_GET\_IND\_DEFINITION 1355  
 MON\_FORMAT\_LOCK\_NAME 482  
 MON\_GET\_APPL\_LOCKWAIT 526  
 MON\_GET\_LOCKS 598  
 MON\_GET\_MEMORY\_POOL 604  
 MON\_GET\_MEMORY\_SET 607  
 MON\_GET\_SECTION\_ROUTINE 635, 647, 659, 666  
 PD\_GET\_DIAG\_HIST 1228  
 PD\_GET\_LOG\_MSGS 1235  
 SNAP\_GET\_AGENT 841, 966  
 SNAP\_GET\_AGENT\_MEMORY\_POOL 1370, 1466  
 SNAP\_GET\_APPL 853, 970  
 SNAP\_GET\_APPL\_INFO 845, 979  
 SNAP\_GET\_BP 862, 987  
 SNAP\_GET\_BP\_PART 868, 993  
 SNAP\_GET\_CONTAINER 872, 997  
 SNAP\_GET\_DB 877, 1001  
 SNAP\_GET\_DBM 890, 1014  
 SNAP\_GET\_DBM\_MEMORY\_POOL 1405, 1476  
 SNAP\_GET\_DB\_MEMORY\_POOL 1397, 1471  
 SNAP\_GET\_DB\_V97 1409  
 SNAP\_GET\_DETAIL\_LOG 894, 1018  
 SNAP\_GET\_DETAIL\_LOG\_V91 1420  
 SNAP\_GET\_DYN\_SQL 898, 1022  
 SNAP\_GET\_FCM 904, 1028  
 SNAP\_GET\_FCM\_PART 908, 1032  
 SNAP\_GET\_HADR 1426, 1480  
 SNAP\_GET\_LOCK 1431, 1484  
 SNAP\_GET\_LOCKWAIT 1438, 1491  
 SNAP\_GET\_STMT 911, 1035  
 SNAP\_GET\_STORAGE\_PATHS\_V97 1447, 1548  
 SNAP\_GET\_SUBSECTION 919, 1043  
 SNAP\_GET\_SWITCHES 923, 1047  
 SNAP\_GET\_TAB 1450, 1552  
 SNAP\_GET\_TAB\_REORG 928, 1052  
 SNAP\_GET\_TBSP 934, 1058  
 SNAP\_GET\_TBSP\_PART 941, 1065  
 SNAP\_GET\_TBSP\_PART\_V97 1457  
 SNAP\_GET\_TBSP\_QUIESCER 946, 1070  
 SNAP\_GET\_TBSP\_RANGE 951, 1075

表関数 (続き)

SNAP\_GET\_UTIL 956, 1080  
SNAP\_GET\_UTIL\_PROGRESS 960, 1084  
WLM\_GET\_SERVICE\_CLASS\_AGENTS 1169  
WLM\_GET\_SERVICE\_CLASS\_WORKLOAD  
\_OCCURRENCES 1178  
WLM\_GET\_SERVICE\_SUBCLASS\_STATS 1181  
WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 1577  
WLM\_GET\_WORKLOAD\_STATS 1196

表の圧縮

情報 1261

プロシージャ

共通 SQL API

概要 335

出力のフィルタリングのための XPath 式 339

使用すべきでない機能 1253

要約 4

ADMIN\_CMD

詳細 25

ADMIN\_COPY\_SCHEMA 232

ADMIN\_DROP\_SCHEMA 237

ADMIN\_MOVE\_TABLE 263

ADMIN\_MOVE\_TABLE\_UTIL 284

ADMIN\_REMOVE\_MSGS 287

ADMIN\_REVALIDATE\_DB\_OBJECTS 287

ADMIN\_SET\_INTRA\_PARALLEL 292

ADMIN\_SET\_MAINT\_MODE 293

ADMIN\_TASK\_ADD 311

ADMIN\_TASK\_REMOVE 319

ADMIN\_TASK\_UPDATE 322

ALTER\_ROUTINE\_PACKAGE 1094

ALTOBJ 1206

AM\_DROP\_TASK 1291

AM\_GET\_LOCK\_CHNS 1294

AM\_GET\_LOCK\_CHN\_TB 1292

AM\_GET\_LOCK\_RPT 1294

AM\_GET\_RPT 1302

AM\_SAVE\_TASK 1304

ANALYZE\_LOG\_SPACE 1105

AUDIT\_ARCHIVE 325

AUDIT\_DELIM\_EXTRACT 326

AUTOMAINT\_GET\_POLICY 328

AUTOMAINT\_GET\_POLICYFILE 330

AUTOMAINT\_SET\_POLICY 331

AUTOMAINT\_SET\_POLICYFILE 333

CANCEL\_WORK 341

CAPTURE\_STORAGEMGMT\_INFO 1115

CREATE\_STORAGEMGMT\_TABLES 1117

DESIGN\_ADVISOR 348

DROP\_STORAGEMGMT\_TABLES 1119

EVMON\_FORMAT\_UE\_TO\_TABLES 442

EVMON\_UPGRADE\_TABLES 456

EXPLAIN\_FROM\_ACTIVITY 427

EXPLAIN\_FROM\_CATALOG 431

EXPLAIN\_FROM\_DATA 433

EXPLAIN\_FROM\_SECTION 436

プロシージャ (続き)

GENERATE\_DISTFILE 1107  
GET\_CONFIG 356  
GET\_DBSIZE\_INFO 1224  
GET\_MESSAGE 363  
GET\_ROUTINE\_NAME プロシージャ 1096  
GET\_ROUTINE\_SAR 1098  
GET\_SWRD\_SETTINGS 1108  
GET\_SYSTEM\_INFO 371  
HEALTH\_HI\_REC 1357  
MON\_INCREMENT\_INTERVAL\_ID 757  
PUT\_ROUTINE\_SAR 1100  
REBIND\_ROUTINE\_PACKAGE 1101  
REORGCHK\_IX\_STATS 1243  
REORGCHK\_TB\_STATS 1246  
SET\_CONFIG 378  
SET\_MAINT\_MODE\_RECORD\_NO\_TEMPORALHISTORY 1248  
SET\_ROUTINE\_OPTS 1104  
SET\_SWRD\_SETTINGS 1111  
SNAPSHOT\_FILEW 1526  
SNAP\_WRITE\_FILE 964, 1088  
STEPWISE\_REDISTRIBUTE\_DBPG 1113  
SYSINSTALLOBJECTS 1251  
SYSINSTALLROUTINES 1557  
SYSTS\_ADMIN\_CMD 1119  
SYSTS\_CREATE 1136  
SYSTS\_START 1133  
WLM\_CANCEL\_ACTIVITY 1159  
WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS 1160  
WLM\_COLLECT\_STATS 1162  
WLM\_SET\_CLIENT\_INFO 1199  
WLM\_SET\_CONN\_ENV 1202

ヘルス・アラート

アラート構成 1351

アラート・アクション構成 1347

ヘルス・インディケータ

定義の検索 1355

ヘルプ

SQL ステートメント 1595

## [マ行]

モニター

ルーチン 439

問題判別

チュートリアル 1600

通知ログ・メッセージ 1235

利用できる情報 1600

## [ラ行]

ラッチの待機

MON\_GET\_EXTENDED\_LATCH\_WAIT 表関数 577

履歴ファイル

情報の検索 1214

ルーチン  
 許可 2  
 モニター 439  
 SQL  
 管理 (使用すべきでない) 1253  
 組み込み (要約) 4  
 レジストリー変数  
 使用中の設定の検索 410, 1368  
 連絡先  
 連絡先グループのリストの検索 1212  
 連絡先リストの検索 1213

## A

ADD CONTACT コマンド  
 ADMIN\_CMD プロシーチャーの使用 27  
 ADD CONTACTGROUP コマンド  
 ADMIN\_CMD プロシーチャーの使用 29  
 ADMINTABCOMPRESSINFO 管理ビュー (使用すべきでない)  
 1261  
 ADMINTABINFO 管理ビュー 294  
 ADMINTEMPCOLUMNS 管理ビュー 304  
 ADMINTEMPTABLES 管理ビュー 308  
 ADMIN\_CMD プロシーチャー  
 コマンド  
 ADD CONTACT 27  
 ADD CONTACTGROUP 29  
 AUTOCONFIGURE 30  
 BACKUP DATABASE 34  
 DESCRIBE 44  
 DROP CONTACT 58  
 DROP CONTACTGROUP 59  
 EXPORT 60  
 FORCE APPLICATION 73  
 GET STMM TUNING 74  
 IMPORT 76  
 INITIALIZE TAPE 106  
 LOAD 108  
 PRUNE HISTORY/LOGFILE 156  
 QUIESCE DATABASE 158  
 QUIESCE TABLESPACES FOR TABLE 160  
 REDISTRIBUTE DATABASE PARTITION GROUP 163  
 REORG INDEXES/TABLE 172  
 RESET ALERT CONFIGURATION 191  
 RESET DATABASE CONFIGURATION 193  
 RESET DATABASE MANAGER CONFIGURATION 195  
 REWIND TAPE 196  
 RUNSTATS 197  
 SET TAPE POSITION 211  
 UNQUIESCE DATABASE 212  
 UPDATE ALERT CONFIGURATION 213  
 UPDATE CONTACT 219  
 UPDATE CONTACTGROUP 220  
 UPDATE DATABASE CONFIGURATION 222

ADMIN\_CMD プロシーチャー (続き)  
 コマンド (続き)  
 UPDATE DATABASE MANAGER CONFIGURATION 225  
 UPDATE HEALTH NOTIFICATION CONTACT LIST 227  
 UPDATE HISTORY 228  
 UPDATE STMM TUNING 230  
 詳細 25  
 メッセージ  
 検索 252  
 除去 287  
 ADMIN\_COPY\_SCHEMA プロシーチャー  
 詳細 232  
 ADMIN\_DROP\_SCHEMA プロシーチャー 237  
 ADMIN\_EST\_INLINE\_LENGTH 関数 239  
 ADMIN\_GET\_DBP\_MEM\_USAGE 表関数 1259  
 ADMIN\_GET\_INDEX\_COMPRESS\_INFO 表関数 241  
 ADMIN\_GET\_INDEX\_INFO 表関数 245  
 ADMIN\_GET\_INTRA\_PARALLEL スカラー関数 250  
 ADMIN\_GET\_MEM\_USAGE 表関数 251  
 ADMIN\_GET\_MSGS 表関数 252  
 ADMIN\_GET\_STORAGE\_PATHS 表関数 254  
 ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数 256  
 ADMIN\_GET\_TAB\_COMPRESS\_INFO 表関数 (使用すべきでない) 1261  
 ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 表関数 1267  
 ADMIN\_GET\_TAB\_DICTIONARY\_INFO 表関数 258  
 ADMIN\_GET\_TAB\_INFO 表関数 294  
 ADMIN\_GET\_TAB\_INFO\_V95 表関数 1273  
 ADMIN\_GET\_TAB\_INFO\_V97 表関数  
 詳細 1280  
 ADMIN\_GET\_TEMP\_COLUMNS 表関数 304  
 ADMIN\_GET\_TEMP\_TABLES 表関数 308  
 ADMIN\_IS\_INLINED 関数 262  
 ADMIN\_LIST\_DB\_PATHS 表関数 1219  
 ADMIN\_MOVE\_TABLE プロシーチャー 263  
 ADMIN\_MOVE\_TABLE\_UTIL プロシーチャー 284  
 ADMIN\_REMOVE\_MSGS プロシーチャー 287  
 ADMIN\_REVALIDATE\_DB\_OBJECTS プロシーチャー 287  
 ADMIN\_SET\_INTRA\_PARALLEL プロシーチャー 292  
 ADMIN\_SET\_MAINT\_MODE プロシーチャー 293  
 ADMIN\_TASK\_ADD プロシーチャー 311  
 ADMIN\_TASK\_LIST 管理ビュー 317  
 ADMIN\_TASK\_REMOVE プロシーチャー 319  
 ADMIN\_TASK\_STATUS 管理ビュー 320  
 ADMIN\_TASK\_UPDATE プロシーチャー 322  
 ALTER\_ROUTINE\_PACKAGE プロシーチャー 1094  
 ALTOBJ プロシーチャー 1206  
 AM\_BASE\_RPTS 表関数 1290  
 AM\_BASE\_RPT\_RECOMS 表関数 1289  
 AM\_DROP\_TASK プロシーチャー 1291  
 AM\_GET\_LOCK\_CHNS プロシーチャー 1294  
 AM\_GET\_LOCK\_CHN\_TB プロシーチャー 1292  
 AM\_GET\_LOCK\_RPT プロシーチャー 1294  
 AM\_GET\_RPT プロシーチャー 1302

AM\_SAVE\_TASK プロシージャー 1304  
ANALYZE\_LOG\_SPACE プロシージャー 1105  
APPLICATIONS 管理ビュー 821  
APPLICATION\_ID スカラー関数 1305  
APPL\_PERFORMANCE 管理ビュー 820  
AUDIT\_ARCHIVE ストアード・プロシージャーおよび表関数  
325  
AUDIT\_DELIM\_EXTRACT ストアード・プロシージャー 326  
AUDIT\_LIST\_LOGS 表関数 327  
AUTHORIZATIONIDS 管理ビュー 816  
AUTH\_GET\_INSTANCE\_AUTHID スカラー関数 806  
AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID 表関数 808  
AUTH\_LIST\_GROUPS\_FOR\_AUTHID 表関数 812  
AUTH\_LIST\_ROLES\_FOR\_AUTHID 関数 813  
AUTOCONFIGURE コマンド  
ADMIN\_CMD プロシージャーの使用 30  
AUTOMAINT\_GET\_POLICY ストアード・プロシージャー  
328  
AUTOMAINT\_GET\_POLICYFILE ストアード・プロシージャー  
330  
AUTOMAINT\_SET\_POLICY ストアード・プロシージャー 331  
AUTOMAINT\_SET\_POLICYFILE ストアード・プロシージャー  
333

## B

BACKUP DATABASE コマンド  
ADMIN\_CMD プロシージャーの使用 34  
BP\_HITRATIO 管理ビュー 827  
BP\_READ\_IO 管理ビュー 829  
BP\_WRITE\_IO 管理ビュー 831

## C

CANCEL\_WORK ストアード・プロシージャー 341  
CAPTURE\_STORAGEMGMT\_INFO プロシージャー 1115  
COMPILATION\_ENV 表関数 1209  
CONTACTGROUPS 管理ビュー 1212  
CONTACTS 管理ビュー 1213  
CONTAINER\_UTILIZATION 管理ビュー 833  
CREATE\_STORAGEMGMT\_TABLES プロシージャー 1117

## D

DB2 pureScale インスタンス  
ホスト  
情報 395  
DB2 インフォメーション・センター  
更新 1595, 1597  
バージョン 1595  
DB2\_CF 管理ビュー  
詳細 400  
DB2\_CLUSTER\_HOST\_STATE 管理ビュー  
詳細 396

DB2\_GET\_CLUSTER\_HOST\_STATE 表関数  
詳細 396  
DB2\_GET\_INSTANCE\_INFO 表関数  
詳細 400  
DB2\_INSTANCE\_ALERTS 管理ビュー  
詳細 399  
DB2\_MEMBER 管理ビュー  
詳細 400  
DBCFG 管理ビュー 388  
DBMCFG 管理ビュー 392  
DBPATHS 管理ビュー 1219  
DB\_GET\_CFG 表関数 388  
DB\_HISTORY 管理ビュー  
詳細 1214  
DB\_MEMBERS 表関数 395  
DB\_PARTITIONS 表関数 1306  
DESCRIBE コマンド  
詳細 44  
DESIGN\_ADVISOR プロシージャー 348  
DROP CONTACT コマンド  
詳細  
ADMIN\_CMD の使用 58  
DROP CONTACTGROUP コマンド  
詳細  
ADMIN\_CMD の使用 59  
DROP\_STORAGEMGMT\_TABLES プロシージャー 1119

## E

ENV\_CF\_SYS\_RESOURCES 管理ビュー  
詳細 404  
ENV\_FEATURE\_INFO 管理ビュー 406  
ENV\_GET\_DB2\_SYSTEM\_RESOURCES 407  
ENV\_GET\_NETWORK\_RESOURCES 408  
ENV\_GET\_REG\_VARIABLES 410  
ENV\_GET\_SYSTEM\_RESOURCES 表関数 412  
ENV\_INST\_INFO 管理ビュー 415  
ENV\_PROD\_INFO 管理ビュー 417  
ENV\_SYS\_INFO 管理ビュー 418  
ENV\_SYS\_RESOURCES 管理ビュー 1310  
EVMON\_FORMAT\_UE\_TO\_TABLES プロシージャー  
詳細 442  
EVMON\_FORMAT\_UE\_TO\_XML 表関数  
詳細 453  
EVMON\_UPGRADE\_TABLES プロシージャー 456  
EXPLAIN\_FORMAT\_STATS スカラー関数 422  
EXPLAIN\_FROM\_ACTIVITY プロシージャー 427  
EXPLAIN\_FROM\_CATALOG プロシージャー 431  
EXPLAIN\_FROM\_DATA プロシージャー 433  
EXPLAIN\_FROM\_SECTION プロシージャー  
詳細 436  
EXPLAIN\_GET\_MSGS 表関数 420  
EXPORT コマンド  
詳細  
ADMIN\_CMD の使用 60

## F

FORCE APPLICATION コマンド  
ADMIN\_CMD の使用 73

## G

GENERATE\_DISTFILE プロシージャ 1107  
GET STMM TUNING コマンド 74  
GET\_CONFIG ストアード・プロシージャ 356  
GET\_DBM\_CONFIG 表関数 1309  
GET\_DBSIZE\_INFO プロシージャ 1224  
GET\_DB\_CONFIG 表関数 1307  
GET\_MESSAGE ストアード・プロシージャ 363  
GET\_ROUTINE\_NAME プロシージャのストアード・プロシ  
ージャ 1096  
GET\_ROUTINE\_OPTS スカラー関数 1098  
GET\_ROUTINE\_SAR プロシージャ 1098  
GET\_SWRD\_SETTINGS プロシージャ 1108  
GET\_SYSTEM\_INFO ストアード・プロシージャ 371

## H

HEALTH\_CONT\_HI 表関数 1320  
HEALTH\_CONT\_HI\_HIS 表関数 1322  
HEALTH\_CONT\_INFO 表関数 1324  
HEALTH\_DBM\_HI 表関数 1341  
HEALTH\_DBM\_HI\_HIS 表関数 1343  
HEALTH\_DBM\_INFO 表関数 1345  
HEALTH\_DB\_HI 表関数 1326  
HEALTH\_DB\_HIC 表関数 1334  
HEALTH\_DB\_HIC\_HIS 表関数 1336  
HEALTH\_DB\_HI\_HIS 表関数 1330  
HEALTH\_DB\_INFO 表関数 1339  
HEALTH\_GET\_ALERT\_ACTION\_CFG 表関数 1347  
HEALTH\_GET\_ALERT\_CFG 表関数 1351  
HEALTH\_GET\_IND\_DEFINITION 表関数 1355  
HEALTH\_HI\_REC プロシージャ 1357  
HEALTH\_TBS\_HI 表関数 1359  
HEALTH\_TBS\_HI\_HIS 表関数 1362  
HEALTH\_TBS\_INFO 表関数 1366

## I

IMPORT コマンド  
詳細  
ADMIN\_CMD プロシージャの使用 76  
INITIALIZE TAPE コマンド  
ADMIN\_CMD プロシージャの使用 106

## L

LOAD コマンド  
詳細  
ADMIN\_CMD プロシージャの使用 108

LOCKS\_HELD 管理ビュー 1313  
LOCKWAIT 管理ビュー 1316  
LOG\_UTILIZATION 管理ビュー 835  
LONG\_RUNNING\_SQL 管理ビュー 837

## M

MON\_BP\_UTILIZATION 管理ビュー 461  
MON\_CONNECTION\_SUMMARY 管理ビュー 470  
MON\_CURRENT\_SQL 管理ビュー 475  
MON\_CURRENT\_UOW 管理ビュー 476  
MON\_DB\_SUMMARY 管理ビュー 478  
MON\_FORMAT\_LOCK\_NAME 表関数 482  
MON\_FORMAT\_XML\_COMPONENT\_TIMES\_BY\_ROW 表関数  
詳細 485  
MON\_FORMAT\_XML\_METRICS\_BY\_ROW 表関数  
詳細 489  
MON\_FORMAT\_XML\_TIMES\_BY\_ROW 表関数  
詳細 502  
MON\_FORMAT\_XML\_WAIT\_TIMES\_BY\_ROW 表関数  
詳細 508  
MON\_GET\_ACTIVITY\_DETAILS 表関数 512  
MON\_GET\_APPLICATION\_HANDLE スカラー関数 530  
MON\_GET\_APPLICATION\_ID スカラー関数 531  
MON\_GET\_APPL\_LOCKWAIT 表関数 526  
MON\_GET\_AUTO\_MAINT\_QUEUE 表関数 532  
MON\_GET\_AUTO\_RUNSTATS\_QUEUE 表関数 535  
MON\_GET\_BUFFERPOOL 表関数 537  
MON\_GET\_CF 表関数  
詳細 544  
MON\_GET\_CF\_CMD 表関数 546  
MON\_GET\_CF\_WAIT\_TIME 表関数 550  
MON\_GET\_CONNECTION 表関数 553  
MON\_GET\_CONNECTION\_DETAILS 表関数 563  
MON\_GET\_CONTAINER 表関数 574  
MON\_GET\_EXTENDED\_LATCH\_WAIT 表関数 577  
MON\_GET\_EXTENT\_MOVEMENT\_STATUS 表関数 578  
MON\_GET\_FCM 表関数 580  
MON\_GET\_FCM\_CONNECTION\_LIST 表関数 582  
MON\_GET\_GROUP\_BUFFERPOOL 表関数  
詳細 583  
MON\_GET\_HADR 表関数  
詳細 584  
MON\_GET\_INDEX 表関数 593  
MON\_GET\_INDEX\_USAGE\_LIST 表関数 596  
MON\_GET\_LOCKS 表関数 598  
MON\_GET\_MEMORY\_POOL 表関数 604  
MON\_GET\_MEMORY\_SET 表関数 607  
MON\_GET\_PAGE\_ACCESS  
詳細 610  
MON\_GET\_PKG\_CACHE\_STMT 表関数  
詳細 612  
MON\_GET\_PKG\_CACHE\_STMT\_DETAILS 表関数  
詳細 623  
MON\_GET\_REBALANCE\_STATUS 表関数  
詳細 633

MON\_GET\_RTS\_RQST 表関数 664  
MON\_GET\_SECTION\_ROUTINE 表関数 635, 647, 659, 666  
MON\_GET\_SERVERLIST 表関数 668  
MON\_GET\_SERVICE\_SUBCLASS 表関数 670  
MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 表関数 683  
MON\_GET\_TABLE 表関数 693  
MON\_GET\_TABLESPACE 表関数 697  
MON\_GET\_TABLE\_USAGE\_LIST 表関数 706  
MON\_GET\_TRANSACTION\_LOG 表関数 710  
MON\_GET\_UNIT\_OF\_WORK 表関数 712  
MON\_GET\_UNIT\_OF\_WORK\_DETAILS 表関数 723  
MON\_GET\_USAGE\_LIST\_STATUS 表関数 735  
MON\_GET\_WORKLOAD 表関数 737  
MON\_GET\_WORKLOAD\_DETAILS 表関数 747  
MON\_INCREMENT\_INTERVAL\_ID ストアード・プロシージャ  
ー 757  
MON\_LOCKWAITS 管理ビュー 758  
MON\_PKG\_CACHE\_SUMMARY 管理ビュー 760  
MON\_SAMPLE\_SERVICE\_CLASS\_METRICS 表関数 762  
MON\_SAMPLE\_WORKLOAD\_METRICS 表関数 766  
MON\_SERVICE\_SUBCLASS\_SUMMARY 管理ビュー 769  
MON\_TBSP\_UTILIZATION 管理ビュー 773  
MON\_WORKLOAD\_SUMMARY 管理ビュー 778  
MQPUBLISH スカラー関数 783  
MQREAD スカラー関数 785  
MQREADALL 表関数 787  
MQREADALLCLOB 表関数 789  
MQREADCLOB スカラー関数 791  
MQRECEIVE スカラー関数 793  
MQRECEIVEALL 表関数 794  
MQRECEIVEALLCLOB 表関数 797  
MQRECEIVECLOB スカラー関数 799  
MQSEND スカラー関数 801  
MQSUBSCRIBE スカラー関数 803  
MQUNSUBSCRIBE スカラー関数 805

## N

NOTIFICATIONLIST 管理ビュー 1227

## O

OBJECTOWNERS 管理ビュー 817

## P

PDLOGMSG\_LAST24HOURS 管理ビュー 1235  
PD\_GET\_DIAG\_HIST 表関数 1228  
PD\_GET\_LOG\_MSGS 表関数 1235  
PRIVILEGES 管理ビュー 818  
PRUNE HISTORY/LOGFILE コマンド  
ADMIN\_CMD の使用 156  
PUT\_ROUTINE\_SAR プロシージャ 1100

## Q

QUERY\_PREP\_COST 管理ビュー 840  
QUIESCE DATABASE コマンド 158  
QUIESCE TABLESPACES FOR TABLE コマンド  
ADMIN\_CMD の使用 160

## R

REBIND\_ROUTINE\_PACKAGE プロシージャ 1101  
REDISTRIBUTE DATABASE PARTITION GROUP コマンド  
ADMIN\_CMD の使用 163  
REG\_VARIABLES 管理ビュー 1368  
REORG INDEXES コマンド  
ADMIN\_CMD の使用 172  
REORG TABLE コマンド  
ADMIN\_CMD の使用 172  
REORGCHK\_IX\_STATS プロシージャ 1243  
REORGCHK\_TB\_STATS プロシージャ 1246  
RESET ALERT CONFIGURATION コマンド  
ADMIN\_CMD の使用 191  
RESET DATABASE CONFIGURATION コマンド  
ADMIN\_CMD の使用 193  
RESET DATABASE MANAGER CONFIGURATION コマンド  
ADMIN\_CMD の使用 195  
REWIND TAPE コマンド  
ADMIN\_CMD の使用 196  
RUNSTATS コマンド  
詳細  
ADMIN\_CMD の使用 197

## S

SET TAPE POSITION コマンド  
ADMIN\_CMD の使用 211  
SET\_CONFIG ストアード・プロシージャ 378  
SET\_MAINT\_MODE\_RECORD\_NO\_TEMPORALHISTORY プロ  
シージャ 1248  
SET\_ROUTINE\_OPTS プロシージャ 1104  
SET\_SWRD\_SETTINGS プロシージャ 1111  
SNAPAGENT 管理ビュー 841, 966  
SNAPAGENT\_MEMORY\_POOL 管理ビュー 1370, 1466  
SNAPAPPL 管理ビュー 853, 970  
SNAPAPPL\_INFO 管理ビュー 845, 979  
SNAPBP 管理ビュー 862, 987  
SNAPBP\_PART 管理ビュー 868, 993  
SNAPCONTAINER 管理ビュー 872, 997  
SNAPDB 管理ビュー 877, 1001  
SNAPDBM 管理ビュー 890, 1014  
SNAPDBM\_MEMORY\_POOL 管理ビュー 1405, 1476  
SNAPDB\_MEMORY\_POOL 管理ビュー 1397, 1471  
SNAPDETAILLOG 管理ビュー 894, 1018  
SNAPDYN\_SQL 管理ビュー 898, 1022  
SNAPFCM 管理ビュー 904, 1028  
SNAPFCM\_PART 管理ビュー 908, 1032



SNAPHADR 管理ビュー  
詳細 1426, 1480  
SNAPLOCK 管理ビュー 1431, 1484  
SNAPLOCKWAIT 管理ビュー 1438, 1491  
SNAPSTMT 管理ビュー 911, 1035  
SNAPSTORAGE\_PATHS 管理ビュー 1447, 1548  
SNAPSUBSECTION 管理ビュー 919, 1043  
SNAPSWITCHES 管理ビュー 923, 1047  
SNAPTAB\_REORG 管理ビュー 928, 1052  
SNAPTbsp 管理ビュー 934, 1058  
SNAPTbsp\_PART 管理ビュー 941, 1065  
SNAPTbsp\_QUIESCER 管理ビュー 946, 1070  
SNAPTbsp\_RANGE 管理ビュー 951, 1075  
SNAPUTIL 管理ビュー 956, 1080  
SNAPUTIL\_PROGRESS 管理ビュー 960, 1084  
SNAP\_GET\_AGENT 表関数 841, 966  
SNAP\_GET\_AGENT\_MEMORY\_POOL 表関数 1370, 1466  
SNAP\_GET\_APPL 表関数 853, 970  
SNAP\_GET\_APPL\_INFO 表関数 845, 979  
SNAP\_GET\_APPL\_INFO\_V95 表関数 1375  
SNAP\_GET\_BP 表関数 862, 987  
SNAP\_GET\_BP\_PART 表関数 868, 993  
SNAP\_GET\_CONTAINER 表関数 872, 997  
SNAP\_GET\_DB 表関数 877, 1001  
SNAP\_GET\_DBM 表関数 890, 1014  
SNAP\_GET\_DBM\_MEMORY\_POOL 表関数 1405, 1476  
SNAP\_GET\_DB\_MEMORY\_POOL 表関数 1397, 1471  
SNAP\_GET\_DB\_V97 表関数 1409  
SNAP\_GET\_DETAIL\_LOG 表関数 894, 1018  
SNAP\_GET\_DETAIL\_LOG\_V91 表関数 1420  
SNAP\_GET\_DYN\_SQL 表関数 898, 1022  
SNAP\_GET\_DYN\_SQL\_V95 表関数 1422  
SNAP\_GET\_FCM 表関数 904, 1028  
SNAP\_GET\_FCM\_PART 表関数 908, 1032  
SNAP\_GET\_HADR 表関数  
詳細 1426, 1480  
SNAP\_GET\_LOCK 表関数 1431, 1484  
SNAP\_GET\_LOCKWAIT 表関数 1438, 1491  
SNAP\_GET\_STMT 表関数 911, 1035  
SNAP\_GET\_STORAGE\_PATHS\_V97 表関数 1447, 1548  
SNAP\_GET\_SUBSECTION 表関数 919, 1043  
SNAP\_GET\_SWITCHES 表関数 923, 1047  
SNAP\_GET\_TAB 表関数 1450, 1552  
SNAP\_GET\_TAB\_REORG 表関数 928, 1052  
SNAP\_GET\_TBSP 表関数 934, 1058  
SNAP\_GET\_TBSP\_PART 表関数 941, 1065  
SNAP\_GET\_TBSP\_PART\_V97 表関数 1457  
SNAP\_GET\_TBSP\_QUIESCER 表関数 946, 1070  
SNAP\_GET\_TBSP\_RANGE 表関数 951, 1075  
SNAP\_GET\_UTIL 表関数 956, 1080  
SNAP\_GET\_UTIL\_PROGRESS 表関数 960, 1084  
SNAP\_WRITE\_FILE プロシージャ 964, 1088

## SQL

管理ルーチン

使用すべきでない 1253

SQL ステートメント

ヘルプ

表示 1595

SQLCODE

戻りメッセージ情報 363

SQLERRM スカラー関数 1248

STEPWISE\_REDISTRIBUTE\_DBPG プロシージャ

詳細 1113

SYSINSTALLOBJECTS プロシージャ 1251

SYSTS\_ADMIN\_CMD プロシージャ 1119

SYSTS\_ALTER ストアード・プロシージャ 1121

SYSTS\_CLEANUP ストアード・プロシージャ 1127

SYSTS\_CLEAR\_COMMANDLOCKS ストアード・プロシージャ  
ー 1129

SYSTS\_CLEAR\_EVENTS ストアード・プロシージャ 1131

SYSTS\_CREATE プロシージャ 1136

SYSTS\_DISABLE プロシージャ 1145

SYSTS\_DROP ストアード・プロシージャ 1147

SYSTS\_ENABLE ストアード・プロシージャ 1150

SYSTS\_START プロシージャ 1133

SYSTS\_UPDATE ストアード・プロシージャ 1152

SYSTS\_UPGRADE\_CATALOG ストアード・プロシージャ  
1155

SYSTS\_UPGRADE\_INDEX ストアード・プロシージャ 1157

## T

TBSP\_UTILIZATION 管理ビュー 1090

TOP\_DYNAMIC\_SQL 管理ビュー 1092

## U

UNQUIESCE DATABASE コマンド

ADMIN\_CMD の使用 212

UPDATE ALERT CONFIGURATION コマンド

ADMIN\_CMD の使用 213

UPDATE CONTACT コマンド

ADMIN\_CMD の使用 219

UPDATE CONTACTGROUP コマンド

ADMIN\_CMD の使用 220

UPDATE DATABASE CONFIGURATION コマンド

ADMIN\_CMD の使用 222

UPDATE DATABASE MANAGER CONFIGURATION コマン  
ド

ADMIN\_CMD の使用 225

UPDATE HEALTH NOTIFICATION CONTACT LIST コマンド

ADMIN\_CMD の使用 227

UPDATE HISTORY コマンド

ADMIN\_CMD の使用 228

UPDATE STMM TUNING コマンド

ADMIN\_CMD の使用 230

## W

WLM\_CANCEL\_ACTIVITY プロシージャー 1159  
WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS プロシージャー  
1160  
WLM\_COLLECT\_STATS プロシージャー  
詳細 1162  
WLM\_GET\_ACTIVITY\_DETAILS 表関数 1558  
WLM\_GET\_CONN\_ENV 表関数 1164  
WLM\_GET\_QUEUE\_STATS 表関数 1166  
WLM\_GET\_SERVICE\_CLASS\_AGENTS 表関数  
詳細 1169  
WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 表関数 1565  
WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES  
表関数  
詳細 1178  
WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES  
\_V97 表関数 1574  
WLM\_GET\_SERVICE\_SUBCLASS\_STATS 表関数  
詳細 1181  
WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 表関数 1577  
WLM\_GET\_SERVICE\_SUPERCLASS\_STATS 表関数 1188  
WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 表関数  
説明 1191  
WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97 表  
関数 1583  
WLM\_GET\_WORKLOAD\_STATS 表関数 1196  
WLM\_GET\_WORKLOAD\_STATS\_V97 表関数 1588  
WLM\_GET\_WORK\_ACTION\_SET\_STATS 表関数  
詳細 1190  
WLM\_SET\_CLIENT\_INFO プロシージャー 1199  
WLM\_SET\_CONN\_ENV プロシージャー 1202

## X

### XML

共通 SQL API 入力 338

### XML 文書

出力文書

共通 SQL API のバージョン管理 336





Printed in Japan

SA88-4672-01



日本アイ・ビー・エム株式会社  
〒103-8510 東京都中央区日本橋箱崎町19-21

Spine information:

IBM DB2 10.1 for Linux, UNIX, and Windows

管理ルーチンおよびビュー

