



## 관리 루틴 및 뷰





주:

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 1193 페이지의 부록 B 『주의사항』의 정보를 읽으십시오.

#### 개정판 주의사항

이 문서에는 IBM에서 소유하고 있는 정보가 있습니다. 이는 라이선스 계약에 따라 제공한 것이며 저작권의 보호를 받습니다. 이 책의 정보에는 제품 보증이 포함되지 않으며, 이 매뉴얼에서 제공된 어떠한 문장도 이와 같이 해석할 수 없습니다.

온라인으로 IBM 서적을 주문하거나 로컬 IBM 담당자를 통해 서적을 주문할 수 있습니다.

- 온라인으로 서적을 주문하려면 IBM Publications Center([www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order))로 이동하십시오.
- 로컬 IBM 담당자를 찾으려면 IBM Directory of Worldwide Contacts([www.ibm.com/planetwide](http://www.ibm.com/planetwide))로 이동하십시오.

미국 또는 캐나다의 DB2 Marketing and Sales에서 DB2 서적을 주문하려면 1-800-IBM-4YOU (426-4968)로 전화하십시오.

IBM은 귀하가 IBM으로 보낸 정보를 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 사용하거나 배포할 수 있습니다.

# 목차

제 1 장 시스템 정의 루틴 및 뷰 . . . . . 1  
 관리 뷰에 대한 권한 부여 . . . . . 1  
 관리 뷰 대 테이블 함수 . . . . . 2

제 2 장 지원되는 시스템 정의 SQL 루틴 및 뷰 . . . 3

제 3 장 활동 모니터 루틴 . . . . . 17  
 AM\_BASE\_RPT\_RECOMS - 활동 보고서에 대한  
 권장사항 . . . . . 17  
 AM\_BASE\_RPTS - 활동 모니터 보고서 . . . . . 18  
 AM\_DROP\_TASK - 모니터링 태스크 삭제 . . . . . 19  
 AM\_GET\_LOCK\_CHN\_TB - 응용프로그램 잠금  
 체인 데이터를 표 형식으로 검색 . . . . . 20  
 AM\_GET\_LOCK\_CHNS - 특정 응용프로그램에  
 대한 잠금 체인 정보 검색 . . . . . 21  
 AM\_GET\_LOCK\_RPT - 응용프로그램 잠금 세부  
 사항 검색 . . . . . 22  
 AM\_GET\_RPT - 활동 모니터 데이터 검색 . . . . . 30  
 AM\_SAVE\_TASK - 모니터링 태스크 작성 또는  
 수정 . . . . . 32

제 4 장 ADMIN\_CMD 프로시저 및 관련 루틴 . . . 35  
 ADMIN\_CMD - 관리 명령 실행 . . . . . 35  
 ADMIN\_CMD 프로시저를 사용하는 ADD  
 CONTACT 명령 . . . . . 37  
 ADMIN\_CMD 프로시저를 사용하는 ADD  
 CONTACTGROUP 명령 . . . . . 39  
 ADMIN\_CMD 프로시저를 사용하는  
 AUTOCONFIGURE 명령 . . . . . 40  
 ADMIN\_CMD 프로시저를 사용하는 BACKUP  
 DATABASE 명령 . . . . . 44  
 ADMIN\_CMD 프로시저를 사용하는  
 DESCRIBE 명령 . . . . . 53  
 ADMIN\_CMD 프로시저를 사용하는 DROP  
 CONTACT 명령 . . . . . 68  
 ADMIN\_CMD 프로시저를 사용하는 DROP  
 CONTACTGROUP 명령 . . . . . 69  
 ADMIN\_CMD 프로시저를 사용하는 EXPORT  
 명령 . . . . . 70  
 ADMIN\_CMD 프로시저를 사용하는 FORCE  
 APPLICATION 명령 . . . . . 82  
 ADMIN\_CMD 프로시저를 사용한 GET STMM  
 TUNING DBPARTITIONNUM 명령 . . . . . 84

ADMIN\_CMD 프로시저를 사용하는 IMPORT  
 명령 . . . . . 85  
 ADMIN\_CMD 프로시저를 사용하는  
 INITIALIZE TAPE 명령 . . . . . 113  
 ADMIN\_CMD 프로시저를 사용하는 LOAD 명  
 령 . . . . . 114  
 ADMIN\_CMD 프로시저를 사용하는 PRUNE  
 HISTORY/LOGFILE 명령 . . . . . 156  
 ADMIN\_CMD 프로시저를 사용하는 QUIESCE  
 DATABASE 명령 . . . . . 158  
 ADMIN\_CMD 프로시저를 사용하는 QUIESCE  
 TABLESPACES FOR TABLE 명령 . . . . . 160  
 ADMIN\_CMD 프로시저를 사용하는  
 REDISTRIBUTE DATABASE PARTITION  
 GROUP 명령 . . . . . 163  
 ADMIN\_CMD 프로시저를 사용하는 REORG  
 INDEXES/TABLE 명령 . . . . . 175  
 ADMIN\_CMD 프로시저를 사용하는 RESET  
 ALERT CONFIGURATION 명령 . . . . . 190  
 ADMIN\_CMD 프로시저를 사용하는 RESET  
 DATABASE CONFIGURATION 명령 . . . . . 192  
 ADMIN\_CMD 프로시저를 사용하는 RESET  
 DATABASE MANAGER  
 CONFIGURATION 명령 . . . . . 194  
 ADMIN\_CMD 프로시저를 사용하는 REWIND  
 TAPE 명령 . . . . . 195  
 ADMIN\_CMD 프로시저를 사용하는  
 RUNSTATS 명령 . . . . . 196  
 ADMIN\_CMD 프로시저를 사용하는 SET  
 TAPE POSITION 명령 . . . . . 210  
 ADMIN\_CMD 프로시저를 사용하는  
 UNQUIESCE DATABASE 명령 . . . . . 211  
 ADMIN\_CMD 프로시저를 사용하는 UPDATE  
 ALERT CONFIGURATION 명령 . . . . . 212  
 ADMIN\_CMD 프로시저를 사용하는 UPDATE  
 CONTACT 명령 . . . . . 219  
 ADMIN\_CMD 프로시저를 사용하는 UPDATE  
 CONTACTGROUP 명령 . . . . . 220  
 ADMIN\_CMD 프로시저를 사용하는 UPDATE  
 DATABASE CONFIGURATION 명령 . . . . . 221

ADMIN_CMD 프로시저를 사용하는 UPDATE DATABASE MANAGER	
CONFIGURATION 명령 . . . . .	225
ADMIN_CMD 프로시저를 사용하는 UPDATE HEALTH NOTIFICATION CONTACT LIST 명령 . . . . .	227
ADMIN_CMD 프로시저를 사용하는 UPDATE HISTORY 명령 . . . . .	228
ADMIN_CMD 프로시저를 사용하는 UPDATE STMM TUNING DBPARTITIONNUM 명령 . . . . .	231
ADMIN_EST_INLINE_LENGTH 함수 - 데이터 인라인에 필요한 길이 추정. . . . .	232
ADMIN_GET_DBP_MEM_USAGE 테이블 함수 - 인스턴스에 대한 전체 메모리 소비 가져오기 . . . . .	234
ADMIN_GET_INDEX_COMPRESS_INFO 테이블 함수 - 압축된 인덱스 정보 리턴. . . . .	236
ADMIN_GET_INDEX_INFO 테이블 함수 - 인덱스 정보 리턴 . . . . .	240
ADMIN_GET_MSGS 테이블 함수 - ADMIN_CMD 프로시저에서 실행된 데이터 이동 유틸리티에서 생성된 메시지 검색. . . . .	244
ADMIN_IS_INLINED 함수 - 데이터가 인라인인지 판별 . . . . .	246
ADMIN_REMOVE_MSGS 프로시저 - ADMIN_CMD 프로시저에서 실행된 데이터 이동 유틸리티에서 생성된 메시지 정리. . . . .	247
ADMIN_REVALIDATE_DB_OBJECTS 프로시저 - 유효하지 않은 데이터베이스 오브젝트 유효성 다시 확인 . . . . .	248
ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN_GET_TAB_COMPRESS_INFO_V97 테이블 함수 - 압축 정보 리턴 . . . . .	251
ADMINTABINFO 관리 뷰 및 ADMIN_GET_TAB_INFO_V97 테이블 함수 - 테이블 크기 및 상태 정보 검색 . . . . .	260
ADMINTEMPCOLUMNS 관리 뷰 및 ADMIN_GET_TEMP_COLUMNS 테이블 함수 - 임시 테이블의 컬럼 정보 검색. . . . .	268
ADMINTEMPTABLES 관리 뷰 및 ADMIN_GET_TEMP_TABLES 테이블 함수 - 임시 테이블에 대한 정보 검색 . . . . .	272
<b>제 5 장 관리 태스크 스케줄러 루틴 및 뷰 . . . . .</b>	<b>277</b>
ADMIN_TASK_ADD 프로시저 - 새 태스크 스케줄 . . . . .	277
UNIX 크론 형식 . . . . .	281

ADMIN_TASK_LIST 관리 뷰 - 스케줄러에서 태스크 정보 검색. . . . .	283
ADMIN_TASK_REMOVE 프로시저 - 스케줄된 태스크 또는 태스크 상태 레코드 제거 . . . . .	284
ADMIN_TASK_STATUS 관리 뷰 - 태스크 상태 정보 검색 . . . . .	286
ADMIN_TASK_UPDATE 프로시저 - 기존 태스크 갱신 . . . . .	288
<b>제 6 장 감사 루틴 및 프로시저 . . . . .</b>	<b>293</b>
AUDIT_ARCHIVE 프로시저 및 테이블 함수 - 아카이브 감사 로그 파일 . . . . .	293
AUDIT_DELIM_EXTRACT - 구분 파일로 추출 수행 . . . . .	294
AUDIT_LIST_LOGS 테이블 함수 - 아카이브된 감사 로그 파일 나열. . . . .	295
<b>제 7 장 자동 유지보수 루틴 . . . . .</b>	<b>297</b>
AUTOMAINT_GET_POLICY 프로시저 - 자동 유지보수 규정 검색 . . . . .	297
AUTOMAINT_GET_POLICYFILE 프로시저 - 자동 유지보수 규정 검색 . . . . .	298
AUTOMAINT_SET_POLICY 프로시저 - 자동 유지보수 규정 구성 . . . . .	299
AUTOMAINT_SET_POLICYFILE 프로시저 - 자동 유지보수 규정 구성 . . . . .	301
<b>제 8 장 일반 SQL API 프로시저 . . . . .</b>	<b>303</b>
일반 입력 및 출력 매개변수 . . . . .	303
XML 문서 버전화 . . . . .	304
XML 입력 문서 . . . . .	305
유효한 XML 입력 문서 리턴을 위한 완료 모드	306
XML 출력 문서 . . . . .	306
출력 필터링을 위한 XPath 표현식 . . . . .	307
XML 메시지 문서 . . . . .	308
CANCEL_WORK 프로시저 - 작업 취소. . . . .	309
GET_CONFIG 프로시저 - 구성 데이터 가져오기	316
GET_MESSAGE 프로시저 - 메시지 텍스트 가져오기 . . . . .	323
GET_SYSTEM_INFO 프로시저 - 시스템 정보 가져오기. . . . .	331
SET_CONFIG 프로시저 - 구성 매개변수 설정	338
<b>제 9 장 구성 루틴 및 뷰 . . . . .</b>	<b>349</b>
DB_PARTITIONS . . . . .	349
DBCFG 관리 뷰 - 데이터베이스 구성 매개변수 정보 검색 . . . . .	350

DBCFCG 관리 뷰 - 데이터베이스 관리 프로그램 구성 매개변수 정보 검색 . . . . .	352
REG_VARIABLES 관리 뷰 - 사용 중인 DB2 레지스트리 설정 검색 . . . . .	355
<b>제 10 장 환경 뷰.</b> . . . . .	357
ENV_FEATURE_INFO 관리 뷰 - DB2 기능에 대한 라이선스 정보 리턴 . . . . .	357
ENV_INST_INFO 관리 뷰 - 현재 인스턴스에 대한 정보 검색 . . . . .	358
ENV_PROD_INFO 관리 뷰 - 설치된 DB2 제품에 대한 정보 검색. . . . .	360
ENV_SYS_INFO 관리 뷰 - 시스템에 대한 정보 검색 . . . . .	361
ENV_SYS_RESOURCES 관리 뷰 - 시스템 정보 리턴 . . . . .	362
<b>제 11 장 모니터 루틴</b> . . . . .	367
EVMON_FORMAT_UE_TO_TABLES 프로시저 - XML 문서를 관계형 테이블로 이동 . . . . .	368
EVMON_FORMAT_UE_TO_XML 테이블 함수 - 비형식화 이벤트를 XML로 변환 . . . . .	376
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기 . . . . .	379
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기 . . . . .	388
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기 . . . . .	391
MON_GET_CONNECTION_DETAILS 테이블 함수 - 연결 메트릭 세부사항 가져오기. . . . .	396
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기 . . . . .	401
MON_GET_EXTENT_MOVEMENT_STATUS - Extent 이동 진행 가져오기. . . . .	405
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 가져오기 . . . . .	406
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시에서 SQL문 활동 메트릭 가져오기 . . . . .	409
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기 . . . . .	414
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 서비스 서브클래스 메트릭 세부사항 가져오기 . . . . .	420
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 가져오기 . . . . .	426
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 테이블 메트릭 가져오기. . . . .	429

MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기 . . . . .	433
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 작업 단위(UOW) 메트릭 세부사항 가져오기	438
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기 . . . . .	444
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 워크로드 메트릭 세부사항 가져오기 . . . . .	449
<b>제 12 장 MQSeries 루틴.</b> . . . . .	457
MQPUBLISH . . . . .	457
MQREAD . . . . .	459
MQREADALL . . . . .	460
MQREADALLCLOB . . . . .	462
MQREADCLOB . . . . .	464
MQRECEIVE . . . . .	465
MQRECEIVEALL . . . . .	467
MQRECEIVEALLCLOB . . . . .	469
MQRECEIVECLOB. . . . .	471
MQSEND . . . . .	473
MQSUBSCRIBE . . . . .	474
MQUNSUBSCRIBE. . . . .	476
<b>제 13 장 보안 루틴 및 뷰.</b> . . . . .	479
AUTH_LIST_AUTHORITIES_FOR_AUTHID	479
AUTH_LIST_GROUPS_FOR_AUTHID 테이블 함수 - 주어진 권한 부여 ID에 대한 그룹 멤버십 목록 검색 . . . . .	483
AUTH_LIST_ROLES_FOR_AUTHID 함수 - 역할 목록 리턴 . . . . .	484
AUTHORIZATIONIDS 관리 뷰 - 권한 부여 ID 및 유형 검색 . . . . .	487
OBJECTOWNERS 관리 뷰 - 오브젝트 소유권 정보 검색 . . . . .	488
PRIVILEGES 관리 뷰 - 특권 정보 검색. . . . .	489
<b>제 14 장 스냅샷 루틴 및 뷰</b> . . . . .	491
APPL_PERFORMANCE 관리 뷰 - 응용프로그램에 대해 선택된 행의 백분을 검색 . . . . .	491
APPLICATIONS 관리 뷰 - 연결된 데이터베이스 응용프로그램 정보 검색 . . . . .	492
BP_HITRATIO 관리 뷰 - 버퍼 풀 사용 비율 정보 검색 . . . . .	497
BP_READ_IO 관리 뷰 - 버퍼 풀 읽기 성능 정보 검색 . . . . .	499
BP_WRITE_IO 관리 뷰 - 버퍼 풀 쓰기 성능 정보 검색 . . . . .	501



CONTAINER_UTILIZATION 관리 뷰 - 테이블 스페이스 컨테이너 및 활용 정보 검색 . . . . .	503
LOCKS_HELD 관리 뷰 - 보유된 잠금에 대한 정 보 검색 . . . . .	505
LOCKWAITS 관리 뷰 - 현재 잠금 대기 정보 검 색 . . . . .	508
LOG_UTILIZATION 관리 뷰 - 로그 사용도 정보 검색 . . . . .	513
LONG_RUNNING_SQL 관리 뷰 . . . . .	514
QUERY_PREP_COST 관리 뷰 - 명령문 준비 시 간 정보 검색 . . . . .	517
SNAPAGENT 관리 뷰 및 SNAP_GET_AGENT 테이블 함수 - 에이전트 논리 데이터 그룹 응용프 로그램 스냅샷 정보 검색 . . . . .	518
SNAPAGENT_MEMORY_POOL 관리 뷰 및 SNAP_GET_AGENT_MEMORY_POOL 테이블 함수 - memory_pool 논리 데이터 그룹 스냅샷 정 보 검색 . . . . .	522
SNAPAPPL_INFO 관리 뷰 및 SNAP_GET_APPL_INFO_V95 테이블 함수 - Retrieve appl_info 논리 데이터 그룹 스냅샷 정보 .	526
SNAPAPPL 관리 뷰 및 SNAP_GET_APPL_V95 테이블 함수 - appl 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	535
SNAPBP 관리 뷰 및 SNAP_GET_BP_V95 테이 블 함수 - 버퍼 풀 논리 그룹 스냅샷 정보 검색 . .	543
SNAPBP_PART 관리 뷰 및 SNAP_GET_BP_PART 테이블 함수 - bufferpool_nodeinfo 논리 데이터 그룹 스냅샷 정 보 검색 . . . . .	548
SNAPCONTAINER 관리 뷰 및 SNAP_GET_CONTAINER_V91 테이블 함수 - tablespace_container 논리 데이터 그룹 스냅샷 정 보 검색 . . . . .	552
SNAPDB 관리 뷰 및 SNAP_GET_DB_V95 테이 블 함수 - dbase 논리 그룹에서 스냅샷 정보 검색 .	557
SNAPDB_MEMORY_POOL 관리 뷰 및 SNAP_GET_DB_MEMORY_POOL 테이블 함수 - 데이터베이스 레벨 메모리 사용 정보 검색 . . .	568
SNAPDBM 관리 뷰 및 SNAP_GET_DBM_V95 테이블 함수 - dbm 논리적 그룹화 스냅샷 정보 검 색 . . . . .	572
SNAPDBM_MEMORY_POOL 관리 뷰 및 SNAP_GET_DBM_MEMORY_POOL 테이블 함 수 - 데이터베이스 관리 프로그램 레벨 메모리 사 용 정보 검색 . . . . .	577

SNAPDETAILLOG 관리 뷰 및 SNAP_GET_DETAILLOG_V91 테이블 함수 - 논 리 데이터 그룹에서 스냅샷 정보 검색 . . . . .	581
SNAPDYN_SQL 관리 뷰 및 SNAP_GET_DYN_SQL_V95 테이블 함수 - dynsql 논리 그룹 스냅샷 정보 검색 . . . . .	584
SNAPFCM 관리 뷰 및 SNAP_GET_FCM 테이블 함수 - fcm 논리 데이터 그룹 스냅샷 정보 검색 .	590
SNAPFCM_PART 관리 뷰 및 SNAP_GET_FCM_PART 테이블 함수 - fcm_node 논리 데이터 그룹 스냅샷 정보 검색 . .	592
SNAPHADR 관리 뷰 및 SNAP_GET_HADR 테 이블 함수 - hadr 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	595
SNAPLOCK 관리 뷰 및 SNAP_GET_LOCK 테 이블 함수 - 잠금 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	600
SNAPLOCKWAIT 관리 뷰 및 SNAP_GET_LOCKWAIT 테이블 함수 - 잠금 대 기 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	606
SNAPSTMT 관리 뷰 및 SNAP_GET_STMT 테 이블 함수 - 명령문 스냅샷 정보 검색 . . . . .	612
SNAPSTORAGE_PATHS 관리 뷰 및 SNAP_GET_STORAGE_PATHS_V97 테이블 함 수 - 자동 스토리지 경로 정보 검색 . . . . .	618
SNAPSUBSECTION 관리 뷰 및 SNAP_GET_SUBSECTION 테이블 함수 - 서브 섹션 논리 모니터 그룹 스냅샷 정보 검색 . . . . .	621
SNAPSWITCHES 관리 뷰 및 SNAP_GET_SWITCHES 테이블 함수 - 데이터베 이스 스냅샷 스위치 상태 정보 검색 . . . . .	625
SNAPTAB 관리 뷰 및 SNAP_GET_TAB_V91 테이블 함수 - 테이블 논리 데이터 그룹 스냅샷 정 보 검색 . . . . .	629
SNAPTAB_REORG 관리 뷰 및 SNAP_GET_TAB_REORG 테이블 함수 - 테이블 재구성 스냅샷 정보 검색 . . . . .	632
SNAPTbsp 관리 뷰 및 SNAP_GET_TBSP_V91 테이블 함수 - 테이블 스페이스 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	638
SNAPTbsp_PART 관리 뷰 및 SNAP_GET_TBSP_PART_V97 테이블 함수 - tablespace_nodeinfo 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	644



SNAPTBSP QUIESCER 관리 뷰 및 SNAP_GET_TBSP QUIESCER 테이블 함수 - Quiescer 테이블 스페이스 스냅샷 정보 검색 . . . . .	649	SNAPDBM_MEMORY_POOL 관리 뷰 및 SNAP_GET_DBM_MEMORY_POOL 테이블 함 수 - 데이터베이스 관리 프로그램 레벨 메모리 사 용 정보 검색 . . . . .	726
SNAPTBSP_RANGE 관리 뷰 및 SNAP_GET_TBSP_RANGE 테이블 함수 - 범위 스냅샷 정보 검색 . . . . .	654	SNAPDETAILLOG 관리 뷰 및 SNAP_GET_DETAILLOG_V91 테이블 함수 - 논 리 데이터 그룹에서 스냅샷 정보 검색 . . . . .	730
SNAPUTIL 관리 뷰 및 SNAP_GET_UTIL 테이 블 함수 - utility_info 논리 데이터 그룹 스냅샷 정 보 검색 . . . . .	658	SNAPDYN_SQL 관리 뷰 및 SNAP_GET_DYN_SQL_V95 테이블 함수 - dynsql 논리 그룹 스냅샷 정보 검색. . . . .	733
SNAPUTIL_PROGRESS 관리 뷰 및 SNAP_GET_UTIL_PROGRESS 테이블 함수 - 진 행 중 논리 데이터 그룹 스냅샷 정보 검색. . . . .	662	SNAPFCM 관리 뷰 및 SNAP_GET_FCM 테이블 함수 - fcm 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	739
SNAP_WRITE_FILE 프로시저 . . . . .	666	SNAPFCM_PART 관리 뷰 및 SNAP_GET_FCM_PART 테이블 함수 - fcm_node 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	741
SNAPAGENT 관리 뷰 및 SNAP_GET_AGENT 테이블 함수 - 에이전트 논리 데이터 그룹 응용프 로그램 스냅샷 정보 검색 . . . . .	667	SNAPHADR 관리 뷰 및 SNAP_GET_HADR 테 이블 함수 - hadr 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	744
SNAPAGENT_MEMORY_POOL 관리 뷰 및 SNAP_GET_AGENT_MEMORY_POOL 테이블 함수 - memory_pool 논리 데이터 그룹 스냅샷 정 보 검색 . . . . .	671	SNAPLOCK 관리 뷰 및 SNAP_GET_LOCK 테 이블 함수 - 잠금 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	749
SNAPAPPL_INFO 관리 뷰 및 SNAP_GET_APPL_INFO_V95 테이블 함수 - Retrieve appl_info 논리 데이터 그룹 스냅샷 정보. . . . .	675	SNAPLOCKWAIT 관리 뷰 및 SNAP_GET_LOCKWAIT 테이블 함수 - 잠금 대 기 논리 데이터 그룹 스냅샷 정보 검색. . . . .	755
SNAPAPPL 관리 뷰 및 SNAP_GET_APPL_V95 테이블 함수 - appl 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	684	SNAPSTMT 관리 뷰 및 SNAP_GET_STMT 테 이블 함수 - 명령문 스냅샷 정보 검색 . . . . .	761
SNAPBP 관리 뷰 및 SNAP_GET_BP_V95 테이 블 함수 - 버퍼 풀 논리 그룹 스냅샷 정보 검색. . . . .	692	SNAPSTORAGE_PATHS 관리 뷰 및 SNAP_GET_STORAGE_PATHS 테이블 함수 - 자동 스토리지 경로 정보 검색. . . . .	767
SNAPBP_PART 관리 뷰 및 SNAP_GET_BP_PART 테이블 함수 - bufferpool_nodeinfo 논리 데이터 그룹 스냅샷 정 보 검색 . . . . .	697	SNAPSUBSECTION 관리 뷰 및 SNAP_GET_SUBSECTION 테이블 함수 - 서버 섹션 논리 모니터 그룹 스냅샷 정보 검색 . . . . .	770
SNAPCONTAINER 관리 뷰 및 SNAP_GET_CONTAINER_V91 테이블 함수 - tablespace_container 논리 데이터 그룹 스냅샷 정 보 검색 . . . . .	701	SNAPSWITCHES 관리 뷰 및 SNAP_GET_SWITCHES 테이블 함수 - 데이터베 이스 스냅샷 스위치 상태 정보 검색. . . . .	774
SNAPDB 관리 뷰 및 SNAP_GET_DB_V95 테이 블 함수 - dbase 논리 그룹에서 스냅샷 정보 검색 . . . . .	706	SNAPTAB 관리 뷰 및 SNAP_GET_TAB_V91 테이블 함수 - 테이블 논리 데이터 그룹 스냅샷 정 보 검색 . . . . .	778
SNAPDB_MEMORY_POOL 관리 뷰 및 SNAP_GET_DB_MEMORY_POOL 테이블 함수 - 데이터베이스 레벨 메모리 사용 정보 검색. . . . .	717	SNAPTAB_REORG 관리 뷰 및 SNAP_GET_TAB_REORG 테이블 함수 - 테이블 재구성 스냅샷 정보 검색 . . . . .	782
SNAPDBM 관리 뷰 및 SNAP_GET_DBM_V95 테이블 함수 - dbm 논리적 그룹화 스냅샷 정보 검 색 . . . . .	721	SNAPTBSP 관리 뷰 및 SNAP_GET_TBSP_V91 테이블 함수 - 테이블 스페이스 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	788

SNAPTBSP_PART 관리 뷰 및 SNAP_GET_TBSP_PART_V91 테이블 함수 - tablespace_nodeinfo 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	794
SNAPTBSP QUIESCER 관리 뷰 및 SNAP_GET_TBSP QUIESCER 테이블 함수 - Quiescer 테이블 스페이스 스냅샷 정보 검색 . . . . .	800
SNAPTBSP_RANGE 관리 뷰 및 SNAP_GET_TBSP_RANGE 테이블 함수 - 범위 스냅샷 정보 검색 . . . . .	805
SNAPUTIL 관리 뷰 및 SNAP_GET_UTIL 테이 블 함수 - utility_info 논리 데이터 그룹 스냅샷 정 보 검색 . . . . .	809
SNAPUTIL_PROGRESS 관리 뷰 및 SNAP_GET_UTIL_PROGRESS 테이블 함수 - 진 행 중 논리 데이터 그룹 스냅샷 정보 검색. . . . .	813
SNAP_WRITE_FILE 프로시저 . . . . .	816
TBSP_UTILIZATION 관리 뷰 - 테이블 스페이스 구성 및 활용 정보 검색. . . . .	818
TOP_DYNAMIC_SQL 관리 뷰 - 상위 동적 SQL 문에 대한 정보 검색. . . . .	822
<b>제 15 장 SQL 프로시저 루틴 . . . . .</b>	<b>825</b>
ALTER_ROUTINE_PACKAGE 프로시저 . . . . .	825
GET_ROUTINE_OPTS . . . . .	826
GET_ROUTINE_SAR . . . . .	826
PUT_ROUTINE_SAR . . . . .	828
REBIND_ROUTINE_PACKAGE 프로시저 - 패키 지 리바인드. . . . .	829
SET_ROUTINE_OPTS. . . . .	831
<b>제 16 장 순차적 재분배 루틴. . . . .</b>	<b>833</b>
ANALYZE_LOG_SPACE 프로시저 - 로그 스페이 스 분석 정보 검색. . . . .	833
GENERATE_DISTFILE 프로시저 - 데이터 분산 파일 생성 . . . . .	835
GET_SWRD_SETTINGS 프로시저 - 재분배 정보 검색 . . . . .	836
SET_SWRD_SETTINGS 프로시저 - 재분배 레지 스트리 작성 또는 변경 . . . . .	839
STEPWISE_REDISTRIBUTE_DBPG 프로시저 - 데이터베이스 파티션 그룹의 일부 재분배 . . . . .	841
<b>제 17 장 스토리지 관리 도구 루틴 . . . . .</b>	<b>845</b>
CAPTURE_STORAGEEMGMT_INFO 프로시저 - 제공된 루트 오브젝트에 대한 스토리지 관련 정보 검색 . . . . .	845

CREATE_STORAGEEMGMT_TABLES 프로시저 - 스토리지 관리 테이블 작성 . . . . .	846
DROP_STORAGEEMGMT_TABLES 프로시저 - 전체 스토리지 관리 테이블 삭제(drop) . . . . .	847
<b>제 18 장 텍스트 검색 루틴 . . . . .</b>	<b>849</b>
SYSTS_ADMIN_CMD 스토어드 프로시저 - 텍스 트 검색 관리 명령 실행. . . . .	849
SYSTS_ALTER 프로시저 - 인덱스의 갱신 특성 변경 . . . . .	850
SYSTS_CLEAR_COMMANDLOCKS 프로시저 - 텍스트 검색 인덱스에 대한 명령 잠금 제거 . . . . .	854
SYSTS_CLEAR_EVENTS 프로시저 - 인덱스의 이벤트 테이블에서 인덱싱 이벤트 삭제. . . . .	856
SYSTS_CREATE 프로시저 - 컬럼에서 텍스트 검 색 인덱스 작성. . . . .	859
SYSTS_DISABLE 프로시저 - 텍스트 검색을 위해 현재 데이터베이스 사용 안함 . . . . .	866
SYSTS_DROP 프로시저 - 텍스트 검색 인덱스 삭 제 . . . . .	869
SYSTS_ENABLE 프로시저 - 텍스트 검색을 위해 현재 데이터베이스 사용 . . . . .	872
SYSTS_UPDATE 프로시저 - 텍스트 검색 인덱스 갱신 . . . . .	873
<b>제 19 장 워크로드 관리 루틴. . . . .</b>	<b>877</b>
WLM_CANCEL_ACTIVITY - 활동 취소 . . . . .	877
WLM_CAPTURE_ACTIVITY_IN_PROGRESS - 활동 이벤트 모니터에 대한 활동 정보 수집 . . . . .	878
WLM_COLLECT_STATS - 워크로드 관리 통계 수집 및 재설정. . . . .	880
WLM_GET_QUEUE_STATS 테이블 함수 - 임계 값 큐 통계 리턴 . . . . .	881
WLM_GET_SERVICE_CLASS_AGENTS_V97 테이블 함수 - 서비스 클래스에서 실행 중인 에이전 트 나열 . . . . .	885
WLM_GET_SERVICE_CLASS_WORKLOAD _OCCURRENCES_V97 - 워크로드 어커런스 나열 . . . . .	893
WLM_GET_SERVICE_SUBCLASS_STATS_V97 테이블 함수 - 서비스 서브클래스의 통계 리턴 . . . . .	897
WLM_GET_SERVICE_SUPERCLASS_STATS - 서비스 수퍼 클래스의 통계 리턴 . . . . .	905
WLM_GET_WORK_ACTION_SET_STATS - 작 업 조치 세트 통계 리턴. . . . .	907
WLM_GET_WORKLOAD_OCCURRENCE _ACTIVITIES_V97 - 활동 목록 리턴. . . . .	909

WLM_GET_WORKLOAD_STATS_V97 테이블 함수 - 워크로드 통계 리턴. . . . .	915
WLM_SET_CLIENT_INFO 프로시저 - 클라이언 트 정보 설정 . . . . .	918
<b>제 20 장 기타 루틴 및 뷰. . . . .</b>	<b>923</b>
ADMIN_COPY_SCHEMA 프로시저 - 특정 스키 마 및 해당 오브젝트 복사 . . . . .	923
ADMIN_DROP_SCHEMA 프로시저 - 특정 스키 마 및 해당 오브젝트 삭제 . . . . .	927
ADMIN_MOVE_TABLE 프로시저 - 온라인 테이블 이동 . . . . .	930
ADMIN_MOVE_TABLE_UTIL 프로시저 - 온라인 이동 테이블 프로시저 수정. . . . .	942
ALTOBJ. . . . .	944
APPLICATION_ID . . . . .	947
COMPILATION_ENV 테이블 함수 - 컴파일 환경 요소 검색 . . . . .	948
CONTACTGROUPS 관리 뷰 - 문의처 그룹 목록 검색 . . . . .	950
CONTACTS 관리 뷰 - 문의처 목록 검색 . . . . .	951
DB_HISTORY 관리 뷰 - 실행기록 파일 정보 검색 . . . . .	952
DBPATHS 관리 뷰 - 데이터베이스 경로 검색 . . . . .	957
EXPLAIN_FORMAT_STATS . . . . .	959
EXPLAIN_GET_MSGS . . . . .	965
GET_DBSIZE_INFO . . . . .	967
NOTIFICATIONLIST 관리 뷰 - Health 통지에 대한 문의처 목록 검색 . . . . .	970
PD_GET_DIAG_HIST - 제공된 기능에서 레코드 리턴 . . . . .	971
PDLOGMSG_LAST24HOURS 관리 뷰 및 PD_GET_LOG_MSGS 테이블 함수 - 문제점 판별 메시지 검색. . . . .	978
REORGCHK_IX_STATS 프로시저 - 재구성 평가에 대한 인덱스 통계 검색 . . . . .	985
REORGCHK_TB_STATS 프로시저 - 재구성 평가에 대한 테이블 통계 검색 . . . . .	987
SQLERRM 스칼라 함수 - 오류 메시지 정보 검색	989
SYSINSTALLOBJECTS . . . . .	991
<b>제 21 장 사용되지 않는 SQL 관리 루틴 및 대체 루틴이나 뷰. . . . .</b>	<b>995</b>
ADMIN_GET_TAB_INFO 테이블 함수 - 테이블에 대한 크기 및 상태 정보 검색. . . . .	997
ADMINTABCOMPRESSINFO 뷰 및 ADMIN_GET_TAB_COMPRESS_INFO . . . . .	1004

GET_DB_CONFIG . . . . .	1009
GET_DBM_CONFIG . . . . .	1010
Health 스냅샷 루틴. . . . .	1011
HEALTH_CONT_HI . . . . .	1011
HEALTH_CONT_HI_HIS . . . . .	1013
HEALTH_CONT_INFO . . . . .	1015
HEALTH_DB_HI . . . . .	1017
HEALTH_DB_HI_HIS . . . . .	1021
HEALTH_DB_HIC . . . . .	1025
HEALTH_DB_HIC_HIS . . . . .	1027
HEALTH_DB_INFO . . . . .	1030
HEALTH_DBM_HI . . . . .	1031
HEALTH_DBM_HI_HIS. . . . .	1033
HEALTH_DBM_INFO . . . . .	1035
HEALTH_GET_ALERT_ACTION_CFG	1036
HEALTH_GET_ALERT_CFG. . . . .	1040
HEALTH_GET_IND_DEFINITION . . . . .	1043
HEALTH_HI_REC. . . . .	1045
HEALTH_TBS_HI. . . . .	1047
HEALTH_TBS_HI_HIS . . . . .	1050
HEALTH_TBS_INFO. . . . .	1054
SNAP_GET_APPL 테이블 함수 - appl 논리 데이터 그룹 스냅샷 정보 검색 . . . . .	1056
SNAP_GET_APPL_INFO 테이블 함수 - appl_info 논리 데이터 그룹 스냅샷 정보 검색. . . . .	1062
SNAP_GET_BP 테이블 함수 - 버퍼 풀 논리 그룹 스냅샷 정보 검색 . . . . .	1069
SNAP_GET_CONTAINER . . . . .	1073
SNAP_GET_DB . . . . .	1074
SNAP_GET_DBM 테이블 함수 - dbm 논리 그룹 스냅샷 정보 검색. . . . .	1080
SNAP_GET_DB_V91 테이블 함수 - dbase 논리적 그룹에서 스냅샷 정보 검색 . . . . .	1083
SNAP_GET_DYN_SQL_V91 테이블 함수 - dynsql 논리적 그룹 스냅샷 정보 검색. . . . .	1093
SNAP_GET_DYN_SQL . . . . .	1097
SNAP_GET_STO_PATHS . . . . .	1099
SNAP_GET_TAB . . . . .	1100
SNAP_GET_TBSP. . . . .	1101
SNAP_GET_TBSP_PART . . . . .	1104
SNAPSHOT_AGENT. . . . .	1106
SNAPSHOT_APPL . . . . .	1107
SNAPSHOT_APPL_INFO . . . . .	1112
SNAPSHOT_BP . . . . .	1114
SNAPSHOT_CONTAINER. . . . .	1116
SNAPSHOT_DATABASE . . . . .	1117

SNAPSHOT_DBM . . . . .	1122	WLM_GET_SERVICE_SUBCLASS_STATS -	
SNAPSHOT_DYN_SQL . . . . .	1124	서비스 서브클래스의 통계 리턴 . . . . .	1165
SNAPSHOT_FCM . . . . .	1126	WLM_GET_WORKLOAD_OCCURRENCE	
SNAPSHOT_FCMNODE . . . . .	1127	_ACTIVITIES - 활동 목록 리턴 . . . . .	1172
SNAPSHOT_FILEW . . . . .	1128	WLM_GET_WORKLOAD_STATS - 워크로드	
SNAPSHOT_LOCK . . . . .	1129	통계 리턴 . . . . .	1177
SNAPSHOT_LOCKWAIT . . . . .	1130		
SNAPSHOT_QUIESCERS . . . . .	1132	<b>부록 A. DB2 기술 정보 개요 . . . . .</b>	1181
SNAPSHOT_RANGES . . . . .	1133	DB2 기술 라이브러리(하드카피 또는 PDF 형식)	1182
SNAPSHOT_STATEMENT . . . . .	1134	인쇄된 DB2 서적 주문 . . . . .	1184
SNAPSHOT_SUBSECT . . . . .	1136	명령행 처리기에서 SQL 상태 도움말 표시 . . .	1185
SNAPSHOT_SWITCHES . . . . .	1137	DB2 정보 센터의 다른 버전에 액세스 . . . . .	1186
SNAPSHOT_TABLE . . . . .	1139	DB2 정보 센터에서 원하는 언어로 항목 표시	1186
SNAPSHOT_TBREORG . . . . .	1140	컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보	
SNAPSHOT_TBS . . . . .	1141	센터 갱신 . . . . .	1187
SNAPSHOT_TBS_CFG . . . . .	1143	컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보	
SQLCACHE_SNAPSHOT . . . . .	1146	센터 수동 갱신 . . . . .	1188
SYSINSTALLROUTINES . . . . .	1147	DB2 지습서 . . . . .	1190
WLM_GET_ACTIVITY_DETAILS - 특정 활동		DB2 문제점 해결 정보 . . . . .	1191
에 대한 세부 정보 리턴 . . . . .	1147	이용약관 . . . . .	1191
WLM_GET_SERVICE_CLASS_AGENTS - 서			
비스 클래스에서 실행 중인 에이전트 나열 . . .	1154	<b>부록 B. 주의사항 . . . . .</b>	1193
WLM_GET_SERVICE_CLASS_WORKLOAD		색인 . . . . .	1197
_OCCURRENCES - 워크로드 어커런스 목록 . .	1161		

---

## 제 1 장 시스템 정의 루틴 및 뷰

시스템 정의 루틴 및 뷰는 SQL을 통해 DB2®를 관리 및 사용하도록 사용하기 쉽게 프로그램화된 기본 인터페이스를 제공합니다. 여기에서는 다양한 DB2 태스크를 수행하는 내장 뷰, 테이블 함수, 프로시저 및 스칼라 함수 컬렉션이 모두 사용됩니다. 예를 들어 시스템 정의 루틴은 테이블 재구성, 모니터 데이터 캡처 및 검색 또는 현재 연결에 대한 응용프로그램 ID 검색에 사용할 수 있습니다.

이 루틴 및 뷰는 SQL 기반 응용프로그램, DB2 명령 또는 명령 스크립트에서 호출할 수 있습니다.

---

### 관리 뷰에 대한 권한 부여

SYSIBMADM 스키마에 있는 모든 관리 뷰에 대해 해당 뷰에 대한 SELECT 특권이 필요합니다. 이는 다음 쿼리를 통해 권한 부여 ID 또는 사용자가 속한 그룹 또는 역할에 SELECT 특권이 있는지 확인하여(즉, 검색 기준을 충족하고 GRANTEE 컬럼에 나열됨) 유효성이 확인됩니다.

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

여기서 <view\_name>은 관리 뷰의 이름입니다.

SYSIBMADM.AUTHORIZATIONIDS, SYSIBMADM.OBJECTOWNERS 및 SYSIBMADM.PRIVILEGES를 제외하고는 기본 관리 테이블 함수에 대해 EXECUTE 특권도 가지고 있어야 합니다. 기본 관리 테이블 함수는 관리 뷰의 권한 부여 섹션에 나열되어 있습니다. 이는 다음 쿼리로 유효성이 확인될 수 있습니다.

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

여기서 <routine\_name>은 문서에 나열된 대로 기본 관리 테이블 함수의 이름입니다.

일부 관리 뷰의 경우 해당 뷰에 대한 SELECT 및 기본 관리 테이블 함수에 대한 EXECUTE 이상의 추가 권한이 필요합니다. 필요한 추가 권한은 해당 뷰를 설명하는 참조 정보에 문서화되어 있습니다.

---

## 관리 뷰 대 테이블 함수

DB2 버전 9.5에서는 SQL을 통해 DB2 관리 함수에 대해 사용하기 쉬운 API를 제공하는 관리 뷰를 소개합니다.

관리 뷰는 다음 세 가지 범주로 나뉘어집니다.

- 카탈로그 뷰에 기초한 뷰
- 입력 매개변수가 없는 테이블 함수에 기초한 뷰
- 하나 이상의 입력 매개변수가 있는 테이블 함수에 기초한 뷰

관리 뷰는 테이블 함수가 추가 정보 또는 성능 이점을 제공하지 않으므로 카탈로그 뷰에 기초한 뷰 및 입력 매개변수가 없는 테이블 함수에 기초한 뷰에 대해 우선적으로 사용되며 유일하게 문서화된 인터페이스입니다.

하나 이상의 입력 매개변수가 있는 테이블 함수에 기초한 관리 뷰의 경우, 관리 뷰 및 테이블 함수 둘 다 사용될 수 있으며 각기 서로 다른 목표를 달성합니다.

- ADMINTABINFO 관리 뷰 및 ADMIN\_GET\_TAB\_INFO\_V95 테이블 함수: 관리 뷰는 데이터베이스에 있는 모든 테이블에 대한 정보를 검색합니다. 대형 데이터베이스의 경우 상당한 성능상의 영향이 있을 수 있습니다. 성능상의 영향은 테이블 함수를 사용하고 스키마 이름, 테이블 이름 또는 둘 다를 입력으로 지정함으로써 줄일 수 있습니다.
- PDLOGMSG\_LAST24HOURS 관리 뷰 및 PD\_GET\_LOG\_MSGS 테이블 함수: 통지 로그 메시지를 검색하는 관리 뷰는 24시간 전의 데이터에 대한 빠른 액세스를 제공하므로 이 테이블 함수를 사용하여 지정된 시간 기간에서 데이터를 검색할 수 있습니다.
- 모든 스냅샷 모니터 관리 뷰 및 테이블 함수(SNAP\* 관리 뷰, SNAP\_GET\_\* 테이블 함수): 스냅샷 모니터 관리 뷰는 각 데이터베이스 파티션의 데이터에 대한 액세스를 제공합니다. 이 테이블 함수는 모든 데이터베이스 파티션에서 수집된 데이터 또는 단일 데이터베이스 파티션의 데이터 사이에서 선택할 수 있는 옵션을 제공합니다.

뷰 대신 테이블 함수를 사용하는 응용프로그램은 변경될 필요가 있을 수 있는데 이는 릴리스가 바뀔 때마다 새 정보가 리턴될 수 있도록 하기 위해 테이블 함수를 변경할 수 있기 때문입니다. 새 테이블 함수는 원래 함수와 동일한 기본 이름을 사용하지만 이들 함수가 추가된 제품의 버전에 ‘\_Vxx’ 접미부가 표시됩니다(예: \_V97). 관리 뷰는 항상 테이블 함수의 가장 현재 버전을 기본으로 하므로 추가 응용프로그램 이식성을 허용합니다. 컬럼이 추후 릴리스로 넘어갈 때마다 다를 수 있으므로, 관리 뷰에서 특정 컬럼을 선택하거나 결과 세트가 SELECT \* 명령문을 응용프로그램에서 사용하는지 여부를 설명하도록 하는 것이 좋습니다.



## 제 2 장 지원되는 시스템 정의 SQL 루틴 및 뷰

다음 테이블에는 지원되는 관리 SQL 루틴 및 뷰에 대한 정보가 요약됩니다.

- 활동 모니터 관리 SQL 루틴: 표 1
- ADMIN\_CMD 스토어드 프로시저 및 연관 관리 SQL 루틴: 4 페이지의 표 2
- 관리 태스크 스케줄러 루틴 및 뷰 5 페이지의 표 3
- 감사 루틴 및 프로시저: 5 페이지의 표 4
- 자동 유지보수 관리 SQL 루틴 및 뷰: 5 페이지의 표 5
- 공통 SQL API 스토어드 프로시저: 5 페이지의 표 6
- 구성 관리 SQL 루틴 및 뷰: 6 페이지의 표 7
- 환경 관리 뷰: 6 페이지의 표 8
- Health 스냅샷 관리 SQL 루틴: 6 페이지의 표 9
- 모니터 관리 SQL 루틴: 8 페이지의 표 10
- MQSeries® 관리 SQL 루틴: 9 페이지의 표 11
- 보안 관리 SQL 루틴 및 뷰: 9 페이지의 표 12
- 스냅샷 관리 SQL 루틴 및 뷰: 10 페이지의 표 13
- SQL 프로시저 관리 SQL 루틴: 12 페이지의 표 14
- Stepwise 재분배 관리 SQL 루틴: 13 페이지의 표 15
- 스토리지 관리 도구 관리 SQL 루틴: 13 페이지의 표 16
- 텍스트 검색 관리 SQL 루틴: 13 페이지의 표 17
- 워크로드 관리 루틴: 14 페이지의 표 18
- 기타 관리 SQL 루틴 및 뷰: 15 페이지의 표 19

표 1. 활동 모니터 관리 SQL 루틴

루틴 이름	스키마	설명
AM_BASE_RPT_RECOMS 테이블 함수	SYSPROC	이 테이블 함수는 활동 모니터에서 사용되는 활동 보고서에 대한 권장사항을 리턴합니다.
AM_BASE_RPTS 테이블 함수	SYSPROC	이 테이블 함수는 활동 모니터에서 사용되는 활동 보고서를 리턴합니다.
AM_DROP_TASK 프로시저	SYSPROC	이 프로시저는 모니터링 태스크를 삭제합니다.
AM_GET_LOCK_CHN_TB 프로시저	SYSPROC	이 프로시저는 응용프로그램 잠금 체인 데이터를 표 형식으로 리턴합니다.
AM_GET_LOCK_CHNS 프로시저	SYSPROC	이 프로시저는 지정한 응용프로그램에 대한 잠금 체인을 출력 문자열을 사용하여 표시합니다.
AM_GET_LOCK_RPT 프로시저	SYSPROC	이 프로시저는 응용프로그램에 대한 잠금 세부사항을 표시합니다.



표 1. 활동 모니터 관리 SQL 루틴 (계속)

루틴 이름	스키마	설명
AM_GET_RPT 프로시저	SYSPROC	이 프로시저는 보고서에 대한 활동 모니터 데이터를 표시합니다.
AM_SAVE_TASK 프로시저	SYSPROC	이 프로시저는 모니터링 태스크를 작성하거나 수정합니다.

표 2. ADMIN\_CMD 스토어드 프로시저 및 연관된 관리 SQL 루틴

루틴 이름	스키마	설명
ADMIN_CMD 프로시저	SYSPROC	이 프로시저를 사용하면 관리자는 CALL문에서 ADMIN_CMD를 실행하여 관리 명령(DB2 명령 행 처리기(CLP) 명령 포함)을 실행할 수 있습니다.
ADMIN_EST_INLINE_LENGTH 함수	SYSIBM	이 함수는 XML 컬럼, BLOB 컬럼, CLOB 컬럼 또는 DBCLOB 컬럼으로 저장된 데이터를 인라인하는 데 필요한 인라인 길이에 대한 예상치를 리턴합니다.
ADMIN_GET_DBP_MEM_USAGE 테이블 함수	SYSPROC	이 테이블 함수는 지정한 인스턴스에 대한 전체 메모리 소비를 가져옵니다.
ADMIN_GET_INDEX_COMPRESS_INFO	SYSPROC	이 테이블 함수는 압축되지 않은 인덱스에 대한 잠재적인 인덱스 압축 절약을 리턴하거나 카탈로그에서 인덱스 압축 통계를 보고합니다.
ADMIN_GET_INDEX_INFO 테이블 함수	SYSPROC	이 테이블 함수는 카탈로그 뷰에서 사용 불가능한 인덱스 정보를 리턴합니다.
ADMIN_GET_MSGS 테이블 함수	SYSPROC	이 테이블 함수를 사용하여 ADMIN_CMD 프로시저에서 실행된 데이터 이동 유틸리티에서 생성된 메시지를 검색합니다.
ADMIN_IS_INLINED 함수	SYSIBM	이 함수는 XML 컬럼, BLOB 컬럼, CLOB 컬럼 또는 DBCLOB 컬럼에 대해 인라인 데이터의 상태 정보를 검색합니다.
ADMIN_REMOVE_MSGS 프로시저	SYSPROC	이 프로시저를 사용하여 ADMIN_CMD 프로시저에서 실행된 데이터 이동 유틸리티에서 생성된 메시지를 제거합니다.
ADMIN_REVALIDATE_DB_OBJECTS 프로시저	SYSPROC	이 프로시저는 유효하지 않은 데이터베이스 오브젝트의 유효성을 다시 확인합니다.
ADMINTABCOMPRESSINFO 뷰 및 ADMIN_GET_TAB_COMPRESS_INFO_V97 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 뷰 및 테이블 함수는 테이블, 구체화된 쿼리 테이블(MQT) 및 계층 구조 테이블에 대한 압축 정보를 리턴합니다.
ADMINTABINFO 및 ADMIN_GET_TAB_INFO_V97	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 뷰 및 테이블 함수는 테이블, 구체화된 쿼리 테이블(MQT) 및 계층 구조 테이블에 대한 크기 및 상태 정보를 리턴합니다.
ADMINTEMPCOLUMNS 뷰 및 ADMIN_GET_TEMP_COLUMNS 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 뷰 및 테이블 함수는 작성된 임시 테이블 및 선언된 임시 테이블에 대한 컬럼 속성 정보를 검색합니다.
ADMINTEMPTABLES 뷰 및 ADMIN_GET_TEMP_TABLES 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 뷰 및 테이블 함수는 작성된 임시 테이블 및 선언된 임시 테이블의 인스턴스에 대한 테이블 속성 및 인스턴스화 시간 정보를 검색합니다.

표 3. 관리 태스크 스케줄러 루틴 및 뷰

루틴 또는 뷰 이름	스키마	설명
ADMIN_TASK_ADD	SYSPROC	이 프로시저는 관리 태스크를 스케줄합니다.
ADMIN_TASK_LIST	SYSTOOLS	이 관리 뷰는 스케줄러에 정의된 각 태스크에 대한 정보를 검색합니다.
ADMIN_TASK_REMOVE	SYSPROC	이 프로시저는 스케줄된 태스크 또는 태스크 상태 레코드를 제거합니다.
ADMIN_TASK_STATUS	SYSTOOLS	이 관리 뷰는 각 태스크 상태 정보를 검색합니다.
ADMIN_TASK_UPDATE	SYSPROC	이 프로시저는 기존 태스크를 갱신합니다.

표 4. 감사 루틴 및 프로시저

루틴 또는 뷰 이름	스키마	설명
AUDIT_ARCHIVE 프로시저 및 테이블 함수	SYSPROC	이 프로시저 및 테이블 함수는 현재 감사 로그를 아카이브합니다.
AUDIT_DELIM_EXTRACT 프로시저	SYSPROC	이 프로시저는 2진 아카이브 로그에서 데이터를 추출하여 구분된 파일로 로드합니다.
AUDIT_LIST_LOGS 테이블 함수	SYSPROC	이 테이블 함수는 현재 데이터베이스에 대해 지정된 경로에서 아카이브된 감사 로그 목록을 리턴합니다.

표 5. 자동 유지보수 관리 SQL 루틴 및 뷰

루틴 또는 뷰 이름	스키마	설명
AUTOMAINT_GET_POLICY 프로시저	SYSPROC	이 프로시저는 데이터베이스에 대한 현재 자동 유지보수 설정을 가져옵니다.
AUTOMAINT_GET_POLICYFILE 프로시저	SYSPROC	이 프로시저는 데이터베이스에 대한 현재 자동 유지보수 설정을 가져옵니다.
AUTOMAINT_SET_POLICY 프로시저	SYSPROC	이 프로시저는 현재 연결된 데이터베이스에 대한 자동 유지보수 규정 설정을 설정합니다.
AUTOMAINT_SET_POLICYFILE 프로시저	SYSPROC	이 프로시저는 현재 연결된 데이터베이스에 대한 자동 유지보수 설정을 설정합니다.

표 6. 일반 SQL API 스토어드 프로시저

루틴 또는 뷰 이름	스키마	설명
CANCEL_WORK 프로시저	SYSPROC	이 프로시저는 지정한 활동을 취소합니다. 고유한 활동 ID를 지정하지 않은 경우에는 연결된 응용프로그램에 대한 모든 활동을 취소하고 시스템에서 강제로 응용프로그램을 종료합니다.
GET_CONFIG 프로시저	SYSPROC	이 프로시저는 nodes.cfg 파일 데이터, 데이터베이스 관리 프로그램 구성 데이터, 데이터베이스 구성 데이터 및 레지스트리 설정을 포함한 데이터 서버 구성 데이터를 모든 데이터베이스 파티션에서 검색합니다.

표 6. 일반 SQL API 스토어드 프로시저 (계속)

루틴 또는 뷰 이름	스키마	설명
GET_MESSAGE 프로시저	SYSPROC	이 프로시저는 짧은 메시지 텍스트, 긴 메시지 텍스트 및 SQLCODE에 대한 SQLSTATE를 검색합니다.
GET_SYSTEM_INFO 프로시저	SYSPROC	이 프로시저는 시스템 정보, 현재 인스턴스, 설치된 DB2 데이터베이스 제품, 환경 변수, 사용 가능한 CPU 및 기타 시스템 정보가 포함된 데이터 서버에 대한 정보를 검색합니다.
SET_CONFIG 프로시저	SYSPROC	이 프로시저는 GET_CONFIG 프로시저로 검색된 구성 매개변수를 갱신합니다.

표 7. 구성 관리 SQL 루틴 및 뷰

루틴 또는 뷰 이름	스키마	설명
DB_PARTITIONS 테이블 함수	SYSPROC	이 테이블 함수는 db2nodes.cfg 파일 콘텐츠를 테이블 폼으로 리턴합니다.
DBCFCG 관리 뷰	SYSIBMADM	이 관리 뷰는 데이터베이스 구성 정보를 리턴합니다.
DBMCFG 관리 뷰	SYSIBMADM	이 관리 뷰는 데이터베이스 관리 프로그램 구성 정보를 리턴합니다.
REG_VARIABLES 관리 뷰	SYSIBMADM	이 관리 뷰는 모든 데이터베이스 파티션에서 DB2 레지스트리 설정을 리턴합니다.

표 8. 환경 관리 뷰

뷰 이름	스키마	설명
ENV_FEATURE_INFO 관리 뷰	SYSPROC	이 관리 뷰는 라이선스가 필요한 모든 사용 가능한 기능에 대한 정보를 리턴합니다.
ENV_INST_INFO 관리 뷰	SYSIBMADM	이 관리 뷰는 현재 인스턴스에 대한 정보를 리턴합니다.
ENV_PROD_INFO 관리 뷰	SYSIBMADM	이 관리 뷰는 설치된 DB2 데이터베이스 제품에 대한 정보를 리턴합니다.
ENV_SYS_INFO 관리 뷰	SYSIBMADM	이 관리 뷰는 시스템에 대한 정보를 리턴합니다.
ENV_SYS_RESOURCES 관리 뷰	SYSIBMADM	이 관리 뷰는 시스템에 관련된 운영 체제, CPU, 메모리 및 기타 정보를 리턴합니다.

표 9. Health 스냅샷 관리 SQL 루틴

루틴 이름	스키마	설명
HEALTH_CONT_HI 테이블 함수	SYSPROC	이 테이블 함수는 데이터베이스의 Health 스냅샷에서 컨테이너에 대한 Health 표시기 정보가 포함된 테이블을 리턴합니다.
HEALTH_CONT_HI_HIS 테이블 함수	SYSPROC	이 테이블 함수는 데이터베이스의 Health 스냅샷에서 컨테이너에 대한 Health 표시기 실행기록이 포함된 테이블을 리턴합니다.
HEALTH_CONT_INFO 테이블 함수	SYSPROC	이 테이블 함수는 데이터베이스의 Health 스냅샷에서 컨테이너에 대한 롤업된 경보 상태 정보가 포함된 테이블을 리턴합니다.

표 9. Health 스냅샷 관리 SQL 루틴 (계속)

루틴 이름	스키마	설명
HEALTH_DB_HI 테이블 함수	SYSPROC	이 테이블 함수는 데이터베이스의 Health 스냅샷에서 Health 표시기 정보가 포함된 테이블을 리턴합니다.
HEALTH_DB_HI_HIS 테이블 함수	SYSPROC	이 테이블 함수는 데이터베이스의 Health 스냅샷에서 Health 표시기 실행기록 정보가 포함된 테이블을 리턴합니다.
HEALTH_DB_HIC 테이블 함수	SYSPROC	이 테이블 함수는 데이터베이스의 Health 스냅샷에서 컬렉션 Health 표시기 정보를 리턴합니다.
HEALTH_DB_HIC_HIS 테이블 함수	SYSPROC	이 테이블 함수는 데이터베이스의 Health 스냅샷에서 컬렉션 Health 표시기 실행기록 정보를 리턴합니다.
HEALTH_DB_INFO 테이블 함수	SYSPROC	이 테이블 함수는 하나 또는 모든 데이터베이스의 Health 스냅샷에서 롤업된 경보 상태 정보가 포함된 테이블을 리턴합니다.
HEALTH_DBM_HI 테이블 함수	SYSPROC	이 테이블 함수는 DB2 데이터베이스 관리 프로그램의 Health 스냅샷에서 Health 표시기 정보가 포함된 테이블을 리턴합니다.
HEALTH_DBM_HI_HIS 테이블 함수	SYSPROC	이 테이블 함수는 DB2 데이터베이스 관리 프로그램의 Health 스냅샷에서 Health 표시기 실행기록 정보가 포함된 테이블을 리턴합니다.
HEALTH_DBM_INFO 테이블 함수	SYSPROC	이 테이블 함수는 DB2 데이터베이스 관리 프로그램의 Health 스냅샷에서 롤업된 경보 상태 정보가 포함된 테이블을 리턴합니다.
HEALTH_GET_ALERT_ACTION_CFG 테이블 함수	SYSPROC	이 테이블 함수는 다양한 유형(dbm, 데이터베이스, 테이블 스페이스 및 테이블 스페이스 컨테이너)의 오브젝트 및 다양한 구성 레벨(설치 디폴트, 인스턴스, 전역 및 오브젝트)에 대한 Health 경보 조치 구성 설정을 리턴합니다.
HEALTH_GET_ALERT_CFG 테이블 함수	SYSPROC	이 테이블 함수는 다양한 유형(dbm, 데이터베이스, 테이블 스페이스, 테이블 스페이스 컨테이너)의 오브젝트 및 다양한 구성 레벨(설치 디폴트, 전역 및 오브젝트)에 대한 Health 경보 구성 설정을 리턴합니다.
HEALTH_GET_IND_DEFINITION 테이블 함수	SYSPROC	이 테이블 함수는 Health 표시기 정의를 리턴합니다.
HEALTH_HI_REC 프로시저	SYSPROC	이 프로시저는 특정 DB2 오브젝트에서 경보 상태의 Health 표시기를 다루는 권장사항 세트를 검색합니다.
HEALTH_TBS_HI 테이블 함수	SYSPROC	이 테이블 함수는 데이터베이스의 Health 스냅샷에서 테이블 스페이스에 대한 Health 표시기 정보가 포함된 테이블을 리턴합니다.
HEALTH_TBS_HI_HIS 테이블 함수	SYSPROC	이 테이블 함수는 데이터베이스의 Health 스냅샷에서 테이블 스페이스에 대한 Health 표시기 실행기록 정보가 포함된 테이블을 리턴합니다.

표 9. Health 스냅샷 관리 SQL 루틴 (계속)

루틴 이름	스키마	설명
HEALTH_TBS_INFO 테이블 함수	SYSPROC	이 테이블 함수는 데이터베이스의 Health 스냅샷에서 테이블 스페이스에 대한 롤업된 경보 상태 정보가 포함된 테이블을 리턴합니다.

표 10. 모니터 SQL 루틴

루틴 이름	스키마	설명
EVMON_FORMAT_UE_TO_TABLES 프로시저	SYSPROC	이 프로시저는 비형식화 이벤트 테이블에 저장된 데이터를 검색하고 XML 문서를 관계형 테이블 세트로 이동합니다.
EVMON_FORMAT_UE_TO_TABLES 테이블 함수	SYSPROC	이 테이블 함수는 비형식화 이벤트 테이블에서 2진 이벤트를 추출하여 XML 문서로 형식화합니다.
MON_GET_ACTIVITY_DETAILS	SYSPROC	이 테이블 함수는 일반 활동 정보 및 활동에 대한 매트릭 세트를 포함하여 활동에 대한 세부사항을 리턴합니다.
MON_GET_BUFFERPOOL 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 버퍼 풀에 대한 모니터 매트릭을 리턴합니다.
MON_GET_CONNECTION 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 연결에 대한 매트릭을 리턴합니다.
MON_GET_CONNECTION_DETAILS 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 연결에 대한 자세한 매트릭을 리턴합니다.
MON_GET_CONTAINER 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 테이블 스페이스 컨테이너에 대한 모니터 매트릭을 리턴합니다.
MON_GET_EXTENT_MOVEMENT_STATUS 테이블 함수	SYSPROC	이 테이블 함수는 Extent 이동 조작 상태를 리턴합니다.
MON_GET_INDEX 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 인덱스에 대한 매트릭을 리턴합니다.
MON_GET_PKG_CACHE_STMT 테이블 함수	SYSPROC	이 테이블 함수는 데이터베이스 패키지 캐시에 있는 정적 및 동적 SQL문 모두에 대한 특정 시점 뷰를 리턴합니다.
MON_GET_SERVICE_SUBCLASS 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 서비스 서브 클래스에 대한 매트릭을 리턴합니다.
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 서비스 서브 클래스에 대한 자세한 매트릭을 리턴합니다.
MON_GET_TABLE 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 테이블에 대한 모니터 매트릭을 리턴합니다.
MON_GET_TABLESPACE 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 테이블 스페이스에 대한 모니터 매트릭을 리턴합니다.
MON_GET_UNIT_OF_WORK 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 작업 단위(UOW)에 대한 매트릭을 리턴합니다.
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 작업 단위(UOW)에 대한 자세한 매트릭을 리턴합니다.

표 10. 모니터 SQL 루틴 (계속)

루틴 이름	스키마	설명
MON_GET_WORKLOAD 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 워크로드에 대한 메트릭을 리턴합니다.
MON_GET_WORKLOAD_DETAILS 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 워크로드에 대한 자세한 메트릭을 리턴합니다.

표 11. MQSeries 관리 SQL 루틴

루틴 이름	스키마	설명
MQPUBLISH 스칼라 함수	DB2MQ, DB2MQ1C	이 스칼라 함수는 데이터를 MQSeries 위치로 발행합니다.
MQREAD 스칼라 함수	DB2MQ, DB2MQ1C	이 스칼라 함수는 MQSeries 위치에서 메시지를 리턴합니다.
MQREADALL 테이블 함수	DB2MQ, DB2MQ1C	이 테이블 함수는 MQSeries 위치에서 메시지 및 메시지 메타데이터가 포함된 테이블을 리턴합니다.
MQREADALLCLOB 테이블 함수	DB2MQ	이 테이블 함수는 MQSeries 위치에서 메시지 및 메시지 메타데이터가 포함된 테이블을 리턴합니다.
MQREADCLOB 스칼라 함수	DB2MQ	이 스칼라 함수는 MQSeries 위치에서 메시지를 리턴합니다.
MQRECEIVE 스칼라 함수	DB2MQ, DB2MQ1C	이 스칼라 함수는 MQSeries 위치에서 메시지를 리턴하고 관련 큐에서 메시지를 제거합니다.
MQRECEIVEALL 테이블 함수	DB2MQ, DB2MQ1C	이 테이블 함수는 MQSeries 위치에서 메시지 및 메시지 메타데이터가 포함된 테이블을 리턴하고 관련 큐에서 메시지를 제거합니다.
MQRECEIVEALLCLOB 테이블 함수	DB2MQ	이 테이블 함수는 MQSeries 위치에서 메시지 및 메시지 메타데이터가 포함된 테이블을 리턴합니다.
MQRECEIVECLOB 스칼라 함수	DB2MQ	이 스칼라 함수는 MQSeries 위치에서 메시지를 리턴합니다.
MQSEND 스칼라 함수	DB2MQ, DB2MQ1C	이 스칼라 함수는 데이터를 MQSeries 위치로 보냅니다.
MQSUBSCRIBE 스칼라 함수	DB2MQ, DB2MQ1C	이 스칼라 함수는 특정 토픽에 발행된 MQSeries 메시지에 서브스크라이브합니다.
MQUNSUBSCRIBE 스칼라 함수	DB2MQ, DB2MQ1C	이 스칼라 함수는 특정 토픽에 발행된 MQSeries 메시지에서 서브스크라이브를 해제합니다.

표 12. 보안 관리 SQL 루틴 및 뷰

루틴 또는 뷰 이름	스키마	설명
AUTH_LIST_AUTHORITIES_FOR_AUTHID 테이블 함수	SYSPROC	이 테이블 함수는 데이터베이스 구성 파일에 있거나 권한 부여 ID에 직접 부여 또는 그룹이나 역할을 통해 간접으로 부여된 권한 부여 ID가 보유하고 있는 모든 권한을 리턴합니다.
AUTH_LIST_GROUPS_FOR_AUTHID 테이블 함수	SYSPROC	이 테이블 함수는 지정한 권한 부여 ID가 구성원인 그룹 목록을 리턴합니다.
AUTH_LIST_ROLES_FOR_AUTHID 함수	SYSPROC	이 함수는 지정한 권한 부여 ID가 구성원인 역할 목록을 리턴합니다.

표 12. 보안 관리 SQL 루틴 및 뷰 (계속)

루틴 또는 뷰 이름	스키마	설명
AUTHORIZATIONIDS 관리 뷰	SYSIBMADM	이 관리 뷰에는 현재 연결된 데이터베이스에 대해 특권이나 권한이 부여된 권한 부여 ID 목록과 해당 유형이 포함됩니다.
OBJECTOWNERS 관리 뷰	SYSIBMADM	이 관리 뷰에는 현재 연결된 데이터베이스에 대한 모든 오브젝트 소유권 정보가 포함됩니다.
PRIVILEGES 관리 뷰	SYSIBMADM	이 관리 뷰에는 현재 연결된 데이터베이스에 대한 모든 명시적 특권이 포함됩니다.

표 13. 스냅샷 관리 SQL 루틴 및 뷰

루틴 또는 뷰 이름	스키마	설명
APPL_PERFORMANCE 관리 뷰	SYSIBMADM	이 관리 뷰에는 응용프로그램별로 선택된 행의 비율 대 읽은 행의 비율에 대한 정보가 표시됩니다.
APPLICATIONS 관리 뷰	SYSIBMADM	이 관리 뷰는 연결된 데이터베이스 응용프로그램의 정보를 리턴합니다.
BP_HITRATIO 관리 뷰	SYSIBMADM	이 관리 뷰는 데이터베이스에서 전체, 데이터 및 인덱스를 포함한 버퍼 풀 사용 비율을 리턴합니다.
BP_READ_IO 관리 뷰	SYSIBMADM	이 관리 뷰는 버퍼 풀 읽기 성능 정보를 리턴합니다.
BP_WRITE_IO 관리 뷰	SYSIBMADM	이 관리 뷰는 버퍼 풀별로 버퍼 풀 쓰기 성능 정보를 리턴합니다.
CONTAINER_UTILIZATION 관리 뷰	SYSIBMADM	이 관리 뷰는 테이블 스페이스 컨테이너 및 활용 비율에 대한 정보를 리턴합니다.
LOCKS_HELD 관리 뷰	SYSIBMADM	이 관리 뷰는 현재 보유된 잠금에 대한 정보를 리턴합니다.
LOCKWAITS 관리 뷰	SYSIBMADM	이 관리 뷰는 권한 부여 대기 중인 잠금에 대한 정보를 리턴합니다.
LOG_UTILIZATION 관리 뷰	SYSIBMADM	이 관리 뷰는 현재 연결된 데이터베이스에 대한 로그 활용에 대한 정보를 리턴합니다.
LONG_RUNNING_SQL 관리 뷰	SYSIBMADM	이 관리 뷰는 현재 연결된 데이터베이스에서 가장 오래 실행 중인 SQL문을 리턴합니다.
QUERY_PREP_COST 관리 뷰	SYSIBMADM	이 관리 뷰는 명령문 준비에 필요한 시간에 대한 정보가 포함된 명령문 목록을 리턴합니다.
SNAP_WRITE_FILE 프로시저	SYSPROC	이 프로시저는 시스템 스냅샷 데이터를 인스턴스 디렉토리의 tmp 서브디렉토리에 파일로 작성합니다.
SNAPAGENT 관리 뷰 및 SNAP_GET_AGENT 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 응용프로그램 스냅샷에서 에이전트 특히, 에이전트 논리 데이터 그룹에 대한 정보를 리턴합니다.
SNAPAGENT_MEMORY_POOL 관리 뷰 및 SNAP_GET_AGENT_MEMORY_POOL 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 에이전트 레벨의 메모리 사용에 대한 정보를 리턴합니다.



표 13. 스냅샷 관리 SQL 루틴 및 뷰 (계속)

루틴 또는 뷰 이름	스키마	설명
SNAPAPPL 관리 뷰 및 SNAP_GET_APPL_V95 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 응용프로그램 스냅샷에서 응용프로그램 특히, 응용프로그램 논리 데이터 그룹에 대한 정보를 리턴합니다.
SNAPAPPL_INFO 관리 뷰 및 SNAP_GET_APPL_INFO_V95 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 응용프로그램 스냅샷에서 응용프로그램 특히, appl_info 논리 데이터 그룹에 대한 정보를 리턴합니다.
SNAPBP 관리 뷰 및 SNAP_GET_BP_V95 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 버퍼 풀 스냅샷에서 버퍼 풀 특히, 버퍼 풀 논리 데이터 그룹에 대한 정보를 리턴합니다.
SNAPBP_PART 관리 뷰 및 SSSNAP_GET_BP_PART 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 버퍼 풀 스냅샷에서 버퍼 풀 특히, bufferpool_nodeinfo 논리 데이터 그룹에 대한 정보를 리턴합니다.
SNAPCONTAINER 관리 뷰 및 SNAP_GET_CONTAINER_V91 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 tablespace_container 논리 데이터 그룹에서 테이블 스페이스 스냅샷 정보를 리턴합니다.
SNAPDB 관리 뷰 및 SNAP_GET_DB_V95 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 데이터베이스 (dbase) 및 데이터베이스 스토리지 (db_storage_group) 논리적 그룹화에서 스냅샷 정보를 리턴합니다.
SNAPDB_MEMORY_POOL 관리 뷰 및 SNAP_GET_DB_MEMORY_POOL 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 UNIX® 플랫폼에만 해당하는 데이터베이스 레벨에서 메모리 사용에 대한 정보를 리턴합니다.
SNAPDBM 관리 뷰 및 SNAP_GET_DBM_V95 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 스냅샷 모니터 DB2 데이터베이스 관리 프로그램(dbm) 논리적 그룹화 정보를 리턴합니다.
SNAPDBM_MEMORY_POOL 관리 뷰 및 SNAP_GET_DBM_MEMORY_POOL 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 데이터베이스 관리 프로그램에서 메모리 사용에 대한 정보를 리턴합니다.
SNAPDETAILLOG 관리 뷰 및 SNAP_GET_DETAILLOG_V91 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 detail_log 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.
SNAPDYN_SQL 관리 뷰 및 SNAP_GET_DYN_SQL_V95 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 dynsql 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.
SNAPFCM 관리 뷰 및 SNAP_GET_FCM 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 데이터베이스 관리 프로그램 스냅샷에서 FCM(Fast Communication Manager) 특히, fcm 논리 데이터 그룹에 대한 정보를 리턴합니다.
SNAPFCM_PART 관리 뷰 및 SNAP_GET_FCM_PART 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 데이터베이스 관리 프로그램 스냅샷에서 FCM(Fast Communication Manager) 특히, fcm_node 논리 데이터 그룹에 대한 정보를 리턴합니다.
SNAPHADR 관리 뷰 및 SNAP_GET_HADR 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 데이터베이스 스냅샷에서 고가용성 재해 복구(HADR) 특히, hadr 논리 데이터 그룹에 대한 정보를 리턴합니다.
SNAPLOCK 관리 뷰 및 SNAP_GET_LOCK 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 잠금 특히, 잠금 논리 데이터 그룹에 대한 스냅샷 정보를 리턴합니다.

표 13. 스냅샷 관리 SQL 루틴 및 뷰 (계속)

루틴 또는 뷰 이름	스키마	설명
SNAPLOCKWAIT 관리 뷰 및 SNAP_GET_LOCKWAIT 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 잠금 대기 특히, 잠금 대기 논리 데이터 그룹에 대한 스냅샷 정보를 리턴합니다.
SNAPSTMT 관리 뷰 및 SNAP_GET_STMT 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 응용프로그램 스냅샷에서 명령문에 대한 정보를 리턴합니다.
SNAPSTORAGE_PATHS 관리 뷰 및 SNAP_GET_STORAGE_PATHS 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 각 스토리지 경로에 대한 파일 시스템 정보를 포함한 데이터베이스에 대한 자동 스토리지 경로 목록을 특히, db_storage_group 논리 데이터 그룹에서 리턴합니다.
SNAPSUBSECTION 관리 뷰 및 SNAP_GET_SUBSECTION 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 응용프로그램 서브섹션 즉, 서브섹션 논리적 모니터링 그룹화에 대한 정보를 리턴합니다.
SNAPSWITCHES 관리 뷰 및 SNAP_GET_SWITCHES 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 데이터베이스 스냅샷 전환 상태에 대한 정보를 리턴합니다.
SNAPTAB 관리 뷰 및 SNAP_GET_TAB_V91 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 테이블 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.
SNAPTAB_REORG 관리 뷰 및 SNAP_GET_TAB_REORG 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 테이블 재구성 정보를 리턴합니다.
SNAPTbsp 관리 뷰 및 SNAP_GET_TBSP_V91 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 테이블 스페이스 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.
SNAPTbsp_PART 관리 뷰 및 SNAP_GET_TBSP_PART_V91 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 tablespace_nodeinfo 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.
SNAPTbsp_QUIESCER 관리 뷰 및 SNAP_GET_TBSP_QUIESCER 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 테이블 스페이스 스냅샷에서 quiescer에 대한 정보를 리턴합니다.
SNAPTbsp_RANGE 관리 뷰 및 SNAP_GET_TBSP_RANGE 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 범위 스냅샷에서 정보를 리턴합니다.
SNAPUTIL 관리 뷰 및 SNAP_GET_UTIL 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 utility_info 논리 데이터 그룹에서 활용에 대한 스냅샷 정보를 리턴합니다.
SNAPUTIL_PROGRESS 관리 뷰 및 SNAP_GET_UTIL_PROGRESS 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 유틸리티 진행 특히, 진행 논리 데이터 그룹에 대한 정보를 리턴합니다.
TBSP_UTILIZATION 관리 뷰	SYSIBMADM	이 관리 뷰는 테이블 스페이스 구성 및 활용 정보를 리턴합니다.
TOP_DYNAMIC_SQL 관리 뷰	SYSIBMADM	이 관리 뷰는 실행 수, 평균 실행 시간, 정렬 수 또는 명령문당 정렬 수로 정렬할 수 있는 가장 동적인 SQL문을 리턴합니다.

표 14. SQL 프로시저 관리 SQL 루틴

루틴 이름	스키마	설명
ALTER_ROUTINE_PACKAGE 프로시저	SYSPROC	이 프로시저는 리바인딩할 필요없이 컴파일된 SQL 루틴 또는 컴파일된 트리거와 연관된 패키지의 값을 변경합니다.

표 14. SQL 프로시저 관리 SQL 루틴 (계속)

루틴 이름	스키마	설명
GET_ROUTINE_OPTS 스칼라 함수	SYSPROC	이 스칼라 함수는 현재 세션에서 SQL 프로시저 작성에 사용되는 옵션의 문자열 값을 리턴합니다.
GET_ROUTINE_SAR 프로시저	SYSFUN	이 프로시저는 동일한 레벨 이상을 실행 중인 다른 데이터베이스 서버 및 운영 체제에 동일한 루틴을 설치하는 데 필요한 정보를 리턴합니다.
PUT_ROUTINE_SAR 프로시저	SYSFUN	이 프로시저는 데이터베이스 서버에 SQL 루틴을 작성하고 정의하는 데 필요한 정보를 전달합니다.
REBIND_ROUTINE_PACKAGE 프로시저	SYSPROC	이 프로시저는 SQL 프로시저와 연관된 패키지를 리바인드합니다.
SET_ROUTINE_OPTS 프로시저	SYSPROC	이 프로시저는 현재 세션에서 SQL 프로시저 작성에 사용되는 옵션을 설정합니다.

표 15. Stepwise 재분배 관리 SQL 루틴

루틴 이름	스키마	설명
ANALYZE_LOG_SPACE 프로시저	SYSPROC	이 프로시저는 로그 스페이스 분석 정보를 리턴합니다.
GENERATE_DISTFILE 프로시저	SYSPROC	이 프로시저는 데이터 분산 파일을 생성합니다.
GET_SWRD_SETTINGS 프로시저	SYSPROC	이 프로시저는 재분배 정보를 리턴합니다.
SET_SWRD_SETTINGS 프로시저	SYSPROC	이 프로시저는 재분배 레지스트리를 작성하거나 변경합니다.
STEPWISE_REDISTRIBUTE_DBPG 프로시저	SYSPROC	이 프로시저는 데이터베이스 파티션 그룹의 파트를 재분배합니다.

표 16. 스토리지 관리 도구 관리 SQL 루틴

루틴 이름	스키마	설명
CAPTURE_STORAGEMGMT_INFO 프로시저	SYSPROC	이 프로시저는 지정한 루트 오브젝트에 대한 스토리지 관련 정보를 리턴합니다.
CREATE_STORAGEMGMT_TABLES 프로시저	SYSPROC	이 프로시저는 스토리지 관리 테이블을 작성합니다.
DROP_STORAGEMGMT_TABLES 프로시저	SYSPROC	이 프로시저는 모든 스토리지 관리 테이블을 삭제(drop)합니다.

표 17. 텍스트 검색 관리 SQL 루틴

루틴 이름	스키마	설명
SYSTS_ADMIN_CMD 스토어드 프로시저	SYSPROC	이 프로시저는 SQL CALL문을 사용하여 텍스트 검색 관리 명령을 실행합니다.
SYSTS_ALTER 프로시저	SYSPROC	이 프로시저는 인덱스의 갱신 특성을 변경합니다.
SYSTS_CLEAR_COMMANDLOCKS 프로시저	SYSPROC	이 프로시저는 데이터베이스에서 특정 텍스트 검색 인덱스 또는 전체 텍스트 검색 인덱스에 대한 모든 명령 잠금을 제거합니다.
SYSTS_CLEAR_EVENTS 프로시저	SYSPROC	이 프로시저는 관리에 사용되는 인덱스의 이벤트 테이블에서 인덱싱 이벤트를 삭제합니다.

표 17. 텍스트 검색 관리 SQL 루틴 (계속)

루틴 이름	스키마	설명
SYSTS_CREATE 프로시저	SYSPROC	이 프로시저는 텍스트 검색 기능으로 컬럼 데이터를 검색할 수 있게 해주는 텍스트 컬럼에 대한 텍스트 검색 인덱스를 작성합니다.
SYSTS_DISABLE 프로시저	SYSPROC	이 프로시저는 현재 데이터베이스에 대해 DB2 텍스트 검색을 사용하지 않습니다.
SYSTS_DROP 프로시저	SYSPROC	이 프로시저는 모든 테이블 컬럼에 연관된 기존의 텍스트 검색 인덱스를 삭제(drop)합니다.
SYSTS_ENABLE 프로시저	SYSPROC	데이터베이스내의 테이블 컬럼에서 텍스트 검색 인덱스를 작성하기 전에 이 프로시저를 제대로 실행해야 합니다.
SYSTS_UPDATE 프로시저	SYSPROC	이 프로시저는 인덱스가 연관된 텍스트 컬럼의 현재 콘텐츠를 반영하기 위해 텍스트 검색 인덱스를 갱신합니다.

표 18. 워크로드 관리를 위한 관리 SQL 루틴

루틴 이름	스키마	설명
WLM_CANCEL_ACTIVITY 프로시저	SYSPROC	이 프로시저는 지정된 활동을 취소합니다.
WLM_CAPTURE_ACTIVITY_IN_PROGRESS 프로시저	SYSPROC	이 프로시저는 지정된 활동의 정보를 활동 이벤트 모니터로 보냅니다.
WLM_COLLECT_STATS 프로시저	SYSPROC	이 프로시저는 서비스 클래스, 워크로드, 작업 클래스 및 임계값 큐에 대한 통계를 통계 이벤트 모니터로 보내고 통계에 대한 메모리내의 사본을 재설정합니다.
WLM_GET_QUEUE_STATS 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 임계값 큐에 대한 기본 통계 정보를 리턴합니다.
WLM_GET_SERVICE_CLASS_AGENTS_V97 테이블 함수	SYSPROC	이 테이블 함수는 SERVICE_SUPERCLASS_NAME 및 SERVICE_SUBCLASS_NAME으로 지정된 서비스 클래스 또는 APPLICATION_HANDLE로 지정된 응용프로그램 대신 실행 중인 지정된 파티션의 에이전트 목록을 리턴합니다.
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES_V97 테이블 함수	SYSPROC	이 테이블 함수는 특정 파티션의 지정된 서비스 클래스에서 실행 중인 전체 워크로드 어커런스 목록을 리턴합니다.
WLM_GET_SERVICE_SUBCLASS_STATS_V97 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 서비스 서브클래스에 대한 기본 통계를 리턴합니다.
WLM_GET_SERVICE_SUPERCLASS_STATS 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 서비스 수퍼 클래스에 대한 기본 통계를 리턴합니다.
WLM_GET_WORK_ACTION_SET_STATS 테이블 함수	SYSPROC	이 테이블 함수는 작업 조치 세트에서 작업 클래스에 대한 기본 통계를 리턴합니다.
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97 테이블 함수	SYSPROC	이 테이블 함수는 지정된 파티션에서 지정된 응용 프로그램을 통해 제출되었으며 아직 완료되지 않은 모든 활동 목록을 리턴합니다.
WLM_GET_WORKLOAD_STATS_V97 테이블 함수	SYSPROC	이 테이블 함수는 하나 이상의 워크로드에 대한 기본 통계를 리턴합니다.

표 18. 워크로드 관리를 위한 관리 SQL 루틴 (계속)

루틴 이름	스키마	설명
WLM_SET_CLIENT_INFO 프로시저	SYSPROC	이 프로시저는 DB2 서버에서 현재 연결에 연관된 클라이언트 정보를 설정합니다.

표 19. 기타 관리 SQL 루틴 및 뷰

루틴 또는 뷰 이름	스키마	설명
ADMIN_COPY_SCHEMA 프로시저	SYSPROC	이 프로시저를 사용하여 특정 스키마와 포함된 모든 오브젝트를 복사합니다.
ADMIN_DROP_SCHEMA 프로시저	SYSPROC	이 프로시저를 사용하여 특정 스키마와 포함된 모든 오브젝트를 삭제(drop)합니다.
ADMIN_MOVE_TABLE 프로시저	SYSPROC	이 프로시저는 데이터가 온라인 상태로 액세스할 수 있는 동안 활성 테이블의 데이터를 동일한 이름으로 새 테이블 오브젝트로 이동합니다.
ADMIN_MOVE_TABLE_UTIL 프로시저	SYSPROC	이 프로시저는 ADMIN_MOVE_TABLE 프로시저에서 사용되는 사용자가 정의할 수 있는 값으로 변경합니다.
ALTOBJ 프로시저	SYSPROC	이 프로시저는 입력 CREATE TABLE문을 목표 테이블 정의로 사용하여 기존 테이블을 변경합니다.
APPLICATION_ID 스칼라 함수	SYSFUN	이 스칼라 함수는 현재 연결의 응용프로그램 ID를 리턴합니다.
COMPILATION_ENV 테이블 함수	SYSPROC	이 테이블 함수는 컴파일 환경 요소를 리턴합니다.
CONTACTGROUPS 관리 뷰	SYSIBMADM	이 관리 뷰는 문의처 그룹의 목록을 리턴합니다.
CONTACTS 관리 뷰	SYSIBMADM	이 관리 뷰는 데이터베이스 서버에 정의된 문의처 목록을 리턴합니다.
DB_HISTORY 관리 뷰	SYSIBMADM	이 관리 뷰는 현재 연결된 데이터베이스 파티션에 연관된 실행기록 파일에서 정보를 리턴합니다.
DBPATHS 관리 뷰	SYSIBMADM	이 관리 뷰는 분할 미리 백업과 같은 태스크에 필요한 데이터베이스 경로 값을 리턴합니다.
EXPLAIN_FORMAT_STATS 스칼라 함수	SYSPROC	이 새 스칼라 함수를 사용하여 지정된 쿼리에 대해 캡처된 Explain 스냅샷에서 구문 분석되고 추출되는 형식화된 통계 정보를 표시합니다.
EXPLAIN_GET_MSGS 테이블 함수	스키마는 Explain 테이블 스키마와 동일합니다.	이 테이블 함수는 EXPLAIN_DIAGNOSTIC 및 EXPLAIN_DIAGNOSTIC_DATA Explain 테이블을 쿼리하고 형식화된 메시지를 리턴합니다.
GET_DBSIZE_INFO 프로시저	SYSPROC	이 프로시저는 데이터베이스 크기 및 최대 용량을 계산합니다.
NOTIFICATIONLIST 관리 뷰	SYSIBMADM	이 관리 뷰는 인스턴스의 Health에 대해 통지된 문의처 및 문의처 그룹 목록을 리턴합니다.
PD_GET_DIAG_HIST 테이블 함수	SYSPROC	테이블 함수는 지정된 기능에서 로그 레코드, 이벤트 레코드 및 통지 레코드를 리턴합니다.

표 19. 기타 관리 SQL 루틴 및 뷰 (계속)

루틴 또는 뷰 이름	스키마	설명
PDLOGMSG_LAST24HOURS 관리 뷰 및 PD_GET_LOG_MSGS 테이블 함수	SYSIBMADM(관리 뷰), SYSPROC(테이블 함수)	이 관리 뷰 및 테이블 함수는 DB2 통지 로그에 로깅된 문제점 판별 로그 메시지를 리턴합니다. 정보는 데이터베이스 및 시스템 관리자가 사용하도록 제공됩니다.
REORGCHK_IX_STATS 프로시저	SYSPROC	이 프로시저는 인덱스 통계를 점검하여 재구성 필요 여부를 판별합니다.
REORGCHK_TB_STATS 프로시저	SYSPROC	이 프로시저는 테이블 통계를 점검하여 재구성 필요 여부를 판별합니다.
SQLERRM 스칼라 함수	SYSPROC	이 스칼라 함수에는 두 개의 버전이 있습니다. 첫 번째 버전을 사용하면 메시지 토큰 및 언어 선택 사용을 포함한 메시지 검색에 대한 완벽한 유연성이 제공됩니다. 두 번째 버전은 SQLCODE를 입력 매개변수로 사용하고 영어로 간략한 메시지만 리턴하는 단순한 인터페이스입니다.
SYSINSTALLOBJECTS 프로시저	SYSPROC	이 프로시저는 특정 도구에 필요한 데이터베이스 오브젝트를 작성하거나 삭제합니다.

---

## 제 3 장 활동 모니터 루틴

---

### AM\_BASE\_RPT\_RECOMS – 활동 보고서에 대한 권장사항

AM\_BASE\_RPT\_RECOMS 테이블 함수는 활동 모니터에서 사용한 활동 보고서에 대한 권장사항을 리턴합니다.

#### 구문

▶▶—AM\_BASE\_RPT\_RECOMS—(—report-id—,—client-locale—)————▶▶

스키마는 SYSPROC입니다.

#### 테이블 함수 매개변수

##### *report-id*

유형 INTEGER의 입력 인수는 보고서 ID를 지정합니다. 인수가 null인 경우 가능한 모든 보고서에 대한 권장사항이 리턴됩니다.

##### *client-locale*

유형 VARCHAR(33)의 입력 인수는 클라이언트 언어 ID를 지정합니다. 인수가 null이거나 비어 있는 문자열인 경우 디폴트값은 'En\_US'(영어)입니다. 지정된 로케일에 대한 메시지 파일을 서버에서 사용할 수 없는 경우 'En-US'가 사용됩니다.

#### 권한 부여

AM\_BASE\_RPT\_RECOMS 테이블 함수에 대한 EXECUTE 특권

#### 예:

예 1: ID가 1인 활동 모니터 보고서에 대한 권장사항(영어)을 요청합니다. 디폴트 클라이언트 언어 ID가 'En\_US'라고 가정합니다.

```
SELECT *  
  FROM TABLE(SYSPROC.AM_BASE_RPT_RECOMS(1, CAST(NULL AS VARCHAR(33))))  
 AS RECOMS
```

예 2: ID가 12인 활동 모니터 보고서에 대한 권장사항(프랑스어)을 요청합니다.

```
SELECT *  
  FROM TABLE(SYSPROC.AM_BASE_RPT_RECOMS(12, CAST('Fr_FR' AS VARCHAR(33))))  
 AS RECOMS
```



## 리턴되는 정보

표 20. AM\_BASE\_RPT\_RECOMS 테이블 함수가 리턴한 정보

컬럼 이름	데이터 유형	설명
REPORT_ID	INTEGER	보고서 ID
RECOM_NAME	VARCHAR(256)	권장사항의 이름 또는 간단한 설명
RECOM_DESCRIPTION	CLOB(32K)	권장사항의 자세한 설명

---

## AM\_BASE\_RPTS – 활동 모니터 보고서

AM\_BASE\_RPTS 테이블 함수를 활동 모니터에서 사용한 활동 보고서를 리턴합니다.

### 구문

```
▶▶ AM_BASE_RPTS(—report-id—, —type—, —client-locale—)◀◀
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *report-id*

유형 INTEGER의 입력 인수는 고유 보고서 ID를 지정합니다. 인수가 null인 경우 보고서 ID와 함께 보고서가 리턴됩니다.

#### *type*

유형 CHAR(4)의 입력 인수는 보고서 유형을 지정합니다. 가능한 값은 다음과 같습니다.

*APPL* 응용프로그램

*STMT* SQL문

*TRAN* 트랜잭션

*CACH* 동적 SQL문 캐시

값은 대문자 또는 소문자로 지정될 수 있습니다. 인수가 null이거나 비어 있는 문자열인 경우 유형의 보고서가 리턴됩니다.

#### *client-locale*

유형 VARCHAR(33)의 입력 인수는 클라이언트 언어 ID를 지정합니다. 인수가 null이거나 비어 있는 문자열인 경우 또는 지정된 로케일의 메시지 파일을 서버에서 사용할 수 없는 경우 'En-US'가 사용됩니다.

### 권한 부여

AM\_BASE\_RPTS 테이블 함수에 대한 EXECUTE 특권

예:

예 1:

```
SELECT * FROM TABLE(SYSPROC.AM_BASE_RPTS(CAST(NULL AS INTEGER),
      CAST(NULL AS CHAR(4)), CAST(NULL AS VARCHAR(33)))) AS REPORTS
```

예 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'
```

## 리턴되는 정보

표 21. AM\_BASE\_RPTS 테이블 함수가 리턴한 정보

컬럼 이름	데이터 유형	설명
ID	INTEGER	고유 보고서 ID
TYPE	CHAR(4)	보고서 유형. 올바른 값: APPL, STMT, TRAN, CACH
NAME	VARCHAR(256)	보고서의 이름 또는 간단한 설명
DESCRIPTION	VARCHAR(16384)	보고서의 자세한 설명
SWITCHES	VARCHAR(100)	이 보고서에 대해 필요한 모니터 스위치

---

## AM\_DROP\_TASK – 모니터링 태스크 삭제

AM\_DROP\_TASK 프로시저는 모니터링 태스크를 삭제합니다. 데이터를 리턴하지 않습니다.

### 구문

▶▶ AM\_DROP\_TASK (—task-id—) ◀◀

스키마는 SYSPROC입니다.

### 프로시저 매개변수

*task-id*

유형 INTEGER의 입력 인수는 고유 모니터링 태스크 ID를 지정합니다.

### 권한 부여

AM\_DROP\_TASK 프로시저에 대한 EXECUTE 특권

예 :

ID가 5인 모니터링 태스크를 삭제합니다.

```
CALL SYSPROC.AM_DROP_TASK(5)
```

---

## AM\_GET\_LOCK\_CHN\_TB - 응용프로그램 잠금 체인 데이터를 표 형식으로 검색

AM\_GET\_LOCK\_CHN\_TB 프로시저는 응용프로그램 잠금 체인 데이터를 표 형식으로 리턴합니다. 잠금 체인은 직접 또는 간접적으로 현재 응용프로그램이 보유 중이거나 대기 중인 모든 응용프로그램으로 구성되어 있습니다.

### 구문

```
▶▶ AM_GET_LOCK_CHN_TB(—agent-id—)◀◀
```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

*agent-id*

잠금 체인 데이터에 대한 응용프로그램의 에이전트 ID를 지정하는 BIGINT 유형의 입력 인수가 검색됩니다.

### 권한 부여

- SYSMON 권한
- AM\_GET\_LOCK\_CHN\_TB 프로시저에 대한 EXECUTE 특권

예 :

에이전트 ID 68에 대한 잠금 체인 정보를 검색합니다.

```
CALL SYSPROC.AM_GET_LOCK_CHN_TB(68)
```

### 리턴되는 정보

프로시저는 아래에 표시된 대로 테이블을 리턴합니다. 테이블 관계의 각 행은 잠금 대기 관계를 표시합니다. 또한 결과 세트는 각 보유 전용 응용프로그램에 대한 행을 포함합니다. 이 경우 HOLDING\_AGENT\_ID 컬럼은 null이며 기타 네 개의 컬럼은 보유 전용 응용프로그램용입니다.

표 22. AM\_GET\_LOCK\_CHN\_TB 프로시저가 리턴한 정보

컬럼 이름	데이터 유형	설명
HOLDING_AGENT_ID	BIGINT	잠금을 보유 중인 응용프로그램의 에이전트 ID

표 22. AM\_GET\_LOCK\_CHN\_TB 프로시저가 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명
AGENT_ID	BIGINT	잠금에 대해 대기 중인 응용프로그램의 에이전트 ID
APPL_NAME	VARCHAR(255)	잠금에 대해 대기 중인 응용프로그램의 이름
AUTH_ID	VARCHAR(128)	잠금에 대해 대기 중인 응용프로그램의 권한 부여 ID
APPL_ID	VARCHAR(64)	잠금에 대해 대기 중인 응용프로그램의 응용프로그램 ID

## AM\_GET\_LOCK\_CHNS – 특정 응용프로그램에 대한 잠금 체인 정보 검색

AM\_GET\_LOCK\_CHNS 프로시저는 출력 문자열로 지정된 응용프로그램에 잠금 체인을 리턴합니다. 잠금 체인은 직접 또는 간접적으로 현재 응용프로그램이 보유 중이거나 대기 중인 모든 응용프로그램으로 구성되어 있습니다.

### 구문

```
▶▶ AM_GET_LOCK_CHNS (—agent-id—, —lock-chains—) ▶▶
```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### agent-id

유형 BIGINT의 입력 인수는 잠금 체인이 표시될 응용프로그램의 에이전트 ID를 지정합니다.

#### lock-chains

유형 CLOB(2M)의 출력 인수는 지정된 응용프로그램에 대해 모든 잠금 체인을 표시합니다.

### 권한 부여

- SYSMON 권한
- AM\_GET\_LOCK\_CHNS 프로시저에 대한 EXECUTE 특권

### 예 :

```
CALL SYSPROC.AM_GET_LOCK_CHNS(17,?)
```

출력 매개변수 값

-----

매개변수 이름 : LOCK\_CHAINS

매개변수 값 : >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 )

리턴 상태 = 0

---

## AM\_GET\_LOCK\_RPT – 응용프로그램 잠금 세부사항 검색

AM\_GET\_LOCK\_RPT 프로시저는 세 개의 출력 결과 세트로 응용프로그램에 대한 잠금 세부사항을 리턴합니다.

### 구문

▶▶ AM\_GET\_LOCK\_RPT(—agent-id—)◀◀

스키마는 SYSPROC입니다.

### 프로시저 매개변수

*agent-id*

잠금 세부사항이 리턴되는 응용프로그램의 에이전트 ID를 지정하는 BIGINT 유형의 입력 인수입니다.

### 권한 부여

- SYSMON 권한
- AM\_GET\_LOCK\_RPT 프로시저의 EXECUTE 특권

예 :

```
CALL SYSPROC.AM_GET_LOCK_RPT(68)
```

### 사용법 참고

정보가 리턴되도록 하려면 이 프로시저에 대해 DFT\_MON\_LOCK 모니터 스위치를 켜야 합니다.

### 리턴된 정보

프로시저는 다음과 같은 세 개의 결과 세트를 리턴합니다. 응용프로그램 일반 정보에 대한 결과 세트, 응용프로그램이 보유한 잠금에 대한 결과 세트 및 응용프로그램이 대기 중인 잠금에 대한 결과 세트입니다.

표 23. AM\_GET\_LOCK\_RPT 프로시저에서 리턴하는 일반 응용프로그램 정보

컬럼 이름	데이터 유형	설명
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
APPL_NAME	VARCHAR(256)	appl_name - 응용프로그램 이름

표 23. AM\_GET\_LOCK\_RPT 프로시저에서 리턴하는 일반 응용프로그램 정보 (계속)

컬럼 이름	데이터 유형	설명
AUTH_ID	VARCHAR (128)	auth_id - 권한 부여 ID
APPL_ID	VARCHAR (128)	appl_id - 응용프로그램 ID
APPL_STATUS	VARCHAR(22)	<p>appl_status - 응용프로그램 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다.</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_PARTITION_NUM	SMALLINT	coord_node - 코디네이팅 노드
SEQUENCE_NO	VARCHAR(4)	sequence_no - 시퀀스 번호

표 23. AM\_GET\_LOCK\_RPT 프로시저에서 리턴하는 일반 응용프로그램 정보 (계속)

컬럼 이름	데이터 유형	설명
CLIENT_PRDID	VARCHAR (128)	client_prdid - 클라이언트 제품/버전 ID
CLIENT_PID	BIGINT	client_pid - 클라이언트 프로세스 ID



표 23. AM\_GET\_LOCK\_RPT 프로시저에서 리턴하는 일반 응용프로그램 정보 (계속)

컬럼 이름	데이터 유형	설명
CLIENT_PLATFORM	VARCHAR(12)	<p>client_platform - 클라이언트 운영 플랫폼. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴합니다.</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> <li>• WINDOWS95</li> </ul>

표 23. AM\_GET\_LOCK\_RPT 프로시저에서 리턴하는 일반 응용프로그램 정보 (계속)

컬럼 이름	데이터 유형	설명
CLIENT_PROTOCOL	VARCHAR(10)	client_protocol - 클라이언트 통신 프로토콜. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴합니다. <ul style="list-style-type: none"> <li>• CPIC</li> <li>• LOCAL</li> <li>• NETBIOS</li> <li>• NPIPE</li> <li>• TCPIP(DB2® Universal Database™ 또는 DB2 UDB용)</li> <li>• TCPIP4</li> <li>• TCPIP6</li> </ul>
CLIENT_NNAME	VARCHAR(128)	client_nname 모니터 요소는 사용되지 않습니다. 리턴된 값은 올바른 값이 아닙니다.
LOCKS_HELD	BIGINT	locks_held - 보유된 잠금 수
LOCK_WAIT_START_TIME	TIMESTAMP	lock_wait_start_time - 잠금 대기 시작 시간소인
LOCK_WAIT_TIME	BIGINT	lock_wait_time - 잠금 대기 시간
LOCK_WAITS	BIGINT	lock_waits - 잠금 대기 수
LOCK_TIMEOUTS	BIGINT	lock_timeouts - 잠금 시간종료 수
LOCK_ESCALS	BIGINT	lock_escalations - 잠금 에스컬레이션 수
X_LOCK_ESCALS	BIGINT	x_lock_escalations - 배타적 잠금 에스컬레이션 수
DEADLOCKS	BIGINT	deadlocks - 발견된 교착 상태 수

표 24. AM\_GET\_LOCK\_RPT 프로시저에서 리턴한 보유된 잠금 정보

컬럼 이름	데이터 유형	설명
TBSP_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름
TABSCHEMA	VARCHAR(128)	table_schema - 테이블 스키마 이름
TABNAME	VARCHAR(128)	table_name - 테이블 이름

표 24. AM\_GET\_LOCK\_RPT 프로시저에서 리턴한 보유된 잠금 정보 (계속)

컬럼 이름	데이터 유형	설명
LOCK_OBJECT_TYPE	VARCHAR(18)	lock_object_type - 대기하는 잠금 오브젝트 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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>

표 24. AM\_GET\_LOCK\_RPT 프로시저에서 리턴한 보유된 잠금 정보 (계속)

컬럼 이름	데이터 유형	설명
LOCK_MODE	VARCHAR(10)	lock_mode - 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON(잠금이 없는 경우)</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 - 잠금 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• CONV</li> <li>• GRNT</li> </ul>
LOCK_ESCALATION	SMALLINT	lock_escalation - 잠금 에스컬레이션
LOCK_NAME	VARCHAR(32)	lock_name - 잠금 이름
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

표 25. AM\_GET\_LOCK\_RPT 프로시저가 잠금 대기 정보 리턴

컬럼 이름	데이터 유형	설명
AGENT_ID_HOLDING_LK	BIGINT	agent_id_holding_lock - 잠금을 보유한 에이전트 ID
APPL_ID_HOLDING_LK	VARCHAR(128)	appl_id_holding_lk - 잠금을 보유한 응용프로그램 ID
LOCK_WAIT_START_TIME	TIMESTAMP	lock_wait_start_time - 잠금 대기 시작 시간소인
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
TBSP_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름
TABSCHEMA	VARCHAR(128)	table_schema - 테이블 스키마 이름
TABNAME	VARCHAR(128)	table_name - 테이블 이름

표 25. AM\_GET\_LOCK\_RPT 프로시저가 잠금 대기 정보 리턴 (계속)

컬럼 이름	데이터 유형	설명
LOCK_OBJECT_TYPE	VARCHAR(18)	lock_object_type - 대기하는 잠금 오브젝트 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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>

표 25. AM\_GET\_LOCK\_RPT 프로시저가 잠금 대기 정보 리턴 (계속)

컬럼 이름	데이터 유형	설명
LOCK_MODE	VARCHAR(10)	lock_mode - 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON(잠금이 없는 경우)</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 - 요청된 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON(잠금이 없는 경우)</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 - 잠금 에스컬레이션

## AM\_GET\_RPT – 활동 모니터 데이터 검색

AM\_GET\_RPT 프로시저는 보고서에 활동 모니터 데이터를 리턴합니다.

### 구문







예 :

예):

```
CALL SYSPROC.AM_SAVE_TASK('M',11,'Task ABC',CAST (NULL AS CLOB(32K)),  
    'N','1,2,4,8,9,12')
```



---

## 제 4 장 ADMIN\_CMD 프로시저 및 관련 루틴

---

### ADMIN\_CMD – 관리 명령 실행

ADMIN\_CMD 프로시저는 SQL CALL문을 사용하여 관리 명령을 실행하는 응용프로그램에 의해 사용됩니다.

#### 구문

▶▶—ADMIN\_CMD—(—*command-string*—)————▶▶

스키마는 SYSPROC입니다.

#### 프로시저 매개변수

##### *command-string*

실행할 단일 명령을 지정하는 CLOB(2M) 유형의 입력 인수.

#### 권한 부여

ADMIN\_CMD 프로시저에 대한 EXECUTE 특권

프로시저는 현재 다음과 같은 DB2 명령행 처리기(CLP) 명령을 지원합니다.

- ADD CONTACT
- ADD CONTACTGROUP
- AUTOCONFIGURE
- BACKUP - 온라인 전용
- 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

주: 일부 명령은 ADMIN\_CMD 프로시저에서 실행할 때 지원되는 구문이 약간씩 서로 다를 수도 있습니다.

또한 프로시저는 CLP에서 지원하지 않은 다음과 같은 명령을 지원합니다.

- GET STMM TUNING DBPARTITIONNUM
- UPDATE STMM TUNING DBPARTITIONNUM

## 사용 시 참고사항

명령 실행 정보 검색:

- ADMIN\_CMD 프로시저가 서버에서 실행되면, 유틸리티 메시지도 서버에 작성됩니다. MESSAGES ON SERVER 옵션(세부사항은 특정 명령 참조)은 서버에 메시지 파일이 작성됨을 표시합니다.
- 명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.
- 관리 명령 실행이 성공하고 명령이 실행 상태보다 많은 정보를 리턴할 경우, 추가 정보는 결과 세트 형식으로 리턴됩니다(최대 2개의 결과 세트). 예를 들어 EXPORT 명령이 성공적으로 실행되면 리턴된 결과 세트에는 익스포트된 행 수에 대한 정보가 들어갑니다. 그러나 RUNSTATS 명령이 성공적으로 실행되면 결과 세트가 리턴되지 않습니다. 결과 세트 정보는 해당하는 명령을 사용하여 표시됩니다.

- 관리 명령 실행이 성공하지 않을 경우, ADMIN\_CMD 프로시저에서 관리 명령 실패 이유에 대한 세부사항을 포함하는 결과 세트와 함께 SQL20397W 경고 메시지를 리턴합니다. ADMIN\_CMD 프로시저를 사용하는 모든 응용프로그램은 프로시저에서 리턴한 SQLCODE를 점검해야 합니다. SQLCODE >= 0의 경우, 관리 명령에 대한 결과 세트를 검색해야 합니다. 다음 테이블은 MESSAGES ON SERVER 옵션 사용 여부에 따라 리턴되는 정보를 표시합니다.

표 26. SQLCODE 및 ADMIN\_CMD 프로시저에 의해 리턴되는 정보

관리 명령 실행 상태	지정되는 MESSAGES ON SERVER 옵션	지정되지 않는 MESSAGES ON SERVER 옵션
성공	리턴되는 SQLCODE >= 0이면, 추가 정보(결과 세트)가 리턴됩니다 (존재하는 경우).	리턴되는 SQLCODE >= 0이면, 추가 정보(결과 세트)가 리턴되지 만(존재하는 경우), MSG_RETRIEVAL 및 MSG_REMOVAL 컬럼은 NULL입니다.
실패	SQLCODE에서 20397을 리턴할 경우 추가 정보(결과 세트)가 리턴 되지만 MSG_RETRIEVAL 및 MSG_REMOVAL 컬럼에만 데이터가 채워집니다.	리턴되는 SQLCODE가 < 0이면 추가 정보(결과 세트)가 리턴되지 않습니다.

- 결과 세트는 CLP 또는 임베디드(embedded) C 응용프로그램이 아닌 JDBC 및 DB2 CLI 응용프로그램에서 검색할 수 있습니다.
- 대소문자 구분 이름과 2바이트 문자 세트(DBCS) 이름은 백슬래시와 큰따옴표 분리 문자 안에 묶어야 합니다(예: #” MyTabLe #”).

ADMIN\_CMD를 통해 실행되는 모든 명령의 경우, 데이터베이스에 대한 연결을 설정한 사용자 ID가 인증에 사용됩니다.

예를 들어, 데이터베이스 서버에 대한 파일 시스템 액세스를 필요로 하는 명령에 필요한 모든 추가 권한이 명령을 설명하는 참조 정보에 나열됩니다.

이 프로시저는 사용자 정의 함수(UDF)(SQLSTATE 38001) 또는 트리거에서 호출할 수 없습니다.

## ADMIN\_CMD 프로시저를 사용하는 ADD CONTACT 명령

이 명령은 시스템 또는 전역 목록에서 로컬로 정의할 수 있는 문의처 목록에 문의처를 추가합니다. 문의처는 스케줄러 및 Health Monitor와 같은 프로세스가 메시지를 보낼 수 있는 사용자입니다. DAS(Database Administration Server) **contact\_host** 구성 매개변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

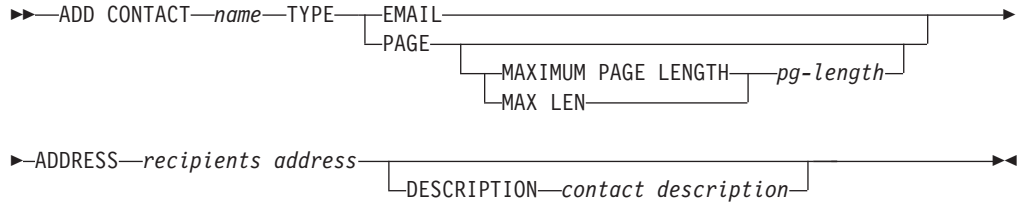
### 권한 부여

없음

## 필수 연결

데이터베이스. DAS가 실행하고 있어야 합니다.

## 명령 구문



## 명령 매개변수

### ADD CONTACT *name*

추가될 문의처 이름. DB2 관리 서버 구성 매개변수 **contact\_host**가 다른 시스템을 지정하는 경우를 제외하고는 문의처는 디폴트로 로컬 시스템에 추가됩니다.

**TYPE** 문의처의 메소드로서 다음 중 하나가 되어야 합니다.

#### EMAIL

이 문의처는 (ADDRESS)에서 전자 우편에 의해 통지됩니다.

#### PAGE

이 문의처는 ADDRESS로 보내진 페이지에 의해 통지됩니다.

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

페이징 서비스에 메시지 길이 제한이 있는 경우 이 예에서 문자로 지정됩니다.

통지 시스템은 SMTP 프로토콜을 사용하여 DB2 관리 서버 구성 매개변수 **smtp\_server** 에 의해 지정된 메일 서버로 통지를 보냅니다. 전자 우편을 보내거나 호출기를 호출하는 것은 SMTP 서버의 역할입니다.

### ADDRESS *recipients-address*

받는 사람의 SMTP 편지함 주소(예: joe@somewhere.org). **smtp\_server** DAS 구성 매개변수는 SMTP 서버의 이름으로 설정되어야 합니다.

### DESCRIPTION *contact description*

문의처의 텍스트 설명. 최대 128 문자를 사용합니다.

## 예 :

전자 우편 주소가 'testuser@test.com'인 사용자 'testuser'에 대한 문의처를 추가합니다.



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

## 사용 시 참고사항

DAS를 작성하여 실행해야 합니다.

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

## ADMIN\_CMD 프로시저를 사용하는 ADD CONTACTGROUP 명령

로컬 시스템에 정의된 그룹 목록에 새 문의처 그룹을 추가합니다. 문의처 그룹은 스케줄러 및 Health Monitor와 같은 모니터링 프로세스가 메시지를 보낼 수 있는 사용자 및 그룹의 목록입니다. DAS(Database Administration Server) **contact\_host** 구성 매개변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

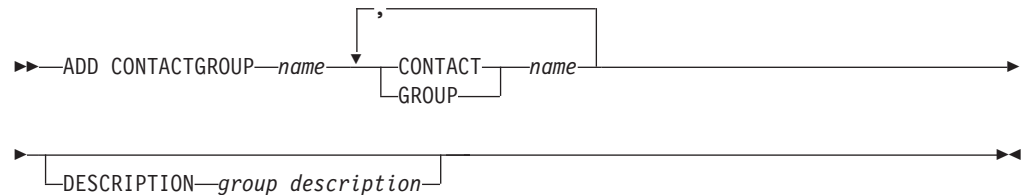
### 권한 부여

없음

### 필수 연결

데이터베이스. DAS가 실행하고 있어야 합니다.

### 명령 구문



### 명령 매개변수

#### ADD CONTACTGROUP *name*

새 문의처 그룹 이름으로, 시스템에 있는 그룹 세트 중에서 고유해야 합니다.

#### CONTACT *name*

그룹의 구성원인 문의처 이름. 문의처를 그룹에 추가한 후 ADD CONTACT 명령을 사용하여 정의할 수 있습니다.

#### GROUP *name*

이 그룹이 구성원인 문의처 그룹 이름

#### DESCRIPTION *group description*

선택사항. 문의처 그룹의 텍스트 설명.

예 :

두 개의 문의처 'cname1' 및 'cname2'를 포함하는 'gname1'이라는 문의처 그룹을 작성하십시오.

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

## 사용 시 참고사항

DAS를 작성하여 실행해야 합니다.

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

## ADMIN\_CMD 프로시저를 사용하는 AUTOCONFIGURE 명령

이들 권장 값을 적용하는 옵션을 사용하여 버퍼 풀 크기, 데이터베이스 구성 및 데이터베이스 관리 프로그램 구성 매개변수에 대한 초기값을 개선하여 표시합니다.

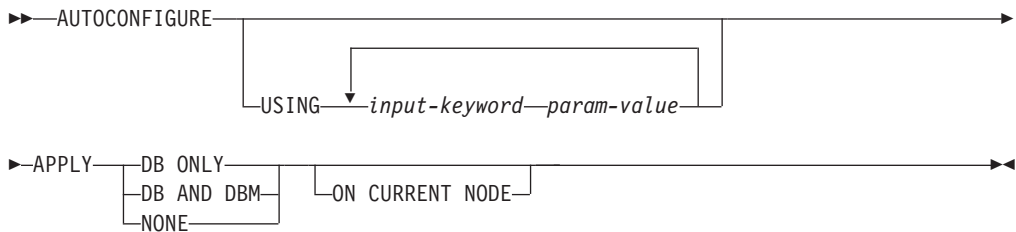
### 권한 부여

*sysadm*

### 필수 연결

데이터베이스

### 명령 구문



### 명령 매개변수

**USING** *input-keyword param-value*

표 27. 유효한 입력 키워드 및 매개변수 값

키워드	유효한 값	디폴트값	설명
mem_percent	1-100	25	전용으로 사용되는 인스턴스 메모리의 초기 백분율. 시스템에서 현재 메모리 사용에 따라 연속 데이터베이스에 대해 구성 어드바이저가 이 값을 조정합니다.

표 27. 유효한 입력 키워드 및 매개변수 값 (계속)

키워드	유효한 값	디폴트값	설명
workload_type	simple, mixed, complex	mixed	단순 워크로드는 I/O 집 중적이고 주로 트랜잭션인 경향이 있는 반면, 복합 워크로드는 CPU 집중적 이며 주로 쿼리로 이루어 지는 경향이 있습니다.
num_stmts	1-1 000 000	10	작업 단위(UOW) 당 명 령문 수
tpm	1-200 000	60	분 당 트랜잭션 수
admin_priority	performance, recovery, both	both	성능 향상(분 당 더 많은 트랜잭션 수) 또는 보다 짧은 복구 시간을 위한 최적화
is_populated	yes, no	yes	데이터베이스가 데이터로 채워지는지 여부
num_local_apps	0-5 000	0	연결된 로컬 응용프로그램 수
num_remote_apps	0-5 000	10	연결된 리모트 응용프로그 램 수
isolation	RR, RS, CS, UR	RR	이 데이터베이스에 연결 중인 응용프로그램의 최대 분리 레벨(반복 읽기, 읽 기 안정성, 커서 안정성, 언커미트 읽기). 다른 구 성 매개변수의 값을 판별 하는 데만 사용됩니다. 응 용프로그램을 특정 분리 레벨로 제한하려면 아무 것도 설정하지 않으며 디 폴트값을 사용하는 것이 안전합니다.
bp_resizeable	yes, no	yes	버퍼 풀의 크기를 조절할 수 있는지 여부

## APPLY

### DB ONLY

현재 데이터베이스 관리 프로그램 구성에 기초하여 데이터베이스 구성 및 버퍼 풀 설정에 대한 권장 값을 표시합니다. 데이터베이스 구성 및 버퍼 풀 설정에 대한 권장 변경사항을 적용합니다.

### DB AND DBM

데이터베이스 관리 프로그램 구성, 데이터베이스 구성 및 버퍼 풀 설정에 대한 권장 변경사항을 표시하고 적용합니다.

## NONE

권장 변경사항을 표시하지만 이를 적용하지 않습니다.

## ON CURRENT NODE

DPF(Database Partitioning Feature)에서 구성 어드바이저가 디폴트로 모든 노드의 데이터베이스 구성을 갱신합니다. ON CURRENT NODE 옵션을 사용하여 실행하면 어드바이저가 권장 데이터베이스 구성을 코디네이터(연결) 노드에만 적용합니다.

버퍼 풀 변경사항은 시스템 카탈로그에 항상 적용됩니다. 따라서 모든 노드가 영향을 받습니다. ON CURRENT NODE 옵션은 버퍼 풀 권장사항에 중요하지 않습니다.

## 예 :

ADMIN\_CMD 스토어드 프로시저를 통해 데이터베이스에 대해 Autoconfigure를 호출합니다.

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

다음은 명령에서 리턴된 결과 세트의 예입니다.

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

## 사용 시 참고사항

- 다중 로컬 파티션이 있는 시스템에서 **mem\_percent** 매개변수는 모든 논리 파티션에서 사용될 메모리의 백분율을 나타냅니다. 예를 들어, DB2가 시스템에서 25%의 메모리를 사용할 경우, 논리 파티션 수에 관계없이 25%를 지정하십시오. 그러나 데이터베이스 구성 권장사항은 한 논리 파티션에 대해 조정됩니다.
- 이 명령은 현재 연결된 데이터베이스만 시스템에서 사용 중이라고 가정하고 이 데이터베이스에 대한 구성 권장사항을 작성합니다. 시스템에서 두 개 이상의 데이터베이스를 사용 중인 경우 **mem\_percent** 매개변수를 조정하여 현재 데이터베이스의 메모리 공유를 반영하십시오. 예를 들어, DB2 데이터베이스가 시스템 메모리의 80%를 사용하며 시스템에 있는 두 개의 사용 중인 데이터베이스가 자원을 동일하게 공유해야 하는 경우 **mem\_percent** 매개변수에 대해 40%(80%를 두 개의 데이터베이스로 나눔)를 지정하십시오.
- AUTOCONFIGURE 명령으로 구성 어드바이저를 명시적으로 호출할 때 DB2\_ENABLE\_AUTOCONFIG\_DEFAULT 레지스트리 변수의 설정을 무시합니다.

- 데이터베이스에 대해 AUTOCONFIGURE 명령을 실행시 자체 성능 조정 메모리 관리자를 사용 가능하게 할 것을 권장합니다. 그러나 SHEAPTHRES가 0이 아닌 인스턴스에서 데이터베이스에 대해 AUTOCONFIGURE 명령을 실행할 경우, 정렬 메모리 조정(SORTHEAP)은 자동으로 사용할 수 있게 되지 않습니다. 정렬 메모리 조정(SORTHEAP)을 사용할 수 있게 하려면 UPDATE DATABASE MANAGER CONFIGURATION 명령을 사용하여 SHEAPTHRES를 0으로 설정해야 합니다. SHEAPTHRES의 값을 변경하면 이전의 기존 데이터베이스에서의 정렬 메모리 사용에 영향을 줄 수 있다는 점을 참고하십시오.
- 명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.
- AUTOCONFIGURE 대신 ADMIN\_CMD 프로시저에서 실행되는 SQL은 Query Patroller에 의해 모니터링됩니다.
- AUTOCONFIGURE 명령은 실행이 종료될 때 COMMIT문을 발행합니다. 유형- 2 연결의 경우, 이 명령으로 인해 ADMIN\_CMD 프로시저가 이유 코드 2와 함께 SQL30090N을 리턴합니다.

## 결과 세트 정보

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다. 실행이 완료되면 명령은 다음 결과 세트로 추가 정보를 리턴합니다.

표 28. AUTOCONFIGURE 명령에서 리턴된 결과 세트

컬럼 이름	데이터 유형	설명
LEVEL	VARCHAR(3)	매개변수 레벨이며 다음과 같습니다. <ul style="list-style-type: none"> <li>• BP - 버퍼 풀 레벨</li> <li>• DBM - 데이터베이스 관리 프로그램 레벨</li> <li>• DB - 데이터베이스 레벨</li> </ul>
NAME	VARCHAR(128)	<ul style="list-style-type: none"> <li>• LEVEL이 DB 또는 DBM인 경우 이 값은 구성 매개변수 키워드를 포함합니다.</li> <li>• LEVEL이 BP인 경우 이 값은 버퍼 풀 이름을 포함합니다.</li> </ul>

표 28. AUTOCONFIGURE 명령에서 리턴된 결과 세트 (계속)

컬럼 이름	데이터 유형	설명
VALUE	VARCHAR (256)	<ul style="list-style-type: none"> <li>• LEVEL이 DB 또는 DBM이고 권장 값이 적용된 경우, 이 컬럼은 권장 값을 적용하기 전에 NAME 컬럼에 식별된 구성 매개변수의 값을 포함합니다 (즉, 이전 값을 포함합니다). 변경을 적용하지 않은 경우, 이 컬럼은 식별된 구성 매개변수의 현재 디스크에 있는 값(지연 값)을 포함합니다.</li> <li>• LEVEL이 BP이고 권장 값이 적용된 경우, 이 컬럼은 권장 값을 적용하기 전에 NAME 컬럼에 식별된 버퍼 풀의 크기(페이지 단위)를 포함합니다(즉 이전 크기를 포함합니다). 변경을 적용하지 않은 경우, 이 컬럼은 식별된 버퍼 풀의 현재 크기(페이지 단위)를 포함합니다.</li> </ul>
RECOMMENDED_VALUE	VARCHAR (256)	<ul style="list-style-type: none"> <li>• LEVEL이 DB 또는 DBM인 경우 이 컬럼은 매개변수 컬럼에 식별된 구성 매개변수의 권장(또는 적용된) 값을 포함합니다.</li> <li>• 유형이 BP인 경우 이 컬럼은 매개변수 컬럼에 식별된 버퍼 풀의 권장(또는 적용된) 크기(페이지 단위)를 포함합니다.</li> </ul>
DATATYPE	VARCHAR (128)	매개변수 데이터 유형

## ADMIN\_CMD 프로시저를 사용하는 BACKUP DATABASE 명령

데이터베이스 또는 테이블 스페이스의 백업 사본을 작성합니다.

다른 운영 체제 및 하드웨어 플랫폼 간에 DB2 데이터베이스 시스템에서 제공하는 백업 조작에 대한 정보를 보려면 『다른 운영 체제 및 하드웨어 플랫폼 간의 백업 및 리스토어 조작』을 참조하십시오.

### 범위

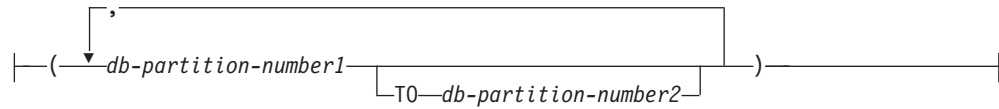
파티션된 데이터베이스 환경에서 데이터베이스 파티션이 지정되지 않으면, 이 명령은 명령이 실행되는 데이터베이스 파티션에만 영향을 미칩니다.

파티션된 백업을 수행하는 옵션을 지정하면, 카탈로그 노드에서만 명령을 호출할 수 있습니다. 모든 데이터베이스 파티션 서버를 백업하도록 옵션을 지정한 경우, db2nodes.cfg 파일에 나열된 모든 데이터베이스 파티션에 영향을 미칩니다. 그렇지 않으면, 명령에서





### Partition number(s):



### Open sessions:



### Options:



## 명령 매개변수

### DATABASE | DB *database-alias*

백업할 데이터베이스의 별명을 지정합니다. 별명은 서버에 정의된 로컬 데이터베이스여야 하며 사용자가 현재 연결되어 있는 데이터베이스 이름이어야 합니다. `database-alias`가 사용자가 연결되어 있는 데이터베이스가 아닐 경우 SQL20322N 오류가 리턴됩니다.

**ON** 데이터베이스 파티션 세트의 데이터베이스를 백업합니다. 이 절은 카탈로그 파티션에서만 지정됩니다.

### DBPARTITIONNUM *db-partition-number1*

데이터베이스 파티션 목록의 데이터베이스 파티션 번호를 지정합니다.

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

데이터베이스 파티션 번호 범위를 지정하여, *db-partition-number1*부터 *db-partition-number2*까지의 모든 파티션을 데이터베이스 파티션 목록에 포함합니다.

### ALL DBPARTITIONNUMS

`db2nodes.cfg` 파일에 지정된 모든 파티션에서 데이터베이스를 백업하도록 지정합니다.

### EXCEPT

데이터베이스 파티션 목록에 지정된 파티션을 제외하고, `db2nodes.cfg` 파일에 지정된 모든 파티션에서 데이터베이스를 백업하도록 지정합니다.

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

데이터베이스 파티션 목록의 데이터베이스 파티션 번호를 지정합니다.

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

데이터베이스 파티션 번호 범위를 지정하여, *db-partition-number1*부터 *db-partition-number2*까지의 모든 파티션을 데이터베이스 파티션 목록에 포함합니다.

**TABLESPACE** *tablespace-name*

백업될 테이블 스페이스를 지정하는 데 사용되는 이름 목록.

**ONLINE**

온라인 백업을 지정합니다. 이는 유일하게 지원되는 모드이며 디폴트값입니다. ONLINE절은 지정할 필요가 없습니다.

**INCREMENTAL**

누적(중분) 백업 이미지를 지정합니다. 중분 백업 이미지는 가장 최근에 완료된 전체 백업 조작 이후 변경된 모든 데이터베이스 데이터의 사본입니다.

**DELTA**

비누적(델타) 백업 이미지를 지정합니다. 델타 백업 이미지는 가장 최근에 완료된 모든 유형의 백업 조작 이후 변경된 모든 데이터베이스 데이터의 사본입니다.

**USE**

**TSM** 백업에 TSM(Tivoli® Storage Manager) 출력이 사용되도록 지정합니다.

**XBSA** XBSA 인터페이스가 사용되도록 지정합니다. 백업 서비스 API(XBSA)는 백업 또는 아카이브용으로 데이터 스토리지 관리를 필요로 하는 응용프로그램 또는 기능에 대한 개방 API입니다.

**SNAPSHOT**

스냅샷 백업이 사용되도록 지정합니다.

다음 매개변수가 있는 SNAPSHOT 매개변수를 사용할 수 없습니다.

- TABLESPACE
- INCREMENTAL
- WITH *num-buffers* BUFFERS
- BUFFER
- PARALLELISM
- COMPRESS

- UTIL\_IMPACT\_PRIORITY
- SESSIONS

스냅샷 백업의 디폴트 동작은 모든 경로의 FULL DATABASE OFFLINE 백업으로, 모든 컨테이너, 로컬 볼륨 디렉토리, 데이터베이스 경로(DBPATH) 및 1차 로그 및 미러 로그 경로를 작성합니다 (EXCLUDE LOGS가 명시적으로 지정된 경우를 제외하고 INCLUDE LOGS는 모든 스냅샷 백업의 디폴트임).

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

IBM Data Server에 통합되며 다음 스토리지 하드웨어의 DB2 ACS API 드라이버입니다.

- IBM TotalStorage SAN 볼륨 제어기
- IBM 엔터프라이즈 스토리지 서버 모델 800
- IBM 시스템 스토리지 DS6000
- IBM 시스템 스토리지 DS8000
- IBM 시스템 스토리지 N 시리즈
- NetApp V-series
- NetApp FAS

다른 스토리지 하드웨어가 있으며 해당 스토리지 하드웨어의 DB2 ACS API 드라이버가 있는 경우, LIBRARY 매개변수를 사용하여 DB2 ACS API 드라이버를 지정할 수 있습니다.

LIBRARY 매개변수의 값은 완전한 라이브러리 파일 이름입니다.

## **OPTIONS**

### *"options-string"*

백업 작업에 사용되는 옵션을 지정합니다. 이 문자열은 큰따옴표 없이 입력된 그대로 정확히 DB2 ACS API 드라이버로 전달됩니다. **VENDOROPT** 데이터베이스 구성 매개변수를 사용하여 스냅샷 백업 작업을 위한 벤더 고유의 옵션을 지정할 수 없습니다. 대신 백업 유틸리티의 **OPTIONS** 매개변수를 사용해야 합니다.

### *@ file-name*

백업 작업에 사용될 옵션이 DB2 서버에 있는 파일에 들어 있음을 지정합니다. 이 문자열은 벤더 지원 라이브러리로 전달됩니다. 파일은 완전한 파일 이름이어야 합니다.

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

DB2와 TSM 또는 다른 백업 벤더 제품 사이에 작성될 I/O 세션 수입니다. 이 매개변수는 테이프, 디스크 또는 다른 로컬 디바이스에 백업할 때는 유효하지 않습니다.

**TO** *dir | dev*

디렉토리 또는 테이프 디바이스 이름 목록입니다. 디렉토리가 있는 전체 경로를 지정해야 합니다. 이 목표 디렉토리 또는 디바이스는 데이터베이스 서버에 있어야 합니다.

파티션된 데이터베이스에서, 대상 디렉토리나 디바이스는 모든 데이터베이스 파티션에 있어야 하며, 선택적으로 공유 경로일 수 있습니다. 디렉토리나 디바이스는 데이터베이스 파티션 표현식을 사용하여 지정될 수 있습니다. 데이터베이스 파티션 표현식에 대한 자세한 정보는 자동 스토리지 데이터베이스를 참조하십시오.

백업 이미지가 위치할 목표 디렉토리 및 디바이스를 지정하기 위해 이 매개변수를 반복할 수 있습니다. 두 개 이상의 목표가 지정될 경우(예: target1, target2, target3) target1이 첫 번째로 열립니다. 미디어 헤더 및 특수 파일(구성 파일, 테이블 스페이스 테이블 및 실행기록 파일을 포함하여)은 target1에 놓입니다. 백업 조작 중 나머지 모든 목표를 열고 병렬로 사용합니다. Windows® 운영 체제에서는 일반 테이프가 지원되지 않으므로 각 유형의 테이프 디바이스에는 고유 디바이스 드라이버가 필요합니다.

테이프 디바이스 또는 플로피 디스크를 사용하려면 프롬프트 및 사용자 상호작용이 필요할 수 있는데, 이 때 오류가 리턴됩니다.

테이프 시스템이 백업 이미지를 고유하게 참조할 수 있는 기능을 지원하지 않는 경우, 같은 데이터베이스의 여러 백업 사본을 동일한 테이프에 보존하지 않는 것이 바람직합니다.

**LOAD** *library-name*

사용될 벤더 백업 및 리스토어 I/O 기능이 들어 있는 공유 라이브러리 이름 (Windows 운영 체제의 DLL). 여기에는 전체 경로가 포함될 수 있습니다. 전체 경로가 제공되지 않을 경우 디폴트 경로는 User Exit 프로그램이 있는 경로입니다.

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

사용될 버퍼 수. DB2는 사용자가 명시적으로 값을 입력하는 경우를 제외하고는 자동으로 이 매개변수에 대한 최적 값을 선택합니다. 그러나 다중 위치로 백업을 생성할 때 성능을 개선하기 위해 더 많은 버퍼 수를 사용할 수 있습니다.

**BUFFER** *buffer-size*

백업 이미지를 빌드할 때 사용되는 버퍼의 크기(4KB 페이지 단위). DB2는 사

용자가 명시적으로 값을 입력하는 경우를 제외하고는 자동으로 이 매개변수에 대한 최적 값을 선택합니다. 이 매개변수의 최소값은 8페이지입니다.

가변 블록 크기를 갖는 테이프를 사용할 경우, 버퍼 크기를 테이프 디바이스가 지원하는 범위 내로 줄이십시오. 그렇지 않으면, 백업 조작은 계속할 수 있지만 결과 이미지를 복구하지 못할 수 있습니다.

Linux<sup>®</sup>의 대부분의 버전에서 SCSI 테이프 디바이스로 백업 조작에 DB2의 디폴트 버퍼 크기를 사용하면 SQL2025N 오류, 이유 코드 75가 리턴됩니다. Linux 내부 SCSI 버퍼가 오버플로우되지 않게 하려면 다음 공식을 사용하십시오.

$$\text{bufferpages} \leq \text{ST\_MAX\_BUFFERS} * \text{ST\_BUFFER\_BLOCKS} / 4$$

여기서 *bufferpages*는 BUFFER 매개변수에 대해 사용하려는 값이고 ST\_MAX\_BUFFERS 및 ST\_BUFFER\_BLOCKS는 drivers/scsi 디렉토리 아래의 Linux 커널에 정의됩니다.

### **PARALLELISM** *n*

백업 유틸리티에서 병렬로 읽을 수 있는 테이블 스페이스 수를 결정합니다. DB2는 사용자가 명시적으로 값을 입력하는 경우를 제외하고는 자동으로 이 매개변수에 대한 최적 값을 선택합니다.

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

백업이 지정된 우선순위를 사용하여 조절 모드로 실행하도록 지정합니다. 조절을 사용하면 백업 조작의 성능 영향을 조정할 수 있습니다. 우선순위는 1 - 100의 숫자를 사용할 수 있는데, 1은 최저 우선순위를 나타내고 100은 최고 우선순위를 나타냅니다. UTIL\_IMPACT\_PRIORITY 키워드를 우선순위 없이 지정할 경우 백업은 디폴트 우선순위인 50으로 실행됩니다.

UTIL\_IMPACT\_PRIORITY를 지정하지 않으면 백업은 조절되지 않은 모드로 실행됩니다. 백업을 조절 모드로 실행되게 하려면 *util\_impact\_lim* 구성 매개변수를 설정하여 영향 규정을 정의해야 합니다.

### **COMPRESS**

백업을 압축할 것을 표시합니다.

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

압축을 수행하는 데 사용될 라이브러리 이름을 표시합니다(예: Windows의 경우 db2compr.dll, Linux/UNIX 시스템의 경우 libdb2compr.so). 이 이름은 서버에 있는 파일을 나타내는 완전한 경로여야 합니다. 이 매개변수를 지정하지 않으면 디폴트 DB2 압축 라이브러리를 사용합니다. 지정된 라이브러리를 로드할 수 없으면 백업에 실패합니다.

#### **EXCLUDE**

압축 라이브러리가 백업 이미지에 저장되지 않을 것을 표시합니다.

#### **COMPROPTS** *string*

압축 라이브러리의 초기화 루틴으로 전달될 2진 데이터 블록을 설명함

니다. DB2는 클라이언트에서 서버로 직접 이 문자열을 전달하므로 모든 바이트 리버설 또는 코드 페이지 변환 문제는 압축 라이브러리에서 처리되어야 합니다. 데이터 블록의 첫 번째 문자가 '@'일 경우, DB2는 이 데이터의 나머지 부분을 서버에 있는 파일 이름으로 해석합니다. 그런 다음 DB2는 문자열의 내용을 이 파일의 내용과 바꾼 후 대신 이 새 값을 초기화 루틴으로 전달합니다. *string*의 최대 길이는 1024바이트입니다.

## EXCLUDE LOGS

백업 이미지가 로그 파일을 포함해서는 안된다는 것을 지정합니다. 오프라인 백업 조사를 수행할 때, 스냅샷 백업 예외를 사용하여 이 옵션이 지정되었는지 여부에 관계없이 로그를 제외합니다.

## INCLUDE LOGS

백업 이미지가 이 이미지를 어떤 일관된 특정 시점으로 리스토어 및 롤 포워드하는 데 필요한 일련의 로그 파일을 반드시 포함하도록 지정합니다. 명시적으로 제외하도록 명시한 경우를 제외하고 이 옵션이 디폴트인 경우, 스냅샷 백업의 예외가 있는 오프라인 백업에는 이 옵션이 유효하지 않습니다.

## WITHOUT PROMPTING

백업이 자동으로 실행되고 일반적으로 사용자 개입이 필요한 모든 조치는 오류 메시지를 리턴할 것을 지정합니다. 이것은 디폴트값입니다.

## 예:

다음은 복구 가능한 데이터베이스에 대한 주별 증분 백업 전략 샘플입니다. 주별 전체 데이터베이스 백업 조작과 일별 비누적(델타) 백업 조작 및 주중 누적(증분) 백업 조작이 포함됩니다.

```
(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')
```

## 사용 시 참고사항

백업의 데이터는 데이터베이스 서버의 보호를 받을 수 없습니다. 특히 백업에 LBAC 보호 데이터가 포함된 경우 백업이 올바르게 보호되는지 확인하십시오.

테이프에 백업시 가변 블록 크기의 사용은 현재 지원되지 않습니다. 이 옵션을 사용해야 하는 경우, 가변 블록 크기로 작성된 백업 이미지를 사용하여 성공적으로 복구할 수 있는 제대로 테스트된 프로시저가 있는지 확인하십시오.

가변 블록 크기 사용시 사용 중인 테이프 디바이스에 대해 최대 한계 이하의 백업 버퍼 크기를 지정해야 합니다. 최적 성능을 위해서는 버퍼 크기를 사용 중인 디바이스의 최대 블록 크기 한계와 동일하게 지정해야 합니다.

파일러/스토리지 시스템에서 실패하는 경우 스냅샷 백업은 일반 디스크 백업으로 보완되어야 합니다.

정기적으로 데이터베이스를 백업할 때, 매우 큰 데이터베이스 백업 이미지, 여러 데이터베이스 로그 및 로드 사본 이미지를 축적할 수 있으며, 이는 모두 많은 디스크 스페이스를 차지할 수 있습니다. 이러한 복구 오브젝트를 관리하는 방법에 대한 정보는 『복구 오브젝트 관리』를 참조하십시오.

## 결과 세트 정보

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다. 실행이 완료되면 명령은 추가 정보를 리턴합니다. 백업 작업은 하나의 결과 세트를 리턴하며, 백업에 참여한 데이터베이스 파티션당 한 행을 포함합니다.

표 29. 백업 작업의 결과 세트

컬럼 이름	데이터 유형	설명
BACKUP_TIME	VARCHAR(14)	백업 이미지의 이름을 지정하는데 사용되는 시간소인 문자열에 해당합니다.
DBPARTITIONNUM	SMALLINT	에이전트가 백업 작업을 실행한 데이터베이스 파티션 번호.
SQLCODE	INTEGER	지정된 데이터베이스 파티션의 백업 처리 결과인 최종 SQLCODE.
SQLERRMC	VARCHAR(70)	지정된 데이터베이스 파티션의 백업 처리 결과인 최종 SQLERRMC.
SQLERRML	SMALLINT	지정된 데이터베이스 파티션의 백업 처리 결과인 최종 SQLERRML.

파티션되지 않은 데이터베이스가 백업되거나, 기존의 단일 파티션 구문을 사용하여 파티션된 데이터베이스가 백업되는 경우, 결과 세트는 단일 행을 포함합니다.

DBPARTITIONNUM은 백업 중인 데이터베이스 파티션의 ID 번호를 포함합니다.

SQLCODE, SQLERRMC 및 SQLERRML은 지정된 데이터베이스 파티션에서 백업에 의해 리턴된 SQLCA의 동일 구성원을 나타냅니다.



## ADMIN\_CMD 프로시저를 사용하는 DESCRIBE 명령

DESCRIBE 명령은 다음 항목에 대한 정보를 표시합니다.

- SELECT, CALL 또는 XQuery문의 출력
- 테이블 또는 뷰의 컬럼
- 테이블 또는 뷰의 인덱스
- 테이블 또는 뷰의 데이터 파티션

### 권한 부여

필요한 권한 부여는 DESCRIBE 명령을 사용하여 표시하려는 정보 유형에 따릅니다.

- SYSTOOLSTMPSPACE 테이블 스페이스가 있는 경우, 다음 표에 표시된 권한 중 하나가 필요합니다.

정보 표시를 위한 오브젝트	필요한 특권 또는 권한
SELECT문 또는 XQuery문의 출력	SELECT문에서 참조되는 각 테이블이나 뷰에 대해 다음 특권이나 권한 중 하나: <ul style="list-style-type: none"> <li>• SELECT 특권</li> <li>• DATAACCESS 권한</li> <li>• DBADM 권한</li> <li>• SQLADM 권한</li> <li>• EXPLAIN 권한</li> </ul>
CALL문의 출력	다음 특권이나 권한 중 하나: <ul style="list-style-type: none"> <li>• DATAACCESS 권한</li> <li>• 스토어드 프로시저에 대한 EXECUTE 특권</li> </ul>

정보 표시를 위한 오브젝트	필요한 특권 또는 권한
<p>테이블 또는 뷰의 컬럼</p>	<p>SYSCAT.COLUMNS 시스템 카탈로그 테이블에 대해 다음 특권이나 권한 중 하나:</p> <ul style="list-style-type: none"> <li>• SELECT 특권</li> <li>• ACCESSCTRL 권한</li> <li>• DATAACCESS 권한</li> <li>• DBADM 권한</li> <li>• SECADM 권한</li> <li>• SQLADM 권한</li> </ul> <p>SHOW DETAIL 매개변수를 사용하려는 경우, SYSCAT.DATAPARTITIONEXPRESSION 시스템 카탈로그 테이블에 대해서도 해당 특권이나 권한이 필요합니다.</p> <p>PUBLIC은 선언된 임시 테이블에 대한 모든 특권을 소유하므로 사용자는 이 명령을 사용하여 해당 연결 내에 존재하는 선언된 임시 테이블에 대한 정보를 표시할 수 있습니다.</p>
<p>테이블 또는 뷰의 인덱스</p>	<p>SYSCAT.INDEXES 시스템 카탈로그 테이블에 대해 다음 특권이나 권한 중 하나:</p> <ul style="list-style-type: none"> <li>• SELECT 특권</li> <li>• ACCESSCTRL 권한</li> <li>• DATAACCESS 권한</li> <li>• DBADM 권한</li> <li>• SECADM 권한</li> <li>• SQLADM 권한</li> </ul> <p>SHOW DETAIL 매개변수를 사용하려는 경우, GET_INDEX_COLNAMES() UDF에 대해서도 EXECUTE 특권이 필요합니다.</p> <p>PUBLIC은 선언된 임시 테이블에 대한 모든 특권을 소유하므로 사용자는 이 명령을 사용하여 해당 연결 내에 존재하는 선언된 임시 테이블에 대한 정보를 표시할 수 있습니다.</p>

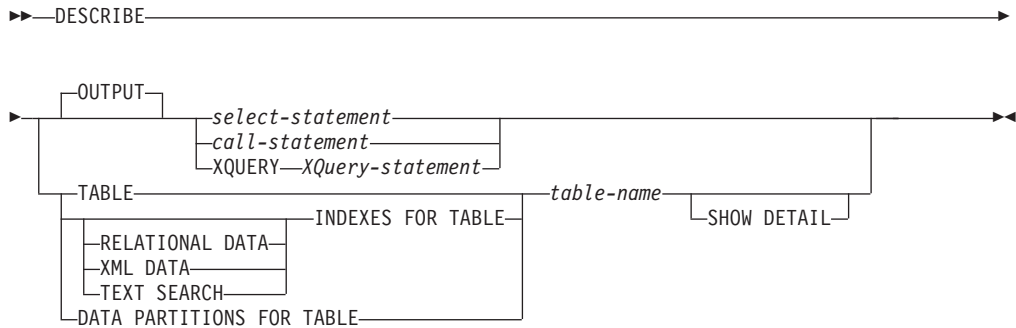
정보 표시를 위한 오브젝트	필요한 특권 또는 권한
테이블 또는 뷰의 데이터 파티션	<p>SYSCAT.DATAPARTITIONS 시스템 카탈로그 테이블에 대해 다음 특권이나 권한 중 하나:</p> <ul style="list-style-type: none"> <li>• SELECT 특권</li> <li>• ACCESSCTRL 권한</li> <li>• DATAACCESS 권한</li> <li>• DBADM 권한</li> <li>• SECADM 권한</li> <li>• SQLADM 권한</li> </ul> <p>PUBLIC은 선언된 임시 테이블에 대한 모든 특권을 소유하므로 사용자는 이 명령을 사용하여 해당 연결 내에 존재하는 선언된 임시 테이블에 대한 정보를 표시할 수 있습니다.</p>

- SYSTOOLSTMPSPACE 테이블 스페이스가 없는 경우, 위의 권한 중 하나 뿐만 아니라 SYSADM 또는 SYSCTRL 권한이 필요합니다.

## 필수 연결

데이터베이스

## 명령 구문



## 명령 매개변수

### OUTPUT

명령문의 출력을 설명해야 함을 나타냅니다. 이 키워드는 선택적입니다.

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

필요한 정보에 대한 명령문을 식별합니다. 명령문은 CLP에 의해 자동으로 준비됩니다. XQuery문을 식별하려면 명령문 앞에 키워드 XQUERY를 붙이십시오. DESCRIBE OUTPUT문은 기술된 쿼리의 최종 결과 테이블의 SELECT 목록 파트로 컬럼이 명시적으로 지정된 경우 내재적으로 숨겨진 컬럼에 대한 정보만을 리턴합니다.

## **TABLE** *table-name*

설명될 테이블 또는 뷰를 지정합니다. *schema.table-name* 양식의 완전한 이름을 사용해야 합니다. 실제 테이블 대신 테이블의 별명을 사용할 수 없습니다. 내재적으로 숨겨진 컬럼에 대한 정보가 리턴됩니다.

DESCRIBE TABLE 명령은 각 컬럼에 대해 다음 정보를 나열합니다.

- 컬럼 이름
- 유형 스키마
- 유형 이름
- 길이
- 스케일
- 널(NULL)(예/아니오)

## **INDEXES FOR TABLE** *table-name*

인덱스가 설명되어야 하는 테이블 또는 뷰를 지정합니다. *schema.table-name* 양식의 완전한 이름을 사용하거나 *table-name*만 지정할 수 있습니다. 이 경우 자동으로 디폴트 스키마가 사용됩니다. 실제 테이블 대신 테이블의 별명을 사용할 수 없습니다.

DESCRIBE INDEXES FOR TABLE 명령은 테이블 또는 뷰의 각 인덱스에 대해 다음 정보를 나열합니다.

- 인덱스 스키마
- 인덱스 이름
- 고유한 규칙
- 컬럼 수
- 인덱스 유형

DESCRIBE INDEXES FOR TABLE 명령이 SHOW DETAIL 옵션과 함께 지정되면, 인덱스 이름이 18바이트보다 큰 경우 잘립니다. 인덱스 유형 옵션이 지정되지 않으면, 모든 인덱스 유형의 정보가 나열됩니다(관계형 데이터 인덱스, XML 데이터에 있는 인덱스 및 텍스트 검색 인덱스). 출력은 다음과 같은 추가 정보를 포함합니다.

- 관계형 데이터 인덱스, XML 경로 인덱스, XML 영역 인덱스 또는 XML 데이터에 있는 인덱스의 인덱스 ID
- XML 데이터에 있는 인덱스의 데이터 유형
- XML 데이터에 있는 인덱스의 해시 정보
- XML 데이터에 있는 인덱스의 최대 VARCHAR 길이
- XML 데이터에 있는 인덱스의 지정 XML 패턴
- 텍스트 검색 인덱스의 코드 페이지

- 텍스트 검색 인덱스의 언어
- 텍스트 검색 인덱스의 지정 형식
- 텍스트 검색 인덱스의 갱신 최소치
- 텍스트 검색 인덱스의 갱신 빈도
- 텍스트 검색 인덱스의 콜렉션 디렉토리
- 컬럼 이름

특정 인덱스 유형 전용 정보를 나열하려면 인덱스 유형을 지정하십시오. 다중 인덱스 유형 지정은 지원되지 않습니다.

### **RELATIONAL DATA**

RELATIONAL DATA 인덱스 유형 옵션이 SHOW DETAIL 옵션 없이 지정되면, 다음 정보만이 나열됩니다.

- 인덱스 스키마
- 인덱스 이름
- 고유한 규칙
- 컬럼 수

SHOW DETAIL이 지정되면, 컬럼 이름 정보도 나열됩니다.

### **XML DATA**

XML DATA 인덱스 유형 옵션이 SHOW DETAIL 옵션 없이 지정되면, 다음 정보만이 나열됩니다.

- 인덱스 스키마
- 인덱스 이름
- 고유한 규칙
- 컬럼 수
- 인덱스 유형

SHOW DETAIL이 지정되면, XML 데이터에 있는 인덱스의 다음 정보도 나열됩니다.

- 인덱스 ID
- 데이터 유형
- 해시
- 최대 Varchar 길이
- XML 패턴
- 컬럼 이름

## TEXT SEARCH

TEXT SEARCH 인덱스 유형 옵션이 SHOW DETAIL 옵션 없이 지정되면, 다음 정보만이 나열됩니다.

- 인덱스 스키마
- 인덱스 이름

SHOW DETAIL이 지정되면, 다음과 같은 텍스트 검색 인덱스 정보도 나열됩니다.

- 컬럼 이름
- 코드 페이지
- 언어
- 형식
- 갱신 최소치
- 갱신 빈도
- 콜렉션 디렉토리

TEXT SEARCH 옵션이 지정되고 텍스트 검색 옵션이 설치되지 않거나 적절하게 구성되지 않은 경우, 오류(SQLSTATE 42724)가 리턴됩니다.

컬럼에 나열된 정보는 DB2 텍스트 검색을 참조하십시오.

## DATA PARTITIONS FOR TABLE *table-name*

데이터 파티션이 설명되어야 하는 테이블 또는 뷰를 지정합니다. 테이블의 각 데이터 파티션에 대해 표시되는 정보는 파티션 ID 및 파티션 간격을 포함합니다. 결과는 파티션 ID 순서에 따라 정렬됩니다. *schema.table-name* 양식의 완전한 이름을 사용해야 합니다. 실제 테이블 대신 테이블의 별명을 사용할 수 없습니다. *schema*는 테이블 또는 뷰가 작성된 사용자 이름입니다.

DESCRIBE DATA PARTITIONS FOR TABLE 명령의 경우 출력에 다음 추가 정보가 있는 2차 테이블이 포함되도록 지정합니다.

- 데이터 파티션 시퀀스 ID
- SQL의 데이터 파티션 표현식

## SHOW DETAIL

DESCRIBE TABLE 명령의 경우, 출력에 테이블 데이터 파티션 표현식(해당 테이블에서 데이터가 파티션되지 않은 경우 0개의 행을 리턴할 수 있음)이 들어 있는 두 번째 결과 세트와 함께 다음 추가 정보가 포함되도록 지정합니다.

- CHARACTER, VARCHAR 또는 LONG VARCHAR 컬럼이 FOR BIT DATA로 정의되었는지 여부
- 컬럼 번호

- 분산 키 순서
- 코드 페이지
- 디폴트값
- 테이블 파티션 유형(범위로 파티션된 테이블의 경우 이 출력은 원본 출력 아래에 표시됨)
- 파티션 키 컬럼(범위로 파티션된 테이블의 경우 이 출력은 원본 출력 아래에 표시됨)
- 인덱스에 사용되는 테이블 스페이스의 ID

예:

### SELECT문의 출력 설명

다음 예는 SELECT문을 설명하는 방법을 보여 줍니다.

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

다음은 이 SELECT문에 대한 출력 예입니다.

결과 세트 1

```
-----
```

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

3개의 레코드가 선택되었습니다.

리턴 상태 = 0

이 SELECT문의 출력(계속)

```
... SQLNAME_LENGTH SQLDATATYPE_NAME_DATA SQLDATATYPE_NAME_LENGTH
```

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

### 테이블 설명

파티션되지 않은 테이블을 설명합니다.

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

다음은 이 CALL문에 대한 출력 예입니다.

결과 세트 1

```
-----
```

COLNAME	TYPESHEMA	TYPENAME	FOR_BINARY_DATA	...
DEPTNUMB	SYSIBM	SMALLINT	N	...

```

DEPTNAME  SYSIBM      VARCHAR    N      ...
MANAGER   SYSIBM      SMALLINT   N      ...
DIVISION  SYSIBM      VARCHAR    N      ...
LOCATION   SYSIBM      VARCHAR    N      ...

```

5개의 레코드가 선택되었습니다.

이 CALL문의 출력(계속)

```

... 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 -

```

이 CALL문의 출력(계속)

결과 세트 2

```

-----
DATA_PARTITION_KEY_SEQ DATA_PARTITION_EXPRESSION
-----

```

0개의 레코드가 선택되었습니다.  
리턴 상태 = 0

파티션된 테이블을 설명합니다.

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

다음은 이 CALL문에 대한 출력 예입니다.

결과 세트 1

```

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

```

1개의 레코드가 선택되었습니다.

이 CALL문의 출력(계속)

```

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

```

이 CALL문의 출력(계속)

결과 세트 2

```

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

```

1개의 레코드가 선택되었습니다.

테이블 인덱스 설명





```

...          1 XML_DATA_VALUES_LOGICAL
...          1 XML_DATA_VALUES_PHYSICAL
...          1 TEXT_SEARCH
...          1 TEXT_SEARCH

```

### 데이터 파티션 설명

다음 예는 데이터 파티션을 설명하는 방법을 보여 줍니다.

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

다음은 이 CALL문에 대한 출력 예입니다.

결과 세트 1

-----

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

3개의 레코드가 선택되었습니다.

이 CALL문의 출력(계속)

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

다음 예는 'SHOW DETAIL' 절을 사용하여 데이터 파티션을 설명하는 방법을 보여 줍니다.

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

다음은 이 CALL문에 대한 출력 예입니다.

결과 세트 1

-----

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

3개의 레코드가 선택되었습니다.

리턴 상태 = 0

이 CALL문의 출력(계속)

```

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

```

이 CALL문의 출력(계속)

결과 세트 2

-----

```

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

```

3개의 레코드가 선택되었습니다.

리턴 상태 = 0

이 CALL문의 출력(계속)

```

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

```

## 사용법 참고

DESCRIBE 명령이 임시 테이블을 작성하려다 실패하면, SYSTOOLSTMPSPACE를 작성한 후 이번에는 SYSTOOLSTMPSPACE에 임시 테이블 작성을 다시 시도합니다. SYSTOOLSTMPSPACE 테이블 스페이스를 작성하려면 SYSCTRL 또는 SYSADM 권한이 필요합니다.

## 결과 세트 정보

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다. 실행이 완료되면 명령은 다음과 같이 결과 세트로 추가 정보를 리턴합니다.

- 64 페이지의 표 30: DESCRIBE *select-statement*, DESCRIBE *call-statement* 및 DESCRIBE XQUERY *XQuery-statement* 명령
- 64 페이지의 표 31: DESCRIBE TABLE 명령에 대한 결과 세트 1
- 65 페이지의 표 32: DESCRIBE TABLE 명령에 대한 결과 세트 2
- 65 페이지의 표 33: DESCRIBE INDEXES FOR TABLE 명령
- 67 페이지의 표 34: DESCRIBE DATA PARTITIONS FOR TABLE 명령에 대한 결과 세트 1
- 67 페이지의 표 35: DESCRIBE DATA PARTITIONS FOR TABLE 명령에 대한 결과 세트 2

표 30. DESCRIBE select-statement, DESCRIBE call-statement 및 DESCRIBE XQUERY XQuery-statement 명령에서 리턴된 결과 세트

컬럼 이름	데이터 유형	LOB <sup>1</sup>	설명
SQLTYPE_ID	SMALLINT	없음	SQL 디스크립터 영역(SQLDA)의 SQLTYPE 필드에 나타날 때 컬럼의 데이터 유형.
SQLTYPE	VARCHAR(257)	없음	SQLTYPE_ID 값에 해당하는 데이터 유형.
SQLLEN	INTEGER	없음	SQLDA의 SQLLEN 필드에 나타날 때 컬럼의 길이 속성.
SQLSCALE	SMALLINT	없음	10진수 값에서 소수 부분의 자리수, 다른 데이터 유형의 경우 0.
SQLNAME_DATA	VARCHAR(128)	없음	컬럼 이름
SQLNAME_LENGTH	SMALLINT	없음	컬럼 이름의 길이.
SQLDATA_TYPSHEMA	VARCHAR(128)	예	데이터 유형 스키마 이름.
SQLDATA_TYPENAME	VARCHAR(128)	예	데이터 유형 이름.

주: <sup>1</sup>: Yes는 LOB 데이터가 설명되는 경우에만 널(NULL)이 아닌 값이 리턴되는 것을 나타냅니다.

표 31. DESCRIBE TABLE 명령에서 리턴된 결과 세트 1

컬럼 이름	데이터 유형	세부사항 <sup>2</sup>	설명
COLNAME	VARCHAR(128)	없음	컬럼 이름
TYPSHEMA	VARCHAR(128)	없음	컬럼 이름이 구별된 경우 스키마 이름을 리턴합니다. 그렇지 않으면 'SYSIBM'을 리턴합니다.
TYPENAME	VARCHAR(128)	없음	컬럼 유형 이름
FOR_BINARY_DATA	CHAR(1)	예	컬럼 유형이 CHAR, VARCHAR 또는 LONG VARCHAR이고 FOR BIT DATA로 정의되었으면 'Y'를 리턴하고 그렇지 않으면 'N'을 리턴합니다.
LENGTH	INTEGER	없음	데이터의 최대 길이. DECIMAL 데이터인 경우 이는 정밀도를 나타냅니다. 구별 유형의 경우 0을 리턴합니다.
SCALE	SMALLINT	없음	DECIMAL 데이터인 경우 스케일을 나타냅니다. 다른 모든 유형의 경우 0을 리턴합니다.
NULLABLE	CHAR(1)	없음	다음 중 하나 <ul style="list-style-type: none"> <li>• 컬럼이 널(NULL) 입력 가능한 경우 'Y'</li> <li>• 컬럼이 널(NULL) 입력이 가능하지 않은 경우 'N'</li> </ul>
COLNO	SMALLINT	예	컬럼의 서수

표 31. DESCRIBE TABLE 명령에서 리턴된 결과 세트 1 (계속)

컬럼 이름	데이터 유형	세부사항 <sup>2</sup>	설명
PARTKEYSEQ	SMALLINT	예	테이블 파티션 키에서 컬럼의 서수. 컬럼이 파티션 키에 포함되지 않은 경우 널(NULL) 또는 0을 리턴하고 서브테이블 및 계층 구조 테이블의 경우 널(NULL)입니다.
CODEPAGE	SMALLINT	예	컬럼의 코드 페이지이며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>FOR BIT DATA로 정의되지 않은 컬럼에 대한 데이터베이스 코드 페이지의 값.</li> <li>그래픽 컬럼에 대한 DBCS 코드 페이지의 값.</li> <li>그렇지 않으면 0입니다.</li> </ul>
DEFAULT	VARCHAR(254)	예	테이블의 컬럼에 대한 디폴트값은 컬럼의 데이터 유형에 적합한 상수, 특수 레지스터 또는 캐스트 함수로 표현됩니다. 널(NULL)이 될 수도 있습니다.

주: <sup>2</sup>: Yes는 SHOW DETAIL 절을 사용할 때만 널(NULL)이 아닌 값이 리턴되는 것을 나타냅니다.

표 32. SHOW DETAIL 절을 사용할 때 DESCRIBE TABLE 명령에서 리턴된 결과 세트 2

컬럼 이름	데이터 유형	설명
DATA_PARTITION_KEY_SEQ	INTEGER	데이터 파티션 키 번호, 예를 들어 첫 번째 데이터 파티션 표현식의 경우 1이고 두 번째 데이터 파티션 표현식의 경우 2입니다 .
DATA_PARTITION_EXPRESSION	CLOB(32K)	SQL 구문에서 이 데이터 파티션 키에 대한 표현식

표 33. DESCRIBE INDEXES FOR TABLE 명령에서 리턴된 결과 세트

컬럼 이름	데이터 유형	세부사항 <sup>3</sup>	인덱스 유형 옵션 <sup>4, 5</sup>	설명
INDSCHEMA	VARCHAR (128)	없음	RELATIONAL DATA XML DATA TEXT SEARCH	인덱스 스키마 이름.
INDNAME	VARCHAR (128)	없음	RELATIONAL DATA XML DATA TEXT SEARCH	인덱스 이름.
UNIQUE_RULE	VARCHAR (30)	없음	RELATIONAL DATA XML DATA	다음 값 중 하나입니다. <ul style="list-style-type: none"> <li>• DUPLICATES_ALLOWED</li> <li>• PRIMARY_INDEX</li> <li>• UNIQUE_ENTRIES_ONLY</li> </ul>

표 33. DESCRIBE INDEXES FOR TABLE 명령에서 리턴된 결과 세트 (계속)

컬럼 이름	데이터 유형	세부사항 <sup>3</sup>	인덱스 유형 옵션 <sup>4, 5</sup>	설명
INDEX_PARTITIONING	CHAR(1)	없음	N/A	인덱스의 파티셔닝 특성을 식별합니다. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• N= 파티션되지 않은 인덱스</li> <li>• P= 파티션된 인덱스</li> <li>• 공백 = 인덱스가 파티션된 테이블에 없음</li> </ul>
COLCOUNT	SMALLINT	없음	RELATIONAL DATA XML DATA	키 컬럼 수 + Include 컬럼 수(있을 경우)
INDEX_TYPE	VARCHAR(30)	없음	RELATIONAL DATA XML DATA TEXT SEARCH	인덱스 유형: <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	예	RELATIONAL DATA XML DATA	관계형 데이터 인덱스, XML 경로 인덱스, XML 영역 인덱스 또는 XML 데이터에 있는 인덱스의 인덱스 ID
DATA_TYPE	VARCHAR(128)	예	XML DATA	XML 데이터에 있는 인덱스용으로 지정된 SQL 데이터 유형. 다음 값 중 하나입니다. <ul style="list-style-type: none"> <li>• VARCHAR</li> <li>• DOUBLE</li> <li>• DATE</li> <li>• TIMESTAMP</li> </ul>
HASHED	CHAR(1)	예	XML DATA	XML 데이터에 있는 인덱스의 값이 해시되는지 여부를 나타냅니다. <ul style="list-style-type: none"> <li>• 값이 해시되는 경우 'Y'.</li> <li>• 값이 해시되지 않는 경우 'N'.</li> </ul>
LENGTH	SMALLINT	예	XML DATA	XML 데이터에 있는 인덱스의 경우, VARCHAR(integer) 길이. 그 외에는 0입니다.
PATTERN	CLOB(2M)	예	XML DATA	XML 데이터에 있는 인덱스의 지정 XML 패턴 표현식
CODEPAGE	INTEGER	예	TEXT SEARCH	텍스트 검색 인덱스의 지정 문서 코드 페이지
LANGUAGE	VARCHAR(5)	예	TEXT SEARCH	텍스트 검색 인덱스의 지정 문서 언어
FORMAT	VARCHAR(30)	예	TEXT SEARCH	텍스트 검색 인덱스의 지정 문서 형식
UPDATEMINIMUM	INTEGER	예	TEXT SEARCH	중분 갱신이 수행되기 전 텍스트 검색 로그 테이블의 최소 항목 수
UPDATEFREQUENCY	VARCHAR(300)	예	TEXT SEARCH	텍스트 인덱스 갱신을 적용하기 위해 지정된 트리거 기준

표 33. DESCRIBE INDEXES FOR TABLE 명령에서 리턴된 결과 세트 (계속)

컬럼 이름	데이터 유형	세부사항 <sup>3</sup>	인덱스 유형 옵션 <sup>4, 5</sup>	설명
COLLECTION DIRECTORY	VARCHAR (512)	예	TEXT SEARCH	텍스트 검색 인덱스 파일의 지정 디렉토리
COLNAMES	VARCHAR (2048)	예	RELATIONAL DATA XML DATA TEXT SEARCH	컬럼 이름 목록. 오름차순 순서를 나타낼 경우 컬럼 이름 앞에 +를 표시하고 내림차순 순서를 나타낼 경우 -을 표시합니다.

주: <sup>3</sup>: Yes는 인덱스 유형 옵션을 지정하지 않고 SHOW DETAIL 절을 사용할 때에만 값이 리턴되는 것을 나타냅니다. 값이 널(NULL)이 될 수도 있습니다.

주: <sup>4</sup>: DESCRIBE *index-type* INDEXES FOR TABLE을 사용할 때 리턴되는 값을 나타냅니다. 예를 들어, TEXT SEARCH가 *index-type*으로 지정되는 경우 INDEX\_ID 값은 리턴되지 않습니다. RELATIONAL DATA 또는 XML DATA 중 하나가 지정되면 INDEX\_ID 값이 리턴됩니다.

주: <sup>5</sup>: DESCRIBE *index-type* INDEXES FOR TABLE SHOW DETAIL을 사용하면, 인덱스 유형이 나열될 때에만 값이 리턴됩니다. 예를 들어, XML DATA가 *index-type*으로 지정되는 경우 DATA\_TYPE 값이 리턴됩니다. TEXT SEARCH 또는 RELATIONAL DATA 중 하나가 *index-type*으로 지정되는 경우 DATA\_TYPE 값은 리턴되지 않습니다.

표 34. DESCRIBE DATA PARTITIONS FOR TABLE 명령에서 리턴된 결과 세트 1

컬럼 이름	데이터 유형	세부사항 <sup>2</sup>	설명
DATA_PARTITION_ID	INTEGER	없음	데이터 파티션 ID.
LOW_KEY_INCLUSIVE	CHAR(1)	없음	하위 키 값이 포함될 경우 'Y', 그렇지 않으면 'N'.
LOW_KEY_VALUE	VARCHAR(512)	없음	이 데이터 파티션의 하위 키 값.
HIGH_KEY_INCLUSIVE	CHAR(1)	없음	상위 키 값이 포함될 경우 'Y', 그렇지 않으면 'N'.
HIGH_KEY_VALUE	VARCHAR(512)	없음	이 데이터 파티션의 상위 키 값.

주: <sup>2</sup>: Yes는 SHOW DETAIL 절을 사용할 때만 널(NULL)이 아닌 값이 리턴되는 것을 나타냅니다.

표 35. SHOW DETAIL 절을 사용할 DESCRIBE DATA PARTITIONS FOR TABLE 명령에서 리턴된 결과 세트 2

컬럼 이름	데이터 유형	설명
DATA_PARTITION_ID	INTEGER	데이터 파티션 ID.
DATA_PARTITION_NAME	VARCHAR (128)	데이터 파티션 이름.
TBSPID	INTEGER	이 데이터 파티션이 저장된 테이블 스페이스 ID.
PARTITION_OBJECT_ID	INTEGER	이 데이터 파티션이 저장된 DMS 오브젝트 ID.
LONG_TBSPID	INTEGER	Long 데이터가 저장된 테이블 스페이스 ID.
INDEX_TBSPID	INTEGER	인덱스 데이터가 저장된 테이블 스페이스 ID.

표 35. SHOW DETAIL절을 사용할 DESCRIBE DATA PARTITIONS FOR TABLE 명령에서 리턴된 결과 세트 2 (계속)

컬럼 이름	데이터 유형	설명
ACCESSMODE	VARCHAR(20)	데이터 파티션의 액세스 가능성을 정의하며 다음 중 하나입니다. <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)	데이터 파티션 상태이며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• NEWLY_ATTACHED</li> <li>• NEWLY_DETACHED: MQT 유지보수가 필요합니다.</li> <li>• INDEX_CLEANUP_PENDING: 인덱스 정리를 위해서만 SYSDATAPARTITIONS의 튜플을 유지보수하는 접속 해제된 데이터 파티션. 접속 해제된 데이터 파티션에 관계된 모든 인덱스가 삭제되면 이 튜플이 제거됩니다.</li> </ul> 그렇지 않으면 컬럼이 공백입니다.

## ADMIN\_CMD 프로시저를 사용하는 DROP CONTACT 명령

로컬 시스템에 정의된 문의처 목록에서 문의처를 제거합니다. 문의처는 스케줄러 및 Health Monitor가 메시지를 보내는 사용자입니다. DAS(Database Administration Server) **contact\_host** 구성 매개변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

### 권한 부여

없음

### 필수 연결

데이터베이스. DAS가 실행하고 있어야 합니다.

### 명령 구문

```
►►—DROP CONTACT—name—◄◄
```

### 명령 매개변수

**CONTACT** *name*

로컬 시스템에서 삭제될 문의처 이름



예 :

서버 시스템의 문의처 목록에서 'testuser'라는 문의처를 삭제하십시오.

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

### 사용 시 참고사항

DAS를 작성하여 실행해야 합니다.

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

## ADMIN\_CMD 프로시저를 사용하는 DROP CONTACTGROUP 명령

로컬 시스템에 정의된 문의처 목록에서 문의처 그룹을 제거합니다. 문의처 그룹에는 스케줄러 및 Health Monitor가 메시지를 보내는 사용자 목록이 들어 있습니다. DAS(Database Administration Server) **contact\_host** 구성 매개변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

### 권한 부여

없음

### 필요한 연결

데이터베이스. DAS가 실행하고 있어야 합니다.

### 명령 구문

```
►►—DROP CONTACTGROUP—name—————◄◄
```

### 명령 매개변수

**CONTACTGROUP** *name*

로컬 시스템에서 삭제될 문의처 그룹의 이름

예 :

문의처 그룹 'gname1'을 삭제하십시오.

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

### 사용 시 참고사항

DAS를 작성하여 실행해야 합니다.

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

## ADMIN\_CMD 프로시저를 사용하는 EXPORT 명령

데이터베이스의 데이터를 여러 외부 파일 형식 중 하나로 익스포트합니다. 사용자는 SQL SELECT문을 제공하거나 유형이 지정된 테이블에 대한 계층 정보를 제공하여 익스포트할 데이터를 지정합니다. 데이터는 서버에만 익스포트됩니다.

76 페이지의 『익스포트 유틸리티의 파일 유형 수정자』로의 빠른 링크.

### 권한 부여

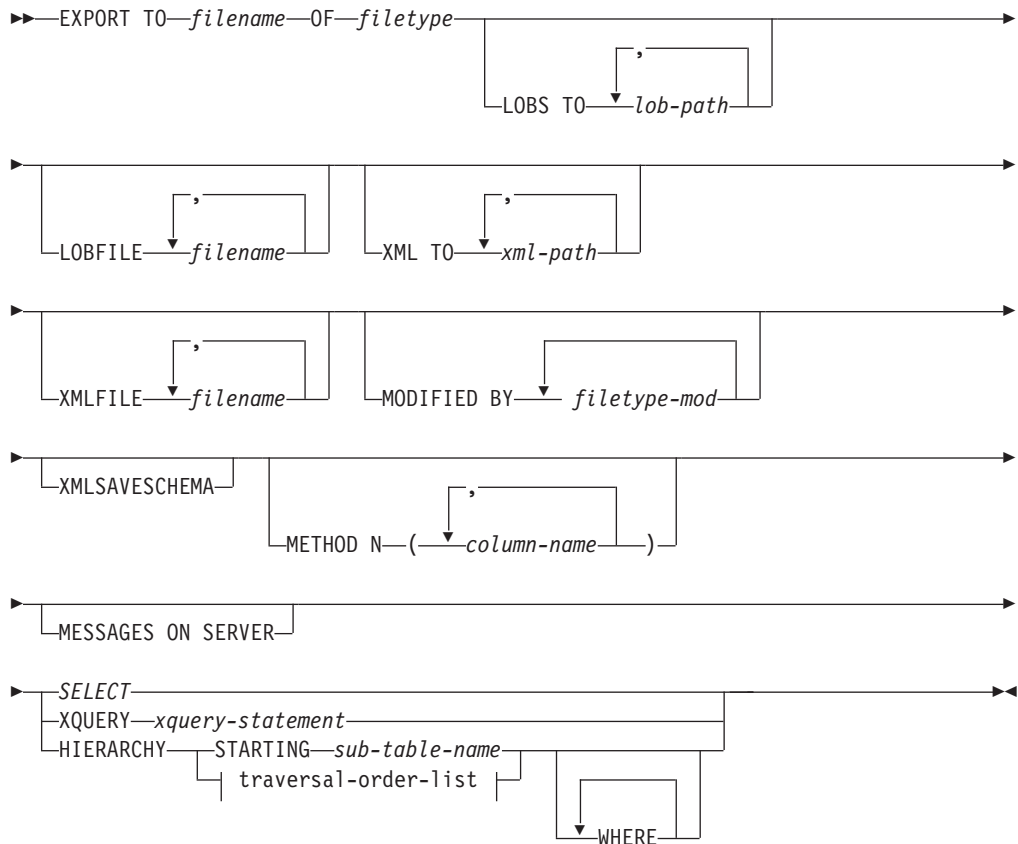
다음 중 하나가 필요합니다.

- *dataaccess* 권한
- 참여 중인 각 테이블 또는 뷰에 대한 CONTROL 또는 SELECT 특권

### 필수 연결

데이터베이스, Linux, UNIX 또는 Windows 클라이언트로부터 Linux, UNIX 또는 Windows 데이터베이스 서버에 대한 유틸리티 액세스는 DB2® Connect™ 게이트웨이 또는 루프 백 환경을 통하지 않고 엔진을 통해 직접 연결되어야 합니다.

### 명령 구문



## traversal-order-list:



## 명령 매개변수

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

지정된 이동 순서를 사용하여 하위 계층 구조를 익스포트합니다. 모든 서브테이블은 PRE-ORDER 형식으로 나열되어야 합니다. 첫 번째 서브테이블 이름을 SELECT문의 목표 테이블 이름으로 사용합니다.

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

디폴트 이동 순서(ASC, DEL 또는 WSF 파일의 경우 OUTER 순서 또는 PC/IXF 데이터 파일에 저장된 순서)를 사용하여 *sub-table-name*에서 시작하는 하위 계층 구조를 익스포트하십시오.

### **LOBFILE** *filename*

LOB 파일에 대한 하나 이상의 기본 파일을 지정합니다. 첫 번째 이름에 대한 이름 공간이 모두 사용되었으면 두 번째 이름을 사용하고 계속 이와 같이 합니다. 이는 내재적으로 LOBSINFILE 동작을 활성화합니다.

익스포트 조작 중에 LOB 파일을 작성할 경우, 파일 이름은 해당 목록의 현재 기본 이름을 현재 경로(*lob-path*의)에 추가한 후 시작할 3자리 시퀀스 번호 및 3자의 ID(*lob*)를 추가하여 구성됩니다. 예를 들어, 현재 LOB 경로가 /u/foo/lob/path/ 디렉토리이고 현재 LOB 파일 이름이 bar이면 작성되는 LOB 파일은 /u/foo/lob/path/bar.001.lob, /u/foo/lob/path/bar.002.lob 등입니다. 999가 사용되고 나면 LOB 파일 이름의 3자리 시퀀스 번호는 4자리가 되며, 9999가 사용되고 나면 4자리는 5자리가 되는 식으로 진행됩니다.

### **LOBS TO** *lob-path*

LOB 파일이 저장될 디렉토리에 대한 하나 이상의 경로를 지정하십시오. 경로는 서버의 코디네이터 파티션에 존재해야 하며 완전한 경로여야 합니다. LOB 경로 당 최소한 하나의 파일이 있으며 각 파일에는 최소한 하나의 LOB가 있습니다. 지정할 수 있는 최대 경로 수는 999입니다. 이는 내재적으로 LOBSINFILE 동작을 활성화합니다.

### **MESSAGES ON SERVER**

EXPORT 명령으로 서버에 작성한 메시지 파일을 저장하도록 지정합니다. 리턴되는 결과 세트에는 이 조작 중에 발생하는 경고 및 오류 메시지를 모두 검색하는 데 필요한 SQL문 MSG\_RETRIEVAL 컬럼과 메시지를 정리하는 데 필요한 SQL문인 MSG\_REMOVAL 컬럼이 필요합니다.

이 절을 지정하지 않으면 ADMIN\_CMD 프로시저가 호출자로 리턴될 때 메시지 파일이 삭제됩니다. 결과 세트의 MSG\_RETRIEVAL 및 MSG\_REMOVAL 컬럼에 널(NULL) 값이 포함됩니다.

이 절을 사용하든 또는 사용하지 않든 분리 사용자 ID는 데이터가 익스포트될 디렉토리 및 마찬가지로 DB2\_UTIL\_MSGPATH 레지스터 변수가 표시하는 디렉토리 아래에 파일을 작성할 수 있는 권한이 있어야 하는 점을 참고하십시오.

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

출력 파일에서 사용될 하나 이상의 컬럼 이름을 지정합니다. 이 매개변수를 지정하지 않으면 테이블에 있는 컬럼 이름이 사용됩니다. 이 매개변수는 WSF 및 IXF 파일에 대해서만 유효하며 계층 데이터를 익스포트할 때는 유효하지 않습니다.

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

파일 유형 수정자 옵션을 지정합니다. 76 페이지의 『익스포트 유틸리티의 파일 유형 수정자』의 내용을 참조하십시오.

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

출력 파일에 있는 데이터의 형식을 지정합니다.

- DEL(컬럼 식별자가 있는 ASCII 형식) - 여러 데이터베이스 관리 프로그램 및 파일 관리자 프로그램에서 사용됩니다.
- WSF(작업시트 형식) - 다음과 같은 프로그램에서 사용됩니다.
  - Lotus® 1-2-3®
  - Lotus Symphony

BIGINT 또는 DECIMAL 데이터를 익스포트할 때 DOUBLE 유형의 범위에 해당하는 값만 정확히 익스포트할 수 있습니다. 이 범위에 해당하지 않는 값은 익스포트되기는 하지만 이들 값을 다시 임포트하거나 로드할 때 운영 체제에 따라 올바르게 나타낼 수 없습니다.

주: WSF 파일 형식에 대한 지원은 사용되지 않으므로 추후 릴리스에서 제거될 수 있습니다. 지원이 제거되기 전에 WSF 파일 대신 지원되는 파일 형식을 사용하여 시작하는 것이 좋습니다.

- IXF(Integration Exchange Format, PC 버전)는 독점 2진 형식입니다.

#### **SELECT**

익스포트될 데이터를 리턴하는 SELECT 또는 XQUERY문을 지정합니다. 명령문이 오류를 일으킬 경우 메시지 파일(또는 표준 출력)에 메시지가 쓰여집니다. 오류 코드가 SQL0012W, SQL0347W, SQL0360W, SQL0437W 또는 SQL1824W 중 하나이면 익스포트 작업을 계속 수행하고 그렇지 않으면 중지합니다.

### **TO filename**

서버에 데이터가 익스포트될 파일 이름을 지정합니다. 이는 완전한 경로여야 하며 서버 코디네이터 파티션에 존재해야 합니다.

이미 존재하는 파일 이름을 지정할 경우 익스포트 유틸리티는 파일 내용을 겹쳐쓰고 정보를 추가하지 않습니다.

### **XMLFILE filename**

XML 파일에 대한 하나 이상의 기본 파일 이름을 지정합니다. 첫 번째 이름에 대한 이름 공간이 모두 사용되었으면 두 번째 이름을 사용하고 계속 이와 같이 합니다.

익스포트 조작 중 XML 파일을 작성할 때 파일 이름은 이 목록의 현재 기본 이름을 현재 경로(*xml-path*로부터)에 추가하고 3자리 숫자 시퀀스 번호를 추가한 후 3자 ID *xml*을 추가합니다. 예를 들어, 현재 XML 경로가 */u/foo/xml/path/* 디렉토리이고 현재 XML 파일 이름이 *bar*이면 작성되는 XML 파일은 */u/foo/xml/path/bar.001.xml*, */u/foo/xml/path/bar.002.xml* 등입니다.

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

XML 파일이 저장될 디렉토리에 대한 하나 이상의 경로를 지정하십시오. XML 경로당 최소한 하나의 파일이 있으며 각 파일에는 최소한 하나의 XDM(XQuery Data Model) 인스턴스가 있습니다. 두 개 이상의 경로를 지정하면 XDM 인스턴스는 해당 경로에 균일하게 분산됩니다.

### **XMLSAVESCHEMA**

모든 XML 컬럼에 대해 XML 스키마 정보가 저장되어야 함을 지정합니다. 각 익스포트된 XM 문서가 삽입될 때 XML 스키마에 대해 유효성이 확인되었으면 해당 스키마의 완전한 SQL ID는 해당 XDS(XML Data Specifier) 내에 (SCH) 속성으로 저장됩니다. 익스포트된 문서가 XML 스키마에 대해 유효성이 확인되지 않았거나 데이터베이스에 스키마 오브젝트가 더 이상 존재하지 않으면 해당 XDS에 SCH 속성이 포함되지 않습니다.

SQL ID의 스키마 및 이름 부분은 XML 스키마에 해당하는 SYSCAT.XSROBJECTS 카탈로그 테이블 행에 "OBJECTSCHEMA" 및 "OBJECTNAME" 값으로 저장됩니다.

XMLSAVESCHEMA 옵션은 올바른 형식의 XML 문서를 생성하지 않는 XQuery 시퀀스와는 호환 가능하지 않습니다.

### **예 :**

다음 예는 SAMPLE 데이터베이스에 있는 STAFF 테이블의 정보를 myfile.ixf 파일로 익스포트하는 방법을 보여 줍니다. 해당 출력은 IXF 형식으로 생성됩니다. 명령을 발행하기 전에 먼저 SAMPLE 데이터베이스에 연결해야 합니다.

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

## 사용 시 참고사항

- EXPORT 명령에 사용되는 경로는 서버에서 유효한 완전한 경로여야 합니다.
- 테이블에 LOB 컬럼이 들어 있으면 LOBS TO 및 LOBFILE 절을 사용하여 최소 한 하나의 완전한 LOB 경로와 LOB 이름을 지정해야 합니다.
- 익스포트 유틸리티는 유형 2 연결의 경우 프로시저가 이유 코드 2와 함께 SQL30090N 을 리턴하게 하는 조건의 시작부분에서 COMMIT문을 실행합니다.
- UCS-2 데이터베이스에서 컬럼 식별자가 있는 ASCII(DEL) 파일로 익스포트할 때 모든 문자 데이터는 프로시저가 실행되는 곳에서 유효한 코드 페이지로 변환됩니다. 문자열과 그래픽 문자열 데이터는 서버와 동일한 SBCS 또는 MBCS 코드 페이지로 변환됩니다.
- 익스포트 조장을 시작하기 전에 반드시 모든 테이블 조장을 완료하고 모든 잠금을 릴리스하십시오. 이는 WITH HOLD로 열려진 모든 커서를 닫고 COMMIT를 발행 하거나 ROLLBACK을 발행하여 수행할 수 있습니다.
- SELECT문에서 테이블 별명을 사용할 수 있습니다.
- 메시지 파일에 넣어진 메시지는 메시지 검색 서비스에서 리턴된 정보를 포함합니다. 각 메시지는 새 행에서 시작합니다.
- 익스포트 유틸리티는 DEL 형식 파일으로 익스포트하기 위해 길이가 254를 초과하는 문자 컬럼을 선택할 때마다 경고 메시지를 생성합니다.
- 데이터베이스 간에 데이터를 이동하려면 PC/IXF 임포트를 사용해야 합니다. 행 구분자가 있는 문자 데이터를 컬럼 식별자가 있는 ASCII(DEL) 파일로 익스포트하며 텍스트 전송 프로그램으로 이를 처리할 경우 행 구분자가 있는 필드는 줄어들거나 확장됩니다.
- 동일한 클라이언트에서 소스 또는 목표 데이터베이스에 모두 액세스할 수 있는 경우 파일 복사 단계는 필요하지 않습니다.
- DB2 Connect는 OS/390®용 DB2, VM 및 VSE용 DB2, OS/400®용 DB2와 같은 DRDA® 서버에서 테이블을 익스포트하는 데 사용할 수 있습니다. PC/IXF 익스포트만 지원됩니다.
- IXF 형식으로 익스포트할 때, ID가 IXF 형식에서 지원하는 최대 크기를 초과하는 경우, 익스포트에 성공하지만 결과 데이터 파일은 CREATE 모드를 사용하는 추후 임포트 조작에서는 사용될 수 없습니다. SQL27984W가 리턴됩니다.
- Windows의 디스켓으로 익스포트하고 단일 디스켓의 용량보다 큰 데이터가 있는 테이블을 익스포트할 때, 시스템은 다른 디스켓을 위해 프롬프트를 표시하며, 다중 파트 PC/IXF 파일(다중 볼륨 PC/IXF 파일 또는 논리 분할 PC/IXF 파일이라고도 함) 이 생성되어 별도의 디스켓에 저장됩니다. 마지막의 경우를 제외하고 각 파일에는, 파일이 논리적으로 분할되었음을 표시하고 다음 파일을 찾을 곳을 나타내기 위해 쓰여

진 DB2 CONTINUATION RECORD(또는 짧게 "AC" 레코드)가 있습니다. AIX 시스템으로 해당 파일을 전송하여, импорт 및 로드 유틸리티에서 읽을 수 있습니다. 익스포트 유틸리티는 AIX 시스템에서 호출될 때 다중 파트 PC/IXF 파일을 작성하지 않습니다. 자세한 사용을 위해서는 IMPORT 명령 또는 LOAD 명령을 참조하십시오.

- 익스포트 유틸리티는 제공된 SELECT문이 SELECT \* FROM tablename 형식인 경우 해당 테이블의 NOT NULL WITH DEFAULT 속성을 IXF 파일에 저장합니다.
- 유형이 지정된 테이블을 익스포트할 때 subselect문은 단지 목표 테이블 이름과 WHERE 절을 지정하여 표현할 수 있습니다. 계층 구조를 익스포트할 때는 Fullselect 및 *select-statement*를 지정할 수 없습니다.
- IXF 이외의 파일 형식에서는 이동 순서 목록을 지정할 것을 권장합니다. 이는 이 순서가 DB2에게 계층 구조를 이동하는 방법과 어떤 서브테이블을 익스포트할지를 알려주기 때문입니다. 이 목록을 지정하지 않으면 해당 계층 구조에 있는 모든 테이블이 익스포트되고 디폴트 순서는 OUTER 순서가 됩니다. 대안은 OUTER 기능이 제공하는 순서인 디폴트 순서를 사용하는 방법입니다.
- импорт 조작 동안 동일한 이동 순서를 사용하십시오. 로드 유틸리티는 계층 구조 또는 하위 계층 구조를 지원하지 않습니다.
- 행이 보호 설정된 테이블에서 데이터를 익스포트할 때 세션 권한 부여 ID가 보유하는 LBAC 증명서가 익스포트되는 행을 제한할 수 있습니다. 세션 권한 부여 ID가 읽기 권한을 가지고 있지 않은 행은 익스포트되지 않습니다. 오류나 경고는 표시되지 않습니다.
- 세션 권한 부여 ID가 보유하는 LBAC 증명서가 익스포트에 포함된 하나 이상의 보호 설정된 컬럼에서 읽기를 허용하지 않을 경우 익스포트를 실패하고 오류(SQLSTATE 42512)가 리턴됩니다.
- 익스포트 패키지는 DATETIME ISO 형식을 사용하여 바운드로 문자열 표현식으로 캐스트할 때 모든 날짜/시간/시간소인 값은 ISO 형식으로 변환됩니다. CLP 패키지는 DATETIME LOC 형식(로케일 특정 형식)을 사용하여 바운드로 CLP DATETIME 형식이 ISO 형식과 다른 경우 CLP 및 익스포트 간에 동작이 불일치할 수 있습니다. 예를 들어, 다음 SELECT문에서는 예상대로 결과가 리턴됩니다.

```
db2 select col2 from tab1 where char(col2)='05/10/2005';
COL2
-----
05/10/2005
05/10/2005
05/10/2005
3개의 레코드가 선택되었습니다.
```

그러나 동일한 Select절을 사용한 Export 명령에서는 예상대로 결과가 리턴되지 않습니다.



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

이제 LOCALE 날짜 형식을 ISO 형식으로 바꾸면 예상대로 결과가 리턴됩니다.

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

## 결과 세트 정보

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다. 실행이 완료되면 명령은 다음과 같이 결과 세트로 추가 정보를 리턴합니다.

표 36. EXPORT 명령에서 리턴된 결과 세트

컬럼 이름	데이터 유형	설명
ROWS_EXPORTED	BIGINT	익스포트된 행의 총 수.
MSG_RETRIEVAL	VARCHAR(512)	이 유틸리티에서 작성된 메시지를 검색하는 데 사용되는 SQL문. 예를 들면, 다음과 같습니다.  SELECT SQLCODE, MSG FROM TABLE (SYSPROC.ADMIN_GET_MSGS ('3203498_txu')) AS MSG
MSG_REMOVAL	VARCHAR(512)	이 유틸리티에서 작성된 메시지를 정리하는 데 사용되는 SQL문. 예를 들면, 다음과 같습니다.  CALL SYSPROC.ADMIN_REMOVE_MSGS ('3203498_txu')

## 익스포트 유틸리티의 파일 유형 수정자

표 37. 익스포트 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식

수정자	설명
lobsinfile	<p><i>lob-path</i>는 LOB 데이터를 포함하는 파일에 대한 경로를 지정합니다.</p> <p>각 경로는 LLS(Lob Location Specifier)에 의해 포인팅되는 최소한 하나의 LOB를 포함하는 최소한 하나의 파일을 데이터 파일에 포함합니다. LLS는 LOB 파일 경로에 저장된 파일에서 LOB 위치의 문자열 표시입니다. LLS의 형식은 <i>filename.ext.nnn.mmm</i>이며, 여기서 <i>filename.ext</i>는 LOB를 포함하는 파일의 이름이며, <i>nnn</i>은 파일 내에서 LOB의 오프셋을 바이트로 나타낸 것이며, <i>mmm</i>은 LOB의 길이를 바이트로 나타낸 것입니다. 예를 들어, 문자열 db2exp.001.123.456/가 데이터 파일에 저장되는 경우, LOB는 db2exp.001 파일에서 오프셋 123에 위치하며 456바이트 길이입니다.</p> <p>EXPORT 사용 시 『lobsinfile』 수정자를 지정하는 경우, LOB 데이터는 LOBS TO 절에서 지정하는 위치에 있습니다. 그렇지 않으면 LOB 데이터는 데이터 파일 디렉토리로 보냅니다. LOBS TO 절은 LOB 파일이 저장될 디렉토리에 대한 하나 이상의 경로를 지정합니다. LOB 경로 당 최소한 하나의 파일이 있으며 각 파일에는 최소한 하나의 LOB가 있습니다. LOBS TO 또는 LOBFILE 옵션은 내재적으로 LOBSINFILE 동작을 활성화합니다.</p> <p>널(NULL) LOB를 표시하려면, 크기를 -1로 입력하십시오. 크기가 0으로 지정되면, 길이가 0인 LOB로 처리됩니다. 길이가 -1인 널(NULL) LOBS의 경우, 오프셋 및 파일 이름은 무시됩니다. 예를 들어, 널(NULL) LOB의 LLS는 db2exp.001.7.-1/입니다.</p>



표 37. 익스포트 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
xmlinsefiles	각 XDM(XQuery Data Model) 인스턴스는 별도의 파일에 쓰여집니다. 디폴트로 복수의 값은 동일한 파일에서 함께 병합됩니다.
lobsinsefiles	각 LOB 값은 별도의 파일에 쓰여집니다. 디폴트로 복수의 값은 동일한 파일에서 함께 병합됩니다.
xmlnodeclaration	XDM 인스턴스는 XML 선언 태그 없이 쓰여집니다. 디폴트로 XDM 인스턴스는 처음에 인코딩 속성을 포함하는 XML 선언 태그를 가지고 익스포트됩니다.
xmlchar	XDM 인스턴스는 문자 코드 페이지에 쓰여집니다. 문자 코드 페이지는 codepage 파일 유형 수정자에서 지정된 값이거나, 지정되지 않은 경우 응용프로그램 코드 페이지입니다. 디폴트로, XDM 인스턴스는 유니코드로 쓰여집니다.
xmlgraphic	xmlgraphic 수정자가 EXPORT 명령에서 지정된 경우, 익스포트된 XML 문서는 응용프로그램 코드 페이지나 codepage 파일 유형 수정자와 관계없이 UTF-16 코드 페이지로 인코딩됩니다.

표 38. 익스포트 유틸리티의 유효한 파일 유형 수정자: DEL(컬럼 식별자가 있는 ASCII) 파일 형식

수정자	설명
chardelx	x는 단일 문자열 분리문자입니다. 디폴트값은 큰따옴표(")입니다. 문자열을 묶기 위해 큰따옴표 대신 지정된 문자가 사용됩니다. <sup>2</sup> 명시적으로 큰따옴표를 문자열 분리문자로 지정하려는 경우 다음과 같이 지정해야 합니다.  modified by chardel"  다음과 같이 작은따옴표(')를 문자열 분리문자로 지정할 수도 있습니다.  modified by chardel''
codepage=x	x는 ASCII 문자열입니다. 해당 값은 출력 데이터 세트에서 데이터의 코드 페이지로서 해석됩니다. 익스포트 조작 중 응용프로그램 코드 페이지로부터 이 코드 페이지로 문자 데이터를 변환합니다.  순수 DBCS(그래픽), 혼합 DBCS 및 EUC의 경우, 분리문자는 x00 - x3F의 범위로 제한됩니다. codepage 수정자는 lobsinfile 수정자와 함께 사용될 수 없습니다.
coldelx	x는 단일 문자 컬럼 분리문자입니다. 디폴트값은 쉼표(,)입니다. 지정된 문자가 쉼표 대신 사용되어 컬럼 끝을 나타냅니다. <sup>2</sup>  다음 예에서, coldel;은 익스포트 유틸리티가 익스포트된 데이터의 컬럼 분리문자로 세미콜론 문자(; )를 사용하게 합니다.  db2 "export to temp of del modified by coldel; select * from staff where dept = 20"
decplusblank	플러스 부호 문자. 플러스 부호(+) 대신 공백을 양의 10진수 값 앞에 붙입니다. 디폴트 조치는 플러스 부호를 양의 10진수 값 앞에 붙이는 것입니다.
decptx	x는 소수점 문자로 마침표를 대신하는 단일 문자입니다. 디폴트값은 마침표(.)입니다. 마침표 대신 지정된 문자가 소수점 문자로 사용됩니다. <sup>2</sup>
nochardel	컬럼 데이터는 문자 분리문자 안에 두지 않습니다. DB2를 사용하여 데이터를 임포트하거나 로드하려는 경우 이 옵션을 지정하지 말아야 합니다. 이 옵션은 문자 분리문자가 없는 벤더 데이터 파일을 지원하기 위해 제공됩니다. 부적절한 사용은 데이터 유실이나 손상을 초래할 수 있습니다.  chardelx 또는 nodoubledel에는 이 옵션을 지정할 수 없습니다. 이들은 상호 배타적인 옵션입니다.



표 38. 익스포트 유틸리티의 유효한 파일 유형 수정자: DEL(컬럼 식별자가 있는 ASCII) 파일 형식 (계속)

수정자	설명
timestampformat="x"	<p>x는 소스 파일에서 시간소인의 형식입니다.<sup>4</sup> 유효한 시간소인 요소는 다음과 같습니다.</p> <p>YYYY - Year(0000 - 9999 사이의 4자리 숫자)  M - Month(1 - 12 사이의 1 또는 2자리 숫자)  MM - Month(01 - 12 사이의 2자리 숫자,  M 및 MM은 상호 배타적)  MMM - Month(월 이름으로 사용되는 3자리 문자로 대소문자를 구분하지 않으며,  M 및 MM은 상호 배타적)  D - Day(1 - 31 사이의 1 또는 2자리 숫자)  DD - Day(1 - 31 사이의 2자리 숫자, D와는 상호 배타적)  DDD - Day of the year(001 - 366 사이의 3자리 숫자,  다른 일 또는 월 요소와 상호 배타적)  H - Hour(12시간 시스템의 경우 0 - 12 범위의 1 또는 2자리 숫자,  24시간 시스템의 경우 0 - 24)  HH - Hour(12시간 시스템의 경우 0 - 12 범위의 2자리 숫자,  24시간 시스템의 경우 0 - 24,  H와는 상호 배타적)  M - Minute(0 - 59 사이의 1 또는 2자리 숫자)  MM - Minute(0 - 59 사이의 2자리 숫자,  M, 분과 상호 배타적)  S - Second(0 - 59 사이의 1 또는 2자리 숫자)  SS - Second(0 - 59 사이의 2자리 숫자,  S와 상호 배타적)  SSSSS - Second of the day after midnight(  00000 - 86399 사이의 5자리 숫자, 다른 시간 요소와  상호 배타적)  U(1 - 12회)  - Fractional seconds(U 어커런스 수는 각 자리가 0 - 9 사이의  자리 수를 나타냄)  TT - Meridian indicator(AM 또는 PM)</p> <p>다음은 시간소인 형식의 예입니다.</p> <p>"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>MMM 요소는 다음 값을 생성합니다. 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',  'Sep', 'Oct', 'Nov' 및 'Dec'. 'Jan'은 1월이며, 'Dec'는 12월과 같습니다.</p> <p>다음 예는 'schedule'이라는 테이블에서 사용자 정의 시간소인을 포함하는 데이터를 익스포트  하는 방법을 설명합니다.</p> <pre>db2 export to delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" select * from schedule</pre>

표 39. 익스포트 유틸리티의 유효한 파일 유형 수정자: IXF 파일 형식

수정자	설명
codepage=x	<p>x는 ASCII 문자열입니다. 해당 값은 출력 데이터 세트에서 데이터의 코드 페이지로서 해석됩니다. 익스포트 조작 중 이 코드 페이지로부터 응용프로그램 코드 페이지로 문자 데이터를 변환합니다.</p> <p>순수 DBCS(그래픽), 혼합 DBCS 및 EUC의 경우, 분리문자는 x00 - x3F의 범위로 제한됩니다. codepage 수정자는 lobsinfile 수정자와 함께 사용될 수 없습니다.</p>

표 40. 익스포트 유틸리티의 유효한 파일 유형 수정자: WSF 파일 형식<sup>6</sup>

수정자	설명
1	Lotus 1-2-3 릴리스 1 또는 Lotus 1-2-3 릴리스 1a와 호환 가능한 WSF 파일을 작성합니다. 5 이것이 디폴트입니다.
2	Lotus Symphony 릴리스 1.0과 호환 가능한 WSF 파일을 작성합니다. <sup>5</sup>
3	Lotus 1-2-3 버전 2 또는 Lotus Symphony 릴리스 1.1과 호환 가능한 WSF 파일을 작성합니다. <sup>5</sup>
4	DBCS 문자를 포함하는 WSF 파일을 작성합니다.

주:

1. MODIFIED BY 옵션과 함께 지원되지 않는 파일 유형을 사용하려고 시도하는 경우 익스포트 유틸리티는 경고를 발행하지 않습니다. 이런 경우, 익스포트 조작에 실패하며 오류 코드가 리턴됩니다.

2. 데이터 이동을 위한 분리문자 고려사항은 분리문자를 겹쳐쓰기할 때 사용할 수 있는 문자에 적용하는 제한사항을 나열합니다.

3. 정상적인 익스포트 유틸리티 쓰기

- YYYYMMDD 형식의 날짜 데이터
- "YYYY-MM-DD" 형식의 문자(날짜) 데이터
- "HH.MM.SS" 형식의 시간 데이터
- "YYYY-MM-DD-HH .MM.SS.aaaaaa" 형식의 시간 소인 데이터

익스포트 조작의 SELECT문에서 지정된 날짜 및 시간 컬럼에 포함된 데이터도 다음 형식입니다.

4. 시간소인 형식의 경우, 문자 M을 사용하는 month 및 minute 디스크립터 간의 모호함을 피하려면 주의해야 합니다. month 필드는 다른 날짜 필드와 인접해야 합니다. minute 필드는 다른 시간 필드와 인접해야 합니다. 다음은 앰비규어스 시간 소인 형식입니다.

- "M"(month 또는 minute일 수 있음)
- "M:M"(month 및 minute 구분 가능?)
- "M:YYYY:M"(둘 다 month로 해석됨)
- "S:M:YYYY"(둘 다 시간 값 및 날짜 값에 인접)

앰비규어스 경우, 유틸리티는 오류 메시지를 발행하며, 조작에 실패합니다.

다음은 명확한 시간소인 형식입니다.

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

5. 또한 이러한 파일은 Lotus 1-2-3의 경우 L, Symphony의 경우 S를 filetype-mod 매개변수 문자열에 지정하여 특정 제품으로 방향지정할 수 있습니다. 하나의 값 또는 제품 지정자만 지정할 수 있습니다. WSF 파일 형식에 대한 지원은 사용되지

않으므로 추후 릴리스에서 제거될 수 있습니다. 지원이 제거되기 전에 WSF 파일 대신 지원되는 파일 형식을 사용하여 시작하는 것이 좋습니다.

6. WSF 파일 형식은 XML 컬럼에서 지원되지 않습니다. 이 파일 형식에 대한 지원은 사용되지 않으므로 추후 릴리스에서 제거될 수 있습니다. 지원이 제거되기 전에 WSF 파일 대신 지원되는 파일 형식을 사용하여 시작하는 것이 좋습니다.
7. XMLFILE 또는 XML TO 절 중 어느 하나도 지정되지 않아도 모든 XDM 인스턴스는 기본 데이터 파일에서 분리된 XML 파일에 쓰여집니다. 디폴트로, XML 파일은 익스포트된 데이터 파일의 경로에 쓰여집니다. 디폴트로 XML 파일의 기본 이름은 익스포트된 데이터 파일의 이름으로 확장자가 ".xml"입니다.
8. XMLNODEDECLARATION 파일 유형 수정자가 지정된 경우를 제외하고, 모든 XDM 인스턴스는 처음에 인코딩 속성을 포함하는 XML 선언을 이용하여 쓰여집니다.
9. XMLCHAR 또는 XMLGRAPHIC 파일 유형 수정자가 지정된 경우를 제외하고, 디폴트로 모든 XDM 인스턴스는 유니코드로 쓰여집니다.
10. XML 데이터 및 LOB 데이터의 디폴트 경로는 기본 데이터 파일 경로입니다. 디폴트 XML 파일 기본 이름이 주 데이터 파일입니다. 디폴트 LOB 파일 기본 이름이 주 데이터 파일입니다. 예를 들어 주 데이터 파일이

/mypath/myfile.del

인 경우 XML 데이터 및 LOB 데이터의 디폴트 경로는

/mypath"

, 디폴트 XML 파일 기본 이름은

myfile.del

, 디폴트 LOB 파일 기본 이름은

myfile.del

입니다. LOB 파일을 생성하려면 LOBSINFILE 파일 유형 수정자를 지정해야 합니다.

11. 익스포트 유틸리티는 각 LOB 파일이나 XML 파일에 숫자 ID를 추가합니다. ID는 0으로 패드된 3자리 시퀀스 값인

.001

로 시작합니다. 999번째 LOG 파일 또는 XML 파일 다음에 ID는 영(0)으로 더 이상 패드되지 않습니다. (예를 들어 10000번째 파일 또는 XML 파일에

.1000

확장자가 없습니다.) 다음 숫자 ID는 데이터 유형을 나타내는 세 개의 문자 유형 ID로

.lob

또는

.xml

을 나타냅니다. 예를 들어 생성된 LOB 파일에

myfile.del.001.lob

형식의 이름이 있고 생성된 XML 파일에

myfile.del.001.xml

12. XQuery를 지정하여 잘 양식화된 문서가 아닌 익스포트 유틸리티 익스포트 XDM 인스턴스를 갖는 것이 가능합니다. 그러나 XML 컬럼은 완전한 문서만을 포함하므로, 해당 익스포트된 문서를 직접 XML 컬럼에 임포트하거나 로드할 수 없습니다.

## ADMIN\_CMD 프로시저를 사용하는 FORCE APPLICATION 명령

강제로 로컬 또는 리모트 사용자 또는 응용프로그램이 시스템을 해제하도록 하여 서버에서 유지보수를 할 수 있도록 합니다.

**경고:** 인터럽트할 수 없는 조작(예를 들어, RESTORE DATABASE)이 강제 수행된 경우 데이터베이스를 사용할 수 있게 되기 전에 먼저 이 조작을 재실행해야 합니다.

### 범위

이 명령은 \$HOME/sql1lib/db2nodes.cfg 파일에 나열된 모든 데이터베이스 파티션에 영향을 줍니다.

파티션된 데이터베이스 환경에서는 이 명령을 강제되고 있는 응용프로그램의 코디네이터 데이터베이스 파티션에서 발행하지 않아도 됩니다. 파티션된 데이터베이스 환경에 있는 모든 노드(데이터베이스 파티션 서버)에서 발행할 수 있습니다.

### 권한 부여

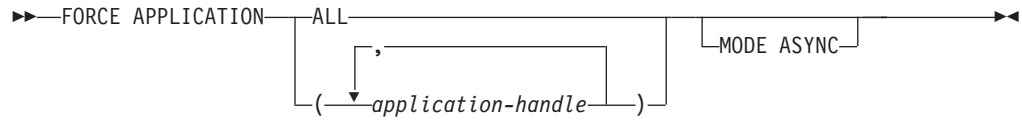
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

데이터베이스

## 명령 구문



## 명령 매개변수

### FORCE APPLICATION

**ALL** 모든 응용프로그램이 데이터베이스에서 연결이 끊어집니다. 이는 ADMIN\_CMD 프로시저가 실행 중인 연결을 닫을 수 있으므로 강제 조작이 완료된 후 ADMIN\_CMD 프로시저에 대해 SQL1224N 오류가 리턴되게 합니다.

#### *application-handle*

종료될 에이전트를 지정합니다. LIST APPLICATIONS 명령을 사용하여 값을 나열하십시오.

### MODE ASYNC

이 명령은 모든 지정된 사용자가 종료될 때까지 대기하지 않고 바로 리턴합니다. 함수가 발행되거나 오류(예: 유효하지 않은 구문)가 발견되면 즉시 리턴합니다.

현재 이 모드만 지원됩니다.

## 예:

다음 예는 *application-handle* 값이 41408 및 55458인 두 사용자가 강제로 데이터베이스에서 연결이 끊어지도록 합니다.

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

## 사용 시 참고사항

데이터베이스 관리 프로그램은 활동 상태로 남아 db2start를 수행할 필요가 없이 후속 데이터베이스 관리 프로그램 조작을 처리할 수 있도록 합니다.

데이터베이스 무결성을 보존하기 위해 유틸 중이거나 인터럽트할 수 있는 데이터베이스 조작을 실행하는 사용자만 종료할 수 있습니다.

다음과 같은 유형의 사용자 및 응용프로그램은 강제할 수 없습니다.

- 데이터베이스를 작성 중인 사용자
- 시스템 응용프로그램

이러한 유형의 사용자 및 응용프로그램을 성공적으로 강제하려면, 데이터베이스를 비활성화하거나 인스턴스를 재시작해야 합니다.

FORCE APPLICATION을 발행한 후 데이터베이스는 연결 요청을 계속 승인합니다. 모든 사용자를 완전히 강제로 해제하려면 추가 강제가 필요할 수 있습니다.

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

## ADMIN\_CMD 프로시저를 사용한 GET STMM TUNING DBPARTITIONNUM 명령

사용자 선호 자체 성능 조정 메모리 관리자(STMM) 조정 데이터베이스 파티션 번호 및 현재 STMM 조정 데이터베이스 파티션 번호를 보고하기 위해 키탈로그 테이블을 읽는데 사용됩니다.

### 권한 부여

명령문의 권한 부여 ID에 의해 보유된 특권에는 다음과 같은 권한 또는 특권 중 하나 이상을 포함해야 합니다

- DBADM
- SECADM
- SQLADM
- ACCESSCTRL
- DATAACCESS
- SYSIBM.SYSTUNINGINFO의 SELECT

### 필수 연결

데이터베이스

### 명령 구문

```
▶▶ GET STMM TUNING DBPARTITIONNUM ◀◀
```

예 :

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

이 쿼리의 출력 예는 다음과 같습니다.

결과 세트 1

-----

```
USER_PREFERRED_NUMBER CURRENT_NUMBER
```

-----

2

2

1개의 레코드가 선택되었습니다.

리턴 상태 = 0



## 사용 시 참고사항

사용자 선호 자체 성능 조정 메모리 관리자(STMM) 조정 데이터베이스 파티션 번호 (USER\_PREFERRED\_NUMBER)는 사용자에게 의해 설정되며 사용자가 메모리 조정 프로그램을 실행하려 하는 데이터베이스 파티션을 지정합니다. 데이터베이스가 실행되는 동안 조정 파티션은 한시간에 여러 번 비동기적으로 갱신됩니다. 따라서 사용자가 선호하는 STMM 파티션 번호를 갱신한 후 리턴된 CURRENT\_NUMBER와 USER\_PREFERRED\_NUMBER는 동기화된 상태가 아닐 수 있습니다. 이를 해결하려면 CURRENT\_NUMBER가 비동기적으로 갱신되기를 기다리거나 데이터베이스를 중지한 후 다시 시작하여 CURRENT\_NUMBER를 강제로 갱신하십시오.

## 결과 세트 정보

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다. 실행이 완료되면 명령은 다음 결과 세트로 추가 정보를 리턴합니다.

표 41. GET STMM TUNING DBPARTITIONNUM 명령에서 리턴된 결과 세트

컬럼 이름	데이터 유형	설명
USER_PREFERRED_NUMBER	INTEGER	사용자 선호 자체 성능 조정 메모리 관리자(STMM) 조정 데이터베이스 파티션 번호. -1값은 디폴트 데이터베이스 파티션을 사용했음을 나타냅니다.
CURRENT_NUMBER	INTEGER	현재 STMM 조정 데이터베이스 파티션 번호. -1값은 디폴트 데이터베이스 파티션을 사용했음을 나타냅니다.

## ADMIN\_CMD 프로시저를 사용하는 IMPORT 명령

지원되는 파일 형식을 갖는 외부 파일의 데이터를 테이블, 계층 구조, 뷰 또는 별칭으로 데이터를 삽입합니다. LOAD가 더 빠른 방법이기도 하지만 로드 유틸리티는 계층 구조 레벨에서 데이터 로드를 지원하지 않습니다.

103 페이지의 『임포트 유틸리티의 파일 유형 수정자』로의 빠른 링크.

### 권한 부여

- **INSERT** 옵션을 사용하여 IMPORT하려면 다음 중 하나가 필요합니다.
  - *dataaccess* 권한
  - 각 참여 중인 테이블, 뷰 또는 별칭에 대한 CONTROL 특권
  - 각 참여 중인 테이블 또는 뷰에 대한 INSERT 및 SELECT 특권
- **INSERT\_UPDATE** 옵션을 사용하여 기존 테이블로 IMPORT하려면 다음 중 하나가 필요합니다.
  - *dataaccess* 권한

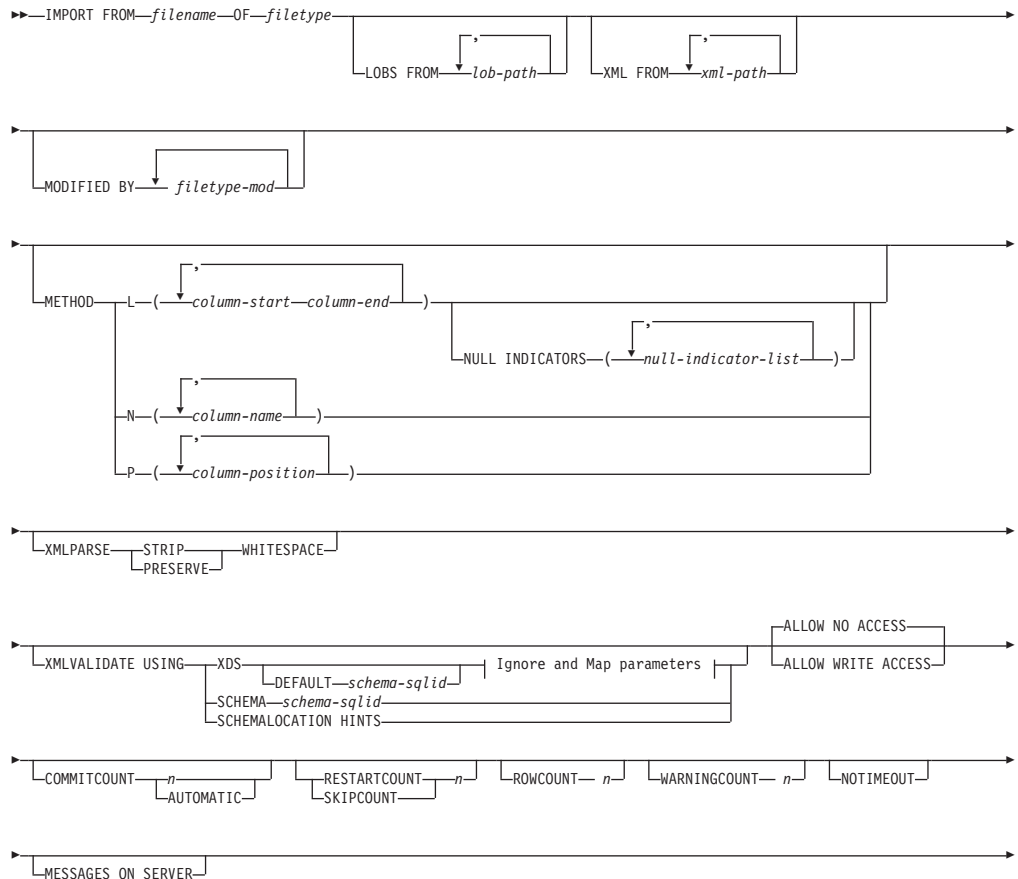
- 각 참여 중인 테이블, 뷰 또는 별칭에 대한 CONTROL 특권
- 각 참여 중인 테이블 또는 뷰에 대한 INSERT, SELECT, UPDATE 및 DELETE 특권
- **REPLACE** 또는 **REPLACE\_CREATE** 옵션을 사용하여 기존 테이블로 IMPORT하려면 다음 중 하나가 필요합니다.
  - *dataaccess* 권한
  - 테이블 또는 뷰에 대한 CONTROL 특권
  - 테이블 또는 뷰에 대한 INSERT, SELECT 및 DELETE 특권
- **CREATE** 또는 **REPLACE\_CREATE** 옵션을 사용하여 새 테이블로 IMPORT하려면 다음 중 하나가 필요합니다.
  - *dbadm* 권한
  - 데이터베이스에 대한 CREATETAB 권한 및 테이블 스페이스에 대한 USE 특권은 다음 경우 중 하나입니다.
    - 테이블의 내재적 또는 명시적 스키마 이름이 존재하지 않을 경우, 데이터베이스에 대한 IMPLICIT\_SCHEMA 권한
    - 테이블의 스키마 이름이 기존의 스키마를 참조할 경우, 스키마에 대한 CREATEIN 특권
- **CREATE** 또는 **REPLACE\_CREATE** 옵션을 사용하여 존재하지 않는 계층 구조로 IMPORT하려면 다음 중 하나가 필요합니다.
  - *dbadm* 권한
  - 데이터베이스에 대한 CREATETAB 권한과 테이블 스페이스에 대한 USE 특권 및 다음 중 하나가 필요합니다.
    - 테이블의 스키마 이름이 존재하지 않을 경우, 데이터베이스에 대한 IMPLICIT\_SCHEMA 권한
    - 테이블의 스키마가 존재할 경우, 스키마에 대한 CREATEIN 특권
    - 전체 계층 구조에 대한 **REPLACE\_CREATE** 옵션이 사용될 경우, 계층 구조의 모든 서브테이블에 대한 CONTROL 특권
- **REPLACE** 옵션을 사용하여 기존 계층 구조로 IMPORT하려면 다음 중 하나가 필요합니다.
  - *dataaccess* 권한
  - 계층 구조에 있는 모든 서브테이블에 대한 CONTROL 특권
- 컬럼이 보호 설정된 테이블로 데이터를 임포트하려면 세션 권한 부여 ID가 테이블에 있는 보호 설정된 모든 컬럼에 대해 쓰기 액세스할 수 있도록 하는 LBAC 증명서를 가지고 있어야 합니다. 그렇지 않으면 임포트를 실패하고 오류(SQLSTATE 42512)가 리턴됩니다.

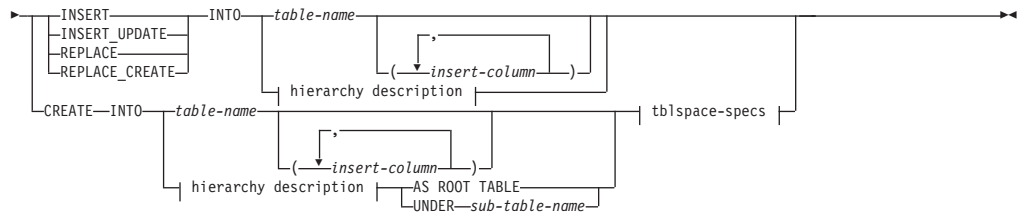
- 행이 보호 설정된 테이블로 데이터를 임포트하려면 세션 권한 부여 ID가 다음 기준을 충족하는 LBAC 증명서를 보유해야 합니다.
  - 테이블을 보호하는 보안 규정에 포함됨
  - 세션 권한 부여 ID에 쓰기 액세스에 대한 권한이 부여됨
 삽입할 행에 대한 레이블, 사용자의 LBAC 신임, 보안 규정 정의 및 LBAC 규칙이 행에 대한 레이블을 판별합니다.
- **REPLACE** 또는 **REPLACE\_CREATE** 옵션을 지정할 경우 세션 권한 부여 ID는 테이블을 삭제할 수 있는 권한이 있어야 합니다.
- 데이터를 별칭으로 임포트하려면 세션 권한 부여 ID에 pass-through 모드에서 지정된 데이터 소스를 액세스하고 사용할 특권이 있어야 합니다.

### 필수 연결

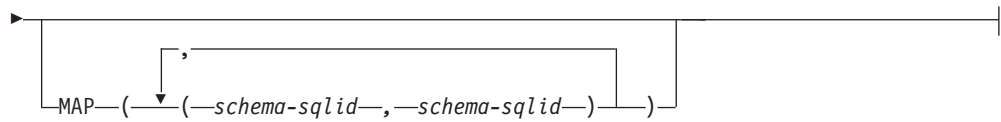
데이터베이스. Linux, UNIX 또는 Windows 클라이언트로부터 Linux, UNIX 또는 Windows 데이터베이스 서버에 대한 유틸리티 액세스는 DB2 Connect 게이트웨이 또는 루프 백 환경을 통하지 않고 엔진을 통해 직접 연결되어야 합니다.

### 명령 구문





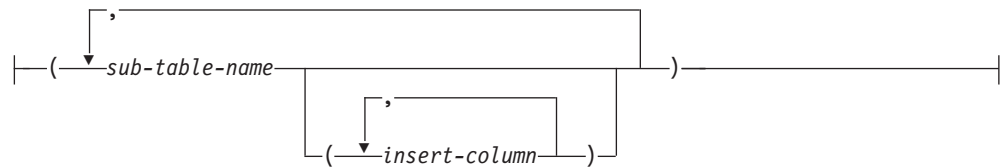
**Ignore 및 Map 매개변수:**



**계층 구조 설명:**



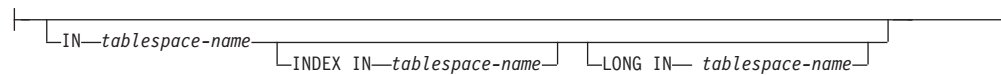
**서브 테이블 목록:**



**traversal-order-list:**



**tblspace-specs:**



## 명령 매개변수

### ALL TABLES

계층 구조 전용 내재적 키워드. 계층 구조를 임포트할 때 디폴트로 이동 순서로 지정된 모든 테이블을 임포트합니다.

### ALLOW NO ACCESS

오프라인 모드로 임포트를 실행합니다. 행이 삽입되기 전에 목표 테이블에 대해 독점(X) 잠금을 획득합니다. 이렇게 하면 동시 응용프로그램이 테이블 데이터에 액세스하지 못하게 합니다. 이는 디폴트 임포트 동작입니다.

### ALLOW WRITE ACCESS

온라인 모드로 임포트를 실행합니다. 첫 번째 행이 삽입될 때 목표 테이블에 대해 의도를 가진 독점(IX) 잠금을 획득합니다. 이렇게 하면 동시 관독기 및 기록기가 테이블 데이터에 액세스할 수 있습니다. 온라인 모드는 **REPLACE**, **CREATE** 또는 **REPLACE\_CREATE** 임포트 옵션과 호환 가능하지 않습니다. 온라인 모드는 버퍼링된 삽입과 함께 지원되지 않습니다. 임포트 조작은 테이블 잠금에 대한 잠금 에스컬레이션을 방지하고 사용 중인 로그 스페이스 외부에서 실행되지 않도록 하기 위해 삽입된 데이터를 주기적으로 커밋합니다. 이러한 커밋은 **COMMITCOUNT** 옵션을 사용하지 않는 경우에도 수행됩니다. 각 커밋을 수행하는 동안 임포트는 해당 IX 테이블 잠금을 유실하므로 커밋 후 이를 다시 획득하려고 시도합니다. 별칭으로 임포트할 때 이 매개변수가 필요하며 유효한 숫자를 사용하여 **COMMITCOUNT**를 지정해야 합니다(AUTOMATIC은 유효한 옵션으로 간주하지 않음).

### AS ROOT TABLE

하나 이상의 서브테이블을 독립형 테이블 계층 구조로 작성합니다.

### COMMITCOUNT *n* | AUTOMATIC

*n*개의 레코드가 모두 임포트된 후 COMMIT를 수행합니다. 숫자 *n*을 지정하면 임포트는 *n*개의 레코드가 모두 임포트된 후 COMMIT를 수행합니다. 복합 삽입을 사용할 때, 사용자 지정 커밋 빈도 *n*은 복합 계수 값의 첫 번째 정수 배수로 반올림됩니다. AUTOMATIC을 지정하면 임포트는 내부적으로 커밋을 수행해야 하는 때를 판별합니다. 이 유틸리티는 다음 두 가지 이유 중 하나에 대해 커밋합니다.

- 사용 중인 로그 스페이스 외부에서 실행하지 않도록 하기 위해
- 행 레벨에서 테이블 레벨로 잠금 에스컬레이션되지 않도록 하기 위해

**ALLOW WRITE ACCESS** 옵션을 지정하고 **COMMITCOUNT** 옵션을 지정하지 않으면, 임포트 유틸리티는 **COMMITCOUNT AUTOMATIC**이 지정된 것과 같이 커밋을 수행합니다.

사용 중인 로그 스페이스를 모두 사용해버리지 않도록 하기 위한 임포트 조작의 기능은 DB2 레지스트리 변수 **DB2\_FORCE\_APP\_ON\_MAX\_LOG**에 의해 영향을 받습니다.

- **DB2\_FORCE\_APP\_ON\_MAX\_LOG**가 FALSE로 설정되고 **COMMITCOUNT AUTOMATIC** 명령 옵션이 지정된 경우, 임포트 유틸리티는 자동으로 사용 중인 로그 스페이스를 모두 사용해버리지 않도록 할 수 있습니다.
- **DB2\_FORCE\_APP\_ON\_MAX\_LOG**가 FALSE로 설정되고 **COMMITCOUNT n** 명령 옵션이 지정된 경우, 레코드를 삽입하거나 갱신하는 중 SQL0964C(트랜잭션 로그가 가득참) 발생 시 임포트 유틸리티는 로그 가득참 조건을 해결하려고 합니다. 무조건 커미트를 수행한 후 레코드를 삽입하거나 갱신하려고 재시도합니다. 이것이 로그 가득참 조건(로그 가득참이 데이터베이스의 다른 활동에 영향을 주는 경우)을 해결하는 데 유용하지 않을 경우, **IMPORT** 명령은 예상대로 실패합니다. 그러나 커미트되는 행 수는 **COMMITCOUNT n** 값의 배수가 되지 않을 수 있습니다. 임포트 조작 재시도 시 이미 커미트된 행 처리를 피하려면, **RESTARTCOUNT** 또는 **SKIPCOUNT** 명령 매개변수를 사용하십시오.
- **DB2\_FORCE\_APP\_ON\_MAX\_LOG**가 TRUE로 설정된 경우(디폴트), 레코드를 삽입하거나 갱신하는 중 SQL0964C 발생 시 임포트 조작이 실패합니다. **COMMITCOUNT AUTOMATIC** 또는 **COMMITCOUNT n** 지정 여부와 관계 없이 이런 현상이 일어날 수 있습니다.

응용프로그램에서 데이터베이스가 강제 해제되고 현재 작업 단위(UOW)가 롤백됩니다. 임포트 조작 재시도 시 이미 커미트된 행 처리를 피하려면, **RESTARTCOUNT** 또는 **SKIPCOUNT** 명령 매개변수를 사용하십시오.

## CREATE

주: **CREATE** 매개변수는 사용되지 않으며 추후 릴리스에서 제거될 수 있습니다. 추가적인 세부사항은 『**IMPORT** 명령 옵션 **CREATE** 및 **REPLACE\_CREATE**가 사용되지 않음』을 참조하십시오.

데이터베이스의 코드 페이지에 테이블 정의 및 행 내용을 작성합니다. DB2 테이블, 서브테이블 또는 계층 구조에서 데이터를 익스포트한 경우, 인덱스가 작성됩니다. 이 옵션이 계층 구조에서 작동하고 DB2에서 데이터를 익스포트한 경우, 유형 계층 구조도 작성됩니다. 이 옵션은 IXF 파일에 대해서만 사용할 수 있습니다.

별칭으로 임포트할 때 이 매개변수는 유효하지 않습니다.

주: **MVS™** 호스트 데이터베이스에서 데이터를 익스포트하고 호스트 페이지 크기에 계산된 길이가 254를 초과하는 **LONGVAR** 필드를 포함할 경우, 행이

너무 길기 때문에 **CREATE**에 실패할 수 있습니다. 제한사항 목록에 대해서는 『임포트된 테이블 재작성』을 참조하십시오. 이 경우, 테이블을 직접 작성한 후 **INSERT**를 사용하여 **IMPORT**를 호출하거나 **LOAD** 명령을 사용해야 합니다.

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

이 옵션은 **USING XDS** 매개변수가 지정된 경우에만 사용할 수 있습니다. **DEFAULT** 절을 통해 지정된 스키마는 임포트된 XML 문서의 XDS(XML Data Specifier)가 XML 스키마를 식별하는 SCH 속성을 포함하지 않을 때 유효성 확인에 사용할 스키마를 식별합니다.

**DEFAULT** 절은 **IGNORE** 및 **MAP** 절보다 우선순위를 갖습니다. XDS가 **DEFAULT** 절을 충족시키면 **IGNORE** 및 **MAP** 스펙은 무시됩니다.

#### **FROM** *filename*

임포트될 데이터가 들어 있는 파일 이름을 지정합니다. 이는 완전한 경로여야 하며 해당 파일이 데이터베이스 서버에 존재해야 합니다.

#### **HIERARCHY**

계층 구조 데이터가 임포트되도록 지정합니다.

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

이 옵션은 **USING XDS** 매개변수가 지정된 경우에만 사용할 수 있습니다. SCH 속성에 의해 식별될 경우 **IGNORE** 절은 무시할 하나 이상의 스키마 목록을 지정합니다. 임포트된 XML 문서에 대한 XDS(XML Data Specifier)에 SCH 속성이 존재하고 SCH 속성에 의해 식별된 스키마가 무시할 스키마 목록에 포함된 경우, 이 임포트된 XML 문서에 대해서는 스키마 유효성 확인이 발생하지 않습니다.

**IGNORE** 절에 스키마가 지정되어 있으면 이 스키마는 또한 **MAP** 절에 있는 스키마 쌍의 왼쪽에 존재할 수 없습니다.

**IGNORE** 절은 XDS에만 적용됩니다. **MAP** 절에 의해 맵핑된 스키마는 **IGNORE** 절에 의해 지정된 경우 계속 무시되지 않습니다.

#### **IN** *tablespace-name*

테이블이 작성될 테이블 스페이스를 식별합니다. 테이블 스페이스가 있어야 하며 이 테이블 스페이스는 **REGULAR** 테이블 스페이스이어야 합니다. 다른 테이블 스페이스를 지정하지 않으면 모든 테이블 부분이 이 테이블 스페이스에 저장됩니다. 이 절을 지정하지 않으면 테이블은 권한 부여 ID가 작성한 테이블 스페이스에 작성됩니다. 테이블 스페이스를 찾을 수 없으면 테이블은 디폴트 테이블 스페이스 **USERSPACE1**에 넣어집니다. **USERSPACE1**가 삭제된 경우 테이블 작성에 실패합니다.

#### **INDEX IN** *tablespace-name*

테이블의 인덱스가 작성될 테이블 스페이스를 식별합니다. 이 옵션은 **IN** 절에



지정된 1차 테이블 스페이스가 DMS 테이블 스페이스일 경우에만 허용됩니다. 지정된 테이블 스페이스가 있어야 하며 이 테이블 스페이스는 REGULAR 또는 LARGE DMS 테이블 스페이스이어야 합니다.

주: 인덱스를 포함할 테이블 스페이스는 테이블이 작성되었을 때만 지정할 수 있습니다.

#### *insert-column*

데이터가 삽입될 테이블 또는 뷰에 있는 컬럼 이름을 지정합니다.

### **INSERT**

기존 테이블 데이터를 변경하지 않고 임포트된 데이터를 테이블에 추가합니다.

### **INSERT\_UPDATE**

임포트된 데이터 행을 목표 테이블에 추가하거나 기본 키가 일치하는 기존 행 (목표 테이블의)을 갱신합니다.

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

데이터가 임포트될 데이터베이스 테이블을 지정합니다. 이 테이블은 시스템 테이블, 작성된 임시 테이블, 선언된 임시 테이블 또는 요약 테이블이 될 수 없습니다.

완전한 테이블 이름 또는 규정되지 않은 테이블 이름을 사용해야 하는 이전 서버의 경우를 제외하고는 **INSERT**, **INSERT\_UPDATE** 또는 **REPLACE**에 별명을 사용할 수 있습니다. 규정된 테이블 이름의 형식은 *schema.tablename* 입니다. *schema*는 테이블이 작성된 사용자 이름입니다.

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

LOB 파일을 저장하는 하나 이상의 완전한 경로를 지정합니다. 이들 경로는 데이터베이스 서버 코디네이터 파티션에 있어야 합니다. LOB 데이터 파일의 이름은 기본 데이터 파일(ASC, DEL 또는 IXF)에 저장되는데 LOB 컬럼으로 로드될 컬럼에 저장됩니다. 지정할 수 있는 최대 경로 수는 999입니다. 이는 내재적으로 LOBSINFILE 동작을 활성화합니다.

별칭으로 임포트할 때 이 매개변수는 유효하지 않습니다.

#### **LONG IN** *tablespace-name*

긴 컬럼(LONG VARCHAR, LONG VARGRAPHIC, LOB 데이터 유형 또는 이들 유형 중 하나를 소스 유형으로 갖는 구별 유형)의 값이 저장될 테이블 스페이스를 식별합니다. 이 옵션은 **IN** 절에 지정된 1차 테이블 스페이스가 DMS 테이블 스페이스일 경우에만 허용됩니다. 테이블 스페이스가 있어야 하며 LARGE DMS 테이블 스페이스이어야 합니다.

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

이 옵션은 **USING XDS** 매개변수가 지정된 경우에만 사용할 수 있습니다. 임포트된 각 XML 문서에 대한 XDS(XML Data Specifier)의 SCH 속성이 지



정하는 스키마 대신 사용할 대체 스키마를 지정하려면 이 **MAP** 절을 사용하십시오. **MAP** 절은 하나 이상의 스키마 쌍의 목록을 지정하며, 여기서 각 쌍은 한 스키마 대 다른 스키마의 매핑을 나타냅니다. 쌍에서 첫 번째 스키마는 XDS에 있는 SCH 속성에 의해 참조되는 스키마를 나타냅니다. 쌍에서 두 번째 스키마는 스키마 유효성 확인을 수행하는 데 사용되어야 하는 스키마를 나타냅니다.

**MAP** 절에 있는 스키마 쌍의 왼쪽에 있는 스키마는 **IGNORE** 절에 지정될 수 없습니다.

스키마 쌍 매핑이 적용된 후, 최종 결과가 됩니다. 매핑 조작은 전이되지 않으므로 선택된 스키마는 다른 스키마 상 매핑에 계속 적용되지 않습니다.

스키마가 두 번 이상 매핑될 수 없다는 것은 쌍의 왼쪽에 두 번 이상 나타날 수 없다는 것을 의미합니다.

## MESSAGES ON SERVER

IMPORT 명령으로 서버에 작성한 메시지 파일을 저장하도록 지정합니다. 리턴되는 결과 세트에는 이 조작 중에 발생하는 경고 및 오류 메시지를 모두 검색하는 데 필요한 SQL문 MSG\_RETRIEVAL 컬럼과 메시지를 정리하는 데 필요한 SQL문인 MSG\_REMOVAL 컬럼이 필요합니다.

이 절을 지정하지 않으면 ADMIN\_CMD 프로시저가 호출자로 리턴될 때 메시지 파일이 삭제됩니다. 결과 세트의 MSG\_RETRIEVAL 및 MSG\_REMOVAL 컬럼에 널(NULL) 값이 포함됩니다.

이 절의 사용여부에 관계없이 분리(fenced) 사용자 ID에는 데이터가 익스포트될 디렉토리와, **DB2\_UTIL\_MSGPATH** 레지스트리 변수가 표시하는 디렉토리 아래에 파일을 작성할 수 있는 권한이 있어야 합니다.

## METHOD

**L** 데이터를 임포트할 시작 및 끝 컬럼 번호를 지정합니다. 컬럼 번호는 데이터 행이 시작되는 바이트 오프셋입니다. 1부터 번호가 매겨집니다.

주: 이 방법은 ASC 파일에만 사용할 수 있으며 이 파일 유형에 대해 유일한 유효 옵션입니다.

**N** 임포트할 데이터 파일의 컬럼 이름을 지정합니다. 이 컬럼 이름의 대소문자는 시스템 카탈로그의 해당 이름의 대소문자와 일치해야 합니다. 널(NULL) 입력이 가능하지 않은 각 테이블 컬럼에는 **METHOD N** 목록에 해당 항목이 있어야 합니다. 예를 들면, 주어진 데이터 필드 F1, F2, F3, F4, F5 및 F6과 테이블 컬럼 C1 INT, C2 INT NOT NULL, C3 INT NOT NULL 및 C4 INT, method N (F2, F1, F4, F3)은 유효한 요청인 반면, method N (F2, F1)은 유효하지 않습니다.

주: 이 방법은 IXF 파일에 대해서만 사용할 수 있습니다.



별칭으로 임포트할 때 WSF 파일 유형은 지원되지 않습니다.

## REPLACE

데이터 오브젝트를 잘라내어 테이블에서 모든 기존 데이터를 삭제한 후 임포트된 데이터를 삽입합니다. 테이블 정의 및 인덱스 정의는 변경되지 않습니다. 이 옵션은 테이블이 존재할 경우에만 사용할 수 있습니다. 계층 구조 간에 데이터를 이동할 때 이 옵션을 사용하면 개별적 서브테이블이 아닌 전체 계층 구조에 대한 데이터만 바꿀 수 있습니다.

별칭으로 임포트할 때 이 매개변수는 유효하지 않습니다.

이 옵션은 CREATE TABLE문의 NOT LOGGED INITIALLY(NLI)절이나 ALTER TABLE문의 ACTIVE NOT LOGGED INITIALLY절을 인정하지 않습니다.

NLI절이 호출된 CREATE TABLE 또는 ALTER TABLE문과 동일한 트랜잭션에서 **REPLACE** 옵션을 사용한 임포트를 수행할 경우, 임포트는 NLI절을 인정하지 않습니다. 모든 삽입은 로그됩니다.

### 일시적인 해결책 1

DELETE문을 사용하여 테이블의 내용을 삭제한 후 INSERT문을 사용하여 임포트를 호출하십시오.

### 일시적인 해결책 2

테이블을 삭제하고 재작성한 후 INSERT문을 사용하여 임포트를 호출하십시오.

이 제한사항은 DB2 Universal Database 버전 7 및 DB2 UDB 버전 8에 적용됩니다.

## REPLACE\_CREATE

주: **REPLACE\_CREATE** 매개변수는 사용되지 않으며 추후 릴리스에서 제거될 수 있습니다. 추가적인 세부사항은 『IMPORT 명령 옵션 CREATE 및 REPLACE\_CREATE가 사용되지 않음』을 참조하십시오.

테이블이 존재할 경우, 데이터 오브젝트를 잘라내어 테이블에서 모든 기존 데이터를 삭제한 후 테이블 정의 또는 인덱스 정의를 변경하지 않고 임포트된 데이터를 삽입합니다.

테이블이 존재하지 않을 경우, 데이터베이스의 코드 페이지에 행 내용과 함께 테이블 및 인덱스 정의를 작성합니다. 제한사항 목록에 대해서는 임포트된 테이블 재작성을 참조하십시오.

이 옵션은 IXF 파일에 대해서만 사용할 수 있습니다. 계층 구조 간에 데이터를 이동하는 경우에 이 옵션을 사용하면 개별 부속 테이블이 아닌 전체 계층 구조에 대한 데이터만 바꿀 수 있습니다.

별칭으로 импорт할 때 이 매개변수는 유효하지 않습니다.

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

임포트 조작이 레코드  $n + 1$ 에서 시작되도록 지정합니다. 첫 번째  $n$  레코드는 건너뜁니다. 이 솔루션은 기능적으로 **SKIPCOUNT**와 동일합니다. **RESTARTCOUNT** 및 **SKIPCOUNT**는 상호 배타적입니다.

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

파일에 있는  $n$ 개의 실제 레코드가 импорт(삽입되거나 갱신됨)되도록 지정합니다. 사용자가 **SKIPCOUNT** 또는 **RESTARTCOUNT** 옵션에 의해 판별된 레코드로부터  $n$ 개의 행만을 파일에서 импорт할 수 있게 합니다. **SKIPCOUNT** 또는 **RESTARTCOUNT** 옵션이 지정되지 않은 경우 첫 번째  $n$  행만 импорт합니다. **SKIPCOUNT**  $m$  또는 **RESTARTCOUNT**  $m$ 이 지정된 경우  $m+1$ 에서  $m+n$ 까지의 행을 импорт합니다. 복합 삽입을 사용할 때, **ROWCOUNT**  $n$ 은 복합 계수 값의 첫 번째 정수 배수로 받아들여집니다.

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

임포트 조작이 레코드  $n + 1$ 에서 시작되도록 지정합니다. 첫 번째  $n$  레코드는 건너뜁니다. 이 옵션은 기능적으로 **RESTARTCOUNT**와 동일합니다. **SKIPCOUNT** 및 **RESTARTCOUNT**는 상호 배타적입니다.

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

*sub-table-name*부터 시작하고 디폴트 순서를 요청하며 계층 구조 전용 키워드입니다. PC/IXF 파일의 경우 디폴트 순서는 입력 파일에 저장된 순서입니다. PC/IXF 파일 형식에는 디폴트 순서만 사용할 수 있습니다.

#### *sub-table-list*

**INSERT** 또는 **INSERT\_UPDATE** 옵션을 사용하는 유형이 지정된 테이블의 경우, 데이터가 импорт될 서브테이블을 표시하기 위해 서브테이블 이름 목록을 사용합니다.

#### *traversal-order-list*

**INSERT**, **INSERT\_UPDATE** 또는 **REPLACE** 옵션을 사용하는 유형이 지정된 테이블의 경우, 계층 구조에서 импорт 중인 서브테이블의 이동 순서를 표시하기 위해 서브테이블 이름 목록을 사용합니다.

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

하나 이상의 서브테이블을 작성하기 위한 상위 테이블을 지정합니다.

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

$n$ 번의 경고 후 импорт 조작을 중지합니다. 경고가 예상되지는 않지만 올바른 파일 및 테이블이 사용되고 있는지에 대한 검증을 원할 경우 이 매개변수를 설정하십시오. импорт 파일 또는 목표 테이블이 올바르지 않게 지정될 경우 импорт

유틸리티는 임포트하려는 각 행에 대해 경고를 생성하므로 임포트를 실패하게 됩니다. *n*이 0이거나 이 옵션이 지정되지 않은 경우 임포트 조작은 발행된 경고 수에 관계없이 계속 수행됩니다.

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

XML 파일이 들어 있는 하나 이상의 경로를 지정합니다.

#### **XMLPARSE**

XML 문서가 구문 분석되는 방법을 지정합니다. 이 옵션을 지정하지 않을 경우, XML 문서에 대한 구문 분석 동작은 CURRENT XMLPARSE OPTION 특수 레지스터의 값으로 판별됩니다.

#### **STRIP WHITESPACE**

XML 문서가 구문 분석될 때 공백을 제거하도록 지정합니다.

#### **PRESERVE WHITESPACE**

XML 문서가 구문 분석될 때 공백을 제거하지 않도록 지정합니다.

#### **XMLVALIDATE**

XML 문서가 스키마에 대해 유효성이 확인되도록 지정합니다(해당되는 경우).

#### **USING XDS**

기본 데이터 파일의 XDS(XML Data Specifier)에 의해 식별된 XML 스키마에 대해 XML 문서의 유효성을 확인합니다. **USING XDS**와 함께 **XMLVALIDATE** 옵션을 호출한 경우, 유효성 확인을 수행하는 데 사용된 스키마는 디폴트로 XDS의 SCH 속성에 의해 판별됩니다. SCH 속성이 XDS에 존재하지 않을 경우, **DEFAULT** 절로 디폴트 스키마를 지정하지 않으면 스키마 유효성 확인이 발생하지 않습니다.

**DEFAULT**, **IGNORE** 및 **MAP** 절은 스키마 판별 동작을 수정하는 데 사용될 수 있습니다. 이들 세 개의 선택적 절은 XDS의 권장 스펙에 직접적으로 적용되며 서로에게는 적용되지 않습니다. 예를 들어, 한 스키마가 **DEFAULT** 절에서 지정되어 선택되었으면 이 스키마는 **IGNORE** 절에 지정되어도 무시되지 않습니다. 마찬가지로 한 스키마가 **MAP** 절에서 첫 번째 파트 쌍으로 지정되어 선택되면 다른 **MAP** 절 쌍의 두 번째 파트에 지정되어도 다시 맵핑되지 않습니다.

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

XML 문서가 지정된 SQL ID가 있는 XML 스키마에 대해 유효성이 확인됩니다. 이 경우 모든 XML 컬럼에 대해 XDS(XML Data Specifier)의 SCH 속성이 무시됩니다.

#### **USING SCHEMALOCATION HINTS**

XML 문서가 소스 XML 문서의 XML 스키마 위치 힌트에 의해 식별된 스키마에 대해 유효성이 확인됩니다. XML 문서에서 `schemaLocation` 속성을 찾을 수 없으면 유효성 확인이 발생하지 않습니다.

니다. **USING SCHEMALOCATION HINTS** 절을 지정하면 모든 XML 컬럼에 대해 XDS(XML Data Specifier)의 SCH 속성이 무시됩니다.

아래의 **XMLVALIDATE** 옵션 예를 참조하십시오.

**예 :**

다음 예는 myfile.ixf 파일에서 SAMPLE 데이터베이스의 STAFF 테이블로 정보를 임포트하는 방법을 보여 줍니다.

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

## 사용 시 참고사항

IMPORT 명령에 사용되는 경로는 서버에 대한 코디네이터 노드에서 유효한 완전한 경로여야 합니다.

**ALLOW WRITE ACCESS** 또는 **COMMITCOUNT** 옵션이 지정된 경우 임포트 유틸리티에 의해 커미트가 수행됩니다. 이로 인해 유형 2 연결의 경우 ADMIN\_CMD 프로시저는 이유 코드 1과 함께 SQL30090N 오류를 리턴합니다.

ADMIN\_CMD 프로시저에서 결과 세트의 컬럼에 지정되는 값이 해당 컬럼의 데이터 유형에 대한 최대값을 초과할 경우 데이터 유형의 최대값이 지정되고 경고 메시지 SQL1155W가 리턴됩니다.

임포트 조작을 시작하기 전에 반드시 모든 테이블 조작을 완료하고 모든 잠금을 릴리스하십시오. 이는 WITH HOLD로 열린 모든 커서를 닫고 COMMIT를 발행하거나 ROLLBACK을 발행하여 수행할 수 있습니다.

임포트 유틸리티는 SQL INSERT문을 사용하여 목표 테이블에 행을 추가합니다. 이 유틸리티는 입력 파일에 있는 각 데이터 행마다 하나의 INSERT문을 발행합니다. INSERT문을 실패하면 다음 두 가지 조치 중 하나가 발생합니다.

- 후속 INSERT문이 성공할 수 있을 것 같으면 메시지 파일에 경고 메시지를 작성하고 처리를 계속합니다.
- 후속 INSERT문이 실패할 것 같으며 데이터베이스 손상이 생길 가능성이 있으면 메시지 파일에 오류 메시지를 작성하고 처리를 정지합니다.

이 유틸리티는 **REPLACE** 또는 **REPLACE\_CREATE** 조작 중 이전 행이 삭제된 후 자동 COMMIT를 수행합니다. 따라서 테이블 오브젝트가 잠려진 후 응용프로그램이 데이터베이스 관리 프로그램을 인터럽트하거나 시스템이 실패할 경우 이전 데이터가 모두 유실됩니다. 이들 옵션을 사용하기 전에 이전 데이터가 더 이상 필요하지 않은지 확인하십시오.



**CREATE**, **REPLACE** 또는 **REPLACE\_CREATE** 조작 중 로그가 가득차게 되면 이 유틸리티는 삽입된 레코드에 대해 자동 **COMMIT**를 수행합니다. 자동 **COMMIT** 후 응용프로그램이 데이터베이스 관리 프로그램을 인터럽트하거나 시스템이 실패할 경우 부분 데이터가 있는 테이블이 데이터베이스에 남아 있습니다. **REPLACE** 또는 **REPLACE\_CREATE** 옵션을 사용하여 전체 импорт 조작을 재실행하거나 성공적으로 импорт된 행 수로 설정된 **RESTARTCOUNT** 매개변수를 사용하여 **INSERT**를 사용하십시오.

디폴트로 **INSERT** 또는 **INSERT\_UPDATE** 옵션에 대해서는 자동 **COMMIT**가 수행되지 않습니다. 단, **COMMITCOUNT** 매개변수가 0이 아닌 경우에는 수행됩니다. 자동 **COMMIT**가 수행되지 않으면 가득찬 로그는 **ROLLBACK**됩니다.

다음 조건 중 하나라도 해당될 경우 오프라인 임포트는 자동 **COMMIT**를 수행하지 않습니다.

- 목표가 테이블이 아니고 뷰인 경우
- 복합 텍스트 삽입이 사용된 경우
- 버퍼링된 삽입이 사용된 경우

디폴트로 온라인 임포트는 자동 **COMMIT**를 수행하여 사용 중인 로그 스페이스와 잠금 목록을 모두 해제합니다. 자동 **COMMIT**는 **COMMITCOUNT** 값이 0으로 지정된 경우에만 수행되지 않습니다.

임포트 유틸리티가 **COMMIT**를 수행할 때마다 메시지 파일에 두 개의 메시지가 작성되는데, 하나는 커밋되는 레코드 수를 나타내고 다른 하나는 **COMMIT** 완료 후 작성됩니다. 실패 후 импорт 조작을 재시작할 때 마지막 완료된 **COMMIT**에서 판별된 대로 생략할 레코드 수를 지정하십시오.

임포트 유틸리티는 사소한 비호환성 문제점이 있는 입력 데이터는 승인합니다(예를 들어, 문자 데이터가 채우기 또는 절단을 사용하여 импорт되고 숫자 데이터가 다른 숫자 데이터 유형으로 импорт될 수 있습니다). 그러나 주요한 비호환성 문제점이 있는 데이터는 승인되지 않습니다.

오브젝트 테이블이 자체 테이블 외에 하위 테이블이 있는 경우 이 오브젝트 테이블을 **REPLACE** 또는 **REPLACE\_CREATE** 할 수 없으며, 기본 테이블에 하위 테이블(자체 테이블을 포함하여)이 있는 경우 해당 오브젝트 뷰를 **REPLACE** 또는 **REPLACE\_CREATE** 할 수 없습니다. 이러한 테이블 또는 뷰를 바꾸려면 다음을 수행하십시오.

1. 해당 테이블이 상위 테이블인 모든 외부 키를 삭제하십시오.
2. 임포트 유틸리티를 실행하십시오.
3. 테이블을 변경하여 외부 키를 재작성하십시오.

외부 키를 재작성하는 동안 오류가 발생하면 데이터를 수정하여 참조 무결성을 유지보수하십시오.

PC/IXF 파일에서 테이블을 재작성할 때 참조 제한조건 및 외부 키 정의는 보존되지 않습니다. (기본 키 정의는 데이터가 이전에 SELECT \*를 사용하여 익스포트된 경우에 보존됩니다.)

리모트 데이터베이스로 임포트하려면 입력 데이터 파일의 사본, 출력 메시지 파일 및 잠재적인 데이터베이스 크기 확장에 필요한 디스크 스페이스가 충분해야 합니다.

리모트 데이터베이스에 대해 임포트 조작이 수행되고 출력 메시지 파일이 매우 긴 경우 (60KB 초과), 클라이언트의 사용자에게 리턴되는 메시지 파일은 임포트 조작 중간부터 메시지가 누락될 수 있습니다. 메시지 정보의 처음 30KB와 메시지 정보의 마지막 30KB는 항상 보유됩니다.

PC/IXF 파일이 디스켓 대신 하드 드라이브에 있을 경우 PC/IXF 파일을 리모트 데이터베이스로 임포트하면 훨씬 더 빨리 수행됩니다.

ASC, DEL 또는 WSF 파일 형식의 데이터를 임포트하려면 먼저 데이터베이스 테이블 또는 계층 구조가 존재해야 합니다. 그러나 테이블이 이미 존재하지 않을 경우 **IMPORT CREATE** 또는 **IMPORT REPLACE\_CREATE**가 PC/IXF 파일에서 데이터를 임포트할 때 테이블을 작성합니다. 유형이 지정된 테이블에서 **IMPORT CREATE**는 유형 계층 구조와 테이블 계층 구조를 작성할 수 있습니다.

데이터베이스 간에 데이터를 이동하려면(계층 데이터를 포함하여) PC/IXF 임포트를 사용해야 합니다. 행 구분자가 있는 문자 데이터를 컬럼 식별자가 있는 ASCII(DEL) 파일로 익스포트하며 텍스트 전송 프로그램으로 이를 처리할 경우 행 구분자가 있는 필드는 줄어들거나 확장됩니다. 동일한 클라이언트에서 소스 또는 목표 데이터베이스에 모두 액세스할 수 있는 경우 파일 복사 단계는 필요하지 않습니다.

ASC 및 DEL 파일의 데이터는 임포트를 수행하는 클라이언트 응용프로그램의 코드 페이지에 있는 것으로 간주됩니다. 서로 다른 코드 페이지의 데이터를 임포트할 때는 서로 다른 코드 페이지에 사용할 수 있는 PC/IXF 파일을 사용할 것을 권장합니다. PC/IXF 파일과 임포트 유틸리티가 동일한 코드 페이지에 있을 경우 일반 응용프로그램에 대해서와 같은 처리가 발생합니다. 두 코드 페이지가 서로 다르고 **FORCEIN** 옵션이 지정된 경우, 임포트 유틸리티는 PC/IXF 파일의 데이터가 임포트를 수행 중인 응용프로그램과 동일한 코드 페이지를 가지고 있다고 간주합니다. 이는 두 코드 페이지에 대한 변환 테이블이 있는 경우에도 마찬가지입니다. 두 코드 페이지가 서로 다르고, **FORCEIN** 옵션이 지정되지 않았으며 변환 테이블이 있는 경우, PC/IXF 파일의 모든 데이터는 파일 코드 페이지에서 응용프로그램 코드 페이지로 변환됩니다. 두 코드 페이지가 서로 다



르고 **FORCEIN** 옵션이 지정되지 않았으며 변환 테이블이 없는 경우 импорт 조작용 실패합니다. 이는 AIX 운영 체제의 DB2 클라이언트에 있는 PC/IXF 파일에만 적용됩니다.

1012 컬럼의 한계에 가까운 8KB 페이지에 있는 테이블 오브젝트의 경우, PC/IXF 데이터 파일을 импорт하면 DB2가 오류를 리턴할 수 있는데, 이는 SQL문의 최대 크기를 초과했기 때문입니다. 이러한 상황은 컬럼 유형이 CHAR, VARCHAR 또는 CLOB인 경우에만 발생할 수 있습니다. **DEL** 또는 **ASC** 파일을 импорт하는 데는 이러한 제한 사항이 적용되지 않습니다. PC/IXF 파일을 사용하여 새 테이블을 작성할 경우 다른 방법은 db2look을 사용하여 테이블을 작성한 DDL문을 덤프한 후 CLP를 통해 이 명령문을 발행하는 것입니다.

DB2 Connect는 OS/390용 DB2, VM 및 VSE용 DB2, OS/400용 DB2와 같은 DRDA 서버로 데이터를 импорт하는 데 사용할 수 있습니다. PC/IXF импорт(**INSERT** 옵션)만 지원됩니다. **RESTARTCOUNT** 매개변수도 지원되지만 **COMMITCOUNT** 매개변수는 지원되지 않습니다.

유형이 지정된 테이블에서 **CREATE** 옵션을 사용할 때 PC/IXF 파일에 정의된 모든 서브테이블을 작성하십시오. 서브테이블 정의는 변경할 수 없습니다. 유형이 지정된 테이블에서 **CREATE** 이외의 옵션을 사용할 때, 이동 순서 목록을 통해 해당 옵션이 이동 순서를 지정하므로 이동 순서 목록은 익스포트 조작 시 사용된 옵션과 일치해야 합니다. PC/IXF 파일 형식의 경우, 옵션은 목표 서브테이블 이름만 지정해야 하며 파일에 저장된 이동 순서를 사용해야 합니다.

임포트 유틸리티를 사용하여 이전에 PC/IXF 파일로 익스포트된 테이블을 복구할 수 있습니다. 이 테이블은 익스포트될 때의 상태로 리턴합니다.

데이터는 시스템 테이블, 작성된 임시 테이블, 선언된 임시 테이블 또는 요약 테이블로 импорт될 수 없습니다.

뷰는 импорт 유틸리티를 통해 작성될 수 없습니다.

개별적 파트가 Windows 시스템에서 AIX 시스템으로 복사되는 다중 파트 PC/IXF 파일 imports가 지원됩니다. 첫 번째 파일의 이름만 **IMPORT** 명령에 지정되어야 합니다. 예를 들어, **IMPORT FROM data.ixf OF IXF INSERT INTO TABLE1**. 파일 data.002, etc가 data.ixf와 동일한 디렉토리에서 사용 가능해야 합니다.

Windows 운영 체제에서:

- 논리적으로 분할된 PC/IXF 파일 imports는 지원되지 않습니다.
- 잘못된 형식의 PC/IXF 또는 WSF 파일 imports는 지원되지 않습니다.

내부 형식으로 된 보안 레이블에는 줄 바꾸기 문자가 포함될 수 있습니다. **DEL** 파일 형식을 사용하여 파일을 imports할 경우 이들 줄 바꾸기 문자는 분리문자로 오인될 수

있습니다. 이러한 문제점이 발생하면 `IMPORT` 명령에 `delprioritychar` 파일 유형 수정자를 지정하여 분리문자에 대해 이전 디폴트 우선순위를 사용하십시오.

## 페더레이티드 고려사항

`IMPORT` 명령 및 `INSERT`, `UPDATE` 또는 `INSERT_UPDATE` 명령 매개변수를 사용할 때 참여 중인 별칭에 대해 `CONTROL` 특권을 가지고 있는지 확인해야 합니다. импорт 작업을 수행할 때 사용할 별칭이 이미 존재하는지 확인해야 합니다. 또한 `IMPORT` 명령 매개변수 섹션에서 표시된 바와 같이 여러 가지 제한사항을 유념해야 합니다.

ODBC와 같은 일부 데이터 소스는 별칭으로의 임포트를 지원하지 않습니다.

## 결과 세트 정보

명령 실행 상태는 `CALL`문에서 생성된 `SQLCA`에 리턴됩니다. 실행이 완료되면 명령은 다음과 같이 결과 세트로 추가 정보를 리턴합니다.

표 42. `IMPORT` 명령에서 리턴된 결과 세트

컬럼 이름	데이터 유형	설명
<code>ROWS_READ</code>	<code>BIGINT</code>	임포트 중 파일에서 읽은 레코드 수
<code>ROWS_SKIPPED</code>	<code>BIGINT</code>	삽입 또는 갱신이 시작되기 전에 건너뛴 레코드 수
<code>ROWS_INSERTED</code>	<code>BIGINT</code>	목표 테이블로 삽입된 행 수
<code>ROWS_UPDATED</code>	<code>BIGINT</code>	임포트된 레코드(테이블에 기본 키 값이 이미 존재하는 레코드)의 정보로 갱신된 목표 테이블의 행 수
<code>ROWS_REJECTED</code>	<code>BIGINT</code>	임포트할 수 없는 레코드 수
<code>ROWS_COMMITTED</code>	<code>BIGINT</code>	데이터베이스에 임포트되고 커밋된 레코드 수
<code>MSG_RETRIEVAL</code>	<code>VARCHAR(512)</code>	이 유틸리티에서 작성된 메시지를 검색하는 데 사용되는 SQL문. 예를 들면, 다음과 같습니다.  <pre>SELECT SQLCODE, MSG       FROM TABLE (SYSPROC.ADMIN_GET_MSGS                    ('1203498_txu')) AS MSG</pre>
<code>MSG_REMOVAL</code>	<code>VARCHAR(512)</code>	이 유틸리티에서 작성된 메시지를 정리하는 데 사용되는 SQL문. 예를 들면, 다음과 같습니다.  <pre>CALL SYSPROC.ADMIN_REMOVE_MSGS       ('1203498_txu')</pre>

## 임포트 유틸리티의 파일 유형 수정자

표 43. 임포트 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식

수정자	설명
compound=x	<p>x는 1 - 100의 숫자입니다. nonatomic 복합 SQL을 사용하여 데이터를 삽입하며, x 명령문이 매번 시도됩니다.</p> <p>수정자가 지정되고 트랜잭션 로그가 충분히 크지 않으면, 임포트 조작이 실패합니다. <b>COMMITCOUNT</b>에서 지정한 행 수나, <b>COMMITCOUNT</b>가 지정되지 않은 경우 데이터 파일의 행 수를 수용할 정도로 트랜잭션 로그가 커야 합니다. 그러므로 트랜잭션 로그 오버플로우를 피하기 위해 <b>COMMITCOUNT</b> 옵션을 지정하도록 권장합니다.</p> <p>이 수정자는 <b>INSERT_UPDATE</b> 모드, 계층 테이블 및 다음 수정자와 호환되지 않습니다. <b>usedefaults</b>, <b>identitymissing</b>, <b>identityignore</b>, <b>generatedmissing</b> 및 <b>generatedignore</b>.</p>
generatedignore	<p>이 수정자는 생성된 모든 컬럼의 데이터가 데이터 파일에 표시되지만 무시해야 함을 임포트 유틸리티에 알립니다. 이로 인해 생성된 컬럼의 모든 값이 유틸리티에 의해 생성됩니다. 이 수정자는 <b>generatedmissing</b> 수정자와 함께 사용될 수 없습니다.</p>
generatedmissing	<p>이 수정자가 지정된 경우, 유틸리티는 입력 데이터 파일에는 생성된 컬럼의 데이터가 포함되지 않는 것으로(NULL 값도 비포함) 가정하므로 각 행의 값을 생성합니다. 이 수정자는 <b>generatedignore</b> 수정자와 함께 사용될 수 없습니다.</p>
identityignore	<p>이 수정자는 ID 컬럼의 데이터가 데이터 파일에 표시되지만 무시해야 함을 임포트 유틸리티에 알립니다. 이로 인해 유틸리티에 의해 모든 ID 값이 생성됩니다. <b>GENERATED ALWAYS</b> 및 <b>GENERATED BY DEFAULT ID</b> 컬럼 둘 다의 동작이 동일합니다. <b>GENERATED ALWAYS</b> 컬럼의 경우, 어떤 행도 거부되지 않음을 의미합니다. 이 수정자는 <b>identitymissing</b> 수정자와 함께 사용될 수 없습니다.</p>
identitymissing	<p>이 수정자가 지정된 경우, 유틸리티는 입력 데이터 파일에는 ID 컬럼의 데이터가 포함되지 않는 것으로(NULL 값도 비포함) 가정하므로 각 행의 값을 생성합니다. <b>GENERATED ALWAYS</b> 및 <b>GENERATED BY DEFAULT ID</b> 컬럼 둘 다의 동작이 동일합니다. 이 수정자는 <b>identityignore</b> 수정자와 함께 사용될 수 없습니다.</p>
lobsinfile	<p><i>lob-path</i>는 LOB 데이터를 포함하는 파일에 대한 경로를 지정합니다.</p> <p>각 경로는 LLS(Lob Location Specifier)에 의해 포인트되는 최소한 하나의 LOB를 포함하는 최소한 하나의 파일을 데이터 파일에 포함합니다. LLS는 LOB 파일 경로에 저장된 파일에서 LOB 위치의 문자열 표시입니다. LLS 형식은 <i>filename.ext.nnn.mmm</i>이며, 여기서 <i>filename.ext</i>는 LOB를 포함하는 파일의 이름이며, <i>nnn</i>은 파일 내에서 LOB의 오프셋을 바이트로 나타낸 것이며, <i>mmm</i>은 LOB의 길이를 바이트로 나타낸 것입니다. 예를 들어, 문자열 db2exp.001.123.456/가 데이터 파일에 저장되는 경우, LOB는 db2exp.001 파일에서 오프셋 123에 위치하며 456바이트 길이입니다.</p> <p><b>LOBS FROM</b> 절은 『lobsinfile』 수정자가 사용될 때 LOB 파일이 위치하는 곳을 지정합니다. <b>LOBS FROM</b> 절은 내재적으로 LOBSINFILE 동작을 활성화합니다. <b>LOBS FROM</b> 절은 데이터 임포트 중 LOB 파일을 검색하기 위해 경로 목록을 IMPORT 유틸리티로 전달합니다.</p> <p>널(NULL) LOB를 표시하려면, 크기를 -1로 입력하십시오. 크기가 0으로 지정되면, 길이가 0인 LOB로 처리됩니다. 길이가 -1인 널(NULL) LOBS의 경우, 오프셋 및 파일 이름은 무시됩니다. 예를 들어, 널(NULL) LOB의 LLS는 db2exp.001.7.-1/입니다.</p>
no_type_id	<p>단일 서브테이블로 임포트할 때에만 유효합니다. 일반 설치는 일반 테이블에서 데이터를 익스포트한 후 임포트 조작을 호출하여(이 수정자 사용) 데이터를 단일 서브테이블로 변환합니다.</p>

표 43. 임포트 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
nodefaults	<p>목표 테이블 컬럼의 소스 컬럼이 명시적으로 지정되지 않고, 테이블 컬럼이 널(NULL) 입력 가능하지 않으면, 디폴트값이 로드되지 않습니다. 목표 테이블 컬럼 중 하나의 소스 컬럼이 명시적으로 지정되지 않은 경우, 이 옵션이 없으면 다음 중 하나가 발생합니다.</p> <ul style="list-style-type: none"> <li>• 컬럼의 디폴트값이 지정될 수 있는 경우, 디폴트값이 로드됩니다.</li> <li>• 컬럼이 널(NULL) 입력 가능하고 해당 컬럼의 디폴트값을 지정할 수 없는 경우, NULL이 로드됩니다.</li> <li>• 컬럼이 널(NULL) 입력 가능하고 디폴트값을 지정할 수 없는 경우, 오류가 리턴되며 유틸리티가 처리를 중지합니다.</li> </ul>
norowwarnings	거부된 행에 대한 모든 경고를 제외시킵니다.
rowchangetimestampignore	이 수정자는 행 변경 시간소인 컬럼의 데이터가 데이터 파일에 표시되지만 무시해야 함을 임포트 유틸리티에 알립니다. 이로 인해 유틸리티에 의해 모든 ROW CHANGE TIMESTAMP가 생성됩니다. GENERATED ALWAYS 및 GENERATED BY DEFAULT 컬럼 둘다의 동작이 동일합니다. GENERATED ALWAYS 컬럼의 경우, 어떤 행도 거부되지 않음을 의미합니다. 이 수정자는 rowchangetimestampmissing 수정자와 함께 사용될 수 없습니다.
rowchangetimestampmissing	이 수정자가 지정된 경우, 유틸리티는 입력 데이터 파일에는 행 변경 시간소인 컬럼의 데이터를 포함하지 않음(NULL 값도 비포함)을 가정하므로 각 행의 값을 생성합니다. GENERATED ALWAYS 및 GENERATED BY DEFAULT 컬럼 둘 다의 동작이 동일합니다. 이 수정자는 rowchangetimestampignore 수정자와 함께 사용될 수 없습니다.
seclabelchar	<p>입력 소스 파일의 보안 레이블이 디폴트로 인코딩된 숫자 형식이 아니라 보안 레이블 값의 문자열 형식임을 표시합니다. IMPORT는 각 보안 레이블을 로드된 대로의 내부 형식으로 변환합니다. 문자열이 적절한 형식으로 되어 있지 않은 경우 행은 로드되지 않으며 경고(SQLSTATE 01H53)가 리턴됩니다. 문자열이 테이블을 보호하는 보안 규정 패턴인 유효한 보안 레이블을 나타내지 않는 경우, 행은 로드되지 않으며 경고(SQLSTATE 01H53, SQLCODE SQL3243W)가 리턴됩니다.</p> <p>seclabelname 수정자가 지정된 경우 이 수정자를 지정할 수 없으며, 그렇지 않으면 임포트에 실패하며 오류(SQLCODE SQL3525N)가 리턴됩니다.</p>
seclabelname	<p>입력 소스 파일의 보안 레이블이 디폴트로 인코딩된 숫자 형식이 아니라 이름으로 표시됨을 나타냅니다. IMPORT는 이름이 있는 경우 적절한 보안 레이블로 변환합니다. 테이블을 보호하는 보안 규정에 대해 표시된 이름이 있는 보안 레이블이 없는 경우, 행은 로드되지 않으며 경고(SQLSTATE 01H53, SQLCODE SQL3244W)가 리턴됩니다.</p> <p>seclabelchar 수정자가 지정된 경우 이 수정자를 지정할 수 없으며, 그렇지 않으면 임포트에 실패하며 오류(SQLCODE SQL3525N)가 리턴됩니다.</p> <p>주: 파일 유형이 ASC인 경우, 보안 레이블의 이름 다음에 오는 모든 스페이스는 이름의 일부로 해석됩니다. 이를 피하려면 striptblanks 파일 유형 수정자를 사용하여 스페이스가 제거되었는지 확인하십시오.</p>

표 43. 임포트 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
usedefaults	<p>목표 테이블 컬럼의 소스 컬럼이 지정되었지만 하나 이상의 행 인스턴스에 대한 데이터를 포함하지 않은 경우, 디폴트값이 로드됩니다. 누락된 데이터의 예:</p> <ul style="list-style-type: none"> <li>• DEL 파일: 임의의 수의 스페이스(" ", ",")로 구분되는 2개의 인접 컬럼 분리문자나 2개의 인접 컬럼 분리문자(",.")가 컬럼 값으로 지정됩니다.</li> <li>• DEL/ASC/WSF 파일: 행의 컬럼이 충분하지 않거나 행의 길이가 원래 스펙만큼 충분하지 않습니다.</li> </ul> <p>주: ASC 파일의 경우, NULL 컬럼 값은 명시적으로 누락된 것으로 간주되지 않으며 디폴트가 NULL 컬럼 값을 대신하지 않습니다. 숫자, 날짜, 시간 및 /시간소인 컬럼의 모든 공백 문자로 NULL 컬럼 값을 표시하거나, 컬럼이 NULL임을 표시하기 위해 모든 유형의 컬럼에 널(NULL) 표시기를 사용함으로써 NULL 컬럼 값을 표시합니다.</p> <p>이 옵션이 없는 경우, 소스 컬럼이 행 인스턴스의 데이터를 포함하지 않으면 다음 중 하나가 발생합니다.</p> <ul style="list-style-type: none"> <li>• DEL/ASC/WSF 파일: 컬럼에 널(NULL) 입력 가능한 경우 NULL이 로드됩니다. 컬럼이 널(NULL) 입력 가능하지 않으면 유틸리티가 행을 거부합니다.</li> </ul>

표 44. 임포트 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL)

수정자	설명
codepage=x	<p>x는 ASCII 문자열입니다. 해당 값은 입력 데이터 세트에서 데이터의 코드 페이지로서 해석됩니다. 임포트 조작 중 이 코드 페이지로부터 응용프로그램 코드 페이지로 문자 데이터를 변환합니다.</p> <p>다음 규칙이 적용됩니다.</p> <ul style="list-style-type: none"> <li>• 순수 DBCS(그래픽), 혼합 DBCS 및 EUC의 경우, 분리문자는 x00 - x3F의 범위로 제한됩니다.</li> <li>• nullindchar은 x20 - x7F 코드 포인트의 표준 ASCII 세트에 포함된 기호를 지정해야 합니다. 이것은 ASCII 기호 및 코드 포인트를 나타냅니다.</li> </ul> <p>주:</p> <ol style="list-style-type: none"> <li>1. codepage 수정자는 lobsinfile 수정자와 함께 사용될 수 없습니다.</li> <li>2. 응용프로그램 코드 페이지에서 데이터베이스 코드 페이지로 코드 페이지를 변환할 때 데이터 확장이 발생하면, 데이터가 절단되고 데이터 유실이 발생할 수 있습니다.</li> </ol>

표 44. 임포트 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
dateformat="x"	<p>x는 소스 파일에서 날짜의 형식입니다.<sup>2</sup> 유효한 날짜 요소는 다음과 같습니다.</p> <p>YYYY - Year(0000 - 9999 범위의 4자리 숫자)  M - Month(1 - 12 범위의 1 또는 2자리 숫자)  MM - Month(1 - 12 범위의 2자리 숫자,  M과 상호 배타적)  D - Day(1 - 31 범위의 1 또는 2자리 숫자)  DD - Day(1 - 31 범위의 2자리 숫자,  D와 상호 배타적)  DDD - Day of the year(001 - 366 범위의 3자리 숫자,  다른 day 또는 month 요소와 상호 배타적)</p> <p>지정되지 않은 각 요소에 대해 디폴트값 1이 지정됩니다. 날짜 형식의 예:</p> <p>"D-M-YYYY"  "MM.DD.YYYY"  "YYYYDDD"</p>
implieddecimal	<p>내포된 소수점의 위치는 컬럼 정의로 판별되며, 값의 끝으로 가정되지 않습니다. 예를 들어, 12345 값은 12345.00이 아닌 123.45로 DECIMAL(8,2) 컬럼에 로드됩니다.</p>
timeformat="x"	<p>x는 소스 파일에서 시간의 형식입니다.<sup>2</sup> 유효한 시간 요소는 다음과 같습니다.</p> <p>H - Hour(12시간 시스템의 경우 0 - 12 범위의  1 또는 2자리 숫자,  24시간 시스템의 경우 0 - 24)  HH - Hour(12시간 시스템의 경우  0 - 12 범위의 2자리 숫자,  24시간 시스템의 경우 0 - 24.  H와 상호 배타적)  M - Minute(0 - 59 사이의 1 또는 2자리 숫자)  MM - Minute(0 - 59 사이의 2자리 숫자,  M과 상호 배타적)  S - Second(0 - 59 사이의 1 또는 2자리 숫자)  SS - Second(0 - 59 사이의 2자리 숫자,  S와 상호 배타적)  SSSSS - Second of the day after midnight(  00000 - 86399 사이의 5자리 숫자, 다른 시간 요소와  상호 배타적)  TT - Meridian indicator(AM 또는 PM)</p> <p>지정되지 않은 각 요소에 대해 디폴트값 0이 지정됩니다. 시간 형식의 예:</p> <p>"HH:MM:SS"  "HH.MM TT"  "SSSSS"</p>

표 44. 임포트 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
timestampformat="x"	<p>x는 소스 파일에서 시간소인의 형식입니다.<sup>2</sup> 유효한 시간소인 요소는 다음과 같습니다.</p> <p>YYYY - Year(0000 - 9999 사이의 4자리 숫자)  M - Month(1 - 12 사이의 1 또는 2자리 숫자)  MM - Month(01 - 12 사이의 2자리 숫자,  M 및 MMM은 상호 배타적)  MMM - Month(월 이름으로 사용되는 3자리 문자로 대소문자를  구분하지 않으며,  M 및 MM은 상호 배타적)  D - Day(1 - 31 사이의 1 또는 2자리 숫자)  DD - Day(1 - 31 사이의 2자리 숫자, D와는 상호 배타적)  DDD - Day of the year(001 - 366 사이의 3자리 숫자,  다른 day 또는 month 요소와 상호 배타적)  H - Hour(12시간 시스템의 경우 0 - 12 범위의 1 또는 2자리 숫자,  24시간 시스템의 경우 0 - 24)  HH - Hour(12시간 시스템의 경우 0 - 12 범위의 2자리 숫자,  24시간 시스템의 경우 0 - 24,  H와는 상호 배타적)  M - Minute(0 - 59 사이의 1 또는 2자리 숫자)  MM - Minute(0 - 59 사이의 2자리 숫자,  M, minute와 상호 배타적)  S - Second(0 - 59 사이의 1 또는 2자리 숫자)  SS - Second(0 - 59 사이의 2자리 숫자,  S와 상호 배타적)  SSSSS - Second of the day after midnight(  00000 - 86399 사이의 5자리 숫자, 다른 시간 요소와  상호 배타적)  U(1 - 12회)  - Fractional seconds(U 어커런스 수는 각 자리가 0 - 9  사이의 자리 수를 나타냄)  TT - Meridian indicator(AM 또는 PM)</p> <p>디폴트값 1이 미지정된 YYYY, M, MM, D, DD 또는 DDD 요소에 지정됩니다. 디폴트값 'Jan'이 미지정된 MMM 요소에 지정됩니다. 미지정된 다른 모든 요소에 디폴트값 0이 지정됩니다. 다음은 시간소인 형식의 예입니다.</p> <p>"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>MMM 요소의 올바른 값은 다음을 포함합니다. 'jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov' 및 'dec'. 이들 값은 대소문자가 구분되지 않습니다.</p> <p>다음 예는 사용자 정의 날짜 및 시간 형식을 포함하는 데이터를 schedule이라는 테이블로 임포트하는 방법을 설명합니다.</p> <pre>db2 import from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>



표 44. импорт 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
usegraphiccodepage	<p>usegraphiccodepage가 제공되면, 그래픽 또는 2바이트 문자 대형 오브젝트(DBCLOB) 데이터 필드로 импорт되는 데이터는 그래픽 코드 페이지에 있음이 가정됩니다. 나머지 데이터는 문자 코드 페이지에 있다고 가정합니다. 그래픽 코드 페이지는 문자 코드 페이지와 연관됩니다. IMPORT는 codepage 수정자가 지정된 경우 이를 통해 문자 코드 페이지를 판별하고 codepage 수정자가 지정되지 않은 경우 응용프로그램의 코드 페이지를 통해 문자 코드 페이지를 판별합니다.</p> <p>복구 중인 테이블에 그래픽 데이터가 있는 경우에만 삭제(drop) 테이블 복구로 생성되는 구분된 데이터 파일과 결합하여 이 수정자를 사용해야 합니다.</p> <p><b>제한사항</b></p> <p>이들 파일이 오직 하나의 코드 페이지에 인코딩된 데이터를 포함하면, usegraphiccodepage 수정자는 EXPORT 유틸리티로 작성된 DEL 파일과 함께 지정되지 않아야 합니다. usegraphiccodepage 수정자는 파일의 2바이트 문자 대형 오브젝트(DBCLOB)에서 무시됩니다.</p>
xmlchar	<p>XML 문서가 문자 코드 페이지에서 인코딩됨을 지정합니다.</p> <p>이 옵션은 지정된 문자 코드 페이지에서 인코딩되었지만 인코딩 선언을 포함하지 않는 XML 문서 처리에 유용합니다.</p> <p>각 문서에서 선언 태그가 존재하고 인코딩 속성을 포함하는 경우 인코딩은 문자 코드 페이지와 일치해야 하며, 그렇지 않으면 문서를 포함하는 행이 거부됩니다. 문자 코드 페이지는 codepage 파일 유형 수정자에서 지정한 값이거나, 지정되지 않은 경우 응용프로그램 코드 페이지입니다. 디폴트로 문서가 유니코드로 인코딩되거나 인코딩 속성을 가진 선언 태그를 포함합니다.</p>
xmlgraphic	<p>XML 문서가 지정된 그래픽 코드 페이지에서 인코딩됨을 지정합니다.</p> <p>이 옵션은 특정 그래픽 코드 페이지에서 인코딩되었지만 인코딩 선언을 포함하지 않는 XML 문서 처리에 유용합니다.</p> <p>각 문서에서 선언 태그가 존재하고 인코딩 속성을 포함하는 경우 인코딩은 그래픽 코드 페이지와 일치해야 하며, 그렇지 않으면 문서를 포함하는 행이 거부됩니다. 그래픽 코드 페이지는 codepage 파일 유형 수정자에서 지정한 값의 그래픽 구성요소이거나, 지정되지 않은 경우 응용프로그램 코드 페이지의 그래픽 구성요소입니다. 디폴트로 문서가 유니코드로 인코딩되거나 인코딩 속성을 가진 선언 태그를 포함합니다.</p> <p>주: xmlgraphic 수정자가 IMPORT 명령으로 지정된 경우, импорт되는 XML 문서는 UTF-16 코드 페이지로 인코딩되어야 합니다. 그렇지 않으면, XML 문서는 구문 분석 오류로 거부되거나 데이터 손상이 있는 테이블로 импорт될 수 있습니다.</p>

표 45. импорт 유틸리티의 유효한 파일 유형 수정자: ASC(컬럼 식별자가 없는 ASCII) 파일 형식

수정자	설명
nochecklengths	<p>nochecklengths가 지정되면, 목표 테이블 컬럼의 크기를 초과하는 컬럼 정의가 소스 데이터에 있는 경우에도 각 행을 импорт하려고 시도합니다. 코드 페이지 변환으로 소스 데이터가 축소되는 경우 그러한 행은 정상적으로 импорт될 수 있습니다. 예를 들어, 소스에서 4바이트 EUC 데이터는 목표에서 2바이트 DBCS 데이터로 축소되어 반 정도의 스페이스가 필요합니다. 불일치 컬럼 정의에도 불구하고 소스 데이터가 모든 경우에 맞는 경우 특히 이 옵션이 유용합니다.</p>



표 45. 임포트 유틸리티의 유효한 파일 유형 수정자: ASC(컬럼 식별자가 없는 ASCII) 파일 형식 (계속)

수정자	설명
nullindchar=x	x는 단일 문자입니다. 널(NULL) 값을 나타내는 문자를 x로 변경합니다. x의 디폴트값은 Y입니다. <sup>3</sup>  문자가 영문자인 경우를 제외하고 EBCDIC 데이터 파일의 경우 이 수정자의 대소문자를 구분합니다. 예를 들어, 널(NULL) 표시기 문자가 N 문자가 되도록 지정되는 경우, n은 널(NULL) 표시기로 인식됩니다.
reclen=x	x는 최대값이 32,767인 정수입니다. 각 행에 대해 x 문자가 읽히지며 행의 끝을 표시하기 위한 줄 바꾸기 문자는 사용되지 않습니다.
striptblanks	데이터를 변수 길이 필드로 로드할 때 뒤 공백을 절단합니다. 이 옵션이 지정되지 않으면, 공백이 보존됩니다.  다음 예에서, striptblanks를 사용하면 임포트 유틸리티가 뒤 공백을 절단합니다.  db2 import from myfile.asc of asc modified by striptblanks method 1 (1 10, 12 15) messages msgs.txt insert into staff  이 옵션은 striptnulls와 함께 지정될 수 없습니다. 이들은 상호 배타적 옵션입니다. 이 옵션은 이전 호환성을 위해서만 지원되는 사용 안하는 t 옵션을 교체합니다.
striptnulls	데이터를 변수 길이 필드로 로드할 때 뒤 NULL 값(0x00 문자)을 절단합니다. 이 옵션이 지정되지 않으면, NULL 값이 보존됩니다.  이 옵션은 striptblanks와 함께 지정될 수 없습니다. 이들은 상호 배타적 옵션입니다. 이 옵션은 이전 호환성을 위해서만 지원되는 사용 안하는 padwithzero 옵션을 교체합니다.

표 46. 임포트 유틸리티의 유효한 파일 유형 수정자: DEL(컬럼 식별자가 있는 ASCII) 파일 형식

수정자	설명
chardelx	x는 단일 문자열 분리문자입니다. 디폴트값은 큰따옴표(")입니다. 큰따옴표 대신 지정된 문자를 사용하여 문자열을 묶습니다. <sup>34</sup> 명시적으로 큰따옴표를 문자열 분리문자로 지정하려는 경우 다음과 같이 지정해야 합니다.  modified by charde1"  작은따옴표(')를 문자열 분리문자로 지정할 수도 있습니다. 다음 예에서, charde1''는 임포트 유틸리티가 작은따옴표를 문자열 분리문자로 해석하게 합니다.  db2 "import from myfile.del of del modified by charde1'' method p (1, 4) insert into staff (id, years)"
coldelx	x는 단일 문자 컬럼 분리문자입니다. 디폴트값은 쉼표(.)입니다. 지정된 문자가 쉼표 대신 사용되어 컬럼 끝을 나타냅니다. <sup>34</sup>  다음 예에서, colde1;은 임포트 유틸리티가 세미콜론을 컬럼 분리문자로 해석하게 합니다.  db2 import from myfile.del of del modified by colde1; messages msgs.txt insert into staff
decplusblank	플러스 부호 문자. 플러스 부호(+) 대신 공백을 양의 10진수 값 앞에 붙입니다. 디폴트 조치는 플러스 부호를 양의 10진수 값 앞에 붙이는 것입니다.

표 46. 임포트 유틸리티의 유효한 파일 유형 수정자: DEL(컬럼 식별자가 있는 ASCII) 파일 형식 (계속)

수정자	설명
decptx	<p>x는 소수점 문자로 마침표를 대신하는 단일 문자입니다. 디폴트값은 마침표(.)입니다. 마침표 대신 지정된 문자가 소수점 문자로 사용됩니다.<sup>34</sup></p> <p>다음 예에서, decpt;는 임포트 유틸리티가 세미콜론(; )을 소수점으로 해석하게 합니다.</p> <pre>db2 "import from myfile.del of del       modified by chardel'"       decpt; messages msgs.txt insert into staff"</pre>
delprioritychar	<p>분리문자의 현재 디폴트 우선순위: 레코드 구분 문자, 문자 분리문자, 컬럼 분리문자. 이 수정자는 분리문자 우선순위를 문자 분리문자, 레코드 구분 문자, 컬럼 분리문자로 되돌림으로써 이전 우선순위에 따른 기존 응용프로그램을 보호합니다. 구분:</p> <pre>db2 import ... modified by delprioritychar ...</pre> <p>예를 들어, 다음과 같은 DEL 데이터 파일이 있습니다.</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 수정자를 지정하고, 이 데이터 파일에는 두 개의 행만이 있습니다. 두 번째 &lt;행 분리문자&gt;는 두 번째 행의 첫 번째 데이터 컬럼의 파트로 해석되지만, 첫 번째 및 세 번째 &lt;행 분리문자&gt;는 실제 레코드 구분 문자로 해석됩니다. 이 수정자가 지정되지 않은 경우, 이 데이터 파일에는 세 개의 행이 있으며, 각 행은 &lt;행 분리문자&gt;로 구분됩니다.</p>
keepblanks	<p>유형 CHAR, VARCHAR, LONG VARCHAR 또는 CLOB의 각 필드에 앞뒤 공백을 둡니다. 이 옵션이 없으면, 문자 분리문자 내에 없는 앞 공백과 뒤 공백은 모두 제거되며 공백 필드의 테이블로 NULL이 삽입됩니다.</p>
nochardel	<p>임포트 유틸리티는 컬럼 분리문자 간의 모든 바이트를 컬럼 데이터의 파트로 가정합니다. 문자 분리문자는 컬럼 데이터의 파트로 구분 분석됩니다. DB2를 사용하여 데이터를 익스포트한 경우(익스포트 시 nochardel을 지정한 경우를 제외하고) 이 옵션을 지정하지 말아야 합니다. 이 옵션은 문자 분리문자가 없는 벤더 데이터 파일을 지원하기 위해 제공됩니다. 부적절한 사용은 데이터 유실이나 손상을 초래할 수 있습니다.</p> <p>이 옵션은 chardelx, delprioritychar 또는 nodoubledel과 함께 지정될 수 없습니다. 이들은 상호 배타적 옵션입니다.</p>
nodoubledel	<p>2바이트 분리문자를 인식하지 않습니다.</p>

표 47. 임포트 유틸리티의 유효한 파일 유형 수정자: IXF 파일 형식

수정자	설명
forcein	<p>코드 페이지 불일치와 관계없이 데이터를 승인하고 코드 페이지 간에 변환하지 않도록 유틸리티에 지시합니다.</p> <p>데이터의 고정 길이 대상 필드가 충분히 큰지 검증하도록 고정 길이 대상 필드를 점검합니다. nochecklengths가 지정된 경우, 검사가 수행되지 않으며 각 행을 임포트하려고 시도합니다.</p>
indexixf	<p>기존 테이블에 현재 정의된 모든 인덱스를 삭제(drop)하고 PC/IXF 파일의 인덱스 정의에서 새로운 인덱스를 작성하도록 유틸리티에 지시합니다. 이 옵션은 목차가 교체될 때에만 사용될 수 있습니다. 뷰 또는 insert-column와 함께 사용될 수 없습니다.</p>
indexschema=schema	<p>색인 작성 중 인덱스 이름에 지정된 schema를 사용합니다. schema가 지정되지 않은 경우(그러나 키워드 indexschema가 지정됨), 연결 사용자 ID를 사용합니다. 해당 키워드가 지정되지 않은 경우, IXF 파일에서 스키마를 사용합니다.</p>

표 47. импорт 유틸리티의 유효한 파일 유형 수정자: IXF 파일 형식 (계속)

수정자	설명
nochecklengths	nochecklengths가 지정되면, 목표 테이블 컬럼의 크기를 초과하는 컬럼 정의가 소스 데이터에 있는 경우에도 각 행을 импорт하려고 시도합니다. 코드 페이지 변환으로 소스 데이터가 축소되는 경우 그러한 행은 정상적으로 импорт될 수 있습니다. 예를 들어, 소스에서 4바이트 EUC 데이터는 목표에서 2바이트 DBCS 데이터로 축소되어 반 정도의 스페이스가 필요합니다. 불일치 컬럼 정의에도 불구하고 소스 데이터가 모든 경우에 맞는 경우 특히 이 옵션이 유용합니다.
forcecreate	인포트 조작 중 SQL3311N 리턴 후에 제한된 정보나 가능한 누락 정보로 테이블이 작성되어야 함을 지정합니다.

표 48. codepage 및 usegraphiccodepage 사용 시 IMPORT 동작

codepage=N	usegraphiccodepage	IMPORT 동작
Absent	Absent	파일의 모든 데이터는 응용프로그램 코드 페이지에 있는 것으로 가정됩니다.
Present	Absent	파일의 모든 데이터는 코드 페이지 N에 있는 것으로 가정됩니다.  경고: N이 1바이트 코드 페이지이면 데이터베이스로 импорт될 때 그래픽 데이터가 손상됩니다.
Absent	Present	파일의 문자 데이터는 응용프로그램 코드 페이지에 있는 것으로 가정됩니다. 그래픽 데이터는 응용프로그램 그래픽 데이터의 코드 페이지에 있는 것으로 가정됩니다.  응용프로그램 코드 페이지가 1바이트이면, 모든 데이터는 응용프로그램 코드 페이지에 있는 것으로 가정됩니다.  경고: 응용프로그램 코드 페이지가 1바이트이면, 데이터베이스가 그래픽 컬럼을 포함해도 데이터베이스로 импорт될 때 그래픽 데이터가 손상됩니다.
Present	Present	문자 데이터는 코드 페이지 N에 있는 것으로 가정됩니다. 그래픽 데이터는 N의 그래픽 코드 페이지에 있는 것으로 가정됩니다.  N이 1바이트 또는 2바이트 코드 페이지인 경우, 모든 데이터는 코드 페이지 N에 있는 것으로 가정됩니다.  경고: N이 1바이트 코드 페이지이면 데이터베이스로 импорт될 때 그래픽 데이터가 손상됩니다.

주:

1. **MODIFIED BY** 옵션과 함께 지원되지 않는 파일 유형을 사용하려고 시도하는 경우 импорт 유틸리티는 경고를 발행하지 않습니다. 이런 경우, импорт 조작에 실패하며 오류 코드가 리턴됩니다.
2. 날짜 출력 문자열을 둘러싼 큰따옴표는 필수입니다. 필드 구분자는 a - z, A - Z 및 0 - 9를 포함할 수 없습니다. 필드 구분자는 DEL 파일 형식의 필드 분리 문자나 문자 분리문자와 같지 않아야 합니다. 요소의 시작 및 종료 위치가 명확한 경우 필드 구분자가 선택적입니다. D, H, M 또는 S와 같은 요소가 사용되는 경우(수정자에 따라) 항목의 변수 길이 때문에 모호함이 있을 수 있습니다.

시간소인 형식의 경우, 문자 M을 사용하는 month 및 minute 디스크립터 간의 모호함을 피하려면 주의해야 합니다. month 필드는 다른 날짜 필드와 인접해야 합니다. minute 필드는 다른 시간 필드와 인접해야 합니다. 다음은 앰비규어스 시간소인 형식입니다.

```
"M"(month 또는 minute일 수 있음)
"M:M"(month 및 minute 구분 가능?)
"M:YYYY:M"(둘 다 month로 해석됨)
"S:M:YYYY"(둘 다 시간 값 및 날짜 값에 인접)
```

앰비규어스 경우, 유틸리티는 오류 메시지를 발행하며, 조작에 실패합니다.

다음은 명확한 시간소인 형식입니다.

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

큰따옴표 및 백슬래시와 같은 일부 문자는 Escape 문자(예: #)가 앞에 와야 합니다.

3. chardel, coldel 또는 decpt 파일 유형에 제공되는 문자 값은 소스 데이터의 코드 페이지에 지정되어야 합니다.

문자 코드 포인트(문자 기호 대신)는 구문 xJJ 또는 0xJJ를 사용하여 지정될 수 있으며, 여기서 JJ는 코드 포인트의 16진수를 나타냅니다. 예를 들어, 컬럼 분리문자로 # 문자를 지정하려면, 다음 중 하나를 사용하십시오.

```
... modified by coldel# ...
... modified by coldel0x23 ...
... modified by coldelX23 ...
```

4. 데이터 이동을 위한 분리문자 고려사항은 분리문자를 겹쳐쓰기할 때 사용할 수 있는 문자에 적용하는 제한사항을 나열합니다.
5. 별칭으로 импорт할 때 다음 파일 유형 수정자는 허용되지 않습니다.

- indexixf
- indexschema
- dldel filetype
- nodefaults
- usedefaults
- no\_type\_id filetype
- generatedignore
- generatedmissing
- identityignore
- identitymissing

- lobsinfile
6. **WSF** 파일 형식은 XML 컬럼에서 지원되지 않습니다. 또한 이 파일 형식에 대한 지원은 사용되지 않으므로 추후 릴리스에서 제거될 수 있습니다. 지원이 제거되기 전에 WSF 파일 대신 지원되는 파일 형식을 사용하여 시작하는 것이 좋습니다.
  7. **CREATE** 모드는 XML 컬럼에서 지원되지 않습니다.
  8. 모든 XML 데이터는 주 데이터 파일에서 분리된 XML 파일에 있어야 합니다. 주 데이터 파일의 각 XML 컬럼에 대해 XDS(XML Data Specifier)(또는 NULL 값)가 있어야 합니다.
  9. XMLCHAR 또는 XMLGRAPHIC 파일 유형 수정자가 지정된 경우를 제외하고, XML 문서는 유니코드 형식으로 되어 있거나 인코딩 속성을 포함하는 선언 태그를 포함한다고 가정합니다.
  10. 잘 양식화되지 않은 문서를 포함하는 행은 거부됩니다.
  11. **XMLVALIDATE** 옵션이 지정된 경우, 일치하는 스키마에 대해 정상적으로 유효성을 확인하는 문서는 삽입될 때 스키마 정보로 주석을 표시합니다. 일치하는 스키마에 대해 유효성을 확인하는 데 실패하는 문서를 포함하는 행은 거부됩니다. 정상적으로 유효성을 확인하려면, 임포트를 호출하는 사용자가 보유한 특권이 최소한 다음 중 하나를 포함해야 합니다.
    - DBADM 권한
    - 유효성 확인에 사용되는 XML 스키마의 USAGE 특권
  12. 내재적으로 숨겨진 행 변경 시간소인 컬럼을 포함하는 테이블로 임포트할 때, 내재적으로 숨겨진 컬럼의 등록 정보는 무시됩니다. 그러므로, 컬럼의 데이터가 임포트되는 데이터에 표시되지 않으며 명시적 컬럼 목록이 표시되지 않으면 rowchangetimestampmissing 파일 유형 수정자가 임포트 명령에 지정되어야 합니다.

## ADMIN\_CMD 프로시저를 사용하는 INITIALIZE TAPE 명령

백업 및 리스토어 작업을 위한 테이프를 스트리밍 테이프 디바이스로 초기화합니다. 이 명령은 Windows 운영 체제에서만 지원됩니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

데이터베이스

## 명령 구문

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

### 명령 매개변수

#### ON *device*

유효한 테이프 디바이스 이름을 지정합니다. 디폴트값은 `WWW.WTAPE0`입니다. 지정된 디바이스는 서버에 관계되어야 합니다.

#### USING *blksize*

디바이스에 대한 블록 크기를 바이트 단위로 지정합니다. 지정된 블록 크기 값이 디바이스에 대한 블록 크기의 지원 범위에 해당될 경우 디바이스는 이 지정된 블록 크기를 사용하도록 초기화됩니다.

BACKUP DATABASE 명령 및 RESTORE DATABASE에 대해 지정된 버퍼 크기는 여기에 지정된 블록 크기로 나눌 수 있어야 합니다.

이 매개변수의 값을 지정하지 않으면 디바이스는 해당 디폴트 블록 크기를 사용하도록 초기화됩니다. 0값을 지정하면 디바이스는 가변 길이 블록 크기를 사용하도록 초기화되고, 디바이스가 가변 길이 블록 코드를 지원하지 않을 경우 오류가 리턴됩니다.

테이프에 백업시 가변 블록 크기의 사용은 현재 지원되지 않습니다. 이 옵션을 사용해야 하는 경우, 가변 블록 크기로 작성된 백업 이미지를 사용하여 성공적으로 복구할 수 있는 제대로 테스트된 프로시저가 있는지 확인하십시오.

가변 블록 크기 사용시 사용 중인 테이프 디바이스에 대해 최대 한계 이하의 백업 버퍼 크기를 지정해야 합니다. 최적 성능을 위해서는 버퍼 크기를 사용 중인 디바이스의 최대 블록 크기 한계와 동일하게 지정해야 합니다.

#### 예 :

2048바이트의 블록 크기가 디바이스에 대한 블록 크기의 지원 범위에 해당될 경우 테이프 디바이스를 이 값을 사용하도록 초기화합니다.

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

### 사용 시 참고사항

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

## ADMIN\_CMD 프로시저를 사용하는 LOAD 명령

DB2 테이블에 데이터를 로드합니다. 서버에 있는 데이터는 파일, 테이프 또는 Named Pipe 양식이 될 수 있습니다. 또한 데이터는 현재 연결된 데이터베이스나 동일 인스턴

스의 다른 데이터베이스에 대해 실행 중인 쿼리에 정의된 커서에서 로드되거나 사용자가 작성한 스크립트나 응용프로그램을 사용하여 로드될 수 있습니다. 테이블의 COMPRESS 속성이 YES로 설정된 경우, 테이블의 XML 스토리지 오브젝트의 데이터를 포함하여 이미 사전이 테이블에 존재하는 모든 데이터 및 데이터베이스 파티션에서 로드된 데이터가 압축됩니다.

143 페이지의 『로드 유틸리티의 파일 유형 수정자』로의 빠른 링크.

## 제한사항

로드 유틸리티는 계층 구조 레벨에서 데이터 로드를 지원하지 않습니다. 로드 유틸리티는 범위로 클러스터된 테이블과 호환 가능하지 않습니다.

## 범위

이 명령은 단일 요청으로 다중 데이터베이스 파티션에 대해 실행할 수 있습니다.

## 권한 부여

다음 중 하나가 필요합니다.

- *dataaccess*
- 데이터베이스의 LOAD 권한 및
  - 로드 유틸리티가 INSERT 모드, TERMINATE 모드 (이전 로드 삽입 조작을 종료하기 위해) 또는 RESTART 모드(이전 로드 삽입 조작을 재시작하기 위해)에서 호출될 때 테이블에 대한 INSERT 특권
  - 로드 유틸리티가 REPLACE 모드, TERMINATE 모드(이전 로드 삽입 조작을 종료하기 위해) 또는 RESTART 모드(이전 로드 삽입 조작을 재시작하기 위해)에서 호출될 때 테이블에 대한 INSERT 및 DELETE 특권
  - 이 테이블이 로드 조작의 일부로 사용되는 경우 예외 테이블에 대한 INSERT 특권.
- 보호 설정된 컬럼이 있는 테이블에 데이터를 로드하려면 세션 권한 부여 ID에 테이블의 모든 보호 컬럼에 대해 쓰기 액세스를 허용하는 LBAC 증명서가 있어야 합니다. 그렇지 않으면 로드에 실패하고 오류(SQLSTATE 5U014)가 리턴됩니다.
- 보호 설정된 행이 있는 테이블에 데이터를 로드하려면 세션 권한 부여 ID는 다음 기준에 부합되는 보안 레이블을 보유해야 합니다.
  - 테이블을 보호하는 보안 규정에 포함됨
  - 세션 권한 부여 ID에 쓰기 액세스 또는 모든 액세스 권한이 부여되었습니다.

세션 권한 부여 ID에 이러한 보안 레이블이 없으면 로드에 실패하고 오류(SQLSTATE 5U014)가 리턴됩니다. 이 보안 레벨은 세션 권한 부여 ID의 LBAC 증명서가 데이터의 해당 행을 보호하는 보안 레벨에 쓰기를 허용하지 않는 경우 로드된 행을 보호하는 데 사용됩니다. 그러나 테이블을 보호하는 보안 규정이 CREATE SECURITY



POLICY 명령문의 RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL 옵션을 사용하여 작성되는 경우에는 발생하지 않습니다. 이러한 경우, 로드 실패하고 오류(SQLSTATE 42519)가 리턴됩니다.

- REPLACE 옵션이 지정되어 있으면 세션 권한 부여 ID에 테이블을 삭제할 수 있는 권한이 있어야 합니다.
- LOCK WITH FORCE 옵션이 지정된 경우, SYSADM 권한이 필요합니다.

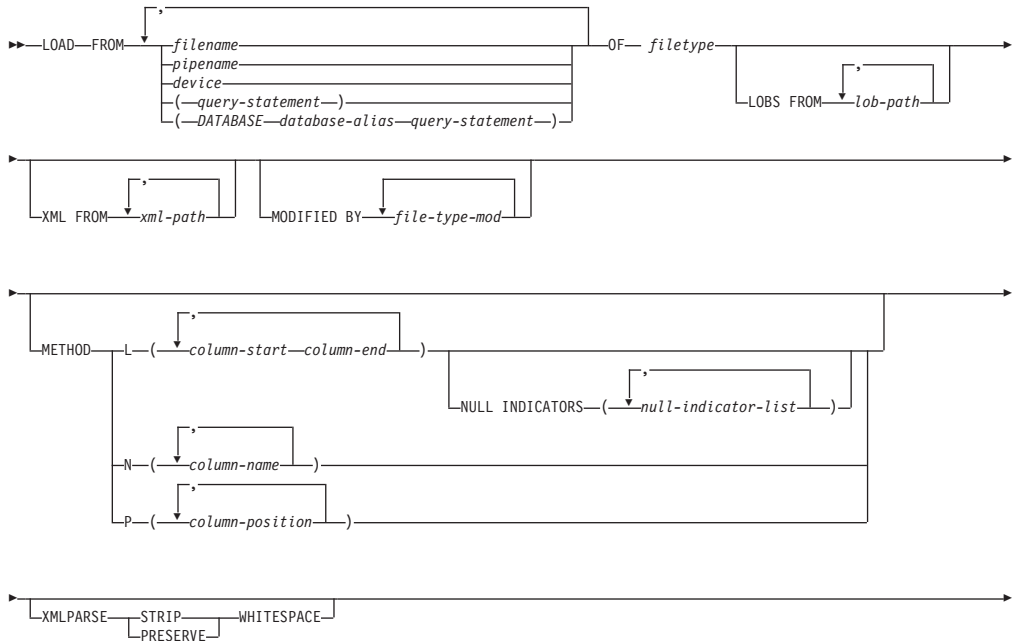
모든 로드 프로세스(및 일반적으로 모든 DB2 서버 프로세스)는 인스턴스 소유자가 소유하고 이들 모든 프로세스는 필요한 파일에 액세스하기 위해 인스턴스 소유자의 식별을 사용하므로 인스턴스 소유자는 입력 데이터 파일에 대한 읽기 액세스가 있어야 합니다. 이러한 입력 데이터는 명령 호출자와 상관없이 인스턴스 소유자가 읽을 수 있어야 합니다.

### 필수 연결

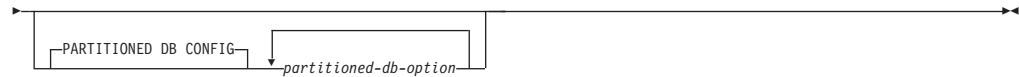
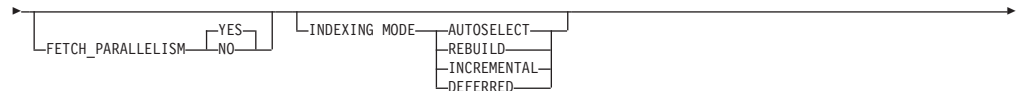
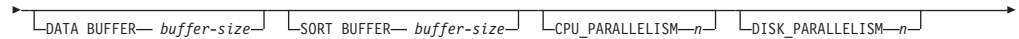
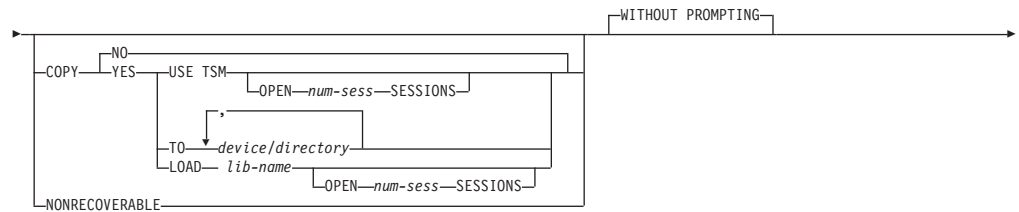
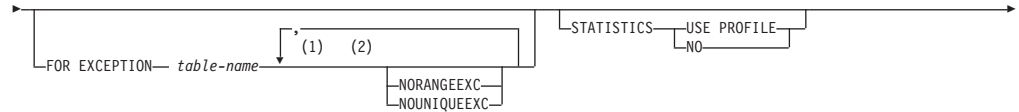
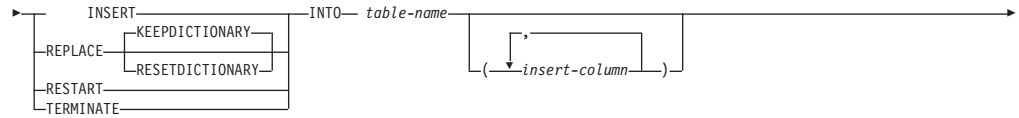
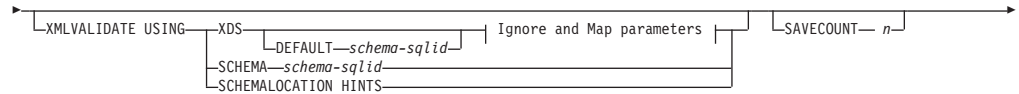
데이터베이스.

인스턴스. 명시적 접속은 필요하지 않습니다. 데이터베이스에 대한 연결이 설정되면 로컬 인스턴스에 대한 내재적 접속이 시도됩니다.

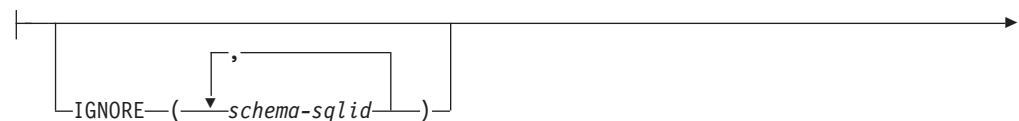
### 명령 구문

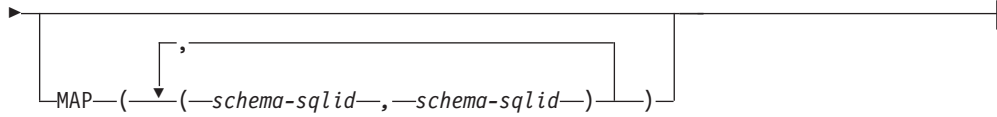




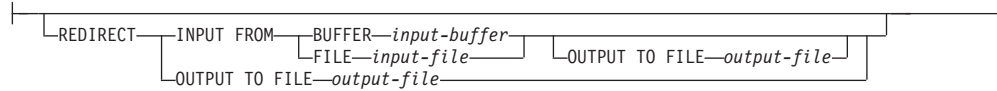


**Ignore 및 Map 매개변수:**





### Redirect Input/Output 매개변수:



### 주:

- 1 이 키워드는 임의의 순서로 표시될 수 있습니다.
- 2 이들 각 키워드는 한 번만 표시될 수 있습니다.

### 명령 매개변수

**FROM** filename | pipename | device (query-statement) |

(DATABASE database-alias query-statement)

로드 중인 데이터를 포함하는 SQL문을 참조하는 파일, 파이프 또는 디바이스를 지정하거나 커서에서 로드하기 위한 선택적 소스 데이터베이스 및 SQL문 자체를 지정합니다.

query-statement 옵션은 커서에서 로드하는 데 사용됩니다. 이 옵션에는 한 개의 쿼리 명령문만 들어 있으며 괄호 안에 표시되어야 하며 VALUES, SELECT 또는 WITH로 시작할 수 있습니다. 예를 들면, 다음과 같습니다.

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

DATABASE database-alias 절이 괄호 안에서 쿼리 명령문보다 앞에 들어 있으면 LOAD 명령이 query-statement를 사용하여 서버에 정의된 database-alias 이름으로 표시된 대로 해당 데이터베이스에서 데이터 로드를 시도합니다. 반드시 서버에 존재하며 응용프로그램이 현재 연결되어 있는 다른 데이터베이스를 가리켜야 합니다. LOAD는 현재 연결된 데이터베이스에 대해 명확하게 제공된 사용자 ID 및 암호를 사용하여 실행되어야 합니다(내재적 연결에서는 LOAD에 실패합니다.)

입력 소스가 파일, 파이프 또는 디바이스인 경우 서버의 코디네이터 파티션에서 액세스할 수 있어야 합니다.

이름이 여러 개 지정되면 순서대로 처리됩니다. 마지막으로 지정된 항목이 테이프 디바이스이고 사용자가 테이프를 입력하도록 프롬프트되면 LOAD에 실패하고 ADMIN\_CMD 프로시저가 오류를 리턴합니다.

### 주:

1. 완전한 경로의 파일 이름을 사용해야 하며 서버에 존재해야 합니다.

2. *ADMIN\_CMD* 프로시저를 사용하는 *EXPORT* 명령을 사용하여 파일로 데이터를 익스포트하는 경우, 데이터 파일은 분리 사용자 ID 소유가 됩니다. 일반적으로 이 파일은 인스턴스 소유자가 액세스할 수 없습니다. *CLP* 또는 *ADMIN\_CMD* 프로시저에서 *LOAD*를 실행시키려면 데이터 파일이 인스턴스 소유자 ID에 의해 액세스 가능해야 하며 따라서 데이터 파일에 대한 읽기 액세스 권한이 인스턴스 소유자에게 부여되어야 합니다.
3. 파일이 실제로는 분리되어 있지만 논리적으로는 한 개인 경우, 다중 *IXF* 파일로부터의 데이터 로드가 지원됩니다. 파일이 논리적 및 물리적으로 분리된 경우에는 지원되지 않습니다. (다중의 실제 파일도 *EXPORT* 명령을 한번 호출하여 모두 작성된 경우 논리적으로는 한 개로 간주됩니다.)
4. 파티션된 데이터베이스 환경에서, *XML* 데이터를 파일에서 테이블로 로드하는 경우에는 로드를 실행 중인 모든 데이터베이스 파티션에서 *XML* 데이터 파일에 읽기 액세스가 가능해야 합니다.

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

다음과 같은 데이터 형식을 지정합니다.

- *ASC*(컬럼 식별자가 없는 *ASCII* 형식)
- *DEL*(컬럼 식별자가 있는 *ASCII* 형식)
- *IXF*(*Integration Exchange Format*, *PC* 버전)는 *DB2*에 의해 독점 사용되는 2진 형식입니다.
- *CURSOR*(*SELECT* 또는 *VALUES*문에 대해 선언된 커서).

주: 분산 데이터베이스 환경에서, *CURSOR* 파일 유형을 사용하여 *XML* 데이터를 테이블로 로드하는 경우에는 *PARTITION\_ONLY* 및 *LOAD\_ONLY* 모드가 지원되지 않습니다.

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

로드할 *LOB* 값이 포함된 데이터 파일의 경로. 경로는 슬래시(/)로 끝나야 합니다. 경로가 완전해야 하며 서버의 코디네이터 파티션에서 액세스할 수 있어야 합니다. *LOB* 데이터 파일의 이름은 기본 데이터 파일(*ASC*, *DEL* 또는 *IXF*)에 저장되는데 *LOB* 컬럼으로 로드될 컬럼에 저장됩니다. 지정할 수 있는 최대 경로 수는 999입니다. 이는 내재적으로 *LOBSINFILE* 동작을 활성화합니다.

이 옵션은 *CURSOR* 파일 유형과 함께 지정되면 무시됩니다.

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

파일 유형 수정자 옵션을 지정합니다. 143 페이지의 『로드 유틸리티의 파일 유형 수정자』의 내용을 참조하십시오.

#### **METHOD**

**L** 데이터를 로드할 시작 및 끝 컬럼을 지정합니다. 컬럼 번호는 데이터 행

이 시작되는 바이트 오프셋입니다. 컬럼 번호는 1부터 시작됩니다. 이 메소드는 ASC 파일에서만 사용될 수 있으며 해당 파일 유형에 유일한 유효 메소드입니다.

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

이 옵션은 METHOD L 매개변수가 지정된 경우(입력 파일이 ASC 파일)에만 사용할 수 있습니다. 널(NULL) 표시기 목록은 각 널(NULL) 표시기 필드의 컬럼 번호를 지정하는 쉼표로 구분된 양의 정수 목록입니다. 컬럼 번호는 데이터 행이 시작되는 널(NULL) 표시기 필드의 바이트 오프셋입니다. METHOD L 매개변수에 정의된 각 데이터 필드에 대해 한 개의 항목이 널(NULL) 표시기 목록에 있어야 합니다. 컬럼 번호 0은 해당 데이터 필드에 데이터가 항상 들어 있음을 나타냅니다.

널(NULL) 표시기의 Y 값은 컬럼 데이터를 널(NULL)로 지정합니다. 널(NULL) 표시기 컬럼에서 Y 이외의 모든 문자는 컬럼 데이터가 널(NULL)이 아니며 METHOD L 옵션이 지정하는 컬럼 데이터가 로드됨을 지정합니다.

널(NULL) 표시기 문자는 MODIFIED BY 옵션을 사용하여 변경할 수 있습니다.

**N** 로드할 데이터 파일의 컬럼 이름을 지정합니다. 이 컬럼 이름의 대소문자는 시스템 카탈로그의 해당 이름의 대소문자와 일치해야 합니다. 널(NULL) 입력이 가능하지 않은 각 테이블 컬럼에는 METHOD N 목록에 해당 항목이 있어야 합니다. 예를 들면, 주어진 데이터 필드 F1, F2, F3, F4, F5 및 F6과 테이블 컬럼 C1 INT, C2 INT NOT NULL, C3 INT NOT NULL 및 C4 INT, method N (F2, F1, F4, F3)은 유효한 요청인 반면, method N (F2, F1)은 유효하지 않습니다. 이 메소드는 파일 유형이 IXF 또는 CURSOR인 경우에만 사용될 수 있습니다.

**P** 로드할 입력 데이터 필드의 필드 번호(1부터 번호를 지정)를 지정합니다. 널(NULL) 입력이 가능하지 않은 각 테이블 컬럼에는 METHOD P 목록에 해당 항목이 있어야 합니다. 예를 들면, 주어진 데이터 필드 F1, F2, F3, F4, F5 및 F6 그리고 테이블 컬럼 C1 INT, C2 INT NOT NULL, C3 INT NOT NULL 및 C4 INT, method P (2, 1, 4, 3)은 유효한 요청인 반면, method P (2, 1)은 유효하지 않습니다. 이 메소드는 파일 유형이 IXF, DEL 또는 CURSOR인 경우에만 사용할 수 있으며 DEL 파일 유형에 유일한 유효 메소드입니다.

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

XML 파일이 들어 있는 하나 이상의 경로를 지정합니다. XDS는 주 데이터 파일(ASC, DEL 또는 IXF)에 포함되며 XML 컬럼으로 로드될 컬럼에 있습니다.

### **XMLPARSE**

XML 문서가 구문 분석되는 방법을 지정합니다. 이 옵션을 지정하지 않을 경우, XML 문서에 대한 구문 분석 동작은 CURRENT XMLPARSE OPTION 특수 레지스터의 값으로 판별됩니다.

### **STRIP WHITESPACE**

XML 문서가 구문 분석될 때 공백을 제거하도록 지정합니다.

### **PRESERVE WHITESPACE**

XML 문서가 구문 분석될 때 공백을 제거하지 않도록 지정합니다.

### **XMLVALIDATE**

XML 문서가 스키마에 대해 유효성이 확인되도록 지정합니다(해당되는 경우).

### **USING XDS**

기본 데이터 파일의 XDS(XML Data Specifier)에 의해 식별된 XML 스키마에 대해 XML 문서의 유효성을 확인합니다. USING XDS와 함께 XMLVALIDATE 옵션을 호출한 경우, 유효성 확인을 수행하는 데 사용된 스키마는 디폴트로 XDS의 SCH 속성에 의해 판별됩니다. SCH 속성이 XDS에 존재하지 않을 경우, DEFAULT 절로 디폴트 스키마를 지정하지 않으면 스키마 유효성 확인이 발생하지 않습니다.

DEFAULT, IGNORE 및 MAP 절은 스키마 판별 동작을 수정하는 데 사용될 수 있습니다. 이들 세 개의 선택적 절은 XDS의 권장 스펙에 직접적으로 적용되며 서로에게는 적용되지 않습니다. 예를 들어, 한 스키마가 DEFAULT 절에서 지정되어 선택되었으면 이 스키마는 IGNORE 절에 지정되어도 무시되지 않습니다. 마찬가지로 한 스키마가 MAP 절에서 첫 번째 파트 쌍으로 지정되어 선택되면 다른 MAP 절 쌍의 두 번째 파트에 지정되어도 다시 맵핑되지 않습니다.

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

XML 문서가 지정된 SQL ID가 있는 XML 스키마에 대해 유효성이 확인됩니다. 이 경우 모든 XML 컬럼에 대해 XDS(XML Data Specifier)의 SCH 속성이 무시됩니다.

### **USING SCHEMALOCATION HINTS**

XML 문서가 소스 XML 문서의 XML 스키마 위치 힌트에 의해 식별된 스키마에 대해 유효성이 확인됩니다. XML 문서에서 schemaLocation 속성을 찾을 수 없으면 유효성 확인이 발생하지 않습니다.

니다. USING SCHEMALOCATION HINTS 절을 지정하면 모든 XML 컬럼에 대해 XDS(XML Data Specifier)의 SCH 속성이 무시됩니다.

아래의 XMLVALIDATE 옵션 예를 참조하십시오.

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

이 옵션은 USING XDS 매개변수가 지정된 경우에만 사용할 수 있습니다. SCH 속성에 의해 식별될 경우 IGNORE 절은 무시할 하나 이상의 스키마 목록을 지정합니다. 로드된 XML 문서에 대한 XDS(XML Data Specifier)에 SCH 속성이 존재하고 SCH 속성에 의해 식별된 스키마가 IGNORE할 스키마 목록에 포함된 경우, 이 로드된 XML 문서에 대해서는 스키마 유효성 확인이 발생하지 않습니다.

주:

IGNORE 절에 스키마가 지정되어 있으면 이 스키마는 또한 MAP 절에 있는 스키마 쌍의 왼쪽에 존재할 수 없습니다.

IGNORE 절은 XDS에만 적용됩니다. MAP 절에 의해 맵핑된 스키마는 IGNORE 절에 의해 지정된 경우 계속 무시되지 않습니다.

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

이 옵션은 USING XDS 매개변수가 지정된 경우에만 사용할 수 있습니다. DEFAULT 절을 통해 지정된 스키마는 로드된 XML 문서의 XDS(XML Data Specifier)가 XML 스키마를 식별하는 SCH 속성을 포함하지 않을 때 유효성 확인에 사용할 스키마를 식별합니다.

DEFAULT 절은 IGNORE 및 MAP 절보다 우선순위를 갖습니다. XDS가 DEFAULT 절을 충족시키면 IGNORE 및 MAP 스펙은 무시됩니다.

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

이 옵션은 USING XDS 매개변수가 지정된 경우에만 사용할 수 있습니다. 로드된 각 XML 문서에 대한 XDS(XML Data Specifier)의 SCH 속성이 지정하는 스키마 대신 사용할 대체 스키마를 지정하려면 이 MAP 절을 사용하십시오. MAP 절은 하나 이상의 스키마 쌍 목록을 지정하며, 여기서 각 쌍은 한 스키마 대 다른 스키마의 맵핑을 나타냅니다. 쌍에서 첫 번째 스키마는 XDS에 있는 SCH 속성에 의해 참조되는 스키마를 나타냅니다. 쌍에서 두 번째 스키마는 스키마 유효성 확인을 수행하는 데 사용되어야 하는 스키마를 나타냅니다.

MAP 절에 있는 스키마 쌍의 왼쪽에 있는 스키마는 IGNORE 절에 지정될 수 없습니다.

스키마 쌍 맵핑이 적용된 후, 최종 결과가 됩니다. 맵핑 조작은 전이되지 않으므로 선택된 스키마는 다른 스키마 상 맵핑에 계속 적용되지 않습니다.

스키마가 두 번 이상 맵핑될 수 없다는 것은 쌍의 왼쪽에 두 번 이상 나타날 수 없다는 것을 의미합니다.

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

로드 유틸리티가 *N*개의 행 뒤에 매번 일관성 지점을 설정하도록 지정합니다. 이 값은 쪽 수로 변환되며 Extent 크기의 간격으로 반올림됩니다. 메시지는 각각의 일관성 지점에서 발행되므로 LOAD QUERY를 사용하여 로드 조작을 모니터링하는 경우 이 옵션을 선택해야 합니다. *N* 값이 충분히 높지 않으면 각 일관성 지점에서 수행된 활동의 동기화가 성능에 영향을 줍니다.

디폴트값은 0으로, 필요하지 않으면 일관성 지점을 설정하지 않음을 의미합니다.

CURSOR 파일 유형과 함께 지정되거나 XML 컬럼을 포함하는 테이블 로드 시 이 옵션은 무시됩니다.

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

로드될 파일에 있는 *N*개의 실제 레코드 수를 지정합니다. 사용자가 첫 번째 *N* 개 행만 로드할 수 있도록 합니다.

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

*N*번의 경고 후에 로드 조작을 중지합니다. 경고가 예상되지는 않지만 올바른 파일 및 테이블이 사용되고 있는지에 대한 검증을 원할 경우 이 매개변수를 설정하십시오. 로드 파일이나 목표 테이블이 잘못 지정된 경우, 로드 유틸리티가 로드를 시도하는 각 행에 대해 경고를 생성하므로 로드 실패하게 됩니다. *N* 이 0이거나 이 옵션을 지정하지 않으면 발행된 경고 수에 관계없이 로드 조작을 계속합니다. 경고의 임계값에 도달하여 로드 조작이 중지된 경우, 다른 로드 조작이 RESTART 모드로 시작될 수 있습니다. 로드 조작은 마지막 일관성 지점에서 자동으로 계속됩니다. 또는 다른 로드 조작을 REPLACE 모드로 초기화하여 입력 파일의 처음부터 시작하게 할 수 있습니다.

#### **MESSAGES ON SERVER**

LOAD 명령으로 서버에 작성한 메시지 파일을 저장하도록 지정합니다. 리턴되는 결과 세트에는 이 조작 중에 발생하는 경고 및 오류 메시지를 모두 검색하는데 필요한 SQL문 MSG\_RETRIEVAL 컬럼과 메시지를 정리하는데 필요한 SQL문인 MSG\_REMOVAL 컬럼이 필요합니다.

이 절을 지정하지 않으면 ADMIN\_CMD 프로시저가 호출자로 리턴될 때 메시지 파일이 삭제됩니다. 결과 세트의 MSG\_RETRIEVAL 및 MSG\_REMOVAL 컬럼에 널(NULL) 값이 포함됩니다.



이 절의 사용 여부와 관계 없이 분리 사용자 ID는 DB2\_UTIL\_MSGPATH 레지스트리 변수가 표시하는 디렉토리 아래에 파일을 작성할 수 있는 권한이 있어야 합니다.

#### **TEMPFILES PATH** *temp-pathname*

로드 조작 중에 임시 파일을 작성할 때 사용되는 경로의 이름을 지정하며 서버 데이터베이스 파티션에 따라 완전한 이름이어야 합니다.

임시 파일은 파일 시스템 스페이스를 차지합니다. 때때로 이 스페이스는 대량으로 요구됩니다. 다음은 모든 임시 파일에 할당되어야 하는 파일 시스템 스페이스의 예상 크기입니다.

- 로드 유틸리티가 생성하는 각각의 메시지 당 136바이트
- 데이터 파일에 Long 필드나 LOB가 포함된 경우 15KB의 오버헤드. 이 수량은 INSERT 옵션이 지정되어 있고 테이블에 대량의 Long 필드나 LOB 데이터가 있는 경우 급격히 증가할 수 있습니다.

#### **INSERT**

로드 유틸리티가 실행될 수 있는 네 가지 모드 중 하나. 기존 테이블 데이터를 변경하지 않고 로드된 데이터를 테이블에 추가합니다.

#### **REPLACE**

로드 유틸리티가 실행될 수 있는 네 가지 모드 중 하나. 테이블에서 기존 데이터를 모두 삭제하고 로드된 데이터를 삽입합니다. 테이블 정의 및 인덱스 정의는 변경되지 않습니다. 계층 구조 간에 데이터를 이동하는 경우에 이 옵션을 사용하면 개별 부속 테이블이 아닌 전체 계층 구조에 대한 데이터만 바뀔 수 있습니다.

#### **KEEPDICTIONARY**

기존 압축 사전은 LOAD REPLACE 조작에서 보존됩니다. 테이블 COMPRESS 속성이 YES로 제공되면, 새로 교체된 데이터는 로드의 호출 이전에 존재하는 사전을 사용하여 압축됩니다. 테이블에 이미 존재하는 사전이 없는 경우, 테이블 COMPRESS 속성이 YES이면 테이블로 교체 중인 데이터를 사용하여 새 사전이 빌드됩니다. 이 경우 압축 사전을 빌드하는 데 필요한 데이터의 양은 ADC의 규정에 따릅니다. 이 데이터는 압축되지 않은 채로 테이블에 채워집니다. 사전이 테이블에 삽입되면, 로드될 남아 있는 데이터는 이 사전과 함께 압축됩니다. 이것은 디폴트 매개변수입니다. 요약 정보는 아래 표를 참조하십시오.

이전 사전이 테이블에 현재 있는 경우 다음 예에서는 이 사전을 보존합니다.

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



표 49. LOAD REPLACE KEEPDICTIONARY

압축	테이블 행 데이터 사전 존재	XML 스토리지 오브젝트 사전 존재 <sup>1</sup>	압축 사전	데이터 압축
YES	YES	YES	테이블 행 데이터 및 XML 사전을 보존합니다.	로드될 데이터는 압축됩니다.
YES	YES	NO	테이블 행 데이터 사전을 보존하고 새 XML 사전을 빌드합니다.	로드되는 테이블 행 데이터는 압축됩니다. XML 사전이 빌드된 후에, 로드될 남아 있는 XML 데이터는 압축됩니다.
YES	NO	YES	테이블 행 데이터 사전을 빌드하고 새 XML 사전을 보존합니다.	테이블 행 데이터 사전이 빌드된 후에, 로드될 남아 있는 테이블 행 데이터는 압축됩니다. 로드되는 XML 데이터는 압축됩니다.
YES	NO	NO	새 테이블 행 데이터 및 XML 사전을 빌드합니다.	사전이 빌드된 후에, 로드될 남아 있는 데이터는 압축됩니다.
NO	YES	YES	테이블 행 데이터 및 XML 사전을 보존합니다.	로드될 데이터는 압축되지 않습니다.
NO	YES	NO	테이블 행 데이터 사전을 보존합니다.	로드될 데이터는 압축되지 않습니다.
NO	NO	YES	테이블 행 사전에 영향을 미치지 않습니다. XML 사전을 보존합니다.	로드될 데이터는 압축되지 않습니다.
NO	NO	NO	영향을 미치지 않습니다.	로드될 데이터는 압축되지 않습니다.

주:

1. XML 컬럼이 DB2 버전 9.7 이상에서 테이블에 추가된 경우 또는 테이블이 온라인 테이블 이동을 사용하여 이주된 경우에만 테이블의 XML 스토리지 오브젝트의 압축 사전이 작성될 수 있습니다.

### RESETDICTIONARY

이 지시문은 LOAD REPLACE 처리 시 테이블 COMPRESS 속성이 YES인 경우 테이블 데이터 오브젝트의 새 사전을 빌드하도록 합니다. COMPRESS 속성이 NO이고 사전이 이미 테이블에 있으면 제거되며 새 사전은 테이블로 삽입되지 않습니다. 압축 사전은 오직 하나의 사용자 레코드로 빌드될 수 있습니다. 로드된 데이터 세트 크기가 0이고 기존에 존재하는 사전이 있는 경우, 이 사전은 보존되지 않습니다. 이 지시문으로 사전을 빌드하도록 요구되는 데이터의 양은 ADC의 규정에 따르지 않습니다. 요약 정보는 아래 표를 참조하십시오.

다음 예는 현재 사전을 재설정하여 새로운 사전을 작성합니다.

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

표 50. LOAD REPLACE RESETDICTIONARY

압축	테이블 행 데이터 사전 존재	XML 스토리지 오브젝트 사전 존재 <sup>1</sup>	압축 사전	데이터 압축
YES	YES	YES	새 사전을 빌드합니다 <sup>2</sup> . DATA CAPTURE CHANGES 옵션이 CREATE TABLE 또는 ALTER TABLE문에서 사용 가능하면, 현재 테이블 행 데이터 사전이 보존됩니다(실행기록 압축 사전이라고도 함).	사전이 빌드된 후에, 로드될 남아 있는 데이터는 압축됩니다.
YES	YES	NO	새 사전을 빌드합니다 <sup>2</sup> . DATA CAPTURE CHANGES 옵션이 CREATE TABLE 또는 ALTER TABLE문에서 사용 가능하면, 현재 테이블 행 데이터 사전이 보존됩니다(실행기록 압축 사전이라고도 함).	사전이 빌드된 후에, 로드될 남아 있는 데이터는 압축됩니다.
YES	NO	YES	새 사전을 빌드합니다.	사전이 빌드된 후에, 로드될 남아 있는 데이터는 압축됩니다.
YES	NO	NO	새 사전을 빌드합니다.	사전이 빌드된 후에, 로드될 남아 있는 데이터는 압축됩니다.
NO	YES	YES	사전을 제거합니다.	로드될 데이터는 압축되지 않습니다.
NO	YES	NO	테이블 행 데이터 사전을 제거합니다.	로드될 데이터는 압축되지 않습니다.
NO	NO	YES	XML 스토리지 오브젝트 사전을 제거합니다.	로드될 데이터는 압축되지 않습니다.
NO	NO	NO	영향을 미치지 않습니다.	모든 테이블 데이터는 압축되지 않습니다.

주:

1. XML 컬럼이 DB2 버전 9.7 이상에서 테이블에 추가된 경우 또는 테이블이 온라인 테이블 이동을 사용하여 이주된 경우에만 테이블의 XML 스토리지 오브젝트의 압축 사전이 작성될 수 있습니다.
2. 사전이 존재하고 압축 속성이 사용 가능하지만 테이블 파티션에 로드할 레코드가 없는 경우, 새 사전은 빌드될 수 없으며 RESETDICTIONARY 조작은 기존의 사전을 보존하지 않습니다.

**TERMINATE**

로드 유틸리티가 실행될 수 있는 네 가지 모드 중 하나. 일관성 지점이 패스된 경우라도 이전에 인터럽트된 로드 조작을 종료하고 조작이 시작된 특정 시점으로 롤백합니다. 조작에 관계된 테이블 스페이스의 상태가 정상으로 리턴되고 모든 테이블 오브젝트는 일관성을 갖게 됩니다(인덱스 오브젝트가 유효하지 않은

것으로 표시되는 경우, 인덱스 재빌드가 다음 액세스에서 자동으로 발생). 종료 되는 로드 조작이 LOAD REPLACE이면, 테이블은 LOAD TERMINATE 조 작 이후에 빈 테이블로 절단됩니다. 종료되는 로드 조작이 LOAD INSERT이 면, LOAD TERMINATE 조작 이후에 테이블은 원래의 모든 레코드를 보유 합니다. 사전 관리 요약 정보는 아래 표를 참조하십시오.

LOAD TERMINATE 옵션은 테이블 스페이스에서 백업 보류 상태를 제거하 지 않습니다.

## **RESTART**

로드 유틸리티가 실행될 수 있는 네 가지 모드 중 하나. 이전에 인터럽트된 로 드 조작을 재시작합니다. 로드 조작은 로드, 빌드 또는 삭제 단계의 마지막 일 관성 지점에서 자동으로 계속됩니다. 사전 관리 요약 정보는 아래 표를 참조하 십시오.

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

데이터가 로드되어야 할 데이터베이스 테이블을 지정합니다. 이 테이블은 시스 템 테이블, 선언된 임시 테이블 또는 작성된 임시 테이블일 수 없습니다. 별명 또는 완전하거나 규정에 맞지 않는 테이블 이름을 지정할 수 있습니다. 규정된 테이블 이름의 양식은 `schema.tablename`입니다. 규정에 맞지 않는 테이블 이 름이 지정되면 테이블은 CURRENT SCHEMA로 규정됩니다.

## *insert-column*

데이터가 삽입되어야 할 테이블 컬럼을 지정합니다.

로드 유틸리티는 하나 이상의 스페이스를 포함하는 이름을 가진 컬럼의 구문을 분석할 수 없습니다. 예를 들면, 다음과 같습니다.

```
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)')
```

Int 4 컬럼 때문에 실패합니다. 솔루션은 이 컬럼 이름을 큰따옴표 안에 두는 것입니다.

```
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*

오류 행이 복사될 예외 테이블을 지정합니다. 고유 인덱스 또는 기본 키를 위 반하는 모든 행이 복사됩니다. 규정에 맞지 않는 테이블 이름이 지정되면 테이 블은 CURRENT SCHEMA로 규정됩니다.

예외 테이블에 작성된 정보는 덤프 파일에는 작성되지 않습니다. 파티션된 데 이터베이스 환경에서는 예외 테이블이 로드 중인 테이블이 정의된 데이터베이 스 파티션에 대해 정의되어야 합니다. 그렇지 않으면, 덤프 파일에는 유효하지 않거나 구문 오류 때문에 로드될 수 없는 행이 포함됩니다.

XML 데이터 로드 시에는 다음과 같은 경우 FOR EXCEPTION절을 사용하여 로드 예외 테이블을 지정할 수 없습니다.

- 레이블 기반 액세스 제어(LBAC)를 사용하는 경우.
- 데이터 파티션 테이블로 데이터를 로드하는 경우.

#### **NORANGEEXC**

행이 범위 위반으로 거부되는 경우에는 예외 테이블에 삽입되지 않는다는 것을 표시합니다.

#### **NOUNIQUEEXC**

행이 고유 제한조건 위반으로 거부되는 경우에는 예외 테이블에 삽입되지 않는다는 것을 표시합니다.

#### **STATISTICS USE PROFILE**

이 테이블에 대해 정의된 프로파일에 따라 로드 중 통계를 수집하도록 로드를 지시합니다. 이 프로파일은 로드가 실행되기 전에 작성되어야 합니다. 이 프로파일은 RUNSTATS 명령으로 작성됩니다. 프로파일이 존재하지 않고 프로파일에 따라 통계를 수집하도록 로드에서 지시된 경우 경고가 리턴되며 통계가 수집되지 않습니다.

#### **STATISTICS NO**

통계가 수집되지 않고 카탈로그의 통계가 변경되지 않도록 지정합니다. 이는 디폴트값입니다.

#### **COPY NO**

포워드 복구 사용 가능한 경우(즉, *logretain* 또는 *userexit*가 설정된 경우), 테이블이 있는 테이블 스페이스가 백업 보류 상태에 있도록 지정합니다. COPY NO 옵션을 사용하면 테이블 스페이스 상태를 로드 진행 중 테이블 스페이스 상태로 둡니다. 이 상태는 로드가 완료되거나 중단되면 소멸되는 일시적인 상태입니다. 테이블 스페이스의 모든 테이블의 데이터는 테이블 스페이스 백업 또는 전체 데이터베이스 백업이 완료되어야 갱신하거나 삭제할 수 있습니다. 그러나 SELECT문을 사용하여 테이블의 데이터에 액세스할 수 있습니다.

복구 가능한 데이터베이스에 대해 COPY NO 옵션이 지정된 LOAD는 테이블 스페이스를 백업 보류 상태로 둡니다. 예를 들어, COPY NO 및 INDEXING MODE DEFERRED 옵션과 함께 LOAD를 수행하면 인덱스를 새로 고쳐야 합니다. 테이블에 대한 특정 쿼리는 인덱스 스캔을 해야 하며 인덱스를 새로 고쳐야 성공할 수 있습니다. 백업 보류 상태에 있는 테이블 스페이스에 위치한 인덱스는 새로 고칠 수 없습니다. 이러한 경우, 백업을 수행해야 테이블에 대한 액세스가 허용됩니다. 인덱스 새로 고침은 쿼리로 인덱스를 액세스할 때 데이터베이스에 의해 자동으로 완료됩니다. COPY NO, COPY YES 또는

NONRECOVERABLE 중 하나가 지정되지 않고 데이터베이스가 복구 가능한 경우(**logretain** 또는 **logarchmeth1**이 사용 가능함), COPY NO가 디폴트입니다.

#### **COPY YES**

로드된 데이터의 사본이 저장되도록 지정합니다. 포워드 복구가 사용 불가능한 경우 이 옵션은 유효하지 않습니다.

#### **USE TSM**

TSM(Tivoli Storage Manager)을 사용하여 사본이 저장되도록 지정합니다.

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

TSM 또는 벤더 제품과 함께 사용되는 입출력 세션 수. 디폴트값은 1입니다.

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

이미지 복사가 작성될 디바이스 또는 디렉토리를 지정합니다.

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

사용될 벤더 백업 및 리스토어 I/O 기능이 들어 있는 공유 라이브러리 이름 (Windows 운영 체제의 DLL). 여기에는 전체 경로가 포함될 수 있습니다. 전체 경로가 지정되지 않으면 디폴트값은 User Exit 프로그램이 있는 경로입니다.

#### **NONRECOVERABLE**

로드 트랜잭션이 복구 불가능하도록 표시하여 후속적인 롤 포워드 조치로 복구할 수 없도록 지정합니다. 롤 포워드 유틸리티가 트랜잭션을 건너뛰고 데이터가 로드 중인 테이블을 "유효하지 않음"으로 표시합니다. 또한 이 유틸리티는 해당 테이블에 대한 후속 트랜잭션을 무시합니다. 롤 포워드 조치가 완료된 후 이러한 테이블은 복구 불가능한 로드 조작 완료 후 커밋 지점 다음에 수행되는 백업(전체 또는 테이블 스페이스)에서 삭제되거나 리스토어될 수 있습니다.

이 옵션을 사용하면 로드 조작 후에 테이블 스페이스가 백업 보류 상태가 되지 않으며 로드 조작 중에 로드된 데이터의 사본을 작성할 필요가 없습니다. COPY NO, COPY YES 또는 NONRECOVERABLE 중 하나가 지정되지 않고 데이터베이스가 복구 가능하지 않은 경우(**logretain** 또는 **logarchmeth1**이 사용 가능하지 않음), NONRECOVERABLE이 디폴트입니다.

#### **WITHOUT PROMPTING**

데이터 파일의 목록에 로드될 모든 파일이 포함되고 나열된 디바이스 또는 디렉토리가 전체 로드 조작에 충분하도록 지정합니다. 연속 입력 파일을 찾을 수 없거나 목표 복사가 로드 조작이 완료되기 전에 채워지면 로드 조작에 실패하고 테이블이 로드 보류 상태가 됩니다.

이것은 디폴트값입니다. 정상적으로 사용자의 개입을 필요로 하는 조치는 오류 메시지를 리턴합니다.

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

유틸리티에서 데이터 전송에 필요한 버퍼 스페이스로 사용할 4KB 페이지의 수를 지정합니다(병렬 처리 수준과 무관함). 지정된 값이 알고리즘의 최소보다 작으면, 최소 필요 자원이 사용되고 경고는 리턴되지 않습니다.

이 메모리는 유틸리티 힙에서 바로 할당되며 크기는 *util\_heap\_sz* 데이터베이스 구성 매개변수로 수정할 수 있습니다.

값이 지정되지 않으면 런타임시 유틸리티에 의해 적절한 디폴트값이 계산됩니다. 디폴트값은 테이블의 일부 등록 정보 뿐 아니라 로더의 인스턴스화 시간에 유틸리티 힙에서 사용 가능한 여유 공간의 백분율을 기초로 합니다.

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

이 옵션은 로드 조작 중 SORTHEAP 데이터베이스 구성 매개변수를 겹쳐쓰는 값을 지정합니다. 이 옵션은 인덱스와 함께 테이블을 로드하는 경우와 INDEXING MODE 매개변수가 DEFERRED로 지정되지 않은 경우에만 의미가 있습니다. 지정된 값은 SORTHEAP의 값을 초과할 수 없습니다. 이 매개변수는 SORTHEAP의 값을 변경하지 않고 많은 인덱스가 있는 테이블을 로드할 때 사용되는 정렬 메모리를 조절하는 데 유용하며 일반 쿼리 처리에도 영향을 줍니다.

#### **CPU\_PARALLELISM** *n*

테이블 오브젝트를 빌드할 때 레코드를 구문 분석, 변환 및 포맷팅하기 위해 로드 유틸리티가 작성하는 프로세스 또는 스레드의 수를 지정합니다. 이 매개변수는 데이터베이스 파티션당 실행하는 프로세스 수를 이용하도록 설계되었습니다. 소스 데이터에 있는 기록 순서가 보존되므로 사전에 정렬된 데이터를 로드할 때 특히 유용합니다. 이 매개변수의 값이 0이거나 지정되지 않으면 로드 유틸리티는 적절한 디폴트값(일반적으로 사용 가능한 CPU 수에 기초함)을 사용합니다.

주:

1. 이 매개변수는 LOB 또는 LONG VARCHAR 필드 중 하나가 있는 테이블과 함께 사용되면 시스템 CPU 수나 사용자가 지정한 값과 관계없이 값은 1이 됩니다.
2. SAVECOUNT 매개변수에 값을 작게 지정하면 로더는 데이터 및 테이블 메타데이터를 둘 다 비우기 위해 훨씬 더 많은 입출력 조작을 수행하게 됩니다. CPU\_PARALLELISM이 1보다 크면 플러시 조작은 비동기로 수행되며 로더가 CPU를 이용할 수 있도록 합니다. CPU\_PARALLELISM이 1로 설정되면 로더는 일관성 지점 동안 입출력을 대기합니다. CPU\_PARALLELISM이 2로 설정되고 SAVECOUNT가 10000으로 설정



된 로드 조작용 CPU가 하나만 있어도 CPU\_PARALLELISM이 1로 설정된 동일한 조작용보다 빠르게 완료됩니다.

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

로드 유틸리티가 데이터를 테이블 스페이스 컨테이너에 작성하기 위해 작성하는 프로세스나 스레드의 수를 지정합니다. 값이 지정되지 않으면 유틸리티가 테이블 스페이스 컨테이너의 수 및 테이블의 등록 정보를 기반으로 하여 적절한 디폴트값을 선택합니다.

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

DATABASE 키워드로 선언된 커서에서 로드를 수행하거나 API sqlu\_remotefetch\_entry 미디어 항목을 사용하고 이 옵션이 YES로 설정되어 있는 경우, 로드 유틸리티는 가능한 리모트 데이터 소스로부터 페치(fetch)를 병렬 처리하려고 합니다. NO로 설정되면 병렬 페치(fetch)는 수행되지 않습니다. 디폴트값은 YES입니다. 자세한 정보는 *CURSOR* 파일 유형을 사용하여 데이터 이동을 참조하십시오.

#### **INDEXING MODE**

로드 유틸리티가 인덱스를 재빌드할 것인지 또는 인덱스를 점차적으로 확장할 것인지를 지정합니다. 가능한 값은 다음과 같습니다.

##### **AUTOSELECT**

로드 유틸리티는 자동으로 REBUILD나 INCREMENTAL 모드를 결정합니다. 이 결정은 로드 중인 데이터 양과 인덱스 트리의 용량을 기반으로 합니다. 인덱스 트리의 용량에 관한 정보는 인덱스 오브젝트에 저장되어 있습니다. 이 정보를 채우기 위해 RUNSTATS는 필요하지 않습니다. 디폴트 인덱스 모드는 AUTOSELECT입니다.

##### **REBUILD**

모든 인덱스가 재빌드됩니다. 이전 및 추가된 테이블 데이터에 대해 모든 인덱스 키 파트를 정렬하려면 유틸리티에 충분한 자원이 있어야 합니다.

##### **INCREMENTAL**

인덱스는 새로운 데이터와 함께 확장됩니다. 이 방법은 인덱스 여유 공간을 사용합니다. 삽입된 레코드에 대한 인덱스 키를 추가하려면 충분한 정렬 스페이스만 있으면 됩니다. 이 메소드는 인덱스 오브젝트가 유효하고 로드 조작용이 시작될 때 액세스할 수 있는 경우에만 지원됩니다 (예를 들어, DEFERRED 모드가 지정된 로드 조작용 직후에는 유효하지 않음). 이 모드가 지정되었지만 인덱스 상태 때문에 지원되지 않는 경우, 경고가 리턴되며 로드 조작용은 계속 REBUILD 모드입니다. 마찬가지로 로드 빌드 단계에서 로드 재시작 조작용이 시작된 경우, INCREMENTAL 모드는 지원되지 않습니다.

다음 조건에 모두 해당될 때 증분 인덱싱은 지원되지 않습니다.

- LOAD COPY 옵션이 지정되었습니다(USEREXIT 또는 LOGRETAIN 옵션을 사용하여 *logarchmeth1*).
- 테이블이 DMS 테이블 스페이스에 있습니다.
- 인덱스 오브젝트가 로드 중인 테이블에 속한 다른 테이블 오브젝트가 공유하는 테이블에 있습니다.

이 제한사항을 통과하려면 인덱스를 별도의 테이블 스페이스에 넣는 것이 바람직합니다.

### DEFERRED

이 모드를 지정하면 로드 유틸리티가 인덱스 작성을 시도하지 않습니다. 인덱스는 새로 고칠 필요가 있는 것으로 표시됩니다. 로드 조작과 관련이 없는 이러한 인덱스에 처음 액세스할 때 강제로 재빌드될 수 있습니다. 그렇지 않으면 인덱스는 데이터베이스가 재시작될 때 재빌드될 수 있습니다. 이러한 방법을 수행하려면 가장 큰 인덱스의 모든 키 부분에 대해 충분한 정렬 스페이스가 있어야 합니다. 인덱스 구성에 연속적으로 사용되는 총 시간은 REBUILD 모드에 필요한 시간보다 깁니다. 따라서 지연된 인덱싱으로 다중 로드 조작을 수행할 때 로드하지 않는 액세스에서 먼저 인덱스를 재빌드하게 하는 것보다는 시퀀스에서 마지막 로드 조작에서 인덱스 재빌드를 수행하게 하는 것이 바람직합니다(성능 관점에서).

지연된 인덱싱은 비고유 인덱스가 있는 테이블에만 지원되므로 로드 단계 중 삽입된 중복 키는 로드 조작 후 지속되지 않습니다.

### ALLOW NO ACCESS

로드가 로드 중 독점 액세스를 위해 목표 테이블을 잠급니다. 로드 중 테이블 상태는 로드 진행 중으로 설정됩니다. ALLOW NO ACCESS는 디폴트 동작입니다. LOAD REPLACE에는 이 옵션만 유효합니다.

테이블에 제한조건이 있을 때 테이블 상태는 로드 진행 중과 함께 무결성 설정 오류로 설정됩니다. 테이블을 무결성 설정 오류 상태에서 해제하려면 SET INTEGRITY문을 사용해야 합니다.

### ALLOW READ ACCESS

로드가 목표 테이블을 공유 모드로 잠급니다. 테이블 상태는 로드 진행 중 및 읽기 액세스 모두로 설정됩니다. 판독기는 테이블이 로드되는 동안 데이터의 비델타 부분에 액세스할 수 있습니다. 다시 말해, 테이블 판독기는 로드가 시작되기 전에 있던 데이터에 액세스할 수 있지만 로드되고 있는 데이터는 로드가 완료될 때까지 사용할 수 없습니다. ALLOW READ ACCESS 로드의 LOAD TERMINATE 또는 LOAD RESTART는 이 옵션을 사용할 수 있지만 ALLOW NO ACCESS 로드의 LOAD TERMINATE 또는 LOAD



RESTART는 이 옵션을 사용할 수 없습니다. 더우기 목표 테이블의 인덱스가 재빌드가 필요한 것으로 표시된 경우 이 옵션은 유효하지 않습니다.

테이블에 제한조건이 있으면 테이블 상태는 로드 진행 중, 읽기 액세스뿐만 아니라 무결성 설정 보류로 설정됩니다. 로드가 종료되면 테이블 상태 중에서 로드 진행 중 상태는 제거되지만 무결성 설정 보류 및 읽기 액세스 상태는 남아 있습니다. 테이블을 무결성 설정 보류 상태에서 해제하려면 SET INTEGRITY 문을 사용해야 합니다. 테이블의 상태가 무결성 설정 보류 및 읽기 액세스인 경우, 판독기는 데이터의 비델타 부분에는 여전히 액세스할 수 있으나 SET INTEGRITY문이 완료될 때까지 데이터의 새 (델타) 부분에는 액세스할 수 없습니다. 사용자는 SET INTEGRITY문을 발행하지 않고 동일한 테이블에서 다중 로드를 수행할 수 있습니다. 그러나 SET INTEGRITY문이 발행될 때까지 원본(확인된) 데이터만 볼 수 있습니다.

ALLOW READ ACCESS는 다음 수정자도 지원합니다.

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

인덱스가 재빌드되는 경우, 인덱스의 웨도우 사본이 테이블 스페이스 *tablespace-name*에 빌드되고 INDEX COPY PHASE 중 로드 맨 끝에서 원본 테이블 스페이스로 복사됩니다. 이 옵션에는 시스템 임시 테이블 스페이스만 사용할 수 있습니다. 테이블 스페이스를 지정하지 않을 경우 음영 인덱스는 인덱스 오브젝트와 동일한 테이블 스페이스에 작성됩니다. 음영 사본이 인덱스 오브젝트와 동일한 테이블 스페이스에 작성된 경우, 이전 인덱스 오브젝트에 대한 음영 인덱스 오브젝트의 복사는 순간적입니다. 음영 사본이 인덱스 오브젝트와 다른 테이블 스페이스에 있을 경우 실제 복사가 수행됩니다. 상당한 I/O 및 시간이 포함될 수 있습니다. INDEX COPY PHASE 중 로드 맨 끝에서 테이블이 오프라인 상태에 있는 동안 복사가 발생합니다.

이 옵션을 사용하지 않으면 음영 인덱스는 원본과 동일한 테이블 스페이스에서 빌드됩니다. 원본 인덱스와 음영 인덱스가 둘 다 디폴트로 동시에 동일한 테이블 스페이스에 있으므로 한 테이블 스페이스에 이들 인덱스를 모두 보유하기 위한 스페이스가 충분하지 않을 수 있습니다. 이 옵션을 사용하면 인덱스에 충분한 테이블 스페이스를 보유할 수 있습니다.

사용자가 INDEXING MODE REBUILD 또는 INDEXING MODE AUTOSELECT를 지정하지 않을 경우 이 옵션은 무시됩니다. INDEXING MODE AUTOSELECT가 선택되고 로드가 인덱스를 점차적으로 갱신하도록 선택할 경우에도 이 옵션이 무시됩니다.

#### **SET INTEGRITY PENDING CASCADE**

LOAD로 인해 테이블이 무결성 설정 보류 상태에 놓일 경우, 사용자는 SET INTEGRITY PENDING CASCADE 옵션을 사용하여 로드된 테이블의 무결

성 설정 보류 상태가 모든 하위(하위 외부 키 테이블, 하위 즉시 구체화된 쿼리 테이블 및 하위 즉시 스테이징 테이블 포함)에 즉시 연쇄되는지 여부를 지정할 수 있습니다.

### **IMMEDIATE**

무결성 설정 보류 상태가 하위 외부 키 테이블, 하위 즉시 구체화된 쿼리 테이블 및 하위 즉시 스테이징 테이블로 즉시 확장됨을 표시합니다. LOAD INSERT 조작의 경우, IMMEDIATE 옵션을 지정해도 무결성 설정 보류 상태는 하위 외부 키 테이블로 확장되지 않습니다.

나중에 로드된 테이블에 제한조건 위반이 있는지 여부를 점검할 때(SET INTEGRITY문의 IMMEDIATE CHECKED 옵션을 사용하여) 무결성 설정 보류 읽기 액세스에 있던 하위 외부 키 테이블은 무결성 설정 보류 권한 없음 상태가 됩니다.

### **DEFERRED**

로드된 테이블만이 무결성 설정 보류 상태에 놓임을 표시합니다. 하위 외부 키 테이블, 하위 즉시 구체화된 쿼리 테이블 및 하위 즉시 스테이징 테이블의 상태는 변경되지 않습니다.

하위 외부 키 테이블에 제한조건 위반이 있는지 여부를 점검할 때(SET INTEGRITY문의 IMMEDIATE CHECKED 옵션을 사용하여) 하위 외부 키 테이블 은 나중에 내재적으로 무결성 설정 보류 상태에 놓일 수 있습니다. 하위 즉시 구체화된 쿼리 테이블 및 하위 즉시 스테이징 테이블은 해당 기본 테이블 중 하나에 무결성 위반이 있는지 여부를 점검할 때 내재적으로 무결성 설정 보류 상태가 됩니다. 하위 테이블이 무결성 설정 보류 상태가 되었음을 나타내는 경고(SQLSTATE 01586)가 발행됩니다. 이들 하위 테이블이 언제 무결성 설정 보류 상태가 되는지에 대해서는 SQL 참조서에서 SET INTEGRITY문의 참고 절을 참조하십시오.

SET INTEGRITY PENDING CASCADE 옵션을 지정하지 않은 경우,

- 로드된 테이블만이 무결성 설정 보류 상태에 놓입니다. 하위 외부 키 테이블, 하위 즉시 구체화된 쿼리 테이블 및 하위 즉시 스테이징 테이블의 상태는 계속 변경되지 않으며 나중에 로드된 테이블에 제한조건 위반이 있는지 점검할 때 내재적으로 무결성 설정 보류 상태가 될 수 있습니다.

LOAD 결과 목표 테이블이 무결성 설정 보류 상태가 되지 않을 경우 SET INTEGRITY PENDING CASCADE 옵션은 무시됩니다.

### **LOCK WITH FORCE**

이 유틸리티는 로드 처리에서 테이블 잠금을 포함하는 다양한 잠금을 획득합니다. 대기라기 보다는 시간종료라고 할 수 있으며, 잠금을 획득할 때 로드는 이 옵션을 통해 목표 테이블에 충돌 잠금을 보유하고 있는 기타 응용프로그램을 강

제로 해제할 수 있습니다. 시스템 카탈로그 테이블에서 충돌 잠금을 보유하고 있는 응용프로그램은 로드 유틸리티로 강제로 해제할 수 없습니다. 강제로 응용프로그램은 롤백되고 로드 유틸리티가 필요로 하는 잠금을 릴리스합니다. 그런 다음 로드 유틸리티는 계속 진행될 수 있습니다. 이 옵션을 사용하려면 FORCE APPLICATIONS 명령(SYSADM 또는 SYSCTRL)과 동일한 권한이 필요합니다.

ALLOW NO ACCESS 로드는 로드 조작이 시작될 때 충돌 잠금을 보유하고 있는 응용프로그램을 강제 실행할 수 있습니다. 로드가 시작될 때 유틸리티는 테이블을 쿼리하거나 수정하려고 시도 중인 응용프로그램을 강제 실행할 수 있습니다.

ALLOW READ ACCESS 로드는 로드 조작이 시작되거나 종료될 때 충돌 잠금을 보유하고 있는 응용프로그램을 강제 실행할 수 있습니다. 로드가 시작될 때 유틸리티는 테이블 수정하려고 시도 중인 응용프로그램을 강제 실행할 수 있습니다. 로드 조작이 종료될 때 로드 유틸리티는 테이블을 쿼리하거나 수정하려고 시도 중인 응용프로그램을 강제 실행할 수 있습니다.

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

유틸리티에 데이터를 입력하기 위해 호출될 실행 파일 이름을 지정합니다.

#### **REDIRECT**

##### **INPUT FROM**

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

*input-buffer*에 지정된 바이트 스트림을 지정된 실행 파일을 실행하는 프로세스의 STDIN 파일 디스크립터로 패스합니다.

###### **FILE** *input-file*

이 클라이언트측 파일의 내용을 지정된 실행 파일을 실행하는 프로세스의 STDIN 파일 디스크립터로 패스합니다.

##### **OUTPUT TO**

###### **FILE** *output-file*

STDOUT 및 STDERR 파일 디스크립터를 지정된 서버측 파일로 캡처합니다.

#### **PARALLELIZE**

여러 User Exit 프로세스를 동시 호출하여 로드 유틸리티로 들어가는 데이터의 처리량을 증가시킵니다. 이 옵션은 다중 파티션 데이터베이스 환경에서만 적용할 수 있으므로 단일 파티션 데이터베이스 환경에서는 무시됩니다.

자세한 정보는 사용자 정의된 응용프로그램을 사용하여 데이터 이동을 참조하십시오.

### **PARTITIONED DB CONFIG** *partitioned-db-option*

다중 데이터베이스 파티션에 분산된 테이블 로드를 실행할 수 있도록 합니다. PARTITIONED DB CONFIG 매개변수를 사용하면 파티션된 데이터베이스에만 해당되는 구성 옵션을 지정할 수 있습니다. *partitioned-db-option* 값은 다음과 같이 지정할 수 있습니다.

```
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
```

이들 옵션의 자세한 설명은 파티션된 데이터베이스 환경을 위한 로드 구성 옵션을 참조하십시오.

### **RESTARTCOUNT**

예약됨.

### **USING** *directory*

예약됨.

### **예 :**

파일로부터 직원 테이블 데이터에 대해 대체 옵션을 사용하여 로드를 발행하십시오.

```
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' )
```

다음은 단일 파티션 데이터베이스로부터 출력 예입니다.

결과 세트 1

-----

ROWS_READ	ROWS_SKIPPED	ROWS_LOADED	ROWS_REJECTED	...
-----------	--------------	-------------	---------------	-----

```
-----...-----...-----...-----...
                32                0                32                0 ...
```

1개의 레코드가 선택되었습니다.

리턴 상태 = 0

단일 파티션 데이터베이스의 출력(계속).

```
... ROWS_DELETED      ROWS_COMMITTED      MSG_RETRIEVAL
-----...-----...-----...
...                0                32 SELECT SQLCODE, MSG_TEXT FROM
...                                     TABLE(SYSPROC.ADMIN_GET_MSGS(
...                                     '2203498_thx')) AS MSG
```

단일 파티션 데이터베이스의 출력(계속).

```
... MSG_REMOVAL
-----...-----...
... CALL SYSPROC.ADMIN_REMOVE_MSGS('2203498_thx')
...
...
```

주: 이 결과 세트에는 ROWS\_PARTITIONED 및 NUM\_AGENTINFO\_ENTRIES 컬럼도 리턴되지만 이들 컬럼은 다중 파티션 데이터베이스로 로드될 때만 채워지므로 널(NULL)로 설정됩니다.

다음은 다중 파티션 데이터베이스로부터의 출력 예입니다.

결과 세트 1

```
-----
ROWS_READ      ROWS_REJECTED      ROWS_PARTITIONED      NUM_AGENTINFO_ENTRIES ...
-----...-----...-----...
                32                0                32                5 ...
...
...
...
...
...
```

1개의 레코드가 선택되었습니다.

다중 파티션 데이터베이스의 출력(계속).

```
... 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
```

주: 이 결과 세트에는 ROWS\_SKIPPED, ROWS\_LOADED, ROWS\_DELETED 및 ROWS\_COMMITTED 컬럼도 리턴되지만 이들 컬럼은 단일 파티션 데이터베이스로 로드될 때만 채워지므로 널(NULL)로 설정됩니다.

다중 파티션 데이터베이스의 출력(계속).

결과 세트 2

```
-----
```

DBPARTITIONNUM	SQLCODE	TABSTATE	AGENTTYPE
10	0	NORMAL	LOAD
20	0	NORMAL	LOAD
30	0	NORMAL	LOAD
20	0	NORMAL	PARTITION
10	0	NORMAL	PRE_PARTITION

1개의 레코드가 선택되었습니다.

리턴 상태 = 0

## XML 문서에서 데이터 로드 예

### XML 데이터 로딩

#### 예 1

테이블에 삽입되는 문서를 설명하기 위해 사용자가 XDS 필드로 데이터 파일을 구성했습니다. 다음과 같이 표시됩니다.

```
1, "<XDS FIL=""file1.xml"" />"
2, "<XDS FIL='file2.xml' OFF='23' LEN='45' />"
```

첫 번째 행의 경우, XML 문서는 file1.xml 파일로 식별됩니다. 문자 분리문자는 큰 따옴표이며 큰따옴표는 XDS 안에 존재하므로, XDS 안에 포함된 큰따옴표는 이중이 됩니다. 두 번째 행의 경우, XML 문서는 file2.xml 파일로 식별되며, 바이트 오프셋 23에서 시작하며 45바이트 길이입니다.

#### 예 2:

사용자는 XML 컬럼의 구문 분석 또는 유효성 확인 옵션 없이 로드 명령을 발행하며, 데이터는 정상적으로 로드됩니다.

```
LOAD FROM data.del of DEL INSERT INTO mytable
```

### 커서에서 XML 데이터 로딩

커서에서 데이터 로딩은 일반 관계형 컬럼 유형과 동일합니다. 두 개의 테이블 T1 및 T2가 있으며, 각각 C1이라는 단일 XML 컬럼으로 구성됩니다. T1에서 T2로 로드하기 위해 사용자는 우선 커서를 선언합니다.

```
DECLARE X1 CURSOR FOR SELECT C1 FROM T1;
```

다음으로 사용자는 커서 유형을 사용하여 LOAD를 발행할 수 있습니다.

```
LOAD FROM X1 of CURSOR INSERT INTO T2
```

XML 특정 LOAD 옵션을 커서 유형에 적용하는 것은 파일에서 로드하는 것과 같습니다.

## 사용 시 참고사항

- 데이터는 입력 파일에 나타나는 시퀀스로 로드됩니다. 특정 시퀀스를 원할 경우 로드를 시도하기 전에 데이터를 정렬해야 합니다. 소스 데이터 순서의 보존이 필요하지 않은 경우, 로드 유틸리티의 파일 유형 수정자 섹션에서 아래 설명된 ANYORDER 파일 유형 수정자 사용을 고려하십시오.
- 로드 유틸리티는 기존 정의에 따라 인덱스를 빌드합니다. 고유 키에 대한 중복을 처리하기 위해 예외 테이블을 사용합니다. 이 유틸리티는 참조 무결성을 강제하거나 제한조건 점검을 수행하거나 로드되는 테이블에 종속된 구체화된 쿼리 테이블을 갱신하지 않습니다. 참조 또는 점검 제한조건을 포함하는 테이블은 무결성 설정 보류 상태에 놓입니다. 로드되고 있는 테이블에 종속되고 REFRESH IMMEDIATE를 사용하여 정의된 요약 테이블도 무결성 설정 보류 상태에 놓입니다. 이들 테이블을 무결성 설정 보류 상태에서 해제하려면 SET INTEGRITY문을 발행하십시오. 복제된 구체화된 쿼리 테이블에서는 로드 조작을 수행할 수 없습니다.
- 테이블에 클러스터링 인덱스가 존재할 경우, 로드하기 전에 클러스터링 인덱스에 대해 데이터를 정렬해야 합니다. 그러나 다차원적으로 클러스터된(MDC) 테이블에 로드할 경우에는 미리 데이터를 정렬할 필요가 없습니다.
- 보호 설정된 테이블에 로드할 때 예외 테이블을 지정하면, 유효하지 않은 보안 레이블로 보호 설정된 행은 이 테이블로 보내집니다. 따라서 예외 테이블에 액세스할 수 있는 사용자는 일반적으로 액세스 권한이 없는 데이터에 액세스할 수 있습니다. 보안 강화를 위해 누구에게 예외 테이블 액세스 권한을 부여할 것인지 유의하고, 행을 수리한 후 로드되고 있는 테이블에 복사한 후에는 곧바로 각 행을 삭제하고 예외 테이블 사용을 완료하면 곧바로 예외 테이블을 삭제하십시오.
- 내부 형식으로 된 보안 레이블에는 줄 바꾸기 문자가 포함될 수 있습니다. DEL 파일 형식을 사용하여 파일을 로드할 경우, 이들 줄 바꾸기 문자는 분리문자로 오인될 수 있습니다. 이러한 문제점이 발생하면 LOAD 명령에 delprioritychar 파일 유형 수정자를 지정하여 분리문자에 대해 이전 디폴트 우선순위를 사용하십시오.
- LOAD 유틸리티는 유형 2 연결의 경우 프로시저가 이유 코드 1과 함께 SQL3009N을 리턴하게 하는 조작이 시작될 때 COMMIT문을 발행합니다.
- LOAD 명령에 사용되는 경로는 서버 코디네이터 파티션에서 유효한 완전한 경로여야 합니다.
- DECLARE CURSOR 명령 중에 DATABASE 키워드가 지정된 CURSOR 파일 유형을 사용하여 로드 조작을 수행할 경우, 현재(로드 중에) 연결된 데이터베이스에 대해 인증할 때 사용한 사용자 ID 및 암호를 사용하여 소스 데이터베이스 (DECLARE CURSOR 명령의 DATABASE 옵션에 의해 지정됨)에 대해서도 인증합니다. 로드 중인 데이터베이스에 대한 연결용으로 지정된 사용자 ID 및 암호가 없을 경우 DECLARE CURSOR 명령 중에 소스 데이터베이스에 대한 사용자 ID 및 암호를 지정해야 합니다.



- 개별적 파트가 Windows 시스템에서 AIX 시스템으로 복사되는 다중 파트 PC/IXF 파일 로딩이 지원됩니다. 모든 파일의 이름이 LOAD 명령에서 지정되어야 합니다. 예를 들어, LOAD FROM DATA.IXF, DATA.002 OF IXF INSERT INTO TABLE1. 논리적으로 분할된 PC/IXF 파일에서 Windows 운영 체제로의 로딩은 지원되지 않습니다.
- 실패한 LOAD 재시작 시, 인덱스의 REBUILD 모드를 사용하도록 BUILD 단계를 강제하는 기존 동작을 따릅니다.
- 데이터베이스 중간에서는 XML 문서를 로드할 수 없으며 이 경우 오류 메시지 SQL1407N이 리턴됩니다.

## LOAD TERMINATE 및 LOAD RESTART 사전 관리의 요약

다음 도표는 TERMINATE 지시문 아래에서 LOAD 처리의 압축 사전 관리 동작을 요약합니다.

표 51. LOAD TERMINATE 사전 관리

테이블 COMPRESS 속성	LOAD 이전에 테이블 행 데이터 사전 존재	XML 스토리지 오브젝 트 사전이 LOAD 이전 에 존재 <sup>1</sup>	TERMINATE: LOAD REPLACE KEEPDICTIONARY 또는 LOAD INSERT	TERMINATE: LOAD REPLACE RESETDICTIONARY
YES	YES	YES	기존 사전을 보존합니다.	어떤 사전도 보존하지 않습니다. <sup>2</sup>
YES	YES	NO	기존 사전을 보존합니다.	어떤 것도 보존하지 않습니다. <sup>2</sup>
YES	NO	YES	기존 사전을 보존합니다.	어떤 것도 보존하지 않습니다.
YES	NO	NO	어떤 것도 보존하지 않습니다.	어떤 것도 보존하지 않습니다.
NO	YES	YES	기존 사전을 보존합니다.	어떤 것도 보존하지 않습니다.
NO	YES	NO	기존 사전을 보존합니다.	어떤 것도 보존하지 않습니다.
NO	NO	YES	기존 사전을 보존합니다.	어떤 것도 보존하지 않습니다.
NO	NO	NO	아무 것도 수행하지 않습니다.	아무 것도 수행하지 않습니다.

주:

1. XML 컬럼이 DB2 버전 9.7 이상에서 테이블에 추가된 경우 또는 테이블이 온라인 테이블 이동을 사용하여 이주된 경우에만 테이블의 XML 스토리지 오브젝트의 압축 사전이 작성될 수 있습니다.
2. 테이블에서 데이터 캡처가 사용 가능한 특수한 경우, 테이블 행 데이터 사전이 보존됩니다.

LOAD RESTART는 최종 일관성 지점까지 테이블을 절단합니다. 압축 사전은 LOAD RESTART 처리의 파트로서, 최종 LOAD 일관성 지점이 지정된 시간에 테이블에 존재합니다. 이 경우, LOAD RESTART는 새 사전을 작성하지 않습니다. 가능한 조건에 대한 요약은 아래 표를 참조하십시오.



표 52. LOAD RESTART 사전 관리

테이블 COMPRESS 속성	LOAD 일관성 지점 이전 테이블 행 데이터 사전 존재 <sup>1</sup>	최종 로드 이전에 XML 스토리지 오브젝트 사전 존재 <sup>2</sup>	RESTART: LOAD REPLACE KEEPDICTIONARY 또는 LOAD INSERT	RESTART: LOAD REPLACE RESETDICTIONARY
YES	YES	YES	기존 사전을 보존합니다.	기존 사전을 보존합니다.
YES	YES	NO	기존의 테이블 행 데이터를 빌드하고 ADC에 따라 XML 사전을 빌드합니다.	기존의 테이블 행 데이터 사 전을 빌드하고 XML 사전을 빌드합니다.
YES	NO	YES	ADC에 따른 테이블 행 사전을 빌드합니다. 기존 XML 사전을 보존합니다.	테이블 행 데이터 사전을 빌 드합니다. 기존 XML 사전을 보존합니다.
YES	NO	NO	ADC에 따른 XML 사전과 테 이블 행 데이터를 빌드합니다.	테이블 행 데이터 및 XML 사전을 빌드합니다.
NO	YES	YES	기존 사전을 보존합니다.	기존 사전을 제거합니다.
NO	YES	NO	기존 테이블 행 데이터 사전을 보존합니다.	기존 테이블 행 데이터 사전 을 제거합니다.
NO	NO	YES	기존 XML 사전을 보존합니다.	기존 XML 사전을 제거합니 다.
NO	NO	NO	아무 것도 수행하지 않습니다.	아무 것도 수행하지 않습니 다.

주:

1. XML 데이터 로드 시, 조작 시작부터 로드 단계 재시작 중 실패하는 로드 조작의 경우 SAVECOUNT 옵션이 무시됩니다.
2. XML 컬럼이 DB2 버전 9.7 이상에서 테이블에 추가된 경우 또는 테이블이 온라인 테이블 이동을 사용하여 이주된 경우에만 테이블의 XML 스토리지 오브젝트의 압축 사전이 작성될 수 있습니다.

## 결과 세트 정보

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다. 실행이 완료되면 명령은 추가 정보를 리턴합니다. 단일 파티션 데이터베이스는 하나의 결과 세트를 리턴하고 다중 파티션 데이터베이스는 두 개의 결과 세트를 리턴합니다.

- 표 53: 로드 조작에 대한 결과 세트.
- 142 페이지의 표 54: 결과 세트 2에는 다중 파티션 로드 조작의 각 데이터베이스 파티션에 대한 정보가 포함됩니다.

표 53. LOAD 명령에서 리턴된 결과 세트

컬럼 이름	데이터 유형	설명
ROWS_READ	BIGINT	로드 조작 중 읽은 행 수
ROWS_SKIPPED	BIGINT	로드 조작이 시작되기 전에 건너뛴 행 수. 이 정보는 단일 파티션 데이터베이스의 경우에만 리턴됩니다.

표 53. LOAD 명령에서 리턴된 결과 세트 (계속)

컬럼 이름	데이터 유형	설명
ROWS_LOADED	BIGINT	목표 테이블로 로드된 행 수. 이 정보는 단일 파티션 데이터베이스의 경우에만 리턴됩니다.
ROWS_REJECTED	BIGINT	목표 테이블로 로드될 수 없는 행 수.
ROWS_DELETED	BIGINT	목표 테이블로 로드되지 않은 중복 행 수. 이 정보는 단일 파티션 데이터베이스의 경우에만 리턴됩니다.
ROWS_COMMITTED	BIGINT	처리된 총 행 수: 목표 테이블로 성공적으로 로드된 행 수 및 건너뛰고 거부된 행 수. 이 정보는 단일 파티션 데이터베이스의 경우에만 리턴됩니다.
ROWS_PARTITIONED	BIGINT	모든 데이터베이스 분배 에이전트에 의해 분배된 행 수. 이 정보는 다중 파티션 데이터베이스의 경우에만 리턴됩니다.
NUM_AGENTINFO_ENTRIES	BIGINT	다중 파티션 데이터베이스에 대한 두 번째 결과 세트에 리턴된 항목 수. 이는 로드 조작으로 생성된 에이전트 정보 항목 수입니다. 이 정보는 다중 파티션 데이터베이스의 경우에 만 리턴됩니다.
MSG_RETRIEVAL	VARCHAR(512)	이 유틸리티에서 작성된 메시지를 검색하는 데 사용되는 SQL문. 예를 들면, 다음과 같습니다.  <pre>SELECT SQLCODE, MSG   FROM TABLE   (SYSPROC.ADMIN_GET_MSGS    ('2203498_thx')) AS MSG</pre> 이 정보는 MESSAGES ON SERVER절이 지정된 경우에만 리턴됩니다.
MSG_REMOVAL	VARCHAR(512)	이 유틸리티에서 작성된 메시지를 정리하는 데 사용되는 SQL문. 예를 들면, 다음과 같습니다.  <pre>CALL SYSPROC.ADMIN_REMOVE_MSGS   ('2203498_thx')</pre> 이 정보는 MESSAGES ON SERVER절이 지정된 경우에만 리턴됩니다.

표 54. 다중 데이터베이스 파티션의 각 데이터베이스 파티션에 대해 LOAD 명령에 의해 리턴된 결과 세트 2.

컬럼 이름	데이터 유형	설명
DBPARTITIONNUM	SMALLINT	에이전트가 로드 조작을 실행한 데이터베이스 파티션 번호.
SQLCODE	INTEGER	로드 처리에서 결과된 최종 SQLCODE.

표 54. 다중 데이터베이스 파티션의 각 데이터베이스 파티션에 대해 LOAD 명령에 의해 리턴된 결과 세트 2. (계속)

컬럼 이름	데이터 유형	설명
TABSTATE	VARCHAR(20)	로드 조작 완료 후 테이블 상태. 파일 이름은 다음 중 하나입니다. <ul style="list-style-type: none"> <li>LOADPENDING: 로드가 완료되지 않았지만 파티션의 테이블이 LOAD PENDING 상태로 남아 있음을 나타냅니다. 데이터베이스 파티션에서 로드 재시작 또는 종료 조작을 수행해야 합니다.</li> <li>NORMAL: 데이터베이스 파티션에서 로드가 완료되었으며 테이블이 LOAD IN PROGRESS(또는 LOAD PENDING) 상태를 벗어났음을 나타냅니다. 추가 제한조건 처리가 필요한 경우 테이블이 계속해서 무결성 설정 보류 상태에 있을 수 있으나 해당 인터페이스에서는 이 상태를 보고하지 않습니다.</li> <li>UNCHANGED: 오류 때문에 로드가 완료되지 않았지만 테이블 상태가 아직 변경되지 않았음을 나타냅니다. 데이터베이스 파티션에서 로드 재시작 또는 종료 조작을 수행할 필요가 없습니다.</li> </ul> 주: 이 인터페이스는 가능한 테이블 상태 중 일부를 리턴합니다.
AGENTTYPE	VARCHAR(20)	에이전트 유형은 다음과 같습니다. <ul style="list-style-type: none"> <li>FILE_TRANSFER</li> <li>LOAD</li> <li>LOAD_TO_FILE</li> <li>PARTITIONING</li> <li>PRE_PARTITIONING</li> </ul>

## 로드 유틸리티의 파일 유형 수정자

표 55. 로드 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식

수정자	설명
anyorder	이 수정자는 <i>cpu_parallelism</i> 매개변수와 결합하여 사용됩니다. SMP 시스템에서 중요한 추가 성능 이점이 발생하며, 소스 데이터 순서의 보존이 필요하지 않음을 지정합니다. <i>cpu_parallelism</i> 의 값이 1이면, 이 옵션은 무시됩니다. 일관성 지점 이후의 응급 복구는 데이터가 시퀀스로 로드되도록 요구하므로 SAVECOUNT > 0인 경우 이 옵션은 지원되지 않습니다.
generatedignore	이 수정자는 생성된 모든 컬럼의 데이터가 데이터 파일에 표시되지만 무시해야 함을 로드 유틸리티에 알립니다. 이로 인해 생성된 모든 컬럼의 값이 유틸리티에 의해 생성됩니다. 이 수정자는 generatedmissing 또는 generatedoverride 수정자 중 하나와 함께 사용될 수 없습니다.
generatedmissing	이 수정자가 지정된 경우, 유틸리티는 입력 데이터 파일에는 생성된 컬럼의 데이터를 포함하지 않음(NULL 값도 포함)을 가정합니다. 이것은 유틸리티에 의해 모든 컬럼의 값을 생성하는 결과를 가져옵니다. 이 수정자는 generatedignore 또는 generatedoverride 수정자 중 하나와 함께 사용될 수 없습니다.

표 55. 로드 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
generatedoverride	<p>이 수정자는 테이블에서 생성되는 모든 컬럼의 사용자 제공 데이터를 승인하도록 로드 유틸리티에 지시합니다(이러한 유형의 컬럼에 대한 일반 규칙과 반대). 다른 데이터베이스 시스템에서 데이터를 이주하거나, ROLLFORWARD DATABASE 명령에서 RECOVER DROPPED TABLE 옵션을 사용하여 복구된 데이터에서 테이블을 로드할 때 유용합니다. 이 수정자가 사용될 때, 데이터가 없는 행이나 널(null) 값을 허용하지 않는 생성된 컬럼의 NULL 데이터는 거부됩니다(SQL3116W). 해당 수정자 사용 시, 테이블이 무결성 설정 보류 상태에 놓입니다. 사용자 제공 값을 검증하지 않고 무결성 설정 보류 상태로부터 테이블을 얻으려면, 로드 조작 이후에 다음 명령을 발행하십시오.</p> <pre>SET INTEGRITY FOR &lt; table-name &gt; GENERATED COLUMN IMMEDIATE UNCHECKED</pre> <p>무결성 설정 보류 상태로부터 테이블을 얻고 사용자 제공 값의 검증을 강제하려면, 로드 조작 이후에 다음 명령을 발행하십시오.</p> <pre>SET INTEGRITY FOR &lt; table-name &gt; IMMEDIATE CHECKED.</pre> <p>이 수정자가 지정되고 임의의 파티셔닝 키, 차원 키 또는 분산 키에 생성된 컬럼이 있으면, LOAD 명령은 자동으로 수정자를 generatedignore로 변환하고 로드를 진행합니다. 이것은 생성된 모든 컬럼 값을 재생성하는 효과를 갖습니다.</p> <p>이 수정자는 generatedmissing 또는 generatedignore 수정자 중 하나와 함께 사용될 수 없습니다.</p>
identityignore	<p>이 수정자는 ID 컬럼의 데이터가 데이터 파일에 표시되지만 무시해야 함을 로드 유틸리티에 알립니다. 이로 인해 유틸리티에 의해 모든 ID 값이 생성됩니다. GENERATED ALWAYS 및 GENERATED BY DEFAULT ID 컬럼 둘 다의 동작이 동일합니다. GENERATED ALWAYS 컬럼의 경우, 어떤 행도 거부되지 않음을 의미합니다. 이 수정자는 identitymissing 또는 identityoverride 수정자 중 하나와 함께 사용될 수 없습니다.</p>
identitymissing	<p>이 수정자가 지정된 경우, 유틸리티는 입력 데이터 파일에는 ID 컬럼의 데이터를 포함하지 않음(NULL 값도 비포함)을 가정하므로 각 행의 값을 생성합니다. GENERATED ALWAYS 및 GENERATED BY DEFAULT ID 컬럼 둘 다의 동작이 동일합니다. 이 수정자는 identityignore 또는 identityoverride 수정자 중 하나와 함께 사용될 수 없습니다.</p>
identityoverride	<p>GENERATED ALWAYS로 정의된 ID 컬럼이 로드되는 테이블에 있을 때에만 이 수정자를 사용해야 합니다. 해당 컬럼에 대해 명시적, 널(NULL)이 아닌 데이터를 승인하도록 유틸리티에 지시합니다(이러한 유형의 ID 컬럼에 대한 일반 규칙과 반대). 테이블이 GENERATED ALWAYS로 정의되어야 할 때 다른 데이터베이스 시스템에서 데이터를 이주하거나, ROLLFORWARD DATABASE 명령에서 DROPPED TABLE RECOVERY 옵션을 사용하여 복구된 데이터에서 테이블을 로드할 때 유용합니다. 이 수정자가 사용될 때, 데이터가 없는 행이나 ID 컬럼의 NULL 데이터는 거부됩니다(SQL3116W). 이 수정자는 identitymissing 또는 identityignore 수정자 중 하나와 함께 사용될 수 없습니다. 이 옵션 사용 시 로드 유틸리티는 테이블의 ID 컬럼에서 값의 고유성을 검증하거나 유지하려고 시도하지 않습니다.</p>
indexfreespace=x	<p>x는 0 - 99의 정수입니다. 로드로 인덱스를 재빌드할 때 여유 공간으로 남게 되는 각 인덱스 페이지의 백분율입니다. INDEXING MODE INCREMENTAL을 사용하여 로드할 때 이 옵션은 무시됩니다. 페이지의 첫 번째 항목은 제한없이 추가되며, 여유 공간 비율 임계값을 유지하도록 추후 항목이 추가됩니다. 디폴트값은 CREATE INDEX 시간에 사용됩니다.</p> <p>이 값은 CREATE INDEX문에 지정된 PCTFREE 값보다 우선합니다. indexfreespace 옵션은 인덱스 단말 페이지에만 영향을 미칩니다.</p>

표 55. 로드 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
lobsinfile	<p><i>lob-path</i>는 LOB 데이터를 포함하는 파일에 대한 경로를 지정합니다. ASC, DEL 또는 IXF 로드 입력 파일은 LOB 컬럼에 LOB 데이터가 있는 파일의 이름을 포함합니다.</p> <p>이 옵션은 CURSOR 파일 유형과 함께 지원되지 않습니다.</p> <p>LOBS FROM 절은 『lobsinfile』 수정자가 사용될 때 LOB 파일이 위치하는 곳을 지정합니다. LOBS FROM 절은 내재적으로 LOBSINFILE 동작을 활성화합니다. LOBS FROM 절은 데이터 임포트 중 LOB 파일을 검색하기 위해 경로 목록을 LOAD 유틸리티로 전달합니다.</p> <p>각 경로는 LLS(Lob Location Specifier)에 의해 포인팅되는 최소한 하나의 LOB를 포함하는 최소한 하나의 파일을 데이터 파일에 포함합니다. LLS는 LOB 파일 경로에 저장된 파일에서 LOB 위치의 문자열 표시입니다. LLS의 형식은 <i>filename.ext.nnn.mmm</i>이며, 여기서 <i>filename.ext</i>는 LOB를 포함하는 파일의 이름이며, <i>nnn</i>은 파일 내에서 LOB의 오프셋을 바이트로 나타낸 것이며, <i>mmm</i>은 LOB의 길이를 바이트로 나타낸 것입니다. 예를 들어, 문자열 db2exp.001.123.456/가 데이터 파일에 저장되는 경우, LOB는 db2exp.001 파일에서 오프셋 123에 위치하며 456바이트 길이입니다.</p> <p>널(NULL) LOB를 표시하려면, 크기를 -1로 입력하십시오. 크기가 0으로 지정되면, 길이가 0인 LOB로 처리됩니다. 길이가 -1인 널(NULL) LOBS의 경우, 오프셋 및 파일 이름은 무시됩니다. 예를 들어, 널(NULL) LOB의 LLS는 db2exp.001.7.-1/입니다.</p>
noheader	<p>헤더 검증 코드를 건너뛰니다(단일 파티션 데이터베이스 파티션 그룹에 있는 테이블로 조작을 로드하는 경우만 적용 가능).</p> <p>디폴트 MPP 로드(모드 PARTITION_AND_LOAD)가 단일 파티션 데이터베이스 파티션 그룹에 있는 테이블에 대해 사용되는 경우, 파일에 헤더가 없습니다. 그러므로 헤더 수정자는 필요하지 않습니다. LOAD_ONLY 모드가 사용되는 경우, 파일에 헤더가 있습니다. 헤더가 없는 수정자를 사용할 필요가 있는 유일한 환경은 헤더가 없는 파일을 사용하여 LOAD_ONLY를 수행하려고 하는 경우입니다.</p>
norowwarnings	<p>거부된 행에 대한 모든 경고를 제외시킵니다.</p>
pagefreespace= <i>x</i>	<p><i>x</i>는 0 - 100의 정수입니다. 여유 공간으로 남겨 되는 각 데이터 페이지의 백분율입니다. 최소 행 크기 때문에 지정된 값이 유효하지 않은 경우, (예를 들어, 길이가 최소한 3000바이트이고, <i>x</i> 값이 50인 행), 새 페이지에 행이 위치합니다. 100 값이 지정된 경우, 각 행은 새 페이지에 있습니다. 테이블의 PCTFREE 값은 페이지 당 지정된 여유 공간의 크기를 결정합니다. 로드 조작의 pagefreespace 값이나 테이블의 PCTFREE 값이 설정되지 않은 경우, 유틸리티는 각 페이지에 가능한 많은 스페이스를 채웁니다. pagefreespace로 설정된 값은 테이블에 지정된 PCTFREE 값을 겹쳐줍니다.</p>
rowchangetimestampignore	<p>이 수정자는 행 변경 시간소인 컬럼의 데이터가 데이터 파일에 표시되지만 무시해야 함을 로드 유틸리티에 알립니다. 이로 인해 유틸리티에 의해 모든 ROW CHANGE TIMESTAMP가 생성됩니다. GENERATED ALWAYS 및 GENERATED BY DEFAULT 컬럼 둘다의 동작이 동일합니다. GENERATED ALWAYS 컬럼의 경우, 어떤 행도 거부되지 않음을 의미합니다. 이 수정자는 rowchangetimestampmissing 또는 rowchangetimestampoverride 수정자 중 하나와 함께 사용될 수 없습니다.</p>
rowchangetimestampmissing	<p>이 수정자가 지정된 경우, 유틸리티는 입력 데이터 파일에는 행 변경 시간소인 컬럼의 데이터를 포함하지 않음(NULL 값도 포함)을 가정하므로 각 행의 값을 생성합니다. GENERATED ALWAYS 및 GENERATED BY DEFAULT 컬럼 둘 다의 동작이 동일합니다. 이 수정자는 rowchangetimestampignore 또는 rowchangetimestampoverride 수정자 중 하나와 함께 사용될 수 없습니다.</p>

표 55. 로드 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
rowchangetimestampoverride	<p>GENERATED ALWAYS로 정의된 행 변경 시간소인 컬럼이 로드되는 테이블에 있을 때에만 이 수정자를 사용해야 합니다. 해당 컬럼에 대해 명시적, 널(NULL)이 아닌 데이터를 승인하도록 유틸리티에 지시합니다(이러한 유형의 시간소인 컬럼에 대한 일반 규칙 과 반대). 테이블이 GENERATED ALWAYS로 정의되어야 할 때 다른 데이터베이스 시스템에서 데이터를 이주하거나, ROLLFORWARD DATABASE 명령에서 DROPPED TABLE RECOVERY 옵션을 사용하여 복구된 데이터에서 테이블을 로드할 때 유용합니다. 이 수정자가 사용될 때, 데이터가 없는 행이나 ROW CHANGE TIMESTAMP 컬럼의 NULL 데이터는 거부됩니다 (SQL3116W). 이 수정자는 rowchangetimestampmissing 또는 rowchangetimestampignore 수정자 중 하나와 함께 사용될 수 없습니다. 이 옵션 사용 시 로드 유틸리티는 테이블의 행 시간소인 컬럼에서 값의 고유성을 검증하거나 유지하려고 시도하지 않습니다.</p>
seclabelchar	<p>입력 소스 파일의 보안 레이블이 다폴트로 인코딩된 숫자 형식이 아니라 보안 레이블 값의 문자열 형식임을 표시합니다. LOAD는 각 보안 레이블을 로드된 대로의 내부 형식으로 변환합니다. 문자열이 적절한 형식으로 되어 있지 않은 경우 행은 로드되지 않으며 경고(SQLSTATE 01H53, SQLCODE SQL3242W)가 리턴됩니다. 문자열이 테이블을 보호하는 보안 규정 파트인 유효한 보안 레이블을 나타내지 않는 경우, 행은 로드되지 않으며 경고(SQLSTATE 01H53, SQLCODE SQL3243W)가 리턴됩니다.</p> <p>seclabelname 수정자가 지정된 경우 이 수정자를 지정할 수 없으며, 그렇지 않으면 로드에 실패하며 오류(SQLCODE SQL3525N)가 리턴됩니다.</p> <p>테이블이 단일 DB2SECURITYLABEL 컬럼으로 구성된 경우, 데이터 파일은 다음과 같습니다.</p> <pre>"CONFIDENTIAL:ALPHA:G2" "CONFIDENTIAL;SIGMA:G2" "TOP SECRET:ALPHA:G2"</pre> <p>이 데이터를 로드하거나 импорт하려면, SECLABELCHAR 파일 유형 수정자가 사용되어야 합니다.</p> <pre>LOAD FROM input.del OF DEL MODIFIED BY SECLABELCHAR INSERT INTO t1</pre>
seclabelname	<p>입력 소스 파일의 보안 레이블이 다폴트로 인코딩된 숫자 형식이 아니라 이름으로 표시됨을 나타냅니다. LOAD는 이름이 있는 경우 적절한 보안 레이블로 변환합니다. 테이블을 보호하는 보안 규정에 대해 표시된 이름이 있는 보안 레이블이 없는 경우, 행은 로드되지 않으며 경고 (SQLSTATE 01H53, SQLCODE SQL3244W)가 리턴됩니다.</p> <p>seclabelchar 수정자가 지정된 경우 이 수정자를 지정할 수 없으며, 그렇지 않으면 로드에 실패하며 오류(SQLCODE SQL3525N)가 리턴됩니다.</p> <p>테이블이 단일 DB2SECURITYLABEL 컬럼으로 구성된 경우, 데이터 파일은 다음과 유사한 보안 레이블 이름으로 구성됩니다.</p> <pre>"LABEL1" "LABEL1" "LABEL2"</pre> <p>이 데이터를 로드하거나 импорт하려면, SECLABELNAME 파일 유형 수정자가 사용되어야 합니다.</p> <pre>LOAD FROM input.del OF DEL MODIFIED BY SECLABELNAME INSERT INTO t1</pre> <p>주: 파일 유형이 ASC인 경우, 보안 레이블의 이름 다음에 오는 모든 스페이스는 이름의 일부로 해석됩니다. 이를 피하려면 stripblanks 파일 유형 수정자를 사용하여 스페이스가 제거되었는지 확인하십시오.</p>



표 55. 로드 유틸리티의 유효한 파일 유형 수정자: 모든 파일 형식 (계속)

수정자	설명
totalfreespace= <i>x</i>	<p><i>x</i>는 0 이상의 정수입니다. 이 값은 여유 공간으로 테이블의 끝에 추가되는 테이블에서 총 페이지의 백분율로 해석됩니다. 예를 들어, <i>x</i>가 20이며 데이터가 로드된 이후에 테이블에 100개의 데이터 페이지가 있는 경우, 20개의 추가적인 빈 페이지가 추가됩니다. 테이블의 총 데이터 페이지 수는 120입니다. 데이터 페이지 총 수는 테이블에서 인덱스 페이지의 수를 계산에 넣지 않습니다. 이 옵션은 인덱스 오브젝트에 영향을 미치지 않습니다. 이 옵션이 지정된 두 개의 로드가 완료되면, 두 번째 로드는 첫 번째 로드에서 끝에 추가한 여유 공간을 재사용하지 않습니다.</p>
usedefaults	<p>목표 테이블 컬럼의 소스 컬럼이 지정되었지만 하나 이상의 행 인스턴스에 대한 데이터를 포함하지 않은 경우, 디폴트값이 로드됩니다. 누락된 데이터의 예:</p> <ul style="list-style-type: none"> <li>• DEL 파일: 임의 수의 스페이스(" ", ",")로 구분되는 2개의 인접 컬럼 분리문자나 2개의 인접 컬럼 분리문자(", ", ",")가 컬럼 값으로 지정됩니다.</li> <li>• DEL/ASC/WSF 파일: 행의 컬럼이 충분하지 않거나 행의 길이가 원래 스펙만큼 충분하지 않습니다. ASC 파일의 경우, NULL 컬럼 값은 명시적으로 누락된 것으로 간주되지 않으며 디폴트가 NULL 컬럼 값을 대신하지 않습니다. 숫자, 날짜, 시간 및 /시간소인 컬럼의 모든 공백 문자로 NULL 컬럼 값을 표시하거나, 컬럼이 NULL임을 표시하기 위해 모든 유형의 컬럼에 널(NULL) 표시기를 사용함으로써 NULL 컬럼 값을 표시합니다.</li> </ul> <p>이 옵션이 없는 경우, 소스 컬럼이 행 인스턴스의 데이터를 포함하지 않으면 다음 중 하나가 발생합니다.</p> <ul style="list-style-type: none"> <li>• DEL/ASC/WSF 파일: 컬럼에 널(NULL) 입력 가능한 경우 NULL이 로드됩니다. 컬럼이 널(NULL) 입력 가능하지 않으면 유틸리티가 행을 거부합니다.</li> </ul>

표 56. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL)

수정자	설명
codepage= <i>x</i>	<p><i>x</i>는 ASCII 문자열입니다. 해당 값은 입력 데이터 세트에서 데이터의 코드 페이지로서 해석됩니다. 로드 조작 중 이 코드 페이지에서 데이터베이스 코드 페이지로 문자 데이터(및 문자로 지정된 숫자 데이터)를 변환합니다.</p> <p>다음 규칙이 적용됩니다.</p> <ul style="list-style-type: none"> <li>• 순수 DBCS(그래픽), 혼합 DBCS 및 EUC의 경우, 분리문자는 x00 - x3F의 범위로 제한됩니다.</li> <li>• EBCDIC 코드 페이지에서 지정된 DEL 데이터의 경우, 분리문자는 shift-in 및 shift-out DBCS 문자와 일치하지 않을 수 있습니다.</li> <li>• nullindchar은 x20 - x7F 코드 포인트의 표준 ASCII 세트에 포함된 기호를 지정해야 합니다. 이것은 ASCII 기호 및 코드 포인트를 나타냅니다. 코드 포인트가 달라도 EBCDIC 데이터는 해당 기호를 사용할 수 있습니다.</li> </ul> <p>이 옵션은 CURSOR 파일 유형과 함께 지원되지 않습니다.</p>

표 56. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
dateformat="x"	<p>x는 소스 파일에서 날짜의 형식입니다.<sup>1</sup> 유효한 날짜 요소는 다음과 같습니다.</p> <p>YYYY - Year(0000 - 9999 사이의 4자리 숫자)  M - Month(1 - 12 사이의 1 또는 2자리 숫자)  MM - Month(1 - 12 사이의 2자리 숫자,  M과 상호 배타적)  D - Day(1 - 31 사이의 1 또는 2자리 숫자)  DD - Day(1 - 31 사이의 2자리 숫자,  D와 상호 배타적)  DDD - Day of the year(001 - 366 사이의 3자리 숫자,  다른 day 또는 month 요소와 상호 배타적)</p> <p>지정되지 않은 각 요소에 대해 디폴트값 1이 지정됩니다. 날짜 형식의 예:</p> <p>"D-M-YYYY"  "MM.DD.YYYY"  "YYYYDDD"</p>
dumpfile = x	<p>x는 거부된 행이 쓰여진 예외 파일의 완전한 이름(서버 데이터베이스 파티션에 따른)입니다. 레코드당 최대 32KB의 데이터가 쓰여집니다. 다음은 덤프 파일을 지정하는 방법을 표시하는 예입니다.</p> <pre>db2 load from data of del modified by dumpfile = /u/user/filename insert into table_name</pre> <p>인스턴스 소유자가 파일을 작성하고 소유합니다. 디폴트 파일 권한을 겹쳐쓰려면, dumpfileaccessall 파일 유형 수정자를 사용하십시오.</p> <p>주:</p> <ol style="list-style-type: none"> <li>1. 파티션된 데이터베이스 환경에서, 경로는 로딩 데이터베이스 파티션에 로컬이어야 하므로 현재 실행 중인 로드 조작은 동일한 파일에 쓰지 않습니다.</li> <li>2. 파일의 콘텐츠는 비동기 버퍼 지정 모드로 디스크에 쓰여집니다. 실패하거나 인터럽트된 로드 조작의 경우, 디스크에 커밋된 레코드 수는 확실히 알 수 없으며 LOAD RESTART 이후에 일관성을 보장할 수 없습니다. 파일에서는 단일 패스로 시작하고 완료하는 로드 조작에 대해서만 완료될 것으로 가정합니다.</li> <li>3. 지정된 파일이 이미 존재하는 경우, 재작성되지 않지만 추가됩니다.</li> </ol>
dumpfileaccessall	<p>덤프 파일 작성 시 'OTHERS'에 대한 읽기 액세스를 권한 부여합니다.</p> <p>이 파일 유형 수정자는 다음 경우에만 유효합니다.</p> <ol style="list-style-type: none"> <li>1. dumpfile 파일 유형 수정자와 결합하여 사용됩니다.</li> <li>2. 사용자는 로드 목표 테이블에 대한 SELECT 특권을 갖습니다.</li> <li>3. UNIX 운영 체제에 있는 DB2 서버 데이터베이스 파티션에서 발행됩니다.</li> </ol> <p>지정된 파일이 이미 존재하는 경우, 권한을 변경하지 못합니다.</p>
fastparse	<p>주의와 함께 사용합니다. 사용자 제공 컬럼 값에 대한 구문 검사를 줄이고, 성능을 향상시킵니다. 테이블은 구조적으로 옳은 것으로 보장되지만(유틸리티는 충분한 데이터 검사를 수행하여 분할 위반 또는 트랩을 방지함), 데이터 일관성의 유효성은 확인되지 않습니다. 데이터가 일관되고 올바름을 확신하는 경우에만 이 옵션을 사용하십시오. 예를 들어, 사용자 제공 데이터가 유효하지 않은 시간소인 컬럼 값 :1&gt;0-00-20-07.11.12.000000을 포함하는 경우, FASTPARSE가 지정되면 이 값이 테이블에 삽입되며 FASTPARSE가 지정되지 않으면 거부됩니다.</p>



표 56. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
implieddecimal	<p>내포된 소수점의 위치는 컬럼 정의로 판별되며, 값의 끝으로 가정되지 않습니다. 예를 들어, 12345 값은 12345.00이 아닌 123.45로 DECIMAL(8,2) 컬럼에 로드됩니다.</p> <p>이 수정자는 packeddecimal 수정자와 함께 사용될 수 없습니다.</p>
timeformat="x"	<p>x는 소스 파일에서 시간의 형식입니다.<sup>1</sup> 유효한 시간 요소는 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>H - Hour(12시간 시스템의 경우 0 - 12 범위의 0 - 12 범위의 2자리 숫자, 24시간 시스템의 경우 0 - 24)</li> <li>HH - Hour(12시간 시스템의 경우 0 - 12 범위의 2자리 숫자, 24시간 시스템의 경우 0 - 24. H와 상호 배타적)</li> <li>M - Minute(0 - 59 사이의 1 또는 2자리 숫자)</li> <li>MM - Minute(0 - 59 사이의 2자리 숫자, M과 상호 배타적)</li> <li>S - Second(0 - 59 사이의 1 또는 2자리 숫자)</li> <li>SS - Second(0 - 59 사이의 2자리 숫자, S와 상호 배타적)</li> <li>SSSSS - Second of the day after midnight(00000 - 86399 사이의 5자리 숫자, 다른 시간 요소와 상호 배타적)</li> <li>TT - Meridian indicator(AM 또는 PM)</li> </ul> <p>지정되지 않은 각 요소에 대해 디폴트값 0이 지정됩니다. 시간 형식의 예:</p> <p>"HH:MM:SS"  "HH.MM TT"  "SSSSS"</p>
timestampformat="x"	<p>x는 소스 파일에서 시간소인의 형식입니다.<sup>1</sup> 유효한 시간소인 요소는 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>YYYY - Year(0000 - 9999 사이의 4자리 숫자)</li> <li>M - Month(1 - 12 사이의 1 또는 2자리 숫자)</li> <li>MM - Month(01 - 12 사이의 2자리 숫자, M 및 MM은 상호 배타적)</li> <li>MMM - Month(월 이름으로 사용되는 3자리 문자로 대소문자를 구분하지 않으며, M 및 MM은 상호 배타적)</li> <li>D - Day(1 - 31 사이의 1 또는 2자리 숫자)</li> <li>DD - Day(1 - 31 사이의 2자리 숫자, D와는 상호 배타적)</li> <li>DDD - Day of the year(001 - 366 사이의 3자리 숫자, 다른 day 또는 month 요소와 상호 배타적)</li> <li>H - Hour(12시간 시스템의 경우 0 - 12 범위의 1 또는 2자리 숫자, 24시간 시스템의 경우 0 - 24)</li> <li>HH - Hour(12시간 시스템의 경우 0 - 12 범위의 2자리 숫자, 24시간 시스템의 경우 0 - 24. H와는 상호 배타적)</li> <li>M - Minute(0 - 59 사이의 1 또는 2자리 숫자)</li> <li>MM - Minute(0 - 59 사이의 2자리 숫자, M, minute와 상호 배타적)</li> <li>S - Second(0 - 59 사이의 1 또는 2자리 숫자)</li> <li>SS - Second(0 - 59 사이의 2자리 숫자, S와 상호 배타적)</li> <li>SSSSS - Second of the day after midnight(00000 - 86399 사이의 5자리 숫자, 다른 시간 요소와 상호 배타적)</li> <li>U(1 - 12회) - Fractional seconds(U 어커런스 수는 각 자리가 0 - 9 사이의 자리 수를 나타냄)</li> <li>TT - Meridian indicator(AM 또는 PM)</li> </ul>

표 56. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
timestampformat="x"(계속)	<p>다폴트값 1이 미지정된 YYYY, M, MM, D, DD 또는 DDD 요소에 지정됩니다. 다폴트값 'Jan'이 미지정된 MMM 요소에 지정됩니다. 미지정된 다른 모든 요소에 다폴트값 0이 지정됩니다. 다음은 시간소인 형식의 예입니다.</p> <p style="text-align: center;">"YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>MMM 요소의 올바른 값은 다음을 포함합니다. 'jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov' 및 'dec'. 이들 값은 대소문자가 구분되지 않습니다.</p> <p>TIMESTAMPFORMAT 수정자가 지정되지 않으면, 로드 유틸리티는 두 개의 가능한 형식 중 하나를 사용하여 시간소인 필드를 형식화합니다.</p> <p>YYYY-MM-DD-HH.MM.SS YYYY-MM-DD HH:MM:SS</p> <p>로드 유틸리티는 DD와 HH 간의 구분자를 확인하여 형식을 선택합니다. 대시 '-'인 경우, 로드 유틸리티는 일반 대시와 점 형식(YYYY-MM-DD-HH.MM.SS)을 사용합니다. 공백인 경우, 로드 유틸리티는 HH, MM 및 SS를 구분하기 위해 콜론 ':'을 예상합니다.</p> <p>어느 쪽 형식이든, 마이크로초 필드(UUUUUU)를 포함하는 경우, 로드 유틸리티는 점 '.'을 구분자로 예상합니다. YYYY-MM-DD-HH.MM.SS.UUUUUU 또는 YYYY-MM-DD HH:MM:SS.UUUUUU를 승인할 수 있습니다.</p> <p>다음 예는 사용자 정의 날짜 및 시간 형식을 포함하는 데이터를 schedule이라는 테이블로 로드하는 방법을 설명합니다.</p> <pre>db2 load from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>
usegraphiccodepage	<p>usegraphiccodepage가 제공되면, 그래픽 또는 2바이트 문자 대형 오브젝트(DBCLOB) 데이터 필드로 로드될 데이터는 그래픽 코드 페이지에 있는 것으로 가정합니다. 나머지 데이터는 문자 코드 페이지에 있다고 가정합니다. 그래픽 코드 페이지는 문자 코드 페이지와 연관됩니다. LOAD는 codepage 수정자가 지정된 경우 이를 통해 문자 코드 페이지를 판별하고 codepage 수정자가 지정되지 않은 경우 데이터베이스의 코드 페이지를 통해 문자 코드 페이지를 판별합니다.</p> <p>복구 중인 테이블에 그래픽 데이터가 있는 경우에만 삭제(drop) 테이블 복구로 생성되는 구분된 데이터 파일과 결합하여 이 수정자를 사용해야 합니다.</p> <p><b>제한사항</b></p> <p>이들 파일이 오직 하나의 코드 페이지에 인코딩된 데이터를 포함하면, usegraphiccodepage 수정자는 EXPORT 유틸리티로 작성된 DEL 파일과 함께 지정되지 않아야 합니다. usegraphiccodepage 수정자는 파일의 2바이트 문자 대형 오브젝트(DBCLOB)에서 무시됩니다.</p>

표 56. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(ASC/DEL) (계속)

수정자	설명
xmlchar	<p>XML 문서가 문자 코드 페이지에서 인코딩됨을 지정합니다.</p> <p>이 옵션은 지정된 문자 코드 페이지에서 인코딩되었지만 인코딩 선언을 포함하지 않는 XML 문서 처리에 유용합니다.</p> <p>각 문서에서 선언 태그가 존재하고 인코딩 속성을 포함하는 경우 인코딩은 문자 코드 페이지와 일치해야 하며, 그렇지 않으면 문서를 포함하는 행이 거부됩니다. 문자 코드 페이지는 codepage 파일 유형 수정자에서 지정한 값이거나, 지정되지 않은 경우 응용프로그램 코드 페이지입니다. 디폴트로 문서가 유니코드로 인코딩되거나 인코딩 속성을 가진 선언 태그를 포함합니다.</p>
xmlgraphic	<p>XML 문서가 지정된 그래픽 코드 페이지에서 인코딩됨을 지정합니다.</p> <p>이 옵션은 특정 그래픽 코드 페이지에서 인코딩되었지만 인코딩 선언을 포함하지 않는 XML 문서 처리에 유용합니다.</p> <p>각 문서에서 선언 태그가 존재하고 인코딩 속성을 포함하는 경우 인코딩은 그래픽 코드 페이지와 일치해야 하며, 그렇지 않으면 문서를 포함하는 행이 거부됩니다. 그래픽 코드 페이지는 codepage 파일 유형 수정자에서 지정한 값의 그래픽 구성요소이거나, 지정되지 않은 경우 응용프로그램 코드 페이지의 그래픽 구성요소입니다. 디폴트로 문서가 유니코드로 인코딩되거나 인코딩 속성을 가진 선언 태그를 포함합니다.</p>

표 57. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(구분되지 않은 ASCII)

수정자	설명
binarynumerics	<p>숫자(10진수는 아님) 데이터는 문자 표시가 아닌 2진 양식이어야 합니다. 손실이 큰 변환은 피합니다.</p> <p>이 옵션은 recLen 옵션으로 지정된 고정 길이 레코드를 사용하며 위치상 ASC에서만 지원됩니다.</p> <p>다음 규칙이 적용됩니다.</p> <ul style="list-style-type: none"> <li>• 데이터 유형 간의 변환은 수행되지 않으며, BIGINT, INTEGER 및 SMALLINT 예외가 있습니다.</li> <li>• 데이터 길이는 목표 컬럼 정의와 일치해야 합니다.</li> <li>• FLOAT는 IEEE 부동 소수점 형식이어야 합니다.</li> <li>• 로드 소스 파일의 2진 데이터는 로드 조작이 실행되는 플랫폼에 관계 없이 빅 엔디안(big-endian)으로 가정됩니다.</li> </ul> <p>이 수정자에 의해 영향을 받는 컬럼의 데이터에 NULL 값을 표시할 수 없습니다. 이 수정자가 사용될 때 공백(보통 NULL로 해석)은 2진 값으로 해석됩니다.</p>
nochecklengths	<p>nochecklengths가 지정되면, 목표 테이블 컬럼의 크기를 초과하는 컬럼 정의가 소스 데이터에 있는 경우에도 각 행을 로드하려고 시도합니다. 코드 페이지 변환으로 소스 데이터가 축소되는 경우 그러한 행은 정상적으로 로드될 수 있습니다. 예를 들어, 소스에서 4바이트 EUC 데이터는 목표에서 2바이트 DBCS 데이터로 축소되어 반 정도의 스페이스가 필요합니다. 불일치 컬럼 정의에도 불구하고 소스 데이터가 모든 경우에 맞는 경우 특히 이 옵션이 유용합니다.</p>

표 57. 로드 유틸리티의 유효한 파일 유형 수정자: ASCII 파일 형식(구분되지 않은 ASCII) (계속)

수정자	설명
nullindchar=x	<p>x는 단일 문자입니다. 널(NULL) 값을 나타내는 문자를 x로 변경합니다. x의 디폴트값은 Y입니다. <sup>2</sup></p> <p>문자가 영문자인 경우를 제외하고 EBCDIC 데이터 파일의 경우 이 수정자의 대소문자를 구분합니다. 예를 들어, 널(NULL) 표시기 문자가 N 문자가 되도록 지정되는 경우, n은 널(NULL) 표시기로 인식됩니다.</p>
packeddecimal	<p>binarynumerics 수정자가 DECIMAL 필드 유형을 포함하지 않으므로 압축 10진수 데이터를 직접 로드합니다.</p> <p>이 옵션은 reclen 옵션으로 지정된 고정 길이 레코드를 사용하며 위치상 ASC에서만 지원됩니다.</p> <p>지원되는 부호 니블의 값은 다음과 같습니다.</p> <p style="margin-left: 40px;">+ = 0xC 0xA 0xE 0xF - = 0xD 0xB</p> <p>이 수정자에 의해 영향을 받는 컬럼의 데이터에 NULL 값을 표시할 수 없습니다. 이 수정자가 사용될 때 공백(보통 NULL로 해석)은 2진 값으로 해석됩니다.</p> <p>서버 플랫폼에 관계 없이, 로드 소스 파일의 2진 데이터의 바이트 순서는 빅 엔디안(big-endian)으로 가정되므로, Windows 운영 체제에서 이 수정자를 사용하면 바이트 순서는 바뀌지 않아야 합니다.</p> <p>이 수정자는 implieddecimal 수정자와 함께 사용될 수 없습니다.</p>
reclen=x	<p>x는 최대값이 32 767인 정수입니다. 각 행에 대해 x 문자가 읽히며 행의 끝을 표시하기 위한 줄 바꾸기 문자는 사용되지 않습니다.</p>
striptblanks	<p>데이터를 변수 길이 필드로 로드할 때 뒤 공백을 절단합니다. 이 옵션이 지정되지 않으면, 공백이 보존됩니다.</p> <p>이 옵션은 striptnulls와 함께 지정될 수 없습니다. 이들은 상호 배타적 옵션입니다. 이 옵션은 이전 호환성을 위해서만 지원되는 사용 안하는 t 옵션을 교체합니다.</p>
striptnulls	<p>데이터를 변수 길이 필드로 로드할 때 뒤 NULL 값(0x00 문자)을 절단합니다. 이 옵션이 지정되지 않으면, NULL 값이 보존됩니다.</p> <p>이 옵션은 striptblanks와 함께 지정될 수 없습니다. 이들은 상호 배타적 옵션입니다. 이 옵션은 이전 호환성을 위해서만 지원되는 사용 안하는 padwithzero 옵션을 교체합니다.</p>
zoneddecimal	<p>BINARYNUMERICS 수정자는 DECIMAL 필드 유형을 포함하지 않으므로 존 10진수(zoned DECIMAL) 데이터를 로드합니다. 이 옵션은 RECLen 옵션으로 지정된 고정 길이 레코드를 사용하며 위치상 ASC에서만 지원됩니다.</p> <p>반 바이트 기호 값은 다음 중 하나일 수 있습니다.</p> <p style="margin-left: 40px;">+ = 0xC 0xA 0xE 0xF - = 0xD 0xB</p> <p>지원되는 숫자 값은 0x0 - 0x9입니다.</p> <p>지원되는 존 값은 0x3 및 0xF입니다.</p>

표 58. 로드 유틸리티의 유효한 파일 유형 수정자: DEL 파일 형식(컬럼 식별자가 있는 ASCII)

수정자	설명
chardelx	<p>x는 단일 문자열 분리문자입니다. 디폴트값은 큰따옴표(")입니다. 큰따옴표 대신 지정된 문자를 사용하여 문자열을 묶습니다. <sup>23</sup> 명시적으로 큰따옴표(")를 문자열 분리문자로 지정하려는 경우 다음과 같이 지정해야 합니다.</p> <pre>modified by chardel""</pre> <p>다음과 같이 작은따옴표(')를 문자열 분리문자로 지정할 수도 있습니다.</p> <pre>modified by chardel''</pre>
coldelx	<p>x는 단일 문자 컬럼 분리문자입니다. 디폴트값은 쉼표(.)입니다. 지정된 문자가 쉼표 대신 사용되어 컬럼 끝을 나타냅니다.<sup>23</sup></p>
decplusblank	<p>플러스 부호 문자. 플러스 부호(+) 대신 공백을 양의 10진수 값 앞에 붙입니다. 디폴트 조치는 플러스 부호를 양의 10진수 값 앞에 붙이는 것입니다.</p>
decptx	<p>x는 소수점 문자로 마침표를 대신하는 단일 문자입니다. 디폴트값은 마침표(.)입니다. 마침표 대신 지정된 문자가 소수점 문자로 사용됩니다.<sup>23</sup></p>
delprioritychar	<p>분리문자의 현재 디폴트 우선순위: 레코드 구분 문자, 문자 분리문자, 컬럼 분리문자. 이 수정자는 분리문자 우선순위를 문자 분리문자, 레코드 구분 문자, 컬럼 분리문자로 되돌림으로써 이전 우선순위에 따른 기존 응용프로그램을 보호합니다. 구분:</p> <pre>db2 load ... modified by delprioritychar ...</pre> <p>예를 들어, 다음과 같은 DEL 데이터 파일이 있습니다.</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 수정자를 지정하고, 이 데이터 파일에는 두 개의 행만이 있습니다. 두 번째 &lt;행 분리문자&gt;는 두 번째 행의 첫 번째 데이터 컬럼의 파트로 해석되지만, 첫 번째 및 세 번째 &lt;행 분리문자&gt;는 실제 레코드 구분 문자로 해석됩니다. 이 수정자가 지정되지 않은 경우, 이 데이터 파일에는 세 개의 행이 있으며, 각 행은 &lt;행 분리문자&gt;로 구분됩니다.</p>
keepblanks	<p>유형 CHAR, VARCHAR, LONG VARCHAR 또는 CLOB의 각 필드에 앞뒤 공백을 둡니다. 이 옵션이 없으면, 문자 분리문자 내에 없는 앞 공백과 뒤 공백은 모두 제거되며 공백 필드의 테이블로 NULL이 삽입됩니다.</p> <p>다음 예는 데이터 파일에서 앞과 뒤의 모든 공백을 보존하면서 TABLE1이라는 테이블로 데이터를 로드하는 방법을 설명합니다.</p> <pre>db2 load from delfile3 of del modified by keepblanks insert into table1</pre>
nochardel	<p>로드 유틸리티는 컬럼 분리문자 간의 모든 바이트를 컬럼 데이터의 파트로 가정합니다. 문자 분리문자는 컬럼 데이터의 파트로 구분 분석됩니다. DB2를 사용하여 데이터를 익스포트한 경우(익스포트 시 nochardel을 지정한 경우를 제외하고) 이 옵션을 지정하지 말아야 합니다. 이 옵션은 문자 분리문자가 없는 벤더 데이터 파일을 지원하기 위해 제공됩니다. 부적절한 사용은 데이터 유실이나 손상을 초래할 수 있습니다.</p> <p>이 옵션은 chardelx, delprioritychar 또는 nodoubledel과 함께 지정될 수 없습니다. 이들은 상호 배타적 옵션입니다.</p>
nodoubledel	<p>2바이트 분리문자를 인식하지 않습니다.</p>

표 59. 로드 유틸리티의 유효한 파일 유형 수정자: IXF 파일 형식

수정자	설명
forcein	코드 페이지 불일치와 관계없이 데이터를 승인하고 코드 페이지 간에 변환하지 않도록 유틸리티에 지시합니다.  데이터의 고정 길이 대상 필드가 충분히 크지 검증하도록 고정 길이 대상 필드를 점검합니다. nochecklengths가 지정된 경우, 검사가 수행되지 않으며 각 행을 로드하려고 시도합니다.
nochecklengths	nochecklengths가 지정되면, 목표 테이블 컬럼의 크기를 초과하는 컬럼 정의가 소스 데이터에 있는 경우에도 각 행을 로드하려고 시도합니다. 코드 페이지 변환으로 소스 데이터가 축소되는 경우 그러한 행은 정상적으로 로드될 수 있습니다. 예를 들어, 소스에서 4바이트 EUC 데이터는 목표에서 2바이트 DBCS 데이터로 축소되어 반 정도의 스페이스가 필요합니다. 불일치 컬럼 정의에도 불구하고 소스 데이터가 모든 경우에 맞는 경우 특히 이 옵션이 유용합니다.

주:

1. 날짜 출력 문자열을 둘러싼 큰따옴표는 필수입니다. 필드 구분자는 a - z, A - Z 및 0 - 9를 포함할 수 없습니다. 필드 구분자는 DEL 파일 형식의 필드 분리문자나 문자 분리문자와 같지 않아야 합니다. 요소의 시작 및 종료 위치가 명확한 경우 필드 구분자가 선택적입니다. D, H, M 또는 S와 같은 요소가 사용되는 경우(수정자에 따라) 항목의 변수 길이 때문에 모호함이 있을 수 있습니다.

시간소인 형식의 경우, 문자 M을 사용하는 month 및 minute 디스크립터 간의 모호함을 피하려면 주의해야 합니다. month 필드는 다른 날짜 필드와 인접해야 합니다. minute 필드는 다른 시간 필드와 인접해야 합니다. 다음은 앰비규어스 시간소인 형식입니다.

```
"M"(month 또는 minute일 수 있음)
"M:M"(month 및 minute 구분 가능?)
"M:YYYY:M"(둘 다 month로 해석됨.)
"S:M:YYYY"(둘 다 시간 값 및 날짜 값에 인접)
```

앰비규어스 경우, 유틸리티는 오류 메시지를 발행하며, 조작에 실패합니다.

다음은 명확한 시간소인 형식입니다.

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

큰따옴표 및 백슬래시와 같은 일부 문자는 Escape 문자(예: #)가 앞에 와야 합니다.

2. chardel, coldel 또는 decpt 파일 유형에 제공되는 문자 값은 소스 데이터의 코드 페이지에 지정되어야 합니다.

문자 코드 포인트(문자 기호 대신)는 구문 xJJ 또는 0xJJ를 사용하여 지정될 수 있으며, 여기서 JJ는 코드 포인트의 16진수를 나타냅니다. 예를 들어, 컬럼 분리문자로 # 문자를 지정하려면, 다음 중 하나를 사용하십시오.

```
... modified by coldel# ...
... modified by coldel0x23 ...
... modified by coldelX23 ...
```

3. 데이터 이동을 위한 분리문자 고려사항은 분리문자를 겹쳐쓰기할 때 사용할 수 있는 문자에 적용하는 제한사항을 나열합니다.
4. MODIFIED BY 옵션과 함께 지원되지 않는 파일 유형을 사용하려고 시도하는 경우 로드 유틸리티는 경고를 발행하지 않습니다. 이런 경우, 로드 조작에 실패하며 오류 코드가 리턴됩니다.
5. 내재적으로 숨겨진 행 변경 시간소인 컬럼을 포함하는 테이블로 импорт할 때, 내재적으로 숨겨진 컬럼의 등록 정보는 무시됩니다. 그러므로, 컬럼의 데이터가 импорт되는 데이터에 표시되지 않으며 명시적 컬럼 목록이 표시되지 않으면 rowchangetimestampmissing 파일 유형 수정자가 импорт 명령에 지정되어야 합니다.

표 60. codepage 및 usegraphiccodepage 사용 시 LOAD 동작

codepage=N	usegraphiccodepage	LOAD 동작
Absent	Absent	CLIENT 옵션이 지정된 경우에도 파일의 모든 데이터는 응용프로그램 코드 페이지가 아닌 데이터베이스 코드 페이지에 있는 것으로 가정됩니다.
Present	Absent	파일의 모든 데이터는 코드 페이지 N에 있는 것으로 가정됩니다.  경고: N이 1바이트 코드 페이지이면 데이터베이스로 로드될 때 그래픽 데이터가 손상됩니다.
Absent	Present	CLIENT 옵션이 지정된 경우에도 파일의 문자 데이터는 데이터베이스 코드 페이지에 있는 것으로 가정됩니다. CLIENT 옵션이 지정된 경우에도 그래픽 데이터는 데이터베이스 그래픽 데이터의 코드 페이지에 있는 것으로 가정됩니다.  데이터베이스 코드 페이지가 1바이트이면, 모든 데이터는 데이터베이스 코드 페이지에 있는 것으로 가정됩니다.  경고: 1바이트 데이터베이스로 로드될 때 그래픽 데이터가 손상됩니다.
Present	Present	문자 데이터는 코드 페이지 N에 있는 것으로 가정됩니다. 그래픽 데이터는 N의 그래픽 코드 페이지에 있는 것으로 가정됩니다.  N이 1바이트 또는 2바이트 코드 페이지인 경우, 모든 데이터는 코드 페이지 N에 있는 것으로 가정됩니다.  경고: N이 1바이트 코드 페이지이면 데이터베이스로 로드될 때 그래픽 데이터가 손상됩니다.



## ADMIN\_CMD 프로시저를 사용하는 PRUNE HISTORY/LOGFILE 명령

복구 실행기록 파일에서 항목을 삭제하거나 현재 연결된 데이터베이스 파티션의 사용 중인 로그 파일 경로에서 로그 파일을 삭제하는 데 사용됩니다. 복구 실행기록 파일이 과도하게 크고 보유 기간이 긴 경우 이 파일에서 항목을 삭제할 필요가 있을 수 있습니다.

### 권한 부여

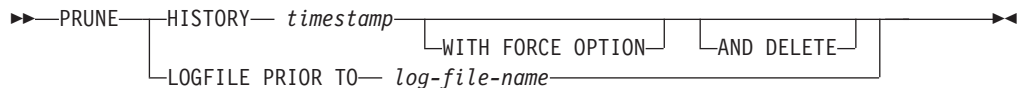
다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

### 필수 연결

데이터베이스

### 명령 구문



### 명령 매개변수

#### HISTORY timestamp

삭제될 복구 실행기록 파일의 항목 범위를 식별합니다. 전체 시간소인 (yyyymmddhhmmss 형식) 또는 맨 앞의 접두부(최소한 yyyy)를 지정할 수 있습니다. 제공된 시간소인 이하의 시간소인을 갖는 모든 항목은 복구 실행기록 파일에서 삭제됩니다.

#### WITH FORCE OPTION

가장 최근의 리스토어 세트의 일부 항목이 파일에서 삭제되었더라도 지정된 시간소인에 따라 항목이 프론되도록 지정합니다. 리스토어 세트는 해당 백업 이미지의 모든 리스토어를 포함하는 가장 최근의 전체 데이터베이스 백업입니다. 이 매개변수를 지정하지 않으면 정방향 백업 이미지의 모든 항목을 실행기록에 유지보수합니다.

#### AND DELETE

실행기록 파일 항목이 제거될 때 연관된 로그 아카이브가 실제로 삭제되도록(위치 정보에 따라) 지정합니다. 이 옵션은 로그 아카이브가 더 이상 필요하지 않



을 때 아카이브 스토리지 스페이스가 복구되는지 확인하는 데 특히 유용합니다. User Exit 프로그램을 통해 로그를 아카이브 중인 경우, 이 옵션을 사용하여 로그를 삭제할 수 없습니다.

**auto\_del\_rec\_obj** 데이터베이스 구성 매개변수를 ON으로 설정하면, AND DELETE 매개변수로 PRUNE HISTORY를 호출 시 실행기록 파일 항목이 프룬(prune)된 경우 실제로 백업 이미지를 삭제하고 복사 이미지를 로드합니다.

#### **LOGFILE PRIOR TO** *log-file-name*

로그 파일 이름의 문자열을 지정합니다(예: S0000100.LOG). 지정된 로그 파일 이전의 모든 로그 파일(지정된 로그 파일은 제외함)을 삭제합니다. **logretain** 데이터베이스 구성 매개변수는 RECOVERY 또는 CAPTURE로 설정해야 합니다.

#### **예 :**

예 1: 2003년 12월 3일 또는 그 이전에 작성된 복구 실행기록 파일에서 모든 항목을 제거하십시오.

```
CALL SYSPROC.ADMIN_CMD ('prune history 20031231')
```

예 2: 사용 중인 로드 파일 경로에서 S0000100.LOG 이전의 모든 로그 파일 (S0000100.LOG는 제외)을 삭제하십시오.

```
CALL SYSPROC.ADMIN_CMD('prune logfile prior to S0000100.LOG')
```

#### **사용 시 참고사항**

WITH FORCE OPTION을 사용할 경우, 데이터베이스의 자동 리스토어에 필요한 항목을 삭제할 수 있습니다. 직접 리스토어는 계속 올바르게 작동합니다. 이 명령을 사용하면 db2ckrst 유틸리티가 필수 백업 이미지의 전체 체인을 정확히 분석하지 못하게 할 수 있습니다. WITH FORCE OPTION을 사용하지 않고 PRUNE HISTORY 명령을 사용하면 필요한 항목이 삭제되지 않게 합니다.

상태가 DB2HISTORY\_STATUS\_DO\_NOT\_DELETE인 항목은 프룬되지 않습니다. WITH FORCE OPTION이 사용되면, DB2HISTORY\_STATUS\_DO\_NOT\_DELETE 로 표시된 오브젝트는 여전히 프룬되거나 삭제됩니다. UPDATE HISTORY 명령, UPDATE\_HISTORY가 있는 ADMIN\_CMD 또는 db2HistoryUpdate API를 사용하여 복구 실행기록 파일 항목 상태를 DB2HISTORY\_STATUS\_DO\_NOT\_DELETE로 설정할 수 있습니다. DB2HISTORY\_STATUS\_DO\_NOT\_DELETE 상태를 사용하여 키 복구 실행기록 파일 항목이 프룬되지 않게 하고 연관된 복구 오브젝트가 삭제되지 않게 할 수 있습니다.

PRUNE HISTORY 명령을 사용하여 스냅샷 백업 데이터베이스 실행기록 파일 항목을 프룬할 수 있지만 AND DELETE 매개변수를 사용하여 관련된 실제 복구 오브젝트를 삭제할 수 없습니다. 스냅샷 백업 오브젝트를 삭제하는 유일한 방법은 db2acsutil 명령을 사용하는 것입니다.

이 명령은 응용프로그램이 현재 연결되어 있는 데이터베이스 파티션에만 영향을 줍니다.

## ADMIN\_CMD 프로시저를 사용하는 QUIESCE DATABASE 명령

강제로 모든 사용자가 지정된 데이터베이스를 해제하고 이를 Quiesce 모드로 넣습니다. 데이터베이스가 Quiesce 모드에 있는 동안 이에서 관리 태스크를 수행할 수 있습니다. 관리 태스크가 완료된 후, 데이터베이스를 활성화하고 다른 사용자가 데이터베이스를 종료한 후 다른 데이터베이스 시작을 수행할 필요없이 데이터베이스에 연결할 수 있도록 하려면 UNQUIESCE 명령을 사용하십시오.

이 모드에서는 이 제한된 모드에서 권한이 있는 사용자만 데이터베이스에 연결할 수 있습니다. *sysadm* 및 *dbadm* 권한이 있는 사용자는 Quiesce 상태에 있는 데이터베이스에 항상 액세스할 수 있습니다.

### 범위

QUIESCE DATABASE의 결과는 데이터베이스에 있는 모든 오브젝트가 Quiesce 모드가 되는 것으로 나타납니다. 허용된 사용자/그룹 및 *sysadm*, *sysmaint*, *dbadm* 또는 *sysctrl*만 데이터베이스 또는 해당 오브젝트에 액세스할 수 있습니다.

### 권한 부여

다음 중 하나가 필요합니다.

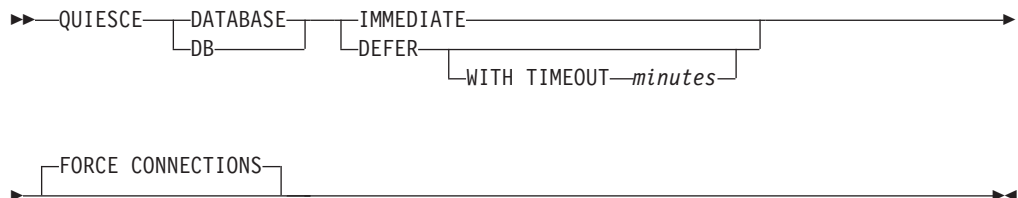
데이터베이스 레벨 Quiesce의 경우:

- *sysadm*
- *dbadm*

### 필수 연결

데이터베이스

### 명령 구문



### 명령 매개변수

#### DEFER

응용프로그램이 현재 작업 단위를 커밋할 때까지 대기하십시오.

### WITH TIMEOUT *minutes*

응용프로그램이 현재 작업 단위를 커밋할 때까지 대기할 시간을 분 단위로 지정합니다. 단일 파티션 데이터베이스 환경에서 값을 지정하지 않을 경우 디폴트값은 10분입니다. 파티션된 데이터베이스 환경에서는 **start\_stop\_time** 데이터베이스 관리 프로그램 구성 매개변수에 의해 지정된 값을 사용합니다.

### IMMEDIATE

트랜잭션이 커밋될 때까지 대기하지 말고 즉시 트랜잭션을 롤백하십시오.

### FORCE CONNECTIONS

연결을 강제로 해제하십시오.

### DATABASE

데이터베이스를 Quiesce하십시오. 데이터베이스의 모든 오브젝트가 Quiesce 모드에 넣어집니다. *sysadm*, *sysmaint* 및 *sysctrl* 권한이 있는 사용자 및 지정된 그룹의 지정된 사용자만 데이터베이스 또는 해당 오브젝트에 액세스할 수 있습니다.

### 예 :

데이터베이스에 연결된 모든 사용자를 강제로 해제합니다.

```
CALL SYSPROC.ADMIN_CMD( 'quiesce db immediate' )
```

- FORCE CONNECTIONS 옵션이 제공된 경우 이 명령은 강제로 모든 사용자가 데이터베이스를 해제하도록 합니다. FORCE CONNECTIONS는 디폴트 동작이며, 호환성을 위해 명령에 이 매개변수를 사용할 수 있습니다.
- 이 명령은 FORCE CONNECTIONS와 동기화되며 FORCE CONNECTIONS가 완료된 후에만 완료됩니다.

### 사용 시 참고사항

- QUIESCE DATABASE 후, *sysadm*, *sysmaint*, *sysctrl* 또는 *dbadm* 권한과, GRANT/REVOKE 특권이 있는 사용자가 연결될 수 있는 사용자를 지정할 수 있습니다. 이 정보는 데이터베이스 카탈로그 테이블에 영구적으로 저장됩니다.

예를 들면, 다음과 같습니다.

```
grant quiesce_connect on database to <username/groupname>  
revoke quiesce_connect on database from <username/groupname>
```

- 명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.



## 명령 매개변수

### TABLE

#### *tablename*

규정되지 않은 테이블 이름을 지정합니다. 테이블은 시스템 카탈로그 테이블이 될 수 없습니다.

#### *schema.tablename*

규정된 테이블 이름을 지정합니다. *schema*를 제공하지 않으면 CURRENT SCHEMA를 사용합니다. 테이블은 시스템 카탈로그 테이블이 될 수 없습니다.

### SHARE

Quiesce가 공유 모드가 되도록 지정합니다.

"Quiesce 상태 공유" 요청을 작성하면 트랜잭션은 테이블 스페이스에 대해 부분 공유 잠금을 요청하고 테이블에 대해서는 공유 잠금을 요청합니다. 트랜잭션이 잠금을 획득할 때 테이블 스페이스의 상태는 QUIESCED SHARE로 변경됩니다. 이 상태는 다른 사용자가 보유하는 상태와 충돌하지 않는 경우에만 Quiescer에게 권한 부여됩니다. 테이블 스페이스의 상태는 Quiescer의 데이터베이스 에이전트 ID 및 권한 부여 ID와 함께 테이블 스페이스 테이블에 기록되므로 상태는 지속됩니다. 테이블에 대한 테이블 스페이스가 QUIESCED SHARE 상태인 동안 테이블을 변경할 수 없습니다. 테이블 및 테이블 스페이스에 대한 다른 공유 모드 요청을 사용할 수 있습니다. 트랜잭션이 커밋하거나 롤백할 때 잠금은 릴리스되지만, 테이블에 대한 테이블 스페이스는 상태가 명시적으로 재설정될 때까지 QUIESCED SHARE 상태로 남아 있습니다.

### INTENT TO UPDATE

Quiesce가 갱신 가능 모드가 되도록 지정합니다.

"Quiesce 갱신 가능" 요청을 작성하면 테이블 스페이스가 의도를 가진 독점(IX) 모드로 잠겨지고 테이블은 갱신(U) 모드로 잠겨집니다. 테이블 스페이스의 상태는 테이블 스페이스 테이블에 기록됩니다.

### EXCLUSIVE

Quiesce가 독점 모드가 되도록 지정합니다.

"Quiesce 독점" 요청을 작성하면 트랜잭션은 테이블 스페이스에 대해 강한 배타적 잠금을 요청하고 테이블에 대해서는 강한 배타적 잠금을 요청합니다. 트랜잭션이 잠금을 획득할 때, 테이블 스페이스의 상태는 QUIESCED EXCLUSIVE로 변경됩니다. 테이블 스페이스의 상태는 Quiescer의 데이터베이스 에이전트 ID 및 권한 부여 ID와 함께 테이블 스페이스 테이블에 기록됩니다. 테이블 스페이스가 강한 독점 모드로 보유되어 있으므로 이 테이블 스페이스에 대한 다른 액세스는 허용되지 않습니다. Quiesce 함수(Quiescer)를 호출하는 사용자는 테이블 및 테이블 스페이스에 대해 독점 액세스를 가집니다.

## RESET

테이블 스페이스의 상태가 정상으로 재설정되도록 지정합니다. Quiesce 요청을 발행한 연결이 계속 사용 중이면 Quiesce 상태를 재설정할 수 없습니다.

예 :

STAFF 테이블이 있는 테이블 스페이스를 Quiesce하십시오.

```
CALL SYSPROC.ADMIN_CMD( 'quiesce tablespaces for table staff share' )
```

## 사용 시 참고사항

선언된 임시 테이블에 대해서는 이 명령이 지원되지 않습니다.

Quiesce는 지속적 잠금입니다. 이 잠금의 이점은 트랜잭션 실패, 연결 실패 및 심지어 시스템 실패(예: 전원 실패 또는 재부팅) 가운데서도 지속된다는 점입니다.

Quiesce는 연결에 소유됩니다. 연결이 유실될 경우 Quiesce는 남아 있지만 소유자가 없으므로 *팬텀 Quiesce*라고 합니다. 예를 들어, 삭제 단계 동안 정전으로 인해 로드 조작이 인터럽트될 경우, 로드된 테이블에 대한 테이블 스페이스는 삭제 보류, Quiesce 독점 상태로 남게 됩니다. 데이터베이스가 재시작될 때 이 Quiesce는 알 수 없는(또는 팬텀) Quiesce가 됩니다. 팬텀 Quiesce를 제거하려면 Quiesce 모드가 설정될 때 사용된 것과 동일한 사용자 ID로 연결되어야 합니다.

팬텀 Quiesce를 제거하려면 다음을 수행하십시오.

1. Quiesce 모드가 설정될 때 사용된 것과 동일한 사용자 ID로 데이터베이스에 연결하십시오.
2. LIST TABLESPACES 명령을 사용하여 Quiesce된 테이블 스페이스를 판별하십시오.
3. 현재 Quiesce 상태를 사용하여 해당 테이블 스페이스를 다시 Quiesce하십시오. 예를 들면, 다음과 같습니다.

```
CALL SYSPROC.ADMIN_CMD('quiesce tablespaces for table mytable exclusive' )
```

완료되면 새 연결이 Quiesce를 소유하게 되며 로드 조작을 재시작할 수 있습니다.

지정된 시간에 한 테이블 스페이스에서 Quiescer를 5개까지 사용하도록 제한합니다.

Quiescer는 테이블 스페이스의 상태를 덜 제한적인 상태에서 더 제한적인 상태(예: S에서 U로 또는 U에서 X로)로 업그레이드할 수 있습니다. 사용자가 이미 보유되어 있는 상태보다 더 낮은 상태를 요청하면 원래 상태가 리턴됩니다. 상태는 강등되지 않습니다.

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

## ADMIN\_CMD 프로시저를 사용하는 REDISTRIBUTE DATABASE PARTITION GROUP 명령

데이터베이스 파티션 그룹에 있는 데이터베이스 파티션 간에 데이터를 재분배합니다. 특정 시스템 요구사항에 맞게 데이터의 목표 분산이 일정(디폴트)하거나 사용자 지정될 수 있습니다.

REDISTRIBUTE DATABASE PARTITION GROUP 명령은 데이터베이스 파티션 그룹에 있는 모든 파티션 사이에 데이터를 재분배합니다. 이는 데이터베이스 파티션 그룹에 있는 모든 오브젝트에 영향을 주며 하나의 오브젝트로만 제한될 수 없습니다.

### 범위

이 명령은 데이터베이스 파티션 그룹에 있는 모든 데이터베이스 파티션에 영향을 줍니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SYSADM
- SYSCTRL
- DBADM

또한 다음과 같은 권한 그룹 중 하나도 필요합니다.

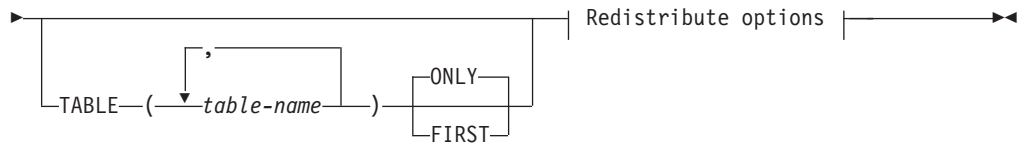
- 재분배 중인 데이터베이스 파티션 그룹의 모든 테이블에 대한 DELETE, INSERT, SELECT 특권
- DATAACCESS 권한

### 필수 연결

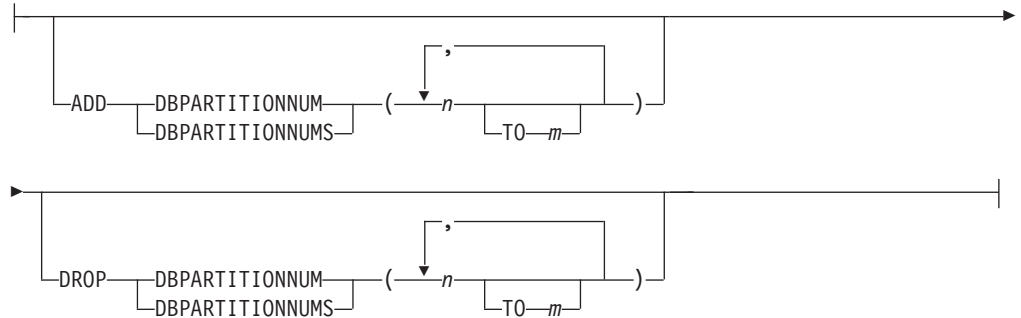
카탈로그 파티션에 대한 연결.

### 명령 구문

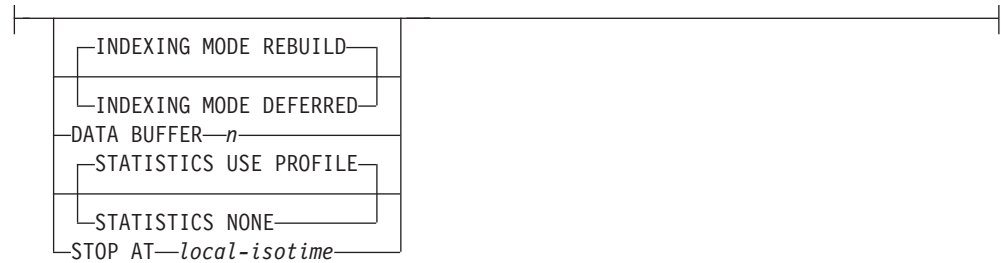
```
▶▶—REDISTRIBUTE DATABASE PARTITION GROUP—db-partition-group—————▶▶
▶———|
▶———| NOT ROLLFORWARD RECOVERABLE |
▶———|
▶———| UNIFORM | Add/Drop DB partition |
▶———| USING DISTFILE—distfilename—|
▶———| USING TARGETMAP—targetmapfilename—|
▶———| CONTINUE |
▶———| ABORT |
```



**추가/삭제 DB 파티션:**



**재분배 옵션:**



**명령 매개변수**

**DATABASE PARTITION GROUP *db-partition-group***

데이터베이스 파티션 그룹 이름. 이 한 부분으로 된 이름은 SYSCAT.DBPARTITIONGROUPS 카탈로그 테이블에 설명된 데이터베이스 파티션 그룹을 식별합니다. 데이터베이스 파티션 그룹은 현재 재분배될 수 없습니다.

주: IBMCATGROUP 및 IBMTEMPGROUP 데이터베이스 파티션 그룹에 있는 테이블을 재분배할 수 없습니다.

**NOT ROLLFORWARD RECOVERABLE**

이 옵션 사용 시, REDISTRIBUTE DATABASE PARTITION GROUP 명령은 롤 포워드 복구 가능하지 않습니다.

- 내부 삽입 및 삭제 조작 대신 대량으로 데이터가 이동됩니다. 이렇게 하면 테이블을 스캔하고 액세스하는 시간을 줄여 더 나은 성능을 가져옵니다.
- 각 삽입 및 삭제 조작에는 로그 레코드가 더 이상 필요하지 않습니다. 이는 데이터 재분배 시 사용자 시스템에 사용 중인 대량의 로그 스페이스 및 로



그 아카이브 스페이스 관리가 필요하지 않음을 의미합니다. 과거에 대량의 사용 중인 로그 스페이스와 스토리지 요구사항으로 인해 단일 데이터 재분배 작업을 여러 개의 더 작은 재분배 태스크로 구분함으로써 단말 간 데이터 재분배 작업 완료에 더 많은 시간이 걸릴 수 있는 상황에서는 특히 유용합니다.

- REDISTRIBUTE DATABASE PARTITION GROUP 명령에 NOT ROLLFORWARD RECOVERABLE 옵션을 사용하는 경우 재분배 작업 시에 XML 컬럼을 포함하는 테이블에 대해 INDEXING MODE DEFERRED 옵션이 사용됩니다. 테이블에 XML 컬럼이 포함되지 않은 경우 재분배 작업은 명령 실행 시에 지정한 인덱스 모드를 사용합니다.

이 옵션이 사용되지 않으면, 모든 행 이동의 확장 로깅이 수행되어 인터럽트, 오류 또는 다른 비즈니스 필요에 따라 나중에 데이터베이스를 복구할 수 있습니다.

## UNIFORM

데이터가 해시 파티션에 균등하게 분산되도록 지정하지만(즉, 모든 해시 파티션이 동일한 행 수를 갖는 것으로 간주됨) 각 데이터베이스 파티션에 동일한 해시 파티션 수가 맵핑되지 않습니다. 재분배 후 데이터베이스 파티션 그룹의 모든 데이터베이스 파티션은 대략 동일한 수의 해시 파티션을 갖게 됩니다.

## USING DISTFILE *distfilename*

분산 키 값의 분산이 균등하지 않을 경우, 이 옵션을 사용하여 데이터베이스 파티션 그룹의 데이터베이스 파티션에서 균등한 데이터 재분배를 달성하십시오.

32 768 해시 파티션에서 현재 데이터 분산을 표시하려면 *distfilename*을 사용하십시오.

각 해시 파티션이 표시하는 데이터의 양을 표시하려면 행 계수, 바이트 볼륨 또는 기타 수치를 사용하십시오. 이 유틸리티는 파티션과 연관된 정수 값을 해당 파티션의 가중치로 읽습니다. *distfilename*이 지정될 때 유틸리티는 데이터베이스 파티션 그룹의 데이터베이스 파티션에서 데이터를 가능한 균등하게 재분배하는 데 사용하는 목표 분산 맵을 생성합니다. 재분배 후, 데이터베이스 파티션 그룹에 있는 각 데이터베이스 파티션의 가중치는 대략적으로 동일합니다(데이터베이스 파티션의 가중치는 해당 데이터베이스 파티션에 맵핑되는 모든 해시 파티션의 가중치 합계임).

예를 들어, 입력 분산 파일에는 다음과 같은 항목이 포함될 수 있습니다.

```
10223
1345
112000
0
100
...
```

예에서, 해시 파티션 2의 가중치는 112000이며 파티션 3(가중치 0)에는 데이터가 전혀 맵핑되지 않습니다.

*distfilename*은 32 768 양의 정수값을 문자 형식으로 포함해야 합니다. 이들 값의 합계는 4 294 967 295 이하가 되어야 합니다.

*distfilename*의 전체 경로 이름이 포함되어야 하며 *distfilename*은 서버에 존재하고 연결된 파티션에서 액세스할 수 있어야 합니다.

#### **USING TARGETMAP** *targetmapfilename*

*targetmapfilename*에 지정된 파일은 목표 분산 맵으로 사용됩니다. 데이터 재분배는 이 파일에 따라 수행됩니다. *targetmapfilename*의 전체 경로 이름이 포함되어야 하며 *targetmapfilename*은 서버에 존재하고 연결된 파티션에서 액세스할 수 있어야 합니다.

목표 맵에 포함된 데이터베이스 파티션이 데이터베이스 파티션 그룹에 있지 않으면 오류가 리턴됩니다. REDISTRIBUTE DATABASE PARTITION GROUP 명령을 실행하기 전에 ALTER DATABASE PARTITION GROUP ADD DBPARTITIONNUM문을 발행하십시오.

목표 맵에서 제외된 데이터베이스 파티션이 데이터베이스 파티션 그룹에 있는 경우, 이 데이터베이스 파티션은 파티션되는 데 포함되지 않습니다. 이러한 데이터베이스 파티션은 REDISTRIBUTE DATABASE PARTITION GROUP 명령 전후에 ALTER DATABASE PARTITION GROUP DROP DBPARTITIONNUM문을 사용하여 삭제될 수 있습니다.

#### **CONTINUE**

이전에 실패하거나 중지된 REDISTRIBUTE DATABASE PARTITION GROUP 조작을 계속 수행합니다. 아무 것도 발생하지 않으면 오류가 리턴됩니다.

#### **ABORT**

이전에 실패하거나 중지된 REDISTRIBUTE DATABASE PARTITION GROUP 조작을 중단합니다. 아무 것도 발생하지 않으면 오류가 리턴됩니다.

#### **ADD**

**DBPARTITIONNUM** *n*

**TO** *m*

*n* 또는 *n TO m*은 데이터베이스 파티션 그룹에 추가된 데이터베이스 파티션 번호 목록을 지정합니다. 지정된 파티션은 데이터베이스 파티션 그룹에 정의되어 있지 않아야 합니다(SQLSTATE 42728). 이것은 ADD DBPARTITIONNUM 절이 지정된 ALTER DATABASE PARTITION GROUP문을 실행하는 것과 같습니다.

**DBPARTITIONNUMS** *n*

## **TO *m***

*n* 또는 *n TO m*은 데이터베이스 파티션 그룹에 추가된 데이터베이스 파티션 번호 목록을 지정합니다. 지정된 파티션은 데이터베이스 파티션 그룹에 정의되어 있지 않아야 합니다(SQLSTATE 42728). 이것은 ADD DBPARTITIONNUM 절이 지정된 ALTER DATABASE PARTITION GROUP문을 실행하는 것과 같습니다.

주: 이 옵션을 사용하여 데이터베이스 파티션을 추가하면, 테이블 스페이스의 컨테이너는 데이터베이스 파티션 그룹에서 가장 번호가 낮은 기존 파티션의 해당 테이블 스페이스의 컨테이너에 기준합니다. 이로 인해 컨테이너 간의 이름지정 충돌이 발생하는 경우, 이는 새 파티션이 기존 컨테이너와 동일한 실제 머신에 있을 경우 발생할 수 있으며, 이때 이 옵션을 사용하지 말아야 합니다. 대신 ALTER DATABASE PARTITION GROUP문은 REDISTRIBUTE DATABASE PARTITION GROUP 명령을 발행하기 전에 WITHOUT TABLESPACES 옵션과 함께 사용되어야 합니다. 그러면 수동으로 적절한 이름을 지정하여 테이블 스페이스 컨테이너를 작성할 수 있습니다.

## **DROP**

### **DBPARTITIONNUM *n***

#### **TO *m***

*n* 또는 *n TO m*은 데이터베이스 파티션 그룹에서 삭제되는 데이터베이스 파티션 번호 목록을 지정합니다. 지정된 파티션은 데이터베이스 파티션 그룹에 정의되어 있어야 합니다(SQLSTATE 42729). 이것은 DROP DBPARTITIONNUM 절이 지정된 ALTER DATABASE PARTITION GROUP문을 실행하는 것과 같습니다.

### **DBPARTITIONNUMS *n***

#### **TO *m***

*n* 또는 *n TO m*은 데이터베이스 파티션 그룹에서 삭제되는 데이터베이스 파티션 번호 목록을 지정합니다. 지정된 파티션은 데이터베이스 파티션 그룹에 정의되어 있어야 합니다(SQLSTATE 42729). 이것은 DROP DBPARTITIONNUM 절이 지정된 ALTER DATABASE PARTITION GROUP문을 실행하는 것과 같습니다.

### **TABLE *tablename***

재분배 처리를 위한 테이블 순서를 지정합니다.

## ONLY

테이블 순서 다음에 **ONLY** 키워드(디폴트)가 오면, 지정된 테이블만이 재분배됩니다. 남아 있는 테이블은 나중에 **REDISTRIBUTE CONTINUE** 명령으로 처리될 수 있습니다. 이는 디폴트값입니다.

## FIRST

테이블 순서 다음에 **FIRST** 키워드가 표시되면, 지정된 테이블이 표시된 순서로 재분배되고 데이터베이스 파티션 그룹에 남아있는 테이블은 무작위 순서로 재분배됩니다.

## INDEXING MODE

**NOT ROLLFORWARD RECOVERABLE** 옵션이 지정되면 이 매개변수는 재분배 중 인덱스가 유지되는 방식을 지정합니다. 가능한 값은 다음과 같습니다.

### REBUILD

인덱스는 스크래치에서 재빌드됩니다. 이 옵션을 사용하기 위해 인덱스를 유효하게 할 필요는 없습니다. 이 옵션의 사용 결과로, 인덱스 페이지가 디스크에서 함께 클러스터됩니다.

### DEFERRED

재분배는 인덱스를 유지하려고 하지 않습니다. 인덱스는 새로 고칠 필요가 있는 것으로 표시됩니다. 그러한 인덱스에 대한 최초 액세스는 재빌드를 강제하거나, 데이터베이스 재시작 시 인덱스를 재빌드할 수 있습니다.

주: 비MDC 테이블의 경우 테이블에 유효하지 않은 인덱스가 있으면, **INDEXING MODE DEFERRED**를 지정하지 않은 경우 **REDISTRIBUTE DATABASE PARTITION GROUP** 명령이 자동으로 인덱스를 재빌드합니다. MDC 테이블의 경우 **INDEXING MODE DEFERRED**를 지정하면, 유틸리티가 MDC 테이블을 처리하기 위해 복합 인덱스를 필요로 하므로 테이블 재분배가 시작되기 전에 유효하지 않은 복합 인덱스가 재빌드됩니다.

## DATA BUFFER *n*

유틸리티에서 데이터 전송에 필요한 버퍼 스페이스로 사용할 4KB 페이지의 수를 지정합니다. 지정된 값이 지원되는 최소값보다 작으면, 최소값이 사용되며 경고는 리턴하지 않습니다. 이 메모리는 유틸리티 힙에서 바로 할당되며 크기는 **util\_heap\_sz** 데이터베이스 구성 매개변수로 수정할 수 있습니다. 값이 지정되지 않으면 처음에 각 테이블 처리 시 런타임에 유틸리티에 의해 적절한 디폴트 값이 계산됩니다. 특히, 테이블의 시간 재분배 시 유틸리티 힙에서 사용 가능한 메모리의 50%를 사용하고 다양한 테이블 등록 정보를 고려하는 것이 디폴트입니다.

## STOP AT *local-isotime*

이 옵션이 지정되면, 각 테이블의 데이터 재분배를 시작하기 전에 *local-isotime* 을 현재 로컬 시간소인과 비교합니다. 지정된 *local-isotime*이 현재 로컬 시간 소인과 같거나 이전이면, 유틸리티는 경고 메시지로 중지합니다. 중지 시 진행 중인 테이블의 데이터 재분배 처리는 인터럽트 없이 완료됩니다. 테이블의 새 데이터 재분배 처리는 시작되지 않습니다. **CONTINUE** 옵션을 사용하여 처리 되지 않은 테이블을 재분배할 수 있습니다. 이 *local-isotime* 값은 조합된 날짜 및 시간을 식별하는 7파트 문자열 시간소인으로 지정됩니다. 형식은 로컬 시간 으로 표시된 *yyyy-mm-dd-hh.mm.ss.nnnnnn*(연도, 월, 일, 시, 분, 초, 마이크로 초)입니다.

## STATISTICS

이 옵션은 통계 프로파일이 있는 테이블의 통계를 유틸리티가 수집해야 함을 지정합니다. 이 옵션 지정은 데이터 재분배 완료 이후 별도로 **RUNSTATS** 명령 을 발행하는 것보다 더 효과적입니다.

## USE PROFILE

통계 프로파일이 있는 테이블에 대한 통계가 수집됩니다. 통계 프로파 일이 없는 테이블의 경우, 어떤 것도 완료되지 않습니다. 이는 디폴트 값입니다.

## NONE

테이블의 통계는 수집되지 않습니다.

## 예: 재분배 단계

노드 그룹에서 노드를 삭제하거나 추가할 수 있습니다. 다음은 노드 그룹에 새 노드를 추가하고 데이터를 재분배하기 위한 단계입니다. 추가된 데이터베이스 파티션은 분산 맵 에 없지만 데이터베이스 파티션 그룹에 있는 테이블 스페이스에 대한 컨테이너는 작성 되었습니다. 데이터베이스 파티션은 데이터베이스 파티션 그룹 재분배 조작이 완료되었 을 때 데이터베이스 파티션이 분산 맵에 추가됩니다.

1. 재분배해야 할 노드 그룹을 식별하십시오. 이 문서에서 재분배해야 하는 노드 그룹은 『sampleNodegrp』입니다.
2. 재분배하기 전에 사용하지 않도록 설정하거나 제거해야 하는 오브젝트를 식별하십시오.
  - a. MQT 복제: 이 유형의 MQT는 **REDISTRIBUTE** 유틸리티의 일부로 지원되지 않습니다. 재분배를 실행하기 전에 삭제하고 이후에 다시 작성해야 합니다.

```
SELECT tabschema, tablename
FROM syscat.tables
WHERE partition_mode = 'R'
```

- b. 테이블에 기록 이벤트 모니터: 재분배되는 데이터베이스 파티션 그룹에 있는 테이블을 포함하는 자동으로 활성화되는 모든 테이블에 기록 이벤트 모니터를 사용하지 않도록 설정해야 합니다.

```
SELECT distinct evmonname FROM syscat.eventtables E
JOIN syscat.tables T on T.tabname = E.tabname AND T.tabschema = E.tabschema
JOIN syscat.tablespaces S on S.tbspace = T.tbspace AND S.ngname = 'sampleNodegrp'
```

- c. Explain 테이블: 단일 파티션 노드 그룹에서 Explain 테이블을 작성할 것을 권장합니다. 그러나 재분배가 필요한 노드 그룹에 정의된 경우, 재분배 이전에 삭제하고 재분배 완료 시 재정의할 것을 고려할 수 있습니다(현재까지 생성된 데이터를 유지보수하지 않아도 되는 경우).

- d. 테이블 액세스 모드 및 로드 상태: 재분배되는 노드 그룹의 모든 테이블이 전체 액세스 모드 상태이고 보류 중이거나 진행 중인 상태의 로드가 없어야 합니다.

```
SELECT distinct trim(T.creator) || '#' || trim(T.name)
AS name, T.access_mode, A.load_status
FROM sysibm.systables T, sysibm.sysnodegroups N, sysibmadm.admintabinfo A
WHERE T.pmap_id = N.pmap_id
AND A.tabschema = T.creator
AND A.tabname = T.name
AND N.name = 'sampleNodegrp'
AND (T.access_mode <> 'F' or A.load_status is not null)
```

- e. 통계 프로파일: 통계 프로파일이 테이블에 대해 정의된 경우 테이블 통계는 재분배 프로세스의 일부로 갱신 가능합니다. REDISTRIBUTE 유틸리티가 테이블의 통계를 갱신하는 경우 재분배를 위해 모든 데이터가 스캔되어 RUNSTATS에 대해 추가 데이터 스캔이 필요하지 않기 때문에 입출력이 감소합니다.

```
RUNSTATS on table schema.table
USE PROFILE runstats_profile
SET PROFILE ONLY
```

3. 데이터베이스 구성을 검토하십시오. **util\_heap\_sz**는 데이터베이스 파티션 사이의 데이터 이동 처리에 매우 중요합니다. 재분배 지속 기간 동안 가능한 최대 메모리를 **util\_heap\_sz**에 할당하십시오. 인덱스 재빌드가 재분배의 일부로 수행되는 경우 충분한 **sortheap**이 필요합니다. 재분배 성능을 향상시키기 위해서는 **util\_heap\_sz** 및 **sortheap**을 필요한 만큼 늘리십시오.
4. 새 데이터베이스 파티션에 사용할 데이터베이스 구성 설정을 검색하십시오. 데이터베이스 파티션을 추가할 때 디폴트 데이터베이스 구성이 사용됩니다. 결과적으로, 전체 웨어하우스 간의 균형을 조정하기 위해 REDISTRIBUTE 명령이 실행되기 전에 새 노드에서 데이터베이스 구성을 갱신하는 것이 중요합니다.

```
SELECT name,
CASE WHEN deferred_value_flags = 'AUTOMATIC'
THEN deferred_value_flags
ELSE substr(deferred_value,1,20)
END
AS deferred_value
FROM sysibmadm.dbcfg
WHERE dbpartitionnum = existing-node
AND deferred_value != ''
```

```

AND name NOT IN ('hadr_local_host','hadr_local_svc','hadr_peer_window',
                 'hadr_remote_host','hadr_remote_inst','hadr_remote_svc',
                 'hadr_syncmode','hadr_timeout','backup_pending','codepage',
                 'codeset','collate_info','country','database_consistent',
                 'database_level','hadr_db_role','log_retain_status',
                 'loghead','logpath','multipage_alloc','numsegs','pagesize',
                 'release','restore_pending','restrict_access',
                 'rollfwd_pending','territory','user_exit_status',
                 'number_compat','varchar2_compat','database_memory')

```

5. 최근 복구점을 확인하려면 재분배 프로세스를 시작하기 전에 데이터베이스(재분배 할 노드 그룹의 테이블 스페이스)를 백업하십시오.
6. db2nodes.cfg 파일을 갱신하고 새 데이터 BCU 데이터베이스 파티션 스펙을 추가하여 DB2에 새 데이터 BCU를 정의하고 ADD NODE WITHOUT TABLESPACES 명령을 사용하여 DB2에 새 데이터베이스 파티션을 정의하십시오.

```

db2start nodenum x export DB2NODE=x
db2 add node without tablespaces
db2stop nodenum x

```

주: 데이터 BCU에 대해 첫 번째 논리적 포트가 아니면, 연속 논리적 포트에 대한 위의 명령 시퀀스 전후에 첫 번째 논리적 포트 번호 시작 및 중지를 실행하십시오.

7. 새로 정의된 데이터베이스 파티션에 대해 시스템 임시 테이블 스페이스 컨테이너를 정의하십시오.

```

ALTER TABLESPACE tablespace_name ADD container_information ON
dbpartitionnums (x to y)

```

8. 데이터 BCU에 걸쳐 있는 데이터베이스 파티션 그룹에 새 논리 데이터베이스 파티션을 추가하십시오.

```

ALTER DATABASE PARTITION GROUP partition_group_name
ADD dbpartitionnums (x to y)
WITHOUT TABLESPACES

```

9. 새로 정의된 데이터베이스 파티션에 대해 영구 데이터 테이블 스페이스 컨테이너를 정의하십시오.

```

ALTER TABLESPACE tablespace_name ADD container_information ON
dbpartitionnums (x to y)

```

10. 4단계에서 검색한 데이터베이스 구성 설정을 새 데이터베이스 파티션에 적용하십시오(또는 구성 지원의 새 DB2 9.5 단일 뷰를 사용하여 모든 데이터베이스 파티션에 대해 단일 UPDATE DB CFG 명령을 실행하십시오).
11. 재분배할 데이터베이스 파티션 그룹의 정의를 캡처한 후 이 그룹에 존재하는 복제된 MQT를 삭제하십시오.

```

db2look -d dbname -e -z schema -t replicated_MQT_table_names -o repMQTs.clp

```

12. 재분배할 데이터베이스 파티션 그룹에 존재하는 테이블에 기록 이벤트 모니터를 사용하지 않도록 설정하십시오.



```
SET EVENT MONITOR monitor_name STATE 0
```

13. REDISTRIBUTE 유틸리티를 실행하여 모든 데이터베이스 파티션 사이에 일정하게 재분배하십시오. 다음은 단순한 재분배 명령을 보여줍니다.

```
REDISTRIBUTE DATABASE PARTITION GROUP sampleNodegrp NOT ROLLFORWARD  
RECOVERABLE uniform;
```

사용자는 또한 REDISTRIBUTE 명령의 입력으로 테이블 목록을 지정하여 테이블이 처리될 순서를 지시할 것을 고려해야 합니다. REDISTRIBUTE 유틸리티는 데이터를 이동시킵니다(압축 및 간략화). 선택적으로, 인덱스가 재빌드되고 통계 프로파일이 정의된 경우 통계가 갱신됩니다. 따라서 이전 명령 대신 다음 스크립트를 실행할 수 있습니다.

```
REDISTRIBUTE DATABASE PARTITION GROUP sampleNodegrp NOT ROLLFORWARD  
RECOVERABLE uniform TABLE (tab1, tab2,...) FIRST;
```

## NOT ROLLFORWARD RECOVERABLE 옵션 사용 결과

REDISTRIBUTE DATABASE PARTITION GROUP 명령이 발행되고 NOT ROLLFORWARD RECOVERABLE 옵션이 지정되면, 이동된 각 행의 로그 레코드 쓰기를 최소화하는 최소 로깅 전략이 사용됩니다. 모든 데이터 이동을 완전히 로그하는 접근 방식이 대형 시스템에 가능하며, 비현실적인 활성 영구 로그 스페이스를 요구하며 일반적으로 성능이 더 저하된 성능 특성을 가지므로 재분배 조작의 사용 편리성에는 이러한 유형의 로깅이 중요합니다. 그러나 이러한 최소 로깅 모델의 결과로서, REDISTRIBUTE DATABASE PARTITION GROUP 명령이 복구 가능한 롤 포워드가 아님을 인식하는 것이 중요합니다. 이것은 재분배 조작을 통해 데이터베이스를 롤 포워드하는 모든 조작이 UNAVAILABLE 상태로 남아 있는 재분배 조작으로 모든 테이블을 처리하는 결과를 가져옴을 의미합니다. 그러한 테이블은 삭제만 가능하며, 이들 테이블의 데이터를 복구할 방법이 없음을 의미합니다. 그 이유는 복구 가능한 데이터베이스의 경우, REDISTRIBUTE DATABASE PARTITION GROUP 유틸리티가 NOT ROLLFORWARD RECOVERABLE 옵션으로 발행되면 사용하는 모든 테이블 스페이스를 BACKUP PENDING 상태로 만들어 정상적인 재분배 조작의 끝에서 사용자가 강제로 모든 재분배 테이블 스페이스를 백업하게 하기 때문입니다. 재분배 조작 이후의 백업 시, 사용자는 재분배 조작 자체를 통해 롤 포워드하지 말아야 합니다.

재분배 유틸리티에서 롤 포워드 복구 가능성이 없다는 점은 간과할 수 없는 중요한 문제입니다. 재분배에 의해 처리되는 테이블 스페이스를 사용자가 백업하는 재분배 주기의 종료 지점을 포함하여 재분배 조작이 실행되는 동안 데이터베이스의 테이블 갱신을 허용한 경우(재분배되는 데이터베이스 파티션 그룹 외부의 테이블인 경우에도) 데이터베이스 컨테이너 손상과 같은 심각한 장애의 경우 해당 갱신사항이 손실될 수 있습니다. 그러한 갱신이 유실되는 이유는 재분배 조작이 복구 가능한 롤 포워드가 아니기 때문입니다. 재분배 조작 이전의 백업에서 데이터베이스를 리스토어하는 것이 필요한 경우, 위에서 설명한 대로 재분배를 통해 롤 포워드하지 않는 재분배 조작 중 수행된 갱신을 재생하기 위해 로그를 롤 포워드하는 것이 가능하지 않고, 재분배된 테이블을 사용 불가능한 상태로 둡니다. 그러므로 이 상황에서 수행할 수 있는 유일한 것은 롤 포



워드 없이 재분배하기 전에 수행한 백업에서 데이터베이스를 리스토어하는 것입니다. 그러면 재분배 조작을 다시 수행할 수 있습니다. 그러나 원래의 재분배 조작 중 발생한 모든 갱신은 유실됩니다.

이러한 사항은 매우 중요합니다. 재분배 조작 중 갱신이 유실되지 않음을 보장하려면, 다음 중 하나가 참이어야 합니다.

- 영향을 받는 테이블 스페이스를 백업 중인 명령 완료 이후 기간을 포함하여 **REDISTRIBUTE DATABASE PARTITION GROUP** 명령의 조작 중 갱신하는 것을 피합니다.
- 재분배 조작 중 적용되는 갱신은 반복 가능한 소스에서 오는데, 이는 언제든지 다시 적용될 수 있음을 의미합니다. 예를 들어, 갱신 소스가 파일에 저장되는 데이터이며 일괄처리 중 갱신이 적용되는 경우, 데이터베이스 리스토어를 필요로 하는 실패에서조차도 갱신이 유실되지 않는 이유는 언제든지 다시 갱신을 적용할 수 있기 때문입니다.

재분배 조작 중 데이터베이스 갱신을 허용하는 경우, 필요하다면 사용자는 그러한 갱신이 적절한지 아니면 데이터베이스 리스토어 이후에 갱신을 반복할 수 있는지 여부를 결정해야 합니다.

**주:** **REDISTRIBUTE DATABASE PARTITION GROUP** 명령을 조작하는 동안 모든 실패가 이러한 문제점을 유발하지는 않습니다. 사실상 대부분은 문제점을 유발하지 않습니다. **REDISTRIBUTE DATABASE PARTITION GROUP** 명령은 완전하게 재시작할 수 있으며 이는 작업 중간에 유틸리티가 실패하는 경우 **CONTINUE** 또는 **ABORT** 옵션을 사용하여 간편하게 계속하거나 중단할 수 있음을 나타냅니다. 위에서 언급하는 실패는 재분배 조작 이전에 수행된 백업에서 리스토어하는 것이 필요한 실패입니다.

**예:**

데이터 분산 파일 `distfile_for_dbpg_1`을 통해 현재 데이터 분산을 제공하고 두 개의 새 데이터베이스 파티션 6과 7로 데이터를 이동시켜 데이터베이스 파티션 그룹 `DBPG_1`을 재분배합니다.

```
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)')
```

## 사용 시 참고사항

- **NOT ROLLFORWARD RECOVERABLE** 옵션을 지정하고 데이터베이스가 복구 가능한 데이터베이스인 경우 유틸리티가 테이블 스페이스에 처음 액세스할 때 **BACKUP PENDING** 상태가 됩니다. 해당 테이블 스페이스의 모든 테이블은 테이블 스페이스가 백업될 때까지 읽기 전용으로 됩니다.

- 재분배 조작이 실행 중이면, 처리되는 각 테이블의 시작 및 종료 시간과 같은 정보와 재분배 조작에 대한 일반 정보가 있는 이벤트 로그 파일을 생성합니다. 이 이벤트 로그 파일은 서버에 작성됩니다.
  - 서브디렉토리와 파일 이름에 대해 *database-name.database-partition-group-name.timestamp.log* 형식을 사용하는 Linux 및 UNIX 시스템의 `homeinst/sqlllib/redis` 디렉토리.
  - 서브디렉토리와 파일 이름에 대해 *database-name.database-partition-group-name.timestamp.log* 형식을 사용하는 Windows 운영 체제의 `DB2INSTPROF\instance\redis` 디렉토리(여기서 **DB2INSTPROF**는 **DB2INSTPROF** 레지스트리 변수 값입니다).
  - 시간소인 값은 명령이 발행된 시간입니다.

재분배 이벤트 로그에 대한 자세한 정보는 『재분배 이벤트 로그 파일』 주제를 참조하십시오.

- 유틸리티 프로그램은 처리 중 간헐적인 COMMIT를 수행합니다. 이로 인해 유형 2 연결에서 SQL30090N 오류를 수신할 수 있습니다.
- 재분배가 일어난 테이블에 종속성을 갖는 모든 패키지는 무효화됩니다. 데이터베이스 파티션 그룹 재분배 조작이 완료된 후 이러한 패키지를 명시적으로 리바인드하는 것이 바람직합니다. 명시적으로 리바인드하면 유효하지 않은 패키지에 대한 첫 번째 SQL 요청을 실행할 때 초기 지연이 발생하지 않습니다. 재분배 메시지 파일에는 재분배가 일어난 모든 테이블 목록이 들어 있습니다.
- 디폴트로 재분배 유틸리티는 통계 프로파일이 있는 테이블의 통계를 갱신합니다. 통계 프로파일이 없는 테이블의 경우, 재분배 조작이 완료된 후에 RUNSTATS 명령을 발행하거나 db2Runstats API를 호출하여 해당 테이블의 테이블 및 인덱스 통계를 별도로 갱신하는 것이 좋습니다.
- DATA CAPTURE CHANGES로 정의된 복제된 테이블이나 복제된 구체화된 쿼리 테이블을 포함하는 데이터베이스 파티션 그룹은 재분배할 수 없습니다.
- 데이터베이스 파티션 그룹에 기존의 선언된 임시 테이블이나 작성된 임시 테이블이 있는 사용자 임시 테이블 스페이스가 있으면 재분배를 수행할 수 없습니다.
- **INDEXING MODE**와 같은 옵션은 해당 옵션이 적용되지 않는 테이블에서 경고 없이 무시됩니다. 예를 들어, **INDEXING MODE**는 인덱스가 없는 테이블에서 무시됩니다.
- 명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.
- **USING DISTFILE** *distfilename* 또는 **USING TARGETMAP** *targetmapfilename* 에서 참조되는 파일은 서버에 있는 파일을 나타내야 합니다.
- 재분배 조작을 시작하기 전에 로드 보류 상태의 테이블이 없는지 확인하십시오. LOAD QUERY 명령을 사용하여 테이블 상태를 점검할 수 있습니다.

- 데이터베이스 파티션 서버 추가 요청이 보류 중 또는 진행 중인 경우에는 REDISTRIBUTE DATABASE PARTITION GROUP 명령이 실패할 수 있습니다(SQLSTATE 55071). 새 데이터베이스 파티션 서버가 온라인으로 인스턴스에 추가되고 모든 응용프로그램이 새 데이터베이스 파티션 서버를 인지하는 것이 아니라면 이 명령도 실패합니다(SQLSTATE 55077).

## 호환성

DB2 버전 9.5 이전의 XML 레코드 형식을 사용하는 XML 컬럼을 포함하는 테이블은 재분배할 수 없습니다. 테이블을 새 형식으로 이주하려면 ADMIN\_MOVE\_TABLE 스토어드 프로시저를 사용하십시오.

버전 8 이전 버전과의 호환성:

- **NODEGROUP** 키워드는 **DATABASE PARTITION GROUP**으로 대체될 수 있습니다.

## ADMIN\_CMD 프로시저를 사용하는 REORG INDEXES/TABLE 명령

인덱스 또는 테이블 재구성

인덱스 데이터를 조각나지 않은 실제로 연속된 페이지로 재빌드하여 테이블에 정의된 모든 인덱스를 재구성할 수 있습니다. 또는 범위 파티션 테이블에서 특정 인덱스를 재구성할 수 있는 옵션도 제공됩니다.

인덱스 절의 CLEANUP ONLY 옵션을 지정하면 인덱스를 재빌드하지 않고 정리를 수행합니다. 이 명령은 선언된 임시 테이블 또는 작성된 임시 테이블의 인덱스에 대해서는 사용할 수 없습니다(SQLSTATE 42995).

테이블 옵션은 행을 재구성하여 조각난 데이터를 제거하고 정보를 간략화함으로써 테이블을 재구성합니다.

## 범위

이 명령은 데이터베이스 파티션 그룹에 있는 모든 데이터베이스 파티션에 영향을 줍니다.

## 권한 부여

다음 중 하나가 필요합니다.

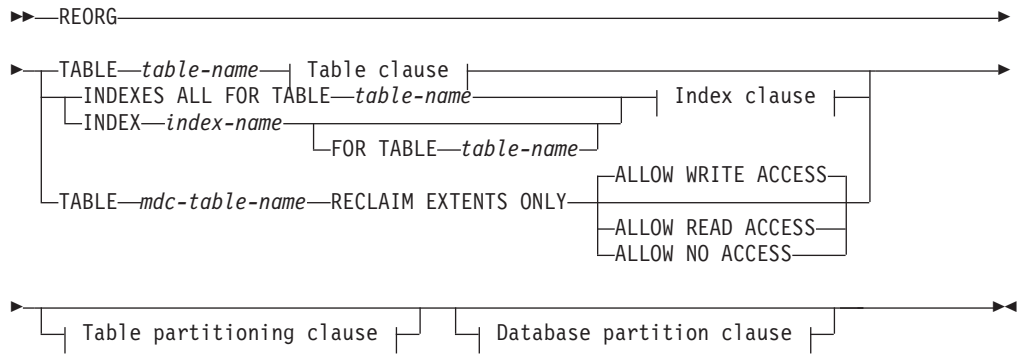
- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

- *sqladm*
- 테이블에 대한 CONTROL 특권

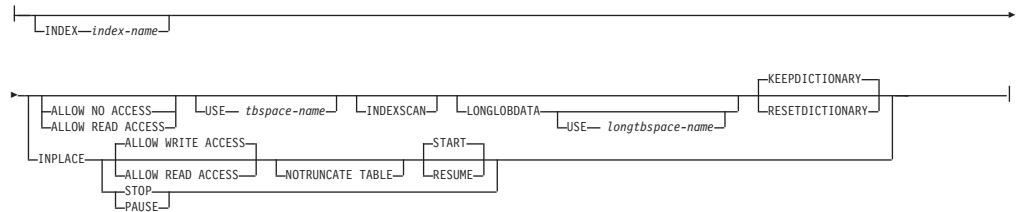
## 필수 연결

데이터베이스

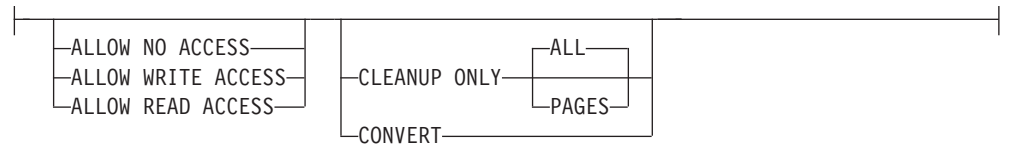
## 명령 구문



### Table clause:



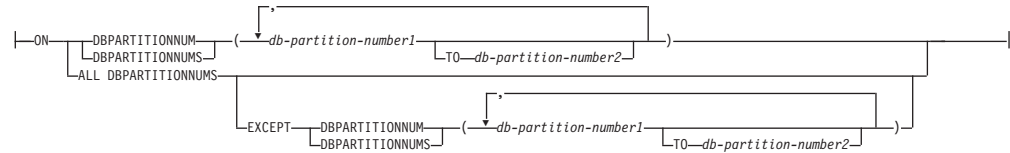
### Index clause:



### 테이블 파티션 절:



## Database partition clause:



## 명령 매개변수

### INDEXES ALL FOR TABLE *table-name*

인덱스가 재구성되는 테이블을 지정합니다. 테이블은 로컬 또는 리모트 데이터베이스에 있을 수 있습니다.

### INDEX *index-name*

파티션된 테이블에 재구성될 개별 인덱스를 지정합니다. 개별 인덱스의 재구성은 파티션된 테이블의 파티션되지 않은 인덱스에만 지원됩니다. 블록 인덱스에 대해서는 이 매개변수가 지원되지 않습니다.

### FOR TABLE *table-name*

파티션된 테이블에서 재구성되는 개별 인덱스의 테이블 이름 위치를 지정합니다. 인덱스 이름이 데이터베이스에서 고유하다고 고려할 때 이 매개변수는 선택적입니다.

REORG INDEXES의 경우, ON DATA PARTITION 옵션이 지정되면 액세스 절은 이름 지정된 파티션에만 적용됩니다. 지정된 파티션의 인덱스가 재구성되는 동안 사용자는 나머지 테이블에서 읽고 쓸 수 있습니다. 이 상황은 디폴트 액세스 등급에도 적용됩니다.

### ALLOW NO ACCESS

REORG INDEX 및 REORG INDEXES의 경우, 인덱스가 재구성되는 동안 다른 사용자가 테이블 또는 데이터 파티션에 액세스할 수 없도록 지정합니다.

### ALLOW READ ACCESS

REORG INDEX 및 REORG INDEXES의 경우, 인덱스가 재구성되는 동안 다른 사용자가 테이블 또는 데이터 파티션에 읽기 전용 액세스할 수 있도록 지정합니다. CLEANUP ONLY 옵션이나 ON DATA PARTITION 옵션이 지정되지 않은 경우 파티션된 테이블의 REORG INDEXES에 대해서는 이 액세스 레벨이 지원되지 않습니다.

### ALLOW WRITE ACCESS

REORG INDEX 및 REORG INDEXES의 경우, 인덱스가 재구성되는 동안 다른 사용자가 테이블 또는 데이터 파티션에서 읽고 쓸 수 있도록 지정합니다.

CLEANUP ONLY 옵션이 지정되지 않은 경우 다차원적으로 클러스터된(MDC) 테이블이나 확장 인덱스에 대해서는 이 액세스 레벨이 지원되지 않습니다. 또한, CLEANUP ONLY 옵션이나 ON DATA PARTITION 옵션이 지정되지 않은 경우 REORG INDEXES 명령이 사용되면 파티션된 테이블에 대해 이 액세스 레벨이 지원되지 않습니다.

ACCESS 모드가 지정되지 않으면, 다음 방법으로 테이블이나 데이터 파티션에 대해 선택됩니다.

표 61. 인덱스 절에 지정된 명령, 테이블 유형 및 추가 매개변수를 기본으로 선택된 디폴트 테이블 액세스

명령	테이블 유형	테이블 파티션 절	인덱스 절에 지정된 추가 매개변수	디폴트 액세스 모드
REORG INDEXES	파티션되지 않은 테이블	해당되지 않음	임의	ALLOW READ ACCESS
REORG INDEX	파티션된 테이블	해당되지 않음	임의	ALLOW READ ACCESS
REORG INDEXES	파티션된 테이블	없음	지정되지 않음	ALLOW NO ACCESS
REORG INDEXES	파티션된 테이블	ON DATA PARTITION	지정되지 않음	ALLOW READ ACCESS
REORG INDEXES	파티션된 테이블	ON DATA PARTITION 절이 있거나 없음	CLEANUP ONLY가 지정됨	ALLOW READ ACCESS

### CLEANUP ONLY

CLEANUP ONLY가 요청된 경우 전체 재구성 대신 정리가 수행됩니다. 인덱스는 재빌드되지 않으며 제거된 페이지는 이 테이블에만 정의된 인덱스를 통해 재사용할 수 있습니다.

CLEANUP ONLY PAGES 옵션은 커미트된 의사(pseudo) 공백 페이지를 검색하여 제거합니다. 커미트된 의사(pseudo) 공백 페이지는 페이지에 있는 모든 키가 삭제된 것으로 표시되고 삭제가 모두 커미트된 것으로 인식되는 페이지입니다. 인덱스에 있는 의사(pseudo) 공백 페이지 수는 RUNSTATS를 실행하여 SYSCAT.INDEXES의 NUM EMPTY LEAFS 컬럼을 보고 판별할 수 있습니다. PAGES 옵션은 NUM EMPTY LEAFS가 커미트된 것으로 판별되면 이를 정리합니다.

CLEANUP ONLY ALL 옵션은 의사(pseudo) 공백이 아닌 페이지에서 커미트된 의사 삭제된 키를 제거할 뿐 아니라 커미트된 의사 공백 페이지를 제거합니다. 이 옵션은 또한 인접 리프 페이지를 병합하는데 이렇게 할 경우 최소한 PCTFREE 여유 공간이 있는 병합된 리프 페이지로 나타납니다. 여기서 PCTFREE는 인덱스 작성시간에 인덱스에 대해 정의된 여유 공간 비율입니다. 디폴트 PCTFREE는 10퍼센트입니다. 두 개의 페이지가 병합될 수 있는 경우 페이지 중 하나가 제거됩니다. 의사(pseudo) 공백 페이지에 있는 경우를 제외하고 인덱스에 있

는 의사(pseudo) 삭제된 키 수는 RUNSTATS를 실행한 후 SYSCAT.INDEXES에서 NUMRIDS DELETED를 삭제하여 판별할 수 있습니다. ALL 옵션은 NUMRIDS DELETED 및 NUM EMPTY LEAFS가 커미트된 것으로 판별되면 이를 정리합니다.

**ALL** 커미트된 의사(pseudo) 삭제된 키 및 커미트된 의사(pseudo) 공백 페이지를 제거하여 인덱스가 정리되도록 지정합니다.

### PAGES

커미트된 의사(pseudo) 공백 페이지가 인덱스 트리에서 제거되도록 지정합니다. 의사(pseudo) 공백이 아닌 페이지에 있는 의사(pseudo) 삭제된 키는 정리하지 않습니다. 의사(pseudo) 공백 리프 페이지만 점검하므로 대부분의 경우에 ALL 옵션을 사용하는 것보다 훨씬 더 신속합니다.

### CONVERT

유형 1 인덱스를 유형 2 인덱스로 변환합니다. 인덱스가 이미 유형 2인 경우 이 옵션은 어떤 영향도 미치지 않습니다.

버전 9.7에서, 유형 1 인덱스는 중지되며 작성되는 모든 인덱스는 유형 2 인덱스입니다. 결과적으로, CONVERT 옵션은 사용되지 않습니다.

버전 8 이전에 작성된 모든 인덱스는 유형 1 인덱스입니다. 버전 9.7 이전에, 이미 유형 1 인덱스를 가지고 있는 테이블에서 인덱스를 작성할 경우를 제외하고는 버전 8 이후에 작성되는 모든 인덱스는 유형 2 인덱스입니다. 이 경우, 새로운 인덱스는 유형 1입니다. 작성되는 모든 인덱스는 유형 2이므로 이는 버전 9.7의 경우가 아닙니다.

인덱스가 재구성되는 동안 다른 트랜잭션이 테이블에 대해 읽기 전용 또는 읽기 쓰기 액세스할 수 있도록 하려면 ALLOW READ ACCESS 또는 ALLOW WRITE ACCESS 옵션을 사용하십시오. ALLOW READ ACCESS 및 ALLOW WRITE ACCESS 옵션을 사용하면 테이블에 액세스할 수 있는 반면, 재구성된 인덱스 사본을 사용할 수 있는 동안에는 테이블에 액세스할 수 없습니다.

### TABLE *mdc-table-name* RECLAIM EXTENTS ONLY

사용 중이지 않은 Extent를 재개하도록 재구성하기 위해 다차원적으로 클러스터된(MDC) 테이블을 지정합니다. *schema.table-name* 형식의 이름 또는 별명을 사용할 수 있습니다. *schema*는 테이블이 작성된 사용자 이름입니다. 스키마 이름을 생략하면 디폴트 스키마가 간주됩니다.

REORG TABLE RECLAIM EXTENTS ONLY의 경우, ON DATA PARTITION 옵션이 지정되면 액세스 절은 이름 지정된 파티션에만 적용됩니다.



다. 지정된 파티션의 Extent가 재개되는 동안 사용자는 나머지 테이블에서 읽고 쓸 수 있습니다. 이 상황은 디폴트 액세스 등급에도 적용됩니다.

#### **ALLOW NO ACCESS**

REORG TABLE RECLAIM EXTENTS ONLY의 경우, Extent가 재개되는 동안 다른 사용자는 해당 테이블을 액세스할 수 없습니다.

#### **ALLOW READ ACCESS**

REORG TABLE RECLAIM EXTENTS ONLY의 경우, Extent가 재개되는 동안 다른 사용자는 테이블에 대한 읽기 전용 액세스 권한을 가질 수 있음을 지정합니다.

#### **ALLOW WRITE ACCESS**

REORG TABLE RECLAIM EXTENTS ONLY의 경우, Extent가 재개되는 동안 다른 사용자는 테이블을 읽고 쓸 수 있음을 지정합니다.

#### **TABLE** *table-name*

재구성할 테이블을 지정합니다. 테이블은 로컬 또는 리모트 데이터베이스에 있을 수 있습니다. *schema.table-name* 형식의 이름 또는 별명을 사용할 수 있습니다. *schema*는 테이블이 작성된 사용자 이름입니다. 스키마 이름을 생략하면 디폴트 스키마가 간주됩니다.

유형이 지정된 테이블의 경우, 지정된 테이블 명은 계층 구조의 루트 테이블 이름이어야 합니다.

다차원적으로 클러스터된(MDC) 테이블의 재구성에 대해서는 인덱스를 지정할 수 없습니다. MDC 테이블에는 테이블의 설정된 재구성을 사용할 수 없습니다.

#### **INDEX** *index-name*

테이블을 재구성할 때 사용할 인덱스를 지정합니다. *schema.index-name* 형식으로 완전한 이름을 지정하지 않으면 디폴트 스키마가 간주됩니다. *schema*는 인덱스가 작성된 사용자 이름입니다. 데이터베이스 관리 프로그램은 인덱스를 사용하여 재구성 중인 테이블의 레코드를 실제로 다시 정렬합니다.

설정된 테이블 재구성의 경우, 테이블에 클러스터링 인덱스가 정의되어 있고 인덱스가 지정되어 있는 경우, 이는 클러스터링 인덱스여야 합니다. Inplace 옵션이 지정되어 있지 않으면 지정된 인덱스가 사용됩니다. 인덱스의 이름을 지정하지 않으면 레코드는 순서에 관계없이 재구성됩니다. 테이블에 클러스터링 인덱스가 정의되어 있지만 인덱스가 지정되어 있지 않으면 클러스터링 인덱스를 사용하여 테이블을 클러스터합니다. MDC 테이블을 재구성하고 있으면 인덱스를 지정할 수 없습니다.

#### **ALLOW NO ACCESS**

테이블이 재구성되는 동안 다른 사용자가 테이블에 액세스할 수 없도



록 지정합니다. Table partitioning 절을 사용하지 않고 파티션된 테이블을 재구성하는 경우 이 값이 디폴트입니다.

#### **ALLOW READ ACCESS**

재구성 중 테이블에 대한 읽기 액세스만 허용합니다. 이 값은 파티션되지 않은 테이블이나 테이블 파티셔닝 절을 사용하는 파티션된 테이블의 디폴트입니다.

ON DATA PARTITION 옵션이 범위 파티션된 테이블의 REORG TABLE 용으로 지정되면, 액세스 절은 이름지정된 파티션에만 적용됩니다. 테이블의 모든 인덱스가 파티션되면, 파티션이 재구성되고 인덱스가 재빌드되는 동안 사용자는 나머지 테이블에서 읽고 쓸 수 있습니다. 그러나 파티션되지 않은 인덱스가 테이블에 정의되면, 전체 테이블의 액세스가 ALLOW NO ACCESS로 확대됩니다. 이 상황에서, 파티션되지 않은 인덱스는 지정된 파티션과 함께 재빌드됩니다.

#### **INPLACE**

사용자 액세스를 허용하면서 테이블을 재구성합니다.

테이블 재구성은 유형 2 인덱스를 포함하지만 확장 인덱스는 없고 테이블의 XML 컬럼에 대해 정의된 인덱스가 없는 파티션되지 않은 테이블 및 비MDC 테이블에서만 허용됩니다. 위치 테이블에서 재구성은 최소한 3페이지 크기의 테이블에서만 수행될 수 있습니다.

설정된 테이블 재구성은 비동기적으로 발생하며 즉시 유효하지 않을 수 있습니다.

#### **ALLOW READ ACCESS**

재구성 중 테이블에 대한 읽기 액세스만 허용합니다.

#### **ALLOW WRITE ACCESS**

재구성 중 테이블에 대한 쓰기 액세스를 허용합니다. 이 설정이 디폴트 동작입니다.

#### **NOTRUNCATE TABLE**

설정된 재구성 후 테이블을 절단하지 마십시오. 절단하는 동안 테이블은 S-잠금됩니다.

#### **START**

설정된 REORG 처리를 시작합니다. 디폴트값이므로 이 키워드는 선택적입니다.

**STOP** 현재 위치에서 설정된 REORG 처리를 중지합니다.

#### **PAUSE**

잠시 동안 설정된 REORG를 일시중단하거나 일시정지합니다.

## RESUME

이전이 일시정지된 설정된 재구성을 계속하거나 재개합니다. 온라인 재구성이 재개되고 재구성이 일시정지되었을 때와 동일한 옵션을 원할 경우, 재개되는 동안 이들 옵션을 다시 지정해야 합니다.

### USE *tbspace-name*

재구성 중인 테이블의 임시 사본을 저장할 시스템 임시 테이블 스페이스 이름을 지정합니다. 테이블 스페이스 이름을 제공하지 않으면 데이터베이스 관리 프로그램은 재구성 중인 테이블이 들어 있는 테이블 스페이스에 테이블의 작업 사본을 저장합니다.

8KB, 16KB 또는 32KB 테이블 오브젝트에서, 사용자가 지정하는 시스템 임시 테이블 스페이스의 페이지 크기가 테이블 데이터가 있는 테이블 스페이스의 페이지 크기와 일치하지 않을 경우 DB2 데이터베이스 제품은 올바른 크기의 LONG/LOB 오브젝트의 임시 테이블 스페이스를 찾으려고 시도합니다. 재구성을 성공하려면 이러한 테이블 스페이스가 존재해야 합니다.

동일한 페이지 크기의 임시 테이블 스페이스가 두 개 있으며 이들 중 하나를 USE절에 지정할 때 재구성 중인 테이블에 인덱스가 있으면 테이블 스페이스는 라운드로빈(round robin) 방식으로 사용됩니다. 두 개의 테이블 스페이스 temp space1 및 temp space2가 있으며 둘 다 동일한 페이지 크기이고 USE 옵션을 사용하는 REORG 명령에 temp space1을 지정한다고 하십시오. 처음 REORG를 수행하면 temp space1이 사용됩니다. 두 번째에는 temp space2가 사용됩니다. 세 번째에는 temp space1이 사용되고 계속 이와 같이 됩니다. 이렇게 하지 않으려면 임시 테이블 스페이스 중 하나를 삭제해야 합니다.

파티션된 테이블의 경우, 테이블 스페이스는 테이블에 있는 데이터 파티션의 재구성에 대한 임시 스토리지로 사용됩니다. 전체 파티션된 테이블을 재구성하면 한 번에 한 데이터 파티션이 재구성됩니다. 필요한 스페이스의 크기는 전체 테이블이 아니라 테이블에서 가장 큰 데이터 파티션 크기와 같습니다.

파티션된 테이블에 테이블 스페이스 이름을 제공하지 않으면 해당 데이터 파티션의 임시 스토리지에 각 데이터 파티션이 위치한 테이블 스페이스를 사용합니다. 각 데이터 파티션의 테이블 스페이스에는 데이터베이스 파티션의 사본을 보유하기에 충분한 여유 공간이 있어야 합니다.

### INDEXSCAN

클러스터링 REORG의 경우 테이블 레코드를 다시 정렬하기 위해 인덱스 스캔을 사용합니다. 인덱스를 통해 테이블에 액세스하여 테이블 행

을 재구성합니다. 디폴트 방법은 필요에 따라 임시 테이블 스페이스를 사용하여 테이블을 스캔한 후 그 결과를 정렬하여 테이블을 재구성하는 것입니다. 인덱스 키가 정렬 순서로 되었다더라도 스캔하고 정렬하는 것이 인덱스에서 먼저 행 ID를 읽고 행을 폐치하는 것보다 일반적으로 더 빠릅니다.

## LONGLOBDATA

Long 필드 및 LOB 데이터가 재구성됩니다.

테이블에 Long 또는 LOB 컬럼이 들어 있더라도 이 옵션은 필수적이지 않습니다. 디폴트값은 이들 오브젝트를 재구성하지 않는 것입니다. 이는 시간이 많이 들고 클러스터링을 향상시키지 않기 때문입니다. 그러나 XML 컬럼이 있는 테이블에서 LONGLOBDATA 옵션으로 재구성을 실행하면 미사용 스페이스를 재개하므로 XML 스토리지 오브젝트 크기를 감축합니다.

기존 LOB 데이터를 인라인된 LOB 데이터로 변환할 때 이 매개변수가 필요합니다.

## USE *longtblspace-name*

이는 선택적 매개변수로서 Long 데이터를 재빌드할 때 사용될 임시 테이블 스페이스의 이름을 지정하는 데 사용될 수 있습니다. 테이블 오브젝트 또는 Long 오브젝트에 대해 임시 테이블 스페이스가 지정되지 않은 경우 오브젝트는 현재 있는 테이블 스페이스에서 구성됩니다. 테이블에 대해 임시 테이블 스페이스가 지정되어 있지만 이 매개변수가 지정되지 않은 경우, 페이지 크기가 서로 다르지 않으면 기본 재구성 데이터에 사용된 테이블 스페이스가 사용됩니다. 이러한 경우 DB2 데이터베이스 시스템은 Long 오브젝트를 작성할 적당한 페이지 크기의 임시 컨테이너를 선택하려고 시도합니다.

USE *longtblspace-name*이 지정된 경우, USE *tblspace-name*도 지정되어야 합니다. 그렇지 않으면 *longtblspace-name* 인수가 무시됩니다.

## KEEPDICTIONARY

테이블에 대한 COMPRESS 속성이 YES 이고 테이블에 압축 사전이 들어 있으면 새 사전이 빌드되지 않습니다. 재구성 중 처리된 모든 행은 기존 사전을 사용하여 압축되기 쉽습니다. COMPRESS 속성이 YES이며 테이블의 압축 사전이 없는 경우, 테이블이 특정 크기(대략 1 - 2MB)이며 충분한 데이터가 이 테이블에 있는 경우 이 시나리오에서 사전만 작성됩니다(테이블은 압축됨). 대신 명시적으로 REORG RESETDICTIONARY를 언급하면, 최소한 1행이 테이블에 있는 한 사전이 빌드됩니다. 테이블에 대한 COMPRESS 속성이 NO이고 테이블에 압축 사전이 들어 있으면 REORG 처리는 이 사전을 보존하고 새로 재

구성된 테이블의 모든 행은 압축되지 않은 형식이 됩니다. 기본 테이블 행에 저장되지 않은 LOB 데이터와 같은 일부 데이터는 압축할 수 없습니다.

LONGLOBDATA 옵션이 지정되지 않으면, 테이블 행 데이터만이 재구성됩니다. LONGLOBDATA 옵션이 지정되지 않으면 다음 표는 REORG 명령에서 KEEPDICTIONARY 구문의 동작을 설명합니다.

표 62. REORG KEEPDICTIONARY

압축	사전이 존재함	결과, 결론
Y	Y	사전 보존; 행 압축
Y	N	사전 빌드; 행 압축
N	Y	사전 보존; 모든 행 압축 해제
N	N	영향을 미치지 않음, 모든 행을 압축 해제합니다.

LONGLOBDATA 옵션이 지정되면 다음 표는 REORG 명령에서 KEEPDICTIONARY 구문의 동작을 설명합니다.

표 63. LONGLOBDATA 옵션이 지정될 때 REORG KEEPDICTIONARY

압축	테이블 행 데이터 사전 존재	XML 스토리지 오브젝트 사전 존재 <sup>1</sup>	압축 사전	데이터 압축
Y	Y	Y	사전을 보존합니다.	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
Y	Y	N	테이블 행 사전을 보존하며 XML 스토리지 오브젝트 사전을 작성합니다.	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
Y	N	Y	테이블 행 사전을 작성하며 XML 사전을 보존합니다.	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
Y	N	N	테이블 행 및 XML 사전을 작성합니다.	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
N	Y	Y	테이블 행 및 XML 사전을 보존합니다.	테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.
N	Y	N	테이블 행 사전을 보존합니다.	테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.
N	N	Y	XML 사전을 보존합니다.	테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.
N	N	N	영향을 미치지 않습니다.	테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.

메모:

1. XML 컬럼이 DB2 V9.7 이상에서 테이블에 추가된 경우나, 테이블이 ONLINE\_TABLE\_MOVE 스토어드 프로시저를 사용하여 이주된 경우에만 XML 스토리지 오브젝트의 압축 사전이 작성될 수 있습니다.

모든 테이블을 다시 초기화하거나 절단할 경우(예: 바꾸기 조작) 테이블의 압축 속성이 NO이면 사전을 버립니다(사전이 있을 경우). 반대로 사전이 있으나 테이블의 압축 속성이 YES이면 절단 조작은 사전을 저장하고 버리지 않습니다. 복구 및 추후 지원을 목적으로 데이터 캡처 변경사항(즉, 복제)과 함께 사전 전체가 로그됩니다.

### RESETDICTIONARY

테이블에 대한 COMPRESS 속성이 YES이면 새 행 압축 사전이 빌드됩니다. 재구성 중 처리된 모든 행은 이 새 사전을 사용하여 압축되기 쉽습니다. 이 사전은 이전 사전을 바꿉니다. 테이블에 대한 COMPRESS 속성이 NO이고 테이블에 기존 압축 사전이 들어 있으면 REORG 처리는 이 사전을 제거하고 새로 재구성된 테이블의 모든 행은 압축되지 않은 형식이 됩니다. 기본 테이블 행에 저장되지 않은 LOB 데이터와 같은 일부 데이터는 압축할 수 없습니다.

LONGLOBDATA 옵션이 지정되지 않으면, 테이블 행 데이터만이 재구성됩니다. LONGLOBDATA 옵션이 지정되지 않으면 다음 표에서는 REORG 명령에서 RESETDICTIONARY 구문의 동작을 설명합니다.

표 64. REORG RESETDICTIONARY

압축	사전이 존재함	결과, 결론
Y	Y	새 사전*을 빌드합니다. 행을 압축합니다. DATA CAPTURE CHANGES 옵션이 CREATE TABLE 또는 ALTER TABLE문에 지정되면, 현재 사전이 보존됩니다(실행기록 압축 사전으로 참조됨).
Y	N	새 사전을 빌드합니다. 행을 압축합니다.
N	Y	사전을 제거합니다. 모든 행을 압축 해제합니다. DATA CAPTURE NONE 옵션이 CREATE TABLE 또는 ALTER TABLE문에 지정되면, 지정된 테이블의 실행기록 압축 사전도 제거됩니다.
N	N	영향을 미치지 않음, 모든 행을 압축 해제합니다.

\* - 사전이 존재하고 압축 속성이 사용 가능하지만 현재 테이블에 데이터가 없는 경우 RESETDICTIONARY 조작은 기존의 사전을 보존합니다. 내부 최소 레코드 길이보다 크기가 작은 행 및 압축해도 레코드 길이가 줄어들지 않는 행은 이 경우 "충분하지 않다"고 간주됩니다.

다음 표에서는 LONGLOBDATA 옵션 지정 시 REORG 명령에서 RESETDICTIONARY 구문의 동작에 대해 설명합니다.

표 65. LONGLOBDATA 옵션이 지정될 때 REORG RESETDICTIONARY

압축	테이블 행 데이터 사전 존재	XML 스토리지 오브젝트 사전 존재 <sup>1</sup>	데이터 사전	데이터 압축
Y	Y	Y	사전을 빌드합니다. <sup>2 3</sup> .	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
Y	Y	N	새 테이블 행 사전을 빌드하고 새 XML 사전을 작성합니다. <sup>3</sup> .	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
Y	N	Y	테이블 행 데이터 사전을 작성하고 새 XML 사전을 빌드합니다.	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
Y	N	N	사전을 작성합니다.	기존 데이터가 압축됩니다. 새 데이터가 압축됩니다.
N	Y	Y	사전을 제거합니다. 기존 및 새 데이터는 압축되지 않습니다.	기존 테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.
N	Y	N	테이블 행 사전을 제거합니다. 모든 데이터가 압축 해제됩니다.	기존 테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.
N	N	Y	XML 스토리지 오브젝트 사전을 제거합니다.	기존 테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.
N	N	N	영향을 미치지 않습니다.	기존 테이블 데이터가 압축 해제됩니다. 새 데이터가 압축되지 않습니다.

참고:

1. XML 컬럼이 DB2 V9.7 이상에서 테이블에 추가된 경우 또는 테이블이 온라인 테이블 이동을 사용하여 이주된 경우에만 테이블의 XML 스토리지 오브젝트의 압축 사전을 작성할 수 있습니다.
2. 사전이 존재하고 압축 속성이 사용 가능하지만 현재 테이블에 데이터가 없는 경우 RESETDICTIONARY 조작은 기존의 사전을 보존합니다. 내부 최소 레코드 길이보다 크기가 작은 행 및 압축해도 레코드 길이가 줄어들지 않는 행은 이 경우 충분하지 않다고 간주합니다.
3. DATA CAPTURE CHANGES 옵션을 CREATE TABLE 또는 ALTER TABLE문에 지정하면 현재 데이터 사전이 보존됩니다(실행기록 압축 사전이라고 함).

## ALL DBPARTITIONNUMS

db2nodes.cfg 파일에 지정된 모든 데이터베이스 파티션에서 조작을 수행하도록 지정합니다. 이는 노드 절이 지정되지 않는 경우 디폴트입니다.

## EXCEPT

노드 목록에 지정된 파티션을 제외하고 db2nodes.cfg 파일에 지정된 모든 데이터베이스 파티션에서 조작을 수행하도록 지정합니다.

## ON DATA PARTITION *partition-name*

재구성을 위한 데이터 파티션을 지정합니다.

이 옵션은 RECLAIM EXTENTS ONLY에만 사용됩니다.

지정된 테이블의 데이터 파티션 이름이 없는 경우, REORG가 실패하며 SQL0204N이 리턴됩니다.

## ON DBPARTITIONNUM | ON DBPARTITIONNUMS

데이터베이스 파티션 세트에 대해 조작을 수행합니다.

### db-partition-number1

데이터베이스 파티션 목록의 데이터베이스 파티션 번호를 지정합니다.

### db-partition-number2

두 번째 데이터베이스 파티션 번호를 지정하여 *db-partition-number1*에서 최고 *db-partition-number2*까지의 모든 데이터베이스 파티션이 데이터베이스 파티션 목록에 포함되도록 합니다.

## 예 :

데이터베이스 파티션 1, 3, 4로 구성된 데이터베이스 파티션 그룹에서 테이블을 재구성하십시오.

```
CALL SYSPROC.ADMIN_CMD ('REORG TABLE employee  
INDEX empid ON DBPARTITIONNUM (1,3,4)')
```

## 사용 시 참고사항

제한사항:

- 명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.
- REORG 유틸리티는 유형 2 연결의 경우 프로시저가 이유 코드 2와 함께 SQL3009N을 리턴하게 하는 조작이 시작될 때 COMMIT문을 발행합니다.
- REORG 유틸리티는 별칭 사용을 지원하지 않습니다.
- REORG TABLE 명령은 선언된 임시 테이블 또는 작성된 임시 테이블에 대해서는 지원되지 않습니다.



- REORG TABLE 명령은 뷰에 대해서는 사용할 수 없습니다.
- 테이블의 재구성은 범위 클러스터 테이블과는 호환 가능하지 않는데, 이는 이 테이블의 범위 영역이 항상 클러스터된 상태로 남아 있기 때문입니다.
- REORG TABLE은 LOB 및 인덱스를 포함하여 해당 테이블이 있는 테이블 스페이스의 온라인 백업이 수행되는 동안 DMS 테이블 스페이스의 파티션된 테이블에서 사용할 수 없습니다.
- REORG TABLE은 인덱스 확장자에 기초한 인덱스를 사용할 수 없습니다.
- 테이블이 REORG 보류 상태에 있는 경우 이 테이블에서 inplace REORG를 사용할 수 없습니다.
- 파티션된 테이블의 경우,
  - 테이블은 SYSCAT.TABLES에서 읽기/쓰기의 ACCESS\_MODE를 가지고 있어야 합니다.
  - 재구성은 접속 또는 접속 해제 조작에 따라 제한된 상태에 있는 데이터 파티션을 건너뛸 수 있습니다. 테이블 파티셔닝 절이 지정된 경우, 해당 파티션이 전체 액세스 가능해야 합니다.
  - 테이블 재구성 동안 오류가 발생하면, 일부 인덱스나 인덱스 파티션이 유효하지 않게 남습니다. 재구성이 최초 데이터 파티션의 교체 단계에 도달하거나 패스한 경우 테이블의 파티션되지 않은 인덱스는 유효하지 않은 것으로 표시됩니다. 이미 교체 단계에 도달하거나 패스한 데이터 파티션의 인덱스 파티션은 유효하지 않은 것으로 표시됩니다. 인덱스는 테이블 또는 데이터 파티션에 대한 다음 액세스에서 재빌드됩니다.
  - ALLOW NONE 액세스 모드 사용 시 인덱스 재구성 중 오류가 발생하면, 테이블의 일부 인덱스가 유효하지 않게 남습니다. 테이블에서 파티션되지 않은 RID 인덱스의 경우, 실패 시점에서 재구성 중인 인덱스만이 유효하지 않게 남습니다. 파티션되지 않은 블록 인덱스가 있는 MDC 테이블의 경우, 오류 발생 시 하나 이상의 블록 인덱스가 유효하지 않게 남습니다. 파티션된 인덱스의 경우, 재구성되는 데이터 파티션의 인덱스 오브젝트만이 유효하지 않게 남습니다. 유효하지 않은 것으로 표시된 인덱스는 테이블 또는 데이터 파티션에 대한 다음 액세스에서 재빌드됩니다.

테이블 재구성의 현재 진행 상태에 대한 정보는 데이터베이스 활동에 대한 실행기록 파일에 쓰여집니다. 이 실행기록 파일에는 각 재구성 이벤트에 대한 레코드가 들어 있습니다. 이 파일을 보려면 재구성 중인 테이블이 있는 데이터베이스에 대해 LIST HISTORY 명령을 실행하십시오.

또한 테이블 스냅샷을 사용하여 테이블 재구성의 진행 상태를 모니터링할 수 있습니다. 테이블 재구성 모니터링 데이터는 데이터베이스 모니터 테이블 전환 설정에 관계없이 기록됩니다.



오류가 발생할 경우 이 실행기록 파일에 SQLCA 덤프가 쓰여집니다. Inplace 테이블 재구성의 경우 상태는 PAUSED로 기록됩니다.

인덱스가 작성된 테이블이 여러 번 수정된 경우, 인덱스에 있는 데이터가 조각난 상태가 될 수 있습니다. 테이블이 인덱스에 관해 클러스터된 경우, 테이블과 인덱스는 클러스터 순서를 벗어날 수 있습니다. 이들 두 요인은 모두 인덱스를 사용하는 스캔 성능에 불리한 영향을 줄 수 있으며 인덱스 페이지 프리페치의 효과에 영향을 줄 수 있습니다. REORG INDEX 또는 REORG INDEXES를 사용하여 테이블에 있는 인덱스 중 하나 또는 전체를 재구성할 수 있습니다. 인덱스 재구성은 모든 단편화를 제거하고 실제 클러스터링을 리프 페이지로 리스토어합니다. REORGCHK를 사용하면 인덱스를 재구성할 필요가 있는지 판별하는 데 도움이 됩니다. 인덱스 재구성을 호출하기 전에 모든 데이터베이스 조작을 완료하고 모든 잠금을 해제해야 합니다. 이는 WITH HOLD로 열린 모든 커서를 닫고 COMMIT를 발행하거나 ROLLBACK을 발행하여 수행할 수 있습니다.

클래식 테이블 재구성(오프라인 재구성)은 재구성의 마지막 단계 동안 인덱스를 재빌드합니다. 그러나 Inplace 테이블 재구성(온라인 재구성)은 인덱스를 재빌드하지 않습니다. Inplace 테이블 재구성 완료 후 REORG INDEXES 명령을 발행하는 것이 좋습니다. Inplace 테이블 재구성은 비동기이므로, REORG INDEXES 명령을 발행하기 전에 inplace 테이블 재구성 완료를 확인하는 것이 필요합니다. Inplace 테이블 재구성 완료 이전에 REORG INDEXES 명령을 발행하면, 재구성이 실패합니다(SQLCODE -2219).

테이블이 여러 번 수정되어 데이터가 분할되고 액세스 성능이 현저하게 느려지게 되면 REORG TABLE 명령의 후보가 됩니다. 또한 변경된 인라인 길이를 이용하기 위해 구조화된 유형 컬럼의 인라인 길이를 변경한 후에는 이 유틸리티를 호출해야 합니다. 테이블을 재구성할 필요가 있는지 여부를 판별하려면 REORGCHK를 사용하십시오. REORG TABLE을 호출하기 전에 반드시 모든 데이터베이스 조작을 완료하고 모든 잠금을 릴리스하십시오. 이는 WITH HOLD로 열린 모든 커서를 닫고 COMMIT를 발행하거나 ROLLBACK을 발행하여 수행할 수 있습니다. 테이블을 재구성한 후 RUNSTATS를 사용하여 테이블 통계를 갱신하고 이 테이블을 사용하는 패키지를 리바인드하십시오. 재구성 유틸리티는 내재적으로 모든 커서를 닫습니다.

테이블 값 압축이 활성화되거나 비활성화되었기 때문에 테이블에 혼합 행 형식이 들어 있는 경우, 오프라인 테이블 재구성이 모든 기존 행을 목표 행 형식으로 변환할 수 있습니다.

테이블이 여러 데이터베이스 파티션에 분산되어 있으며 영향을 받는 데이터베이스 파티션 중 하나에서 테이블 재구성이 실패할 경우, 실패한 데이터베이스 파티션에서만 테이블 또는 인덱스 재구성이 롤백됩니다.

재구성이 실패할 경우 임시 파일이 삭제되어서는 안 됩니다. 데이터베이스 관리 프로그램이 데이터베이스를 복구할 때 이들 파일을 사용합니다.

인덱스 이름이 지정된 경우, 데이터베이스 관리 프로그램은 이 인덱스에 있는 순서에 따라 데이터를 재구성합니다. 성능을 최대화하려면 SQL 쿼리에서 자주 사용되는 인덱스를 지정하십시오. 인덱스 이름이 지정되지 않고 클러스터링 인덱스가 존재할 경우 데이터는 클러스터링 인덱스에 따라 정렬됩니다.

테이블의 PCTFREE 값은 페이지 당 지정된 여유 공간의 크기를 결정합니다. 이 값이 설정되어 있지 않으면 유틸리티는 각 페이지에서 가능한 많은 공간을 채웁니다.

테이블 재구성 후 테이블 스페이스 롤포워드 복구를 완료하려면 일반 및 대형 테이블 스페이스가 모두 롤포워드 복구에 사용할 수 있어야 합니다.

테이블에 COMPACT 옵션을 사용하지 않는 LOB 컬럼이 있는 경우, 테이블 재구성 후 LOB DATA 스토리지 오브젝트가 훨씬 더 커질 수 있습니다. 이는 행이 재구성된 순서와 사용된 테이블 스페이스의 유형(SMS 또는 DMS)의 결과가 될 수 있습니다.

XML 데이터의 인덱스는 REORG INDEXES/TABLE 명령으로 재작성될 수 있습니다. 자세한 내용은 『XML 데이터에서 인덱스 재작성』을 참조하십시오.

## ADMIN\_CMD 프로시저를 사용하는 RESET ALERT CONFIGURATION

### 명령

특정 오브젝트의 Health 표시기 설정값을 해당 오브젝트 유형의 현재 디폴트값으로 재설정하거나 오브젝트 유형의 현재 디폴트 Health 표시기 설정값을 설치 디폴트값으로 재설정합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 명령 또는 API는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 권한 부여

다음 중 하나가 필요합니다.

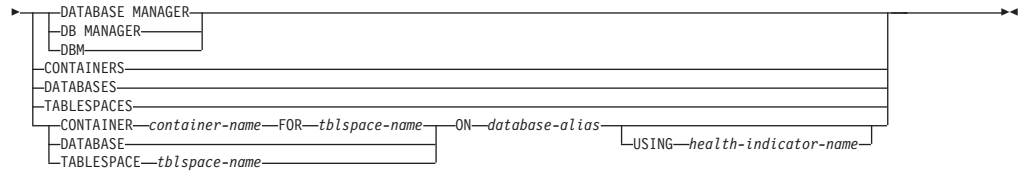
- *sysadm*
- *sysmaint*
- *sysctrl*

### 필수 연결

데이터베이스

### 명령 구문

```
►►—RESET ALERT—CONFIGURATION—FOR—  
                  —CONFIG—  
                  —CFG—
```



## 명령 매개변수

### DATABASE MANAGER | DB MANAGER | DBM

데이터베이스 관리 프로그램에 대한 경보 설정값을 재설정합니다.

### CONTAINERS

데이터베이스 관리 프로그램이 관리하는 모든 테이블 스페이스 컨테이너에 대한 경보 설정값을 설치 디폴트값으로 재설정합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 테이블 스페이스 컨테이너에 적용됩니다. 사용자 정의 설정은 CONTAINER *container-name* FOR *tblspace-name* ON *database-alias* 절을 사용하여 정의됩니다.

### DATABASES

데이터베이스 관리 프로그램이 관리하는 모든 데이터베이스에 대한 경보 설정값을 재설정합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 데이터베이스에 적용됩니다. 사용자 정의 설정값은 DATABASE ON *database-alias* 절을 사용하여 정의됩니다.

### TABLESPACES

데이터베이스 관리 프로그램이 관리하는 모든 테이블 스페이스에 대한 디폴트 경보 설정값을 설치 디폴트값으로 재설정합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 테이블 스페이스에 적용됩니다. 사용자 정의 설정은 TABLESPACE *tblspace-name* ON *database-alias* 절을 사용하여 정의됩니다.

### CONTAINER *container-name* FOR *tblspace-name* ON *database-alias*

ON *database-alias* 절을 사용하여 지정된 데이터베이스에서 FOR *tblspace-name* 절을 사용하여 지정된 테이블 스페이스의 경우 *container-name*이라는 테이블 스페이스 컨테이너의 경보 설정을 재설정합니다. 이 테이블 스페이스 컨테이너에 사용자 정의 설정값이 들어 있으면 이들 설정값을 제거하고 현재 테이블 스페이스 컨테이너 디폴트값을 사용합니다.

### DATABASE ON *database-alias*

ON *database-alias* 절을 사용하여 지정된 데이터베이스의 경보 설정을 재설정합니다. 이 데이터베이스가 사용자 정의 설정값을 가지고 있으면 이들 설정값을 제거하고 설치 디폴트값을 사용합니다.

### TABLESPACE *tblspace-name* ON *database-alias*

ON *database-alias* 절을 사용하여 지정된 데이터베이스에서 *tblspace-name*이

라는 테이블 스페이스의 경보 설정을 재설정합니다. 이 테이블 스페이스가 사용자 정의 설정값을 가지고 있으면 이들 설정값을 제거하고 설치 디폴트값을 사용합니다.

#### **USING** *health-indicator-name*

경보 구성이 재설정될 Health 표시기 세트를 지정합니다. Health 표시기 이름은 두 문자 오브젝트 ID 다음에 표시기가 측정하는 것을 설명하는 이름으로 구성됩니다. 예를 들면, 다음과 같습니다.

```
db.sort_privmem_util
```

이 옵션을 지정하지 않으면 지정된 오브젝트 또는 오브젝트 유형에 대한 모든 Health 표시기가 재설정됩니다.

#### **예 :**

ADMIN\_CMD 프로시저가 있는 데이터베이스를 소유하는 데이터베이스 관리 프로그램에 대한 경보 설정값을 재설정합니다.

```
CALL SYSPROC.ADMIN_CMD( 'reset alert cfg for dbm' )
```

#### **사용 시 참고사항**

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

ADMIN\_CMD 프로시저가 서버에서만 실행되므로 *database-alias*는 서버의 카탈로그에 정의된 로컬 데이터베이스여야 합니다.

## **ADMIN\_CMD 프로시저를 사용하는 RESET DATABASE CONFIGURATION 명령**

특정 데이터베이스의 구성을 시스템 디폴트값으로 재설정합니다.

### **범위**

이 명령은 응용프로그램이 연결된 데이터베이스 파티션에만 영향을 미칩니다.

### **권한 부여**

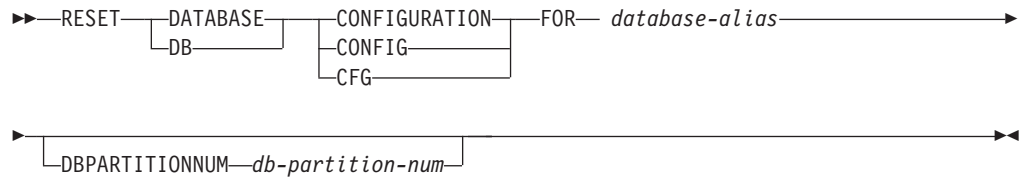
다음 중 하나가 필요합니다.

- SYSADM
- SYSCTRL
- SYSMANT

### **필수 연결**

데이터베이스

## 명령 구문



## 명령 매개변수

### FOR *database-alias*

해당 구성이 시스템 디폴트값으로 재설정될 데이터베이스의 별명을 지정합니다. 데이터베이스 별명은 서버의 카탈로그에 정의되어 있어야 하며 서버에 있는 로컬 데이터베이스를 나타내야 합니다.

### DBPARTITIONNUM *db-partition-num*

데이터베이스 구성 재설정이 특정 데이터베이스 파티션에 적용되는 경우, 이 매개변수를 사용할 수 있습니다. 이 매개변수가 제공되지 않으면, 모든 데이터베이스 파티션에서 재설정이 유효합니다.

## 예 :

서버에서 별명 SAMPLE로 카탈로그된 데이터베이스의 구성을 재설정합니다.

```
CALL SYSPROC.ADMIN_CMD( 'reset db cfg for SAMPLE' )
```

## 사용 시 참고사항

데이터베이스 구성 매개변수 목록을 보거나 인쇄하려면 `SYSIBMADM.DBCFG` 관리 뷰를 사용하십시오.

구성 가능 매개변수의 값을 변경하려면, `UPDATE DATABASE CONFIGURATION` 명령을 사용하십시오.

데이터베이스 구성 파일에 대한 변경사항은 메모리에 로드된 후에만 유효하게 됩니다. 이렇게 되려면 데이터베이스에서 모든 응용프로그램의 연결을 끊어야 합니다.

오류가 발생하면 데이터베이스 구성 파일이 변경되지 않습니다.

체크섬이 유효하지 않으면 데이터베이스 구성 파일을 재설정할 수 없습니다. 적절한 명령을 사용하지 않고 데이터베이스 구성 파일을 변경할 경우 이러한 일이 발생할 수 있습니다. 이러한 경우 데이터베이스를 리스토어하여 데이터베이스 구성 파일을 재설정해야 합니다.



## 사용 시 참고사항

이 명령은 설치 프로그램에 의해 설정된 모든 매개변수를 재설정합니다. 이는 DB2를 재시작할 때 오류 메시지가 리턴되게 할 수 있습니다. 예를 들어, **svccname** 매개변수를 재설정할 경우 사용자는 DB2를 재시작하려고 할 때 SQL5043N 오류 메시지를 받습니다.

이 명령을 실행하기 전에 SYSIBMADM.DBMCFG 관리 뷰의 출력을 파일에 저장하여 기존 설정값을 참조할 수 있도록 하십시오. ADMIN\_CMD 프로시저를 통해 UPDATE DATABASE MANAGER CONFIGURATION 명령을 사용하여 개별 설정값을 갱신할 수 있습니다.

설치 프로그램에 의해 설정된 **svccname** 매개변수를 사용자가 수정하는 것은 바람직하지 않습니다.

데이터베이스 관리 프로그램 구성 매개변수 목록을 보거나 인쇄하려면 SYSIBMADM.DBMCFG 관리 뷰를 사용하십시오. 구성 가능 매개변수의 값을 변경하려면 ADMIN\_CMD 프로시저를 통해 UPDATE DATABASE MANAGER CONFIGURATION 명령을 사용하십시오.

이들 매개변수에 대한 자세한 정보는 구성 매개변수의 요약 목록 및 개별 매개변수를 참조하십시오.

데이터베이스 관리 프로그램 구성 파일에 대한 일부 변경사항은 메모리로 로드된 후에 만 유효하게 됩니다. 어떤 매개변수가 온라인으로 구성 가능하고 어떤 매개변수가 그렇지 않은지에 대한 자세한 정보는 구성 매개변수 요약을 참조하십시오. 즉시 재설정되지 않는 서버 구성 매개변수는 db2start 실행 중 재설정됩니다. 클라이언트 구성 매개변수의 경우, 매개변수는 다음 번에 응용프로그램을 재시작할 때 재설정됩니다. 클라이언트가 명령행 처리기일 경우 TERMINATE를 호출해야 합니다.

오류가 발생하면 데이터베이스 관리 프로그램 구성 파일이 변경되지 않습니다.

체크섬이 유효하지 않으면 데이터베이스 관리 프로그램 구성 파일을 재설정할 수 없습니다. 이는 구성 파일을 수동으로 편집하고 적절한 명령을 사용하지 않을 경우 발생할 수 있습니다. 체크섬이 유효하지 않으면, 인스턴스를 재작성해야 합니다.

## ADMIN\_CMD 프로시저를 사용하는 REWIND TAPE 명령

백업 및 리스토어 조작을 위한 테이프를 스트리밍 테이프 디바이스로 되감습니다. 이 명령은 Windows 운영 체제에서만 지원됩니다.

## 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

## 필수 연결

데이터베이스

## 명령 구문

```
▶▶ REWIND TAPE [ON device] ▶▶
```

## 명령 매개변수

*ON device*

유효한 테이프 디바이스 이름을 지정합니다. 디폴트값은 `WWW.WTAPE0`입니다. 지정된 디바이스는 서버에 관계되어야 합니다.

예 :

'`WWW.WTAPE1`'으로 이름 지정된 디바이스에서 테이프를 되감으십시오.

```
CALL SYSPROC.ADMIN_CMD( 'rewind tape on WWW.WTAPE1' )
```

## 사용 시 참고사항

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

## ADMIN\_CMD 프로시저를 사용하는 RUNSTATS 명령

테이블 및/또는 연관된 인덱스 또는 통계적 뷰의 등록 정보에 대한 통계를 갱신하십시오. 이러한 등록 정보에는 레코드 수, 페이지 수 및 평균 레코드 길이가 포함됩니다. 옵티마이저는 데이터에 대한 액세스 경로를 판별할 때 이들 통계를 사용합니다.

테이블의 경우, 테이블을 재구성한 후 또는 테이블에 갱신사항이 많이 있을 때 이 유틸리티를 호출해야 합니다. 통계적 뷰의 경우, 기본이 되는 테이블에 대한 변경사항이 뷰에서 리턴된 행에 실질적으로 영향을 줄 때 이 유틸리티를 호출해야 합니다. 이 뷰는 ALTER VIEW 명령을 사용하여 쿼리 최적화에 사용될 수 있도록 미리 설정되어 되어 있어야 합니다.



## 범위

이 명령은 `db2nodes.cfg` 파일에서 모든 데이터베이스 파티션으로부터 발행될 수 있습니다. 카탈로그 데이터베이스 파티션에 있는 카탈로그를 갱신하는 데 사용될 수 있습니다.

테이블의 경우, 이 명령은 명령을 호출한 데이터베이스 파티션의 테이블에 대한 통계를 수집합니다. 해당 테이블이 이 데이터베이스 파티션에 존재하지 않으면 데이터베이스 파티션 그룹의 첫 번째 데이터베이스 파티션이 선택됩니다.

뷰의 경우, 이 명령은 모든 참여 중인 데이터베이스 파티션에 있는 테이블의 데이터를 사용하여 통계를 수집합니다.

## 권한 부여

테이블의 경우, 다음 중 하나이어야 합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*
- *sqladm*
- 테이블에 대한 CONTROL 특권
- LOAD 권한

해당 연결에 존재하는 선언된 임시 테이블에서 이 명령을 사용하는 데 명시적 특권이 필요하지는 않습니다.

통계 뷰의 경우, 다음 중 하나이어야 합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*
- *sqladm*
- 통계적 뷰에 대한 CONTROL 특권

그 밖에, 통계적 뷰에서 행에 액세스하려면 적절한 특권이 있어야 합니다. 특히, 통계적 뷰 정의에 참조된 각 테이블, 통계적 뷰 또는 별명에 대해 사용자는 다음 특권 중 하나가 있어야 합니다.

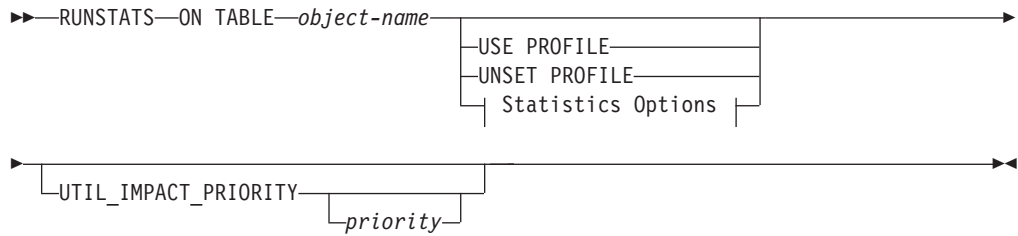
- *dataaccess*
- CONTROL

• SELECT

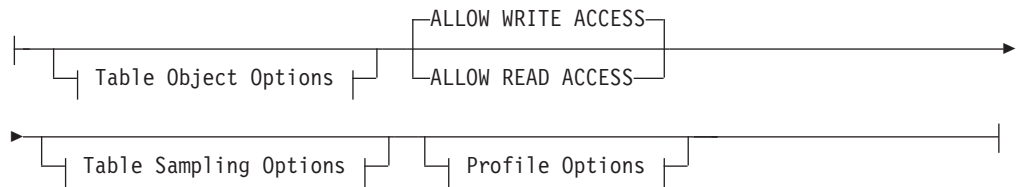
## 필수 연결

데이터베이스

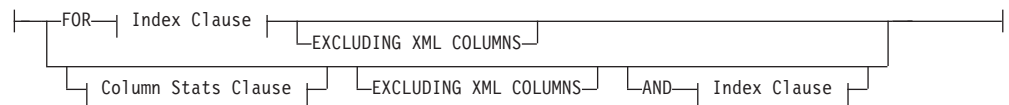
## 명령 구문



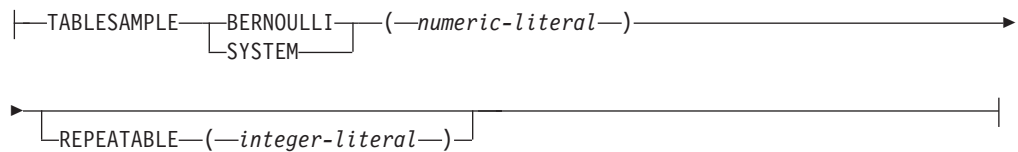
### Statistics Options:



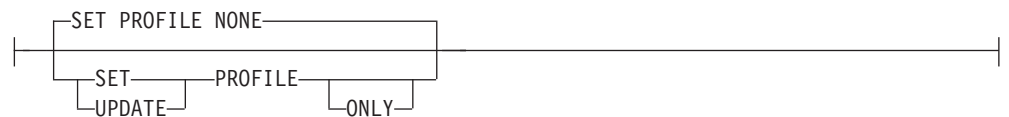
### Table Object Options:



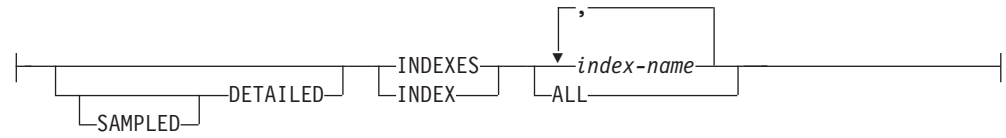
### Table Sampling Options:



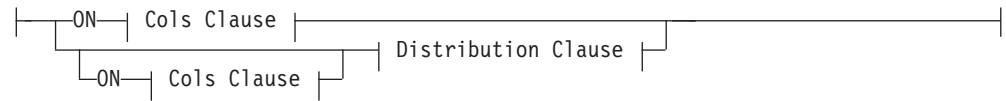
### Profile Options:



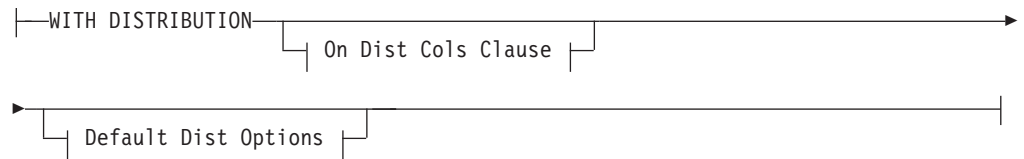
**Index Clause:**



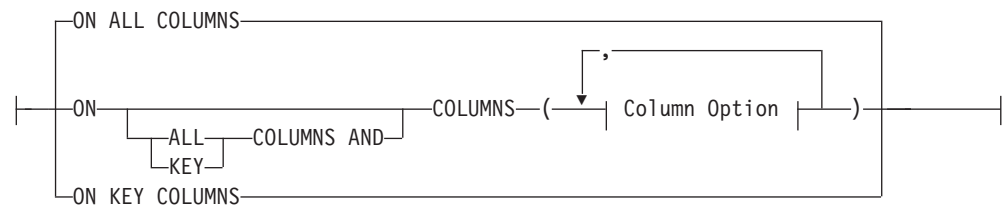
**Column Stats Clause:**



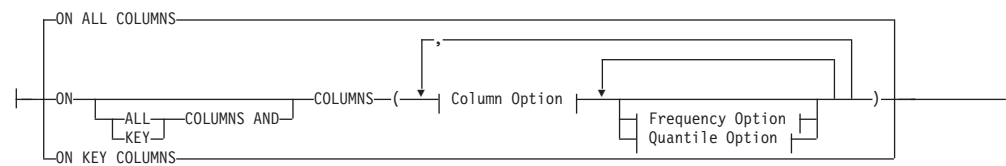
**Distribution Clause:**



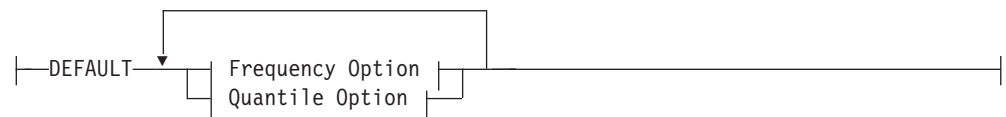
**On Col's Clause:**



**On Dist Col's Clause:**



**Default Dist Option:**



### Frequency Option:

|—NUM\_FREQVALUES—*integer*—|

### Quantile Option:

|—NUM\_QUANTILES—*integer*—|

### Column Option:

|—*column-name*—|

|—LIKE STATISTICS—|

|—(—*column-name*—)—|

## 명령 매개변수

### *object-name*

통계가 수집될 테이블 또는 통계 뷰를 식별합니다. 이는 계층 구조 테이블어서는 안 됩니다. 유형이 지정된 테이블의 경우, *object-name*는 테이블 계층 구조의 루트 테이블의 이름이어야 합니다. *schema.object-name* 양식의 완전한 이름 또는 별명을 사용해야 합니다. 스키마는 테이블이 작성된 사용자 이름입니다.

### *index-name*

테이블에 정의된 기존 인덱스를 식별합니다. *schema.index-name* 양식의 완전한 이름을 사용해야 합니다. 뷰에는 이 옵션을 사용할 수 없습니다.

## USE PROFILE

이 옵션은 RUNSTATS가 이전에 저장된 통계 프로파일을 사용하여 테이블 또는 통계 뷰에 대한 통계를 수집할 수 있도록 합니다. 통계 파일은 SET PROFILE 옵션을 사용하여 작성되고 UPDATE PROFILE 옵션을 사용하여 갱신됩니다.

## UNSET PROFILE

기존 통계 프로파일을 제거하려면 이 옵션을 지정하십시오. 예를 들면, 다음과 같습니다.

```
runstats on tablemyschema.mytable unset profile
```

## FOR INDEXES

인덱스 전용 통계를 수집하고 갱신합니다. 테이블에서 이전에 수집된 테이블 통계가 없는 경우 기본 테이블 통계가 수집되기도 합니다. 이들 기본 통계에는 분산 통계가 포함되지 않습니다. 뷰에는 이 옵션을 사용할 수 없습니다.

## AND INDEXES

테이블 및 인덱스 둘 다에 대한 통계를 수집하고 갱신합니다. 뷰에는 이 옵션을 사용할 수 없습니다.

## DETAILED

확장된 인덱스 통계를 계산합니다. 이는 비교적 대형 인덱스에 대해 수집되는 CLUSTERFACTOR 및 PAGE\_FETCH\_PAIRS 통계입니다. 뷰에는 이 옵션을 사용할 수 없습니다.

## SAMPLED

이 옵션은 DETAILED 옵션과 함께 사용될 때 RUNSTATS가 확장된 인덱스 통계를 컴파일할 때 CPU 샘플링 기술을 사용할 수 있도록 합니다. 이 옵션을 지정하지 않으면 인덱스의 모든 항목을 조사하여 확장된 인덱스 통계를 계산합니다. 뷰에는 이 옵션을 사용할 수 없습니다.

## ON ALL COLUMNS

적당한 모든 컬럼에서 통계를 수집하려면, ON ALL COLUMNS 절을 사용하십시오. 컬럼은 기본 통계 컬렉션(Cols 절에서)에 대해 또는 WITH DISTRIBUTION 절(Dist Cols 절에서)과 함께 지정될 수 있습니다. 해당 컬럼에만 해당되는 절이 지정되지 않은 경우 디폴트 옵션은 ON ALL COLUMNS 스펙입니다.

On Cols 절에 지정된 경우 WITH DISTRIBUTION 절에서 특정 컬럼을 선택한 경우를 제외하고는 모든 컬럼에 대해 기본 컬럼 통계만 수집됩니다. WITH DISTRIBUTION 절에 지정된 컬럼에 대해서는 또한 기본 및 분산 통계가 수집됩니다.

WITH DISTRIBUTION ON ALL COLUMNS을 지정하면 모든 적당한 컬럼에 대해 기본 통계 및 분산 통계가 수집됩니다. On Cols 절에 지정된 내용은 중복되므로 필요하지 않습니다.

## ON COLUMNS

이 절을 통해 사용자는 통계를 수집할 컬럼 목록을 지정할 수 있습니다. 컬럼 그룹을 지정하면 그룹에 대한 구별 값 수가 수집됩니다. 인덱스 통계를 수집하지 않고 테이블에서 RUNSTATS를 수행하고 통계가 수집될 컬럼의 서브세트를 지정하면 다음이 수행됩니다.

1. RUNSTATS 명령에 지정되지 않았지만 인덱스에서 첫 번째 컬럼인 컬럼에 대한 통계는 재설정되지 않습니다.
2. RUNSTATS 명령에 지정되지 않은 모든 다른 컬럼에 대한 통계는 재설정됩니다.

이 절은 On Cols 절 및 On Dist Cols 절에서 사용될 수 있습니다. 컬럼 그룹에 대한 분산 통계 수집은 현재 지원되지 않습니다.

컬럼 그룹에 XML 유형 컬럼이 지정된 경우, 그룹에 대한 구별 값을 수집하기 위해 XML 유형 컬럼을 무시합니다. 그러나 컬럼 그룹에 있는 XML 유형 컬럼을 위해 기본 XML 컬럼 통계는 수집됩니다.

### **EXCLUDING XML COLUMNS**

이 절은 통계 컬렉션에서 모든 XML 유형 컬럼을 생략할 수 있도록 합니다. 이 절은 XML 데이터를 포함하려면 더 많은 시스템 자원이 필요할 수 있으므로 비XML 컬럼에 대한 통계 컬렉션을 사용하도록 조정합니다. EXCLUDING XML COLUMNS 절은 통계 컬렉션에 XML 컬럼을 지정하는 다른 절보다 우선순위를 갖습니다. 예를 들어, EXCLUDING XML COLUMNS 절을 사용하며 ON COLUMNS 절을 사용하여 XML 유형 컬럼을 지정하거나 ON ALL COLUMNS 절을 사용할 경우, 통계 컬렉션 중 모든 XML 유형 컬럼이 무시됩니다.

### **ON KEY COLUMNS**

특정 컬럼을 나열하는 대신 테이블에 정의된 모든 인덱스를 나열하는 컬럼에서 통계를 수집하도록 선택할 수 있습니다. 여기에서는 쿼리에서 중요한 컬럼이 테이블에서 인덱스를 작성하는 데 사용되는 컬럼이기도 하다고 간주됩니다. 테이블에 인덱스가 없는 경우 이는 공백 목록과 같으며 컬럼 통계는 수집되지 않습니다. 이 절은 on-cols-clause 또는 on-dist-cols-clause에서 사용할 수 있습니다. 두 절에 모두 지정될 경우 WITH DISTRIBUTION 절이 기본 및 분산 통계 컬렉션을 지정하는 데 사용되므로 on-cols-clause에서 중복됩니다. XML 유형 컬럼은 정의이며 키 컬럼이 아니므로 ON KEY COLUMNS 절에 의한 통계 컬렉션에 포함되지 않습니다. 뷰에는 이 옵션을 사용할 수 없습니다.

#### *column-name*

테이블 또는 통계 뷰에 있는 컬럼의 이름. 존재하지 않는 컬럼 또는 잘못된 입력된 컬럼 이름과 같이 부적당한 컬럼 이름을 통계 컬렉션에 지정하면 오류(-205)가 리턴됩니다. 두 개의 컬럼 목록을 하나는 분산을 포함하고 다른 하나는 분산을 포함하지 않고 지정할 수 있습니다. 컬럼이 WITH DISTRIBUTION 절과 연관되지 않은 목록에 지정된 경우 기본 컬럼 통계만 수집됩니다. 컬럼이 두 목록에 모두 나타날 경우 분산 통계가 수집됩니다(NUM\_FREQVALUES 및 NUM\_QUANTILES가 0으로 설정되지 않은 경우).

### **NUM\_FREQVALUES**

수집할 최대 빈도 수 값을 정의합니다. ON COLUMNS 절에서 개별 컬럼에 대해 지정될 수 있습니다. 개별 컬럼에 대한 값이 지정되어 있지 않으면 DEFAULT 절에 지정되어 있는 값에서 빈도 한계 값을 선택합니다. 여기에도 지정되어 있지 않으면 NUM\_FREQVALUES 데이터베이스 구성 매개변수에 설정된 값이 최대 빈도 수 값으로 수집됩니다.

### **NUM\_QUANTILES**

수집할 최대 분산 Quantile 수 값을 정의합니다. ON COLUMNS 절에서 개별

컬럼에 대해 지정될 수 있습니다. 개별 컬럼에 대한 값이 지정되어 있지 않으면 DEFAULT절에 지정되어 있는 값에서 Quantile 한계 값을 선택합니다. 여기에도 지정되어 있지 않으면 NUM\_QUANTILES 데이터베이스 구성 매개변수에 설정된 값이 최대 Quantile 수 값으로 수집됩니다.

#### **WITH DISTRIBUTION**

이 절은 컬럼에서 기본 통계와 분산 통계가 모두 수집되도록 지정합니다. ON COLUMNS 절을 지정하지 않으면 테이블 또는 통계 뷰의 모든 컬럼(CLOB 및 LONG VARCHAR과 같이 부적당한 컬럼은 제외)에서 분산 통계가 수집됩니다. ON COLUMNS절을 지정하면 제공된 컬럼 목록(통계 콜렉션에 부적당한 컬럼은 제외)에서만 분산 통계가 수집됩니다. 이 절을 지정하지 않으면 기본 통계만 수집됩니다.

컬럼 그룹에서 분산 통계 콜렉션은 현재 지원되지 않습니다. WITH DISTRIBUTION ON COLUMNS절에 컬럼 그룹이 지정되어 있을 때 분산 통계는 수집되지 않습니다.

#### **DEFAULT**

NUM\_FREQVALUES 또는 NUM\_QUANTILES가 지정된 경우 이들 값은 해당 컬럼에 대해 수집될 최대 빈도 및 Quantile 수 통계를 판별하는 데 사용됩니다(ON COLUMNS절에서 개별 컬럼에 대해 지정되지 않은 경우). DEFAULT절이 지정되어 있지 않으면 해당 데이터베이스 구성 매개변수에 있는 값을 사용합니다.

#### **LIKE STATISTICS**

이 옵션을 지정하면 추가 컬럼 통계가 수집됩니다. 이들 통계는 SYSSTAT.COLUMNS의 SUB\_COUNT 및 SUB\_DELIM\_LENGTH 통계입니다. 1바이트 문자 세트(SBCS), FOR BIT DATA 또는 UTF-8의 코드 페이지 속성이 있는 유형 CHAR 및 VARCHAR의 컬럼에 대한 통계가 수집됩니다. 쿼리 옵티마이저는 이들 통계를 사용하여 "column LIKE '%xyz'" 및 "column LIKE '%xyz%'" 유형의 술어에 대한 선택 빈도 추정을 향상시킵니다.

#### **ALLOW WRITE ACCESS**

통계가 계산되는 동안 다른 사용자가 테이블에서 읽고 쓸 수 있도록 지정합니다. 통계 뷰의 경우 이는 뷰 정의에 참조된 기본 테이블입니다.

삽입, 갱신 또는 삭제가 동시에 대량으로 발생하는 테이블에는 ALLOW WRITE ACCESS 옵션을 사용하지 마십시오. RUNSTATS 명령은 먼저 테이블 통계를 수행한 다음 인덱스 통계를 수행합니다. 테이블 통계와 인덱스 통계를 수집하는 사이에 상태가 변경되면 불일치가 발생할 수 있습니다. 쿼리를 최적화하기 위해서는 최신 통계가 있어야 하지만 통계의 일지도 중요합니다. 따라서 삽입, 갱신 또는 삭제량이 최소일 때 통계를 수집해야 합니다.

## ALLOW READ ACCESS

통계가 계산되는 동안 다른 사용자가 테이블에 읽기 전용 액세스를 할 수 있도록 지정합니다. 통계 뷰의 경우 이는 뷰 정의에 참조된 기본 테이블입니다.

## TABLESAMPLE BERNOULLI

이 옵션은 RUNSTATS가 테이블 또는 통계 뷰에서 샘플 행에 대한 통계를 수집할 수 있도록 합니다. BERNOULLI 샘플링은 각 행을 개별적으로 검토하여 확률이 P/100인 행은 포함시키고 확률이 1-P/100인 행은 제외시킵니다(여기서 P는 numeric-literal의 값임). 따라서 numeric-literal의 값이 10(10퍼센트 샘플을 나타냄)으로 평가된 경우 확률이 0.1인 행은 포함되고 확률이 0.9인 행은 제외됩니다. 선택적 REPEATABLE절이 지정된 경우를 제외하고는 RUNSTATS를 실행할 때마다 보통 다른 샘플의 테이블이 생성됩니다. 테이블 스캔을 통해 모든 데이터 페이지가 검색되지만 통계 컬렉션에는 numeric-literal 매개변수를 통해 지정된 퍼센트의 행만 사용됩니다.

## TABLESAMPLE SYSTEM

이 옵션은 RUNSTATS가 테이블에서 샘플 데이터 페이지에 대한 통계를 수집할 수 있도록 합니다. SYSTEM 샘플링은 각 페이지를 개별적으로 검토하여 확률이 P/100인 페이지는 포함시키고 확률이 1-P/100인 페이지는 제외시킵니다(여기서 P는 numeric-literal의 값임). 선택적 REPEATABLE절이 지정된 경우를 제외하고는 RUNSTATS를 실행할 때마다 보통 다른 샘플의 테이블이 생성됩니다. 샘플 크기는 괄호 안의 numeric-literal 매개변수에 의해 제어되며 이는 테이블의 대략적인 백분을 P를 나타냅니다. numeric-literal 매개변수를 통해 지정된 퍼센트의 데이터 페이지만 검색되며 통계 컬렉션에 사용됩니다.

통계 뷰에서 SYSTEM 샘플링은 단일 기본 테이블에서 정의를 선택하는 뷰로 제한됩니다. 뷰가 다중 테이블을 포함하면, 다음의 경우 SYSTEM 샘플링도 가능합니다.

- 테이블 간에 정의된 참조 무결성 제한조건에 포함된 모든 기본 키와 외부 키에 대해 등호 술어를 사용하여 테이블이 조인됩니다.
- 어떤 검색 조건도 관계의 모든 상위 테이블 행을 필터하지 않습니다.
- 상위 테이블이 아닌 단일 하위 테이블은 모든 테이블 간에 식별될 수 있습니다.

통계 뷰가 이들 조건을 충족시키지 않는 경우, BERNOULLI 샘플링이 대신 사용되며 경고가 리턴됩니다(SQL2317W).

## REPEATABLE (*integer-literal*)

TABLESAMPLE절에 REPEATABLE절을 추가하면 RUNSTATS가 반복 실행할 때 동일한 샘플이 리턴됩니다. *integer-literal* 매개변수는 음수가 아닌 정수로서 샘플링에 사용되는 시드를 나타냅니다. 음수 시드를 패스하면 오류가 발생합니다(SQL1197N). 마지막 TABLESAMPLE REPEATABLE이 실행된 후 테이블 또는 통계 뷰에 대한 활동으로 테이블 또는 통계 뷰 데이터가 변경된



경우 샘플 세트는 반복 가능한 RUNSTATS 호출 사이에서 계속 다를 수 있습니다. 또한 일치된 결과를 얻기 위해서는 BERNOULLI 또는 SYSTEM 키워드가 지정한 대로 샘플을 확보하는 방법도 동일해야 합니다.

#### *numeric-literal*

numeric-literal 매개변수는 가져올 샘플의 크기를 백분을 P로 지정합니다. 이 값은 100 이하의 양수이어야 하며 1과 0 사이의 숫자가 될 수 있습니다. 예를 들어, 값 0.01은 1/100퍼센트를 나타내고 평균적으로 10,000개의 행 중 하나의 행이 샘플링되는 것을 의미합니다. 값 0 또는 100은 TABLESAMPLE BERNOULLI 또는 TABLESAMPLE SYSTEM이 지정되었는지 여부에 관계 없이 DB2 데이터베이스 시스템에 의해 샘플링이 지정되지 않은 것처럼 처리됩니다. 100보다 크거나 0보다 작은 값은 DB2에 의해 오류(SQL1197N)로 처리됩니다.

#### **SET PROFILE NONE**

이 RUNSTATS 호출에 대해 통계 프로파일이 설정되지 않음을 지정합니다.

#### **SET PROFILE**

RUNSTATS가 시스템 카탈로그 테이블에서 특정 통계 프로파일을 생성하고 저장할 수 있도록 하며 RUNSTATS 명령 옵션을 실행하여 통계를 수집합니다.

#### **SET PROFILE ONLY**

RUNSTATS가 RUNSTATS 명령 옵션을 실행하지 않고 시스템 카탈로그 테이블에 특정 프로파일을 생성 및 저장할 수 있게 합니다.

#### **UPDATE PROFILE**

RUNSTATS가 시스템 카탈로그 테이블에서 기존 통계 프로파일을 수정할 수 있도록 하고 갱신된 통계 프로파일의 RUNSTATS 명령 옵션을 실행하여 통계를 수집합니다.

#### **UPDATE PROFILE ONLY**

갱신된 통계 프로파일의 RUNSTATS 명령 옵션을 실행하지 않고 RUNSTATS가 시스템 카탈로그 테이블에서 기존 통계 프로파일을 수정할 수 있도록 합니다.

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

RUNSTATS가 *priority*가 지정하는 레벨에서 조절되도록 지정합니다. *priority*는 1 - 100 범위의 수이며 100은 가장 높은 우선순위를 나타내고 1은 가장 낮은 우선순위를 나타냅니다. 이 우선순위는 유틸리티가 필요로 하는 조절 기능의 양입니다. 동일한 우선순위의 유틸리티는 모두 동일한 양으로 조절되지만 하위 우선순위의 유틸리티는 상위 우선순위의 유틸리티보다 더 많이 조절됩니다. *priority*가 지정되지 않으면 RUNSTATS는 디폴트 우선순위 50을 갖게 됩니다. UTIL\_IMPACT\_PRIORITY 키워드를 생략하면 조절 지원이 없이 RUNSTATS 유틸리티를 호출합니다. UTIL\_IMPACT\_PRIORITY 키워드를 지

정하지만 `util_impact_lim` 구성 매개변수를 100으로 설정할 경우 유틸리티는 조절되지 않고 실행됩니다. 뷰에는 이 옵션을 사용할 수 없습니다.

파티션된 데이터베이스에서 `RUNSTATS` 명령을 테이블에서 사용하면 단일 데이터베이스 파티션에 대해서만 통계를 수집합니다. `RUNSTATS` 명령이 실행된 데이터베이스 파티션에 해당 테이블의 파티션이 있으면 명령은 이 데이터베이스 파티션에서 실행됩니다. 그렇지 않으면 명령은 해당 테이블이 파티션된 데이터베이스 파티션 그룹의 첫 번째 데이터베이스 파티션에서 실행됩니다.

## 예 :

인덱스에 사용된 모든 컬럼과 모든 인덱스에서 통계를 수집하십시오.

```
CALL SYSPROC.ADMIN_CMD ('RUNSTATS ON TABLE db2user.employee  
ON KEY COLUMNS and INDEXES ALL')
```

## 사용 시 참고사항

1. 파티션된 테이블에 접속이 해제된 파티션이 있을 때, 정리할 필요가 있는 접속 해제된 데이터 파티션에 계속 속해 있는 인덱스 키는 통계에서 키로 계수되지 않습니다. 이들 키는 표시되지 않으며 더 이상 테이블에 포함되어 있지 않으므로 계수되지 않습니다. 결국 비동기 인덱스 정리에 의해 인덱스에서 제거됩니다. 따라서 비동기 인덱스 정리가 실행되기 전에 수집된 통계는 잘못된 것입니다. 비동기 인덱스 정리가 완료되기 전에 `RUNSTATS` 명령이 발행될 경우 부정확한 통계에 따른 인덱스 재구성 또는 인덱스 정리에 대한 거짓 알람을 생성할 수 있습니다. 비동기 인덱스 정리가 실행된 후, 정리할 필요가 있는 접속 해제된 데이터 파티션에 여전히 속해 있는 인덱스 키는 모두 제거되므로 인덱스를 재구성할 필요가 없도록 합니다.

파티션된 테이블의 경우, 접속 해제된 데이터 파티션이 있는 상태에서 올바른 인덱스 통계를 생성하려면 비동기 인덱스 정리가 완료된 후 `RUNSTATS` 명령을 발행하는 것이 바람직합니다. 테이블에 접속 해제된 데이터 파티션이 있는지 여부를 판별하려면 `SYSDATAPARTITIONS` 테이블의 상태 필드를 점검하여 I(인덱스 정리) 또는 D(중속 MQT와 접속 해제됨)를 찾아 볼 수 있습니다.

`RUNSTATS` 명령은 파티션된 인덱스의 모든 인덱스 파티션의 통계를 수집합니다. 파티션된 인덱스의 `SYSTAT.INDEXES` 뷰의 통계는 `FIRSTKEYCARD`, `FIRST2KEYCARD`, `FIRST3KEYCARD`, `FIRST4KEYCARD` 및 `FULLKEYCARD` 통계를 제외한 인덱스 파티션을 표시합니다. 이 통계는 카디널리티(cardinality) 개략에서 사용되므로, 전체 인덱스에 대한 것이며 인덱스 파티션에 대한 것은 아닙니다. 분산된 인덱스의 분산 통계(자주 사용되는 값 및 quantiles)는 수집되지 않지만, `RUNSTATS`가 테이블에서 실행되는 경우 수집됩니다. 파티션된 인덱스의 선행 컬럼에 대한 통계는 파티션되지 않은 인덱스의 선행 컬럼에 대한 통계만큼 정확하지 않을 수 있습니다.

2. 명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.
3. RUNSTATS 명령을 실행하는 것이 좋습니다.
  - 상당한 부분이 수정된 테이블(예를 들어, 여러 번 갱신되었거나 상당한 데이터 량이 삽입 또는 삭제된 경우 또는 LOAD 중 통계 옵션을 사용하지 않고 LOAD가 수행된 경우)
  - 재구성된 테이블(REORG, REDISTRIBUTE DATABASE PARTITION GROUP을 사용하여).
  - 행이 압축된 테이블
  - 새 인덱스가 작성된 경우
  - 성능이 중요한 응용프로그램을 바인드하기 전
  - 프리페치량이 변경된 경우
  - 기본 테이블이 실질적으로 수정되어 뷰에서 리턴된 행이 변경된 통계 뷰
  - STATISTICS 옵션을 사용하여 LOAD가 실행된 후 RUNSTATS 유틸리티를 사용하여 XML 컬럼에서 통계를 수집하십시오. LOAD가 STATISTICS 옵션을 사용하여 실행될 때라도 LOAD 중 XML 컬럼에 대한 통계는 결코 수집되지 않습니다. 단지 XML 컬럼에 대한 통계를 수집하기 위해 RUNSTATS를 사용할 때 LOAD 또는 이전에 RUNSTATS 유틸리티를 실행하여 수집한 비XML 컬럼에 대한 기존 통계는 보유됩니다. 일부 XML 컬럼에 대한 통계가 이전에 수집된 경우, 현재 명령으로 해당 XML 컬럼에 통계가 수집되지 않으면 XML 컬럼에 대해 이전에 수집된 통계는 삭제되며 현재 명령으로 해당 XML 컬럼에 대한 통계가 수집되면 이전에 수집된 통계를 대체합니다.
4. 선택된 옵션은 특정 테이블 및 응용프로그램에 따라 달라야 합니다. 일반적으로 다음과 같습니다.
  - 테이블이 중요한 쿼리에서 매우 중요한 테이블이고 비교적 작은 편이거나 너무 많이 변경되지 않으며 시스템 자체에서 활동이 너무 많지 않은 경우, 가능한 자세한 통계를 수집하는 데 노력을 기울일 만한 가치가 있을 수 있습니다.
  - 통계를 수집할 시간이 제한된 경우 테이블이 비교적 크거나 테이블이 자주 갱신되는 경우 술어에 사용된 컬럼 세트에 제한된 RUNSTATS를 실행하는 것이 유용할 수 있습니다. 이 같이 사용자는 RUNSTATS 명령을 좀 더 자주 실행할 수 있게 됩니다.
  - 통계 수집 시간이 매우 제한적이고 테이블 기반으로 테이블에 대해 RUNSTATS 명령을 조정하기 위한 노력이 주요 관건이 될 경우 "KEY" 컬럼에 대해서만 통계를 수집하는 것을 고려하십시오. 인덱스가 테이블에 중요한 컬럼 세트를 포함하고 있으며 필시 술어에 표시될 것으로 간주됩니다.
  - 통계 수집 시간이 매우 제한적이고 테이블 통계가 수집될 경우, TABLESAMPLE 옵션을 사용하여 테이블 데이터의 서브세트에 대한 통계를 수집하는 것을 고려하십시오.

- 테이블에 많은 인덱스가 있으며 인덱스에 대한 DETAILED(확장) 정보가 액세스 플랜을 개선할 수 있는 경우, SAMPLED 옵션을 사용하여 통계를 수집하는 데 걸리는 시간을 줄여볼 것을 고려하십시오.
  - "column = constant" 유형의 특정 컬럼 및 술어에 편차가 있는 경우, 해당 컬럼에 더 큰 NUM\_FREQVALUES 값을 지정하는 것이 유용할 수 있습니다.
  - 등호 술어에 사용되고 분산 값이 편차가 있을 수 있는 모든 컬럼에 대해 분산 통계를 수집하십시오.
  - 범위 술어(예: "column >= constant", "column BETWEEN constant1 AND constant2") 또는 "column LIKE '%xyz'" 유형을 갖는 컬럼의 경우, 더 큰 NUM\_QUANTILES 값을 지정하는 것이 유용할 수 있습니다.
  - 스토리지 스페이스가 문제가 있고 통계를 수집하는 데 너무 많은 시간을 제공할 수 없는 경우 술어에 사용되지 않는 컬럼에 대해 높은 NUM\_FREQVALUES 또는 NUM\_QUANTILES 값을 지정하지 마십시오.
  - 인덱스 통계가 요청되었으며 인덱스가 있는 테이블에서 통계가 실행된 적이 없는 경우, 테이블 및 인덱스에 대한 통계가 계산됩니다.
  - 테이블의 XML 컬럼에 대한 통계가 필요하지 않은 경우 EXCLUDING XML COLUMNS 옵션을 사용하여 모든 XML 컬럼을 제외시킬 수 있습니다. 이 옵션은 통계 컬렉션에 XML 컬럼을 지정하는 다른 모든 절보다 우선순위를 갖습니다.
5. 명령 실행 이후에 다음을 참고하십시오.
- COMMIT를 발행하여 잠금을 릴리스해야 합니다.
  - 새 액세스 플랜이 생성될 수 있게 하려면 목표 테이블을 참조하는 패키지를 리바인드해야 합니다.
  - 테이블의 부분에서 명령을 실행하면 명령이 마지막 발행된 후 테이블에 대한 활동의 결과로 불일치한 결과가 나타날 수 있습니다. 이 경우 경고 메시지가 리턴됩니다. 테이블에서만 RUNSTATS를 실행하면 테이블 및 인덱스 레벨 통계가 불일치할 수 있습니다. 예를 들어, 테이블에서 인덱스 레벨 통계를 수집한 후 테이블에서 상당한 수의 행을 삭제할 수 있습니다. 그런 다음 테이블에 대해서만 RUNSTATS를 발행하면 테이블 카디널리티가 FIRSTKEYCARD 보다 적을 수 있으며 불일치하게 됩니다. 마찬가지로 인덱스를 작성시 새 인덱스에서 통계를 수집하면 테이블 레벨 통계가 불일치할 수 있습니다.
6. 테이블 통계가 요청될 경우 RUNSTATS 명령은 이전에 수집된 분산 통계를 삭제합니다. 예를 들어, RUNSTATS ON TABLE 또는 RUNSTATS ON TABLE ... AND INDEXES ALL은 이전에 수집된 분산 통계가 삭제되게 합니다. 명령이 인덱스에 대해서만 실행될 경우 이전에 수집된 분산 통계는 보유됩니다. 예를 들어, RUNSTATS ON TABLE ... FOR INDEXES ALL은 이전에 수집된 분산 통계가 보유되게 합니다. RUNSTATS 명령이 XML 컬럼에 대해서만 실행될 경우 이전에 수집된 기본 컬럼 통계 및 분산 통계는 보유됩니다. 일부 XML 컬럼에

대한 통계가 이전에 수집된 경우, 현재 명령으로 해당 XML 컬럼에 통계가 수집되지 않으면 XML 컬럼에 대해 이전에 수집된 통계는 삭제되며 현재 명령으로 해당 XML 컬럼에 대한 통계가 수집되면 이전에 수집된 통계를 대체합니다.

7. 범위 클러스터 테이블의 경우, 이 테이블의 범위 순서화 등록 정보를 나타내는 특수 시스템 생성 인덱스가 카탈로그 테이블에 들어 있습니다. 이 유형의 테이블에서 통계가 수집될 때 테이블이 통계 컬렉션의 일부로 포함될 경우, 시스템 생성 인덱스에 대해서도 통계가 수집됩니다. 이 통계는 인덱스를 기본 데이터 테이블만큼 많은 페이지가 있는 두 레벨 인덱스로 나타내고 인덱스 순서에 따라 기본 데이터를 완벽하게 클러스터되게 하여 범위 찾아보기의 빠른 액세스를 반영합니다.
8. 명령 구문의 Dist Cols 절에서 빈도 옵션 및 Quantile 옵션 매개변수는 현재 컬럼 GROUPS에서 지원되지 않습니다. 이들 옵션은 단일 컬럼에만 지원됩니다.
9. DMS 모드에서 작업할 때 계산될 수 없는 세 가지 프리페치 통계가 있습니다. 인덱스 카탈로그에서 인덱스 통계를 보면 다음 통계의 경우 -1값이 표시됩니다.
  - AVERAGE\_SEQUENCE\_FETCH\_PAGES
  - AVERAGE\_SEQUENCE\_FETCH\_GAP
  - AVERAGE\_RANDOM\_FETCH\_PAGES
10. TABLESAMPLE을 통한 RUNSTATS 샘플링은 테이블 데이터 페이지에서만 발생하며 인덱스 페이지에서는 발생하지 않습니다. 샘플링과 함께 인덱스 통계가 요청되면 통계 수집을 위해 모든 인덱스 페이지가 스캔됩니다. TABLESAMPLE은 테이블 통계의 컬렉션에서만 적용됩니다. 그러나 SAMPLED DETAILED 옵션을 통해 보다 효율적인 자세한 인덱스 통계 컬렉션을 사용할 수 있습니다. 이는 TABLESAMPLE에서 사용된 것과 다른 샘플링 방법으로 자세한 인덱스 통계 세트에만 적용됩니다.
11. 프로파일 설정 또는 프로파일 갱신 옵션을 사용하여 RUNSTATS 명령에 지정된 테이블 또는 통계 뷰에 대한 통계 프로파일을 설정하거나 갱신할 수 있습니다. 통계 프로파일은 SYSIBM.SYSTABLES 시스템 카탈로그 테이블의 STATISTICS\_PROFILE 컬럼에서 RUNSTATS 명령을 나타내는 표시 가능한 문자열 형식으로 저장됩니다.
12. XML 유형 컬럼에 대한 통계 컬렉션은 다음과 같은 두 개의 DB2 데이터베이스 시스템 레지스트리 값으로 제어됩니다. DB2\_XML\_RUNSTATS\_PATHID\_K 및 DB2\_XML\_RUNSTATS\_PATHVALUE\_K. 이들 두 매개변수는 수집할 빈도 수 값을 지정한다는 점에서 NUM\_FREQVALUES 매개변수와 유사합니다. 이들 매개변수를 설정하지 않으면 디폴트값 200이 두 매개변수에 사용됩니다.
13. RUNSTATS는 SYSTABLES에 대해서는 IX 테이블 잠금을 획득하고 RUNSTATS가 시작될 때 stats가 수집되는 테이블의 행에 대해서는 U 잠금을 획득합니다. U 잠금이 설정된 행을 포함하는 SYSTABLES에서 여전히 읽기 조작을 수행할 수 있

습니다. U 잠금이 설정된 행에 대해 발생하지 않았으면 쓰기 조작도 가능합니다. 그러나 다른 판독기나 기록기는 RUNSTATS' IX 잠금 때문에 SYSTABLES에서 S 잠금을 획득할 수 없습니다.

14. 구조화된 유형이 있는 컬럼에 대한 통계는 수집되지 않습니다. 통계가 지정되면, 해당 데이터 유형이 있는 컬럼은 무시됩니다.
15. LOB 또는 LONG 데이터 유형이 있는 컬럼에 대한 AVGCOLLEN 및 NUMNULLS만 수집됩니다.
16. 컬럼이 데이터베이스 메모리 또는 임시 테이블에 저장될 때 AVGCOLLEN은 바이트 단위로 평균 스페이스를 표시합니다. LOB 데이터가 데이터 페이지에서 인라인될 때를 제외하고 이 값은 LOB 또는 LONG 데이터 유형의 데이터 디스크립터의 길이를 표시합니다.

주: 디스크에서 컬럼을 저장하기 위해 필요한 평균 스페이스는 이 통계로 표시되는 값과 다를 수 있습니다.

## ADMIN\_CMD 프로시저를 사용하는 SET TAPE POSITION 명령

백업 및 리스토어 조작을 위한 테이프의 위치를 스트리밍 테이프 디바이스로 설정하십시오. 이 명령은 Windows 운영 체제에서만 지원됩니다.

### 권한 부여

다음 중 하나가 필요합니다.

- *sysadm*
- *sysctrl*
- *sysmaint*

### 필수 연결

데이터베이스

### 명령 구문

```

▶▶ SET TAPE POSITION ON device TO position ▶▶

```

### 명령 매개변수

**ON device**

유효한 테이프 디바이스 이름을 지정합니다. 디폴트값은 `WWW.WTAPE0`입니다. 지정된 디바이스는 서버에 관계되어야 합니다.

**TO position**

테이프가 위치 지정될 표시를 지정합니다. Windows용 DB2는 모든 백업 이미



지 다음에 테이프 표시를 씁니다. 값 1은 첫 번째 위치를 지정하고 2는 두 번째 위치를 지정하며 계속 이와 같이 지정합니다. 테이프가 테이프 표시 1에 위치 지정된 경우 예를 들어 아카이브 2가 리스토어되도록 위치 지정됩니다.

예 :

DB2가 모든 백업 이미지 다음에 테이프 표시를 쓰므로, 위치 1을 지정하면 테이프의 두 번째 아카이브가 시작되는 위치로 테이프를 이동시킵니다.

```
CALL SYSPROC.ADMIN_CMD( 'set tape position to 1' )
```

## 사용 시 참고사항

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

## ADMIN\_CMD 프로시저를 사용하는 UNQUIESCE DATABASE 명령

유지보수 또는 기타 이유로 Quiesce 상태에 있는 데이터베이스로 사용자 액세스를 리스토어합니다. UNQUIESCE는 종료 및 데이터베이스 재시작을 필요로 하지 않고 사용자 액세스를 리스토어합니다.

특정하게 지정된 경우를 제외하고는 *sysadm*, *sysmaint* 또는 *sysctrl* 권한이 있는 사용자만 Quiesce 상태에 있는 데이터베이스에 액세스할 수 있습니다. 따라서 UNQUIESCE된 데이터베이스로 일반 액세스를 리스토어하려면 UNQUIESCE가 필요합니다.

## 범위

UNQUIESCE DB는 Quiesce 상태의 데이터베이스에 있는 모든 오브젝트로 사용자 액세스를 리스토어합니다.

인스턴스를 중지하고 이 인스턴스 및 해당되는 모든 데이터베이스를 Unquiesce하려면 `db2stop` 명령을 발행하십시오. DB2를 중지한 후 재시작하면 모든 인스턴스 및 데이터베이스가 Unquiesce됩니다.

## 권한 부여

다음 중 하나가 필요합니다.

데이터베이스 레벨 Unquiesce를 수행하려는 경우:

- *sysadm*
- *dbadm*

## 명령 구문

▶—UNQUIESCE—DB—▶

### 필수 연결

데이터베이스

### 명령 매개변수

**DB** 데이터베이스를 Unquiesce하십시오. 데이터베이스에 있는 모든 오브젝트로 사용자 액세스가 리스토어됩니다.

예:

데이터베이스 **Unquiesce**

```
CALL SYSPROC.ADMIN_CMD( 'unquiesce db' )
```

이 명령은 이전에 Quiesce되었던 데이터베이스를 Unquiesce합니다.

### 사용 시 참고사항

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

## ADMIN\_CMD 프로시저를 사용하는 UPDATE ALERT CONFIGURATION

### 명령

Health 표시기의 경보 구성 설정값을 갱신합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 명령 또는 API는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 권한 부여

다음 중 하나가 필요합니다.

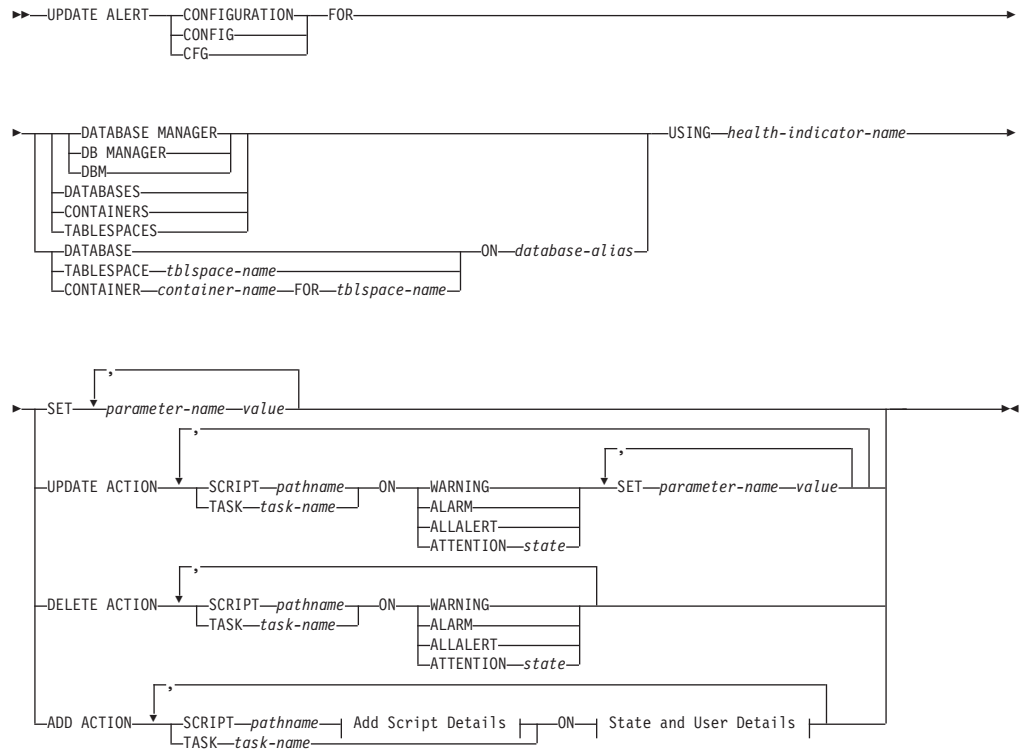
- *sysadm*
- *sysmaint*
- *sysctrl*

### 필요한 연결

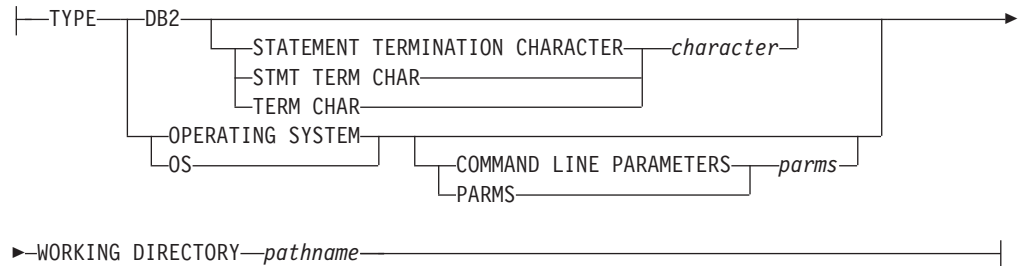
데이터베이스



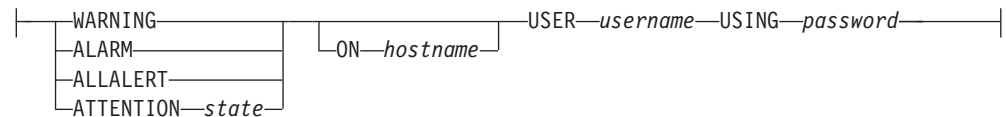
## 명령 구문



### Add Script Details:



### State and User Details:



## 명령 매개변수

### DATABASE MANAGER

데이터베이스 관리 프로그램에 대한 정보 설정값을 갱신합니다.

### DATABASES

데이터베이스 관리 프로그램이 관리하는 모든 데이터베이스에 대한 정보 설정

값을 갱신합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 데이터베이스에 적용됩니다. 사용자 정의 설정값은 `DATABASE ON database-alias` 절을 사용하여 정의됩니다.

## CONTAINERS

데이터베이스 관리 프로그램이 관리하는 모든 테이블 스페이스 컨테이너에 대한 경보 설정값을 갱신합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 테이블 스페이스 컨테이너에 적용됩니다. 사용자 정의 설정은 `CONTAINER container-name ON database-alias` 절을 사용하여 정의됩니다.

## TABLESPACES

데이터베이스 관리 프로그램이 관리하는 모든 테이블 스페이스에 대한 경보 설정값을 갱신합니다. 이들 설정값은 사용자 정의 설정값이 없는 모든 테이블 스페이스에 적용됩니다. 사용자 정의 설정은 `TABLESPACE tblspace-name ON database-alias` 절을 사용하여 정의됩니다.

## DATABASE ON database-alias

`ON database-alias` 절을 사용하여 지정된 데이터베이스의 경보 설정을 갱신합니다. 이 데이터베이스가 사용자 정의 설정을 가지고 있을 경우, 이 설정은 `DATABASES` 매개변수를 사용하여 지정된 해당 인스턴스의 모든 데이터베이스에 대한 설정보다 우선합니다.

## CONTAINER container-name FOR tblspace-name ON database-alias

`ON database-alias` 절을 사용하여 지정된 데이터베이스에서 `FOR tblspace-name` 절을 사용하여 지정된 테이블 스페이스에 대해 `container-name`이라고 하는 테이블 스페이스 컨테이너의 경보 설정을 갱신합니다. 이 테이블 스페이스 컨테이너에 사용자 정의 설정이 있는 경우, 이 설정은 `CONTAINERS` 매개변수를 사용하여 지정된 데이터베이스의 모든 테이블 스페이스 컨테이너에 대한 설정보다 우선합니다.

## TABLESPACE tblspace-name ON database-alias

`ON database-alias` 절을 사용하여 지정된 데이터베이스에서 `name`이라고 하는 테이블 스페이스의 경보 설정을 갱신합니다. 이 테이블 스페이스에 사용자 정의 설정이 있는 경우, 이 설정은 `TABLESPACES` 매개변수를 사용하여 지정된 데이터베이스의 모든 테이블 스페이스에 대한 설정보다 우선합니다.

## USING health-indicator-name

경보 구성이 갱신될 Health 표시기 세트를 지정합니다. Health 표시기 이름은 두 문자 오브젝트 ID와 그 뒤에 표시기가 측정하는 것을 설명하는 이름으로 구성됩니다. 예를 들면, 다음과 같습니다.

```
db.sort_privmem_util
```

### **SET** *parameter-name value*

지정된 값에 대한 Health 표시기의 경보 구성 요소(*parameter-name*)를 갱신합니다. *parameter-name*은 다음 중 하나여야 합니다.

- ALARM: *value*는 Health 표시기 단위입니다.
- WARNING: *value*는 Health 표시기 단위입니다.
- SENSITIVITY: *value*는 초 단위입니다.
- ACTIONSENABLED: *value*는 YES 또는 NO가 될 수 있습니다.
- THRESHOLDSCHECKED: *value*는 YES 또는 NO가 될 수 있습니다.

특정 DB2 버전에 대해 가능한 Health 표시기 단위 목록은 다음 쿼리를 실행하여 수집할 수 있습니다.

```
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*]

절대 경로 이름이 *pathname*인 사전 정의된 스크립트의 스크립트 속성이 다음 절에 따라 갱신됨을 지정합니다.

### **SET** *parameter-name value*

스크립트 속성 *parameter-name*을 지정된 값으로 갱신합니다. *parameter-name*은 다음 중 하나여야 합니다.

- SCRIPTTYPE  
OS 또는 DB2가 유효한 유형입니다.
- WORKINGDIR
- TERMCHAR
- CMDLINEPARMS

운영 체제 스크립트에 대해 사용자가 지정하는 명령 매개변수는 디폴트 제공 매개변수 앞에 놓입니다. 운영 체제 스크립트로 보내는 매개변수는 다음과 같습니다.

- 사용자가 제공한 매개변수 목록
- Health 표시기 단축 이름
- 완전한 오브젝트 이름
- Health 표시기 값
- 경보 상태

- USERID
- PASSWORD
- SYSTEM

**UPDATE ACTION TASK** *task-name* ON [WARNING | ALARM | ALLALERT | ATTENTION *state*]

이름이 *name*인 태스크의 속성이 다음 절에 따라 갱신됨을 지정합니다.

**SET** *parameter-name value*

태스크 속성 *parameter-name*을 지정된 값으로 갱신합니다. *parameter-name*은 다음 중 하나여야 합니다.

- USERID
- PASSWORD
- SYSTEM

**DELETE ACTION SCRIPT** *pathname* ON [WARNING | ALARM | ALLALERT | ATTENTION *state*]

경보 조치 스크립트 목록에서 절대 경로 이름이 *pathname*인 조치 스크립트를 제거합니다.

**DELETE ACTION TASK** *task-name* ON [WARNING | ALARM | ALLALERT | ATTENTION *state*]

경보 조치 태스크 목록에서 이름이 *name*인 조치 태스크를 제거합니다.

**ADD ACTION SCRIPT** *pathname* ON [WARNING | ALARM | ALLALERT | ATTENTION *state*]

절대 경로 이름이 *pathname*인 새 조치 스크립트가 추가됨을 지정합니다. 속성은 다음과 같이 제공됩니다.

**TYPE** 조치 스크립트는 DB2 명령 스크립트나 운영 체제 스크립트여야 합니다.

- DB2
- OPERATING SYSTEM

DB2 명령 스크립트인 경우, 다음 절을 사용하여 명령문을 종료하기 위한 스크립트에서 사용되는 문자 *character*를 선택적으로 지정할 수 있습니다.

STATEMENT TERMINATION CHARACTER ;

운영 체제 스크립트인 경우, COMMAND LINE PARAMETERS *parms* 절을 사용하여, 호출 시 스크립트에 전달되는 명령 매개변수 *parms*를 선택적으로 지정할 수 있습니다.

**WORKING DIRECTORY** *pathname*

스크립트가 실행될 디렉토리의 절대 경로 이름 *pathname*을 지정합니다.

**USER** *username* USING *password*

스크립트가 실행될 사용자 어카운트 *username* 및 연관된 암호 *password*를 지정합니다. ADD ACTION 옵션을 사용할 때 *username*

및 *password*는 네트워크(*username* 및 *password*가 암호화되지 않은 상태로 전송된), db2diag 로그 파일, 추적 파일, 덤프 파일, 스냅샷 모니터(동적 SQL 스냅샷), 시스템 모니터 스냅샷, 여러 개의 이벤트 모니터(예: 명령문, 교착 상태), Query Patroller, Explain 테이블, db2pd 출력(예: 패키지 캐시 및 잠금 시간종료 메커니즘) 및 db2 감사 레코드에서 노출될 수 있습니다.

**ADD ACTION TASK *name* ON [WARNING | ALARM | ALLALERT | ATTENTION *state*]**

*name*이라고 하는 새 태스크가 지정된 조건에서 실행되도록 추가됨을 지정합니다.

**ON [WARNING | ALARM | ALLALERT | ATTENTION *state*]**

조치 또는 태스크가 실행될 조건을 지정합니다. 임계값 기반 Health 표시기(HI)의 경우, 이는 WARNING 또는 ALARM입니다. 상태 기반 HI의 경우, 이는 각 상태 기반 HI에 대해 문서화된 대로 숫자 상태가 됩니다(예를 들어, ts.ts\_op\_status Health 표시기의 경우 테이블 스페이스 상태에 대해 tablespace\_state 모니터 요소를 참조).

**ATTENTION *state***

일부 데이터베이스 Health 표시기 상태에 대한 유효한 숫자 값이 ADD ACTION SCRIPT CLP 명령 옵션의 예로 아래에 제공되어 있습니다.

- 0 - 활성. 일반(ACTIVE)
- 1 - Quiesce 보류(QUIESCE\_PEND)
- 2 - Quiesce 상태(QUIESCED)
- 3 - 롤 포워드(ROLLFWD)

추가적인 Health 표시기가 헤더 파일 sqlmon.h 및 sqlutil.h에 정의되어 있습니다.

ADMIN\_CMD 스토어드 프로시저가 호출한 UPDATE ALERT CFG 명령은 *state*에 대한 숫자 값이나 텍스트 ID를 지원합니다. 추가 Health 표시기 상태에 대한 유효한 숫자 값 및 텍스트 ID(테이블 스페이스 작동 상태 Health 표시기(ts.ts\_op\_status)에 대한 예로)는 다음과 같습니다.

- 0x1 - QUIESCED\_SHARE
- 0x2 - QUIESCED\_UPDATE
- 0x4 - QUIESCED\_EXCLUSIVE

UPDATE ALERT CFG 명령과 위의 Health 표시기 값을 사용하는 경우, 다음 명령행 항목은

```
ADD ACTION SCRIPT ... ON ATTENTION 2
```

다음과 같습니다.

```
ADD ACTION SCRIPT ... ON ATTENTION QUIESCED_UPDATE
```

또한 테이블 스페이스 작동 상태 Health 표시기(ts.ts\_op\_status)의 경우 상태를 함께 OR 처리하여 단일 숫자 값으로 여러 상태를 지정할 수 있습니다. 예를 들어, 상태 7(= 0x1 + 0x2 + 0x4)을 지정할 수 있습니다. 조치는 테이블 스페이스가 Quiesced: SHARE, Quiesced: UPDATE 또는 Quiesce: EXCLUSIVE 상태 중 하나가 될 때 수행됩니다. 또한 세 개의 독립 UPDATE ALERT CFG 명령 실행에서 QUIESCED\_SHARE, QUIESCED\_UPDATE 및 QUIESCED\_EXCLUSIVE를 지정할 수도 있습니다.

예 :

호스트 이름이 'plato'인 시스템에 정보가 있는 경우 /home/test/scripts/logfsutilact 스크립트를 실행할 조치를 db.log\_fs\_util 표시기에 대해 추가하십시오.

```
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 mypassw dv' )
```

정보 구성이 설정된 후 이를 확인하려면, 다음과 같이

HEALTH\_GET\_IND\_DEFINITION 및 HEALTH\_GET\_ALERT\_ACTION\_CFG 테이블 함수를 사용하십시오.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

OBJECTTYPE	ID	CONDITION	ACTIONTYPE	ACTION_NAME
DB	1006	ALARM	S	/home/dricard/scripts/logfsutilact

1개의 레코드가 선택되었습니다.

## 사용 시 참고사항

ADD ACTION 옵션의 경우, 제공된 *username* 및 *password*는 SQL문 텍스트가 캡처되는 다양한 위치에서 노출될 수 있습니다.

- 네트워크(사용자 이름/암호가 암호화되지 않은 연결을 통해 전달됨)
- db2diag 로그 파일
- 추적 파일
- 덤프 파일

- 스냅샷 모니터(동적 SQL 스냅샷)
- 시스템 모니터 스냅샷
- 여러 이벤트 모니터(명령문, 교착 상태)
- Query Patroller
- Explain 테이블
- db2pd 출력(그 중에서도 특히, 패키지 캐시 및 잠금 시간종료 메커니즘)
- DB2 감사 레코드

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

*database-alias*는 서버의 카탈로그에서 정의해야 하며 서버에 로컬 상태여야 합니다.

*pathname*은 완전한 서버 경로 이름이어야 합니다.

## ADMIN\_CMD 프로시저를 사용하는 UPDATE CONTACT 명령

로컬 시스템에 정의되어 있는 문의처의 속성을 갱신합니다. 문의처는 스케줄러 및 Health Monitor가 메시지를 보내는 사용자입니다. 문의처를 작성하려면 ADD CONTACT 명령을 사용하십시오. DAS(Database Administration Server) **contact\_host** 구성 매개 변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

### 권한 부여

없음

### 필수 연결

데이터베이스. DAS가 실행하고 있어야 합니다.

### 명령 구문

```

▶▶ UPDATE CONTACT name USING keyword value

```

### 명령 매개변수

**UPDATE CONTACT** *name*

갱신될 문의처 이름.

**USING** *keyword value*

갱신될 문의처 매개변수(*keyword*) 및 이 매개변수가 설정될 값(*value*)을 지정합니다. 유효한 키워드 세트는 다음과 같습니다.

### ADDRESS

SMTP 서버가 통지를 보내기 위해 사용하는 전자 우편 주소.

**TYPE** 주소가 전자 우편 주소인지 또는 호출기인지를 나타냅니다.

### MAXPAGELEN

호출기가 승인할 수 있는 최대 문자 수.

### DESCRIPTION

문의처의 텍스트 설명. 최대 128 문자를 사용합니다.

예 :

사용자 'test'의 주소를 'newaddress@test.com'으로 갱신하십시오.

```
CALL SYSPROC.ADMIN_CMD( 'update contact test using address newaddress@test.com' )
```

### 사용 시 참고사항

DAS를 작성하여 실행해야 합니다.

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

## ADMIN\_CMD 프로시저를 사용하는 UPDATE CONTACTGROUP 명령

로컬 시스템에 정의되어 있는 문의처 그룹의 속성을 갱신합니다. 문의처 그룹은 스케줄러 및 Health Monitor가 통지해야 하는 사용자 목록입니다. DAS(Database Administration Server) **contact\_host** 구성 매개변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

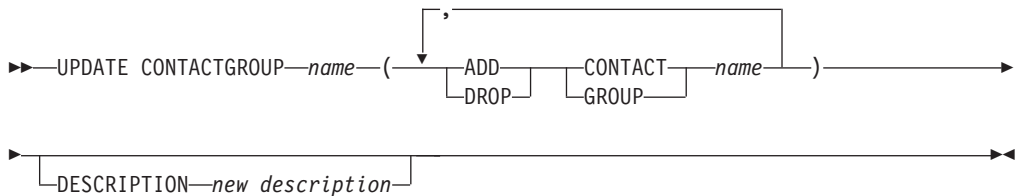
### 권한 부여

없음

### 필요한 연결

데이터베이스. DAS가 실행하고 있어야 합니다.

### 명령 구문





## 명령 매개변수

### **CONTACTGROUP** *name*

갱신될 문의처 그룹 이름

### **ADD CONTACT** *name*

그룹에 추가될 새 문의처 이름을 지정합니다. 문의처를 그룹에 추가한 후 ADD CONTACT 명령을 사용하여 정의할 수 있습니다.

### **DROP CONTACT** *name*

그룹에서 삭제될 그룹의 문의처 이름을 지정합니다.

### **ADD GROUP** *name*

그룹에 추가될 새 문의처 그룹의 이름을 지정합니다.

### **DROP GROUP** *name*

그룹에서 삭제될 문의처 그룹 이름을 지정합니다.

### **DESCRIPTION** 새 설명

선택사항. 문의처 그룹의 새 텍스트 설명.

## 예 :

문의처 그룹 'gname1'에 문의처 'cname2'를 추가하십시오.

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

## 사용 시 참고사항

DAS를 작성하여 실행해야 합니다.

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

## **ADMIN\_CMD** 프로시저를 사용하는 **UPDATE DATABASE CONFIGURATION** 명령

특정 데이터베이스 구성 파일에 있는 개별 항목을 수정합니다.

데이터베이스 구성 파일은 해당 데이터베이스가 작성된 모든 데이터베이스 파티션에 있습니다.

## 범위

DBPARTITIONNUM이 하나의 데이터베이스 파티션만을 갱신하도록 지정된 경우를 제외하고, 이 명령은 디폴트로 모든 데이터베이스 파티션을 갱신합니다.



## DEFERRED

구성 파일에서만 변경사항을 작성하여 다음 번에 데이터베이스를 다시 활성화할 때 이 변경사항이 적용되도록 하십시오.

## FOR *database-alias*

구성을 갱신할 데이터베이스의 별명을 지정합니다. 데이터베이스 연결이 이미 설정되었을 때는 데이터베이스 별명을 지정할 필요가 없습니다. 데이터베이스 별명은 서버에 로컬로 정의되어 있어야 합니다. 동일한 데이터베이스 인스턴스를 사용하는 다른 데이터베이스의 구성 파일을 갱신할 수 있습니다. 예를 들어, db11 데이터베이스에만 연결되어 있는 경우 `update db config for alias db22 using .... immediate:`

- db22에 사용 중인 연결이 없는 경우, 구성 파일만 갱신하면 되기 때문에 갱신이 완료됩니다. 새 연결(데이터베이스를 활성화함)은 메모리 내의 새로운 변경사항을 표시합니다.
- db22에 다른 응용프로그램에서 사용 중인 연결이 있으면 갱신이 디스크에서는 작동하나 메모리에서는 작동하지 않습니다. 데이터베이스를 재시작하라는 경고가 수신됩니다.

## DBPARTITIONNUM *db-partition-num*

데이터베이스 구성 갱신이 특정 데이터베이스 파티션에 적용되는 경우, 이 매개변수를 사용할 수 있습니다. 이 매개변수가 제공되지 않으면, 모든 데이터베이스 파티션에서 갱신이 유효합니다.

## IMMEDIATE

데이터베이스가 실행되는 동안 변경사항을 즉시 작성합니다. IMMEDIATE는 디폴트 조치입니다. ADMIN\_CMD 프로시저를 수행하려면 데이터베이스 연결이 필요하므로 변경사항은 연결된 데이터베이스에 대한 동적으로 구성 가능한 매개변수에 즉시 적용됩니다.

이것은 CLPPlus 인터페이스에서 조작 시 디폴트 절입니다. CLPPlus 프로세서 사용 시 IMMEDIATE를 호출할 필요가 없습니다.

## MANUAL

구성 매개변수의 자동 성능 조절을 사용하지 않습니다. 매개변수는 현재 내부 값으로 설정되며 더 이상 자동으로 갱신되지 않습니다.

## USING *config-keyword value*

*config-keyword*는 갱신될 데이터베이스 구성 매개변수를 지정합니다. *value*는 매개변수에 지정될 값을 지정합니다.

예 :

응용프로그램이 현재 연결되어 있는 데이터베이스 파티션에서 데이터베이스 구성 매개변수 `sorthheap`을 1000값으로 설정하십시오.

```
CALL SYSPROC.ADMIN_CMD ('UPDATE DB CFG USING sortheap 1000')
```

## 사용 시 참고사항

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

*database-alias*는 서버에 정의되어 있는 별명 이름이어야 합니다.

DBPARTITIONNUM이 지정되지 않는 한, 해당 명령은 모든 데이터베이스 파티션에 영향을 미칩니다.

데이터베이스 구성 매개변수 목록을 보거나 인쇄하려면 SYSIBMADM.DBCFG 관리 뷰를 사용하십시오.

모든 데이터베이스 구성 매개변수를 권장 디폴트값으로 재설정하려면 ADMIN\_CMD 프로시저를 사용하여 RESET DATABASE CONFIGURATION 명령을 사용하십시오.

데이터베이스 구성 매개변수를 변경하려면 ADMIN\_CMD 프로시저를 통해 UPDATE DATABASE CONFIGURATION 명령을 사용하십시오. 예를 들어, ZELLMART라는 데이터베이스가 들어 있는 단일 파티션 데이터베이스 환경에서 로깅 모드를 『아카이브 로깅』으로 변경하려면 다음 명령을 사용하십시오.

```
CALL SYSPROC.ADMIN_CMD ('update db cfg for zellmart using logretain recovery')
```

**logretain** 구성 매개변수가 변경되었는지 확인하려면 다음 명령을 사용하십시오.

```
SELECT * FROM SYSIBMADM.DBCFG WHERE NAME='logretain'
```

특정 데이터베이스 파티션에서 데이터베이스 구성 매개변수를 갱신하려면, 다음을 수행하십시오.

1. DB2NODE 변수를 데이터베이스 파티션 번호로 설정합니다.
2. 데이터베이스 파티션에 연결합니다.
3. ADMIN\_CMD 프로시저를 통해 UPDATE DATABASE CONFIGURATION 명령을 사용하여 데이터베이스 구성 매개변수를 갱신합니다.
4. 데이터베이스 파티션에서 연결을 끊습니다.

또는 DBPARTITIONNUM을 사용할 수 있습니다. 예를 들어, DBPARTITIONNUM을 사용하여 하나의 특정 파티션(30)에만 로깅 모드로 갱신합니다.

```
CALL SYSPROC.ADMIN_CMD ('update db cfg for zellmart dbpartitionnum 30 using logretain recovery')
```

DB2 구성 매개변수 및 각 데이터베이스 노드 유형에 사용할 수 있는 값에 대한 자세한 정보는 개별 구성 매개변수 설명을 참조하십시오. 이들 매개변수의 값은 구성된 각 데이터베이스 노드 유형(서버, 클라이언트 또는 리모트 클라이언트가 있는 서버)에 따라 다릅니다.

일부 매개변수는 갱신할 수 있습니다.

데이터베이스 구성 파일에 대한 일부 변경사항은 메모리로 로드된 후에만 유효하게 됩니다. 이렇게 되려면 데이터베이스에서 모든 응용프로그램의 연결을 끊어야 합니다. 어떤 매개변수가 온라인으로 구성 가능하고 어떤 매개변수가 그렇지 않은지에 대한 자세한 정보는 구성 매개변수의 요약 목록을 참조하십시오.

오류가 발생하면 데이터베이스 구성 파일이 변경되지 않습니다. 체크섬이 유효하지 않으면 데이터베이스 구성 파일을 갱신할 수 없습니다. 적절한 명령을 사용하지 않고 데이터베이스 구성 파일을 변경할 경우 이러한 일이 발생할 수 있습니다. 이러한 경우 데이터베이스를 리스토어하여 데이터베이스 구성 파일을 재설정해야 합니다.

## ADMIN\_CMD 프로시저를 사용하는 UPDATE DATABASE MANAGER CONFIGURATION 명령

현재 연결된 데이터베이스가 있는 인스턴스의 데이터베이스 관리 프로그램 구성 파일에 있는 개별적 항목을 수정합니다.

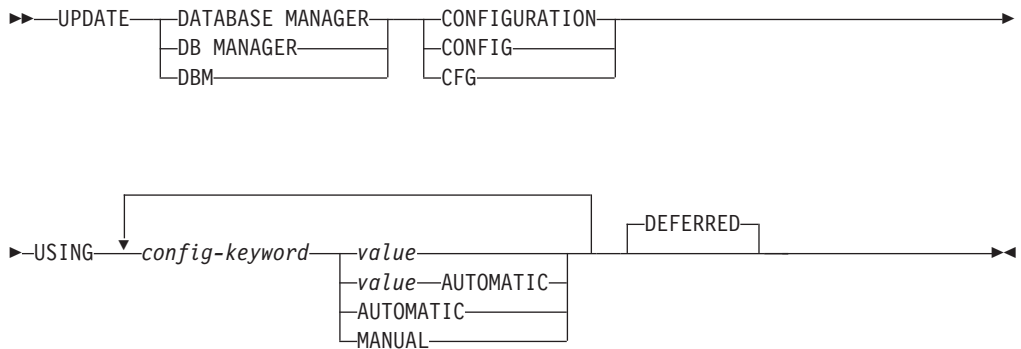
### 권한 부여

*sysadm*

### 필수 연결

데이터베이스

### 명령 구문



### 명령 매개변수

#### AUTOMATIC

일부 구성 매개변수는 AUTOMATIC으로 설정되어 DB2가 자동으로 이 매개변수를 조정하여 현재 자원 요구사항을 반영할 수 있습니다. AUTOMATIC 키워드를 지원하는 구성 매개변수 목록에 대해서는 구성 매개변수 요약을 참조하십시오.

십시오. AUTOMATIC 키워드를 따라 값이 지정된 경우, 자동 계산에 영향을 미칠 수 있습니다. 이 작동에 대한 특정 정보는 구성 매개변수의 문서를 참조하십시오.

주: **federated\_async** 데이터베이스 관리 프로그램 구성 매개변수는 명령행 처리기를 사용하여 AUTOMATIC으로만 설정될 수 있음을 참고하십시오.

## DEFERRED

구성 파일에서만 변경사항을 작성하여 인스턴스가 재시작될 때 이 변경사항이 적용되도록 하십시오. 이것은 디폴트값입니다.

이것은 CLPPlus 인터페이스에서 조작 시 디폴트 절입니다. CLPPlus 프로세서 사용 시 DEFERRED를 호출할 필요가 없습니다.

## MANUAL

구성 매개변수의 자동 성능 조절을 사용하지 않습니다. 매개변수는 현재 내부 값으로 설정되며 더 이상 자동으로 갱신되지 않습니다.

## USING *config-keyword value*

갱신될 데이터베이스 관리 프로그램 구성 매개변수를 지정합니다. 구성 매개변수 목록에 대해서는 구성 매개변수 요약을 참조하십시오. *value*는 매개변수에 지정될 값을 지정합니다.

## 예 :

데이터베이스 관리 프로그램 구성에 대해 진단 레벨을 1로 갱신하십시오.

```
CALL SYSPROC.ADMIN_CMD('db2 update dbm cfg using DIAGLEVEL 1')
```

## 사용 시 참고사항

데이터베이스 관리 프로그램 구성 매개변수 목록을 보거나 인쇄하려면 SYSIBMADM.DBMCFG 관리 뷰를 사용하십시오. 데이터베이스 관리 프로그램 구성 매개변수를 권장 데이터베이스 관리 프로그램 디폴트값으로 재설정하려면 ADMIN\_CMD 프로시저를 통해 RESET DATABASE MANAGER CONFIGURATION 명령을 사용하십시오. 구성된 각 데이터베이스 노드 유형에 적절한 데이터베이스 관리 프로그램 구성 매개변수 및 이들 매개변수의 값에 대한 자세한 정보는 개별 구성 매개변수 설명을 참조하십시오.

일부 매개변수는 갱신할 수 있습니다.

데이터베이스 관리 프로그램 구성 파일에 대한 일부 변경사항은 메모리로 로드된 후에만 적용됩니다. 어떤 매개변수가 온라인으로 구성 가능하고 어떤 매개변수가 그렇지 않은지에 대한 자세한 정보는 구성 매개변수 요약을 참조하십시오. 즉시 재설정되지 않는 서버 구성 매개변수는 db2start 실행 중 재설정됩니다. 클라이언트 구성 매개변수의 경

우, 매개변수는 다음 번에 응용프로그램을 재시작할 때 재설정됩니다. 클라이언트가 명령행 처리기일 경우 TERMINATE를 호출해야 합니다.

오류가 발생하면 데이터베이스 관리 프로그램 구성 파일이 변경되지 않습니다.

체크섬이 유효하지 않으면 데이터베이스 관리 프로그램 구성 파일을 갱신할 수 없습니다. 데이터베이스 관리 프로그램 구성 파일을 편집하며 적절한 명령을 사용하지 않을 경우 이러한 일이 발생할 수 있습니다. 체크섬이 유효하지 않으면 데이터베이스 관리 프로그램을 재설치하여 데이터베이스 관리 프로그램 구성 파일을 재설정해야 합니다.

현재 인스턴스에 대한 SVCENAME 또는 TPNAME 데이터베이스 관리 프로그램 구성 매개변수를 갱신할 때 LDAP 지원이 사용 가능하거나 이 인스턴스에 대해 LDAP 서버가 등록되어 있으면 LDAP 서버가 이 새 값으로 갱신됩니다.

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

갱신사항은 연결된 데이터베이스가 있는 데이터베이스 인스턴스에만 작성될 수 있습니다.

매개변수가 동적 갱신을 지원할 경우 IMMEDIATE 키워드가 지정되어 있지 않더라도 이 매개변수를 동적으로 갱신하려는 시도가 작성됩니다. 사용된 권한은 SYSTEM\_USER ID입니다.

## ADMIN\_CMD 프로시저를 사용하는 UPDATE HEALTH NOTIFICATION CONTACT LIST 명령

인스턴스에 의해 발행된 Health 경보에 대한 통지의 문의처 목록을 갱신합니다.

### 권한 부여

다음 중 하나가 필요합니다.

- sysadm
- sysctrl
- sysmaint

### 필요한 연결

데이터베이스

### 명령 구문

►► UPDATE HEALTH NOTIFICATION CONTACT LIST  
                  └── NOTIFICATION ───┘



## 명령 매개변수

### **ADD GROUP** *name*

인스턴스의 Health를 통지하는 새 문의처 그룹을 추가합니다.

### **ADD CONTACT** *name*

인스턴스의 Health를 통지할 새 문의처를 추가합니다.

### **DROP GROUP** *name*

인스턴스의 Health를 통지할 문의처 목록에서 문의처 그룹을 제거합니다.

### **DROP CONTACT** *name*

인스턴스의 Health를 통지할 문의처 목록에서 문의처를 제거합니다.

## 예 :

Health 통지 문의처 목록에 문의처 그룹 'gname1'을 추가하십시오.

```
CALL SYSPROC.ADMIN_CMD( 'update notification list add group gname1' )
```

## 사용법 참고

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

## ADMIN\_CMD 프로시저를 사용하는 UPDATE HISTORY 명령

현재 연결되어 있는 데이터베이스 파티션의 실행기록 파일 항목에서 위치, 디바이스 유형, 주석 또는 상태를 갱신합니다.

## 권한 부여

다음 중 하나가 필요합니다.

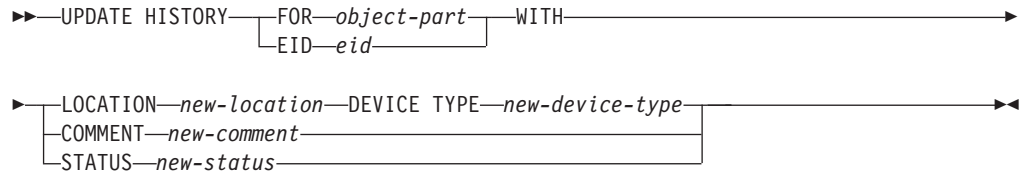
- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

## 필수 연결

데이터베이스



## 명령 구문



## 명령 매개변수

### FOR *object-part*

갱신될 실행기록 항목의 ID를 지정합니다. 001부터 999까지의 선택적 시퀀스 번호가 있는 시간소인입니다. 항목 상태를 갱신하는 데는 이 매개변수를 사용할 수 없습니다. 항목 상태를 갱신하려면 대신 EID를 지정하십시오.

### EID *eid*

실행기록 항목 ID를 지정합니다.

### LOCATION *new-location*

백업 이미지의 새 실제 위치를 지정합니다. 이 매개변수의 해석은 디바이스 유형에 따라 다릅니다.

### DEVICE TYPE *new-device-type*

백업 이미지를 저장할 새 디바이스 유형을 지정합니다. 유효한 디바이스 유형은 다음과 같습니다.

- D** 디스크
- K** 디스켓
- T** 테이프
- A** Tivoli Storage Manager
- F** 스냅샷 백업
- U** User exit
- P** 파이프
- N** 널(NULL) 디바이스
- X** XBSA
- Q** SQL문
- O** 기타

### COMMENT *new-comment*

항목을 설명할 새 주석을 지정합니다.

## STATUS *new-status*

항목에 대한 새 상태를 지정합니다. 백업 항목만 상태를 갱신할 수 있습니다. 가능한 값은 다음과 같습니다.

- A**    사용 중. 백업 이미지가 사용 중인 로그 체인에 있습니다. 대부분의 항목이 사용 중입니다.
- I**    비활동. 현재 로그 시퀀스(현재 로그 체인이라고도 함)에 해당하지 않는 백업 이미지는 비활성으로 플래그됩니다.
- E**    만기됨. NUM\_DB\_BACKUPS를 초과한 활성 이미지가 있으므로 더 이상 필요하지 않은 백업 이미지는 만기된 것으로 플래그됩니다.
- D**    삭제됨. 더 이상 복구에 사용할 수 없는 백업 이미지는 삭제된 것으로 표시되어야 합니다.
- X**    삭제되지 않음. DB2HISTORY\_STATUS\_DO\_NOT\_DELETE로 표시된 복구 실행기록 파일 항목은 PRUNE HISTORY 명령을 호출하여 프른되지 않으며, PRUNE HISTORY로 ADMIN\_CMD 프로시저를 실행하며, db2Prune API나 자동화된 복구 실행기록 파일 프른을 호출합니다. DB2HISTORY\_STATUS\_DO\_NOT\_DELETE 상태를 사용하여 키 복구 파일 항목이 프른되지 않게 하고 연관된 복구 오브젝트가 삭제되지 않게 할 수 있습니다. 로그 파일, 백업 이미지 및 로드 복사 이미지만 DB2HISTORY\_STATUS\_DO\_NOT\_DELETE로 표시될 수 있습니다.

## 예 :

1997년 4월 13일 오전 10시에 수행된 전체 데이터베이스 백업에 대한 실행기록 파일을 갱신하려면 다음을 입력하십시오.

```
CALL SYSPROC.ADMIN_CMD('update history
for 19970413100000001 with location
/backup/dbbackup.1 device type d')
```

## 사용 시 참고사항

데이터베이스 실행기록 파일의 1차 목적은 정보를 기록하는 것이지만 실행기록에 있는 데이터는 자동 리스토어 조작에서 직접 사용됩니다. AUTOMATIC 옵션을 지정하여 리스토어하는 동안 리스토어 유틸리티는 백업 이미지의 실행기록과 해당 위치를 참조 및 사용하여 자동 리스토어 요청을 이행합니다. 자동 리스토어 기능을 사용하며 백업 이미지가 작성된 이후 그 위치가 변경된 경우, 현재 위치를 반영하도록 해당 이미지에 대한 데이터베이스 실행기록 레코드를 갱신할 것을 권장합니다. 데이터베이스 실행기록에 있는 백업 이미지 위치를 갱신하지 않으면 자동 리스토어로 백업 이미지를 찾을 수 없으므로 직접 리스토어 명령을 사용할 수 있습니다.

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

*object-part* 또는 *eid*는 연결된 데이터베이스 파티션의 로그 실행기록 항목을 나타내야 합니다.

## ADMIN\_CMD 프로시저를 사용하는 UPDATE STMM TUNING DBPARTITIONNUM 명령

사용자 선호 자체 성능 조정 메모리 관리자(STMM) 조정 데이터베이스 파티션을 갱신 하십시오.

### 권한 부여

명령문의 권한 부여 ID에 의해 보유된 특권에는 다음과 같은 권한 중 하나 이상을 포함해야 합니다.

- DBADM
- DATAACCESS
- SQLADM

### 필수 연결

데이터베이스

### 명령 구문

►►—UPDATE—STMM—TUNING—DBPARTITIONNUM—*partitionnum*—————►►

### 명령 매개변수

*partitionnum*

*partitionnumdms* 정수입니다. -1 또는 nonexistent 데이터베이스 파티션 수가 사용된 경우, DB2가 자동으로 적합한 데이터베이스 파티션을 선택하여 STMM 메모리 조정 프로그램을 실행합니다.

### 예 :

사용자 선호 자체 성능 조정 메모리 관리자(STMM) 조정 데이터베이스 파티션을 데이터베이스 파티션 3으로 갱신하십시오.

```
CALL SYSPROC.ADMIN_CMD( 'update stmm tuning dbpartitionnum 3' )
```

### 사용 시 참고사항

STMM 조정 프로세스는 사용자 선호 STMM 조정 데이터베이스 파티션 수 값의 변경 사항을 정기적으로 점검합니다. *partitionnum*이 존재하고 활성 데이터베이스 파티션인 경우 STMM 조정 프로세스는 사용자 선호 STMM 조정 데이터베이스 파티션으로 이

동합니다. 이 명령이 STMM 조정 데이터베이스 파티션 수를 한 번 변경하면 현재 STMM 조정 데이터베이스 파티션 수가 즉시 변경됩니다.

명령 실행 상태는 CALL문에서 생성된 SQLCA에 리턴됩니다.

이 명령은 ADMIN\_CMD 프로시저에서 해당 변경사항을 커밋합니다.

---

## ADMIN\_EST\_INLINE\_LENGTH 함수 - 데이터 인라인에 필요한 길이 추정

ADMIN\_EST\_INLINE\_LENGTH 함수는 XML 컬럼, BLOB 컬럼, CLOB 컬럼 또는 DBCLOB 컬럼으로 저장된 데이터를 인라인하는 데 필요한 인라인 길이에 대한 예상치를 리턴합니다.

데이터를 인라인할 수 없는 경우 함수는 음수 값을 리턴합니다.

데이터가 이미 인라인되어 있는 경우 함수는 인라인된 데이터의 실제 길이를 리턴합니다.

### 구문

▶▶—ADMIN\_EST\_INLINE\_LENGTH—(—*column-name*—)————▶▶

스키마는 SYSIBM입니다.

### 리턴 값

이 함수는 데이터의 개략적인 인라인 길이(바이트)를 나타내는 INTEGER 값 또는 다음 값 중 하나를 리턴합니다.

NULL 입력이 NULL입니다.

- 1 컬럼 값이 인라인될 수 있도록 하는 유효한 인라인 길이가 없기 때문에 데이터가 인라인될 수 없음을 나타냅니다.
- 2 문서가 Linux, UNIX 및 Windows용 DB2 버전 9.7 이전의 릴리스에 삽입되어 저장되었기 때문에 문서의 개략적인 인라인 길이를 판별할 수 없음을 나타냅니다.

### 함수 매개변수

*column-name*

XML, BLOB, CLOB 또는 DBCLOB 데이터 유형의 기본 테이블 컬럼을 식별합니다(SQLSTATE 42884). 컬럼은 표현식을 기반으로 생성되지 않은 기본 테이블 컬럼을 직접 또는 간접적으로 참조해야 합니다(SQLSTATE 42815).

## 예

예 1: 다음 예는 TAB1 테이블의 xml\_doc1 XML 컬럼에 포함된 XML 문서 세 개의 개략적인 인라인 길이를 리턴합니다.

```
db2 => SELECT PK, ADMIN_IS_INLINED(xml_doc1) as IS_INLINED,
        ADMIN_EST_INLINE_LENGTH(xml_doc1) as EST_INLINE_LENGTH
        from TAB1
```

이 쿼리는 다음 출력을 작성합니다.

PK	IS_INLINED	EST_INLINE_LENGTH
1	1	292
2	0	450
3	0	454

3개의 레코드가 선택되었습니다.

예에서 ADMIN\_IS\_INLINED 함수는 첫 번째 문서가 인라인되었음을 보여줍니다. 따라서 ADMIN\_EST\_INLINE\_LENGTH 함수는 인라인된 XML 문서의 실제 길이를 리턴합니다. 두 번째 문서는 인라인되지 않았기 때문에 ADMIN\_EST\_INLINE\_LENGTH 함수는 두 번째 XML 문서를 인라인하는 데 필요한 개략적인 인라인 길이를 리턴합니다.

예 2: 다음 예는 TAB1 테이블의 xml\_doc1 XML 컬럼에 포함된 XML 문서 한 개의 개략적인 인라인 길이를 리턴합니다. 이 예에는 술어가 포함됩니다.

```
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
```

이 쿼리는 다음 출력을 작성합니다.

PK	IS_INLINED	EST_INLINE_LENGTH
2	0	450

1개의 레코드가 선택되었습니다.

예 3: 다음 예는 TAB1 테이블의 clob\_1 CLOB 컬럼에 포함된 CLOB 데이터 세 개의 개략적인 인라인 길이를 리턴합니다.

```
db2 => SELECT PK, ADMIN_IS_INLINED(clob_1) as IS_INLINED,
        ADMIN_EST_INLINE_LENGTH(clob_1) as EST_INLINE_LENGTH
        from TAB1
```

이 쿼리는 다음 출력을 작성합니다.

PK	IS_INLINED	EST_INLINE_LENGTH
1	1	68

2	0	3665
3	0	-1

3개의 레코드가 선택되었습니다.

### 사용 시 참고사항

- XML 컬럼은 XML 문서가 Linux, UNIX 및 Windows용 DB2 버전 9.7 이상으로 삽입된 경우에만 지원됩니다. 이 릴리스 이전에 삽입된 XML 문서의 스토리지 형식이 다릅니다. ADMIN\_EST\_INLINE\_LENGTH 함수가 올바르게 읽은 스토리지 형식을 발견하고 값으로 -2를 리턴합니다.
- 컬럼 인라인 길이를 늘리려면 이 길이를 줄일 수 없습니다.
- 인라인 길이를 늘리면 전체 행 크기도 증가되어 버퍼 풀의 성능에도 영향을 줄 수 있습니다. 전체 행 크기에는 다음과 같은 한계가 있습니다.

표 66. 행 크기 한계

페이지 크기	행 크기 한계	인라인 길이 한계
4K	4005	4001
8K	8101	8097
16K	16 293	16 289
32K	32 677	32 673

- XML 스토리지 오브젝트 페이지 크기가 기본 테이블 페이지 크기와 다른 경우 추정된 인라인 길이는 정확하지 않을 수도 있습니다.

## ADMIN\_GET\_DBP\_MEM\_USAGE 테이블 함수 - 인스턴스에 대한 전체 메모리 소비 가져오기

ADMIN\_GET\_DBP\_MEM\_USAGE 테이블 함수는 제공된 인스턴스에 대한 전체 메모리 소비를 가져옵니다.

ADMIN\_GET\_DBP\_MEM\_USAGE 테이블 함수는 유효한 데이터베이스 파티션 번호를 지정하는 선택적 입력 인수 *dbpartitionnum* (INTEGER 유형)를 취하여 해당 단일 데이터베이스 파티션의 통계만 리턴합니다. 인수가 생략되면, 모든 활성 데이터베이스 파티션에 대한 통계가 리턴됩니다. DPF(Database Partitioning Feature)를 사용할 때 *dbpartitionnum*에 -1 또는 널(NULL) 값을 지정하면 현재 연결된 파티션에서 데이터가 리턴됩니다.

### 구문

```

ADMIN_GET_DBP_MEM_USAGE( ( dbpartitionnum ) )

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

메모리 사용 통계를 검색할 데이터베이스 파티션을 지정하는 정수 유형의 선택적 입력 인수. -1 또는 널(NULL) 값이 지정되면 현재 연결된 파티션에서 데이터가 리턴됩니다.

### 권한 부여

ADMIN\_GET\_DBP\_MEM\_USAGE 함수에 대한 EXECUTE 특권

### 리턴되는 정보

표 67. ADMIN\_GET\_DBP\_MEM\_USAGE에 대한 결과 세트

컬럼 이름	데이터 유형	설명
DBPARTITIONNUM	SMALLINT	메모리 사용 통계를 검색한 데이터베이스 파티션 번호
MAX_PARTITION_MEM	BIGINT	데이터베이스 파티션에 소비하도록 허용된 최대 인스턴스 메모리 양(바이트)
CURRENT_PARTITION_MEM	BIGINT	현재 데이터베이스 파티션에 소비된 인스턴스 메모리의 양(바이트)
PEAK_PARTITION_MEM	BIGINT	데이터베이스 파티션의 인스턴스 메모리의 최대 또는 상위 워터 마크(water mark) 소비량(바이트)

### 예:

예 1: 데이터베이스 파티션 3에서 메모리 사용 통계 검색

```
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

1개의 레코드가 선택되었습니다.

예 2: 현재 연결된 파티션의 메모리 사용 통계 검색 (사용자가 파티션 2의 데이터베이스에 연결된 것으로 가정)

```
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

1개의 레코드가 선택되었습니다.

예 3: 모든 파티션에서 메모리 사용 통계 검색

```
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

4개의 레코드가 선택되었습니다.

**예 4: 메가바이트(MB) 값으로 메모리 사용 통계 검색**

```
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

3개의 레코드가 선택되었습니다.

---

## ADMIN\_GET\_INDEX\_COMPRESS\_INFO 테이블 함수 - 압축된 인덱스 정보 리턴

ADMIN\_GET\_INDEX\_COMPRESS\_INFO 테이블 함수는 압축되지 않은 인덱스에 대한 잠재적인 인덱스 압축 절약을 리턴하거나 카탈로그에서 인덱스 압축 통계를 보고합니다.

### 구문

```
▶—ADMIN_GET_INDEX_COMPRESS_INFO—(—objecttype—,—objectschema—,—objectname—,—
▶—dbpartitionnum—,—datapartitionid—)
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### objecttype

오브젝트 유형을 표시하는 VARCHAR(1) 유형의 입력 인수입니다. 값은 다음과 같은 대소문자가 구분되는 값 중 하나여야 합니다.

- 테이블을 표시하는 'T', NULL 또는 비어 있는 문자열
- 인덱스에 대한 'I'

#### objectschema

오브젝트 스키마를 지정하는 VARCHAR (128) 유형의 대소문자가 구분되는 입력 매개변수입니다.

objecttype이 'T', NULL 또는 비어 있는 문자열("")인 경우 objectschema는 테이블 스키마를 나타냅니다.

- objectschema가 지정되고 objectname이 NULL 또는 비어 있는 문자열("")인 경우, 지정한 스키마의 모든 테이블에서 모든 인덱스에 대해 정보가 리턴됩니다.



- *objectschema* 및 *objectname*을 모두 지정한 경우, 지정한 테이블에서 모든 인덱스에 대해 정보가 리턴됩니다.

*objecttype*이 'I'인 경우, *objectschema*는 인덱스 스키마를 표시합니다.

- *objectschema*가 지정되고 *objectname*이 NULL 또는 비어 있는 문자열("")인 경우, 지정한 스키마의 모든 인덱스에 대해 정보가 리턴됩니다.
- *objectschema* 및 *objectname*을 모두 지정한 경우, 지정한 인덱스에 대해 정보가 리턴됩니다.
- *objectschema* 또는 *objectname*을 모두 지정하지 않은 경우, 모든 스키마의 모든 인덱스에 대해 정보가 리턴됩니다.

*objectname*은 지정하고 *objectschema*는 지정하지 않은 경우, 함수가 SQL 오류를 리턴합니다. 매개변수 값이 비어 있는 문자열("") 또는 NULL 값인 경우 해당 값은 지정되지 않은 것입니다.

#### *objectname*

오브젝트 이름을 지정하는 VARCHAR

(128) 유형의 대소문자가 구분되는 입력 매개변수입니다. *objectschema* 매개변수에 대한 설명을 참조하십시오.

#### *dbpartitionnum*

데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 입력 매개변수입니다. 지정한 경우 지정한 데이터베이스 파티션에 있는 인덱스에 대한 정보만 리턴됩니다. 사용 중인 모든 데이터베이스 파티션에 대해 데이터가 리턴되도록 지정하려면 *dbpartitionnum* 매개변수 값을 -2 또는 NULL로 설정하십시오. 파티션되지 않은 데이터베이스 환경에서는 -2 또는 NULL을 지정하십시오.

#### *datapartitionid*

데이터 파티션 ID를 지정하는 INTEGER 유형의 입력 매개변수입니다. 지정한 경우 지정한 데이터베이스 파티션에 정의된 인덱스 파티션에 대한 정보만 리턴됩니다. 데이터 파티션 ID는 SYSCAT.DATAPARTITIONS 뷰의 DATAPARTITIONID에 대응해야 합니다. 모든 데이터베이스 파티션에 대해 데이터가 리턴되도록 지정하려면 *datapartitionid* 매개변수 값을 -2 또는 NULL로 설정하십시오. 파티션되지 않은 인덱스에 대해서는 -2, 0 또는 NULL을 지정하십시오.

## 권한 부여

ADMIN\_GET\_INDEX\_COMPRESS\_INFO 테이블 함수에 대한 EXECUTE 특권

예 :

데이터베이스 이주 후에는 기존의 모든 인덱스가 압축되지 않습니다. 데이터 파티션 ID가 3이고 데이터베이스 파티션 번호 2에 있는 "S.T1" 테이블의 기존 인덱스에 대한 인덱

스 압축으로 인한 잠재적인 절약을 추정하려고 할 수 있습니다. 이 예에서 S는 스키마 이름이고 T1은 테이블 이름으로 T1은 압축되지 않습니다.

```
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
```

다음은 이 명령문의 출력 샘플입니다.

```
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
```

압축으로 인한 절약의 효과를 인지하고 인덱스 압축을 사용하고자 할 것입니다.

```
ALTER INDEX INDEX1 compress yes
ALTER INDEX INDEX2 compress yes
REORG INDEXES all FOR table S.T1
```

시간 경과에 따라 테이블에 대해 새 인덱스를 작성하고 이를 압축하기 전에 해당 인덱스 압축으로 작성된 절약의 예상치를 계산하려고 할 수 있습니다. 이미 압축된 인덱스에서 압축 통계를 확인하려고 할 수도 있습니다.

```
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
```

다음은 이 명령문의 출력 샘플입니다.

```
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
```

첫 번째 두 인덱스가 index\_compressed 컬럼에서 표시된 대로 이미 압축되어 있기 때문에 명령문은 시스템 카탈로그에서 값을 리턴합니다. 이 경우 카탈로그의 값은 수집되지 않습니다.

테이블에서 RUNSTATS를 실행한 후에, 다음에 인덱스 함수를 실행하면 수정된 결과가 작성됩니다.

```

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

```

다음은 이 명령문의 출력 샘플입니다.

```

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

```

## ADMIN\_GET\_INDEX\_COMPRESS\_INFO 테이블 함수 메타데이터

표 68. ADMIN\_GET\_INDEX\_COMPRESS\_INFO 테이블 함수 메타데이터

컬럼 이름	데이터 유형	설명
INDSCHEMA	VARCHAR (128)	인덱스가 정의된 스키마 이름
INDNAME	VARCHAR (128)	인덱스 이름.
TABSCHEMA	VARCHAR (128)	테이블이 정의된 스키마 이름
TABNAME	VARCHAR (128)	테이블 이름.
DBPARTITIONNUM	SMALLINT	데이터베이스 파티션 번호.
IID	SMALLINT	인덱스 ID.
DATAPARTITIONID	INTEGER	데이터 파티션 ID
COMPRESS_ATTR	CHAR(1)	인덱스의 COMPRESSION 속성 상태 <ul style="list-style-type: none"> <li>『Y』 = 인덱스 압축이 사용됩니다.</li> <li>『N』 = 인덱스 압축이 사용되지 않습니다.</li> </ul>
INDEX_COMPRESSED	CHAR(1)	실제 인덱스 형식 <ul style="list-style-type: none"> <li>『Y』 = 인덱스가 압축된 형식입니다.</li> <li>『N』 = 인덱스가 압축되지 않은 형식입니다.</li> </ul> <p>실제 인덱스 형식이 압축 속성과 일치하지 않는 경우에 인덱스를 정의된 형식으로 변환하려면 인덱스를 재구성해야 합니다. 이 함수 실행 시에 테이블이나 인덱스에 오류가 있는 경우 이 값은 NULL입니다.</p>

표 68. ADMIN\_GET\_INDEX\_COMPRESS\_INFO 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
PCT_PAGES_SAVED	SMALLINT	인덱스가 실제로 압축되지 않은 경우(INDEX_COMPRESSED가 『N』) 이 값은 인덱스가 실제로 압축된 것처럼 저장된 리프 페이지의 개략적인 백분율을 나타냅니다. 인덱스가 실제로 압축된 경우(INDEX_COMPRESSED가 『Y』) 이 값은 시스템 카탈로그 뷰(SYSCAT.INDEXES 또는 SYSCAT.INDEXPARTITIONS)에서 PCTPAGESSAVED 값을 보고합니다. 주: 이 값은 DPF 환경에서 각 데이터베이스 파티션에 대한 인덱스 또는 인덱스 파티션의 각 항목에 대해 동일합니다. 이 함수 실행 시에 테이블이나 인덱스에 오류가 있는 경우 이 값은 NULL입니다.
NUM_LEAF_PAGES_SAVED	BIGINT	인덱스가 실제로 압축되지 않은 경우(INDEX_COMPRESSED가 『N』) 이 값은 인덱스가 실제로 압축된 것처럼 저장된 리프 페이지의 개략적인 수를 나타냅니다. 인덱스가 실제로 압축된 경우(INDEX_COMPRESSED가 『Y』) 이 값은 시스템 카탈로그 뷰(SYSCAT.INDEXES 또는 SYSCAT.INDEXPARTITIONS)에서 PCTPAGESSAVED 및 NLEAF 값을 바탕으로 저장된 리프 페이지의 계산된 수를 보고합니다. PCTPAGESSAVED 또는 NLEAF가 유효하지 않은 값(-1)인 경우 이 값도 -1로 설정됩니다. 주: 이 값은 DPF 환경에서 각 데이터베이스 파티션에 대한 인덱스 또는 인덱스 파티션의 각 항목에 대해 동일합니다. 이 함수 실행 시에 테이블이나 인덱스에 오류가 있는 경우 이 값은 NULL입니다.

## ADMIN\_GET\_INDEX\_INFO 테이블 함수 - 인덱스 정보 리턴

ADMIN\_GET\_INDEX\_INFO 테이블 함수는 카탈로그 뷰에서 사용할 수 없는 인덱스 정보(예: 압축 정보와 논리 및 실제 인덱스 크기)를 리턴합니다.

### 구문

▶▶ ADMIN\_GET\_INDEX\_INFO(—objecttype—, —objectschema—, —objectname—) ▶▶

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### objecttype

오브젝트 유형을 표시하는 VARCHAR(1) 유형의 입력 인수입니다. 값은 다음과 같은 대소문자가 구분되는 값 중 하나여야 합니다.

- 테이블을 표시하는 'T', NULL 또는 비어 있는 문자열("")
- 인덱스에 대한 'I'

#### objectschema

오브젝트 스키마를 지정하는 VARCHAR

(128) 유형의 대소문자가 구분되는 입력 매개변수입니다.

*objecttype*이 'T', NULL 또는 비어 있는 문자열('')인 경우, *objectschema*는 테이블 스키마를 표시합니다.

- *objectschema*가 지정되고 *objectname*이 NULL 또는 비어 있는 문자열('')인 경우, 지정한 스키마의 모든 테이블에서 모든 인덱스에 대해 정보가 리턴됩니다.
- *objectschema* 및 *objectname*을 모두 지정한 경우, 지정한 테이블에서 모든 인덱스에 대해 정보가 리턴됩니다.

*objecttype*이 'I'인 경우, *objectschema*는 인덱스 스키마를 표시합니다.

- *objectschema*가 지정되고 *objectname*이 NULL 또는 비어 있는 문자열('')인 경우, 지정한 스키마의 모든 인덱스에 대해 정보가 리턴됩니다.
- *objectschema* 및 *objectname*을 모두 지정한 경우, 지정한 인덱스에 대해 정보가 리턴됩니다.
- *objectschema* 또는 *objectname*을 모두 지정하지 않은 경우, 모든 스키마의 모든 인덱스에 대해 정보가 리턴됩니다.

*objectname*은 지정하고 *objectschema*는 지정하지 않은 경우, 함수가 SQL 오류를 리턴합니다. 매개변수 값이 비어 있는 문자열('') 또는 NULL 값인 경우 해당 값은 지정되지 않은 것입니다.

#### *objectname*

오브젝트 이름을 지정하는 VARCHAR

(128) 유형의 대소문자가 구분되는 입력 매개변수입니다. *objectschema* 매개변수에 대한 설명을 참조하십시오.

## 권한 부여

ADMIN\_GET\_INDEX\_INFO 테이블 함수에 대한 EXECUTE 특권

예 :

테이블에서 여러 인덱스에 대해 인덱스 압축을 설정한 후에는 압축된 인덱스 및 압축하기 위해 재빌드해야 하는 인덱스를 판별하려고 할 수 있습니다. 이 예에서, S는 스키마 이름이고 T1은 테이블 이름입니다.

```
db2 SELECT iid, compress_attr, index_compressed
      FROM TABLE(sysproc.admin_get_index_info('', 'S', 'T1')) AS t
```

다음은 이 쿼리의 출력 예입니다.

IID	COMPRESS_ATTR	INDEX_COMPRESSED
1	Y	Y
2	Y	Y
3	Y	N
4	N	N

또는 스키마 S2의 모든 인덱스에 대해 다른 인덱스 정보를 표시하려고 할 수도 있습니다. 이 예에서는 다음과 같습니다.

- T2 = 두 개의 데이터 파티션이 있는 파티션된 테이블
- T3 = 파티션되지 않은 테이블
- IND\_1 = T2에서 파티션되지 않은 인덱스
- IND\_2 = T2에서 파티션된 인덱스
- IND\_3 = T2에서 파티션된 인덱스
- IND\_4 = T3의 인덱스
- IND\_5 = T3의 인덱스

```
db2 SELECT tabname, 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
```

다음은 이 쿼리로부터의 출력 예입니다.

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

이 프로시저의 출력(계속):

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

## ADMIN\_GET\_INDEX\_INFO 테이블 함수 메타데이터

표 69. ADMIN\_GET\_INDEX\_INFO 테이블 함수 메타데이터

컬럼 이름	데이터 유형	설명
INDSHEMA	VARCHAR (128)	인덱스가 정의된 스키마 이름
INDNAME	VARCHAR (128)	인덱스 이름.
TABSHEMA	VARCHAR (128)	테이블이 정의된 스키마 이름
TABNAME	VARCHAR (128)	테이블 이름.

표 69. ADMIN\_GET\_INDEX\_INFO 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
DBPARTITIONNUM	SMALLINT	데이터베이스 파티션 번호.
IID	SMALLINT	인덱스 ID.
DATAPARTITIONID	INTEGER	데이터 파티션 ID
COMPRESS_ATTR	CHAR(1)	인덱스의 COMPRESSION 속성 상태 <ul style="list-style-type: none"> <li>『Y』 = 인덱스 압축이 사용됩니다.</li> <li>『N』 = 인덱스 압축이 사용되지 않습니다.</li> </ul>
INDEX_COMPRESSED	CHAR(1)	실제 인덱스 형식 <ul style="list-style-type: none"> <li>『Y』 = 인덱스가 압축된 형식입니다.</li> <li>『N』 = 인덱스가 압축되지 않은 형식입니다.</li> </ul> <p>실제 인덱스 형식이 압축 속성과 일치하지 않는 경우에 인덱스를 정의된 형식으로 변환하려면 인덱스를 재구성해야 합니다. 이 함수 실행 시 테이블 또는 인덱스에 오류가 있는 경우, 이 값은 NULL입니다.</p>
INDEX_PARTITIONING	CHAR(1)	인덱스의 파티셔닝 특성을 식별합니다. <ul style="list-style-type: none"> <li>『N』 = 파티션되지 않은 인덱스</li> <li>『P』 = 파티션된 인덱스</li> <li>공백 = 인덱스가 파티션된 테이블에 없음</li> </ul>
INDEX_OBJECT_L_SIZE	BIGINT	인덱스 오브젝트의 논리적 크기. 파티션되지 않은 테이블의 경우 테이블에 정의된 모든 인덱스에 논리적으로 할당된 디스크 스페이스 크기입니다. 파티션된 테이블에서 파티션되지 않은 인덱스의 경우 이는 인덱스에 대해 논리적으로 할당된 디스크 스페이스 크기입니다. 파티션된 테이블의 파티션된 인덱스의 경우 데이터 파티션에 정의된 모든 인덱스 파티션에 대해 논리적으로 할당된 디스크 스페이스 용량입니다. 모든 크기는 KB 단위로 보고됩니다.  <p>논리적 크기는 테이블이나 데이터 파티션이 인식하는 스페이스 용량입니다. 이는 테이블 또는 데이터 파티션에 대한 인덱스 데이터를 보유하도록 실제로 할당된 스페이스 용량보다 작을 수도 있습니다(예: 논리 테이블 절단의 경우). 리턴된 크기는 인덱스, DMS 테이블 스페이스에 작성된 인덱스 및 EMP Extent의 예상치에 대해 논리적으로 할당된 전체 Extent가 됩니다. 이 함수 실행 시 테이블 또는 인덱스에 오류가 있는 경우, 이 값은 NULL입니다.</p>
INDEX_OBJECT_P_SIZE	BIGINT	인덱스 오브젝트의 실제 크기. 파티션되지 않은 테이블의 경우 테이블에 정의된 모든 인덱스에 실제로 할당된 디스크 스페이스 용량입니다. 파티션된 테이블에서 파티션되지 않은 인덱스의 경우 인덱스에 대해 실제로 할당된 디스크 스페이스 용량입니다. 파티션된 테이블의 파티션된 인덱스의 경우 데이터 파티션에 정의된 모든 인덱스 파티션에 대해 실제로 할당된 디스크 스페이스 용량입니다. 모든 크기는 KB 단위로 보고됩니다.  <p>리턴된 크기는 인덱스에 대해 할당된 전체 크기로 간주되며 DMS 테이블 스페이스에서 작성된 인덱스에 대한 EMP extent를 포함합니다. 이 함수 실행 시 테이블 또는 인덱스에 오류가 있는 경우, 이 값은 NULL입니다.</p>

표 69. ADMIN\_GET\_INDEX\_INFO 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
INDEX_REQUIRES_REBUILD	CHAR(1)	인덱스의 재빌드 상태 <ul style="list-style-type: none"> <li>『Y』 - 테이블 또는 데이터 파티션에 정의된 인덱스를 재빌드해야 하는 경우</li> <li>『N』 - 이외의 경우</li> </ul> 이 함수 실행 시 테이블에 오류가 있는 경우, 이 값은 NULL입니다.
LARGE_RIDS	CHAR(1)	인덱스가 대형 행 ID(RID)(4바이트의 페이지 수, 2바이트의 슬롯 수)를 사용하는지 여부를 표시합니다. <ul style="list-style-type: none"> <li>『Y』 - 인덱스가 대형 RID를 사용 중임을 표시합니다.</li> <li>『N』 - 인덱스가 대형 RID를 사용 중이 아님을 표시합니다.</li> <li>『P』(보류) - 인덱스가 정의된 테이블이 대형 RID를 지원하지만(즉, 테이블이 대형 테이블 스페이스에 있는 경우) 테이블 또는 데이터 파티션의 인덱스가 아직 재구성 또는 재빌드되지 않았음을 표시합니다. 따라서 테이블이 4바이트 RID를 사용하므로 테이블 또는 인덱스를 대형 RID로 변환하는 조치를 수행해야 합니다.</li> </ul> 이 함수 실행 시 테이블에 오류가 있는 경우, 이 값은 NULL입니다.

## ADMIN\_GET\_MSGS 테이블 함수 - ADMIN\_CMD 프로시저에서 실행된 데이터 이동 유틸리티에서 생성된 메시지 검색

ADMIN\_GET\_MSGS 테이블 함수는 ADMIN\_CMD 프로시저를 통해 데이터 이동 유틸리티 명령의 단독 실행으로 생성된 메시지를 검색하는 데 사용됩니다. 입력 매개변수 *operation\_id* 는 해당 조작을 식별합니다.

### 구문

▶▶ ADMIN\_GET\_MSGS(—*operation\_id*—)▶▶

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *operation\_id*

ADMIN\_CMD 프로시저를 통해 실행된 데이터 이동 유틸리티에서 생성된 메시지 파일의 조작 ID를 지정하는 VARCHAR(139) 유형의 입력 인수. 조작 ID는 ADMIN\_CMD 프로시저에서 생성됩니다.

### 권한 부여

ADMIN\_GET\_MSGS 테이블 함수에 대한 EXECUTE 특권. 분리 사용자 ID는 레지스트리 변수 DB2\_UTIL\_MSGSPATH로 표시된 디렉토리 아래의 파일에 대한 읽기 액세스



세스 권한이 있어야 합니다. 레지스트리 변수가 설정되어 있지 않을 경우, 분리 사용자 ID에 인스턴스 디렉토리 아래 tmp 서브디렉토리의 파일에 대한 읽기 액세스 권한이 있어야 합니다.

**예 :**

조작 ID가 '24523\_THERESAX'인 ADMIN\_CMD 프로시저를 통해 실행된 EXPORT 유틸리티에서 리턴된 모든 메시지를 확인하십시오.

```
SELECT * FROM TABLE(SYSPROC.ADMIN_GET_MSGS('24523_THERESAX')) AS MSG
```

이 쿼리의 출력 예는 다음과 같습니다.

```
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.
```

2개의 레코드가 선택되었습니다.

## 사용 시 참고사항

적절한 *operation\_id*로 이 테이블 함수를 호출하는 쿼리 명령문은 ADMIN\_CMD 프로시저에서 리턴된 첫 번째 결과 세트의 MSG\_RETRIEVAL 컬럼에 있습니다.

## 리턴되는 정보

표 70. ADMIN\_GET\_MSGS 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명
DBPARTITIONNUM	INTEGER	데이터베이스 파티션 번호. 이 값은 분산 로드 에 대해서만 리턴되며 해당 메시지에 대한 데이터베이스 파티션을 나타냅니다.
AGENTTYPE	CHAR(4)	에이전트 유형. 이 값은 분산 로드 에 대해서만 리턴됩니다. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• 'LOAD': 로드 에이전트용</li> <li>• 'PART': 파티션 에이전트용</li> <li>• 'PREP': 사전 파티션 에이전트용</li> <li>• NULL: 사용 가능한 에이전트 유형 정보가 없음</li> </ul>
SQLCODE	VARCHAR(9)	리턴되는 메시지의 SQLCODE
MSG	VARCHAR(1024)	SQLCODE에 해당하는 간단한 오류 메시지

## ADMIN\_IS\_INLINED 함수 - 데이터가 인라인인지 판별

ADMIN\_IS\_INLINED 함수는 XML 컬럼, BLOB 컬럼, CLOB 컬럼 또는 DBCLOB 컬럼의 인라인 데이터에 대한 상태 정보를 검색합니다.

### 구문

```
▶▶ ADMIN_IS_INLINED(—column-name—)▶▶
```

스키마는 SYSIBM입니다.

### 리턴 값

이 함수는 SMALLINT 유형의 다음 값 중 하나 또는 널(NULL) 값을 리턴합니다.

1 데이터가 인라인입니다.

0 데이터가 인라인이 아닙니다.

NULL 입력이 NULL입니다.

### 함수 매개변수

*column-name*

XML, BLOB, CLOB 또는 DBCLOB 데이터 유형의 기본 테이블 컬럼을 식별합니다(SQLSTATE 42884). 컬럼은 표현식을 기반으로 생성되지 않은 기본 테이블 컬럼을 직접 또는 간접적으로 참조해야 합니다(SQLSTATE 42815).

### 예

예 1: 다음 예는 TAB1 테이블의 xml\_doc1 XML 컬럼에서 세 개의 XML 문서가 인라인되어 있는지를 나타냅니다.

```
db2 => SELECT PK, ADMIN_IS_INLINED(xml_doc1) as IS_INLINED
        from TAB1
```

이 쿼리는 다음 출력을 작성합니다.

PK	IS_INLINED
1	1
2	0
3	0

3개의 레코드가 선택되었습니다.

예 2: 다음 예는 TAB1 테이블의 xml\_doc1 XML 컬럼에서 XML 문서 중 하나가 인라인되어 있는지를 나타냅니다.

```
db2 => SELECT PK, ADMIN_IS_INLINED(xml_doc1) as IS_INLINED
       from TAB1 where PK=1
```

이 쿼리는 다음 출력을 작성합니다.

PK	IS_INLINED
1	1

1개의 레코드가 선택되었습니다.

예 3: 다음 예는 TAB1 테이블의 clob\_1 CLOB 컬럼에 포함된 세 개의 CLOB 데이터가 인라인되었는지를 나타냅니다.

```
db2 => SELECT PK, ADMIN_IS_INLINED(clob_1) as IS_INLINED
       from TAB1
```

이 쿼리는 다음 출력을 작성합니다.

PK	IS_INLINED
1	0
2	0
3	1

3개의 레코드가 선택되었습니다.

---

## ADMIN\_REMOVE\_MSGS 프로시저 - ADMIN\_CMD 프로시저에서 실행된 데이터 이동 유틸리티에서 생성된 메시지 정리

ADMIN\_REMOVE\_MSGS 프로시저는 ADMIN\_CMD 프로시저를 통해 데이터 이동 유틸리티 명령의 단독 실행으로 인해 생성된 메시지를 정리하는 데 사용됩니다. 입력 매개변수 *operation\_id* 는 조작을 식별합니다.

### 구문

```
►►—ADMIN_REMOVE_MSGS—(—operation_id—)—————►►
```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *operation\_id*

ADMIN\_CMD 프로시저를 통해 실행된 데이터 이동 유틸리티에서 생성된 메시지 파일의 조작 ID를 지정하는 VARCHAR(139) 유형의 입력 인수. 조작 ID는 ADMIN\_CMD 프로시저에서 생성됩니다.

## 권한 부여

ADMIN\_REMOVE\_MSGS 프로시저의 EXECUTE 특권. 분리 사용자 ID는 레지스 트리 변수 DB2\_UTIL\_MSGPATH로 표시된 디렉토리 아래의 파일을 삭제할 수 있어야 합니다. 레지스 트리 변수가 설정되어 있지 않을 경우, 분리 사용자 ID가 인스턴스 디렉토리 아래 tmp 서브디렉토리의 파일을 삭제할 수 있어야 합니다.

예 :

조작 ID가 '24523\_THERESAX'인 메시지를 정리하십시오.

```
CALL SYSPROC.ADMIN_REMOVE_MSGS('24523_THERESAX')
```

## 사용 시 참고사항

적절한 *operation\_id*로 이 프로시저를 호출하는 CALL문은 ADMIN\_CMD 프로시저에서 리턴된 첫 번째 결과 세트의 MSG\_REMOVAL 컬럼에 있습니다.

---

## ADMIN\_REVALIDATE\_DB\_OBJECTS 프로시저 - 유효하지 않은 데이터베이스 오브젝트 유효성 다시 확인

ADMIN\_REVALIDATE\_DB\_OBJECTS 프로시저는 유효하지 않은 데이터베이스 오브젝트의 유효성을 다시 확인합니다.

이 프로시저는 *object\_type*, *object\_schema* 및 *object\_name*의 3개의 입력 매개변수가 있으며 수행되는 유효성 다시 확인의 레벨을 제어합니다.

- 데이터베이스에서 유효하지 않은 오브젝트 모두의 유효성을 다시 확인하려면 모든 매개변수에 대해 널(NULL)을 지정하거나 매개변수 없이 프로시저를 호출합니다.
- 특정 스키마에서 유효하지 않은 모든 데이터베이스 오브젝트의 유효성을 다시 확인하려면 *object\_schema*에 대해 값을 지정하고 *object\_name* 및 *object\_type*에 대해 널(NULL)을 지정합니다.
- 유효하지 않은 특정 데이터베이스 오브젝트의 유효성을 다시 확인하려면 모든 매개변수에 대해 유효한 값을 지정합니다.

## 구문

```
▶▶—ADMIN_REVALIDATE_DB_OBJECTS—(—object_type—,—object_schema—,——————▶  
▶—object_name—)—————▶▶
```

스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *object\_type*

데이터베이스 오브젝트 유형을 식별하는 VARCHAR(30) 유형의 입력 인수. 다음 유형은 유효합니다.

- FUNCTION
- GLOBAL VARIABLE
- METHOD
- MODULE
- PROCEDURE
- SPECIFIC
- TABLE
- TRIGGER
- TYPE
- VIEW

이 값은 대소문자를 구분하지 않습니다. 이 값은 널(NULL)이 될 수 있습니다.

이들 유형 중 하나가 지정되면 이 프로시저는 MODULE에 속한 오브젝트를 제외한 해당 유형에서 유효하지 않은 모든 오브젝트의 유효성을 다시 확인합니다. 모듈 내에 있는 오브젝트의 유효성을 다시 확인하려면 특정 모듈의 이름으로 MODULE 유형을 사용하며 해당 모듈 내의 유효하지 않은 모든 오브젝트의 유효성이 다시 확인됩니다.

둘 이상의 매개변수 서명이 있는 루틴이 있고 이들 루틴 중 하나의 유효성을 다시 확인하려면, 유효성을 다시 확인할 루틴 이름으로 SPECIFIC 유형을 사용하십시오.

TABLE 유형을 사용하면 지정된 테이블이 재구성되며 통계가 수집됩니다. 프로시저는 reorg-pending 상태에 있는 구체화된 쿼리 테이블이나 일반 테이블에 대해 reorg 유틸리티 그리고 runstats 유틸리티를 호출합니다. 프로시저는 runstats에 대한 사용자 프로파일이 있는 경우 이를 사용하려고 시도합니다. 그렇지 않으면 기본 runstats 조작이 호출됩니다.

### *object\_schema*

데이터베이스 오브젝트 참조를 규정하는 데 사용되는 스키마 이름을 식별하는 VARCHAR(128) 유형의 입력 인수. 이름은 대소문자를 구분합니다. 이 값은 널(NULL)이 될 수 있습니다.

### *object\_name*

데이터베이스 오브젝트를 식별하는 VARCHAR(128) 유형의 입력 인수. 이름은 대소문자를 구분합니다. 프로시저가 이들 유형의 오브젝트를 지원하지 않기 때문에 이 값은 유형이 지정된 테이블이나 행 함수의 값이 될 수 없습니다. 이러한 오브젝트의 이름이 지정되면 오류가 리턴됩니다. 이 값은 널(NULL)이 될 수 있습니다.

## 권한 부여

ADMIN\_REVALIDATE\_DB\_OBJECTS 프로시저의 EXECUTE 특권.

### 예:

예 1: 현재 데이터베이스에 있는 모든 항목의 유효성을 다시 확인합니다.

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS(NULL, NULL, NULL)
```

또는 매개변수 없이 프로시저를 호출합니다.

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS()
```

예 2: 스키마 MY\_SCHEMA에서 규정된 모든 오브젝트의 유효성을 다시 확인합니다.

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS(NULL, 'MY_SCHEMA', NULL)
```

예 3: 데이터베이스의 모든 트리거 오브젝트의 유효성을 다시 확인합니다.

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS('trigger', NULL, NULL)
```

예 4: 특정 뷰 오브젝트의 유효성을 다시 확인합니다.

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS('view', 'MY_SCHEMA', 'MY_VIEW')
```

예 5: MY\_SCHEMA의 모든 프로시저의 유효성을 다시 확인합니다. 이 예의 경우 이 스키마에 3개의 프로시저(proc1, proc2 및 proc3)가 있습니다. proc1에서 사용된 참조 오브젝트는 존재하지 않습니다. 다음 호출은 proc2 및 proc3의 유효성을 다시 확인하지만 proc1은 유효하지 않은 상태로 남습니다. 이 상황에서는 호출이 경고를 리턴합니다.

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS('procedure', 'MY_SCHEMA', NULL)
```

예 6: 존재하지 않은 오브젝트의 유효성을 다시 확인합니다. 이 예는 오류를 리턴합니다.

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS('procedure', 'MY_SCHEMA', 'MY_VIEW')
```

예 7: Oracle 표기를 사용하여 MY\_SCHEMA의 모든 프로시저의 유효성을 다시 확인합니다.

```
CALL SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS(object_type=>'PROCEDURE',object_schema=>'MY_SCHEMA')
```

## 사용 시 참고사항

ADMIN\_REVALIDATE\_DB\_OBJECTS 프로시저로 전달된 널이 아닌 모든 매개변수 값이 충족되어야 하거나, 프로시저는 유효성을 다시 확인해야 할 오브젝트를 식별할

수 없습니다. 예를 들어 트리거 유형을 제외하고 뷰 이름을 지정하면 유형이 일치하지 않으므로 프로시저는 뷰의 유효성을 다시 확인하지 않습니다.

이 프로시저는 reorg-pending 상태의 구체화된 쿼리 테이블이나 일반 테이블 또는 유효하지 않은 오브젝트의 유효성만 다시 확인합니다. 유효하지 않은 모든 오브젝트는 SYSCAT.INVALIDOBJECTS에서 찾을 수 있습니다. reorg-pending 상태에 있는 테이블을 찾으려면 ADMIN\_GET\_TAB\_INFO 테이블 함수를 사용하십시오.

유효한 오브젝트가 입력으로 지정되면 프로시저는 모든 조사를 수행하지 않으며 성공 코드를 리턴합니다. 테이블의 유효성 다시 확인에 실패하면 프로시저가 실패합니다. 다른 오브젝트의 유효성을 다시 확인하는 중에 실패하면 프로시저는 실패를 무시하고 계속해서 기타 오브젝트의 유효성을 다시 확인합니다. 적어도 하나가 실패하면 프로시저는 경고(SQLSTATE 0168B)를 리턴합니다. 모든 오브젝트의 유효성 다시 확인이 실패하면 프로시저는 오류(SQLSTATE 429C4)를 리턴합니다. 테이블을 제외한 오브젝트의 모든 유효성 다시 확인 실패에 대한 세부사항을 SYSCAT.INVALIDOBJECTS에서 찾을 수 있습니다.

전역 변수의 유효성이 다시 확인되면 현재 세션에 대해 인스턴스화될 수 있습니다.

테이블 유효성 다시 확인의 진행을 모니터링하려면, 연관된 테이블 reorg 조作的 진행을 모니터링할 수 있습니다. 기타 모든 오브젝트의 경우 SYSCAT.INVALIDOBJECTS 카탈로그 뷰를 쿼리하십시오. 오브젝트의 유효성이 다시 확인되면 이 뷰에서 오브젝트가 삭제되며 유효성이 제대로 다시 확인되지 못하면 항목이 갱신됩니다.

---

## ADMINTABCOMPRESSINFO 관리 뷰 및

### ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수 - 압축 정보 리턴

ADMINTABCOMPRESSINFO 관리 뷰 및

ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수는 테이블, 구체화된 쿼리 테이블(MQT) 및 계층 구조 테이블에 대한 압축 정보를 리턴합니다.

### ADMINTABCOMPRESSINFO 관리 뷰

ADMINTABCOMPRESSINFO 관리 뷰는 구체화된 쿼리 테이블(MQT), 계층 구조 테이블 및 테이블에 대한 압축 정보만을 리턴합니다. SYSCAT.TABLES 카탈로그 뷰에서 이들 테이블 유형은 테이블에 대해 T로, 구체화된 쿼리 테이블에 대해서는 S, 계층 구조 테이블에 대해서는 H로 보고됩니다. 테이블에 대한 데이터베이스 파티션 레벨 및 데이터 파티션 레벨 모두로 정보가 리턴됩니다.

스키마는 SYSIBMADM입니다.

리턴할 수 있는 전체 정보 목록은 #ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수 메타데이터 테이블을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- ADMINTABCOMPRESSINFO 관리 뷰에 대한 SELECT 특권
- ADMINTABCOMPRESSINFO 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

## 예:

예 1: 모든 테이블에 대한 모든 압축 정보를 검색합니다.

```
SELECT * FROM SYSIBMADM.ADMINTABCOMPRESSINFO
```

이 쿼리의 출력 예는 다음과 같습니다.

TABSCHEMA	TABNAME	DBPARTITIONNUM	DATA_PARTITION_ID	COMPRESS_ATTR	DICT_BUILDER	...
SYSIBM	SYSTABLES	0	0	N	NOT BUILT	...
SYSIBM	SYSTABLES	0	0	N	NOT BUILT	...
...						
SIMAP2	STAFF	0	4	Y	REORG	...
SIMAP2	STAFF	0	4	Y	REORG	...
...						

156개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속):

DICT_BUILD_TIMESTAMP	COMPRESS_DICT_SIZE	EXPAND_DICT_SIZE	ROWS_SAMPLED	...
-	0	0	0	...
-	0	0	0	...
...				
2009-03-31-11.08.18.000000	3968	3000	6	...
2009-03-31-11.08.18.000000	13312	10944	6	...
...				

이 쿼리의 출력(계속):

PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH	OBJECT_TYPE
0	0	0	DATA
0	0	0	XML
...			
70	70	31	DATA
66	66	235	XML
...			



예 2: 사전이 작성된 오브젝트, 사전 빌드 조치 및 모든 테이블의 사전 작성 시간을 판별합니다.

```
SELECT TABSCHEMA, TABNAME, DBPARTITIONNUM, DATA_PARTITION_ID,
       OBJECT_TYPE, DICT_BUILDER, DICT_BUILD_TIMESTAMP
FROM SYSIBMADM.ADMINTABCOMPRESSINFO
```

이 쿼리의 출력 예는 다음과 같습니다.

TABSCHEMA	TABNAME	DBPARTITIONNUM	DATA_PARTITION_ID	...
----	----	----	----	----
SYSIBM	SYSTABLES	0	0	...
SYSIBM	SYSTABLES	0	0	...
...				
SIMAP2	STAFF	0	4	...
SIMAP2	STAFF	0	4	...
SYSTOOLS	HMON_COLLECTION	0	0	...
SYSTOOLS	HMON_COLLECTION	0	0	...

156개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속):

OBJECT_TYPE	DICT_BUILDER	DICT_BUILD_TIMESTAMP
----	----	----
DATA	NOT BUILT	-
XML	NOT BUILT	-
...		
DATA	REORG	2009-03-31-11.08.18.000000
XML	REORG	2009-03-31-11.08.18.000000
DATA	REDISTRIBUTE	2009-03-29-06.44.32.000000
XML	REDISTRIBUTE	2009-03-29-06.44.32.000000

## ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수

ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수는

ADMINTABCOMPRESSINFO 관리 뷰와 동일한 정보를 리턴하지만 스키마, 테이블 이름 및 실행 모드를 지정할 수 있습니다.

리턴할 수 있는 전체 정보 목록은 ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수 메타데이터 테이블을 참조하십시오.

이 함수는 모든 테이블에 대해 두 행을 리턴합니다. 이 중 한 행은 OBJECT\_TYPE 컬럼의 DATA 값이며 다른 한 행의 값은 해당 컬럼의 XML 값입니다. DATA로 표시된 행은 사용되지 않는 1004 페이지의 『ADMINTABCOMPRESSINFO 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO』 테이블 함수에서 리턴된 값과 동일합니다. XML로 표시된 행은 XML 압축 사전을 설명합니다.

## 구문

►►—ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97—(—*tabschema*—,—*tablename*—,—*execmode*—)—►►

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *tabschema*

스키마 이름을 지정하는 VARCHAR  
(128) 유형의 입력 인수.

### *tablename*

테이블 이름, 구체화된 쿼리 테이블 이름이나 계층 구조 테이블 이름을 지정하는  
VARCHAR  
(128) 유형의 입력 인수.

### *execmode*

실행 모드를 지정하는 VARCHAR(30) 유형의 입력 인수. 실행 모드는 다음 중 하나  
나일 수 있습니다.

- 'REPORT' -- 마지막 생성 날짜로 압축 정보를 보고합니다. 이것은 디폴트값입니다.
- 'ESTIMATE' -- 현재 테이블을 기반으로 하는 새 압축 정보를 생성합니다.

## 권한 부여

ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 함수에 대한 EXECUTE 특권

### 예:

예 1: 테이블 SIMAP2.STAFF에 대한 기존 압축 정보를 검색합니다.

```
SELECT *
FROM TABLE(
  SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO_V97('SIMAP2', 'STAFF', 'REPORT'))
AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

TABSCHEMA	TABNAME	DBPARTITIONNUM	DATA_PARTITION_ID	COMPRESS_ATTR	DICT_BUILDER	...
SIMAP2	STAFF	0	4	Y	REORG	...
SIMAP2	STAFF	0	4	Y	NOT BUILT	...

2개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속):

DICT_BUILD_TIMESTAMP	COMPRESS_DICT_SIZE	EXPAND_DICT_SIZE	ROWS_SAMPLED	...
2009-03-31-12.19.30.000000	13312	5296	35	...
-	0	0	0	...

이 쿼리의 출력(계속):

PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH	OBJECT_TYPE
38	38		27 DATA
0	0		0 XML

예 2: 테이블 SIMAP2.STAFF에 대한 예상 압축 정보를 지금 날짜로 해서 검색합니다.

```
SELECT *
FROM TABLE(
  SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO_V97('SIMAP2', 'STAFF', 'ESTIMATE'))
AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

TABSCHEMA	TABNAME	DBPARTITIONNUM	DATA_PARTITION_ID	COMPRESS_ATTR	DICT_BUILDER	...
SIMAP2	STAFF	0		4 Y	TABLE FUNCTION	...
SIMAP2	STAFF	0		4 Y	TABLE FUNCTION	...

2개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속):

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	...

이 쿼리의 출력(계속):

PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH	OBJECT_TYPE
38	38		27 DATA
75	75		95 XML

예 3: SIMAP2스키마의 테이블에 있는 모든 오브젝트에 대한 전체 디렉토리 크기를 판별합니다.

```
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
```

이 쿼리의 출력:

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  ...

```

44개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속):

```

TOTAL_DICT_SIZE DBPARTITIONNUM DATA_PARTITION_ID
-----
0                0                0                0                0                0
1890             0                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

```

예 4: SIMAP2 스키마의 테이블 사전 정보의 보고서를 봅니다.

```

SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO_V97('SIMAP2','','REPORT'))
AS T

```

이 쿼리의 출력:

```

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     ...

```

44개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속):

```

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 ...

```

이 쿼리의 출력(계속):

```

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

```

예 5: SIMAP2 스키마에 있는 테이블의 DATA 오브젝트에 대한 사전 정보 보고서를 봅니다.

```

SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO_V97('SIMAP2','','REPORT'))
WHERE OBJECT_TYPE='DATA'

```

이 쿼리의 출력:

```

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 ...

```

22개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속):

```

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 ...

```

이 쿼리의 출력(계속):

```

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

예 6: SIMAP2 스키마에 있는 CUSTOMER 테이블의 XML 오브젝트에 대한 사전 정보 보고서를 봅니다.

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO_V97('SIMAP2', 'CUSTOMER', 'REPORT'))
WHERE OBJECT_TYPE='XML'
```

이 쿼리의 출력:

TABSHEMA	TABNAME	DBPARTITIONNUM	DATA_PARTITION_ID	COMPRESS_ATTR	DICT_BUILDER	...
SIMAP2	CUSTOMER	0	1	Y	REORG	...

이 쿼리의 출력(계속):

DICT_BUILD_TIMESTAMP	COMPRESS_DICT_SIZE	EXPAND_DICT_SIZE	ROWS_SAMPLED	...
2009-03-31-11.08.18.000000	13312	10944	6	...

이 쿼리의 출력(계속):

PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH	OBJECT_TYPE
66	66	235	XML

## 사용 시 참고사항

- *tabschema* 및 *tabname* 모두 지정된 경우 정보가 해당 특정 테이블에만 리턴됩니다.
- *tabschema*이 지정되지만 *tabname*이 비어 있거나("") 널(NULL)인 경우, 정보는 주어진 스키마의 모든 테이블에 대해 리턴됩니다.
- *tabschema*가 비어 있거나("") 널(NULL)이고 *tabname*이 지정되면 오류가 리턴됩니다. 특정 테이블에 대한 정보를 검색하려면 테이블이 스키마와 테이블 이름 모두에서 식별되어야 합니다.
- *tabschema* 및 *tabname*이 모두 비어 있거나("") 널(NULL)이면 정보는 모든 테이블에 대해 리턴됩니다.
- *tabschema* 또는 *tabname*이 존재하지 않거나 *tabname*이 테이블 이름(유형 T), 구체화된 쿼리 테이블 이름(유형 S) 또는 계층 구조 테이블 이름(유형 H)에 해당되지 않으면, 비어 있는 결과 세트가 리턴됩니다.
- ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수가 제공된 테이블에 대해 데이터를 검색하고 있는 경우, SYSTABLES의 해당 행에 대한 공유 잠금을 획득하게 되어 리턴되는 데이터의 일관성이 유지됩니다. (예를 들어, 정보가 검색되고

있는 동안 테이블이 변경되지 않도록 합니다.) 테이블 함수 호출 기간 동안이 아니라 테이블에 대한 압축 정보를 검색하는 동안만 잠금이 보유됩니다.

- 쿼리된 테이블이 비XML 테이블인 경우 XML 스토리지 오브젝트(XDA)에 리턴된 행이 있습니다.

## ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수 메타데이터

표 71. ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수 메타데이터

컬럼 이름	데이터 유형	설명
TABSCHEMA	VARCHAR (128)	스키마 이름
TABNAME	VARCHAR (128)	테이블 이름
DBPARTITIONNUM	SMALLINT	데이터베이스 파티션 번호
DATA_PARTITION_ID	INTEGER	데이터 파티션 번호
COMPRESS_ATTR	CHAR(1)	테이블에서의 COMPRESS 속성 상태는 다음 중 하나가 될 수 있습니다. <ul style="list-style-type: none"> <li>• 'Y' = 행 압축이 예로 설정됩니다.</li> <li>• 'N' = 행 압축이 아니므로 설정됩니다.</li> </ul>
DICT_BUILDER	VARCHAR (30)	사전을 빌드하는 데 사용된 코드 경로는 다음 중 하나일 수 있습니다. <ul style="list-style-type: none"> <li>• 'INSPECT' = INSPECT ROWCOMPESTIMATE</li> <li>• 'LOAD' = LOAD INSERT/REPLACE</li> <li>• 'NOT BUILT' = 사용 가능한 사전 없음</li> <li>• 'REDISTRIBUTE' = REDISTRIBUTE</li> <li>• 'REORG' = REORG RESETDICTIONARY</li> <li>• 'TABLE GROWTH' = INSERT</li> </ul>
DICT_BUILD_TIMESTAMP	TIMESTAMP	사전이 빌드된 시간소인. 시간소인은 초 단위입니다. 사용 가능한 사전이 없는 경우 시간소인은 널(NULL)입니다.
COMPRESS_DICT_SIZE	BIGINT	바이트 단위로 측정된 압축 사전 크기.
EXPAND_DICT_SIZE	BIGINT	바이트로 측정된 확장 사전 크기. 실행 기록 사전이 존재하면 이 값은 현재 및 실행 기록 사전 크기의 합입니다.
ROWS_SAMPLED	INTEGER	사전을 빌드하는 데 사용된 레코드 수. 압축 사전을 포함한 이주된 테이블은 이 컬럼에서 널(NULL)을 리턴합니다.
PAGES_SAVED_PERCENT	SMALLINT	압축으로 저장된 페이지의 백분율. 이 정보는 샘플 버퍼의 레코드 데이터만을 기반으로 합니다. 압축 사전을 포함한 이주된 테이블은 이 컬럼에서 널(NULL)을 리턴합니다.
BYTES_SAVED_PERCENT	SMALLINT	압축으로 절약되는 바이트의 백분율. 이 정보는 샘플 버퍼의 레코드 데이터만을 기반으로 합니다. 압축 사전을 포함한 이주된 테이블은 이 컬럼에서 널(NULL)을 리턴합니다.

표 71. ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
AVG_COMPRESS_REC_LENGTH	SMALLINT	사전을 빌드하는 데 사용되는 레코드의 압축된 레코드 평균 길이. 압축 사전을 포함한 이주된 테이블은 이 컬럼에서 널(NULL)을 리턴합니다.
OBJECT_TYPE	VARCHAR(4)	오브젝트의 유형. 유형에 따라 이 행에는 지정된 오브젝트에 속한 값이 포함됩니다. 값은 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• 'XML'</li> <li>• 'DATA'</li> </ul>

## ADMINTABINFO 관리 뷰 및 ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수 - 테이블 크기 및 상태 정보 검색

ADMINTABINFO 관리 뷰 및 ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수는 카탈로그 뷰에서 현재 사용 불가능한 테이블 크기와 상태 정보를 검색하는 방법을 제공합니다.

### ADMINTABINFO 관리 뷰

ADMINTABINFO 관리 뷰는 구체화된 쿼리 테이블(MQT) 및 계층 구조 테이블 및 테이블에 대한 크기와 상태 정보만을 리턴합니다. SYSCAT.TABLES 카탈로그 뷰에서 이들 테이블 유형은 테이블에 대해 T로, 구체화된 쿼리 테이블에 대해서는 S, 계층 구조 테이블에 대해서는 H로 보고됩니다. 테이블에 대한 데이터베이스 파티션 레벨 및 데이터 파티션 레벨 모두에서 정보가 리턴됩니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 전체 정보 목록은 ADMINTABINFO 관리 뷰 및 ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수 메타데이터 테이블을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- ADMINTABINFO 관리 뷰에 대한 SELECT 특권
- ADMINTABINFO 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한



예:

예 1: 모든 테이블에 대한 크기와 상태 정보를 검색합니다.

```
SELECT * FROM SYSIBMADM.ADMINTABINFO
```

예 2: 드문드문 채워진 다수의 테이블에서 사용된 실제 스페이스 크기를 판별합니다.

```
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
```

예 3: 큰 RID를 사용하는데 적절한 테이블을 식별하지만 현재 큰 RID를 사용할 수 없습니다.

```
SELECT TABSCHEMA, TABNAME FROM SYSIBMADM.ADMINTABINFO  
      WHERE LARGE_RIDS = 'P'
```

예 4: type-1 인덱스를 사용하는 테이블을 식별하고 type-2 인덱스로 변환하려면 재구성성이 필요합니다.

```
SELECT TABSCHEMA, TABNAME FROM SYSIBMADM.ADMINTABINFO  
      WHERE INDEX_TYPE = 1
```

예 5: type-1 형식의 XML 데이터를 포함하는 테이블을 식별하고 type-2 형식으로 변환하기 위하여 온라인 테이블 이동이 필요합니다.

```
SELECT TABSCHEMA, TABNAME FROM SYSIBMADM.ADMINTABINFO  
      WHERE XML_RECORD_TYPE=1
```

예 4: 테이블 T1에 대해 수집된 통계 정보의 현재 유형을 점검합니다.

```
SELECT SUBSTR(TABSCHEMA, 1, 10) AS TBSHEMA, SUBSTR(TABNAME, 1, 10)  
      AS TBNAME, STATSTYPE FROM SYSIBMADM.ADMINTABINFO WHERE TABNAME = 'T1';
```

TBSHEMA	TBNAME	STATSTYPE
DB2USER1	T1	U

1개의 레코드가 선택되었습니다.

## ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수

ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수는 ADMINTABINFO 관리 뷰와 같은 정보를 리턴하지만 스키마와 테이블 이름을 지정할 수 있습니다.

리턴될 수 있는 전체 정보 목록은 ADMINTABINFO 관리 뷰 및 ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수 메타데이터 테이블을 참조하십시오.

## 구문

```
►►—ADMIN_GET_TAB_INFO_V97—(—tabschema—,—tablename—)—————►►
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *tabschema*

스키마 이름을 지정하는 유형 VARCHAR  
(128)의 입력 인수.

### *tablename*

테이블 이름, 구체화된 쿼리 테이블 이름이나 계층 구조 테이블 이름을 지정하는 유형 VARCHAR  
(128)의 입력 인수.

## 권한 부여

ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수에서의 EXECUTE 특권.

### 예:

예 1: 테이블 DBUSER1.EMPLOYEE에 대한 크기와 상태 정보를 검색합니다.

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_INFO_V97('DBUSER1', 'EMPLOYEE'))  
AS T
```

예 2: 파티션되지 않은 테이블(DBUSER1.EMPLOYEE)이 단일 테이블 스페이스에 저장된 모든 연관된 오브젝트(예를 들어 인덱스 및 LOB)와 함께 존재한다고 가정합니다. 테이블 스페이스에서 테이블이 실제 사용하고 있는 스페이스 크기를 계산합니다.

```
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
```

새 테이블 스페이스에 동일한 페이지 크기가 있고 Extent 크기가 원래 테이블 스페이스와 같은 다른 테이블 스페이스로 테이블이 이동한 경우 필요한 스페이스 크기를 계산합니다.

```
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
```

예 3: 테이블 DBUSER1.EMPLOYEE에 대한 압축 사전의 전체 크기를 판별합니다.

```
SELECT SUBSTR(TABSCHEMA,1,10) AS TABSCHEMA, 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'))
```

예 4: 다차원적으로 클러스터된(MDC) 테이블인 SAMPLE.STAFF에서 재개 가능한 스페이스 양을 판별합니다.

```
SELECT RECLAIMABLE_SPACE FROM TABLE(SYSPROC.ADMIN_GET_TAB_INFO_V97('SAMPLE','STAFF'))
```

## 사용 시 참고사항

- 두 *tabschema* 및 *tablename*이 지정된 경우 정보는 해당 특정 테이블에만 리턴됩니다.
- *tabschema*는 지정되었지만 *tablename*이 NULL 또는 비어 있는 문자열("")인 경우에는 지정된 스키마의 모든 테이블에 대한 정보가 리턴됩니다.
- *tabschema*가 NULL 또는 비어 있는 문자열("")이고 *tablename*이 지정된 경우에는 오류가 리턴됩니다. 특정 테이블에 대한 정보를 검색하려면 테이블이 스키마와 테이블 이름 모두에서 식별되어야 합니다.
- *tabschema* 및 *tablename* 둘 다 NULL 또는 비어 있는 문자열("")인 경우 모든 테이블에 대해 정보가 리턴됩니다.
- *tabschema* 또는 *tablename*이 존재하지 않거나 *tablename*이 테이블 이름(유형 T), 구체화된 쿼리 테이블 이름(유형 S) 또는 계층 구조 테이블 이름(유형 H)에 해당되지 않으면, 비어 있는 결과 세트가 리턴됩니다.
- DMIN\_GET\_TAB\_INFO\_V97 테이블 함수가 주어진 테이블에 대해 데이터를 검색하고 있는 경우, SYSTABLES의 해당 행에 대한 공유 잠금을 획득하게 되어 리턴되는 데이터의 일관성이 유지됩니다(예를 들어 정보가 검색되고 있는 동안 테이블이 삭제되지 않도록 하기 위해). 테이블 함수 호출 기간 동안이 아니라 테이블에 대한 크기와 상태 정보를 검색하는 동안만 잠금이 보유됩니다.
- SMS 테이블 스페이스의 테이블에 대해 보고된 실제 크기는 논리적 크기와 동일합니다.
- Inplace reorg가 테이블에서 사용 중이면 데이터 오브젝트(DATA\_OBJECT\_P\_SIZE)에 대한 실제 크기가 계산되지 않습니다. 논리적 크기만이 리턴됩니다. INPLACE\_REORG\_STATUS 출력 컬럼을 보면 inplace reorg가 사용 중인지 알 수 있습니다.
- 오브젝트가 아직 재구성되지 않은 경우 DB2 UDB 버전 8이 실제 크기보다 커지기 전에 LOB 오브젝트에 대해 보고된 논리적 크기가 작성됩니다.

## REDISTRIBUTING\_PENDING

1. 주어진 테이블에 대해 실행되는 재분배 없음 N
2. 테이블이 아닌 데이터베이스 파티션 그룹에서 실행하기 위해 재분배가 시작된 N
3. 데이터 이동 전의 단계에서 재분배 실패함 N
4. 데이터 이동 단계에서 재분배 실패함 Y
5. 재분배가 완료되고 테이블이 커밋됨 N

## ADMINTABINFO 관리 뷰 및 ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수 메타데이터

표 72. ADMINTABINFO 관리 뷰 및 ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수 메타데이터

컬럼 이름	데이터 유형	설명
TABSCHEMA	VARCHAR (128)	스키마 이름.
TABNAME	VARCHAR (128)	테이블 이름.
TABTYPE	CHAR(1)	테이블 유형: <ul style="list-style-type: none"> <li>• ‘H’ = 계층 구조 테이블</li> <li>• ‘S’ = 구체화된 쿼리 테이블</li> <li>• ‘T’ = 테이블</li> </ul>
DBPARTITIONNUM	SMALLINT	데이터베이스 파티션 번호.
DATA_PARTITION_ID	INTEGER	데이터 파티션 번호.
AVAILABLE	CHAR(1)	테이블 상태: <ul style="list-style-type: none"> <li>• ‘N’ = 테이블이 사용 불가능합니다. 테이블이 사용 불가능하면 크기와 상태와 관련된 기타 모든 출력 컬럼이 널(NULL)이 됩니다.</li> <li>• ‘Y’ = 테이블이 사용 가능합니다.</li> </ul> <p>주: 복구 불가능한 로드를 통한 롤 포워드의 경우 테이블을 사용 불가능 상태로 설정합니다.</p>
DATA_OBJECT_L_SIZE	BIGINT	데이터 오브젝트 논리적 크기. 테이블에 대해 논리적으로 할당된 디스크 스페이스의 크기로 KB 단위로 보고됩니다. 논리적 크기는 테이블이 알고 있는 스페이스 크기입니다. 테이블에 대해 할당된 실제 스페이스 크기보다 작을 수 있습니다(예를 들어 논리적 테이블 절단의 경우). 다차원적으로 클러스터된(MDC) 테이블의 경우 이 크기에는 블록 맵 오브젝트의 논리적 크기가 포함됩니다. 리턴된 크기는 DMS 테이블 스페이스에서 작성된 오브젝트와 테이블에 대해 논리적으로 할당된 전체 크기로 대략의 Extent Map 페이지(EMP) extent로 간주됩니다. 이 크기는 기본 테이블의 논리적 크기만을 나타냅니다. LOB 데이터, Long 데이터, 인덱스 및 XML 오브젝트로 사용된 스페이스는 다른 컬럼에서 보고됩니다.
DATA_OBJECT_P_SIZE	BIGINT	데이터 오브젝트 실제 크기. 테이블에 대해 실제 할당된 디스크 스페이스의 크기로 KB 단위로 보고됩니다. MDC 테이블의 경우 이 크기에는 블록 맵 오브젝트의 크기가 포함됩니다. 리턴된 크기는 테이블에 대해 할당된 전체 크기로 간주되며 DMS 테이블 스페이스에서 작성된 오브젝트에 대한 EMP extent를 포함합니다. 이 크기는 기본 테이블의 실제 크기만을 나타냅니다. LOB 데이터, Long 데이터, 인덱스 및 XML 오브젝트로 사용된 스페이스는 다른 컬럼에서 보고됩니다.

표 72. ADMINTABINFO 관리 뷰 및 ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
INDEX_OBJECT_L_SIZE	BIGINT	<p>인덱스 오브젝트 논리적 크기. 테이블에서 정의된 인덱스에 대해 논리적으로 할당된 디스크 스페이스의 크기로 KB 단위로 보고됩니다. 논리적 크기는 테이블이 알고 있는 스페이스 크기입니다. 테이블의 인덱스 데이터를 보유하도록(예를 들어 논리적 테이블 절단의 경우) 할당된 실제 크기보다 작을 수 있습니다. 리턴된 크기는 DMS 테이블 스페이스에서 작성된 인덱스와 인덱스에 논리적으로 할당된 전체 크기로 대략의 EMP extent로 간주됩니다.</p> <p>파티션된 테이블의 파티션된 인덱스의 경우 DATA_PARTITION_ID 로 식별된 데이터 파티션의 인덱스 파티션이 포함된 인덱스 오브젝트의 논리적 크기입니다. 이 값은 파티션된 테이블의 파티션되지 않은 인덱스는 고려하지 않습니다. 파티션된 인덱스와 파티션되지 않은 인덱스 모두에 대한 정보가 필요한 경우에는 ADMIN_GET_INDEX_INFO 함수를 사용할 수 있습니다.</p>
INDEX_OBJECT_P_SIZE	BIGINT	<p>인덱스 오브젝트 실제 크기. 테이블에서 정의된 인덱스에 대해 할당된 디스크 스페이스의 실제 크기로 KB 단위로 보고됩니다. 리턴된 크기는 인덱스에 대해 할당된 전체 크기로 간주되며 DMS 테이블 스페이스에서 작성된 인덱스에 대한 EMP extent를 포함합니다.</p> <p>파티션된 테이블의 파티션된 인덱스의 경우 DATA_PARTITION_ID 로 식별된 데이터 파티션의 인덱스 파티션이 포함된 인덱스 오브젝트의 실제 크기입니다. 이 값은 파티션된 테이블의 파티션되지 않은 인덱스는 고려하지 않습니다. 파티션된 인덱스와 파티션되지 않은 인덱스 모두에 대한 정보가 필요한 경우에는 ADMIN_GET_INDEX_INFO 함수를 사용할 수 있습니다.</p>
LONG_OBJECT_L_SIZE	BIGINT	<p>Long 오브젝트 논리적 크기. 테이블의 long 필드 데이터에 논리적으로 할당된 디스크 스페이스 크기로 KB 단위로 보고됩니다. 논리적 크기는 테이블이 알고 있는 스페이스 크기입니다. 테이블의 long 필드 데이터를 보유하도록(예를 들어 논리적 테이블 절단의 경우) 할당된 실제 스페이스 크기보다 작을 수 있습니다. 리턴된 크기는 DMS 테이블 스페이스에서 작성된 long 필드 데이터와 long 필드 데이터에 논리적으로 할당된 전체 크기로 대략의 EMP extent로 간주됩니다.</p>
LONG_OBJECT_P_SIZE	BIGINT	<p>Long 오브젝트 실제 크기. 테이블의 long 필드 데이터에 실제 할당된 디스크 스페이스 크기로 KB 단위로 보고됩니다. 리턴된 크기는 long 필드 데이터의 전체 크기로 간주되며 DMS 테이블 스페이스에서 작성된 long 필드 데이터에 대한 EMP extent를 포함합니다.</p>
LOB_OBJECT_L_SIZE	BIGINT	<p>LOB 오브젝트 논리적 크기. 테이블의 LOB 데이터에 논리적으로 할당된 디스크 스페이스 크기로 KB 단위로 보고됩니다. 논리적 크기는 테이블이 알고 있는 스페이스 크기입니다. 테이블의 LOB 데이터를 보유하도록(예를 들어 논리적 테이블 절단의 경우) 할당된 실제 스페이스 크기보다 작을 수 있습니다. 이 크기는 LOB 할당 오브젝트에 대해 논리적으로 할당된 스페이스를 포함합니다. 리턴된 크기는 DMS 테이블 스페이스에서 작성된 LOB 데이터와 LOB 데이터에 논리적으로 할당된 전체 크기로 대략의 EMP extent로 간주됩니다.</p>

표 72. ADMIN\_TABINFO 관리 뷰 및 ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
LOB_OBJECT_P_SIZE	BIGINT	LOB 오브젝트 실제 크기. 테이블의 LOB 데이터에 실제 할당된 디스크 스페이스 크기로 KB 단위로 보고됩니다. 이 크기는 LOB 할당 오브젝트에 대해 할당된 스페이스를 포함합니다. 리턴된 크기는 LOB 데이터에 대해 할당된 전체 크기로 간주되며 DMS 테이블 스페이스에서 작성된 LOB 데이터에 대한 EMP extent를 포함합니다.
XML_OBJECT_L_SIZE	BIGINT	XML 오브젝트 논리적 크기. 테이블의 XML 데이터에 논리적으로 할당된 디스크 스페이스 크기로 KB 단위로 보고됩니다. 논리적 크기는 테이블이 알고 있는 스페이스 크기입니다. 테이블의 XML 데이터를 보유하도록(예를 들어 논리적 테이블 절단의 경우) 할당된 실제 스페이스 크기보다 작을 수 있습니다. 리턴된 크기는 DMS 테이블 스페이스에서 작성된 XML 데이터와 XML 데이터에 논리적으로 할당된 전체 크기로 대략의 EMP extent로 간주됩니다.
XML_OBJECT_P_SIZE	BIGINT	XML 오브젝트 실제 크기. 테이블의 XML 데이터에 실제 할당된 디스크 스페이스 크기로 KB 단위로 보고됩니다. 리턴된 크기는 XML 데이터에 대해 할당된 전체 크기로 간주되며 DMS 테이블 스페이스에서 작성된 XML 데이터에 대한 EMP extent를 포함합니다.
INDEX_TYPE	SMALLINT	테이블에 대해 현재 사용되는 인덱스 유형을 나타냅니다. 다음을 리턴합니다. <ul style="list-style-type: none"> <li>• type-1 인덱스가 사용 중인 경우 1</li> <li>• type-2 인덱스가 사용 중인 경우 2</li> </ul>
REORG_PENDING	CHAR(1)	'Y'는 reorg 권장 변경사항이 테이블에 적용되어 클래식(오프라인) reorg가 필요하다는 것을 나타냅니다. 그렇지 않은 경우 'N'이 리턴됩니다.
INPLACE_REORG_STATUS	VARCHAR(10)	테이블에서 inplace 테이블 재구성의 현재 상태. 상태 값은 다음 중 하나가 될 수 있습니다. <ul style="list-style-type: none"> <li>• ABORTED(PAUSED 상태의 경우 RESUME할 수 없으며 STOP이 필요합니다)</li> <li>• EXECUTING</li> <li>• NULL(inplace reorg가 테이블에서 수행되지 않은 경우)</li> <li>• PAUSED</li> </ul>
LOAD_STATUS	VARCHAR(12)	테이블에서 로드 조작의 현재 상태. 상태 값은 다음 중 하나가 될 수 있습니다. <ul style="list-style-type: none"> <li>• IN_PROGRESS</li> <li>• NULL(테이블에서 로드가 진행되지 않았고 테이블이 로드 보류 상태가 아닌 경우)</li> <li>• PENDING</li> </ul>
READ_ACCESS_ONLY	CHAR(1)	테이블이 읽기 액세스 전용 상태인 경우 'Y', 그렇지 않은 경우 'N'. 'N' 값은 테이블에 모두 액세스할 수 있음을 의미하는 것으로 해석되어서는 안 됩니다. 로드가 진행 중이거나 보류 중인 경우 'Y' 값은 테이블 데이터가 읽기 액세스로 가능하다는 것을 의미하며 'N' 값은 테이블에 액세스할 수 없음을 의미합니다. 마찬가지로 테이블 상태가 무결성 설정 보류인 경우(SYS-CAT.TABLES STATUS 컬럼 참조) 'N' 값은 테이블에 액세스할 수 없음을 의미합니다.

표 72. ADMINTABINFO 관리 뷰 및 ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
NO_LOAD_RESTART	CHAR(1)	'Y' 값은 테이블이 로드 재시작이 불가능한 부분 로드 상태를 나타냅니다. 그렇지 않은 경우 'N'이 리턴됩니다.
NUM_REORG_REC_ALTERS	SMALLINT	마지막 재조직 이후 이 테이블에 대해 수행된 reorg 권장 변경 조작 수(예를 들어 재구성이 필요한 이후의 변경 조작).
INDEXES_REQUIRE_REBUILD	CHAR(1)	파티션되지 않은 테이블의 경우, 테이블에서 정의된 인덱스를 재빌드해야 하는 경우 'Y', 그렇지 않은 경우 'N'. 파티션된 테이블의 경우, DATA_PARTITION_ID로 식별된 데이터 파티션에 대한 인덱스 파티션을 재빌드해야 하는 경우 'Y', 그렇지 않은 경우 'N'.
LARGE_RIDS	CHAR(1)	테이블이 큰 행 ID(RID)(4바이트 페이지 번호, 2바이트 슬롯 번호)를 사용하는지 여부를 나타냅니다. 'Y' 값은 테이블이 큰 RID를 사용하고 있음을 나타내며 'N'은 큰 RID를 사용하고 있지 않음을 나타냅니다. 테이블이 큰 RID를 지원하는 경우(즉, 테이블이 큰 테이블 스페이스에 있는 경우) 'P'(보류) 값이 리턴되지만 테이블에 대한 적어도 하나의 인덱스가 재구성되거나 아직 재빌드되지 않았으므로 테이블은 여전히 4바이트 RID(테이블이나 인덱스를 변환하기 위해서는 해당 조치가 취해져야 함을 의미)를 사용합니다.
LARGE_SLOTS	CHAR(1)	테이블이 큰 슬롯(페이지 당 256행 이상 허용)을 사용하고 있는지 여부를 표시합니다. 'Y' 값은 테이블이 큰 슬롯을 사용하고 있음을 나타내며 'N'은 큰 슬롯을 사용하고 있지 않음을 나타냅니다. 테이블이 큰 슬롯(즉, 테이블이 큰 테이블 스페이스에 있는 경우)을 지원하는 경우 'P'(보류) 값이 리턴되지만 아직 테이블에서 수행된 테이블 절단 조작이나 오프라인 테이블 재구성이 없으므로 여전히 페이지당 최대 255행을 사용합니다.
DICTIONARY_SIZE	BIGINT	테이블에 대한 행 압축 사전이 있는 경우에는 행 압축에 사용되는 테이블 사전 크기(바이트). 실행 기록 사전이 존재하면 이 값은 현재 및 실행 기록 사전 크기의 합입니다.
BLOCKS_PENDING_CLEANUP	BIGINT	MDC 테이블의 경우 정리가 보류 중인 블록 수. MDC 테이블이 아닌 경우 이 값은 항상 영(0)입니다.
STATSTYPE	CHAR(1)	<ul style="list-style-type: none"> <li>'F' = 테이블이나 인덱스 스캔이 없는 시스템 제조 통계. 이들 통계는 메모리에 저장되며 시스템 카탈로그에 저장된 것과 다릅니다. 이는 임시 상태이며 최종에는 전체 통계가 DB2에서 수집되고 시스템 카탈로그에 저장됩니다.</li> <li>'A' = 비동기식으로 수집된 시스템 통계. 통계는 백그라운드 프로세스에서 DB2로 자동으로 수집되며 시스템 카탈로그에 저장됩니다.</li> <li>'S' = 동기식으로 수집된 시스템 통계. 통계는 SQL문 컴파일 중에 DB2에서 자동으로 수집됩니다. 이들 통계는 메모리에 저장되며 시스템 카탈로그에 저장된 것과 다릅니다. 이는 임시 상태이며 최종에는 DB2가 시스템 카탈로그에 통계를 저장합니다.</li> <li>'U' = 사용자가 수집한 통계. 통계 수집은 RUNSTATS, CREATE INDEX, LOAD, REDISTRIBUTE와 같은 유틸리티를 통해 사용자가 시작하거나 수동으로 시스템 카탈로그 통계를 갱신하여 시작합니다.</li> <li>NULL = 알 수 없는 유형</li> </ul>



표 72. ADMINTABINFO 관리 뷰 및 ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
XML_RECORD_TYPE	SMALLINT	테이블에 대해 현재 사용되는 XML 레코드 유형을 나타냅니다. <ul style="list-style-type: none"> <li>• type-1(단일 노드) XML 레코드 형식이 사용된 경우 1입니다.</li> <li>• type-2(다중 노드) XML 레코드 형식이 사용된 경우 2입니다.</li> <li>• 테이블에 XML 컬럼이 없으면 널(null)입니다.</li> </ul>
RECLAIMABLE_SPACE	BIGINT	DMS 테이블 스페이스의 MDC 테이블의 경우 이 값은 RECLAIM 옵션과 함께 REORG 명령을 실행하여 재개할 수 있는 디스크 스페이스의 크기를 나타냅니다. 디스크 스페이스는 KB 단위로 보고됩니다. 기타 테이블의 경우 값은 영(0)입니다.
XML_DICTIONARY_SIZE	BIGINT	데이터 압축 사전이 XML 스토리지 오브젝트에 대해 존재하는 경우 데이터 압축에 사용되는 XML 사전의 크기(바이트 단위). 테이블에 XML 컬럼이 없거나 압축 사전이 작성되지 않은 경우 값은 영(0)입니다.

## ADMINTEMPCOLUMNS 관리 뷰 및 ADMIN\_GET\_TEMP\_COLUMNS 테이블 함수 - 임시 테이블의 컬럼 정보 검색

ADMINTEMPCOLUMNS 관리 뷰 및 ADMIN\_GET\_TEMP\_COLUMNS 테이블 함수는 작성된 임시 테이블 및 선언된 임시 테이블에 대한 컬럼 속성 정보를 검색하는 메소드를 제공합니다.

카탈로그 뷰에 작성된 임시 테이블 인스턴스에 대한 컬럼 속성 정보가 포함되기는 하지만 선언된 임시 테이블에 대해서는 해당 정보를 포함하지 않습니다.

### ADMINTEMPCOLUMNS 관리 뷰

ADMINTEMPCOLUMNS 관리 뷰는 작성된 임시 테이블 및 선언된 임시 테이블의 인스턴스에 대한 컬럼 속성 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 완전한 정보 목록은 ADMINTEMPCOLUMNS 관리 뷰 및 ADMIN\_GET\_TEMP\_COLUMNS 테이블 함수 메타데이터 테이블을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- ADMINTEMPCOLUMNS 관리 뷰에 대한 SELECT 특권
- ADMINTEMPCOLUMNS 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.



- ADMIN\_GET\_TEMP\_COLUMNS 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

예:

예 1: 작성된 임시 테이블의 모든 인스턴스 및 현재 데이터베이스에 존재하는 모든 선언된 임시 테이블에 대한 컬럼 속성 정보를 검색하십시오.

```
SELECT * FROM SYSIBMADM.ADMINTEMPCOLUMNS
```

예 2: 데이터베이스의 임시 테이블 활성이 사용자 정의 데이터 유형 USMONEY를 사용 중인지 판별하십시오.

```
SELECT APPLICATION_HANDLE, TABSCHEMA, TABNAME
FROM SYSIBMADM.ADMINTEMPCOLUMNS
WHERE TYPENAME = 'USMONEY'
```

예 3: SYSTEM\_USER로 선언된 모든 선언된 임시 테이블에 대한 테이블 스키마, 테이블 이름 및 컬럼 이름을 검색하십시오.

```
SELECT T.TABSCHEMA, T.TABNAME, C.COLNAME
FROM SYSIBMADM.ADMINTEMPCOLUMNS C, SYSIBMADM.ADMINTEMPTABLES T
WHERE T.TEMPABTYPE = 'D'
AND T.INSTANTIATOR = SYSTEM_USER
AND T.TABSCHEMA = C.TABSCHEMA
AND T.TABNAME = C.TABNAME
```

## ADMIN\_GET\_TEMP\_COLUMNS 테이블 함수

ADMIN\_GET\_TEMP\_TABLES 테이블 함수는 ADMINTEMPCOLUMNS 관리 뷰와 동일한 정보를 리턴하나, 스키마 이름 및 테이블 이름은 사용자가 지정할 수 있습니다.

리턴될 수 있는 완전한 정보 목록은 ADMINTEMPCOLUMNS 관리 뷰 및 ADMIN\_GET\_TEMP\_COLUMNS 테이블 함수 메타데이터 테이블을 참조하십시오.

### 구문

```
►►—ADMIN_GET_TEMP_COLUMNS—(—application_handle—,—tabschema—,—tablename—)——►►
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *application\_handle*

응용프로그램 핸들을 지정하는 BIGINT 유형의 입력 인수. *application\_handle*이 지정되면, 데이터는 지정된 연결에만 리턴되며, *application\_handle*이 NULL이면, 데이터는 모든 연결에 리턴됩니다.

*tabschema*

스키마 이름을 지정하는 VARCHAR  
(128) 유형의 입력 인수.

*tablename*

작성 또는 선언된 임시 테이블 이름을 지정하는 VARCHAR  
(128) 유형의 입력 인수.

## 권한 부여

ADMIN\_GET\_TEMP\_COLUMNS 테이블 함수에 대한 EXECUTE 특권

### 예:

예 1: 현재 연결의 선언된 임시 테이블 TEMPEMLOYEE에 대한 컬럼 정보를 검색하십시오.

```
SELECT *  
FROM TABLE(  
    SYSPROC.ADMIN_GET_TEMP_COLUMNS(  
        APPLICATION_ID(), 'SESSION', 'TEMPEMLOYEE'))  
AS T
```

## 사용 시 참고사항

- *tabschema* 및 *tablename* 모두가 지정되면 해당 특정 임시 테이블에 대해서만 정보가 리턴됩니다.
- *tabschema*는 지정되었지만 *tablename*은 NULL이나 비어 있는 문자열("")인 경우 지정된 스키마의 모든 테이블에 대해 정보가 리턴됩니다.
- *tabschema*가 NULL이나 비어 있는 문자열("")이고 *tablename*이 지정된 경우에는 오류가 리턴됩니다. 특정 임시 테이블에 대한 정보를 검색하려면, 스키마와 테이블 이름 모두로 테이블을 식별해야 합니다.
- *tabschema* 및 *tablename* 모두가 NULL이나 비어 있는 문자열("")인 경우 *application\_handle*의 값에 따라 특정 연결이나 모든 연결에 대한 모든 임시 테이블에 대해 정보가 리턴됩니다.
- *tabschema* 또는 *tablename*이 없거나 또는 *tablename*이 임시 테이블 이름에 대응하지 않거나 식별된 임시 테이블의 인스턴스가 데이터베이스에 없는 경우 비어 있는 결과 세트가 리턴됩니다.

## ADMINTEMPCOLUMNS                    관리                    뷰                    및 ADMIN\_GET\_TEMP\_COLUMNS 테이블 함수 메타데이터

표 73. ADMINTEMPCOLUMNS 관리 뷰 및 ADMIN\_GET\_TEMP\_COLUMNS 테이블 함수 메타데이터

컬럼 이름	데이터 유형	설명
APPLICATION_HANDLE	BIGINT	응용프로그램에 대한 system-wide 고유 ID. 단일 파티션 데이터베이스에서 이 ID는 16비트 카운터로 구성됩니다. 다중 파티션 데이터베이스에서 이 ID는 16비트 카운터로 병합된 조정된 파티션 번호로 구성됩니다. 또한 이 식별자는 응용프로그램의 보조 연결이 가능한 모든 파티션에서 동일합니다.
APPLICATION_NAME	VARCHAR (256)	응용프로그램의 이름.
TABSCHEMA	VARCHAR (128)	컬럼이 있는 임시 테이블의 스키마 이름.
TABNAME	VARCHAR (128)	컬럼이 있는 임시 테이블의 테이블 이름.
COLNAME	VARCHAR (128)	컬럼 이름
COLNO	SMALLINT	테이블에서 이 컬럼의 번호(0에서 시작).
TYPESCHEMA	VARCHAR (128)	컬럼에 대한 데이터 유형의 스키마 이름.
TYPENAME	VARCHAR (128)	컬럼에 대한 데이터 유형의 규정되지 않은 이름.
LENGTH	INTEGER	데이터의 최대 길이. 구별 유형에 대해 0. LENGTH 컬럼은 DECIMAL 필드의 경우 정밀도를 표시하고, 10진수 부동 소수점 컬럼에 필요한 스토리지의 바이트 수(DECFLOAT(16) 및 DECFLOAT(34)의 경우 각각 8과 16)를 표시합니다.
SCALE	SMALLINT	컬럼 유형이 DECIMAL인 경우 스케일, 컬럼 유형이 TIMESTAMP인 경우 분수 초의 자릿수, 그렇지 않으면 0입니다.
DEFAULT	VARCHAR(254)	테이블의 컬럼에 대한 디폴트값은 컬럼의 데이터 유형에 적합한 상수, 특수 레지스터 또는 캐스트 함수로 표현됩니다. 키워드 NULL일 수도 있습니다. 값은 디폴트값으로 지정된 값에서 변환될 수 있습니다. 예를 들어 날짜 및 시간 상수는 ISO 형식으로 표시되고, 캐스트 함수 이름은 스키마 이름으로 규정되며, ID는 구분됩니다. DEFAULT절이 지정되지 않았거나 컬럼이 뷰 컬럼인 경우 널(NULL) 값입니다.
NULLS	CHAR(1)	컬럼의 널(null) 값 허용 속성. <ul style="list-style-type: none"> <li>• 『Y』 = 컬럼에 널(NULL) 입력 가능</li> <li>• 『N』 = 컬럼에 널(NULL) 입력 불가능</li> </ul> 표현식 또는 함수로부터 파생되는 뷰 컬럼의 경우 값이 『N』일 수 있습니다. 그러나 그러한 컬럼은 뷰를 사용하는 명령문이 산술 오류에 대한 경고와 함께 처리될 때 널(NULL) 값을 허용합니다.
CODEPAGE	SMALLINT	이 컬럼의 데이터에 사용되는 코드 페이지이며, 컬럼이 FOR BIT DATA로 정의되거나 문자열 유형이 아닌 경우 0입니다.

표 73. ADMINTEMPCOLUMNS 관리 뷰 및 ADMIN\_GET\_TEMP\_COLUMNS 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
LOGGED	CHAR(1)	유형이 LOB이거나 LOB에 기반한 구별인 컬럼에만 적용됩니다. 그렇지 않으면 공백입니다. <ul style="list-style-type: none"> <li>『Y』 = 컬럼이 로그됨</li> <li>『N』 = 컬럼이 로그되지 않음</li> </ul>
COMPACT	CHAR(1)	유형이 LOB이거나 LOB에 기반한 구별인 컬럼에만 적용됩니다. 그렇지 않으면 공백입니다. <ul style="list-style-type: none"> <li>『Y』 = 컬럼이 스토리지에서 간략화됨</li> <li>『N』 = 컬럼이 간략화되지 않음</li> </ul>
INLINE_LENGTH	INTEGER	기본 테이블에 저장할 수 있는 XML 문서 또는 구조화된 유형의 인스턴스의 내부 표현의 최대 크기(바이트)이며, 적용할 수 없는 경우 0입니다.
IDENTITY	CHAR(1)	<ul style="list-style-type: none"> <li>『Y』 = ID 컬럼</li> <li>『N』 = ID 컬럼이 아님</li> </ul>
GENERATED	CHAR(1)	생성된 컬럼의 유형. <ul style="list-style-type: none"> <li>『A』 = 컬럼 값이 항상 생성됨</li> <li>『D』 = 컬럼 값이 디폴트로 생성됨</li> <li>공백 = 컬럼이 생성되지 않음</li> </ul>

## ADMINTEMPTABLES 관리 뷰 및 ADMIN\_GET\_TEMP\_TABLES 테이블 함수 - 임시 테이블에 대한 정보 검색

ADMINTEMPTABLES 관리 뷰 및 ADMIN\_GET\_TEMP\_TABLES 테이블 함수는 작성된 임시 테이블 및 선언된 임시 테이블 인스턴스에 대한 테이블 속성 및 인스턴스화 시간 정보를 검색하는 방법을 제공합니다.

카탈로그 뷰에 작성된 임시 테이블에 대한 테이블 속성 정보가 포함되어도 선언된 임시 테이블에 대한 테이블 속성 정보는 포함되지 않습니다. 또한 카탈로그 뷰는 작성된 임시 테이블 또는 선언된 임시 테이블에 대한 테이블 인스턴스화 시간 정보를 포함하지 않습니다.

### ADMINTEMPTABLES 관리 뷰

ADMINTEMPTABLES 관리 뷰는 작성 및 선언된 임시 테이블의 인스턴스에 대한 테이블 속성 및 인스턴스화 시간 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 완전한 정보 목록은 ADMINTEMPTABLES 관리 뷰 및 ADMIN\_GET\_TEMP\_TABLES 테이블 함수 메타데이터 테이블을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- ADMINTEMPTABLES 관리 뷰에 대한 SELECT 특권
- ADMINTEMPTABLES 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- ADMIN\_GET\_TEMP\_TABLES 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

### 예:

예 1: 작성된 임시 테이블의 모든 인스턴스 및 현재 데이터베이스에 존재하는 선언된 임시 테이블에 대한 테이블 속성 및 인스턴스화 시간 정보를 검색하십시오.

```
SELECT * FROM SYSIBMADM.ADMINTEMPTABLES
```

예 2: 연결에 작성된 임시 테이블의 인스턴스가 있는지 판별하십시오.

```
SELECT APPLICATION_HANDLE, TABSCHEMA, TABNAME  
FROM SYSIBMADM.ADMINTEMPTABLES  
WHERE TEMPTABTYPE = 'C'
```

예 3: 데이터베이스에 연결된 사용자가 인스턴스화한 테이블에 대해 선언된 모든 선언된 임시 테이블에 대한 테이블 속성 및 인스턴스화 시간 정보를 검색하십시오.

```
SELECT TABSCHEMA, TABNAME, ONCOMMIT, ONROLLBACK,  
INSTANTIATION_TIME FROM SYSIBMADM.ADMINTEMPTABLES  
WHERE TEMPTABTYPE = 'D' AND INSTANTIATOR = SYSTEM_USER
```

## ADMIN\_GET\_TEMP\_TABLES 테이블 함수

ADMIN\_GET\_TEMP\_TABLES 테이블 함수는 ADMINTABINFO 관리 뷰와 동일한 정보를 리턴하나, 스키마 이름 및 테이블 이름은 사용자가 지정할 수 있습니다.

리턴될 수 있는 완전한 정보 목록은 ADMINTABINFO 관리 뷰 및 ADMIN\_GET\_TEMP\_TABLES 테이블 함수 메타데이터 테이블을 참조하십시오.

### 구문

```
►►—ADMIN_GET_TEMP_TABLES—(—application_handle—,—tabschema—,—tablename—)————►►
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *application\_handle*

응용프로그램 핸들을 지정하는 BIGINT 유형의 입력 인수. *application\_handle*이 지정되면, 데이터는 지정된 연결에만 리턴되며, *application\_handle*이 NULL이면, 데이터는 모든 연결에 리턴됩니다.

### *tabschema*

스키마 이름을 지정하는 VARCHAR (128) 유형의 입력 인수.

### *tablename*

작성 또는 선언된 임시 테이블 이름을 지정하는 VARCHAR (128) 유형의 입력 인수.

## 권한 부여

ADMIN\_GET\_TEMP\_TABLES 테이블 함수에 대한 EXECUTE 특권.

### 예:

예 1: 모든 연결에 대한 작성된 임시 테이블 DBUSER1.EMPLOYEE의 모든 인스턴스에 대한 테이블 속성 및 인스턴스화 시간 정보를 검색하십시오.

```
SELECT TABSCHEMA, TABNAME, ONCOMMIT, ONROLLBACK, INSTANTIATION_TIME
FROM TABLE(SYSPROC.ADMIN_GET_TEMP_TABLES(NULL, 'DBUSER1', 'EMPLOYEE'))
AS T
```

예 2: 현재 연결에 대한 사용자 임시 테이블의 모든 인스턴스에 대한 인스턴스화 시간 및 테이블 스페이스 ID를 검색하십시오.

```
SELECT TABSCHEMA, TABNAME, INSTANTIATION_TIME, TBSP_ID
FROM TABLE(SYSPROC.ADMIN_GET_TEMP_TABLES(APPLICATION_ID(), '', ''))
AS T
```

## 사용 시 참고사항

- 
- *tabschema* 및 *tablename* 모두가 지정되면 해당 특정 임시 테이블에 대해서만 정보가 리턴됩니다.
- *tabschema*는 지정되었지만 *tablename*은 NULL이나 비어 있는 문자열("")인 경우 지정된 스키마의 모든 테이블에 대해 정보가 리턴됩니다.
- *tabschema*가 NULL이나 비어 있는 문자열("")이고 *tablename*이 지정된 경우에는 오류가 리턴됩니다. 특정 임시 테이블에 대한 정보를 검색하려면, 스키마와 테이블 이름 모두로 테이블을 식별해야 합니다.
- *tabschema* 및 *tablename* 모두가 NULL이나 비어 있는 문자열("")인 경우 *application\_handle*의 값에 따라 특정 연결이나 모든 연결에 대한 모든 임시 테이블에 대해 정보가 리턴됩니다.

- *tabschema* 또는 *tablename*이 없거나 또는 *tablename*이 임시 테이블 이름에 대응하지 않거나 식별된 임시 테이블의 인스턴스가 데이터베이스에 없는 경우 비어 있는 결과 세트가 리턴됩니다.

## ADMINTEMPFILES 관리 뷰 및 ADMIN\_GET\_TEMP\_FILES 테이블 함수 메타데이터

표 74. ADMINTEMPFILES 관리 뷰 및 ADMIN\_GET\_TEMP\_FILES 테이블 함수 메타데이터

컬럼 이름	데이터 유형	설명
APPLICATION_HANDLE	BIGINT	응용프로그램에 대한 system-wide 고유 ID. 단일 파티션 데이터베이스에서 이 ID는 16비트 카운터로 구성됩니다. 다중 파티션 데이터베이스에서 이 ID는 16비트 카운터로 병합된 조정된 파티션 번호로 구성됩니다. 또한 이 식별자는 응용프로그램의 보조 연결이 가능한 모든 파티션에서 동일합니다.
APPLICATION_NAME	VARCHAR (256)	응용프로그램의 이름.
TABSCHEMA	VARCHAR (128)	스키마 이름.
TABNAME	VARCHAR (128)	테이블 이름.
INSTANTIATOR	VARCHAR (128)	작성된 임시 테이블이 인스턴스화되거나 선언된 임시 테이블이 선언된 권한 부여 ID.
INSTANTIATOR_TYPE	CHAR(1)	<ul style="list-style-type: none"> <li>• 『U』 = 인스턴스 작성자가 개별적 사용자임</li> </ul>
TEMPFILETYPE	CHAR(1)	<p>임시 테이블 유형은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• 『C』 = 작성된 임시 테이블</li> <li>• 『D』 = 선언된 임시 테이블</li> </ul>
INSTANTIATION_TIME	TIMESTAMP	작성된 임시 테이블 인스턴스가 인스턴스화되거나 선언된 임시 테이블이 선언되었을 때 시간.
COLCOUNT	SMALLINT	상속된 컬럼(있는 경우)을 포함한 컬럼 수.
TAB_FILE_ID	BIGINT	table_file_id - 테이블에 대한 파일 ID(FID).
TBSP_ID	BIGINT	tablespace_id - 현재 데이터베이스에서 사용된 테이블 스페이스를 고유하게 나타내는 정수
PMAP_ID	SMALLINT	현재 이 테이블에서 사용된 분산 맵에 대한 ID.
PARTITION_MODE	CHAR(1)	<p>데이터가 파티션된 데이터베이스 시스템의 데이터베이스 파티션 사이에서 분배되는 방법을 나타냅니다.</p> <ul style="list-style-type: none"> <li>• 『H』 = 해싱</li> <li>• 공백 = 데이터베이스 파티셔닝 없음</li> </ul>
CODEPAGE	SMALLINT	오브젝트의 코드 페이지입니다. 모든 문자 컬럼 및 표현식 생성 컬럼에 사용되는 디폴트 코드 페이지입니다.
ONCOMMIT	CHAR(1)	<p>COMMIT 조작 수행 시 이 테이블에서 취할 조치를 지정합니다.</p> <ul style="list-style-type: none"> <li>• 『D』 = 행 삭제</li> <li>• 『P』 = 행 보존</li> </ul>

표 74. ADMINTEMPFILES 관리 뷰 및  
ADMIN\_GET\_TEMP\_TABLES 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
ONROLLBACK	CHAR(1)	ROLLBACK 조작 수행 시 이 테이블에서 취할 조치를 지정합니다. <ul style="list-style-type: none"> <li>• 『D』 = 행 삭제</li> <li>• 『P』 = 행 보존</li> </ul>
LOGGED	CHAR(1)	이 테이블이 로그되는지 여부를 지정합니다. <ul style="list-style-type: none"> <li>• 『N』 = 로그되지 않음</li> <li>• 『Y』 = 로그됨</li> </ul>



---

## 제 5 장 관리 태스크 스케줄러 루틴 및 뷰

---

### ADMIN\_TASK\_ADD 프로시저 - 새 태스크 스케줄

ADMIN\_TASK\_ADD 프로시저는 프로시저 내부에 캡슐화될 수 있는 모든 작업인 관리 태스크를 스케줄합니다.

#### 구문

```
▶—ADMIN_TASK_ADD—(—name—,—begin_timestamp—,—end_timestamp—,——————▶  
▶—max_invocations—,—schedule—,—procedure_schema—,—procedure_name—,—————▶  
▶—procedure_input—,—options—,—remarks—)——————▶▶
```

스키마는 SYSPROC입니다.

#### 프로시저 매개변수

##### *name*

태스크 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. 이 인수는 NULL 이 될 수 없습니다.

##### *begin\_timestamp*

태스크 실행이 시작될 수 있는 가장 빠른 시간을 지정하는 TIMESTAMP 유형의 입력 인수입니다. 이 인수 값은 지나간 과거 시간일 수 없으며 *end\_timestamp* 이 후 시간일 수 없습니다.

태스크 실행 시작 시점은 이 인수 및 *schedule* 인수 정의에 따라 달라집니다.

- *begin\_timestamp* 인수가 NULL이 아닌 경우,
  - *schedule* 인수가 NULL인 경우 태스크 실행은 *begin\_timestamp*에 시작됩니다.
  - *schedule* 인수가 NULL이 아닌 경우 태스크 실행은 *begin\_timestamp* 시점 또는 이후의 다음 스케줄된 시간에 시작됩니다.
- *begin\_timestamp* 인수가 NULL인 경우,
  - *schedule* 인수가 NULL인 경우 태스크 실행은 즉시 시작됩니다.
  - *schedule* 인수가 NULL이 아닌 경우 태스크 실행은 다음 스케줄 시간에 시작됩니다.

##### *end\_timestamp*

태스크 실행이 시작될 수 있는 마지막 시간을 지정하는 TIMESTAMP 유형의 입

력 인수입니다. 이 인수 값은 지나간 과거 시간일 수 없으며 *begin\_timestamp* 이전 시간일 수 없습니다. 인수가 NULL인 경우 태스크는 스케줄된 대로 무제한 지속될 수 있습니다.

태스크 실행은 해당 *end\_timestamp* 시에도 인터럽트되지 않습니다.

#### *max\_invocations*

태스크에 허용되는 최대 실행 수를 지정하는 INTEGER 유형의 입력 인수입니다. 인수가 NULL인 경우 태스크 실행 가능 횟수에 제한이 없습니다. 인수가 0인 경우 태스크는 실행되지 않습니다.

*schedule*이 NULL이 아닌 경우 이 값이 스케줄에 적용됩니다.

*end\_timestamp* 및 *max\_invocations* 모두 지정되는 경우 *end\_timestamp*에 우선권이 있습니다. 즉, *end\_timestamp* 시간소인에 도달하면 태스크 실행 수가 아직 *max\_invocations* 값에 도달하지 않아도 태스크는 다시 실행되지 않습니다.

#### *schedule*

시간 내에서 고정된 시점에 태스크가 실행되도록 지정하는 VARCHAR(1024) 유형의 입력 인수입니다. 인수가 NULL인 경우 시간 내의 고정된 시점에 태스크가 스케줄되지 않습니다.

*schedule* 문자열은 UNIX 크론 형식으로 지정해야 합니다.

스케줄은 여러 개 작성할 수 없습니다.

#### *procedure\_schema*

이 태스크가 실행되는 프로시저의 스키마를 지정하는 VARCHAR(128) 유형의 입력 인수입니다. 이 인수는 NULL이 될 수 없습니다.

#### *procedure\_name*

이 태스크가 실행되는 프로시저 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. 이 인수는 NULL이 될 수 없습니다.

#### *procedure\_input*

이 태스크가 실행되는 프로시저의 입력 인수를 지정하는 CLOB(2M) 유형의 입력 인수입니다. 이 인수는 한 개의 데이터 행을 리턴하는 SQL문을 포함해야 합니다. 리턴된 값은 프로시저에 인수로 전달됩니다. 이 인수가 NULL인 경우 프로시저에 인수가 전달되지 않습니다.

SQL문에서 리턴되는 컬럼 수는 프로시저의 전체 인수 수(및 유형)와 일치해야 하며 한 개의 행을 포함해야 합니다. 출력 인수의 경우 값 자체는 무시되지만 프로시저에서 필요한 SQL 데이터 유형과 동일한 유형이어야 합니다.

이 SQL문은 태스크가 실행될 때마다 실행됩니다. SQL문이 실패하면 태스크 상태가 NOTRUN으로 설정되고 특정 SQLCODE 정보가 기록됩니다. 명령문이 한 개의 결과 세트가 리턴하지 않거나, 한 개의 행을 리턴하지 않거나, 여러 개의 행이나 결

과 세트를 리턴하면 태스크가 실행되지 않습니다. 태스크 상태는 NOTRUN으로 설정되고 SQLCODE SQL1465N은 이 인수가 유효하지 않은 것으로 표시하도록 설정됩니다.

명령문 결과에 직렬화된 XML 매개변수가 포함된 경우 결합된 모든 XML 매개변수의 전체 크기는 256KB로 제한됩니다. 결과가 이 임계값을 초과하면 태스크 상태가 NOTRUN으로 설정됩니다. 데이터 절단이 발생한 것을 표시하기 위해 SQLCODE -302 및 SQLSTATE 22001로 설정됩니다.

SYSTOOL.ADMIN\_TASK\_STATUS 뷰를 사용하여 태스크 상태를 확인하십시오.

#### *options*

VARCHAR(512) 유형의 입력 인수입니다. 인수는 NULL이어야 합니다.

#### *remarks*

태스크 설명을 지정하는 VARCHAR(254) 유형의 입력 인수입니다. 이 인수는 선택적이며 NULL이 될 수 있습니다.

## 권한 부여

ADMIN\_TASK\_ADD 프로시저에 대한 EXECUTE 특권. **RESTRICTIVE** 옵션을 사용하여 데이터베이스를 작성한 경우를 제외하고는 디폴트로 PUBLIC에 EXECUTE 특권이 부여됩니다.

## 사용 시 참고사항

ADMIN\_TASK\_ADD 프로시저를 호출하기 전에 SYSTOOLSPACE 테이블 스페이스가 반드시 있어야 합니다. 없는 경우에는 프로시저가 SQL0204N 오류 메시지를 리턴합니다.

태스크가 스케줄된 경우 현재 세션 사용자의 권한 부여 ID가 기록됩니다. 스케줄러는 태스크 실행 시에 이 세션 권한 부여 ID로 전환합니다.

관리 태스크 스케줄러는 지정된 사용자 ID 및 암호를 사용하지 않는 데이터베이스 연결을 수행하는 프로시저 실행은 지원하지 않습니다. 예를 들어 ADMIN\_CMD 프로시저를 사용하여 데이터베이스에서 LOAD를 수행할 수 있습니다. 소스 데이터베이스는 현재 연결된 데이터베이스에 대해 입력된 사용자 ID와 암호를 사용하여 연결됩니다. 이 유형의 LOAD 조작은 태스크 스케줄러에서 실행할 수 없습니다.

유효하지 않은 인수가 프로시저에 전달되면 SQL0171N이 리턴됩니다. 메시지 토큰은 유효하지 않은 인수 및 프로시저 이름을 표시합니다.

작업 단위(UOW)가 커밋되고 스케줄러가 태스크 정의를 폐치하지 않으면 태스크 실행을 스케줄할 수 없습니다.

매 5분마다 스케줄러는 새 태스크 또는 갱신된 태스크를 점검합니다. 태스크가 *begin\_timestamp*에 정의된 대로 가장 빠른 시작 시간에 예상대로 시작되도록 하려면 *end\_timestamp* 및 *schedule* 인수는 작업 단위(UOW) 커밋 후에 최소 5분 후로 설정되어야 합니다.

스케줄러가 태스크를 실행할 수 있도록 하려면 모든 데이터베이스 파티션에서 데이터베이스가 활성화되어 있어야 합니다.

파티션된 데이터베이스 환경에서, ADMIN\_TASK\_ADD 프로시저는 모든 데이터베이스 파티션에서 호출할 수 있습니다. 그렇지만 스케줄러는 카탈로그 데이터베이스 파티션에서 모든 태스크를 실행합니다.

*begin\_timestamp*, *end\_timestamp* 및 *schedule*은 서버의 시간대를 사용합니다. 일광절약시간(DST) 전환 시기에는 태스크 스케줄 시에 특히 주의해야 합니다. 태스크가 오전 2:01에 실행되도록 스케줄되어 있고 1시간을 앞으로 당기는 시기가 되면 시계가 오전 2:00에서 오전 3:00로 건너뛰기 때문에 태스크가 실행되지 않습니다. 반면에 시간을 다시 원래 시간으로 돌려야될 때가 되면 원래 오전 2:00에서 3:00 사이에 스케줄되었던 태스크는 2번 실행됩니다. 원하는 대로 작동할 수 있도록 사용자가 직접 일광절약시간에는 적절하게 조정해야 합니다.

스케줄러는 항상 *procedure\_schema* 및 *procedure\_name*에 지정된 프로시저를 호출한 후에 커밋합니다. 트랜잭션 롤백이 필요한 경우 롤백은 프로시저 내에서 수행되어야 합니다.

태스크 이름이 고유하지 않으면 프로시저가 SQL0601N으로 실패합니다.

## 예

예 1: 2008년 2월 4일 오전 12:00시에 매일 온라인 TSM 백업이 수행되도록 태스크를 작성합니다.

```
CALL SYSPROC.ADMIN_TASK_ADD
( 'DAILY TSM BACKUP',
  '2007-02-04-00.00.00.000000',
  NULL,
  NULL,
  '0 0 * * *',
  'SYSPROC',
  'ADMIN_CMD',
  'VALUES(''BACKUP DATABASE SALES ONLINE USE TSM WITHOUT PROMPTING'')',
  NULL,
  NULL )
```

예 2: 매 시에 이벤트 모니터를 플래시하도록 태스크를 스케줄합니다.

1. 이벤트 모니터 "em"을 플래시하는 SQL 프로시저를 PROD 스키마에 작성합니다.

```

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

```

주: 프로시저에서 FLUSH EVENT MONITOR SQL문을 직접 호출할 수는 없습니다. 그렇지만 EXECUTE IMMEDIATE는 사용할 수 있습니다.

2. ADMIN\_TASK\_ADD를 호출하여 태스크를 스케줄합니다.

```

CALL SYSPROC.ADMIN_TASK_ADD
('FLUSH_EVENT_MONITOR EVERY HOUR',
 NULL,
 NULL,
 NULL,
 '0 0-23 * * *',
 'PROD',
 'FLUSH_EVENT_MONITOR',
 NULL,
 NULL,
 NULL )

```

## UNIX 크론 형식

UNIX 크론 형식을 사용하여 ADMIN\_TASK\_ADD 및 ADMIN\_TASK\_UPDATE 프로시저의 *schedule* 매개변수에 시간을 지정합니다.

크론 형식에는 하나 이상의 공백으로 구분된 5개의 시간 및 날짜 필드가 있습니다. 필드 값에는 공백이 있을 수 없습니다. 스케줄된 태스크는 *minute*, *hour* 및 *month of year* 필드가 현재 시간 및 날짜에 일치하고 두 개 필드(>*day of month* 또는 *day of week*) 중 하나 이상이 현재 날짜에 일치하면 실행됩니다.

테이블 1에는 시간 및 날짜 필드 및 허용된 값이 크론 형식으로 나열됩니다.

표 75. UNIX 크론 형식의 필드 이름 및 값

필드 이름	허용된 값
분	0-59
시	0-23
일	1-31
월	<ul style="list-style-type: none"> <li>1-12(여기서 1은 1월이고 2는 2월입니다).</li> <li>월의 영어 이름을 기반으로 대문자, 소문자, 대소문자 혼용으로 되어 있는 세 자의 문자열. 예: jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, 또는 dec.</li> </ul>

표 75. UNIX 크론 형식의 필드 이름 및 값 (계속)

필드 이름	허용된 값
요일	<ul style="list-style-type: none"> <li>• 0-7(여기서 0 또는 7은 일요일이고 1은 월요일 등입니다).</li> <li>• 요일의 영어 이름을 기반으로 대문자, 소문자 또는 대소문자 혼용으로 되어 있는 세 자의 문자열 (예: mon, tue, wed, thu, fri, sat 또는 sun).</li> </ul>

## 범위 및 목록

숫자 범위도 허용됩니다. 범위는 하이픈으로 구분된 두 자리의 숫자입니다. 지정된 범위는 포함됩니다. 예를 들어 시 항목에 대한 범위 8-11은 8, 9, 10, 11시에 실행되도록 지정합니다.

목록도 허용됩니다. 목록은 쉼표로 구분된 숫자 세트 또는 범위입니다. 예를 들어, 다음과 같습니다.

1,2,5,9

0-4,8-12

## 무제한의 범위

필드에는 필드의 가능한 모든 값을 나타내는 별표(\*)를 사용할 수 있습니다.

명령 실행 일은 *day of month* 및 *day of week*의 두 필드로 지정할 수 있습니다. 두 필드 모두 별표 이외의 값을 사용하도록 제한된 경우 두 필드 중 하나가 현재 시간에 일치하게 되면 명령이 실행됩니다. 예를 들어 값이 30 4 1,15 \* 5인 경우 명령은 각 달의 1일과 15일 및 매 금요일 오전 4:30에 실행됩니다.

## 단계 값

단계 값은 범위와 같이 사용할 수 있습니다. 구문 *range/step*은 범위 및 실행 간격을 정의합니다.

*first-last/step*을 지정하면 *first*에 실행되고 *step*으로 *first*에서 *last*까지의 모든 후속 값에서 실행됩니다.

예를 들어 매 2시간 마다 명령이 실행되도록 지정하려면 0-23/2를 사용하십시오. 이 표현식은 0,2,4,6,8,10,12,14,16,18,20,22 값과 동등합니다.

*\*/step*을 지정하면 무제한 범위로 *step*의 모든 간격에서 실행됩니다. 예를 들어, 매 2시간 마다 실행되도록 하는 0-23/2의 대체 방법으로 *\*/2*를 사용하십시오.

## 예

테이블 2에는 다양한 스케줄링 시나리오가 가능하도록 ADMIN\_TASK\_ADD 또는 ADMIN\_TASK\_UPDATE 프로시저에서 *schedule* 인수에 사용할 수 있는 값이 나열됩니다.

표 76. 예제 태스크 스케줄 및 해당 *schedule* 인수 값

원하는 태스크 스케줄	<i>schedule</i> 값
매 월요일 오후 2:10	10 14 * * 1
매일 자정	0 0 * * *
주 중의 매일 자정	0 0 * * 1-5
매월 1일 및 15일 자정	0 0 1,15 * *
매년 11월의 17일, 21일 및 29일 및 11월의 월요일 및 수요일의 오후 6:32	32 18 17,21,29 11 mon,wed

## ADMIN\_TASK\_LIST 관리 뷰 - 스케줄러에서 태스크 정보 검색

ADMIN\_TASK\_LIST 관리 뷰는 관리 태스크 스케줄러에 정의된 각 태스크에 대한 정보를 검색합니다.

스키마는 SYSTOOLS입니다.

이 뷰는 ADMIN\_TASK\_ADD 프로시저를 처음 호출할 때 작성됩니다.

### 권한 부여

ADMIN\_TASK\_LIST 관리 뷰의 SELECT 또는 CONTROL 특권. **RESTRICTIVE** 옵션을 사용하여 데이터베이스를 작성한 경우를 제외하고는 디폴트로 PUBLIC에 SELECT 특권이 부여됩니다.

ADMIN\_TASK\_LIST 뷰를 조회하면 사용자의 세션 권한 부여 ID로 작성된 태스크만 리턴됩니다. SYSADM, SYSCTRL, SYSMAINT 또는 DBADM 권한이 있는 경우 모든 태스크가 리턴됩니다.

## 예

스케줄러에서 태스크 상태를 요청합니다.

```
SELECT * from SYSTOOLS.ADMIN_TASK_LIST
```

### 리턴되는 정보

표 77. ADMIN\_TASK\_LIST 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
NAME	VARCHAR (128)	태스크 이름

표 77. ADMIN\_TASK\_LIST 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
TASKID	INTEGER	태스크 ID
OWNER	VARCHAR (128)	태스크를 작성한 사용자의 세션 권한 부여 ID.
OWNERTYPE	VARCHAR(1)	권한 부여 ID 유형. 올바른 값은 다음과 같습니다. • U - 사용자
BEGIN_TIME	TIMESTAMP	처음으로 태스크를 실행할 수 있는 시기의 시간소인. <sup>1</sup>
END_TIME	TIMESTAMP	마지막으로 태스크를 실행할 수 있는 시기의 시간소인. <sup>1</sup>  이 컬럼이 NULL인 경우 MAX_INVOCATIONS를 지정하지 않으면 태스크는 무한 실행될 수 있습니다.
MAX_INVOCATIONS	INTEGER	태스크에 허용된 최대 실행 수. 이 컬럼이 NULL인 경우 END_TIME을 지정하지 않으면 태스크는 무한 실행될 수 있습니다.
SCHEDULE	VARCHAR(1024)	UNIX 크론 형식의 태스크에 대한 스케줄.
PROCEDURE_SCHEMA	VARCHAR (128)	이 태스크가 실행하는 프로시저 스키마.
PROCEDURE_NAME	VARCHAR (128)	이 태스크가 실행하는 프로시저 이름.
PROCEDURE_INPUT	CLOB(2M)	이 태스크가 실행하는 프로시저의 입력 매개 변수. 이 컬럼이 NULL인 경우 입력 매개변수가 없습니다.
OPTIONS	VARCHAR(512)	태스크 동작에 영향을 주는 옵션.
UPDATE_TIME	TIMESTAMP	태스크가 마지막으로 갱신된 시간소인.
REMARKS	VARCHAR(254)	태스크에 대한 설명.

주:

- <sup>1</sup> BEGIN\_TIME 및 END\_TIME은 데이터베이스 서버 시간대를 사용합니다. 사용자가 직접 일광절약시간에는 적절하게 조정해야 합니다.

## ADMIN\_TASK\_REMOVE 프로시저 - 스케줄된 태스크 또는 태스크 상태 레코드 제거

ADMIN\_TASK\_REMOVE 프로시저는 프로시저내에 캡슐화할 수 있는 작업의 일부인 스케줄된 관리 태스크를 제거합니다. 태스크 상태 레코드도 제거합니다.



## 구문

▶▶—ADMIN\_TASK\_REMOVE—(—*name*—,—*end\_timestamp*—)————▶▶

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *name*

태스크 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다.

#### *end\_timestamp*

상태 레코드 *end\_timestamp* 시간소인을 지정하는 TIMESTAMP 유형의 출력 인수입니다.

### 권한 부여

ADMIN\_TASK\_REMOVE 프로시저에 대한 EXECUTE 특권입니다. **RESTRICTIVE** 옵션을 사용하여 데이터베이스를 작성한 경우를 제외하고는 디폴트로 PUBLIC에 EXECUTE 특권이 부여됩니다.

명령문 권한 부여 ID에서 프로시저 실행을 허용하더라도 현재 세션 권한 부여 ID의 값에 따라 태스크 및 상태 레코드 제거의 성공 여부가 달라집니다. 현재 세션 권한 부여 ID는 태스크 작성 시 기록된 세션 권한 부여 ID와 일치해야 합니다. SYSADM, SYSCTRL, SYSMMAINT 또는 DBADM 권한의 사용자는 모든 태스크 및 상태 레코드를 제거할 수 있습니다. 권한이 없는 사용자가 태스크나 상태 레코드를 제거하려고 하면 SQL0551N이 리턴됩니다.

### 사용 시 참고사항

작업 단위(UOW)를 커밋하지 않으면 태스크가 제거되지 않습니다.

태스크 제거 동작은 *name* 및 *end\_timestamp* 인수 정의 방법에 따라 다릅니다.

- *end\_timestamp* 인수가 NULL인 경우,
  - *name* 인수가 NULL이면 모든 태스크 및 상태 레코드가 제거됩니다. 현재 하나 이상의 태스크가 실행 중인 경우 태스크 및 관련 상태 레코드가 제거되지 않습니다. 이 경우 SQL1464W가 리턴됩니다.
  - *name* 인수가 NULL이 아닌 경우 *name*에 일치하는 태스크 레코드가 제거됩니다. 현재 지정한 태스크가 실행 중인 경우 태스크는 제거되지 않고 SQL20453N이 리턴됩니다. 지정한 태스크가 제거되면 관련된 모든 상태 레코드도 제거됩니다.
- *end\_timestamp* 인수가 NULL이 아닌 경우,

- *name* 인수가 NULL인 경우 *end\_timestamp*이하인 *end\_timestamp* 시간소인의 모든 상태 레코드가 제거됩니다. 태스크 레코드는 제거되지 않습니다. 프로시저는 상태 값이 RUNNING인 모든 상태 레코드를 제거하지 않습니다.
- *name* 인수가 NULL이 아닌 경우 *end\_timestamp* 시간소인이 *end\_timestamp* 이하인 경우 *name*에 일치하는 태스크의 상태 레코드가 제거됩니다. 태스크 레코드는 제거되지 않습니다. 프로시저는 상태 값이 RUNNING인 모든 상태 레코드를 제거하지 않습니다.

현재 없는 태스크를 제거하려고 하면 SQL0204N이 리턴됩니다.

## 예

백업 태스크 'DAILY TSM BACKUP'을 제거합니다.

```
CALL SYSPROC.ADMIN_TASK_REMOVE('DAILY TSM BACKUP', NULL)
```

---

## ADMIN\_TASK\_STATUS 관리 뷰 - 태스크 상태 정보 검색

ADMIN\_TASK\_STATUS 관리 뷰는 관리 태스크 스케줄러에서 태스크 실행 상태에 대한 정보를 검색합니다.

스키마는 SYSTOOLS입니다.

이 뷰는 ADMIN\_TASK\_ADD 프로시저를 처음 호출할 때 작성됩니다.

## 권한 부여

ADMIN\_TASK\_STATUS 관리 뷰의 SELECT 또는 CONTROL 특권. **RESTRICTIVE** 옵션을 사용하여 데이터베이스를 작성한 경우를 제외하고는 디폴트로 PUBLIC에 SELECT 특권이 부여됩니다.

ADMIN\_TASK\_STATUS 뷰를 쿼리하면 사용자의 세션 권한 부여 ID로 작성된 태스크 상태 레코드만 리턴됩니다.

## 예

예 1: 스케줄러에서 태스크 상태를 요청합니다.

```
SELECT * from SYSTOOLS.ADMIN_TASK_STATUS
```

예 2: SQLERRM 함수를 사용하여 SQLERRMC 컬럼에서 데이터를 형식화합니다.

```
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
```

## 리턴되는 정보

표 78. ADMIN\_TASK\_STATUS 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
NAME	VARCHAR (128)	태스크 이름
TASKID	INTEGER	태스크 ID.
STATUS	VARCHAR(10)	태스크 상태. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• RUNNING - 현재 태스크가 실행 중입니다.</li> <li>• COMPLETED - 태스크 실행이 완료되었습니다.</li> <li>• NOTRUN - 오류로 인해 스케줄러가 태스크 프로시저를 호출하지 못했습니다.</li> <li>• UNKNOWN - 태스크 실행이 시작되었지만 예기치 않은 조건으로 인해 스케줄러가 태스크 출력을 기록하지 못했습니다. 이 문제는 태스크 실행 중에 시스템이 비정상적으로 종료되었거나 전원 장애가 생긴 경우에 발생할 수 있습니다.</li> </ul>
INVOCATION	INTEGER	현재 호출 계수.
BEGIN_TIME	TIMESTAMP	태스크가 시작되는 시간. <sup>1</sup>  STATUS가 RUNNING, COMPLETED 또는 UNKNOWN인 경우 이 값은 태스크 실행이 시작된 시간을 나타냅니다.  STATUS가 NOTRUN인 경우 이 값은 태스크가 시작되어야 하는 시간을 나타냅니다.
END_TIME	TIMESTAMP	태스크 실행이 완료된 시간. <sup>1</sup>  STATUS가 RUNNING인 경우 이 값은 NULL이 됩니다.  STATUS가 UNKNOWN인 경우 이 값은 태스크 스케줄러에서 태스크가 더 이상 실행되지 않음을 발견하고 상태 테이블이 갱신된 시간입니다.
AGENT_ID	BIGINT	태스크를 실행하는 응용프로그램의 에이전트 ID. 에이전트 ID는 응용프로그램 핸들과 같은 의미입니다. 이 값은 태스크가 실행되는 동안에만 유효합니다.
SQLCODE	INTEGER	STATUS가 COMPLETED인 경우 이 값은 프로시저에 대한 CALL로 리턴된 SQLCODE를 나타냅니다.  STATUS가 NOTRUN인 경우 이 값은 태스크가 실행되지 않도록 하는 오류의 SQLCODE를 나타냅니다.  STATUS가 RUNNING 또는 UNKNOWN인 경우 이 값은 NULL입니다.
SQLSTATE	CHAR(5)	STATUS가 COMPLETED인 경우 이 값은 프로시저에 대한 CALL로 리턴된 SQLSTATE를 나타냅니다.  STATUS가 NOTRUN인 경우 이 값은 태스크가 실행되지 않도록 하는 오류의 SQLSTATE를 나타냅니다.  STATUS가 RUNNING 또는 UNKNOWN인 경우 이 값은 NULL입니다.

표 78. ADMIN\_TASK\_STATUS 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
SQLERRMC	VARCHAR(70) FOR BIT DATA	<p>하나 이상의 토큰을 포함하고 X'FF'로 구분되며, SQLCA의 SQLERRMC 필드에 표시됩니다. 이 토큰은 오류 조건 설명에서 변수로 대체됩니다.</p> <p>STATUS가 COMPLETED인 경우 이 값은 프로시저에 대한 CALL로 리턴된 SQLERRMC를 나타냅니다.</p> <p>STATUS가 NOTRUN인 경우 이 값은 태스크가 실행되지 않도록 하는 오류의 SQLERRMC를 나타냅니다.</p> <p>STATUS가 RUNNING 또는 UNKNOWN인 경우 이 값은 NULL입니다.</p>
RC	INTEGER	STATUS가 COMPLETED인 경우 프로시저에 리턴 코드가 있는 경우 여기에는 프로시저에 대한 CALL에서의 리턴 코드가 포함됩니다. 그 이외의 경우에는 NULL이 됩니다.

- <sup>1</sup> BEGIN\_TIME 및 END\_TIME은 데이터베이스 서버 시간대를 사용합니다. 사용자가 직접 일광절약시간에는 적절하게 조정해야 합니다.

## ADMIN\_TASK\_UPDATE 프로시저 - 기존 태스크 갱신

ADMIN\_TASK\_UPDATE 프로시저는 프로시저 내부에 캡슐화될 수 있는 작업의 모든 부분을 포함하는 관리 태스크를 갱신합니다.

### 구문

```

▶—ADMIN_TASK_UPDATE—(—name—,—begin_timestamp—,—end_timestamp—,—
▶—max_invocations—,—schedule—,—options—,—remarks—)

```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *name*

기존 태스크 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. 이 인수는 NULL이 될 수 없습니다.

#### *begin\_timestamp*

태스크 실행이 시작될 수 있는 가장 빠른 시간을 지정하는 TIMESTAMP 유형의 입력 인수입니다. 이 인수 값은 지나간 과거 시간일 수 없으며 *end\_timestamp* 이후 시간일 수 없습니다.

태스크 실행 시작 시점은 이 매개변수 및 *schedule* 매개변수 정의에 따라 달라집니다.

- *begin\_timestamp* 인수가 NULL이 아닌 경우,

- *schedule* 인수가 NULL인 경우 태스크 실행은 *begin\_timestamp*에 시작됩니다.
- *schedule* 인수가 NULL이 아닌 경우 태스크 실행은 *begin\_timestamp* 시점 또는 이후의 다음 스케줄된 시간에 시작됩니다.
- *begin\_timestamp* 인수가 NULL인 경우,
  - *schedule* 인수가 NULL인 경우 태스크 실행은 즉시 시작됩니다.
  - *schedule* 인수가 NULL이 아닌 경우 태스크 실행은 다음 스케줄 시간에 시작됩니다.

#### *end\_timestamp*

태스크 실행이 시작될 수 있는 마지막 시간을 지정하는 **TIMESTAMP** 유형의 입력 인수입니다. 이 인수 값은 지나간 과거 시간일 수 없으며 *begin\_timestamp* 이전 시간일 수 없습니다. 인수가 NULL인 경우 태스크는 스케줄된 대로 무제한 지속될 수 있습니다.

태스크 실행은 해당 *end\_timestamp* 시에도 인터럽트되지 않습니다.

#### *max\_invocations*

태스크에 허용되는 최대 실행 수를 지정하는 **INTEGER** 유형의 입력 인수입니다. 인수가 NULL인 경우 태스크 실행 가능 횟수에 제한이 없습니다. 인수가 0인 경우 태스크는 실행되지 않습니다.

*schedule*이 NULL이 아닌 경우 이 값이 스케줄에 적용됩니다.

*end\_timestamp* 및 *max\_invocations* 모두 지정되는 경우 *end\_timestamp*에 우선권이 있습니다. 즉, *end\_timestamp* 시간소인에 도달하면 태스크 실행 수가 아직 *max\_invocations* 값에 도달하지 않아도 태스크는 다시 실행되지 않습니다.

#### *schedule*

시간내에서 고정된 시점에 태스크가 실행되도록 지정하는 **VARCHAR(1024)** 유형의 입력 인수입니다. 인수가 NULL인 경우 시간내의 고정된 시점에 태스크가 스케줄되지 않습니다.

*schedule* 문자열은 UNIX 크론 형식으로 지정해야 합니다.

스케줄은 여러 개 지원되지 않습니다.

#### *options*

**VARCHAR(512)** 유형의 입력 인수입니다. 인수는 NULL이어야 합니다.

#### *remarks*

태스크 설명을 지정하는 **VARCHAR(254)** 유형의 입력 인수입니다. 이 인수는 NULL로 설정할 수 있는 선택적 인수입니다.

## 권한 부여

ADMIN\_TASK\_UPDATE 프로시저에 대한 EXECUTE 특권. **RESTRICTIVE** 옵션을 사용하여 데이터베이스를 작성한 경우를 제외하고는 디폴트로 PUBLIC에 EXECUTE 특권이 부여됩니다.

명령문 권한 부여 ID에서 프로시저 실행을 허용하더라도 현재 세션의 권한 부여 ID가 태스크 작성 시 기록된 세션 권한 부여 ID에 일치하지 않으면 태스크를 갱신할 수 없습니다. SYSADM, SYSCTRL, SYSMANT 또는 DBADM 권한의 사용자는 모든 태스크를 갱신합니다. 다른 사용자로 추가된 태스크 갱신을 시도하면 SQL0551N이 리턴됩니다.

## 사용 시 참고사항

유효하지 않은 인수가 프로시저에 전달되면 SQL0171N이 리턴됩니다. 메시지 토큰은 유효하지 않은 인수 및 프로시저 이름을 표시합니다.

태스크의 변경사항은 작업 단위(UOW)가 커밋되고 스케줄러가 갱신된 태스크 정의를 폐지하기 전까지는 적용되지 않습니다. 작업 단위(UOW)를 커밋하지 않은 채로 두면 기존 태스크 실행이 지연되거나 실행되지 않습니다.

스케줄러는 매 5분마다 갱신된 태스크를 점검합니다. 태스크가 *begin\_timestamp*에 정의된 대로 가장 빠른 시간에 예상대로 실행되도록 하려면 *end\_timestamp* 및 *schedule* 매개변수는 작업 단위(UOW) 커밋 후에 최소 5분 후로 설정되어야 합니다.

스케줄러가 태스크를 실행할 수 있도록 하려면 모든 데이터베이스 파티션에서 데이터베이스가 활성화되어 있어야 합니다.

*begin\_timestamp*, *end\_timestamp* 및 *schedule*은 데이터베이스 서버의 시간대를 기반으로 합니다. 일광절약시간(DST) 전환 시기에는 태스크 스케줄 시에 특히 주의해야 합니다. 태스크가 오전 2:01에 실행되도록 스케줄되어 있고 1시간을 앞으로 당기는 시기가 되면 시계가 오전 2:00에서 오전 3:00로 건너뛰기 때문에 태스크가 실행되지 않습니다. 반면에 시간을 다시 원래 시간으로 돌려야될 때가 되면 원래 오전 2:00에서 3:00 사이에 스케줄되었던 태스크는 2번 실행됩니다. 원하는 대로 작동할 수 있도록 사용자가 직접 일광절약시간에는 적절하게 조정해야 합니다.

태스크가 갱신되면 태스크의 내부 호출 카운터가 재설정됩니다. 이를 설명하려면 *max\_invocations* 값을 10으로 하여 태스크를 반복한다고 가정해 보십시오. 태스크가 3번 실행되면 ADMIN\_TASK\_STATUS 출력에 해당하는 3개의 상태가 있게 됩니다. 항목의 INVOCATION 값은 각각 1, 2, 3이 됩니다. 이제 태스크 작성자가 태스크를 갱신한다고 가정해 보십시오. 이 갱신으로 인해 내부 호출 카운터가 재설정됩니다. 원래 상태 레코드는 그대로 유지됩니다. 시간이 흐르면 새 상태 레코드가 INVOCATION

값 1, 2, 3 등으로 작성됩니다. BEGIN\_TIME을 사용하여 원래 태스크 실행과 갱신된 태스크 실행을 구별할 수 있습니다.





---

## 제 6 장 감사 루틴 및 프로시저

---

### AUDIT\_ARCHIVE 프로시저 및 테이블 함수 - 아카이브 감사 로그 파일

AUDIT\_ARCHIVE 프로시저 및 테이블 함수 모두 연결된 데이터베이스에 대한 감사 로그 파일을 아카이브합니다.

#### 구문

▶—AUDIT\_ARCHIVE—(—directory—,—dbpartitionnum—)————▶

스키마는 SYSPROC입니다.

구문이 프로시저 및 테이블 함수 모두와 동일합니다.

#### 프로시저 및 테이블 함수 매개변수

##### *directory*

아카이브된 감사 파일이 작성될 디렉토리를 지정하는 VARCHAR(1024) 유형의 입력 인수입니다. 디렉토리는 서버에 존재해야 하며 인스턴스 소유자는 해당 디렉토리에서 파일을 작성할 수 있어야 합니다. 인수가 널(NULL)이거나 비어 있는 문자열인 경우, 디폴트 디렉토리가 사용됩니다.

##### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션의 집계에는 NULL 또는 -2를 지정하십시오.

#### 권한 부여

AUDIT\_ARCHIVE 프로시저 또는 테이블 함수에 대한 실행 특권.

#### 예:

예 1: 프로시저를 사용하는 디폴트 디렉토리에 모든 데이터베이스 파티션의 감사 로그를 아카이브하십시오.

```
CALL SYSPROC.AUDIT_ARCHIVE(NULL, NULL)
```

예 2: 테이블 함수를 사용하는 디폴트 디렉토리에 모든 데이터베이스 파티션의 감사 로그를 아카이브하십시오.

```
SELECT * FROM TABLE(SYSPROC.AUDIT_ARCHIVE(' ', -2)) AS T1
```

## 리턴된 정보

표 79. *AUDIT\_ARCHIVE* 프로시저 및 테이블 함수로 리턴된 정보

컬럼 이름	데이터 유형	설명
DBPARTITIONNUM	INTEGER	아카이브된 파일의 파티션 수.
PATH	VARCHAR(1024)	아카이브된 파일의 디렉토리 위치.
FILE	VARCHAR(1024)	아카이브된 파일의 이름.
SQLCODE	INTEGER	파일 아카이브 시도 중 수신된 SQLCODE.
SQLSTATE	VARCHAR(5)	파일 아카이브 시도 중 수신된 SQLSTATE. SQLSTATE가 NULL이면 값은 0입니다.
SQLERRMC	VARCHAR(70) FOR BIT DATA	파일 아카이브 시도 중 수신된 sqlerrmc. SQLSTATE가 NULL이면 값은 0입니다.

## AUDIT\_DELIM\_EXTRACT - 구분 파일로 추출 수행

AUDIT\_DELIM\_EXTRACT 스토어드 프로시저는 연결된 데이터베이스의 아카이브된 감사 파일의 구분 파일로 추출합니다. 특히 지정된 마스크 패턴과 일치하는 파일 이름이 있는 아카이브된 감사 파일로 추출합니다.

### 구문

```
▶—AUDIT_DELIM_EXTRACT—(—delimiter—,—target_directory—,—source_directory—,—  
▶—file_mask—,—event_options—)
```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *delimiter*

구분 파일에 사용되는 분리문자를 지정하는 VARCHAR(1) 유형의 선택적 입력 인수. 인수가 널(null)이거나 빈 문자열이면 큰따옴표는 분리문자로 사용됩니다.

#### *target\_directory*

구분 파일이 저장된 디렉토리를 지정하는 VARCHAR(1024) 유형의 선택적 입력 인수. 인수가 널(null)이거나 빈 문자열이면 *source\_directory*와 동일한 디렉토리가 사용됩니다.

#### *source\_directory*

아카이브된 감사 로그 파일이 저장된 디렉토리를 지정하는 VARCHAR(1024) 유형의 선택적 입력 인수. 인수가 널(null)이거나 빈 문자열이면 감사 디폴트가 사용됩니다.

### *file\_mask*

VARCHAR(1024) 유형의 선택적 입력 인수는 추출할 파일에 대한 마스크입니다. 인수가 널(null)이거나 빈 문자열이면 소스 디렉토리의 모든 감사 로그 파일로부터 추출됩니다.

### *event\_options*

추출할 이벤트를 정의하는 문자열을 지정하는 VARCHAR(128) 유형의 선택적 입력 인수. db2audit 유틸리티의 동일한 문자열과 일치합니다. 인수가 널(null)이거나 빈 문자열이면 모든 이벤트를 추출합니다.

## 권한 부여

AUDIT\_DELIM\_EXTRACT 및 AUDIT\_LIST\_LOGS 함수에서 특권을 실행합니다.

### 예:

주: 감사 로그 파일은 이름 지정 규칙의 파트로 시간소인을 포함합니다.

예 1: 디폴트 아카이브 디렉토리에서 2007년 6월 18일에 아카이브된 모든 감사 로그 파일에서 구분 추출을 수행합니다. 이 예는 실행 이벤트만을 추출하며 큰따옴표(") 분리 문자를 사용하고 결과 추출 파일(<category>.del)을 \$HOME/audit\_delim\_extract 디렉토리에 작성하거나 추가합니다.

```
CALL SYSPROC.AUDIT_DELIM_EXTRACT(NULL, '$HOME/AUDIT_DELIM_EXTRACT', NULL, '%20070618%', 'CATEGORIES EXECUTE STATUS BOTH')
```

---

## AUDIT\_LIST\_LOGS 테이블 함수 - 아카이브된 감사 로그 파일 나열

AUDIT\_LIST\_LOGS 테이블 함수는 지정된 디렉토리에서 표시되는 데이터베이스의 아카이브된 감사 로그 파일을 나열합니다.

### 구문

```
▶▶ AUDIT_LIST_LOGS (—디렉토리—) ▶▶
```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *directory*

아카이브된 감사 파일이 작성될 디렉토리를 지정하는 VARCHAR(1024) 유형의 선택적 입력 인수입니다. 디렉토리는 서버에 존재해야 하며 인스턴스 소유자는 해당 디렉토리에서 파일을 작성할 수 있어야 합니다. 인수가 널(NULL)이거나 비어 있는 문자열이면 검색 디폴트 디렉토리가 사용됩니다.

## 권한 부여

AUDIT\_LIST\_LOGS 테이블 함수에 대한 EXECUTE 특권.

### 예:

예 1: 디폴트 감사 아카이브 디렉토리에서 모든 아카이브된 감사 로그를 나열합니다.

```
SELECT * FROM TABLE(SYSPROC.AUDIT_LIST_LOGS('')) AS T1
```

주: 이는 쿼리가 실행되는 데이터베이스에 대한 디렉토리의 로그만 나열합니다. 아카이브된 파일에 db2audit.db.<dbname>.log.<timestamp> 형식이 있습니다.

## 리턴된 정보

표 80. AUDIT\_LIST\_LOGS에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
PATH	VARCHAR(1024)	아카이브된 파일의 경로 위치입니다.
FILE	VARCHAR(1024)	아카이브된 파일의 파일 이름입니다.
SIZE	BIGINT	아카이브된 파일의 파일 크기입니다.

---

## 제 7 장 자동 유지보수 루틴

---

### AUTOMAINT\_GET\_POLICY 프로시저 - 자동 유지보수 규정 검색

AUTOMAINT\_GET\_POLICY 시스템 스토어드 프로시저는 데이터베이스에 대한 자동 유지보수 구성을 검색합니다. 이 프로시저에는 두 개의 매개변수(정보를 수집할 자동 유지보수의 유형과 구성 정보를 리턴할 BLOB에 대한 포인터)가 있습니다. 구성 정보는 XML 형식으로 리턴됩니다.

#### 구문

```
▶▶AUTOMAINT_GET_POLICY(—policy_type—,—policy—)▶▶
```

스키마는 SYSPROC입니다.

#### 프로시저 매개변수

##### *policy\_type*

검색할 자동 유지보수의 유형을 지정하는 VARCHAR(128) 유형의 입력 인수. 값은 다음 중 하나가 될 수 있습니다.

##### *AUTO\_BACKUP*

자동 백업

##### *AUTO\_REORG*

자동 테이블 및 인덱스 재구성

##### *AUTO\_RUNSTATS*

자동 테이블 runstats 조작

##### *MAINTENANCE\_WINDOW*

유지보수 창

##### *policy*

제공된 규정 유형에 대한 자동 유지보수 설정을 XML 형식으로 지정하는 BLOB(2M) 유형의 출력 인수

#### 권한 부여

AUTOMAINT\_GET\_POLICY 프로시저에 대한 EXECUTE 특권

예 :

다음은 Embedded SQL C 소스 코드 내에서 AUTOMAINT\_GET\_POLICY 프로시저로 호출하는 예입니다.

- BLOB 변수가 프로시저 출력 매개변수에 선언됩니다.
- 자동화된 백업을 자동 유지보수 규정의 유형으로 지정하고 프로시저가 현재 연결된 데이터베이스의 백업 규정을 리턴하는 출력 매개변수로 BLOB 변수를 지정하여 프로시저가 호출됩니다.

```
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 프로시저 - 자동 유지보수 규정 검색

AUTOMAINT\_GET\_POLICYFILE 시스템 스토어드 프로시저는 데이터베이스에 대한 자동 유지보수 구성을 검색합니다. 이 프로시저에는 두 개의 매개변수(정보를 수집할 자동 유지보수의 유형과 구성 정보를 리턴할 파일의 이름)가 있습니다. 구성 정보는 XML 형식으로 리턴됩니다.

### 구문

```
▶▶AUTOMAINT_GET_POLICYFILE(—policy_type—,—policy_file_name—)————▶▶
```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *policy\_type*

검색할 자동 유지보수의 유형을 지정하는 VARCHAR(128) 유형의 입력 인수. 값은 다음 중 하나가 될 수 있습니다.

#### *AUTO\_BACKUP*

자동 백업

#### *AUTO\_REORG*

자동 테이블 및 인덱스 재구성

#### *AUTO\_RUNSTATS*

자동 테이블 runstats 조작

#### *MAINTENANCE\_WINDOW*

유지보수 창

*policy\_file\_name*

DB2 인스턴스 디렉토리의 tmp 서브디렉토리에 작성된 파일의 이름을 지정하는 VARCHAR(2048) 유형의 입력 인수

주: 파일 이름에 tmp와 관련된 경로가 앞에 붙습니다. 이 경우 디렉토리가 존재해야 하고 파일 작성/접촉권 권한이 있어야 하며 DB2 서버의 올바른 경로 구분자를 사용해야 합니다.

예를 들어, 다음과 같습니다.

UNIX에서 인스턴스 디렉토리가 \$HOME/sqllib로 정의된 경우, 이름이 'policy.xml'인 정책 파일은 파일 이름이 '\$HOME/sqllib/tmp/policy.xml'이 됩니다.

Windows에서 인스턴스 디렉토리 이름은 **DB2INSTPROF** 레지스트리 변수 및 **DB2INSTANCE** 환경 변수의 값으로 판별될 수 있습니다. 이름이 'policy.xml'인 정책 파일의 경우 db2set가 DB2INSTPROF=C:\#DB2PROF 및 %DB2INSTANCE%=db2를 제공하면 파일 이름이 C:\#DB2PROF\#db2\#tmp\#policy.xml이 됩니다.

## 권한 부여

AUTOMAINT\_GET\_POLICYFILE 프로시저에 대한 EXECUTE 특권

예 :

백업 작업에 대한 현재 자동 유지보수 설정 가져오기:

```
call sysproc.automaint_get_policyfile( 'AUTO_BACKUP', 'AutoBackup.xml' )
```

이것은 DB2 인스턴스 디렉토리 밑의 tmp 서브디렉토리에 이름이 AutoBackup.xml인 XML 파일을 작성합니다.

---

## AUTOMAINT\_SET\_POLICY 프로시저 - 자동 유지보수 규정 구성

AUTOMAINT\_SET\_POLICY 시스템 스토어드 프로시저를 사용하여 데이터베이스에 대한 자동 유지보수를 구성할 수 있습니다. 이 프로시저에는 구성을 지정하는 XML을 포함한 BLOB 및 구성할 자동 유지보수 유형의 두 매개변수가 있습니다.

다차원적으로 클러스터된(MDC) 테이블에서 자동 재구성 조작 중 Extent 재개를 가능하게 하려면 『reclaimExtentsSizeForMDCTables』 속성을 ReorgOptions 요소로 지정하고 XML 입력 파일의 임계값도 지정해야 합니다. 이 임계값은 재개 가능한 테이블의 최소 여유 공간 크기(KB)입니다. 이 값은 0이상 이어야 합니다. 예를 들어, 임계값으로 1024KB 값을 지정하면 1MB 이상의 여유 공간이 있는 테이블만 Extent를 재개할 자동 재구성 대상으로 고려됩니다.

## 구문

▶▶—AUTOMAINT\_SET\_POLICY—(—*policy\_type*—,—*policy*—)————▶▶

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *policy\_type*

구성할 자동 유지보수 규정의 유형을 지정하는 VARCHAR(128) 유형의 입력 인수. 값은 다음 중 하나가 될 수 있습니다.

#### **AUTO\_BACKUP**

자동 백업

#### **AUTO\_REORG**

자동 테이블 및 인덱스 재구성

#### **AUTO\_RUNSTATS**

자동 테이블 runstats 조작

#### **MAINTENANCE\_WINDOW**

유지보수 창

### *policy*

XML 형식으로 자동 유지보수 규정을 지정하는 BLOB(2M) 유형의 입력 인수.

## 권한 부여

SYSPROC.AUTOMAINT\_SET\_POLICY 프로시저의 EXECUTE 특권.

## 예:

**예 1:** runstats 조작으로 현재 자동 유지보수 설정을 설정:

```
CALL SYSPROC.AUTOMAINT_SET_POLICY
( 'AUTO_RUNSTATS',
  BLOB(' <?xml version="1.0" encoding="UTF-8"?>
    <DB2AutoRunstatsPolicy xmlns=
    #"http://www.ibm.com/xmlns/prod/db2/autonomic/config#">
    <RunstatsTableScope><FilterCondition/></RunstatsTableScope>
    </DB2AutoRunstatsPolicy>')
)
```

현재 자동 통계 컬렉션 구성을 두 번째 매개변수로 프로시저에 전달된 XML 문서에 포함된 새 구성으로 교체됩니다.”

**예 2:** DB2의 자동 재구성 기능에서는 새로 추가된 "RECLAIM EXTENTS ONLY" 옵션을 사용하여 다차원적으로 클러스터된(MDC) 테이블을 재구성할 수 있습니다. 이 기능을 사용하려면 "reclaimExtentsSizeForMDCTables" 값을 AUTO\_REORG 규정에 설정하십시오.



```

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" reclaimExtentsSizeForMDCTables ="1024" >
      <ReorgTableScope>
        <FilterClause>TABSCHEMA NOT LIKE
          'EMP%'</FilterClause>
        </ReorgTableScope>
      </DB2AutoReorgPolicy>')
)

```

사용자의 요구사항에 맞게 수정한 다음 예제와 같이 BLOB() 스칼라 함수를 통해 XML 콘텐츠를 전달할 수 있는 샘플 XML 입력 파일이 SQLLIB/samples/automaintcfg 디렉토리에 있습니다.

---

## AUTOMAINT\_SET\_POLICYFILE 프로시저 - 자동 유지보수 규정 구성

AUTOMAINT\_SET\_POLICYFILE 시스템 스토어드 프로시저를 사용하여 데이터베이스에 대한 자동 유지보수를 구성할 수 있습니다. 이 프로시저에는 두 개의 매개변수(구성할 자동 유지보수의 유형과 구성을 지정하는 XML 문서의 이름)가 있습니다.

이 프로시저는 SQL 성공 또는 SQL 오류 코드를 리턴합니다.

### 구문

```

▶▶—AUTOMAINT_SET_POLICYFILE—(—policy_type—,—policy_file_name—)————▶▶

```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *policy\_type*

구성할 자동 유지보수 규정의 유형을 지정하는 VARCHAR(128) 유형의 입력 인수. 값은 다음 중 하나가 될 수 있습니다.

#### *AUTO\_BACKUP*

자동 백업

#### *AUTO\_REORG*

자동 테이블 및 인덱스 재구성

#### *AUTO\_RUNSTATS*

자동 테이블 runstats 조작

#### *MAINTENANCE\_WINDOW*

유지보수 창

*policy\_file\_name*

DB2 인스턴스 디렉토리의 tmp 서브디렉토리에 사용할 수 있는 파일의 이름을 지정하는 VARCHAR(2048) 유형의 입력 인수

주: 파일 이름이 관련 경로로 지정되면 DB2 서버의 올바른 경로 구분자를 사용해야 하며 읽기 사용 권한이 있는 디렉토리 및 파일이 존재해야 합니다.

예를 들어, 다음과 같습니다.

UNIX에서 인스턴스 디렉토리가 \$HOME/sqllib로 정의된 경우. 이름이 'automaint/policy.xml'인 정책 파일은 파일 이름이 '\$HOME/sqllib/tmp/automaint/policy.xml' 이 됩니다.

Windows에서 인스턴스 디렉토리 이름은 **DB2INSTPROF** 레지스트리 변수 및 **DB2INSTANCE** 환경 변수의 값으로 판별될 수 있습니다. 이름이 'automaint#policy.xml'인 정책 파일의 경우 db2set가 DB2INSTPROF=C:#DB2PROF 및 %DB2INSTANCE%=db2를 제공하면 파일 이름이 C:#DB2PROF#db2#tmp#automaint#policy.xml이 됩니다.

## 권한 부여

SYSPROC.AUTOMAINT\_SET\_POLICYFILE 프로시저에 대한 EXECUTE 특권

예 :

자동 백업을 위한 현재 자동 유지보수 설정 가져오기:

```
call sysproc.automaint_set_policyfile( 'AUTO_BACKUP', 'AutoBackup.xml' )
```

현재 자동 백업 구성 설정을 DB2 인스턴스 디렉토리의 tmp 디렉토리에 있는 AutoBackup.xml 파일에 포함된 새 구성으로 교체합니다.

정책 xml 파일을 작성하도록 참조로 사용할 수 있는 SQLLIB/samples/automaintcfg 디렉토리에 있는 샘플 XML 입력 파일이 있습니다.

## 제 8 장 일반 SQL API 프로시저

일반 SQL API는 IBM® Data Server 사이에서 이식 가능한 일반 서명 및 안전한 서명의 스토어드 프로시저 콜렉션을 제공합니다. 이 스토어드 프로시저를 사용하여 다양한 일반 관리 함수(예: 구성 매개변수 가져오기 및 설정, 시스템 정보 가져오기)를 수행하는 응용프로그램을 작성할 수 있습니다.

스토어드 프로시저는 구문상으로 동일한 XML 매개변수 및 데이터 서버 버전이 종속되지 않도록 모든 데이터 서버에서 오류 조절을 제공합니다. 서명 안정성 및 일반성은 단순 XML 문서(일반 DTD 포함)를 매개변수로 사용하여 확보됩니다. 버전, 플랫폼 및 기술 차이는 계층적인 등록 정보 목록에 다른 키 값 쌍으로 표시됩니다.

### 일반 입력 및 출력 매개변수

일반 SQL API 스토어드 프로시저는 입력 및 출력 매개변수의 세트를 공유합니다.

다음 표는 이 매개변수에 대한 간단한 설명을 제공합니다. 자세한 정보는 일반 SQL API 스토어드 프로시저에 대한 참조 주제를 참조하십시오.

표 81. 입력 및 출력 매개변수를 공유한 일반 SQL API

매개변수	설명
<i>major_version</i>	프로시저에서 매개변수로 전달되는 XML 문서에 대해 호출자가 지원하는 문서 유형 기본 버전을 나타냅니다.
<i>minor_version</i>	<p>프로시저에서 매개변수로 전달되는 XML 문서에 대해 호출자가 지원하는 문서 유형 보조 버전을 나타냅니다.</p> <p><i>major_version</i> 및 <i>minor_version</i> 매개변수를 사용하여 호출자가 올바르지 않은 버전의 XML 입력 문서를 사용하지 않도록 합니다. 프로시저는 지정된 <i>major_version</i> 및 <i>minor_version</i>의 모든 XML 문서를 처리하거나 버전이 유효하지 않은 경우 오류를 리턴합니다. 새로 추가되는 문서 유형 버전은 기존 응용프로그램에 영향을 주지 않으면서 추가 가능하기 때문에 이 디자인은 추후 릴리스에서 확장성을 지원 합니다.</p>
<i>requested_locale</i>	변환된 콘텐츠를 <i>xml_output</i> 및 <i>xml_message</i> 매개변수로 리턴된 XML 문서로 리턴하는 데 사용하려는 로케일을 지정합니다. 값만 변환되며 키 이름은 변환되지 않습니다.
<i>xml_input</i>	프로시저에 대한 입력 값이 포함되는 XML 입력 문서를 지정합니다.

표 81. 입력 및 출력 매개변수를 공유한 일반 SQL API (계속)

매개변수	설명
<i>xml_filter</i>	출력 매개변수 문서에서 한 개의 값만 검색하는 데 사용되는 유효한 XPath 쿼리 문자열을 지정합니다.
<i>xml_output</i>	UTF-8로 인코딩된 전체 XML 출력 문서를 리턴합니다. 호출 중인 프로시저에 따라 이 문서는 구성 매개변수 및 해당 값, 시스템 정보 또는 메시지 텍스트를 포함할 수 있습니다. 프로시저가 완료 모드에서 작동하는 경우 이 매개변수는 수정할 수 있으며 프로시저를 다시 <i>xml_input</i> 매개변수로 되돌릴 수 있는 XML 문서를 리턴합니다. 이 접근방식은 유효한 XML 입력 문서를 작성하는 프로그래밍된 방법입니다.
<i>xml_message</i>	SQL 경고 조건에 대한 자세한 정보를 제공하는 데이터 서버 메시지 유형의 전체 XML 출력 문서를 UTF-8로 리턴합니다.

## XML 문서 버전화

추후 릴리스에서 확장을 지원하기 위해 일반 SQL API 스토어드 프로시저는 버전 정보를 포함하는 XML 출력 문서를 리턴합니다.

XML 출력 문서 구조가 변경될 때마다(예: 요소가 추가되거나 제거될 때) 버전 레벨이 증가됩니다. 따라서 프로시저는 XML 출력 문서의 여러 버전을 지원할 수도 있습니다.

XML 문서의 버전 정보는 문서 유형 기본 버전 및 문서 유형 보조 버전에 대한 키 값 쌍으로 표시됩니다. 예를 들어 XML 출력 문서는 다음 키 및 값을 사전 요소로 정의할 수 있습니다.

```
<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>
```

프로시저를 호출할 때 리턴하려는 XML 문서의 기본 및 보조 버전을 지정합니다. XML 출력 문서의 콘텐츠는 지정한 값에 따라 다릅니다.

예를 들어 GET\_CONFIG 프로시저는 데이터베이스 및 특정 인스턴스에 대해 설정된 데이터베이스 관리 프로그램 구성 매개변수를 검색합니다. 이 프로시저에서 *major\_version* 2 및 *minor\_version* 0이 호출된 경우 이 프로시저는 범주로 그룹화된 구성 매개변수를 포함하는 XML 문서를 리턴합니다. 그렇지만 동일한 프로시저에서 *major\_version* 1 및 *minor\_version* 0이 호출된 경우 구성 매개변수가 포함된 XML 문서는 리턴되지만 범주로 그룹화되지는 않습니다.

이와 유사하게 GET\_MESSAGE 프로시저는 지정한 SQLCODE에 대한 메시지 텍스트 및 SQLSTATE를 검색합니다. 이 프로시저에서 *major\_version* 2 및 *minor\_version* 0이 호출된 경우 해당 SQLCODE에 대한 간략한 텍스트 메시지, 자세한 텍스트 메시

지 및 SQLSTATE가 포함된 XML 문서가 리턴됩니다. 그렇지만 동일한 프로시저에서 *major\_version* 1 및 *minor\_version* 0이 호출된 경우 간략한 텍스트 메시지 및 SQLSTATE만 포함된 XML 문서가 리턴됩니다. 문서의 버전 1에서는 자세한 텍스트 메시지를 사용할 수 없습니다.

프로시저에 대해 지원되는 가장 높은 문서 버전을 판별하려면 *major\_version*, *minor\_version* 및 다른 기타 입력 매개변수에 대해 NULL을 지정하십시오. 프로시저는 지원되는 가장 높은 문서 버전을 *major\_version* 및 *minor\_version* 출력 매개변수에 값으로 리턴하고 *xml\_output* 및 *xml\_message* 출력 매개변수를 NULL로 설정합니다.

*major\_version* 및 *minor\_version*에 대해 널(NULL)이 아닌 값을 지정하려면 지원되는 문서 버전을 지정해야 합니다. 지정하지 않으면 프로시저에서는 지원되지 않는 버전의 발생을 알리는 오류(-20457)가 표시됩니다.

XML 입력 문서는 선택적으로 문서 유형 기본 버전 및 문서 유형 보조 버전에 대한 값을 포함합니다. 이 값을 XML 입력 문서에 지정하는 경우 프로시저 호출에서 *major\_version* 및 *minor\_version*에 전달된 값은 XML 문서에 지정된 값과 정확하게 일치해야 하며 일치하지 않으면 프로시저에서 오류(+20458)가 발생합니다. 이 동작으로 인해 호출자는 지원되지 않는 버전의 XML 입력 문서를 지정하지 않습니다.

---

## XML 입력 문서

일반 SQL API 스토어드 프로시저의 입력으로 전달되는 XML 문서는 일반 DTD를 기반으로 하는 단순 XML 형식을 공유합니다.

XML 입력 문서는 모든 스토어드 프로시저에 공통되는 항목 세트 및 각 스토어드 프로시저에만 고유하게 적용되는 항목 세트로 구성됩니다. XML 입력 문서는 다음과 같은 일반 구조로 되어 있습니다.

```
<?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>
```

**중요사항:** XML 입력 문서는 UTF-8로 인코딩되어야 하며 영어 문자만 포함합니다.

## 유효한 XML 입력 문서 리턴을 위한 완료 모드

완료 모드를 사용하여 입력을 허용하는 모든 일반 SQL API 스토어드 프로시저에 대해 유효한 XML 문서를 작성할 수 있습니다. 그런 다음 문서를 사용자 정의하여 다시 프로시저로 전달합니다.

완료 모드로 프로시저를 실행하려면 입력 XML 문서에서 Complete 키에 "true"를 지정한 후 다음 최소 콘텐츠를 전달하십시오.

```
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Complete</key><true/>
</dict>
</plist>
```

필요하지 않은 모든 XML 요소는 무시되며 출력 문서에 리턴되지 않습니다.

프로시저 실행 시 전체 XML 입력 문서가 스토어드 프로시저의 *xml\_output* 매개변수로 리턴됩니다. 리턴된 XML 문서에는 모든 가능한 필수 매개변수 및 선택적 매개변수에 대한 문서 유형 및 섹션이 포함됩니다. 리턴된 XML 문서에는 필수는 아니지만 클라이언트 응용프로그램에서 문서를 렌더링할 때 일반적으로 필요한 기타 항목(예: 표시 이름, 힌트 및 문서 로케일)도 포함됩니다.

플랫폼에 구애받지 않는 방식으로 XML 문서를 렌더링하고 수정한 후에는 동일한 스토어드 프로시저를 실행하고 수정된 XML 문서를 입력으로 전달할 수 있습니다.

---

## XML 출력 문서

일반 SQL API 스토어드 프로시저에서 출력으로 리턴된 XML 문서는 일반 세트의 항목을 공유합니다.

최소한 *xml\_output* 매개변수로 리턴된 XML 문서에는 다음과 같은 필수 키 값 쌍이 포함됩니다.

```
<?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 출력 문서의 항목은 중첩된 사전을 사용하여 그룹화될 수 있습니다. XML 출력 문서의 각 항목은 각각 한 가지 정보에 대해 설명합니다. 항목은 값, 표시 이름 및 힌트로 구성됩니다. 선택적으로 표시 장치도 제공될 수 있습니다. 표시 이름, 힌트 및 표시 장치는 언어에 따라 변경되며 *requested\_locale* 매개변수 값으로 지정된 언어(또는 요청된 로케일이 아직 지원되지 않는 경우에는 디폴트)로 변환됩니다. 일반적으로 항목의 구조는 다음 예와 유사합니다.

```
<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 Data Server에는 모든 데이터 서버에 적용되는 일부 키워드와 특정 데이터 서버에만 해당하는 키워드를 포함하는 일반 매개변수 문서가 포함됩니다. 데이터 서버가 새 키워드를 추가하거나 제거할 때마다 버전 번호(모든 데이터 서버용)가 증가됩니다. 변경 사항에 따라 기본 버전 번호가 증가하고 보조 버전 번호는 0으로 설정되거나 보조 버전 번호만 증가됩니다.

XML 출력 문서는 UTF-8로 생성되며 영어 문자만 포함합니다.

## 출력 필터링을 위한 XPath 표현식

XPath 표현식을 사용하여 일반 SQL API 스토어드 프로시저에서 리턴된 XML 출력을 필터링할 수 있습니다.

출력을 필터링하려면 프로시저의 *xml\_filter* 매개변수에 유효한 XPath 쿼리 문자열을 지정하십시오. 다음과 같은 제한사항이 지정한 XPath 표현식에 적용됩니다.

- XPath 표현식은 한 개의 값을 참조해야 합니다.
- XPath 표현식은 항상 루트 노드의 절대 경로여야 합니다. 예를 들어 다음 경로 표현식을 사용할 수 있습니다. */*, *nodename*, *..*, ... 다음 표현식은 사용할 수 없습니다. *//* 및 *@*
- 허용되는 유일한 술어는 [*path*='value'] 및 [*n*]입니다.
- 허용되는 유일한 축은 *following-sibling*입니다.
- XPath 표현식은 다음 중 하나로 끝나야 하며 필요한 경우 술어 [1]을 추가해야 합니다. *following-sibling::string*, *following-sibling::data*, *following-sibling::date*, *following-sibling::real* 또는 *following-sibling::integer*.
- XPath 표현식 끝에 축이 있는 경우를 제외하고는 뒤에 *::dict*, *::string*, *::data*, *::date*, *::real* 또는 *::integer*가 와야 하며 필요한 경우 술어 [1]을 추가해야 합니다.
- 지원되는 유일한 XPath 연산자는 *=*입니다.



- XPath 표현식은 함수, 이름 스페이스, 처리 명령어 또는 주석을 포함할 수 없습니다.

팁: 스토어드 프로시저가 완료 모드로 작동하는 경우 필터링을 적용하지 마십시오. 적용하면 SQLCODE (+20458)가 발생합니다.

*xml\_output* 매개변수로 리턴된 XML 문서 처리를 잘 제어하기 위해 DB2 pureXML®에서 사용할 수 있는 XMLPARSE 함수를 사용할 수 있습니다.

## 예

다음 XPath 표현식은 XML 출력 문서에서 Data Server Product Version 키 값을 선택합니다.

```
/plist/dict/key[.='Data Server Product Version']following-sibling::string[1]
```

프로시저는 8.1.0.356 문자열을 *xml\_output* 매개변수로 리턴합니다. 따라서 프로시저 호출은 XML 문서가 아닌 한 개의 값을 리턴합니다.

---

## XML 메시지 문서

일반 SQL API 스토어드 프로시저에서 내부 처리 오류 또는 유효하지 않은 매개변수가 발생하면 데이터 서버는 SQLCODE 및 해당 SQL 메시지를 호출자에 리턴합니다. 이 경우 프로시저는 경고 상황에 대한 자세한 정보가 포함된 XML 메시지 문서를 *xml\_message* 매개변수로 리턴합니다.

XML 메시지 문서는 다음과 같은 일반 구조로 되어 있습니다.

```
<?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 메시지 문서는 UTF-8로 생성되며 영어 문자만 포함합니다.



## 예

다음 예에서 GET\_MESSAGE 프로시저 호출은 SQL 경고를 발생시킵니다.

```
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
```

매개변수 7(*xml\_message*)로 리턴된 XML 문서에는 다음 콘텐츠가 포함됩니다.

```
<?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>
```

간단한 메시지 텍스트 키 값은 경고에 대한 추가 정보를 제공합니다.

---

## CANCEL\_WORK 프로시저 - 작업 취소

CANCEL\_WORK 스토어드 프로시저는 특정 활동(예: SQL문) 또는 연결된 응용프로그램의 모든 활동을 취소합니다.

특정 활동을 취소하려면 응용프로그램 핸들, 작업 단위(UOW) ID 및 취소하려는 활동의 활동 ID를 전달합니다. 연결된 응용프로그램에 대한 모든 활동을 취소하려면 응용프로그램 핸들로 전달합니다. 취소된 활동에 연관된 모든 변경사항이 롤백됩니다.

### 구문

```
►► CANCEL_WORK (—major_version—, —minor_version—, —requested_locale—, —————►
► —xml_input—, —xml_filter—, —xml_output—, —xml_message—) —————►►
```

스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *major\_version*

기본 문서 버전을 나타내는 INTEGER 유형의 입력 및 출력 인수입니다. 입력에서 이 인수는 호출자가 프로시저에서 매개변수로 전달되는 XML 문서를 지원하는 기본 문서 버전을 나타냅니다(*xml\_input*, *xml\_output* 및 *xml\_message*에 대해서는 매개변수 설명 참조). 프로시저는 지정된 버전의 모든 XML 문서를 처리하거나 버전이 유효하지 않은 경우 오류를 리턴합니다(+20458). 출력에서 이 매개변수는 프로시저에서 지원되는 가장 높은 기본 문서 버전을 지정합니다. 지원되는 가장 높은 문서 버전을 판별하려면 이 입력 매개변수 및 모든 필수 매개변수에 대해 NULL을 지정하십시오.

지원되는 버전: 1

### *minor\_version*

보조 문서 버전을 나타내는 INTEGER 유형의 입력 및 출력 인수입니다. 입력에서 이 인수는 호출자가 이 프로시저에서 매개변수로 전달되는 XML 문서를 지원하는 보조 문서 버전을 지정합니다(*xml\_input*, *xml\_output* 및 *xml\_message*에 대해서는 매개변수 설명 참조). 프로시저는 지정된 버전의 모든 XML 문서를 처리하거나 버전이 유효하지 않은 경우 오류를 리턴합니다. 출력에서 이 매개변수는 지원되는 가장 높은 기본 문서 버전에 대해 지원되는 가장 높은 보조 문서 버전을 나타냅니다. 지원되는 가장 높은 문서 버전을 판별하려면 이 입력 매개변수 및 모든 필수 매개변수에 대해 NULL을 지정하십시오.

지원되는 버전: 0

### *requested\_locale*

로케일을 지정하는 VARCHAR(33) 유형의 입력 인수입니다. 지정된 언어가 서버에서 지원되는 경우 변환된 콘텐츠가 *xml\_output* 및 *xml\_message* 매개변수로 리턴됩니다. 그 이외의 경우에는 콘텐츠가 디폴트 언어로 리턴됩니다. 언어와 가능한 지역 정보만 로케일에서 사용됩니다. 로케일은 숫자를 형식화하거나 문서 인코딩을 변경하는 데 사용되지 않습니다. 예를 들어 주요 이름 및 값은 변환되지 않습니다. XML 출력 및 XML 메시지 문서에서는 힌트를 위한 텍스트, 표시 이름 및 각 항목의 표시 부분만 변환됩니다. 호출자는 항상 요청된 언어를 XML 출력 문서에서 사용된 언어와 비교해야 합니다(XML 출력 문서의 문서 로케일 항목 참조).

현재 *requested\_locale*에 대해서는 en\_US 값만 지원됩니다.

### *xml\_input*

프로시저의 입력 값을 포함하는 XML 입력 문서(UTF-8로 인코딩)를 지정하는 BLOB(32MB) 유형의 입력 인수입니다.

이 프로시저의 경우 XML 입력 문서는 응용프로그램 핸들을 지정해야 합니다. 특정 활동을 취소하려면 XML 입력 문서가 작업 단위(UOW) ID 및 활동 ID를 식별하는 선택적 매개변수로 지정해야 합니다. 이 스토어드 프로시저의 전체 XML 입력 문서는 다음 문서와 유사합니다.

```
<?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>
  <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>
</dict>
</plist>
```

스토어드 프로시저가 실행 중인 응용프로그램의 응용프로그램 핸들을 지정하는 경우 프로시저는 경고를 리턴합니다(SQL20458).

### *xml\_filter*

유효한 XPath 쿼리 문자열을 지정하는 BLOB(4K) 유형의 입력 인수입니다. XML 출력 문서에서 한 개의 값만 검색하려는 경우에는 필터를 사용하십시오. 자세한 정보는 XPath 필터링을 설명하는 주제를 참조하십시오.

다음 예에서는 데이터 서버 제품 버전 값이 XML 출력 문서에서 선택됩니다. /plist/dict/key[.='Data Server Product Version']/following-sibling::string 키 뒤에 지정한 동위가 오지 않으면 오류가 리턴됩니다.

### *xml\_output*

UTF-8로 전체 XML 출력 문서를 리턴하는 BLOB(32MB) 유형의 출력 매개변수입니다. 필터가 지정된 경우 이 매개변수는 문자열 값을 리턴합니다. 스토어드 프로시저가 전체 출력 문서를 리턴할 수 없는 경우(예: 처리 오류가 발생하여 SQL 경고 또는 오류가 표시되는 경우) 이 매개변수가 NULL로 설정됩니다.

*major\_version* 및 *minor\_version*에 대해 지정한 값으로 XML 출력이 판별됩니다.

기본 버전	보조 버전	<i>xml_output</i> 값
NULL	NULL	NULL
1	0	프로시저에서 취소를 시도한 활동의 상태입니다.

프로시저가 완료 모드에서 작동하는 경우 이 매개변수는 수정할 수 있으며 프로시저를 다시 *xml\_input* 매개변수로 되돌릴 수 있는 XML 문서를 리턴합니다. 이 접근 방식은 유효한 XML 입력 문서를 작성하는 프로그램화된 방법입니다. 자세한 정보에 대해서는 완료 모드에 대한 주제를 참조하십시오.

### *xml\_message*

SQL 경고 조건에 대한 자세한 정보를 제공하는 데이터 서버 메시지 유형의 전체 XML 출력 문서를 UTF-8로 리턴하는 BLOB(64K) 형식의 출력 매개변수입니다. 프로시저 호출로 인해 추가 정보가 XML 메시지 출력 문서로 리턴됨을 나타내는 SQL 경고가 발생할 때 이 문서가 리턴됩니다. 경고 메시지 내용이 추가 정보가 리턴됨을 나타내지 않는 경우에는 이 매개변수가 NULL로 설정됩니다.

## 권한 부여

- SYSADM 또는 DBADM 권한
- CANCEL\_WORK 프로시저에 대한 EXECUTE 특권

## 예

예 1: 프로시저의 지원되는 가장 높은 버전을 리턴합니다.

```
db2 "call sysproc.cancel_work(null,null,null,null,null,?,?)"
```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

```
-----  
매개변수 이름 : MAJOR_VERSION  
매개변수 값 : 1
```

```
매개변수 이름 : MINOR_VERSION  
매개변수 값 : 0
```

```
매개변수 이름 : XML_OUTPUT  
매개변수 값 : -
```

매개변수 이름 : XML\_MESSAGE  
매개변수 값 : -

리턴 상태 = 0

예 2: 특정 활동을 취소합니다.

```
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,?,?)"
```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

-----  
매개변수 이름 : MAJOR\_VERSION  
매개변수 값 : 1

매개변수 이름 : MINOR\_VERSION  
매개변수 값 : 0

매개변수 이름 : XML\_OUTPUT  
매개변수 값 : x'3C3F78...'

매개변수 이름 : XML\_MESSAGE  
매개변수 값 : -

리턴 상태 = 0

CANCEL\_WORK 프로시저에서 활동을 취소할 수 있는 경우 XML 출력 문서에는 다음 콘텐츠가 포함됩니다.

```
<?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>
```

예 : 응용 프로그램을 취소합니다.

```
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,?,?)"
```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

-----  
매개변수 이름 : MAJOR\_VERSION  
매개변수 값 : 1

매개변수 이름 : MINOR\_VERSION  
매개변수 값 : 0

매개변수 이름 : XML\_OUTPUT  
매개변수 값 : x'3C3F78...'

매개변수 이름 : XML\_MESSAGE  
매개변수 값 : -

리턴 상태 = 0

CANCEL\_WORK 프로시저에서 응용프로그램을 취소할 수 있는 경우 XML 출력 문서에는 다음 콘텐츠가 포함됩니다.

```
<?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>
```

예 3: 성공적인 작업 취소 메시지 값을 리턴하는 필터를 지정합니다.

```
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]'),?,?)"
```

이 쿼리의 출력 예는 다음과 같습니다.

```
출력 매개변수 값
-----
매개변수 이름 : MAJOR_VERSION
매개변수 값 : 1

매개변수 이름 : MINOR_VERSION
매개변수 값 : 0

매개변수 이름 : XML_OUTPUT
매개변수 값 : x'3C3F78...'

매개변수 이름 : XML_MESSAGE
매개변수 값 : -

리턴 상태 = 0
```

다음 값이 *xml\_output*에 대해 리턴됩니다.

```
"The application has been cancelled successfully"
```

---

## GET\_CONFIG 프로시저 - 구성 데이터 가져오기

GET\_CONFIG 스토어드 프로시저는 특정 인스턴스에 대해 설정된 데이터베이스 구성, 데이터베이스 관리 프로그램 구성 및 레지스트리 변수를 검색합니다.

파티션된 데이터베이스 환경에서 이 프로시저는 데이터베이스 구성 및 레지스트리 변수 설정을 모든 파티션에서 검색합니다.

### 구문

```
▶▶ GET_CONFIG(—major_version—, —minor_version—, —requested_locale—, —————▶
▶—xml_input—, —xml_filter—, —xml_output—, —xml_message—)—————▶▶
```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *major\_version*

기본 문서 버전을 나타내는 INTEGER 유형의 입력 및 출력 인수입니다. 입력에서 이 인수는 호출자가 프로시저에서 매개변수로 전달되는 XML 문서를 지원하는 기본 문서 버전을 나타냅니다(*xml\_input*, *xml\_output* 및 *xml\_message*에 대해서는 매개변수 설명 참조). 프로시저는 지정된 버전의 모든 XML 문서를 처리하거나 버전이 유효하지 않은 경우 오류를 리턴합니다(+20458). 출력에서 이 매개변수는 프로시저에서 지원되는 가장 높은 기본 문서 버전을 지정합니다. 지원되는 가장 높은 문서 버전을 판별하려면 이 입력 매개변수 및 모든 필수 매개변수에 대해 NULL을 지정하십시오.



지원되는 버전: 1 및 2

#### *minor\_version*

보조 문서 버전을 나타내는 INTEGER 유형의 입력 및 출력 인수입니다. 입력에서 이 인수는 호출자가 이 프로시저에서 매개변수로 전달되는 XML 문서를 지원하는 보조 문서 버전을 지정합니다(*xml\_input*, *xml\_output* 및 *xml\_message*에 대해서는 매개변수 설명 참조). 프로시저는 지정된 버전의 모든 XML 문서를 처리하거나 버전이 유효하지 않은 경우 오류를 리턴합니다. 출력에서 이 매개변수는 지원되는 가장 높은 기본 문서 버전에 대해 지원되는 가장 높은 보조 문서 버전을 나타냅니다. 지원되는 가장 높은 문서 버전을 판별하려면 이 입력 매개변수 및 모든 필수 매개변수에 대해 NULL을 지정하십시오.

지원되는 버전: 0

#### *requested\_locale*

로케일을 지정하는 VARCHAR(33) 유형의 입력 인수입니다. 지정된 언어가 서버에서 지원되는 경우 변환된 콘텐츠가 *xml\_output* 및 *xml\_message* 매개변수로 리턴됩니다. 그 이외의 경우에는 콘텐츠가 디폴트 언어로 리턴됩니다. 언어와 가능한 지역 정보만 로케일에서 사용됩니다. 로케일은 숫자를 형식화하거나 문서 인코딩을 변경하는 데 사용되지 않습니다. 예를 들어 주요 이름 및 값은 변환되지 않습니다. XML 출력 및 XML 메시지 문서에서는 힌트를 위한 텍스트, 표시 이름 및 각 항목의 표시 부분만 변환됩니다. 호출자는 항상 요청된 언어를 XML 출력 문서에서 사용된 언어와 비교해야 합니다(XML 출력 문서의 문서 로케일 항목 참조).

현재 *requested\_locale*에 대해서는 en\_US 값만 지원됩니다.

#### *xml\_input*

현재 이 프로시저에서는 입력을 허용하지 않습니다. 이 매개변수에 대해 NULL을 지정하십시오. 지정하지 않으면 입력이 유효하지 않음을 나타내는 오류(+20458)가 표시됩니다.

#### *xml\_filter*

유효한 XPath 쿼리 문자열을 지정하는 BLOB(4K) 유형의 입력 인수입니다. XML 출력 문서에서 한 개의 값만 검색하려는 경우에는 필터를 사용하십시오. 자세한 정보는 XPath 필터링을 설명하는 주제를 참조하십시오.

다음 예에서는 데이터 서버 제품 버전 값이 XML 출력 문서에서 선택됩니다. `/plist/dict/key[.='Data Server Product Version']/following-sibling::string`. 키 뒤에 지정한 동위가 오지 않으면 오류가 리턴됩니다.

#### *xml\_output*

UTF-8로 전체 XML 출력 문서를 리턴하는 BLOB(32MB) 유형의 출력 매개변수입니다. 필터가 지정된 경우 이 매개변수는 문자열 값을 리턴합니다. 스토어드 프로시저가 전체 출력 문서를 리턴할 수 없는 경우(예: 처리 오류가 발생하여 SQL 경고 또는 오류가 표시되는 경우) 이 매개변수가 NULL로 설정됩니다.

*major\_version* 및 *minor\_version*에 대해 지정한 값으로 XML 출력이 판별됩니다.

기본 버전	보조 버전	<i>xml_output</i> 값
NULL	NULL	NULL
1	0	데이터베이스 관리 프로그램 및 데이터베이스 구성 매개변수 및 레지스트리 변수와 해당 값이 포함됩니다.
2	0	데이터베이스 관리 프로그램 및 데이터베이스 구성 매개변수가 범주로 그룹화됩니다. 각 매개변수에 대해 매개변수의 갱신 가능 여부가 표시됩니다. 인스턴스의 레지스트리 변수 및 값 세트도 리턴됩니다.

프로시저가 완료 모드에서 작동하는 경우 이 매개변수는 수정할 수 있으며 프로시저를 다시 *xml\_input* 매개변수로 되돌릴 수 있는 XML 문서를 리턴합니다. 이 접근 방식은 유효한 XML 입력 문서를 작성하는 프로그램화된 방법입니다. 자세한 정보에 대해서는 완료 모드에 대한 주제를 참조하십시오.

#### *xml\_message*

SQL 경고 조건에 대한 자세한 정보를 제공하는 데이터 서버 메시지 유형의 전체 XML 출력 문서를 UTF-8로 리턴하는 BLOB(64K) 형식의 출력 매개변수입니다. 프로시저 호출로 인해 추가 정보가 XML 메시지 출력 문서로 리턴됨을 나타내는 SQL 경고가 발생할 때 이 문서가 리턴됩니다. 경고 메시지 내용이 추가 정보가 리턴됨을 나타내지 않는 경우에는 이 매개변수가 NULL로 설정됩니다.

### 권한 부여

- SYSADM 또는 DBADM 권한
- GET\_CONFIG 프로시저에 대한 EXECUTE 특권

### 예

예 1: 프로시저의 최신 버전을 리턴합니다.

```
db2 "call sysproc.get_config(null,null,null,null,null,?,?)"
```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

```
-----
매개변수 이름 : MAJOR_VERSION
매개변수 값 : 2
```

```
매개변수 이름 : MINOR_VERSION
매개변수 값 : 0
```

```
매개변수 이름 : XML_OUTPUT
매개변수 값 : -
```

매개변수 이름 : XML\_MESSAGE  
매개변수 값 : -

리턴 상태 = 0

예 2: 범주로 그룹화된 데이터베이스 및 데이터베이스 관리 프로그램 구성 매개변수를 리턴합니다.

```
db2 "call sysproc.get_config(2,0,'en_US',null, null, ?,?)"
```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

-----  
매개변수 이름 : MAJOR\_VERSION  
매개변수 값 : 2

매개변수 이름 : MINOR\_VERSION  
매개변수 값 : 0

매개변수 이름 : XML\_OUTPUT  
매개변수 값 : x'3C3F78.....'

매개변수 이름 : XML\_MESSAGE  
매개변수 값 : -

리턴 상태 = 0

출력 XML 문서에는 다음 콘텐츠가 포함됩니다.

```
<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>
        <key>Updatable</key><string>No</string>
        <key>Hint</key><string></string>
      </dict>
    </dict>
  </dict>
</plist>
```

```

</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>
운영 체제 스케줄러가 에이전트 및 기타 데이터베이스 관리 프로그램
인스턴스 프로세스 및 스레드에 부여되는 우선순위를 지정합니다.
이 매개변수를 변경한 다음 응용프로그램 리바인드를 고려하십시오.
</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>
        비유니코드 데이터베이스에서 유니코드 테이블에 사용될
        조합 시퀀스를 지정합니다. 이 매개변수가 설정될 때까지
        비유니코드 데이터베이스에
        유니코드 테이블 및 루틴을 작성할 수 없습니다. 이 매개변수가
        설정되면 변경하거나 재설정할 수 없습니다. 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>1</string>
      <key>Hint</key><string></string>
    </dict>
    <key>Hint</key><string></string>
  </dict>
  .
  .
  .
</dict>
<key>Hint</key><string></string>
</dict>
</plist>

```

예 3: 데이터베이스 및 데이터베이스 관리 프로그램 구성 매개변수를 리턴합니다.

```
db2 "call sysproc.get_config(1,0,'en_US',null, null, ?,?)"
```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

```

-----
매개변수 이름 : MAJOR_VERSION
매개변수 값 : 1

```

```

매개변수 이름 : MINOR_VERSION
매개변수 값 : 0

```

```

매개변수 이름 : XML_OUTPUT
매개변수 값 : x'3C3F78.....'

```

```

매개변수 이름 : XML_MESSAGE
매개변수 값 : -

```

리턴 상태 = 0

XML 출력 문서에는 예 2와 유사한 콘텐츠가 포함되지만 구성 매개변수를 범주로 그룹화하지는 않습니다.

예 4: 함수에서 프로시저를 호출합니다.

```

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 프로시저 - 메시지 텍스트 가져오기

GET\_MESSAGE 프로시저는 SQLCODE에 대한 간략한 메시지 텍스트, 자세한 메시지 텍스트 및 SQLSTATE를 리턴합니다.

### 구문

```

▶▶ GET_MESSAGE(—major_version—, —minor_version—, —requested_locale—, —
▶—xml_input—, —xml_filter—, —xml_output—, —xml_message—)

```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *major\_version*

기본 문서 버전을 나타내는 INTEGER 유형의 입력 및 출력 인수입니다. 입력에서 이 인수는 호출자가 프로시저에서 매개변수로 전달되는 XML 문서를 지원하는 기본 문서 버전을 나타냅니다(*xml\_input*, *xml\_output* 및 *xml\_message*에 대해서는 매개변수 설명 참조). 프로시저는 지정된 버전의 모든 XML 문서를 처리하거나 버전이 유효하지 않은 경우 오류를 리턴합니다(+20458). 출력에서 이 매개변수는 프로시저에서 지원되는 가장 높은 기본 문서 버전을 지정합니다. 지원되는 가장 높은 문서 버전을 판별하려면 이 입력 매개변수 및 모든 필수 매개변수에 대해 NULL을 지정하십시오.

*xml\_input* 매개변수의 XML 문서가 문서 유형 기본 버전 키를 지정하는 경우 해당 키 값은 *major\_version* 매개변수로 제공한 값과 동등해야 하며 그렇지 않으면 오류(+20458)가 발생합니다.

지원되는 버전: 1 및 2

### *minor\_version*

보조 문서 버전을 나타내는 INTEGER 유형의 입력 및 출력 인수입니다. 입력에서 이 인수는 호출자가 이 프로시저에서 매개변수로 전달되는 XML 문서를 지원하는 보조 문서 버전을 지정합니다(*xml\_input*, *xml\_output* 및 *xml\_message*에 대해서는 매개변수 설명 참조). 프로시저는 지정된 버전의 모든 XML 문서를 처리하거나 버전이 유효하지 않은 경우 오류를 리턴합니다. 출력에서 이 매개변수는 지원되는 가장 높은 기본 문서 버전에 대해 지원되는 가장 높은 보조 문서 버전을 나타냅니다. 지원되는 가장 높은 문서 버전을 판별하려면 이 입력 매개변수 및 모든 필수 매개변수에 대해 NULL을 지정하십시오.

*xml\_input* 매개변수의 XML 문서가 문서 유형 보조 버전 키를 지정하는 경우 해당 키 값은 *minor\_version* 매개변수로 제공한 값과 동등해야 하며 그렇지 않으면 오류(+20458)가 발생합니다.

지원되는 버전: 0

### *requested\_locale*

로케일을 지정하는 VARCHAR(33) 유형의 입력 인수입니다. 지정된 언어가 서버에서 지원되는 경우 변환된 콘텐츠가 *xml\_output* 및 *xml\_message* 매개변수로 리턴됩니다. 그 이외의 경우에는 콘텐츠가 디폴트 언어로 리턴됩니다. 언어와 가능한 지역 정보만 로케일에서 사용됩니다. 로케일은 숫자를 형식화하거나 문서 인코딩을 변경하는 데 사용되지 않습니다. 예를 들어 주요 이름 및 값은 변환되지 않습니다. XML 출력 및 XML 메시지 문서에서는 힌트를 위한 텍스트, 표시 이름 및 각 항목의 표시 부분만 변환됩니다. 호출자는 항상 요청된 언어를 XML 출력 문서에서 사용된 언어와 비교해야 합니다(XML 출력 문서의 문서 로케일 항목 참조).

현재 *requested\_locale*에 대해서는 en\_US 값만 지원됩니다.

### *xml\_input*

프로시저의 입력 값을 포함하는 XML 입력 문서(UTF-8로 인코딩)를 지정하는 BLOB(32MB) 유형의 입력 인수입니다.

이 프로시저의 경우 XML 입력 문서에는 SQLCODE가 포함되고 다음 형식을 사용합니다.

```
<?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

유효한 XPath 쿼리 문자열을 지정하는 BLOB(4K) 유형의 입력 인수입니다. XML 출력 문서에서 한 개의 값만 검색하려는 경우에는 필터를 사용하십시오. 자세한 정보는 XPath 필터링을 설명하는 주제를 참조하십시오.

다음 예에서는 SQLSTATE에 대한 값이 XML 출력 문서에서 선택됩니다. `/plist/dict/key[.="SQLSTATE"]/following-sibling::dict[1]/key[.="Value"]/following-sibling::string[1]`. 키 뒤에 지정된 동위가 오지 않으면 오류가 리턴됩니다.

### xml\_output

UTF-8로 전체 XML 출력 문서를 리턴하는 BLOB(32MB) 유형의 출력 매개변수입니다. 필터가 지정된 경우 이 매개변수는 문자열 값을 리턴합니다. 스토어드 프로시저가 전체 출력 문서를 리턴할 수 없는 경우(예: 처리 오류가 발생하여 SQL 경고 또는 오류가 표시되는 경우) 이 매개변수가 NULL로 설정됩니다.

`major_version` 및 `minor_version`에 대해 지정한 값으로 XML 출력이 판별됩니다.

기본 버전	보조 버전	xml_output 값
NULL	NULL	NULL
1	0	xml_input에서 전달된 해당 SQLCODE에 대한 간략한 텍스트 메시지 및 SQLCODE가 리턴됩니다.
2	0	xml_input에서 전달된 해당 SQLCODE에 대한 간략한 텍스트 메시지, 자세한 텍스트 메시지 및 SQLCODE가 리턴됩니다.

프로시저가 완료 모드에서 작동하는 경우 이 매개변수는 수정할 수 있으며 프로시저를 다시 `xml_input` 매개변수로 되돌릴 수 있는 XML 문서를 리턴합니다. 이 접근 방식은 유효한 XML 입력 문서를 작성하는 프로그램화된 방법입니다. 자세한 정보에 대해서는 완료 모드에 대한 주제를 참조하십시오.

### xml\_message

SQL 경고 조건에 대한 자세한 정보를 제공하는 데이터 서버 메시지 유형의 전체 XML 출력 문서를 UTF-8로 리턴하는 BLOB(64K) 형식의 출력 매개변수입니다. 프로시저 호출로 인해 추가 정보가 XML 메시지 출력 문서로 리턴됨을 나타내는 SQL 경고가 발생할 때 이 문서가 리턴됩니다. 경고 메시지 내용이 추가 정보가 리턴됨을 나타내지 않는 경우에는 이 매개변수가 NULL로 설정됩니다.

## 권한 부여

- SYSADM 또는 DBADM 권한
- GET\_MESSAGE 프로시저에 대한 EXECUTE 특권

## 예

예 1: 프로시저의 지원되는 가장 높은 버전을 리턴합니다.

```
db2 "call sysproc.get_message(null,null,null,null,null,?,?)"
```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

```
-----  
매개변수 이름 : MAJOR_VERSION  
매개변수 값 : 2
```

```
매개변수 이름 : MINOR_VERSION  
매개변수 값 : 0
```

```
매개변수 이름 : XML_OUTPUT  
매개변수 값 : -
```

```
매개변수 이름 : XML_MESSAGE  
매개변수 값 : -
```

리턴 상태 = 0

예 2: SQL1034의 간략한 텍스트 메시지 및 자세한 텍스트 메시지를 리턴하는 getmsglong.sql 스크립트를 실행합니다.

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, ?, ?)@
```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

```
-----  
매개변수 이름 : MAJOR_VERSION  
매개변수 값 : 2
```

```
매개변수 이름 : MINOR_VERSION  
매개변수 값 : 0
```

```
매개변수 이름 : XML_OUTPUT  
매개변수 값 : x'3C3F786D6C20766572.....'
```

매개변수 이름 : XML\_MESSAGE  
매개변수 값 : -

리턴 상태 = 0

출력 XML 문서에는 다음 콘텐츠가 포함됩니다.

```
<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 데이터베이스가 손상되었습니다. 데이터베이스를 처리하는 모든 응용프로그램이
중지되었습니다.
</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 데이터베이스가 손상되었습니다. 데이터베이스를 처리하는
모든 응용프로그램이
</string>
      <string> 중지되었습니다.</string>
      <string></string>
      <string>Explanation: </string>
      <string></string>
      <string>
데이터베이스가 손상되었습니다. 복구될 때까지
</string>
      <string>
사용할 수 없습니다. 데이터베이스에 연결된 모든 응용프로그램들의 연결이 끊어지고
</string>
      <string>
데이터베이스에서 응용프로그램을 실행 중인
모든 프로세스가

```

```

</string>
<string>중지되었습니다.</string>
<string></string>
<string>명령을 처리할 수 없습니다.</string>
<string></string>
<string>User response: </string>
<string></string>
<string>
RESTART DATABASE 명령을 발행하여 데이터베이스를 복구하십시오. RESTART 명령에
</string>
<string>
계속 실패하는 경우, 백업으로 데이터베이스를 리스토어할 수도
</string>
<string>
있습니다. 파티션된 데이터베이스 서버 환경에서
</string>
<string>
백업으로 데이터베이스를 리스토어하기 전에 syslog를 확인하여
</string>
<string>
노드 또는 통신 실패로 인해 RESTART 명령에 실패했는지 알아보십시오. 이 경우,
</string>
<string>
데이터베이스 관리 프로그램이 시작되어 실행 중이고, 모든 노드간에
</string>
<string>
통신이 가능한지 확인한 다음 재시작 명령을 다시 제출하십시오.
</string>
<string></string>
<string>
를 포워드 처리 중에 이 오류가 발생하면, 데이터베이스를 백업으로
</string>
<string>
리스토어하고 를 포워드를 다시 수행해야 합니다.
</string>
<string></string>
<string>
파티션된 데이터베이스 환경에서 데이터베이스 RESTART 명령은
</string>
<string>
각 노드 단위로 실행됩니다. 모든 노드에서 데이터베이스가 재시작되었는지 확인하려면,
</string>
<string>다음의 명령을 사용하십시오. </string>
<string></string>
<string>db2_all db2 restart database</string>
<string><database_name></string>
<string></string>
<string>
모든 인다우트 트랜잭션이 해결되었는지 확인하려면,
</string>
<string>이 명령을 여러 번 실행해야 할 것입니다.</string>
<string></string>
<string>
샘플 데이터베이스를 설치하는 경우, 이를 삭제하고
</string>
<string>샘플 데이터베이스를 다시 설치하십시오.</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>

```

예 3: SQL1034에 대한 간략한 텍스트 메시지만 리턴하는 getmsgshort.sql 스크립트를 실행합니다.

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, ? , ?)@
```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

-----  
매개변수 이름 : MAJOR\_VERSION  
매개변수 값 : 2

매개변수 이름 : MINOR\_VERSION  
매개변수 값 : 0

매개변수 이름 : XML\_OUTPUT  
매개변수 값 : x'3C3F786D6C20766572.....'

매개변수 이름 : XML\_MESSAGE  
매개변수 값 : -

리턴 상태 = 0

SQL20460W "SYSPROC.GET\_MESSAGE" 프로시저는 "1" 매개변수에 대해, 지정된 버전 "1"보다 더 높은 버전인 "2"를 지원합니다.

출력 XML 문서에는 다음 콘텐츠가 포함됩니다.

```
<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 데이터베이스가 손상되었습니다. 데이터베이스를 처리하는
모든 응용프로그램이 중지되었습니다.
  </string>
  </dict>
  </dict>
</plist>
```

```

    <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>

```

예 4: SQL1034의 SQLSTATE를 리턴하는 필터를 지정합니다.

```

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]'), ? , ?)"

```

이 쿼리의 출력 예는 다음과 같습니다.

```

출력 매개변수 값
-----
매개변수 이름 : MAJOR_VERSION
매개변수 값 : 2

매개변수 이름 : MINOR_VERSION
매개변수 값 : 0

매개변수 이름 : XML_OUTPUT
매개변수 값 : x'203538303331'

매개변수 이름 : XML_MESSAGE
매개변수 값 : -

리턴 상태 = 0

```

다음 값이 *xml\_output*에 대해 리턴됩니다.

```
58031
```

예 5: 함수에서 프로시저를 호출합니다.

```

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 프로시저 - 시스템 정보 가져오기

GET\_SYSTEM\_INFO 프로시저는 시스템에 대한 정보, 현재 인스턴스, 설치된 데이터 서버 제품, 환경 변수, 사용 가능한 CPU 및 기타 시스템 정보가 포함된 데이터 서버에 대한 정보를 리턴합니다.

### 구문

```

▶▶ GET_SYSTEM_INFO(—major_version—, —minor_version—, —requested_locale—, —————▶
▶ —xml_input—, —xml_filter—, —xml_output—, —xml_message—) —————▶▶

```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### major\_version

기본 문서 버전을 나타내는 INTEGER 유형의 입력 및 출력 인수입니다. 입력에서 이 인수는 호출자가 프로시저에서 매개변수로 전달되는 XML 문서를 지원하는 기본 문서 버전을 나타냅니다(*xml\_input*, *xml\_output* 및 *xml\_message*에 대해서는 매개변수 설명 참조). 프로시저는 지정된 버전의 모든 XML 문서를 처리하거나 버전이 유효하지 않은 경우 오류를 리턴합니다(+20458). 출력에서 이 매개변수는 프로시저에서 지원되는 가장 높은 기본 문서 버전을 지정합니다. 지원되는 가장 높은 문서 버전을 판별하려면 이 입력 매개변수 및 모든 필수 매개변수에 대해 NULL을 지정하십시오.

*xml\_input* 매개변수의 XML 문서가 문서 유형 기본 버전 키를 지정하는 경우 해당 키 값은 *major\_version* 매개변수로 제공한 값과 동등해야 하며 그렇지 않으면 오류(+20458)가 발생합니다.

지원되는 버전: 1

### *minor\_version*

보조 문서 버전을 나타내는 INTEGER 유형의 입력 및 출력 인수입니다. 입력에서 이 인수는 호출자가 이 프로시저에서 매개변수로 전달되는 XML 문서를 지원하는 보조 문서 버전을 지정합니다(*xml\_input*, *xml\_output* 및 *xml\_message*에 대해서는 매개변수 설명 참조). 프로시저는 지정된 버전의 모든 XML 문서를 처리하거나 버전이 유효하지 않은 경우 오류를 리턴합니다. 출력에서 이 매개변수는 지원되는 가장 높은 기본 문서 버전에 대해 지원되는 가장 높은 보조 문서 버전을 나타냅니다. 지원되는 가장 높은 문서 버전을 판별하려면 이 입력 매개변수 및 모든 필수 매개변수에 대해 NULL을 지정하십시오.

지원되는 버전: 0

### *requested\_locale*

로케일을 지정하는 VARCHAR(33) 유형의 입력 인수입니다. 지정된 언어가 서버에서 지원되는 경우 변환된 콘텐츠가 *xml\_output* 및 *xml\_message* 매개변수로 리턴됩니다. 그 이외의 경우에는 콘텐츠가 디폴트 언어로 리턴됩니다. 언어와 가능한 지역 정보만 로케일에서 사용됩니다. 로케일은 숫자를 형식화하거나 문서 인코딩을 변경하는 데 사용되지 않습니다. 예를 들어 주요 이름 및 값은 변환되지 않습니다. XML 출력 및 XML 메시지 문서에서는 힌트를 위한 텍스트, 표시 이름 및 각 항목의 표시 부분만 변환됩니다. 호출자는 항상 요청된 언어를 XML 출력 문서에서 사용된 언어와 비교해야 합니다(XML 출력 문서의 문서 로케일 항목 참조).

현재 *requested\_locale*에 대해서는 en\_US 값만 지원됩니다.

### *xml\_input*

현재 이 프로시저에서는 입력을 허용하지 않습니다. 이 매개변수에 대해 NULL을 지정하십시오. 지정하지 않으면 입력이 유효하지 않음을 나타내는 오류(+20458)가 표시됩니다.

### *xml\_filter*

유효한 XPath 쿼리 문자열을 지정하는 BLOB(4K) 유형의 입력 인수입니다. XML 출력 문서에서 한 개의 값만 검색하려는 경우에는 필터를 사용하십시오. 자세한 정보는 XPath 필터링을 설명하는 주제를 참조하십시오.

다음 예에서는 데이터 서버 제품 버전 값이 XML 출력 문서에서 선택됩니다. `/plist/dict/key[.='Data Server Product Version']/following-sibling::string` 키 뒤에 지정한 동위가 오지 않으면 오류가 리턴됩니다.

### *xml\_output*

UTF-8로 전체 XML 출력 문서를 리턴하는 BLOB(32MB) 유형의 출력 매개변수입니다. 필터가 지정된 경우 이 매개변수는 문자열 값을 리턴합니다. 스토어드 프로시저가 전체 출력 문서를 리턴할 수 없는 경우(예: 처리 오류가 발생하여 SQL 경고 또는 오류가 표시되는 경우) 이 매개변수가 NULL로 설정됩니다.



XML 출력 문서에는 FixPack 레벨, 릴리스, 시스템 정보 및 환경 변수가 포함된 인스턴스 정보가 포함됩니다.

#### *xml\_message*

SQL 경고 조건에 대한 자세한 정보를 제공하는 데이터 서버 메시지 유형의 전체 XML 출력 문서를 UTF-8로 리턴하는 BLOB(64K) 형식의 출력 매개변수입니다. 프로시저 호출로 인해 추가 정보가 XML 메시지 출력 문서로 리턴됨을 나타내는 SQL 경고가 발생할 때 이 문서가 리턴됩니다. 경고 메시지 내용이 추가 정보가 리턴됨을 나타내지 않는 경우에는 이 매개변수가 NULL로 설정됩니다.

### 권한 부여

- SYSADM 또는 DBADM 권한
- GET\_SYSTEM\_INFO 프로시저에 대한 EXECUTE 특권

### 예

예 1: 프로시저의 가장 높은 버전을 리턴합니다.

```
db2 "call sysproc.get_system_info(null,null,null,null,null,?,?)"
```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

```
-----  
매개변수 이름 : MAJOR_VERSION  
매개변수 값 : 1  
  
매개변수 이름 : MINOR_VERSION  
매개변수 값 : 0  
  
매개변수 이름 : XML_OUTPUT  
매개변수 값 : -  
  
매개변수 이름 : XML_MESSAGE  
매개변수 값 : -  
  
리턴 상태 = 0
```

예 2: 시스템 정보를 리턴합니다.

```
db2 "call sysproc.get_system_info(1,0,'en_US',null,null,?,?)"
```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

```
-----  
매개변수 이름 : MAJOR_VERSION  
매개변수 값 : 1  
  
매개변수 이름 : MINOR_VERSION  
매개변수 값 : 0
```

매개변수 이름 : XML\_OUTPUT  
매개변수 값 : x'3C3F786D6C20766572.....

매개변수 이름 : XML\_MESSAGE  
매개변수 값 : -

리턴 상태 = 0

XML 출력 문서에는 다음 콘텐츠와 유사한 콘텐츠가 포함됩니다.

```
<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>
      <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>
    </dict>
  </dict>
</plist>
```

```

    <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>l</key>
    <dict>
      <key>Display Name</key><string>l</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>
    </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>

```

예 3: GET\_SYSTEM\_INFO 프로시저를 호출하고 지원되지 않는 로케일을 전달합니다.

```
db2 "call sysproc. get_system_info(1,0,'ja_JP',null,null,?,?)"
```

이 쿼리의 출력 예는 다음과 같습니다.

```

출력 매개변수 값
-----
매개변수 이름  : MAJOR_VERSION
매개변수 값   : 1

매개변수 이름  : MINOR_VERSION
매개변수 값   : 0

매개변수 이름  : XML_OUTPUT
매개변수 값   : x'3C3F786D6C20766572.....

매개변수 이름  : XML_MESSAGE
매개변수 값   : -

리턴 상태 = 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 출력 문서는 예 2에 표시된 동일한 콘텐츠를 포함합니다.

예 4: 함수에서 프로시저를 호출합니다.

```

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 프로시저 - 구성 매개변수 설정

SET\_CONFIG 스토어드 프로시저는 데이터베이스 및 GET\_CONFIG 프로시저로 리턴된 데이터베이스 관리 프로그램 구성 매개변수를 갱신합니다.

SET\_CONFIG 프로시저는 구성 매개변수와 해당 값을 포함하는 입력 XML 문서를 승인하고 이 정보를 사용하여 지정한 구성 매개변수를 갱신하며 각 구성 매개변수의 갱신 상태를 나타내는 출력 XML 문서를 리턴합니다.

### 구문

```
▶▶ SET_CONFIG (—major_version—, —minor_version—, —requested_locale—, —
—xml_input—, —xml_filter—, —xml_output—, —xml_message—) ▶▶
```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### major\_version

기본 문서 버전을 나타내는 INTEGER 유형의 입력 및 출력 인수입니다. 입력에서 이 인수는 호출자가 프로시저에서 매개변수로 전달되는 XML 문서를 지원하는 기본 문서 버전을 나타냅니다(*xml\_input*, *xml\_output* 및 *xml\_message*에 대해서는 매개변수 설명 참조). 프로시저는 지정된 버전의 모든 XML 문서를 처리하거나 버전이 유효하지 않은 경우 오류를 리턴합니다(+20458). 출력에서 이 매개변수는 프로시저에서 지원되는 가장 높은 기본 문서 버전을 지정합니다. 지원되는 가장 높은 문서 버전을 판별하려면 이 입력 매개변수 및 모든 필수 매개변수에 대해 NULL을 지정하십시오.

*xml\_input* 매개변수의 XML 문서가 문서 유형 기본 버전 키를 지정하는 경우 해당 키 값은 *major\_version* 매개변수로 제공한 값과 동등해야 하며 그렇지 않으면 오류(+20458)가 발생합니다.

지원되는 버전: 1

#### minor\_version

보조 문서 버전을 나타내는 INTEGER 유형의 입력 및 출력 인수입니다. 입력에서 이 인수는 호출자가 이 프로시저에서 매개변수로 전달되는 XML 문서를 지원하는 보조 문서 버전을 지정합니다(*xml\_input*, *xml\_output* 및 *xml\_message*에 대해서는 매개변수 설명 참조). 프로시저는 지정된 버전의 모든 XML 문서를 처리하거나 버

전이 유효하지 않은 경우 오류를 리턴합니다. 출력에서 이 매개변수는 지원되는 가장 높은 기본 문서 버전에 대해 지원되는 가장 높은 보조 문서 버전을 나타냅니다. 지원되는 가장 높은 문서 버전을 판별하려면 이 입력 매개변수 및 모든 필수 매개변수에 대해 NULL을 지정하십시오.

*xml\_input* 매개변수의 XML 문서가 문서 유형 보조 버전 키를 지정하는 경우 해당 키 값은 *minor\_version* 매개변수로 제공한 값과 동등해야 하며 그렇지 않으면 오류(+20458)가 발생합니다.

지원되는 버전: 0

#### *requested\_locale*

로케일을 지정하는 VARCHAR(33) 유형의 입력 인수입니다. 지정된 언어가 서버에서 지원되는 경우 변환된 콘텐츠가 *xml\_output* 및 *xml\_message* 매개변수로 리턴됩니다. 그 이외의 경우에는 콘텐츠가 디폴트 언어로 리턴됩니다. 언어와 가능한 지역 정보만 로케일에서 사용됩니다. 로케일은 숫자를 형식화하거나 문서 인코딩을 변경하는 데 사용되지 않습니다. 예를 들어 주요 이름 및 값은 변환되지 않습니다. XML 출력 및 XML 메시지 문서에서는 힌트를 위한 텍스트, 표시 이름 및 각 항목의 표시 부분만 변환됩니다. 호출자는 항상 요청된 언어를 XML 출력 문서에서 사용된 언어와 비교해야 합니다(XML 출력 문서의 문서 로케일 항목 참조).

현재 *requested\_locale*에 대해서는 en\_US 값만 지원됩니다.

#### *xml\_input*

프로시저의 입력 값을 포함하는 XML 입력 문서(UTF-8로 인코딩)를 지정하는 BLOB(32MB) 유형의 입력 인수입니다.

이 프로시저의 경우 XML 입력 문서에는 데이터베이스 및 데이터베이스 관리 프로그램 구성 설정이 포함됩니다.

#### *xml\_filter*

유효한 XPath 쿼리 문자열을 지정하는 BLOB(4K) 유형의 입력 인수입니다. XML 출력 문서에서 한 개의 값만 검색하려는 경우에는 필터를 사용하십시오. 자세한 정보는 XPath 필터링을 설명하는 주제를 참조하십시오.

다음 예에서는 특정 구성 매개변수 설정 값이 XML 출력 문서에서 선택됩니다. `/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]`. 키 뒤에 지정된 동위가 오지 않으면 오류가 리턴됩니다.

#### *xml\_output*

UTF-8로 전체 XML 출력 문서를 리턴하는 BLOB(32MB) 유형의 출력 매개변수입니다. 필터가 지정된 경우 이 매개변수는 문자열 값을 리턴합니다. 스토어드 프로시저가 전체 출력 문서를 리턴할 수 없는 경우(예: 처리 오류가 발생하여 SQL 경고 또는 오류가 표시되는 경우) 이 매개변수가 NULL로 설정됩니다.

이 프로시저가 완료 모드에서 작동하는 경우 이 매개변수는 서버의 현재 구성 값 세트를 포함하는 XML 문서를 리턴합니다. 사용자는 이 문서를 수정하여 다시 프로시저에 `xml_input` 매개변수로 전달할 수 있습니다. 이 접근방식은 유효한 XML 입력 문서를 작성하는 프로그래밍된 방법입니다.

#### *xml\_message*

SQL 경고 조건에 대한 자세한 정보를 제공하는 데이터 서버 메시지 유형의 전체 XML 출력 문서를 UTF-8로 리턴하는 BLOB(64K) 형식의 출력 매개변수입니다. 프로시저 호출로 인해 추가 정보가 XML 메시지 출력 문서로 리턴됨을 나타내는 SQL 경고가 발생할 때 이 문서가 리턴됩니다. 경고 메시지 내용이 추가 정보가 리턴됨을 나타내지 않는 경우에는 이 매개변수가 NULL로 설정됩니다.

### 권한 부여

- SYSADM 또는 DBADM 권한
- SET\_CONFIG 프로시저에 대한 EXECUTE 특권

### 예

예 1: 프로시저의 최신 버전을 리턴합니다.

```
db2 "call sysproc.set_config (null,null,null,null,null,?,?)"
```

이 쿼리의 출력 예는 다음과 같습니다.

```
출력 매개변수 값
-----
매개변수 이름 : MAJOR_VERSION
매개변수 값   : 1

매개변수 이름 : MINOR_VERSION
매개변수 값   : 0

매개변수 이름 : XML_OUTPUT
매개변수 값   : -

매개변수 이름 : XML_MESSAGE
매개변수 값   : -

리턴 상태 = 0
```

예 2: 일부 데이터베이스 및 데이터베이스 관리 프로그램 구성 매개변수를 갱신하는 `setconfig.sql` 스크립트를 실행합니다.

```
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>
<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, ?,?)@

```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

```

-----
매개변수 이름 : MAJOR_VERSION
매개변수 값 : 1

```

```

매개변수 이름 : MINOR_VERSION
매개변수 값 : 0

```

매개변수 이름 : XML\_OUTPUT  
매개변수 값 : x'3C3F78...'

매개변수 이름 : XML\_MESSAGE  
매개변수 값 : -

리턴 상태 = 0

출력 XML 문서에는 다음 콘텐츠와 유사한 콘텐츠가 포함됩니다.

```
<?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>
    <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>
<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>
    <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>
  </dict>
  <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>

```



```

<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>
<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]'),?,?)"

```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

-----  
매개변수 이름 : MAJOR\_VERSION  
매개변수 값 : 1

매개변수 이름 : MINOR\_VERSION  
매개변수 값 : 0

매개변수 이름 : XML\_OUTPUT  
매개변수 값 : x'34303936'

매개변수 이름 : XML\_MESSAGE  
매개변수 값 : -

리턴 상태 = 0

다음 값이 *xml\_output*에 대해 리턴됩니다.

4096

예 4: 함수에서 프로시저를 호출합니다.

```
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>
    <key>instance_memory</key>
    <dict>
      <key>Deferred Value</key>
      <dict>
```

```

    <key>Value</key><string>7424</string>
  </dict>
</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>
</dict>
</plist>'),
null,
:xmlOutput,
:xmlOutMessage );

```





---

## 제 9 장 구성 루틴 및 뷰

---

### DB\_PARTITIONS

DB\_PARTITIONS 테이블 함수는 테이블 형식으로 db2nodes.cfg 파일의 콘텐츠를 리턴합니다.

#### 구문

▶▶—DB\_PARTITIONS—(—)—————▶▶

스키마는 SYSPROC입니다.

#### 권한 부여

DB\_PARTITIONS 테이블 함수에 대한 EXECUTE 특권

#### 테이블 함수 매개변수

함수에 입력 매개변수가 없습니다.

#### 예 :

3개의 논리적 파티션을 갖는 데이터베이스에서 정보를 검색합니다.

```
SELECT * FROM TABLE(DB_PARTITIONS()) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

PARTITION_NUMBER	HOST_NAME	PORT_NUMBER	SWITCH_NAME
0	jessicae.torolab.ibm.com	0	jessicae
1	jessicae.torolab.ibm.com	1	jessicae
2	jessicae.torolab.ibm.com	2	jessicae

3개의 레코드가 선택되었습니다.

#### 리턴되는 정보

표 82. DB\_PARTITIONS 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명
PARTITION_NUMBER	SMALLINT	파티션된 데이터베이스 환경에서 데이터베이스 파티션 서버를 식별하는 0에서 999 사이의 고유한 숫자
HOST_NAME	VARCHAR (128)	데이터베이스 파티션 서버의 TCP/IP 호스트 이름

표 82. DB\_PARTITIONS 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명
PORT_NUMBER	SMALLINT	데이터베이스 파티션 서버의 포트 번호
SWITCH_NAME	VARCHAR (128)	데이터베이스 파티션 통신을 위한 고속 상호 연결 또는 스위치의 이름

## DBCFCG 관리 뷰 - 데이터베이스 구성 매개변수 정보 검색

DBCFCG 관리 뷰는 모든 데이터베이스 파티션에 현재 연결된 데이터베이스에 대한 데이터베이스 구성 매개변수 정보를 검색합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- DBCFCG 관리 뷰에 대한 SELECT 특권
- DBCFCG 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- DB\_GET\_CFG 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

### 예:

예 1: 모든 데이터베이스 파티션의 메모리에 저장된 데이터베이스 구성에서 자동 유지 보수 설정을 검색하십시오.

```
SELECT DBPARTITIONNUM, NAME, VALUE FROM SYSIBMADM.DBCFCG WHERE NAME LIKE 'auto_%'
```

다음은 이 쿼리에 대한 출력의 예입니다.

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

8개의 레코드가 선택되었습니다.

예: 모든 데이터베이스 파티션의 디스크에 저장된 모든 데이터베이스 구성 매개변수 값을 검색하십시오.

```
SELECT NAME, DEFERRED_VALUE, DBPARTITIONNUM FROM SYSIBMADM.DBCFG
```

다음은 이 쿼리에 대한 출력의 예입니다.

```

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
...

```

### 리턴되는 정보

표 83. DBCFG 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
이름	VARCHAR(32)	구성 매개변수 이름.
VALUE	VARCHAR(1024)	메모리에 저장된 구성 매개변수의 현재 값.
VALUE_FLAGS	VARCHAR(10)	구성 매개변수 현재 값에 대한 특정 정보를 제공합니다. 올바른 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• NONE - 추가 정보 없음</li> <li>• AUTOMATIC - 구성 매개변수가 자동으로 설정됨</li> </ul>
DEFERRED_VALUE	VARCHAR(1024)	디스크의 구성 매개변수 값. 일부 데이터베이스 구성 매개변수의 경우, 데이터베이스를 재활성화할 경우에만 변경사항이 적용됩니다. 이 경우, 모든 응용프로그램은 먼저 데이터베이스에서 연결을 끊어야 합니다. (데이터베이스가 활성화되었으면, 비활성화된 후 재활성화되어야 합니다.) 변경사항은 데이터베이스에 다음 번에 연결할 때 적용됩니다.
DEFERRED_VALUE_FLAGS	VARCHAR(10)	지연된 구성 매개변수 값에 대한 특정 정보를 제공합니다. 올바른 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• NONE - 추가 정보 없음</li> <li>• AUTOMATIC - 구성 매개변수가 자동으로 설정됨</li> </ul>

표 83. DBCFG 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
DATATYPE	VARCHAR (128)	구성 매개변수 데이터 유형.
DBPARTITIONNUM	SMALLINT	데이터베이스 파티션 번호.

## DBCFG 관리 뷰 - 데이터베이스 관리 프로그램 구성 매개변수 정보 검색

DBMCFG 관리 뷰는 디스크에 저장된 값과 메모리의 값을 포함한 데이터베이스 관리 프로그램 구성 매개변수 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- DBMCFG 관리 뷰에 대한 SELECT 특권
- DBMCFG 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- DBM\_GET\_CFG 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

### 예:

예 1: 디스크에 저장된 모든 데이터베이스 관리 프로그램 구성 매개변수의 값을 검색합니다.

```
SELECT NAME, DEFERRED_VALUE FROM SYSIBMADM.DBMCFG
```

이 쿼리의 출력 예는 다음과 같습니다.

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/b1dsupp/AIX5L
...
ssl_svcename            22711
ssl_svr_keydb           /GSKit/Keystore/key.kdb
ssl_svr_label           /GSKit/Keystore/key.kdb
ssl_svr_stash           /GSKit/Keystore/key.sth

```

예 2: 모든 데이터베이스 관리 프로그램 구성 매개변수 값을 검색합니다.

```
SELECT * FROM SYSIBMADM.DBMCFG
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...
...			...
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	...

이 쿼리의 출력(계속).

...	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)
...                ...
... YES            NONE    VARCHAR(3)
... SEARCH         NONE    VARCHAR(8)
... ENABLE         NONE    VARCHAR(8)
... 0              AUTOMATIC    BIGINT
... 512            AUTOMATIC    BIGINT
... 0              AUTOMATIC    BIGINT
...

```

## 리턴된 정보

표 84. DBMCFG 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
NAME	VARCHAR(32)	구성 매개변수 이름.
VALUE	VARCHAR(256)	메모리에 저장된 구성 매개변수의 현재 값.
VALUE_FLAGS	VARCHAR(10)	구성 매개변수 현재 값에 대한 특정 정보를 제공합니다. 올바른 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• NONE - 추가 정보 없음</li> <li>• AUTOMATIC - 구성 매개변수가 자동으로 설정됨</li> </ul>
DEFERRED_VALUE	VARCHAR(256)	디스크의 구성 매개변수 값. 일부 데이터베이스 관리 프로그램 구성 매개변수의 경우, 이 값이 적용되려면 데이터베이스 관리 프로그램이 중지되어야 하며(db2stop) 재시작되어야(db2start) 합니다.
DEFERRED_VALUE_FLAGS	VARCHAR(10)	지연된 구성 매개변수 값에 대한 특정 정보를 제공합니다. 올바른 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• NONE - 추가 정보 없음</li> <li>• AUTOMATIC - 구성 매개변수가 자동으로 설정됨</li> </ul>
DATATYPE	VARCHAR(128)	구성 매개변수 데이터 유형.

## REG\_VARIABLES 관리 뷰 - 사용 중인 DB2 레지스트리 설정 검색

REG\_VARIABLES 관리 뷰는 모든 데이터베이스 파티션에서 DB2 레지스트리 설정 값을 리턴합니다. REG\_VARIABLES 관리 뷰를 쿼리할 때 리턴되는 DB2 레지스트리 변수는 인스턴스가 시작된 후 db2set 명령을 사용하여 DB2 레지스트리 변수를 구성한 경우 db2set 명령에서 리턴된 변수 값과 다를 수 있습니다. 이러한 차이는 REG\_VARIABLES만이 인스턴스가 시작되었을 때 적용된 값을 리턴하므로 발생합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- REG\_VARIABLES 관리 뷰에 대한 SELECT 특권
- REG\_VARIABLES 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- REG\_LIST\_VARIABLES 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

### 예 :

현재 사용 중인 DB2 레지스트리 설정값을 요청하십시오.

```
SELECT * from SYSIBMADM.REG_VARIABLES
```

이 쿼리의 출력 예는 다음과 같습니다.

DBPARTITIONNUM	REG_VAR_NAME	REG_VAR_VALUE	IS_AGGREGATE	AGGREGATE_NAME
0	DB2ADMINSERVER	DB2DAS00	0	-
0	DB2INSTPROF	D:WSQLLIB	0	-
0	DB2PATH	D:WSQLLIB	0	-
0	DB2SYSTEM	D570	0	-
0	DB2TEMPDIR	D:WSQLLIBW	0	-
0	DB2_EXTSECURITY	YES	0	-

6개의 레코드가 선택되었습니다.

### 리턴되는 정보

표 85. REG\_VARIABLES 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
DBPARTITIONNUM	SMALLINT	함수가 작동하는 각 데이터베이스 파티션의 논리적 파티션 번호
REG_VAR_NAME	VARCHAR (256)	DB2 레지스트리 변수 이름

표 85. REG\_VARIABLES 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
REG_VAR_VALUE	VARCHAR(2048)	DB2 레지스트리 변수의 현재 설정
IS_AGGREGATE	SMALLINT	DB2 레지스트리 변수가 집계 변수인지 여부를 표시합니다. 가능한 리턴값은 집계 변수가 아닌 경우 0 이고 집계 변수인 경우 1입니다.
AGGREGATE_NAME	VARCHAR (256)	DB2 레지스트리 변수가 현재 구성된 집계에서 해당 값을 가져오는 경우 집계 이름. 레지스트리 변수가 집계를 통해 설정되지 않거나 집계를 통해 설정되었지만 겹쳐지지 않은 경우, AGGREGATE_NAME 값은 NULL입니다.
LEVEL	CHAR(1)	DB2 레지스트리 변수가 해당 값을 획득하는 레벨을 표시합니다. 가능한 리턴값과 해당 레벨은 다음과 같습니다. <ul style="list-style-type: none"> <li>• I = 인스턴스</li> <li>• G = 전역</li> <li>• N = 데이터베이스 파티션</li> <li>• E = 환경</li> </ul>



---

## 제 10 장 환경 뷰

---

### ENV\_FEATURE\_INFO 관리 뷰 - DB2 기능에 대한 라이선스 정보 리턴

ENV\_FEATURE\_INFO 관리 뷰는 라이선스가 필요한 모든 사용 가능한 기능에 대한 정보를 리턴합니다. 각 기능마다 기능에 유효한 라이선스가 설치되었는지 여부에 대한 정보가 있습니다.

스키마는 SYSIBMADM입니다.

#### 권한 부여

다음 권한 중 하나가 필요합니다.

- ENV\_FEATURE\_INFO 관리 뷰에 대한 SELECT 특권
- ENV\_FEATURE\_INFO 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- ENV\_GET\_FEATURE\_INFO 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

#### 예 :

설치된 DB2 기능 라이선스 정보를 요청하십시오.

```
SELECT * FROM SYSIBMADM.ENV_FEATURE_INFO
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...
GEO	DB2_GEODETTIC_DATA_MANAGEMENT_FEATURE	...
HFESE	IBM_HOMOGENEOUS_FEDERATION_FEATURE_FOR_ESE	...
XMLESE	DB2_PUREXML_FEATURE_FOR_ESE	...

이 쿼리의 출력(계속).

...	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
... Y	ESE	NOT_USED
... N	ESE	IN_VIOLATION

## ENV\_FEATURE\_INFO 관리 뷰 메타데이터

표 86. ENV\_FEATURE\_INFO 관리 뷰 메타데이터

컬럼 이름	데이터 유형	설명
FEATURE_NAME	VARCHAR(26)	사용 허가된 DB2 서버에 사용 가능한 DB2 기능
FEATURE_FULLNAME	VARCHAR(100)	DB2 기능의 전체 이름. 컬럼 값은 영어 대문자로 표시됩니다. 단어는 공백 문자 대신 밑줄 문자로 구분됩니다.
LICENSE_INSTALLED	CHAR(1)	기능의 사용 허가 여부를 표시합니다. 값이 'N'이면 기능의 사용 허가가 없습니다. 값이 'Y'이면 기능의 사용 허가가 있습니다.
PRODUCT_NAME	VARCHAR(26)	기능을 사용할 수 있는 DB2 서버 제품의 ID 가능한 리턴 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• ESE - DB2 Enterprise Server Edition</li> <li>• WSE - DB2 Workgroup Server Edition</li> <li>• EXP - DB2 Express™ Edition</li> </ul>
FEATURE_USE_STATUS	VARCHAR (30)	라이선스 준수 상태를 표시합니다. 이 값은 기능의 사용 상태를 표시합니다. 가능한 값은 다음 세 가지입니다. <ul style="list-style-type: none"> <li>• IN_COMPLIANCE: 기능이 최소 한 번은 사용되었으며 기능에 대한 유효한 라이선스가 있습니다.</li> <li>• IN_VIOLATION: 기능이 최소 한 번은 사용되었으며 기능에 대한 유효한 라이선스가 없습니다.</li> <li>• NOT_USED: 기능이 사용되지 않았습니다.</li> </ul>

## ENV\_INST\_INFO 관리 뷰 - 현재 인스턴스에 대한 정보 검색

ENV\_INST\_INFO 관리 뷰는 현재 인스턴스에 대한 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- ENV\_INST\_INFO 관리 뷰에 대한 SELECT 특권
- ENV\_INST\_INFO 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- ENV\_GET\_INST\_INFO 테이블 함수에 대한 EXECUTE 특권

- DATAACCESS 권한

예 :

현재 인스턴스에 대한 정보를 요청하십시오.

```
SELECT * FROM SYSIBMADM.ENV_INST_INFO
```

다음은 이 쿼리에 대한 출력의 예입니다.

```
INST_NAME          IS_INST_PARTITIONABLE NUM_DBPARTITIONS INST_PTR_SIZE ...
-----
DB2                0                      1                32 ...
```

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... RELEASE_NUM    SERVICE_LEVEL        BLD_LEVEL           PTF                 FIXPACK_NUM
... RELEASE_NUM    SERVICE_LEVEL        BLD_LEVEL           PTF                 FIXPACK_NUM
-----
... 01010107      DB2 v9.1.0.115      n051106             0
```

## 리턴되는 정보

표 87. ENV\_INST\_INFO 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
INST_NAME	VARCHAR (128)	현재 인스턴스 이름
IS_INST_PARTITIONABLE	SMALLINT	현재 인스턴스가 파티션 가능한 데이터베이스 서버 인스턴스인지 여부를 표시합니다. 파티션 가능한 데이터베이스 서버 인스턴스가 아닌 경우, 가능한 리턴값은 0이고 파티션 가능한 데이터베이스 서버 인스턴스인 경우, 1입니다.
NUM_DBPARTITIONS	INTEGER	데이터베이스 파티션 수. 파티션된 데이터베이스 환경이 아닌 경우 값 1을 리턴합니다.
INST_PTR_SIZE	INTEGER	현재 인스턴스의 비트 크기(32 또는 64)
RELEASE_NUM	VARCHAR (128)	db2level 명령이 리턴한 내부 릴리스 번호(예 : 03030106)
SERVICE_LEVEL	VARCHAR (128)	db2level 명령이 리턴한 서비스 레벨. 예를 들어, DB2 v8.1.1.80.
BLD_LEVEL	VARCHAR (128)	db2level 명령이 리턴한 빌드 레벨(예: n041021)
PTF	VARCHAR (128)	db2level 명령이 리턴한 프로그램 임시 수정 (PTF) ID(예: U498350)
FIXPACK_NUM	INTEGER	db2level 명령이 리턴한 FixPak 번호(예: 9)

## ENV\_PROD\_INFO 관리 뷰 - 설치된 DB2 제품에 대한 정보 검색

ENV\_PROD\_INFO 관리 뷰는 설치된 DB2 제품에 대한 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- ENV\_PROD\_INFO 관리 뷰에 대한 SELECT 특권
- ENV\_PROD\_INFO 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- ENV\_GET\_PROD\_INFO\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

### 예 :

설치된 DB2 제품 정보를 요청합니다.

```
SELECT * FROM SYSIBMADM.ENV_PROD_INFO
```

이 쿼리의 출력 예는 다음과 같습니다.

```
INSTALLED_PROD  INSTALLED_PROD_FULLNAME  ...
-----
ESE              DB2_ENTERPRISE_SERVER_EDITION  ...
WSE              DB2_WORKGROUP_SERVER_EDITION  ...
EXP              DB2_EXPRESS_EDITION            ...
```

이 쿼리의 출력(계속).

```
... 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 관리 뷰 메타데이터

표 88. ENV\_PROD\_INFO 관리 뷰 메타데이터

컬럼 이름	데이터 유형	설명
INSTALLED_PROD	VARCHAR(26)	시스템에 설치된 DB2 제품 ID
INSTALLED_PROD_FULLNAME	VARCHAR(100)	설치된 DB2 제품의 전체 이름. 컬럼 값은 영어 대문자로 표시됩니다. 단어는 밑줄 문자로 구분됩니다.
LICENSE_INSTALLED	CHAR(1)	제품의 사용 허가 여부를 표시합니다. 값이 'N'이면 제품의 사용 허가가 없습니다. 값이 'Y'이면 제품의 사용 허가가 있습니다.

표 88. ENV\_PROD\_INFO 관리 뷰 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
PROD_RELEASE	VARCHAR(26)	제품 릴리스 번호
LICENSE_TYPE	VARCHAR(50)	제품에 설치된 라이선스 유형의 이름. 가능한 리턴 값은 다음과 같습니다. <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 관리 뷰 - 시스템에 대한 정보 검색

ENV\_SYS\_INFO 관리 뷰는 시스템에 대한 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- ENV\_SYS\_INFO 관리 뷰에 대한 SELECT 특권
- ENV\_SYS\_INFO 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- ENV\_GET\_SYS\_INFO 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

예 :

시스템에 대한 정보를 요청하십시오.

```
SELECT * from SYSIBMADM.ENV_SYS_INFO
```

이 쿼리의 출력 예는 다음과 같습니다.

OS_NAME	OS_VERSION	OS_RELEASE	HOST_NAME
WIN32_NT	5.1	Service Pack 1	D570

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	TOTAL_CPUS	CONFIGURED_CPUS	TOTAL_MEMORY
...	1	2	1527

## 리턴되는 정보

표 89. ENV\_SYS\_INFO 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
OS_NAME	VARCHAR (256)	운영 체제 이름
OS_VERSION	VARCHAR (256)	운영 체제 버전 번호
OS_RELEASE	VARCHAR (256)	운영 체제 릴리스 번호
HOST_NAME	VARCHAR (256)	시스템의 이름.
TOTAL_CPUS	INTEGER	시스템의 실제 총 CPU 수
CONFIGURED_CPUS	INTEGER	시스템에 구성된 실제 CPU 수
TOTAL_MEMORY	INTEGER	시스템의 총 메모리량(MB)

## ENV\_SYS\_RESOURCES 관리 뷰 - 시스템 정보 리턴

ENV\_SYS\_RESOURCES 관리 뷰는 시스템에 관련된 운영 체제, CPU, 메모리 및 기타 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- ENV\_SYS\_RESOURCES 관리 뷰에 대한 SELECT 특권
- ENV\_SYS\_RESOURCES 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- ENV\_GET\_SYS\_RESOURCES 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

예 :

```
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'
```

이 쿼리의 출력 예는 다음과 같습니다.

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

4개의 레코드가 선택되었습니다.

## ENV\_SYS\_RESOURCES 관리 뷰 메타데이터

표 90. ENV\_SYS\_RESOURCES 관리 뷰 메타데이터

컬럼 이름	데이터 유형	설명
NAME	VARCHAR (128)	속성의 이름입니다. 가능한 값은 364 페이지의 표 91의 내용을 참조하십시오. 주: 일부 속성은 운영 체제 및 서버의 하드웨어 구성에 따라 사용이 가능하지 않을 수 있습니다.
VALUE	VARCHAR(1024)	속성의 값
DATATYPE	VARCHAR (128)	속성 데이터 유형
UNIT	VARCHAR (128)	VALUE 컬럼에 사용되는 단위(적용 가능한 경우). 적용할 수 없는 경우, NULL이 리턴됩니다.
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

표 91. NAME 컬럼의 가능한 값

정보 유형	이름	데이터 유형	설명	이 정보를 리턴하는 플랫폼	UNIT
운영 체제	OS_NAME	VARCHAR(256)	운영 체제 소프트웨어의 이름.	모두	NULL
	HOST_NAME	VARCHAR(256)	시스템의 호스트 이름	모두	NULL
	OS_VERSION	VARCHAR(256)	운영 체제의 버전(예: AIX: 4.3 버전 = 4)	모두	NULL
	OS_RELEASE	VARCHAR(256)	운영 체제의 릴리스(예: AIX: 4.3 릴리스 = 3)	모두	NULL
	MACHINE_IDENTIFICATION	VARCHAR(256)	머신 하드웨어 식별.	모두	NULL
	OS_LEVEL	VARCHAR(256)	현재 버전 및 릴리스의 유지보수 레벨(예: LINUX: 2.4.9, 레벨 = 9)	Linux	NULL
CPU	CPU_TOTAL	BIGINT	총 CPU 수.	모두	NULL
	CPU_ONLINE	BIGINT	CPU 온라인 수.	모두	NULL
	CPU_CONFIGURED	BIGINT	구성된 CPU 수.	모두	NULL
	CPU_SPEED	BIGINT	CPU 속도	모두	MHz
	CPU_TIMEBASE	BIGINT	시간대 레지스터 증분 빈도	Linux PowerPC®	Hz
	CPU_HMT_DEGREE	BIGINT	하드웨어 멀티스레드(HMT)를 지원하는 시스템에서 이것은 실제 프로세서가 운영 체제에 표시되는 프로세서의 수입니다. 비HMT 시스템에서, 이 값은 1입니다. HMT 시스템에서, "total"은 논리적 CPU 수를 반영합니다. 실제 CPU 수를 가져오려면, "총계"를 "threadingDegree"로 나 누십시오.	모두	NULL
	CPU_CORES_PER_SOCKET	BIGINT	소켓당 CPU 코어 수. 단일 코어 시스템에서, 이 값은 1입니다.	모두	NULL
실제 메모리	MEMORY_TOTAL	BIGINT	실제 메모리의 전체 크기	모두	MB
	MEMORY_FREE	BIGINT	여유 실제 메모리의 양	모두	MB
	MEMORY_SWAP_TOTAL	BIGINT	스왑 영역의 총량	모두	MB
	MEMORY_SWAP_FREE	BIGINT	여유 스왑 영역의 양	모두	MB



표 91. NAME 컬럼의 가능한 값 (계속)

정보 유형	이름	데이터 유형	설명	이 정보를 리턴하는 플랫폼	UNIT
가상 메모리	VIRTUAL_MEM_TOTAL	BIGINT	시스템의 가상 메모리 총량	모두	MB
	VIRTUAL_MEM_RESERVED	BIGINT	예약된 가상 메모리의 양	모두	MB
	VIRTUAL_MEM_FREE	BIGINT	여유 가상 메모리의 양	모두	MB
CPU 로드	CPU_LOAD_SHORT	DECIMAL	최단 기간 지속 기간. 예를 들어 지난 5분간의 샘플을 로드합니다.	모두(Windows 운영 체제 제외)	NULL
	CPU_LOAD_MEDIUM	DECIMAL	매체 기간 지속 기간. 예를 들어 지난 10분간의 샘플을 로드합니다.	모두(Windows 운영 체제 제외)	NULL
	CPU_LOAD_LONG	DECIMAL	장기간 지속 기간. 예를 들어 지난 15분간의 샘플을 로드합니다.	모두(Windows 운영 체제 제외)	NULL
	CPU_USAGE_TOTAL	DECIMAL	머신의 전체 CPU 사용 백분율	모두	퍼센트



---

## 제 11 장 모니터 루틴

모니터 테이블 함수는 MON\_GET\_SERVICE\_SUBCLASS와 같이 이름이 "MON"으로 시작하는 테이블 함수입니다. 이 테이블 함수는 DB2 버전 9.7에 소개된 모니터링 인프라스트럭처에서 사용 가능한 모니터 요소에 대한 액세스를 제공합니다.

### 모니터(MON) 테이블 함수

한 세트의 모니터 테이블 함수는 전체 시스템 워크로드의 다양한 여러 면에 대해 보고합니다.

- MON\_GET\_CONNECTION 및 MON\_GET\_CONNECTION\_DETAILS
- MON\_GET\_SERVICE\_SUBCLASS 및 MON\_GET\_SERVICE\_SUBCLASS\_DETAILS
- MON\_GET\_UNIT\_OF\_WORK 및 MON\_GET\_UNIT\_OF\_WORK\_DETAILS
- MON\_GET\_WORKLOAD 및 MON\_GET\_WORKLOAD\_DETAILS

이 테이블 함수에는 두 개의 버전이 있으며 이 중 하나의 접미부는 "\_DETAILS"입니다. "\_DETAILS" 접미부가 없는 버전은 가장 일반적으로 사용되는 데이터를 리턴하는 관계형 SQL 인터페이스입니다. "\_DETAILS" 접미부가 포함된 버전은 모니터 데이터에 대한 XML 기반 액세스를 제공하며 가장 포괄적인 데이터 세트를 리턴합니다.

모든 테이블 함수에는 일반적인 모니터 요소 세트가 포함됩니다. 이 요소는 응용프로그램 응답 시간에 영향을 줄 수 있는 다양한 시스템 성능 표시기에 대한 정보를 제공합니다. 관심있는 워크로드 서버세트에 대한 모니터 데이터를 가져올 수도 있습니다.

다른 세트의 테이블 함수는 특정 유형의 데이터 오브젝트에 대한 데이터를 리턴합니다.

- MON\_GET\_BUFFERPOOL
- MON\_GET\_CONTAINER
- MON\_GET\_INDEX
- MON\_GET\_TABLE
- MON\_GET\_TABLESPACE
- MON\_GET\_PKG\_CACHE\_STMT

이 테이블 함수를 사용하여 특정 데이터 오브젝트에 연관된 성능 문제를 조사하십시오.

다른 테이블 함수는 각 활동 및 명령문에 대한 세부사항을 조사하는 데 유용합니다.

- MON\_GET\_ACTIVITY\_DETAILS는 시스템에서 현재 실행 중인 특정 활동에 대한 세부사항을 리턴하며 이 세부사항에는 일반 활동 정보(예: 명령문 텍스트) 및 메트릭 세트가 포함됩니다.

또한 이 테이블 함수는 진행 상태 모니터링 역할도 수행합니다.

- MON\_GET\_EXTENT\_MOVEMENT\_STATUS는 Extent 이동 조작의 상태를 리턴합니다.

## 모니터(MON) 테이블 함수의 특성

- 모니터링 테이블 함수에서 리턴된 메트릭은 절대 재설정되지 않습니다. 데이터베이스가 활성화될 때 0에서 시작되며 데이터베이스가 비활성화될 때까지 누적됩니다.
- 대부분의 테이블 함수를 사용하여 오브젝트 한 개(예: 서비스 클래스 A) 또는 여러 오브젝트에 대한 데이터를 수신하도록 선택할 수 있습니다.
- 대부분의 테이블 함수의 경우 파티션된 데이터베이스 환경에서 이 테이블 함수를 사용하면 단일 파티션 또는 전체 파티션에 대한 데이터를 수신하도록 선택할 수 있습니다. 전체 파티션의 데이터를 수신하도록 선택하면 테이블 함수가 각 파티션에 대해 한 행을 리턴합니다. 파티션의 값을 추가하여 파티션의 모니터 요소 값을 얻을 수 있습니다.

## 이벤트 모니터(EVMON) 루틴

DB2 버전 9.7 릴리스에는 다른 "MON" 테이블 함수와는 조금 다른 용도의 새 루틴이 2개 추가되었습니다. 이 루틴은 이벤트를 비형식화 이벤트 테이블에 기록하는 이벤트 모니터에서 데이터를 추출하여 형식화합니다. LOCKING 및 UNIT OF WORK 이벤트 모니터 유형은 비형식화 이벤트 테이블을 사용합니다. 루틴 이름은 다음과 같습니다.

- EVMON\_FORMAT\_UE\_TO\_XML 테이블 함수
- EVMON\_FORMAT\_UE\_TO\_TABLES 프로시저

이 루틴을 사용하면 EVMON\_FORMAT\_UE\_TO\_XML 테이블 함수를 사용하여 XML 문서를 통해 또는 EVMON\_FORMAT\_UE\_TO\_TABLES 프로시저를 사용하여 관계형 테이블을 통해 이벤트 모니터 데이터에 액세스할 수 있습니다.

---

## EVMON\_FORMAT\_UE\_TO\_TABLES 프로시저 - XML 문서를 관계형 테이블로 이동

EVMON\_FORMAT\_UE\_TO\_TABLES 프로시저는

EVMON\_FORMAT\_UE\_TO\_XML 테이블 함수를 사용하여 비형식화 이벤트 테이블에 저장된 데이터를 검색한 다음 분석 기술을 사용하여 XML 문서를 관계형 테이블 세트로 이동합니다.

### 구문

```
▶▶—EVMON_FORMAT_UE_TO_TABLES—(—evmon_type—,—xsrschema—,——————▶▶
```

▶*-xsobjectname—,—xmlschemafilename—,—tabschema—,—*—————▶  
▶*-tblsp\_name—,—options—,—commit\_count—,—fullselect—)*—————▶

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *evmon\_type*

비형식화 이벤트 테이블에 저장된 데이터의 유형을 나타내는 VARCHAR(128) 유형의 입력 인수. 가능한 값은 다음과 같습니다.

#### **LOCKING**

비형식화 이벤트 테이블에 저장된 데이터는 잠금 이벤트 모니터에서 가져옵니다.

**UOW** 비형식화 이벤트 테이블에 저장된 데이터는 UOW 이벤트 모니터에 있습니다.

### *xrschema*

관계형 XSR 오브젝트 이름의 첫 번째 파트를 지정하는 VARCHAR(128) 유형의 입력. 두 번째 파트는 *xsobjectname* 매개변수에서 파생됩니다. 전체 XSR 오브젝트 이름은 *xrschema.xsobjectname*으로 정의되어 있습니다. 이 값이 NULL인 경우, 현재 세션 사용자의 권한 부여 ID가 사용됩니다.

### *xsobjectname*

XSR 오브젝트 이름의 두 번째 파트를 지정하는 VARCHAR(128) 유형의 입력 인수. XSR 오브젝트 이름의 세 번째 파트는 *xrschema* 매개변수에서 파생됩니다. 전체 XSR 오브젝트 이름은 *xrschema.xsobjectname*으로 정의되며 XSR의 모든 오브젝트 중에서 고유한 이름입니다.

이 값이 NULL인 경우, *xsobjectname*은 다음과 같이 파생됩니다. `EVMON_<evmon_type>_SCHEMA_<SQL release level>`. 예를 들어 DB2 버전 9.7의 잠금 이벤트 모니터는 `EVMON_LOCKING_SCHEMA_SQL09070`의 *xsrname*을 파생합니다. XSR 오브젝트는 XSR 스키마 저장소에 저장되어 분석에 사용된 XML 스키마를 참조하며 이것은 주석과 주석이 달린 XML 스키마에서 참조된 테이블 간의 종속성을 작성하는 프로세스를 참조합니다.

### *xmlschemafilename*

디스크의 완전한 주석이 달린 XML 스키마 문서에 대한 완전한 경로인 VARCHAR(1024) 유형의 입력 인수. 여기에서 주석은 XML 요소 및 속성을 기본 관계형 테이블 및 컬럼에 맵핑하는 DB2 분석 주석을 참조합니다.

이 값이 NULL인 경우 프로시저는 다음과 같이 지정된 XML 스키마를 사용합니다.

## LOCKING

sql1lib/misc/DB2EvmonLocking.xsd

UOW sql1lib/misc/DB2EvmonUOW.xsd

### tabschema

이벤트 모니터의 관계형 테이블이 작성되는 SQL 스키마 이름 나타내는 VARCHAR(128) 유형의 입력 인수입니다. 이 값이 NULL인 경우, 현재 세션 사용자의 권한 부여 ID가 사용됩니다. 테이블이 작성된 SQL 스키마는 다음과 같이 판별됩니다.

- <db2-xdb:SQLSchema>가 지정된 경우 이 스키마를 사용하십시오.
- <db2-xdb:defaultSchema>가 지정된 경우 이 스키마를 사용하십시오.
- 모두 지정되지 않은 경우에는 *sqlschema* 입력 인수의 값을 사용하십시오.

주: XML 스키마가 분석에 등록되면 XSR 스키마 저장소가 스키마에서 참조된 각 테이블과 이 스키마에 해당하는 XSR 오브젝트 간의 종속성을 작성합니다. 이것은 XSR 오브젝트 이름이 데이터베이스의 관계형 테이블의 고유 세트에 연결됨을 의미합니다. 기존 XSR 오브젝트를 참조하면 데이터가 항상 분석되어 XSR 오브젝트가 연결된 테이블에 삽입됩니다.

### tblsp\_name

관계형 테이블이 작성된 테이블 스페이스를 표시하는 VARCHAR(128) 유형의 입력 인수. 이 매개변수의 디폴트값은 NULL입니다. XML 스키마 파일내의 CREATE TABLE문에 지정된 테이블 스페이스 이름이 이 입력 매개변수보다 우선권이 있습니다.

### options

이 테이블 함수에서 지원된 키워드 옵션의 목록을 나타내는 VARCHAR(1024) 유형의 입력 인수. 각 옵션은 세미콜론(;) 문자로 구분해야 합니다. 사용할 수 있는 값은 다음과 같습니다.

#### RECREATE\_FORCE

관계형 테이블이 삭제되어 분석 전에 다시 작성되어야 함을 표시합니다.

#### RECREATE\_ONERROR

다음과 같은 상황에서는 관계형 테이블을 삭제하고 다시 작성해야 함을 나타냅니다.

1. XSR 오브젝트가 등록되지 않았지만 테이블이 있습니다.
2. 실패한 첫 번째 분리 시도. 연속된 실패가 리턴되지만 테이블을 다시 작성하는 시도가 없습니다.

테이블 스페이스에 전체 오류나 권한 부여 오류가 발생하면 프로시저는 분리 프로시저에서 리턴된 SQLCODE를 필터링하지 않습니다. 프로시저는 모든 음수의 SQLCODES를 동일하게 처리하고 테이블 다시 작성을 시도합니다.

#### *commit\_count*

INTEGER 유형의 입력 인수. 가능한 값은 다음과 같습니다.

- 1 100개의 문서 분석이 성공할 때 마다 커밋. 이는 디폴트값입니다.
- 0 커밋 안함.
- n n개의 문서 분석이 성공할 때 마다 커밋.

#### *fullselect*

비형식화 이벤트 테이블에서 fullselect문을 나타내는 CLOB(2M) 유형의 입력 인수. fullselect문은 SELECT문 규칙을 준수하는 쿼리입니다. 쿼리는 다음 규칙을 따라야 합니다.

- 쿼리는 "\*"절을 사용하거나 비형식화 이벤트 테이블의 모든 컬럼을 지정해야 합니다. 그렇지 않으면 오류가 발생합니다. 컬럼은 비형식화 이벤트 테이블의 DESCRIBE문에서 리턴된 순서와 동일한 순서로 지정해야 합니다.
- 쿼리는 형식이 지정되지 않은 이벤트 테이블에서만 선택됩니다.
- WHERE절은 형식이 지정되지 않은 이벤트 테이블의 비LOB 컬럼을 사용하여 이벤트를 필터링합니다.

## 권한 부여

EVMON\_FORMAT\_UE\_TO\_TABLES 스토어드 프로시저에 대한 EXECUTE 특권  
비형식화 이벤트 테이블의 SELECT 특권(아직 작성하지 않은 경우).

지정한 SQL 스키마에 관계형 테이블을 작성하기 위해 CREATE 특권.

관계형 테이블에 삽입하기 위해 INSERT 특권(아직 작성하지 않은 경우).

XDB\_DECOMP\_XMP\_FROM\_QUERY 프로시저에 필요한 모든 특권.

## 예:

예 1: 디폴트 매개변수 사용.

사용자 Paul이 디폴트 매개변수를 사용하여 프로시저를 호출하고 서비스 클래스 STUDENTS의 일부인 모든 이벤트를 관계형 테이블에 삽입하도록 요청합니다.

```

EVMON_FORMAT_UE_TO_TABLES (
  'UOW', NULL, NULL, NULL, NULL, NULL, -1,
  'SELECT * FROM UOWUE
   WHERE service_subclass_name = 'STUDENTS'
   ORDER BY event_id, event_timestamp')

```

호출 결과는 다음과 같습니다.

1. 프로시저는 디폴트 XML 스키마 파일인 DB2EvmonUOW.xsd 파일을 구문 분석하여 작성할 관계형 테이블 세트를 식별합니다.
2. 관계형 테이블이 SQL 스키마 Paul 밑에 작성됩니다.
3. XSR XML 스키마는 PAUL.EVMON\_UOW\_SCHEMA\_SQL09070 이름의 XSR 오브젝트로 등록됩니다.
4. XSR 오브젝트를 분석에 사용할 수 있습니다.
5. 데이터가 분석되어 SQL 스키마 Paul에 삽입됩니다.

예 2: 다른 스키마에서 테이블 사용 시도.

이전 예에 이어서, 사용자 Dave가 *tabschema* 매개변수를 Paul로 설정하여 스토어드 프로시저를 호출합니다.

```

EVMON_FORMAT_UE_TO_TABLES (
  'UOW', NULL, NULL, NULL, 'Paul', NULL, NULL, -1,
  'SELECT * FROM UOWTBLE
   ORDER BY event_timestamp')

```

호출 결과는 다음과 같습니다.

1. 프로시저는 디폴트 XML 스키마 파일인 DB2EvmonUOW.xsd 파일을 구문 분석하여 작성할 관계형 테이블 세트를 식별합니다.
2. 프로시저가 스키마 Paul 아래 테이블 작성을 시도합니다. 그렇지만 현재 SQL 스키마 PAUL에 관계형 테이블이 있기 때문에 오류가 리턴됩니다. 이전부터 있던 테이블은 새 XSR 오브젝트 등록 시에 사용할 수 없습니다.

예 3: 다른 스키마에서 테이블 사용 시도.

이전 예에 이어서, 사용자 Greg가 입력 매개변수 *xsrschema*를 Paul로 설정하여 스토어드 프로시저를 호출합니다.

```

EVMON_FORMAT_UE_TO_TABLES (
  'UOW', 'Paul', NULL, NULL, NULL, NULL, NULL, -1,
  'SELECT * FROM UOWTBL
   ORDER BY event_timestamp')

```

호출 결과는 다음과 같습니다.

1. 이미 있는 XSR 오브젝트 Paul.EVMON\_UOW\_SCHEMA\_SQL09070을 분석할 수 있습니다.



- Greg가 테이블에 대해 INSERT 특권이 있는 경우 데이터가 분석되어 SQL 스키마 Paul의 관계형 테이블에 삽입됩니다. 기존의 XSR 오브젝트 Paul.EVMON\_UOW\_SCHEMA\_SQL09070이 사용되기 때문에 관계형 테이블의 SQL 스키마는 프로시저의 입력 매개변수로 제공되는 대신 XSR 오브젝트에서 가져옵니다.

예 4: RECREATE\_FORCE 옵션 사용.

이전 예에 이어서, Paul은 테이블을 다시 작성하지만 이번에는 테이블 스페이스 MYSPACE에 작성합니다. Paul은 RECREATE\_FORCE 옵션 및 *tblsp\_name* 매개변수를 사용하여 프로시저를 호출합니다.

```

EVMON_FORMAT_UO_TO_TABLES (
  'UOW', NULL, NULL, NULL, NULL, 'MYSPACE', 'RECREATE_FORCE', -1,
  'SELECT * FROM UOWTBL
   ORDER BY event_timestamp')

```

호출 결과는 다음과 같습니다.

- 이미 있는 XSR 오브젝트 Paul.EVMON\_UOW\_SCHEMA\_SQL09070을 분석할 수 있습니다.
- RECREATE\_FORCE 옵션이 설정됩니다.
- XML 스키마 파일이 스키마 저장소에서 검색되어 관계형 파일 세트를 식별하기 위해 구문 분석됩니다.
- 현재 테이블은 삭제되고 MYSPACE 테이블 스페이스에 다시 작성됩니다.
- 데이터가 분석되어 새 테이블에 삽입됩니다.

예 5: FixPack 갱신에서 새 요소 선택.

새 XML 요소인 『db2EventNew』가 최신 FixPack에서 잠금 이벤트 모니터의 XML 스키마 파일로 추가되었습니다. Paul은 새 요소를 선택하여 XML 파일 분석에 사용하려고 합니다. 사용하기 위해 다음과 같은 단계를 수행합니다.

- 임의의 편집기를 사용하여 잠금 이벤트 모니터 파일인 DB2EvmonLocking.xml을 엽니다.
- LOCK\_EVENT 테이블의 CREATE TABLE문이 포함된 <db2-mon:createStmt> 요소를 찾습니다.
- Paul은 다음 예에 표시된 것처럼 rowset 값에 주의하여 테이블에 새 컬럼을 작성합니다.

```

<xs:annotation>
  <xs:appinfo>
    <db2-xdb:table>
      <db2-xdb:name>LOCK_CONTROL</db2-xdb:name>
      <db2-mon:createStmt>
        CREATE TABLE LOCK_EVENT (EVENT_ID INT,
          EVENT_TIMESTAMP TIMESTAMP,
          EVENT_TYPE VARCHAR(128),
          :
          EVENT_NEW VARCHAR(128)

```

```

)
</db2-mon:createStmnt> <db2-xdb:rowSet>lock_event_rowset</db2-xdb:rowSet>
:
</db2-xdb:table>
</xs:appinfo>
</xs:annotation>

```

4. 그런 다음, 이벤트 모니터 XML 스키마 파일에서 새 XML 요소를 찾아 분석을 위해 새 요소가 파트가 되는 rowset 및 컬럼 이름을 추가하여 주석 처리합니다.

```

<xs:complexType name="db2lockevent">
  <xs:annotation>
    <xs:documentation>
      This schema describes the structure of each locking event captured by the
      event monitor. The schema consists of the following elements:
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="db2EventNew"
      type="xs:string"
      minOccurs="0"
      maxOccurs="1"
      db2-xdb:rowRowSet="lock_event_rowset"
      db2-xdb:column="EVENT_NEW" >
      <xs:annotation>
        <xs:documentation>
          Event New is a column added during FIXPACK X.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
  :

```

5. 그런 다음 Paul은 원래 릴리스에서 작성된 XSR 오브젝트를 삭제(drop)합니다.

```
DROP XSROBJECT EVMON_LOCKING_SCHEMA_SQL09070
```

6. 마지막으로 RECREATE\_ONERROR 옵션을 지정하여 프로시저를 호출합니다.

```

EVMON_FORMAT_UE_TO_TABLES (
  'LOCKING', NULL, NULL, NULL, NULL, NULL, 'RECREATE_ONERROR', -1,
  'SELECT * FROM LOCK
  ORDER BY event_timestamp')

```

호출 결과는 다음과 같습니다.

- XSR 오브젝트가 없기 때문에 관계형 테이블 세트를 식별하기 위해 디폴트 DB2EvmonLocking.xsd 스키마 파일이 구문 분석됩니다.
- RECREATE\_ONERROR 옵션이 지정되었기 때문에 기존 테이블은 삭제되어 다시 작성됩니다.

예 6: 릴리스 갱신에서 새 요소 선택

Paul은 새 DB2 릴리스로 업그레이드하여 이벤트 모니터 XML 스키마 파일에서 새 변경사항을 선택하려고 합니다. Paul은 RECREATE\_ONERROR 옵션을 지정하여 프로시저를 호출합니다.

```

EVMON_FORMAT_UE_TO_TABLES (
  'LOCKING', NULL, NULL, NULL, NULL, NULL, 'RECREATE_ONERROR', -1,
  'SELECT * FROM LOCK
  ORDER BY event_timestamp')

```

호출 결과는 다음과 같습니다.

1. XSR 오브젝트 Paul.EVMON\_LOCKING\_SCHEMA\_SQL1000이 존재하지 않습니다.
2. RECREATE\_ONERROR 옵션이 지정되었기 때문에 테이블은 삭제되어 다시 작성됩니다.

예 7: 릴리스 갱신에 이전 관계형 테이블 사용.

Greg는 새 DB2 릴리스로 업그레이드했지만 이벤트 모니터 XML 스키마 파일의 새 변경사항을 선택하려고 하지 않습니다. Greg는 이전 릴리스의 *xsrobjectname* 값을 사용하여 프로시저를 호출합니다.

```
EVMON_FORMAT_UE_TO_TABLES (  
  'LOCKING', NULL, 'EVMON_LOCKING_SCHEMA_SQL09070', NULL, NULL, NULL, NULL, -1,  
  'SELECT * FROM LOCK  
  ORDER BY event_timestamp')
```

## 사용 시 참고사항

### 테이블 작성

분석이 발생하려면 관계형 테이블 세트가 존재해야 합니다.

EVMON\_FORMAT\_UE\_TO\_TABLES 프로시저는 다음과 같이 관계형 테이블을 자동으로 작성합니다.

- 프로시저는 <db2-mon:createStmt> 요소를 찾기 위해 이벤트 모니터의 XML 스키마 파일을 구문 분석합니다. 각 요소에는 완벽한 CREATE TABLE문이 포함됩니다.
- 프로시저는 CREATE TABLE문을 추출하여 실행합니다.

<db2-mon:createStmt>는 기존 <db2-xdb:table> 요소의 하위 요소입니다. EVMON\_FORMAT\_UE\_TO\_TABLES 프로시저만 이 요소를 인식하여 사용합니다. XSR 오브젝트와 같은 XML 스키마 파일을 구문 분석하는 다른 모든 프로시저는 이 요소를 무시합니다.

<db2-mon:createStmt>에서는 테이블 이름을 규정화하지 마십시오.

### FixPack 갱신

DB2 FixPack에는 스키마에 대한 새 XML 요소가 있는 갱신된 버전의 이벤트 모니터 XML 스키마 파일이 포함될 수 있습니다. 그렇지만 <db2-mon:createStmt> 요소의 create문에는 새 컬럼이 포함되지 않습니다. 또한 새 XML 요소나 속성은 주석 처리되지 않습니다. 따라서 아무 영향없이 특정 FixPack 레벨에서 작성된 디폴트 관계형 테이블을 사용할 수 있습니다.

### 릴리스 갱신

이후의 DB2 릴리스에서는 새 컬럼이 이벤트 모니터 관계형 테이블에 추가될 수도 있습니다. 주석을 새 컬럼에 맵핑하여 새 XML 요소 또는 속성에 주석이 포함됩니다. 이전 테이블을 사용하려면 이전 릴리스의 XSR 오브젝트 이름을 사용하여 프로시저를 호출하십시오. 이외의 경우에는 이전 테이블은 삭제(drop)되고 새 테이블이 작성될 수 있도록 RECREATE\_ONERROR 옵션을 사용하여 프로시저를 호출하십시오.

### 릴리스 간의 XML 스키마 파일

각 이벤트 모니터가 제공하는 디폴트 XML 스키마 파일은 항상 현재 릴리스의 XML 스키마를 반영합니다.

### 리턴되는 정보

SQLCA를 제외한 프로시저로부터의 출력이 없습니다. SQLCA는 완료 상태를 표시합니다. 가능한 SQLCODES는 다음과 같습니다.

0 모든 이벤트가 관계형 테이블에 성공적으로 삽입되었습니다.

16278 하나 이상의 이벤트가 관계형 테이블에 삽입되지 않았습니다. SQLCA 내의 토큰에 시도된 전체 문서 수 및 분석이 완료된 전체 문서 수가 포함됩니다.

진단 파일도 작성됩니다. 해당 진단 파일의 이름 및 위치가 DB2 diagnostic 경로의 db2diag 로그 파일에 저장됩니다.

음수 *sqlcode*

오류가 발생했음을 나타내며 SQLCODE 메시지를 조사하여 실패에 대한 추가 정보를 확인할 수 있습니다. 추가적인 진단 메시지에 대해서는 DB2 diagnostic 경로에 있는 db2diag 로그 파일을 참조하십시오.

---

## EVMON\_FORMAT\_UE\_TO\_XML 테이블 함수 - 비형식화 이벤트를 XML로 변환

EVMON\_FORMAT\_UE\_TO\_XML 테이블 함수는 비형식화 이벤트 테이블에서 2진 이벤트를 추출하여 XML 문서로 형식화합니다.

### 구문

```
▶▶—EVMON_FORMAT_UE_TO_XML—(—options—, ——————▶  
▶—FOR EACH ROW OF—(—fullselect-statement—)—▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### options

이 테이블 함수에서 지원하는 키워드 옵션 목록을 나타내는 VARCHAR(1024) 유형의 입력 인수입니다.

#### LOG\_TO\_FILE

XML 문서가 100MB 이상인 경우 테이블 함수가 XML 문서를 파일로 작성하는 것을 나타냅니다. 이 테이블 함수에서 행별로 리턴되는 각 문서의 최대 크기는 100MB입니다. 파일은 <xml\_document\_id>.xml 파일로 작성되며 여기서, <xml\_document\_id>는 각 문서에 대해 생성되는 고유 ID입니다. 출력 파일은 DB2 진단 경로 디렉토리에 작성됩니다.

#### LOG\_PARTIAL\_EVENTS

테이블 함수가 모든 부분(전체가 아닌) 이벤트를 파일로 작성하는지 여부를 나타냅니다. 특정 파일을 지시하는 진단 메시지가 db2diag 로그 파일에 삽입됩니다.

NULL 선택된 옵션이 없습니다.

### fullselect-statement

fullselect문은 SELECT문 규칙을 준수하는 쿼리입니다. 쿼리는 다음 규칙을 따라야 합니다.

- 쿼리는 "\*"절을 사용하거나 비형식화 이벤트 테이블의 모든 컬럼을 지정해야 합니다. 그렇지 않으면 오류가 발생합니다. 컬럼은 비형식화 이벤트 테이블의 DESCRIBE문에서 리턴된 순서와 동일한 순서로 지정해야 합니다.
- 쿼리는 형식이 지정되지 않은 이벤트 테이블에서만 선택됩니다.
- WHERE절은 형식이 지정되지 않은 이벤트 테이블의 비LOB 컬럼을 사용하여 이벤트를 필터링합니다.
- SELECT문은 FOR EACH ROWS OF 키워드를 대괄호로 묶어 지정해야 합니다.

## 권한 부여

EVMON\_FORMAT\_UE\_TO\_XML 함수에서의 EXECUTE 특권.

형식이 지정되지 않은 이벤트 테이블의 SELECT 특권.

### 예:

예 1: 비형식화 이벤트 테이블 『MYLOCKS』에서 모든 이벤트를 조회합니다.

```
SELECT evmon.* FROM TABLE (  
  EVMON_FORMAT_UE_TO_XML (  
    NULL ,
```

```

        FOR EACH ROW OF (
        select * from MYLOCKS
        order by EVENT_TIMESTAMP )))
AS evmon;

```

예 2: 비형식화 이벤트 테이블 『LOCK』에서 최근 5시간 내에 발생한 『LOCKWAIT』 유형의 모든 이벤트를 조회합니다.

```

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;

```

예 3: 비형식화 이벤트 테이블 『UOW』에서 최근 32시간 내에 발생한 워크로드 『PAYROLL』에 속하는 모든 이벤트를 가져옵니다. 100MB 이상인 문서가 있는 경우에는 결과가 파일로 작성됩니다.

```

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;

```

예 4: 『UOWEVMON』 테이블에서 전체 작업 단위(UOW) 이벤트를 조회하고 XMLTABLE 테이블 함수를 사용하여 UOW ID, UOW 시작 및 중지 시간 작업 단위(UOW)를 발행한 개인의 사용자 ID를 표시합니다.

```

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

```

## 사용 시 참고사항

EVMON\_FORMAT\_UE\_TO\_XML 테이블 함수를 잠금 및 작업 단위(UOW) 이벤트 모니터와 같이 사용하여 비형식화 이벤트 테이블에서 데이터를 추출할 수 있습니다.

이벤트 모니터 유형에 따라 테이블 함수는 형식이 지정되지 않은 이벤트 테이블에서 단일 이벤트로 다중 레코드를 맵핑합니다. 이런 경우 해당 이벤트를 구성하는 모든 레코드가 수신될 때까지 레코드는 메모리에 캐시됩니다. 테이블 함수로 전달된 레코드가 작성되어 테이블에 삽입된 순서대로 정렬되지 않으면 더 큰 메모리가 필요하게 될 수도 있습니다. 레코드가 이런 방식으로 정렬되지 않은 경우에는 테이블 함수가 여러 이벤트에 대해 레코드를 캐시해야 합니다. 이 문제를 방지하려면 *fullselect-statement* 매개변수를 EVENT\_ID, EVENT\_TIMESTAMP, EVENT\_TYPE 및 MEMBER 컬럼이 포함된 ORDER BY절로 규정화하십시오. 지정된 시점에서 테이블 함수가 단일 이벤트의 레코드만을 처리하여 캐시하기 때문에 메모리 사용이 줄어듭니다.

## 리턴된 정보

표 92. EVMON\_FORMAT\_UE\_TO\_XML에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
XMLID	VARCHAR (1024)	고유 문서 ID. ID는 다음과 같이 파생됩니다. <event_header>_<event_id>_<event_type>_<event_timestamp>_<partition>
XMLREPORT	BLOB(100M)	한개의 완료된 이벤트가 포함되는 XML 문서. 각 문서의 최대 크기는 100MB입니다.

## MON\_GET\_ACTIVITY\_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기

MON\_GET\_ACTIVITY\_DETAILS 테이블 함수는 일반 활동 정보(명령문 텍스트와 같은) 및 활동에 대한 메트릭 세트를 포함한 활동에 관한 세부사항을 리턴합니다.

### 구문

```

▶▶MON_GET_ACTIVITY_DETAILS(—application_handle—,—uow_id—,—
▶—activity_id—,—member—)

```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### application\_handle

유효한 응용프로그램 핸들을 지정하는 유형 BIGINT의 입력 인수. 인수가 널(null)이면 이 함수로부터 행이 리턴되지 않으며, SQL0171N 오류가 리턴됩니다.

#### uow\_id

응용프로그램 내에서 유효한 고유 작업 단위 ID를 지정하는 INTEGER 유형의 입력 인수. 인수가 널(null)이면 이 함수로부터 행이 리턴되지 않으며, SQL0171N 오류가 리턴됩니다.

### *activity\_id*

작업 단위 내에서 유효한 고유 활동 ID를 지정하는 INTEGER 유형의 입력 인수. 인수가 널(null)이면 이 함수로부터 행이 리턴되지 않으며, SQL0171N 오류가 리턴됩니다.

### *member*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 구성원 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

## 권한 부여

MON\_GET\_ACTIVITY\_DETAILS 함수에서의 EXECUTE 특권.

### 예 :

실행 또는 대기 시간(예를 들어 잠금 또는 입출력 차단)을 소요하고 있는지 여부를 판별하려면 긴 실행 쿼리를 조사합니다.

1. 먼저 WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 테이블 함수를 사용하여 활동 및 시작 시간을 나열합니다.

```
SELECT application_handle,
       activity_id,
       uow_id,
       local_start_time
FROM TABLE(
  WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES(
    cast(NULL as bigint), -1)
) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

APPLICATION_HANDLE	ACTIVITY_ID	UOW_ID	LOCAL_START_TIME
7	1	2	2008-06-10-10.06.55.675668
16	1	7	2008-06-10-10.08.38.613610

2개의 레코드가 선택되었습니다.

2. 그런 다음 MON\_GET\_ACTIVITY\_DETAILS 테이블 함수를 사용하여 활동이 대기하는 데 소요된 시간의 백분율을 봅니다.

```
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;

```

이 쿼리의 출력 예는 다음과 같습니다.

```

APPLICATION_HANDLE  ACTIVITY_ID  UOW_ID      ...
-----
              7              1              2  ...

```

1개의 레코드가 선택되었습니다.

쿼리의 출력(계속).

```

... STMT_TEXT      ...
... -----      ...
... select * from syscat.tables optimize for 1 row      ...

```

쿼리의 출력(계속).

```

... TOTAL_ACT_TIME  TOTAL_ACT_WAIT_TIME  PERCENTAGE_WAIT_TIME
... -----
...                459                0                0.00

```

## 사용 시 참고사항

단일 활동에 대한 자세한 정보가 XML 문서로 리턴되기 때문에

MON\_GET\_ACTIVITY\_DETAILS 함수는 포매팅 출력에 대한 최대 유연성을 제공합니다. XML 출력에는 설명 정보(예를 들어 명령문 텍스트)와 메트릭 모두가 포함됩니다. 출력은 XML 구문 분석기에 의해 직접 구문 분석되거나 XMLTABLE 함수에 의해 관계형 형식으로 변환될 수 있습니다.

이 함수를 통해 보고된 메트릭은(예를 들어 CPU 사용) 활동 존재 기간 동안 주기적으로 활동에 롤업됩니다. 그러므로 이 테이블 함수로 보고된 값은 최근 롤업 시 시스템의 현재 상태를 반영합니다.

활동 메트릭은 워크로드에서 COLLECT ACTIVITY METRICS절을 통해 제어되거나 데이터베이스 레벨에서 mon\_act\_metrics 데이터베이스 구성 매개변수를 통해 제어됩니다. 활동을 제출한 연결이 활동 메트릭이 사용 가능한 데이터베이스나 워크로드와 연관되면 메트릭이 수집됩니다. 활동 메트릭이 활동에 대해 수집되지 않으면 모든 메트릭이 영(0)으로 보고됩니다.

MON\_GET\_ACTIVITY\_DETAILS 테이블 함수는 활동이 존재하는 각 구성원에 대해 한 개의 데이터 행을 리턴합니다. 메트릭에 대해서는 구성원에서 집계가 수행되지 않습니다. 하지만, SQL 쿼리를 통해 집계가 이루어질 수 있습니다.

DETAILS 컬럼에 리턴된 XML 문서의 스키마는 sqllib/misc/DB2MonRoutines.xsd 파일로 사용 가능합니다. sqllib/misc/DB2MonCommon.xsd 파일에서 더 자세한 내용을 볼 수 있습니다.

## 리턴된 정보

표 93. MON\_GET\_ACTIVITY\_DETAILS에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
APPLICATION_HANDLE	BIGINT	application_handle - 응용프로그램 핸들
UOW_ID	INTEGER	uow_id - 작업 단위(UOW) ID
ACTIVITY_ID	INTEGER	activity_id - 활동 ID
MEMBER	SMALLINT	member - 데이터베이스 구성원
DETAILS	BLOB(8M)	활동 세부사항을 포함한 XML 문서. 이 문서에서 요소 설명은 383 페이지의 표 94의 내용을 참조하십시오.

다음 예는 DETAILS 컬럼에서 리턴되는 XML 문서의 구조를 표시합니다.

```
<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>
```

전체 스키마에 대해서는 sqllib/misc/DB2MonRoutines.xsd의 내용을 참조하십시오. 이 문서는 다음 XML 비원시 유형 정의를 사용합니다.

```
<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>
```

## 리턴된 메트릭 세부사항

표 94. MON\_GET\_ACTIVITY\_DETAILS에 대해 리턴된 메트릭 세부사항

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
구성원	xs:nonNegativeInteger	member - 데이터베이스 구성원
client_userid	xs:string(255)	CURRENT CLIENT_USERID 특수 레지스터
client_wrkstnname	xs:string(255)	CURRENT CLIENT_WRKSTNNAME 특수 레지스터
client_applname	xs:string(255)	CURRENT CLIENT_APPLNAME 특수 레지스터
client_acctng	xs:string(255)	CURRENT CLIENT_ACCTNG 특수 레지스터
application_handle	xs:nonNegativeInteger	application_handle - 응용프로그램 핸들
coord_member	xs:nonNegativeInteger	coord_member - 코디네이터 구성원
uow_id	xs:nonNegativeInteger	uow_id - 작업 단위(UOW) ID
activity_id	xs:nonNegativeInteger	activity_id - 활동 ID
parent_uow_id	xs:nonNegativeInteger	parent_uow_id - 상위 작업 단위 ID
parent_activity_id	xs:nonNegativeInteger	parent_activity_id - 상위 활동 ID
activity_state	xs:string	activity_state - 활동 상태
activity_type	xs:string	activity_type - 활동 유형
nesting_level	xs:nonNegativeInteger	nesting_level - 중첩 레벨
invocation_id	xs:nonNegativeInteger	invocation_id - 호출 ID
routine_id	xs:nonNegativeInteger	routine_id - 루틴 ID
utility_id	xs:nonNegativeInteger	utility_id - 유틸리티 ID
service_class_id	xs:integer	service_class_id - 서비스 클래스
database_work_action_set_id	xs:nonNegativeInteger	db_work_action_set_id - 데이터베이스 작업 조치 세트 ID
database_work_class_id	xs:nonNegativeInteger	db_work_class_id - 데이터베이스 작업 클래스 ID
service_class_work_action_set_id	xs:nonNegativeInteger	sc_work_action_set_id - 서비스 클래스 작업 조치 세트 ID
service_class_work_class_id	xs:nonNegativeInteger	sc_work_class_id - 서비스 클래스 작업 클래스 ID
entry_time	xs:dateTime	entry_time - 입력 시간이 활동이 시스템에 도달된 시간.
local_start_time	xs:dateTime	local_start_time - 로컬 시작 시간.
last_reference_time	xs:dateTime	last_reference_time - 최종 참조 시간. 이 활동에서 요청이 발생될 때마다 이 필드가 갱신됩니다.
package_name	xs:string(128)	package_name - 패키지 이름
package_schema	xs:string(128)	package_schema - 패키지 스키마
package_version_id	xs:string(128)	package_version_id - 패키지 버전
section_number	xs:integer	section_number - 섹션 번호
stmt_pkg_cache_id	xs:nonNegativeInteger	stmt_pkgcache_id - 명령문 패키지 캐시 ID
stmt_text	xs:string	stmt_text - SQL문 텍스트. 활동이 동적 SQL이거나 명령문 텍스트가 사용 가능한 정적 SQL이면, 이 필드는 명령문 텍스트의 처음 1024자를 포함합니다. 그렇지 않으면 빈 문자열을 포함합니다.
effective_isolation	xs:string	effective_isolation - 유효 분리
effective_lock_timeout	xs:integer	effective_lock_timeout - 유효 잠금 시간중료
effective_query_degree	xs:integer	effective_query_degree - 유효 쿼리 등급
query_cost_estimate	xs:integer	query_cost_estimate - 추정 쿼리 비용

표 94. MON\_GET\_ACTIVITY\_DETAILS에 대해 리턴된 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
qp_query_id	xs:nonNegativeInteger	qp_query_id - Query patroller 쿼리 ID
num_remaps	xs:nonNegativeInteger	num_remaps - 리맵핑 번호
concurrentdbcoordactivities_db_threshold_id	xs:int	concurrentdbcoordactivities_db_threshold_id - 동시 데이터베이스 코디네이터 활동 임계값 ID
concurrentdbcoordactivities_db_threshold_value	xs:long	concurrentdbcoordactivities_db_threshold_value - 동시 데이터베이스 코디네이터 활동 수
concurrentdbcoordactivities_db_threshold_queued	xs:short (1 = yes, 0 = no)	concurrentdbcoordactivities_db_threshold_queued - 동시 데이터베이스 코디네이터 활동 수
concurrentdbcoordactivities_db_threshold_violated	xs:short (1 = yes, 0 = no)	concurrentdbcoordactivities_db_threshold_violated - 동시 데이터베이스 코디네이터 활동 임계값 위반
concurrentdbcoordactivities_superclass_threshold_id	xs:int	concurrentdbcoordactivities_superclass_threshold_id - 동시 데이터베이스 코디네이터 활동 수퍼 클래스
concurrentdbcoordactivities_superclass_threshold_value	xs:long	concurrentdbcoordactivities_superclass_threshold_value - 동시 데이터베이스 코디네이터 활동 수퍼 클래스 임계값
concurrentdbcoordactivities_superclass_threshold_queued	xs:short (1 = yes, 0 = no)	concurrentdbcoordactivities_superclass_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 수퍼 클래스 임계값
concurrentdbcoordactivities_superclass_threshold_violated	xs:short (1 = yes, 0 = no)	concurrentdbcoordactivities_superclass_threshold_violated - 동시 데이터베이스 코디네이터 활동 수퍼 클래스 임계값 위반
concurrentdbcoordactivities_subclass_threshold_id	xs:int	concurrentdbcoordactivities_subclass_threshold_id - 동시 데이터베이스 코디네이터 활동 서브클래스 임계값 ID
concurrentdbcoordactivities_subclass_threshold_value	xs:long	concurrentdbcoordactivities_subclass_threshold_value - 동시 데이터베이스 코디네이터 활동 서브클래스 임계값
concurrentdbcoordactivities_subclass_threshold_queued	xs:short (1 = yes, 0 = no)	concurrentdbcoordactivities_subclass_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 서브클래스 임계값
concurrentdbcoordactivities_subclass_threshold_violated	xs:short (1 = yes, 0 = no)	concurrentdbcoordactivities_subclass_threshold_violated - 위반된 동시 데이터베이스 코디네이터 활동 서브클래스 임계값
concurrentdbcoordactivities_work_action_set_threshold_id	xs:int	concurrentdbcoordactivities_work_action_set_threshold_id - 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값 ID
concurrentdbcoordactivities_work_action_set_threshold_value	xs:long	concurrentdbcoordactivities_work_action_set_threshold_value - 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값
concurrentdbcoordactivities_work_action_set_threshold_queued	xs:short (1 = yes, 0 = no)	concurrentdbcoordactivities_work_action_set_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값
concurrentdbcoordactivities_work_action_set_threshold_violated	xs:short (1 = yes, 0 = no)	concurrentdbcoordactivities_work_action_set_threshold_violated - 동시 데이터베이스 코디네이터 활동 작업 조치 임계값 위반
estimatedsqlcost_threshold_id	xs:int	estimatedsqlcost_threshold_id - SQL 추정 비용 임계값 ID
estimatedsqlcost_threshold_value	xs:long	estimatedsqlcost_threshold_value - SQL 추정 비용 임계값
estimatedsqlcost_threshold_violated	xs:short (1 = yes, 0 = no)	estimatedsqlcost_threshold_violated - SQL 추정 비용 임계값 위반
sqltempespace_threshold_id	xs:int	sqltempespace_threshold_id - SQL 임시 스페이스 임계값 ID

표 94. MON\_GET\_ACTIVITY\_DETAILS에 대해 리턴된 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
sqltempespace_threshold_value	xs:long	sqltempespace_threshold_value - SQL 임시 스페이스 임계값
sqltempespace_threshold_violated	xs:short (1 = yes, 0 = no)	sqltempespace_threshold_violated - SQL 임시 스페이스 임계값 위반
sqlrowsreturned_threshold_id	xs:int	sqlrowsreturned_threshold_id - 임계값 ID를 리턴하는 SQL 행 읽기
sqlrowsreturned_threshold_value	xs:long	sqlrowsreturned_threshold_value - 임계값을 리턴하는 SQL 행 읽기
sqlrowsreturned_threshold_violated	xs:short (1 = yes, 0 = no)	sqlrowsreturned_threshold_violated - 위반된 임계값을 리턴하는 SQL 행 읽기
activitytotaltime_threshold_id	xs:int	activitytotaltime_threshold_id - 활동 총 시간 임계값 ID
activitytotaltime_threshold_value	xs:dateTime	activitytotaltime_threshold_value - 활동 총 시간 임계값
activitytotaltime_threshold_violated	xs:short (1 = yes, 0 = no)	activitytotaltime_threshold_violated - 위반된 활동 총 시간 임계값
cputime_threshold_id	xs:int	cputime_threshold_id - CPU 시간 임계값 ID
cputime_threshold_value	xs:long	cputime_threshold_value - CPU 시간 임계값
cputime_threshold_violated	xs:short (1 = yes, 0 = no)	cputime_threshold_violated - CPU 시간 임계값 위반
cputimeinsec_threshold_id	xs:int	cputimeinsec_threshold_id - 서비스 CPU 시간 임계값 ID
cputimeinsec_threshold_value	xs:long	cputimeinsec_threshold_value - 서비스 CPU 시간 임계값
cputimeinsec_threshold_violated	xs:short (1 = yes, 0 = no)	cputimeinsec_threshold_violated - 서비스 CPU 시간 임계값 위반
sqlrowsread_threshold_id	xs:int	sqlrowsread_threshold_ID - SQL 행 읽기 임계값 ID
sqlrowsread_threshold_value	xs:long	sqlrowsread_threshold_value - SQL 행 읽기 임계값
sqlrowsread_threshold_violated	xs:short (1 = yes, 0 = no)	sqlrowsread_threshold_violated - SQL 행 읽기 임계값 위반
sqlrowsreadinsec_threshold_id	xs:int	sqlrowsreadinsec_threshold_id - 서비스 임계값 ID에서 SQL 행 읽기
sqlrowsreadinsec_threshold_value	xs:long	sqlrowsreadinsec_threshold_value - 서비스 임계값에서 SQL 행 읽기
sqlrowsreadinsec_threshold_violated	xs:short (1 = yes, 0 = no)	sqlrowsreadinsec_threshold_violated - 위반된 서비스 임계값에서 SQL 행 읽기
aggsqltempespace_threshold_id	xs:int	aggsqltempespace_threshold_id - AggSQL 임시 스페이스 임계값 ID.
aggsqltempespace_threshold_value	xs:long	aggsqltempespace_threshold_value - AggSQL 임시 스페이스 임계값
aggsqltempespace_threshold_violated	xs:short (1 = yes, 0 = no)	aggsqltempespace_threshold_violated - AggSQL 임시 스페이스 임계값 위반
audit_events_total	xs:nonNegativeInteger	audit_events_total - 총 감사 이벤트 수
audit_subsystem_wait_time	xs:nonNegativeInteger	audit_subsystem_wait_time - 감사 서브시스템 대기 시간
audit_subsystem_waits_total	xs:nonNegativeInteger	audit_subsystem_waits_total - 전체 감사 서브시스템 대기
audit_file_write_wait_time	xs:nonNegativeInteger	audit_file_write_wait_time - 감사 파일 작성 대기 시간

표 94. MON\_GET\_ACTIVITY\_DETAILS에 대해 리턴된 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
audit_file_writes_total	xs:nonNegativeInteger	audit_file_writes_total - 작성된 총 감사 파일 수
deadlocks	xs:nonNegativeInteger	deadlocks - 발견된 교착 상태 수
diaglog_writes_total	xs:nonNegativeInteger	diaglog_writes_total - 도표 로그 전체 작성 수
diaglog_write_wait_time	xs:nonNegativeInteger	diaglog_write_wait_time - 도표 로그 작성 시간
direct_read_time	xs:nonNegativeInteger	direct_read_time - 직접 읽기 시간
direct_write_time	xs:nonNegativeInteger	direct_write_time - 직접 쓰기 시간
direct_read_reqs	xs:nonNegativeInteger	direct_read_reqs - 직접 읽기 요청 수
direct_reads	xs:nonNegativeInteger	direct_reads - 데이터베이스에서 직접 읽기 수
direct_write_reqs	xs:nonNegativeInteger	direct_write_reqs - 직접 쓰기 요청 수
direct_writes	xs:nonNegativeInteger	direct_writes - 데이터베이스에 직접 쓰기 수
fcm_recv_volume	xs:nonNegativeInteger	fcm_recv_volume - FCM 수신 볼륨
fcm_recv_wait_time	xs:nonNegativeInteger	fcm_recv_wait_time - FCM 수신 대기 시간
fcm_recvs_total	xs:nonNegativeInteger	fcm_recvs_total - 총 FCM 수신 수
fcm_message_recv_volume	xs:nonNegativeInteger	fcm_message_recv_volume - FCM 메시지 수신 볼륨
fcm_message_recvs_total	xs:nonNegativeInteger	fcm_message_recvs_total - 총 FCM 메시지 수신 수
fcm_message_recv_wait_time	xs:nonNegativeInteger	fcm_message_recv_wait_time - FCM 메시지 수신 대기 시간
fcm_message_send_volume	xs:nonNegativeInteger	fcm_message_send_volume - FCM 메시지 전송 볼륨
fcm_message_send_wait_time	xs:nonNegativeInteger	fcm_message_send_wait_time - FCM 메시지 전송 대기 시간
fcm_message_sends_total	xs:nonNegativeInteger	fcm_message_sends_total - 총 FCM 메시지 전송 수
fcm_send_volume	xs:nonNegativeInteger	fcm_send_volume - FCM 전송 볼륨
fcm_send_wait_time	xs:nonNegativeInteger	fcm_send_wait_time - FCM 전송 대기 시간
fcm_sends_total	xs:nonNegativeInteger	fcm_sends_total - 총 FCM 전송 수
fcm_tq_recv_wait_time	xs:nonNegativeInteger	fcm_tq_recv_wait_time - FCM 테이블 큐 수신 대기 시간
fcm_tq_send_wait_time	xs:nonNegativeInteger	fcm_tq_send_wait_time - FCM 테이블 큐 전송 대기 시간
fcm_tq_recv_volume	xs:nonNegativeInteger	fcm_tq_recv_volume - FCM 테이블 큐 수신 볼륨
fcm_tq_recvs_total	xs:nonNegativeInteger	fcm_tq_recvs_total - FCM 테이블 큐 수신 총계
fcm_tq_send_volume	xs:nonNegativeInteger	fcm_tq_send_volume - FCM 테이블 큐 전송 볼륨
fcm_tq_sends_total	xs:nonNegativeInteger	fcm_tq_sends_total - FCM 테이블 큐 전송 총계
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총계
lock_escals	xs:nonNegativeInteger	lock_escals - 잠금 에스컬레이션 수
lock_timeouts	xs:nonNegativeInteger	lock_timeouts - 잠금 시간종료 수
lock_wait_time	xs:nonNegativeInteger	lock_wait_time - 잠금 대기 시간
lock_waits	xs:nonNegativeInteger	lock_waits - 잠금 대기 수
log_buffer_wait_time	xs:nonNegativeInteger	log_buffer_wait_time - 로그 버퍼 대기 시간
log_disk_wait_time	xs:nonNegativeInteger	log_disk_wait_time - 로그 디스크 대기 시간
log_disk_waits_total	xs:nonNegativeInteger	log_disk_waits_total - 총 로그 디스크 대기 수
pool_data_l_reads	xs:nonNegativeInteger	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
pool_data_p_reads	xs:nonNegativeInteger	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
pool_data_writes	xs:nonNegativeInteger	pool_data_writes - 버퍼 풀 데이터 쓰기 수
pool_index_l_reads	xs:nonNegativeInteger	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수

표 94. MON\_GET\_ACTIVITY\_DETAILS에 대해 리턴된 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
pool_index_p_reads	xs:nonNegativeInteger	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
pool_index_writes	xs:nonNegativeInteger	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
pool_read_time	xs:nonNegativeInteger	pool_read_time - 총 버퍼 풀 실제 읽기 시간
pool_temp_data_l_reads	xs:nonNegativeInteger	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
pool_temp_data_p_reads	xs:nonNegativeInteger	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
pool_temp_index_l_reads	xs:nonNegativeInteger	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
pool_temp_index_p_reads	xs:nonNegativeInteger	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
pool_temp_xda_l_reads	xs:nonNegativeInteger	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
pool_temp_xda_p_reads	xs:nonNegativeInteger	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수
pool_write_time	xs:nonNegativeInteger	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
pool_xda_l_reads	xs:nonNegativeInteger	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
pool_xda_p_reads	xs:nonNegativeInteger	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
pool_xda_writes	xs:nonNegativeInteger	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
num_log_buffer_full	xs:nonNegativeInteger	num_log_buffer_full - 전체 로그 버퍼 수
rows_modified	xs:nonNegativeInteger	rows_modified - 수정된 행 수
rows_read	xs:nonNegativeInteger	rows_read - 읽은 행 수
rows_returned	xs:nonNegativeInteger	rows_returned - 리턴된 행 수
total_cpu_time	xs:nonNegativeInteger	total_cpu_time - 총 CPU 시간
total_act_time	xs:nonNegativeInteger	total_act_time - 전체 활동 시간
total_act_wait_time	xs:nonNegativeInteger	total_act_wait_time - 전체 활동 대기 시간
total_section_sort_time	xs:nonNegativeInteger	total_section_sort_time - 전체 섹션 정렬 시간.
total_section_sort_proc_time	xs:nonNegativeInteger	total_section_sort_proc_time - 전체 섹션 정렬 처리 시간
total_section_sorts	xs:nonNegativeInteger	total_section_sorts - 전체 섹션 정렬.
total_sorts	xs:nonNegativeInteger	total_sorts - 전체 정렬
post_threshold_sorts	xs:nonNegativeInteger	post_threshold_sorts - 포스트 임계값 정렬
post_shrthreshold_sorts	xs:nonNegativeInteger	post_shrthreshold_sorts - 포스트 공유 임계값 정렬
sort_overflows	xs:nonNegativeInteger	sort_overflows - 정렬 오버플로우
executable_id	executable_id_type	executable_id - 실행 가능 ID
wlm_queue_time_total	xs:nonNegativeInteger	wlm_queue_time_total - 워크로드 관리 프로그램 총 큐 시간
wlm_queue_assignments_total	xs:nonNegativeInteger	wlm_queue_assignments_total - 워크로드 관리 프로그램 총 큐 지정 수
eff_stmt_text	xs:string	eff_stmt_text - 적용 가능한 명령문 텍스트. 명령문 집중기에 의해 이루어지는 리터럴 교체가 이어지는 집중 명령문 텍스트의 처음 1024자. 명령문 집중기가 사용 가능하고 이 명령문이 명령문 집중기에 의해 변경된 경우에만 표시됩니다.



## MON\_GET\_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기

MON\_GET\_BUFFERPOOL 테이블 함수는 하나 이상의 버퍼 풀에 대한 모니터 메트릭을 리턴합니다.

### 구문

▶▶ MON\_GET\_BUFFERPOOL(—*bp\_name*—, —*member*—)▶▶

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *bp\_name*

이 함수 호출 시 현재 연결된 데이터베이스에서 유효한 버퍼 풀 이름을 지정하는 VARCHAR (128) 유형의 입력 인수. 인수가 널(NULL) 또는 비어 있는 문자열인 경우 데이터베이스의 모든 데이터베이스에 대해 메트릭이 검색됩니다.

#### *member*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

### 권한 부여

MON\_GET\_BUFFERPOOL 테이블 함수의 EXECUTE 특권

### 예 :

버퍼 풀 사용 비율을 계산합니다.

```
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,  
         dbpartitionnum FROM BPMETRICS;
```

다음은 이 쿼리의 출력 예입니다.



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

5개의 레코드가 선택되었습니다.

쿼리의 출력(계속).

```

... HIT_RATIO MEMBER
... -----
...      37.80          0
...      -            0... - 0... - 0... - 0

```

## 사용 시 참고사항

MON\_GET\_BUFFERPOOL 테이블 함수는 데이터베이스 버퍼 풀 및 데이터베이스 구성원별로 한 개의 데이터 행을 리턴합니다. 데이터베이스 구성원간의 집계는 수행되지 않습니다. 그렇지만 예에서 표시된 대로 SQL 쿼리를 사용하여 집계를 수행할 수 있습니다.

이 함수에 의해 수집된 메트릭은 mon\_obj\_metrics 구성 매개변수를 사용하여 데이터베이스 레벨에서 제어됩니다. 디폴트로 메트릭 컬렉션이 사용 가능합니다.

## 리턴된 정보

표 95. MON\_GET\_BUFFERPOOL에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
BP_NAME	VARCHAR (128)	
MEMBER	SMALLINT	member - 데이터베이스 구성원
AUTOMATIC	SMALLINT	automatic - 버퍼 풀 자동
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수

표 95. MON\_GET\_BUFFERPOOL에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 수
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 수
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 수
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - 버퍼 풀 희생 버퍼 없음
POOL_LSN_GAP_CLNS	BIGINT	pool_lsn_gap_clns - 버퍼 풀 로그 스페이스 클리너 트리거
POOL_DRDY_PG_STEAL_CLNS	BIGINT	pool_drty_pg_steal_clns - 버퍼 풀 희생(victim) 페이지 클리너 트리거
POOL_DRDY_PG_THRSH_CLNS	BIGINT	pool_drty_pg_thrsh_clns - 버퍼 풀 임계값 클리너 트리거
VECTORED_IOS	BIGINT	vectored_ios - 벡터 입출력 요청 수
PAGES_FROM_VECTORED_IOS	BIGINT	pages_from_vectored_ios - 벡터 입출력으로 읽은 총 페이지 수
BLOCK_IOS	BIGINT	block_ios - 블록 입출력 요청 수
PAGES_FROM_BLOCK_IOS	BIGINT	pages_from_block_ios - 블록 입출력으로 읽은 총 페이지 수
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리페치 페이지 수
FILES_CLOSED	BIGINT	files_closed - 닫힌 데이터베이스 파일 수
ADDITIONAL_DETAILS	BLOB(100K)	나중에 사용하기 위해 예약됨

## MON\_GET\_CONNECTION 테이블 함수 - 연결 메트릭 가져오기

MON\_GET\_CONNECTION 테이블 함수는 하나 이상의 연결에 대한 메트릭을 리턴합니다.

### 구문

```
►► MON_GET_CONNECTION(—application_handle—,—member—)◄◄
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *application\_handle*

메트릭 리턴되는 연결을 식별하는 특정 응용프로그램 핸들을 지정하는 BIGINT 유형의 입력 인수입니다. 인수가 널(NULL)인 경우 모든 연결에 대해 메트릭이 리턴됩니다.

#### *member*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

### 권한 부여

MON\_GET\_CONNECTION 테이블 함수의 EXECUTE 특권

### 예 :

클라이언트에 가장 큰 볼륨의 데이터를 리턴하는 연결을 리턴된 행 수별로 정렬하여 표시합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

APPLICATION_HANDLE	ROWS_RETURNED	TCPIP_SEND_VOLUME
-----	-----	-----
55	6	0

1개의 레코드가 선택되었습니다.

## 사용 시 참고사항

MON\_GET\_CONNECTION 테이블 함수가 리턴한 메트릭은 연결에서 제출된 요청에 대한 모든 메트릭 누적을 나타냅니다. 메트릭은 요청 실행 중에 주기적으로 작업 단위(UOW) 경계에서 롤업됩니다. 따라서 이 테이블 함수가 보고한 값은 최신 롤업 시의 시스템 현재 상태를 반영합니다. 메트릭의 값은 반드시 증가합니다. 시간 간격에 대해 지정된 메트릭 값을 판별하려면 MON\_GET\_CONNECTION 테이블 함수를 사용하여 간격의 시작과 종료 시점의 메트릭을 조회하고 그 차이를 계산하십시오.

요청 메트릭은 데이터베이스 레벨에서의 *mon\_req\_metrics* 데이터베이스 구성 매개변수 및 서비스 수퍼 클래스의 COLLECT REQUEST METRICS절을 통해 제어됩니다. 상위 서비스 수퍼 클래스에서 요청 메트릭을 사용할 수 있는 서비스 서브클래스에서 에이전트가 요청을 처리하거나 전체 데이터베이스에 대해 요청 메트릭 컬렉션을 사용할 수 있는 경우에만 요청에 대해 메트릭이 수집됩니다. 디폴트로 데이터베이스 레벨에서 요청 메트릭을 사용할 수 있습니다. 요청 메트릭이 데이터베이스 레벨에서 사용 불가능하고 서비스 수퍼 클래스에 대해 사용 불가능한 경우, 해당 서비스 수퍼 클래스에 맵핑된 각 연결에 보고된 메트릭은 증가를 중지시킵니다. (또는 요청 메트릭이 데이터베이스 활성화 시간에 사용 불가능한 경우 0이 됩니다.)

**팁:** 연결 수명 동안 연결이 둘 이상의 서비스 수퍼 클래스에 맵핑될 수 있기 때문에 MON\_GET\_CONNECTION 테이블 함수를 통해 보고된 메트릭은 연결을 통해 제출된 모든 요청에 대한 메트릭 서브세트를 나타낼 수도 있습니다. 이는 연결에서 맵핑된 일부 수퍼 클래스에 대해 메트릭 컬렉션을 사용할 수 없는 경우에 발생할 수 있습니다.

MON\_GET\_CONNECTION 테이블 함수는 연결 및 구성원별로 한 개의 데이터 행을 리턴합니다. 구성원(하나 이상의 서비스 클래스)에서의 집계는 수행되지 않습니다. 그렇지만 SQL 쿼리를 사용하여 집계를 수행할 수 있습니다.

## 리턴된 정보

표 96. MON\_GET\_CONNECTION에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
APPLICATION_HANDLE	BIGINT	application_handle - 응용프로그램 핸들
APPLICATION_NAME	VARCHAR(128)	나중에 사용하기 위해 예약됨
APPLICATION_ID	VARCHAR(128)	나중에 사용하기 위해 예약됨
MEMBER	SMALLINT	member - 데이터베이스 구성원
CLIENT_WRKSTNNAME	VARCHAR(255)	CURRENT CLIENT_WRKSTNNAME 특수 레지스터
CLIENT_ACCTNG	VARCHAR(255)	CURRENT CLIENT_ACCTNG 특수 레지스터
CLIENT_USERID	VARCHAR(255)	CURRENT CLIENT_USERID 특수 레지스터
CLIENT_APPLNAME	VARCHAR(255)	CURRENT CLIENT_APPLNAME 특수 레지스터
CLIENT_PID	BIGINT	나중에 사용하기 위해 예약됨

표 96. MON\_GET\_CONNECTION에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
CLIENT_PRDID	VARCHAR (128)	나중에 사용하기 위해 예약됨
CLIENT_PLATFORM	VARCHAR(12)	나중에 사용하기 위해 예약됨
CLIENT_PROTOCOL	VARCHAR(10)	나중에 사용하기 위해 예약됨
SYSTEM_AUTH_ID	VARCHAR (128)	나중에 사용하기 위해 예약됨
SESSION_AUTH_ID	VARCHAR (128)	나중에 사용하기 위해 예약됨
COORD_MEMBER	SMALLINT	나중에 사용하기 위해 예약됨
CONNECTION_START_TIME	TIMESTAMP	나중에 사용하기 위해 예약됨
ACT_ABORTED_TOTAL	BIGINT	act_aborted_total - 중단된 총 활동 수
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 완료된 총 활동 수
ACT_REJECTED_TOTAL	BIGINT	act_rejected_total - 거부된 총 활동 수
AGENT_WAIT_TIME	BIGINT	agent_wait_time - 에이전트 대기 시간
AGENT_WAITS_TOTAL	BIGINT	agent_waits_total - 총 에이전트 대기 수
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
CLIENT_IDLE_WAIT_TIME	BIGINT	client_idle_wait_time - 클라이언트 유휴 대기 시간
DEADLOCKS	BIGINT	deadlocks - 발견된 교착 상태 수
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수

표 96. MON\_GET\_CONNECTION에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
FCM_RECV_VOLUME	BIGINT	fcm_recv_volume - FCM 수신 볼륨
FCM_RECVS_TOTAL	BIGINT	fcm_recvs_total - 총 FCM 수신 수
FCM_SEND_VOLUME	BIGINT	fcm_send_volume - FCM 전송 볼륨
FCM_SENDS_TOTAL	BIGINT	fcm_sends_total - 총 FCM 전송 수
FCM_RECV_WAIT_TIME	BIGINT	fcm_recv_wait_time - FCM 수신 대기 시간
FCM_SEND_WAIT_TIME	BIGINT	fcm_send_wait_time - FCM 전송 대기 시간
IPC_RECV_VOLUME	BIGINT	ipc_recv_volume - 프로세스간 통신 수신 볼륨
IPC_RECV_WAIT_TIME	BIGINT	ipc_recv_wait_time - 프로세스간 통신 수신 대기 시간
IPC_RECVS_TOTAL	BIGINT	ipc_recvs_total - 총 프로세스간 통신 수신 수
IPC_SEND_VOLUME	BIGINT	ipc_send_volume - 프로세스간 통신 전송 볼륨
IPC_SEND_WAIT_TIME	BIGINT	ipc_send_wait_time - 프로세스간 통신 전송 대기 시간
IPC_SENDS_TOTAL	BIGINT	ipc_sends_total - 총 프로세스간 통신 전송 수
LOCK_ESCALS	BIGINT	lock_escalations - 잠금 에스컬레이션 수
LOCK_TIMEOUTS	BIGINT	lock_timeouts - 잠금 시간종료 수
LOCK_WAIT_TIME	BIGINT	lock_wait_time - 잠금 대기 시간
LOCK_WAITS	BIGINT	lock_waits - 잠금 대기 수
LOG_BUFFER_WAIT_TIME	BIGINT	log_buffer_wait_time - 로그 버퍼 대기 시간
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - 전체 로그 버퍼 수
LOG_DISK_WAIT_TIME	BIGINT	log_disk_wait_time - 로그 디스크 대기 시간
LOG_DISK_WAITS_TOTAL	BIGINT	log_disk_waits_total - 총 로그 디스크 대기 수
NUM_LOCKS_HELD	BIGINT	나중에 사용하기 위해 예약됨
RQSTS_COMPLETED_TOTAL	BIGINT	rqsts_completed_total - 완료된 총 요청 수
ROWS_MODIFIED	BIGINT	rows_modified - 수정된 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_RETURNED	BIGINT	rows_returned - 리턴된 행 수
TCPIP_RECV_VOLUME	BIGINT	tcPIP_recv_volume - TCP/IP 수신 볼륨
TCPIP_SEND_VOLUME	BIGINT	tcPIP_send_volume - TCP/IP 전송 볼륨
TCPIP_RECV_WAIT_TIME	BIGINT	tcPIP_recv_wait_time - TCP/IP 수신 대기 시간
TCPIP_RECVS_TOTAL	BIGINT	tcPIP_recvs_total - 총 TCP/IP 수신 수
TCPIP_SEND_WAIT_TIME	BIGINT	tcPIP_send_wait_time - TCP/IP 전송 대기 시간
TCPIP_SENDS_TOTAL	BIGINT	tcPIP_sends_total - 총 TCP/IP 전송 수
TOTAL_APP_RQST_TIME	BIGINT	total_app_rqst_time - 총 응용프로그램 요청 시간
TOTAL_RQST_TIME	BIGINT	total_rqst_time - 총 요청 시간
WLM_QUEUE_TIME_TOTAL	BIGINT	wlm_queue_time_total - 워크로드 관리 프로그램 총 큐 시간

표 96. MON\_GET\_CONNECTION에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
WLM_QUEUE_ASSIGNMENTS_TOTAL	BIGINT	wlm_queue_assignments_total - 워크로드 관리 프로그램 총 큐 지정 수
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 총 CPU 시간
TOTAL_WAIT_TIME	BIGINT	total_wait_time - 총 대기 시간
APP_RQSTS_COMPLETED_TOTAL	BIGINT	app_rqsts_completed_total - 완료된 총 응용프로그램 요청 수
TOTAL_SECTION_SORT_TIME	BIGINT	total_section_sort_time - 전체 섹션 정렬 시간
TOTAL_SECTION_SORT_PROC_TIME	BIGINT	total_section_sort_proc_time - 전체 섹션 정렬 처리 시간
TOTAL_SECTION_SORTS	BIGINT	total_section_sorts - 전체 섹션 정렬 수
TOTAL_SORTS	BIGINT	total_sorts - 전체 정렬 수
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - 포스트 임계값 정렬
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - 포스트 공유 임계값 정렬
SORT_OVERFLOW	BIGINT	sort_overflows - 정렬 오버플로우
TOTAL_COMPILE_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMPILE_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMPILATIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_IMPLICIT_COMPILE_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_IMPLICIT_COMPILE_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_IMPLICIT_COMPILATIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_SECTION_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_SECTION_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_APP_SECTION_EXECUTIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ACT_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ACT_WAIT_TIME	BIGINT	나중에 사용하기 위해 예약됨
ACT_RQSTS_TOTAL	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROUTINE_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROUTINE_INVOCATIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMMIT_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMMIT_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_APP_COMMITS	BIGINT	나중에 사용하기 위해 예약됨
INT_COMMITS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROLLBACK_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROLLBACK_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_APP_ROLLBACKS	BIGINT	나중에 사용하기 위해 예약됨
INT_ROLLBACKS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_RUNSTATS_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_RUNSTATS_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_RUNSTATS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_REORG_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_REORG_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨



표 96. MON\_GET\_CONNECTION에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
TOTAL_REORGS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_LOAD_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_LOAD_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_LOADS	BIGINT	나중에 사용하기 위해 예약됨
CAT_CACHE_INSERTS	BIGINT	나중에 사용하기 위해 예약됨
CAT_CACHE_LOOKUPS	BIGINT	나중에 사용하기 위해 예약됨
PKG_CACHE_INSERTS	BIGINT	나중에 사용하기 위해 예약됨
PKG_CACHE_LOOKUPS	BIGINT	나중에 사용하기 위해 예약됨
THRESH_VIOLATIONS	BIGINT	나중에 사용하기 위해 예약됨
NUM_LW_THRESH_EXCEEDED	BIGINT	나중에 사용하기 위해 예약됨
ADDITIONAL_DETAILS	BLOB(100K)	나중에 사용하기 위해 예약됨

## MON\_GET\_CONNECTION\_DETAILS 테이블 함수 - 연결 메트릭 세부사항 가져오기

MON\_GET\_CONNECTION\_DETAILS 테이블 함수는 하나 이상의 연결에 대해 메트릭 세부사항을 리턴합니다.

### 구문

▶▶—MON\_GET\_CONNECTION\_DETAILS—(—*application\_handle*—,—*member*—)—▶▶

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *application\_handle*

메트릭이 리턴될 연결을 식별하는 특정 응용프로그램 핸들을 지정하는 BIGINT 유형의 입력 인수. 인수가 NULL인 경우, 모든 연결에 대해 메트릭이 리턴됩니다.

#### *member*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

### 권한 부여

MON\_GET\_CONNECTION\_DETAILS 함수에서의 EXECUTE 특권.



예 :

리턴된 행의 순서로 가장 높은 볼륨의 데이터를 클라이언트에 리턴하는 연결을 표시합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

APPLICATION_HANDLE	ROWS_RETURNED	TCPIP_SEND_VOLUME
-----	-----	-----
	21	4 0

1개의 레코드가 선택되었습니다.

## 사용 시 참고사항

MON\_GET\_CONNECTION\_DETAILS 테이블 함수에서 리턴된 메트릭은 연결에서 제출되는 요청에 대한 모든 메트릭의 누적을 나타냅니다. 이 함수는 MON\_GET\_CONNECTION 테이블 함수와 유사합니다.

- MON\_GET\_CONNECTION 테이블 함수는 컬럼 기반의 형식에서 가장 일반적으로 사용되는 메트릭을 리턴하며 메트릭을 검색하는 성능이 가장 효율적인 방법입니다.
- MON\_GET\_CONNECTION\_DETAILS 테이블 함수는 XML 문서 형식에서 사용 가능한 전체 메트릭 세트를 리턴하며 출력 형식에 대해 최대 유연성을 제공합니다. XML 기반 출력은 XML 구문 분석기에서 직접 구문 분석되거나 XMLTABLE 함수를 사용하여 관계형 형식으로 변환될 수 있습니다(예 참조).

메트릭은 요청 실행 중에 주기적으로 작업 단위(UOW) 경계에서 롤업됩니다. 그러므로 이 테이블 함수로 보고된 값은 가장 최근 롤업 시 시스템의 현재 상태를 반영합니다. 메트릭의 값이 정확히 증가됩니다. 시간 간격에 대해 주어진 메트릭 값을 판별하려면 MON\_GET\_CONNECTION\_DETAILS 테이블 함수를 사용하여 간격 시작과 종료 시 메트릭을 쿼리하며 차이를 계산합니다.

요청 메트릭은 데이터베이스 레벨에서의 **mon\_req\_metrics** 데이터베이스 구성 매개변수 및 서비스 수퍼 클래스의 COLLECT REQUEST METRICS절을 통해 제어됩니다. 상위 서비스 수퍼 클래스에서 요청 메트릭을 사용할 수 있는 서비스 서브클래스에서 에이전트가 요청을 처리하거나 전체 데이터베이스에 대해 요청 메트릭 컬렉션을 사용할 수 있는 경우에만 요청에 대해 메트릭이 수집됩니다. 디폴트로 데이터베이스 레벨에서 요

청 메트릭을 사용할 수 있습니다. 요청 메트릭이 데이터베이스 레벨에서 사용 불가능하고 서비스 수퍼 클래스에 대해 사용 불가능한 경우, 해당 서비스 수퍼 클래스에 맵핑된 각 연결에 보고된 메트릭은 증가를 중지시킵니다. (또는 요청 메트릭이 데이터베이스 활성화 시간에 사용 불가능한 경우 0이 됩니다.)

**팁:** 연결 수명 동안 연결이 둘 이상의 서비스 수퍼 클래스에 맵핑될 수 있기 때문에(데이터베이스 레벨에서 콜렉션을 사용할 수 없는 경우) MON\_GET\_CONNECTION 테이블 함수를 통해 보고된 메트릭은 연결을 통해 제출된 모든 요청에 대한 메트릭 서브셋을 나타낼 수도 있습니다. 이는 연결이 맵핑된 일부 수퍼 클래스에 대해 메트릭 콜렉션을 사용할 수 없는 경우에 발생할 수 있습니다.

MON\_GET\_CONNECTION\_DETAILS 테이블 함수는 연결 및 구성원별로 한 개의 데이터 행을 리턴합니다. 구성원(하나 이상의 서비스 클래스)에서의 집계는 수행되지 않습니다. 하지만, SQL 쿼리를 통해 집계가 이루어질 수 있습니다.

DETAILS 컬럼에 리턴된 XML 문서의 스키마는 sqllib/misc/DB2MonRoutines.xsd 파일로 사용 가능합니다. sqllib/misc/DB2MonCommon.xsd 파일에서 더 자세한 내용을 볼 수 있습니다.

## 리턴된 정보

표 97. MON\_GET\_CONNECTION\_DETAILS에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
APPLICATION_HANDLE	BIGINT	application_handle - 응용프로그램 핸들
MEMBER	SMALLINT	member - 데이터베이스 구성원
DETAILS	BLOB(1M)	작업 단위로 메트릭 세부사항을 포함한 XML 문서. 이 문서에서 요소 설명은 399 페이지의 표 98의 내용을 참조하십시오.

다음 예는 DETAILS 컬럼에서 리턴되는 XML 문서의 구조를 표시합니다.

```
<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>
```

전체 스키마에 대해서는 sqllib/misc/DB2MonRoutines.xsd를 참조하십시오.

이 문서는 다음 비원시 XML 유형 정의를 사용합니다.

```
<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>

```

표 98. MON\_GET\_CONNECTION\_DETAILS에 대해 리턴되는 메트릭 세부사항

요소 이름	데이터 유형	설명
application_handle	xs:nonNegativeInteger	application_handle - 응용프로그램 핸들
client_userid	xs:string(255)	CURRENT CLIENT_USERID 특수 레지스터
client_wrkstnname	xs:string(255)	CURRENT CLIENT_WRKSTNNAME 특수 레지스터
client_applname	xs:string(255)	CURRENT CLIENT_APPLNAME 특수 레지스터
client_acctng	xs:string(255)	CURRENT CLIENT_ACCTNG 특수 레지스터
member	xs:nonNegativeInteger	member - 데이터베이스 구성원
act_aborted_total	xs:nonNegativeInteger	act_aborted_total - 중단된 총 활동 수
act_completed_total	xs:nonNegativeInteger	act_completed_total - 완료된 총 활동 수
act_rejected_total	xs:nonNegativeInteger	act_rejected_total - 거부된 총 활동 수
agent_wait_time	xs:nonNegativeInteger	agent_wait_time - 에이전트 대기 시간
agent_waits_total	xs:nonNegativeInteger	agent_waits_total - 총 에이전트 대기 수
app_rqsts_completed_total	xs:nonNegativeInteger	app_rqsts_completed_total - 완료된 총 응용프로그램 요청 수
audit_events_total	xs:nonNegativeInteger	audit_events_total - 총 감사 이벤트 수
audit_subsystem_wait_time	xs:nonNegativeInteger	audit_subsystem_wait_time - 감사 서브시스템 대기 시간
audit_subsystem_waits_total	xs:nonNegativeInteger	audit_subsystem_waits_total - 전체 감사 서브시스템 대기
audit_file_write_wait_time	xs:nonNegativeInteger	audit_file_write_wait_time - 감사 파일 작성 대기 시간
audit_file_writes_total	xs:nonNegativeInteger	audit_file_writes_total - 작성된 총 감사 파일 수
client_idle_wait_time	xs:nonNegativeInteger	client_idle_wait_time - 클라이언트 유휴 대기 시간
교착 상태	xs:nonNegativeInteger	deadlocks - 발견된 교착 상태 수
diaglog_writes_total	xs:nonNegativeInteger	diaglog_writes_total - 도표 로그 전체 작성 수
diaglog_write_wait_time	xs:nonNegativeInteger	diaglog_write_wait_time - 도표 로그 작성 시간
direct_read_time	xs:nonNegativeInteger	direct_read_time - 직접 읽기 시간
direct_write_time	xs:nonNegativeInteger	direct_write_time - 직접 쓰기 시간
direct_read_reqs	xs:nonNegativeInteger	direct_read_reqs - 직접 읽기 요청 수
direct_reads	xs:nonNegativeInteger	direct_reads - 데이터베이스에서 직접 읽기 수
direct_write_reqs	xs:nonNegativeInteger	direct_write_reqs - 직접 쓰기 요청 수
direct_writes	xs:nonNegativeInteger	direct_writes - 데이터베이스에 직접 쓰기 수
fcm_rcv_volume	xs:nonNegativeInteger	fcm_rcv_volume - FCM 수신 볼륨
fcm_rcv_wait_time	xs:nonNegativeInteger	fcm_rcv_wait_time - FCM 수신 대기 시간
fcm_recvs_total	xs:nonNegativeInteger	fcm_recvs_total - 총 FCM 수신 수
fcm_message_rcv_volume	xs:nonNegativeInteger	fcm_message_rcv_volume - FCM 메시지 수신 볼륨
fcm_message_recvs_total	xs:nonNegativeInteger	fcm_message_recvs_total - 총 FCM 메시지 수신 수
fcm_message_rcv_wait_time	xs:nonNegativeInteger	fcm_message_rcv_wait_time - FCM 메시지 수신 대기 시간
fcm_message_send_volume	xs:nonNegativeInteger	fcm_message_send_volume - FCM 메시지 전송 볼륨
fcm_message_send_wait_time	xs:nonNegativeInteger	fcm_message_send_wait_time - FCM 메시지 전송 대기 시간

표 98. MON\_GET\_CONNECTION\_DETAILS에 대해 리턴되는 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명
fcm_message_sends_total	xs:nonNegativeInteger	fcm_message_sends_total - 총 FCM 메시지 전송 수
fcm_send_volume	xs:nonNegativeInteger	fcm_send_volume - FCM 전송 볼륨
fcm_send_wait_time	xs:nonNegativeInteger	fcm_send_wait_time - FCM 전송 대기 시간
fcm_sends_total	xs:nonNegativeInteger	fcm_sends_total - 총 FCM 전송 수
fcm_tq_recv_wait_time	xs:nonNegativeInteger	fcm_tq_recv_wait_time - FCM 테이블 큐 수신 대기 시간
fcm_tq_send_wait_time	xs:nonNegativeInteger	fcm_tq_send_wait_time - FCM 테이블 큐 전송 대기 시간
fcm_tq_recv_volume	xs:nonNegativeInteger	fcm_tq_recv_volume - FCM 테이블 큐 수신 볼륨
fcm_tq_recvs_total	xs:nonNegativeInteger	fcm_tq_recvs_total - FCM 테이블 큐 수신 총계
fcm_tq_send_volume	xs:nonNegativeInteger	fcm_tq_send_volume - FCM 테이블 큐 전송 볼륨
fcm_tq_sends_total	xs:nonNegativeInteger	fcm_tq_sends_total - FCM 테이블 큐 전송 총계
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총계
ipc_recv_volume	xs:nonNegativeInteger	ipc_recv_volume - 프로세스간 통신 수신 볼륨
ipc_recv_wait_time	xs:nonNegativeInteger	ipc_recv_wait_time - 프로세스간 통신 수신 대기 시간
ipc_recvs_total	xs:nonNegativeInteger	ipc_recvs_total - 총 프로세스간 통신 수신 수
ipc_send_volume	xs:nonNegativeInteger	ipc_send_volume - 프로세스간 통신 전송 볼륨
ipc_send_wait_time	xs:nonNegativeInteger	ipc_send_wait_time - 프로세스간 통신 전송 대기 시간
ipc_sends_total	xs:nonNegativeInteger	ipc_sends_total - 총 프로세스간 통신 전송 수
lock_escals	xs:nonNegativeInteger	lock_escals - 잠금 에스컬레이션 수
lock_timeouts	xs:nonNegativeInteger	lock_timeouts - 잠금 시간종료 수
lock_wait_time	xs:nonNegativeInteger	lock_wait_time - 잠금 대기 시간
lock_waits	xs:nonNegativeInteger	lock_waits - 잠금 대기 수
log_buffer_wait_time	xs:nonNegativeInteger	log_buffer_wait_time - 로그 버퍼 대기 시간
log_disk_wait_time	xs:nonNegativeInteger	log_disk_wait_time - 로그 디스크 대기 시간
log_disk_waits_total	xs:nonNegativeInteger	log_disk_waits_total - 총 로그 디스크 대기 수
pool_data_l_reads	xs:nonNegativeInteger	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
pool_data_p_reads	xs:nonNegativeInteger	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
pool_data_writes	xs:nonNegativeInteger	pool_data_writes - 버퍼 풀 데이터 쓰기 수
pool_index_l_reads	xs:nonNegativeInteger	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
pool_index_p_reads	xs:nonNegativeInteger	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
pool_index_writes	xs:nonNegativeInteger	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
pool_read_time	xs:nonNegativeInteger	pool_read_time - 총 버퍼 풀 실제 읽기 시간
pool_temp_data_l_reads	xs:nonNegativeInteger	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
pool_temp_data_p_reads	xs:nonNegativeInteger	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
pool_temp_index_l_reads	xs:nonNegativeInteger	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
pool_temp_index_p_reads	xs:nonNegativeInteger	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
pool_temp_xda_l_reads	xs:nonNegativeInteger	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
pool_temp_xda_p_reads	xs:nonNegativeInteger	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수

표 98. MON\_GET\_CONNECTION\_DETAILS에 대해 리턴되는 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명
pool_write_time	xs:nonNegativeInteger	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
pool_xda_l_reads	xs:nonNegativeInteger	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
pool_xda_p_reads	xs:nonNegativeInteger	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
pool_xda_writes	xs:nonNegativeInteger	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
num_log_buffer_full	xs:nonNegativeInteger	num_log_buffer_full - 전체 로그 버퍼 수
rqsts_completed_total	xs:nonNegativeInteger	rqsts_completed_total - 완료된 총 요청 수
rows_modified	xs:nonNegativeInteger	rows_modified - 수정된 행 수
rows_read	xs:nonNegativeInteger	rows_read - 읽은 행 수
rows_returned	xs:nonNegativeInteger	rows_returned - 리턴된 행 수
tcpip_recv_volume	xs:nonNegativeInteger	tcpip_recv_volume - TCP/IP 수신 볼륨
tcpip_recv_wait_time	xs:nonNegativeInteger	tcpip_recv_wait_time - TCP/IP 수신 대기 시간
tcpip_recvs_total	xs:nonNegativeInteger	tcpip_recvs_total - 총 TCP/IP 수신 수
tcpip_send_volume	xs:nonNegativeInteger	tcpip_send_volume - TCP/IP 전송 볼륨
tcpip_send_wait_time	xs:nonNegativeInteger	tcpip_send_wait_time - TCP/IP 전송 대기 시간
tcpip_sends_total	xs:nonNegativeInteger	tcpip_sends_total - 총 TCP/IP 전송 수
total_app_rqst_time	xs:nonNegativeInteger	total_app_rqst_time - 총 응용프로그램 요청 시간
total_cpu_time	xs:nonNegativeInteger	total_cpu_time - 총 CPU 시간
total_rqst_time	xs:nonNegativeInteger	total_rqst_time - 총 요청 시간
total_wait_time	xs:nonNegativeInteger	total_wait_time - 총 대기 시간
total_section_sort_time	xs:nonNegativeInteger	total_section_sort_time - 전체 섹션 정렬 시간
total_section_sort_proc_time	xs:nonNegativeInteger	total_section_sort_proc_time - 전체 섹션 정렬 처리 시간
total_section_sorts	xs:nonNegativeInteger	total_section_sorts - 전체 섹션 정렬 수
total_sorts	xs:nonNegativeInteger	total_sorts - 전체 정렬 수
post_threshold_sorts	xs:nonNegativeInteger	post_threshold_sorts - 포스트 임계값 정렬
post_shrthreshold_sorts	xs:nonNegativeInteger	post_shrthreshold_sorts - 포스트 공유 임계값 정렬
sort_overflows	xs:nonNegativeInteger	sort_overflows - 정렬 오버플로우 수
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총계
wlm_queue_time_total	xs:nonNegativeInteger	wlm_queue_time_total - 워크로드 관리 프로그램 총 큐 시간
wlm_queue_assignments_total	xs:nonNegativeInteger	wlm_queue_assignments_total - 워크로드 관리 프로그램 총 큐 지정 수

## MON\_GET\_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기

MON\_GET\_CONTAINER 테이블 함수는 하나 이상의 테이블 스페이스 컨테이너에 대한 모니터 메트릭을 리턴합니다.

## 구문

```
►► MON_GET_CONTAINER(—tbsp_name—,—member—)◄◄
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *tbsp\_name*

이 함수 호출 시 현재 연결된 임의 데이터베이스와 동일한 데이터베이스에 유효한 테이블 스페이스 이름을 지정하는 VARCHAR (128) 유형의 입력 인수입니다. 인수가 널(NULL) 또는 비어 있는 문자열인 경우 데이터베이스의 모든 테이블 스페이스에서 모든 컨테이너에 대해 메트릭이 리턴됩니다.

### *member*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

## 권한 부여

MON\_GET\_CONTAINER 테이블 함수의 EXECUTE 특권

## 예 :

예 1: 읽은 시간이 가장 긴 모든 데이터베이스 구성원의 컨테이너가 나열됩니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

```
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 ...
```

3개의 레코드가 선택되었습니다.

쿼리의 출력(계속).

```
... TBSP_NAME          POOL_READ_TIME  
... -----  
... SYSCATSPACE              597  
... USERSPACE1              42  
... TEMPSPACE1              0
```

예 2: 액세스할 수 없는 모든 컨테이너가 나열됩니다.

```
SELECT varchar(container_name, 70) as container_name
FROM TABLE(MON_GET_CONTAINER('',-1)) AS t
WHERE accessible = 0
```

다음은 이 쿼리의 출력 예입니다.

```
CONTAINER_NAME
-----
0개의 레코드가 선택되었습니다.
```

예 3: 컨테이너 파일 시스템의 이용도를 가장 높은 이용률 순으로 나열합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

```
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 ...
```

3개의 레코드가 선택되었습니다.

쿼리의 출력(계속).

FS_ID	FS_USED_SIZE	FS_TOTAL_SIZE	UTILIZATION
64768	106879311872	317068410880	33.70
64768	106879311872	317068410880	33.70
64768	106879311872	317068410880	33.70

## 사용 시 참고사항

MON\_GET\_CONTAINER 테이블 함수는 컨테이너 및 데이터베이스 구성원별로 한 개의 데이터 행을 리턴합니다. 지정된 테이블 스페이스의 모든 컨테이너 또는 데이터베이스의 모든 컨테이너에 대해 데이터를 리턴할 수 있습니다. 데이터베이스 파티션 사이의 집계는 수행되지 않습니다. 그렇지만 SQL 쿼리를 사용하여 집계를 수행할 수 있습니다.

이 함수에 의해 수집된 메트릭스는 mon\_obj\_metrics 구성 매개변수를 사용하여 데이터베이스 레벨에서 제어됩니다. 디폴트로 메트릭 콜렉션이 사용 가능합니다.



## 리턴된 정보

표 99. MON\_GET\_CONTAINER에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TBSP_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
CONTAINER_NAME	VARCHAR(256)	container_name - 컨테이너 이름
CONTAINER_ID	BIGINT	container_id - 컨테이너 ID
MEMBER	SMALLINT	member - 데이터베이스 구성원
CONTAINER_TYPE	VARCHAR(16)	container_type - 컨테이너 유형. 이는 sqlutil.h의 정의 중 하나를 기반으로 하는 텍스트 ID로 다음 중 하나입니다. <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 - 스트라이프 세트
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
PAGES_READ	BIGINT	pages_read - 읽은 페이지 수
PAGES_WRITTEN	BIGINT	pages_written - 쓴 페이지 수
VECTORED_IOS	BIGINT	vectored_ios - 벡터 입출력 요청 수
PAGES_FROM_VECTORED_IOS	BIGINT	pages_from_vectored_ios - 벡터 입출력으로 읽은 총 페이지 수
BLOCK_IOS	BIGINT	block_ios - 블록 입출력 요청 수
PAGES_FROM_BLOCK_IOS	BIGINT	pages_from_block_ios - 블록 입출력으로 읽은 총 페이지 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
TOTAL_PAGES	BIGINT	container_total_pages - 컨테이너의 전체 페이지 수
USABLE_PAGES	BIGINT	container_usable_pages - 컨테이너에서 사용할 수 있는 페이지 수
ACCESSIBLE	SMALLINT	container_accessible - 컨테이너의 액세스 가능성
FS_ID	VARCHAR(22)	fs_id - 고유 파일 시스템 ID 번호
FS_TOTAL_SIZE	BIGINT	fs_total_size - 파일 시스템의 전체 크기
FS_USED_SIZE	BIGINT	fs_used_size - 파일 시스템에서 사용된 스페이스 크기
ADDITIONAL_DETAILS	BLOB(100K)	향후에 사용하기 위해 예약되어 있습니다.



## MON\_GET\_EXTENT\_MOVEMENT\_STATUS - Extent 이동 진행 가져오기

MON\_GET\_EXTENT\_MOVEMENT\_STATUS 테이블 함수는 Extent 이동 조작의 상태를 리턴합니다.

### 구문

▶▶—MON\_GET\_EXTENT\_MOVEMENT\_STATUS—(—*tbasp\_name*—,—*member*—)—▶▶

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *tbasp\_name*

쿼리할 테이블 스페이스를 지정하는 VARCHAR

(128) 유형의 입력 인수. 인수 값이 널(NULL)인 경우, 함수는 모든 테이블 스페이스에 대한 정보를 리턴합니다.

#### *member*

현재 연결된 데이터베이스와 동일한 인스턴스 내의 유효한 구성원을 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. 인수 값이 널(NULL)인 경우, -1이 내재적으로 설정됩니다.

### 권한 부여

MON\_GET\_EXTENT\_MOVEMENT\_STATUS 함수의 EXECUTE 특권입니다.

### 예 :

모든 테이블 스페이스의 현재 Extent 진행에 대한 모든 정보를 다음과 같이 검색합니다.

```
SELECT * FROM TABLE(SYSPROC.MON_GET_EXTENT_MOVEMENT_STATUS('', -1))
```

다음은 위 쿼리로부터의 출력 예입니다.

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

5개의 레코드가 선택되었습니다.

쿼리의 출력이 계속됨.

```

... NUM_EXTENTS_LEFT TOTAL_MOVE_TIME ADDITIONAL_DETAILS
... -----
...          -1          -1          -
...          -1          -1          -
...          -1          -1          -
...          4           0           -
...          -1          -1          -

```

## 리턴되는 정보

표 100. MON\_GET\_EXTENT\_MOVEMENT\_STATUS에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TBSP_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
MEMBER	SMALLINT	member - 이 정보를 수집한 구성원
CURRENT_EXTENT	INTEGER	current_extent - 이동 중인 현재 Extent
LAST_EXTENT	INTEGER	last_extent - 이동한 마지막 Extent
NUM_EXTENTS_MOVED	INTEGER	num_extents_moved - 이 Extent 이동 조작 중에 현재까지 이동한 Extent 수
NUM_EXTENTS_LEFT	INTEGER	num_extents_left - 이 Extent 이동 조작 중에 이동 대기 중인 Extent 수
TOTAL_MOVE_TIME	BIGINT	total_move_time - 이동한 전체 Extent에 대한 총 이동 시간(밀리초)
ADDITIONAL_DETAILS	BLOB(100K)	나중에 사용하기 위해 예약됩니다.

## MON\_GET\_INDEX 테이블 함수 - 인덱스 메트릭 가져오기

MON\_GET\_INDEX 테이블 함수는 하나 이상의 인덱스에 대한 메트릭을 리턴합니다.

### 구문

```

▶▶ MON_GET_INDEX(—tabschema—, —tablename—, —member—)

```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *tabschema*

이 함수 호출 시 현재 연결된 동일한 데이터베이스에서 유효한 테이블 스키마 이름을 지정하는 VARCHAR (128) 유형의 입력 인수. 인수가 널(null)이거나 빈 문자열이면 데이터베이스의 모든 스키마에 대해 테이블의 인덱스가 검색됩니다. 인수가 지정되면 지정된 스키마의 테이블에 대해서만 메트릭이 리턴됩니다.

#### *tablename*

이 함수 호출 시 현재 연결된 동일한 데이터베이스에서 유효한 테이블 이름을 지정

하는 VARCHAR

(128) 유형의 입력 인수. 지정된 테이블의 모든 인덱스에 대해 메트릭이 리턴됩니다. 인수가 널(null)이거나 빈 문자열이면 데이터베이스의 모든 테이블의 모든 인덱스에 대해 메트릭이 검색됩니다.

*member*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에 -1을 지정하거나 모든 데이터베이스 구성원에 -2를 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

### 권한 부여

MON\_GET\_INDEX 함수에서의 EXECUTE 특권.

### 예 :

마지막 데이터베이스 활성화 이후 DMEXT002.TABLE1 테이블에서 가장 자주 사용된 인덱스를 식별합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

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

6개의 레코드가 선택되었습니다.

### 사용 시 참고사항

MON\_GET\_INDEX 테이블 함수는 인덱스 및 데이터베이스 구성원별로 한 개의 데이터 행을 리턴합니다. 파티션된 인덱스를 사용 중인 경우에는 데이터베이스 구성원별로 각 인덱스 파티션에 대해 한 개의 행이 리턴됩니다. 데이터베이스 구성원간의 집계는 수행되지 않습니다. 그러나 위 예에서 보는 것과 같이 SQL 쿼리를 통해 집계가 이루어질 수 있습니다.

데이터베이스가 활성화된 이후 액세스했던 테이블의 인덱스에 대해 메트릭만 리턴됩니다. 현재 데이터베이스 활성화 이후 모든 카운터가 데이터를 나타냅니다. 예를 들어, *pseudo\_empty\_pages* 카운터는 데이터베이스가 활성화된 이후 비어 있는 유사(pseudo) 페이지처럼 식별되는 페이지의 번호입니다. 인덱스에서 비어 있는 유사 페이지의 현재 번호가 아닙니다.

메트릭은 항상 사용 가능합니다. 이 함수를 통해 테이블 메트릭에 액세스하기 위해 시스템 모니터 스위치를 켤 필요는 없습니다.

## 리턴된 정보

표 101. MON\_GET\_INDEX에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TABSCHEMA	VARCHAR (128)	table_schema - 테이블 스키마 이름
TABNAME	VARCHAR (128)	table_name - 테이블 이름
IID	SMALLINT	iid - 인덱스 ID
MEMBER	SMALLINT	member - 데이터베이스 구성원
DATA_PARTITION_ID	INTEGER	data_partition_id - 데이터 파티션 ID. 인덱스가 파티션되지 않으면, NULL이 리턴됩니다.
NLEAF	BIGINT	nleaf - 리프 페이지 수
NLEVELS	SMALLINT	nlevels - 인덱스 레벨 수
INDEX_SCANS	BIGINT	index_scans - 인덱스 스캔
INDEX_ONLY_SCANS	BIGINT	index_only_scans - 인덱스 전용 스캔
KEY_UPDATES	BIGINT	key_updates - 키 갱신
INCLUDE_COL_UPDATES	BIGINT	include_col_updates - Include 컬럼 갱신
PSEUDO_DELETES	BIGINT	pseudo_deletes - 의사(Pseudo) 삭제
DEL_KEYS_CLEANED	BIGINT	del_keys_cleaned - 의사(Pseudo) 삭제 키 정리
ROOT_NODE_SPLITS	BIGINT	root_node_splits - 루트 노드 분할
INT_NODE_SPLITS	BIGINT	int_node_splits - 중간 노드 분할
BOUNDARY_LEAF_NODE_SPLITS	BIGINT	boundary_leaf_node_splits - 경계 리프 노드 분할
NONBOUNDARY_LEAF_NODE_SPLITS	BIGINT	nonboundary_leaf_node_splits - 비경계 리프 노드 분할
PAGE_ALLOCATIONS	BIGINT	page_allocations - 페이지 할당
PSEUDO_EMPTY_PAGES	BIGINT	pseudo_empty_pages - 의사(Pseudo) 비어 있는 페이지
EMPTY_PAGES_REUSED	BIGINT	empty_pages_reused - 재사용된 비어 있는 페이지
EMPTY_PAGES_DELETED	BIGINT	empty_pages_deleted - 삭제된 비어 있는 페이지
PAGES_MERGED	BIGINT	pages_merged - 병합된 페이지
ADDITIONAL_DETAILS	BLOB(100K)	나중에 사용하기 위해 예약됨

## MON\_GET\_PKG\_CACHE\_STMT 테이블 함수 - 패키지 캐시에서 SQL문 활동 메트릭 가져오기

MON\_GET\_PKG\_CACHE\_STMT 테이블 함수는 데이터베이스 패키지 캐시에서 정적 및 동적 SQL문 모두의 point-in-time 뷰를 리턴합니다.

### 구문

```
▶▶ MON_GET_PKG_CACHE_STMT (—section_type—, —————▶  
▶ —executable_id—, —search_args—, —member—) —————▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *section\_type*

리턴된 명령문의 정보 유형을 지정하는 CHAR(1) 유형의 입력 인수("D" 또는 "S"). 인수가 NULL 또는 비어 있는 문자열이면, 모든 SQL문에 대해 정보가 리턴됩니다. 대소문자를 구분하지 않음: 『D』는 동적을 의미하고 『S』는 정적을 의미합니다.

#### *executable\_id*

데이터베이스 패키지 캐시의 고유 섹션을 지정하는 비트 데이터용 VARCHAR(32) 유형의 입력 인수. 널(NULL) 값이 지정된 경우, 정보가 모든 SQL문에 대해 리턴됩니다. *executable\_id*가 지정되면 *section\_type* 인수가 무시됨에 유의하십시오. 예를 들어, *executable\_id*가 동적 명령문으로 지정된 경우, *section\_type*이 정적("S")으로 지정되더라도 동적 명령문 세부사항은 이 테이블 함수로 리턴됩니다.

#### *search\_args*

나중에 사용하기 위해 현재 예약된 CLOB(1K) 유형의 입력 매개변수

#### *member*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 구성원을 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. 널(NULL) 값이 지정된 경우, -1이 내재적으로 설정됩니다.

### 권한 부여

MON\_GET\_PKG\_CACHE\_STMT 함수에 대한 EXECUTE 특권.

### 예 :

평균 CPU 시간으로 정렬된 데이터베이스 패키지 캐시에서 모든 동적 SQL문을 나열합니다.

```

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

```

이 쿼리의 출력 예는 다음과 같습니다.

MEMBER	SECTION_TYPE	AVG_CPU_TIME	EXECUTABLE_ID
0 D		754	x'0100000000000007A0000000000000000000000000000000020020081126171554951791'
0 D		2964	x'0100000000000000790000000000000000000000000000000020020081126171533551120'
0 D		5664	x'01000000000000007C0000000000000000000000000000000020020081126171720728997'
0 D		5723	x'01000000000000007B0000000000000000000000000000000020020081126171657272914'
0 D		9762	x'01000000000000007D0000000000000000000000000000000020020081126172409987719'

5개의 레코드가 선택되었습니다.

주: 컴파일 환경을 빌드하고 구성원 간 명령문 텍스트(2MB까지 가능)를 전송하는 데 시간이 더 걸립니다. 패키지 캐시에서 전체 명령문 목록 검색 시 성능을 향상시키려면 **STMT\_TEXT** 및 **COMP\_ENV\_DESC** 컬럼을 선택하지 마십시오.

위의 출력과 함께, *executable\_id*를 사용하여 비용이 가장 많이 드는 명령문에 대한 세부사항을 찾을 수 있습니다(평균 CPU 시간으로 환산하여).

```

db2 SELECT STMT_TEXT FROM TABLE(MON_GET_PKG_CACHE_STMT)
      (null, x'0100000000000007D0000000000000000000000000000000020020081126172409987719', null, -2)

```

```

STMT_TEXT
-----
SELECT * FROM EMPLOYEE

```

다른 예를 들자면, 사용자인 Alex는 **COLLECT ACTIVITY METRICS**가 설정된 워크로드 A와 연관된 연결을 가지고 있다고 가정해 보십시오. 다른 사용자인 Brent는 **COLLECT ACTIVITY METRICS**가 **NONE**으로 설정된 워크로드 B에 연관되어 있습니다. 또한 데이터베이스 **mon\_act\_metrics** 구성 매개변수가 **NONE**으로 설정되어 있습니다. Brent가 다음 쿼리를 실행하면,

```
SELECT count(*) FROM syscat.tables
```

모든 메트릭은 0으로 리턴되고 **num\_exec\_with\_metrics** 값도 0입니다. 그런 다음, Alex가 그 후에 동일한 명령문을 실행하고 이번에는 명령문 실행에 대해 메트릭이 수집되며 **num\_exec\_with\_metrics**가 증분됩니다. 그래서, Brent와 Alex가 해당 명령문을 실행한 후에 이 쿼리의 결과는 다음과 같습니다.

```

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%'

```

결과는 **SELECT**문이 두 번 실행되고 실행 중 하나에서는 활동 메트릭이 수집되었음을 보여줍니다.

```

NUM_EXECUTIONS NUM_EXEC_WITH_METRICS STMT_TEXT
-----
2 1 SELECT count(*) FROM syscat.tables

```

1개의 레코드가 선택되었습니다.

## 사용 시 참고사항

MON\_GET\_PKG\_CACHE\_STMT 테이블 함수는 데이터베이스 패키지 캐시에서 정적 및 동적 SQL문 모두의 point-in-time 뷰를 리턴합니다. 이를 사용하면 특정 SQL문에 대한 집계 메트릭을 조사하여 쿼리 성능 저하 원인을 신속하게 판별할 수 있습니다. 리턴된 메트릭은 각각의 명령문 실행 중에 수집된 메트릭 집계입니다.

또한 이를 사용하면 캐시된 각 섹션 동작, 다른 명령문과의 연관성을 비교하여 실행 비용 면에서 가장 비용이 많이 드는 섹션이나 명령문을 식별할 수 있습니다.

이 함수로 보고된 활동 메트릭은 활동의 실행 종료 시 데이터베이스 캐시로 롤업됩니다.

모든 명령문의 실행에 대한 메트릭 수집은 워크로드에서 COLLECT ACTIVITY METRICS절 또는 데이터베이스 레벨에서 **mon\_act\_metrics** 데이터베이스 구성 매개변수를 통해 제어합니다. 활동 메트릭이 사용 가능한 워크로드나 데이터베이스와 연결된 연결에서 명령문을 제출한 경우에는 명령문 실행에 대해서만 메트릭이 수집됩니다. MON\_GET\_PKG\_CACHE\_STMT 함수에서 리턴된 **num\_exec\_with\_metrics** 요소는 메트릭이 수집되고 보고된 메트릭 집계에 사용된 명령문의 실행 수를 나타냅니다. 명령문 실행에 대해 메트릭이 수집되지 않은 경우에는 **num\_exec\_with\_metrics** 요소가 0이고 모든 메트릭 값이 0으로 리턴됩니다.

## 리턴된 정보

표 102. MON\_GET\_PKG\_CACHE\_STMT로 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
MEMBER	SMALLINT	member - 데이터베이스 구성원
SECTION_TYPE	CHAR(1)	section_type - 섹션 유형 표시기.
INSERT_TIMESTAMP	TIMESTAMP	insert_timestamp - 명령문 삽입 시간소인
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	executable_id - 실행 가능 ID.
PACKAGE_NAME	VARCHAR (128)	package_name - 패키지 이름. 이 출력은 정적 SQL문에만 유효합니다. 명령문이 동적인 경우 널(NULL) 값이 리턴됩니다.
PACKAGE_SCHEMA	VARCHAR (128)	package_schema - 패키지 스키마. 이 출력은 정적 SQL문에만 유효합니다. 명령문이 동적인 경우 널(NULL) 값이 리턴됩니다.

표 102. MON\_GET\_PKG\_CACHE\_STMT로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
PACKAGE_VERSION_ID	VARCHAR(64)	package_version_id - 패키지 버전. 이 출력은 정적 SQL문에만 유효합니다. 명령문이 동적인 경우 또는 정적 명령문에 대한 패키지 버전을 지정하지 않은 경우 널(NULL) 값이 리턴됩니다. 패키지가 작성되었을 때 사용자가 패키지 버전 ID를 지정하지 않으면 정적 문자열에 비어 있는 문자열이 리턴됩니다.
SECTION_NUMBER	BIGINT	section_number - 섹션 번호. 명령문이 동적인 경우 널(NULL) 값이 리턴됩니다.
EFFECTIVE_ISOLATION	CHAR(2)	effective_isolation - 유효 분리. 섹션에 유효한 분리 값이며, 컴파일 시간에 원래 요청한 것과 다를 수 있습니다.
NUM_EXECUTIONS	BIGINT	num_executions - 명령문 실행
NUM_EXEC_WITH_METRICS	BIGINT	num_exec_with_metrics - 메트릭이 수집된 실행 수.
PREP_TIME	BIGINT	prep_time - 준비 시간. 동적 SQL문에 대해서만 PREP_TIME이 유효합니다. PREP_TIME은 정적 SQL문에 0(영)으로 보고됩니다.
TOTAL_ACT_TIME	BIGINT	total_act_time - 전체 활동 시간
TOTAL_ACT_WAIT_TIME	BIGINT	total_act_wait_time - 전체 활동 대기 시간
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 총 CPU 시간
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
LOCK_WAIT_TIME	BIGINT	lock_wait_time - 잠금 대기 시간
TOTAL_SECTION_SORT_TIME	BIGINT	total_section_sort_time - 전체 섹션 정렬 시간
TOTAL_SECTION_SORT_PROC_TIME	BIGINT	total_section_sort_proc_time - 전체 섹션 정렬 처리 시간
TOTAL_SECTION_SORTS	BIGINT	total_section_sorts - 전체 섹션 정렬
LOCK_ESCALS	BIGINT	lock_escals - 잠금 에스컬레이션 수
LOCK_WAITS	BIGINT	lock_waits - 잠금 대기 수
ROWS_MODIFIED	BIGINT	rows_modified - 수정된 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_RETURNED	BIGINT	rows_returned - 리턴된 행 수
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수



표 102. MON\_GET\_PKG\_CACHE\_STMT로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
TOTAL_SORTS	BIGINT	total_sorts - 전체 정렬 수
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - 포스트 임계값 정렬
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - 포스트 공유 임계값 정렬
SORT_OVERFLOWS	BIGINT	sort_overflows - 정렬 오버플로우
WLM_QUEUE_TIME_TOTAL	BIGINT	wlm_queue_time_total - 워크로드 관리 프로그램 총 큐 시간
WLM_QUEUE_ASSIGNMENTS_TOTAL	BIGINT	wlm_queue_assignments_total - 워크로드 관리 프로그램 총 큐 지정 수
DEADLOCKS	BIGINT	deadlocks - 발견된 교착 상태 수
FCM_RECV_VOLUME	BIGINT	fcm_recv_volume - FCM 수신 볼륨
FCM_RECVS_TOTAL	BIGINT	fcm_recvs_total - 총 FCM 수신 수
FCM_SEND_VOLUME	BIGINT	fcm_send_volume - FCM 전송 볼륨
FCM_SENDS_TOTAL	BIGINT	fcm_sends_total - 총 FCM 전송 수
FCM_RECV_WAIT_TIME	BIGINT	fcm_recv_wait_time - FCM 수신 대기 시간
FCM_SEND_WAIT_TIME	BIGINT	fcm_send_wait_time - FCM 전송 대기 시간
LOCK_TIMEOUTS	BIGINT	lock_timeouts - 잠금 시간종료 수
LOG_BUFFER_WAIT_TIME	BIGINT	log_buffer_wait_time - 로그 버퍼 대기 시간
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - 전체 로그 버퍼 수
LOG_DISK_WAIT_TIME	BIGINT	log_disk_wait_time - 로그 디스크 대기 시간
LOG_DISK_WAITS_TOTAL	BIGINT	log_disk_waits_total - 총 로그 디스크 대기 수
LAST_METRICS_UPDATE	TIMESTAMP	메트릭으로 마지막 갱신한 시간소인. 나중에 사용하기 위해 예약됨
NUM_COORD_EXEC	BIGINT	나중에 사용하기 위해 예약됨
NUM_COORD_EXEC_WITH_METRICS	BIGINT	나중에 사용하기 위해 예약됨
VALID	CHAR(1)	valid - 섹션 유효성 표시기.
TOTAL_ROUTINE_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROUTINE_INVOCATIONS	BIGINT	나중에 사용하기 위해 예약됨

표 102. MON\_GET\_PKG\_CACHE\_STMT로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
ROUTINE_ID	BIGINT	나중에 사용하기 위해 예약됨
STMT_TYPE_ID	VARCHAR(32)	나중에 사용하기 위해 예약됨
QUERY_COST_ESTIMATE	BIGINT	나중에 사용하기 위해 예약됨
STMT_PKG_CACHE_ID	BIGINT	나중에 사용하기 위해 예약됨
COORD_STMT_EXEC_TIME	BIGINT	나중에 사용하기 위해 예약됨
STMT_EXEC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_SECTION_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_SECTION_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROUTINE_NON_SECT_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROUTINE_NON_SECT_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
STMT_TEXT	CLOB(2MB)	stmt_text - SQL문 텍스트
COMP_ENV_DESC	BLOB(10K)	comp_env_desc - 컴파일 환경 핸들. 기존 COMPILATION_ENV 테이블 함수를 사용하여 필요한 경우 특정 명령문의 자세한 컴파일 환경을 가져올 수 있습니다.
ADDITIONAL_DETAILS	BLOB(100K)	추후 추가 메트릭을 위해 예약됨.

## MON\_GET\_SERVICE\_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기

MON\_GET\_SERVICE\_SUBCLASS 테이블 함수는 하나 이상의 가져오기에 대한 메트릭을 리턴합니다.

### 구문

```
▶▶ MON_GET_SERVICE_SUBCLASS(—service_superclass_name—, —————▶
▶—service_subclass_name—, —member—)————▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### service\_superclass\_name

이 함수 호출 시 현재 연결된 데이터베이스에서 유효한 서비스 슈퍼 클래스 이름을 지정하는 VARCHAR (128) 유형의 입력 인수입니다. 인수가 널(NULL) 또는 비어 있는 문자열인 경우 데이터베이스의 모든 슈퍼 클래스에 대해 메트릭이 검색됩니다.

#### service\_subclass\_name

이 함수 호출 시 현재 연결된 데이터베이스에서 유효한 서비스 서브클래스 이름을

지정하는 VARCHAR

(128) 유형의 입력 인수입니다. 인수가 널(NULL) 또는 비어 있는 문자열인 경우 데이터베이스의 모든 서브클래스에 대해 메트릭이 검색됩니다.

*member*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

### 권한 부여

MON\_GET\_SERVICE\_SUBCLASS 테이블 함수의 EXECUTE 특권

### 예 :

각 서비스 클래스에 대해 사용된 전체 CPU 시간 및 처리된 전체 요청 수를 CPU 사용순으로 표시합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

SERVICE_SUPERCLASS	SERVICE_SUBCLASS	...
-----	-----	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	...
SYSDEFAULTMAINTENANCECLASS	SYSDEFAULTSUBCLASS	...
SYSDEFAULTSYSTEMCLASS	SYSDEFAULTSUBCLASS	...

3개의 레코드가 선택되었습니다.

쿼리의 출력(계속).

...	TOTAL_CPU	TOTAL_RQSTS
...	-----	-----
...	967673	100
.. .	0	0
...	0	0

### 사용 시 참고사항

MON\_GET\_SERVICE\_SUBCLASS 테이블 함수가 리턴한 메트릭은 표시된 서비스 서브클래스에서 실행된 요청에 대한 전체 메트릭 누적을 나타냅니다. 메트릭은 요청 실행 중에 주기적으로 작업 단위(UOW) 경계에서 서비스 클래스로 롤업됩니다. 따라서 이 테이블 함수가 보고한 값은 최신 롤업 시의 시스템 현재 상태를 반영합니다. 메트릭의 값은 반드시 증가합니다. 시간 간격에 대해 지정된 메트릭 값을 판별하려면

MON\_GET\_SERVICE\_SUBCLASS 테이블 함수를 사용하여 간격의 시작과 종료 시점의 메트릭을 쿼리하고 그 차이를 계산하십시오.

요청 메트릭은 데이터베이스 레벨에서의 *mon\_req\_metrics* 데이터베이스 구성 매개변수 및 서비스 수퍼 클래스의 COLLECT REQUEST METRICS절을 통해 제어됩니다. 상위 서비스 수퍼 클래스에서 요청 메트릭을 사용할 수 있는 서비스 서브클래스에서 에이전트가 요청을 처리하거나 전체 데이터베이스에 대해 요청 메트릭 컬렉션을 사용할 수 있는 경우에만 요청에 대해 메트릭이 수집됩니다. 디폴트로 데이터베이스 레벨에서 요청 메트릭을 사용할 수 있습니다. 요청 메트릭이 데이터베이스 레벨에서 사용 불가능하고 서비스 수퍼 클래스에 대해 사용 불가능한 경우, 해당 서비스 수퍼 클래스에 맵핑된 각 연결에 보고된 메트릭은 증가를 중지시킵니다. (또는 요청 메트릭이 데이터베이스 활성화 시간에 사용 불가능한 경우 0이 됩니다.)

MON\_GET\_SERVICE\_SUBCLASS 테이블 함수는 서비스 서브클래스 및 구성원별로 한 개의 데이터 행을 리턴합니다. 서비스 클래스(구성원) 또는 구성원(하나 이상의 서비스 클래스)에서는 집계는 수행되지 않습니다. 그렇지만 예에서 표시된 대로 SQL 쿼리를 사용하여 집계를 수행할 수 있습니다. 입력 매개변수 모두 AND가 됩니다. 따라서 충돌하는 입력 매개변수를 지정하면(예: 수퍼 클래스 이름 SUPA와 수퍼 클래스 SUPA의 서브클래스가 아닌 서브클래스 이름 SUBB) 행이 리턴되지 않습니다.

**팁:** 요청은 둘 이상의 서비스 서브클래스에서 실행될 수도 있습니다. 예를 들어 REMAP ACTIVITY 조치가 포함된 워크로드 관리 프로그램(WLM) 임계값을 사용하는 경우에 요청이 임의의 서비스 서브클래스에서 다른 서비스 서브클래스로 맵핑되면 이런 상황이 발생할 수 있습니다. 요청이 실행되는 각 서비스 서브클래스에 대해 메트릭이 갱신되는데 시간이 소요되지만 요청이 완료된 서비스 서브클래스에 대해 요청 카운터가 증분됩니다. 그러므로 단일 서브클래스에 대한 평균 요청 시간을 분석하지 마십시오. 활동이 맵핑될 수 있는 모든 서브클래스가 다른 서브클래스와 함께 분석됩니다. 예를 들어 서비스 서브클래스 A에서 서비스 서브클래스 B로 활동을 맵핑할 수 있는 임계값이 존재한다면, 평균 요청을 계산할 때 서비스 서브클래스 A와 B에 대한 카운터와 메트릭을 집계하고 집계를 사용하여 평균을 계산해야 합니다.

## 리턴된 정보

표 103. MON\_GET\_SERVICE\_SUBCLASS에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	service_superclass_name - 서비스 수퍼 클래스 이름
SERVICE_SUBCLASS_NAME	VARCHAR (128)	service_subclass_name - 서비스 서브클래스 이름
SERVICE_CLASS_ID	INTEGER	service_class_id - 서비스 클래스 ID
MEMBER	SMALLINT	member - 데이터베이스 구성원
ACT_ABORTED_TOTAL	BIGINT	act_aborted_total - 중단된 총 활동 수

표 103. MON\_GET\_SERVICE\_SUBCLASS에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 완료된 총 활동 수
ACT_REJECTED_TOTAL	BIGINT	act_rejected_total - 거부된 총 활동 수
AGENT_WAIT_TIME	BIGINT	agent_wait_time - 에이전트 대기 시간
AGENT_WAITS_TOTAL	BIGINT	agent_waits_total - 총 에이전트 대기 수
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
CLIENT_IDLE_WAIT_TIME	BIGINT	client_idle_wait_time - 클라이언트 유휴 대기 시간
DEADLOCKS	BIGINT	deadlocks - 발견된 교착 상태 수
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
FCM_RECV_VOLUME	BIGINT	fcm_recv_volume - FCM 수신 볼륨
FCM_RECVS_TOTAL	BIGINT	fcm_recvs_total - 총 FCM 수신 수
FCM_SEND_VOLUME	BIGINT	fcm_send_volume - FCM 전송 볼륨
FCM_SENDS_TOTAL	BIGINT	fcm_sends_total - 총 FCM 전송 수

표 103. MON\_GET\_SERVICE\_SUBCLASS에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
FCM_RECV_WAIT_TIME	BIGINT	fcm_recv_wait_time - FCM 수신 대기 시간
FCM_SEND_WAIT_TIME	BIGINT	fcm_send_wait_time - FCM 전송 대기 시간
IPC_RECV_VOLUME	BIGINT	ipc_recv_volume - 프로세스간 통신 수신 볼륨
IPC_RECV_WAIT_TIME	BIGINT	ipc_recv_wait_time - 프로세스간 통신 수신 대기 시간
IPC_RECVS_TOTAL	BIGINT	ipc_recvs_total - 총 프로세스간 통신 수신 수
IPC_SEND_VOLUME	BIGINT	ipc_send_volume - 프로세스간 통신 전송 볼륨
IPC_SEND_WAIT_TIME	BIGINT	ipc_send_wait_time - 프로세스간 통신 전송 대기 시간
IPC_SENDS_TOTAL	BIGINT	ipc_sends_total - 총 프로세스간 통신 전송 수
LOCK_ESCALS	BIGINT	lock_escalations - 잠금 에스컬레이션 수
LOCK_TIMEOUTS	BIGINT	lock_timeouts - 잠금 시간종료 수
LOCK_WAIT_TIME	BIGINT	lock_wait_time - 잠금 대기 시간
LOCK_WAITS	BIGINT	lock_waits - 잠금 대기 수
LOG_BUFFER_WAIT_TIME	BIGINT	log_buffer_wait_time - 로그 버퍼 대기 시간
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - 전체 로그 버퍼 수
LOG_DISK_WAIT_TIME	BIGINT	log_disk_wait_time - 로그 디스크 대기 시간
LOG_DISK_WAITS_TOTAL	BIGINT	log_disk_waits_total - 총 로그 디스크 대기 수
RQSTS_COMPLETED_TOTAL	BIGINT	rqsts_completed_total - 완료된 총 요청 수
ROWS_MODIFIED	BIGINT	rows_modified - 수정된 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_RETURNED	BIGINT	rows_returned - 리턴된 행 수
TCPIP_RECV_VOLUME	BIGINT	tcPIP_recv_volume - TCP/IP 수신 볼륨
TCPIP_SEND_VOLUME	BIGINT	tcPIP_send_volume - TCP/IP 전송 볼륨
TCPIP_RECV_WAIT_TIME	BIGINT	tcPIP_recv_wait_time - TCP/IP 수신 대기 시간
TCPIP_RECVS_TOTAL	BIGINT	tcPIP_recvs_total - 총 TCP/IP 수신 수
TCPIP_SEND_WAIT_TIME	BIGINT	tcPIP_send_wait_time - TCP/IP 전송 대기 시간
TCPIP_SENDS_TOTAL	BIGINT	tcPIP_sends_total - 총 TCP/IP 전송 수
TOTAL_APP_RQST_TIME	BIGINT	total_app_rqst_time - 총 응용프로그램 요청 시간
TOTAL_RQST_TIME	BIGINT	total_rqst_time - 총 요청 시간
WLM_QUEUE_TIME_TOTAL	BIGINT	wlm_queue_time_total - 워크로드 관리 프로그램 총 큐 시간
WLM_QUEUE_ASSIGNMENTS_TOTAL	BIGINT	wlm_queue_assignments_total - 워크로드 관리 프로그램 총 큐 지정 수
TOTAL_RQST_MAPPED_IN	BIGINT	total_rqst_mapped_in - 맵핑된 전체 요청
TOTAL_RQST_MAPPED_OUT	BIGINT	total_rqst_mapped_out - 맵핑되는 전체 요청
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 총 CPU 시간
TOTAL_WAIT_TIME	BIGINT	total_wait_time - 총 대기 시간
APP_RQSTS_COMPLETED_TOTAL	BIGINT	app_rqsts_completed_total - 완료된 총 응용프로그램 요청 수
TOTAL_SECTION_SORT_TIME	BIGINT	total_section_sort_time - 전체 섹션 정렬 시간



표 103. MON\_GET\_SERVICE\_SUBCLASS에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TOTAL_SECTION_SORT_PROC_TIME	BIGINT	total_section_sort_proc_time - 전체 섹션 정렬 처리 시간
TOTAL_SECTION_SORTS	BIGINT	total_section_sorts - 전체 섹션 정렬 수
TOTAL_SORTS	BIGINT	total_sorts - 전체 정렬 수
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - 포스트 임계값 정렬
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - 포스트 공유 임계값 정렬
SORT_OVERFLOW	BIGINT	sort_overflows - 정렬 오버플로우
TOTAL_COMPILE_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMPILE_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMPILATIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_IMPLICIT_COMPILE_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_IMPLICIT_COMPILE_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_IMPLICIT_COMPILATIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_SECTION_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_SECTION_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_APP_SECTION_EXECUTIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ACT_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ACT_WAIT_TIME	BIGINT	나중에 사용하기 위해 예약됨
ACT_RQSTS_TOTAL	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROUTINE_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROUTINE_INVOCATIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMMIT_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMMIT_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_APP_COMMITS	BIGINT	나중에 사용하기 위해 예약됨
INT_COMMITS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROLLBACK_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROLLBACK_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_APP_ROLLBACKS	BIGINT	나중에 사용하기 위해 예약됨
INT_ROLLBACKS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_RUNSTATS_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_RUNSTATS_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_RUNSTATS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_REORG_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_REORG_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_REORGS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_LOAD_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_LOAD_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_LOADS	BIGINT	나중에 사용하기 위해 예약됨
CAT_CACHE_INSERTS	BIGINT	나중에 사용하기 위해 예약됨

표 103. MON\_GET\_SERVICE\_SUBCLASS에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
CAT_CACHE_LOOKUPS	BIGINT	나중에 사용하기 위해 예약됨
PKG_CACHE_INSERTS	BIGINT	나중에 사용하기 위해 예약됨
PKG_CACHE_LOOKUPS	BIGINT	나중에 사용하기 위해 예약됨
THRESH_VIOLATIONS	BIGINT	나중에 사용하기 위해 예약됨
NUM_LW_THRESH_EXCEEDED	BIGINT	나중에 사용하기 위해 예약됨
ADDITIONAL_DETAILS	BLOB(100K)	나중에 사용하기 위해 예약됨

## MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 테이블 함수 - 서비스 서브클래스 메트릭 세부사항 가져오기

MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 테이블 함수는 하나 이상의 서비스 서브클래스에 대한 메트릭 세부사항을 리턴합니다.

### 구문

```
▶▶—MON_GET_SERVICE_SUBCLASS_DETAILS—(—service_superclass_name—, —————▶
▶—service_subclass_name—, —member—)—————▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *service\_superclass\_name*

이 함수 호출 시 현재 연결된 데이터베이스에서 유효한 서비스 슈퍼 클래스 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수. 인수가 널(null)이거나 빈 문자열이면 데이터베이스의 모든 슈퍼 클래스에 대해 메트릭이 검색됩니다.

#### *service\_subclass\_name*

이 함수 호출 시 현재 연결된 데이터베이스에서 유효한 서비스 서브클래스 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수. 인수가 널(null)이거나 빈 문자열이면 데이터베이스의 모든 서브클래스에 대해 메트릭이 검색됩니다.

#### *member*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. 널(NULL) 값이 지정된 경우, -1이 내재적으로 설정됩니다.



## 권한 부여

MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 함수에서의 EXECUTE 특권.

예 :

각 서비스 슈퍼 클래스에 대해 처리된 총 요청 수 및 총 CPU 사용 시간을 관계형 형식(XMLTABLE을 사용하여)으로 표시합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

SERVICE_SUPERCLASS	TOTAL_CPU	TOTAL_RQSTS
SYSDEFAULTUSERCLASS	2428188	26
SYSDEFAULTMAINTENANCECLASS	0	0
SYSDEFAULTSYSTEMCLASS	0	0

3개의 레코드가 선택되었습니다.

## 사용 시 참고사항

MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 테이블 함수에서 리턴된 메트릭은 표시된 서비스 서브클래스에서 실행된 요청에 대한 모든 메트릭의 누적을 나타냅니다. 이 함수는

MON\_GET\_SERVICE\_SUBCLASS 테이블 함수와 유사합니다.

- MON\_GET\_SERVICE\_SUBCLASS 테이블 함수는 컬럼 기반의 형식으로 가장 일반적으로 사용되는 메트릭을 리턴하며 메트릭을 검색하는 성능이 가장 효율적인 방법입니다.
- MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 테이블 함수는 XML 문서 형식에서 사용 가능한 전체 메트릭 세트를 리턴하며 출력 형식에 대해 최대 유연성을 제공합니다. XML 기반 출력은 XML 구문 분석기에서 직접 구문 분석되거나 XMLTABLE 함수를 사용하여 관계형 형식으로 변환될 수 있습니다(예 참조).

메트릭은 요청 실행 중 주기적으로 작업 단위 경계에 있는 서비스 클래스로 롤업됩니다. 그러므로 이 테이블 함수로 보고된 값은 최근 롤업 시 시스템의 현재 상태를 반영합니다. 메트릭의 값이 정확히 증가됩니다. 시간 간격에 대해 주어진 메트릭 값을 판별

하려면 MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 테이블 함수를 사용하여 간격의 시작과 종료 시 메트릭을 조회하며 차이를 계산합니다.

요청 메트릭은 데이터베이스 레벨에서의 **mon\_req\_metrics** 데이터베이스 구성 매개변수 및 서비스 수퍼 클래스의 COLLECT REQUEST METRICS절을 통해 제어됩니다. 상위 서비스 수퍼 클래스에서 요청 메트릭을 사용할 수 있는 서비스 서브클래스에서 에이전트가 요청을 처리하거나 전체 데이터베이스에 대해 요청 메트릭 컬렉션을 사용할 수 있는 경우에만 요청에 대해 메트릭이 수집됩니다. 디폴트로 데이터베이스 레벨에서 요청 메트릭을 사용할 수 있습니다. 요청 메트릭이 데이터베이스 레벨에서 사용 불가능하고 서비스 수퍼 클래스에 대해 사용 불가능한 경우, 해당 서비스 수퍼 클래스에 맵핑된 각 연결에 보고된 메트릭은 증가를 중지시킵니다. (또는 요청 메트릭이 데이터베이스 활성화 시간에 사용 불가능한 경우 0이 됩니다.)

MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 테이블 함수는 서비스 서브클래스와 구성원별로 한 개의 데이터 행을 리턴합니다. 서비스 클래스(구성원) 또는 구성원(하나 이상의 서비스 클래스)에서는 집계는 수행되지 않습니다. 그러나 SQL 쿼리를 통해 집계는 이루어질 수 있습니다(예 참조). 입력 매개변수는 함께 AND되는 효과가 있습니다. 그러므로 입력 매개변수와의 충돌을 지정하면(SUPA의 서브클래스가 아닌 서브클래스 이름 SUBB와 수퍼 클래스 이름 SUPA) 행이 리턴되지 않습니다.

**팁:** 요청이 둘 이상의 서비스 서브클래스에서 실행될 수 있습니다. 예를 들어 REAMP ACTIVITY 조치로 워크로드 관리 프로그램(WLM) 임계값을 사용하여 요청이 서비스 서브클래스에서 다른 서브클래스로 맵핑되면 이 상황이 발생할 수 있습니다. 요청이 실행되는 각 서비스 서브클래스에 대해 메트릭이 갱신되는 데 시간이 소요되지만 요청이 완료된 서비스 서브클래스에 대해 요청 카운터가 증분됩니다. 그러므로 단일 서브클래스에 대한 평균 요청 시간을 분석하지 마십시오. 활동이 맵핑될 수 있는 모든 서브클래스가 다른 서브클래스와 함께 분석됩니다. 예를 들어 서비스 서브클래스 A에서 서비스 서브클래스 B로 활동을 맵핑할 수 있는 임계값이 존재한다면, 평균 요청을 계산할 때 서비스 서브클래스 A와 B에 대한 카운터와 메트릭을 집계하고 집계를 사용하여 평균을 계산해야 합니다.

DETAILS 컬럼에 리턴된 XML 문서의 스키마는 sqllib/misc/DB2MonRoutines.xsd 파일로 사용 가능합니다. sqllib/misc/DB2MonCommon.xsd 파일에서 더 자세한 내용을 볼 수 있습니다.

## 리턴된 정보

표 104. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	service_superclass_name - 서비스 수퍼 클래스 이름
SERVICE_SUBCLASS_NAME	VARCHAR (128)	service_subclass_name - 서비스 서브클래스 이름

표 104. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
SERVICE_CLASS_ID	INTEGER	service_class_id - 서비스 클래스 ID
MEMBER	SMALLINT	member - 데이터베이스 구성원
DETAILS	BLOB(1M)	서비스 클래스에 대한 메트릭 세부사항을 포함한 XML 문서. 이 문서에서 요소 설명은 표 105의 내용을 참조하십시오.

다음 예는 DETAILS 컬럼에서 리턴되는 XML 문서의 구조를 표시합니다.

```
<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>
```

전체 스키마에 대해서는 `sqllib/misc/DB2MonRoutines.xsd`의 내용을 참조하십시오.

표 105. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS에 대해 리턴되는 메트릭 세부사항

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
service_superclass_name	xs:string(128)	service_superclass_name - 서비스 슈퍼 클래스 이름
service_subclass_name	xs:string(128)	service_subclass_name - 서비스 서브클래스 이름
service_class_id	xs:nonNegativeInteger	service_class_id - 서비스 클래스 ID
member	xs:nonNegativeInteger	member - 데이터베이스 구성원
act_aborted_total	xs:nonNegativeInteger	act_aborted_total - 중단된 총 활동 수
act_completed_total	xs:nonNegativeInteger	act_completed_total - 완료된 총 활동 수
act_rejected_total	xs:nonNegativeInteger	act_rejected_total - 거부된 총 활동 수
agent_wait_time	xs:nonNegativeInteger	agent_wait_time - 에이전트 대기 시간
agent_waits_total	xs:nonNegativeInteger	agent_waits_total - 총 에이전트 대기 수
app_rqsts_completed_total	xs:nonNegativeInteger	app_rqsts_completed_total - 완료된 총 응용프로그램 요청 수
audit_events_total	xs:nonNegativeInteger	audit_events_total - 총 감사 이벤트 수
audit_subsystem_wait_time	xs:nonNegativeInteger	audit_subsystem_wait_time - 감사 서브시스템 대기 시간
audit_subsystem_waits_total	xs:nonNegativeInteger	audit_subsystem_waits_total - 전체 감사 서브시스템 대기
audit_file_write_wait_time	xs:nonNegativeInteger	audit_file_write_wait_time - 감사 파일 작성 대기 시간
audit_file_writes_total	xs:nonNegativeInteger	audit_file_writes_total - 작성된 총 감사 파일 수
client_idle_wait_time	xs:nonNegativeInteger	client_idle_wait_time - 클라이언트 유휴 대기 시간
deadlocks	xs:nonNegativeInteger	deadlocks - 발견된 교착 상태 수
diaglog_writes_total	xs:nonNegativeInteger	diaglog_writes_total - 도표 로그 전체 작성 수
diaglog_write_wait_time	xs:nonNegativeInteger	diaglog_write_wait_time - 도표 로그 작성 시간
direct_read_time	xs:nonNegativeInteger	direct_read_time - 직접 읽기 시간
direct_write_time	xs:nonNegativeInteger	direct_write_time - 직접 쓰기 시간
direct_read_reqs	xs:nonNegativeInteger	direct_read_reqs - 직접 읽기 요청 수
direct_reads	xs:nonNegativeInteger	direct_reads - 데이터베이스에서 직접 읽기 수

표 105. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS에 대해 리턴되는 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
direct_write_reqs	xs:nonNegativeInteger	direct_write_reqs - 직접 쓰기 요청 수
direct_writes	xs:nonNegativeInteger	direct_writes - 데이터베이스에 직접 쓰기 수
fcm_recv_volume	xs:nonNegativeInteger	fcm_recv_volume - FCM 수신 볼륨
fcm_recv_wait_time	xs:nonNegativeInteger	fcm_recv_wait_time - FCM 수신 대기 시간
fcm_recvs_total	xs:nonNegativeInteger	fcm_recvs_total - 총 FCM 수신 수
fcm_message_recv_volume	xs:nonNegativeInteger	fcm_message_recv_volume - FCM 메시지 수신 볼륨
fcm_message_recvs_total	xs:nonNegativeInteger	fcm_message_recvs_total - 총 FCM 메시지 수신 수
fcm_message_recv_wait_time	xs:nonNegativeInteger	fcm_message_recv_wait_time - FCM 메시지 수신 대기 시간
fcm_message_send_volume	xs:nonNegativeInteger	fcm_message_send_volume - FCM 메시지 전송 볼륨
fcm_message_send_wait_time	xs:nonNegativeInteger	fcm_message_send_wait_time - FCM 메시지 전송 대기 시간
fcm_message_sends_total	xs:nonNegativeInteger	fcm_message_sends_total - 총 FCM 메시지 전송 수
fcm_send_volume	xs:nonNegativeInteger	fcm_send_volume - FCM 전송 볼륨
fcm_send_wait_time	xs:nonNegativeInteger	fcm_send_wait_time - FCM 전송 대기 시간
fcm_sends_total	xs:nonNegativeInteger	fcm_sends_total - 총 FCM 전송 수
fcm_tq_recv_wait_time	xs:nonNegativeInteger	fcm_tq_recv_wait_time - FCM 테이블 큐 수신 대기 시간
fcm_tq_send_wait_time	xs:nonNegativeInteger	fcm_tq_send_wait_time - FCM 테이블 큐 전송 대기 시간
fcm_tq_recv_volume	xs:nonNegativeInteger	fcm_tq_recv_volume - FCM 테이블 큐 수신 볼륨
fcm_tq_recvs_total	xs:nonNegativeInteger	fcm_tq_recvs_total - FCM 테이블 큐 수신 총계
fcm_tq_send_volume	xs:nonNegativeInteger	fcm_tq_send_volume - FCM 테이블 큐 전송 볼륨
fcm_tq_sends_total	xs:nonNegativeInteger	fcm_tq_sends_total - FCM 테이블 큐 전송 총계
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총계
ipc_recv_volume	xs:nonNegativeInteger	ipc_recv_volume - 프로세스간 통신 수신 볼륨
ipc_recv_wait_time	xs:nonNegativeInteger	ipc_recv_wait_time - 프로세스간 통신 수신 대기 시간
ipc_recvs_total	xs:nonNegativeInteger	ipc_recvs_total - 총 프로세스간 통신 수신 수
ipc_send_volume	xs:nonNegativeInteger	ipc_send_volume - 프로세스간 통신 전송 볼륨
ipc_send_wait_time	xs:nonNegativeInteger	ipc_send_wait_time - 프로세스간 통신 전송 대기 시간
ipc_sends_total	xs:nonNegativeInteger	ipc_sends_total - 총 프로세스간 통신 전송 수
lock_escals	xs:nonNegativeInteger	lock_escals - 잠금 에스컬레이션 수
lock_timeouts	xs:nonNegativeInteger	lock_timeouts - 잠금 시간종료 수
lock_wait_time	xs:nonNegativeInteger	lock_wait_time - 잠금 대기 시간
lock_waits	xs:nonNegativeInteger	lock_waits - 잠금 대기 수
log_buffer_wait_time	xs:nonNegativeInteger	log_buffer_wait_time - 로그 버퍼 대기 시간
log_disk_wait_time	xs:nonNegativeInteger	log_disk_wait_time - 로그 디스크 대기 시간
log_disk_waits_total	xs:nonNegativeInteger	log_disk_waits_total - 총 로그 디스크 대기 수
pool_data_l_reads	xs:nonNegativeInteger	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
pool_data_p_reads	xs:nonNegativeInteger	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
pool_data_writes	xs:nonNegativeInteger	pool_data_writes - 버퍼 풀 데이터 쓰기 수
pool_index_l_reads	xs:nonNegativeInteger	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
pool_index_p_reads	xs:nonNegativeInteger	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
pool_index_writes	xs:nonNegativeInteger	pool_index_writes - 버퍼 풀 인덱스 쓰기 수

표 105. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS에 대해 리턴되는 매트릭 세부사항 (계속)

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
pool_read_time	xs:nonNegativeInteger	pool_read_time - 총 버퍼 풀 실제 읽기 시간
pool_temp_data_l_reads	xs:nonNegativeInteger	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
pool_temp_data_p_reads	xs:nonNegativeInteger	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
pool_temp_index_l_reads	xs:nonNegativeInteger	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
pool_temp_index_p_reads	xs:nonNegativeInteger	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
pool_temp_xda_l_reads	xs:nonNegativeInteger	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
pool_temp_xda_p_reads	xs:nonNegativeInteger	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수
pool_write_time	xs:nonNegativeInteger	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
pool_xda_l_reads	xs:nonNegativeInteger	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
pool_xda_p_reads	xs:nonNegativeInteger	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
pool_xda_writes	xs:nonNegativeInteger	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
num_log_buffer_full	xs:nonNegativeInteger	num_log_buffer_full - 전체 로그 버퍼 수
rqsts_completed_total	xs:nonNegativeInteger	rqsts_completed_total - 완료된 총 요청 수
total_rqst_mapped_in	xs:nonNegativeInteger	total_rqst_mapped_in - 맵핑된 전체 요청
total_rqst_mapped_out	xs:nonNegativeInteger	total_rqst_mapped_out - 맵핑되는 전체 요청
rows_modified	xs:nonNegativeInteger	rows_modified - 수정된 행 수
rows_read	xs:nonNegativeInteger	rows_read - 읽은 행 수
rows_returned	xs:nonNegativeInteger	rows_returned - 리턴된 행 수
tcpip_rcv_volume	xs:nonNegativeInteger	tcpip_rcv_volume - TCP/IP 수신 볼륨
tcpip_rcv_wait_time	xs:nonNegativeInteger	tcpip_rcv_wait_time - TCP/IP 수신 대기 시간
tcpip_rcvs_total	xs:nonNegativeInteger	tcpip_rcvs_total - 총 TCP/IP 수신 수
tcpip_send_volume	xs:nonNegativeInteger	tcpip_send_volume - TCP/IP 전송 볼륨
tcpip_send_wait_time	xs:nonNegativeInteger	tcpip_send_wait_time - TCP/IP 전송 대기 시간
tcpip_sends_total	xs:nonNegativeInteger	tcpip_sends_total - 총 TCP/IP 전송 수
total_app_rqst_time	xs:nonNegativeInteger	total_app_rqst_time - 총 응용프로그램 요청 시간
total_cpu_time	xs:nonNegativeInteger	total_cpu_time - 총 CPU 시간
total_rqst_time	xs:nonNegativeInteger	total_rqst_time - 총 요청 시간
total_section_sort_time	xs:nonNegativeInteger	total_section_sort_time - 전체 섹션 정렬 시간
total_section_sort_proc_time	xs:nonNegativeInteger	total_section_sort_proc_time - 전체 섹션 정렬 처리 시간
total_section_sorts	xs:nonNegativeInteger	total_section_sorts - 전체 섹션 정렬 수
total_sorts	xs:nonNegativeInteger	total_sorts - 전체 정렬 수
post_threshold_sorts	xs:nonNegativeInteger	post_shrthreshold_sorts - 포스트 공유 임계값 정렬
post_shrthreshold_sorts	xs:nonNegativeInteger	post_shrthreshold_sorts - 포스트 공유 임계값 정렬
sort_overflows	xs:nonNegativeInteger	sort_overflows - 정렬 오버플로우 수
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총계
total_wait_time	xs:nonNegativeInteger	total_wait_time - 총 대기 시간
wlm_queue_time_total	xs:nonNegativeInteger	wlm_queue_time_total - 워크로드 관리 프로그램 총 큐 시간

표 105. MON\_GET\_SERVICE\_SUBCLASS\_DETAILS에 대해 리턴되는 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
wlm_queue_assignments_total	xs:nonNegativeInteger	wlm_queue_assignments_total - 워크로드 관리 프로그램 총 큐 지정 수

## MON\_GET\_TABLE 테이블 함수 - 테이블 메트릭 가져오기

MON\_GET\_TABLE 테이블 함수는 하나 이상의 테이블에 대한 모니터 메트릭을 리턴합니다.

### 구문

▶▶—MON\_GET\_TABLE—(—*tabschema*—, —*tablename*—, —*member*—)▶▶

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *tabschema*

이 함수 호출 시 현재 연결된 데이터베이스에서 유효한 테이블 스키마 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수입니다. 인수가 널(NULL) 또는 비어 있는 문자열인 경우 데이터베이스의 모든 스키마에서 모든 테이블에 대해 메트릭이 검색됩니다. 인수가 지정된 경우 메트릭은 지정된 스키마의 테이블에 대해서만 리턴됩니다.

#### *tablename*

이 함수 호출 시 현재 연결된 데이터베이스에서 유효한 테이블 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수입니다. 인수가 널(NULL) 또는 비어 있는 문자열인 경우 데이터베이스의 모든 테이블에 대해 메트릭이 검색됩니다.

#### *member*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. NULL 값이 지정되면 -1이 내재적으로 설정됩니다.

### 권한 부여

MON\_GET\_TABLE 테이블 함수의 EXECUTE 특권

### 예 :

데이터베이스가 활성화된 이후에 모든 데이터베이스 구성원에서 집계된 액세스한 모든 테이블의 활동을 가장 많이 읽은 순서대로 표시합니다.





```

...           0           0           0
...           0           0           0
...           0           0           0

```

## 사용 시 참고사항

MON\_GET\_TABLE 테이블 함수는 데이터베이스 테이블 및 데이터베이스 구성원별로 한 개의 데이터 행을 리턴합니다. 범위 파티션 테이블을 사용 중인 경우 데이터베이스 구성원별로 각 테이블 파티션에 대해 한 행이 리턴됩니다. 데이터베이스 구성원간의 집계는 수행되지 않습니다. 그렇지만 예에서 표시된 대로 SQL 쿼리를 사용하여 집계를 수행할 수 있습니다.

데이터베이스가 활성화된 이후에 액세스한 테이블에 대해서만 메트릭이 리턴됩니다.

메트릭은 항상 사용할 수 있습니다. 이 함수를 통해 테이블 메트릭에 액세스하기 위해 시스템 모니터 스위치를 켜 필요 없습니다.

## 리턴된 정보

표 106. MON\_GET\_TABLE에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
TABSCHEMA	VARCHAR (128)	table_schema - 테이블 스키마 이름
TABNAME	VARCHAR (128)	table_name - 테이블 이름
MEMBER	SMALLINT	member - 데이터베이스 구성원
TAB_TYPE	VARCHAR(14)	table_type - 테이블 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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>
TAB_FILE_ID	BIGINT	table_file_id - 테이블 파일 ID
DATA_PARTITION_ID	INTEGER	data_partition_id - 데이터 파티션 ID
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
INDEX_TBSP_ID	BIGINT	index_tbsp_id - 인덱스 테이블 스페이스 ID
LONG_TBSP_ID	BIGINT	long_tbsp_id - 긴 테이블 스페이스 ID
TABLE_SCANS	BIGINT	table_scans - 테이블 스캔
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_INSERTED	BIGINT	rows_inserted - 삽입된 행 수
ROWS_UPDATED	BIGINT	rows_updated - 갱신된 행 수
ROWS_DELETED	BIGINT	rows_deleted - 삭제된 행 수
OVERFLOW_ACCESSES	BIGINT	overflow_accesses - 오버플로우된 레코드에 액세스



표 106. MON\_GET\_TABLE에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
OVERFLOW_CREATES	BIGINT	overflow_creates - 오버플로우 작성
PAGE_REORGS	BIGINT	향후에 사용하기 위해 예약되어 있습니다.
ADDITIONAL_DETAILS	BLOB(100K)	향후에 사용하기 위해 예약되어 있습니다.

## MON\_GET\_TABLESPACE 테이블 함수 - 테이블 스페이스 테이블 메트릭 가져오기

MON\_GET\_TABLESPACE 테이블 함수는 하나 이상의 테이블 스페이스에 대한 모니터 메트릭을 리턴합니다.

### 구문

▶▶ MON\_GET\_TABLESPACE (—*tblsp\_name*—, —*member*—) ▶▶

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *tblsp\_name*

이 함수를 호출할 때 현재 연결된 데이터베이스에서 유효한 데이터베이스 스페이스 이름을 지정하는 VARCHAR (128) 유형의 입력 인수. 인수가 null이거나 비어 있는 문자열인 경우, 메트릭이 데이터베이스의 테이블 스페이스에 대해 검색됩니다.

#### *member*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. NULL 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

### 권한 부여

MON\_GET\_TABLESPACE 함수에 대한 EXECUTE 특권.

### 예 :

테이블 스페이스 목록은 테이블 스페이스 컨테이너의 실제 읽기 번호 순으로 정렬됩니다.

```

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

```

이 쿼리의 출력 예는 다음과 같습니다.

TBSP_NAME	MEMBER	TBSP_TYPE	POOL_DATA_P_READS
SYSCATSPACE		0 DMS	79
USERSPACE1		0 DMS	34
TEMPSPACE1		0 SMS	0

3개의 레코드가 선택되었습니다.

## 사용 시 참고사항

MON\_GET\_TABLESPACE 테이블 함수는 데이터베이스 테이블 및 데이터베이스 구성원별로 하나의 데이터 행을 리턴합니다. 데이터베이스 구성원간의 집계는 수행되지 않습니다. 그러나 SQL 쿼리를 통해 집계가 달성될 수 있습니다.

이 함수에 의해 수집된 매트릭스는 mon\_obj\_metrics 구성 매개변수를 사용하여 데이터베이스 레벨에서 제어됩니다. 디폴트로 매트릭 컬렉션이 사용 가능합니다.

## 리턴된 정보

표 107. MON\_GET\_TABLESPACE로 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TBSP_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
MEMBER	SMALLINT	member - 데이터베이스 구성원
TBSP_TYPE	VARCHAR(10)	tablespace_type - 테이블 스페이스 유형. 이 인터페이스는 sqlutil.h로 정의하는 것을 기초로 하는 텍스트 ID를 리턴하며, 이는 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• DMS</li> <li>• SMS</li> </ul>
TBSP_CONTENT_TYPE	VARCHAR(10)	tablespace_content_type - 테이블 스페이스 콘텐츠 유형. 이 인터페이스는 sqlmon.h로 정의한 것을 기초로 하는 텍스트 ID를 리턴하며, 이는 다음 중 하나입니다. <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 - 테이블 스페이스 페이지 크기
TBSP_EXTENT_SIZE	BIGINT	tablespace_extent_size - 테이블 스페이스 Extent 크기
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 테이블 스페이스 프리페치 크기

표 107. MON\_GET\_TABLESPACE로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TBSP_CUR_POOL_ID	BIGINT	tablespace_cur_pool_id - 현재 사용 중인 버퍼 풀
TBSP_NEXT_POOL_ID	BIGINT	tablespace_next_pool_id - 다음 시작 시 사용될 버퍼 풀
FS_CACHING	SMALLINT	fs_caching - 파일 시스템 캐싱
TBSP_REBALANCER_MODE	VARCHAR (30)	tablespace_rebalancer_mode - 재조정 프로그램 모드. 이 인터페이스는 sqlmon.h로 정의한 것을 기초로 하는 텍스트 ID를 리턴하며, 이는 다음 중 하나입니다. <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 - 테이블 공간을 자동 스토리지에 사용 가능
TBSP_AUTO_RESIZE_ENABLED	SMALLINT	tablespace_auto_resize_enabled - 테이블 공간 자동 크기 조정 가능
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간

표 107. MON\_GET\_TABLESPACE로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 수
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 수
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 수
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수
VECTORED_IOS	BIGINT	vectored_ios - 벡터 입출력 요청 수
PAGES_FROM_VECTORED_IOS	BIGINT	pages_from_vectored_ios - 벡터 입출력으로 읽은 총 페이지 수
BLOCK_IOS	BIGINT	block_ios - 블록 입출력 요청 수
PAGES_FROM_BLOCK_IOS	BIGINT	pages_from_block_ios - 블록 입출력으로 읽은 총 페이지 수
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리페치 페이지 수
FILES_CLOSED	BIGINT	files_closed - 닫힌 데이터베이스 파일 수
TBSP_STATE	VARCHAR(256)	tablespace_state - 테이블 스페이스 상태
TBSP_USED_PAGES	BIGINT	tablespace_used_pages - 테이블 스페이스에서 사용된 페이지
TBSP_FREE_PAGES	BIGINT	tablespace_free_pages - 테이블 스페이스에서 사용 가능한 페이지
TBSP_USABLE_PAGES	BIGINT	tablespace_usable_pages - 테이블 스페이스에서 사용할 수 있는 페이지
TBSP_TOTAL_PAGES	BIGINT	tablespace_total_pages - 테이블 스페이스의 전체 페이지
TBSP_PENDING_FREE_PAGES	BIGINT	tablespace_pending_free_pages - 테이블 스페이스에서 보류 중인 사용 가능한 페이지
TBSP_PAGE_TOP	BIGINT	tablespace_page_top - 테이블 스페이스 상위 워터 마크(water mark)
TBSP_MAX_PAGE_TOP	BIGINT	tablespace_max_size - 최대 테이블 스페이스 크기
RECLAIMABLE_SPACE_ENABLED	SMALLINT	reclaimable_space_enabled - 재개 가능한 스페이스 사용 표시기
AUTO_STORAGE_HYBRID	SMALLINT	auto_storage_hybrid - 하이브리드 자동 스토리지 테이블 스페이스 표시기
TBSP_PATHS_DROPPED	SMALLINT	tablespace_paths_dropped - 삭제된 경로를 사용하는 테이블 스페이스



```

APPLICATION_HANDLE  UOW_ID  TOTAL_CPU_TIME  ...
-----
                        46      5      27959  ...

```

1개의 레코드가 선택되었습니다.

쿼리의 출력(계속).

```

... APP_RQSTS_COMPLETED_TOTAL  RQSTS_COMPLETED_TOTAL
... -----
...                          72      48

```

## 사용 시 참고사항

MON\_GET\_UNIT\_OF\_WORK 테이블 함수가 리턴하는 메트릭은 작업 단위(UOW) 중에 제출된 요청에 대한 모든 메트릭의 누적을 나타냅니다. 메트릭은 작업 단위(UOW) 중에 주기적으로 롤업됩니다. 따라서 이 테이블 함수가 보고한 값은 최신 롤업 시의 시스템 현재 상태를 반영합니다. 메트릭의 값은 반드시 증가합니다. 시간 간격에 대해 제공된 메트릭 값을 판별하려면, 함수를 사용하여 간격의 시작 및 종료 시점의 메트릭을 쿼리하여 차이를 계산하십시오.

요청 메트릭은 데이터베이스 레벨에서의 *mon\_req\_metrics* 데이터베이스 구성 매개변수 및 서비스 수퍼 클래스의 COLLECT REQUEST METRICS 절을 통해 제어됩니다. 상위 서비스 수퍼 클래스에서 요청 메트릭을 사용할 수 있는 서비스 서브클래스에서 에이전트가 요청을 처리하거나 전체 데이터베이스에 대해 요청 메트릭 콜렉션을 사용할 수 있는 경우에만 요청에 대해 메트릭이 수집됩니다. 디폴트로 데이터베이스 레벨에서 요청 메트릭을 사용할 수 있습니다. 요청 메트릭이 데이터베이스 레벨에서 사용 불가능하고 서비스 수퍼 클래스에 대해 사용 불가능한 경우, 해당 서비스 수퍼 클래스에 맵핑된 각 작업 단위(UOW)에 보고된 메트릭은 증가를 중지시킵니다. (또는 요청 메트릭이 데이터베이스 활성화 시간에 사용 불가능한 경우 0이 됩니다.)

MON\_GET\_UNIT\_OF\_WORK 테이블 함수는 작업 단위(UOW) 및 구성원별로 하나의 데이터 행을 리턴합니다. 작업 단위(UOW)(구성원에서) 또는 구성원(하나 이상의 서비스 클래스)에서는 집계는 수행되지 않습니다. 그렇지만 SQL 쿼리를 사용하여 집계를 수행할 수 있습니다. 입력 매개변수는 모두 AND가 됩니다. 따라서 충돌하는 입력 매개변수를 지정하면(예: 수퍼 클래스 이름 SUPA와 수퍼 클래스 SUPA의 서브클래스가 아닌 서브클래스 이름 SUBB) 행이 리턴되지 않습니다.

## 리턴된 정보

표 108. MON\_GET\_UNIT\_OF\_WORK에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	service_superclass_name - 서비스 수퍼 클래스 이름
SERVICE_SUBCLASS_NAME	VARCHAR (128)	service_subclass_name - 서비스 서브클래스 이름

표 108. MON\_GET\_UNIT\_OF\_WORK에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SERVICE_CLASS_ID	INTEGER	service_class_id - 서비스 클래스 ID
MEMBER	SMALLINT	member - 데이터베이스 구성원
COORD_MEMBER	SMALLINT	coord_member - 코디네이터 구성원
APPLICATION_HANDLE	BIGINT	application_handle - 응용프로그램 핸들
APPLICATION_ID	VARCHAR (128)	나중에 사용하기 위해 예약됨
WORKLOAD_NAME	VARCHAR (128)	workload_name - 워크로드 이름
WORKLOAD_OCCURRENCE_ID	INTEGER	workload_occurrence_id - 워크로드 어커런스 ID. 이 ID는 코디네이터 구성원 및 워크로드 이름과 결합하지 않으면 워크로드 어커런스를 고유하게 식별하지 않습니다.
UOW_ID	INTEGER	uow_id - 작업 단위(UOW) ID
WORKLOAD_OCCURRENCE_STATE	VARCHAR(32)	workload_occurrence_state - 워크로드 어커런스 상태
CLIENT_WRKSTNNAME	VARCHAR(255)	CURRENT CLIENT_WRKSTNNAME 특수 레지스터
CLIENT_ACCTNG	VARCHAR(255)	CURRENT CLIENT_ACCTNG 특수 레지스터
CLIENT_USERID	VARCHAR(255)	CURRENT CLIENT_USERID 특수 레지스터
CLIENT_APPLNAME	VARCHAR(255)	CURRENT CLIENT_APPLNAME 특수 레지스터
UOW_START_TIME	TIMESTAMP	나중에 사용하기 위해 예약됨
SESSION_AUTH_ID	VARCHAR (128)	나중에 사용하기 위해 예약됨
ACT_ABORTED_TOTAL	BIGINT	act_aborted_total - 중단된 총 활동 수
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 완료된 총 활동 수
ACT_REJECTED_TOTAL	BIGINT	act_rejected_total - 거부된 총 활동 수
AGENT_WAIT_TIME	BIGINT	agent_wait_time - 에이전트 대기 시간
AGENT_WAITS_TOTAL	BIGINT	agent_waits_total - 총 에이전트 대기 수
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수



표 108. MON\_GET\_UNIT\_OF\_WORK에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
CLIENT_IDLE_WAIT_TIME	BIGINT	client_idle_wait_time - 클라이언트 유휴 대기 시간
DEADLOCKS	BIGINT	deadlocks - 발견된 교착 상태 수
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
FCM_RECV_VOLUME	BIGINT	fcm_recv_volume - FCM 수신 볼륨
FCM_RECVS_TOTAL	BIGINT	fcm_recvs_total - 총 FCM 수신 수
FCM_SEND_VOLUME	BIGINT	fcm_send_volume - FCM 전송 볼륨
FCM_SENDS_TOTAL	BIGINT	fcm_sends_total - 총 FCM 전송 수
FCM_RECV_WAIT_TIME	BIGINT	fcm_recv_wait_time - FCM 수신 대기 시간
FCM_SEND_WAIT_TIME	BIGINT	fcm_send_wait_time - FCM 전송 대기 시간
IPC_RECV_VOLUME	BIGINT	ipc_recv_volume - 프로세스간 통신 수신 볼륨
IPC_RECV_WAIT_TIME	BIGINT	ipc_recv_wait_time - 프로세스간 통신 수신 대기 시간
IPC_RECVS_TOTAL	BIGINT	ipc_recvs_total - 총 프로세스간 통신 수신 수
IPC_SEND_VOLUME	BIGINT	ipc_send_volume - 프로세스간 통신 전송 볼륨
IPC_SEND_WAIT_TIME	BIGINT	ipc_send_wait_time - 프로세스간 통신 전송 대기 시간
IPC_SENDS_TOTAL	BIGINT	ipc_sends_total - 총 프로세스간 통신 전송 수
LOCK_ESCALS	BIGINT	lock_escalations - 잠금 에스컬레이션 수
LOCK_TIMEOUTS	BIGINT	lock_timeouts - 잠금 시간종료 수
LOCK_WAIT_TIME	BIGINT	lock_wait_time - 잠금 대기 시간
LOCK_WAITS	BIGINT	lock_waits - 잠금 대기 수
LOG_BUFFER_WAIT_TIME	BIGINT	log_buffer_wait_time - 로그 버퍼 대기 시간
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - 전체 로그 버퍼 수
LOG_DISK_WAIT_TIME	BIGINT	log_disk_wait_time - 로그 디스크 대기 시간
LOG_DISK_WAITS_TOTAL	BIGINT	log_disk_waits_total - 총 로그 디스크 대기 수
NUM_LOCKS_HELD	BIGINT	나중에 사용하기 위해 예약됨
RQSTS_COMPLETED_TOTAL	BIGINT	rqsts_completed_total - 완료된 총 요청 수
ROWS_MODIFIED	BIGINT	rows_modified - 수정된 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_RETURNED	BIGINT	rows_returned - 리턴된 행 수
TCPIP_RECV_VOLUME	BIGINT	tcpip_recv_volume - TCP/IP 수신 볼륨



표 108. MON\_GET\_UNIT\_OF\_WORK에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TCPIP_SEND_VOLUME	BIGINT	tcPIP_send_volume - TCP/IP 전송 볼륨
TCPIP_RECV_WAIT_TIME	BIGINT	tcPIP_recv_wait_time - TCP/IP 수신 대기 시간
TCPIP_RECVS_TOTAL	BIGINT	tcPIP_recvs_total - 총 TCP/IP 수신 수
TCPIP_SEND_WAIT_TIME	BIGINT	tcPIP_send_wait_time - TCP/IP 전송 대기 시간
TCPIP_SENDS_TOTAL	BIGINT	tcPIP_sends_total - 총 TCP/IP 전송 수
TOTAL_APP_RQST_TIME	BIGINT	total_app_rqst_time - 총 응용프로그램 요청 시간
TOTAL_RQST_TIME	BIGINT	total_rqst_time - 총 요청 시간
WLM_QUEUE_TIME_TOTAL	BIGINT	wlm_queue_time_total - 워크로드 관리 프로그램 총 큐 시간
WLM_QUEUE_ASSIGNMENTS_TOTAL	BIGINT	wlm_queue_assignments_total - 워크로드 관리 프로그램 총 큐 지정 수
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 총 CPU 시간
TOTAL_WAIT_TIME	BIGINT	total_wait_time - 총 대기 시간
APP_RQSTS_COMPLETED_TOTAL	BIGINT	app_rqsts_completed_total - 완료된 총 응용프로그램 요청 수
TOTAL_SECTION_SORT_TIME	BIGINT	total_section_sort_time - 전체 섹션 정렬 시간
TOTAL_SECTION_SORT_PROC_TIME	BIGINT	total_section_sort_proc_time - 전체 섹션 정렬 처리 시간
TOTAL_SECTION_SORTS	BIGINT	total_section_sorts - 전체 섹션 정렬 수
TOTAL_SORTS	BIGINT	total_sorts - 전체 정렬 수
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - 포스트 임계값 정렬
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - 포스트 공유 임계값 정렬
SORT_OVERFLOW	BIGINT	sort_overflows - 정렬 오버플로우
TOTAL_COMPILE_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMPILE_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMPILATIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_IMPLICIT_COMPILE_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_IMPLICIT_COMPILE_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_IMPLICIT_COMPILATIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_SECTION_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_SECTION_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_APP_SECTION_EXECUTIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ACT_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ACT_WAIT_TIME	BIGINT	나중에 사용하기 위해 예약됨
ACT_RQSTS_TOTAL	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROUTINE_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROUTINE_INVOCATIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMMIT_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMMIT_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_APP_COMMITS	BIGINT	나중에 사용하기 위해 예약됨

표 108. MON\_GET\_UNIT\_OF\_WORK에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
INT_COMMITS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROLLBACK_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROLLBACK_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_APP_ROLLBACKS	BIGINT	나중에 사용하기 위해 예약됨
INT_ROLLBACKS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_RUNSTATS_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_RUNSTATS_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_RUNSTATS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_REORG_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_REORG_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_REORGS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_LOAD_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_LOAD_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_LOADS	BIGINT	나중에 사용하기 위해 예약됨
CAT_CACHE_INSERTS	BIGINT	나중에 사용하기 위해 예약됨
CAT_CACHE_LOOKUPS	BIGINT	나중에 사용하기 위해 예약됨
PKG_CACHE_INSERTS	BIGINT	나중에 사용하기 위해 예약됨
PKG_CACHE_LOOKUPS	BIGINT	나중에 사용하기 위해 예약됨
THRESH_VIOLATIONS	BIGINT	나중에 사용하기 위해 예약됨
NUM_LW_THRESH_EXCEEDED	BIGINT	나중에 사용하기 위해 예약됨
UOW_LOG_SPACE_USED	BIGINT	나중에 사용하기 위해 예약됨
ADDITIONAL_DETAILS	BLOB(100K)	나중에 사용하기 위해 예약됨

## MON\_GET\_UNIT\_OF\_WORK\_DETAILS 테이블 함수 - 작업 단위(UOW) 메트릭 세부사항 가져오기

MON\_GET\_UNIT\_OF\_WORK\_DETAILS 테이블 함수는 하나 이상의 작업단위에 대한 메트릭 세부사항을 리턴합니다.

### 구문

▶▶—MON\_GET\_UNIT\_OF\_WORK\_DETAILS—(—*application\_handle*—,—*member*—)————▶▶

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

*application\_handle*

이 함수 호출 시 현재 연결된 것과 동일한 데이터베이스에서 유효한 응용프로그램

핸들을 지정하는 BIGINT 유형의 입력 인수. 인수가 널(NULL)이면, 데이터베이스의 모든 수퍼 클래스에서 실행 중인 작업 단위(UOW)에 대해 메트릭이 검색됩니다.

*member*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

**권한 부여**

MON\_GET\_UNIT\_OF\_WORK\_DETAILS 함수에서의 EXECUTE 특권.

**예 :**

시스템에서 가장 높은 CPU 시간을 사용하는 작업 단위를 식별합니다.

```
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"
COLUMNS  "APPLICATION_HANDLE" INTEGER PATH 'application_handle',
         "UOW_ID" INTEGER PATH 'uow_id',
         "TOTAL_CPU_TIME" INTEGER PATH 'system_metrics/total_cpu_time',
         "APP_RQSTS_COMPLETED_TOTAL" INTEGER PATH 'system_metrics/app_rqsts_completed_total',
         "RQSTS_COMPLETED_TOTAL" INTEGER PATH 'system_metrics/rqsts_completed_total'
) AS DETMETRICS
ORDER BY total_cpu_time DESC
```

이 쿼리의 출력 예는 다음과 같습니다.

APPLICATION_HANDLE	UOW_ID	TOTAL_CPU_TIME	...
-----	-----	-----	...
	46	5	27959 ...

1개의 레코드가 선택되었습니다.

쿼리의 출력(계속).

...	APP_RQSTS_COMPLETED_TOTAL	RQSTS_COMPLETED_TOTAL
...	-----	-----
...	72	48

**사용 시 참고사항**

MON\_GET\_UNIT\_OF\_WORK\_DETAILS 함수에서 리턴된 메트릭은 작업 단위 중 제출된 요청에 대한 모든 메트릭의 누적을 나타냅니다. 이 함수는 MON\_GET\_UNIT\_OF\_WORK 테이블 함수와 유사합니다.

- MON\_GET\_UNIT\_OF\_WORK 테이블 함수는 컬럼 기반의 형식으로 가장 일반적으로 사용되는 메트릭을 리턴하며 메트릭을 검색하는 성능이 가장 효율적인 방법입니다.
- MON\_GET\_UNIT\_OF\_WORK\_DETAILS 테이블 함수는 XML 문서 형식에서 사용 가능한 전체 메트릭 세트를 리턴하며 출력 형식에 대해 최대 유연성을 제공합니다. XML 기반 출력은 XML 구문 분석기에서 직접 구문 분석되거나 XMLTABLE 함수를 사용하여 관계형 형식으로 변환될 수 있습니다(예 참조).

메트릭은 작업 단위(UOW) 중에 주기적으로 롤업됩니다. 그러므로 이 테이블 함수로 보고된 값은 최근 롤업 시 시스템의 현재 상태를 반영합니다. 메트릭의 값이 정확히 증가됩니다. 시간 간격에 대해 주어진 메트릭 값을 판별하려면 MON\_GET\_UNIT\_OF\_WORK\_DETAILS 테이블 함수를 사용하여 간격의 시작과 종료 시 메트릭을 조회하며 차이를 계산합니다.

요청 메트릭은 데이터베이스 레벨에서의 **mon\_req\_metrics** 데이터베이스 구성 매개변수 및 서비스 수퍼 클래스의 COLLECT REQUEST METRICS절을 통해 제어됩니다. 상위 서비스 수퍼 클래스에서 요청 메트릭을 사용할 수 있는 서비스 서브클래스에서 에이전트가 요청을 처리하거나 전체 데이터베이스에 대해 요청 메트릭 콜렉션을 사용할 수 있는 경우에만 요청에 대해 메트릭이 수집됩니다. 디폴트로 데이터베이스 레벨에서 요청 메트릭을 사용할 수 있습니다. 요청 메트릭이 데이터베이스 레벨에서 사용 불가능하고 서비스 수퍼 클래스에 대해 사용 불가능한 경우, 해당 서비스 수퍼 클래스에 맵핑된 각 작업 단위(UOW)에 보고된 메트릭은 증가를 중지시킵니다. (또는 요청 메트릭이 데이터베이스 활성화 시간에 사용 불가능한 경우 0이 됩니다.)

MON\_GET\_UNIT\_OF\_WORK\_DETAILS 테이블 함수는 작업 단위(UOW) 및 구성 원별로 한 행의 데이터를 리턴합니다. 작업 단위(UOW)(구성원에서) 또는 구성원(하나 이상의 서비스 클래스)에서는 집계는 수행되지 않습니다. 그러나 SQL 쿼리를 통해 집계는 이루어질 수 있습니다. 입력 매개변수는 함께 AND되는 효과가 있습니다. 그러므로 입력 매개변수와 충돌을 지정하면(SUPA의 서브클래스가 아닌 서브클래스 이름 SUBB와 수퍼 클래스 이름 SUPA) 행이 리턴되지 않습니다.

DETAILS 컬럼에 리턴된 XML 문서의 스키마는 sqllib/misc/DB2MonRoutines.xsd 파일로 사용 가능합니다. sqllib/misc/DB2MonCommon.xsd 파일에서 더 자세한 내용을 볼 수 있습니다.

## 리턴된 정보

표 109. MON\_GET\_UNIT\_OF\_WORK\_DETAILS에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	service_superclass_name - 서비스 수퍼 클래스 이름

표 109. MON\_GET\_UNIT\_OF\_WORK\_DETAILS에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SERVICE_SUBCLASS_NAME	VARCHAR (128)	service_subclass_name - 서비스 서브클래스 이름
SERVICE_CLASS_ID	INTEGER	service_class_id - 서비스 클래스 ID
MEMBER	SMALLINT	member - 데이터베이스 구성원
COORD_MEMBER	SMALLINT	coord_partition_num - 코디네이터 구성원. 지정된 작업 단위(UOW)의 코디네이터 파티션에 대한 데이터베이스 구성원.
APPLICATION_HANDLE	BIGINT	application_handle - 응용프로그램 핸들
WORKLOAD_NAME	VARCHAR (128)	workload_name - 워크로드 이름
WORKLOAD_OCCURRENCE_ID	INTEGER	workload_occurrence_id - 워크로드 어커런스 ID. 코디네이터 데이터베이스 파티션 번호와 워크로드 이름으로 연결되어 있지 않으면 이 ID는 워크로드 어커런스를 고유하게 식별하지 않습니다.
UOW_ID	INTEGER	uow_id - 작업 단위(UOW) ID
DETAILS	BLOB(1M)	작업 단위로 메트릭 세부사항을 포함한 XML 문서. 이 문서에서 요소 설명은 표 110의 내용을 참조하십시오.

다음 예는 DETAILS 컬럼에서 리턴되는 XML 문서의 구조를 표시합니다.

```
<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>
```

전체 스키마에 대해서는 sqllib/misc/DB2MonRoutines.xsd의 내용을 참조하십시오.

표 110. MON\_GET\_UNIT\_OF\_WORK\_DETAILS에 대해 리턴된 메트릭 세부사항

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
service_superclass_name	xs:string(128)	service_superclass_name - 서비스 슈퍼 클래스 이름
service_subclass_name	xs:string(128)	service_subclass_name - 서비스 서브클래스 이름
service_class_id	xs:nonNegativeInteger	service_class_id - 서비스 클래스 ID
workload_name	xs:string(128)	workload_name - 워크로드 이름
member	xs:nonNegativeInteger	member - 데이터베이스 구성원
coord_member	xs:nonNegativeInteger	coord_member - 코디네이터 구성원
application_handle	xs:nonNegativeInteger	application_handle - 응용프로그램 핸들

표 110. MON\_GET\_UNIT\_OF\_WORK\_DETAILS에 대해 리턴된 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
workload_occurrence_id	xs:nonNegativeInteger	workload_occurrence_id - 워크로드 어커런스 ID. 이 ID는 코디네이터 구성원 및 워크로드 이름과 결합된 경우에만 워크로드 어커런스를 고유하게 식별합니다.
uow_id	xs:nonNegativeInteger	uow_id - 작업 단위(UOW) ID
workload_occurrence_state	xs:string	workload_occurrence_state - 워크로드 어커런스 상태
client_userid	xs:string	CURRENT_CLIENT_USERID 특수 레지스터
client_wrkstnname	xs:string	CURRENT_CLIENT_WRKSTNNAME 특수 레지스터
client_applname	xs:string	CURRENT_CLIENT_APPLNAME 특수 레지스터
client_acctng	xs:string	CURRENT_CLIENT_ACCTNG 특수 레지스터
act_aborted_total	xs:nonNegativeInteger	act_aborted_total - 중단된 총 활동 수
act_completed_total	xs:nonNegativeInteger	act_completed_total - 완료된 총 활동 수
act_rejected_total	xs:nonNegativeInteger	act_rejected_total - 거부된 총 활동 수
agent_wait_time	xs:nonNegativeInteger	agent_wait_time - 에이전트 대기 시간
agent_waits_total	xs:nonNegativeInteger	agent_waits_total - 총 에이전트 대기 수
app_rqsts_completed_total	xs:nonNegativeInteger	app_rqsts_completed_total - 완료된 총 응용프로그램 요청 수
audit_events_total	xs:nonNegativeInteger	audit_events_total - 총 감사 이벤트 수
audit_subsystem_wait_time	xs:nonNegativeInteger	audit_subsystem_wait_time - 감사 서브시스템 대기 시간
audit_subsystem_waits_total	xs:nonNegativeInteger	audit_subsystem_waits_total - 전체 감사 서브시스템 대기
audit_file_write_wait_time	xs:nonNegativeInteger	audit_file_write_wait_time - 감사 파일 작성 대기 시간
audit_file_writes_total	xs:nonNegativeInteger	audit_file_writes_total - 작성된 총 감사 파일 수
client_idle_wait_time	xs:nonNegativeInteger	client_idle_wait_time - 클라이언트 유휴 대기 시간
교착 상태	xs:nonNegativeInteger	deadlocks - 발견된 교착 상태 수
diaglog_writes_total	xs:nonNegativeInteger	diaglog_writes_total - 도표 로그 전체 작성 수
diaglog_write_wait_time	xs:nonNegativeInteger	diaglog_write_wait_time - 도표 로그 작성 시간
direct_read_time	xs:nonNegativeInteger	direct_read_time - 직접 읽기 시간
direct_write_time	xs:nonNegativeInteger	direct_write_time - 직접 쓰기 시간
direct_read_reqs	xs:nonNegativeInteger	direct_read_reqs - 직접 읽기 요청 수
direct_reads	xs:nonNegativeInteger	direct_reads - 데이터베이스에서 직접 읽기 수
direct_write_reqs	xs:nonNegativeInteger	direct_write_reqs - 직접 쓰기 요청 수
direct_writes	xs:nonNegativeInteger	direct_writes - 데이터베이스에 직접 쓰기 수
fcm_recv_volume	xs:nonNegativeInteger	fcm_recv_volume - FCM 수신 볼륨
fcm_recv_wait_time	xs:nonNegativeInteger	fcm_recv_wait_time - FCM 수신 대기 시간
fcm_recvs_total	xs:nonNegativeInteger	fcm_recvs_total - 총 FCM 수신 수
fcm_message_recv_volume	xs:nonNegativeInteger	fcm_message_recv_volume - FCM 메시지 수신 볼륨
fcm_message_recvs_total	xs:nonNegativeInteger	fcm_message_recvs_total - 총 FCM 메시지 수신 수
fcm_message_recv_wait_time	xs:nonNegativeInteger	fcm_message_recv_wait_time - FCM 메시지 수신 대기 시간
fcm_message_send_volume	xs:nonNegativeInteger	fcm_message_send_volume - FCM 메시지 전송 볼륨
fcm_message_send_wait_time	xs:nonNegativeInteger	fcm_message_send_wait_time - FCM 메시지 전송 대기 시간

표 110. MON\_GET\_UNIT\_OF\_WORK\_DETAILS에 대해 리턴된 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
fcm_message_sends_total	xs:nonNegativeInteger	fcm_message_sends_total - 총 FCM 메시지 전송 수
fcm_send_volume	xs:nonNegativeInteger	fcm_send_volume - FCM 전송 볼륨
fcm_send_wait_time	xs:nonNegativeInteger	fcm_send_wait_time - FCM 전송 대기 시간
fcm_sends_total	xs:nonNegativeInteger	fcm_sends_total - 총 FCM 전송 수
fcm_tq_recv_wait_time	xs:nonNegativeInteger	fcm_tq_recv_wait_time - FCM 테이블 큐 수신 대기 시간
fcm_tq_send_wait_time	xs:nonNegativeInteger	fcm_tq_send_wait_time - FCM 테이블 큐 전송 대기 시간
fcm_tq_recv_volume	xs:nonNegativeInteger	fcm_tq_recv_volume - FCM 테이블 큐 수신 볼륨
fcm_tq_recvs_total	xs:nonNegativeInteger	fcm_tq_recvs_total - FCM 테이블 큐 수신 총계
fcm_tq_send_volume	xs:nonNegativeInteger	fcm_tq_send_volume - FCM 테이블 큐 전송 볼륨
fcm_tq_sends_total	xs:nonNegativeInteger	fcm_tq_sends_total - FCM 테이블 큐 전송 총계
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총계
ipc_recv_volume	xs:nonNegativeInteger	ipc_recv_volume - 프로세스간 통신 수신 볼륨
ipc_recv_wait_time	xs:nonNegativeInteger	ipc_recv_wait_time - 프로세스간 통신 수신 대기 시간
ipc_recvs_total	xs:nonNegativeInteger	ipc_recvs_total - 총 프로세스간 통신 수신 수
ipc_send_volume	xs:nonNegativeInteger	ipc_send_volume - 프로세스간 통신 전송 볼륨
ipc_send_wait_time	xs:nonNegativeInteger	ipc_send_wait_time - 프로세스간 통신 전송 대기 시간
ipc_sends_total	xs:nonNegativeInteger	ipc_sends_total - 총 프로세스간 통신 전송 수
lock_escals	xs:nonNegativeInteger	lock_escals - 잠금 에스컬레이션 수
lock_timeouts	xs:nonNegativeInteger	lock_timeouts - 잠금 시간종료 수
lock_wait_time	xs:nonNegativeInteger	lock_wait_time - 잠금 대기 시간
lock_waits	xs:nonNegativeInteger	lock_waits - 잠금 대기 수
log_buffer_wait_time	xs:nonNegativeInteger	log_buffer_wait_time - 로그 버퍼 대기 시간
log_disk_wait_time	xs:nonNegativeInteger	log_disk_wait_time - 로그 디스크 대기 시간
log_disk_waits_total	xs:nonNegativeInteger	log_disk_waits_total - 총 로그 디스크 대기 수
pool_data_l_reads	xs:nonNegativeInteger	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
pool_data_p_reads	xs:nonNegativeInteger	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
pool_data_writes	xs:nonNegativeInteger	pool_data_writes - 버퍼 풀 데이터 쓰기 수
pool_index_l_reads	xs:nonNegativeInteger	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
pool_index_p_reads	xs:nonNegativeInteger	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
pool_index_writes	xs:nonNegativeInteger	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
pool_read_time	xs:nonNegativeInteger	pool_read_time - 총 버퍼 풀 실제 읽기 시간
pool_temp_data_l_reads	xs:nonNegativeInteger	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
pool_temp_data_p_reads	xs:nonNegativeInteger	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
pool_temp_index_l_reads	xs:nonNegativeInteger	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
pool_temp_index_p_reads	xs:nonNegativeInteger	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
pool_temp_xda_l_reads	xs:nonNegativeInteger	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수



표 110. MON\_GET\_UNIT\_OF\_WORK\_DETAILS에 대해 리턴된 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
pool_temp_xda_p_reads	xs:nonNegativeInteger	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수
pool_write_time	xs:nonNegativeInteger	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
pool_xda_l_reads	xs:nonNegativeInteger	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
pool_xda_p_reads	xs:nonNegativeInteger	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
pool_xda_writes	xs:nonNegativeInteger	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
num_log_buffer_full	xs:nonNegativeInteger	num_log_buffer_full - 전체 로그 버퍼 수
rqsts_completed_total	xs:nonNegativeInteger	rqsts_completed_total - 완료된 총 요청 수
rows_modified	xs:nonNegativeInteger	rows_modified - 수정된 행 수
rows_read	xs:nonNegativeInteger	rows_read - 읽은 행 수
rows_returned	xs:nonNegativeInteger	rows_returned - 리턴된 행 수
tcpip_rcv_volume	xs:nonNegativeInteger	tcpip_rcv_volume - TCP/IP 수신 볼륨
tcpip_rcv_wait_time	xs:nonNegativeInteger	tcpip_rcv_wait_time - TCP/IP 수신 대기 시간
tcpip_recvs_total	xs:nonNegativeInteger	tcpip_recvs_total - 총 TCP/IP 수신 수
tcpip_send_volume	xs:nonNegativeInteger	tcpip_send_volume - TCP/IP 전송 볼륨
tcpip_send_wait_time	xs:nonNegativeInteger	tcpip_send_wait_time - TCP/IP 전송 대기 시간
tcpip_sends_total	xs:nonNegativeInteger	tcpip_sends_total - 총 TCP/IP 전송 수
total_app_rqst_time	xs:nonNegativeInteger	total_app_rqst_time - 총 응용프로그램 요청 시간
total_cpu_time	xs:nonNegativeInteger	total_cpu_time - 총 CPU 시간
total_rqst_time	xs:nonNegativeInteger	total_rqst_time - 총 요청 시간
total_section_sort_time	xs:nonNegativeInteger	total_section_sort_time - 전체 섹션 정렬 시간
total_section_sort_proc_time	xs:nonNegativeInteger	total_section_sort_proc_time - 전체 섹션 정렬 처리 시간
total_section_sorts	xs:nonNegativeInteger	total_section_sorts - 전체 섹션 정렬
total_sorts	xs:nonNegativeInteger	total_sorts - 전체 정렬
post_threshold_sorts	xs:nonNegativeInteger	post_threshold_sorts - 포스트 임계값 정렬
post_shrthreshold_sorts	xs:nonNegativeInteger	post_shrthreshold_sorts - 포스트 공유 임계값 정렬
sort_overflows	xs:nonNegativeInteger	sort_overflows - 정렬 오버플로우
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총계
total_wait_time	xs:nonNegativeInteger	total_wait_time - 총 대기 시간
wlm_queue_time_total	xs:nonNegativeInteger	wlm_queue_time_total - 워크로드 관리 프로그램 총 큐 시간
wlm_queue_assignments_total	xs:nonNegativeInteger	wlm_queue_assignments_total - 워크로드 관리 프로그램 총 큐 지정 수

## MON\_GET\_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기

MON\_GET\_WORKLOAD 테이블 함수는 하나 이상의 워크로드에 대한 메트릭을 리턴합니다.



## 구문

▶▶ MON\_GET\_WORKLOAD(—workload\_name—, —member—) ◀◀

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *workload\_name*

리턴될 메트릭에 대한 워크로드를 지정하는 VARCHAR

(128) 유형의 입력 인수입니다. 인수가 NULL이거나 비어 있는 문자열인 경우, 메트릭이 모든 워크로드에 대해 리턴됩니다.

#### *member*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. NULL 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

### 권한 부여

MON\_GET\_WORKLOAD 함수에 대한 EXECUTE 특권.

### 예 :

가장 긴 잠금 대기 시간순으로 정렬하여 구성원에서 집계된 잠금 정보를 각 워크로드별로 표시합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

WORKLOAD_NAME	TOTAL_LOCK_WAIT_TIME	TOTAL_LOCK_WAITS	...
-----	-----	-----	...
SYSDEFAULTADMWORKLOAD	0	0	...
SYSDEFAULTUSERWORKLOAD	0	0	...

2개의 레코드가 선택되었습니다.

쿼리의 출력(계속).

...	TOTAL_LOCK_TIMEOUTS	TOTAL_LOCK_ESCALS
-----	-----	-----
...	0	0
...	0	0

## 사용 시 참고사항

메트릭은 식별된 워크로드 오브젝트에 맵핑된 연결로 제출한 요청에 대한 모든 메트릭의 누적을 나타내는 MON\_GET\_WORKLOAD 테이블 함수로 리턴됩니다. 메트릭은 작업 단위(UOW) 경계의 워크로드로, 요청 실행 중 정기적으로 롤업됩니다. 그러므로 이 테이블 함수로 보고된 값은 최근 롤업 시 시스템의 현재 상태를 반영합니다. 메트릭의 값은 엄격하게 증가합니다. 시간 간격에 대해 제공된 메트릭 값을 판별하려면, MON\_GET\_WORKLOAD 테이블 함수를 사용하여 간격의 시작 및 종료 시점의 메트릭을 쿼리하여 차이를 계산하십시오.

요청 메트릭은 데이터베이스 레벨에서의 *mon\_req\_metrics* 데이터베이스 구성 매개변수 및 서비스 수퍼 클래스의 COLLECT REQUEST METRICS 절을 통해 제어됩니다. 상위 서비스 수퍼 클래스에서 요청 메트릭을 사용할 수 있는 서비스 서브클래스에서 에이전트가 요청을 처리하거나 전체 데이터베이스에 대해 요청 메트릭 콜렉션을 사용할 수 있는 경우에만 요청에 대해 메트릭이 수집됩니다. 디폴트로 데이터베이스 레벨에서 요청 메트릭을 사용할 수 있습니다. 요청 메트릭이 데이터베이스 레벨에서 사용 불가능하고 서비스 수퍼 클래스에 대해 사용 불가능한 경우, 해당 서비스 수퍼 클래스에 맵핑된 각 워크로드에 보고된 메트릭은 증가를 중지시킵니다. (또는 요청 메트릭이 데이터베이스 활성화 시간에 사용 불가능한 경우 0이 됩니다.)

MON\_GET\_WORKLOAD 테이블 함수는 워크로드 및 구성원별로 한 개의 행을 리턴합니다. 워크로드(구성원) 또는 구성원(하나 이상의 서비스 클래스)에서의 집계는 수행되지 않습니다. 그러나 SQL 쿼리를 통해 집계가 이루어질 수 있습니다(예 참조).

## 리턴된 정보

표 111. MON\_GET\_WORKLOAD에 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
WORKLOAD_NAME	VARCHAR (128)	workload_name - 워크로드 이름
WORKLOAD_ID	INTEGER	workload_id - 워크로드 ID
MEMBER	SMALLINT	member - 데이터베이스 구성원
ACT_ABORTED_TOTAL	BIGINT	act_aborted_total - 중단된 총 활동 수
ACT_COMPLETED_TOTAL	BIGINT	act_completed_total - 완료된 총 활동 수
ACT_REJECTED_TOTAL	BIGINT	act_rejected_total - 거부된 총 활동 수
AGENT_WAIT_TIME	BIGINT	agent_wait_time - 에이전트 대기 시간
AGENT_WAITS_TOTAL	BIGINT	agent_waits_total - 총 에이전트 대기 수
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수

표 111. MON\_GET\_WORKLOAD에 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
CLIENT_IDLE_WAIT_TIME	BIGINT	client_idle_wait_time - 클라이언트 유휴 대기 시간
DEADLOCKS	BIGINT	deadlocks - 발견된 교착 상태 수
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
FCM_RECV_VOLUME	BIGINT	fcm_recv_volume - FCM recv 볼륨
FCM_RECVS_TOTAL	BIGINT	fcm_recvs_total - 총 FCM 수신 수
FCM_SEND_VOLUME	BIGINT	fcm_send_volume - FCM 전송 볼륨
FCM_SENDS_TOTAL	BIGINT	fcm_sends_total - 총 FCM 전송 수
FCM_RECV_WAIT_TIME	BIGINT	fcm_recv_wait_time - FCM recv 대기 시간
FCM_SEND_WAIT_TIME	BIGINT	fcm_send_wait_time - FCM 전송 대기 시간
IPC_RECV_VOLUME	BIGINT	ipc_recv_volume - 프로세스간 통신 recv 볼륨
IPC_RECV_WAIT_TIME	BIGINT	ipc_recv_wait_time - 프로세스간 통신 recv 대기 시간
IPC_RECVS_TOTAL	BIGINT	ipc_recvs_total - 총 프로세스간 통신 수신 수
IPC_SEND_VOLUME	BIGINT	ipc_send_volume - 프로세스간 통신 전송 볼륨
IPC_SEND_WAIT_TIME	BIGINT	ipc_send_wait_time - 프로세스간 통신 전송 대기 시간
IPC_SENDS_TOTAL	BIGINT	ipc_sends_total - 총 프로세스간 통신 전송 수
LOCK_ESCALS	BIGINT	lock_escalations - 잠금 에스컬레이션 수
LOCK_TIMEOUTS	BIGINT	lock_timeouts - 잠금 시간종료 수
LOCK_WAIT_TIME	BIGINT	lock_wait_time - 잠금 대기 시간

표 111. MON\_GET\_WORKLOAD에 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_WAITS	BIGINT	lock_waits - 잠금 대기 수
LOG_BUFFER_WAIT_TIME	BIGINT	log_buffer_wait_time - 로그 버퍼 대기 시간
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - 전체 로그 버퍼 수
LOG_DISK_WAIT_TIME	BIGINT	log_disk_wait_time - 로그 디스크 대기 시간
LOG_DISK_WAITS_TOTAL	BIGINT	log_disk_waits_total - 총 로그 디스크 대기 수
RQSTS_COMPLETED_TOTAL	BIGINT	rqsts_completed_total - 완료된 총 요청 수
ROWS_MODIFIED	BIGINT	rows_modified - 수정된 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_RETURNED	BIGINT	rows_returned - 리턴된 행 수
TCPIP_RECV_VOLUME	BIGINT	tcPIP_recv_volume - TCP/IP 수신 볼륨
TCPIP_SEND_VOLUME	BIGINT	tcPIP_send_volume - TCP/IP 전송 볼륨
TCPIP_RECV_WAIT_TIME	BIGINT	tcPIP_recv_wait_time - TCP/IP 수신 대기 시간
TCPIP_RECVS_TOTAL	BIGINT	tcPIP_recvS_total - 총 TCP/IP 수신 수
TCPIP_SEND_WAIT_TIME	BIGINT	tcPIP_send_wait_time - TCP/IP 전송 대기 시간
TCPIP_SENDS_TOTAL	BIGINT	tcPIP_sendS_total - 총 TCP/IP 전송 수
TOTAL_APP_RQST_TIME	BIGINT	total_app_rqst_time - 총 응용프로그램 요청 시간
TOTAL_RQST_TIME	BIGINT	total_rqst_time - 총 요청 시간
WLM_QUEUE_TIME_TOTAL	BIGINT	wlm_queue_time_total - 워크로드 관리 프로그램 총 큐 시간
WLM_QUEUE_ASSIGNMENTS_TOTAL	BIGINT	wlm_queue_assignments_total - 워크로드 관리 프로그램 총 큐 지정 수
TOTAL_CPU_TIME	BIGINT	total_cpu_time - 총 CPU 시간
TOTAL_WAIT_TIME	BIGINT	total_wait_time - 총 대기 시간
APP_RQSTS_COMPLETED_TOTAL	BIGINT	app_rqsts_completed_total - 완료된 총 응용프로그램 요청 수
TOTAL_SECTION_SORT_TIME	BIGINT	total_section_sort_time - 전체 섹션 정렬 시간
TOTAL_SECTION_SORT_PROC_TIME	BIGINT	total_section_sort_proc_time - 전체 섹션 정렬 처리 시간
TOTAL_SECTION_SORTS	BIGINT	total_section_sorts - 전체 섹션 정렬
TOTAL_SORTS	BIGINT	total_sorts - 전체 정렬 수
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - 포스트 임계값 정렬
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - 포스트 공유 임계값 정렬
SORT_OVERFLOWS	BIGINT	sort_overflows - 정렬 오버플로우
TOTAL_COMPILE_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMPILE_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMPILATIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_IMPLICIT_COMPILE_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_IMPLICIT_COMPILE_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_IMPLICIT_COMPILATIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_SECTION_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_SECTION_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨

표 111. MON\_GET\_WORKLOAD에 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TOTAL_APP_SECTION_EXECUTIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ACT_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ACT_WAIT_TIME	BIGINT	나중에 사용하기 위해 예약됨
ACT_RQSTS_TOTAL	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROUTINE_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROUTINE_INVOCATIONS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMMIT_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_COMMIT_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_APP_COMMITS	BIGINT	나중에 사용하기 위해 예약됨
INT_COMMITS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROLLBACK_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_ROLLBACK_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_APP_ROLLBACKS	BIGINT	나중에 사용하기 위해 예약됨
INT_ROLLBACKS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_RUNSTATS_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_RUNSTATS_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_RUNSTATS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_REORG_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_REORG_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_REORGS	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_LOAD_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_LOAD_PROC_TIME	BIGINT	나중에 사용하기 위해 예약됨
TOTAL_LOADS	BIGINT	나중에 사용하기 위해 예약됨
CAT_CACHE_INSERTS	BIGINT	나중에 사용하기 위해 예약됨
CAT_CACHE_LOOKUPS	BIGINT	나중에 사용하기 위해 예약됨
PKG_CACHE_INSERTS	BIGINT	나중에 사용하기 위해 예약됨
PKG_CACHE_LOOKUPS	BIGINT	나중에 사용하기 위해 예약됨
THRESH_VIOLATIONS	BIGINT	나중에 사용하기 위해 예약됨
NUM_LW_THRESH_EXCEEDED	BIGINT	나중에 사용하기 위해 예약됨
ADDITIONAL_DETAILS	BLOB(100K)	나중에 사용하기 위해 예약됨

## MON\_GET\_WORKLOAD\_DETAILS 테이블 함수 - 워크로드 메트릭 세부사항 가져오기

MON\_GET\_WORKLOAD\_DETAILS 테이블 함수는 하나 이상의 워크로드에 대한 메트릭 세부사항을 리턴합니다.

## 구문

►►—MON\_GET\_WORKLOAD\_DETAILS—(—workload\_name—,—member—)—————►►

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *workload\_name*

메트릭이 리턴될 특정 워크로드를 지정하는 유형 VARCHAR (128)의 입력 인수. 인수가 널(null)이거나 빈 문자열이면 모든 워크로드에 대해 메트릭이 리턴됩니다.

#### *member*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 구성원에는 -1을 지정하고 모든 데이터베이스 구성원에는 -2를 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

### 권한 부여

MON\_GET\_WORKLOAD\_DETAILS 함수에서의 EXECUTE 특권.

### 예 :

가장 높은 잠금 대기 시간 순서로 구성원에서 집계된 각 워크로드에 대한 잠금 정보를 표시합니다.

```
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;
```

이 쿼리의 출력 예는 다음과 같습니다.

WORKLOAD_NAME	TOTAL_LOCK_WAIT_TIME	TOTAL_LOCK_WAITS	...
SYSDEFAULTADMWORKLOAD	0	0	...
SYSDEFAULTUSERWORKLOAD	0	0	...

2개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	TOTAL_LOCK_TIMEOUTS	TOTAL_LOCK_ESCALS
...	0	0
...	0	0

## 사용 시 참고사항

MON\_GET\_WORKLOAD\_DETAILS 함수에서 리턴된 메트릭은 식별된 워크로드 오브젝트에 매핑된 연결에서 제출되는 요청에 대한 모든 메트릭의 누적을 나타냅니다. 이 함수는 MON\_GET\_WORKLOAD 테이블 함수와 유사합니다.

- MON\_GET\_WORKLOAD 테이블 함수는 컬럼 기반의 형식에서 가장 일반적으로 사용되는 메트릭을 리턴하며 메트릭을 검색하는 성능이 가장 효율적인 방법입니다.
- MON\_GET\_WORKLOAD\_DETAILS 테이블 함수는 XML 문서 형식에서 사용 가능한 전체 메트릭 세트를 리턴하며 출력 형식에 대해 최대 유연성을 제공합니다. XML 기반 출력은 XML 구문 분석기에서 직접 구문 분석되거나 XMLTABLE 함수를 사용하여 관계형 형식으로 변환될 수 있습니다(예 참조).

메트릭은 요청 실행 중 주기적으로 작업 단위 경계에 있는 워크로드로 롤업됩니다. 그러므로 이 테이블 함수로 보고된 값은 최근 롤업 시 시스템의 현재 상태를 반영합니다. 메트릭의 값이 정확히 증가됩니다. 시간 간격에 대해 주어진 메트릭 값을 판별하려면 MON\_GET\_WORKLOAD\_DETAILS 테이블 함수를 사용하여 간격의 시작과 종료 시 메트릭을 조회하며 차이를 계산합니다.

요청 메트릭은 데이터베이스 레벨에서의 **mon\_req\_metrics** 데이터베이스 구성 매개변수 및 서비스 수퍼 클래스의 COLLECT REQUEST METRICS절을 통해 제어됩니다. 상위 서비스 수퍼 클래스에서 요청 메트릭을 사용할 수 있는 서비스 서브클래스에서 에이전트가 요청을 처리하거나 전체 데이터베이스에 대해 요청 메트릭 콜렉션을 사용할 수 있는 경우에만 요청에 대해 메트릭이 수집됩니다. 디폴트로 데이터베이스 레벨에서 요청 메트릭을 사용할 수 있습니다. 요청 메트릭이 데이터베이스 레벨에서 사용 불가능하고 서비스 수퍼 클래스에 대해 사용 불가능한 경우, 해당 서비스 수퍼 클래스에 매핑된 각 워크로드에 보고된 메트릭은 증가를 중지시킵니다. (또는 요청 메트릭이 데이터베이스 활성화 시간에 사용 불가능한 경우 0이 됩니다.)



MON\_GET\_WORKLOAD\_DETAILS 테이블 함수는 워크로드 및 구성원별로 한 개의 데이터 행을 리턴합니다. 워크로드(구성원) 또는 구성원(하나 이상의 서비스 클래스)에서의 집계는 수행되지 않습니다. 그러나 예에서 보는 것과 같이 SQL 쿼리를 통해 집계는 이루어질 수 있습니다.

DETAILS 컬럼에 리턴된 XML 문서의 스키마는 sqllib/misc/DB2MonRoutines.xsd 파일로 사용 가능합니다. sqllib/misc/DB2MonCommon.xsd 파일에서 더 자세한 내용을 볼 수 있습니다.

## 리턴된 정보

표 112. MON\_GET\_WORKLOAD\_DETAILS에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
WORKLOAD_NAME	VARCHAR (128)	workload_name - 워크로드 이름
WORKLOAD_ID	INTEGER	workload_id - 워크로드 ID
MEMBER	SMALLINT	member - 데이터베이스 구성원
DETAILS	BLOB(1M)	워크로드의 세부사항 메트릭을 포함하는 XML 문서. 이 문서의 요소 설명은 표 113의 내용을 참조하십시오.

다음 예는 DETAILS 컬럼에서 리턴되는 XML 문서의 구조를 표시합니다.

```
<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>
```

전체 스키마에 대해서는 sqllib/misc/DB2MonRoutines.xsd의 내용을 참조하십시오.

표 113. MON\_GET\_WORKLOAD\_DETAILS에 대해 리턴되는 메트릭 세부사항

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
workload_name	xs:string(128)	workload_name - 워크로드 이름
workload_id	xs:nonNegativeInteger	workload_id - 워크로드 ID
member	xs:nonNegativeInteger	member - 데이터베이스 구성원
act_aborted_total	xs:nonNegativeInteger	act_aborted_total - 중단된 총 활동 수
act_completed_total	xs:nonNegativeInteger	act_completed_total - 완료된 총 활동 수
act_rejected_total	xs:nonNegativeInteger	act_rejected_total - 거부된 총 활동 수
agent_wait_time	xs:nonNegativeInteger	agent_wait_time - 에이전트 대기 시간
agent_waits_total	xs:nonNegativeInteger	agent_waits_total - 총 에이전트 대기 수
app_rqsts_completed_total	xs:nonNegativeInteger	app_rqsts_completed_total - 완료된 총 응용프로그램 요청 수
audit_events_total	xs:nonNegativeInteger	audit_events_total - 총 감사 이벤트 수
audit_subsystem_wait_time	xs:nonNegativeInteger	audit_subsystem_wait_time - 감사 서브시스템 대기 시간
audit_subsystem_waits_total	xs:nonNegativeInteger	audit_subsystem_waits_total - 전체 감사 서브시스템 대기



표 113. MON\_GET\_WORKLOAD\_DETAILS에 대해 리턴되는 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
audit_file_write_wait_time	xs:nonNegativeInteger	audit_file_write_wait_time - 감사 파일 작성 대기 시간
audit_file_writes_total	xs:nonNegativeInteger	audit_file_writes_total - 작성된 총 감사 파일 수
client_idle_wait_time	xs:nonNegativeInteger	client_idle_wait_time - 클라이언트 유휴 대기 시간
교착 상태	xs:nonNegativeInteger	deadlocks - 발견된 교착 상태 수
diaglog_writes_total	xs:nonNegativeInteger	diaglog_writes_total - 도표 로그 전체 작성 수
diaglog_write_wait_time	xs:nonNegativeInteger	diaglog_write_wait_time - 도표 로그 작성 시간
direct_read_time	xs:nonNegativeInteger	direct_read_time - 직접 읽기 시간
direct_write_time	xs:nonNegativeInteger	direct_write_time - 직접 쓰기 시간
direct_read_reqs	xs:nonNegativeInteger	direct_read_reqs - 직접 읽기 요청 수
direct_reads	xs:nonNegativeInteger	direct_reads - 데이터베이스에서 직접 읽기 수
direct_write_reqs	xs:nonNegativeInteger	direct_write_reqs - 직접 쓰기 요청 수
direct_writes	xs:nonNegativeInteger	direct_writes - 데이터베이스에 직접 쓰기 수
fcm_rcv_volum	xs:nonNegativeInteger	fcm_rcv_volum - FCM 수신 볼륨
fcm_rcv_wait_time	xs:nonNegativeInteger	fcm_rcv_wait_time - FCM 수신 대기 시간
fcm_rcvs_total	xs:nonNegativeInteger	fcm_rcvs_total - 총 FCM 수신 수
fcm_message_rcv_volum	xs:nonNegativeInteger	fcm_message_rcv_volum - FCM 메시지 수신 볼륨
fcm_message_rcvs_total	xs:nonNegativeInteger	fcm_message_rcvs_total - 총 FCM 메시지 수신 수
fcm_message_rcv_wait_time	xs:nonNegativeInteger	fcm_message_rcv_wait_time - FCM 메시지 수신 대기 시간
fcm_message_send_volum	xs:nonNegativeInteger	fcm_message_send_volum - FCM 메시지 전송 볼륨
fcm_message_send_wait_time	xs:nonNegativeInteger	fcm_message_send_wait_time - FCM 메시지 전송 대기 시간
fcm_message_sends_total	xs:nonNegativeInteger	fcm_message_sends_total - 총 FCM 메시지 전송 수
fcm_send_volum	xs:nonNegativeInteger	fcm_send_volum - FCM 전송 볼륨
fcm_send_wait_time	xs:nonNegativeInteger	fcm_send_wait_time - FCM 전송 대기 시간
fcm_sends_total	xs:nonNegativeInteger	fcm_sends_total - 총 FCM 전송 수
fcm_tq_rcv_wait_time	xs:nonNegativeInteger	fcm_tq_rcv_wait_time - FCM 테이블 큐 수신 대기 시간
fcm_tq_send_wait_time	xs:nonNegativeInteger	fcm_tq_send_wait_time - FCM 테이블 큐 전송 대기 시간
fcm_tq_rcv_volum	xs:nonNegativeInteger	fcm_tq_rcv_volum - FCM 테이블 큐 수신 볼륨
fcm_tq_rcvs_total	xs:nonNegativeInteger	fcm_tq_rcvs_total - FCM 테이블 큐 수신 총계
fcm_tq_send_volum	xs:nonNegativeInteger	fcm_tq_send_volum - FCM 테이블 큐 전송 볼륨
fcm_tq_sends_total	xs:nonNegativeInteger	fcm_tq_sends_total - FCM 테이블 큐 전송 총계
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총계
ipc_rcv_volum	xs:nonNegativeInteger	ipc_rcv_volum - 프로세스간 통신 수신 볼륨
ipc_rcv_wait_time	xs:nonNegativeInteger	ipc_rcv_wait_time - 프로세스간 통신 수신 대기 시간
ipc_rcvs_total	xs:nonNegativeInteger	ipc_rcvs_total - 총 프로세스간 통신 수신 수
ipc_send_volum	xs:nonNegativeInteger	ipc_send_volum - 프로세스간 통신 전송 볼륨
ipc_send_wait_time	xs:nonNegativeInteger	ipc_send_wait_time - 프로세스간 통신 전송 대기 시간
ipc_sends_total	xs:nonNegativeInteger	ipc_sends_total - 총 프로세스간 통신 전송 수
lock_escals	xs:nonNegativeInteger	lock_escals - 잠금 에스컬레이션 수
lock_timeouts	xs:nonNegativeInteger	lock_timeouts - 잠금 시간종료 수
lock_wait_time	xs:nonNegativeInteger	lock_wait_time - 잠금 대기 시간

표 113. MON\_GET\_WORKLOAD\_DETAILS에 대해 리턴되는 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
lock_waits	xs:nonNegativeInteger	lock_waits - 잠금 대기 수
log_buffer_wait_time	xs:nonNegativeInteger	log_buffer_wait_time - 로그 버퍼 대기 시간
log_disk_wait_time	xs:nonNegativeInteger	log_disk_wait_time - 로그 디스크 대기 시간
log_disk_waits_total	xs:nonNegativeInteger	log_disk_waits_total - 총 로그 디스크 대기 수
pool_data_l_reads	xs:nonNegativeInteger	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
pool_data_p_reads	xs:nonNegativeInteger	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
pool_data_writes	xs:nonNegativeInteger	pool_data_writes - 버퍼 풀 데이터 쓰기 수
pool_index_l_reads	xs:nonNegativeInteger	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
pool_index_p_reads	xs:nonNegativeInteger	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
pool_index_writes	xs:nonNegativeInteger	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
pool_read_time	xs:nonNegativeInteger	pool_read_time - 총 버퍼 풀 실제 읽기 시간
pool_temp_data_l_reads	xs:nonNegativeInteger	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
pool_temp_data_p_reads	xs:nonNegativeInteger	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
pool_temp_index_l_reads	xs:nonNegativeInteger	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
pool_temp_index_p_reads	xs:nonNegativeInteger	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
pool_temp_xda_l_reads	xs:nonNegativeInteger	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
pool_temp_xda_p_reads	xs:nonNegativeInteger	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수
pool_write_time	xs:nonNegativeInteger	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
pool_xda_l_reads	xs:nonNegativeInteger	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
pool_xda_p_reads	xs:nonNegativeInteger	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
pool_xda_writes	xs:nonNegativeInteger	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
num_log_buffer_full	xs:nonNegativeInteger	num_log_buffer_full - 전체 로그 버퍼 수
rqsts_completed_total	xs:nonNegativeInteger	rqsts_completed_total - 완료된 총 요청 수
rows_modified	xs:nonNegativeInteger	rows_modified - 수정된 행 수
rows_read	xs:nonNegativeInteger	rows_read - 읽은 행 수
rows_returned	xs:nonNegativeInteger	rows_returned - 리턴된 행 수
tcpip_rcv_volume	xs:nonNegativeInteger	tcpip_rcv_volume - TCP/IP 수신 볼륨
tcpip_rcv_wait_time	xs:nonNegativeInteger	tcpip_rcv_wait_time - TCP/IP 수신 대기 시간
tcpip_recvs_total	xs:nonNegativeInteger	tcpip_recvs_total - 총 TCP/IP 수신 수
tcpip_send_volume	xs:nonNegativeInteger	tcpip_send_volume - TCP/IP 전송 볼륨
tcpip_send_wait_time	xs:nonNegativeInteger	tcpip_send_wait_time - TCP/IP 전송 대기 시간
tcpip_sends_total	xs:nonNegativeInteger	tcpip_sends_total - 총 TCP/IP 전송 수
total_app_rqst_time	xs:nonNegativeInteger	total_app_rqst_time - 총 응용프로그램 요청 시간
total_cpu_time	xs:nonNegativeInteger	total_cpu_time - 총 CPU 시간
total_rqst_time	xs:nonNegativeInteger	total_rqst_time - 총 요청 시간
total_section_sort_time	xs:nonNegativeInteger	total_section_sort_time - 전체 섹션 정렬 시간
total_section_sort_proc_time	xs:nonNegativeInteger	total_section_sort_proc_time - 전체 섹션 정렬 처리 시간
total_section_sorts	xs:nonNegativeInteger	total_section_sorts - 전체 섹션 정렬 수

표 113. MON\_GET\_WORKLOAD\_DETAILS에 대해 리턴되는 메트릭 세부사항 (계속)

요소 이름	데이터 유형	설명 또는 해당 모니터 요소
total_sorts	xs:nonNegativeInteger	total_sorts - 전체 정렬 수
post_threshold_sorts	xs:nonNegativeInteger	post_threshold_sorts - 포스트 임계값 정렬
post_shrthreshold_sorts	xs:nonNegativeInteger	post_shrthreshold_sorts - 포스트 공유 임계값 정렬
sort_overflows	xs:nonNegativeInteger	sort_overflows - 정렬 오버플로우
tq_tot_send_spills	xs:nonNegativeInteger	tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총계
total_wait_time	xs:nonNegativeInteger	total_wait_time - 총 대기 시간
wlm_queue_time_total	xs:nonNegativeInteger	wlm_queue_time_total - 워크로드 관리 프로그램 총 큐 시간
wlm_queue_assignments_total	xs:nonNegativeInteger	wlm_queue_assignments_total - 워크로드 관리 프로그램 총 큐 지정 수



---

## 제 12 장 MQSeries 루틴

---

### MQPUBLISH

MQPUBLISH 함수는 MQSeries에 데이터를 발행합니다. 자세한 내용은 <http://www.ibm.com/software/MQSeries>를 방문하십시오.

MQPUBLISH 함수는 *msg-data*에 포함되는 데이터를 *publisher-service*에 지정된 MQSeries publisher에 발행하고 *service-policy*에서 정의된 서비스 규정 품질을 사용합니다. 메시지에 대한 선택적 주제를 지정할 수 있으며, 선택적 사용자 정의 메시지 상관 ID도 지정할 수 있습니다.

결과의 데이터 유형은 VARCHAR(1)입니다. 함수의 결과는 성공할 경우 '1' 또는 실패할 경우 '0'입니다.

#### 구문

```
MQPUBLISH ( ( publisher-service, service-policy, msg-data,  
, topic, (1), correl-id ) )
```

#### 주:

1 *service* 및 *policy*가 먼저 지정되어야 *correl-id*를 지정할 수 있습니다.

스키마는 비트랜잭션 메시지 큐 함수의 경우 DB2MQ이고, 1단계 커밋 트랜잭션 MQ 함수의 경우 DB2MQ1C입니다.

#### 함수 매개변수

##### *publisher-service*

메시지를 보낼 논리적 MQSeries 목적지를 포함하는 문자열. 지정되는 경우, *publisher-service*는 퍼블리셔 서비스 유형 값으로 'P'를 가진 DB2MQ.MQPUBSUB 테이블에 정의된 퍼블리셔 서비스 지점을 참조해야 합니다. *publisher-service*가 지정되지 않으면, DB2.DEFAULT.PUBLISHER가 사용됩니다. *publisher-service*의 최대 크기는 48바이트입니다.

##### *service-policy*

해당 메시지의 처리에 사용할 MQSeries 서비스 규정을 포함하는 문자열. 지정되는

경우, *service-policy*는 DB2MQ.MQPOLICY 테이블에 정의된 규정을 참조해야 합니다. 서비스 규정은 이 메시징 조작에 적용되어야 하는 서비스 품질 옵션 세트를 정의합니다. 이들 옵션에는 메시지 우선순위와 메시지 지속성이 들어 있습니다. *service-policy*가 지정되지 않으면, 디폴트 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48바이트입니다.

#### *msg-data*

MQSeries를 통해 보낼 데이터를 포함하는 문자열 표현식. VARCHAR 문자열 표현식의 최대 크기는 32 000바이트이고 CLOB 문자열 표현식의 최대 크기는 1MB입니다.

#### *topic*

메시지 발행에 대한 주제를 포함하는 문자열. 표현식 주제가 지정되지 않으면 어떤 것도 메시지와 연관되지 않습니다. *topic*의 최대 크기는 40바이트입니다. 한 문자열에서 여러 주제를 지정할 수 있습니다(최대 40자 길이). 각 주제는 콜론으로 분리되어야 합니다. 예를 들어, "t1:t2:the third topic"은 메시지가 t1, t2 및 "the third topic"의 세 주제와 모두 연관되어 있음을 나타냅니다.

#### *correl-id*

이 메시지와 연관되는 상관 ID를 포함하는 선택적 문자열 표현식. *correl-id*가 종종 요청을 응답과 연관시키는 요청 및 응답 시나리오에 지정됩니다. 지정되지 않으면 상관 ID가 메시지에 추가되지 않습니다. *correl-id*의 최대 크기는 24바이트입니다.

### 예:

예 1: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 퍼블리셔 서비스(DB2.DEFAULT.PUBLISHER)에 문자열 "Testing 123"을 발행합니다. 메시지에 대한 상관 ID 또는 주제는 지정되지 않습니다.

```
VALUES MQPUBLISH('Testing 123')
```

예 2: 이 예에서는 "TESTS" 주제에서 publisher 서비스 "MYPUBLISHER"에 문자열 "Testing 345"를 발행합니다. 디폴트 규정이 사용되고 상관 ID는 지정되지 않습니다.

```
VALUES MQPUBLISH('MYPUBLISHER','Testing 345', 'TESTS')
```

예 3: 이 예에서는 "MYPOLICY" 규정을 사용하여 상관 ID "TEST1"과 함께 publisher 서비스 "MYPUBLISHER"에 문자열 "Testing 678"을 발행합니다. 디폴트 규정이 사용되고 상관 ID는 지정되지 않습니다.

```
VALUES MQPUBLISH('MYPUBLISHER','MYPOLICY','Testing 678','TESTS','TEST1')
```

예 4: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 상관 ID 없이 "TESTS" 주제에서 퍼블리셔 서비스 "MYPUBLISHER"에 문자열 "Testing 901"을 받습니다.

```
VALUES MQPUBLISH('Testing 901','TESTS')
```

## MQREAD

MQREAD 함수는 *service-policy*에 정의된 서비스 규정 품질을 사용하여 *receive-service*에서 지정된 MQSeries 위치로부터 메시지를 리턴합니다. 이 작업을 실행하면 *receive-service*와 연관된 큐에서 메시지가 제거되지 않지만, 대신 큐의 헤드에 있는 메시지를 리턴합니다.

결과 데이터 유형은 VARCHAR(32000)입니다. 리턴할 수 있는 메시지가 없으면 결과는 널(NULL) 값이 됩니다.

### 구문

```
MQREAD( ( receive-service [, service-policy ] ) )
```

스키마는 비트랜잭션 메시지 큐 함수의 경우 DB2MQ이고, 1단계 커밋 트랜잭션 MQ 함수의 경우 DB2MQ1C입니다.

### 함수 매개변수

#### *receive-service*

메시지가 수신될 논리적 MQSeries 목적지를 포함하는 문자열. 지정되는 경우, *receive-service*는 DB2MQ.MQSERVICE 테이블에 정의된 서비스 지점을 참조해야 합니다. 서비스 지점은 메시지를 보내거나 수신하는 논리적 끝점입니다. 서비스 지점 정의는 MQSeries 큐 관리 프로그램 및 큐의 이름을 포함합니다. *receive-service*가 지정되지 않으면, DB2.DEFAULT.SERVICE가 사용됩니다. *receive-service*의 최대 크기는 48바이트입니다.

#### *service-policy*

해당 메시지의 처리에 사용되는 MQSeries 서비스 규정을 포함하는 문자열. 지정되는 경우, *service-policy*는 DB2MQ.MQPOLICY 테이블에 정의된 규정을 참조해야 합니다. 서비스 규정은 이 메시징 조작에 적용되어야 하는 서비스 품질 옵션 세트를 정의합니다. 이들 옵션에는 메시지 우선순위와 메시지 지속성이 들어 있습니다. *service-policy*가 지정되지 않으면, 디폴트 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48바이트입니다.

예:

예 1: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐 헤드의 메시지를 읽습니다.

```
VALUES MQREAD()
```

예 2: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 "MYSERVICE" 서비스에서 지정된 큐 헤드의 메시지를 읽습니다.

```
VALUES MQREAD('MYSERVICE')
```

예 3: 이 예에서는 "MYPOLICY" 규정을 사용하여 "MYSERVICE" 서비스에서 지정된 큐 헤드의 메시지를 읽습니다.

```
VALUES MQREAD('MYSERVICE', 'MYPOLICY')
```

---

## MQREADALL

MQREADALL 테이블 함수는 *service-policy* 서비스 규정 품질을 사용하여 *receive-service*가 지정한 MQSeries 위치에서 메시지 및 메시지 메타데이터를 포함하는 테이블을 리턴합니다. 이 조작을 수행하면 *receive-service*와 연관된 큐에서 메시지가 제거되지 않습니다.

### 구문

```
MQREADALL ( ( receive-service , service-policy ) num-rows )
```

스키마는 비트랜잭션 메시지 큐 함수의 경우 DB2MQ이고, 1단계 커밋 트랜잭션 MQ 함수의 경우 DB2MQ1C입니다.

### 테이블 함수 매개변수

#### *receive-service*

메시지를 읽어들이는 논리적 MQSeries 목적지를 포함하는 문자열. 지정되는 경우, *receive-service*는 DB2MQ.MQSERVICE 테이블에 정의된 서비스 지점을 참조해야 합니다. 서비스 지점은 메시지를 보내거나 수신하는 논리적 끝점입니다. 서비스 지점 정의는 MQSeries 큐 관리 프로그램 및 큐의 이름을 포함합니다. *receive-service*가 지정되지 않으면, DB2.DEFAULT.SERVICE가 사용됩니다. *receive-service*의 최대 크기는 48바이트입니다.

#### *service-policy*

해당 메시지의 처리에 사용되는 MQSeries 서비스 규정을 포함하는 문자열. 지정되는 경우, *service-policy*는 DB2MQ.MQPOLICY 테이블에 정의된 규정을 참조합니다. 서비스 규정은 이 메시징 조작에 적용되어야 하는 서비스 품질 옵션 세트를



정의합니다. 이들 옵션에는 메시지 우선순위와 메시지 지속성이 들어 있습니다. *service-policy*가 지정되지 않으면, 디폴트 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48바이트입니다.

*num-rows*

함수에서 리턴될 최대 메시지 수를 포함하는 양의 정수

*num-rows*가 지정되면 *num-rows* 메시지의 최대값이 리턴됩니다. *num-rows*가 지정되지 않으면 사용 가능한 모든 메시지가 리턴됩니다.

**권한 부여**

MQREADALL 테이블 함수에 대한 EXECUTE 특권

**예:**

예 1: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐에서 모든 메시지를 수신합니다. 메시지와 모든 메타데이터가 테이블로 리턴됩니다.

```
SELECT * FROM table (MQREADALL()) AS T
```

예 2: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 MYSERVICE 서비스에서 지정된 큐 헤드에서 모든 메시지를 수신합니다. MSG 및 CORRELID 컬럼만 리턴됩니다.

```
SELECT T.MSG, T.CORRELID FROM table (MQREADALL('MYSERVICE')) AS T
```

예 3: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐 헤드를 읽습니다. '1234'의 CORRELID를 갖는 메시지만 리턴됩니다. 모든 컬럼이 리턴됩니다.

```
SELECT * FROM table (MQREADALL()) AS T WHERE T.CORRELID = '1234'
```

예 4: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐 헤드에서 처음 10개의 메시지를 수신합니다. 모든 컬럼이 리턴됩니다.

```
SELECT * FROM table (MQREADALL(10)) AS T
```

**리턴되는 정보**

표 114. MQREADALL 테이블 함수가 리턴한 정보

컬럼 이름	데이터 유형	설명
MSG	VARCHAR(32000)	MQSeries 메시지의 내용을 포함합니다.

표 114. MQREADALL 테이블 함수가 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명
CORRELID	VARCHAR(24)	메시지를 식별하기 위한 상관 ID를 포함합니다. 이 ID를 사용하여 큐에서 메시지를 선택할 수 있습니다. 요청 및 응답 시나리오의 경우, 상관 ID는 특정 요청에 대해 응답을 연관할 수 있게 합니다.
TOPIC	VARCHAR(40)	가능한 경우, 메시지를 발행한 주제를 포함합니다.
QNAME	VARCHAR(48)	메시지를 수신한 큐 이름을 포함합니다.
MSGID	CHAR(24)	이 메시지에 지정된 고유 MQSeries ID를 포함합니다.
MSGFORMAT	VARCHAR(8)	MQSeries에 의해 정의된 대로 메시지의 형식을 포함합니다. 일반적인 문자열은 MQSTR 형식입니다.

## MQREADALLCLOB

MQREADALLCLOB 테이블 함수는 *service-policy* 서비스 규정 품질을 사용하여 *receive-service*에서 지정한 MQSeries 위치로부터 메시지와 메시지 메타데이터가 포함된 테이블을 리턴합니다. 이 조사를 수행하면 *receive-service*와 연관된 큐에서 메시지가 제거되지 않습니다.

### 구문

```

MQREADALLCLOB
(
  (
    receive-service
    , service-policy
  )
  num-rows
)

```

스키마는 DB2MQ입니다.

### 테이블 함수 매개변수

#### *receive-service*

메시지를 읽어들이는 논리적 MQSeries 목적지를 포함하는 문자열. 지정되는 경우, *receive-service*는 DB2MQ.MQSERVICE 테이블에 정의된 서비스 지점을 참조해야 합니다. 서비스 지점은 메시지를 보내거나 수신하는 논리적 끝점입니다. 서비스 지점 정의는 MQSeries 큐 관리 프로그램 및 큐의 이름을 포함합니다. *receive-service*가 지정되지 않으면, DB2.DEFAULT.SERVICE가 사용됩니다. *receive-service*의 최대 크기는 48바이트입니다.

### *service-policy*

해당 메시지의 처리에 사용되는 MQSeries 서비스 규정을 포함하는 문자열. 지정되는 경우, *service-policy*는 DB2MQ.MQPOLICY 테이블에 정의된 규정을 참조합니다. 서비스 규정은 이 메시징 조작에 적용되어야 하는 서비스 품질 옵션 세트를 정의합니다. 이들 옵션에는 메시지 우선순위와 메시지 지속성이 들어 있습니다. *service-policy*가 지정되지 않으면, 디폴트 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48바이트입니다.

### *num-rows*

함수에서 리턴될 최대 메시지 수를 포함하는 양의 정수

*num-rows*가 지정되면 *num-rows* 메시지의 최대값이 리턴됩니다. *num-rows*가 지정되지 않으면 사용 가능한 모든 메시지가 리턴됩니다.

## 권한 부여

MQREADALLCLOB 테이블 함수에 대한 EXECUTE 특권

### 예:

예 1: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐에서 모든 메시지를 수신합니다. 메시지와 모든 메타데이터가 테이블로 리턴됩니다.

```
SELECT * FROM table (MQREADALLCLOB()) AS T
```

예 2: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 MYSERVICE 서비스에서 지정된 큐 헤드에서 모든 메시지를 수신합니다. MSG 및 CORRELID 컬럼만 리턴됩니다.

```
SELECT T.MSG, T.CORRELID FROM table (MQREADALLCLOB('MYSERVICE')) AS T
```

예 3: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐 헤드를 읽습니다. '1234'의 CORRELID를 갖는 메시지만 리턴됩니다. 모든 컬럼이 리턴됩니다.

```
SELECT * FROM table (MQREADALLCLOB()) AS T WHERE T.CORRELID = '1234'
```

예 4: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐 헤드에서 처음 10개의 메시지를 수신합니다. 모든 컬럼이 리턴됩니다.

```
SELECT * FROM table (MQREADALLCLOB(10)) AS T
```

## 리턴되는 정보

표 115. MQREADALLCLOB 테이블 함수가 리턴한 정보

컬럼 이름	데이터 유형	설명
MSG	CLOB(1M)	MQSeries 메시지의 내용을 포함합니다.
CORRELID	VARCHAR(24)	메시지를 식별하기 위한 상관 ID를 포함합니다. 이 ID를 사용하여 큐에서 메시지를 선택할 수 있습니다. 요청 및 응답 시나리오의 경우, 상관 ID는 특정 요청에 대해 응답을 연관할 수 있게 합니다.
TOPIC	VARCHAR(40)	가능한 경우, 메시지를 발행한 주제를 포함합니다.
QNAME	VARCHAR(48)	메시지를 수신한 큐 이름을 포함합니다.
MSGID	CHAR(24)	이 메시지에 지정된 고유 MQSeries ID를 포함합니다.
MSGFORMAT	VARCHAR(8)	MQSeries에 의해 정의된 대로 메시지의 형식을 포함합니다. 일반적인 문자열은 MQSTR 형식입니다.

## MQREADCLOB

MQREADCLOB 함수는 *service-policy*에 정의된 서비스 규정 품질을 사용하여 *receive-service*에서 지정한 MQSeries 위치로부터 메시지를 리턴합니다. 이 작업을 실행하면 *receive-service*와 연관된 큐에서 메시지가 제거되지 않지만, 대신 큐의 헤드에 있는 메시지를 리턴합니다.

결과 데이터 유형은 CLOB(1M)입니다. 리턴할 수 있는 메시지가 없으면 결과는 널 (NULL) 값이 됩니다.

### 구문

```

MQREADCLOB( ( receive-service [, service-policy ] ) )

```

스키마는 DB2MQ입니다.

### 함수 매개변수

#### *receive-service*

메시지가 수신될 논리적 MQSeries 목적지를 포함하는 문자열. 지정되는 경우, *receive-service*는 DB2MQ.MQSERVICE 테이블에 정의된 서비스 지점을 참조해야 합니다. 서비스 지점은 메시지를 보내거나 수신하는 논리적 끝점입니다. 서비스

지점 정의는 MQSeries 큐 관리 프로그램 및 큐의 이름을 포함합니다. *receive-service*가 지정되지 않으면, DB2.DEFAULT.SERVICE가 사용됩니다. *receive-service*의 최대 크기는 48바이트입니다.

#### *service-policy*

해당 메시지의 처리에 사용되는 MQSeries 서비스 규정을 포함하는 문자열. 지정되는 경우, *service-policy*는 DB2MQ.MQPOLICY 테이블에 정의된 규정을 참조해야 합니다. 서비스 규정은 이 메시지 조작에 적용되어야 하는 서비스 품질 옵션 세트를 정의합니다. 이들 옵션에는 메시지 우선순위와 메시지 지속성이 들어 있습니다. *service-policy*가 지정되지 않으면, 디폴트 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48바이트입니다.

#### 예:

예 1: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐 헤드의 메시지를 읽습니다.

```
VALUES MQREADCLOB()
```

예 2: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 "MYSERVICE" 서비스에서 지정된 큐 헤드의 메시지를 읽습니다.

```
VALUES MQREADCLOB('MYSERVICE')
```

예 3: 이 예에서는 "MYPOLICY" 규정을 사용하여 "MYSERVICE" 서비스에서 지정된 큐 헤드의 메시지를 읽습니다.

```
VALUES MQREADCLOB('MYSERVICE', 'MYPOLICY')
```

---

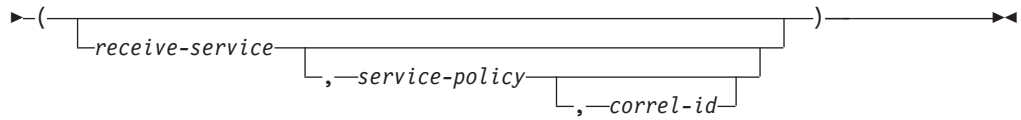
## MQRECEIVE

MQRECEIVE 함수는 *service-policy* 서비스 규정 품질을 사용하여 *receive-service*에서 지정된 MQSeries 위치로부터 메시지를 리턴합니다. 이 조작을 수행하면 *receive-service*와 연관된 큐에서 메시지가 제거됩니다. *correl-id*가 지정되면 일치하는 상관 ID를 갖는 첫 번째 메시지가 리턴됩니다. *correl-id*가 지정되지 않으면 큐 헤드의 메시지가 리턴됩니다.

결과의 데이터 유형은 VARCHAR(32000)입니다. 리턴할 수 있는 메시지가 없으면 결과는 널(NULL) 값이 됩니다.

### 구문

▶▶ MQRECEIVE —————▶▶



스키마는 비트랜잭션 메시지 큐 함수의 경우 DB2MQ이고, 1단계 커밋 트랜잭션 MQ 함수의 경우 DB2MQ1C입니다.

## 함수 매개변수

### *receive-service*

메시지가 수신되는 논리적 MQSeries 목적지를 포함하는 문자열. 지정되는 경우, *receive-service*는 DB2MQ.MQSERVICE 테이블에 정의된 서비스 지점을 참조해야 합니다. 서비스 지점은 메시지를 보내거나 수신하는 논리적 끝점입니다. 서비스 지점 정의는 MQSeries 큐 관리 프로그램 및 큐의 이름을 포함합니다. *receive-service*가 지정되지 않으면, DB2.DEFAULT.SERVICE가 사용됩니다. *receive-service*의 최대 크기는 48바이트입니다.

### *service-policy*

해당 메시지의 처리에 사용할 MQSeries 서비스 규정을 포함하는 문자열. 지정되는 경우, *service-policy*는 DB2MQ.MQPOLICY 테이블에 정의된 규정을 참조해야 합니다. 서비스 규정은 이 메시징 조각에 적용되어야 하는 서비스 품질 옵션 세트를 정의합니다. 이들 옵션에는 메시지 우선순위와 메시지 지속성이 들어 있습니다. *service-policy*가 지정되지 않으면, 디폴트 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48바이트입니다.

### *correl-id*

이 메시지와 연관된 선택적 상관 ID를 포함하는 문자열. *correl-id*가 종종 요청을 응답과 연관시키는 요청 및 응답 시나리오에 지정됩니다. *correl-id*가 지정되지 않으면 상관 ID가 지정되지 않습니다. *correl-id*의 최대 크기는 24바이트입니다.

## 예:

예 1: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐 헤드의 메시지를 수신합니다.

```
VALUES MQRECEIVE()
```

예 2: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 "MYSERVICE" 서비스에서 지정된 큐 헤드의 메시지를 수신합니다.

```
VALUES MQRECEIVE('MYSERVICE')
```

예 3: 이 예에서는 "MYPOLICY" 규정을 사용하여 "MYSERVICE" 서비스에서 지정되는 큐 헤드의 메시지를 수신합니다.

```
VALUES MQRECEIVE('MYSERVICE', 'MYPOLICY')
```

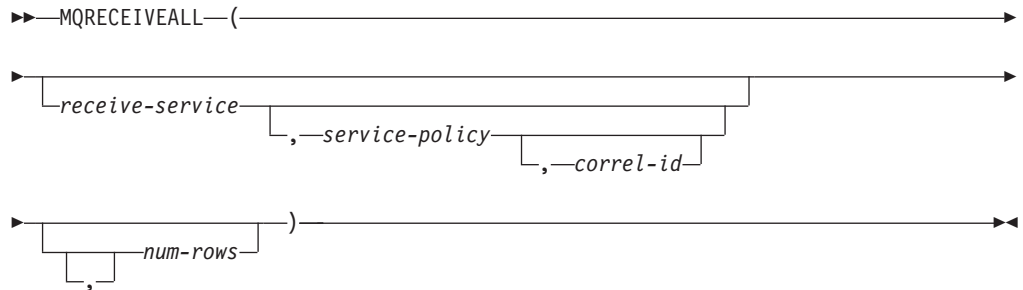
예 4: 이 예에서는 "MYPOLICY" 규정을 사용하여 "MYSERVICE" 서비스에서 지정된 큐 헤드로부터 '1234'와 일치하는 상관 ID를 갖는 첫 번째 메시지를 수신합니다.

```
VALUES MQRECEIVE('MYSERVICE', 'MYPOLICY', '1234')
```

## MQRECEIVEALL

MQRECEIVEALL 테이블 함수는 *service-policy* 서비스 규정 품질을 사용하여 *receive-service*가 지정한 MQSeries 위치에서 메시지 및 메시지 메타데이터를 포함하는 테이블을 리턴합니다. 이 조작을 수행하면 *receive-service*와 연관된 큐에서 메시지가 제거됩니다.

### 구문



스키마는 비트랜잭션 메시지 큐 함수의 경우 DB2MQ이고, 1단계 커밋 트랜잭션 MQ 함수의 경우 DB2MQ1C입니다.

### 테이블 함수 매개변수

#### *receive-service*

메시지가 수신되는 논리적 MQSeries 목적지를 포함하는 문자열. 지정되는 경우, *receive-service*는 DB2MQ.MQSERVICE 테이블에 정의된 서비스 지점을 참조해야 합니다. 서비스 지점은 메시지를 보내거나 수신하는 논리적 끝점입니다. 서비스 지점 정의는 MQSeries 큐 관리 프로그램 및 큐의 이름을 포함합니다. *receive-service*가 지정되지 않으면, DB2.DEFAULT.SERVICE가 사용됩니다. *receive-service*의 최대 크기는 48바이트입니다.

#### *service-policy*

해당 메시지의 처리에 사용되는 MQSeries 서비스 규정을 포함하는 문자열. 지정되는 경우, *service-policy*는 DB2MQ.MQPOLICY 테이블에 정의된 규정을 참조합니다. 서비스 규정은 이 메시징 조작에 적용되어야 하는 서비스 품질 옵션 세트를 정의합니다. 이들 옵션에는 메시지 우선순위와 메시지 지속성이 들어 있습니다. *service-policy*가 지정되지 않으면, 디폴트 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48바이트입니다.

### *correl-id*

이 메시지와 연관된 상관 ID를 포함하는 선택적 문자열. *correl-id*가 종종 요청을 응답과 연관시키는 요청 및 응답 시나리오에 지정됩니다. 지정되지 않으면 상관 ID가 지정되지 않습니다. *correl-id*의 최대 크기는 24바이트입니다.

*correl-id*가 지정되면, 일치하는 상관 ID를 갖는 모든 메시지는 큐에서 리턴되고 제거됩니다. *correl-id*가 지정되지 않으면 큐 헤드의 메시지가 리턴됩니다.

### *num-rows*

함수에서 리턴될 최대 메시지 수를 포함하는 양의 정수

*num-rows*가 지정되면 *num-rows* 메시지의 최대값이 리턴됩니다. *num-rows*가 지정되지 않으면 사용 가능한 모든 메시지가 리턴됩니다.

## 권한 부여

MQRECEIVEALL 테이블 함수에 대한 EXECUTE 특권

### 예:

예 1: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐에서 모든 메시지를 수신합니다. 메시지와 모든 메타데이터가 테이블로 리턴됩니다.

```
SELECT * FROM table (MQRECEIVEALL()) AS T
```

예 2: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 MYSERVICE 서비스에서 지정된 큐 헤드에서 모든 메시지를 수신합니다. MSG 및 CORRELID 컬럼만 리턴됩니다.

```
SELECT T.MSG, T.CORRELID FROM table (MQRECEIVEALL('MYSERVICE')) AS T
```

예 3: 이 예에서는 "MYPOLICY" 규정을 사용하여 "MYSERVICE" 서비스에서 지정된 큐 헤드의 모든 메시지를 수신합니다. '1234'의 CORRELID를 갖는 메시지만 리턴됩니다. MSG 및 CORRELID 컬럼만 리턴됩니다.

```
SELECT T.MSG, T.CORRELID FROM table  
(MQRECEIVEALL('MYSERVICE','MYPOLICY','1234')) AS T
```

예 4: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐 헤드에서 처음 10개의 메시지를 수신합니다. 모든 컬럼이 리턴됩니다.

```
SELECT * FROM table (MQRECEIVEALL(10)) AS T
```



## 리턴되는 정보

표 116. MQRECEIVEALL 테이블 함수가 리턴한 정보

컬럼 이름	데이터 유형	설명
MSG	VARCHAR(32000)	MQSeries 메시지의 내용을 포함합니다.
CORRELID	VARCHAR(24)	메시지를 식별하기 위한 상관 ID를 포함합니다. 이 ID를 사용하여 큐에서 메시지를 선택할 수 있습니다. 요청 및 응답 시나리오의 경우, 상관 ID는 특정 요청에 대해 응답을 연관할 수 있게 합니다.
TOPIC	VARCHAR(40)	가능한 경우, 메시지를 발행한 주제를 포함합니다.
QNAME	VARCHAR(48)	메시지를 수신한 큐 이름을 포함합니다.
MSGID	CHAR(24)	이 메시지에 지정된 고유 MQSeries ID를 포함합니다.
MSGFORMAT	VARCHAR(8)	MQSeries에 의해 정의된 대로 메시지의 형식을 포함합니다. 일반적인 문자열은 MQSTR 형식입니다.

## MQRECEIVEALLCLOB

MQRECEIVEALLCLOB 테이블 함수는 *service-policy* 서비스 규정 품질을 사용하여 *receive-service*에서 지정한 MQSeries 위치로부터 메시지와 메시지 메타데이터가 포함된 테이블을 리턴합니다. 이 조작을 수행하면 *receive-service*와 연관된 큐에서 메시지가 제거됩니다.

### 구문

```

▶▶ MQRECEIVEALLCLOB ( (
    receive-service
    , service-policy
    , correl-id
)
    num-rows
)
  
```

스키마는 DB2MQ입니다.

### 테이블 함수 매개변수

*receive-service*

메시지가 수신되는 논리적 MQSeries 목적지를 포함하는 문자열. 지정되는 경우,

*receive-service*는 DB2MQ.MQSERVICE 테이블에 정의된 서비스 지점을 참조해야 합니다. 서비스 지점은 메시지를 보내거나 수신하는 논리적 끝점입니다. 서비스 지점 정의는 MQSeries 큐 관리 프로그램 및 큐의 이름을 포함합니다. *receive-service*가 지정되지 않으면, DB2.DEFAULT.SERVICE가 사용됩니다. *receive-service*의 최대 크기는 48바이트입니다.

#### *service-policy*

해당 메시지의 처리에 사용되는 MQSeries 서비스 규정을 포함하는 문자열. 지정되는 경우, *service-policy*는 DB2MQ.MQPOLICY 테이블에 정의된 규정을 참조합니다. 서비스 규정은 이 메시징 조작에 적용해야 하는 서비스 품질 옵션 세트를 정의합니다. 이들 옵션에는 메시지 우선순위와 메시지 지속성이 들어 있습니다. *service-policy*가 지정되지 않으면, 디폴트 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48바이트입니다.

#### *correl-id*

이 메시지와 연관된 상관 ID를 포함하는 선택적 문자열. *correl-id*가 종종 요청을 응답과 연관시키는 요청 및 응답 시나리오에 지정됩니다. 지정되지 않으면 상관 ID가 지정되지 않습니다. *correl-id*의 최대 크기는 24바이트입니다.

*correl-id*가 지정되면 일치하는 상관 ID를 갖는 메시지만 리턴됩니다. *correl-id*가 지정되지 않으면 큐 헤드의 메시지가 리턴됩니다.

#### *num-rows*

함수에서 리턴될 최대 메시지 수를 포함하는 양의 정수

*num-rows*가 지정되면 *num-rows* 메시지의 최대값이 리턴됩니다. *num-rows*가 지정되지 않으면 사용 가능한 모든 메시지가 리턴됩니다.

## 권한 부여

MQRECEIVEALLCLOB 테이블 함수에 대한 EXECUTE 특권

### 예:

예 1: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐에서 모든 메시지를 수신합니다. 메시지와 모든 메타데이터가 테이블로 리턴됩니다.

```
SELECT * FROM table (MQRECEIVEALLCLOB()) AS T
```

예 2: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 MYSERVICE 서비스에서 지정된 큐 헤드에서 모든 메시지를 수신합니다. MSG 및 CORRELID 컬럼만 리턴됩니다.

```
SELECT T.MSG, T.CORRELID  
FROM table (MQRECEIVEALLCLOB('MYSERVICE')) AS T
```

예 3: 이 예에서는 "MYPOLICY" 규정을 사용하여 "MYSERVICE" 서비스에서 지정된 큐 헤드의 모든 메시지를 수신합니다. '1234'의 CORRELID를 갖는 메시지만 리턴됩니다. MSG 및 CORRELID 컬럼만 리턴됩니다.

```
SELECT T.MSG, T.CORRELID
FROM table (MQRECEIVEALLCLOB('MYSERVICE','MYPOLICY','1234')) AS T
```

예 4: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐 헤드에서 처음 10개의 메시지를 수신합니다. 모든 컬럼이 리턴됩니다.

```
SELECT * FROM table (MQRECEIVEALLCLOB(10)) AS T
```

## 리턴되는 정보

표 117. MQRECEIVEALLCLOB 테이블 함수가 리턴한 정보

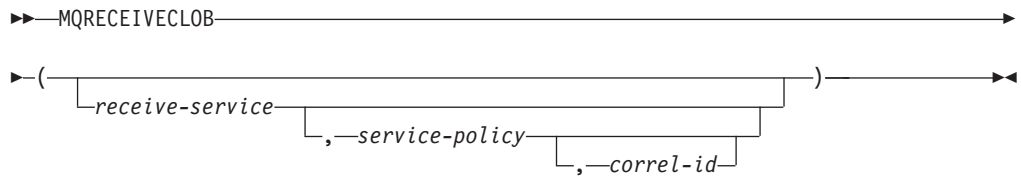
컬럼 이름	데이터 유형	설명
MSG	CLOB(1M)	MQSeries 메시지의 내용을 포함합니다.
CORRELID	VARCHAR(24)	메시지를 식별하기 위한 상관 ID를 포함합니다. 이 ID를 사용하여 큐에서 메시지를 선택할 수 있습니다. 요청 및 응답 시나리오의 경우, 상관 ID는 특정 요청에 대해 응답을 연관할 수 있게 합니다.
TOPIC	VARCHAR(40)	가능한 경우, 메시지를 발행한 주제를 포함합니다.
QNAME	VARCHAR(48)	메시지를 수신한 큐 이름을 포함합니다.
MSGID	CHAR(24)	이 메시지에 지정된 고유 MQSeries ID를 포함합니다.
MSGFORMAT	VARCHAR(8)	MQSeries에 의해 정의된 대로 메시지의 형식을 포함합니다. 일반적인 문자열은 MQSTR 형식입니다.

## MQRECEIVECLOB

MQRECEIVECLOB 함수는 *service-policy* 서비스 규정 품질을 사용하여 *receive-service*에서 지정된 MQSeries 위치로부터 메시지를 리턴합니다. 이 조작을 수행하면 *receive-service*와 연관된 큐에서 메시지가 제거됩니다. *correl-id*가 지정되면 일치하는 상관 ID를 갖는 첫 번째 메시지가 리턴됩니다. *correl-id*가 지정되지 않으면 큐의 헤드에 있는 메시지가 리턴됩니다.

결과의 데이터 유형은 CLOB(1M)입니다. 리턴할 수 있는 메시지가 없으면 결과는 널(NULL) 값이 됩니다.

## 구문



스키마는 DB2MQ입니다.

## 함수 매개변수

### *receive-service*

메시지가 수신되는 논리적 MQSeries 목적지를 포함하는 문자열. 지정되는 경우, *receive-service*는 DB2MQ.MQSERVICE 테이블에 정의된 서비스 지점을 참조해야 합니다. 서비스 지점은 메시지를 보내거나 수신하는 논리적 끝점입니다. 서비스 지점 정의는 MQSeries 큐 관리 프로그램 및 큐의 이름을 포함합니다. *receive-service*가 지정되지 않으면, DB2.DEFAULT.SERVICE가 사용됩니다. *receive-service*의 최대 크기는 48바이트입니다.

### *service-policy*

해당 메시지의 처리에 사용할 MQSeries 서비스 규정을 포함하는 문자열. 지정되는 경우, *service-policy*는 DB2MQ.MQPOLICY 테이블에 정의된 규정을 참조해야 합니다. 서비스 규정은 이 메시징 조각에 적용되어야 하는 서비스 품질 옵션 세트를 정의합니다. 이들 옵션에는 메시지 우선순위와 메시지 지속성이 들어 있습니다. *service-policy*가 지정되지 않으면, 디폴트 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48바이트입니다.

### *correl-id*

이 메시지와 연관된 선택적 상관 ID를 포함하는 문자열. *correl-id*가 종종 요청을 응답과 연관시키는 요청 및 응답 시나리오에 지정됩니다. 지정되지 않으면 상관 ID가 사용되지 않습니다. *correl-id*의 최대 크기는 24바이트입니다.

## 예:

예 1: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 디폴트 서비스(DB2.DEFAULT.SERVICE)에서 지정된 큐 헤드의 메시지를 수신합니다.

```
VALUES MQRECEIVECLOB()
```

예 2: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 "MYSERVICE" 서비스에서 지정된 큐 헤드의 메시지를 수신합니다.

```
VALUES MQRECEIVECLOB('MYSERVICE')
```

예 3: 이 예에서는 "MYPOLICY" 규정을 사용하여 "MYSERVICE" 서비스에서 지정되는 큐 헤드의 메시지를 수신합니다.

```
VALUES MQRECEIVECLOB('MYSERVICE','MYPOLICY')
```

예 4: 이 예에서는 "MYPOLICY" 규정을 사용하여 "MYSERVICE" 서비스에서 지정된 큐 헤드에서 '1234'와 일치하는 상관 ID를 갖는 첫 번째 메시지를 수신합니다.

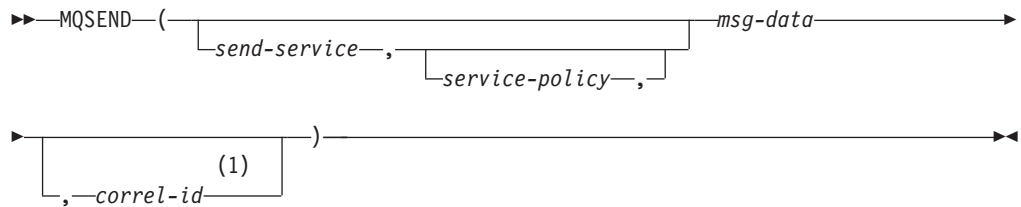
```
VALUES MQRECEIVECLOB('MYSERVICE',MYPOLICY','1234')
```

## MQSEND

MQSEND 함수는 *service-policy*에서 정의된 서비스 규정 품질을 사용하여 *msg-data*에 포함된 데이터를 *send-service*에서 지정된 MQSeries 위치로 보냅니다. 선택적 사용자 정의 메시지 상관 ID는 *correl-id*를 사용하여 정의할 수 있습니다.

결과 데이터 유형은 VARCHAR(1)입니다. 함수의 결과는 성공할 경우 '1' 또는 실패할 경우 '0'입니다.

### 구문



### 주:

1 *service* 및 *policy*가 먼저 지정되어야 *correl-id*를 지정할 수 있습니다.

스키마는 비트랜잭션 메시지 큐 함수의 경우 DB2MQ이고, 1단계 커밋 트랜잭션 MQ 함수의 경우 DB2MQ1C입니다.

### 함수 매개변수

#### *msg-data*

MQSeries를 통해 보낼 데이터를 포함하는 문자열 표현식. VARCHAR 문자열 표현식의 최대 크기는 32 000바이트이고 CLOB 문자열 표현식의 최대 크기는 1MB입니다.

#### *send-service*

메시지를 보낼 논리적 MQSeries 목적지를 포함하는 문자열. 지정되는 경우, *send-service*는 DB2MQ.MQSERVICE 테이블에 정의된 서비스 지점을 참조합니다. 서비스 지점은 메시지를 보내거나 수신하는 논리적 끝점입니다. 서비스 지정은 MQSeries 큐 관리 프로그램 및 큐의 이름을 포함합니다. *send-service*가 지정되지 않으면, DB2.DEFAULT.SERVICE 값이 사용됩니다. *send-service*의 최대 크기는 48바이트입니다.

### *service-policy*

해당 메시지의 처리에 사용되는 MQSeries 서비스 규정을 포함하는 문자열. 지정되는 경우, *service-policy*는 DB2MQ.MQPOLICY 테이블에 정의된 서비스 규정을 참조해야 합니다. 서비스 규정은 이 메시징 조작에 적용되어야 하는 서비스 품질 옵션 세트를 정의합니다. 이들 옵션에는 메시지 우선순위와 메시지 지속성이 들어 있습니다. *service-policy*가 지정되지 않으면, 디폴트값 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48바이트입니다.

### *correl-id*

이 메시지와 연관된 상관 ID를 포함하는 선택적 문자열. *correl-id*가 종종 요청을 응답과 연관시키는 요청 및 응답 시나리오에 지정됩니다. 지정되지 않으면 상관 ID를 보내지 않습니다. *correl-id*의 최대 크기는 24바이트입니다.

### 예:

예 1: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 상관 ID 없이 디폴트 서비스(DB2.DEFAULT.SERVICE)에 문자열 "Testing 123"을 보냅니다.

```
VALUES MQSEND('Testing 123')
```

예 2: 이 예에서는 "MYPOLICY" 규정을 사용하여 상관 ID 없이 "MYSERVICE" 서비스에 문자열 "Testing 345"를 보냅니다.

```
VALUES MQSEND('MYSERVICE','MYPOLICY','Testing 345')
```

예 3: 이 예에서는 "MYPOLICY" 규정을 사용하여 상관 ID "TEST3"과 함께 "MYSERVICE" 서비스에 "Testing 678" 문자열을 보냅니다.

```
VALUES MQSEND('MYSERVICE','MYPOLICY','Testing 678','TEST3')
```

예 4: 이 예에서는 디폴트 규정(DB2.DEFAULT.POLICY)을 사용하여 상관 ID 없이 "MYSERVICE" 서비스에 문자열 "Testing 901"을 보냅니다.

```
VALUES MQSEND('MYSERVICE','Testing 901')
```

---

## MQSUBSCRIBE

MQSUBSCRIBE 함수는 지정된 주제에 대해 발행된 MQSeries 메시지에 관심사항을 등록하는 데 사용됩니다. 이 함수를 성공적으로 실행하면 발행 및 서브스크라이브 서버가 주제와 일치하는 메시지를 *subscriber-service*에서 정의된 서비스 지점으로 전달합니다. *subscriber-service*는 지정된 주제와 일치하는 메시지에 대한 논리적 목적지를 지정합니다. *topic*과 일치하는 메시지가 *subscriber-service*에서 정의된 큐에 위치하며 MQREAD, MQRECEIVE, MQREADALL 또는 MQRECEIVEALL에 대한 후속 호출을 통해 읽거나 수신할 수 있습니다. 자세한 내용은 <http://www.ibm.com/software/MQSeries>를 방문하십시오.

결과의 데이터 유형은 VARCHAR(1)입니다. 함수의 결과는 성공할 경우 '1' 또는 실패할 경우 '0'입니다.

## 구문

```
MQSUBSCRIBE ( ( subscriber-service , service-policy ) topic )
```

스키마는 비트랜잭션 메시지 큐 함수의 경우 DB2MQ이고, 1단계 커밋 트랜잭션 MQ 함수의 경우 DB2MQ1C입니다.

## 함수 매개변수

### *subscriber-service*

*topic*과 일치하는 메시지를 보낼 논리적 MQSeries 서브스크립션 지점을 포함하는 문자열. 지정되는 경우, *subscriber-service*는 퍼블리셔 서비스 유형 값으로 'S'를 가진 DB2MQ.MQPUBSUB 테이블에 정의된 서브스크라이버 서비스 지점을 참조해야 합니다. *subscriber-service*가 지정되지 않으면, DB2.DEFAULT.SUBSCRIBER가 대신 사용됩니다. *subscriber-service*의 최대 크기는 48바이트입니다.

### *service-policy*

메시지의 처리에 사용할 MQSeries 서비스 규정을 포함하는 문자열. 지정되는 경우, *service-policy*는 DB2MQ.MQPOLICY 테이블에 정의된 규정을 참조해야 합니다. 서비스 규정은 이 메시징 조작에 적용되어야 하는 서비스 품질 옵션 세트를 정의합니다. 이들 옵션에는 메시지 우선순위와 메시지 지속성이 들어 있습니다. *service-policy*가 지정되지 않으면, 디폴트 DB2.DEFAULT.POLICY가 대신 사용됩니다. *service-policy*의 최대 크기는 48바이트입니다.

### *topic*

수신할 메시지의 유형을 정의하는 문자열. 지정된 주제로 발행된 메시지만이 이 서브스크립션에서 수신됩니다. 다중 서브스크립션이 공존할 수 있습니다. *topic*의 최대 크기는 40바이트입니다. 한 문자열에 여러 주제를 지정할 수 있습니다(최대 40 바이트). 각 주제는 콜론으로 분리되어야 합니다. 예를 들어, "t1:t2:the third topic"은 메시지가 t1, t2 및 "the third topic"의 세 주제와 모두 연관되어 있음을 나타냅니다.

## 예:

예 1: 이 예에서는 "Weather" 주제를 포함하는 메시지 관련사항을 등록합니다. 디폴트 *subscriber-service*(DB2.DEFAULT.SUBSCRIBER)가 서브스크라이버로 등록되고 디폴트 *service-policy*(DB2.DEFAULT.POLICY)가 서비스 품질을 지정합니다.

```
VALUES MQSUBSCRIBE('Weather')
```



예 2: 이 예에서는 "Stocks"를 포함하는 메시지 관련사항을 등록하는 서브스크라이버를 나타냅니다. 서브스크라이버는 "BASIC-POLICY" 규정을 갖는 "PORTFOLIO-UPDATES"로 등록됩니다.

```
VALUES MQSUBSCRIBE('PORTFOLIO-UPDATES','BASIC-POLICY','Stocks')
```

## MQUNSUBSCRIBE

MQUNSUBSCRIBE 함수는 기존 메시지 서브스크립션을 등록 해제하는 데 사용됩니다. *subscriber-service*, *service-policy* 및 *topic*이 취소할 서브스크립션을 식별하는 데 사용됩니다. 이 함수를 성공적으로 실행하면 발행 및 서브스크라이브 서버가 정의된 서브스크립션을 제거합니다. 지정된 *topic*을 갖는 메시지를 더 이상 *subscriber-service*에서 정의된 논리적 목적지로 보내지 않습니다. 자세한 내용은 <http://www.ibm.com/software/MQSeries>를 방문하십시오.

결과 데이터 유형은 VARCHAR(1)입니다. 함수의 결과는 성공할 경우 '1' 또는 실패할 경우 '0'입니다.

### 구문

```
MQUNSUBSCRIBE(
  (
    subscriber-service,
    service-policy,
    topic
  )
)
```

스키마는 비트랜잭션 메시지 큐 함수의 경우 DB2MQ이고, 1단계 커밋 트랜잭션 MQ 함수의 경우 DB2MQ1C입니다.

### 함수 매개변수

#### *subscriber-service*

지정되는 경우, *subscriber-service*는 퍼블리셔 서비스 유형 값으로 'S'를 가진 DB2MQ.MQPUBSUB 테이블에 정의된 서브스크라이버 서비스 지점을 참조해야 합니다. *subscriber-service*가 지정되지 않으면, DB2.DEFAULT.SUBSCRIBER가 대신 사용됩니다. *subscriber-service*의 최대 크기는 48바이트입니다.

#### *service-policy*

지정되는 경우, *service-policy*는 DB2MQ.MQPOLICY 테이블에 정의된 규정을 참조해야 합니다. 서비스 규정은 이 메시징 조작에 적용되어야 하는 서비스 품질 옵션 세트를 정의합니다. *service-policy*가 지정되지 않으면, 디폴트 DB2.DEFAULT.POLICY가 사용됩니다. *service-policy*의 최대 크기는 48바이트입니다.



*topic*

수신되지 않을 메시지의 주제를 지정하는 문자열. *topic*의 최대 크기는 40바이트입니다. 한 문자열에 여러 주제를 지정할 수 있습니다(최대 40바이트). 각 주제는 콜론으로 분리되어야 합니다. 예를 들어, "t1:t2:the third topic"은 메시지가 t1, t2 및 "the third topic"의 세 주제와 모두 연관되어 있음을 나타냅니다.

**예:**

예 1: 이 예에서는 "Weather" 주제를 포함하는 메시지 관련사항을 취소합니다. 디폴트 subscriber-service(DB2.DEFAULT.SUBSCRIBER)가 서브스크라이버 취소자로 등록되고 디폴트 service-policy(DB2.DEFAULT.POLICY)가 서비스 품질을 지정합니다.

```
VALUES MQUNSUBSCRIBE('Weather')
```

예 2: 이 예에서는 "Stocks"를 포함하는 메시지 관련사항을 취소하는 서브스크라이버를 나타냅니다. 서브스크라이버는 "BASIC-POLICY" 규정을 갖는 "PORTFOLIO-UPDATES"로 등록됩니다.

```
VALUES MQUNSUBSCRIBE('PORTFOLIO-UPDATES', 'BASIC-POLICY', 'Stocks')
```



---

## 제 13 장 보안 루틴 및 뷰

---

### AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID

AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID 테이블 함수는 데이터베이스 구성 파일에 있거나 그룹 또는 역할을 통해 권한 부여 ID에 직접 또는 간접적으로 부여된 권한 부여 ID에 의해 보유된 모든 권한을 리턴합니다.

#### 구문

▶▶—AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID—(—*authid*—,—*authidtype*—)————▶▶

스키마는 SYSPROC입니다.

#### 테이블 함수 매개변수

##### *authid*

조회 중인 권한 부여 ID를 지정하는 VARCHAR

(128) 유형의 입력 인수. 권한 부여 ID는 사용자, 그룹 또는 역할입니다. *authid*가 NULL 또는 비어 있는 문자열인 경우, 비어 있는 결과 테이블이 리턴됩니다.

##### *authidtype*

쿼리 중인 권한 부여 ID 유형을 지정하는 VARCHAR(1) 유형의 입력 인수.

*authidtype*가 존재하지 않고 NULL 또는 비어 있는 문자열인 경우, 비어 있는 결과 테이블이 리턴됩니다. *authidtype*의 가능한 값은 다음과 같습니다.

- G: 그룹
- R: 역할
- U: 사용자

#### 권한 부여

AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID 함수에 대한 EXECUTE 특권

#### 리턴되는 정보

표 118. AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
AUTHORITY	VARCHAR (128)	권한 부여 ID에 의해 보유된 권한

표 118. AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
D_USER	CHAR(1)	<p><i>authidtype</i>이 사용자(U)일 때 <i>authid</i>에 직접 부여된 권한. <i>authidtype</i>이 그룹(G) 또는 역할(R)이면 값을 적용할 수 없습니다(*).</p> <ul style="list-style-type: none"> <li>• N = 보유되지 않음</li> <li>• Y= 보유됨</li> <li>• * = 적용되지 않음</li> </ul>
D_GROUP	CHAR(1)	<p><i>authidtype</i>이 그룹(G)일 때 <i>authid</i>에 직접 부여된 권한 또는 <i>authidtype</i>이 사용자(U)일 때 <i>authid</i>가 속한 그룹에 직접 부여된 권한. <i>authidtype</i>이 역할(R)이면 값을 적용할 수 없습니다(*).</p> <ul style="list-style-type: none"> <li>• N = 보유되지 않음</li> <li>• Y= 보유됨</li> <li>• * = 적용되지 않음</li> </ul>
D_PUBLIC	CHAR(1)	<p><i>authidtype</i>이 사용자(U) 또는 그룹(G)일 때 PUBLIC이라고 하는 <i>authid</i>에 직접 부여된 권한. <i>authidtype</i>이 역할(R)이면 값을 적용할 수 없습니다(*).</p> <ul style="list-style-type: none"> <li>• N = 보유되지 않음</li> <li>• Y= 보유됨</li> <li>• * = 적용되지 않음</li> </ul>
ROLE_USER	CHAR(1)	<p><i>authidtype</i>이 사용자(U)일 때 <i>authid</i>가 부여된 역할에 직접 부여된 권한. <i>authidtype</i>이 그룹(G) 또는 역할(R)이면 값을 적용할 수 없습니다(*). 역할은 역할 계층 구조의 일부가 될 수 있습니다.</p> <ul style="list-style-type: none"> <li>• N = 보유되지 않음</li> <li>• Y= 보유됨</li> <li>• * = 적용되지 않음</li> </ul>
ROLE_GROUP	CHAR(1)	<p><i>authidtype</i>이 그룹(G)일 때 <i>authid</i>에 부여된 역할에 직접 부여된 권한. <i>authidtype</i>이 사용자(U)이거나 역할(R)이면 값을 적용할 수 없습니다(*). 역할은 역할 계층 구조의 일부가 될 수 있습니다.</p> <ul style="list-style-type: none"> <li>• N = 보유되지 않음</li> <li>• Y= 보유됨</li> <li>• * = 적용되지 않음</li> </ul>
ROLE_PUBLIC	CHAR(1)	<p><i>authidtype</i>이 사용자(U) 또는 그룹(G)일 때 PUBLIC이라고 하는 <i>authid</i>에 부여된 역할에 직접 부여된 권한. <i>authidtype</i>이 역할(R)이면 값을 적용할 수 없습니다(*). 역할은 역할 계층 구조의 일부가 될 수 있습니다.</p> <ul style="list-style-type: none"> <li>• N = 보유되지 않음</li> <li>• Y= 보유됨</li> <li>• * = 적용되지 않음</li> </ul>
D_ROLE	CHAR(1)	<p>역할에 부여된 권한 또는 역할에 부여된 역할. <i>authidtype</i>이 사용자(U)이거나 그룹(G)이면 값을 적용할 수 없습니다(*). 역할은 역할 계층 구조의 일부가 될 수 있습니다.</p> <ul style="list-style-type: none"> <li>• N = 보유되지 않음</li> <li>• Y= 보유됨</li> <li>• * = 적용되지 않음</li> </ul>

예 :

특수 그룹 PUBLIC을 통해서 기본으로 BIND, CONNECT, CREATETAB 및 IMPLICIT\_SCHEMA 특권을 보류하는 사용자 ALICE를 고려합니다. ALICE는 SYSADM, SYSCTRL 및 SYSMANT의 시스템 권한이 있는 그룹 ADMIN1의 구성원입니다. 또한 DBADM 권한이 있는 그룹 ADMIN2의 구성원이기도 합니다. 그리고 ALICE에게는 DBADM 및 SECADM 데이터베이스 권한이 있습니다. 역할 R1이 ALICE에게 부여되었습니다. LOAD 권한이 역할 R1에 부여되었습니다. 역할 R2는 그룹 ADMIN1에 부여되었습니다. CREATE\_NOT\_FENCED\_ROUTINE 권한은 역할 R2에 부여되었습니다.

예 1: 사용자 ALICE가 직접 자신에게 부여하거나 그룹 PUBLIC 또는 역할을 통해 간접적으로 부여한 모든 권한을 검색합니다.

```
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	*	*	*	*	*
SYSMANT	*	Y	*	*	*	*	*
SYSMON	*	N	*	*	*	*	*
WLMADM	N	N	N	N	N	N	*

예 2: 그룹 ADMIN1이 직접 자신에게 부여하거나 그룹 PUBLIC 또는 역할을 통해 간접적으로 부여한 모든 권한을 검색합니다.

```
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	*	*	*	*	*
SYSMAINT	*	Y	*	*	*	*	*
SYSMON	*	N	*	*	*	*	*
WLMADM	*	N	*	*	N	*	*

예 3: 특수 그룹 PUBLIC이 직접 자신에게 부여하거나 역할을 통해 간접적으로 부여한 모든 권한을 검색합니다.

```
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	*

예 4: 역할 R1이 직접 자신에게 부여하거나 역할을 통해 간접적으로 부여한 모든 권한을 검색합니다. 이 경우 역할 R2도 역할 R1에 권한이 부여된 것으로 간주합니다.

```
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

## 사용법 참고

AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID 테이블 함수의 출력은 *authidtype*에 따라 달라집니다. 예를 들어 USER의 *authidtype*는 다음과 같은 방법을 통해 *authid*가 보유한 모든 권한을 리턴합니다.

- *authid*에 직접 권한 부여됨
- *authid*가 속한 그룹(또는 그룹에 부여된 역할)에 권한 부여됨
- *authid*에 부여된 역할(또는 역할에 부여된 역할)에 권한 부여됨
- PUBLIC(또는 PUBLIC에 부여된 역할)에 권한 부여됨

---

## AUTH\_LIST\_GROUPS\_FOR\_AUTHID 테이블 함수 - 주어진 권한 부여 ID에 대한 그룹 멤버십 목록 검색

AUTH\_LIST\_GROUPS\_FOR\_AUTHID 테이블 함수는 주어진 권한 부여 ID가 구성 원인 그룹의 목록을 리턴합니다.

### 구문

►►—AUTH\_LIST\_GROUPS\_FOR\_AUTHID—(—*authid*—)—————►►

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *authid*

조회 중인 권한 부여 ID를 지정하는 VARCHAR(128) 유형의 입력 인수. 권한 부여 ID는 사용자만 표시할 수 있습니다. *authid*가 존재하지 않는 경우, NULL 또는 비어 있는 문자열인 경우, 비어 있는 결과 테이블이 리턴됩니다.

### 권한 부여

AUTH\_LIST\_GROUPS\_FOR\_AUTHID 테이블 함수에 대한 EXECUTE 특권.

### 예 :

AMY가 속한 모든 그룹을 검색하십시오.

```
SELECT * FROM TABLE(SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID('AMY')) AS T
```

다음은 이 쿼리에 대한 출력의 예입니다.

```
GROUP-----...-----
BUILD
PDXDB2
```

2개의 레코드가 선택되었습니다.

### 사용 시 참고사항

리턴된 그룹 정보는 다음 이유로 예상된 것과 다를 수 있습니다.

- Windows 활성 디렉토리 환경에서 데이터베이스 관리 프로그램은 다음과 같이 수행합니다.
  - 로컬 그룹 내에서 중첩된 그룹 레벨 중 하나를 지원합니다(로컬 그룹 내 도메인 로컬 그룹의 중첩은 제외). 예를 들어, *authid*가 전역 그룹 G1에 속하고, G1이 로컬 그룹 L1에 속하는 경우, 로컬 그룹 L1은 *authid*에 대한 그룹으로 리턴됩니다. 단, *authid*가 도메인 로컬 그룹 DL1에 속하고, DL1이 로컬 그룹 L1에 속하는 경우, *authid*에 대한 그룹은 리턴되지 않습니다.
  - 전역 그룹의 중첩을 지원하지 않습니다. 예를 들어, *authid*가 전역 G2에 속하고, G2가 전역 G3에 속하는 경우, G2만 *authid*에 대한 그룹으로 리턴됩니다.
- 레지스트리 변수 DB2\_GRP\_LOOKUP은 Windows 보안 메커니즘이 사용자가 속한 그룹을 열거하는 데 사용됨을 지정합니다.
- 특정 도메인에 속한 권한 부여 ID의 경우, 도메인이 *authid*의 일부로 지정되지 않고, 로컬 및 도메인 *authid* 모두가 동일한 이름으로 존재하면, 로컬 권한 부여 ID에 대한 그룹이 리턴됩니다.

## 리턴된 정보

표 119. AUTH\_LIST\_GROUPS\_FOR\_AUTHID 테이블 함수로 리턴된 정보

컬럼 이름	데이터 유형	설명
GROUP	VARCHAR (128)	권한 부여 ID가 속한 그룹.

## AUTH\_LIST\_ROLES\_FOR\_AUTHID 함수 - 역할 목록 리턴

AUTH\_LIST\_ROLES\_FOR\_AUTHID 함수는 제공된 권한 부여 ID가 구성원인 역할 목록을 리턴합니다.

### 구문

▶—AUTH\_LIST\_ROLES\_FOR\_AUTHID—(—*authid*—,—*authidtype*—)—▶

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *authid*

조회 중인 권한 부여 ID를 지정하는 VARCHAR (128) 유형의 입력 인수. 권한 부여 ID는 사용자, 그룹 또는 역할입니다. *authid*가 NULL 또는 비어 있는 문자열인 경우, 비어 있는 결과 테이블이 리턴됩니다.

#### *authidtype*

쿼리 중인 권한 부여 ID 유형을 지정하는 VARCHAR(1) 유형의 입력 인수.



`authidtype`가 존재하지 않고 NULL 또는 비어 있는 문자열인 경우, 비어 있는 결과 테이블이 리턴됩니다. `authidtype`의 가능한 값은 다음과 같습니다.

- G: 그룹
- R: 역할
- U: 사용자

## 권한 부여

`AUTH_LIST_ROLES_FOR_AUTHID` 함수에 대한 EXECUTE 특권

## 리턴되는 정보

표 120. `AUTH_LIST_ROLES_FOR_AUTHID`에 대한 결과 세트

컬럼 이름	데이터 유형	설명
GRANTOR	VARCHAR (128)	역할의 권한 준 사용자
GRANTORTYPE	CHAR(1)	권한 부여자 유형: • U = 권한 준 사용자가 개별 사용자임
GRANTEE	VARCHAR (128)	역할을 부여한 사용자
GRANTEETYPE	CHAR(1)	권한 받은 사용자 유형: • G = 권한 받은 사용자가 그룹임 • R = 권한 받은 사용자가 역할임 • U = 권한 받은 사용자가 사용자임
ROLENAME	VARCHAR (128)	그룹 또는 다른 역할을 통해 직접 또는 간접적으로 권한 부여 ID에 부여된 역할의 이름
CREATE_TIME	TIMESTAMP	역할이 작성된 시간
ADMIN	CHAR(1)	역할 부여, 역할 호출 또는 역할에 주식 작성 특권: • N = 보유되지 않음 • Y = 보유됨

예 :

INTERN 역할을 DOCTOR 역할에 DOCTOR 역할을 SPECIALIST 역할에 부여하는 것을 고려한 다음 SPECIALIST 역할을 사용자 ALICE에게 부여합니다. ALICE는 HOSPITAL 그룹에 속하며 EMPLOYEE이 역할이 HOSPITAL 그룹에 부여됩니다. ALICE는 특수 PUBLIC 그룹에도 속하며 PATIENTS 역할이 PUBLIC에 부여됩니다.

예 1: 사용자 ALICE에게 부여된 모든 역할을 검색합니다.

```
SELECT GRANTOR, GRANTORTYPE, GRANTEE, GRANTEETYPE, ROLENAME,
       CREATE_TIME, ADMIN
FROM TABLE (SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID ('ALICE', 'U') ) AS T
```

다음은 이 쿼리에 대한 출력 예입니다.

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

5개의 레코드가 선택되었습니다.

예 2: 그룹 HOSPITAL에 부여된 모든 역할을 검색합니다.

```
SELECT GRANTOR, GRANTORTYPE, GRANTEE, GRANTEETYPE, ROLENAME,
       CREATE_TIME, ADMIN
FROM TABLE (SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID ('HOSPITAL', 'G') ) AS T
```

다음은 이 쿼리로부터의 출력 예입니다.

GRANTOR	GRANTORTYPE	GRANTEE	GRANTEETYPE	ROLENAME	CREATE_TIME	ADMIN
ZURBIE	U	HOSPITAL	G	EMPLOYEE	2006-08-01-15.10.14.277576	N

1개의 레코드가 선택되었습니다.

예 3: 역할 SPECIALIST에 부여된 모든 역할을 검색합니다.

```
SELECT GRANTOR, GRANTORTYPE, GRANTEE, GRANTEETYPE, ROLENAME,
       CREATE_TIME, ADMIN
FROM TABLE (SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID ('SPECIALIST', 'R') ) AS T
```

다음은 이 쿼리에 대한 출력의 예입니다.

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

2개의 레코드가 선택되었습니다.

예 4: 그룹 PUBLIC에 부여된 모든 역할을 검색합니다.

```
SELECT GRANTOR, GRANTORTYPE, GRANTEE, GRANTEETYPE, ROLENAME,
       CREATE_TIME, ADMIN
FROM TABLE (SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID ('PUBLIC', 'G') ) AS T
```

다음은 이 쿼리로부터의 출력 예입니다.

GRANTOR	GRANTORTYPE	GRANTEE	GRANTEETYPE	ROLENAME	CREATE_TIME	ADMIN
ZURBIE	U	PUBLIC	G	PATIENTS	2006-08-01-15.10.18.878609	N

1개의 레코드가 선택되었습니다.

## 사용 시 참고사항

AUTH\_LIST\_ROLES\_FOR\_AUTHID 테이블 함수의 출력은 AUTHIDTYPE에 따라 다음과 같이 달라집니다.

- 사용자의 경우 다른 역할, 사용자가 속한 그룹(또는 PUBLIC)을 통해 직접 또는 간접적으로 사용자에게 부여된 역할을 리턴합니다.
- 그룹의 경우 다른 역할을 통해 직접 또는 간접적으로 그룹에 부여된 역할을 리턴합니다.

- 역할의 경우 다른 역할을 통해 직접 또는 간접적으로 역할에 부여된 역할을 리턴합니다.

## AUTHORIZATIONIDS 관리 뷰 - 권한 부여 ID 및 유형 검색

AUTHORIZATIONIDS 관리 뷰는 특권이나 권한을 부여하는 권한 부여 ID 목록과 현재 연결된 데이터베이스의 시스템 카탈로그에 정의된 모든 권한 부여 ID에 대한 해당 유형을 리턴합니다. 특권이나 권한이 그룹이나 역할로 부여된 경우 해당 그룹이나 역할 이름만 리턴됩니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- AUTHORIZATIONIDS 관리 뷰에 대한 SELECT 특권
- AUTHORIZATIONIDS 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

예 :

특권이나 권한이 부여된 모든 권한 부여 ID와 해당 유형을 검색합니다.

```
SELECT * FROM SYSIBMADM.AUTHORIZATIONIDS
```

다음은 이 쿼리에 대한 출력 예입니다.

AUTHID	AUTHIDTYPE
PUBLIC	G
JESSICAE	U
DOCTOR	R

3개의 레코드가 선택되었습니다.

### 리턴된 정보

표 121. AUTHORIZATIONIDS 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
AUTHID	VARCHAR (128)	명시적으로 특권이나 권한이 부여된 권한 부여 ID입니다.
AUTHIDTYPE	CHAR(1)	권한 부여 ID 유형: <ul style="list-style-type: none"> <li>• U: 사용자</li> <li>• R: 역할</li> <li>• G: 그룹</li> </ul>

## OBJECTOWNERS 관리 뷰 - 오브젝트 소유권 정보 검색

OBJECTOWNERS 관리 뷰는 현재 연결된 데이터베이스에서 시스템 카탈로그에 정의된 오브젝트를 소유한 USER 유형의 모든 권한 부여 ID에 대한 모든 오브젝트 소유권 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- OBJECTOWNERS 관리 뷰에 대한 SELECT 특권
- OBJECTOWNERS 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

### 예 :

오브젝트 스키마 'THERESAX'에 대한 모든 오브젝트 소유권 정보를 검색합니다.

```
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'
```

다음은 이 쿼리에 대한 출력 예입니다.

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	SQL0F00	NULLID	PACKAGE
...				
THERESAX	U	HI_OBJ_UNIQ	SYSTOOLS	TABLE CONSTRAINT

257개의 레코드가 선택되었습니다.

### 리턴된 정보

표 122. OBJECTOWNERS 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
OWNER	VARCHAR (128)	이 오브젝트를 소유하는 권한 부여 ID
OWNERTYPE	VARCHAR(1)	권한 부여 ID 유형: • U: 사용자
OBJECTNAME	VARCHAR (128)	데이터베이스 오브젝트 이름

표 122. OBJECTOWNERS 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
OBJECTSCHEMA	VARCHAR (128)	데이터베이스 오브젝트 스키마
OBJECTTYPE	VARCHAR(24)	데이터베이스 오브젝트 유형

## PRIVILEGES 관리 뷰 - 특권 정보 검색

PRIVILEGES 관리 뷰는 현재 연결된 데이터베이스에서 시스템 카탈로그에 정의된 모든 권한 부여 ID에 대한 모든 명시적 특권을 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- PRIVILEGES 관리 뷰에 대한 SELECT 특권
- PRIVILEGES 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

### 예 :

모든 권한 부여 ID에 대해 오브젝트 이름, 스키마 및 유형과 같이 권한 부여된 특권을 검색합니다.

```
SELECT AUTHID, PRIVILEGE, OBJECTNAME, OBJECTSCHEMA, OBJECTTYPE
FROM SYSIBMADM.PRIVILEGES
```

다음은 이 쿼리에 대한 출력 예입니다.

AUTHID	PRIVILEGE	OBJECTNAME	OBJECTSCHEMA	OBJECTTYPE
JESSICAE	EXECUTE	SQL0F00	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		POLICY_UNQ	SYSTOOLS	INDEX

PUBLIC	CREATEIN		NULLID	SCHEMA
PUBLIC	UPDATE	COLUMNS	SYSSTAT	VIEW
PUBLIC	UPDATE	COLGROUPS	SYSSTAT	VIEW
...				

## 리턴된 정보

표 123. PRIVILEGES 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
AUTHID	VARCHAR (128)	이 특권에 명시적으로 부여된 권한 부여 ID입니다.
AUTHIDTYPE	CHAR(1)	권한 부여 ID 유형: <ul style="list-style-type: none"> <li>• U: 사용자</li> <li>• R: 역할</li> <li>• G: 그룹</li> </ul>
PRIVILEGE	VARCHAR(11)	이 권한 부여 ID에 명시적으로 부여된 특권입니다.
GRANTABLE	VARCHAR(1)	특권이 부여 가능한지를 표시합니다. <ul style="list-style-type: none"> <li>• Y: 부여 가능</li> <li>• N: 부여 불가능</li> </ul>
OBJECTNAME	VARCHAR (128)	데이터베이스 오브젝트 이름
OBJECTSCHEMA	VARCHAR (128)	데이터베이스 오브젝트 스키마
OBJECTTYPE	VARCHAR(24)	데이터베이스 오브젝트 유형

---

## 제 14 장 스냅샷 루틴 및 뷰

---

### APPL\_PERFORMANCE 관리 뷰 - 응용프로그램에 대해 선택된 행의 백분율 검색

APPL\_PERFORMANCE 관리 뷰에는 응용프로그램이 선택한 행의 백분율에 대한 정보가 표시됩니다. 리턴된 정보는 현재 연결된 데이터베이스에 대한 모든 데이터베이스 파티션에 해당합니다. 이 뷰를 사용하여 큰 테이블 스캔을 수행하는 응용프로그램을 검색하거나 이후에 문제를 발생시킬 수도 있는 쿼리를 검색할 수 있습니다.

스키마는 SYSIBMADM입니다.

#### 권한 부여

다음 권한 중 하나가 필요합니다.

- APPL\_PERFORMANCE 관리 뷰에 대한 SELECT 특권
- APPL\_PERFORMANCE 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

#### 예 :

응용프로그램 성능에 대한 보고서도 검색합니다.

```
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
```

다음은 이 쿼리에 대한 출력 예입니다.

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	...
			...

3개의 레코드가 선택되었습니다.

...

이 쿼리의 출력(계속).

```

... AGENT_ID          PERCENT_ROWS_SELECTED DBPARTITIONNUM
... -----
...              67                -                1
...              68                -                0
...              67                57.14           0
...

```

## 리턴된 정보

표 124. APPL\_PERFORMANCE 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
AUTHID	VARCHAR (128)	auth_id - 권한 부여 ID
APPL_NAME	VARCHAR (256)	appl_name - 응용프로그램 이름
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
PERCENT_ROWS_SELECTED	DECIMAL(5,2)	응용프로그램에 실제로 리턴된 디스크에서 읽은 행에 대한 퍼센트입니다. 주: 표시된 백분율은 100.00퍼센트를 넘지 않습니다.
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## APPLICATIONS 관리 뷰 - 연결된 데이터베이스 응용프로그램 정보 검색

APPLICATIONS 관리 뷰는 연결된 데이터베이스 응용프로그램에 대한 정보를 리턴합니다. 뷰는 LIST APPLICATIONS SHOW DETAIL CLP 명령에 대한 SQL 인터페이스이지만 현재 연결된 데이터베이스에만 적용됩니다. 해당 정보는 SNAPAPPL\_INFO 관리 뷰를 바탕으로 합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- APPLICATIONS 관리 뷰에 대한 SELECT 특권
- APPLICATIONS 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON



- SYSCTRL
- SYSMOINT
- SYSADM

예 :

예 1: 단일 파티션 데이터베이스 SAMPLE의 모든 활성 응용프로그램에 대한 정보를 나열합니다.

```
SELECT AGENT_ID, SUBSTR(APPL_NAME,1,10) AS APPL_NAME, AUTHID,
       APPL_STATUS FROM SYSIBMADM.APPLICATIONS WHERE DB_NAME = 'SAMPLE'
```

다음은 이 쿼리에 대한 출력 예입니다.

AGENT_ID	APPL_NAME	AUTHID	APPL_STATUS
23	db2bp.exe	JESSICAE	UOWEXEC

1개의 레코드가 선택되었습니다.

예 2: 다중 파티션 데이터베이스 SAMPLE에 대해 데이터베이스 파티션 0의 응용프로그램별로 에이전트 수를 표시합니다.

```
SELECT SUBSTR(APPL_NAME, 1, 10) AS APPL_NAME, COUNT(*) AS NUM
       FROM SYSIBMADM.APPLICATIONS WHERE DBPARTITIONNUM = 0
       AND DB_NAME = 'SAMPLE' GROUP BY APPL_NAME
```

다음은 이 쿼리에 대한 출력 예입니다.

APPL_NAME	NUM
db2bp.exe	3
javaw.exe	1

2개의 레코드가 선택되었습니다.

### 사용 시 참고사항

뷰는 CLP에서 사용할 수 있는 GLOBAL 구문을 지원하지 않습니다. 그렇지만 모든 데이터베이스 파티션의 데이터가 뷰에서 리턴되기 때문에 SQL 집계 함수를 사용하여 집계를 수행할 수 있습니다.

### 리턴된 정보

표 125. APPLICATIONS 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
CLIENT_DB_ALIAS	VARCHAR (128)	client_db_alias - 응용프로그램에서 사용되는 데이터베이스 별명
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름

표 125. APPLICATIONS 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
APPL_NAME	VARCHAR (256)	appl_name - 응용프로그램 이름
AUTHID	VARCHAR (128)	auth_id - 권한 부여 ID
APPL_ID	VARCHAR (128)	appl_id - 응용프로그램 ID

표 125. APPLICATIONS 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
APPL_STATUS	VARCHAR(22)	<p>appl_status - 응용프로그램 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다.</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>
STATUS_CHANGE_TIME	TIMESTAMP	status_change_time - 응용프로그램 상태 변경 시간
SEQUENCE_NO	VARCHAR(4)	sequence_no - 시퀀스 번호
CLIENT_PRDID	VARCHAR(128)	client_prdid - 클라이언트 제품/버전 ID
CLIENT_PID	BIGINT	client_pid - 클라이언트 프로세스 ID

표 125. APPLICATIONS 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
CLIENT_PLATFORM	VARCHAR(12)	<p>client_platform - 클라이언트 운영 플랫폼. 이 인터페이스는 sqlmon.h로 정의한 것을 기초로 하는 텍스트 ID를 리턴하며, 이는 다음 중 하나입니다.</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> <li>• WINDOWS95</li> </ul>

표 125. APPLICATIONS 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
CLIENT_PROTOCOL	VARCHAR(10)	client_protocol - 클라이언트 통신 프로토콜. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴합니다. <ul style="list-style-type: none"> <li>• CPIC</li> <li>• LOCAL</li> <li>• NETBIOS</li> <li>• NPIPE</li> <li>• TCPIP ( DB2 UDB용)</li> <li>• TCPIP4</li> <li>• TCPIP6</li> </ul>
CLIENT_NNAME	VARCHAR (128)	client_nname 모니터 요소는 더 이상 사용되지 않습니다. 리턴된 값은 올바른 값이 아닙니다.
COORD_NODE_NUM	SMALLINT	coord_node - 코디네이팅 노드
COORD_AGENT_PID	BIGINT	coord_agent_pid - 코디네이터 에이전트
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 연관 에이전트 수
TPMON_CLIENT_USERID	VARCHAR (256)	tpmon_client_userid - TP 모니터 클라이언트 사용자 ID
TPMON_CLIENT_WKSTN	VARCHAR (256)	tpmon_client_wkstn - TP 모니터 클라이언트 워크스테이션 이름
TPMON_CLIENT_APP	VARCHAR (256)	tpmon_client_app - TP 모니터 클라이언트 응용프로그램 이름
TPMON_ACC_STR	VARCHAR(200)	tpmon_acc_str - TP 모니터 클라이언트 어카운팅 문자열
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## BP\_HITRATIO 관리 뷰 - 버퍼 풀 사용 비율 정보 검색

BP\_HITRATIO 관리 뷰는 현재 연결된 데이터베이스에 대한 전체 버퍼 풀 및 모든 데이터베이스 파티션에 대한 전체 사용 비율, 데이터 사용 비율, XDA 사용 비율 및 인덱스 사용 비율을 포함한 버퍼 풀 사용 비율을 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- BP\_HITRATIO 관리 뷰에 대한 SELECT 특권
- BP\_HITRATIO 관리 뷰에 대한 CONTROL 특권

- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

연결된 데이터베이스의 모든 버퍼 풀에 대한 보고서를 검색합니다.

```
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
```

다음은 이 쿼리에 대한 출력 예입니다.

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	-	-	...

이 쿼리의 출력(계속).

...	INDEX_HIT_RATIO_PERCENT	XDA_HIT_RATIO_PERCENT	DBPARTITIONNUM
...	43.20	-	0
...	-	-	0
...	-	-	0
...	-	-	0
...	-	-	0

## 사용 시 참고사항

논리적 읽기에 대한 실제 읽기 비율을 사용하여 버퍼 풀에 대한 사용 비율을 알 수 있습니다. 사용 비율이 낮을수록 대용량 오퍼레이션의 캐시된 버퍼 풀보다는 디스크에서 더 많은 데이터를 읽습니다.

## 리턴된 정보

표 126. BP\_HITRATIO 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	보고서 요청시 시간소인
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름

표 126. BP\_HITRATIO 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
BP_NAME	VARCHAR (128)	bp_name - 버퍼 풀 이름
TOTAL_LOGICAL_READS	BIGINT	버퍼 풀에서 전체 논리적 읽기입니다(인덱스, XDA 및 데이터).
TOTAL_PHYSICAL_READS	BIGINT	버퍼 풀에서 전체 실제 읽기입니다(인덱스, XDA 및 데이터).
TOTAL_HIT_RATIO_PERCENT	DECIMAL(5,2)	전체 사용 비율입니다(인덱스, XDA 및 데이터 읽기).
DATA_LOGICAL_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
DATA_PHYSICAL_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
DATA_HIT_RATIO_PERCENT	DECIMAL(5,2)	데이터 사용 비율입니다.
INDEX_LOGICAL_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
INDEX_PHYSICAL_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
INDEX_HIT_RATIO_PERCENT	DECIMAL(5,2)	인덱스 사용 비율입니다.
XDA_LOGICAL_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
XDA_PHYSICAL_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
XDA_HIT_RATIO_PERCENT	DECIMAL(5,2)	보조 기억장치 오브젝트 사용 비율입니다.
DBPARTITIONNUM	SMALLINT	행의 데이터가 검색되는 데이터베이스 파티션입니다.

## BP\_READ\_IO 관리 뷰 - 버퍼 풀 읽기 성능 정보 검색

BP\_READ\_IO 관리 뷰는 버퍼 풀 읽기 성능 정보를 리턴합니다. 이 뷰를 사용하여 각 각의 버퍼 풀에서 프리페치가 미치는 효과를 알 수 있습니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- BP\_READ\_IO 관리 뷰에 대한 SELECT 특권
- BP\_READ\_IO 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

현재 연결된 데이터베이스의 모든 파티션에서 모든 버퍼 풀에 대한 총 실제 읽기 횟수 및 평균 읽기 시간을 검색합니다.

```
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
```

다음은 이 쿼리에 대한 출력의 예입니다.

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

10개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 127. BP\_READ\_IO 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	보고서가 생성된 날짜 및 시간
BP_NAME	VARCHAR (128)	bp_name - 버퍼 풀 이름
TOTAL_PHYSICAL_READS	BIGINT	총 실제 읽기 횟수
AVERAGE_READ_TIME_MS	BIGINT	평균 읽기 시간(밀리초)
TOTAL_ASYNC_READS	BIGINT	총 비동기 읽기 횟수
AVERAGE_ASYNC_READ_TIME_MS	BIGINT	평균 비동기 읽기 시간(밀리초)
TOTAL_SYNC_READS	BIGINT	총 동기 읽기 횟수
AVERAGE_SYNC_READ_TIME_MS	BIGINT	평균 동기 읽기 시간(밀리초)



표 127. BP\_READ\_IO 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
PERCENT_SYNC_READS	DECIMAL(5,2)	프리페칭 없이 동기적으로 읽은 페이지의 백분율. 많은 응용프로그램이 프리페칭 없이 동기적으로 데이터를 읽을 경우 시스템이 최적으로 설정되지 않았을 수도 있습니다.
ASYNC_NOT_READ_PERCENT	DECIMAL(5,2)	디스크에서 비동기적으로 읽었으나, 쿼리에서 액세스하지 않은 페이지의 백분율. 많은 페이지를 디스크에서 버퍼 풀로 비동기적으로 읽었으나 쿼리에서 해당 페이지를 결코 액세스하지 않은 경우, 프리페칭이 성능을 저하시킬 수도 있습니다.
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## BP\_WRITE\_IO 관리 뷰 - 버퍼 풀 쓰기 성능 정보 검색

BP\_WRITE\_IO 관리 뷰는 버퍼 풀당 버퍼 풀 쓰기 성능 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- BP\_WRITE\_IO 관리 뷰에 대한 SELECT 특권
- BP\_WRITE\_IO 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

현재 연결된 데이터베이스의 모든 데이터베이스 파티션에서 모든 버퍼 풀에 대한 총 쓰기 횟수 및 평균 쓰기 시간을 검색합니다.

```
SELECT SUBSTR(BP_NAME, 1, 15) AS BP_NAME, TOTAL_WRITES,
       AVERAGE_WRITE_TIME_MS, DBPARTITIONNUM
FROM SYSIBMADM.BP_WRITE_IO ORDER BY DBPARTITIONNUM
```

다음은 이 쿼리에 대한 출력의 예입니다.

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

11개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 128. BP\_WRITE\_IO 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	보고서를 생성한 날짜 및 시간
BP_NAME	VARCHAR (128)	bp_name - 버퍼 풀 이름
TOTAL_WRITES	BIGINT	총 쓰기 횟수
AVERAGE_WRITE_TIME_MS	BIGINT	평균 쓰기 시간(밀리초)
TOTAL_ASYNC_WRITES	BIGINT	총 비동기 쓰기 횟수
PERCENT_WRITES_ASYNC	BIGINT	비동기 쓰기 퍼센트
AVERAGE_ASYNC_WRITE_TIME_MS	BIGINT	평균 비동기 쓰기 시간(밀리초)
TOTAL_SYNC_WRITES	BIGINT	총 동기 쓰기 횟수
AVERAGE_SYNC_WRITE_TIME_MS	BIGINT	평균 동기 쓰기 시간(밀리초)
DBPARTITIONNUM	SMALLINT	행에 대한 데이터의 데이터베이스 파티션 검색.

## CONTAINER\_UTILIZATION 관리 뷰 - 테이블 스페이스 컨테이너 및 활용 정보 검색

CONTAINER\_UTILIZATION 관리 뷰는 테이블 스페이스 컨테이너 및 사용도 비율에 대한 정보를 리턴합니다. 이 뷰는 LIST TABLESPACE CONTAINERS CLP 명령에 대한 SQL 인터페이스입니다. 해당 정보는 SNAPCONTAINER 관리 뷰를 기반으로 합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- CONTAINER\_UTILIZATION 관리 뷰에 대한 SELECT 특권
- CONTAINER\_UTILIZATION 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

### 예 :

총 페이지 수 및 사용 가능 페이지와 액세스 가능성 상태를 포함하여 연결된 파티션 데이터베이스에 있는 모든 테이블 스페이스 컨테이너 목록을 검색합니다.

```
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
```

다음은 이 쿼리에 대한 출력 예입니다.

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	...

5개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```

... 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

```

## 리턴되는 정보

파일 시스템 정보를 리턴하려면 데이터베이스 관리 프로그램에서 BUFFERPOOL 스냅샷 모니터 스위치를 사용 가능하게 해야 합니다.

표 129. CONTAINER\_UTILIZATION 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TBSP_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
CONTAINER_NAME	VARCHAR(256)	container_name - 컨테이너 이름
CONTAINER_ID	BIGINT	container_id - 컨테이너 ID
CONTAINER_TYPE	VARCHAR(16)	container_type - 컨테이너 유형  이는 sqlutil.h의 정의 중 하나를 기반으로 하는 텍스트 ID로 다음 중 하나입니다. <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 - 컨테이너의 전체 페이지 수
USABLE_PAGES	BIGINT	container_usable_pages - 컨테이너에서 사용할 수 있는 페이지 수
ACCESSIBLE	SMALLINT	container_accessible - 컨테이너의 액세스 가능성
STRIPE_SET	BIGINT	container_stripe_set - 스트라이프 세트
FS_ID	VARCHAR(22)	fs_id - 고유 파일 시스템 ID 번호
FS_TOTAL_SIZE_KB	BIGINT	fs_total_size - 파일 시스템의 전체 크기. 이 인터페이스는 KB 단위로 값을 리턴합니다.
FS_USED_SIZE_KB	BIGINT	fs_used_size - 파일 시스템에서 사용된 스페이스 크기. 이 인터페이스는 KB 단위로 값을 리턴합니다.
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## LOCKS\_HELD 관리 뷰 - 보유한 잠금에 대한 정보 검색

LOCKS\_HELD 관리 뷰는 현재 보유한 잠금에 대한 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- LOCKS\_HELD 관리 뷰에 대한 SELECT 특권
- LOCKS\_HELD 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### 예 :

예 1: 데이터베이스 SAMPLE의 각 테이블에서 보유하고 있는 전체 잠금 수가 표시됩니다.

```
SELECT TABSCHEMA, TABNAME, COUNT(*) AS NUMBER_OF_LOCKS_HELD
FROM SYSIBMADM.LOCKS_HELD WHERE DB_NAME = 'SAMPLE'
GROUP BY DBPARTITIONNUM, TABSCHEMA, TABNAME
```

다음은 이 쿼리에 대한 출력 예입니다.

TABSCHEMA	TABNAME	NUMBER_OF_LOCKS_HELD
JESSICAE	EMPLOYEE	5
JESSICAE	EMP_RESUME	1
JESSICAE	ORG	3

예 2: 현재 연결된 데이터베이스인 SAMPLE에서 에스컬레이션되지 않은 모든 잠금이 표시됩니다.

```
SELECT AGENT_ID, TABSCHEMA, TABNAME, LOCK_OBJECT_TYPE, LOCK_MODE,
LOCK_STATUS FROM SYSIBMADM.LOCKS_HELD WHERE LOCK_ESCALATION = 0
AND DBPARTITIONNUM = 0
```

다음은 이 쿼리에 대한 출력 예입니다.

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

예 3: 에이전트 ID가 310인 응용프로그램이 현재 보유하고 있는 잠금에 대한 잠금 정보를 표시합니다.

```
SELECT TABSCHEMA, TABNAME, LOCK_OBJECT_TYPE, LOCK_MODE, LOCK_STATUS,
       LOCK_ESCALATION FROM SYSIBMADM.LOCKS_HELD WHERE AGENT_ID = 310
```

다음은 이 쿼리에 대한 출력 예입니다.

```

      TABSCHEMA      TABNAME      LOCK_OBJECT_TYPE  LOCK_MODE  LOCK_STATUS
-----
JESSICAE      EMP_RESUME  TABLE_LOCK      S          GRNT
JESSICAE      EMPLOYEE   ROW_LOCK         S          GRNT

```

## 리턴된 정보

표 130. LOCKS\_HELD 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	보고서가 생성된 날짜 및 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
APPL_NAME	VARCHAR (256)	appl_name - 응용프로그램 이름
AUTHID	VARCHAR (128)	auth_id - 권한 부여 ID
TBSP_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
TABSCHEMA	VARCHAR (128)	table_schema - 테이블 스키마 이름
TABNAME	VARCHAR (128)	table_name - 테이블 이름
TAB_FILE_ID	BIGINT	table_file_id - 테이블 파일 ID

표 130. LOCKS\_HELD 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_OBJECT_TYPE	VARCHAR(18)	lock_object_type - 대기하는 잠금 오브젝트 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 잠금 이름

표 130. LOCKS\_HELD 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_MODE	VARCHAR(10)	lock_mode - 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (잠금이 없는 경우)</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 - 잠금 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• CONV</li> <li>• GRNT</li> </ul>
LOCK_ESCALATION	SMALLINT	lock_escalation - 잠금 에스컬레이션
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## LOCKWAITS 관리 뷰 - 현재 잠금 대기 정보 검색

LOCKWAITS 관리 뷰는 잠금 확보 대기 중인 응용프로그램 대신 작동 중인 DB2 에이전트에 대한 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- LOCKWAITS 관리 뷰에 대한 SELECT 특권
- LOCKWAITS 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON



- SYSCTRL
- SYSMANT
- SYSADM

**예:**

예 1: 에이전트 ID가 89인 응용프로그램에 대한 전체 잠금 대기 정보를 표시합니다.

```
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
```

다음은 이 쿼리에 대한 출력 예입니다.

TABSCHEMA	TABNAME	LOCK_OBJECT_TYPE	LOCK_MODE	...
JESSICAE	T1	ROW_LOCK	X	...

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	LOCK_MODE_REQUESTED	AGENT_ID_HOLDING_LK
...	NS	7

예 2: 데이터베이스 SAMPLE의 테이블별로 전체 미해결 잠금 요청 수를 표시합니다. 요청 수별로 출력을 정렬하여 가장 높은 경쟁률의 테이블을 식별할 수 있습니다.

```
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
```

다음은 이 쿼리에 대한 출력 예입니다.

TABSCHEMA	TABNAME	NUM_OF_LOCK_REQUESTS_WAITING	DBPARTITIONNUM
JESSICAE	T3	2	0
JESSICAE	T1	1	0
JESSICAE	T2	1	0

3개의 레코드가 선택되었습니다.

**리턴된 정보**

표 131. LOCKWAITS 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	보고서가 생성된 날짜 및 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름

표 131. LOCKWAITS 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
APPL_NAME	VARCHAR (256)	appl_name - 응용프로그램 이름
AUTHID	VARCHAR (128)	auth_id - 권한 부여 ID
TBSP_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
TABSCHEMA	VARCHAR (128)	table_schema - 테이블 스키마 이름
TABNAME	VARCHAR (128)	table_name - 테이블 이름
SUBSECTION_NUMBER	BIGINT	ss_number - 서브섹션 번호

표 131. LOCKWAITS 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_OBJECT_TYPE	VARCHAR(18)	lock_object_type - 대기하는 잠금 오브젝트 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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_WAIT_START_TIME	TIMESTAMP	lock_wait_start_time - 잠금 대기 시작 시간소인
LOCK_NAME	VARCHAR(32)	lock_name - 잠금 이름

표 131. LOCKWAITS 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_MODE	VARCHAR(10)	lock_mode - 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (잠금이 없는 경우)</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 - 요청된 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (잠금이 없는 경우)</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 - 잠금을 보유한 에이전트 ID
APPL_ID_HOLDING_LK	VARCHAR(128)	appl_id_holding_lk - 잠금을 보유한 응용프로그램 ID
LOCK_ESCALATION	SMALLINT	lock_escalation - 잠금 에스컬레이션
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

---

## LOG\_UTILIZATION 관리 뷰 - 로그 사용도 정보 검색

LOG\_UTILIZATION 관리 뷰는 현재 연결된 데이터베이스에 대한 로그 사용도 정보를 리턴합니다. 각각의 데이터베이스 파티션에 대하여 단일 행이 리턴됩니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- LOG\_UTILIZATION 관리 뷰에 대한 SELECT 특권
- LOG\_UTILIZATION 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

### 예 :

현재 연결된 데이터베이스, SAMPLE에 대한 로그 사용도를 나열합니다.

```
SELECT * FROM SYSIBMADM.LOG_UTILIZATION
```

다음은 이 쿼리에 대한 출력의 예입니다.

```
DB_NAME    ... LOG_UTILIZATION_PERCENT TOTAL_LOG_USED_KB    ...
-----    ... -----
SAMPLE     ...                9.75                1989 ...
                                         ...
```

1개의 레코드가 선택되었습니다.

...

이 쿼리의 출력(계속).

```
... TOTAL_LOG_AVAILABLE_KB TOTAL_LOG_USED_TOP_KB DBPARTITIONNUM
... -----
...                18411                1990                0
...
...
...
...
```

### 사용법 참고

무제한 로깅을 지원하도록 구성된 데이터베이스의 경우,

LOG\_UTILIZATION\_PERCENT 및 TOTAL\_LOG\_AVAILABLE\_KB가 널(NULL)입니다.

## 리턴되는 정보

표 132. LOG\_UTILIZATION 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
LOG_UTILIZATION_PERCENT	DECIMAL(5,2)	총 로그 스페이스의 사용도
TOTAL_LOG_USED_KB	BIGINT	total_log_used - 사용된 전체 로그 스페이스. 이 인터페이스는 KB 단위로 값을 리턴합니다.
TOTAL_LOG_AVAILABLE_KB	BIGINT	total_log_available - 사용 가능한 전체 로그. 이 인터페이스는 KB 단위로 값을 리턴합니다.
TOTAL_LOG_USED_TOP_KB	BIGINT	tot_log_used_top - 사용된 최대 전체 로그 스페이스. 이 인터페이스는 KB 단위로 값을 리턴합니다.
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

---

## LONG\_RUNNING\_SQL 관리 뷰

LONG\_RUNNING\_SQL 관리 뷰는 현재 연결된 데이터베이스의 최장기 실행 SQL문을 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- LONG\_RUNNING\_SQL 관리 뷰에 대한 SELECT 특권
- LONG\_RUNNING\_SQL 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

현재 연결된 데이터베이스의 장기 실행 SQL문에 대한 보고서를 검색합니다.

```
SELECT SUBSTR(STMT_TEXT, 1, 50) AS STMT_TEXT, AGENT_ID,
       ELAPSED_TIME_MIN, APPL_STATUS, DBPARTITIONNUM
FROM SYSIBMADM.LONG_RUNNING_SQL ORDER BY DBPARTITIONNUM
```

다음은 이 쿼리에 대한 출력의 예입니다.

```
STMT_TEXT                      AGENT_ID    ...
-----
select * from dbuser.employee   228 ...
select * from dbuser.employee   228 ...
select * from dbuser.employee   228 ...
...
...
...
...
...
```

3개의 레코드가 선택되었습니다.  
...

이 쿼리의 출력(계속).

```
... ELAPSED_TIME_MIN APPL_STATUS    DBPARTITIONNUM
... -----
...                2 UOWWAIT          0
...                0 CONNECTED        1
...                0 CONNECTED        2
```

## 사용법 참고

이 뷰를 사용하여 데이터베이스에서 장기 실행 SQL문을 식별할 수 있습니다. 현재 실행 중인 쿼리를 조사하여 최장기 실행 명령문과 쿼리의 현재 상태를 알 수 있습니다. 고유 ID로 에이전트 ID를 사용하여, SQL문을 포함하는 응용프로그램에 대한 조사를 수행할 수 있습니다. 장기 실행 중이며 잠금 대기 상태일 경우, LOCKWAITS 또는 LOCKS\_HELD 관리 뷰를 사용하여 보다 깊게 조사하고자 할 수도 있습니다. “waiting on User”의 경우, DB2 서버가 아무 것도 수행하고 있지 않지만 응용프로그램이 작업 다음 페이지 발행 또는 다음 SQL문 제출을 수행할 때까지 대기 중임을 의미합니다.

## 리턴되는 정보

표 133. LONG\_RUNNING\_SQL 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	보고서를 생성한 시간
ELAPSED_TIME_MIN	INTEGER	명령문의 경과 시간(분)
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
APPL_NAME	VARCHAR (256)	appl_name - 응용프로그램 이름

표 133. LONG\_RUNNING\_SQL 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
APPL_STATUS	VARCHAR(22)	<p>appl_status - 응용프로그램 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다.</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>
AUTHID	VARCHAR (128)	auth_id - 권한 부여 ID
INBOUND_COMM_ADDRESS	VARCHAR(32)	inbound_comm_address - 인바운드 통신 주소
STMT_TEXT	CLOB(16 M)	stmt_text - SQL문 텍스트
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.



## QUERY\_PREP\_COST 관리 뷰 - 명령문 준비 시간 정보 검색

QUERY\_PREP\_COST 관리 뷰는 명령문 목록을 명령문을 준비하는 데 필요한 시간에 대한 정보와 함께 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- QUERY\_PREP\_COST 관리 뷰에 대한 SELECT 특권
- SNAPAGENT 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAPDYN\_SQL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

### 예 :

준비에 소비된 시간의 백분율이 가장 높은 쿼리에 대한 보고서를 검색합니다.

```
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
```

다음은 이 쿼리에 대한 출력의 예입니다.

```
NUM_EXECUTIONS    AVERAGE_EXECUTION_TIME_S ...  
-----  
1                25 ...
```

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... PREP_TIME_PERCENT STMT_TEXT                DBPARTITIONNUM  
... -----  
...                0.0 select * from dbuser.employee                0
```

### 사용 시 참고사항

뷰에서 선택할 경우, ORDER BY절을 사용하여 준비 비용이 가장 큰 쿼리를 식별할 수 있습니다. 이 뷰를 조사하여 쿼리 실행 빈도와 각각의 해당 쿼리에 대한 평균 실행 시간을 알 수 있습니다. 쿼리를 컴파일하고 최적화하는 데 걸리는 시간이 거의 쿼리를 실행하는 데 걸리는 시간만큼 오래 걸릴 경우, 사용 중인 최적화 클래스를 조사하고자

할 수도 있습니다. 최적화 클래스를 낮추면 쿼리에서 최적화를 보다 빠르게 완료하므로 보다 빠르게 결과를 리턴할 수도 있습니다. 그러나 쿼리에서 준비하는 데 시간이 오래 걸리기는 하지만 다시 준비할 필요없이 여러 번 실행될 경우, 최적화 클래스는 문제가 되지 않을 수도 있습니다.

## 리턴되는 정보

표 134. QUERY\_PREP\_COST 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	보고서를 생성한 날짜 및 시간
NUM_EXECUTIONS	BIGINT	num_executions - 명령문 실행
AVERAGE_EXECUTION_TIME_S	BIGINT	평균 실행시간(초)
PREP_TIME_MS	BIGINT	prep_time_worst - 명령문의 가장 안좋은 준비 시간
PREP_TIME_PERCENT	DECIMAL(5,2)	준비에 소비된 실행시간 퍼센트
STMT_TEXT	CLOB(2 M)	stmt_text - SQL문 텍스트
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPAGENT 관리 뷰 및 SNAP\_GET\_AGENT 테이블 함수 - 에이전트 논리 데이터 그룹 응용프로그램 스냅샷 정보 검색

SNAPAGENT 관리 뷰 및 SNAP\_GET\_AGENT 테이블 함수는 응용프로그램 스냅샷에서 에이전트, 특히 에이전트 논리 데이터 그룹에 대한 정보를 리턴합니다.

### SNAPAGENT 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 에이전트 논리 데이터 그룹 응용프로그램 스냅샷 정보를 검색할 수 있습니다.

SNAPAGENT\_MEMORY\_POOL, SNAPAPPL, SNAPAPPL\_INFO, SNAPSTMT 및 SNAPSUBSECTION 관리 뷰와 함께 사용되는 경우, SNAPAGENT 관리 뷰는 GET SNAPSHOT FOR APPLICATIONS ON database-alias CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 521 페이지의 표 135의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPAGENT 관리 뷰에 대한 SELECT 특권

- SNAPAGENT 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_AGENT 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

**예 :**

에이전트 논리 데이터 그룹에서 현재 연결된 데이터베이스에 대한 모든 응용프로그램 스냅샷 정보를 검색합니다.

```
SELECT * FROM SYSIBMADM.SNAPAGENT
```

이 쿼리의 출력 예는 다음과 같습니다.

SNAPSHOT_TIMESTAMP	DB_NAME	AGENT_ID	...
2005-07-19-11.03.26.740423	SAMPLE	101	...
2005-07-19-11.03.26.740423	SAMPLE	49	...
			...

2개의 레코드가 선택되었습니다. ...

이 쿼리의 출력(계속).

...	AGENT_PID	LOCK_TIMEOUT_VAL	DBPARTITIONNUM
...	AGENT_PID	LOCK_TIMEOUT_VAL	DBPARTITIONNUM
...		11980	-1 0
...		15940	-1 0
...			
...			

## SNAP\_GET\_AGENT 테이블 함수

SNAP\_GET\_AGENT 테이블 함수는 SNAPAGENT 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_AGENT\_MEMORY\_POOL, SNAP\_GET\_APPL\_V95, SNAP\_GET\_APPL\_INFO\_V95, SNAP\_GET\_STMT 및 SNAP\_GET\_SUBSECTION 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_AGENT 테이블 함수는 GET

SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

리턴될 수 있는 정보의 완전한 목록은 521 페이지의 표 135의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_AGENT ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스와 동일한 인터페이스 내의 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_AGENT 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_AGENT 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

예 :

모든 활성 데이터베이스에서 모든 응용프로그램에 대한 모든 응용프로그램 스냅샷 정보를 검색합니다.

```
SELECT * FROM TABLE(SNAP_GET_AGENT(CAST(NULL AS VARCHAR(128)), -1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...

6개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	AGENT_PID	LOCK_TIMEOUT_VAL	DBPARTITIONNUM
...	7696	-1	0
...	8536	-1	0
...	6672	-1	0
...	2332	-1	0
...	8360	-1	0
...	6736	-1	0
...			

## 리턴되는 정보

표 135. SNAPAGENT 관리 뷰 및 SNAP\_GET\_AGENT 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
AGENT_PID	BIGINT	agent_pid - EDU(Engine Dispatchable Unit)
LOCK_TIMEOUT_VAL	BIGINT	lock_timeout_val - 잠금 시간종료(초)
DBPARTITIONNUM	SMALLINT	행에 대한 데이터의 데이터베이스 파티션 검색.

---

## SNAPAGENT\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수 – memory\_pool 논리 데이터 그룹 스냅샷 정보 검색

SNAPAGENT\_MEMORY\_POOL 관리 뷰 및

SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수는 에이전트 레벨에서 메모리 사용에 대한 정보를 리턴합니다.

### SNAPAGENT\_MEMORY\_POOL 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 에이전트 레벨 메모리 사용에 관하여 memory\_pool 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPAGENT, SNAPAPPL, SNAPAPPL\_INFO, SNAPSTMT 및 SNAPSUBSECTION 관리 뷰와 함께 사용되는 경우, SNAPAGENT\_MEMORY\_POOL 관리 뷰는 GET SNAPSHOT FOR APPLICATIONS ON database-alias CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 525 페이지의 표 136의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPAGENT\_MEMORY\_POOL 관리 뷰에 대한 SELECT 특권
- SNAPAGENT\_MEMORY\_POOL 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

메모리 풀 및 현재 크기 목록을 검색합니다.

```
SELECT AGENT_ID, POOL_ID, POOL_CUR_SIZE FROM SYSIBMADM.SNAPAGENT_MEMORY_POOL
```

이 쿼리의 출력 예는 다음과 같습니다.

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

9개의 레코드가 선택되었습니다.

## SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수

SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수는

SNAPAGENT\_MEMORY\_POOL 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_AGENT, SNAP\_GET\_APPL\_V95, SNAP\_GET\_APPL\_INFO\_V95, SNAP\_GET\_STMT 및 SNAP\_GET\_SUBSECTION 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수는 GET SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 525 페이지의 표 136의 내용을 참조하십시오.

## 구문

```

▶▶—SNAP_GET_AGENT_MEMORY_POOL—(—dbname—)————▶▶
                                   └──, dbpartitionnum──┘

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

*dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하

려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스와 동일한 인터페이스 내의 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### 예 :

모든 데이터베이스에 대한 메모리 풀 및 현재 크기 목록을 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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

18개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 136. SNAPAGENT\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
AGENT_PID	BIGINT	agent_pid - EDU(Engine Dispatchable Unit)

표 136. SNAPAGENT\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_ID	VARCHAR(14)	pool_id - 메모리 풀 ID. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 메모리 풀의 현재 크기
POOL_WATERMARK	BIGINT	pool_watermark - 메모리 풀 워터 마크 (water mark)
POOL_CONFIG_SIZE	BIGINT	pool_config_size - 메모리 풀의 구성된 크기
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수 - Retrieve appl\_info 논리 데이터 그룹 스냅샷 정보

SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수는 응용 프로그램 스냅샷에서 응용프로그램 특히, appl\_info 논리 데이터 그룹에 대한 정보를 리턴합니다.

## SNAPAPPL\_INFO 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 appl\_info 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPAGENT, SNAPAGENT\_MEMORY\_POOL, SNAPAPPL, SNAPSTMT 및 SNAPSUBSECTION 관리 뷰와 같이 사용하여 SNAPAPPL\_INFO 관리 뷰는 GET SNAPSHOT FOR APPLICATIONS ON database-alias CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 530 페이지의 표 137의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPAPPL\_INFO 관리 뷰에 대한 SELECT 특권
- SNAPAPPL\_INFO 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또한 다음 권한 중 하나도 필요합니다.

- SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또한 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSMAINT
- SYSCTRL
- SYSADM

### 예 :

현재 데이터베이스에 연결된 응용프로그램의 상태를 검색합니다.

```
SELECT AGENT_ID, SUBSTR(APPL_NAME,1,10) AS APPL_NAME, APPL_STATUS
FROM SYSIBMADM.SNAPAPPL_INFO
```

다음은 이 쿼리의 출력 예입니다.

AGENT_ID	APPL_NAME	APPL_STATUS
101	db2bp.exe	UOWEXEC
49	db2bp.exe	CONNECTED

2개의 레코드가 선택되었습니다.

## SNAP\_GET\_APPL\_INFO\_V95 테이블 함수

SNAP\_GET\_APPL\_INFO\_V95 테이블 함수는 SNAPAPPL\_INFO 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

SNAP\_GET\_AGENT, SNAP\_GET\_AGENT\_MEMORY\_POOL, SNAP\_GET\_APPL\_V95, SNAP\_GET\_STMT 및 SNAP\_GET\_SUBSECTION 테이블 함수와 함께 사용되는 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수는 GET SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

리턴 가능한 전체 정보 목록은 530 페이지의 표 137의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_APPL_INFO_V95 ( (dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우 -1이 *dbpartitionnum*에 대해 내재적으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신

데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대해 스냅샷을 작성합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예:

연결된 데이터베이스 파티션의 모든 응용프로그램에 대한 상태를 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

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

4개의 레코드가 선택되었습니다.

다음은 테이블 함수 결과에서 선택할 때 확보한 내용입니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

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)

```

3개의 레코드가 선택되었습니다.

## 리턴된 정보

표 137. SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)

표 137. SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
APPL_STATUS	VARCHAR(22)	<p>appl_status - 응용프로그램 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다.</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 - 응용프로그램에서 사용되는 코드 페이지 ID
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 연관 에이전트 수
COORD_NODE_NUM	SMALLINT	coord_node - 코디네이팅 노드

표 137. SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
AUTHORITY_LVL	VARCHAR(512)	<p>authority_bitmap - 사용자 권한 부여 레벨.</p> <p>이 sql.h 및 해당 소스에 정의된 데이터베이스 권한을 기반으로 하는 텍스트 ID를 리턴하고 권한(소스, ...) + 권한(소스, ...) + ... 형식을 사용합니다. 권한의 소스는 여러 가지 유형이 가능하며 USER, GROUP 또는 USER 및 GROUP 중 하나입니다.</p> <p>"권한"에 대한 가능한 값은 다음과 같습니다.</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>"소스"에 대한 가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• USER - 사용자에게 부여된 권한 또는 사용자에게 부여된 역할</li> <li>• GROUP - 사용자가 속하는 그룹에 부여된 권한 또는 사용자가 속하는 그룹에 부여된 역할</li> </ul>
CLIENT_PID	BIGINT	client_pid - 클라이언트 프로세스 ID
COORD_AGENT_PID	BIGINT	coord_agent_pid - 코디네이터 에이전트
STATUS_CHANGE_TIME	TIMESTAMP	status_change_time - 응용프로그램 상태 변경 시간



표 137. SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
CLIENT_PLATFORM	VARCHAR(12)	<p>client_platform - 클라이언트 운영 플랫폼. 이 인터페이스는 sqlmon.h의 정의를 기반으로 텍스트 ID를 리턴합니다.</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>

표 137. SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
CLIENT_PROTOCOL	VARCHAR(10)	client_protocol - 클라이언트 통신 프로토콜. 이 인터페이스는 sqlmon.h의 정의를 기반으로 텍스트 ID를 리턴합니다.  <ul style="list-style-type: none"> <li>• CPIC</li> <li>• LOCAL</li> <li>• NETBIOS</li> <li>• NPIPE</li> <li>• TCPIP ( DB2 UDB용)</li> <li>• TCPIP4</li> <li>• TCPIP6</li> </ul>
TERRITORY_CODE	SMALLINT	territory_code - 데이터베이스 지역 코드
APPL_NAME	VARCHAR (256)	appl_name - 응용프로그램 이름
APPL_ID	VARCHAR (128)	appl_id - 응용프로그램 ID
SEQUENCE_NO	VARCHAR(4)	sequence_no - 시퀀스 번호
PRIMARY_AUTH_ID	VARCHAR (128)	auth_id - 권한 부여 ID
SESSION_AUTH_ID	VARCHAR (128)	session_auth_id - 세션 권한 부여 ID
CLIENT_NNAME	VARCHAR (128)	client_nname 모니터 요소는 더 이상 사용되지 않습니다. 리턴된 값은 올바른 값이 아닙니다.
CLIENT_PRDID	VARCHAR (128)	client_prdid - 클라이언트 제품/버전 ID
INPUT_DB_ALIAS	VARCHAR (128)	input_db_alias - 데이터베이스 별명 입력
CLIENT_DB_ALIAS	VARCHAR (128)	client_db_alias - 응용프로그램에서 사용되는 데이터베이스 별명
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
DB_PATH	VARCHAR(1024)	db_path - 데이터베이스 경로
EXECUTION_ID	VARCHAR (128)	execution_id - 사용자 로그인 ID
CORR_TOKEN	VARCHAR (128)	corr_token - DRDA 상환 토큰
TPMON_CLIENT_USERID	VARCHAR (256)	tpmon_client_userid - TP 모니터 클라이언트 사용자 ID
TPMON_CLIENT_WKSTN	VARCHAR (256)	tpmon_client_wkstn - TP 모니터 클라이언트 워크스테이션 이름
TPMON_CLIENT_APP	VARCHAR (256)	tpmon_client_app - TP 모니터 클라이언트 응용프로그램 이름
TPMON_ACC_STR	VARCHAR(200)	tpmon_acc_str - TP 모니터 클라이언트 어카운팅 문자열
DBPARTITIONNUM	SMALLINT	행의 데이터가 검색되는 데이터베이스 파티션입니다.

표 137. SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
WORKLOAD_ID	INTEGER	현재 워크로드 ID입니다.
IS_SYSTEM_APPL	SMALLINT	IS_SYSTEM_APPL 값은 응용프로그램이 DB2 내부 시스템 응용프로그램인지 여부를 나타냅니다.  0인 경우 사용자 응용프로그램입니다.  1인 경우 시스템 응용프로그램입니다.  DB2 시스템 응용프로그램의 예로는 DB2 이벤트 모니터가 있습니다.  일반적으로 DB2 시스템 응용프로그램의 이름은 "db2"로 시작됩니다. 예: db2stmm, db2taskd.

## SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수 - appl 논리 데이터 그룹 스냅샷 정보 검색

『SNAPAPPL 관리 뷰』 및 536 페이지의 『SNAP\_GET\_APPL\_V95 테이블 함수』에서는 응용프로그램 스냅샷, 특히 appl 논리 데이터 그룹의 응용프로그램에 대한 정보를 리턴합니다.

### SNAPAPPL 관리 뷰

이 관리 뷰로 현재 연결된 데이터베이스에 대한 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPAGENT, SNAPAGENT\_MEMORY\_POOL, SNAPAPPL\_INFO, SNAPSTMT 및 SNAPSUBSECTION 관리 뷰와 함께 사용된 SNAPAPPL 관리 뷰는 GET SNAPSHOT FOR APPLICATIONS ON database-alias CLP 명령과 동일한 정보를 제공하나, 모든 데이터베이스 파티션에서 데이터를 검색합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 538 페이지의 표 138의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPAPPL 관리 뷰에 대한 SELECT 특권
- SNAPAPPL 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_APPL\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

연결된 데이터베이스의 각 응용프로그램을 읽고 쓴 행에 대한 세부사항을 검색하십시오.

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AGENT_ID, ROWS_READ, ROWS_WRITTEN
FROM SYSIBMADM.SNAPAPPL
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	AGENT_ID	ROWS_READ	ROWS_WRITTEN
SAMPLE		7	25
			0

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_APPL\_V95 테이블 함수

SNAP\_GET\_APPL\_V95 테이블 함수는 SNAPAPPL 관리 뷰와 동일한 정보를 리턴하나, 특정 데이터베이스 파티션의 특정 데이터베이스에 대한 정보, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션은 사용자가 검색할 수 있습니다.

SNAP\_GET\_AGENT, SNAP\_GET\_AGENT\_MEMORY\_POOL, SNAP\_GET\_APPL\_INFO\_V95, SNAP\_GET\_STMT 및 SNAP\_GET\_SUBSECTION 테이블 함수와 함께 사용될 때, SNAP\_GET\_APPL\_V95 테이블 함수는 GET SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동일한 정보를 제공하나, 모든 데이터베이스 파티션에서 데이터를 검색합니다.

리턴될 수 있는 정보의 완전한 목록은 538 페이지의 표 138의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_APPL_V95 ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우, *dbpartitionnum*에 -1이 내재적으로 설정됩니다. 이 입력 옵션이 사용되지 않으면 즉, *dbname*만 제공되면, 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스가 응용프로그램별 연결 및 사용 가능한 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 언제든지 작성될 수 있으며, 이는 데이터가 통용되지 않음을 의미한다는 것을 알아두십시오. 해당 스냅샷 API 요청 유형이 있는 파일이 존재하지 않으면, SNAP\_GET\_APPL\_V95 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 수에 대한 스냅샷을 취합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_APPL\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

모든 활성 데이터베이스에 대한 각 응용프로그램을 읽거나 작성하는 행에 대한 세부사항을 검색하십시오.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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

## 리턴된 정보

표 138. SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수로 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 취한 날짜 및 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
UOW_LOG_SPACE_USED	BIGINT	uow_log_space_used - 사용된 작업 단위(UOW) 로그 스페이스
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
INACT_STMTHIST_SZ	BIGINT	stmt_history_list_size - 명령문 실행기록 목록 크기
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수

표 138. SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리페치 페이지 수
LOCKS_HELD	BIGINT	locks_held - 보유된 잠금 수
LOCK_WAITS	BIGINT	lock_waits - 잠금 대기 수
LOCK_WAIT_TIME	BIGINT	lock_wait_time - 잠금 대기 시간
LOCK_ESCALS	BIGINT	lock_escalations - 잠금 에스컬레이션 수
X_LOCK_ESCALS	BIGINT	x_lock_escalations - 배타적 잠금 에스컬레이션 수
DEADLOCKS	BIGINT	deadlocks - 발견된 교착 상태 수
TOTAL_SORTS	BIGINT	total_sorts - 전체 정렬 수
TOTAL_SORT_TIME	BIGINT	total_sort_time - 총 정렬 시간
SORT_OVERFLOWS	BIGINT	sort_overflows - 정렬 오버플로우 수
COMMIT_SQL_STMTS	BIGINT	commit_sql_stmts - 시도된 명령문 커밋
ROLLBACK_SQL_STMTS	BIGINT	rollback_sql_stmts - 시도된 롤백 명령문
DYNAMIC_SQL_STMTS	BIGINT	dynamic_sql_stmts - 시도된 동적 SQL문
STATIC_SQL_STMTS	BIGINT	static_sql_stmts - 시도된 정적 SQL문
FAILED_SQL_STMTS	BIGINT	failed_sql_stmts - 실패한 명령문 조각

표 138. SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SELECT_SQL_STMTS	BIGINT	select_sql_stmts - 실행된 Select SQL문
DDL_SQL_STMTS	BIGINT	ddl_sql_stmts - DDL(Data Definition Language) SQL문
UID_SQL_STMTS	BIGINT	uid_sql_stmts - 실행된 UPDATE/INSERT/DELETE SQL문
INT_AUTO_REBINDS	BIGINT	int_auto_rebinds - 내부 자동 리바인드 수
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 삭제된 내부 행 수
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 갱신된 내부 행 수
INT_COMMITS	BIGINT	int_commits - 내부 커미트 수
INT_ROLLBACKS	BIGINT	int_rollback - 내부 롤백 수
INT_DEADLOCK_ROLLBACKS	BIGINT	int_deadlock_rollback - 교착 상태로 인한 내부 롤백 수
ROWS_DELETED	BIGINT	rows_deleted - 삭제된 행 수
ROWS_INSERTED	BIGINT	rows_inserted - 삽입된 행 수
ROWS_UPDATED	BIGINT	rows_updated - 갱신된 행 수
ROWS_SELECTED	BIGINT	rows_selected - 선택된 행 수
BINDS_PRECOMPILES	BIGINT	binds_precompiles - 시도된 바인드/프리 컴파일 수
OPEN_REM_CURS	BIGINT	open_rem_curs - 열린 리모트 커서 수
OPEN_REM_CURS_BLK	BIGINT	open_rem_curs_blk - 블로킹 상태의 열린 리모트 커서 수
REJ_CURS_BLK	BIGINT	rej_curs_blk - 거부된 블록 커서 요청 수
ACC_CURS_BLK	BIGINT	acc_curs_blk - 승인된 블록 커서 요청 수
SQL_REQS_SINCE_COMMIT	BIGINT	sql_reqs_since_commit - 마지막 커미트 이후 SQL 요청 수
LOCK_TIMEOUTS	BIGINT	lock_timeouts - 잠금 시간종료 수
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 삽입된 내부 행 수
OPEN_LOC_CURS	BIGINT	open_loc_curs - 열린 로컬 커서 수
OPEN_LOC_CURS_BLK	BIGINT	open_loc_curs_blk - 블로킹 상태의 열린 로컬 커서 수
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - 패키지 캐시 찾아보기 수
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - 패키지 캐시 삽입 수
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - 카탈로그 캐시 찾아보기 수
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - 카탈로그 캐시 삽입 수
CAT_CACHE_OVERFLOWS	BIGINT	cat_cache_overflows - 카탈로그 캐시 오버플로우 수
NUM_AGENTS	BIGINT	num_agents - 명령문에서 작업하는 에이전트 수



표 138. SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
AGENTS_STOLEN	BIGINT	agents_stolen - 분실 에이전트 수
ASSOCIATED_AGENTS_TOP	BIGINT	associated_agents_top - 연관 에이전트의 최대 수
APPL_PRIORITY	BIGINT	appl_priority - 응용프로그램 에이전트 우선순위
APPL_PRIORITY_TYPE	VARCHAR(16)	appl_priority_type - 응용프로그램 우선순위 유형. 이 인터페이스는 sqlmon.h로 정의하는 것을 기초로 하고 다음 중 하나인 텍스트 ID를 리턴합니다. <ul style="list-style-type: none"> <li>• DYNAMIC_PRIORITY</li> <li>• FIXED_PRIORITY</li> </ul>
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - 프리페치 대기 시간
APPL_SECTION_LOOKUPS	BIGINT	appl_section_lookups - 섹션 찾아보기 수
APPL_SECTION_INSERTS	BIGINT	appl_section_inserts - 섹션 삽입 수
LOCKS_WAITING	BIGINT	locks_waiting - 잠금 대기 중인 현재 에이전트 수
TOTAL_HASH_JOINS	BIGINT	total_hash_joins - 전체 해시 조인 수
TOTAL_HASH_LOOPS	BIGINT	total_hash_loops - 전체 해시 루프 수
HASH_JOIN_OVERFLOW	BIGINT	hash_join_overflows - 해시 조인 오버플로우 수
HASH_JOIN_SMALL_OVERFLOW	BIGINT	hash_join_small_overflows - 해시 조인 작은 오버플로우 수
APPL_IDLE_TIME	BIGINT	appl_idle_time - 응용프로그램 유휴 시간
UOW_LOCK_WAIT_TIME	BIGINT	uow_lock_wait_time - 잠금 상태로 대기한 전체 작업 단위(UOW) 시간
UOW_COMP_STATUS	VARCHAR(14)	uow_comp_status - 작업 단위(UOW) 완료 상태. 이 인터페이스는 sqlmon.h로 정의하는 것을 기초로 하고 다음 중 하나인 텍스트 ID를 리턴합니다. <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 - 에이전트가 사용한 사용자 CPU 시간

표 138. SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
AGENT_USR_CPU_TIME_MS	BIGINT	agent_usr_cpu_time - 에이전트가 사용한 사용자 CPU 시간
AGENT_SYS_CPU_TIME_S	BIGINT	agent_sys_cpu_time - 에이전트가 사용한 시스템 CPU 시간
AGENT_SYS_CPU_TIME_MS	BIGINT	agent_sys_cpu_time - 에이전트가 사용한 시스템 CPU 시간
APPL_CON_TIME	TIMESTAMP	appl_con_time - 연결 요청 시작 시간소인
CONN_COMPLETE_TIME	TIMESTAMP	conn_complete_time - 연결 요청 완료 시간소인
LAST_RESET	TIMESTAMP	last_reset - 마지막 재설정 시간소인
UOW_START_TIME	TIMESTAMP	uow_start_time - 작업 단위(UOW) 시작 시간소인
UOW_STOP_TIME	TIMESTAMP	uow_stop_time - 작업 단위(UOW) 중지 시간소인
PREV_UOW_STOP_TIME	TIMESTAMP	prev_uow_stop_time - 이전 작업 단위(UOW) 완료 시간소인
UOW_ELAPSED_TIME_S	BIGINT	uow_elapsed_time - 최근 작업 단위(UOW) 경과 시간
UOW_ELAPSED_TIME_MS	BIGINT	uow_elapsed_time - 최근 작업 단위(UOW) 경과 시간
ELAPSED_EXEC_TIME_S	BIGINT	elapsed_exec_time - 명령문 실행 경과 시간
ELAPSED_EXEC_TIME_MS	BIGINT	elapsed_exec_time - 명령문 실행 경과 시간
INBOUND_COMM_ADDRESS	VARCHAR(32)	inbound_comm_address - 인바운드 통신 주소
LOCK_TIMEOUT_VAL	BIGINT	lock_timeout_val - 잠금 시간종료(초)
PRIV_WORKSPACE_NUM_OVERFLOWS	BIGINT	priv_workspace_num_overflows - 개인용 작업 스페이스 오버플로우 수
PRIV_WORKSPACE_SECTION_INSERTS	BIGINT	priv_workspace_section_inserts - 개인용 작업 스페이스 섹션 삽입 수
PRIV_WORKSPACE_SECTION_LOOKUPS	BIGINT	priv_workspace_section_lookups - 개인용 작업 스페이스 섹션 찾아보기 수
PRIV_WORKSPACE_SIZE_TOP	BIGINT	priv_workspace_size_top - 개인용 작업 스페이스 최대 크기
SHR_WORKSPACE_NUM_OVERFLOWS	BIGINT	shr_workspace_num_overflows - 공유 작업 스페이스 오버플로우 수

표 138. SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SHR_WORKSPACE_SECTION_INSERTS	BIGINT	shr_workspace_section_inserts - 공유 작업 스페이스 섹션 삽입 수
SHR_WORKSPACE_SECTION_LOOKUPS	BIGINT	shr_workspace_section_lookups - 공유 작업 스페이스 섹션 찾아보기 수
SHR_WORKSPACE_SIZE_TOP	BIGINT	shr_workspace_size_top - 공유 작업 스페이스 최대 크기
DBPARTITIONNUM	SMALLINT	행에 대한 데이터의 데이터베이스 파티션 검색.
CAT_CACHE_SIZE_TOP	BIGINT	cat_cache_size_top - 카탈로그 캐시 상위 워터 마크(water mark)
TOTAL_OLAP_FUNCS	BIGINT	실행된 총 OLAP 함수 수
OLAP_FUNC_OVERFLOW	BIGINT	OLAP 함수 데이터가 사용 가능한 정렬 힙 스페이스를 초과한 횟수

## SNAPBP 관리 뷰 및 SNAP\_GET\_BP\_V95 테이블 함수 - 버퍼 풀 논리 그룹 스냅샷 정보 검색

SNAPBP 관리 뷰 및 SNAP\_GET\_BP\_V95 테이블 함수는 버퍼 풀 스냅샷에서 버퍼 풀, 특히 버퍼 풀 논리 데이터 그룹에 대한 정보를 리턴합니다.

### SNAPBP 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 버퍼 풀 논리 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPBP\_PART 관리 뷰와 함께 사용되는 경우, SNAPBP 관리 뷰는 GET SNAPSHOT FOR BUFFERPOOLS ON database-alias CLP 명령과 동등한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 전체 정보 목록은 546 페이지의 표 139의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPBP 관리 뷰에 대한 SELECT 특권
- SNAPBP 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_BP\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

현재 연결된 데이터베이스의 모든 버퍼 풀에 대한 데이터 및 인덱스 쓰기 횟수를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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

5개의 레코드가 선택되었습니다.

## SNAP\_GET\_BP\_V95 테이블 함수

SNAP\_GET\_BP\_V95 테이블 함수는 SNAPBP 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에 대한 정보는 검색할 수 있습니다.

SNAP\_GET\_BP\_PART 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_BP\_V95 테이블 함수는 GET SNAPSHOT FOR ALL BUFFERPOOLS CLP 명령과 동등한 데이터를 제공합니다.

리턴될 수 있는 전체 정보 목록은 546 페이지의 표 139의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_BP_V95 ( (dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 있는 파일이 존재하지 않으면, SNAP\_GET\_BP\_V95 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_BP\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

현재 연결된 데이터베이스 파티션의 모든 활성 데이터베이스에 대한 모든 버퍼 풀의 전체 실제 및 논리적 읽기를 검색하십시오.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
DB_NAME      BP_NAME      TOTAL_LOGICAL_READS  ...
-----
SAMPLE      IBMDEFAULTBP      0 ...
TOOLSDB     IBMDEFAULTBP      0 ...
TOOLSDB     BP32K0000         0 ...
```

3개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... TOTAL_PHYSICAL_READS DBPARTITIONNUM
... -----
...                0                0
...                0                0
...                0                0
```

## 리턴되는 정보

표 139. SNAPBP 관리 뷰 및 SNAP\_GET\_BP\_V95 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
BP_NAME	VARCHAR (128)	bp_name - 버퍼 풀 이름
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
DB_PATH	VARCHAR(1024)	db_path - 데이터베이스 경로
INPUT_DB_ALIAS	VARCHAR (128)	input_db_alias - 데이터베이스 별명 입력
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수

표 139. SNAPBP 관리 뷰 및 SNAP\_GET\_BP\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 수
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 수
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - 버퍼 풀 비동기 읽기 시간
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - 버퍼 풀 비동기 쓰기 시간
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 수
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수

표 139. SNAPBP 관리 뷰 및 SNAP\_GET\_BP\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리 페치 페이지 수
FILES_CLOSED	BIGINT	files_closed - 닫힌 데이터베이스 파일 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - 버퍼 풀 희생 버퍼 없음
PAGES_FROM_BLOCK_IOS	BIGINT	pages_from_block_ios - 블록 입출력으로 읽은 총 페이지 수
PAGES_FROM_VECTORED_IOS	BIGINT	pages_from_vectored_ios - 벡터된 입출력으로 읽은 총 페이지 수
VECTORED_IOS	BIGINT	vectored_ios - 벡터된 입출력 요청 수
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPBP\_PART 관리 뷰 및 SNAP\_GET\_BP\_PART 테이블 함수 - bufferpool\_nodeinfo 논리 데이터 그룹 스냅샷 정보 검색

SNAPBP\_PART 관리 뷰 및 SNAP\_GET\_BP\_PART 테이블 함수는 버퍼 풀 스냅샷에서 버퍼 풀, 특히 bufferpool\_nodeinfo 논리 데이터 그룹에 대한 정보를 리턴합니다.

### SNAPBP\_PART 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 bufferpool\_nodeinfo 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPBP 관리 뷰와 함께 사용되는 경우, SNAPBP\_PART 관리 뷰는 GET SNAPSHOT FOR BUFFERPOOLS ON database-alias CLP 명령과 동등한 데이터를 제공합니다.



스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 552 페이지의 표 140의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPBP\_PART 관리 뷰에 대한 SELECT 특권
- SNAPBP\_PART 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_BP\_PART 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

SAMPLE 데이터베이스에 연결된 경우 모든 버퍼 풀에 대한 데이터를 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

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	...

4개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	BP_PAGES_LEFT_TO_REMOVE	BP_TBSP_USE_COUNT
...	0	3
...	0	0
...	0	0
...	0	0
...		

## SNAP\_GET\_BP\_PART 테이블 함수

SNAP\_GET\_BP\_PART 테이블 함수는 SNAPBP\_PART 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_BP\_V95 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_BP\_PART 테이블 함수는 GET SNAPSHOT FOR ALL BUFFERPOOLS CLP 명령과 동등한 데이터를 제공합니다.

리턴될 수 있는 정보의 완전한 목록은 552 페이지의 표 140의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_BP_PART ( ( dbname [ , dbpartitionnum ] ) )
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스와 동일한 인터페이스 내의 모든 데이터베이스에 있는 모든 버퍼 풀에 대한 스냅샷을 가져오려면 NULL 값을 지정하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_BP\_PART 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_BP\_PART 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

SAMPLE 데이터베이스에 연결된 경우 모든 활성 데이터베이스의 모든 버퍼풀에 대한 데이터를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...

...

이 쿼리의 출력(계속).

...	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

...

## 리턴되는 정보

표 140. SNAPBP\_PART 관리 뷰 및 SNAP\_GET\_BP\_PART 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
BP_NAME	VARCHAR (128)	bp_name - 버퍼 풀 이름
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
BP_CUR_BUFFSZ	BIGINT	bp_cur_buffsz - 버퍼 풀의 현재 크기
BP_NEW_BUFFSZ	BIGINT	bp_new_buffsz - 새 버퍼 풀 크기
BP_PAGES_LEFT_TO_REMOVE	BIGINT	bp_pages_left_to_remove - 제거할 나머지 페이지 수
BP_TBSP_USE_COUNT	BIGINT	bp_tbsp_use_count - 버퍼 풀에 맵핑된 테이블 스페이스 수
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPCONTAINER 관리 뷰 및 SNAP\_GET\_CONTAINER\_V91 테이블 함수 - tablespace\_container 논리 데이터 그룹 스냅샷 정보 검색

SNAPCONTAINER 관리 뷰 및 SNAP\_GET\_CONTAINER\_V91 테이블 함수는 tablespace\_container 논리 데이터 그룹에서 테이블 스페이스 스냅샷 정보를 리턴합니다.

### SNAPCONTAINER 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 tablespace\_container 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPTbsp, SNAPTbsp\_part, SNAPTbsp\_quiescer 및 SNAPTbsp\_range 관리 뷰와 함께 사용될 경우, SNAPCONTAINER 관리 뷰는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 556 페이지의 표 141의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPCONTAINER 관리 뷰에 대한 SELECT 특권
- SNAPCONTAINER 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_CONTAINER\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

현재 연결된 데이터베이스에 대한 모든 데이터베이스 파티션의 테이블 스페이스 컨테이너 세부사항을 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...
			...

8개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	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	...

이 쿼리의 출력(계속).

...	ACCESSIBLE	DBPARTITIONNUM
...	1	0

...	1	0
...	1	0
...	1	0
...	1	1
...	1	1
...	1	2
...	1	2

## SNAP\_GET\_CONTAINER\_V91 테이블 함수

SNAP\_GET\_CONTAINER\_V91 테이블 함수는 SNAPCONTAINER 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_TBSP\_V91, SNAP\_GET\_TBSP\_PART\_V91, SNAP\_GET\_TBSP\_QUIESCER 및 SNAP\_GET\_TBSP\_RANGE 테이블 함수와 함께 사용될 경우, SNAP\_GET\_CONTAINER\_V91 테이블 함수는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 데이터를 리턴합니다.

리턴 가능한 전체 정보 목록은 556 페이지의 표 141의 내용을 참조하십시오.

## 구문

```

▶▶ SNAP_GET_CONTAINER_V91 ( (dbname [ , dbpartitionnum ] ) )

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### dbname

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. NULL 또는 빈 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 얻으십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. dbname이 NULL로 설정되어 있지 않고 dbpartitionnum이 NULL로 설정되어 있다면, -1은 내재적으로 dbpartitionnum으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, dbname만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결에 사용할 수 있으며 응용프로그램에서 사용할 수 있는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_CONTAINER\_V91 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_CONTAINER\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 데이터베이스 파티션에서 현재 연결된 데이터베이스의 테이블 스페이스 컨테이너 세부사항을 검색합니다.

```
SELECT SNAPSHOT_TIMESTAMP, TBSP_NAME, TBSP_ID, CONTAINER_NAME,
       CONTAINER_ID, CONTAINER_TYPE, ACCESSIBLE
FROM TABLE(SNAP_GET_CONTAINER_V91(' ', -1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

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	WHATNEWTEMPSPACE	5	...

이 쿼리의 출력(계속).

...	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:#DGTTsWhatsNewContainer	0	...

이 쿼리의 출력(계속).

```

... CONTAINER_TYPE ACCESSIBLE
... -----
... CONT_PATH          1
... CONT_PATH          1
... CONT_PATH          1
... CONT_PATH          1
... CONT_PATH          1
... CONT_PATH          1

```

## 리턴되는 정보

메모: 파일 시스템 정보를 리턴하려면 BUFFERPOOL 데이터베이스 관리 프로그램 모니터 스위치를 설정해야 합니다.

표 141. SNAPCONTAINER 관리 뷰 및 SNAP\_GET\_CONTAINER\_V91 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TBSP_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
CONTAINER_NAME	VARCHAR(256)	container_name - 컨테이너 이름
CONTAINER_ID	BIGINT	container_id - 컨테이너 ID
CONTAINER_TYPE	VARCHAR(16)	container_type - 컨테이너 유형. 이는 sqlutil.h의 정의를 기반으로 하는 텍스트 ID로 다음 중 하나입니다. <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 - 컨테이너의 전체 페이지 수
USABLE_PAGES	BIGINT	container_usable_pages - 컨테이너에서 사용할 수 있는 페이지 수
ACCESSIBLE	SMALLINT	container_accessible - 컨테이너의 액세스 가능성
STRIPE_SET	BIGINT	container_stripe_set - 스트라이프 세트
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
FS_ID	VARCHAR(22)	fs_id - 고유 파일 시스템 ID 번호
FS_TOTAL_SIZE	BIGINT	fs_total_size - 파일 시스템의 전체 크기
FS_USED_SIZE	BIGINT	fs_used_size - 파일 시스템에서 사용된 스페이스 용량



---

## SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수 - dbase 논리 그룹에서 스냅샷 정보 검색

『SNAPDB 관리 뷰』 및 558 페이지의 『SNAP\_GET\_DB\_V95 테이블 함수』에서는 데이터베이스(dbase) 논리 그룹으로부터 스냅샷 정보를 리턴합니다.

### SNAPDB 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 dbase 논리 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPDB\_MEMORY\_POOL, SNAPDETAILLOG, SNAPHADR 및 SNAPSTORAGE\_PATHS 관리 뷰와 함께 사용되는 경우, SNAPDB 관리 뷰는 GET SNAPSHOT FOR DATABASE on database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴되는 전체 정보 목록은 561 페이지의 표 142의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPDB 관리 뷰에 대한 SELECT 특권
- SNAPDB 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_DB\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### 예:

현재 연결된 데이터베이스의 모든 데이터베이스 파티션에 대한 상태, 플랫폼, 위치 및 연결 시간을 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	DB_STATUS	SERVER_PLATFORM	DB_LOCATION	...
TEST	ACTIVE	AIX64	LOCAL	...
TEST	ACTIVE	AIX64	LOCAL	...
TEST	ACTIVE	AIX64	LOCAL	...

3개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	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

명령행에 다음을 호출하여 이 루틴을 사용할 수 있습니다.

```
SELECT TOTAL_OLAP_FUNCS, OLAP_FUNC_OVERFLOW, ACTIVE_OLAP_FUNCS
FROM SYSIBMADM.SNAPDB
```

TOTAL_OLAP_FUNCS	OLAP_FUNC_OVERFLOW	ACTIVE_OLAP_FUNCS
7	2	1

1개의 레코드가 선택되었습니다.

워크로드를 실행한 후 사용자가 다음 쿼리를 사용할 수 있습니다.

```
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

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_DB\_V95 테이블 함수

SNAP\_GET\_DB\_V95 테이블 함수는 SNAPDB 관리 뷰와 동일한 정보를 리턴합니다.

SNAP\_GET\_DB\_MEMORY\_POOL, SNAP\_GET\_DETAILLOG\_V91, SNAP\_GET\_HADR 및 SNAP\_GET\_STORAGE\_PATHS 테이블 함수와 함께 사용

되는 경우, SNAP\_GET\_DB\_V95 테이블 함수는 GET SNAPSHOT FOR ALL DATABASES CLP 명령과 동등한 정보를 제공합니다.

리턴되는 전체 정보 목록은 561 페이지의 표 142의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_DB_V95 ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 있는 파일이 존재하지 않으면, SNAP\_GET\_DB\_V95 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_DB\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예:

예 1: 현재 연결된 데이터베이스의 모든 데이터베이스 파티션에서 집계 뷰로 상태, 플랫폼, 위치 및 연결 시간을 검색합니다.

```
SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,
       DB_LOCATION, DB_CONN_TIME FROM TABLE(SNAP_GET_DB_V95(' ', -2)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	DB_STATUS	SERVER_PLATFORM	...
-----	-----	-----	...
SAMPLE	ACTIVE	AIX64	...

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	DB_LOCATION	DB_CONN_TIME
...	-----	-----
...	LOCAL	2005-07-24-22.09.22.013196

예 2: 현재 연결된 데이터베이스가 포함된 동일한 인스턴스에서 모든 활성 데이터베이스에 대해 모든 데이터베이스 파티션에서 집계 뷰로 상태, 플랫폼, 위치 및 연결 시간을 검색합니다.

```
SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,
       DB_LOCATION, DB_CONN_TIME
FROM TABLE(SNAP_GET_DB_V95(CAST (NULL AS VARCHAR
(128)), -2)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	DB_STATUS	SERVER_PLATFORM	...
-----	-----	-----	...
TOOLSDB	ACTIVE	AIX64	...
SAMPLE	ACTIVE	AIX64	...

이 쿼리의 출력(계속).

...	DB_LOCATION	DB_CONN_TIME
...	-----	-----
...	LOCAL	2005-07-24-22.26.54.396335
...	LOCAL	2005-07-24-22.09.22.013196

예 3: 명령행에 다음을 호출하여 이 루틴을 사용할 수 있습니다.

데이터베이스에 연결된 경우:

```
SELECT TOTAL_OLAP_FUNCS, OLAP_FUNC_OVERFLOW, ACTIVE_OLAP_FUNCS
FROM TABLE (SNAP_GET_DB_V95('', 0)) AS T
```

출력은 다음과 같습니다.

```
TOTAL_OLAP_FUNCS  OLAP_FUNC_OVERFLOW  ACTIVE_OLAP_FUNCS
-----
                          7                          2                          1
```

1개의 레코드가 선택되었습니다.

예 4: 워크로드를 실행한 후 사용자가 테이블 함수와 함께 다음 쿼리를 사용할 수 있습니다.

```
SELECT STATS_CACHE_SIZE, STATS_FABRICATIONS, SYNC_RUNSTATS,
ASYNC_RUNSTATS, STATS_FABRICATE_TIME, SYNC_RUNSTATS_TIME
FROM TABLE (SNAP_GET_DB_V95('mytestdb', -1)) AS SNAPDB
```

```
STATS_CACHE_SIZE  STATS_FABRICATIONS  SYNC_RUNSTATS  ASYNC_RUNSTATS  ...
-----
                200                          1                2                0  ...
```

계속됨

```
...STATS_FABRICATE_TIME  SYNC_RUNSTATS_TIME
...-----
...                2                32
```

1개의 레코드가 선택되었습니다.

## SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수 메타데이터

표 142. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
DB_PATH	VARCHAR(1024)	db_path - 데이터베이스 경로
INPUT_DB_ALIAS	VARCHAR (128)	input_db_alias - 데이터베이스 별명 입력
DB_STATUS	VARCHAR(12)	db_status - 데이터베이스 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>ACTIVE</li> <li>QUIESCE_PEND</li> <li>QUIESCED</li> <li>ROLLFWD</li> </ul>
CATALOG_PARTITION	SMALLINT	catalog_node - 카탈로그 노드 번호
CATALOG_PARTITION_NAME	VARCHAR (128)	catalog_node_name - 카탈로그 노드 네트워크 이름

표 142. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SERVER_PLATFORM	VARCHAR(12)	<p>server_platform - 서버 운영 체제. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다.</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>

표 142. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DB_LOCATION	VARCHAR(12)	db_location - 데이터베이스 위치. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. • LOCAL • REMOTE
DB_CONN_TIME	TIMESTAMP	db_conn_time - 데이터베이스 활성화 시간소인
LAST_RESET	TIMESTAMP	last_reset - 마지막 재설정 시간소인
LAST_BACKUP	TIMESTAMP	last_backup - 최종 백업 시간소인
CONNECTIONS_TOP	BIGINT	connections_top - 최대 동시 연결 수
TOTAL_CONS	BIGINT	total_cons - 데이터베이스 활성화 이후 연결 수
TOTAL_SEC_CONS	BIGINT	total_sec_cons - 2차 연결 수
APPLS_CUR_CONS	BIGINT	appls_cur_cons - 현재 연결된 응용프로그램 수
APPLS_IN_DB2	BIGINT	appls_in_db2 - 현재 데이터베이스에서 실행 중인 응용프로그램 수
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 연관 에이전트 수
AGENTS_TOP	BIGINT	agents_top - 작성된 에이전트 수
COORD_AGENTS_TOP	BIGINT	coord_agents_top - 최대 코디네이팅 에이전트 수
LOCKS_HELD	BIGINT	locks_held - 보유된 잠금 수
LOCK_WAITS	BIGINT	lock_waits - 잠금 대기 수
LOCK_WAIT_TIME	BIGINT	lock_wait_time - 잠금 대기 시간
LOCK_LIST_IN_USE	BIGINT	lock_list_in_use - 사용 중인 전체 잠금 목록 메모리
DEADLOCKS	BIGINT	deadlocks - 발견된 교착 상태 수
LOCK_ESCALS	BIGINT	lock_escals - 잠금 에스컬레이션 수
X_LOCK_ESCALS	BIGINT	x_lock_escals - 배타적 잠금 에스컬레이션 수
LOCKS_WAITING	BIGINT	locks_waiting - 잠금 대기 중인 현재 에이전트 수
LOCK_TIMEOUTS	BIGINT	lock_timeouts - 잠금 시간종료 수
NUM_INDOUBT_TRANS	BIGINT	num_indoubt_trans - 인다우트(Indoubt) 트랜잭션 수
SORT_HEAP_ALLOCATED	BIGINT	sort_heap_allocated - 할당된 전체 정렬 힙
SORT_SHRHEAP_ALLOCATED	BIGINT	sort_shrheap_allocated - 현재 할당된 정렬 공유 힙
SORT_SHRHEAP_TOP	BIGINT	sort_shrheap_top - 공유 힙 상위 워터 마크(water mark) 정렬
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - 포스트 공유 임계값 정렬 수
TOTAL_SORTS	BIGINT	total_sorts - 전체 정렬 수
TOTAL_SORT_TIME	BIGINT	total_sort_time - 총 정렬 시간
SORT_OVERFLOWS	BIGINT	sort_overflows - 정렬 오버플로우 수
ACTIVE_SORTS	BIGINT	active_sorts - 활성 정렬 수
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수

표 142. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - 버퍼 풀 비동기 읽기 시간
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - 버퍼 풀 비동기 쓰기 시간
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 수



표 142. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - 버퍼 풀 희생 버퍼 없음
POOL_LSN_GAP_CLNS	BIGINT	pool_lsn_gap_clns - 버퍼 풀 로그 스페이스 클리너 트리거
POOL_DRTY_PG_STEAL_CLNS	BIGINT	pool_drty_pg_steal_clns - 버퍼 풀 희생(victim) 페이지 클리너 트리거
POOL_DRTY_PG_THRSH_CLNS	BIGINT	pool_drty_pg_thrsh_clns - 버퍼 풀 임계값 클리너 트리거
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - 프리페치 대기 시간
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리페치 페이지 수
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
FILES_CLOSED	BIGINT	files_closed - 닫힌 데이터베이스 파일 수
ELAPSED_EXEC_TIME_S	BIGINT	elapsed_exec_time - 명령문 실행 경과 시간
ELAPSED_EXEC_TIME_MS	BIGINT	elapsed_exec_time - 명령문 실행 경과 시간
COMMIT_SQL_STMTS	BIGINT	commit_sql_stmts - 시도된 명령문 커밋
ROLLBACK_SQL_STMTS	BIGINT	rollback_sql_stmts - 시도된 롤백 명령문
DYNAMIC_SQL_STMTS	BIGINT	dynamic_sql_stmts - 시도된 동적 SQL문
STATIC_SQL_STMTS	BIGINT	static_sql_stmts - 시도된 정적 SQL문
FAILED_SQL_STMTS	BIGINT	failed_sql_stmts - 실패한 명령문 조작
SELECT_SQL_STMTS	BIGINT	select_sql_stmts - 실행된 Select SQL문
UID_SQL_STMTS	BIGINT	uid_sql_stmts - 실행된 UPDATE/INSERT/DELETE SQL문
DDL_SQL_STMTS	BIGINT	ddl_sql_stmts - DDL(Data Definition Language) SQL문
INT_AUTO_REBINDS	BIGINT	int_auto_rebinds - 내부 자동 리바인드 수
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 삭제된 내부 행 수
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 삽입된 내부 행 수
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 갱신된 내부 행 수
INT_COMMITS	BIGINT	int_commits - 내부 커밋 수
INT_ROLLBACKS	BIGINT	int_rollback - 내부 롤백 수

표 142. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
INT_DEADLOCK_ROLLBACKS	BIGINT	int_deadlock_rollback - 교착 상태로 인한 내부 롤백 수
ROWS_DELETED	BIGINT	rows_deleted - 삭제된 행 수
ROWS_INSERTED	BIGINT	rows_inserted - 삽입된 행 수
ROWS_UPDATED	BIGINT	rows_updated - 갱신된 행 수
ROWS_SELECTED	BIGINT	rows_selected - 선택된 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
BINDS_PRECOMPILES	BIGINT	binds_precompiles - 시도된 바인드/프리컴파일 수
TOTAL_LOG_AVAILABLE	BIGINT	total_log_available - 사용 가능한 전체 로그
TOTAL_LOG_USED	BIGINT	total_log_used - 사용된 전체 로그 스페이스
SEC_LOG_USED_TOP	BIGINT	sec_log_used_top - 사용된 최대 2차 로그 스페이스
TOT_LOG_USED_TOP	BIGINT	tot_log_used_top - 사용된 최대 전체 로그 스페이스
SEC_LOGS_ALLOCATED	BIGINT	sec_logs_allocated - 현재 할당된 2차 로그
LOG_READS	BIGINT	log_reads - 읽은 로그 페이지 수
LOG_READ_TIME_S	BIGINT	log_read_time - 로그 읽기 시간
LOG_READ_TIME_NS	BIGINT	log_read_time - 로그 읽기 시간
LOG_WRITES	BIGINT	log_writes - 쓴 로그 페이지 수
LOG_WRITE_TIME_S	BIGINT	log_write_time - 로그 쓰기 시간
LOG_WRITE_TIME_NS	BIGINT	log_write_time - 로그 쓰기 시간
NUM_LOG_WRITE_IO	BIGINT	num_log_write_io - 로그 쓰기 수
NUM_LOG_READ_IO	BIGINT	num_log_read_io - 로그 읽기 수
NUM_LOG_PART_PAGE_IO	BIGINT	num_log_part_page_io - 부분 로그 페이지 쓰기 수
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - 전체 로그 버퍼 수
NUM_LOG_DATA_FOUND_IN_BUFFER	BIGINT	num_log_data_found_in_buffer - 버퍼에 있는 로그 데이터 수
APPL_ID_OLDEST_XACT	BIGINT	appl_id_oldest_xact - 가장 오래된 트랜잭션이 포함된 응용프로그램
LOG_TO_REDO_FOR_RECOVERY	BIGINT	log_to_redo_for_recovery - 복구를 위해 재실행되는 로그 수량
LOG_HELD_BY_DIRTY_PAGES	BIGINT	log_held_by_dirty_pages - 더티 페이지별로 계산되는 로그 스페이스 수량
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - 패키지 캐시 찾아보기 수
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - 패키지 캐시 삽입 수
PKG_CACHE_NUM_OVERFLOW	BIGINT	pkg_cache_num_overflows - 패키지 캐시 오버플로우 수
PKG_CACHE_SIZE_TOP	BIGINT	pkg_cache_size_top - 패키지 캐시 상위 워터 마크 (water mark)
APPL_SECTION_LOOKUPS	BIGINT	appl_section_lookups - 섹션 찾아보기 수

표 142. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
APPL_SECTION_INSERTS	BIGINT	appl_section_inserts - 섹션 삽입 수
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - 카탈로그 캐시 찾아보기 수
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - 카탈로그 캐시 삽입 수
CAT_CACHE_OVERFLOWS	BIGINT	cat_cache_overflows - 카탈로그 캐시 오버플로우 수
CAT_CACHE_SIZE_TOP	BIGINT	cat_cache_size_top - 카탈로그 캐시 상위 워터 마크(water mark)
PRIV_WORKSPACE_SIZE_TOP	BIGINT	priv_workspace_size_top - 개인용 작업 스페이스 최대 크기
PRIV_WORKSPACE_NUM_OVERFLOWS	BIGINT	priv_workspace_num_overflows - 개인용 작업 스페이스 오버플로우 수
PRIV_WORKSPACE_SECTION_INSERTS	BIGINT	priv_workspace_section_inserts - 개인용 작업 스페이스 섹션 삽입 수
PRIV_WORKSPACE_SECTION_LOOKUPS	BIGINT	priv_workspace_section_lookups - 개인용 작업 스페이스 섹션 찾아보기 수
SHR_WORKSPACE_SIZE_TOP	BIGINT	shr_workspace_size_top - 공유 작업 스페이스 최대 크기
SHR_WORKSPACE_NUM_OVERFLOWS	BIGINT	shr_workspace_num_overflows - 공유 작업 스페이스 오버플로우 수
SHR_WORKSPACE_SECTION_INSERTS	BIGINT	shr_workspace_section_inserts - 공유 작업 스페이스 섹션 삽입 수
SHR_WORKSPACE_SECTION_LOOKUPS	BIGINT	shr_workspace_section_lookups - 공유 작업 스페이스 섹션 찾아보기 수
TOTAL_HASH_JOINS	BIGINT	total_hash_joins - 전체 해시 조인 수
TOTAL_HASH_LOOPS	BIGINT	total_hash_loops - 전체 해시 루프 수
HASH_JOIN_OVERFLOWS	BIGINT	hash_join_overflows - 해시 조인 오버플로우 수
HASH_JOIN_SMALL_OVERFLOWS	BIGINT	hash_join_small_overflows - 해시 조인 작은 오버플로우 수
POST_SHRTHRESHOLD_HASH_JOINS	BIGINT	post_shrthreshold_hash_joins - 포스트 임계값 해시 조인 수
ACTIVE_HASH_JOINS	BIGINT	active_hash_joins - 활성 해시 조인 수
NUM_DB_STORAGE_PATHS	BIGINT	num_db_storage_paths - 자동 스토리지 경로 수
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
SMALLEST_LOG_AVAIL_NODE	INTEGER	smallest_log_avail_node - 최소 사용 가능 로그 스페이스가 포함된 노드
TOTAL_OLAP_FUNCS	BIGINT	실행된 총 OLAP 함수 수

표 142. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
OLAP_FUNC_OVERFLOWS	BIGINT	OLAP 함수 데이터가 사용 가능한 정렬 힙 스페이스를 초과한 횟수
ACTIVE_OLAP_FUNCS	BIGINT	정렬 힙 메모리를 현재 실행하여 소비 중인 전체 OLAP 함수의 수
STATS_CACHE_SIZE	BIGINT	통계 캐시의 크기(바이트)
STATS_FABRICATIONS	BIGINT	테이블 또는 인덱스 스캔 없이 시스템이 통계를 작성하기 위한 전체 통계 수집 활동 수
SYNC_RUNSTATS	BIGINT	쿼리 컴파일 중의 전체 동기 통계 수집 활동 수
ASYNC_RUNSTATS	BIGINT	이 컬럼을 완료된 전체 비동기 통계 수집 활동 수로 변경합니다.
STATS_FABRICATE_TIME	BIGINT	쿼리 컴파일 중에 테이블 또는 인덱스 스캔 없이 통계를 작성하는데 시스템이 소비한 밀리초 단위의 전체 시간
SYNC_RUNSTATS_TIME	BIGINT	동기 통계 수집 활동에 소비된 밀리초 단위의 전체 시간
NUM_THRESHOLD_VIOLATIONS	BIGINT	데이터베이스에 발생한 임계값 위반 수

## SNAPDB\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수 - 데이터베이스 레벨 메모리 사용 정보 검색

SNAPDB\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수는 UNIX 플랫폼에만 해당하는 데이터베이스 레벨에서 메모리 사용에 대한 정보를 리턴합니다.

### SNAPDB\_MEMORY\_POOL 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 데이터베이스 레벨 메모리 사용 정보를 검색할 수 있습니다.

SNAPDB, SNAPDETAILLOG, SNAPHDR 및 SNAPSTORAGE\_PATHS 관리 뷰와 함께 사용되는 경우, SNAPDB\_MEMORY\_POOL 관리 뷰는 GET SNAPSHOT FOR DATABASE ON database-alias CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 571 페이지의 표 143의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPDB\_MEMORY\_POOL 관리 뷰에 대한 SELECT 특권
- SNAPDB\_MEMORY\_POOL 관리 뷰에 대한 CONTROL 특권

- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

**예 :**

현재 연결된 데이터베이스, SAMPLE에 대한 메모리 풀 및 현재 크기 목록을 검색합니다.

```
SELECT POOL_ID, POOL_CUR_SIZE FROM SYSIBMADM.SNAPDB_MEMORY_POOL
```

이 쿼리의 출력 예는 다음과 같습니다.

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

12개의 레코드가 선택되었습니다.

## SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수

SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수는 SNAPDB\_MEMORY\_POOL 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 특정 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_DB\_V95, SNAP\_GET\_DETAILLOG\_V91, SNAP\_GET\_HADR 및 SNAP\_GET\_STORAGE\_PATHS 테이블 함수와 함께 사용되는 경우,

SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수는 GET SNAPSHOT FOR ALL DATABASES CLP 명령과 동등한 정보를 제공합니다.

리턴될 수 있는 정보의 완전한 목록은 571 페이지의 표 143의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_DB_MEMORY_POOL ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

모든 데이터베이스에 대한 메모리 풀 및 현재 크기 목록을 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

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

24개의 레코드가 선택되었습니다.

### 리턴되는 정보

표 143. SNAPDB\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름

표 143. SNAPDB\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_ID	VARCHAR(14)	pool_id - 메모리 풀 ID. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 메모리 풀 2차 ID
POOL_CUR_SIZE	BIGINT	pool_cur_size - 메모리 풀의 현재 크기
POOL_WATERMARK	BIGINT	pool_watermark - 메모리 풀 워터 마크 (water mark)
POOL_CONFIG_SIZE	BIGINT	pool_config_size - 메모리 풀의 구성된 크기
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPDBM 관리 뷰 및 SNAP\_GET\_DBM\_V95 테이블 함수 - dbm 논리적 그룹화 스냅샷 정보 검색

SNAPDBM 관리 뷰 및 SNAP\_GET\_DBM\_V95 테이블 함수는 스냅샷 모니터 DB2 데이터베이스 관리 프로그램(dbm) 논리적 그룹화 정보를 리턴합니다.



## SNAPDBM 관리 뷰

SNAPDBM\_MEMORY\_POOL, SNAPFCM, SNAPFCM\_PART 및 SNAPSWITCHES 관리 뷰와 같이 사용하여 SNAPDBM 관리 뷰는 GET SNAPSHOT FOR DBM 명령과 동등한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 575 페이지의 표 144의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPDBM 관리 뷰에 대한 SELECT 특권
- SNAPDBM 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_DBM\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

### 예 :

모든 데이터베이스 파티션에 대해 데이터베이스 관리 프로그램 상태 및 연결 정보를 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

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	-	...

3개의 레코드가 선택되었습니다.

...

이 쿼리의 출력(계속).

...	LOCAL_CONS	REM_CONS_IN	AGENT_USAGE	DBPARTITIONNUM
...	1	1	0	0
...	0	0	0	1
...	0	0	0	2

## SNAP\_GET\_DBM\_V95 테이블 함수

SNAP\_GET\_DBM\_V95 테이블 함수는 SNAPDBM 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에 대한 정보를 검색할 수도 있습니다.

SNAP\_GET\_DBM\_MEMORY\_POOL, SNAP\_GET\_FCM, SNAP\_GET\_FCM\_PART 및 SNAP\_GET\_SWITCHES 테이블 함수와 함께 사용되는 SNAP\_GET\_DBM\_V95 테이블 함수는 GET SNAPSHOT FOR DBM 명령과 동등한 데이터를 제공합니다.

리턴 가능한 전체 정보 목록은 575 페이지의 표 144의 내용을 참조하십시오.

## 구문

```

▶▶ SNAP_GET_DBM_V95 ( [ dbpartitionnum ] )

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_DBM\_V95 테이블 함수가 메모리에서 스냅샷을 호출합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_DBM\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

데이터베이스 파티션 번호 2의 시작 시간 및 현재 상태를 검색합니다.

```
SELECT DB2START_TIME, DB2_STATUS FROM TABLE(SNAP_GET_DBM_V95(2)) AS T
```

다음은 이 쿼리의 출력 예입니다.

```
DB2START_TIME          DB2_STATUS
-----
2006-01-06-14.59.59.062798 ACTIVE
```

### 리턴된 정보

표 144. SNAPDBM 관리 뷰 및 SNAP\_GET\_DBM\_V95 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
SORT_HEAP_ALLOCATED	BIGINT	sort_heap_allocated - 할당된 전체 정렬 힙
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - 포스트 임계값 정렬 수
PIPED_SORTS_REQUESTED	BIGINT	piped_sorts_requested - 요청된 파이프 정렬 수
PIPED_SORTS_ACCEPTED	BIGINT	piped_sorts_accepted - 수락된 파이프 정렬 수
REM_CONS_IN	BIGINT	rem_cons_in - 데이터베이스 관리 프로그램에 리모트 연결 수
REM_CONS_IN_EXEC	BIGINT	rem_cons_in_exec - 데이터베이스 관리 프로그램 모니터 요소에서 실행되는 리모트 연결 수
LOCAL_CONS	BIGINT	local_cons - 로컬 연결 수
LOCAL_CONS_IN_EXEC	BIGINT	local_cons_in_exec - 데이터베이스 관리 프로그램 모니터 요소에서 실행되는 로컬 연결 수
CON_LOCAL_DBASES	BIGINT	con_local_dbases - 현재 연결된 로컬 데이터베이스 수
AGENTS_REGISTERED	BIGINT	agents_registered - 등록된 에이전트 수
AGENTS_WAITING_ON_TOKEN	BIGINT	agents_waiting_on_token - 토큰을 기다리는 에이전트 수
DB2_STATUS	VARCHAR(12)	db2_status - DB2 인스턴스 상태  이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• ACTIVE</li> <li>• QUIESCE_PEND</li> <li>• QUIESCED</li> </ul>
AGENTS_REGISTERED_TOP	BIGINT	agents_registered_top - 등록된 최대 에이전트 수
AGENTS_WAITING_TOP	BIGINT	agents_waiting_top - 대기 중인 최대 에이전트 수

표 144. SNAPDBM 관리 뷰 및 SNAP\_GET\_DBM\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
COMM_PRIVATE_MEM	BIGINT	comm_private_mem - 커미트된 전용 메모리
IDLE_AGENTS	BIGINT	idle_agents - 유휴 에이전트 수
AGENTS_FROM_POOL	BIGINT	agents_from_pool - 풀로부터 할당된 에이전트 수
AGENTS_CREATED_EMPTY_POOL	BIGINT	agents_created_empty_pool - 비어 있는 에이전트 풀로 인해 작성된 에이전트 수
COORD_AGENTS_TOP	BIGINT	coord_agents_top - 최대 코디네이팅 에이전트 수
MAX_AGENT_OVERFLOW	BIGINT	max_agent_overflows - 최대 에이전트 오버플로우 수
AGENTS_STOLEN	BIGINT	agents_stolen - 분실 에이전트 수
GW_TOTAL_CONS	BIGINT	gw_total_cons - DB2 Connect에 대한 전체 연결 시도 수
GW_CUR_CONS	BIGINT	gw_cur_cons - DB2 Connect에 대한 현재 연결 수
GW_CONS_WAIT_HOST	BIGINT	gw_cons_wait_host - 호스트 응답을 대기 중인 연결 수
GW_CONS_WAIT_CLIENT	BIGINT	gw_cons_wait_client - 클라이언트의 요청 전송을 대기 중인 연결 수
POST_THRESHOLD_HASH_JOINS	BIGINT	post_threshold_hash_joins - 해시 조인 임계값
NUM_GW_CONN_SWITCHES	BIGINT	num_gw_conn_switches - 연결 전환 수
DB2START_TIME	TIMESTAMP	db2start_time - 데이터베이스 관리 프로그램 시작 시간소인
LAST_RESET	TIMESTAMP	last_reset - 마지막 재설정 시간소인
NUM_NODES_IN_DB2_INSTANCE	INTEGER	num_nodes_in_db2_instance - 데이터베이스 파티션의 노드 수
PRODUCT_NAME	VARCHAR(32)	product_name - 제품 이름
SERVICE_LEVEL	VARCHAR(18)	service_level - 서비스 레벨
SORT_HEAP_TOP	BIGINT	sort_heap_top - 전용 힙 상위 워터 마크(water mark) 정렬
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

표 144. SNAPDBM 관리 뷰 및 SNAP\_GET\_DBM\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POST_THRESHOLD_OLAP_FUNCS	BIGINT	정렬 힙 임계값이 초과된 후에 정렬 힙을 요청한 OLAP 함수의 수입니다.  정렬 힙을 사용하는 조건의 예로는 정렬, 해시 조인 및 OLAP 함수가 있습니다. 일반적인 조건인 경우 데이터베이스 관리 프로그램은 <code>sorheap</code> 구성 매개변수로 지정한 값을 사용하여 정렬 힙을 할당합니다. 정렬 힙에 할당된 메모리 양이 정렬 힙 임계값( <code>sheapthres</code> 구성 매개변수)을 초과하는 경우 데이터베이스 관리 프로그램은 <code>sorheap</code> 구성 매개변수로 지정한 값보다 작은 값을 사용하여 후속 정렬 힙을 할당합니다.  정렬 힙 임계값에 도달한 후에 시작된 OLAP 함수는 실행할 최적의 메모리 양을 수신하지 못할 수도 있습니다.

## SNAPDBM\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수 - 데이터베이스 관리 프로그램 레벨 메모리 사용 정보 검색

SNAPDBM\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수는 데이터베이스 관리 프로그램에서 메모리 사용에 대한 정보를 리턴합니다.

### SNAPDBM\_MEMORY\_POOL 관리 뷰

SNAPDBM, SNAPFCM, SNAPFCM\_PART 및 SNAPSWITCHES 관리 뷰와 함께 사용되는 경우, SNAPDBM\_MEMORY\_POOL 관리 뷰는 GET SNAPSHOT FOR DBM 명령과 동등한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 580 페이지의 표 145의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPDBM\_MEMORY\_POOL 관리 뷰에 대한 SELECT 특권
- SNAPDBM\_MEMORY\_POOL 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

예 :

연결된 데이터베이스의 데이터베이스 관리 프로그램에 대한 메모리 풀 및 현재 크기 목록을 검색합니다.

```
SELECT POOL_ID, POOL_CUR_SIZE FROM SNAPDBM_MEMORY_POOL
```

이 쿼리의 출력 예는 다음과 같습니다.

POOL_ID	POOL_CUR_SIZE
MONITOR	65536
OTHER	29622272
FCMBP	57606144
...	

## SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수

SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수는

SNAPAGENT\_MEMORY\_POOL 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_DBM\_V95, SNAP\_GET\_FCM, SNAP\_GET\_FCM\_PART 및 SNAP\_GET\_SWITCHES 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수는 GET SNAPSHOT FOR DBM 명령과 동등한 데이터를 제공합니다.

리턴 가능한 전체 정보 목록은 580 페이지의 표 145의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_DBM_MEMORY_POOL ( ( dbpartitionnum ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMANT
- SYSADM

### 예 :

연결된 데이터베이스의 데이터베이스 관리 프로그램의 모든 데이터베이스 파티션에 대한 메모리 풀 및 현재 크기 목록을 검색합니다.

```
SELECT POOL_ID, POOL_CUR_SIZE, DBPARTITIONNUM
      FROM TABLE(SYSPROC.SNAP_GET_DBM_MEMORY_POOL())
      AS T ORDER BY DBPARTITIONNUM
```

다음은 이 쿼리의 출력 예입니다.

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

## 리턴되는 정보

표 145. SNAPDBM\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
POOL_ID	VARCHAR(14)	pool_id - 메모리 풀 ID. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 메모리 풀의 현재 크기
POOL_WATERMARK	BIGINT	pool_watermark - 메모리 풀 워터 마크 (water mark)
POOL_CONFIG_SIZE	BIGINT	pool_config_size - 메모리 풀의 구성된 크기
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.



---

## SNAPDETAILLOG 관리 뷰 및 SNAP\_GET\_DETAILLOG\_V91 테이블 함수 - 논리 데이터 그룹에서 스냅샷 정보 검색

SNAPDETAILLOG 관리 뷰 및 SNAP\_GET\_DETAILLOG\_V91 테이블 함수는 detail\_log 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.

### SNAPDETAILLOG 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 detail\_log 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPDB, SNAPDB\_MEMORY\_POOL, SNAPHADR 및 SNAPSTORAGE\_PATHS 관리 뷰와 함께 사용되는 경우, SNAPDETAILLOG 관리 뷰는 GET SNAPSHOT FOR DATABASE on database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴되는 전체 정보 목록은 584 페이지의 표 146의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPDETAILLOG 관리 뷰에 대한 SELECT 특권
- SNAPDETAILLOG 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_DETAILLOG\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

### 예 :

현재 연결된 데이터베이스에 대한 모든 데이터베이스 파티션의 로그 정보를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	FIRST_ACTIVE_LOG	LAST_ACTIVE_LOG	...
TEST	0	8	...
TEST	0	8	...
TEST	0	8	...

3개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	CURRENT_ACTIVE_LOG	CURRENT_ARCHIVE_LOG	DBPARTITIONNUM
...	0	-	0
...	0	-	1
...	0	-	2

## SNAP\_GET\_DETAILLOG\_V91 테이블 함수

SNAP\_GET\_DETAILLOG\_V91 테이블 함수는 SNAPDETAILLOG 관리 뷰와 동일한 정보를 리턴합니다.

SNAP\_GET\_DB\_V95, SNAP\_GET\_DB\_MEMORY\_POOL, SNAP\_GET\_HADR 및 SNAP\_GET\_STORAGE\_PATHS 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_DETAILLOG 테이블 함수는 GET SNAPSHOT FOR ALL DATABASES CLP 명령과 동등한 정보를 제공합니다.

리턴되는 전체 정보 목록은 584 페이지의 표 146의 내용을 참조하십시오.

## 구문

```

▶▶ SNAP_GET_DETAILLOG_V91 ( ( dbname ) [ , dbpartitionnum ] )

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수.

현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결에 사용할 수 있으며 응용프로그램에서 사용할 수 있는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_DETAILLOG\_V91 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_DETAILLOG\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### 예 :

현재 연결된 데이터베이스에 대한 데이터베이스 파티션 1의 로그 정보를 검색합니다.

```
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', 1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	FIRST_ACTIVE_LOG	LAST_ACTIVE_LOG	...
TEST	0	8	...

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	CURRENT_ACTIVE_LOG	CURRENT_ARCHIVE_LOG
...	0	-

## SNAPDETAILLOG 관리 뷰 및 SNAP\_GET\_DETAILLOG\_V91 테이블 함수 메타데이터

표 146. SNAPDETAILLOG 관리 뷰 및 SNAP\_GET\_DETAILLOG\_V91 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
FIRST_ACTIVE_LOG	BIGINT	first_active_log - 처음에 사용되는 로그 파일 번호
LAST_ACTIVE_LOG	BIGINT	last_active_log - 마지막에 사용되는 로그 파일 번호
CURRENT_ACTIVE_LOG	BIGINT	current_active_log - 현재 사용 중인 로그 파일 번호
CURRENT_ARCHIVE_LOG	BIGINT	current_archive_log - 현재 아카이브 로그 파일 번호
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPDYN\_SQL 관리 뷰 및 SNAP\_GET\_DYN\_SQL\_V95 테이블 함수 - dynsql 논리 그룹 스냅샷 정보 검색

『SNAPDYN\_SQL 관리 뷰』 및 586 페이지의 『SNAP\_GET\_DYN\_SQL\_V95 테이블 함수』에서는 dynsql 논리 데이터 그룹으로부터 스냅샷 정보를 리턴합니다.

### SNAPDYN\_SQL 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 dynsql 논리 그룹 스냅샷 정보를 검색할 수 있습니다.

이 뷰는 GET SNAPSHOT FOR DYNAMIC SQL ON database-alias CLP 명령과 동일한 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 전체 정보 목록은 588 페이지의 표 147의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPDYN\_SQL 관리 뷰에 대한 SELECT 특권
- SNAPDYN\_SQL 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_DYN\_SQL\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

현재 연결된 데이터베이스의 모든 데이터베이스 파티션에서 실행되는 동적 SQL 목록을 읽은 행 번호 순으로 검색합니다.

```
SELECT PREP_TIME_WORST, NUM_COMPILATIONS, SUBSTR(STMT_TEXT, 1, 60)
       AS STMT_TEXT, DBPARTITIONNUM
FROM SYSIBMADM.SNAPDYN_SQL ORDER BY ROWS_READ
```

이 쿼리의 출력 예는 다음과 같습니다.

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 ...

9개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... 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)
...
```

이 쿼리의 출력(계속).

```

... DBPARTITIONNUM
... -----
...          0
...          0
...          0
...          2
...          1
...          2
...          2
...          1
...          0

```

## SNAP\_GET\_DYN\_SQL\_V95 테이블 함수

SNAP\_GET\_DYN\_SQL\_V95 테이블 함수는 SNAPDYN\_SQL 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

이 테이블 함수는 GET SNAPSHOT FOR DYNAMIC SQL ON database-alias CLP 명령과 동등한 정보를 리턴합니다.

리턴될 수 있는 전체 정보 목록은 588 페이지의 표 147의 내용을 참조하십시오.

## 구문

```

▶▶ SNAP_GET_DYN_SQL_V95 (—dbname—, dbpartitionnum)

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### dbname

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. NULL 또는 빈 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 얻으십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. dbname이 NULL로 설정되어 있지 않고 dbpartitionnum이 NULL로 설정되어 있다면, -1은 내재적으로 dbpartitionnum으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, dbname만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 있는 파일이 존재하지 않으면, SNAP\_GET\_DYN\_SQL\_V95 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_DYN\_SQL\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 데이터베이스의 현재 연결된 데이터베이스 파티션에서 읽은 행 수 별로 동적 SQL 실행 목록을 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
PREP_TIME_WORST      ...
-----
0 ...
3 ...
...
4 ...
...
4 ...
...
4 ...
...
3 ...
...
4 ...
...
...
```

이 쿼리의 출력(계속).

```
... 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,

```

워크로드를 실행한 후 사용자가 테이블 함수와 함께 다음 쿼리를 사용할 수 있습니다.

```

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

```

이 테이블 함수 기반의 뷰의 경우:

```

SELECT STATS_FABRICATE_TIME, SYNC_RUNSTATS_TIME
       FROM SYSIBMADM.SNAPDYN_SQL

```

```

STATS_FABRICATE_TIME  SYNC_RUNSTATS_TIME
-----
                        5              10
                        3              20

```

2개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 147. SNAPDYN\_SQL 관리 뷰 및 SNAP\_GET\_DYN\_SQL\_V95 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
NUM_EXECUTIONS	BIGINT	num_executions - 명령문 실행
NUM_COMPILATIONS	BIGINT	num_compilations - 명령문 컴파일
PREP_TIME_WORST	BIGINT	prep_time_worst - 명령문의 가장 안좋은 준비 시간
PREP_TIME_BEST	BIGINT	prep_time_best - 명령문의 가장 좋은 준비 시간
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 삭제된 내부 행 수
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 삽입된 내부 행 수
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 갱신된 내부 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
STMT_SORTS	BIGINT	stmt_sorts - 명령문 정렬 수
SORT_OVERFLOWS	BIGINT	sort_overflows - 정렬 오버플로우 수
TOTAL_SORT_TIME	BIGINT	total_sort_time - 총 정렬 시간



표 147. SNAPDYN\_SQL 관리 뷰 및 SNAP\_GET\_DYN\_SQL\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
TOTAL_EXEC_TIME	BIGINT	total_exec_time - 경과된 명령문 실행 시간
TOTAL_EXEC_TIME_MS	BIGINT	total_exec_time - 경과된 명령문 실행 시간
TOTAL_USR_CPU_TIME	BIGINT	total_usr_cpu_time - 명령문에 대한 전체 사용자 CPU
TOTAL_USR_CPU_TIME_MS	BIGINT	total_usr_cpu_time - 명령문에 대한 전체 사용자 CPU
TOTAL_SYS_CPU_TIME	BIGINT	total_sys_cpu_time - 명령문에 대한 전체 시스템 CPU
TOTAL_SYS_CPU_TIME_MS	BIGINT	total_sys_cpu_time - 명령문에 대한 전체 시스템 CPU
STMT_TEXT	CLOB(2 M)	stmt_text - SQL문 텍스트
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
STATS_FABRICATE_TIME	BIGINT	동적 명령문의 쿼리 컴파일 중에 테이블 또는 인덱스 없이 필요한 통계를 작성하는데 시스템이 소비한 전체 시간(밀리초)
SYNC_RUNSTATS_TIME	BIGINT	동적 명령문의 쿼리 컴파일 중에 동기 통계 수집 활동에 소비된 전체 시간(밀리초)

---

## SNAPFCM 관리 뷰 및 SNAP\_GET\_FCM 테이블 함수 – fcm 논리 데이터 그룹 스냅샷 정보 검색

SNAPFCM 관리 뷰 및 SNAP\_GET\_FCM 테이블 함수는 데이터베이스 관리 프로그램 스냅샷에서 FCM(Fast Communication Manager), 특히 논리 데이터 그룹에 대한 정보를 리턴합니다.

### SNAPFCM 관리 뷰

SNAPDBM, SNAPDBM\_MEMORY\_POOL, SNAPFCM\_PART 및 SNAPSWITCHES 관리 뷰와 함께 사용되는 경우, SNAPFCM 관리 뷰는 GET SNAPSHOT FOR DBM 명령과 동등한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 592 페이지의 표 148의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPFCM 관리 뷰에 대한 SELECT 특권
- SNAPFCM 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_FCM 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### 예 :

모든 데이터베이스 파티션에서 FCM(Fast Communication Manager)의 메시지 버퍼에 대한 정보를 검색합니다.

```
SELECT BUFF_FREE, BUFF_FREE_BOTTOM, DBPARTITIONNUM  
FROM SYSIBMADM.SNAPFCM ORDER BY DBPARTITIONNUM
```

이 쿼리의 출력 예는 다음과 같습니다.

BUFF_FREE	BUFF_FREE_BOTTOM	DBPARTITIONNUM
5120	5120	5100
5120	5100	2
		0
		1

## SNAP\_GET\_FCM 테이블 함수

SNAP\_GET\_FCM 테이블 함수는 SNAPFCM 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_DBM\_V95, SNAP\_GET\_DBM\_MEMORY\_POOL, SNAP\_GET\_FCM\_PART 및 SNAP\_GET\_SWITCHES 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_FCM 테이블 함수는 GET SNAPSHOT FOR DBM 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 592 페이지의 표 148의 내용을 참조하십시오.

## 구문

```

▶▶ SNAP_GET_FCM ( ( dbpartitionnum ) )

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_FCM 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_FCM 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

데이터베이스 파티션 1에서 FCM(Fast Communication Manager)의 메시지 버퍼에 대한 정보를 검색합니다.

```
SELECT BUFF_FREE, BUFF_FREE_BOTTOM, DBPARTITIONNUM
      FROM TABLE(SYSPROC.SNAP_GET_FCM( 1 )) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

BUFF_FREE	BUFF_FREE_BOTTOM	DBPARTITIONNUM
5120	5100	1

### 리턴되는 정보

표 148. SNAPFCM 관리 뷰 및 SNAP\_GET\_FCM 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
BUFF_FREE	BIGINT	buff_free - 현재 여유 FCM 버퍼
BUFF_FREE_BOTTOM	BIGINT	buff_free_bottom - 최소 여유 FCM 버퍼
CH_FREE	BIGINT	ch_free - 현재 여유 채널
CH_FREE_BOTTOM	BIGINT	ch_free_bottom - 최소 여유 채널
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPFCM\_PART 관리 뷰 및 SNAP\_GET\_FCM\_PART 테이블 함수 - fcm\_node 논리 데이터 그룹 스냅샷 정보 검색

SNAPFCM\_PART 관리 뷰 및 SNAP\_GET\_FCM\_PART 테이블 함수는 데이터베이스 관리 프로그램 스냅샷에서 FCM(Fast Communication Manager), 특히 논리 데이터 그룹에 대한 정보를 리턴합니다.

### SNAPFCM\_PART 관리 뷰

SNAPDBM, SNAPDBM\_MEMORY\_POOL, SNAPFCM 및 SNAPSWITCHES 관리 뷰와 함께 사용되는 경우, SNAPFCM\_PART 관리 뷰는 GET SNAPSHOT FOR DBM 명령과 동등한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 595 페이지의 표 149의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPFCM\_PART 관리 뷰에 대한 SELECT 특권
- SNAPFCM\_PART 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_FCM\_PART 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

FCM(Fast Communication Manager)에 대한 버퍼의 송수신 정보를 검색합니다.

```
SELECT CONNECTION_STATUS, TOTAL_BUFFERS_SENT, TOTAL_BUFFERS_RECEIVED  
FROM SYSIBMADM.SNAPFCM_PART WHERE DBPARTITIONNUM = 0
```

이 쿼리의 출력 예는 다음과 같습니다.

CONNECTION_STATUS	TOTAL_BUFFERS_SENT	TOTAL_BUFFERS_RCVD
INACTIVE	2	1

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_FCM\_PART 테이블 함수

SNAP\_GET\_FCM\_PART 테이블 함수는 SNAPFCM\_PART 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_DBM\_V95, SNAP\_GET\_DBM\_MEMORY\_POOL, SNAP\_GET\_FCM 및 SNAP\_GET\_SWITCHES 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_FCM\_PART 테이블 함수는 GET SNAPSHOT FOR DBM 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 595 페이지의 표 149의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_FCM_PART ( dbpartitionnum ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_FCM\_PART 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_FCM\_PART 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

모든 데이터베이스 파티션에서 FCM(Fast Communication Manager)에 대한 버퍼의 송수신 정보를 검색합니다.

```
SELECT FCM_DBPARTITIONNUM, TOTAL_BUFFERS_SENT, TOTAL_BUFFERS_RCVD,  
       DBPARTITIONNUM FROM TABLE(SNAP_GET_FCM_PART()) AS T  
ORDER BY DBPARTITIONNUM
```

이 쿼리의 출력 예는 다음과 같습니다.

FCM_DBPARTITIONNUM	TOTAL_BUFFERS_SENT	TOTAL_BUFFERS_RCVD	DBPARTITIONNUM	
	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

## 리턴되는 정보

표 149. SNAPFCM\_PART 관리 뷰 및 SNAP\_GET\_FCM\_PART 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
CONNECTION_STATUS	VARCHAR(10)	connection_status - 연결 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• INACTIVE</li> <li>• ACTIVE</li> <li>• CONGESTED</li> </ul>
TOTAL_BUFFERS_SENT	BIGINT	total_buffers_sent - 전송된 전체 FCM 버퍼
TOTAL_BUFFERS_RCVD	BIGINT	total_buffers_rcvd - 수신된 전체 FCM 버퍼
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
FCM_DBPARTITIONNUM	SMALLINT	데이터를 전송하거나 데이터가 수신된 데이터베이스 파티션 번호 (TOTAL_BUFFERS_SENT 및 TOTAL_BUFFERS_RCVD 컬럼별도)

## SNAPHADR 관리 뷰 및 SNAP\_GET\_HADR 테이블 함수 - hadr 논리 데이터 그룹 스냅샷 정보 검색

SNAPHADR 관리 뷰 및 SNAP\_GET\_HADR 테이블 함수는 데이터베이스 스냅샷에서 고가용성 재해 복구(HADR), 특히 hadr 논리 데이터 그룹에 대한 정보를 리턴합니다.

### SNAPHADR 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 hadr 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다. 데이터베이스가 1차 또는 대기 고가용성 재해 복구(HADR) 데이터베이스인 경우 해당 뷰에서만 데이터를 리턴합니다.

SNAPDB, SNAPDB\_MEMORY\_POOL, SNAPDETAILLOG 및 SNAPSTORAGE\_PATHS 관리 뷰와 함께 사용되는 경우, SNAPHADR 관리 뷰는 GET SNAPSHOT FOR DATABASE ON database-alias CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 598 페이지의 표 150의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPHADR 관리 뷰에 대한 SELECT 특권
- SNAPHADR 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_HADR 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

1차 HADR 데이터베이스에서 HADR에 대한 구성 및 상태 정보를 검색합니다.

```
SELECT SUBSTR(DB_NAME, 1, 8) AS DBNAME, HADR_ROLE, HADR_STATE,
       HADR_SYNCMODE, HADR_CONNECT_STATUS
FROM SYSIBMADM.SNAPHADR
```

이 쿼리의 출력 예는 다음과 같습니다.

DBNAME	HADR_ROLE	HADR_STATE	HADR_SYNCMODE	HADR_CONNECT_STATUS
SAMPLE	PRIMARY	PEER	SYNC	CONNECTED

1개의 레코드가 선택되었습니다.



## SNAP\_GET\_HADR 테이블 함수

SNAP\_GET\_HADR 테이블 함수는 SNAPHADR 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_DB\_V95, SNAP\_GET\_DB\_MEMORY\_POOL, SNAP\_GET\_DETAILLOG\_V91 및 SNAP\_GET\_STORAGE\_PATHS 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_HADR 테이블 함수는 GET SNAPSHOT FOR ALL DATABASES CLP 명령과 동등한 정보를 제공합니다.

리턴될 수 있는 정보의 완전한 목록은 598 페이지의 표 150의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_HADR (—dbname— [ , dbpartitionnum ] ) ▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스와 동일한 인터페이스 내의 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우, *dbpartitionnum*에 -1이 내재적으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_HADR 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_HADR 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

모든 데이터베이스에서 HADR에 대한 구성 및 상태 정보를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

DBNAME	HADR_ROLE	HADR_STATE	HADR_SYNCMODE	HADR_CONNECT_STATUS
SAMPLE	PRIMARY	PEER	SYNC	CONNECTED
TESTDB	PRIMARY	DISCONNECTED	NEARSYNC	DISCONNECTED

2개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 150. SNAPHADR 관리 뷰 및 SNAP\_GET\_HADR 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
HADR_ROLE	VARCHAR(10)	hadr_role - HADR 역할. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"><li>• PRIMARY</li><li>• STANDARD</li><li>• STANDBY</li></ul>

표 150. SNAPHADR 관리 뷰 및 SNAP\_GET\_HADR 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
HADR_STATE	VARCHAR(14)	hadr_state - HADR 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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)	hadr_syncmode - HADR 동기화 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• ASYNC</li> <li>• NEARSYNC</li> <li>• SYNC</li> </ul>
HADR_CONNECT_STATUS	VARCHAR(12)	hadr_connect_status - HADR 연결 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• CONGESTED</li> <li>• CONNECTED</li> <li>• DISCONNECTED</li> </ul>
HADR_CONNECT_TIME	TIMESTAMP	hadr_connect_time - HADR 연결 시간
HADR_HEARTBEAT	INTEGER	hadr_heartbeat - HADR 하트비트
HADR_LOCAL_HOST	VARCHAR(255)	hadr_local_host - HADR 로컬 호스트
HADR_LOCAL_SERVICE	VARCHAR(40)	hadr_local_service - HADR 로컬 서비스
HADR_REMOTE_HOST	VARCHAR(255)	hadr_remote_host - HADR 리모트 호스트
HADR_REMOTE_SERVICE	VARCHAR(40)	hadr_remote_service - HADR 리모트 서비스
HADR_REMOTE_INSTANCE	VARCHAR(128)	hadr_remote_instance - HADR 리모트 인스턴스
HADR_TIMEOUT	BIGINT	hadr_timeout - HADR 시간종료
HADR_PRIMARY_LOG_FILE	VARCHAR(255)	hadr_primary_log_file - HADR 1차 로그 파일
HADR_PRIMARY_LOG_PAGE	BIGINT	hadr_primary_log_page - HADR 1차 로그 페이지
HADR_PRIMARY_LOG_LSN	BIGINT	hadr_primary_log_lsn - HADR 1차 로그 LSN

표 150. SNAPHADR 관리 뷰 및 SNAP\_GET\_HADR 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
HADR_STANDBY_LOG_FILE	VARCHAR(255)	hadr_standby_log_file - HADR 대기 로그 파일
HADR_STANDBY_LOG_PAGE	BIGINT	hadr_standby_log_page - HADR 대기 로그 페이지
HADR_STANDBY_LOG_LSN	BIGINT	hadr_standby_log_lsn - HADR 대기 로그 LSN
HADR_LOG_GAP	BIGINT	hadr_log_gap - HADR 로그 갭
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPLOCK 관리 뷰 및 SNAP\_GET\_LOCK 테이블 함수 - 잠금 논리 데이터 그룹 스냅샷 정보 검색

SNAPLOCK 관리 뷰 및 SNAP\_GET\_LOCK 테이블 함수는 잠금, 특히 잠금 논리 데이터 그룹에 대한 스냅샷 정보를 리턴합니다.

### SNAPLOCK 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 잠금 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPLOCKWAIT 관리 뷰와 같이 사용하여 SNAPLOCK 관리 뷰는 GET SNAPSHOT FOR LOCKS ON database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 603 페이지의 표 151의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPLOCK 관리 뷰에 대한 SELECT 특권
- SNAPLOCK 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_LOCK 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

예 :

현재 연결된 데이터베이스의 데이터베이스 파티션 0에 대한 잠금 정보를 검색합니다.

```
SELECT AGENT_ID, LOCK_OBJECT_TYPE, LOCK_MODE, LOCK_STATUS
FROM SYSIBMADM.SNAPLOCK WHERE DBPARTITIONNUM = 0
```

다음은 이 쿼리의 출력 예입니다.

AGENT_ID	LOCK_OBJECT_TYPE	LOCK_MODE	LOCK_STATUS
7	TABLE	IX	GRNT

1개의 레코드가 선택되었습니다.

### SNAP\_GET\_LOCK 테이블 함수

SNAP\_GET\_LOCK 테이블 함수는 SNAPLOCK 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

SNAP\_GET\_LOCKWAIT 테이블 함수와 함께 사용되는 SNAP\_GET\_LOCK 테이블 함수는 GET SNAPSHOT FOR LOCKS ON database-alias CLP 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 603 페이지의 표 151의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_LOCK ( ( dbname [ , dbpartitionnum ] ) )
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

*dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 널 (NULL) 값 또는 비어 있는 문자열을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형의 파일이 없는 경우 SNAP\_GET\_LOCK 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 작성합니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_LOCK 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

### 예 :

현재 연결된 데이터베이스의 현재 데이터베이스 파티션에 대한 잠금 정보를 검색합니다.

```
SELECT AGENT_ID, LOCK_OBJECT_TYPE, LOCK_MODE, LOCK_STATUS  
FROM TABLE(SNAP_GET_LOCK(' ', -1)) as T
```

다음은 이 쿼리의 출력 예입니다.

AGENT_ID	LOCK_OBJECT_TYPE	LOCK_MODE	LOCK_STATUS
680	INTERNALV_LOCK	S	GRNT
680	INTERNALP_LOCK	S	GRNT

2개의 레코드가 선택되었습니다.

## 리턴된 정보

표 151. SNAPLOCK 관리 뷰 및 SNAP\_GET\_LOCK 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
TAB_FILE_ID	BIGINT	table_file_id - 테이블 파일 ID
LOCK_OBJECT_TYPE	VARCHAR(18)	lock_object_type - 대기하는 잠금 오브젝트 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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>

표 151. SNAPLOCK 관리 뷰 및 SNAP\_GET\_LOCK 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_MODE	VARCHAR(10)	lock_mode - 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (잠금이 없는 경우)</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 - 잠금 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• CONV</li> <li>• GRNT</li> </ul>
LOCK_ESCALATION	SMALLINT	lock_escalation - 잠금 에스컬레이션
TABNAME	VARCHAR(128)	table_name - 테이블 이름
TABSCHEMA	VARCHAR(128)	table_schema - 테이블 스키마 이름
TBSP_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름



표 151. SNAPLOCK 관리 뷰 및 SNAP\_GET\_LOCK 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_ATTRIBUTES	VARCHAR (128)	lock_attributes - 잠금 속성. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴합니다. 잠금이 없을 경우 텍스트 ID는 NONE이며, 그렇지 않을 경우 다음의 조합으로 '+' 부호에 의해 분리됩니다. <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 - 잠금 개수
LOCK_CURRENT_MODE	VARCHAR(10)	lock_current_mode - 변환 이전의 원래 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (잠금이 없는 경우)</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 - 잠금 보유 개수
LOCK_NAME	VARCHAR(32)	lock_name - 잠금 이름
LOCK_RELEASE_FLAGS	BIGINT	lock_release_flags - 잠금 릴리스 플래그
DATA_PARTITION_ID	INTEGER	data_partition_id - 데이터 파티션 ID. 파티션되지 않은 테이블의 경우 이 요소는 NULL입니다.
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

---

## SNAPLOCKWAIT 관리 뷰 및 SNAP\_GET\_LOCKWAIT 테이블 함수 - 잠금 대기 논리 데이터 그룹 스냅샷 정보 검색

SNAPLOCKWAIT 관리 뷰 및 SNAP\_GET\_LOCKWAIT 테이블 함수는 잠금 대기 특히 잠금 대기 논리 데이터 그룹에 대한 스냅샷 정보를 리턴합니다.

### SNAPLOCKWAIT 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 잠금 대기 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPLOCK 관리 뷰와 같이 사용하여 SNAPLOCKWAIT 관리 뷰는 GET SNAPSHOT FOR LOCKS ON database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 609 페이지의 표 152의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPLOCKWAIT 관리 뷰에 대한 SELECT 특권
- SNAPLOCKWAIT 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_LOCKWAIT 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

### 예 :

현재 연결된 데이터베이스에 대한 데이터베이스 파티션 0에서 잠금 대기 정보를 검색합니다.

```
SELECT AGENT_ID, LOCK_MODE, LOCK_OBJECT_TYPE, AGENT_ID_HOLDING_LK,  
       LOCK_MODE_REQUESTED FROM SYSIBMADM.SNAPLOCKWAIT  
WHERE DBPARTITIONNUM = 0
```

다음은 이 쿼리의 출력 예입니다.

```
AGENT_ID      LOCK_MODE LOCK_OBJECT_TYPE ...  
-----  
          7 IX          TABLE          ...
```

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... AGENT_ID_HOLDING_LK  LOCK_MODE_REQUESTED  
... -----  
...                   12 IS
```

## SNAP\_GET\_LOCKWAIT 테이블 함수

SNAP\_GET\_LOCKWAIT 테이블 함수는 SNAPLOCKWAIT 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

SNAP\_GET\_LOCK 테이블 함수와 함께 사용되는 SNAP\_GET\_LOCKWAIT 테이블 함수는 GET SNAPSHOT FOR LOCKS ON database-alias CLP 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 609 페이지의 표 152의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_LOCKWAIT (—dbname— [ , dbpartitionnum ] ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 널 (NULL) 값 또는 비어 있는 문자열을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로

설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형의 파일이 없는 경우 SNAP\_GET\_LOCKWAIT 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 작성합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_LOCKWAIT 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

현재 연결된 데이터베이스에 대한 현재 데이터베이스 파티션에서 잠금 대기 정보를 검색합니다.

```
SELECT AGENT_ID, LOCK_MODE, LOCK_OBJECT_TYPE, AGENT_ID_HOLDING_LK,
       LOCK_MODE_REQUESTED FROM TABLE(SNAP_GET_LOCKWAIT('',-1)) AS T
```

다음은 이 쿼리의 출력 예입니다.

```
AGENT_ID      LOCK_MODE  LOCK_OBJECT_TYPE  ...
-----
          12 X          ROW_LOCK          ...
```

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... AGENT_ID_HOLDING_LK  LOCK_MODE_REQUESTED
... -----
...                      7 X
```

## 사용법 참고

잠금 대기 정보를 확인하려면 우선 데이터베이스 관리 프로그램 구성에서 디폴트 LOCK 모니터 스위치를 켜야 합니다. 변경사항을 바로 적용하려면 CLP를 사용하여 인스턴스에 명시적으로 첨부한 다음 CLP 명령을 실행하십시오.

```
UPDATE DATABASE MANAGER CONFIGURATION CLP USING DFT_MON_LOCK ON
```

디폴트 설정은 ADMIN\_CMD 스토어드 프로시저를 통해서도 켤 수 있습니다. 예를 들어, 다음과 같습니다.

```
CALL SYSPROC.ADMIN_CMD('update dbm cfg using DFT_MON_LOCK ON')
```

ADMIN\_CMD 스토어드 프로시저를 사용하거나 이전에 인스턴스에 접속하지 않은 상태로 clp 명령을 실행하는 경우 변경사항을 적용하기 전에 인스턴스를 재사용해야 합니다.

## 리턴된 정보

표 152. SNAPLOCKWAIT 관리 뷰 및 SNAP\_GET\_LOCKWAIT 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
SUBSECTION_NUMBER	BIGINT	ss_number - 서브섹션 번호
LOCK_MODE	VARCHAR(10)	lock_mode - 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (잠금이 없는 경우)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>

표 152. SNAPLOCKWAIT 관리 뷰 및 SNAP\_GET\_LOCKWAIT 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_OBJECT_TYPE	VARCHAR(18)	lock_object_type - 대기하는 잠금 오브젝트 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 잠금을 보유한 에이전트 ID
LOCK_WAIT_START_TIME	TIMESTAMP	lock_wait_start_time - 잠금 대기 시작 시간소인

표 152. SNAPLOCKWAIT 관리 뷰 및 SNAP\_GET\_LOCKWAIT 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_MODE_REQUESTED	VARCHAR(10)	lock_mode_requested - 요청된 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (잠금이 없는 경우)</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 - 잠금 에스컬레이션
TABNAME	VARCHAR(128)	table_name - 테이블 이름
TABSCHEMA	VARCHAR(128)	table_schema - 테이블 스키마 이름
TBSP_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름
APPL_ID_HOLDING_LK	VARCHAR(128)	appl_id_holding_lk - 잠금을 보유한 응용프로그램 ID
LOCK_ATTRIBUTES	VARCHAR(128)	lock_attributes - 잠금 속성. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴합니다. 잠금이 없을 경우 텍스트 ID는 NONE이며, 그렇지 않을 경우 다음의 조합으로 '+' 부호에 의해 분리됩니다. <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>

표 152. SNAPLOCKWAIT 관리 뷰 및 SNAP\_GET\_LOCKWAIT 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_CURRENT_MODE	VARCHAR(10)	lock_current_mode - 변환 이전의 원래 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (잠금이 없는 경우)</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 - 잠금 이름
LOCK_RELEASE_FLAGS	BIGINT	lock_release_flags - 잠금 릴리스 플래그
DATA_PARTITION_ID	INTEGER	data_partition_id - 데이터 파티션 ID. 파티션되지 않은 테이블의 경우 이 요소는 NULL입니다.
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPSTMT 관리 뷰 및 SNAP\_GET\_STMT 테이블 함수 - 명령문 스냅샷 정보 검색

SNAPSTMT 관리 뷰 및 SNAP\_GET\_STMT 테이블 함수는 응용프로그램 스냅샷에서 SQL 또는 XQuery문에 대한 정보를 리턴합니다.

### SNAPSTMT 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 명령문 스냅샷 정보를 검색할 수 있습니다.

SNAPAGENT, SNAPAGENT\_MEMORY\_POOL, SNAPAPPL, SNAPAPPL\_INFO 및 SNAPSUBSECTION 관리 뷰와 함께 사용되는 경우, SNAPSTMT 관리 뷰는 GET SNAPSHOT FOR APPLICATIONS on database-alias CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

스키마는 SYSIBMADM입니다.



리턴될 수 있는 정보의 완전한 목록은 615 페이지의 표 153의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPSTMT 관리 뷰에 대한 SELECT 특권
- SNAPSTMT 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_STMT 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

현재 연결된 단일 파티션 데이터베이스에서 실행되는 명령문에 대해 읽은 행, 쓴 행 및 수행된 조작을 검색합니다.

```
SELECT SUBSTR(STMT_TEXT,1,30) AS STMT_TEXT, ROWS_READ, ROWS_WRITTEN,  
       STMT_OPERATION FROM SYSIBMADM.SNAPSTMT
```

다음은 이 쿼리의 출력 예입니다.

STMT_TEXT	ROWS_READ	ROWS_WRITTEN	STMT_OPERATION
-----...- - - - -	-----...- - - - -	-----...- - - - -	-----
-		0	0 FETCH
-		0	0 STATIC_COMMIT

2개의 레코드가 선택되었습니다.

## SNAP\_GET\_STMT 테이블 함수

SNAP\_GET\_STMT 테이블 함수는 SNAPSTMT 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_AGENT, SNAP\_GET\_AGENT\_MEMORY\_POOL, SNAP\_GET\_APPL\_V95, SNAP\_GET\_APPL\_INFO\_V95 및 SNAP\_GET\_SUBSECTION 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_STMT

테이블 함수는 GET SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

리턴될 수 있는 정보의 완전한 목록은 615 페이지의 표 153의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_STMT ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_STMT 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_STMT 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

예 :

현재 연결된 데이터베이스의 현재 데이터베이스 파티션에서 실행되는 명령문에 대해 읽은 행, 쓴 행 및 수행된 조작을 검색합니다.

```
SELECT SUBSTR(STMT_TEXT,1,30) AS STMT_TEXT, ROWS_READ,
       ROWS_WRITTEN, STMT_OPERATION FROM TABLE(SNAP_GET_STMT('-',-1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...
	...	...

5개의 레코드가 선택되었습니다.

...

이 쿼리의 출력(계속).

ROWS_WRITTEN	STMT_OPERATION
-----	-----
...	0 EXECUTE_IMMEDIATE
...	0 FETCH
...	0 NONE
...	0 NONE
...	1 EXECUTE_IMMEDIATE
...	

## 리턴되는 정보

표 153. SNAPSTMT 관리 뷰 및 SNAP\_GET\_STMT 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
NUM_AGENTS	BIGINT	num_agents - 명령문에서 작업하는 에이전트 수
AGENTS_TOP	BIGINT	agents_top - 작성된 에이전트 수

표 153. SNAPSTMT 관리 뷰 및 SNAP\_GET\_STMT 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
STMT_TYPE	VARCHAR(20)	stmt_type - 명령문 유형. 이 인터페이스는 sqlmon.h의 정의를 기본으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• DYNAMIC</li> <li>• NON_STMT</li> <li>• STATIC</li> <li>• STMT_TYPE_UNKNOWN</li> </ul>
STMT_OPERATION	VARCHAR(20)	stmt_operation/operation - 명령문 조작. 이 인터페이스는 sqlmon.h의 정의를 기본으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 섹션 번호

표 153. SNAPSTMT 관리 뷰 및 SNAP\_GET\_STMT 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
QUERY_COST_ESTIMATE	BIGINT	query_cost_estimate - 추정 쿼리 비용
QUERY_CARD_ESTIMATE	BIGINT	query_card_estimate - 추정 쿼리 행 수
DEGREE_PARALLELISM	BIGINT	degree_parallelism - 병렬 처리 수준
STMT_SORTS	BIGINT	stmt_sorts - 명령문 정렬 수
TOTAL_SORT_TIME	BIGINT	total_sort_time - 총 정렬 시간
SORT_OVERFLOWS	BIGINT	sort_overflows - 정렬 오버플로우 수
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 삭제된 내부 행 수
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 갱신된 내부 행 수
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 삽입된 내부 행 수
FETCH_COUNT	BIGINT	fetch_count - 성공한 페치 수
STMT_START	TIMESTAMP	stmt_start - 명령문 조작 시작 시간소인
STMT_STOP	TIMESTAMP	stmt_stop - 명령문 조작 중지 시간소인
STMT_USR_CPU_TIME_S	BIGINT	stmt_usr_cpu_time - 명령문이 사용한 사용자 CPU 시간
STMT_USR_CPU_TIME_MS	BIGINT	stmt_usr_cpu_time - 명령문이 사용한 사용자 CPU 시간
STMT_SYS_CPU_TIME_S	BIGINT	stmt_sys_cpu_time - 명령문이 사용한 시스템 CPU 시간
STMT_SYS_CPU_TIME_MS	BIGINT	stmt_sys_cpu_time - 명령문이 사용한 시스템 CPU 시간
STMT_ELAPSED_TIME_S	BIGINT	stmt_elapsed_time - 가장 최근의 명령문 경과 시간
STMT_ELAPSED_TIME_MS	BIGINT	stmt_elapsed_time - 가장 최근의 명령문 경과 시간
BLOCKING_CURSOR	SMALLINT	blocking_cursor - 블로킹 커서
STMT_NODE_NUMBER	SMALLINT	stmt_node_number - 명령문 노드
CURSOR_NAME	VARCHAR (128)	cursor_name - 커서 이름
CREATOR	VARCHAR (128)	creator - 응용프로그램 작성자
PACKAGE_NAME	VARCHAR (128)	package_name - 패키지 이름
STMT_TEXT	CLOB(16 M)	stmt_text - SQL문 텍스트
CONSISTENCY_TOKEN	VARCHAR (128)	consistency_token - 패키지 일관성 토큰
PACKAGE_VERSION_ID	VARCHAR (128)	package_version_id - 패키지 버전
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수

표 153. SNAPSTMT 관리 뷰 및 SNAP\_GET\_STMT 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수 모니터 요소
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수 모니터 요소
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPSTORAGE\_PATHS 관리 뷰 및 SNAP\_GET\_STORAGE\_PATHS\_V97 테이블 함수 - 자동 스토리지 경로 정보 검색

SNAPSTORAGE\_PATHS 관리 뷰 및 SNAP\_GET\_STORAGE\_PATHS\_V97 테이블 함수는 각 스토리지 경로에 대한 파일 시스템 정보를 포함하여 데이터베이스에 대한 자동 스토리지 경로 목록 특히 db\_storage\_group 논리 데이터 그룹에서 리턴합니다.

### SNAPSTORAGE\_PATHS 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 자동 스토리지 경로 정보를 검색할 수 있습니다.

SNAPDB, SNAPDETAILLOG, SNAPHADR 및 SNAPDB\_MEMORY\_POOL 관리 뷰와 같이 사용하여 SNAPSTORAGE\_PATHS 관리 뷰는 GET SNAPSHOT FOR DATABASE ON database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 621 페이지의 표 154의 내용을 참조하십시오.

## 권한 부여

- SYSMON 권한
- SNAPSTORAGE\_PATHS 관리 뷰의 SELECT 또는 CONTROL 특권 및 SNAP\_GET\_STORAGE\_PATHS\_V97 테이블 함수의 EXECUTE 특권입니다.

## 예 :

현재 연결된 한 개 파티션의 데이터베이스에 대한 스토리지 경로를 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

```
DB_NAME  DB_STORAGE_PATH  HOSTNAME
-----  -
STOPATH  d:                      JESSICAE
```

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_STORAGE\_PATHS\_V97 테이블 함수

SNAP\_GET\_STORAGE\_PATHS\_V97 테이블 함수는 SNAPSTORAGE\_PATHS 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

SNAP\_GET\_DB\_V95, SNAP\_GET\_DETAILLOG\_V91, SNAP\_GET\_HADR 및 SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수와 함께 사용되는 SNAP\_GET\_STORAGE\_PATHS\_V97 테이블 함수는 GET SNAPSHOT FOR ALL DATABASES CLP 명령과 동등한 정보를 제공합니다.

리턴될 수 있는 정보의 완전한 목록은 621 페이지의 표 154의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_STORAGE_PATHS_V97 ( ( dbname [ , dbpartitionnum ] ) )
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

*dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지

정하는 VARCHAR

(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우 -1이 *dbpartitionnum*에 대해 내재적으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_STORAGE\_PATHS\_V97 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대해 스냅샷을 작성합니다.

### 권한 부여

- SYSMON 권한
- SNAP\_GET\_STORAGE\_PATHS\_V97 테이블 함수의 EXECUTE 특권

### 예:

모든 활성 데이터베이스에 대한 스토리지 경로 정보를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
DB_NAME  DB_STORAGE_PATH
-----  -
STOPATH  /home/jessicae/sdb
MYDB     /home/jessicae/mdb

2 record(s) selected
```



## 리턴되는 정보

파일 시스템 정보가 리턴되도록 하려면 BUFFERPOOL 모니터 스위치를 켜야 합니다.

표 154. SNAPSTORAGE\_PATHS 관리 뷰 및 SNAP\_GET\_STORAGE\_PATHS\_V97 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
DB_STORAGE_PATH	VARCHAR (256)	db_storage_path - 자동 스토리지 경로
DB_STORAGE_PATH_WITH_DPE	VARCHAR (256)	평가되지 않은 데이터베이스 파티션 표현식(DPE)이 포함된 자동 스토리지 경로입니다. 스토리지 경로가 DPE를 포함하지 않으면 NULL이 리턴됩니다.
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
DB_STORAGE_PATH_STATE	VARCHAR(16)	자동 스토리지 경로 상태입니다(값은 현재 『IN_USE』, 『NOT_IN_USE』, 『DROP_PENDING』 중 하나일 수 있습니다).
FS_ID	VARCHAR(22)	fs_id - 고유 파일 시스템 ID 번호
FS_TOTAL_SIZE	BIGINT	fs_total_size - 파일 시스템의 전체 크기
FS_USED_SIZE	BIGINT	fs_used_size - 파일 시스템에서 사용된 스페이스 용량
STO_PATH_FREE_SIZE	BIGINT	sto_path_free_sz - 자동 스토리지 경로 여유 공간

## SNAPSUBSECTION 관리 뷰 및 SNAP\_GET\_SUBSECTION 테이블 함수 - 서브섹션 논리 모니터 그룹 스냅샷 정보 검색

SNAPSUBSECTION 관리 뷰 및 SNAP\_GET\_SUBSECTION 테이블 함수는 응용프로그램 서브섹션, 즉 서브섹션 논리 모니터 그룹화에 대한 정보를 리턴합니다.

### SNAPSUBSECTION 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 서브섹션 논리 모니터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPAGENT, SNAPAGENT\_MEMORY\_POOL, SNAPAPPL, SNAPAPPL\_INFO 및 SNAPSTMT 관리 뷰와 함께 사용되는 경우, SNAPSUBSECTION 관리 뷰는 GET SNAPSHOT FOR APPLICATIONS on database-alias CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 624 페이지의 표 155의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPSUBSECTION 관리 뷰에 대한 SELECT 특권
- SNAPSUBSECTION 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_SUBSECTION 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

모든 데이터베이스 파티션에서 실행 중인 서브섹션의 상태를 가져옵니다.

```
SELECT DB_NAME, STMT_TEXT, SS_STATUS, DBPARTITIONNUM
      FROM SYSIBMADM.SNAPSUBSECTION
      ORDER BY DB_NAME, SS_STATUS, DBPARTITIONNUM
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	STMT_TEXT	SS_STATUS	DBPARTITIONNUM
SAMPLE	select * from EMPLOYEE	EXEC	0
SAMPLE	select * from EMPLOYEE	EXEC	1

## SNAP\_GET\_SUBSECTION 테이블 함수

SNAP\_GET\_SUBSECTION 테이블 함수는 SNAPSUBSECTION 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

리턴 가능한 전체 정보 목록은 624 페이지의 표 155의 내용을 참조하십시오.

SNAP\_GET\_AGENT, SNAP\_GET\_AGENT\_MEMORY\_POOL, SNAP\_GET\_APPL\_V95, SNAP\_GET\_APPL\_INFO\_V95 및 SNAP\_GET\_STMT 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_SUBSECTION 테이블 함수는 GET SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

## 구문

```
▶▶ SNAP_GET_SUBSECTION ( (dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스와 동일한 인터페이스 내의 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_SUBSECTION 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_SUBSECTION 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYDMAINT
- SYSADM

예 :

모든 데이터베이스 파티션에서 실행 중인 서브섹션의 상태를 가져옵니다.

```
SELECT DB_NAME, STMT_TEXT, SS_STATUS, DBPARTITIONNUM
      FROM TABLE(SYSPROC.SNAP_GET_SUBSECTION( '', 0 )) as T
      ORDER BY DB_NAME, SS_STATUS, DBPARTITIONNUM
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	STMT_TEXT	SS_STATUS	DBPARTITIONNUM
SAMPLE	select * from EMPLOYEE	EXEC	0
SAMPLE	select * from EMPLOYEE	EXEC	1

## 리턴되는 정보

표 155. SNAPSUBSECTION 관리 뷰 및 SNAP\_GET\_SUBSECTION 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
STMT_TEXT	CLOB(16 M)	stmt_text - SQL문 텍스트
SS_EXEC_TIME	BIGINT	ss_exec_time - 서브섹션 실행 경과 시간
TQ_TOT_SEND_SPILLS	BIGINT	tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총계
TQ_CUR_SEND_SPILLS	BIGINT	tq_cur_send_spills - 오버플로우된 현재 테이블 큐 버퍼 수
TQ_MAX_SEND_SPILLS	BIGINT	tq_max_send_spills - 최대 테이블 큐 버퍼 오버플로우 수
TQ_ROWS_READ	BIGINT	tq_rows_read - 테이블 큐에서 읽은 행 수
TQ_ROWS_WRITTEN	BIGINT	tq_rows_written - 테이블 큐에 기록된 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
SS_USR_CPU_TIME_S	BIGINT	ss_usr_cpu_time - 서브섹션이 사용한 사용자 CPU 시간
SS_USR_CPU_TIME_MS	BIGINT	ss_usr_cpu_time - 서브섹션이 사용한 사용자 CPU 시간

표 155. SNAPSUBSECTION 관리 뷰 및 SNAP\_GET\_SUBSECTION 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SS_SYS_CPU_TIME_S	BIGINT	ss_sys_cpu_time - 서버섹션이 사용한 시스템 CPU 시간
SS_SYS_CPU_TIME_MS	BIGINT	ss_sys_cpu_time - 서버섹션이 사용한 시스템 CPU 시간
SS_NUMBER	INTEGER	ss_number - 서버섹션 번호
SS_STATUS	VARCHAR(20)	ss_status - 서버섹션 상태. 이 인터페이스는 sqlmon.h의 정의를 기본으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 서버섹션 노드 번호
TQ_NODE_WAITED_FOR	SMALLINT	tq_node_waited_for - 테이블 큐의 노드를 기다림
TQ_WAIT_FOR_ANY	INTEGER	tq_wait_for_any - 테이블 큐에 전송할 노드를 기다리는 중
TQ_ID_WAITING_ON	INTEGER	tq_id_waiting_on - 테이블 큐의 노드에서 대기
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPSWITCHES 관리 뷰 및 SNAP\_GET\_SWITCHES 테이블 함수 - 데이터베이스 스냅샷 스위치 상태 정보 검색

SNAPSWITCHES 관리 뷰 및 SNAP\_GET\_SWITCHES 테이블 함수는 데이터베이스 스냅샷 전환 상태에 대한 정보를 리턴합니다.

### SNAPSWITCHES 관리 뷰

이 뷰에서는 GET DBM MONITOR SWITCHES CLP 명령과 동등한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 628 페이지의 표 156의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPSWITCHES 관리 뷰에 대한 SELECT 특권
- SNAPSWITCHES 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_SWITCHES 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

모든 데이터베이스 파티션에 대한 DBM 모니터 전환 상태 정보를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
UOW_SW_STATE STATEMENT_SW_STATE TABLE_SW_STATE BUFFPOOL_SW_STATE ...  
-----  
                0                0                0                0 ...  
                0                0                0                0 ...  
                0                0                0                0 ...  
                ...
```

3개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... 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 테이블 함수

SNAP\_GET\_SWITCHES 테이블 함수는 SNAPSWITCHES 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

이 테이블 함수는 GET DBM MONITOR SWITCHES CLP 명령과 동등한 데이터를 제공합니다.

리턴될 수 있는 정보의 완전한 목록은 628 페이지의 표 156의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_SWITCHES ( dbpartitionnum )
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_SWITCHES 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_SWITCHES 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예:

현재 데이터베이스 파티션에 대한 DBM 모니터 전환 상태 정보를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
UOW_SW_STATE STATEMENT_SW_STATE TABLE_SW_STATE...
-----
1 1 1...
```

1개의 레코드가 선택되었습니다.  
...

이 쿼리의 출력(계속).

```
... BUFFPOOL_SW_STATE LOCK_SW_STATE SORT_SW_STATE TIMESTAMP_SW_STATE
... -----
... 1 1 0 1
```

## 리턴되는 정보

표 156. SNAPSWITCHES 관리 뷰 및 SNAP\_GET\_SWITCHES 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
UOW_SW_STATE	SMALLINT	작업 단위(UOW) 모니터 기록 전환 상태 (0 또는 1).
UOW_SW_TIME	TIMESTAMP	작업 단위(UOW) 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간.
STATEMENT_SW_STATE	SMALLINT	SQL문 모니터 기록 전환 상태(0 또는 1).
STATEMENT_SW_TIME	TIMESTAMP	SQL문 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간.
TABLE_SW_STATE	SMALLINT	테이블 활동 모니터 기록 전환 상태(0 또는 1).
TABLE_SW_TIME	TIMESTAMP	테이블 활동 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간
BUFFPOOL_SW_STATE	SMALLINT	버퍼 풀 활동 모니터 기록 전환 상태(0 또는 1)
BUFFPOOL_SW_TIME	TIMESTAMP	버퍼 풀 활동 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간
LOCK_SW_STATE	SMALLINT	잠금 모니터 기록 전환 상태(0 또는 1)
LOCK_SW_TIME	TIMESTAMP	잠금 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간
SORT_SW_STATE	SMALLINT	정렬 모니터 기록 전환 상태(0 또는 1)
SORT_SW_TIME	TIMESTAMP	정렬 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간
TIMESTAMP_SW_STATE	SMALLINT	시간소인 모니터 기록 전환 상태(0 또는 1)



표 156. SNAPSWITCHES 관리 뷰 및 SNAP\_GET\_SWITCHES 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
TIMESTAMP_SW_TIME	TIMESTAMP	시간소인 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPTAB 관리 뷰 및 SNAP\_GET\_TAB\_V91 테이블 함수 - 테이블 논리 데이터 그룹 스냅샷 정보 검색

SNAPTAB 관리 뷰 및 SNAP\_GET\_TAB\_V91 테이블 함수는 테이블 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.

### SNAPTAB 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 테이블 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPTAB\_REORG 관리 뷰와 함께 사용될 경우, SNAPTAB 관리 뷰는 GET SNAPSHOT FOR TABLES ON database-alias CLP 명령과 동일한 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 631 페이지의 표 157의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPTAB 관리 뷰에 대한 SELECT 특권
- SNAPTAB 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_TAB\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

모든 활성 테이블에 대한 스키마 및 이름을 검색합니다.

```
SELECT SUBSTR(TABSHEMA,1,8), SUBSTR(TABNAME,1,15) AS TABNAME, TAB_TYPE,
       DBPARTITIONNUM FROM SYSIBMADM.SNAPTAB
```

다음은 이 쿼리의 출력 예입니다.

TABSHEMA	TABNAME	TAB_TYPE	DBPARTITIONNUM
SYSTOOLS	HMON_ATM_INFO	USER_TABLE	0

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_TAB\_V91 테이블 함수

SNAP\_GET\_TAB\_V91 테이블 함수는 SNAPTAB 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_TAB\_REORG 테이블 함수와 함께 사용될 경우, SNAP\_GET\_TAB\_V91 테이블 함수는 GET SNAPSHOT FOR TABLES ON database-alias CLP 명령과 동일한 정보를 리턴합니다.

리턴 가능한 전체 정보 목록은 631 페이지의 표 157의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_TAB_V91( (—dbname—) [ , dbpartitionnum ] ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. NULL 또는 빈 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 얻으십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이

입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_TAB\_V91 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_TAB\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

현재 연결된 데이터베이스에 대한 집계 보기로 활성 테이블 목록을 검색합니다.

```
SELECT SUBSTR(TABSCHEMA,1,8) AS TABSCHEMA, SUBSTR(TABNAME,1,15) AS TABNAME,
       TAB_TYPE, DBPARTITIONNUM FROM TABLE(SNAP_GET_TAB(' ', -2)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

TABSCHEMA	TABNAME	TAB_TYPE	DBPARTITIONNUM
SYSTOOLS	HMON_ATM_INFO	USER_TABLE	-
JESSICAE	EMPLOYEE	USER_TABLE	-

## 리턴되는 정보

표 157. SNAPTAB 관리 뷰 및 SNAP\_GET\_TAB\_V91 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TABSCHEMA	VARCHAR (128)	table_schema - 테이블 스키마 이름
TABNAME	VARCHAR (128)	table_name - 테이블 이름

표 157. SNAPTAB 관리 뷰 및 SNAP\_GET\_TAB\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TAB_FILE_ID	BIGINT	table_file_id - 테이블 파일 ID
TAB_TYPE	VARCHAR(14)	table_type - 테이블 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 데이터 오브젝트 페이지 수
INDEX_OBJECT_PAGES	BIGINT	index_object_pages - 인덱스 오브젝트 페이지 수
LOB_OBJECT_PAGES	BIGINT	lob_object_pages - LOB 오브젝트 페이지 수
LONG_OBJECT_PAGES	BIGINT	long_object_pages - Long 오브젝트 페이지 수
XDA_OBJECT_PAGES	BIGINT	xda_object_pages - XDA 오브젝트 페이지 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
OVERFLOW_ACCESSES	BIGINT	overflow_accesses - 오버플로우된 레코드에 액세스
PAGE_REORGS	BIGINT	page_reorgs - 페이지 재구성
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
DATA_PARTITION_ID	INTEGER	data_partition_id - 데이터 파티션 ID. 파티션되지 않은 테이블의 경우 이 요소는 널(NULL)이 됩니다.

## SNAPTAB\_REORG 관리 뷰 및 SNAP\_GET\_TAB\_REORG 테이블 함수 - 테이블 재구성 스냅샷 정보 검색

SNAPTAB\_REORG 관리 뷰 및 SNAP\_GET\_TAB\_REORG 테이블 함수는 테이블 재구성 정보를 리턴합니다. 재구성된 테이블이 없다면 0행이 리턴됩니다.

### SNAPTAB\_REORG 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 테이블 재구성 스냅샷 정보를 검색할 수 있습니다.

SNAPTAB 관리 뷰를 사용한 SNAPTAB\_REORG 관리 뷰는 GET SNAPSHOT FOR TABLES ON database-alias CLP 명령과 동일한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 전체 정보 목록은 636 페이지의 표 158의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPTAB\_REORG 관리 뷰에 대한 SELECT 특권
- SNAPTAB\_REORG 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_TAB\_REORG 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 데이터베이스에서 모든 데이터베이스 파티션에 대한 재구성 조작에 관한 세 부사항을 선택합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

TAB_NAME	TAB_SCHEMA	REORG_PHASE	...
EMPLOYEE	DBUSER	REPLACE	...
EMPLOYEE	DBUSER	REPLACE	...
EMPLOYEE	DBUSER	REPLACE	...
			...

3개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```

... 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

```

재구성 조작에 관한 모든 정보를 선택하여 SNAPTAB\_REORG 관리 뷰의 다차원적으로 클러스터된(MDC) 테이블에서부터 Extent를 재개합니다.

```
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 테이블 함수

SNAP\_GET\_TAB\_REORG 테이블 함수는 SNAPTAB\_REORG 관리 뷰와 동일한 정보를 리턴하지만 특정 데이터베이스 파티션에 대한 특정 데이터베이스의 정보를 검색할 수 있으며, 모든 데이터베이스 파티션을 집계할 수 있습니다.

SNAP\_GET\_TAB 테이블 함수를 사용하여 SNAP\_GET\_TAB\_REORG 테이블 함수는 GET SNAPSHOT FOR TABLES ON database-alias CLP 명령과 동일한 데이터를 제공합니다.

리턴될 수 있는 전체 정보 목록은 636 페이지의 표 158의 내용을 참조하십시오.

## 구문

```

▶▶ SNAP_GET_TAB_REORG ( ( dbname [ , dbpartitionnum ] ) )

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴 시 "Indirect" 또는 "Home"의 디렉토리 항목 유형을 갖는 데이터베이스 이름을 지정합니다. NULL 또는 빈 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 얻으십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집

계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결에 사용할 수 있으며 응용프로그램에서 사용할 수 있는 파티션입니다.

두 *dbname* 및 *dbpartitionnum*이 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저에서 작성된 파일로부터 데이터를 읽으려고 합니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형을 포함한 파일이 존재하지 않으면 SNAP\_GET\_TAB\_REORG 테이블 함수는 현재 연결된 데이터베이스와 데이터베이스 파티션 번호로 스냅샷을 작성합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_TAB\_REORG 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 데이터베이스에서 데이터베이스 파티션 1에 대한 재구성 조작에 관한 세부 사항을 선택합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

TAB_NAME	TAB_SCHEMA	REORG_PHASE	REORG_TYPE	...
EMPLOYEE	DBUSER	REPLACE	RECLAIM+OFFLINE+ALLO	...

1개의 레코드가 선택되었습니다.

...

이 쿼리의 출력(계속).

```

... REORG_STATUS REORG_COMPLETION DBPARTITIONNUM
... -----
... COMPLETED SUCCESS 1
...

```

재구성 조작에 관한 모든 정보를 선택하여 SNAP\_GET\_TAB\_REORG 테이블 함수를 사용하여 다차원적으로 클러스터된(MDC) 테이블에서 Extent를 재개합니다.

```
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

```

## 리턴된 정보

표 158. SNAPTAB\_REORG 관리 뷰 및 SNAP\_GET\_TAB\_REORG 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TABNAME	VARCHAR (128)	table_name - 테이블 이름
TABSCHEMA	VARCHAR (128)	table_schema - 테이블 스키마 이름
PAGE_REORGS	BIGINT	page_reorgs - 페이지 재구성
REORG_PHASE	VARCHAR (16)	reorg_phase - 테이블 재구성 단계. 이 인터페이스는 sqlmon.h의 정의를 기본으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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> 또는 SORT+DICT_SAMPLE.
REORG_MAX_PHASE	INTEGER	reorg_max_phase - 최대 테이블 재구성 단계
REORG_CURRENT_COUNTER	BIGINT	reorg_current_counter - 테이블 재구성 진행
REORG_MAX_COUNTER	BIGINT	reorg_max_counter - 전체 테이블 재구성 크기



표 158. SNAPTAB\_REORG 관리 뷰 및 SNAP\_GET\_TAB\_REORG 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
REORG_TYPE	VARCHAR (128)	<p>reorg_type - 테이블 재구성 속성. 이 인터페이스는 '+'로 구분되는 다음 ID의 조합을 사용하여 텍스트 ID를 리턴합니다.</p> <p>또는 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• RECLAIM</li> <li>• RECLUSTER</li> <li>• RECLAIM_EXTS</li> </ul> <p>또는 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• +OFFLINE</li> <li>• +ONLINE</li> </ul> <p>액세스 모드 지정 시 다음 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• +ALLOW_NONE</li> <li>• +ALLOW_READ</li> <li>• +ALLOW_WRITE</li> </ul> <p>오프라인 및 RECLUSTER 옵션의 경우 다음 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• +INDEXSCAN</li> <li>• +TABLESCAN</li> </ul> <p>오프라인의 경우 다음 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• +LONGLOB</li> <li>• +DATAONLY</li> </ul> <p>오프라인이고 옵션이 지정된 경우 다음 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• +CHOOSE_TEMP</li> <li>• +KEEPDICTIONARY</li> <li>• +RESETDICTIONARY</li> </ul> <p>온라인이고 옵션이 지정된 경우,</p> <ul style="list-style-type: none"> <li>• +NOTRUNCATE</li> </ul> <p>예 1: REORG TABLE TEST.EMPLOYEE가 실행되면 다음과 같이 표시됩니다.</p> <p>RECLAIM+OFFLINE+ALLOW_READ+DATAONLY +KEEPDICTIONARY</p> <p>예 2: REORG TABLE TEST.EMPLOYEE INDEX EMPIDX INDEXSCAN이 실행되면 다음과 같이 표시됩니다.</p> <p>RECLUSTER+OFFLINE+ALLOW_READ+INDEXSCAN +DATAONLY+KEEPDICTIONARY</p>

표 158. SNAPTAB\_REORG 관리 뷰 및 SNAP\_GET\_TAB\_REORG 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
REORG_STATUS	VARCHAR (10)	reorg_status - 테이블 재구성 상태. 이 인터페이스는 sqlmon.h의 정의를 기본으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 테이블 재구성 완료 플래그. 이 인터페이스는 sqlmon.h의 정의를 기본으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• FAIL</li> <li>• SUCCESS</li> </ul>
REORG_START	TIMESTAMP	reorg_start - 테이블 재구성 시작 시간
REORG_END	TIMESTAMP	reorg_end - 테이블 재구성 종료 시간
REORG_PHASE_START	TIMESTAMP	reorg_phase_start - 테이블 재구성 단계 시작 시간
REORG_INDEX_ID	BIGINT	reorg_index_id - 테이블 재구성에 사용된 인덱스
REORG_TBSPC_ID	BIGINT	reorg_tbsp_id - 테이블이 재구성되는 테이블 스페이스
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
DATA_PARTITION_ID	INTEGER	data_partition_id - 데이터 파티션 ID. 파티션되지 않은 테이블의 경우 이 요소는 널(NULL)이 됩니다.
REORG_ROWSCOMPRESSED	BIGINT	reorg_rows_compressed - 압축된 행
REORG_ROWSREJECTED	BIGINT	reorg_rows_rejected_for_compression - 압축에 대해 거부된 행
REORG_LONG_TBSPC_ID	BIGINT	reorg_long_tbsp_id - long 오브젝트가 재구성된 테이블 스페이스

## SNAPTBSP 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수 - 테이블 스페이스 논리 데이터 그룹 스냅샷 정보 검색

SNAPTBSP 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수는 테이블 스페이스 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.

### SNAPTBSP 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 테이블 스페이스 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPTbsp\_PART, SNAPTbsp\_QUIESCER, SNAPTbsp\_RANGE, SNAPCONTAINER 관리 뷰와 같이 사용하여 SNAPTbsp 관리 뷰는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 641 페이지의 표 159의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPTbsp 관리 뷰에 대한 SELECT 특권
- SNAPTbsp 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_TBSP\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 데이터베이스에 대한 카탈로그 데이터베이스 파티션에서 테이블 스페이스 목록을 검색합니다.

```
SELECT SUBSTR(TBSP_NAME,1,30) AS TBSP_NAME, TBSP_ID, TBSP_TYPE,
       TBSP_CONTENT_TYPE FROM SYSIBMADM.SNAPTbsp WHERE DBPARTITIONNUM = 1
```

다음은 이 쿼리의 출력 예입니다.

TBSP_NAME	TBSP_ID	TBSP_TYPE	TBSP_CONTENT_TYPE
TEMPSPACE1	1	SMS	SYSTEMP
USERSPACE1	2	DMS	LONG

2개의 레코드가 선택되었습니다.

## SNAP\_GET\_TBSP\_V91 테이블 함수

SNAP\_GET\_TBSP\_V91 테이블 함수는 SNAPTbsp 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

SNAP\_GET\_TBSP\_PART\_V91, SNAP\_GET\_TBSP\_QUIESCER, SNAP\_GET\_TBSP\_RANGE, SNAP\_GET\_CONTAINER\_V91 테이블 함수와 함께 사용되는 SNAP\_GET\_TBSP\_V91 테이블 함수는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 641 페이지의 표 159의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_TBSP_V91 ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. NULL 또는 빈 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 얻으십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_TBSP\_V91 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대해 스냅샷을 작성합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_TBSP\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 데이터베이스의 모든 데이터베이스 파티션에 대한 테이블 스페이스 목록을 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

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

8개의 레코드가 선택되었습니다.

## 리턴된 정보

표 159. SNAPTbsp 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
TBSP_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID

표 159. SNAPTBSP 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TBSP_TYPE	VARCHAR(10)	tablespace_type - 테이블 스페이스 유형. 이 인터페이스는 sqlutil.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• DMS</li> <li>• SMS</li> </ul>
TBSP_CONTENT_TYPE	VARCHAR(10)	tablespace_content_type - 테이블 스페이스 콘텐츠 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 테이블 스페이스 페이지 크기
TBSP_EXTENT_SIZE	BIGINT	tablespace_extent_size - 테이블 스페이스 Extent 크기
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 테이블 스페이스 프리페치 크기
TBSP_CUR_POOL_ID	BIGINT	tablespace_cur_pool_id - 현재 사용 중인 버퍼 풀
TBSP_NEXT_POOL_ID	BIGINT	tablespace_next_pool_id - 다음 시작 시에 사용되는 버퍼 풀
FS_CACHING	SMALLINT	fs_caching - 파일 시스템 캐싱
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수

표 159. SNAPTBSP 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - 버퍼 풀 비동기 읽기 시간
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - 버퍼 풀 비동기 쓰기 시간
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 수
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - 버퍼 풀 희생 버퍼 없음

표 159. SNAPTbsp 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
FILES_CLOSED	BIGINT	files_closed - 닫힌 데이터베이스 파일 수
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리페치 페이지 수
TBSP_REBALANCER_MODE	VARCHAR(10)	tablespace_rebalancer_mode - 재조정 프로그램 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 테이블 공간을 자동 스토리지에 사용 가능
TBSP_AUTO_RESIZE_ENABLED	SMALLINT	tablespace_auto_resize_enabled - 테이블 공간 자동 크기 조정 가능
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPTbsp\_PART 관리 뷰 및 SNAP\_GET\_TBSP\_PART\_V97 테이블 함수 - tablespace\_nodeinfo 논리 데이터 그룹 스냅샷 정보 검색

SNAPTbsp\_PART 관리 뷰 및 SNAP\_GET\_TBSP\_PART\_V97 테이블 함수는 tablespace\_nodeinfo 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.

### SNAPTbsp\_PART 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 tablespace\_nodeinfo 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPTbsp, SNAPTbsp\_QUIESCER, SNAPTbsp\_RANGE, SNAPCONTAINER 관리 뷰와 같이 사용하여 SNAPTbsp\_PART 관리 뷰는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.



리턴될 수 있는 정보의 완전한 목록은 647 페이지의 표 160의 내용을 참조하십시오.

### 권한 부여

- SYSMON 권한
- SNAPTbsp\_PART 관리 뷰의 SELECT 또는 CONTROL 특권 및 SNAP\_GET\_TBSP\_PART\_V97 테이블 함수의 EXECUTE 특권입니다.

### 예 :

현재 연결된 데이터베이스의 모든 데이터베이스 파티션에 대한 테이블 스페이스 목록 및 해당 상태를 검색합니다.

```
SELECT SUBSTR(TBSP_NAME,1,30) AS TBSP_NAME, TBSP_ID,
       SUBSTR(TBSP_STATE,1,30) AS TBSP_STATE, DBPARTITIONNUM
FROM SYSIBMADM.SNAPTbsp_PART
```

다음은 이 쿼리의 출력 예입니다.

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

5개의 레코드가 선택되었습니다.

### SNAP\_GET\_TBSP\_PART\_V97 테이블 함수

SNAP\_GET\_TBSP\_PART\_V97 테이블 함수는 SNAPTbsp\_PART 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

SNAP\_GET\_TBSP\_V97, SNAP\_GET\_TBSP QUIESCER, SNAP\_GET\_TBSP\_RANGE, SNAP\_GET\_CONTAINER\_V91 테이블 함수와 함께 사용되는 SNAP\_GET\_TBSP\_PART\_V97 테이블 함수는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 647 페이지의 표 160의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_TBSP_PART_V97 ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. NULL 또는 빈 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 얻으십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_TBSP\_PART\_V97 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대해 스냅샷을 작성합니다.

## 권한 부여

- SYSMON 권한
- SNAP\_GET\_TBSP\_PART\_V97 테이블 함수의 EXECUTE 특권

## 예 :

연결된 데이터베이스의 연결된 데이터베이스 파티션에 대한 테이블 스페이스 목록 및 해당 상태를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

TBSP_NAME	TBSP_ID	TBSP_STATE
SYSCATSPACE		0 NORMAL
TEMPSPACE1		1 NORMAL
USERSPACE1		2 NORMAL
SYSTOOLSPACE		3 NORMAL
SYSTOOLSTMPSPACE		4 NORMAL

5개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 160. SNAPTBSP\_PART 관리 뷰 및 SNAP\_GET\_TBSP\_PART\_V97 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TBSP_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
TBSP_STATE	VARCHAR (256)	<p>tablespace_state - 테이블 스페이스 상태. 이 인터페이스는 sqlutil.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 '+' 기호로 구분된 다음의 조합입니다.</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 - 테이블 스페이스 프리페치 크기
TBSP_NUM_QUIESCERS	BIGINT	tablespace_num_quiescers - quiescer 수

표 160. SNAPTBSP\_PART 관리 뷰 및 SNAP\_GET\_TBSP\_PART\_V97 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TBSP_STATE_CHANGE_OBJECT_ID	BIGINT	tablespace_state_change_object_id - 상태 변경 오브젝트 ID
TBSP_STATE_CHANGE_TBSP_ID	BIGINT	tablespace_state_change_ts_id - 상태 변경 테이블 스페이스 ID
TBSP_MIN_RECOVERY_TIME	TIMESTAMP	tablespace_min_recovery_time - 롤 포워드를 위한 최소 복구 시간
TBSP_TOTAL_PAGES	BIGINT	tablespace_total_pages - 테이블 스페이스의 전체 페이지
TBSP_USABLE_PAGES	BIGINT	tablespace_usable_pages - 테이블 스페이스에서 사용할 수 있는 페이지
TBSP_USED_PAGES	BIGINT	tablespace_used_pages - 테이블 스페이스에서 사용된 페이지
TBSP_FREE_PAGES	BIGINT	tablespace_free_pages - 테이블 스페이스에서 사용 가능한 페이지
TBSP_PENDING_FREE_PAGES	BIGINT	tablespace_pending_free_pages - 테이블 스페이스에서 보류 중인 사용 가능한 페이지
TBSP_PAGE_TOP	BIGINT	tablespace_page_top - 테이블 스페이스 상위 워터마크(water mark)
REBALANCER_MODE	VARCHAR (30)	tablespace_rebalancer_mode - 재조정 프로그램 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 재조정 프로그램에서 처리되는 전체 Extent 수
REBALANCER_EXTENTS_PROCESSED	BIGINT	tablespace_rebalancer_extents_processed - 재조정 프로그램이 처리한 Extent 수
REBALANCER_PRIORITY	BIGINT	tablespace_rebalancer_priority - 현재 재조정 프로그램 우선순위
REBALANCER_START_TIME	TIMESTAMP	tablespace_rebalancer_start_time - 재조정 프로그램 시작 시간

표 160. SNAPTbsp\_Part 관리 뷰 및 Snap\_Get\_Tbsp\_Part\_V97 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
REBALANCER_RESTART_TIME	TIMESTAMP	tablespace_rebalancer_restart_time - 재조정 프로그램 재시작 시간
REBALANCER_LAST_EXTENT_MOVED	BIGINT	tablespace_rebalancer_last_extent_moved - 재조정 프로그램이 이동한 마지막 Extent
TBSP_NUM_RANGES	BIGINT	tablespace_num_ranges - 테이블 스페이스 맵의 범위 수
TBSP_NUM_CONTAINERS	BIGINT	tablespace_num_containers - 테이블 스페이스의 컨테이너 수
TBSP_INITIAL_SIZE	BIGINT	tablespace_initial_size - 초기 테이블 스페이스 크기
TBSP_CURRENT_SIZE	BIGINT	tablespace_current_size - 현재 테이블 스페이스 크기
TBSP_MAX_SIZE	BIGINT	tablespace_max_size - 최대 테이블 스페이스 크기
TBSP_INCREASE_SIZE	BIGINT	tablespace_increase_size - 바이트 단위로 크기 늘리기
TBSP_INCREASE_SIZE_PERCENT	SMALLINT	tablespace_increase_size_percent - 퍼센트 단위로 크기 늘리기
TBSP_LAST_RESIZE_TIME	TIMESTAMP	tablespace_last_resize_time - 마지막으로 성공한 크기 조정 시간
TBSP_LAST_RESIZE_FAILED	SMALLINT	tablespace_last_resize_failed - 마지막으로 실패한 크기 조정 시도
TBSP_PATHS_DROPPED	SMALLINT	테이블 스페이스가 삭제된 하나 이상의 스토리지 경로에 있음을 나타냅니다(0 - 아니오, 1 - 예).
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPTbsp\_Quiescer 관리 뷰 및 Snap\_Get\_Tbsp\_Quiescer 테이블 함수 - Quiescer 테이블 스페이스 스냅샷 정보 검색

SNAPTbsp\_Quiescer 관리 뷰 및 Snap\_Get\_Tbsp\_Quiescer 테이블 함수는 테이블 스페이스 스냅샷에서 Quiescer에 대한 정보를 리턴합니다.

### SNAPTbsp\_Quiescer 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 Quiescer 테이블 스페이스 스냅샷 정보를 검색할 수 있습니다.

SNAPTbsp, SNAPTbsp\_Part, SNAPTbsp\_Range, SNAPCONTAINER 관리 뷰와 같이 사용하여 SNAPTbsp\_QUIESCER 관리 뷰는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 653 페이지의 표 161의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPTbsp\_QUIESCER 관리 뷰에 대한 SELECT 특권
- SNAPTbsp\_QUIESCER 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_Tbsp\_QUIESCER 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 데이터베이스의 모든 데이터베이스 파티션에 대한 quiesce된 테이블 스페이스에서 정보를 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

Tbsp_NAME	QUIESCER_TS_ID	QUIESCER_OBJ_ID	QUIESCER_AUTH_ID	..
USERSPACE1	2		5 SWALKTY	..
USERSPACE1	2		5 SWALKTY	..

2개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```

... QUIESCER_AGENT_ID    QUIESCER_STATE DBPARTITIONNUM
... -----
...                0 EXCLUSIVE                0
...                65983 EXCLUSIVE            1

```

**예: 범위 파티션 테이블 이름을 판별합니다.**

테이블 범위가 파티션되었고 Quiesce 상태인 경우 테이블 스페이스 ID 및 테이블 ID는 SYSCAT.TABLES에 표시된 값과 다르게 표시됩니다. 이 ID는 부호없이 짧게 표시됩니다. Quiesce 상태의 테이블 이름을 찾으려면 다음과 같은 방법으로 우선 부호있는 짧은 표시를 찾으십시오. QEUIESCER\_TS\_ID에서 65536(최대값)을 빼고 이 테이블 스페이스 ID를 사용하여 Quiesce 상태의 테이블을 찾으십시오. (실제 테이블 스페이스 ID는 테이블에서 각 범위 파티션에 대한 SYSCAT.DATAPARTITIONS에 있습니다.)

```

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

```

이 쿼리의 출력 예는 다음과 같습니다.

```

TBSP_NAME      TBSPACEID      TABLEID
-----
TABDATA        -6              -32768
DATAMART       -6              -32765
SMALL          5               17

```

3개의 레코드가 선택되었습니다.

위 쿼리에서 제공된, 지정된 TBSPACEID 및 TABLEID를 사용하여 SYSCAT.TABLES에서 테이블 스키마 및 이름을 찾으십시오.

```

SELECT CHAR(tabschema, 10)tabschema, CHAR(tabname,15)tabname
FROM SYSCAT.TABLES
WHERE tbspaceid = -6 AND tableid in (-32768,-32765)

```

다음은 이 쿼리의 출력 예입니다.

```

TABSHEMA      TABNAME
-----
TPCD           ORDERS_RP
TPCD           ORDERS_DMART

```

2개의 레코드가 선택되었습니다.

```

SELECT CHAR(tabschema, 10)tabschema, CHAR(tabname,15)tabname
FROM SYSCAT.TABLES
WHERE tbspaceid = 5 AND tableid = 17

```

다음은 이 쿼리의 출력 예입니다.

TABSCHEMA	TABNAME
-----	-----
TPCD	NATION

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_TBSP QUIESCER 테이블 함수

SNAP\_GET\_TBSP QUIESCER 테이블 함수는 SNAPT BSP QUIESCER 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

SNAP\_GET\_TBSP\_V91, SNAP\_GET\_TBSP\_PART\_V91, SNAP\_GET\_TBSP\_RANGE, SNAP\_GET\_CONTAINER\_V91 테이블 함수와 함께 사용되는 SNAP\_GET\_TBSP QUIESCER 테이블 함수는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 653 페이지의 표 161의 내용을 참조하십시오.

## 구문

```

▶▶ SNAP_GET_TBSP QUIESCER (—dbname— [ , dbpartitionnum ] )

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### dbname

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 NULL 또는 비어 있는 문자열을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. dbname이 NULL로 설정되지 않고 dbpartitionnum이 NULL로 설정된 경우 -1이 dbpartitionnum에 대해 내재적으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, dbname만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.



*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_TBSP QUIESCER 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대해 스냅샷을 작성합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_TBSP QUIESCER 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 데이터베이스의 데이터베이스 파티션 1에 대한 quiesce된 테이블 스페이스에서 정보를 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

```
TBSP_NAME  QUIESCER_TS_ID  QUIESCER_OBJ_ID  QUIESCER_AUTH_ID  ...
-----
USERSPACE1          2                5 SWALKTY          ...
```

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... QUIESCER_AGENT_ID  QUIESCER_STATE  DBPARTITIONNUM
... -----
...                65983  EXCLUSIVE                1
```

## 리턴된 정보

표 161. SNAP\_TBSP QUIESCER 관리 뷰 및 SNAP\_GET\_TBSP QUIESCER 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.

표 161. SNAPTbsp\_QUIESCER 관리 뷰 및 SNAP\_GET\_TBSP\_QUIESCER 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TBSP_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
QUIESCER_TS_ID	BIGINT	quiescer_ts_id - Quiescer 테이블 스페이스 ID
QUIESCER_OBJ_ID	BIGINT	quiescer_obj_id - Quiescer 오브젝트 ID
QUIESCER_AUTH_ID	VARCHAR (128)	quiescer_auth_id - Quiescer 사용자 권한 부여 ID
QUIESCER_AGENT_ID	BIGINT	quiescer_agent_id - Quiescer 에이전트 ID
QUIESCER_STATE	VARCHAR(14)	quiescer_state - Quiescer 상태. 이 인터페이스는 sqlutil.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• EXCLUSIVE</li> <li>• UPDATE</li> <li>• SHARE</li> </ul>
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPTbsp\_RANGE 관리 뷰 및 SNAP\_GET\_TBSP\_RANGE 테이블 함수 - 범위 스냅샷 정보 검색

SNAPTbsp\_RANGE 관리 뷰 및 SNAP\_GET\_TBSP\_RANGE 테이블 함수는 범위 스냅샷에서 정보를 리턴합니다.

### SNAPTbsp\_RANGE 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 범위 스냅샷 정보를 검색할 수 있습니다.

SNAPTbsp, SNAPTbsp\_PART, SNAPTbsp\_QUIESCER 및 SNAPCONTAINER 관리 뷰와 함께 사용될 경우, SNAPTbsp\_RANGE 관리 뷰는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 658 페이지의 표 162의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPTbsp\_RANGE 관리 뷰에 대한 SELECT 특권
- SNAPTbsp\_RANGE 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_Tbsp\_RANGE 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 데이터베이스에 대한 모든 데이터베이스 파티션의 테이블 스페이스 범위 정보를 선택합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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 ...

5개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	RANGE_OFFSET	RANGE_MAX_PAGE	RANGE_MAX_EXTENT	...
...	0	11515	2878	...
...	0	479	14	...
...	0	251	62	...
...	0	479	14	...
...	0	479	14	...

이 쿼리의 출력(계속).

...	RANGE_START_STRIPE	RANGE_END_STRIPE	RANGE_ADJUSTMENT	...
...	0	2878	0	...
...	0	14	0	...
...	0	62	0	...
...	0	14	0	...
...	0	14	0	...

이 쿼리의 출력(계속).

...	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 테이블 함수

SNAP\_GET\_TBSP\_RANGE 테이블 함수는 SNAPTbsp\_RANGE 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_TBSP\_V91, SNAP\_GET\_TBSP\_PART\_V91, SNAP\_GET\_TBSP QUIESCER 및 SNAP\_GET\_CONTAINER\_V91 테이블 함수와 함께 사용될 경우, SNAP\_GET\_TBSP\_RANGE 테이블 함수는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동일한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 658 페이지의 표 162의 내용을 참조하십시오.

## 구문

```

▶▶ SNAP_GET_TBSP_RANGE ( (dbname [ , dbpartitionnum ] ) )

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### dbname

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 NULL 또는 비어 있는 문자열을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집

계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결에 사용할 수 있으며 응용프로그램에서 사용할 수 있는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_TBSP\_RANGE 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_TBSP\_RANGE 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예:

현재 연결된 데이터베이스 파티션에서 *tbasp\_id* = 2인 테이블 스페이스에 대해 테이블 스페이스 범위 정보를 선택합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
TBSP_ID      TBSP_NAME      RANGE_NUMBER      ...
-----
          2 USERSPACE1          0 ...
```

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... RANGE_STRIPE_SET_NUMBER RANGE_OFFSET      RANGE_MAX_PAGE      ...
... RANGE_STRIPE_SET_NUMBER RANGE_OFFSET      RANGE_MAX_PAGE      ...
... -----
...                                0                0                3967 ...
```

이 쿼리의 출력(계속).

```
... RANGE_MAX_EXTENT      RANGE_START_STRIPE  RANGE_END_STRIPE      ...
... -----
...                      123                          0                      123 ...
```

이 쿼리의 출력(계속).

```
... RANGE_ADJUSTMENT      RANGE_NUM_CONTAINER  RANGE_CONTAINER_ID
... -----
...                      0                      1                      0
```

## 리턴되는 정보

표 162. SNAPTBSP\_RANGE 관리 뷰 및 SNAP\_GET\_TBSP\_RANGE 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
TBSP_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
RANGE_NUMBER	BIGINT	range_number - 범위 번호
RANGE_STRIPE_SET_NUMBER	BIGINT	range_stripe_set_number - 스트라이프 세트 번호
RANGE_OFFSET	BIGINT	range_offset - 범위 오프셋
RANGE_MAX_PAGE	BIGINT	range_max_page_number - 범위의 최대 페이지
RANGE_MAX_EXTENT	BIGINT	range_max_extent - 범위의 최대 Extent
RANGE_START_STRIPE	BIGINT	range_start_stripe - 시작 스트라이프
RANGE_END_STRIPE	BIGINT	range_end_stripe - 끝 스트라이프
RANGE_ADJUSTMENT	BIGINT	range_adjustment - 범위 조정
RANGE_NUM_CONTAINER	BIGINT	range_num_containers - 범위의 컨테이너 수
RANGE_CONTAINER_ID	BIGINT	range_container_id - 범위 컨테이너
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPUTIL 관리 뷰 및 SNAP\_GET\_UTIL 테이블 함수 - utility\_info 논리 데이터 그룹 스냅샷 정보 검색

SNAPUTIL 관리 뷰 및 SNAP\_GET\_UTIL 테이블 함수는 utility\_info 논리 데이터 그룹에서 유틸리티에 대한 스냅샷 정보를 리턴합니다.

### SNAPUTIL 관리 뷰

SNAPUTIL\_PROGRESS 관리 뷰와 함께 사용될 경우, SNAPUTIL 관리 뷰는 LIST UTILITIES SHOW DETAIL CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 661 페이지의 표 163의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPUTIL 관리 뷰에 대한 SELECT 특권
- SNAPUTIL 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_UTIL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

연결된 데이터베이스를 포함하는 인스턴스에서 모든 활성 데이터베이스에 대한 모든 데이터베이스 파티션의 유틸리티 및 상태 목록을 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
UTILITY_TYPE      UTILITY_PRIORITY ...
-----...- - - - - ...
LOAD              - ...
LOAD              - ...
LOAD              - ...
```

3개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... 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 ...
```

이 쿼리의 출력(계속).

UTILITY_DBNAME	UTILITY_STATE	UTILITY_INVOKER_TYPE	DBPARTITIONNUM
...	-----	-----	-----
... SAMPLE	EXECUTE	USER	0
... SAMPLE	EXECUTE	USER	1
... SAMPLE	EXECUTE	USER	2

## SNAP\_GET\_UTIL 테이블 함수

SNAP\_GET\_UTIL 테이블 함수는 SNAPUTIL 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_UTIL\_PROGRESS 테이블 함수와 함께 사용될 경우, SNAP\_GET\_UTIL 테이블 함수는 LIST UTILITIES SHOW DETAIL CLP 명령과 동일한 정보를 제공 합니다.

리턴될 수 있는 정보의 완전한 목록은 661 페이지의 표 163의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_UTIL ( dbpartitionnum )
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_UTIL 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_UTIL 테이블 함수에 대한 EXECUTE 특권



- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMMAINT
- SYSADM

예 :

SAMPLE 데이터베이스의 현재 연결된 데이터베이스 파티션에 대한 유틸리티 ID, 유형 및 상태 목록을 검색합니다.

```
SELECT UTILITY_ID, UTILITY_TYPE, STATE
FROM TABLE(SNAP_GET_UTIL(-1)) AS T WHERE UTILITY_DBNAME='SAMPLE'
```

이 쿼리의 출력 예는 다음과 같습니다.

UTILITY_ID	UTILITY_TYPE	STATE
-----	-----	-----
	1 BACKUP	EXECUTE

1개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 163. SNAPUTIL 관리 뷰 및 SNAP\_GET\_UTIL 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
UTILITY_ID	INTEGER	utility_id - 유틸리티 ID. 데이터베이스 파티션에 고유함.
UTILITY_TYPE	VARCHAR(26)	utility_type - 유틸리티 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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>• REORG</li> <li>• RESTART_RECREATE_INDEX</li> <li>• RESTORE</li> <li>• ROLLFORWARD_RECOVERY</li> <li>• RUNSTATS</li> </ul>

표 163. SNAPUTIL 관리 뷰 및 SNAP\_GET\_UTIL 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
UTILITY_PRIORITY	INTEGER	utility_priority - 유틸리티 우선순위. 유틸리티에서 조절 기능을 지원하는 경우 우선순위, 그렇지 않은 경우 널(NULL).
UTILITY_DESCRIPTION	VARCHAR(2048)	utility_description - 유틸리티 설명. 결과는 널(NULL)이 될 수 있습니다.
UTILITY_DBNAME	VARCHAR(128)	utility_dbname - 유틸리티가 작동하는 데이터베이스
UTILITY_START_TIME	TIMESTAMP	utility_start_time - 유틸리티 시작 시간
UTILITY_STATE	VARCHAR(10)	utility_state - 유틸리티 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• ERROR</li> <li>• EXECUTE</li> <li>• WAIT</li> </ul>
UTILITY_INVOKER_TYPE	VARCHAR(10)	utility_invoker_type - 유틸리티 호출자 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• AUTO</li> <li>• USER</li> </ul>
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
PROGRESS_LIST_ATTR	VARCHAR(10)	progress_list_attr - 현재 진행 목록 속성
PROGRESS_LIST_CUR_SEQ_NUM	INTEGER	progress_list_current_seq_num - Current® 진행 목록 시퀀스 수

## SNAPUTIL\_PROGRESS 관리 뷰 및 SNAP\_GET\_UTIL\_PROGRESS 테이블 함수 - 진행 중 논리 데이터 그룹 스냅샷 정보 검색

SNAPUTIL\_PROGRESS 관리 뷰 및 SNAP\_GET\_UTIL\_PROGRESS 테이블 함수는 유틸리티 진행에 대한 스냅샷 정보 특히 진행 논리 데이터 그룹에 대한 스냅샷 정보를 리턴합니다.

### SNAPUTIL\_PROGRESS 관리 뷰

SNAPUTIL 관리 뷰와 함께 사용될 경우, SNAPUTIL\_PROGRESS 관리 뷰는 LIST UTILITIES SHOW DETAIL CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 665 페이지의 표 164의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPUTIL\_PROGRESS 관리 뷰에 대한 SELECT 특권
- SNAPUTIL\_PROGRESS 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_UTIL\_PROGRESS 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

진행의 총계 및 완료된 단위에 대한 세부사항을 유틸리티 ID별로 검색합니다.

```
SELECT SELECT UTILITY_ID, PROGRESS_TOTAL_UNITS, PROGRESS_COMPLETED_UNITS,
        DBPARTITIONNUM FROM SYSIBMADM.SNAPUTIL_PROGRESS
```

이 쿼리의 출력 예는 다음과 같습니다.

UTILITY_ID	PROGRESS_TOTAL_UNITS	PROGRESS_COMPLETED_UNITS	DBPARTITIONNU
7	10	5	0
9	10	5	1

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_UTIL\_PROGRESS 테이블 함수

SSNAP\_GET\_UTIL\_PROGRESS 테이블 함수는 SNAPUTIL\_PROGRESS 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_UTIL 테이블 함수와 함께 사용될 경우, SNAP\_GET\_UTIL\_PROGRESS 테이블 함수는 LIST UTILITIES SHOW DETAIL CLP 명령과 동일한 정보를 제공합니다.

리턴될 수 있는 정보의 완전한 목록은 665 페이지의 표 164의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_UTIL_PROGRESS ( dbpartitionnum ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_UTIL\_PROGRESS 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_UTIL\_PROGRESS 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

현재 연결 파티션에 대한 유틸리티 진행 세부사항을 검색합니다.

```
SELECT UTILITY_ID, PROGRESS_TOTAL_UNITS, PROGRESS_COMPLETED_UNITS,  
       DBPARTITIONNUM FROM TABLE(SNAP_GET_UTIL_PROGRESS(-1)) as T
```

이 쿼리의 출력 예는 다음과 같습니다.

UTILITY\_ID PROGRESS\_TOTAL\_UNITS PROGRESS\_COMPLETED\_UNITS DBPARTITIONNUM

7

10

5

0

1개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 164. SNAPUTIL\_PROGRESS 관리 뷰 및 SNAP\_GET\_UTIL\_PROGRESS 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
UTILITY_ID	INTEGER	utility_id - 유틸리티 ID. 데이터베이스 파티션에 고유함.
PROGRESS_SEQ_NUM	INTEGER	progress_seq_num - 진행 시퀀스 수. 직렬일 경우 단계의 수, 동시일 경우 널(NULL)일 수 있습니다.
UTILITY_STATE	VARCHAR(16)	utility_state - 유틸리티 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• ERROR</li> <li>• EXECUTE</li> <li>• WAIT</li> </ul>
PROGRESS_DESCRIPTION	VARCHAR(2048)	progress_description - 진행 설명
PROGRESS_START_TIME	TIMESTAMP	progress_start_time - 진행 시작 시간. 단계가 시작된 경우 시작 시간, 그렇지 않은 경우 널(NULL).
PROGRESS_WORK_METRIC	VARCHAR(16)	progress_work_metric - 진행 작업 메트릭. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• NOT_SUPPORT</li> <li>• BYTES</li> <li>• EXTENTS</li> <li>• 인덱스</li> <li>• PAGES</li> <li>• ROWS</li> <li>• TABLES</li> </ul>
PROGRESS_TOTAL_UNITS	BIGINT	progress_total_units - 전체 진행 작업 단위
PROGRESS_COMPLETED_UNITS	BIGINT	progress_completed_units - 완료된 진행 작업 단위
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

---

## SNAP\_WRITE\_FILE 프로시저

SNAP\_WRITE\_FILE 프로시저는 인스턴스 디렉토리의 tmp 서브디렉토리에 있는 파일에 시스템 스냅샷 데이터를 기록합니다.

### 구문

▶—SNAP\_WRITE\_FILE—(—requestType—, —dbname—, —dbpartitionnum—)————▶

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *requestType*

유효한 스냅샷 요청 유형을 지정하는 유형 VARCHAR(32)의 입력 인수. 가능한 요청 유형은 sqlmon.h의 정의를 기반으로 하는 텍스트 ID이며 다음 중 하나입니다.

- APPL\_ALL
- BUFFERPOOLS\_ALL
- DB2
- DBASE\_ALL
- DBASE\_LOCKS
- DBASE\_TABLES
- DBASE\_TABLESPACES
- DYNAMIC\_SQL

#### *dbname*

해당 함수 호출시 현재 연결된 데이터베이스와 동일한 인스턴스에 있는 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. NULL 또는 빈 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 얻으십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정되면, -1이 내재적으로 설정됩니다.

## 권한 부여

프로시저를 실행하려면, 사용자에게 SYSADM, SYSCTRL, SYSMOINT 또는 SYSMON 권한이 있어야 합니다. 저장된 스냅샷은 스냅샷 테이블 함수에 대한 입력으로 널(NULL) 값을 전달함으로써 SYSADM, SYSCTRL, SYSMOINT 또는 SYSMON 권한이 없는 사용자가 읽을 수 있습니다.

### 예 :

요청 유형 'DB2'(SQLMA\_DB2에 해당)을 지정하고 현재 연결된 데이터베이스 및 현재 데이터베이스 파티션을 디폴트값으로 하여 데이터베이스 관리 프로그램 정보의 스냅샷을 취하십시오.

```
CALL SYSPROC.SNAP_WRITE_FILE ('DB2', '', -1)
```

이는 인스턴스 임시 디렉토리(UNIX 운영 체제의 경우 sql1lib/tmp/SQLMA\_DB2.dat 및 Windows 운영 체제의 경우 sql1lib\DB2\tmp\SQLMA\_DB2.dat)에 스냅샷 데이터를 작성합니다.

## 사용 시 참고사항

인식되지 않는 입력 매개변수를 제공할 경우, 다음과 같은 오류가 리턴됩니다.  
SQL2032N: "REQUEST\_TYPE" 매개변수가 유효하지 않습니다.

---

## SNAPAGENT 관리 뷰 및 SNAP\_GET\_AGENT 테이블 함수 - 에이전트 논리 데이터 그룹 응용프로그램 스냅샷 정보 검색

SNAPAGENT 관리 뷰 및 SNAP\_GET\_AGENT 테이블 함수는 응용프로그램 스냅샷에서 에이전트, 특히 에이전트 논리 데이터 그룹에 대한 정보를 리턴합니다.

### SNAPAGENT 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 에이전트 논리 데이터 그룹 응용프로그램 스냅샷 정보를 검색할 수 있습니다.

SNAPAGENT\_MEMORY\_POOL, SNAPAPPL, SNAPAPPL\_INFO, SNAPSTMT 및 SNAPSUBSECTION 관리 뷰와 함께 사용되는 경우, SNAPAGENT 관리 뷰는 GET SNAPSHOT FOR APPLICATIONS ON database-alias CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 521 페이지의 표 135의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPAGENT 관리 뷰에 대한 SELECT 특권
- SNAPAGENT 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_AGENT 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

에이전트 논리 데이터 그룹에서 현재 연결된 데이터베이스에 대한 모든 응용프로그램 스냅샷 정보를 검색합니다.

```
SELECT * FROM SYSIBMADM.SNAPAGENT
```

이 쿼리의 출력 예는 다음과 같습니다.

SNAPSHOT_TIMESTAMP	DB_NAME	AGENT_ID	...
2005-07-19-11.03.26.740423	SAMPLE	101	...
2005-07-19-11.03.26.740423	SAMPLE	49	...
			...

2개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

... AGENT_PID	LOCK_TIMEOUT_VAL	DBPARTITIONNUM	
... AGENT_PID	LOCK_TIMEOUT_VAL	DBPARTITIONNUM	
...	11980	-1	0
...	15940	-1	0
...			
...			

## SNAP\_GET\_AGENT 테이블 함수

SNAP\_GET\_AGENT 테이블 함수는 SNAPAGENT 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.



SNAP\_GET\_AGENT\_MEMORY\_POOL, SNAP\_GET\_APPL\_V95, SNAP\_GET\_APPL\_INFO\_V95, SNAP\_GET\_STMT 및 SNAP\_GET\_SUBSECTION 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_AGENT 테이블 함수는 GET SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

리턴될 수 있는 정보의 완전한 목록은 521 페이지의 표 135의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_AGENT ( ( dbname [ , dbpartitionnum ] ) )
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스와 동일한 인터페이스 내의 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_AGENT 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_AGENT 테이블 함수에 대한 EXECUTE 특권

- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

예 :

모든 활성 데이터베이스에서 모든 응용프로그램에 대한 모든 응용프로그램 스냅샷 정보를 검색합니다.

```
SELECT * FROM TABLE(SNAP_GET_AGENT(CAST(NULL AS VARCHAR(128)), -1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

```

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 ...

```

6개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```

... AGENT_PID      LOCK_TIMEOUT_VAL      DBPARTITIONNUM
... -----
...      7696      -1      0
...      8536      -1      0
...      6672      -1      0
...      2332      -1      0
...      8360      -1      0
...      6736      -1      0
...

```

## 리턴되는 정보

표 165. SNAPAGENT 관리 뷰 및 SNAP\_GET\_AGENT 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
AGENT_PID	BIGINT	agent_pid - EDU(Engine Dispatchable Unit)

표 165. SNAPAGENT 관리 뷰 및 SNAP\_GET\_AGENT 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_TIMEOUT_VAL	BIGINT	lock_timeout_val - 잠금 시간종료(초)
DBPARTITIONNUM	SMALLINT	행에 대한 데이터의 데이터베이스 파티션 검색.

## SNAPAGENT\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수 – memory\_pool 논리 데이터 그룹 스냅샷 정보 검색

SNAPAGENT\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수는 에이전트 레벨에서 메모리 사용에 대한 정보를 리턴합니다.

### SNAPAGENT\_MEMORY\_POOL 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 에이전트 레벨 메모리 사용에 관하여 memory\_pool 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPAGENT, SNAPAPPL, SNAPAPPL\_INFO, SNAPSTMT 및 SNAPSUBSECTION 관리 뷰와 함께 사용되는 경우, SNAPAGENT\_MEMORY\_POOL 관리 뷰는 GET SNAPSHOT FOR APPLICATIONS ON database-alias CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 525 페이지의 표 136의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPAGENT\_MEMORY\_POOL 관리 뷰에 대한 SELECT 특권
- SNAPAGENT\_MEMORY\_POOL 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL

- SYSMANT
- SYSADM

예 :

메모리 풀 및 현재 크기 목록을 검색합니다.

```
SELECT AGENT_ID, POOL_ID, POOL_CUR_SIZE FROM SYSIBMADM.SNAPAGENT_MEMORY_POOL
```

이 쿼리의 출력 예는 다음과 같습니다.

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

9개의 레코드가 선택되었습니다.

## SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수

SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수는

SNAPAGENT\_MEMORY\_POOL 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_AGENT, SNAP\_GET\_APPL\_V95, SNAP\_GET\_APPL\_INFO\_V95, SNAP\_GET\_STMT 및 SNAP\_GET\_SUBSECTION 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수는 GET SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 525 페이지의 표 136의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_AGENT_MEMORY_POOL ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

*dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정

하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스와 동일한 인터페이스 내의 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

모든 데이터베이스에 대한 메모리 풀 및 현재 크기 목록을 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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

18개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 166. SNAPAGENT\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
AGENT_PID	BIGINT	agent_pid - EDU(Engine Dispatchable Unit)

표 166. SNAPAGENT\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_ID	VARCHAR(14)	pool_id - 메모리 풀 ID. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 메모리 풀의 현재 크기
POOL_WATERMARK	BIGINT	pool_watermark - 메모리 풀 워터 마크 (water mark)
POOL_CONFIG_SIZE	BIGINT	pool_config_size - 메모리 풀의 구성된 크기
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수 - Retrieve appl\_info 논리 데이터 그룹 스냅샷 정보

SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수는 응용 프로그램 스냅샷에서 응용프로그램 특히, appl\_info 논리 데이터 그룹에 대한 정보를 리턴합니다.

## SNAPAPPL\_INFO 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 appl\_info 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPAGENT, SNAPAGENT\_MEMORY\_POOL, SNAPAPPL, SNAPSTMT 및 SNAPSUBSECTION 관리 뷰와 같이 사용하여 SNAPAPPL\_INFO 관리 뷰는 GET SNAPSHOT FOR APPLICATIONS ON database-alias CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 530 페이지의 표 137의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPAPPL\_INFO 관리 뷰에 대한 SELECT 특권
- SNAPAPPL\_INFO 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또한 다음 권한 중 하나도 필요합니다.

- SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또한 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSMAINT
- SYSCTRL
- SYSADM

### 예 :

현재 데이터베이스에 연결된 응용프로그램의 상태를 검색합니다.

```
SELECT AGENT_ID, SUBSTR(APPL_NAME,1,10) AS APPL_NAME, APPL_STATUS
      FROM SYSIBMADM.SNAPAPPL_INFO
```

다음은 이 쿼리의 출력 예입니다.

AGENT_ID	APPL_NAME	APPL_STATUS
101	db2bp.exe	UOWEXEC
49	db2bp.exe	CONNECTED

2개의 레코드가 선택되었습니다.



## SNAP\_GET\_APPL\_INFO\_V95 테이블 함수

SNAP\_GET\_APPL\_INFO\_V95 테이블 함수는 SNAPAPPL\_INFO 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

SNAP\_GET\_AGENT, SNAP\_GET\_AGENT\_MEMORY\_POOL, SNAP\_GET\_APPL\_V95, SNAP\_GET\_STMT 및 SNAP\_GET\_SUBSECTION 테이블 함수와 함께 사용되는 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수는 GET SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

리턴 가능한 전체 정보 목록은 530 페이지의 표 137의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_APPL_INFO_V95 ( (dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우 -1이 *dbpartitionnum*에 대해 내재적으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신

데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대해 스냅샷을 작성합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예:

연결된 데이터베이스 파티션의 모든 응용프로그램에 대한 상태를 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

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

4개의 레코드가 선택되었습니다.

다음은 테이블 함수 결과에서 선택할 때 확보한 내용입니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

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)

```

3개의 레코드가 선택되었습니다.

## 리턴된 정보

표 167. SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)

표 167. SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
APPL_STATUS	VARCHAR(22)	<p>appl_status - 응용프로그램 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다.</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 - 응용프로그램에서 사용되는 코드 페이지 ID
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 연관 에이전트 수
COORD_NODE_NUM	SMALLINT	coord_node - 코디네이팅 노드

표 167. SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
AUTHORITY_LVL	VARCHAR(512)	<p>authority_bitmap - 사용자 권한 부여 레벨.</p> <p>이 sql.h 및 해당 소스에 정의된 데이터베이스 권한을 기반으로 하는 텍스트 ID를 리턴하고 권한(소스, ...) + 권한(소스, ...) + ... 형식을 사용합니다. 권한의 소스는 여러 가지 유형이 가능하며 USER, GROUP 또는 USER 및 GROUP 중 하나입니다.</p> <p>"권한"에 대한 가능한 값은 다음과 같습니다.</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>"소스"에 대한 가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• USER - 사용자에게 부여된 권한 또는 사용자에게 부여된 역할</li> <li>• GROUP - 사용자가 속하는 그룹에 부여된 권한 또는 사용자가 속하는 그룹에 부여된 역할</li> </ul>
CLIENT_PID	BIGINT	client_pid - 클라이언트 프로세스 ID
COORD_AGENT_PID	BIGINT	coord_agent_pid - 코디네이터 에이전트
STATUS_CHANGE_TIME	TIMESTAMP	status_change_time - 응용프로그램 상태 변경 시간

표 167. SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
CLIENT_PLATFORM	VARCHAR(12)	<p>client_platform - 클라이언트 운영 플랫폼. 이 인터페이스는 sqlmon.h의 정의를 기반으로 텍스트 ID를 리턴합니다.</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>

표 167. SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
CLIENT_PROTOCOL	VARCHAR(10)	client_protocol - 클라이언트 통신 프로토콜. 이 인터페이스는 sqlmon.h의 정의를 기반으로 텍스트 ID를 리턴합니다.  <ul style="list-style-type: none"> <li>• CPIC</li> <li>• LOCAL</li> <li>• NETBIOS</li> <li>• NPIPE</li> <li>• TCPIP ( DB2 UDB용)</li> <li>• TCPIP4</li> <li>• TCPIP6</li> </ul>
TERRITORY_CODE	SMALLINT	territory_code - 데이터베이스 지역 코드
APPL_NAME	VARCHAR(256)	appl_name - 응용프로그램 이름
APPL_ID	VARCHAR(128)	appl_id - 응용프로그램 ID
SEQUENCE_NO	VARCHAR(4)	sequence_no - 시퀀스 번호
PRIMARY_AUTH_ID	VARCHAR(128)	auth_id - 권한 부여 ID
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - 세션 권한 부여 ID
CLIENT_NNAME	VARCHAR(128)	client_nname 모니터 요소는 더 이상 사용되지 않습니다. 리턴된 값은 올바른 값이 아닙니다.
CLIENT_PRDID	VARCHAR(128)	client_prdid - 클라이언트 제품/버전 ID
INPUT_DB_ALIAS	VARCHAR(128)	input_db_alias - 데이터베이스 별명 입력
CLIENT_DB_ALIAS	VARCHAR(128)	client_db_alias - 응용프로그램에서 사용되는 데이터베이스 별명
DB_NAME	VARCHAR(128)	db_name - 데이터베이스 이름
DB_PATH	VARCHAR(1024)	db_path - 데이터베이스 경로
EXECUTION_ID	VARCHAR(128)	execution_id - 사용자 로그인 ID
CORR_TOKEN	VARCHAR(128)	corr_token - DRDA 상환 토큰
TPMON_CLIENT_USERID	VARCHAR(256)	tpmon_client_userid - TP 모니터 클라이언트 사용자 ID
TPMON_CLIENT_WKSTN	VARCHAR(256)	tpmon_client_wkstn - TP 모니터 클라이언트 워크스테이션 이름
TPMON_CLIENT_APP	VARCHAR(256)	tpmon_client_app - TP 모니터 클라이언트 응용프로그램 이름
TPMON_ACC_STR	VARCHAR(200)	tpmon_acc_str - TP 모니터 클라이언트 어카운팅 문자열
DBPARTITIONNUM	SMALLINT	행의 데이터가 검색되는 데이터베이스 파티션입니다.

표 167. SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
WORKLOAD_ID	INTEGER	현재 워크로드 ID입니다.
IS_SYSTEM_APPL	SMALLINT	IS_SYSTEM_APPL 값은 응용프로그램이 DB2 내부 시스템 응용프로그램인지 여부를 나타냅니다.  0인 경우 사용자 응용프로그램입니다.  1인 경우 시스템 응용프로그램입니다.  DB2 시스템 응용프로그램의 예로는 DB2 이벤트 모니터가 있습니다.  일반적으로 DB2 시스템 응용프로그램의 이름은 "db2"로 시작됩니다. 예: db2stmm, db2taskd.

## SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수 - appl 논리 데이터 그룹 스냅샷 정보 검색

535 페이지의 『SNAPAPPL 관리 뷰』 및 536 페이지의 『SNAP\_GET\_APPL\_V95 테이블 함수』에서는 응용프로그램 스냅샷, 특히 appl 논리 데이터 그룹의 응용프로그램에 대한 정보를 리턴합니다.

### SNAPAPPL 관리 뷰

이 관리 뷰로 현재 연결된 데이터베이스에 대한 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPAGENT, SNAPAGENT\_MEMORY\_POOL, SNAPAPPL\_INFO, SNAPSTMT 및 SNAPSUBSECTION 관리 뷰와 함께 사용된 SNAPAPPL 관리 뷰는 GET SNAPSHOT FOR APPLICATIONS ON database-alias CLP 명령과 동일한 정보를 제공하나, 모든 데이터베이스 파티션에서 데이터를 검색합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 538 페이지의 표 138의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPAPPL 관리 뷰에 대한 SELECT 특권
- SNAPAPPL 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한



또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_APPL\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

연결된 데이터베이스의 각 응용프로그램을 읽고 쓴 행에 대한 세부사항을 검색하십시오.

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, AGENT_ID, ROWS_READ, ROWS_WRITTEN
FROM SYSIBMADM.SNAPAPPL
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	AGENT_ID	ROWS_READ	ROWS_WRITTEN
SAMPLE		7	25
			0

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_APPL\_V95 테이블 함수

SNAP\_GET\_APPL\_V95 테이블 함수는 SNAPAPPL 관리 뷰와 동일한 정보를 리턴하나, 특정 데이터베이스 파티션의 특정 데이터베이스에 대한 정보, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션은 사용자가 검색할 수 있습니다.

SNAP\_GET\_AGENT, SNAP\_GET\_AGENT\_MEMORY\_POOL, SNAP\_GET\_APPL\_INFO\_V95, SNAP\_GET\_STMT 및 SNAP\_GET\_SUBSECTION 테이블 함수와 함께 사용될 때, SNAP\_GET\_APPL\_V95 테이블 함수는 GET SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동일한 정보를 제공하나, 모든 데이터베이스 파티션에서 데이터를 검색합니다.

리턴될 수 있는 정보의 완전한 목록은 538 페이지의 표 138의 내용을 참조하십시오.

## 구문

```
SNAP_GET_APPL_V95 ( (dbname [ , dbpartitionnum ] ) )
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우, *dbpartitionnum*에 -1이 내재적으로 설정됩니다. 이 입력 옵션이 사용되지 않으면 즉, *dbname*만 제공되면, 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스가 응용프로그램별 연결 및 사용 가능한 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 언제든지 작성될 수 있으며, 이는 데이터가 통용되지 않음을 의미한다는 것을 알아두십시오. 해당 스냅샷 API 요청 유형이 있는 파일이 존재하지 않으면, SNAP\_GET\_APPL\_V95 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 수에 대한 스냅샷을 취합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_APPL\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

모든 활성 데이터베이스에 대한 각 응용프로그램을 읽거나 작성하는 행에 대한 세부사항을 검색하십시오.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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

## 리턴된 정보

표 168. SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수로 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 취한 날짜 및 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
UOW_LOG_SPACE_USED	BIGINT	uow_log_space_used - 사용된 작업 단위(UOW) 로그 스페이스
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
INACT_STMTHIST_SZ	BIGINT	stmt_history_list_size - 명령문 실행기록 목록 크기
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수

표 168. SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리페치 페이지 수
LOCKS_HELD	BIGINT	locks_held - 보유된 잠금 수
LOCK_WAITS	BIGINT	lock_waits - 잠금 대기 수
LOCK_WAIT_TIME	BIGINT	lock_wait_time - 잠금 대기 시간
LOCK_ESCALS	BIGINT	lock_escals - 잠금 에스컬레이션 수
X_LOCK_ESCALS	BIGINT	x_lock_escals - 배타적 잠금 에스컬레이션 수
DEADLOCKS	BIGINT	deadlocks - 발견된 교착 상태 수
TOTAL_SORTS	BIGINT	total_sorts - 전체 정렬 수
TOTAL_SORT_TIME	BIGINT	total_sort_time - 총 정렬 시간
SORT_OVERFLOWS	BIGINT	sort_overflows - 정렬 오버플로우 수
COMMIT_SQL_STMTS	BIGINT	commit_sql_stmts - 시도된 명령문 커밋
ROLLBACK_SQL_STMTS	BIGINT	rollback_sql_stmts - 시도된 롤백 명령문
DYNAMIC_SQL_STMTS	BIGINT	dynamic_sql_stmts - 시도된 동적 SQL문
STATIC_SQL_STMTS	BIGINT	static_sql_stmts - 시도된 정적 SQL문
FAILED_SQL_STMTS	BIGINT	failed_sql_stmts - 실패한 명령문 조각

표 168. SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SELECT_SQL_STMTS	BIGINT	select_sql_stmts - 실행된 Select SQL문
DDL_SQL_STMTS	BIGINT	ddl_sql_stmts - DDL(Data Definition Language) SQL문
UID_SQL_STMTS	BIGINT	uid_sql_stmts - 실행된 UPDATE/INSERT/DELETE SQL문
INT_AUTO_REBINDS	BIGINT	int_auto_rebinds - 내부 자동 리바인드 수
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 삭제된 내부 행 수
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 갱신된 내부 행 수
INT_COMMITS	BIGINT	int_commits - 내부 커미트 수
INT_ROLLBACKS	BIGINT	int_rollback - 내부 롤백 수
INT_DEADLOCK_ROLLBACKS	BIGINT	int_deadlock_rollback - 교착 상태로 인한 내부 롤백 수
ROWS_DELETED	BIGINT	rows_deleted - 삭제된 행 수
ROWS_INSERTED	BIGINT	rows_inserted - 삽입된 행 수
ROWS_UPDATED	BIGINT	rows_updated - 갱신된 행 수
ROWS_SELECTED	BIGINT	rows_selected - 선택된 행 수
BINDS_PRECOMPILES	BIGINT	binds_precompiles - 시도된 바인드/프리컴파일 수
OPEN_REM_CURS	BIGINT	open_rem_curs - 열린 리모트 커서 수
OPEN_REM_CURS_BLK	BIGINT	open_rem_curs_blk - 블로킹 상태의 열린 리모트 커서 수
REJ_CURS_BLK	BIGINT	rej_curs_blk - 거부된 블록 커서 요청 수
ACC_CURS_BLK	BIGINT	acc_curs_blk - 승인된 블록 커서 요청 수
SQL_REQS_SINCE_COMMIT	BIGINT	sql_reqs_since_commit - 마지막 커미트 이후 SQL 요청 수
LOCK_TIMEOUTS	BIGINT	lock_timeouts - 잠금 시간종료 수
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 삽입된 내부 행 수
OPEN_LOC_CURS	BIGINT	open_loc_curs - 열린 로컬 커서 수
OPEN_LOC_CURS_BLK	BIGINT	open_loc_curs_blk - 블로킹 상태의 열린 로컬 커서 수
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - 패키지 캐시 찾아보기 수
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - 패키지 캐시 삽입 수
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - 카탈로그 캐시 찾아보기 수
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - 카탈로그 캐시 삽입 수
CAT_CACHE_OVERFLOWS	BIGINT	cat_cache_overflows - 카탈로그 캐시 오버플로우 수
NUM_AGENTS	BIGINT	num_agents - 명령문에서 작업하는 에이전트 수

표 168. SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
AGENTS_STOLEN	BIGINT	agents_stolen - 분실 에이전트 수
ASSOCIATED_AGENTS_TOP	BIGINT	associated_agents_top - 연관 에이전트의 최대 수
APPL_PRIORITY	BIGINT	appl_priority - 응용프로그램 에이전트 우선순위
APPL_PRIORITY_TYPE	VARCHAR(16)	appl_priority_type - 응용프로그램 우선순위 유형. 이 인터페이스는 sqlmon.h로 정의하는 것을 기초로 하고 다음 중 하나인 텍스트 ID를 리턴합니다. <ul style="list-style-type: none"> <li>• DYNAMIC_PRIORITY</li> <li>• FIXED_PRIORITY</li> </ul>
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - 프리페치 대기 시간
APPL_SECTION_LOOKUPS	BIGINT	appl_section_lookups - 섹션 찾아보기 수
APPL_SECTION_INSERTS	BIGINT	appl_section_inserts - 섹션 삽입 수
LOCKS_WAITING	BIGINT	locks_waiting - 잠금 대기 중인 현재 에이전트 수
TOTAL_HASH_JOINS	BIGINT	total_hash_joins - 전체 해시 조인 수
TOTAL_HASH_LOOPS	BIGINT	total_hash_loops - 전체 해시 루프 수
HASH_JOIN_OVERFLOWS	BIGINT	hash_join_overflows - 해시 조인 오버플로우 수
HASH_JOIN_SMALL_OVERFLOWS	BIGINT	hash_join_small_overflows - 해시 조인 작은 오버플로우 수
APPL_IDLE_TIME	BIGINT	appl_idle_time - 응용프로그램 유휴 시간
UOW_LOCK_WAIT_TIME	BIGINT	uow_lock_wait_time - 잠금 상태로 대기한 전체 작업 단위(UOW) 시간
UOW_COMP_STATUS	VARCHAR(14)	uow_comp_status - 작업 단위(UOW) 완료 상태. 이 인터페이스는 sqlmon.h로 정의하는 것을 기초로 하고 다음 중 하나인 텍스트 ID를 리턴합니다. <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 - 에이전트가 사용한 사용자 CPU 시간

표 168. SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
AGENT_USR_CPU_TIME_MS	BIGINT	agent_usr_cpu_time - 에이전트가 사용한 사용자 CPU 시간
AGENT_SYS_CPU_TIME_S	BIGINT	agent_sys_cpu_time - 에이전트가 사용한 시스템 CPU 시간
AGENT_SYS_CPU_TIME_MS	BIGINT	agent_sys_cpu_time - 에이전트가 사용한 시스템 CPU 시간
APPL_CON_TIME	TIMESTAMP	appl_con_time - 연결 요청 시작 시간소인
CONN_COMPLETE_TIME	TIMESTAMP	conn_complete_time - 연결 요청 완료 시간소인
LAST_RESET	TIMESTAMP	last_reset - 마지막 재설정 시간소인
UOW_START_TIME	TIMESTAMP	uow_start_time - 작업 단위(UOW) 시작 시간소인
UOW_STOP_TIME	TIMESTAMP	uow_stop_time - 작업 단위(UOW) 중지 시간소인
PREV_UOW_STOP_TIME	TIMESTAMP	prev_uow_stop_time - 이전 작업 단위(UOW) 완료 시간소인
UOW_ELAPSED_TIME_S	BIGINT	uow_elapsed_time - 최근 작업 단위(UOW) 경과 시간
UOW_ELAPSED_TIME_MS	BIGINT	uow_elapsed_time - 최근 작업 단위(UOW) 경과 시간
ELAPSED_EXEC_TIME_S	BIGINT	elapsed_exec_time - 명령문 실행 경과 시간
ELAPSED_EXEC_TIME_MS	BIGINT	elapsed_exec_time - 명령문 실행 경과 시간
INBOUND_COMM_ADDRESS	VARCHAR(32)	inbound_comm_address - 인바운드 통신 주소
LOCK_TIMEOUT_VAL	BIGINT	lock_timeout_val - 잠금 시간종료(초)
PRIV_WORKSPACE_NUM_OVERFLOWS	BIGINT	priv_workspace_num_overflows - 개인용 작업 스페이스 오버플로우 수
PRIV_WORKSPACE_SECTION_INSERTS	BIGINT	priv_workspace_section_inserts - 개인용 작업 스페이스 섹션 삽입 수
PRIV_WORKSPACE_SECTION_LOOKUPS	BIGINT	priv_workspace_section_lookups - 개인용 작업 스페이스 섹션 찾아보기 수
PRIV_WORKSPACE_SIZE_TOP	BIGINT	priv_workspace_size_top - 개인용 작업 스페이스 최대 크기
SHR_WORKSPACE_NUM_OVERFLOWS	BIGINT	shr_workspace_num_overflows - 공유 작업 스페이스 오버플로우 수

표 168. SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SHR_WORKSPACE_SECTION_INSERTS	BIGINT	shr_workspace_section_inserts - 공유 작업 스페이스 섹션 삽입 수
SHR_WORKSPACE_SECTION_LOOKUPS	BIGINT	shr_workspace_section_lookups - 공유 작업 스페이스 섹션 찾아보기 수
SHR_WORKSPACE_SIZE_TOP	BIGINT	shr_workspace_size_top - 공유 작업 스페이스 최대 크기
DBPARTITIONNUM	SMALLINT	행에 대한 데이터의 데이터베이스 파티션 검색.
CAT_CACHE_SIZE_TOP	BIGINT	cat_cache_size_top - 카탈로그 캐시 상위 워터 마크(water mark)
TOTAL_OLAP_FUNCS	BIGINT	실행된 총 OLAP 함수 수
OLAP_FUNC_OVERFLOWS	BIGINT	OLAP 함수 데이터가 사용 가능한 정렬 힙 스페이스를 초과한 횟수

## SNAPBP 관리 뷰 및 SNAP\_GET\_BP\_V95 테이블 함수 - 버퍼 풀 논리 그룹 스냅샷 정보 검색

SNAPBP 관리 뷰 및 SNAP\_GET\_BP\_V95 테이블 함수는 버퍼 풀 스냅샷에서 버퍼 풀, 특히 버퍼 풀 논리 데이터 그룹에 대한 정보를 리턴합니다.

### SNAPBP 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 버퍼 풀 논리 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPBP\_PART 관리 뷰와 함께 사용되는 경우, SNAPBP 관리 뷰는 GET SNAPSHOT FOR BUFFERPOOLS ON database-alias CLP 명령과 동등한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 전체 정보 목록은 546 페이지의 표 139의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPBP 관리 뷰에 대한 SELECT 특권
- SNAPBP 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한



또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_BP\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

현재 연결된 데이터베이스의 모든 버퍼 풀에 대한 데이터 및 인덱스 쓰기 횟수를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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

5개의 레코드가 선택되었습니다.

## SNAP\_GET\_BP\_V95 테이블 함수

SNAP\_GET\_BP\_V95 테이블 함수는 SNAPBP 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에 대한 정보는 검색할 수 있습니다.

SNAP\_GET\_BP\_PART 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_BP\_V95 테이블 함수는 GET SNAPSHOT FOR ALL BUFFERPOOLS CLP 명령과 동등한 데이터를 제공합니다.

리턴될 수 있는 전체 정보 목록은 546 페이지의 표 139의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_BP_V95 ( (dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR (128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 있는 파일이 존재하지 않으면, SNAP\_GET\_BP\_V95 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_BP\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

현재 연결된 데이터베이스 파티션의 모든 활성 데이터베이스에 대한 모든 버퍼 풀의 전체 실제 및 논리적 읽기를 검색하십시오.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	BP_NAME	TOTAL_LOGICAL_READS	...
SAMPLE	IBMDEFAULTBP	0	...
TOOLSDB	IBMDEFAULTBP	0	...
TOOLSDB	BP32K0000	0	...

3개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	TOTAL_PHYSICAL_READS	DBPARTITIONNUM
...	0	0
...	0	0
...	0	0

## 리턴되는 정보

표 169. SNAPBP 관리 뷰 및 SNAP\_GET\_BP\_V95 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
BP_NAME	VARCHAR (128)	bp_name - 버퍼 풀 이름
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
DB_PATH	VARCHAR(1024)	db_path - 데이터베이스 경로
INPUT_DB_ALIAS	VARCHAR (128)	input_db_alias - 데이터베이스 별명 입력
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수

표 169. SNAPBP 관리 뷰 및 SNAP\_GET\_BP\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 수
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 수
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - 버퍼 풀 비동기 읽기 시간
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - 버퍼 풀 비동기 쓰기 시간
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 수
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수

표 169. SNAPBP 관리 뷰 및 SNAP\_GET\_BP\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리 페치 페이지 수
FILES_CLOSED	BIGINT	files_closed - 닫힌 데이터베이스 파일 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - 버퍼 풀 희생 버퍼 없음
PAGES_FROM_BLOCK_IOS	BIGINT	pages_from_block_ios - 블록 입출력으로 읽은 총 페이지 수
PAGES_FROM_VECTORED_IOS	BIGINT	pages_from_vectored_ios - 벡터된 입출력으로 읽은 총 페이지 수
VECTORED_IOS	BIGINT	vectored_ios - 벡터된 입출력 요청 수
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPBP\_PART 관리 뷰 및 SNAP\_GET\_BP\_PART 테이블 함수 - bufferpool\_nodeinfo 논리 데이터 그룹 스냅샷 정보 검색

SNAPBP\_PART 관리 뷰 및 SNAP\_GET\_BP\_PART 테이블 함수는 버퍼 풀 스냅샷에서 버퍼 풀, 특히 bufferpool\_nodeinfo 논리 데이터 그룹에 대한 정보를 리턴합니다.

### SNAPBP\_PART 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 bufferpool\_nodeinfo 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPBP 관리 뷰와 함께 사용되는 경우, SNAPBP\_PART 관리 뷰는 GET SNAPSHOT FOR BUFFERPOOLS ON database-alias CLP 명령과 동등한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 552 페이지의 표 140의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPBP\_PART 관리 뷰에 대한 SELECT 특권
- SNAPBP\_PART 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_BP\_PART 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

SAMPLE 데이터베이스에 연결된 경우 모든 버퍼 풀에 대한 데이터를 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

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	...

4개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	BP_PAGES_LEFT_TO_REMOVE	BP_TBSP_USE_COUNT
...	0	3
...	0	0
...	0	0
...	0	0
...		

## SNAP\_GET\_BP\_PART 테이블 함수

SNAP\_GET\_BP\_PART 테이블 함수는 SNAPBP\_PART 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_BP\_V95 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_BP\_PART 테이블 함수는 GET SNAPSHOT FOR ALL BUFFERPOOLS CLP 명령과 동등한 데이터를 제공합니다.

리턴될 수 있는 정보의 완전한 목록은 552 페이지의 표 140의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_BP_PART ( ( dbname [ , dbpartitionnum ] ) )
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스와 동일한 인터페이스 내의 모든 데이터베이스에 있는 모든 버퍼 풀에 대한 스냅샷을 가져오려면 NULL 값을 지정하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_BP\_PART 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_BP\_PART 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

SAMPLE 데이터베이스에 연결된 경우 모든 활성 데이터베이스의 모든 버퍼풀에 대한 데이터를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...

...

이 쿼리의 출력(계속).

...	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

...



## 리턴되는 정보

표 170. SNAPBP\_PART 관리 뷰 및 SNAP\_GET\_BP\_PART 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
BP_NAME	VARCHAR (128)	bp_name - 버퍼 풀 이름
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
BP_CUR_BUFFSZ	BIGINT	bp_cur_buffsz - 버퍼 풀의 현재 크기
BP_NEW_BUFFSZ	BIGINT	bp_new_buffsz - 새 버퍼 풀 크기
BP_PAGES_LEFT_TO_REMOVE	BIGINT	bp_pages_left_to_remove - 제거할 나머지 페이지 수
BP_TBSP_USE_COUNT	BIGINT	bp_tbsp_use_count - 버퍼 풀에 맵핑된 테이블 스페이스 수
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPCONTAINER 관리 뷰 및 SNAP\_GET\_CONTAINER\_V91 테이블 함수 - tablespace\_container 논리 데이터 그룹 스냅샷 정보 검색

SNAPCONTAINER 관리 뷰 및 SNAP\_GET\_CONTAINER\_V91 테이블 함수는 tablespace\_container 논리 데이터 그룹에서 테이블 스페이스 스냅샷 정보를 리턴합니다.

### SNAPCONTAINER 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 tablespace\_container 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPTBSP, SNAPTBSP\_PART, SNAPTBSP\_QUIESCER 및 SNAPTBSP\_RANGE 관리 뷰와 함께 사용될 경우, SNAPCONTAINER 관리 뷰는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 556 페이지의 표 141의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPCONTAINER 관리 뷰에 대한 SELECT 특권
- SNAPCONTAINER 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_CONTAINER\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

현재 연결된 데이터베이스에 대한 모든 데이터베이스 파티션의 테이블 스페이스 컨테이너 세부사항을 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...
			...

8개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	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	...

이 쿼리의 출력(계속).

...	ACCESSIBLE	DBPARTITIONNUM
...	1	0

...	1	0
...	1	0
...	1	0
...	1	1
...	1	1
...	1	2
...	1	2

## SNAP\_GET\_CONTAINER\_V91 테이블 함수

SNAP\_GET\_CONTAINER\_V91 테이블 함수는 SNAPCONTAINER 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_TBSP\_V91, SNAP\_GET\_TBSP\_PART\_V91, SNAP\_GET\_TBSP\_QUIESCER 및 SNAP\_GET\_TBSP\_RANGE 테이블 함수와 함께 사용될 경우, SNAP\_GET\_CONTAINER\_V91 테이블 함수는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 데이터를 리턴합니다.

리턴 가능한 전체 정보 목록은 556 페이지의 표 141의 내용을 참조하십시오.

## 구문

```

▶▶ SNAP_GET_CONTAINER_V91 ( (dbname [ , dbpartitionnum ] ) )

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. NULL 또는 빈 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 얻으십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결에 사용할 수 있으며 응용프로그램에서 사용할 수 있는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_CONTAINER\_V91 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_CONTAINER\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 데이터베이스 파티션에서 현재 연결된 데이터베이스의 테이블 스페이스 컨테이너 세부사항을 검색합니다.

```
SELECT SNAPSHOT_TIMESTAMP, TBSP_NAME, TBSP_ID, CONTAINER_NAME,
       CONTAINER_ID, CONTAINER_TYPE, ACCESSIBLE
FROM TABLE(SNAP_GET_CONTAINER_V91(' ', -1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

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	WHATNEWTEMPSPACE	5	...

이 쿼리의 출력(계속).

...	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:#DGTTsWhatsNewContainer	0	...

이 쿼리의 출력(계속).

```

... CONTAINER_TYPE ACCESSIBLE
... -----
... CONT_PATH          1
... CONT_PATH          1
... CONT_PATH          1
... CONT_PATH          1
... CONT_PATH          1
... CONT_PATH          1

```

## 리턴되는 정보

메모: 파일 시스템 정보를 리턴하려면 BUFFERPOOL 데이터베이스 관리 프로그램 모니터 스위치를 설정해야 합니다.

표 171. SNAPCONTAINER 관리 뷰 및 SNAP\_GET\_CONTAINER\_V91 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TBSP_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
CONTAINER_NAME	VARCHAR(256)	container_name - 컨테이너 이름
CONTAINER_ID	BIGINT	container_id - 컨테이너 ID
CONTAINER_TYPE	VARCHAR(16)	container_type - 컨테이너 유형. 이는 sqlutil.h의 정의를 기반으로 하는 텍스트 ID로 다음 중 하나입니다. <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 - 컨테이너의 전체 페이지 수
USABLE_PAGES	BIGINT	container_usable_pages - 컨테이너에서 사용할 수 있는 페이지 수
ACCESSIBLE	SMALLINT	container_accessible - 컨테이너의 액세스 가능성
STRIPE_SET	BIGINT	container_stripe_set - 스트라이프 세트
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
FS_ID	VARCHAR(22)	fs_id - 고유 파일 시스템 ID 번호
FS_TOTAL_SIZE	BIGINT	fs_total_size - 파일 시스템의 전체 크기
FS_USED_SIZE	BIGINT	fs_used_size - 파일 시스템에서 사용된 스페이스 용량

---

## SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수 - dbase 논리 그룹에서 스냅샷 정보 검색

557 페이지의 『SNAPDB 관리 뷰』 및 558 페이지의 『SNAP\_GET\_DB\_V95 테이블 함수』에서는 데이터베이스(dbase) 논리 그룹으로부터 스냅샷 정보를 리턴합니다.

### SNAPDB 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 dbase 논리 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPDB\_MEMORY\_POOL, SNAPDETAILLOG, SNAPHADR 및 SNAPSTORAGE\_PATHS 관리 뷰와 함께 사용되는 경우, SNAPDB 관리 뷰는 GET SNAPSHOT FOR DATABASE on database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴되는 전체 정보 목록은 561 페이지의 표 142의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPDB 관리 뷰에 대한 SELECT 특권
- SNAPDB 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_DB\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### 예:

현재 연결된 데이터베이스의 모든 데이터베이스 파티션에 대한 상태, 플랫폼, 위치 및 연결 시간을 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	DB_STATUS	SERVER_PLATFORM	DB_LOCATION	...
TEST	ACTIVE	AIX64	LOCAL	...
TEST	ACTIVE	AIX64	LOCAL	...
TEST	ACTIVE	AIX64	LOCAL	...

3개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

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

명령행에 다음을 호출하여 이 루틴을 사용할 수 있습니다.

```
SELECT TOTAL_OLAP_FUNCS, OLAP_FUNC_OVERFLOW, ACTIVE_OLAP_FUNCS
FROM SYSIBMADM.SNAPDB
```

TOTAL_OLAP_FUNCS	OLAP_FUNC_OVERFLOW	ACTIVE_OLAP_FUNCS
7	2	1

1개의 레코드가 선택되었습니다.

워크로드를 실행한 후 사용자가 다음 쿼리를 사용할 수 있습니다.

```
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

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_DB\_V95 테이블 함수

SNAP\_GET\_DB\_V95 테이블 함수는 SNAPDB 관리 뷰와 동일한 정보를 리턴합니다.

SNAP\_GET\_DB\_MEMORY\_POOL, SNAP\_GET\_DETAILLOG\_V91, SNAP\_GET\_HADR 및 SNAP\_GET\_STORAGE\_PATHS 테이블 함수와 함께 사용

되는 경우, SNAP\_GET\_DB\_V95 테이블 함수는 GET SNAPSHOT FOR ALL DATABASES CLP 명령과 동등한 정보를 제공합니다.

리턴되는 전체 정보 목록은 561 페이지의 표 142의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_DB_V95 ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 있는 파일이 존재하지 않으면, SNAP\_GET\_DB\_V95 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_DB\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한



또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

**예:**

예 1: 현재 연결된 데이터베이스의 모든 데이터베이스 파티션에서 집계 뷰로 상태, 플랫폼, 위치 및 연결 시간을 검색합니다.

```
SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,
       DB_LOCATION, DB_CONN_TIME FROM TABLE(SNAP_GET_DB_V95(' ', -2)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	DB_STATUS	SERVER_PLATFORM	...
-----	-----	-----	...
SAMPLE	ACTIVE	AIX64	...

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	DB_LOCATION	DB_CONN_TIME
...	-----	-----
...	LOCAL	2005-07-24-22.09.22.013196

예 2: 현재 연결된 데이터베이스가 포함된 동일한 인스턴스에서 모든 활성 데이터베이스에 대해 모든 데이터베이스 파티션에서 집계 뷰로 상태, 플랫폼, 위치 및 연결 시간을 검색합니다.

```
SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,
       DB_LOCATION, DB_CONN_TIME
FROM TABLE(SNAP_GET_DB_V95(CAST (NULL AS VARCHAR
(128)), -2)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	DB_STATUS	SERVER_PLATFORM	...
-----	-----	-----	...
TOOLSDB	ACTIVE	AIX64	...
SAMPLE	ACTIVE	AIX64	...

이 쿼리의 출력(계속).

...	DB_LOCATION	DB_CONN_TIME
...	-----	-----
...	LOCAL	2005-07-24-22.26.54.396335
...	LOCAL	2005-07-24-22.09.22.013196

예 3: 명령행에 다음을 호출하여 이 루틴을 사용할 수 있습니다.

데이터베이스에 연결된 경우:

```
SELECT TOTAL_OLAP_FUNCS, OLAP_FUNC_OVERFLOW, ACTIVE_OLAP_FUNCS
FROM TABLE (SNAP_GET_DB_V95(' ', 0)) AS T
```

출력은 다음과 같습니다.

```
TOTAL_OLAP_FUNCS  OLAP_FUNC_OVERFLOW  ACTIVE_OLAP_FUNCS
-----
                          7                          2                          1
```

1개의 레코드가 선택되었습니다.

예 4: 워크로드를 실행한 후 사용자가 테이블 함수와 함께 다음 쿼리를 사용할 수 있습니다.

```
SELECT STATS_CACHE_SIZE, STATS_FABRICATIONS, SYNC_RUNSTATS,
ASYNC_RUNSTATS, STATS_FABRICATE_TIME, SYNC_RUNSTATS_TIME
FROM TABLE (SNAP_GET_DB_V95('mytestdb', -1)) AS SNAPDB
```

```
STATS_CACHE_SIZE  STATS_FABRICATIONS  SYNC_RUNSTATS  ASYNC_RUNSTATS  ...
-----
                200                1                2                0  ...
```

계속됨

```
...STATS_FABRICATE_TIME  SYNC_RUNSTATS_TIME
...-----
...                2                32
```

1개의 레코드가 선택되었습니다.

## SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수 메타데이터

표 172. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
DB_PATH	VARCHAR(1024)	db_path - 데이터베이스 경로
INPUT_DB_ALIAS	VARCHAR (128)	input_db_alias - 데이터베이스 별명 입력
DB_STATUS	VARCHAR(12)	db_status - 데이터베이스 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>ACTIVE</li> <li>QUIESCE_PEND</li> <li>QUIESCED</li> <li>ROLLFWD</li> </ul>
CATALOG_PARTITION	SMALLINT	catalog_node - 카탈로그 노드 번호
CATALOG_PARTITION_NAME	VARCHAR (128)	catalog_node_name - 카탈로그 노드 네트워크 이름

표 172. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SERVER_PLATFORM	VARCHAR(12)	<p>server_platform - 서버 운영 체제. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다.</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>

표 172. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DB_LOCATION	VARCHAR(12)	db_location - 데이터베이스 위치. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• LOCAL</li> <li>• REMOTE</li> </ul>
DB_CONN_TIME	TIMESTAMP	db_conn_time - 데이터베이스 활성화 시간소인
LAST_RESET	TIMESTAMP	last_reset - 마지막 재설정 시간소인
LAST_BACKUP	TIMESTAMP	last_backup - 최종 백업 시간소인
CONNECTIONS_TOP	BIGINT	connections_top - 최대 동시 연결 수
TOTAL_CONS	BIGINT	total_cons - 데이터베이스 활성화 이후 연결 수
TOTAL_SEC_CONS	BIGINT	total_sec_cons - 2차 연결 수
APPLS_CUR_CONS	BIGINT	appls_cur_cons - 현재 연결된 응용프로그램 수
APPLS_IN_DB2	BIGINT	appls_in_db2 - 현재 데이터베이스에서 실행 중인 응용프로그램 수
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 연관 에이전트 수
AGENTS_TOP	BIGINT	agents_top - 작성된 에이전트 수
COORD_AGENTS_TOP	BIGINT	coord_agents_top - 최대 코디네이팅 에이전트 수
LOCKS_HELD	BIGINT	locks_held - 보유된 잠금 수
LOCK_WAITS	BIGINT	lock_waits - 잠금 대기 수
LOCK_WAIT_TIME	BIGINT	lock_wait_time - 잠금 대기 시간
LOCK_LIST_IN_USE	BIGINT	lock_list_in_use - 사용 중인 전체 잠금 목록 메모리
DEADLOCKS	BIGINT	deadlocks - 발견된 교착 상태 수
LOCK_ESCALS	BIGINT	lock_escals - 잠금 에스컬레이션 수
X_LOCK_ESCALS	BIGINT	x_lock_escals - 배타적 잠금 에스컬레이션 수
LOCKS_WAITING	BIGINT	locks_waiting - 잠금 대기 중인 현재 에이전트 수
LOCK_TIMEOUTS	BIGINT	lock_timeouts - 잠금 시간종료 수
NUM_INDOUBT_TRANS	BIGINT	num_indoubt_trans - 인다우트(Indoubt) 트랜잭션 수
SORT_HEAP_ALLOCATED	BIGINT	sort_heap_allocated - 할당된 전체 정렬 힙
SORT_SHRHEAP_ALLOCATED	BIGINT	sort_shrheap_allocated - 현재 할당된 정렬 공유 힙
SORT_SHRHEAP_TOP	BIGINT	sort_shrheap_top - 공유 힙 상위 워터 마크(water mark) 정렬
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - 포스트 공유 임계값 정렬 수
TOTAL_SORTS	BIGINT	total_sorts - 전체 정렬 수
TOTAL_SORT_TIME	BIGINT	total_sort_time - 총 정렬 시간
SORT_OVERFLOWS	BIGINT	sort_overflows - 정렬 오버플로우 수
ACTIVE_SORTS	BIGINT	active_sorts - 활성 정렬 수
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수

표 172. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - 버퍼 풀 비동기 읽기 시간
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - 버퍼 풀 비동기 쓰기 시간
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 수

표 172. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - 버퍼 풀 희생 버퍼 없음
POOL_LSN_GAP_CLNS	BIGINT	pool_lsn_gap_clns - 버퍼 풀 로그 스페이스 클리너 트리거
POOL_DRTY_PG_STEAL_CLNS	BIGINT	pool_drty_pg_steal_clns - 버퍼 풀 희생(victim) 페이지 클리너 트리거
POOL_DRTY_PG_THRSH_CLNS	BIGINT	pool_drty_pg_thrsh_clns - 버퍼 풀 임계값 클리너 트리거
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - 프리페치 대기 시간
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리페치 페이지 수
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
FILES_CLOSED	BIGINT	files_closed - 닫힌 데이터베이스 파일 수
ELAPSED_EXEC_TIME_S	BIGINT	elapsed_exec_time - 명령문 실행 경과 시간
ELAPSED_EXEC_TIME_MS	BIGINT	elapsed_exec_time - 명령문 실행 경과 시간
COMMIT_SQL_STMTS	BIGINT	commit_sql_stmts - 시도된 명령문 커밋
ROLLBACK_SQL_STMTS	BIGINT	rollback_sql_stmts - 시도된 롤백 명령문
DYNAMIC_SQL_STMTS	BIGINT	dynamic_sql_stmts - 시도된 동적 SQL문
STATIC_SQL_STMTS	BIGINT	static_sql_stmts - 시도된 정적 SQL문
FAILED_SQL_STMTS	BIGINT	failed_sql_stmts - 실패한 명령문 조작
SELECT_SQL_STMTS	BIGINT	select_sql_stmts - 실행된 Select SQL문
UID_SQL_STMTS	BIGINT	uid_sql_stmts - 실행된 UPDATE/INSERT/DELETE SQL문
DDL_SQL_STMTS	BIGINT	ddl_sql_stmts - DDL(Data Definition Language) SQL문
INT_AUTO_REBINDS	BIGINT	int_auto_rebinds - 내부 자동 리바인드 수
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 삭제된 내부 행 수
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 삽입된 내부 행 수
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 갱신된 내부 행 수
INT_COMMITS	BIGINT	int_commits - 내부 커밋 수
INT_ROLLBACKS	BIGINT	int_rollback - 내부 롤백 수

표 172. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
INT_DEADLOCK_ROLLBACKS	BIGINT	int_deadlock_rollback - 교착 상태로 인한 내부 롤백 수
ROWS_DELETED	BIGINT	rows_deleted - 삭제된 행 수
ROWS_INSERTED	BIGINT	rows_inserted - 삽입된 행 수
ROWS_UPDATED	BIGINT	rows_updated - 갱신된 행 수
ROWS_SELECTED	BIGINT	rows_selected - 선택된 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
BINDS_PRECOMPILES	BIGINT	binds_precompiles - 시도된 바인드/프리컴파일 수
TOTAL_LOG_AVAILABLE	BIGINT	total_log_available - 사용 가능한 전체 로그
TOTAL_LOG_USED	BIGINT	total_log_used - 사용된 전체 로그 스페이스
SEC_LOG_USED_TOP	BIGINT	sec_log_used_top - 사용된 최대 2차 로그 스페이스
TOT_LOG_USED_TOP	BIGINT	tot_log_used_top - 사용된 최대 전체 로그 스페이스
SEC_LOGS_ALLOCATED	BIGINT	sec_logs_allocated - 현재 할당된 2차 로그
LOG_READS	BIGINT	log_reads - 읽은 로그 페이지 수
LOG_READ_TIME_S	BIGINT	log_read_time - 로그 읽기 시간
LOG_READ_TIME_NS	BIGINT	log_read_time - 로그 읽기 시간
LOG_WRITES	BIGINT	log_writes - 쓴 로그 페이지 수
LOG_WRITE_TIME_S	BIGINT	log_write_time - 로그 쓰기 시간
LOG_WRITE_TIME_NS	BIGINT	log_write_time - 로그 쓰기 시간
NUM_LOG_WRITE_IO	BIGINT	num_log_write_io - 로그 쓰기 수
NUM_LOG_READ_IO	BIGINT	num_log_read_io - 로그 읽기 수
NUM_LOG_PART_PAGE_IO	BIGINT	num_log_part_page_io - 부분 로그 페이지 쓰기 수
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - 전체 로그 버퍼 수
NUM_LOG_DATA_FOUND_IN_BUFFER	BIGINT	num_log_data_found_in_buffer - 버퍼에 있는 로그 데이터 수
APPL_ID_OLDEST_XACT	BIGINT	appl_id_oldest_xact - 가장 오래된 트랜잭션이 포함된 응용프로그램
LOG_TO_REDO_FOR_RECOVERY	BIGINT	log_to_redo_for_recovery - 복구를 위해 재실행되는 로그 수량
LOG_HELD_BY_DIRTY_PAGES	BIGINT	log_held_by_dirty_pages - 더티 페이지별로 계산되는 로그 스페이스 수량
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - 패키지 캐시 찾아보기 수
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - 패키지 캐시 삽입 수
PKG_CACHE_NUM_OVERFLOW	BIGINT	pkg_cache_num_overflows - 패키지 캐시 오버플로우 수
PKG_CACHE_SIZE_TOP	BIGINT	pkg_cache_size_top - 패키지 캐시 상위 워터 마크 (water mark)
APPL_SECTION_LOOKUPS	BIGINT	appl_section_lookups - 섹션 찾아보기 수

표 172. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
APPL_SECTION_INSERTS	BIGINT	appl_section_inserts - 섹션 삽입 수
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - 카탈로그 캐시 찾기 수
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - 카탈로그 캐시 삽입 수
CAT_CACHE_OVERFLOWS	BIGINT	cat_cache_overflows - 카탈로그 캐시 오버플로우 수
CAT_CACHE_SIZE_TOP	BIGINT	cat_cache_size_top - 카탈로그 캐시 상위 워터 마크(water mark)
PRIV_WORKSPACE_SIZE_TOP	BIGINT	priv_workspace_size_top - 개인용 작업 스페이스 최대 크기
PRIV_WORKSPACE_NUM_OVERFLOWS	BIGINT	priv_workspace_num_overflows - 개인용 작업 스페이스 오버플로우 수
PRIV_WORKSPACE_SECTION_INSERTS	BIGINT	priv_workspace_section_inserts - 개인용 작업 스페이스 섹션 삽입 수
PRIV_WORKSPACE_SECTION_LOOKUPS	BIGINT	priv_workspace_section_lookups - 개인용 작업 스페이스 섹션 찾기 수
SHR_WORKSPACE_SIZE_TOP	BIGINT	shr_workspace_size_top - 공유 작업 스페이스 최대 크기
SHR_WORKSPACE_NUM_OVERFLOWS	BIGINT	shr_workspace_num_overflows - 공유 작업 스페이스 오버플로우 수
SHR_WORKSPACE_SECTION_INSERTS	BIGINT	shr_workspace_section_inserts - 공유 작업 스페이스 섹션 삽입 수
SHR_WORKSPACE_SECTION_LOOKUPS	BIGINT	shr_workspace_section_lookups - 공유 작업 스페이스 섹션 찾기 수
TOTAL_HASH_JOINS	BIGINT	total_hash_joins - 전체 해시 조인 수
TOTAL_HASH_LOOPS	BIGINT	total_hash_loops - 전체 해시 루프 수
HASH_JOIN_OVERFLOWS	BIGINT	hash_join_overflows - 해시 조인 오버플로우 수
HASH_JOIN_SMALL_OVERFLOWS	BIGINT	hash_join_small_overflows - 해시 조인 작은 오버플로우 수
POST_SHRTHRESHOLD_HASH_JOINS	BIGINT	post_shrthreshold_hash_joins - 포스트 임계값 해시 조인 수
ACTIVE_HASH_JOINS	BIGINT	active_hash_joins - 활성 해시 조인 수
NUM_DB_STORAGE_PATHS	BIGINT	num_db_storage_paths - 자동 스토리지 경로 수
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
SMALLEST_LOG_AVAIL_NODE	INTEGER	smallest_log_avail_node - 최소 사용 가능 로그 스페이스가 포함된 노드
TOTAL_OLAP_FUNCS	BIGINT	실행된 총 OLAP 함수 수



표 172. SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
OLAP_FUNC_OVERFLOWS	BIGINT	OLAP 함수 데이터가 사용 가능한 정렬 힙 스페이스를 초과한 횟수
ACTIVE_OLAP_FUNCS	BIGINT	정렬 힙 메모리를 현재 실행하여 소비 중인 전체 OLAP 함수의 수
STATS_CACHE_SIZE	BIGINT	통계 캐시의 크기(바이트)
STATS_FABRICATIONS	BIGINT	테이블 또는 인덱스 스캔 없이 시스템이 통계를 작성하기 위한 전체 통계 수집 활동 수
SYNC_RUNSTATS	BIGINT	쿼리 컴파일 중의 전체 동기 통계 수집 활동 수
ASYNC_RUNSTATS	BIGINT	이 컬럼을 완료된 전체 비동기 통계 수집 활동 수로 변경합니다.
STATS_FABRICATE_TIME	BIGINT	쿼리 컴파일 중에 테이블 또는 인덱스 스캔 없이 통계를 작성하는데 시스템이 소비한 밀리초 단위의 전체 시간
SYNC_RUNSTATS_TIME	BIGINT	동기 통계 수집 활동에 소비된 밀리초 단위의 전체 시간
NUM_THRESHOLD_VIOLATIONS	BIGINT	데이터베이스에 발생한 임계값 위반 수

## SNAPDB\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수 - 데이터베이스 레벨 메모리 사용 정보 검색

SNAPDB\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수는 UNIX 플랫폼에만 해당하는 데이터베이스 레벨에서 메모리 사용에 대한 정보를 리턴합니다.

### SNAPDB\_MEMORY\_POOL 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 데이터베이스 레벨 메모리 사용 정보를 검색할 수 있습니다.

SNAPDB, SNAPDETAILLOG, SNAPHDR 및 SNAPSTORAGE\_PATHS 관리 뷰와 함께 사용되는 경우, SNAPDB\_MEMORY\_POOL 관리 뷰는 GET SNAPSHOT FOR DATABASE ON database-alias CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 571 페이지의 표 143의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPDB\_MEMORY\_POOL 관리 뷰에 대한 SELECT 특권
- SNAPDB\_MEMORY\_POOL 관리 뷰에 대한 CONTROL 특권

- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

**예 :**

현재 연결된 데이터베이스, SAMPLE에 대한 메모리 풀 및 현재 크기 목록을 검색합니다.

```
SELECT POOL_ID, POOL_CUR_SIZE FROM SYSIBMADM.SNAPDB_MEMORY_POOL
```

이 쿼리의 출력 예는 다음과 같습니다.

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

12개의 레코드가 선택되었습니다.

## **SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수**

SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수는 SNAPDB\_MEMORY\_POOL 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 특정 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_DB\_V95, SNAP\_GET\_DETAILLOG\_V91, SNAP\_GET\_HADR 및 SNAP\_GET\_STORAGE\_PATHS 테이블 함수와 함께 사용되는 경우,

SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수는 GET SNAPSHOT FOR ALL DATABASES CLP 명령과 동등한 정보를 제공합니다.

리턴될 수 있는 정보의 완전한 목록은 571 페이지의 표 143의 내용을 참조하십시오.

## 구문

```
▶▶—SNAP_GET_DB_MEMORY_POOL—(—dbname— , dbpartitionnum— )▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### dbname

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계 -2를 지정하십시오. dbname이 NULL로 설정되어 있지 않고 dbpartitionnum이 NULL로 설정되어 있다면, -1은 내재적으로 dbpartitionnum으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, dbname만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

dbname과 dbpartitionnum이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

예 :

모든 데이터베이스에 대한 메모리 풀 및 현재 크기 목록을 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

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

24개의 레코드가 선택되었습니다.

### 리턴되는 정보

표 173. SNAPDB\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름

표 173. SNAPDB\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_ID	VARCHAR(14)	pool_id - 메모리 풀 ID. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 메모리 풀 2차 ID
POOL_CUR_SIZE	BIGINT	pool_cur_size - 메모리 풀의 현재 크기
POOL_WATERMARK	BIGINT	pool_watermark - 메모리 풀 워터 마크 (water mark)
POOL_CONFIG_SIZE	BIGINT	pool_config_size - 메모리 풀의 구성된 크기
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPDBM 관리 뷰 및 SNAP\_GET\_DBM\_V95 테이블 함수 - dbm 논리적 그룹화 스냅샷 정보 검색

SNAPDBM 관리 뷰 및 SNAP\_GET\_DBM\_V95 테이블 함수는 스냅샷 모니터 DB2 데이터베이스 관리 프로그램(dbm) 논리적 그룹화 정보를 리턴합니다.

## SNAPDBM 관리 뷰

SNAPDBM\_MEMORY\_POOL, SNAPFCM, SNAPFCM\_PART 및 SNAPSWITCHES 관리 뷰와 같이 사용하여 SNAPDBM 관리 뷰는 GET SNAPSHOT FOR DBM 명령과 동등한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 575 페이지의 표 144의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPDBM 관리 뷰에 대한 SELECT 특권
- SNAPDBM 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_DBM\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

### 예 :

모든 데이터베이스 파티션에 대해 데이터베이스 관리 프로그램 상태 및 연결 정보를 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

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	-	...

3개의 레코드가 선택되었습니다.

...

이 쿼리의 출력(계속).

...	LOCAL_CONS	REM_CONS_IN	AGENT_USAGE	DBPARTITIONNUM
...	1	1	0	0
...	0	0	0	1
...	0	0	0	2

## SNAP\_GET\_DBM\_V95 테이블 함수

SNAP\_GET\_DBM\_V95 테이블 함수는 SNAPDBM 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에 대한 정보를 검색할 수도 있습니다.

SNAP\_GET\_DBM\_MEMORY\_POOL, SNAP\_GET\_FCM, SNAP\_GET\_FCM\_PART 및 SNAP\_GET\_SWITCHES 테이블 함수와 함께 사용되는 SNAP\_GET\_DBM\_V95 테이블 함수는 GET SNAPSHOT FOR DBM 명령과 동등한 데이터를 제공합니다.

리턴 가능한 전체 정보 목록은 575 페이지의 표 144의 내용을 참조하십시오.

## 구문

```

▶▶ SNAP_GET_DBM_V95 ( ( dbpartitionnum ) )

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_DBM\_V95 테이블 함수가 메모리에서 스냅샷을 호출합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_DBM\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

예 :

데이터베이스 파티션 번호 2의 시작 시간 및 현재 상태를 검색합니다.

```
SELECT DB2START_TIME, DB2_STATUS FROM TABLE(SNAP_GET_DBM_V95(2)) AS T
```

다음은 이 쿼리의 출력 예입니다.

```
DB2START_TIME          DB2_STATUS
-----
2006-01-06-14.59.59.062798 ACTIVE
```

### 리턴된 정보

표 174. SNAPDBM 관리 뷰 및 SNAP\_GET\_DBM\_V95 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
SORT_HEAP_ALLOCATED	BIGINT	sort_heap_allocated - 할당된 전체 정렬 힙
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - 포스트 임계값 정렬 수
PIPED_SORTS_REQUESTED	BIGINT	piped_sorts_requested - 요청된 파이프 정렬 수
PIPED_SORTS_ACCEPTED	BIGINT	piped_sorts_accepted - 수락된 파이프 정렬 수
REM_CONS_IN	BIGINT	rem_cons_in - 데이터베이스 관리 프로그램에 리모트 연결 수
REM_CONS_IN_EXEC	BIGINT	rem_cons_in_exec - 데이터베이스 관리 프로그램 모니터 요소에서 실행되는 리모트 연결 수
LOCAL_CONS	BIGINT	local_cons - 로컬 연결 수
LOCAL_CONS_IN_EXEC	BIGINT	local_cons_in_exec - 데이터베이스 관리 프로그램 모니터 요소에서 실행되는 로컬 연결 수
CON_LOCAL_DBASES	BIGINT	con_local_dbases - 현재 연결된 로컬 데이터베이스 수
AGENTS_REGISTERED	BIGINT	agents_registered - 등록된 에이전트 수
AGENTS_WAITING_ON_TOKEN	BIGINT	agents_waiting_on_token - 토큰을 기다리는 에이전트 수
DB2_STATUS	VARCHAR(12)	db2_status - DB2 인스턴스 상태  이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• ACTIVE</li> <li>• QUIESCE_PEND</li> <li>• QUIESCED</li> </ul>
AGENTS_REGISTERED_TOP	BIGINT	agents_registered_top - 등록된 최대 에이전트 수
AGENTS_WAITING_TOP	BIGINT	agents_waiting_top - 대기 중인 최대 에이전트 수



표 174. SNAPDBM 관리 뷰 및 SNAP\_GET\_DBM\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
COMM_PRIVATE_MEM	BIGINT	comm_private_mem - 커미트된 전용 메모리
IDLE_AGENTS	BIGINT	idle_agents - 유휴 에이전트 수
AGENTS_FROM_POOL	BIGINT	agents_from_pool - 풀로부터 할당된 에이전트 수
AGENTS_CREATED_EMPTY_POOL	BIGINT	agents_created_empty_pool - 비어 있는 에이전트 풀로 인해 작성된 에이전트 수
COORD_AGENTS_TOP	BIGINT	coord_agents_top - 최대 코디네이팅 에이전트 수
MAX_AGENT_OVERFLOW	BIGINT	max_agent_overflows - 최대 에이전트 오버플로우 수
AGENTS_STOLEN	BIGINT	agents_stolen - 분실 에이전트 수
GW_TOTAL_CONS	BIGINT	gw_total_cons - DB2 Connect에 대한 전체 연결 시도 수
GW_CUR_CONS	BIGINT	gw_cur_cons - DB2 Connect에 대한 현재 연결 수
GW_CONS_WAIT_HOST	BIGINT	gw_cons_wait_host - 호스트 응답을 대기 중인 연결 수
GW_CONS_WAIT_CLIENT	BIGINT	gw_cons_wait_client - 클라이언트의 요청 전송을 대기 중인 연결 수
POST_THRESHOLD_HASH_JOINS	BIGINT	post_threshold_hash_joins - 해시 조인 임계값
NUM_GW_CONN_SWITCHES	BIGINT	num_gw_conn_switches - 연결 전환 수
DB2START_TIME	TIMESTAMP	db2start_time - 데이터베이스 관리 프로그램 시작 시간소인
LAST_RESET	TIMESTAMP	last_reset - 마지막 재설정 시간소인
NUM_NODES_IN_DB2_INSTANCE	INTEGER	num_nodes_in_db2_instance - 데이터베이스 파티션의 노드 수
PRODUCT_NAME	VARCHAR(32)	product_name - 제품 이름
SERVICE_LEVEL	VARCHAR(18)	service_level - 서비스 레벨
SORT_HEAP_TOP	BIGINT	sort_heap_top - 전용 힙 상위 워터 마크(water mark) 정렬
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

표 174. SNAPDBM 관리 뷰 및 SNAP\_GET\_DBM\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POST_THRESHOLD_OLAP_FUNCS	BIGINT	정렬 힙 임계값이 초과된 후에 정렬 힙을 요청한 OLAP 함수의 수입니다.  정렬 힙을 사용하는 조건의 예로는 정렬, 해시 조인 및 OLAP 함수가 있습니다. 일반적인 조건인 경우 데이터베이스 관리 프로그램은 <code>sorheap</code> 구성 매개변수로 지정한 값을 사용하여 정렬 힙을 할당합니다. 정렬 힙에 할당된 메모리 양이 정렬 힙 임계값( <code>sheapthres</code> 구성 매개변수)을 초과하는 경우 데이터베이스 관리 프로그램은 <code>sorheap</code> 구성 매개변수로 지정한 값보다 작은 값을 사용하여 후속 정렬 힙을 할당합니다.  정렬 힙 임계값에 도달한 후에 시작된 OLAP 함수는 실행할 최적의 메모리 양을 수신하지 못할 수도 있습니다.

## SNAPDBM\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수 - 데이터베이스 관리 프로그램 레벨 메모리 사용 정보 검색

SNAPDBM\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수는 데이터베이스 관리 프로그램에서 메모리 사용에 대한 정보를 리턴합니다.

### SNAPDBM\_MEMORY\_POOL 관리 뷰

SNAPDBM, SNAPFCM, SNAPFCM\_PART 및 SNAPSWITCHES 관리 뷰와 함께 사용되는 경우, SNAPDBM\_MEMORY\_POOL 관리 뷰는 GET SNAPSHOT FOR DBM 명령과 동등한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 580 페이지의 표 145의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPDBM\_MEMORY\_POOL 관리 뷰에 대한 SELECT 특권
- SNAPDBM\_MEMORY\_POOL 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

연결된 데이터베이스의 데이터베이스 관리 프로그램에 대한 메모리 풀 및 현재 크기 목록을 검색합니다.

```
SELECT POOL_ID, POOL_CUR_SIZE FROM SNAPDBM_MEMORY_POOL
```

이 쿼리의 출력 예는 다음과 같습니다.

POOL_ID	POOL_CUR_SIZE
MONITOR	65536
OTHER	29622272
FCMBP	57606144
...	

## SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수

SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수는

SNAPAGENT\_MEMORY\_POOL 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_DBM\_V95, SNAP\_GET\_FCM, SNAP\_GET\_FCM\_PART 및 SNAP\_GET\_SWITCHES 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수는 GET SNAPSHOT FOR DBM 명령과 동등한 데이터를 제공합니다.

리턴 가능한 전체 정보 목록은 580 페이지의 표 145의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_DBM_MEMORY_POOL ( ( dbpartitionnum ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### 예 :

연결된 데이터베이스의 데이터베이스 관리 프로그램의 모든 데이터베이스 파티션에 대한 메모리 풀 및 현재 크기 목록을 검색합니다.

```
SELECT POOL_ID, POOL_CUR_SIZE, DBPARTITIONNUM
      FROM TABLE(SYSPROC.SNAP_GET_DBM_MEMORY_POOL())
      AS T ORDER BY DBPARTITIONNUM
```

다음은 이 쿼리의 출력 예입니다.

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

## 리턴되는 정보

표 175. SNAPDBM\_MEMORY\_POOL 관리 뷰 및 SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
POOL_ID	VARCHAR(14)	pool_id - 메모리 풀 ID. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 메모리 풀의 현재 크기
POOL_WATERMARK	BIGINT	pool_watermark - 메모리 풀 워터 마크 (water mark)
POOL_CONFIG_SIZE	BIGINT	pool_config_size - 메모리 풀의 구성된 크기
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

---

## SNAPDETAILLOG 관리 뷰 및 SNAP\_GET\_DETAILLOG\_V91 테이블 함수 - 논리 데이터 그룹에서 스냅샷 정보 검색

SNAPDETAILLOG 관리 뷰 및 SNAP\_GET\_DETAILLOG\_V91 테이블 함수는 detail\_log 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.

### SNAPDETAILLOG 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 detail\_log 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPDB, SNAPDB\_MEMORY\_POOL, SNAPHADR 및 SNAPSTORAGE\_PATHS 관리 뷰와 함께 사용되는 경우, SNAPDETAILLOG 관리 뷰는 GET SNAPSHOT FOR DATABASE on database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴되는 전체 정보 목록은 584 페이지의 표 146의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPDETAILLOG 관리 뷰에 대한 SELECT 특권
- SNAPDETAILLOG 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_DETAILLOG\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

### 예 :

현재 연결된 데이터베이스에 대한 모든 데이터베이스 파티션의 로그 정보를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	FIRST_ACTIVE_LOG	LAST_ACTIVE_LOG	...
TEST	0	8	...
TEST	0	8	...
TEST	0	8	...

3개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	CURRENT_ACTIVE_LOG	CURRENT_ARCHIVE_LOG	DBPARTITIONNUM
...	0	-	0
...	0	-	1
...	0	-	2

## SNAP\_GET\_DETAILLOG\_V91 테이블 함수

SNAP\_GET\_DETAILLOG\_V91 테이블 함수는 SNAPDETAILLOG 관리 뷰와 동일한 정보를 리턴합니다.

SNAP\_GET\_DB\_V95, SNAP\_GET\_DB\_MEMORY\_POOL, SNAP\_GET\_HADR 및 SNAP\_GET\_STORAGE\_PATHS 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_DETAILLOG 테이블 함수는 GET SNAPSHOT FOR ALL DATABASES CLP 명령과 동등한 정보를 제공합니다.

리턴되는 전체 정보 목록은 584 페이지의 표 146의 내용을 참조하십시오.

## 구문

```

▶▶ SNAP_GET_DETAILLOG_V91 ( ( dbname ) [ , dbpartitionnum ] )

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수.

현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결에 사용할 수 있으며 응용프로그램에서 사용할 수 있는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_DETAILLOG\_V91 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_DETAILLOG\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### 예 :

현재 연결된 데이터베이스에 대한 데이터베이스 파티션 1의 로그 정보를 검색합니다.

```
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', 1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	FIRST_ACTIVE_LOG	LAST_ACTIVE_LOG	...
TEST	0	8	...

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	CURRENT_ACTIVE_LOG	CURRENT_ARCHIVE_LOG
...	0	-



## SNAPDETAILLOG 관리 뷰 및 SNAP\_GET\_DETAILLOG\_V91 테이블 함수 메타데이터

표 176. SNAPDETAILLOG 관리 뷰 및 SNAP\_GET\_DETAILLOG\_V91 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
FIRST_ACTIVE_LOG	BIGINT	first_active_log - 처음에 사용되는 로그 파일 번호
LAST_ACTIVE_LOG	BIGINT	last_active_log - 마지막에 사용되는 로그 파일 번호
CURRENT_ACTIVE_LOG	BIGINT	current_active_log - 현재 사용 중인 로그 파일 번호
CURRENT_ARCHIVE_LOG	BIGINT	current_archive_log - 현재 아카이브 로그 파일 번호
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPDYN\_SQL 관리 뷰 및 SNAP\_GET\_DYN\_SQL\_V95 테이블 함수 - dynsql 논리 그룹 스냅샷 정보 검색

584 페이지의 『SNAPDYN\_SQL 관리 뷰』 및 586 페이지의 『SNAP\_GET\_DYN\_SQL\_V95 테이블 함수』에서는 dynsql 논리 데이터 그룹으로부터 스냅샷 정보를 리턴합니다.

### SNAPDYN\_SQL 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 dynsql 논리 그룹 스냅샷 정보를 검색할 수 있습니다.

이 뷰는 GET SNAPSHOT FOR DYNAMIC SQL ON database-alias CLP 명령과 동일한 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 전체 정보 목록은 588 페이지의 표 147의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPDYN\_SQL 관리 뷰에 대한 SELECT 특권
- SNAPDYN\_SQL 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_DYN\_SQL\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

현재 연결된 데이터베이스의 모든 데이터베이스 파티션에서 실행되는 동적 SQL 목록을 읽은 행 번호 순으로 검색합니다.

```
SELECT PREP_TIME_WORST, NUM_COMPILATIONS, SUBSTR(STMT_TEXT, 1, 60)
       AS STMT_TEXT, DBPARTITIONNUM
FROM SYSIBMADM.SNAPDYN_SQL ORDER BY ROWS_READ
```

이 쿼리의 출력 예는 다음과 같습니다.

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 ...

9개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... 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) ...
```

이 쿼리의 출력(계속).

```

... DBPARTITIONNUM
... -----
...          0
...          0
...          0
...          2
...          1
...          2
...          2
...          1
...          0

```

## SNAP\_GET\_DYN\_SQL\_V95 테이블 함수

SNAP\_GET\_DYN\_SQL\_V95 테이블 함수는 SNAPDYN\_SQL 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

이 테이블 함수는 GET SNAPSHOT FOR DYNAMIC SQL ON database-alias CLP 명령과 동등한 정보를 리턴합니다.

리턴될 수 있는 전체 정보 목록은 588 페이지의 표 147의 내용을 참조하십시오.

## 구문

```

▶▶ SNAP_GET_DYN_SQL_V95 (—dbname—, dbpartitionnum)

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### dbname

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. NULL 또는 빈 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 얻으십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. dbname이 NULL로 설정되어 있지 않고 dbpartitionnum이 NULL로 설정되어 있다면, -1은 내재적으로 dbpartitionnum으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, dbname만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 있는 파일이 존재하지 않으면, SNAP\_GET\_DYN\_SQL\_V95 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_DYN\_SQL\_V95 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 데이터베이스의 현재 연결된 데이터베이스 파티션에서 읽은 행 수 별로 동적 SQL 실행 목록을 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
PREP_TIME_WORST      ...
-----
0 ...
3 ...
...
4 ...
...
4 ...
...
4 ...
...
3 ...
...
4 ...
...
...
```

이 쿼리의 출력(계속).

```
... 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,

```

워크로드를 실행한 후 사용자가 테이블 함수와 함께 다음 쿼리를 사용할 수 있습니다.

```

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

```

이 테이블 함수 기반의 뷰의 경우:

```

SELECT STATS_FABRICATE_TIME, SYNC_RUNSTATS_TIME
FROM SYSIBMADM.SNAPDYN_SQL

```

```

STATS_FABRICATE_TIME  SYNC_RUNSTATS_TIME
-----
                        5          10
                        3          20

```

2개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 177. SNAPDYN\_SQL 관리 뷰 및 SNAP\_GET\_DYN\_SQL\_V95 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
NUM_EXECUTIONS	BIGINT	num_executions - 명령문 실행
NUM_COMPILATIONS	BIGINT	num_compilations - 명령문 컴파일
PREP_TIME_WORST	BIGINT	prep_time_worst - 명령문의 가장 안좋은 준비 시간
PREP_TIME_BEST	BIGINT	prep_time_best - 명령문의 가장 좋은 준비 시간
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 삭제된 내부 행 수
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 삽입된 내부 행 수
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 갱신된 내부 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
STMT_SORTS	BIGINT	stmt_sorts - 명령문 정렬 수
SORT_OVERFLOW	BIGINT	sort_overflows - 정렬 오버플로우 수
TOTAL_SORT_TIME	BIGINT	total_sort_time - 총 정렬 시간

표 177. SNAPDYN\_SQL 관리 뷰 및 SNAP\_GET\_DYN\_SQL\_V95 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
TOTAL_EXEC_TIME	BIGINT	total_exec_time - 경과된 명령문 실행 시간
TOTAL_EXEC_TIME_MS	BIGINT	total_exec_time - 경과된 명령문 실행 시간
TOTAL_USR_CPU_TIME	BIGINT	total_usr_cpu_time - 명령문에 대한 전체 사용자 CPU
TOTAL_USR_CPU_TIME_MS	BIGINT	total_usr_cpu_time - 명령문에 대한 전체 사용자 CPU
TOTAL_SYS_CPU_TIME	BIGINT	total_sys_cpu_time - 명령문에 대한 전체 시스템 CPU
TOTAL_SYS_CPU_TIME_MS	BIGINT	total_sys_cpu_time - 명령문에 대한 전체 시스템 CPU
STMT_TEXT	CLOB(2 M)	stmt_text - SQL문 텍스트
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
STATS_FABRICATE_TIME	BIGINT	동적 명령문의 쿼리 컴파일 중에 테이블 또는 인덱스 없이 필요한 통계를 작성하는데 시스템이 소비한 전체 시간(밀리초)
SYNC_RUNSTATS_TIME	BIGINT	동적 명령문의 쿼리 컴파일 중에 동기 통계 수집 활동에 소비된 전체 시간(밀리초)

---

## SNAPFCM 관리 뷰 및 SNAP\_GET\_FCM 테이블 함수 – fcm 논리 데이터 그룹 스냅샷 정보 검색

SNAPFCM 관리 뷰 및 SNAP\_GET\_FCM 테이블 함수는 데이터베이스 관리 프로그램 스냅샷에서 FCM(Fast Communication Manager), 특히 논리 데이터 그룹에 대한 정보를 리턴합니다.

### SNAPFCM 관리 뷰

SNAPDBM, SNAPDBM\_MEMORY\_POOL, SNAPFCM\_PART 및 SNAPSWITCHES 관리 뷰와 함께 사용되는 경우, SNAPFCM 관리 뷰는 GET SNAPSHOT FOR DBM 명령과 동등한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 592 페이지의 표 148의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPFCM 관리 뷰에 대한 SELECT 특권
- SNAPFCM 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_FCM 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

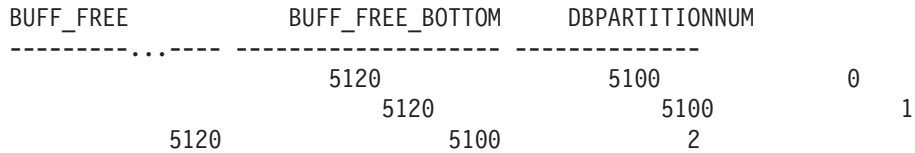
- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### 예 :

모든 데이터베이스 파티션에서 FCM(Fast Communication Manager)의 메시지 버퍼에 대한 정보를 검색합니다.

```
SELECT BUFF_FREE, BUFF_FREE_BOTTOM, DBPARTITIONNUM  
FROM SYSIBMADM.SNAPFCM ORDER BY DBPARTITIONNUM
```

이 쿼리의 출력 예는 다음과 같습니다.



## SNAP\_GET\_FCM 테이블 함수

SNAP\_GET\_FCM 테이블 함수는 SNAPFCM 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_DBM\_V95, SNAP\_GET\_DBM\_MEMORY\_POOL, SNAP\_GET\_FCM\_PART 및 SNAP\_GET\_SWITCHES 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_FCM 테이블 함수는 GET SNAPSHOT FOR DBM 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 592 페이지의 표 148의 내용을 참조하십시오.

## 구문

```

▶▶ SNAP_GET_FCM ( ( dbpartitionnum ) )

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_FCM 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_FCM 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한



또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

예 :

데이터베이스 파티션 1에서 FCM(Fast Communication Manager)의 메시지 버퍼에 대한 정보를 검색합니다.

```
SELECT BUFF_FREE, BUFF_FREE_BOTTOM, DBPARTITIONNUM
FROM TABLE(SYSPROC.SNAP_GET_FCM( 1 )) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

BUFF_FREE	BUFF_FREE_BOTTOM	DBPARTITIONNUM
5120	5100	1

### 리턴되는 정보

표 178. SNAPFCM 관리 뷰 및 SNAP\_GET\_FCM 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
BUFF_FREE	BIGINT	buff_free - 현재 여유 FCM 버퍼
BUFF_FREE_BOTTOM	BIGINT	buff_free_bottom - 최소 여유 FCM 버퍼
CH_FREE	BIGINT	ch_free - 현재 여유 채널
CH_FREE_BOTTOM	BIGINT	ch_free_bottom - 최소 여유 채널
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPFCM\_PART 관리 뷰 및 SNAP\_GET\_FCM\_PART 테이블 함수 - fcm\_node 논리 데이터 그룹 스냅샷 정보 검색

SNAPFCM\_PART 관리 뷰 및 SNAP\_GET\_FCM\_PART 테이블 함수는 데이터베이스 관리 프로그램 스냅샷에서 FCM(Fast Communication Manager), 특히 논리 데이터 그룹에 대한 정보를 리턴합니다.

### SNAPFCM\_PART 관리 뷰

SNAPDBM, SNAPDBM\_MEMORY\_POOL, SNAPFCM 및 SNAPSWITCHES 관리 뷰와 함께 사용되는 경우, SNAPFCM\_PART 관리 뷰는 GET SNAPSHOT FOR DBM 명령과 동등한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 595 페이지의 표 149의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPFCM\_PART 관리 뷰에 대한 SELECT 특권
- SNAPFCM\_PART 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_FCM\_PART 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

FCM(Fast Communication Manager)에 대한 버퍼의 송수신 정보를 검색합니다.

```
SELECT CONNECTION_STATUS, TOTAL_BUFFERS_SENT, TOTAL_BUFFERS_RECEIVED  
FROM SYSIBMADM.SNAPFCM_PART WHERE DBPARTITIONNUM = 0
```

이 쿼리의 출력 예는 다음과 같습니다.

CONNECTION_STATUS	TOTAL_BUFFERS_SENT	TOTAL_BUFFERS_RCVD
INACTIVE	2	1

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_FCM\_PART 테이블 함수

SNAP\_GET\_FCM\_PART 테이블 함수는 SNAPFCM\_PART 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_DBM\_V95, SNAP\_GET\_DBM\_MEMORY\_POOL, SNAP\_GET\_FCM 및 SNAP\_GET\_SWITCHES 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_FCM\_PART 테이블 함수는 GET SNAPSHOT FOR DBM 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 595 페이지의 표 149의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_FCM_PART ( dbpartitionnum ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_FCM\_PART 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_FCM\_PART 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

모든 데이터베이스 파티션에서 FCM(Fast Communication Manager)에 대한 버퍼의 송수신 정보를 검색합니다.

```
SELECT FCM_DBPARTITIONNUM, TOTAL_BUFFERS_SENT, TOTAL_BUFFERS_RCVD,  
       DBPARTITIONNUM FROM TABLE(SNAP_GET_FCM_PART()) AS T  
ORDER BY DBPARTITIONNUM
```

이 쿼리의 출력 예는 다음과 같습니다.

FCM_DBPARTITIONNUM	TOTAL_BUFFERS_SENT	TOTAL_BUFFERS_RCVD	DBPARTITIONNUM	
0	305	305	0	0
1	5647	1664	0	0
2	5661	1688	0	0
0	19	19	1	1
1	305	301	1	1
2	1688	5661	1	1
0	1664	5647	2	2
1	10	10	2	2
2	301	305	2	2

## 리턴되는 정보

표 179. SNAPFCM\_PART 관리 뷰 및 SNAP\_GET\_FCM\_PART 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
CONNECTION_STATUS	VARCHAR(10)	connection_status - 연결 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• INACTIVE</li> <li>• ACTIVE</li> <li>• CONGESTED</li> </ul>
TOTAL_BUFFERS_SENT	BIGINT	total_buffers_sent - 전송된 전체 FCM 버퍼
TOTAL_BUFFERS_RCVD	BIGINT	total_buffers_rcvd - 수신된 전체 FCM 버퍼
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
FCM_DBPARTITIONNUM	SMALLINT	데이터를 전송하거나 데이터가 수신된 데이터베이스 파티션 번호 (TOTAL_BUFFERS_SENT 및 TOTAL_BUFFERS_RCVD 컬럼별도)

## SNAPHADR 관리 뷰 및 SNAP\_GET\_HADR 테이블 함수 - hadr 논리 데이터 그룹 스냅샷 정보 검색

SNAPHADR 관리 뷰 및 SNAP\_GET\_HADR 테이블 함수는 데이터베이스 스냅샷에서 고가용성 재해 복구(HADR), 특히 hadr 논리 데이터 그룹에 대한 정보를 리턴합니다.

### SNAPHADR 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 hadr 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다. 데이터베이스가 1차 또는 대기 고가용성 재해 복구(HADR) 데이터베이스인 경우 해당 뷰에서만 데이터를 리턴합니다.

SNAPDB, SNAPDB\_MEMORY\_POOL, SNAPDETAILOG 및 SNAPSTORAGE\_PATHS 관리 뷰와 함께 사용되는 경우, SNAPHADR 관리 뷰는 GET SNAPSHOT FOR DATABASE ON database-alias CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 598 페이지의 표 150의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPHADR 관리 뷰에 대한 SELECT 특권
- SNAPHADR 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_HADR 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

1차 HADR 데이터베이스에서 HADR에 대한 구성 및 상태 정보를 검색합니다.

```
SELECT SUBSTR(DB_NAME, 1, 8) AS DBNAME, HADR_ROLE, HADR_STATE,
       HADR_SYNCMODE, HADR_CONNECT_STATUS
FROM SYSIBMADM.SNAPHADR
```

이 쿼리의 출력 예는 다음과 같습니다.

DBNAME	HADR_ROLE	HADR_STATE	HADR_SYNCMODE	HADR_CONNECT_STATUS
SAMPLE	PRIMARY	PEER	SYNC	CONNECTED

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_HADR 테이블 함수

SNAP\_GET\_HADR 테이블 함수는 SNAPHADR 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_DB\_V95, SNAP\_GET\_DB\_MEMORY\_POOL, SNAP\_GET\_DETAILLOG\_V91 및 SNAP\_GET\_STORAGE\_PATHS 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_HADR 테이블 함수는 GET SNAPSHOT FOR ALL DATABASES CLP 명령과 동등한 정보를 제공합니다.

리턴될 수 있는 정보의 완전한 목록은 598 페이지의 표 150의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_HADR ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스와 동일한 인터페이스 내의 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우, *dbpartitionnum*에 -1이 내재적으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_HADR 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_HADR 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

모든 데이터베이스에서 HADR에 대한 구성 및 상태 정보를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

DBNAME	HADR_ROLE	HADR_STATE	HADR_SYNCMODE	HADR_CONNECT_STATUS
SAMPLE	PRIMARY	PEER	SYNC	CONNECTED
TESTDB	PRIMARY	DISCONNECTED	NEARSYNC	DISCONNECTED

2개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 180. SNAPHADR 관리 뷰 및 SNAP\_GET\_HADR 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR(128)	db_name - 데이터베이스 이름
HADR_ROLE	VARCHAR(10)	hadr_role - HADR 역할. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• PRIMARY</li> <li>• STANDARD</li> <li>• STANDBY</li> </ul>

표 180. SNAPHADR 관리 뷰 및 SNAP\_GET\_HADR 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
HADR_STATE	VARCHAR(14)	<p>hadr_state - HADR 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다.</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 동기화 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• ASYNC</li> <li>• NEARSYNC</li> <li>• SYNC</li> </ul>
HADR_CONNECT_STATUS	VARCHAR(12)	<p>hadr_connect_status - HADR 연결 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다.</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 연결 시간
HADR_HEARTBEAT	INTEGER	hadr_heartbeat - HADR 하트비트
HADR_LOCAL_HOST	VARCHAR(255)	hadr_local_host - HADR 로컬 호스트
HADR_LOCAL_SERVICE	VARCHAR(40)	hadr_local_service - HADR 로컬 서비스
HADR_REMOTE_HOST	VARCHAR(255)	hadr_remote_host - HADR 리모트 호스트
HADR_REMOTE_SERVICE	VARCHAR(40)	hadr_remote_service - HADR 리모트 서비스
HADR_REMOTE_INSTANCE	VARCHAR(128)	hadr_remote_instance - HADR 리모트 인스턴스
HADR_TIMEOUT	BIGINT	hadr_timeout - HADR 시간종료
HADR_PRIMARY_LOG_FILE	VARCHAR(255)	hadr_primary_log_file - HADR 1차 로그 파일
HADR_PRIMARY_LOG_PAGE	BIGINT	hadr_primary_log_page - HADR 1차 로그 페이지
HADR_PRIMARY_LOG_LSN	BIGINT	hadr_primary_log_lsn - HADR 1차 로그 LSN



표 180. SNAPHADR 관리 뷰 및 SNAP\_GET\_HADR 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
HADR_STANDBY_LOG_FILE	VARCHAR(255)	hadr_standby_log_file - HADR 대기 로그 파일
HADR_STANDBY_LOG_PAGE	BIGINT	hadr_standby_log_page - HADR 대기 로그 페이지
HADR_STANDBY_LOG_LSN	BIGINT	hadr_standby_log_lsn - HADR 대기 로그 LSN
HADR_LOG_GAP	BIGINT	hadr_log_gap - HADR 로그 갭
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPLOCK 관리 뷰 및 SNAP\_GET\_LOCK 테이블 함수 - 잠금 논리 데이터 그룹 스냅샷 정보 검색

SNAPLOCK 관리 뷰 및 SNAP\_GET\_LOCK 테이블 함수는 잠금, 특히 잠금 논리 데이터 그룹에 대한 스냅샷 정보를 리턴합니다.

### SNAPLOCK 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 잠금 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPLOCKWAIT 관리 뷰와 같이 사용하여 SNAPLOCK 관리 뷰는 GET SNAPSHOT FOR LOCKS ON database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 603 페이지의 표 151의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPLOCK 관리 뷰에 대한 SELECT 특권
- SNAPLOCK 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_LOCK 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

예 :

현재 연결된 데이터베이스의 데이터베이스 파티션 0에 대한 잠금 정보를 검색합니다.

```
SELECT AGENT_ID, LOCK_OBJECT_TYPE, LOCK_MODE, LOCK_STATUS
      FROM SYSIBMADM.SNAPLOCK WHERE DBPARTITIONNUM = 0
```

다음은 이 쿼리의 출력 예입니다.

AGENT_ID	LOCK_OBJECT_TYPE	LOCK_MODE	LOCK_STATUS
7	TABLE	IX	GRNT

1개의 레코드가 선택되었습니다.

### SNAP\_GET\_LOCK 테이블 함수

SNAP\_GET\_LOCK 테이블 함수는 SNAPLOCK 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

SNAP\_GET\_LOCKWAIT 테이블 함수와 함께 사용되는 SNAP\_GET\_LOCK 테이블 함수는 GET SNAPSHOT FOR LOCKS ON database-alias CLP 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 603 페이지의 표 151의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_LOCK ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

*dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 널 (NULL) 값 또는 비어 있는 문자열을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형의 파일이 없는 경우 SNAP\_GET\_LOCK 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 작성합니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_LOCK 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### 예 :

현재 연결된 데이터베이스의 현재 데이터베이스 파티션에 대한 잠금 정보를 검색합니다.

```
SELECT AGENT_ID, LOCK_OBJECT_TYPE, LOCK_MODE, LOCK_STATUS
FROM TABLE(SNAP_GET_LOCK(' ', -1)) as T
```

다음은 이 쿼리의 출력 예입니다.

AGENT_ID	LOCK_OBJECT_TYPE	LOCK_MODE	LOCK_STATUS
680	INTERNALV_LOCK	S	GRNT
680	INTERNALP_LOCK	S	GRNT

2개의 레코드가 선택되었습니다.

## 리턴된 정보

표 181. SNAPLOCK 관리 뷰 및 SNAP\_GET\_LOCK 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
TAB_FILE_ID	BIGINT	table_file_id - 테이블 파일 ID
LOCK_OBJECT_TYPE	VARCHAR(18)	lock_object_type - 대기하는 잠금 오브젝트 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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>

표 181. SNAPLOCK 관리 뷰 및 SNAP\_GET\_LOCK 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_MODE	VARCHAR(10)	lock_mode - 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (잠금이 없는 경우)</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 - 잠금 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• CONV</li> <li>• GRNT</li> </ul>
LOCK_ESCALATION	SMALLINT	lock_escalation - 잠금 에스컬레이션
TABNAME	VARCHAR(128)	table_name - 테이블 이름
TABSCHEMA	VARCHAR(128)	table_schema - 테이블 스키마 이름
TBSP_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름

표 181. SNAPLOCK 관리 뷰 및 SNAP\_GET\_LOCK 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_ATTRIBUTES	VARCHAR (128)	lock_attributes - 잠금 속성. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴합니다. 잠금이 없을 경우 텍스트 ID는 NONE이며, 그렇지 않을 경우 다음의 조합으로 '+' 부호에 의해 분리됩니다. <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 - 잠금 계수
LOCK_CURRENT_MODE	VARCHAR(10)	lock_current_mode - 변환 이전의 원래 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (잠금이 없는 경우)</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 - 잠금 보유 계수
LOCK_NAME	VARCHAR(32)	lock_name - 잠금 이름
LOCK_RELEASE_FLAGS	BIGINT	lock_release_flags - 잠금 릴리스 플래그
DATA_PARTITION_ID	INTEGER	data_partition_id - 데이터 파티션 ID. 파티션되지 않은 테이블의 경우 이 요소는 NULL입니다.
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

---

## SNAPLOCKWAIT 관리 뷰 및 SNAP\_GET\_LOCKWAIT 테이블 함수 - 잠금 대기 논리 데이터 그룹 스냅샷 정보 검색

SNAPLOCKWAIT 관리 뷰 및 SNAP\_GET\_LOCKWAIT 테이블 함수는 잠금 대기 특히 잠금 대기 논리 데이터 그룹에 대한 스냅샷 정보를 리턴합니다.

### SNAPLOCKWAIT 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 잠금 대기 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPLOCK 관리 뷰와 같이 사용하여 SNAPLOCKWAIT 관리 뷰는 GET SNAPSHOT FOR LOCKS ON database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 609 페이지의 표 152의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPLOCKWAIT 관리 뷰에 대한 SELECT 특권
- SNAPLOCKWAIT 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_LOCKWAIT 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

### 예 :

현재 연결된 데이터베이스에 대한 데이터베이스 파티션 0에서 잠금 대기 정보를 검색합니다.

```
SELECT AGENT_ID, LOCK_MODE, LOCK_OBJECT_TYPE, AGENT_ID_HOLDING_LK,  
       LOCK_MODE_REQUESTED FROM SYSIBMADM.SNAPLOCKWAIT  
WHERE DBPARTITIONNUM = 0
```

다음은 이 쿼리의 출력 예입니다.

```
AGENT_ID      LOCK_MODE LOCK_OBJECT_TYPE ...
-----
          7 IX          TABLE          ...
```

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... AGENT_ID_HOLDING_LK  LOCK_MODE_REQUESTED
... -----
...                    12 IS
```

## SNAP\_GET\_LOCKWAIT 테이블 함수

SNAP\_GET\_LOCKWAIT 테이블 함수는 SNAPLOCKWAIT 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

SNAP\_GET\_LOCK 테이블 함수와 함께 사용되는 SNAP\_GET\_LOCKWAIT 테이블 함수는 GET SNAPSHOT FOR LOCKS ON database-alias CLP 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 609 페이지의 표 152의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_LOCKWAIT (—dbname— [ , dbpartitionnum ] ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 널 (NULL) 값 또는 비어 있는 문자열을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로



설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형의 파일이 없는 경우 SNAP\_GET\_LOCKWAIT 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 작성합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_LOCKWAIT 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

현재 연결된 데이터베이스에 대한 현재 데이터베이스 파티션에서 잠금 대기 정보를 검색합니다.

```
SELECT AGENT_ID, LOCK_MODE, LOCK_OBJECT_TYPE, AGENT_ID_HOLDING_LK,
       LOCK_MODE_REQUESTED FROM TABLE(SNAP_GET_LOCKWAIT('',-1)) AS T
```

다음은 이 쿼리의 출력 예입니다.

```
AGENT_ID      LOCK_MODE  LOCK_OBJECT_TYPE  ...
-----
          12 X          ROW_LOCK          ...
```

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... AGENT_ID_HOLDING_LK  LOCK_MODE_REQUESTED
... -----
...                      7 X
```

## 사용법 참고

잠금 대기 정보를 확인하려면 우선 데이터베이스 관리 프로그램 구성에서 디폴트 LOCK 모니터 스위치를 켜야 합니다. 변경사항을 바로 적용하려면 CLP를 사용하여 인스턴스에 명시적으로 첨부한 다음 CLP 명령을 실행하십시오.

```
UPDATE DATABASE MANAGER CONFIGURATION CLP USING DFT_MON_LOCK ON
```

디폴트 설정은 ADMIN\_CMD 스토어드 프로시저를 통해서도 켤 수 있습니다. 예를 들어, 다음과 같습니다.

```
CALL SYSPROC.ADMIN_CMD('update dbm cfg using DFT_MON_LOCK ON')
```

ADMIN\_CMD 스토어드 프로시저를 사용하거나 이전에 인스턴스에 접속하지 않은 상태로 clp 명령을 실행하는 경우 변경사항을 적용하기 전에 인스턴스를 재사용해야 합니다.

## 리턴된 정보

표 182. SNAPLOCKWAIT 관리 뷰 및 SNAP\_GET\_LOCKWAIT 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
SUBSECTION_NUMBER	BIGINT	ss_number - 서브섹션 번호
LOCK_MODE	VARCHAR(10)	lock_mode - 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (잠금이 없는 경우)</li> <li>• NS</li> <li>• NW</li> <li>• S</li> <li>• SIX</li> <li>• U</li> <li>• X</li> <li>• Z</li> </ul>

표 182. SNAPLOCKWAIT 관리 뷰 및 SNAP\_GET\_LOCKWAIT 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_OBJECT_TYPE	VARCHAR(18)	lock_object_type - 대기하는 잠금 오브젝트 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 잠금을 보유한 에이전트 ID
LOCK_WAIT_START_TIME	TIMESTAMP	lock_wait_start_time - 잠금 대기 시작 시간소인

표 182. SNAPLOCKWAIT 관리 뷰 및 SNAP\_GET\_LOCKWAIT 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_MODE_REQUESTED	VARCHAR(10)	lock_mode_requested - 요청된 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (잠금이 없는 경우)</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 - 잠금 에스컬레이션
TABNAME	VARCHAR(128)	table_name - 테이블 이름
TABSCHEMA	VARCHAR(128)	table_schema - 테이블 스키마 이름
TBSP_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름
APPL_ID_HOLDING_LK	VARCHAR(128)	appl_id_holding_lk - 잠금을 보유한 응용프로그램 ID
LOCK_ATTRIBUTES	VARCHAR(128)	lock_attributes - 잠금 속성. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴합니다. 잠금이 없을 경우 텍스트 ID는 NONE이며, 그렇지 않을 경우 다음의 조합으로 '+' 부호에 의해 분리됩니다. <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>

표 182. SNAPLOCKWAIT 관리 뷰 및 SNAP\_GET\_LOCKWAIT 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_CURRENT_MODE	VARCHAR(10)	lock_current_mode - 변환 이전의 원래 잠금 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• IN</li> <li>• IS</li> <li>• IX</li> <li>• NON (잠금이 없는 경우)</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 - 잠금 이름
LOCK_RELEASE_FLAGS	BIGINT	lock_release_flags - 잠금 릴리스 플래그.
DATA_PARTITION_ID	INTEGER	data_partition_id - 데이터 파티션 ID. 파티션되지 않은 테이블의 경우 이 요소는 NULL입니다.
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPSTMT 관리 뷰 및 SNAP\_GET\_STMT 테이블 함수 - 명령문 스냅샷 정보 검색

SNAPSTMT 관리 뷰 및 SNAP\_GET\_STMT 테이블 함수는 응용프로그램 스냅샷에서 SQL 또는 XQuery문에 대한 정보를 리턴합니다.

### SNAPSTMT 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 명령문 스냅샷 정보를 검색할 수 있습니다.

SNAPAGENT, SNAPAGENT\_MEMORY\_POOL, SNAPAPPL, SNAPAPPL\_INFO 및 SNAPSUBSECTION 관리 뷰와 함께 사용되는 경우, SNAPSTMT 관리 뷰는 GET SNAPSHOT FOR APPLICATIONS on database-alias CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 615 페이지의 표 153의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPSTMT 관리 뷰에 대한 SELECT 특권
- SNAPSTMT 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_STMT 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

현재 연결된 단일 파티션 데이터베이스에서 실행되는 명령문에 대해 읽은 행, 쓴 행 및 수행된 조작을 검색합니다.

```
SELECT SUBSTR(STMT_TEXT,1,30) AS STMT_TEXT, ROWS_READ, ROWS_WRITTEN,  
       STMT_OPERATION FROM SYSIBMADM.SNAPSTMT
```

다음은 이 쿼리의 출력 예입니다.

STMT_TEXT	ROWS_READ	ROWS_WRITTEN	STMT_OPERATION
-----	-----	-----	-----
-	0	0	0 FETCH
-	0	0	0 STATIC_COMMIT

2개의 레코드가 선택되었습니다.

## SNAP\_GET\_STMT 테이블 함수

SNAP\_GET\_STMT 테이블 함수는 SNAPSTMT 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_AGENT, SNAP\_GET\_AGENT\_MEMORY\_POOL, SNAP\_GET\_APPL\_V95, SNAP\_GET\_APPL\_INFO\_V95 및 SNAP\_GET\_SUBSECTION 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_STMT

테이블 함수는 GET SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

리턴될 수 있는 정보의 완전한 목록은 615 페이지의 표 153의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_STMT (—dbname— [ , dbpartitionnum ] )
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_STMT 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_STMT 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

예 :

현재 연결된 데이터베이스의 현재 데이터베이스 파티션에서 실행되는 명령문에 대해 읽은 행, 쓴 행 및 수행된 조작을 검색합니다.

```
SELECT SUBSTR(STMT_TEXT,1,30) AS STMT_TEXT, ROWS_READ,
       ROWS_WRITTEN, STMT_OPERATION FROM TABLE(SNAP_GET_STMT('-',-1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...
	...	...

5개의 레코드가 선택되었습니다.

...

이 쿼리의 출력(계속).

...	ROWS_WRITTEN	STMT_OPERATION
...	-----	-----
...	0	EXECUTE_IMMEDIATE
...	0	FETCH
...	0	NONE
...	0	NONE
...	1	EXECUTE_IMMEDIATE
...		

## 리턴되는 정보

표 183. SNAPSTMT 관리 뷰 및 SNAP\_GET\_STMT 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
NUM_AGENTS	BIGINT	num_agents - 명령문에서 작업하는 에이전트 수
AGENTS_TOP	BIGINT	agents_top - 작성된 에이전트 수



표 183. SNAPSTMT 관리 뷰 및 SNAP\_GET\_STMT 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
STMT_TYPE	VARCHAR(20)	stmt_type - 명령문 유형. 이 인터페이스는 sqlmon.h의 정의를 기본으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• DYNAMIC</li> <li>• NON_STMT</li> <li>• STATIC</li> <li>• STMT_TYPE_UNKNOWN</li> </ul>
STMT_OPERATION	VARCHAR(20)	stmt_operation/operation - 명령문 조작. 이 인터페이스는 sqlmon.h의 정의를 기본으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 섹션 번호

표 183. SNAPSTMT 관리 뷰 및 SNAP\_GET\_STMT 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
QUERY_COST_ESTIMATE	BIGINT	query_cost_estimate - 추정 쿼리 비용
QUERY_CARD_ESTIMATE	BIGINT	query_card_estimate - 추정 쿼리 행 수
DEGREE_PARALLELISM	BIGINT	degree_parallelism - 병렬 처리 수준
STMT_SORTS	BIGINT	stmt_sorts - 명령문 정렬 수
TOTAL_SORT_TIME	BIGINT	total_sort_time - 총 정렬 시간
SORT_OVERFLOWS	BIGINT	sort_overflows - 정렬 오버플로우 수
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 삭제된 내부 행 수
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 갱신된 내부 행 수
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 삽입된 내부 행 수
FETCH_COUNT	BIGINT	fetch_count - 성공한 페치 수
STMT_START	TIMESTAMP	stmt_start - 명령문 조작 시작 시간소인
STMT_STOP	TIMESTAMP	stmt_stop - 명령문 조작 중지 시간소인
STMT_USR_CPU_TIME_S	BIGINT	stmt_usr_cpu_time - 명령문이 사용한 사용자 CPU 시간
STMT_USR_CPU_TIME_MS	BIGINT	stmt_usr_cpu_time - 명령문이 사용한 사용자 CPU 시간
STMT_SYS_CPU_TIME_S	BIGINT	stmt_sys_cpu_time - 명령문이 사용한 시스템 CPU 시간
STMT_SYS_CPU_TIME_MS	BIGINT	stmt_sys_cpu_time - 명령문이 사용한 시스템 CPU 시간
STMT_ELAPSED_TIME_S	BIGINT	stmt_elapsed_time - 가장 최근의 명령문 경과 시간
STMT_ELAPSED_TIME_MS	BIGINT	stmt_elapsed_time - 가장 최근의 명령문 경과 시간
BLOCKING_CURSOR	SMALLINT	blocking_cursor - 블로킹 커서
STMT_NODE_NUMBER	SMALLINT	stmt_node_number - 명령문 노드
CURSOR_NAME	VARCHAR (128)	cursor_name - 커서 이름
CREATOR	VARCHAR (128)	creator - 응용프로그램 작성자
PACKAGE_NAME	VARCHAR (128)	package_name - 패키지 이름
STMT_TEXT	CLOB(16 M)	stmt_text - SQL문 텍스트
CONSISTENCY_TOKEN	VARCHAR (128)	consistency_token - 패키지 일관성 토큰
PACKAGE_VERSION_ID	VARCHAR (128)	package_version_id - 패키지 버전
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수

표 183. SNAPSTMT 관리 뷰 및 SNAP\_GET\_STMT 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수 모니터 요소
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수 모니터 요소
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPSTORAGE\_PATHS 관리 뷰 및 SNAP\_GET\_STORAGE\_PATHS 테이블 함수 - 자동 스토리지 경로 정보 검색

주: 이 테이블 함수는 사용되지 않으며 618 페이지의 『SNAPSTORAGE\_PATHS 관리 뷰 및 SNAP\_GET\_STORAGE\_PATHS\_V97 테이블 함수 - 자동 스토리지 경로 정보 검색』으로 교체됩니다.

SNAPSTORAGE\_PATHS 관리 뷰 및 SNAP\_GET\_STORAGE\_PATHS 테이블 함수는 각 스토리지 경로에 대한 파일 시스템 정보를 포함하여 데이터베이스에 대한 자동 스토리지 경로 목록을, 특히 db\_storage\_group 논리 데이터 그룹에서 리턴합니다.

### SNAPSTORAGE\_PATHS 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 자동 스토리지 경로 정보를 검색할 수 있습니다.

SNAPDB, SNAPDETAILLOG, SNAPHADR 및 SNAPDB\_MEMORY\_POOL 관리 뷰와 같이 사용하여 SNAPSTORAGE\_PATHS 관리 뷰는 GET SNAPSHOT FOR DATABASE ON database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 770 페이지의 표 184의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPSTORAGE\_PATHS 관리 뷰에 대한 SELECT 특권
- SNAPSTORAGE\_PATHS 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_STORAGE\_PATHS 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 한 개 파티션의 데이터베이스에 대한 스토리지 경로를 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

```
DB_NAME  DB_STORAGE_PATH  HOSTNAME
-----  -
STOPATH  d:                   JESSICAE
```

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_STORAGE\_PATHS 테이블 함수

SNAP\_GET\_STORAGE\_PATHS 테이블 함수는 SNAPSTORAGE\_PATHS 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_DB\_V95, SNAP\_GET\_DETAILLOG\_V91, SNAP\_GET\_HADR 및 SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수와 함께 사용되는 경우,

SNAP\_GET\_STORAGE\_PATHS 테이블 함수는 GET SNAPSHOT FOR ALL DATABASES CLP 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 770 페이지의 표 184의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_STORAGE_PATHS (—dbname— [ , dbpartitionnum ] ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우 -1이 *dbpartitionnum*에 대해 내재적으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_STORAGE\_PATHS 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_STORAGE\_PATHS 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

예:

모든 활성 데이터베이스에 대한 스토리지 경로 정보를 검색합니다.

```
SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME, DB_STORAGE_PATH
FROM TABLE(SNAP_GET_STORAGE_PATHS(CAST (NULL AS VARCHAR(128)), -1)) AS T
```

다음은 이 쿼리의 출력 예입니다.

```
DB_NAME  DB_STORAGE_PATH
-----  -
STOPATH  /home/jessicae/sdb
MYDB     /home/jessicae/mdb
```

2개의 레코드가 선택되었습니다.

## 리턴된 정보

파일 시스템 정보가 리턴되도록 하려면 BUFFERPOOL 모니터 스위치를 켜야 합니다.

표 184. SNAPSTORAGE\_PATHS 관리 뷰 및 SNAP\_GET\_STORAGE\_PATHS 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
DB_STORAGE_PATH	VARCHAR (256)	db_storage_path - 자동 스토리지 경로
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
FS_ID	VARCHAR(22)	fs_id - 고유 파일 시스템 ID 번호
FS_TOTAL_SIZE	BIGINT	fs_total_size - 파일 시스템의 전체 크기
FS_USED_SIZE	BIGINT	fs_used_size - 파일 시스템에서 사용된 스페이스 용량
STO_PATH_FREE_SIZE	BIGINT	sto_path_free_sz - 자동 스토리지 경로 여유 공간

## SNAPSUBSECTION 관리 뷰 및 SNAP\_GET\_SUBSECTION 테이블 함수 - 서브섹션 논리 모니터 그룹 스냅샷 정보 검색

SNAPSUBSECTION 관리 뷰 및 SNAP\_GET\_SUBSECTION 테이블 함수는 응용프로그램 서브섹션, 즉 서브섹션 논리 모니터 그룹화에 대한 정보를 리턴합니다.

## SNAPSUBSECTION 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 서브섹션 논리 모니터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPAGENT, SNAPAGENT\_MEMORY\_POOL, SNAPAPPL, SNAPAPPL\_INFO 및 SNAPSTMT 관리 뷰와 함께 사용되는 경우, SNAPSUBSECTION 관리 뷰는 GET SNAPSHOT FOR APPLICATIONS on database-alias CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 624 페이지의 표 155의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPSUBSECTION 관리 뷰에 대한 SELECT 특권
- SNAPSUBSECTION 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_SUBSECTION 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

### 예 :

모든 데이터베이스 파티션에서 실행 중인 서브섹션의 상태를 가져옵니다.

```
SELECT DB_NAME, STMT_TEXT, SS_STATUS, DBPARTITIONNUM
      FROM SYSIBMADM.SNAPSUBSECTION
      ORDER BY DB_NAME, SS_STATUS, DBPARTITIONNUM
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	STMT_TEXT	SS_STATUS	DBPARTITIONNUM
SAMPLE	select * from EMPLOYEE	EXEC	0
SAMPLE	select * from EMPLOYEE	EXEC	1

## SNAP\_GET\_SUBSECTION 테이블 함수

SNAP\_GET\_SUBSECTION 테이블 함수는 SNAPSUBSECTION 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

리턴 가능한 전체 정보 목록은 624 페이지의 표 155의 내용을 참조하십시오.

SNAP\_GET\_AGENT, SNAP\_GET\_AGENT\_MEMORY\_POOL, SNAP\_GET\_APPL\_V95, SNAP\_GET\_APPL\_INFO\_V95 및

SNAP\_GET\_STMT 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_SUBSECTION 테이블 함수는 GET SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

### 구문

```
▶▶ SNAP_GET_SUBSECTION ( ( dbname ) [ , dbpartitionnum ] ) ▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스와 동일한 인터페이스 내의 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않



을 경우, SNAP\_GET\_SUBSECTION 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_SUBSECTION 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

모든 데이터베이스 파티션에서 실행 중인 서브섹션의 상태를 가져옵니다.

```
SELECT DB_NAME, STMT_TEXT, SS_STATUS, DBPARTITIONNUM
      FROM TABLE(SYSPROC.SNAP_GET_SUBSECTION( '', 0 )) as T
      ORDER BY DB_NAME, SS_STATUS, DBPARTITIONNUM
```

이 쿼리의 출력 예는 다음과 같습니다.

DB_NAME	STMT_TEXT	SS_STATUS	DBPARTITIONNUM
SAMPLE	select * from EMPLOYEE	EXEC	0
SAMPLE	select * from EMPLOYEE	EXEC	1

## 리턴되는 정보

표 185. SNAPSUBSECTION 관리 뷰 및 SNAP\_GET\_SUBSECTION 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
STMT_TEXT	CLOB(16 M)	stmt_text - SQL문 텍스트
SS_EXEC_TIME	BIGINT	ss_exec_time - 서브섹션 실행 경과 시간
TQ_TOT_SEND_SPILLS	BIGINT	tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총계
TQ_CUR_SEND_SPILLS	BIGINT	tq_cur_send_spills - 오버플로우된 현재 테이블 큐 버퍼 수
TQ_MAX_SEND_SPILLS	BIGINT	tq_max_send_spills - 최대 테이블 큐 버퍼 오버플로우 수
TQ_ROWS_READ	BIGINT	tq_rows_read - 테이블 큐에서 읽은 행 수

표 185. SNAPSUBSECTION 관리 뷰 및 SNAP\_GET\_SUBSECTION 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TQ_ROWS_WRITTEN	BIGINT	tq_rows_written - 테이블 큐에 기록된 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
SS_USR_CPU_TIME_S	BIGINT	ss_usr_cpu_time - 서버섹션이 사용한 사용자 CPU 시간
SS_USR_CPU_TIME_MS	BIGINT	ss_usr_cpu_time - 서버섹션이 사용한 사용자 CPU 시간
SS_SYS_CPU_TIME_S	BIGINT	ss_sys_cpu_time - 서버섹션이 사용한 시스템 CPU 시간
SS_SYS_CPU_TIME_MS	BIGINT	ss_sys_cpu_time - 서버섹션이 사용한 시스템 CPU 시간
SS_NUMBER	INTEGER	ss_number - 서버섹션 번호
SS_STATUS	VARCHAR(20)	ss_status - 서버섹션 상태. 이 인터페이스는 sqlmon.h의 정의를 기본으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 서버섹션 노드 번호
TQ_NODE_WAITED_FOR	SMALLINT	tq_node_waited_for - 테이블 큐의 노드를 기다림
TQ_WAIT_FOR_ANY	INTEGER	tq_wait_for_any - 테이블 큐에 전송할 노드를 기다리는 중
TQ_ID_WAITING_ON	INTEGER	tq_id_waiting_on - 테이블 큐의 노드에서 대기
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPSWITCHES 관리 뷰 및 SNAP\_GET\_SWITCHES 테이블 함수 - 데이터베이스 스냅샷 스위치 상태 정보 검색

SNAPSWITCHES 관리 뷰 및 SNAP\_GET\_SWITCHES 테이블 함수는 데이터베이스 스냅샷 전환 상태에 대한 정보를 리턴합니다.

### SNAPSWITCHES 관리 뷰

이 뷰에서는 GET DBM MONITOR SWITCHES CLP 명령과 동등한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 628 페이지의 표 156의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPSWITCHES 관리 뷰에 대한 SELECT 특권
- SNAPSWITCHES 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_SWITCHES 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

모든 데이터베이스 파티션에 대한 DBM 모니터 전환 상태 정보를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
UOW_SW_STATE STATEMENT_SW_STATE TABLE_SW_STATE BUFFPOOL_SW_STATE ...  
-----  
                0                0                0                0 ...  
                0                0                0                0 ...  
                0                0                0                0 ...  
                ...
```

3개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... 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 테이블 함수

SNAP\_GET\_SWITCHES 테이블 함수는 SNAPSWITCHES 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

이 테이블 함수는 GET DBM MONITOR SWITCHES CLP 명령과 동등한 데이터를 제공합니다.

리턴될 수 있는 정보의 완전한 목록은 628 페이지의 표 156의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_SWITCHES ( ( dbpartitionnum ) )
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_SWITCHES 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_SWITCHES 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예:

현재 데이터베이스 파티션에 대한 DBM 모니터 전환 상태 정보를 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
UOW_SW_STATE STATEMENT_SW_STATE TABLE_SW_STATE...
-----
1 1 1...
```

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... BUFFPOOL_SW_STATE LOCK_SW_STATE SORT_SW_STATE TIMESTAMP_SW_STATE
... -----
... 1 1 0 1
```

### 리턴되는 정보

표 186. SNAPSWITCHES 관리 뷰 및 SNAP\_GET\_SWITCHES 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
UOW_SW_STATE	SMALLINT	작업 단위(UOW) 모니터 기록 전환 상태 (0 또는 1).
UOW_SW_TIME	TIMESTAMP	작업 단위(UOW) 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간.
STATEMENT_SW_STATE	SMALLINT	SQL문 모니터 기록 전환 상태(0 또는 1).
STATEMENT_SW_TIME	TIMESTAMP	SQL문 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간.
TABLE_SW_STATE	SMALLINT	테이블 활동 모니터 기록 전환 상태(0 또는 1).
TABLE_SW_TIME	TIMESTAMP	테이블 활동 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간
BUFFPOOL_SW_STATE	SMALLINT	버퍼 풀 활동 모니터 기록 전환 상태(0 또는 1)
BUFFPOOL_SW_TIME	TIMESTAMP	버퍼 풀 활동 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간
LOCK_SW_STATE	SMALLINT	잠금 모니터 기록 전환 상태(0 또는 1)
LOCK_SW_TIME	TIMESTAMP	잠금 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간
SORT_SW_STATE	SMALLINT	정렬 모니터 기록 전환 상태(0 또는 1)

표 186. SNAPSWITCHES 관리 뷰 및 SNAP\_GET\_SWITCHES 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
SORT_SW_TIME	TIMESTAMP	정렬 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간
TIMESTAMP_SW_STATE	SMALLINT	시간소인 모니터 기록 전환 상태(0 또는 1)
TIMESTAMP_SW_TIME	TIMESTAMP	시간소인 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPTAB 관리 뷰 및 SNAP\_GET\_TAB\_V91 테이블 함수 - 테이블 논리 데이터 그룹 스냅샷 정보 검색

SNAPTAB 관리 뷰 및 SNAP\_GET\_TAB\_V91 테이블 함수는 테이블 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.

### SNAPTAB 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 테이블 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPTAB\_REORG 관리 뷰와 함께 사용될 경우, SNAPTAB 관리 뷰는 GET SNAPSHOT FOR TABLES ON database-alias CLP 명령과 동일한 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 631 페이지의 표 157의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPTAB 관리 뷰에 대한 SELECT 특권
- SNAPTAB 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_TAB\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON

- SYSCTRL
- SYSMOINT
- SYSADM

예 :

모든 활성 테이블에 대한 스키마 및 이름을 검색합니다.

```
SELECT SUBSTR(TABSHEMA,1,8), SUBSTR(TABNAME,1,15) AS TABNAME, TAB_TYPE,
       DBPARTITIONNUM FROM SYSIBMADM.SNAPTAB
```

다음은 이 쿼리의 출력 예입니다.

TABSHEMA	TABNAME	TAB_TYPE	DBPARTITIONNUM
SYSTOOLS	HMON_ATM_INFO	USER_TABLE	0

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_TAB\_V91 테이블 함수

SNAP\_GET\_TAB\_V91 테이블 함수는 SNAPTAB 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_TAB\_REORG 테이블 함수와 함께 사용될 경우, SNAP\_GET\_TAB\_V91 테이블 함수는 GET SNAPSHOT FOR TABLES ON database-alias CLP 명령과 동일한 정보를 리턴합니다.

리턴 가능한 전체 정보 목록은 631 페이지의 표 157의 내용을 참조하십시오.

## 구문

```
▶▶—SNAP_GET_TAB_V91—(—dbname—, dbpartitionnum—)
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. NULL 또는 빈 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 얻으십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_TAB\_V91 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_TAB\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

현재 연결된 데이터베이스에 대한 집계 보기로 활성 테이블 목록을 검색합니다.

```
SELECT SUBSTR(TABSCHEMA,1,8) AS TABSCHEMA, SUBSTR(TABNAME,1,15) AS TABNAME,  
       TAB_TYPE, DBPARTITIONNUM FROM TABLE(SNAP_GET_TAB('','-2)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

TABSCHEMA	TABNAME	TAB_TYPE	DBPARTITIONNUM
SYSTOOLS	HMON_ATM_INFO	USER_TABLE	-
JESSICAE	EMPLOYEE	USER_TABLE	-



## 리턴되는 정보

표 187. SNAPTAB 관리 뷰 및 SNAP\_GET\_TAB\_V91 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TABSCHEMA	VARCHAR (128)	table_schema - 테이블 스키마 이름
TABNAME	VARCHAR (128)	table_name - 테이블 이름
TAB_FILE_ID	BIGINT	table_file_id - 테이블 파일 ID
TAB_TYPE	VARCHAR(14)	table_type - 테이블 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 데이터 오브젝트 페이지 수
INDEX_OBJECT_PAGES	BIGINT	index_object_pages - 인덱스 오브젝트 페이지 수
LOB_OBJECT_PAGES	BIGINT	lob_object_pages - LOB 오브젝트 페이지 수
LONG_OBJECT_PAGES	BIGINT	long_object_pages - Long 오브젝트 페이지 수
XDA_OBJECT_PAGES	BIGINT	xda_object_pages - XDA 오브젝트 페이지 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
OVERFLOW_ACCESSES	BIGINT	overflow_accesses - 오버플로우된 레코드에 액세스
PAGE_REORGS	BIGINT	page_reorgs - 페이지 재구성
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
DATA_PARTITION_ID	INTEGER	data_partition_id - 데이터 파티션 ID. 파티션되지 않은 테이블의 경우 이 요소는 널(NULL)이 됩니다.

---

## SNAPTAB\_REORG 관리 뷰 및 SNAP\_GET\_TAB\_REORG 테이블 함수 - 테이블 재구성 스냅샷 정보 검색

SNAPTAB\_REORG 관리 뷰 및 SNAP\_GET\_TAB\_REORG 테이블 함수는 테이블 재구성 정보를 리턴합니다. 재구성된 테이블이 없다면 0행이 리턴됩니다.

### SNAPTAB\_REORG 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 테이블 재구성 스냅샷 정보를 검색할 수 있습니다.

SNAPTAB 관리 뷰를 사용한 SNAPTAB\_REORG 관리 뷰는 GET SNAPSHOT FOR TABLES ON database-alias CLP 명령과 동일한 데이터를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 전체 정보 목록은 636 페이지의 표 158의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPTAB\_REORG 관리 뷰에 대한 SELECT 특권
- SNAPTAB\_REORG 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_TAB\_REORG 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

### 예 :

현재 연결된 데이터베이스에서 모든 데이터베이스 파티션에 대한 재구성 조작에 관한 세 부사항을 선택합니다.

```
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 SYSIBMADM.SNAPTAB_REORG ORDER BY DBPARTITIONNUM
```

이 쿼리의 출력 예는 다음과 같습니다.

TAB_NAME	TAB_SCHEMA	REORG_PHASE	...
EMPLOYEE	DBUSER	REPLACE	...
EMPLOYEE	DBUSER	REPLACE	...
EMPLOYEE	DBUSER	REPLACE	...

3개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	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

재구성 조작에 관한 모든 정보를 선택하여 SNAPTAB\_REORG 관리 뷰의 다차원적으로 클러스터된(MDC) 테이블에서부터 Extent를 재개합니다.

```
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 테이블 함수

SNAP\_GET\_TAB\_REORG 테이블 함수는 SNAPTAB\_REORG 관리 뷰와 동일한 정보를 리턴하지만 특정 데이터베이스 파티션에 대한 특정 데이터베이스의 정보를 검색할 수 있으며, 모든 데이터베이스 파티션을 집계할 수 있습니다.

SNAP\_GET\_TAB 테이블 함수를 사용하여 SNAP\_GET\_TAB\_REORG 테이블 함수는 GET SNAPSHOT FOR TABLES ON database-alias CLP 명령과 동일한 데이터를 제공합니다.

리턴될 수 있는 전체 정보 목록은 636 페이지의 표 158의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_TAB_REORG ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

*dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지

정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴 시 "Indirect" 또는 "Home"의 디렉토리 항목 유형을 갖는 데이터베이스 이름을 지정합니다. NULL 또는 빈 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 얻으십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결에 사용할 수 있으며 응용프로그램에서 사용할 수 있는 파티션입니다.

두 *dbname* 및 *dbpartitionnum*이 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저에서 작성된 파일로부터 데이터를 읽으려고 합니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형을 포함한 파일이 존재하지 않으면 SNAP\_GET\_TAB\_REORG 테이블 함수는 현재 연결된 데이터베이스와 데이터베이스 파티션 번호로 스냅샷을 작성합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_TAB\_REORG 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

현재 연결된 데이터베이스에서 데이터베이스 파티션 1에 대한 재구성 조작에 관한 세부 사항을 선택합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
TAB_NAME      TAB_SCHEMA    REORG_PHASE    REORG_TYPE      ...
-----
EMPLOYEE      DBUSER        REPLACE        RECLAIM+OFFLINE+ALLO ...
...
1개의 레코드가 선택되었습니다.
```

이 쿼리의 출력(계속).

```
... REORG_STATUS REORG_COMPLETION DBPARTITIONNUM
... -----
... COMPLETED SUCCESS 1
...

```

재구성 조작에 관한 모든 정보를 선택하여 `SNAP_GET_TAB_REORG` 테이블 함수를 사용하여 다차원적으로 클러스터된(MDC) 테이블에서 Extent를 재개합니다.

```
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
```

## 리턴된 정보

표 188. `SNAPTAB_REORG` 관리 뷰 및 `SNAP_GET_TAB_REORG` 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TABNAME	VARCHAR (128)	table_name - 테이블 이름
TABSCHEMA	VARCHAR (128)	table_schema - 테이블 스키마 이름
PAGE_REORGS	BIGINT	page_reorgs - 페이지 재구성
REORG_PHASE	VARCHAR (16)	reorg_phase - 테이블 재구성 단계. 이 인터페이스는 <code>sqlmon.h</code> 의 정의를 기본으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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> 또는 SORT+DICT_SAMPLE.
REORG_MAX_PHASE	INTEGER	reorg_max_phase - 최대 테이블 재구성 단계

표 188. SNAPTAB\_REORG 관리 뷰 및 SNAP\_GET\_TAB\_REORG 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
REORG_CURRENT_COUNTER	BIGINT	reorg_current_counter - 테이블 재구성 진행
REORG_MAX_COUNTER	BIGINT	reorg_max_counter - 전체 테이블 재구성 크기

표 188. SNAPTAB\_REORG 관리 뷰 및 SNAP\_GET\_TAB\_REORG 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
REORG_TYPE	VARCHAR (128)	<p>reorg_type - 테이블 재구성 속성. 이 인터페이스는 '+'로 구분되는 다음 ID의 조합을 사용하여 텍스트 ID를 리턴합니다.</p> <p>또는 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• RECLAIM</li> <li>• RECLUSTER</li> <li>• RECLAIM_EXTS</li> </ul> <p>또는 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• +OFFLINE</li> <li>• +ONLINE</li> </ul> <p>액세스 모드 지정 시 다음 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• +ALLOW_NONE</li> <li>• +ALLOW_READ</li> <li>• +ALLOW_WRITE</li> </ul> <p>오프라인 및 RECLUSTER 옵션의 경우 다음 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• +INDEXSCAN</li> <li>• +TABLESCAN</li> </ul> <p>오프라인의 경우 다음 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• +LONGLOB</li> <li>• +DATAONLY</li> </ul> <p>오프라인이고 옵션이 지정된 경우 다음 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• +CHOOSE_TEMP</li> <li>• +KEEPDICTIONARY</li> <li>• +RESETDICTIONARY</li> </ul> <p>온라인이고 옵션이 지정된 경우,</p> <ul style="list-style-type: none"> <li>• +NOTRUNCATE</li> </ul> <p>예 1: REORG TABLE TEST.EMPLOYEE가 실행되면 다음과 같이 표시됩니다.</p> <p>RECLAIM+OFFLINE+ALLOW_READ+DATAONLY +KEEPDICTIONARY</p> <p>예 2: REORG TABLE TEST.EMPLOYEE INDEX EMPIDX INDEXSCAN이 실행되면 다음과 같이 표시됩니다.</p> <p>RECLUSTER+OFFLINE+ALLOW_READ+INDEXSCAN +DATAONLY+KEEPDICTIONARY</p>

표 188. SNAPTAB\_REORG 관리 뷰 및 SNAP\_GET\_TAB\_REORG 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
REORG_STATUS	VARCHAR (10)	reorg_status - 테이블 재구성 상태. 이 인터페이스는 sqlmon.h의 정의를 기본으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 테이블 재구성 완료 플래그. 이 인터페이스는 sqlmon.h의 정의를 기본으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• FAIL</li> <li>• SUCCESS</li> </ul>
REORG_START	TIMESTAMP	reorg_start - 테이블 재구성 시작 시간
REORG_END	TIMESTAMP	reorg_end - 테이블 재구성 종료 시간
REORG_PHASE_START	TIMESTAMP	reorg_phase_start - 테이블 재구성 단계 시작 시간
REORG_INDEX_ID	BIGINT	reorg_index_id - 테이블 재구성에 사용된 인덱스
REORG_TBSPC_ID	BIGINT	reorg_tbsp_id - 테이블이 재구성되는 테이블 스페이스
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
DATA_PARTITION_ID	INTEGER	data_partition_id - 데이터 파티션 ID. 파티션되지 않은 테이블의 경우 이 요소는 널(NULL)이 됩니다.
REORG_ROWSCOMPRESSED	BIGINT	reorg_rows_compressed - 압축된 행
REORG_ROWSREJECTED	BIGINT	reorg_rows_rejected_for_compression - 압축에 대해 거부된 행
REORG_LONG_TBSPC_ID	BIGINT	reorg_long_tbsp_id - long 오브젝트가 재구성된 테이블 스페이스

## SNAPTBSP 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수 - 테이블 스페이스 논리 데이터 그룹 스냅샷 정보 검색

SNAPTBSP 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수는 테이블 스페이스 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.

### SNAPTBSP 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 테이블 스페이스 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.



SNAPTbsp\_PART, SNAPTbsp\_QUIESCER, SNAPTbsp\_RANGE, SNAPCONTAINER 관리 뷰와 같이 사용하여 SNAPTbsp 관리 뷰는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 641 페이지의 표 159의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPTbsp 관리 뷰에 대한 SELECT 특권
- SNAPTbsp 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_TBSP\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 데이터베이스에 대한 카탈로그 데이터베이스 파티션에서 테이블 스페이스 목록을 검색합니다.

```
SELECT SUBSTR(TBSP_NAME,1,30) AS TBSP_NAME, TBSP_ID, TBSP_TYPE,
       TBSP_CONTENT_TYPE FROM SYSIBMADM.SNAPTbsp WHERE DBPARTITIONNUM = 1
```

다음은 이 쿼리의 출력 예입니다.

TBSP_NAME	TBSP_ID	TBSP_TYPE	TBSP_CONTENT_TYPE
TEMPSPACE1	1	SMS	SYSTEMP
USERSPACE1	2	DMS	LONG

2개의 레코드가 선택되었습니다.

## SNAP\_GET\_TBSP\_V91 테이블 함수

SNAP\_GET\_TBSP\_V91 테이블 함수는 SNAPTbsp 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

SNAP\_GET\_TBSP\_PART\_V91, SNAP\_GET\_TBSP\_QUIESCER, SNAP\_GET\_TBSP\_RANGE, SNAP\_GET\_CONTAINER\_V91 테이블 함수와 함께 사용되는 SNAP\_GET\_TBSP\_V91 테이블 함수는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 641 페이지의 표 159의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_TBSP_V91 ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. NULL 또는 빈 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 얻으십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_TBSP\_V91 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대해 스냅샷을 작성합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_TBSP\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

현재 연결된 데이터베이스의 모든 데이터베이스 파티션에 대한 테이블 스페이스 목록을 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

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

8개의 레코드가 선택되었습니다.

## 리턴된 정보

표 189. SNAPTbsp 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
TBSP_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID

표 189. SNAPTBSP 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TBSP_TYPE	VARCHAR(10)	tablespace_type - 테이블 스페이스 유형. 이 인터페이스는 sqlutil.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• DMS</li> <li>• SMS</li> </ul>
TBSP_CONTENT_TYPE	VARCHAR(10)	tablespace_content_type - 테이블 스페이스 콘텐츠 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 테이블 스페이스 페이지 크기
TBSP_EXTENT_SIZE	BIGINT	tablespace_extent_size - 테이블 스페이스 Extent 크기
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 테이블 스페이스 프리페치 크기
TBSP_CUR_POOL_ID	BIGINT	tablespace_cur_pool_id - 현재 사용 중인 버퍼 풀
TBSP_NEXT_POOL_ID	BIGINT	tablespace_next_pool_id - 다음 시작 시에 사용되는 버퍼 풀
FS_CACHING	SMALLINT	fs_caching - 파일 시스템 캐싱
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수

표 189. SNAPTBSP 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - 버퍼 풀 비동기 읽기 시간
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - 버퍼 풀 비동기 쓰기 시간
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 수
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - 버퍼 풀 희생 버퍼 없음

표 189. SNAPTbsp 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
FILES_CLOSED	BIGINT	files_closed - 닫힌 데이터베이스 파일 수
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리페치 페이지 수
TBSP_REBALANCER_MODE	VARCHAR(10)	tablespace_rebalancer_mode - 재조정 프로그램 모드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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 - 테이블 공간을 자동 스토리지에 사용 가능
TBSP_AUTO_RESIZE_ENABLED	SMALLINT	tablespace_auto_resize_enabled - 테이블 공간 자동 크기 조정 가능
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPTbsp\_PART 관리 뷰 및 SNAP\_GET\_TBSP\_PART\_V91 테이블 함수 - tablespace\_nodeinfo 논리 데이터 그룹 스냅샷 정보 검색

주: 이 테이블 함수는 사용되지 않으며 644 페이지의 『SNAPTbsp\_PART 관리 뷰 및 SNAP\_GET\_TBSP\_PART\_V97 테이블 함수 - tablespace\_nodeinfo 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

SNAPTbsp\_PART 관리 뷰 및 SNAP\_GET\_TBSP\_PART\_V91 테이블 함수는 tablespace\_nodeinfo 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.

### SNAPTbsp\_PART 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 tablespace\_nodeinfo 논리 데이터 그룹 스냅샷 정보를 검색할 수 있습니다.

SNAPTBSP, SNAPTBSP\_QUIESCER, SNAPTBSP\_RANGE, SNAPCONTAINER 관리 뷰와 같이 사용하여 SNAPTBSP\_PART 관리 뷰는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 797 페이지의 표 190의 내용을 참조하십시오.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPTBSP\_PART 관리 뷰에 대한 SELECT 특권
- SNAPTBSP\_PART 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_TBSP\_PART\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

현재 연결된 데이터베이스의 모든 데이터베이스 파티션에 대한 테이블 스페이스 목록 및 해당 상태를 검색합니다.

```
SELECT SUBSTR(TBSP_NAME,1,30) AS TBSP_NAME, TBSP_ID,
       SUBSTR(TBSP_STATE,1,30) AS TBSP_STATE, DBPARTITIONNUM
FROM SYSIBMADM.SNAPTBSP_PART
```

다음은 이 쿼리의 출력 예입니다.

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

5개의 레코드가 선택되었습니다.

## SNAP\_GET\_TBSP\_PART\_V91 테이블 함수

SNAP\_GET\_TBSP\_PART\_V91 테이블 함수는 SNAPTbsp 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_TBSP\_V91, SNAP\_GET\_TBSP\_QUIESCER, SNAP\_GET\_TBSP\_RANGE, SNAP\_GET\_CONTAINER\_V91 테이블 함수와 함께 사용되는 경우, SNAP\_GET\_TBSP\_PART\_V91 테이블 함수는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 정보를 리턴합니다.

리턴 가능한 전체 정보 목록은 797 페이지의 표 190의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_TBSP_PART_V91 ( (dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 NULL 또는 비어 있는 문자열을 지정하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우 -1이 *dbpartitionnum*에 대해 내재적으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_TBSP\_PART\_V91 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.



## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_TBSP\_PART\_V91 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

연결된 데이터베이스의 연결된 데이터베이스 파티션에 대한 테이블 스페이스 목록 및 해당 상태를 검색합니다.

```
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_V91(CAST(NULL AS VARCHAR(128)),-1)) AS T
```

다음은 이 쿼리의 출력 예입니다.

TBSP_NAME	TBSP_ID	TBSP_STATE
SYSCATSPACE		0 NORMAL
TEMPSPACE1		1 NORMAL
USERSPACE1		2 NORMAL
SYSTOOLSPACE		3 NORMAL
SYSTOOLSTMPSPACE		4 NORMAL

5개의 레코드가 선택되었습니다.

## 리턴된 정보

표 190. SNAP\_TBSP\_PART 관리 뷰 및 SNAP\_GET\_TBSP\_PART\_V91 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
TBSP_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID

표 190. SNAPTBSP\_PART 관리 뷰 및 SNAP\_GET\_TBSP\_PART\_V91 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TBSP_STATE	VARCHAR (256)	<p>tablespace_state - 테이블 스페이스 상태. 이 인터페이스는 sqlutil.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 '+' 기호로 구분된 다음의 조합입니다.</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 - 테이블 스페이스 프리페치 크기
TBSP_NUM QUIESCERS	BIGINT	tablespace_num_quiescers - quiescer 수
TBSP_STATE_CHANGE_OBJECT_ID	BIGINT	tablespace_state_change_object_id - 상태 변경 오브젝트 ID
TBSP_STATE_CHANGE_TBSP_ID	BIGINT	tablespace_state_change_ts_id - 상태 변경 테이블 스페이스 ID

표 190. SNAPTBSP\_PART 관리 뷰 및 SNAP\_GET\_TBSP\_PART\_V91 테이블 함수에서 리턴된 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TBSP_MIN_RECOVERY_ TIME	TIMESTAMP	tablespace_min_recovery_time - 롤 포워드를 위한 최소 복구 시간
TBSP_TOTAL_PAGES	BIGINT	tablespace_total_pages - 테이블 스페이스의 전체 페이지
TBSP_USABLE_PAGES	BIGINT	tablespace_usable_pages - 테이블 스페이스에서 사 용할 수 있는 페이지
TBSP_USED_PAGES	BIGINT	tablespace_used_pages - 테이블 스페이스에서 사 용된 페이지
TBSP_FREE_PAGES	BIGINT	tablespace_free_pages - 테이블 스페이스에서 사용 가능한 페이지
TBSP_PENDING_FREE_ PAGES	BIGINT	tablespace_pending_free_pages - 테이블 스페이스 에서 보류 중인 사용 가능한 페이지
TBSP_PAGE_TOP	BIGINT	tablespace_page_top - 테이블 스페이스 상위 워터 마크(water mark)
REBALANCER_MODE	VARCHAR (10)	tablespace_rebalancer_mode - 재조정 프로그램 모 드. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• FWD_REBAL</li> <li>• NO_REBAL</li> <li>• REV_REBAL</li> </ul>
REBALANCER_EXTENTS_ REMAINING	BIGINT	tablespace_rebalancer_extents_remaining - 재조정 프로그램에서 처리되는 전체 Extent 수
REBALANCER_EXTENTS_ PROCESSED	BIGINT	tablespace_rebalancer_extents_processed - 재조정 프로그램이 처리한 Extent 수
REBALANCER_PRIORITY	BIGINT	tablespace_rebalancer_priority - 현재 재조정 프로 그램 우선순위
REBALANCER_START_ TIME	TIMESTAMP	tablespace_rebalancer_start_time - 재조정 프로그 램 시작 시간
REBALANCER_RESTART_ TIME	TIMESTAMP	tablespace_rebalancer_restart_time - 재조정 프로 그램 재시작 시간
REBALANCER_LAST_ EXTENT_MOVED	BIGINT	tablespace_rebalancer_last_extent_moved - 재조정 프로그램이 이동한 마지막 Extent
TBSP_NUM_RANGES	BIGINT	tablespace_num_ranges - 테이블 스페이스 맵의 범 위 수
TBSP_NUM_CONTAINERS	BIGINT	tablespace_num_containers - 테이블 스페이스의 컨 테이너 수

표 190. SNAPTbsp\_Part 관리 뷰 및 Snap\_Get\_Tbsp\_Part\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
Tbsp_Initial_Size	BIGINT	tablespace_initial_size - 초기 테이블 스페이스 크기
Tbsp_Current_Size	BIGINT	tablespace_current_size - 현재 테이블 스페이스 크기
Tbsp_Max_Size	BIGINT	tablespace_max_size - 최대 테이블 스페이스 크기
Tbsp_Increase_Size	BIGINT	tablespace_increase_size - 바이트 단위로 크기 늘리기
Tbsp_Increase_Size_Percent	SMALLINT	tablespace_increase_size_percent - 퍼센트 단위로 크기 늘리기
Tbsp_Last_Resize_Time	TIMESTAMP	tablespace_last_resize_time - 마지막으로 성공한 크기 조정 시간
Tbsp_Last_Resize_Failed	SMALLINT	tablespace_last_resize_failed - 마지막으로 실패한 크기 조정 시도
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPTbsp\_Quiescer 관리 뷰 및 Snap\_Get\_Tbsp\_Quiescer 테이블 함수 - Quiescer 테이블 스페이스 스냅샷 정보 검색

SNAPTbsp\_Quiescer 관리 뷰 및 Snap\_Get\_Tbsp\_Quiescer 테이블 함수는 테이블 스페이스 스냅샷에서 Quiescer에 대한 정보를 리턴합니다.

### SNAPTbsp\_Quiescer 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 Quiescer 테이블 스페이스 스냅샷 정보를 검색할 수 있습니다.

SNAPTbsp, SNAPTbsp\_Part, SNAPTbsp\_Range, SnapContainer 관리 뷰와 같이 사용하여 SNAPTbsp\_Quiescer 관리 뷰는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴 가능한 전체 정보 목록은 653 페이지의 표 161의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPTbsp\_Quiescer 관리 뷰에 대한 SELECT 특권

- SNAPTbsp\_QUIESCER 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_Tbsp\_QUIESCER 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

**예 :**

현재 연결된 데이터베이스의 모든 데이터베이스 파티션에 대한 quiesce된 테이블 스페이스에서 정보를 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

Tbsp_NAME	QUIESCER_TS_ID	QUIESCER_OBJ_ID	QUIESCER_AUTH_ID	..
USERSPACE1	2	5	SWALKTY	..
USERSPACE1	2	5	SWALKTY	..

2개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	QUIESCER_AGENT_ID	QUIESCER_STATE	DBPARTITIONNUM
...	0	EXCLUSIVE	0
...	65983	EXCLUSIVE	1

**예: 범위 파티션 테이블 이름을 판별합니다.**

테이블 범위가 파티션되었고 Quiesce 상태인 경우 테이블 스페이스 ID 및 테이블 ID는 SYSCAT.TABLES에 표시된 값과 다르게 표시됩니다. 이 ID는 부호없이 짧게 표시됩니다. Quiesce 상태의 테이블 이름을 찾으려면 다음과 같은 방법으로 우선 부호있는 짧은 표시를 찾으십시오. QUIESCER\_TS\_ID에서 65536(최대값)을 빼고 이 테이블 스페이스 ID를 사용하여 Quiesce 상태의 테이블을 찾으십시오. (실제 테이블 스페이스 ID는 테이블에서 각 범위 파티션에 대한 SYSCAT.DATAPARTITIONS에 있습니다.)

```

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

```

이 쿼리의 출력 예는 다음과 같습니다.

TBSP_NAME	TBSPACEID	TABLEID
TABDATA	-6	-32768
DATAMART	-6	-32765
SMALL	5	17

3개의 레코드가 선택되었습니다.

위 쿼리에서 제공된, 지정된 TBSPACEID 및 TABLEID를 사용하여 SYSCAT.TABLES에서 테이블 스키마 및 이름을 찾으십시오.

```

SELECT CHAR(tabschema, 10)tabschema, CHAR(tabname,15)tabname
FROM SYSCAT.TABLES
WHERE tbspaceid = -6 AND tableid in (-32768,-32765)

```

다음은 이 쿼리의 출력 예입니다.

TABSCHEMA	TABNAME
TPCD	ORDERS_RP
TPCD	ORDERS_DMART

2개의 레코드가 선택되었습니다.

```

SELECT CHAR(tabschema, 10)tabschema, CHAR(tabname,15)tabname
FROM SYSCAT.TABLES
WHERE tbspaceid = 5 AND tableid = 17

```

다음은 이 쿼리의 출력 예입니다.

TABSCHEMA	TABNAME
TPCD	NATION

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_TBSP QUIESCER 테이블 함수

SNAP\_GET\_TBSP QUIESCER 테이블 함수는 SNAPTbsp QUIESCER 관리 뷰와 동일한 정보를 리턴하지만 이를 사용하여 특정 데이터베이스 파티션, 모든 데이터베이스 파티션 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있습니다.

SNAP\_GET\_TBSP\_V91, SNAP\_GET\_TBSP\_PART\_V91, SNAP\_GET\_TBSP\_RANGE, SNAP\_GET\_CONTAINER\_V91 테이블 함수와 함께

사용되는 SNAP\_GET\_TBSP\_QUIESCER 테이블 함수는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동등한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 653 페이지의 표 161의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_TBSP_QUIESCER ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 NULL 또는 비어 있는 문자열을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우 -1이 *dbpartitionnum*에 대해 내재적으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_TBSP\_QUIESCER 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대해 스냅샷을 작성합니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_TBSP\_QUIESCER 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON

- SYSCTRL
- SYSMOINT
- SYSADM

예 :

현재 연결된 데이터베이스의 데이터베이스 파티션 1에 대한 quiesce된 테이블 스페이스에서 정보를 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

```
TBSP_NAME QUIESCER_TS_ID QUIESCER_OBJ_ID QUIESCER_AUTH_ID ...
-----
USERSPACE1 2 5 SWALKTY ...
```

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... QUIESCER_AGENT_ID QUIESCER_STATE DBPARTITIONNUM
... -----
... 65983 EXCLUSIVE 1
```

### 리턴된 정보

표 191. SNAP\_TBSP QUIESCER 관리 뷰 및 SNAP\_GET\_TBSP QUIESCER 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
TBSP_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름
QUIESCER_TS_ID	BIGINT	quiescer_ts_id - Quiescer 테이블 스페이스 ID
QUIESCER_OBJ_ID	BIGINT	quiescer_obj_id - Quiescer 오브젝트 ID
QUIESCER_AUTH_ID	VARCHAR(128)	quiescer_auth_id - Quiescer 사용자 권한 부여 ID
QUIESCER_AGENT_ID	BIGINT	quiescer_agent_id - Quiescer 에이전트 ID
QUIESCER_STATE	VARCHAR(14)	quiescer_state - Quiescer 상태. 이 인터페이스는 sqlutil.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• EXCLUSIVE</li> <li>• UPDATE</li> <li>• SHARE</li> </ul>



표 191. SNAPTbsp\_QUIESCER 관리 뷰 및 SNAP\_GET\_TBSP\_QUIESCER 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPTbsp\_RANGE 관리 뷰 및 SNAP\_GET\_TBSP\_RANGE 테이블 함수 - 범위 스냅샷 정보 검색

SNAPTbsp\_RANGE 관리 뷰 및 SNAP\_GET\_TBSP\_RANGE 테이블 함수는 범위 스냅샷에서 정보를 리턴합니다.

### SNAPTbsp\_RANGE 관리 뷰

이 관리 뷰를 사용하여 현재 연결된 데이터베이스에 대한 범위 스냅샷 정보를 검색할 수 있습니다.

SNAPTbsp, SNAPTbsp\_PART, SNAPTbsp\_QUIESCER 및 SNAPCONTAINER 관리 뷰와 함께 사용될 경우, SNAPTbsp\_RANGE 관리 뷰는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 658 페이지의 표 162의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPTbsp\_RANGE 관리 뷰에 대한 SELECT 특권
- SNAPTbsp\_RANGE 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_TBSP\_RANGE 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

현재 연결된 데이터베이스에 대한 모든 데이터베이스 파티션의 테이블 스페이스 범위 정보를 선택합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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 ...

5개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	RANGE_OFFSET	RANGE_MAX_PAGE	RANGE_MAX_EXTENT	...
...	0	11515	2878	...
...	0	479	14	...
...	0	251	62	...
...	0	479	14	...
...	0	479	14	...

이 쿼리의 출력(계속).

...	RANGE_START_STRIPE	RANGE_END_STRIPE	RANGE_ADJUSTMENT	...
...	0	2878	0	...
...	0	14	0	...
...	0	62	0	...
...	0	14	0	...
...	0	14	0	...

이 쿼리의 출력(계속).

...	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 테이블 함수

SNAP\_GET\_TBSP\_RANGE 테이블 함수는 SNAPTbsp\_RANGE 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_TBSP\_V91, SNAP\_GET\_TBSP\_PART\_V91, SNAP\_GET\_TBSP QUIESCER 및 SNAP\_GET\_CONTAINER\_V91 테이블 함수와 함께 사용될 경우, SNAP\_GET\_TBSP\_RANGE 테이블 함수는 GET SNAPSHOT FOR TABLESPACES ON database-alias CLP 명령과 동일한 정보를 제공합니다.

리턴 가능한 전체 정보 목록은 658 페이지의 표 162의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_TBSP_RANGE ( (dbname [ , dbpartitionnum ] ) )
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 NULL 또는 비어 있는 문자열을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. *dbname*이 NULL로 설정되어 있지 않고 *dbpartitionnum*이 NULL로 설정되어 있다면, -1은 내재적으로 *dbpartitionnum*으로 설정됩니다. 이 입력 옵션이 사용되지 않은 경우 즉, *dbname*만이 제공된 경우, 데이터는 모든 활성 데이터베이스 파티션으로부터 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결에 사용할 수 있으며 응용프로그램에서 사용할 수 있는 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_TBSP\_RANGE 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_TBSP\_RANGE 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예:

현재 연결된 데이터베이스 파티션에서 `tbasp_id = 2`인 테이블 스페이스에 대해 테이블 스페이스 범위 정보를 선택합니다.

```
SELECT TBASP_ID, SUBSTR(TBASP_NAME, 1, 15) AS TBASP_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_TBASP_RANGE('',-1)) AS T WHERE TBASP_ID = 2
```

이 쿼리의 출력 예는 다음과 같습니다.

```
TBASP_ID    TBASP_NAME    RANGE_NUMBER    ...
-----
          2  USERSPACE1                0 ...
```

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... RANGE_STRIPE_SET_NUMBER RANGE_OFFSET    RANGE_MAX_PAGE    ...
... RANGE_STRIPE_SET_NUMBER RANGE_OFFSET    RANGE_MAX_PAGE    ...
... -----
...                               0            0                3967 ...
```

이 쿼리의 출력(계속).

```
... RANGE_MAX_EXTENT    RANGE_START_STRIPE    RANGE_END_STRIPE    ...
... -----
...                    123                0                    123 ...
```

이 쿼리의 출력(계속).

```
... RANGE_ADJUSTMENT    RANGE_NUM_CONTAINER    RANGE_CONTAINER_ID
... -----
...                    0                1                    0
```

## 리턴되는 정보

표 192. `SNAPTASP_RANGE` 관리 뷰 및 `SNAP_GET_TBASP_RANGE` 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
<code>SNAPSHOT_TIMESTAMP</code>	<code>TIMESTAMP</code>	스냅샷을 작성한 날짜와 시간.
<code>TBASP_ID</code>	<code>BIGINT</code>	<code>tablespace_id</code> - 테이블 스페이스 ID
<code>TBASP_NAME</code>	<code>VARCHAR</code> (128)	<code>tablespace_name</code> - 테이블 스페이스 이름

표 192. SNAPTBSP\_RANGE 관리 뷰 및 SNAP\_GET\_TBSP\_RANGE 테이블 함수에서 리턴한 정보  
(계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
RANGE_NUMBER	BIGINT	range_number - 범위 번호
RANGE_STRIPE_SET_NUMBER	BIGINT	range_stripe_set_number - 스트라이프 세트 번호
RANGE_OFFSET	BIGINT	range_offset - 범위 오프셋
RANGE_MAX_PAGE	BIGINT	range_max_page_number - 범위의 최대 페이지
RANGE_MAX_EXTENT	BIGINT	range_max_extent - 범위의 최대 Extent
RANGE_START_STRIPE	BIGINT	range_start_stripe - 시작 스트라이프
RANGE_END_STRIPE	BIGINT	range_end_stripe - 끝 스트라이프
RANGE_ADJUSTMENT	BIGINT	range_adjustment - 범위 조정
RANGE_NUM_CONTAINER	BIGINT	range_num_containers - 범위의 컨테이너 수
RANGE_CONTAINER_ID	BIGINT	range_container_id - 범위 컨테이너
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAPUTIL 관리 뷰 및 SNAP\_GET\_UTIL 테이블 함수 - utility\_info 논리 데이터 그룹 스냅샷 정보 검색

SNAPUTIL 관리 뷰 및 SNAP\_GET\_UTIL 테이블 함수는 utility\_info 논리 데이터 그룹에서 유틸리티에 대한 스냅샷 정보를 리턴합니다.

### SNAPUTIL 관리 뷰

SNAPUTIL\_PROGRESS 관리 뷰와 함께 사용될 경우, SNAPUTIL 관리 뷰는 LIST UTILITIES SHOW DETAIL CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 661 페이지의 표 163의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPUTIL 관리 뷰에 대한 SELECT 특권
- SNAPUTIL 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_UTIL 테이블 함수에 대한 EXECUTE 특권

- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

연결된 데이터베이스를 포함하는 인스턴스에서 모든 활성 데이터베이스에 대한 모든 데이터베이스 파티션의 유틸리티 및 상태 목록을 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
UTILITY_TYPE      UTILITY_PRIORITY ...
-----...- ----- ...
LOAD              - ...
LOAD              - ...
LOAD              - ...
```

3개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... 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 ...
```

이 쿼리의 출력(계속).

```
... UTILITY_DBNAME      UTILITY_STATE UTILITY_INVOKER_TYPE DBPARTITIONNUM
... -----
... SAMPLE              EXECUTE      USER              0
... SAMPLE              EXECUTE      USER              1
... SAMPLE              EXECUTE      USER              2
```

## SNAP\_GET\_UTIL 테이블 함수

SNAP\_GET\_UTIL 테이블 함수는 SNAPUTIL 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_UTIL\_PROGRESS 테이블 함수와 함께 사용될 경우, SNAP\_GET\_UTIL 테이블 함수는 LIST UTILITIES SHOW DETAIL CLP 명령과 동일한 정보를 제공합니다.

리턴될 수 있는 정보의 완전한 목록은 661 페이지의 표 163의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_UTIL ( [ dbpartitionnum ] ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_UTIL 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_UTIL 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

## 예 :

SAMPLE 데이터베이스의 현재 연결된 데이터베이스 파티션에 대한 유틸리티 ID, 유형 및 상태 목록을 검색합니다.

```
SELECT UTILITY_ID, UTILITY_TYPE, STATE
FROM TABLE(SNAP_GET_UTIL(-1)) AS T WHERE UTILITY_DBNAME='SAMPLE'
```

이 쿼리의 출력 예는 다음과 같습니다.

```
UTILITY_ID          UTILITY_TYPE          STATE
-----
1 BACKUP                               EXECUTE
```

1개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 193. SNAPUTIL 관리 뷰 및 SNAP\_GET\_UTIL 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
UTILITY_ID	INTEGER	utility_id - 유틸리티 ID. 데이터베이스 파티션에 고유함.
UTILITY_TYPE	VARCHAR(26)	utility_type - 유틸리티 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <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>• REORG</li> <li>• RESTART_RECREATE_INDEX</li> <li>• RESTORE</li> <li>• ROLLFORWARD_RECOVERY</li> <li>• RUNSTATS</li> </ul>
UTILITY_PRIORITY	INTEGER	utility_priority - 유틸리티 우선순위. 유틸리티에서 조절 기능을 지원하는 경우 우선순위, 그렇지 않은 경우 널(NULL).
UTILITY_DESCRIPTION	VARCHAR(2048)	utility_description - 유틸리티 설명. 결과는 널(NULL)이 될 수 있습니다.
UTILITY_DBNAME	VARCHAR(128)	utility_dbname - 유틸리티가 작동하는 데이터베이스
UTILITY_START_TIME	TIMESTAMP	utility_start_time - 유틸리티 시작 시간



표 193. SNAPUTIL 관리 뷰 및 SNAP\_GET\_UTIL 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
UTILITY_STATE	VARCHAR(10)	utility_state - 유틸리티 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• ERROR</li> <li>• EXECUTE</li> <li>• WAIT</li> </ul>
UTILITY_INVOKER_TYPE	VARCHAR(10)	utility_invoker_type - 유틸리티 호출자 유형. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• AUTO</li> <li>• USER</li> </ul>
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
PROGRESS_LIST_ATTR	VARCHAR(10)	progress_list_attr - 현재 진행 목록 속성
PROGRESS_LIST_CUR_SEQ_NUM	INTEGER	progress_list_current_seq_num - 현재 진행 목록 시퀀스 수

## SNAPUTIL\_PROGRESS 관리 뷰 및 SNAP\_GET\_UTIL\_PROGRESS 테이블 함수 - 진행 중 논리 데이터 그룹 스냅샷 정보 검색

SNAPUTIL\_PROGRESS 관리 뷰 및 SNAP\_GET\_UTIL\_PROGRESS 테이블 함수는 유틸리티 진행에 대한 스냅샷 정보 특히 진행 논리 데이터 그룹에 대한 스냅샷 정보를 리턴합니다.

### SNAPUTIL\_PROGRESS 관리 뷰

SNAPUTIL 관리 뷰와 함께 사용될 경우, SNAPUTIL\_PROGRESS 관리 뷰는 LIST UTILITIES SHOW DETAIL CLP 명령과 동일한 정보를 제공합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 665 페이지의 표 164의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAPUTIL\_PROGRESS 관리 뷰에 대한 SELECT 특권
- SNAPUTIL\_PROGRESS 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- SNAP\_GET\_UTIL\_PROGRESS 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

진행의 총계 및 완료된 단위에 대한 세부사항을 유틸리티 ID별로 검색합니다.

```
SELECT SELECT UTILITY_ID, PROGRESS_TOTAL_UNITS, PROGRESS_COMPLETED_UNITS,
        DBPARTITIONNUM FROM SYSIBMADM.SNAPUTIL_PROGRESS
```

이 쿼리의 출력 예는 다음과 같습니다.

UTILITY_ID	PROGRESS_TOTAL_UNITS	PROGRESS_COMPLETED_UNITS	DBPARTITIONNUM
7	10	5	0
9	10	5	1

1개의 레코드가 선택되었습니다.

## SNAP\_GET\_UTIL\_PROGRESS 테이블 함수

SSNAP\_GET\_UTIL\_PROGRESS 테이블 함수는 SNAPUTIL\_PROGRESS 관리 뷰와 동일한 정보를 리턴하지만, 특정 데이터베이스 파티션, 모든 데이터베이스 파티션의 집계 또는 모든 데이터베이스 파티션에서 특정 데이터베이스에 대한 정보를 검색할 수 있게 해줍니다.

SNAP\_GET\_UTIL 테이블 함수와 함께 사용될 경우, SNAP\_GET\_UTIL\_PROGRESS 테이블 함수는 LIST UTILITIES SHOW DETAIL CLP 명령과 동일한 정보를 제공합니다.

리턴될 수 있는 정보의 완전한 목록은 665 페이지의 표 164의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_UTIL_PROGRESS ( ( dbpartitionnum ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 대응하는 스냅샷 API 요청 유형을 가진 파일이 존재하지 않을 경우, SNAP\_GET\_UTIL\_PROGRESS 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대한 스냅샷을 가져옵니다.

## 권한 부여

다음 권한 중 하나가 필요합니다.

- SNAP\_GET\_UTIL\_PROGRESS 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMAINT
- SYSADM

## 예 :

현재 연결 파티션에 대한 유틸리티 진행 세부사항을 검색합니다.

```
SELECT UTILITY_ID, PROGRESS_TOTAL_UNITS, PROGRESS_COMPLETED_UNITS,  
       DBPARTITIONNUM FROM TABLE(SNAP_GET_UTIL_PROGRESS(-1)) as T
```

이 쿼리의 출력 예는 다음과 같습니다.

```
UTILITY_ID PROGRESS_TOTAL_UNITS PROGRESS_COMPLETED_UNITS DBPARTITIONNUM  
-----  
              7                10                        5                0
```

1개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 194. SNAPUTIL\_PROGRESS 관리 뷰 및 SNAP\_GET\_UTIL\_PROGRESS 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
UTILITY_ID	INTEGER	utility_id - 유틸리티 ID. 데이터베이스 파티션에 고유함.
PROGRESS_SEQ_NUM	INTEGER	progress_seq_num - 진행 시퀀스 수. 직렬일 경우 단계의 수, 동시일 경우 널(NULL)일 수 있습니다.
UTILITY_STATE	VARCHAR(16)	utility_state - 유틸리티 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• ERROR</li> <li>• EXECUTE</li> <li>• WAIT</li> </ul>
PROGRESS_DESCRIPTION	VARCHAR(2048)	progress_description - 진행 설명
PROGRESS_START_TIME	TIMESTAMP	progress_start_time - 진행 시작 시간. 단계가 시작된 경우 시작 시간, 그렇지 않은 경우 널(NULL).
PROGRESS_WORK_METRIC	VARCHAR(16)	progress_work_metric - 진행 작업 메트릭. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• NOT_SUPPORT</li> <li>• BYTES</li> <li>• EXTENTS</li> <li>• 인텍스</li> <li>• PAGES</li> <li>• ROWS</li> <li>• TABLES</li> </ul>
PROGRESS_TOTAL_UNITS	BIGINT	progress_total_units - 전체 진행 작업 단위
PROGRESS_COMPLETED_UNITS	BIGINT	progress_completed_units - 완료된 진행 작업 단위
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAP\_WRITE\_FILE 프로시저

SNAP\_WRITE\_FILE 프로시저는 인스턴스 디렉토리의 tmp 서브디렉토리에 있는 파일에 시스템 스냅샷 데이터를 기록합니다.

## 구문

▶▶—SNAP\_WRITE\_FILE—(—requestType—,—dbName—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *requestType*

유효한 스냅샷 요청 유형을 지정하는 유형 VARCHAR(32)의 입력 인수. 가능한 요청 유형은 sqlmon.h의 정의를 기반으로 하는 텍스트 ID이며 다음 중 하나입니다.

- APPL\_ALL
- BUFFERPOOLS\_ALL
- DB2
- DBASE\_ALL
- DBASE\_LOCKS
- DBASE\_TABLES
- DBASE\_TABLESPACES
- DYNAMIC\_SQL

#### *dbName*

해당 함수 호출시 현재 연결된 데이터베이스와 동일한 인스턴스에 있는 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. NULL 또는 빈 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 얻으십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정되면, -1이 내재적으로 설정됩니다.

### 권한 부여

프로시저를 실행하려면, 사용자에게 SYSADM, SYSCTRL, SYSMANT 또는 SYSMON 권한이 있어야 합니다. 저장된 스냅샷은 스냅샷 테이블 함수에 대한 입력으로 널(NULL) 값을 전달함으로써 SYSADM, SYSCTRL, SYSMANT 또는 SYSMON 권한이 없는 사용자가 읽을 수 있습니다.

예 :

요청 유형 'DB2'(SQLMA\_DB2에 해당)을 지정하고 현재 연결된 데이터베이스 및 현재 데이터베이스 파티션을 디폴트값으로 하여 데이터베이스 관리 프로그램 정보의 스냅샷을 취하십시오.

```
CALL SYSPROC.SNAP_WRITE_FILE ('DB2', '', -1)
```

이는 인스턴스 임시 디렉토리(UNIX 운영 체제의 경우 sqllib/tmp/SQLMA\_DB2.dat 및 Windows 운영 체제의 경우 sqllib\#DB2\#tmp\SQLMA\_DB2.dat)에 스냅샷 데이터를 작성합니다.

### 사용 시 참고사항

인식되지 않는 입력 매개변수를 제공할 경우, 다음과 같은 오류가 리턴됩니다.  
SQL2032N: "REQUEST\_TYPE" 매개변수가 유효하지 않습니다.

---

## TBSP\_UTILIZATION 관리 뷰 - 테이블 스페이스 구성 및 활용 정보 검색

TBSP\_UTILIZATION 관리 뷰는 테이블 스페이스 구성 및 활용 정보를 리턴합니다. 뷰는 LIST TABLESPACES CLP 명령에 대한 SQL 인터페이스입니다. 해당 정보는 SNAPTbsp, SNAPTbsp\_PART 관리 뷰 및 TABLESPACES 카탈로그 뷰를 바탕으로 합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- TBSP\_UTILIZATION 관리 뷰에 대한 SELECT 특권
- TBSP\_UTILIZATION 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

단일 파티션 데이터베이스에서 LIST TABLESPACES 명령과 동일한 보고서를 검색합니다.

```
SELECT TBSP_ID, SUBSTR(TBSP_NAME,1,20) as TBSP_NAME, TBSP_TYPE,
       TBSP_CONTENT_TYPE, TBSP_STATE FROM SYSIBMADM.TBSP_UTILIZATION
```

다음은 이 쿼리에 대한 출력 예입니다.

```
TBSP_ID    TBSP_NAME          TBSP_TYPE    ...
-----
          0 SYSCATSPACE      SMS          ...
          1 TEMPSPACE1    SMS          ...
          2 USERSPACE1    SMS          ...
          3 SYSTOOLSPACE  SMS          ...
          4 SYSTOOLSTMPSPACE SMS          ...
```

이 쿼리의 출력(계속).

```
... TBSP_CONTENT_TYPE TBSP_STATE
... -----
... ANY                NORMAL
... SYSTEMP           NORMAL
... ANY                NORMAL
... ANY                NORMAL
... USRTEMP           NORMAL
```

## 리턴된 정보

표 195. TBSP\_UTILIZATION 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
TBSP_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
TBSP_TYPE	VARCHAR(10)	tablespace_type - 테이블 스페이스 유형. 이 인터페이스는 sqlutil.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• DMS</li> <li>• SMS</li> </ul>
TBSP_CONTENT_TYPE	VARCHAR(10)	tablespace_content_type - 테이블 스페이스 콘텐츠 유형. 이 인터페이스는 sqlutil.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• ANY</li> <li>• LONG</li> <li>• SYSTEMP</li> <li>• USRTEMP</li> </ul>
TBSP_CREATE_TIME	TIMESTAMP	테이블 스페이스 작성시간입니다.

표 195. TBSP\_UTILIZATION 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TBSP_STATE	VARCHAR (256)	<p>tablespace_state - 테이블 스페이스 상태. 이 인터페이스는 sqlutil.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 '+' 기호로 구분된 다음의 조합입니다.</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_TOTAL_SIZE_KB	BIGINT	total_pages*pagesize/1024로 계산된 테이블 스페이스의 전체 크기(KB)입니다.
TBSP_USABLE_SIZE_KB	BIGINT	usable_pages*pagesize/1024로 계산된 테이블 스페이스의 사용할 수 있는 전체 크기(KB)입니다.
TBSP_USED_SIZE_KB	BIGINT	used_pages*pagesize/1024로 계산된 테이블 스페이스의 사용된 전체 크기(KB)입니다.



표 195. TBSP\_UTILIZATION 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TBSP_FREE_SIZE_KB	BIGINT	free_pages*pagesize/1024로 계산된 테이블 스페이스의 전체 사용 가능 크기(KB)입니다.
TBSP_UTILIZATION_PERCENT	BIGINT	테이블 스페이스의 활용도를 백분율로 표시합니다. usable_pages를 사용할 수 있는 경우에는 (used_pages/usable_pages)*100으로 계산됩니다. 그 이외의 경우에는 -1이 표시됩니다.
TBSP_TOTAL_PAGES	BIGINT	tablespace_total_pages - 테이블 스페이스의 전체 페이지
TBSP_USABLE_PAGES	BIGINT	tablespace_usable_pages - 테이블 스페이스에서 사용할 수 있는 페이지
TBSP_USED_PAGES	BIGINT	tablespace_used_pages - 테이블 스페이스에서 사용된 페이지
TBSP_FREE_PAGES	BIGINT	tablespace_free_pages - 테이블 스페이스에서 사용 가능한 페이지
TBSP_PAGE_TOP	BIGINT	tablespace_page_top - 테이블 스페이스 상위 워터 마크(water mark)
TBSP_PAGE_SIZE	INTEGER	tablespace_page_size - 테이블 스페이스 페이지 크기
TBSP_EXTENT_SIZE	INTEGER	tablespace_extent_size - 테이블 스페이스 Extent 크기
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 테이블 스페이스 프리페치 크기
TBSP_MAX_SIZE	BIGINT	tablespace_max_size - 최대 테이블 스페이스 크기
TBSP_INCREASE_SIZE	BIGINT	tablespace_increase_size - 바이트 단위로 크기 늘리기
TBSP_INCREASE_SIZE_PERCENT	SMALLINT	tablespace_increase_size_percent - 퍼센트 단위로 크기 늘리기
TBSP_LAST_RESIZE_TIME	TIMESTAMP	tablespace_last_resize_time - 마지막으로 성공한 크기 조정 시간
TBSP_LAST_RESIZE_FAILED	SMALLINT	tablespace_last_resize_failed - 마지막으로 실패한 크기 조정 시도
TBSP_USING_AUTO_STORAGE	SMALLINT	tablespace_using_auto_storage - 테이블 스페이스를 자동 스토리지에 사용 가능
TBSP_AUTO_RESIZE_ENABLED	SMALLINT	tablespace_auto_resize_enabled - 테이블 스페이스 자동 크기 조정 가능
DBPGNAME	VARCHAR(128)	테이블 스페이스에 대한 데이터베이스 파티션 그룹의 이름입니다.
TBSP_NUM_CONTAINERS	BIGINT	tablespace_num_containers - 테이블 스페이스의 컨테이너 수
REMARKS	VARCHAR(254)	사용자가 입력한 주석입니다.

표 195. TBSP\_UTILIZATION 관리 뷰에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## TOP\_DYNAMIC\_SQL 관리 뷰 - 상위 등적 SQL문에 대한 정보 검색

TOP\_DYNAMIC\_SQL 관리 뷰에서는 실행 횟수, 평균 실행시간, 정렬 횟수 또는 명령문당 정렬 횟수별로 정렬 가능한 상위 등적 SQL문을 리턴합니다. 이들은 명령문이 잘 조정되었는지 확인하는 데 초점을 맞추어야 하는 쿼리입니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- TOP\_DYNAMIC\_SQL 관리 뷰에 대한 SELECT 특권
- TOP\_DYNAMIC\_SQL 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 스냅샷 모니터 데이터에 액세스하려면 다음 권한 중 하나도 필요합니다.

- SYSMON
- SYSCTRL
- SYSMOINT
- SYSADM

예 :

가장 자주 실행되는 상위 5개 SQL문을 식별합니다.

```
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
```

다음은 이 쿼리에 대한 출력의 예입니다.

NUM_EXECUTIONS	AVERAGE_EXECUTION_TIME_S	STMT_SORTS	...
148	0	0	...
123	0	0	...
2	0	0	...
1	0	0	...
1	0	0	...

5개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```

... SORTS_PER_EXECUTION ...
... ----- ...
...                0 ...
...                0 ...
...                0 ...
...                0 ...
...                0 ...

```

이 쿼리의 출력(계속).

```

... 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

```

## 리턴되는 정보

표 196. TOP\_DYNAMIC\_SQL 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	보고서 시간소인
NUM_EXECUTIONS	BIGINT	num_compilations - 명령문 컴파일
AVERAGE_EXECUTION_TIME_S	BIGINT	평균 실행시간
STMT_SORTS	BIGINT	stmt_sorts - 명령문 정렬 수
SORTS_PER_EXECUTION	BIGINT	명령문 실행당 정렬 횟수
STMT_TEXT	CLOB(2 M)	stmt_text - SQL문 텍스트
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.



---

## 제 15 장 SQL 프로시저 루틴

---

### ALTER\_ROUTINE\_PACKAGE 프로시저

▶—ALTER\_ROUTINE\_PACKAGE—(—*type*—,—*schema*—,—*module*—,—*name*—,—*options*—)—▶

스키마는 SYSPROC입니다.

이 프로시저는 리바인딩할 필요없이 컴파일된 SQL 루틴 또는 컴파일된 트리거와 연관된 패키지의 값을 변경합니다. 패키지 이름 대신 오브젝트 이름을 인수로 갖는다는 점을 제외하고는 ALTER PACKAGE 명령과 기능적으로 동일합니다. ALTER\_ROUTINE\_PACKAGE 프로시저는 명령행에서 호출하거나 응용프로그램에서 호출할 수 있습니다.

#### *type*

다음 값 중 하나를 사용하여 루틴 유형이나 컴파일된 트리거를 지정하는 CHAR(2) 유형의 입력 인수.

- 프로시저의 'P'
- 프로시저의 특정 이름의 'SP'
- 컴파일된 함수의 'F'
- 컴파일된 함수의 특정 이름의 'SF'
- 컴파일된 트리거의 'T'

#### *schema*

루틴 또는 트리거의 스키마를 지정하는 VARCHAR(128) 유형의 선택적 입력 인수. 스키마를 지정하지 않은 경우, 값은 CURRENT SCHEMA 특수 레지스터의 값이 디폴트값이 됩니다. 이 매개변수는 대소문자가 구분됩니다.

#### *module*

루틴이 있는 모듈의 이름을 지정하는 VARCHAR(128) 유형의 선택적 입력 인수. 트리거에 대해서는 이 매개변수를 지정할 수 없습니다. 이 매개변수가 지정되지 않으면, 모듈 루틴이 무시됩니다. 이 매개변수는 대소문자가 구분됩니다.

#### *name*

루틴 또는 트리거의 스키마를 지정하는 VARCHAR(128) 유형의 입력 인수. 이 매개변수는 대소문자가 구분됩니다.

*options*

ALTER PACKAGE문에서 지원되는 옵션 목록을 지정하는 VARCHAR(1024) 유형의 입력 인수. 적어도 하나의 ALTER PACKAGE절을 *options* 매개변수에 제공해야 합니다.

**예:**

기존 스토어드 프로시저의 밑줄 굿기 패키지를 UPDATE\_EMPLOYEE의 이름에 따라 변경합니다.

```
CALL SYSPROC.ALTER_ROUTINE_PACKAGE ('P','','','UPDATE_EMPLOYEE',
    'ACCESS PLAN REUSE YES OPTIMIZATION PROFILE AYYANG.INDEXHINTS')
```

컴파일된 트리거 MIN\_SALARY의 패키지를 DRICARD 스키마로 변경합니다.

```
CALL SYSPROC.ALTER_ROUTINE_PACKAGE ('T','DRICARD','','MIN_SALARY',
    'OPTIMIZATION PROFILE AYYANG.INDEXHINTS')
```

세 파트 이름을 사용하여 컴파일된 함수의 패키지를 변경합니다.

```
CALL SYSPROC.ALTER_ROUTINE_PACKAGE ('F','DRICARD','MODULE','FUNCTION','APREUSE YES')
```

---

## GET\_ROUTINE\_OPTS

▶▶ GET\_ROUTINE\_OPTS(—) —————▶▶

스키마는 SYSPROC입니다.

GET\_ROUTINE\_OPTS 함수는 현재 세션에서 SQL 프로시저 작성에 사용될 옵션의 문자 스트링 값을 리턴합니다.

함수의 결과는 길이 속성이 1024인 기변 길이 문자열(VARCHAR) 값입니다.

**예:**

쿼리 결과로 SQL 프로시저 작성에 사용될 옵션을 리턴하십시오.

```
SELECT GET_ROUTINE_OPTS()
FROM SYSIBM.SYSDUMMY1
```

---

## GET\_ROUTINE\_SAR

▶▶ GET\_ROUTINE\_SAR —————▶▶

▶(—sarblob—,—type—,—routine-name-string— [,—hide-body-flag—]) —————▶▶

스키마는 SYSFUN입니다.

GET\_ROUTINE\_SAR 프로시저는 동일한 운영 체제의 동일한 레벨을 실행 중인 다른 데이터베이스 서버에 동일한 루틴을 설치하기 위해 필요한 정보를 검색합니다. 이 정보는 SQL 아카이브 파일을 표현하는 단일 BLOB 문자열에 검색됩니다.

## 권한 부여

권한 부여 ID에 의해 보유된 특권에는 다음과 같은 권한 또는 특권 중 하나 이상을 포함해야 합니다.

- DBADM
- SQLADM
- DATAACCESS
- EXECUTE

### *sarblob*

루틴 SAR 파일 내용을 포함하는 유형 BLOB(3M)의 출력 인수

### *type*

다음 값 중 하나를 사용하여 루틴의 유형을 지정하는 유형 CHAR(2)의 입력 인수

- 프로시저의 'P'
- 프로시저의 특정 이름의 'SP'

### *routine-name-string*

루틴의 규정된 이름을 지정하는 유형 VARCHAR(257)의 입력 인수. 스키마 이름이 지정되지 않으면 루틴이 처리될 때 디폴트값은 CURRENT SCHEMA입니다.

*routine-name-string*은 큰따옴표(")를 포함할 수 없습니다.

### *hide-body-flag*

루틴 텍스트가 카탈로그에서 추출될 때 루틴 부분이 숨겨져야 하는지 여부를 지정하는(다음 값 중 하나를 사용하여) 유형 INTEGER의 입력 인수. 가능한 값은 다음과 같습니다.

0 루틴 텍스트를 그대로 둡니다. 이것은 디폴트값입니다.

1 루틴 텍스트가 카탈로그에서 추출될 때 루틴 부분을 빈 부분으로 대체합니다.

루틴의 규정된 이름은 검색할 루틴을 판별하는 데 사용됩니다. 발견되는 루틴은 SQL 루틴이어야 합니다. 특정 이름을 사용하지 않으면 루틴이 둘 이상이 될 수 있으며 오류가 발생합니다(SQLSTATE 42725). 오류가 리턴되면 특정 이름의 원하는 루틴이 사용되어야 합니다.

SAR 파일은 서버에서 사용 가능하지 않을 수도 있는 바인드 파일을 포함해야 합니다. 바인드 파일을 찾아서 SAR 파일에 저장할 수 없으면 오류가 발생합니다(SQLSTATE 55045).

---

## PUT\_ROUTINE\_SAR

PUT\_ROUTINE\_SAR 프로시저는 필요한 파일을 전달하여 서버에 SQL 루틴을 작성한 다음 루틴을 정의합니다.

### 권한 부여

권한 부여 ID에 의해 보유된 특권에는 다음과 같은 권한 또는 특권 중 하나 이상을 포함해야 합니다.

- DBADM
- SQLADM
- DATAACCESS
- EXECUTE

```
▶▶ PUT_ROUTINE_SAR ( ( sarblob [ , new-owner [ , use-register-flag ] ] ) ) ▶▶
```

스키마는 SYSFUN입니다.

#### *sarblob*

루틴 SAR 파일 콘텐츠를 포함한 BLOB(3M) 유형의 입력 인수.

#### *new-owner*

루틴을 점검하는 권한 부여로 사용되는 권한 부여 이름을 포함한 VARCHAR(128) 유형의 입력 인수. *new-owner*는 정의할 루틴에 필요한 특권이 있어야 합니다. *new-owner*가 지정되지 않으면 원래 루틴 정의자의 권한 부여 이름이 사용됩니다.

#### *use-register-flag*

CURRENT SCHEMA 및 CURRENT PATH 특수 레지스터가 루틴을 정의하는 데 사용되는지 여부를 나타내는 INTEGER 유형의 입력 인수. 특수 레지스터가 사용되지 않은 경우 디폴트 스키마 및 SQL 경로에 대한 설정은 원래 루틴이 정의되었을 때 사용된 설정입니다. *use-register-flag*에 대한 가능한 값은 다음과 같습니다.

0 현재 환경의 특수 레지스터를 사용하지 마십시오.

1 CURRENT SCHEMA 및 CURRENT PATH 특수 레지스터를 사용하십시오.

값이 1이면 CURRENT SCHEMA는 루틴 정의(루틴 이름 포함)에서 규정되지 않은 오브젝트 이름으로 사용되며 CURRENT PATH는 루틴 정의에서 규정되지 않



은 루틴과 데이터 유형을 분석하는 데 사용됩니다. *use-registers-flag*가 지정되지 않은 경우 동작은 값 0이 지정되었던 것과 같습니다.

*sarblob*에 포함된 ID 정보는 입력이 환경에 적당한지 확인하기 위해 점검되며, 그렇지 않은 경우 오류가 발생합니다(SQLSTATE 55046). PUT\_ROUTINE\_SAR 프로시저는 *sarblob*의 콘텐츠를 사용하여 서버에서 루틴을 정의합니다.

*sarblob* 인수의 콘텐츠는 SQL 아카이브 파일을 구성하는 별도의 파일로 추출됩니다. 공유 라이브러리와 바인드 파일은 임시 디렉토리의 파일에 작성됩니다. 처리되는 루틴 정의 명령문이 컴파일과 링크가 필요하지 않으며 공유 라이브러리와 바인드 파일의 위치가 사용 가능함을 인식하도록 환경이 설정됩니다. DDL 파일의 콘텐츠는 루틴 정의 명령문을 동적으로 실행하기 위해 사용됩니다.

단 하나의 프로시저만이 주어진 스키마 아래에 동시에 설치될 수 있습니다.

이 명령문을 처리하면 다른 인터페이스를 사용하여 루틴 정의 명령문을 실행할 때와 동일한 오류가 발생할 수 있습니다. 루틴 정의 처리 시 공유 라이브러리와 바인드 파일의 존재가 표시되고 프리컴파일, 컴파일 및 링크 단계를 건너뛩니다. 바인드 처리 시 바인드 파일이 사용되며 두 파일의 콘텐츠는 SQL 루틴의 일반 디렉토리로 복사됩니다.

GET ROUTINE 또는 PUT ROUTINE 조작(또는 해당 프로시저)을 실행할 수 없는 경우 실패 원인에 대한 정보를 제공하는 진단 텍스트와 함께 항상 오류를 리턴합니다(SQLSTATE 38000). 예를 들어 GET ROUTINE에 제공된 프로시저 이름이 SQL 프로시저를 식별하지 않은 경우, 진단 "-204, 42704" 텍스트가 리턴됩니다. 여기서 "-204" 및 "42704"는 각각 문제점의 원인을 식별하는 SQLCODE 및 SQLSTATE입니다. 이 예의 SQLCODE와 SQLSTATE는 GET ROUTINE 명령에 제공된 프로시저 이름이 정의되지 않음을 나타냅니다.

---

## REBIND\_ROUTINE\_PACKAGE 프로시저 - 패키지 리바인드

REBIND\_ROUTINE\_PACKAGE 프로시저는 SQL 프로시저, 루틴, 컴파일된 함수 또는 트리거와 연관된 패키지를 리바인드합니다. 패키지 이름 대신 프로시저 이름을 인수로 갖는다는 점을 제외하고는 REBIND 명령과 기능적으로 동일합니다. REBIND\_ROUTINE\_PACKAGE 프로시저는 명령행에서 호출되거나 응용프로그램에서 호출될 수 있습니다.

### 구문

유효성이 동등한 두 개의 메소드로 REBIND\_ROUTINE\_PACKAGE를 호출할 수 있습니다. 두 호출 간의 유일한 차이는 루틴 이름을 지정하는 방법입니다. 첫 번째 인스턴스에서 *routine-name-string* 변수는 마침표로 구분된 ID 이름으로 구성됩니다. 두 번째 메소드에서 루틴은 *schema*, *module* 및 *name* 값에 대한 별개의 값으로 식별됩니다.

메소드 1:

►►—REBIND\_ROUTINE\_PACKAGE—(—type—,—routine-name-string—,—options—)——►◄

메소드 2:

►►—REBIND\_ROUTINE\_PACKAGE—(—type—,——————►

►—schema—,—module—,—name—,—options—)——►◄

스키마는 SYSPROC입니다.

## 프로시저 매개변수

*type*

다음 값 중 하나를 사용하여 루틴 유형이나 컴파일된 트리거를 지정하는 CHAR(2) 유형의 입력 인수.

- 프로시저의 'P'
- 프로시저의 특정 이름의 'SP'
- 컴파일된 함수의 'F'
- 컴파일된 함수의 특정 이름의 'SF'
- 컴파일된 트리거의 'T'

*routine-name-string* (메소드 1만 해당)

루틴이나 트리거의 이름을 지정하는 VARCHAR(386) 유형의 입력 인수. 트리거 이름은 점으로 구분되는 두 파트로 구성되며 스키마가 선택적인 schema.trigger 형식입니다. 루틴 이름은 점으로 구분된 세 파트의 이름으로 구성되며 스키마와 모듈이 선택적인 schema.module.routine 형식입니다. 스키마를 지정하지 않은 경우 값은 CURRENT SCHEMA 특수 레지스터의 값이 디폴트값이 됩니다. 두 파트 이름이 지정되면 첫 번째 파트는 내부적으로 스키마 이름으로 해석됩니다. 해당 스키마에서 루틴을 찾지 못했다면 첫 번째 파트는 모듈 이름으로 해석되며 CURRENT SCHEMA에 있는 해당 이름의 모듈에서 루틴을 찾으려고 합니다. 스키마, 모듈 또는 오브젝트 이름은 큰따옴표(") 또는 마침표(.)를 포함할 수 없습니다.

*schema* (메소드 2만 해당)

루틴 또는 트리거의 스키마를 지정하는 VARCHAR(128) 유형의 선택적 입력 인수. 스키마를 지정하지 않은 경우, 값은 CURRENT SCHEMA 특수 레지스터의 값이 디폴트값이 됩니다. 이 매개변수는 대소문자가 구분됩니다.

*module* (메소드 2만 해당)

루틴이 있는 모듈의 이름을 지정하는 VARCHAR(128) 유형의 선택적 입력 인수. 트리거에 대해 이 매개변수를 지정하지 마십시오. 이 매개변수를 지정하지 않으면 모듈 루틴이 무시됩니다. 이 매개변수는 대소문자가 구분됩니다.

### *name* (메소드 2만 해당)

루틴 또는 트리거의 스키마를 지정하는 VARCHAR(128) 유형의 입력 인수. 이 매개변수는 대소문자가 구분됩니다.

### *options*

REBIND 명령 구문이 뒤따르는 리바인드 옵션 목록을 지정하는 VARCHAR(1024) 유형의 선택적 입력 인수. 『ANY』 또는 『CONSERVATIVE』의 단일 값도 이전 버전과의 호환성이 지원되며 RESOLVE 리바인드 옵션의 값으로 해석됩니다.

루틴의 규정된 이름은 검색할 루틴을 판별하는 데 사용됩니다. 찾은 루틴은 SQL 루틴이어야 합니다. 그렇지 않은 경우 오류가 발생합니다(SQLSTATE 428F7). 특정 이름이 사용되지 않으면 둘 이상의 루틴을 찾을 수 있으며 오류가 리턴됩니다(SQLSTATE 42725). 오류가 리턴되면 특정 이름의 원하는 루틴이 사용되어야 합니다.

### 예:

예 1: RESOLVE, REOPT 및 APREUSE 옵션을 사용하여 UPDATE\_EMPLOYEE 루틴의 패키지를 리바인드하십시오.

메소드 1:

```
CALL SYSPROC.REBIND_ROUTINE_PACKAGE  
( 'P', 'UPDATE_EMPLOYEE', 'RESOLVE ANY REOPT ONCE APREUSE YES' )
```

메소드 2:

```
CALL SYSPROC.REBIND_ROUTINE_PACKAGE  
( 'P', '', '', 'UPDATE_EMPLOYEE', 'RESOLVE ANY REOPT ONCE APREUSE YES' )
```

예 2: 옵션 없이 UPDATE\_EMPLOYEE 루틴의 패키지를 리바인드하십시오.

메소드 1:

```
CALL SYSPROC.REBIND_ROUTINE_PACKAGE ( 'P', 'UPDATE_EMPLOYEE', '' )
```

메소드 2:

```
CALL SYSPROC.REBIND_ROUTINE_PACKAGE ( 'P', '', '', 'UPDATE_EMPLOYEE', '' )
```

예 3: 컴파일된 트리거의 패키지를 리바인드하십시오.

메소드 1:

```
CALL SYSPROC.REBIND_ROUTINE_PACKAGE ( 'T', 'DRICARD.MIN_SALARY', 'REOPT ALWAYS' )
```

메소드 2:

```
CALL SYSPROC.REBIND_ROUTINE_PACKAGE ( 'T', 'DRICARD', '', 'MIN_SALARY', 'REOPT ALWAYS' )
```

예 4: 세 부분으로 구성된 이름을 사용하여 컴파일된 함수의 패키지를 리바인드하십시오.

메소드 1

```
CALL SYSPROC.REBIND_ROUTINE_PACKAGE ( 'F', 'DRICARD.MODULE.FUNCTION', 'REOPT ALWAYS' )
```

메소드 2

```
CALL SYSPROC.REBIND_ROUTINE_PACKAGE ( 'F', 'DRICARD', 'MODULE', 'FUNCTION', 'REOPT ALWAYS' )
```

---

## SET\_ROUTINE\_OPTS

▶▶—SET\_ROUTINE\_OPTS—(—*character-expression*—)————▶▶

스키마는 SYSPROC입니다.

SET\_ROUTINE\_OPTS 프로시저는 현재 세션에서 SQL 프로시저 작성에 사용될 옵션을 설정합니다. 해당 설정은 DB2\_SQLROUTINE\_PREPOPTS 레지스트리 변수에 지정된 인스턴스 전체 설정을 겹쳐쓰기합니다.

*character-expression*

유형 VARCHAR(1024)의 입력 인수는 현재 세션에 대한 옵션 설정을 지정합니다.

지정된 옵션은 세션 지속기간 동안 유효합니다. 널(NULL) 값이 인수로 지정된 경우 DB2\_SQLROUTINE\_PREPOPTS 레지스트리 변수 값은 현재 세션에 디폴트 옵션 설정으로 리스토어됩니다. 허용된 옵션 목록은 『쿼리 컴파일러 변수』의 DB2\_SQLROUTINE\_PREPOPTS 레지스트리 변수에 대한 설명을 참조하십시오.

---

## 제 16 장 순차적 재분배 루틴

---

### ANALYZE\_LOG\_SPACE 프로시저 - 로그 스페이스 분석 정보 검색

ANALYZE\_LOG\_SPACE 프로시저는 지정된 데이터베이스 파티션 그룹의 각 데이터베이스 파티션에 대한 로그 스페이스 분석 결과를 리턴합니다.

#### 구문

```
▶▶—ANALYZE_LOG_SPACE—(—inDBPGroup—,—inMainTbSchema—,—inMainTable—,—————▶  
▶—analysisType—,—inStmgTime—,—addDropOption—,—addDropList—,—pNumber—,—————▶  
▶—pWeight—)—————▶▶▶▶
```

스키마는 SYSPROC입니다.

#### 프로시저 매개변수

##### *inDBPGroup*

데이터베이스 파티션 그룹 이름을 지정하는 VARCHAR(128) 유형의 입력 인수.

##### *inMainTbSchema*

기본 테이블의 스키마를 지정하는 VARCHAR(128) 유형의 입력 인수.

##### *inMainTable*

데이터베이스 파티션 그룹의 기본 테이블(보통 데이터베이스 파티션 그룹에서 가장 큰 테이블)을 지정하는 VARCHAR(128) 유형의 입력 인수.

##### *analysisType*

분석 유형에 대한 표시기를 지정하는 SMALLINT 유형의 입력 인수.

- SWRD\_USE\_STMG\_TABLE (1): 데이터베이스 파티션 당 테이블 행 계수를 찾는 데 스토리지 관리 테이블에 있는 정보가 사용되는 것을 나타냅니다. 이 유형은 스토리지 관리 테이블이 설정되고 재분배될 데이터베이스 파티션 그룹에 대해 최소한 하나의 스토리지 스냅샷이 사용된 경우에만 사용해야 합니다.
- SWRD\_USE\_REALTIME\_ANALYSIS (2): 데이터베이스 파티션 당 테이블 행 계수를 찾는 데 SELECT 쿼리가 사용되는 것을 나타냅니다.

##### *inStmgTime*

스토리지 관리 레코드에 대한 시간소인을 지정하는 VARCHAR(26) 유형의 입력 인수. *analysisType*을 SWRD\_USE\_REALTIME\_ANALYSIS로 설정할 때 이 매개변수를 무시합니다.

### *addDropOption*

추가되거나 삭제되는 데이터베이스 파티션을 지정하는 CHAR(1) 유형의 입력 인수.

- 'A': 데이터베이스 파티션 추가.
- 'D': 데이터베이스 파티션 삭제.
- 'N': 추가하거나 삭제하지 않음.

### *addDropList*

추가되거나 삭제될 데이터베이스 파티션을 지정하는 VARCHAR(6000) 유형의 입력 인수. 이 데이터베이스 파티션 번호는 쉼표로 분리된 문자열 형식으로 지정되며 문자열에 공백을 사용할 수 없습니다.

### *pNumber*

데이터베이스 파티션 가중치에 해당하는 모든 데이터베이스 파티션 번호를 지정하는 VARCHAR(6000) 유형의 입력 인수. 각 데이터베이스 파티션 번호는 0 - 999이며 데이터베이스 파티션 번호는 공백이 없는 쉼표로 분리된 문자열 형식으로 지정됩니다.

### *pWeight*

*pNumber* 문자열에 있는 데이터베이스 파티션 번호에 해당하는 사용자가 지정한 모든 데이터베이스 파티션 가중치를 지정하는 VARCHAR(6000) 유형의 입력 인수. 각 데이터베이스 파티션 가중치는 0 - 32767이며, 데이터베이스 파티션 가중치는 문자열에 공백 없이 쉼표로 구분되는 문자열로 지정됩니다.

## 권한 부여

- SYSADM, SYSMON, SYSCTRL 또는 SYSMAINT
- ANALYZE\_LOG\_SPACE 프로시저에 대한 EXECUTE 특권

## 예 :

변경사항을 적용하지 않고 데이터베이스 파티션을 추가한 효과를 분석하십시오. 다음의 경우, 데이터베이스 파티션 그룹에 데이터베이스 파티션 40, 50 및 60을 추가하고 데이터베이스 파티션 10,20,30,40,50,60에 각각 목표 비율 1:2:1:2:1:2를 사용한다고 가정하십시오. 이 예에서 파티션 10, 20, 30만 실제로 데이터베이스 파티션 그룹에 존재한다는 것을 참고하십시오.

```
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')
```

변경사항을 적용하지 않고 데이터베이스 파티션을 삭제한 효과를 분석하십시오. 다음의 경우, 데이터베이스 파티션 그룹에서 데이터베이스 파티션 30을 삭제하고 1:1의 각 목표 비율을 사용하여 데이터베이스 파티션 10과 20에 있는 데이터를 재분배한다고 가정하십시오. 이 예에서 모든 데이터베이스 파티션 10, 20, 30이 데이터베이스 파티션 그룹에 존재해야 한다는 점을 참고하십시오.

```
CALL SYSPROC.ANALYZE_LOG_SPACE('IBMDEFAULTGROUP', 'TEST',
                                'EMP', 2, ' ', 'D', '30', '10,20','1,1')
```

## 사용 시 참고사항

해당 값을 확보할 수 없을 때 매개변수에 대한 출력 값으로 『-1』이 사용됩니다.

재분배 스토어드 프로시저 및 함수는 각 테이블에 대해 분산 키가 정의되어 있는 파티션된 데이터베이스 환경에서만 작업합니다.

## 리턴되는 정보

ANALYZE\_LOG\_SPACE 프로시저는 지정된 데이터베이스 파티션 그룹의 각 데이터베이스 파티션에 대해 다음 필드가 있는 로그 스페이스 분석 결과의 결과 세트(열린 커서)를 리턴합니다.

표 197. ANALYZE\_LOG\_SPACE 프로시저가 리턴한 정보

컬럼 이름	컬럼 유형	설명
PARTITION_NUM	SMALLINT	로그 스페이스 분석의 데이터베이스 파티션 번호
TOTAL_LOG_SIZE	BIGINT	할당된 총 로그 스페이스(바이트)로서 -1은 무제한 크기를 나타냅니다.
AVAIL_LOG_SPACE	BIGINT	사용 가능하며 재분배 프로세스에서 사용될 수 있는 로그 스페이스 크기(바이트)
DATA_SKEW	BIGINT	목표 레벨과 다른 데이터 크기의 절대값(바이트)
REQ_LOG_SPACE	BIGINT	원하는 데이터 재분배에 도달해야 하는 스페이스 크기(바이트)
NUM_OF_STEPS	SMALLINT	데이터 편차를 0으로 줄이는 데 필요한 단계 수
MAX_STEP_SIZE	BIGINT	로그 가득참 오류를 일으키지 않고 한 번에 이동될 수 있는 최대 데이터량(바이트)

## GENERATE\_DISTFILE 프로시저 - 데이터 분산 파일 생성

GENERATE\_DISTFILE 프로시저는 지정된 테이블에 대한 데이터 분산 파일을 생성하여 이를 지정된 fileName 아래에 저장합니다.

### 구문

```
►►—GENERATE_DISTFILE—(—inTbSchema—,—inTbName—,—fileName—)—————►
```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### inTbSchema

테이블 스키마 이름을 지정하는 VARCHAR(128) 유형의 입력 인수

*inTbName*

테이블 이름을 지정하는 VARCHAR(128) 유형의 입력 인수

*fileName*

데이터 분산 파일 이름을 지정하는 VARCHAR(255) 유형의 입력 또는 출력 인수. 지정된 파일 이름이 단지 파일 이름 뿐인 경우 파일은 해당 인스턴스 디렉토리 아래의 tmp 서브디렉토리에 저장되고 전체 파일 경로 이름은 매개변수에 리턴됩니다.

### 권한 부여

- GENERATE\_DISTFILE 프로시저에 대한 EXECUTE 특권
- SYSCAT.TABLES, SYSCAT.COLUMNS 및 지정된 테이블에 대한 SELECT 특권.

그 밖의 분리 사용자 ID는 인스턴스 디렉토리 아래의 tmp 서브디렉토리에서 파일을 작성할 수 있어야 합니다.

예 :

재분배 프로세스에서 사용될 데이터 분산 파일을 생성하십시오.

```
CALL SYSPROC.GENERATE_DISTFILE('TEST', 'EMP',  
    '$HOME/sql1lib/function/SAMPLE.IBMDEFAULTGROUP_swrData.dst')
```

### 사용 시 참고사항

재분배 스토어드 프로시저 및 함수는 각 테이블에 대해 분산 키가 정의되어 있는 파티션된 데이터베이스 환경에서만 작업합니다.

---

## GET\_SWRD\_SETTINGS 프로시저 - 재분배 정보 검색

GET\_SWRD\_SETTINGS 프로시저는 지정된 데이터베이스 파티션 그룹에 대한 기존 재분배 레지스트리 레코드를 읽습니다.

### 구문

```
▶▶ GET_SWRD_SETTINGS (—dbpgName—, —matchingSpec—, —redistMethod—, —————▶  
▶ —pMapFile—, —distFile—, —stepSize—, —totalSteps—, —stageSize—, —————▶  
▶ —nextStep—, —processState—, —pNumber—, —pWeight—) —————▶▶
```

스키마는 SYSPROC입니다.



## 프로시저 매개변수

### *dbpgName*

재분배 프로세스가 실행될 데이터베이스 파티션 그룹 이름을 지정하는 VARCHAR(128) 유형의 입력 인수.

### *matchingSpec*

표 198의 비트 방식 필드 ID를 지정하며 출력 매개변수가 리턴할 목표 필드를 나타내는 SMALLINT 유형의 입력 인수. 필요하지 않은 출력 매개변수는 널(NULL)로 설정될 수 있습니다.

예를 들어, *matchingSpec*을 (REDIST\_STAGE\_SIZE | REDIST\_NEXT\_STEP)의 정수 값인 96으로 설정할 경우, 이 함수의 호출자는 값을 수산하기 위해 *stageSize* 및 *nextStep*만 제공하면 되며 나머지 출력 매개변수는 널(NULL)이 될 수 있습니다.

표 198. 비트 방식 필드 ID

필드 이름	16진수 값	10진수 값
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*

재분배가 데이터 분산 파일을 사용하여 실행되는지 또는 목표 분산 맵을 사용하여 실행되는지를 지정하는 SMALLINT 유형의 출력 인수. 두 가지 가능한 리턴 값은 다음과 같습니다.

- 2: 재분배 프로세스가 데이터 분산 파일을 입력으로 사용하여 작업할 것을 나타냅니다.
- 3: 재분배 프로세스가 목표 분산 맵을 입력으로 사용하여 작업할 것을 나타냅니다.

### *pMapFile*

데이터베이스 서버에서 목표 분산 맵의 전체 경로 파일 이름을 지정하는 VARCHAR(255) 유형의 출력 인수

### *distFile*

데이터베이스 서버에서 데이터 분산 파일의 전체 경로 파일 이름을 지정하는 VARCHAR(255) 유형의 출력 인수

### *stepSize*

로그가 가득차는 상황을 방지하기 위해 커미트가 호출되어야 하기 전에 이동될 수 있는 최대 행 수를 지정하는 BIGINT 유형의 출력 인수. 이 숫자는 각 재분배 단계에서 변경될 수 있습니다.

### *totalSteps*

지정된 데이터베이스 파티션 그룹을 완전히 재분배하기 위해 수행하는 단계 수를 지정하는 SMALLINT 유형의 출력 인수

### *stageSize*

연속적으로 실행될 단계 수를 지정하는 SMALLINT 유형의 출력 인수

### *nextStep*

완료된 단계와 계속 실행되어야 할 단계를 분리시키는 인덱스를 지정하는 SMALLINT 유형의 출력 인수

### *processState*

재분배 프로세스가 다음 점검 지점에서 중지될 것인지 여부를 나타내는 SMALLINT 유형의 출력 인수. 점검 지점은 각 재분배 단계의 시작 부분에 있습니다. 이 인수가 1로 설정되었으면 단계가 시작되지 않고 해당 값이 0이면 단계가 진행됩니다.

### *pNumber*

쉽게로 분리된 데이터베이스 파티션 번호 목록을 문자열 형식으로 리턴할 수 있는 VARCHAR(6000) 유형의 출력 인수. 이들 파티션 번호는 현재 데이터베이스 파티션 그룹에서 사용되거나 추가 또는 삭제될 데이터베이스 파티션이 될 수 있습니다. 이들 파티션 번호의 시퀀스와 계수는 *pWeight* 변수가 리턴하는 목표 파티션 가중치와 일치합니다.

### *pWeight*

쉽게로 분리된 목표 데이터베이스 파티션 가중치 번호 목록을 리턴할 수 있는 VARCHAR(6000) 유형의 출력 인수. 이들 파티션 가중치의 시퀀스와 계수는 *pNumber* 변수가 리턴하는 파티션 번호와 일치합니다.

## 권한 부여

GET\_SWRD\_SETTINGS 프로시저에 대한 EXECUTE 특권

예 :

지정된 데이터베이스 파티션 그룹에 대한 순차적 재분배 플랜의 내용을 보고하십시오.

```
CALL SYSPROC.GET_SWRD_SETTINGS  
( 'IBMDEFAULTGROUP', 255, ?, ?, ?, ?, ?, ?, ?, ? )
```

## 사용법 참고

재분배 스토어드 프로시저 및 함수는 각 테이블에 대해 분산 키가 정의되어 있는 파티션된 데이터베이스 환경에서만 작업합니다.

### SET\_SWRD\_SETTINGS 프로시저 - 재분배 레지스트리 작성 또는 변경

SET\_SWRD\_SETTINGS 프로시저는 재분배 레지스트리를 작성하거나 변경합니다. 레지스트리가 존재하지 않을 경우 레지스트리를 새로 작성하여 레코드를 추가합니다. 레지스트리가 이미 존재할 경우 *overwriteSpec*을 사용하여 겹쳐쓸 필요가 있는 필드 값을 식별합니다. 이 함수는 *overwriteSpec* 필드를 통해 갱신할 필요가 없는 필드에 널(NULL) 입력값을 가져옵니다.

#### 구문

```
▶▶ SET_SWRD_SETTINGS (—dbpgName—, —overwriteSpec—, —redistMethod—, —————▶  
▶ —pMapFile—, —distFile—, —stepSize—, —totalSteps—, —stageSize—, —————▶  
▶ —nextStep—, —processState—, —pNumber—, —pWeight—) —————▶▶
```

스키마는 SYSPROC입니다.

#### 프로시저 매개변수

##### *dbpgName*

재분배 프로세스가 실행될 데이터베이스 파티션 그룹 이름을 지정하는 VARCHAR(128) 유형의 입력 인수.

##### *overwriteSpec*

재분배 설정값 레지스트리에 쓰여지거나 겹쳐쓰여질 목표 필드를 나타내는 표 199의 비트 방식 필드 ID.

표 199. 비트 방식 필드 ID

필드 이름	16진수 값	10진수 값
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*

재분배가 데이터 분산 파일을 사용하여 실행되는지 또는 목표 분산 맵을 사용하여 실행되는지를 지정하는 SMALLINT 유형의 입력 인수. 다음은 두 가지 유효한 입력 값입니다.

- 2: 재분배 프로세스가 데이터 분산 파일을 입력으로 사용하여 작업할 것을 나타냅니다.
- 3: 재분배 프로세스가 목표 분산 맵을 입력으로 사용하여 작업할 것을 나타냅니다.

### *pMapFile*

데이터베이스 서버에서 목표 분산 맵의 전체 경로 파일 이름을 지정하는 VARCHAR(255) 유형의 입력 인수.

### *distFile*

데이터베이스 서버에서 데이터 분산 파일의 전체 경로 파일 이름을 지정하는 VARCHAR(255) 유형의 입력 인수.

### *stepSize*

로그가 가득차는 상황을 방지하기 위해 커미트가 호출되어야 하기 전에 이동될 수 있는 최대 행 수를 지정하는 BIGINT 유형의 입력 인수. 이 숫자는 각 재분배 단계에서 변경될 수 있습니다. 숫자가 무제한인 것을 나타내기 위해 *stepSize*에 『-2』 값을 사용할 수 있습니다.

### *totalSteps*

지정된 데이터베이스 파티션 그룹을 완전히 재분배하기 위해 수행하는 단계 수를 지정하는 SMALLINT 유형의 입력 인수. 숫자가 무제한인 것을 나타내기 위해 *totalSteps*에 『-2』 값을 사용할 수 있습니다.

### *stageSize*

연속적으로 실행될 단계 수를 지정하는 SMALLINT 유형의 입력 인수.

### *nextStep*

완료된 단계와 계속 실행되어야 할 단계를 분리시키는 인덱스를 지정하는 SMALLINT 유형의 입력 인수.

### *processState*

재분배 프로세스가 다음 점검 지점에서 중지될 것인지 여부를 나타내는 SMALLINT 유형의 입력 인수. 점검 지점은 각 재분배 단계의 시작 부분에 있습니다. 이 인수가 1로 설정되었으면 단계가 시작되지 않고 해당 값이 0이면 단계가 진행됩니다.

### *pNumber*

범표로 분리된 데이터베이스 파티션 번호 목록을 문자열 형식으로 포함할 수 있는 VARCHAR(6000) 형식의 입력 인수. 이들 파티션 번호는 현재 데이터베이스 파티션 그룹에서 사용되거나 추가 또는 삭제될 데이터베이스 파티션이 될 수 있습니다.

이들 파티션 번호의 시퀀스와 계수는 *pWeight* 변수가 리턴하는 목표 파티션 가중치와 일치합니다. 각 데이터베이스 파티션 번호는 0 - 999이며, 이 문자열에는 공백을 사용할 수 없습니다.

#### *pWeight*

*pNumber* 문자열에 있는 데이터베이스 파티션 번호에 해당하는 사용자가 지정한 샘플로 분리된 문자열의 모든 데이터베이스 파티션 가중치를 포함할 수 있는 VARCHAR(6000) 유형의 입력 인수. 각 데이터베이스 파티션 가중치는 0 - 32767이며, 이 문자열에는 공백을 사용할 수 없습니다.

### 권한 부여

SET\_SWRD\_SETTINGS 프로시저에 대한 EXECUTE 특권

예 :

레지스트리에 순차적 재분배 플랜을 쓰십시오. *processState*를 1로 설정하면 현재 실행 중인 순차적 재분배 스토어드 프로시저가 현재 단계를 완료하고 이 매개변수가 0으로 재설정될 때까지 중지했다가 다시 호출되게 할 수 있습니다.

```
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')
```

### 사용 시 참고사항

재분배 스토어드 프로시저 및 함수는 각 테이블에 대해 분산 키가 정의되어 있는 파티션된 데이터베이스 환경에서만 작업합니다.

---

## STEPWISE\_REDISTRIBUTE\_DBPG 프로시저 - 데이터베이스 파티션 그룹의 일부 재분배

STEPWISE\_REDISTRIBUTE\_DBPG 프로시저는 이 프로시저에 지정된 입력 및 SET\_SWRD\_SETTINGS 프로시저에 의해 작성되거나 갱신된 설정 파일에 따라 데이터베이스 파티션 그룹의 일부를 재분배합니다.

### 구문

```
▶▶STEPWISE_REDISTRIBUTE_DBPG(—inDBPGroup—,—inStartingPoint—,—  
▶—inNumSteps—)▶▶
```

스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *inDBPGroup*

목표 데이터베이스 파티션 그룹의 이름을 지정하는 VARCHAR(128) 유형의 입력 인수.

### *inStartingPoint*

사용할 시작 지점을 지정하는 SMALLINT 유형의 입력 인수. 매개변수가 NULL이 아닌 양수로 설정될 경우, STEPWISE\_REDISTRIBUTE\_DBPG 프로시저는 설정 파일에 지정된 *nextStep* 값을 사용하는 대신 이 값을 사용합니다. 이 옵션은 STEPWISE\_REDISTRIBUTE\_DBPG 프로시저를 특정 단계부터 재실행할 때 유용합니다. 매개변수가 NULL로 설정될 경우, *nextStep* 값이 사용됩니다.

### *inNumSteps*

실행할 단계 수를 지정하는 SMALLINT 유형의 입력 인수. 매개변수가 NULL이 아닌 양수로 설정될 경우, STEPWISE\_REDISTRIBUTE\_DBPG 프로시저는 설정 파일에 지정된 *stageSize* 값을 사용하는 대신 이 값을 사용합니다. 이 옵션은 설정값에 지정된 것과 다른 단계 수로 STEPWISE\_REDISTRIBUTE\_DBPG 프로시저를 재실행할 때 유용합니다. 예를 들어, 스케줄된 단계에 5개의 단계가 있으며 재분배 프로세스가 3단계에서 실패한 경우, 오류 조건을 정정한 후 나머지 세 단계를 실행하도록 STEPWISE\_REDISTRIBUTE\_DBPG 프로시저를 호출할 수 있습니다. 매개변수가 NULL로 설정될 경우, *stageSize* 값이 사용됩니다. 숫자가 무제한인 것을 나타내기 위해 이 프로시저에서 -2값을 사용할 수 있습니다.

주: REDISTRIBUTE DATABASE PARTITION GROUP 명령의 **NOT ROLLFORWARD RECOVERABLE** 옵션과 동등한 옵션을 지정할 매개변수가 없습니다. 로깅은 STEPWISE\_REDISTRIBUTE\_DBPG 프로시저를 사용할 때 수행되는 행 데이터 재분배에 항상 수행됩니다.

## 권한 부여

- STEPWISE\_REDISTRIBUTE\_DBPG 프로시저에 대한 EXECUTE 특권
- SYSADM, SYSCTRL 또는 DBADM

## 예 :

SET\_SWRD\_SETTINGS 프로시저에 의해 레지스트리에 저장된 재분배 플랜에 따라 데이터베이스 파티션 그룹 "IBMDEFAULTGROUP"을 재분배하십시오. 3단계에서 시작하며 재분배 계획에서 두 단계가 완료될 때까지 데이터를 재분배합니다.

```
CALL SYSPROC.STEPWISE_REDISTRIBUTE_DBPG('IBMDEFAULTGROUP', 3, 2)
```

## 사용 시 참고사항

STEPWISE\_REDISTRIBUTE\_DBPG 프로시저 실행이 시작된 후 SET\_SWRD\_SETTINGS 프로시저를 사용하여 *processState*에 대한 레지스트리 값을 1로 갱신한 경우, 이 프로세스는 다음 단계가 시작될 때 중지되고 경고 메시지가 리턴됩니다.

재분배 프로세스가 SQL COMMIT문을 호출했기 때문에 유형 2 연결에서는 재분배 프로세스를 실행할 수 없습니다.





## 제 17 장 스토리지 관리 도구 루틴

### CAPTURE\_STORAGEMGMT\_INFO 프로시저 - 제공된 루트 오브젝트에 대한 스토리지 관련 정보 검색

CAPTURE\_STORAGEMGMT\_INFO 프로시저는 해당 범위에 정의된 스토리지 오브젝트 및 제공된 루트 오브젝트에 대한 스토리지 관련 정보 수집을 시도합니다. 모든 스토리지 오브젝트는 SYSTOOLS.STMG\_OBJECT\_TYPE 테이블에 지정되어 있습니다.

표 200. STMG\_OBJECT\_TYPE 테이블

컬럼 이름	데이터 유형	널 값 가능	설명
OBJ_TYPE	INTEGER	N	정수 값은 스토리지 오브젝트의 유형에 해당합니다. <ul style="list-style-type: none"> <li>• 0 - 데이터베이스</li> <li>• 1 - 데이터베이스 파티션 그룹</li> <li>• 2 - 테이블 스페이스</li> <li>• 3 - 테이블 스페이스 컨테이너</li> <li>• 4 - 테이블</li> <li>• 5 - 인덱스</li> </ul>
TYPE_NAME	VARCHAR	N	스토리지 오브젝트 유형의 설명적 이름 <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>

#### 구문

```

▶▶—CAPTURE_STORAGEMGMT_INFO—(—in_rootType—,—in_rootSchema—,——————▶
▶—in_rootName—)——————▶▶
    
```

스키마는 SYSPROC입니다.

#### 프로시저 매개변수

*in\_rootType*

SMALLINT 유형의 입력 인수입니다. 유효한 옵션 유형은 다음과 같습니다.

- 0 - 데이터베이스

- 1 - 데이터베이스 파티션 그룹
- 2 - 테이블 스페이스
- 4 - 테이블
- 5 - 인덱스

입력 인수는 널(NULL)일 수 없습니다. 널(NULL) 값을 지정하면 SQLSTATE 38553 및 DBA7617 토큰의 SQL0443 오류가 리턴됩니다.

#### *in\_rootSchema*

스토리지 스냅샷 루트 오브젝트의 스키마 이름을 지정하는 VARCHAR(128) 유형의 입력 인수.

#### *in\_rootName*

루트 오브젝트의 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. 입력 인수는 널(NULL)일 수 없습니다. 널(NULL) 값을 지정하면 SQLSTATE 38553 및 DBA7617 토큰의 SQL0443 오류가 리턴됩니다.

### 권한 부여

- CAPTURE\_STORAGEMGMT\_INFO 프로시저에 대한 EXECUTE 특권.
- SYSPROC.DB\_PARTITIONS, SYSPROC.SNAP\_GET\_CONTAINER, SYSPROC.SNAPSHOT\_CNTRFS 테이블 함수에 대한 EXECUTE 특권.
- SYSCAT.TABLES, SYSCAT.TABLESPACES, SYSCAT.NODEGROUPDEF, SYSCAT.DATABASEPARTITIONS, SYSCAT.DATAPARTITIONEXPRESSION, SYSCAT.INDEXES 및 SYSCAT.COLUMNS에 대한 SELECT 특권.

---

## CREATE\_STORAGEMGMT\_TABLES 프로시저 - 스토리지 관리 테이블 작성

CREATE\_STORAGEMGMT\_TABLES 프로시저는 입력에서 지정된 테이블 스페이스의 고정 "DB2TOOLS" 스키마에 모든 스토리지 관리 테이블을 작성합니다.

### 구문

```
▶▶—CREATE_STORAGEMGMT_TABLES—(—in_tbspace—)—————▶▶
```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *in\_tbspace*

테이블 스페이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. 입력 인수는 널(null)이 될 수 없습니다. 널(null) 값이 지정되면 SQLSTATE 38553를 포함한 SQL0443 오류 및 토큰 DBA7617이 리턴됩니다.

## 권한 부여

CREATE\_STORAGEMGMT\_TABLES 프로시저의 EXECUTE 특권.

데이터베이스에 대한 CREATETAB 특권과 테이블 스페이스에 대한 USE 특권 및 다음 중 하나의 특권도 있어야 합니다.

- 내재된 또는 명시적 스키마 이름 DB2TOOLS가 존재하지 않을 경우 데이터베이스에 대한 IMPLICIT\_SCHEMA 권한.
- 테이블의 스키마 이름이 존재하는 경우 스키마에 대한 CREATEIN 특권.
- DBADM 권한

## 사용 시 참고사항

다음 테이블이 DB2TOOLS 스키마에 작성됩니다.

- 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 프로시저 - 전체 스토리지 관리 테이블 삭제(drop)

DROP\_STORAGEMGMT\_TABLES 프로시저에서 전체 스토리지 관리 테이블 삭제(drop)를 시도합니다.

### 구문

►►—DROP\_STORAGEMGMT\_TABLES—(—dropSpec—)◄◄

스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *dropSpec*

SMALLINT 유형의 입력 인수입니다. *dropSpec*이 0으로 설정된 경우 오류가 발생하면 프로세스가 중지됩니다. *dropSpec*이 1로 설정된 경우 발생한 모든 오류를 무시하고 프로세스가 계속됩니다. 입력 인수는 널(NULL)일 수 없습니다. 널(NULL) 값을 지정하면 SQLSTATE 38553 및 DBA7617 토큰의 SQL0443 오류가 리턴됩니다.

## 권한 부여

DROP\_STORAGEMGMT\_TABLES 프로시저의 EXECUTE 특권입니다.

데이터베이스를 연결한 사용자 ID는 SYSCAT.TABLES의 DEFINER 컬럼에 기록된 스토리지 관리 테이블의 정의자이거나 다음 특권 중 하나를 가지고 있어야 합니다.

- DBADM 권한
- 이 테이블에 대한 스키마의 DROPIN 특권
- 이 테이블의 CONTROL 특권

---

## 제 18 장 텍스트 검색 루틴

---

### SYSTS\_ADMIN\_CMD 스토어드 프로시저 - 텍스트 검색 관리 명령 실행

응용프로그램은 SYSTS\_ADMIN\_CMD 프로시저를 사용하여 SQL CALL문을 통해 텍스트 검색 관리 명령을 실행합니다.

#### 구문

```
▶▶SYSTS_ADMIN_CMD(—command-string—,—message-locale—,—message—)▶▶
```

스키마는 SYSPROC입니다.

#### 프로시저 매개변수

##### *command-string*

실행되는 단일 텍스트 검색 인덱스 관리 명령을 지정하는 VARCHAR (32K) 유형의 입력 인수입니다. 명령 구문은 연결 옵션을 제외하고는 DB2 텍스트 검색 명령과 동일합니다. 이 프로시저에서는 연결 옵션이 지원되지 않습니다. 이 프로시저를 통해 실행되는 명령은 현재 연결을 사용합니다.

##### *message-locale*

리턴된 모든 오류 메시지에 대해 원하는 언어를 지정하는 VARCHAR (33) 유형의 입력 인수입니다. 인수가 널(NULL)이거나 비어 있는 문자열인 경우 또는 지정된 로케일에 대한 메시지 파일을 서버에서 사용할 수 없는 경우에는 'en\_US'가 사용됩니다.

##### *message*

제대로 실행된 것으로 간주되는 조작에 대한 경고 또는 정보 메시지를 지정하는 VARCHAR (32K) 유형의 출력 인수입니다.

#### 권한 부여

SYSTS\_ADMIN\_CMD 프로시저의 EXECUTE 특권입니다.

프로시저는 현재 다음 DB2 텍스트 검색 명령을 지원합니다.

- ALTER INDEX
- CLEAR COMMAND LOCKS
- CLEAR EVENTS
- CREATE INDEX
- DISABLE DATABASE

- DROP INDEX
- ENABLE DATABASE
- UPDATE INDEX

예 :

스키마 DB2TS에서 텍스트 검색 인덱스 MYTEXTINDEX를 갱신하고 모든 오류 메시지를 영어로 리턴합니다.

```
CALL SYSPROC.SYSTS_ADMIN_CMD
('UPDATE INDEX DB2TS.MYTEXTINDEX FOR TEXT','en_US', ?)";
```

다음은 이 쿼리의 출력 예입니다.

출력 매개변수 값

```
-----
매개변수 이름   : MESSAGE
매개변수 값     : CIE00001 조작이 완료되었습니다.
```

리턴 상태 = 0

## 사용 시 참고사항

- 명령 실행이 실패한 경우 텍스트 검색에만 적용되는 오류 메시지가 포함된 채로 SQLCODE -20427 및 SQLSTATE 38H14가 리턴됩니다. 예를 들어 인덱스 MYTEXTINDEX가 이미 있는 경우 다음 명령문이 실행됩니다.

```
CALL SYSPROC.SYSTS_ADMIN_CMD ('CREATE INDEX MYTEXTINDEX FOR TEXT
ON DB2TS.TEXTBOOKS (STORY)', 'en_US', ?)
```

인덱스 작성은 실패하며 다음 오류 메시지가 발생합니다.

```
SQL20427N 텍스트 검색 관리 프로시저 또는 명령 중
오류가 발생했습니다. 오류 메시지는 "CIE00201 텍스트 검색
인덱스 "DB2TS ". "MYTEXTINDEX"가 이미 존재합니다.". SQLSTATE=38H14
```

- SQLCODE가 프로시저에서 리턴된 경우 메시지는 절단될 수도 있습니다. 전체 메시지는 db2diag 로그 파일에 있습니다.

---

## SYSTS\_ALTER 프로시저 - 인덱스의 갱신 특성 변경

이 프로시저는 인덱스의 갱신 특성을 변경합니다.

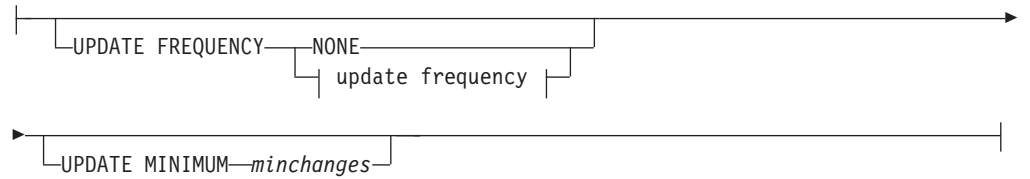
이 프로시저는 데이터베이스 서버에서 ALTER INDEX 텍스트 검색 관리 명령을 실행합니다.

### 구문

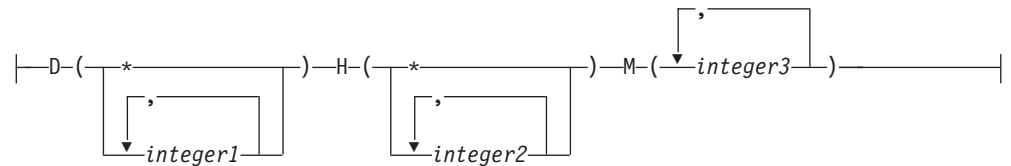
```
►►—SYSTS_ALTER—(—index_schema—,—index_name—,—| update characteristics |—►►
```

▶,—message\_locale—,—message—)—————▶

갱신 특성:



갱신 빈도:



스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *index\_schema*

텍스트 검색 인덱스의 스키마를 지정하는 유형 VARCHAR(128)의 입력 인수입니다. *index\_schema*는 DB2 스키마 이름에 대한 제한 이름 지정을 따라야 합니다. 인수가 널(NULL)이거나 비어 있는 문자열인 경우, CURRENT SCHEMA의 값이 사용됩니다. *index\_schema*는 대소문자를 구분합니다.

### *index\_name*

인덱스의 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. *index\_schema*와 함께, 데이터베이스에서 텍스트 검색 인덱스를 고유하게 식별합니다. *index\_name*은 대소문자를 구분합니다.

갱신 특성

변경 옵션을 지정하는 VARCHAR(32K) 유형의 입력 인수. 허용된 변경 옵션은 다음과 같습니다.

### UPDATE FREQUENCY

인덱스 갱신 빈도를 지정합니다. 인덱스는 변경 수가 UPDATE MINIMUM에 설정된 값 이상인 경우 갱신됩니다. 갱신 빈도 NONE은 인덱스가 더 이상 갱신되지 않음을 나타냅니다. 이 옵션은 변경되지 않는 데이터가 있는 테이블의 텍스트 컬럼에 유용합니다. 또한 사용자가 UPDATE INDEX 명령을 사용하여 인덱스를 수동으로 갱신하려는 경우에도 유용합니다. 자동 갱신은 START FOR TEXT 명령이 실행되고 DB2 텍스트 검색 인스턴스 서비스가 실행 중인 경우에만 수행될 수 있습니다.

디폴트 빈도 값은 SYSIBMTS.TSDEFAULTS 뷰  
(DEFAULTNAME='UPDATEFREQUENCY')에서 가져옵니다.

#### **NONE**

텍스트 인덱스에 자동 갱신이 적용되지 않습니다. 추가 인덱스 갱신은 수  
동으로 시작해야 합니다.

#### **D** 인덱스가 갱신된 요일.

\* 모든 요일.

*integer1*

일요일에서 토요일까지의 특정 요일(0 - 6)

#### **H** 인덱스가 갱신된 지정된 날짜의 시간.

\* 모든 시간.

*integer2*

자정에서 오후 11시까지 하루 중 특정 시간(0 - 23)

#### **M** 인덱스가 갱신된 지정된 시간의 분.

*integer3*

시간의 맨 위(0) 또는 해당 시간 이후 5분 간격으로 증가하는 시간(0,  
5, 10, 15, 20, 25, 30, 35, 40, 45, 50 또는 55)

UPDATE FREQUENCY 옵션을 지정하지 않으면 빈도 설정이 변경되지 않습  
니다.

#### **UPDATE MINIMUM** *minchanges*

인덱스가 점차적으로 갱신되기 전에 발생해야 하는 텍스트 문서에 대한 최소 변  
경 횟수를 지정합니다. 동일한 텍스트 문서를 여러 번 변경하는 경우 개별 변경  
사항으로 간주됩니다. UPDATE MINIMUM 옵션을 지정하지 않으면 설정이  
변경되지 않습니다.

#### *message\_locale*

리턴된 오류 메시지에 사용될 로케일을 지정하는 VARCHAR(33) 유형의 입력 인  
수. 인수가 널(null)이거나 빈 문자열이거나 지정된 로케일의 메시지 파일을 서버에  
서 사용할 수 없는 경우 'en\_US'가 사용됩니다.

#### *message*

완료된 조작에 대해 경고나 정보 메시지를 지정하는 VARCHAR(32K) 유형의 출  
력 인수.

### **권한 부여**

명령문의 권한 부여 ID는 적어도 다음과 같은 특권 중 하나를 가지고 있어야 합니다.

- 텍스트 인덱스가 정의되는 테이블에 대한 CONTROL 특권



- DBADM 권한

## 예:

예 1: 다음 예에서는 텍스트 검색 인덱스의 갱신 특성이 변경 중입니다. 이 인덱스는 원래 *index\_schema* 'db2ts' 및 *index\_name* 'myTextIndex'를 사용하여 작성되었습니다. 'UPDATE FREQUENCY NONE'을 사용하는 목적은 연관된 테이블 컬럼에 변경이 예측되지 않도록 텍스트 검색 인덱스에 더 이상의 갱신을 하지 않는 것입니다. 리턴될 오류 메시지가 영어로 요청되었습니다. 프로시저가 완료되면 출력 매개변수 메시지가 완료된 조작이 호출자에게 리턴되었음을 표시합니다.

```
CALL SYSPROC.SYSTS_ALTER('db2ts', 'myTextIndex',
    'UPDATE FREQUENCY NONE', 'en_US', ?)
```

다음은 이 쿼리의 출력 예입니다.

```
출력 매개변수 값
-----
매개변수 이름   : MESSAGE
매개변수 값     : 조작이 완료되었습니다.

리턴 상태 = 0
```

예 2: 다음 예에서, SYSTS\_ALTER는 *index\_schema* 'db2ts' 및 *index\_name* 'myTextIndex'가 있는 텍스트 검색 인덱스에 대한 갱신 특성을 변경하기 위해 호출됩니다. 이것은 인덱스에 대한 갱신이 설정된 매 시간 마다 발생하도록 하기 위한 것입니다. 그러나 이 인덱스가 존재하지 않으므로 오류가 발생했습니다.

```
CALL SYSPROC.SYSTS_ALTER('db2ts', 'myTextIndex',
    'update frequency D(*) H(*) M(0)', 'en_US', ?)
```

이 쿼리의 출력 예는 다음과 같습니다.

```
SQL20427N 텍스트 검색 관리 프로시저 또는 명령 중
오류가 발생했습니다. 오류 메시지는 다음과 같습니다. "CIE00316 텍스트 검색
인덱스 "db2ts"."myTextIndex"가 존재하지 않습니다.".
SQLSTATE 38H14
```

## 사용 시 참고사항

- 텍스트 검색 관리 프로시저는 데이터베이스에 대한 기존 연결을 사용합니다. 프로시저 완료에 따라 현재 트랜잭션이 커밋되거나 롤백될 수 있습니다. 모든 트랜잭션 변경을 커밋하여 이러한 커밋나 롤백으로부터의 예기치 않은 영향을 피하려고 할 수 있습니다. AUTOCOMMIT를 켜는 것이 유일한 방법입니다.
- 충돌 가능성이 있는 경우에는 텍스트 검색 인덱스에서 동시에 여러 프로시저 또는 명령을 실행할 수 없습니다. 충돌 프로시저와 명령의 예는 다음과 같습니다.
  - SYSTS\_ALTER 프로시저 또는 ALTER INDEX db2ts 명령

- SYSTS\_CLEAR\_EVENTS 프로시저 또는 CLEAR EVENTS FOR INDEX db2ts 명령
- SYSTS\_DISABLE 프로시저 또는 DISABLE DATABASE FOR TEXT db2ts 명령
- SYSTS\_DROP 프로시저 또는 DROP INDEX db2ts 명령
- STOP FOR TEXT db2ts 명령
- SYSTS\_UPDATE 프로시저 또는 UPDATE INDEX db2ts 명령

충돌이 있는 경우 프로시저는 SQLCODE -20426 및 SQLSTATE 38H13을 리턴합니다.

- 이 프로시저를 실행할 경우,
  - DB2 텍스트 검색 뷰 SYSIBMTS.TSLOCKS의 콘텐츠가 갱신됩니다.
  - 텍스트 검색 인덱스 데이터 파일의 인덱스 항목이 갱신됩니다. 파일에 인스턴스의 각 인덱스에 대한 갱신 스케줄 및 비어 있는 스케줄이 지속적으로 표시됩니다.

## SYSTS\_CLEAR\_COMMANDLOCKS 프로시저 - 텍스트 검색 인덱스에 대한 명령 잠금 제거

이 프로시저는 데이터베이스에서 특정 텍스트 검색 인덱스 또는 전체 텍스트 검색 인덱스에 대한 모든 명령 잠금을 제거합니다.

명령 잠금은 텍스트 검색 인덱스 명령 시작 시 작성되며 명령 완료 시 삭제됩니다. 이렇게 하면 다른 명령 간의 바람직하지 않은 충돌을 방지할 수 있습니다.

정리는 더 이상 사용하지 않는 프로세스와 연관된 모든 잠금에 대해 자동으로 수행됩니다. 이는 새 검색 요청 시 텍스트 검색 인덱스에 액세스할 수 있도록 하기 위해 수행됩니다. 이 프로시저는 자주 사용되지는 않지만 예기치 않은 시스템 동작으로 인해 잠금이 남아 있고 명시적으로 제거해야 하는 경우 사용합니다.

이 프로시저는 데이터베이스 서버에서 CLEAR COMMAND LOCKS 텍스트 검색 관리 명령을 발행합니다.

### 구문

```
►►SYSTS_CLEAR_COMMANDLOCKS(—index_schema—,—index_name—,——————►
►message_locale—,—message—)—————►◄◄
```

스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *index\_schema*

텍스트 인덱스의 스키마를 지정하는 VARCHAR(128) 유형의 입력 인수. *index\_schema*는 DB2 스키마 이름에 대한 제한 이름 지정을 따라야 합니다. 인수가 널(NULL)이거나 비어 있는 문자열인 경우, CURRENT SCHEMA의 값이 사용됩니다. *index\_schema*는 대소문자를 구분합니다.

### *index\_name*

인덱스의 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. *index\_schema*와 함께, 데이터베이스에서 텍스트 검색 인덱스를 고유하게 식별합니다. 인수가 널(NULL) 또는 비어 있는 문자열이면, 이 프로시저는 데이터베이스의 모든 텍스트 검색 인덱스에 대한 명령 잠금을 삭제합니다. *index\_name*은 대소문자를 구분합니다.

### *message\_locale*

리턴된 오류 메시지에 사용될 로케일을 지정하는 VARCHAR(33) 유형의 입력 인수. 인수가 널(null)이거나 빈 문자열이거나 지정된 로케일의 메시지 파일을 서버에서 사용할 수 없는 경우 'en\_US'가 사용됩니다.

### *message*

완료된 조작에 대해 경고나 정보 메시지를 지정하는 VARCHAR(32K) 유형의 출력 인수.

## 권한 부여

인덱스 이름이 지정되지 않은 경우 데이터베이스 연결의 *username*에는 DBADM 권한이 있어야 합니다. 특정 인덱스에 대한 명령 잠금을 지우려면 데이터베이스 연결의 *username*에 텍스트 검색 인덱스가 작성된 테이블에 대한 CONTROL 특권이 있어야 합니다.

## 예:

예 1: 다음 예에서, SYSTS\_CLEAR\_COMMANDLOCKS는 *index\_schema* 'db2ts' 및 *index\_name* 'myTextIndex'가 있는 텍스트 검색 인덱스에 대해 발행됩니다. 영어로 리턴되도록 오류 메시지가 요청됩니다. 프로시저가 완료되면 출력 매개변수 메시지가 완료된 조작이 호출자에게 리턴되었음을 표시합니다.

```
CALL SYSPROC.SYSTS_CLEAR_COMMANDLOCKS('db2ts', 'myTextIndex', 'en_US', ?)
```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

-----

매개변수 이름 : MESSAGE

매개변수 값 : 조작이 완료되었습니다.

리턴 상태 = 0

예 2: 다음 예에서, SYSTS\_CLEAR\_COMMANDLOCKS는 *index\_schema* 'db2ts' 및 *index\_name* 'myTextIndex'가 있는 텍스트 검색 인덱스에 대한 명령 잠금을 지우기 위해 호출됩니다. 이 인덱스가 존재하지 않으므로 프로시저가 오류 메시지를 리턴합니다.

```
CALL SYSPROC.SYSTS_CLEAR_COMMANDLOCKS('db2ts', 'myTextIndex', 'en_US', ?)
```

이 쿼리의 출력 예는 다음과 같습니다.

```
SQL20427N 텍스트 검색 관리 프로시저 또는 명령 중  
오류가 발생했습니다. 오류 메시지는 "CIE00316 텍스트 검색  
인덱스 "db2ts"."myTextIndex"가 존재하지 않습니다."  
SQLSTATE 38H14
```

## 사용 시 참고사항

- 텍스트 검색 관리 프로시저는 데이터베이스에 대한 기존 연결을 사용합니다. 프로시저 완료에 따라 현재 트랜잭션이 커밋되거나 롤백될 수 있습니다. 모든 트랜잭션 변경을 커밋하여 이러한 커밋이나 롤백으로부터의 예기치 않은 영향을 피하려고 할 수 있습니다. AUTOCOMMIT를 켜는 것이 유일한 방법입니다.
- SYSIBMTS.TSLOCKS 뷰의 프로세스 및 임계값 정보를 사용하여 잠금을 보유하고 있는 임계값 또는 프로세스가 아직 존재하는지 여부를 확인할 수 있습니다. 텍스트 검색 관리 프로시저 또는 명령(예: SYSTS\_UPDATE 또는 UPDATE INDEX) 실행에 속하는 기존 프로세스의 잠금을 지우면 안됩니다.
- 이 프로시저는 명령 잠금을 소유하는 프로세스가 종료된 경우 호출합니다. 이 경우 명령(잠금으로 표시)이 완료되지 않고 인덱스가 실행되지 않을 수 있습니다. 적절한 조치를 수행해야 합니다. 예를 들어, DROP INDEX 명령을 실행하는 프로세스가 갑자기 종료됩니다. 이 경우 일부 인덱스 데이터가 삭제되었지만 모든 카탈로그 및 컬렉션 정보가 삭제되지는 않았습니다. 명령 잠금은 그대로 유지됩니다. DROP INDEX 명령 잠금을 지운 후 SYSTS\_DROP 프로시저를 재실행할 수 있습니다. 또는 SYSTS\_CREATE 프로세스를 실행하는 프로세스가 갑자기 종료됩니다. 이 경우 일부 인덱스 카탈로그 및 컬렉션 정보를 작성했지만 모든 정보를 삭제하지는 않습니다. 명령 잠금은 그대로 유지됩니다. 명령 잠금을 지우면 SYSTS\_DROP 및 SYSTS\_CREATE 프로시저를 실행할 수 있습니다.
- 이 프로시저가 실행되면, DB2 텍스트 검색 뷰 SYSIBMTS.TSLOCKS의 콘텐츠가 갱신됩니다.

---

## SYSTS\_CLEAR\_EVENTS 프로시저 - 인덱스의 이벤트 테이블에서 인덱싱 이벤트 삭제

이 프로시저는 관리에 사용되는 인덱스의 이벤트 테이블에서 인덱싱 이벤트를 삭제합니다.

이벤트 테이블의 이름은 컬럼 EVENTVIEWNAME의 뷰 SYSIBMTS.TSINDEXES에 있습니다. 모든 인덱스 갱신 조작은 이벤트 테이블에서 정보 및 경우에 따라 오류 항목을 생성합니다. 자동 갱신의 경우, 이벤트 테이블을 정기적으로 검사해야 합니다. 문서의 특정 오류는 문서 콘텐츠를 변경하여 정정해야 합니다. 오류를 정정한 후 이벤트를 지울 수 있으며 너무 많은 공간을 소비하지 않도록 지워야 합니다.

이 프로시저는 데이터베이스 서버에서 CLEAR EVENTS FOR INDEX 텍스트 검색 관리 명령을 실행합니다.

## 구문

```
►►—SYSTS_CLEAR_EVENTS—(—index_schema—,—index_name—,——————►
►—message_locale—,—message—)—————►◄
```

스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *index\_schema*

텍스트 검색 인덱스의 스키마를 지정하는 유형 VARCHAR(128)의 입력 인수입니다. *index\_schema*는 DB2 스키마 이름에 대한 제한 이름 지정을 따라야 합니다. 인수가 널(NULL)이거나 비어 있는 문자열인 경우, CURRENT SCHEMA의 값이 사용됩니다. *index\_schema*는 대소문자를 구분합니다.

### *index\_name*

인덱스의 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. *index\_schema*와 함께, 데이터베이스에서 텍스트 검색 인덱스를 고유하게 식별합니다. *index\_name*은 대소문자를 구분합니다.

### *message\_locale*

리턴된 오류 메시지에 사용될 로케일을 지정하는 VARCHAR(33) 유형의 입력 인수. 인수가 널(null)이거나 빈 문자열이거나 지정된 로케일의 메시지 파일을 서버에서 사용할 수 없는 경우 'en\_US'가 사용됩니다.

### *message*

완료된 조작에 대해 경고나 정보 메시지를 지정하는 VARCHAR(32K) 유형의 출력 인수.

## 권한 부여

명령문의 권한 부여 ID는 적어도 다음과 같은 특권 중 하나를 가지고 있어야 합니다.

- 인덱스가 정의되는 테이블에 대한 CONTROL 특권
- DBADM 권한

## 예:

예 1: 다음 예에서, SYSTS\_CLEAR\_EVENTS는 *index\_schema* 'db2ts' 및 *index\_name* 'myTextIndex'를 사용하여 작성된 텍스트 검색 인덱스에 대해 호출 중입니다. 리턴될 오류 메시지가 영어로 요청되었습니다. 프로시저가 완료되면 출력 매개 변수 메시지가 완료된 조작이 호출자에게 리턴되었음을 표시합니다.

```
CALL SYSPROC.SYSTS_CLEAR_EVENTS('db2ts', 'myTextIndex', 'en_US', ?)
```

이 쿼리의 출력 예는 다음과 같습니다.

```
출력 매개변수 값
-----
매개변수 이름   : MESSAGE
매개변수 값     : 조작이 완료되었습니다.

리턴 상태 = 0
```

예 2: 다음 예에서, SYSTS\_CLEAR\_EVENTS는 *index\_schema* 'db2ts' 및 *index\_name* 'myTextIndex'가 있는 텍스트 검색 인덱스에 대한 이벤트 테이블 항목을 지우기 위해 호출됩니다. 이 인덱스가 존재하지 않으므로 오류가 발생하였습니다.

```
CALL SYSPROC.SYSTS_CLEAR_EVENTS('db2ts', 'myTextIndex', 'en_US', ?)
```

이 쿼리의 출력 예는 다음과 같습니다.

```
SQL20427N 텍스트 검색 관리 프로시저 또는 명령 중
오류가 발생했습니다. 오류 메시지는 "CIE00316 텍스트 검색
인덱스 "db2ts"."myTextIndex"가 존재하지 않습니다.". SQLSTATE 38H14
```

## 사용 시 참고사항

- 텍스트 검색 관리 프로시저는 데이터베이스에 대한 기존 연결을 사용합니다. 프로시저 완료에 따라 현재 트랜잭션이 커밋되거나 롤백될 수 있습니다. 모든 트랜잭션 변경을 커밋하여 이러한 커밋나 롤백으로부터의 예기치 않은 영향을 피하려고 할 수 있습니다. AUTOCOMMIT를 켜는 것이 유일한 방법입니다.
- 충돌 가능성이 있는 경우에는 텍스트 검색 인덱스에서 동시에 여러 프로시저 또는 명령을 실행할 수 없습니다. 충돌 프로시저와 명령의 예는 다음과 같습니다.
  - SYSTS\_ALTER 프로시저 또는 ALTER INDEX db2ts 명령
  - SYSTS\_DISABLE 프로시저 또는 DISABLE DATABASE FOR TEXT db2ts 명령
  - SYSTS\_DROP 프로시저 또는 DROP INDEX db2ts 명령
  - STOP FOR TEXT db2ts 명령
  - SYSTS\_UPDATE 프로시저 또는 UPDATE INDEX db2ts 명령

충돌이 있는 경우 프로시저는 SQLCODE -20426 및 SQLSTATE 38H13을 리턴합니다.

- 정기적인 갱신이 스케줄되는 경우(SYSTS\_CREATE 또는 SYSTS\_ALTER 프로시저의 UPDATE FREQUENCY 옵션 참조), 이벤트 테이블을 정기적으로 점검해야 합니다.
- 텍스트 검색 인덱스의 DB2 텍스트 검색 이벤트 테이블을 정리하려면, 이벤트 이유를 점검하고 오류 소스를 제거한 후 SYSTS\_CLEAR\_EVENTS 프로시저 또는 CLEAR EVENTS FOR INDEX db2ts 명령을 사용하십시오.
- 이벤트 테이블에서 참조되는 모든 행을 변경해야 합니다. 사용자 테이블의 행을 변경하여 SYSTS\_UPDATE 프로시저 또는 UPDATE INDEX db2ts 명령을 다시 실행할 때 오류가 발생한 문서를 다시 인덱스화하도록 시도합니다.
- 이 명령이 발행되면, 이벤트 테이블이 지워집니다.

---

## SYSTS\_CREATE 프로시저 - 컬럼에서 텍스트 검색 인덱스 작성

이 프로시저는 텍스트 검색 기능을 사용하여 컬럼 데이터를 검색할 수 있는 텍스트 컬럼의 텍스트 검색 인덱스를 작성합니다.

텍스트 검색 인덱스가 한 번 작성되면, 컬럼은 쿼리의 텍스트 검색 기능을 사용하여 검색될 수 있습니다. 인덱스에는 인덱스에 정의된 갱신 빈도에 따라 사용자가 텍스트 검색 UPDATE INDEX 명령 또는 SYSTS\_UPDATE 프로시저를 명시적으로 실행하거나, 텍스트 검색 인스턴스 레벨 서비스로 내재적으로 실행할 때까지 데이터가 포함되지 않습니다.

프로시저는 데이터베이스 서버에서 CREATE INDEX 텍스트 검색 관리 명령을 발행합니다.

### 구문

```

▶▶ SYSTS_CREATE(—index_schema—, —index_name—, — text source |—, —————▶
▶ | options |—, —message_locale—, —message—)—————▶▶

```

텍스트 소스:

```

|—table-name—(—| text column name |—),—————|

```

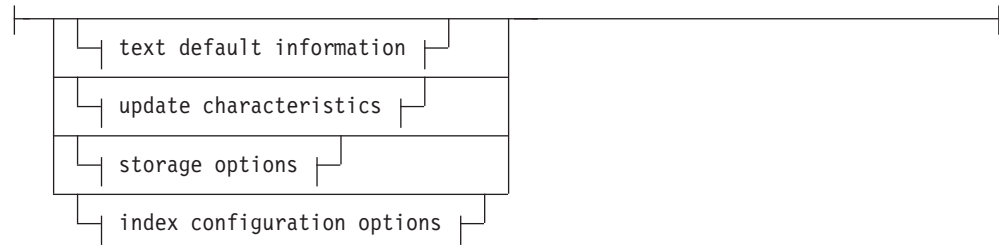
텍스트 컬럼 이름:

```

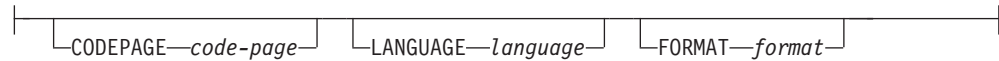
|—column-name—————|
|—function-name(column-name)—|

```

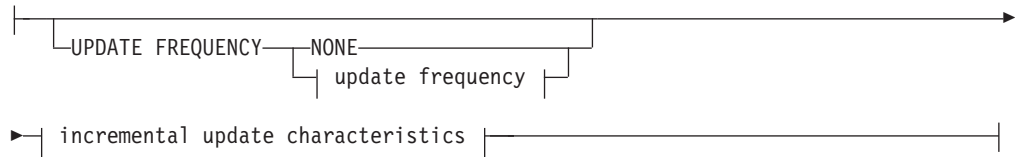
옵션:



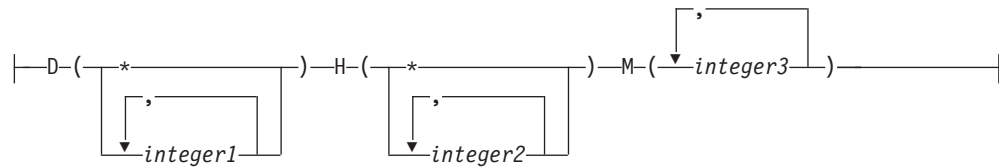
텍스트 디폴트 정보:



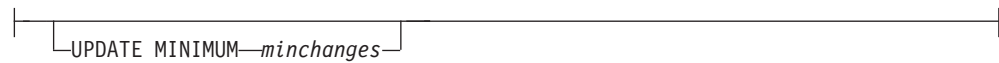
갱신 특성:



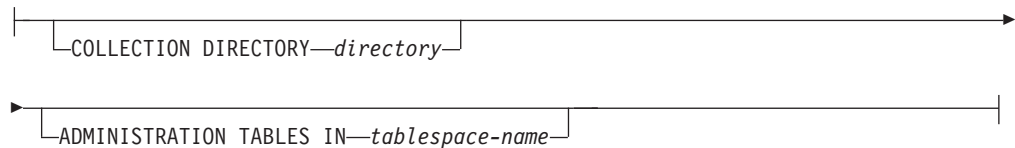
갱신 빈도:



증분 갱신 특성:

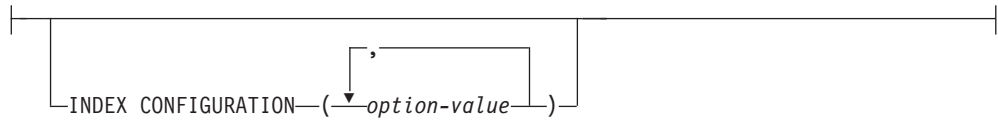


스토리지 옵션:



인덱스 구성 옵션:





스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *index\_schema*

텍스트 검색 인덱스의 스키마를 지정하는 유형 VARCHAR(128)의 입력 인수입니다. *index\_schema*는 DB2 스키마 이름에 대한 제한 이름 지정을 따라야 합니다. 인수가 널(NULL)이거나 비어 있는 문자열인 경우, CURRENT SCHEMA의 값이 사용됩니다. *index\_schema*는 대소문자를 구분합니다.

### *index\_name*

인덱스의 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. *index\_schema*와 함께, 데이터베이스에서 텍스트 검색 인덱스를 고유하게 식별합니다. *index\_name*은 대소문자를 구분합니다.

### 텍스트 소스

인덱스화될 컬럼의 이름을 지정하는 VARCHAR(1024) 유형의 입력 인수입니다. 옵션은 다음과 같습니다.

#### *table-name*

텍스트 컬럼을 포함하는 테이블 이름. 다음 테이블에서는 텍스트 검색 인덱스를 작성할 수 없습니다.

- 범위 파티션 테이블
- 페더레이티드 테이블
- 구체화된 쿼리 테이블
- 뷰

*table-name*은 대소문자를 구분합니다.

### 텍스트 컬럼 이름

인덱스화될 컬럼의 컬럼 이름.

#### *column-name*

컬럼의 올바른 데이터 유형은 CHAR, VARCHAR, LONG VARCHAR, CLOB, DBCLOB, BLOB, GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC 또는 XML입니다. 컬럼의 데이터 유형이 이러한 유형이 아닌 경우 *function-schema.function-name*으로 지정된 변환 함수를 사용하여 컬럼 유형을 유효한 유형 중 하나로 변환하십시오. 구문 및 세부사항은 *function-name(column-name)*의 내용을 참조하십시오. 또는 인덱스화될 텍

스트 문서에 액세스하는 사용자 정의 외부 함수를 지정할 수 있습니다. 하나의 컬럼에 단일 텍스트 검색 인덱스만 작성할 수 있습니다. *column-name* 은 대소문자를 구분합니다.

#### *function-name(column-name)*

텍스트 검색에 지원되는 유형이 아닌 컬럼에서 텍스트 문서에 액세스하며 DB2 이름 지정 규칙을 준수하는 외부 스칼라 함수의 스키마 규정 이름을 지정합니다. 해당 값의 데이터 유형 변환을 수행하고, 텍스트 검색에 지원되는 데이터 유형 중 하나로 값을 리턴합니다. 해당 태스크는 컬럼 자료형 변환을 수행하는 것입니다. 이 함수는 하나의 매개변수만 사용하고 하나의 값만 리턴해야 합니다. *function-name(column-name)*은 대소문자를 구분합니다.

### **options**

사용될 옵션을 지정하는 VARCHAR(1024) 유형의 입력 인수입니다. 옵션이 필요하지 않은 경우, 인수는 널(NULL) 또는 비어 있는 문자열이 될 수 있습니다. 사용 가능한 옵션은 다음과 같습니다.

#### **CODEPAGE** *code-page*

텍스트 문서를 인덱싱할 때 사용될 DB2 코드 페이지(CODEPAGE)를 지정합니다. 디폴트값은 SYSIBMTS.TSDEFAULTS의 값으로 지정됩니다. 여기서 DEFAULTNAME은 'CODEPAGE'(데이터베이스 코드 페이지)입니다. 이 인수는 2진 데이터 유형에만 적용됩니다. 즉, 변환 함수의 컬럼 유형 또는 리턴 유형은 BLOB 또는 FOR BIT DATA 문자 유형이어야 합니다.

#### **LANGUAGE** *language*

인덱싱 중에 문서의 언어 특정 처리를 위해 DB2 텍스트 검색에서 사용될 언어를 지정합니다. 로케일을 지정하지 않으면, 데이터베이스 지역을 사용하여 LANGUAGE의 디폴트 설정을 판별합니다. 문서를 자동으로 스캔하여 로케일을 판별하려면, 로케일을 AUTO로 지정하십시오.

#### **FORMAT** *format*

컬럼에서 텍스트 문서의 형식을 지정합니다. 지원되는 형식은 TEXT, XML 및 HTML 등입니다. DB2 텍스트 검색에서는 문서를 인덱싱할 때 이 정보가 필요합니다. 형식이 지정되지 않으면 디폴트값이 사용됩니다. 디폴트값은 SYSIBMTS.TSDEFAULTS 뷰 (DEFAULTNAME='FORMAT')에 있습니다. XML 데이터 유형 컬럼의 경우 DEFAULTNAME 값에 관계없이 디폴트 형식 'XML'이 사용됩니다.

#### **UPDATE FREQUENCY**

인덱스 갱신 빈도를 지정합니다. 인덱스는 변경 수가 UPDATE MINIMUM에 설정된 값 이상인 경우 갱신됩니다. 갱신 빈도 NONE은 인덱스가 더 이상 갱신되지 않음을 나타냅니다. 이 옵션은 변경되지 않는 데이터가 있는 테이블의 텍스트 컬럼에 유용합니다. 또한 사용자가 UPDATE INDEX 명령을 사용하여

인덱스를 수동으로 갱신하려는 경우에도 유용합니다. 자동 갱신은 START FOR TEXT 명령이 실행되고 DB2 텍스트 검색 인스턴스 서비스가 실행 중인 경우에만 수행될 수 있습니다.

디폴트 빈도 값은 SYSIBMTS.TSDEFAULTS 뷰 (DEFAULTNAME='UPDATEFREQUENCY')에서 가져옵니다.

#### **NONE**

더 이상 인덱스가 갱신되지 않습니다. 갱신은 수동으로 시작해야 합니다.

#### **D** 인덱스가 갱신된 요일.

\* 모든 요일.

*integer1*

일요일에서 토요일까지의 특정 요일(0 - 6)

#### **H** 인덱스가 갱신된 지정된 날짜의 시간.

\* 모든 시간.

*integer2*

자정에서 오후 11시까지 하루 중 특정 시간(0 - 23)

#### **M** 인덱스가 갱신된 지정된 시간의 분.

*integer3*

시간의 맨 위(0) 또는 해당 시간 이후 5분 간격으로 증가하는 시간(0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 또는 55)

#### **UPDATE MINIMUM** *minchanges*

UPDATE FREQUENCY에 지정된 시간에 인덱스가 점진적으로 갱신되기 전에 텍스트 문서에 대한 최소 변경사항 수를 지정합니다. 양수 값만 허용됩니다. 디폴트 값은 SYSIBMTS.TSDEFAULTS 뷰 (DEFAULTNAME='UPDATEFREQUENCY')에서 가져옵니다.

주: 이 값은 UPDATE INDEX 명령에서 무시됩니다(USING UPDATE MINIMUM 옵션이 사용되지 않는 경우). 작은 값으로도 테이블 컬럼과 텍스트 검색 인덱스 사이의 일관성이 증가합니다. 그러나 성능 오버헤드 또한 증가합니다.

#### **COLLECTION DIRECTORY** *directory*

텍스트 검색 인덱스가 저장되는 디렉토리. 디폴트로, 콜렉션 데이터는 *DBPATH/NODExxxx/SQLxxxx/db2collections/name>index identifier/data*에 있습니다. 절대 경로를 지정해야 합니다. 절대 경로 이름의 최대 길이는 215자입니다.

### ADMINISTRATION TABLES IN *tablespace-name*

인덱스에 대해 작성되는 관리 테이블의 기존 일반 테이블 스페이스 이름을 지정합니다. 지정되지 않는 경우 인덱스가 작성되는 기본 테이블의 테이블 스페이스가 사용됩니다.

### INDEX CONFIGURATION (*option-value*)

추가 인덱스 관련 값을 옵션 값 문자열 쌍으로 지정합니다. 이러한 값은 작은 따옴표로 묶어야 합니다.

주: 문자열 값 내부의 작은따옴표 문자는 두 개의 연속 작은따옴표로 나타내야 합니다. 지원되는 값은 다음과 같습니다.

표 201. 옵션 값 스펙

옵션	허용되는 값(디폴트값)	의미
COMMENT	512바이트보다 짧은 문자열 값.	DB2 텍스트 검색 카탈로그 뷰 TSINDEXES의 REMARKS 컬럼에 문자열 주석 값을 추가합니다. 또한 문자열 주석 값을 컬렉션의 설명으로서 추가합니다.

예: INDEX CONFIGURATION(COMMENT 'Index on User''s Guide column')

### *message\_locale*

리턴된 오류 메시지에 사용될 로케일을 지정하는 VARCHAR(33) 유형의 입력 인수. 인수가 널(NULL)이거나 비어 있는 문자열인 경우 또는 지정된 로케일의 메시지 파일을 서버에서 사용 불가능한 경우, 'en\_US'가 사용됩니다.

### *message*

완료된 조작에 대해 경고나 정보 메시지를 지정하는 VARCHAR(32K) 유형의 출력 인수.

## 권한 부여

명령문의 권한 부여 ID는 적어도 다음과 같은 특권 중 하나를 가지고 있어야 합니다.

다음 중 하나

- 인덱스가 정의되는 테이블에 대한 CONTROL 특권
- 인덱스가 정의되는 테이블에 대한 INDEX 특권

다음 중 하나이어야 합니다.

- 인덱스의 내재적 또는 명시적 스키마 이름이 존재하지 않을 경우, 데이터베이스에 대한 IMPLICIT\_SCHEMA 권한
- 인덱스의 스키마 이름이 기존의 스키마를 참조할 경우, 스키마에 대한 CREATEIN 특권
- DBADM 권한

예:

예 1: 다음 예에서, *index\_schema* 'db2ts' 및 *index\_name* 'myTextIndex'가 있는 텍스트 검색 인덱스는 SYSTS\_CREATE 프로시저를 사용하여 작성됩니다. 'UPDATE MINIMUM 10' 옵션은 인덱스의 증분 갱신이 수행되기 전에 인덱스와 연관된 텍스트 문서에 최소 10가지가 변경되어야 함을 지정합니다. 리턴될 오류 메시지가 영어로 요청되었습니다. 밑줄 굵기 텍스트 검색 명령이 성공적으로 실행된 경우, 출력 매개변수 message는 명령 실행의 상태를 표시하도록 설정됩니다.

```
CALL SYSPROC.SYSTS_CREATE('db2ts', 'myTextIndex',
    'myUserSchema.myBaseTable(myTextColumn)', 'UPDATE MINIMUM 10',
    'en_US', ?)
```

이 쿼리의 출력 예는 다음과 같습니다.

```
출력 매개변수 값
-----
매개변수 이름 : MESSAGE
매개변수 값 : 성공적으로 완료된 작업입니다.
리턴 상태 = 0
```

예 2: 다음 예에서, SYSTS\_CREATE는 *index\_schema* 'db2ts' 및 *index\_name* 'myTextIndex'가 있는 텍스트 검색 인덱스를 작성하기 위해 호출됩니다. 옵션이 지정되지 않았습니다. 이 예에서, 인덱스는 이미 존재하며 호출자에 리턴 중인 오류 메시지를 발생시켰습니다.

```
CALL SYSPROC.SYSTS_CREATE('db2ts', 'myTextIndex',
    'myUserSchema.myBaseTable(myTextColumn)', '', 'en_US', ?)
```

이 쿼리의 출력 예는 다음과 같습니다.

```
SQL20427N 텍스트 검색 관리 중에
오류가 발생했습니다.
프로시저 또는 명령입니다. 오류 메시지는 "CIE00201 텍스트 검색
인덱스 "db2ts"입니다. "myTextIndex"가 이미 존재합니다. ".
```

## 사용 시 참고사항

- 텍스트 검색 관리 프로시저는 데이터베이스에 대한 기존 연결을 사용합니다. 프로시저 완료에 따라 현재 트랜잭션이 커밋되거나 롤백될 수 있습니다. 모든 트랜잭션 변경을 커밋하여 이러한 커밋이나 롤백으로부터의 예기치 않은 영향을 피하려고 할 수 있습니다. AUTOCOMMIT를 켜는 것이 유일한 방법입니다.
- 충돌 가능성이 있는 경우에는 텍스트 검색 인덱스에서 동시에 여러 프로시저나 명령을 실행할 수 없습니다. 충돌 프로시저와 명령의 예는 다음과 같습니다.
  - SYSTS\_ALTER 프로시저 또는 ALTER INDEX db2ts 명령
  - SYSTS\_CLEAR\_EVENTS 프로시저 또는 CLEAR EVENTS FOR INDEX db2ts 명령

- SYSTS\_DISABLE 프로시저 또는 DISABLE DATABASE FOR TEXT db2ts 명령
- STOP FOR TEXT db2ts 명령
- SYSTS\_UPDATE 프로시저 또는 UPDATE INDEX db2ts 명령

충돌이 있는 경우 프로시저는 SQLCODE -20426 및 SQLSTATE 38H13을 리턴합니다.

- CREATE INDEX 명령 실행에 성공하는 경우:
  - DB2 텍스트 검색 서버 데이터가 갱신됩니다. 다음 예와 같이 이름이 *instance\_database-name\_index-identifier\_number*인 컬렉션이 작성됩니다.
 

```
tigertail_MYTSDB_TS250517_0000
```

컬렉션 이름은 SYSIBMTS.TSCOLLECTIONNAMES 뷰(COLLECTIONNAME 컬럼)에서 검색할 수 있습니다.
  - DB2 텍스트 검색 카탈로그 정보가 갱신됩니다. 적합한 DB2 인덱스와 함께 지정된 테이블 스페이스에 인덱스 스테이징 테이블이 작성됩니다. 또한 지정된 테이블 스페이스에 인덱스 이벤트 테이블이 작성됩니다.
  - 새로 작성된 텍스트 검색 인덱스는 자동으로 채워지지 않습니다. SYSTS\_UPDATE 프로시저 또는 UPDATE INDEX 명령은 채워질 텍스트 검색 인덱스에 대해 수동 또는 자동으로 실행되어야 합니다(UPDATE FREQUENCY 옵션 스펙을 통해 인덱스에 대해 정의된 갱신 스케줄의 결과로서).
  - DB2 데이터베이스 서버의 텍스트 검색 인덱스 데이터 파일은 갱신됩니다. 인스턴스의 각 인덱스마다 스케줄된 갱신 정보가 기록됩니다.

사용 제한사항:

- 테이블에 대한 기본 키가 정의되어야 합니다. DB2 텍스트 검색에서는 유형 제한 없이 다중 컬럼 DB2 기본 키를 사용할 수 있습니다. 기본 키 컬럼의 수는 DB2에서 허용되는 기본 키 컬럼의 수보다 작은 두 개 컬럼으로 제한됩니다.
- DB2 텍스트 검색 인덱스가 있는 테이블의 모든 기본 키 컬럼의 총 길이는 DB2에서 허용되는 최대 총 기본 키 길이보다 작은 15바이트로 제한됩니다. DB2 CREATE INDEX문의 DB2 제한사항을 참조하십시오.

---

## SYSTS\_DISABLE 프로시저 - 텍스트 검색을 위해 현재 데이터베이스 사용 안 함

현재 데이터베이스에 DB2 텍스트 검색을 사용 불가능하게 합니다.

텍스트 검색 기능이 사용 불가능하면 텍스트 검색 인덱스 및 명령을 더 이상 데이터베이스와 함께 사용할 수 없습니다.

프로시저는 데이터베이스 서버에서 DISABLE DATABASE FOR TEXT 텍스트 검색 관리 명령을 발행합니다.

## 구문

```
►—SYSTS_DISABLE—(—options—, —message_locale—, —message—)—————►
```

스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *options*

데이터베이스가 사용 불가능할 때 사용될 옵션을 지정하는 VARCHAR(128) 유형의 입력 인수. 인수는 FORCE로 설정될 수 있습니다. 값이 지정되면 모든 인덱스가 삭제되며 텍스트 검색 기능이 강제로 사용 불가능하게 됩니다. 텍스트 검색 인덱스가 보존되지 않으며 오류 메시지나 경고가 리턴되지 않습니다. 인수가 널(null)이거나 빈 문자열이면 데이터베이스에 대해 텍스트 검색 기능을 사용할 수 없게 합니다.

### *message\_locale*

리턴된 오류 메시지에 사용될 로케일을 지정하는 VARCHAR(33) 유형의 입력 인수. 인수가 널(null)이거나 빈 문자열이거나 지정된 로케일의 메시지 파일을 서버에서 사용할 수 없는 경우 'en\_US'가 사용됩니다.

### *message*

완료된 조작에 대해 경고나 정보 메시지를 지정하는 VARCHAR(32K) 유형의 출력 인수.

## 권한 부여

명령문의 권한 부여 ID가 보유하는 특권에는 DBADM 권한이 포함되어야 합니다.

## 예:

예 1: 다음 예에서는 SYSTS\_DISABLE 프로시저를 사용한 데이터베이스에 대한 텍스트 검색이 불가능합니다. 텍스트 검색 인덱스가 데이터베이스의 테이블에 여전히 있더라도 기능을 사용 불가능하게 하도록 FORCE 옵션이 지정됩니다. 영어로 되돌아가도록 요청하는 오류 메시지가 지정됩니다. *message* 출력 매개변수는 정보 메시지 문자열로 설정됩니다.

```
CALL SYSPROC.SYSTS_DISABLE('FORCE', 'en_US', ?)
```

이 쿼리의 출력 예는 다음과 같습니다.



출력 매개변수 값

-----  
매개변수 이름 : MESSAGE  
매개변수 값 : 조작이 완료되었습니다.

리턴 상태 = 0

예 2: 다음 예에서는 FORCE 옵션을 지정하지 않고 SYSTS\_DISABLE 프로시저를 사용한 기존 텍스트 검색 인덱스가 있는 데이터베이스에 대해 텍스트 검색이 불가능합니다. 호출자에게 오류 메시지를 표시하게 됩니다. 텍스트 검색 기능을 사용 불가능하게 하기 전에 기존 모든 텍스트 검색 인덱스를 삭제하거나 또는 FORCE 옵션을 options 입력 매개변수 값으로 지정하는 것이 좋습니다.

```
CALL SYSPROC.SYSTS_DISABLE('', 'en_US', ?)
```

이 쿼리의 출력 예는 다음과 같습니다.

```
SQL20427N 텍스트 검색 관리 프로시저 또는 명령 중  
오류가 발생했습니다. 오류 메시지는 다음과 같습니다.  
"CIE00326 지정된 데이터베이스 또는  
디폴트 데이터베이스에서 텍스트 검색 인덱스가 활성화됩니다."  
SQLSTATE 38H14
```

## 사용 시 참고사항

- 텍스트 검색 관리 프로시저는 데이터베이스에 대한 기존 연결을 사용합니다. 프로시저 완료에 따라 현재 트랜잭션이 커밋되거나 롤백될 수 있습니다. 모든 트랜잭션 변경을 커밋하여 이러한 커밋나 롤백으로부터의 예기치 않은 영향을 피하려고 할 수 있습니다. AUTOCOMMIT를 켜는 것이 유일한 방법입니다.
- 충돌 가능성이 있는 경우에는 텍스트 검색 인덱스에서 동시에 여러 프로시저나 명령을 실행할 수 없습니다. 충돌 프로시저와 명령의 예는 다음과 같습니다.

- SYSTS\_ALTER 프로시저 또는 ALTER INDEX db2ts 명령
- SYSTS\_CLEAR\_EVENTS 프로시저 또는 CLEAR EVENTS FOR INDEX db2ts 명령
- SYSTS\_DISABLE 프로시저 또는 DISABLE DATABASE FOR TEXT db2ts 명령
- STOP FOR TEXT db2ts 명령
- SYSTS\_UPDATE 프로시저 또는 UPDATE INDEX db2ts 명령

충돌이 있는 경우 프로시저는 SQLCODE -20426 및 SQLSTATE 38H13을 리턴합니다.

- 이 프로시저를 실행할 경우,
  - DB2 텍스트 검색 카탈로그 정보가 갱신됩니다. 인덱스 로그 및 이벤트 테이블이 삭제됩니다. 사용자 텍스트 테이블의 트리거가 삭제됩니다.
  - FORCE 옵션이 지정된 경우 모든 텍스트 인덱스 정보가 데이터베이스에서 제거되고 연관된 모든 컬렉션이 삭제됩니다. 또한 나머지 갱신 스케줄 정보를 제거하



기 위해 텍스트 서비스가 갱신됩니다. "db2ts DROP INDEX 명령" 또는 "SYSTS\_DROP 프로시저"를 참조하십시오.

- 이 프로시저는 데이터베이스의 DB2 Net Search Extender 사용 가능 상태에는 영향을 주지 않습니다. SYSTS\_ENABLE 프로시저나 ENABLE FOR TEXT 명령에서 작성된 DB2 텍스트 검색 카탈로그 테이블 및 뷰를 삭제합니다.
- 텍스트 검색 인덱스 정의가 있는 DB2 데이터베이스를 삭제하기 전에, 이 프로시저를 실행하여 텍스트 인덱스 및 컬렉션이 성공적으로 제거되었는지 확인하십시오.
- 일부 인덱스가 FORCE 옵션 사용으로 삭제될 수 없는 경우, 컬렉션 이름이 db2diag 로그 파일에 기록됩니다. 텍스트 검색 인덱스 프로시저 SYSTS\_DISABLE 또는 명령 DISABLE DATABASE FOR TEXT는 CLP 명령 DROP DATABASE 이전에 실행되지 않은 경우, 텍스트 검색 인덱스 서비스도 CLEANUP FOR TEXT 명령을 사용하여 제거해야 합니다. 인덱스 삭제에 대한 자세한 정보는 SYSTS\_DROP 프로시저 또는 DROP INDEX 명령을 참조하고, 텍스트 검색 컬렉션 및 텍스트 검색 인덱스와의 해당 관계에 대한 자세한 정보는 CLEANUP FOR TEXT 명령을 참조하십시오.

주: 사용자는 사용되지 않는 컬렉션(즉, 텍스트 검색 서버에 정의된 상태로 남아 있지만 DB2에서 사용되지 않는 컬렉션)이 발생하도록 사용하면 안됩니다. 다음은 사용되지 않는 컬렉션이 작성되는 몇 가지 원인입니다.

- SYSTS\_DISABLE 프로시저나 DISABLE DATABASE FOR TEXT 명령을 실행하지 않고 DROP DATABASE CLP 명령 또는 DROP TABLE문을 실행하는 경우
- FORCE 옵션을 사용하여 SYSTS\_DISABLE 프로시저가 실행되거나 DISABLE DATABASE FOR TEXT 명령을 실행하는 경우
- 기타 오류 조건. CLEANUP FOR TEXT 명령은 일부 시나리오에서 사용될 수 있습니다.

---

## SYSTS\_DROP 프로시저 - 텍스트 검색 인덱스 삭제

이 프로시저는 모든 테이블 컬럼에 연관된 기존의 텍스트 검색 인덱스를 삭제(drop)합니다.

이 프로시저의 실행이 완료된 후 텍스트 검색 쿼리를 해당 컬럼에 실행할 수 없습니다.

이 프로시저는 데이터베이스 서버에서 DROP INDEX 텍스트 검색 관리 명령을 발행합니다.

### 구문

►►SYSTS\_DROP(—*index\_schema*—,—*index\_name*—,——————►

스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *index\_schema*

텍스트 검색 인덱스의 스키마를 지정하는 유형 VARCHAR(128)의 입력 인수입니다. *index\_schema*는 DB2 스키마 이름에 대한 제한 이름 지정을 따라야 합니다. 인수가 널(NULL)이거나 비어 있는 문자열인 경우, CURRENT SCHEMA의 값이 사용됩니다. *index\_schema*는 대소문자를 구분합니다.

### *index\_name*

인덱스의 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. *index\_schema*와 함께, 데이터베이스에서 텍스트 검색 인덱스를 고유하게 식별합니다. *index\_name*은 대소문자를 구분합니다.

### *message\_locale*

리턴된 오류 메시지에 사용될 로케일을 지정하는 VARCHAR(33) 유형의 입력 인수. 인수가 널(null)이거나 빈 문자열이거나 지정된 로케일의 메시지 파일을 서버에서 사용할 수 없는 경우 'en\_US'가 사용됩니다.

### *message*

완료된 조작에 대해 경고나 정보 메시지를 지정하는 VARCHAR(32K) 유형의 출력 인수.

## 권한 부여

명령문의 권한 부여 ID는 적어도 다음과 같은 특권 중 하나를 가지고 있어야 합니다.

- 인덱스가 정의되는 테이블에 대한 CONTROL 특권
- DBADM 권한

### 예:

예 1: 다음 예에서는 *index\_schema* 'db2ts' 및 *index\_name* 'myTextIndex'를 사용하여 작성된 텍스트 검색 인덱스가 삭제 중입니다. 리턴될 오류 메시지가 영어로 요청되었습니다. 프로시저가 완료되면 출력 매개변수 메시지가 완료된 조작이 호출자에게 리턴되었음을 표시합니다.

```
CALL SYSPROC.SYSTS_DROP('db2ts', 'myTextIndex', 'en_US', ?)
```

이 쿼리의 출력 예는 다음과 같습니다.

출력 매개변수 값

-----  
매개변수 이름 : MESSAGE  
매개변수 값 : 조작이 완료되었습니다.

리턴 상태 = 0

예 2: 다음 예에서, SYSTS\_DROP는 *index\_schema* 'db2ts' 및 *index\_name* 'myTextIndex'가 있는 텍스트 검색 인덱스를 삭제하기 위해 호출됩니다. 이 인덱스가 존재하지 않으므로 오류가 발생했습니다.

```
CALL SYSPROC.SYSTS_DROP('db2ts', 'myTextIndex', 'en_US', ?)
```

이 쿼리의 출력 예는 다음과 같습니다.

```
SQL20427N 텍스트 검색 관리 프로시저 또는 명령 중  
오류가 발생했습니다. 오류 메시지는 "CIE00316 텍스트 검색  
인덱스 "db2ts"."myTextIndex"가 존재하지 않습니다."  
SQLSTATE 38H14
```

## 사용 시 참고사항

- 텍스트 검색 관리 프로시저는 데이터베이스에 대한 기존 연결을 사용합니다. 프로시저 완료에 따라 현재 트랜잭션이 커밋되거나 롤백될 수 있습니다. 모든 트랜잭션 변경을 커밋하여 이러한 커밋나 롤백으로부터의 예기치 않은 영향을 피하려고 할 수 있습니다. AUTOCOMMIT를 켜는 것이 유일한 방법입니다.
- 충돌 가능성이 있는 경우에는 텍스트 검색 인덱스에서 동시에 여러 프로시저나 명령을 실행할 수 없습니다. 충돌 프로시저와 명령의 예는 다음과 같습니다.
  - SYSTS\_ALTER 프로시저 또는 ALTER INDEX db2ts 명령
  - SYSTS\_CLEAR\_EVENTS 프로시저 또는 CLEAR EVENTS FOR INDEX db2ts 명령
  - SYSTS\_DISABLE 프로시저 또는 DISABLE DATABASE FOR TEXT db2ts 명령
  - STOP FOR TEXT db2ts 명령
  - SYSTS\_UPDATE 프로시저 또는 UPDATE INDEX db2ts 명령충돌이 있는 경우 프로시저는 SQLCODE -20426 및 SQLSTATE 38H13을 리턴합니다.
- DB2에서 사용자 테이블을 삭제해도 인덱스 삭제를 트리거하지 않으므로 테이블 삭제 전후에 수동으로 삭제해야 합니다.
- 이 프로시저를 실행할 경우,
  - 텍스트 검색 카탈로그 정보가 갱신됩니다. 인덱스 스테이징 및 이벤트 테이블이 삭제됩니다. 사용자 테이블의 트리거가 삭제됩니다.
  - 텍스트 검색 인덱스 데이터 파일의 인덱스 항목이 삭제됩니다. 파일에 인스턴스의 각 인덱스에 대한 갱신 스케줄 및 비어 있는 스케줄이 지속적으로 표시됩니다.

- 텍스트 검색 인덱스 정의와 연관된 컬렉션이 제거됩니다.
- 텍스트 검색 인덱스 삭제 후 같은 텍스트 컬럼에 새로 작성하려면 우선 연결을 끊은 다음 새 텍스트 검색 인덱스를 작성하기 전에 데이터베이스에 다시 연결해야 합니다.

---

## SYSTS\_ENABLE 프로시저 - 텍스트 검색을 위해 현재 데이터베이스 사용

이 프로시저는 현재 데이터베이스에 대해 DB2 텍스트 검색을 사용 가능하게 합니다.

데이터베이스내의 테이블 컬럼에서 텍스트 검색 인덱스를 작성하기 전에 이 프로시저를 제대로 발행해야 합니다.

이 프로시저는 데이터베이스 서버에서 ENABLE DATABASE FOR TEXT 텍스트 검색 관리 명령을 발행합니다.

### 구문

►►SYSTS\_ENABLE(—*message\_locale*—,—*message*—)◄◄

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *message\_locale*

리턴된 오류 메시지에 사용될 로케일을 지정하는 VARCHAR(33) 유형의 입력 인수. 인수가 널(null)이거나 빈 문자열이거나 지정된 로케일의 메시지 파일을 서버에서 사용할 수 없는 경우 'en\_US'가 사용됩니다.

#### *message*

완료된 조작에 대해 경고나 정보 메시지를 지정하는 VARCHAR(32K) 유형의 출력 인수.

### 권한 부여

사용자에게 ENABLE DATABASE 명령을 실행하기 위한 DBADM 특권이 있어야 합니다.

### 예:

예 1: 텍스트 검색을 위한 데이터베이스를 사용 가능하게 하고 모든 오류 메시지를 영어로 리턴합니다.

```
CALL SYSPROC.SYSTS_ENABLE('en_US', ?)
```

다음은 이 쿼리로부터의 출력 예입니다.

출력 매개변수 값

```
-----  
매개변수 이름   : MESSAGE  
매개변수 값     : 조작이 완료되었습니다.
```

리턴 상태 = 0

예 2: 다음 예에서 텍스트 검색에 이미 사용할 수 있는 데이터베이스에 SYSTS\_ENABLE을 호출합니다. 호출자에게 오류 메시지를 표시하게 됩니다.

```
CALL SYSPROC.SYSTS_ENABLE('en_US', ?)
```

다음은 이 쿼리로부터의 출력 예입니다.

```
SQL20427N 텍스트 검색 관리 프로시저 또는 명령 중  
오류가 발생했습니다. 텍스트 검색 제품의 오류 메시지는  
다음과 같습니다. "CIE00322 지정된 데이터베이스 또는 디폴트 데이터베이스를 텍스트에  
이미 사용할 수 있습니다". SQLSTATE 38H14
```

## 사용 시 참고사항

- 텍스트 검색 관리 프로시저는 데이터베이스에 대한 기존 연결을 사용합니다. 프로시저 완료에 따라 현재 트랜잭션이 커밋되거나 롤백될 수 있습니다. 모든 트랜잭션 변경을 커밋하여 이러한 커밋나 롤백으로부터의 예기치 않은 영향을 피하려고 할 수 있습니다. AUTOCOMMIT를 켜는 것이 유일한 방법입니다.
- 이 프로시저를 실행할 경우,
  - 이 프로시저는 텍스트 검색 관리 카탈로그 테이블 및 뷰와 같은 데이터베이스 오브젝트를 SYSIBMTS 스키마에 작성합니다. 이 오브젝트는 데이터베이스의 기본 테이블 스페이스에 있습니다(IBMDEFAULTGROUP).
  - 텍스트 검색 인덱스에 설정된 데이터베이스 디폴트값은 뷰 SYSIBMTS.TSDEFAULTS에 사용할 수 있습니다.
  - 명령이 완료되면 텍스트 검색 카탈로그 테이블 및 뷰가 작성되어 사용됩니다.

---

## SYSTS\_UPDATE 프로시저 - 텍스트 검색 인덱스 갱신

이 프로시저는 인덱스가 연관된 텍스트 컬럼의 현재 콘텐츠를 반영하기 위해 텍스트 검색 인덱스를 갱신합니다.

갱신이 수행되는 동안에는 검색할 수 있습니다. 갱신이 완료될 때까지 부분적으로 갱신된 인덱스에 대해 검색이 수행됩니다.

이 프로시저는 데이터베이스 서버에서 UPDATE INDEX 텍스트 검색 관리 명령을 실행합니다.

## 구문

```
►►—SYSTS_UPDATE—(—index_schema—,—index_name—,——————►  
►—update_options—,—message_locale—,—message—)—————►
```

스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *index\_schema*

텍스트 검색 인덱스의 스키마를 지정하는 유형 VARCHAR(128)의 입력 인수입니다. *index\_schema*는 DB2 스키마 이름에 대한 제한 이름 지정을 따라야 합니다. 인수가 널(NULL)이거나 비어 있는 문자열인 경우, CURRENT SCHEMA의 값이 사용됩니다. *index\_schema*는 대소문자를 구분합니다.

### *index\_name*

인덱스의 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. *index\_schema*와 함께, 데이터베이스에서 텍스트 검색 인덱스를 고유하게 식별합니다. *index\_name*은 대소문자를 구분합니다.

### *update\_options*

갱신 옵션을 지정하는 VARCHAR(32K) 유형의 입력 인수. 가능한 값은 다음과 같습니다.

- USING UPDATE MINIMUM: 이 설정은 CREATE INDEX 텍스트 검색 관리 명령 및 SYSTS\_CREATE 프로시저의 UPDATE MINIMUM 설정을 지킵니다.
- 널(NULL) 또는 비어 있는 문자열 ("): 프로시저가 호출되면 무조건 갱신이 시작됩니다.

### *message\_locale*

리턴된 오류 메시지에 사용될 로케일을 지정하는 VARCHAR(33) 유형의 입력 인수. 인수가 널(null)이거나 빈 문자열이거나 지정된 로케일의 메시지 파일을 서버에서 사용할 수 없는 경우 'en\_US'가 사용됩니다.

### *message*

완료된 조작에 대해 경고나 정보 메시지를 지정하는 VARCHAR(32K) 유형의 출력 인수.

## 권한 부여

명령문의 권한 부여 ID는 적어도 다음과 같은 특권 중 하나를 가지고 있어야 합니다.

- 텍스트 인덱스가 정의되는 테이블에 대한 CONTROL 특권
- DATAACCESS 권한

## 예:

예 1: 다음 예에서는 *index\_schema* 'db2ts' 및 *index\_name* 'myTextIndex'를 사용하여 작성된 텍스트 검색 인덱스가 갱신 중입니다. *update\_options*의 위치에 있는 널 (NULL) 값은 스토어드 프로시저가 호출되면 무조건 갱신이 시작되는 것을 의미합니다. 리턴될 오류 메시지가 영어로 요청되었습니다. 프로시저가 완료되면 출력 매개변수 메시지가 완료된 조작이 호출자에게 리턴되었음을 표시합니다.

```
CALL SYSPROC.SYSTS_UPDATE('db2ts', 'myTextIndex', '', 'en_US', ?)
```

이 쿼리의 출력 예는 다음과 같습니다.

```
출력 매개변수 값
-----
매개변수 이름   : MESSAGE
매개변수 값     : 조작이 완료되었습니다.

리턴 상태 = 0
```

예 2: 다음 예에서, SYSTS\_UPDATE는 *index\_schema* 'db2ts' 및 *index\_name* 'myTextIndex'가 있는 텍스트 검색 인덱스를 갱신하기 위해 호출됩니다. 이 인덱스가 존재하지 않으므로 오류가 발생하였습니다.

```
CALL SYSPROC.SYSTS_UPDATE('db2ts', 'myTextIndex', 'USING UPDATE MINIMUM',
'en_US', ?)
```

이 쿼리의 출력 예는 다음과 같습니다.

```
SQL20427N 텍스트 검색 관리 프로시저 또는 명령 중
오류가 발생했습니다. 오류 메시지는 "CIE00316 텍스트 검색
인덱스 "db2ts"."myTextIndex"가 존재하지 않습니다.".
SQLSTATE 38H14
```

## 사용 시 참고사항

- 텍스트 검색 관리 프로시저는 데이터베이스에 대한 기존 연결을 사용합니다. 프로시저 완료에 따라 현재 트랜잭션이 커밋되거나 롤백될 수 있습니다. 모든 트랜잭션 변경을 커밋하여 이러한 커밋나 롤백으로부터의 예기치 않은 영향을 피하려고 할 수 있습니다. AUTOCOMMIT를 켜는 것이 유일한 방법입니다.
- 충돌 가능성이 있는 경우에는 텍스트 검색 인덱스에서 동시에 여러 프로시저 또는 명령을 실행할 수 없습니다. 충돌 프로시저와 명령의 예는 다음과 같습니다.
  - SYSTS\_ALTER 프로시저 또는 ALTER INDEX db2ts 명령
  - SYSTS\_CLEAR\_EVENTS 프로시저 또는 CLEAR EVENTS FOR INDEX db2ts 명령
  - SYSTS\_DISABLE 프로시저 또는 DISABLE DATABASE FOR TEXT db2ts 명령
  - SYSTS\_DROP 프로시저 또는 DROP INDEX db2ts 명령

- STOP FOR TEXT db2ts 명령
- SYSTS\_UPDATE 프로시저 또는 UPDATE INDEX db2ts 명령

충돌이 있는 경우 프로시저는 SQLCODE -20426 및 SQLSTATE 38H13을 리턴합니다.

- 이 프로시저는 모든 인덱스 갱신 처리가 완료될 때까지 리턴되지 않습니다. 지속기간은 인덱스화될 문서의 수와 이미 인덱스된 문서의 수에 따라 다릅니다. 인덱스의 컬렉션 이름은 SYSIBMTS.TSCOLLECTIONNAMES 뷰(COLLECTIONNAME 컬럼)에서 검색할 수 있습니다.
- 개별 문서 오류가 있는 경우, 문서를 정정해야 합니다. 오류 문서의 기본 키는 인덱스의 이벤트 테이블에서 검색할 수 있습니다. 사용자 테이블의 해당 행을 변경하면 SYSTS\_UPDATE 다음 호출이 이 문서를 다시 처리합니다.
- 이 프로시저를 실행할 경우,
  - 이벤트 테이블에 행이 삽입됩니다(구문 분석기 오류 정보 포함). 증분 갱신의 경우 인덱스 스테이징 테이블에서 정보가 삭제됩니다. 첫 번째 갱신 전에 사용자 테이블의 트리거를 작성합니다.
  - 컬렉션이 갱신됩니다. 새 문서 또는 변경된 문서를 구문 분석하여 인덱스화하며 삭제된 문서를 인덱스에서 버립니다.



---

## 제 19 장 워크로드 관리 루틴

---

### WLM\_CANCEL\_ACTIVITY - 활동 취소

이 프로시저는 제공된 활동을 취소합니다. 취소가 발생하면 취소된 활동을 제출한 응용 프로그램에 오류 메시지가 리턴됩니다.

#### 구문

```
▶▶—WLM_CANCEL_ACTIVITY—(—application_handle—,—uow_id—,—activity_id—)————▶▶
```

스키마는 SYSPROC입니다.

#### 프로시저 매개변수

##### *application\_handle*

활동을 취소할 응용프로그램 핸들을 지정하는 BIGINT 유형의 입력 인수. 인수가 널(NULL)이면 활동을 찾을 수 없으므로 SQLSTATE 5U035가 포함된 SQL4702N이 리턴됩니다.

##### *uow\_id*

취소할 활동의 작업 단위(UOW) ID를 지정하는 INTEGER 유형의 입력 인수. 인수가 널(NULL)이면 활동을 찾을 수 없으므로 SQLSTATE 5U035가 포함된 SQL4702N이 리턴됩니다.

##### *activity\_id*

취소할 작업 단위(UOW) 내의 활동을 고유하게 식별하는 활동 ID를 지정하는 INTEGER 유형의 입력 인수. 인수가 널(NULL)이면 활동을 찾을 수 없으며 SQLSTATE 5U035이 포함된 SQL4702N이 리턴됩니다.

#### 권한 부여

WLM\_CANCEL\_ACTIVITY 프로시저에 대한 EXECUTE 특권

#### 예 :

관리자가 WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 테이블 함수를 사용하여 응용프로그램 핸들, 작업 단위 ID 및 활동의 활동 ID를 찾을 수 있습니다. 응용프로그램 핸들 1, 작업 단위 ID2 및 활동 ID 3으로 활동을 취소하려면 다음을 수행하십시오.

```
CALL WLM_CANCEL_ACTIVITY(1, 2, 3)
```

## 사용 시 참고사항

- 활동을 찾을 수 없으면 SQLSTATE 5U035가 포함된 SQL4702N이 리턴됩니다.
- 올바른 상태가 아니므로(초기화되지 않음) 활동이 취소될 수 없는 경우 SQLSTATE 5U016이 포함된 SQL4703N (이유 코드 1)이 리턴됩니다.
- 활동이 취소되면 SQLSTATE 57014가 포함된 SQL4725N이 취소된 응용프로그램에 리턴됩니다.
- 취소할 때 코디네이터가 다른 활동의 요청을 처리 중이거나 유희 상태인 경우 활동이 CANCEL\_PENDING 상태가 되어 코디네이터가 다음 활동 요청을 처리할 때 취소됩니다.

---

## WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS - 활동 이벤트 모니터에 대한 활동 정보 수집

WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS 프로시저는 지정한 활동에 대한 정보를 수집하여 해당 정보를 활성 활동 이벤트 모니터에 기록합니다.

하위 활동이 있는 활동에 이 프로시저를 적용하는 경우 프로시저는 각 하위 활동에 대해 반복적으로 레코드를 생성합니다. 이 정보는 프로시저를 호출할 때 수집되어 전송되며 프로시저는 상위 활동 실행이 완료될 때까지 대기하지는 않습니다. 이벤트 모니터의 활동 레코드는 부분 레코드로 마크됩니다.

### 구문

```
▶▶—WLM_CAPTURE_ACTIVITY_IN_PROGRESS—(—application_handle—, —————→  
▶—uow_id—, —activity_id—)—————→▶▶
```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

다음 매개변수 중 모두를 지정하지 않으면 활동을 찾지 못하고 SQLSTATE 5U035의 SQL4702N이 리턴됩니다.

#### *application\_handle*

활동 정보가 캡처되는 응용프로그램 핸들을 지정하는 BIGINT 유형의 입력 인수입니다.

#### *uow\_id*

정보가 캡처되는 활동의 작업 단위(UOW) ID를 지정하는 INTEGER 유형의 입력 인수입니다.

*activity\_id*

정보가 캡처되는 작업 단위(UOW)내의 활동을 고유하게 식별하는 활동 ID를 지정하는 INTEGER 유형의 입력 인수입니다.

## 권한 부여

WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS 프로시저에 대한 EXECUTE 특권

예 :

스토어드 프로시저 MYSCHEMA.MYSLOWSTP가 평소보다 느리게 실행되는 것으로 사용자가 불만을 제기합니다. 관리자는 감속의 원인을 조사합니다. 스토어드 프로시저 실행 중에는 조사를 진행할 수 없으므로 관리자는 스토어드 프로시저 활동 및 그 내부에 중첩된 모든 활동을 캡처하기로 결정합니다.

DB2 활동 DB2ACTIVITIES가 활성화되었습니다. 관리자는 WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 함수를 사용하여 이 스토어드 프로시저 호출에 대한 응용프로그램 핸들, 작업 단위(UOW) ID 및 활동 ID를 찾아냅니다. 활동이 응용프로그램 핸들 1, 작업 ID 단위 2 및 활동 ID 3으로 활동을 식별한다고 가정하고 관리자가 이제 WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS에 대한 호출을 다음과 같이 발행할 수 있습니다.

```
CALL WLM_CAPTURE_ACTIVITY_IN_PROGRESS(1,2,3)
```

프로시저가 완료되면 관리자는 다음 테이블 함수를 사용하여 시간을 소비한 활동의 위치를 찾아낼 수 있습니다. 함수는 DB2ACTIVITIES 이벤트 모니터에서 정보를 검색합니다.

```
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, CHILDESTMT.STMT_TEXT, CHILD.ACT_EXEC_TIME
   FROM RAH PARENT, ACTIVITY_DB2ACTIVITIES CHILD,
        ACTIVITYSTMT_DB2ACTIVITIES CHILDESTMT
  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

```

다음 샘플 쿼리는 테이블 함수를 사용합니다.

```

    SELECT * FROM TABLE(SHOWCAPTUREDACTIVITY(1, 2, 3))
    AS ACTS ORDER BY UOW_ID, ACTIVITY_ID

```

## 사용 시 참고사항

활성 활동 이벤트 모니터가 없으면 SQLSTATE 01H53이 포함된 SQL1633W을 리턴합니다.

활동 정보는 활동의 코디네이터 파티션에서만 수집됩니다.

---

## WLM\_COLLECT\_STATS - 워크로드 관리 통계 수집 및 재설정

WLM\_COLLECT\_STATS 프로시저는 서비스 클래스, 워크로드, 작업 클래스 및 임계값 큐에 대한 통계를 수집하고 수집된 통계를 통계 이벤트 모니터에 기록합니다. 프로시저는 서비스 클래스, 워크로드, 작업 클래스 및 임계값 큐에 대한 통계를 재설정하기도 합니다. 사용 중인 통계 이벤트 모니터가 없는 경우 프로시저는 통계를 재설정만 합니다.

### 구문

▶▶—WLM\_COLLECT\_STATS—(—)—————▶▶

스키마는 SYSPROC입니다.

### 권한 부여

WLM\_COLLECT\_STATS 프로시저에 대한 EXECUTE 특권

### 예:

예 1: WLM\_COLLECT\_STATS를 호출하여 통계를 수집하여 재설정합니다.

```
CALL WLM_COLLECT_STATS()
```

이 쿼리의 출력 예는 다음과 같습니다.

리턴 상태 = 0

예 2: WLM\_COLLECT\_STATS를 호출하여 다른 호출이 진행 중에 통계를 수집하여 재설정합니다.

```
CALL WLM_COLLECT_STATS()
```

이 쿼리의 출력 예는 다음과 같습니다.

```
SQL1632W 다른 수집 및 재설정 통계 요청이 이미 진행 중이므로  
이 수집 및 재설정 통계 요청이 무시되었습니다.
```

## 사용 시 참고사항

WLM\_COLLECT\_STATS 프로시저는 **wlm\_collect\_int** 데이터베이스 구성 매개변수로 정의된 간격에서 자동으로 발생하는 동일한 수집 조작(활성 통계 이벤트 모니터에 통계 전송) 및 재설정 조작을 수행합니다.

다른 수집 및 재설정 요청이 진행 중인 동안 프로시저를 호출하려면(예를 들어, 다른 스포시저 호출이 실행 중이거나 자동 수집이 수행 중인 경우) SQLSTATE 01H53의 SQL1632W가 리턴되고 새 요청은 무시됩니다.

WLM\_COLLECT\_STATS 프로시저만 수집 및 재설정 프로세스를 시작합니다. 모든 통계가 사용 중인 통계 이벤트 모니터에 작성되기 전에 프리시저가 호출자에게 리턴될 수도 있습니다. 통계 수집 및 재설정이 얼마나 빨리 발생하는지에 따라 WLM\_COLLECT\_STATS 프로시저 호출(그 자체가 활동)이 이전 수집 간격 또는 바로 전에 시작된 새 수집 간격에 대한 통계로 계산됩니다.

---

## WLM\_GET\_QUEUE\_STATS 테이블 함수 - 임계값 큐 통계 리턴

WLM\_GET\_QUEUE\_STATS 함수는 모든 활성 파티션에서 하나 이상의 임계값 큐에 대한 기본 통계를 리턴합니다. 이 함수는 각 임계값 큐의 통계를 한 행 리턴합니다.

### 구문

```
►►—WLM_GET_QUEUE_STATS—(—threshold_predicate—,—threshold_domain—,——————►  
►—threshold_name—,—threshold_id—)——————►►
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

*threshold\_predicate*

임계값 술어를 지정하는 VARCHAR 유형의 입력 인수입니다. 가능한 값은 다음과 같습니다.

**CONCDBC**

동시 데이터베이스 코디네이터 활동 임계값

### *DBCONN*

전체 데이터베이스 파티션 연결 임계값

### *SCCONN*

전체 서비스 클래스 파티션 연결 임계값

인수가 널(NULL) 또는 비어 있는 문자열이면, 다른 기준에 만족하는 모든 임계값에 대해 데이터가 리턴됩니다.

*threshold\_predicate* 값은 SYSCAT.THRESHOLDS 뷰의 THRESHOLDPREDICATE 컬럼의 값과 일치합니다.

### *threshold\_domain*

임계값 도메인을 지정하는 VARCHAR 유형의 입력 인수입니다. 가능한 값은 다음과 같습니다.

*DB* 데이터베이스

*SB* 서비스 서브클래스

*SP* 서비스 수퍼 클래스

*WA* 작업 조치 세트

인수가 널(NULL) 또는 비어 있는 문자열이면, 다른 기준에 만족하는 모든 임계값에 대해 데이터가 리턴됩니다.

*threshold\_domain* 값은 SYSCAT.THRESHOLDS 뷰의 DOMAIN 컬럼의 값과 일치합니다.

### *threshold\_name*

임계값 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수입니다. 인수가 널(NULL) 또는 비어 있는 문자열이면, 다른 기준에 만족하는 모든 임계값에 대해 데이터가 리턴됩니다. *threshold\_name* 값은 SYSCAT.THRESHOLDS 뷰의 THRESHOLDNAME 컬럼의 값과 일치합니다.

### *threshold\_id*

임계값 ID를 지정하는 INTEGER 유형의 입력 인수입니다. 인수가 널(NULL) 또는 -1인 경우 다른 기준에 맞는 모든 임계값에 대해 데이터가 리턴됩니다.

*threshold\_id* 값은 SYSCAT.THRESHOLDS 뷰의 THRESHOLDID 컬럼의 값과 일치합니다.

## 권한 부여

WLM\_GET\_QUEUE\_STATS 함수에 대한 EXECUTE 특권

예 :

다음 쿼리는 모든 파티션에서 시스템의 모든 큐에 대한 기본 통계를 표시합니다.

```
SELECT substr(THRESHOLD_NAME, 1, 6) THRESHNAME,
       THRESHOLD_PREDICATE,
       THRESHOLD_DOMAIN,
       DBPARTITIONNUM PART,
       QUEUE_SIZE_TOP,
       QUEUE_TIME_TOTAL,
       QUEUE_ASSIGNMENTS_TOTAL QUEUE_ASSIGN
FROM table(WLM_GET_QUEUE_STATS('',' ', '-1)) as QSTATS
```

샘플 출력은 다음과 같습니다.

```
THRESHNAME THRESHOLD_PREDICATE THRESHOLD_DOMAIN ...
-----
LIMIT1     CONCDBC                      DB                      ...
LIMIT2     SCCONN                       SP                      ...
LIMIT3     DBCONN                       DB                      ...

... PART QUEUE_SIZE_TOP QUEUE_TIME_TOTAL QUEUE_ASSIGN
... -----
... 0         12         1238540         734
... 0         4          741249         24
... 0         7          412785         128
```

## 사용법 참고

함수는 큐(파티션) 또는 파티션(하나 이상의 큐)에서는 데이터 집계를 수행하지 않습니다. 그렇지만 SQL 쿼리를 사용하여 이전 예에 표시된 대로 데이터를 수집할 수는 있습니다.

## 리턴되는 정보

표 202. WLM\_GET\_QUEUE\_STATS에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
THRESHOLD_PREDICATE	VARCHAR(27)	이 큐를 담당하는 임계값의 임계값 술어. 가능한 값은 다음과 같습니다.  <i>CONCDBC</i> 동시 데이터베이스 코디네이터 활동 임계값  <i>DBCONN</i> 전체 데이터베이스 파티션 연결 임계값  <i>SCCONN</i> 전체 서비스 클래스 파티션 연결 임계값  임계값 술어 값은 SYSCAT.THRESHOLDS 뷰의 THRESHOLDPREDICATE 컬럼의 값과 일치합니다.

표 202. WLM\_GET\_QUEUE\_STATS에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
THRESHOLD_DOMAIN	VARCHAR(18)	이 큐를 담당하는 임계값의 도메인. 가능한 값은 다음과 같습니다.  DB     데이터베이스 SB     서비스 서브클래스 SP     서비스 수퍼 클래스 WA     작업 조치 세트 임계값 도메인 값은 SYSCAT.THRESHOLDS 뷰의 DOMAIN 컬럼의 값과 일치합니다.
THRESHOLD_NAME	VARCHAR(128)	이 큐를 담당하는 임계값의 고유 이름. 임계값 이름 값은 SYSCAT.THRESHOLDS 뷰의 THRESHOLDNAME 컬럼의 값과 일치합니다.
THRESHOLD_ID	INTEGER	이 큐를 담당하는 임계값의 고유 ID. 임계값 ID 값은 SYSCAT.THRESHOLDS 뷰의 THRESHOLDID 컬럼의 값과 일치합니다.
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호입니다.
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	이 큐를 담당하는 임계값의 도메인인 서비스 수퍼 클래스의 이름. 임계값의 도메인이 서비스 수퍼 클래스 또는 서비스 서브클래스가 아닌 경우에는 컬럼 값이 널(NULL)입니다.
SERVICE_SUBCLASS_NAME	VARCHAR(128)	이 큐를 담당하는 임계값의 도메인인 서비스 서브클래스의 이름. 임계값의 도메인이 서비스 서브클래스가 아닌 경우에는 컬럼 값이 널(NULL)입니다.
WORK_ACTION_SET_NAME	VARCHAR(128)	이 큐를 담당하는 임계값의 도메인인 작업 조치 세트의 이름. 임계값의 도메인이 작업 조치 세트가 아닌 경우에는 컬럼 값이 널(NULL)입니다.
WORK_CLASS_NAME	VARCHAR(128)	이 큐를 담당하는 임계값의 도메인인 작업 조치 세트에 속한 작업 조치의 작업 클래스 이름. 임계값의 도메인이 작업 조치 세트가 아닌 경우에는 컬럼 값이 널(NULL)입니다.
WORKLOAD_NAME	VARCHAR(128)	이 큐를 담당하는 임계값의 도메인인 워크로드의 이름. 임계값의 도메인이 워크로드가 아닌 경우에는 컬럼 값이 널(NULL)입니다.



표 202. WLM\_GET\_QUEUE\_STATS에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
LAST_RESET	TIMESTAMP	통계가 마지막으로 재설정된 시간입니다. 다음과 같은 4개의 이벤트가 통계 재설정을 트리거합니다. <ul style="list-style-type: none"> <li>• WLM_COLLECT_STATS 프로시저를 호출합니다.</li> <li>• <code>wlm_collect_int</code> 구성 매개변수로 인해 수집 및 재설정이 수행됩니다.</li> <li>• 데이터베이스를 재할성화합니다.</li> <li>• 큐 통계를 보고 중인 임계값을 수정하고 변경사항을 커밋합니다.</li> </ul> LAST_RESET 시간소인은 로컬 시간입니다.
QUEUE_SIZE_TOP	INTEGER	마지막 재설정 이후 큐에서 가장 많은 연결 또는 활동 수
QUEUE_TIME_TOTAL	BIGINT	마지막 재설정 이후 이 큐에 저장된 모든 연결 또는 활동의 큐에 소비된 시간의 합계. 단위는 밀리초입니다.
QUEUE_ASSIGNMENTS_TOTAL	BIGINT	마지막 재설정 이후 이 큐에 지정된 연결 또는 활동 수
QUEUE_SIZE_CURRENT	INTEGER	큐의 연결 또는 활동 수
QUEUE_TIME_LATEST	BIGINT	큐를 떠나기 위해 마지막 연결 또는 활동이 큐에 소비한 시간. 단위는 밀리초입니다.
QUEUE_EXIT_TIME_LATEST	TIMESTAMP	마지막 연결 또는 활동이 큐를 떠난 시간
THRESHOLD_CURRENT_CONCURRENCY	INTEGER	임계값에 따라 현재 실행 중인 연결 또는 활동 수.
THRESHOLD_MAX_CONCURRENCY	INTEGER	임계값이 동시에 실행되도록 허용한 최대 연결 또는 활동 수.

## WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 테이블 함수 - 서비스 클래스에서 실행 중인 에이전트 나열

WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 함수는 지정한 서비스 클래스에서 또는 지정한 응용프로그램 대신 실행 중인 지정한 파티션에서 에이전트, 분리 모드 프로세스(db2fmp 프로세스) 및 시스템 엔티티 목록을 리턴합니다. 시스템 엔티티는 페이지 클리너와 프리페처와 같이 비에이전트 스레드 및 프로세스입니다.

### 구문

```

▶▶—WLM_GET_SERVICE_CLASS_AGENTS_V97—(—service_superclass_name—,—————▶
▶—service_subclass_name—,—application_handle—,—dbpartitionnum—)————▶▶
    
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *service\_superclass\_name*

현재 연결된 데이터베이스에서 서비스 슈퍼 클래스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수입니다. 인수가 널(NULL) 또는 비어 있는 문자열인 경우 데이터베이스의 모든 슈퍼 클래스에 대해 데이터가 검색됩니다.

### *service\_subclass\_name*

슈퍼 클래스 내의 특정 서브클래스를 참조하는 VARCHAR(128) 유형의 입력 인수. 인수가 널(NULL) 또는 비어 있는 문자열인 경우 데이터베이스의 모든 서브클래스에 대해 데이터가 검색됩니다.

### *application\_handle*

리턴되는 에이전트 정보의 응용프로그램 핸들을 지정하는 VARCHAR(255) 유형의 입력 인수입니다. 인수가 널(NULL)인 경우 데이터베이스의 모든 응용프로그램에 대해 데이터가 검색됩니다. 응용프로그램 핸들 0은 시스템 엔티티만 리턴합니다.

### *dbpartitionnum*

현재 연결된 데이터베이스와 동일한 인스턴스의 파티션 번호를 지정하는 INTEGER 유형의 입력 인수입니다. 현재 데이터베이스 파티션으로 1을 지정하거나 모든 데이터베이스 파티션으로 -2를 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

## 권한 부여

WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 함수의 EXECUTE 특권.

예 :

### 예 1

다음 쿼리는 모든 데이터베이스 파티션에 대해 응용프로그램 핸들 1과 연관된 에이전트 목록을 리턴합니다. LIST APPLICATIONS 명령이나 WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97 테이블 함수를 사용하여 응용프로그램 핸들을 판별할 수 있습니다.

```
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
```

샘플 출력은 다음과 같습니다.

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

출력은 UOW ID 1 및 활동 ID 5의 활동을 대신하는 파티션 0의 코디네이터 에이전트와 서브에이전트 및 파티션 1의 서브에이전트를 보여줍니다. 값이 COORDINATOR인 AGENTTYPE 컬럼의 경우 REQTYPE(기본 또는 초기 요청 유형을 나타냄) 컬럼의 값은 FETCH입니다. 즉, 요청 유형이 코디네이터 에이전트의 페치 요청임을 나타냅니다.

## 예 2

다음 쿼리로 대기 중인 에이전트 잠금을 판별합니다.

```
db2 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
```

샘플 출력은 다음과 같습니다.

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	0200050000000000000000000054
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	-

21개의 레코드가 선택되었습니다.

이후에 동일한 쿼리를 사용하여 WLM 임계값이 에이전트를 큐에 대기시켰음을 보여줍니다.

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	-
WLM_QUEUE	WAIT	IDLE	MYCONCDBCkoordth
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	-

21개의 레코드가 선택되었습니다.

## 사용 시 참고사항

매개변수는 AND되는 효과가 있습니다. 즉 서비스 수퍼 클래스 SUP\_A와 서브클래스 SUB\_B(SUB\_B는 SUP\_A의 서브클래스가 아님)와 같이 충돌하는 입력 매개변수를 지정하면 행이 리턴되지 않습니다.

## 리턴된 정보

표 203. WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97에서 리턴된 정보

컬럼 이름	데이터 유형	설명
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	이 레코드가 수집된 서비스 수퍼 클래스의 이름.
SERVICE_SUBCLASS_NAME	VARCHAR (128)	이 레코드가 수집된 서비스 서브클래스의 이름.
APPLICATION_HANDLE	BIGINT	응용프로그램에 대해 시스템 전체에서 사용되는 고유 ID. 단일 파티션의 데이터베이스에서 이 ID는 16비트의 카운터로 구성됩니다. 다중 파티션의 데이터베이스에서 이 ID는 16비트 카운터로 병합된 조정 파티션 번호로 구성됩니다. 또한 이 ID는 응용프로그램이 보조 연결을 작성하는 모든 파티션에서 동일합니다.
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호.
ENTITY	VARCHAR(32)	다음 값 중 하나입니다. <ul style="list-style-type: none"> <li>엔티티 유형이 에이전트인 경우 값은 db2agent입니다.</li> <li>엔티티 유형이 분리 모드 프로세스인 경우 값은 db2fmp(pid)이며 여기서 pid는 분리 모드 프로세스의 프로세스 ID입니다.</li> <li>그 이외 경우 값은 시스템 엔티티 이름입니다.</li> </ul>
WORKLOAD_NAME	VARCHAR (128)	이 레코드가 수집된 워크로드의 이름.
WORKLOAD_OCCURRENCE_ID	INTEGER	워크로드 어커런스의 ID. 코디네이터 데이터베이스 파티션 번호와 워크로드 이름으로 연결되어 있지 않으면 이 ID는 워크로드 어커런스를 고유하게 식별하지 않습니다.
UOW_ID	INTEGER	이 활동이 시작된 작업 단위(UOW)의 고유 ID.
ACTIVITY_ID	INTEGER	작업 단위(UOW)내의 고유 활동 ID
PARENT_UOW_ID	INTEGER	활동의 상위 활동이 시작된 작업 단위(UOW)의 고유 ID. 활동의 상위가 없는 경우에는 컬럼 값이 널(NULL)입니다.

표 203. WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
PARENT_ACTIVITY_ID	INTEGER	ID가 activity_id와 동일한 활동 상위의 작업 단위(UOW)내에서 고유 활동 ID. 활동에 상위이 없는 경우에는 이 컬럼 값이 널(NULL)입니다.
AGENT_TID	BIGINT	에이전트 또는 시스템 엔티티의 스레드 ID. 이 ID를 사용할 수 없는 경우에는 컬럼 값이 널(NULL)입니다.
AGENT_TYPE	VARCHAR(32)	에이전트 유형. 에이전트 유형은 다음과 같습니다. <ul style="list-style-type: none"> <li>• COORDINATOR</li> <li>• OTHER</li> <li>• PDBSUBAGENT</li> <li>• SMPSUBAGENT</li> </ul> 값이 COORDINATOR인 경우 에이전트 ID는 집중기(concentrator) 환경에서 변경될 수 있습니다.
SMP_COORDINATOR	INTEGER	에이전트가 SMP 코디네이터인지 여부를 나타내며 코디네이터인 경우 1이고 아닌 경우에는 0입니다.
AGENT_SUBTYPE	VARCHAR(32)	에이전트 부속 유형. 가능한 부속 유형은 다음과 같습니다. <ul style="list-style-type: none"> <li>• DSS</li> <li>• OTHER</li> <li>• RPC</li> <li>• SMP</li> </ul>
AGENT_STATE	VARCHAR(32)	에이전트가 연관 또는 활성 상태인지 여부를 나타냅니다. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• ASSOCIATED</li> <li>• ACTIVE</li> </ul>
EVENT_TYPE	VARCHAR(32)	이 에이전트에서 마지막으로 처리하는 이벤트 유형. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• ACQUIRE</li> <li>• PROCESS</li> <li>• WAIT</li> </ul>

표 203. WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
EVENT_OBJECT	VARCHAR(32)	이 에이전트에서 마지막으로 처리한 이벤트 오브젝트. 가능한 값은 다음과 같습니다. <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>
EVENT_STATE	VARCHAR(32)	이 에이전트에서 마지막으로 처리한 이벤트 상태. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• EXECUTING</li> <li>• IDLE</li> </ul>
REQUEST_ID	VARCHAR (64)	요청 ID. 이 값은 <i>application_handle</i> 값의 조합에서만 고유합니다. 이 조합을 사용하여 오래 걸리는 임의의 요청과 다중 요청을 구분할 수 있습니다(예: 임의의 장기 폐치와 다중 폐치).

표 203. WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
REQUEST_TYPE	VARCHAR(32)	<p>요청 유형. 가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• 코디네이터 에이전트의 경우                             <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>, 여기서 <i>number</i>는 내부 상수 값</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_SUBTYPE이 DSS 또는 SMP인 서브 에이전트                             <ul style="list-style-type: none"> <li>- 서브섹션 번호가 0이 아닌 경우 서브섹션 번호 양식은 SUBSECTION:<i>subsection number</i>이며 그 이외의 경우에는 NULL이 리턴됩니다.</li> </ul> </li> </ul>

표 203. WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
REQUEST_TYPE (계속)	VARCHAR(32)	<ul style="list-style-type: none"> <li>• AGENT_SUBTYPE이 RPC인 서브 에이전트                             <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>• SUBTYPE이 OTHER인 서브 에이전트                             <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>, 여기서 <i>number</i>는 내부 상수 값</li> <li>- INTERRUPT</li> <li>- NOOP(요청이 없는 경우)</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	ID가 activity_id인 활동의 중첩 레벨. 중첩 레벨은 이 활동이 최상위 활동에 중첩된 깊이입니다.
INVOCATION_ID	INTEGER	활동의 특정 호출을 동일한 중첩 레벨의 다른 호출과 구별하는 호출 ID
ROUTINE_ID	INTEGER	루틴의 고유 ID. 활동이 루틴의 파트가 아닌 경우에는 이 컬럼 값이 널(NULL)입니다.
EVENT_OBJECT_NAME	VARCHAR (1024)	이벤트 오브젝트 이름. EVENT_OBJECT 값이 LOCK인 경우 해당 컬럼 값은 에이전트가 대기 중인 잠금 이름입니다. EVENT_OBJECT 값이 WLM_QUEUE인 경우 컬럼 값은 에이전트가 큐에서 대기 중인 WLM 일계값 이름입니다. 그 이외의 경우 값은 NULL입니다.
APPLICATION_NAME	VARCHAR (128)	나중에 사용하도록 예약됩니다.



표 203. WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
APPLICATION_ID	VARCHAR (128)	나중에 사용하도록 예약됩니다.
CLIENT_PID	BIGINT	나중에 사용하도록 예약됩니다.
SESSION_AUTH_ID	VARCHAR (128)	나중에 사용하도록 예약됩니다.
REQUEST_START_TIME	TIMESTAMP	나중에 사용하도록 예약됩니다.
AGENT_STATE_LAST_UPDATE_TIME	TIMESTAMP	나중에 사용하도록 예약됩니다.
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	에이전트가 작업 중인 섹션을 고유하게 식별하는 데이터 서버에서 생성된 2진 토큰. 실행 가능한 ID를 다른 모니터링 인터페이스의 입력으로 사용하여 섹션에 대한 데이터를 가져올 수 있습니다. 에이전트가 섹션에서 작업 중이지 않은 경우에는 널(NULL) 값이 리턴됩니다.

## WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97 - 워크로드 어커런스 나열

WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97 함수는 특정 파티션의 지정한 서비스 클래스에서 실행되는 전체 워크로드 어커런스 목록을 나열합니다. 워크로드 어커런스는 속성이 워크로드 정의에 일치하는 특정 데이터베이스 연결로 워크로드와 연관되거나 워크로드에 지정됩니다.

### 구문

```
►►—WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES_V97—(—service_superclass_name—, —————►
►—service_subclass_name—, —dbpartitionnum—)—————►►
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### service\_superclass\_name

현재 연결된 데이터베이스에서 서비스 슈퍼 클래스 이름을 지정하는

VARCHAR

(128) 유형의 입력 인수입니다. 인수가 널(null)이거나 비어 있는 문자열이면 기타 매개변수 값과 일치하는 데이터베이스의 모든 슈퍼 클래스에 대해 데이터가 검색됩니다.

#### service\_subclass\_name

현재 연결된 데이터베이스에서 서비스 서브클래스 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수입니다. 인수가 널(null)이거나 비어 있는 문자열이면 기타 매개변수 값과 일치하는 데이터베이스의 모든 서브클래스에 대해 데이터가 검색됩니다.

### *dbpartitionnum*

현재 연결된 데이터베이스와 동일한 인스턴스에 있는 파티션 수를 지정하는 INTEGER 유형의 입력 인수입니다. 현재 데이터베이스 파티션으로 1을 지정하거나 모든 데이터베이스 파티션으로 -2를 지정하십시오. 널(NULL) 값이 지정된 경우, -1이 내재적으로 설정됩니다.

### 권한 부여

WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97 함수에 대한 EXECUTE 특권

### 예 :

관리자가 시스템 전반에서 실행 중인 워크로드 어커런스를 확인하려는 경우 관리자는 *service\_superclass\_name* 및 *service\_subclass\_name*에 대해 널(NULL) 값 또는 비어 있는 문자열을 지정하고 *dbpartitionnum*에 대해 -2를 지정하여 WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97 함수를 호출할 수 있습니다.

```
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
```

시스템에 네 개의 데이터베이스 파티션이 있고 지금 두 개의 워크로드가 실행 중인 경우 이전 쿼리로 인해 다음과 같은 결과가 작성될 수 있습니다.

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
... 1 SYSDEFAULTUSERWORKLOAD 1
... 2 SYSDEFAULTUSERWORKLOAD 2

```

## 사용법 참고

매개변수는 AND되는 효과가 있습니다. 즉 서비스 수퍼 클래스 SUP\_A와 서브클래스 SUB\_B(SUB\_B는 SUP\_A의 서브클래스가 아님)와 같이 충돌하는 입력 매개변수를 지정하면 행이 리턴되지 않습니다.

주: 워크로드 어커런스에 보고된 통계(예: coord\_act\_completed\_total)는 해당 워크로드 통계와 조합되는 경우 각 작업 단위(UOW) 시작 시에 재설정됩니다.

## 리턴되는 정보

표 204. WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	이 레코드가 수집된 서비스 수퍼 클래스의 이름.
SERVICE_SUBCLASS_NAME	VARCHAR (128)	이 레코드가 수집된 서비스 서브클래스의 이름.
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호입니다.
COORD_PARTITION_NUM	SMALLINT	지정한 워크로드 어커런스의 코디네이터 파티션에 대한 파티션 번호.
APPLICATION_HANDLE	BIGINT	응용프로그램에 대해 시스템 전체에서 사용되는 고유 ID. 단일 파티션의 데이터베이스에서 이 ID는 16비트의 카운터로 구성됩니다. 다중 파티션의 데이터베이스에서 이 ID는 16비트 카운터로 병합된 조정 파티션 번호로 구성됩니다. 또한 이 ID는 응용프로그램이 보조 연결을 작성하는 모든 파티션에서 동일합니다.
WORKLOAD_NAME	VARCHAR (128)	이 레코드가 수집된 워크로드의 이름.
WORKLOAD_OCCURRENCE_ID	INTEGER	워크로드 어커런스의 ID. 코디네이터 데이터베이스 파티션 번호와 워크로드 이름으로 연결되어 있지 않으면 이 ID는 워크로드 어커런스를 고유하게 식별하지 않습니다.

표 204. WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
WORKLOAD_OCCURRENCE_STATE	VARCHAR(32)	<p>워크로드 어커런스 상태. 값은 다음과 같습니다.</p> <p><i>DECOUPLED</i> 워크로드 어커런스에는 지정된 코디네이터 에이전트가 없습니다(콘센트레이터 case).</p> <p><i>DISCONNECTPEND</i> 데이터베이스에서 워크로드 어커런스 연결 해제 중입니다.</p> <p><i>FORCED</i> 워크로드 어커런스가 데이터베이스를 강제로 종료했습니다.</p> <p><i>INTERRUPTED</i> 워크로드 어커런스가 인터럽트되었습니다.</p> <p><i>QUEUED</i> Query Patroller 또는 워크로드 관리 큐잉 임계값으로 워크로드 어커런스 코디네이터 에이전트가 큐에 대기되었습니다. 데이터베이스 파티셔닝 기능(DPF) 환경에서 이 상태는 임계값 티켓을 가져오기 위해 코디네이터 에이전트가 카탈로그 파티션에 대한 RPC를 작성했지만 아직 응답을 수신하지 않았음을 나타냅니다.</p> <p><i>TRANSIENT</i> 워크로드 어커런스가 서비스 수퍼 클래스에 맵핑되지 않았습니다.</p> <p><i>UOWEXEC</i> 워크로드 어커런스가 요청을 처리 중입니다.</p> <p><i>UOWWAIT</i> 워크로드 어커런스가 클라이언트에서 요청을 대기 중입니다.</p>
UOW_ID	INTEGER	이 워크로드 어커런스가 시작된 작업 단위(UOW)의 고유 ID입니다.
SYSTEM_AUTH_ID	VARCHAR(128)	워크로드 어커런스가 시스템에 삽입된 시스템 권한 부여 ID입니다.
SESSION_AUTH_ID	VARCHAR(128)	워크로드 어커런스가 시스템에 삽입된 세션 권한 부여 ID입니다.
APPLICATION_NAME	VARCHAR(128)	이 워크로드 어커런스를 작성한 응용프로그램 이름입니다.
CLIENT_WRKSTNNAME	VARCHAR(255)	이 워크로드 어커런스에 대한 CLIENT_WRKSTNNAME 특수 레지스터의 현재 값입니다.

표 204. WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
CLIENT_ACCTNG	VARCHAR(255)	이 워크로드 어커런스에 대한 CLIENT_ACCTNG 특수 레지스터의 현재 값입니다.
CLIENT_USER	VARCHAR(255)	이 워크로드 어커런스에 대한 CLIENT_USERID 특수 레지스터의 현재 값입니다.
CLIENT_APPLNAME	VARCHAR(255)	이 워크로드 어커런스에 대한 CLIENT_APPLNAME 특수 레지스터의 현재 값입니다.
COORD_ACT_COMPLETED_TOTAL	INTEGER	이 워크로드 어커런스의 현재 작업 단위(UOW)에서 현재까지 완료된 모든 중첩 레벨의 코디네이터 활동 수입니다. 이 통계는 이 워크로드 어커런스의 활동이 완료되어 각 작업 단위(UOW) 시작에서 재설정될 때마다 갱신됩니다.
COORD_ACT_ABORTED_TOTAL	INTEGER	이 워크로드 어커런스의 현재 작업 단위(UOW)에서 현재까지 중단된 코디네이터 활동 수입니다. 이 통계는 이 워크로드 어커런스의 활동이 중단되어 각 작업 단위(UOW) 시작에서 재설정될 때마다 갱신됩니다.
COORD_ACT_REJECTED_TOTAL	INTEGER	이 워크로드 어커런스의 현재 작업 단위(UOW)에서 현재까지 거부된 코디네이터 활동 수입니다. 활동은 실행 금지 작업 조치 또는 예측적 임계값으로 인해 실행되지 않은 경우 거부됨으로 간주됩니다. 이 통계는 이 워크로드 어커런스의 활동이 거부되어 각 작업 단위(UOW) 시작에서 재설정될 때마다 갱신됩니다.
CONCURRENT_ACT_TOP	INTEGER	현재 작업 단위(UOW)에서 이 워크로드 어커런스에 도달한 실행 상태(유티 및 대기 중 포함) 또는 큐에 대기된 상태의 모든 중첩 레벨에서 가장 높은 동시 활동 수. 이 통계는 각 작업 단위가 시작될 때 재설정됩니다.
ADDRESS	VARCHAR(255)	이 워크로드 어커런스를 작성한 IP 주소 또는 보안 도메인 이름입니다. 보안 도메인 이름은 IP 주소로 변환되어 표시됩니다.

## WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97

### 테이블 함수 - 서비스 서브클래스의 통계 리턴

WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 함수는 하나 이상의 서비스 서브클래스에 대한 기본 통계를 리턴합니다.

#### 구문

```

▶▶—WLM_GET_SERVICE_SUBCLASS_STATS_V97—(—service_superclass_name—,—————▶
▶—service_subclass_name—,—dbpartitionnum—)—————▶▶

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *service\_superclass\_name*

현재 연결된 데이터베이스에서 서비스 슈퍼 클래스 이름을 지정하는

VARCHAR

(128) 유형의 입력 인수입니다. 인수가 널(null)이거나 빈 문자열이면 데이터베이스의 모든 슈퍼 클래스에 대해 데이터가 검색됩니다.

### *service\_subclass\_name*

현재 연결된 데이터베이스에서 서비스 서브클래스 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수입니다. 인수가 널(null)이거나 빈 문자열이면 데이터베이스의 모든 서브클래스에 대해 데이터가 검색됩니다.

### *dbpartitionnum*

현재 연결된 데이터베이스와 동일한 인스턴스에 있는 유효한 파티션 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션으로 1을 지정하거나 모든 데이터베이스 파티션으로 -2를 지정하십시오. 널(NULL) 값이 지정된 경우, -1이 내재적으로 설정됩니다.

## 권한 부여

WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 함수에서의 EXECUTE 특권.

### 예:

예 1: 모든 활동은 실행되기 전에 DB2 서비스 클래스에 맵핑되어야 하기 때문에 서비스 클래스 통계 테이블 함수를 사용하고 모든 파티션에서 모든 서비스 클래스를 쿼리하여 시스템의 전역 상태를 모니터링할 수 있습니다. 다음 예에서 널(null) 값이 *service\_superclass\_name* 및 *service\_subclass\_name*에 대해 전달되어 모든 서비스 클래스에 대한 통계가 리턴되며 *dbpartitionnum*에 대해서 -2 값이 지정되어 모든 파티션에 대한 통계가 리턴됩니다.

```
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
```

명령문은 다음 샘플 출력에서 표시된 대로 초 단위로 평균 활동 수명 및 표준 편차와 같은 서비스 클래스 통계를 리턴합니다.

```

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

```

예 2: 동일한 테이블 함수는 각 파티션의 서비스 클래스에서 실행 중인 코디네이터 활동에 대한 평균 동시성에 가장 높은 값을 지정할 수도 있습니다.

```

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

```

샘플 출력은 다음과 같습니다.

```

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

```

이 테이블 함수 출력에서 평균 실행 시간 및 활동 수를 점검하여 특정 데이터베이스의 각 파티션 로드와 대한 높은 수준의 뷰를 확인할 수 있습니다. 이 테이블 함수가 리턴하는 높은 수준의 게이지(gauge)에서 모든 중요한 변형은 시스템 로드의 변형을 나타낼 수도 있습니다.

예 3: 활동에서 REMAP ACTIVITY TO 조치가 포함된 임계값을 사용하는 경우 해당 활동은 수명 기간 동안 둘 이상의 서비스 클래스에서 수행될 수 있습니다. 다음 예에 표시된 대로 ACTIVITIES\_MAPPED\_IN 및 ACTIVITIES\_MAPPED\_OUT 컬럼을 확인하여 서비스 클래스 세트를 통해 전달된 활동 수를 판별할 수 있습니다.

```

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
```

샘플 출력은 다음과 같습니다.

SUPERCLASS_NAME	SUBCLASS_NAME	MAPPED_IN	MAPPED_OUT
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	0
SUPERCLASS1	SYSDEFAULTSUBCLASS	0	0
SUPERCLASS1	SUBCLASS1	0	7
SUPERCLASS1	SUBCLASS2	7	0

## 사용 시 참고사항

일부 통계는 해당 서비스 서브클래스에 대해 COLLECT AGGREGATE ACTIVITY DATA 및 COLLECT AGGREGATE REQUEST DATA 매개변수를 NONE 이외의 값으로 설정한 경우에만 리턴됩니다.

WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 테이블 함수는 서비스 서브클래스와 파티션 당 데이터의 한 행을 리턴합니다. 함수는 서비스 클래스(파티션) 또는 파티션(하나 이상의 서비스 클래스)에서는 데이터 집계를 수행하지 않습니다. 그렇지만 SQL 쿼리를 사용하여 데이터를 수집할 수는 있습니다.

매개변수는 AND되는 효과가 있습니다. 즉, 충돌하는 입력 매개변수(예: 슈퍼 클래스 SUPA 및 SUBB가 SUPA의 서브클래스가 아닌 SUBB 서브클래스)를 지정하면 행이 리턴되지 않습니다.

## 리턴된 정보

표 205. WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	이 레코드가 수집된 서비스 슈퍼 클래스의 이름.
SERVICE_SUBCLASS_NAME	VARCHAR (128)	이 레코드가 수집된 서비스 서브클래스의 이름.
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호.



표 205. WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
LAST_RESET	TIMESTAMP	<p>통계가 마지막으로 재설정된 시간. 다음과 같은 4개의 이벤트가 통계 재설정을 트리거합니다.</p> <ul style="list-style-type: none"> <li>• WLM_COLLECT_STATS 프로시저를 호출합니다.</li> <li>• wlm_collect_int 구성 매개변수로 인해 수집 및 재설정이 수행됩니다.</li> <li>• 데이터베이스를 재활성화합니다.</li> <li>• 통계를 보고 중인 서비스 서브클래스를 수정하고 변경사항을 커밋합니다.</li> </ul> <p>LAST_RESET 시간소인은 로컬 시간입니다.</p>
COORD_ACT_COMPLETED_TOTAL	BIGINT	<p>마지막 재설정 이후에 제출되어 제대로 완료된 전체 코디네이터 활동 수. 이 계수는 각 활동이 완료되면 갱신됩니다.</p> <p>활동을 다른 서비스 서브클래스에 다시 맵핑하면 해당 활동은 완료된 서브클래스의 총계만을 계산합니다.</p>
COORD_ACT_ABORTED_TOTAL	BIGINT	<p>마지막 재설정 이후에 제출되어 오류가 발생한 상태로 완료된 전체 코디네이터 활동 수. 이 계수는 각 활동이 중단될 때마다 갱신됩니다.</p> <p>활동을 다른 서비스 서브클래스에 다시 맵핑하면 해당 활동은 중단된 서브클래스의 총계만을 계산합니다.</p>
COORD_ACT_REJECTED_TOTAL	BIGINT	<p>마지막 재설정 이후에 제출되어 실행 전에 거부된 전체 코디네이터 활동 수. 실행 방지 작업 조치 또는 예측 임계값으로 활동이 실행되지 않는 경우에만 활동은 거부된 것으로 계산됩니다. 이 계수는 각 활동이 거부될 때 갱신됩니다.</p>
CONCURRENT_ACT_TOP	INTEGER	<p>이 서비스 서브클래스에 도달한 실행 상태에서 중첩된 레벨로의 가장 높은 동시 활동 수(유티 및 대기 포함).</p>

표 205. WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
COORD_ACT_LIFETIME_TOP	BIGINT	<p>모든 중첩 레벨에서 평가된 코디네이터 활동 수명의 상위 워터 마크(water mark)입니다. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정되면 컬럼 값은 널(NULL)입니다. 단위는 밀리초입니다.</p> <p>서비스 클래스에 다시 맵핑된 서브클래스가 포함된 경우에 이 통계를 효율적으로 사용하려면 동일한 다시 맵핑 임계값의 영향을 받는 기타 서브클래스의 상위 워터 마크(water mark)와 같이 서비스 서브클래스의 COORD_ACT_LIFETIME_TOP 상위 워터 마크를 집계해야 합니다. 서브클래스가 다른 서비스 서브클래스에 다시 맵핑된 후에 활동이 완료될 수 있기 때문에 이 값을 집계해야 합니다. 다시 맵핑되기 전에 활동이 다른 서비스 서브클래스에서 소요한 시간은 활동이 완료된 서비스 클래스에 대해서만 계산됩니다.</p>
COORD_ACT_LIFETIME_AVG	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 중첩 레벨 0에서의 코디네이터 활동에 대한 수명의 산술 평균. 내부적으로 트랙된 평균이 오버플로우되면 값 -2가 리턴됩니다. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정되면 컬럼 값은 널(NULL)입니다. 단위는 밀리초입니다.</p> <p>서비스 서브클래스의 COORD_ACT_LIFETIME_AVG 값은 서브클래스를 통해 전달되는 활동의 영향을 받지 않지만 완료되기 전에 다른 서브클래스에 다시 맵핑됩니다.</p>
COORD_ACT_LIFETIME_STDDEV	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 중첩 레벨 0에서의 코디네이터 활동에 대한 수명의 표준 편차. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정되면 컬럼 값은 널(NULL)입니다. 단위는 밀리초입니다.</p> <p>이 표준 편차는 코디네이터의 활동 수명 막대 그래프로 계산되며 막대 그래프가 데이터에 적합하게 제대로 크기 조정되지 않은 경우에는 정확하지 않을 수도 있습니다. 임의 값이 막대 그래프의 마지막 막대에 포함되면 -1이 값으로 리턴됩니다.</p> <p>서비스 서브클래스의 COORD_ACT_LIFETIME_STDDEV 값은 서비스 서브클래스를 통해 전달되는 활동에는 영향을 받지 않지만 완료되기 전에 다른 서브클래스에 다시 맵핑됩니다.</p>

표 205. WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
COORD_ACT_EXEC_TIME_AVG	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 중첩 레벨 0에서의 코디네이터 활동에 대한 실행 시간의 산술 평균. 내부적으로 트랙된 평균이 오버플로우되면 값 -2가 리턴됩니다. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정되면 컬럼 값은 널(NULL)입니다. 단위는 밀리초입니다.</p> <p>서비스 서브클래스의 평균 실행 시간은 서브클래스를 통해 전달되는 활동에는 영향을 받지 않지만 완료되기 전에 다른 서브클래스에 다시 맵핑됩니다.</p>
COORD_ACT_EXEC_TIME_STDDEV	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 중첩 레벨 0에서의 코디네이터 활동에 대한 실행 시간의 표준 편차. 단위는 밀리초입니다.</p> <p>이 표준 편차는 코디네이터의 활동 실행 시간 막대 그래프로 계산되며 막대 그래프가 데이터에 적합하게 제대로 크기 조정되지 않은 경우에는 정확하지 않을 수도 있습니다. 임의 값이 마지막 막대 그래프 바이너리에 포함되면 -1 값이 리턴됩니다.</p> <p>서비스 서브클래스의 실행 시간 표준 편차는 서브클래스를 통해 전달되는 활동에는 영향을 받지 않지만 완료되기 전에 다른 서브클래스에 다시 맵핑됩니다.</p>
COORD_ACT_QUEUE_TIME_AVG	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 중첩 레벨 0에서의 코디네이터 활동에 대한 큐 시간의 산술 평균. 내부적으로 트랙된 평균이 오버플로우되면 값 -2가 리턴됩니다. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정되면 컬럼 값은 널(NULL)입니다. 단위는 밀리초입니다.</p> <p>큐 시간 평균은 활동이 큐에 대기되는 서비스 서브클래스에 대해서만 계산됩니다.</p>

표 205. WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
COORD_ACT_QUEUE_TIME_STDDEV	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 중첩 레벨 0에서의 코디네이터 활동에 대한 큐 시간의 표준 편차. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정되면 컬럼 값은 널(NULL)입니다. 단위는 밀리초입니다.</p> <p>이 표준 편차는 코디네이터의 큐 시간 수명 막대 그래프로 계산되며 막대 그래프가 데이터에 적합하게 제대로 크기 조정되지 않은 경우에는 정확하지 않을 수도 있습니다. 임의 값이 막대 그래프의 마지막 막대에 포함되면 -1이 값으로 리턴됩니다.</p> <p>큐 시간 표준 편차는 활동이 큐에 대기되는 서비스 서브클래스에 대해서만 계산됩니다.</p>
NUM_REQUESTS_ACTIVE	BIGINT	이 테이블 함수가 실행되는 동안 서비스 서브클래스에서 실행 중인 요청 수.
NUM_REQUESTS_TOTAL	BIGINT	<p>마지막 재설정 이후 이 서비스 서브클래스에서 실행 완료된 요청 수. 이 완료 상태는 활동에서의 멤버십에 상관 없이 모든 요청에 적용됩니다. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정되면 컬럼 값은 널(NULL)입니다.</p> <p>서비스 서브클래스의 NUM_REQUESTS_TOTAL 값은 서비스 서브클래스를 통해 전달되는 요청에는 영향을 받지 않지만 해당 서비스 서브클래스에서 완료되지는 않습니다.</p>
REQUEST_EXEC_TIME_AVG	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 요청에 대한 실행 시간의 산술 평균. 단위는 밀리초입니다. 내부적으로 트랙된 평균이 오버플로우되면 값 -2가 리턴됩니다. 이 서비스 클래스에서 COLLECT AGGREGATE REQUEST DATA 매개변수가 NONE으로 설정되어 있으면 이 컬럼의 값은 NULL입니다.</p> <p>서비스 서브클래스의 실행 시간 평균은 서브클래스를 통해 전달되는 요청에 의해 영향을 받지 않지만 해당 서브클래스에서 완료되지는 않습니다.</p>

표 205. WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
REQUEST_EXEC_TIME_STDDEV	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 요청에 대한 실행 시간의 표준 편차. 단위는 밀리초입니다. 서비스 클래스에서 COLLECT AGGREGATE REQUEST DATA 매개변수가 NONE으로 설정되어 있으면 이 컬럼의 값은 NULL입니다.</p> <p>이 표준 편차는 요청 실행 시간 막대 그래프로 계산되며 막대 그래프가 데이터에 적합하게 제대로 크기 조정되지 않은 경우에는 정확하지 않을 수도 있습니다. 임의 값이 막대 그래프의 마지막 막대에 포함되면 -1이 값으로 리턴됩니다.</p> <p>서비스 서브클래스의 실행 시간 표준 편차는 서브클래스를 통해 전달되는 요청에 의해 영향을 받지 않지만 해당 서브클래스에서 완료되지는 않습니다.</p>
REQUEST_EXEC_TIME_TOTAL	BIGINT	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 요청에 대한 실행 시간의 합계. 단위는 밀리초입니다. 서비스 클래스에서 COLLECT AGGREGATE REQUEST DATA 매개변수가 NONE으로 설정되어 있으면 이 컬럼의 값은 NULL입니다.</p> <p>이 총계는 요청 실행 시간 막대 그래프로 계산되며 막대 그래프가 데이터에 적합하게 제대로 크기 조정되지 않은 경우에는 정확하지 않을 수도 있습니다. 임의 값이 막대 그래프의 마지막 막대에 포함되면 -1이 값으로 리턴됩니다.</p> <p>서비스 서브클래스의 총 실행 시간은 서브클래스를 통해 전달되는 요청에 의해 영향을 받지 않지만 해당 서브클래스에서 완료되지는 않습니다.</p>
ACT_REMAPMED_IN	BIGINT	마지막 재설정 이후 REMAP ACTIVITY 조치로 이 서비스 서브클래스에 다시 맵핑된 활동 수.
ACT_REMAPMED_OUT	BIGINT	마지막 재설정 이후 REMAP ACTIVITY 조치로 이 서비스 서브클래스에서 다시 맵핑된 활동 수.
CONCURRENT_WLO_TOP	INTEGER	마지막으로 재설정된 이후 이 파티션에서 지정된 워크로드에 대한 가장 높은 동시 어커런스 수입니다.
UOW_TOTAL_TIME_TOP	BIGINT	나중에 사용하기 위해 예약됨

## WLM\_GET\_SERVICE\_SUPERCLASS\_STATS - 서비스 슈퍼 클래스의 통계 리턴

WLM\_GET\_SERVICE\_SUPERCLASS\_STATS 함수는 하나 이상의 서비스 슈퍼 클래스에 대한 기본 통계를 리턴합니다.

## 구문

```
▶▶—WLM_GET_SERVICE_SUPERCLASS_STATS—(—service_superclass_name—, —————▶  
▶—dbpartitionnum—)—————▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *service\_superclass\_name*

현재 연결된 데이터베이스에서 서비스 수퍼 클래스 이름을 지정하는

VARCHAR

(128) 유형의 입력 인수입니다. 인수가 널(NULL) 또는 비어 있는 문자열인 경우 데이터베이스의 모든 수퍼 클래스에 대해 데이터가 검색됩니다.

### *dbpartitionnum*

현재 연결된 데이터베이스와 동일한 인스턴스에 있는 유효한 파티션 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션으로 1을 지정하거나 모든 데이터베이스 파티션으로 -2를 지정하십시오. 널(NULL) 값이 지정된 경우, -1 이 내재적으로 설정됩니다.

## 권한 부여

WLM\_GET\_SERVICE\_SUPERCLASS\_STATS 함수에 대한 EXECUTE 특권

## 예 :

다음 쿼리는 모든 데이터베이스 파티션에서 시스템의 모든 서비스 수퍼 클래스에 대한 기본 통계를 표시합니다.

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME, 1, 26) SERVICE_SUPERCLASS_NAME,  
        DBPARTITIONNUM,  
        LAST_RESET,  
        CONCURRENT_CONNECTION_TOP CONCURRENT_CONN_TOP  
FROM TABLE(WLM_GET_SERVICE_SUPERCLASS_STATS(' ', -2)) as SCSTATS
```

샘플 출력은 다음과 같습니다.

```
SERVICE_SUPERCLASS_NAME   DBPARTITIONNUM ...  
-----  
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
```

## 사용법 참고

WLM\_GET\_SERVICE\_SUPERCLASS\_STATS 테이블 함수는 서비스 수퍼 클래스 및 파티션당 하나의 데이터 행을 리턴합니다. 함수는 서비스 수퍼 클래스(파티션) 또는 파티션(하나 이상의 서비스 수퍼 클래스)에서는 데이터 집계를 수행하지 않습니다. 그렇지만 SQL 쿼리를 사용하여 이전 예에 표시된 대로 데이터를 수집할 수는 있습니다.

## 리턴되는 정보

표 206. WLM\_GET\_SERVICE\_SUPERCLASS\_STATS에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	이 레코드가 수집된 서비스 수퍼 클래스의 이름.
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호입니다.
LAST_RESET	TIMESTAMP	통계가 마지막으로 재설정된 시간입니다. 다음과 같은 4개의 이벤트가 통계 재설정을 트리거합니다. <ul style="list-style-type: none"> <li>• WLM_COLLECT_STATS 프로시저를 호출합니다.</li> <li>• <b>wlm_collect_int</b> 구성 매개변수로 인해 수집 및 재설정이 수행됩니다.</li> <li>• 데이터베이스를 재활성화합니다.</li> <li>• 통계를 보고 중인 서비스 수퍼 클래스를 수정하고 변경 사항을 커밋합니다.</li> </ul> LAST_RESET 시간소인은 로컬 시간입니다.
CONCURRENT_CONNECTION_TOP	INTEGER	마지막 재설정 이후 이 클래스에서 가장 높은 많은 코디네이터 연결 수입니다.

## WLM\_GET\_WORK\_ACTION\_SET\_STATS - 작업 조치 세트 통계 리턴

WLM\_GET\_WORK\_ACTION\_SET\_STATS 함수는 작업 조치 세트에 대한 통계를 리턴합니다.

### 구문

```

▶▶—WLM_GET_WORK_ACTION_SET_STATS—(—work_action_set_name—,—————▶
▶—dbpartitionnum—)—————▶▶

```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

*work\_action\_set\_name*

통계를 리턴할 작업 조치 세트를 지정하는 VARCHAR

(128) 유형의 입력 인수입니다. 인수가 널(NULL) 또는 비어 있는 문자열이면, 모든 작업 조치 세트에 대한 통계가 리턴됩니다.

#### *dbpartitionnum*

현재 연결된 데이터베이스와 동일한 인스턴스에 있는 유효한 파티션 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션으로 1을 지정하거나 모든 데이터베이스 파티션으로 -2를 지정하십시오. 널(NULL) 값이 지정된 경우, -1이 내재적으로 설정됩니다.

## 권한 부여

WLM\_GET\_WORK\_ACTION\_SET\_STATS 함수에 대한 EXECUTE 특권

### 예 :

ReadClass, WriteClass 및 LoadClass와 같은 세 개의 작업 클래스가 있다고 가정합니다. ReadClass와 연관된 작업 조치와 LoadClass와 연관된 작업 조치는 있으나 WriteClass와 연관된 작업 조치는 없습니다. 파티션 0에서 현재 실행 중이거나 큐에 있는 활동 수는 다음과 같습니다.

- ReadClass 클래스: 8개
- WriteClass 클래스: 4개
- LoadClass 클래스: 2개
- 지정되지 않음: 3개

```

ELECT SUBSTR(WORK_ACTION_SET_NAME,1,18) AS WORK_ACTION_SET_NAME,
       SUBSTR(CHAR(DBPARTITIONNUM),1,4) AS PART,
       SUBSTR(WORK_CLASS_NAME,1,15) AS WORK_CLASS_NAME,
       LAST_RESET,
       SUBSTR(CHAR(WLO_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, PART

```

샘플 출력은 다음과 같습니다. WriteClass 작업 클래스와 연관된 작업 조치가 없으므로 적용되는 4개의 활동은 출력에서 별표(\*)로 표시되는 인위적인 클래스에서 계산됩니다. 아무 작업 클래스에도 지정되지 않은 3개의 활동도 인위적인 클래스에 포함됩니다.

WORK_ACTION_SET_NAME	PART	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



## 리턴되는 정보

표 207. WLM\_GET\_WORK\_ACTION\_SET\_STATS에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
WORK_ACTION_SET_NAME	VARCHAR (128)	작업 조치 세트 이름. 작업 조치 세트를 사용한 경우에만 이름이 리턴됩니다.
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호입니다.
LAST_RESET	TIMESTAMP	통계가 마지막으로 재설정된 시간입니다. 다음과 같은 4개의 이벤트가 통계 재설정을 트리거합니다. <ul style="list-style-type: none"> <li>• WLM_COLLECT_STATS 프로시저를 호출합니다.</li> <li>• <code>wlm_collect_int</code> 구성 매개변수로 인해 수집 및 재설정이 수행됩니다.</li> <li>• 데이터베이스를 재활성화합니다.</li> <li>• 통계를 보고 중인 작업 조치 세트를 수정하고 변경사항을 커밋합니다.</li> </ul> <p>LAST_RESET 시간소인은 로컬 시간입니다.</p>
WORK_CLASS_NAME	VARCHAR (128)	지정한 작업 조치 세트와 관련된 작업 클래스의 이름. 작업 클래스에 연관된 작업 조치를 사용한 경우에만 작업 클래스 이름이 리턴됩니다. 별표(*)는 하나 이상의 작업 조치를 연관시킨 다른 작업 클래스에 속하지 않는 모든 해당 활동을 계산하기 위해 작성된 인위적인 작업 클래스를 나타냅니다.
ACT_TOTAL	BIGINT	WORK_CLASS_NAME을 지정하여 작업 클래스에 지정된 모든 중첩 레벨의 활동 이름

## WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97 - 활동 목록 리턴

WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97 함수는 지정한 파티션에서 지정한 응용프로그램으로 제출되어 아직 완료되지 않은 전체 활동 목록을 리턴합니다.

### 구문

```
▶▶—WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97—(—application_handle—,—————▶
▶—dbpartitionnum—)—————▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *application\_handle*

활동 목록이 리턴되는 응용프로그램 핸들을 지정하는 BIGINT 유형의 입력 인수입니다. 인수가 널(NULL)인 경우 데이터베이스의 모든 응용프로그램에 대해 데이터가 검색됩니다.

### *dbpartitionnum*

현재 연결된 데이터베이스와 동일한 인스턴스에 있는 유효한 파티션 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 널(NULL) 값이 지정된 경우, -1이 내재적으로 설정됩니다.

## 권한 부여

WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97 함수의 EXECUTE 특권입니다.

## 예:

### 예 1: 알려진 응용프로그램 핸들로 현재 실행 중인 활동

응용프로그램 핸들을 식별한 후에 현재 이 응용프로그램에서 실행 중인 모든 활동을 검색할 수 있습니다. 예를 들어, 관리자가 LIST APPLICATIONS 명령으로 판별된 응용프로그램 핸들이 1인 응용프로그램 활동을 나열하려는 경우 관리자는 다음 쿼리를 실행합니다.

```
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
```

쿼리의 샘플 출력은 다음과 같습니다.

COORD	PART	UOWID	ACTID	PARUOWID	PARACTID	ACTTYPE	NESTING
0	0	2	3	-	-	CALL	0
0	0	2	5	2	3	READ_DML	1

### 예 2: 시스템에서 현재 실행 중인 활동

다음 쿼리는 WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97 출력을 EXECUTABLE\_ID의 MON\_GET\_PKG\_CACHE\_STMT 출력과 조인하여 현재 시스템에서 실행 중인 모든 SQL 활동에 대해 명령문 텍스트를 제공합니다.

```

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

```

샘플 출력은 다음과 같습니다.

```

APPLICATION_HANDLE  UOW_ID  ACTIVITY_ID  STMT_TEXT
-----
1          1          1  SELECT * FROM SYSCAT.TABLES
47         1         36  INSERT INTO T1 VALUES(123)

```

## 리턴된 정보

표 208. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97에서 리턴된 정보

컬럼 이름	데이터 유형	설명
APPLICATION_HANDLE	BIGINT	응용프로그램에 대해 시스템 전체에서 사용되는 고유 ID. 단일 파티션의 데이터베이스에서 이 ID는 16비트의 카운터로 구성됩니다. 다중 파티션의 데이터베이스에서 이 ID는 16비트 카운터로 병합된 조정 파티션 번호로 구성됩니다. 또한 이 ID는 응용프로그램이 보조 연결을 작성하는 모든 파티션에서 동일합니다.
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호입니다.
COORD_PARTITION_NUM	SMALLINT	활동의 코디네이터 파티션 번호
LOCAL_START_TIME	TIMESTAMP	이 활동이 파티션에서 작업을 시작한 로컬 시간. 시스템에 활동이 입력되었지만 아직 큐에 있고 실행은 시작되지 않은 경우에 컬럼 값은 널(NULL)입니다.
UOW_ID	INTEGER	활동이 시작된 원래 작업 단위(UOW)의 고유 ID
ACTIVITY_ID	INTEGER	작업 단위(UOW)내의 고유 활동 ID
PARENT_UOW_ID	INTEGER	활동의 상위 활동이 시작된 원래 작업 단위(UOW)의 고유 ID. 활동에 상위 활동이 없거나 리모트 파티션에 있는 경우에는 이 컬럼 값이 널(NULL)입니다.
PARENT_ACTIVITY_ID	INTEGER	ACTIVITY_ID로 ID가 지정된 활동 상위의 작업 단위(UOW)내에서 고유한 활동 ID. 활동에 상위 활동이 없거나 리모트 파티션에 있는 경우에는 이 컬럼 값이 널(NULL)입니다.

표 208. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
ACTIVITY_STATE	VARCHAR(32)	<p>활동 상태. 가능한 값은 다음과 같습니다.</p> <p><i>CANCEL_PENDING</i>                      활동 요청에 대해 실제로 작동 중인 에이전트가 없기 때문에 활동이 취소되었습니다. 활동 중에 다음에 요청이 제출될 때 활동이 취소되고 SQL4725N 오류가 생성됩니다.</p> <p><i>EXECUTING</i>                      활동 요청에 대해 실제로 에이전트가 작동 중입니다.</p> <p><i>IDLE</i>    활동 요청을 실제로 처리 중인 에이전트가 없습니다.</p> <p><i>INITIALIZING</i>                      활동이 제출되었지만 아직 실행이 시작되지 않았습니다. 초기화 상태에서 예측적 임계값이 활동에 적용되어 활동 실행 허용 여부가 결정됩니다.</p>

표 208. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
ACTIVITY_STATE(계속)	VARCHAR(32)	<p>활동 상태. 가능한 값은 다음과 같습니다.</p> <p><i>QP_CANCEL_PENDING</i> 이 상태는 활동이 WLM_CANCEL_ACTIVITY 프로시저가 아닌 Query Patroller로 취소되었다는 점을 제외하고는 CANCEL_PENDING 상태와 동일합니다.</p> <p><i>QP_QUEUED</i> Query Patroller가 활동을 큐에 대기시켰습니다.</p> <p><i>QUEUED</i> 워크로드 관리 큐잉 임계값으로 활동이 큐에 대기되었습니다. 데이터베이스 파티셔닝 기능(DPF) 환경에서 이 상태는 임계값 티켓을 가져오기 위해 코디네이터 에이전트가 카탈로그 파티션에 대한 RPC를 작성했지만 아직 응답을 수신하지 않았음을 나타냅니다. 이 상태는 워크로드 관리 큐잉 임계값으로 활동이 큐에 대기되었음을 나타내거나 그리 오래 시간이 경과되지 않은 경우에는 활동이 해당 티켓을 가져오는 프로세스 중임을 나타낼 수 있습니다. 활동이 실제로 큐에 대기되어 있는지 정확하게 알려면 활동에서 작업 중인 에이전트를 판별하고 카탈로그 파티션에서 오브젝트의 EVENT_OBJECT 값에 WLM_QUEUE 값이 있는지 확인하십시오.</p> <p><i>TERMINATING</i> 활동의 실행이 완료되었으며 시스템에서 제거 중입니다.</p>

표 208. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
ACTIVITY_TYPE	VARCHAR(32)	<p>활동 유형. 가능한 값은 다음과 같습니다.</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> <p>각 활동 유형에 연관된 SQL문의 다른 유형에 대한 설명은 <i>Workload Manager Guide and Reference</i>의 『작업 클래스로 작업 유형 식별』의 내용을 참조하십시오.</p>
NESTING_LEVEL	INTEGER	이 활동이 최상위 활동에 중첩된 깊이입니다.
INVOCATION_ID	INTEGER	이 활동의 특정 호출을 동일한 중첩 레벨의 다른 호출과 구별하는 호출 ID.
ROUTINE_ID	INTEGER	루틴의 고유 ID.
UTILITY_ID	INTEGER	<p>다음 값 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• 활동이 유틸리티인 경우 값은 유틸리티 ID입니다.</li> <li>• 활동이 유틸리티가 아닌 경우 값은 널(NULL)입니다.</li> </ul>
SERVICE_CLASS_ID	INTEGER	이 활동이 속한 서비스 클래스의 고유 ID입니다.
DATABASE_WORK_ACTION_SET_ID	INTEGER	<p>다음 값 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• 이 활동이 데이터베이스 범위의 작업 클래스로 구분된 경우 값은 이 작업 클래스가 구성원인 작업 클래스 세트의 ID입니다.</li> <li>• 이 활동이 데이터베이스 범위의 작업 클래스로 구분되지 않은 경우 값은 널(NULL)입니다.</li> </ul>
DATABASE_WORK_CLASS_ID	INTEGER	<p>다음 값 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• 이 활동이 데이터베이스 범위의 작업 클래스로 구분된 경우 값은 작업 클래스 ID입니다.</li> <li>• 이 활동이 데이터베이스 범위의 작업 클래스로 구분되지 않은 경우 값은 널(NULL)입니다.</li> </ul>
SERVICE_CLASS_WORK_ACTION_SET_ID	INTEGER	<p>다음 값 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• 이 활동이 서비스 클래스 범위의 작업 클래스로 구분된 경우 값은 작업 클래스가 속한 작업 클래스 세트와 연관된 작업 조치 세트의 ID입니다.</li> <li>• 이 활동이 서비스 클래스 범위의 작업 클래스로 구분되지 않은 경우 값은 널(NULL)입니다.</li> </ul>

표 208. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
SERVICE_CLASS_WORK_CLASS_ID	INTEGER	다음 값 중 하나입니다. <ul style="list-style-type: none"> <li>이 활동이 서비스 클래스 범위의 작업 클래스로 구분된 경우 값은 이 활동에 지정된 작업 클래스의 ID입니다.</li> <li>이 활동이 서비스 클래스 범위의 작업 클래스로 구분되지 않은 경우 값은 널(NULL)입니다.</li> </ul>
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	섹션을 고유하게 식별하는 데이터 서버에서 생성된 opaque 2진 토큰입니다. 실행 가능한 ID를 다른 모니터링 인터페이스의 입력으로 사용하여 섹션에 대한 데이터를 가져올 수 있습니다. LOAD와 같이 SQL이 아닌 활동의 경우 널(NULL) 값이 리턴됩니다.
TOTAL_CPU_TIME	BIGINT	나중에 사용하도록 예약됩니다.
ROWS_READ	BIGINT	나중에 사용하도록 예약됩니다.
ROWS_RETURNED	BIGINT	나중에 사용하도록 예약됩니다.
QUERY_COST_ESTIMATE	BIGINT	나중에 사용하도록 예약됩니다.
DIRECT_READS	BIGINT	나중에 사용하도록 예약됩니다.
DIRECT_WRITES	BIGINT	나중에 사용하도록 예약됩니다.

## WLM\_GET\_WORKLOAD\_STATS\_V97 테이블 함수 - 워크로드 통계 리턴

WLM\_GET\_WORKLOAD\_STATS\_V97 함수는 워크로드 이름과 데이터베이스 파티션 번호의 모든 조합에 대해 한 개의 워크로드 통계 행을 리턴합니다.

### 구문

```
►►—WLM_GET_WORKLOAD_STATS_V97—(—workload_name—,—dbpartitionnum—)————►►
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *workload\_name*

통계가 리턴되는 워크로드를 지정하는 VARCHAR (128) 유형의 입력 인수입니다. 인수가 NULL 또는 비어 있는 문자열인 경우 모든 워크로드에 대한 통계가 리턴됩니다.

#### *dbpartitionnum*

현재 연결된 데이터베이스와 동일한 인스턴스에 있는 파티션 수를 지정하는 INTEGER 유형의 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 널(NULL) 값을 지정한 경우 내재적으로 -1이 설정됩니다.

## 권한 부여

WLM\_GET\_WORKLOAD\_STATS\_V97 함수의 EXECUTE 특권입니다.

예 :

다음 쿼리는 워크로드 통계를 표시합니다.

```
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
```

쿼리의 샘플 출력은 다음과 같습니다.

```
WL_DEF_NAME      PART COORD_ACT_LIFETIME_TOP ...
-----
SYSDEFAULTADMWORKL 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
```

## 사용법 참고

함수는 워크로드, 파티션 또는 서비스 클래스에서 데이터 집계를 수행하지 않습니다. 그렇지만 SQL 쿼리를 사용하여 데이터를 수집할 수는 있습니다.

## 리턴된 정보

표 209. WLM\_GET\_WORKLOAD\_STATS\_V97에서 리턴된 정보

컬럼 이름	데이터 유형	설명
WORKLOAD_NAME	VARCHAR (128)	이 레코드가 수집된 워크로드의 이름입니다.
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호입니다.
LAST_RESET	TIMESTAMP	<p>통계가 마지막으로 재설정된 시간입니다. 다음과 같은 4개의 이벤트가 통계 재설정을 트리거합니다.</p> <ul style="list-style-type: none"> <li>WLM_COLLECT_STATS 프로시저를 호출합니다.</li> <li>wlm_collect_int 구성 매개변수로 인해 수집 및 재설정 이 수행됩니다.</li> <li>데이터베이스를 재활성화합니다.</li> <li>통계를 보고 중인 워크로드를 수정하고 변경사항을 커밋합니다.</li> </ul> <p>LAST_RESET 시간소인은 로컬 시간입니다.</p>



표 209. WLM\_GET\_WORKLOAD\_STATS\_V97에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
CONCURRENT_WLO_TOP	INTEGER	마지막 재설정 이후 이 파티션에서 지정된 워크로드에 대한 가장 높은 동시 어커런스 수입니다.
CONCURRENT_WLO_ACT_TOP	INTEGER	마지막 재설정 이후에 이 워크로드에 대한 모든 어커런스에 도달한 실행 상태(유휴 및 대기 포함) 또는 큐에 대기된 상태의 가장 높은 동시 활동(코디네이터 및 중첩 모두) 수입니다. 컬럼 값은 해당 작업 단위(UOW) 완료 시점에서 각 워크로드 어커런스로 갱신됩니다.
COORD_ACT_COMPLETED_TOTAL	BIGINT	마지막 재설정 이후에 완료된 이 워크로드의 임의 어커런스에 지정된 모든 중첩 레벨에서 전체 코디네이터 활동 수입니다. 이 컬럼 값은 해당 작업 단위(UOW) 완료 시점에서 각 워크로드 어커런스로 갱신됩니다.
COORD_ACT_ABORTED_TOTAL	BIGINT	마지막 재설정 이후에 완료되기 전에 중단된 이 워크로드의 임의 어커런스에 지정된 모든 중첩 레벨에서 전체 코디네이터 활동 수입니다. 이 컬럼 값은 해당 작업 단위(UOW) 완료 시점에서 각 워크로드 어커런스로 갱신됩니다.
COORD_ACT_REJECTED_TOTAL	BIGINT	<p>마지막 재설정 이후에 실행 전에 거부된 이 워크로드의 임의 어커런스에 지정된 모든 중첩 레벨에서 전체 코디네이터 활동 수입니다. 이 컬럼 값은 해당 작업 단위(UOW) 완료 시점에서 각 워크로드 어커런스로 갱신됩니다.</p> <p>활동은 실행 금지 작업 조치 또는 예측적 임계값으로 인해 실행되지 않은 경우 거부됨으로 간주됩니다.</p> <p>WLM_GET_SERVICE_SUBCLASS_STATS_V97 함수에서 동일한 이름의 컬럼과는 달리</p> <p>WLM_GET_WORKLOAD_STATS_V97 컬럼은 서비스 클래스에 활동을 지정하기 전에 발생하는 거부 수도 포함합니다. 예를 들어, 이런 거부는 활동이 ConcurrentWorkloadOccurrences 임계값을 위반할 때 발생합니다.</p>
WLO_COMPLETED_TOTAL	BIGINT	마지막 재설정 이후에 완료되는 워크로드 어커런스 수입니다.
COORD_ACT_LIFETIME_TOP	BIGINT	모든 중첩 레벨에서 수집된 코디네이터 활동 수명의 상위 워터 마크(water mark)입니다. 단위는 밀리초입니다. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정된 경우 컬럼 값은 널(NULL)입니다.
COORD_ACT_LIFETIME_AVG	DOUBLE	이 워크로드와 연관된 중첩 레벨 0에서 완료 또는 중단된 코디네이터의 활동 수명에 대한 산술적인 의미입니다. 단위는 밀리초입니다. 내부적으로 추적되는 평균이 오버플로우된 경우 값으로 -2가 리턴됩니다. 워크로드의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정되면 컬럼 값은 널(NULL)입니다.

표 209. WLM\_GET\_WORKLOAD\_STATS\_V97에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
COORD_ACT_LIFETIME_STDDEV	DOUBLE	이 워크로드와 연관된 중첩 레벨 0에서 완료 또는 중단된 코디네이터의 활동 수명에 대한 표준 편차입니다. 단위는 밀리초입니다. 워크로드의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정되면 컬럼 값은 널(NULL)입니다. 이 표준 편차는 코디네이터의 활동 수명 막대 그래프로 계산되며 막대 그래프가 데이터에 적합하게 제대로 크기 조정되지 않은 경우에는 정확하지 않을 수도 있습니다. 임의 값이 마지막 막대 그래프 바이너리에 포함되면 -1 값이 리턴됩니다.
COORD_ACT_EXEC_TIME_AVG	DOUBLE	이 워크로드와 연관된 중첩 레벨 0에서 완료 또는 중단된 코디네이터의 실행 시간에 대한 산술적인 의미입니다. 단위는 밀리초입니다. 내부적으로 추적되는 평균이 오버플로우된 경우 값으로 -2가 리턴됩니다. 워크로드의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정되면 컬럼 값은 널(NULL)입니다.
COORD_ACT_EXEC_TIME_STDDEV	DOUBLE	이 워크로드와 연관된 중첩 레벨 0에서 완료 또는 중단된 코디네이터의 실행 시간에 대한 표준 편차입니다. 단위는 밀리초입니다. 이 표준 편차는 코디네이터의 활동 실행 시간 막대 그래프로 계산되며 막대 그래프가 데이터에 적합하게 제대로 크기 조정되지 않은 경우에는 정확하지 않을 수도 있습니다. 임의 값이 마지막 막대 그래프 바이너리에 포함되면 -1 값이 리턴됩니다. 워크로드의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정되면 컬럼 값은 널(NULL)입니다.
COORD_ACT_QUEUE_TIME_AVG	DOUBLE	이 워크로드와 연관된 중첩 레벨 0에서 완료 또는 중단된 코디네이터의 큐 시간에 대한 산술적인 의미입니다. 단위는 밀리초입니다. 내부적으로 추적되는 평균이 오버플로우된 경우 값으로 -2가 리턴됩니다. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정된 경우 컬럼 값은 널(NULL)입니다.
COORD_ACT_QUEUE_TIME_STDDEV	DOUBLE	이 워크로드와 연관된 중첩 레벨 0에서 완료 또는 중단된 코디네이터의 큐 시간에 대한 표준 편차입니다. 단위는 밀리초입니다. 워크로드의 COLLECT AGGREGATE ACTIVITY DATA 매개변수가 NONE으로 설정되면 컬럼 값은 널(NULL)입니다. 이 표준 편차는 코디네이터의 큐 시간 수명 막대 그래프로 계산되며 막대 그래프가 데이터에 적합하게 제대로 크기 조정되지 않은 경우에는 정확하지 않을 수도 있습니다. 임의 값이 마지막 막대 그래프 바이너리에 포함되면 -1 값이 리턴됩니다.
UOW_TOTAL_TIME_TOP	BIGINT	나중에 사용하기 위해 예약됨

## WLM\_SET\_CLIENT\_INFO 프로시저 - 클라이언트 정보 설정

WLM\_SET\_CLIENT\_INFO 프로시저는 DB2 서버에서 현재 연결과 연관된 클라이언트 정보를 설정합니다.

이 프로시저를 사용하여 클라이언트의 사용자 ID, 응용프로그램 이름, 워크스테이션 이름, 어카운팅 정보 또는 워크로드 정보를 DB2 서버에 설정할 수 있습니다. 이 프로시저를 호출하면 연관된 트랜잭션 프로세서(TP) 모니터 클라이언트 정보 필드 및 이 연결에 대한 특수 레지스터 설정에 저장된 값이 변경됩니다.

이 클라이언트 정보 필드는 DB2 서버에서 사용되어 응용프로그램이나 연결을 사용하는 사용자의 ID를 판별합니다. 연결에 대한 클라이언트 정보 필드는 DB2 워크로드 평가 시 고려되며 이 연결을 위해 생성된 응용프로그램 스냅샷이나 DB2 감사 레코드에 나타납니다.

sqlseti API와 달리 이 프로시저는 클라이언트에서 클라이언트 정보를 설정하지는 않지만 대신 DB2 서버에서 해당 클라이언트 속성을 설정합니다. 따라서 sqlqry API는 이 프로시저를 사용하여 DB2 서버에 설정된 클라이언트 정보를 조회할 수 없습니다.

이 프로시저에서 제공되는 데이터 값은 관련 TP 모니터 필드나 특수 레지스터에 저장되기 전에 해당 데이터베이스 코드 페이지로 변환됩니다. 데이터베이스 코드 페이지로 변환한 후에 지원되는 최대 크기를 초과하는 임의 데이터 값은 서버에 저장되기 전에 절단됩니다. 절단된 값은 저장된 해당 값이 조회될 때 TP 모니터 필드와 특수 레지스터 모두에서 리턴됩니다.

WLM\_SET\_CLIENT\_INFO 프로시저는 트랜잭션에서 제어되지 않으며 프로시저에서 수행된 클라이언트 정보 변경은 작업 단위(UOW) 커미트나 롤백에 영향을 받지 않습니다. 그렇지만 워크로드 재평가가 각 응용프로그램의 다음 작업 단위(UOW) 처음에 수행되기 때문에 COMMIT 또는 ROLLBACK문을 실행하여 클라이언트 정보 변경이 적용되도록 해야 합니다.

응용프로그램이 IBM Data Server에서 이식 가능해야 하는 경우에는 WLM\_SET\_CLIENT\_INFO 프로시저를 사용하는 대신 데이터베이스 응용프로그램 프로그래밍 인터페이스 중 하나를 사용하여 서버에 클라이언트 정보를 설정하십시오. 프로시저는 Linux, UNIX 및 Windows용 DB2 데이터베이스에서만 사용할 수 있습니다.

## 구문

```
►►—WLM_SET_CLIENT_INFO—(—client_userid—,—client_wrkstnname—,—  
►—client_applname—,—client_acctstr—,—client_workload—)—————►
```

스키마는 SYSPROC입니다.

## 프로시저 매개변수

*client\_userid*

클라이언트의 사용자 ID를 지정하는 VARCHAR(255) 유형의 입력 인수. NULL

을 지정하면 값이 변경되지 않습니다. 디폴트값인 비어 있는 문자열("")을 지정하면 클라이언트의 사용자 ID는 공백인 디폴트 값으로 재설정됩니다.

#### *client\_wrkstnname*

클라이언트의 워크스테이션 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. NULL을 지정하면 값이 변경되지 않습니다. 디폴트값인 비어 있는 문자열("")을 지정하면 클라이언트의 워크스테이션 이름은 공백인 디폴트 값으로 재설정됩니다.

#### *client\_applname*

클라이언트의 응용프로그램 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. NULL을 지정하면 값이 변경되지 않습니다. 디폴트값인 비어 있는 문자열("")을 지정하면 클라이언트의 응용프로그램 이름은 공백인 디폴트 값으로 재설정됩니다.

#### *client\_acctstr*

클라이언트의 어카운팅 문자열을 지정하는 VARCHAR(255) 유형의 입력 인수. NULL을 지정하면 값이 변경되지 않습니다. 디폴트값인 비어 있는 문자열("")을 지정하면 클라이언트의 어카운팅 문자열은 공백인 디폴트 값으로 재설정됩니다.

#### *client\_workload*

클라이언트의 워크로드 지정 모드를 지정하는 VARCHAR(255) 유형의 입력 인수. NULL을 지정하면 값이 변경되지 않습니다. 값은 다음과 같습니다.

#### **SYSDEFAULTADMWORKLOAD**

데이터베이스 연결이 SYSDEFAULTADMWORKLOAD에 지정되어 ACCESSCTRL, DATAACCESS, DBADM, SECADM 또는 WLMADM 권한이 있는 사용자는 일반 워크로드 평가를 생략할 수 있습니다.

#### **AUTOMATIC**

서버가 자동으로 수행하는 워크로드 평가에서 선택된 워크로드에 데이터베이스 연결이 지정됨을 지정합니다.

주: *client\_workload* 인수는 대소문자가 구분됩니다.

### **권한 부여**

WLM\_SET\_CLIENT\_INFO 프로시저에서의 EXECUTE 특권.

#### **예:**

다음 프로시저 호출은 클라이언트에 대한 사용자 ID, 워크스테이션 이름, 응용프로그램 이름, 어카운팅 문자열 및 워크로드 지정 모드를 설정합니다.

```
CALL SYSPROC.WLM_SET_CLIENT_INFO('db2user', 'machine.torolab.ibm.com',  
    'auditor', 'Accounting department', 'AUTOMATIC')
```

다음 프로시저 호출은 다른 클라이언트 속성을 설정하지 않고 클라이언트의 사용자 ID를 db2user2로만 설정합니다.

```
CALL SYSPROC.WLM_SET_CLIENT_INFO('db2user2', NULL, NULL, NULL, NULL)
```

다음 프로시저 호출은 다른 클라이언트 속성 값을 수정하지 않고 클라이언트의 사용자 ID를 공백으로 재설정합니다.

```
CALL SYSPROC.WLM_SET_CLIENT_INFO('', NULL, NULL, NULL, NULL)
```



---

## 제 20 장 기타 루틴 및 뷰

---

### ADMIN\_COPY\_SCHEMA 프로시저 - 특정 스키마 및 해당 오브젝트 복사

ADMIN\_COPY\_SCHEMA 프로시저는 특정 스키마 및 스키마에 포함된 모든 오브젝트를 복사하는 데 사용됩니다. 새 목표 스키마 오브젝트는 소스 스키마의 오브젝트와 같은 오브젝트 이름을 사용하지만 스키마 규정자는 목표 스키마 규정자를 사용합니다. ADMIN\_COPY\_SCHEMA 프로시저는 원래 테이블의 데이터를 포함하거나 포함하지 않고 테이블을 복사하는 데 사용할 수 있습니다.

#### 구문

```
▶—ADMIN_COPY_SCHEMA—(—sourceschema—,—targetschema—,—copymode—,——————▶  
▶—objectowner—,—sourcetbsp—,—targettbsp—,—errortabschema—,—errortab—)—▶
```

스키마는 SYSPROC입니다.

#### 프로시저 매개변수

##### *sourceschema*

복사 중인 오브젝트의 스키마 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. 이름은 대소문자를 구분합니다.

##### *targetschema*

복사된 오브젝트를 작성하기 위해 고유한 스키마 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. 이름은 대소문자를 구분합니다. 스키마 이름이 이미 존재하는 경우 프로시저 호출이 실패하며 프로시저를 호출하기 전에 스키마가 제거되어야 한다는 메시지를 리턴합니다.

##### *copymode*

복사 조작 모드를 지정하는 VARCHAR(128) 유형의 입력 인수. 유효한 옵션은 다음과 같습니다.

- 'DDL': 소스 스키마로부터 지원되는 모든 오브젝트의 비어 있는 복사본을 작성합니다.
- 'COPY': 소스 스키마로부터 모든 오브젝트의 비어 있는 복사본을 작성한 다음 데이터가 있는 각 목표 스키마 테이블을 로드합니다. 로드는 'NONRECOVERABLE' 모드에서 수행됩니다. ADMIN\_COPY\_SCHEMA 호출 후에 백업이 이루어져야 하며, 그렇지 않은 경우 복사된 테이블이 다음 복구에 액세스할 수 없습니다.

- ‘COPYNO’: 소스 스키마로부터 모든 오브젝트의 비어 있는 복사본을 작성한 다음 데이터가 있는 각 목표 스키마 테이블을 로드합니다. 로드는 ‘COPYNO’ 모드에서 수행됩니다.

주: *copymode*가 ‘COPY’ 또는 ‘COPYNO’인 경우 완전한 파일 이름(예를 들어 ‘COPYNO /home/mckeough/loadoutput’)은 *copymode* 매개변수 값과 함께 지정될 수 있습니다. 경로가 전달된 후 로드 메시지는 표시된 파일에 로그됩니다. 파일 이름은 인스턴스에서 분리 루틴 호출로 사용된 사용자 ID로 작성 가능해야 합니다. 경로가 지정되지 않은 경우 로드 메시지 파일이 버려집니다(기본 동작).

#### *objectowner*

복사된 오브젝트의 소유자로 사용된 권한 부여 ID를 지정하는 VARCHAR(128) 유형의 입력 인수. 널(NULL)인 경우 소유자는 복사 조작을 수행하는 사용자의 권한 부여 ID가 됩니다.

#### *sourcectbsp*

쉽게로 구분된 복사할 소스 테이블 스페이스 목록을 지정하는 CLOB(2 M) 유형의 입력 인수. 구분된 테이블 스페이스 이름이 지원됩니다. 작성 중인 각 테이블의 경우 이 목록에 있는 테이블 스페이스와 테이블 정의는 *targettbsp* 목록에서 n번째 항목으로 변환됩니다. 널(NULL)이 이 매개변수에 대해 지정되면 소스 오브젝트가 사용한 동일한 테이블 스페이스를 사용하여 새 오브젝트가 작성됩니다.

#### *targettbsp*

쉽게로 구분된 복사할 목표 테이블 스페이스 목록을 지정하는 CLOB(2 M) 유형의 입력 인수. 구분된 테이블 스페이스 이름이 지원됩니다. 테이블 스페이스는 테이블 스페이스의 *sourcectbsp* 목록의 각 항목에 대해 지정되어야 합니다. *sourcectbsp* 목록의 nth 테이블 스페이스는 DDL 재연 시 *targettbsp* 목록의 nth 테이블 스페이스에 맵핑됩니다. 최종 테이블 스페이스(추가 테이블 스페이스 이름으로 소스 목록의 이름에 해당되지 않습니다.)로 ‘SYS\_ANY’를 지정할 수 있습니다. ‘SYS\_ANY’가 지정된 경우 오브젝트 작성 시 기본 테이블 스페이스 선택 알고리즘이 사용됩니다(선택 알고리즘에 관한 세부 정보는 CREATE TABLE 명령문 문서의 IN *tablespace-name1* 옵션 참조). 널(NULL)이 이 매개변수에 대해 지정되면 소스 오브젝트가 사용한 동일한 테이블 스페이스를 사용하여 새 오브젝트가 작성됩니다.

#### *errortabschema*

복사할 수 없는 오브젝트의 오류 정보를 포함한 테이블의 스키마 이름을 지정하는 VARCHAR(128) 유형의 입력 및 출력 인수. 이 테이블은 SYSTOOLSPACE 테이블 스페이스의 ADMIN\_COPY\_SCHEMA 프로시저에서 사용자에게 의해 작성됩니다. 오류가 발생하지 않으면 이 매개변수는 출력에서 널(NULL)입니다.

#### *errortab*

복사할 수 없는 오브젝트의 오류 정보를 포함한 테이블의 이름을 지정하는



VARCHAR(128) 유형의 입력 및 출력 인수. 이 테이블은 SYSTOOLSPACE 테이블 스페이스의 ADMIN\_COPY\_SCHEMA 프로시저에서 사용자에게 의해 작성됩니다. 이 테이블은 프로시저에서 호출한 사용자 ID에서 소유합니다. 오류가 발생하지 않으면 이 매개변수는 출력에서 널(NULL)입니다. 테이블이 작성될 수 없거나 이미 존재하면 프로시저 조작성이 실패하며 오류 메시지가 리턴됩니다. ADMIN\_COPY\_SCHEMA 프로시저에 대한 호출이 뒤따르는 사용자에게 의해 테이블이 정리되어야 합니다. 즉 테이블은 SYSTOOLSPACE에서 사용 중인 스페이스를 재개하기 위해 삭제되어야 합니다.

표 210. ADMIN\_COPY\_SCHEMA errortab 형식

컬럼 이름	데이터 유형	설명
OBJECT_SCHEMA	VARCHAR(128)	복사 명령이 실패한 오브젝트의 스키마 이름.
OBJECT_NAME	VARCHAR(128)	복사 명령이 실패한 오브젝트의 이름.
OBJECT_TYPE	VARCHAR(30)	오브젝트의 유형.
SQLCODE	INTEGER	오류 SQLCODE.
SQLSTATE	CHAR(5)	오류 SQLSTATE.
ERROR_TIMESTAMP	TIMESTAMP	실패한 조작성의 실패 시간.
STATEMENT	CLOB(2 M)	오브젝트 실패 DDL. 데이터가 목표 테이블에 로드되었을 경우 실패하면 이 필드는 실패한 로드 명령에 해당되는 텍스트를 포함합니다.
DIAGTEXT	CLOB(2 K)	실패한 조작성의 오류 메시지 텍스트.

## 권한 부여

스키마 복사가 성공하려면 이 프로시저를 호출하는 사용자 ID는 소스 테이블에서 선택할 권한 및 로드할 권한 모두를 포함한 적절한 오브젝트 작성 권한이 있어야 합니다. 소스 스키마의 테이블이 LBAC(label based access control)에 의해 보호되면, 사용자 ID에 목표 테이블에서 동일한 보호를 작성할 수 있게 허용하는 LBAC 증명서가 있어야 합니다. 데이터로 복사하는 경우 사용자 ID에 소스 테이블로부터의 데이터 읽기 및 목표 테이블에 데이터 작성 모두를 허용하는 LBAC 증명서가 있어야 합니다.

ADMIN\_COPY\_SCHEMA 프로시저의 EXECUTE 특권도 필요합니다.

예 :

```
CALL SYSPROC.ADMIN_COPY_SCHEMA('SOURCE_SCHEMA', 'TARGET_SCHEMA',
    'COPY', NULL, 'SOURCETS1', SOURCETS2', 'TARGETTS1', TARGETTS2,
    SYS_ANY', 'ERRORSCHEMA', 'ERRORNAME')
```

## 제한사항

- DDL *copymode*만이 HADR 데이터베이스를 지원합니다.
- COPY 또는 COPY NO를 포함한 XML은 지원되지 않습니다.

- COPYNO를 포함한 ADMIN\_COPY\_SCHEMA 프로시저를 사용하면 백업 보류 중인 상태에 있는 목표 데이터베이스 오브젝트에 테이블 스페이스를 배치합니다. 로드 조작이 완료된 후 목표 스키마 테이블은 무결성 설정 보류 상태로 설정되며 ADMIN\_COPY\_SCHEMA 프로시저는 SET INTEGRITY 명령문을 발행하여 이 상태의 테이블을 확보합니다. 테이블 스페이스가 이미 백업 보류 중인 상태이기 때문에 SET INTEGRITY 명령문이 실패합니다. 이 문제 해결 방법에 관한 정보는 『스키마 복사』를 참조하십시오.

## 사용 시 참고사항

- 복사 중인 오브젝트 내의 완전한 오브젝트에 대한 참조는 수정되지 않습니다. ADMIN\_COPY\_SCHEMA 프로시저는 작성 중인 오브젝트의 규정 스키마만을 변경하며, 이들 오브젝트에 대한 SQL 표현식 내에 나타나는 스키마 이름은 변경하지 않습니다. 생성된 컬럼과 트리거 내용과 같은 오브젝트가 포함됩니다.
- 이 프로시저는 다음 오브젝트 복사를 지원하지 않습니다.
  - 인덱스 확장
  - 별칭
  - 패키지
  - 입력된 테이블
  - 배열 유형
  - 사용자 정의 구조화된 유형(및 변형된 함수)
  - 유형이 지정된 뷰
  - jars(Java™ 루틴 아카이브)
  - 스테이징 테이블
- 위의 오브젝트 중 하나가 복사 중인 스키마에 존재하면 오브젝트는 복사되지 않고 항목이 오브젝트를 복사할 수 없음을 표시하는 오류 테이블에 추가됩니다.
- 복제된 테이블이 복사되면 테이블의 새 복사본에서는 서브스크립션을 사용할 수 없습니다. 테이블은 기본 테이블로만 재작성됩니다.
- 이 프로시저의 조작은 SYSTOOLSPACE 테이블 스페이스가 있어야 합니다. 이 테이블 스페이스는 ADMIN\_COPY\_SCHEMA 프로시저에서 사용된 메타데이터뿐 아니라 이 프로시저에서 리턴된 오류 테이블을 보류하는 데 사용됩니다. 테이블 스페이스가 존재하지 않으면 오류가 리턴됩니다.
- 목표 스키마에서의 오브젝트에 대한 통계가 기본으로 설정됩니다.
- 테이블에 생성된 ID 컬럼이 있고 *copymode*가 'COPY' 또는 'COPYNO'라면, 소스 테이블로부터의 데이터 값이 로드 시 유지됩니다.
- 새 카탈로그 항목은 각 외부 루틴에 대해 작성되며 원래 소스 루틴의 실행 파일을 참조합니다.

- 테이블이 복사 조작 시작 시 무결성 설정 보류 상태로 설정되면 데이터는 목표 테이블로 로드되지 않으며 데이터가 해당 테이블에 대해 로드되지 않았음을 표시하면서 *errortab*에 항목이 로그됩니다.
- 로드 또는 DDL 조작이 실패하면 오브젝트가 생성되지 않았기 때문에 항목은 *errortab*에 로그됩니다. 작성된 모든 오브젝트가 남습니다. 복구하기 위해 수동 로드를 시작하거나 새 스키마를 ADMIN\_DROP\_SCHEMA 프로시저를 사용하여 삭제하거나 ADMIN\_COPY\_SCHEMA 프로시저를 다시 호출할 수 있습니다.
- DDL 재연 시 소스 스키마와 일치하면 기본 스키마는 목표 스키마를 겹쳐씹니다.
- 트리거나 뷰 또는 SQL 함수를 컴파일하는 데 사용된 함수 경로는 다음 예외가 있는 소스 오브젝트를 작성하는 데 사용되는 경로입니다. 오브젝트의 함수 경로에 소스 스키마 이름이 포함되면 경로의 이 항목은 DDL 재연 시 목표 스키마 이름으로 수정됩니다.
- 다중 ADMIN\_COPY\_SCHEMA 프로시저를 실행하면 교착 상태에 빠지게 됩니다. 한번에 하나의 ADMIN\_COPY\_SCHEMA 프로시저만 발행해야 합니다. 복사 처리 중에 소스 스키마의 테이블 변경은 목표 스키마의 데이터가 복사 후에 동일하지 않음을 의미할 수 있습니다.
- 단일 파티션 데이터베이스 파티션 그룹에서부터 다중 파티션 데이터베이스 파티션 그룹까지 테이블과 함께 스키마를 복사할 때 고려사항을 고려해야 합니다. 자동 분산 키 선택이 우선되지 않으면 스키마 복사 조작을 실행하기 전에 분산 키를 테이블에서 정의해야 합니다. 분산 키 변경은 단일 파티션 데이터베이스 파티션 그룹과 연관된 테이블 스페이스의 테이블에 대해 수행될 수 있습니다.

### 트랜잭션 고려사항

- ADMIN\_COPY\_SCHEMA 프로시저가 처리 중에 교착 상태 또는 잠금 시간종료로 인해 강제로 롤백되었다면, ADMIN\_COPY\_SCHEMA 프로시저를 호출하는 작업 단위로 수행되는 작업도 롤백될 수 있습니다.
- 복사의 DDL 단계 중 실패하면 목표 스키마에 대한 모든 변경이 세이프포인트로 롤백됩니다.
- *copymode*가 'COPY' 또는 'COPYNO'로 설정된 경우, 복사의 DDL 단계가 완료되면 ADMIN\_COPY\_SCHEMA 프로시저가 커밋되고 프로시저를 호출한 작업 단위에서 수행된 임의의 작업을 커밋합니다.

---

## ADMIN\_DROP\_SCHEMA 프로시저 - 특정 스키마 및 해당 오브젝트 삭제

ADMIN\_DROP\_SCHEMA 프로시저는 특정 스키마와 이 스키마에 포함된 모든 오브젝트를 삭제하는 데 사용됩니다.

## 구문

```
▶▶—ADMIN_DROP_SCHEMA—(—schema—,—dropmode—,—errortabschema—,—errortab—)
```

스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *schema*

오브젝트가 삭제되고 있는 스키마 이름을 지정하는 VARCHAR(128) 유형의 입력 인수 이름은 대문자로 지정해야 합니다.

### *dropmode*

나중에 사용하기 위해 예약되었으며 NULL로 설정해야 합니다.

### *errortabschema*

삭제할 수 없는 오브젝트에 대한 오류 정보가 들어 있는 테이블의 스키마 이름을 지정하는 VARCHAR(128) 유형의 입력 및 출력 인수. 이름은 대소문자를 구분합니다. 이 테이블은 SYSTOOLSPACE 테이블 스페이스의 ADMIN\_DROP\_SCHEMA 프로시저에서 사용자가 작성합니다. 오류가 발생하지 않으면 이 매개변수는 출력에서 널(NULL)입니다.

### *errortab*

삭제할 수 없는 오브젝트에 대한 오류 정보가 들어 있는 테이블 이름을 지정하는 VARCHAR(128) 유형의 입력 및 출력 인수. 이름은 대소문자를 구분합니다. 이 테이블은 SYSTOOLSPACE 테이블 스페이스의 ADMIN\_DROP\_SCHEMA 프로시저에서 사용자가 작성합니다. 이 테이블은 프로시저를 호출하는 사용자 ID가 소유합니다. 오류가 발생하지 않으면 이 매개변수는 출력에서 널(NULL)입니다. 테이블이 작성될 수 없거나 이미 존재하면 프로시저 조작이 실패하며 오류 메시지가 리턴됩니다. 테이블은 ADMIN\_DROP\_SCHEMA 프로시저를 호출한 다음에 사용자가 정리해야 합니다. 즉, SYSTOOLSPACE에서 사용하는 스페이스를 다시 사용하려면 테이블을 삭제해야 합니다.

표 211. ADMIN\_DROP\_SCHEMA errortab 형식

컬럼 이름	데이터 유형	설명
OBJECT_SCHEMA	VARCHAR(128)	삭제 명령에 실패한 오브젝트의 스키마 이름
OBJECT_NAME	VARCHAR(128)	삭제 명령에 실패한 오브젝트 이름
OBJECT_TYPE	VARCHAR(30)	오브젝트의 유형.
SQLCODE	INTEGER	오류 SQLCODE.
SQLSTATE	CHAR(5)	오류 SQLSTATE.
ERROR_TIMESTAMP	TIMESTAMP	실패한 삭제 조작의 실패 시간

표 211. ADMIN\_DROP\_SCHEMA errortab 형식 (계속)

컬럼 이름	데이터 유형	설명
STATEMENT	CLOB(2 M)	오브젝트 실패 DDL.
DIAGTEXT	CLOB(2 K)	실패한 삭제 조작에 대한 오류 메시지의 텍스트

## 권한 부여

이 프로시저를 호출하는 사용자가 제거하고 있는 모든 오브젝트에는 삭제 권한이 필요합니다.

ADMIN\_DROP\_SCHEMA 프로시저에 대한 EXECUTE 특권이 또한 필요합니다.

### 예 :

```
CALL SYSPROC.ADMIN_DROP_SCHEMA('SCHNAME', NULL, 'ERRORSCHEMA', 'ERRORTABLE')
```

다음은 해당 프로시저의 출력 예입니다.

출력 매개변수 값

```
-----
매개변수 이름 : ERRORTABSCHEMA
매개변수 값 : ERRORSCHEMA <-- error!
```

```
매개변수 이름 : ERRORTAB
매개변수 값 : ERRORTABLE <-- error!
```

리턴 상태 = 0

내부 오류가 발견된 경우에만 리턴 상태가 0이 아닙니다(예를 들어, SYSTOOLSPACE 가 존재하지 않을 경우).

오류 테이블을 쿼리하여 오류를 점검할 수 있습니다.

```
SELECT * FROM ERRORSCHEMA.ERRORTABLE
```

## 사용 시 참고사항

- 다른 스키마의 오브젝트가 삭제되고 있는 오브젝트에 따라 달라지면, 디폴트 DROP 문 시맨틱이 적용됩니다.
- 이 프로시저는 다음 오브젝트를 삭제할 수 없습니다.
  - 인덱스 확장
  - 별칭
  - 패키지
  - 입력된 테이블
  - 배열 유형
  - 사용자 정의 구조화된 유형(및 변형된 함수)

- 유형이 지정된 뷰
- jars(Java 루틴 아카이브)
- 스테이징 테이블
- 위의 오브젝트 중 하나가 삭제되고 있는 스키마에 존재하면, 오브젝트 및 스키마 모두 삭제되지 않고 오브젝트가 삭제되지 않았음을 알리는 항목이 오류 테이블에 추가됩니다.
- 이 프로시저의 조작은 SYSTOOLSPACE 테이블 스페이스가 있어야 합니다. 이 테이블 스페이스는 ADMIN\_DROP\_SCHEMA 프로시저에서 사용하는 메타데이터와 이 프로시저에서 리턴된 오류 테이블을 보유하는 데 사용됩니다. 테이블 스페이스가 존재하지 않으면 오류가 리턴됩니다.

---

## ADMIN\_MOVE\_TABLE 프로시저 - 온라인 테이블 이동

ADMIN\_MOVE\_TABLE 스토어드 프로시저는 데이터가 온라인으로 남아있어서 액세스할 수 있는 동안에 활성 테이블의 데이터를 이름이 같은 새 테이블 오브젝트로 이동합니다. 이 스토어드 프로시저는 상태 정보와 이동할 테이블과 관련된 구성 옵션이 포함된 행으로 구성된 프로토크 테이블을 작성합니다. 이 프로시저의 리턴 세트는 이동할 테이블과 관련된 해당 프로토크 테이블의 행입니다.

이 스토어드 프로시저는 다음 용어를 사용합니다.

### 소스 테이블

스토어드 프로시저에 매개변수로서 전달된 원래 테이블 이름. 이 테이블은 이 동시킴 테이블입니다.

### 목표 테이블

스토어드 프로시저를 통해 전달된 테이블 정의를 사용하여 스토어드 프로시저에 의해 작성된 테이블. 소스 테이블의 모든 데이터는 이 테이블에 복사된 다음 소스 테이블과 같은 이름으로 변경됩니다.

### 스테이징 테이블

스토어드 프로시저에 의해 작성된 테이블. 스테이징 테이블은 테이블 이동을 실행하는 중에 소스 테이블에서 발생한 갱신, 삭제 또는 삽입 변경사항을 저장합니다. 이동이 완료되면 이 테이블이 삭제됩니다.

## 구문

유효성이 동등한 두 개의 메소드로 ADMIN\_MOVE\_TABLE을 호출할 수 있습니다. 첫 번째 메소드는 목표 테이블의 테이블 정의의 특정 부분만 수정할 수 있습니다. 예를 들어 테이블 정의가 매우 커서(3~4 KB) 테이블의 테이블 스페이스를 수정하고자 할 경우 소스 테이블 재작성에 필요한 전체 CREATE TABLE문을 판별하지 않아도 수정할 수 있습니다. 기타 선택적 매개변수를 공백으로 두고 data\_tbsp, index\_tbsp 및 lob\_tbsp 매개변수를 채우기만 하면 됩니다.

두 번째 메소드는 스토어드 프로시저가 목표 테이블을 작성하도록 하기 보다는 사용자가 미리 목표 테이블을 작성하도록 함으로써 제어 및 유연성이 더 향상됩니다. 이 메소드는 첫 번째 메소드로는 할 수 없는 목표 테이블을 작성할 수 있습니다.

메소드 1:

```
▶▶ADMIN_MOVE_TABLE—(—tabschema—,—tablename—,—data_tbsp—,—index_tbsp—,—  
▶▶lob_tbsp—,—mdc_cols—,—partkey_cols—,—data_part—,—coldef—,—  
▶▶options—,—operation—)
```

메소드 2:

```
▶▶ADMIN_MOVE_TABLE—(—tabschema—,—tablename—,—target_tabname—,—  
▶▶options—,—operation—)
```

두 메소드의 스키마는 모두 SYSPROC입니다.

## 프로시저 매개변수

### *tabschema*

이 입력 매개변수는 이동시킬 테이블이 포함된 스키마의 이름을 지정합니다. 이 매개변수는 대소문자를 구분하며 데이터 유형은 VARCHAR (128)입니다.

### *tablename*

이 입력 매개변수는 이동시킬 테이블의 이름을 지정합니다. 이 매개변수는 대소문자를 구분하며 데이터 유형은 VARCHAR (128)입니다.

### *data\_tbsp*

이 입력 매개변수는 목표 테이블의 새 데이터 테이블 스페이스를 지정합니다. 값을 입력하면 *index\_tbsp* 및 *lob\_tbsp* 매개변수가 필요합니다. 값을 입력하지 않으면 소스 테이블의 데이터 테이블 스페이스가 사용됩니다. 이 매개변수는 대소문자를 구분하며 데이터 유형은 VARCHAR (128)입니다. 이 매개변수는 NULL 또는 비어 있는 문자열입니다.



### *index\_tbsp*

이 입력 매개변수는 목표 테이블의 새 인덱스 테이블 스페이스를 지정합니다. 값을 입력하면 *index\_tbsp* 및 *lob\_tbsp* 매개변수가 필요합니다. 값을 입력하지 않으면 소스 테이블의 인덱스 테이블 스페이스가 사용됩니다. 이 매개변수는 대소문자를 구분하며 데이터 유형은 VARCHAR

(128)입니다. 이 매개변수는 NULL 또는 비어 있는 문자열입니다.

### *lob\_tbsp*

이 입력 매개변수는 목표 테이블의 새 LOB 테이블 스페이스를 지정합니다. 값을 입력하면 *index\_tbsp* 및 *lob\_tbsp* 매개변수가 필요합니다. 값을 입력하지 않으면 소스 테이블의 LOB 테이블 스페이스가 사용됩니다. 이 매개변수는 대소문자를 구분하며 데이터 유형은 VARCHAR

(128)입니다. 이 매개변수는 NULL 또는 비어 있는 문자열입니다.

### *mdc\_cols*

이 입력 매개변수는 목표 테이블의 다차원 컬럼(MDC) 스펙을 제공합니다. 다중 차원을 따라서 목표 테이블의 데이터 클러스터에 사용되는 컬럼의 목록을 쉼표로 구분하여 값을 입력합니다. 값이 NULL이거나 "-"인 경우 ORGANIZE BY DIMENSIONS절이 사용되지 않습니다. 비어 있는 문자열 또는 단일 공백이 있으면 프로시저가 소스 테이블에 MDC 스펙이 있는지를 확인하고 있는 경우 해당 스펙을 사용합니다. 이 매개변수의 데이터 유형은 VARCHAR(32672)이며 형식은 CREATE TABLE문의 ORGANIZE BY DIMENSIONS절과 동일합니다. 이 매개변수는 NULL, 비어 있는 문자열 또는 단일 공백입니다.

예: 'C1, C4, (C3,C1), C2'

### *partkey\_cols*

이 입력 매개변수는 목표 테이블에 파티셔닝 키 컬럼 스펙을 제공합니다. 다중 데이터베이스 파티션에 데이터를 분산하는 방법을 지정하는 키 컬럼의 목록을 쉼표로 구분하여 값을 입력합니다. 값이 NULL이거나 "-"인 경우 PARTITIONING KEY절이 사용되지 않습니다. 비어 있는 문자열 또는 단일 공백이 있으면 프로시저가 소스 테이블에 파티셔닝 키 컬럼 스펙이 있는지를 확인하고 있는 경우 해당 스펙을 사용합니다. 이 매개변수의 데이터 유형은 VARCHAR(32672)이며 형식은 CREATE TABLE문의 DISTRIBUTE BY HASH절과 동일합니다.

예: 'C1, C3'

### *data\_part*

이 입력 매개변수는 목표 테이블에 대해 데이터 파티셔닝 스펙을 제공합니다. 이 명령문은 하나 이상의 테이블 컬럼의 값에 따라 다중 스토리지 오브젝트(데이터 파티셔닝이라고 함)의 테이블 데이터를 나누는 방법을 정의합니다. 값이 널(NULL)이거나 "-"인 경우 PARTITION BY RANGE절이 사용되지 않습니다. 비어 있는 문자열 또는 단일 공백이 있으면 프로시저가 소스 테이블에 데이터 파티션 스킴이 있는지를 확인하고 있는 경우 해당 정보(파티션 이름 포함)를 사용합니다. 이 매개변



수의 데이터 유형은 VARCHAR(32672)이며 형식은 REATE TABLE문의 PARTITION BY RANGE절과 동일합니다.

예: '(C1) (STARTING FROM (1) EXCLUSIVE ENDING AT (1000) EVERY (100))'

#### *coldef*

이 입력 매개변수는 목표 테이블의 새 컬럼 정의를 규정하여 컬럼 유형이 호환 가능한 경우 컬럼 유형을 변경할 수 있습니다. 그러나 컬럼 이름은 계속 동일해야 합니다.

또한 이 입력 매개변수는 새 컬럼을 추가하고 기존 컬럼을 삭제할 수 있습니다. 컬럼을 추가할 때 NULL 입력 가능으로 정의하거나 디폴트값을 설정해야 합니다. 컬럼은 테이블에 고유 인덱스 또는 1차 인덱스가 있는 경우에만 삭제될 수 있으며 삭제할 컬럼은 그 고유 인덱스 또는 1차 인덱스의 일부가 아닙니다. 이 매개변수의 데이터 유형은 VARCHAR(32672)입니다. 이 매개변수는 NULL 또는 비어 있는 문자열입니다.

예: 'C1 INT, C2 INT DEFAULT 0'

#### *target\_tabname*

이 입력 매개변수는 이동 중에 목표 테이블로 사용할 기존 테이블의 이름을 제공합니다. 다음 변경사항을 전달 중인 목표 테이블에 작성할 수 있습니다.

- 데이터, 인덱스 및 LOB 테이블 스페이스를 변경할 수 있습니다.
- 다차원 컬럼(MDC) 스펙을 추가하거나 변경할 수 있습니다.
- 파티셔닝 키 컬럼 스펙을 추가하거나 변경할 수 있습니다.
- 데이터 파티셔닝 스펙을 추가하거나 변경할 수 있습니다.
- 데이터 압축을 추가하거나 제거할 수 있습니다.
- 새 컬럼 정의를 지정할 수 있으나 *coldef* 매개변수를 지정할 때와 같은 제한사항이 여기에 적용됩니다.

다음 제한사항은 이름 지정된 테이블에 적용됩니다.

- 테이블은 소스 테이블과 동일한 스키마에 존재해야 합니다.
- 테이블은 비어 있어야 합니다.
- 입력된 테이블, 구체화된 쿼리 테이블(MQT), 스테이징 테이블, 리모트 테이블 또는 클러스터된 테이블이 허용되지 않습니다.

이 매개변수가 NULL 또는 비어 있는 문자열로 설정되면 스토어드 프로시저는 소스 테이블과 같은 정의를 사용합니다. 이 매개변수는 대소문자를 구분하며 데이터 유형은 VARCHAR

(128)입니다.

### options

쉽게로 구분된 이 입력 매개변수의 세트는 스토어드 프로시저에 사용되는 옵션을 정의합니다.

- **KEEP:** 이 옵션은 다른 이름 밑의 원래 소스 테이블의 사본을 보존합니다. 소스 테이블 이름이 T1이면 이동 후에 해당 테이블의 이름이 T1AAAAVxo와 같은 이름으로 자동으로 변경됩니다. ORIGINAL 키 밑의 리턴된 프로토콜 테이블의 소스 테이블의 정확한 이름을 검색할 수 있습니다. SWAP 단계 이하에서 이 옵션을 설정할 수 있습니다.
- **COPY\_USE\_LOAD:** 이 옵션은 비회복가능 db2Load API를 사용하여 소스 테이블의 데이터를 목표 테이블로 복사합니다. COPY 단계 이하에서 이 옵션을 설정할 수 있습니다.
- **COPY\_WITH\_INDEXES:** 이 옵션은 소스 테이블을 복사하기 전에 인덱스를 작성합니다. 그러나 디폴트는 소스 테이블을 복사한 후 인덱스를 작성하는 것입니다. 이 옵션의 장점은 복사 후 인덱스를 작성할 때 인덱스마다 전체 테이블 스캔을 해야하고 인덱스 작성은 활성 로그 스페이스가 필요한 트랜잭션이라는 점입니다. LOGINDEXREBUILD 데이터베이스 구성 매개변수가 설정되면 짧은 시간 프레임으로 인덱스를 빌드하는데 에 상당한 로그 스페이스가 필요합니다. 이 옵션의 한 가지 단점은 인덱스가 목표 테이블에 인덱스가 유지 보수되어야 하므로 복사 성능이 감소되는 것입니다. 또한 결과 인덱스에 "유사" 삭제된 키가 여러 개 포함되어 있고 복사 후에 인덱스가 작성된 것처럼 인덱스의 밸런스가 잘 맞지 않습니다. COPY 단계 이하에서 COPY\_WITH\_INDEXES 옵션을 설정할 수 있습니다.
- **FORCE:** 강제 옵션이 설정되면 SWAP 단계에서 소스 테이블이 테이블 정의를 변경했는지 여부를 확인하지 않습니다. 또한 LOAD가 사용되면 FORCE가 필요하므로 LOAD가 사용되고 FORCE가 설정되지 않은 경우 오류 메시지가 표시 됩니다. SWAP 단계 이하에서 이 옵션을 설정할 수 있습니다.
- **NO\_STATS:** 이 옵션은 목표 테이블에 RUNSTATS 또는 통계 복사를 시작하지 않습니다. AUTO\_RUNSTATS 또는 AUTO\_STMT\_STATS 데이터베이스 구성 매개변수를 사용하면 DB2가 이후에 자동으로 새 통계를 작성합니다. 이전 버전과의 호환성에 STATS\_NO도 승인됩니다. SWAP 단계 이하에서 NO\_STATS 옵션을 설정할 수 있습니다.
- **COPY\_STATS:** 이 옵션은 스왑을 수행하기 전에 소스 테이블의 통계를 목표 테이블로 복사합니다. 이 옵션은 특히 페이지 크기가 변경되면 정확하지 않은 실제 통계의 원인이 될 수 있습니다. 그러나 이 옵션을 설정하면 새 통계를 계산하기 위해 RUNSTATS를 호출하지 않으므로 계산 시간을 절약할 수 있습니다. 또한 통계가 같으므로 옵티마이저가 같은 액세스 플랜을 선택할 수 있습니다. 이전 버전과의 호환성에 STATS\_COPY도 승인됩니다. SWAP 단계 이하에서 STATS\_COPY 옵션을 설정할 수 있습니다.

- NO\_AUTO\_REVAL: 이 옵션은 테이블에 자동 유효성 다시 확인을 방지하는 대신 모든 트리거 및 뷰를 재작성합니다. NO\_AUTO\_REVAL 옵션은 INIT 단계에서만 설정될 수 있습니다.
- REORG: 이 옵션은 스왑을 수행하기 전에 목표 테이블에 특별 오프라인 REORG를 설정합니다. 압축 사전을 향상시키기 위해 이 옵션보다 디폴트 샘플링 접근 방법을 사용하는 것이 최적의 압축 사전을 작성하는 데 더 나은 메소드입니다. 그러나 최적의 XML 압축 사전이 필요한 경우 REORG가 유일한 메소드입니다. SWAP 단계 이하에서 REORG 옵션을 설정할 수 있습니다.

이 옵션 목록은 대소문자를 구분하지 않으며 데이터 유형은 VARCHAR (128)입니다. 목록 값은 NULL 또는 비어 있는 문자열입니다.

### *operation*

이 입력 매개변수는 스토어드 프로시저가 실행될 조작을 지정합니다. 스토어드 프로시저 호출 방법에는 두 가지가 있습니다. MOVE 명령을 사용하여 모든 조작을 한 번에 실행하는 방법과 개별 명령을 사용하여 한 번에 한 단계씩 테이블 이동을 실행하는 방법입니다. 이 두 번째 메소드의 주요 장점은 SWAP 단계가 실제로 발생할 때를 제어하여 테이블이 잠시 오프라인이 될 때를 판별한다는 점입니다. 이로 인하여 낮은 시스템 활동 기간 중에 이동이 가능하게 됩니다. 개별 명령을 사용하는 경우 INIT, COPY, REPLAY, VERIFY(선택적) 및 SWAP의 순서로 명령이 호출됩니다.

- MOVE: 한 단계에서 전체 테이블 이동(INIT, COPY, REPLAY 및 SWAP 조작)을 수행합니다.
- INIT: 테이블 이동이 발생할 수 있는지 확인하고 테이블 이동 프로세스(목표 테이블, 스테이징 테이블 및 소스 테이블의 트리거) 중에 필요한 모든 데이터를 초기화합니다.
- COPY: 소스 테이블에서 목표 테이블로 콘텐츠를 복사합니다. 이 때 소스 테이블에 발생하는 갱신, 삭제 또는 삽입이 캡처되어 스테이징 테이블에 저장됩니다. COPY\_WITH\_INDEXES 옵션을 선택하지 않으면 COPY 단계 끝에서 새 인덱스가 작성됩니다. 또한 필요한 경우 REPLAY 단계 중에 성능을 향상시키기 위하여 소스 및 목표 테이블에 2차 인덱스가 작성됩니다. INIT 단계가 완료되어야 COPY를 사용할 수 있습니다.
- REPLAY: COPY 단계가 시작된 이후 소스 테이블에 변경된 행을 목표 테이블에 복사합니다. COPY 단계가 완료되어야 REPLAY를 사용할 수 있습니다.
- VERIFY: 소스 및 목표 테이블의 테이블 콘텐츠가 동일한지를 선택적으로 확인합니다. 이 프로세스는 소스 및 목표 테이블에 공유된 잠금 확보, 소스 테이블에 발생한 변경사항 재생 및 비교 수행으로 구성됩니다. 테이블에 고유 인덱스가 있으면 이 명령은 양 쪽 테이블에 있는 컬럼 사이의 모든 값을 비교합니다. 아니면 이 명령은 양 쪽 테이블에 있는 컬럼 사이의 모든 값(LONG, LOB 또는

XML 컬럼은 제외)을 비교합니다. 이것은 값비싼 조작이므로 이동에 유용할지 여부를 결정하는데 주의해야 합니다. COPY 또는 REPLAY 단계가 완료되어야 VERIFY를 사용할 수 있습니다.

- **SWAP:** 스테이징 테이블의 마지막 스캔 중에 적용된 변경사항의 수가 이 프로토콜 테이블에 저장된 REPLAY\_THRESHOLD 값보다 적어질 때까지 REPLAY 단계를 실행합니다. 그러면 소스 테이블이 잠시 오프라인이 되어 마지막 REPLAY를 완료한 다음 이 명령이 소스 테이블을 목표 테이블로 스왑하고 테이블을 다시 온라인으로 되돌립니다. COPY 단계가 완료되어야 SWAP을 사용할 수 있지만 REPLAY 단계가 호출된 후 사용하는 것이 바람직합니다.
- **CLEANUP:** 스테이징 테이블, 스토어드 프로시저에 의해 소스 테이블에 작성된 비고유 인덱스 또는 트리거 그리고 KEEP 옵션이 설정되지 않은 경우 소스 테이블을 삭제합니다. SWAP 단계 중에 명령이 실패하면 CLEANUP을 호출할 수 있습니다.
- **CANCEL:** 단계 사이에서 다중 단계 테이블 이동을 취소하거나 실패된 테이블 이동 조작을 취소합니다. 이 명령을 실행하려면 조작 상태가 COMPLETED 또는 CLEANUP이 아니어야 합니다. CANCEL은 모든 중간 데이터(인덱스, 스테이징 테이블, 목표 테이블 및 소스 테이블의 트리거)를 지웁니다.

이 매개변수는 대소문자를 구분하지 않으며 데이터 유형은 VARCHAR (128)입니다.

## 권한 부여

ADMIN\_MOVE\_TABLE 스토어드 프로시저를 호출하는 데 명시적 권한 부여가 필요하지 않습니다. 그러나 스토어드 프로시저를 실행하려면, 적절한 오브젝트 작성 권한(소스 테이블에서 선택할 권한 및 복사 조작에 대해 삽입할 권한 포함)이 있어야 합니다.

## 예:

이 예는 첫 번째 메소드를 사용하여 스토어드 프로시저를 호출하며 여기에서 프로시저 내에 목표 테이블이 정의되어 스키마 "SVALENTI"에 있는 T1 테이블을 이동합니다.

```
CALL SYSPROC.ADMIN_MOVE_TABLE(
'SVALENTI',
'T1',
'ACCOUNTING',
'ACCOUNT_IDX',
'ACCOUNT_LONG',
'',
'',
'',
'',
'CUSTOMER VARCHAR(80), REGION CHAR(5), YEAR INTEGER, CONTENTS CLOB',
'',
'MOVE')
```

다음은 이 쿼리로부터의 출력 예입니다.

결과 세트 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

22개의 레코드가 선택되었습니다.

리턴 상태 = 0

이 예는 첫 번째 메소드를 사용하여 스토어드 프로시저를 호출하며 여기에서 프로시저 외부에 목표 테이블을 작성한 다음 *target\_tabname* 매개변수 내에 이름을 지정하여 위와 같은 테이블을 이동합니다.

첫 단계는 테이블을 수동으로 작성하는 것입니다.

```
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'
```

그러면 스토어드 프로시저를 호출하여 목표 테이블의 이름으로 전달합니다.

```
CALL SYSPROC.ADMIN_MOVE_TABLE(  
'SVALENTI',  
'T1',  
'T1_TARGET',  
'',  
'MOVE')
```

다음은 이 쿼리로부터의 출력 예입니다.

결과 세트 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

```

22개의 레코드가 선택되었습니다.

리턴 상태 = 0

## 사용 시 참고사항

이 프로시저 사용 시 최고의 결과를 위한 제안사항:

- 같은 테이블 스페이스에 여러 개를 한 번에 이동하지 마십시오. 이렇게 하면 목표 테이블에서 분할화가 되지 않습니다.
- 테이블의 활동이 낮은 경우 이 프로시저를 실행하십시오. 데이터를 많이 로드하거나 삭제하지 않으면 병렬 읽기 액세스 문제가 발생하지 않습니다.
- 다중 단계 이동 조작을 사용하십시오. 언제든지 INIT 및 COPY 단계를 호출할 수 있습니다. 스테이징 테이블 크기를 작게 유지하려면 REPLAY 단계를 여러 번 실행한 다음 테이블에 활동이 낮을 때 SAWAP을 발행하십시오.
- 특히 고유 인덱스가 없는 테이블을 고려할 때 및 인덱스가 없는 테이블의 경우 테이블 이동에 오프라인 메소드를 사용하는 것이 더 나은 선택인지 확인하십시오.

## 제한사항

다음 제한사항은 ADMIN\_MOVE\_TABLE 스토어드 프로시저에 적용됩니다.

- 단순 테이블만 소스 테이블로 지원됩니다. 구체화된 쿼리 테이블, 입력된 테이블, 클러스터된 테이블, 시스템 테이블, 뷰, 별칭 또는 별명이 허용되지 않습니다.
- 이벤트 모니터가 테이블에서 현재 활성 상태인 경우, 테이블을 이동시킬 수 없습니다.
- 상위 또는 하위의 외부 키(참조 제한조건)가 지원되지 않습니다. 외부 키로 테이블을 이동하려면 db2look 명령을 사용하여 외부 키를 캡처한 다음 외부 키를 삭제(drop)하고 이동 조작을 수행하여 키를 다시 작성할 수 있습니다.
- 고유 인덱스가 없는 테이블은 복합적이고 비싼 응답 단계에 종속됩니다.
- 테이블에 LOB, XML 또는 LONG 컬럼이 포함된 경우 고유 인덱스가 필요합니다.

- 생성된 컬럼은 MDC 스펙의 일부가 될 수 없습니다.
- 텍스트 검색 인덱스에 대해서는 지원되지 않습니다.
- 프로시저가 테이블 및 인덱스 그리고 스테이징 테이블 및 로그 스페이스의 두 개의 사본을 작성할 때 대형 디스크 스페이스 요구사항을 고려하십시오.
- 대부분의 데이터가 "선택에서 삽입" 형식을 사용하여 새 테이블에 이동될 때 복사 성능이 문제가 될 수 있습니다.
- 고유 인덱스가 없는 테이블의 VERIFY 조각이 LOB가 포함된 테이블에서 수행되지 않습니다.
- DB2 레지스트리 변수 DB2\_SKIPDELETED를 ON으로 설정할 수 없습니다.
- SYSTOOLSPACE 테이블 스페이스를 작성하여 'PUBLIC'에 액세스할 수 있어야 합니다.
- 소스 테이블에 트랜잭션이 오래 실행되므로 COPY 단계 중에 잠금 시간종료가 가능합니다.
- SWAP 단계 중에 교착 상태가 발생할 수 있습니다.
- 비고유 인덱스 및 몇 개의 갱신 프로세스가 있는 소스 테이블에서 교착 상태가 발생할 수 있습니다.
- VARCHAR2 지원이 가능하면 데이터베이스가 비어 있는 문자열 및 NULL을 동등한 값으로 처리하지만 단일 공백은 구별 값입니다. VARCHAR2 지원이 가능하면 *mdc\_cols*, *partkey\_cols* 및 *data\_part* 매개변수에 비어 있는 문자열 및 NULL과 구분되는 것으로 단일 공백을 사용할 수 있습니다.

## 정보 리턴

표 212. ADMIN\_MOVE\_TABLE 스토어드 프로시저에서 리턴된 정보

컬럼 이름	데이터 유형	설명
TABSCHEMA	VARCHAR (128)	이동시킬 테이블의 스키마. 전체 시스템 디폴트의 비어 있는 문자열.
TABNAME	VARCHAR (128)	이동시킬 테이블의 테이블 이름. 전체 시스템 디폴트의 비어 있는 문자열.
KEY	VARCHAR(32)	속성의 이름
VALUE	CLOB(10M)	속성 값.

결과 세트에 리턴된 키와 값 쌍을 표 213에서 찾을 수 있습니다. 결과 세트에서 사용자가 구성할 수 있는 키를 수정하려면 ADMIN\_MOVE\_TABLE\_UTIL 스토어드 프로시저를 사용하십시오.

표 213. ADMIN\_MOVE\_TABLE 스토어드 프로시저에서 리턴된 키 및 값 쌍

키	리턴 값	사용자 구성 가능
VERSION	스토어드 프로시저의 버전을 표시합니다.	아니오



표 213. ADMIN\_MOVE\_TABLE 스토어드 프로시저에서 리턴된 키 및 값 쌍 (계속)

키	리턴 값	사용자 구성 가능
AUTHID	스토어드 프로시저를 호출한 사용자의 권한 부여 ID를 표시합니다.	아니오
LOCK	다른 온라인 테이블 이동 스토어드 프로시저 호출이 활성화된 경우 LOCK 시작 시간을 표시하고 그렇지 않은 경우 비어 있습니다.	아니오
STATUS	온라인 테이블 이동의 현재 상태를 다음과 같이 표시합니다. <ul style="list-style-type: none"> <li>• INIT: INIT가 진행 중</li> <li>• COPY: COPY가 진행 중이거나 가능함</li> <li>• REPLAY: REPLAY가 진행 중이거나 REPLAY 및 SWAP 이 가능함</li> <li>• CLEANUP: MOVE이 완료되었으나 정리가 아직 완료되지 않았거나 CLEANUP이 가능함</li> <li>• COMPLETE: MOVE 및 CLEANUP이 완료됨</li> <li>• COMPLETE_WITH_WARNINGS: MOVE 및 CLEANUP 이 완료되었으나 경고가 있음(WARNINGS 키 밑에 목록이 있음)</li> </ul>	아니오
STAGING	스테이징 테이블의 이름을 표시합니다.	아니오
TARGET	목표 테이블의 이름을 표시합니다.	아니오
ORIGINAL	스왑 후 원래 테이블의 이름을 표시합니다.	아니오
INDEXSCHEMA	테이블에 인덱스가 없는 경우 인덱스의 스키마 또는 비어 있는 문자열을 표시합니다.	아니오
INDEXNAME	테이블에 인덱스가 없는 경우 인덱스의 이름 또는 비어 있는 문자열을 표시합니다.	아니오
COMMIT_AFTER_N_ROWS	COPY 단계 중에 이 많은 행이 복사된 후 확약이 실행됩니다. 0은 COPY 중에 확약이 없음을 의미합니다. 디폴트값은 10000입니다.	예
DEEPCOMPRESSSION_SAMPLE	소스 테이블에 압축이 가능하면 이 필드는 샘플화된 데이터 양 (KB 단위)과 압축 사전 작성 시기를 지정합니다. 0은 샘플링이 수행되지 않았음을 의미합니다. 디폴트값은 20MB(20480KB)입니다.	예
COPY_ARRAY_SIZE	COPY_ARRAY_INSERT의 ARRAY 크기를 지정합니다. 0 이하의 값은 COPY_ARRAY_INSERT를 사용하지 않음을 의미합니다. 디폴트값은 100입니다.	예
COPY_OPTS	COPY 단계 중에 사용되는 복사 옵션	아니오
COPY_INDEXSCHEMA	COPY 단계 중에 목표 테이블의 데이터 클러스터에 사용된 인덱스의 스키마. 이 값은 COPY 단계 전에 설정해야 합니다. 디폴트 스키마는 존재하는 경우 소스 테이블의 클러스터 인덱스의 스키마 이름이며 그렇지 않은 경우 소스 테이블의 고유 또는 1차 인덱스의 스키마 이름입니다.	예



표 213. ADMIN\_MOVE\_TABLE 스토어드 프로시저에서 리턴된 키 및 값 쌍 (계속)

키	리턴 값	사용자 구성 가능
COPY_INDEXNAME	COPY 단계 중에 목표 테이블의 데이터 클러스터에 사용된 인덱스의 이름. 이 값은 COPY 단계 전에 설정해야 합니다. 디폴트 이름은 존재하는 경우 소스 테이블의 클러스터 인덱스의 이름이며 그렇지 않은 경우 소스 테이블의 고유 또는 1차 인덱스의 이름입니다.	예
INDEX_CREATION_TOTAL_TIME	2차 인덱스 작성에 필요한 총 시간을 표시합니다.	아니오
INIT_START	INIT 단계 시작 시간을 표시합니다.	아니오
INIT_END	INIT 단계 종료 시간을 표시합니다.	아니오
COPY_START	COPY 단계 시작 시간을 표시합니다.	아니오
COPY_END	COPY 단계 종료 시간을 표시합니다.	아니오
COPY_TOTAL_ROWS	COPY 단계 중에 복사한 총 행 수를 표시합니다.	아니오
REPLAY_START	REPLAY 단계 시작 시간을 표시합니다.	아니오
REPLAY_END	REPLAY 단계 종료 시간을 표시합니다.	아니오
REPLAY_TOTAL_ROWS	재생된 행의 누적 수를 표시합니다.	아니오
REPLAY_TOTAL_TIME	행을 재생하는데 사용되는 데 사용된 누적 시간을 초 단위로 표시합니다.	아니오
REPLAY_MAX_ERR_RETRIES	REPLAY 단계 중에 발생할 수 있는 최대 오류 재시도 계수 (잠금 시간종료 또는 교착 상태)를 지정합니다. 디폴트값은 100입니다.	예
REPLAY_THRESHOLD	REPLAY 단계를 한 번 반복하는 경우 스테이징 테이블에 적용된 행 수가 이 값보다 적으면 새 항목이 중간에 작성되더라도 REPLAY가 중지됩니다. 디폴트값은 100입니다.	예
REORG_USE_TEMPSPACE	REORG 옵션을 호출하면 REORG 명령의 USE절에 임시 테이블 스페이스를 지정할 수도 있습니다. 여기에서 값이 지정되지 않으면 REORG 명령이 재구성 중인 테이블과 같은 테이블 스페이스를 사용합니다.	예
VERIFY_START	검증 시작 시간을 표시합니다.	아니오
VERIFY_END	검증 종료 시간을 표시합니다.	아니오
SWAP_START	SWAP 단계 시작 시간을 표시합니다.	아니오
SWAP_END	SWAP 단계 종료 시간을 표시합니다.	아니오
SWAP_MAX_RETRIES	(잠금 시간 종료 또는 교착 상태가 발생하는 경우) SWAP 단계 중에 허용된 최대 재시도 수를 지정합니다. 디폴트값은 10입니다.	예
SWAP_RETRIES	SWAP 단계 중에 수행한 재시도 횟수를 표시합니다.	아니오
CLEANUP_START	CLEANUP 단계 시작 시간을 표시합니다.	아니오
CLEANUP_END	CLEANUP 단계 종료 시간을 표시합니다.	아니오
WARNINGS	사용자에게 전달할 경고를 표시합니다. 이 경고에는 다음이 포함됩니다. <ul style="list-style-type: none"> <li>• 실패한 모든 오브젝트의 유효성 다시 확인</li> <li>• 제어자 사용자, 그룹 또는 역할에 부여될 수 없음</li> <li>• 인덱스가 참조하는 컬럼이 더 이상 존재하지 않으므로 인덱스를 작성할 수 없음</li> </ul>	아니오

## ADMIN\_MOVE\_TABLE\_UTIL 프로시저 - 온라인 이동 테이블 프로시저 수정

활성 테이블 데이터를 이동할 때 ADMIN\_MOVE\_TABLE\_UTIL 프로시저는 SYSPROC.ADMIN\_MOVE\_TABLE 스토어드 프로시저와 함께 수행됩니다. 이 스토어드 프로시저는 ADMIN\_MOVE\_TABLE 프로토콜 테이블에 사용자가 정의할 수 있는 값을 변경하는 메커니즘을 제공하며 이 메커니즘은 ADMIN\_MOVE\_TABLE 프로시저에 의해 작성되어 사용됩니다.

TABSCHEMA 및 TABNAME 매개변수가 참조한 테이블의 테이블 이동이 이미 진행 중인 경우 그리고 프로시저의 호출자 권한 부여 ID가 사용자가 실행하는 테이블 이동과 같은 경우 이 프로시저는 ADMIN\_MOVE\_TABLE 프로토콜 테이블의 값만 수정합니다.

### 구문

```
►►—ADMIN_MOVE_TABLE_UTIL—(—tabschema—,—tablename—,—action—,—key—,—value—)—►►
```

이 스토어드 프로시저에 대한 스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *tabschema*

이 입력 매개변수는 이동 중인 테이블이 포함된 스키마의 이름을 지정합니다. 이 이름은 대소문자가 구분되며 데이터 유형은 VARCHAR(128)입니다.

#### *tablename*

이 입력 매개변수는 이동 중인 테이블의 이름을 지정합니다. 이 매개변수는 대소문자를 구분하며 데이터 유형은 VARCHAR(128)입니다.

#### *action*

이 입력 매개변수는 실행할 프로시저에 대한 조치를 지정합니다.

가능한 값은 다음과 같습니다.

- UPSERT: 지정된 TABSCHEMA.TABNAME.KEY가 ADMIN\_MOVE\_TABLE 프로토콜 테이블에 존재하는 경우 이 값이 해당 VALUE를 새 *value* 매개변수를 사용하여 갱신합니다. 아니면 이 값이 KEY와 VALUE 쌍을 ADMIN\_MOVE\_TABLE 프로토콜 테이블에 삽입합니다.
- DELETE: 지정된 TABSCHEMA.TABNAME. KEY가 ADMIN\_MOVE\_TABLE 프로토콜 테이블에 존재하는 경우 이 값이 ADMIN\_MOVE\_TABLE 프로토콜 테이블에서 지정된 KEY와 VALUE 쌍을 삭제합니다.

이 매개변수의 데이터 유형은 VARCHAR(128)입니다.

### key

이 입력 매개변수는 ADMIN\_MOVE\_TABLE 프로토콜 테이블의 "upsert" 또는 삭제할 키를 지정합니다.

올바른 값은 다음과 같습니다.

- COMMIT\_AFTER\_N\_ROWS: COPY 단계 중에 이 많은 행이 복사된 후 확약이 실행됩니다. 값 0은 COPY 중에 확약이 실행되지 않음을 의미합니다.
- DEEPCOMPRESSSION\_SAMPLE: 소스 테이블에 압축이 가능하면 이 필드는 샘플화된 데이터 양(KB 단위)과 압축 사전 작성 시기를 지정합니다. 값 0은 샘플링이 수행되지 않았음을 의미합니다.
- COPY\_ARRAY\_SIZE: COPY\_ARRAY\_INSERT의 ARRAY 크기를 지정합니다. 0 이하의 값은 COPY\_ARRAY\_INSERT를 사용하지 않음을 의미합니다.
- COPY\_INDEXSCHEMA: COPY 단계 중에 목표 테이블의 데이터 클러스터에 사용된 인덱스의 스키마.
- COPY\_INDEXNAME: COPY 단계 중에 목표 테이블의 데이터 클러스터에 사용된 인덱스의 이름.
- REPLAY\_MAX\_ERR\_RETRIES: REPLAY 단계 중에 발생할 수 있는 최대 오류 재시도 계수(잠금 시간종료 또는 교착 상태)를 지정합니다.
- REPLAY\_THRESHOLD: REPLAY 단계를 한 번 반복하는 경우 스테이징 테이블에 적용된 행 수가 이 값보다 적으면 새 항목이 중간에 작성되더라도 REPLAY가 중지됩니다.
- REORG\_USE\_TEMPSPACE: 테이블 이동의 REORG 옵션을 호출하면 REORG 명령의 USE절에 임시 테이블 스페이스를 지정할 수도 있습니다. 여기에서 값이 지정되지 않으면 REORG 명령이 재구성 중인 테이블과 같은 테이블 스페이스를 사용합니다.
- SWAP\_MAX\_RETRIES: (잠금 시간 종료 또는 교착 상태가 발생하는 경우) SWAP 단계 중에 허용된 최대 재시도 수를 지정합니다.

이 매개변수의 데이터 유형은 VARCHAR(128)입니다.

### value

이 입력 매개변수를 ADMIN\_MOVE\_TABLE 프로토콜 테이블에 "upsert"할 값을 지정합니다. 이 매개변수의 데이터 유형은 CLOB(10M)입니다. 이 매개변수는 NULL 또는 비어 있는 문자열입니다.

## 권한 부여

스토어드 프로시저를 호출하는 명시적 권한 부여가 필요하지 않습니다. 그러나 사용된 권한 부여 ID는 ADMIN\_MOVE\_TABLE 스토어드 프로시저 호출에 사용되는 권한 부여 ID와 같아야 합니다.

## 예:

이 예는 압축 값을 갱신하고 목표 테이블 복사에 사용되는 특정 인덱스 정보를 제거하기 위해 스토어드 프로시저에 대한 기본 호출을 설명합니다.

우선 ADMIN\_MOVE\_TABLE 프로토콜 테이블의 값을 갱신 또는 삭제하기 위하여 ADMIN\_MOVE\_TABLE\_UTIL 프로시저를 호출하기 전에 ADMIN\_MOVE\_TABLE 프로시저를 호출하여 테이블 이동 프로세스를 시작합니다.

```
CALL SYSPROC.ADMIN_MOVE_TABLE('SVALENTI','T1','','','','','','','','','INIT')
```

다음으로 DEEP\_COMPRESSION\_SAMPLE 값을 30720 KB로 갱신합니다.

```
CALL SYSPROC.ADMIN_MOVE_TABLE_UTIL('SVALENTI','T1','UPSERT',  
    'DEEPCOMPRESSSION_SAMPLE','30720')
```

이제 COPY\_INDEXSCHEMA 및 COPY\_INDEXNAME 값을 삭제합니다.

```
CALL SYSPROC.ADMIN_MOVE_TABLE_UTIL('SVALENTI','T1','DELETE','COPY_INDEXSCHEMA','')  
CALL SYSPROC.ADMIN_MOVE_TABLE_UTIL('SVALENTI','T1','DELETE','COPY_INDEXNAME','')
```

이 변경사항 후에 메타 테이블의 새 값을 사용하여 ADMIN\_MOVE\_TABLE 프로시저를 계속합니다.

```
CALL SYSPROC.ADMIN_MOVE_TABLE('SVALENTI','T1','','','','','','','','','COPY')  
CALL SYSPROC.ADMIN_MOVE_TABLE('SVALENTI','T1','','','','','','','','','REPLAY')  
CALL SYSPROC.ADMIN_MOVE_TABLE('SVALENTI','T1','','','','','','','','','SWAP')
```

## 사용 시 참고사항

ADMIN\_MOVE\_TABLE 프로토콜 테이블의 변경 가능한 KEY 값에 대한 자세한 정보는 ADMIN\_MOVE\_TABLE 프로시저의 사용법 참고 섹션을 참조하십시오.

---

## ALTOBJ

ALTOBJ 프로시저는 변경할 기존 테이블에 대해 목표 DDL(Data Definition Language)로 지원되는 입력 CREATE TABLE문을 구문 분석합니다. 이 프로시저는 변경 중인 테이블의 데이터를 백업한 다음 원래 테이블을 삭제하고 DDL 명령문을 사용하여 새 버전을 작성합니다. 마지막 단계에서 저장된 데이터를 다시 새 테이블에 로드합니다.

이 프로시저는 다음 테이블 변경 조작을 지원하며 복구 가능한 종속성을 유지합니다.

- 컬럼 이름 바꾸기
- 컬럼의 크기를 늘리거나 줄이기
- DB2 스칼라 함수를 사용한 기존 데이터 변형 및 컬럼 유형 변경
- 정밀도 또는 10진수 값의 스케일 변경
- 컬럼의 기본 값 변경

- 컬럼의 널(NULL) 입력 가능 속성을 널(null) 값을 허용하는 속성으로 변경
- 컬럼 삭제

## 구문

▶▶—ALTOBJ—(—*exec-mode*—,—*sql-stmt*—,—*alter-id*—,—*msg*—)————▶▶

스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *exec-mode*

다음 실행 모드 중 하나를 지정하는 VARCHAR(30) 유형의 입력 인수.

#### ‘GENERATE’

VALIDATE, APPLY 및 UNDO 모드에서 필요한 모든 스크립트가 생성되도록 지정합니다.

#### ‘VALIDATE’

명령문 구문을 유효성 확인하도록 지정합니다. 또한 이 옵션은 스크립트를 생성하여 변경될 테이블에 대한 관계 및 관련된 오브젝트 처리를 관리합니다.

#### ‘APPLY\_CONTINUE\_ON\_ERROR’ 또는 ‘APPLY\_STOP\_ON\_ERROR’

관련된 오브젝트 처리를 관리할 스크립트를 지정하고 변경될 테이블에 대한 관계가 생성됩니다. 원래 테이블의 데이터가 익스포트되거나 변형되거나 새 테이블을 채우는데 사용됩니다.

#### ‘UNDO’

테이블 변경으로 인한 변경을 취소하도록 지정하며 롤백의 경우 조작성은 발생할 수도 있는 오류를 복구할 수 없습니다. 원래 테이블과 생성된 스크립트가 삭제되지 않은 경우에만 이 모드가 가능합니다.

#### ‘FINISH’

이름을 바꾼 원래 테이블을 삭제하도록 지정합니다.

### *sql-stmt*

기존 테이블을 변경하기 위해 템플릿으로 사용될 CREATE TABLE 명령문을 지정하는 VARCHAR(2048) 유형의 입력 인수. *exec-mode*가 ‘GENERATE’인 경우, *sql-stmt*는 널(null) 값이면 안됩니다. 그렇지 않은 경우 *sql-stmt*는 *alter-id*가 -1이 아닌 경우에만 널(NULL) 값이 될 수 있습니다.

### *alter-id*

이 호출로 생성된 명령문 모두를 식별하는 INTEGER 유형의 입력과 출력 인수. -1이 지정되면 새 ID가 생성되며 호출자로 리턴됩니다. 지정된 정수에서 식별되는 기존 명령문을 겹쳐줍니다.

*msg*

지정된 실행 모드에서 테이블 변경 프로세스에서 사용되거나 생성된 모든 SQL문을 표시하기 위해 실행할 수 있는 SQL 쿼리를 포함한 VARCHAR(2048) 유형의 출력 인수.

## 권한 부여

ALTOBJ 프로시저에 대한 EXECUTE 특권.

LOAD 권한을 가진 DBADM 및 SETSESSIONUSER도 필요합니다.

### 예:

예 1: ALTOBJ 프로시저를 실행하여 유형 INTEGER에서 BIGINT까지 T1 테이블의 CL2 컬럼을 변경하십시오. 테이블 T1에 대한 원래 DDL은 다음과 같습니다.

```
CREATE TABLE T1 (CL1 VARCHAR(5), CL2 INTEGER)
```

컬럼 데이터 유형을 변경할 ALTOBJ 프로시저 호출은 다음과 같습니다.

```
CALL SYSPROC.ALTOBJ('APPLY_CONTINUE_ON_ERROR',  
    'CREATE TABLE T1 (CL1 VARCHAR(5), CL2 BIGINT)', -1, ?)
```

주: 다음 오류가 발생하면 APPLHEAPSZ 매개변수 값을 늘려보십시오.

SQL0443N 루틴 "SYSPROC.ALTOBJ"(특정 이름 "ALTOBJ")는 진단 텍스트 "SQL0954"와 함께 오류 SQLSTATE를 리턴합니다. SQLSTATE=38553

예 2: VALIDATE 모드에서 *alter-id* 입력과 함께 ALTOBJ 프로시저를 실행하십시오.

```
CALL SYSPROC.ALTOBJ('VALIDATE', CAST (NULL AS VARCHAR(2048)), 123, ?)
```

## 사용 시 참고사항

프로시저가 테이블을 삭제하고 다시 작성해도 원래 테이블을 작성한 사용자는 테이블 정의자로 남습니다. 그러나 감사가 프로시저를 실행한 사용자에 의해 테이블이 삭제된 후 다시 작성되었음을 표시합니다.

이 프로시저는 다음 테이블 변경 조작을 지원하지 않습니다.

- 구체화된 쿼리 테이블(MQT) 변경은 지원되지 않습니다. MQT를 포함한 테이블 변경이 지원됩니다.
- 유형이 지정된 테이블 변경이 지원되지 않습니다.
- 별칭을 사용하는 리모트 테이블 변경은 지원되지 않습니다.
- 컬럼 시퀀스를 다시 정렬할 수 없습니다.
- 한 번의 호출로 프로시저로 컬럼 추가와 제거 또는 이름 변경과 제거가 지원되지 않지만 컬럼 추가와 이름 변경은 지원됩니다. 정보와 일치되는 컬럼이 아닌 대상 DDL

의 사용에 의해 테이블이 변경된 방법을 나타내기 위한 유일한 방법이기 때문입니다. 다음 규칙은 기존 테이블에서 변경된 테이블로 데이터를 전달 시 ALTOBJ 프로시저에서 따릅니다.

1. 기존 테이블의 컬럼 수가 변경된 테이블과 동일하다면 추가되거나 제거된 컬럼이 없다고 가정합니다. 이 경우 컬럼의 이름만 변경될 수 있으며 컬럼 인덱스와 일치됩니다.
  2. 기존 테이블의 컬럼 수가 변경된 테이블에서 보다 적다면 컬럼이 추가될 것이라고 가정합니다. 컬럼의 이름이 변경될 수 있으며 새 컬럼이 끝에 추가됩니다. 기존 컬럼은 인덱스에서 일치됩니다.
  3. 기존 테이블의 컬럼 수가 변경된 테이블에서 보다 크다면 컬럼이 제거될 것이라고 가정합니다. 컬럼의 이름이 변경될 수 없으며 이름으로 일치될 수 없습니다. 삭제되는 컬럼이 테이블의 기존 컬럼일 수 있습니다.
- 구조화된 유형 UDT 및 참조 유형 UDT는 지원되지 않습니다.
  - 변경된 기본 테이블에서 정의된 MQT는 테이블 프로세스 변경 시 채워지지 않습니다.

ALTOBJ 프로시저를 사용하여 테이블이 변경되고 테이블에 정의된 MQT가 있으면 MQT가 작성되지만 데이터로 채워지지 않습니다.

ALTOBJ 프로시저를 사용하여 테이블이 변경되고 테이블에 정의된 MQT가 있으면, MQT 콘텐츠가 새 기본 테이블로 재빌드되기 때문에 변경된 테이블에서 select 결과의 파트가 아닌 컬럼이 유실됩니다.

다른 세션들에서 지속된 오브젝트 잠금이 없기 때문에 오브젝트 정의는 ALTOBJ 프로시저 호출 간에 변경될 수 있습니다.

테이블과 연관된 테이블 프로파일(runstats 프로파일과 같은)은 이 확장 변경 프로세스를 통해 처리한 후 유실됩니다.

SYSTOOLSPACE는 메타데이터를 저장하기 위해 루틴의 조작 테이블로 사용됩니다. 즉 데이터베이스 오브젝트와 조작을 설명하기 위해 데이터가 사용됩니다.

---

## APPLICATION\_ID

APPLICATION\_ID 함수는 현재 연결의 응용프로그램 ID를 리턴합니다. 결과의 데이터 유형은 VARCHAR(128)입니다.

함수에서 리턴되는 값은 고유하며 함수를 호출하기 전에 설정된 연결이 지속되는 기간에만 유효합니다.



## 구문

```
▶▶ APPLICATION_ID—(—)—————▶▶
```

스키마는 SYSFUN입니다.

예 :

```
SELECT APPLICATION_ID() AS APPL_ID FROM SYSIBM.SYSDUMMY1
```

---

## COMPILATION\_ENV 테이블 함수 - 컴파일 환경 요소 검색

COMPILATION\_ENV 테이블 함수는 컴파일 환경 요소를 리턴합니다.

## 구문

```
▶▶ COMPILATION_ENV—(—compilation-env—)—————▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

*compilation-env*

BLOB(2M) 유형의 입력 인수는 교착 상태 이벤트 모니터가 제공하는 컴파일 환경을 포함합니다.

함수는 NAME VARCHAR(256) 및 VALUE VARCHAR(1024)의 두 개의 컬럼(949 페이지의 표 214 참조)을 리턴합니다. 컴파일 환경 요소 이름의 가능한 값은 949 페이지의 표 215에 설명되어 있습니다.

최초 요소 값은 SQL문이 동적으로 실행되었는지 패키지의 일부로 바인드되었는지 여부에 따라 기본적으로 달라집니다.

컴파일 환경에서 항목의 수와 유형은 DB2 데이터베이스 관리 프로그램에 성능이 추가됨에 따라 변경될 수 있습니다. 컴파일 환경이 함수를 실행 중인 레벨과 다른 DB2 데이터베이스 관리 프로그램 레벨에서 온 경우, 함수 레벨로 인식되는 요소만 리턴됩니다. 요소에 대한 설명은 릴리스마다 다를 수도 있습니다.

예:

예 1: 이전에 교착 상태 이벤트 모니터에서 캡처된 특정 컴파일 환경의 모든 요소를 요청하십시오. WITH DETAILS HISTORY 옵션을 지정하여 작성되는 교착 상태 이벤트 모니터는 동적 SQL문에 대한 컴파일 환경을 캡처합니다. 이 캡처된 환경은 테이블 함수의 입력으로 승인된 것입니다.



```
SELECT NAME, VALUE
FROM TABLE(SYSPROC.COMPILOATION_ENV(:hv1)) AS t
```

예 2: 컴파일 환경의 특정 요소(디폴트 스키마)를 요청하십시오.

```
SELECT NAME, VALUE
FROM TABLE(SYSPROC.COMPILOATION_ENV(:hv1)) AS t
WHERE NAME = 'SCHEMA'
```

## 리턴되는 정보

표 214. COMPILATION\_ENV 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명
이름	VARCHAR (256)	컴파일 환경 요소. 자세한 내용은 표 215의 내용을 참조하십시오.
VALUE	VARCHAR(1024)	요소의 값

표 215. COMPILATION\_ENV 테이블 함수가 리턴하는 컴파일 환경 요소

요소 이름	설명
ISOLATION	SQL 컴파일러에 전달되는 분리 레벨. 값은 CURRENT ISOLATION 특수 레지스터 또는 현재 패키지의 ISOLATION 바인드 옵션에서 가져옵니다.
QUERY_OPTIMIZATION	SQL 컴파일러에 전달되는 쿼리 최적화 레벨. 값은 CURRENT QUERY OPTIMIZATION 특수 레지스터 또는 현재 패키지의 QUERYOPT 바인드 옵션에서 가져옵니다.
MIN_DEC_DIV_3	SQL 컴파일러에 전달되는 요청된 10진수 계산 스케일. 값은 <i>min_dec_div_3</i> 데이터베이스 구성 매개변수에서 가져옵니다.
DEGREE	SQL 컴파일러에 전달되는 요청된 내부 병렬구조 등급. 값은 CURRENT DEGREE 특수 레지스터 또는 현재 패키지의 DEGREE 바인드 옵션에서 가져옵니다.
SQLRULES	SQL 컴파일러에 전달되는 요청된 SQL문 동작. 값은 현재 패키지의 LANGLVL 바인드 옵션 설정값에서 생성됩니다. 가능한 값은 'DB2' 또는 'SQL92'입니다.
REFRESH_AGE	SQL 컴파일러에 전달되는 허용 가능한 데이터 지연. 값은 CURRENT REFRESH AGE 특수 레지스터 또는 현재 패키지의 REFRESHAGE 바인드 옵션에서 가져옵니다.
SCHEMA	SQL 컴파일러에 전달되는 디폴트 스키마. 값은 CURRENT SCHEMA 특수 레지스터 또는 현재 패키지의 QUALIFIER 바인드 옵션에서 가져옵니다.
PATH	SQL 컴파일러에 전달되는 함수 경로. 값은 CURRENT PATH 특수 레지스터 또는 현재 패키지의 FUNC_PATH 바인드 옵션에서 가져옵니다.
TRANSFORM_GROUP	SQL 컴파일러에 전달되는 변환 그룹 정보. 값은 CURRENT DEFAULT TRANSFORM GROUP 특수 레지스터 또는 현재 패키지의 TRANSFORMGROUP 바인드 옵션에서 가져옵니다.
MAINTAINED_TABLE_TYPE	SQL 컴파일러에 전달되며 최적화를 위해 고려해볼 수 있는 테이블 유형 표시기. 값은 CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION 특수 레지스터에서 가져옵니다.

표 215. COMPILATION\_ENV 테이블 함수가 리턴하는 컴파일 환경 요소 (계속)

요소 이름	설명
RESOLUTION_TIMESTAMP	SQL문에서 함수 및 데이터 유형 참조와 같은 항목을 해석하기 위해 SQL 컴파일러에서 사용하는 시간소인. 이 시간소인은 현재 시간소인이거나 현재 패키지의 마지막 명시적 바인드 조작의 시간소인입니다.
FEDERATED_ASYNCRONY	SQL 컴파일러에 전달되는 페더레이티드 비동기 병렬 처리의 요청된 등급. 값은 CURRENT FEDERATED ASYNCHRONY 특수 레지스터 또는 현재 패키지의 FEDERATED_ASYNCRONY 바인드 옵션에서 가져옵니다.

## CONTACTGROUPS 관리 뷰 - 문의처 그룹 목록 검색

CONTACTGROUPS 관리 뷰는 문의처 시스템 또는 전역 목록에서 로컬로 정의할 수 있는 문의처 그룹 목록을 리턴합니다. DAS(Database Administration Server) CONTACT\_HOST 구성 매개변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- CONTACTGROUPS 관리 뷰에 대한 SELECT 특권
- CONTACTGROUPS 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- ADMIN\_GET\_CONTACTGROUPS 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

예 :

모든 문의처 그룹 목록을 검색하십시오.

```
SELECT * FROM SYSIBMADM.CONTACTGROUPS
```

다음은 이 쿼리에 대한 출력의 예입니다.

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          group7          name1          CONTACT

```

7개의 레코드가 선택되었습니다.

## 사용법 참고

DAS를 작성하여 실행해야 합니다.

## 리턴되는 정보

표 216. CONTACTGROUPS 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
NAME	VARCHAR (128)	문의처 그룹 이름
DESCRIPTION	VARCHAR (128)	문의처 그룹 설명.
MEMBERNAME	VARCHAR (128)	문의처 그룹의 구성원 이름. 이 이름은 문의처 또는 다른 문의처 그룹을 참조할 수 있습니다.
MEMBERTYPE	VARCHAR(7)	문의처 그룹의 구성원 유형. 유형은 CONTACT 또는 GROUP입니다.

## CONTACTS 관리 뷰 - 문의처 목록 검색

CONTACTS 관리 뷰는 데이터베이스 서버에 정의된 문의처 목록을 리턴합니다. DAS(Database Administration Server) CONTACT\_HOST 구성 매개변수의 설정은 목록이 로컬인지 또는 전역인지를 판별합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- CONTACTS 관리 뷰에 대한 SELECT 특권
- CONTACTS 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- ADMIN\_GET\_CONTACTS 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

예 :

모든 문의처를 검색하십시오.

```
SELECT * FROM SYSIBMADM.CONTACTS
```

다음은 이 쿼리에 대한 출력의 예입니다.

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

4개의 레코드가 선택되었습니다.

## 사용법 참고

DAS를 작성하여 실행해야 합니다.

## 리턴되는 정보

표 217. CONTACTS 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
NAME	VARCHAR (128)	문의처 이름
TYPE	VARCHAR(5)	문의처 유형 • 'EMAIL' • 'PAGE'
ADDRESS	VARCHAR (128)	받는 사람의 SMTP 편지함 주소. 예: joe@somewhere.org
MAX_PAGE_LENGTH	INTEGER	최대 메시지 길이. 페이징 서비스에 메시지 길이 제한이 있는 경우, 예 에서 사용됩니다.
DESCRIPTION	VARCHAR (128)	문의처 설명

---

## DB\_HISTORY 관리 뷰 - 실행기록 파일 정보 검색

DB\_HISTORY 관리 뷰는 모든 데이터베이스 파티션의 실행기록 파일로부터 정보를 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- DB\_HISTORY 관리 뷰에 대한 SELECT 특권

- DB\_HISTORY 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- ADMIN\_LIST\_HIST 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

예 :

클라이언트가 현재 연결된 데이터베이스의 모든 데이터베이스 파티션에 대한 데이터베이스 실행기록 파일에서 데이터베이스 파티션 번호, 항목 ID, 조작, 시작 시간 및 상태 정보를 선택하십시오.

```
SELECT DBPARTITIONNUM, EID, OPERATION, START_TIME, ENTRY_STATUS
FROM SYSIBMADM.DB_HISTORY
```

다음은 이 쿼리에 대한 출력의 예입니다.

DBPARTITIONNUM	EID	OPERATION	START_TIME	ENTRY_STATUS
0		1 A	20051109185510	A

1개의 레코드가 선택되었습니다.

## 리턴된 정보

표 218. DB\_HISTORY 관리 뷰로 리턴된 정보

컬럼 이름	데이터 유형	설명
DBPARTITIONNUM	SMALLINT	데이터베이스 파티션 번호.
EID	BIGINT	실행기록 파일의 항목을 고유하게 식별한 번호
START_TIME	VARCHAR(14)	로그된 이벤트의 시작을 표시하는 시간소인
SEQNUM	SMALLINT	시퀀스 번호
END_TIME	VARCHAR(14)	로그된 이벤트의 종료를 표시하는 시간소인
FIRSTLOG	VARCHAR(254)	이벤트와 연관된 최초 트랜잭션 로그의 이름
LASTLOG	VARCHAR(254)	이벤트와 연관된 최신 트랜잭션 로그의 이름
BACKUP_ID	VARCHAR(24)	백업 ID 또는 고유 테이블 ID
TABSCHEMA	VARCHAR(128)	테이블 스키마
TABNAME	VARCHAR(128)	테이블 이름.
COMMENT	VARCHAR(254)	로그된 이벤트와 연관된 시스템 생성 주석 텍스트
CMD_TEXT	CLOB(2 M)	로그된 이벤트와 연관된 DDL

표 218. DB\_HISTORY 관리 뷰로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
NUM_TBSPS	INTEGER	로그된 이벤트와 연관된 테이블 스페이스 수
TBSPNAMES	CLOB(5 M)	로그된 이벤트와 연관된 테이블 스페이스 이름
OPERATION	CHAR(1)	조작 ID. 가능한 값은 955 페이지의 표 219의 내용을 참조하십시오.
OPERATIONTYPE	CHAR(1)	조작에 대한 조치 ID. 가능한 값은 955 페이지의 표 219의 내용을 참조하십시오.
OBJECTTYPE	CHAR(1)	조작의 목표 오브젝트에 대한 ID. 가능한 값은 다음과 같습니다. 전체 데이터베이스에 D, 테이블 스페이스에 P, 테이블에 T입니다.
LOCATION	VARCHAR(255)	로그된 이벤트와 연관된 파일의 전체 경로 이름(예: 백업 이미지 또는 로드 입력 파일)
DEVICETYPE	CHAR(1)	로그된 이벤트와 연관된 디바이스 유형에 대한 ID. 이 필드는 LOCATION 필드가 해석되는 방식을 판별합니다. 가능한 값은 다음과 같습니다. TSM에 A, 클라이언트에 C, 디스크에 D, 스냅샷 백업에 F, 디스켓에 K, 로컬에 L, N(DB2로 내부적으로 생성된), 기타에 O(기타 벤더 디바이스 지원의 경우), 파이프에 P, 커서에 Q, 리모트 페치 데이터에 R, 서버에 S, 테이프에 T, User Exit에 U, X/Open XBSA 인터페이스에 X입니다.
ENTRY_STATUS	CHAR(1)	실행기록 파일에 있는 항목의 상태에 대한 ID. 가능한 값은 다음과 같습니다. 활성화에 A, 삭제됨에 D(나중에 사용), 만기됨에 E, 비활성에 I, 아직 커밋되지 않음에 N, 커밋됨 또는 활성화에 Y입니다.
SQLCAID	VARCHAR(8)	'SQLCA'를 포함하는 스토리지 덤 프로그 "eye catcher"로, SQL 통신 영역(SQLCA)의 SQLCAID 필드에 표시됩니다.
SQLCABC	INTEGER	SQLCA의 길이로, SQLCA의 SQLCABC 필드에 표시됩니다.
SQLCODE	INTEGER	SQL 리턴 코드로, SQLCA의 SQLCODE 필드에 표시됩니다.

표 218. DB\_HISTORY 관리 뷰로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
SQLERRML	SMALLINT	SQLERRMC에 대한 길이 표시기로, SQLCA의 SQLERRML 필드에 표시됩니다.
SQLERRMC	VARCHAR(70)	하나 이상의 토큰을 포함하고 X'FF'로 구분되며, SQLCA의 SQLERRMC 필드에 표시됩니다. 이러한 토큰은 오류 조건의 설명에 있는 변수로 대체됩니다.
SQLERRP	VARCHAR(8)	제품의 수정 레벨과 버전, 릴리스를 나타내는 5개의 영숫자가 따라오며 제품을 나타내는 세 개의 문자 ID로, SQLCA의 SQLERRP 필드에 표시됩니다.
SQLERRD1	INTEGER	SQLCA(SQL 통신 영역)를 참조하십시오.
SQLERRD2	INTEGER	SQLCA(SQL 통신 영역)를 참조하십시오.
SQLERRD3	INTEGER	SQLCA(SQL 통신 영역)를 참조하십시오.
SQLERRD4	INTEGER	SQLCA(SQL 통신 영역)를 참조하십시오.
SQLERRD5	INTEGER	SQLCA(SQL 통신 영역)를 참조하십시오.
SQLERRD6	INTEGER	SQLCA(SQL 통신 영역)를 참조하십시오.
SQLWARN	VARCHAR(11)	공백이나 'W'를 포함하는 각 경고 표시기의 설정 SQLCA(SQL 통신 영역)를 참조하십시오.
SQLSTATE	VARCHAR(5)	최근에 실행된 SQL문의 결과를 나타내는 리턴 코드로, SQLCA의 SQLSTATE 필드에서 표시됩니다.

표 219. OPERATION 및 OPERATIONTYPE 값

조작 값	조작 값 설명	조작 유형
A	테이블 스페이스 추가	없음
B	백업	조작 유형은 다음과 같습니다. <ul style="list-style-type: none"> <li>• D = 델타 오프라인</li> <li>• E = 델타 온라인</li> <li>• F = 오프라인</li> <li>• I = 증분 오프라인</li> <li>• N = 온라인</li> <li>• O = 증분 온라인</li> </ul>
C	로드 사본	없음

표 219. OPERATION 및 OPERATIONTYPE 값 (계속)

조작 값	조작 값 설명	조작 유형
D	삭제된 테이블	없음
F	롤 포워드	조작 유형은 다음과 같습니다. • E = 로그 종료 • P = 특정 시점
G	테이블 재구성	조작 유형은 다음과 같습니다. • F = 오프라인 • N = 온라인
L	로드	조작 유형은 다음과 같습니다. • I = 삽입 • R = 교체
N	테이블 스페이스 이름 바꾸기	없음
O	테이블 스페이스 삭제	없음
Q	Quiesce	조작 유형은 다음과 같습니다. • S = Quiesce 상태 공유 • U = Quiesce 상태 갱신 • X = Quiesce 독점 • Z = Quiesce 재설정
R	리스트어	조작 유형은 다음과 같습니다. • F = 오프라인 • I = 증분 오프라인 • N = 온라인 • O = 증분 온라인 • R = 재빌드
T	테이블 스페이스 변경	조작 유형은 다음과 같습니다. • C = 컨테이너 추가 • R = 재조정
U	언로드	없음
X	아카이브 로그	조작 유형은 다음과 같습니다. • F = 아카이브 경로 실패 • M = 미러 로그 경로 • N = ARCHIVE LOG 명령으로 강제 실행된 절단 • P = 1차 로그 경로 • 1 = 첫 번째 로그 아카이브 메소드 • 2 = 두 번째 로그 아카이브 메소드



## DBPATHS 관리 뷰 - 데이터베이스 경로 검색

DBPATHS 관리 뷰는 분할 미리 백업과 같은 태스크에 필요한 데이터베이스 경로에 대한 값을 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- DBPATHS 관리 뷰에 대한 SELECT 특권
- DBPATHS 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- ADMIN\_LIST\_DB\_PATHS 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

### 예 :

모든 데이터베이스 경로를 검색하십시오.

```
SELECT * FROM SYSIBMADM.DBPATHS
```

다음은 이 쿼리에 대한 출력의 예입니다.

```
DBPARTITIONNUM TYPE          ...
-----
          0 LOGPATH           ...
          0 MIRRORLOGPATH      ...
          0 DB_STORAGE_PATH    ...
          0 DB_STORAGE_PATH    ...
          0 TBSP_CONTAINER     ...
          0 TBSP_CONTAINER     ...
          0 TBSP_CONTAINER     ...
          0 TBSP_DIRECTORY    ...
          0 TBSP_DIRECTORY    ...
          0 LOCAL_DB_DIRECTORY ...
          0 DBPATH             ...
```

11개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
...      PATH
... -----
... S:\wdbfiles\INST5\NODE0000\SQL00001\SQLLOGDIRW
... S:\wmirrorlogs\NODE0000#
... S:\wdbfiles#
... S:\wdbfile2#
... S:\wdbfiles\INST5\NODE0000\SQL00001\TS3
```

```

... S:\wdbfiles\INST5\NODE0000\SQL00001\long3
... S:\wdbfiles\INST5\NODE0000\SQL00001\regular05
... S:\wdbfiles\INST5\NODE0000\SQL00001\usertemp3\
... S:\wdbfiles\INST5\NODE0000\SQL00001\systemp3\
... S:\wdbfiles\INST5\NODE0000\SQLDBDIR\
... S:\wdbfiles\INST5\NODE0000\SQL00001\

```

## 리턴되는 정보

표 220. DBPATHS 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
DBPARTITIONNUM	SMALLINT	데이터베이스 파티션 번호.
TYPE	VARCHAR(64)	경로가 속한 데이터베이스 오브젝트 유형을 설명합니다. 예를 들면, LOGPATH 데이터베이스 구성 매개변수로 표시되는 로그 디렉토리에 대한 경로가 LOGPATH로 이 컬럼에 표시됩니다. 표 221에서 리턴 가능한 값의 목록을 참조하십시오.
PATH	VARCHAR(5000)	데이터베이스 관리 프로그램에 파일 또는 디렉토리가 있는 위치에 대한 경로. 경로가 파일 시스템 분리문자(UNIX 환경의 경우 '/', Windows 환경의 경우 '\')로 끝나면, 경로는 디렉토리를 가리킵니다.

표 221. TYPE 컬럼 값

유형 값	설명
TBSP_DEVICE	데이터베이스 관리 스페이스(DMS) 테이블 스페이스에 대한 원시 디바이스
TBSP_CONTAINER	DMS 테이블 스페이스에 대한 파일 컨테이너
TBSP_DIRECTORY	시스템 관리 스페이스(SMS) 테이블 스페이스에 대한 디렉토리.
LOGPATH	1차 로그 경로
LOGPATH_DEVICE	1차 로그 경로의 원시 디바이스
MIRRORLOGPATH	데이터베이스 구성 미러 로그 경로
DB_STORAGE_PATH	자동 스토리지 경로
DBPATH	데이터베이스 디렉토리 경로
LOCAL_DB_DIRECTORY	로컬 데이터베이스 디렉토리 경로.

- 자동 스토리지를 사용하는 테이블 스페이스의 경우, 사용된 스토리지 경로와 사용되지 않은 스토리지 경로 둘 모두 리턴됩니다. 사용되지 않은 자동 스토리지 경로는 분할 미러 백업을 리스토어하는 경우에 필요합니다. 다음 예를 고려하십시오. 분할 미러 백업은 프로덕션 시스템에서 수행됩니다. 백업이 완료된 후에, 백업 전에 사용되지 않았던 자동 스토리지 경로가 현재 프로덕션에서 사용 중입니다. 이제 분할 미러

백업을 리스토어해야 한다고 가정합니다. 이 시점에서, 프로덕션 데이터베이스에서 로그를 롤 포워드해야 합니다. 로그를 롤 포워드하려면, 모든 자동 스토리지 경로가 현재 사용 중이기 때문에 모든 자동 스토리지 경로가 필요합니다.

- 자동 스토리지에서 관리하는 테이블 스페이스 컨테이너는 개별적으로 리턴되지 않습니다. 대신, 자동 스토리지 경로 컬럼에 반영됩니다.
- 자동 스토리지 경로는 데이터베이스 파티션당 한 번만 리턴됩니다.
- LOGPATH 및 MIRRORLOGPATH에 대해 리턴된 값은 메모리에 저장된 값입니다. 데이터베이스 재시작 후에만 적용 가능한 디스크에 저장된 변경된 값은 리턴되지 않습니다.
- db2relocatedb 구성 파일(데이터베이스 재위치에 필요한 구성 정보를 포함하는 파일)을 작성하는 데 SELECT \* FROM SYSIBMADM.DBPATHS 출력을 사용하는 경우, 구성 파일에서 해당 출력을 사용하려면 먼저 DBPATH 출력을 적절하게 수정해야 합니다.

예를 들어, 다음 DBPATH 출력의 경우를 보십시오.

```
/storage/svtdbm3/svtdbm3/NODE0000/SQL00001/
```

이 출력은 다음과 같이 db2relocatedb 구성 파일에서 DB\_PATH 매개변수를 지정하는 데 사용될 수 있습니다.

```
DB_PATH=/storage/svtdbm3,/storage_copy2/svtdbm3
```

- LOCAL\_DB\_DIRECTORY 경로는 여러 데이터베이스에 있는 정보를 포함할 수 있습니다. sqldbdir은 같은 디렉토리에서 작성된 여러 데이터베이스를 구분하지 않기 때문에 파일을 복사할 대상 시스템의 해당 경로에 같은 데이터베이스가 있는지 미리 확인하십시오.
- 두 개 이상의 데이터베이스가 최소한 하나의 자동 스토리지 경로를 공유하는 경우, 분리하지 않을 데이터베이스에 대한 I/O 문제점 때문에 이 데이터베이스 중 하나에 대한 분리 미리 조작용 둘 이상의 데이터베이스에 영향을 미칠 수 있습니다.

## 제한사항

데이터베이스가 WRITE SUSPEND 모드일 때 이 관리 보기를 호출할 수 없습니다. 데이터베이스 관리자는 데이터베이스의 실제 레이아웃이 보기 호출시점과 분할 미리 조작용을 수행하는 데 필요한 WRITE SUSPEND 모드 활성화 시점 사이에서 변경되지 않았는지 확인해야 합니다. 예를 들어, 테이블 스페이스 레이아웃이 해당 시점에서 변경된 경우 분할 미리 백업 이미지가 성공적으로 복원되지 않을 수도 있습니다.

---

## EXPLAIN\_FORMAT\_STATS

이 새 스칼라 함수는 지정된 쿼리에 대해 캡처된 Explain 스냅샷에서 구문 분석되고 추출되는 형식화된 통계 정보를 표시하는 데 사용됩니다. 결과의 데이터 유형은 CLOB(50M)입니다.

## 구문

▶▶—EXPLAIN\_FORMAT\_STATS—(—*snapshot*—)————▶▶

스키마는 SYSPROC입니다.

## 함수 매개변수

### *snapshot*

제공된 쿼리에 대해 캡처된 Explain 스냅샷인 BLOB(10M) 유형의 입력 인수.  
Explain 테이블 *EXPLAIN\_STATEMENT*의 스냅샷 컬럼으로 저장됩니다.

## 권한 부여

EXPLAIN\_FORMAT\_STATS 함수에 대한 EXECUTE 특권

## 예 :

```
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
```

다음은 이 함수의 샘플 출력입니다.

테이블 스페이스 컨텍스트:

```
-----
이름:                                USERSPACE1
오버헤드:                            7.500000
전송률:                              0.060000
프리페치 크기:                       32
Extent 크기:                          32
유형:                                 관리된 데이터베이스
파티션 그룹 이름:                     NULLP
버퍼 풀 ID:                           0
```

기본 테이블 통계:

```
-----
이름 : T1
스키마: DB2USER2
컬럼 수:                              3
행이 있는 페이지 수:                 1
페이지 수:                            1
행 수:                                 5
테이블 오버플로우 레코드 계수:      0
행 너비:                              26
작성 시간:                            2006-06-16-11.46.53.041085
마지막 통계 갱신:                    2006-06-26-12.23.44.814201
통계 유형:                            Fabrication
```

```

기본 테이블 스페이스:          USERSPACE1
인덱스용 테이블 스페이스:     USERSPACE1
Long 데이터용 테이블 스페이스: NULLP
참조된 컬럼 수:                2
인덱스 수:                    1
일시적 테이블:                아니오
테이블 활성 블록:            1
컬럼 그룹 수:                 0
데이터 파티션 수:             1
평균 행 압축 비율:            -9.000000
압축된 퍼센트 행:            -9.000000
평균 압축된 행 크기:         -9
통계 유형:                    U

```

컬럼 정보:

```

-----
개수:                          1
이름:                          C1
사용 가능 통계:                 예

```

컬럼 통계:

```

-----
컬럼 유형의 스키마 이름:          SYSIBM
컬럼 유형 이름:                 INTEGER
최대 컬럼 길이:                 4
10진수 컬럼 스케일:             0
구별 컬럼 값 수:                4
평균 컬럼 길이:                 5
가장 자주 사용되는 값 수:       1
변위치 수:                      5
두 번째로 높은 데이터 값:       3
두 번째로 낮은 데이터 값:     2
파티션 키의 컬럼 시퀀스:        0
하위 요소의 평균 수:            -1
분리문자의 평균 길이:          -1

```

컬럼 분포 통계:

```

-----
빈도 통계:
Valcount  값
-----
2          1

Quantile 통계:
Valcount  할인  값
-----
          0          1          1
          2          1          1
          3          2          2
          4          3          3
          5          4          4

```

컬럼 정보:

```

-----
개수:                          2
이름:                          C2
사용 가능 통계:                 예

```

컬럼 통계:

컬럼 유형의 스키마 이름:	SYSIBM
컬럼 유형 이름:	INTEGER
최대 컬럼 길이:	4
10진수 컬럼 스케일:	0
구별 컬럼 값 수:	4
평균 컬럼 길이:	5
가장 자주 사용되는 값 수:	1
변위치 수:	5
두 번째로 높은 데이터 값:	3
두 번째로 낮은 데이터 값:	2
파티션 키의 컬럼 시퀀스:	0
하위 요소의 평균 수:	-1
분리문자의 평균 길이:	-1

컬럼 분포 통계:

빈도 통계:

Valcount	값
2	1

Quantile 통계:

	Valcount	할인	값
0	0	1	
	2	0	1
	3	0	2
4	0	4	
5	0	4	

테이블에 정의된 인덱스:

이름	:IDX_T1C1C2
스키마	:DB2USER2
고유한 규칙:	중복 인덱스
연산자에 사용:	예
페이지 페치 쌍:	사용 불가능
컬럼 수:	2
인덱스 단말 페이지:	1
인덱스 트리 레벨:	1
인덱스 첫 번째 키 카디널리티(cardinality):	4
인덱스 전체 키 카디널리티(cardinality):	4
인덱스 클러스터 비율:	100
인덱스 클러스터 인수:	-1.000000
작성 시간:	2006-06-16-11.46.53.596717
마지막 통계 갱신:	2006-06-26-12.23.44.814201
인덱스 순차 페이지:	0
인덱스 첫 번째 2 키 카디널리티(cardinality):	4
인덱스 첫 번째 3 키 카디널리티(cardinality):	-1
인덱스 첫 번째 4 키 카디널리티(cardinality):	-1
시퀀스 간 인덱스 평균 갭:	0.000000
시퀀스 간 페치 평균 갭:	-1.000000
인덱스 평균 순차 페이지:	0.000000
페치 평균 순차 페이지:	-1.000000
인덱스 평균 무작위 페이지:	1.000000
페치 평균 무작위 페이지:	-1.000000

인덱스 RID 계수:	5	
인덱스 삭제된 RID 계수:	0	
인덱스 비어 있는 리프 페이지:	0	
평균 파티션 클러스터 비율:	-1	
평균 파티션 클러스터 인수:	-1.000000	
데이터 파티션 클러스터 인수:	1.000000	
데이터 파티션 페이지 페치 쌍:		사용 불가능

기본 테이블 통계:

```

-----
이름      : T1
스키마    : DB2USER2
컬럼 수   : 3
행이 있는 페이지 수: 1
페이지 수: 1
행 수     : 2
테이블 오버플로우 레코드 계수: 0
행 너비   : 26
작성 시간 : 2006-06-16-11.46.53.398092
마지막 통계 갱신: 2006-06-26-12.23.45.157028
통계 유형 : 동기
기본 테이블 스페이스: USERSPACE1
인덱스용 테이블 스페이스: USERSPACE1
Long 데이터용 테이블 스페이스: NULLP
참조된 컬럼 수: 2
인덱스 수 : 1
일시적 테이블: 아니오
테이블 활성 블록: -1
컬럼 그룹 수: 0
데이터 파티션 수: 1

```

컬럼 정보:

```

-----
개수      : 1
이름      : C1
사용 가능 통계: 예

```

컬럼 통계:

```

-----
컬럼 유형의 스키마 이름:          SYSIBM
컬럼 유형 이름:                INTEGER
최대 컬럼 길이:                 4
10진수 컬럼 스케일:             0
구별 컬럼 값 수:                 2
평균 컬럼 길이:                  5
가장 자주 사용되는 값 수:       -1
변위치 수:                       2
두 번째로 높은 데이터 값:        2
두 번째로 낮은 데이터 값:       1
파티션 키의 컬럼 시퀀스:         0
하위 요소의 평균 수:              -1
분리문자의 평균 길이:            -1

```

컬럼 분포 통계:

```

-----
Quantile 통계:
      Valcount   할인   값
-----

```

```

1          1          1
           2          2          2

```

컬럼 정보:

```

-----
개수:                2
이름:                C2
사용 가능 통계:     예

```

컬럼 통계:

```

-----
컬럼 유형의 스키마 이름:                SYSIBM
컬럼 유형 이름:                        INTEGER
최대 컬럼 길이:                          4
10진수 컬럼 스케일:                      0
구별 컬럼 값 수:                          2
평균 컬럼 길이:                          5
가장 자주 사용되는 값 수:                 -1
변위치 수:                                2
두 번째로 높은 데이터 값:                 2
두 번째로 낮은 데이터 값:               1
파티션 키의 컬럼 시퀀스:                  0
하위 요소의 평균 수:                      -1
분리문자의 평균 길이:                    -1

```

컬럼 분포 통계:

```

-----
Quantile 통계:
          Valcount   할인   값
-----
          1          0          1
          2          0          2

```

테이블에 정의된 인덱스:

```

-----
이름   :IDX_T2C1
스키마: DB2USER2
고유한 규칙:                중복 인덱스
연산자에 사용:              아니오
페이지 페치 쌍:             사용 불가능
컬럼 수: 1
인덱스 단말 페이지:         1
인덱스 트리 레벨:           1
인덱스 첫 번째 키 카디널리티(cardinality): 2
인덱스 전체 키 카디널리티(cardinality): 2
인덱스 클러스터 비율:       100
인덱스 클러스터 인수:       -1.000000
작성 시간:                   2006-06-16-11.46.53.857520
마지막 통계 갱신:           2006-06-26-12.23.45.157028
인덱스 순차 페이지:         0
  인덱스 첫 번째 2 키 카디널리티(cardinality): -1
  인덱스 첫 번째 3 키 카디널리티(cardinality): -1
  인덱스 첫 번째 4 키 카디널리티(cardinality): -1
시퀀스 간 인덱스 평균 갭:    0.000000
시퀀스 간 페치 평균 갭:      -1.000000
인덱스 평균 순차 페이지:     0.000000
페치 평균 순차 페이지:       -1.000000
인덱스 평균 무작위 페이지:   1.000000

```



페치 평균 무작위 페이지:	-1.000000	
인덱스 RID 계수:	2	
인덱스 삭제된 RID 계수:	0	
인덱스 비어 있는 리프 페이지:	0	
평균 파티션 클러스터 비율:	-1	
평균 파티션 클러스터 인수:	-1.000000	
데이터 파티션 클러스터 인수:	1.000000	
데이터 파티션 페이지 페치 쌍:		사용 불가능

## EXPLAIN\_GET\_MSGS

EXPLAIN\_GET\_MSGS 테이블 함수는 EXPLAIN\_DIAGNOSTIC 및 EXPLAIN\_DIAGNOSTIC\_DATA Explain 테이블을 쿼리하고 형식화된 메시지를 리턴합니다.

### 구문

```
►►—EXPLAIN_GET_MSGS—(—explain-requester—,—explain-time—,—source-name—,—►
►—source-schema—,—source-version—,—explain-level—,—stmtno—,—sectno—,—►
►—locale—)—►►
```

스키마는 Explain 테이블 스키마와 동일합니다.

### 테이블 함수 매개변수

다음 입력 인수는 널(NULL) 값일 수 있습니다. 인수가 널(NULL) 값인 경우 쿼리를 제한하는 데 사용되지 않습니다.

#### *explain-requester*

해당 Explain 요청 초기자의 권한 부여 ID를 지정하는 VARCHAR (128) 유형의 입력 인수. 널(NULL) 값은 쿼리 검색 조건에서 이 매개변수를 제외합니다.

#### *explain-time*

Explain 요청 시작 시간을 지정하는 TIMESTAMP 유형의 입력 인수. 널(NULL) 값은 쿼리 검색 조건에서 이 매개변수를 제외합니다.

#### *source-name*

동적 명령문이 Explain될 때 실행 중인 패키지 이름 또는 정적 SQL문이 Explain될 때의 소스 파일 이름을 지정하는 VARCHAR (128) 유형의 입력 인수. 널(NULL) 값은 쿼리 검색 조건에서 이 매개변수를 제외합니다.

### *source-schema*

Explain 요청 소스의 스키마 또는 규정자를 지정하는 VARCHAR (128) 유형의 입력 인수. 널(NULL) 값은 쿼리 검색 조건에서 이 매개변수를 제외합니다.

### *source-version*

Explain 요청 소스의 버전을 지정하는 VARCHAR(64) 유형의 입력 인수. 널(NULL) 값은 쿼리 검색 조건에서 이 매개변수를 제외합니다.

### *explain-level*

이 행이 관련된 Explain 정보 레벨을 지정하는 CHAR(1) 유형의 입력 인수. 널(NULL) 값은 쿼리 검색 조건에서 이 매개변수를 제외합니다.

### *stmtno*

이 Explain 정보가 관련된 패키지 안에 명령문 번호를 지정하는 INTEGER 유형의 입력 인수. 널(NULL) 값은 쿼리 검색 조건에서 이 매개변수를 제외합니다.

### *sectno*

이 Explain 정보가 관련된 패키지 안에 섹션 번호를 지정하는 INTEGER 유형의 입력 인수. 널(NULL) 값은 쿼리 검색 조건에서 이 매개변수를 제외합니다.

### *locale*

리턴된 메시지의 로케일을 지정하는 VARCHAR(33) 유형의 입력 인수. 지정된 로케일이 DB2 서버에 설치되지 않은 경우 값은 무시됩니다.

## 리턴되는 정보

표 222. EXPLAIN\_GET\_MSGS 테이블 함수가 리턴한 정보

컬럼 이름	데이터 유형	설명
EXPLAIN_REQUESTER	VARCHAR (128)	이 Explain 요청 초기자의 권한 부여 ID
EXPLAIN_TIME	TIMESTAMP	Explain 요청 시작 시간
SOURCE_NAME	VARCHAR (128)	동적 명령문이 Explain될 때 실행 중인 패키지 이름 또는 정적 SQL문이 Explain될 때의 소스 파일 이름
SOURCE_SCHEMA	VARCHAR (128)	Explain 요청 소스의 스키마 또는 규정자
SOURCE_VERSION	VARCHAR(64)	Explain 요청의 소스 버전
EXPLAIN_LEVEL	CHAR(1)	이 행과 관련된 Explain 정보의 레벨
STMTNO	INTEGER	이 Explain 정보가 관련된 패키지 안의 명령문 번호
SECTNO	INTEGER	이 Explain 정보가 관련된 패키지 안의 섹션 번호
DIAGNOSTIC_ID	INTEGER	EXPLAIN_STATEMENT 테이블에 있는 명령문의 특정 인스턴스에 대한 진단의 ID.

표 222. EXPLAIN\_GET\_MSGS 테이블 함수가 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명
LOCALE	VARCHAR(33)	리턴된 메시지의 로케일. 이 로케일은 지정된 로케일이 DB2 서버에 설치되어 있지 않은 경우 지정된 로케일과 일치하지 않습니다.
MSG	VARCHAR(4096)	형식화된 메시지 텍스트

## 예:

마지막에 생성된 리퀘스터 SIMMEN의 디폴트 스키마에 있는 Explain 테이블로부터 형식화된 영어 메시지를 요청하십시오. 소스 이름 SQLC2E03을 지정하십시오.

```

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
    
```

이 쿼리의 출력 예는 다음과 같습니다.

```

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".
    
```

## GET\_DBSIZE\_INFO

GET\_DBSIZE\_INFO 프로시저가 데이터베이스 크기 및 최대 용량을 계산합니다.

### 구문

```

▶▶—GET_DBSIZE_INFO—(—snapshot-timestamp—,—dbsize—,—dbcapacity—,—
▶—refresh-window—)
    
```

스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *snapshot-timestamp*

유형 `TIMESTAMP`의 출력 매개변수는 `dbsize` 및 `dbcapacity`가 계산된 시간을 리턴합니다. `refresh-window` 값과 함께 시간소인은 `SYSTOOLS.STMG_DBSIZE_INFO` 테이블에 있는 캐시 값을 언제 새로 고침해야 하는지 판별하는 데 사용됩니다.

### *dbsize*

유형 `BIGINT`의 출력 매개변수는 데이터베이스의 크기를 바이트로 리턴합니다. 데이터베이스 크기는 다음과 같이 계산됩니다.  $dbsize = \text{sum}(\text{used\_pages} * \text{page\_size})$  for each table space (SMS & DMS).

### *dbcapacity*

유형 `BIGINT`의 출력 매개변수는 데이터베이스의 용량을 바이트로 리턴합니다. 이 값은 파티션된 데이터베이스 시스템에서는 사용 불가능합니다. 데이터베이스 용량은 다음과 같이 계산합니다.  $dbcapacity = \text{SUM}(\text{DMS usable\_pages} * \text{page size}) + \text{SUM}(\text{SMS container size} + \text{file system free size per container})$ . 복수의 SMS 컨테이너를 동일한 파일 시스템에 정의한 경우, 파일 시스템의 사용 가능 크기가 용량 계산에 한 번만 포함됩니다.

### *refresh-window*

유형 `INTEGER`의 입력 인수는 데이터베이스 크기 및 용량에 대해 캐시된 값이 새로 고쳐질 때까지의 시간(분)을 지정합니다. 30분의 디폴트 새로 고침 창에 -1을 지정하십시오. 0의 새로 고침 창은 즉시 캐시 값을 강제로 새로 고칩니다.

## 권한 부여

- SYSMON 권한
- GET\_DBSIZE\_INFO 프로시저에 대한 EXECUTE 특권

## 예:

예 1: 30분의 디폴트 새로 고침 창을 사용하여 데이터베이스 크기 및 용량을 가져옵니다. 캐시된 데이터가 30분보다 오래 걸리면 데이터베이스 크기 및 용량이 다시 계산됩니다.

```
CALL GET_DBSIZE_INFO(?, ?, ?, -1)
```

프로시저는 다음을 리턴합니다.

출력 매개변수 값

```
-----  
매개변수 이름   : SNAPSHOTTIMESTAMP  
매개변수 값    : 2004-02-29-18.31.55.178000
```

```
매개변수 이름   : DATABASESIZE  
매개변수 값    : 22302720
```

매개변수 이름 : DATABASECAPACITY  
매개변수 값 : 4684793856

리턴 상태 = 0

예 2: 0분의 새로 고침 창을 사용하여 데이터베이스 크기 및 용량을 가져옵니다. 데이터베이스 크기 및 용량이 즉시 다시 계산됩니다.

```
CALL GET_DBSIZE_INFO(?, ?, ?, 0)
```

프로시저는 다음을 리턴합니다.

출력 매개변수 값

-----  
매개변수 이름 : SNAPSHOTTIMESTAMP  
매개변수 값 : 2004-02-29-18.33.34.561000

매개변수 이름 : DATABASESIZE  
매개변수 값 : 22302720

매개변수 이름 : DATABASECAPACITY  
매개변수 값 : 4684859392

리턴 상태 = 0

예 3: 24시간의 새로 고침 창을 사용하여 데이터베이스 크기 및 용량을 가져옵니다. 캐시된 데이터가 1440분보다 오래 걸리면 데이터베이스 크기 및 용량이 다시 계산됩니다.

```
CALL GET_DBSIZE_INFO(?, ?, ?, 1440)
```

프로시저는 다음을 리턴합니다.

출력 매개변수 값

-----  
매개변수 이름 : SNAPSHOTTIMESTAMP  
매개변수 값 : 2004-02-29-18.33.34.561000

매개변수 이름 : DATABASESIZE  
매개변수 값 : 22302720

매개변수 이름 : DATABASECAPACITY  
매개변수 값 : 4684859392

리턴 상태 = 0

## 사용 시 참고사항

계산된 값은 프로시저 출력 매개변수로서 리턴되며 SYSTOOLS.STMG\_DBSIZE\_INFO 테이블에 캐시됩니다. 계산은 손실적이므로 프로시저는 해당 값을 캐시합니다. 프로시저를 처음으로 실행하면 SYSTOOLS.STMG\_DBSIZE\_INFO 테이블이 자동으로 작성됩니다. *snapshot-timestamp* 및 *refresh-window* 값이 판별한 대로 SYSTOOLS.STMG\_DBSIZE\_INFO 테이블에 캐시된 값이 있고 현재 충분한 경우 캐

시 값이 리턴됩니다. 캐시 값이 현재 충분하지 않은 경우 새 캐시 값이 계산되고 SYSTOOLS.STMG\_DBSIZE\_INFO 테이블에 삽입되며 리턴되고 *snapshot-timestamp* 값이 갱신됩니다.

전역 테이블 스페이스 스냅샷을 위해 모든 파티션에서 데이터를 리턴하도록 하려면 데이터베이스를 활성화해야 합니다.

SYSTOOLSPACE는 메타데이터를 저장하기 위해 루틴의 조작 테이블로 사용됩니다. 즉 데이터베이스 오브젝트와 조작을 설명하기 위해 데이터가 사용됩니다.

---

## NOTIFICATIONLIST 관리 뷰 - Health 통지에 대한 문의처 목록 검색

NOTIFICATIONLIST 관리 뷰는 인스턴스의 Health에 대해 통지된 문의처 및 문의처 그룹 목록을 리턴합니다.

스키마는 SYSIBMADM입니다.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- NOTIFICATIONLIST 관리 뷰에 대한 SELECT 특권
- NOTIFICATIONLIST 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- HEALTH\_GET\_NOTIFICATION\_LIST 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

### 예 :

Health 경보 통지를 수신할 모든 문의처를 검색하십시오.

```
SELECT * FROM SYSIBMADM.NOTIFICATIONLIST
```

다음은 이 쿼리에 대한 출력의 예입니다.

NAME	TYPE
group3	GROUP
user4	CONTACT
group3	GROUP

3개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 223. NOTIFICATIONLIST 관리 뷰에서 리턴된 정보

컬럼 이름	데이터 유형	설명
NAME	VARCHAR (128)	문의처 이름
TYPE	VARCHAR(7)	문의처 유형 • 'CONTACT' • 'GROUP'

## PD\_GET\_DIAG\_HIST - 제공된 기능에서 레코드 리턴

PD\_GET\_DIAG\_HIST 테이블 함수는 제공된 기능에서 로그 레코드, 이벤트 레코드 및 통지 레코드를 리턴합니다. 레코드 유형, 레코드의 고객 영향 값 및 시작-끝 시간소인을 기반으로 필터하는 옵션도 지원됩니다.

### 구문

```
►►—PD_GET_DIAG_HIST—(—facility—,—rectype—,—impact—,—start_time—,—end_time—)——►►
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### facility

레코드가 리턴되는 기능을 지정하는 VARCHAR(20) 유형의 선택적 입력 인수. 기능은 레코드가 관련된 논리 그룹화입니다. 가능한 값은 다음과 같습니다.

- ALL: 모든 기능에서 레코드를 리턴합니다.
- MAIN: DB2 일반 진단 로그에서 레코드를 리턴합니다. 현재 이것은 db2diag 로그 파일, 관리 통지 로그 및 회전 이벤트 로그를 의미합니다.
- OPTSTATS: 옵티마이저 통계와 관련된 레코드를 리턴합니다.

이 매개변수가 널(NULL) 또는 비어 있는 문자열('')이면, 'ALL'이 디폴트값입니다.

#### rectype

리턴할 레코드 유형을 지정하는 VARCHAR(30) 유형의 선택적 입력 인수. 'D + E'와 같이 '+'로 구분되는 유형의 조합이 지원됩니다. 가능한 값은 다음과 같습니다.

- 'ALL': 모든 레코드 유형을 리턴합니다.
- 'D': 모든 진단 레코드를 리턴합니다.
- 'E': 모든 이벤트 레코드를 리턴합니다.

- 'DI': 내부 진단 레코드. 이것은 진단 상황에서 IBM 지원에 사용되는 변환되지 않은 진단 레코드입니다.
  - 'DX': 외부 진단 레코드. 이것은 사용자에게 유용한 변환된 진단입니다. 이러한 레코드는 통지 레코드입니다.
  - 'EI': 내부 이벤트 레코드. 이것은 진단 상황에서 IBM 지원에 사용되는 이벤트 레코드입니다.
  - 'EX': 외부 이벤트 레코드. 이것은 사용자에게 유용한 진단 레코드입니다.
- 이 매개변수가 널(NULL) 또는 비어 있는 문자열('')이면, 모든 레코드가 리턴됩니다.

#### *impact*

리턴된 레코드의 최소 고객 영향 레벨을 지정하는 VARCHAR(18) 유형의 선택적 입력 인수. 사용할 수 있는 값은 다음과 같습니다.

- 'NONE'
- 'UNLIKELY'
- 'POTENTIAL'
- 'IMMEDIATE'
- 'CRITICAL'

이 매개변수가 널(NULL) 또는 비어 있는 문자열('')이면, 모든 레코드가 리턴됩니다.

#### *start\_time*

유효한 시간소인을 지정하는 TIMESTAMP 유형의 선택적 입력 인수. 시간소인이 이 값보다 더 최근이면 항목이 리턴됩니다. 이 매개변수가 널(NULL)이거나 비어 있는 문자열('')이면 얼마나 이전 것인지에 관계없이 레코드가 리턴됩니다.

#### *end\_time*

유효한 시간소인을 지정하는 TIMESTAMP 유형의 선택적 입력 인수. 시간소인이 이 값보다 더 나중이면 항목이 리턴됩니다. 이 매개변수가 널(NULL)이거나 비어 있는 문자열('')이면 얼마나 최근 것인지에 관계없이 레코드가 리턴됩니다.

## 권한 부여

PD\_GET\_DIAG\_HIST 테이블 함수에 대한 EXECUTE 특권

예 :

```
SELECT 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) ) ) AS T
WHERE T.PROCESS_NAME = 'db2star2' OR T.PROCESS_NAME = 'db2stop2'
```

이 쿼리의 출력 예는 다음과 같습니다.



```

FACILITY          RECTYPE  TIMESTAMP          ...
-----
MAIN             EX       2007-06-25-11.34.05.756171 ...
MAIN             EX       2007-06-25-11.34.25.946646 ...

```

2개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```

... IMPACT          MSG
... -----
... -              ADM7514W Database manager has stopped.
... -              ADM7513W Database manager has started.

```

## 리턴되는 정보

표 224. PD\_GET\_DIAG\_HIST 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명
FACILITY	VARCHAR(20)	가능한 레코드가 관련된 논리 그룹화입니다. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• ALL: 모든 기능에서 레코드를 리턴합니다.</li> <li>• MAIN: DB2 일반 진단 로그에서 레코드를 리턴합니다. 현재 이것은 db2diag 로그 파일, 관리 통지 로그 및 회전 이벤트 로그를 의미합니다.</li> <li>• OPTSTATS: 옵티마이저 통계와 관련된 레코드를 리턴합니다.</li> </ul>
RECTYPE	VARCHAR(3)	레코드의 유형. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• 'DI': 내부 진단 레코드</li> <li>• 'DX': 외부 진단 레코드</li> <li>• 'EI': 내부 이벤트 레코드</li> <li>• 'EX': 외부 이벤트 레코드</li> </ul>
TIMESTAMP	TIMESTAMP	메시지가 작성된 시간
TIMEZONE	INTEGER	협정 세계시(UCT)의 시간 차이(분). 예를 들어, -300은 EST입니다.
INSTANCENAME	VARCHAR(128)	메시지를 작성한 인스턴스의 이름
DBPARTITIONNUM	SMALLINT	메시지가 작성된 파티션 번호. 파티션되지 않은 데이터베이스의 경우, 0이 리턴됩니다.
LEVEL	CHAR(1)	레코드의 심각도 레벨. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• 'C': 중요</li> <li>• 'E': 오류</li> <li>• 'I': 정보용</li> <li>• 'S': 위험</li> <li>• 'W': 경고</li> </ul>
IMPACT	VARCHAR(18)	사용자의 관점에서 이 메시지의 영향을 규정합니다. 이것은 DB2가 일부인 비즈니스 프로세스에 대한 메시지의 영향을 명백하게 합니다. 가능한 값은 다음과 같습니다. <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)	이 메시지가 작성되는 동안 액세스 중인 데이터베이스의 이름
EDU_ID	BIGINT	이 메시지를 작성한 EDU(Engine Dispatched Unit) ID
EDUNAME	VARCHAR(64)	이 메시지를 작성한 EDU(Engine Dispatched Unit)의 이름
PID	BIGINT	이 메시지를 작성한 운영 체제 프로세스 ID
PROCESS_NAME	VARCHAR(255)	이 메시지를 작성한 운영 체제 프로세스 이름
TID	BIGINT	이 메시지를 작성한 스레드 숫자 ID
APPLNAME	VARCHAR(255)	사용 가능한 경우 연결을 시작한 클라이언트 응용프로그램의 이름
APPL_ID	VARCHAR(64)	연결을 시작한 응용프로그램 ID(사용 가능한 경우). 예: 'G91A3955.F33A.02DD18143340'

표 224. PD\_GET\_DIAG\_HIST 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
APPLHANDLE	VARCHAR(9)	사용 가능한 경우 연결을 시작한 응용프로그램의 전체 시스템 고유 ID. 이 ID는 에이전트 ID의 동의어입니다. ID는 코디네이팅 파티션 번호로 구성되며 '-'로 구분되는 16비트 카운터입니다. 형식은 다음과 같습니다. 'nnn-xxxx'
AUTH_ID	VARCHAR (30)	프로세스의 시스템 권한 부여 ID
PRODUCT	VARCHAR(50)	메시지를 작성한 제품의 이름(예: 'DB2 Common')
COMPONENT	VARCHAR(255)	메시지를 작성한 구성요소의 이름
FUNCTION	VARCHAR(255)	메시지를 생성한 함수의 이름
PROBE	INTEGER	함수에서 메시지가 생성된 위치를 식별하는데 사용되는 프로브 포인트 수
CALLEDPRODUCT	VARCHAR(50)	오류 소스의 제품 이름. 이것은 오류 소스가 메시지가 작성된 위치가 아닌 경우에 사용됩니다.
CALLEDCOMPONENT	VARCHAR(255)	오류 소스의 구성요소 이름. 이것은 오류 소스가 메시지가 작성된 위치가 아닌 경우에 사용됩니다.
CALLEDFUNCTION	VARCHAR(255)	오류 소스의 함수 이름. 이것은 오류 소스가 메시지가 작성된 위치가 아닌 경우에 사용됩니다.
OSERR	INTEGER	운영 체제 오류 번호
RETCODE	INTEGER	제품별 리턴 코드
MSGNUM	INTEGER	메시지와 연관된 숫자 메시지 번호(사용 가능한 경우). 예를 들어, 이 번호는 ADM7513W의 숫자 부분입니다.
MSGTYPE	CHAR(3)	메시지 ID와 관련된 유형(사용 가능한 경우). 예를 들어, ADM은 관리 통지 로그 메시지에 사용됩니다.
MSG	CLOB(16KB)	이 레코드에 대한 간단한 설명 텍스트. 이것은 MSGNUM에 해당하는 변환된 메시지 텍스트이며 변환된 메시지의 MSGTYPE입니다. 변환되지 않은 메시지의 경우 이것은 간단한 설명입니다. 예: 'db2stop의 일부로 모든 db2fmp 프로세스를 가져오는 중'

표 224. PD\_GET\_DIAG\_HIST 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
OBJTYPE	VARCHAR(64)	이벤트가 적용되는 오브젝트 이벤트의 유형(사용 가능한 경우). 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• 'APM'</li> <li>• 'CATALOG CACHE ENTRY'</li> <li>• 'CFG'</li> <li>• 'CLI'</li> <li>• 'CLP'</li> <li>• 'CONTAINER'</li> <li>• 'COUNTER'</li> <li>• 'DAS'</li> <li>• 'DB2AGENT'</li> <li>• 'DB PART MAP ID'</li> <li>• 'DB PART NUM'</li> <li>• 'DBA'</li> <li>• 'DBM'</li> <li>• 'DMS'</li> <li>• 'DPS'</li> <li>• 'EDU'</li> <li>• 'EVALUATION'</li> <li>• 'EXTENDER'</li> <li>• 'FCM'</li> <li>• 'HISTOGRAM TEMPLATE'</li> <li>• 'INDEX STATS'</li> <li>• 'INITIAL SAMPLING'</li> <li>• 'REDIST DB PART GROUP'</li> <li>• 'REDIST TABLE'</li> <li>• 'RDS'</li> <li>• 'SAMPLING TEST'</li> <li>• 'SERVICE CLASS'</li> <li>• 'STATS'</li> <li>• 'STATS DAEMON'</li> <li>• 'TABLE'</li> <li>• 'TABLE STATS'</li> <li>• 'TABLE AND INDEX STATS'</li> <li>• 'THRESHOLD'</li> <li>• 'UDF'</li> <li>• 'WORK ACTION SET'</li> <li>• 'WORK CLASS SET'</li> <li>• 'WORKLOAD'</li> </ul>
OBJNAME	VARCHAR(255)	이벤트가 관련된 오브젝트의 이름(사용 가능한 경우)
OBJNAME_QUALIFIER	VARCHAR(255)	오브젝트에 대한 추가 정보(사용 가능한 경우)

표 224. PD\_GET\_DIAG\_HIST 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
EVENTTYPE	VARCHAR(24)	이벤트 유형은 이 이벤트와 연관된 조치 또는 명령어입니다. 가능한 값은 다음과 같습니다. <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)	이 이벤트의 키 필드의 간단한 표시

표 224. PD\_GET\_DIAG\_HIST 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
FIRST_EVENTQUALIFIERTYPE	VARCHAR(64)	첫 번째 이벤트 규정자의 유형. 이벤트 규정자를 사용하여 이벤트에 의한 영향에 대해 설명합니다. 가능한 값은 다음과 같습니다. <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> 가 OPTSTATS이면 'AT'가 유일한 값입니다.
FIRST_EVENTQUALIFIER	CLOB(16K)	이벤트의 첫 번째 규정자. <i>facility</i> 가 OPTSTATS이면 통계 콜렉션이 발생한 때를 표시하는 시간소인이 됩니다.
SECOND_EVENTQUALIFIERTYPE	VARCHAR(64)	두 번째 이벤트 규정자의 유형. <i>facility</i> 가 OPTSTATS이면 값은 'BY'입니다.
SECOND_EVENTQUALIFIER	CLOB(16K)	이벤트의 두 번째 규정자. <i>facility</i> 가 OPTSTATS이면 다음과 같은 값이 가능합니다. <ul style="list-style-type: none"> <li>• 비동기</li> <li>• FABRICATE</li> <li>• FABRICATE PARTIAL</li> <li>• SYNCHRONOUS</li> <li>• SYNCHRONOUS SAMPLED</li> <li>• USER</li> </ul>
THIRD_EVENTQUALIFIERTYPE	VARCHAR(64)	세 번째 이벤트 규정자의 유형. <i>facility</i> 가 OPTSTATS이면 값은 '8DUE TO'입니다.
THIRD_EVENTQUALIFIER	CLOB(16K)	이벤트의 세 번째 규정자. <i>facility</i> 가 OPTSTATS이면 다음과 같은 값이 가능합니다. <ul style="list-style-type: none"> <li>• 충돌</li> <li>• 오류</li> <li>• 사용 불가능한 오브젝트</li> <li>• RUNSTATS 오류</li> <li>• 시간소인</li> </ul>
EVENTSTATE	VARCHAR(255)	이벤트 결과로 오브젝트의 상태 또는 조치. 여기에는 이벤트 진행을 표시하는 백분율이 포함됩니다.
EVENTATTRIBUTE	VARCHAR(255)	이벤트 속성. 이것은 이벤트와 연관된 속성 목록입니다. 하나 이상의 속성이 사용되면 '+' 문자로 목록을 구분합니다.(예: 'CACHED + LOGICAL + AUTO') 가능한 값은 다음과 같습니다. <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)	적용 가능한 경우 레코드가 로그된 위치의 논리적 이벤트 스택
CALLSTACK	CLOB(16K)	적용 가능한 경우 이 레코드를 생성한 스레드의 운용 체제 스택 덤프
DUMPPFILE	CLOB(5000)	적용 가능한 경우 로그 레코드와 연관된 보조 덤프 파일 이름. 이것은 메시지와 관련된 추가 정보를 검색할 수 있는 파일 또는 디렉토리의 완전한 경로입니다.
FULLREC	CLOB(16K)	전체 레코드의 형식화된 텍스트 버전. 이 섹션에는 추가 DATA 필드도 포함되어 있습니다.

---

## PDLOGMSGGS\_LAST24HOURS 관리 뷰 및 PD\_GET\_LOG\_MSGS 테이블 함수 - 문제점 판별 메시지 검색

PDLOGMSGGS\_LAST24HOURS 관리 뷰 및 PD\_GET\_LOG\_MSGS 테이블 함수는 DB2 통지 로그에 로그된 문제점 판별 로그 메시지를 리턴합니다. 정보는 데이터베이스 및 시스템 관리자가 사용하도록 제공됩니다.

### PDLOGMSGGS\_LAST24HOURS 관리 뷰

PDLOGMSGGS\_LAST24HOURS 관리 뷰는 최근 24시간 내에 DB2 통지 로그에 기록된 문제점 판별 로그 메시지를 리턴합니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 정보의 완전한 목록은 984 페이지의 표 225의 내용을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- PDLOGMSGGS\_LAST24HOURS 관리 뷰에 대한 SELECT 특권
- PDLOGMSGGS\_LAST24HOURS 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- PD\_GET\_LOG\_MSGS 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한

### 예 :

가장 최근 순으로 순서가 지정된 최근 24시간 내에 기록된 중요한 모든 로그 메시지를 가져오십시오.

```
SELECT * FROM SYSIBMADM.PDLOGMSGGS_LAST24HOURS
      WHERE MSGSEVERITY = 'C' ORDER BY TIMESTAMP DESC
```

이 쿼리의 출력 예는 다음과 같습니다.

TIMESTAMP	TIMEZONE	INSTANCENAME	...
-----	-----	-----	...
2005-11-23-21.56.41.240066	-300	svtdbm4	...
			...
			...
			...
			...
			...
			...

```

2005-11-23-21.56.39.150597      -300 svtdbm4      ...
2005-11-23-21.56.37.363384      -300 svtdbm4      ...
...
...
...
2005-11-23-21.56.35.880314      -300 svtdbm4      ...
...

```

4개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```

... 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 ...
...
...

```

이 쿼리의 출력(계속).

```

... 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 ...
...
...

```

이 쿼리의 출력(계속).

```

... FUNCTION      PROBE  MSGNUM      MSGTYPE      ...
... -----
... sqloSleepInstance      38      504 ADM      ...
...
...
...

```

```

...
...
...
... sqlMarkDBad          10          7518 ADM ...
... sqlrr_dump_ffdc      10           1 ADM ...
...
...
...
... sqlrr_dump_ffdc      10           1 ADM ...
...

```

이 쿼리의 출력(계속).

```

... 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 테이블 함수

PD\_GET\_LOG\_MSGS 테이블 함수는 PDLOGMSGs\_LAST24HOURS 관리 뷰와 동일한 정보를 리턴하지만, 사용자가 최근 24시간으로 제한하지 않은 특정 기간을 지정할 수 있습니다.

리턴 가능한 전체 정보 목록은 984 페이지의 표 225의 내용을 참조하십시오.

### 구문

```

▶▶—PD_GET_LOG_MSGS—(—oldest_timestamp—)—————▶▶

```

스키마는 SYSPROC입니다.



## 테이블 함수 매개변수

### *oldest\_timestamp*

유효한 시간소인을 지정하는 TIMESTAMP 유형의 입력 인수. 가장 최근의 시간소인으로 시작하여 이 입력 인수에 지정된 시간소인을 가진 로그 항목으로 끝나는 항목이 리턴됩니다. 널(NULL) 값이 지정되면 모든 로그 항목이 리턴됩니다.

## 권한 부여

PD\_GET\_LOG\_MSGS 테이블 함수에 대한 EXECUTE 특권

### 예:

예 1: 모든 데이터베이스 파티션에 대해 마지막 주의 DB2 인스턴스에서 데이터베이스 SAMPLE에 로그된 모든 통지 메시지를 검색합니다. 메시지를 시간 순으로 보고하십시오.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...
...			
...			
...			

이 쿼리의 출력(계속).

```
... 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" tolock 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.

```

예 2: 가장 최근 순으로 정렬된 마지막 날의 데이터베이스 파티션 0에 대한 DB2 인스턴스에 기록된 모든 중요한 오류를 검색합니다.

```

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

```

이 쿼리의 출력 예는 다음과 같습니다.

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.

2개의 레코드가 선택되었습니다.

예 3: 마지막 날에 응용프로그램 ID가 \*LOCAL.DB2.050927195337인 응용프로그램을 서비스하는 DB2 프로세스에서 작성된 메시지를 검색합니다.

```

SELECT TIMESTAMP, MSG
      FROM TABLE (PD_GET_LOG_MSGS(CURRENT_TIMESTAMP - 1 DAYS)) AS T
      WHERE APPL_ID = '*LOCAL.DB2.050927195337'

```

이 쿼리의 출력 예는 다음과 같습니다.

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.

...

예 4: 통지 로그에서 메시지 ADM0504C의 모든 인스턴스를 찾으십시오. 고려된 메시지가 시간소인으로 제한되지 않는다는 점에 유의하십시오. 이는 통지 로그 파일이 매우 클 경우 비용이 많이 드는 작업일 수 있습니다.

```
SELECT TIMESTAMP, DBPARTITIONNUM, DBNAME, MSG
FROM TABLE (PD_GET_LOG_MSGS(CAST(NULL AS TIMESTAMP))) AS T
WHERE MSGNUM = 504 AND MSGTYPE = 'ADM' AND MSGSEVERITY = 'C'
```

이 쿼리의 출력 예는 다음과 같습니다.

TIMESTAMP	DBPARTITIONNUM	DBNAME	...
2005-11-23-21.56.41.240066	0	CAPTAIN	...
...			
...			
...			
...			
...			
...			
...			
...			
...			

이 쿼리의 출력(계속).

...	APPL_ID	MSG
...	9.26.15.148.36942.051124025612	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.
...		
...		
...		
...		
...		
...		
...		
...		
...		
...		
...		
...		
...		
...		
...		
...		
...		

### 리턴되는 정보

참고: 파티션된 데이터베이스 환경에서, 로그 메시지가 리턴되는 순서는 보증할 수 없습니다. 로그 레코드 순서가 중요한 경우, 결과는 시간소인별로 정렬되어야 합니다.

표 225. PDLOGMSGs\_LAST24HOURS 관리 뷰 및 PD\_GET\_LOG\_MSGS 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명
TIMESTAMP	TIMESTAMP	항목이 로그된 시간
TIMEZONE	INTEGER	협정 세계시(UCT)의 시간 차이 (분). 예를 들어, -300은 EST입니다.
INSTANCENAME	VARCHAR (128)	메시지를 생성한 인스턴스 이름
DBPARTITIONNUM	SMALLINT	메시지를 생성한 데이터베이스 파티션. 파티션되지 않은 데이터베이스 환경의 경우, 0이 리턴됩니다.
DBNAME	VARCHAR (128)	오류 및 이벤트가 발생한 데이터베이스.
PID	BIGINT	메시지를 생성한 프로세스의 ID
PROCESSNAME	VARCHAR(255)	메시지를 생성한 프로세스 이름
TID	BIGINT	메시지를 생성한 프로세스 내의 스레드 ID
APPL_ID	VARCHAR(64)	프로세스가 작업 중인 응용프로그램 ID
COMPONENT	VARCHAR(255)	메시지를 제공하는 DB2 구성요소 이름. db2AdminMsgWrite API를 사용 중인 사용자 응용프로그램이 기록한 메시지의 경우, "사용자 응용프로그램"이 리턴됩니다.
FUNCTION	VARCHAR(255)	메시지를 제공하는 DB2 함수 이름. db2AdminMsgWrite API를 사용 중인 사용자 응용프로그램이 기록한 메시지의 경우, "사용자 함수"가 리턴됩니다.
PROBE	INTEGER	DB2 고객 지원 및 개발에서 메시지를 생성한 DB2 소스 코드의 위치를 찾을 수 있는 고유 내부 ID.
MSGNUM	INTEGER	오류 또는 이벤트에 대한 숫자 메시지 번호
MSGTYPE	CHAR(3)	메시지 유형을 나타냄: 메시지 유형을 판별할 수 없는 경우, ADM(관리 통지 로그에 기록된 메시지) 또는 널(NULL)을 나타냅니다.
MSGSEVERITY	CHAR(1)	메시지 심각도: C(심각), E(오류), W(경고), I(정보용) 또는 널(메시지 심각도를 판별할 수 없는 경우)
MSG	CLOB(16 KB)	통지 로그 메시지 텍스트

---

## REORGCHK\_IX\_STATS 프로시저 – 재구성 평가에 대한 인덱스 통계 검색

REORGCHK\_IX\_STATS 프로시저는 재구성의 필요 여부를 표시하는 인덱스 통계가 있는 결과 세트를 리턴합니다.

### 구문

►►—REORGCHK\_IX\_STATS—(—*scope*—,—*criteria*—)—————►►

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### *scope*

평가될 테이블의 범위를 지정하는 CHAR(1) 유형의 입력 인수이며, 다음 값 중 하나를 사용합니다.

‘T’

테이블

‘S’

스키마

#### *criteria*

VARCHAR(259) 유형의 입력 인수입니다. *scope*에 ‘T’ 값이 있으면, 완전한 테이블 이름을 지정하거나, ALL, USER 또는 SYSTEM의 값 중 하나를 승인합니다. 범위에 ‘S’ 값이 있으면, 스키마 이름을 지정합니다.

### 권한 부여

- 카탈로그 테이블의 SELECT 특권.
- REORGCHK\_IX\_STATS 프로시저의 EXECUTE 특권.

### 예 :

```
CALL SYSPROC.REORGCHK_IX_STATS('T','JESCOTT.EMPLOYEE')
```

### 사용법 참고

프로시저는 SYSTOOLSTMPSPACE 테이블 스페이스를 사용합니다. SYSTOOLSTMPSPACE가 아직 존재하지 않으면, 프로시저가 이 테이블 스페이스를 작성합니다.

## 리턴된 정보

표 226. REORGCHK\_IX\_STATS 프로시저가 리턴한 정보

컬럼 이름	데이터 유형	설명
TABLE_SCHEMA	VARCHAR (128)	스키마 이름.
TABLE_NAME	VARCHAR (128)	테이블 이름.
DATAPARTITIONNAME	VARCHAR (128)	데이터 파티션 이름 파티션되지 않은 테이블에 대해서는 NULL.
INDEX_SCHEMA	VARCHAR (128)	인덱스 스키마 이름.
INDEX_NAME	VARCHAR (128)	인덱스 이름.
INDCARD	BIGINT	인덱스에 있는 인덱스 항목 수. 일부 인덱스의 경우 테이블 카디널리티 (cardinality)와 다를 수 있습니다. 예를 들어, XML 컬럼의 인덱스 카디널리티 (cardinality)는 테이블 카디널리티 (cardinality)보다 클 수 있습니다.
NLEAF	BIGINT	인덱스 단말 페이지의 총 수.
NUM_EMPTY_LEAFS	BIGINT	pseudo-empty 인덱스 단말 페이지의 수.
NLEVELS	INTEGER	인덱스 레벨의 수.
NUMRIDS_DELETED	BIGINT	pseudo-deleted RID의 수.
FULLKEYCARD	BIGINT	삭제 대상으로 표시되지 않은 고유 인덱스 항목 수.
LEAF_RECSize	BIGINT	리프 페이지에서 인덱스 항목의 레코드 크기. 오버헤드를 제외한 인덱스 항목의 평균 크기로, 인덱스에 참여하는 모든 컬럼의 평균 컬럼 길이에서 계산됩니다.
NONLEAF_RECSize	BIGINT	리프가 아닌 페이지에서 인덱스 항목의 레코드 크기. 오버헤드를 제외한 인덱스 항목의 평균 크기로, INCLUDE 컬럼을 제외하고 인덱스에 참여하는 모든 컬럼의 평균 컬럼 길이에서 계산됩니다.
LEAF_PAGE_OVERHEAD	BIGINT	내부 사용을 목적으로 인덱스 리프 페이지에서 예약된 스페이스
NONLEAF_PAGE_OVERHEAD	BIGINT	내부 사용을 목적으로 인덱스 리프가 아닌 페이지에서 예약된 스페이스
PCT_PAGES_SAVED	SMALLINT	인덱스 압축을 사용하여 저장된 페이지의 퍼센트. 인덱스가 압축됨을 표시하는 0이 아닌 수.
F4	INTEGER	F4 공식 값.
F5	INTEGER	F5 공식 값.
F6	INTEGER	F6 공식 값.
F7	INTEGER	F7 공식 값.

표 226. REORGCHK\_IX\_STATS 프로시저가 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명
F8	INTEGER	F8 공식 값.
REORG	CHAR(5)	5문자 필드의 각 문자를 다섯 가지의 공식(F4, F5, F6, F7 및 F8) 중 하나로 맵핑합니다. 대시는 공식 값이 권장되는 범위 내에 있음을 의미하고, 별표는 공식 값이 권장된 범위 밖에 있음을 의미하며, 이때 권장된 범위는 재구성의 필요 여부를 표시합니다.

## REORGCHK\_TB\_STATS 프로시저 - 재구성 평가에 대한 테이블 통계 검색

REORGCHK\_TB\_STATS 프로시저는 재구성이 필요한지 여부를 표시하는 테이블 통계가 있는 결과 세트를 리턴합니다.

### 구문

```
►►—REORGCHK_TB_STATS—(—scope—, —criteria—)—————►►
```

스키마는 SYSPROC입니다.

### 프로시저 매개변수

#### scope

평가될 테이블의 범위를 지정하는 CHAR(1) 유형의 입력 인수이며, 다음 값 중 하나를 사용합니다.

‘T’

테이블

‘S’

스키마

#### criteria

VARCHAR(259) 유형의 입력 인수입니다. scope에 ‘T’ 값이 있으면, 완전한 테이블 이름을 지정하거나, ALL, USER 또는 SYSTEM의 값 중 하나를 승인합니다. 범위에 ‘S’ 값이 있으면, 스키마 이름을 지정합니다.

### 권한 부여

- 카탈로그 테이블의 SELECT 특권.
- REORGCHK\_TB\_STATS 프로시저에 대한 EXECUTE 특권

### 예 :

```
CALL SYSPROC.REORGCHK_TB_STATS('T', 'JESCOTT.EMPLOYEE')
```

## 사용법 참고

프로시저는 SYSTOOLSTMPSPACE 테이블 스페이스를 사용합니다. SYSTOOLSTMPSPACE가 아직 존재하지 않으면, 프로시저가 이 테이블 스페이스를 작성합니다.

## 리턴되는 정보

표 227. REORGCHK\_TB\_STATS 프로시저에서 리턴한 정보

컬럼 이름	데이터 유형	설명
TABLE_SCHEMA	VARCHAR (128)	스키마 이름.
TABLE_NAME	VARCHAR (128)	테이블 이름.
DATAPARTITIONNAME	VARCHAR (128)	데이터 파티션 이름 파티션되지 않은 테이블에 대해서는 NULL.
CARD	BIGINT	카디널리티(테이블의 행 수)
OVERFLOW	BIGINT	오버플로우 행 수
NPAGES	BIGINT	테이블 행이 존재하는 총 페이지 수. 뷰나 별명의 경우 또는 통계가 수집되지 않은 경우 -1이며, 서브테이블이나 계층 구조 테이블의 경우 -2입니다.
FPAGES	BIGINT	총 페이지 수. 뷰나 별명의 경우 또는 통계가 수집되지 않은 경우 -1이며, 서브테이블이나 계층 구조 테이블의 경우 -2입니다.
ACTIVE_BLOCKS	BIGINT	다차원적으로 클러스터된(MDC) 테이블에 대한 활성 블록의 총 수. 이 필드는 ORGANIZE BY 절을 사용하여 정의된 테이블에만 적용 가능합니다. 데이터를 포함하는 테이블의 블록 수를 표시합니다.
TSIZE	BIGINT	테이블 크기
F1	INTEGER	F1 공식 값
F2	INTEGER	F2 공식 값
F3	INTEGER	F3 공식 값
REORG	CHAR(3)	3 문자 필드, 각 문자는 세 개의 공식 F1, F2 및 F3 중 하나로 맵핑됨. 대시는 공식 값이 권장 범위에 있음을 의미하고, 별표는 공식 값이 권장 범위 밖에 있음을 의미하며 재구성이 필요함을 나타냄



## SQLERRM 스칼라 함수 - 오류 메시지 정보 검색

SQLERRM 스칼라 함수에는 두 가지 버전이 있습니다. 첫 번째 버전을 사용하면 메시지 토큰 및 언어 선택 사용을 포함한 메시지 검색에 대한 완벽한 유연성이 제공됩니다. 두 번째는 입력 매개변수로 SQLCODE만 가져와 영어로 된 간단한 메시지를 리턴합니다.

### SQLERRM 스칼라 함수

이 SQLERRM 스칼라 함수는 메시지 ID, 로케일 및 토큰 입력을 가져와서 지정된 로케일로 VARCHAR(32672) 유형의 간단한 또는 자세한 메시지를 리턴합니다. 입력 로케일이 서버에서 지원되지 않을 경우, 메시지는 영어로 리턴됩니다.

### 구문

```
▶▶ SQLERRM(—msgid—, —tokens—, —token_delimiter—, —locale—, —shortmsg—▶▶  
▶-)▶▶
```

스키마는 SYSPROC입니다.

### 스칼라 함수 매개변수

#### *msgid*

정보를 검색해야 하는 메시지 번호를 나타내는 VARCHAR(9) 유형의 입력 인수. 메시지 번호는 접두부가 'SQL', 'DBA' 또는 'CLI'인 응용프로그램 리턴 코드입니다. 예를 들어, 'SQL551', 'CLI0001'이 있습니다. 메시지 번호는 SQLSTATE가 될 수도 있습니다(예: '42829').

#### *tokens*

오류 메시지 토큰 목록을 나타내는 VARCHAR(70) 유형의 입력 인수. 일부 메시지에는 토큰이 없을 수도 있습니다. 이 매개변수가 널(NULL)이면, 리턴된 메시지에서는 토큰이 대체되지 않습니다. 간단한 디폴트 메시지를 리턴할 경우에만 토큰 대체가 발생합니다. 자세한 메시지 옵션이 선택되면 토큰 대체는 발생하지 않습니다.

#### *token\_delimiter*

토큰 분리문자를 나타내는 VARCHAR(1) 유형의 입력 인수. 이 분리문자는 고유해야 하며 스칼라 함수로 전달된 토큰에 포함되어서는 안 됩니다. 제공된 분리문자가 없으면, 사용된 디폴트 분리문자는 세미콜론입니다.

#### *locale*

해당 언어로 오류 메시지를 검색하기 위해 서버로 전달할 로케일을 나타내는

VARCHAR(33) 유형의 입력 인수. 지정된 로케일이 없거나 서버에서 로케일을 지원하지 않으면, 메시지는 영어로 리턴되며 경고가 리턴됩니다.

#### *shortmsg*

간단한 디폴트 메시지 대신에 자세한 메시지가 리턴되어야 하는지를 나타내는 데 사용되는 INTEGER 유형의 입력 인수. 자세한 메시지를 리턴하려면, 이 값은 0 또는 CAST(INTEGER로서 NULL)로 설정되어야 합니다.

## 권한 부여

SQLERRM 스칼라 함수에 대한 EXECUTE 특권

### 예:

예 1: 토큰이 "AYYANG", "UPDATE" 및 "SYSCAT.TABLES"인 SQL0551N에 대한 간단한 영어 메시지를 검색하십시오.

```
VALUES (SYSPROC.SQLERRM
        ('SQL551', 'AYYANG;UPDATE;SYSCAT.TABLES', ';', 'en_US', 1))
```

다음은 리턴된 출력 예입니다.

```
1
-----
SQL0551N "AYYANG" does not have the privilege to perform operation
"UPDATE" on object "SYSCAT.TABLES"
```

예 2: SQLSTATE 42501과 연관된 영어 오류 메시지를 검색하십시오.

```
VALUES (SYSPROC.SQLERRM ('42501', '', '', 'en_US', 1))
```

다음은 리턴된 출력 예입니다.

```
1
-----
SQLSTATE 42501: The authorization ID does not have the privilege to
perform the specified operation on the identified object.
```

예 3: SQL1001N에 대한 자세한 영어 오류 메시지를 검색하십시오.

```
VALUES (SYSPROC.SQLERRM ('SQL1001', '', '', 'en_US', 0))
```

다음은 리턴된 출력 예입니다.

```
1
-----
SQL1001N "<name>"은(는) 유효한 데이터베이스 이름이 아닙니다.
```

설명:

명령에 지정된 데이터베이스 이름의 구문이 유효하지 않습니다. 데이터베이스 이름은 1-8자로 구성되며, 모든 문자는 데이터베이스 관리 프로그램의 기본 문자 세트에 있는 것이어야 합니다.

명령을 처리할 수 없습니다.

User Response:

올바른 데이터베이스 이름을 지정하여 명령을 다시 제출하십시오.

sqlcode : -1001

sqlstate : 2E000

## SQLERRM 스칼라 함수

이 SQLERRM 스칼라 함수는 입력만으로 SQLCODE를 가져와 영어로 지정된 SQLCODE에 대한 VARCHAR(32672) 유형의 간단한 메시지를 리턴합니다.

### 구문

▶▶—SQLERRM—(—sqlcode—)—————▶▶

스키마는 SYSPROC입니다.

### 스칼라 함수 매개변수

*sqlcode*

SQLCODE를 나타내는 INTEGER 유형의 입력 인수입니다.

### 권한 부여

SQLERRM 스칼라 함수에 대한 EXECUTE 특권

예 :

SQLCODE SQL0551N에 대한 간단한 메시지를 검색하십시오.

VALUES (SYSPROC.SQLERRM (551))

다음은 리턴된 출력 예입니다.

1

-----  
SQL0551N "" does not have the privilege to perform operation  
"" on object "".

---

## SYSPROC.SQLERRM

SYSPROC.SQLERRM 프로시저는 특정 도구를 필요로 하는 데이터베이스 오브젝트를 작성하거나 삭제합니다.

## 구문

```
►►—SYSINSTALLOBJECTS—(—tool-name—,—action—,—tablespace-name—,——————►  
►—schema-name—)—————►
```

스키마는 SYSPROC입니다.

## 프로시저 매개변수

### *tool-name*

다음 값 중 하나를 사용하여 로드될 도구의 이름을 지정하는 VARCHAR(128) 유형의 입력 인수.

- 활동 모니터 오브젝트 작성을 위한 'AM'
- 자율 컴퓨팅을 위한 'DB2AC'(Health 모니터)
- 스토리지 관리를 위한 'STMG\_DBSIZE\_INFO'
- 최적화 프로파일 테이블 작성을 위한 'OPT\_PROFILES'
- 규정을 위한 'POLICY'(테이블과 트리거)
- Explain 테이블 작성을 위한 'EXPLAIN'

### *action*

취해질 조치를 지정하는 CHAR(1) 유형의 입력 인수. 올바른 값은 다음과 같습니다.

- C*     오브젝트를 작성합니다.
- D*     오브젝트를 삭제합니다.
- V*     오브젝트를 검증합니다.

### *tablespace-name*

작성될 오브젝트의 테이블 스페이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. 값이 지정되지 않았거나 값이 비어있거나 빈 문자열이면 도구 이름이 AM 인 경우 기본 사용자 스페이스가 사용됩니다. 그렇지 않으면 SYSTOOLSPACE 테이블 스페이스가 사용됩니다. SYSTOOLSPACE가 이미 존재하지 않으면 작성됩니다.

### *schema-name*

'EXPLAIN' 도구 이름 옵션을 제외하고 SYSTOOLS는 입력 매개변수로 전달된 스키마 이름과 상관 없는 스키마로 항상 사용됩니다.

'EXPLAIN' 도구 이름 옵션의 경우 입력 스키마 이름이 전달될 수 있으며 테이블은 지정된 스키마 이름으로 생성됩니다. 스키마 이름이 입력 매개변수로 전달되지 않으면 SYSTOOLS 스키마가 사용됩니다.

예 :

```
CALL SYSPROC.SYSINSTALLOBJECTS('AM', 'C', CAST (NULL AS VARCHAR(128)),  
CAST (NULL AS VARCHAR(128)))
```



## 제 21 장 사용되지 않는 SQL 관리 루틴 및 대체 루틴이나 뷰

기존 관리 루틴에 대해 Linux, UNIX 및 Windows용 DB2 버전 9.7에 확장된 지원을 제공하기 위해 일부 DB2 버전 9.5 루틴이 새롭고 포괄적인 루틴 또는 뷰로 교체되었습니다.

Linux, UNIX 및 Windows용 DB2 버전 9.7 테이블 함수를 사용하는 응용프로그램은 새 함수 또는 관리 뷰를 사용하도록 수정해야 합니다. 새 테이블 함수에는 원래 함수와 동일한 기본 이름이 있지만 추가된 제품의 버전에 대해 ‘\_Vxx’가 접미부로 추가됩니다(예: \_V97). 대부분의 경우 새 테이블 함수와 관리 뷰는 추가 정보를 리턴합니다. 관리 뷰는 항상 테이블 함수의 가장 현재 버전을 기본으로 하므로 추가 응용프로그램 이식성을 허용합니다. 컬럼이 추후 릴리스로 넘어갈 때마다 다룰 수 있으므로(즉, 일부가 추가되고 일부가 삭제됨), 관리 뷰에서 특정 컬럼을 선택하거나 결과 세트가 SELECT \* 명령문을 응용프로그램에서 사용하는지 여부를 설명하도록 하는 것이 좋습니다.

표 228. DB2 버전 9.7에서 사용되지 않는 SQL 관리 루틴 및 해당 대체 루틴 또는 뷰

DB2 버전 9.5 사용되지 않는 함수	Linux, UNIX 및 Windows용 새 DB2 버전 9.7 함수 또는 뷰
1004 페이지의 『ADMINTABCOMPRESSINFO 뷰 및 ADMIN_GET_TAB_COMPRESS_INFO』	251 페이지의 『ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN_GET_TAB_COMPRESS_INFO_V97 테이블 함수 - 압축 정보 리턴』
997 페이지의 『ADMIN_GET_TAB_INFO 테이블 함수 - 테이블에 대한 크기 및 상태 정보 검색』	260 페이지의 『ADMINTABINFO 관리 뷰 및 ADMIN_GET_TAB_INFO_V97 테이블 함수 - 테이블 크기 및 상태 정보 검색』
767 페이지의 『SNAPSTORAGE_PATHS 관리 뷰 및 SNAP_GET_STORAGE_PATHS 테이블 함수 - 자동 스토리지 경로 정보 검색』	618 페이지의 『SNAPSTORAGE_PATHS 관리 뷰 및 SNAP_GET_STORAGE_PATHS_V97 테이블 함수 - 자동 스토리지 경로 정보 검색』
794 페이지의 『SNAPTbsp_PART 관리 뷰 및 SNAP_GET_TBSP_PART_V91 테이블 함수 - tablespace_nodeinfo 논리 데이터 그룹 스냅샷 정보 검색』	644 페이지의 『SNAPTbsp_PART 관리 뷰 및 SNAP_GET_TBSP_PART_V97 테이블 함수 - tablespace_nodeinfo 논리 데이터 그룹 스냅샷 정보 검색』
1147 페이지의 『WLM_GET_ACTIVITY_DETAILS - 특정 활동에 대한 세부 정보 리턴』	379 페이지의 『MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기』
1165 페이지의 『WLM_GET_SERVICE_SUBCLASS_STATS - 서비스 서브클래스의 통계 리턴』	897 페이지의 『WLM_GET_SERVICE_SUBCLASS_STATS_V97 테이블 함수 - 서비스 서브클래스의 통계 리턴』
1177 페이지의 『WLM_GET_WORKLOAD_STATS - 워크로드 통계 리턴』	915 페이지의 『WLM_GET_WORKLOAD_STATS_V97 테이블 함수 - 워크로드 통계 리턴』
1172 페이지의 『WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES - 활동 목록 리턴』	909 페이지의 『WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97 - 활동 목록 리턴』

표 228. DB2 버전 9.7에서 사용되지 않는 SQL 관리 루틴 및 해당 대체 루틴 또는 뷰 (계속)

DB2 버전 9.5 사용되지 않는 함수	Linux, UNIX 및 Windows용 새 DB2 버전 9.7 함수 또는 뷰
1161 페이지의 『WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES - 워크로드 어커런스 목록』	893 페이지의 『WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES_V97 - 워크로드 어커런스 나열』
1154 페이지의 『WLM_GET_SERVICE_CLASS_AGENTS - 서비스 클래스에서 실행 중인 에이전트 나열』	885 페이지의 『WLM_GET_SERVICE_CLASS_AGENTS_V97 테이블 함수 - 서비스 클래스에서 실행 중인 에이전트 나열』

Health 모니터는 Linux, UNIX 및 Windows용 DB2 버전 9.7에서는 더 이상 사용되지 않습니다. 버전 9.7에서는 사용되지 않는 Health 모니터 인터페이스가 여전히 지원됩니다. Linux, UNIX 및 Windows용 DB2 데이터 및 데이터 중심 응용프로그램 관리를 위한 새 GUI 도구 스위트가 사용 가능하며 제어 센터 도구 대신 사용할 수 있습니다. 자세한 정보는 데이터베이스 관리 및 응용프로그램 개발 도구를 참조하십시오.

표 229. 사용되지 않는 Health 모니터 루틴

1011 페이지의 『HEALTH_CONT_HI』
1013 페이지의 『HEALTH_CONT_HI_HIS』
1015 페이지의 『HEALTH_CONT_INFO』
1017 페이지의 『HEALTH_DB_HI』
1021 페이지의 『HEALTH_DB_HI_HIS』
1025 페이지의 『HEALTH_DB_HIC』
1027 페이지의 『HEALTH_DB_HIC_HIS』
1030 페이지의 『HEALTH_DB_INFO』
1031 페이지의 『HEALTH_DBM_HI』
1033 페이지의 『HEALTH_DBM_HI_HIS』
1035 페이지의 『HEALTH_DBM_INFO』
1036 페이지의 『HEALTH_GET_ALERT_ACTION_CFG』
1040 페이지의 『HEALTH_GET_ALERT_CFG』
1043 페이지의 『HEALTH_GET_IND_DEFINITION』
1045 페이지의 『HEALTH_HI_REC』
1047 페이지의 『HEALTH_TBS_HI』
1050 페이지의 『HEALTH_TBS_HI_HIS』
1054 페이지의 『HEALTH_TBS_INFO』

이전 릴리스, DB2 버전 9.5에는 DB2 버전 9.1 함수를 대체하는 새 함수가 있습니다.

표 230. DB2 버전 9.5에서 사용되지 않는 SQL 관리 루틴 및 대체 루틴이나 뷰

DB2 버전 9.1 사용되지 않는 함수	새 DB2 버전 9.5 함수 또는 뷰
1056 페이지의 『SNAP_GET_APPL 테이블 함수 - appl 논리 데이터 그룹 스냅샷 정보 검색』	535 페이지의 『SNAPAPPL 관리 뷰 및 SNAP_GET_APPL_V95 테이블 함수 - appl 논리 데이터 그룹 스냅샷 정보 검색』



표 230. DB2 버전 9.5에서 사용되지 않는 SQL 관리 루틴 및 대체 루틴이나 뷰 (계속)

DB2 버전 9.1 사용되지 않는 함수	새 DB2 버전 9.5 함수 또는 뷰
1062 페이지의 『SNAP_GET_APPL_INFO 테이블 함수 - appl_info 논리 데이터 그룹 스냅샷 정보 검색』	526 페이지의 『SNAPAPPL_INFO 관리 뷰 및 SNAP_GET_APPL_INFO_V95 테이블 함수 - Retrieve appl_info 논리 데이터 그룹 스냅샷 정보』
1069 페이지의 『SNAP_GET_BP 테이블 함수 - 버퍼 풀 논리 그룹 스냅샷 정보 검색』	543 페이지의 『SNAPBP 관리 뷰 및 SNAP_GET_BP_V95 테이블 함수 - 버퍼 풀 논리 그룹 스냅샷 정보 검색』
1083 페이지의 『SNAP_GET_DB_V91 테이블 함수 - dbase 논리적 그룹에서 스냅샷 정보 검색』	557 페이지의 『SNAPDB 관리 뷰 및 SNAP_GET_DB_V95 테이블 함수 - dbase 논리 그룹에서 스냅샷 정보 검색』
1080 페이지의 『SNAP_GET_DBM 테이블 함수 - dbm 논리 그룹화 스냅샷 정보 검색』	572 페이지의 『SNAPDBM 관리 뷰 및 SNAP_GET_DBM_V95 테이블 함수 - dbm 논리적 그룹화 스냅샷 정보 검색』
1093 페이지의 『SNAP_GET_DYN_SQL_V91 테이블 함수 - dynsql 논리적 그룹 스냅샷 정보 검색』	584 페이지의 『SNAPDYN_SQL 관리 뷰 및 SNAP_GET_DYN_SQL_V95 테이블 함수 - dynsql 논리 그룹 스냅샷 정보 검색』

## ADMIN\_GET\_TAB\_INFO 테이블 함수 - 테이블에 대한 크기 및 상태 정보 검색

주: 이 테이블 함수가 사용되지 않으며 260 페이지의 『ADMINTABINFO 관리 뷰 및 ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수 - 테이블 크기 및 상태 정보 검색』으로 교체됩니다.

ADMIN\_GET\_TAB\_INFO 테이블 함수는 카탈로그 뷰에서 현재 사용할 수 없는 테이블 크기와 상태 정보를 검색하는 방법을 제공합니다.

리턴될 수 있는 전체 정보 목록은 ADMIN\_GET\_TAB\_INFO 테이블 함수 메타데이터 테이블을 참조하십시오.

### 구문

```
►►—ADMIN_GET_TAB_INFO—(—tabschema—,—tablename—)—————►►
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

*tabschema*

스키마 이름을 지정하는 유형 VARCHAR(128)의 입력 인수.

*tablename*

테이블 이름, 구체화된 쿼리 테이블 이름이나 계층 구조 테이블 이름을 지정하는 VARCHAR(128) 유형의 입력 인수.

### 권한 부여

ADMIN\_GET\_TAB\_INFO 테이블 함수에서의 EXECUTE 특권.

예 :

예 1: 테이블 DBUSER1.EMPLOYEE에 대한 크기와 상태 정보를 검색합니다.

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_INFO('DBUSER1', 'EMPLOYEE'))
AS T
```

예 2: 파티션되지 않은 테이블(DBUSER1.EMPLOYEE)이 단일 테이블 스페이스에 저장된 모든 연관된 오브젝트(예를 들어 인덱스 및 LOB)와 함께 존재한다고 가정합니다. 테이블 스페이스에서 테이블이 실제 사용하고 있는 스페이스 크기를 계산합니다.

```
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
```

새 테이블 스페이스에 동일한 페이지 크기가 있고 Extent 크기가 원래 테이블 스페이스와 같은 다른 테이블 스페이스로 테이블이 이동한 경우 필요한 스페이스 크기를 계산합니다.

```
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
```

## 사용 시 참고사항

- 두 *tabschema* 및 *tablename*이 지정된 경우 정보는 해당 특정 테이블에만 리턴됩니다.
- *tabschema*이 지정되지만 *tablename*이 비어 있거나("") 널(NULL)인 경우, 정보는 주어진 스키마의 모든 테이블에 대해 리턴됩니다.
- *tabschema*가 비어 있거나("") 널(NULL)이고 *tablename*이 지정되면 오류가 리턴됩니다. 특정 테이블에 대한 정보를 검색하려면 테이블이 스키마와 테이블 이름 모두에서 식별되어야 합니다.
- *tabschema* 및 *tablename*이 모두 비어 있거나("") 널(NULL)이면 정보는 모든 테이블에 대해 리턴됩니다.
- *tabschema* 또는 *tablename*이 존재하지 않거나 *tablename*이 테이블 이름(유형 T), 구체화된 쿼리 테이블 이름(유형 S) 또는 계층 구조 테이블 이름(유형 H)에 해당되지 않으면, 비어 있는 결과 세트가 리턴됩니다.
- ADMIN\_GET\_TAB\_INFO 테이블 함수가 주어진 테이블에 대해 데이터를 검색하고 있는 경우, SYSTABLES의 해당 행에 대한 공유 잠금을 획득하게 되어 리턴되는 데이터의 일관성이 유지됩니다(예를 들어 정보가 검색되고 있는 동안 테이블이 삭제되지 않도록 합니다). 테이블 함수 호출 기간 동안이 아니라 테이블에 대한 크기와 상태 정보를 검색하는 동안만 잠금이 보유됩니다.
- SMS 테이블 스페이스의 테이블에 대해 보고된 실제 크기는 논리적 크기와 동일합니다.

- inplace reorg가 테이블에서 사용 중이면 데이터 오브젝트(DATA\_OBJECT\_P\_SIZE)에 대한 실제 크기가 계산되지 않습니다. 논리적 크기만 리턴됩니다. INPLACE\_REORG\_STATUS 출력 컬럼을 보면 inplace reorg가 사용 중인지 알 수 있습니다.
- 오브젝트가 아직 재구성되지 않은 경우 DB2 UDB 버전 8이 실제 크기보다 커지기 전에 LOB 오브젝트에 대해 보고된 논리적 크기가 작성됩니다.

## ADMIN\_GET\_TAB\_INFO 테이블 함수 메타데이터

표 231. ADMIN\_GET\_TAB\_INFO 테이블 함수 메타데이터

컬럼 이름	데이터 유형	설명
TABSCHEMA	VARCHAR (128)	스키마 이름.
TABNAME	VARCHAR (128)	테이블 이름.
TABTYPE	CHAR(1)	테이블 유형 <ul style="list-style-type: none"> <li>• 'H' = 계층 구조 테이블</li> <li>• 'S' = 구체화된 쿼리 테이블</li> <li>• 'T' = 테이블</li> </ul>
DBPARTITIONNUM	SMALLINT	데이터베이스 파티션 번호.
DATA_PARTITION_ID	INTEGER	데이터 파티션 번호.
AVAILABLE	CHAR(1)	테이블 상태 <ul style="list-style-type: none"> <li>• 'N' = 테이블이 사용 불가능합니다. 테이블이 사용 불가능하면 크기와 상태와 관련된 기타 모든 출력 컬럼이 널(NULL)이 됩니다.</li> <li>• 'Y' = 테이블이 사용 가능합니다.</li> </ul> <p>주: 복구 불가능한 로드를 통한 롤 포워드의 경우 테이블을 사용 불가능 상태로 설정합니다.</p>

표 231. ADMIN\_GET\_TAB\_INFO 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
DATA_OBJECT_L_SIZE	BIGINT	데이터 오브젝트 논리적 크기. 테이블에 대해 논리적으로 할당된 디스크 스페이스의 크기로 KB 단위로 보고됩니다. 논리적 크기는 테이블이 알고 있는 스페이스 크기입니다. 테이블에 대해 할당된 실제 스페이스 크기보다 작을 수 있습니다(예를 들어 논리적 테이블 절단의 경우). 다차원적으로 클러스터된 (MDC) 테이블의 경우 이 크기에는 블록 맵 오브젝트의 논리적 크기가 포함됩니다. 리턴된 크기는 DMS 테이블 스페이스에서 작성된 오브젝트와 테이블에 대해 논리적으로 할당된 전체 크기로 대략의 Extent Map 페이지(EMP) extent로 간주됩니다. 이 크기는 기본 테이블의 논리적 크기만을 나타냅니다. LOB 데이터, Long 데이터, 인덱스 및 XML 오브젝트로 사용된 스페이스는 다른 컬럼에서 보고됩니다.
DATA_OBJECT_P_SIZE	BIGINT	데이터 오브젝트 실제 크기. 테이블에 대해 실제 할당된 디스크 스페이스의 크기로 KB 단위로 보고됩니다. MDC 테이블의 경우 이 크기에는 블록 맵 오브젝트의 크기가 포함됩니다. 리턴된 크기는 테이블에 대해 할당된 전체 크기로 간주되며 DMS 테이블 스페이스에서 작성된 오브젝트에 대한 EMP extent를 포함합니다. 이 크기는 기본 테이블의 실제 크기만을 나타냅니다. LOB 데이터, Long 데이터, 인덱스 및 XML 오브젝트로 사용된 스페이스는 다른 컬럼에서 보고됩니다.
INDEX_OBJECT_L_SIZE	BIGINT	인덱스 오브젝트 논리적 크기. 테이블에서 정의된 인덱스에 대해 논리적으로 할당된 디스크 스페이스의 크기로 KB 단위로 보고됩니다. 논리적 크기는 테이블이 알고 있는 스페이스 크기입니다. 테이블의 인덱스 데이터를 보유하도록(예를 들어 논리적 테이블 절단의 경우) 할당된 실제 크기보다 작을 수 있습니다. 리턴된 크기는 DMS 테이블 스페이스에서 작성된 인덱스와 인덱스에 논리적으로 할당된 전체 크기로 대략의 EMP extent로 간주됩니다. 이 값은 파티션되지 않은 테이블에 대해서만 보고됩니다. 파티션된 테이블의 경우 이 값은 0이 됩니다.

표 231. ADMIN\_GET\_TAB\_INFO 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
INDEX_OBJECT_P_SIZE	BIGINT	인덱스 오브젝트 실제 크기. 테이블에서 정의된 인덱스에 대해 할당된 디스크 스페이스의 실제 크기로 KB 단위로 보고됩니다. 리턴된 크기는 인덱스에 대해 할당된 전체 크기로 간주되며 DMS 테이블 스페이스에서 작성된 인덱스에 대한 EMP extent를 포함합니다. 이 값은 파티션되지 않은 테이블에 대해서만 보고됩니다. 파티션된 테이블의 경우 이 값은 영(0)이 됩니다.
LONG_OBJECT_L_SIZE	BIGINT	Long 오브젝트 논리적 크기. 테이블의 long 필드 데이터에 논리적으로 할당된 디스크 스페이스 크기로 KB 단위로 보고됩니다. 논리적 크기는 테이블이 알고 있는 스페이스 크기입니다. 테이블의 long 필드 데이터를 보유하도록(예를 들어 논리적 테이블 절단의 경우) 할당된 실제 스페이스 크기보다 작을 수 있습니다. 리턴된 크기는 DMS 테이블 스페이스에서 작성된 long 필드 데이터와 long 필드 데이터에 논리적으로 할당된 전체 크기로 대략의 EMP extent로 간주됩니다.
LONG_OBJECT_P_SIZE	BIGINT	Long 오브젝트 실제 크기. 테이블의 long 필드 데이터에 실제 할당된 디스크 스페이스 크기로 KB 단위로 보고됩니다. 리턴된 크기는 long 필드 데이터의 전체 크기로 간주되며 DMS 테이블 스페이스에서 작성된 long 필드 데이터에 대한 EMP extent를 포함합니다.
LOB_OBJECT_L_SIZE	BIGINT	LOB 오브젝트 논리적 크기. 테이블의 LOB 데이터에 논리적으로 할당된 디스크 스페이스 크기로 KB 단위로 보고됩니다. 논리적 크기는 테이블이 알고 있는 스페이스 크기입니다. 테이블의 LOB 데이터를 보유하도록(예를 들어 논리적 테이블 절단의 경우) 할당된 실제 스페이스 크기보다 작을 수 있습니다. 이 크기는 LOB 할당 오브젝트에 대해 논리적으로 할당된 스페이스를 포함합니다. 리턴된 크기는 DMS 테이블 스페이스에서 작성된 LOB 데이터와 LOB 데이터에 논리적으로 할당된 전체 크기로 대략의 EMP extent로 간주됩니다.
LOB_OBJECT_P_SIZE	BIGINT	LOB 오브젝트 실제 크기. 테이블의 LOB 데이터에 실제 할당된 디스크 스페이스 크기로 KB 단위로 보고됩니다. 이 크기는 LOB 할당 오브젝트에 대해 할당된 스페이스를 포함합니다. 리턴된 크기는 LOB 데이터에 대해 할당된 전체 크기로 간주되며 DMS 테이블 스페이스에서 작성된 LOB 데이터에 대한 EMP extent를 포함합니다.

표 231. ADMIN\_GET\_TAB\_INFO 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
XML_OBJECT_L_SIZE	BIGINT	XML 오브젝트 논리적 크기. 테이블의 XML 데이터에 논리적으로 할당된 디스크 스페이스 크기로 KB 단위로 보고됩니다. 논리적 크기는 테이블이 알고 있는 스페이스 크기입니다. 테이블의 XML 데이터를 보유 하도록(예를 들어 논리적 테이블 절단의 경우) 할당된 실제 스페이스 크기보다 작을 수 있습니다. 리턴된 크기는 DMS 테이블 스페이스에서 작성된 XML 데이터와 XML 데이터에 논리적으로 할당된 전체 크기로 대략의 EMP extent로 간주됩니다.
XML_OBJECT_P_SIZE	BIGINT	XML 오브젝트 실제 크기. 테이블의 XML 데이터에 실제 할당된 디스크 스페이스 크기로 KB 단위로 보고됩니다. 리턴된 크기는 XML 데이터에 대해 할당된 전체 크기로 간주되며 DMS 테이블 스페이스에서 작성된 XML 데이터에 대한 EMP extent를 포함합니다.
INDEX_TYPE	SMALLINT	테이블에 대해 현재 사용되는 인덱스 유형을 나타냅니다. 다음을 리턴합니다. <ul style="list-style-type: none"> <li>• type-1 인덱스가 사용 중인 경우 1</li> <li>• type-2 인덱스가 사용 중인 경우 2</li> </ul>
REORG_PENDING	CHAR(1)	'Y'는 reorg 권장 변경이 테이블에 적용되었고 클래식(오프라인) reorg가 필요하다는 것을 나타냅니다. 그렇지 않은 경우 'N'이 리턴됩니다.
INPLACE_REORG_STATUS	VARCHAR(10)	테이블에서 inplace 테이블 재구성의 현재 상태. 상태 값은 다음 중 하나가 될 수 있습니다. <ul style="list-style-type: none"> <li>• ABORTED(PAUSED 상태의 경우 RESUME할 수 없으며 STOP이 필요합니다)</li> <li>• EXECUTING</li> <li>• NULL(inplace reorg가 테이블에서 수행되지 않은 경우)</li> <li>• PAUSED</li> </ul>
LOAD_STATUS	VARCHAR(12)	테이블에서 로드 조작의 현재 상태. 상태 값은 다음 중 하나가 될 수 있습니다. <ul style="list-style-type: none"> <li>• IN_PROGRESS</li> <li>• NULL(테이블에서 로드가 진행되지 않았고 테이블이 로드 보류 상태가 아닌 경우)</li> <li>• PENDING</li> </ul>

표 231. ADMIN\_GET\_TAB\_INFO 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
READ_ACCESS_ONLY	CHAR(1)	테이블이 읽기 액세스 전용 상태인 경우 'Y', 그렇지 않은 경우 'N'입니다. 'N' 값은 테이블에 모두 액세스할 수 있음을 의미하는 것으로 해석되어서는 안됩니다. 로드가 진행 중이거나 보류 중인 경우 'Y' 값은 테이블 데이터가 읽기 액세스로 가능하다는 것을 의미하며 'N' 값은 테이블에 액세스할 수 없음을 의미합니다. 마찬가지로 테이블 상태가 무결성 설정 보류 상태인 경우 (SYSCAT.TABLES STATUS 컬럼 참조) 'N' 값은 테이블에 액세스할 수 없음을 의미합니다.
NO_LOAD_RESTART	CHAR(1)	'Y' 값은 테이블이 로드 재시작이 불가능한 부분 로드 상태를 나타냅니다. 그렇지 않은 경우 'N'이 리턴됩니다.
NUM_REORG_REC_ALTERS	SMALLINT	마지막 재조작 이후 이 테이블에 대해 수행된 reorg 권장 변경 조작 수(예를 들어 재구성이 필요한 이후의 변경 조작).
INDEXES_REQUIRE_REBUILD	CHAR(1)	테이블에서 정의된 인덱스를 재빌드해야 하는 경우 'Y', 그렇지 않은 경우 'N'. 테이블에 정의된 인덱스가 없는 경우, 재빌드할 인덱스가 없기 때문에 'N'도 리턴됩니다.
LARGE_RIDS	CHAR(1)	테이블이 큰 행 ID(RID)(4바이트 페이지 번호, 2바이트 슬롯 번호)를 사용하는지 여부를 나타냅니다. 'Y' 값은 테이블이 큰 RID를 사용하고 있음을 나타내며 'N'은 큰 RID를 사용하고 있지 않음을 나타냅니다. 테이블이 큰 RID를 지원하는 경우(즉, 테이블이 큰 테이블 스페이스에 있는 경우)'P'(보류) 값이 리턴되지만 테이블에 대한 적어도 하나의 인덱스가 재구성되어 있지 않거나 아직 재빌드되어 있지 않으므로 테이블은 여전히 4바이트 RID(테이블이나 인덱스를 변환하기 위해서는 해당 조치가 취해져야 함을 의미)를 사용합니다.
LARGE_SLOTS	CHAR(1)	테이블이 큰 슬롯(페이지 당 256행 이상 허용)을 사용하고 있는지 여부를 표시합니다. 'Y' 값은 테이블이 큰 슬롯을 사용하고 있음을 나타내며 'N'은 큰 슬롯을 사용하지 않음을 나타냅니다. 테이블이 큰 슬롯(즉, 테이블이 큰 테이블 스페이스에 있는 경우)을 지원하는 경우 'P'(보류)가 리턴되지만 아직 테이블에서 수행된 테이블 절단 조작이나 오프라인 테이블 재구성이 없으므로 여전히 페이지당 최대 255행을 사용합니다.

표 231. ADMIN\_GET\_TAB\_INFO 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
DICTIONARY_SIZE	BIGINT	행 압축 사전이 테이블에 대해 존재하는 경우 행 압축에 사용되는 사전 크기(바이트로 표시).

## ADMINTABCOMPRESSINFO 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO

주: 이 테이블 함수가 사용되지 않으며 251 페이지의 『ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수 - 압축 정보 리턴』으로 교체됩니다.

ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO 테이블 함수는 구체화된 쿼리 테이블(MQT), 계층 구조 테이블 및 테이블에 대한 압축 정보를 리턴합니다.

### ADMINTABCOMPRESSINFO 관리 뷰

ADMINTABCOMPRESSINFO 관리 뷰는 구체화된 쿼리 테이블(MQT), 계층 구조 테이블 및 테이블에 대한 압축 정보만을 리턴합니다. SYSCAT.TABLES 카탈로그 뷰에서 이들 테이블 유형은 테이블에 대해 T로, 구체화된 쿼리 테이블에 대해서는 S, 계층 구조 테이블에 대해서는 H로 보고됩니다. 테이블에 대한 데이터베이스 파티션 레벨 및 데이터 파티션 레벨 모두로 정보가 리턴됩니다.

스키마는 SYSIBMADM입니다.

리턴될 수 있는 전체 정보 목록은 ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO 테이블 함수 메타데이터 테이블을 참조하십시오.

### 권한 부여

다음 권한 중 하나가 필요합니다.

- ADMINTABCOMPRESSINFO 관리 뷰에 대한 SELECT 특권
- ADMINTABCOMPRESSINFO 관리 뷰에 대한 CONTROL 특권
- DATAACCESS 권한

또는 다음 특권 또는 권한 중 하나도 필요합니다.

- ADMIN\_GET\_TAB\_COMPRESS\_INFO 테이블 함수에 대한 EXECUTE 특권
- DATAACCESS 권한



예:

예 1: 모든 테이블에 대한 모든 압축 정보를 검색합니다.

```
SELECT * FROM SYSIBMADM.ADMINTABCOMPRESSINFO
```

이 쿼리의 출력 예는 다음과 같습니다.

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
...	...	...	...	...	...	...

156개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속):

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
...	...	...	...	...	...

예 2: 모든 테이블에 대해 사전 작성 시간 및 조치를 빌드하는 사전을 판별합니다.

```
SELECT TABSCHEMA, TABNAME, DBPARTITIONNUM, DATA_PARTITION_ID, DICT_BUILDER, DICT_BUILD_TIMESTAMP  
FROM SYSIBMADM.ADMINTABCOMPRESSINFO
```

이 쿼리의 출력 예는 다음과 같습니다.

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	-
...	...	...	...	...	...

156개의 레코드가 선택되었습니다.

## ADMIN\_GET\_TAB\_COMPRESS\_INFO 테이블 함수

ADMIN\_GET\_TAB\_COMPRESS\_INFO 테이블 함수는

ADMINTABCOMPRESSINFO 관리 뷰와 동일한 정보를 리턴하지만 스키마, 테이블 이름 및 실행 모드를 지정할 수 있습니다.

리턴될 수 있는 전체 정보 목록은 ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO 테이블 함수 메타데이터 테이블을 참조하십시오.

주: 이 테이블 함수가 사용되지 않으며 251 페이지의 『ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수 - 압축 정보 리턴』으로 교체됩니다.

## 구문

```
▶▶—ADMIN_GET_TAB_COMPRESS_INFO—(—tabschema—,—tabname—,—execmode—)————▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *tabschema*

스키마 이름을 지정하는 VARCHAR  
(128) 유형의 입력 인수.

### *tabname*

테이블 이름, 구체화된 쿼리 테이블 이름이나 계층 구조 테이블 이름을 지정하는 유형 VARCHAR  
(128)의 입력 인수.

### *execmode*

실행 모드를 지정하는 VARCHAR(30) 유형의 입력 인수. 실행 모드는 다음 중 하나일 수 있습니다.

- ‘REPORT’ -- 마지막 생성 날짜로 압축 정보를 보고합니다. 이것은 디폴트값입니다.
- ‘ESTIMATE’ -- 현재 테이블을 기반으로 하는 새 압축 정보를 생성합니다.

## 권한 부여

ADMIN\_GET\_TAB\_COMPRESS\_INFO 함수에서의 EXECUTE 특권.

## 예:

예 1: 테이블 SIMAP2.STAFF에 대한 기존 압축 정보를 검색합니다.

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO('SIMAP2', 'STAFF', 'REPORT'))
AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

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

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속):

COMPRESS_DICT_SIZE	EXPAND_DICT_SIZE	ROWS_SAMPLED	PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH
13312	5312	35	65	84	100

예 2: 테이블 SIMAP2.STAFF에 대한 예상 압축 정보를 지금 날짜로 해서 검색합니다.

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO('SIMAP2', 'STAFF', 'ESTIMATE'))
AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

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

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속):

COMPRESS_DICT_SIZE	EXPAND_DICT_SIZE	ROWS_SAMPLED	PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH
13508	6314	68	72	89	98

예 3: 스키마 SIMAP2의 모든 테이블에 대한 전체 디렉토리 크기를 판별합니다.

```
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
```

이 쿼리의 출력:

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

28개의 레코드가 선택되었습니다.

예 4: SIMAP2 스키마의 테이블 사전 정보의 보고서를 봅니다.

```
SELECT * FROM TABLE (SYSPROC.ADMIN_GET_TAB_COMPRESS_INFO('SIMAP2', '', 'REPORT'))
AS T
```

이 쿼리의 출력:

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	-

4개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속):

COMPRESS_DICT_SIZE	EXPAND_DICT_SIZE	ROWS_SAMPLED	PAGES_SAVED_PERCENT	BYTES_SAVED_PERCENT	AVG_COMPRESS_REC_LENGTH
0	0	0	0	0	0
1280	2562	-	-	-	-
1340	2232	-	-	-	-
0	0	0	0	0	0

## 사용 시 참고사항

- *tabschema* 및 *tablename* 모두 지정된 경우 정보가 해당 특정 테이블에만 리턴됩니다.
- *tabschema*이 지정되지만 *tablename*이 비어 있거나("") 널(NULL)인 경우, 정보는 주어진 스키마의 모든 테이블에 대해 리턴됩니다.
- *tabschema*가 비어 있거나("") 널(NULL)이고 *tablename*이 지정되면 오류가 리턴됩니다. 특정 테이블에 대한 정보를 검색하려면 테이블이 스키마와 테이블 이름 모두에서 식별되어야 합니다.
- *tabschema* 및 *tablename*이 모두 비어 있거나("") 널(NULL)이면 정보는 모든 테이블에 대해 리턴됩니다.
- *tabschema* 또는 *tablename*이 존재하지 않거나 *tablename*이 테이블 이름(유형 T), 구체화된 쿼리 테이블 이름(유형 S) 또는 계층 구조 테이블 이름(유형 H)에 해당되지 않으면, 비어 있는 결과 세트가 리턴됩니다.
- ADMIN\_GET\_TAB\_COMPRESS\_INFO 테이블 함수가 주어진 테이블에 대해 데이터를 검색하고 있는 경우, SYSTABLES의 해당 행에 대한 공유 잠금을 획득하게 되어 리턴되는 데이터의 일관성이 유지됩니다(예를 들어 정보가 검색되고 있는 동안 테이블이 변경되지 않도록 합니다). 테이블 함수 호출 기간 동안이 아니라 테이블에 대한 압축 정보를 검색하는 동안만 잠금이 보유됩니다.

## ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO 테이블 함수 메타데이터

표 232. ADMINTABCOMPRESSINFO 관리 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO 테이블 함수 메타데이터

컬럼 이름	데이터 유형	설명
TABSCHEMA	VARCHAR (128)	스키마 이름
TABNAME	VARCHAR (128)	테이블 이름
DBPARTITIONNUM	SMALLINT	데이터베이스 파티션 번호
DATA_PARTITION_ID	INTEGER	데이터 파티션 번호
COMPRESS_ATTR	CHAR(1)	테이블에서의 COMPRESS 속성 상태는 다음 중 하나가 될 수 있습니다. <ul style="list-style-type: none"> <li>• 'Y' = 행 압축이 예로 설정됩니다.</li> <li>• 'N' = 행 압축이 아니므로 설정됩니다.</li> </ul>

표 232. ADMIN\_TAB\_COMPRESS\_INFO 관리 뷰 및 ADMIN\_GET\_TAB\_COMPRESS\_INFO 테이블 함수 메타데이터 (계속)

컬럼 이름	데이터 유형	설명
DICTIONARY_BUILDER	VARCHAR (30)	사전을 빌드하는 데 사용된 코드 경로는 다음 중 하나일 수 있습니다. <ul style="list-style-type: none"> <li>• 'INSPECT' = INSPECT ROWCOMPESTIMATE</li> <li>• 'LOAD' = LOAD INSERT/REPLACE</li> <li>• 'NOT BUILT' = 사용 가능한 사전 없음</li> <li>• 'REDISTRIBUTE' = REDISTRIBUTE</li> <li>• 'REORG' = REORG RESETDICTIONARY</li> <li>• 'TABLE GROWTH' = INSERT</li> </ul>
DICTIONARY_BUILD_TIMESTAMP	TIMESTAMP	사전이 빌드된 시간소인. 시간소인은 초 단위입니다. 사용 가능한 사전이 없는 경우 시간소인은 널(NULL)입니다.
COMPRESS_DICT_SIZE	BIGINT	바이트 단위로 측정된 압축 사전 크기.
EXPAND_DICT_SIZE	BIGINT	바이트로 측정된 확장 사전 크기. 실행 기록 사전이 존재하면 이 값은 현재 및 실행 기록 사전 크기의 합입니다.
ROWS_SAMPLED	INTEGER	사전을 빌드하는 데 사용된 레코드 수. 압축 사전을 포함한 이주된 테이블은 이 컬럼에서 널(NULL)을 리턴합니다.
PAGES_SAVED_PERCENT	SMALLINT	압축으로 저장된 페이지의 백분율. 이 정보는 샘플 버퍼의 레코드 데이터만을 기반으로 합니다. 압축 사전을 포함한 이주된 테이블은 이 컬럼에서 널(NULL)을 리턴합니다.
BYTES_SAVED_PERCENT	SMALLINT	압축으로 절약되는 바이트의 백분율. 이 정보는 샘플 버퍼의 레코드 데이터만을 기반으로 합니다. 압축 사전을 포함한 이주된 테이블은 이 컬럼에서 널(NULL)을 리턴합니다.
AVG_COMPRESS_REC_LENGTH	SMALLINT	사전을 빌드하는 데 사용되는 레코드의 압축된 레코드 평균 길이. 압축 사전을 포함한 이주된 테이블은 이 컬럼에서 널(NULL)을 리턴합니다.

## GET\_DB\_CONFIG

주: 이 프로시저는 사용되지 않으며 350 페이지의 『DBCFG 관리 뷰 - 데이터베이스 구성 매개변수 정보 검색』으로 바뀌었습니다.

▶▶ GET\_DB\_CONFIG (—) ◀◀

스키마는 SYSPROC입니다.

GET\_DB\_CONFIG 프로시저는 데이터베이스 구성 정보를 리턴합니다. 프로시저는 어떠한 인수도 취하지 않습니다.

프로시저는 각 매개변수에 대한 컬럼을 포함하는 두 개의 행이 있는 단일 결과 세트를 리턴합니다. 첫 번째 컬럼은 아래에서와 같이 DBMCONFIG\_TYPE이라고 합니다.

표 233. GET\_DB\_CONFIG 프로시저가 리턴한 정보

컬럼 이름	데이터 유형	설명
DBCONFIG_TYPE	INTEGER	이 컬럼에서 값 0을 갖는 행이 디스크에 저장된 데이터베이스 구성 매개변수의 값을 포함합니다. 이 컬럼에서 값 1을 갖는 행이 메모리에 저장된 데이터베이스 구성 매개변수의 현재 값을 포함합니다.

이 프로시저에는 결과 세트를 저장하기 위해 전역 임시 테이블 DB\_CONFIG를 작성하는 데 사용되는 사용자 임시 테이블 스페이스가 필요합니다.

예 :

명령행 처리기(CLP)를 사용하여 *logretain* 및 *userexit* 데이터베이스 구성 매개변수 값을 변경하십시오. GET\_DB\_CONFIG 프로시저를 호출한 후 결과로서 생성되는 전역 임시 테이블(DB\_CONFIG)을 쿼리하여 최초(디스크 상에) 및 갱신된(메모리 상에) 값을 검색하십시오.

```
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 LOGRETAIN RECOVERY USEREXIT ON

CALL SYSPROC.GET_DB_CONFIG()

SELECT DBCONFIG_TYPE, LOGRETAIN, USEREXIT
      FROM SESSION.DB_CONFIG

CONNECT RESET
```

이 쿼리의 출력 예는 다음과 같습니다.

```
DBCONFIG_TYPE LOGRETAIN  USEREXIT
-----
0              1          1
1              0          0
```

2개의 레코드가 선택되었습니다.

## GET\_DBM\_CONFIG

주: 이 테이블 함수는 사용되지 않으며 352 페이지의 『DBCFCG 관리 뷰 - 데이터베이스 관리 프로그램 구성 매개변수 정보 검색』으로 교체됩니다.

▶▶—GET\_DBM\_CONFIG—(—)—▶▶

스키마는 SYSFUN입니다.

GET\_DBM\_CONFIG 테이블 함수는 데이터베이스 관리 프로그램 구성 정보를 리턴합니다. 함수는 어떠한 인수도 취하지 않습니다.

함수는 각 매개변수에 대한 컬럼을 포함하는 두 개의 행이 있는 테이블을 리턴합니다. 첫 번째 컬럼은 아래에서와 같이 DBMCONFIG\_TYPE이라고 합니다.

표 234. GET\_DBM\_CONFIG 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명
DBMCONFIG_TYPE	INTEGER	이 컬럼에서 값 0을 갖는 행이 디스크에 저장된 데이터베이스 관리 프로그램 구성 매개변수의 값을 포함합니다. 이 컬럼에서 값 1을 갖는 행이 메모리에 저장된 데이터베이스 관리 프로그램 구성 매개변수의 현재 값을 포함합니다.

예 :

명령행 처리기(CLP)를 사용하여 *numdb* 및 *diaglevel* 데이터베이스 관리 프로그램 구성 매개변수의 값을 변경한 후 최초(디스크 상에) 및 갱신된(메모리 상에) 값을 검색합니다.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
DBMCONFIG_TYPE NUMDB      DIAGLEVEL
-----
0                32          4
1                8           3
```

2개의 레코드가 선택되었습니다.

## Health 스냅샷 루틴

### HEALTH\_CONT\_HI

HEALTH\_CONT\_HI 테이블 함수는 데이터베이스에 있는 테이블 스페이스의 Health 스냅샷에서 테이블 스페이스 컨테이너에 대한 Health 표시기 정보를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

## 구문

```
▶▶—HEALTH_CONT_HI—(—dbname—,—dbpartitionnum—)————▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

## 권한 부여

HEALTH\_CONT\_HI 테이블 함수에 대한 EXECUTE 특권.

## 예 :

```
SELECT * FROM TABLE(HEALTH_CONT_HI(' ', -1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

```
SNAPSHOT_TIMESTAMP          CONTAINER_NAME          ...
-----
2006-02-13-12.30.40.759542  D:WDB2#NODE0000#SAMPLE#T0000000#C0000000.CAT  ...
2006-02-13-12.30.40.759542  D:WDB2#NODE0000#SAMPLE#T0000003#C0000000.LRG  ...
2006-02-13-12.30.40.759542  D:WDB2#NODE0000#SAMPLE#T0000004#C0000000.UTM  ...
2006-02-13-12.30.40.759542  D:WDB2#NODE0000#SAMPLE#T0000001#C0000000.TMP  ...
2006-02-13-12.30.40.759542  D:WDB2#NODE0000#SAMPLE#T0000002#C0000000.LRG  ...
```

5개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... 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 ...
```



이 쿼리의 출력(계속).

```

... 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              -

```

## 리턴되는 정보

표 235. HEALTH\_CONT\_HI 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
CONTAINER_NAME	VARCHAR(256)	container_name - 컨테이너 이름
NODE_NUMBER	INTEGER	node_number - 노드 번호
HI_ID	BIGINT	스냅샷 데이터 스트림에서 Health 표시기를 고유하게 식별하는 숫자
HI_VALUE	SMALLINT	Health 표시기 값
HI_TIMESTAMP	TIMESTAMP	경고를 생성한 날짜 및 시간
HI_ALERT_STATE	BIGINT	경고의 심각도
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 컬럼의 텍스트 설명
HI_FORMULA	VARCHAR(2048)	Health 표시기를 계산하는 데 사용되는 공식
HI_ADDITIONAL_INFO	VARCHAR(4096)	Health 표시기에 대한 추가 정보

## HEALTH\_CONT\_HI\_HIS

데이터베이스의 Health 스냅샷에서 컨테이너에 대한 Health 표시기 실행기록 정보를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

## 구문

```

▶▶—HEALTH_CONT_HI_HIS—(—dbname—,—dbpartitionnum—)————▶▶

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

*dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST

DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

#### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

### 권한 부여

HEALTH\_CONT\_HI\_HIS 테이블 함수에 대한 EXECUTE 특권

예 :

```
SELECT * FROM TABLE(HEALTH_CONT_HI_HIS(' ', -1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

SNAPSHOT_TIMESTAMP	CONTAINER_NAME	...
2006-02-13-12.30.41.915646	D:WDB2WNODE0000WSAMPLEWT0000000WC0000000.CAT	...
2006-02-13-12.30.41.915646	D:WDB2WNODE0000WSAMPLEWT0000000WC0000000.CAT	...
2006-02-13-12.30.41.915646	D:WDB2WNODE0000WSAMPLEWT0000003WC0000000.LRG	...
2006-02-13-12.30.41.915646	D:WDB2WNODE0000WSAMPLEWT0000003WC0000000.LRG	...
2006-02-13-12.30.41.915646	D:WDB2WNODE0000WSAMPLEWT0000004WC0000000.UTM	...
2006-02-13-12.30.41.915646	D:WDB2WNODE0000WSAMPLEWT0000004WC0000000.UTM	...
2006-02-13-12.30.41.915646	D:WDB2WNODE0000WSAMPLEWT0000001WC0000000.TMP	...
2006-02-13-12.30.41.915646	D:WDB2WNODE0000WSAMPLEWT0000001WC0000000.TMP	...
2006-02-13-12.30.41.915646	D:WDB2WNODE0000WSAMPLEWT0000002WC0000000.LRG	...
2006-02-13-12.30.41.915646	D:WDB2WNODE0000WSAMPLEWT0000002WC0000000.LRG	...

10개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	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	...

이 쿼리의 출력(계속).

...	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	-

## 리턴되는 정보

표 236. HEALTH\_CONT\_HI\_HIS 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
CONTAINER_NAME	VARCHAR(256)	<b>container_name</b> - 컨테이너 이름
NODE_NUMBER	INTEGER	<b>node_number</b> - 노드 번호
HI_ID	BIGINT	스냅샷 데이터 스트림에서 Health 표시기를 고유하게 식별하는 숫자
HI_TIMESTAMP	TIMESTAMP	경고를 생성한 날짜 및 시간
HI_VALUE	SMALLINT	Health 표시기 값
HI_ALERT_STATE	BIGINT	경고의 심각도
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 컬럼의 텍스트 설명
HI_FORMULA	VARCHAR(2048)	Health 표시기를 계산하는 데 사용되는 공식
HI_ADDITIONAL_INFO	VARCHAR(4096)	Health 표시기에 대한 추가 정보

## HEALTH\_CONT\_INFO

HEALTH\_CONT\_INFO 테이블 함수는 데이터베이스의 Health 스냅샷에서 컨테이너 정보를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

## 구문

▶▶—HEALTH\_CONT\_INFO—(—dbname—,—dbpartitionnum—)—▶▶

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

*dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST

DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

*dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

**권한 부여**

HEALTH\_CONT\_INFO 테이블 함수에 대한 EXECUTE 특권

**예 :**

```
SELECT * FROM TABLE(HEALTH_CONT_INFO(' ', -1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

SNAPSHOT_TIMESTAMP	CONTAINER_NAME	...
2006-02-13-12.30.40.541209	D:\WDB2\NODE0000\SAMPLE\T0000000\C0000000.CAT	...
2006-02-13-12.30.40.541209	D:\WDB2\NODE0000\SAMPLE\T0000003\C0000000.LRG	...
2006-02-13-12.30.40.541209	D:\WDB2\NODE0000\SAMPLE\T0000004\C0000000.UTM	...
2006-02-13-12.30.40.541209	D:\WDB2\NODE0000\SAMPLE\T0000001\C0000000.TMP	...
2006-02-13-12.30.40.541209	D:\WDB2\NODE0000\SAMPLE\T0000002\C0000000.LRG	...

5개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

TABLESPACE_NAME	NODE_NUMBER	...
SYSCATSPACE	-	...
SYSTOOLSPACE	-	...
SYSTOOLSTMPSPACE	-	...
TEMPSPACE1	-	...
USERSPACE1	-	...

이 쿼리의 출력(계속).

ROLLED_UP_ALERT_STATE	ROLLED_UP_ALERT_STATE_DETAIL
...	1 Normal
...	1 Normal
...	1 Normal
...	1 Normal
...	1 Normal

## 리턴되는 정보

표 237. HEALTH\_CONT\_INFO 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
CONTAINER_NAME	VARCHAR(256)	container_name - 컨테이너 이름
TABLESPACE_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름
NODE_NUMBER	INTEGER	node_number - 노드 번호
ROLLED_UP_ALERT_STATE	BIGINT	해당 스냅샷에서 캡처한 가장 심각한 경고 상태
ROLLED_UP_ALERT_STATE_DETAIL	VARCHAR(20)	ROLLED_UP_ALERT_STATE 컬럼의 텍스트 설명

## HEALTH\_DB\_HI

HEALTH\_DB\_HI 테이블 함수는 데이터베이스의 Health 스냅샷에서 Health 표시기 정보를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

## 구문

▶▶—HEALTH\_DB\_HI—(—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 데이터베이스 인스턴스 아래 있는 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

## 권한 부여

HEALTH\_DB\_HI 테이블 함수에 대한 EXECUTE 특권

예 :

```
SELECT * FROM TABLE(HEALTH_DB_HI('','-1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...

11개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	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	...

이 쿼리의 출력(계속).

...	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)
... -
...
...
...
...
...
... -
...
...
...
...
...
...
...
...
... -
...
...
...

```

이 쿼리의 출력(계속).

```

... 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". ... high watermark for shared sort
... memory is "0"
... The following are the related

```

```

... database configuration parameter
... settings: logprimary is "3",
... logsecond is "2", and logfilesiz
... 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 logfilesiz
... is "1000", blk_log_dsk_ful is
... "NO", userexit 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 "NWA". 자동
... statistics collection (AUTO_RUNSTATS)
... is set to "OFF".

```

## 리턴되는 정보

표 238. HEALTH\_DB\_HI 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
HI_ID	BIGINT	스냅샷 데이터 스트림에서 Health 표시기를 고유하게 식별하는 숫자
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
HI_VALUE	SMALLINT	Health 표시기 값
HI_TIMESTAMP	TIMESTAMP	경고를 생성한 날짜 및 시간
HI_ALERT_STATE	BIGINT	경고의 심각도



표 238. HEALTH\_DB\_HI 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 컬럼의 텍스트 설명
HI_FORMULA	VARCHAR(2048)	Health 표시기를 계산하는 데 사용되는 공식
HI_ADDITIONAL_INFO	VARCHAR(4096)	Health 표시기에 대한 추가 정보

## HEALTH\_DB\_HI\_HIS

HEALTH\_DB\_HI\_HIS 테이블 함수는 데이터베이스의 Health 스냅샷에서 Health 표시기 실행기록 정보를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 구문

▶▶HEALTH\_DB\_HI\_HIS(—dbname—,—dbpartitionnum—)▶▶

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 데이터베이스 인스턴스 아래 있는 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

#### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

### 권한 부여

HEALTH\_DB\_HI\_HIS 테이블 함수에 대한 EXECUTE 특권

예 :

```
SELECT * FROM TABLE(HEALTH_DB_HI_HIS(' ',-1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...
...				...

이 쿼리의 출력(계속).

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	...
...			...

이 쿼리의 출력(계속).

HI_FORMULA	...
0	...
...	...
((0 / 5000) * 100)	...



```

... memory is "15". "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 "15". ... 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". ... last automated backup returned with
... SQLCODE = "N/A". The longest
... estimated backup time is "N/A".

...
... The scope is "NWA". 자동
... statistics collection
... (AUTO_RUNSTATS) is set to "OFF".

...

```

## 리턴되는 정보

표 239. HEALTH\_DB\_HI\_HIS 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
HI_ID	BIGINT	스냅샷 데이터 스트림에서 Health 표시기를 고유하게 식별하는 숫자

표 239. HEALTH\_DB\_HI\_HIS 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
HI_VALUE	SMALLINT	Health 표시기 값
HI_TIMESTAMP	TIMESTAMP	경고를 생성한 날짜 및 시간
HI_ALERT_STATE	BIGINT	경고의 심각도
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 컬럼의 텍스트 설명
HI_FORMULA	VARCHAR(2048)	Health 표시기를 계산하는 데 사용되는 공식
HI_ADDITIONAL_INFO	VARCHAR(4096)	Health 표시기에 대한 추가 정보

## HEALTH\_DB\_HIC

HEALTH\_DB\_HIC 함수는 데이터베이스의 Health 표시기에서 콜렉션 Health 표시기를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 구문

►►—HEALTH\_DB\_HIC—(—dbname—,—dbpartitionnum—)—————►►

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 데이터베이스 인스턴스 아래 있는 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

#### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

## 권한 부여

HEALTH\_DB\_HIC 테이블 함수에 대한 EXECUTE 특권

예 :

```
SELECT * FROM TABLE(HEALTH_DB_HIC('','-1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

```
SNAPSHOT_TIMESTAMP          HI_ID    DB_NAME    ...
-----
2006-02-13-12.30.33.870959    1015    SAMPLE    ...
2006-02-13-12.30.33.870959    1022    SAMPLE    ...
```

2개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... HI_OBJ_NAME                HI_OBJ_DETAIL    ...
... -----
... "JESSICAE"."EMPLOYEE"      REORG TABLE     ...
... "SYSIBM"."SYSDATAPARTITIONEXPRESSION" RUNSTATS         ...
```

이 쿼리의 출력(계속).

```
... 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
```

## 리턴되는 정보

표 240. HEALTH\_DB\_HIC 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
HI_ID	BIGINT	스냅샷 데이터 스트림에서 Health 표시기를 고유하게 식별하는 숫자
DB_NAME	VARCHAR (128)	<b>db_name</b> - 데이터베이스 이름
HI_OBJ_NAME	VARCHAR(256)	컬렉션에서 오브젝트를 고유하게 식별하는 이름.
HI_OBJ_DETAIL	VARCHAR(4096)	오브젝트가 컬렉션에 추가된 이유를 설명하는 텍스트

표 240. HEALTH\_DB\_HIC 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
HI_OBJ_STATE	BIGINT	오브젝트 상태. 유효한 상태 (sqlmon.h에 정의된)는 다음과 같습니다. <ul style="list-style-type: none"> <li>• NORMAL (1). 해당 오브젝트에 대해 조치가 필요하지 않습니다.</li> <li>• ATTENTION (2). 해당 Health 표시기에 대해 자동화가 사용 가능하지 않으며, 수동으로 조치를 수행해야 합니다.</li> <li>• AUTOMATED (5). 해당 Health 표시기에 대해 자동화가 사용 가능하며, 자동으로 조치가 시작됩니다.</li> <li>• AUTOMATE_FAILED (6). 해당 Health 표시기에 대해 자동화가 사용 가능합니다. 조치가 시작되었으나 성공적으로 완료할 수 없습니다. 지금 수동 개입이 필요합니다.</li> </ul>
HI_OBJ_STATE_DETAIL	VARCHAR(20)	HI_OBJ_STATE 컬럼의 값에 대한 변환된 문자열 버전
HI_TIMESTAMP	TIMESTAMP	경고를 생성한 날짜 및 시간

## HEALTH\_DB\_HIC\_HIS

데이터베이스의 Health 스냅샷에서 콜렉션 Health 표시기 실행기록 정보를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 구문

▶▶—HEALTH\_DB\_HIC\_HIS—(—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST

DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 데이터베이스 인스턴스 아래 있는 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

*dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

**권한 부여**

HEALTH\_DB\_HIC\_HIS 테이블 함수에 대한 EXECUTE 특권

예 :

```
SELECT * FROM TABLE(HEALTH_DB_HIC_HIS(' ', -1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...

10개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

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	...

이 쿼리의 출력(계속).

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

```

## 리턴되는 정보

표 241. HEALTH\_DB\_HIC\_HIS 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
HI_HIS_ENTRY_NUM	INTEGER	실행기록 항목을 고유하게 식별하는 숫자
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
HI_ID	BIGINT	스냅샷 데이터 스트림에서 Health 표시기를 고유하게 식별하는 숫자
DB_NAME	VARCHAR(128)	<b>db_name</b> - 데이터베이스 이름
HI_OBJ_NAME	VARCHAR(256)	컬렉션에서 오브젝트를 고유하게 식별하는 이름.
HI_OBJ_STATE	BIGINT	오브젝트 상태. 유효한 상태 (sqlmon.h에 정의된)는 다음과 같습니다. <ul style="list-style-type: none"> <li>• NORMAL (1). 해당 오브젝트에 대해 조치가 필요하지 않습니다.</li> <li>• ATTENTION (2). 해당 Health 표시기에 대해 자동화가 사용 가능하지 않으며, 수동으로 조치를 수행해야 합니다.</li> <li>• AUTOMATED (5). 해당 Health 표시기에 대해 자동화가 사용 가능하며, 자동으로 조치가 시작됩니다.</li> <li>• AUTOMATE_FAILED (6). 해당 Health 표시기에 대해 자동화가 사용 가능합니다. 조치가 시작되었으나 성공적으로 완료할 수 없습니다. 지금 수동 개입이 필요합니다.</li> </ul>
HI_OBJ_STATE_DETAIL	VARCHAR(20)	HI_OBJ_STATE 컬럼의 값에 대한 변환된 문자열 버전
HI_TIMESTAMP	TIMESTAMP	경고를 생성한 날짜 및 시간

## HEALTH\_DB\_INFO

HEALTH\_DB\_INFO 테이블 함수는 데이터베이스의 Health 스냅샷에서 정보를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 구문

```
▶▶—HEALTH_DB_INFO—(—dbname—,—dbpartitionnum—)————▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 데이터베이스 인스턴스 아래 있는 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

### 권한 부여

HEALTH\_DB\_INFO 테이블 함수에 대한 EXECUTE 특권

**예 :**

```
SELECT * FROM TABLE(HEALTH_DB_INFO(' ', -1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

SNAPSHOT_TIMESTAMP	DB_NAME	INPUT_DB_ALIAS	...
2006-02-13-12.30.23.340081	SAMPLE	SAMPLE	...

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... DB_PATH DB_LOCATION SERVER_PLATFORM ...
... -----
... D:#DB2#NODE0000#SQL00003# 1 5 ...
```

이 쿼리의 출력(계속).

```
... ROLLED_UP_ALERT_STATE ROLLED_UP_ALERT_STATE_DETAIL
... -----
... 4 Alarm
```

## 리턴되는 정보

표 242. HEALTH\_DB\_INFO 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR(128)	db_name - 데이터베이스 이름
INPUT_DB_ALIAS	VARCHAR(128)	input_db_alias - 데이터베이스 별명 입력
DB_PATH	VARCHAR(1024)	db_path - 데이터베이스 경로
DB_LOCATION	INTEGER	db_location - 데이터베이스 위치
SERVER_PLATFORM	INTEGER	server_platform - 서버 운영 체제
ROLLED_UP_ALERT_STATE	BIGINT	해당 스냅샷에서 캡처한 가장 심각한 경고 상태
ROLLED_UP_ALERT_STATE_DETAIL	VARCHAR(20)	ROLLED_UP_ALERT_STATE 컬럼의 텍스트 설명

## HEALTH\_DBM\_HI

HEALTH\_DBM\_HI 테이블 함수는 DB2 데이터베이스 관리 프로그램의 Health 스냅샷에서 Health 표시기 정보를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 구문

```
▶▶—HEALTH_DBM_HI—(—dbpartitionnum—)————▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

*dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데

이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

## 권한 부여

HEALTH\_DBM\_HI 테이블 함수에 대한 EXECUTE 특권

예 :

```
SELECT * FROM TABLE(HEALTH_DBM_HI(-1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

```
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      ...
```

2개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... HI_VALUE HI_TIMESTAMP      HI_ALERT_STATE HI_ALERT_STATE_DETAIL ...
... HI_VALUE HI_TIMESTAMP      HI_ALERT_STATE HI_ALERT_STATE_DETAIL ...
-----
...      0 2006-02-13-12.26.26.158000      1 Normal      ...
...      100 2006-02-13-12.26.26.158000      4 Alarm      ...
```

이 쿼리의 출력(계속).

```
... HI_FORMULA      HI_ADDITIONAL_INFO
... -----
... 0      -
... ((327680 / 327680) * 100)      -
```

표 243. HEALTH\_DBM\_HI 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
HI_ID	BIGINT	스냅샷 데이터 스트림에서 Health 표시기를 고유하게 식별하는 숫자
SERVER_INSTANCE_NAME	VARCHAR(128)	server_instance_name - 서버 인스턴스 이름
HI_VALUE	SMALLINT	Health 표시기 값
HI_TIMESTAMP	TIMESTAMP	경고를 생성한 날짜 및 시간
HI_ALERT_STATE	BIGINT	경고의 심각도
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 컬럼의 텍스트 설명
HI_FORMULA	VARCHAR(2048)	Health 표시기를 계산하는 데 사용되는 공식

표 243. HEALTH\_DBM\_HI 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
HI_ADDITIONAL_INFO	VARCHAR(4096)	Health 표시기에 대한 추가 정보

## HEALTH\_DBM\_HI\_HIS

HEALTH\_DBM\_HI\_HIS 테이블 함수는 DB2 데이터베이스 관리 프로그램의 Health 스냅샷에서 Health 표시기 실행기록 정보를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 구문

▶▶—HEALTH\_DBM\_HI\_HIS—(—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

### 권한 부여

HEALTH\_DBM\_HI\_HIS 테이블 함수에 대한 EXECUTE 특권

**예 :**

```
SELECT * FROM TABLE(HEALTH_DBM_HI_HIS(-1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

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	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 ...
2006-02-13-12.30.20.460905	4 DB2	60 ...

18개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

... 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	...

이 쿼리의 출력(계속).

... HI_FORMULA	HI_ADDITIONAL_INFO
...	-----
... 0	-
... 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)	-

## 리턴되는 정보

표 244. HEALTH\_DBM\_HI\_HIS 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
HI_ID	BIGINT	스냅샷 데이터 스트림에서 Health 표시기를 고유하게 식별하는 숫자
SERVER_INSTANCE_NAME	VARCHAR (128)	server_instance_name - 서버 인스턴스 이름
HI_VALUE	SMALLINT	Health 표시기 값
HI_TIMESTAMP	TIMESTAMP	경고를 생성한 날짜 및 시간
HI_ALERT_STATE	BIGINT	경고의 심각도
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 컬럼의 텍스트 설명
HI_FORMULA	VARCHAR(2048)	Health 표시기를 계산하는 데 사용되는 공식
HI_ADDITIONAL_INFO	VARCHAR(4096)	Health 표시기에 대한 추가 정보

## HEALTH\_DBM\_INFO

HEALTH\_DBM\_INFO 함수는 DB2 데이터베이스 관리 프로그램의 Health 스냅샷에서 정보를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

## 구문

▶▶—HEALTH\_DBM\_INFO—(—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

## 권한 부여

HEALTH\_DBM\_INFO 테이블 함수에 대한 EXECUTE 특권

예 :

```
SELECT * FROM TABLE(HEALTH_DBM_INFO(-1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

```
SNAPSHOT_TIMESTAMP      SERVER_INSTANCE_NAME      ROLLED_UP_ALERT_STATE ...
-----
2006-02-13-12.30.19.663924 DB2                                4 ...
```

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... ROLLED_UP_ALERT_STATE_DETAIL DB2START_TIME      ...
... -----
... Alarm                                2006-02-09-10.56.18.126182 ...
```

이 쿼리의 출력(계속).

```
... LAST_RESET          NUM_NODES_IN_DB2_INSTANCE
... -----
... -                                1
```

## 리턴되는 정보

표 245. HEALTH\_DBM\_INFO 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
SERVER_INSTANCE_NAME	VARCHAR (128)	server_instance_name - 서버 인스턴스 이름
ROLLED_UP_ALERT_STATE	BIGINT	해당 스냅샷에서 캡처한 가장 심각한 경고 상태
ROLLED_UP_ALERT_STATE_DETAIL	VARCHAR(20)	ROLLED_UP_ALERT_STATE 컬럼의 텍스트 설명
DB2START_TIME	TIMESTAMP	db2start_time - 데이터베이스 관리 프로그램 시작 시간소인
LAST_RESET	TIMESTAMP	last_reset - 마지막 재설정 시간소인
NUM_NODES_IN_DB2_INSTANCE	INTEGER	num_nodes_in_db2_instance - 데이터베이스 파티션의 노드 수

## HEALTH\_GET\_ALERT\_ACTION\_CFG

다양한 오브젝트 유형(데이터베이스 관리 프로그램, 데이터베이스, 테이블 스페이스 및 테이블 스페이스 컨테이너) 및 다양한 구성 레벨(설치 디폴트, 인스턴스, 전역 및 오브젝트)에 대한 Health 경고 조치 구성 설정값을 리턴합니다.



**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

## 구문

```
▶—HEALTH_GET_ALERT_ACTION_CFG—(—objecttype—,—cfg_level—,—dbname—,—  
▶—objectname—)
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *objecttype*

오브젝트 유형을 나타내는 VARCHAR(3) 유형의 입력 인수. 값은 대소문자를 구별하지 않는 다음 값 중 하나여야 합니다.

- 데이터베이스 관리 프로그램의 경우 'DBM'
- 데이터베이스의 경우 'DB'
- 테이블 스페이스의 경우 'TS'
- 테이블 스페이스 컨테이너의 경우 'TSC'

주: 앞뒤 공백은 무시됩니다.

### *cfg\_level*

구성 레벨을 나타내는 VARCHAR(1) 유형의 입력 인수. 값은 대소문자를 구별하지 않는 다음 값 중 하나여야 합니다.

- *objecttype* 'DBM'의 경우: 설치 디폴트값의 경우 'D', 인스턴스 레벨의 경우 'G' 또는 'O'
- 'DBM'이 아닌 *objecttype*의 경우: 설치 디폴트값의 경우 'D', 전역 레벨의 경우 'G', 오브젝트 레벨의 경우 'O'

### *dbname*

데이터베이스 이름을 나타내는 VARCHAR(128) 유형의 입력 인수. 데이터베이스 이름은 *objecttype*이 'DB', 'TS' 또는 'TSC'이고 *cfg\_level*이 'O'인 경우 제공되어야 합니다. *objecttype* 및 *cfg\_level*의 기타 모든 조합의 경우, *dbname* 매개변수는 NULL(또는 공백 문자열)이어야 합니다.

### *objectname*

오브젝트 이름을 나타내는 VARCHAR(1024) 유형의 입력 인수(예: <테이블 스페이스 이름> 또는 <테이블 스페이스 이름>.<컨테이너 이름>). 오브젝트 이름은 *objecttype*이 'TS' 또는 'TSC'이고 *cfg\_level*이 'O'인 경우 제공되어야 합니다. *objecttype* 및 *cfg\_level*의 기타 모든 조합의 경우, *objectname* 매개변수는 NULL(또는 공백 문자열)이어야 합니다.

## 권한 부여

HEALTH\_GET\_ALERT\_ACTION\_CFG 테이블 함수에 대한 EXECUTE 특권

예:

예 1: Health 표시기 ID 1004에 대한 데이터베이스 SAMPLE의 오브젝트 레벨 정보 조치 구성 설정값을 검색하십시오.

```
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
```

다음은 이 쿼리에 대한 출력의 예입니다.

OBJECTTYPE	CFG_LEVEL	DBNAME	OBJECTNAME	ID	IS_DEFAULT	CONDITION
DB	0	SAMPLE		1004	1	ALARM
DB	0	SAMPLE		1004	1	ALARM

2개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

ACTIONTYPE	ACTIONNAME	USERID	HOSTNAME
S	~/health_center/script/scrpn6	uid1	-
T	00.0005	uid1	HOST3

이 쿼리의 출력(계속).

SCRIPT_TYPE	WORKING_DIR	TERMINATION_CHAR	PARAMETERS
0	~/health_c	-	-
-	-	-	-

예 2: Health 표시기 ID 1004에 대한 데이터베이스 SAMPLE의 조건, 조치 유형, 조치 이름, 호스트 이름 및 스크립트 유형을 검색하십시오.

```
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
```

다음은 이 쿼리에 대한 출력의 예입니다.

CONDITION	ACTIONTYPE	ACTIONNAME	...
ALARM	S	~/health_center/script/scrpn6	...
ALARM	T	00.0005	...

2개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```

... USERID  HOSTNAME  SCRIPT_TYPE
... -----
... uid1    -          0
... uid1    HOST3     -

```

## 사용 시 참고사항

HEALTH\_GET\_IND\_DEFINITION 테이블 함수는 Health 표시기 이름에 Health 표시기 ID를 맵핑하는 데 사용할 수 있습니다.

## 리턴되는 정보

표 246. HEALTH\_GET\_ALERT\_ACTION\_CFG 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명
OBJECTTYPE	VARCHAR(3)	오브젝트 유형
CFG_LEVEL	CHAR(1)	구성 레벨
DBNAME	VARCHAR(128)	데이터베이스 이름
OBJECTNAME	VARCHAR(512)	오브젝트 이름
ID	BIGINT	Health 표시기 ID
IS_DEFAULT	SMALLINT	설정값이 디폴트값인지 여부: 디폴트값인 경우 1, 디폴트값이 아닌 경우 0, 적용할 수 없는 경우 널(NULL)
CONDITION	VARCHAR(512)	조치가 트리거된 경보 조건
ACTIONTYPE	CHAR(1)	조치 유형: 스크립트 조치의 경우 'S' 또는 태스크 조치의 경우 'T'
ACTIONNAME	VARCHAR(5000)	ACTIONTYPE이 'S'인 경우 스크립트 경로 이름, ACTIONTYPE이 'T'인 경우 태스크 ID
USERID	VARCHAR(1024)	조치가 실행될 사용자 이름
HOSTNAME	VARCHAR(255)	호스트 시스템 이름
SCRIPT_TYPE	CHAR(1)	스크립트 유형: ACTIONTYPE이 'S'이면, 운영 체제 명령 스크립트의 경우 'O' 또는 DB2 명령 스크립트의 경우 'D'이고, ACTIONTYPE이 'T'이면, 널(NULL).
WORKING_DIR	VARCHAR(5000)	ACTIONTYPE이 'S'인 경우 작업 디렉토리 또는 ACTIONTYPE이 'T'인 경우 널(NULL)
TERMINATION_CHAR	VARCHAR(4)	DB2 명령 스크립트 조치인 경우 명령문 종료 문자, 그렇지 않을 경우 널(NULL).
PARAMETERS	VARCHAR(200)	운영 체제 명령 스크립트 조치인 경우 명령행 매개변수

## HEALTH\_GET\_ALERT\_CFG

다양한 오브젝트 유형(데이터베이스 관리 프로그램, 데이터베이스, 테이블 스페이스, 테이블 스페이스 컨테이너) 및 다양한 구성 레벨(설치 디폴트, 전역 및 오브젝트)에 대한 Health 정보 구성 설정값을 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 구문

```
►►—HEALTH_GET_ALERT_CFG—(—objecttype—,—cfg_level—,—dbname—,——————►  
►—objectname—)—————►
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *objecttype*

오브젝트 유형을 나타내는 VARCHAR(3) 유형의 입력 인수. 값은 대소문자를 구별하지 않는 다음 값 중 하나여야 합니다.

- 데이터베이스 관리 프로그램의 경우 ‘DBM’
- 데이터베이스의 경우 ‘DB’
- 테이블 스페이스의 경우 ‘TS’
- 테이블 스페이스 컨테이너의 경우 ‘TSC’

주: 앞뒤 공백은 무시됩니다.

#### *cfg\_level*

구성 레벨을 나타내는 VARCHAR(1) 유형의 입력 인수. 값은 대소문자를 구별하지 않는 다음 값 중 하나여야 합니다.

- *objecttype* ‘DBM’의 경우: 설치 디폴트값의 경우 ‘D’, 인스턴스 레벨의 경우 ‘G’ 또는 ‘O’
- ‘DBM’이 아닌 *objecttype*의 경우: 설치 디폴트값의 경우 ‘D’, 전역 레벨의 경우 ‘G’, 오브젝트 레벨의 경우 ‘O’

#### *dbname*

데이터베이스 이름을 나타내는 VARCHAR(128) 유형의 입력 인수. 데이터베이스 이름은 *objecttype*이 ‘DB’, ‘TS’ 또는 ‘TSC’이고 *cfg\_level*이 ‘O’인 경우 제공되

어야 합니다. *objecttype* 및 *cfg\_level*의 기타 모든 조합의 경우, *dbname* 매개변수는 NULL(또는 공백 문자열)이어야 합니다.

*objectname*

오브젝트 이름을 나타내는 VARCHAR(1024) 유형의 입력 인수(예: <테이블 스페이스 이름> 또는 <테이블 스페이스 이름>.<컨테이너 이름>). 오브젝트 이름은 *objecttype*이 ‘TS’ 또는 ‘TSC’이고 *cfg\_level*이 ‘O’인 경우 제공되어야 합니다. *objecttype* 및 *cfg\_level*의 기타 모든 조합의 경우, *objectname* 매개변수는 NULL(또는 공백 문자열)이어야 합니다.

**권한 부여**

HEALTH\_GET\_ALERT\_CFG 테이블 함수에 대한 EXECUTE 특권

**예:**

예 1: 데이터베이스 SAMPLE의 오브젝트 레벨 정보 조치 구성 설정값을 검색하십시오.

```
SELECT * FROM TABLE(SYSPROC.HEALTH_GET_ALERT_CFG('DB','O','SAMPLE',''))
AS ALERT_CFG
```

다음은 이 쿼리에 대한 출력의 예입니다.

OBJECTTYPE	CFG_LEVEL	DBNAME	OBJECTNAME	...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
DB	0	SAMPLE		...
...				...

이 쿼리의 출력(계속).

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 ...

```

이 쿼리의 출력(계속).

```

... ALARM_THRESHOLD      SENSITIVITY      EVALUATE ACTION_ENABLED
... -----
...           0           0           0           0
...           0           0           1           0
...           0           0           1           0
...           0           0           1           0
...           0           0           1           0
...           100         0           0           0
...           50          0           1           0
...           30          0           1           0
...           85          0           1           0
...           85          0           1           0
...           10          0           1           0
...           85          0           1           0
...           10          0           1           0
...           70          0           1           0
...           70          0           0           0

```

예 2: 데이터베이스 SAMPLE에서 USERSPACE1 테이블 스페이스에 대한 Health 표시기 '2002'의 경고 및 알람 임계값을 검색하십시오.

```

SELECT WARNING_THRESHOLD, ALARM_THRESHOLD
      FROM TABLE(SYSPROC.HEALTH_GET_ALERT_CFG('TS','O','SAMPLE','USERSPACE1'))
      AS T WHERE ID = 2002

```

다음은 이 쿼리에 대한 출력의 예입니다.

```

WARNING_THRESHOLD      ALARM_THRESHOLD
-----
                               80                               90
SQL22004N 제공된 오브젝트의 요청된 구성을 찾을 수 없습니다.
"tablespaces"의 디폴트 구성을 리턴합니다.

```

1개의 경고 메시지가 인쇄된 1개의 레코드가 선택되었습니다.

## 사용 시 참고사항

HEALTH\_GET\_IND\_DEFINITION 테이블 함수는 Health 표시기 이름에 Health 표시기 ID를 맵핑하는 데 사용할 수 있습니다.

예: 데이터베이스 SAMPLE에서 USERSPACE1 테이블 스페이스에 대한 Health 표시기 테이블 스페이스 사용도(ts.ts\_util)의 경고 및 알람 임계값을 검색하십시오.

```

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','O','SAMPLE','USERSPACE1')) AS T,
      HINAME AS H
      WHERE T.ID = H.ID

```

다음은 이 쿼리에 대한 출력의 예입니다.

```

WARNING_THRESHOLD    ALARM_THRESHOLD
-----
                                80                                90
SQL22004N 제공된 오브젝트의 요청된 구성을 찾을 수 없습니다.
"tablespaces"의 디폴트 구성을 리턴합니다.

```

1개의 경고 메시지가 인쇄된 1개의 레코드가 선택되었습니다.

## 리턴되는 정보

표 247. HEALTH\_GET\_ALERT\_CFG 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명
OBJECTTYPE	VARCHAR(3)	오브젝트 유형
CFG_LEVEL	VARCHAR(1)	구성 레벨
DBNAME	VARCHAR (128)	데이터베이스 이름
OBJECTNAME	VARCHAR(512)	오브젝트 이름
ID	BIGINT	Health 표시기 ID
IS_DEFAULT	SMALLINT	설정값이 디폴트값인지 여부: 디폴트값인 경우 1, 디폴트값이 아닌 경우 0, 적용할 수 없는 경우 널(NULL)
WARNING_THRESHOLD	BIGINT	경고 임계값. 적용할 수 없는 경우 널(NULL)
ALARM_THRESHOLD	BIGINT	알람 임계값. 적용할 수 없는 경우 널(NULL)
SENSITIVITY	BIGINT	Health 표시기 민감도
EVALUATE	SMALLINT	Health 표시기가 평가되고 있는 경우 1 또는 평가되고 있지 않은 경우 0
ACTION_ENABLED	SMALLINT	경보 어커런스에 대해 조치를 실행할 수 있는 경우 1 또는 경보 어커런스에 대해 조치를 실행할 수 없는 경우 0.

## HEALTH\_GET\_IND\_DEFINITION

Health 표시기 정의 리턴

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

## 구문

▶▶—HEALTH\_GET\_IND\_DEFINITION—(—*locale*—)————▶▶

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *locale*

번역 가능한 출력이 리턴될 로케일을 나타내는 VARCHAR(33) 유형의 입력 인수. 데이터베이스 서버에서 입력 로케일이 지원되지 않는 경우, SQL 경고 메시지가 발행되며 디폴트값(영어)이 사용됩니다. 입력 로케일이 제공되지 않으면, 즉 해당 값이 NULL(또는 공백 문자열)이면, 디폴트 언어가 사용됩니다.

## 권한 부여

HEALTH\_GET\_IND\_DEFINITION 테이블 함수에 대한 EXECUTE 특권

### 예:

예 1: 프랑스어로 Health 표시기 db.db\_op\_status에 대한 유형 및 간단한 설명을 검색하십시오.

```
SELECT TYPE, SHORT_DESCRIPTION
       FROM TABLE(SYSPROC.HEALTH_GET_IND_DEFINITION('fr_FR'))
AS IND_DEFINITION WHERE NAME = 'db.db_op_status'
```

다음은 이 쿼리에 대한 출력의 예입니다.

TYPE	SHORT_DESCRIPTION
STATE	Etat opérationnel de la base de données

1개의 레코드가 선택되었습니다.

예 2: 영어로 Health 표시기 ID 1001에 대한 간단한 설명을 검색하십시오.

```
SELECT SHORT_DESCRIPTION FROM TABLE(SYSPROC.HEALTH_GET_IND_DEFINITION('en_US'))
AS IND_DEFINITION WHERE ID = 1001
```

다음은 이 쿼리에 대한 출력의 예입니다.

SHORT_DESCRIPTION
Database Operational State

예 3: 모든 Health 표시기 ID 및 이름을 검색하십시오.

```
SELECT ID, NAME FROM TABLE(HEALTH_GET_IND_DEFINITION('')) AS T
```

다음은 이 쿼리에 대한 출력의 예입니다.



```

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
...

```

## 리턴되는 정보

표 248. HEALTH\_GET\_IND\_DEFINITION 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명
ID	BIGINT	Health 표시기 ID
NAME	VARCHAR (128)	Health 표시기 이름
SHORT_DESCRIPTION	VARCHAR(1024)	Health 표시기 간단한 설명
LONG_DESCRIPTION	VARCHAR(32672)	Health 표시기 자세한 설명
TYPE	VARCHAR(16)	Health 표시기 유형. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• 'THRESHOLD_UPPER': 위로 바운드된 임계값 기반 Health 표시기</li> <li>• 'THRESHOLD_LOWER': 아래로 바운드된 임계값 기반 Health 표시기</li> <li>• 'STATE': 상태 기반 Health 표시기</li> <li>• 'COLLECTION_STATE': 콜렉션 상태 기반 Health 표시기</li> </ul>
UNIT	VARCHAR(1024)	Health 표시기 값 및 임계값 단위 또는 적용할 수 없는 경우 널 (NULL)
CATEGORY	VARCHAR(1024)	Health 표시기 범주
FORMULA	VARCHAR(512)	Health 표시기 공식
REFRESH_INTERVAL	BIGINT	Health 표시기 평가 간격(초)

## HEALTH\_HI\_REC

특정 DB2 오브젝트에서 경보 상태의 Health 표시기를 처리하는 권장사항 세트를 검색합니다. 권장사항은 경보 상태를 해결하기 위해 수행될 수 있는 조치에 대한 정보(예: 실행될 수 있는 스크립트)를 포함하는 XML 문서에 리턴됩니다.

**중요사항:** Health 모니터가 버전 9.7에서 더 이상 사용되지 않기 때문에 이 프로시저는 더 이상 사용되지 않으며 추후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책의 『Health 모니터가 사용되지 않음』 주제를 참조하십시오.

## 구문

```
►►—HEALTH_HI_REC—(—schema-version—,—indicator-id—,—dbname—,——————►
►—object-type—,—object-name—,—dbpartitionnum—,—client-locale—,——————►
►—recommendation-doc—)——————►►
```

스키마는 SYSPROC입니다.

해당 프로시저가 리턴하는 스크립트는 Health 표시기가 경고 상태를 입력한 인스턴스에서 호출되어야 합니다.

식별된 오브젝트에서 지정된 Health 표시기가 경고 상태가 아닌 경우 오류가 리턴됩니다(SQLSTATE 5U0ZZ).

## 프로시저 매개변수

### *schema-version*

유형 INTEGER의 입력 인수는 XML 문서를 표시하는 데 사용되는 스키마의 버전 ID를 지정합니다. 권장사항 문서는 해당 스키마 버전용으로 정의된 요소 및 속성만을 포함합니다. 유효한 스키마 버전은 sqllib 디렉토리의 포함 서브디렉토리에 있는 db2ApiDf.h에 정의됩니다.

### *indicator-id*

유형 INTEGER의 입력 인수는 요청 중인 권장사항에 대한 Health 표시기의 숫자 ID를 지정합니다. 유효한 Health 표시기 ID는 sqllib 디렉토리의 포함 서브디렉토리에 있는 sqlmon.h에 정의됩니다.

### *dbname*

VARCHAR(255) 유형의 입력 인수는 오브젝트 유형이 DB2HEALTH\_OBJTYPE\_TS\_CONTAINER, DB2HEALTH\_OBJTYPE\_TABLESPACE 또는 DB2HEALTH\_OBJTYPE\_DATABASE인 경우 및 경고 상태를 입력한 Health 표시기에 대해 데이터베이스의 별명 이름을 지정합니다. 그렇지 않으면, NULL을 지정하십시오.

### *object-type*

유형 INTEGER의 입력 인수는 경고 상태를 입력하는 Health 표시기에 대한 오브젝트 유형을 지정합니다. 유효한 오브젝트 유형은 sqllib 디렉토리의 포함 서브디렉토리에 있는 sqlmon.h에 정의됩니다.

### *object-name*

VARCHAR(255) 유형의 입력 인수는 오브젝트 유형이 DB2HEALTH\_OBJTYPE\_TABLESPACE 또는 DB2HEALTH\_OBJTYPE\_TS\_CONTAINER로 설정되는 경우 테이블 스페이스 컨테이너 또는 테이블 스페이스의 이름을 지정합니다. 오브젝트 유형이 DB2HEALTH\_OBJTYPE\_DATABASE 또는 DB2HEALTH\_OBJTYPE\_DATABASE\_MANAGER 인 경우 NULL을 지정하십시오. 테이블 스페이스 컨테이너의 경우, 오브젝트 이름은 *table\_space\_name.container\_name*으로 지정됩니다.

### *dbpartitionnum*

Health 표시기에서 경보 상태를 입력한 데이터베이스 파티션의 수를 지정하는 유형 INTEGER의 입력 인수. 올바른 값은 0 - 999, -1(현재 연결된 데이터베이스 파티션 지정) 및 -2(모든 데이터베이스 파티션 지정)입니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

### *client-locale*

유형 VARCHAR(33)의 입력 인수는 클라이언트 언어 ID를 지정합니다. 리턴할 권장사항의 언어를 지정하려면, 이 매개변수를 사용하십시오. 값이 지정되지 않으면, 'En\_US'(영어)가 사용됩니다. 지정된 로케일에 대한 메시지 파일을 서버에서 사용할 수 없는 경우, 디폴트값으로 'En\_US'가 사용됩니다.

### *recommendation-doc*

BLOB(2M) 유형의 출력 인수는 DB2 Health 권장사항 스키마 정의(sqllib 디렉토리의 misc 서브디렉토리에 있는 XML 스키마 DB2RecommendationSchema.xsd 참조)에 따라 형식화된 권장사항 문서(XML)를 포함합니다. XML 문서는 UTF-8로 인코드되며 문서 텍스트는 호출자의 로케일 또는 목표 인스턴스에 있는 호출자의 로케일에서 메시지를 사용할 수 없는 경우 영어로 인코드됩니다.

## HEALTH\_TBS\_HI

데이터베이스에 있는 테이블 스페이스의 Health 스냅샷에서 테이블 스페이스에 대한 Health 표시기 정보를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

### 구문

►►—HEALTH\_TBS\_HI—(—*dbname*—,—*dbpartitionnum*—)—————►►

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

## 권한 부여

HEALTH\_TBS\_HI 테이블 함수에 대한 EXECUTE 특권

예 :

```
SELECT * FROM TABLE(HEALTH_TBS_HI(' ', -1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...

11개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	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          ...

```

이 쿼리의 출력(계속).

```

... 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"
                           respectively. 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".
                           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"
                           respectively. 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".
                           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" respectively. 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".

current table space size (bytes) is
"6291456".

```

## 리턴되는 정보

표 249. HEALTH\_TBS\_HI 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TABLESPACE_NAME	VARCHAR (128)	<b>tablespace_name</b> - 테이블 스페이스 이름
HI_ID	BIGINT	스냅샷 데이터 스트림에서 Health 표시기를 고유하게 식별하는 숫자
HI_VALUE	SMALLINT	Health 표시기 값
HI_TIMESTAMP	TIMESTAMP	경고를 생성한 날짜 및 시간
HI_ALERT_STATE	BIGINT	경고의 심각도
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 컬럼의 텍스트 설명
HI_FORMULA	VARCHAR(2048)	Health 표시기를 계산하는 데 사용되는 공식
HI_ADDITIONAL_INFO	VARCHAR(4096)	Health 표시기에 대한 추가 정보

## HEALTH\_TBS\_HI\_HIS

HEALTH\_TBS\_HI\_HIS 테이블 함수는 데이터베이스의 Health 스냅샷에서 테이블 스페이스에 대한 Health 표시기 실행기록 정보를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

## 구문

```

▶▶—HEALTH_TBS_HI_HIS—(—dbname—,—dbpartitionnum—)————▶▶

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST

DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

*dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

**권한 부여**

HEALTH\_TBS\_HI\_HIS 테이블 함수에 대한 EXECUTE 특권

**예 :**

```
SELECT * FROM TABLE(HEALTH_TBS_HI_HIS(' ', -1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

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	...

22개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

...	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      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     ...

```

이 쿼리의 출력(계속).

```

... 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" respectively.
                        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" respectively.
                        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".          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".          current table space size (bytes) is
                        "38797312".

... 0                   -
... 0                   -
... ((156 / 252) * 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
table space is defined with automatic
storage set to "YES" and automatic resize
enabled set to "YES".

to be "N/A" and "N/A" respectively.

... ((156 / 252) * 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
table space is defined with automatic
storage set to "YES" and automatic resize
enabled set to "YES".

to be "N/A" and "N/A" respectively.

... 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". current table space size (bytes) is
"1048576".

... 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". current table space size (bytes) is
"1048576".

... 0 -
... 0 -
... 0 -
... 0 -
... 0 -
... 0 -
... ((1504 / 1504) * 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"
respectively. The table space is defined
with automatic storage set to "YES" and
automatic resize enabled set to "YES".

... ((1504 / 1504) * 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"
respectively. 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".
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

```

```

resize enabled set to "YES".
following are the automatic resize
storage set to "YES" and automatic
settings: increase size (bytes) "-1",
increase size (percent) "N/A", maximum
size (bytes) "-1". The current table
space size (bytes) is "6291456".

```

## 리턴되는 정보

표 250. HEALTH\_TBS\_HI\_HIS 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TABLESPACE_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
HI_ID	BIGINT	스냅샷 데이터 스트림에서 Health 표시기를 고유하게 식별하는 숫자
HI_TIMESTAMP	TIMESTAMP	경고를 생성한 날짜 및 시간
HI_VALUE	SMALLINT	Health 표시기 값
HI_ALERT_STATE	BIGINT	경고의 심각도
HI_ALERT_STATE_DETAIL	VARCHAR(20)	HI_ALERT_STATE 컬럼의 텍스트 설명
HI_FORMULA	VARCHAR(2048)	Health 표시기를 계산하는 데 사용되는 공식
HI_ADDITIONAL_INFO	VARCHAR(4096)	Health 표시기에 대한 추가 정보

## HEALTH\_TBS\_INFO

데이터베이스의 Health 스냅샷에서 테이블 스페이스 정보를 리턴합니다.

**중요사항:** Health Monitor가 버전 9.7에서 사용되지 않으므로 이 테이블 함수는 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『Health Monitor는 사용되지 않음』 주제를 참조하십시오.

## 구문

```

▶▶—HEALTH_TBS_INFO—(—dbname—,—dbpartitionnum—)—▶▶

```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에 -1을 지정하거나 모든 활성 데이터베이스 파티션의 집계에 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

### 권한 부여

HEALTH\_TBS\_INFO 테이블 함수에 대한 EXECUTE 특권

예 :

```
SELECT * FROM TABLE(HEALTH_TBS_INFO(' ', -1)) AS T
```

이 쿼리의 출력 예는 다음과 같습니다.

```

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              ...

```

5개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```

... ROLLED_UP_ALERT_STATE ROLLED_UP_ALERT_STATE_DETAIL
... -----
...                4 Alarm
...                1 Normal
...                1 Normal
...                1 Normal
...                4 Alarm

```

### 리턴되는 정보

표 251. HEALTH\_TBS\_INFO 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TABLESPACE_NAME	VARCHAR (128)	<b>tablespace_name</b> - 테이블 스페이스 이름
ROLLED_UP_ALERT_STATE	BIGINT	해당 스냅샷에서 캡처한 가장 심각한 경고 상태
ROLLED_UP_ALERT_STATE_DETAIL	VARCHAR(20)	ROLLED_UP_ALERT_STATE 컬럼의 텍스트 설명

## SNAP\_GET\_APPL 테이블 함수 - appl 논리 데이터 그룹 스냅샷 정보 검색

주: 이 테이블 함수는 사용되지 않으며 535 페이지의 『SNAPAPPL 관리 뷰 및 SNAP\_GET\_APPL\_V95 테이블 함수 - appl 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

SNAP\_GET\_APPL 테이블 함수는 응용프로그램 스냅샷, 특히 appl 논리 데이터 그룹에서 응용프로그램에 대한 정보를 리턴합니다.

SNAP\_GET\_AGENT, SNAP\_GET\_AGENT\_MEMORY\_POOL, SNAP\_GET\_APPL\_INFO, SNAP\_GET\_STMT 및 SNAP\_GET\_SUBSECTION 테이블 함수와 함께 사용된, SNAP\_GET\_APPL 테이블 함수는 GET SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동일한 정보를 제공하나, 모든 데이터베이스 파티션에서 데이터를 검색합니다.

리턴될 수 있는 전체 정보 목록은 1057 페이지의 표 252의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_APPL ( (dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 비어 있는 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 취하십시오. 널(NULL) 값을 지정하여 현재 연결된 데이터베이스와 동일한 인스턴스 내 모든 데이터베이스에서 스냅샷을 취하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우, *dbpartitionnum*에 -1이 내재적으로 설정됩니다. 이 입력 옵션이 사용되지 않으면, 즉, *dbname*만 제공되면, 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스가 응용프로그램에 의한 연결 및 사용 가능한 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 언제든 작성될 수 있으며, 이는 데이터가 통용되지 않음을 의미한다는 것을 알아두십시오. 해당 스냅샷 API 요청 유형이 있는 파일이 존재하지 않으면, SNAP\_GET\_APPL 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 수에 대한 스냅샷을 취합니다.

## 권한 부여

- SYSMON 권한
- SNAP\_GET\_APPL 테이블 함수에 대한 EXECUTE 특권.

## 예 :

모든 활성 데이터베이스에 대한 각 응용프로그램을 읽거나 작성하는 행에 대한 세부사항을 검색하십시오.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

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

## 리턴된 정보

표 252. SNAP\_GET\_APPL 테이블 함수로 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 취한 날짜 및 시간.
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
UOW_LOG_SPACE_USED	BIGINT	uow_log_space_used - 사용된 작업 단위(UOW) 로그 스페이스
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
INACT_STMTHIST_SZ	BIGINT	stmt_history_list_size - 명령문 실행기록 목록 크기
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수

표 252. SNAP\_GET\_APPL 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리페치 페이지 수
LOCKS_HELD	BIGINT	locks_held - 보유된 잠금 수

표 252. SNAP\_GET\_APPL 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_WAITS	BIGINT	lock_waits - 잠금 대기 수
LOCK_WAIT_TIME	BIGINT	lock_wait_time - 잠금 대기 시간
LOCK_ESCALS	BIGINT	lock_escalations - 잠금 에스컬레이션 수
X_LOCK_ESCALS	BIGINT	x_lock_escalations - 배타적 잠금 에스컬레이션 수
DEADLOCKS	BIGINT	deadlocks - 발견된 교착 상태 수
TOTAL_SORTS	BIGINT	total_sorts - 전체 정렬 수
TOTAL_SORT_TIME	BIGINT	total_sort_time - 총 정렬 시간
SORT_OVERFLOWS	BIGINT	sort_overflows - 정렬 오버플로우 수
COMMIT_SQL_STMTS	BIGINT	commit_sql_stmts - 시도된 명령문 커밋
ROLLBACK_SQL_STMTS	BIGINT	rollback_sql_stmts - 시도된 롤백 명령문
DYNAMIC_SQL_STMTS	BIGINT	dynamic_sql_stmts - 시도된 동적 SQL문
STATIC_SQL_STMTS	BIGINT	static_sql_stmts - 시도된 정적 SQL문
FAILED_SQL_STMTS	BIGINT	failed_sql_stmts - 실패한 명령문 조각
SELECT_SQL_STMTS	BIGINT	select_sql_stmts - 실행된 Select SQL문
DDL_SQL_STMTS	BIGINT	ddl_sql_stmts - DDL(Data Definition Language) SQL문
UID_SQL_STMTS	BIGINT	uid_sql_stmts - 실행된 UPDATE/INSERT/DELETE SQL문
INT_AUTO_REBINDS	BIGINT	int_auto_rebinds - 내부 자동 리바인드 수
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 삭제된 내부 행 수
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 갱신된 내부 행 수
INT_COMMITS	BIGINT	int_commits - 내부 커밋 수
INT_ROLLBACKS	BIGINT	int_rollback - 내부 롤백 수
INT_DEADLOCK_ROLLBACKS	BIGINT	int_deadlock_rollback - 교착 상태로 인한 내부 롤백 수
ROWS_DELETED	BIGINT	rows_deleted - 삭제된 행 수
ROWS_INSERTED	BIGINT	rows_inserted - 삽입된 행 수
ROWS_UPDATED	BIGINT	rows_updated - 갱신된 행 수
ROWS_SELECTED	BIGINT	rows_selected - 선택된 행 수
BINDS_PRECOMPILES	BIGINT	binds_precompiles - 시도된 바인드/프리컴파일 수
OPEN_REM_CURS	BIGINT	open_rem_curs - 열린 리모트 커서 수
OPEN_REM_CURS_BLK	BIGINT	open_rem_curs_blk - 블로킹 상태의 열린 리모트 커서 수
REJ_CURS_BLK	BIGINT	rej_curs_blk - 거부된 블록 커서 요청 수
ACC_CURS_BLK	BIGINT	acc_curs_blk - 승인된 블록 커서 요청 수
SQL_REQS_SINCE_COMMIT	BIGINT	sql_reqs_since_commit - 마지막 커밋 이후 SQL 요청 수

표 252. SNAP\_GET\_APPL 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_TIMEOUTS	BIGINT	lock_timeouts - 잠금 시간종료 수
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 삽입된 내부 행 수
OPEN_LOC_CURS	BIGINT	open_loc_curs - 열린 로컬 커서 수
OPEN_LOC_CURS_BLK	BIGINT	open_loc_curs_blk - 블로킹 상태의 열린 로컬 커서 수
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - 패키지 캐시 찾아보기 수
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - 패키지 캐시 삽입 수
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - 카탈로그 캐시 찾아보기 수
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - 카탈로그 캐시 삽입 수
CAT_CACHE_OVERFLOWS	BIGINT	cat_cache_overflows - 카탈로그 캐시 오버플로우 수
NUM_AGENTS	BIGINT	num_agents - 명령문에서 작업하는 에이전트 수
AGENTS_STOLEN	BIGINT	agents_stolen - 분실 에이전트 수
ASSOCIATED_AGENTS_TOP	BIGINT	associated_agents_top - 연관 에이전트의 최대 수
APPL_PRIORITY	BIGINT	appl_priority - 응용프로그램 에이전트 우선순위
APPL_PRIORITY_TYPE	VARCHAR(16)	appl_priority_type - 응용프로그램 우선순위 유형. 이 인터페이스는 sqlmon.h로 정의하는 것을 기초로 하고 다음 중 하나인 텍스트 ID를 리턴합니다. <ul style="list-style-type: none"> <li>• DYNAMIC_PRIORITY</li> <li>• FIXED_PRIORITY</li> </ul>
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - 프리페치 대기 시간
APPL_SECTION_LOOKUPS	BIGINT	appl_section_lookups - 섹션 찾아보기 수
APPL_SECTION_INSERTS	BIGINT	appl_section_inserts - 섹션 삽입 수
LOCKS_WAITING	BIGINT	locks_waiting - 잠금 대기 중인 현재 에이전트 수
TOTAL_HASH_JOINS	BIGINT	total_hash_joins - 전체 해시 조인 수
TOTAL_HASH_LOOPS	BIGINT	total_hash_loops - 전체 해시 루프 수
HASH_JOIN_OVERFLOWS	BIGINT	hash_join_overflows - 해시 조인 오버플로우 수
HASH_JOIN_SMALL_OVERFLOWS	BIGINT	hash_join_small_overflows - 해시 조인 작은 오버플로우 수
APPL_IDLE_TIME	BIGINT	appl_idle_time - 응용프로그램 유휴 시간



표 252. SNAP\_GET\_APPL 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
UOW_LOCK_WAIT_TIME	BIGINT	uow_lock_wait_time - 잠금 상태로 대기한 전체 작업 단위(UOW) 시간
UOW_COMP_STATUS	VARCHAR(14)	uow_comp_status - 작업 단위(UOW) 완료 상태. 이 인터페이스는 sqlmon.h로 정의하는 것을 기초로 하고 다음 중 하나인 텍스트 ID를 리턴합니다. <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 - 에이전트가 사용한 사용자 CPU 시간
AGENT_USR_CPU_TIME_MS	BIGINT	agent_usr_cpu_time - 에이전트가 사용한 사용자 CPU 시간
AGENT_SYS_CPU_TIME_S	BIGINT	agent_sys_cpu_time - 에이전트가 사용한 시스템 CPU 시간
AGENT_SYS_CPU_TIME_MS	BIGINT	agent_sys_cpu_time - 에이전트가 사용한 시스템 CPU 시간
APPL_CON_TIME	TIMESTAMP	appl_con_time - 연결 요청 시작 시간소인
CONN_COMPLETE_TIME	TIMESTAMP	conn_complete_time - 연결 요청 완료 시간소인
LAST_RESET	TIMESTAMP	last_reset - 마지막 재설정 시간소인
UOW_START_TIME	TIMESTAMP	uow_start_time - 작업 단위(UOW) 시작 시간소인
UOW_STOP_TIME	TIMESTAMP	uow_stop_time - 작업 단위(UOW) 중지 시간소인
PREV_UOW_STOP_TIME	TIMESTAMP	prev_uow_stop_time - 이전 작업 단위(UOW) 완료 시간소인
UOW_ELAPSED_TIME_S	BIGINT	uow_elapsed_time - 최근 작업 단위(UOW) 경과 시간
UOW_ELAPSED_TIME_MS	BIGINT	uow_elapsed_time - 최근 작업 단위(UOW) 경과 시간
ELAPSED_EXEC_TIME_S	BIGINT	elapsed_exec_time - 명령문 실행 경과 시간
ELAPSED_EXEC_TIME_MS	BIGINT	elapsed_exec_time - 명령문 실행 경과 시간
INBOUND_COMM_ADDRESS	VARCHAR(32)	inbound_comm_address - 인바운드 통신 주소

표 252. SNAP\_GET\_APPL 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOCK_TIMEOUT_VAL	BIGINT	lock_timeout_val - 잠금 시간종료(초)
PRIV_WORKSPACE_NUM_OVERFLOWS	BIGINT	priv_workspace_num_overflows - 개인용 작업 스페이스 오버플로우 수
PRIV_WORKSPACE_SECTION_INSERTS	BIGINT	priv_workspace_section_inserts - 개인용 작업 스페이스 섹션 삽입 수
PRIV_WORKSPACE_SECTION_LOOKUPS	BIGINT	priv_workspace_section_lookups - 개인용 작업 스페이스 섹션 찾아보기 수
PRIV_WORKSPACE_SIZE_TOP	BIGINT	priv_workspace_size_top - 개인용 작업 스페이스 최대 크기
SHR_WORKSPACE_NUM_OVERFLOWS	BIGINT	shr_workspace_num_overflows - 공유 작업 스페이스 오버플로우 수
SHR_WORKSPACE_SECTION_INSERTS	BIGINT	shr_workspace_section_inserts - 공유 작업 스페이스 섹션 삽입 수
SHR_WORKSPACE_SECTION_LOOKUPS	BIGINT	shr_workspace_section_lookups - 공유 작업 스페이스 섹션 찾아보기 수
SHR_WORKSPACE_SIZE_TOP	BIGINT	shr_workspace_size_top - 공유 작업 스페이스 최대 크기
DBPARTITIONNUM	SMALLINT	행에 대한 데이터의 데이터베이스 파티션 검색.
CAT_CACHE_SIZE_TOP	BIGINT	cat_cache_size_top - 카탈로그 캐시 상위 워터 마크(water mark)

## SNAP\_GET\_APPL\_INFO 테이블 함수 - appl\_info 논리 데이터 그룹 스냅샷 정보 검색

주: 이 테이블 함수는 더 이상 사용되지 않으며 526 페이지의 『SNAPAPPL\_INFO 관리 뷰 및 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수 - Retrieve appl\_info 논리 데이터 그룹 스냅샷 정보』에서 교체됩니다.

SNAP\_GET\_APPL\_INFO 테이블 함수는 응용프로그램 스냅샷, 특히 appl\_info 논리 데이터 그룹에서 응용프로그램에 대한 정보를 리턴합니다.

SNAP\_GET\_AGENT, SNAP\_GET\_AGENT\_MEMORY\_POOL, SNAP\_GET\_APPL, SNAP\_GET\_APPL\_INFO, SNAP\_GET\_STMT 및 SNAP\_GET\_SUBSECTION 테이블 함수와 함께 사용되는 SNAP\_GET\_APPL\_INFO 테이블 함수는 GET

SNAPSHOT FOR ALL APPLICATIONS CLP 명령과 동등한 정보를 제공하지만 모든 데이터베이스 파티션에서 데이터를 검색합니다.

리턴 가능한 전체 정보 목록은 1064 페이지의 표 253의 내용을 참조하십시오.

## 구문

```
▶▶—SNAP_GET_APPL_INFO—(—dbname— [ , dbpartitionnum ] )—▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우 -1이 *dbpartitionnum*에 대해 내재적으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_APPL\_INFO 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대해 스냅샷을 작성합니다.

## 권한 부여

- SYSMON 권한
- SNAP\_GET\_APPL\_INFO 테이블 함수의 EXECUTE 특권

## 예:

연결된 데이터베이스 파티션의 모든 응용프로그램에 대한 상태를 검색합니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

```
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
```

4개의 레코드가 선택되었습니다.

다음은 테이블 함수 결과에서 선택할 때 확보한 내용입니다.

```
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
```

다음은 이 쿼리의 출력 예입니다.

```
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)
```

3개의 레코드가 선택되었습니다.

## 리턴된 정보

표 253. SNAP\_GET\_APPL\_INFO 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)

표 253. SNAP\_GET\_APPL\_INFO 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
APPL_STATUS	VARCHAR(22)	<p>appl_status - 응용프로그램 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다.</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 - 응용프로그램에서 사용되는 코드 페이지 ID
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 연관 에이전트 수
COORD_NODE_NUM	SMALLINT	coord_node - 코디네이팅 노드

표 253. SNAP\_GET\_APPL\_INFO 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
AUTHORITY_LVL	VARCHAR(512)	<p>authority_lvl - 사용자 권한 부여 레벨.</p> <p>이 인터페이스는 sql.h에 정의된 데이터베이스 권한을 기반으로 하는 텍스트 ID 및 해당 소스를 리턴하고 다음 형식을 사용합니다. 권한(소스, ...) + 권한(소스,...) + ... 권한의 소스는 여러 가지 유형이 가능하며 USER, GROUP 또는 USER 및 GROUP 중 하나입니다.</p> <p>"권한"에 대한 가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• BINDADD</li> <li>• CONNECT</li> <li>• CREATE_EXT_RT</li> <li>• CREATE_NOT_FENC</li> <li>• CREATETAB</li> <li>• DBADM</li> <li>• IMPLICIT_SCHEMA</li> <li>• LOAD</li> <li>• LIBADM</li> <li>• QUIESCE_CONN</li> <li>• SECADM</li> <li>• SYSADM</li> <li>• SYSCTRL</li> <li>• SYSMANT</li> <li>• SYSMON</li> <li>• SYSQUIESCE</li> </ul> <p>"소스"에 대한 가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• USER - 사용자에게 부여된 권한 또는 사용자에게 부여된 역할</li> <li>• GROUP - 사용자가 속하는 그룹에 부여된 권한 또는 사용자가 속하는 그룹에 부여된 역할</li> </ul>
CLIENT_PID	BIGINT	client_pid - 클라이언트 프로세스 ID
COORD_AGENT_PID	BIGINT	coord_agent_pid - 코디네이터 에이전트
STATUS_CHANGE_TIME	TIMESTAMP	status_change_time - 응용프로그램 상태 변경 시간

표 253. SNAP\_GET\_APPL\_INFO 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
CLIENT_PLATFORM	VARCHAR(12)	<p>client_platform - 클라이언트 운영 플랫폼. 이 인터페이스는 sqlmon.h의 정의를 기반으로 텍스트 ID를 리턴합니다.</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> <li>• WINDOWS95</li> </ul>

표 253. SNAP\_GET\_APPL\_INFO 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
CLIENT_PROTOCOL	VARCHAR(10)	client_protocol - 클라이언트 통신 프로토콜. 이 인터페이스는 sqlmon.h의 정의를 기반으로 텍스트 ID를 리턴합니다.  <ul style="list-style-type: none"> <li>• CPIC</li> <li>• LOCAL</li> <li>• NETBIOS</li> <li>• NPIPE</li> <li>• TCPIP( DB2 UDB용)</li> <li>• TCPIP4</li> <li>• TCPIP6</li> </ul>
TERRITORY_CODE	SMALLINT	territory_code - 데이터베이스 지역 코드
APPL_NAME	VARCHAR(256)	appl_name - 응용프로그램 이름
APPL_ID	VARCHAR(128)	appl_id - 응용프로그램 ID
SEQUENCE_NO	VARCHAR(4)	sequence_no - 시퀀스 번호
PRIMARY_AUTH_ID	VARCHAR(128)	auth_id - 권한 부여 ID
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - 세션 권한 부여 ID
CLIENT_NNAME	VARCHAR(128)	client_nname 모니터 요소는 사용되지 않습니다. 리턴된 값은 올바른 값이 아닙니다.
CLIENT_PRDID	VARCHAR(128)	client_prdid - 클라이언트 제품/버전 ID
INPUT_DB_ALIAS	VARCHAR(128)	input_db_alias - 데이터베이스 별명 입력
CLIENT_DB_ALIAS	VARCHAR(128)	client_db_alias - 응용프로그램에서 사용되는 데이터베이스 별명
DB_NAME	VARCHAR(128)	db_name - 데이터베이스 이름
DB_PATH	VARCHAR(1024)	db_path - 데이터베이스 경로
EXECUTION_ID	VARCHAR(128)	execution_id - 사용자 로그인 ID
CORR_TOKEN	VARCHAR(128)	corr_token - DRDA 상환 토큰
TPMON_CLIENT_USERID	VARCHAR(256)	tpmon_client_userid - TP 모니터 클라이언트 사용자 ID
TPMON_CLIENT_WKSTN	VARCHAR(256)	tpmon_client_wkstn - TP 모니터 클라이언트 워크스테이션 이름
TPMON_CLIENT_APP	VARCHAR(256)	tpmon_client_app - TP 모니터 클라이언트 응용프로그램 이름
TPMON_ACC_STR	VARCHAR(200)	tpmon_acc_str - TP 모니터 클라이언트 어카운팅 문자열
DBPARTITIONNUM	SMALLINT	행의 데이터가 검색되는 데이터베이스 파티션입니다.



## SNAP\_GET\_BP 테이블 함수 - 버퍼 풀 논리 그룹 스냅샷 정보 검색

주: 이 테이블 함수는 사용되지 않으며 543 페이지의 『SNAPBP 관리 뷰 및 SNAP\_GET\_BP\_V95 테이블 함수 - 버퍼 풀 논리 그룹 스냅샷 정보 검색』으로 교체됩니다.

SNAP\_GET\_BP 테이블 함수는 버퍼 풀 논리적 데이터 그룹, 특히 버퍼 풀 스냅샷에서 버퍼 풀에 대한 정보를 리턴합니다.

SNAP\_GET\_BP\_PART 테이블 함수와 함께 사용되는 SNAP\_GET\_BP 테이블 함수는 GET SNAPSHOT FOR ALL BUFFERPOOLS CLP 명령과 동일한 데이터를 제공합니다.

리턴될 수 있는 정보의 완전한 목록은 1070 페이지의 표 254의 내용을 참조하십시오.

### 구문

```
▶▶ SNAP_GET_BP ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(128) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 비어 있는 문자열을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 취하십시오. 널(NULL) 값을 지정하여 현재 연결된 데이터베이스와 동일한 인스턴스 내 모든 데이터베이스에서 스냅샷을 취하십시오.

#### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 선택적 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우, *dbpartitionnum*에 -1이 내재적으로 설정됩니다. 이러한 입력 옵션이 사용되지 않으면 즉, *dbname*만 제공되면, 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스가 응용프로그램에 의한 연결 및 사용 가능한 파티션입니다.

*dbname*과 *dbpartitionnum*이 모두 NULL로 설정되면, SNAP\_WRITE\_FILE 프로시저로 작성된 파일의 데이터를 읽도록 시도됩니다. 이 파일은 언제든지 작성될 수 있으며, 이는 데이터가 통용되지 않음을 의미한다는 것을 알아두십시오. 해당 스냅샷 API 요청

유형이 있는 파일이 존재하지 않으면, SNAP\_GET\_BP 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 수에 대한 스냅샷을 취합니다.

## 권한 부여

- SYSMON 권한
- SNAP\_GET\_BP 테이블 함수에 대한 EXECUTE 특권.

## 예 :

현재 연결된 데이터베이스 파티션의 모든 활성 데이터베이스에 대한 모든 버퍼 풀의 전체 실제 및 논리적 읽기를 검색하십시오.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.

```
DB_NAME      BP_NAME      TOTAL_LOGICAL_READS  ...
-----
SAMPLE      IBMDEFAULTBP      0 ...
TOOLSDB     IBMDEFAULTBP      0 ...
TOOLSDB     BP32K0000         0 ...
```

3개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... TOTAL_PHYSICAL_READS DBPARTITIONNUM
... -----
...                0                0
...                0                0
...                0                0
```

## 리턴된 정보

표 254. SNAP\_GET\_BP 테이블 함수로 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 취한 날짜 및 시간.
BP_NAME	VARCHAR (128)	bp_name - 버퍼 풀 이름
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
DB_PATH	VARCHAR(1024)	db_path - 데이터베이스 경로
INPUT_DB_ALIAS	VARCHAR (128)	input_db_alias - 데이터베이스 별명 입력
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수

표 254. SNAP\_GET\_BP 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 수
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 수
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - 버퍼 풀 비동기 읽기 시간
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - 버퍼 풀 비동기 쓰기 시간
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 수
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수

표 254. SNAP\_GET\_BP 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리페치 페이지 수
FILES_CLOSED	BIGINT	files_closed - 닫힌 데이터베이스 파일 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - 버퍼 풀 희생 버퍼 없음
PAGES_FROM_BLOCK_IOS	BIGINT	pages_from_block_ios - 블록 입출력으로 읽은 총 페이지 수
PAGES_FROM_VECTORED_IOS	BIGINT	pages_from_vectored_ios - 벡터된 입출력으로 읽은 총 페이지 수
PHYSICAL_PAGE_MAPS	BIGINT	physical_page_maps 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
VECTORED_IOS	BIGINT	vectored_ios - 벡터된 입출력 요청 수
DBPARTITIONNUM	SMALLINT	이 행에서 검색된 데이터의 데이터베이스 파티션.

## SNAP\_GET\_CONTAINER

주: 이 테이블 함수는 사용되지 않으며 552 페이지의 『SNAPCONTAINER 관리 뷰 및 SNAP\_GET\_CONTAINER\_V91 테이블 함수 - tablespace\_container 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAP\_GET\_CONTAINER—(—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

SNAP\_GET\_CONTAINER 테이블 함수는 tablespace\_container 논리 데이터 그룹으로부터의 스냅샷 정보를 리턴합니다.

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하십시오. 널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수를 모두 NULL로 설정하면 해당 스냅샷 API 요청 유형에 대해 SNAPSHOT\_FILEW 스토어드 프로시저가 파일을 이전에 작성하지 않은 경우에만 스냅샷이 작성됩니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 255. SNAP\_GET\_CONTAINER 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TBSP_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
CONTAINER_NAME	VARCHAR(256)	container_name - 컨테이너 이름
CONTAINER_ID	BIGINT	container_id - 컨테이너 ID
CONTAINER_TYPE	SMALLINT	container_type - 컨테이너 유형
TOTAL_PAGES	BIGINT	container_total_pages - 컨테이너의 전체 페이지 수

표 255. SNAP\_GET\_CONTAINER 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
USABLE_PAGES	BIGINT	container_usable_pages - 컨테이너에서 사용할 수 있는 페이지 수
ACCESSIBLE	SMALLINT	container_accessible - 컨테이너의 액세스 가능성
STRIPE_SET	BIGINT	container_stripe_set - 스트라이프 세트
DBPARTITIONNUM	SMALLINT	node_number - 노드 번호

## SNAP\_GET\_DB

주: 이 테이블 함수는 사용되지 않으며 1083 페이지의 『SNAP\_GET\_DB\_V91 테이블 함수 - dbase 논리적 그룹에서 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAP\_GET\_DB—(—dbname—, —dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

SNAP\_GET\_DB 테이블 함수는 데이터베이스에서 스냅샷 정보를 리턴합니다.

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 256. SNAP\_GET\_DB 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
DB_NAME	VARCHAR (128)	<b>db_name</b> - 데이터베이스 이름
DB_PATH	VARCHAR(1024)	<b>db_path</b> - 데이터베이스 경로
INPUT_DB_ALIAS	VARCHAR (128)	<b>input_db_alias</b> - 입력 데이터베이스 별명
DB_STATUS	BIGINT	<b>db_status</b> - 데이터베이스 상태
CATALOG_PARTITION	SMALLINT	<b>catalog_node</b> - 카탈로그 노드 번호
CATALOG_PARTITION_NAME	VARCHAR (128)	<b>catalog_node_name</b> - 카탈로그 노드 네트워크 이름
SERVER_PLATFORM	INTEGER	<b>server_platform</b> - 서버 운영 체제
DB_LOCATION	INTEGER	<b>db_location</b> - 데이터베이스 위치
DB_CONN_TIME	TIMESTAMP	<b>db_conn_time</b> - 데이터베이스 활성화 시간소인
LAST_RESET	TIMESTAMP	<b>last_reset</b> - 마지막 재설정 시간소인
LAST_BACKUP	TIMESTAMP	<b>last_backup</b> - 최종 백업 시간소인
CONNECTIONS_TOP	BIGINT	<b>connections_top</b> - 최대 동시 연결 수
TOTAL_CONS	BIGINT	<b>total_cons</b> - 데이터베이스 활성화 이후 연결 수
TOTAL_SEC_CONS	BIGINT	<b>total_sec_cons</b> - 2차 연결 수
APPLS_CUR_CONS	BIGINT	<b>appls_cur_cons</b> - 현재 연결된 응용프로그램 수
APPLS_IN_DB2	BIGINT	<b>appls_in_db2</b> - 현재 데이터베이스에서 실행 중인 응용프로그램 수
NUM_ASSOC_AGENTS	BIGINT	<b>num_assoc_agents</b> - 연관 에이전트 수
AGENTS_TOP	BIGINT	<b>agents_top</b> - 작성된 에이전트 수
COORD_AGENTS_TOP	BIGINT	<b>coord_agents_top</b> - 최대 코디네이팅 에이전트 수
LOCKS_HELD	BIGINT	<b>locks_held</b> - 보유된 잠금 수
LOCK_WAITS	BIGINT	<b>lock_waits</b> - 잠금 대기 수
LOCK_WAIT_TIME	BIGINT	<b>lock_wait_time</b> - 잠금 대기 시간
LOCK_LIST_IN_USE	BIGINT	<b>lock_list_in_use</b> - 사용 중인 전체 잠금 목록 메모리
DEADLOCKS	BIGINT	<b>deadlocks</b> - 발견된 교착 상태 수
LOCK_ESCALS	BIGINT	<b>lock_escals</b> - 잠금 에스컬레이션 수
X_LOCK_ESCALS	BIGINT	<b>x_lock_escals</b> - 배타적 잠금 에스컬레이션 수
LOCKS_WAITING	BIGINT	<b>locks_waiting</b> - 잠금 대기 중인 현재 에이전트 수
LOCK_TIMEOUTS	BIGINT	<b>lock_timeouts</b> - 잠금 시간종료 수

표 256. SNAP\_GET\_DB 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
NUM_INDOUBT_TRANS	BIGINT	<b>num_indoubt_trans</b> - 인다우트(Indoubt) 트랜잭션 수
SORT_HEAP_ALLOCATED	BIGINT	<b>sort_heap_allocated</b> - 할당된 전체 정렬 힙
SORT_SHRHEAP_ALLOCATED	BIGINT	<b>sort_shrheap_allocated</b> - 현재 할당된 정렬 공유 힙
SORT_SHRHEAP_TOP	BIGINT	<b>sort_shrheap_top</b> - 정렬 공유 힙 상위 워터 마크(water mark)
TOTAL_SORTS	BIGINT	<b>total_sorts</b> - 전체 정렬 수
TOTAL_SORT_TIME	BIGINT	<b>total_sort_time</b> - 총 정렬 시간
SORT_OVERFLOWS	BIGINT	<b>sort_overflows</b> - 정렬 오버플로우 수
ACTIVE_SORTS	BIGINT	<b>active_sorts</b> - 활성 정렬 수
POOL_DATA_L_READS	BIGINT	<b>pool_data_l_reads</b> - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	<b>pool_data_p_reads</b> - 버퍼 풀 데이터 실제 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	<b>pool_temp_data_l_reads</b> - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	<b>pool_temp_data_p_reads</b> - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_ASYNC_DATA_READS	BIGINT	<b>pool_async_data_reads</b> - 버퍼 풀 비동기 데이터 읽기 수
POOL_DATA_WRITES	BIGINT	<b>pool_data_writes</b> - 버퍼 풀 데이터 쓰기 수
POOL_ASYNC_DATA_WRITES	BIGINT	<b>pool_async_data_writes</b> - 버퍼 풀 비동기 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	<b>pool_index_l_reads</b> - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	<b>pool_index_p_reads</b> - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	<b>pool_temp_index_l_reads</b> - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	<b>pool_temp_index_p_reads</b> - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_INDEX_WRITES	BIGINT	<b>pool_index_writes</b> - 버퍼 풀 인덱스 쓰기 수
POOL_ASYNC_INDEX_READS	BIGINT	<b>pool_async_index_reads</b> - 버퍼 풀 비동기 인덱스 읽기 수
POOL_ASYNC_INDEX_WRITES	BIGINT	<b>pool_async_index_writes</b> - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_READ_TIME	BIGINT	<b>pool_read_time</b> - 총 버퍼 풀 실제 읽기 시간



표 256. SNAP\_GET\_DB 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_WRITE_TIME	BIGINT	<b>pool_write_time</b> - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_READ_TIME	BIGINT	<b>pool_async_read_time</b> - 버퍼 풀 비동기 읽기 시간
POOL_ASYNC_WRITE_TIME	BIGINT	<b>pool_async_write_time</b> - 버퍼 풀 비동기 쓰기 시간
POOL_ASYNC_DATA_READ_REQS	BIGINT	<b>pool_async_data_read_reqs</b> - 버퍼 풀 비동기 읽기 요청 수
POOL_ASYNC_INDEX_READ_REQS	BIGINT	<b>pool_async_index_read_reqs</b> - 버퍼 풀 비동기 인덱스 읽기 요청 수
POOL_NO_VICTIM_BUFFER	BIGINT	<b>pool_no_victim_buffer</b> - 버퍼 풀 희생 버퍼 없음
POOL_LSN_GAP_CLNS	BIGINT	<b>pool_lsn_gap_clns</b> - 버퍼 풀 로그 스페이스 클리너 트리거
POOL_DRTY_PG_STEAL_CLNS	BIGINT	<b>pool_drty_pg_steal_clns</b> - 버퍼 풀 희생(victim) 페이지 클리너 트리거
POOL_DRTY_PG_THRSH_CLNS	BIGINT	<b>pool_drty_pg_thrsh_clns</b> - 버퍼 풀 임계값 클리너 트리거
PREFETCH_WAIT_TIME	BIGINT	<b>prefetch_wait_time</b> - 프리페치 대기 시간
UNREAD_PREFETCH_PAGES	BIGINT	<b>unread_prefetch_pages</b> - 읽지 않은 프리페치 페이지 수
DIRECT_READS	BIGINT	<b>direct_reads</b> - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	<b>direct_writes</b> - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	<b>direct_read_reqs</b> - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	<b>direct_write_reqs</b> - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	<b>direct_read_time</b> - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	<b>direct_write_time</b> - 직접 쓰기 시간
FILES_CLOSED	BIGINT	<b>files_closed</b> - 닫힌 데이터베이스 파일 수
POOL_DATA_TO_ESTORE	BIGINT	<b>pool_data_to_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_INDEX_TO_ESTORE	BIGINT	<b>pool_index_to_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_INDEX_FROM_ESTORE	BIGINT	<b>pool_index_from_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.

표 256. SNAP\_GET\_DB 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_DATA_FROM_ESTORE	BIGINT	<b>pool_data_from_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
ELAPSED_EXEC_TIME_S	BIGINT	<b>elapsed_exec_time</b> - 명령문 실행 경과 시간
ELAPSED_EXEC_TIME_MS	BIGINT	<b>elapsed_exec_time</b> - 명령문 실행 경과 시간
COMMIT_SQL_STMTS	BIGINT	<b>commit_sql_stmts</b> - 시도된 커밋 명령문
ROLLBACK_SQL_STMTS	BIGINT	<b>rollback_sql_stmts</b> - 시도된 롤백 명령문
DYNAMIC_SQL_STMTS	BIGINT	<b>dynamic_sql_stmts</b> - 시도된 동적 SQL문
STATIC_SQL_STMTS	BIGINT	<b>static_sql_stmts</b> - 시도된 정적 SQL문
FAILED_SQL_STMTS	BIGINT	<b>failed_sql_stmts</b> - 실패한 명령문 조각
SELECT_SQL_STMTS	BIGINT	<b>select_sql_stmts</b> - 실행된 Select SQL문
UID_SQL_STMTS	BIGINT	<b>uid_sql_stmts</b> - 실행된 UPDATE/INSERT/DELETE SQL문
DDL_SQL_STMTS	BIGINT	<b>ddl_sql_stmts</b> - DDL(Data Definition Language) SQL문
INT_AUTO_REBINDS	BIGINT	<b>int_auto_rebinds</b> - 내부 자동 리바인드 수
INT_ROWS_DELETED	BIGINT	<b>int_rows_deleted</b> - 삭제된 내부 행 수
INT_ROWS_INSERTED	BIGINT	<b>int_rows_inserted</b> - 삽입된 내부 행 수
INT_ROWS_UPDATED	BIGINT	<b>int_rows_updated</b> - 갱신된 내부 행 수
INT_COMMITS	BIGINT	<b>int_commits</b> - 내부 커밋 수
INT_ROLLBACKS	BIGINT	<b>int_rollback</b> - 내부 롤백 수
INT_DEADLOCK_ROLLBACKS	BIGINT	<b>int_deadlock_rollback</b> - 교착 상태로 인한 내부 롤백 수
ROWS_DELETED	BIGINT	<b>rows_deleted</b> - 삭제된 행 수
ROWS_INSERTED	BIGINT	<b>rows_inserted</b> - 삽입된 행 수
ROWS_UPDATED	BIGINT	<b>rows_updated</b> - 갱신된 행 수
ROWS_SELECTED	BIGINT	<b>rows_selected</b> - 선택된 행 수
ROWS_READ	BIGINT	<b>rows_read</b> - 읽은 행 수
BINDS_PRECOMPILES	BIGINT	<b>binds_precompiles</b> - 시도된 바인드/프리컴파일 수
TOTAL_LOG_AVAILABLE	BIGINT	<b>total_log_available</b> - 사용 가능한 전체 로그
TOTAL_LOG_USED	BIGINT	<b>total_log_used</b> - 사용된 전체 로그 스페이스
SEC_LOG_USED_TOP	BIGINT	<b>sec_log_used_top</b> - 사용된 최대 2차 로그 스페이스

표 256. SNAP\_GET\_DB 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TOT_LOG_USED_TOP	BIGINT	<b>tot_log_used_top</b> - 사용된 최대 전체 로그 스페이스
SEC_LOGS_ALLOCATED	BIGINT	<b>sec_logs_allocated</b> - 현재 할당된 2차 로그
LOG_READS	BIGINT	<b>log_reads</b> - 읽은 로그 페이지 수
LOG_READ_TIME_S	BIGINT	<b>log_read_time</b> - 로그 읽기 시간
LOG_READ_TIME_NS	BIGINT	<b>log_read_time</b> - 로그 읽기 시간
LOG_WRITES	BIGINT	<b>log_writes</b> - 쓴 로그 페이지 수
LOG_WRITE_TIME_S	BIGINT	<b>log_write_time</b> - 로그 쓰기 시간
LOG_WRITE_TIME_NS	BIGINT	<b>log_write_time</b> - 로그 쓰기 시간
NUM_LOG_WRITE_IO	BIGINT	<b>num_log_write_io</b> - 로그 쓰기 수
NUM_LOG_READ_IO	BIGINT	<b>num_log_read_io</b> - 로그 읽기 수
NUM_LOG_PART_PAGE_IO	BIGINT	<b>num_log_part_page_io</b> - 부분 로그 페이지 쓰기 수
NUM_LOG_BUFFER_FULL	BIGINT	<b>num_log_buffer_full</b> - 전체 로그 버퍼 수
NUM_LOG_DATA_FOUND_IN_BUFFER	BIGINT	<b>num_log_data_found_in_buffer</b> - 버퍼에 있는 로그 데이터 수
APPL_ID_OLDEST_XACT	BIGINT	<b>appl_id_oldest_xact</b> - 가장 오래된 트랜잭션이 포함된 응용프로그램
LOG_TO_REDO_FOR_RECOVERY	BIGINT	<b>log_to_redo_for_recovery</b> - 복구를 위해 재실행할 로그 수량
LOG_HELD_BY_DIRTY_PAGES	BIGINT	<b>log_held_by_dirty_pages</b> - 더티 페이지 별로 계산되는 로그 스페이스 수량
PKG_CACHE_LOOKUPS	BIGINT	<b>pkg_cache_lookups</b> - 패키지 캐시 찾아보기 수
PKG_CACHE_INSERTS	BIGINT	<b>pkg_cache_inserts</b> - 패키지 캐시 삽입 수
PKG_CACHE_NUM_OVERFLOW	BIGINT	<b>pkg_cache_num_overflows</b> - 패키지 캐시 오버플로우 수
PKG_CACHE_SIZE_TOP	BIGINT	<b>pkg_cache_size_top</b> - 패키지 캐시 상위 워터 마크(water mark)
APPL_SECTION_LOOKUPS	BIGINT	<b>appl_section_lookups</b> - 섹션 찾아보기 수
APPL_SECTION_INSERTS	BIGINT	<b>appl_section_inserts</b> - 섹션 삽입 수
CAT_CACHE_LOOKUPS	BIGINT	<b>cat_cache_lookups</b> - 카탈로그 캐시 찾아보기 수
CAT_CACHE_INSERTS	BIGINT	<b>cat_cache_inserts</b> - 카탈로그 캐시 삽입 수
CAT_CACHE_OVERFLOW	BIGINT	<b>cat_cache_overflows</b> - 카탈로그 캐시 오버플로우 수

표 256. SNAP\_GET\_DB 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
CAT_CACHE_SIZE_TOP	BIGINT	<b>cat_cache_size_top</b> - 카탈로그 캐시 상위 워터 마크(water mark)
PRIV_WORKSPACE_SIZE_TOP	BIGINT	<b>priv_workspace_size_top</b> - 개인용 작업 스페이스 최대 크기
PRIV_WORKSPACE_NUM_OVERFLOW	BIGINT	<b>priv_workspace_num_overflows</b> - 개인용 작업 스페이스 오버플로우 수
PRIV_WORKSPACE_SECTION_INSERTS	BIGINT	<b>priv_workspace_section_inserts</b> - 개인용 작업 스페이스 섹션 삽입 수
PRIV_WORKSPACE_SECTION_LOOKUPS	BIGINT	<b>priv_workspace_section_lookups</b> - 개인용 작업 스페이스 섹션 찾아보기 수
SHR_WORKSPACE_SIZE_TOP	BIGINT	<b>shr_workspace_size_top</b> - 공유 작업 스페이스 최대 크기
SHR_WORKSPACE_NUM_OVERFLOW	BIGINT	<b>shr_workspace_num_overflows</b> - 공유 작업 스페이스 오버플로우 수
SHR_WORKSPACE_SECTION_INSERTS	BIGINT	<b>shr_workspace_section_inserts</b> - 공유 작업 스페이스 섹션 삽입 수
SHR_WORKSPACE_SECTION_LOOKUPS	BIGINT	<b>shr_workspace_section_lookups</b> - 공유 작업 스페이스 섹션 찾아보기 수
TOTAL_HASH_JOINS	BIGINT	<b>total_hash_joins</b> - 전체 해시 조인 수
TOTAL_HASH_LOOPS	BIGINT	<b>total_hash_loops</b> - 전체 해시 루프 수
HASH_JOIN_OVERFLOW	BIGINT	<b>hash_join_overflows</b> - 해시 조인 오버플로우 수
HASH_JOIN_SMALL_OVERFLOW	BIGINT	<b>hash_join_small_overflows</b> - 해시 조인 작은 오버플로우 수
NUM_DB_STORAGE_PATHS	BIGINT	<b>num_db_storage_paths</b> - 자동 스토리지 경로 수
DBPARTITIONNUM	SMALLINT	<b>node_number</b> - 노드 번호

## SNAP\_GET\_DBM 테이블 함수 - dbm 논리 그룹화 스냅샷 정보 검색

주: 이 테이블 함수는 사용되지 않으며 572 페이지의 『SNAPDBM 관리 뷰 및 SNAP\_GET\_DBM\_V95 테이블 함수 - dbm 논리적 그룹화 스냅샷 정보 검색』에서 교체됩니다.

SNAP\_GET\_DBM 테이블 함수는 스냅샷 모니터 DB2 데이터베이스 관리 프로그램 (dbm) 논리적 그룹화 정보를 리턴합니다.

SNAP\_GET\_DBM\_MEMORY\_POOL, SNAP\_GET\_FCM, SNAP\_GET\_FCM\_PART 및 SNAP\_GET\_SWITCHES 테이블 함수와 함께 사용되는 SNAP\_GET\_DBM 테이블 함수는 GET SNAPSHOT FOR DBM 명령과 동등한 데이터를 제공합니다.

리턴 가능한 전체 정보 목록은 1082 페이지의 표 257의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_DBM( ( dbpartitionnum ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. 이 입력 옵션을 사용하지 않는 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbpartitionnum*이 NULL로 설정되고 SNAP\_WRITE\_FILE 프로시저에서 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_DBM 테이블 함수가 메모리에서 스냅샷을 호출합니다.

## 권한 부여

- SYSMON 권한
- SNAP\_GET\_DBM 테이블 함수의 EXECUTE 특권

## 예 :

데이터베이스 파티션 번호 2의 시작 시간 및 현재 상태를 검색합니다.

```
SELECT DB2START_TIME, DB2_STATUS FROM TABLE(SNAP_GET_DBM(2)) AS T
```

다음은 이 쿼리의 출력 예입니다.

```
DB2START_TIME          DB2_STATUS
-----
2006-01-06-14.59.59.062798 ACTIVE
```

## 리턴된 정보

표 257. SNAP\_GET\_DBM 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
SORT_HEAP_ALLOCATED	BIGINT	sort_heap_allocated - 할당된 전체 정렬 힙
POST_THRESHOLD_SORTS	BIGINT	post_threshold_sorts - 포스트 임계값 정렬 수
PIPED_SORTS_REQUESTED	BIGINT	piped_sorts_requested - 요청된 파이프 정렬 수
PIPED_SORTS_ACCEPTED	BIGINT	piped_sorts_accepted - 수락된 파이프 정렬 수
REM_CONS_IN	BIGINT	rem_cons_in - 데이터베이스 관리 프로그램에 리모트 연결 수
REM_CONS_IN_EXEC	BIGINT	rem_cons_in_exec - 데이터베이스 관리 프로그램 모니터 요소에서 실행되는 리모트 연결 수
LOCAL_CONS	BIGINT	local_cons - 로컬 연결 수
LOCAL_CONS_IN_EXEC	BIGINT	local_cons_in_exec - 데이터베이스 관리 프로그램 모니터 요소에서 실행되는 로컬 연결 수
CON_LOCAL_DBASES	BIGINT	con_local_dbases - 현재 연결된 로컬 데이터베이스 수
AGENTS_REGISTERED	BIGINT	agents_registered - 등록된 에이전트 수
AGENTS_WAITING_ON_TOKEN	BIGINT	agents_waiting_on_token - 토큰을 기다리는 에이전트 수
DB2_STATUS	VARCHAR(12)	db2_status - DB2 인스턴스 상태 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• ACTIVE</li> <li>• QUIESCE_PEND</li> <li>• QUIESCED</li> </ul>
AGENTS_REGISTERED_TOP	BIGINT	agents_registered_top - 등록된 최대 에이전트 수
AGENTS_WAITING_TOP	BIGINT	agents_waiting_top - 대기 중인 최대 에이전트 수
COMM_PRIVATE_MEM	BIGINT	comm_private_mem - 커밋된 전용 메모리
IDLE_AGENTS	BIGINT	idle_agents - 유휴 에이전트 수
AGENTS_FROM_POOL	BIGINT	agents_from_pool - 풀로부터 할당된 에이전트 수
AGENTS_CREATED_EMPTY_POOL	BIGINT	agents_created_empty_pool - 비어 있는 에이전트 풀로 인해 작성된 에이전트 수

표 257. SNAP\_GET\_DBM 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
COORD_AGENTS_TOP	BIGINT	coord_agents_top - 최대 코디네이팅 에이전트 수
MAX_AGENT_OVERFLOW	BIGINT	max_agent_overflows - 최대 에이전트 오버플로우 수
AGENTS_STOLEN	BIGINT	agents_stolen - 분실 에이전트 수
GW_TOTAL_CONS	BIGINT	gw_total_cons - DB2 Connect에 대한 전체 연결 시도 수
GW_CUR_CONS	BIGINT	gw_cur_cons - DB2 Connect에 대한 현재 연결 수
GW_CONS_WAIT_HOST	BIGINT	gw_cons_wait_host - 호스트 응답을 대기 중인 연결 수
GW_CONS_WAIT_CLIENT	BIGINT	gw_cons_wait_client - 클라이언트의 요청 전송을 대기 중인 연결 수
POST_THRESHOLD_HASH_JOINS	BIGINT	post_threshold_hash_joins - 해시 조인 임계값
NUM_GW_CONN_SWITCHES	BIGINT	num_gw_conn_switches - 연결 전환 수
DB2START_TIME	TIMESTAMP	db2start_time - 데이터베이스 관리 프로그램 시작 시간소인
LAST_RESET	TIMESTAMP	last_reset - 마지막 재설정 시간소인
NUM_NODES_IN_DB2_INSTANCE	INTEGER	num_nodes_in_db2_instance - 데이터베이스 파티션의 노드 수
PRODUCT_NAME	VARCHAR(32)	product_name - 제품 이름
SERVICE_LEVEL	VARCHAR(18)	service_level - 서비스 레벨
SORT_HEAP_TOP	BIGINT	sort_heap_top - 전용 힙 상위 워터 마크 (water mark) 정렬
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAP\_GET\_DB\_V91 테이블 함수 - dbase 논리적 그룹에서 스냅샷 정보 검색

주: 이 테이블 함수는 더 이상 사용되지 않으며 557 페이지의 『SNAPDB 관리 뷰 및 SNAP\_GET\_DB\_V95 테이블 함수 - dbase 논리 그룹에서 스냅샷 정보 검색』에서 교체됩니다.

SNAP\_GET\_DB\_V91 테이블 함수는 데이터베이스(dbase) 논리적 그룹에서 스냅샷 정보를 리턴합니다.

SNAP\_GET\_DB\_MEMORY\_POOL, SNAP\_GET\_DETAILLOG\_V91, SNAP\_GET\_HADR and SNAP\_GET\_STORAGE\_PATHS 테이블 함수와 함께 사

용되는 SNAP\_GET\_DB\_V91 테이블 함수는 GET SNAPSHOT FOR ALL DATABASES CLP 명령과 동등한 정보를 제공합니다.

리턴되는 전체 정보 목록은 1085 페이지의 표 258의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_DB_V91 ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 빈 문자열을 지정하십시오. 현재 연결된 데이터베이스의 동일한 인스턴스 내의 모든 데이터베이스에서 스냅샷을 작성하려면 NULL 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우 -1이 *dbpartitionnum*에 대해 내재적으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_DB\_V91 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대해 스냅샷을 작성합니다.

## 권한 부여

다음 권한 또는 특권 중 하나가 필요합니다.

- SNAP\_GET\_DB\_V91 테이블 함수의 EXECUTE 특권
- DATAACCESS 권한
- SYSMON 권한



- SYSMANT 권한
- SYSCTRL 권한
- SYSADM 권한

예:

예 1: 현재 연결된 데이터베이스의 모든 데이터베이스 파티션에서 집계 뷰로 상태, 플랫폼, 위치 및 연결 시간을 검색합니다.

```
SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,
       DB_LOCATION, DB_CONN_TIME FROM TABLE(SNAP_GET_DB_V91(' ', -2)) AS T
```

다음은 이 쿼리의 출력 예입니다.

```
DB_NAME      DB_STATUS    SERVER_PLATFORM ...
-----
SAMPLE      ACTIVE      AIX64          ...
```

1개의 레코드가 선택되었습니다.

이 쿼리의 출력(계속).

```
... DB_LOCATION DB_CONN_TIME
... -----
... LOCAL      2005-07-24-22.09.22.013196
```

예 2: 현재 연결된 데이터베이스가 포함된 동일한 인스턴스에서 모든 활성 데이터베이스에 대해 모든 데이터베이스 파티션에서 집계 뷰로 상태, 플랫폼, 위치 및 연결 시간을 검색합니다.

```
SELECT SUBSTR(DB_NAME, 1, 20) AS DB_NAME, DB_STATUS, SERVER_PLATFORM,
       DB_LOCATION, DB_CONN_TIME
FROM TABLE(SNAP_GET_DB_V91(CAST (NULL AS VARCHAR(128)), -2)) AS T
```

다음은 이 쿼리의 출력 예입니다.

```
DB_NAME      DB_STATUS    SERVER_PLATFORM ...
-----
TOOLSDB     ACTIVE      AIX64          ...
SAMPLE     ACTIVE      AIX64          ...
```

이 쿼리의 출력(계속).

```
... DB_LOCATION DB_CONN_TIME
... -----
... LOCAL      2005-07-24-22.26.54.396335
... LOCAL      2005-07-24-22.09.22.013196
```

## SNAP\_GET\_DB\_V91 테이블 함수 메타데이터

표 258. SNAP\_GET\_DB\_V91 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.

표 258. SNAP\_GET\_DB\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
DB_PATH	VARCHAR(1024)	db_path - 데이터베이스 경로
INPUT_DB_ALIAS	VARCHAR (128)	input_db_alias - 데이터베이스 별명 입력
DB_STATUS	VARCHAR(12)	db_status - 데이터베이스 상태. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• ACTIVE</li> <li>• QUIESCE_PEND</li> <li>• QUIESCED</li> <li>• ROLLFWD</li> </ul>
CATALOG_PARTITION	SMALLINT	catalog_node - 카탈로그 노드 번호
CATALOG_PARTITION_NAME	VARCHAR (128)	catalog_node_name - 카탈로그 노드 네트워크 이름

표 258. SNAP\_GET\_DB\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SERVER_PLATFORM	VARCHAR(12)	<p>server_platform - 서버 운영 체제. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다.</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> <li>• WINDOWS95</li> </ul>

표 258. SNAP\_GET\_DB\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DB_LOCATION	VARCHAR(12)	db_location - 데이터베이스 위치. 이 인터페이스는 sqlmon.h의 정의를 기반으로 하는 텍스트 ID를 리턴하며 다음 중 하나입니다. • LOCAL • REMOTE
DB_CONN_TIME	TIMESTAMP	db_conn_time - 데이터베이스 활성화 시간소인
LAST_RESET	TIMESTAMP	last_reset - 마지막 재설정 시간소인
LAST_BACKUP	TIMESTAMP	last_backup - 최종 백업 시간소인
CONNECTIONS_TOP	BIGINT	connections_top - 최대 동시 연결 수
TOTAL_CONS	BIGINT	total_cons - 데이터베이스 활성화 이후 연결 수
TOTAL_SEC_CONS	BIGINT	total_sec_cons - 2차 연결 수
APPLS_CUR_CONS	BIGINT	appls_cur_cons - 현재 연결된 응용프로그램 수
APPLS_IN_DB2	BIGINT	appls_in_db2 - 현재 데이터베이스에서 실행 중인 응용프로그램 수
NUM_ASSOC_AGENTS	BIGINT	num_assoc_agents - 연관 에이전트 수
AGENTS_TOP	BIGINT	agents_top - 작성된 에이전트 수
COORD_AGENTS_TOP	BIGINT	coord_agents_top - 최대 코디네이팅 에이전트 수
LOCKS_HELD	BIGINT	locks_held - 보유된 잠금 수
LOCK_WAITS	BIGINT	lock_waits - 잠금 대기 수
LOCK_WAIT_TIME	BIGINT	lock_wait_time - 잠금 대기 시간
LOCK_LIST_IN_USE	BIGINT	lock_list_in_use - 사용 중인 전체 잠금 목록 메모리
DEADLOCKS	BIGINT	deadlocks - 발견된 교착 상태 수
LOCK_ESCALS	BIGINT	lock_escals - 잠금 에스컬레이션 수
X_LOCK_ESCALS	BIGINT	x_lock_escals - 배타적 잠금 에스컬레이션 수
LOCKS_WAITING	BIGINT	locks_waiting - 잠금 대기 중인 현재 에이전트 수
LOCK_TIMEOUTS	BIGINT	lock_timeouts - 잠금 시간종료 수
NUM_INDOUBT_TRANS	BIGINT	num_indoubt_trans - 인다우트(Indoubt) 트랜잭션 수
SORT_HEAP_ALLOCATED	BIGINT	sort_heap_allocated - 할당된 전체 정렬 힙
SORT_SHRHEAP_ALLOCATED	BIGINT	sort_shrheap_allocated - 현재 할당된 정렬 공유 힙
SORT_SHRHEAP_TOP	BIGINT	sort_shrheap_top - 공유 힙 상위 워터 마크(water mark) 정렬

표 258. SNAP\_GET\_DB\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POST_SHRTHRESHOLD_SORTS	BIGINT	post_shrthreshold_sorts - 포스트 공유 임계값 정렬 수
TOTAL_SORTS	BIGINT	total_sorts - 전체 정렬 수
TOTAL_SORT_TIME	BIGINT	total_sort_time - 총 정렬 시간
SORT_OVERFLOWS	BIGINT	sort_overflows - 정렬 오버플로우 수
ACTIVE_SORTS	BIGINT	active_sorts - 활성 정렬 수
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_WRITES	BIGINT	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_ASYNC_XDA_READS	BIGINT	pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수

표 258. SNAP\_GET\_DB\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_ASYNC_XDA_WRITES	BIGINT	pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - 버퍼 풀 비동기 읽기 시간
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - 버퍼 풀 비동기 쓰기 시간
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 수
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수
POOL_ASYNC_XDA_READ_REQS	BIGINT	pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - 버퍼 풀 희생 버퍼 없음
POOL_LSN_GAP_CLNS	BIGINT	pool_lsn_gap_clns - 버퍼 풀 로그 스페이스 클리너 트리거
POOL_DRTY_PG_STEAL_CLNS	BIGINT	pool_drty_pg_steal_clns - 버퍼 풀 희생 (victim) 페이지 클리너 트리거
POOL_DRTY_PG_THRSH_CLNS	BIGINT	pool_drty_pg_thrsh_clns - 버퍼 풀 임계값 클리너 트리거
PREFETCH_WAIT_TIME	BIGINT	prefetch_wait_time - 프리페치 대기 시간
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리페치 페이지 수
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
FILES_CLOSED	BIGINT	files_closed - 닫힌 데이터베이스 파일 수

표 258. SNAP\_GET\_DB\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
ELAPSED_EXEC_TIME_S	BIGINT	elapsed_exec_time - 명령문 실행 경과 시간
ELAPSED_EXEC_TIME_MS	BIGINT	elapsed_exec_time - 명령문 실행 경과 시간
COMMIT_SQL_STMTS	BIGINT	commit_sql_stmts - 시도된 명령문 커미트
ROLLBACK_SQL_STMTS	BIGINT	rollback_sql_stmts - 시도된 롤백 명령문
DYNAMIC_SQL_STMTS	BIGINT	dynamic_sql_stmts - 시도된 동적 SQL문
STATIC_SQL_STMTS	BIGINT	static_sql_stmts - 시도된 정적 SQL문
FAILED_SQL_STMTS	BIGINT	failed_sql_stmts - 실패한 명령문 조각
SELECT_SQL_STMTS	BIGINT	select_sql_stmts - 실행된 Select SQL문
UID_SQL_STMTS	BIGINT	uid_sql_stmts - 실행된 UPDATE/INSERT/DELETE SQL문
DDL_SQL_STMTS	BIGINT	ddl_sql_stmts - DDL(Data Definition Language) SQL문
INT_AUTO_REBINDS	BIGINT	int_auto_rebinds - 내부 자동 리바인드 수
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 삭제된 내부 행 수
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 삽입된 내부 행 수
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 갱신된 내부 행 수
INT_COMMITS	BIGINT	int_commits - 내부 커미트 수
INT_ROLLBACKS	BIGINT	int_rollback - 내부 롤백 수
INT_DEADLOCK_ROLLBACKS	BIGINT	int_deadlock_rollback - 교착 상태로 인한 내부 롤백 수
ROWS_DELETED	BIGINT	rows_deleted - 삭제된 행 수
ROWS_INSERTED	BIGINT	rows_inserted - 삽입된 행 수
ROWS_UPDATED	BIGINT	rows_updated - 갱신된 행 수
ROWS_SELECTED	BIGINT	rows_selected - 선택된 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
BINDS_PRECOMPILES	BIGINT	binds_precompiles - 시도된 바인드/프리컴파일 수
TOTAL_LOG_AVAILABLE	BIGINT	total_log_available - 사용 가능한 전체 로그
TOTAL_LOG_USED	BIGINT	total_log_used - 사용된 전체 로그 스페이스
SEC_LOG_USED_TOP	BIGINT	sec_log_used_top - 사용된 최대 2차 로그 스페이스
TOT_LOG_USED_TOP	BIGINT	tot_log_used_top - 사용된 최대 전체 로그 스페이스
SEC_LOGS_ALLOCATED	BIGINT	sec_logs_allocated - 현재 할당된 2차 로그
LOG_READS	BIGINT	log_reads - 읽은 로그 페이지 수

표 258. SNAP\_GET\_DB\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
LOG_READ_TIME_S	BIGINT	log_read_time - 로그 읽기 시간
LOG_READ_TIME_NS	BIGINT	log_read_time - 로그 읽기 시간
LOG_WRITES	BIGINT	log_writes - 쓴 로그 페이지 수
LOG_WRITE_TIME_S	BIGINT	log_write_time - 로그 쓰기 시간
LOG_WRITE_TIME_NS	BIGINT	log_write_time - 로그 쓰기 시간
NUM_LOG_WRITE_IO	BIGINT	num_log_write_io - 로그 쓰기 수
NUM_LOG_READ_IO	BIGINT	num_log_read_io - 로그 읽기 수
NUM_LOG_PART_PAGE_IO	BIGINT	num_log_part_page_io - 부분 로그 페이지 쓰기 수
NUM_LOG_BUFFER_FULL	BIGINT	num_log_buffer_full - 전체 로그 버퍼 수
NUM_LOG_DATA_FOUND_IN_BUFFER	BIGINT	num_log_data_found_in_buffer - 버퍼에 있는 로그 데이터 수
APPL_ID_OLDEST_XACT	BIGINT	appl_id_oldest_xact - 가장 오래된 트랜잭션이 포함된 응용프로그램
LOG_TO_REDO_FOR_RECOVERY	BIGINT	log_to_redo_for_recovery - 복구를 위해 재실행되는 로그 수량
LOG_HELD_BY_DIRTY_PAGES	BIGINT	log_held_by_dirty_pages - 더티 페이지 별로 계산되는 로그 스페이스 수량
PKG_CACHE_LOOKUPS	BIGINT	pkg_cache_lookups - 패키지 캐시 찾아보기 수
PKG_CACHE_INSERTS	BIGINT	pkg_cache_inserts - 패키지 캐시 삽입 수
PKG_CACHE_NUM_OVERFLOW	BIGINT	pkg_cache_num_overflows - 패키지 캐시 오버플로우 수
PKG_CACHE_SIZE_TOP	BIGINT	pkg_cache_size_top - 패키지 캐시 상위 워터 마크(water mark)
APPL_SECTION_LOOKUPS	BIGINT	appl_section_lookups - 섹션 찾아보기 수
APPL_SECTION_INSERTS	BIGINT	appl_section_inserts - 섹션 삽입 수
CAT_CACHE_LOOKUPS	BIGINT	cat_cache_lookups - 카탈로그 캐시 찾아보기 수
CAT_CACHE_INSERTS	BIGINT	cat_cache_inserts - 카탈로그 캐시 삽입 수
CAT_CACHE_OVERFLOW	BIGINT	cat_cache_overflows - 카탈로그 캐시 오버플로우 수
CAT_CACHE_SIZE_TOP	BIGINT	cat_cache_size_top - 카탈로그 캐시 상위 워터 마크(water mark)
PRIV_WORKSPACE_SIZE_TOP	BIGINT	priv_workspace_size_top - 개인용 작업 스페이스 최대 크기
PRIV_WORKSPACE_NUM_OVERFLOW	BIGINT	priv_workspace_num_overflows - 개인용 작업 스페이스 오버플로우 수



표 258. SNAP\_GET\_DB\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
PRIV_WORKSPACE_SECTION_INSERTS	BIGINT	priv_workspace_section_inserts - 개인용 작업 스페이스 섹션 삽입 수
PRIV_WORKSPACE_SECTION_LOOKUPS	BIGINT	priv_workspace_section_lookups - 개인용 작업 스페이스 섹션 찾아보기 수
SHR_WORKSPACE_SIZE_TOP	BIGINT	shr_workspace_size_top - 공유 작업 스페이스 최대 크기
SHR_WORKSPACE_NUM_OVERFLOWS	BIGINT	shr_workspace_num_overflows - 공유 작업 스페이스 오버플로우 수
SHR_WORKSPACE_SECTION_INSERTS	BIGINT	shr_workspace_section_inserts - 공유 작업 스페이스 섹션 삽입 수
SHR_WORKSPACE_SECTION_LOOKUPS	BIGINT	shr_workspace_section_lookups - 공유 작업 스페이스 섹션 찾아보기 수
TOTAL_HASH_JOINS	BIGINT	total_hash_joins - 전체 해시 조인 수
TOTAL_HASH_LOOPS	BIGINT	total_hash_loops - 전체 해시 루프 수
HASH_JOIN_OVERFLOWS	BIGINT	hash_join_overflows - 해시 조인 오버플로우 수
HASH_JOIN_SMALL_OVERFLOWS	BIGINT	hash_join_small_overflows - 해시 조인 작은 오버플로우 수
POST_SHRTHRESHOLD_HASH_JOINS	BIGINT	post_shrthreshold_hash_joins - 포스트 임계값 해시 조인 수
ACTIVE_HASH_JOINS	BIGINT	active_hash_joins - 활성 해시 조인 수
NUM_DB_STORAGE_PATHS	BIGINT	num_db_storage_paths - 자동 스토리지 경로 수
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.
SMALLEST_LOG_AVAIL_NODE	INTEGER	smallest_log_avail_node - 최소 사용 가능 로그 스페이스가 포함된 노드

## SNAP\_GET\_DYN\_SQL\_V91 테이블 함수 - dynsql 논리적 그룹 스냅샷 정보 검색

주: 이 테이블 함수는 더 이상 사용되지 않으며 584 페이지의 『SNAPDYN\_SQL 관리 뷰 및 SNAP\_GET\_DYN\_SQL\_V95 테이블 함수 - dynsql 논리 그룹 스냅샷 정보 검색』에서 교체됩니다.

SNAP\_GET\_DYN\_SQL\_V91 테이블 함수는 dynsql 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.

이 테이블 함수는 GET SNAPSHOT FOR DYNAMIC SQL ON database-alias CLP 명령과 동등한 정보를 리턴합니다.

리턴 가능한 전체 정보 목록은 1095 페이지의 표 259의 내용을 참조하십시오.

## 구문

```
▶▶ SNAP_GET_DYN_SQL_V91 ( ( dbname [ , dbpartitionnum ] ) ) ▶▶
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *dbname*

현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(128)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 NULL 또는 비어 있는 문자열을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 선택적 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 활성 데이터베이스 파티션의 집계에는 -2를 지정하십시오. *dbname*이 NULL로 설정되지 않고 *dbpartitionnum*이 NULL로 설정된 경우 -1이 *dbpartitionnum*에 대해 내재적으로 설정됩니다. 이 입력 옵션을 사용하지 않으면 즉, *dbname*만 제공한 경우 모든 활성 데이터베이스 파티션에서 데이터가 리턴됩니다. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

*dbname* 및 *dbpartitionnum* 모두 NULL로 설정된 경우 SNAP\_WRITE\_FILE 프로시저로 작성된 파일에서 데이터 읽기가 시도됩니다. 이 파일은 이미 작성되었으므로 최신 데이터가 아닐 수 있습니다. 해당 스냅샷 API 요청 유형이 포함된 파일이 없는 경우 SNAP\_GET\_DYN\_SQL\_V91 테이블 함수는 현재 연결된 데이터베이스 및 데이터베이스 파티션 번호에 대해 스냅샷을 작성합니다.

## 권한 부여

- SYSMON 권한
- SNAP\_GET\_DYN\_SQL\_V91 테이블 함수의 EXECUTE 특권

예 :

현재 연결된 데이터베이스의 현재 연결된 데이터베이스 파티션에서 읽은 행 수 별로 동적 SQL 실행 목록을 검색합니다.

```
SELECT PREP_TIME_WORST, NUM_COMPILATIONS, SUBSTR(STMT_TEXT, 1, 60)
       AS STMT_TEXT FROM TABLE(SNAP_GET_DYN_SQL_V91(' ', -1)) as T
ORDER BY ROWS_READ
```

다음은 이 쿼리의 출력 예입니다.

```
PREP_TIME_WORST      ...
-----
0 ...
3 ...
4 ...
4 ...
4 ...
4 ...
3 ...
4 ...
...
```

이 쿼리의 출력(계속).

```
... 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,
```

### 리턴된 정보

표 259. SNAP\_GET\_DYN\_SQL\_V91 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
NUM_EXECUTIONS	BIGINT	num_executions - 명령문 실행
NUM_COMPILATIONS	BIGINT	num_compilations - 명령문 컴파일
PREP_TIME_WORST	BIGINT	prep_time_worst - 명령문의 가장 안좋은 준비 시간

표 259. SNAP\_GET\_DYN\_SQL\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
PREP_TIME_BEST	BIGINT	prep_time_best - 명령문의 가장 좋은 준비 시간
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 삭제된 내부 행 수
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 삽입된 내부 행 수
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 갱신된 내부 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
STMT_SORTS	BIGINT	stmt_sorts - 명령문 정렬 수
SORT_OVERFLOW	BIGINT	sort_overflows - 정렬 오버플로우 수
TOTAL_SORT_TIME	BIGINT	total_sort_time - 총 정렬 시간
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_XDA_L_READS	BIGINT	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_XDA_P_READS	BIGINT	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_TEMP_XDA_L_READS	BIGINT	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_P_READS	BIGINT	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 모니터 요소
TOTAL_EXEC_TIME	BIGINT	total_exec_time - 경과된 명령문 실행 시간
TOTAL_EXEC_TIME_MS	BIGINT	total_exec_time - 경과된 명령문 실행 시간
TOTAL_USR_CPU_TIME	BIGINT	total_usr_cpu_time - 명령문에 대한 전체 사용자 CPU
TOTAL_USR_CPU_TIME_MS	BIGINT	total_usr_cpu_time - 명령문에 대한 전체 사용자 CPU

표 259. SNAP\_GET\_DYN\_SQL\_V91 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TOTAL_SYS_CPU_TIME	BIGINT	total_sys_cpu_time - 명령문에 대한 전체 시스템 CPU
TOTAL_SYS_CPU_TIME_MS	BIGINT	total_sys_cpu_time - 명령문에 대한 전체 시스템 CPU
STMT_TEXT	CLOB(2 M)	stmt_text - SQL문 텍스트
DBPARTITIONNUM	SMALLINT	이 행에 대해 데이터가 검색된 데이터베이스 파티션.

## SNAP\_GET\_DYN\_SQL

주: 이 테이블 함수는 사용되지 않으며 1093 페이지의 『SNAP\_GET\_DYN\_SQL\_V91 테이블 함수 - dynsql 논리적 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAP\_GET\_DYN\_SQL—(—dbname—, —dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

SNAP\_GET\_DYN\_SQL 테이블 함수는 dynsql 논리 데이터 그룹으로부터의 스냅샷 정보를 리턴합니다.

### *dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 260. SNAP\_GET\_DYN\_SQL 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
NUM_EXECUTIONS	BIGINT	num_executions - 명령문 실행
NUM_COMPILATIONS	BIGINT	num_compilations - 명령문 컴파일
PREP_TIME_WORST	BIGINT	prep_time_worst - 명령문의 가장 안좋은 준비 시간
PREP_TIME_BEST	BIGINT	prep_time_best - 명령문의 가장 좋은 준비 시간
INT_ROWS_DELETED	BIGINT	int_rows_deleted - 삭제된 내부 행 수
INT_ROWS_INSERTED	BIGINT	int_rows_inserted - 삽입된 내부 행 수
INT_ROWS_UPDATED	BIGINT	int_rows_updated - 갱신된 내부 행 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
STMT_SORTS	BIGINT	stmt_sorts - 명령문 정렬 수
SORT_OVERFLOWS	BIGINT	sort_overflows - 정렬 오버플로우 수
TOTAL_SORT_TIME	BIGINT	total_sort_time - 총 정렬 시간
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
TOTAL_EXEC_TIME	BIGINT	total_exec_time - 경과된 명령문 실행 시간
TOTAL_EXEC_TIME_MS	BIGINT	total_exec_time - 경과된 명령문 실행 시간
TOTAL_USR_TIME	BIGINT	total_usr_cpu_time - 명령문에 대한 전체 사용자 CPU
TOTAL_USR_TIME_MS	BIGINT	total_usr_cpu_time - 명령문에 대한 전체 사용자 CPU
TOTAL_SYS_TIME	BIGINT	total_sys_cpu_time - 명령문에 대한 전체 시스템 CPU

표 260. SNAP\_GET\_DYN\_SQL 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TOTAL_SYS_TIME_MS	BIGINT	total_sys_cpu_time - 명령문에 대한 전체 시스템 CPU
STMT_TEXT	CLOB	stmt_text - SQL문 텍스트

## SNAP\_GET\_STO\_PATHS

주: 이 테이블 함수는 사용되지 않으며 767 페이지의 『SNAPSTORAGE\_PATHS 관리 뷰 및 SNAP\_GET\_STORAGE\_PATHS 테이블 함수 - 자동 스토리지 경로 정보 검색』으로 교체됩니다.

▶▶—SNAP\_GET\_STO\_PATHS—(—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

SNAP\_GET\_STO\_PATHS 테이블 함수는 storage\_paths 논리 데이터 그룹으로부터의 스냅샷 정보를 리턴합니다.

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 NULL 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 261. SNAP\_GET\_STO\_PATHS 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.

표 261. SNAP\_GET\_STO\_PATHS 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
DB_STORAGE_PATH	VARCHAR(256)	db_storage_path - 자동 스토리지 경로

## SNAP\_GET\_TAB

주: 이 테이블 함수는 사용되지 않으며 629 페이지의 『SNAPTAB 관리 뷰 및 SNAP\_GET\_TAB\_V91 테이블 함수 - 테이블 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶ SNAP\_GET\_TAB(—dbname—, —dbpartitionnum—) ◀◀

스키마는 SYSPROC입니다.

SNAP\_GET\_TAB 테이블 함수는 table 논리 데이터 그룹으로부터의 스냅샷 정보를 리턴합니다.

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 NULL 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 262. SNAP\_GET\_TAB 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.



표 262. SNAP\_GET\_TAB 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TABSCHEMA	VARCHAR (128)	table_schema - 테이블 스키마 이름
TABNAME	VARCHAR (128)	table_name - 테이블 이름
TAB_FILE_ID	BIGINT	table_file_id - 테이블 파일 ID
TAB_TYPE	BIGINT	table_type - 테이블 유형
DATA_OBJECT_PAGES	BIGINT	data_object_pages - 데이터 오브젝트 페이지 수
INDEX_OBJECT_PAGES	BIGINT	index_object_pages - 인덱스 오브젝트 페이지 수
LOB_OBJECT_PAGES	BIGINT	lob_object_pages - LOB 오브젝트 페이지 수
LONG_OBJECT_PAGES	BIGINT	long_object_pages - Long 오브젝트 페이지 수
ROWS_READ	BIGINT	rows_read - 읽은 행 수
ROWS_WRITTEN	BIGINT	rows_written - 기록된 행 수
OVERFLOW_ACCESSES	BIGINT	overflow_accesses - 오버플로우된 레코드에 액세스
PAGE_REORGS	BIGINT	page_reorgs - 페이지 재구성
DBPARTITIONNUM	SMALLINT	node_number - 노드 번호

## SNAP\_GET\_TBSP

주: 이 테이블 함수는 더 이상 사용되지 않으며 638 페이지의 『SNAPTbsp 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수 - 테이블 스페이스 논리 데이터 그룹 스냅샷 정보 검색』에서 교체됩니다.

▶▶ SNAP\_GET\_TBSP(—dbname—, —dbpartitionnum—) ◀◀

스키마는 SYSPROC입니다.

SNAP\_GET\_TBSP 테이블 함수는 테이블 스페이스 논리 데이터 그룹에서 스냅샷 정보를 리턴합니다.

### dbname

이 함수 호출 시에 현재 연결된 데이터베이스와 동일한 인스턴스에 유효한 데이터베이스 이름을 지정하는 유형 VARCHAR(255)의 입력 인수입니다. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 작성하려면 널(NULL) 값을 지정하십시오.

*dbpartitionnum*

유효한 데이터베이스 파티션 번호를 지정하는 INTEGER 유형의 입력 인수입니다. 현재 데이터베이스 파티션에는 -1을 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

두 매개변수를 모두 NULL로 설정하면 해당 스냅샷 API 요청 유형에 대해 SNAPSHOT\_FILEW 스토어드 프로시저가 파일을 이전에 작성하지 않은 경우에만 스냅샷이 작성됩니다.

함수는 아래에 표시된 대로 테이블을 리턴합니다.

표 263. SNAP\_GET\_TBSP 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷이 작성된 날짜 및 시간입니다.
TBSP_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
TBSP_TYPE	SMALLINT	tablespace_type - 테이블 스페이스 유형
TBSP_CONTENT_TYPE	SMALLINT	tablespace_content_type - 테이블 스페이스 콘텐츠 유형
TBSP_PAGE_SIZE	BIGINT	tablespace_page_size - 테이블 스페이스 페이지 크기
TBSP_EXTENT_SIZE	BIGINT	tablespace_extent_size - 테이블 스페이스 Extent 크기
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 테이블 스페이스 프리페치 크기
TBSP_CUR_POOL_ID	BIGINT	tablespace_cur_pool_id - 현재 사용 중인 버퍼 풀
TBSP_NEXT_POOL_ID	BIGINT	tablespace_next_pool_id - 다음 시작 시에 사용되는 버퍼 풀
FS_CACHING <sup>1</sup>	SMALLINT	fs_caching - 파일 시스템 캐싱
POOL_DATA_L_READS	BIGINT	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_TEMP_DATA_L_READS	BIGINT	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_DATA_P_READS	BIGINT	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_ASYNC_DATA_READS	BIGINT	pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 수
POOL_DATA_WRITES	BIGINT	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_ASYNC_DATA_WRITES	BIGINT	pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수

표 263. SNAP\_GET\_TBSP 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_INDEX_L_READS	BIGINT	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_L_READS	BIGINT	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_TEMP_INDEX_P_READS	BIGINT	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_ASYNC_INDEX_READS	BIGINT	pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 수
POOL_INDEX_WRITES	BIGINT	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
POOL_ASYNC_INDEX_WRITES	BIGINT	pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_READ_TIME	BIGINT	pool_read_time - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	pool_write_time - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_READ_TIME	BIGINT	pool_async_read_time - 버퍼 풀 비동기 읽기 시간
POOL_ASYNC_WRITE_TIME	BIGINT	pool_async_write_time - 버퍼 풀 비동기 쓰기 시간
POOL_ASYNC_DATA_READ_REQS	BIGINT	pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 수
POOL_ASYNC_INDEX_READ_REQS	BIGINT	pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수
POOL_NO_VICTIM_BUFFER	BIGINT	pool_no_victim_buffer - 버퍼 풀 희생 버퍼 없음
DIRECT_READS	BIGINT	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	direct_writes - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	direct_read_reqs - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	direct_write_reqs - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	direct_read_time - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	direct_write_time - 직접 쓰기 시간
FILES_CLOSED	BIGINT	files_closed - 닫힌 데이터베이스 파일 수
UNREAD_PREFETCH_PAGES	BIGINT	unread_prefetch_pages - 읽지 않은 프리페치 페이지 수
POOL_DATA_TO_ESTORE	BIGINT	pool_data_to_estore ESTORE 모니터 요소는 중단됩니다. NULL 값이 중지된 모니터 요소에 대해 리턴됩니다.

표 263. SNAP\_GET\_TBSP 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_INDEX_TO_ESTORE	BIGINT	pool_index_to_estore ESTORE 모니터 요소는 중단됩니다. NULL 값이 중지된 모니터 요소에 대해 리턴됩니다.
POOL_INDEX_FROM_ESTORE	BIGINT	pool_index_from_estore ESTORE 모니터 요소는 중단됩니다. NULL 값이 중지된 모니터 요소에 대해 리턴됩니다.
POOL_DATA_FROM_ESTORE	BIGINT	pool_data_from_estore ESTORE 모니터 요소는 중단됩니다. NULL 값이 중지된 모니터 요소에 대해 리턴됩니다.
TBSP_REBALANCER_MODE	BIGINT	tablespace_rebalancer_mode - 재조정 프로그램 모드
TBSP_USING_AUTO_STORAGE	SMALLINT	tablespace_using_auto_storage - 테이블 공간을 자동 스토리지에 사용 가능
TBSP_AUTO_RESIZE_ENABLED	SMALLINT	tablespace_auto_resize_enabled - 테이블 공간 자동 크기 조정 가능

<sup>1</sup> FS\_CACHING이 0인 경우 파일 시스템 캐싱이 사용되고 FS\_CACHING이 1인 경우 파일 시스템 캐싱이 사용되지 않습니다.

## SNAP\_GET\_TBSP\_PART

주: 이 테이블 함수는 사용되지 않으며 794 페이지의 『SNAPTBSP\_PART 관리 뷰 및 SNAP\_GET\_TBSP\_PART\_V91 테이블 함수 - tablespace\_nodeinfo 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAP\_GET\_TBSP\_PART—(—dbname—, —dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

SNAP\_GET\_TBSP\_PART 테이블 함수는 tablespace\_nodeinfo 논리 데이터 그룹으로부터의 스냅샷 정보를 리턴합니다.

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령에서 리턴된 대로 "간접" 또는 "홈"인 디렉토리 항목 유형이 포함된 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하십시오. 널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수를 모두 NULL로 설정하면 해당 스냅샷 API 요청 유형에 대해 SNAPSHOT\_FILEW 스토어드 프로시저가 파일을 이전에 작성하지 않은 경우에만 스냅샷이 작성됩니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 264. SNAP\_GET\_TBSP\_PART 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TBSP_NAME	VARCHAR (128)	tablespace_name - 테이블 스페이스 이름
TBSP_ID	BIGINT	tablespace_id - 테이블 스페이스 ID
TBSP_STATE	BIGINT	tablespace_state - 테이블 스페이스 상태
TBSP_PREFETCH_SIZE	BIGINT	tablespace_prefetch_size - 테이블 스페이스 프리 페치 크기
TBSP_NUM QUIESCERS	BIGINT	tablespace_num_quiescers - quiescer 수
TBSP_STATE_CHANGE_OBJECT_ID	BIGINT	tablespace_state_change_object_id - 상태 변경 오브젝트 ID
TBSP_STATE_CHANGE_TBSP_ID	BIGINT	tablespace_state_change_ts_id - 상태 변경 테이블 스페이스 ID
TBSP_MIN_RECOVERY_TIME	TIMESTAMP	tablespace_min_recovery_time - 롤 포워드를 위한 최소 복구 시간
TBSP_TOTAL_PAGES	BIGINT	tablespace_total_pages - 테이블 스페이스의 전체 페이지
TBSP_USABLE_PAGES	BIGINT	tablespace_usable_pages - 테이블 스페이스에서 사용할 수 있는 페이지
TBSP_USED_PAGES	BIGINT	tablespace_used_pages - 테이블 스페이스에서 사용된 페이지
TBSP_FREE_PAGES	BIGINT	tablespace_free_pages - 테이블 스페이스에서 사용 가능한 페이지
TBSP_PENDING_FREE_PAGES	BIGINT	tablespace_pending_free_pages - 테이블 스페이스에서 보류 중인 사용 가능한 페이지
TBSP_PAGE_TOP	BIGINT	tablespace_page_top - 테이블 스페이스 상위 워터 마크(water mark)
REBALANCER_MODE	BIGINT	tablespace_rebalancer_mode - 재조정 프로그램 모드
REBALANCER_EXTENTS_REMAINING	BIGINT	tablespace_rebalancer_extents_remaining - 재조정 프로그램에서 처리되는 전체 Extent 수
REBALANCER_EXTENTS_PROCESSED	BIGINT	tablespace_rebalancer_extents_processed - 재조정 프로그램이 처리한 Extent 수

표 264. SNAP\_GET\_TBSP\_PART 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
REBALANCER_PRIORITY	BIGINT	tablespace_rebalancer_priority - 현재 재조정 프로그램 우선순위
REBALANCER_START_TIME	TIMESTAMP	tablespace_rebalancer_start_time - 재조정 프로그램 시작 시간
REBALANCER_RESTART_TIME	TIMESTAMP	tablespace_rebalancer_restart_time - 재조정 프로그램 재시작 시간
REBALANCER_LAST_EXTENT_MOVED	BIGINT	tablespace_rebalancer_last_extent_moved - 재조정 프로그램이 이동한 마지막 Extent
TBSP_NUM_RANGES	BIGINT	tablespace_num_ranges - 테이블 스페이스 맵의 범위 수
TBSP_NUM_CONTAINERS	BIGINT	tablespace_num_containers - 테이블 스페이스의 컨테이너 수
TBSP_INITIAL_SIZE	BIGINT	tablespace_initial_size - 초기 테이블 스페이스 크기
TBSP_CURRENT_SIZE	BIGINT	tablespace_current_size - 현재 테이블 스페이스 크기
TBSP_MAX_SIZE	BIGINT	tablespace_max_size - 최대 테이블 스페이스 크기
TBSP_INCREASE_SIZE	BIGINT	tablespace_increase_size - 바이트 단위로 크기 늘리기
TBSP_INCREASE_SIZE_PERCENT	SMALLINT	tablespace_increase_size_percent - 퍼센트 단위로 크기 늘리기
TBSP_LAST_RESIZE_TIME	TIMESTAMP	tablespace_last_resize_time - 마지막으로 성공한 크기 조정 시간
TBSP_LAST_RESIZE_FAILED	SMALLINT	tablespace_last_resize_failed - 마지막으로 실패한 크기 조정 시도
DBPARTITIONNUM	SMALLINT	node_number - 노드 번호

## SNAPSHOT\_AGENT

주: 이 테이블 함수는 사용되지 않으며 518 페이지의 『SNAPAGENT 관리 뷰 및 SNAP\_GET\_AGENT 테이블 함수 - 에이전트 논리 데이터 그룹 응용프로그램 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_AGENT—(—dbname—,—dbpartitionnum—)—▶▶

스키마는 SYSPROC입니다.

SNAPSHOT\_AGENT 함수는 응용프로그램 스냅샷에서 에이전트에 대한 정보를 리턴합니다.

*dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 데이터베이스 인스턴스 아래 있는 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

*dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 265. COMPILATION\_ENV 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
AGENT_ID	BIGINT	agent_id - 응용프로그램 핸들(에이전트 ID)
AGENT_PID	BIGINT	agent_pid - EDU(Engine Dispatchable Unit)

## SNAPSHOT\_APPL

응용프로그램 스냅샷에서 일반 정보를 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며 1056 페이지의 『SNAP\_GET\_APPL 테이블 함수 - appl 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_APPL—(—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

### *dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 데이터베이스 인스턴스 아래 있는 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 266. SNAPSHOT\_APPL 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
AGENT_ID	BIGINT	<b>agent_id</b> - 응용프로그램 핸들(에이전트 ID)
UOW_LOG_SPACE_USED	BIGINT	<b>uow_log_space_used</b> - 사용된 작업 단위(UOW) 로그 스페이스
ROWS_READ	BIGINT	<b>rows_read</b> - 읽은 행 수
ROWS_WRITTEN	BIGINT	<b>rows_written</b> - 기록된 행 수
POOL_DATA_L_READS	BIGINT	<b>pool_data_l_reads</b> - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	<b>pool_data_p_reads</b> - 버퍼 풀 데이터 실제 읽기 수
POOL_DATA_WRITES	BIGINT	<b>pool_data_writes</b> - 버퍼 풀 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	<b>pool_index_l_reads</b> - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	<b>pool_index_p_reads</b> - 버퍼 풀 인덱스 실제 읽기 수
POOL_INDEX_WRITES	BIGINT	<b>pool_index_writes</b> - 버퍼 풀 인덱스 쓰기 수
POOL_READ_TIME	BIGINT	<b>pool_read_time</b> - 총 버퍼 풀 실제 읽기 시간



표 266. SNAPSHOT\_APPL 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_WRITE_TIME	BIGINT	<b>pool_write_time</b> - 총 버퍼 풀 실제 쓰기 시간
DIRECT_READS	BIGINT	<b>direct_reads</b> - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	<b>direct_writes</b> - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	<b>direct_read_reqs</b> - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	<b>direct_write_reqs</b> - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	<b>direct_read_time</b> - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	<b>direct_write_time</b> - 직접 쓰기 시간
POOL_DATA_TO_ESTORE	BIGINT	<b>pool_data_to_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_INDEX_TO_ESTORE	BIGINT	<b>pool_index_to_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_INDEX_FROM_ESTORE	BIGINT	<b>pool_index_from_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_DATA_FROM_ESTORE	BIGINT	<b>pool_data_from_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
UNREAD_PREFETCH_PAGES	BIGINT	<b>unread_prefetch_pages</b> - 읽지 않은 프리페치 페이지 수
LOCKS_HELD	BIGINT	<b>locks_held</b> - 보유된 잠금 수
LOCK_WAITS	BIGINT	<b>lock_waits</b> - 잠금 대기 수
LOCK_WAIT_TIME	BIGINT	<b>lock_wait_time</b> - 잠금 대기 시간
LOCK_ESCALS	BIGINT	<b>lock_escals</b> - 잠금 에스컬레이션 수
X_LOCK_ESCALS	BIGINT	<b>x_lock_escals</b> - 배타적 잠금 에스컬레이션 수
DEADLOCKS	BIGINT	<b>deadlocks</b> - 발견된 교착 상태 수
TOTAL_SORTS	BIGINT	<b>total_sorts</b> - 전체 정렬 수
TOTAL_SORT_TIME	BIGINT	<b>total_sort_time</b> - 총 정렬 시간
SORT_OVERFLOWS	BIGINT	<b>sort_overflows</b> - 정렬 오버플로우 수
COMMIT_SQL_STMTS	BIGINT	<b>commit_sql_stmts</b> - 시도된 커밋 명령문
ROLLBACK_SQL_STMTS	BIGINT	<b>rollback_sql_stmts</b> - 시도된 롤백 명령문
DYNAMIC_SQL_STMTS	BIGINT	<b>dynamic_sql_stmts</b> - 시도된 동적 SQL문
STATIC_SQL_STMTS	BIGINT	<b>static_sql_stmts</b> - 시도된 정적 SQL문
FAILED_SQL_STMTS	BIGINT	<b>failed_sql_stmts</b> - 실패한 명령문 조작
SELECT_SQL_STMTS	BIGINT	<b>select_sql_stmts</b> - 실행된 Select SQL문

표 266. SNAPSHOT\_APPL 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DDL_SQL_STMTS	BIGINT	<b>ddl_sql_stmts</b> - DDL(Data Definition Language) SQL문
UID_SQL_STMTS	BIGINT	<b>uid_sql_stmts</b> - 실행된 UPDATE/INSERT/DELETE SQL문
INT_AUTO_REBINDS	BIGINT	<b>int_auto_rebinds</b> - 내부 자동 리바인드 수
INT_ROWS_DELETED	BIGINT	<b>int_rows_deleted</b> - 삭제된 내부 행 수
INT_ROWS_UPDATED	BIGINT	<b>int_rows_updated</b> - 갱신된 내부 행 수
INT_COMMITS	BIGINT	<b>int_commits</b> - 내부 커밋 수
INT_ROLLBACKS	BIGINT	<b>int_rollback</b> - 내부 롤백 수
INT_DEADLOCK_ROLLBACKS	BIGINT	<b>int_deadlock_rollback</b> - 교착 상태로 인한 내부 롤백 수
ROWS_DELETED	BIGINT	<b>rows_deleted</b> - 삭제된 행 수
ROWS_INSERTED	BIGINT	<b>rows_inserted</b> - 삽입된 행 수
ROWS_UPDATED	BIGINT	<b>rows_updated</b> - 갱신된 행 수
ROWS_SELECTED	BIGINT	<b>rows_selected</b> - 선택된 행 수
BINDS_PRECOMPILES	BIGINT	<b>binds_precompiles</b> - 시도된 바인드/프리컴파일 수
OPEN_REM_CURS	BIGINT	<b>open_rem_curs</b> - 열린 리모트 커서 수
OPEN_REM_CURS_BLK	BIGINT	<b>open_rem_curs_blk</b> - 블로킹 상태의 열린 리모트 커서 수
REJ_CURS_BLK	BIGINT	<b>rej_curs_blk</b> - 거부된 블록 커서 요청 수
ACC_CURS_BLK	BIGINT	<b>acc_curs_blk</b> - 승인된 블록 커서 요청 수
SQL_REQS_SINCE_COMMIT	BIGINT	<b>sql_reqs_since_commit</b> - 커밋 이후 SQL 요청 수
LOCK_TIMEOUTS	BIGINT	<b>lock_timeouts</b> - 잠금 시간종료 수
INT_ROWS_INSERTED	BIGINT	<b>int_rows_inserted</b> - 삽입된 내부 행 수
OPEN_LOC_CURS	BIGINT	<b>open_loc_curs</b> - 열린 로컬 커서 수
OPEN_LOC_CURS_BLK	BIGINT	<b>open_loc_curs_blk</b> - 블로킹 상태의 열린 로컬 커서 수
PKG_CACHE_LOOKUPS	BIGINT	<b>pkg_cache_lookups</b> - 패키지 캐시 찾아보기 수
PKG_CACHE_INSERTS	BIGINT	<b>pkg_cache_inserts</b> - 패키지 캐시 삽입 수
CAT_CACHE_LOOKUPS	BIGINT	<b>cat_cache_lookups</b> - 카탈로그 캐시 찾아보기 수
CAT_CACHE_INSERTS	BIGINT	<b>cat_cache_inserts</b> - 카탈로그 캐시 삽입 수
CAT_CACHE_OVERFLOW	BIGINT	<b>cat_cache_overflows</b> - 카탈로그 캐시 오버플로우 수

표 266. SNAPSHOT\_APPL 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
CAT_CACHE_HEAP_FULL	BIGINT	<b>cat_cache_overflows</b> - 카탈로그 캐시 오버플로우 수
NUM_AGENTS	BIGINT	<b>num_agents</b> - 명령문에서 작업하는 에이전트 수
AGENTS_STOLEN	BIGINT	<b>agents_stolen</b> - 분실 에이전트 수
ASSOCIATED_AGENTS_TOP	BIGINT	<b>associated_agents_top</b> - 최대 연관 에이전트 수
APPL_PRIORITY	BIGINT	<b>appl_priority</b> - 응용프로그램 에이전트 우선순위
APPL_PRIORITY_TYPE	BIGINT	<b>appl_priority_type</b> - 응용프로그램 우선순위 유형
PREFETCH_WAIT_TIME	BIGINT	<b>prefetch_wait_time</b> - 프리페치 대기 시간
APPL_SECTION_LOOKUPS	BIGINT	<b>appl_section_lookups</b> - 섹션 찾아보기 수
APPL_SECTION_INSERTS	BIGINT	<b>appl_section_inserts</b> - 섹션 삽입 수
LOCKS_WAITING	BIGINT	<b>locks_waiting</b> - 잠금 대기 중인 에이전트 수
TOTAL_HASH_JOINS	BIGINT	<b>total_hash_joins</b> - 전체 해시 조인 수
TOTAL_HASH_LOOPS	BIGINT	<b>total_hash_loops</b> - 전체 해시 루프 수
HASH_JOIN_OVERFLOW	BIGINT	<b>hash_join_overflows</b> - 해시 조인 오버플로우 수
HASH_JOIN_SMALL_OVERFLOW	BIGINT	<b>hash_join_small_overflows</b> - 해시 조인 작은 오버플로우 수
APPL_IDLE_TIME	BIGINT	<b>appl_idle_time</b> - 응용프로그램 유휴 시간
UOW_LOCK_WAIT_TIME	BIGINT	<b>uow_lock_wait_time</b> - 잠금 상태로 대기한 전체 작업 단위(UOW) 시간
UOW_COMP_STATUS	BIGINT	<b>uow_comp_status</b> - 작업 단위(UOW) 완료 상태
AGENT_USR_CPU_TIME_S	BIGINT	<b>agent_usr_cpu_time</b> - 에이전트가 사용한 사용자 CPU 시간
AGENT_USR_CPU_TIME_MS	BIGINT	<b>agent_usr_cpu_time</b> - 에이전트가 사용한 사용자 CPU 시간
AGENT_SYS_CPU_TIME_S	BIGINT	<b>agent_sys_cpu_time</b> - 에이전트가 사용한 시스템 CPU 시간
AGENT_SYS_CPU_TIME_MS	BIGINT	<b>agent_sys_cpu_time</b> - 에이전트가 사용한 시스템 CPU 시간
APPL_CON_TIME	TIMESTAMP	<b>appl_con_time</b> - 연결 요청 시작 시간 소인
CONN_COMPLETE_TIME	TIMESTAMP	<b>conn_complete_time</b> - 연결 요청 완료 시간소인
LAST_RESET	TIMESTAMP	<b>last_reset</b> - 마지막 재설정 시간소인

표 266. SNAPSHOT\_APPL 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
UOW_START_TIME	TIMESTAMP	<b>uow_start_time</b> - 작업 단위(UOW) 시작 시간소인
UOW_STOP_TIME	TIMESTAMP	<b>uow_stop_time</b> - 작업 단위(UOW) 중지 시간소인
PREV_UOW_STOP_TIME	TIMESTAMP	<b>prev_uow_stop_time</b> - 이전 작업 단위(UOW) 완료 시간소인
UOW_ELAPSED_TIME_S	BIGINT	<b>uow_elapsed_time</b> - 최근 작업 단위(UOW) 경과 시간
UOW_ELAPSED_TIME_MS	BIGINT	<b>uow_elapsed_time</b> - 최근 작업 단위(UOW) 경과 시간
ELAPSED_EXEC_TIME_S	BIGINT	<b>elapsed_exec_time</b> - 명령문 실행 경과 시간
ELAPSED_EXEC_TIME_MS	BIGINT	<b>elapsed_exec_time</b> - 명령문 실행 경과 시간
INBOUND_COMM_ADDRESS	VARCHAR(32)	<b>inbound_comm_address</b> - 인바운드 통신 주소

## SNAPSHOT\_APPL\_INFO

응용프로그램 스냅샷에서 일반 정보를 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며 1062 페이지의 『SNAP\_GET\_APPL\_INFO 테이블 함수 - appl\_info 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_APPL\_INFO—(—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 데이터베이스 인스턴스 아래 있는 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 267. SNAPSHOT\_APPL\_INFO 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
AGENT_ID	BIGINT	<b>agent_id</b> - 응용프로그램 핸들(에이전트 ID)
APPL_STATUS	BIGINT	<b>appl_status</b> - 응용프로그램 상태
CODEPAGE_ID	BIGINT	<b>codepage_id</b> - 응용프로그램에서 사용되는 코드 페이지 ID
NUM_ASSOC_AGENTS	BIGINT	<b>num_assoc_agents</b> - 연관 에이전트 수
COORD_PARTITION_NUM	BIGINT	<b>coord_node</b> - 코디네이팅 노드
AUTHORITY_LVL	BIGINT	<b>authority_lvl</b> - 사용자 권한 부여 레벨
CLIENT_PID	BIGINT	<b>client_pid</b> - 클라이언트 프로세스 ID
COORD_AGENT_PID	BIGINT	<b>coord_agent_pid</b> - 코디네이터 에이전트
STATUS_CHANGE_TIME	TIMESTAMP	<b>status_change_time</b> - 응용프로그램 상태 변경 시간
CLIENT_PLATFORM	SMALLINT	<b>client_platform</b> - 클라이언트 운영 플랫폼
CLIENT_PROTOCOL	SMALLINT	<b>client_protocol</b> - 클라이언트 통신 프로토콜
COUNTRY_CODE	SMALLINT	<b>territory_code</b> - 데이터베이스 지역 코드
APPL_NAME	VARCHAR(256)	<b>appl_name</b> - 응용프로그램 이름
APPL_ID	VARCHAR(128)	<b>appl_id</b> - 응용프로그램 ID
SEQUENCE_NO	VARCHAR(4)	<b>sequence_no</b> - 시퀀스 번호
AUTH_ID	VARCHAR(128)	<b>auth_id</b> - 권한 부여 ID
CLIENT_NNAME	VARCHAR(128)	<b>client_nname</b> 모니터 요소가 사용되지 않습니다. 리턴된 값은 올바른 값이 아닙니다.
CLIENT_PRDID	VARCHAR(128)	<b>client_prdid</b> - 클라이언트 제품/버전 ID
INPUT_DB_ALIAS	VARCHAR(128)	<b>input_db_alias</b> - 입력 데이터베이스 별명
CLIENT_DB_ALIAS	VARCHAR(128)	<b>client_db_alias</b> - 응용프로그램에서 사용되는 데이터베이스 별명

표 267. SNAPSHOT\_APPL\_INFO 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DB_NAME	VARCHAR (128)	<b>db_name</b> - 데이터베이스 이름
DB_PATH	VARCHAR(1024)	<b>db_path</b> - 데이터베이스 경로
EXECUTION_ID	VARCHAR (128)	<b>execution_id</b> - 사용자 로그인 ID
CORR_TOKEN	VARCHAR (128)	<b>corr_token</b> - DRDA 상관 토큰
TPMON_CLIENT_USERID	VARCHAR(256)	<b>tpmon_client_userid</b> - TP 모니터 클라이언트 사용자 ID
TPMON_CLIENT_WKSTN	VARCHAR(256)	<b>tpmon_client_wkstn</b> - TP 모니터 클라이언트 워크스테이션 이름
TPMON_CLIENT_APP	VARCHAR(256)	<b>tpmon_client_app</b> - TP 모니터 클라이언트 응용프로그램 이름
TPMON_ACC_STR	VARCHAR(200)	<b>tpmon_acc_str</b> - TP 모니터 클라이언트 여카운팅 문자열

## SNAPSHOT\_BP

버퍼 풀 스냅샷에서 정보를 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며 1069 페이지의 『SNAP\_GET\_BP 테이블 함수 - 버퍼 풀 논리 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_BP—(—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 데이터베이스 인스턴스 아래 있는 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 268. SNAPSHOT\_BP 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
POOL_DATA_L_READS	BIGINT	<b>pool_data_l_reads</b> - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	<b>pool_data_p_reads</b> - 버퍼 풀 데이터 실제 읽기 수
POOL_DATA_WRITES	BIGINT	<b>pool_data_writes</b> - 버퍼 풀 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	<b>pool_index_l_reads</b> - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	<b>pool_index_p_reads</b> - 버퍼 풀 인덱스 실제 읽기 수
POOL_INDEX_WRITES	BIGINT	<b>pool_index_writes</b> - 버퍼 풀 인덱스 쓰기 수
POOL_READ_TIME	BIGINT	<b>pool_read_time</b> - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	<b>pool_write_time</b> - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_DATA_READS	BIGINT	<b>pool_async_data_reads</b> - 버퍼 풀 비동기 데이터 읽기 수
POOL_ASYNC_DATA_WRITES	BIGINT	<b>pool_async_data_writes</b> - 버퍼 풀 비동기 데이터 쓰기 수
POOL_ASYNC_INDEX_WRITES	BIGINT	<b>pool_async_index_writes</b> - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_ASYNC_READ_TIME	BIGINT	<b>pool_async_read_time</b> - 버퍼 풀 비동기 읽기 시간
POOL_ASYNC_WRITE_TIME	BIGINT	<b>pool_async_write_time</b> - 버퍼 풀 비동기 쓰기 시간
POOL_ASYNC_DATA_READ_REQS	BIGINT	<b>pool_async_data_read_reqs</b> - 버퍼 풀 비동기 읽기 요청 수
DIRECT_READS	BIGINT	<b>direct_reads</b> - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	<b>direct_writes</b> - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	<b>direct_read_reqs</b> - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	<b>direct_write_reqs</b> - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	<b>direct_read_time</b> - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	<b>direct_write_time</b> - 직접 쓰기 시간

표 268. SNAPSHOT\_BP 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_ASYNC_INDEX_READS	BIGINT	<b>pool_async_index_reads</b> - 버퍼 풀 비동기 인덱스 읽기 수
POOL_DATA_TO_ESTORE	BIGINT	<b>pool_data_to_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_INDEX_TO_ESTORE	BIGINT	<b>pool_index_to_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_INDEX_FROM_ESTORE	BIGINT	<b>pool_index_from_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_DATA_FROM_ESTORE	BIGINT	<b>pool_data_from_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
UNREAD_PREFETCH_PAGES	BIGINT	<b>unread_prefetch_pages</b> - 읽지 않은 프리페치 페이지 수
FILES_CLOSED	BIGINT	<b>files_closed</b> - 닫힌 데이터베이스 파일 수
BP_NAME	VARCHAR (128)	<b>bp_name</b> - 버퍼 풀 이름
DB_NAME	VARCHAR (128)	<b>db_name</b> - 데이터베이스 이름
DB_PATH	VARCHAR(1024)	<b>db_path</b> - 데이터베이스 경로
INPUT_DB_ALIAS	VARCHAR (128)	<b>input_db_alias</b> - 입력 데이터베이스 별명

## SNAPSHOT\_CONTAINER

테이블 스페이스 스냅샷에서 컨테이너 구성 정보를 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며 552 페이지의 『SNAPCONTAINER 관리 뷰 및 SNAP\_GET\_CONTAINER\_V91 테이블 함수 - tablespace\_container 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_CONTAINER—(—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.



### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 269. SNAPSHOT\_CONTAINER 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TABLESPACE_ID	BIGINT	<b>tablespace_id</b> - 테이블 스페이스 ID
TABLESPACE_NAME	VARCHAR (128)	<b>tablespace_name</b> - 테이블 스페이스 이름
CONTAINER_ID	BIGINT	<b>container_id</b> - 컨테이너 ID
CONTAINER_NAME	VARCHAR(256)	<b>container_name</b> - 컨테이너 이름
CONTAINER_TYPE	SMALLINT	<b>container_type</b> - 컨테이너 유형
TOTAL_PAGES	BIGINT	<b>container_total_pages</b> - 컨테이너의 전체 페이지 수
USABLE_PAGES	BIGINT	<b>container_usable_pages</b> - 컨테이너에서 사용할 수 있는 페이지 수
ACCESSIBLE	BIGINT	<b>container_accessible</b> - 컨테이너의 액세스 가능성
STRIPE_SET	BIGINT	<b>container_stripe_set</b> - 스트라이프 세트

---

## SNAPSHOT\_DATABASE

데이터베이스 스냅샷에서 정보를 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며 1083 페이지의 『SNAP\_GET\_DB\_V91 테이블 함수 - dbase 논리적 그룹에서 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_DATABASE—(—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

*dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 데이터베이스 인스턴스 아래 있는 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

*dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 270. SNAPSHOT\_DATABASE 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
SEC_LOG_USED_TOP	BIGINT	<b>sec_log_used_top</b> - 사용된 최대 2차 로그 스페이스
TOT_LOG_USED_TOP	BIGINT	<b>tot_log_used_top</b> - 사용된 최대 전체 로그 스페이스
TOTAL_LOG_USED	BIGINT	<b>total_log_used</b> - 사용된 전체 로그 스페이스
TOTAL_LOG_AVAILABLE	BIGINT	<b>total_log_available</b> - 사용 가능한 전체 로그
ROWS_READ	BIGINT	<b>rows_read</b> - 읽은 행 수
POOL_DATA_L_READS	BIGINT	<b>pool_data_l_reads</b> - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	<b>pool_data_p_reads</b> - 버퍼 풀 데이터 실제 읽기 수
POOL_DATA_WRITES	BIGINT	<b>pool_data_writes</b> - 버퍼 풀 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	<b>pool_index_l_reads</b> - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	<b>pool_index_p_reads</b> - 버퍼 풀 인덱스 실제 읽기 수

표 270. SNAPSHOT\_DATABASE 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_INDEX_WRITES	BIGINT	<b>pool_index_writes</b> - 버퍼 풀 인덱스 쓰기 수
POOL_READ_TIME	BIGINT	<b>pool_read_time</b> - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	<b>pool_write_time</b> - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_INDEX_READS	BIGINT	<b>pool_async_index_reads</b> - 버퍼 풀 비동기 인덱스 읽기 수
POOL_DATA_TO_ESTORE	BIGINT	<b>pool_data_to_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_INDEX_TO_ESTORE	BIGINT	<b>pool_index_to_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_INDEX_FROM_ESTORE	BIGINT	<b>pool_index_from_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_DATA_FROM_ESTORE	BIGINT	<b>pool_data_from_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_ASYNC_DATA_READS	BIGINT	<b>pool_async_data_reads</b> - 버퍼 풀 비동기 데이터 읽기 수
POOL_ASYNC_DATA_WRITES	BIGINT	<b>pool_async_data_writes</b> - 버퍼 풀 비동기 데이터 쓰기 수
POOL_ASYNC_INDEX_WRITES	BIGINT	<b>pool_async_index_writes</b> - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_ASYNC_READ_TIME	BIGINT	<b>pool_async_read_time</b> - 버퍼 풀 비동기 읽기 시간
POOL_ASYNC_WRITE_TIME	BIGINT	<b>pool_async_write_time</b> - 버퍼 풀 비동기 쓰기 시간
POOL_ASYNC_DATA_READ_REQS	BIGINT	<b>pool_async_data_read_reqs</b> - 버퍼 풀 비동기 읽기 요청 수
DIRECT_READS	BIGINT	<b>direct_reads</b> - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	<b>direct_writes</b> - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	<b>direct_read_reqs</b> - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	<b>direct_write_reqs</b> - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	<b>direct_read_time</b> - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	<b>direct_write_time</b> - 직접 쓰기 시간
UNREAD_PREFETCH_PAGES	BIGINT	<b>unread_prefetch_pages</b> - 읽지 않은 프리페치 페이지 수
FILES_CLOSED	BIGINT	<b>files_closed</b> - 닫힌 데이터베이스 파일 수

표 270. SNAPSHOT\_DATABASE 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_LSN_GAP_CLNS	BIGINT	<b>pool_lsn_gap_clns</b> - 버퍼 풀 로그 스페이스 클리너 트리거
POOL_DRDY_PG_STEAL_CLNS	BIGINT	<b>pool_drty_pg_steal_clns</b> - 버퍼 풀 희생(victim) 페이지 클리너 트리거
POOL_DRDY_PG_THRSH_CLNS	BIGINT	<b>pool_drty_pg_thrsh_clns</b> - 버퍼 풀 임계값 클리너 트리거
LOCKS_HELD	BIGINT	<b>locks_held</b> - 보유된 잠금 수
LOCK_WAITS	BIGINT	<b>lock_waits</b> - 잠금 대기 수
LOCK_WAIT_TIME	BIGINT	<b>lock_wait_time</b> - 잠금 대기 시간
LOCK_LIST_IN_USE	BIGINT	<b>lock_list_in_use</b> - 사용 중인 전체 잠금 목록 메모리
DEADLOCKS	BIGINT	<b>deadlocks</b> - 발견된 교착 상태 수
LOCK_ESCALS	BIGINT	<b>lock_escals</b> - 잠금 에스컬레이션 수
X_LOCK_ESCALS	BIGINT	<b>x_lock_escals</b> - 배타적 잠금 에스컬레이션 수
LOCKS_WAITING	BIGINT	<b>locks_waiting</b> - 잠금 대기 중인 에이전트 수
SORT_HEAP_ALLOCATED	BIGINT	<b>sort_heap_allocated</b> - 할당된 전체 정렬 힙
TOTAL_SORTS	BIGINT	<b>total_sorts</b> - 전체 정렬 수
TOTAL_SORT_TIME	BIGINT	<b>total_sort_time</b> - 총 정렬 시간
SORT_OVERFLOWS	BIGINT	<b>sort_overflows</b> - 정렬 오버플로우 수
ACTIVE_SORTS	BIGINT	<b>active_sorts</b> - 활성 정렬 수
COMMIT_SQL_STMTS	BIGINT	<b>commit_sql_stmts</b> - 시도된 커밋 명령문
ROLLBACK_SQL_STMTS	BIGINT	<b>rollback_sql_stmts</b> - 시도된 롤백 명령문
DYNAMIC_SQL_STMTS	BIGINT	<b>dynamic_sql_stmts</b> - 시도된 동적 SQL문
STATIC_SQL_STMTS	BIGINT	<b>static_sql_stmts</b> - 시도된 정적 SQL문
FAILED_SQL_STMTS	BIGINT	<b>failed_sql_stmts</b> - 실패한 명령문 조작
SELECT_SQL_STMTS	BIGINT	<b>select_sql_stmts</b> - 실행된 Select SQL문
DDL_SQL_STMTS	BIGINT	<b>ddl_sql_stmts</b> - DDL(Data Definition Language) SQL문
UID_SQL_STMTS	BIGINT	<b>uid_sql_stmts</b> - 실행된 UPDATE/INSERT/DELETE SQL문
INT_AUTO_REBINDS	BIGINT	<b>int_auto_rebinds</b> - 내부 자동 리바인드 수
INT_ROWS_DELETED	BIGINT	<b>int_rows_deleted</b> - 삭제된 내부 행 수
INT_ROWS_UPDATED	BIGINT	<b>int_rows_updated</b> - 갱신된 내부 행 수
INT_COMMITS	BIGINT	<b>int_commits</b> - 내부 커밋 수
INT_ROLLBACKS	BIGINT	<b>int_rollback</b> - 내부 롤백 수

표 270. SNAPSHOT\_DATABASE 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
INT_DEADLOCK_ROLLBACKS	BIGINT	<b>int_deadlock_rollback</b> - 교착 상태로 인한 내부 롤백 수
ROWS_DELETED	BIGINT	<b>rows_deleted</b> - 삭제된 행 수
ROWS_INSERTED	BIGINT	<b>rows_inserted</b> - 삽입된 행 수
ROWS_UPDATED	BIGINT	<b>rows_updated</b> - 갱신된 행 수
ROWS_SELECTED	BIGINT	<b>rows_selected</b> - 선택된 행 수
BINDS_PRECOMPILES	BIGINT	<b>binds_precompiles</b> - 시도된 바인드/프리컴파일 수
TOTAL_CONS	BIGINT	<b>total_cons</b> - 데이터베이스 활성화 이후 연결 수
APPLS_CUR_CONS	BIGINT	<b>appls_cur_cons</b> - 현재 연결된 응용프로그램 수
APPLS_IN_DB2	BIGINT	<b>appls_in_db2</b> - 현재 데이터베이스에서 실행 중인 응용프로그램 수
SEC_LOGS_ALLOCATED	BIGINT	<b>sec_logs_allocated</b> - 현재 할당된 2차 로그
DB_STATUS	BIGINT	<b>db_status</b> - 데이터베이스 상태
LOCK_TIMEOUTS	BIGINT	<b>lock_timeouts</b> - 잠금 시간종료 수
CONNECTIONS_TOP	BIGINT	<b>connections_top</b> - 최대 동시 연결 수
DB_HEAP_TOP	BIGINT	<b>db_heap_top</b> - 할당된 최대 데이터베이스 힙
INT_ROWS_INSERTED	BIGINT	<b>int_rows_inserted</b> - 삽입된 내부 행 수
LOG_READS	BIGINT	<b>log_reads</b> - 읽은 로그 페이지 수
LOG_WRITES	BIGINT	<b>log_writes</b> - 쓴 로그 페이지 수
PKG_CACHE_LOOKUPS	BIGINT	<b>pkg_cache_lookups</b> - 패키지 캐시 찾아보기 수
PKG_CACHE_INSERTS	BIGINT	<b>pkg_cache_inserts</b> - 패키지 캐시 삽입 수
CAT_CACHE_LOOKUPS	BIGINT	<b>cat_cache_lookups</b> - 카탈로그 캐시 찾아보기 수
CAT_CACHE_INSERTS	BIGINT	<b>cat_cache_inserts</b> - 카탈로그 캐시 삽입 수
CAT_CACHE_OVERFLOW	BIGINT	<b>cat_cache_overflows</b> - 카탈로그 캐시 오버플로우 수
CAT_CACHE_HEAP_FULL	BIGINT	<b>cat_cache_overflows</b> - 카탈로그 캐시 오버플로우 수
CATALOG_PARTITION	SMALLINT	<b>catalog_node</b> - 카탈로그 노드 번호
TOTAL_SEC_CONS	BIGINT	<b>total_sec_cons</b> - 2차 연결 수
NUM_ASSOC_AGENTS	BIGINT	<b>num_assoc_agents</b> - 연관 에이전트 수
AGENTS_TOP	BIGINT	<b>agents_top</b> - 작성된 에이전트 수
COORD_AGENTS_TOP	BIGINT	<b>coord_agents_top</b> - 최대 코디네이팅 에이전트 수

표 270. SNAPSHOT\_DATABASE 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
PREFETCH_WAIT_TIME	BIGINT	<b>prefetch_wait_time</b> - 프리페치 대기 시간
APPL_SECTION_LOOKUPS	BIGINT	<b>appl_section_lookups</b> - 섹션 찾아보기 수
APPL_SECTION_INSERTS	BIGINT	<b>appl_section_inserts</b> - 섹션 삽입 수
TOTAL_HASH_JOINS	BIGINT	<b>total_hash_joins</b> - 전체 해시 조인 수
TOTAL_HASH_LOOPS	BIGINT	<b>total_hash_loops</b> - 전체 해시 루프 수
HASH_JOIN_OVERFLOWS	BIGINT	<b>hash_join_overflows</b> - 해시 조인 오버플로우 수
HASH_JOIN_SMALL_OVERFLOWS	BIGINT	<b>hash_join_small_overflows</b> - 해시 조인 작은 오버플로우 수
PKG_CACHE_NUM_OVERFLOWS	BIGINT	<b>pkg_cache_num_overflows</b> - 패키지 캐시 오버플로우 수
PKG_CACHE_SIZE_TOP	BIGINT	<b>pkg_cache_size_top</b> - 패키지 캐시 상위 워터 마크(water mark)
DB_CONN_TIME	TIMESTAMP	<b>db_conn_time</b> - 데이터베이스 활성화 시간소인
SQLM_ELM_LAST_RESET	TIMESTAMP	<b>last_reset</b> - 마지막 재설정 시간소인
SQLM_ELM_LAST_BACKUP	TIMESTAMP	<b>last_backup</b> - 최종 백업 시간소인
APPL_CON_TIME	TIMESTAMP	<b>appl_con_time</b> - 연결 요청 시작 시간소인
DB_LOCATION	INTEGER	<b>db_location</b> - 데이터베이스 위치
SERVER_PLATFORM	INTEGER	<b>server_platform</b> - 서버 운영 체제
APPL_ID_OLDEST_XACT	BIGINT	<b>appl_id_oldest_xact</b> - 가장 오래된 트랜잭션이 포함된 응용프로그램
CATALOG_PARTITION_NAME	VARCHAR(128)	<b>catalog_node_name</b> - 카탈로그 노드 네트워크 이름
INPUT_DB_ALIAS	VARCHAR(128)	<b>input_db_alias</b> - 입력 데이터베이스 별명
DB_NAME	VARCHAR(128)	<b>db_name</b> - 데이터베이스 이름
DB_PATH	VARCHAR(1024)	<b>db_path</b> - 데이터베이스 경로

## SNAPSHOT\_DBM

DB2 데이터베이스 관리 프로그램의 스냅샷에서 정보를 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며 1080 페이지의 『SNAP\_GET\_DBM 테이블 함수 - dbm 논리 그룹화 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_DBM—(—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

*dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

널(NULL) 값이 지정되면, 스냅샷 API 요청 유형 1의 SNAPSHOT\_FILEW 스토어드 프로시저에서 이전에 파일을 작성하지 않은 경우에만 해당 스냅샷을 가져옵니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 271. SNAPSHOT\_DBM 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
SORT_HEAP_ALLOCATED	BIGINT	<b>sort_heap_allocated</b> - 할당된 전체 정렬 힙
POST_THRESHOLD_SORTS	BIGINT	<b>post_threshold_sorts</b> - 포스트 임계값 정렬 수
PIPED_SORTS_REQUESTED	BIGINT	<b>piped_sorts_requested</b> - 요청된 파이프 정렬 수
PIPED_SORTS_ACCEPTED	BIGINT	<b>piped_sorts_accepted</b> - 수락된 파이프 정렬 수
REM_CONS_IN	BIGINT	<b>rem_cons_in</b> - 데이터베이스 관리 프로그램에 대한 리모트 연결 수
REM_CONS_IN_EXEC	BIGINT	<b>rem_cons_in_exec</b> - 데이터베이스 관리 프로그램 모니터 요소에서 실행되는 리모트 연결 수
LOCAL_CONS	BIGINT	<b>local_cons</b> - 로컬 연결 수
LOCAL_CONS_IN_EXEC	BIGINT	<b>local_cons_in_exec</b> - 데이터베이스 관리 프로그램 모니터 요소에서 실행되는 로컬 연결 수
CON_LOCAL_DBASES	BIGINT	<b>con_local_dbases</b> - 현재 연결된 로컬 데이터베이스 수
AGENTS_REGISTERED	BIGINT	<b>agents_registered</b> - 등록된 에이전트 수
AGENTS_WAITING_ON_TOKEN	BIGINT	<b>agents_waiting_on_token</b> - 토큰을 기다리는 에이전트 수
DB2_STATUS	BIGINT	<b>db_status</b> - 데이터베이스 상태

표 271. SNAPSHOT\_DBM 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
AGENTS_REGISTERED_TOP	BIGINT	<b>agents_registered_top</b> - 등록된 최대 에이전트 수
AGENTS_WAITING_TOP	BIGINT	<b>agents_waiting_top</b> - 대기 중인 최대 에이전트 수
COMM_PRIVATE_MEM	BIGINT	<b>comm_private_mem</b> - 커밋된 전용 메모리
IDLE_AGENTS	BIGINT	<b>idle_agents</b> - 유휴 에이전트 수
AGENTS_FROM_POOL	BIGINT	<b>agents_from_pool</b> - 풀로부터 지정된 에이전트 수
AGENTS_CREATED_EMPTY_POOL	BIGINT	<b>agents_created_empty_pool</b> - 비어 있는 에이전트 풀로 인해 작성된 에이전트 수
COORD_AGENTS_TOP	BIGINT	<b>coord_agents_top</b> - 최대 코디네이팅 에이전트 수
MAX_AGENT_OVERFLOW	BIGINT	<b>max_agent_overflows</b> - 최대 에이전트 오버플로우 수
AGENTS_STOLEN	BIGINT	<b>agents_stolen</b> - 분실 에이전트 수
GW_TOTAL_CONS	BIGINT	<b>gw_total_cons</b> - DB2 Connect에 대한 전체 연결 시도 수
GW_CUR_CONS	BIGINT	<b>gw_cur_cons</b> - DB2 Connect에 대한 현재 연결 수
GW_CONS_WAIT_HOST	BIGINT	<b>gw_cons_wait_host</b> - 호스트 응답을 대기 중인 연결 수
GW_CONS_WAIT_CLIENT	BIGINT	<b>gw_cons_wait_client</b> - 클라이언트의 요청 전송을 대기 중인 연결 수
POST_THRESHOLD_HASH_JOINS	BIGINT	<b>post_threshold_hash_joins</b> - 해시 조인 임계값
INACTIVE_GW_AGENTS	BIGINT	<b>idle_agents</b> - 유휴 에이전트 수
NUM_GW_CONN_SWITCHES	BIGINT	<b>num_gw_conn_switches</b> - 연결 전환 수
DB2START_TIME	TIMESTAMP	<b>db2start_time</b> - 데이터베이스 관리 프로그램 시작 시간소인
LAST_RESET	TIMESTAMP	<b>last_reset</b> - 마지막 재설정 시간소인

## SNAPSHOT\_DYN\_SQL

동적 SQL 스냅샷에서 정보를 리턴합니다. 호환상의 이유로 인해 여전히 사용 가능한 SQLCACHE\_SNAPSHOT 함수를 대체합니다.

주: 이 테이블 함수는 사용되지 않으며 1093 페이지의 『SNAP\_GET\_DYN\_SQL\_V91 테이블 함수 - dynsql 논리적 그룹 스냅샷 정보 검색』으로 교체됩니다.



스키마는 SYSPROC입니다.

*dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

*dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 272. SNAPSHOT\_DYN\_SQL 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
ROWS_READ	BIGINT	<b>rows_read</b> - 읽은 행 수
ROWS_WRITTEN	BIGINT	<b>rows_written</b> - 기록된 행 수
NUM_EXECUTIONS	BIGINT	<b>num_executions</b> - 명령문 실행
NUM_COMPILATIONS	BIGINT	<b>num_compilations</b> - 명령문 컴파일
PREP_TIME_WORST	BIGINT	<b>prep_time_worst</b> - 명령문의 가장 안좋은 준비 시간
PREP_TIME_BEST	BIGINT	<b>prep_time_best</b> - 명령문의 가장 좋은 준비 시간
INT_ROWS_DELETED	BIGINT	<b>int_rows_deleted</b> - 삭제된 내부 행 수
INT_ROWS_INSERTED	BIGINT	<b>int_rows_inserted</b> - 삽입된 내부 행 수
INT_ROWS_UPDATED	BIGINT	<b>int_rows_updated</b> - 갱신된 내부 행 수
STMT_SORTS	BIGINT	<b>stmt_sorts</b> - 명령문 정렬 수

표 272. *SNAPSHOT\_DYN\_SQL* 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TOTAL_EXEC_TIME	BIGINT	<b>total_exec_time</b> - 경과된 명령문 실행 시간
TOTAL_SYS_CPU_TIME	BIGINT	<b>total_sys_cpu_time</b> - 명령문에 대한 전체 시스템 CPU
TOTAL_USR_CPU_TIME	BIGINT	<b>total_usr_cpu_time</b> - 명령문에 대한 전체 사용자 CPU
STMT_TEXT	CLOB(16M) <sup>1</sup>	<b>stmt_text</b> - SQL문 텍스트

<sup>1</sup> STMT\_TEXT는 장래 확장 전용으로 허용되는 CLOB(16M)으로 정의됩니다. 실제 명령문 텍스트 출력은 64K에서 절단됩니다.

## SNAPSHOT\_FCM

주: 이 테이블 함수는 사용되지 않으며 590 페이지의 『SNAPFCM 관리 뷰 및 SNAP\_GET\_FCM 테이블 함수 - fcm 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶—SNAPSHOT\_FCM—(—dbpartitionnum—)————▶

스키마는 SYSPROC입니다.

SNAPSHOT\_FCM 함수는 FCM(Fast Communication Manager)에 관한 데이터베이스 관리 프로그램의 레벨 정보를 리턴합니다.

### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 273. *SNAPSHOT\_FCM* 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
BUFF_FREE	BIGINT	<b>buff_free</b> - 현재 여유 FCM 버퍼
BUFF_FREE_BOTTOM	BIGINT	<b>buff_free_bottom</b> - 최소 여유 FCM 버퍼
MA_FREE	BIGINT	<b>ma_free</b> 모니터 요소가 중지됩니다. 널(NULL) 값이 중지된 모니터 요소에 대해 리턴됩니다.

표 273. *SNAPSHOT\_FCM* 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
MA_FREE_BOTTOM	BIGINT	<b>ma_free_bottom</b> 모니터 요소가 중지됩니다. 널(NULL) 값이 중지된 모니터 요소에 대해 리턴됩니다.
CE_FREE	BIGINT	<b>ce_free</b> 모니터 요소가 중지됩니다. 널(NULL) 값이 중지된 모니터 요소에 대해 리턴됩니다.
CE_FREE_BOTTOM	BIGINT	<b>ce_free_bottom</b> 모니터 요소가 중지됩니다. 널(NULL) 값이 중지된 모니터 요소에 대해 리턴됩니다.
RB_FREE	BIGINT	<b>rb_free</b> 모니터 요소가 중지됩니다. 널(NULL) 값이 중지된 모니터 요소에 대해 리턴됩니다.
RB_FREE_BOTTOM	BIGINT	<b>rb_free_bottom</b> 모니터 요소가 중지됩니다. 널(NULL) 값이 중지된 모니터 요소에 대해 리턴됩니다.
PARTITION_NUMBER	SMALLINT	<b>node_number</b> - 노드 번호

## SNAPSHOT\_FCMNODE

데이터베이스 관리 프로그램에 있는 FCM(Fast Communication Manager)의 스냅샷에서 정보를 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며 592 페이지의 『SNAPFCM\_PART 관리 뷰 및 SNAP\_GET\_FCM\_PART 테이블 함수 - fcm\_node 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_FCMNODE—(—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

*dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

널(NULL) 값이 지정되면, 스냅샷 API 요청 유형 1의 SNAPSHOT\_FILEW 스토어드 프로시저에서 이전에 파일을 작성하지 않은 경우에만 해당 스냅샷을 가져옵니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 274. *SNAPSHOT\_FCMNODE* 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
CONNECTION_STATUS	BIGINT	<b>connection_status</b> - 연결 상태
TOTAL_BUFFERS_SENT	BIGINT	<b>total_buffers_sent</b> - 전송된 전체 FCM 버퍼
TOTAL_BUFFERS_RCVD	BIGINT	<b>total_buffers_rcvd</b> - 수신된 전체 FCM 버퍼
PARTITION_NUMBER	SMALLINT	<b>node_number</b> - 노드 번호

## SNAPSHOT\_FILEW

주: 이 프로시저는 사용되지 않으며 666 페이지의 『SNAP\_WRITE\_FILE 프로시저』로 바뀌었습니다.

▶▶—SNAPSHOT\_FILEW—(—requestType—,—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

SNAPSHOT\_FILEW 프로시저는 인스턴스 디렉토리의 tmp 서브디렉토리에 있는 파일에 시스템 스냅샷 데이터를 기록합니다. SNAPSHOT\_FILEW 프로시저를 실행하려면, 사용자는 SYSADM, SYSCTRL, 또는 SYSMOINT 권한이 있어야 합니다. 저장된 스냅샷은 스냅샷 함수에 대한 입력으로 널(NULL) 값을 전달함으로써 SYSADM, SYSCTRL 또는 SYSMOINT 권한이 없는 사용자가 읽을 수 있습니다.

### *requestType*

유형 SMALLINT의 입력 인수는 sqlmon.h에서 정의된 대로 유효한 스냅샷 요청 유형을 지정합니다.

### *dbname*

유형 VARCHAR(128)의 입력 인수는 프로시저 호출시 현재 연결된 데이터베이스와 동일한 인스턴스에 있는 유효한 데이터베이스 이름을 지정합니다. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션을 지정하는 SMALLINT 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

예: 요청 유형 1(SQLMA\_DB2에 해당)를 지정하고 현재 연결된 데이터베이스 및 현재 데이터베이스 파티션을 디폴트값으로 하여 데이터베이스 관리 프로그램 정보의 스냅샷을 취하십시오.

```
CALL SNAPSHOT_FILEW (1, CAST (NULL AS VARCHAR(128)), CAST (NULL AS SMALLINT))
```

이는 UNIX 운영 체제의 인스턴스 디렉토리에 있는 /tmp/SQLMA\_DB2.dat 또는 Windows 운영 체제의 인스턴스 디렉토리에 있는 %tmp%\SQLMA\_DB2.dat에 스냅샷 데이터를 작성합니다.

---

## SNAPSHOT\_LOCK

잠금 스냅샷에서 정보를 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며 600 페이지의 『SNAPLOCK 관리 뷰 및 SNAP\_GET\_LOCK 테이블 함수 - 잠금 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_LOCK—(—dbname—,—dbpartitionnum—)—————▶▶

스키마는 SYSPROC입니다.

### *dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 275. *SNAPSHOT\_LOCK* 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
AGENT_ID	BIGINT	<b>agent_id</b> - 응용프로그램 핸들(에이전트 ID)
TABLE_FILE_ID	BIGINT	<b>table_file_id</b> - 테이블 파일 ID
LOCK_OBJECT_TYPE	BIGINT	<b>lock_object_type</b> - 대기하는 잠금 오브젝트 유형
LOCK_MODE	BIGINT	<b>lock_mode</b> - 잠금 모드
LOCK_STATUS	BIGINT	<b>lock_status</b> - 잠금 상태
LOCK_OBJECT_NAME	BIGINT	<b>lock_object_name</b> - 잠금 오브젝트 이름
PARTITION_NUMBER	SMALLINT	<b>node_number</b> - 노드 번호
LOCK_ESCALATION	SMALLINT	<b>lock_escalation</b> - 잠금 에스컬레이션
TABLE_NAME	VARCHAR (128)	<b>table_name</b> - 테이블 이름
TABLE_SCHEMA	VARCHAR (128)	<b>table_schema</b> - 테이블 스키마 이름
TABLESPACE_NAME	VARCHAR (128)	<b>tablespace_name</b> - 테이블 스페이스 이름

## SNAPSHOT\_LOCKWAIT

응용프로그램 스냅샷에서 잠금 대기 정보를 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며 606 페이지의 『SNAPLOCKWAIT 관리 뷰 및 SNAP\_GET\_LOCKWAIT 테이블 함수 - 잠금 대기 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_LOCKWAIT—(—*dbname*—,—*dbpartitionnum*—)—————▶▶

스키마는 SYSPROC입니다.

*dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 데이터베이스 인스턴스 아래 있는 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

*dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 276. SNAPSHOT\_LOCKWAIT 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
AGENT_ID	BIGINT	<b>agent_id</b> - 응용프로그램 핸들(에이전트 ID)
SUBSECTION_NUMBER	BIGINT	<b>ss_number</b> - 서브섹션 번호
LOCK_MODE	BIGINT	<b>lock_mode</b> - 잠금 모드
LOCK_OBJECT_TYPE	BIGINT	<b>lock_object_type</b> - 대기하는 잠금 오브젝트 유형
AGENT_ID_HOLDING_LK	BIGINT	<b>agent_id_holding_lock</b> - 잠금을 보유한 에이전트 ID
LOCK_WAIT_START_TIME	TIMESTAMP	<b>lock_wait_start_time</b> - 잠금 대기 시작 시간소인
LOCK_MODE_REQUESTED	BIGINT	<b>lock_mode_requested</b> - 요청된 잠금 모드
PARTITION_NUMBER	SMALLINT	<b>node_number</b> - 노드 번호
LOCK_ESCALATION	SMALLINT	<b>lock_escalation</b> - 잠금 에스컬레이션
TABLE_NAME	VARCHAR (128)	<b>table_name</b> - 테이블 이름
TABLE_SCHEMA	VARCHAR (128)	<b>table_schema</b> - 테이블 스키마 이름
TABLESPACE_NAME	VARCHAR (128)	<b>tablespace_name</b> - 테이블 스페이스 이름
APPL_ID_HOLDING_LK	VARCHAR (128)	<b>appl_id_holding_lk</b> - 잠금을 보유한 응용프로그램 ID

## SNAPSHOT QUIESCERS

주: 이 테이블 함수는 사용되지 않으며 649 페이지의 『SNAPTbsp\_QUIESCER 관리 뷰 및 SNAP\_GET\_TBSP\_QUIESCER 테이블 함수 - Quiescer 테이블 스페이스 스냅샷 정보 검색』으로 교체됩니다.

▶—SNAPSHOT\_QUIESCERS—(—dbname—,—dbpartitionnum—)————▶

스키마는 SYSPROC입니다.

SNAPSHOT\_QUIESCERS 함수는 테이블 스페이스 스냅샷에서 quiescer에 관한 정보를 리턴합니다.

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 277. SNAPSHOT\_QUIESCERS 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TABLESPACE_NAME	VARCHAR (128)	<b>tablespace_name</b> - 테이블 스페이스 이름
QUIESCER_TBS_ID	BIGINT	<b>quiescer_ts_id</b> - Quiescer 테이블 스페이스 ID
QUIESCER_OBJ_ID	BIGINT	<b>quiescer_obj_id</b> - Quiescer 오브젝트 ID
QUIESCER_AUTH_ID	BIGINT	<b>quiescer_auth_id</b> - Quiescer 사용자 권한 부여 ID
QUIESCER_AGENT_ID	BIGINT	<b>quiescer_agent_id</b> - Quiescer 에이전트 ID
QUIESCER_STATE	BIGINT	<b>quiescer_state</b> - Quiescer 상태



## SNAPSHOT\_RANGES

주: 이 테이블 함수는 사용되지 않으며 654 페이지의 『SNAPTbsp\_RANGE 관리 뷰 및 SNAP\_GET\_TBSP\_RANGE 테이블 함수 - 범위 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_RANGES—(—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

SNAPSHOT\_RANGES 함수는 범위 스냅샷에서 정보를 리턴합니다.

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 278. SNAPSHOT\_RANGES 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TABLESPACE_ID	BIGINT	<b>tablespace_id</b> - 테이블 스페이스 ID
TABLESPACE_NAME	VARCHAR (128)	<b>tablespace_name</b> - 테이블 스페이스 이름
RANGE_NUMBER	BIGINT	<b>range_number</b> - 범위 번호
RANGE_STRIPE_SET_NUMBER	BIGINT	<b>range_stripe_set_number</b> - 스트라이프 세트 번호
RANGE_OFFSET	BIGINT	<b>range_offset</b> - 범위 오프셋
RANGE_MAX_PAGE	BIGINT	<b>range_max_page_number</b> - 범위의 최대 페이지
RANGE_MAX_EXTENT	BIGINT	<b>range_max_extent</b> - 범위의 최대 Extent
RANGE_START_STRIPE	BIGINT	<b>range_start_stripe</b> - 시작 스트라이프
RANGE_END_STRIPE	BIGINT	<b>range_end_stripe</b> - 끝 스트라이프

표 278. SNAPSHOT\_RANGES 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
RANGE_ADJUSTMENT	BIGINT	<b>range_adjustment</b> - 범위 조정
RANGE_NUM_CONTAINER	BIGINT	<b>range_num_containers</b> - 범위의 컨테이너 수
RANGE_CONTAINER_ID	BIGINT	<b>range_container_id</b> - 범위 컨테이너

## SNAPSHOT\_STATEMENT

응용프로그램 스냅샷에서 명령문에 대한 정보를 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며 612 페이지의 『SNAPSTMT 관리 뷰 및 SNAP\_GET\_STMT 테이블 함수 - 명령문 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_STATEMENT—(—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 데이터베이스 인스턴스 아래 있는 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 279. SNAPSHOT\_STATEMENT 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.

표 279. SNAPSHOT\_STATEMENT 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
AGENT_ID	BIGINT	<b>agent_id</b> - 응용프로그램 핸들(에이전트 ID)
ROWS_READ	BIGINT	<b>rows_read</b> - 읽은 행 수
ROWS_WRITTEN	BIGINT	<b>rows_written</b> - 기록된 행 수
NUM_AGENTS	BIGINT	<b>num_agents</b> - 명령문에서 작업하는 에이전트 수
AGENTS_TOP	BIGINT	<b>agents_top</b> - 작성된 에이전트 수
STMT_TYPE	BIGINT	<b>stmt_type</b> - 명령문 유형
STMT_OPERATION	BIGINT	<b>stmt_operation/operation</b> - 명령문 조작
SECTION_NUMBER	BIGINT	<b>section_number</b> - 섹션 번호
QUERY_COST_ESTIMATE	BIGINT	<b>query_cost_estimate</b> - 추정 쿼리 비용
QUERY_CARD_ESTIMATE	BIGINT	<b>query_card_estimate</b> - 추정 쿼리 행 수
DEGREE_PARALLELISM	BIGINT	<b>degree_parallelism</b> - 병렬 처리 수준
STMT_SORTS	BIGINT	<b>stmt_sorts</b> - 명령문 정렬 수
TOTAL_SORT_TIME	BIGINT	<b>total_sort_time</b> - 총 정렬 시간
SORT_OVERFLOWS	BIGINT	<b>sort_overflows</b> - 정렬 오버플로우 수
INT_ROWS_DELETED	BIGINT	<b>int_rows_deleted</b> - 삭제된 내부 행 수
INT_ROWS_UPDATED	BIGINT	<b>int_rows_updated</b> - 갱신된 내부 행 수
INT_ROWS_INSERTED	BIGINT	<b>int_rows_inserted</b> - 삽입된 내부 행 수
FETCH_COUNT	BIGINT	<b>fetch_count</b> - 성공한 페치 수
STMT_START	TIMESTAMP	<b>stmt_start</b> - 명령문 조작 시작 시간소인
STMT_STOP	TIMESTAMP	<b>stmt_stop</b> - 명령문 조작 중지 시간소인
STMT_USR_CPU_TIME_S	BIGINT	<b>stmt_usr_cpu_time</b> - 명령문이 사용한 사용자 CPU 시간
STMT_USR_CPU_TIME_MS	BIGINT	<b>stmt_usr_cpu_time</b> - 명령문이 사용한 사용자 CPU 시간
STMT_SYS_CPU_TIME_S	BIGINT	<b>stmt_sys_cpu_time</b> - 명령문이 사용한 시스템 CPU 시간
STMT_SYS_CPU_TIME_MS	BIGINT	<b>stmt_sys_cpu_time</b> - 명령문이 사용한 시스템 CPU 시간
STMT_ELAPSED_TIME_S	BIGINT	<b>stmt_elapsed_time</b> - 가장 최근의 명령문 경과 시간
STMT_ELAPSED_TIME_MS	BIGINT	<b>stmt_elapsed_time</b> - 가장 최근의 명령문 경과 시간
BLOCKING_CURSOR	SMALLINT	<b>blocking_cursor</b> - 블로킹 커서
STMT_PARTITION_NUMBER	SMALLINT	<b>stmt_node_number</b> - 명령문 노드
CURSOR_NAME	VARCHAR (128)	<b>cursor_name</b> - 커서 이름
CREATOR	VARCHAR (128)	<b>creator</b> - 응용프로그램 작성자

표 279. *SNAPSHOT\_STATEMENT* 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
PACKAGE_NAME	VARCHAR (128)	<b>package_name</b> - 패키지 이름
STMT_TEXT	CLOB(16M) <sup>1</sup>	<b>stmt_text</b> - SQL문 텍스트

<sup>1</sup> STMT\_TEXT는 장래 확장 전용으로 허용되는 CLOB(16M)으로 정의됩니다. 실제 명령문 텍스트 출력은 64K에서 절단됩니다.

## SNAPSHOT\_SUBSECT

응용프로그램 스냅샷에서 액세스 플랜의 서브섹션에 대한 정보를 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며 621 페이지의 『SNAPSUBSECTION 관리 뷰 및 SNAP\_GET\_SUBSECTION 테이블 함수 - 서브섹션 논리 모니터 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_SUBSECT—(—dbname—, —dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

### *dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 데이터베이스 인스턴스 아래 있는 모든 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### *dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 280. SNAPSHOT\_SUBSECT 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
STMT_TEXT	CLOB(16M) <sup>1</sup>	<b>stmt_text</b> - SQL문 텍스트
SS_EXEC_TIME	BIGINT	<b>ss_exec_time</b> - 서브섹션 실행 경과 시간
TQ_TOT_SEND_SPILLS	BIGINT	<b>tq_tot_send_spills</b> - 오버플로우된 테이블 큐 버퍼의 총계
TQ_CUR_SEND_SPILLS	BIGINT	<b>tq_cur_send_spills</b> - 오버플로우된 현재 테이블 큐 버퍼 수
TQ_MAX_SEND_SPILLS	BIGINT	<b>tq_max_send_spills</b> - 최대 테이블 큐 버퍼 오버플로우 수
TQ_ROWS_READ	BIGINT	<b>tq_rows_read</b> - 테이블 큐에서 읽은 행 수
TQ_ROWS_WRITTEN	BIGINT	<b>tq_rows_written</b> - 테이블 큐에 기록된 행 수
ROWS_READ	BIGINT	<b>rows_read</b> - 읽은 행 수
ROWS_WRITTEN	BIGINT	<b>rows_written</b> - 기록된 행 수
SS_USR_CPU_TIME	BIGINT	<b>ss_usr_cpu_time</b> - 서브섹션이 사용한 사용자 CPU 시간
SS_SYS_CPU_TIME	BIGINT	<b>ss_sys_cpu_time</b> - 서브섹션이 사용한 시스템 CPU 시간
SS_NUMBER	INTEGER	<b>ss_number</b> - 서브섹션 번호
SS_STATUS	INTEGER	<b>ss_status</b> - 서브섹션 상태
SS_PARTITION_NUMBER	SMALLINT	<b>ss_node_number</b> - 서브섹션 노드 번호
TQ_PARTITION_WAITED_FOR	SMALLINT	<b>tq_node_waited_for</b> - 테이블 큐의 노드를 기다림
TQ_WAIT_FOR_ANY	INTEGER	<b>tq_wait_for_any</b> - 테이블 큐에 전송할 노드를 기다리는 중
TQ_ID_WAITING_ON	INTEGER	<b>tq_id_waiting_on</b> - 테이블 큐의 노드에서 대기

<sup>1</sup> STMT\_TEXT는 장래 확장 전용으로 허용되는 CLOB(16M)으로 정의됩니다. 실제 명령문 텍스트 출력은 64K에서 절단됩니다.

## SNAPSHOT\_SWITCHES

데이터베이스 스냅샷 전환 상태에 대한 정보를 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며 625 페이지의 『SNAPSHOT\_SWITCHES 관리 뷰 및 SNAP\_GET\_SWITCHES 테이블 함수 - 데이터베이스 스냅샷 스위치 상태 정보 검색』으로 교체됩니다.

스키마는 SYSPROC입니다.

*dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 281. SNAPSHOT\_SWITCHES 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
UOW_SW_STATE	SMALLINT	작업 단위(UOW) 모니터 기록 전환 상태(0 또는 1).
UOW_SW_TIME	TIMESTAMP	작업 단위(UOW) 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간.
STATEMENT_SW_STATE	SMALLINT	SQL문 모니터 기록 전환 상태(0 또는 1).
STATEMENT_SW_TIME	TIMESTAMP	SQL문 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간.
TABLE_SW_STATE	SMALLINT	테이블 활동 모니터 기록 전환 상태(0 또는 1).
TABLE_SW_TIME	TIMESTAMP	테이블 활동 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간
BUFFPOOL_SW_STATE	SMALLINT	버퍼 풀 활동 모니터 기록 전환 상태(0 또는 1)
BUFFPOOL_SW_TIME	TIMESTAMP	버퍼 풀 활동 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간
LOCK_SW_STATE	SMALLINT	잠금 모니터 기록 전환 상태(0 또는 1)
LOCK_SW_TIME	TIMESTAMP	잠금 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간
SORT_SW_STATE	SMALLINT	정렬 모니터 기록 전환 상태(0 또는 1)
SORT_SW_TIME	TIMESTAMP	정렬 모니터 기록 전환이 설정된 경우, 해당 전환이 설정된 날짜 및 시간

표 281. SNAPSHOT\_SWITCHES 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
PARTITION_NUMBER	SMALLINT	<b>node_number</b> - 노드 번호

## SNAPSHOT\_TABLE

테이블 스냅샷에서 활동 정보를 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며 629 페이지의 『SNAPTAB 관리 뷰 및 SNAP\_GET\_TAB\_V91 테이블 함수 - 테이블 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_TABLE—(—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 282. SNAPSHOT\_TABLE 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
ROWS_WRITTEN	BIGINT	<b>rows_written</b> - 기록된 행 수
ROWS_READ	BIGINT	<b>rows_read</b> - 읽은 행 수
OVERFLOW_ACCESSES	BIGINT	<b>overflow_accesses</b> - 오버플로우된 레코드에 액세스

표 282. SNAPSHOT\_TABLE 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
TABLE_FILE_ID	BIGINT	<b>table_file_id</b> - 테이블 파일 ID
TABLE_TYPE	BIGINT	<b>table_type</b> - 테이블 유형
PAGE_REORGS	BIGINT	<b>page_reorgs</b> - 페이지 재구성
TABLE_NAME	VARCHAR (128)	<b>table_name</b> - 테이블 이름
TABLE_SCHEMA	VARCHAR (128)	<b>table_schema</b> - 테이블 스키마 이름

## SNAPSHOT\_TBREORG

주: 이 테이블 함수는 사용되지 않으며 632 페이지의 『SNAPTAB\_REORG 관리 뷰 및 SNAP\_GET\_TAB\_REORG 테이블 함수 - 테이블 재구성 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_TBREORG—(—dbname—, —dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

SNAPSHOT\_TBREORG 함수는 결과 세트 형식으로 테이블 재구성 정보를 리턴합니다. 테이블이 재구성되지 않았다면 0행이 리턴됩니다. 실시간 스냅샷 정보를 확보하려면 사용자가 SYSADM, SYSCTRL 또는 SYSMOINT 권한이 있어야 합니다.

### dbname

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령이 리턴한 대로 "간접" 또는 "홈" 디렉토리 항목 유형을 갖는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

### dbpartitionnum

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.



함수는 테이블을 아래 표시된대로 리턴합니다.

표 283. *SNAPSHOT\_TBREORG* 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
TABLE_NAME	VARCHAR (128)	<b>table_name</b> - Table name
TABLE_SCHEMA	VARCHAR (128)	<b>table_schema</b> - 테이블 스키마 이름
PAGE_REORGS	BIGINT	<b>page_reorgs</b> - 페이지 재구성
REORG_PHASE	BIGINT	<b>reorg_phase</b> - 테이블 재구성 단계
REORG_MAX_PHASE	INTEGER	<b>reorg_max_phase</b> - 최대 테이블 재구성 단계
REORG_CURRENT_COUNTER	BIGINT	<b>reorg_current_counter</b> - 테이블 재구성 진행
REORG_MAX_COUNTER	BIGINT	<b>reorg_max_counter</b> - 전체 테이블 재구성 크기
REORG_TYPE	INTEGER	<b>reorg_type</b> - 테이블 재구성 속성
REORG_STATUS	BIGINT	<b>reorg_status</b> - 테이블 재구성 상태
REORG_COMPLETION	INTEGER	<b>reorg_completion</b> - 테이블 재구성 완료 플래그
REORG_START	TIMESTAMP	<b>reorg_start</b> - 테이블 재구성 시작 시간
REORG_END	TIMESTAMP	<b>reorg_end</b> - 테이블 재구성 종료 시간
REORG_PHASE_START	TIMESTAMP	<b>reorg_phase_start</b> - 테이블 재구성 단계 시작 시간
REORG_INDEX_ID	BIGINT	<b>reorg_index_id</b> - 테이블 재구성에 사용된 인덱스
REORG_TBSPC_ID	BIGINT	<b>reorg_tbspc_id</b> - 테이블이 재구성된 테이블 스페이스
PARTITION_NUMBER	SMALLINT	<b>node_number</b> - 노드 번호

## SNAPSHOT\_TBS

테이블 스페이스 스냅샷에서 활동 정보를 리턴합니다.

주: 이 테이블 함수는 더 이상 사용되지 않으며 638 페이지의 『SNAPTbsp 관리 뷰 및 SNAP\_GET\_TBSP\_V91 테이블 함수 - 테이블 스페이스 논리 데이터 그룹 스냅샷 정보 검색』에서 교체됩니다.

▶▶—SNAPSHOT\_TBS—(—dbname—,—dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

*dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데

이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 현재 연결된 데이터베이스에서 스냅샷을 가져오려면 널(NULL) 값을 지정하십시오.

*dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스를 연결할 수 있고 응용프로그램에서 사용하는 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 284. SNAPSHOT\_TBS 테이블 함수에서 리턴한 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 작성한 날짜와 시간.
POOL_DATA_L_READS	BIGINT	<b>pool_data_l_reads</b> - 버퍼 풀 데이터 논리적 읽기 수
POOL_DATA_P_READS	BIGINT	<b>pool_data_p_reads</b> - 버퍼 풀 데이터 실제 읽기 수
POOL_ASYNC_DATA_READS	BIGINT	<b>pool_async_data_reads</b> - 버퍼 풀 비동기 데이터 읽기 수
POOL_DATA_WRITES	BIGINT	<b>pool_data_writes</b> - 버퍼 풀 데이터 쓰기 수
POOL_ASYNC_DATA_WRITES	BIGINT	<b>pool_async_data_writes</b> - 버퍼 풀 비동기 데이터 쓰기 수
POOL_INDEX_L_READS	BIGINT	<b>pool_index_l_reads</b> - 버퍼 풀 인덱스 논리적 읽기 수
POOL_INDEX_P_READS	BIGINT	<b>pool_index_p_reads</b> - 버퍼 풀 인덱스 실제 읽기 수
POOL_INDEX_WRITES	BIGINT	<b>pool_index_writes</b> - 버퍼 풀 인덱스 쓰기 수
POOL_ASYNC_INDEX_WRITES	BIGINT	<b>pool_async_index_writes</b> - 버퍼 풀 비동기 인덱스 쓰기 수
POOL_READ_TIME	BIGINT	<b>pool_read_time</b> - 총 버퍼 풀 실제 읽기 시간
POOL_WRITE_TIME	BIGINT	<b>pool_write_time</b> - 총 버퍼 풀 실제 쓰기 시간
POOL_ASYNC_READ_TIME	BIGINT	<b>pool_async_read_time</b> - 버퍼 풀 비동기 읽기 시간

표 284. SNAPSHOT\_TBS 테이블 함수에서 리턴한 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
POOL_ASYNC_WRITE_TIME	BIGINT	<b>pool_async_write_time</b> - 버퍼 풀 비동기 쓰기 시간
POOL_ASYNC_DATA_READ_REQS	BIGINT	<b>pool_async_data_read_reqs</b> - 버퍼 풀 비동기 읽기 요청 수
DIRECT_READS	BIGINT	<b>direct_reads</b> - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	BIGINT	<b>direct_writes</b> - 데이터베이스에 직접 쓰기 수
DIRECT_READ_REQS	BIGINT	<b>direct_read_reqs</b> - 직접 읽기 요청 수
DIRECT_WRITE_REQS	BIGINT	<b>direct_write_reqs</b> - 직접 쓰기 요청 수
DIRECT_READ_TIME	BIGINT	<b>direct_read_time</b> - 직접 읽기 시간
DIRECT_WRITE_TIME	BIGINT	<b>direct_write_time</b> - 직접 쓰기 시간
UNREAD_PREFETCH_PAGES	BIGINT	<b>unread_prefetch_pages</b> - 읽지 않은 프리페치 페이지 수
POOL_ASYNC_INDEX_READS	BIGINT	<b>pool_async_index_reads</b> - 버퍼 풀 비동기 인덱스 읽기 수
POOL_DATA_TO_ESTORE	BIGINT	<b>pool_data_to_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_INDEX_TO_ESTORE	BIGINT	<b>pool_index_to_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_INDEX_FROM_ESTORE	BIGINT	<b>pool_index_from_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
POOL_DATA_FROM_ESTORE	BIGINT	<b>pool_data_from_estore</b> ESTORE 모니터 요소가 중지됩니다. 중지된 모니터 요소에 NULL 값이 리턴됩니다.
FILES_CLOSED	BIGINT	<b>files_closed</b> - 닫힌 데이터베이스 파일 수
TABLESPACE_NAME	VARCHAR (128)	<b>tablespace_name</b> - 테이블 스페이스 이름

## SNAPSHOT\_TBS\_CFG

주: 이 테이블 함수는 사용되지 않으며 794 페이지의 『SNAPTBSP\_PART 관리 뷰 및 SNAP\_GET\_TBSP\_PART\_V91 테이블 함수 - tablespace\_nodeinfo 논리 데이터 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SNAPSHOT\_TBS\_CFG—(—dbname—, —dbpartitionnum—)————▶▶

스키마는 SYSPROC입니다.

SNAPSHOT\_TBS\_CFG 함수는 테이블 스페이스 스냅샷에서 구성 정보를 리턴합니다.

*dbname*

이 함수를 호출할 때 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 데이터베이스 이름을 지정하는 VARCHAR(255) 유형의 입력 인수. LIST DATABASE DIRECTORY 명령으로 리턴되었으므로, "Indirect" 또는 "Home" 유형의 디렉토리 항목이 있는 데이터베이스 이름을 지정하십시오. 널(NULL) 값을 지정하여 현재 연결된 데이터베이스에서 스냅샷을 취하십시오.

*dbpartitionnum*

유효한 데이터베이스 파티션 수를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 지정하고 모든 데이터베이스 파티션에는 -2를 지정하십시오. 활성 데이터베이스 파티션은 데이터베이스가 응용프로그램에 의한 연결 및 사용 가능한 파티션입니다.

널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

두 매개변수가 모두 NULL로 설정되면, 이전에 스냅샷 API 요청 유형에 해당하는 SNAPSHOT\_FILEW 스토어드 프로시저로 파일을 작성하지 않은 경우에만 스냅샷이 취해집니다.

함수는 테이블을 아래 표시된대로 리턴합니다.

표 285. SNAPSHOT\_TBS\_CFG 테이블 함수로 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
SNAPSHOT_TIMESTAMP	TIMESTAMP	스냅샷을 취한 날짜 및 시간.
TABLESPACE_ID	BIGINT	<b>tablespace_id</b> - 테이블 스페이스 ID
TABLESPACE_NAME	VARCHAR (128)	<b>tablespace_name</b> - 테이블 스페이스 이름
TABLESPACE_TYPE	SMALLINT	<b>tablespace_type</b> - 테이블 스페이스 유형
TABLESPACE_STATE	BIGINT	<b>tablespace_state</b> - 테이블 스페이스 상태
NUM QUIESCERS	BIGINT	<b>tablespace_num_quiescers</b> - quiescers 수
STATE_CHANGE_OBJ_ID	BIGINT	<b>tablespace_state_change_object_id</b> - 상태 변경 오브젝트 ID
STATE_CHANGE_TBS_ID	BIGINT	<b>tablespace_state_change_ts_id</b> - 상태 변경 테이블 스페이스 ID
MIN_RECOVERY_TIME	TIMESTAMP	<b>tablespace_min_recovery_time</b> - 롤 포워드의 최소 복구 시간
TBS_CONTENTS_TYPE	SMALLINT	<b>tablespace_content_type</b> - 테이블 스페이스 콘텐츠 유형
BUFFERPOOL_ID	BIGINT	<b>tablespace_cur_pool_id</b> - 현재 사용 중인 버퍼 풀
NEXT_BUFFERPOOL_ID	BIGINT	<b>tablespace_next_pool_id</b> - 다음 시작 시 사용될 버퍼 풀
PAGE_SIZE	BIGINT	<b>tablespace_page_size</b> - 테이블 스페이스 페이지 크기

표 285. SNAPSHOT\_TBS\_CFG 테이블 함수로 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
EXTENT_SIZE	BIGINT	<b>tablespace_extent_size</b> - 테이블 스페이스 Extent 크기
PREFETCH_SIZE	BIGINT	<b>tablespace_prefetch_size</b> - 테이블 스페이스 프리 페치 크기
TOTAL_PAGES	BIGINT	<b>tablespace_total_pages</b> - 테이블 스페이스의 총 페이지
USABLE_PAGES	BIGINT	<b>tablespace_usable_pages</b> - 테이블 스페이스에서 사용 불가능한 페이지
USED_PAGES	BIGINT	<b>tablespace_used_pages</b> - 테이블 스페이스에서 사용된 페이지
FREE_PAGES	BIGINT	<b>tablespace_free_pages</b> - 테이블 스페이스에서 사용 가능한 페이지
PENDING_FREE_PAGES	BIGINT	<b>tablespace_pending_free_pages</b> - 테이블 스페이스에서 보류 중인 사용 가능한 페이지
HIGH_WATER_MARK	BIGINT	<b>pool_watermark</b> - 메모리 풀 워터 마크(water mark)
REBALANCER_MODE	BIGINT	<b>tablespace_rebalancer_mode</b> - 재조정 프로그램 모드
REBALANCER_EXTENTS_REMAINING	BIGINT	<b>tablespace_rebalancer_extents_remaining</b> - 재조정 프로그램으로 처리될 총 Extent 수
REBALANCER_EXTENTS_PROCESSED	BIGINT	<b>tablespace_rebalancer_extents_processed</b> - 재조정 프로그램이 처리한 총 Extent 수
REBALANCER_PRIORITY	BIGINT	<b>tablespace_rebalancer_priority</b> - 현재 재조정 프로그램 우선순위
REBALANCER_START_TIME	TIMESTAMP	<b>tablespace_rebalancer_start_time</b> - 재조정 프로그램 시작 시간
REBALANCER_RESTART_TIME	TIMESTAMP	<b>tablespace_rebalancer_restart_time</b> - 재조정 프로그램 재시작 시간
LAST_EXTENT_MOVED	BIGINT	<b>tablespace_rebalancer_last_extent_moved</b> - 재조정 프로그램으로 이동한 마지막 Extent
NUM_RANGES	BIGINT	<b>tablespace_num_ranges</b> - 테이블 스페이스 맵의 범위 수
NUM_CONTAINERS	BIGINT	<b>tablespace_num_containers</b> - 테이블 스페이스의 컨테이너 수

## SQLCACHE\_SNAPSHOT

주: 이 테이블 함수는 사용되지 않으며 1093 페이지의 『SNAP\_GET\_DYN\_SQL\_V91 테이블 함수 - dynsql 논리적 그룹 스냅샷 정보 검색』으로 교체됩니다.

▶▶—SQLCACHE\_SNAPSHOT—(—)————▶▶

스키마는 SYSFUN입니다.

SQLCACHE\_SNAPSHOT 함수는 DB2 동적 SQL문 캐시의 스냅샷 결과를 리턴합니다.

함수는 어떠한 인수도 취하지 않습니다. 아래에 표시된 대로 테이블을 리턴합니다.

표 286. SQLCACHE\_SNAPSHOT 테이블 함수에서 리턴된 정보

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
NUM_EXECUTIONS	INTEGER	num_executions - 명령문 실행
NUM_COMPILATIONS	INTEGER	num_compilations - 명령문 컴파일
PREP_TIME_WORST	INTEGER	prep_time_worst - 명령문의 가장 안좋은 준비 시간
PREP_TIME_BEST	INTEGER	prep_time_best - 명령문의 가장 좋은 준비 시간
INT_ROWS_DELETED	INTEGER	int_rows_deleted - 삭제된 내부 행 수
INT_ROWS_INSERTED	INTEGER	int_rows_inserted - 삽입된 내부 행 수
ROWS_READ	INTEGER	rows_read - 읽은 행 수
INT_ROWS_UPDATED	INTEGER	int_rows_updated - 갱신된 내부 행 수
ROWS_WRITE	INTEGER	rows_written - 기록된 행 수
STMT_SORTS	INTEGER	stmt_sorts - 명령문 정렬 수
TOTAL_EXEC_TIME_S	INTEGER	total_exec_time - 경과된 명령문 실행 시간
TOTAL_EXEC_TIME_MS	INTEGER	total_exec_time - 경과된 명령문 실행 시간
TOT_U_CPU_TIME_S	INTEGER	total_usr_cpu_time - 명령문에 대한 전체 사용자 CPU
TOT_U_CPU_TIME_MS	INTEGER	total_usr_cpu_time - 명령문에 대한 전체 사용자 CPU
TOT_S_CPU_TIME_S	INTEGER	total_sys_cpu_time - 명령문에 대한 전체 시스템 CPU
TOT_S_CPU_TIME_MS	INTEGER	total_sys_cpu_time - 명령문에 대한 전체 시스템 CPU

표 286. *SQLCACHE\_SNAPSHOT* 테이블 함수에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명 또는 해당 모니터 요소
DB_NAME	VARCHAR (128)	db_name - 데이터베이스 이름
STMT_TEXT	CLOB(16M) <sup>1</sup>	stmt_text - SQL문 텍스트

<sup>1</sup> STMT\_TEXT는 장래 확장 전용으로 허용되는 CLOB(16M)으로 정의됩니다. 실제 명령문 텍스트 출력은 64K에서 절단됩니다.

## SYSINSTALLROUTINES

주: 이 프로시저는 현재 사용되지 않습니다. Linux, UNIX 및 Windows용 DB2 UDB 버전 8에서 새 프로시저 및 함수 작성에 사용된 프로시저입니다.

▶▶—SYSINSTALLROUTINES—(—)—————▶▶

스키마는 SYSPROC입니다.

## WLM\_GET\_ACTIVITY\_DETAILS - 특정 활동에 대한 세부 정보 리턴

주: 이 테이블 함수는 사용되지 않으며

379 페이지의 『MON\_GET\_ACTIVITY\_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기』로 교체됩니다.

이 함수는 하나 이상의 서비스 서브클래스의 기본 통계를 리턴합니다.

이 함수는 응용프로그램 핸들, 작업 ID 단위 및 활동 ID로 식별된 특정 활동에 대한 자세한 정보를 리턴합니다. 이 정보에는 활동이 위반된 임계값에 대한 세부사항이 포함됩니다.

### 구문

▶▶—WLM\_GET\_ACTIVITY\_DETAILS—(—application\_handle—,—uow\_id—,—  
▶▶—activity\_id—,—dbpartitionnum—)—————▶▶

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

*application\_handle*

유효한 응용프로그램 핸들을 지정하는 유형 BIGINT의 입력 인수. 인수가 널(null)이면, 이 함수로부터 행이 리턴되지 않습니다. 인수가 널(NULL)이면, SQL171N 오류가 리턴됩니다.

*uow\_id*

응용프로그램 내에서 유효한 고유 작업 단위 ID를 지정하는 INTEGER 유형의 입력 인수. 인수가 널(null)이면, 이 함수로부터 행이 리턴되지 않습니다. 인수가 널(NULL)이면, SQL171N 오류가 리턴됩니다.

*activity\_id*

작업 단위 내에서 유효한 고유 활동 ID를 지정하는 INTEGER 유형의 입력 인수. 인수가 널(null)이면, 이 함수로부터 행이 리턴되지 않습니다. 인수가 널(NULL)이면, SQL171N 오류가 리턴됩니다.

*dbpartitionnum*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 파티션 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션으로 1을 지정하거나 모든 데이터베이스 파티션으로 -2를 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

## 권한 부여

WLM\_GET\_ACTIVITY\_DETAILS 함수에 대한 EXECUTE 특권

예 :

개별 활동에 대한 자세한 정보는 WLM\_GET\_ACTIVITY\_DETAILS 테이블 함수를 사용하여 얻을 수 있습니다. 이 테이블 함수는 각 파티션의 이름값 쌍으로 활동 정보를 리턴합니다. 이 예는 응용프로그램 핸들 1, 작업 ID 단위 1 및 활동 ID 5로 식별된 활동의 각 파티션의 11개의 구성원 이름값 쌍 서브세트만 표시하도록 제한됩니다. 이름값 쌍의 전체 목록은 1149 페이지의 표 288 및 1152 페이지의 표 289의 내용을 참조하십시오.

```
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
```

이 쿼리의 출력 예는 다음과 같습니다.



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

## 사용법 참고

QUEUED의 ACTIVITY\_STATE는 임계값 티켓을 가져오기 위해 코디네이터 활동이 카탈로그 파티션에 대한 RPC를 작성했지만 아직 응답을 수신하지 않았음을 나타냅니다. 이 상태가 표시되면 WLM이 활동을 큐에 추가했음을 나타내거나 짧은 기간 동안 활동이 해당 티켓을 가져오는 프로세스 중임을 나타낼 수 있습니다. 활동이 실제로 큐에 있는지 정확하게 알기 위해 활동에서 작업 중인 에이전트를 판별하여 (WLM\_GET\_SERVICE\_CLASS\_AGENTS 테이블 함수 사용) 카탈로그 파티션에서 이 에이전트의 event\_object 값이 WLM\_QUEUE인지 확인할 수 있습니다.

## 리턴되는 정보

표 287. WLM\_GET\_ACTIVITY\_DETAILS에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호입니다.
NAME	VARCHAR(256)	요소 이름. 가능한 값은 표 288 및 1152 페이지의 표 289의 내용을 참조하십시오.
VALUE	VARCHAR(1024)	요소 값. 가능한 값은 표 288 및 1152 페이지의 표 289의 내용을 참조하십시오.

표 288. 리턴된 요소

요소 이름	설명
ACTIVITY_ID	응용프로그램 내의 고유 활동 ID

표 288. 리턴된 요소 (계속)

요소 이름	설명
ACTIVITY_STATE	가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• CANCEL_PENDING</li> <li>• EXECUTING</li> <li>• IDLE</li> <li>• INITIALIZING</li> <li>• QP_CANCEL_PENDING</li> <li>• QP_QUEUED</li> <li>• QUEUED</li> <li>• TERMINATING</li> <li>• 알 수 없음</li> </ul>
ACTIVITY_TYPE	가능한 값은 다음과 같습니다. <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>
APPLICATION_HANDLE	응용프로그램에 대한 시스템 전반 고유 ID. 단일 파티션 데이터베이스에서 이 ID는 16비트 카운터로 구성됩니다. 다중 파티션 데이터베이스에서 이 ID는 16비트 카운터로 병합된 조정된 파티션 번호로 구성됩니다. 또한, 이러한 ID는 응용프로그램에서 2차로 연결할 수 있는 모든 파티션에서 동일합니다.
COORD_PARTITION_NUM	활동의 코디네이터 파티션입니다.
DATABASE_WORK_ACTION_SET_ID	이 활동이 데이터베이스에 적용된 작업 조치 세트에 매핑된 경우, 이 컬럼에는 작업 조치 세트의 ID가 포함됩니다. 활동이 데이터베이스에 적용된 작업 조치 세트에 매핑된 경우, 이 컬럼에는 0이 포함됩니다.
DATABASE_WORK_CLASS_ID	이 활동이 데이터베이스에 적용된 작업 조치 세트에 매핑된 경우, 이 컬럼에는 이 활동의 작업 클래스 ID가 포함됩니다. 활동이 데이터베이스에 적용된 작업 조치 세트에 매핑된 경우, 이 컬럼에는 0이 포함됩니다.
EFFECTIVE_ISOLATION	이 활동의 유효 분리 레벨
EFFECTIVE_LOCK_TIMEOUT	이 활동에 대한 유효 잠금 시간종료 값
EFFECTIVE_QUERY_DEGREE	이 활동의 유효 쿼리 등급 값
ENTRY_TIME	이 활동이 시스템에 도착한 시간
INVOCATION_ID	동일한 중첩 레벨에서 이 활동에 대한 임의의 특정 호출을 다른 호출과 구별합니다. 활동이 중첩되지 않으면, 영(0)을 리턴합니다.
LAST_REFERENCE_TIME	이 활동에서 요청이 발생될 때마다 이 필드가 갱신됩니다.

표 288. 리턴된 요소 (계속)

요소 이름	설명
LOCAL_START_TIME	이 활동이 파티션에서 작업을 시작하는 시간입니다. 이 시간은 로컬 시간입니다. 활동이 시스템에 입력되었지만 큐에 있어서 실행이 시작되지는 않은 경우, 이 필드는 비어 있는 문자열일 수 있습니다.
NESTING_LEVEL	이 활동의 중첩 레벨을 나타냅니다. 중첩 레벨은 이 활동이 해당하는 최상위 활동에 중첩되는 정도입니다.
PACKAGE_NAME	활동이 SQL문이면 이것은 패키지의 이름을 나타냅니다.
PACKAGE_SCHEMA	활동이 SQL문이면 이것은 패키지의 스키마 이름을 나타냅니다.
PACKAGE_VERSION_ID	활동이 SQL문이면 이것은 패키지의 버전을 나타냅니다.
PARENT_ACTIVITY_ID	ID가 ACTIVITY_ID인 활동 상위에 대한 작업 단위(UOW)내의 고유 활동 ID입니다. 활동에 상위 활동이 없으면, 비어 있는 문자열을 리턴합니다.
PARENT_UOW_ID	응용프로그램 내의 고유 작업 단위 ID. 이 활동의 상위 활동이 시작된 원래 작업 단위를 참조하십시오. 활동에 상위 활동이 없거나 리모트 파티션에 있는 경우에는 비어 있는 문자열을 리턴합니다.
QP_QUERY_ID	활동이 쿼리인 경우 Query Patroller에 의해 이 활동에 지정된 쿼리 ID. 쿼리 ID 0은 Query Patroller가 이 활동에 쿼리 ID를 지정하지 않았음을 나타냅니다.
QUERY_COST_ESTIMATE	SQL 컴파일러에 의해 판별된 쿼리의 계산된 비용(timerons 단위)
ROUTINE_ID	루틴 고유 ID. 활동이 루틴의 파트가 아닌 경우, 영(0)을 리턴합니다.
ROWS_FETCHED	테이블에서 읽은 행 수입니다. 이 보고서는 이 보고서가 기록된 데이터베이스 파티션의 값만 보고합니다. DPF 시스템에서는 이 값이 전체 활동의 올바른 총계를 반영하지 않을 수도 있습니다. 명령문 모니터 스위치가 켜지지 않으면 이 요소가 수집되지 않고 대신 -1이 작성됩니다.
ROWS_MODIFIED	삽입, 갱신 또는 삭제된 행 수입니다. 이 보고서는 이 보고서가 기록된 데이터베이스 파티션의 값만 보고합니다. DPF 시스템에서는 이 값이 전체 활동의 올바른 총계를 반영하지 않을 수도 있습니다. 명령문 모니터 스위치가 켜지지 않으면 이 요소가 수집되지 않고 대신 -1이 작성됩니다.
SECTION_NUMBER	활동이 SQL문이면 이것은 섹션 번호를 나타냅니다.
SERVICE_CLASS_ID	이 활동이 속한 서비스 클래스의 고유 ID입니다.
SERVICE_CLASS_WORK_ACTION_SET_ID	이 활동이 서비스 클래스에 적용된 작업 조치 세트에 매핑된 경우, 이 컬럼에는 작업 조치 세트의 ID가 포함됩니다. 활동이 서비스 클래스에 적용된 작업 조치 세트에 매핑된 경우, 이 컬럼에는 0이 포함됩니다.
SERVICE_CLASS_WORK_CLASS_ID	이 활동이 서비스 클래스에 적용된 작업 조치 세트에 매핑된 경우, 이 컬럼에는 이 활동의 작업 클래스 ID가 포함됩니다. 활동이 서비스 클래스에 적용된 작업 조치 세트에 매핑된 경우, 이 컬럼에는 0이 포함됩니다.
STMT_PKG_CACHE_ID	명령문 패키지 캐시 ID

표 288. 리턴된 요소 (계속)

요소 이름	설명
STMT_TEXT	활동이 동적 SQL이거나 명령문 텍스트가 사용 가능한 정적 SQL이면, 이 필드는 명령문 텍스트의 처음 1024자를 포함합니다. 그 밖의 경우에는 비어 있는 문자열입니다.
SYSTEM_CPU_TIME	데이터베이스 관리 프로그램 에이전트 프로세스, 작업 단위 또는 명령문에 사용된 전체 시스템 CPU 시간(초 및 마이크로초 단위). 명령문 모니터 스위치 또는 시간소인 스위치가 켜지지 않으면 이 요소가 수집되지 않고 대신 -1이 작성됩니다.
UOW_ID	응용프로그램 내의 고유 작업 단위 ID. 이 활동이 시작된 원래 작업 단위를 참조하십시오.
USER_CPU_TIME	데이터베이스 관리 프로그램 에이전트 프로세스, 작업 단위 또는 명령문에 사용된 전체 사용자 CPU 시간(초 및 마이크로초 단위). 명령문 모니터 스위치 또는 시간소인 스위치가 켜지지 않으면 이 요소가 수집되지 않고 대신 -1이 작성됩니다.
UTILITY_ID	활동이 유틸리티인 경우 해당 유틸리티 ID입니다. 그렇지 않으면, 이 필드는 0입니다.

**중요사항:** WLM\_GET\_ACTIVITY\_DETAILS 테이블 함수는 현재 활동에 적용 중인 임계값만 표시합니다.

해당 임계값이 활동에 적용될 경우에만 다음 요소를 리턴합니다.

표 289. 적용 가능한 경우 리턴된 요소

요소 이름	설명
ACTIVITYTOTALTIME_THRESHOLD_ID	활동에 적용된 ACTIVITYTOTALTIME 임계값의 ID
ACTIVITYTOTALTIME_THRESHOLD_VALUE	ACTIVITYTOTALTIME 임계값 지속기간을 활동 입력 시간에 추가하여 계산된 시간소인. 이 시간소인에 도달했을 때 활동이 계속 실행 중이면 임계값에 위반됩니다.
ACTIVITYTOTALTIME_THRESHOLD_VIOLATED	'예'는 활동이 ACTIVITYTOTALTIME 임계값을 위반했음을 표시합니다. '아니오'는 활동이 아직 임계값을 위반하지 않았음을 표시합니다.
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_ID	활동에 적용된 CONCURRENTDBCOORDACTIVITIES_DB 임계값의 ID
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_QUEUED	'예'는 활동이 CONCURRENTDBCOORDACTIVITIES_DB 임계값에 의해 큐에 대기되었음을 표시합니다. '아니오'는 활동이 아직 큐에 대기되지 않았음을 표시합니다.
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_VALUE	활동에 적용된 CONCURRENTDBCOORDACTIVITIES_DB 임계값의 상한
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_VIOLATED	'예'는 활동이 CONCURRENTDBCOORDACTIVITIES_DB 임계값을 위반했음을 표시합니다. '아니오'는 활동이 아직 임계값을 위반하지 않았음을 표시합니다.

표 289. 적용 가능한 경우 리턴된 요소 (계속)

요소 이름	설명
CONCURRENTDBCOORDACTIVITIES_SUBCLASS_THRESHOLD_ID	활동에 적용된 CONCURRENTDBCOORDACTIVITIES_SUBCLASS 임계값의 ID
CONCURRENTDBCOORDACTIVITIES_SUBCLASS_THRESHOLD_QUEUED	'예'는 활동이 CONCURRENTDBCOORDACTIVITIES_SUBCLASS 임계값에 의해 큐에 대기되었음을 표시합니다. '아니오'는 활동이 아직 큐에 대기되지 않았음을 표시합니다.
CONCURRENTDBCOORDACTIVITIES_SUBCLASS_THRESHOLD_VALUE	활동에 적용된 CONCURRENTDBCOORDACTIVITIES_SUBCLASS 임계값의 상한
CONCURRENTDBCOORDACTIVITIES_SUBCLASS_THRESHOLD_VIOLATED	'예'는 활동이 CONCURRENTDBCOORDACTIVITIES_SUBCLASS 임계값을 위반했음을 표시합니다. '아니오'는 활동이 아직 임계값을 위반하지 않았음을 표시합니다.
CONCURRENTDBCOORDACTIVITIES_SUPERCLASS_THRESHOLD_ID	활동에 적용된 CONCURRENTDBCOORDACTIVITIES _SUPERCLASS 임계값의 상한
CONCURRENTDBCOORDACTIVITIES_SUPERCLASS_THRESHOLD_QUEUED	'예'는 활동이 CONCURRENTDBCOORDACTIVITIES _SUPERCLASS 임계값을 위반했음을 표시합니다. '아니오'는 활동이 대기되지 않았음을 표시합니다.
CONCURRENTDBCOORDACTIVITIES_SUPERCLASS_THRESHOLD_VALUE	활동에 적용된 CONCURRENTDBCOORDACTIVITIES _SUPERCLASS 임계값의 상한
CONCURRENTDBCOORDACTIVITIES_SUPERCLASS_THRESHOLD_VIOLATED	'예'는 활동이 CONCURRENTDBCOORDACTIVITIES _SUPERCLASS 임계값을 위반했음을 표시합니다. '아니오'는 활동이 임계값을 아직 위반하지 않았음을 표시합니다.
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_ID	활동에 적용된 CONCURRENTDBCOORDACTIVITIES_WORK _ACTION_SET 임계값의 상한
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_QUEUED	'예'는 활동이 CONCURRENTDBCOORDACTIVITIES _WORK_ACTION_SET 임계값에 의해 큐에 대기되었음을 표시합니다. 아니오'는 활동이 큐에 대기되지 않았음을 표시합니다.
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_VALUE	활동에 적용된 CONCURRENTDBCOORDACTIVITIES_WORK _ACTION_SET 임계값의 상한

표 289. 적용 가능한 경우 리턴된 요소 (계속)

요소 이름	설명
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_VIOLATED	'예'는 활동이 CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET 임계값을 위반했음을 표시합니다. '아니오'는 활동이 임계값을 아직 위반하지 않았음을 표시합니다.
CONCURRENTWORKLOADACTIVITIES_THRESHOLD_ID	활동에 적용된 CONCURRENTWORKLOADACTIVITIES 임계값의 ID
CONCURRENTWORKLOADACTIVITIES_THRESHOLD_VALUE	활동에 적용된 CONCURRENTWORKLOADACTIVITIES 임계값의 상한
CONCURRENTWORKLOADACTIVITIES_THRESHOLD_VIOLATED	'예'는 활동이 CONCURRENTWORKLOADACTIVITIES 임계값을 위반했음을 표시합니다. '아니오'는 활동이 아직 임계값을 위반하지 않았음을 표시합니다.
ESTIMATEDSQLCOST_THRESHOLD_ID	활동에 적용된 ESTIMATEDSQLCOST 임계값의 ID
ESTIMATEDSQLCOST_THRESHOLD_VALUE	활동에 적용된 ESTIMATEDSQLCOST 임계값의 상한
ESTIMATEDSQLCOST_THRESHOLD_VIOLATED	'1'은 활동이 ESTIMATEDSQLCOST 임계값을 위반했음을 표시합니다. '0'은 활동이 아직 임계값을 위반하지 않았음을 표시합니다.
SQLROWSRETURNED_THRESHOLD_ID	활동에 적용된 SQLROWSRETURNED 임계값의 ID
SQLROWSRETURNED_THRESHOLD_VALUE	활동에 적용된 SQLROWSRETURNED 임계값의 상한
SQLROWSRETURNED_THRESHOLD_VIOLATED	'예'는 활동이 SQLROWSRETURNED 임계값을 위반했음을 표시합니다. '아니오'는 활동이 아직 임계값을 위반하지 않았음을 표시합니다.
SQLTEMPSPACE_THRESHOLD_ID	활동에 적용된 SQLTEMPSPACE 임계값의 ID
SQLTEMPSPACE_THRESHOLD_VALUE	활동에 적용된 SQLTEMPSPACE 임계값의 상한
SQLTEMPSPACE_THRESHOLD_VIOLATED	'예'는 활동이 SQLTEMPSPACE 임계값을 위반했음을 표시합니다. '아니오'는 활동이 아직 임계값을 위반하지 않았음을 표시합니다.

## WLM\_GET\_SERVICE\_CLASS\_AGENTS - 서비스 클래스에서 실행 중인 에이전트 나열

WLM\_GET\_SERVICE\_CLASS\_AGENTS 함수는 지정한 서비스 클래스에서 또는 지정한 응용프로그램 대신 실행 중인 지정한 파티션에서 에이전트, 분리 모드 프로세스 (db2fmp 프로세스) 및 시스템 엔티티 목록을 리턴합니다. 시스템 엔티티는 페이지 클리너와 프리페처와 같이 비에이전트 스레드 및 프로세스입니다.

주: 이 테이블 함수는 사용되지 않으며

885 페이지의 『WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 테이블 함수 - 서비스 클래스에서 실행 중인 에이전트 나열』로 교체됩니다.

## 구문

```
►►—WLM_GET_SERVICE_CLASS_AGENTS—(—service_superclass_name—,—————►  
►—service_subclass_name—,—application_handle—,—dbpartitionnum—)—————►
```

스키마는 SYSPROC입니다.

## 테이블 함수 매개변수

### *service\_superclass\_name*

현재 연결된 데이터베이스에서 서비스 슈퍼 클래스 이름을 지정하는

VARCHAR

(128) 유형의 입력 인수입니다. 인수가 널(null)이거나 비어 있는 문자열이면 기타 매개변수 값과 일치하는 데이터베이스의 모든 슈퍼 클래스에 대해 데이터가 검색됩니다.

### *service\_subclass\_name*

슈퍼 클래스 내의 특정 서브클래스를 참조하는 VARCHAR

(128) 유형의 입력 인수. 인수가 널(null)이거나 비어 있는 문자열이면 기타 매개변수 값과 일치하는 데이터베이스의 모든 서브클래스에 대해 데이터가 검색됩니다.

### *application\_handle*

리턴되는 에이전트 정보의 응용프로그램 핸들을 지정하는 VARCHAR(255) 유형의 입력 인수입니다. 인수가 널(null)이면 기타 매개변수 값과 일치하는 데이터베이스의 모든 응용프로그램에 대해 데이터가 검색됩니다. 0을 지정하면 시스템 엔티티만 리턴됩니다.

### *dbpartitionnum*

현재 연결된 데이터베이스와 동일한 인스턴스에 있는 유효한 파티션 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션으로 1을 지정하거나 모든 데이터베이스 파티션으로 -2를 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

## 권한 부여

WLM\_GET\_SERVICE\_CLASS\_AGENTS 함수에 대한 EXECUTE 특권

## 예 :

다음 쿼리는 모든 데이터베이스 파티션에 대해 응용프로그램 핸들 1과 연관된 에이전트 목록을 리턴합니다. 응용프로그램 핸들은 LIST APPLICATIONS 명령 또는 WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES 테이블 함수를 사용하여 판별됩니다.

```

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(CAST(NULL AS VARCHAR
(128)),
      CAST(NULL AS VARCHAR
(128)), 1, -2)) AS SCDetails
ORDER BY APPHANDLE, PART, AGENT_TID

```

샘플 출력은 다음과 같습니다.

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

출력은 UOW ID 1 및 활동 ID 5의 활동을 대신하는 파티션 0의 코디네이터 에이전트와 서브에이전트 및 파티션 1의 서브에이전트를 보여줍니다. 코디네이터 에이전트는 요청이 페치 요청임을 보고합니다.

## 사용법 참고

매개변수는 AND되는 효과가 있습니다. 즉 서비스 수퍼 클래스 SUP\_A와 서브클래스 SUB\_B(SUB\_B는 SUP\_A의 서브클래스가 아님)와 같이 충돌하는 입력 매개변수를 지정하면 행이 리턴되지 않습니다.

## 리턴되는 정보

표 290. WLM\_GET\_SERVICE\_CLASS\_AGENTS에서 리턴된 정보

컬럼 이름	데이터 유형	설명
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	이 레코드가 수집된 서비스 수퍼 클래스의 이름.
SERVICE_SUBCLASS_NAME	VARCHAR (128)	이 레코드가 수집된 서비스 서브클래스의 이름.
APPLICATION_HANDLE	BIGINT	응용프로그램에 대한 시스템 전반 고유 ID. 단일 파티션의 데이터베이스에서 이 ID는 16비트의 카운터로 구성됩니다. 다중 파티션의 데이터베이스에서 이 ID는 16비트 카운터로 병합된 조정 파티션 번호로 구성됩니다. 또한 이 ID는 응용프로그램이 보조 연결을 작성하는 모든 파티션에서 동일합니다.
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호입니다.



표 290. WLM\_GET\_SERVICE\_CLASS\_AGENTS에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
ENTITY	VARCHAR(32)	다음 값 중 하나입니다. <ul style="list-style-type: none"> <li>db2agent - 엔티티 유형이 에이전트입니다.</li> <li>db2fmp(pid) - 엔티티가 분리 모드 프로세스이며 여기서 pid는 분리 모드 프로세스의 프로세스 ID입니다.</li> <li>시스템 엔티티 이름</li> </ul>
WORKLOAD_NAME	VARCHAR(128)	이 레코드가 수집된 워크로드의 이름.
WORKLOAD_OCCURRENCE_ID	INTEGER	워크로드 어커런스의 ID. 코디네이터 데이터베이스 파티션 번호와 워크로드 이름으로 연결되어 있지 않으면 이 ID는 워크로드 어커런스를 고유하게 식별하지 않습니다.
UOW_ID	INTEGER	이 활동이 시작된 작업 단위(UOW)의 고유 ID.
ACTIVITY_ID	INTEGER	작업 단위(UOW)내의 고유 활동 ID
PARENT_UOW_ID	INTEGER	활동의 상위 활동이 시작된 작업 단위(UOW)의 고유 ID. 활동에 상위가 없는 경우 컬럼 값은 널(NULL)입니다.
PARENT_ACTIVITY_ID	INTEGER	ID가 activity_id인 활동 상위의 작업 단위(UOW)내에서 고유 활동 ID. 활동에 상위가 없는 경우 컬럼 값은 널(NULL)입니다.
AGENT_TID	BIGINT	에이전트 또는 시스템 엔티티의 스레드 ID. 이 ID를 사용할 수 없는 경우에는 컬럼 값이 널(NULL)입니다.
AGENT_TYPE	VARCHAR(32)	에이전트 유형. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>COORDINATOR</li> <li>OTHER</li> <li>PDBSUBAGENT</li> <li>SMPSUBAGENT</li> </ul> 에이전트 유형이 COORDINATOR인 경우 에이전트 ID는 집중기(concentrator) 환경에서 변경될 수 있습니다.
SMP_COORDINATOR	INTEGER	에이전트가 SMP 코디네이터인지 여부를 나타냄. 코디네이터인 경우 값은 1이며 아닌 경우 값은 0입니다.
AGENT_SUBTYPE	VARCHAR(32)	에이전트 부속 유형. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>DSS</li> <li>OTHER</li> <li>RPC</li> <li>SMP</li> </ul>
AGENT_STATE	VARCHAR(32)	에이전트 상태. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>ACTIVE</li> <li>ASSOCIATED</li> </ul>
EVENT_TYPE	VARCHAR(32)	이 에이전트에서 마지막으로 처리하는 이벤트 유형. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>ACQUIRE</li> <li>PROCESS</li> <li>WAIT</li> </ul>

표 290. WLM\_GET\_SERVICE\_CLASS\_AGENTS에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
EVENT_OBJECT	VARCHAR(32)	이 에이전트에서 마지막으로 처리한 이벤트 오브젝트. 가능한 값은 다음과 같습니다. <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>
EVENT_STATE	VARCHAR(32)	이 에이전트에서 마지막으로 처리한 이벤트 상태. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>• EXECUTING</li> <li>• IDLE</li> </ul>
REQUEST_ID	VARCHAR(64)	요청 ID. 이 값은 <i>application_handle</i> 값의 조합에서만 고유합니다. 이를 사용하여 오래 걸리는 임의의 요청과 다중 요청을 구분할 수 있습니다(예: 임의의 장기 폐치 및 다중 폐치).

표 290. WLM\_GET\_SERVICE\_CLASS\_AGENTS에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
REQUEST_TYPE	VARCHAR(32)	<p>요청 유형. 가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• 코디네이터 에이전트의 경우                             <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>, 여기서 <i>number</i>는 내부 상수 값</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_SUBTYPE이 DSS 또는 SMP인 서브 에이전트                             <ul style="list-style-type: none"> <li>- 서브섹션 번호가 0이 아닌 경우 서브섹션 번호 양식은 SUBSECTION:<i>subsection number</i>이며 그 이외의 경우에는 NULL입니다.</li> </ul> </li> </ul>

표 290. WLM\_GET\_SERVICE\_CLASS\_AGENTS에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
REQUEST_TYPE (계속)	VARCHAR(32)	<ul style="list-style-type: none"> <li>• AGENT_SUBTYPE이 RPC인 서브 에이전트                             <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>• AGENT_SUBTYPE이 OTHER인 서브 에이전트                             <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>, 여기서 <i>number</i>는 내부 상수 값</li> <li>- INTERRUPT</li> <li>- NOOP(요청이 없는 경우)</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	ID가 activity_id인 활동의 중첩 레벨. 중첩 레벨은 이 활동이 최상위 활동에 중첩된 깊이입니다.
INVOCATION_ID	INTEGER	동일한 중첩 레벨에서 활동의 특정 호출을 다른 호출과 구별하는 ID입니다.
ROUTINE_ID	INTEGER	루틴의 고유 ID. 컬럼이 루틴의 파트가 아닌 경우 컬럼 값은 널(NULL)입니다.

## WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES - 워크로드 어커런스 목록

특정 파티션의 제공된 서비스 클래스에서 실행 중인 전체 워크로드 어커런스 목록을 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며

893 페이지의 『WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97

- 워크로드 어커런스 나열』로 교체됩니다.

워크로드 어커런스는 속성이 워크로드 정의와 일치하므로 워크로드와 연관되거나 워크로드에 지정됩니다.

### 구문

```
►►—WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES—(—service_superclass_name—, —————→  
►—service_subclass_name—, —dbpartitionnum—)—————→►►
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

*service\_superclass\_name*

현재 연결된 데이터베이스에서 유효한 서비스 수퍼 클래스 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수. 인수가 널(NULL) 또는 비어 있는 문자열이면, 기타 매개변수가 일치하는 데이터베이스의 모든 수퍼 클래스에 대해 데이터가 검색됩니다.

*service\_subclass\_name*

현재 연결된 데이터베이스의 유효한 서비스 서브클래스를 지정하는 VARCHAR

(128) 유형의 입력 인수. 인수가 널(NULL) 또는 비어 있는 문자열이면, 기타 매개변수가 일치하는 데이터베이스의 모든 서브클래스에 대해 데이터가 검색됩니다.

*dbpartitionnum*

현재 연결된 데이터베이스와 동일한 인스턴스에 있는 유효한 파티션 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션에는 -1을 표시하고 모든 데이터베이스 파티션에는 -2를 표시하십시오. 널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

### 권한 부여

WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES 함수에 대한 EXECUTE 특권

예 :

관리자가 시스템 전반에 실행 중인 워크로드 어커런스를 보고자 하는 경우 *service\_superclass\_name* 및 *service\_subclass\_name*의 널(NULL) 값 또는 비어 있는 문자열 그리고 *dbpartitionnum*의 -2를 사용하여 *WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES* 함수를 호출할 수 있습니다.

```
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
(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
```

시스템에 네 개의 데이터베이스 파티션이 있고 지금 두 개의 워크로드를 실행 중이라고 가정하고 위의 쿼리가 다음과 같은 결과를 작성합니다.

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	...

이 쿼리의 출력(계속).

...	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

## 사용법 참고

매개변수는 함께 AND되는 효과가 있습니다. 즉 서비스 수퍼 클래스 SUP\_A와 서브클래스 이름 SUB\_B와 같은 입력 매개변수와의 충돌을 지정하여 SUB\_B가 SUP\_A의 서브클래스가 아니면 행이 리턴되지 않습니다.

주: 워크로드 어커런스에 보고된 통계(예: coord\_act\_completed\_total)는 해당 워크로드 통계와 조합될 때 각 작업 단위가 시작될 때 재설정됩니다.

## 리턴되는 정보

표 291. WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	이 레코드가 수집된 서비스 수퍼 클래스의 이름.
SERVICE_SUBCLASS_NAME	VARCHAR (128)	이 레코드가 수집된 서비스 서브클래스의 이름.
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호입니다.
COORD_PARTITION_NUM	SMALLINT	제공된 워크로드 어커런스에 대한 코디네이터 파티션의 파티션 번호
APPLICATION_HANDLE	BIGINT	응용프로그램의 시스템 전체의 고유 ID입니다. 단일 파티션 데이터베이스에서 이 ID는 16비트 카운터로 구성됩니다. 다중 파티션 데이터베이스에서 이 ID는 16비트 카운터로 병합된 조정된 파티션 번호로 구성됩니다. 또한, 이러한 ID는 응용프로그램에서 2차로 연결할 수 있는 모든 파티션에서 동일합니다.
WORKLOAD_NAME	VARCHAR (128)	이 레코드가 수집된 워크로드의 이름.
WORKLOAD_OCCURRENCE_ID	INTEGER	워크로드 어커런스 ID. 코디네이터 데이터베이스 파티션 번호와 워크로드 이름으로 연결되어 있지 않으면 워크로드 어커런스를 고유하게 식별하지 않습니다.

표 291. WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
WORKLOAD_OCCURRENCE_STATE	VARCHAR(32)	<p>가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• DECOUPLED - 워크로드 어커런스에는 지정된 코디네이터가 없습니다(컨센트레이터 case).</li> <li>• DISCONNECTPEND - 워크로드 어커런스를 데이터베이스에서 연결 해제중입니다.</li> <li>• FORCED - 워크로드 어커런스가 강제 실행되었습니다.</li> <li>• INTERRUPTED - 워크로드 어커런스가 인터럽트되었습니다.</li> <li>• QUEUED - 워크로드 어커런스 코디네이터가 Query Patroller 또는 워크로드 관리 큐 임계값에 의해 큐에 대기되었습니다. 파티션된 데이터베이스 환경에서, 이 상태는 임계값 티켓을 가져오기 위해 코디네이터 에이전트가 카탈로그 파티션에 대한 RPC를 작성했지만 아직 응답을 수신하지 않았음을 나타냅니다.</li> <li>• TRANSIENT - 워크로드 어커런스가 아직 서비스 수퍼클래스에 맵핑되지 않았습니다.</li> <li>• UOWEXEC - 워크로드 어커런스가 요청을 처리 중입니다.</li> <li>• UOWWAIT - 워크로드 어커런스가 클라이언트의 요청을 대기 중입니다.</li> </ul>
UOW_ID	INTEGER	응용프로그램 내의 고유 작업 단위 ID.
SYSTEM_AUTH_ID	VARCHAR(128)	워크로드 어커런스가 시스템에 주입된 시스템 권한 부여 ID
SESSION_AUTH_ID	VARCHAR(128)	워크로드 어커런스가 시스템에 주입된 세션 권한 부여 ID
APPLICATION_NAME	VARCHAR(128)	이 워크로드 어커런스를 작성한 응용프로그램의 이름
CLIENT_WRKSTNNAME	VARCHAR(255)	이 워크로드 어커런스에 대한 CLIENT_WRKSTNNAME 특수 레지스터의 현재 값
CLIENT_ACCTNG	VARCHAR(255)	이 워크로드 어커런스에 대한 CLIENT_ACCTNG 특수 레지스터의 현재 값
CLIENT_USER	VARCHAR(255)	이 워크로드 어커런스에 대한 CLIENT_USERID 특수 레지스터의 현재 값
CLIENT_APPLNAME	VARCHAR(255)	이 워크로드 어커런스에 대한 CLIENT_APPLNAME 특수 레지스터의 현재 값
COORD_ACT_COMPLETED_TOTAL	INTEGER	이 워크로드 어커런스의 현재 작업 단위에 지금까지 완료된 중첩 레벨의 코디네이터 활동 수 이 통계는 이 워크로드 어커런스의 활동이 완료될 때마다 갱신되며 각 작업 단위가 시작될 때 재설정됩니다.



표 291. WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
COORD_ACT_ABORTED_TOTAL	INTEGER	이 워크로드 어커런스의 현재 작업 단위에 지금까지 중단된 코디네이터 활동 수 이 통계는 이 워크로드 어커런스의 활동이 중단될 때마다 갱신되며 각 작업 단위가 시작될 때 재설정됩니다.
COORD_ACT_REJECTED_TOTAL	INTEGER	이 워크로드 어커런스의 현재 작업 단위에 지금까지 거부된 코디네이터 활동 수 활동은 실행 금지 작업 조치 또는 예측적 임계값으로 인해 실행되지 않은 경우 거부됨으로 간주됩니다. 이 통계는 이 워크로드 어커런스의 활동이 거부될 때마다 갱신되며 각 작업 단위가 시작될 때 재설정됩니다.
CONCURRENT_ACT_TOP	INTEGER	현재 작업 단위(UOW)에서 이 워크로드 어커런스에 도달한 실행(유휴 및 대기 중 포함) 또는 큐에 대기됨 상태의 중첩 레벨에서 가장 높은 동시 활동 수. 이 통계는 각 작업 단위가 시작될 때 재설정됩니다.

## WLM\_GET\_SERVICE\_SUBCLASS\_STATS - 서비스 서브클래스의 통계 리턴

주: 이 테이블 함수는 사용되지 않으며

897 페이지의 『WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97

테이블 함수 - 서비스 서브클래스의 통계 리턴』으로 교체됩니다.

이 함수는 하나 이상의 서비스 서브클래스의 기본 통계를 리턴합니다.

### 구문

```

▶▶—WLM_GET_SERVICE_SUBCLASS_STATS—(—service_superclass_name—, —————▶
▶—service_subclass_name—, —dbpartitionnum—)—————▶▶
    
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

*service\_superclass\_name*

이 함수 호출 시 현재 연결된 동일한 데이터베이스에서 유효한 서비스 수퍼 클래스 이름을 지정하는 VARCHAR (128) 유형의 입력 인수. 인수가 널(NULL) 또는 비어 있는 문자열이면, 데이터베이스의 모든 수퍼 클래스에 대해 데이터가 검색됩니다.

*service\_subclass\_name*

이 함수 호출 시 현재 연결된 동일한 데이터베이스에서 유효한 서비스 서브클래스 이름을 지정하는 VARCHAR

(128) 유형의 입력 인수. 인수가 널(NULL) 또는 비어 있는 문자열이면, 데이터베이스의 모든 서브클래스에 대해 데이터가 검색됩니다.

### *dbpartitionnum*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 파티션 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션으로 1을 지정하거나 모든 데이터베이스 파티션으로 -2를 지정하십시오. 널(NULL) 값이 지정된 경우, 내재적으로 -1이 설정됩니다.

## 권한 부여

WLM\_GET\_SERVICE\_SUBCLASS\_STATS 함수에 대한 EXECUTE 특권

### 예:

예 1: 실행되기 전에 모든 활동이 DB2 서비스 클래스에 맵핑되므로 모든 파티션의 모든 서비스 클래스를 쿼리하는 서비스 클래스 통계 테이블 함수를 사용하여 전역 상태를 정기적으로 모니터링할 수 있습니다. 인수에 널(NULL) 값을 전달하면 최종 인수 *dbpartitionnum*를 제외하고 그 인수에 의해 결과가 제한되어서는 안된다는 것을 나타냄을 참조하십시오. 여기에서 -2 값은 모든 데이터베이스 파티션의 데이터가 리턴됨을 나타냅니다. 이 예는 서비스 클래스 통계(예: 평균 활동 수명 및 표준 편차(초))를 리턴합니다.

```
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(CAST(NULL AS VARCHAR
(128)),
      CAST(NULL AS VARCHAR
(128)), -2)) AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, PART
```

다음은 이 쿼리로부터의 출력 예입니다.

```
SUPERCLASS_NAME    SUBCLASS_NAME      PART ...
-----
SYSDEFAULTUSERCLASS SYSDEFAULTSUBCLASS 0    ...
SYSDEFAULTUSERCLASS SYSDEFAULTSUBCLASS 1    ...
SYSDEFAULTUSERCLASS SYSDEFAULTSUBCLASS 2    ...
SYSDEFAULTUSERCLASS SYSDEFAULTSUBCLASS 3    ...
```

이 쿼리의 출력(계속).

```
... 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

```

예 2: 또한 동일한 테이블 함수는 각 파티션의 서비스 클래스에서 실행하는 코디네이터 활동의 평균 동시성에 대해 가장 높은 값을 제공할 수도 있습니다.

```

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(CAST(NULL AS VARCHAR
(128)),
      CAST(NULL AS VARCHAR
(128)), -2)) AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, PART

```

다음은 이 쿼리로부터의 출력 예입니다.

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

이 테이블 함수의 출력은 특정 데이터베이스에 대해 각 파티션에서 상위 레벨 뷰의 "로드"를 제공합니다. 출력은 평균 실행 시간 및 활동 수 검사에서 나옵니다. 이러한 테이블 함수에서 리턴된 상위 레벨 값의 중요한 변형은 시스템에 로드 변경을 나타낼 수 있습니다.

## 사용 시 참고사항

해당 서비스 서브클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA 및 COLLECT AGGREGATE REQUEST DATA 설정이 "NONE"이 아닌 값으로 설정 되면 일부 통계만이 리턴됩니다.

WLM\_GET\_SERVICE\_SUBCLASS\_STATS 테이블 함수는 서비스 서브클래스 및 파티션당 하나의 데이터 행을 리턴합니다. 서비스 클래스(파티션) 또는 파티션(서비스 클래스 이상)에 집계는 없습니다. 그러나 위 예에서처럼 SQL 쿼리를 통해 집계를 수행할 수 있습니다.

매개변수는 "AND"절이 되는 효과가 있습니다. 즉, 충돌하는 입력 매개변수(예: 수퍼 클래스 SUPA 및 수퍼 클래스 SUBB)를 지정하면 SUBB가 SUPA의 서브클래스가 아니며 행이 리턴되지 않습니다.

## 리턴되는 정보

표 292. WLM\_GET\_SERVICE\_SUBCLASS\_STATS에 대해 리턴된 정보

컬럼 이름	데이터 유형	설명
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	이 레코드가 수집된 서비스 수퍼 클래스의 이름.
SERVICE_SUBCLASS_NAME	VARCHAR (128)	이 레코드가 수집된 서비스 서브클래스의 이름.
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호입니다.
LAST_RESET	TIMESTAMP	<p>통계가 마지막으로 재설정된 시간입니다. 통계의 재 설정을 트리거하는 일이 발생할 수 있는 4개의 이벤 트가 있으며, 이 이벤트들이 이 시간소인을 갱신합니 다.</p> <ul style="list-style-type: none"> <li>• WLM_COLLECT_STATS 프로시저가 호출됩니다.</li> <li>• WLM_COLLECT_INT 구성 매개변수로 제어되는 주기적인 콜렉션 및 재설정 프로세스로 인해 콜렉 션과 재설정이 수행됩니다.</li> <li>• 데이터베이스가 다시 활성화됩니다.</li> <li>• 보고 중인 통계에 대한 서비스 서브클래스가 수정 되며 변경이 커밋됩니다.</li> </ul> <p>LAST_RESET 시간소인이 로컬 시간에 있습니다.</p>
COORD_ACT_COMPLETED_TOTAL	BIGINT	<p>마지막 재설정 및 완료된 후 사용자가 제출한 총 코 디네이터의 활동 수. 이 계수는 각 활동이 완료될 때 마다 갱신됩니다.</p> <p>활동을 다른 서비스 서브클래스에 다시 맵핑하면 활 동은 완료된 서브클래스의 총계만을 계산합니다.</p>
COORD_ACT_ABORTED_TOTAL	BIGINT	<p>마지막 재설정 및 오류로 완료된 후 사용자가 제출한 총 코디네이터의 활동 수. 이 계수는 각 활동이 중단 될 때마다 갱신됩니다.</p> <p>활동을 다른 서비스 서브클래스에 다시 맵핑하면 활 동은 중단된 서브클래스의 총계만을 계산합니다.</p>
COORD_ACT_REJECTED_TOTAL	BIGINT	<p>사용자가 마지막 재설정 이후 제출하여 실행하도록 허 용되지 않고 실행 전에 거부된 전체 코디네이터 활동 수. 활동은 실행 금지 작업 조치 또는 예측적 임계값 으로 인해 실행되지 않은 경우 거부됨으로 간주됩니다. 이 계수는 각 활동이 거부될 때마다 갱신됩니다.</p>
CONCURRENT_ACT_TOP	INTEGER	이 서비스 서브클래스에 도달한 실행(유휴 및 대기 중 포함) 또는 큐에 대기됨 상태의 중첩 레벨에서 가장 높은 동시 활동 수

표 292. WLM\_GET\_SERVICE\_SUBCLASS\_STATS에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
COORD_ACT_LIFETIME_TOP	BIGINT	<p>모든 중첩 레벨에서 계산된 코디네이터 활동 수명의 상위 워터 마크(water mark)입니다. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA가 NONE인 경우 널(NULL)입니다. 단위는 밀리초입니다.</p> <p>서비스 클래스가 맵핑된 서브클래스를 포함할 경우 이 통계를 효율적으로 사용하려면, 동일한 리맵핑 임계값 또는 임계값들에 의해 영향을 받는 다른 서브클래스의 워터 마크(water mark)와 함께 서비스 서브클래스의 COORD_ACT_LIFETIME_TOP 상위 워터 마크(water mark)를 집계해야 합니다. 다른 서비스 서브클래스로 다시 맵핑된 후에 활동을 완료할 수 있기 때문에 이들 값을 집계해야 하지만, 다시 맵핑되기 전에 다른 서비스 서브클래스에서 활동이 보낸 시간은 완료된 서비스 클래스에 대해서만 계산됩니다.</p>
COORD_ACT_LIFETIME_AVG	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 중첩 레벨 0에서의 코디네이터에 대한 수명의 산술 평균. 내부에서 추적된 평균이 오버플로우된 경우 값 -2가 리턴됩니다. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA가 NONE일 경우 널(NULL)입니다. 단위는 밀리초입니다.</p> <p>서비스 서브클래스의 COORD_ACT_LIFETIME_AVG는 서브클래스를 통해 전달되는 활동에는 영향을 받지 않지만 완료되기 전에 다른 서브클래스에 다시 맵핑됩니다.</p>
COORD_ACT_LIFETIME_STDDEV	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 중첩 레벨 0에서의 코디네이터에 대한 수명의 표준 편차. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA가 NONE일 경우 널(NULL)입니다. 단위는 밀리초입니다. 이 표준 편차는 코디네이터의 활동 수명 막대 그래프로 계산되며 막대 그래프가 데이터에 적합하게 제대로 크기 조정되지 않은 경우에는 정확하지 않을 수도 있습니다. 값이 마지막 막대 그래프 바이너리로 떨어지면 -1의 값이 리턴됩니다.</p> <p>서비스 서브클래스의 COORD_ACT_LIFETIME_STDDEV는 서비스 서브클래스에서는 영향을 받지 않지만 완료되기 전에 다른 서브클래스에 다시 맵핑됩니다.</p>

표 292. WLM\_GET\_SERVICE\_SUBCLASS\_STATS에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
COORD_ACT_EXEC_TIME_AVG	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 중첩 레벨 0에서의 코디네이터 활동에 대한 실행 시간의 산술 평균. 내부에서 추적된 평균이 오버플로우된 경우 값 -2가 리턴됩니다. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA가 NONE일 경우 널(NULL)입니다. 단위는 밀리초입니다.</p> <p>서비스 서브클래스의 실행 시간 평균은 서브클래스를 통해 전달되는 활동에는 영향을 받지 않지만 완료되기 전에 다른 서브클래스에 다시 맵핑됩니다.</p>
COORD_ACT_EXEC_TIME_STDDEV	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 중첩 레벨 0에서의 코디네이터 활동에 대한 실행 시간의 표준 편차. 단위는 밀리초입니다. 막대 그래프의 크기가 데이터에 맞게 제대로 되어 있지 않다면 이 표준 편차는 코디네이터 활동 실행 막대 그래프로 계산되어 정확하지 않을 수 있습니다. 값이 마지막 막대 그래프 바이너리로 떨어지면 -1의 값이 리턴됩니다.</p> <p>서비스 서브클래스의 실행 시간 표준 편차는 서브클래스를 통해 전달되는 활동에는 영향을 받지 않지만 완료되기 전에 다른 서브클래스에 다시 맵핑됩니다.</p>
COORD_ACT_QUEUE_TIME_AVG	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 중첩 레벨 0에서의 코디네이터 활동에 대한 큐 시간의 산술 평균. 내부에서 추적된 평균이 오버플로우된 경우 값 -2가 리턴됩니다. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA가 NONE일 경우 널(NULL)입니다. 단위는 밀리초입니다.</p> <p>큐 시간 평균은 활동이 큐에 대기되는 서비스 서브클래스에 대해서만 계산됩니다.</p>
COORD_ACT_QUEUE_TIME_STDDEV	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 중첩 레벨 0에서의 코디네이터 활동에 대한 큐의 표준 편차. 서비스 클래스의 COLLECT AGGREGATE ACTIVITY DATA가 NONE일 경우 널(NULL)입니다. 단위는 밀리초입니다. 이 표준 편차는 코디네이터의 활동 큐 시간 막대 그래프로 계산되며 막대 그래프가 데이터에 적합하게 제대로 크기 조정되지 않은 경우에는 정확하지 않을 수도 있습니다. 값이 마지막 막대 그래프 바이너리로 떨어지면 -1의 값이 리턴됩니다.</p> <p>큐 시간 표준 편차는 활동이 큐에 대기되는 서비스 서브클래스에 대해서만 계산됩니다.</p>
NUM_REQUESTS_ACTIVE	BIGINT	이 테이블 함수가 실행되는 서비스 서브클래스에서 실행 중인 요청 수.

표 292. WLM\_GET\_SERVICE\_SUBCLASS\_STATS에 대해 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
NUM_REQUESTS_TOTAL	BIGINT	<p>마지막 재설정 이후 이 서비스 서브클래스에서 실행 완료 요청 수. 이는 활동의 구성원에 상관 없이 요청을 적용합니다. 이 서비스 서브클래스에서 COLLECT AGGREGATE REQUEST DATA가 NONE으로 설정되어 있으면, 이 컬럼의 값은 NULL입니다.</p> <p>서비스 서브클래스의 NUM_REQUESTS_TOTAL은 서비스 서브클래스를 통해 전달되는 요청에는 영향을 받지 않지만, 완료되지 않습니다.</p>
REQUEST_EXEC_TIME_AVG	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 요청에 대한 실행 시간의 산술 평균. 단위는 밀리초입니다. 내부에서 추적된 평균이 오버플로우된 경우 값 -2가 리턴됩니다. 이 서비스 클래스에서 COLLECT AGGREGATE REQUEST DATA가 NONE으로 설정되어 있으면, 이 컬럼의 값은 NULL입니다.</p> <p>서비스 서브클래스의 실행 시간 평균은 서브클래스를 통해 전달되는 요청에 의해 영향을 받지 않지만 완료되지 않습니다.</p>
REQUEST_EXEC_TIME_STDDEV	DOUBLE	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 요청에 대한 실행 시간의 표준 편차. 단위는 밀리초입니다. 이 서비스 클래스에서 COLLECT AGGREGATE REQUEST DATA가 NONE으로 설정되어 있으면, 이 컬럼의 값은 NULL입니다. 이 표준 편차는 요청 실행 시간 막대 그래프로 계산되며 막대 그래프가 데이터에 적합하게 제대로 크기 조정되지 않은 경우에는 정확하지 않을 수도 있습니다. 값이 마지막 막대 그래프 바이너리로 떨어지면 -1의 값이 리턴됩니다.</p> <p>서비스 서브클래스의 실행 시간 표준 편차는 서브클래스를 통해 전달되는 요청에 의해 영향을 받지 않지만 완료되지 않습니다.</p>
REQUEST_EXEC_TIME_TOTAL	BIGINT	<p>마지막 재설정 이후 이 서비스 서브클래스와 연관된 요청에 대한 실행 시간의 총합. 단위는 밀리초입니다. 이 서비스 클래스에서 COLLECT AGGREGATE REQUEST DATA가 NONE으로 설정되어 있으면, 이 컬럼의 값은 NULL입니다. 이 총계는 요청 실행 시간 막대 그래프로 계산되며 막대 그래프가 데이터에 적합하게 제대로 크기 조정되지 않은 경우에는 정확하지 않을 수도 있습니다. 값이 마지막 막대 그래프 바이너리로 떨어지면 -1의 값이 리턴됩니다.</p> <p>서비스 서브클래스의 전체 실행 시간은 서브클래스를 통해 전달되는 요청에 의해 영향을 받지 않지만 완료되지 않습니다.</p>

## WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES - 활동 목록 리턴

WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 함수는 지정한 파티션에서 지정한 응용프로그램으로 제출되어 아직 완료되지 않은 전체 활동 목록을 리턴합니다.

주: 이 테이블 함수는 사용되지 않으며

909 페이지의 『WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97 - 활동 목록 리턴』으로 교체됩니다.

### 구문

```
▶▶ WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES (—application_handle—, —————▶  
▶ dbpartitionnum—) —————▶▶
```

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### application\_handle

활동 목록이 리턴되는 응용프로그램 핸들을 지정하는 BIGINT 유형의 입력 인수입니다. 인수가 널(NULL)인 경우 데이터베이스의 모든 응용프로그램에 대해 데이터가 검색됩니다.

#### dbpartitionnum

현재 연결된 데이터베이스와 동일한 인스턴스에 있는 유효한 파티션 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션으로 1을 지정하거나 모든 데이터베이스 파티션으로 -2를 지정하십시오. 널(NULL) 값이 지정된 경우, -1이 내재적으로 설정됩니다.

### 권한 부여

WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 함수에 대한 EXECUTE 특권

### 예 :

응용프로그램 핸들을 식별한 후에 현재 이 응용프로그램에서 실행 중인 모든 활동을 검색할 수 있습니다. 예를 들어, 관리자가 LIST APPLICATIONS 명령으로 판별된 응용프로그램 핸들이 1인 응용프로그램 활동을 나열하려는 경우 관리자는 다음 쿼리를 실행합니다.

```
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(1, -2)) AS WLOACTS
ORDER BY PART, UOWID, ACTID

```

쿼리의 샘플 출력은 다음과 같습니다.

```

COORD PART UOWID ACTID PARUOWID PARACTID ACTTYPE NESTING
-----
0 0 2 3 - - CALL 0
0 0 2 5 2 3 READ_DML 1

```

### 리턴되는 정보

표 293. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES에서 리턴된 정보

컬럼 이름	데이터 유형	설명
APPLICATION_HANDLE	BIGINT	응용프로그램에 대해 시스템 전체에서 사용되는 고유 ID. 단일 파티션의 데이터베이스에서 이 ID는 16비트의 카운터로 구성됩니다. 다중 파티션의 데이터베이스에서 이 ID는 16비트 카운터로 병합된 조정 파티션 번호로 구성됩니다. 또한 이 ID는 응용프로그램이 보조 연결을 작성하는 모든 파티션에서 동일합니다.
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호입니다.
COORD_PARTITION_NUM	SMALLINT	활동의 코디네이터 파티션
LOCAL_START_TIME	TIMESTAMP	이 활동이 파티션에서 작업을 시작한 로컬 시간. 시스템이 활동 상태이지만 아직 큐에 있고 실행은 시작되지 않은 경우에 컬럼 값은 널(NULL)입니다.
UOW_ID	INTEGER	활동이 시작된 작업 단위(UOW)의 고유 ID.
ACTIVITY_ID	INTEGER	작업 단위(UOW)내의 고유 활동 ID
PARENT_UOW_ID	INTEGER	활동의 상위 활동이 시작된 작업 단위(UOW)의 고유 ID. 활동에 상위 없거나 리모트 파티션에 있는 경우 이 컬럼 값은 널(NULL)입니다.
PARENT_ACTIVITY_ID	INTEGER	ID가 ACTIVITY_ID 컬럼 값인 활동 상위의 작업 단위(UOW)내에서 고유 활동 ID. 활동에 상위 없거나 리모트 파티션에 있는 경우 이 컬럼 값은 널(NULL)입니다.

표 293. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
ACTIVITY_STATE	VARCHAR(32)	<p>활동 상태. 가능한 값은 다음과 같습니다.</p> <p><i>CANCEL_PENDING</i>                      활동 요청에 대해 실제로 작동 중인 에이전트가 없기 때문에 활동이 취소되었습니다. 활동 중에 다음에 요청이 제출될 때 활동이 취소되고 SQL4725N 오류가 생성됩니다.</p> <p><i>EXECUTING</i>                      활동 요청에 대해 실제로 에이전트가 작동 중입니다.</p> <p><i>IDLE</i>    활동 요청을 실제로 처리 중인 에이전트가 없습니다.</p> <p><i>INITIALIZING</i>                      활동이 제출되었지만 아직 실행이 시작되지 않았습니다. 초기화 상태에서 예측적 임계값이 활동에 적용되어 활동 실행 허용 여부가 결정됩니다.</p>

표 293. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
ACTIVITY_STATE(계속)	VARCHAR(32)	<p>활동 상태. 가능한 값은 다음과 같습니다.</p> <p><i>QP_CANCEL_PENDING</i> 이 상태는 활동이 WLM_CANCEL_ACTIVITY 프로시저가 아닌 Query Patroller로 취소되었다는 점을 제외하고는 CANCEL_PENDING 상태와 동일합니다.</p> <p><i>QP_QUEUED</i> Query Patroller가 활동을 큐에 대기시켰습니다.</p> <p><i>QUEUED</i> 워크로드 관리 큐 대기 임계값으로 활동이 큐에 대기되었습니다. 파티션된 데이터베이스 환경에서, 이 상태는 임계값 티켓을 가져오기 위해 코디네이터 에이전트가 카탈로그 파티션에 대한 RPC를 작성했지만 아직 응답을 수신하지 않았음을 나타냅니다. 이 상태는 워크로드 관리 큐 임계값으로 활동이 큐에 대기되었음을 나타내거나 그리 오래 시간이 경과되지 않은 경우에는 활동이 해당 티켓을 가져오는 프로세스 중임을 나타낼 수 있습니다. 활동이 큐에 대기되어 있는지 정확하게 알려면 활동에서 작업 중인 에이전트를 판별하고 카탈로그 파티션에서 에이전트의 EVENT_OBJECT 값이 WLM_QUEUE인지 확인하십시오.</p> <p><i>TERMINATING</i> 활동의 실행이 완료되어 시스템에서 제거 중입니다.</p>

표 293. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
ACTIVITY_TYPE	VARCHAR(32)	<p>활동 유형. 가능한 값은 다음과 같습니다.</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> <p>각 활동 유형에 연관된 SQL문의 다른 유형에 대한 설명은 <i>Workload Manager Guide and Reference</i>의 『작업 클래스로 작업 유형 식별』의 내용을 참조하십시오.</p>
NESTING_LEVEL	INTEGER	이 활동이 최상위 활동에 중첩된 깊이입니다.
INVOCATION_ID	INTEGER	이 활동의 특정 호출을 동일한 중첩 레벨의 다른 호출과 구별하는 호출 ID.
ROUTINE_ID	INTEGER	루틴의 고유 ID
UTILITY_ID	INTEGER	<p>다음 값 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• 활동이 유틸리티인 경우 값은 유틸리티 ID입니다.</li> <li>• 활동이 유틸리티가 아닌 경우 값은 널(NULL)입니다.</li> </ul>
SERVICE_CLASS_ID	INTEGER	이 활동이 속한 서비스 클래스의 고유 ID
DATABASE_WORK_ACTION_SET_ID	INTEGER	<p>다음 값 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• 이 활동이 데이터베이스 범위의 작업 클래스로 구분된 경우 값은 이 작업 클래스가 구성원인 작업 클래스 세트의 ID입니다.</li> <li>• 이 활동이 데이터베이스 범위의 작업 클래스로 구분되지 않은 경우 값은 널(NULL)입니다.</li> </ul>
DATABASE_WORK_CLASS_ID	INTEGER	<p>다음 값 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• 이 활동이 데이터베이스 범위의 작업 클래스로 구분된 경우 값은 작업 클래스 ID입니다.</li> <li>• 이 활동이 데이터베이스 범위의 작업 클래스로 구분되지 않은 경우 값은 널(NULL)입니다.</li> </ul>
SERVICE_CLASS_WORK_ACTION_SET_ID	INTEGER	<p>다음 값 중 하나입니다.</p> <ul style="list-style-type: none"> <li>• 이 활동이 서비스 클래스 범위의 작업 클래스로 구분된 경우 값은 작업 클래스가 속한 작업 클래스 세트와 연관된 작업 조치 세트의 ID입니다.</li> <li>• 이 활동이 서비스 클래스 범위의 작업 클래스로 구분되지 않은 경우 값은 널(NULL)입니다.</li> </ul>

표 293. WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
SERVICE_CLASS_WORK_CLASS_ID	INTEGER	다음 값 중 하나입니다. <ul style="list-style-type: none"> <li>이 활동이 서비스 클래스 범위의 작업 클래스로 구분된 경우 값은 이 활동에 지정된 작업 클래스의 ID입니다.</li> <li>이 활동이 서비스 클래스 범위의 작업 클래스로 구분되지 않은 경우 값은 널(NULL)입니다.</li> </ul>

## WLM\_GET\_WORKLOAD\_STATS - 워크로드 통계 리턴

주: 이 테이블 함수는 사용되지 않으며

915 페이지의 『WLM\_GET\_WORKLOAD\_STATS\_V97 테이블 함수 - 워크로드 통계 리턴』으로 교체됩니다.

이 함수는 워크로드 이름과 데이터베이스 파티션 번호의 모든 조합에 대한 워크로드 통계를 리턴합니다.

### 구문

►►—WLM\_GET\_WORKLOAD\_STATS—(—workload\_name—,—dbpartitionnum—)—————►►

스키마는 SYSPROC입니다.

### 테이블 함수 매개변수

#### *workload\_name*

통계가 리턴되는 특정 워크로드를 지정하는 VARCHAR

(128) 유형의 입력 인수입니다. 인수가 NULL 또는 비어 있는 문자열인 경우 모든 워크로드에 대한 통계가 리턴됩니다.

#### *dbpartitionnum*

이 함수 호출 시 현재 연결된 데이터베이스와 동일한 인스턴스에서 유효한 파티션 번호를 지정하는 INTEGER 유형의 입력 인수. 현재 데이터베이스 파티션으로 1을 지정하거나 모든 데이터베이스 파티션으로 -2를 지정하십시오. 널(NULL) 값이 지정되면 -1이 내재적으로 설정됩니다.

### 권한 부여

WLM\_GET\_WORKLOAD\_STATS 함수에 대한 EXECUTE 특권

예 :

관리자가 워크로드에 대한 통계를 검토하려고 할 수도 있습니다. 다음과 같은 쿼리를 사용하여 수행할 수 있습니다.

```
SELECT SUBSTR(WORKLOAD_NAME,1,22) AS WL_DEF_NAME,
       SUBSTR(CHAR(DBPARTITIONNUM),1,4) AS PART,
       CONCURRENT_WLO_TOP AS WLO_TOP,
       CONCURRENT_WLO_ACT_TOP AS WLO_ACT_TOP
FROM TABLE(WLM_GET_WORKLOAD_STATS(CAST(NULL AS VARCHAR
(128)), -2))
AS WLSTATS
ORDER BY WL_DEF_NAME, PART
```

다음은 이 쿼리의 출력 예입니다.

WL_DEF_NAME	PART	WLO_TOP	WLO_ACT_TOP
MYUSERWORKLOAD	0	2	8
MYUSERWORKLOAD	1	0	0
SYSDEFAULTUSERWORKLOAD	0	1	1
SYSDEFAULTUSERWORKLOAD	1	0	0

여기에서 파티션 0에서 MYUSERWORKLOAD 워크로드의 동시 어커런스의 가장 높은 수가 2이고 이 워크로드 어커런스 중 하나의 동시 활동의 가장 높은 수가 8임을 알 수 있습니다.

## 사용법 참고

이 함수는 워크로드 이름과 데이터베이스 파티션 번호의 모든 조합에 대해 한 개의 행을 리턴합니다. 워크로드간, 파티션간 또는 서비스 클래스간의 집계는 수행되지 않습니다. 그렇지만 SQL 쿼리를 사용하여 집계를 수행할 수 있습니다.

## 리턴되는 정보

표 294. WLM\_GET\_WORKLOAD\_STATS에서 리턴된 정보

컬럼 이름	데이터 유형	설명
WORKLOAD_NAME	VARCHAR (128)	이 레코드가 수집된 워크로드의 이름.
DBPARTITIONNUM	SMALLINT	이 레코드가 수집된 파티션 번호입니다.

표 294. WLM\_GET\_WORKLOAD\_STATS에서 리턴된 정보 (계속)

컬럼 이름	데이터 유형	설명
LAST_RESET	TIMESTAMP	<p>통계가 마지막으로 재설정된 시간입니다. 통계의 재설정을 트리거하는 일이 발생할 수 있는 4개의 이벤트가 있으며, 이 이벤트들이 이 시간소인을 갱신합니다.</p> <ul style="list-style-type: none"> <li>• WLM_COLLECT_STATS 프로시저가 호출됩니다.</li> <li>• WLM_COLLECT_INT 구성 매개변수로 제어되는 주기적인 콜렉션 및 재설정 프로세스로 인해 콜렉션과 재설정이 수행됩니다.</li> <li>• 데이터베이스가 다시 활성화됩니다.</li> <li>• 통계가 보고 중인 워크로드가 수정되어 변경사항이 커밋되었습니다.</li> </ul> <p>LAST_RESET 시간소인은 로컬 시간입니다.</p>
CONCURRENT_WLO_TOP	INTEGER	마지막으로 재설정된 이후 이 파티션에서 지정된 워크로드에 대한 가장 높은 동시 어커런스 수입니다.
CONCURRENT_WLO_ACT_TOP	INTEGER	마지막 재설정 이후에 이 워크로드에 대한 모든 어커런스에 도달한 실행(유휴 및 대기 포함) 또는 큐에 대기된 상태의 가장 높은 동시 활동(코디네이터 및 중첩 포함) 수입니다. 해당 작업 단위(UOW) 종료 시점에 각 워크로드 어커런스로 갱신됩니다.
COORD_ACT_COMPLETED_TOTAL	BIGINT	마지막 재설정 이후에 완료된 이 워크로드의 임의 어커런스에 지정된 모든 중첩 레벨에서 전체 코디네이터 활동 수입니다. 해당 작업 단위(UOW) 종료 시점에 각 워크로드 어커런스로 갱신됩니다.
COORD_ACT_ABORTED_TOTAL	BIGINT	마지막 재설정 이후에 완료 이전에 중단된 이 워크로드의 임의 어커런스에 지정된 모든 중첩 레벨에서 전체 코디네이터 활동 수입니다. 해당 작업 단위(UOW) 종료 시점에 각 워크로드 어커런스로 갱신됩니다.
COORD_ACT_REJECTED_TOTAL	BIGINT	<p>마지막 재설정 이후에 실행 이전에 거부된 이 워크로드의 임의 어커런스에 지정된 모든 중첩 레벨에서 전체 코디네이터 활동 수입니다. 해당 작업 단위(UOW) 종료 시점에 각 워크로드 어커런스로 갱신됩니다. 활동은 실행 금지 작업 조치 또는 예측적 임계값으로 인해 실행되지 않은 경우 거부됨으로 간주됩니다. WLM_GET_SERVICE_SUBCLASS_STATS 함수에서 동일한 이름의 컬럼과는 달리 활동을 서비스 클래스에 지정하기 전에 발생하는 거부도 계산됩니다. 활동이 ConcurrentWorkloadOccurrences 임계값을 위반할 때 거부가 발생하는 것을 예로 들 수 있습니다.</p>
WLO_COMPLETED_TOTAL	BIGINT	마지막 재설정 이후에 완료된 워크로드 어커런스 수입니다.





---

## 부록 A. DB2 기술 정보 개요

DB2 기술 정보는 다음 도구 및 메소드를 통해 사용할 수 있습니다.

- DB2 정보 센터
  - 주제 항목(태스크, 개념 및 참조 항목)
  - DB2 도구에 대한 도움말
  - 샘플 프로그램
  - 자습서
- DB2 서적
  - PDF 파일(다운로드)
  - PDF 파일(DB2 PDF DVD)
  - 인쇄된 서적
- 명령행 도움말
  - 명령 도움말
  - 메시지 도움말

주: DB2 정보 센터 주제는 PDF 또는 하드카피 서적보다 자주 갱신됩니다. 최신 정보를 보려면 사용 가능한 문서 갱신사항을 설치하거나 [ibm.com](http://ibm.com)에서 DB2 정보 센터를 참조하십시오.

[ibm.com](http://www.ibm.com)에서 추가 DB2 기술 정보(예: 기술 노트, 백서 및 IBM Redbooks® 서적)를 온라인으로 액세스할 수 있습니다. DB2 정보 관리 라이브러리 소프트웨어 사이트 <http://www.ibm.com/software/data/sw-library/>에 액세스하십시오.

### 문서 피드백

DB2 문서에 대한 피드백을 환영합니다. DB2 문서를 향상시키는 방법에 대해서 제안 사항이 있는 경우 [db2docs@ca.ibm.com](mailto:db2docs@ca.ibm.com)으로 전자 우편을 보내십시오. DB2 문서 팀에서는 고객의 모든 피드백을 읽지만 직접 응답할 수는 없습니다. 고객의 문제를 더 잘 이해할 수 있도록 가능한 위치에 특정 예를 제공하십시오. 특정 주제 또는 도움말 파일에 대한 피드백을 보내실 경우, 제목 및 URL을 알려주십시오.

DB2 고객 지원에 문의할 때 이 전자 우편 주소를 사용하지 마십시오. 문서에서 해결할 수 없는 DB2 기술 문제점이 있는 경우, 해당 지역의 IBM 서비스 센터에 도움을 요청하십시오.

## DB2 기술 라이브러리(하드카피 또는 PDF 형식)

다음 표는 IBM Publications Center([www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order))에서 사용할 수 있는 DB2 라이브러리에 대해 설명합니다. PDF 형식의 영문 DB2 버전 9.7 매뉴얼 및 번역된 버전은 [www.ibm.com/support/docview.wss?rs=71&uid=swg2700947](http://www.ibm.com/support/docview.wss?rs=71&uid=swg2700947)에서 다운로드할 수 있습니다.

표에 인쇄할 수 있는 책이 나와 있는 경우에도, 사용 국가 또는 지역에서 해당 책을 사용할 수 없을 수도 있습니다.

매뉴얼이 갱신될 때마다 문서 번호가 증가합니다. 다음 사항을 참조하여 읽고 있는 매뉴얼이 최신 버전인지 확인하십시오.

주: DB2 정보 센터는 PDF 또는 하드카피 서적보다 자주 갱신됩니다.

표 295. DB2 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
관리 API 참조서	SA30-3958-00	예	2009년 8월
관리 루틴 및 뷰	SA30-3955-00	아니오	2009년 8월
<i>Call Level Interface Guide and Reference, Volume 1</i>	SC27-2437-00	예	2009년 8월
<i>Call Level Interface Guide and Reference, Volume 2</i>	SC27-2438-00	예	2009년 8월
명령어 참조서	SA30-3959-00	예	2009년 8월
데이터 이동 유틸리티 안내서 및 참조서	SA30-3969-00	예	2009년 8월
데이터 복구 및 고가용성 안내서 및 참조서	SA30-3970-00	예	2009년 8월
데이터베이스 관리 개념 및 구성 참조서	SA30-3951-00	예	2009년 8월
데이터베이스 모니터링 안내서 및 참조서	SA30-3953-00	예	2009년 8월
데이터베이스 보안 안내서	SA30-3971-00	예	2009년 8월
<i>DB2 Text Search Guide</i>	SC27-2459-00	예	2009년 8월
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-2444-00	예	2009년 8월
<i>Developing Embedded SQL Applications</i>	SC27-2445-00	예	2009년 8월
<i>Developing Java Applications</i>	SC27-2446-00	예	2009년 8월
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-2447-00	아니오	2009년 8월

표 295. DB2 기술 정보 (계속)

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>Developing User-defined Routines (SQL and External)</i>	SC27-2448-00	예	2009년 8월
<i>Getting Started with Database Application Development</i>	GI11-9410-00	예	2009년 8월
Linux 및 Windows에서 DB2 설치 및 관리 시작하기	GA30-3960-00	예	2009년 8월
자국어 안내서	SA30-3972-00	예	2009년 8월
DB2 Server 설치	GA30-3962-00	예	2009년 8월
IBM Data Server Clients 설치	GA30-3963-00	아니오	2009년 8월
<i>Message Reference Volume 1</i>	SC27-2450-00	아니오	2009년 8월
<i>Message Reference Volume 2</i>	SC27-2451-00	아니오	2009년 8월
<i>Net Search Extender Administration and User's Guide</i>	SC27-2469-00	아니오	2009년 8월
파티셔닝 및 클러스터링 안내서	SA30-3973-00	예	2009년 8월
<i>pureXML Guide</i>	SC27-2465-00	예	2009년 8월
Query Patroller 관리 및 사용자 안내서	SA30-3974-00	아니오	2009년 8월
<i>Spatial Extender and Geodetic Data Management Feature User's Guide and Reference</i>	SC27-2468-00	아니오	2009년 8월
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-2470-00	예	2009년 8월
SQL 참조서, 볼륨 1	SA30-3956-00	예	2009년 8월
SQL 참조서, 볼륨 2	SA30-3957-00	예	2009년 8월
문제점 해결 및 데이터베이스 성능 조정	SA30-3952-00	예	2009년 8월
DB2 버전 9.7로 업그레이드	SA30-3961-00	예	2009년 8월
Visual Explain 자습서	SA30-3968-00	아니오	2009년 8월
DB2 버전 9.7의 새로운 내용	SA30-3967-00	예	2009년 8월
<i>Workload Manager Guide and Reference</i>	SC27-2464-00	예	2009년 8월

표 295. DB2 기술 정보 (계속)

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>XQuery Reference</i>	SC27-2466-00	아니오	2009년 8월

표 296. DB2 Connect 특정 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>DB2 Connect Personal Edition 설치 및 구성</i>	SA30-3965-00	예	2009년 8월
<i>DB2 Connect Server 설치 및 구성</i>	SA30-3966-00	예	2009년 8월
<i>DB2 Connect 사용자 안내서</i>	SA30-3964-00	예	2009년 8월

표 297. Information Integration 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>Information Integration: Administration Guide for Federated Systems</i>	SC19-1020-02	예	2009년 8월
<i>Information Integration: ASNCLP Program Reference for Replication and Event Publishing</i>	SC19-1018-04	예	2009년 8월
<i>Information Integration: Configuration Guide for Federated Data Sources</i>	SC19-1034-02	아니오	2009년 8월
<i>Information Integration: SQL Replication Guide and Reference</i>	SC19-1030-02	예	2009년 8월
<i>Information Integration: Introduction to Replication and Event Publishing</i>	GC19-1028-02	예	2009년 8월

## 인쇄된 DB2 서적 주문

인쇄된 DB2 서적이 필요한 경우, 대부분 온라인으로 구매할 수 있으나 모든 국가 또는 지역에 해당되지는 않습니다. 언제든지 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문할 수 있습니다. *DB2 PDF* 문서 DVD의 일부 소프트웨어 서적은 인쇄할 수 없다는 점에 유의하십시오. 예를 들면, *DB2 메시지 참조서*의 어떤 볼륨도 인쇄된 서적으로 사용할 수 없습니다.

DB2 PDF 문서 DVD에서 사용할 수 있는 다수의 DB2 서적의 인쇄된 버전은 IBM에서 유료로 주문할 수 있습니다. 주문하는 위치에 따라 IBM Publications Center에서 온라인으로 서적을 주문할 수도 있습니다. 해당 국가 또는 지역에서 온라인 주문이

불가능하면, 언제든지 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문할 수 있습니다. DB2 PDF 문서 DVD의 모든 서적을 인쇄할 수는 없다는 점에 유의하십시오.

주: 가장 최신 및 완료된 DB2 문서는 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7>의 DB2 정보 센터에서 유지보수됩니다.

인쇄된 DB2 서적을 주문하려면 다음을 수행하십시오.

- 해당 국가 또는 지역에서 인쇄된 DB2 서적을 온라인으로 주문할 수 있는지 여부를 확인하려면 <http://www.ibm.com/shop/publications/order>의 IBM Publications Center를 확인하십시오. 서적 주문 정보에 액세스하려면 국가/지역/언어를 선택한 다음 해당 위치에서 주문 지시사항을 따르십시오.
- 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문하려면 다음을 수행하십시오.
  1. 다음 웹 사이트 중 하나에서 해당 지역 담당자에 대한 문의처 정보를 찾으십시오.
    - [www.ibm.com/planetwide](http://www.ibm.com/planetwide)에 있는 IBM 월드 와이드 문의처 디렉토리
    - <http://www.ibm.com/shop/publications/order>의 IBM Publications 웹 사이트. 사용 지역의 해당 서적 홈 페이지에 액세스하려면 해당 국가, 지역 또는 언어를 선택해야 합니다. 이 페이지에서 "이 제품의 정보" 링크를 수행하십시오.
  2. 전화로 주문할 경우, 주문할 DB2 서적을 지정하십시오.
  3. 담당자에게 주문하려는 서적의 제목 및 문서 번호를 제공하십시오. 서적의 제목 및 문서 번호는 1182 페이지의 『DB2 기술 라이브러리(하드카피 또는 PDF 형식)』를 참조하십시오.

---

## 명령행 처리기에서 SQL 상태 도움말 표시

DB2 제품은 SQL문의 결과로 나타나는 상태에 대한 SQLSTATE 값을 리턴합니다. SQLSTATE 도움말은 SQL 상태 및 SQL 상태 클래스 코드의 의미를 설명합니다.

SQL 상태 도움말을 시작하려면 명령행 처리기를 열고 다음을 입력하십시오.

```
? sqlstate or ? class code
```

여기서, *sqlstate*는 유효한 5자리 숫자로 된 SQL 상태이고 *class code*는 SQL 상태의 처음 2자리 숫자를 나타냅니다.

예를 들어, ? 08003은 08003 SQL 상태에 대한 도움말을 표시하고, ? 08은 08 클래스 코드에 대한 도움말을 표시합니다.

---

## DB2 정보 센터의 다른 버전에 액세스

DB2 버전 9.7 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>입니다.

DB2 버전 9.5 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>입니다.

DB2 버전 9 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>입니다.

DB2 버전 8 주제에 대한 버전 8 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>입니다.

---

## DB2 정보 센터에서 원하는 언어로 항목 표시

DB2 정보 센터는 브라우저 환경 설정에 지정된 언어로 주제 항목을 표시합니다. 주제가 원하는 언어로 변환되지 않은 경우, DB2 정보 센터는 영어로 주제 항목을 표시합니다.

- Internet Explorer 브라우저에서 원하는 언어로 항목을 표시하려면 다음을 수행하십시오.

1. Internet Explorer에서 도구 → 인터넷 옵션 → 언어... 단추를 누르십시오. 언어 환경 설정 창이 열립니다.
2. 원하는 언어가 언어 목록의 첫 번째 항목으로 지정되었는지 확인하십시오.
  - 목록에 새 언어를 추가하려면 추가... 단추를 누르십시오.

주: 언어를 추가했다고 컴퓨터에 원하는 언어로 항목을 표시하는 데 필요한 글꼴이 설치되는 것은 아닙니다.

- 언어를 목록 맨위로 이동하려면, 언어를 선택한 후 언어가 언어 목록의 첫 번째 항목이 될 때까지 위로 이동 단추를 누르십시오.
3. 브라우저 캐시를 지운 후 페이지를 새로 고쳐 원하는 언어로 DB2 정보 센터를 표시하십시오.

- Firefox 또는 Mozilla 브라우저에서 원하는 언어로 주제 항목을 표시하려면 다음을 수행하십시오.

1. 도구 → 옵션 → 고급 대화 상자의 언어 섹션에서 단추를 선택하십시오. 환경 설정 창에 언어 패널이 표시됩니다.
2. 원하는 언어가 언어 목록의 첫 번째 항목으로 지정되었는지 확인하십시오.
  - 목록에 새 언어를 추가하려면 추가... 단추를 눌러 언어 추가 창에서 언어를 선택합니다.

- 언어를 목록 맨위로 이동하려면, 언어를 선택한 후 언어가 언어 목록의 첫 번째 항목이 될 때까지 위로 이동 단추를 누르십시오.
- 3. 브라우저 캐시를 지운 후 페이지를 새로 고쳐 원하는 언어로 DB2 정보 센터를 표시하십시오.

일부 브라우저 및 운영 체제 조합에서 운영 체제의 국가별 설정을 선택한 로케일 및 언어로 변경해야 합니다.

---

## 컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 갱신

로컬로 설치된 DB2 정보 센터는 주기적으로 갱신해야 합니다.

### 시작하기 전에

DB2 버전 9.7 정보 센터는 등록된 상태여야 합니다. 자세한 내용은 *DB2 Server* 설치의 『DB2 설치 마법사를 사용하여 DB2 정보 센터 설치』 주제를 참조하십시오. 정보 센터 설치에 적용되는 모든 전제조건 및 제한사항은 정보 센터 갱신에도 적용됩니다.

### 이 태스크에 대한 정보

기존의 DB2 정보 센터는 자동 또는 수동으로 갱신할 수 있습니다.

- 자동 갱신 - 기존 정보 센터 기능 및 언어를 갱신합니다. 자동 갱신의 또 다른 이점으로는 갱신 동안 정보 센터를 아주 잠시 동안만 사용 불가능하다는 점입니다. 또한 자동 갱신은 주기적으로 실행되는 기타 일괄처리 작업의 일부로 실행되도록 설정 가능합니다.
- 수동 갱신 - 갱신 프로세스 중에 기능이나 언어를 추가하려는 경우 사용하십시오. 예를 들어, 로컬 정보 센터는 기본적으로 영어와 프랑스어로 설치되어 있으며 기존 정보 센터의 기능 및 언어 갱신 외에도 수동 갱신으로 독어도 설치할 수 있습니다. 단, 수동 갱신을 수행하려면 정보 센터를 중지한 다음 갱신하고 재시작해야 합니다. 정보 센터는 갱신 프로세스 동안에는 사용할 수 없습니다.

### 프로시저

이 주제는 자동 갱신 프로세스에 대한 설명입니다. 수동 갱신에 대한 지시사항은 『컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신』 주제를 참조하십시오.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터를 자동으로 갱신하려면 다음을 수행하십시오.

1. Linux 운영 체제에서,
  - a. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 `/opt/ibm/db2ic/V9.7` 디렉토리에 디폴트로 설치됩니다.
  - b. 설치 디렉토리에서 `doc/bin` 디렉토리까지 탐색하십시오.

c. 다음과 같이 ic-update 스크립트를 실행하십시오.

```
ic-update
```

2. Windows 운영 체제에서,

a. 명령 창을 여십시오.

b. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 <Program Files>\IBM\DB2 Information Center\Version 9.7 디렉토리에 디폴트로 설치됩니다. 여기서 <Program Files>는 프로그램 파일 디렉토리의 위치를 나타냅니다.

c. 설치 디렉토리에서 doc\bin 디렉토리까지 탐색하십시오.

d. 다음과 같이 ic-update.bat 파일을 실행하십시오.

```
ic-update.bat
```

### 결과

DB2 정보 센터가 자동으로 재시작됩니다. 갱신사항이 사용 가능한 경우, 정보 센터에는 새로 갱신된 주제가 표시됩니다. 정보 센터 갱신을 사용할 수 없는 경우, 메시지가 로그에 추가됩니다. 로그 파일은 doc\weclipse\configuration 디렉토리에 있습니다. 이 로그 파일 이름은 임의로 생성된 번호입니다. (예: 1239053440785.log).

---

## 컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신

DB2 정보 센터를 로컬로 설치한 경우, IBM으로부터 문서 갱신사항을 받아 설치할 수 있습니다.

로컬로 설치된 DB2 정보 센터를 수동으로 갱신하려면 다음을 수행하십시오.

1. 컴퓨터에서 DB2 정보 센터를 중지한 후 독립형 모드에서 다시 시작하십시오. 독립형 모드에서 정보 센터를 실행하면 사용자의 네트워크와 연결된 다른 사용자는 정보 센터에 액세스할 수 없으므로 갱신사항을 적용할 수 있습니다. DB2 정보 센터의 워크스테이션 버전은 항상 독립형 모드에서 실행됩니다. .
2. 어떤 갱신사항이 사용 가능한지 확인하려면 갱신 기능을 사용하십시오. 설치해야 할 갱신사항이 있는 경우, 갱신 기능을 사용하여 이를 가져온 후 설치할 수 있습니다.

주: 인터넷에 연결되지 않은 머신에 DB2 정보 센터 갱신사항을 설치해야 할 경우, 인터넷에 연결된 머신을 사용하여 갱신 사이트를 로컬 파일 시스템에 미러해야 DB2 정보 센터가 설치됩니다. 네트워크 상에 문서 갱신사항을 설치하려는 사용자가 많을 경우에는 갱신 사이트를 로컬로 미러링하거나 갱신 사이트의 프록시를 작성하여 갱신을 수행하면 각 개인에게 필요한 시간을 줄일 수 있습니다.

갱신 패키지가 사용 가능하면 갱신 기능을 사용하여 패키지를 가져오십시오. 그러나 갱신 기능은 독립형 모드에서만 사용할 수 있습니다.

3. 독립형 정보 센터를 중지한 후 컴퓨터에서 DB2 정보 센터를 재시작하십시오.



주: Windows 2008, Windows Vista 이상의 경우 이 섹션 다음에 나오는 명령은 관리자로 실행해야 합니다. 전체 관리자 권한으로 명령 프롬프트 또는 그래픽 도구를 열려면 단축키를 마우스 오른쪽 단추로 누른 후 관리자로 실행을 선택하십시오.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터를 갱신하려면 다음을 수행하십시오.

1. DB2 정보 센터를 중지하십시오.

- Windows에서는 시작 → 제어판 → 관리 도구 → 서비스를 누르십시오. 그런 다음 **DB2 정보 센터** 서비스를 마우스 오른쪽 단추로 누른 후 중지를 선택하십시오.

- Linux에서는 다음 명령을 입력하십시오.

```
/etc/init.d/db2icdv97 stop
```

2. 독립형 모드에서 정보 센터를 시작하십시오.

- Windows 사용자:

- a. 명령 창을 여십시오.

- b. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 <Program Files>\IBM\DB2 Information Center\Version 9.7 디렉토리에 디폴트로 설치됩니다. 여기서 <Program Files>는 프로그램 파일 디렉토리의 위치를 나타냅니다.

- c. 설치 디렉토리에서 doc\bin 디렉토리까지 탐색하십시오.

- d. 다음과 같이 help\_start.bat 파일을 실행하십시오.

```
help_start.bat
```

- Linux 사용자:


- a. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 /opt/ibm/db2ic/V9.7 디렉토리에 디폴트로 설치됩니다.

- b. 설치 디렉토리에서 doc/bin 디렉토리까지 탐색하십시오.

- c. 다음과 같이 help\_start 스크립트를 실행하십시오.

```
help_start
```

독립형 정보 센터를 표시하기 위해 시스템의 기본 웹 브라우저가 열립니다.

3. 갱신 단추()를 누르십시오. (JavaScript™가 브라우저에서 사용 가능해야 합니다.) 정보 센터의 오른쪽 패널에서 갱신사항 찾기를 누르십시오. 기존 문서의 갱신사항 목록이 표시됩니다.

4. 설치 프로세스를 시작하려면 설치할 선택란을 체크한 후 갱신사항 설치를 누르십시오.

5. 설치 프로세스가 완료되면 완료를 누르십시오.

6. 독립형 정보 센터를 중지하십시오.

- Windows에서 설치 디렉토리의 doc\bin 디렉토리를 탐색한 후 다음과 같이 help\_end.bat 파일을 실행하십시오.

```
help_end.bat
```

주: help\_end 일괄처리 파일에는 help\_start 일괄처리 파일로 시작된 프로세스를 안전하게 중지하는 데 필요한 명령이 포함되어 있습니다. help\_start.bat 를 중지하는 데 Ctrl-C 또는 다른 메소드를 사용하지 마십시오.

- Linux에서 설치 디렉토리의 doc/bin 디렉토리를 탐색한 후 다음과 같이 help\_end 스크립트를 실행하십시오.

```
help_end
```

주: help\_end 스크립트에는 help\_start 스크립트로 시작된 프로세스를 안전하게 중지하는 데 필요한 명령이 포함되어 있습니다. help\_start 스크립트를 중지하는 데 다른 메소드를 사용하지 마십시오.

#### 7. DB2 정보 센터를 재시작하십시오.

- Windows에서는 시작 → 제어판 → 관리 도구 → 서비스를 누르십시오. 그런 다음 **DB2 정보 센터 서비스**를 마우스 오른쪽 단추로 누른 후 시작을 선택하십시오.
- Linux에서는 다음 명령을 입력하십시오.

```
/etc/init.d/db2icdv97 start
```

갱신된 DB2 정보 센터에는 새로 갱신된 주제가 표시됩니다.

## DB2 자습서

DB2 자습서는 DB2 제품의 다양한 측면에 대해 학습하는 데 유용합니다. 각 레슨은 단계별 지시사항을 제공합니다.

### 시작하기 전에

<http://publib.boulder.ibm.com/infocenter/db2help/>의 정보 센터에서 XHTML 버전의 자습서를 볼 수 있습니다.

일부 레슨에서는 샘플 데이터나 코드를 사용합니다. 특정 태스크에 대한 전제조건 설명은 자습서를 참조하십시오.

### DB2 자습서

자습서를 보려면 제목을 누르십시오.

#### 『pureXML』(pureXML Guide)

XML 데이터를 저장하고 원시 XML 데이터 스토어로 기본 조작을 수행하려면 DB2 데이터베이스를 설정하십시오.

## Visual Explain 자습서의 『Visual Explain』

Visual Explain을 사용하여 성능을 향상시킬 수 있도록 SQL문을 분석, 최적화 및 조정합니다.

---

## DB2 문제점 해결 정보

DB2 데이터베이스 제품을 사용하는 데 도움이 되는 광범위한 문제를 해결하고 판별할 수 있는 정보가 있습니다.

### DB2 문서

문제점 해결 정보는 *DB2 문제점 해결 안내서* 또는 *DB2 정보 센터*의 데이터베이스 기본 섹션을 참조하십시오. DB2 진단 도구 및 유틸리티를 사용하여 문제점을 찾아내고 식별하는 방법, 가장 일반적인 문제점에 대한 솔루션 및 DB2 데이터베이스 제품에서 발생할 수 있는 문제점을 해결하는 방법 등의 정보가 있습니다.

### DB2 기술 지원 웹 사이트

문제점이 있는 경우 원인 및 솔루션을 찾으려면 DB2 기술 지원 웹 사이트를 참조하십시오. 기술 지원 사이트에는 최신 DB2 서적, 기술 노트, APAR(Authorized Program Analysis Report 또는 버그 수정), FixPack 및 기타 자원에 대한 링크가 있습니다. 이러한 기술 자료를 검색하여 문제에 대한 가능한 솔루션을 찾을 수 있습니다.

[http://www.ibm.com/software/data/db2/support/db2\\_9/](http://www.ibm.com/software/data/db2/support/db2_9/)에서 DB2 기술 지원 웹 사이트에 액세스하십시오.

---

## 이용약관

다음 조건에 따라 이 책을 사용할 수 있습니다.

**개인적 사용:** 모든 소유권 사항을 표시하는 경우에 한하여 귀하는 본 문서를 개인적, 비상업적 용도로 복제할 수 있습니다. 귀하는 IBM의 명시적 동의 없이 본 문서 또는 그 일부를 배포 또는 전시하거나 2차적 저작물을 만들 수 없습니다.

**상업적 사용:** 모든 소유권 사항을 표시하는 경우에 한하여 귀하는 본 문서를 귀하 사업장 내에서만 복제, 배포 및 전시할 수 있습니다. 귀하는 IBM의 명시적 동의 없이 본 문서의 2차적 저작물을 만들거나 본 문서 또는 그 일부를 복제, 배포 또는 전시할 수 없습니다.

본 허가에서 명시적으로 부여된 경우를 제외하고, 이 책이나 이 책에 포함된 정보, 데이터, 소프트웨어 또는 기타 지적 재산권에 대한 어떠한 허가나 라이선스 또는 권한도 명시적 또는 묵시적으로 부여되지 않습니다.

IBM은 본 문서의 사용이 IBM의 이익을 해친다고 판단되거나 위에서 언급된 지시사항이 준수되지 않는다고 판단하는 경우 언제든지 이 사이트에서 부여한 허가를 철회할 수 있습니다.

귀하는 미국 수출법 및 관련 규정을 포함하여 모든 적용 가능한 법률 및 규정을 철저히 준수하는 경우에만 본 정보를 다운로드, 송신 또는 재송신할 수 있습니다.

IBM은 본 문서의 내용에 대해 어떠한 보증도 제공하지 않습니다. 타인의 권리 침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 현 상태대로 제공합니다.

---

## 부록 B. 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. 비IBM 제품에 대한 정보는 이 책을 처음 발행할 때의 정보에 기초하고 있으며 변경될 수 있습니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2바이트 문자 세트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan, Ltd.

3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증없이 이 책을 『현상 그대로』 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책 사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독자적으로 작성된 프로그램과 다른 프로그램(본 프로그램 포함) 간의 정보 교환 및  
(ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등) 하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 측정치는 개발 레벨 시스템에서 작성되었을 수 있으며, 따라서 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 다른 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

#### 저작권 라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 되는 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 IBM에 추가 비용을 지불하지 않고 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다. 샘플 프로그램은 어떠한 보증없이 "있는 그대로" 제공됩니다. IBM은 샘플 프로그램의 사용으로 인해 발생하는 모든 손해에 대해 책임을 지지 않습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다.

© (귀하의 회사명) (연도). 이 코드의 일부는 IBM Corp.의 샘플 프로그램에서 파생됩니다. © Copyright IBM Corp. *\_enter 연도\_*. All rights reserved.

#### 상표

IBM, IBM 로고 및 [ibm.com](http://ibm.com)<sup>®</sup>은 여러 국가에 등록된 International Business Machines Corp.의 상표 또는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 기타 회사의 상표입니다. 현재 IBM 상표 목록은 웹 "저작권 및 상표 정보"([www.ibm.com/legal/kr/copytrade.shtml](http://www.ibm.com/legal/kr/copytrade.shtml))에 있습니다.

다음 표장은 기타 회사의 상표 또는 등록상표입니다.

- Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.
- Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.
- UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.
- Intel<sup>®</sup>, Intel 로고, Intel Inside<sup>®</sup>, Intel Inside 로고, Intel<sup>®</sup> Centrino<sup>®</sup>, Intel Centrino 로고, Celeron<sup>®</sup>, Intel<sup>®</sup> Xeon<sup>®</sup>, Intel SpeedStep<sup>®</sup>, Itanium<sup>®</sup> 및 Pentium<sup>®</sup>은 미국 또는 기타 국가에서 사용되는 Intel Corporation의 상표 또는 등록상표입니다.
- Microsoft<sup>®</sup>, Windows, Windows NT<sup>®</sup> 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.





---

# 색인

## [ 가 ]

### 갱신사항

DB2 정보 센터 1187, 1188

### 관리 루틴

AUTH\_LIST\_ROLES\_FOR\_AUTHID 484

AUTOMAINT\_SET\_POLICY 프로시저 299

### 관리 보기

권한 부여 1

대 테이블 함수 2

ADMINTABCOMPRESSINFO 251

ADMIN\_TASK\_LIST 283

ADMIN\_TASK\_STATUS 286

BP\_READ\_IO 499

BP\_WRITE\_IO 501

CONTACTGROUPS 950

CONTACTS 951

CONTAINER\_UTILIZATION 503

DBCFCG 350

DBPATHS 957

ENV\_FEATURE\_INFO 357

ENV\_INST\_INFO 358

ENV\_PROD\_INFO 360

ENV\_SYS\_INFO 361

ENV\_SYS\_RESOURCES 362

LOG\_UTILIZATION 513

LONG\_RUNNING\_SQL 514

NOTIFICATIONLIST 970

PDLOGMSG\_LAST24HOURS 978

QUERY\_PREP\_COST 517

REG\_VARIABLES 355

SNAPAGENT\_MEMORY\_POOL 522, 671

SNAPBP 543, 692

SNAPBP\_PART 548, 697

SNAPCONTAINER 552, 701

SNAPDB 557, 706

SNAPDBM\_MEMORY\_POOL 577, 726

SNAPDB\_MEMORY\_POOL 568, 717

SNAPDETAILLOG 581, 730

SNAPDYN\_SQL 584, 733

SNAPFCM\_PART 592, 741

SNAPHADR 595, 744

SNAPSTMT 612, 761

SNAPSTORAGE\_PATHS 618

### 관리 보기 (계속)

SNAPSUBSECTION 621, 770

SNAPTAB 629, 778

SNAPTBSPPART 644

SNAPTbsp\_RANGE 654, 805

SNAPUTIL 658, 809

SNAPUTIL\_PROGRESS 662, 813

TOP\_DYNAMIC\_SQL 822

### 관리 뷰

개요 1

지원되는 3

ADMINTABINFO 260

ADMINTEMPCOLUMNS 268

ADMINTEMPTABLES 272

APPLICATIONS 492

APPL\_PERFORMANCE 491

AUTHORIZATIONIDS 487

BP\_HITRATIO 497

DBMCFG 352

DB\_HISTORY 952

LOCKS\_HELD 505

LOCKWAIT 508

OBJECTOWNERS 488

PRIVILEGES 489

SNAPAGENT 518, 667

SNAPAPPL 535, 684

SNAPAPPL\_INFO 526, 675

SNAPDBM 572, 721

SNAPFCM 590, 739

SNAPLOCK 600, 749

SNAPLOCKWAIT 606, 755

SNAPSTORAGE\_PATHS 767

SNAPSWITCHES 625, 774

SNAPTAB\_REORG 632, 782

SNAPTbsp 638, 788

SNAPTBSPPART 794

SNAPTbsp QUIESCER 649, 800

Tbsp\_UTILIZATION 818

### 관리 태스크 스케줄러

태스크 스케줄 정의 281

### 관리 SQL 루틴

지원되는 3

### 구성 매개변수

가져오기 316

구성 매개변수 (계속)

SET\_CONFIG 프로시저로 설정 338

권한 부여

관리 뷰용 1

권한 부여 ID 검색 487

그룹 멤버십 검색 중 483

그룹

그룹 멤버십 검색 중 483

## [ 다 ]

데이터 유효성 다시 확인

프로시저 248

데이터 재분배

프로시저 833, 835, 836, 839, 841

데이터베이스 경로

검색 957

데이터베이스 관리 프로그램 구성 매개변수

값 검색 352

데이터베이스 구성 매개변수

검색 350

도움말

언어 구성 1186

SQL문 1185

## [ 라 ]

레지스트리 변수

사용 중인 설정값 검색 355

루틴

모니터 367

SQL 관리 995

지원되는 3

## [ 마 ]

명령

프로시저에서 호출 35, 849

ADD CONTACT 37

ADD CONTACTGROUP 39

AUTOCONFIGURE 40

BACKUP DATABASE 44

DESCRIBE 53

DROP CONTACT 68

DROP CONTACTGROUP 69

EXPORT 70

FORCE APPLICATION 82

GET STMM TUNING DBPARTITIONNUM 84

명령 (계속)

IMPORT 85

INITIALIZE TAPE 113

LOAD 114

PRUNE HISTORY/LOGFILE 156

QUIESCE DATABASE 158

QUIESCE TABLESPACES FOR TABLE 160

REDISTRIBUTE DATABASE PARTITION GROUP 163

REORG INDEXES/TABLE 175

RESET ALERT CONFIGURATION 190

RESET DATABASE CONFIGURATION 192

RESET DATABASE MANAGER CONFIGURATION 194

REWIND TAPE 195

RUNSTATS 196

SET TAPE POSITION 210

UNQUIESCE DATABASE 211

UPDATE ALERT CONFIGURATION 212

UPDATE CONTACT 219

UPDATE CONTACTGROUP 220

UPDATE DATABASE CONFIGURATION 221

UPDATE DATABASE MANAGER CONFIGURATION 225

UPDATE HEALTH NOTIFICATION CONTACT LIST 227

UPDATE HISTORY 228

UPDATE STMM TUNING DBPARTITIONNUM 231

모니터

루틴 367

문서

개요 1181

이용약관 1191

인쇄됨 1182

PDF 1182

문의처

문의처 그룹 검색 950

문의처 목록 검색 951

문의처 목록

문의처 검색 951

문의처 그룹 목록 검색 950

문제점 판별

사용 가능 정보 1191

자습서 1191

통지 로그 메시지 978

문제점 해결

온라인 정보 1191

자습서 1191



사용되지 않는 기능 (계속)

테이블 함수 (계속)

SNAPSHOT\_LOCK 1129  
 SNAPSHOT\_LOCKWAIT 1130  
 SNAPSHOT\_QUIESCERS 1132  
 SNAPSHOT\_RANGES 1133  
 SNAPSHOT\_STATEMENT 1134  
 SNAPSHOT\_SUBSECT 1136  
 SNAPSHOT\_SWITCHES 1137  
 SNAPSHOT\_TABLE 1139  
 SNAPSHOT\_TBREORG 1140  
 SNAPSHOT\_TBS 1141  
 SNAPSHOT\_TBS\_CFG 1143  
 SNAPSTORAGE\_PATHS 767  
 SNAP\_GET\_APP 1056  
 SNAP\_GET\_BP 1069  
 SNAP\_GET\_CONTAINER 1073  
 SNAP\_GET\_DB 1074  
 SNAP\_GET\_DYN\_SQL 1097  
 SNAP\_GET\_STO\_PATHS 1099  
 SNAP\_GET\_TAB 1100  
 SNAP\_GET\_TBSP\_PART 1104  
 SNAP\_GET\_TBSP\_PART\_V91 794  
 SQLCACHE\_SNAPSHOT 1146

프로시저

GET\_DB\_CONFIG 1009  
 HEALTH\_CONT\_HI 1011  
 HEALTH\_CONT\_HI\_HIS 1013  
 HEALTH\_CONT\_INFO 1015  
 HEALTH\_DBM\_HI 1031  
 HEALTH\_DBM\_HI\_HIS 1033  
 HEALTH\_DBM\_INFO 1035  
 HEALTH\_DB\_HI 1017  
 HEALTH\_DB\_HIC 1025  
 HEALTH\_DB\_HIC\_HIS 1027  
 HEALTH\_DB\_HI\_HIS 1021  
 HEALTH\_DB\_INFO 1030  
 HEALTH\_GET\_ALERT\_ACTION\_CFG 1037  
 HEALTH\_GET\_ALERT\_CFG 1040  
 HEALTH\_GET\_IND\_DEFINITION 1043  
 HEALTH\_HI\_REC 1046  
 HEALTH\_TBS\_HI 1047  
 HEALTH\_TBS\_HI\_HIS 1050  
 HEALTH\_TBS\_INFO 1054  
 SNAPSHOT\_FILEW 1128  
 SYSINSTALLROUTINES 1147  
 WLM\_GET\_SERVICE\_SUBCLASS\_STATS 1165  
 WLM\_GET\_WORKLOAD\_STATS 1177

사용되지 않는 기능 (계속)

SQL 관리 루틴 995

사용되지 않는 함수

테이블 함수

SNAP\_GET\_DBM 1080  
 SNAP\_GET\_DB\_V91 1083  
 SNAP\_GET\_TBSP 1101

삭제

스키마 및 해당 오브젝트 927

서적

인쇄됨

주문 1184

설치

DB2 제품 라이선스 정보 리턴 357, 362

DB2 제품 정보 검색 360

스칼라 함수

SQLERRM 989

스키마

스키마와 오브젝트 복사 923

오브젝트 삭제 927

스토리지 관리 도구

스토어드 프로시저 845, 846, 847

스토어드 프로시저

AUDIT\_ARCHIVE 293

AUDIT\_DELIM\_EXTRACT 294

시스템 정보

가져오기 331

검색 361

실행기록 파일

정보 검색 중 952

## [ 아 ]

오류 메시지

검색

SQLERRM 스칼라 함수 989

오브젝트

오브젝트 소유권 검색 488

온라인 테이블 이동 930, 942

완료 모드

일반 SQL API 306

이용약관

서적 사용 1191

인스턴스

현재 인스턴스 정보 검색 358

일반 SQL API 303

서명 303

완료 모드 306

일반 SQL API (계속)

- 출력 필터링 307
- XML 메시지 문서 308
- XML 입력 문서 305
- XML 출력 306

일반 SQL API 스토어드 프로시저

- 버전화 304

## [ 자 ]

자습서

- 문제점 판별 1191
- 문제점 해결 1191
- Visual Explain 1190

주의사항 1193

지원되는 함수 3

## [ 타 ]

테이블

- 압축 정보 251
- 온라인 이동 930, 942
- 임시 테이블 정보 검색 중 272
- 임시 테이블의 컬럼 정보 검색 중 268
- 크기와 상태 검색 260, 997

테이블 함수

- 대 관리 뷰 2
- 사용되지 않는 기능 995
  - ADMIN\_GET\_TAB\_INFO 997
  - SNAP\_GET\_BP 1069
- 사용되지 않는 함수
  - SNAP\_GET\_APPL\_INFO 1062
  - SNAP\_GET\_DBM 1080
  - SNAP\_GET\_DB\_V91 1083
  - SNAP\_GET\_DYN\_SQL\_V91 1093

지원되는 3

- admin\_get\_dbp\_mem\_usage 234
- ADMIN\_GET\_INDEX\_COMPRESS\_INFO 테이블 함수 236
- ADMIN\_GET\_INDEX\_INFO 테이블 함수 240
- ADMIN\_GET\_MSGS 244
- ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 251
- ADMIN\_GET\_TAB\_INFO\_V97 260
- ADMIN\_GET\_TEMP\_COLUMNS 268
- ADMIN\_GET\_TEMP\_TABLES 272
- AUDIT\_ARCHIVE 293
- AUTH\_LIST\_GROUPS\_FOR\_AUTHID 483
- HEALTH\_GET\_ALERT\_ACTION\_CFG 1037
- HEALTH\_GET\_ALERT\_CFG 1040

테이블 함수 (계속)

- HEALTH\_GET\_IND\_DEFINITION 1043
- PD\_GET\_DIAG\_HIST 971
- PD\_GET\_LOG\_MSGS 978
- SNAP\_GET\_AGENT 518, 667
- SNAP\_GET\_AGENT\_MEMORY\_POOL 522, 671
- SNAP\_GET\_APPL\_INFO\_V95 526, 675
- SNAP\_GET\_APPL\_V95 535, 684
- SNAP\_GET\_BP\_PART 548, 697
- SNAP\_GET\_BP\_V95 543, 692
- SNAP\_GET\_CONTAINER\_V91 552, 701
- SNAP\_GET\_DBM\_MEMORY\_POOL 577, 726
- SNAP\_GET\_DBM\_V95 572, 721
- SNAP\_GET\_DB\_MEMORY\_POOL 568, 717
- SNAP\_GET\_DB\_V95 557, 706
- SNAP\_GET\_DETAIL\_LOG\_V91 581, 730
- SNAP\_GET\_DYN\_SQL\_V95 584, 733
- SNAP\_GET\_FCM 590, 739
- SNAP\_GET\_FCM\_PART 592, 741
- SNAP\_GET\_HADR 595, 744
- SNAP\_GET\_LOCK 600, 749
- SNAP\_GET\_LOCKWAIT 606, 755
- SNAP\_GET\_STMT 612, 761
- SNAP\_GET\_STORAGE\_PATHS 767
- SNAP\_GET\_STORAGE\_PATHS\_V97 618
- SNAP\_GET\_SUBSECTION 621, 770
- SNAP\_GET\_SWITCHES 625, 774
- SNAP\_GET\_TAB\_REORG 632, 782
- SNAP\_GET\_TAB\_V91 629, 778
- SNAP\_GET\_TBSP\_PART\_V91 794
- SNAP\_GET\_TBSP\_PART\_V97 644
- SNAP\_GET\_TBSP QUIESCER 649, 800
- SNAP\_GET\_TBSP\_RANGE 654, 805
- SNAP\_GET\_TBSP\_V91 638, 788
- SNAP\_GET\_UTIL 658, 809
- SNAP\_GET\_UTIL\_PROGRESS 662, 813

통지 로그 메시지

- 검색 978

통지 목록

- 문의처 목록 검색 970

특권

- 권한 부여에 대한 정보
  - PRIVILEGES 관리 뷰 489

## [ 파 ]

패키지

- 리바인드 829

## 프로시저

사용되지 않는 기능 995  
일반 SQL API 303, 307  
ADMIN\_CMD  
  설명 35  
ADMIN\_COPY\_SCHEMA 923  
ADMIN\_DROP\_SCHEMA 927  
ADMIN\_MOVE\_TABLE 930  
ADMIN\_MOVE\_TABLE\_UTIL 942  
ADMIN\_REMOVE\_MSGS 247  
ADMIN\_REVALIDATE\_DB\_OBJECTS 248  
ADMIN\_TASK\_ADD 277  
ADMIN\_TASK\_REMOVE 285  
ADMIN\_TASK\_UPDATE 288  
ALTER\_ROUTINE\_PACKAGE 825  
ALTOBJ 944  
AM\_DROP\_TASK 19  
AM\_GET\_LOCK\_CHNS 21  
AM\_GET\_LOCK\_CHN\_TB 20  
AM\_GET\_LOCK\_RPT 22  
AM\_GET\_RPT 30  
AM\_SAVE\_TASK 32  
ANALYZE\_LOG\_SPACE 833  
AUDIT\_ARCHIVE 293  
AUDIT\_DELIM\_EXTRACT 294  
AUTOMAINT\_GET\_POLICY 297  
AUTOMAINT\_GET\_POLICYFILE 298  
AUTOMAINT\_SET\_POLICYFILE 301  
CANCEL\_WORK 309  
CAPTURE\_STORAGE\_MGMT\_INFO 845  
CREATE\_STORAGE\_MGMT\_TABLES 846  
DROP\_STORAGE\_MGMT\_TABLES 847  
EVMON\_FORMAT\_UE\_TO\_TABLES 368  
GENERATE\_DISTFILE 835  
GET\_CONFIG 316  
GET\_DBSIZE\_INFO 967  
GET\_MESSAGE 323  
GET\_ROUTINE\_SAR 826  
GET\_SWRD\_SETTINGS 836  
GET\_SYSTEM\_INFO 331  
HEALTH\_HI\_REC 1046  
PUT\_ROUTINE\_SAR 828  
REBIND\_ROUTINE\_PACKAGE 829  
REORGCHK\_IX\_STATS 985  
REORGCHK\_TB\_STATS 987  
SET\_CONFIG 338  
SET\_ROUTINE\_OPTS 831  
SET\_SWRD\_SETTINGS 839

## 프로시저 (계속)

SNAPSHOT\_FILEW 1128  
SNAP\_WRITE\_FILE 666, 816  
STEPWISE\_REDISTRIBUTE\_DBPG 841  
summary 3  
SYSINSTALLOBJECTS 991  
SYSINSTALLROUTINES 1147  
SYSTS\_ADMIN\_CMD 849  
WLM\_CANCEL\_ACTIVITY 877  
WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS 878  
WLM\_COLLECT\_STATS 880  
WLM\_SET\_CLIENT\_INFO 919

## [ 하 ]

### 함수

#### 스칼라

APPLICATION\_ID 947  
GET\_ROUTINE\_OPTS 826  
MQPUBLISH 457  
MQREAD 459  
MQREADCLOB 464  
MQRECEIVE 465  
MQRECEIVECLOB 471  
MQSEND 473  
MQSUBSCRIBE 474  
MQUNSUBSCRIBE 476  
SQLERRM 989

#### 스칼라 함수

EXPLAIN\_FORMAT\_STATS 959

#### 스토어드 프로시저

SYSTS\_ALTER 850  
SYSTS\_CLEAR\_COMMANDLOCKS 854  
SYSTS\_CLEAR\_EVENTS 857  
SYSTS\_DROP 869  
SYSTS\_ENABLE 872  
SYSTS\_UPDATE 873

#### 지원되는 3

#### 테이블

WLM\_GET\_SERVICE\_CLASS\_AGENTS 1154  
WLM\_GET\_WORKLOAD\_STATS\_V97 915

#### 테이블 함수 1

대 관리 뷰 2

사용되지 않는 995

#### 지원되는 3

ADMIN\_GET\_MSGS 244  
ADMIN\_GET\_TAB\_COMPRESS\_INFO 1004  
ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 251

함수 (계속)

테이블 함수 (계속)

ADMIN\_GET\_TAB\_INFO 997  
ADMIN\_GET\_TAB\_INFO\_V97 260  
ADMIN\_GET\_TEMP\_COLUMNS 268  
ADMIN\_GET\_TEMP\_TABLES 272  
AM\_BASE\_RPTS 18  
AM\_BASE\_RPT\_RECOMS 17  
AUDIT\_ARCHIVE 293  
AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID 479  
AUTH\_LIST\_GROUPS\_FOR\_AUTHID 483  
COMPILATION\_ENV 948  
DB\_PARTITIONS 349  
EVMON\_FORMAT\_UE\_TO\_XML 376  
EXPLAIN\_GET\_MSGS 965  
GET\_DBM\_CONFIG 1010  
GET\_DB\_CONFIG 1009  
HEALTH\_CONT\_HI 1011  
HEALTH\_CONT\_HI\_HIS 1013  
HEALTH\_CONT\_INFO 1015  
HEALTH\_DBM\_HI 1031  
HEALTH\_DBM\_HI\_HIS 1033  
HEALTH\_DBM\_INFO 1035  
HEALTH\_DB\_HI 1017  
HEALTH\_DB\_HIC 1025  
HEALTH\_DB\_HIC\_HIS 1027  
HEALTH\_DB\_HI\_HIS 1021  
HEALTH\_DB\_INFO 1030  
HEALTH\_GET\_ALERT\_ACTION\_CFG 1037  
HEALTH\_GET\_ALERT\_CFG 1040  
HEALTH\_GET\_IND\_DEFINITION 1043  
HEALTH\_TBS\_HI 1047  
HEALTH\_TBS\_HI\_HIS 1050  
HEALTH\_TBS\_INFO 1054  
MON\_GET\_ACTIVITY\_DETAILS 379  
MON\_GET\_BUFFERPOOL 388  
MON\_GET\_CONNECTION 391  
MON\_GET\_CONNECTION\_DETAILS 396  
MON\_GET\_CONTAINER 402  
MON\_GET\_EXTENT\_MOVEMENT\_STATUS 405  
MON\_GET\_INDEX 406  
MON\_GET\_PKG\_CACHE\_STMT 409  
MON\_GET\_SERVICE\_SUBCLASS 414  
MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 420  
MON\_GET\_TABLE 426  
MON\_GET\_TABLESPACE 429  
MON\_GET\_UNIT\_OF\_WORK 433  
MON\_GET\_UNIT\_OF\_WORK\_DETAILS 438

함수 (계속)

테이블 함수 (계속)

MON\_GET\_WORKLOAD 445  
MON\_GET\_WORKLOAD\_DETAILS 450  
MQREADALL 460  
MQREADALLCLOB 462  
MQRECEIVEALL 467  
MQRECEIVEALLCLOB 469  
PD\_GET\_DIAG\_HIST 971  
PD\_GET\_LOG\_MSGS 978  
SNAPSHOT\_AGENT(사용되지 않음) 1106  
SNAPSHOT\_APPL(사용되지 않음) 1107  
SNAPSHOT\_APPL\_INFO(사용되지 않음) 1112  
SNAPSHOT\_BP(사용되지 않음) 1114  
SNAPSHOT\_CONTAINER(사용되지 않음) 1116  
SNAPSHOT\_DATABASE(사용되지 않음) 1117  
SNAPSHOT\_DBM(사용되지 않음) 1123  
SNAPSHOT\_DYN\_SQL(사용되지 않음) 1124  
SNAPSHOT\_FCMNODE(사용되지 않음) 1127  
SNAPSHOT\_FCM(사용되지 않음) 1126  
SNAPSHOT\_LOCKWAIT(사용되지 않음) 1130  
SNAPSHOT\_LOCK(사용되지 않음) 1129  
SNAPSHOT QUIESCERS(사용되지 않음) 1132  
SNAPSHOT\_RANGES(사용되지 않음) 1133  
SNAPSHOT\_STATEMENT(사용되지 않음) 1134  
SNAPSHOT\_SUBSECT(사용되지 않음) 1136  
SNAPSHOT\_SWITCHES(사용되지 않음) 1137  
SNAPSHOT\_TABLE(사용되지 않음) 1139  
SNAPSHOT\_TBREORG(사용되지 않음) 1140  
SNAPSHOT\_TBS(사용되지 않음) 1141  
SNAPSHOT\_TBS\_CFG(사용되지 않음) 1143  
SNAP\_GET\_AGENT 518, 667  
SNAP\_GET\_AGENT\_MEMORY\_POOL 522, 671  
SNAP\_GET\_APPL 1056  
SNAP\_GET\_APPL\_INFO 1062  
SNAP\_GET\_APPL\_INFO\_V95 526, 675  
SNAP\_GET\_APPL\_V95 535, 684  
SNAP\_GET\_BP 1069  
SNAP\_GET\_BP\_PART 548, 697  
SNAP\_GET\_BP\_V95 543, 692  
SNAP\_GET\_CONTAINER(사용되지 않음) 1073  
SNAP\_GET\_CONTAINER\_V91 552, 701  
SNAP\_GET\_DBM 1080  
SNAP\_GET\_DBM\_MEMORY\_POOL 577, 726  
SNAP\_GET\_DBM\_V95 572, 721  
SNAP\_GET\_DB(사용되지 않음) 1074  
SNAP\_GET\_DB\_MEMORY\_POOL 568, 717  
SNAP\_GET\_DB\_V91 1083

함수 (계속)

테이블 함수 (계속)

SNAP\_GET\_DB\_V95 557, 706  
 SNAP\_GET\_DETAIL\_LOG\_V91 581, 730  
 SNAP\_GET\_DYN\_SQL(사용되지 않음) 1097  
 SNAP\_GET\_DYN\_SQL\_V91 1093  
 SNAP\_GET\_DYN\_SQL\_V95 584, 733  
 SNAP\_GET\_FCM 590, 739  
 SNAP\_GET\_FCM\_PART 592, 741  
 SNAP\_GET\_HADR 595, 744  
 SNAP\_GET\_LOCK 600, 749  
 SNAP\_GET\_LOCKWAIT 606, 755  
 SNAP\_GET\_STMT 612, 761  
 SNAP\_GET\_STORAGE\_PATHS 767  
 SNAP\_GET\_STORAGE\_PATHS\_V97 618  
 SNAP\_GET\_STO\_PATHS(사용되지 않음) 1099  
 SNAP\_GET\_SUBSECTION 621, 770  
 SNAP\_GET\_SWITCHES 625, 774  
 SNAP\_GET\_TAB(사용되지 않음) 1100  
 SNAP\_GET\_TAB\_REORG 632, 782  
 SNAP\_GET\_TAB\_V91 629, 778  
 SNAP\_GET\_TBSP(사용되지 않음) 1101  
 SNAP\_GET\_TBSP\_PART(사용되지 않음) 1104  
 SNAP\_GET\_TBSP\_PART\_V91 794  
 SNAP\_GET\_TBSP\_PART\_V97 644  
 SNAP\_GET\_TBSP QUIESCER 649, 800  
 SNAP\_GET\_TBSP\_RANGE 654, 805  
 SNAP\_GET\_TBSP\_V91 638, 788  
 SNAP\_GET\_UTIL 658, 809  
 SNAP\_GET\_UTIL\_PROGRESS 662, 813  
 SQLCACHE\_SNAPSHOT(사용되지 않음) 1146  
 WLM\_GET\_ACTIVITY\_DETAILS 1147  
 WLM\_GET\_SERVICE\_CLASS\_WORKLOAD  
 \_OCCURRENCES 1161  
 ADMIN\_IS\_INLINED 232, 246  
 AUDIT\_LIST\_LOGS 295

table

WLM\_GET\_QUEUE\_STATS 881  
 WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 885  
 WLM\_GET\_SERVICE\_CLASS\_WORKLOAD  
 \_OCCURRENCES\_V97 893  
 WLM\_GET\_SERVICE\_SUBCLASS\_STATS 1165  
 WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 897  
 WLM\_GET\_SERVICE\_SUPERCLASS\_STATS 906  
 WLM\_GET\_WORKLOAD\_OCCURRENCE  
 \_ACTIVITIES 1172  
 WLM\_GET\_WORKLOAD\_OCCURRENCE  
 \_ACTIVITIES\_V97 909

함수 (계속)

table (계속)

WLM\_GET\_WORKLOAD\_STATS 1177  
 WLM\_GET\_WORK\_ACTION\_SET\_STATS 907

**A**

ADD CONTACT 명령

ADMIN\_CMD를 사용한 구문 37

ADD CONTACTGROUP 명령

ADMIN\_CMD를 사용한 구문 39

ADMINTABCOMPRESSINFO 1004

ADMINTABCOMPRESSINFO 관리 뷰 251

ADMINTABINFO 관리 뷰 260

ADMINTEMPCOLUMNS 관리 뷰 268

ADMINTEMPFILES 관리 뷰 272

ADMIN\_CMD 프로시저 35

메시지 검색 244

메시지 제거 247

지원되는 명령

ADD CONTACT 37

ADD CONTACTGROUP 39

AUTOCONFIGURE 40

BACKUP DATABASE 44

DESCRIBE 53

DROP CONTACT 68

DROP CONTACTGROUP 69

EXPORT 70

FORCE APPLICATION 82

GET STMM TUNING DBPARTITIONNUM 84

IMPORT 85

INITIALIZE TAPE 113

LOAD 114

PRUNE HISTORY/LOGFILE 156

QUIESCE DATABASE 158

QUIESCE TABLESPACES FOR TABLE 160

REDISTRIBUTE DATABASE PARTITION GROUP 163

REORG INDEXES/TABLE 175

RESET ALERT CONFIGURATION 190

RESET DATABASE CONFIGURATION 192

RESET DATABASE MANAGER

CONFIGURATION 194

REWIND TAPE 195

RUNSTATS 196

SET TAPE POSITION 210

UNQUIESCE DATABASE 211

UPDATE ALERT CONFIGURATION 212

UPDATE CONTACT 219



ADMIN\_CMD 프로시저 (계속)  
 지원되는 명령 (계속)  
 UPDATE CONTACTGROUP 220  
 UPDATE DATABASE CONFIGURATION 221  
 UPDATE DATABASE MANAGER  
 CONFIGURATION 225  
 UPDATE HEALTH NOTIFICATION CONTACT  
 LIST 227  
 UPDATE HISTORY 228  
 UPDATE STMM TUNING DBPARTITIONNUM 231  
 ADMIN\_COPY\_SCHEMA 프로시저 923  
 ADMIN\_DROP\_SCHEMA 프로시저 927  
 admin\_get\_dbp\_mem\_usage  
 테이블 함수 234  
 ADMIN\_GET\_INDEX\_COMPRESS\_INFO 테이블 함수 236  
 ADMIN\_GET\_INDEX\_INFO 테이블 함수 240  
 ADMIN\_GET\_MSGS 테이블 함수 244  
 ADMIN\_GET\_TAB\_COMPRESS\_INFO 테이블 함수 1004  
 ADMIN\_GET\_TAB\_COMPRESS\_INFO\_V97 테이블 함수 251  
 ADMIN\_GET\_TAB\_INFO 테이블 함수 997  
 ADMIN\_GET\_TAB\_INFO\_V97 테이블 함수 260  
 ADMIN\_GET\_TEMP\_COLUMNS 테이블 함수 268  
 ADMIN\_GET\_TEMP\_TABLES 테이블 함수 272  
 ADMIN\_IS\_INLINED 함수 232, 246  
 ADMIN\_MOVE\_TABLE 프로시저 930  
 ADMIN\_MOVE\_TABLE\_UTIL 프로시저 942  
 ADMIN\_REMOVE\_MSGS 프로시저 247  
 ADMIN\_REVALIDATE\_DB\_OBJECTS 프로시저 248  
 ADMIN\_TASK\_ADD 프로시저 277  
 ADMIN\_TASK\_LIST 관리 뷰 283  
 ADMIN\_TASK\_REMOVE 프로시저 285  
 ADMIN\_TASK\_STATUS 관리 뷰 286  
 ADMIN\_TASK\_UPDATE 프로시저 288  
 ALTER\_ROUTINE\_PACKAGE 프로시저 825  
 ALTOBJ 프로시저 944  
 AM\_BASE\_RPTS 테이블 함수 18  
 AM\_BASE\_RPT\_RECOMS 테이블 함수 17  
 AM\_DROP\_TASK 프로시저 19  
 AM\_GET\_LOCK\_CHNS 프로시저 21  
 AM\_GET\_LOCK\_CHN\_TB 프로시저 20  
 AM\_GET\_LOCK\_RPT 프로시저 22  
 AM\_GET\_RPT 프로시저 30  
 AM\_SAVE\_TASK 프로시저 32  
 ANALYZE\_LOG\_SPACE 프로시저 833  
 APPLICATIONS 관리 뷰 492  
 APPLICATION\_ID 스칼라 함수 947  
 APPL\_PERFORMANCE 관리 뷰 491  
 AUDIT\_ARCHIVE 293

AUDIT\_DELIM\_EXTRACT 294  
 AUDIT\_LIST\_LOGS 295  
 AUTHORIZATIONIDS 관리 뷰 487  
 AUTH\_LIST\_AUTHORITIES\_FOR\_AUTHID 테이블 함수 479  
 AUTH\_LIST\_GROUPS\_FOR\_AUTHID 테이블 함수 483  
 AUTH\_LIST\_ROLES\_FOR\_AUTHID 484  
 AUTOCONFIGURE 명령  
 ADMIN\_CMD 사용 40  
 AUTOMAINT\_GET\_POLICY 스토어드 프로시저 297  
 AUTOMAINT\_GET\_POLICYFILE 스토어드 프로시저 298  
 AUTOMAINT\_SET\_POLICY 스토어드 프로시저 299  
 AUTOMAINT\_SET\_POLICYFILE 스토어드 프로시저 301

## B

BACKUP DATABASE 명령  
 ADMIN\_CMD 사용 44  
 BP\_HITRATIO 관리 뷰 497  
 BP\_READ\_IO 관리 뷰 499  
 BP\_WRITE\_IO 관리 뷰 501

## C

CANCEL\_WORK 스토어드 프로시저 309  
 CAPTURE\_STORAGEMGMT\_INFO 프로시저 845  
 COMPILATION\_ENV 테이블 함수 948  
 CONTACTGROUPS 관리 뷰 950  
 CONTACTS 관리 뷰 951  
 CONTAINER\_UTILIZATION 관리 뷰 503  
 CREATE\_STORAGEMGMT\_TABLES 프로시저 846

## D

DB2 서적 주문 1184  
 DB2 정보 센터  
 갱신 1187, 1188  
 다른 언어로 보기 1186  
 버전 1186  
 언어 1186  
 DBCFG 관리 뷰 350  
 DBMCFG 관리 뷰 352  
 DBPATHS 관리 뷰 957  
 DB\_HISTORY 관리 뷰  
 description 952  
 DB\_PARTITIONS 테이블 함수 349  
 DESCRIBE 명령  
 설명  
 ADMIN\_CMD 프로시저 포함 53

DROP CONTACT 명령  
ADMIN\_CMD 사용 68  
DROP CONTACTGROUP 명령  
ADMIN\_CMD 사용 69  
DROP\_STORAGEMGMT\_TABLES 프로시저 847

## E

ENV\_FEATURE\_INFO 관리 뷰  
설명 357  
ENV\_INST\_INFO 관리 뷰 358  
ENV\_PROD\_INFO 관리 뷰 360  
ENV\_SYS\_INFO 관리 뷰 361  
ENV\_SYS\_RESOURCES 관리 뷰 362  
EVMON\_FORMAT\_UE\_TO\_TABLES 프로시저  
description 368  
EVMON\_FORMAT\_UE\_TO\_XML 테이블 함수  
description 376  
EXPLAIN\_FORMAT\_STATS 스칼라 함수 959  
EXPLAIN\_GET\_MSGS 테이블 함수 965  
EXPORT 명령  
설명  
ADMIN\_CMD 프로시저 포함 70  
Extent  
이동, 상태 405

## F

FORCE APPLICATION 명령  
ADMIN\_CMD 사용 82

## G

GENERATE\_DISTFILE 프로시저 835  
GET\_STMM\_TUNING\_DBPARTITIONNUM 명령  
ADMIN\_CMD 사용 84  
GET\_CONFIG 스토어드 프로시저 316  
GET\_DBM\_CONFIG 테이블 함수 1010  
GET\_DBSIZE\_INFO 프로시저 967  
GET\_DB\_CONFIG 테이블 함수 1009  
GET\_MESSAGE 스토어드 프로시저 323  
GET\_ROUTINE\_OPTS 스칼라 함수 826  
GET\_ROUTINE\_SAR 프로시저 826  
GET\_SWRD\_SETTINGS 프로시저 836  
GET\_SYSTEM\_INFO 스토어드 프로시저 331

## H

Health 경보  
경보 구성 1040  
경보 조치 구성 1037  
Health 표시기  
Health 표시기 정의 검색 1043  
HEALTH\_CONT\_HI 테이블 함수 1011  
HEALTH\_CONT\_HI\_HIS 테이블 함수 1013  
HEALTH\_CONT\_INFO 테이블 함수 1015  
HEALTH\_DBM\_HI 테이블 함수 1031  
HEALTH\_DBM\_HI\_HIS 테이블 함수 1033  
HEALTH\_DBM\_INFO 테이블 함수 1035  
HEALTH\_DB\_HI 테이블 함수 1017  
HEALTH\_DB\_HIC 테이블 함수 1025  
HEALTH\_DB\_HIC\_HIS 테이블 함수 1027  
HEALTH\_DB\_HI\_HIS 테이블 함수 1021  
HEALTH\_DB\_INFO 테이블 함수 1030  
HEALTH\_GET\_ALERT\_ACTION\_CFG 테이블 함수 1037  
HEALTH\_GET\_ALERT\_CFG 테이블 함수 1040  
HEALTH\_GET\_IND\_DEFINITION 테이블 함수 1043  
HEALTH\_HI\_REC 프로시저 1046  
HEALTH\_TBS\_HI 테이블 함수 1047  
HEALTH\_TBS\_HI\_HIS 테이블 함수 1050  
HEALTH\_TBS\_INFO 테이블 함수 1054

## I

IMPORT 명령  
ADMIN\_CMD 사용 85  
INITIALIZE TAPE 명령  
ADMIN\_CMD 사용 113

## L

LOAD 명령  
ADMIN\_CMD 사용 114  
LOCKS\_HELD 관리 뷰 505  
LOCKWAIT 관리 뷰 508  
LOG\_UTILIZATION 관리 뷰 513  
LONG\_RUNNING\_SQL 관리 뷰 514

## M

MON\_GET\_ACTIVITY\_DETAILS 테이블 함수  
description 379  
MON\_GET\_BUFFERPOOL 테이블 함수  
설명 388

MON\_GET\_CONNECTION 테이블 함수  
 설명 391

MON\_GET\_CONNECTION\_DETAILS 테이블 함수  
 설명 396

MON\_GET\_CONTAINER 테이블 함수  
 설명 402

MON\_GET\_EXTENT\_MOVEMENT\_STATUS  
 description 405

MON\_GET\_INDEX 테이블 함수  
 description 406

MON\_GET\_PKG\_CACHE\_STMT 테이블 함수  
 description 409

MON\_GET\_SERVICE\_SUBCLASS 테이블 함수  
 description 414

MON\_GET\_SERVICE\_SUBCLASS\_DETAILS 테이블 함수  
 설명 420

MON\_GET\_TABLE 테이블 함수  
 설명 426

MON\_GET\_TABLESPACE 테이블 함수  
 설명 429

MON\_GET\_UNIT\_OF\_WORK 테이블 함수  
 description 433

MON\_GET\_UNIT\_OF\_WORK\_DETAILS 테이블 함수  
 description 438

MON\_GET\_WORKLOAD 테이블 함수  
 description 445

MON\_GET\_WORKLOAD\_DETAILS 테이블 함수  
 설명 450

MQPUBLISH 스칼라 함수 457

MQREAD 스칼라 함수 459

MQREADALL 테이블 함수 460

MQREADALLCLOB 테이블 함수 462

MQREADCLOB 스칼라 함수 464

MQRECEIVE 스칼라 함수 465

MQRECEIVEALL 테이블 함수 467

MQRECEIVEALLCLOB 테이블 함수 469

MQRECEIVECLOB 스칼라 함수 471

MQSEND 스칼라 함수 473

MQSUBSCRIBE 스칼라 함수 474

MQUNSUBSCRIBE 스칼라 함수 476

## N

NOTIFICATIONLIST 관리 뷰 970

## O

OBJECTOWNERS 관리 뷰 488

## P

PDLOGMSG\_LAST24HOURS 관리 뷰  
 설명 978

PD\_GET\_DIAG\_HIST 테이블 함수  
 설명 971

PD\_GET\_LOG\_MSGS 테이블 함수  
 설명 978

PRIVILEGES 관리 뷰 489

PRUNE HISTORY/LOGFILE 명령  
 ADMIN\_CMD 프로시저 포함 156

PUT\_ROUTINE\_SAR 프로시저 828

## Q

QUERY\_PREP\_COST 관리 뷰 517

QUIESCE DATABASE 명령  
 ADMIN\_CMD 사용 158

QUIESCE TABLESPACES FOR TABLE 명령  
 ADMIN\_CMD 사용 160

## R

REBIND\_ROUTINE\_PACKAGE 프로시저 829

REDISTRIBUTE DATABASE PARTITION GROUP 명령  
 ADMIN\_CMD 사용 163

REG\_VARIABLES 관리 뷰 355

REORG INDEXES/TABLE 명령  
 ADMIN\_CMD 사용 175

REORGCHK\_IX\_STATS 프로시저 985

REORGCHK\_TB\_STATS 프로시저 987

RESET ALERT CONFIGURATION 명령  
 ADMIN\_CMD 사용 190

RESET DATABASE CONFIGURATION 명령  
 ADMIN\_CMD 사용 192

RESET DATABASE MANAGER CONFIGURATION 명령  
 ADMIN\_CMD 사용 194

REWIND TAPE 명령  
 ADMIN\_CMD 사용 195

RUNSTATS 명령  
 ADMIN\_CMD 사용 196

## S

SET TAPE POSITION 명령  
 ADMIN\_CMD 사용 210

SET\_CONFIG 스토어드 프로시저 338

SET\_ROUTINE\_OPTS 프로시저 831

SET\_SWRD\_SETTINGS 프로시저 839  
 SNAPAGENT 관리 뷰 518, 667  
 SNAPAGENT\_MEMORY\_POOL 관리 뷰 522, 671  
 SNAPAPPL 관리 뷰 535, 684  
 SNAPAPPL\_INFO 관리 뷰 526, 675  
 SNAPBP 관리 뷰 543, 692  
 SNAPBP\_PART 관리 뷰 548, 697  
 SNAPCONTAINER 관리 뷰 552, 701  
 SNAPDB 관리 뷰 557, 706  
 SNAPDBM 관리 뷰 572, 721  
 SNAPDBM\_MEMORY\_POOL 관리 뷰 577, 726  
 SNAPDB\_MEMORY\_POOL 관리 뷰 568, 717  
 SNAPDETAILLOG 관리 뷰 581, 730  
 SNAPDYN\_SQL 관리 뷰 584, 733  
 SNAPFCM 관리 뷰 590, 739  
 SNAPFCM\_PART 관리 뷰 592, 741  
 SNAPHADR 관리 뷰 595, 744  
 SNAPLOCK 관리 뷰 600, 749  
 SNAPLOCKWAIT 관리 뷰 606, 755  
 SNAPSHOT\_AGENT 사용되지 않는 테이블 함수 1106  
 SNAPSHOT\_APPL 사용되지 않는 테이블 함수 1107  
 SNAPSHOT\_APPL\_INFO 사용되지 않는 테이블 함수 1112  
 SNAPSHOT\_BP 사용되지 않는 테이블 함수 1114  
 SNAPSHOT\_CONTAINER 사용되지 않는 테이블 함수 1116  
 SNAPSHOT\_DATABASE 사용되지 않는 테이블 함수 1117  
 SNAPSHOT\_DBM 사용되지 않는 테이블 함수 1123  
 SNAPSHOT\_DYN\_SQL 사용되지 않는 테이블 함수 1124  
 SNAPSHOT\_FCM 사용되지 않는 테이블 함수 1126  
 SNAPSHOT\_FCMNODE 사용되지 않는 테이블 함수 1127  
 SNAPSHOT\_FILEW 사용되지 않는 프로시저 1128  
 SNAPSHOT\_LOCK 사용되지 않는 테이블 함수 1129  
 SNAPSHOT\_LOCKWAIT 사용되지 않는 테이블 함수 1130  
 SNAPSHOT\_QUIESCERS 사용되지 않는 테이블 함수 1132  
 SNAPSHOT\_RANGES 사용되지 않는 테이블 함수 1133  
 SNAPSHOT\_STATEMENT 사용되지 않는 테이블 함수 1134  
 SNAPSHOT\_SUBSECT 사용되지 않는 테이블 함수 1136  
 SNAPSHOT\_SWITCHES 사용되지 않는 테이블 함수 1137  
 SNAPSHOT\_TABLE 사용되지 않는 테이블 함수 1139  
 SNAPSHOT\_TBREORG 사용되지 않는 테이블 함수 1140  
 SNAPSHOT\_TBS 사용되지 않는 테이블 함수 1141  
 SNAPSHOT\_TBS\_CFG 사용되지 않는 테이블 함수 1143  
 SNAPSTMT 관리 뷰 612, 761  
 SNAPSTORAGE\_PATHS 관리 뷰 618, 767  
 SNAPSUBSECTION 관리 뷰 621, 770  
 SNAPSWITCHES 관리 뷰 625, 774  
 SNAPTAB 관리 뷰 629, 778  
 SNAPTAB\_REORG 관리 뷰 632, 782  
 SNAPTbsp 관리 뷰 638, 788  
 SNAPTBSPPART 관리 뷰 644, 794  
 SNAPTbsp\_QUIESCER 관리 뷰 649, 800  
 SNAPTbsp\_RANGE 관리 뷰 654, 805  
 SNAPUTIL 관리 뷰 658, 809  
 SNAPUTIL\_PROGRESS 관리 뷰 662, 813  
 SNAP\_GET\_AGENT 테이블 함수 518, 667  
 SNAP\_GET\_AGENT\_MEMORY\_POOL 테이블 함수 522, 671  
 SNAP\_GET\_APPL\_INFO 테이블 함수 1062  
 SNAP\_GET\_APPL\_INFO\_V95 테이블 함수 526, 675  
 SNAP\_GET\_APPL\_V95 테이블 함수 535, 684  
 SNAP\_GET\_BP\_PART 테이블 함수 548, 697  
 SNAP\_GET\_BP\_V95 테이블 함수 543, 692  
 SNAP\_GET\_CONTAINER 사용되지 않는 테이블 함수 1073  
 SNAP\_GET\_CONTAINER\_V91 테이블 함수 552, 701  
 SNAP\_GET\_DB 사용되지 않는 테이블 함수 1074  
 SNAP\_GET\_DBM\_MEMORY\_POOL 테이블 함수 577, 726  
 SNAP\_GET\_DBM\_V95 테이블 함수 572, 721  
 SNAP\_GET\_DB\_MEMORY\_POOL 테이블 함수 568, 717  
 SNAP\_GET\_DB\_V95 테이블 함수 557, 706  
 SNAP\_GET\_DETAIL\_LOG\_V91 테이블 함수 581, 730  
 SNAP\_GET\_DYN\_SQL 사용되지 않는 테이블 함수 1097  
 SNAP\_GET\_DYN\_SQL\_V91 테이블 함수 1093  
 SNAP\_GET\_DYN\_SQL\_V95 테이블 함수 584, 733  
 SNAP\_GET\_FCM 테이블 함수 590, 739  
 SNAP\_GET\_FCM\_PART 테이블 함수 592, 741  
 SNAP\_GET\_HADR 테이블 함수 595, 744  
 SNAP\_GET\_LOCK 테이블 함수 600, 749  
 SNAP\_GET\_LOCKWAIT 테이블 함수 606, 755  
 SNAP\_GET\_STMT 테이블 함수 612, 761  
 SNAP\_GET\_STORAGE\_PATHS 테이블 함수 767  
 SNAP\_GET\_STORAGE\_PATHS\_V97 테이블 함수 618  
 SNAP\_GET\_STO\_PATHS 사용되지 않는 테이블 함수 1099  
 SNAP\_GET\_SUBSECTION 테이블 함수 621, 770  
 SNAP\_GET\_SWITCHES 테이블 함수 625, 774  
 SNAP\_GET\_TAB 사용되지 않는 테이블 함수 1100  
 SNAP\_GET\_TAB\_REORG 테이블 함수 632, 782  
 SNAP\_GET\_TAB\_V91 테이블 함수 629, 778  
 SNAP\_GET\_TBSP 사용되지 않는 테이블 함수 1101  
 SNAP\_GET\_TBSP\_PART 사용되지 않는 테이블 함수 1104  
 SNAP\_GET\_TBSP\_PART\_V91 테이블 함수 794  
 SNAP\_GET\_TBSP\_PART\_V97 테이블 함수 644  
 SNAP\_GET\_TBSP\_QUIESCER 테이블 함수 649, 800  
 SNAP\_GET\_TBSP\_RANGE 테이블 함수 654, 805  
 SNAP\_GET\_TBSP\_V91 테이블 함수 638, 788  
 SNAP\_GET\_UTIL 테이블 함수 658, 809  
 SNAP\_GET\_UTIL\_PROGRESS 테이블 함수 662, 813  
 SNAP\_WRITE\_FILE 프로시저 666, 816  
 SQL 관리 루틴 859

SQL 관리 루틴 (계속)

사용되지 않는 루틴 995

SQLCACHE\_SNAPSHOT 사용되지 않는 테이블 함수 1146

SQLCODE

메시지 정보 리턴 323

SQLERRM 스칼라 함수 989

SQL문

도움말 표시 1185

STEPWISE\_REDISTRIBUTE\_DBPG 프로시저 841

SYSINSTALLOBJECTS 프로시저 991

SYSINSTALLROUTINES 사용되지 않는 프로시저 1147

SYSTS\_ADMIN\_CMD 프로시저 849

SYSTS\_ALTER 스토어드 프로시저 850

SYSTS\_CLEAR\_COMMANDLOCKS 스토어드 프로시저 854

SYSTS\_CLEAR\_EVENTS 스토어드 프로시저 857

SYSTS\_CREATE 859

SYSTS\_CREATE 프로시저 859

SYSTS\_DISABLE 866

SYSTS\_DISABLE 프로시저 866

SYSTS\_DROP 스토어드 프로시저 869

SYSTS\_ENABLE 스토어드 프로시저 872

SYSTS\_UPDATE 스토어드 프로시저 873

## T

TBSP\_UTILIZATION 관리 뷰 818

TOP\_DYNAMIC\_SQL 관리 뷰 822

## U

UNQUIESCE DATABASE 명령

ADMIN\_CMD 사용 211

UPDATE ALERT CONFIGURATION 명령

ADMIN\_CMD 사용 212

UPDATE CONTACT 명령

ADMIN\_CMD 사용 219

UPDATE CONTACTGROUP 명령

ADMIN\_CMD 사용 220

UPDATE DATABASE CONFIGURATION 명령

ADMIN\_CMD 사용 221

UPDATE DATABASE MANAGER CONFIGURATION 명령

ADMIN\_CMD 사용 225

UPDATE HEALTH NOTIFICATION CONTACT LIST 명령

ADMIN\_CMD 사용 227

UPDATE HISTORY 명령

ADMIN\_CMD 사용 228

UPDATE STMM TUNING DBPARTITIONNUM 명령

ADMIN\_CMD 사용 231

## V

Visual Explain

자습서 1190

## W

WLM\_CANCEL\_ACTIVITY 프로시저 877

WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS 프로시저 878

WLM\_COLLECT\_STATS 프로시저

설명 880

WLM\_GET\_ACTIVITY\_DETAILS 테이블 함수

설명 1147

WLM\_GET\_QUEUE\_STATS 테이블 함수 881

WLM\_GET\_SERVICE\_CLASS\_AGENTS 테이블 함수 1154

WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 테이블 함수

description 885

WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES

테이블 함수

description 1161

WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES\_V97

테이블 함수

description 893

WLM\_GET\_SERVICE\_SUBCLASS\_STATS 테이블 함수

설명 1165

WLM\_GET\_SERVICE\_SUBCLASS\_STATS\_V97 테이블 함수

설명 897

WLM\_GET\_SERVICE\_SUPERCLASS\_STATS 테이블 함수 906

WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 테이블

함수 1172

WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES\_V97 테

이블 함수

description 909

WLM\_GET\_WORKLOAD\_STATS 테이블 함수

설명 1177

WLM\_GET\_WORKLOAD\_STATS\_V97 테이블 함수

설명 915

WLM\_GET\_WORK\_ACTION\_SET\_STATS 테이블 함수

description 907

WLM\_SET\_CLIENT\_INFO 프로시저 919

## X

XML

일반 SQL API의 유효한 입력 306

XML 출력 문서

일반 SQL API의 버전화 304







SA30-3955-00





Spine information:

Linux, UNIX 및 Windows용 IBM DB2 9.7

관리 루틴 및 뷰

