

IBM DB2 Universal Database



# Dienstprogramme für das Versetzen von Daten Handbuch und Referenz

*Version 8.2*



IBM DB2 Universal Database



# Dienstprogramme für das Versetzen von Daten Handbuch und Referenz

*Version 8.2*

#### Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen unter *Bemerkungen* gelesen werden.

- Die IBM Homepage finden Sie im Internet unter: **ibm.com**
- IBM und das IBM Logo sind eingetragene Marken der International Business Machines Corporation.
- Das e-business Symbol ist eine Marke der International Business Machines Corporation
- Infoprint ist eine eingetragene Marke der IBM.
- ActionMedia, LANDesk, MMX, Pentium und ProShare sind Marken der Intel Corporation in den USA und/oder anderen Ländern.
- C-bus ist eine Marke der Corollary, Inc. in den USA und/oder anderen Ländern.
- Java und alle Java-basierenden Marken und Logos sind Marken der Sun Microsystems, Inc. in den USA und/oder anderen Ländern.
- Microsoft Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.
- PC Direct ist eine Marke der Ziff Communications Company in den USA und/oder anderen Ländern.
- SET und das SET-Logo sind Marken der SET Secure Electronic Transaction LLC.
- UNIX ist eine eingetragene Marke der Open Group in den USA und/oder anderen Ländern.
- Marken anderer Unternehmen/Hersteller werden anerkannt.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs  
*IBM DB2 Universal Database Data Movement Utilities Guide and Reference Version 8.2*,  
IBM Form SC09-4830-01,  
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 1999, 2004  
© Copyright IBM Deutschland GmbH 2004

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:  
SW TSC Germany  
Kst. 2877  
April 2004

# Inhaltsverzeichnis

<b>Zu diesem Handbuch</b> . . . . .	<b>vii</b>	Verwenden von LOAD mit Identitätsspalten . . . . .	98
Zielgruppe . . . . .	vii	Verwenden von LOAD mit generierten Spalten . . . . .	100
Aufbau dieses Handbuchs . . . . .	viii	Überprüfen auf ungültige Integritätsbedingungen . . . . .	102
<b>Kapitel 1. EXPORT</b> . . . . .	<b>1</b>	Aktualisieren von abhängigen sofort gespeicherten Abfragetabellen . . . . .	105
EXPORT - Übersicht . . . . .	1	Weitergeben von abhängigen sofort gespeicherten Zwischenspeichertabellen . . . . .	106
Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von EXPORT . . . . .	3	Hinweise zum mehrdimensionalen Clustering . . . . .	108
Verwenden von EXPORT . . . . .	3	Erneutes Starten einer unterbrochenen Ladeoperation . . . . .	109
Verwenden von EXPORT mit Identitätsspalten . . . . .	4	Erneutes Starten oder Beenden einer Ladeoperation mit ALLOW READ ACCESS . . . . .	109
Erneutes Erstellen einer exportierten Tabelle . . . . .	4	Wiederherstellen von Daten mit der Datei mit den Angaben zur Speicherposition der Ladekopie . . . . .	110
Exportieren von großen Objekten (LOBS) . . . . .	5	Befehl LOAD . . . . .	112
Paralleles Exportieren von Daten . . . . .	6	LOAD QUERY . . . . .	137
Befehl EXPORT . . . . .	9	Lade-API "db2Load" . . . . .	139
Export-API "db2Export" . . . . .	14	Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs . . . . .	163
Änderungswerte für Dateityp für EXPORT . . . . .	22	Änderungswerte für Dateityp für LOAD . . . . .	168
CLP-Beispiele für EXPORT-Sitzungen . . . . .	26	LOAD-Ausnahmetabelle . . . . .	183
<b>Kapitel 2. IMPORT</b> . . . . .	<b>27</b>	LOAD-Speicherauszugsdatei . . . . .	184
IMPORT - Übersicht . . . . .	27	Temporäre Dateien für das Dienstprogramm LOAD . . . . .	184
Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von IMPORT . . . . .	29	Protokollsätze für das Dienstprogramm LOAD . . . . .	185
Verwenden von IMPORT . . . . .	29	Tabellensperren, Statuswerte für Tabellen und Statuswerte für Tabellenbereiche . . . . .	186
Verwenden von IMPORT in einer Client/Server-Umgebung . . . . .	30	Zeichensatz und NLS . . . . .	189
Verwenden von IMPORT mit gepufferten INSERT-Operationen . . . . .	31	Statusangaben für anstehende Aktionen nach einer Ladeoperation . . . . .	190
Verwenden von IMPORT mit Identitätsspalten . . . . .	32	Optimieren der Leistung von LOAD . . . . .	191
Verwenden von IMPORT mit generierten Spalten . . . . .	34	CLP-Beispiele für LOAD . . . . .	197
Erneutes Erstellen einer exportierten Tabelle mit IMPORT . . . . .	35	<b>Kapitel 4. Laden von Daten in einer Umgebung mit partitionierten Datenbanken</b> . . . . .	<b>201</b>
Importieren von großen Objekten (LOBS) . . . . .	36	Übersicht über das Laden in partitionierten Datenbanken . . . . .	201
Importieren von benutzerdefinierten einzigartigen Datentypen (UDTs) . . . . .	37	Verwenden von LOAD in einer Umgebung mit partitionierten Datenbanken . . . . .	203
Sperren von Tabellen während des Imports . . . . .	37	Überwachen des Ladens in partitionierte Datenbanken mit LOAD QUERY . . . . .	209
Befehl IMPORT . . . . .	39	Erneutes Starten oder Beenden einer Ladeoperation in einer Umgebung mit partitionierten Datenbanken . . . . .	211
Import-API "db2Import" . . . . .	53	LOAD-Konfigurationsoptionen für partitionierte Datenbanken . . . . .	213
Änderungswerte für Dateityp für IMPORT . . . . .	66	Beispiele für LOAD-Sitzungen für partitionierte Datenbanken . . . . .	219
Hinweise zu Zeichensatz und NLS . . . . .	76	Migration und Kompatibilität zu früheren Versionen . . . . .	222
CLP-Beispiele für IMPORT-Sitzungen . . . . .	76	Hinweise und Tipps für das Laden von Daten in einer Umgebung mit partitionierten Datenbanken . . . . .	224
<b>Kapitel 3. LOAD</b> . . . . .	<b>81</b>	<b>Kapitel 5. Versetzen von DB2 Data Links Manager-Daten</b> . . . . .	<b>227</b>
LOAD - Übersicht . . . . .	82		
Änderungen der früheren Funktionsweise des Dienstprogramms LOAD (eingeführt in Version 6 und 7) . . . . .	86		
Änderungen der früheren Funktionsweise des Dienstprogramms LOAD (eingeführt in Version 8) . . . . .	87		
Parallelität und Laden . . . . .	89		
Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von LOAD . . . . .	90		
Verwenden von LOAD . . . . .	91		
Ladeoperationen mit Lesezugriff . . . . .	93		
Erstellen von Indizes . . . . .	96		

Versetzen von DB2 Data Links Manager-Daten mit EXPORT - Konzepte . . . . .	227
Versetzen von DB2 Data Links Manager-Daten mit EXPORT . . . . .	230
Versetzen von DB2 Data Links Manager-Daten mit IMPORT . . . . .	231
Versetzen von DB2 Data Links Manager-Daten mit LOAD . . . . .	232

**Kapitel 6. Versetzen von Daten zwischen Systemen . . . . . 233**

Versetzen von Daten zwischen Plattformen - Hin- weise zum Dateiformat . . . . .	233
PC/IXF-Dateiformat . . . . .	233
DEL-Dateiformat (ASCII-Format ohne univer- selle Zeilenbegrenzer) . . . . .	234
WSF-Dateiformat . . . . .	234
Versetzen von Daten mit DB2 Connect . . . . .	235
Befehl "db2move" - Tool zum Versetzen von Daten	237
Befehl "db2relocatedb" zum Verlagern einer Daten- bank . . . . .	242
Einschränkungen für Begrenzer beim Versetzen von Daten . . . . .	247
Versetzen von Daten zwischen typisierten Tabellen	248
Versetzen von Daten zwischen typisierten Tabellen - Details . . . . .	249
Traversierfolge . . . . .	249
Datenauswahl beim Versetzen von Daten . . . . .	251
Beispiele zum Versetzen von Daten zwischen typisierten Tabellen . . . . .	251
Verwenden der Replikation zum Versetzen von Daten . . . . .	253
IBM Replikationstools . . . . .	254
IBM Replikationstools nach Komponenten. . . . .	254
Versetzen von Daten mit der Data Warehouse-Zen- trale . . . . .	255
Versetzen von Daten mit Dateityp CURSOR . . . . .	257

**Anhang A. Lesen von Syntax- diagrammen . . . . . 259**

**Anhang B. Unterschiede zwischen den Dienstprogrammen IMPORT und LOAD . . . . . 263**

**Anhang C. API-Beispielprogramm für EXPORT-, IMPORT- und LOAD-Sitzun- gen. . . . . 265**

**Anhang D. Dateiformate . . . . . 275**

Dateiformate der Dienstprogramme EXPORT, IMPORT und LOAD . . . . .	275
DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer) . . . . .	276
Beispiel und Datentypbeschreibungen . . . . .	278
Beispiel für DEL-Datei . . . . .	278
Beschreibungen von DEL-Datentypen . . . . .	279
ASC-Dateiformat (ASCII-Format mit universellen Zeilenbegrenzern) . . . . .	281

Beispiel und Datentypbeschreibungen . . . . .	282
Beispiel für ASC-Datei . . . . .	282
Beschreibungen von ASC-Datentypen . . . . .	283
PC-Version des IXF-Dateiformats . . . . .	285
PC-Version des IXF-Dateiformats - Details . . . . .	287
PC/IXF-Satztypen . . . . .	287
PC/IXF-Datentypen . . . . .	304
Beschreibungen von PC/IXF-Datentypen . . . . .	311
Allgemeine Regeln für den Import von PC/IXF- Dateien in Datenbanken . . . . .	315
Datentypspezifische Regeln für den Import von PC/IXF-Dateien in Datenbanken . . . . .	317
Option FORCEIN . . . . .	320
Unterschiede zwischen PC/IXF und Version 0 System/370 IXF . . . . .	328
WSF-Dateiformat (Arbeitsblattformat) . . . . .	329

**Anhang E. Hinweise zu Unicode bei Verwendung der Dienstprogramme EXPORT/IMPORT/LOAD . . . . . 331**

Einschränkungen für Codepages 1394, 1392 und 5488 . . . . .	332
Inkompatibilität . . . . .	333

**Anhang F. Durch Dienstprogramme EXPORT, IMPORT und LOAD verwen- dete Bindedateien . . . . . 335**

**Anhang G. Warnungen, Fehler- und Beendigungsnachrichten . . . . . 337**

**Anhang H. Technische Informationen zu DB2 Universal Database . . . . . 339**

DB2-Dokumentation und Hilfe . . . . .	339
Aktualisierungen der DB2-Dokumentation . . . . .	339
DB2 Information - Unterstützung. . . . .	340
DB2 Information - Unterstützung: Installations- szenarios . . . . .	341
Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (UNIX) . . . . .	344
Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (Windows) . . . . .	347
Aufrufen von 'DB2 Information - Unterstützung' . . . . .	349
Aktualisieren der auf Ihrem Computer oder Intra- net-Server installierten Komponente 'DB2 Informa- tion - Unterstützung' . . . . .	350
Anzeigen von Themen in der gewünschten Sprache in 'DB2 Information - Unterstützung' . . . . .	351
DB2-Dokumentation in PDF-Format und gedruck- tem Format . . . . .	352
DB2-Kerninformationen . . . . .	352
Verwaltungsinformationen . . . . .	353
Informationen zur Anwendungsentwicklung . . . . .	354
Informationsmanagement . . . . .	354
Informationen zu DB2 Connect . . . . .	355
Einführungsinformationen . . . . .	355
Lernprogramminformationen . . . . .	355
Informationen zu Zusatzkomponenten . . . . .	356
Release-Informationen . . . . .	356

Drucken von DB2-Büchern mit PDF-Dateien . . .	357	Dokumentation im behindertengerechten For-	
Bestellen gedruckter DB2-Bücher . . . . .	358	mat. . . . .	364
Aufrufen der Kontexthilfe über ein DB2-Tool . . .	359	Syntaxdiagramme in der Schreibweise mit Trenn-	
Aufrufen der Hilfe für Nachrichten über den		zeichen . . . . .	365
Befehlszeilenprozessor . . . . .	360	Common Criteria-Zertifizierung von DB2 Universal	
Aufrufen der Hilfe für Befehle über den Befehls-		Database-Produkten . . . . .	367
zeilenprozessor . . . . .	360		
Aufrufen der Hilfe für den SQL-Status über den		<b>Anhang I. Bemerkungen. . . . .</b>	<b>369</b>
Befehlszeilenprozessor . . . . .	361	Marken . . . . .	371
DB2-Lernprogramme . . . . .	361		
Informationen zur Fehlerbehebung in DB2 . . .	362	<b>Index . . . . .</b>	<b>373</b>
Eingabehilfen . . . . .	363		
Tastatureingabe und Navigation . . . . .	363	<b>Kontaktaufnahme mit IBM . . . . .</b>	<b>377</b>
Eingabehilfen für Bildschirme . . . . .	364	Produktinformationen . . . . .	377
Kompatibilität mit Unterstützungseinrichtungen	364		



---

## Zu diesem Handbuch

Dieses Handbuch bietet Informationen zu den folgenden IBM DB2 Universal Database (UDB)-Dienstprogrammen zum Versetzen von Daten und ihrer Verwendung:

- Mit Hilfe der Dienstprogramme IMPORT und EXPORT werden Daten zwischen einer Tabelle oder Sicht und einem anderen Datenbank- oder Tabellenkalkulationsprogramm, zwischen DB2-Datenbanken sowie zwischen DB2-Datenbanken und Host-Datenbanken mit Hilfe von DB2 Connect versetzt. Mit dem Dienstprogramm EXPORT können Daten von einer Datenbank in Betriebssystemdateien versetzt werden. Diese Dateien können Sie dann verwenden, um diese Daten in eine andere Datenbank zu importieren oder zu laden.
- Mit dem Dienstprogramm LOAD können Daten in Tabellen versetzt, vorhandene Indizes erweitert und Statistikdaten generiert werden. Mit LOAD werden die Daten viel schneller als mit dem Dienstprogramm IMPORT versetzt, wenn es sich um umfangreiche Datenmengen handelt. Daten, die mit dem Dienstprogramm EXPORT aus Tabellen geladen wurden, können mit dem Dienstprogramm LOAD wieder geladen werden.
- Wird das Dienstprogramm LOAD in einer Umgebung mit partitionierten Datenbanken eingesetzt, können große Datenmengen partitioniert und in unterschiedliche Datenbankpartitionen geladen werden.
- DataPropagator (DPROP) ist eine Komponente von DB2 Universal Database, die das automatische Kopieren von Tabellenaktualisierungen in andere Tabellen anderer relationaler DB2-Datenbanken ermöglicht.
- Mit Hilfe der Data Warehouse-Zentrale können Daten aus Betriebsdatenbanken in eine Warehouse-Datenbank versetzt werden.

Fremdprodukte zum Versetzen von Daten in und aus Datenbanken stehen ebenfalls zur Verfügung, allerdings werden sie im vorliegenden Handbuch nicht näher behandelt.

---

## Zielgruppe

Dieses Handbuch richtet sich an Datenbankadministratoren, Anwendungsprogrammierer und andere DB2 UDB-Benutzer, welche die folgenden Aufgaben ausführen:

- Laden von Daten in DB2-Tabellen aus Betriebssystemdateien
- Versetzen von Daten zwischen DB2-Datenbanken und zwischen DB2 und anderen Anwendungen (z. B. Tabellenkalkulationsprogramme)
- Archivieren von Daten

Es wird vorausgesetzt, dass Sie mit DB2 Universal Database, Structured Query Language (SQL) und der Betriebssystemumgebung vertraut sind, in der DB2 UDB ausgeführt wird.

---

## Aufbau dieses Handbuchs

Die folgenden Themen werden behandelt:

- Kapitel 1** Beschreibt das DB2-Dienstprogramm EXPORT, mit dem Daten aus DB2-Tabellen in Dateien versetzt werden.
- Kapitel 2** Beschreibt das DB2-Dienstprogramm IMPORT, mit dem Daten aus Dateien in DB2-Tabellen oder -Sichten versetzt werden.
- Kapitel 3** Beschreibt das DB2-Dienstprogramm LOAD, mit dem große Datenmengen in DB2-Tabellen versetzt werden.
- Kapitel 4** Beschreibt das Laden von Daten in einer Umgebung mit partitionierten Datenbanken.
- Kapitel 5** Beschreibt, wie die DB2-Dienstprogramme EXPORT, IMPORT und LOAD verwendet werden, um DB2 Data Links Manager-Daten zu versetzen.
- Kapitel 6** Beschreibt, wie die DB2-Dienstprogramme EXPORT, IMPORT und LOAD verwendet werden, um Daten zwischen verschiedenen Plattformen sowie in und aus DRDA-Host-Datenbanken zu übertragen. Auch DataPropagator (DPROP), eine weitere Methode zum Versetzen von Daten innerhalb eines Unternehmens, wird beschrieben. Dieses Kapitel enthält außerdem eine Einführung in die Data Warehouse-Zentrale, mit deren Hilfe Daten aus Betriebsdatenbanken in eine Warehouse-Datenbank versetzt werden können.
- Anhang A** Erläutert die in Syntaxdiagrammen verwendeten Konventionen.
- Anhang B** Fasst wichtige Unterschiede zwischen den DB2-Dienstprogrammen LOAD und IMPORT zusammen.
- Anhang C** Enthält ein API-Beispielprogramm, das veranschaulicht, wie Daten in eine Datei exportiert, in eine Tabelle importiert sowie in eine Tabelle geladen werden und wie der Status einer Ladeoperation überprüft werden kann.
- Anhang D** Beschreibt externe Dateiformate, die von den Datenbankmanager-Dienstprogrammen EXPORT, IMPORT und LOAD unterstützt werden.
- Anhang E** Enthält Hinweise zu Unicode bei Verwendung der Dienstprogramme EXPORT, IMPORT und LOAD.
- Anhang F** Listet Bindedateien mit ihren jeweiligen Standardisolationsstufen auf. Außerdem ist angegeben, welche Dienstprogramme diese Dateien einsetzen und zu welchem Zweck sie verwendet werden.
- Anhang G** Bietet Informationen zur Auswertung von Nachrichten, die vom Datenbankmanager generiert werden, wenn eine Warn- oder Fehlerbedingung erkannt wird.

---

## Kapitel 1. EXPORT

In diesem Kapitel wird das DB2 UDB-Dienstprogramm EXPORT beschrieben, mit dem Daten aus einer DB2-Datenbank in eine oder mehrere Dateien geschrieben werden können, die außerhalb der Datenbank gespeichert werden. Die exportierten Daten können dann mit dem DB2-Dienstprogramm IMPORT bzw. LOAD in eine andere DB2-Datenbank importiert bzw. geladen werden, oder sie können in eine andere Anwendung (z. B. ein Tabellenkalkulationsprogramm) importiert werden.

Die folgenden Themen werden behandelt:

- „EXPORT - Übersicht“
- „Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von EXPORT“ auf Seite 3
- „Verwenden von EXPORT“ auf Seite 3
- „Verwenden von EXPORT mit Identitätsspalten“ auf Seite 4
- „Erneutes Erstellen einer exportierten Tabelle“ auf Seite 4
- „Exportieren von großen Objekten (LOBS)“ auf Seite 5
- „Paralleles Exportieren von Daten“ auf Seite 6
- „Befehl EXPORT“ auf Seite 9
- „Export-API "db2Export"" auf Seite 14
- „CLP-Beispiele für EXPORT-Sitzungen“ auf Seite 26

Informationen zum Exportieren von DB2 Data Links Manager-Daten finden Sie unter „Versetzen von DB2 Data Links Manager-Daten mit EXPORT“ auf Seite 230. Angaben zum Exportieren von Daten aus typisierten Tabellen enthält der Abschnitt „Versetzen von Daten zwischen typisierten Tabellen“ auf Seite 248. Informationen zum Exportieren von Daten aus einer DRDA-Server-Datenbank in eine Datei auf der DB2 Connect-Workstation und umgekehrt können Sie unter „Versetzen von Daten mit DB2 Connect“ auf Seite 235 nachlesen.

---

### EXPORT - Übersicht

Das Dienstprogramm EXPORT exportiert Daten aus einer Datenbank in eine Betriebssystemdatei, die eines von mehreren externen Dateiformaten haben kann. Mit Hilfe dieser Betriebssystemdatei können anschließend die Tabellendaten auf einen anderen Server wie beispielsweise DB2<sup>®</sup> UDB für iSeries<sup>™</sup> versetzt werden.

Folgende Informationen werden benötigt, wenn Daten exportiert werden:

- Eine SQL-SELECT-Anweisung, welche die zu exportierenden Daten angibt.
- Der Pfad und Name der Betriebssystemdatei, in der die exportierten Daten gespeichert werden.
- Das Format der Daten in der Eingabedatei. Mögliche Formate sind: IXF, WSF oder DEL.
- Beim Exportieren von typisierten Tabellen müssen Sie möglicherweise die Traversierfolge für die untergeordneten Tabellen innerhalb der Hierarchie angeben. Bei Verwendung des IXF-Formats wird empfohlen, die Standardreihenfolge zu verwenden. Beachten Sie beim Angeben der Reihenfolge, dass die untergeordneten Tabellen in der vorgegebenen Anordnung (PRE-ORDER) traversiert werden

müssen. Beim Exportieren von typisierten Tabellen können Sie eine SELECT-Anweisung nicht direkt angeben. Stattdessen müssen Sie den Namen der untergeordneten Zieltabelle und (wahlfrei) eine Klausel WHERE angeben. Das Dienstprogramm EXPORT verwendet diese Informationen zusammen mit der Traversierfolge zum Generieren und Ausführen der benötigten SELECT-Anweisung.

Sie können auch Folgendes angeben:

- Neue Spaltennamen (beim Exportieren in IXF- oder WSF-Dateien). Wenn Sie keine neuen Spaltennamen angeben möchten, werden die Spaltennamen in der vorhandenen Tabelle oder Sicht in der exportierten Datei verwendet.
- Zusätzliche Optionen, mit denen die Exportoperation angepasst wird.
- Einen Nachrichtendateinamen. Bei DB2-Operationen wie dem Exportieren, Importieren, Laden, Binden oder Wiederherstellen von Daten können Sie angeben, dass Nachrichtendateien erstellt werden, die Fehlermeldungen, Warnungen und informative Nachrichten zu diesen Operationen enthalten. Geben Sie den Namen dieser Dateien mit dem Parameter MESSAGES an. Diese Nachrichtendateien sind gewöhnliche ASCII-Dateien. Jede Nachricht in einer Nachrichtendatei beginnt in einer neuen Zeile und enthält Informationen, die von der DB2-Funktion zum Abrufen von Nachrichten bereitgestellt werden. Sie können sie mit dem bei Ihrem Betriebssystem üblichen Verfahren drucken oder mit einem beliebigen ASCII-Editor anzeigen.

Wenn Sie das Dienstprogramm EXPORT in einer Datenbankumgebung mit mehreren Partitionen verwenden möchten, können Sie **db2batch** verwenden, um die Task in jeder Datenbankpartition auszuführen. Die SELECT-Anweisung muss in der Lage sein, nur die lokal gefundenen Daten zurückzugeben. Die Auswahlbedingung sieht folgendermaßen aus:

```
SELECT * FROM tabellenname WHERE NODENUMBER(spaltenname) = CURRENT NODE
```

#### Zugehörige Konzepte:

- „Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von EXPORT“ auf Seite 3
- „Verwenden von EXPORT mit Identitätsspalten“ auf Seite 4
- „Erneutes Erstellen einer exportierten Tabelle“ auf Seite 4
- „Exportieren von großen Objekten (LOBS)“ auf Seite 5
- „Paralleles Exportieren von Daten“ auf Seite 6
- „Versetzen von Daten zwischen typisierten Tabellen“ auf Seite 248
- „Beispiele für db2batch-Tests“ in *Systemverwaltung: Optimierung*

#### Zugehörige Tasks:

- „Verwenden von EXPORT“ auf Seite 3

#### Zugehörige Referenzen:

- „Export-API "db2Export"" auf Seite 14
- „db2batch - Benchmark Tool Command“ in *Command Reference*
- „CLP-Beispiele für EXPORT-Sitzungen“ auf Seite 26
- „Dateiformate der Dienstprogramme EXPORT, IMPORT und LOAD“ auf Seite 275
- „Befehl EXPORT“ auf Seite 9

---

## Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von EXPORT

Zugriffsrechte ermöglichen es Benutzern, Datenbankressourcen zu erstellen oder auf sie zuzugreifen. Berechtigungsstufen stellen eine Methode dar, um Berechtigungen sowie Operationen von Dienstprogrammen und zur Datenbankmanagerpflege auf höherer Ebene zu gruppieren. Sie dienen zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte. Benutzer können nur auf solche Objekte zugreifen, für die sie die entsprechende Berechtigung besitzen, d. h., für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigung verfügen.

Sie benötigen die Berechtigung SYSADM oder DBADM bzw. das Zugriffsrecht CONTROL oder SELECT für jede Tabelle, die an dem Exportvorgang beteiligt ist.

### Zugehörige Referenzen:

- „Export-API "db2Export"“ auf Seite 14
- „Befehl EXPORT“ auf Seite 9

---

## Verwenden von EXPORT

### Vorbedingungen:

Bevor Sie das Dienstprogramm EXPORT ausführen, müssen Sie mit der Datenbank, aus der die Daten exportiert werden, verbunden bzw. in der Lage sein, implizit eine Verbindung zur Datenbank herzustellen. Da das Dienstprogramm die Anweisung COMMIT absetzt, sollten Sie vor dem Aufrufen von EXPORT alle Transaktionen beenden und alle Sperren aufheben, indem Sie die Anweisung COMMIT oder ROLLBACK ausführen. Andere Anwendungen, die in separaten Verbindungen auf die Tabelle zugreifen, brauchen die Verbindung nicht zu trennen.

### Einschränkungen:

Für das Dienstprogramm EXPORT gelten die folgenden Einschränkungen:

- Dieses Dienstprogramm bietet keine Unterstützung für Kurznamen.
- Tabellen mit Spalten strukturierten Typs werden nicht unterstützt.

### Prozedur:

| *Das Dienstprogramm EXPORT kann über den Befehlszeilenprozessor (CLP), das Notiz-*  
| *buch "Exportieren" in der Steuerzentrale oder eine Anwendungsprogrammierschnittstelle*  
| *(API) namens **db2Export** aufgerufen werden.*

Es folgt ein Beispiel für den Befehl EXPORT, der über den CLP abgesetzt wird:

```
db2 export to staff.ixf of ixf select * from userid.staff
```

So öffnen Sie das Notizbuch **Exportieren**:

1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis Sie den Ordner **Tabellen** oder **Sichten** sehen.
2. Klicken Sie den Ordner an, mit dem Sie arbeiten möchten. Alle vorhandenen Tabellen oder Sichten werden auf der rechten Seite des Fensters, dem Inhaltsteilfenster, angezeigt.

3. Klicken Sie im Inhaltsteilfenster die gewünschte Tabelle oder Sicht mit Maustaste 2 an, und wählen Sie **Exportieren** im Kontextmenü aus. Das Notizbuch **Exportieren** wird geöffnet.

Ausführliche Informationen zur Steuerzentrale finden Sie in der Onlinehilfefunktion der Steuerzentrale.

**Zugehörige Referenzen:**

- „Export-API "db2Export"“ auf Seite 14

**Zugehörige Beispiele:**

- „tbmove.out -- HOW TO MOVE TABLE DATA (C)“
- „tbmove.sqc -- How to move table data (C)“
- „tbmove.out -- HOW TO MOVE TABLE DATA (C++)“
- „tbmove.sqC -- How to move table data (C++)“

---

## Verwenden von EXPORT mit Identitätsspalten

Mit dem Dienstprogramm EXPORT können Sie Daten aus einer Tabelle exportieren, die eine Identitätsspalte enthält. Hat die SELECT-Anweisung, die für die Exportoperation angegeben wird, das Format "select \* from tabellenname" und wird die Option METHOD nicht verwendet, wird das Exportieren der Identitätsspaltenmerkmale in IXF-Dateien unterstützt. Anschließend können die Optionen REPLACE\_CREATE und CREATE des Befehls IMPORT verwendet werden, um die Tabelle einschließlich ihrer Identitätsspaltenmerkmale erneut zu erstellen. Wird eine IXF-Datei aus einer Tabelle mit einer als GENERATED ALWAYS definierten Identitätsspalte erstellt, kann die Datendatei nur dann erfolgreich importiert werden, wenn der Änderungswert `identityignore` angegeben wird. Andernfalls werden alle Zeilen zurückgewiesen (SQL3550W).

**Zugehörige Konzepte:**

- „Identitätsspalten“ in *Systemverwaltung: Konzept*

---

## Erneutes Erstellen einer exportierten Tabelle

Eine Tabelle kann durch Verwenden des Dienstprogramms EXPORT und Angabe des IXF-Dateiformats gesichert werden. Die gesicherte Tabelle (einschließlich der Indizes) kann dann mit Hilfe des Dienstprogramms IMPORT erneut erstellt werden.

Die Exportoperation schlägt fehl, wenn der verfügbare Speicherbereich auf dem Dateisystem, auf dem die exportierte Datei erstellt wird, für die zu exportierenden Daten nicht ausreicht. In diesem Fall müssen Sie die Menge der ausgewählten Daten durch Angeben von Bedingungen in der Klausel WHERE begrenzen, damit die Exportdatei auf dem Zielsystem Platz findet. Sie können das Dienstprogramm EXPORT mehrmals aufrufen, um alle Daten zu exportieren.

Die DEL- und ASC-Dateiformate enthalten zwar die Satzdaten, jedoch keine Beschreibungen der Zieltabelle. Um eine Tabelle mit Daten erneut zu erstellen, die in diesen Dateiformaten gespeichert wurden, erstellen Sie die Zieltabelle und verwenden anschließend das Dienstprogramm LOAD oder IMPORT, um die Tabelle aus diesen Dateien zu füllen. Mit **db2look** (DB2 Statistics and DDL Extraction Tool)

können Sie die ursprünglichen Tabellendefinitionen erfassen und die entsprechende Datendefinitionssprache generieren.

**Zugehörige Konzepte:**

- „Erneutes Erstellen einer exportierten Tabelle mit IMPORT“ auf Seite 35

**Zugehörige Referenzen:**

- „db2look - DB2 Statistics and DDL Extraction Tool Command“ in *Command Reference*

---

## Exportieren von großen Objekten (LOBs)

Beim Exportieren von Daten aus LOB-Spalten besteht die Standardaktion darin, die ersten 32 KB Daten auszuwählen und diese in die gleiche Datei wie die restlichen Spaltendaten zu stellen.

**Anmerkung:** Das IXF-Dateiformat speichert die LOB-Optionen der Spalte (beispielsweise über die Protokollierung der LOB-Spalte) nicht. Dies bedeutet, dass das Dienstprogramm IMPORT eine Tabelle mit einer LOB-Spalte, deren Größe mit 1 GB oder mehr definiert ist, nicht erneut erstellen kann.

Mit einer LOB-Positionskenung (LOB Location Specifier - LLS) können mehrere große Objekte (LOB) beim Exportieren von LOB-Informationen in einer gemeinsamen Datei gespeichert werden. Wird beim Exportieren der Daten der Änderungswert `lobsinfile` verwendet, wählt das Dienstprogramm EXPORT die gesamte LOB-Datei aus und stellt sie in eine der LOB-Dateien. Unter Umständen kann es mehrere große Objekte (LOBs) pro LOB-Datei und mehrere LOB-Dateien in einem LOB-Pfad geben. Die Datendatei enthält die LLS-Datensätze.

Eine LOB-Positionskennung ist eine Zeichenfolge, die angibt, wo sich LOB-Daten in einer Datei befinden. Das Format der LOB-Positionskennung lautet `dateiname.erw.nnn.mmm/`. Hierbei steht `dateiname.erw` für den Namen der Datei, die das große Objekt (LOB) enthält, `nnn` für die (in Byte angegebene) relative Position des großen Objekts in der Datei und `mmm` für die Länge des großen Objekts (in Byte). Die LOB-Positionskennung `db2exp.001:123:456/` gibt beispielsweise an, dass sich das große Objekt in der Datei `db2exp.001` befindet, an einer relativen Position von 123 Byte in der Datei beginnt und 456 Byte lang ist. Wenn in der LOB-Positionskennung die Größe 0 angegeben ist, wird angenommen, dass das große Objekt die Länge 0 hat. Lautet der Wert für die Länge -1, wird angenommen, dass das große Objekt NULL ist, und relative Position sowie Dateiname werden ignoriert.

**Zugehörige Referenzen:**

- „Export-API "db2Export"“ auf Seite 14
- „Befehl EXPORT“ auf Seite 9
- „Large objects (LOBs)“ in *SQL Reference, Volume 1*

---

## Paralleles Exportieren von Daten

Durch das parallele Exportieren von Daten wird die Datenübertragung reduziert, und das Schreiben der Ergebnismenge wird - ebenso wie die Generierung der formatierten Ausgabe - in einer effizienteren Weise auf die Knoten verteilt, als dies bei einer Exportoperation ohne Parallelverarbeitung der Fall wäre. Beim parallelen Exportieren von Daten (durch das Aufrufen mehrerer Exportoperationen, nämlich einer Operation für jede Partition einer Tabelle) werden die Daten extrahiert, auf den lokalen Knoten konvertiert und anschließend in das lokale Dateisystem geschrieben. Im Gegensatz hierzu werden die Daten beim seriellen Exportieren (also beim Exportieren durch eine einzige Exportoperation) parallel extrahiert und anschließend an den Client geliefert, auf dem ein Einzelprozess die Konvertierung vornimmt und die Ergebnismenge in ein lokales Dateisystem schreibt.

Mit dem Befehl **db2batch** können die Leistungskennndaten und die Ausführungsdauer von SQL-Anweisungen überwacht werden. Dieses Dienstprogramm enthält außerdem eine Funktion für das parallele Exportieren in Umgebungen mit partitionierten Datenbanken, das Folgendes ausführt:

- Es werden Abfragen ausgeführt, um die zu exportierenden Daten zu definieren.
- Auf jeder Partition wird eine Datei mit den exportierten Daten, die sich auf dieser Partition befinden, erstellt.

**Anmerkung:** Wird der Zieldateipfad von mehreren Datenbankpartitionen gemeinsam benutzt, ist die Ausgabe aller Partitionen, die den Pfad gemeinsam benutzen, in einer einzigen Datei enthalten.

Für jede Partition wird parallel eine Abfrage ausgeführt, um die Daten auf dieser Partition abzurufen. Wird der Befehl **db2batch** mit der Option `-p s` ausgeführt, wird die ursprüngliche SELECT-Abfrage parallel ausgeführt. Wird der Befehl **db2batch** entweder mit der Option `-p t` oder der Option `-p d` ausgeführt, wird eine Zwischenspeichertabelle unter Verwendung der angegebenen Abfrage mit den Exportdaten geladen, und eine Abfrage `SELECT *` wird für die Zwischenspeichertabelle parallel auf jeder Partition ausgeführt, um die Daten zu exportieren. Um lediglich die Daten zu exportieren, die sich auf einer bestimmten Partition befinden, fügt der Befehl "db2batch" das Prädikat `NODENUMBER(spaltenname) = CURRENT NODE` zur Klausel `WHERE` der Abfrage hinzu, die für diese Partition ausgeführt wird. Für den Parameter "spaltenname" muss der qualifizierte Name oder der nicht qualifizierte Name einer Tabellenspalte angegeben werden. Um diesen Parameter festzulegen, wird der erste Spaltenname in der ursprünglichen Abfrage verwendet.

Es ist wichtig zu verstehen, dass der Befehl **db2batch** eine SQL-Abfrage ausführt und die Ausgabe an die Zieldatei sendet. Der Befehl verwendet nicht das Dienstprogramm `EXPORT`. Die Optionen des Dienstprogramms `EXPORT` gelten nicht für das parallele Exportieren mit dem Befehl **db2batch**. Mit dem Befehl **db2batch** können keine LOB-Spalten exportiert werden.

Um eine vollständige Beschreibung der Befehloptionen anzuzeigen, führen Sie den Befehl **db2batch** mit der Option `-h` über das Befehlsfenster aus.

Der Befehl **db2batch** führt eine parallele SQL-Abfrage aus und sendet die Ausgabe an eine angegebene Datei. Hierbei ist zu beachten, dass der Befehl eine `SELECT`-Anweisung und nicht das Dienstprogramm `EXPORT` ausführt. LOB-Spalten (unabhängig von der Datenlänge) können mit dieser Methode nicht exportiert werden.

Um Inhalte aus der Tabelle 'staff' parallel zu exportieren, verwenden Sie folgenden Befehl:

```
db2batch -p s -d sample -f staff.batch -r /home/userid/staff.asc -q on
```

In diesem Beispiel gilt:

- Die Abfrage wird parallel für eine einzelne Tabelle ausgeführt (Option -p s).
- Eine Verbindung wird zur Beispieldatenbank ('sample') hergestellt (Option -d sample).
- Die Steuerdatei staff.batch enthält die mit einem Semikolon endende SQL-SELECT-Anweisung `select * from staff;`.
- Die Ausgabe wird in den Dateien /home/userid/staff.asc gespeichert, auf die auf jeder Datenbankpartition Zugriff besteht. Das Standardausgabeformat ist positionsgebundenes ASCII. Weiterhin ist zu beachten, dass der Befehl 'db2batch' nicht das Dienstprogramm EXPORT verwendet.
- Nur die Ausgabe der Abfrage wird an die Datei gesendet (Option -q).

Um Inhalte in eine ASCII-Datei ohne universellen Zeilenbegrenzer zu exportieren, verwenden Sie den folgenden Befehl:

```
db2batch -p s -d sample -f emp_resume.batch -r /home/userid/emp_resume.del,  
/home/mmilek/userid/emp_resume.out -q del
```

In diesem Beispiel gilt:

- Es werden lediglich Nicht-LOB-Spalten aus der Tabelle emp\_resume ausgewählt (die Steuerdatei emp\_resume.batch enthält `select empno,resume_format from emp_resume;`).
- Die Datei emp\_resume.del enthält die Abfrageausgabe im ASCII-Format ohne universelle Zeilenbegrenzer (Option -q del), das Komma ist der Standardspaltenbegrenzer und | ist der Standardzeichenbegrenzer.
- Die Datei emp\_resume.out enthält die Abfragestatistik.

Abhängig davon, welche Tabellen abgefragt werden und wo sich diese Tabellen in der Umgebung mit partitionierten Datenbanken befinden, stehen für das parallele Exportieren zwei Methoden zur Verfügung:

- Das parallele Exportieren aus einer Partitionstabelle bzw. eine Verknüpfung oder Unterabfrage für mehrere durch Kollokation zusammengefassten Tabellen (geben Sie hierzu die Option -p s für den Befehl **db2batch** an).

Es gibt zwei Situationen, in denen Tabellen als durch Kollokation zusammengefasst betrachtet werden können:

- Die Tabellen befinden sich in einer Einzeldatenbankpartitionsgruppe, die auf derselben Partition definiert ist.
- Die Tabellen gehören zu der gleichen Datenbankpartitionsgruppe und enthalten Partitionierungsschlüssel mit identischer Spaltenanzahl und identischem Spaltentyp. Die entsprechenden Spalten des Partitionierungsschlüssels sind partitionskompatibel, und die Tabellen werden hinsichtlich des gesamten Partitionierungsschlüssels oder einer Obermenge des Partitionierungsschlüssels mit Gleichheitsattribut (equijoin) verknüpft.

In beiden Fällen kann die Abfrage für jede Partition ausgeführt werden, um die Exportdatendatei dieser Partition zu generieren. Zu diesem Zweck wird die Funktion NODENUMBER verwendet, die im Folgenden beschrieben wird. (Bitte beachten Sie, dass das parallele Exportieren zurückgewiesen wird, wenn eine Tabelle nur in einer einzigen Partition vorhanden ist, da die Daten nur aus einer

Partition abgerufen werden. Informationen dazu, wie Sie in einem solchen Fall das parallele Exportieren aktivieren können, finden Sie unter dem nächsten Listenpunkt.)

- Eine SELECT-Anweisung, die für mehrere nicht durch Kollokation zusammengefasste Tabellen ausgeführt wird. (Geben Sie im Befehl **db2batch** die Option `-p t tabellenname` oder die Option `-p d` an. Mit der ersten Option können Sie angeben, dass eine vorhandene Tabelle als Zwischenspeichertabelle verwendet werden soll. Die zweite Option bewirkt, dass das Dienstprogramm EXPORT eine Zwischenspeichertabelle erstellt.)

Das Dienstprogramm EXPORT verwendet eine Zwischenspeichertabelle, die durch die Exportabfrage gefüllt wird. Diese Zwischenspeichertabelle wird verwendet, um die Zeilen der "Exportergebnismenge" zu lokalisieren, indem eine Anweisung INSERT für die Gesamtauswahlabfrage in der Zwischenspeichertabelle ausgeführt wird. Nachdem die Zwischenspeichertabelle erstellt wurde, generiert das Dienstprogramm EXPORT auf jeder Partition eine Exportdatendatei. Zu diesem Zweck wird die Anweisung

```
"select * WHERE NODENUMBER(spaltenname) = CURRENT NODE"
```

für die Zwischenspeichertabelle ausgeführt.

Eine Zwischenspeichertabelle kann außerdem zum parallelen Exportieren einer Einzelpartitionstabelle verwendet werden. In den meisten Fällen ist das Übertragen der Daten aus einer Zwischenspeichertabelle mit Einzelpartition in eine Zwischenspeichertabelle mit mehreren Partitionen und das anschließende parallele Exportieren der Zwischenspeichertabelle in alle Partitionen schneller als das serielle Exportieren der Einzelpartitionstabelle.

Das Dienstprogramm EXPORT führt parallel für jede Partition eine Abfrage aus, um die Daten auf dieser Partition abzurufen. Bei Verwendung des Befehls `db2batch -p s` wird die ursprüngliche SELECT-Abfrage parallel ausgeführt. Wenn der Befehl `db2batch -p t` bzw. der Befehl `db2batch -p d` verwendet wird, wird eine Zwischenspeichertabelle unter Verwendung der angegebenen Abfrage mit den Exportdaten geladen, und eine Abfrage `SELECT *` wird für die Zwischenspeichertabelle parallel auf jeder Partition ausgeführt, um die Daten zu exportieren. Um lediglich die Daten zu exportieren, die sich auf einer bestimmten Partition befinden, fügt der Befehl **db2batch** das Prädikat `NODENUMBER(spaltenname) = CURRENT NODE` zur Klausel WHERE der Abfrage hinzu, die für diese Partition ausgeführt wird. Für den Parameter *spaltenname* muss der qualifizierte Name oder der Namen ohne Qualifikationsmerkmal einer Tabellenspalte angegeben werden. Das Dienstprogramm EXPORT verwendet den ersten Spaltennamen in der ursprünglichen Abfrage, um diesen Parameter festzulegen.

Für Abfragen, die durch das Dienstprogramm EXPORT verwendet werden, gelten die folgenden Einschränkungen:

- Bei Angabe von `db2batch -p s` darf die Abfrage nicht nur Spaltenfunktionen enthalten, da für das Prädikat `NODENUMBER spaltenname` ein Spaltenname benötigt wird.
- Bei Angabe von `db2batch -p s` müssen die Ergebnistabellen (wie z. B. MIN, MAX und AVG) auf einer Gruppierung basieren, die den Partitionierungsschlüssel enthält.
- Bei Angabe von `db2batch -p t` oder `db2batch -p d` darf die Abfrage keine Klausel ORDER BY enthalten, da ORDER BY für eine Gesamtauswahl innerhalb einer Anweisung INSERT durch DB2<sup>®</sup> UDB nicht unterstützt wird.

Wenn Sie bei Angabe von `-p s` im Befehl **db2batch** zusätzlich die Option `-r` verwenden, um Ergebnisausgabedateien zu erstellen, werden die Dateien auf jeder Partition sortiert, sofern Sie eine Klausel `ORDER BY` angegeben haben. Soll eine einzige sortierte Datei erstellt werden, fügen Sie die sortierten Dateien aus allen Partitionen in einer sortierten Datei zusammen. Verwenden Sie beispielsweise auf Systemen mit UNIX®-Basis den Befehl `sort -m`, um die Dateien in einer einzigen sortierten Datei zusammenzufügen. Falls Sie Ihre Ausgabe an ein angehängtes NFS-Dateisystem senden, wird die Ausgabe nicht sortiert, und zwar selbst dann nicht, wenn Sie die Klausel `ORDER BY` angeben.

**Zugehörige Konzepte:**

- „EXPORT - Übersicht“ auf Seite 1

**Zugehörige Referenzen:**

- „db2batch - Benchmark Tool Command“ in *Command Reference*

## Befehl EXPORT

Exportiert Daten aus einer Datenbank in eines von mehreren externen Dateiformaten. Der Benutzer gibt die zu exportierenden Daten durch eine SQL-SELECT-Anweisung oder über hierarchische Informationen für typisierte Tabellen an.

**Berechtigung:**

Eine der Folgenden:

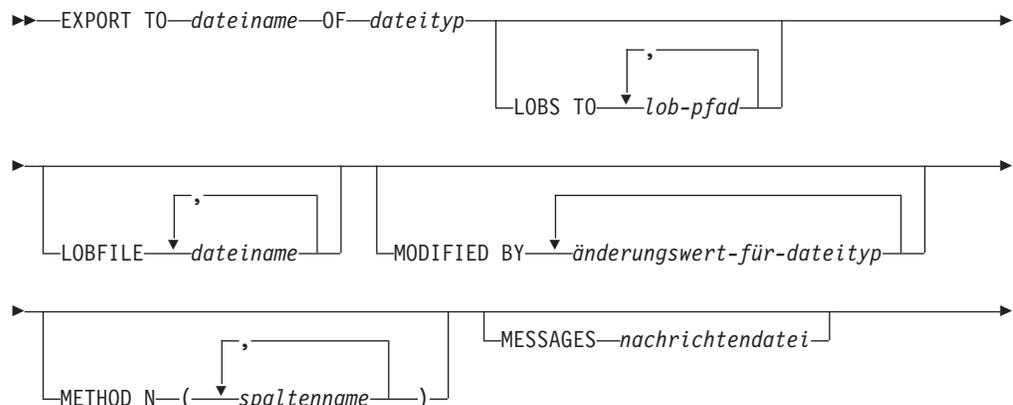
- `sysadm`
- `dbadm`

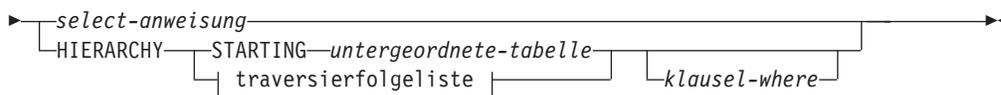
oder Zugriffsrecht `CONTROL` bzw. `SELECT` für jede betroffene Tabelle oder Sicht.

**Erforderliche Verbindung:**

Datenbank. Wenn das implizite Herstellen von Verbindungen aktiviert ist, wird eine Verbindung zur Standarddatenbank hergestellt.

**Befehlssyntax:**





## traversierfolgeliste:



## Befehlsparameter:

### HIERARCHY traversierfolgeliste

Exportiert eine untergeordnete Hierarchie mit der angegebenen Traversierfolge. Alle untergeordneten Tabellen müssen in PRE-ORDER-Weise aufgelistet sein. Der Namen der ersten untergeordneten Tabelle wird als Zieltabellenname für die SELECT-Anweisung verwendet.

### HIERARCHY STARTING untergeordnete-tabelle

Exportiert eine untergeordnete Hierarchie unter Verwendung der hierarchischen Standardreihenfolge (OUTER-Reihenfolge für ASC-, DEL- oder WSF-Dateien oder die in PC/IXF-Datendateien gespeicherte Reihenfolge), beginnend mit *untergeordnete-tabelle*.

### LOBBFILE dateiname

Gibt einen oder mehrere Basisdateinamen für LOB-Dateien an. Wenn der Namensbereich für den ersten Namen erschöpft ist, wird der zweite Name verwendet, und so weiter.

Bei der Erstellung von LOB-Dateien während einer Exportoperation werden die Dateinamen erzeugt, indem der aktuelle Basisdateiname aus dieser Liste an den aktuellen Pfad (aus *lob-pfad*) und daran wiederum eine dreistellige Folgenummer angefügt wird. Beispiel: Wenn der aktuelle LOB-Pfad das Verzeichnis */u/foo/lob/pfad/* und der aktuelle LOB-Dateiname *bar* ist, heißen die erstellten LOB-Dateien */u/foo/lob/pfad/bar.001*, */u/foo/lob/pfad/bar.002* und so weiter.

### LOBS TO lob-pfad

Gibt einen oder mehrere Pfade zu Verzeichnissen an, in denen die LOB-Dateien gespeichert werden sollen. Jeder LOB-Pfad enthält mindestens eine Datei, und jede Datei enthält mindestens ein großes Objekt (LOB).

### MESSAGES nachrichtendatei

Gibt das Ziel für Warnungen und Fehlermeldungen an, die bei einer Exportoperation auftreten. Wenn die Datei bereits vorhanden ist, werden die Informationen vom Dienstprogramm EXPORT an sie angefügt. Wenn *nachrichtendatei* weggelassen wird, werden die Nachrichten in die Standardausgabe geschrieben.

### METHOD N spaltenname

Gibt einen oder mehrere Spaltennamen an, die in der Ausgabedatei verwendet werden sollen. Wenn dieser Parameter nicht angegeben wird, werden die Spaltennamen in der Tabelle verwendet. Dieser Parameter ist nur für WSF- und IXF-Dateien gültig. Er ist nicht gültig beim Exportieren von hierarchischen Daten.

### MODIFIED BY änderungswert-für-dateityp

Gibt Optionen für Änderungswerte für Dateityp an. Siehe File type modifiers for export.

**OF dateityp**

Gibt das Format der Daten in der Ausgabedatei an:

- DEL (ASCII-Format ohne universelle Zeilenbegrenzer), das von einer Vielzahl von Datenbankmanager- und Dateimanagerprogrammen verwendet wird.
- WSF (Arbeitsblattformat), das von Programmen wie den folgenden verwendet wird:
  - Lotus 1-2-3
  - Lotus Symphony

**Anmerkung:** Beim Exportieren von BIGINT- oder DECIMAL-Daten können nur Werte, die innerhalb des Bereichs des Typs DOUBLE liegen, korrekt exportiert werden. Es werden zwar auch Werte exportiert, die nicht in diesem Bereich liegen, beim Zurückimportieren oder -laden dieser Werte können jedoch je nach Betriebssystem falsche Daten entstehen.

- IXF (Integrated Exchange Format, PC-Version). In diesem Format werden die meisten Tabellenattribute sowie vorhandene Indizes in der IXF-Datei gesichert, sofern keine Spalten in der SELECT-Anweisung angegeben werden. Bei diesem Format kann die Tabelle erneut erstellt werden, während sie bei den anderen Formaten bereits vorhanden sein muss, damit Daten in sie importiert werden können.

**select-anweisung**

Gibt die SELECT-Anweisung an, welche die zu exportierenden Daten zurückgibt. Wenn die SELECT-Anweisung zu einem Fehler führt, wird eine Nachricht in die Nachrichtendatei (oder in die Standardausgabe) geschrieben. Bei dem Fehlercode SQL0012W, SQL0347W, SQL0360W, SQL0437W oder SQL1824W wird die Exportoperation fortgesetzt, andernfalls wird sie gestoppt.

**TO dateiname**

Gibt den Namen der Datei an, in welche die Daten exportiert werden sollen. Wenn der vollständige Pfad zu der Datei nicht angegeben wird, verwendet das Dienstprogramm EXPORT das aktuelle Verzeichnis und das Standardlaufwerk als Ziel.

Wenn der Name einer Datei, die bereits vorhanden ist, angegeben wird, überschreibt das Dienstprogramm EXPORT den Inhalt der Datei. Die Informationen werden nicht angefügt.

**Beispiele:**

Das folgende Beispiel zeigt, wie Daten aus der Tabelle STAFF der Beispieldatenbank SAMPLE in die Datei myfile.ixf exportiert werden. Die Ausgabe wird im IXF-Format erstellt. Bitte beachten Sie, dass eine Verbindung zur Datenbank SAMPLE bestehen muss, bevor der Befehl abgesetzt wird. Die Indexdefinitionen werden (falls vorhanden) in der Ausgabedatei gespeichert. Hiervor ausgenommen sind Fälle, in denen die Datenbankverbindung über DB2 Connect hergestellt wurde.

```
db2 export to myfile.ixf of ixf messages msgs.txt select * from staff
```

Das folgende Beispiel zeigt, wie Daten über Mitarbeiter in Abteilung 20 aus der Tabelle STAFF der Beispieldatenbank SAMPLE exportiert werden. Die Ausgabe wird im IXF-Format erstellt und in die Datei awards.ixf gestellt.

## Befehl EXPORT

Bitte beachten Sie, dass zunächst eine Verbindung zur Datenbank SAMPLE hergestellt werden muss, bevor der Befehl abgesetzt wird. Außerdem ist zu beachten, dass der tatsächliche Spaltenname in der Tabelle "dept" statt "department" lautet.

```
db2 export to awards.ixf of ixf messages msgs.txt select * from staff
where dept = 20
```

Das folgende Beispiel zeigt, wie große Objekte in eine DEL-Datei exportiert werden:

```
| db2 export to myfile.del of del lobs to mylobs/
| lobfile lobs1, lobs2 modified by lobsinfile
| select * from emp_photo
```

Das folgende Beispiel zeigt, wie große Objekte in eine DEL-Datei exportiert werden, wobei ein zweites Verzeichnis für Dateien angegeben wird, die im ersten Verzeichnis möglicherweise keinen Platz haben:

```
| db2 export to myfile.del of del
| lobs to /db2exp1/, /db2exp2/ modified by lobsinfile
| select * from emp_photo
```

Das folgende Beispiel zeigt den Export von Daten in eine DEL-Datei, wobei ein einfaches Anführungszeichen als Zeichenfolgebegrenzer, ein Semikolon als Spaltenbegrenzer und ein Komma als Dezimalzeichen verwendet wird. Die gleiche Konvention muss beim Rückimportieren der Daten in die Datenbank verwendet werden:

```
db2 export to myfile.del of del
modified by chardel'' coldel; decpt,
select * from staff
```

### Anmerkungen zur Verwendung:

Achten Sie darauf, alle Tabellenoperationen abzuschließen und alle Sperren freizugeben, bevor Sie eine Exportoperation starten. Dies erreichen Sie, indem Sie eine COMMIT-Operation absetzen, nachdem Sie alle geöffneten Cursor mit WITH HOLD geschlossen haben, oder indem Sie eine ROLLBACK-Operation absetzen.

Die Verwendung von Tabellenaliasnamen in der SELECT-Anweisung ist zulässig.

Die in die Nachrichtendatei gestellten Nachrichten enthalten die Informationen, die vom Service für die Nachrichtenabfrage zurückgegeben werden. Jede Nachricht beginnt in einer neuen Zeile.

Das Dienstprogramm EXPORT erzeugt immer dann eine Warnung, wenn eine Spalte des Typs CHARACTER mit einer größeren Länge als 254 für das Exportieren in Dateien mit dem DEL-Format ausgewählt wird.

Zum Versetzen von Daten zwischen Datenbanken sollte der PC/IXF-Import verwendet werden. Falls Zeichendaten, die Zeilentrennzeichen enthalten, in eine ASCII-Datei ohne universelle Zeilenbegrenzer (DEL) exportiert und durch ein Textübertragungsprogramm verarbeitet werden, werden Felder, die die Zeilentrennzeichen enthalten, verkleinert oder erweitert.

Der Schritt zum Kopieren der Datei ist nicht erforderlich, wenn sowohl auf die Quelldatenbank als auch auf die Zieldatenbank von demselben Client aus zugegriffen werden kann.

Mit DB2 Connect können Tabellen von DRDA-Servern wie DB2 für OS/390, DB2 für VM und VSE sowie DB2 für OS/400 exportiert werden. Nur der PC/IXF-Export wird unterstützt.

Wenn das Dienstprogramm EXPORT über ein AIX-System aufgerufen wird, werden keine mehrteiligen PC/IXF-Dateien erstellt.

Das Dienstprogramm EXPORT speichert das Attribut NOT NULL WITH DEFAULT der Tabelle in einer IXF-Datei, sofern die SELECT-Anweisung im Format `SELECT * FROM tabellenname` angegeben wird.

Beim Exportieren von typisierten Tabellen können Anweisungen für eine Unterauswahl nur dadurch ausgedrückt werden, indem der Name der Zieltabelle und die Klausel WHERE angegeben wird. Beim Exportieren einer Hierarchie können Gesamtauswahl und *select-anweisung* nicht angegeben werden.

Bei anderen Dateiformaten als IXF empfiehlt es sich, die Traversierfolgeliste anzugeben, weil diese Liste DB2 mitteilt, wie die Hierarchie traversiert werden muss und welche untergeordneten Tabellen exportiert werden sollen. Wenn diese Liste nicht angegeben wird, werden alle Tabellen in der Hierarchie exportiert und die Standardreihenfolge lautet OUTER. Alternativ kann die Standardreihenfolge verwendet werden, die durch die Funktion OUTER vorgegeben wird.

**Anmerkung:** Während einer Importoperation sollte die gleiche Traversierfolge verwendet werden. Das Dienstprogramm LOAD unterstützt das Laden von Hierarchien oder Unterhierarchien nicht.

### Hinweise zu DB2 Data Links Manager:

So können Sie sicherstellen, dass eine konsistente Kopie der Tabelle und der entsprechenden Dateien, auf die in den DATALINK-Spalten verwiesen wird, für den Export kopiert wird:

1. Setzen Sie den Befehl **QUIESCE TABLESPACES FOR TABLE tabellenname SHARE** ab.  
Hierdurch wird sichergestellt, dass bei der Ausführung von EXPORT keine Aktualisierungstransaktionen verarbeitet werden.
2. Setzen Sie den Befehl EXPORT ab.
3. Führen Sie das Dienstprogramm **dlfm\_export** auf jedem Data Links-Server aus.  
Die Eingabe für das Dienstprogramm **dlfm\_export** ist der Name der Steuerdatei, der durch das Dienstprogramm EXPORT generiert wird. Hierdurch wird ein TAR-Archiv (oder ein äquivalentes Archiv) der Dateien erzeugt, die in der Steuerdatei aufgelistet sind.
4. Setzen Sie den Befehl **QUIESCE TABLESPACES FOR TABLE tabellenname RESET** ab.  
Hierdurch wird die Tabelle für Aktualisierungen verfügbar gemacht.

EXPORT wird als SQL-Anwendung ausgeführt. Die Zeilen und Spalten, die die Bedingungen der SELECT-Anweisung erfüllen, werden aus der Datenbank extrahiert. Für DATALINK-Spalten darf in der SELECT-Anweisung keine Skalarfunktion angegeben werden.

## Befehl EXPORT

Bei einer erfolgreichen Ausführung von EXPORT werden die folgenden Dateien generiert:

- Die Exportdatendatei, die im Befehl EXPORT angegeben wurde. Ein DATA-LINK-Spaltenwert in dieser Datei hat das gleiche Format wie das von den Dienstprogrammen LOAD und IMPORT verwendete. Wenn der DATALINK-Spaltenwert der SQL-NULL-Wert ist, ist die Verarbeitung mit den anderen Datentypen identisch.
- Steuerdateien *servername*, die für jeden Data Links-Server generiert werden. Bei Windows-Betriebssystemen wird eine einzige Steuerdatei *ctrlfile.lst* durch alle Data Links-Server verwendet. Diese Steuerdateien werden in das Verzeichnis `<datendateipfad>/dlfm/JJJJMMTT/HHMMSS` gestellt (beim Betriebssystem Windows wird die Datei *ctrlfile.lst* in das Verzeichnis `<datendateipfad>\dlfm\JJJJMMTT\HHMMSS` gestellt). Hierbei steht JJJJMMTT für das Datum (Jahr Monat Tag), und HHMMSS gibt die Uhrzeit an (Stunde Minute Sekunde).

Mit dem Dienstprogramm **dlfm\_export** können Dateien von einem Data Links-Server exportiert werden. Dieses Dienstprogramm generiert eine Archivdatei, die zur Wiederherstellung von Dateien auf dem Data Links-Zielserverserver verwendet werden kann.

### Zugehörige Konzepte:

- „EXPORT - Übersicht“ auf Seite 1
- „Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von EXPORT“ auf Seite 3

### Zugehörige Tasks:

- „Verwenden von EXPORT“ auf Seite 3

### Zugehörige Referenzen:

- „Export-API "db2Export"“ auf Seite 14
- „CLP-Beispiele für EXPORT-Sitzungen“ auf Seite 26
- „Änderungswerte für Dateityp für EXPORT“ auf Seite 22
- „Einschränkungen für Begrenzer beim Versetzen von Daten“ auf Seite 247

---

## Export-API "db2Export"

Exportiert Daten aus einer Datenbank in eines von mehreren externen Dateiformaten. Der Benutzer gibt die zu exportierenden Daten durch eine SQL-SELECT-Anweisung oder über hierarchische Informationen für typisierte Tabellen an.

### Berechtigung:

Eine der Folgenden:

- *sysadm*
- *dbadm*

oder Zugriffsrecht CONTROL bzw. SELECT für jede betroffene Tabelle oder Sicht.

### Erforderliche Verbindung:

Datenbank. Wenn das implizite Herstellen von Verbindungen aktiviert ist, wird eine Verbindung zur Standarddatenbank hergestellt.

**API-Include-Datei:***db2ApiDf.h***C-API-Syntax:**

```

/* File: db2ApiDf.h */
/* API: db2Export */
/* ... */

SQL_API_RC SQL_API_FN
db2Export (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2ExportStruct
{
    char *piDataFileName;
    struct sqlu_media_list *piLobPathList;
    struct sqlu_media_list *piLobFileList;
    struct sqldcol *piDataDescriptor;
    struct sqllob *piActionString;
    char *piFileType;
    struct sqlchar *piFileTypeMod;
    char *piMsgFileName;
    db2int16 iCallerAction;
    struct db2ExportOut *poExportInfoOut;
} db2ExportStruct;

typedef SQL_STRUCTURE db2ExportOut
{
    db2UInt64 oRowsExported;
} db2ExportOut;
/* ... */

```

**Generische API-Syntax:**

```

/* File: db2ApiDf.h */
/* API: db2gExport */
/* ... */

SQL_API_RC SQL_API_FN
db2gExport (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2gExportStruct
{
    char *piDataFileName;
    struct sqlu_media_list *piLobPathList;
    struct sqlu_media_list *piLobFileList;
    struct sqldcol *piDataDescriptor;
    struct sqllob *piActionString;
    char *piFileType;
    struct sqlchar *piFileTypeMod;
    char *piMsgFileName;
    db2int16 iCallerAction;
    struct db2ExportOut *poExportInfoOut;
    db2UInt16 iDataFileNameLen;
    db2UInt16 iFileTypeLen;
    db2UInt16 iMsgFileNameLen;
} db2gExportStruct;
/* ... */

```

## Export-API "db2Export"

### API-Parameter:

#### versionNumber

Eingabe. Gibt die Version und den Releasestand der Struktur an, die als zweiter Parameter, *pParmStruct*, übergeben wird.

#### pParmStruct

Eingabe. Ein Zeiger auf die *db2ExportStruct*-Struktur.

#### pSqlca

Ausgabe. Ein Zeiger auf die *sqlca*-Struktur.

#### iDataFileNameLen

Eingabe. Eine 2 Byte große ganze Zahl ohne Vorzeichen, die die Länge des Namens der Datendatei in Byte darstellt.

#### iFileTypeLen

Eingabe. Eine 2 Byte große ganze Zahl ohne Vorzeichen, die die Länge des Dateityps in Byte darstellt.

#### iMsgFileNameLen

Eingabe. Eine 2 Byte große ganze Zahl ohne Vorzeichen, die die Länge des Namens der Nachrichtendatei in Byte darstellt.

#### piDataFileName

Eingabe. Eine Zeichenfolge, die den Pfad und den Namen einer externen Datei enthält, in die die Daten exportiert werden sollen.

#### piLobPathList

Eingabe. Eine *sqlu\_media\_list* unter Verwendung von *media\_type* `SQLU_LOCAL_MEDIA` und der Struktur *sqlu\_media\_entry*, die Pfade auf dem Client angibt, in denen die LOB-Dateien gespeichert werden sollen. Wenn der Dateibereich im ersten Pfad in dieser Liste erschöpft ist, wird von der API der zweite Pfad verwendet, und so weiter.

#### piLobFileList

Eingabe. Eine *sqlu\_media\_list* unter Verwendung von *media\_type* `SQLU_CLIENT_LOCATION` und der Struktur *sqlu\_location\_entry*, die Basisdateinamen enthält.

Wenn der Namensbereich bei Verwendung des ersten Namens in dieser Liste erschöpft ist, wird von der API der zweite Name verwendet, und so weiter.

Bei der Erstellung von LOB-Dateien während einer Exportoperation werden die Dateinamen erzeugt, indem der aktuelle Basisdateiname aus dieser Liste an den aktuellen Pfad (aus *pLobFilePath*) und daran wiederum eine dreistellige Folgenummer angefügt wird. Beispiel: Wenn der aktuelle LOB-Pfad das Verzeichnis `/u/foo/lob/pfad` und der aktuelle LOB-Dateiname `bar` ist, heißen die erstellten LOB-Dateien `/u/foo/lob/pfad/bar.001`, `/u/foo/lob/pfad/bar.002` usw.

#### piDataDescriptor

Eingabe. Zeiger auf eine *sqldcol*-Struktur, die die Spaltennamen für die Ausgabedatei angibt. Der Wert des Feldes *dcolmeth* bestimmt, wie die restlichen in diesem Parameter bereitgestellten Informationen vom Dienstprogramm EXPORT ausgewertet werden. Gültige Werte für diesen Parameter (definiert in *sqlutil*) sind:

#### SQL\_METH\_N

Namen. Gibt Spaltennamen an, die in der Ausgabedatei verwendet werden sollen.

**SQL\_METH\_D**

Standardeinstellung. Vorhandene Spaltennamen aus der Tabelle sollen in der Ausgabedatei verwendet werden. In diesem Fall werden die Anzahl der Spalten und die Spaltenspezifikationsmatrix ignoriert. Die Spaltennamen werden aus der Ausgabe der SELECT-Anweisung abgeleitet, die in *pActionString* angegeben ist.

**piActionString**

Eingabe. Zeiger auf eine *sqllob*-Struktur, die eine gültige dynamische SQL-SELECT-Anweisung enthält. Die Struktur enthält ein 4-Byte-Langfeld, gefolgt von den Zeichen, die die SELECT-Anweisung bilden. Die SELECT-Anweisung gibt die Daten an, die aus der Datenbank extrahiert und in die externe Datei geschrieben werden sollen.

Die Spalten für die externe Datei (aus *piDataDescriptor*) und die Datenbankspalten aus der SELECT-Anweisung werden gemäß ihrer jeweiligen Listen-/Strukturpositionen abgeglichen. Die erste aus der Datenbank gewählte Spalte mit Daten wird in die erste Spalte der externen Datei gestellt, und ihr Spaltenname wird dem ersten Element der externen Spaltenmatrix entnommen.

**piFileType**

Eingabe. Eine Zeichenfolge, die das Format der Daten in der externen Datei angibt. Unterstützte externe Dateiformate (definiert in *sqlutil*) sind:

**SQL\_DEL**

ASCII-Format ohne universelle Zeilenbegrenzer für den Datenaustausch mit dBase, BASIC und den IBM Personal Decision Series-Programmen und vielen anderen Datenbank- und Dateimanagern.

**SQL\_WSF**

Arbeitsblattformate zum Datenaustausch mit den Programmen Lotus Symphony und 1-2-3.

**SQL\_IXF**

PC-Version des Integrated Exchange Format. Dies ist das bevorzugte Format für den Export von Daten aus einer Tabelle. In diesem Format exportierte Daten können später in dieselbe Tabelle oder in eine andere Datenbankmanagertabelle importiert oder geladen werden.

**piFileTypeMod**

Eingabe. Ein Zeiger auf eine *sqldcol*-Struktur, die ein 2-Byte-Langfeld enthält, gefolgt von einer Matrix mit Zeichen, die eine oder mehrere Verarbeitungsoptionen angeben. Wenn dieser Zeiger NULL ist oder die Struktur, auf die gezeigt wird, null Zeichen enthält, wird diese Aktion als Auswahl einer Standardspezifikation gewertet.

Nicht alle Optionen können mit allen unterstützten Dateitypen verwendet werden. Weitere Informationen hierzu finden Sie in *File type modifiers for export*.

**piMsgFileName**

Eingabe. Eine Zeichenfolge, die den Bestimmungsort für vom Dienstprogramm zurückgegebene Fehlernachrichten, informative Nachrichten und Warnungen enthält. Dies kann der Pfad und der Name einer Betriebssystemdatei oder einer Standardeinheit sein. Wenn die Datei bereits vorhanden ist, wird sie überschrieben. Wenn sie nicht vorhanden ist, wird eine neue Datei erstellt.

### **iCallerAction**

Eingabe. Eine vom aufrufenden Programm angeforderte Aktion. Gültige Werte (definiert in `sqlutil`) sind:

#### **SQLU\_INITIAL**

Erstaufruf. Dieser Wert muss beim ersten Aufruf der API verwendet werden.

Wenn der Erstaufruf oder ein nachfolgender Aufruf zurückgegeben wird und erfordert, dass die aufrufende Anwendung eine Aktion ausführt, bevor die angeforderte Exportoperation ausgeführt wird, muss die Aktion des aufrufenden Programms auf eine der folgenden Optionen gesetzt werden:

#### **SQLU\_CONTINUE**

Verarbeitung fortsetzen. Dieser Wert kann nur bei nachfolgenden Aufrufen der API verwendet werden, nachdem der Erstaufruf zurückgegeben wurde und das Dienstprogramm eine Benutzereingabe anfordert (z. B. um auf eine Bandendebedingung zu reagieren). Er gibt an, dass die vom Dienstprogramm angeforderte Benutzeraktion beendet wurde und das Dienstprogramm die Verarbeitung der Erstanforderung fortsetzen kann.

#### **SQLU\_TERMINATE**

Verarbeitung beenden. Dieser Wert kann nur bei nachfolgenden Aufrufen der API verwendet werden, nachdem der Erstaufruf zurückgegeben wurde und das Dienstprogramm eine Benutzereingabe anfordert (z. B. um auf eine Bandendebedingung zu reagieren). Er gibt an, dass die vom Dienstprogramm angeforderte Benutzeraktion nicht ausgeführt wurde und das Dienstprogramm die Verarbeitung der Erstanforderung beenden soll.

### **poExportInfoOut**

Ein Zeiger auf die `db2ExportOut`-Struktur.

### **oRowsExported**

Ausgabe. Gibt die Anzahl der in die Zielformat exportierten Datensätze zurück.

### **REXX-API-Syntax:**

```
EXPORT :stmt TO datafile OF filetype  
[MODIFIED BY :filemod] [USING :dcoldata]  
MESSAGES msgfile [ROWS EXPORTED :number]
```

```
CONTINUE EXPORT
```

```
STOP EXPORT
```

### **REXX-API-Parameter:**

**stmt** Eine REXX-Host-Variable, die eine gültige dynamische SQL-SELECT-Anweisung enthält. Die Anweisung gibt die Daten an, die aus der Datenbank extrahiert werden sollen.

#### **datafile**

Name der Datei, in die die Daten exportiert werden sollen.

#### **filetype**

Das Format der Daten in der Exportdatei. Folgende Dateiformate werden unterstützt:

**DEL** ASCII-Format ohne universelle Zeilenbegrenzer

WSF Arbeitsblattformat

IXF PC-Version des Integrated Exchange Format

#### **filetmod**

Eine Hostvariable, die zusätzliche Verarbeitungsoptionen enthält.

#### **dcoldata**

Eine REXX-Verbund-Host-Variable, die die Spaltennamen enthält, die in der Exportdatei verwendet werden sollen. Im folgenden steht XXX für den Namen der Host-Variablen:

XXX.0 Anzahl der Spalten (Anzahl der Elemente im restlichen Teil der Variablen).

XXX.1 Erster Spaltenname.

XXX.2 Zweiter Spaltenname.

XXX.3 und so weiter.

Wenn dieser Parameter NULL ist oder kein Wert für *dcoldata* angegeben wurde, verwendet das Dienstprogramm die Spaltennamen aus der Datenbanktabelle.

#### **msgfile**

Datei-, Pfad- oder Einheitenname für Fehlermeldungen und Warnungen.

#### **number**

Eine Host-Variable, die die Anzahl der exportierten Zeilen enthält.

#### **Anmerkungen zur Verwendung:**

Achten Sie darauf, alle Tabellenoperationen abzuschließen und alle Sperren freizugeben, bevor Sie eine Exportoperation starten. Dies erreichen Sie, indem Sie eine COMMIT-Operation absetzen, nachdem Sie alle geöffneten Cursor mit WITH HOLD geschlossen haben, oder indem Sie eine ROLLBACK-Operation absetzen.

Die Verwendung von Tabellenaliasnamen in der SELECT-Anweisung ist zulässig.

Die in die Nachrichtendatei gestellten Nachrichten enthalten die Informationen, die vom Service für die Nachrichtenabfrage zurückgegeben werden. Jede Nachricht beginnt in einer neuen Zeile.

Das Dienstprogramm EXPORT erzeugt immer dann eine Warnung, wenn eine Spalte des Typs CHARACTER mit einer größeren Länge als 254 für das Exportieren in Dateien mit dem DEL-Format ausgewählt wird.

Eine Warnung wird ausgegeben, falls die Anzahl der Spalten (*dcolnum*) in der externen Spaltennamenmatrix *piDataDescriptor* nicht mit der Anzahl der Spalten identisch ist, die durch die SELECT-Anweisung generiert werden. In diesem Fall wird für die Anzahl der in die externe Datei geschriebenen Spalten der kleinere der beiden Werte verwendet. Überzählige Datenbankspalten oder externe Spaltennamen werden beim Generieren der Ausgabedatei nicht verwendet.

Falls das Modul db2uexpm.bnd oder andere bereitgestellte Dateien .bnd manuell gebunden sind, darf die Option **format** für den Binder nicht verwendet werden.

Zum Versetzen von Daten zwischen Datenbanken sollte der PC/IXF-Import verwendet werden. Falls Zeichendaten, die Zeilentrennzeichen enthalten, in eine ASCII-Datei ohne universelle Zeilenbegrenzer (DEL) exportiert und durch ein

## Export-API "db2Export"

Textübertragungsprogramm verarbeitet werden, werden Felder, die die Zeilentrennzeichen enthalten, verkleinert oder erweitert.

Mit DB2 Connect können Tabellen von DRDA-Servern wie DB2 für z/OS und OS/390, DB2 für VM und VSE sowie DB2 für iSeries exportiert werden. Nur der PC/IXF-Export wird unterstützt.

Wenn das Dienstprogramm EXPORT über ein AIX-System aufgerufen wird, werden keine mehrteiligen PC/IXF-Dateien erstellt.

Indexdefinitionen für eine Tabelle werden in die PC/IXF-Datei aufgenommen, wenn der Inhalt einer einzelnen Datenbanktabelle in eine PC/IXF-Datei exportiert wird und hierbei *piActionString* mit der Angabe `SELECT * FROM tabellenname` beginnt und der Parameter *piDataDescriptor* Standardnamen angibt. Indizes werden für Sichten nicht gesichert. Gleiches gilt, falls die Klausel `SELECT` von *piActionString* eine Verknüpfung enthält. Eine Klausel `WHERE`, eine Klausel `GROUP BY` oder eine Klausel `HAVING` in *piActionString* verhindert das Sichern der Indizes nicht. In allen diesen Fällen muss bei einem Export aus typisierten Tabellen die gesamte Hierarchie exportiert werden.

Das Dienstprogramm EXPORT speichert das Attribut `NOT NULL WITH DEFAULT` der Tabelle in einer IXF-Datei, sofern die `SELECT`-Anweisung im Format `SELECT * FROM tabellenname` angegeben wird.

Beim Exportieren von typisierten Tabellen können Anweisungen für eine Unterauswahl nur dadurch ausgedrückt werden, indem der Name der Zieltabelle und die Klausel `WHERE` angegeben wird. Beim Exportieren einer Hierarchie können Gesamtauswahl und *select-anweisung* nicht angegeben werden.

Bei anderen Dateiformaten als IXF empfiehlt es sich, die Traversierfolgeliste anzugeben, weil diese Liste DB2 mitteilt, wie die Hierarchie traversiert werden muss und welche untergeordneten Tabellen exportiert werden sollen. Wenn diese Liste nicht angegeben wird, werden alle Tabellen in der Hierarchie exportiert und die Standardreihenfolge lautet `OUTER`. Alternativ kann die Standardreihenfolge verwendet werden, die durch die Funktion `OUTER` vorgegeben wird.

**Anmerkung:** Während einer Importoperation sollte die gleiche Traversierfolge verwendet werden. Das Dienstprogramm `LOAD` unterstützt das Laden von Hierarchien oder Unterhierarchien nicht.

### Hinweise zu DB2 Data Links Manager:

So können Sie sicherstellen, dass eine konsistente Kopie der Tabelle und der entsprechenden Dateien, auf die in den `DATALINK`-Spalten verwiesen wird, für den Export kopiert wird:

1. Setzen Sie den Befehl `QUIESCE TABLESPACES FOR TABLE tabellenname SHARE` ab.  
Hierdurch wird sichergestellt, dass bei der Ausführung von `EXPORT` keine Aktualisierungsoperationen verarbeitet werden.
2. Setzen Sie den Befehl `EXPORT` ab.
3. Führen Sie das Dienstprogramm `dlfm_export` auf jedem Data Links-Server aus. Die Eingabe für das Dienstprogramm `dlfm_export` ist der Name der Steuerdatei, der durch das Dienstprogramm `EXPORT` generiert wird. Hierdurch wird ein `TAR`-Archiv (oder ein äquivalentes Archiv) der Dateien erzeugt, die in der

Steuerdatei aufgelistet sind. Das Dienstprogramm `dlfm_export` erfasst die ACL-Informationen der archivierten Dateien nicht.

4. Setzen Sie den Befehl **QUIESCE TABLESPACES FOR TABLE `tabellenname` RESET** ab.

Hierdurch wird die Tabelle für Aktualisierungen verfügbar gemacht.

EXPORT wird als SQL-Anwendung ausgeführt. Die Zeilen und Spalten, die die Bedingungen der SELECT-Anweisung erfüllen, werden aus der Datenbank extrahiert. Für DATALINK-Spalten darf in der SELECT-Anweisung keine Skalarfunktion angegeben werden.

Bei einer erfolgreichen Ausführung von EXPORT werden die folgenden Dateien generiert:

- Die Exportdatei, die im Befehl EXPORT angegeben wurde. Ein DATALINK-Spaltenwert in dieser Datei hat das gleiche Format wie das von den Dienstprogrammen LOAD und IMPORT verwendete. Wenn der DATALINK-Spaltenwert der SQL-NULL-Wert ist, ist die Verarbeitung mit den anderen Datentypen identisch.
- Steuerdateien `servername`, die für jeden Data Links-Server generiert werden. Beim Betriebssystem Windows NT wird eine einzige Steuerdatei `ctrlfile.lst` durch alle Data Links-Server verwendet. Diese Steuerdateien werden in das Verzeichnis `<datendateipfad>/dlfm/JJJJMMTT/HHMMSS` gestellt (beim Betriebssystem Windows wird die Datei `ctrlfile.lst` in das Verzeichnis `<datendateipfad>\dlfm\JJJJMMTT\HHMMSS` gestellt). Hierbei steht JJJJMMTT für das Datum (Jahr Monat Tag), und HHMMSS gibt die Uhrzeit an (Stunde Minute Sekunde).

Mit dem Dienstprogramm `dlfm_export` können Dateien von einem Data Links-Server exportiert werden. Dieses Dienstprogramm generiert eine Archivdatei, die zur Wiederherstellung von Dateien auf dem Data Links-Zielservers verwendet werden kann.

#### Zugehörige Konzepte:

- „Versetzen von DB2 Data Links Manager-Daten mit EXPORT - Konzepte“ auf Seite 227

#### Zugehörige Referenzen:

- „SQLCA“ in *Administrative API Reference*
- „SQLCHAR“ in *Administrative API Reference*
- „SQLDCOL“ in *Administrative API Reference*
- „SQLU-MEDIA-LIST“ in *Administrative API Reference*
- „Änderungswerte für Dateityp für EXPORT“ auf Seite 22
- „Einschränkungen für Begrenzer beim Versetzen von Daten“ auf Seite 247

#### Zugehörige Beispiele:

- „`expsamp.sqb` -- Export and import tables with table data to a DRDA database (IBM COBOL)“
- „`impexp.sqb` -- Export and import tables with table data (IBM COBOL)“
- „`tload.sqb` -- How to export and load table data (IBM COBOL)“
- „`tbmove.sqc` -- How to move table data (C)“
- „`tbmove.sqC` -- How to move table data (C++)“

## Änderungswerte für Dateityp für EXPORT

Tabelle 1. Gültige Änderungswerte für Dateityp für EXPORT: Alle Dateiformate

Änderungswert	Beschreibung
lobsinfile	<p><i>lob-pfad</i> gibt den Pfad zu den Dateien an, die LOB-Daten enthalten.</p> <p>Jeder Pfad enthält mindestens eine Datei, die wiederum mindestens ein großes Objekt (LOB) enthält, auf das durch eine LOB-Positionskenung (LOB Location Specifier - LLS) in der Datendatei verwiesen wird. Die LOB-Positionskenung ist eine Zeichenfolgendarstellung der Position eines großen Objekts (LOB) in einer Datei, die im LOB-Dateipfad gespeichert ist. Das Format einer LOB-Positionskenung lautet <i>dateiname.erv.nnn.mmm/</i>. Hierbei steht <i>dateiname.erv</i> für den Namen der Datei, die das große Objekt (LOB) enthält, <i>nnn</i> für die (in Byte angegebene) relative Position des großen Objekts in der Datei und <i>mmm</i> für die Länge des großen Objekts in Byte. Falls beispielsweise die Zeichenfolge <i>db2exp.001:123:456/</i> in der Datendatei gespeichert ist, befindet sich das große Objekt (LOB) an der relativen Position 123 in der Datei 'db2exp.001' und ist 456 Byte lang.</p> <p>Wenn Sie den Änderungswert „lobsinfile“ mit dem Befehl EXPORT angeben, werden die LOB-Daten an der Speicherposition gespeichert, die durch die Klausel LOBS TO angegeben wird. Andernfalls werden die LOB-Daten an das momentan arbeitende Verzeichnis gesendet. Die Klausel LOBS TO gibt einen oder mehrere Pfade zu Verzeichnissen an, in denen die LOB-Dateien gespeichert werden sollen. Jeder LOB-Pfad enthält mindestens eine Datei, und jede Datei enthält mindestens ein großes Objekt (LOB).</p> <p>Um ein Null-LOB anzugeben, geben Sie als Größe den Wert -1 ein. Wenn die Größe mit dem Wert 0 angegeben wird, gilt die Angabe als ein großes Objekt mit der Länge 0. Bei Null-LOBs mit der Länge -1 werden die relative Adresse und der Dateiname ignoriert. Die LOB-Positionskenung eines Null-LOBs könnte beispielsweise <i>db2exp.001:7:-1/</i> lauten.</p>

Tabelle 2. Gültige Änderungswerte für Dateityp für EXPORT: DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer)

Änderungswert	Beschreibung
chardelx	<p><i>x</i> ist ein einzelner Zeichenfolgebegrenzer. Der Standardwert ist ein doppeltes Anführungszeichen ("). Das angegebene Zeichen wird statt doppelter Anführungszeichen verwendet, um eine Zeichenfolge einzuschließen.<sup>2</sup> Falls Sie die doppelten Anführungszeichen explizit als Zeichenfolgebegrenzer angeben wollen, müssen Sie dies folgendermaßen tun:</p> <p><code>modified by charde1""</code></p> <p>Auch das einfache Anführungszeichen (') kann wie folgt als Zeichenfolgebegrenzer festgelegt werden:</p> <p><code>modified by charde1''</code></p>
codepage=x	<p><i>x</i> ist eine ASCII-Zeichenfolge. Der Wert wird als Codepage der Daten im Ausgabedatensatz gewertet. Während der Exportoperation werden Zeichendaten aus der Codepage der Anwendung in diese Codepage konvertiert.</p> <p>Bei reinem DBCS (Grafik), gemischtem DBCS und EUC sind Begrenzer auf den Bereich von x00 bis x3F (jeweils einschließlich) beschränkt.</p> <p><b>Anmerkung:</b> Der Änderungswert <code>codepage</code> kann nicht zusammen mit dem Änderungswert <code>lobsinfile</code> verwendet werden.</p>



## Export-API "db2Export"

Tabelle 2. Gültige Änderungswerte für Dateityp für EXPORT: DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer) (Forts.)

Änderungswert	Beschreibung
timestampformat="x"	<p>x ist das Format der Zeitmarke in der Quellendatei.<sup>4</sup> Gültige Zeitmarkenelemente sind:</p> <p>YYYY - Jahr (vier Ziffern von 0000 bis 9999)</p> <p>M - Monat (eine oder zwei Ziffern von 1 bis 12)</p> <p>MM - Monat (zwei Ziffern von 01 bis 12; dieses Element und das Element M schließen sich gegenseitig aus)</p> <p>MMM - Monat (Abkürzung des Monatsnamen, die aus drei Buchstaben besteht und bei der die Großschreibung zu beachten ist; dieses Element und die Elemente M und MM schließen sich gegenseitig aus)</p> <p>D - Tag (eine oder zwei Ziffern von 1 bis 31)</p> <p>DD - Tag (zwei Ziffern von 1 bis 31; dieses Element und Element D schließen sich gegenseitig aus)</p> <p>DDD - Tag des Jahres (drei Ziffern von 001 bis 366; dieses Element und andere Elemente für Tag oder Monat schließen sich gegenseitig aus)</p> <p>H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System)</p> <p>HH - Stunde (zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus)</p> <p>M - Minute (ein oder zwei Ziffern von 0 bis 59)</p> <p>MM - Minute (zwei Ziffern von 0 bis 59; dieses Element und das Minutenelement M schließen sich gegenseitig aus)</p> <p>S - Sekunde (ein oder zwei Ziffern von 0 bis 59)</p> <p>SS - Sekunde (zwei Ziffern von 0 bis 59; dieses Element und das Element S schließen sich gegenseitig aus)</p> <p>SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86399; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus)</p> <p>UUUUUU - Mikrosekunde (6 Ziffern von 000000 bis 999999; dieses Element und alle anderen Mikrosekundenelemente schließen sich gegenseitig aus)</p> <p>UUUUU - Mikrosekunde (5 Ziffern von 00000 bis 99999; dieses Element wird dem Bereich 000000 bis 999990 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>UUUU - Mikrosekunde (4 Ziffern von 0000 bis 9999; dieses Element wird dem Bereich 000000 bis 999900 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>UUU - Mikrosekunde (3 Ziffern von 000 bis 999; dieses Element wird dem Bereich 000000 bis 999000 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>UU - Mikrosekunde (2 Ziffern von 00 bis 99; dieses Element wird dem Bereich 000000 bis 990000 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>U - Mikrosekunde (1 Ziffer von 0 bis 9; dieses Element wird dem Bereich 000000 bis 900000 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>TT - Meridiananzeiger (AM oder PM)</p>

Tabelle 2. Gültige Änderungswerte für Dateityp für EXPORT: DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer) (Forts.)

Änderungswert	Beschreibung
timestampformat="x"	<p>Beispiel für ein Zeitmarkenformat: "YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <p>Das Element MMM bringt die folgenden Werte hervor: 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', und 'Dec'. 'Jan' entspricht dem Monat 1 und 'Dec' entspricht dem Monat 12.</p> <p>Das folgende Beispiel veranschaulicht das Exportieren von Daten, die benutzerdefinierte Zeitmarkenformate aus der Tabelle 'schedule' enthalten:</p> <pre>db2 export to delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" select * from schedule</pre>

Tabelle 3. Gültige Änderungswerte für Dateityp für EXPORT: WSF-Dateiformat

Änderungswert	Beschreibung
1	Erstellt eine WSF-Datei, die mit Lotus 1-2-3, Release 1, oder Lotus 1-2-3, Release 1a, kompatibel ist. <sup>5</sup> Dies ist der Standardwert.
2	Erstellt eine WSF-Datei, die mit Lotus Symphony, Release 1.0, kompatibel ist. <sup>5</sup>
3	Erstellt eine WSF-Datei, die mit Lotus 1-2-3, Version 2, oder Lotus Symphony, Release 1.1, kompatibel ist. <sup>5</sup>
4	Erstellt eine WSF-Datei, die DBCS-Zeichen enthält.

#### Anmerkungen:

- Das Dienstprogramm EXPORT setzt keine Warnung ab, wenn versucht wird, nicht unterstützte Dateitypen mit der Option MODIFIED BY zu verwenden. Wenn dies versucht wird, schlägt die Exportoperation fehl, und es wird ein Fehlercode zurückgegeben.
- Unter Delimiter restrictions for moving data sind Einschränkungen für die Zeichen aufgeführt, die als Überschreibungswerte für Begrenzer verwendet werden können.
- Normalerweise schreibt das Dienstprogramm EXPORT
  - Datumsdaten im Format *JJJJMMTT*
  - Daten des Typs char(date) im Format *JJJJ-MM-TT*
  - Zeitdaten im Format *HH.MM.SS*
  - Zeitmarkendaten im Format *JJJJ-MM-TT-HH.MM.SS.uuuuuu*

Daten in allen Spalten des Typs DATETIME, die in der SELECT-Anweisung für die Exportoperation angegeben sind, stehen ebenfalls in diesen Formaten.

- Bei Zeitmarkenformaten muss darauf geachtet werden, eine Mehrdeutigkeit zwischen den Deskriptoren für Monat und Minute zu vermeiden, da beide den Buchstaben M verwenden. Ein Monatsfeld muss neben anderen Datumsfeldern stehen. Ein Minutenfeld muss neben anderen Zeitfeldern stehen. Die folgenden Zeitmarkenformate sind beispielsweise mehrdeutig:

```
"M" (Hierdurch könnte sowohl eine Monats- als auch eine
Minutenangabe dargestellt sein.)
"M:M" (In diesem Fall ist nicht eindeutig, welches Element für den
Monat und welches für die Minutenangabe steht.)
"M:YYYY:M" (Hier werden beide Elemente M als Monat interpretiert.)
"S:M:YYYY" (In diesem Fall steht das Element M sowohl neben einem
Zeit- als auch neben einem Datumswert.)
```

Im Fall einer Mehrdeutigkeit meldet das Dienstprogramm eine Fehlermeldung, und die Operation schlägt fehl.

## Export-API "db2Export"

Die folgenden Zeitmarkenformate sind beispielsweise eindeutig:

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

5. Diese Dateien können auch für ein bestimmtes Produkt erstellt werden, indem L für Lotus 1-2-3 oder S für Symphony in der Parameterzeichenfolge *änderungswert-für-dateityp* angegeben wird. Es kann nur ein Wert bzw. Produktbezeichner angegeben werden.

### Zugehörige Referenzen:

- „Export-API "db2Export"“ auf Seite 14
- „Befehl EXPORT“ auf Seite 9
- „Einschränkungen für Begrenzer beim Versetzen von Daten“ auf Seite 247

---

## CLP-Beispiele für EXPORT-Sitzungen

Das folgende Beispiel zeigt, wie Daten aus der Tabelle STAFF der Beispieldatenbank SAMPLE (mit der der Benutzer verbunden sein muss) im IXF-Format in die Datei myfile.ixf exportiert werden. Wenn die Datenbankverbindung nicht über DB2 Connect hergestellt wurde, werden die Indexdefinitionen (falls vorhanden) in der Ausgabedatei gespeichert. Andernfalls werden nur die Daten gespeichert.

```
db2 export to myfile.ixf of ixf messages msgs.txt select * from staff
```

Das folgende Beispiel zeigt, wie Daten über Mitarbeiter in Abteilung 20 aus der Tabelle STAFF der Beispieldatenbank SAMPLE (mit der der Benutzer verbunden sein muss) im IXF-Format in die Datei awards.ixf exportiert werden.

```
db2 export to awards.ixf of ixf messages msgs.txt select * from staff
where dept = 20
```

Das folgende Beispiel zeigt, wie große Objekte in eine DEL-Datei exportiert werden:

```
db2 export to myfile.del of del lobs to mylobs/
lobfile lobs1, lobs2 modified by lobsinfile
select * from emp_photo
```

Das folgende Beispiel zeigt, wie große Objekte (LOBs) in eine DEL-Datei exportiert werden, wobei ein zweites Verzeichnis für Dateien angegeben wird, die im ersten Verzeichnis möglicherweise keinen Platz haben:

```
db2 export to myfile.del of del
lobs to /db2exp1/, /db2exp2/ modified by lobsinfile
select * from emp_photo
```

Das folgende Beispiel zeigt den Export von Daten in eine DEL-Datei, wobei ein einfaches Anführungszeichen als Zeichenfolgebegrenzer, ein Semikolon als Spaltenbegrenzer und ein Komma als Dezimalzeichen verwendet wird. Die gleiche Konvention muss beim Rückimportieren der Daten in die Datenbank verwendet werden:

```
db2 export to myfile.del of del
modified by chardel'' coldel; decpt,
select * from staff
```

### Zugehörige Referenzen:

- „Befehl EXPORT“ auf Seite 9

---

## Kapitel 2. IMPORT

In diesem Kapitel wird das DB2 UDB-Dienstprogramm IMPORT beschrieben, das die SQL-Anweisung INSERT verwendet, um Daten aus einer Eingabedatei in eine Tabelle oder Sicht zu schreiben. Wenn die Zieltabelle oder -sicht bereits Daten enthält, können Sie die importierten Daten entweder an die vorhandenen Daten anfügen oder die vorhandenen Daten ersetzen.

Die folgenden Themen werden behandelt:

- „IMPORT - Übersicht“
- „Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von IMPORT“ auf Seite 29
- „Verwenden von IMPORT“ auf Seite 29
- „Verwenden von IMPORT in einer Client/Server-Umgebung“ auf Seite 30
- „Verwenden von IMPORT mit gepufferten INSERT-Operationen“ auf Seite 31
- „Verwenden von IMPORT mit Identitätsspalten“ auf Seite 32
- „Verwenden von IMPORT mit generierten Spalten“ auf Seite 34
- „Erneutes Erstellen einer exportierten Tabelle mit IMPORT“ auf Seite 35
- „Importieren von großen Objekten (LOBS)“ auf Seite 36
- „Importieren von benutzerdefinierten einzigartigen Datentypen (UDTs)“ auf Seite 37
- „Sperren von Tabellen während des Imports“ auf Seite 37
- „Befehl IMPORT“ auf Seite 39
- „Import-API "db2Import"" auf Seite 53
- „Hinweise zu Zeichensatz und NLS“ auf Seite 76
- „CLP-Beispiele für IMPORT-Sitzungen“ auf Seite 76

Informationen zum Importieren von DB2 Data Links Manager-Daten finden Sie unter „Versetzen von DB2 Data Links Manager-Daten mit IMPORT“ auf Seite 231. Angaben zum Importieren von Daten aus typisierten Tabellen enthält der Abschnitt „Versetzen von Daten zwischen typisierten Tabellen“ auf Seite 248. Informationen zum Importieren von Daten aus einer Datei auf der DB2 Connect-Workstation in eine DRDA-Server-Datenbank und umgekehrt können Sie unter „Versetzen von Daten mit DB2 Connect“ auf Seite 235 nachlesen.

---

### IMPORT - Übersicht

Das Dienstprogramm IMPORT fügt Daten aus einer Eingabedatei in eine Tabelle oder aktualisierbare Sicht ein. Wenn die Tabelle oder Sicht, die die importierten Daten aufnehmen soll, bereits Daten enthält, können Sie die importierten Daten entweder an die vorhandenen Daten anfügen oder die vorhandenen Daten ersetzen.

Folgende Informationen werden benötigt, wenn Daten importiert werden:

- Der Pfad und der Name der Eingabedatei.
- Der Name oder Aliasname der Zieltabelle oder -sicht.
- Das Format der Daten in der Eingabedatei. Mögliche Formate sind: IXF, WSF, DEL oder ASC.

- Die Angabe, ob die Eingabedaten in die Tabelle oder Sicht eingefügt werden oder ob die vorhandenen Daten in der Tabelle oder Sicht durch die Eingabedaten aktualisiert oder ersetzt werden sollen.
- Ein Nachrichtendateiname, wenn das Dienstprogramm über die Anwendungsprogrammierschnittstelle (API) **sqlimpr** aufgerufen wird.
- Beim Arbeiten mit typisierten Tabellen müssen Sie möglicherweise die Methode oder Reihenfolge für den Durchgang durch alle strukturierte Typen angeben. Die Reihenfolge des Durchgangs durch alle übergeordneten und untergeordneten Tabellen in der Hierarchie von oben nach unten und von links nach rechts wird als *Traversierfolge* bezeichnet. Diese Reihenfolge ist wichtig beim Versetzen von Daten zwischen Tabellenhierarchien, weil dabei im Verhältnis zu anderen Daten ermittelt wird, wohin die Daten versetzt werden.

Beim Arbeiten mit typisierten Tabellen müssen Sie möglicherweise auch die Liste der untergeordneten Tabellen angeben. Diese Liste gibt an, in welche untergeordneten Tabellen und Attribute Daten importiert werden sollen.

Sie können auch Folgendes angeben:

- Die Methode, die zum Importieren der Daten verwendet werden soll: Spaltenposition, Spaltenname oder relative Spaltenposition.
- Die Anzahl der mit INSERT einzufügenden Zeilen, bevor die Tabellenänderungen festgeschrieben werden. Wenn Sie in regelmäßigen Abständen COMMIT-Operationen ausführen, reduziert sich die Anzahl der Zeilen, die verloren gehen, wenn während der Importoperation ein Fehler auftritt und eine ROLLBACK-Operation ausgeführt wird. Dadurch wird auch verhindert, dass DB2<sup>®</sup>-Protokolle bei der Verarbeitung einer großen Eingabedatei voll werden.
- Die Anzahl der Datensätze in der Datei, die vor Beginn der Importoperation übersprungen werden sollen. Im Falle eines Fehlers können Sie die Importoperation unmittelbar nach der letzten Zeile, die erfolgreich importiert und festgeschrieben wurde, erneut starten.
- Die Namen der Spalten innerhalb der Tabelle oder Sicht, in die die Daten eingefügt werden sollen.
- Ein Nachrichtendateiname. Bei DB2-Operationen wie dem Exportieren, Importieren, Laden, Binden oder Wiederherstellen von Daten können Sie angeben, dass Nachrichtendateien erstellt werden, die Fehlermeldungen, Warnungen und informative Nachrichten zu diesen Operationen enthalten. Geben Sie den Namen dieser Dateien mit dem Parameter MESSAGES an. Diese Nachrichtendateien sind gewöhnliche ASCII-Dateien. Jede Nachricht in einer Nachrichtendatei beginnt in einer neuen Zeile und enthält Informationen, die von der DB2-Funktion zum Abrufen von Nachrichten bereitgestellt werden. Sie können sie mit dem bei Ihrem Betriebssystem üblichen Verfahren drucken oder mit einem beliebigen ASCII-Editor anzeigen.

**Anmerkung:** Die Angabe von Zieltabellenspaltennamen oder einer bestimmten Importmethode verlangsamt den Import in eine ferne Datenbank.

#### **Zugehörige Konzepte:**

- „Versetzen von Daten zwischen typisierten Tabellen“ auf Seite 248

#### **Zugehörige Referenzen:**

- „Import-API "db2Import"“ auf Seite 53
- „CLP-Beispiele für IMPORT-Sitzungen“ auf Seite 76
- „Dateiformate der Dienstprogramme EXPORT, IMPORT und LOAD“ auf Seite 275
- „Befehl IMPORT“ auf Seite 39

---

## Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von IMPORT

Zugriffsrechte ermöglichen es Benutzern, Datenbankressourcen zu erstellen oder auf sie zuzugreifen. Berechtigungsstufen stellen eine Methode dar, um Berechtigungen sowie Operationen von Dienstprogrammen und zur Datenbankmanagerpflege auf höherer Ebene zu gruppieren. Sie dienen zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte. Benutzer können nur auf solche Objekte zugreifen, für die sie die entsprechende Berechtigung besitzen, d. h., für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigung verfügen.

Zum Erstellen einer neuen Tabelle mit dem Dienstprogramm IMPORT benötigen Sie die Berechtigung SYSADM, DBADM oder das Zugriffsrecht CREATETAB für die Datenbank. Zum Ersetzen von Daten in einer vorhandenen Tabelle oder Sicht benötigen Sie die Berechtigung SYSADM, die Berechtigung DBADM oder das Zugriffsrecht CONTROL für die Tabelle oder Sicht bzw. die Zugriffsrechte INSERT, SELECT, UPDATE und DELETE für die Tabelle oder Sicht. Zum Anhängen von Daten an eine vorhandene Tabelle oder Sicht benötigen Sie die Zugriffsrechte SELECT und INSERT für die Tabelle oder Sicht.

### Zugehörige Referenzen:

- „Import-API "db2Import"“ auf Seite 53
- „Befehl IMPORT“ auf Seite 39

---

## Verwenden von IMPORT

### Vorbedingungen:

Bevor Sie das Dienstprogramm IMPORT ausführen, müssen Sie mit der Datenbank, in die die Daten importiert werden, verbunden bzw. in der Lage sein, implizit eine Verbindung zur Datenbank herzustellen. Da das Dienstprogramm die Anweisung COMMIT oder ROLLBACK absetzt, sollten Sie vor dem Aufrufen von IMPORT alle Transaktionen beenden und alle Sperren aufheben, indem Sie die Anweisung COMMIT oder ROLLBACK ausführen.

### Einschränkungen:

Für das Dienstprogramm IMPORT gelten die folgenden Einschränkungen:

- Dieses Dienstprogramm bietet keine Unterstützung für Kurznamen.
- Ist die vorhandene Tabelle eine übergeordnete Tabelle mit einem Primärschlüssel, auf den ein Fremdschlüssel einer abhängigen Tabelle verweist, können keine Daten ersetzt werden. Es ist nur möglich, die neuen Daten anzufügen.
- Es ist nicht möglich, eine IMPORT REPLACE-Operation für eine zu Grunde liegende Tabelle einer gespeicherten Abfragetabelle auszuführen, die im Modus "Sofort aktualisieren" (REFRESH IMMEDIATE) definiert wurde.
- Sie können keine Daten in eine Systemtabelle, eine Übersichtstabelle oder in eine Tabelle mit Spalten strukturierten Typs importieren.
- Sie können keine Daten in deklarierte temporäre Tabellen importieren.
- Mit dem Dienstprogramm IMPORT können keine Sichten erstellt werden.

- Referenzielle Integritätsbedingungen und Fremdschlüsseldefinitionen werden beim Erstellen von Tabellen aus PC/IXF-Dateien nicht beibehalten. (Primärschlüsseldefinitionen *werden* beibehalten, wenn die Daten zuvor mit SELECT \* exportiert wurden.)
- Da vom Dienstprogramm IMPORT eigene SQL-Anweisungen generiert werden, kann die maximale Anweisungsgröße von 64 KB in einigen Fällen überschritten werden.

Für das Dienstprogramm IMPORT gilt die folgende Einschränkung:

Überschreitet der Umfang der Ausgabenachrichten, die durch eine Importoperation für eine ferne Datenbank generiert wurden, eine Größe von 60 KB, behält das Dienstprogramm die ersten 30 KB und die letzten 30 KB bei.

#### Prozedur:

Das Dienstprogramm IMPORT kann über den Befehlszeilenprozessor (CLP), das Notizbuch "Importieren" in der Steuerzentrale oder eine Anwendungsprogrammierschnittstelle (API) namens **sqluimpr** aufgerufen werden.

Es folgt ein Beispiel für den Befehl IMPORT, der über den CLP abgesetzt wird:

```
db2 import from stafftab.ixf of ixf insert into userid.staff
```

So öffnen Sie das Notizbuch **Importieren**:

1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis Sie den Ordner **Tabellen** sehen.
2. Klicken Sie den Ordner **Tabellen** an. Alle vorhandenen Tabellen werden auf der rechten Seite des Fensters, im Inhaltsteilfenster, angezeigt.
3. Klicken Sie im Inhaltsteilfenster die gewünschte Tabelle mit Maustaste 2 an, und wählen Sie **Importieren** im Kontextmenü aus. Das Notizbuch **Importieren** wird geöffnet.

Ausführliche Informationen zur Steuerzentrale finden Sie in der Onlinehilfefunktion der Steuerzentrale.

#### Zugehörige Referenzen:

- „Import-API "db2Import"“ auf Seite 53

#### Zugehörige Beispiele:

- „tbmove.out -- HOW TO MOVE TABLE DATA (C)“
- „tbmove.sqc -- How to move table data (C)“
- „tbmove.out -- HOW TO MOVE TABLE DATA (C++)“
- „tbmove.sqC -- How to move table data (C++)“

---

## Verwenden von IMPORT in einer Client/Server-Umgebung

Wenn Sie eine Datei in eine ferne Datenbank importieren, kann eine gespeicherte Prozedur aufgerufen werden, um den Import auf dem Server auszuführen. In folgenden Situationen wird jedoch keine gespeicherte Prozedur aufgerufen:

- Die Anwendung und die Datenbank verwenden unterschiedliche Codepages.
- Die Datei, die importiert wird, ist eine aus mehreren Teilen bestehende PC/IXF-Datei.

- Die Methode, die zum Importieren der Daten verwendet werden soll, beruht entweder auf dem Spaltennamen oder der relativen Spaltenposition.
- Die bereitgestellte Zielspaltenliste ist länger als 4 KB.
- Die Klausel LOBS FROM oder der Änderungswert lobsinfile ist angegeben.
- Die Klausel NULL INDICATORS ist für ASC-Dateien angegeben.

Beim Importieren mit Hilfe einer gespeicherten Prozedur werden in der Nachrichtendatei Nachrichten in der auf dem Server installierten Standardsprache erstellt. Die Nachrichten liegen in der Sprache der Anwendung vor, wenn Client und Server dieselbe Sprache benutzen.

Das Dienstprogramm IMPORT erstellt zwei temporäre Dateien im Unterverzeichnis tmp des Verzeichnisses sqllib (oder des Verzeichnisses, das durch die Registrierdatenbank-Variable DB2INSTPROF angegeben wird). Eine Datei ist für die Daten vorgesehen, die andere Datei für die Nachrichten, die das Dienstprogramm IMPORT generiert.

Wenn beim Schreiben oder Öffnen von Daten auf dem Server ein Fehler gemeldet wird, prüfen Sie, ob folgende Bedingungen erfüllt sind:

- Das Verzeichnis existiert.
- Der Plattenspeicherplatz für die Dateien ist groß genug.
- Der Exemplareigner hat Schreibzugriff auf das Verzeichnis.

#### Zugehörige Konzepte:

- „IMPORT - Übersicht“ auf Seite 27

---

## Verwenden von IMPORT mit gepufferten INSERT-Operationen

In einer Umgebung mit partitionierten Datenbanken kann das Dienstprogramm IMPORT für die Verwendung gepufferter INSERT-Operationen aktiviert werden. Das verringert die Nachrichtenübertragung, die beim Importieren der Daten stattfindet, was zu einer besseren Leistung führt. Da jedoch keine Details zu einer fehlgeschlagenen gepufferten INSERT-Operation zurückgegeben werden, sollte diese Option nur aktiviert werden, wenn die Meldung von Fehlern nicht von Bedeutung ist.

Werden gepufferte INSERT-Operationen verwendet, setzt IMPORT den Wert für WARNINGCOUNT standardmäßig auf 1, sodass das Dienstprogramm fehlschlägt, falls eine beliebige Zeile zurückgewiesen wird. Wird ein Datensatz zurückgewiesen, setzt das Dienstprogramm die aktuelle Transaktion mittels einer ROLLBACK-Operation zurück. Anhand der Anzahl der festgeschriebenen Datensätze kann ermittelt werden, welche Datensätze erfolgreich in die Datenbank eingefügt wurden. Wurde die Option COMMITCOUNT angegeben, kann die Anzahl der festgeschriebenen Datensätze nicht Null sein.

Wird für WARNINGCOUNT im Befehl IMPORT explizit ein anderer Wert angegeben und wurden einige Zeilen zurückgewiesen, ist die zusammenfassende Zeilenausgabe des Dienstprogramms möglicherweise nicht korrekt. Dies liegt an einer Kombination aus der mit gepufferten INSERT-Operationen verwendeten asynchronen Fehlermeldung und der Tatsache, dass ein Fehler, der während des Einfügens einer Gruppe von Zeilen festgestellt wird, dazu führt, dass alle Zeilen der betreffenden Gruppe zurückgesetzt werden. Da das Dienstprogramm nicht zuverlässig zurückmeldet, welche Eingabedatensätze zurückgewiesen wurden, ist es schwierig

festzustellen, welche Datensätze festgeschrieben wurden und welche Datensätze erneut in die Datenbank eingefügt werden müssen.

Verwenden Sie das DB2®-Dienstprogramm BIND, um gepufferte INSERT-Operationen anzufordern. Das IMPORT-Paket db2uimpm.bnd muss mit der Option INSERT BUF erneut an die Datenbank gebunden werden. Beispiel:

```
db2 connect to ihre-datenbank
db2 bind db2uimpm.bnd insert buf
```

Die Funktion der gepufferten INSERT-Operationen kann nicht zusammen mit Importoperationen verwendet werden, bei denen der Parameter INSERT\_UPDATE angegeben wird. Um diese Einschränkung umzusetzen, wird eine neue Bindedatei (db2uimpm2.bnd) eingeführt. Die neue Datei darf nie mit der Option INSERT BUF gebunden werden, da die Importoperationen, bei denen der Parameter INSERT\_UPDATE angegeben ist, ansonsten fehlschlagen. Importoperationen, bei denen der Parameter INSERT, REPLACE oder REPLACE\_CREATE angegeben ist, sind vom Binden der neuen Datei nicht betroffen.

#### Zugehörige Konzepte:

- „IMPORT - Übersicht“ auf Seite 27

---

## Verwenden von IMPORT mit Identitätsspalten

Mit dem Dienstprogramm IMPORT können Sie Daten in eine Tabelle importieren, die eine Identitätsspalte enthält. Sofern keine identitätsbezogenen Änderungswerte für den Dateityp verwendet werden, liegen der Ausführung des Dienstprogramms die folgenden Regeln zugrunde:

- Ist die Identitätsspalte als GENERATED ALWAYS definiert, wird für eine Tabellenzeile immer dann ein Identitätswert generiert, wenn die entsprechende Zeile in der Eingabedatei keinen Wert für die Identitätsspalte enthält oder explizit einen Nullwert vorgibt. Ist für die Identitätsspalte ein Wert angegeben, der kein Nullwert ist, wird die Zeile zurückgewiesen (SQL3550W).
- Ist die Identitätsspalte als GENERATED BY DEFAULT definiert, verwendet das Dienstprogramm IMPORT benutzerdefinierte Werte, sofern diese vorhanden sind. Wenn diese Daten fehlen oder explizit Nullwerte sind, wird ein Wert generiert.

Das Dienstprogramm IMPORT führt keine zusätzliche Gültigkeitsprüfung der benutzerdefinierten Identitätswerte durch, die über die übliche Gültigkeitsprüfung für Werte mit dem Datentyp der Identitätsspalte (also SMALLINT, INT, BIGINT oder DECIMAL) hinausgeht. Falls Werte doppelt vorhanden sind, wird keine entsprechende Meldung ausgegeben. Außerdem kann der Änderungswert compound=x beim Importieren von Daten in eine Tabelle mit einer Identitätsspalte nicht verwendet werden.

Das Dienstprogramm IMPORT unterstützt zwei Änderungswerte für den Dateityp, die die Verwendung dieses Dienstprogramms für Tabellen mit einer Identitätsspalte vereinfachen:

- Der Änderungswert identitymissing vereinfacht das Importieren einer Tabelle mit einer Identitätsspalte, wenn die Eingabedatendatei für die Identitätsspalte keine Werte (auch keine Nullwerte) enthält.

Angenommen, es wurde beispielsweise eine Tabelle mit der folgenden SQL-Anweisung definiert:

```
create table table1 (c1 char(30),
                    c2 int generated by default as identity,
                    c3 real,
                    c4 char(1))
```

Ein Benutzer möchte unter Umständen Daten aus einer Datei (`import.del`) in die Tabelle `TABLE1` importieren, und diese Daten wurden möglicherweise aus einer Tabelle exportiert, die keine Identitätsspalte enthält. Es folgt ein Beispiel für eine solche Datei.

```
Robert, 45.2, J
Mike, 76.9, K
Leo, 23.4, I
```

Eine Methode für das Importieren dieser Datei wäre das explizite Auflisten der zu importierenden Spalten durch den folgenden Befehl `IMPORT`:

```
db2 import from import.del of del replace into table1 (c1, c3, c4)
```

Bei einer Tabelle mit vielen Spalten ist die Verwendung dieser Syntax jedoch eventuell umständlich und fehlerträchtig. Eine alternative Methode für das Importieren der Datei ist die folgende Verwendung des Änderungswerts `identitymissing` für den Datentyp:

```
db2 import from import.del of del modified by identitymissing
replace into table1
```

- Der Änderungswert `identityignore` stellt in gewisser Hinsicht das Gegenteil des Änderungswerts `identitymissing` dar: Er weist das Dienstprogramm `IMPORT` an, dass die in der Eingabedatendatei vorhandenen Werte für die Identitätsspalte ignoriert werden sollen und dass für jede Zeile ein Identitätswert generiert werden soll. Beispiel: Ein Benutzer möchte die folgenden Daten aus einer Datei (`import.del`) in die wie oben beschrieben definierte Tabelle `TABLE1` importieren:

```
Robert, 1, 45.2, J
Mike, 2, 76.9, K
Leo, 3, 23.4, I
```

Wenn die benutzerdefinierten Werte 1, 2 und 3 nicht für die Identitätsspalte verwendet werden sollen, könnte der Benutzer den folgenden Befehl `IMPORT` absetzen:

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

Aber auch diese Methode kann möglicherweise umständlich und fehlerträchtig sein, wenn die Tabelle zu viele Spalten enthält. Der Änderungswert `identityignore` vereinfacht die Syntax folgendermaßen:

```
db2 import from import.del of del modified by identityignore
replace into table1
```

Beim Exportieren einer Tabelle mit einer Identitätsspalte in eine IXF-Datei können die Optionen `REPLACE_CREATE` und `CREATE` des Befehls `IMPORT` verwendet werden, um die Tabelle, einschließlich ihrer Identitätsspaltenmerkmale, erneut zu erstellen. Wird eine solche IXF-Datei aus einer Tabelle mit einer als `GENERATED ALWAYS` definierten Identitätsspalte erstellt, kann die Daten-datei nur dann erfolgreich importiert werden, wenn der Änderungswert `identityignore` angegeben wird. Andernfalls werden alle Zeilen zurückgewiesen (SQL3550W).

#### Zugehörige Konzepte:

- „Identitätsspalten“ in *Systemverwaltung: Konzept*

---

## Verwenden von IMPORT mit generierten Spalten

Mit dem Dienstprogramm IMPORT können Sie Daten in eine Tabelle importieren, die generierte Spalten enthält, bei denen es sich nicht um Identitätsspalten handelt.

Sofern keine Änderungswerte für den Dateityp verwendet werden, die sich auf generierte Spalten beziehen, liegen der Ausführung des Dienstprogramms die folgenden Regeln zugrunde:

- Für eine generierte Spalte wird immer dann ein Wert generiert, wenn die entsprechende Zeile in der Eingabedatei keinen Wert für die Spalte enthält oder einen Nullwert explizit vorgibt. Ist für eine generierte Spalte ein Wert angegeben, der kein Nullwert ist, wird die Zeile zurückgewiesen (SQL3550W).
- Generiert der Server einen Nullwert für eine generierte Spalte, die keinen Nullwert enthalten darf, wird die Datenzeile, zu der dieses Feld gehört, zurückgewiesen (SQL0407N). Dies könnte beispielsweise dann eintreten, wenn eine generierte Spalte, die keinen Nullwert enthalten darf, als Summe zweier Tabellenspalten definiert wurde und für diese Spalten in der Eingabedatei Nullwerte zur Verfügung gestellt wurden.

Das Dienstprogramm IMPORT unterstützt zwei Änderungswerte für den Dateityp, die die Verwendung dieses Dienstprogramms für Tabellen mit generierten Spalten vereinfachen:

- Der Änderungswert `generatedmissing` vereinfacht das Importieren einer Tabelle mit generierten Spalten, wenn die Eingabedatei für alle in der Tabelle vorhandenen generierten Spalten keine Werte (auch keine Nullwerte) enthält. Angenommen, es wurde beispielsweise eine Tabelle mit der folgenden SQL-Anweisung definiert:

```
create table table1 (c1 int,
                    c2 int,
                    g1 int generated always as (c1 + c2),
                    g2 int generated always as (2 * c1),
                    c3 char(1))
```

Ein Benutzer möchte unter Umständen Daten aus einer Datei (`load.del`) in die Tabelle `TABLE1` importieren, und diese Daten wurden möglicherweise aus einer Tabelle exportiert, die keine generierten Spalten enthält. Es folgt ein Beispiel für eine solche Datei.

```
1, 5, J
2, 6, K
3, 7, I
```

Eine Methode für das Importieren dieser Datei wäre das explizite Auflisten der zu importierenden Spalten durch den folgenden Befehl `IMPORT`:

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

Bei einer Tabelle mit vielen Spalten ist die Verwendung dieser Syntax jedoch eventuell umständlich und fehlerträchtig. Eine alternative Methode für das Importieren der Datei ist die folgende Verwendung des Änderungswerts `generatedmissing` für den Dateityp:

```
db2 import from import.del of del modified by generatedmissing
replace into table1
```

- Der Änderungswert `generatedignore` stellt in gewisser Hinsicht das Gegenteil des Änderungswerts `generatedmissing` dar: Er weist das Dienstprogramm `IMPORT` an, dass die in der Eingabedatei vorhandenen Werte für alle generierten Spalten ignoriert und für jede Zeile Werte generiert werden sollen. Beispiel: Ein Benutzer möchte die folgenden Daten aus einer Datei (`import.del`) in die wie oben beschrieben definierte Tabelle `TABLE1` importieren:

```
1, 5, 10, 15, J
2, 6, 11, 16, K
3, 7, 12, 17, I
```

Die benutzerdefinierten Werte 10, 11, und 12 (für g1) sowie 15, 16 und 17 (für g2), die keine Nullwerte sind, bewirken, dass die Zeile zurückgewiesen wird (SQL3550W). Um dies zu verhindern, könnte der Benutzer den folgenden Befehl IMPORT absetzen:

```
db2 import from import.del of del method P(1, 2, 5)
replace into table1 (c1, c2, c3)
```

Aber auch diese Methode kann möglicherweise umständlich und fehlerträchtig sein, wenn die Tabelle zu viele Spalten enthält. Der Änderungswert generatedignore vereinfacht die Syntax folgendermaßen:

```
db2 import from import.del of del modified by generatedignore
replace into table1
```

#### Zugehörige Konzepte:

- „Generated Columns“ in *Application Development Guide: Programming Client Applications*

---

## Erneutes Erstellen einer exportierten Tabelle mit IMPORT

Mit dem Dienstprogramm IMPORT können Sie eine Tabelle erneut erstellen, die mit dem Dienstprogramm EXPORT gesichert wurde. Die Tabelle muss in eine IXF-Datei exportiert worden sein, und die bei der Exportoperation verwendete SELECT-Anweisung muss bestimmte Bedingungen erfüllt haben (z. B. dürfen keine Spaltennamen in der Klausel SELECT verwendet werden; es ist nur `select *` erlaubt). Beim Erstellen einer Tabelle aus einer IXF-Datei bleiben nicht alle Attribute der Originaltabelle erhalten. Zum Beispiel werden referenzielle Integritätsbedingungen, Fremdschlüsseldefinitionen und benutzerdefinierte Datentypen nicht beibehalten. Attribute der Originaltabelle, die erhalten bleiben, sind Folgende:

- Name und Definition des Primärschlüssels
- Namen und Definitionen von eindeutigen Integritätsbedingungen, jedoch keine anderen Arten von Integritätsbedingungen oder Auslösern
- Spalteninformationen:
  - Spaltenname
  - Spaltendatentypen, einschließlich benutzerdefinierte einzigartige Datentypen, die als Basistyp beibehalten werden
  - Identitätsmerkmale
  - Längen (außer bei Typen `lob_file`)
  - Codepage (sofern zutreffend)
  - DATALINK-Optionen
  - Identitätsoptionen
  - Angabe, ob die Spalte laut Definition Nullwerte enthalten darf oder nicht
  - Standardwerte für Konstanten, sofern vorhanden, jedoch keine anderen Arten von Standardwerten
- Indexinformationen:
  - Indexname
  - Name des Indexerstellers
  - Spaltennamen sowie die Angabe, ob die jeweilige Spalte in aufsteigenden oder absteigender Reihenfolge sortiert wird
  - Angabe, ob der Index als eindeutig definiert ist

- Angabe, ob es sich um einen Clusterindex handelt
- Angabe, ob der Index Rückwärtsdurchsuchen zulässt
- Werte für *pctfree*
- Werte für *minpctused*

Die folgenden Attribute der Originaltabelle bleiben *nicht* erhalten:

- Angabe, ob die Quelle eine normale Tabelle, eine gespeicherte Abfragetabelle, eine Sicht oder eine Spaltengruppe aus einer oder allen diesen Quellen war
- Tabelleninformationen:
  - Definition der gespeicherten Abfragetabelle (sofern zutreffend)
  - Optionen für die gespeicherte Abfragetabelle (sofern zutreffend)
  - Optionen für den Tabellenbereich (diese Informationen können jedoch durch den Befehl IMPORT angegeben werden)
- Spalteninformationen:
  - Gegebenenfalls vorhandene Standardwerte (außer Werte für Konstanten)
  - LOB-Optionen (sofern vorhanden)
  - Verweisklausel der Anweisung CREATE TABLE (sofern vorhanden)
  - Referenzielle Integritätsbedingungen (sofern vorhanden)
  - Prüfung auf Integritätsbedingungen (sofern vorhanden)
  - Optionen für generierte Spalten (sofern vorhanden)
- Indexinformationen:
  - INCLUDE-Spalten (sofern vorhanden)

**Zugehörige Konzepte:**

- „Erneutes Erstellen einer exportierten Tabelle“ auf Seite 4

## Importieren von großen Objekten (LOBs)

Beim Importieren von Daten in LOB-Spalten können die betreffenden Daten entweder aus derselben Datei, aus der auch die übrigen Spaltendaten entnommen werden, oder aus separaten Dateien kommen. Falls die Daten aus separaten Dateien stammen, muss der Änderungswert für den Datentyp LOBSINFILE angegeben werden.

Die Spalte in der Haupteingabedatei enthält entweder die Importdaten (Standardwert) oder den Namen einer Datei, in der die Importdaten gespeichert sind.

**Anmerkungen:**

1. Wenn LOB-Daten in der Haupteingabedatei gespeichert sind, sind nur 32 KB Daten zulässig. Warnungen, dass Daten abgeschnitten werden, werden ignoriert.
2. Alle LOB-Daten müssen in der Hauptdatei oder jedes große Objekt muss in einer separaten Datei gespeichert sein. Die Hauptdatei kann keine Mischung aus LOB-Daten und Dateinamen enthalten. LOB-Werte werden mit Hilfe des Änderungswerts *lobsinfile* und der Klausel LOBS FROM aus separaten Dateien importiert.

Mit einer LOB-Positionskennung (LOB Location Specifier - LLS) können beim Importieren, Exportieren und Laden von LOB-Informationen mehrere große Objekte in einer gemeinsamen Datei gespeichert werden.

Eine LOB-Positionskennung ist eine Zeichenfolge, die angibt, wo sich LOB-Daten in einer Datei befinden. Das Format der LOB-Positionskennung lautet `dateiname.erw.nnn.mmm/`. Hierbei steht `dateiname.erw` für den Namen der Datei, die das große Objekt (LOB) enthält, `nnn` für die (in Byte angegebene) relative Position des großen Objekts in der Datei und `mmm` für die Länge des großen Objekts (in Byte). Die LOB-Positionskennung `db2exp.001:123:456/` gibt beispielsweise an, dass sich das große Objekt in der Datei `db2exp.001` befindet, an einer relativen Position von 123 Byte in der Datei beginnt und 456 Byte lang ist. Wenn in der LOB-Positionskennung die Größe 0 angegeben ist, wird angenommen, dass das große Objekt die Länge 0 hat. Lautet der Wert für die Länge -1, wird angenommen, dass das große Objekt NULL ist, und relative Position sowie Dateiname werden ignoriert.

Wird beim Importieren oder Laden von Daten die Option `modified by lobinfile` angegeben, wird für alle entsprechenden LOB-Spalten eine LOB-Positionskennung erwartet. Falls für eine LOB-Spalte eine andere Angabe als die LOB-Positionskennung gefunden wird, behandelt die Datenbank sie als eine LOB-Datei und lädt die gesamte Datei als großes Objekt.

#### Zugehörige Referenzen:

- „Allgemeine Regeln für den Import von PC/IXF-Dateien in Datenbanken“ auf Seite 315
- „Datentypspezifische Regeln für den Import von PC/IXF-Dateien in Datenbanken“ auf Seite 317
- „Befehl IMPORT“ auf Seite 39
- „Large objects (LOBs)“ in *SQL Reference, Volume 1*

---

## Importieren von benutzerdefinierten einzigartigen Datentypen (UDTs)

Das Dienstprogramm IMPORT setzt die benutzerdefinierten einzigartigen Datentypen (UDTs) automatisch in ähnliche Basisdatentypen um. Dies erspart Ihnen das explizite Umsetzen der UDTs in die Basisdatentypen. Durch die Umsetzung werden Vergleiche zwischen UDTs und den Basisdatentypen in SQL ermöglicht.

#### Zugehörige Konzepte:

- „User-defined distinct types“ in *Application Development Guide: Programming Server Applications*

---

## Sperrungen von Tabellen während des Imports

Das Dienstprogramm IMPORT unterstützt zwei Modi zum Sperren von Tabellen. Der Offlinemodus (`ALLOW NO ACCESS`) verhindert den Zugriff auf Tabellendaten durch gleichzeitig ablaufende Anwendungen. Dies ist der Standardmodus. Der Onlinemodus (`ALLOW WRITE ACCESS`) ermöglicht gleichzeitig ablaufenden Anwendungen sowohl Lese- als auch Schreibzugriff auf die Importzieltabelle.

In der Standardeinstellung ist das Dienstprogramm IMPORT an die Datenbank mit der Isolationsstufe RS (Read Stability - Lesestabilität) gebunden.

#### Onlineimport (`ALLOW WRITE ACCESS`):

Das Dienstprogramm IMPORT fordert eine nichtexklusive Sperre (IX) für die Zieltabelle an.

Das Halten dieser Sperre für die Tabelle hat folgende Auswirkungen:

- Wenn andere Anwendungen eine inkompatible Tabellensperre halten, beginnt das Dienstprogramm IMPORT erst dann mit dem Einfügen von Daten, nachdem alle betreffenden Anwendungen ihre Änderungen mit einer COMMIT-Operation festgeschrieben oder mit einer ROLLBACK-Operation zurückgesetzt haben.
- Während der Ausführung der Importoperation warten alle anderen Anwendungen, die eine inkompatible Tabellensperre anfordern, bis die Importoperation die aktuelle Transaktion entweder festschreibt oder zurücksetzt. Hierbei ist zu beachten, dass die Tabellensperre von IMPORT nicht über Transaktionsgrenzen hinaus bestehen bleibt. Aus diesem Grund muss beim Onlineimport nach jeder COMMIT-Operation eine Tabellensperre angefordert und unter Umständen auf diese gewartet werden.
- Wenn andere Anwendungen eine inkompatible Zeilensperre halten, stoppt das Dienstprogramm IMPORT das Einfügen von Daten solange, bis alle betreffenden Anwendungen ihre Änderungen entweder festgeschrieben oder zurückgesetzt haben.
- Während der Ausführung der Importoperation warten alle anderen Anwendungen, die eine inkompatible Zeilensperre anfordern, bis die Importoperation die aktuelle Transaktion entweder festschreibt oder zurücksetzt.

Um die Onlinemerkmale beizubehalten und das Risiko eines gegenseitigen Sperrens zu reduzieren, wird die Onlineimportoperation die aktuelle Transaktion periodisch festschreiben und alle Zeilensperren freigeben, bevor eine Eskalation zu einer exklusiven Tabellensperre (X) erfolgt. Daher werden während einer Onlineimportoperation möglicherweise COMMIT-Operationen ausgeführt, auch wenn die Option 'commitcount' nicht verwendet wurde. Es kann entweder die Häufigkeit der COMMIT-Operationen explizit angegeben oder der Festschreibungsmodus AUTOMATIC verwendet werden. Wird für 'commitcount' explizit der Wert Null angegeben, werden keine COMMIT-Operationen ausgeführt. Hierbei ist zu beachten, dass ein gegenseitiges Sperren erfolgt, wenn die gleichzeitig ablaufende Anwendung, die eine inkompatible Zeilensperre hält, versucht, eine Eskalation zu einer Tabellensperre durchzuführen.

IMPORT wird im Onlinemodus ausgeführt, wenn 'ALLOW WRITE ACCESS' angegeben ist. Der Onlinemodus ist mit folgenden Vorgängen nicht kompatibel:

- Importmodi REPLACE, CREATE und REPLACE\_CREATE
- Gepufferte INSERT-Operationen
- Importen in eine Zielsicht
- Importen in eine Hierarchietabelle
- Importen in eine Zieltabelle mit Tabellensperrgröße

#### **Offlineimport (ALLOW NO ACCESS):**

Wird eine große Anzahl von Zeilen in eine Tabelle importiert, kann die vorhandene Sperre unter Umständen zu einer exklusiven Sperre eskaliert, d. h. erweitert werden. Wenn eine andere Anwendung, die mit derselben Tabelle arbeitet, einige Zeilensperren hält, tritt ein gegenseitiges Sperren auf, falls sich die Sperre zu einer exklusiven Sperre erweitert. Um dies zu verhindern, fordert das Dienstprogramm IMPORT zu Beginn der Verarbeitung eine exklusive Sperre für die Tabelle an. Dies ist das Standardverhalten beim Import.

Das Halten einer Sperre für die Tabelle hat zwei Auswirkungen. Erstens wartet das Dienstprogramm IMPORT, wenn andere Anwendungen eine Tabellensperre halten oder für die Zieltabelle der Importoperation Zeilensperren vorhanden sind, bis alle diese Anwendungen ihre Änderungen festschreiben oder rückgängig machen.

Zweitens warten während der Ausführung der Importoperation alle anderen Anwendungen, die eine Sperre anfordern, bis die Importoperation vollständig ausgeführt wurde. IMPORT wird im Offlinemodus ausgeführt, wenn 'ALLOW WRITE ACCESS' nicht angegeben ist.

**Zugehörige Konzepte:**

- „Tabellensperren, Statuswerte für Tabellen und Statuswerte für Tabellenbereiche“ auf Seite 186

---

## Befehl IMPORT

Fügt Daten aus einer externen Datei mit einem unterstützten Dateiformat in eine Tabelle, eine Hierarchie oder eine Sicht ein. Die Verwendung von LOAD ist eine schnellere Alternative. Das Dienstprogramm LOAD unterstützt jedoch nicht das Laden von Daten auf Hierarchieebene.

**Berechtigung:**

- Für die Verwendung von IMPORT mit der Option INSERT ist eine der folgenden Berechtigungen erforderlich:
  - *sysadm*
  - *dbadm*
  - Zugriffsrecht CONTROL für jede betroffene Tabelle oder Sicht
  - Zugriffsrecht INSERT und SELECT für jede betroffene Tabelle oder Sicht
- Für die Verwendung von IMPORT für eine vorhandene Tabelle mit der Option INSERT\_UPDATE ist eine der folgenden Berechtigungen erforderlich:
  - *sysadm*
  - *dbadm*
  - Zugriffsrecht CONTROL für die Tabelle bzw. Sicht
  - Zugriffsrecht INSERT, SELECT, UPDATE und DELETE für jede betroffene Tabelle oder Sicht
- Für die Verwendung von IMPORT für eine bestehende Tabelle mit der Option REPLACE oder REPLACE\_CREATE ist eine der folgenden Berechtigungen erforderlich:
  - *sysadm*
  - *dbadm*
  - Zugriffsrecht CONTROL für die Tabelle bzw. Sicht
  - Zugriffsrecht INSERT, SELECT und DELETE für die Tabelle oder Sicht
- Für die Verwendung von IMPORT für eine neue Tabelle mit der Option CREATE oder REPLACE\_CREATE ist eine der folgenden Berechtigungen erforderlich:
  - *sysadm*
  - *dbadm*
  - Berechtigung CREATETAB für die Datenbank und Zugriffsrecht USE für den Tabellenbereich sowie eine der beiden Folgenden:
    - Berechtigung IMPLICIT\_SCHEMA für die Datenbank, falls der Name des impliziten oder expliziten Schemas der Tabelle nicht vorhanden ist
    - Zugriffsrecht CREATIN für das Schema, wenn der Schemaname der Tabelle auf ein vorhandenes Schema verweist

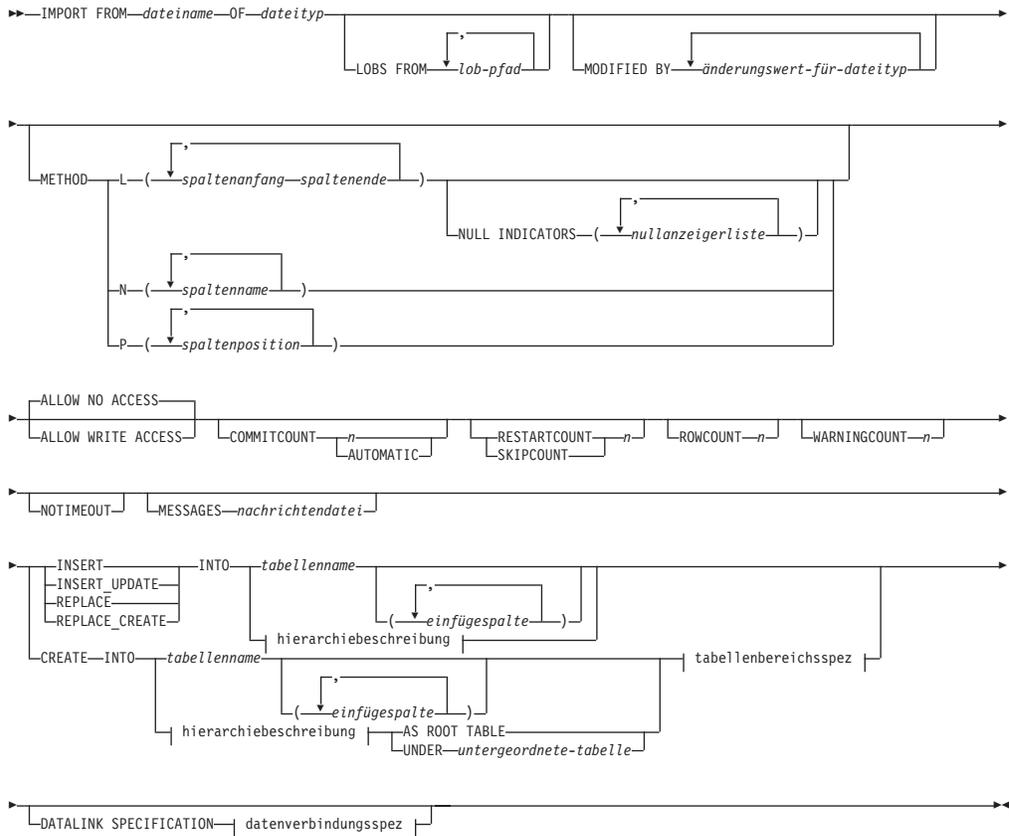
# Befehl IMPORT

- Für die Verwendung von IMPORT für eine vorhandene Hierarchie mit der Option REPLACE ist eine der folgenden Berechtigungen erforderlich:
  - *sysadm*
  - *dbadm*
  - Zugriffsrecht CONTROL für alle untergeordneten Tabellen in der Hierarchie

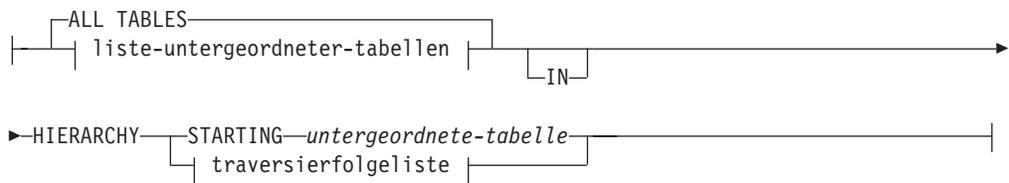
## Erforderliche Verbindung:

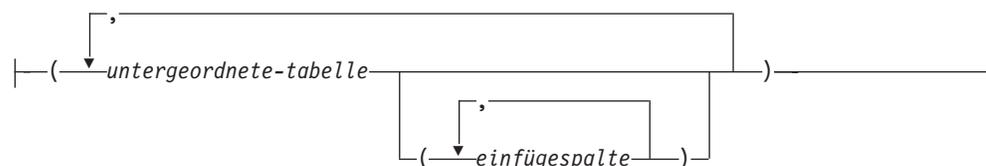
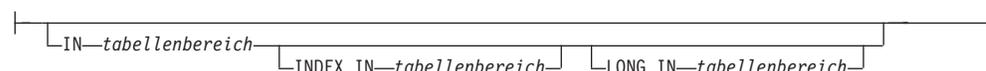
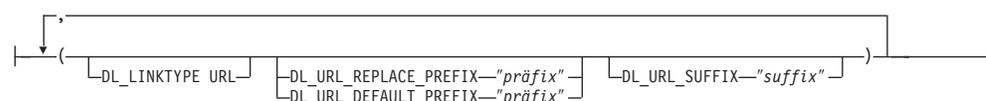
Datenbank. Wenn das implizite Herstellen von Verbindungen aktiviert ist, wird eine Verbindung zur Standarddatenbank hergestellt.

## Befehlssyntax:



## hierarchiebeschreibung:



**liste-untergeordneter-tabellen:****traversierfolgeliste:****tabellenbereichsspez:****datenverbindungsspez:****Befehlsparameter:****ALL TABLES**

Ein implizites Schlüsselwort nur für die Hierarchie. Beim Importieren einer Hierarchie werden standardmäßig alle Tabellen importiert, die in einer Traversierfolge angegeben sind.

**ALLOW NO ACCESS**

Führt IMPORT im Offlinemodus aus. Eine exklusive Sperre (X) für die Zieltabelle wird angefordert, bevor Zeilen eingefügt werden. Dies verhindert, dass gleichzeitig ablaufende Anwendungen auf Tabellendaten zugreifen. Dies ist das Standardverhalten beim Import.

**ALLOW WRITE ACCESS**

Führt IMPORT im Onlinemodus aus. Eine exklusive Anwartschaftssperre (IX = Intention Exclusive Lock) für die Zieltabelle wird angefordert, wenn die erste Zeile eingefügt wird. Dadurch können gleichzeitig ablaufende Eingabe- und Ausgabeprogramme auf Tabellendaten zugreifen. Der Onlinemodus ist mit den Importoptionen REPLACE, CREATE und REPLACE\_CREATE nicht kompatibel. Der Onlinemodus wird nicht zusammen mit gepufferten INSERT-Operationen unterstützt. Die Importoperation führt für eingefügte Daten periodisch COMMIT-Operationen durch, um eine Sperreneskalation auf eine Tabellensperre zu vermeiden und um zu gewährleisten, dass die aktive Protokolldatei über ausreichend Speicher verfügt. Diese COMMIT-Operationen werden auch dann ausgeführt, wenn die Option COMMITCOUNT nicht verwendet wurde. Bei jeder COMMIT-Operation verliert IMPORT die Tabellensperre IX und versucht, diese nach jeder COMMIT-Operation erneut anzufordern.

## Befehl IMPORT

### AS ROOT TABLE

Erstellt eine oder mehrere untergeordnete Tabellen als eigenständige Tabellenhierarchie.

### COMMITCOUNT *n*/AUTOMATIC

Führt eine COMMIT-Operation aus, nachdem jeweils *n* Datensätze importiert wurden. Ist ein Wert für *n* angegeben, führt IMPORT eine COMMIT-Operation aus, nachdem jeweils *n* Datensätze importiert wurden. Werden Compound-INSERT-Operationen verwendet, wird eine benutzerdefinierte COMMIT-Häufigkeit von *n* auf das erste ganzzahlige Vielfache des Wertes für den Compound-Zähler aufgerundet. Bei Angabe von AUTOMATIC ermittelt IMPORT intern, wann eine COMMIT-Operation ausgeführt werden muss. Das Dienstprogramm führt COMMIT-Operationen aus einem der beiden folgenden Gründe durch:

- Um zu gewährleisten, dass für die aktive Protokolldatei ausreichend Speicher verfügbar ist.
- Um eine Sperrenskalation von Zeilenebene auf Tabellenebene zu vermeiden.

Ist die Option ALLOW WRITE ACCESS angegeben und die Option COMMITCOUNT ist nicht angegeben, führt das Dienstprogramm IMPORT die COMMIT-Operationen so aus, als wäre COMMITCOUNT AUTOMATIC angegeben.

### CREATE

Erstellt die Tabellendefinition und den Zeileninhalt in der Codepage der Datenbank. Wenn die Daten aus einer DB2-Tabelle, untergeordneter Tabelle oder Hierarchie exportiert wurden, werden Indizes erstellt. Wenn diese Option für eine Hierarchie verwendet wird und Daten aus DB2 exportiert wurden, wird auch eine Typhierarchie erstellt. Diese Option kann nur mit IXF-Dateien verwendet werden.

**Anmerkung:** Wenn die Daten aus einer MVS-Host-Datenbank exportiert wurden und LONGVAR-Felder enthalten, deren Länge, berechnet anhand der Seitengröße, kürzer ist als 254, kann CREATE fehlschlagen, weil die Zeilen zu lang sind. Eine Liste der Einschränkungen finden Sie unter Erneutes Erstellen einer exportierten Tabelle mit IMPORT. In diesem Fall muss die Tabelle manuell erstellt und IMPORT mit INSERT aufgerufen werden. Alternativ kann auch der Befehl LOAD verwendet werden.

### DATALINK SPECIFICATION

Für jede DATALINK-Spalte ist eine Spaltenangabe in runden Klammern möglich. Jede Spaltenangabe umfasst mindestens einen DL\_LINKTYPE-Wert, eine Präfix- und eine DL\_URL\_SUFFIX-Angabe. Die Präfixangabe kann entweder DL\_URL\_REPLACE\_PREFIX oder DL\_URL\_DEFAULT\_PREFIX sein.

Es sind so viele DATALINK-Spaltenangaben möglich, wie DATALINK-Spalten in der Tabelle definiert sind. Die Reihenfolge der Angaben richtet sich nach der Reihenfolge der DATALINK-Spalten in der für *einfügespalte* angegebenen Liste oder in der Tabellendefinition (wenn keine Liste für *einfügespalte* angegeben wird).

### DL\_LINKTYPE

Muss, wenn angegeben, mit dem LINKTYPE-Wert der Spaltendefinition

übereinstimmen. Demzufolge ist DL\_LINKTYPE URL akzeptabel, wenn die Spaltendefinition LINKTYPE URL angibt.

#### **DL\_URL\_DEFAULT\_PREFIX "präfix"**

Muss, wenn angegeben, als Standardpräfix für alle DATALINK-Werte in der gleichen Spalte verwendet werden. In diesem Zusammenhang bezieht sich Präfix auf den Teil "Schema-Host-Port" der URL-Angabe.

Beispiele für Präfixe sind:

```
"http://server"
"file://server"
"file:"
"http://server:80"
```

Enthalten die Daten einer Spalte kein Präfix und wird mit DL\_URL\_DEFAULT\_PREFIX ein Standardpräfix angegeben, wird dieses dem Spaltenwert (sofern nicht NULL) vorangestellt.

Beispiel: Wenn DL\_URL\_DEFAULT\_PREFIX das Standardpräfix "http://toronto" angibt, gilt Folgendes:

- Der Spalteneingabewert "/x/y/z" wird als "http://toronto/x/y/z" gespeichert.
- Der Spalteneingabewert "http://coyote/a/b/c" wird als "http://coyote/a/b/c" gespeichert.
- Der Spalteneingabewert NULL wird als NULL gespeichert.

#### **DL\_URL\_REPLACE\_PREFIX "präfix"**

Diese Klausel ist zum Laden oder Importieren von Daten nützlich, die zuvor vom Dienstprogramm EXPORT generiert wurden, wenn der Benutzer den Host-Namen in den Daten global durch einen anderen Host-Namen ersetzen möchte. Wird, wenn angegeben, das Präfix für *alle* Spaltenwerte, die nicht NULL sind. Wenn ein Spaltenwert mit einem Präfix versehen ist, wird dieses durch das mit DL\_URL\_REPLACE\_PREFIX angegebene Präfix ersetzt. Wenn ein Spaltenwert kein Präfix aufweist, wird das mit DL\_URL\_REPLACE\_PREFIX angegebene Präfix dem Spaltenwert vorangestellt.

Beispiel: Wenn DL\_URL\_REPLACE\_PREFIX das Präfix "http://toronto" angibt, gilt Folgendes:

- Der Spalteneingabewert "/x/y/z" wird als "http://toronto/x/y/z" gespeichert.
- Der Spalteneingabewert "http://coyote/a/b/c" wird als "http://toronto/a/b/c" gespeichert. Beachten Sie, dass "toronto" "coyote" ersetzt.
- Der Spalteneingabewert NULL wird als NULL gespeichert.

#### **DL\_URL\_SUFFIX "suffix"**

Wird, wenn angegeben, an jeden Spaltenwert für die Spalte, der nicht NULL ist, angefügt. Wird an die Pfadkomponente des URL-Teils des DATALINK-Werts angefügt.

#### **FROM dateiname**

Gibt die Datei an, die die zu importierenden Daten enthält. Wenn kein Pfad angegeben wird, wird das aktuelle Arbeitsverzeichnis verwendet.

#### **HIERARCHY**

Gibt an, dass hierarchische Daten importiert werden sollen.

### IN tabellenbereich

Gibt den Tabellenbereich an, in dem die Tabelle erstellt wird. Der Tabellenbereich muss vorhanden und ein Tabellenbereich des Typs REGULAR sein. Wenn kein anderer Tabellenbereich angegeben ist, werden alle Teile der Tabelle in diesem Tabellenbereich gespeichert. Wenn diese Klausel nicht angegeben wird, wird die Tabelle in einem Tabellenbereich erstellt, der von der Berechtigungs-ID erstellt wird. Wenn keiner gefunden wird, wird die Tabelle in den Standardtabellenbereich USERSPACE1 gestellt. Wenn USERSPACE1 gelöscht wurde, schlägt die Erstellung der Tabelle fehl.

### INDEX IN tabellenbereich

Gibt den Tabellenbereich an, in dem Indizes für die Tabelle erstellt werden. Diese Option ist nur zulässig, wenn der primäre Tabellenbereich, der in der Klausel IN angegeben ist, ein DMS-Tabellenbereich ist. Der Tabellenbereich muss vorhanden und ein DMS-Tabellenbereich des Typs REGULAR oder LARGE sein.

**Anmerkung:** Welcher Tabellenbereich einen Index enthält, kann nur beim Erstellen der Tabelle angegeben werden.

### einfügespalte

Gibt den Namen einer Spalte in der Tabelle oder der Sicht an, in die Daten eingefügt werden sollen.

### INSERT

Fügt die importierten Daten in die Tabelle ein, ohne die vorhandenen Tabellendaten zu ändern.

### INSERT\_UPDATE

Fügt Zeilen mit importierten Daten in die Zieltabelle ein oder aktualisiert vorhandene Zeilen (der Zieltabelle) mit entsprechenden Primärschlüsseln.

### INTO tabellenname

Gibt die Datenbanktabelle an, in die die Daten importiert werden sollen. Diese Tabelle darf keine Systemtabelle, keine deklarierte temporäre Tabelle und keine Übersichtstabelle sein.

Es kann ein Aliasname für INSERT, INSERT\_UPDATE oder REPLACE verwendet werden. Ausgenommen hiervon ist die Verwendung eines Servers mit einem älteren Versionsstand. In diesem Fall muss der vollständig qualifizierte Tabellename oder der Tabellename ohne Qualifikationsmerkmal verwendet werden. Ein qualifizierter Tabellename hat folgendes Format: *schema.tabellenname*. Dabei ist *schema* der Benutzername, mit dem die Tabelle erstellt wurde.

### LOBS FROM lob-pfad

Gibt einen oder mehrere Pfade an, in denen LOB-Dateien gespeichert werden. Die Namen der LOB-Dateien werden in der Hauptdatendatei (ASC, DEL oder IXF) in der Spalte gespeichert, die in die LOB-Spalte geladen wird. Diese Option wird ignoriert, wenn der Änderungswert *lobsinfile* nicht angegeben wird.

### LONG IN tabellenbereich

Gibt den Tabellenbereich an, in dem die Werte von Spalten für große Objekte (LOBs) (LONG VARCHAR, LONG VARGRAPHIC, LOB-Datentypen oder einzigartige Datentypen mit einem dieser Typen als Quellentyp) gespeichert werden. Diese Option ist nur zulässig, wenn der primäre Tabellenbereich, der in der Klausel IN angegeben ist, ein DMS-Tabellenbereich ist. Der Tabellenbereich muss vorhanden und ein DMS-Tabellenbereich des Typs LARGE sein.

**MESSAGES nachrichtendatei**

Gibt das Ziel für Warnungen und Fehlernachrichten an, die bei einer Importoperation auftreten. Wenn die Datei bereits vorhanden ist, werden die Informationen vom Dienstprogramm IMPORT an sie angefügt. Wenn der vollständige Pfad zu der Datei nicht angegeben wird, verwendet das Dienstprogramm das aktuelle Verzeichnis und das Standardlaufwerk als Ziel. Wenn kein Wert für *nachrichtendatei* angegeben wird, werden die Nachrichten in die Standardausgabe geschrieben.

**METHOD**

**L** Gibt die Anfangs- und Endnummer der Spalten an, aus denen Daten importiert werden sollen. Eine Spaltennummer ist eine relative Byteadresse in Bezug auf den Anfang einer Datenzeile. Ihre Nummerierung beginnt bei 1.

**Anmerkung:** Diese Methode kann nur mit ASC-Dateien verwendet werden und ist die einzige gültige Option für diesen Dateityp.

**N** Gibt die Namen der zu importierenden Spalten an.

**Anmerkung:** Diese Methode kann nur mit IXF-Dateien verwendet werden.

**P** Gibt die Feldnummern der zu importierenden Eingabedatenfelder an.

**Anmerkung:** Diese Methode kann nur mit IXF- oder DEL-Dateien verwendet werden und ist die einzige gültige Option für den Dateityp DEL.

**MODIFIED BY änderungswert-für-dateityp**

Gibt Optionen für Änderungswerte für Dateitypen an. Weitere Informationen hierzu finden Sie in File type modifiers for import.

**NOTIMEOUT**

Gibt an, dass es für das Dienstprogramm IMPORT beim Warten auf Sperren kein Zeitlimit gibt. Diese Option hat Vorrang vor dem Datenbankkonfigurationsparameter *locktimeout*. Andere Anwendungen sind nicht betroffen.

**NULL INDICATORS nullanzeigerliste**

Diese Option kann nur verwendet werden, wenn der Parameter METHOD L angegeben wurde. D. h., die Eingabedatei ist eine ASC-Datei. Die Nullanzeigerliste ist eine durch Kommata gegliederte Liste positiver ganzer Zahlen, die für jedes Nullanzeigerfeld die Spaltennummer angeben. Die Spaltennummer ist die relative Byteadresse des Nullanzeigerfelds in Bezug auf den Anfang einer Datenzeile. Die Nullanzeigerliste muss für jedes Datenfeld, das im Parameter METHOD L definiert ist, einen Wert enthalten. Eine Spaltennummer 0 gibt an, dass das entsprechende Datenfeld immer Daten enthält.

Y in der Nullanzeigerspalte gibt an, dass die Spaltendaten NULL sind. Jedes *andere* Zeichen als Y in der Nullanzeigerspalte gibt an, dass die Spaltendaten nicht NULL sind und dass die mit der Option METHOD L angegebenen Spaltendaten importiert werden.

Das Nullanzeigerzeichen kann geändert werden, indem die Option MODIFIED BY mit dem Änderungswert `nullindchar` für Dateityp verwendet wird.

## Befehl IMPORT

### OF dateityp

Gibt das Format der Daten in der Eingabedatei an:

- ASC (ASCII-Format mit universellen Zeilenbegrenzern)
- DEL (ASCII-Format ohne universelle Zeilenbegrenzer), das von einer Vielzahl von Datenbankmanager- und Dateimanagerprogrammen verwendet wird.
- WSF (Arbeitsblattformat), das von Programmen wie den folgenden verwendet wird:
  - Lotus 1-2-3
  - Lotus Symphony
- IXF (Integrated Exchange Format, PC-Version), d. h. es wurde aus der gleichen oder einer anderen DB2-Tabelle importiert. Eine IXF-Datei enthält auch die Tabellendefinition und Definitionen vorhandener Indizes, sofern keine Spalten in der SELECT-Anweisung angegeben sind.

### REPLACE

Löscht alle vorhandenen Daten aus der Tabelle, indem das Datenobjekt abgeschnitten wird, und fügt die importierten Daten ein. Die Tabellendefinition und die Indexdefinitionen werden nicht verändert. Diese Option kann nur verwendet werden, wenn die Tabelle vorhanden ist. Sie ist für Tabellen mit DATALINK-Spalten unzulässig. Wenn diese Option beim Versetzen von Daten zwischen Hierarchien verwendet wird, können nur die Daten für eine vollständige Hierarchie, nicht jedoch für einzelne untergeordnete Tabellen ersetzt werden.

### REPLACE\_CREATE

Löscht, wenn die Tabelle vorhanden ist, alle vorhandenen Daten aus der Tabelle, indem das Datenobjekt abgeschnitten wird, und fügt die importierten Daten ein, ohne die Tabellendefinition oder die Indexdefinitionen zu ändern.

Ist die Tabelle nicht vorhanden, werden die Tabelle und die Indexdefinitionen sowie der Zeileninhalt in der Codepage der Datenbank erstellt. Eine Liste der Einschränkungen finden Sie unter Erneutes Erstellen einer exportierten Tabelle mit IMPORT.

Diese Option kann nur mit IXF-Dateien verwendet werden. Sie ist für Tabellen mit DATALINK-Spalten unzulässig. Wenn diese Option beim Versetzen von Daten zwischen Hierarchien verwendet wird, können nur die Daten für eine vollständige Hierarchie, nicht jedoch für einzelne untergeordnete Tabellen ersetzt werden.

### RESTARTCOUNT $n$

Gibt an, dass eine Importoperation bei Datensatz  $n + 1$  gestartet werden soll. Die ersten  $n$  Datensätze werden übersprungen. Die Funktionsweise dieser Option ist äquivalent zu der Funktionsweise von SKIPCOUNT. RESTARTCOUNT und SKIPCOUNT schließen sich gegenseitig aus.

### ROWCOUNT $n$

Gibt die Anzahl  $n$  für physische Sätze in der Datei an, die importiert (eingefügt oder aktualisiert) werden sollen. Ein Benutzer kann lediglich  $n$  Zeilen aus einer Datei importieren, angefangen bei dem Datensatz, der von der Option SKIPCOUNT oder RESTARTCOUNT ermittelt wurde. Ist die Option SKIPCOUNT oder RESTARTCOUNT nicht angegeben, werden die ersten  $n$  Zeilen importiert. Ist SKIPCOUNT  $m$  oder RESTARTCOUNT  $m$  angegeben, werden die Zeilen  $m+1$  bis  $m+n$  importiert. Werden Com-

pound-INSERT-Operationen verwendet, wird ein benutzerdefinierter Wert von 'n' für 'rowcount' auf das erste ganzzahlige Vielfache des Wertes für den Compound-Zähler aufgerundet.

#### SKIPCOUNT *n*

Gibt an, dass eine Importoperation bei Datensatz  $n + 1$  gestartet werden soll. Die ersten  $n$  Datensätze werden übersprungen. Die Funktionsweise dieser Option ist äquivalent zu der Funktionsweise von RESTARTCOUNT. SKIPCOUNT und RESTARTCOUNT schließen sich gegenseitig aus.

#### STARTING untergeordnete-tabelle

Ein Schlüsselwort nur für Hierarchie, das die Standardreihenfolge anfordert, beginnend mit *untergeordnete-tabelle*. Bei PC/IXF-Dateien ist die Standardreihenfolge die Reihenfolge, die in der Eingabedatei gespeichert ist. Die Standardreihenfolge ist die einzige gültige Reihenfolge für das PC/IXF-Dateiformat.

#### liste-untergeordneter-tabellen

Für typisierte Tabellen mit der Option INSERT oder INSERT\_UPDATE wird eine Liste mit untergeordneten Tabellennamen verwendet, um die untergeordneten Tabellen anzugeben, in die die Daten importiert werden sollen.

#### traversierfolgeliste

Für typisierte Tabellen mit der Option INSERT, INSERT\_UPDATE oder REPLACE wird eine Liste mit untergeordneten Tabellennamen verwendet, um die Traversierfolge der zu importierenden untergeordneten Tabellen in der Hierarchie anzugeben.

#### UNDER untergeordnete-tabelle

Gibt eine übergeordnete Tabelle an, wenn eine oder mehrere untergeordnete Tabellen erstellt werden.

#### WARNINGCOUNT *n*

Stoppt die Importoperation nach  $n$  Warnungen. Definieren Sie diesen Parameter, wenn keine Warnungen erwartet werden, jedoch eine Prüfung darauf, ob die richtige Datei und Tabelle verwendet wird, gewünscht wird. Wird die Importdatei oder die Zieltabelle falsch angegeben, generiert das Dienstprogramm IMPORT eine Warnung für jede Zeile, die es versucht zu importieren, sodass der Import fehlschlägt. Wenn  $n$  Null ist oder diese Option nicht angegeben wird, wird die Importoperation unabhängig von der Anzahl der abgesetzten Warnungen fortgesetzt.

#### Beispiele:

##### Beispiel 1

Das folgende Beispiel zeigt, wie Sie die Daten aus myfile.ixf in die Tabelle STAFF importieren können:

```
db2 import from myfile.ixf of ixf messages msg.txt insert into staff
```

```
SQL3150N  Der H-Satz in der PC/IXF-Datei enthält das Produkt "DB2
01.00", das Datum "19970220" und die Uhrzeit "140848".
```

```
SQL3153N  Der T-Satz in der PC/IXF-Datei hat den Namen "myfile", das
Qualifikationsmerkmal " " und die Quelle " ".
```

```
SQL3109N  Das Dienstprogramm beginnt mit dem Laden von Daten aus
der Datei "myfile".
```

```
SQL3110N  Die Verarbeitung des Dienstprogramms ist abgeschlossen. Es
```

## Befehl IMPORT

wurden "58" Zeile(n) aus der Eingabedatei gelesen.

SQL3221W ...COMMIT WORK beginnt. Zähler für Eingabesätze: "58".

SQL3222W ...COMMIT für alle Änderungen der Datenbank wurde erfolgreich ausgeführt.

SQL3149N "58" Zeile(n) aus der Eingabedatei wurden verarbeitet. "58" Zeile(n) wurden ohne Fehler der Tabelle hinzugefügt. "0" Zeile(n) wurden zurückgewiesen.

### Beispiel 2

Das folgende Beispiel zeigt, wie die Tabelle MOVIE TABLE aus der Eingabedatei del file1 importiert werden kann, die Daten im DEL-Format enthält:

```
db2 import from del file1 of del
modified by dl del |
insert into movietable (actorname, description, url_making_of,
url_movie) datalink specification (dl_url_default_prefix
"http://narang"), (dl_url_replace_prefix "http://bomdel"
dl_url_suffix ".mpeg")
```

#### Anmerkungen:

1. Die Tabelle enthält vier Spalten:

actorname	VARCHAR(n)
description	VARCHAR(m)
url_making_of	DATALINK (mit LINKTYPE URL)
url_movie	DATALINK (mit LINKTYPE URL)

2. Die DATALINK-Daten in der Eingabedatei verwenden den vertikalen Balken (|) als Teilfeldbegrenzer.
3. Wenn ein Spaltenwert für url\_making\_of keine Präfixzeichenfolge aufweist, wird "http://narang" verwendet.
4. Jeder Spaltenwert ungleich NULL für url\_movie erhält "http://bomdel" als Präfix. Vorhandene Werte werden ersetzt.
5. An jeden Spaltenwert ungleich NULL für url\_movie wird ".mpeg" an den Pfad angefügt. Beispiel: Wenn ein Spaltenwert für url\_movie "http://server1/x/y/z" ist, wird er als "http://bomdel/x/y/z.mpeg" gespeichert; ist der Wert "/x/y/z", wird er als "http://bomdel/x/y/z.mpeg" gespeichert.

### Beispiel 3 (Importieren in eine Tabelle mit einer Identitätsspalte)

TABLE1 enthält 4 Spalten:

- C1 VARCHAR(30)
- C2 INT GENERATED BY DEFAULT AS IDENTITY
- C3 DECIMAL(7,2)
- C4 CHAR(1)

TABLE2 entspricht TABLE1, wobei jedoch C2 eine als GENERATED ALWAYS definierte Identitätsspalte ist.

Die Datensätze in DATAFILE1 (DEL-Format) lauten:

```
"Liszt"
"Holst",,187.43, H
"Grieg",100, 66.34, G
"Satie",101, 818.23, I
```

Die Datensätze in DATAFILE2 (DEL-Format) lauten:

```
"Liszt", 74.49, A
"Holst", 0.01, H
"Grieg", 66.34, G
"Satie", 818.23, I
```

Der folgende Befehl generiert Identitätswerte für die Zeilen 1 und 2, da in der Datei DATAFILE1 keine Identitätswerte für diese Zeilen zur Verfügung gestellt werden. Den Zeilen 3 und 4 wird jedoch der benutzerdefinierte Identitätswert 100 bzw. 101 zugeordnet.

```
db2 import from datafile1.del of del replace into table1
```

Um die Datei DATAFILE1 so in die Tabelle TABLE1 zu importieren, dass für alle Zeilen Identitätswerte generiert werden, setzen Sie einen der folgenden Befehle ab:

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

Um die Datei DATAFILE2 so in die Tabelle TABLE1 zu importieren, dass für alle Zeilen Identitätswerte generiert werden, setzen Sie einen der folgenden Befehle ab:

```
db2 import from datafile2.del of del replace into table1 (c1, c3, c4)
db2 import from datafile2.del of del modified by identitymissing
replace into table1
```

Wird die Datei DATAFILE1 in die Tabelle TABLE2 importiert, ohne dass einer der identitätsbezogenen Änderungswerte für den Dateityp verwendet wird, werden die Zeilen 1 und 2 eingefügt, die Zeilen 3 und 4 jedoch zurückgewiesen, da diese Zeilen eigene Werte, die keine Nullwerte sind, liefern und die Identitätsspalte als GENERATED ALWAYS definiert wurde.

### Anmerkungen zur Verwendung:

Achten Sie darauf, alle Tabellenoperationen abzuschließen und alle Sperren freizugeben, bevor Sie eine Importoperation starten. Dies erreichen Sie, indem Sie eine COMMIT-Operation absetzen, nachdem Sie alle geöffneten Cursor mit WITH HOLD geschlossen haben, oder indem Sie eine ROLLBACK-Operation absetzen.

Das Dienstprogramm IMPORT fügt unter Verwendung der SQL-Anweisung INSERT Zeilen zur Zieltabelle hinzu. Das Dienstprogramm setzt für jede Zeile von Daten in der Eingabedatei eine gesonderte Anweisung INSERT ab. Falls eine Anweisung INSERT fehlschlägt, hat dies eine der beiden folgenden Aktionen zur Folge:

- Falls nachfolgende Anweisungen INSERT aller Wahrscheinlichkeit nach erfolgreich verarbeitet werden können, wird eine Warnung in die Nachrichtendatei geschrieben, und die Verarbeitung wird fortgesetzt.
- Schlagen nachfolgende Anweisungen INSERT wahrscheinlich fehl und besteht die Möglichkeit, dass die Datenbank beschädigt wird, wird eine Fehlernachricht in die Nachrichtendatei geschrieben, und die Verarbeitung wird angehalten.

Das Dienstprogramm führt eine automatische COMMIT-Operation aus, nachdem die alten Zeilen während einer REPLACE- oder einer REPLACE\_CREATE-Operation gelöscht wurden. Daher gehen alle alten Daten verloren, falls ein Systemfehler auftritt oder die Anwendung den Datenbankmanager unterbricht, nachdem das Tabellenobjekt abgeschnitten wurde. Vor Verwendung dieser Optionen sollten Sie sich daher vergewissern, dass die alten Daten nicht mehr benötigt werden.

## Befehl IMPORT

Falls das Protokoll während einer CREATE-, REPLACE- oder REPLACE\_CREATE-Operation keine weiteren Einträge mehr aufnehmen kann, führt das Dienstprogramm eine automatische COMMIT-Operation für eingefügte Datensätze aus. Falls ein Systemfehler auftritt oder die Anwendung den Datenbankmanager unterbricht, nachdem eine automatische COMMIT-Operation stattgefunden hat, verbleibt eine Tabelle mit teilweise vorhandenen Daten in der Datenbank. Sie können die gesamte Importoperation wiederholen, indem Sie die Option REPLACE oder REPLACE\_CREATE oder aber INSERT mit dem Parameter RESTARTCOUNT verwenden, der auf die Anzahl der erfolgreich importierten Zeilen gesetzt ist.

In der Standardeinstellung werden bei der Option INSERT oder INSERT\_UPDATE keine automatischen COMMIT-Operationen ausgeführt. Dies gilt jedoch nicht, wenn der Parameter COMMITCOUNT einen anderen Wert als 0 hat. Werden keine automatischen COMMIT-Operationen durchgeführt, resultiert eine volle Protokolldatei in einer ROLLBACK-Operation.

Beim Offlineimport werden keine automatischen COMMIT-Operationen durchgeführt, wenn eine beliebige der folgenden Bedingungen zutrifft:

- Das Ziel ist eine Sicht und keine Tabelle
- Compound-INSERT-Operationen werden verwendet
- Gepufferte INSERT-Operationen werden verwendet

Beim Onlineimport werden automatische COMMIT-Operationen standardmäßig durchgeführt, um sowohl den Speicherbereich für die aktive Protokolldatei als auch die Sperrenliste freizugeben. Automatische COMMIT-Operationen werden nur dann nicht ausgeführt, wenn für COMMITCOUNT der Wert Null angegeben ist.

Immer dann, wenn das Dienstprogramm IMPORT eine COMMIT-Operation ausführt, werden zwei Nachrichten in die Nachrichtendatei geschrieben: Die eine Nachricht gibt die Anzahl der Datensätze an, die festgeschrieben werden sollen, und die andere Nachricht wird nach einer erfolgreichen COMMIT-Operation geschrieben. Beim erneuten Starten der Importoperation nach einem Fehler geben Sie die Anzahl der zu überspringenden Datensätze an, die Sie anhand der letzten erfolgreichen COMMIT-Operation ermitteln können.

Das Dienstprogramm IMPORT akzeptiert Eingabedaten mit geringen Inkompatibilitätsproblemen (Zeichendaten können beispielsweise mit Auffüllen oder Abschneiden importiert werden, und numerische Daten können mit einem anderen numerischen Datentyp importiert werden). Daten mit größeren Inkompatibilitätsproblemen werden jedoch nicht akzeptiert.

Eine REPLACE- oder REPLACE\_CREATE-Operation kann nicht für eine Objekt-tabelle ausgeführt werden, wenn von dieser Tabelle andere Tabellen als sie selbst abhängig sind. Gleiches gilt für eine Objektsicht, wenn andere Tabellen von ihrer Basistabelle abhängig sind (einschließlich sie selbst). Um eine solche Tabelle oder Sicht zu ersetzen, führen Sie Folgendes aus:

1. Löschen Sie alle Fremdschlüssel, in denen die Tabelle als übergeordnete Tabelle angegeben ist.
2. Führen Sie das Dienstprogramm IMPORT aus.
3. Ändern Sie die Tabelle, um die Fremdschlüssel erneut zu erstellen.

Falls bei der erneuten Erstellung der Fremdschlüssel ein Fehler auftritt, ändern Sie die Daten, um die referenzielle Integrität zu erhalten.

Referenzielle Integritätsbedingungen und Fremdschlüsseldefinitionen werden beim Erstellen von Tabellen aus PC/IXF-Dateien nicht beibehalten. (Primärschlüsseldefinitionen *werden* beibehalten, wenn die Daten zuvor mit SELECT \* exportiert wurden.)

Bei Importoperationen in eine ferne Datenbank muss auf dem Server genügend Plattenspeicherplatz für eine Kopie der Eingabedatendatei, der Ausgabenachrichtendatei und die potenzielle Vergrößerung der Datenbank vorhanden sein.

Falls eine Importoperation für eine ferne Datenbank ausgeführt wird und die Ausgabenachrichtendatei sehr lang ist (mehr als 60 KB), kann es sein, dass ab der Mitte der Importoperation Nachrichten in der Nachrichtendatei fehlen, die an den Clientbenutzer zurückgegeben wird. Die ersten 30 KB der Nachrichtendaten und die letzten 30 KB der Nachrichtendaten bleiben immer erhalten.

Das Importieren von PC/IXF-Dateien in eine ferne Datenbank wird erheblich beschleunigt, wenn sich die PC/IXF-Datei nicht auf Disketten, sondern auf einer Festplatte befindet.

Die Datenbanktabelle oder Hierarchie muss vorhanden sein, bevor Daten im ASC-, DEL- oder WSF-Dateiformat importiert werden können. Falls eine Tabelle nicht vorhanden ist, kann sie jedoch durch die Verwendung von IMPORT CREATE oder IMPORT REPLACE\_CREATE beim Importieren von Daten aus einer PC/IXF-Datei erstellt werden. Bei typisierten Tabellen kann durch Verwendung von IMPORT CREATE außerdem die Typhierarchie und die Tabellenhierarchie erstellt werden.

Zum Versetzen von Daten zwischen Datenbanken (einschließlich hierarchischer Daten) sollte der PC/IXF-Import verwendet werden. Falls Zeichendaten, die Zeilentrennzeichen enthalten, in eine ASCII-Datei ohne universelle Zeilenbegrenzer (DEL) exportiert und durch ein Textübertragungsprogramm verarbeitet werden, werden Felder, die die Zeilentrennzeichen enthalten, verkleinert oder erweitert. Der Schritt zum Kopieren der Datei ist nicht erforderlich, wenn sowohl auf die Quelldatenbank als auch auf die Zieldatenbank vom gleichen Client aus zugegriffen werden kann.

Die Daten in ASC- und DEL-Dateien müssen in der Codepage der Clientanwendung vorliegen, die den Import ausführt. Beim Importieren von Daten in unterschiedlichen Codepages empfiehlt sich die Verwendung von PC/IXF-Dateien, die unterschiedliche Codepages zulassen. Falls die PC/IXF-Datei und das Dienstprogramm IMPORT dieselbe Codepage verwenden, erfolgt die Verarbeitung wie bei einer normalen Anwendung. Unterscheiden sich die Codepages und ist die Option FORCEIN angegeben, geht das Dienstprogramm IMPORT davon aus, dass die Daten in der PC/IXF dieselbe Codepage wie die Anwendung verwenden, die den Import ausführt. Dies gilt selbst dann, wenn eine Konvertierungstabelle für die beiden Codepages vorhanden ist. Falls sich die beiden Codepages unterscheiden, die Option FORCEIN nicht angegeben ist und eine Konvertierungstabelle vorhanden ist, werden alle Daten in der PC/IXF-Datei von der Dateicodepage in die Anwendungscodepage konvertiert. Falls sich die beiden Codepages unterscheiden, die Option FORCEIN nicht angegeben ist und keine Konvertierungstabelle vorhanden ist, schlägt die Importoperation fehl. Dies gilt nur für PC/IXF-Dateien auf DB2 UDB-Clients unter dem Betriebssystem AIX.

Bei Tabellenobjekten auf einer 8 KB-Seite, die nahe an der Grenze von 1012 Spalten liegen, kann das Importieren von PC/IXF-Datendateien dazu führen, dass DB2 einen Fehler zurückgibt, weil die maximale Größe einer SQL-Anweisung überschritten wurde. Diese Situation kann nur dann eintreten, wenn die Spalten den

## Befehl IMPORT

Typ CHAR, VARCHAR oder CLOB haben. Die Einschränkung gilt nicht für den Import von DEL- oder ASC-Dateien. Falls die PC/IXF-Dateien zum Erstellen einer neuen Tabelle verwendet werden, kann mit **db2look** alternativ ein Speicherauszug der DDL-Anweisung erstellt werden, mit der die Tabelle erstellt wurde, und anschließend diese Anweisung über den Befehlszeilenprozessor abgesetzt werden.

Mit DB2 Connect können Daten auf DRDA-Servern wie DB2 für OS/390, DB2 für VM und VSE sowie DB2 für OS/400 importiert werden. Nur der PC/IXF-Import (Option INSERT) wird unterstützt. Der Parameter RESTARTCOUNT wird ebenfalls unterstützt, nicht jedoch der Parameter COMMITCOUNT.

Bei Verwendung der Option CREATE für typisierte Tabellen müssen alle in der PC/IXF-Datei definierten untergeordneten Tabellen erstellt werden. Die Definitionen von untergeordneten Tabellen können nicht geändert werden. Werden andere Optionen als CREATE bei typisierten Tabellen verwendet, können Sie über die Traversierfolgeliste die Traversierfolge angeben. Daher muss die Traversierfolgeliste mit der Liste übereinstimmen, die während der Exportoperation verwendet wurde. Beim PC/IXF-Dateiformat muss lediglich der Name der untergeordneten Zieltabelle angegeben und die in der Datei gespeicherte Traversierfolge verwendet werden.

Mit dem Dienstprogramm IMPORT können Sie eine Tabelle wiederherstellen, die zuvor in eine PC/IXF-Datei exportiert wurde. Die Tabelle hat dann denselben Status wie zum Zeitpunkt des Exports.

Das Importieren von Daten in eine Systemtabelle, eine deklarierte temporäre Tabelle oder eine Übersichtstabelle ist nicht möglich.

Mit dem Dienstprogramm IMPORT können keine Sichten erstellt werden.

Für das Betriebssystem Windows gilt:

- Das Importieren von logisch geteilten PC/IXF-Dateien wird nicht unterstützt.
- Das Importieren von PC/IXF- oder WSF-Dateien mit einem fehlerhaften Format wird nicht unterstützt.

### Hinweise zu DB2 Data Links Manager:

Bitte führen Sie Folgendes aus, bevor Sie das DB2-Dienstprogramm IMPORT verwenden:

1. Kopieren Sie die Dateien, auf die verwiesen wird, auf die entsprechenden Data Links-Server. Mit dem Dienstprogramm **dlfm\_import** können Dateien aus einem Archiv extrahiert werden, das vom Dienstprogramm **dlfm\_export** erstellt wurde.
2. Registrieren Sie die erforderlichen Präfixnamen in den DB2 Data Links Managern. Sie müssen eventuell auch andere Verwaltungsaufgaben ausführen, wie ggfs. das Registrieren der Datenbank.
3. Aktualisieren Sie die Informationen zu den Data Links-Servern in den URL-Adressen (der DATALINK-Spalten) aus den exportierten Daten für die SQL-Tabelle (falls erforderlich). (Wenn die Data Links-Server der ursprünglichen Konfiguration mit den Servern an der Zielposition übereinstimmen, müssen die Namen der Data Links-Server nicht aktualisiert werden.)
4. Definieren Sie die Data Links-Server in der Zielkonfiguration in der Konfigurationsdatei für DB2 Data Links Manager.

Wenn das Dienstprogramm IMPORT für die Zieldatenbank ausgeführt wird, werden Dateien, auf die in DATALINK-Spalten verwiesen wird, mit den entsprechenden Data Links-Servern verknüpft.

Während der Einfügeoperation werden bei der Verarbeitung von DATALINK-Spalten die Dateien auf den entsprechenden Data Links-Servern gemäß den Spaltenspezifikationen der Zieldatenbank verknüpft.

**Zugehörige Konzepte:**

- „IMPORT - Übersicht“ auf Seite 27
- „Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von IMPORT“ auf Seite 29

**Zugehörige Tasks:**

- „Verwenden von IMPORT“ auf Seite 29

**Zugehörige Referenzen:**

- „Import-API "db2Import"“ auf Seite 53
- „db2look - DB2 Statistics and DDL Extraction Tool Command“ in *Command Reference*
- „CLP-Beispiele für IMPORT-Sitzungen“ auf Seite 76
- „Befehl LOAD“ auf Seite 112
- „Änderungswerte für Dateityp für IMPORT“ auf Seite 66
- „Einschränkungen für Begrenzer beim Versetzen von Daten“ auf Seite 247

---

## Import-API "db2Import"

Fügt Daten aus einer externen Datei mit einem unterstützten Dateiformat in eine Tabelle, eine Hierarchie oder eine Sicht ein. Die Verwendung von LOAD ist zwar eine schnellere Alternative, aber das Dienstprogramm LOAD unterstützt das Laden von Daten auf Hierarchieebene nicht.

**Berechtigung:**

- Für die Verwendung von IMPORT mit der Option INSERT ist eine der folgenden Berechtigungen erforderlich:
  - *sysadm*
  - *dbadm*
  - Zugriffsrecht CONTROL für jede betroffene Tabelle oder Sicht
  - Zugriffsrecht INSERT und SELECT für jede betroffene Tabelle oder Sicht
- Für die Verwendung von IMPORT für eine vorhandene Tabelle mit der Option INSERT\_UPDATE ist eine der folgenden Berechtigungen erforderlich:
  - *sysadm*
  - *dbadm*
  - Zugriffsrecht CONTROL für die Tabelle bzw. Sicht
  - Zugriffsrecht INSERT, SELECT, UPDATE und DELETE für jede betroffene Tabelle oder Sicht
- Für die Verwendung von IMPORT für eine bestehende Tabelle mit der Option REPLACE oder REPLACE\_CREATE ist eine der folgenden Berechtigungen erforderlich:
  - *sysadm*

## Import-API "db2Import"

- *dbadm*
- Zugriffsrecht CONTROL für die Tabelle bzw. Sicht
- Zugriffsrecht INSERT, SELECT und DELETE für die Tabelle oder Sicht
- Für die Verwendung von IMPORT für eine neue Tabelle mit der Option CREATE oder REPLACE\_CREATE ist eine der folgenden Berechtigungen erforderlich:
  - *sysadm*
  - *dbadm*
  - Berechtigung CREATETAB für die Datenbank und Zugriffsrecht USE für den Tabellenbereich sowie eine der beiden Folgenden:
    - Berechtigung IMPLICIT\_SCHEMA für die Datenbank, falls der Name des impliziten oder expliziten Schemas der Tabelle nicht vorhanden ist
    - Zugriffsrecht CREATIN für das Schema, wenn der Schemaname der Tabelle auf ein vorhandenes Schema verweist
- Für die Verwendung von IMPORT für eine nicht vorhandene Tabelle oder Hierarchie mit der Option CREATE oder REPLACE\_CREATE ist eine der folgenden Berechtigungen erforderlich:
  - *sysadm*
  - *dbadm*
  - Berechtigung CREATETAB für die Datenbank sowie eine der folgenden Berechtigungen:
    - Berechtigung IMPLICIT\_SCHEMA für die Datenbank, falls der Name des Schemas der Tabelle nicht vorhanden ist
    - Zugriffsrecht CREATEIN für das Schema, falls das Schema der Tabelle vorhanden ist
    - Zugriffsrecht CONTROL für alle untergeordneten Tabellen in der Hierarchie, falls die Option REPLACE\_CREATE für die gesamte Hierarchie verwendet wird
- Für die Verwendung von IMPORT für eine vorhandene Hierarchie mit der Option REPLACE ist eine der folgenden Berechtigungen erforderlich:
  - *sysadm*
  - *dbadm*
  - Zugriffsrecht CONTROL für alle untergeordneten Tabellen in der Hierarchie

### Erforderliche Verbindung:

Datenbank. Wenn das implizite Herstellen von Verbindungen aktiviert ist, wird eine Verbindung zur Standarddatenbank hergestellt.

### API-Include-Datei:

*db2ApiDf.h*

### C-API-Syntax:

```
/* db2Import - API */
SQL_API_RC SQL_API_FN
db2Import (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

/* db2Import parameter structure */
typedef SQL_STRUCTURE db2ImportStruct
{
```

```

char                                     *pDataFileName;
struct sqlu_media_list *piLobPathList;
struct sqlcol *pDataDescriptor;
struct sqlchar *piActionString;
char *piFileType;
struct sqlchar *piFileTypeMod;
char                                     *pMsgFileName;
db2int16 iCallerAction;
struct db2ImportIn                       *piImportInfoIn;
struct db2ImportOut                     *poImportInfoOut;
db2int32 *piNullIndicators;
} db2ImportStruct;

/* Import input structure */
typedef SQL_STRUCTURE db2ImportIn
{
    db2UInt64                               iRowCount;
    db2UInt64                               iRestartcount;
    db2UInt64                               iSkipcount;
    db2int32                                 *piCommitcount;
    db2UInt32                               iWarningcount;
    db2UInt16                               iNoTimeout;
    db2UInt16                               iAccessLevel;
} db2ImportIn;

/* Import output structure */
typedef SQL_STRUCTURE db2ImportOut
{
    db2UInt64                               oRowsRead;
    db2UInt64                               oRowsSkipped;
    db2UInt64                               oRowsInserted;
    db2UInt64                               oRowsUpdated;
    db2UInt64                               oRowsRejected;
    db2UInt64                               oRowsCommitted;
} db2ImportOut;
    
```

### Generische API-Syntax:

```

/* db2gImport - Generic API */
SQL_API_RC SQL_API_FN
db2gImport (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

/* db2gImport parameter structure */
typedef SQL_STRUCTURE db2gImportStruct
{
    char                                     *pDataFileName;
    struct sqlu_media_list *piLobPathList;
    struct sqlcol *pDataDescriptor;
    struct sqlchar *piActionString;
    char *piFileType;
    struct sqlchar *piFileTypeMod;
    char                                     *pMsgFileName;
    db2int16 iCallerAction;
    struct db2gImportIn                       *piImportInfoIn;
    struct db2gImportOut                     *poImportInfoOut;
    db2int32 *piNullIndicators;
    db2UInt16                               iDataFileNameLen;
    db2UInt16                               iFileTypeLen;
    db2UInt16                               iMsgFileNameLen;
} db2gImportStruct;

/* Generic Import input structure */
typedef SQL_STRUCTURE db2gImportIn
{
    
```

## Import-API "db2Import"

```
|
|         db2UInt64           iRowCount;
|         db2UInt64           iRestartcount;
|         db2UInt64           iSkipcount;
|         db2int32            *piCommitcount;
|         db2UInt32           iWarningcount;
|         db2UInt16           iNoTimeout;
|         db2UInt16           iAccessLevel;
|     } db2gImportIn;
|
|     /* Generic Import output structure */
|     typedef SQL_STRUCTURE db2gImportOut
|     {
|         db2UInt64           oRowsRead;
|         db2UInt64           oRowsSkipped;
|         db2UInt64           oRowsInserted;
|         db2UInt64           oRowsUpdated;
|         db2UInt64           oRowsRejected;
|         db2UInt64           oRowsCommitted;
|     } db2gImportOut;
```

### API-Parameter:

#### versionNumber

Eingabe. Gibt die Version und den Releasestand der Struktur an, die als zweiter Parameter, *pParmStruct*, übergeben wird.

#### pParmStruct

Ein-/Ausgabe. Ein Zeiger auf die *db2ImportStruct*-Struktur.

#### pSqlca

Ausgabe. Ein Zeiger auf die *sqlca*-Struktur.

#### piDataFileName

Eingabe. Eine Zeichenfolge, die den Pfad und den Namen einer externen Eingabedatei enthält, aus der die Daten importiert werden sollen.

#### piLobPathList

Eingabe. Eine *sqlu\_media\_list* unter Verwendung von *media\_type* `SQLU_LOCAL_MEDIA` und der *sqlu\_media\_entry*-Struktur, die Pfade auf dem Client auflistet, in denen sich LOB-Dateien befinden.

#### piDataDescriptor

Eingabe. Zeiger auf eine *sqldcol*-Struktur, die Informationen zu den Spalten enthält, die zum Import aus der externen Datei ausgewählt werden. Der Wert des Feldes *dcolmeth* bestimmt, wie die restlichen in diesem Parameter bereitgestellten Informationen vom Dienstprogramm `IMPORT` ausgewertet werden. Gültige Werte für diesen Parameter sind:

##### SQL\_METH\_N

Namen. Die Auswahl von Spalten in der externen Eingabedatei erfolgt nach Spaltenname.

##### SQL\_METH\_P

Spaltenpositionen. Die Auswahl von Spalten in der externen Eingabedatei erfolgt durch Angabe der Nummer der zu ladenden Spalten.

##### SQL\_METH\_L

Anfangs- und Endposition von Spalten. Die Auswahl von Spalten in der externen Eingabedatei erfolgt durch Angabe der Nummer der ersten und der letzten Position, aus der Daten geladen werden sollen. Der Datenbankmanager weist einen Importaufruf mit einem Positionspaar zurück, das aufgrund einer der folgenden Bedingungen ungültig ist:

- Entweder die Anfangs- oder die Endposition liegt nicht im Bereich von 1 bis zur größten 2 Byte großen ganzen Zahl mit Vorzeichen.
- Die Nummer der Endposition ist kleiner als die Nummer der Anfangsposition.
- Die Eingabespaltenbreite, die vom Positionspaar definiert wird, ist zum Typ und der Länge der Zielspalte nicht kompatibel.

Ein Positionspaar, bei dem beide Positionsangaben gleich Null sind, gibt an, dass eine Spalte, die Nullwerte annehmen kann, mit Nullen gefüllt werden soll.

### SQL\_METH\_D

Standardeinstellung. Wenn *piDataDescriptor* NULL oder auf SQL\_METH\_D gesetzt ist, erfolgt eine Standardauswahl von Spalten aus der externen Eingabedatei. In diesem Fall werden die Anzahl der Spalten und die Spaltenspezifikationsmatrix ignoriert. Bei DEL-, IXF- oder WSF-Dateien werden die ersten *n* Spalten mit Daten in der externen Eingabedatei in der bestehenden Reihenfolge verwendet, wobei *n* für die Anzahl der Datenbankspalten steht, in die die Daten importiert werden sollen.

### piActionString

Eingabe. Ein Zeiger auf eine *sqlchar*-Struktur, die ein 2-Byte-Langfeld enthält, gefolgt von einer Matrix mit Zeichen, die die Spalten angeben, in die die Daten importiert werden sollen.

Der Zeichenbereich besitzt folgendes Format:

```
{INSERT | INSERT_UPDATE | REPLACE | CREATE | REPLACE_CREATE}
INTO {tname[(tspaltenliste)] |
[ALL TABLES | (tname[(tspaltenliste))],
tname[(tspaltenliste))]}
[IN] HIERARCHY {STARTING tname | (tname[, tname])}
[UNDER untergeordnete-tabelle | AS ROOT TABLE]
[DATA LINK SPECIFICATION datalink-spez]
```

### INSERT

Fügt die importierten Daten in die Tabelle ein, ohne die vorhandenen Tabellendaten zu ändern.

### INSERT\_UPDATE

Fügt die importierten Zeilen hinzu, wenn sich ihre Primärschlüsselwerte nicht in der Tabelle befinden, und verwendet sie zur Aktualisierung, wenn ihre Primärschlüsselwerte gefunden werden. Diese Option ist nur zulässig, wenn die Zieltabelle einen Primärschlüssel enthält und die angegebene (oder implizierte) Liste mit Zielspalten, die importiert werden, alle Spalten für den Primärschlüssel enthält. Diese Option kann nicht auf Sichten angewendet werden.

### REPLACE

Löscht alle vorhandenen Daten aus der Tabelle, indem das Tabellenobjekt abgeschnitten wird, und fügt die importierten Daten ein. Die Tabellendefinition und die Indexdefinitionen werden nicht verändert. (Indizes werden gelöscht und ersetzt, wenn *indexixf* in *FileTypeMod* und *FileType* SQL\_IXF ist.) Ist die Tabelle noch nicht definiert, wird ein Fehler zurückgegeben.

**Achtung:** Wenn ein Fehler auftritt, nachdem die vorhandenen Daten gelöscht wurden, gehen die Daten verloren.

### CREATE

Erstellt die Tabellendefinition und den Zeileninhalt unter Verwendung der in der PC/IXF-Datei angegebenen Informationen, wenn die angegebene Tabelle nicht definiert ist. Wenn die Datei vorher von DB2 exportiert wurde, werden auch Indizes erstellt. Wenn die angegebene Tabelle bereits definiert ist, wird ein Fehler zurückgegeben. Diese Option ist nur für das PC/IXF-Dateiformat gültig.

### REPLACE\_CREATE

Ersetzt den Tabelleninhalt durch die PC/IXF-Zeilendaten aus der PC/IXF-Datei, wenn die angegebene Tabelle definiert ist. Wenn die Tabelle noch nicht definiert ist, werden die Tabellendefinition und der Zeileninhalt unter Verwendung der Daten in der angegebenen PC/IXF-Datei erstellt. Wenn die PC/IXF-Datei vorher von DB2 exportiert wurde, werden auch Indizes erstellt. Diese Option ist nur für das PC/IXF-Dateiformat gültig.

**Achtung:** Wenn ein Fehler auftritt, nachdem die vorhandenen Daten gelöscht wurden, gehen die Daten verloren.

*tname* Der Name der Tabelle, typisierten Tabelle, Sicht oder Objektsicht, in die die Daten eingefügt werden sollen. Es kann ein Aliasname für REPLACE, INSERT\_UPDATE oder INSERT angegeben werden (außer im Falle eines Servers mit einem älteren Versionsstand, in dem ein qualifizierter oder nicht qualifizierter Name angegeben werden muss). Erfolgt dies in einer Sicht, darf es keine Sicht nur mit Lesezugriff sein.

#### *tspaltenliste*

Eine Liste der Tabellen- oder Sichtspaltennamen, in die die Daten eingefügt werden sollen. Die Spaltennamen müssen durch Kommas getrennt werden. Wenn keine Spaltennamen angegeben werden, werden die in den Anweisungen CREATE TABLE oder ALTER TABLE definierten Spaltennamen verwendet. Wenn keine Spaltenliste für typisierte Tabellen angegeben wird, werden Daten in alle Spalten innerhalb einer jeden untergeordneten Tabelle eingefügt.

#### *untergeordnete-tabelle*

Gibt eine übergeordnete Tabelle an, wenn eine oder mehrere untergeordnete Tabellen mit der Option CREATE erstellt werden.

### ALL TABLES

Ein implizites Schlüsselwort nur für die Hierarchie. Beim Importieren einer Hierarchie werden standardmäßig alle Tabellen importiert, die in *traversierfolgeliste* angegeben sind.

### HIERARCHY

Gibt an, dass hierarchische Daten importiert werden sollen.

### STARTING

Schlüsselwort nur für die Hierarchie. Gibt an, dass ausgehend vom Namen einer bestimmten untergeordneten Tabelle die Standardreihenfolge verwendet werden soll.

### UNDER

Schlüsselwort nur für die Hierarchie und CREATE. Gibt an, dass die neue Hierarchie, untergeordnete Hierarchie oder untergeordnete Tabelle unter einer gegebenen untergeordneten Tabelle erstellt werden soll.

**AS ROOT TABLE**

Schlüsselwort nur für die Hierarchie und CREATE. Gibt an, dass die neue Hierarchie, untergeordnete Hierarchie oder untergeordnete Tabelle als eigenständige Hierarchie erstellt werden soll.

**DATALINK SPECIFICATION** *datalink-spez*

Gibt Parameter an, die sich auf DB2 Data Links Manager beziehen. Diese Parameter können mit der gleichen Syntax wie beim Befehl IMPORT angegeben werden.

Die Parameter *tname* und *tspaltenliste* entsprechen den Listen *tabellenname* und *spaltenname* von SQL-Anweisungen INSERT, und es gelten dafür die gleichen Einschränkungen.

Die Spalten in *tspaltenliste* und die externen Spalten (entweder angegeben oder impliziert) werden anhand ihrer Position in der Liste oder Struktur abgeglichen (Daten aus der ersten Spalte, die in der *squdcol*-Struktur angegeben ist, werden in das Tabellen- oder Sichtfeld eingefügt, das dem ersten Element von *tspaltenliste* entspricht).

Wenn eine ungleiche Anzahl von Spalten angegeben wird, wird die kleinere der angegebenen Anzahl von Spalten verarbeitet. Das kann zu einem Fehler führen (weil für einige Tabellenfelder, die keine Nullwerte annehmen dürfen, keine Werte vorhanden sind) oder eine Informationsnachricht (weil einige externe Dateispalten ignoriert werden) auslösen.

**piFileType**

Eingabe. Eine Zeichenfolge, die das Format der Daten in der externen Datei angibt. Folgende Formate für externe Dateien werden unterstützt:

**SQL\_ASC**

ASCII-Format mit universellen Zeilenbegrenzern.

**SQL\_DEL**

ASCII-Format ohne universelle Zeilenbegrenzer für den Datenaustausch mit dBase, BASIC und den IBM Personal Decision Series-Programmen und vielen anderen Datenbank- und Dateimanagern.

**SQL\_IXF**

PC-Version des Integrated Exchange Format. Dies ist das bevorzugte Format für den Export von Daten aus einer Tabelle, so dass sie später in dieselbe Tabelle oder eine andere Datenbankmanagertabelle importiert werden können.

**SQL\_WSF**

Arbeitsblattformate zum Datenaustausch mit den Programmen Lotus Symphony und 1-2-3.

**piFileTypeMod**

Eingabe. Ein Zeiger auf eine Struktur, die ein 2-Byte-Langfeld enthält, gefolgt von einer Matrix mit Zeichen, die eine oder mehrere Verarbeitungsoptionen angeben. Wenn dieser Zeiger NULL ist oder die Struktur, auf die gezeigt wird, null Zeichen enthält, wird diese Aktion als Auswahl einer Standardspezifikation gewertet.

Nicht alle Optionen können mit allen unterstützten Dateitypen verwendet werden. Weitere Informationen hierzu finden Sie in File type modifiers for import.

**piMsgFileName**

Eingabe. Eine Zeichenfolge, die den Bestimmungsort für vom Dienstprogramm zurückgegebene Fehlermeldungen, informative Nachrichten

und Warnungen enthält. Dies kann der Pfad und der Name einer Betriebssystemdatei oder einer Standardeinheit sein. Wenn die Datei bereits vorhanden ist, wird an sie angefügt. Wenn sie nicht vorhanden ist, wird eine neue Datei erstellt.

### **iCallerAction**

Eingabe. Eine vom aufrufenden Programm angeforderte Aktion. Gültige Werte:

#### **SQLU\_INITIAL**

Erstaufruf. Dieser Wert muss beim ersten Aufruf der API verwendet werden.

Wenn der Erstaufruf oder ein nachfolgender Aufruf zurückgegeben wird und erfordert, dass die aufrufende Anwendung eine Aktion ausführt, bevor die angeforderte Importoperation ausgeführt wird, muss die Aktion des aufrufenden Programms auf eine der folgenden Optionen gesetzt werden:

#### **SQLU\_CONTINUE**

Verarbeitung fortsetzen. Dieser Wert kann nur bei nachfolgenden Aufrufen der API verwendet werden, nachdem der Erstaufruf zurückgegeben wurde und das Dienstprogramm eine Benutzereingabe anfordert (z. B. um auf eine Bandendebedingung zu reagieren). Er gibt an, dass die vom Dienstprogramm angeforderte Benutzeraktion beendet wurde und das Dienstprogramm die Verarbeitung der Erstanforderung fortsetzen kann.

#### **SQLU\_TERMINATE**

Verarbeitung beenden. Dieser Wert kann nur bei nachfolgenden Aufrufen der API verwendet werden, nachdem der Erstaufruf zurückgegeben wurde und das Dienstprogramm eine Benutzereingabe anfordert (z. B. um auf eine Bandendebedingung zu reagieren). Er gibt an, dass die vom Dienstprogramm angeforderte Benutzeraktion nicht ausgeführt wurde und das Dienstprogramm die Verarbeitung der Erstanforderung beenden soll.

### **piImportInfoIn**

Eingabe. Ein Zeiger auf die *db2ImportIn*-Struktur.

### **poImportInfoOut**

Ausgabe. Ein Zeiger auf die *db2ImportOut*-Struktur.

### **piNullIndicators**

Eingabe. Nur für ASC-Dateien. Eine Matrix von ganzen Zahlen, die angeben, ob die Spaltendaten Nullwerte annehmen dürfen oder nicht. Die Anzahl der Elemente in dieser Matrix muss mit der Anzahl von Spalten in der Eingabedatei übereinstimmen. Es besteht eine 1-zu-1-Entsprechung zwischen den Elementen dieser Matrix und den Spalten, die aus der Datendatei importiert werden. Daher muss die Anzahl der Elemente mit dem Feld *dcolnum* des Parameters *piDataDescriptor* übereinstimmen. Jedes Element der Matrix enthält eine Zahl, die eine Spalte in der Datendatei angibt, die als Nullanzeigerfeld verwendet werden soll, oder eine Null, um anzugeben, dass die Tabellenspalte keine Nullwerte enthalten darf. Wenn das Element nicht Null ist, muss die angegebene Spalte in der Datendatei Y oder N enthalten. Y gibt an, dass die Tabellenspaltendaten NULL sind, N gibt an, dass die Tabellenspaltendaten nicht NULL sind.

### **iRowcount**

Eingabe. Die Anzahl der physischen Sätze, die geladen werden sollen.

Ermöglicht es einem Benutzer, nur die ersten mit *iRowcount* angegebenen Zeilen in einer Datei zu laden. Wenn *iRowcount* Null ist, wird beim Importieren versucht, alle Zeilen aus der Datei zu verarbeiten.

### **iSkipcount**

Eingabe. Die Anzahl der Datensätze, die übersprungen werden, bevor mit der Einfügung oder Aktualisierung von Datensätzen begonnen wird. Funktionsweise äquivalent zu *iRestartcount*.

### **piCommitcount**

Eingabe. Die Anzahl der Datensätze, die importiert werden sollen, bevor sie in der Datenbank festgeschrieben werden. Jedes Mal, wenn *piCommitcount*-Datensätze importiert werden, wird eine COMMIT-Operation ausgeführt. Ein Nullwert (NULL) gibt den Standardwert der COMMIT-Zählung an, der für den Offlineimport Null und für den Onlineimport AUTOMATIC beträgt. Die COMMIT-Zählung AUTOMATIC wird durch Übergabe im Wert DB2IMPORT\_COMMIT\_AUTO angegeben.

### **iWarningcount**

Eingabe. Stoppt die Importoperation nach *iWarningcount*-Warnungen. Definieren Sie diesen Parameter, wenn keine Warnungen erwartet werden, jedoch eine Prüfung darauf, ob die richtige Datei und Tabelle verwendet wird, gewünscht wird. Wird die Importdatei oder die Zieltabelle falsch angegeben, generiert das Dienstprogramm IMPORT eine Warnung für jede Zeile, die es versucht zu importieren, sodass der Import fehlschlägt. Wenn *iWarningcount* Null ist oder diese Option nicht angegeben ist, wird die Importoperation unabhängig von der Anzahl der abgesetzten Warnungen fortgesetzt.

### **iNoTimeout**

Eingabe. Gibt an, dass es für das Dienstprogramm IMPORT beim Warten auf Sperren kein Zeitlimit gibt. Diese Option hat Vorrang vor dem Datenbankkonfigurationsparameter *locktimeout*. Andere Anwendungen sind nicht betroffen. Gültige Werte:

#### **DB2IMPORT\_LOCKTIMEOUT**

Gibt an, dass der Wert des Konfigurationsparameters *locktimeout* beachtet wird.

#### **DB2IMPORT\_NO\_LOCKTIMEOUT**

Gibt an, dass kein Zeitlimit vorhanden ist.

### **iAccessLevel**

Eingabe. Gibt die Zugriffsebene an. Gültige Werte:

#### **SQLU\_ALLOW\_NO\_ACCESS**

Gibt an, dass das Dienstprogramm IMPORT die Tabelle exklusiv sperrt.

#### **SQLU\_ALLOW\_WRITE\_ACCESS**

Gibt an, dass während des Importvorgangs weiterhin Lese- und Schreibzugriff auf die Daten in der Tabelle bestehen soll.

### **oRowsRead**

Ausgabe. Anzahl der Datensätze, die bei der Importoperation aus der Datei gelesen wurden.

### **oRowsSkipped**

Ausgabe. Anzahl der Datensätze, die vor Beginn der Einfügung oder Aktualisierung übersprungen wurden.

## Import-API "db2Import"

### **oRowsInserted**

Ausgabe. Anzahl der Zeilen, die in die Zieltabelle eingefügt wurden.

### **oRowsUpdated**

Ausgabe. Anzahl der Zeilen in der Zieltabelle, die mit Daten aus den importierten Datensätzen aktualisiert wurden (Datensätzen, deren Primärschlüsselwert bereits in der Tabelle vorhanden ist).

### **oRowsRejected**

Ausgabe. Anzahl der Datensätze, die nicht importiert werden konnten.

### **oRowsCommitted**

Ausgabe. Anzahl der erfolgreich importierten und in der Datenbank festgeschriebenen Datensätze.

### **Anmerkungen zur Verwendung:**

Achten Sie darauf, alle Tabellenoperationen abzuschließen und alle Sperren freizugeben, bevor Sie eine Importoperation starten. Dies erreichen Sie, indem Sie eine COMMIT-Operation absetzen, nachdem Sie alle geöffneten Cursor mit WITH HOLD geschlossen haben, oder indem Sie eine ROLLBACK-Operation absetzen.

Das Dienstprogramm IMPORT fügt unter Verwendung der SQL-Anweisung INSERT Zeilen zur Zieltabelle hinzu. Das Dienstprogramm setzt für jede Zeile von Daten in der Eingabedatei eine gesonderte Anweisung INSERT ab. Falls eine Anweisung INSERT fehlschlägt, hat dies eine der beiden folgenden Aktionen zur Folge:

- Falls nachfolgende Anweisungen INSERT aller Wahrscheinlichkeit nach erfolgreich verarbeitet werden können, wird eine Warnung in die Nachrichtendatei geschrieben, und die Verarbeitung wird fortgesetzt.
- Schlagen nachfolgende Anweisungen INSERT wahrscheinlich fehl und besteht die Möglichkeit, dass die Datenbank beschädigt wird, wird eine Fehlernachricht in die Nachrichtendatei geschrieben, und die Verarbeitung wird angehalten.

Das Dienstprogramm führt eine automatische COMMIT-Operation aus, nachdem die alten Zeilen während einer REPLACE- oder einer REPLACE\_CREATE-Operation gelöscht wurden. Daher gehen alle alten Daten verloren, falls ein Systemfehler auftritt oder die Anwendung den Datenbankmanager unterbricht, nachdem das Tabellenobjekt abgeschnitten wurde. Vor Verwendung dieser Optionen sollten Sie sich daher vergewissern, dass die alten Daten nicht mehr benötigt werden.

Falls das Protokoll während einer CREATE-, REPLACE- oder REPLACE\_CREATE-Operation keine weiteren Einträge mehr aufnehmen kann, führt das Dienstprogramm eine automatische COMMIT-Operation für eingefügte Datensätze aus. Falls ein Systemfehler auftritt oder die Anwendung den Datenbankmanager unterbricht, nachdem eine automatische COMMIT-Operation stattgefunden hat, verbleibt eine Tabelle mit teilweise vorhandenen Daten in der Datenbank. Sie können die gesamte Importoperation wiederholen, indem Sie die Option REPLACE oder REPLACE\_CREATE verwenden, oder aber INSERT mit dem Parameter *iRestart-count* verwenden, der auf die Anzahl der erfolgreich importierten Zeilen gesetzt ist.

In der Standardeinstellung werden bei der Option INSERT oder INSERT\_UPDATE keine automatischen COMMIT-Operationen ausgeführt. Dies gilt jedoch nicht, wenn der Parameter *\*piCommitcount* einen anderen Wert als Null hat. Falls das Protokoll keine weiteren Einträge mehr aufnehmen kann, wird eine ROLLBACK-Operation ausgeführt.

Immer dann, wenn das Dienstprogramm IMPORT eine COMMIT-Operation ausführt, werden zwei Nachrichten in die Nachrichtendatei geschrieben: Die eine Nachricht gibt die Anzahl der Datensätze an, die festgeschrieben werden sollen, und die andere Nachricht wird nach einer erfolgreichen COMMIT-Operation geschrieben. Beim erneuten Starten der Importoperation nach einem Fehler geben Sie die Anzahl der zu überspringenden Datensätze an, die Sie anhand der letzten erfolgreichen COMMIT-Operation ermitteln können.

Das Dienstprogramm IMPORT akzeptiert Eingabedaten mit geringen Inkompatibilitätsproblemen (Zeichendaten können beispielsweise mit Auffüllen oder Abschneiden importiert werden, und numerische Daten können mit einem anderen numerischen Datentyp importiert werden). Daten mit größeren Inkompatibilitätsproblemen werden jedoch nicht akzeptiert.

Eine REPLACE- oder REPLACE\_CREATE-Operation kann nicht für eine Objektabelle ausgeführt werden, wenn von dieser Tabelle andere Tabellen als sie selbst abhängig sind. Gleiches gilt für eine Objektsicht, wenn andere Tabellen von ihrer Basistabelle abhängig sind (einschließlich sie selbst). Um eine solche Tabelle oder Sicht zu ersetzen, führen Sie Folgendes aus:

1. Löschen Sie alle Fremdschlüssel, in denen die Tabelle als übergeordnete Tabelle angegeben ist.
2. Führen Sie das Dienstprogramm IMPORT aus.
3. Ändern Sie die Tabelle, um die Fremdschlüssel erneut zu erstellen.

Falls bei der erneuten Erstellung der Fremdschlüssel ein Fehler auftritt, ändern Sie die Daten, um die referenzielle Integrität zu erhalten.

Referenzielle Integritätsbedingungen und Fremdschlüsseldefinitionen werden beim Erstellen von Tabellen aus PC/IXF-Dateien nicht beibehalten. (Primärschlüsseldefinitionen *werden* beibehalten, wenn die Daten zuvor mit SELECT \* exportiert wurden.)

Bei Importoperationen in eine ferne Datenbank muss auf dem Server genügend Plattenspeicherplatz für eine Kopie der Eingabedatendatei, der Ausgabenachrichtendatei und die potenzielle Vergrößerung der Datenbank vorhanden sein.

Falls eine Importoperation für eine ferne Datenbank ausgeführt wird und die Ausgabenachrichtendatei sehr lang ist (mehr als 60 KB), kann es sein, dass ab der Mitte der Importoperation Nachrichten in der Nachrichtendatei fehlen, die an den Clientbenutzer zurückgegeben wird. Die ersten 30 KB der Nachrichtendaten und die letzten 30 KB der Nachrichtendaten bleiben immer erhalten.

Das Importieren von PC/IXF-Dateien in eine ferne Datenbank wird erheblich beschleunigt, wenn sich die PC/IXF-Datei nicht auf Disketten, sondern auf einer Festplatte befindet. Wenn für *piDataDescriptor* andere Werte als die Standardwerte verwendet werden oder wenn im Parameter *piActionString* eine explizite Liste von Tabellenspalten angegeben wird, verlangsamt dies den Import in eine ferne Datenbank.

Die Datenbanktabelle oder Hierarchie muss vorhanden sein, bevor Daten im ASC-, DEL- oder WSF-Dateiformat importiert werden können. Falls eine Tabelle nicht vorhanden ist, kann sie jedoch durch die Verwendung von IMPORT CREATE oder IMPORT REPLACE\_CREATE beim Importieren von Daten aus einer PC/IXF-Datei erstellt werden. Bei typisierten Tabellen kann durch Verwendung von IMPORT CREATE außerdem die Typhierarchie und die Tabellenhierarchie erstellt werden.

## Import-API "db2Import"

Zum Versetzen von Daten zwischen Datenbanken (einschließlich hierarchischer Daten) sollte der PC/IXF-Import verwendet werden. Falls Zeichendaten, die Zeilentrennzeichen enthalten, in eine ASCII-Datei ohne universelle Zeilenbegrenzer (DEL) exportiert und durch ein Textübertragungsprogramm verarbeitet werden, werden Felder, die die Zeilentrennzeichen enthalten, verkleinert oder erweitert.

Die Daten in ASC- und DEL-Dateien müssen in der Codepage der Clientanwendung vorliegen, die den Import ausführt. Beim Importieren von Daten in unterschiedlichen Codepages empfiehlt sich die Verwendung von PC/IXF-Dateien, die unterschiedliche Codepages zulassen. Falls die PC/IXF-Datei und das Dienstprogramm IMPORT dieselbe Codepage verwenden, erfolgt die Verarbeitung wie bei einer normalen Anwendung. Unterscheiden sich die Codepages und ist die Option FORCEIN angegeben, geht das Dienstprogramm IMPORT davon aus, dass die Daten in der PC/IXF dieselbe Codepage wie die Anwendung verwenden, die den Import ausführt. Dies gilt selbst dann, wenn eine Konvertierungstabelle für die beiden Codepages vorhanden ist. Falls sich die beiden Codepages unterscheiden, die Option FORCEIN nicht angegeben ist und eine Konvertierungstabelle vorhanden ist, werden alle Daten in der PC/IXF-Datei von der Dateicodepage in die Anwendungscodepage konvertiert. Falls sich die beiden Codepages unterscheiden, die Option FORCEIN nicht angegeben ist und keine Konvertierungstabelle vorhanden ist, schlägt die Importoperation fehl. Dies gilt nur für PC/IXF-Dateien auf Clients mit DB2 für AIX.

Bei Tabellenobjekten auf einer 8 KB-Seite, die nah an der Grenze von 1012 Spalten liegen, kann das Importieren von PC/IXF-Dateien dazu führen, dass DB2 einen Fehler zurückgibt, weil die maximale Größe einer SQL-Anweisung überschritten wurde. Diese Situation kann nur dann eintreten, wenn die Spalten den Typ CHAR, VARCHAR oder CLOB haben. Die Einschränkung gilt nicht für den Import von DEL- oder ASC-Dateien.

Mit DB2 Connect können Daten auf DRDA-Servern wie DB2 für OS/390, DB2 für VM und VSE sowie DB2 für OS/400 importiert werden. Nur der PC/IXF-Import (Option INSERT) wird unterstützt. Der Parameter **restartcnt** wird ebenfalls unterstützt, nicht jedoch der Parameter **commitcnt**.

Bei Verwendung der Option CREATE für typisierte Tabellen müssen alle in der PC/IXF-Datei definierten untergeordneten Tabellen erstellt werden. Die Definitionen von untergeordneten Tabellen können nicht geändert werden. Werden andere Optionen als CREATE bei typisierten Tabellen verwendet, können Sie über die Traversierfolgeliste die Traversierfolge angeben. Daher muss die Traversierfolgeliste mit der Liste übereinstimmen, die während der Exportoperation verwendet wurde. Beim PC/IXF-Dateiformat muss lediglich der Name der untergeordneten Zieltabelle angegeben und die in der Datei gespeicherte Traversierfolge verwendet werden.

Mit dem Dienstprogramm IMPORT können Sie eine Tabelle wiederherstellen, die zuvor in eine PC/IXF-Datei exportiert wurde. Die Tabelle hat dann denselben Status wie zum Zeitpunkt des Exports.

Das Importieren von Daten in eine Systemtabelle, eine deklarierte temporäre Tabelle oder eine Übersichtstabelle ist nicht möglich.

Mit dem Dienstprogramm IMPORT können keine Sichten erstellt werden.

Für das Betriebssystem Windows NT gilt:

- Das Importieren von logisch geteilten PC/IXF-Dateien wird nicht unterstützt.
- Das Importieren von PC/IXF- oder WSF-Dateien mit einem fehlerhaften Format wird nicht unterstützt.

### Hinweise zu DB2 Data Links Manager

Bitte führen Sie Folgendes aus, bevor Sie das DB2-Dienstprogramm IMPORT verwenden:

1. Kopieren Sie die Dateien, auf die verwiesen wird, auf die entsprechenden Data Links-Server. Mit dem Dienstprogramm **dlfm\_import** können Dateien aus einem Archiv extrahiert werden, das vom Dienstprogramm **dlfm\_export** erstellt wurde.
2. Registrieren Sie die erforderlichen Präfixnamen in den DB2 Data Links Managern. Sie müssen eventuell auch andere Verwaltungsaufgaben ausführen, wie ggfs. das Registrieren der Datenbank.
3. Aktualisieren Sie die Informationen zu den Data Links-Servern in den URL-Adressen (der DATALINK-Spalten) aus den exportierten Daten für die SQL-Tabelle (falls erforderlich). (Wenn die Data Links-Server der ursprünglichen Konfiguration mit den Servern an der Zielposition übereinstimmen, müssen die Namen der Data Links-Server nicht aktualisiert werden.)
4. Definieren Sie die Data Links-Server in der Zielkonfiguration in der Konfigurationsdatei für DB2 Data Links Manager.

Wenn das Dienstprogramm IMPORT für die Zieldatenbank ausgeführt wird, werden Dateien, auf die in DATALINK-Spalten verwiesen wird, mit den entsprechenden Data Links-Servern verknüpft.

Während der Einfügeoperation werden bei der Verarbeitung von DATALINK-Spalten die Dateien auf den entsprechenden Data Links-Servern gemäß den Spaltenspezifikationen der Zieldatenbank verknüpft.

### Zugehörige Referenzen:

- „SQLCA“ in *Administrative API Reference*
- „SQLDCOL“ in *Administrative API Reference*
- „SQLU-MEDIA-LIST“ in *Administrative API Reference*
- „Änderungswerte für Dateityp für IMPORT“ auf Seite 66
- „Einschränkungen für Begrenzer beim Versetzen von Daten“ auf Seite 247

### Zugehörige Beispiele:

- „dtformat.sqc -- Load and import data format extensions (C)“
- „tbmove.sqc -- How to move table data (C)“
- „expsamp.sqb -- Export and import tables with table data to a DRDA database (IBM COBOL)“
- „impexp.sqb -- Export and import tables with table data (IBM COBOL)“
- „tbmove.sqC -- How to move table data (C++)“

## Änderungswerte für Dateityp für IMPORT

Tabelle 4. Gültige Änderungswerte für Dateityp für IMPORT: Alle Dateiformate

Änderungswert	Beschreibung
compound= <i>x</i>	<p><i>x</i> ist eine Zahl zwischen 1 und 100 (jeweils einschließlich). Verwendet keine ganzzahlige zusammengesetzte SQL-Anweisung zum Einfügen der Daten, und es werden jeweils <i>x</i> Anweisungen versucht.</p> <p>Wenn dieser Änderungswert angegeben wird und das Transaktionsprotokoll nicht groß genug ist, schlägt die Importoperation fehl. Das Transaktionsprotokoll muss groß genug sein, um entweder die durch COMMITCOUNT angegebene Anzahl Zeilen oder die Anzahl der Zeilen in der Datendatei aufnehmen zu können (wenn COMMITCOUNT nicht angegeben wurde). Es empfiehlt sich daher, die Option COMMITCOUNT anzugeben, um einen Überlauf des Transaktionsprotokolls zu verhindern.</p> <p>Dieser Änderungswert ist mit dem Modus INSERT_UPDATE, mit hierarchischen Tabellen und den folgenden Änderungswerten nicht kompatibel: usedefaults, identitymissing, identityignore, generatedmissing und generatedignore.</p>
generatedignore	Dieser Änderungswert informiert das Dienstprogramm IMPORT darüber, dass für alle generierten Spalten in der Datendatei Daten vorhanden sind, aber ignoriert werden sollen. Dies bewirkt, dass alle Werte für die generierten Spalten durch das Dienstprogramm generiert werden. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert generatedmissing verwendet werden.
generatedmissing	Wird dieser Änderungswert angegeben, geht das Dienstprogramm davon aus, dass die Eingabedatendatei keine Daten (nicht einmal Nullwerte) für die generierten Spalten enthält, und generiert daher für jede Zeile einen Wert. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert generatedignore verwendet werden.
identityignore	Dieser Änderungswert informiert das Dienstprogramm IMPORT darüber, dass für die Identitätsspalte in der Datendatei Daten vorhanden sind, aber ignoriert werden sollen. Dies bewirkt, dass alle Identitätswerte durch das Dienstprogramm generiert werden. Auf diese Weise wird sowohl bei als GENERATED ALWAYS definierten Identitätsspalten als auch bei als GENERATED BY DEFAULT definierten Identitätsspalten verfahren. Dies bedeutet, dass bei als GENERATED ALWAYS definierten Spalten keine Zeilen zurückgewiesen werden. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert identitymissing verwendet werden.
identitymissing	Wird dieser Änderungswert angegeben, geht das Dienstprogramm davon aus, dass die Eingabedatendatei keine Daten (nicht einmal Nullwerte) für die Identitätsspalte enthält, und generiert daher für jede Zeile einen Wert. Auf diese Weise wird sowohl bei als GENERATED ALWAYS definierten Identitätsspalten als auch bei als GENERATED BY DEFAULT definierten Identitätsspalten verfahren. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert identityignore verwendet werden.

Tabelle 4. Gültige Änderungswerte für Dateityp für IMPORT: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
lobsinfile	<p><i>lob-pfad</i> gibt den Pfad zu den Dateien an, die LOB-Daten enthalten.</p> <p>Jeder Pfad enthält mindestens eine Datei, die wiederum mindestens ein großes Objekt (LOB) enthält, auf das durch eine LOB-Positionskenung (LOB Location Specifier - LLS) in der Datendatei verwiesen wird. Die LOB-Positionskenung ist eine Zeichenfolgendarstellung der Position eines großen Objekts (LOB) in einer Datei, die im LOB-Dateipfad gespeichert ist. Das Format einer LOB-Positionskenung lautet <i>dateiname.erw.nnn.mmm/</i>. Hierbei steht <i>dateiname.erw</i> für den Namen der Datei, die das große Objekt (LOB) enthält, <i>nnn</i> für die (in Byte angegebene) relative Position des großen Objekts in der Datei und <i>mmm</i> für die Länge des großen Objekts in Byte. Falls beispielsweise die Zeichenfolge <i>db2exp.001:123:456/</i> in der Datendatei gespeichert ist, befindet sich das große Objekt (LOB) an der relativen Position 123 in der Datei 'db2exp.001' und ist 456 Byte lang.</p> <p>Die Klausel LOBS FROM gibt an, wo sich die LOB-Dateien befinden, wenn der Änderungswert „lobsinfile“ verwendet wird. Außerhalb des Kontextes des Änderungswerts <i>lobsinfile</i> hat die Klausel LOBS FROM keine Bedeutung. Die Klausel LOBS FROM überträgt die Liste der Pfade an das Dienstprogramm IMPORT, die beim Importieren der Daten nach LOB-Dateien durchsucht werden sollen.</p> <p>Um ein Null-LOB anzugeben, geben Sie als Größe den Wert -1 ein. Wenn die Größe mit dem Wert 0 angegeben wird, gilt die Angabe als ein großes Objekt mit der Länge 0. Bei Null-LOBs mit der Länge -1 werden die relative Adresse und der Dateiname ignoriert. Die LOB-Positionskenung eines Null-LOBs könnte beispielsweise <i>db2exp.001:7:-1/</i> lauten.</p>
no_type_id	Nur beim Import in eine einzelne untergeordnete Tabelle gültig. Wird typischerweise für den Export von Daten aus einer regulären Tabelle und dem nachfolgenden Aufruf einer Importoperation (mit diesem Änderungswert) zur Konvertierung der Daten in einer einzelnen untergeordneten Tabelle verwendet.
nodefaults	<p>Wenn nicht ausdrücklich eine Quellenspalte für eine Zieltabellenspalte angegeben ist und die Tabellenspalte keine Nullwerte enthalten darf, werden keine Standardwerte geladen. Ohne diese Option geschieht Folgendes, wenn eine Quellenspalte für eine der Zieltabellenspalten nicht ausdrücklich angegeben ist:</p> <ul style="list-style-type: none"> <li>• Wenn ein Standardwert für eine Spalte angegeben werden kann, wird der Standardwert geladen.</li> <li>• Wenn die Spalte Nullwerte enthalten darf und für diese Spalte kein Standardwert angegeben werden kann, wird ein Nullwert geladen.</li> <li>• Wenn die Spalte keine Nullwerte enthalten darf und kein Standardwert angegeben werden kann, wird ein Fehler zurückgegeben, und das Dienstprogramm beendet die Verarbeitung.</li> </ul>
norowwarnings	Unterdrückt alle Warnungen zu zurückgewiesenen Zeilen.

## Import-API "db2Import"

Tabelle 4. Gültige Änderungswerte für Dateityp für IMPORT: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
usedefaults	<p>Wenn eine Quellenspalte für eine Zieltabellenspalte angegeben wurde, jedoch für mindestens ein Zeilenexemplar keine Daten enthält, werden Standardwerte geladen. Beispiele für fehlende Daten sind:</p> <ul style="list-style-type: none"> <li>• Für DEL-Dateien: ",," wird für die Spalte angegeben.</li> <li>• Für ASC-Dateien: Der NULL-Anzeiger wird für die Spalte auf Ja gesetzt.</li> <li>• Für DEL/ASC/WSF-Dateien: Eine Zeile, die nicht genug Spalten enthält oder für die ursprüngliche Spezifikation nicht lang genug ist.</li> </ul> <p>Ohne diese Option geschieht Folgendes, wenn eine Quellenspalte keine Daten für ein Zeilenexemplar enthält:</p> <ul style="list-style-type: none"> <li>• Wenn die Spalte Nullwerte enthalten darf, wird NULL geladen.</li> <li>• Wenn die Spalte keine Nullwerte enthalten darf, wird die Zeile vom Dienstprogramm zurückgewiesen.</li> </ul>

Tabelle 5. Gültige Änderungswerte für Dateityp für IMPORT: ASCII-Dateiformate (ASC/DEL)

Änderungswert	Beschreibung
codepage=x	<p>x ist eine ASCII-Zeichenfolge. Der Wert wird als Codepage der Daten im Ausgabedatensatz gewertet. Während der Importoperation werden Zeichendaten aus der Codepage der Anwendung in diese Codepage konvertiert.</p> <p>Es gelten folgende Regeln:</p> <ul style="list-style-type: none"> <li>• Bei reinem DBCS (Grafik), gemischtem DBCS und EUC sind Begrenzer auf den Bereich von x00 bis x3F (jeweils einschließlich) beschränkt.</li> <li>• nullindchar muss Symbole angeben, die im Standard-ASCII-Satz zwischen den Codepunkten x20 und x7F (jeweils einschließlich) enthalten sind. Das gilt für ASCII-Symbole und -Codepunkte.</li> </ul> <p><b>Anmerkungen:</b></p> <ol style="list-style-type: none"> <li>1. Der Änderungswert codepage kann nicht zusammen mit dem Änderungswert lobsinfile verwendet werden.</li> <li>2. Wenn während der Codepagekonvertierung von der Anwendungscodepage in die Datenbankcodepage eine Datenerweiterung stattfindet, kann es sein, dass die Daten abgeschnitten werden und ein Datenverlust entsteht.</li> </ol>
dateformat="x"	<p>x ist das Datumsformat in der Quellendatei.<sup>2</sup> Gültige Datumselemente sind:</p> <p>YYYY - Jahr (vier Ziffern von 0000 bis 9999)</p> <p>M - Monat (eine oder zwei Ziffern von 1 bis 12)</p> <p>MM - Monat (zwei Ziffern von 1 bis 12; dieses Element und das Element M schließen sich gegenseitig aus)</p> <p>D - Tag (eine oder zwei Ziffern von 1 bis 31)</p> <p>DD - Tag (zwei Ziffern von 1 bis 31; dieses Element und das Element D schließen sich gegenseitig aus)</p> <p>DDD - Tag des Jahres (drei Ziffern von 001 bis 366; dieses Element und andere Elemente für Tag oder Monat schließen sich gegenseitig aus)</p> <p>Für jedes Element, das nicht angegeben wird, wird der Standardwert 1 zugeordnet. Einige Beispiele für Datumsformate:</p> <p>"D-M-YYYY"</p> <p>"MM.DD.YYYY"</p> <p>"YYYYDDD"</p>

Tabelle 5. Gültige Änderungswerte für Dateityp für IMPORT: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
implieddecimal	Die Position eines implizierten Dezimalzeichens wird durch die Spaltendefinition bestimmt. Sie wird nicht mehr am Ende des Werts angenommen. Zum Beispiel wird der Wert 12345 in eine Spalte, für die DECIMAL(8,2) definiert ist, als 123.45 und <i>nicht</i> als 12345.00 geladen.
noeofchar	Das wahlfreie Dateiendezeichen x'1A' wird nicht als Ende der Datei erkannt. Die Verarbeitung wird wie bei einem normalen Zeichen fortgesetzt.
timeformat="x"	<p>x ist das Zeitformat in der Quellendatei.<sup>2</sup> Gültige Zeitelemente sind:</p> <ul style="list-style-type: none"> <li>H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System)</li> <li>HH - Stunde (zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus)</li> <li>M - Minute (eine oder zwei Ziffern von 0 bis 59)</li> <li>MM - Minute (zwei Ziffern von 0 bis 59; dieses Element und das Element M schließen sich gegenseitig aus)</li> <li>S - Sekunde (eine oder zwei Ziffern von 0 bis 59)</li> <li>SS - Sekunde (zwei Ziffern von 0 bis 59; dieses Element und das Element S schließen sich gegenseitig aus)</li> <li>SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86399; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus)</li> <li>TT - Meridiananzeiger (AM oder PM)</li> </ul> <p>Für jedes Element, das nicht angegeben wird, wird der Standardwert 0 zugeordnet. Einige Beispiele für Zeitformate:</p> <pre>"HH:MM:SS" "HH.MM TT" "SSSSS"</pre>

## Import-API "db2Import"

Tabelle 5. Gültige Änderungswerte für Dateityp für IMPORT: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
timestampformat="x"	<p>x ist das Format der Zeitmarke in der Quellendatei.<sup>2</sup> Gültige Zeitmarkenelemente sind:</p> <p>YYYY - Jahr (vier Ziffern von 0000 bis 9999)</p> <p>M - Monat (eine oder zwei Ziffern von 1 bis 12)</p> <p>MM - Monat (zwei Ziffern von 01 bis 12; dieses Element und das Element M schließen sich gegenseitig aus)</p> <p>MMM - Monat (Abkürzung des Monatsnamen, die aus drei Buchstaben besteht und bei der die Großschreibung zu beachten ist; dieses Element und die Elemente M und MM schließen sich gegenseitig aus)</p> <p>D - Tag (eine oder zwei Ziffern von 1 bis 31)</p> <p>DD - Tag (zwei Ziffern von 1 bis 31; dieses Element und Element D schließen sich gegenseitig aus)</p> <p>DDD - Tag des Jahres (drei Ziffern von 001 bis 366; dieses Element und andere Elemente für Tag oder Monat schließen sich gegenseitig aus)</p> <p>H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System)</p> <p>HH - Stunde (zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus)</p> <p>M - Minute (ein oder zwei Ziffern von 0 bis 59)</p> <p>MM - Minute (zwei Ziffern von 0 bis 59; dieses Element und das Minutenelement M schließen sich gegenseitig aus)</p> <p>S - Sekunde (ein oder zwei Ziffern von 0 bis 59)</p> <p>SS - Sekunde (zwei Ziffern von 0 bis 59; dieses Element und das Element S schließen sich gegenseitig aus)</p> <p>SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86399; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus)</p> <p>UUUUUU - Mikrosekunde (6 Ziffern von 000000 bis 999999; dieses Element und alle anderen Mikrosekundenelemente schließen sich gegenseitig aus)</p> <p>UUUUU - Mikrosekunde (5 Ziffern von 00000 bis 99999; dieses Element wird dem Bereich 000000 bis 999990 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>UUUU - Mikrosekunde (4 Ziffern von 0000 bis 9999; dieses Element wird dem Bereich 000000 bis 999900 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>UUU - Mikrosekunde (3 Ziffern von 000 bis 999; dieses Element wird dem Bereich 000000 bis 999000 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>UU - Mikrosekunde (2 Ziffern von 00 bis 99; dieses Element wird dem Bereich 000000 bis 990000 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>U - Mikrosekunde (1 Ziffer von 0 bis 9; dieses Element wird dem Bereich 000000 bis 900000 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>TT - Meridiananzeiger (AM oder PM)</p>

Tabelle 5. Gültige Änderungswerte für Dateityp für IMPORT: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
timestampformat="x"	<p>Für nicht angegebene Elemente YYYY, M, MM, D, DD oder DDD wird der Standardwert 1 zugeordnet. Der Standardwert 'Jan' wird einem nicht spezifizierten Element MMM zugeordnet. Für alle anderen nicht angegebenen Elemente wird der Standardwert 0 zugeordnet. Beispiel für ein Zeitmarkenformat:</p> <pre>"YYYY/MM/DD HH:MM:SS.UUUUU"</pre> <p>Die gültigen Werte für das Element MMM sind folgende: 'jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov' und 'dec'. Bei diesen Werten wird die Groß-/Kleinschreibung nicht beachtet.</p> <p>Das folgende Beispiel veranschaulicht, wie Daten in eine Tabelle namens SCHEDULE importiert werden, die benutzerdefinierte Datums- und Zeitformate enthalten:</p> <pre>db2 import from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>
usegraphiccodepage	<p>Bei Angabe von usegraphiccodepage wird angenommen, dass sich die Daten, die in Datenfelder von Grafiken oder großen Doppelbytezeichenobjekten (DBCLOB) importiert werden, in der Grafikcodepage befinden. Es wird angenommen, dass sich die übrigen Daten in der Zeichencodepage befinden. Die Grafikcodepage ist der Zeichencodepage zugeordnet. Das Dienstprogramm IMPORT ermittelt die Zeichencodepage entweder über den Änderungswert codepage, falls angegeben, oder über die Codepage der Anwendung, wenn der Änderungswert codepage nicht angegeben ist.</p> <p>Dieser Änderungswert sollte zusammen mit der begrenzten Datendatei verwendet werden, die bei einer Wiederherstellung nach einer DROP TABLE-Operation generiert wird, wenn die wiederherzustellende Tabelle Grafikdaten enthält.</p> <p><b>Einschränkungen</b></p> <p>Der Änderungswert usegraphiccodepage DARF NICHT mit DEL- oder ASC-Dateien verwendet werden, die durch das Dienstprogramm EXPORT erstellt werden, weil diese Dateien Daten enthalten, die nur in einer Codepage codiert sind. Der Änderungswert usegraphiccodepage wird auch von den großen Doppelbytezeichenobjekten (DBCLOBs) in Dateien ignoriert.</p>

Tabelle 6. Gültige Änderungswerte für Dateityp für IMPORT: ASC-Dateiformate (ASCII-Format mit universellen Zeilenbegrenzern)

Änderungswert	Beschreibung
nochecklengths	<p>Wenn nochecklengths angegeben wird, wird versucht, jede Zeile zu importieren, auch wenn die Quelldaten eine Spaltendefinition enthalten, die die Größe der Zieltabellenspalte überschreitet. Solche Zeilen können erfolgreich importiert werden, wenn die Codepagekonvertierung eine Verkleinerung der Quelldaten bewirkt. Beispielsweise könnten 4-Byte-EUC-Daten in der Quelle auf 2-Byte-DBCS-Daten im Ziel verkleinert werden und den halben Speicherplatz belegen. Diese Option ist besonders nützlich, wenn bekannt ist, dass die Quelldaten in jedem Fall Platz finden, unabhängig von abweichenden Spaltendefinitionen.</p>
nullindchar=x	<p>x ist ein einzelnes Zeichen. Ändert das Zeichen, das einen NULL-Wert angibt, in x. Der Standardwert für x ist Y.<sup>3</sup></p> <p>Bei diesem Änderungswert wird bei EBCDIC-Datendateien zwischen Groß-/Kleinschreibung unterschieden, sofern es sich bei dem Zeichen nicht um ein lateinisches Zeichen handelt. Wenn beispielsweise für das NULL-Anzeigerzeichen der Buchstabe N angegeben wird, wird auch n als NULL-Anzeiger erkannt.</p>

## Import-API "db2Import"

Tabelle 6. Gültige Änderungswerte für Dateityp für IMPORT: ASC-Dateiformate (ASCII-Format mit universellen Zeilenbegrenzern) (Forts.)

Änderungswert	Beschreibung
reclen= <i>x</i>	<i>x</i> ist eine ganze Zahl mit dem Maximalwert 32 767. <i>x</i> Zeichen werden für jede Zeile gelesen, und es wird kein Zeilenvorschubzeichen verwendet, um das Ende der Zeile anzugeben.
striptblanks	<p>Schneidet abschließende Leerzeichen ab, wenn Daten in ein Feld mit variabler Länge geladen werden. Wenn diese Option nicht angegeben wird, werden Leerstellen beibehalten.</p> <p>Im folgenden Beispiel bewirkt <code>striptblanks</code>, dass das Dienstprogramm IMPORT abschließende Leerzeichen abschneidet:</p> <pre>db2 import from myfile.asc of asc modified by striptblanks method l (1 10, 12 15) messages msgs.txt insert into staff</pre> <p>Diese Option kann nicht zusammen mit <code>striptnulls</code> angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p> <p><b>Anmerkung:</b> Diese Option ersetzt die veraltete Option <code>t</code>, die nur noch aus Gründen der Kompatibilität zu früheren Versionen unterstützt wird.</p>
striptnulls	<p>Schneidet abschließende NULL-Zeichen (0x00-Zeichen) ab, wenn Daten in ein Feld mit variabler Länge geladen werden. Wenn diese Option nicht angegeben wird, werden NULL-Zeichen beibehalten.</p> <p>Diese Option kann nicht zusammen mit <code>striptblanks</code> angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p> <p><b>Anmerkung:</b> Diese Option ersetzt die veraltete Option <code>padwithzero</code>, die nur noch aus Gründen der Kompatibilität mit früheren Versionen unterstützt wird.</p>

Tabelle 7. Gültige Änderungswerte für Dateityp für IMPORT: DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer)

Änderungswert	Beschreibung
chardel <i>x</i>	<p><i>x</i> ist ein einzelner Zeichenfolgebegrenzer. Der Standardwert ist ein doppeltes Anführungszeichen ("). Das angegebene Zeichen wird anstelle von doppelten Anführungszeichen verwendet, um eine Zeichenfolge einzuschließen.<sup>34</sup> Falls Sie die doppelten Anführungszeichen explizit als Zeichenfolgebegrenzer angeben wollen, müssen Sie dies folgendermaßen tun:</p> <pre>modified by chardel"</pre> <p>Auch das einfache Anführungszeichen (') kann als Zeichenfolgebegrenzer angegeben werden. Im folgenden Beispiel bewirkt <code>chardel''</code>, dass das Dienstprogramm IMPORT jedes einfache Anführungszeichen (') als Zeichenfolgebegrenzer wertet:</p> <pre>db2 "import from myfile.del of del modified by chardel'' method p (1, 4) insert into staff (id, years)"</pre>
coldel <i>x</i>	<p><i>x</i> ist ein einzelner Zeichenspaltenbegrenzer. Der Standardwert ist ein Komma (,). Das angegebene Zeichen wird anstelle eines Kommas verwendet, um das Ende einer Spalte anzugeben.<sup>34</sup></p> <p>Im folgenden Beispiel bewirkt <code>coldel;</code>, dass das Dienstprogramm IMPORT jedes Semikolon (;) als Spaltenbegrenzungszeichen wertet:</p> <pre>db2 import from myfile.del of del modified by coldel; messages msgs.txt insert into staff</pre>
datesiso	Datumsformat. Bewirkt, dass alle Datumswerte im ISO-Format importiert werden.

Tabelle 7. Gültige Änderungswerte für Dateityp für IMPORT: DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer) (Forts.)

Änderungswert	Beschreibung
decplusblank	Zeichen für Pluszeichen. Bewirkt, dass positiven Dezimalwerten statt eines Pluszeichens (+) ein Leerzeichen vorangestellt wird. Die Standardaktion besteht darin, positiven Dezimalwerten ein Pluszeichen voranzustellen.
decptx	<p><math>x</math> ist ein einzelnes Ersatzzeichen für den Punkt als Dezimalzeichen. Der Standardwert ist ein Punkt (.). Das angegebene Zeichen wird anstelle eines Punktes als Dezimalzeichen verwendet.<sup>34</sup></p> <p>Im folgenden Beispiel bewirkt decpt;, dass das Dienstprogramm IMPORT jedes Semikolon (;) als Dezimalzeichen wertet:</p> <pre>db2 "import from myfile.del of del modified by char del ' decpt; messages msgs.txt insert into staff"</pre>
delprioritychar	<p>Die aktuelle Standardpriorität für Begrenzer ist: Satzbegrenzer, Zeichenbegrenzer, Spaltenbegrenzer. Durch diesen Änderungswert werden vorhandene Anwendungen geschützt, die auf die früher geltende Priorität angewiesen sind, indem die Begrenzerpriorität wie folgt geändert wird: Zeichenbegrenzer, Satzbegrenzer, Spaltenbegrenzer. Syntax:</p> <pre>db2 import ... modified by delprioritychar ...</pre> <p>Beispiel anhand der folgenden DEL-Datendatei:</p> <pre>"Smith, Joshua",4000,34.98&lt;zeilenbegrenzer&gt; "Vincent,&lt;zeilenbegrenzer&gt;, is a manager", ... ... 4005,44.37&lt;zeilenbegrenzer&gt;</pre> <p>Wenn der Änderungswert delprioritychar angegeben wird, gibt es in dieser Datendatei nur zwei Zeilen. Der zweite &lt;zeilenbegrenzer&gt; wird als Teil der ersten Datenspalte der zweiten Zeile gewertet, während der erste und dritte &lt;zeilenbegrenzer&gt; als eigentliche Satzbegrenzer gewertet werden. Wenn dieser Änderungswert <i>nicht</i> angegeben wird, gibt es in dieser Datendatei drei Zeilen, jeweils getrennt durch einen &lt;zeilenbegrenzer&gt;.</p>
dldelx	<p><math>x</math> ist ein DATALINK-Begrenzer (Einzelzeichen). Der Standardwert ist ein Semikolon (;). Das angegebene Zeichen wird anstelle eines Semikolons als Trennzeichen zwischen Feldern für einen DATALINK-Wert verwendet. Es ist erforderlich, weil ein DATALINK-Wert mehrere Teilwerte enthalten kann.<sup>34</sup></p> <p><b>Anmerkung:</b> <math>x</math> darf nicht mit dem Zeichen übereinstimmen, das als Zeilen-, Spalten- oder Zeichenfolgebegrenzer angegeben wurde.</p>
keepblanks	Erhält die führenden und abschließenden Leerzeichen in allen Feldern des Typs CHAR, VARCHAR, LONG VARCHAR oder CLOB. Ohne diese Option werden alle führenden und abschließenden Leerzeichen, die nicht innerhalb von Zeichenbegrenzern stehen, entfernt, und in die Tabelle wird für jedes leere Feld ein Nullwert eingefügt.
nochardel	<p>Das Importdienstprogramm wird davon ausgehen, dass alle Bytes zwischen den Spaltenbegrenzern Teil der Spaltendaten sind. Zeichenbegrenzer werden als Teil der Spaltendaten syntaktisch analysiert. Diese Option sollte nicht angegeben werden, wenn die Daten mit Hilfe von DB2 exportiert wurden (sofern nochardel zum Zeitpunkt des Exports angegeben wurde). Sie wird bereitgestellt, um Datendateien von Herstellern zu unterstützen, die nicht über Zeichenbegrenzer verfügen. Eine unsachgemäße Verwendung kann zu Datenverlust oder Fehlern führen.</p> <p>Diese Option kann nicht zusammen mit char del x, delprioritychar oder nodouble del angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p>
nodoubledel	Unterdrückt die Erkennung von doppelten Zeichenbegrenzern.

## Import-API "db2Import"

Tabelle 8. Gültige Änderungswerte für Dateityp für IMPORT: IXF-Dateiformat

Änderungswert	Beschreibung
forcein	Weist das Dienstprogramm an, Daten trotz Codepage-Abweichungen zu akzeptieren und die Umsetzung zwischen Codepages zu unterdrücken.  Zielfelder mit fester Länge werden überprüft, um sicherzustellen, dass sie groß genug für die Daten sind. Wenn <code>nochecklengths</code> angegeben ist, wird keine Überprüfung durchgeführt und nach Möglichkeit jede Zeile geladen.
indexif	Weist das Dienstprogramm an, alle Indizes zu löschen, die derzeit für die vorhandene Tabelle definiert sind, und aus den Indexdefinitionen in der PC/IXF-Datei neue Indizes zu erstellen. Diese Option kann nur verwendet werden, wenn der Inhalt der Tabelle ersetzt wird. Sie kann nicht mit einer Sicht verwendet werden, oder wenn eine <i>einfügespalte</i> angegeben ist.
indexschema= <i>schema</i>	Verwendet bei der Indexerstellung das angegebene <i>schema</i> für den Indexnamen. Wenn <i>schema</i> nicht angegeben wird (aber das Schlüsselwort <code>indexschema</code> angegeben ist), wird die Verbindungs-Benutzer-ID verwendet. Wenn das Schlüsselwort nicht angegeben ist, wird das Schema in der IXF-Datei verwendet.
nochecklengths	Wenn <code>nochecklengths</code> angegeben wird, wird versucht, jede Zeile zu importieren, auch wenn die Quelldaten eine Spaltendefinition enthalten, die die Größe der Zieltabellenspalte überschreitet. Solche Zeilen können erfolgreich importiert werden, wenn die Codepagekonvertierung eine Verkleinerung der Quelldaten bewirkt. Beispielsweise könnten 4-Byte-EUC-Daten in der Quelle auf 2-Byte-DBCS-Daten im Ziel verkleinert werden und den halben Speicherplatz belegen. Diese Option ist besonders nützlich, wenn bekannt ist, dass die Quelldaten in jedem Fall Platz finden, unabhängig von abweichenden Spaltendefinitionen.

### Anmerkungen:

1. Das Dienstprogramm IMPORT setzt keine Warnung ab, wenn versucht wird, nicht unterstützte Dateitypen mit der Option MODIFIED BY zu verwenden. Wenn dies versucht wird, schlägt die Importoperation fehl, und es wird ein Fehlercode zurückgegeben.
2. Die Datumsformatzeichenfolge muss unbedingt in doppelte Anführungszeichen eingeschlossen werden. Feldtrennzeichen dürfen die folgenden Zeichen nicht enthalten: a-z, A-Z und 0-9. Das Feldtrennzeichen darf nicht mit dem Zeichenbegrenzer oder dem Feldbegrenzer im DEL-Dateiformat identisch sein. Die Verwendung eines Feldtrennzeichens ist wahlfrei, sofern die Anfangs- und Endposition eines Elements eindeutig ist. Dies ist beispielsweise dann nicht der Fall, wenn (abhängig vom Änderungswert) Elemente wie D, H, M oder S verwendet werden, da die Einträge in ihrer Länge variabel sind.

Bei Zeitmarkenformaten muss darauf geachtet werden, eine Mehrdeutigkeit zwischen den Deskriptoren für Monat und Minute zu vermeiden, da beide den Buchstaben M verwenden. Ein Monatsfeld muss neben anderen Datumsfeldern stehen. Ein Minutenfeld muss neben anderen Zeitfeldern stehen. Die folgenden Zeitmarkenformate sind beispielsweise mehrdeutig:

"M" (Hierdurch könnte sowohl eine Monats- als auch eine Minutenangabe dargestellt sein.)  
"M:M" (In diesem Fall ist nicht eindeutig, welches Element für den Monat und welches für die Minutenangabe steht.)  
"M:YYYY:M" (Hier werden beide Elemente M als Monat interpretiert.)  
"S:M:YYYY" (In diesem Fall steht das Element M sowohl neben einem Zeit- als auch neben einem Datumswert.)

Im Fall einer Mehrdeutigkeit meldet das Dienstprogramm eine Fehlermeldung, und die Operation schlägt fehl.

Die folgenden Zeitmarkenformate sind beispielsweise eindeutig:

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

Einigen Zeichen (z. B. doppelten Anführungszeichen und umgekehrten Schrägstrichen) muss ein Escape-Zeichen vorangestellt sein (beispielsweise \).

3. Das Zeichen muss in der Codepage der Quelldaten angegeben werden.

Der Zeichencodpunkt (statt des Zeichensymbols) kann mit der Syntax xJJ bzw. 0xJJ angegeben werden, wobei JJ die hexadezimale Darstellung des Codepunkts ist. Beispiel: Das Zeichen # kann wie folgt als Spaltenbegrenzungszeichen angegeben werden:

- ... modified by coldel# ...
- ... modified by coldel0x23 ...
- ... modified by coldelX23 ...

4. Unter Delimiter restrictions for moving data sind Einschränkungen für die Zeichen aufgeführt, die als Überschreibungswerte für Begrenzer verwendet werden können.

Tabelle 9. Verhalten des Dienstprogramms IMPORT bei Verwendung von 'codepage' und 'usegraphiccodepage'

codepage=N	usegraphiccodepage	Verhalten von IMPORT
Nicht vorhanden	Nicht vorhanden	Es wird angenommen, dass sich alle Daten in der Datei in der Anwendungscodepage befinden.
Vorhanden	Nicht vorhanden	Es wird angenommen, dass sich alle Daten in der Datei in der Codepage N befinden.  <b>Warnung:</b> Grafikdaten werden beim Importieren in die Datenbank beschädigt, wenn N eine Einzelbyte-Codepage ist.
Nicht vorhanden	Vorhanden	Es wird angenommen, dass sich Zeichendaten in der Datei in der Anwendungscodepage befinden. Es wird angenommen, dass sich Grafikdaten in der Codepage der Anwendungsgrafikdaten befinden.  Wenn die Anwendungscodepage eine Einzelbyte-Codepage ist, wird angenommen, dass sich alle Daten in der Anwendungscodepage befinden.  <b>Warnung:</b> Wenn die Anwendungscodepage eine Einzelbyte-Codepage ist, werden Grafikdaten beim Importieren in die Datenbank beschädigt, auch wenn die Datenbank Grafikspalten enthält.
Vorhanden	Vorhanden	Es wird angenommen, dass sich Zeichendaten in der Codepage N befinden. Es wird angenommen, dass sich Grafikdaten in der Grafikcodepage von N befinden.  Wenn N eine Einzelbyte- oder Doppelbyte-Codepage ist, wird angenommen, dass alle Daten sich in der Codepage N befinden.  <b>Warnung:</b> Grafikdaten werden beim Importieren in die Datenbank beschädigt, wenn N eine Einzelbyte-Codepage ist.

### Zugehörige Referenzen:

- „Import-API "db2Import"“ auf Seite 53
- „Befehl IMPORT“ auf Seite 39
- „Einschränkungen für Begrenzer beim Versetzen von Daten“ auf Seite 247

---

## Hinweise zu Zeichensatz und NLS

Manchmal kann es aufgrund ungleicher Codepages zu Situationen kommen, in denen die Zeichendaten möglicherweise verlängert oder gekürzt werden. Solche Situationen können bei Zeichensätzen mit erweitertem UNIX-Code (Extended UNIX<sup>®</sup> Code, EUC) und Doppelbytezeichensätzen (DBCS) für Japanisch oder traditionelles Chinesisch auftreten, die möglicherweise verschiedene Längencodierungen für dasselbe Zeichen verwenden. Normalerweise wird die Länge der Eingabedaten vor dem Einlesen von Daten mit der Länge der Zielspalte verglichen. Wenn die Länge der Eingabedaten die Länge der Zielspalte überschreitet, werden Nullwerte in die Spalte eingefügt, wenn für die Spalte Nullwerte zulässig sind. Andernfalls wird die Anforderung zurückgewiesen. Wenn der Änderungswert `nochecklengths` angegeben ist, wird vorab kein Vergleich durchgeführt, und die Daten werden nach Möglichkeit importiert. Wenn die Daten nach der Umsetzung zu lang sind, wird die entsprechende Zeile zurückgewiesen. Ansonsten werden die Daten importiert.

### Zugehörige Konzepte:

- „Zeichensatz und NLS“ auf Seite 189

### Zugehörige Referenzen:

- „Befehl IMPORT“ auf Seite 39

---

## CLP-Beispiele für IMPORT-Sitzungen

### Beispiel 1

Das folgende Beispiel zeigt, wie Sie die Daten aus `myfile.ixf` in die Tabelle `STAFF` importieren können:

```
db2 import from myfile.ixf of ixf messages msg.txt insert into staff
```

```
SQL3150N  Der H-Satz in der PC/IXF-Datei enthält das Produkt "DB2  
01.00", das Datum "19970220" und die Uhrzeit "140848".
```

```
SQL3153N  Der T-Satz in der PC/IXF-Datei hat den Namen "myfile", das  
Qualifikationsmerkmal " " und die Quelle " ".
```

```
SQL3109N  Das Dienstprogramm beginnt mit dem Laden von Daten aus  
der Datei "myfile".
```

```
SQL3110N  Die Verarbeitung des Dienstprogramms ist abgeschlossen. Es  
wurden "58" Zeile(n) aus der Eingabedatei gelesen.
```

```
SQL3221W  ...COMMIT WORK beginnt. Zähler für Eingabesätze: "58".
```

```
SQL3222W  ...COMMIT für alle Änderungen der Datenbank wurde  
erfolgreich ausgeführt.
```

```
SQL3149N  "58" Zeile(n) aus der Eingabedatei wurden verarbeitet. "58"  
Zeile(n) wurden ohne Fehler der Tabelle hinzugefügt. "0" Zeile(n)  
wurden zurückgewiesen.
```

**Beispiel 2**

Das folgende Beispiel zeigt, wie die Tabelle MOVIE TABLE aus der Eingabedatei delfile1 importiert werden kann, die Daten im DEL-Format enthält:

```
db2 import from delfile1 of del
modified by dldel|
insert into movietable (actorname, description, url_making_of,
url_movie) datalink specification (dl_url_default_prefix
"http://narang"), (dl_url_replace_prefix "http://bomdel"
dl_url_suffix ".mpeg")
```

**Anmerkungen:**

1. Die Tabelle enthält vier Spalten:

actorname	VARCHAR(n)
description	VARCHAR(m)
url_making_of	DATALINK (mit LINKTYPE URL)
url_movie	DATALINK (mit LINKTYPE URL)

2. Die DATALINK-Daten in der Eingabedatei verwenden den vertikalen Balken (|) als Teilfeldbegrenzer.
3. Wenn ein Spaltenwert für url\_making\_of keine Präfixzeichenfolge aufweist, wird "http://narang" verwendet.
4. Jeder Spaltenwert ungleich NULL für url\_movie erhält "http://bomdel" als Präfix. Vorhandene Werte werden ersetzt.
5. An jeden Spaltenwert ungleich NULL für url\_movie wird ".mpeg" an den Pfad angefügt. Beispiel: Wenn ein Spaltenwert für url\_movie "http://server1/x/y/z" ist, wird er als "http://bomdel/x/y/z.mpeg" gespeichert; ist der Wert "/x/y/z", wird er als "http://bomdel/x/y/z.mpeg" gespeichert.

**Beispiel 3 (Importieren in eine Tabelle mit einer Identitätsspalte)**

TABLE1 enthält 4 Spalten:

- C1 VARCHAR(30)
- C2 INT GENERATED BY DEFAULT AS IDENTITY
- C3 DECIMAL(7,2)
- C4 CHAR(1)

TABLE2 entspricht TABLE1, wobei jedoch C2 eine als GENERATED ALWAYS definierte Identitätsspalte ist.

Die Datensätze in DATAFILE1 (DEL-Format) lauten:

```
"Liszt"
"Holst",,187.43, H
"Grieg",100, 66.34, G
"Satie",101, 818.23, I
```

Die Datensätze in DATAFILE2 (DEL-Format) lauten:

```
"Liszt", 74.49, A
"Holst", 0.01, H
"Grieg", 66.34, G
"Satie", 818.23, I
```

## Import-API "db2Import"

Der folgende Befehl generiert Identitätswerte für die Zeilen 1 und 2, da in der Datei DATAFILE1 keine Identitätswerte für diese Zeilen zur Verfügung gestellt werden. Den Zeilen 3 und 4 wird jedoch der benutzerdefinierte Identitätswert 100 bzw. 101 zugeordnet.

```
db2 import from datafile1.del of del replace into table1
```

Um die Datei DATAFILE1 so in die Tabelle TABLE1 zu importieren, dass für alle Zeilen Identitätswerte generiert werden, setzen Sie einen der folgenden Befehle ab:

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

Um die Datei DATAFILE2 so in die Tabelle TABLE1 zu importieren, dass für alle Zeilen Identitätswerte generiert werden, setzen Sie einen der folgenden Befehle ab:

```
db2 import from datafile2.del of del replace into table1 (c1, c3, c4)
db2 import from datafile2.del of del modified by identitymissing
  replace into table1
```

Wird die Datei DATAFILE1 in die Tabelle TABLE2 importiert, ohne dass einer der identitätsbezogenen Änderungswerte für den Dateityp verwendet wird, werden die Zeilen 1 und 2 eingefügt, die Zeilen 3 und 4 jedoch zurückgewiesen, da diese Zeilen eigene Werte, die keine Nullwerte sind, liefern und die Identitätsspalte als GENERATED ALWAYS definiert wurde.

### Beispiel 4 (Importieren mit Nullanzeigern)

TABLE1 enthält 5 Spalten:

- COL1 VARCHAR 20 NOT NULL WITH DEFAULT
- COL2 SMALLINT
- COL3 CHAR 4
- COL4 CHAR 2 NOT NULL WITH DEFAULT
- COL5 CHAR 2 NOT NULL

ASCFE1 enthält 6 Elemente:

- ELE1 Positionen 01 bis 20
- ELE2 Positionen 21 bis 22
- ELE5 Positionen 23 bis 23
- ELE3 Positionen 24 bis 27
- ELE4 Positionen 28 bis 31
- ELE6 Positionen 32 bis 32
- ELE6 Positionen 33 bis 40

Datensätze:

```
1...5...10...15...20...25...30...35...40
Testdaten 1      XXN 123abcdN
Testdaten 2 und 3  QQY   wxyzN
Testdaten 4,5 und 6 WVN6789  Y
```

Mit dem folgenden Befehl werden Datensätze aus ASCFILE1 in TABLE1 importiert:

```
db2 import from ascfile1 of asc
  method L (1 20, 21 22, 24 27, 28 31)
  null indicators (0, 0, 23, 32)
  insert into table1 (col1, col5, col2, col3)
```

### Anmerkungen:

1. Da COL4 nicht in der Eingabedatei bereitgestellt wird, wird sie in TABLE1 mit dem Standardwert eingefügt (sie ist mit NOT NULL WITH DEFAULT definiert).
2. Die Positionen 23 und 32 werden verwendet, um anzugeben, ob in COL2 und COL3 von TABLE1 für eine gegebene Zeile NULL geladen wird. Wenn die Nullanzeigerposition der Spalte für einen gegebenen Satz Y enthält, ist die Spalte NULL. Enthält sie N, werden die Datenwerte in den Datenpositionen der Spalte des Eingabedatensatzes (wie in L(.....) definiert) als Quelle der Spalten-  
daten für die Zeile verwendet. In diesem Beispiel ist keine Spalte in Zeile 1 NULL. COL2 in Zeile 2 ist NULL, und COL3 in Zeile 3 ist NULL.
3. In diesem Beispiel werden die NULL-Anzeiger für COL1 und COL5 als 0 (Null) angegeben, wodurch festgelegt wird, dass die Daten keine Nullwerte enthalten dürfen.
4. Der NULL-Anzeiger für eine gegebene Spalte kann sich an beliebiger Position im Eingabedatensatz befinden, diese muss jedoch angegeben werden, und der Wert Y oder N muss ebenfalls angegeben werden.

### Zugehörige Konzepte:

- „Kurznamen und Datenquellenobjekte“ in *Systeme zusammengeschlüssener Datenbanken*

## Import-API "db2Import"

---

## Kapitel 3. LOAD

In diesem Kapitel wird das DB2 UDB-Dienstprogramm LOAD beschrieben, mit dem Daten aus Dateien, benannten Pipes, Einheiten oder einem Cursor in eine DB2-Tabelle versetzt werden. Diese Datenquellen können sich entweder auf dem Knoten befinden, auf dem sich auch die Datenbank befindet, oder auf einem Client mit Fernverbindung. Die zu ladende Tabelle muss vorhanden sein. Wenn die Tabelle, die die neuen Daten empfängt, bereits Daten enthält, können Sie diese ersetzen oder die neuen Daten an die vorhandenen Daten anfügen.

In diesem Kapitel werden die folgenden Themen behandelt:

- „LOAD - Übersicht“ auf Seite 82
- „Parallelität und Laden“ auf Seite 89
- „Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von LOAD“ auf Seite 90
- „Verwenden von LOAD“ auf Seite 91
- „Ladeoperationen mit Lesezugriff“ auf Seite 93
- „Erstellen von Indizes“ auf Seite 96
- „Verwenden von LOAD mit Identitätsspalten“ auf Seite 98
- „Verwenden von LOAD mit generierten Spalten“ auf Seite 100
- „Überprüfen auf ungültige Integritätsbedingungen“ auf Seite 102
- „Aktualisieren von abhängigen sofort gespeicherten Abfragetabellen“ auf Seite 105
- „Weitergeben von abhängigen sofort gespeicherten Zwischenspeichertabellen“ auf Seite 106
- „Hinweise zum mehrdimensionalen Clustering“ auf Seite 108
- „Erneutes Starten einer unterbrochenen Ladeoperation“ auf Seite 109
- „Wiederherstellen von Daten mit der Datei mit den Angaben zur Speicherposition der Ladekopie“ auf Seite 110
- „Befehl LOAD“ auf Seite 112
- „LOAD QUERY“ auf Seite 137
- „Lade-API "db2Load"“ auf Seite 139
- „Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs“ auf Seite 163
- „LOAD-Ausnahmetabelle“ auf Seite 183
- „LOAD-Speicherauszugsdatei“ auf Seite 184
- „Temporäre Dateien für das Dienstprogramm LOAD“ auf Seite 184
- „Protokollsätze für das Dienstprogramm LOAD“ auf Seite 185
- „Tabellensperren, Statuswerte für Tabellen und Statuswerte für Tabellenbereiche“ auf Seite 186
- „Zeichensatz und NLS“ auf Seite 189
- „Statusangaben für anstehende Aktionen nach einer Ladeoperation“ auf Seite 190
- „Optimieren der Leistung von LOAD“ auf Seite 191
- „CLP-Beispiele für LOAD“ auf Seite 197

Weitere Informationen zum Laden von DB2 Data Links Manager-Daten finden Sie unter „Versetzen von DB2 Data Links Manager-Daten mit IMPORT“ auf Seite 231.

## LOAD - Übersicht

Mit dem Dienstprogramm LOAD können große Datenmengen effizient in neu erstellte Tabellen oder in Tabellen, die bereits Daten enthalten, versetzt werden. Es können die meisten Datentypen verarbeitet werden, einschließlich großer Objekte (LOBs) und benutzerdefinierter Typen (UDTs). Das Dienstprogramm LOAD arbeitet schneller als das Dienstprogramm IMPORT, da es formatierte Seiten direkt in die Datenbank schreibt, während das Dienstprogramm IMPORT die Daten mit SQL-Anweisungen INSERT einfügt. Das Dienstprogramm LOAD startet keine Auslöser und führt außer der Prüfung auf eindeutige Indizes keine Prüfungen auf referenzielle Integrität oder Integritätsbedingungen in Tabellen aus.

Der LOAD-Prozess umfasst vier Einzelphasen (siehe Abb. 1):

- **LOAD.** Dabei werden die Daten in die Tabelle geschrieben.

Während der LOAD-Phase werden die Daten in die Tabelle geladen. Bei Bedarf können Indexschlüssel und Tabellenstatistiken erfasst werden. Sicherungspunkte oder Konsistenzzustände werden in Intervallen eingerichtet, die Sie mit dem Parameter SAVECOUNT des Befehls LOAD angegeben haben. Es werden Nachrichten generiert, durch die Sie erfahren, wie viele Eingabezeilen zum Zeitpunkt des Sicherungspunkts erfolgreich geladen wurden. Für mit FILE LINK CONTROL definierte DATALINK-Spalten werden Verbindungsoperationen für Spaltenwerte ungleich NULL ausgeführt. Wenn ein Fehler auftritt, können Sie die Ladeoperation erneut starten. Mit der Option RESTART wird die Ladeoperation automatisch vom letzten erfolgreichen Konsistenzzustand ausgehend erneut gestartet. Mit der Option TERMINATE wird die fehlgeschlagene Ladeoperation rückgängig gemacht.



Abbildung 1. Die vier Phasen des LOAD-Prozesses: LOAD, BUILD, DELETE und INDEX COPY. Während die Ladeoperation stattfindet, gilt für die Zieltabelle der Status "Laden läuft". Falls für die Tabelle Integritätsbedingungen gelten, befindet sich die Tabelle außerdem im Status "Überprüfung anstehend". Sofern die Option ALLOW READ ACCESS angegeben wurde, gilt außerdem der Status "Nur Lesezugriff" für die Tabelle.

- **BUILD.** Dabei werden die Indizes erstellt.

Während der BUILD-Phase werden auf der Grundlage der während der LOAD-Phase erfassten Indexschlüssel Indizes erstellt. Die Indexschlüssel werden während der LOAD-Phase sortiert, und Indexstatistiken werden erfasst (wenn bei der Option INDEXES der Wert STATISTICS YES angegeben wurde). Die erfassten Statistikdaten sind denen des Befehls RUNSTATS ähnlich. Wenn in der BUILD-Phase ein Fehler auftritt, können Sie die Ladeoperation mit der Option RESTART automatisch vom letzten erfolgreichen Konsistenzzustand ausgehend erneut starten.

- **DELETE.** Dabei werden die Zeilen, die zu Verstößen gegen eindeutige Schlüssel oder einer DATALINK-Verletzung geführt haben, aus der Tabelle gelöscht. Verstöße gegen eindeutige Schlüssel werden in die Ausnahmetabelle eingetragen, sofern eine solche Tabelle angegeben wurde. Nachrichten zu zurückgewiesenen Zeilen werden in die Nachrichtendatei aufgenommen. Führen Sie im Anschluss an den LOAD-Prozess folgende Aktionen aus: Prüfen Sie die Nachrichten, korrigieren Sie eventuelle Fehler, und fügen Sie die korrigierten Zeilen in die Tabelle ein.

Versuchen Sie nicht, temporäre Dateien, die vom Dienstprogramm LOAD erstellt wurden, zu löschen oder zu ändern. Manche temporäre Dateien sind für die DELETE-Phase kritisch. Wenn in der DELETE-Phase ein Fehler auftritt, wird die Ladeoperation mit der Option RESTART automatisch vom letzten erfolgreichen Konsistenzzustand ausgehend erneut gestartet.

**Anmerkung:** Jedes Löschereignis wird protokolliert. Wenn eine größere Zahl von Datensätzen die Eindeutigkeitsbedingung verletzt, kommt es möglicherweise während der DELETE-Phase zu einem Protokollüberlauf.

- INDEX COPY. Hierbei werden die Indexdaten aus einem systemtemporären Tabellenbereich in den ursprünglichen Tabellenbereich kopiert. Dies findet nur dann statt, falls während einer Ladeoperation, bei der die Option READ ACCESS angegeben war, ein systemtemporärer Tabellenbereich für die Indexerstellung angegeben wurde.

**Anmerkung:**

Nachdem das Dienstprogramm LOAD aufgerufen wurde, können Sie den Verarbeitungsfortschritt der Ladeoperation mit Hilfe des Befehls LIST UTILITIES überwachen. Weitere Informationen finden Sie unter LIST UTILITIES Command.

Folgende Informationen werden benötigt, wenn Daten geladen werden:

- Der Pfad und der Name der Eingabedatei, benannten Pipe oder Einheit.
- Der Name oder Aliasname der Zieltabelle.
- Das Format der Eingabequelle. Mögliche Formate sind: DEL, ASC, PC/IXF oder CURSOR.
- Angabe, ob die Eingabedaten an die Tabelle angefügt werden oder die vorhandenen Daten in der Tabelle ersetzen sollen.
- Ein Nachrichtendateiname, wenn das Dienstprogramm über die Anwendungsprogrammierschnittstelle (API) **db2Load** aufgerufen wird.

Sie können auch Folgendes angeben:

- Die zu ladenden Daten befinden sich auf dem Client - sofern das Dienstprogramm LOAD von einem Client mit Fernverbindung aufgerufen wird.
- Die Methode, die zum Laden der Daten verwendet werden soll: Spaltenposition, Spaltenname oder relative Spaltenposition.
- Die Häufigkeit, mit der das Dienstprogramm Konsistenzzustände herstellen soll. Verwenden Sie zur Angabe dieses Wertes den Parameter SAVECOUNT. Wenn dieser Parameter angegeben wird, beginnt eine LOAD RESTART-Operation vom letzten Konsistenzzustand ausgehend und nicht vom Anfang.
- Die Namen der Tabellenspalten, in die die Daten eingefügt werden sollen.
- Die Angabe, ob die in der Tabelle bereits vorhandenen Daten abgefragt werden können oder nicht, während die Ladeoperation läuft.

**Anmerkung:** Zu diesem Zweck kann die Option READ ACCESS verwendet werden. Diese Funktionsweise wird nicht unterstützt, wenn das Dienstprogramm LOAD im Modus REPLACE aufgerufen wird.

- Die Angabe, ob die Ladeoperation warten soll, bis andere Dienstprogramme oder Anwendungen die Verwendung der Tabelle beenden haben, oder vor der Fortsetzung das Beenden der anderen Anwendungen erzwingen soll.

- Einen alternativen systemtemporären Tabellenbereich, in dem der Index erstellt werden soll.

**Anmerkung:** Diese Funktionsweise wird nur unterstützt, wenn die Option READ ACCESS bei einer Wiederherstellung aller Indizes angegeben wird.

- Die Pfade und die Namen der Eingabedateien, in denen LOBs gespeichert sind. Der Änderungswert `lobsinfile` teilt dem Dienstprogramm LOAD mit, dass alle LOB-Daten aus Dateien geladen werden.
- Ein Nachrichtendateiname. Bei DB2®-Operationen wie dem Exportieren, Importieren, Laden, Binden oder Wiederherstellen von Daten können Sie angeben, dass Nachrichtendateien erstellt werden, die Fehlermeldungen, Warnungen und informative Nachrichten zu diesen Operationen enthalten. Geben Sie den Namen dieser Dateien mit dem Parameter MESSAGES an. Diese Nachrichtendateien sind gewöhnliche ASCII-Dateien. Sie können sie mit dem bei Ihrem Betriebssystem üblichen Verfahren drucken oder mit einem beliebigen ASCII-Editor anzeigen.

**Anmerkungen:**

1. Sie können den Inhalt einer Nachrichtendatei erst anzeigen, wenn die Operation beendet ist.
  2. Jede Nachricht in einer Nachrichtendatei beginnt in einer neuen Zeile und enthält Informationen, die von der DB2-Funktion zum Abrufen von Nachrichten bereitgestellt werden.
- Angabe, ob Spaltenwerte, die geladen werden, ein impliziertes Dezimalzeichen haben oder nicht. Der Änderungswert `implieddecimal` teilt dem Dienstprogramm LOAD mit, dass Dezimalzeichen auf die Daten angewendet werden sollen, wenn sie in die Tabelle geschrieben werden. Zum Beispiel wird der Wert 12345 in eine Spalte, für die DECIMAL(8,2) definiert ist, als 123.45 und nicht als 12345.00 geladen.
  - Angabe, ob das Dienstprogramm die Menge des freien Speicherbereichs nach dem Laden einer Tabelle ändern soll. Zusätzlicher freier Speicherbereich ermöglicht ein Anwachsen der Tabelle durch INSERT und UPDATE nach Abschluss einer Ladeoperation. Verkleinerter freier Speicherbereich hält zusammengehörige Zeilen näher zusammen und kann zu einem verbesserten Durchsatz führen.
  - Angabe, ob während des LOAD-Prozesses Statistikdaten zusammengestellt werden. Diese Option wird nur unterstützt, wenn die Ladeoperation im Ersetzungsmodus (REPLACE) ausgeführt wird.

Wenn Daten an eine Tabelle angefügt werden, werden keine Statistikdaten erfasst. Rufen Sie zum Erfassen von aktuellen Statistikdaten zu einer angefügten Tabelle nach Beendigung des Ladeprozesses das Dienstprogramm RUNSTATS auf. Wenn Statistikdaten zu einer Tabelle mit einem eindeutigen Index zusammengestellt und während der DELETE-Phase doppelte Schlüssel gelöscht werden, erfolgt keine Aktualisierung der Statistikdaten zur Berücksichtigung der gelöschten Datensätze. Wenn Sie eine signifikante Anzahl von doppelten Datensätzen erwarten, dürfen Sie während der Ladeoperation keine Statistikdaten erfassen. Rufen Sie stattdessen nach Beendigung des Ladeprozesses das Dienstprogramm RUNSTATS auf.

- Angabe, ob während der Ladeoperation Statistiken erfasst werden sollen. Statistiken werden gemäß dem für die Tabelle definierten Profil erfasst. Das Profil muss vom Befehl RUNSTATS erstellt werden, bevor der Befehl LOAD ausgeführt wird. Ist das Profil nicht vorhanden und die Ladeoperation wird aufgefordert, Statistiken gemäß dem Profil zu erfassen, schlägt die Ladeoperation fehl und es wird eine Fehlermeldung zurückgegeben.

- Angabe, ob eine Kopie der vorgenommenen Änderungen erhalten bleiben soll. Diese wird benötigt, um eine aktualisierende Wiederherstellung der Datenbank zu ermöglichen. Diese Option wird nicht unterstützt, wenn für die Datenbank die aktualisierende Wiederherstellung über Protokolle inaktiviert ist, d. h. wenn die Datenbankkonfigurationsparameter *logretain* und *userexit* inaktiviert sind. Wenn keine Kopie erstellt wird und die aktualisierende Wiederherstellung über Protokolle aktiviert ist, verbleibt der Tabellenbereich nach Beendigung der Ladeoperation im Status "Sicherheit anstehend".

Für vollständig wiederherstellbare Datenbanken ist eine Protokollierung erforderlich. Das Dienstprogramm LOAD schließt die mit dem Laden von Daten verbundene Protokollierung fast völlig aus. Anstelle der Protokollierung haben Sie die Möglichkeit, eine Kopie des geladenen Tabellenteils zu erstellen. Bei einer Datenbankumgebung, die die Datenbankwiederherstellung nach einem Fehler zulässt, haben Sie folgende Möglichkeiten:

- Explizites Anfordern der Erstellung einer Kopie des geladenen Tabellenteils
- Nach Beendigung der Ladeoperation Erstellen einer Sicherung der Tabellenbereiche, in denen sich die Tabelle befindet

Wenn Sie eine Tabelle laden, die bereits Daten enthält, und die Datenbank nicht wiederherstellbar ist, müssen Sie sicherstellen, dass Sie über eine Sicherungskopie der Datenbank oder über die Tabellenbereiche für die geladene Tabelle verfügen, bevor Sie das Dienstprogramm LOAD aufrufen, damit Sie Fehler beheben können.

Wenn Sie eine Folge von mehreren Ladeoperationen für eine wiederherstellbare Datenbank ausführen möchten, erfolgt die Operationsfolge schneller, wenn Sie jede Ladeoperation als nicht wiederherstellbar definieren und am Ende der Ladefolge eine Sicherung erstellen, als wenn Sie jede der Ladeoperationen mit der Option COPY YES aufrufen. Sie können die Option NONRECOVERABLE verwenden, um festzulegen, dass eine Ladetransaktion als nicht wiederherstellbar gekennzeichnet werden soll und dass es nicht möglich sein wird, sie durch eine nachfolgende Operation zur aktualisierenden Wiederherstellung wiederherzustellen. Das Dienstprogramm zur aktualisierenden Wiederherstellung überspringt die Transaktion und kennzeichnet die Tabelle, in die die Daten geladen wurden, als "ungültig". Vom Dienstprogramm werden auch alle nachfolgenden Transaktionen für diese Tabelle ignoriert. Nach Beendigung der aktualisierenden Wiederherstellung kann eine solche Tabelle nur gelöscht werden (siehe Abb. 2). Mit dieser Option werden Tabellenbereiche nach der Ladeoperation nicht in den Status "Sicherheit anstehend" gesetzt, und während der Ladeoperation muss keine Kopie der geladenen Daten erstellt werden.

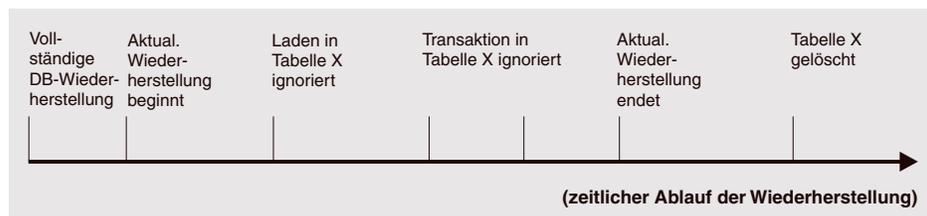


Abbildung 2. Nicht wiederherstellbare Verarbeitung bei einer aktualisierenden Wiederherstellung

- Angabe, ob alle Indexänderungen protokolliert werden sollen. Wird der Datenbankkonfigurationsparameter *logindexbuild* gesetzt und wird die Ladeoperation mit der Wiederherstellbarkeitsoption COPY YES und der Indexierungsoption INCREMENTAL aufgerufen wird, werden beim Laden alle Indexänderungen protokolliert. Der Vorteil bei der Verwendung dieser Optionen liegt darin, dass bei der aktualisierenden Wiederherstellung der Protokollsätze für diese Ladeoperation auch die Indizes wiederhergestellt werden, was normalerweise nur der Fall wäre, wenn beim Laden der Indexierungsmodus REBUILD verwendet worden wäre.
- Den vollständig qualifizierten Pfad, der beim Erstellen von temporären Dateien während einer Ladeoperation verwendet werden soll. Der Name wird mit dem Parameter TEMPPFILES PATH des Befehls LOAD angegeben. Der Standardwert ist der Datenbankpfad. Der Pfad befindet sich auf der Server-Maschine, und es wird darauf ausschließlich vom DB2-Exemplar zugegriffen. Daher muss jede Angabe eines Pfadnamens in diesem Parameter die Verzeichnisstruktur des Servers und nicht des Clients berücksichtigen. Außerdem benötigt der DB2-Exemplareigner Lese- und Schreibzugriff für den Pfad. Das gilt auch, wenn Sie der Exemplareigner sind. Wenn Sie nicht der Exemplareigner sind, müssen Sie eine Speicherposition angeben, für die der Exemplareigner Schreibzugriff besitzt.

## Änderungen der früheren Funktionsweise des Dienstprogramms LOAD (eingeführt in Version 6 und 7)

Die Versionen 6 und 7 des Dienstprogramms LOAD bieten eine uneingeschränkte Abwärtskompatibilität zu früheren Releases, d. h. sie akzeptieren die Syntax aus früheren Releases und funktionieren ordnungsgemäß. Im Folgenden werden die Syntaxänderungen und die Änderungen der Funktionsweise des Dienstprogramms LOAD zusammengefasst, die in Version 6 und Version 7 eingeführt wurden:

- Beim erneuten Starten des Dienstprogramms LOAD wird nicht mehr ein Wert RESTARTCOUNT verwendet. Der Parameter ist jetzt reserviert. Wenn eine zuvor unterbrochene Ladeoperation erneut gestartet wird, wird die Ladeoperation automatisch ausgehend vom letzten Konsistenzzustand in der LOAD-, BUILD- oder DELETE-Phase fortgesetzt.
- Das Sortieren der Indexschlüssel während der Indexerstellung nutzt einen neuen Sortieralgorithmus, der von der Datenbanksteuerkomponente der Version 6 von DB2 UDB verwendet wird. Die für den Sortiervorgang bestimmte Speicherkapazität wird durch den Wert des Datenbankkonfigurationsparameters für den Sortierspeicher (*sortheap*) sowie den Wert des Konfigurationsparameters für den Zwischenspeicher für Sortierlisten (*sheapthres*) des Datenbankmanagers gesteuert. Wenn diese Optionen in Version 6 angegeben werden, wird eine Informationsnachricht mit einer Warnung zurückgegeben, die Ladeoperation jedoch normal fortgesetzt.

Sortierüberläufe, die während der Ladeoperation bei der Indexerstellung auftreten, werden nun in einem temporären Tabellenbereich ausgeführt. Ein Überlauf der Sortiervorgänge findet nicht direkt auf die Platte statt, sondern vielmehr in den Pufferpool, der dem temporären Tabellenbereich zugeordnet ist. Die für die Indexerstellung benötigte Zeit kann verbessert werden, wenn dem temporären Tabellenbereich ein großer Pufferpool zugeordnet wird. Um denselben Typ der E/A-Parallelität zu erzielen, der bei LOAD SORT-Operationen vor Version 6 (durch Angabe mehrerer temporärer Sortierverzeichnisse) verfügbar war, empfiehlt es sich, temporäre Tabellenbereiche mit mehreren Behältern zu deklarieren, die sich jeweils auf unterschiedlichen Platteneinheiten befinden. Es empfiehlt sich außerdem, temporäre Tabellenbereiche als SMS (System Managed Space) zu

deklarieren, damit diese zur Aufnahme von umfangreichen Datenmengen vergrößert werden können, ohne jedoch Plattenressourcen zu belegen, wenn diese nicht verwendet werden.

- Die Option REMOTE FILE wurde umbenannt. (Das Dienstprogramm akzeptiert jedoch weiterhin den Parameter REMOTE FILE bei der Angabe des Pfads von temporären Dateien.) Dies ist eine rein syntaktische Änderung, die Bedeutung und Zweck dieses Parameters besser wiedergeben soll. Der Parameter TEMPFILES PATH verweist auf ein Verzeichnis anstelle einer Datei.
- Das Dienstprogramm LOAD unterstützt jetzt mehrere Indexierungsmodi: REBUILD (Alle Indizes wiederherstellen), INCREMENTAL (Indizes mit neuen Daten erweitern), DEFERRED (Indexaktualisierung verzögern, nach Abschluss einer Ladeoperation) sowie AUTOSELECT (Automatisch auswählen). Der letzte Modus wählt bei Laufzeit zwischen der Wiederherstellung aller Indizes und der Erweiterung der Indizes mit neuen Daten aus. Der Modus REBUILD (Alle Indizes wiederherstellen) spiegelt die Funktionsweise von Releases vor Version 6 wider. Standardmodus für Version 6 ist der Modus AUTOSELECT (Automatisch auswählen).
- In Version 6 kann eine Ladeoperation mit der Option TERMINATE zurückgesetzt werden. Zuvor wurde durch diese Option ein Tabellenbereich in den Status "Wiederherstellung anstehend" gesetzt. Bitte beachten Sie jedoch, dass nach einer fehlgeschlagenen LOAD REPLACE-Operation die Tabellendaten durch eine Anforderung TERMINATE *nicht* wiederhergestellt werden.

Version 7 des Dienstprogramms LOAD kann Daten laden, die sich auf einem Client mit Fernverbindung, in vollständig qualifizierten Pipes oder in benannten Pipes befinden. (Separate Dateien mit LOB-Werten sollten sich auf dem Server befinden, wenn der Änderungswert lobsinfile für den Dateityp angegeben wird.)

## Änderungen der früheren Funktionsweise des Dienstprogramms LOAD (eingeführt in Version 8)

Im Folgenden werden die Syntaxänderungen und die Änderungen der Funktionsweise des Dienstprogramms LOAD zusammengefasst, die in Version 8 eingeführt wurden:

- Vor Version 8 benötigte das Dienstprogramm LOAD einen exklusiven Zugriff auf Tabellenbereiche, die Objekte enthielten, welche zur geladenen Tabelle gehörten. In Version 8 wird das Dienstprogramm LOAD auf Tabellenebene ausgeführt und benötigt keinen exklusiven Zugriff auf den Tabellenbereich mehr. Das Dienstprogramm LOAD sperrt lediglich die Tabellenobjekte, die von einer stattfindenden Ladeoperation betroffen sind. Der gleichzeitige Zugriff auf andere Tabellenobjekte in den gleichen Tabellenbereichen ist zulässig.

**Anmerkung:** Vor Version 8 wurde bei Angabe der Option COPY NO für eine wiederherstellbare Datenbank der Tabellenbereich erst dann in den Status "Sicherung anstehend" versetzt, nachdem die Ladeoperation festgeschrieben wurde. In Version 8 wird der Tabellenbereich in den Status "Sicherung anstehend" versetzt, sobald die Ladeoperation beginnt. Er verbleibt sogar dann in diesem Status, wenn die Ladeoperation fehlschlägt und rückgängig gemacht wird. Wird die Option COPY NO angegeben und die Ladeoperation erfolgreich ausgeführt, setzt das Dienstprogramm zur aktualisierenden Wiederherstellung abhängige Tabellenbereiche - wie auch in früheren Releases - während einer aktualisierenden Wiederherstellung in den Status "Wiederherstellung anstehend".

- Sie können außerdem angeben, dass Benutzern der Lesezugriff auf die Daten, die vor der Ladeoperation in der Tabelle vorhanden waren, möglich sein soll. Dies bedeutet, dass Sie nach Abschluss der Ladeoperation nicht in der Lage sind, die neuen Daten anzuzeigen, falls für die Tabelle Integritätsbedingungen gelten und die Überprüfung der Integritätsbedingungen nicht vollständig ausgeführt wurde. Sie können ebenfalls angeben, dass der Index während einer Ladeoperation in einem separaten Tabellenbereich wiederhergestellt werden soll. Hierzu geben Sie die Optionen READ ACCESS und INDEXING MODE REBUILD an. Der Index wird während der INDEX COPY-Phase, die im Anschluss an die anderen Phasen der Ladeoperation stattfindet, in den ursprünglichen Tabellenbereich zurückkopiert.
- Die Funktionalität des Befehls LOAD QUERY wurde dahingehend erweitert, dass der Befehl zusätzlich zu den Statusinformationen, die zuvor für eine laufende Ladeoperation enthalten waren, jetzt auch den Tabellenstatus des Ziels zurückgibt, in das die Daten geladen werden. Mit dem Befehl LOAD QUERY kann auch der Tabellenstatus abgefragt werden, und zwar unabhängig davon, ob für die betreffende Tabelle gerade eine Ladeoperation ausgeführt wird oder nicht.
- Speicherbereichszuordnungen in DMS-Tabellenbereichen werden jetzt protokolliert. Der Befehl LOAD schreibt nun für jeden Speicherbereich, den er in einem DMS-Tabellenbereich zuordnet, zwei Protokollsätze. Außerdem werden bei Angabe der Optionen READ ACCESS und INDEXING MODE INCREMENTAL einige Protokollsätze geschrieben, während Daten inkrementell in den Index eingefügt werden.
- Abhängige Tabellenbereiche werden vor einer Ladeoperation nicht mehr in den Wartemodus versetzt. Wenn die Option COPY NO angegeben wird, wird der neue Tabellenbereichsstatus *Laden läuft* verwendet. Der Tabellenbereichsstatus "Laden läuft" verhindert, dass abhängige Tabellen während einer Ladeoperation gesichert werden. Dieser Status unterscheidet sich dadurch vom Tabellenstatus "Laden läuft", dass zwar alle Ladeoperationen den Tabellenstatus "Laden läuft" verwenden, aber Ladeoperationen, bei denen die Option COPY NO angegeben wurde, außerdem den Tabellenbereichsstatus "Laden läuft" verwenden.
- Wenn eine Ladeoperation mit den Optionen ALLOW READ ACCESS und INDEXING MODE REBUILD ausgeführt wird, wird zusätzlich zu den ursprünglichen Indizes eine neue Kopie der Indizes erstellt. Dies bedeutet, dass der Speicherbereich, der für den Tabellenbereich für Indizes benötigt wird, möglicherweise verdoppelt werden muss. Um dies zu vermeiden, kann mit der Option USE TABLESPACE ein temporärer Tabellenbereich für das Speichern der neuen Indizes angegeben werden. Nachdem die neuen Indizes im temporären Tabellenbereich erstellt wurden, wird die Zieltabelle in den Offlinemodus versetzt, bevor die neuen Indizes in den Zieltabellenbereich kopiert werden.
- Aufrufe des Befehls LOAD, mit denen Tabellenbereiche in den Wartemodus versetzt werden, wurden entfernt. Wenn Sie Tabellenbereiche im exklusiven Modus vor einer Ladeoperation in den Wartemodus versetzen, müssen Sie nun den exklusiven Wartestatus für die Tabellenbereiche explizit aufheben. In früheren Releases wurden die in den Wartemodus versetzten Tabellenbereiche durch die folgenden Befehle LOAD implizit zurückgesetzt und für andere Anwendungen verfügbar gemacht:

```

quiesce tablespaces for table t1 exclusive
load from data.del of del insert into t1

```

In Version 8 müssen Sie den folgenden Befehl absetzen, um den exklusiven Wartestatus für den Tabellenbereich aufzuheben:

```
quiesce tablespaces for table t1 reset
```

- Der Befehl LOAD wurde durch eine Option LOCK WITH FORCE ergänzt. Mit dieser Option können Sie andere Anwendungen zum Freigeben von Sperren für eine Tabelle zwingen und es der Ladeoperation ermöglichen, fortzufahren und die von ihr benötigten Sperren anzufordern.
- Das Dienstprogramm LOAD kann nun mit Hilfe des neuen Dateityps CURSOR Ladeoperationen aus einer SQL-Anwendung ausführen.
- Das Laden von Daten, die sich auf einem Client mit Fernverbindung befinden, wird jetzt unter den folgenden Bedingungen unterstützt:
  - Die Datenbank, mit der der Client verbunden ist, befindet sich in einer Umgebung mit partitionierten Datenbanken.
  - Die Datenbank, mit der der Client verbunden ist, wird für eine bereits katalogisierte Datenbank katalogisiert.
- Das Laden von Daten in MDC-Tabellen wird unterstützt.
- Vor Version 8 verblieb die Zieltabelle nach einer Ladeoperation im Status "Überprüfung anstehend", wenn sie generierte Spalten enthielt. Das Dienstprogramm LOAD generiert jetzt Spaltenwerte. Daher müssen Sie im Anschluss an eine Ladeoperation nicht mehr die Anweisung SET INTEGRITY absetzen.
- Tabellen können jetzt in eine Umgebung mit partitionierten Datenbanken geladen werden. Hierfür wird nicht mehr das als "Autoloader" bezeichnete Dienstprogramm für automatisches Laden (**db2atld**) benötigt. Die Lade-API (**db2Load**) wurde außerdem erweitert und unterstützt jetzt die Ladeoptionen für partitionierte Datenbanken.

#### Zugehörige Konzepte:

- „Aktualisierende Wiederherstellung“ in *Datenwiederherstellung und hohe Verfügbarkeit Handbuch und Referenz*

#### Zugehörige Referenzen:

- „RUNSTATS Command“ in *Command Reference*
- „LOAD QUERY“ auf Seite 137
- „Befehl LOAD“ auf Seite 112
- „Lade-API "db2Load"“ auf Seite 139
- „LIST UTILITIES Command“ in *Command Reference*

---

## Parallelität und Laden

Das Dienstprogramm LOAD kann die Vorteile einer Hardwarekonfiguration nutzen, in der mehrere Prozessoren oder mehrere Speichereinheiten verwendet werden, wie z. B. in einer symmetrischen Mehrprozessorumgebung (SMP - Symmetric Multiprocessor). Es gibt mehrere Möglichkeiten, wie eine parallele Verarbeitung umfangreicher Datenmengen mit dem Dienstprogramm LOAD durchgeführt werden kann. Eine Möglichkeit besteht in der Verwendung mehrerer Speichereinheiten, die eine E/A-Parallelität während der Ladeoperation ermöglicht (siehe Abb. 3 auf Seite 90). Eine weitere Möglichkeit ist der Einsatz mehrerer Prozessoren in einer SMP-Umgebung, der eine partitionsinterne Parallelität ermöglicht (siehe Abb. 4 auf Seite 90). Beide Möglichkeiten können kombiniert verwendet werden, um ein noch schnelleres Laden der Daten zu erreichen.

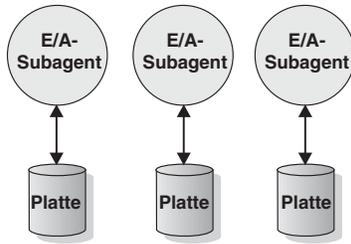


Abbildung 3. E/A-Parallelität beim Laden von Daten

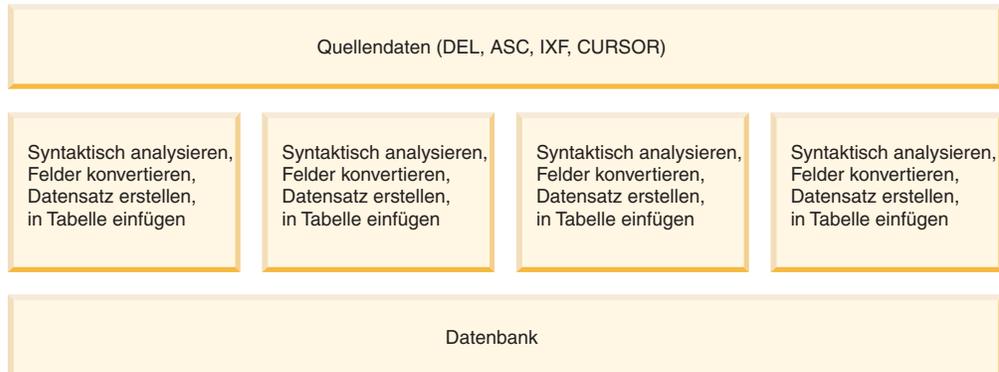


Abbildung 4. Partitionsinterne Parallelität beim Laden von Daten

#### Zugehörige Konzepte:

- „Optimieren der Leistung von LOAD“ auf Seite 191

## Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von LOAD

Sie benötigen eine der folgenden Berechtigungen, damit Sie Daten in eine Tabelle laden können:

- Berechtigung SYSADM
- Berechtigung DBADM
- Berechtigung LOAD für die Datenbank und
  - Zugriffsrecht INSERT für die Tabelle, wenn das Dienstprogramm LOAD im Einfügemodus (INSERT), im Beendigungsmodus (TERMINATE) zur Beendigung einer vorherigen LOAD INSERT-Operation oder im Neustartmodus (RESTART) zum erneuten Starten einer vorherigen LOAD INSERT-Operation aufgerufen wird.
  - Zugriffsrecht INSERT und DELETE für die Tabelle, wenn das Dienstprogramm LOAD im Ersetzungsmodus (REPLACE), im Beendigungsmodus (TERMINATE) zur Beendigung einer vorherigen LOAD REPLACE-Operation oder im Neustartmodus (RESTART) zum erneuten Starten einer vorherigen LOAD REPLACE-Operation aufgerufen wird.
  - Zugriffsrecht INSERT für die Ausnahmetabelle, wenn eine solche Tabelle im Rahmen der Ladeoperation verwendet wird.

Da der Exemplareigner Eigner aller LOAD-Prozesse (und im Allgemeinen auch aller DB2<sup>®</sup>-Server-Prozesse) ist und alle diese Prozesse die ID des Exemplareigners

verwenden, um auf benötigte Dateien zuzugreifen, muss der Exemplareigner über einen Lesezugriff auf Eingabedatendateien verfügen. Diese Eingabedatendateien müssen für den Exemplareigner lesbar sein, unabhängig davon, wer den Befehl aufruft.

Bei den Betriebssystemen Windows® NT, Windows 2000 und Windows.NET, unter denen DB2 als Windows-Service ausgeführt wird, müssen Sie, falls Sie Daten aus Dateien laden, die sich auf einem Netzlaufwerk befinden, den DB2-Service zur Ausführung unter einem Benutzereintrag konfigurieren, der über Lesezugriff auf diese Dateien verfügt.

**Zugehörige Referenzen:**

- „Befehl LOAD“ auf Seite 112
- „Lade-API "db2Load"“ auf Seite 139

---

## Verwenden von LOAD

**Vorbedingungen:**

Bevor Sie das Dienstprogramm LOAD ausführen, müssen Sie mit der Datenbank, in die die Daten geladen werden, verbunden bzw. in der Lage sein, implizit eine Verbindung zur Datenbank herzustellen. Da das Dienstprogramm die Anweisung COMMIT absetzt, sollten Sie vor dem Aufrufen von LOAD alle Transaktionen beenden und alle Sperren aufheben, indem Sie die Anweisung COMMIT oder ROLLBACK ausführen.

Da Daten in der gleichen Reihenfolge geladen werden, wie sie in der Eingabedatei vorliegen (außer bei Verwendung von MDC-Tabellen), müssen sie, wenn eine bestimmte Reihenfolge gewünscht wird, vor dem Durchführen der Ladeoperation sortiert werden.

Wenn Cluster erforderlich sind, müssen die Daten vor dem Laden nach dem Clusterindex sortiert werden. Beim Laden von Daten in MDC-Tabellen ist vor der Ladeoperation keine Sortierung erforderlich, und die Daten werden gemäß der MDC-Tabellendefinition in Gruppen zusammengefasst.

**Einschränkungen:**

Für das Dienstprogramm LOAD gelten die folgenden Einschränkungen:

- Das Laden von Daten in Kurznamen wird nicht unterstützt.
- Das Laden von Daten in typisierte Tabellen oder in Tabellen mit Spalten strukturierten Typs wird nicht unterstützt.
- Das Laden von Daten in deklarierte temporäre Tabellen wird nicht unterstützt.
- Versuche, Tabellen in einem Tabellenbereich im Status "Sicherung anstehend" zu erstellen oder zu löschen, schlagen fehl.
- Sie können keine Daten in eine Datenbank laden, auf die über DB2 Connect oder einen Server einer älteren Version als DB2 Version 2 zugegriffen wird. Optionen, die nur mit diesem Release von DB2 verfügbar sind, können nicht mit einem Server aus einem früheren Release verwendet werden.
- Wenn während einer LOAD REPLACE-Operation ein Fehler auftritt, gehen die Originaldaten in der Tabelle verloren. Behalten Sie eine Kopie der Eingabedaten, damit die Ladeoperation erneut gestartet werden kann.

- Für neu geladene Zeilen werden keine Auslöser aktiviert. Geschäftsregeln, die Auslösern zugeordnet sind, werden vom Dienstprogramm LOAD nicht umgesetzt.

**Prozedur:**

Das Dienstprogramm LOAD kann über den Befehlszeilenprozessor (CLP), das Notizbuch "Laden" in der Steuerzentrale oder eine Anwendungsprogrammierschnittstelle (API) namens **db2Load** aufgerufen werden.

Es folgt ein Beispiel für den Befehl LOAD, der über den CLP abgesetzt wird:

```
db2 load from stafftab.ixf of ixf messages staff.msgs
      insert into userid.staff copy yes use tsm data buffer 4000
```

In diesem Beispiel gilt:

- Alle Warnungen oder Fehlermeldungen werden in die Datei `staff.msgs` gestellt.
- Eine Kopie der vorgenommenen Änderungen wird in Tivoli Storage Manager (TSM, vormals ADSM) gespeichert.
- Bei der Ladeoperation sind viertausend Seiten Pufferspeicherbereich zu verwenden.

Es folgt ein weiteres Beispiel für den Befehl LOAD, der über den CLP abgesetzt wird:

```
db2 load from stafftab.ixf of ixf messages staff.msgs
      tempfiles path /u/myuser replace into staff
```

In diesem Beispiel gilt:

- Die Tabellendaten werden ersetzt.
- Der Parameter `TEMPFILES PATH` wird verwendet, um `/u/myuser` als Serverpfad anzugeben, in den temporäre Dateien geschrieben werden.

**Anmerkung:** In diesen Beispielen werden relative Pfadnamen für die LOAD-Eingabedatei verwendet. Relative Pfadnamen sind nur für Aufrufe eines Clients auf demselben Knoten wie die Datenbank zulässig. Es empfiehlt sich die Verwendung von vollständig qualifizierten Pfadnamen.

So öffnen Sie das Notizbuch **Laden**:

1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis Sie den Ordner **Tabellen** sehen.
2. Klicken Sie den Ordner **Tabellen** an. Alle vorhandenen Tabellen werden auf der rechten Seite des Fensters, im Inhaltsteilfenster, angezeigt.
3. Klicken Sie im Inhaltsteilfenster die gewünschte Tabelle mit Maustaste 2 an, und wählen Sie **Laden** im Kontextmenü aus. Das Notizbuch **Laden** wird geöffnet.

Ausführliche Informationen zur Steuerzentrale finden Sie in der Onlinehilfefunktion der Steuerzentrale.

Nachdem das Dienstprogramm LOAD aufgerufen wurde, können Sie den Verarbeitungsfortschritt der Ladeoperation mit Hilfe des Befehls `LIST UTILITIES` überwachen. Wird eine Ladeoperation entweder im Modus `INSERT`, `REPLACE` oder `RESTART` ausgeführt, steht Unterstützung für eine ausführliche Überwachung des Verarbeitungsfortschritts zur Verfügung.

Setzen Sie den Befehl LIST UTILITIES mit der Option SHOW DETAILS ab, um ausführliche Informationen über die aktuelle LOAD-Phase des Dienstprogramms anzuzeigen. Wird eine Ladeoperation jedoch im Modus TERMINATE ausgeführt, stehen keine Details zur Verfügung. Der Befehl LIST UTILITIES zeigt in diesem Fall lediglich an, dass gerade ein Dienstprogramm LOAD im Modus TERMINATE ausgeführt wird. Weitere Informationen finden Sie unter LIST UTILITIES Command.

LOAD unterliegt keinen Integritätsbedingungen außer den Integritätsbedingungen von UNIQUE. Statt dessen wird die Tabelle zu Beginn der Ladeoperation in den Status CHECK PENDING ("Überprüfung anstehend") versetzt. Nachdem die Ladeoperation vollständig beendet wurde, muss der Status CHECK PENDING für die Tabelle mit Hilfe des Befehls SET INTEGRITY aufgehoben werden.

#### Zugehörige Referenzen:

- „LIST UTILITIES Command“ in *Command Reference*
- „Tivoli Storage Manager“ in *Datenwiederherstellung und hohe Verfügbarkeit Handbuch und Referenz*
- „Befehl LOAD“ auf Seite 112
- „Lade-API "db2Load"“ auf Seite 139

#### Zugehörige Beispiele:

- „tbmove.out -- HOW TO MOVE TABLE DATA (C)“
- „tbmove.sqc -- How to move table data (C)“
- „tbmove.out -- HOW TO MOVE TABLE DATA (C++)“
- „tbmove.sqC -- How to move table data (C++)“

---

## Ladeoperationen mit Lesezugriff

Das Dienstprogramm LOAD bietet zwei Optionen, mit denen der Umfang des Zugriffs gesteuert werden kann, der anderen Anwendungen auf eine geladene Tabelle eingeräumt wird. Die Option ALLOW NO ACCESS sperrt die Tabelle exklusiv und ermöglicht keinen Zugriff auf die Tabellendaten, während die Tabelle geladen wird. Dies ist das Standardverhalten. Die Option ALLOW READ ACCESS hingegen verhindert jeglichen Schreibzugriff auf die Tabelle durch andere Anwendungen, lässt jedoch einen Lesezugriff auf zuvor geladene Daten zu. Der folgende Abschnitt behandelt die Option ALLOW READ ACCESS.

Tabellendaten und Indexdaten, die vor dem Beginn einer Ladeoperation vorhanden waren, stehen für Abfragen zur Verfügung, während die Ladeoperation läuft. Hierzu die folgenden Beispiele:

1. Erstellen Sie eine Spalte mit einer Spalte des Typs INTEGER:

```
create table ED (ed int)
```

2. Laden Sie drei Zeilen:

```
load from File1 of del insert into ED
...
Anzahl gelesener Zeilen           = 3
Anzahl übersprungener Zeilen     = 0
Anzahl geladener Zeilen          = 3
Anzahl zurückgewiesener Zeilen   = 0
Anzahl gelöschter Zeilen         = 0
Anzahl festgeschriebener Zeilen  = 3
```

3. Führen Sie eine Abfrage der Tabelle aus:

```
select * from ED
```

```
ED  
-----
```

```
1  
2  
3
```

3 Satz/Sätze ausgewählt.

4. Führen Sie eine Ladeoperation mit der Option ALLOW READ ACCESS aus, und laden Sie zwei weitere Datenzeilen:

```
load from File2 of del insert into ED allow read access
```

5. Gleichzeitig fragen Sie die Tabelle über eine andere Verbindung ab, während die Ladeoperation läuft:

```
select * from ED
```

```
ED  
-----
```

```
1  
2  
3
```

3 Satz/Sätze ausgewählt.

6. Warten Sie, bis die Ladeoperation abgeschlossen ist, und fragen Sie anschließend die Tabelle ab:

```
select * from ED
```

```
ED  
-----
```

```
1  
2  
3  
4  
5
```

5 Satz/Sätze ausgewählt.

Die Option ALLOW READ ACCESS ist sehr nützlich, wenn große Datenmengen geladen werden, weil sie den Benutzern jederzeit - sogar bei einer laufenden Ladeoperation oder nach einer fehlgeschlagenen Ladeoperation - einen Zugriff auf die Tabellendaten ermöglicht. Das Verhalten einer Ladeoperation im Modus "Lesezugriff zulassen" (ALLOW READ ACCESS) ist unabhängig von der Isolationsstufe der Anwendung. Dies bedeutet, dass Eingabeprogramme mit einer beliebigen Isolationsstufe die bereits vorhandenen Daten jederzeit lesen können, jedoch nicht in der Lage sind, die neu geladenen Daten zu lesen, bevor die Ladeoperation abgeschlossen ist.

Der Lesezugriff ist - mit Ausnahme der unmittelbaren Schlussphase - während der gesamten Ladeoperation möglich. Vor dem Festschreiben der Daten fordert das Dienstprogramm LOAD eine exklusive Sperre des Typs "Z" (Super Exclusive Lock) für die Tabelle an. Das Dienstprogramm LOAD wartet, bis alle Anwendungen, die Sperren für die Tabelle erhalten haben, diese freigeben. Hierdurch kann sich vor dem Festschreiben der Daten möglicherweise eine Verzögerung ergeben. Durch die Verwendung der Option LOCK WITH FORCE kann die Beendigung von Anwendungen, die Konflikte verursachen, erzwungen und so eine Fortsetzung der Ladeoperation ohne Verzögerung ermöglicht werden.

Normalerweise fordert eine Ladeoperation im Modus "Lesezugriff zulassen" (ALLOW READ ACCESS) die exklusive Sperre für einen kurzen Zeitraum an. Wenn jedoch die Option USE <tabellenbereichsname> angegeben wird, bleibt die exklusive Sperre für die gesamte Dauer der INDEX COPY-Phase bestehen.

#### **Anmerkungen:**

1. Wenn eine Ladeoperation abgebrochen wird, behält sie dieselbe Zugriffsebene bei, die beim Absetzen der Ladeoperation angegeben war. Wenn also eine Ladeoperation im Modus "Keinen Zugriff zulassen" (ALLOW NO ACCESS) abgebrochen wird, ist ein Zugriff auf die Tabellendaten erst dann möglich, nachdem ein Befehl LOAD TERMINATE oder LOAD RESTART abgesetzt wurde. Wird eine Ladeoperation im Modus "Lesezugriff zulassen" (ALLOW READ ACCESS) abgebrochen, sind die zuvor geladenen Tabellendaten weiterhin für einen Lesezugriff verfügbar.
2. Falls die Option ALLOW READ ACCESS für eine abgebrochene Ladeoperation angegeben war, kann sie auch für die LOAD RESTART- oder LOAD TERMINATE-Operation angegeben werden. War für die abgebrochene Ladeoperation jedoch die Option ALLOW NO ACCESS angegeben, kann die Option ALLOW READ ACCESS weder für die LOAD RESTART- noch für die LOAD TERMINATE-Operation angegeben werden.

In den folgenden Fällen wird die Option ALLOW READ ACCESS nicht unterstützt:

- Die Option REPLACE ist angegeben. Da eine LOAD REPLACE-Operation die vorhandenen Tabellendaten abschneidet, bevor die neuen Daten geladen werden, können erst dann vorab vorhandene Daten abgefragt werden, nachdem die Ladeoperation abgeschlossen ist.
- Die Indizes wurden als ungültig gekennzeichnet, und ihre Wiederherstellung steht an. Indizes können in bestimmten Szenarien für eine aktualisierende Wiederherstellung oder über die Verwendung des Befehls **db2dart** als ungültig gekennzeichnet werden.
- Die Option INDEXING MODE DEFERRED ist angegeben. Dieser Modus gibt an, dass eine Wiederherstellung der Indizes erforderlich ist.
- Eine Ladeoperation mit der Option ALLOW NO ACCESS wird erneut gestartet oder beendet. Eine Ladeoperation im Modus "Lesezugriff zulassen" (ALLOW READ ACCESS) kann erst dann für die Tabelle stattfinden, nachdem diese vollständig in den Onlinestatus versetzt wurde.
- Eine Ladeoperation findet für eine Tabelle statt, deren Status "Überprüfung anstehend" lautet und die sich nicht im Status "Lesezugriff" befindet. Dies gilt auch bei mehreren Ladeoperationen für Tabellen mit Integritätsbedingungen. Eine Tabelle wird erst dann in den Onlinestatus versetzt, nachdem die Anweisung SET INTEGRITY abgesetzt wurde.

Wenn sich Tabellendaten im Offlinestatus befinden, ist ein Lesezugriff während einer Ladeoperation generell erst dann möglich, wenn die Tabelle wieder im Onlinestatus steht.

#### **Zugehörige Konzepte:**

- „Überprüfen auf ungültige Integritätsbedingungen“ auf Seite 102
- „Tabellensperren, Statuswerte für Tabellen und Statuswerte für Tabellenbereiche“ auf Seite 186
- „Erstellen von Indizes“ auf Seite 96

---

## Erstellen von Indizes

Indizes werden im Rahmen der BUILD-Phase einer Ladeoperation erstellt. Es gibt vier Indexierungsmodi, die im Befehl LOAD angegeben werden können:

1. REBUILD. Alle Indizes werden erneut erstellt.
2. INCREMENTAL. Indizes werden mit neuen Daten erweitert.
3. AUTOSELECT. Das Dienstprogramm LOAD wählt automatisch den Modus REBUILD oder INCREMENTAL aus. Dies ist der Standardwert.

**Anmerkung:** In bestimmten Fällen kann es sinnvoll sein, einen Indexierungsmodus explizit auszuwählen, da das Verhalten der Modi REBUILD und INCREMENTAL relativ unterschiedlich ist.

4. DEFERRED. Das Dienstprogramm LOAD versucht nicht, Indizes zu erstellen, wenn dieser Modus angegeben wird. Die Indizes werden als aktualisierungsbedürftig gekennzeichnet, und möglicherweise wird eine erneute Erstellung erzwungen, wenn zum ersten Mal auf die Indizes zugegriffen wird. Diese Option ist nicht mit der Option ALLOW READ ACCESS kompatibel, weil sie die Indizes nicht verwaltet, Suchfunktionen für Indizes jedoch einen gültigen Index erfordern.

Bei Ladeoperationen mit angegebener Option ALLOW READ ACCESS müssen - abhängig vom gewählten Indexierungsmodus - besondere Gegebenheiten hinsichtlich der Speicherauslastung und Protokollierung berücksichtigt werden. Wenn die Option ALLOW READ ACCESS angegeben ist, hält das Dienstprogramm LOAD Indizes sogar während ihrer Wiederherstellung für Abfragen zur Verfügung.

Gibt eine Ladeoperation im Modus "Lesezugriff zulassen" (ALLOW READ ACCESS) die Option INDEXING MODE INCREMENTAL an, schreibt das Dienstprogramm LOAD einige Protokollsätze, die die Integrität der Indexstruktur schützen. Die Anzahl der geschriebenen Protokollsätze entspricht einem Bruchteil der Anzahl der eingefügten Schlüssel. Sie ist erheblich kleiner, als dies bei einer ähnlichen SQL-Operation INSERT erforderlich wäre. Eine Ladeoperation im Modus "Keinen Zugriff zulassen" (ALLOW NO ACCESS) mit angegebener Option INDEXING MODE INCREMENTAL schreibt neben den üblichen Protokollen zur Speicherbereichszuordnung nur einen kleinen Protokollsatz.

Wenn eine Ladeoperation im Modus "Lesezugriff zulassen" (ALLOW READ ACCESS) die Option INDEXING MODE REBUILD angibt, werden neue Indizes als *Schattenindizes* erstellt, und zwar entweder im gleichen Tabellenbereich wie der ursprüngliche Index oder in einem systemtemporären Tabellenbereich. Die ursprünglichen Indizes bleiben erhalten und sind während der Ladeoperation verfügbar. Sie werden erst am Ende der Ladeoperation durch die neuen Indizes ersetzt, während die Tabelle exklusiv gesperrt ist. Falls die Ladeoperation fehlschlägt und die Transaktion zurückgesetzt wird, bleiben die ursprünglichen Indizes erhalten.

### **Erstellen von neuen Indizes im gleichen Tabellenbereich wie der ursprüngliche Index**

In der Standardeinstellung wird der *Schattenindex* im gleichen Tabellenbereich wie der ursprüngliche Index erstellt. Da der ursprüngliche Index und der neue Index gleichzeitig verwaltet werden, muss ausreichend Tabellenbereich für die gleichzeitige Aufnahme beider Indizes vorhanden sein. Wenn die Ladeoperation abgebrochen wird, wird der zusätzliche Bereich, der zur Erstellung des neuen Indexes ver-

wendet wird, freigegeben. Beim Festschreiben der Ladeoperation wird der Bereich, der für den ursprünglichen Index verwendet wurde, freigegeben, und der neue Index wird zum aktuellen Index. Wenn neue Indizes im gleichen Tabellenbereich wie die ursprünglichen Indizes erstellt werden, findet das Ersetzen der ursprünglichen Indizes fast sofort statt.

Werden die Indizes in einem DMS-Tabellenbereich erstellt, ist der neue *Schattenindex* für den Benutzer nicht erkennbar. Wenn die Indizes in einem SMS-Tabellenbereich erstellt werden, kann der Benutzer Indextabellen im Tabellenbereichsverzeichnis mit dem Suffix `.IN1` und dem Suffix `.INX` erkennen. Diese Suffixe lassen nicht darauf schließen, welches der ursprüngliche Index ist und welches der *Schattenindex* ist.

### **Erstellen von neuen Indizes in einem systemtemporären Tabellenbereich**

Der neue Index kann in einem systemtemporären Tabellenbereich erstellt werden. Auf diese Weise kann verhindert werden, dass im eigentlichen Tabellenbereich nicht genügend Speicher zur Verfügung steht. Mit der Option `USE <tabellenbereichsname>` können die Indizes bei Verwendung der Optionen `INDEXING MODE REBUILD` und `ALLOW READ ACCESS` in einem systemtemporären Tabellenbereich wiederhergestellt werden. Der systemtemporäre Tabellenbereich kann ein SMS-Tabellenbereich oder ein DMS-Tabellenbereich sein. Die Seitengröße des systemtemporären Tabellenbereichs muss jedoch mit der Seitengröße des Tabellenbereichs für den ursprünglichen Index identisch sein.

Die Option `USE <tabellenbereichsname>` wird ignoriert, wenn die Ladeoperation nicht im Modus "Lesezugriff zulassen" (`ALLOW READ ACCESS`) ausgeführt wird oder wenn der Indexierungsmodus nicht kompatibel ist. Die Option `USE <tabellenbereichsname>` wird nur bei den Optionen `INDEXING MODE REBUILD` oder `INDEXING MODE AUTOSELECT` unterstützt. Wenn die Option `INDEXING MODE AUTOSELECT` angegeben ist und das Dienstprogramm `LOAD` die inkrementelle Verwaltung der Indizes auswählt, wird die Option `USE <tabellenbereichsname>` ignoriert.

Eine `LOAD RESTART`-Operation kann einen alternativen Tabellenbereich zur Indexerstellung sogar dann einsetzen, wenn die ursprüngliche Ladeoperation keinen alternativen Tabellenbereich verwendete. Eine `LOAD RESTART`-Operation kann nicht im Modus "Lesezugriff zulassen" (`ALLOW READ ACCESS`) abgesetzt werden, wenn die ursprüngliche Ladeoperation nicht im Modus "Lesezugriff zulassen" abgesetzt wurde. `LOAD TERMINATE`-Operationen führen keine Indexwiederherstellung durch. Daher wird die Option `USE <tabellenbereichsname>` ignoriert.

Während der `BUILD`-Phase der Ladeoperation werden die Indizes im systemtemporären Tabellenbereich erstellt. Anschließend wird der Index während der `INDEX COPY`-Phase aus dem systemtemporären Tabellenbereich in den Tabellenbereich des ursprünglichen Indexes kopiert. Um zu gewährleisten, dass im Tabellenbereich des ursprünglichen Indexes genug Speicherbereich für den neuen Index vorhanden ist, wird während der `BUILD`-Phase im ursprünglichen Tabellenbereich Speicher zugeordnet. Wenn also bei einer Ladeoperation nicht genügend Speicherbereich für den Index vorhanden ist, dann tritt dies während der `BUILD`-Phase ein. In einem solchen Fall geht der ursprüngliche Index nicht verloren.

Die `INDEX COPY`-Phase schließt sich an die `BUILD`- und `DELETE`-Phase an. Bevor die `INDEX COPY`-Phase beginnt, wird die Tabelle exklusiv gesperrt. Dies bedeutet, dass sie während der gesamten `INDEX COPY`-Phase nicht im Lesezugriff verfügbar

ist. Da die INDEX COPY-Phase eine physische Kopie erstellt, kann es sein, dass die Tabelle über einen beträchtlichen Zeitraum nicht verfügbar ist.

**Anmerkung:** Wenn es sich entweder bei dem systemtemporären Tabellenbereich oder beim Tabellenbereich für den Index um einen DMS-Tabellenbereich handelt, kann das Lesen aus dem systemtemporären Tabellenbereich eine wahlfreie Ein-/Ausgabe für den systemtemporären Tabellenbereich bewirken und zu einer Verzögerung führen. Das Schreiben in den Tabellenbereich für den Index wird weiterhin optimiert, und die Werte DISK\_PARALLELISM werden verwendet.

**Zugehörige Konzepte:**

- „LOAD - Übersicht“ auf Seite 82
- „Ladeoperationen mit Lesezugriff“ auf Seite 93

---

## Verwenden von LOAD mit Identitätsspalten

Mit dem Dienstprogramm LOAD können Sie Daten in eine Tabelle laden, die eine Identitätsspalte enthält. Sofern keine identitätsbezogenen Änderungswerte für den Dateityp verwendet werden, liegen der Ausführung des Dienstprogramms die folgenden Regeln zugrunde:

- Ist die Identitätsspalte als GENERATED ALWAYS definiert, wird für eine Tabellenzeile immer dann ein Identitätswert generiert, wenn die entsprechende Zeile in der Eingabedatei keinen Wert für die Identitätsspalte enthält oder explizit einen Nullwert vorgibt. Ist für die Identitätsspalte ein Wert angegeben, der kein Nullwert ist, wird die Zeile zurückgewiesen (SQL3550W).
- Ist die Identitätsspalte als GENERATED BY DEFAULT definiert, verwendet das Dienstprogramm LOAD benutzerdefinierte Werte, sofern diese vorhanden sind. Wenn diese Daten fehlen oder explizit Nullwerte sind, wird ein Wert generiert.

Das Dienstprogramm LOAD führt keine zusätzliche Gültigkeitsprüfung der benutzerdefinierten Identitätswerte durch, die über die übliche Gültigkeitsprüfung für Werte mit dem Datentyp der Identitätsspalte (also SMALLINT, INT, BIGINT oder DECIMAL) hinausgeht. Falls Werte doppelt vorhanden sind, wird keine entsprechende Meldung ausgegeben.

Für partitionierte Datenbanken gilt: Wenn sich im Partitionierungsschlüssel für eine Tabelle eine Identitätsspalte befindet oder in einer generierten Spalte, die Teil des Partitionierungsschlüssel ist, auf die Identitätsspalte verwiesen wird und der Änderungswert `identityoverride` nicht angegeben wird, ist die Ladeoperation RESTART nicht zulässig, wenn nicht alle Ladepartitionen von der "LOAD-Phase" aus erneut gestartet werden. Eine solche Ladeoperation ist nicht zulässig, da sich das Hashverfahren für Zeilen während des erneut gestarteten Ladens möglicherweise vom Hashverfahren des einleitenden Ladens aufgrund der Abhängigkeit von der Identitätsspalte unterscheidet. In diesem Fall müssen Sie normalerweise die LOAD-Option TERMINATE verwenden, um das Laden zu beenden.

Das Dienstprogramm LOAD unterstützt drei (sich gegenseitig ausschließende) Änderungswerte für den Dateityp, die die Verwendung dieses Dienstprogramms für Tabellen mit einer Identitätsspalte vereinfachen:

- Der Änderungswert `identitymissing` vereinfacht das Laden einer Tabelle mit einer Identitätsspalte, wenn die Eingabedatendatei für die Identitätsspalte keine

Werte (auch keine Nullwerte) enthält. Angenommen, es wurde beispielsweise eine Tabelle mit der folgenden SQL-Anweisung definiert:

```
create table table1 (c1 varchar(30),
                    c2 int generated by default as identity,
                    c3 decimal(7,2),
                    c4 char(1))
```

Ein Benutzer möchte unter Umständen Daten aus einer Datei (load.del) in die Tabelle TABLE1 laden und diese Datei wurde aus einer Tabelle exportiert, die keine Identitätsspalte enthält. Es folgt ein Beispiel für eine solche Datei.

```
Robert, 45.2, J
Mike, 76.9, K
Leo, 23.4, I
```

Eine Methode für das Laden dieser Datei wäre das explizite Auflisten der zu ladenden Spalten durch den folgenden Befehl LOAD:

```
db2 load from load.del of del replace into table1 (c1, c3, c4)
```

Bei einer Tabelle mit vielen Spalten ist die Verwendung dieser Syntax jedoch eventuell umständlich und fehlerträchtig. Eine alternative Methode für das Laden der Datei ist die folgende Verwendung des Änderungswerts `identitymissing` für den Dateityp:

```
db2 load from load.del of del modified by identitymissing replace into table1
```

- Der Änderungswert `identityignore` stellt in gewisser Hinsicht das Gegenteil des Änderungswerts `identitymissing` dar: Er weist das Dienstprogramm LOAD an, dass die in der Eingabedatendatei vorhandenen Werte für die Identitätsspalte ignoriert werden sollen und dass für jede Zeile ein Identitätswert generiert werden soll. Beispiel: Ein Benutzer möchte die wie oben definierte Tabelle TABLE1 aus einer Datendatei (load.del) mit den folgenden Daten laden:

```
Robert, 1, 45.2, J
Mike, 2, 76.9, K
Leo, 3, 23.4, I
```

Wenn die benutzerdefinierten Werte 1, 2 und 3 nicht für die Identitätsspalte verwendet werden sollen, könnte der Benutzer den folgenden Befehl LOAD absetzen:

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

Aber auch diese Methode kann möglicherweise umständlich und fehlerträchtig sein, wenn die Tabelle zu viele Spalten enthält. Der Änderungswert `identityignore` vereinfacht die Syntax folgendermaßen:

```
db2 load from load.del of del modified by identityignore replace into table1
```

- Der Änderungswert `identityoverride` wird verwendet, um benutzerdefinierte Werte in eine Tabelle mit einer als `GENERATED ALWAYS` definierten Identitätsspalte zu laden. Dies kann bei der Migration von Daten aus einem anderen Datenbanksystem, bei der die Tabelle als `GENERATED ALWAYS` definiert sein muss, oder beim Laden einer Tabelle aus Daten nützlich sein, die mit der Option `DROPPED TABLE RECOVERY` des Befehls `ROLLFORWARD DATABASE` wiederhergestellt wurden. Bei Verwendung dieses Änderungswerts werden alle Zeilen ohne Daten (oder mit Nulldaten) für die Identitätsspalte zurückgewiesen (SQL3116W).

**Anmerkung:** Die Verwendung dieses Änderungswerts kann das Eindeutigkeitsmerkmal von Spalten verletzen, die als `GENERATED ALWAYS` definiert sind.

### Zugehörige Konzepte:

- „Identitätsspalten“ in *Systemverwaltung: Konzept*

---

## Verwenden von LOAD mit generierten Spalten

Mit dem Dienstprogramm LOAD können Sie Daten in eine Tabelle laden, die generierte Spalten enthält, bei denen es sich nicht um Identitätsspalten handelt. Die Spaltenwerte werden durch dieses Dienstprogramm generiert.

**Anmerkung:** Falls Sie eine Ladeoperation zwischen einem Client der Version 7 (oder früher) und einem Server der Version 8 (oder höher) einleiten, versetzt das Dienstprogramm LOAD Tabellen mit generierten Spalten in den Status "Überprüfung anstehend".

Falls eine Tabelle in den Status "Überprüfung anstehend" versetzt wurde, weil zum Laden von Daten in eine Tabelle mit generierten Spalten ein Client der Version 7 (oder früher) verwendet wurde, können Sie mit dem folgenden Befehl den Status "Überprüfung anstehend" für die Tabelle aufheben und die Generierung von Werten erzwingen:

```
SET INTEGRITY FOR tabellenname IMMEDIATE CHECKED FORCE GENERATED;
```

Sofern keine Änderungswerte für den Dateityp verwendet werden, die sich auf generierte Spalten beziehen, liegen der Ausführung des Dienstprogramms LOAD die folgenden Regeln zugrunde:

- Für generierte Spalten werden Werte erstellt, wenn in der entsprechende Zeile der Datendatei ein Wert für die Spalte fehlt oder ein Nullwert angegeben ist. Ist für eine generierte Spalte ein Wert angegeben, der kein Nullwert ist, wird die Zeile zurückgewiesen (SQL3550W).
- Wenn für eine generierte Spalte, die keine Nullwerte enthalten kann, ein Nullwert erstellt wird, wird die gesamte Datenzeile zurückgewiesen (SQL0407N). Dies könnte beispielsweise dann eintreten, wenn eine generierte Spalte, die keinen Nullwert enthalten kann, als Summe zweier Tabellenspalten definiert ist und diese Spalten in der Datendatei Nullwerte enthalten.

Das Dienstprogramm LOAD unterstützt drei (sich gegenseitig ausschließende) Änderungswerte für den Dateityp, die die Verwendung dieses Dienstprogramms für Tabellen mit generierten Spalten vereinfachen:

- Der Änderungswert `generatedmissing` vereinfacht das Laden einer Tabelle mit generierten Spalten, wenn die Eingabedatendatei für alle in der Tabelle vorhandenen generierten Spalten keine Werte (auch keine Nullwerte) enthält. Angenommen, es wurde beispielsweise eine Tabelle mit der folgenden SQL-Anweisung definiert:

```
create table table1 (c1 int,  
                    c2 int,  
                    g1 int generated always as (c1 + c2),  
                    g2 int generated always as (2 * c1),  
                    c3 char(1))
```

Ein Benutzer möchte unter Umständen Daten aus einer Datei (`load.del`) in die Tabelle `TABLE1` laden und diese Datei wurde aus einer Tabelle exportiert, die keine generierten Spalten enthält. Es folgt ein Beispiel für eine solche Datei.

```
1, 5, J  
2, 6, K  
3, 7, I
```

Eine Methode für das Laden dieser Datei wäre das explizite Auflisten der zu ladenden Spalten durch den folgenden Befehl LOAD:

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

Bei einer Tabelle mit vielen Spalten ist die Verwendung dieser Syntax jedoch eventuell umständlich und fehlerträchtig. Eine alternative Methode für das Laden der Datei ist die folgende Verwendung des Änderungswerts `generatedmissing` für den Dateityp:

```
db2 load from load.del of del modified by generatedmissing replace into table1
```

- Der Änderungswert `generatedignore` stellt in gewisser Hinsicht das Gegenteil des Änderungswerts `generatedmissing` dar: Er weist das Dienstprogramm LOAD an, dass die in der Eingabedatendatei vorhandenen Werte für alle in der Zieltabelle vorhandenen generierten Spalten ignoriert und in jede generierte Spalte die berechneten Werte geladen werden sollen. Beispiel: Ein Benutzer möchte die wie oben definierte Tabelle TABLE1 aus einer Datendatei (load.del) mit den folgenden Daten laden:

```
1, 5, 10, 15, J
2, 6, 11, 16, K
3, 7, 12, 17, I
```

Die benutzerdefinierten Werte 10, 11, und 12 (für g1) sowie 15, 16 und 17 (für g2), die keine Nullwerte sind, bewirken, dass die Zeile zurückgewiesen wird (SQL3550W). Um dies zu verhindern, könnte der Benutzer den folgenden Befehl LOAD absetzen:

```
db2 load from load.del of del method P(1, 2, 5)
replace into table1 (c1, c2, c3)
```

Aber auch diese Methode kann möglicherweise umständlich und fehlerträchtig sein, wenn die Tabelle zu viele Spalten enthält. Der Änderungswert `generatedignore` vereinfacht die Syntax folgendermaßen:

```
db2 load from load.del of del modified by generatedignore replace into table1
```

- Der Änderungswert `generatedoverride` wird verwendet, um benutzerdefinierte Werte in eine Tabelle mit generierten Spalten zu laden. Dies kann bei der Migration von Daten aus einem anderen Datenbanksystem nützlich sein oder beim Laden einer Tabelle aus Daten, die mit der Option RECOVER DROPPED TABLE des Befehls ROLLFORWARD DATABASE wiederhergestellt wurden. Bei Verwendung dieses Änderungswerts werden alle Zeilen ohne Daten (oder mit Null-daten) für generierte Spalten, die keine Nullwerte enthalten können, zurückgewiesen (SQL3116W).

Bei Verwendung dieses Änderungswerts wird die Tabelle nach der Ladeoperation in den Status "Überprüfung anstehend" versetzt. Um den Status "Überprüfung anstehend" für die Tabelle aufzuheben, ohne die benutzerdefinierten Werte zu überprüfen, setzen Sie den folgenden Befehl ab:

```
SET INTEGRITY FOR tabellenname GENERATED COLUMN IMMEDIATE
UNCHECKED
```

Um den Status "Überprüfung anstehend" für die Tabelle aufzuheben und die Überprüfung der benutzerdefinierten Werte zu erzwingen, setzen Sie den folgenden Befehl ab:

```
SET INTEGRITY FOR tabellenname IMMEDIATE CHECKED.
```

Das Dienstprogramm LOAD unterstützt das Laden von Tabellen mit generierten Spalten im Partitionierungsschlüssel. Bei diesen generierten Spalten müssen die Daten für die abhängigen Spalten innerhalb der ersten 32 KB an Daten für jede geladene Zeile angezeigt werden.

Angenommen, es wurde beispielsweise eine Tabelle mit der folgenden SQL-Anweisung erstellt:

```
create table table1 (c1 int, c2 int, g1 int generated always as (c1 + c2))
partitioning key (g1)
```

Um Daten erfolgreich in diese Tabelle zu laden, müssen sich sämtliche Daten für die Spalten c1 und c2 in den ersten 32 KB der einzelnen geladenen Zeilen befinden. Jede Zeile, die diese Voraussetzung nicht erfüllt, wird zurückgewiesen.

**Anmerkung:** In einem Fall unterstützt LOAD Werte von generierten Spalten NICHT, und zwar dann nicht, wenn einer der Ausdrücke von generierten Spalten eine benutzerdefinierte Funktion enthält, die abgeschirmt (FENCED) ist. Wird versucht, Daten in eine solche Tabelle zu laden, schlägt das Dienstprogramm fehl. Sie können jedoch für diese Typen generierter Spalten Ihre eigenen Werte angeben, indem Sie den Änderungswert für Dateityp generatedoverride von LOAD verwenden.

**Zugehörige Konzepte:**

- „Generated Columns“ in *Application Development Guide: Programming Client Applications*

---

## Überprüfen auf ungültige Integritätsbedingungen

Nach einer Ladeoperation kann sich die geladene Tabelle unter Umständen im Status "Überprüfung anstehend" (Modus READ für "Lesezugriff" oder NO ACCESS für "Kein Zugriff") befinden, wenn eine der folgenden Bedingungen zutrifft:

- Für die Tabelle sind Prüfungen auf Integritätsbedingungen oder referenzielle Integritätsbedingungen definiert.
- In der Tabelle sind DATALINK-Spalten definiert.
- Die Tabelle enthält generierte Spalten, und die Ladeoperation wurde mit einem Client der Version 7 oder früher eingeleitet.
- Der Tabelle sind sofort gespeicherte Abfragetabellen oder sofort gespeicherte Zwischenspeichertabellen untergeordnet, die auf sie verweisen.
- Die Tabelle ist eine Zwischenspeichertabelle oder eine gespeicherte Abfragetabelle.

Die Markierung STATUS des Eintrags für die geladene Tabelle in SYSCAT.TABLES gibt den Status "Überprüfung anstehend" der Tabelle an. Damit die geladene Tabelle vollumfänglich verwendbar ist, muss für STATUS der Wert N und für ACCESS MODE der Wert F eingetragen sein. Diese Werte geben an, dass der Zugriff auf die Tabelle vollständig möglich ist und dass die Tabelle einen normalen Status aufweist.

Falls der geladenen Tabelle andere Tabellen untergeordnet sind, kann mit dem Parameter CHECK PENDING CASCADE angegeben werden, ob der Status "Überprüfung anstehend" der geladenen Tabelle direkt an die untergeordneten Tabellen weitergegeben werden soll.

Wenn für die Tabelle sowohl Integritätsbedingungen als auch untergeordnete Fremdschlüsseltabellen, abhängige gespeicherte Abfragetabellen sowie abhängige Zwischenspeichertabellen definiert sind und alle Tabellen vor der Ladeoperation einen normalen Status aufweisen, entstehen - je nach angegebenen Ladeparametern - die folgenden Ergebnisse:

**INSERT, ALLOW READ ACCESS und CHECK PENDING CASCADE IMMEDIATE** Die geladene Tabelle, ihre abhängigen gespeicherten Abfragetabellen und ihre abhängigen Zwischenspeichertabellen werden in den Status "Überprüfung anstehend" mit Lesezugriff versetzt.

**INSERT, ALLOW READ ACCESS und CHECK PENDING CASCADE DEFERRED** Nur die geladene Tabelle wird in den Status "Überprüfung anstehend" mit Lesezugriff versetzt. Untergeordnete Fremdschlüsseltabellen, untergeordnete gespeicherte Abfragetabellen und untergeordnete Zwischenspeichertabellen behalten ihren ursprünglichen Status bei.

**INSERT, ALLOW NO ACCESS und CHECK PENDING CASCADE IMMEDIATE** Die geladene Tabelle, ihre abhängigen gespeicherten Abfragetabellen und ihre abhängigen Zwischenspeichertabellen werden in den Status "Überprüfung anstehend, kein Zugriff" versetzt.

**INSERT oder REPLACE, ALLOW NO ACCESS und CHECK PENDING CASCADE DEFERRED**

Nur die geladene Tabelle wird in den Status "Überprüfung anstehend, kein Zugriff" versetzt. Untergeordnete Fremdschlüsseltabellen, untergeordnete sofort gespeicherte Abfragetabellen und untergeordnete sofort gespeicherte Zwischenspeichertabellen behalten ihren ursprünglichen Status bei.

**REPLACE, ALLOW NO ACCESS und CHECK PENDING CASCADE IMMEDIATE** Die Tabelle und alle ihre untergeordneten Fremdschlüsseltabellen, untergeordneten sofort gespeicherten Abfragetabellen und untergeordneten sofort gespeicherten Zwischenspeichertabellen werden in den Status "Überprüfung anstehend, kein Zugriff" versetzt.

**Anmerkung:** Die Angabe der Option ALLOW READ ACCESS in einer LOAD REPLACE-Operation führt zu einem Fehler.

Verwenden Sie die Anweisung SET INTEGRITY, um den Status "Überprüfung anstehend" zu entfernen. Mit der Anweisung SET INTEGRITY wird eine Tabelle auf ungültige Integritätsbedingungen überprüft, und die Tabelle wird aus dem Status "Überprüfung anstehend" herausgenommen. Wenn alle Ladeoperationen im Einfügemodus (INSERT) ausgeführt werden, können die Integritätsbedingungen mit der Anweisung SET INTEGRITY inkrementell verarbeitet werden (d. h. es wird nur der angefügte Teil der Tabelle auf ungültige Integritätsbedingungen überprüft). Beispiel:

```
db2 load from infile1.ixf of ixf insert into table1
db2 set integrity for table1 immediate checked
```

Nur der angefügte Teil von TABLE1 wird auf ungültige Integritätsbedingungen überprüft. Die Überprüfung nur des angefügten Teils auf ungültige Integritätsbedingungen ist schneller als die Überprüfung der vollständigen Tabelle, vor allem bei einer umfangreichen Tabelle, an die kleine Datenmengen angefügt werden.

Wenn beim Laden einer Tabelle die Option CHECK PENDING CASCADE DEFERRED angegeben wird und die Anweisung SET INTEGRITY zur Überprüfung von ungültigen Integritätsbedingungen eingesetzt wird, werden die untergeordneten Tabellen in den Status "Überprüfung anstehend, kein Zugriff" versetzt. Sie müssen eine explizite Anforderung absetzen, um diesen Status für die Tabellen aufzuheben.

Wenn beim Laden einer Tabelle mit abhängigen gespeicherten Abfragetabellen oder abhängigen Zwischenspeichertabellen die Option INSERT angegeben wird und die Anweisung SET INTEGRITY zur Überprüfung von ungültigen Integritätsbedingungen eingesetzt wird, wird der Status "Überprüfung anstehend" für die Tabelle aufgehoben, und die Tabelle wird in den Modus "Kein Versetzen von Daten" versetzt. Dies geschieht, um die nachfolgenden inkrementellen Aktualisierungen der abhängigen gespeicherten Abfragetabellen und die inkrementelle Weitergabe der

abhängigen Zwischenspeichertabellen zu vereinfachen. Im Modus NO DATA MOVEMENT sind Operationen, die das Versetzen von Zeilen innerhalb der Tabelle bewirken könnten, nicht zulässig.

Sie können den Modus NO DATA MOVEMENT außer Kraft setzen, indem Sie beim Absetzen der Anweisung SET INTEGRITY die Option FULL ACCESS angeben. Der Zugriff auf die Tabelle ist vollständig möglich. Bei nachfolgenden Anweisungen REFRESH TABLE findet jedoch eine vollständig Neuberechnung der abhängigen gespeicherten Abfragetabellen statt, und die abhängigen Zwischenspeichertabellen werden in einen unvollständigen Status versetzt.

Falls die Option ALLOW READ ACCESS bei einer Ladeoperation angegeben wird, verbleibt die Tabelle im Status "Lesezugriff", bis die Anweisung SET INTEGRITY zur Überprüfung auf ungültige Integritätsbedingungen verwendet wird. Sobald die Tabelle festgeschrieben wurde, können Anwendungen sie zwar nach Daten abfragen, die vor der Ladeoperation vorhanden waren. Die Anwendungen können die neu geladenen Daten jedoch erst dann erkennen, nachdem die Anweisung SET INTEGRITY abgesetzt wurde.

Vor einer Überprüfung auf ungültige Integritätsbedingungen können mehrere Ladeoperationen für eine Tabelle stattfinden. Wenn alle Ladeoperationen im Modus "Lesezugriff zulassen" (ALLOW READ ACCESS) abgeschlossen werden, stehen nur die Daten für Abfragen zur Verfügung, die vor der ersten Ladeoperation in der Tabelle vorhanden waren.

Es können eine oder mehrere Tabellen in einem einzigen Aufruf dieser Anweisung überprüft werden. Wenn eine abhängige Tabelle separat überprüft werden soll, kann sich die übergeordnete Tabelle nicht im Status "Überprüfung anstehend" befinden. Andernfalls müssen die übergeordnete Tabelle und die abhängige Tabelle gleichzeitig überprüft werden. In Falle eines Zyklus für referenzielle Integrität müssen alle von diesem Zyklus betroffenen Tabellen in einen einzigen Aufruf der Anweisung SET INTEGRITY aufgenommen werden. Es kann unter Umständen von Vorteil sein, die übergeordnete Tabelle auf ungültige Integritätsbedingungen zu überprüfen, während eine abhängige Tabelle geladen wird. Dies kann nur geschehen, wenn sich die beiden Tabellen nicht im selben Tabellenbereich befinden.

Beim Absetzen der Anweisung SET INTEGRITY können Sie die Option INCREMENTAL angeben, um die inkrementelle Verarbeitung explizit anzufordern. In den meisten Fällen wird diese Option nicht benötigt, weil DB2<sup>®</sup> die inkrementelle Verarbeitung auswählt. Wenn die inkrementelle Verarbeitung nicht möglich ist, wird automatisch die volle Verarbeitung verwendet. Wenn die Option INCREMENTAL angegeben ist, die inkrementelle Verarbeitung jedoch nicht möglich ist, wird in folgenden Fällen ein Fehler gemeldet:

- Zur Tabelle wurden neue Integritätsbedingungen hinzugefügt, als sie sich im Status "Überprüfung anstehend" befand.
- Eine LOAD REPLACE-Operation wurde ausgeführt, oder die Option NOT LOGGED INITIALLY WITH EMPTY TABLE wurde aktiviert, nachdem die letzte Überprüfung auf Integritätsbedingungen für die Tabelle stattgefunden hat.
- Für eine übergeordnete Tabelle wurde eine LOAD REPLACE-Operation ausgeführt, oder eine solche Tabelle wurde nicht inkrementell auf Integrität überprüft.
- Vor der Migration befand sich die Tabelle im Status "Überprüfung anstehend". Wenn für die Tabelle zum ersten Mal nach der Migration eine Überprüfung auf Integritätsbedingungen vorgenommen wird, ist eine vollständige Verarbeitung erforderlich.

- Der Tabellenbereich, der die Tabelle oder ihre übergeordnete Tabelle enthält, wurde bis zu einem bestimmten Zeitpunkt aktualisierend wiederhergestellt, und die Tabelle sowie ihre übergeordnete Tabelle befinden sich in unterschiedlichen Tabellenbereichen.

Wenn eine Tabelle einen oder mehrere Werte W in der Spalte CONST\_CHECKED des Katalogs SYSCAT.TABLES enthält und die Option NOT INCREMENTAL in der Anweisung SET INTEGRITY nicht angegeben wird, wird die Tabelle inkrementell verarbeitet, und die Spalte CONST\_CHECKED des Katalogs SYSCAT.TABLES wird mit U gekennzeichnet, um anzugeben, dass nicht alle Daten vom System überprüft wurden.

Verwenden Sie die Option zum Laden der Ausnahmetabelle, um Informationen zu Zeilen mit ungültigen Integritätsbedingungen zu erfassen.

Die Anweisung SET INTEGRITY aktiviert keine DELETE-Auslöser als Ergebnis des Löschens von Zeilen, die gegen Integritätsbedingungen verstoßen. Die Auslöser sind jedoch aktiv, sobald für die Tabelle der Status "Überprüfung anstehend" aufgehoben wird. Demzufolge werden, wenn Sie Daten korrigieren und Zeilen aus der Ausnahmetabelle in die geladene Tabelle einfügen, auch die INSERT-Auslöser aktiviert, die möglicherweise für die Tabelle definiert sind. Die Implikationen davon sollten berücksichtigt werden. Eine Möglichkeit wäre z. B., den INSERT-Auslöser zu löschen, Zeilen aus der Ausnahmetabelle einzufügen und anschließend den INSERT-Auslöser erneut zu erstellen.

#### **Zugehörige Konzepte:**

- „Statusangaben für anstehende Aktionen nach einer Ladeoperation“ auf Seite 190
- „Ladeoperationen mit Lesezugriff“ auf Seite 93

#### **Zugehörige Referenzen:**

- „SET INTEGRITY statement“ in *SQL Reference, Volume 2*

---

## **Aktualisieren von abhängigen sofort gespeicherten Abfragetabellen**

Wenn die zu Grunde liegende Tabelle einer aktualisierten sofort gespeicherten Abfragetabelle unter Verwendung der Option INSERT geladen wird, bewirkt die Ausführung der Anweisung SET INTEGRITY für die abhängigen gespeicherten Abfragetabellen, für die REFRESH IMMEDIATE definiert ist, dass die gespeicherte Abfragetabelle inkrementell aktualisiert wird. Während einer inkrementellen Aktualisierung werden die Zeilen, die den angehängten Zeilen in den zu Grunde liegenden Tabellen entsprechen, aktualisiert und in die gespeicherten Abfragetabellen eingefügt. Die inkrementelle Aktualisierung beschleunigt sich, wenn große zu Grunde liegende Tabellen kleine Mengen von angehängten Daten enthalten. Es gibt Fälle, in denen eine inkrementelle Aktualisierung nicht zulässig ist und stattdessen eine vollständige Aktualisierung (also die Neuberechnung der Definitionsabfrage für die gespeicherte Abfragetabelle) verwendet wird.

Wenn die Option INCREMENTAL angegeben ist, eine inkrementelle Verarbeitung der gespeicherten Abfragetabelle jedoch nicht zulässig ist, wird in den folgenden Fällen ein Fehler zurückgegeben:

- Eine LOAD REPLACE-Operation wurde in einer zu Grunde liegenden Tabelle der gespeicherten Abfragetabelle ausgeführt, oder die Option NOT LOGGED

INITIALLY WITH EMPTY TABLE wurde aktiviert, seitdem die letzte Überprüfung auf Integritätsbedingungen für die zu Grunde liegende Tabelle stattgefunden hat.

- Die gespeicherte Abfragetabelle wurde geladen (im Ersetzungsmodus oder im Einfügemodus).
- Für eine zu Grunde liegende Tabelle wurde der Status "Überprüfung anstehend" aufgehoben, bevor die gespeicherte Abfragetabelle aktualisiert wurde (durch Verwendung der Option FULL ACCESS während der Überprüfung der Integrität).
- Die Integrität einer zu Grunde liegenden Tabelle der gespeicherten Abfragetabelle wurde nicht inkrementell überprüft.
- Die gespeicherte Abfragetabelle befand sich vor der Migration im Status "Überprüfung anstehend".
- Der Tabellenbereich, der die gespeicherte Abfragetabelle oder ihre zu Grunde liegende Tabelle enthält, wurde bis zu einem bestimmten Zeitpunkt aktualisierend wiederhergestellt, und die gespeicherte Abfragetabelle und ihre zu Grunde liegende Tabelle befinden sich in unterschiedlichen Tabellenbereichen.

Wenn die gespeicherte Abfragetabelle einen oder mehrere Werte W in der Spalte CONST\_CHECKED des Katalogs SYSCAT.TABLES enthält und die Option NOT INCREMENTAL in der Anweisung SET INTEGRITY nicht angegeben wird, wird die Tabelle inkrementell aktualisiert, und die Spalte CONST\_CHECKED des Katalogs SYSCAT.TABLES wird mit U gekennzeichnet, um deutlich zu machen, dass nicht alle Daten vom System überprüft wurden.

Das folgende Beispiel veranschaulicht eine Ladeoperation in die zu Grunde liegende Tabelle UT1 der gespeicherten Abfragetabelle AST1. Die Tabelle UT1 wird auf Datenintegrität überprüft und in den Modus "Kein Versetzen von Daten" versetzt. Sobald die inkrementelle Aktualisierung der Tabelle AST1 abgeschlossen ist, wird die Tabelle UT1 wieder in den Status "Vollständiger Zugriff" versetzt. In diesem Szenario werden sowohl die Überprüfung der Integrität für UT1 als auch die Aktualisierung von AST1 inkrementell verarbeitet.

```
LOAD FROM IMTFILE1.IXF of IXF INSERT INTO UT1;  
LOAD FROM IMTFILE2.IXF of IXF INSERT INTO UT1;  
SET INTEGRITY FOR UT1 IMMEDIATE CHECKED;  
REFRESH TABLE AST1;
```

#### Zugehörige Konzepte:

- „Überprüfen auf ungültige Integritätsbedingungen“ auf Seite 102

---

## Weitergeben von abhängigen sofort gespeicherten Zwischenspeichertabellen

Wenn die geladene Tabelle die zu Grunde liegende Tabelle einer Zwischenspeichertabelle mit dem Attribut IMMEDIATE PROPAGATE (sofortige Weitergabe) ist und die Ladeoperation im Einfügemodus erfolgt, wird die nachfolgende Weitergabe in die abhängigen sofort gespeicherten Zwischenspeichertabellen inkrementell verarbeitet.

Während einer inkrementellen Weitergabe werden die Zeilen, die den angehängten Zeilen in den zu Grunde liegenden Tabellen entsprechen, an die Zwischenspeichertabellen angehängt. Die inkrementelle Weitergabe beschleunigt sich, wenn große zu Grunde liegende Tabellen kleine Mengen von angehängten Daten enthalten. Die Leistung wird ebenfalls verbessert, wenn die Zwischenspeichertabelle zur

Aktualisierung ihrer abhängigen verzögert gespeicherten Abfragetabelle verwendet wird. Es gibt Fälle, in denen eine inkrementelle Weitergabe nicht zulässig ist und die Zwischenspeichertabelle als unvollständig gekennzeichnet wird. Dies bedeutet, dass das Byte für die Zwischenspeicherung in der Spalte CONST\_CHECKED den Wert F erhält. In diesem Status kann die Zwischenspeichertabelle nicht zur Aktualisierung ihrer abhängigen verzögerten gespeicherten Abfragetabelle verwendet werden. Stattdessen muss im Verwaltungsprozess für die gespeicherte Abfragetabelle eine vollständige Aktualisierung erfolgen.

Falls eine Tabelle einen Status "Unvollständig" aufweist und die Option INCREMENTAL angegeben wurde, eine inkrementelle Weitergabe der Tabelle jedoch nicht möglich ist, wird ein Fehler zurückgegeben. Wenn eine der folgenden Operationen stattgefunden hat, inaktiviert das System die sofortige Datenweitergabe und ändert den Tabellenstatus in "Unvollständig":

- Eine LOAD REPLACE-Operation wurde für eine zu Grunde liegende Tabelle der Zwischenspeichertabelle ausgeführt, oder die Option NOT LOGGED INITIALLY WITH EMPTY TABLE wurde nach der letzten Überprüfung der Integrität für die zu Grunde liegende Tabelle aktiviert.
- Die abhängige gespeicherte Abfragetabelle der Zwischenspeichertabelle oder die Zwischenspeichertabelle selbst wurde im Ersetzungsmodus oder im Einfügemodus geladen.
- Für eine zu Grunde liegende Tabelle wurde der Status "Überprüfung anstehend" aufgehoben, bevor die Zwischenspeichertabelle weitergegeben wurde (durch Verwendung der Option FULL ACCESS während der Überprüfung der Integrität).
- Die Integrität einer zu Grunde liegenden Tabelle der Zwischenspeichertabelle wurde nicht inkrementell überprüft.
- Der Tabellenbereich, der die Zwischenspeichertabelle oder ihre zu Grunde liegende Tabelle enthält, wurde bis zu einem bestimmten Zeitpunkt aktualisierend wiederhergestellt, und die Zwischenspeichertabelle und ihre zu Grunde liegende Tabelle befinden sich in unterschiedlichen Tabellenbereichen.

Wenn die Zwischenspeichertabelle einen Wert W in der Spalte CONST\_CHECKED des Katalogs SYSCAT.TABLES enthält und die Option NOT INCREMENTAL nicht angegeben wird, findet eine inkrementelle Weitergabe der Zwischenspeichertabelle statt, und die Spalte CONST\_CHECKED des Katalogs SYSCAT.TABLES wird mit U gekennzeichnet, um deutlich zu machen, dass nicht alle Daten vom System überprüft wurden.

Das folgende Beispiel veranschaulicht eine LOAD INSERT-Operation in die zu Grunde liegende Tabelle UT1 der Zwischenspeichertabelle G1 und ihre abhängige verzögert gespeicherte Abfragetabelle AST1. In diesem Szenario werden sowohl die Überprüfung der Integrität für UT1 als auch die Aktualisierung von AST1 inkrementell verarbeitet:

```
LOAD FROM IMTFILE1.IXF OF IXF INSERT INTO UT1;  
LOAD FROM IMTFILE2.IXF OF IXF INSERT INTO UT1;  
SET INTEGRITY FOR UT1,G1 IMMEDIATE CHECKED;  
  
REFRESH TABLE AST1 INCREMENTAL;
```

#### **Zugehörige Konzepte:**

- „Überprüfen auf ungültige Integritätsbedingungen“ auf Seite 102

---

## Hinweise zum mehrdimensionalen Clustering

Für MDC-Tabellen (Multi-Dimensional Clustering - mehrdimensionales Clustering) gelten die folgenden Einschränkungen:

- Die Option SAVECOUNT des Befehls LOAD wird nicht unterstützt.
- Der Änderungswert für den Datentyp TOTALFREESPACE wird nicht unterstützt, da diese Tabellen ihren freien Speicherbereich selbst verwalten.
- Der Änderungswert ANYORDER ist für MDC-Tabellen erforderlich. Werden Daten in eine MDC-Tabelle ohne den Änderungswert ANYORDER geladen, wird dieser Änderungswert vom Dienstprogramm explizit aktiviert.

Bei Verwendung des Befehls LOAD mit MDC werden ungültige eindeutige Integritätsbedingungen wie folgt behandelt:

- Wenn die Tabelle vor der Ladeoperation einen eindeutigen Schlüssel enthielt und doppelte Datensätze in die Tabelle geladen werden, verbleiben die ursprünglichen Datensätze in der Tabelle, und die neuen Datensätze werden während der DELETE-Phase gelöscht.
- Enthielt die Tabelle vor der Ladeoperation keinen eindeutigen Schlüssel und werden sowohl ein eindeutiger Schlüssel als auch doppelte Datensätze in die Tabelle geladen, wird nur einer der Datensätze mit dem eindeutigen Schlüssel geladen, und die anderen Datensätze werden während der DELETE-Phase gelöscht.

**Anmerkung:** Es gibt keine explizite Methode, mit der bestimmt werden kann, welche Datensätze geladen und welche gelöscht werden.

### Hinweise zur Leistung

Um eine bessere Leistung des Dienstprogramms LOAD beim Laden von MDC-Tabellen zu erzielen, sollte der Wert des Datenbankkonfigurationsparameters UTIL\_HEAP\_SZ erhöht werden. Die Leistung des Algorithmus 'mdc-load' ist deutlich besser, wenn dem Dienstprogramm mehr Speicher zur Verfügung steht. Dies verringert die Platten-E/A während des Datenclustering, das in der LOAD-Phase stattfindet. Wenn die Option DATA BUFFER des Befehls LOAD angegeben wird, sollte auch ihr Wert erhöht werden. Falls der Befehl LOAD eingesetzt wird, um mehrere MDC-Tabellen gleichzeitig zu laden, sollte der Wert des Konfigurationsparameters UTIL\_HEAP\_SZ entsprechend erhöht werden.

MDC-Ladeoperationen enthalten immer eine BUILD-Phase, da alle MDC-Tabellen mit Blockindizes ausgestattet sind.

Während der LOAD-Phase werden zusätzliche Protokollierungen zur Verwaltung der Blockzuordnung ausgeführt. Pro zugeordnetem Speicherbereich werden ca. zwei zusätzliche Protokollsätze erstellt. Um eine zufrieden stellende Leistung zu gewährleisten, sollte der Datenbankkonfigurationsparameter LOGBUFSZ auf einen Wert gesetzt werden, der diesen Umstand berücksichtigt.

Zum Laden von Daten in MDC-Tabellen wird eine systemtemporäre Tabelle mit einem Index eingesetzt. Die Größe der Tabelle ist proportional zur Anzahl der unterschiedlichen geladenen Zellen. Die Größe jeder Zeile in der Tabelle ist proportional zur Größe des Schlüssels für die MDC-Dimension. Um die Platten-E/A zu reduzieren, die durch die Bearbeitung dieser Tabelle während einer Ladeoperation entsteht, muss sichergestellt sein, dass der Pufferpool für den temporären Tabellenbereich groß genug ist.

### Zugehörige Konzepte:

- „Optimieren der Leistung von LOAD“ auf Seite 191
- „Tabellen mit mehrdimensionalem Clustering“ in *Systemverwaltung: Konzept*

---

## Erneutes Starten einer unterbrochenen Ladeoperation

Wenn das Dienstprogramm LOAD aufgrund eines Benutzerfehlers nicht gestartet werden kann, etwa aufgrund einer nicht vorhandenen Datendatei oder ungültiger Spaltennamen, wird es beendet, und die Tabelle verbleibt in einem normalen Status.

Tritt beim Laden von Daten ein Fehler auf, können Sie die Ladeoperation mit der Option RESTART vom letzten Konsistenzzustand ausgehend erneut starten oder die gesamte Tabelle mit der Option REPLACE erneut laden. Geben Sie die gleichen Parameter an wie beim vorherigen Aufruf, damit das Dienstprogramm die erforderlichen temporären Dateien findet. Da der Parameter SAVECOUNT bei MDC-Tabellen nicht unterstützt wird, findet die LOAD RESTART-Operation nur am Anfang der LOAD-, BUILD- oder DELETE-Phase statt.

**Anmerkung:** Eine Ladeoperation, bei der die Option ALLOW READ ACCESS angegeben wurde, kann entweder mit der Option ALLOW READ ACCESS oder mit der Option ALLOW NO ACCESS erneut gestartet werden. Hingegen kann eine Ladeoperation, bei der die Option ALLOW NO ACCESS angegeben wurde, nicht mit der Option ALLOW READ ACCESS erneut gestartet werden.

## Erneutes Starten oder Beenden einer Ladeoperation mit ALLOW READ ACCESS

Eine abgebrochene Ladeoperation, bei der die Option ALLOW READ ACCESS angegeben wurde, kann unter Umständen auch mit der Option ALLOW READ ACCESS erneut gestartet oder beendet werden. Hierdurch können andere Anwendungen die Tabellendaten abfragen, während die LOAD TERMINATE- bzw. LOAD RESTART-Operation läuft. Wie auch bei einer Ladeoperation im Modus "Lesezugriff zulassen" (ALLOW READ ACCESS) wird die Tabelle vor dem Festschreiben der Daten exklusiv gesperrt.

Falls das Indexobjekt nicht verfügbar ist oder als ungültig gekennzeichnet wurde, ist eine LOAD RESTART- oder LOAD TERMINATE-Operation im Modus "Lesezugriff zulassen" nicht zulässig.

Falls die ursprüngliche Ladeoperation in der INDEX COPY-Phase abgebrochen wurde, ist eine LOAD RESTART-Operation im Modus ALLOW READ ACCESS ("Lesezugriff zulassen") nicht zulässig, da der Index beschädigt sein könnte.

Falls eine Ladeoperation im Modus "Lesezugriff zulassen" während der LOAD-Phase abgebrochen wurde, wird sie in der LOAD-Phase erneut gestartet. Wurde sie während einer der anderen Phasen abgebrochen, findet der Neustart in der BUILD-Phase statt. Sofern die ursprüngliche Ladeoperation im Modus "Keinen Zugriff zulassen" (ALLOW NO ACCESS) ausgeführt wurde, kann die LOAD RESTART-Operation möglicherweise in der DELETE-Phase einsetzen, wenn die ursprüngliche Ladeoperation diesen Punkt erreicht hatte und der Index gültig ist. Wurde der Index als ungültig gekennzeichnet, startet das Dienstprogramm LOAD die Ladeoperation in der BUILD-Phase neu.

**Anmerkung:** Alle LOAD RESTART-Operationen wählen den Indexierungsmodus REBUILD selbst dann aus, wenn die Option INDEXING MODE INCREMENTAL angegeben wird.

Durch Absetzen eines Befehls LOAD TERMINATE kann die abgebrochene Ladeoperation im Allgemeinen mit einer minimalen Verzögerung zurückgesetzt werden. Wird jedoch ein Befehl LOAD TERMINATE für eine Ladeoperation abgesetzt, bei der ALLOW READ ACCESS und INDEXING MODE INCREMENTAL angegeben wurde, kann es unter Umständen zu einer Verzögerung kommen, während das Dienstprogramm LOAD die Indizes durchsucht und eventuelle Inkonsistenzen korrigiert. Die Länge dieser Verzögerung ist von der Größe der Indizes abhängig. Sie tritt unabhängig davon ein, ob die Option ALLOW READ ACCESS für die LOAD TERMINATE-Operation angegeben wurde. Eine Verzögerung findet dann nicht statt, wenn die ursprüngliche Ladeoperation vor der BUILD-Phase fehlschlug.

**Anmerkung:** Eine Verzögerung, die aus Korrekturen an Inkonsistenzen im Index entsteht, ist wesentlich geringer als die Verzögerung, die dadurch verursacht wird, dass die Indizes als ungültig gekennzeichnet und wiederhergestellt werden.

Eine LOAD RESTART-Operation kann nicht für eine Tabelle ausgeführt werden, deren Status "Für Laden nicht neu startbar" lautet. Die Tabelle kann während einer aktualisierenden Wiederherstellung in einen solchen Tabellenstatus versetzt werden, beispielsweise, wenn die aktualisierende Wiederherstellung bis zu einem Zeitpunkt ausgeführt wird, der vor dem Ende einer Ladeoperation lag, oder wenn eine aktualisierende Wiederherstellung über eine abgebrochene Ladeoperation hinweg, jedoch nicht bis zum Ende der LOAD TERMINATE- oder LOAD RESTART-Operation ausgeführt wird.

**Zugehörige Konzepte:**

- „Tabellensperren, Statuswerte für Tabellen und Statuswerte für Tabellenbereiche“ auf Seite 186
- „Erneutes Starten oder Beenden einer Ladeoperation in einer Umgebung mit partitionierten Datenbanken“ auf Seite 211

**Zugehörige Referenzen:**

- „LOAD-Konfigurationsoptionen für partitionierte Datenbanken“ auf Seite 213

---

## Wiederherstellen von Daten mit der Datei mit den Angaben zur Speicherposition der Ladekopie

Anhand der Registrierdatenbank-Variablen DB2LOADREC wird die Datei mit den Angaben zur Speicherposition der Ladekopie identifiziert. Diese Datei wird während der aktualisierenden Wiederherstellung zum Lokalisieren der Ladekopie verwendet. Sie enthält folgende Informationen:

- Datenträgertyp
- Anzahl der zu verwendenden Datenträgereinheiten
- Speicherposition der Ladekopie, die bei einer Ladeoperation für eine Tabelle generiert wurde
- Dateiname der Ladekopie (sofern zutreffend)

Falls die Speicherpositionsdatei nicht existiert oder darin kein übereinstimmender Eintrag gefunden wurde, werden die Informationen aus dem Protokollsatz verwendet.

Die Informationen in der Datei könnten überschrieben werden, bevor die aktualisierende Wiederherstellung stattfindet.

**Anmerkungen:**

1. In einer Umgebung mit partitionierten Datenbanken muss die Registrierdatenbank-Variable DB2LOADREC unter Verwendung des Befehls **db2set** für alle Datenbankpartitionsserver festgelegt werden.
2. In einer Umgebung mit partitionierten Datenbanken muss die Datei der Ladekopie auf jedem Datenbankpartitionsserver vorhanden sein, und der Dateiname (einschließlich des Pfads) muss derselbe sein.
3. Falls ein Eintrag in der Datei, die durch die Registrierdatenbank-Variable DB2LOADREC angegeben ist, ungültig ist, wird der ungültige Eintrag unter Verwendung der Informationen aus der alten Datei mit den Angaben zur Speicherposition der Ladekopie ersetzt.

Die Datei mit den Angaben zur Speicherposition enthält die folgenden Informationen. Die ersten fünf Parameter müssen gültige Werte aufweisen und werden zum Identifizieren der Ladekopie verwendet. Die gesamte Struktur wird für jede aufzeichnete Ladekopie wiederholt. Beispiel:

```
|
| TIMestamp      19950725182542      * Zur Ladezeit generierte Zeitmarke
| SCHema        PAYROLL              * Schema der geladenen Tabelle
| TABlename     EMPLOYEEES          * Tabellename
| DATabasename  DBT                  * Datenbankname
| DB2instance   toronto              * DB2INSTANCE
| BUFFernumber  NULL                 * Anzahl der Puffer für die
| Wiederherstellung
| SESSionnumber NULL                 * Anzahl der Sitzungen für die
| Wiederherstellung
| TYPeofmedia   L                    * Datenträgertyp - L für lokale Einheit
|                                           A für TSM
|                                           0 für andere Lieferanten
|
| LOCationnumber 3                    * Anzahl der Speicherpositionen
|   ENTRy       /u/toronto/dbt.payroll.employes.001
|   ENT         /u/toronto/dbt.payroll.employes.002
|   ENT         /dev/rmt0
|
| TIM           19950725192054
| SCH           PAYROLL
| TAB           DEPT
| DAT           DBT
| DB2®         toronto
| BUF           NULL
| SES           NULL
| TYP           A
|
| TIM           19940325192054
| SCH           PAYROLL
| TAB           DEPT
| DAT           DBT
| DB2           toronto
| BUF           NULL
| SES           NULL
| TYP           0
|
| SHRlib       /@sys/lib/backup_vendor.a
```

### Anmerkungen:

1. Die ersten drei Zeichen in jedem Schlüsselwort sind wichtig. Alle Schlüsselwörter sind in der angegebenen Reihenfolge erforderlich. Leerzeilen werden nicht akzeptiert.
2. Die Zeitmarke hat das Format *jjjmmmttssmmss* (Jahr Monat Tag Stunde Minute Sekunde).
3. Alle Felder sind verbindlich, mit Ausnahme von BUF und SES, die den Wert NULL haben können. Falls SES Null ist, wird der durch den Konfigurationsparameter *numloadrecses* angegebene Wert verwendet. Ist BUF Null, lautet der Standardwert SES+2.
4. Falls auch nur einer der der Einträge in der Datei mit den Angaben zur Speicherposition ungültig ist, werden die entsprechenden Werte aus der früheren Datei mit den Angaben zur Speicherposition der Ladekopie verwendet.
5. Es gibt folgende Datenträgertypen: lokale Einheit (L für Band, Platte oder Diskette), TSM (A) oder Datenträger anderer Lieferanten (0). Lautet der Typ L, ist die Anzahl der Speicherpositionen, gefolgt von den Einträgen zur Speicherposition, erforderlich. Lautet der Typ A, ist keine weitere Eingabe erforderlich. Lautet der Typ 0, ist der Name der gemeinsam benutzten Bibliothek erforderlich.
6. Der Parameter SHRlib zeigt auf eine Bibliothek, die zum Speichern der Ladekopiedaten dient.
7. Wenn Sie eine Ladeoperation aufrufen und hierbei die Option COPY NO oder NONRECOVERABLE angeben und nach Abschluss der Operation keine Sicherungskopie der Datenbank oder der betroffenen Tabellenbereiche erstellen, können Sie die Datenbank oder die Tabellenbereiche nicht für einen Zeitpunkt wiederherstellen, der nach der Ladeoperation liegt. Das heißt, dass Sie keine aktualisierende Wiederherstellung ausführen können, um die Datenbank oder die Tabellenbereiche auf dem Stand wiederherzustellen, der nach der Ladeoperation vorlag. Sie können die Datenbank oder die Tabellenbereiche lediglich für einen Zeitpunkt vor der Ladeoperation wiederherstellen.

Für den Fall, dass Sie eine bestimmte Ladekopie verwenden wollen, können Sie in der Datei des Wiederherstellungsprotokolls für die Datenbank die Zeitmarke für diese spezifische Ladeoperation ermitteln. In einer Umgebung mit partitionierten Datenbanken ist die Datei des Wiederherstellungsprotokolls für jede Datenbankpartition lokal.

### Zugehörige Referenzen:

- „Tivoli Storage Manager“ in *Datenwiederherstellung und hohe Verfügbarkeit Handbuch und Referenz*

---

## Befehl LOAD

Lädt Daten in eine DB2-Tabelle. Die Daten können auf dem Server in Form einer Datei, eines Bands oder einer benannten Pipe vorliegen. Daten, die sich auf einem Client mit Fernverbindung befinden, können als vollständig qualifizierte Datei oder als benannte Pipe vorliegen. Daten können auch über einen benutzerdefinierten Cursor geladen werden.

### Einschränkungen:

Das Dienstprogramm LOAD unterstützt das Laden von Daten auf Hierarchieebene nicht. Das Dienstprogramm LOAD ist mit Bereichsclustertabellen nicht kompatibel.

**Geltungsbereich:**

Dieser Befehl kann in einer einzigen Anforderung für mehrere Datenbankpartitionen abgesetzt werden.

**Berechtigung:**

Eine der Folgenden:

- *sysadm*
- *dbadm*
- Berechtigung LOAD für die Datenbank und
  - Zugriffsrecht INSERT für die Tabelle, wenn das Dienstprogramm LOAD im Einfügemodus (INSERT), im Beendigungsmodus (TERMINATE) zur Beendigung einer vorherigen LOAD INSERT-Operation oder im Neustartmodus (RESTART) zum erneuten Starten einer vorherigen LOAD INSERT-Operation aufgerufen wird.
  - Zugriffsrecht INSERT und DELETE für die Tabelle, wenn das Dienstprogramm LOAD im Ersetzungsmodus (REPLACE), im Beendigungsmodus (TERMINATE) zur Beendigung einer vorherigen LOAD REPLACE-Operation oder im Neustartmodus (RESTART) zum erneuten Starten einer vorherigen LOAD REPLACE-Operation aufgerufen wird.
  - Zugriffsrecht INSERT für die Ausnahmetabelle, wenn eine solche Tabelle im Rahmen der Ladeoperation verwendet wird.

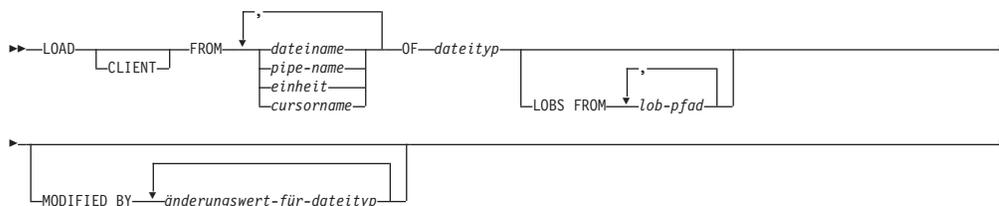
Da der Exemplareigner auch Eigner aller LOAD-Prozesse (und im Allgemeinen auch aller DB2-Serverprozesse) ist und alle diese Prozesse die ID des Exemplareigners verwenden, um auf benötigte Dateien zuzugreifen, muss der Exemplareigner über einen Lesezugriff auf Eingabedatendateien verfügen. Diese Eingabedatendateien müssen für den Exemplareigner lesbar sein, unabhängig davon, wer den Befehl aufruft.

**Erforderliche Verbindung:**

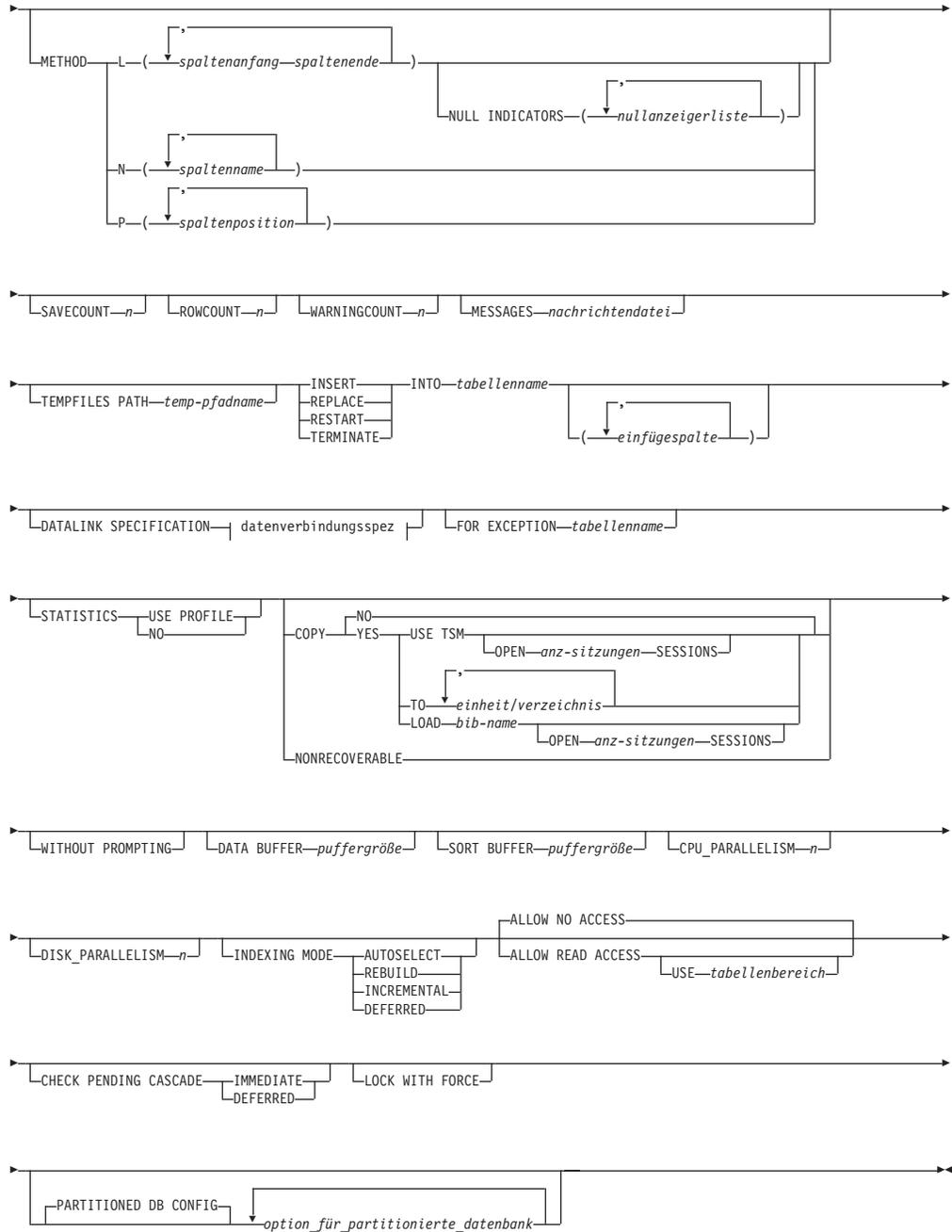
Datenbank. Wenn das implizite Herstellen von Verbindungen aktiviert ist, wird eine Verbindung zur Standarddatenbank hergestellt.

Exemplar. Eine explizite Zuordnung ist nicht erforderlich. Wenn eine Verbindung zur Datenbank hergestellt wurde, wird versucht, eine implizite Zuordnung zum lokalen Exemplar vorzunehmen.

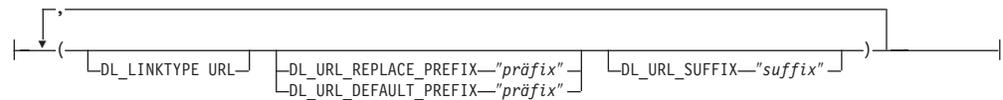
**Befehlssyntax:**



# Befehl LOAD



## datenverbindungsspez:



## Befehlsparameter:

### ALLOW NO ACCESS

Während des Ladens sperrt LOAD die Zieltabelle für den exklusiven Zugriff. Der Tabellenstatus wird während des Ladens auf "Laden läuft"

gesetzt. ALLOW NO ACCESS ist das Standardverhalten. Dies ist die einzige gültige Option für LOAD REPLACE.

Wenn für die Tabelle Integritätsbedingungen gelten, wird der Tabellenstatus auf "Überprüfung anstehend" sowie "Laden läuft" gesetzt. Die Anweisung SET INTEGRITY muss verwendet werden, um den Status "Überprüfung anstehend" (CHECK PENDING) für die Tabelle aufzuheben.

#### ALLOW READ ACCESS

LOAD sperrt die Zieltabelle im Modus für die gemeinsame Benutzung (SHARE). Der Tabellenstatus wird auf "Laden läuft" und "Lesezugriff" gesetzt. Eingabeprogramme können auf die Daten zugreifen, die nicht zum Delteteil gehören, während die Tabelle geladen wird. Anders ausgedrückt können Eingabeprogramme für die Tabelle auf diejenigen Daten zugreifen, die vor dem Beginn des Ladens vorhanden waren. Daten, die geladen werden, sind erst nach Abschluss des Ladevorgangs verfügbar. Die Befehle LOAD TERMINATE oder LOAD RESTART eines Ladens im Modus "Lesezugriff zulassen" (ALLOW READ ACCESS) können diese Option verwenden. Bei einem Laden im Modus "Keinen Zugriff zulassen" (ALLOW NO ACCESS) kann sie von den Befehlen LOAD TERMINATE oder LOAD RESTART nicht verwendet werden. Des Weiteren ist diese Option dann ungültig, wenn die Indizes für die Zieltabelle so gekennzeichnet wurden, dass eine erneute Erstellung erforderlich ist.

Wenn für die Tabelle Integritätsbedingungen gelten, wird der Tabellenstatus auf "Überprüfung anstehend" sowie "Laden läuft" und "Lesezugriff" gesetzt. Am Ende des Ladevorgangs wird der Tabellenstatus "Laden läuft" aufgehoben. Die Statuswerte "Überprüfung anstehend" und "Lesezugriff" für die Tabelle bleiben jedoch gültig. Die Anweisung SET INTEGRITY muss verwendet werden, um den Status "Überprüfung anstehend" (CHECK PENDING) für die Tabelle aufzuheben. Während sich die Tabelle im Status "Überprüfung anstehend" und "Lesezugriff" (READ ACCESS) befindet, können Eingabeprogramme weiter auf die Daten zugreifen, die nicht zum Delteteil gehören, wohingegen der neue Teil (= Delteteil) der Daten erst nach vollständiger Ausführung der Anweisung SET INTEGRITY verfügbar ist. Ein Benutzer kann mehrere Ladevorgänge für dieselbe Tabelle vornehmen, ohne eine Anweisung SET INTEGRITY abzusetzen. Bis zum Absetzen der Anweisung SET INTEGRITY sind jedoch nur die ursprünglichen (überprüften) Daten erkennbar.

ALLOW READ ACCESS unterstützt außerdem die folgenden Änderungswerte:

#### USE tabellenbereichsname

Falls die Indizes erneut erstellt werden, wird eine Schattenkopie des Indexes in dem mit *tabellenbereichsname* angegebenen Tabellenbereich erstellt und am Ende des Ladevorgangs während der INDEX COPY-Phase in den ursprünglichen Tabellenbereich kopiert. Diese Option kann nur bei systemtemporären Tabellen verwendet werden. Ist sie nicht angegeben, wird der Schattenindex in demselben Tabellenbereich wie das Indexobjekt erstellt. Falls die Schattenkopie in demselben Tabellenbereich wie das Indexobjekt erstellt wird, wird das Schattenindexobjekt sofort über das alte Indexobjekt kopiert. Befindet sich die Schattenkopie in einem anderen Tabellenbereich als das Indexobjekt, wird eine physische Kopie ausgeführt. Dies kann sich erheblich auf die Ein-/Ausgabe und die

benötigte Zeit auswirken. Das Kopieren findet statt, während die Tabelle am Ende eines Ladevorgangs in der INDEX COPY-Phase offline ist.

Ohne Angabe dieser Option wird der Schattenindex in demselben Tabellenbereich wie das Original erstellt. Da sich sowohl der Originalindex als auch der Schattenindex standardmäßig gleichzeitig in demselben Tabellenbereich befinden, kann es sein, dass nicht genügend Speicherbereich für die Aufnahme beider Indizes im gleichen Tabellenbereich verfügbar ist. Die Verwendung dieser Option stellt sicher, dass ausreichend Tabellenbereich für die Indizes erhalten bleibt.

Diese Option wird ignoriert, wenn der Benutzer INDEXING MODE REBUILD oder INDEXING MODE AUTOSELECT nicht angibt. Außerdem wird sie ignoriert, wenn INDEXING MODE AUTOSELECT ausgewählt wird und der Index beim Laden inkrementell aktualisiert wird.

### CHECK PENDING CASCADE

Falls LOAD die Tabelle in den Status "Überprüfung anstehend" (CHECK PENDING) versetzt, kann ein Benutzer mit der Option CHECK PENDING CASCADE angeben, ob der Status "Überprüfung anstehend" der geladenen Tabelle unmittelbar an alle untergeordneten Tabellen weitergegeben wird (einschließlich untergeordneter Fremdschlüsseltabellen, untergeordneter sofort gespeicherter Abfragetabellen und untergeordneter sofort gespeicherter Zwischenspeichertabellen).

### IMMEDIATE

Gibt an, dass der Status "Überprüfung anstehend" (Lesemodus oder Modus ohne Zugriff) für Integritätsbedingungen über Fremdschlüssel sofort auf alle untergeordneten Fremdschlüsseltabellen ausgeweitet wird. Falls der Tabelle sofort gespeicherte Abfragetabellen oder sofort gespeicherte Zwischenspeichertabellen untergeordnet sind, wird der Status "Überprüfung anstehend" unmittelbar auf die gespeicherten Abfragetabellen und die Zwischenspeichertabellen ausgeweitet. Bitte beachten Sie, dass der Status "Überprüfung anstehend" bei einer LOAD INSERT-Operation selbst dann nicht auf untergeordnete Fremdschlüsseltabellen ausgeweitet wird, wenn die Option IMMEDIATE angegeben ist.

Wenn die geladene Tabelle zu einem späteren Zeitpunkt auf ungültige Integritätsbedingungen überprüft wird (mit der Option IMMEDIATE CHECKED der Anweisung SET INTEGRITY), werden untergeordnete Fremdschlüsseltabellen, die in den Status "Überprüfung anstehend" mit Lesezugriff gestellt wurden, in den Status "Überprüfung anstehend" ohne Zugriff versetzt.

### DEFERRED

Gibt an, dass nur die geladene Tabelle in den Status "Überprüfung anstehend" (Lesemodus oder Modus ohne Zugriff) versetzt wird. Die Statuswerte von untergeordneten Fremdschlüsseltabellen, untergeordneten sofort gespeicherten Abfragetabellen und untergeordneten sofort gespeicherten Zwischenspeichertabellen werden nicht geändert.

Untergeordnete Fremdschlüsseltabellen können später implizit in den Status "Überprüfung anstehend" ohne Zugriff versetzt werden, wenn die ihnen übergeordneten Tabellen auf ungültige Integritäts-

bedingungen überprüft werden (unter Verwendung der Option IMMEDIATE CHECKED der Anweisung SET INTEGRITY). Untergeordnete sofort gespeicherte Abfragetabellen und untergeordnete sofort gespeicherte Zwischenspeichertabellen werden implizit in den Status "Überprüfung anstehend" ohne Zugriff versetzt, sobald eine der ihnen zu Grunde liegenden Tabellen auf ungültige Integritätsbedingungen überprüft wird. Eine Warnung (SQLSTATE 01586) wird ausgegeben, um anzugeben, dass abhängige Tabellen in den Status "Überprüfung anstehend" versetzt wurden. Angaben dazu, wann diese untergeordneten Tabellen in den Status "Überprüfung anstehend" versetzt werden, finden Sie im Abschnitt mit den Hinweisen zur Anweisung SET INTEGRITY im Handbuch "SQL Reference".

Wird die Option CHECK PENDING CASCADE nicht angegeben, gilt Folgendes:

- Nur die geladene Tabelle wird in den Status "Überprüfung anstehend" versetzt. Die Statuswerte von untergeordneten Fremdschlüsseltabellen, untergeordneten sofort gespeicherten Abfragetabellen und untergeordneten sofort gespeicherten Zwischenspeichertabellen bleiben unverändert. Diese Tabellen können später implizit in den Status "Überprüfung anstehend" versetzt werden, sobald die geladene Tabelle auf ungültige Integritätsbedingungen hin überprüft wird.

Falls LOAD die Zieltabelle nicht in den Status "Überprüfung anstehend" versetzt, wird die Option CHECK PENDING CASCADE ignoriert.

## CLIENT

Gibt an, dass sich die zu ladenden Daten auf einem Client mit Fernverbindung befinden. Diese Option wird ignoriert, wenn die Ladeoperation nicht durch einen fernen Client aufgerufen wird. Die Verwendung dieser Option in Verbindung mit dem Dateityp CURSOR wird nicht unterstützt.

### Anmerkungen:

1. Die Änderungswerte `dumpfile` und `lobsinfile` verweisen selbst bei Angabe des Schlüsselworts CLIENT auf Dateien, die sich auf dem Server befinden.
2. Die Codepage wird während einer fernen Ladeoperation nicht konvertiert. Wenn sich die Codepage der Daten von der Codepage des Servers unterscheidet, muss die Codepage der Daten mit dem Änderungswert `codepage` angegeben werden.

Im folgenden Beispiel soll eine Datendatei (`/u/user/data.del`), die sich auf einem Client mit Fernverbindung befindet, in die Tabelle MYTABLE der Server-Datenbank geladen werden:

```
db2 load client from /u/user/data.del of del
      modified by codepage=850 insert into mytable
```

## COPY NO

Gibt an, dass der Tabellenbereich, in dem sich die Tabelle befindet, in den Status "Sicherung anstehend" versetzt wird, wenn die aktualisierende Wiederherstellung aktiviert ist (dass heißt, `logretain` oder `userexit` ist aktiv). Bei Verwendung der Option COPY NO wird auch der Tabellenbereich in den Tabellenbereichsstatus "Laden läuft" (LOAD IN PROGRESS) versetzt. Dies ist ein vorübergehender Status, der aufgehoben wird, sobald der Ladevorgang abgebrochen wird oder abgeschlossen ist. Die Daten in allen Tabellen des Tabellenbereichs können erst dann aktualisiert oder gelöscht werden, nachdem eine Sicherung des Tabellenbereichs oder eine vollständige Siche-

## Befehl LOAD

lung der Datenbank durchgeführt wurde. Der Zugriff auf die Daten in allen Tabellen ist jedoch durch die SELECT-Anweisung möglich.

Bei Verwendung von LOAD mit COPY NO für eine wiederherstellbare Datenbank bleiben die Tabellenbereiche im Status "Sicherung anstehend" (BACKUP PENDING). Wenn beispielsweise LOAD mit COPY NO und INDEXING MODE DEFERRED ausgeführt wird, sollten die Indizes aktualisiert werden. Bestimmte Abfragen der Tabelle erfordern möglicherweise eine Indexsuche und schlagen fehl, bis der Index aktualisiert wurde. Der Index kann nicht aktualisiert werden, wenn er sich in einem Tabellenbereich befindet, der den Status "Sicherung anstehend" hat. In diesem Fall ist der Zugriff auf die Tabelle nicht zulässig, bis eine Sicherung ausgeführt wurde.

**Anmerkung:** Die Indexaktualisierung wird automatisch durch die Datenbank ausgeführt, wenn eine Abfrage auf den Index zugreift.

### COPY YES

Gibt an, dass eine Kopie der geladenen Daten gesichert wird. Diese Option ist unzulässig, wenn die aktualisierende Wiederherstellung inaktiviert ist (sowohl *logretain* als auch *userexit* sind inaktiv). Die Option wird für Tabellen mit DATALINK-Spalten nicht unterstützt.

### USE TSM

Gibt an, dass die Kopie unter Verwendung von Tivoli Storage Manager (TSM) gespeichert wird.

### OPEN anz-sitzungen SESSIONS

Die Anzahl der E/A-Sitzungen, die mit Tivoli Storage Manager (TSM) oder einem Produkt eines anderen Lieferanten verwendet werden sollen. Der Standardwert ist 1.

### TO einheit/verzeichnis

Gibt die Einheit oder das Verzeichnis an, auf der bzw. in dem das Kopierabbild erstellt wird.

### LOAD bib-name

Der Name der gemeinsam benutzten Bibliothek (DLL unter Windows-Betriebssystemen), die die Sicherungs- und Wiederherstellungs-E/A-Funktionen des Lieferanten enthält, die verwendet werden sollen. Kann den vollständigen Pfad enthalten. Wenn der vollständige Pfad nicht angegeben wird, wird standardmäßig der Pfad verwendet, in dem sich die Benutzerexitprogramme befinden.

### CPU\_PARALLELISM n

Gibt die Anzahl der Prozesse oder Threads an, die vom Dienstprogramm LOAD gestartet werden, um Datensätze beim Erstellen von Tabellenobjekten syntaktisch zu analysieren, zu konvertieren und zu formatieren. Dieser Parameter soll die Nutzung der partitionsinternen Parallelität ermöglichen. Er ist besonders nützlich beim Laden von vorsortierten Daten, weil die Satzreihenfolge der Quelldaten beibehalten wird. Wenn der Wert dieses Parameters Null ist oder nicht angegeben wurde, verwendet das Dienstprogramm LOAD zur Laufzeit einen sinnvollen Standardwert. Dieser basiert in der Regel auf der Anzahl der verfügbaren Zentraleinheiten.

**Anmerkungen:**

1. Wird dieser Parameter bei Tabellen verwendet, die entweder LOB- oder LONG VARCHAR-Felder enthalten, wird sein Wert Eins, unabhängig von der Anzahl der System-CPU's oder dem vom Benutzer angegebenen Wert.
2. Die Angabe eines kleinen Werts für den Parameter SAVECOUNT bewirkt, dass das Ladeprogramm viel mehr E/A-Vorgänge ausführen muss, um sowohl Daten als auch Tabellenmetadaten zu löschen. Wenn CPU\_PARALLELISM größer als Eins ist, erfolgen die Löschvorgänge asynchron, so dass das Ladeprogramm die CPU nutzen kann. Wenn CPU\_PARALLELISM auf Eins gesetzt ist, wartet das Ladeprogramm bei Konsistenzzuständen auf E/A. Eine Ladeoperation wird bei Angabe von Zwei für CPU\_PARALLELISM und 10 000 für SAVECOUNT schneller ausgeführt als die gleiche Operation, wenn CPU\_PARALLELISM auf Eins gesetzt ist, auch wenn nur eine CPU vorhanden ist.

**DATA BUFFER puffergröße**

Gibt die Anzahl der 4 KB-Seiten (unabhängig vom Grad der Parallelität) an, die als Pufferbereich zur Übertragung von Daten im Dienstprogramm verwendet werden sollen. Wenn der angegebene Wert kleiner als der algorithmische Mindestwert ist, werden die erforderlichen Mindestressourcen verwendet, und es wird keine Warnung zurückgegeben.

Dieser Speicher wird direkt aus dem Zwischenspeicher der Dienstprogramme zugeordnet, dessen Größe mit dem Datenbankkonfigurationsparameter *util\_heap\_sz* geändert werden kann.

Wenn kein Wert angegeben wird, wird vom Dienstprogramm zur Laufzeit ein sinnvoller Standardwert berechnet. Der Standardwert basiert auf einem Prozentsatz des freien Speicherbereichs, der bei der Exemplarerstellung des Ladeprogramms im Zwischenspeicher der Dienstprogramme verfügbar ist, sowie auf einigen Merkmalen der Tabelle.

**DATALINK SPECIFICATION**

Für jede DATALINK-Spalte ist eine Spaltenangabe in runden Klammern möglich. Jede Spaltenangabe umfasst mindestens einen DL\_LINKTYPE-Wert, eine Präfix- und eine DL\_URL\_SUFFIX-Angabe. Die Präfixangabe kann entweder DL\_URL\_REPLACE\_PREFIX oder DL\_URL\_DEFAULT\_PREFIX sein.

Es sind so viele DATALINK-Spaltenpezifikationen möglich, wie DATALINK-Spalten in der Tabelle definiert sind. Die Reihenfolge der Angaben richtet sich nach der Reihenfolge der DATALINK-Spalten in der Liste *einfügespalte* oder in der Tabellendefinition (wenn keine Liste *einfügespalte* angegeben wird).

**DISK\_PARALLELISM n**

Gibt die Anzahl der Prozesse oder Threads an, die vom Dienstprogramm LOAD gestartet werden, um Daten in die Tabellenbereichsbehälter zu schreiben. Wenn kein Wert angegeben wird, wählt das Dienstprogramm einen sinnvollen Standardwert auf Basis der Anzahl der Tabellenbereichsbehälter und der Merkmale der Tabelle aus.

**DL\_LINKTYPE**

Muss, wenn angegeben, mit dem LINKTYPE-Wert der Spaltendefinition übereinstimmen. Demzufolge ist DL\_LINKTYPE URL akzeptabel, wenn die Spaltendefinition LINKTYPE URL angibt.

### **DL\_URL\_DEFAULT\_PREFIX "präfix"**

Muss, wenn angegeben, als Standardpräfix für alle DATALINK-Werte in der gleichen Spalte verwendet werden. In diesem Zusammenhang bezieht sich Präfix auf den Teil "Schema-Host-Anschluss" der URL-Angabe.

Beispiele für Präfixe sind:

```
"http://server"  
"file://server"  
"file:"  
"http://server:80"
```

Enthalten die Spaltendaten kein Präfix und wird mit DL\_URL\_DEFAULT\_PREFIX ein Standardpräfix angegeben, wird dieses Standardpräfix dem Spaltenwert (sofern dieser nicht NULL ist) vorangestellt.

Beispiel: Wenn DL\_URL\_DEFAULT\_PREFIX das Standardpräfix "http://toronto" angibt, gilt Folgendes:

- Der Spalteneingabewert "/x/y/z" wird als "http://toronto/x/y/z" gespeichert.
- Der Spalteneingabewert "http://coyote/a/b/c" wird als "http://coyote/a/b/c" gespeichert.
- Der Spalteneingabewert NULL wird als NULL gespeichert.

### **DL\_URL\_REPLACE\_PREFIX "präfix"**

Diese Klausel ist zum Laden oder Importieren von Daten nützlich, die zuvor vom Dienstprogramm EXPORT generiert wurden, wenn der Benutzer den Host-Namen in den Daten global durch einen anderen Host-Namen ersetzen möchte. Wird, wenn angegeben, das Präfix für *alle* Spaltenwerte, die nicht NULL sind. Wenn ein Spaltenwert mit einem Präfix versehen ist, wird dieses durch das mit DL\_URL\_REPLACE\_PREFIX angegebene Präfix ersetzt. Wenn ein Spaltenwert kein Präfix aufweist, wird das mit DL\_URL\_REPLACE\_PREFIX angegebene Präfix dem Spaltenwert vorangestellt.

Beispiel: Wenn DL\_URL\_REPLACE\_PREFIX das Präfix "http://toronto" angibt, gilt Folgendes:

- Der Spalteneingabewert "/x/y/z" wird als "http://toronto/x/y/z" gespeichert.
- Der Spalteneingabewert "http://coyote/a/b/c" wird als "http://toronto/a/b/c" gespeichert. Beachten Sie, dass "coyote" durch "toronto" ersetzt wird.
- Der Spalteneingabewert NULL wird als NULL gespeichert.

### **DL\_URL\_SUFFIX "suffix"**

Wird, wenn angegeben, an jeden Spaltenwert für die Spalte, der nicht NULL ist, angefügt. Wird an die Pfadkomponente des Datenspeicherpositionsteils des DATALINK-Werts angefügt.

### **FOR EXCEPTION tabellenname**

Gibt die Ausnahmetabelle an, in die fehlerhafte Zeilen kopiert werden. Jede Zeile, die gegen einen eindeutigen Index oder einen Primärschlüsselindex verstößt, wird kopiert. DATALINK-Ausnahmen werden ebenfalls in der Ausnahmetabelle erfasst. Wenn ein Tabellenname ohne Qualifikationsmerkmal angegeben wird, wird die Tabelle mit dem aktuellen Schema (CURRENT SCHEMA) qualifiziert.

Informationen, die in die Ausnahmetabelle geschrieben werden, werden *nicht* in die Speicherauszugsdatei geschrieben. In einer Umgebung mit par-

tionierten Datenbanken muss eine Ausnahmetabelle für diejenigen Partitionen definiert werden, auf denen die zu ladende Tabelle definiert ist. Die Speicherauszugsdatei andererseits enthält Zeilen, die nicht geladen werden können, weil sie ungültig sind oder Syntaxfehler aufweisen.

#### **FROM dateiname/pipe-name/einheit/cursorname**

Gibt die Datei, die Pipe, die Einheit oder den auf eine SQL-Anweisung verweisenden Cursor an, die/der die zu ladenden Daten enthält. Sofern nicht die Option CLIENT angegeben wird, muss sich die Eingabequelle, wenn es sich bei ihr um eine Datei, Pipe oder Einheit handelt, auf der Datenbankpartition befinden, auf der sich auch die Datenbank befindet. Wenn mehrere Namen angegeben werden, werden sie nacheinander verarbeitet. Wenn das letzte angegebene Element ein Bandlaufwerk ist, wird der Benutzer aufgefordert, ein weiteres Band einzulegen. Gültige Antwortoptionen sind:

- c** Fortsetzen. Verwendung der Einheit fortsetzen, die die Warnung generiert hat (zum Beispiel, wenn ein neues Band eingelegt wurde).
- d** Einheit beenden. Verwendung der Einheit beenden, die die Warnung generiert hat (zum Beispiel, wenn keine Bänder mehr vorhanden sind).
- t** Beenden. Alle Einheiten beenden.

#### **Anmerkungen:**

1. Es empfiehlt sich, den vollständig qualifizierten Dateinamen zu verwenden. Wenn es sich um einen fernen Server handelt, muss der vollständig qualifizierte Dateiname verwendet werden. Wenn sich die Datenbank auf der gleichen Datenbankpartition befindet wie das aufrufende Programm, können relative Pfade verwendet werden.
2. Das Laden von Daten aus mehreren IXF-Dateien wird unterstützt, wenn es sich physisch um separate Dateien, logisch jedoch um eine einzige Datei handelt. Es wird *nicht* unterstützt, wenn die Dateien sowohl logisch als auch physisch separate Dateien sind. (Mehrere physische Dateien werden als logische Dateien betrachtet, wenn alle Dateien in einem einzigen Aufruf des Befehls EXPORT erstellt wurden.)
3. Werden Daten geladen, die sich auf einer Clientmaschine befinden, müssen die Daten entweder in Form einer vollständig qualifizierten Datei oder einer benannten Pipe vorliegen.

#### **INDEXING MODE**

Gibt an, ob das Dienstprogramm LOAD Indizes erneut erstellen oder sie inkrementell erweitern soll. Gültige Werte:

##### **AUTOSELECT**

Das Dienstprogramm LOAD wählt automatisch den Modus REBUILD oder INCREMENTAL aus.

##### **REBUILD**

Alle Indizes werden erneut erstellt. Das Dienstprogramm benötigt ausreichende Ressourcen, um alle Indexschlüsselteile sowohl für alte als auch angefügte Tabellendaten zu sortieren.

##### **INCREMENTAL**

Indizes werden mit neuen Daten erweitert. Dieser Ansatz verbraucht freien Indexspeicherbereich. Er erfordert nur genug Sortierspeicherplatz, um Indexschlüssel für die eingefügten Datensätze anzufügen. Diese Methode wird nur in Fällen unterstützt, in denen das Indexobjekt bei Beginn einer Ladeoperation gültig ist und dar-

## Befehl LOAD

auf zugegriffen werden kann (es ist zum Beispiel unmittelbar nach einer Ladeoperation, bei der der Modus DEFERRED angegeben wurde, nicht gültig). Wenn dieser Modus angegeben ist, jedoch aufgrund des Status des Index nicht unterstützt wird, wird eine Warnung zurückgegeben, und die Ladeoperation wird im Modus REBUILD fortgesetzt. Ähnliches gilt, wenn eine LOAD RESTART-Operation während der LOAD BUILD-Phase gestartet wird. In diesem Fall wird der Modus INCREMENTAL nicht unterstützt.

Die inkrementelle Indexierung wird nicht unterstützt, wenn alle folgenden Bedingungen zutreffen:

- Die Option LOAD COPY wurde angegeben (*logretain* oder *userexit* ist aktiv).
- Die Tabelle befindet sich in einem DMS-Tabellenbereich.
- Das Indexobjekt befindet sich in einem Tabellenbereich, der mit anderen Tabellenobjekten gemeinsam benutzt wird, die zu der geladenen Tabelle gehören.

Um diese Einschränkung zu umgehen, wird empfohlen, die Indizes in einen separaten Tabellenbereich zu stellen.

### DEFERRED

Das Dienstprogramm LOAD erstellt keine Indizes, wenn dieser Modus angegeben wird. Die Indizes werden als aktualisierungsbedürftig gekennzeichnet. Der erste Zugriff auf solche Indizes, der nicht im Zusammenhang mit einer Ladeoperation steht, kann zwangsweise zu einer erneuten Erstellung führen, oder Indizes können erneut erstellt werden, wenn die Datenbank erneut gestartet wird. Dieser Ansatz erfordert genug Sortierspeicherplatz für alle Schlüsselteile für den größten Index. Die Gesamtzeit, die für den Aufbau des Index benötigt wird, ist länger als im Modus REBUILD erforderlich. Unter Leistungsgesichtspunkten ist es daher bei der Durchführung mehrerer Ladeoperationen mit verzögerter Indexierung ratsam, bei der letzten Ladeoperation eine erneute Erstellung des Indexes durchführen zu lassen, statt die Indizes beim ersten, nicht mit der Ladeoperation im Zusammenhang stehenden Zugriff erneut erstellen zu lassen.

Die verzögerte Indexierung wird nur für Tabellen mit nicht eindeutigen Indizes unterstützt, sodass doppelte Schlüssel, die während der LOAD-Phase eingefügt werden, nach der Ladeoperation nicht erhalten bleiben.

**Anmerkung:** Bei Tabellen mit DATALINK-Spalten wird die verzögerte Indexierung nicht unterstützt.

### INSERT

Einer von vier Modi, in denen das Dienstprogramm LOAD ausgeführt werden kann. Fügt die geladenen Daten der Tabelle hinzu, ohne die vorhandenen Tabellendaten zu ändern.

#### einfügespalte

Gibt die Tabellenspalte an, in die die Daten eingefügt werden sollen.

Das Dienstprogramm LOAD kann keine Spalten syntaktisch analysieren, deren Namen Leerzeichen enthalten. Beispiel:

```
db2 load from delfile1 of del modified by noeofchar 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)
```

Dieser Befehl schlägt aufgrund der Spalte Int 4 fehl. Die Lösung besteht darin, solche Spaltennamen in doppelte Anführungszeichen einzuschließen:

```
db2 load from delfile1 of del modified by noeofchar 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)
```

#### INTO tabellenname

Gibt die Datenbanktabelle an, in die die Daten geladen werden sollen. Diese Tabelle darf keine Systemtabelle oder deklarierte temporäre Tabelle sein. Es kann ein Aliasname, der vollständig qualifizierte Name oder ein Name ohne Qualifikationsmerkmal angegeben werden. Ein qualifizierter Tabellename hat das Format *schema.tabellenname*. Wenn ein Tabellename ohne Qualifikationsmerkmal angegeben wird, wird die Tabelle mit dem aktuellen Schema (CURRENT SCHEMA) qualifiziert.

#### LOBS FROM lob-pfad

Der Pfad zu den Datendateien mit LOB-Werten, die geladen werden sollen. Der Pfad muss mit einem Schrägstrich (/) enden. Falls die Option CLIENT angegeben wird, muss es sich um einen vollständig qualifizierten Pfad handeln. Die Namen der LOB-Datendateien werden in der Hauptdaten-datei (ASC, DEL oder IXF) in der Spalte gespeichert, die in die LOB-Spalte geladen wird. Diese Option wird ignoriert, wenn lobsinfile nicht in der Zeichenfolge *änderungswert-für-dateityp* angegeben wird.

Die Verwendung dieser Option in Verbindung mit dem Dateityp CURSOR wird nicht unterstützt.

#### LOCK WITH FORCE

Während des Ladeprozesses erhält das Dienstprogramm unterschiedliche Sperren (inklusive Tabellensperren). Um sicherzustellen, dass kein Wartestatus und keine mögliche Zeitlimitüberschreitung eintritt, kann durch die Verwendung dieser Option die Beendigung von anderen Anwendungen erzwungen werden, die Konflikte verursachende Sperren für die Zieltabelle besitzen definiert haben. Anwendungen, die Konflikte verursachende Sperren für die Systemkatalogtabellen besitzen, werden nicht durch das Dienstprogramm LOAD erzwungen abgemeldet. Diese Anwendungen werden zurückgesetzt und geben die vom Dienstprogramm LOAD benötigten Sperren frei. Anschließend kann das Dienstprogramm LOAD fortfahren. Für diese Option wird dieselbe Berechtigung wie für den Befehl FORCE APPLICATIONS benötigt (SYSADM oder SYSCTRL).

Ladevorgänge im Modus "Keinen Zugriff zulassen" (ALLOW NO ACCESS) können die Beendigung von Anwendungen, die Konflikte verursachende Sperren halten, am Beginn der Ladeoperation erzwingen. Am Anfang des Ladevorgangs kann das Dienstprogramm die Beendigung von Anwendungen erzwingen, die versuchen, die Tabelle entweder abzufragen oder zu ändern.

Ladevorgänge im Modus "Lesezugriff zulassen" (ALLOW READ ACCESS) können die Beendigung von Anwendungen, die Konflikte verursachende Sperren halten, am Beginn oder am Ende der Ladeoperation erzwingen. Am Anfang des Ladevorgangs kann das Dienstprogramm LOAD die Beendigung von Anwendungen erzwingen, die versuchen, die Tabelle zu ändern. Am Ende des Ladevorgangs kann das Dienstprogramm LOAD die Beendigung von Anwendungen erzwingen, die versuchen, die Tabelle entweder abzufragen oder zu ändern.

#### MESSAGES nachrichtendatei

Gibt das Ziel für Warnungen und Fehlermeldungen an, die während der Ladeoperation auftreten. Wenn keine Nachrichtendatei angegeben wird,

## Befehl LOAD

werden die Nachrichten in die Standardausgabe geschrieben. Wenn der vollständige Pfad zu der Datei nicht angegeben wird, verwendet das Dienstprogramm LOAD das aktuelle Verzeichnis und das Standardlaufwerk als Ziel. Wenn der Name einer Datei, die bereits vorhanden ist, angegeben wird, fügt das Dienstprogramm die Informationen an.

Die Nachrichtendatei wird für gewöhnlich am Ende der Ladeoperation mit Nachrichten gefüllt und ist insofern nicht geeignet, um den Fortschritt der Operation zu überwachen.

### METHOD

**L** Gibt die Anfangs- und Endnummer der Spalten an, aus denen Daten geladen werden sollen. Eine Spaltennummer ist eine relative Byteadresse in Bezug auf den Anfang einer Datenzeile. Ihre Nummerierung beginnt bei 1.

**Anmerkung:** Diese Methode kann nur mit ASC-Dateien verwendet werden und ist die einzige gültige Methode für diesen Dateityp.

**N** Gibt die Namen der Spalten in der zu ladenden Datendatei an. Die Schreibweise dieser Spaltennamen muss mit der Schreibweise der entsprechenden Namen in den Systemkatalogen übereinstimmen. Für jede Tabellenspalte, die keine Nullwerte enthalten kann, muss ein entsprechender Eintrag in der Liste METHOD N vorhanden sein. Beispiel: Für die Datenfelder F1, F2, F3, F4, F5 und F6 sowie die Tabellenspalten C1 INT, C2 INT NOT NULL, C3 INT NOT NULL und C4 INT ist method N (F2, F1, F4, F3) eine gültige Anforderung, wohingegen method N (F2, F1) nicht gültig ist.

**Anmerkung:** Diese Methode kann nur mit den Dateitypen IXF oder CURSOR verwendet werden.

**P** Gibt die Feldnummern (nummeriert beginnend bei 1) der zu ladenden Eingabedatenfelder an. Für jede Tabellenspalte, die keine Nullwerte enthalten kann, muss ein entsprechender Eintrag in der Liste METHOD P vorhanden sein. Beispiel: Für die Datenfelder F1, F2, F3, F4, F5 und F6 sowie die Tabellenspalten C1 INT, C2 INT NOT NULL, C3 INT NOT NULL und C4 INT ist method P (2, 1, 4, 3) eine gültige Anforderung, wohingegen method P (2, 1) nicht gültig ist.

**Anmerkung:** Diese Methode kann nur bei den Dateitypen IXF, DEL oder CURSOR verwendet werden und ist die einzige zulässige Methode für den Dateityp DEL.

### MODIFIED BY **änderungswert-für-dateityp**

Gibt Optionen für Änderungswerte für Dateitypen an. Weitere Informationen hierzu finden Sie in File type modifiers for load.

### NONRECOVERABLE

Gibt an, dass eine Ladetransaktion als nicht wiederherstellbar gekennzeichnet werden soll und dass sie nicht durch eine nachfolgende aktualisierende Wiederherstellung wiedergestellt werden kann. Das Dienstprogramm zur aktualisierenden Wiederherstellung überspringt die Transaktion und kennzeichnet die Tabelle, in die die Daten geladen wurden, als "ungültig". Vom Dienstprogramm werden auch alle nachfolgenden Transaktionen für diese Tabelle ignoriert. Nach Abschluss der Operation für die aktualisierende

Wiederherstellung kann eine solche Tabelle nur nach einem COMMIT-Punkt, der auf den Abschluss der nicht wiederherstellbaren Ladeoperation folgt, gelöscht oder aus einer Sicherung (vollständige Sicherung oder Sicherung des Tabellenbereichs) wiederhergestellt werden.

Mit dieser Option werden Tabellenbereiche nach der Ladeoperation nicht in den Status "Sicherung anstehend" versetzt, und während der Ladeoperation muss keine Kopie der geladenen Daten erstellt werden.

Diese Option sollte nicht verwendet werden, wenn DATALINK-Spalten mit dem Attribut FILE LINK CONTROL in der Tabelle vorhanden sind oder ihr hinzugefügt werden.

#### NULL INDICATORS nullanzeigerliste

Diese Option kann nur verwendet werden, wenn der Parameter METHOD L angegeben ist, d. h. wenn es sich bei der Eingabedatei um eine ASC-Datei handelt. Die Nullanzeigerliste ist eine durch Kommata gegliederte Liste positiver ganzer Zahlen, die für jedes Nullanzeigerfeld die Spaltennummer angeben. Die Spaltennummer ist die relative Byteadresse des Nullanzeigerfelds in Bezug auf den Anfang einer Datenzeile. Die Nullanzeigerliste muss für jedes Datenfeld, das im Parameter METHOD L definiert ist, einen Wert enthalten. Eine Spaltennummer 0 gibt an, dass das entsprechende Datenfeld immer Daten enthält.

Y in der Nullanzeigerspalte gibt an, dass die Spaltendaten NULL sind. Jedes *andere* Zeichen als Y in der Nullanzeigerspalte gibt an, dass die Spaltendaten nicht NULL sind und dass die mit der Option METHOD L angegebenen Spaltendaten geladen werden.

Das Nullanzeigerzeichen kann mit der Option MODIFIED BY geändert werden.

#### OF dateityp

Gibt das Format der Daten an:

- ASC (ASCII-Format mit universellen Zeilenbegrenzern)
- DEL (ASCII-Format ohne universelle Zeilenbegrenzer)
- IXF (Integrated Exchange Format, PC-Version), aus derselben oder einer anderen DB2-Tabelle exportiert
- CURSOR (ein Cursor, der für eine SELECT-Anweisung oder VALUES deklariert wurde)

#### PARTITIONED DB CONFIG

Dieser Parameter ermöglicht das Laden von Daten in eine partitionierte Tabelle. Mit dem Parameter PARTITIONED DB CONFIG können Sie außerdem spezifische Konfigurationsoptionen für partitionierte Datenbanken angeben. Als *option\_für\_partitionierte\_datenbank* kann Folgendes angegeben werden:

```

HOSTNAME x
FILE_TRANSFER_CMD x
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

```

## Befehl LOAD

```
TRACE x  
NEWLINE  
DISTFILE x  
OMIT_HEADER  
RUN_STAT_DBPARTNUM x
```

Eine ausführliche Beschreibung dieser Optionen steht in folgendem Abschnitt zur Verfügung: LOAD-Konfigurationsoptionen für partitionierte Datenbanken.

### REPLACE

Einer von vier Modi, in denen das Dienstprogramm LOAD ausgeführt werden kann. Löscht alle vorhandenen Daten aus der Tabelle und fügt die importierten Daten ein. Die Tabellendefinition und die Indexdefinitionen werden nicht verändert. Wenn diese Option beim Versetzen von Daten zwischen Hierarchien verwendet wird, können nur die Daten für eine vollständige Hierarchie, nicht jedoch für einzelne untergeordnete Tabellen ersetzt werden.

Diese Option wird für Tabellen mit DATALINK-Spalten nicht unterstützt.

### RESTART

Einer von vier Modi, in denen das Dienstprogramm LOAD ausgeführt werden kann. Startet eine zuvor unterbrochene Ladeoperation erneut. Die Ladeoperation wird automatisch vom letzten Konsistenzzustand in der LOAD-, BUILD- oder DELETE-Phase ausgehend fortgesetzt.

### RESTARTCOUNT

Reserviert.

### ROWCOUNT *n*

Gibt die Anzahl von *n* physischen Sätzen in der Datei an, die geladen werden sollen. Ermöglicht es einem Benutzer, nur die ersten *n* Zeilen in einer Datei zu laden.

### SAVECOUNT *n*

Gibt an, dass das Dienstprogramm LOAD alle *n* Zeilen Konsistenzzustände einrichten soll. Dieser Wert wird in einen Seitenzähler konvertiert und auf Intervalle in der Größe des Speicherbereichs gerundet. Da bei jedem Konsistenzzustand eine Nachricht abgesetzt wird, sollte diese Option ausgewählt werden, wenn die Ladeoperation mit LOAD QUERY überwacht wird. Wenn der Wert *n* nicht groß genug ist, wirkt sich die Synchronisation der Aktivitäten, die bei jedem Konsistenzzustand ausgeführt werden, nachteilig auf die Leistung aus.

Der Standardwert ist Null, d. h., dass Konsistenzzustände nur definiert werden, sofern dies notwendig ist.

Die Verwendung dieser Option in Verbindung mit dem Dateityp CURSOR wird nicht unterstützt.

### SORT BUFFER *puffergröße*

Mit dieser Option wird ein Wert angegeben, der den Datenbankkonfigurationsparameter SORTHEAP während einer Ladeoperation außer Kraft setzt. Sie ist nur dann von Bedeutung, wenn Tabellen mit Indizes geladen werden und für den Parameter INDEXING MODE ein anderer Wert als DEFERRED angegeben ist. Der angegebene Wert kann den Wert des Parameters SORTHEAP nicht überschreiten. Dieser Parameter ist nützlich, um den beim Laden von Tabellen mit vielen Indizes verwendeten Sortierspeicher zu verkleinern, ohne den Wert für SORTHEAP ändern zu müssen, der sich auch auf die allgemeine Abfrageverarbeitung auswirkt.

**STATISTICS USE PROFILE**

Weist das Dienstprogramm LOAD an, während des Ladens Statistiken gemäß dem für diese Tabelle definierten Profils zu erfassen. Dieses Profil muss vor dem Ausführen von LOAD erstellt werden. Zum Erstellen des Profils wird der Befehl RUNSTATS verwendet. Ist das Profil nicht vorhanden und die LOAD wird aufgefordert, Statistiken gemäß dem Profil zu erfassen, wird eine Warnung zurückgegeben und es werden keine Statistiken erfasst.

**STATISTICS NO**

Gibt an, dass keine Statistikdaten erfasst werden sollen und dass die Statistikdaten in den Katalogen nicht geändert werden dürfen. Dies ist der Standardwert.

**TEMPFILES PATH temp-pfadname**

Gibt den Namen des Pfads an, der beim Erstellen von temporären Dateien während einer Ladeoperation verwendet werden soll. Er muss bezogen auf die Serverdatenbankpartition vollständig qualifiziert sein.

Temporäre Dateien belegen Speicherplatz im Dateisystem. Manchmal ist dieser Platzbedarf beträchtlich. Im Folgenden finden Sie Richtwerte dazu, wie viel Dateisystemplatz für alle temporären Dateien zugeordnet werden sollte:

- 4 Byte für jede doppelte oder zurückgewiesene Zeile, die DATALINK-Werte enthält
- 136 Byte für jede Nachricht, die vom Dienstprogramm LOAD generiert wird
- 15 KB zusätzlich, wenn die Datendatei Langfelddaten oder LOBs enthält. Diese Menge kann erheblich anwachsen, wenn die Option INSERT angegeben wird und sich bereits eine große Menge von Langfeld- oder LOB-Daten in der Tabelle befindet.

**TERMINATE**

Einer von vier Modi, in denen das Dienstprogramm LOAD ausgeführt werden kann. Beendet eine zuvor unterbrochene Ladeoperation und macht die Operation bis zu dem Zeitpunkt, zu dem sie gestartet wurde, rückgängig, auch wenn Konsistenzzustände übergangen wurden. Die Statuswerte von betroffenen Tabellenbereichen werden in "Normal" zurückgeändert, und alle Tabellenobjekte werden konsistent gemacht (Indexobjekte können als ungültig gekennzeichnet werden, so dass beim nächsten Zugriff automatisch eine Indexwiederherstellung erfolgt). Wenn die beendete Ladeoperation eine LOAD REPLACE-Operation ist, wird die Tabelle nach der LOAD TERMINATE-Operation abgeschnitten, so dass eine leere Tabelle übrig bleibt. Wenn die beendete Ladeoperation eine LOAD INSERT-Operation ist, bleiben in der Tabelle nach der LOAD TERMINATE-Operation alle ursprünglich vorhandenen Datensätze erhalten.

Mit der Option LOAD TERMINATE wird der Status "Sicherung anstehend" (BACKUP PENDING) für Tabellenbereiche nicht aufgehoben.

**Anmerkung:** Diese Option wird für Tabellen mit DATALINK-Spalten nicht unterstützt.

**USING verzeichnis**

Reserviert.

**WARNINGCOUNT n**

Stoppt die Ladeoperation nach *n* Warnungen. Definieren Sie diesen Parameter, wenn keine Warnungen erwartet werden, jedoch eine Prüfung dar-

## Befehl LOAD

auf, ob die richtige Datei und Tabelle verwendet wird, gewünscht wird. Wird die Ladedatei oder die Zieltabelle falsch angegeben, generiert das Dienstprogramm LOAD eine Warnung für jede Zeile, die es versucht zu laden, sodass der Ladevorgang fehlschlägt. Wenn  $n$  Null ist oder diese Option nicht angegeben wird, wird die Ladeoperation unabhängig von der Anzahl der abgesetzten Warnungen fortgesetzt. Wenn die Ladeoperation gestoppt wird, da die Schwelle für Warnungen erreicht wurde, kann eine weitere Ladeoperation im Neustartmodus (RESTART) gestartet werden. Die Ladeoperation wird automatisch vom letzten Konsistenzzustand ausgehend fortgesetzt. Alternativ kann eine weitere Ladeoperation im Ersetzungsmodus (REPLACE) eingeleitet werden, die am Anfang der Eingabedatei beginnt.

### WITHOUT PROMPTING

Gibt an, dass die Liste mit Datendateien alle Dateien enthält, die geladen werden sollen, und dass die aufgelisteten Einheiten oder Verzeichnisse für die gesamte Ladeoperation genügen. Wenn keine weitere Eingabedatei gefunden wird oder die Kopierziele gefüllt sind, bevor die Ladeoperation beendet ist, schlägt die Ladeoperation fehl, und die Tabelle verbleibt im Status "Laden anstehend".

Wenn diese Option nicht angegeben wird und beim Bandlaufwerk für das Kopierabbild das Bandende erreicht wird oder das letzte aufgelistete Element ein Bandlaufwerk ist, wird der Benutzer aufgefordert, ein neues Band in das betreffende Laufwerk einzulegen.

### Beispiele:

#### Beispiel 1

TABLE1 enthält 5 Spalten:

- COL1 VARCHAR 20 NOT NULL WITH DEFAULT
- COL2 SMALLINT
- COL3 CHAR 4
- COL4 CHAR 2 NOT NULL WITH DEFAULT
- COL5 CHAR 2 NOT NULL

ASCFILE1 enthält 6 Elemente:

- ELE1 Positionen 01 bis 20
- ELE2 Positionen 21 bis 22
- ELE5 Positionen 23 bis 23
- ELE3 Positionen 24 bis 27
- ELE4 Positionen 28 bis 31
- ELE6 Positionen 32 bis 32
- ELE6 Positionen 33 bis 40

Datensätze:

```
1...5...10...15...20...25...30...35...40
Testdaten 1          XXN 123abcdN
Testdaten 2 und 3   QQY   wxyzN
Testdaten 4,5 und 6 WVN6789  Y
```

Mit dem folgenden Befehl wird die Tabelle aus der Datei geladen:

```
db2 load from ascfile1 of asc modified by striptblanks reflen=40
method L (1 20, 21 22, 24 27, 28 31)
null indicators (0,0,23,32)
insert into table1 (col1, col5, col2, col3)
```

**Anmerkungen:**

1. Die Angabe `striptblanks` im Parameter `MODIFIED BY` bewirkt, dass Leerzeichen in `VARCHAR`-Spalten (zum Beispiel `COL1`, die 11, 17 und 19 Byte lang ist, in den Zeilen 1, 2 bzw. 3) zwangsweise abgeschnitten werden.
2. Die Angabe `reflen=40` im Parameter `MODIFIED BY` legt fest, dass es am Ende der Eingabedatensätze kein Zeilenvorschubzeichen gibt und jeder Datensatz 40 Byte lang ist. Die letzten 8 Byte werden nicht zum Laden der Tabelle verwendet.
3. Da die Spalte `COL4` nicht in der Eingabedatei bereitgestellt wird, wird sie in die Tabelle `TABLE1` mit dem Standardwert eingefügt (sie ist mit `NOT NULL WITH DEFAULT` definiert).
4. Die Positionen 23 und 32 werden verwendet, um anzugeben, ob in `COL2` und `COL3` von `TABLE1` für eine gegebene Zeile `NULL` geladen wird. Wenn die Nullanzeigerposition der Spalte für einen gegebenen Satz `Y` enthält, ist die Spalte `NULL`. Enthält sie `N`, werden die Datenwerte in den Datenpositionen der Spalte des Eingabedatensatzes (wie in `L(.....)` definiert) als Quelle der Spalten-`daten` für die Zeile verwendet. In diesem Beispiel ist keine Spalte in Zeile 1 `NULL`. `COL2` in Zeile 2 ist `NULL`, und `COL3` in Zeile 3 ist `NULL`.
5. In diesem Beispiel werden die `NULL`-Anzeiger für `COL1` und `COL5` als 0 (Null) angegeben, wodurch festgelegt wird, dass die Daten keine Nullwerte enthalten dürfen.
6. Der Nullanzeiger (`NULL INDICATOR`) für eine bestimmte Spalte kann sich an einer beliebigen Position im Eingabedatensatz befinden. Diese Position muss jedoch angegeben werden, ebenso wie der Wert `Y` oder `N`.

**Beispiel 2 (Laden von großen Objekten aus Dateien)**

`TABLE1` enthält 3 Spalten:

- `COL1 CHAR 4 NOT NULL WITH DEFAULT`
- `LOB1 LOB`
- `LOB2 LOB`

`ASCFILE1` enthält 3 Elemente:

- `ELE1` Positionen 01 bis 04
- `ELE2` Positionen 06 bis 13
- `ELE3` Positionen 15 bis 22

Die folgenden Dateien befinden sich in `/u/user1` oder `/u/user1/bin`:

- `ASCFILE2` enthält `LOB`-Daten
- `ASCFILE3` enthält `LOB`-Daten
- `ASCFILE4` enthält `LOB`-Daten
- `ASCFILE5` enthält `LOB`-Daten
- `ASCFILE6` enthält `LOB`-Daten
- `ASCFILE7` enthält `LOB`-Daten

## Befehl LOAD

Datensätze in ASCFILE1:

```
1...5...10...15...20...25...30.  
REC1 ASCFILE2 ASCFILE3  
REC2 ASCFILE4 ASCFILE5  
REC3 ASCFILE6 ASCFILE7
```

Mit dem folgenden Befehl wird die Tabelle aus der Datei geladen:

```
db2 load from ascfile1 of asc  
lobs from /u/user1, /u/user1/bin  
modified by lobsinfile reclen=22  
method L (1 4, 6 13, 15 22)  
insert into table1
```

### Anmerkungen:

1. Die Angabe lobsinfile im Parameter MODIFIED BY teilt dem Ladeprogramm mit, dass alle LOB-Daten aus Dateien geladen werden sollen.
2. Die Angabe reclen=22 im Parameter MODIFIED BY legt fest, dass es am Ende der Eingabedatensätze kein Zeilenvorschubzeichen gibt und jeder Datensatz 22 Byte lang ist.
3. LOB-Daten sind in 6 Dateien enthalten, ASCFILE2 bis ASCFILE7. Jede Datei enthält die Daten, die verwendet werden, um eine LOB-Spalte für eine bestimmte Zeile zu laden. Die Beziehung zwischen großen Objekten und anderen Daten wird in ASCFILE1 angegeben. Der erste Satz dieser Datei weist das Ladeprogramm an, REC1 in COL1 von Zeile 1 zu stellen. Der Inhalt von ASCFILE2 wird verwendet, um LOB1 von Zeile 1 zu laden, und der Inhalt von ASCFILE3 wird verwendet, um LOB2 von Zeile 1 zu laden. ASCFILE4 und ASCFILE5 werden wiederum verwendet, um LOB1 und LOB2 von Zeile 2 zu laden, ASCFILE6 und ASCFILE7 werden verwendet, um die großen Objekte (LOBs) von Zeile 3 zu laden.
4. Der Parameter LOBS FROM enthält zwei Pfade, die nach den benannten LOB-Dateien durchsucht werden, wenn diese Dateien vom Ladeprogramm angefordert werden.
5. Zum Laden von großen Objekten aus ASCFILE1 (einer ASCII-Datei mit universellen Zeilenbegrenzern) ohne den Änderungswert lobsinfile müssen die folgenden Regeln eingehalten werden:
  - Die Gesamtlänge eines jeden Datensatzes, einschließlich großer Objekte, darf 32 KB nicht überschreiten.
  - LOB-Felder in Eingabedatensätzen müssen eine feste Länge haben, und LOB-Daten müssen gegebenenfalls mit Leerzeichen aufgefüllt werden.
  - Der Änderungswert striptblanks muss angegeben werden, damit die abschließenden Leerzeichen, die zum Auffüllen von großen Objekten verwendet werden, entfernt werden können, wenn die großen Objekte in die Datenbank eingefügt werden.

### Beispiel 3 (unter Verwendung von Speicherauszugsdateien)

Tabelle FRIENDS ist wie folgt definiert:

```
table friends "( c1 INT NOT NULL, c2 INT, c3 CHAR(8) )"
```

Wenn versucht wird, die folgenden Datensätze in diese Tabelle zu laden:

```
23, 24, bobby  
, 45, john  
4,, mary
```

wird die zweite Zeile zurückgewiesen, weil das erste INT NULL ist und die Spaltendefinition NOT NULL lautet. Spalten, die Anfangszeichen enthalten, die nicht mit dem DEL-Format konsistent sind, führen zu einem Fehler, und der Datensatz wird zurückgewiesen. Solche Datensätze können in eine Speicherausgangsdatei geschrieben werden.

DEL-Daten, die in einer Spalte außerhalb von Zeichenbegrenzern vorkommen, werden ignoriert, führen jedoch zur Generierung einer Warnung. Beispiel:

```
22,34,"bob"
24,55,"sam" sdf
```

Das Dienstprogramm lädt "sam" in die dritte Spalte der Tabelle, und die Zeichen "sdf" werden in einer Warnung markiert. Der Datensatz wird nicht zurückgewiesen. Anderes Beispiel: 22 3, 34,"bob"

Das Dienstprogramm lädt 22,34,"bob" und generiert eine Warnung, dass einige Daten in Spalte eins nach 22 ignoriert wurden. Der Datensatz wird nicht zurückgewiesen.

#### Beispiel 4 (Laden von DATALINK-Daten)

Mit dem folgenden Befehl wird die Tabelle MOVIE TABLE aus der Eingabedatei delfile1 geladen, die Daten im DEL-Format enthält:

```
db2 load from delfile1 of del modified by dldel |
insert into movietable (actorname, description, url_making_of,
url_movie) datalink specification (dl_url_default_prefix
"http://narang"), (dl_url_replace_prefix "http://bomdel"
dl_url_suffix ".mpeg") for exception excptab
```

#### Anmerkungen:

1. Die Tabelle enthält vier Spalten:

actorname	VARCHAR(n)
description	VARCHAR(m)
url_making_of	DATALINK (mit LINKTYPE URL)
url_movie	DATALINK (with LINKTYPE URL)

2. Die DATALINK-Daten in der Eingabedatei verwenden den vertikalen Balken (|) als Teilfeldbegrenzer.

3. Wenn ein Spaltenwert für url\_making\_of keine Präfixzeichenfolge aufweist, wird "http://narang" verwendet.

4. Jeder Spaltenwert ungleich NULL für url\_movie erhält "http://bomdel" als Präfix. Vorhandene Werte werden ersetzt.

5. An jeden Spaltenwert ungleich NULL für url\_movie wird ".mpeg" an den Pfad angefügt. Beispiel: Wenn ein Spaltenwert für url\_movie "http://server1/x/y/z" lautet, wird er als "http://bomdel/x/y/z.mpeg" gespeichert; ist der Wert "/x/y/z", wird er als "http://bomdel/x/y/z.mpeg" gespeichert.

6. Wenn Ausnahmebedingungen für den eindeutigen Index oder DATALINK auftreten, während die Tabelle geladen wird, werden die betroffenen Datensätze aus der Tabelle gelöscht und in die Ausnahmetabelle excptab gestellt.

#### Beispiel 5 (Laden einer Tabelle mit einer Identitätsspalte)

TABLE1 enthält 4 Spalten:

- C1 VARCHAR(30)
- C2 INT GENERATED BY DEFAULT AS IDENTITY
- C3 DECIMAL(7,2)
- C4 CHAR(1)

## Befehl LOAD

TABLE2 entspricht TABLE1, wobei jedoch C2 eine als GENERATED ALWAYS definierte Identitätsspalte ist.

Die Datensätze in DATAFILE1 (DEL-Format) lauten:

```
"Liszt"  
"Holst",,187.43, H  
"Grieg",100, 66.34, G  
"Satie",101, 818.23, I
```

Die Datensätze in DATAFILE2 (DEL-Format) lauten:

```
"Liszt", 74.49, A  
"Holst", 0.01, H  
"Grieg", 66.34, G  
"Satie", 818.23, I
```

### Anmerkungen:

1. Der folgende Befehl generiert Identitätswerte für die Zeilen 1 und 2, da in der Datei DATAFILE1 für diese Zeilen keine Identitätswerte zur Verfügung gestellt werden. Den Zeilen 3 und 4 wird jedoch der benutzerdefinierte Identitätswert 100 bzw. 101 zugeordnet.

```
db2 load from datafile1.del of del replace into table1
```

2. Um die Datei DATAFILE1 so in die Tabelle TABLE1 zu laden, dass für alle Zeilen Identitätswerte generiert werden, setzen Sie einen der folgenden Befehle ab:

```
db2 load from datafile1.del of del method P(1, 3, 4)  
replace into table1 (c1, c3, c4)  
db2load from datafile1.del of del modified by identityignore  
replace into table1
```

3. Um die Datei DATAFILE2 so in die Tabelle TABLE1 zu laden, dass für alle Zeilen Identitätswerte generiert werden, setzen Sie einen der folgenden Befehle ab:

```
db2 load from datafile2.del of del replace into table1 (c1, c3, c4)  
db2 load from datafile2.del of del modified by identitymissing  
replace into table1
```

4. Um die Datei DATAFILE1 so in die Tabelle TABLE2 zu laden, dass den Zeilen 3 und 4 die Identitätswerte 100 bzw. 101 zugeordnet werden, setzen Sie den folgenden Befehl ab:

```
db2 load from datafile1.del of del modified by identityoverride  
replace into table2
```

In diesem Fall werden die Zeilen 1 und 2 zurückgewiesen, da das Dienstprogramm angewiesen wurde, die durch das System generierten Identitätswerte mit den benutzerdefinierten Werten zu überschreiben. Sind jedoch keine benutzerdefinierten Werte vorhanden, muss die Zeile zurückgewiesen werden, da Identitätsspalten implizit nicht NULL sind.

5. Wird die Datei DATAFILE1 in die Tabelle TABLE2 geladen, ohne dass einer der identitätsbezogenen Änderungswerte für Dateityp verwendet wird, werden die Zeilen 1 und 2 geladen, die Zeilen 3 und 4 jedoch zurückgewiesen, da diese Zeilen eigene Werte, die nicht NULL sind, liefern und die Identitätsspalte als GENERATED ALWAYS definiert wurde.

### Beispiel 6 (Laden unter Verwendung des Dateityps CURSOR)

Die Tabelle ABC.TABLE1 enthält 3 Spalten:

```
ONE INT  
TWO CHAR(10)  
THREE DATE
```

Die Tabelle ABC.TABLE2 enthält 3 Spalten:

```
ONE VARCHAR
TWO INT
THREE DATE
```

Durch Ausführung der folgenden Befehle werden alle Daten aus der Tabelle ABC.TABLE1 in die Tabelle ABC.TABLE2 geladen:

```
db2 declare mycurs cursor for select two,one,three from abc.table1
db2 load from mycurs of cursor insert into abc.table2
```

#### Anmerkungen zur Verwendung:

Die Daten werden in der Reihenfolge geladen, die in der Eingabedatei vorliegt. Falls eine bestimmte Reihenfolge gewünscht wird, müssen die Daten vor dem Durchführen der Ladeoperation sortiert werden.

Das Dienstprogramm LOAD erstellt Indizes gemäß den vorhandenen Definitionen. Mit Hilfe von Ausnahmetabellen werden doppelt vorhandene Werte für eindeutige Schlüssel verarbeitet. Das Dienstprogramm nimmt für Übersichtstabellen, die von den geladenen Tabellen abhängig sind, weder eine Aktualisierung vor, noch setzt es die referenzielle Integrität durch oder überprüft die Integritätsbedingungen. Tabellen, die referenzielle oder zu prüfende Integritätsbedingungen enthalten, werden in den Status "Überprüfung anstehend" versetzt. Übersichtstabellen, für die das Attribut REFRESH IMMEDIATE definiert ist und die von geladenen Tabellen abhängig sind, werden ebenfalls in den Status "Überprüfung anstehend" versetzt. Durch Absetzen der Anweisung SET INTEGRITY können Sie den Status "Überprüfung anstehend" für die Tabellen aufheben. Die Ausführung von Ladeoperationen für replizierte Übersichtstabellen ist nicht möglich.

Falls ein Clusterindex für die Tabelle vorhanden ist, müssen die Daten vor dem Laden nach dem Clusterindex sortiert werden. Vor dem Laden von Daten in eine MDC-Tabelle ist jedoch keine Sortierung erforderlich.

#### Hinweise zu DB2 Data Links Manager:

Für jede DATALINK-Spalte ist eine Spaltenspezifikation in runden Klammern möglich. Jede Spaltenspezifikation umfasst mindestens einen DL\_LINKTYPE-Wert, ein *präfix* und eine DL\_URL\_SUFFIX-Angabe. Der Wert für *präfix* kann entweder DL\_URL\_REPLACE\_PREFIX oder die Spezifikation aus DL\_URL\_DEFAULT\_PREFIX sein.

Es sind so viele DATALINK-Spaltenspezifikationen möglich, wie DATALINK-Spalten in der Tabelle definiert sind. Die Reihenfolge der Spezifikationen richtet sich nach der Reihenfolge der DATALINK-Spalten in der Liste der Einfügespalten (sofern diese durch INSERT INTO (*einfügespalte*, ...) angegeben wurde) oder in der Tabellendefinition (wenn keine Liste für *einfügespalte* angegeben wird).

Wenn eine Tabelle beispielsweise die Spalten C1, C2, C3, C4 und C5 enthält, von denen nur C2 und C5 Spalten des Typs DATALINK sind, und die Liste der Einfügespalten (C1, C5, C3, C2) lautet, sollte es zwei Spezifikationen für DATALINK-Spalten geben. Die erste Spaltenspezifikation gilt für C5, die zweite für C2. Falls keine Liste der Einfügespalten angegeben ist, gilt die erste Spaltenspezifikation für C2 und die zweite für C5.

## Befehl LOAD

Wenn mehrere DATALINK-Spalten vorhanden sind und manche Spalten keine bestimmte Spezifikation erfordern, sollte die Spaltenspezifikation mindestens die runden Klammern enthalten, damit die Reihenfolge der Spezifikationen eindeutig erkennbar ist. Sind für keine der Spalten Spezifikationen vorhanden, kann die gesamte Liste der leeren Klammern gelöscht werden. Daher ist in Fällen, in denen die Standardwerte ausreichend sind, keine DATALINK-Spezifikation erforderlich.

Wenn Sie Daten in eine Tabelle mit einer DATALINK-Spalte laden, die mit FILE LINK CONTROL definiert ist, führen Sie die folgenden Schritte aus, bevor Sie das Dienstprogramm LOAD aufrufen. (Die folgenden Schritte sind nicht erforderlich, falls alle DATALINK-Spalten mit NO LINK CONTROL definiert sind.)

1. Stellen Sie sicher, dass DB2 Data Links Manager auf den Data Links-Servern, auf die die DATALINK-Spaltenwerte verweisen, installiert ist.
2. Stellen Sie sicher, dass die Datenbank in DB2 Data Links Manager registriert ist.
3. Kopieren Sie alle Dateien, die als DATALINK-Werte eingefügt werden, auf die entsprechenden Data Links-Server.
4. Definieren Sie für die DB2 Data Links Manager auf den Data Links-Servern den bzw. die Präfixnamen.
5. Registrieren Sie die Data Links-Server, auf die in den (zu ladenden) DATALINK-Daten verwiesen wird, in der Konfigurationsdatei von DB2 Data Links Manager.

Es kann vorkommen, dass die Verbindung zwischen DB2 und dem Data Links-Server während der Ausführung des Dienstprogramms LOAD fehlschlägt, wodurch die Ladeoperation fehlschlägt. Gehen Sie in diesem Fall wie folgt vor:

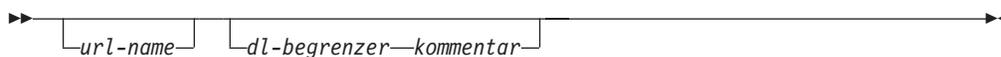
1. Starten Sie den Data Links-Server und den DB2 Data Links Manager.
2. Rufen Sie eine LOAD RESTART-Operation auf.

Während der Ladeoperation fehlgeschlagene Verbindungen werden als eine Art von Datenintegritätsverletzung eingestuft und wie Verletzungen eindeutiger Indizes gehandhabt. Es wurde daher eine besondere Ausnahmeregelung für das Laden von Tabellen definiert, die mindestens eine DATALINK-Spalte enthalten.

### Darstellung von DATALINK-Informationen in einer Eingabedatei

Der LINKTYPE-Wert (gegenwärtig wird nur URL unterstützt) wird nicht als Teil der DATALINK-Informationen angegeben. Der LINKTYPE-Wert wird im Befehl LOAD oder IMPORT angegeben sowie bei Eingabedateien des Typs PC/IXF in den entsprechenden Spaltendeskriptorsätzen.

DATALINK-Informationen für den LINKTYPE-Wert URL haben die folgende Syntax:



Bitte beachten Sie, dass die Angabe von Werten für *url-name* und *kommentar* optional ist. Wenn keiner der beiden Werte angegeben ist, wird der Wert NULL zugeordnet.

#### **url-name**

Der URL-Name muss die gültige URL-Syntax beachten.

**Anmerkungen:**

1. Gegenwärtig sind die Werte "http", "file" und "unc" als Schemaname zulässig.
2. Das Präfix (Schema, Host und Port) des URL-Namens ist optional. Wenn kein Präfix vorhanden ist, wird es aus der Spezifikation DL\_URL\_DEFAULT\_PREFIX oder DL\_URL\_REPLACE\_PREFIX für das Dienstprogramm LOAD bzw. das Dienstprogramm IMPORT übernommen. Sind diese nicht angegeben, nimmt das Präfix standardmäßig den Wert "file://localhost" an. Daher kann bei lokalen Dateien der Dateiname mit dem vollständigen Pfadnamen als URL-Name eingegeben werden, ohne dass eine Spezifikation für DATALINK-Spalten im Befehl LOAD oder IMPORT angegeben sein muss.
3. Auch wenn Präfixe in URL-Namen vorhanden sind, werden sie während einer Lade- oder Importoperation außer Kraft gesetzt, falls die Spezifikation für DL\_URL\_REPLACE\_PREFIX einen anderen Präfixnamen enthält.
4. Der Wert für "path" (nach dem Anhängen von DL\_URL\_SUFFIX, sofern angegeben) ist der vollständige Pfadname der fernen Datei auf dem fernen Server. Relative Pfadnamen sind nicht zulässig. Das Standardpfadpräfix "http" für den Server wird nicht berücksichtigt.

**dl-begrenzer**

Beim ASCII-Format ohne universelle Zeilenbegrenzer (DEL) ein Zeichen, das über den Änderungswert `dl=del` angegeben wird oder im Befehl LOAD bzw. IMPORT standardmäßig verwendet wird. Beim ASCII-Format mit universellen Zeilenbegrenzern (ASC) sollte dies der Zeichenfolge `\;` (Backslash gefolgt von einem Semikolon) entsprechen. Leerraumzeichen (Leerzeichen, Tabulatorzeichen usw.) sind vor und nach dem Wert, der für diesen Parameter angegeben wird, zulässig.

**kommentar**

Der Kommentarteil eines DATALINK-Werts. Bei einer Angabe für das ASCII-Format ohne universelle Zeilenbegrenzer (DEL) muss der für *kommentar* angegebene Text durch den definierten Zeichenfolgebegrenzer eingeschlossen sein. In der Standardeinstellung sind dies doppelte Anführungszeichen (`"`). Dieser Zeichenfolgebegrenzer kann durch die Spezifikation MODIFIED BY *änderungswert-für-dateityp* des Befehls LOAD oder IMPORT überschrieben werden.

Falls kein Kommentar angegeben wird, wird standardmäßig eine Zeichenfolge mit einer Länge von Null verwendet.

Beispiele für DATALINK-Daten für das ASCII-Dateiformat ohne universelle Zeilenbegrenzer (DEL):

- `http://www.almaden.ibm.com:80/mrep/intro.mpeg; "Intro Movie"`  
Diese Angabe wird mit den folgenden Bestandteilen gespeichert:
  - `scheme = http`
  - `server = www.almaden.ibm.com`
  - `path = /mrep/intro.mpeg`
  - `comment = "Intro Movie"`
- `file://narang/u/narang; "InderPal's Home Page"`  
Diese Angabe wird mit den folgenden Bestandteilen gespeichert:
  - `scheme = file`
  - `server = narang`
  - `path = /u/narang`
  - `comment = "InderPal's Home Page"`

## Befehl LOAD

Beispiele für DATALINK-Daten für das ASCII-Dateiformat mit universellen Zeilenbegrenzern (ASC):

- `http://www.almaden.ibm.com:80/mrep/intro.mpeg\;Intro Movie`  
Diese Angabe wird mit den folgenden Bestandteilen gespeichert:
  - `schema = http`
  - `server = www.almaden.ibm.com`
  - `path = /mrep/intro.mpeg`
  - `comment = "Intro Movie"`
- `file://narang/u/narang\; InderPal's Home Page`  
Diese Angabe wird mit den folgenden Bestandteilen gespeichert:
  - `schema = file`
  - `server = narang`
  - `path = /u/narang`
  - `comment = "InderPal's Home Page"`

Beispiele für DATALINK-Daten, in denen als Spaltenspezifikation für LOAD oder IMPORT der Wert `DL_URL_REPLACE_PREFIX` ("`http://qso`") angenommen wird:

- `http://www.almaden.ibm.com/mrep/intro.mpeg`  
Diese Angabe wird mit den folgenden Bestandteilen gespeichert:
  - `schema = http`
  - `server = qso`
  - `path = /mrep/intro.mpeg`
  - `comment = NULL string`
- `/u/me/myfile.ps`  
Diese Angabe wird mit den folgenden Bestandteilen gespeichert:
  - `schema = http`
  - `server = qso`
  - `path = /u/me/myfile.ps`
  - `comment = NULL string`

### Zugehörige Konzepte:

- „LOAD - Übersicht“ auf Seite 82
- „Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von LOAD“ auf Seite 90

### Zugehörige Tasks:

- „Verwenden von LOAD“ auf Seite 91

### Zugehörige Referenzen:

- „QUIESCE TABLESPACES FOR TABLE Command“ in *Command Reference*
- „db2atld - Autoloader Command“ in *Command Reference*
- „CLP-Beispiele für LOAD“ auf Seite 197
- „LOAD-Konfigurationsoptionen für partitionierte Datenbanken“ auf Seite 213
- „Lade-API "db2Load"“ auf Seite 139
- „Änderungswerte für Dateityp für LOAD“ auf Seite 168

## LOAD QUERY

Überprüft den Status einer Ladeoperation während der Verarbeitung und gibt den Tabellenstatus zurück. Wenn keine Ladeoperation verarbeitet wird, wird nur der Tabellenstatus zurückgegeben. Sie benötigen eine Verbindung zu derselben Datenbank und eine separate Sitzung des Befehlszeilenprozessors, um diesen Befehl eingeben zu können. Er kann von lokalen und fernen Benutzern verwendet werden.

### Berechtigung:

Keine

### Erforderliche Verbindung:

Datenbank

### Befehlssyntax:

```

▶▶—LOAD QUERY—TABLE—tabellenname—┬──────────────────────────────────────────▶
                                     └─TO—lokale-nachrichtendatei—┘
▶──────────────────────────────────────────▶
┌─NOSUMMARY—┐ ┌─SHOWDELTA—┐
└─SUMMARYONLY—┘

```

### Befehlsparameter:

#### NOSUMMARY

Gibt an, dass keine LOAD-Ergebnistexte (gelesene Zeilen, übersprungene Zeilen, geladene Zeilen, zurückgewiesene Zeilen, gelöschte Zeilen, festgeschriebene Zeilen und Anzahl der Warnungen) gemeldet werden.

#### SHOWDELTA

Gibt an, dass nur neue Informationen (bezüglich LOAD-Ereignissen, die seit dem letzten Aufruf des Befehls LOAD QUERY eingetreten sind) gemeldet werden.

#### SUMMARYONLY

Gibt an, dass nur Ergebnistexte gemeldet werden.

#### TABLE *tabellenname*

Gibt den Namen der Tabelle an, in die die Daten geladen werden. Wenn ein Tabellenname ohne Qualifikationsmerkmal angegeben wird, wird die Tabelle mit dem aktuellen Schema (CURRENT SCHEMA) qualifiziert.

#### TO *lokale-nachrichtendatei*

Gibt das Ziel für Warnungen und Fehlernachrichten an, die während der Ladeoperation auftreten. Diese Datei darf nicht mit der für den Befehl LOAD angegebenen *nachrichtendatei* identisch sein. Wenn die Datei bereits vorhanden ist, werden die Informationen vom Dienstprogramm LOAD an sie angefügt.

### Beispiele:

Ein Benutzer, der eine große Datenmenge in die Tabelle STAFF lädt, sollte den Status der Ladeoperation überprüfen. Der Benutzer kann Folgendes angeben:

```

db2 connect to <datenbank>
db2 load query table staff to /u/mydir/staff.tempmsg

```

## Befehl LOAD QUERY

Die Ausgabedatei /u/mydir/staff.tempmsg könnte folgendermaßen aussehen:

SQL3501W Der oder die Tabellenbereich, die die Tabelle enthalten, werden nicht in den Status 'Sicherung anstehend' versetzt, da die aktualisierende Wiederherstellung für die Datenbank nicht aktiviert wurde.

SQL3109N Das Dienstprogramm beginnt mit dem Laden von Daten aus "/u/mydir/data/staffbig.del"

SQL3500W Die Phase "LOAD" wird gestartet (Zeit: "03-21-2002 11:31:16.597045").

SQL3519W Synchronisationspunkt am Beginn des Ladevorgangs. Zähler für Eingabesätze: "0".

SQL3520W Synchronisationspunkt für Ladevorgang erfolgreich.

SQL3519W Synchronisationspunkt am Beginn des Ladevorgangs. Zähler für Eingabesätze: "104416".

SQL3520W Synchronisationspunkt für Ladevorgang erfolgreich.

SQL3519W Synchronisationspunkt am Beginn des Ladevorgangs. Zähler für Eingabesätze: "205757".

SQL3520W Synchronisationspunkt für Ladevorgang erfolgreich.

SQL3519W Synchronisationspunkt am Beginn des Ladevorgangs. Zähler für Eingabesätze: "307098".

SQL3520W Synchronisationspunkt für Ladevorgang erfolgreich.

SQL3519W Synchronisationspunkt am Beginn des Ladevorgangs. Zähler für Eingabesätze: "408439".

SQL3520W Synchronisationspunkt für Ladevorgang erfolgreich.

SQL3532I Das Dienstprogramm LOAD befindet sich derzeit in Phase "LOAD".

Anzahl gelesener Zeilen	= 453376
Anzahl übersprungener Zeilen	= 0
Anzahl geladener Zeilen	= 453376
Anzahl zurückgewiesener Zeilen	= 0
Anzahl gelöschter Zeilen	= 0
Anzahl festgeschriebener Zeilen	= 408439
Anzahl Warnungen	= 0

Tabellenstatus:  
Laden läuft

### Anmerkungen zur Verwendung:

Neben Sperren verwendet das Dienstprogramm LOAD Statuswerte für Tabellen, um den Zugriff auf die Tabelle einzuschränken. Mit dem Befehl LOAD QUERY kann der Tabellenstatus ermittelt werden. Er kann auch für Tabellen verwendet werden, die gegenwärtig nicht geladen werden. Die von LOAD QUERY verwendeten Beschreibungen für den Tabellenstatus befinden sich unter Tabellensperren, Statuswerte für Tabellen und Statuswerte für Tabellenbereiche.

Der Bearbeitungsfortschritt einer Ladeoperation kann auch mit Hilfe des Befehls LIST UTILITIES überwacht werden.

**Zugehörige Konzepte:**

- „LOAD - Übersicht“ auf Seite 82
- „Tabellensperren, Statuswerte für Tabellen und Statuswerte für Tabellenbereiche“ auf Seite 186

**Zugehörige Referenzen:**

- „Befehl LOAD“ auf Seite 112
- „LIST UTILITIES Command“ in *Command Reference*

---

**Lade-API "db2Load"**

Lädt Daten in eine DB2-Tabelle. Die Daten können auf dem Server in Form einer Datei, eines Cursors, eines Bands oder einer benannten Pipe vorliegen. Daten, die sich auf einem fernen verbundenen Client befinden, können als vollständig qualifizierte Datei, als Cursor oder als benannte Pipe vorliegen. Das Dienstprogramm LOAD unterstützt das Laden von Daten auf Hierarchieebene nicht.

**Berechtigung:**

Eine der Folgenden:

- *sysadm*
- *dbadm*
- Berechtigung LOAD für die Datenbank und
  - Zugriffsrecht INSERT für die Tabelle, wenn das Dienstprogramm LOAD im Einfügemodus (INSERT), im Beendigungsmodus (TERMINATE) zur Beendigung einer vorherigen LOAD INSERT-Operation oder im Neustartmodus (RESTART) zum erneuten Starten einer vorherigen LOAD INSERT-Operation aufgerufen wird.
  - Zugriffsrecht INSERT und DELETE für die Tabelle, wenn das Dienstprogramm LOAD im Ersetzungsmodus (REPLACE), im Beendigungsmodus (TERMINATE) zur Beendigung einer vorherigen LOAD REPLACE-Operation oder im Neustartmodus (RESTART) zum erneuten Starten einer vorherigen LOAD REPLACE-Operation aufgerufen wird.
  - Zugriffsrecht INSERT für die Ausnahmetabelle, wenn eine solche Tabelle im Rahmen der Ladeoperation verwendet wird.

**Anmerkung:** Im Allgemeinen ist der Exemplareigner Eigner aller Ladeprozesse und aller DB2-Serverprozesse. Alle diese Prozesse verwenden die Identifikation des Exemplareigners, um auf die benötigten Dateien zuzugreifen. Unabhängig davon, wer den Befehl aufruft, muss daher der Exemplareigner einen Lesezugriff auf die Eingabedateien haben.

**Erforderliche Verbindung:**

Datenbank. Wenn das implizite Herstellen von Verbindungen aktiviert ist, wird eine Verbindung zur Standarddatenbank hergestellt.

Exemplar. Eine explizite Zuordnung ist nicht erforderlich. Wenn eine Verbindung zur Datenbank hergestellt wurde, wird versucht, eine implizite Zuordnung zum lokalen Exemplar vorzunehmen.

### API-Include-Datei:

*db2ApiDf.h*

### C-API-Syntax:

```
/* File: db2ApiDf.h */
/* API: Load */
/* ... */
SQL_API_RC SQL_API_FN
db2Load (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2LoadStruct
{
    struct sqlu_media_list *piSourceList;
    struct sqlu_media_list *piLobPathList;
    struct sqldcol *piDataDescriptor;
    struct sqlchar *piActionString;
    char *piFileType;
    struct sqlchar *piFileTypeMod;
    char *piLocalMsgFileName;
    char *piTempFilesPath;
    struct sqlu_media_list *piVendorSortWorkPaths;
    struct sqlu_media_list *piCopyTargetList;
    db2int32 *piNullIndicators;
    struct db2LoadIn *piLoadInfoIn;
    struct db2LoadOut *poLoadInfoOut;
    struct db2PartLoadIn *piPartLoadInfoIn;
    struct db2PartLoadOut *poPartLoadInfoOut;
    db2int16 iCallerAction;
} db2LoadStruct;

typedef SQL_STRUCTURE db2LoadIn
{
    db2UInt64 iRowcount;
    db2UInt64 iRestartcount;
    char *piUseTablespace;
    db2UInt32 iSavecount;
    db2UInt32 iDataBufferSize;
    db2UInt32 iSortBufferSize;
    db2UInt32 iWarningcount;
    db2UInt16 iHoldQuiesce;
    db2UInt16 iCpuParallelism;
    db2UInt16 iDiskParallelism;
    db2UInt16 iNonrecoverable;
    db2UInt16 iIndexingMode;
    db2UInt16 iAccessLevel;
    db2UInt16 iLockWithForce;
    db2UInt16 iCheckPending;
    char iRestartphase;
    char iStatsOpt;
} db2LoadIn;

typedef SQL_STRUCTURE db2LoadOut
{
    db2UInt64 oRowsRead;
    db2UInt64 oRowsSkipped;
    db2UInt64 oRowsLoaded;
    db2UInt64 oRowsRejected;
    db2UInt64 oRowsDeleted;
    db2UInt64 oRowsCommitted;
} db2LoadOut;

typedef SQL_STRUCTURE db2PartLoadIn
```

```

{
    char                *piHostname;
    char                *piFileTransferCmd;
    char                *piPartFileLocation;
    struct db2LoadNodeList *piOutputNodes;
    struct db2LoadNodeList *piPartitioningNodes;
    db2UInt16           *piMode;
    db2UInt16           *piMaxNumPartAgents;
    db2UInt16           *piIsolatePartErrs;
    db2UInt16           *piStatusInterval;
    struct db2LoadPortRange *piPortRange;
    db2UInt16           *piCheckTruncation;
    char                *piMapFileInput;
    char                *piMapFileOutput;
    db2UInt16           *piTrace;
    db2UInt16           *piNewline;
    char                *piDistfile;
    db2UInt16           *piOmitHeader;
    SQL_PDB_NODE_TYPE   *piRunStatDBPartNum;
} db2PartLoadIn;

typedef SQL_STRUCTURE db2LoadNodeList
{
    SQL_PDB_NODE_TYPE   *piNodeList;
    db2UInt16           iNumNodes;
} db2LoadNodeList;

typedef SQL_STRUCTURE db2LoadPortRange
{
    db2UInt16           iPortMin;
    db2UInt16           iPortMax;
} db2LoadPortRange;

typedef SQL_STRUCTURE db2PartLoadOut
{
    db2UInt64           oRowsRdPartAgents;
    db2UInt64           oRowsRejPartAgents;
    db2UInt64           oRowsPartitioned;
    struct db2LoadAgentInfo *poAgentInfoList;
    db2UInt32           iMaxAgentInfoEntries;
    db2UInt32           oNumAgentInfoEntries;
} db2PartLoadOut;

typedef SQL_STRUCTURE db2LoadAgentInfo
{
    db2int32            oSqlcode;
    db2UInt32           oTableState;
    SQL_PDB_NODE_TYPE   oNodeNum;
    db2UInt16           oAgentType;
} db2LoadAgentInfo;
/* ... */

```

**Generische API-Syntax:**

```

/* File: db2ApiDf.h */
/* API: Load */
/* ... */
SQL_API_RC SQL_API_FN
db2gLoad (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2gLoadStruct
{
    struct sqlu_media_list *piSourceList;
    struct sqlu_media_list *piLobPathList;
}

```

## Lade-API "db2Load"

```
struct sqlDcol *piDataDescriptor;
struct sqlchar *piActionString;
char *piFileType;
struct sqlchar *piFileTypeMod;
char *piLocalMsgFileName;
char *piTempFilesPath;
struct sqlu_media_list *piVendorSortWorkPaths;
struct sqlu_media_list *piCopyTargetList;
db2int32 *piNullIndicators;
struct db2gLoadIn *piLoadInfoIn;
struct db2LoadOut *poLoadInfoOut;
struct db2gPartLoadIn *piPartLoadInfoIn;
struct db2PartLoadOut *poPartLoadInfoOut;
db2int16 iCallerAction;
db2Uint16 iFileTypeLen;
db2Uint16 iLocalMsgFileLen;
db2Uint16 iTempFilesPathLen;
} db2gLoadStruct;

typedef SQL_STRUCTURE db2gLoadIn
{
    db2Uint64                iRowcount;
    db2Uint64                iRestartcount;
    char                    *piUseTablespace;
    db2Uint32                iSavecount;
    db2Uint32                iDataBufferSize;
    db2Uint32                iSortBufferSize;
    db2Uint32                iWarningcount;
    db2Uint16                iHoldQuiesce;
    db2Uint16                iCpuParallelism;
    db2Uint16                iDiskParallelism;
    db2Uint16                iNonrecoverable;
    db2Uint16                iIndexingMode;
    db2Uint16                iAccessLevel;
    db2Uint16                iLockWithForce;
    db2Uint16                iCheckPending;
    char                    iRestartphase;
    char                    iStatsOpt;
    db2Uint16                iUseTablespaceLen;
} db2gLoadIn;

typedef SQL_STRUCTURE db2LoadOut
{
    db2Uint64                oRowsRead;
    db2Uint64                oRowsSkipped;
    db2Uint64                oRowsLoaded;
    db2Uint64                oRowsRejected;
    db2Uint64                oRowsDeleted;
    db2Uint64                oRowsCommitted;
} db2LoadOut;

typedef SQL_STRUCTURE db2gPartLoadIn
{
    char                    *piHostname;
    char                    *piFileTransferCmd;
    char                    *piPartFileLocation;
    struct db2LoadNodeList *piOutputNodes;
    struct db2LoadNodeList *piPartitioningNodes;
    db2Uint16                *piMode;
    db2Uint16                *piMaxNumPartAgents;
    db2Uint16                *piIsolatePartErrs;
    db2Uint16                *piStatusInterval;
    struct db2LoadPortRange *piPortRange;
    db2Uint16                *piCheckTruncation;
    char                    *piMapFileInput;
    char                    *piMapFileOutput;
    db2Uint16                *piTrace;
```

```

    db2UInt16                *piNewline;
    char                    *piDistfile;
    db2UInt16                *piOmitHeader;
    SQL_PDB_NODE_TYPE       *piRunStatDBPartNum;
    db2UInt16                iHostnameLen;
    db2UInt16                iFileTransferLen;
    db2UInt16                iPartFileLocLen;
    db2UInt16                iMapFileInputLen;
    db2UInt16                iMapFileOutputLen;
    db2UInt16                iDistfileLen;
} db2gPartLoadIn;

typedef SQL_STRUCTURE db2LoadNodeList
{
    SQL_PDB_NODE_TYPE       *piNodeList;
    db2UInt16                iNumNodes;
} db2LoadNodeList;

typedef SQL_STRUCTURE db2LoadPortRange
{
    db2UInt16                iPortMin;
    db2UInt16                iPortMax;
} db2LoadPortRange;

typedef SQL_STRUCTURE db2PartLoadOut
{
    db2UInt64                oRowsRdPartAgents;
    db2UInt64                oRowsRejPartAgents;
    db2UInt64                oRowsPartitioned;
    struct db2LoadAgentInfo  *poAgentInfoList;
    db2UInt32                iMaxAgentInfoEntries;
    db2UInt32                oNumAgentInfoEntries;
} db2PartLoadOut;

typedef SQL_STRUCTURE db2LoadAgentInfo
{
    db2int32                 oSqlcode;
    db2UInt32                oTableState;
    SQL_PDB_NODE_TYPE       oNodeNum;
    db2UInt16                oAgentType;
} db2LoadAgentInfo;
/* ... */

```

**API-Parameter:****versionNumber**

Eingabe. Gibt die Version und den Releasestand der Struktur an, die als zweiter Parameter, *pParmStruct*, übergeben wird.

**pParmStruct**

Eingabe. Ein Zeiger auf die *db2LoadStruct*-Struktur.

**pSqlca**

Ausgabe. Ein Zeiger auf die *sqlca*-Struktur.

**piSourceList**

Eingabe. Ein Zeiger auf eine *sqlu\_media\_list*-Struktur, die verwendet wird, um eine Liste mit Quellendateien, Einheiten, Lieferanten, Pipes oder SQL-Anweisungen bereitzustellen.

Die in dieser Struktur bereitgestellten Daten hängen vom Wert des Feldes *media\_type* ab. Gültige Werte (definiert in *sqlutil*) sind:

**SQLU\_SQL\_STMT**

Wenn das Feld *media\_type* auf diesen Wert gesetzt ist, stellt das aufrufende Programm eine SQL-Abfrage über das Feld *pStatement* des

Zielfelds bereit. Das Feld *pStatement* hat den Typ *sqlu\_statement\_entry*. Das Feld "sessions" muss auf den Wert 1 gesetzt sein, da das Dienstprogramm LOAD pro Ladevorgang nur eine SQL-Abfrage akzeptiert.

### SQLU\_SERVER\_LOCATION

Wenn das Feld *media\_type* auf diesen Wert gesetzt ist, stellt das aufrufende Programm Daten über die *sqlu\_location\_entry*-Strukturen bereit. Das Feld *sessions* gibt die Anzahl der bereitgestellten *sqlu\_location\_entry*-Strukturen an. Es wird für Dateien, Einheiten und benannte Pipes verwendet.

### SQLU\_CLIENT\_LOCATION

Wenn das Feld *media\_type* auf diesen Wert gesetzt ist, stellt das aufrufende Programm Daten über die *sqlu\_location\_entry*-Strukturen bereit. Das Feld *sessions* gibt die Anzahl der bereitgestellten *sqlu\_location\_entry*-Strukturen an. Es wird für vollständig qualifizierte Dateien und benannte Pipes verwendet. Es ist zu beachten, dass dieses Feld *media\_type* nur dann gültig ist, wenn die API über einen Client mit Fernverbindung aufgerufen wird.

### SQLU\_TSM\_MEDIA

Wenn das Feld *media\_type* auf diesen Wert gesetzt ist, wird die *sqlu\_vendor*-Struktur verwendet, wobei *filename* die eindeutige Kennung für die zu ladenden Daten ist. Es darf nur einen *sqlu\_vendor*-Eintrag geben, unabhängig vom Wert von *sessions*. Das Feld *sessions* gibt die Anzahl der zu startenden TSM-Sitzungen an. Das Dienstprogramm LOAD startet die Sitzungen mit unterschiedlichen Folgenummern, jedoch mit denselben Daten in dem einen *sqlu\_vendor*-Eintrag.

### SQLU\_OTHER\_MEDIA

Wenn das Feld *media\_type* auf diesen Wert gesetzt ist, wird die *sqlu\_vendor*-Struktur verwendet, wobei *shr\_lib* der Name der gemeinsam benutzten Bibliothek und *filename* die eindeutige Kennung für die zu ladenden Daten ist. Es darf nur einen *sqlu\_vendor*-Eintrag geben, unabhängig vom Wert von *sessions*. Das Feld *sessions* gibt die Anzahl der übrigen einzuleitenden Lieferantensitzungen an. Das Dienstprogramm LOAD startet die Sitzungen mit unterschiedlichen Folgenummern, jedoch mit denselben Daten in dem einen *sqlu\_vendor*-Eintrag.

### piLobPathList

Eingabe. Ein Zeiger auf eine *sqlu\_media\_list*-Struktur. Für die Dateitypen IXF, ASC und DEL eine Liste mit vollständig qualifizierten Pfaden oder Einheiten, die die Speicherposition der einzelnen zu ladenden LOB-Dateien angeben. Die Dateinamen befinden sich in den IXF-, ASC- oder DEL-Dateien und werden an die bereitgestellten Pfade angefügt.

Die in dieser Struktur bereitgestellten Daten hängen vom Wert des Feldes *media\_type* ab. Gültige Werte (definiert in `sqlutil`) sind:

### SQLU\_LOCAL\_MEDIA

Bei diesem Wert stellt das aufrufende Programm Daten über die *sqlu\_media\_entry*-Strukturen bereit. Das Feld *sessions* gibt die Anzahl der bereitgestellten *sqlu\_media\_entry*-Strukturen an.

### SQLU\_TSM\_MEDIA

Bei diesem Wert wird die *sqlu\_vendor*-Struktur verwendet, wobei *filename* die eindeutige Kennung für die zu ladenden Daten ist. Es

darf nur einen *sqlu\_vendor*-Eintrag geben, unabhängig vom Wert von *sessions*. Das Feld *sessions* gibt die Anzahl der zu startenden TSM-Sitzungen an. Das Dienstprogramm LOAD startet die Sitzungen mit unterschiedlichen Folgenummern, jedoch mit denselben Daten in dem einen *sqlu\_vendor*-Eintrag.

### SQLU\_OTHER\_MEDIA

Bei diesem Wert wird die *sqlu\_vendor*-Struktur verwendet, wobei *shr\_lib* der Name der gemeinsam benutzten Bibliothek und *filename* die eindeutige Kennung der zu ladenden Daten ist. Es darf nur einen *sqlu\_vendor*-Eintrag geben, unabhängig vom Wert von *sessions*. Das Feld *sessions* gibt die Anzahl der übrigen einzuleitenden Lieferantensitzungen an. Das Dienstprogramm LOAD startet die Sitzungen mit unterschiedlichen Folgenummern, jedoch mit denselben Daten in dem einen *sqlu\_vendor*-Eintrag.

### piDataDescriptor

Eingabe. Zeiger auf eine *sqldcol*-Struktur, die Informationen zu den Spalten enthält, die zum Laden aus der externen Datei ausgewählt werden.

Wenn der Parameter *pFileType* auf den Wert SQL\_ASC gesetzt ist, muss das Feld *dcolmeth* dieser Struktur entweder auf SQL\_METH\_L oder auf SQL\_METH\_D gesetzt sein, und gibt einen Dateinamen mit dem Änderungswert *pFileTypeMod* für POSITIONSFILe an, der Start- und Endpaare sowie Nullanzeigerpositionen enthält. Der Benutzer gibt die Anfangs- und Endposition für jede zu ladende Spalte an.

Wenn der Dateityp SQL\_DEL ist, kann *dcolmeth* entweder SQL\_METH\_P oder SQL\_METH\_D sein. Bei SQL\_METH\_P muss der Benutzer die Quellenspaltenposition angeben. Bei SQL\_METH\_D wird die erste Spalte in der Datei in die erste Spalte der Tabelle geladen usw.

Wenn der Dateityp SQL\_IXF ist, kann *dcolmeth* entweder SQL\_METH\_P, SQL\_METH\_D oder SQL\_METH\_N sein. Hier gelten die Regeln für DEL-Dateien, außer dass SQL\_METH\_N angibt, dass Dateispaltennamen in der *sqldcol*-Struktur bereitgestellt werden sollen.

### piActionString

Eingabe. Ein Zeiger auf eine *sqlchar*-Struktur, gefolgt von einer Matrix mit Zeichen, die eine Maßnahme angeben, die sich auf die Tabelle auswirkt.

Die Zeichenmatrix besitzt folgendes Format:

```
"INSERT|REPLACE|RESTART|TERMINATE
INTO tname [(spaltenliste)]
[ATALINK SPECIFICATION datalink-spez]
[FOR EXCEPTION a_tname]"
```

#### INSERT

Fügt die geladenen Daten zur Tabelle hinzu, ohne die vorhandenen Tabellendaten zu ändern.

#### REPLACE

Löscht alle vorhandenen Daten aus der Tabelle und fügt die importierten Daten ein. Die Tabellendefinition und die Indexdefinitionen werden nicht verändert.

#### RESTART

Startet eine zuvor unterbrochene Ladeoperation erneut. Die Ladeoperation wird automatisch vom letzten Konsistenzzustand in der LOAD-, BUILD- oder DELETE-Phase ausgehend fortgesetzt.

### TERMINATE

Beendet eine zuvor unterbrochene Ladeoperation und macht die Operation bis zu dem Zeitpunkt, zu dem sie gestartet wurde, rückgängig, auch wenn Konsistenzzustände übergangen wurden. Die Statuswerte von betroffenen Tabellenbereichen werden in "Normal" zurückgeändert, und alle Tabellenobjekte werden konsistent gemacht (Indexobjekte können als ungültig gekennzeichnet werden, so dass beim nächsten Zugriff automatisch eine Indexwiederherstellung erfolgt). Wenn sich die Tabellenbereiche, in denen die Tabelle enthalten ist, nicht im Status "Laden anstehend" befinden, wirkt sich diese Option nicht auf den Status der Tabellenbereiche aus.

Mit der Option LOAD TERMINATE wird der Status "Sicherung anstehend" für Tabellenbereiche nicht aufgehoben.

*tbname* Der Name der Tabelle, in die die Daten geladen werden sollen. Diese Tabelle darf keine Systemtabelle oder deklarierte temporäre Tabelle sein. Es kann ein Aliasname, der vollständig qualifizierte Name oder ein Name ohne Qualifikationsmerkmal angegeben werden. Ein qualifizierter Tabellename hat das Format *schema.tabellenname*. Wenn ein Tabellename ohne Qualifikationsmerkmal angegeben wird, wird die Tabelle mit dem aktuellen Schema (CURRENT SCHEMA) qualifiziert.

(*spaltenliste*)

Eine Liste der Tabellenspaltennamen, in die die Daten eingefügt werden sollen. Die Spaltennamen müssen durch Kommas getrennt werden. Wenn ein Name Leerzeichen oder Kleinbuchstaben enthält, muss er in Anführungszeichen gesetzt werden.

### DATALINK SPECIFICATION *datalink-spez*

Gibt Parameter an, die sich auf DB2 Data Links beziehen. Diese Parameter können mit der gleichen Syntax wie beim Befehl LOAD angegeben werden.

### FOR EXCEPTION *a\_tbname*

Gibt die Ausnahmetabelle an, in die fehlerhafte Zeilen kopiert werden. Jede Zeile, die gegen einen eindeutigen Index oder einen Primärschlüsselindex verstößt, wird kopiert. DATALINK-Ausnahmen werden ebenfalls in der Ausnahmetabelle erfasst.

### piFileType

Eingabe. Eine Zeichenfolge, die das Format der Quelle für die Eingabedaten angibt. Unterstützte externe Formate (definiert in `sqlutil`) sind:

#### SQL\_ASC

ASCII-Format mit universellen Zeilenbegrenzern.

#### SQL\_DEL

ASCII-Format ohne universelle Zeilenbegrenzer für den Datenaustausch mit dBase, BASIC und den IBM Personal Decision Series-Programmen und vielen anderen Datenbank- und Dateimanagern.

#### SQL\_IXF

PC-Version des Integrated Exchange Format. Dies ist das bevorzugte Format für den Export von Daten aus einer Tabelle, so dass sie später in dieselbe Tabelle oder eine andere Datenbankmanagertabelle geladen werden können.

**SQL\_CURSOR**

Eine SQL-Abfrage. Die *sqlu\_media\_list*-Struktur, die über den Parameter *piSourceList* übergeben wird, hat den Typ `SQLU_SQL_STMT` und verweist auf eine tatsächliche SQL-Abfrage und nicht auf einen Cursor, der für eine SQL-Abfrage deklariert ist.

**piFileTypeMod**

Eingabe. Ein Zeiger auf die *sqlchar*-Struktur, gefolgt von einer Matrix mit Zeichen, die eine oder mehrere Verarbeitungsoptionen angeben. Wenn dieser Zeiger NULL ist oder die Struktur, auf die gezeigt wird, null Zeichen enthält, wird diese Aktion als Auswahl einer Standardspezifikation gewertet.

Nicht alle Optionen können mit allen unterstützten Dateitypen verwendet werden. Siehe File type modifiers for load.

**piLocalMsgFileName**

Eingabe. Eine Zeichenfolge, die den Namen einer lokalen Datei enthält, in die Ausgabenachrichten geschrieben werden sollen.

**piTempFilesPath**

Eingabe. Eine Zeichenfolge, die den Pfadnamen enthält, der auf dem Server für temporäre Dateien verwendet werden soll. Temporäre Dateien werden erstellt, um Nachrichten, Konsistenzzustände und Informationen zu DELETE-Phasen zu speichern.

**piVendorSortWorkPaths**

Eingabe. Ein Zeiger auf die *sqlu\_media\_list*-Struktur, die die Sortierarbeitsverzeichnisse des Lieferanten angibt.

**piCopyTargetList**

Eingabe. Ein Zeiger auf eine *sqlu\_media\_list*-Struktur, die (wenn ein Kopierabbild erstellt werden soll) verwendet wird, um eine Liste mit Zielpfaden, Einheiten oder eine gemeinsam benutzte Bibliothek bereitzustellen, in die das Kopierabbild geschrieben werden soll.

Die in dieser Struktur bereitgestellten Werte hängen vom Wert des Feldes *media\_type* ab. Gültige Werte für dieses Feld (definiert in `sqlutil`) sind:

**SQLU\_LOCAL\_MEDIA**

Wenn die Kopie auf einen lokalen Datenträger geschrieben werden soll, muss *media\_type* auf diesen Wert gesetzt werden, und in den *sqlu\_media\_entry*-Strukturen müssen Informationen zu den Zielen bereitgestellt werden. Das Feld *sessions* gibt die Anzahl der bereitgestellten *sqlu\_media\_entry*-Strukturen an.

**SQLU\_TSM\_MEDIA**

Verwenden Sie diesen Wert, wenn die Kopie in Tivoli Storage Manager (TSM) geschrieben werden soll. Es sind keine weitere Informationen erforderlich.

**SQLU\_OTHER\_MEDIA**

Wenn ein Produkt eines Lieferanten verwendet werden soll, müssen Sie diesen Wert verwenden und weitere Informationen über eine *sqlu\_vendor*-Struktur bereitstellen. Setzen Sie das Feld *shr\_lib* dieser Struktur auf den Namen der gemeinsam benutzten Bibliothek des Produkts des Lieferanten. Geben Sie nur einen *sqlu\_vendor*-Eintrag an, unabhängig vom Wert von *sessions*. Das Feld *sessions* gibt die Anzahl der bereitgestellten *sqlu\_media\_entry*-Strukturen an. Das Dienstprogramm LOAD startet die Sitzungen mit unter-

schiedlichen Folgenummern, jedoch mit denselben Daten, die in dem einen *sqlu\_vendor*-Eintrag bereitgestellt werden.

### **piNullIndicators**

Eingabe. Nur für ASC-Dateien. Eine Matrix von ganzen Zahlen, die angeben, ob die Spaltendaten Nullwerte annehmen dürfen oder nicht. Es besteht eine 1-zu-1-Entsprechung zwischen den Elementen dieser Matrix und den Spalten, die aus der Datendatei geladen werden. Das heißt, die Anzahl der Elemente muss mit dem Feld *dcolnum* des Parameters *pDataDescriptor* übereinstimmen. Jedes Element der Matrix enthält eine Zahl, die eine Speicherposition in der Datendatei angibt, die als Nullanzeigerfeld verwendet werden soll, oder eine Null, um anzugeben, dass die Tabellenspalte keine Nullwerte enthalten darf. Wenn das Element nicht Null ist, muss die angegebene Speicherposition in der Datendatei entweder Y oder N enthalten. Y gibt an, dass die Tabellenspaltendaten NULL sind, N gibt an, dass die Tabellenspaltendaten nicht NULL sind.

### **piLoadInfoIn**

Eingabe. Ein Zeiger auf die *db2LoadIn*-Struktur.

### **poLoadInfoOut**

Eingabe. Ein Zeiger auf die *db2LoadOut*-Struktur.

### **piPartLoadInfoIn**

Eingabe. Ein Zeiger auf die *db2PartLoadIn*-Struktur.

### **poPartLoadInfoOut**

Ausgabe. Ein Zeiger auf die *db2PartLoadOut*-Struktur.

### **iCallerAction**

Eingabe. Eine vom aufrufenden Programm angeforderte Aktion. Gültige Werte (definiert in *sqlutil*) sind:

#### **SQLU\_INITIAL**

Erstaufruf. Dieser Wert (oder *SQLU\_NOINTERRUPT*) muss beim ersten Aufruf der API verwendet werden.

#### **SQLU\_NOINTERRUPT**

Erstaufruf. Verarbeitung nicht aussetzen. Dieser Wert (oder *SQLU\_INITIAL*) muss beim ersten Aufruf der API verwendet werden.

Wenn der Erstaufruf oder ein nachfolgender Aufruf zurückkehrt und erfordert, dass die aufrufende Anwendung eine Aktion ausführt, bevor die angeforderte Ladeoperation ausgeführt wird, muss die Aktion des aufrufenden Programms auf eine der folgenden Optionen gesetzt werden:

#### **SQLU\_CONTINUE**

Verarbeitung fortsetzen. Dieser Wert kann nur bei nachfolgenden Aufrufen der API verwendet werden, nachdem der Erstaufruf zurückgegeben wurde und das Dienstprogramm eine Benutzereingabe anfordert (z. B. um auf eine Bandendebedingung zu reagieren). Er gibt an, dass die vom Dienstprogramm angeforderte Benutzeraktion vollständig beendet wurde und das Dienstprogramm die Verarbeitung der Erstanforderung fortsetzen kann.

#### **SQLU\_TERMINATE**

Verarbeitung beenden. Bewirkt, dass das Dienstprogramm *LOAD* vorzeitig beendet wird, wobei die zu ladenden Tabellenbereiche im Status *LOAD\_PENDING* verbleiben. Diese Option sollte angegeben werden, wenn keine weitere Verarbeitung der Daten erfolgen soll.

**SQLU\_ABORT**

Verarbeitung beenden. Bewirkt, dass das Dienstprogramm LOAD vorzeitig beendet wird, wobei die zu ladenden Tabellenbereiche im Status LOAD\_PENDING verbleiben. Diese Option sollte angegeben werden, wenn keine weitere Verarbeitung der Daten erfolgen soll.

**SQLU\_RESTART**

Verarbeitung erneut starten.

**SQLU\_DEVICE\_TERMINATE**

Eine einzelne Einheit beenden. Diese Option sollte angegeben werden, wenn das Dienstprogramm das Lesen von Daten von der Einheit stoppen, jedoch eine weitere Verarbeitung der Daten erfolgen soll.

**iFileTypeLen**

Eingabe. Gibt die Länge von *iFileType* in Byte an.

**iLocalMsgFileLen**

Eingabe. Gibt die Länge von *iLocalMsgFileName* in Byte an.

**iTempFilesPathLen**

Eingabe. Gibt die Länge von *iTempFilesPath* in Byte an.

**iRowcount**

Eingabe. Die Anzahl der physischen Sätze, die geladen werden sollen. Ermöglicht es einem Benutzer, nur die ersten mit *rowcnt* angegebenen Zeilen in einer Datei zu laden.

**iRestartcount**

Eingabe. Zur zukünftigen Verwendung reserviert.

**piUseTablespace**

Eingabe. Falls die Indizes erneut erstellt werden, wird eine Schattenkopie des Indexes in dem mit *iUseTablespaceName* angegebenen Tabellenbereich erstellt und am Ende des Ladevorgangs in den ursprünglichen Tabellenbereich kopiert. Diese Option kann nur bei systemtemporären Tabellen verwendet werden. Ist sie nicht angegeben, wird der Schattenindex in demselben Tabellenbereich wie das Indexobjekt erstellt.

Falls die Schattenkopie in demselben Tabellenbereich wie das Indexobjekt erstellt wird, wird das Schattenindexobjekt sofort über das alte Indexobjekt kopiert. Befindet sich die Schattenkopie in einem anderen Tabellenbereich als das Indexobjekt, wird eine physische Kopie ausgeführt. Dies kann sich erheblich auf die Ein-/Ausgabe und die benötigte Zeit auswirken. Das Kopieren findet statt, während die Tabelle am Ende eines Ladevorgangs offline ist.

Dieses Feld wird ignoriert, wenn *iAccessLevel* den Wert SQLU\_ALLOW\_NO\_ACCESS hat.

Diese Option wird ignoriert, wenn der Benutzer INDEXING MODE REBUILD oder INDEXING MODE AUTOSELECT nicht angibt. Sie wird außerdem ignoriert, wenn INDEXING MODE AUTOSELECT ausgewählt wird und der Index beim Laden inkrementell aktualisiert wird.

**iSavecount**

Die Anzahl der Datensätze, die geladen werden sollen, bevor ein Konsistenzzustand definiert wird. Dieser Wert wird in einen Seitenzähler konvertiert und auf Intervalle in der Größe des Speicherbereichs gerundet. Da bei jedem Konsistenzzustand eine Nachricht abgesetzt wird, sollte diese Option ausgewählt werden, wenn die Ladeoperation mit *db2LoadQuery*

(API zum Abfragen des Status' eines Ladevorgangs) überwacht wird. Wenn der Wert von *savecnt* nicht groß genug ist, wirkt sich die Synchronisation der Aktivitäten, die bei jedem Konsistenzzustand ausgeführt werden, nachteilig auf die Leistung aus.

Der Standardwert ist 0, d. h., dass Konsistenzzustände nur definiert werden, sofern dies notwendig ist.

### **iDataBufferSize**

Die Anzahl der 4 KB-Seiten (unabhängig vom Grad der Parallelität), die als Pufferbereich zur Übertragung von Daten im Dienstprogramm verwendet werden sollen. Wenn der angegebene Wert kleiner als der algorithmische Mindestwert ist, wird der erforderliche Mindestwert verwendet, und es wird keine Warnung zurückgegeben.

Dieser Speicher wird direkt aus dem Zwischenspeicher der Dienstprogramme zugeordnet, dessen Größe mit dem Datenbankkonfigurationsparameter *util\_heap\_sz* geändert werden kann.

Wenn kein Wert angegeben wird, wird vom Dienstprogramm zur Laufzeit ein sinnvoller Standardwert berechnet. Der Standardwert basiert auf einem Prozentsatz des freien Speicherbereichs, der bei der Exemplarerstellung des Ladeprogramms im Zwischenspeicher der Dienstprogramme verfügbar ist, sowie auf einigen Merkmalen der Tabelle.

### **iSortBufferSize**

Eingabe. Mit dieser Option wird ein Wert angegeben, der den Datenbankkonfigurationsparameter SORTHEAP während einer Ladeoperation überschreibt. Sie ist nur dann von Bedeutung, wenn Tabellen mit Indizes geladen werden und für den Parameter *iIndexingMode* ein anderer Wert als *SQLU\_INX\_DEFERRED* angegeben ist. Der angegebene Wert kann den Wert des Parameters SORTHEAP nicht überschreiten. Dieser Parameter ist nützlich, um den von LOAD verwendeten Sortierspeicher zu verkleinern, ohne den Wert für SORTHEAP ändern zu müssen, der sich auch auf die allgemeine Abfrageverarbeitung auswirkt.

### **iWarningcount**

Eingabe. Stoppt die Ladeoperation nach der mit *warningcnt* angegebenen Anzahl von Warnungen. Definieren Sie diesen Parameter, wenn keine Warnungen erwartet werden, jedoch eine Prüfung darauf, ob die richtige Datei und Tabelle verwendet wird, gewünscht wird. Wird die Ladedatei oder die Zieltabelle falsch angegeben, generiert das Dienstprogramm LOAD eine Warnung für jede Zeile, die es versucht zu laden, sodass der Ladevorgang fehlschlägt. Wenn *warningcnt* den Wert 0 hat oder diese Option nicht angegeben wird, wird die Ladeoperation unabhängig von der Anzahl der abgesetzten Warnungen fortgesetzt.

Wenn die Ladeoperation gestoppt wird, da die Schwelle für Warnungen überschritten wurde, kann eine weitere Ladeoperation im Neustartmodus (RESTART) gestartet werden. Die Ladeoperation wird automatisch vom letzten Konsistenzzustand ausgehend fortgesetzt. Alternativ kann eine weitere Ladeoperation im Ersetzungsmodus (REPLACE) eingeleitet werden, die am Anfang der Eingabedatei beginnt.

### **iHoldQuiesce**

Eingabe. Eine Markierung, deren Wert auf TRUE gesetzt wird, wenn das Dienstprogramm die Tabelle nach dem Laden im Status "Im Wartemodus: EXCLUSIVE" belassen soll, bzw. auf FALSE, wenn das nicht erwünscht ist.

**iCpuParallelism**

Eingabe. Die Anzahl der Prozesse oder Threads, die vom Dienstprogramm LOAD gestartet werden, um Datensätze beim Erstellen von Tabellenobjekten syntaktisch zu analysieren, zu konvertieren und zu formatieren. Dieser Parameter soll die Nutzung der partitionsinternen Parallelität ermöglichen. Er ist besonders nützlich beim Laden von vorsortierten Daten, weil die Satzreihenfolge der Quelldaten beibehalten wird. Wenn der Wert dieses Parameters Null ist, verwendet das Dienstprogramm LOAD zur Laufzeit einen sinnvollen Standardwert. Anmerkung: Wenn dieser Parameter bei Tabellen verwendet wird, die LOB- oder LONG VARCHAR-Felder enthalten, wird sein Wert Eins, unabhängig von der Anzahl der System-CPU's oder dem vom Benutzer angegebenen Wert.

**iDiskParallelism**

Eingabe. Die Anzahl der Prozesse oder Threads, die vom Dienstprogramm LOAD gestartet werden, um Daten in die Tabellenbereichsbehälter zu schreiben. Wenn kein Wert angegeben wird, wählt das Dienstprogramm einen sinnvollen Standardwert auf Basis der Anzahl der Tabellenbereichsbehälter und der Merkmale der Tabelle aus.

**iNonrecoverable**

Eingabe. Der Wert `SQLU_NON_RECOVERABLE_LOAD` gibt an, dass eine Ladetransaktion als nicht wiederherstellbar gekennzeichnet werden soll und dass es nicht möglich sein wird, sie durch eine nachfolgende aktualisierende Wiederherstellung wiederherzustellen. Das Dienstprogramm zur aktualisierenden Wiederherstellung überspringt die Transaktion und kennzeichnet die Tabelle, in die die Daten geladen wurden, als "ungültig". Vom Dienstprogramm werden auch alle nachfolgenden Transaktionen für diese Tabelle ignoriert. Nach Beendigung der aktualisierenden Wiederherstellung kann eine solche Tabelle nur gelöscht werden. Mit dieser Option werden Tabellenbereiche nach der Ladeoperation nicht in den Status "Sicherung anstehend" versetzt, und während der Ladeoperation muss keine Kopie der geladenen Daten erstellt werden. Der Wert `SQLU_RECOVERABLE_LOAD` gibt an, dass die Ladetransaktion als wiederherstellbar gekennzeichnet werden soll.

**iIndexingMode**

Eingabe. Gibt den Indexierungsmodus an. Gültige Werte (definiert in `sqlutil`) sind:

**SQLU\_INX\_AUTOSELECT**

LOAD wählt zwischen den Indexierungsmodi REBUILD und INCREMENTAL.

**SQLU\_INX\_REBUILD**

Tabellenindizes werden erneut erstellt.

**SQLU\_INX\_INCREMENTAL**

Vorhandene Indizes werden erweitert.

**SQLU\_INX\_DEFERRED**

Tabellenindizes sollen nicht aktualisiert werden.

**iAccessLevel**

Eingabe. Gibt die Zugriffsebene an. Gültige Werte:

**SQLU\_ALLOW\_NO\_ACCESS**

Gibt an, dass LOAD die Tabelle exklusiv sperrt.

**SQLU\_ALLOW\_READ\_ACCESS**

Gibt an, dass die ursprünglichen Daten in der Tabelle (die nicht

zum Deltateil gehören) weiterhin für Eingabeprogramme erkennbar sein sollen, während der Ladevorgang ausgeführt wird. Diese Option ist nur beim Laden mit Anhängen gültig, wie z. B. bei einer LOAD INSERT-Operation. Bei einer LOAD REPLACE-Operation wird sie ignoriert.

### **iLockWithForce**

Eingabe. Eine Boolesche Markierung. Wenn sie auf TRUE gesetzt ist, werden andere Anwendungen bei Bedarf beendet, um sicherzustellen, dass Tabellensperren sofort verfügbar sind. Für diese Option wird dieselbe Berechtigung wie für den Befehl FORCE APPLICATIONS benötigt (SYSADM oder SYSCTRL).

Ladevorgänge mit `SQLU_ALLOW_NO_ACCESS` können die Beendigung von Konflikte verursachenden Anwendungen am Beginn der Ladeoperation erzwingen. Am Anfang des Ladevorgangs kann das Dienstprogramm die Beendigung von Anwendungen erzwingen, die versuchen, die Tabelle entweder abzufragen oder zu ändern.

Ladevorgänge mit `SQLU_ALLOW_READ_ACCESS` können die Beendigung von Konflikte verursachenden Anwendungen am Beginn oder am Ende der Ladeoperation erzwingen. Am Anfang des Ladevorgangs kann das Dienstprogramm LOAD die Beendigung von Anwendungen erzwingen, die versuchen, die Tabelle zu ändern. Am Ende des Ladevorgangs kann das Dienstprogramm LOAD die Beendigung von Anwendungen erzwingen, die versuchen, die Tabelle entweder abzufragen oder zu ändern.

### **iCheckPending**

Eingabe. Gibt an, dass die Tabelle in den Status "Überprüfung anstehend" (CHECK PENDING) versetzt werden soll. Falls `SQLU_CHECK_PENDING_CASCADE_IMMEDIATE` angegeben ist, wird der Status "Überprüfung anstehend" sofort an alle abhängigen und untergeordneten Tabellen weitergegeben. Ist `SQLU_CHECK_PENDING_CASCADE_DEFERRED` angegeben, wird die Weitergabe des Statuswerts "Überprüfung anstehend" an abhängige Tabellen verzögert, bis die Zieltabelle auf ungültige Integritätsbedingungen hin überprüft wurde. `SQLU_CHECK_PENDING_CASCADE_DEFERRED` ist der Standardwert, wenn die Option nicht angegeben wird.

### **iRestartphase**

Eingabe. Reserviert. Gültiger Wert ist ein einzelnes Leerzeichen ' '.

### **iStatsOpt**

Eingabe. Granularität der zu erfassenden Statistikdaten. Gültige Werte:

#### **SQLU\_STATS\_NONE**

Es werden keine Statistikdaten zusammengestellt.

#### **SQLU\_STATS\_USE\_PROFILE**

Statistiken werden auf Grundlage des für die aktuelle Tabelle definierten Profils erfasst. Dieses Profil muss mit Hilfe des Befehls `RUNSTATS` erstellt werden. Ist für die aktuelle Tabelle kein Profil vorhanden, wird eine Warnung zurückgegeben und es werden keine Statistiken erfasst.

### **iUseTablespaceLen**

Eingabe. Gibt die Länge von *piUseTablespace* in Byte an.

### **oRowsRead**

Ausgabe. Anzahl der während der Ladeoperation gelesenen Datensätze

**oRowsSkipped**

Ausgabe. Anzahl der Datensätze, die vor Beginn der Ladeoperation übersprungen wurden.

**oRowsLoaded**

Ausgabe. Anzahl der Zeilen, die in die Zieltabelle geladen wurden.

**oRowsRejected**

Ausgabe. Anzahl der Datensätze, die nicht geladen werden konnten.

**oRowsDeleted**

Ausgabe. Anzahl der gelöschten doppelten Zeilen.

**oRowsCommitted**

Ausgabe. Die Gesamtzahl der verarbeiteten Datensätze: die Anzahl der erfolgreich in die Datenbank geladenen und dort festgeschriebenen Datensätze und zusätzlich die Anzahl der übersprungenen und zurückgewiesenen Datensätze.

**piHostname**

Eingabe. Der Hostname für den Parameter *iFileTransferCmd*. Bei Angabe von NULL nimmt der Hostname standardmäßig den Wert "nohost" an.

**piFileTransferCmd**

Eingabe. Parameter für den Dateiübertragungsbefehl. Falls er nicht erforderlich ist, muss er auf NULL gesetzt sein. Eine vollständige Beschreibung dieses Parameters finden Sie in "Dienstprogramme für das Versetzen von Daten Handbuch und Referenz".

**piPartFileLocation**

Eingabe. In den Modi PARTITION\_ONLY, LOAD\_ONLY und LOAD\_ONLY\_VERIFY\_PART kann mit diesem Parameter die Position der partitionierten Dateien angegeben werden. Diese Position muss auf allen Partitionen vorhanden sein, die durch die Option *piOutputNodes* angegeben sind.

Beim Dateityp SQL\_CURSOR kann dieser Parameter nicht den Wert NULL haben, und die Position verweist nicht auf einen Pfad, sondern auf einen vollständig qualifizierten Dateinamen. Dies ist der vollständig qualifizierte Basisdateiname der partitionierten Dateien, die auf jeder Ausgabepartition erstellt werden (beim Modus PARTITION\_ONLY), bzw. die Position der Dateien, die aus jeder Partition gelesen werden sollen (beim Modus LOAD\_ONLY). Beim Modus PARTITION\_ONLY werden möglicherweise mehrere Dateien mit dem angegebenen Basisnamen erstellt, falls die Zieltabelle LOB-Spalten enthält. Bei anderen Dateitypen als SQL\_CURSOR wird, wenn der Wert für diesen Parameter NULL ist, standardmäßig auf das aktuelle Verzeichnis verwiesen.

**piOutputNodes**

Eingabe. Die Liste der Ausgabepartitionen für LOAD. Der Wert NULL gibt an, dass alle Knoten verwendet werden, auf denen die Zieltabelle definiert ist.

**piPartitioningNodes**

Eingabe. Die Liste der Partitionierungsknoten. Der Wert NULL gibt an, dass der Standardwert verwendet wird. Eine Beschreibung darüber, wie der Standardwert ermittelt wird, finden Sie im Abschnitt über den Befehl LOAD in "Dienstprogramme für das Versetzen von Daten Handbuch und Referenz".

### piMode

Eingabe. Gibt den Lademodus für partitionierte Datenbanken an. Gültige Werte (definiert in db2ApiDf) sind:

#### DB2LOAD\_PARTITION\_AND\_LOAD

Die Daten werden partitioniert (vielleicht in Parallelverarbeitung) und gleichzeitig auf die entsprechenden Datenbankpartitionen geladen.

#### DB2LOAD\_PARTITION\_ONLY

Die Daten werden partitioniert (vielleicht in Parallelverarbeitung), und die Ausgabe wird in Dateien an einer angegebenen Speicherposition auf jeder Ladepartition geschrieben. Für andere Dateitypen als SQL\_CURSOR hat der Name der Ausgabedatei in jeder Partition die Form `dateiname.xxx`. Dabei ist `dateiname` der ersten Eingabedatei, die von `piSourceList` und `xxx` angegeben wird, die Partitionsnummer. Für den Dateityp SQL\_CURSOR wird der Name der Ausgabedatei in jeder Partition vom Parameter `piPartFileLocation` ermittelt. Weitere Informationen zum Angeben der Speicherposition der Partitionsdatei in jeder Partition finden Sie im Parameter `piPartFileLocation`.

**Anmerkung:** Dieser Modus kann für CLI LOAD nicht verwendet werden.

#### DB2LOAD\_LOAD\_ONLY

Es wird davon ausgegangen, dass die Daten bereits partitioniert sind. Der Partitionierungsprozess wird übersprungen, und die Daten werden gleichzeitig auf die entsprechenden Datenbankpartitionen geladen. Für andere Dateitypen als SQL\_CURSOR wird der Eingabedateiname in jeder Partition in der Form `dateiname.xxx` erwartet. Dabei ist `dateiname` der Name der ersten Datei, die von `piSourceList` angegeben wird und `xxx` ist die dreistellige Partitionsnummer. Für den Dateitypen SQL\_CURSOR wird der Name der Eingabedatei in jeder Partition vom Parameter `piPartFileLocation` ermittelt. Weitere Informationen zum Angeben der Speicherposition der Partitionsdatei in jeder Partition finden Sie im Parameter `piPartFileLocation`.

**Anmerkung:** Dieser Modus kann nicht verwendet werden, wenn eine Datendatei geladen wird, die sich auf einem fernen Client befindet und er kann nicht für CLI LOAD verwendet werden.

#### DB2LOAD\_LOAD\_ONLY\_VERIFY\_PART

Es wird davon ausgegangen, dass die Daten bereits partitioniert sind, aber die Datendatei enthält keine Partitionskopfdaten. Der Partitionierungsprozess wird übersprungen, und die Daten werden gleichzeitig auf die entsprechenden Datenbankpartitionen geladen. Während der Ladeoperation wird für jede Zeile geprüft, ob sie sich auf der korrekten Partition befindet. Zeilen, die ungültige Partitionierungen enthalten, werden in eine Speicherauszugsdatei gestellt, sofern der Änderungswert für den Dateityp `dumpfile` angegeben wurde. Andernfalls werden die Zeilen gelöscht. Wenn für eine bestimmte Ladepartition eine Partitionsungültigkeit vorliegt, wird für diese Partition eine Warnung in die LOAD-Nachrichtendatei geschrieben. Der Eingabedateiname in jeder Partition wird in der

Form `dateiname.xxx` erwartet. Dabei ist `dateiname` der Name der ersten Datei, die von `piSourceList` angegeben wird und `xxx` ist die dreistellige Partitionsnummer.

**Anmerkung:** Dieser Modus kann nicht verwendet werden, wenn eine Datendatei geladen wird, die sich auf einem fernem Client befindet und er kann nicht für CLI LOAD verwendet werden.

#### DB2LOAD\_ANALYZE

Es wird eine optimale Partitionierungszuordnung mit einer gleichmäßigen Verteilung auf alle Datenbankpartitionen generiert.

#### piMaxNumPartAgents

Eingabe. Die maximal zulässige Anzahl von Partitionierungsagenten. Der Wert NULL gibt an, dass der Standardwert 25 verwendet wird.

#### piIsolatePartErrs

Eingabe. Gibt an, wie die Ladeoperation auf Fehler reagiert, die auf einzelnen Partitionen auftreten. Gültige Werte (definiert in `db2ApiDf`) sind:

#### DB2LOAD\_SETUP\_ERRS\_ONLY

In diesem Modus wird bei Fehlern, die während der SETUP-Phase auf einer Partition auftreten (beispielsweise Fehler beim Zugriff auf eine Partition oder Probleme beim Zugriff auf einen Tabellenbereich bzw. auf eine Tabelle auf einer Partition), die Ladeoperation auf der Partition mit dem Fehler gestoppt, auf den übrigen Partitionen jedoch fortgesetzt. Fehler, die auf einer Partition auftreten, während Daten geladen werden, bewirken das Fehlschlagen der gesamten Operation und das Zurücksetzen in den letzten Konsistenzzustand auf jeder Partition.

#### DB2LOAD\_LOAD\_ERRS\_ONLY

In diesem Modus bewirken Fehler, die während der SETUP-Phase auf einer Partition auftreten, das Fehlschlagen der gesamten Ladeoperation. Wenn beim Laden der Daten ein Fehler auftritt, werden die Partitionen mit Fehlern auf ihren letzten Konsistenzzustand zurückgesetzt. Die Ladeoperation wird für die verbleibenden Partitionen fortgesetzt, bis ein Fehler auftritt oder bis alle Daten geladen wurden. Auf den Partitionen, auf denen alle Daten geladen wurden, sind die Daten nach der Ladeoperation nicht erkennbar. Aufgrund der Fehler auf den anderen Partitionen wird die Transaktion abgebrochen. Die Daten bleiben auf allen Partitionen nicht erkennbar, bis eine LOAD RESTART-Operation ausgeführt wird. Dies macht die neu geladenen Daten auf den Partitionen erkennbar, auf denen die Ladeoperation vollständig ausgeführt wurde. Auf Partitionen mit Fehlern wird die Ladeoperation hierdurch wieder aufgenommen.

**Anmerkung:** Dieser Modus kann nicht verwendet werden, wenn `iAccessLevel` auf `SQLU_ALLOW_READ_ACCESS` festgelegt ist und auch ein Kopierziel angegeben ist.

#### DB2LOAD\_SETUP\_AND\_LOAD\_ERRS

In diesem Modus führen Fehler auf Partitionesebene während der SETUP-Phase oder dem Laden von Daten dazu, dass die Verarbeitung nur auf den betroffenen Partitionen gestoppt wird. Wie im Modus `DB2LOAD_LOAD_ERRS_ONLY` bleiben die Daten - falls

beim Laden der Daten Partitionsfehler auftreten - auf allen Partitionen nicht erkennbar, bis eine LOAD RESTART-Operation ausgeführt wird.

**Anmerkung:** Dieser Modus kann nicht verwendet werden, wenn *iAccessLevel* auf `SQLU_ALLOW_READ_ACCESS` festgelegt ist und auch ein Kopierziel angegeben ist.

### **DB2LOAD\_NO\_ISOLATION**

Bei diesem Modus bewirkt jeder Fehler während der Ladeoperation, dass die Transaktion abgebrochen wird.

Wenn dieser Parameter NULL ist, wird standardmäßig der Modus `DB2LOAD_LOAD_ERRS_ONLY` verwendet, sofern nicht *iAccessLevel* auf den Wert `SQLU_ALLOW_READ_ACCESS` gesetzt und ein Kopierziel angegeben ist. Dann wird der Modus `DB2LOAD_NO_ISOLATION` als Standardwert verwendet.

### **piStatusInterval**

Eingabe. Gibt an, wie viele Megabyte (MB) Daten geladen werden sollen, bevor eine Statusnachricht generiert wird. Zulässige Werte sind ganze Zahlen im Bereich von 1 bis 4000. Bei Angabe von NULL wird der Standardwert 100 verwendet.

### **piPortRange**

Eingabe. Der TCP-Portbereich für die interne Kommunikation. Bei NULL wird der Portbereich 6000-6063 verwendet.

### **piCheckTruncation**

Eingabe. Bewirkt, dass LOAD bei der Ein-/Ausgabe eine Überprüfung auf abgeschnittene Datensätze vornimmt. Gültige Werte sind TRUE und FALSE. Bei NULL lautet der Standardwert FALSE.

### **piMapFileInput**

Eingabe. Eingabedateiname für die Partitionszuordnung. Bei einem anderen Modus als ANALYZE muss dieser Parameter auf NULL gesetzt sein. Lautet der Modus ANALYZE, muss dieser Parameter angegeben werden.

### **piMapFileOutput**

Eingabe. Ausgabedateiname für die Partitionszuordnung. Die für 'piMapFileInput' angegebenen Regeln gelten auch für diesen Parameter.

### **piTrace**

Eingabe. Gibt die Anzahl der Datensätze an, für die ein Trace ausgeführt werden soll, wenn Sie einen Speicherauszug des gesamten Datenkonvertierungsprozesses und die Ausgabe der Hash-Werte prüfen müssen. Bei NULL wird standardmäßig der Wert 0 für die Anzahl der Datensätze verwendet.

### **piNewline**

Eingabe. Bewirkt, dass LOAD prüft, ob am Ende von ASC-Datensätzen Zeilenvorschubzeichen vorhanden sind, wenn der Änderungswert für Dateityp RECLLEN ebenfalls angegeben ist. Gültige Werte sind TRUE und FALSE. Bei NULL lautet der Standardwert FALSE.

### **piDistfile**

Eingabe. Gibt den Namen der Partitionsverteilungsdatei an. Bei Angabe von NULL wird der Wert DISTFILE als Standardwert verwendet.

### **piOmitHeader**

Eingabe. Gibt an, dass die Partitionsdatei keine Kopfdaten für die

Partitionszuordnung enthalten soll, wenn der Modus DB2LOAD\_PARTITION\_ONLY verwendet wird. Gültige Werte sind TRUE und FALSE. Bei NULL lautet der Standardwert FALSE.

**piRunStatDBPartNum**

Gibt die Datenbankpartition an, für die Statistikdaten erfasst werden sollen. Der Standardwert ist die erste Datenbankpartition in der Ausgabepartitionsliste.

**iHostnameLen**

Eingabe. Gibt die Länge von *piHostname* in Byte an.

**iFileTransferLen**

Eingabe. Gibt die Länge von *piFileTransferCmd* in Byte an.

**iPartFileLocLen**

Eingabe. Gibt die Länge von *piPartFileLocation* in Byte an.

**iMapFileInputLen**

Eingabe. Gibt die Länge von *piMapFileInput* in Byte an.

**iMapFileOutputLen**

Eingabe. Gibt die Länge von *piMapFileOutput* in Byte an.

**iDistfileLen**

Eingabe. Gibt die Länge von *piDistfile* in Byte an.

**piNodeList**

Eingabe. Eine Matrix von Knotennummern.

**iNumNodes**

Eingabe. Die Anzahl der Knoten in der mit *piNodeList* angegebenen Matrix. Der Standardwert 0 gibt an, dass alle Knoten verwendet werden, auf denen die Zieltabelle definiert ist.

**iPortMin**

Eingabe. Niedrigste Portnummer.

**iPortMax**

Eingabe. Höchste Portnummer.

**oRowsRdPartAgents**

Ausgabe. Gesamtanzahl der Zeilen, die von allen Partitionierungsagenten gelesen wurden.

**oRowsRejPartAgents**

Ausgabe. Gesamtanzahl der Zeilen, die von allen Partitionierungsagenten zurückgewiesen wurden.

**oRowsPartitioned**

Ausgabe. Gesamtanzahl der Zeilen, die von allen Partitionierungsagenten partitioniert wurden.

**poAgentInfoList**

Ausgabe. An einer Ladeoperation in einer partitionierten Datenbank können die folgenden Ladeverarbeitungsentitäten beteiligt sein: Ladeagenten, Partitionierungsagenten, Agenten für die Partitionierungsvorbereitung, Agenten für Dateiübertragungsbefehle und Dateiladeagenten (Beschreibungen finden Sie in "Dienstprogramme für das Versetzen von Daten Handbuch und Referenz"). Zweck des Ausgabeparameters *poAgentInfoList* ist die Rückgabe von Informationen über alle an einer Ladeoperation beteiligten Ladeagenten an das aufrufende Programm.

Jeder Eintrag in der Liste enthält die folgenden Angaben:

- *oAgentType*. Eine Kennung dafür, welcher Ladeagenttyp durch den Eintrag beschrieben wird.
- *oNodeNum*. Die Nummer der Partition, auf der der Agent ausgeführt wurde.
- *oSqlcode*. Der endgültige SQLCODE-Wert, der aus der Verarbeitung des Agenten resultierte.
- *oTableState*. Der endgültige Status der Tabelle auf der Partition, auf der der Agent ausgeführt wurde (nur bei Ladeagenten von Bedeutung).

Das aufrufende Programm der API muss dafür sorgen, dass für diese Liste Speicherbereich zugeordnet wird, bevor der Aufruf der API erfolgt. Das aufrufende Programm sollte außerdem die Anzahl der Einträge angeben, für die Speicherbereich zugeordnet wurde. Hierzu wird der Parameter *iMaxAgentInfoEntries* verwendet. Wenn das aufrufende Programm den Parameter *poAgentInfoList* auf NULL oder den Parameter *iMaxAgentInfoEntries* auf 0 setzt, werden keine Informationen zu den Ladeagenten zurückgegeben.

### **iMaxAgentInfoEntries**

Eingabe. Die maximale Anzahl von Informationseinträgen zu Agenten, die durch den Benutzer für *poAgentInfoList* zugeordnet wurde. Im Allgemeinen ist es ausreichend, wenn dieser Parameter auf einen Wert gesetzt ist, der drei Mal so groß ist wie die Anzahl der von der Ladeoperation betroffenen Partitionen.

### **oNumAgentInfoEntries**

Ausgabe. Die tatsächliche Anzahl von Informationseinträgen zu Agenten, die durch die Ladeoperation erstellt wurden. Diese Anzahl der Einträge wird an den Benutzer im Parameter *poAgentInfoList* zurückgegeben, sofern der Wert für *iMaxAgentInfoEntries* größer-gleich dem Wert von *oNumAgentInfoEntries* ist. Falls der Wert für *iMaxAgentInfoEntries* kleiner als der Wert von *oNumAgentInfoEntries* ist, ist die Anzahl der Einträge, die im Parameter *poAgentInfoList* zurückgegeben wird, mit dem Wert von *iMaxAgentInfoEntries* identisch.

### **oSqlcode**

Ausgabe. Der endgültige SQLCODE-Wert, der aus der Verarbeitung des Agenten resultierte.

### **oTableState**

Ausgabe. Zweck dieses Ausgabeparameters ist es, dass nicht jeder mögliche Status der Tabelle nach der Ladeoperation gemeldet wird, sondern nur eine kleine Untermenge möglicher Tabellenstatuswerte. Auf diese Weise hat das aufrufende Programm eine allgemeine Vorstellung davon, was während der Ladeverarbeitung mit der Tabelle passierte. Dieser Wert ist nur bei Ladeagenten von Bedeutung. Gültige Werte:

#### **DB2LOADQUERY\_NORMAL**

Gibt an, dass der Ladevorgang auf der Partition erfolgreich abgeschlossen wurde und dass der Status "Laden läuft" (oder "Laden anstehend") für die Tabelle aufgehoben wurde. In diesem Fall kann die Tabelle weiterhin den Status "Überprüfung anstehend" haben, falls eine weitere Verarbeitung der Integritätsbedingungen erforderlich ist. Dieser Umstand wird jedoch nicht gemeldet, da es sich hierbei um den Normalfall handelt.

#### **DB2LOADQUERY\_UNCHANGED**

Gibt an, dass der Ladejob auf Grund eines Fehlers abgebrochen wurde, jedoch den Status der Tabelle auf der Partition noch nicht

geändert hat. Daher ist der entsprechende Status gültig, der vor dem Aufruf von db2Load vorlag. Es ist nicht erforderlich, für solche Partitionen eine LOAD RESTART- oder LOAD TERMINATE-Operation auszuführen.

#### **DB2LOADQUERY\_LOADPENDING**

Gibt an, dass der Ladejob während der Verarbeitung abgebrochen wurde, die Tabelle auf der Partition jedoch im Status "Laden anstehend" belassen wurde. Dies gibt an, dass der Ladejob auf der entsprechenden Partition entweder beendet oder erneut gestartet werden muss.

#### **oNodeNum**

Ausgabe. Die Nummer der Partition, auf der der Agent ausgeführt wurde.

#### **oAgentType**

Ausgabe. Der Agenttyp. Gültige Werte (definiert in db2ApiDf) sind:

**DB2LOAD\_LOAD\_AGENT**

**DB2LOAD\_PARTITIONING\_AGENT**

**DB2LOAD\_PRE\_PARTITIONING\_AGENT**

**DB2LOAD\_FILE\_TRANSFER\_AGENT**

**DB2LOAD\_LOAD\_TO\_FILE\_AGENT**

#### **Anmerkungen zur Verwendung:**

Die Daten werden in der Reihenfolge geladen, die in der Eingabedatei vorliegt. Falls eine bestimmte Reihenfolge gewünscht wird, müssen die Daten vor dem Durchführen der Ladeoperation sortiert werden.

Das Dienstprogramm LOAD erstellt Indizes gemäß den vorhandenen Definitionen. Mit Hilfe von Ausnahmetabellen werden doppelt vorhandene Werte für eindeutige Schlüssel verarbeitet. Das Dienstprogramm nimmt für Übersichtstabellen, die von den geladenen Tabellen abhängig sind, weder eine Aktualisierung vor, noch setzt es die referenzielle Integrität durch oder überprüft die Integritätsbedingungen. Tabellen, die referenzielle oder zu prüfende Integritätsbedingungen enthalten, werden in den Status "Überprüfung anstehend" versetzt. Übersichtstabellen, für die das Attribut REFRESH IMMEDIATE definiert ist und die von geladenen Tabellen abhängig sind, werden ebenfalls in den Status "Überprüfung anstehend" versetzt. Durch Absetzen der Anweisung SET INTEGRITY können Sie den Status "Überprüfung anstehend" für die Tabellen aufheben. Die Ausführung von Ladeoperationen für replizierte Übersichtstabellen ist nicht möglich.

Bei Clusterindizes müssen die Daten vor dem Laden nach dem Clusterindex sortiert werden. Ein Sortieren der Daten ist nicht erforderlich, wenn sie in eine MDC-Tabelle geladen werden.

#### **Hinweise zu DB2 Data Links Manager**

Für jede DATALINK-Spalte ist eine Spaltenspezifikation in runden Klammern möglich. Jede Spaltenspezifikation umfasst mindestens einen DL\_LINKTYPE-Wert, ein *präfix* und eine DL\_URL\_SUFFIX-Angabe. Der Wert für *präfix* kann entweder DL\_URL\_REPLACE\_PREFIX oder die Spezifikation aus DL\_URL\_DEFAULT\_PREFIX sein.

Es sind so viele DATALINK-Spaltspezifikationen möglich, wie DATALINK-Spalten in der Tabelle definiert sind. Die Reihenfolge der Spezifikationen richtet sich nach der Reihenfolge der DATALINK-Spalten in der Liste der Einfügespalten (sofern diese durch INSERT INTO (*einfügespalte*, ...) angegeben wurde) oder in der Tabellendefinition (wenn keine Liste für *einfügespalte* angegeben wird).

Wenn eine Tabelle beispielsweise die Spalten C1, C2, C3, C4 und C5 enthält, von denen nur C2 und C5 Spalten des Typs DATALINK sind, und die Liste der Einfügespalten (C1, C5, C3, C2) lautet, sollte es zwei Spezifikationen für DATALINK-Spalten geben. Die erste Spaltspezifikation gilt für C5, die zweite für C2. Falls keine Liste der Einfügespalten angegeben ist, gilt die erste Spaltspezifikation für C2 und die zweite für C5.

Wenn mehrere DATALINK-Spalten vorhanden sind und manche Spalten keine bestimmte Spezifikation erfordern, sollte die Spaltspezifikation mindestens die runden Klammern enthalten, damit die Reihenfolge der Spezifikationen eindeutig erkennbar ist. Sind für keine der Spalten Spezifikationen vorhanden, kann die gesamte Liste der leeren Klammern gelöscht werden. Daher ist in Fällen, in denen die Standardwerte ausreichend sind, keine DATALINK-Spezifikation erforderlich.

Wenn Sie Daten in eine Tabelle mit einer DATALINK-Spalte laden, die mit FILE LINK CONTROL definiert ist, führen Sie die folgenden Schritte aus, bevor Sie das Dienstprogramm LOAD aufrufen. (Die folgenden Schritte sind nicht erforderlich, falls alle DATALINK-Spalten mit NO LINK CONTROL definiert sind.)

1. Stellen Sie sicher, dass DB2 Data Links Manager auf den Data Links-Servern, auf die die DATALINK-Spaltenwerte verweisen, installiert ist.
2. Stellen Sie sicher, dass die Datenbank in DB2 Data Links Manager registriert ist.
3. Kopieren Sie alle Dateien, die als DATALINK-Werte eingefügt werden, auf die entsprechenden Data Links-Server.
4. Definieren Sie für die DB2 Data Links Manager auf den Data Links-Servern den bzw. die Präfixnamen.
5. Registrieren Sie die Data Links-Server, auf die in den (zu ladenden) DATALINK-Daten verwiesen wird, in der Konfigurationsdatei von DB2 Data Links Manager.

Es kann vorkommen, dass die Verbindung zwischen DB2 und dem Data Links-Server während der Ausführung des Dienstprogramms LOAD fehlschlägt, wodurch die Ladeoperation fehlschlägt. Gehen Sie in diesem Fall wie folgt vor:

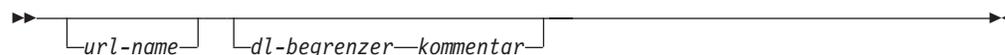
1. Starten Sie den Data Links-Server und den DB2 Data Links Manager.
2. Rufen Sie eine LOAD RESTART-Operation auf.

Während der Ladeoperation fehlgeschlagene Verbindungen werden als eine Art von Datenintegritätsverletzung eingestuft und wie Verletzungen eindeutiger Indizes gehandhabt. Es wurde daher eine besondere Ausnahmebedingung für das Laden von Tabellen definiert, die mindestens eine DATALINK-Spalte enthalten.

### Darstellung von DATALINK-Informationen in einer Eingabedatei

Der LINKTYPE-Wert (gegenwärtig wird nur URL unterstützt) wird nicht als Teil der DATALINK-Informationen angegeben. Der LINKTYPE-Wert wird im Befehl LOAD oder IMPORT angegeben sowie bei Eingabedateien des Typs PC/IXF in den entsprechenden Spaltendeskriptorsätzen.

DATALINK-Informationen für den LINKTYPE-Wert URL haben die folgende Syntax:



Bitte beachten Sie, dass die Angabe von Werten für *url-name* und *kommentar* optional ist. Wenn keiner der beiden Werte angegeben ist, wird der Wert NULL zugeordnet.

#### **url-name**

Der URL-Name muss die gültige URL-Syntax beachten.

#### **Anmerkungen:**

1. Gegenwärtig sind die Werte "http", "file" und "unc" als Schemaname zulässig.
2. Das Präfix (Schema, Host und Port) des URL-Namens ist optional. Wenn kein Präfix vorhanden ist, wird es aus der Spezifikation DL\_URL\_DEFAULT\_PREFIX oder DL\_URL\_REPLACE\_PREFIX für das Dienstprogramm LOAD bzw. das Dienstprogramm IMPORT übernommen. Sind diese nicht angegeben, nimmt das Präfix standardmäßig den Wert "file://localhost" an. Daher kann bei lokalen Dateien der Dateiname mit dem vollständigen Pfadnamen als URL-Name eingegeben werden, ohne dass eine Spezifikation für DATALINK-Spalten im Befehl LOAD oder IMPORT angegeben sein muss.
3. Auch wenn Präfixe in URL-Namen vorhanden sind, werden sie während einer Lade- oder Importoperation außer Kraft gesetzt, falls die Spezifikation für DL\_URL\_REPLACE\_PREFIX einen anderen Präfixnamen enthält.
4. Der Wert für "path" (nach dem Anhängen von DL\_URL\_SUFFIX, sofern angegeben) ist der vollständige Pfadname der fernen Datei auf dem fernen Server. Relative Pfadnamen sind nicht zulässig. Das Standardpfadpräfix "http" für den Server wird nicht berücksichtigt.

#### **dl-begrenzer**

Beim ASCII-Format ohne universelle Zeilenbegrenzer (DEL) ein Zeichen, das über den Änderungswert *dldel* angegeben wird oder im Befehl LOAD bzw. IMPORT standardmäßig verwendet wird. Beim ASCII-Format mit universellen Zeilenbegrenzern (ASC) sollte dies der Zeichenfolge \; (Backslash gefolgt von einem Semikolon) entsprechen. Leerraumzeichen (Leerzeichen, Tabulatorzeichen usw.) sind vor und nach dem Wert, der für diesen Parameter angegeben wird, zulässig.

#### **kommentar**

Der Kommentarteil eines DATALINK-Werts. Bei einer Angabe für das ASCII-Format ohne universelle Zeilenbegrenzer (DEL) muss der für *kommentar* angegebene Text durch den definierten Zeichenfolgebegrenzer eingeschlossen sein. In der Standardeinstellung sind dies doppelte Anführungszeichen ("). Dieser Zeichenfolgebegrenzer kann durch die Spezifikation MODIFIED BY *änderungswert-für-dateityp* des Befehls LOAD oder IMPORT überschrieben werden.

Falls kein Kommentar angegeben wird, wird standardmäßig eine Zeichenfolge mit einer Länge von Null verwendet.

Beispiele für DATALINK-Daten für das ASCII-Dateiformat ohne universelle Zeilenbegrenzer (DEL):

- `http://www.almaden.ibm.com:80/mrep/intro.mpeg; "Intro Movie"`  
Diese Angabe wird mit den folgenden Bestandteilen gespeichert:
  - `scheme = http`
  - `server = www.almaden.ibm.com`
  - `path = /mrep/intro.mpeg`
  - `comment = "Intro Movie"`
- `file://narang/u/narang; "InderPal's Home Page"`  
Diese Angabe wird mit den folgenden Bestandteilen gespeichert:
  - `scheme = file`
  - `server = narang`
  - `path = /u/narang`
  - `comment = "InderPal's Home Page"`

Beispiele für DATALINK-Daten für das ASCII-Dateiformat mit universellen Zeilenbegrenzern (ASC):

- `http://www.almaden.ibm.com:80/mrep/intro.mpeg\;Intro Movie`  
Diese Angabe wird mit den folgenden Bestandteilen gespeichert:
  - `scheme = http`
  - `server = www.almaden.ibm.com`
  - `path = /mrep/intro.mpeg`
  - `comment = "Intro Movie"`
- `file://narang/u/narang\; InderPal's Home Page`  
Diese Angabe wird mit den folgenden Bestandteilen gespeichert:
  - `scheme = file`
  - `server = narang`
  - `path = /u/narang`
  - `comment = "InderPal's Home Page"`

Beispiele für DATALINK-Daten, in denen als Spaltenspezifikation für LOAD oder IMPORT der Wert `DL_URL_REPLACE_PREFIX` ("`http://qso`") angenommen wird:

- `http://www.almaden.ibm.com/mrep/intro.mpeg`  
Diese Angabe wird mit den folgenden Bestandteilen gespeichert:
  - `schema = http`
  - `server = qso`
  - `path = /mrep/intro.mpeg`
  - `comment = NULL string`
- `/u/me/myfile.ps`  
Diese Angabe wird mit den folgenden Bestandteilen gespeichert:
  - `schema = http`
  - `server = qso`
  - `path = /u/me/myfile.ps`
  - `comment = NULL string`

**Zugehörige Referenzen:**

- „sqluvqdp - Quiesce Table Spaces for Table“ in *Administrative API Reference*
- „Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs“ auf Seite 163
- „SQLDCOL“ in *Administrative API Reference*
- „SQLU-MEDIA-LIST“ in *Administrative API Reference*
- „Export-API "db2Export"“ auf Seite 14
- „Import-API "db2Import"“ auf Seite 53
- „db2DatabaseQuiesce - Database Quiesce“ in *Administrative API Reference*
- „db2InstanceQuiesce - Instance Quiesce“ in *Administrative API Reference*
- „Änderungswerte für Dateityp für LOAD“ auf Seite 168
- „Einschränkungen für Begrenzer beim Versetzen von Daten“ auf Seite 247

**Zugehörige Beispiele:**

- „dtformat.sqc -- Load and import data format extensions (C)“
- „tbload.sqc -- How to load into a partitioned database (C)“
- „tbmove.sqc -- How to move table data (C)“
- „tbmove.sqC -- How to move table data (C++)“

---

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Überprüft den Status einer Ladeoperation während der Verarbeitung.

**Berechtigung:**

Keine

**Erforderliche Verbindung:**

Datenbank

**API-Include-Datei:**

*db2ApiDf.h*

**C-API-Syntax:**

```

/* File: db2ApiDf.h */
/* API: db2LoadQuery */
/* ... */
SQL_API_RC SQL_API_FN
db2LoadQuery (
    db2UInt32 versionNumber,
    void *pParmStruct,
    struct sqlca *pSqlca);

typedef struct
{
    db2UInt32 iStringType;
    char *piString;
    db2UInt32 iShowLoadMessages;
    db2LoadQueryOutputStruct *poOutputStruct;
    char *piLocalMessageFile;
} db2LoadQueryStruct;

typedef struct
{

```

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

```
db2UInt32 oRowsRead;
db2UInt32 oRowsSkipped;
db2UInt32 oRowsCommitted;
db2UInt32 oRowsLoaded;
db2UInt32 oRowsRejected;
db2UInt32 oRowsDeleted;
db2UInt32 oCurrentIndex;
db2UInt32 oNumTotalIndexes;
db2UInt32 oCurrentMPPNode;
db2UInt32 oLoadRestarted;
db2UInt32 oWhichPhase;
db2UInt32 oWarningCount;
db2UInt32 oTableState;
} db2LoadQueryOutputStruct;
/* ... */
```

### Generische API-Syntax:

```
/* File: db2ApiDf.h */
/* API: db2gLoadQuery */
/* ... */
SQL_API_RC SQL_API_FN
db2gLoadQuery (
    db2UInt32 versionNumber,
    void *pParmStruct,
    struct sqlca *pSqlca);

typedef struct
{
    db2UInt32 iStringType;
    db2UInt32 iStringLen;
    char *piString;
    db2UInt32 iShowLoadMessages;
    db2LoadQueryOutputStruct *poOutputStruct;
    db2UInt32 iLocalMessageFileLen;
    char *piLocalMessageFile
} db2gLoadQueryStruct;

typedef struct
{
    db2UInt32 oRowsRead;
    db2UInt32 oRowsSkipped;
    db2UInt32 oRowsCommitted;
    db2UInt32 oRowsLoaded;
    db2UInt32 oRowsRejected;
    db2UInt32 oRowsDeleted;
    db2UInt32 oCurrentIndex;
    db2UInt32 oNumTotalIndexes;
    db2UInt32 oCurrentMPPNode;
    db2UInt32 oLoadRestarted;
    db2UInt32 oWhichPhase;
    db2UInt32 oWarningCount;
    db2UInt32 oTableState;
} db2LoadQueryOutputStruct;
/* ... */
```

### API-Parameter:

#### versionNumber

Eingabe. Gibt den Versions- und Release-Stand der Struktur an, die als zweiter Parameter, *pParmStruct*, übergeben wird.

#### pParmStruct

Eingabe. Ein Zeiger auf die *db2LoadQueryStruct*-Struktur.

#### pSqlca

Ausgabe. Ein Zeiger auf die *sqlca*-Struktur.

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

### **iStringType**

Eingabe. Gibt einen Typ für *piString* an. Gültige Werte (definiert in *db2ApiDf.h*) sind:

#### **DB2LOADQUERY\_TABLENAME**

Gibt einen Tabellennamen an, der durch die API "db2LoadQuery" verwendet werden soll.

### **iStringLen**

Eingabe. Gibt die Länge von *piString* in Byte an.

### **piString**

Eingabe. Gibt abhängig vom Wert von *iStringType* einen Pfadnamen für temporäre Dateien oder einen Tabellennamen an.

### **iShowLoadMessages**

Eingabe. Gibt die Stufe der Nachrichten an, die vom Dienstprogramm LOAD zurückgegeben werden. Gültige Werte (definiert in *db2ApiDf.h*) sind:

#### **DB2LOADQUERY\_SHOW\_ALL\_MSGS**

Alle LOAD-Nachrichten zurückgeben.

#### **DB2LOADQUERY\_SHOW\_NO\_MSGS**

Keine LOAD-Nachrichten zurückgeben.

#### **DB2LOADQUERY\_SHOW\_NEW\_MSGS**

Seit dem letzten Aufruf dieser API generierte Nachrichten zurückgeben.

### **poOutputStruct**

Ausgabe. Ein Zeiger auf die *db2LoadQueryOutputStruct*-Struktur, die einen LOAD-Ergebnistext enthält. Auf NULL setzen, wenn keine Zusammenfassung erforderlich ist.

### **iLocalMessageFileLen**

Eingabe. Gibt die Länge von *piLocalMessageFile* in Byte an.

### **piLocalMessageFile**

Eingabe. Gibt den Namen der lokalen Datei an, die für Ausgabenachrichten verwendet werden soll.

### **oRowsRead**

Ausgabe. Anzahl der Datensätze, die bislang vom Dienstprogramm LOAD gelesen wurden.

### **oRowsSkipped**

Ausgabe. Anzahl der Datensätze, die vor Beginn der Ladeoperation übersprungen wurden.

### **oRowsCommitted**

Ausgabe. Anzahl der Zeilen, die bislang in der Zieltabelle festgeschrieben wurden.

### **oRowsLoaded**

Ausgabe. Anzahl der Zeilen, die bislang in die Zieltabelle geladen wurden.

### **oRowsRejected**

Ausgabe. Anzahl der Zeilen, die bislang von der Zieltabelle zurückgewiesen wurden.

### **oRowsDeleted**

Ausgabe. Anzahl der Zeilen, die bislang aus der Zieltabelle gelöscht wurden (in der DELETE-Phase).

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

### **oCurrentIndex**

Ausgabe. Index, der gerade erstellt wird (während der BUILD-Phase).

### **oCurrentMPPNode**

Ausgabe. Gibt an, welcher Datenbankpartitionsserver abgefragt wird (nur beim Modus für Umgebungen mit partitionierten Datenbanken).

### **oLoadRestarted**

Ausgabe. Eine Markierung, deren Wert TRUE ist, wenn die abgefragte Ladeoperation eine LOAD RESTART-Operation ist.

### **oWhichPhase**

Ausgabe. Gibt die aktuelle Phase der abgefragten Ladeoperation an. Gültige Werte (definiert in db2ApiDf.h) sind:

#### **DB2LOADQUERY\_LOAD\_PHASE**

LOAD-Phase

#### **DB2LOADQUERY\_BUILD\_PHASE**

BUILD-Phase

#### **DB2LOADQUERY\_DELETE\_PHASE**

DELETE-Phase.

### **oNumTotalIndexes**

Ausgabe. Gesamtzahl der zu erstellenden Indizes (während der BUILD-Phase).

### **oWarningCount**

Ausgabe. Gesamtzahl der bislang zurückgegebenen Warnungen.

### **oTableState**

Ausgabe. Die Tabellenstatuswerte. Gültige Werte (wie in db2ApiDf definiert) sind:

#### **DB2LOADQUERY\_NORMAL**

Die Tabelle wird nicht durch Statuswerte beeinflusst.

#### **DB2LOADQUERY\_CHECK\_PENDING**

Für die Tabelle gelten Integritätsbedingungen, die noch überprüft werden müssen. Mit dem Befehl SET INTEGRITY können Sie den Status DB2LOADQUERY\_CHECK\_PENDING für die Tabelle aufheben. Das Dienstprogramm LOAD versetzt eine Tabelle in den Status DB2LOADQUERY\_CHECK\_PENDING, sobald eine Ladeoperation für eine Tabelle mit Integritätsbedingungen beginnt.

#### **DB2LOADQUERY\_LOAD\_IN\_PROGRESS**

Es wird gerade eine Ladeoperation für diese Tabelle ausgeführt.

#### **DB2LOADQUERY\_LOAD\_PENDING**

Für diese Tabelle war eine Ladeoperation aktiv, die jedoch abgebrochen wurde, bevor der Ladevorgang festgeschrieben werden konnte. Mit dem Befehl LOAD TERMINATE oder LOAD REPLACE können Sie den Status DB2LOADQUERY\_LOAD\_PENDING für die Tabelle aufheben.

#### **DB2LOADQUERY\_READ\_ACCESS**

Die Tabellendaten sind für Abfragen im Lesezugriff verfügbar. Ladeoperationen mit der Option DB2LOADQUERY\_READ\_ACCESS versetzen die Tabelle in den Status "Nur Lesezugriff".

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

### DB2LOADQUERY\_NOTAVAILABLE

Die Tabelle ist nicht verfügbar. Die Tabelle kann lediglich gelöscht oder aus einer Sicherung wiederhergestellt werden. Bei einer aktualisierenden Wiederherstellung über einen nicht wiederherstellbaren Ladevorgang wird eine Tabelle in den Status "Nicht verfügbar" versetzt.

### DB2LOADQUERY\_NO\_LOAD\_RESTART

Die Tabelle befindet sich in einem teilweise geladenen Status, der eine LOAD RESTART-Operation nicht zulässt. Für die Tabelle gilt außerdem der Status "Laden anstehend". Mit dem Befehl LOAD TERMINATE oder LOAD REPLACE können Sie den Status "Für Laden nicht neu startbar" für die Tabelle aufheben. Die Tabelle kann während einer aktualisierenden Wiederherstellung in den Status DB2LOADQUERY\_NO\_LOAD\_RESTART versetzt werden, beispielsweise, wenn die aktualisierende Wiederherstellung bis zu einem Zeitpunkt ausgeführt wird, der vor dem Ende einer Ladeoperation lag, oder wenn eine aktualisierende Wiederherstellung über eine abgebrochene Ladeoperation hinweg, jedoch nicht bis zum Ende der LOAD TERMINATE- oder LOAD RESTART-Operation ausgeführt wird.

### DB2LOADQUERY\_TYPE1\_INDEXES

Die Tabelle verwendet momentan Indizes des Typs 1. Die Indizes können mit Hilfe der Option CONVERT in Indizes des Typs 2 konvertiert werden, wenn Sie das Dienstprogramm REORG auf die Indizes anwenden.

### Anmerkungen zur Verwendung:

Diese API liest den Status der Ladeoperation für die Tabelle, die mit *piString* angegeben ist, und schreibt den Status in die Datei, die mit *pLocalMsgFileName* angegeben ist.

### Zugehörige Konzepte:

- „Überwachen des Ladens in partitionierte Datenbanken mit LOAD QUERY“ auf Seite 209

### Zugehörige Referenzen:

- „SQLCA“ in *Administrative API Reference*

### Zugehörige Beispiele:

- „loadqry.sqb -- Query the current status of a load (MF COBOL)“
- „tblog.sqc -- How to load into a partitioned database (C)“
- „tbmove.sqc -- How to move table data (C)“
- „tbmove.sqC -- How to move table data (C++)“

## Änderungswerte für Dateityp für LOAD

Tabelle 10. Gültige Änderungswerte für Dateityp für LOAD: Alle Dateiformate

Änderungswert	Beschreibung
anyorder	Dieser Änderungswert wird zusammen mit dem Parameter <i>cpu_parallelism</i> verwendet. Er gibt an, dass die Beibehaltung der Quelldatenreihenfolge nicht erforderlich ist, was zu erheblichen Leistungssteigerungen auf SMP-Systemen führt. Wenn der Wert von <i>cpu_parallelism</i> 1 ist, wird diese Option ignoriert. Diese Option wird nicht unterstützt, wenn <i>SAVECOUNT</i> > 0, da die Wiederherstellung nach einem Systemabsturz nach einem Konsistenzzustand erfordert, dass die Daten geordnet geladen werden.
generatedignore	Dieser Änderungswert informiert das Dienstprogramm LOAD darüber, dass für alle generierten Spalten in der Datendatei Daten vorhanden sind, aber ignoriert werden sollen. Dies bewirkt, dass alle generierten Spaltenwerte durch das Dienstprogramm generiert werden. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert <i>generatedmissing</i> oder dem Änderungswert <i>generatedoverride</i> verwendet werden.
generatedmissing	Wird dieser Änderungswert angegeben, geht das Dienstprogramm davon aus, dass die Eingabedatendatei keine Daten (nicht einmal Nullwerte) für die generierte Spalte enthält. Dies bewirkt, dass alle generierten Spaltenwerte durch das Dienstprogramm generiert werden. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert <i>generatedignore</i> oder dem Änderungswert <i>generatedoverride</i> verwendet werden.
generatedoverride	<p>Dieser Änderungswert weist das Dienstprogramm LOAD an (entgegen der normalen Regeln für diese Spaltentypen), benutzerdefinierte Daten für alle generierten Spalten in der Tabelle zu akzeptieren. Dies ist bei der Migration von Daten aus einem anderen Datenbanksystem oder beim Laden einer Tabelle aus Daten nützlich, die mit der Option <i>RECOVER DROPPED TABLE</i> des Befehls <i>ROLL-FORWARD DATABASE</i> wiederhergestellt wurden. Bei Verwendung dieses Änderungswerts werden alle Zeilen ohne Daten oder mit Nulldaten für eine generierte Spalte, die keine Nullwerte enthalten darf, zurückgewiesen (SQL3116W).</p> <p><b>Anmerkung:</b> Bei Verwendung dieses Änderungswerts wird die Tabelle in den Status "Überprüfung anstehend" versetzt. Um den Status "Überprüfung anstehend" für die Tabelle aufzuheben, ohne die benutzerdefinierten Werte zu überprüfen, setzen Sie nach der Ladeoperation den folgenden Befehl ab:</p> <pre>SET INTEGRITY FOR &lt;tabellenname&gt; GENERATED COLUMN IMMEDIATED UNCHECKED</pre> <p>Um den Status "Überprüfung anstehend" für die Tabelle aufzuheben und die Überprüfung der benutzerdefinierten Werte zu erzwingen, setzen Sie nach der Ladeoperation den folgenden Befehl ab:</p> <pre>SET INTEGRITY FOR &lt; tabellenname &gt; IMMEDIATE CHECKED.</pre> <p>Dieser Änderungswert kann nicht zusammen mit dem Änderungswert <i>generatedmissing</i> oder dem Änderungswert <i>generatedignore</i> verwendet werden.</p>
identityignore	Dieser Änderungswert informiert das Dienstprogramm LOAD darüber, dass für die Identitätsspalte in der Datendatei Daten vorhanden sind, aber ignoriert werden sollen. Dies bewirkt, dass alle Identitätswerte durch das Dienstprogramm generiert werden. Auf diese Weise wird sowohl bei als <i>GENERATED ALWAYS</i> definierten Identitätsspalten als auch bei als <i>GENERATED BY DEFAULT</i> definierten Identitätsspalten verfahren. Dies bedeutet, dass bei als <i>GENERATED ALWAYS</i> definierten Spalten keine Zeilen zurückgewiesen werden. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert <i>identitymissing</i> oder dem Änderungswert <i>identityoverride</i> verwendet werden.

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Tabelle 10. Gültige Änderungswerte für Dateityp für LOAD: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
identitymissing	Wird dieser Änderungswert angegeben, geht das Dienstprogramm davon aus, dass die Eingabedatendatei keine Daten (nicht einmal Nullwerte) für die Identitätsspalte enthält, und generiert daher für jede Zeile einen Wert. Auf diese Weise wird sowohl bei als GENERATED ALWAYS definierten Identitätsspalten als auch bei als GENERATED BY DEFAULT definierten Identitätsspalten verfahren. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert identityignore oder dem Änderungswert identityoverride verwendet werden.
identityoverride	Dieser Änderungswert sollte nur dann verwendet werden, wenn die zu ladende Tabelle eine als GENERATED ALWAYS definierte Identitätsspalte enthält. Er weist das Dienstprogramm an, entgegen der normalen Regeln für diese Typen von Identitätsspalten für eine solche Spalte explizite Daten zu akzeptieren, die keine Nulldaten sind. Dies ist bei der Migration von Daten aus einem anderen Datenbanksystem, bei der die Tabelle als GENERATED ALWAYS definiert sein muss, oder beim Laden einer Tabelle aus Daten nützlich, die mit der Option DROPPED TABLE RECOVERY des Befehls ROLLFORWARD DATABASE wiederhergestellt wurden. Bei Verwendung dieses Änderungswerts werden alle Zeilen ohne Daten oder mit Nulldaten für die Identitätsspalte zurückgewiesen (SQL3116W). Dieser Änderungswert kann nicht zusammen mit dem Änderungswert identitymissing oder dem Änderungswert identityignore verwendet werden. <b>Anmerkung:</b> Wird diese Option verwendet, versucht das Dienstprogramm LOAD nicht, die Eindeutigkeit der Werte in der Identitätsspalte der Tabelle zu verwalten oder zu prüfen.
indexfreespace=x	$x$ ist eine ganze Zahl zwischen 0 und 99 (einschließlich). Der Wert wird als Prozentsatz jeder Indexseite gewertet, der als freier Speicherbereich erhalten bleiben soll, wenn der Index durch das Dienstprogramm LOAD erneut erstellt wird. Bei Ladeoperationen mit INDEXING MODE INCREMENTAL wird diese Option ignoriert. Der erste Eintrag auf einer Seite wird ohne Einschränkung hinzugefügt. Weitere Einträge werden hinzugefügt, wenn die prozentuale Schwelle für freien Speicherbereich eingehalten werden kann. Der Standardwert ist der zum Zeitpunkt der Indexerstellung (CREATE INDEX) verwendete Wert.  Dieser Wert hat Vorrang vor dem in der Anweisung CREATE INDEX angegebenen PCTFREE-Wert. Die Registrierdatenbank-Variable DB2 INDEX FREE hat Vorrang vor "indexfreespace". Die Option "indexfreespace" wirkt sich nur auf Indexblattseiten aus.

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Tabelle 10. Gültige Änderungswerte für Dateityp für LOAD: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
lobsinfile	<p><i>lob-pfad</i> gibt den Pfad zu den Dateien an, die LOB-Daten enthalten. Die ASC-, DEL- oder IXF-Eingabedateien für LOAD enthalten die Namen der Dateien, die LOB-Daten in den LOB-Spalten aufweisen.</p> <p>Die Verwendung dieser Option in Verbindung mit dem Dateityp CURSOR wird nicht unterstützt.</p> <p>Die Klausel LOBS FROM gibt an, wo sich die LOB-Dateien befinden, wenn der Änderungswert „lobsinfile“ verwendet wird. Außerhalb des Kontext des Änderungswerts „lobsinfile“ hat die Klausel LOBS FROM keine Bedeutung. Die Klausel LOBS FROM überträgt die Liste der Pfade an das Dienstprogramm LOAD, die beim Laden der Daten nach LOB-Dateien durchsucht werden sollen.</p> <p>Jeder Pfad enthält mindestens eine Datei, die wiederum mindestens ein großes Objekt (LOB) enthält, auf das durch eine LOB-Positionskenung (LOB Location Specifier - LLS) in der Datendatei verwiesen wird. Die LOB-Positionskenung ist eine Zeichenfolgendarstellung der Position eines großen Objekts (LOB) in einer Datei, die im LOB-Dateipfad gespeichert ist. Das Format einer LOB-Positionskenung lautet <i>dateiname.erw.nnn.mmm/</i>. Hierbei steht <i>dateiname.erw</i> für den Namen der Datei, die das große Objekt (LOB) enthält, <i>nnn</i> für die (in Byte angegebene) relative Position des großen Objekts in der Datei und <i>mmm</i> für die Länge des großen Objekts in Byte. Falls beispielsweise die Zeichenfolge <i>db2exp.001:123:456/</i> in der Datendatei gespeichert ist, befindet sich das große Objekt (LOB) an der relativen Position 123 in der Datei 'db2exp.001' und ist 456 Byte lang.</p> <p>Um ein Null-LOB anzugeben, geben Sie als Größe den Wert -1 ein. Wenn die Größe mit dem Wert 0 angegeben wird, gilt die Angabe als ein großes Objekt mit der Länge 0. Bei Null-LOBs mit der Länge -1 werden die relative Adresse und der Dateiname ignoriert. Die LOB-Positionskenung eines Null-LOBs könnte beispielsweise <i>db2exp.001:7:-1/</i> lauten.</p>
noheader	<p>Überspringt den Prüfcode für die Kopfdaten. (Dieser Änderungswert kann nur beim Laden in Tabellen angewendet werden, die sich in einer Datenbankpartitionsgruppe mit einer Einzelpartition befinden.)</p> <p>Das Dienstprogramm AutoLoader schreibt Kopfdaten in jede Datei, die Daten für eine Tabelle in einer Datenbankpartitionsgruppe mit mehreren Partitionen liefert. Falls der Standardwert für MPP beim Laden für eine Tabelle verwendet wird (Modus PARTITION_AND_LOAD), die sich in einer Datenbankpartitionsgruppe mit einer Partition befindet, wird davon ausgegangen, dass die Datei keine Kopfdaten besitzt. Daher ist die Angabe des Änderungswerts "noheader" nicht erforderlich. Wird der Modus LOAD_ONLY verwendet, muss die Datei Kopfdaten besitzen. Die Verwendung des Änderungswerts "noheader" ist ausschließlich in solchen Situationen erforderlich, in denen eine LOAD_ONLY-Operation unter Verwendung einer Datei ausgeführt werden soll, die keine Kopfdaten besitzt.</p>
norowwarnings	Unterdrückt alle Warnungen zu zurückgewiesenen Zeilen.

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Tabelle 10. Gültige Änderungswerte für Dateityp für LOAD: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
pagefreespace= <i>x</i>	<p><i>x</i> ist eine ganze Zahl zwischen 0 und 100 (einschließlich). Der Wert wird als Prozentsatz jeder Datenseite gewertet, der als freier Speicherbereich freigelassen werden soll. Wenn der angegebene Wert aufgrund der Mindestzeilengröße ungültig ist (z. B. bei einer Zeile, die mindestens 3 000 Byte lang ist und bei einem <i>x</i>-Wert von 50), wird die Zeile auf eine neue Seite gestellt. Wenn der Wert 100 angegeben wird, wird jede Zeile auf eine neue Seite gestellt.</p> <p><b>Anmerkung:</b> Der PCTFREE-Wert einer Tabelle legt die Menge des freien Speicherbereichs pro Seite fest. Wenn kein pagefreespace-Wert für die Ladeoperation oder kein PCTFREE-Wert für die Tabelle festgelegt wurde, füllt das Dienstprogramm so viel Speicherbereich wie möglich auf jeder Seite. Der mit "pagefreespace" festgelegte Wert überschreibt den für die Tabelle angegebenen PCTFREE-Wert.</p>
subtableconvert	<p>Nur beim Laden in eine einzelne untergeordnete Tabelle gültig. Wird typischerweise für den Export von Daten aus einer regulären Tabelle und dem nachfolgenden Aufruf einer Ladeoperation (mit diesem Änderungswert) zur Konvertierung der Daten in einer einzelnen untergeordneten Tabelle verwendet.</p>
totalfreespace= <i>x</i>	<p><i>x</i> ist eine ganze Zahl größer-gleich 0. Der Wert wird als Prozentsatz der Gesamtseiten in der Tabelle gewertet, der als freier Speicherbereich am Ende der Tabelle angefügt werden soll. Wenn Sie beispielsweise für <i>x</i> 20 angeben und die Tabelle nach dem Laden der Daten 100 Datenseiten umfasst, werden 20 zusätzliche leere Seiten angefügt. Die Gesamtzahl der Datenseiten für die Tabelle ist somit 120. Die Gesamtzahl der Datenseiten schlägt sich nicht in der Anzahl der Indexseiten in der Tabelle nieder. Diese Option wirkt sich auf das Indexobjekt nicht aus.</p> <p><b>Anmerkung:</b> Wenn zwei Ladevorgänge mit dieser Option ausgeführt werden, wird der zusätzliche Speicherbereich, der am Ende durch den ersten Ladevorgang angehängt wurde, vom zweiten Ladevorgang nicht erneut verwendet.</p>
usedefaults	<p>Wenn eine Quellenspalte für eine Zieltabellenspalte angegeben wurde, jedoch für mindestens ein Zeilenexemplar keine Daten enthält, werden Standardwerte geladen. Beispiele für fehlende Daten sind:</p> <ul style="list-style-type: none"> <li>• Für DEL-Dateien: ",," wird für die Spalte angegeben.</li> <li>• Für DEL/ASC/WSF-Dateien: Eine Zeile, die nicht genug Spalten enthält oder für die ursprüngliche Spezifikation nicht lang genug ist.</li> </ul> <p>Ohne diese Option geschieht Folgendes, wenn eine Quellenspalte keine Daten für ein Zeilenexemplar enthält:</p> <ul style="list-style-type: none"> <li>• Wenn die Spalte Nullwerte enthalten darf, wird NULL geladen.</li> <li>• Wenn die Spalte keine Nullwerte enthalten darf, wird die Zeile vom Dienstprogramm zurückgewiesen.</li> </ul>

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Tabelle 11. Gültige Änderungswerte für Dateityp für LOAD: ASCII-Dateiformate (ASC/DEL)

Änderungswert	Beschreibung
codepage=x	<p>x ist eine ASCII-Zeichenfolge. Der Wert wird als Codepage der Daten im Eingabedatensatz gewertet. Konvertiert Zeichendaten (und in Zeichenform angegebene numerische Daten) aus dieser Codepage während der Ladeoperation in die Codepage der Datenbank.</p> <p>Dabei gelten die folgenden Regeln:</p> <ul style="list-style-type: none"> <li>• Bei reinem DBCS (Grafik), gemischtem DBCS und EUC sind Begrenzer auf den Bereich von x00 bis x3F (jeweils einschließlich) beschränkt.</li> <li>• Bei DEL-Daten, die in einer EBCDIC-Codepage angegeben werden, dürfen die Begrenzer nicht mit den DBCS-Startzeichen und -Endezeichen identisch sein.</li> <li>• nullindchar muss Symbole angeben, die im Standard-ASCII-Satz zwischen den Codepunkten x20 und x7F (jeweils einschließlich) enthalten sind. Das gilt für ASCII-Symbole und -Codepunkte. Bei EBCDIC-Daten können die entsprechenden Symbole verwendet werden, auch wenn die Codepunkte anders sind.</li> </ul> <p>Die Verwendung dieser Option in Verbindung mit dem Dateityp CURSOR wird nicht unterstützt.</p>
dateformat="x"	<p>x ist das Datumsformat in der Quellendatei.<sup>1</sup> Gültige Datumselemente sind:</p> <p>YYYY - Jahr (vier Ziffern von 0000 bis 9999)  M - Monat (eine oder zwei Ziffern von 1 bis 12)  MM - Monat (zwei Ziffern von 1 bis 12; dieses Element und das Element M schließen sich gegenseitig aus)  D - Tag (eine oder zwei Ziffern von 1 bis 31)  DD - Tag (zwei Ziffern von 1 bis 31; dieses Element und das Element D schließen sich gegenseitig aus)  DDD - Tag des Jahres (drei Ziffern von 001 bis 366; dieses Element und andere Elemente für Tag oder Monat schließen sich gegenseitig aus)</p> <p>Für jedes Element, das nicht angegeben wird, wird der Standardwert 1 zugeordnet. Einige Beispiele für Datumsformate:</p> <p>"D-M-YYYY"  "MM.DD.YYYY"  "YYYYDDD"</p>

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Tabelle 11. Gültige Änderungswerte für Dateityp für LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
dumpfile = x	<p>x ist der vollständig qualifizierte (bezüglich der Serverdatenbankpartition) Name einer Ausnahmedatei, in die zurückgewiesene Zeilen geschrieben werden. Pro Satz werden maximal 32 KB Daten geschrieben. Es folgt ein Beispiel, das zeigt, wie eine Speicherauszugsdatei angegeben wird:</p> <pre>db2 load from data of del modified by dumpfile = /u/user/dateiname insert into tabellenname</pre> <p>Die Datei wird erstellt, und ihr Eigner ist der Exemplareigner. Die Standarddateiberechtigungen können mit dem Änderungswert für Dateityp dumpfileaccessall überschrieben werden.</p> <p><b>Anmerkungen:</b></p> <ol style="list-style-type: none"> <li>1. In einer Umgebung mit partitionierten Datenbanken muss der Pfad für die ladende Datenbankpartition lokal sein, damit gleichzeitig aktive Ladeoperationen nicht versuchen, in dieselbe Datei zu schreiben.</li> <li>2. Der Inhalt der Datei wird in einem asynchronen, gepufferten Modus auf Platte geschrieben. Bei einer fehlgeschlagenen oder unterbrochenen Ladeoperation ist die auf Platte festgeschriebene Anzahl von Datensätzen nicht zweifelsfrei bekannt, und nach einem LOAD RESTART kann die Konsistenz nicht garantiert werden. Die Datei kann nur bei einer Ladeoperation, die nach dem Start in einem Durchlauf ausgeführt wird, als vollständig angesehen werden.</li> <li>3. Dieser Änderungswert unterstützt keine Dateinamen mit mehreren Dateierweiterungen. Beispiel: <pre>dumpfile = /home/svtdbm6/DUMP.FILE</pre> wird vom Dienstprogramm LOAD akzeptiert, <pre>dumpfile = /home/svtdbm6/DUMP.LOAD.FILE</pre> hingegen nicht. </li> </ol>
dumpfileaccessall = x	<p>Erteilt Lesezugriff auf 'OTHERS', wenn eine Speicherauszugsdatei erstellt wird.</p> <p>Dieser Änderungswert für Dateityp ist nur gültig,</p> <ol style="list-style-type: none"> <li>1. wenn er zusammen mit dem Änderungswert für Dateityp dumpfile verwendet wird,</li> <li>2. der Benutzer über das Zugriffsrecht SELECT für die Ladezieltabelle verfügt,</li> <li>3. er auf einer DB2-Serverdatenbankpartition abgesetzt wird, die sich unter einem UNIX-Betriebssystem befindet.</li> </ol>
fastparse	<p>Für vom Benutzer angegebene Spaltenwerte erfolgt eine geringere Syntaxprüfung, und die Leistung wird verbessert. Tabellen, die mit dieser Option geladen werden, besitzen einen richtigen Aufbau, und das Dienstprogramm führt eine hinreichende Datenprüfung durch, um eine Segmentierungsverletzung oder einen Trap-Fehler zu verhindern. Daten im richtigen Format werden auch richtig geladen.</p> <p>Beispiel: Wenn der Wert 123qwr4 als Feldeintrag für eine INTEGER-Spalte in einer ASC-Datei vorkommt, markiert das Dienstprogramm normalerweise einen Syntaxfehler, da dieser Wert keine gültige Zahl darstellt. Bei Angabe von fastparse wird kein Syntaxfehler erkannt, und in das INTEGER-Feld wird eine Zufallszahl geladen. Dieser Änderungswert darf daher nur für fehlerfreie Daten verwendet werden. Die Verwendung dieser Option bei ASCII-Daten kann zu erheblichen Leistungssteigerungen führen..</p> <p>Die Verwendung dieser Option in Verbindung mit dem Dateityp CURSOR oder IXF wird nicht unterstützt.</p>

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Tabelle 11. Gültige Änderungswerte für Dateityp für LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
implieddecimal	<p>Die Position eines implizierten Dezimalzeichens wird durch die Spaltendefinition bestimmt. Sie wird nicht mehr am Ende des Werts angenommen. Zum Beispiel wird der Wert 12345 in eine Spalte, für die DECIMAL(8,2) definiert ist, als 123.45 und <i>nicht</i> als 12345.00 geladen.</p> <p>Dieser Änderungswert kann nicht zusammen mit dem Änderungswert packeddecimal verwendet werden.</p>
timeformat="x"	<p>x ist das Zeitformat in der Quellendatei.<sup>1</sup> Gültige Zeitelemente sind:</p> <ul style="list-style-type: none"> <li>H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System)</li> <li>HH - Stunde (zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus)</li> <li>M - Minute (eine oder zwei Ziffern von 0 bis 59)</li> <li>MM - Minute (zwei Ziffern von 0 bis 59; dieses Element und das Element M schließen sich gegenseitig aus)</li> <li>S - Sekunde (eine oder zwei Ziffern von 0 bis 59)</li> <li>SS - Sekunde (zwei Ziffern von 0 bis 59; dieses Element und das Element S schließen sich gegenseitig aus)</li> <li>SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86399; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus)</li> <li>TT - Meridiananzeiger (AM oder PM)</li> </ul> <p>Für jedes Element, das nicht angegeben wird, wird der Standardwert 0 zugeordnet. Einige Beispiele für Zeitformate:</p> <pre>"HH:MM:SS" "HH.MM TT" "SSSSS"</pre>

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Tabelle 11. Gültige Änderungswerte für Dateityp für LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
timestampformat="x"	<p>x ist das Format der Zeitmarke in der Quellendatei. <sup>1</sup> Gültige Zeitmarkenelemente sind:</p> <p>YYYY - Jahr (vier Ziffern von 0000 bis 9999)</p> <p>M - Monat (eine oder zwei Ziffern von 1 bis 12)</p> <p>MM - Monat (zwei Ziffern von 01 bis 12; dieses Element und das Element M schließen sich gegenseitig aus)</p> <p>MMM - Monat (Abkürzung des Monatsnamen, die aus drei Buchstaben besteht und bei der die Großschreibung zu beachten ist; dieses Element und die Elemente M und MM schließen sich gegenseitig aus)</p> <p>D - Tag (eine oder zwei Ziffern von 1 bis 31)</p> <p>DD - Tag (zwei Ziffern von 1 bis 31; dieses Element und Element D schließen sich gegenseitig aus)</p> <p>DDD - Tag des Jahres (drei Ziffern von 001 bis 366; dieses Element und andere Elemente für Tag oder Monat schließen sich gegenseitig aus)</p> <p>H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System)</p> <p>HH - Stunde (zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus)</p> <p>M - Minute (ein oder zwei Ziffern von 0 bis 59)</p> <p>MM - Minute (zwei Ziffern von 0 bis 59; dieses Element und das Minutenelement M schließen sich gegenseitig aus)</p> <p>S - Sekunde (ein oder zwei Ziffern von 0 bis 59)</p> <p>SS - Sekunde (zwei Ziffern von 0 bis 59; dieses Element und das Element S schließen sich gegenseitig aus)</p> <p>SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86399; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus)</p> <p>UUUUUU - Mikrosekunde (6 Ziffern von 000000 bis 999999; dieses Element und alle anderen Mikrosekundenelemente schließen sich gegenseitig aus)</p> <p>UUUUU - Mikrosekunde (5 Ziffern von 00000 bis 99999; dieses Element wird dem Bereich 000000 bis 999990 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>UUUU - Mikrosekunde (4 Ziffern von 0000 bis 9999; dieses Element wird dem Bereich 000000 bis 999900 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>UUU - Mikrosekunde (3 Ziffern von 000 bis 999; dieses Element wird dem Bereich 000000 bis 999000 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>UU - Mikrosekunde (2 Ziffern von 00 bis 99; dieses Element wird dem Bereich 000000 bis 990000 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>U - Mikrosekunde (1 Ziffer von 0 bis 9; dieses Element wird dem Bereich 000000 bis 900000 zugeordnet und schließt sich mit allen anderen Mikrosekundenelementen gegenseitig aus)</p> <p>TT - Meridiananzeiger (AM oder PM)</p>

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Tabelle 11. Gültige Änderungswerte für Dateityp für LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
timestampformat="x"	<p>Für nicht angegebene Elemente YYYY, M, MM, D, DD oder DDD wird der Standardwert 1 zugeordnet. Der Standardwert 'Jan' wird einem nicht spezifizierten Element MMM zugeordnet. Für alle anderen nicht angegebenen Elemente wird der Standardwert 0 zugeordnet. Beispiel für ein Zeitmarkenformat:</p> <pre>"YYYY/MM/DD HH:MM:SS.UUUUUU"</pre> <p>Die gültigen Werte für das Element MMM sind folgende: 'jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov' und 'dec'. Bei diesen Werten wird die Groß-/Kleinschreibung nicht beachtet.</p> <p>Das folgende Beispiel veranschaulicht, wie Daten in eine Tabelle namens SCHEDULE importiert werden, die benutzerdefinierte Datums- und Zeitformate enthalten:</p> <pre>db2 import from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>
noeofchar	<p>Das wahlfreie Dateiondezeichen x'1A' wird nicht als Ende der Datei erkannt. Die Verarbeitung wird wie bei einem normalen Zeichen fortgesetzt.</p>
usegraphiccodepage	<p>Wenn usegraphiccodepage angegeben wird, wird angenommen, dass sich die Daten, die in Datenfelder von Grafiken oder großen Doppelbytezeichenobjekten (DBCLOB) geladen werden, in der Grafikcodepage befinden. Es wird angenommen, dass sich die übrigen Daten in der Zeichencodepage befinden. Die Grafikcodepage ist der Zeichencodepage zugeordnet. Das Dienstprogramm LOAD ermittelt die Zeichencodepage entweder über den Änderungswert codepage, falls angegeben, oder über die Codepage der Datenbank, wenn der Änderungswert codepage nicht angegeben ist.</p> <p>Dieser Änderungswert sollte zusammen mit der begrenzten Datendatei verwendet werden, die bei einer Wiederherstellung nach einer DROP TABLE-Operation generiert wird, wenn die wiederherzustellende Tabelle Grafikdaten enthält.</p> <p><b>Einschränkungen</b></p> <p>Der Änderungswert usegraphiccodepage DARF NICHT mit DEL- oder ASC-Dateien verwendet werden, die durch das Dienstprogramm EXPORT erstellt werden, weil diese Dateien Daten enthalten, die nur in einer Codepage codiert sind. Der Änderungswert usegraphiccodepage wird auch von den großen Doppelbytezeichenobjekten (DBCLOBs) in Dateien ignoriert.</p>

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Tabelle 12. Gültige Änderungswerte für Dateityp für LOAD: ASC-Dateiformate (ASCII-Format mit universellen Zeilenbegrenzern)

Änderungswert	Beschreibung
binarynumerics	<p>Numerische Daten (nicht jedoch DECIMAL-Daten) müssen im Binärformat und nicht in der Zeichendarstellung vorliegen. Dadurch werden aufwendige Konvertierungen vermieden.</p> <p>Diese Option wird nur mit positionsgebundenem ASC unterstützt, bei dem Sätze mit fester Länge verwendet werden, die mit der Option reclen festgelegt wird. Die Option noeofchar wird vorausgesetzt.</p> <p>Es gelten folgende Regeln:</p> <ul style="list-style-type: none"> <li>• Es wird keine Konvertierung zwischen Datentypen durchgeführt, mit Ausnahme von BIGINT, INTEGER und SMALLINT.</li> <li>• Die Datenlänge muss mit der jeweiligen Zielspaltendefinition übereinstimmen.</li> <li>• FLOAT-Werte müssen im IEEE-Gleitkommaformat vorliegen.</li> <li>• Binärdaten in der Ladequellendatei werden als Big-Endian gewertet, unabhängig von der Plattform, auf der die Ladeoperation ausgeführt wird.</li> </ul> <p><b>Anmerkung:</b> NULL-Werte dürfen in den Daten der Spalten, für die dieser Änderungswert gilt, nicht vorkommen. Leerzeichen (normalerweise als NULL interpretiert) werden als Binärwert interpretiert, wenn dieser Änderungswert verwendet wird.</p>
nochecklengths	<p>Wenn nochecklengths angegeben wird, wird versucht, jede Zeile zu laden, auch wenn die Quelldaten eine Spaltendefinition enthalten, die die Größe der Zieltabellenspalte überschreitet. Solche Zeilen können erfolgreich geladen werden, wenn die Codepagekonvertierung eine Verkleinerung der Quelldaten bewirkt. Beispielsweise könnten 4-Byte-EUC-Daten in der Quelle auf 2-Byte-DBCS-Daten im Ziel verkleinert werden und den halben Speicherplatz belegen. Diese Option ist besonders nützlich, wenn bekannt ist, dass die Quelldaten in jedem Fall Platz finden, unabhängig von abweichenden Spaltendefinitionen.</p>
nullindchar=x	<p>x ist ein einzelnes Zeichen. Ändert das Zeichen, das einen NULL-Wert angibt, in x. Der Standardwert für x ist Y.<sup>2</sup></p> <p>Bei diesem Änderungswert wird bei EBCDIC-Datendateien zwischen Groß-/Kleinschreibung unterschieden, sofern es sich beim Zeichen nicht um ein lateinisches Zeichen handelt. Wenn beispielsweise für das NULL-Anzeigerzeichen der Buchstabe N angegeben wird, wird auch n als NULL-Anzeiger erkannt.</p>

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Tabelle 12. Gültige Änderungswerte für Dateityp für LOAD: ASC-Dateiformate (ASCII-Format mit universellen Zeilenbegrenzern) (Forts.)

Änderungswert	Beschreibung
packeddecimal	<p>Lädt gepackte Dezimaldaten direkt, da der Änderungswert binarynumerics den Feldtyp DECIMAL nicht beinhaltet.</p> <p>Diese Option wird nur mit positionsgebundenem ASC unterstützt, bei dem Sätze mit fester Länge verwendet werden, die mit der Option reclen festgelegt wird. Die Option noeofchar wird vorausgesetzt.</p> <p>Unterstützte Werte für das Vorzeichenhalbbyte sind:</p> <pre> + = 0xC 0xA 0xE 0xF - = 0xD 0xB </pre> <p>NULL-Werte dürfen in den Daten der Spalten, für die dieser Änderungswert gilt, nicht vorkommen. Leerzeichen (normalerweise als NULL interpretiert) werden als Binärwert interpretiert, wenn dieser Änderungswert verwendet wird.</p> <p>Unabhängig von der Server-Plattform wird die Byteanordnung der Binärdaten in der Ladequellendatei als Big-Endian gewertet. Wenn dieser Änderungswert bei einem Windows-Betriebssystem verwendet wird, darf die Byteanordnung nicht zurückgenommen werden.</p> <p>Dieser Änderungswert kann nicht zusammen mit dem Änderungswert implieddecimal verwendet werden.</p>
reclen=x	<p>x ist eine ganze Zahl mit dem Maximalwert 32 767. x Zeichen werden für jede Zeile gelesen, und es wird kein Zeilenvorschubzeichen verwendet, um das Ende der Zeile anzugeben.</p>
striptblanks	<p>Schneidet abschließende Leerzeichen ab, wenn Daten in ein Feld mit variabler Länge geladen werden. Wenn diese Option nicht angegeben wird, werden Leerstellen beibehalten.</p> <p>Diese Option kann nicht zusammen mit striptnulls angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p> <p><b>Anmerkung:</b> Diese Option ersetzt die veraltete Option t, die nur noch aus Gründen der Kompatibilität mit früheren Versionen unterstützt wird.</p>
striptnulls	<p>Schneidet abschließende NULL-Zeichen (0x00-Zeichen) ab, wenn Daten in ein Feld mit variabler Länge geladen werden. Wenn diese Option nicht angegeben wird, werden NULL-Zeichen beibehalten.</p> <p>Diese Option kann nicht zusammen mit striptblanks angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p> <p><b>Anmerkung:</b> Diese Option ersetzt die veraltete Option padwithzero, die nur noch aus Gründen der Kompatibilität mit früheren Versionen unterstützt wird.</p>
zoneddecimal	<p>Lädt gezonte Dezimaldaten, da der Änderungswert BINARYNUMERICS die Feldart DECIMAL nicht enthält. Diese Option wird nur mit positionsgebundenem ASC unterstützt, bei dem Sätze mit fester Länge verwendet werden, die mit der Option RECLen festgelegt wird. Die Option NOEOFCHAR wird vorausgesetzt.</p> <p>Die folgenden Halbbytevorzeichenwerte sind möglich:</p> <pre> + = 0xC 0xA 0xE 0xF - = 0xD 0xB </pre> <p>Für Ziffern werden Werte von 0x0 bis 0x9 unterstützt.</p> <p>Für Zonen werden die Werte 0x3 und 0xF unterstützt.</p>

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Tabelle 13. Gültige Änderungswerte für Dateityp für LOAD: DEL-Dateiformate (ASCII-Format ohne universelle Zeilenbegrenzer)

Änderungswert	Beschreibung
chardelx	<p><math>x</math> ist ein einzelner Zeichenfolgebegrenzer. Der Standardwert ist ein doppeltes Anführungszeichen ("). Das angegebene Zeichen wird anstelle von doppelten Anführungszeichen verwendet, um eine Zeichenfolge einzuschließen.<sup>23</sup> Falls Sie die doppelten Anführungszeichen (") explizit als Zeichenfolgebegrenzer angeben wollen, müssen Sie dies folgendermaßen tun:</p> <pre>modified by chardel""</pre> <p>Auch das einfache Anführungszeichen (') kann wie folgt als Zeichenfolgebegrenzer festgelegt werden:</p> <pre>modified by chardel''</pre>
coldelx	<p><math>x</math> ist ein aus einem einzelnen Zeichen bestehender Spaltenbegrenzer. Der Standardwert ist ein Komma (.). Das angegebene Zeichen wird anstelle eines Kommas verwendet, um das Ende einer Spalte anzugeben.<sup>23</sup></p>
datesiso	Datumsformat. Bewirkt, dass alle Datumswerte im ISO-Format geladen werden.
decplusblank	Zeichen für Pluszeichen. Bewirkt, dass positiven Dezimalwerten statt eines Pluszeichens (+) ein Leerzeichen vorangestellt wird. Die Standardaktion besteht darin, positiven Dezimalwerten ein Pluszeichen voranzustellen.
decptx	<p><math>x</math> ist ein einzelnes Ersatzzeichen für den Punkt als Dezimalzeichen. Der Standardwert ist ein Punkt (.). Das angegebene Zeichen wird anstelle eines Punktes als Dezimalzeichen verwendet.<sup>23</sup></p>
delprioritychar	<p>Die aktuelle Standardpriorität für Begrenzer ist: Satzbegrenzer, Zeichenbegrenzer, Spaltenbegrenzer. Durch diesen Änderungswert werden vorhandene Anwendungen geschützt, die auf die früher geltende Priorität angewiesen sind, indem die Begrenzerpriorität wie folgt geändert wird: Zeichenbegrenzer, Satzbegrenzer, Spaltenbegrenzer. Syntax:</p> <pre>db2 load ... modified by delprioritychar ...</pre> <p>Beispiel anhand der folgenden DEL-Datendatei:</p> <pre>"Smith, Joshua",4000,34.98&lt;zeilenbegrenzer&gt; "Vincent,&lt;zeilenbegrenzer&gt;, is a manager", ... ... 4005,44.37&lt;zeilenbegrenzer&gt;</pre> <p>Wenn der Änderungswert delprioritychar angegeben wird, gibt es in dieser Datendatei nur zwei Zeilen. Der zweite &lt;zeilenbegrenzer&gt; wird als Teil der ersten Datenspalte der zweiten Zeile gewertet, während der erste und dritte &lt;zeilenbegrenzer&gt; als eigentliche Satzbegrenzer gewertet werden. Wenn dieser Änderungswert <i>nicht</i> angegeben wird, gibt es in dieser Datendatei drei Zeilen, jeweils getrennt durch einen &lt;zeilenbegrenzer&gt;.</p>
dldelx	<p><math>x</math> ist ein DATALINK-Begrenzer (Einzelzeichen). Der Standardwert ist ein Semikolon (;). Das angegebene Zeichen wird anstelle eines Semikolons als Trennzeichen zwischen Feldern für einen DATALINK-Wert verwendet. Es ist erforderlich, weil ein DATALINK-Wert mehrere Teilwerte enthalten kann.<sup>234</sup></p> <p><b>Anmerkung:</b> <math>x</math> darf nicht mit dem Zeichen übereinstimmen, das als Zeilen-, Spalten- oder Zeichenfolgebegrenzer angegeben wurde.</p>

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Tabelle 13. Gültige Änderungswerte für Dateityp für LOAD: DEL-Dateiformate (ASCII-Format ohne universelle Zeilenbegrenzer) (Forts.)

Änderungswert	Beschreibung
keepblanks	<p>Erhält die führenden und abschließenden Leerzeichen in allen Feldern des Typs CHAR, VARCHAR, LONG VARCHAR oder CLOB. Ohne diese Option werden alle führenden und abschließenden Leerzeichen, die nicht innerhalb von Zeichenbegrenzern stehen, entfernt, und in die Tabelle wird für jedes leere Feld ein Nullwert eingefügt.</p> <p>Das folgende Beispiel veranschaulicht, wie Daten in eine Tabelle namens TABLE1 geladen werden, wobei alle führenden und abschließenden Leerzeichen in der Datendatei erhalten bleiben:</p> <pre>db2 load from delfile3 of del modified by keepblanks insert into table1</pre>
nochardel	<p>Das Dienstprogramm LOAD wird davon ausgehen, dass alle Bytes zwischen den Spaltenbegrenzern Teil der Spaltendaten sind. Zeichenbegrenzer werden als Teil der Spaltendaten syntaktisch analysiert. Diese Option sollte nicht angegeben werden, wenn die Daten mit Hilfe von DB2 exportiert wurden (sofern nochardel zum Zeitpunkt des Exports angegeben wurde). Sie wird bereitgestellt, um Datendateien von Herstellern zu unterstützen, die nicht über Zeichenbegrenzer verfügen. Eine unsachgemäße Verwendung kann zu Datenverlust oder Fehlern führen.</p> <p>Diese Option kann nicht zusammen mit chardelx, delprioritychar oder nodoubledel angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p>
nodoubledel	Unterdrückt die Erkennung von doppelten Zeichenbegrenzern.

Tabelle 14. Gültige Änderungswerte für Dateityp für LOAD: IXF-Dateiformat

Änderungswert	Beschreibung
forcein	<p>Weist das Dienstprogramm an, Daten trotz Codepage-Abweichungen zu akzeptieren und die Umsetzung zwischen Codepages zu unterdrücken.</p> <p>Zielfelder mit fester Länge werden überprüft, um sicherzustellen, dass sie groß genug für die Daten sind. Wenn der Änderungswert nochecklengths angegeben ist, wird keine Überprüfung durchgeführt, und es wird versucht, jede Zeile zu laden.</p>
nochecklengths	<p>Wenn nochecklengths angegeben wird, wird versucht, jede Zeile zu laden, auch wenn die Quelldaten eine Spaltendefinition enthalten, die die Größe der Zieltabellenspalte überschreitet. Solche Zeilen können erfolgreich geladen werden, wenn die Codepagekonvertierung eine Verkleinerung der Quelldaten bewirkt. Beispielsweise könnten 4-Byte-EUC-Daten in der Quelle auf 2-Byte-DBCS-Daten im Ziel verkleinert werden und den halben Speicherplatz belegen. Diese Option ist besonders nützlich, wenn bekannt ist, dass die Quelldaten in jedem Fall Platz finden, unabhängig von abweichenden Spaltendefinitionen.</p>

### Anmerkungen:

1. Die Datumsformatzeichenfolge muss unbedingt in doppelte Anführungszeichen eingeschlossen werden. Feldtrennzeichen dürfen die folgenden Zeichen nicht enthalten: a-z, A-Z und 0-9. Das Feldtrennzeichen darf nicht mit dem Zeichenbegrenzer oder dem Feldbegrenzer im DEL-Dateiformat identisch sein. Die Verwendung eines Feldtrennzeichens ist wahlfrei, sofern die Anfangs- und Endposition eines Elements eindeutig ist. Dies ist beispielsweise dann nicht der Fall, wenn (abhängig vom Änderungswert) Elemente wie D, H, M oder S verwendet werden, da die Einträge in ihrer Länge variabel sind.

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Bei Zeitmarkenformaten muss darauf geachtet werden, eine Mehrdeutigkeit zwischen den Deskriptoren für Monat und Minute zu vermeiden, da beide den Buchstaben M verwenden. Ein Monatsfeld muss neben anderen Datumsfeldern stehen. Ein Minutenfeld muss neben anderen Zeitfeldern stehen. Die folgenden Zeitmarkenformate sind beispielsweise mehrdeutig:

"M" (Hierdurch könnte sowohl eine Monats- als auch eine Minutenangabe dargestellt sein.)  
"M:M" (In diesem Fall ist nicht eindeutig, welches Element für den Monat und welches für die Minutenangabe steht.)  
"M:YYYY:M" (Hier werden beide Elemente M als Monat interpretiert.)  
"S:M:YYYY" (In diesem Fall steht das Element M sowohl neben einem Zeit- als auch neben einem Datumswert.)

Im Fall einer Mehrdeutigkeit meldet das Dienstprogramm eine Fehlermeldung, und die Operation schlägt fehl.

Die folgenden Zeitmarkenformate sind beispielsweise eindeutig:

"M:YYYY" (Monat)  
"S:M" (Minute)  
"M:YYYY:S:M" (Monat...Minute)  
"M:H:YYYY:M:D" (Minute...Monat)

Einigen Zeichen (z. B. doppelten Anführungszeichen und umgekehrten Schrägstrichen) muss ein Escape-Zeichen vorangestellt sein (beispielsweise \).

- Das Zeichen muss in der Codepage der Quelldaten angegeben werden.

Der Zeichencodepunkt (statt des Zeichensymbols) kann mit der Syntax xJJ bzw. 0xJJ angegeben werden, wobei JJ die hexadezimale Darstellung des Codepunkts ist. Beispiel: Das Zeichen # kann wie folgt als Spaltenbegrenzungszeichen angegeben werden:

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

- Unter Delimiter restrictions for moving data sind Einschränkungen für die Zeichen aufgeführt, die als Überschreibungswerte für Begrenzer verwendet werden können.
- Auch wenn das DATALINK-Begrenzungszeichen in der URL-Syntax ein gültiges Zeichen ist, verliert es im Rahmen der Ladeoperation seine besondere Bedeutung.
- Das Dienstprogramm LOAD setzt keine Warnung ab, wenn versucht wird, nicht unterstützte Dateitypen mit der Option MODIFIED BY zu verwenden. Wenn dies versucht wird, schlägt die Ladeoperation fehl, und es wird ein Fehlercode zurückgegeben.

## Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs

Tabelle 15. Verhalten des Dienstprogramms LOAD bei Verwendung von 'codepage' und 'usegraphiccodepage'

codepage=N	usegraphiccodepage	Verhalten von LOAD
Nicht vorhanden	Nicht vorhanden	Es wird angenommen, dass sich alle Daten in der Datei in der Datenbankcodepage befinden und nicht in der Anwendungscodepage, selbst wenn die Option CLIENT angegeben ist.
Vorhanden	Nicht vorhanden	Es wird angenommen, dass sich alle Daten in der Datei in der Codepage N befinden.  <b>Warnung:</b> Grafikdaten werden beim Laden in die Datenbank beschädigt, wenn N eine Einzelbyte-Codepage ist.
Nicht vorhanden	Vorhanden	Es wird angenommen, dass sich Zeichendaten in der Datei in der Datenbankcodepage befinden, selbst wenn die Option CLIENT angegeben ist. Es wird angenommen, dass sich Grafikdaten in der Codepage der Datenbankgrafikdaten befinden, selbst wenn die Option CLIENT angegeben ist.  Wenn die Datenbankcodepage eine Einzelbyte-Codepage ist, wird angenommen, dass sich alle Daten in der Datenbankcodepage befinden.  <b>Warnung:</b> Grafikdaten werden beim Laden in eine Einzelbyte-Datenbank beschädigt.
Vorhanden	Vorhanden	Es wird angenommen, dass sich Zeichendaten in der Codepage N befinden. Es wird angenommen, dass sich Grafikdaten in der Grafikcodepage von N befinden.  Wenn N eine Einzelbyte- oder Doppelbyte-Codepage ist, wird angenommen, dass alle Daten sich in der Codepage N befinden.  <b>Warnung:</b> Grafikdaten werden beim Laden in die Datenbank beschädigt, wenn N eine Einzelbyte-Codepage ist.

### Zugehörige Referenzen:

- „Befehl LOAD“ auf Seite 112
- „Lade-API "db2Load"“ auf Seite 139
- „Einschränkungen für Begrenzer beim Versetzen von Daten“ auf Seite 247

---

## LOAD-Ausnahmetabelle

Die Ausnahmetabelle ist eine durch den Benutzer erstellte Tabelle, die die Definition der zu ladenden Tabelle nachbildet und einige zusätzliche Spalten beinhaltet. Angegeben wird sie mit der Klausel FOR EXCEPTION des Befehls LOAD. Eine Ausnahmetabelle darf keine Identitätsspalte oder generierte Spalten anderen Typs enthalten. Ist in der Primärtabelle eine Identitätsspalte vorhanden, sollte die entsprechende Spalte in der Ausnahmetabelle nur den Spaltentyp, die Spaltenlänge und die Attribute enthalten, die angeben, ob die Spalte Nullwerte enthalten kann oder nicht. Die Ausnahmetabelle wird verwendet, um Kopien der Zeilen zu speichern, die die Regeln eines eindeutigen Indexes verletzen. Vom Dienstprogramm werden keine Überprüfungen auf ungültige Integritätsbedingungen oder Fremdschlüssel vorgenommen. Hiervon ausgenommen sind Verletzungen der Eindeutigkeit. DATALINK-Ausnahmen werden ebenfalls in der Ausnahmetabelle erfasst.

Eine LOAD-Ausnahmetabelle kann dem Tabellenbereich, in dem sich die geladene Tabelle befindet, oder einem anderen Tabellenbereich zugeordnet sein. In beiden Fällen sollte die LOAD-Ausnahmetabelle derselben Datenbankpartitionsgruppe zugeordnet sein und dieselben Partitionierungsschlüssel aufweisen wie die geladene Tabelle.

Ein eindeutiger Schlüssel ist ein Schlüssel, bei dem keine zwei Werte gleich sind. Der zur Umsetzung dieser Integritätsbedingung verwendete Mechanismus wird als eindeutiger Index bezeichnet. Ein Primärschlüssel ist ein Sonderfall eines eindeutigen Schlüssels. Eine Tabelle kann nicht mehr als einen Primärschlüssel aufweisen.

**Anmerkung:** Zeilen, die vor der Indexerstellung aufgrund ungültiger Daten zurückgewiesen wurden, werden nicht in die Ausnahmetabelle aufgenommen.

Zeilen werden an vorhandene Informationen in der Ausnahmetabelle angefügt. Dies können auch ungültige Zeilen aus früheren Ladeoperationen sein. Wenn die Ausnahmetabelle nur die ungültigen Zeilen der aktuellen Ladeoperation enthalten soll, müssen Sie die bereits vorhandenen Zeilen löschen, bevor Sie das Dienstprogramm aufrufen.

Die Ausnahmetabelle, die mit dem Dienstprogramm LOAD verwendet wird, ist identisch mit den Ausnahmetabellen, die von der Anweisung SET INTEGRITY verwendet werden.

Eine Ausnahmetabelle sollte verwendet werden, wenn Daten geladen werden, die einen eindeutigen Index, aber möglicherweise auch doppelte Datensätze aufweisen. Wenn keine Ausnahmetabelle angegeben wird und doppelte Datensätze auftreten, wird die Ladeoperation fortgesetzt, und es erscheint nur eine Warnung, die auf die gelöschten doppelten Datensätze hinweist. Die Datensätze selbst werden nicht protokolliert.

Nach Beendigung der Ladeoperation können Sie die Informationen in der Ausnahmetabelle verwenden, um fehlerhafte Daten zu korrigieren. Die korrigierten Daten können dann in die Tabelle eingefügt werden.

### Zugehörige Referenzen:

- „Exception tables“ in *SQL Reference, Volume 1*

---

## LOAD-Speicherauszugsdatei

Der Änderungswert *dumpfile* teilt dem Dienstprogramm LOAD den Namen und die Speicherposition der Ausnahmedatei mit, in die die zurückgewiesenen Zeilen geschrieben werden. Bei Ausführung in einer Umgebung mit partitionierten Datenbanken können Zeilen entweder von den Unteragenten für Partitionierung oder den Unteragenten für Laden zurückgewiesen werden. Aus diesem Grund erhält der Name der Speicherauszugsdatei eine Erweiterung, die den Typ des Unteragenten angibt sowie die Partitionsnummer, in der die Ausnahmen generiert wurden. Beispiel: Angenommen, der folgende Wert wurde für die Speicherauszugsdatei (*dumpfile*) angegeben:

```
dumpfile = "/u/username/dumpit"
```

In diesem Fall werden die Zeilen, die vom Unteragenten für Laden auf Partition fünf zurückgewiesen wurden, in einer Datei namens */u/username/dumpit.load.005* gespeichert. Zeilen, die vom Unteragenten für Laden auf Partition zwei zurückgewiesen wurden, werden in einer Datei namens */u/username/dumpit.load.002* gespeichert, und Zeilen, die vom Unteragenten für Partitionierung auf Partition zwei zurückgewiesen wurden, werden in einer Datei namens */u/username/dumpit.part.002* gespeichert usw.

Für Zeilen, die vom Unteragenten für Laden zurückgewiesen werden, gilt Folgendes: Ist die Zeile kürzer als 32 768 Byte, wird der Datensatz vollständig in die Speicherauszugsdatei kopiert. Ist die Zeile länger, wird ein Zeilenfragment (das auch die Endbyte des Datensatzes enthält) in die Datei geschrieben.

Zeilen, die vom Unteragenten für Partitionierung zurückgewiesen werden, werden unabhängig von der Länge des Datensatzes vollständig in die Speicherauszugsdatei kopiert.

### Zugehörige Referenzen:

- „Befehl LOAD“ auf Seite 112

---

## Temporäre Dateien für das Dienstprogramm LOAD

DB2<sup>®</sup> erstellt bei der LOAD-Verarbeitung temporäre Binärdateien. Diese Dateien werden zur Wiederherstellung einer Ladeoperation nach einem Systemabsturz, bei LOAD TERMINATE-Operationen, für Warnungen und Fehlermeldungen sowie die Laufzeitsteuerdaten verwendet. Die temporären Dateien werden entfernt, nachdem die Ladeoperation ohne Fehler beendet wurde.

Die temporären Dateien werden in einen Pfad geschrieben, der mit dem Parameter *temp-pfadname* des Befehls LOAD oder mit dem Parameter *piTempFilesPath* der API **db2Load** angegeben werden kann. Der Standardpfad ist ein Unterverzeichnis des Datenbankverzeichnisses.

Die temporären Dateien befinden sich auf der Server-Maschine, und es wird darauf ausschließlich vom DB2-Exemplar zugegriffen. Daher muss jede Angabe eines Pfadnamens im Parameter *temp-pfadname* die Verzeichnisstruktur auf dem Server und nicht die auf dem Client berücksichtigen. Der Eigner des DB2-Exemplars muss darüber hinaus Schreib- und Lesezugriff für den Pfad haben.

**Anmerkung:** In einem MPP-System sollte sich der Pfad für die temporären Dateien auf einem lokalen Datenträger und nicht auf einem NFS-Mount befinden. Befindet sich der Pfad auf einem NFS-Mount, kommt es bei der Ladeoperation zu signifikanten Leistungseinbußen.

**Achtung:** Die in diesen Pfad geschriebenen temporären Dateien dürfen unter keinen Umständen manipuliert werden. Andernfalls treten bei der Ladeoperation Störungen auf, und die Integrität der Datenbank wird gefährdet.

**Zugehörige Referenzen:**

- „Befehl LOAD“ auf Seite 112
- „Lade-API "db2Load"" auf Seite 139

---

## Protokollsätze für das Dienstprogramm LOAD

Der Dienstprogrammmanager erzeugt Protokollsätze für eine Reihe von DB2<sup>®</sup>-Dienstprogrammen. Dazu zählt auch das Dienstprogramm LOAD. Die folgenden Protokollsätze kennzeichnen den Anfang oder das Ende eines bestimmten Vorgangs während einer Ladeoperation:

- LOAD START. Dieser Protokollsatz ist dem Anfang einer Ladeoperation zugeordnet.
- LOAD DELETE START. Dieser Protokollsatz ist dem Anfang der DELETE-Phase einer Ladeoperation zugeordnet. Die DELETE-Phase wird nur gestartet, wenn doppelte Primärschlüsselwerte vorliegen. Während der DELETE-Phase wird jede Löschoperation für einen Tabellensatz oder einen Indexschlüssel protokolliert.
- LOAD DELETE END. Dieser Protokollsatz ist dem Ende der DELETE-Phase einer Ladeoperation zugeordnet. Diese DELETE-Phase wird während der aktualisierenden Wiederherstellung einer erfolgreichen Ladeoperation wiederholt.
- LOAD PENDING LIST. Dieser Protokollsatz wird geschrieben, wenn eine Ladetransaktion festgeschrieben wird. Er wird anstelle eines normalen Protokollsatzes für das Festschreiben einer Transaktion verwendet.

Die folgende Liste vermittelt einen Überblick über die Protokollsätze, die das Dienstprogramm LOAD - abhängig vom Umfang der Eingabedaten - erstellt:

- Für jeden Speicherbereich, der durch das Dienstprogramm in einem DMS-Tabellenbereich für einen Tabellenbereich zugeordnet oder gelöscht wurde, werden zwei Protokollsätze erstellt.
- Für jeden belegten Block von Identitätswerten wird ein Protokollsatz erstellt.
- Für alle Datenzeilen oder Indexschlüssel, die während der DELETE-Phase einer Ladeoperation gelöscht werden, werden Protokollsätze erstellt.
- Wenn eine Ladeoperation unter Angabe der Optionen ALLOW READ ACCESS und INDEXING MODE INCREMENTAL ausgeführt wird, werden Protokollsätze erstellt, die die Integrität der Indexstruktur verwalten. Die Anzahl der protokollierten Sätze ist wesentlich geringer als bei einer vollständig protokollierten Einfügung in den Index.

**Zugehörige Referenzen:**

- „Befehl LOAD“ auf Seite 112
- „Lade-API "db2Load"" auf Seite 139

---

## Tabellensperren, Statuswerte für Tabellen und Statuswerte für Tabellenbereiche

In den meisten Fällen setzt das Dienstprogramm LOAD Sperren auf Tabellenebene ein, um den Zugriff auf Tabellen einzuschränken. Das Dienstprogramm LOAD versetzt die Tabellenbereiche, die von der Ladeoperation betroffen sind, nicht in den Wartemodus und verwendet Statuswerte für Tabellenbereiche nur bei Ladeoperationen, bei denen die Option COPY NO angegeben ist. Die Stufe der Sperren richtet sich danach, ob bei der Ladeoperation ein Lesezugriff zulässig ist oder nicht. Eine Ladeoperation im Modus "Keinen Zugriff zulassen" (ALLOW NO ACCESS) verwendet für die Dauer des Ladens eine exklusive Sperre des Typs "Z" (Super Exclusive Lock) für die Tabelle. Eine Ladeoperation im Modus "Lesezugriff zulassen" (ALLOW READ ACCESS) erhält und verwaltet eine gemeinsam benutzte Sperre des Typs "S" (Share Lock) für die Dauer der Ladeoperation und erweitert die Sperre auf eine exklusive Sperre des Typs "Z", wenn die Daten festgeschrieben werden.

Bevor eine Ladeoperation im Modus "Lesezugriff zulassen" gestartet wird, wartet das Dienstprogramm LOAD, bis alle Anwendungen, die vor der Ladeoperation gestartet wurden, ihre Sperren für die Zieltabelle freigeben. Da Sperren nicht als persistent definiert sind, werden sie durch Statuswerte für Tabellen ergänzt, die auch dann gültig bleiben, wenn eine Ladeoperation abgebrochen wird. Diese Statuswerte können mit dem Befehl LOAD QUERY geprüft werden. Durch die Verwendung der Option LOCK WITH FORCE zwingt das Dienstprogramm LOAD Anwendungen, die Konflikte verursachende Sperren definiert haben, zum Freigeben der Tabelle, in die Daten geladen werden sollen.

### Sperrverhalten bei Ladeoperationen im Modus "Lesezugriff zulassen" (ALLOW READ ACCESS)

Am Anfang einer Ladeoperation fordert das Dienstprogramm LOAD eine gemeinsam benutzte Sperre des Typs "S" (Share Lock) für die Tabelle an. Diese Sperre wird solange beibehalten, bis die Daten festgeschrieben werden. Die gemeinsam benutzte Sperre des Typs "S" (Share Lock) ermöglicht es Anwendungen mit kompatiblen Sperren, während der Ladeoperation auf die Tabelle zuzugreifen. Anwendungen, die beispielsweise Lesezugriffsabfragen verwenden, können auf die Tabelle zugreifen, wohingegen Anwendungen, die Daten in die Tabelle einfügen wollen, der Zugriff verweigert wird. Sobald das Dienstprogramm LOAD eine gemeinsam benutzte Sperre des Typs "S" (Share Lock) anfordert, wartet es, bis alle Anwendungen, die vor dem Beginn der Ladeoperation Sperren für die Tabelle definiert hatten, diese Sperren freigeben. Dies gilt auch dann, wenn es sich hierbei um kompatible Sperren handelt. Da das Dienstprogramm LOAD die gemeinsam benutzte Sperre des Typs "S" (Share Lock) beim Festschreiben der Daten auf eine exklusive Sperre des Typs "Z" (Super Exclusive Lock) erweitert, kann sich das Festschreiben etwas verzögern, während das Dienstprogramm LOAD darauf wartet, dass Anwendungen mit Konflikte verursachenden Sperren beendet werden.

**Anmerkung:** Während die Ladeoperation darauf wartet, dass die Anwendungen ihre Sperren für die Tabelle freigeben, tritt keine Zeitlimitüberschreitung ein.

### Option LOCK WITH FORCE

Mit der Option LOCK WITH FORCE kann die Beendigung von Anwendungen, die Konflikte verursachende Sperren für die Zieltabelle besitzen, erzwungen werden,

| damit die Ladeoperation fortgesetzt werden kann. Anwendungen, die Konflikte  
| verursachende Sperren für die Systemkatalogtabellen besitzen, werden nicht durch  
| das Dienstprogramm LOAD erzwungen abgemeldet. Wenn die Beendigung einer  
| Anwendung auf dem System durch das Dienstprogramm LOAD erzwungen wird,  
| geht ihre Datenbankverbindung verloren, und ein Fehler (SQL1224N) wird zurück-  
| gegeben.

Bei einer Ladeoperation im Modus "Keinen Zugriff zulassen" (ALLOW NO ACCESS) wird die Beendigung aller Anwendungen, die Tabellensperren definiert haben, beim Start der Ladeoperation erzwungen.

Bei einer Ladeoperation im Modus "Lesezugriff zulassen" (ALLOW READ ACCESS) wird die Beendigung von Anwendungen erzwungen, die die folgenden Sperren definiert haben:

- Tabellensperren, die Konflikte mit einer gemeinsam benutzten Sperre des Typs "S" (Share Lock) für eine Tabelle verursachen (beispielsweise IMPORT oder INSERT).
- Alle Tabellensperren, die in der Festschreibungsphase der Ladeoperation vorhanden sind.

Wird die Option COPY NO bei einer Ladeoperation für eine wiederherstellbare Datenbank angegeben, werden alle Objekte im Zieltabellenbereich im Modus für die gemeinsame Benutzung (SHARE) gesperrt, bevor der Tabellenbereich in den Status "Sicherung anstehend" versetzt wird. Dies erfolgt unabhängig vom Zugriffsmodus. Wenn die Option LOCK WITH FORCE angegeben ist, wird die Beendigung aller Anwendungen erzwungen, die Sperren für Objekte im Tabellenbereich definiert haben, die Konflikte mit einer gemeinsam benutzten Sperre des Typs "S" (Share Lock) verursachen.

### **Statuswerte für Tabellen**

| Neben Sperren verwendet das Dienstprogramm LOAD Statuswerte für Tabellen,  
| um den Zugriff auf Tabellen einzuschränken. Ein Tabellenstatus kann mit dem  
| Befehl LOAD QUERY geprüft werden. Der Befehl LOAD QUERY gibt die folgen-  
| den Statuswerte zurück:

#### **Normal**

Die Tabelle wird nicht durch Statuswerte beeinflusst.

#### **Überprüfung anstehend**

Für die Tabelle gelten Integritätsbedingungen, die noch nicht überprüft wurden. Mit der Anweisung SET INTEGRITY können Sie den Status "Überprüfung anstehend" für die Tabelle aufheben. Das Dienstprogramm LOAD versetzt eine Tabelle in den Status "Überprüfung anstehend", sobald eine Ladeoperation für eine Tabelle mit Integritätsbedingungen beginnt.

#### **Laden läuft**

Es wird gerade eine Ladeoperation für diese Tabelle ausgeführt.

#### **Laden anstehend**

Für diese Tabelle war eine Ladeoperation aktiv, die jedoch abgebrochen wurde, bevor die Daten festgeschrieben werden konnten. Mit dem Befehl LOAD TERMINATE, LOAD RESTART oder LOAD REPLACE können Sie diesen Status für die Tabelle aufheben.

### **Lesezugriff**

Die Tabellendaten sind für Abfragen im Lesezugriff verfügbar. Ladeoperationen mit der Option ALLOW READ ACCESS versetzen die Tabelle in den Status "Lesezugriff".

### **Nicht verfügbar**

Die Tabelle ist nicht verfügbar. Die Tabelle kann lediglich gelöscht oder aus einer Sicherung wiederhergestellt werden. Eine Tabelle wird durch eine aktualisierende Wiederherstellung, die über eine nicht wiederherstellbare Ladeoperation erfolgt, in den Status "Nicht verfügbar" versetzt.

### **Für Laden nicht neu startbar**

Die Tabelle befindet sich in einem teilweise geladenen Status, der eine LOAD RESTART-Operation nicht zulässt. Für die Tabelle gilt außerdem der Status "Laden anstehend". Mit dem Befehl LOAD TERMINATE oder LOAD REPLACE können Sie den Status "Für Laden nicht neu startbar" für die Tabelle aufheben. Eine Tabelle wird in diesen Status versetzt, wenn eine aktualisierende Wiederherstellung nach einer fehlgeschlagenen Ladeoperation ausgeführt wird, die nicht erfolgreich erneut gestartet oder beendet wurde, bzw. wenn eine Wiederherstellungsoperation mit einer Online-sicherung ausgeführt wird, die erstellt wurde, als die Tabelle den Status "Laden läuft" oder "Laden anstehend" aufwies. In beiden Fällen sind die Informationen, die für eine LOAD RESTART-Operation erforderlich sind, unzuverlässig. Der Status "Für Laden nicht neu startbar" verhindert, dass eine LOAD RESTART-Operation stattfinden kann.

### **Indizes des Typs 1**

Die Tabelle verwendet momentan Indizes des Typs 1. Die Indizes können mit Hilfe der Option CONVERT in Indizes des Typs 2 konvertiert werden, wenn Sie das Dienstprogramm REORG auf die Indizes anwenden.

### **Unbekannt**

Der Befehl LOAD QUERY ist nicht in der Lage, den Tabellenstatus zu ermitteln.

Für eine Tabelle können gleichzeitig unterschiedliche Statuswerte gültig sein. Wenn Daten beispielsweise in eine Tabelle mit Integritätsbedingungen geladen werden und die Option ALLOW READ ACCESS angegeben ist, gelten für die Tabelle alle folgenden Statuswerte:

Tabellenstatus:  
Überprüfung anstehend  
Laden läuft  
Lesezugriff

Nach der Ladeoperation, jedoch vor Absetzen der Anweisung SET INTEGRITY wären die folgenden Statuswerte gültig:

Tabellenstatus:  
Überprüfung anstehend  
Lesezugriff

Nach dem Absetzen der Anweisung SET INTEGRITY würde der Tabellenstatus wie folgt lauten:

Tabellenstatus:  
Normal

## Statuswerte für Tabellenbereiche bei angegebener Option COPY NO

Wenn eine Ladeoperation mit der Option COPY NO in einer wiederherstellbaren Datenbank ausgeführt wird, werden die Tabellenbereiche, die der Ladeoperation zugeordnet sind, in den Tabellenbereichsstatus "Sicherung anstehend" und den Tabellenbereichsstatus "Laden läuft" versetzt. Dies findet am Beginn der Ladeoperation statt. Die Ladeoperation kann sich an diesem Punkt verzögern, während Sperren für die Tabellen im Tabellenbereich angefordert werden.

Wenn sich ein Tabellenbereich im Status "Sicherung anstehend" befindet, ist er weiterhin im Lesezugriff verfügbar. Der Status "Sicherung anstehend" für einen Tabellenbereich kann nur dadurch aufgehoben werden, dass eine Sicherung des Tabellenbereichs vorgenommen wird. Der Tabellenbereich verbleibt auch bei einem Abbruch der Ladeoperation im Status "Sicherung anstehend", weil der Status des Tabellenbereichs am Beginn der Ladeoperation geändert wurde und bei einem Fehlschlagen der Operation nicht zurückgesetzt werden kann. Der Status "Laden läuft" für einen Tabellenbereich verhindert, dass während dem Laden von Daten Onlinesicherungen von einer Ladeoperation mit angegebener Option COPY NO ausgeführt werden. Der Status "Laden läuft" wird entfernt, sobald die Ladeoperation abgeschlossen ist oder abgebrochen wird.

Während einer aktualisierenden Wiederherstellung über einen Befehl LOAD mit angegebener Option COPY NO werden die zugeordneten Tabellenbereiche in den Status "Wiederherstellung anstehend" versetzt. Um den Status "Wiederherstellung anstehend" für Tabellenbereiche aufzuheben, muss eine Wiederherstellung ausgeführt werden. Eine aktualisierende Wiederherstellung versetzt einen Tabellenbereich nur dann in den Status "Wiederherstellung anstehend", wenn die Ladeoperation erfolgreich abgeschlossen wurde.

### Zugehörige Konzepte:

- „Statusangaben für anstehende Aktionen nach einer Ladeoperation“ auf Seite 190

---

## Zeichensatz und NLS

Die DB2<sup>®</sup> UDB-Dienstprogramme zum Versetzen von Daten bieten die folgende Unterstützung in der Landessprache (National Language Support - NLS):

- Die Dienstprogramme IMPORT und EXPORT stellen eine automatische Konvertierung aus einer Client-Codepage in die Server-Codepage bereit.
- Für das Dienstprogramme LOAD können Daten aus einer beliebigen Codepage in die Server-Codepage konvertiert werden, indem bei DEL- und ASC-Dateien der Änderungswert codepage verwendet wird.
- Bei allen Dienstprogrammen werden IXF-Daten automatisch aus ihrer ursprünglichen Codepage (wie in der IXF-Datei gespeichert) in die Server-Codepage konvertiert.

Manchmal kann es aufgrund ungleicher Codepages zu Situationen kommen, in denen die Zeichendaten möglicherweise verlängert oder gekürzt werden. Solche Situationen können bei Zeichensätzen mit erweitertem UNIX-Code (Extended UNIX<sup>®</sup> Code, EUC) und Doppelbytezeichensätzen (DBCS) für Japanisch oder traditionelles Chinesisch auftreten, die möglicherweise verschiedene Längencodierungen für dasselbe Zeichen verwenden. Normalerweise wird die Länge der Eingabedaten vor dem Einlesen von Daten mit der Länge der Zielspalte verglichen. Wenn die Länge der Eingabedaten die Länge der Zielspalte überschreitet, werden Nullwerte

in die Spalte eingefügt, wenn für die Spalte Nullwerte zulässig sind. Andernfalls wird die Anforderung zurückgewiesen. Wenn der Änderungswert `nochecklengths` angegeben ist, wird vorab kein Vergleich durchgeführt, und die Daten werden nach Möglichkeit geladen. Wenn die Daten nach der Umsetzung zu lang sind, wird die entsprechende Zeile zurückgewiesen. Ansonsten werden die Daten geladen.

**Zugehörige Referenzen:**

- „Befehl LOAD“ auf Seite 112

---

## Statusangaben für anstehende Aktionen nach einer Ladeoperation

Das Dienstprogramm LOAD setzt Angaben zum *Tabellenstatus* ein, um die Datenbankkonsistenz während einer Ladeoperation zu gewährleisten. Diese Statuswerte können mit dem Befehl `LOAD QUERY` überprüft werden.

Die LOAD- und BUILD-Phase des LOAD-Prozesses versetzen die Zieltabelle in den Status "Laden läuft". Das Dienstprogramm LOAD versetzt auch Tabellenbereiche in diesen Status, wenn die Option `COPY NO` bei einer wiederherstellbaren Datenbank angegeben wurde. Die Tabellenbereiche behalten diesen Status für die Dauer der Ladeoperation bei und werden wieder in einen normalen Status versetzt, sobald die Transaktion festgeschrieben oder zurückgesetzt wird.

Falls die Option `NO ACCESS` angegeben wurde, kann nicht auf die Tabelle zugegriffen werden, während der Ladevorgang aktiv ist. Wenn die Option `ALLOW READ ACCESS` angegeben wurde, sind die Daten, die vor dem Aufruf des Befehls `LOAD` in der Tabelle vorhanden waren, während der Ladeoperation nur im Lesezugriff verfügbar. Falls die Ladeoperation bei angegebener Option `ALLOW READ ACCESS` fehlschlägt, sind die Daten, die vor der Ladeoperation in der Tabelle vorhanden waren, nach dem Fehler weiterhin im Lesezugriff verfügbar.

Führen Sie eine der folgenden Aktionen aus, um den Status "Laden läuft" aufzuheben, wenn die Ladeoperation fehlgeschlagen ist oder unterbrochen wurde:

- Starten Sie die Ladeoperation erneut. Ermitteln Sie zunächst die Ursache des Fehlers. Verfügte das Dienstprogramm LOAD beispielsweise nicht über genügend Plattenspeicherplatz, fügen Sie dem Tabellenbereich Behälter hinzu, bevor Sie versuchen, eine `LOAD RESTART`-Operation auszuführen.
- Beenden Sie die Ladeoperation.
- Rufen Sie eine `LOAD REPLACE`-Operation für die Tabelle auf, bei der eine Ladeoperation fehlgeschlagen ist.
- Stellen Sie Tabellenbereiche für die Ladetabelle wieder her. Verwenden Sie dazu den Befehl `RESTORE DATABASE` mit der letzten Sicherung eines Tabellenbereichs oder einer Datenbank, und führen Sie danach weitere Wiederherstellungsfunktionen aus.

Tabellenbereiche werden in den Status "Sicherung anstehend" versetzt, wenn der LOAD-Prozess beendet ist und folgende Bedingungen erfüllt sind:

- Datenbankkonfigurationsparameter `logretain` ist auf `Recovery` gesetzt, oder `userexit` ist aktiviert, und
- LOAD-Option `COPY YES` ist nicht angegeben, und
- LOAD-Option `NONRECOVERABLE` ist nicht angegeben.

Der vierte mögliche Status, der dem LOAD-Prozess zugeordnet ist (Status "Überprüfung anstehend"), bezieht sich auf referenzielle Integritätsbedingungen und

Prüfungen auf Integritätsbedingungen, auf DATALINKS-Integritätsbedingungen, auf Integritätsbedingungen für generierte Spalten, auf die Berechnung von gespeicherten Abfragen oder auf die Weitergabe von Zwischenspeichertabellen. Ist beispielsweise eine vorhandene Tabelle eine übergeordnete Tabelle mit einem Primärschlüssel, auf den ein Fremdschlüssel in einer abhängigen Tabelle verweist, werden beide Tabellen (nicht der Tabellenbereich) durch das Ersetzen der Daten in der übergeordneten Tabelle in den Status "Überprüfung anstehend" versetzt. Setzen Sie zur Auswertung einer Tabelle auf referenzielle Integrität und Prüfung auf Integritätsbedingungen nach Beendigung des LOAD-Prozesses die Anweisung SET INTEGRITY ab, wenn die Tabelle im Status "Überprüfung anstehend" verblieben ist.

**Zugehörige Konzepte:**

- „Überprüfen auf ungültige Integritätsbedingungen“ auf Seite 102
- „Tabellensperren, Statuswerte für Tabellen und Statuswerte für Tabellenbereiche“ auf Seite 186

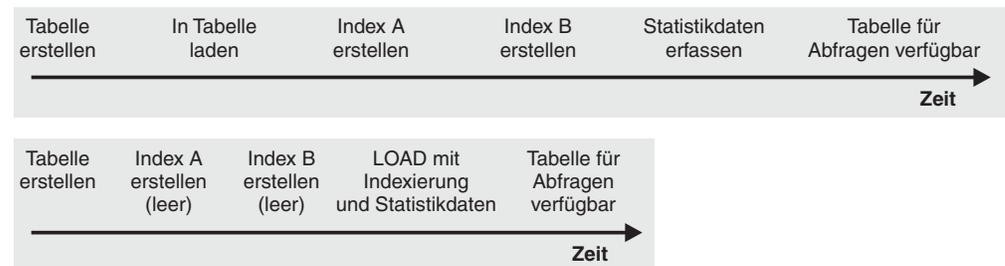
**Zugehörige Referenzen:**

- „LIST TABLESPACES Command“ in *Command Reference*

## Optimieren der Leistung von LOAD

Die Leistung des Dienstprogramms LOAD hängt von der Art und dem Umfang der Daten, der Anzahl der Indizes sowie den angegebenen LOAD-Optionen ab.

Eindeutige Indizes verringern die Leistung von LOAD, wenn Indizes doppelt vorhanden sind. In den meisten Fällen ist es effizienter, Indizes während der Ladeoperation zu erstellen, als nach Beendigung der Ladeoperation für jeden Index die Anweisung CREATE INDEX aufzurufen (siehe Abb. 5).



*Abbildung 5. Erhöhen der Leistung von LOAD durch gleichzeitige Indexierung und Sammlung von Statistikdaten.* Tabellen werden normalerweise in drei Schritten erstellt: Laden von Daten, Erstellen von Indizes und Erfassen Statistikdaten. Das führt zu mehrfacher Daten-E/A während der Ladeoperation, während der Indexerstellung (es sind mehrere Indizes pro Tabelle möglich) und während der Sammlung von Statistikdaten (die zu E/A bei den Tabellendaten und allen Indizes führt). Eine viel schnellere Methode ist, alle diese Aufgaben vom Dienstprogramm LOAD in einem Datendurchlauf ausführen zu lassen.

Bei der Optimierung der Indexierung wird die Speichermenge, die zum Sortieren der Indexschlüssel während einer Ladeoperation verwendet wird, mit dem Datenbankkonfigurationsparameter *sorthheap* gesteuert. Beispiel: Damit vom Dienstprogramm LOAD pro Index 4000 Seiten Hauptspeicher zum Sortieren der Schlüssel verwendet werden, muss der Datenbankkonfigurationsparameter *sorthheap* auf 4000 Seiten gesetzt werden. Dann müssen alle Anwendungen von der Datenbank getrennt werden, und anschließend muss der Befehl LOAD aufgerufen werden. Wenn ein Index so umfangreich ist, dass er nicht im Hauptspeicher sortiert werden

kann, tritt ein Sortierüberlauf ein. Dies bedeutet, dass die Daten auf mehrere "Sortierläufe" aufgeteilt und in einem temporären Tabellenbereich gespeichert werden, der später zusammengeführt wird. Besteht keine Möglichkeit, einen Sortierüberlauf durch ein Heraufsetzen des Parameters *sorthheap* zu verhindern, muss der Pufferpool für temporäre Tabellenbereiche unbedingt groß genug sein, um den Umfang der durch den Überlauf verursachten Platten-E/A zu minimieren. Es empfiehlt sich des Weiteren, temporäre Tabellenbereiche mit mehreren Behältern zu deklarieren, die sich jeweils auf unterschiedlichen Platteneinheiten befinden, damit während der Zusammenführung der Sortierläufe eine E/A-Parallelität erzielt wird.

Die Leistung von LOAD kann durch Installieren von Hochleistungssortierbibliotheken anderer Hersteller verbessert werden, um Indizes während der Ladeoperation zu erstellen. Ein Beispiel für Sortierprogramme anderer Hersteller ist das Produkt "SyncSort". Verwenden Sie die Umgebungsvariable **DB2SORT** (Registrierdatenbankwert), um die Speicherposition der Sortierbibliothek anzugeben, die zur Laufzeit geladen werden soll.

Bei Verwendung der Anweisung SET INTEGRITY verlängert sich unter Umständen die Zeitspanne, die insgesamt erforderlich ist, um eine Tabelle zu laden und wieder verwendbar zu machen. Wenn alle Ladeoperationen im Einfügemodus (INSERT) ausgeführt werden, wird die Tabelle mit der Anweisung SET INTEGRITY inkrementell auf ungültige Integritätsbedingungen überprüft, indem nur der angefügte Teil der Tabelle überprüft wird. Wenn die Tabelle nicht inkrementell auf ungültige Integritätsbedingungen überprüft werden kann, wird die vollständige Tabelle überprüft, und es kann möglicherweise einige Zeit dauern, bis die Tabelle wieder verwendet werden kann.

Ähnlich kann die Verwendung der Anweisung REFRESH TABLE bei einer Ladeoperation für die zu Grunde liegenden Tabellen einer gespeicherten Abfragetabelle unter Umständen die Zeit verlängern, nach der sowohl die zu Grunde liegenden Tabellen als auch die gespeicherte Abfragetabelle wieder vollständig verwendbar sind. Falls mehrere sequenzielle Ladeoperationen im Einfügemodus in den zu Grunde liegenden Tabellen einer gespeicherten Abfragetabelle mit REFRESH IMMEDIATE ausgeführt werden, wird die gespeicherte Abfragetabelle durch die Anweisung SET INTEGRITY in den meisten Fällen inkrementell aktualisiert. Stellt das System fest, dass eine vollständige Aktualisierung erforderlich ist, wird die Definitionsabfrage der gespeicherten Abfragetabelle neu berechnet, und es kann möglicherweise einige Zeit dauern, bis die Tabelle wieder verwendet werden kann.

Das Dienstprogramm LOAD arbeitet im Einfügemodus (INSERT) und im Ersetzungsmodus (REPLACE) annähernd gleich gut.

Das Dienstprogramm liefert die bestmögliche Leistung, indem optimale Werte für DISK\_PARALLELISM, CPU\_PARALLELISM und DATA BUFFER ermittelt werden, wenn diese Parameter nicht vom Benutzer angegeben wurden. Die Optimierung erfolgt auf der Grundlage der Größe und des freien Speicherbereichs, der im Zwischenspeicher der Dienstprogramme verfügbar ist. Ziehen Sie in Betracht, das Dienstprogramm LOAD die Werte für diese Parameter wählen zu lassen und anschließend zu versuchen, die Parameterwerte für Ihre speziellen Anforderungen zu optimieren.

Die folgenden Informationen beziehen sich auf die Auswirkungen der verschiedenen, beim Dienstprogramm LOAD verfügbaren Optionen auf die Leistung.

### ANYORDER

Geben Sie diesen Änderungswert für den Dateityp an, um die Beibehaltung der Reihenfolge der geladenen Daten auszusetzen und die Leistung zu verbessern. Wenn die zu ladenden Daten bereits vorsortiert sind, beschädigt die Angabe von anyorder möglicherweise die Vorsortierung, und die Vorteile der Vorsortierung gehen für nachfolgende Abfragen verloren.

### BINARY NUMERICS und PACKED DECIMAL

Verwenden Sie diese Änderungswerte für den Dateityp, um die Leistung beim Laden von positionsgebundenen numerischen ASC-Daten in Sätze mit fester Länge zu verbessern.

### COPY YES oder NO

Verwenden Sie diesen Parameter, um anzugeben, ob während einer Ladeoperation eine Kopie der Eingabedaten erstellt werden soll. COPY YES verringert die Leistung von LOAD, weil alle Ladedaten während der Ladeoperation kopiert werden (aktualisierende Wiederherstellung muss aktiviert sein). Der Anstieg von E/A-Vorgängen kann unter Umständen zu einer längeren Ladezeit auf einem E/A-abhängigen System führen. Das Angeben mehrerer Einheiten oder Verzeichnisse (auf verschiedenen Platten) kann die durch diese Operation verursachte Leistungseinbuße ausgleichen. COPY NO kann unter Umständen zu einer geringeren Gesamtleistung führen, da der Tabellenbereich bei aktivierter aktualisierender Wiederherstellung in den Status "Sicherung anstehend" gesetzt wird und die Datenbank bzw. ausgewählte Tabellenbereiche gesichert werden müssen, bevor auf die Tabelle zugegriffen werden kann.

### CPU\_PARALLELISM

Mit diesem Parameter können Sie die partitionsinterne Parallelität ausschöpfen (wenn diese von Ihrer Maschine unterstützt wird) und die Leistung von LOAD signifikant verbessern. Mit diesem Parameter wird die Anzahl von Prozessen bzw. Threads angegeben, die vom Dienstprogramm LOAD zur Syntaxanalyse, Konvertierung und Formatierung von Datensätzen verwendet werden. Der maximal zulässige Wert ist 30. Wenn für den angegebenen Wert zu wenig Speicher verfügbar ist, wird der Wert vom Dienstprogramm angepasst. Wenn dieser Parameter nicht angegeben wird, wählt das Dienstprogramm LOAD einen Standardwert auf Grundlage der Anzahl CPUs auf dem System aus.

Die Reihenfolge der Datensätze in den Quelldaten (siehe Abb. 6) bleibt unabhängig vom Wert dieses Parameters erhalten.

Wenn Tabellen Daten des Typs LOB oder LONG VARCHAR enthalten, wird CPU\_PARALLELISM auf den Wert 1 gesetzt. In diesem Fall wird Parallelität nicht unterstützt.

Die Verwendung dieses Parameters ist zwar nicht auf SMP-Hardware begrenzt, doch wird bei Verwendung in Nicht-SMP-Umgebungen möglicherweise keine erkennbare Leistungsverbesserung erzielt.



Abbildung 6. Satzreihenfolge der Quelldaten wird beibehalten, wenn partitionsinterne Parallelität bei einer Ladeoperation ausgeschöpft wird

## DATA BUFFER

Mit dem Parameter DATA BUFFER wird die Gesamtmenge von Speicher angegeben, die dem Dienstprogramm LOAD als Puffer zugeordnet wird. Es wird empfohlen, diesen Puffer auf ein Vielfaches von *extentsz* einzustellen. Ein Speicherbereich ist die Einheit für das Versetzen von Daten innerhalb von DB2®. Die Größe des Speicherbereichs kann eine oder mehr 4 KB-Seiten betragen. Der Parameter DATA BUFFER ist beim Arbeiten mit großen Objekten (LOBs) nützlich; er setzt die E/A-Wartezeit herab. Der Datenpuffer wird aus dem Zwischenspeicher der Dienstprogramme zugeordnet. Je nach der Menge des verfügbaren Speichers auf dem System sollten Sie das Zuordnen von mehr Speicher zur Verwendung durch die DB2-Dienstprogramme in Erwägung ziehen. Der Datenbankkonfigurationsparameter *util\_heap\_sz* kann entsprechend geändert werden. Der Standardwert für den Konfigurationsparameter *util\_heap\_sz* ist 5 000 4 KB-Seiten. Da LOAD nur eines der zahlreichen Dienstprogramme ist, die Speicher aus dem Zwischenspeicher für Dienstprogramme verwenden, wird empfohlen, nicht mehr als fünfzig Prozent der Seiten, die durch diesen Parameter definiert sind, für das Dienstprogramm LOAD zur Verfügung zu stellen. Darüber hinaus muss der Zwischenspeicher für Dienstprogramme groß genug definiert werden.

## DISK\_PARALLELISM

Mit dem Parameter DISK\_PARALLELISM wird die Anzahl von Prozessen bzw. Threads angegeben, die vom Dienstprogramm LOAD zum Schreiben von Datensätzen auf die Platte verwendet werden. Mit diesem Parameter können Sie verfügbare Behälter beim Laden von Daten ausschöpfen und die Leistung von LOAD signifikant verbessern. Der maximal zulässige Wert ist der größere Wert von 50 oder des Vierfachen von CPU\_PARALLELISM. DISK\_PARALLELISM entspricht standardmäßig der Summe der Tabellenbereichsbehälter in allen Tabellenbereichen, die Objekte für die zu ladende Tabelle enthalten, außer wenn dieser Wert den maximal zulässigen Standardwert übersteigt.

## FASTPARSE

Mit dem Änderungswert *fastparse* für den Dateityp können Sie die Datenprüfung verringern, die für vom Benutzer bereitgestellte Spaltenwerte ausgeführt wird, und die Leistung verbessern. Diese Option sollte verwendet werden, wenn an der Gültigkeit der zu ladenden Daten kein Zweifel besteht. Sie kann zu einer um 10 oder 20 Prozent besseren Leistung beitragen.

## NONRECOVERABLE

Verwenden Sie diesen Parameter, wenn LOAD-Transaktionen für eine Tabelle nicht wiederherstellbar sein müssen. Die Leistung von LOAD wird verbessert, weil neben dem Versetzen der Daten in die Tabelle keine zusätzlichen Vorgänge erforderlich sind und die Ladeoperation durchgeführt wird, ohne dass die Tabellenbereiche im Status "Sicherung anstehend" verbleiben.

**Anmerkung:** Wenn diese LOAD-Transaktionen bei nachfolgenden Wiederherstellungen und aktualisierenden Wiederherstellungen vorgefunden werden, wird die Tabelle nicht aktualisiert und als "ungültig" gekennzeichnet. Weitere Funktionen für diese Tabelle werden ignoriert. Nach Abschluss der aktualisierenden Wiederherstellung kann die Tabelle entweder gelöscht oder mit einem Befehl LOAD TERMINATE wieder in den Onlinemodus versetzt werden.

## NOROWWARNINGS

Verwenden Sie den Änderungswert `norowwarnings` für den Dateityp, um die Aufzeichnung von Warnungen zu zurückgewiesenen Zeilen zu unterdrücken und die Leistung zu verbessern, wenn Sie eine große Zahl von Warnungen erwarten.

## ALLOW READ ACCESS

Bei Verwendung dieser Option können Benutzer eine Tabelle abfragen, während eine Ladeoperation läuft. Die Benutzer können nur die Daten anzeigen, die vor der Ladeoperation in der Tabelle vorhanden waren. Wenn die Option `INDEXING MODE INCREMENTAL` ebenfalls angegeben wird und die Ladeoperation fehlschlägt, muss die nachfolgende `LOAD TERMINATE`-Operation möglicherweise Inkonsistenzen im Index korrigieren. Hierzu muss eine Indexsuche ausgeführt werden, die umfangreiche E/A-Vorgänge mit sich bringt. Wenn die Option `ALLOW READ ACCESS` auch für die `LOAD TERMINATE`-Operation angegeben wird, wird der Pufferpool für die Ein-/Ausgabe verwendet.

## SAVECOUNT

Verwenden Sie diesen Parameter, um ein Intervall für die Festlegung von Konsistenzzuständen während einer Ladeoperation einzustellen. Die Synchronisation der Vorgänge zum Herstellen eines Konsistenzzustands dauert einige Zeit. Wenn dies zu häufig geschieht, sinkt die Leistung von `LOAD` merklich. Wenn eine sehr große Zahl von Zeilen geladen werden muss, sollte ein großer Wert für `SAVECOUNT` angegeben werden (z. B. bei einer Ladeoperation mit 100 Millionen Sätzen ein Wert von 10 Millionen).

Eine `LOAD RESTART`-Operation wird automatisch vom letzten Konsistenzzustand ausgehend fortgesetzt.

## STATISTICS YES

Verwenden Sie diesen Parameter, um Datenverteilungs- und Indexstatistiken effizienter zu erfassen als durch den Aufruf des Dienstprogramms `RUNSTATS` nach Beendigung der Ladeoperation. Die Leistung der Ladeoperation selbst hingegen wird verringert (vor allem, wenn `DETAILED INDEXES ALL` angegeben ist).

Die Leistungsstärke der Anwendungen ist am höchsten, wenn sie die bestmöglichen Verteilungs- und Indexstatistiken verwenden. Nach der Aktualisierung der Statistikdaten können die Anwendungen die neuen Zugriffspfade zu den Tabellendaten auf der Grundlage der letzten Statistikdaten verwenden. Neue Zugriffspfade zu einer Tabelle können erstellt werden, indem Sie die Anwendungspakete mit dem DB2-Befehl `BIND` erneut binden.

Beim Laden von Daten in große Tabellen empfiehlt es sich, einen größeren Wert für den Datenbankkonfigurationsparameter `stat_heap_sz` anzugeben.

## USE <tabellenbereichsname>

Bei Verwendung dieses Parameters kann ein Index in einem systemtemporären Tabellenbereich wiederhergestellt und während der `INDEX COPY`-Phase einer Ladeoperation wieder zurück in den Tabellenbereich für den Index kopiert werden. Wenn eine Ladeoperation im Modus "Lesezugriff zulassen" (`ALLOW READ ACCESS`) die Indizes vollständig wiederherstellt, werden die neuen Indizes als *Schattenindizes* erstellt. Am Ende der Ladeoperation werden die ursprünglichen Indizes durch die neuen Indizes ersetzt.

In der Standardeinstellung wird der *Schattenindex* im gleichen Tabellenbereich wie der ursprüngliche Index erstellt. Dies kann unter Umständen zu Ressourcenproblemen führen, da sich sowohl der ursprüngliche Index als auch der *Schattenindex* gleichzeitig in demselben Tabellenbereich befinden. Falls der *Schattenindex* im gleichen Tabellenbereich wie der ursprüngliche Index erstellt wird, wird der ursprüngliche Index sofort durch den *Schattenindex* ersetzt. Wenn der *Schattenindex* jedoch in einem systemtemporären Tabellenbereich erstellt wird, ist bei der Ladeoperation eine INDEX COPY-Phase erforderlich, die den Index aus einem systemtemporären Tabellenbereich in den Tabellenbereich für den Index kopiert. Der Kopiervorgang umfasst umfangreiche E/A-Aktionen. Wenn es sich bei einem der Tabellenbereiche um einen DMS-Tabellenbereich handelt, kann die Ein-/Ausgabe im temporären Systemtabellenbereich möglicherweise nicht sequenziell erfolgen. Während der INDEX COPY-Phase werden die durch die Option DISK\_PARALLELISM angegebenen Werte berücksichtigt.

#### **WARNINGCOUNT**

Verwenden Sie diesen Parameter, um die Anzahl von Warnungen anzugeben, die vom Dienstprogramm zurückgegeben werden kann, bevor eine Ladeoperation beendet wird. Wenn Sie mit nur wenigen oder keinen Warnungen rechnen, stellen Sie den Parameter WARNINGCOUNT auf annähernd die erwartete Anzahl ein. Stellen Sie ihn auf 20 ein, wenn Sie keine Warnungen erwarten. Die Ladeoperation wird nach dem Erreichen des Werts für WARNINGCOUNT gestoppt. Damit haben Sie die Möglichkeit, die Daten zu korrigieren (bzw. zu löschen und die zu ladende Tabelle erneut zu erstellen), bevor Sie versuchen, die Ladeoperation zu beenden. Die Definition einer WARNINGCOUNT-Schwelle hat zwar keine direkte Auswirkung auf die Leistung der Ladeoperation, kann jedoch verhindern, dass Sie erst nach Beendigung der gesamten Ladeoperation feststellen, dass ein Fehler vorliegt.

#### **Zugehörige Konzepte:**

- „Hinweise zum mehrdimensionalen Clustering“ auf Seite 108
- „Registrierdatenbank- und Umgebungsvariablen von DB2“ in *Systemverwaltung: Optimierung*

#### **Zugehörige Referenzen:**

- „util\_heap\_sz - Zwischenspeichergröße für Dienstprogramme“ in *Systemverwaltung: Optimierung*
- „stat\_heap\_sz - Größe des Statistikzwischenspeichers“ in *Systemverwaltung: Optimierung*
- „SET INTEGRITY statement“ in *SQL Reference, Volume 2*
- „BIND Command“ in *Command Reference*
- „UPDATE DATABASE CONFIGURATION Command“ in *Command Reference*

---

## CLP-Beispiele für LOAD

### Beispiel 1

TABLE1 enthält 5 Spalten:

- COL1 VARCHAR 20 NOT NULL WITH DEFAULT
- COL2 SMALLINT
- COL3 CHAR 4
- COL4 CHAR 2 NOT NULL WITH DEFAULT
- COL5 CHAR 2 NOT NULL

ASCFILE1 enthält 6 Elemente:

- ELE1 Positionen 01 bis 20
- ELE2 Positionen 21 bis 22
- ELE3 Positionen 23 bis 23
- ELE4 Positionen 24 bis 27
- ELE5 Positionen 28 bis 31
- ELE6 Positionen 32 bis 32
- ELE7 Positionen 33 bis 40

Datensätze:

```
1...5...10...15...20...25...30...35...40
Testdaten 1          XXN 123abcdN
Testdaten 1 und 3   QQY   XXN
Testdaten 4,5 und 6 WVN6789   Y
```

Mit dem folgenden Befehl wird die Tabelle aus der Datei geladen:

```
db2 load from ascfile1 of asc modified by striptblanks reflen=40
method L (1 20, 21 22, 24 27, 28 31)
null indicators (0,0,23,32)
insert into table1 (col1, col5, col2, col3)
```

### Anmerkungen:

1. Die Angabe `striptblanks` im Parameter `MODIFIED BY` bewirkt, dass Leerzeichen in `VARCHAR`-Spalten (zum Beispiel `COL1`, die 11, 17 und 19 Byte lang ist, in den Zeilen 1, 2 bzw. 3) abgeschnitten werden.
2. Die Angabe `reflen=40` im Parameter `MODIFIED BY` legt fest, dass es am Ende der Eingabedatensätze kein Zeilenvorschubzeichen gibt und jeder Datensatz 40 Byte lang ist. Die letzten 8 Byte werden nicht zum Laden der Tabelle verwendet.
3. Da die Spalte `COL4` nicht in der Eingabedatei bereitgestellt wird, wird sie in die Tabelle `TABLE1` mit dem Standardwert eingefügt (sie ist mit `NOT NULL WITH DEFAULT` definiert).
4. Die Positionen 23 und 32 werden verwendet, um anzugeben, ob in `COL2` und `COL3` von `TABLE1` für eine gegebene Zeile `NULL` geladen wird. Wenn die Nullanzeigerposition der Spalte für einen gegebenen Satz `Y` enthält, ist die Spalte `NULL`. Enthält sie `N`, werden die Datenwerte in den Datenpositionen der Spalte des Eingabedatensatzes (wie in `L(.....)` definiert) als Quelle der Spalten Daten für die Zeile verwendet. In diesem Beispiel ist keine Spalte in Zeile 1 `NULL`. `COL2` in Zeile 2 ist `NULL`, und `COL3` in Zeile 3 ist `NULL`.
5. In diesem Beispiel werden die `NULL`-Anzeiger für `COL1` und `COL5` als 0 (Null) angegeben, wodurch festgelegt wird, dass die Daten keine Nullwerte enthalten dürfen.
6. Der `NULL`-Anzeiger für eine gegebene Spalte kann sich an beliebiger Position im Eingabedatensatz befinden, diese muss jedoch angegeben werden, und der Wert `Y` oder `N` muss ebenfalls angegeben werden.

## Beispiel 2 (unter Verwendung von Speicherauszugsdateien)

Tabelle FRIENDS ist wie folgt definiert:

```
table friends "( c1 INT NOT NULL, c2 INT, c3 CHAR(8) )"
```

Wenn versucht wird, die folgenden Datensätze in diese Tabelle zu laden:

```
23, 24, bobby  
, 45, john  
4,, mary
```

wird die zweite Zeile zurückgewiesen, weil das erste INT NULL ist und die Spaltendefinition NOT NULL lautet. Spalten, die Anfangszeichen enthalten, die nicht mit dem DEL-Format konsistent sind, führen zu einem Fehler, und der Datensatz wird zurückgewiesen. Solche Datensätze können in eine Speicherauszugsdatei geschrieben werden.

DEL-Daten, die in einer Spalte außerhalb von Zeichenbegrenzern vorkommen, werden ignoriert, führen jedoch zur Generierung einer Warnung. Beispiel:

```
22,34,"bob"  
24,55,"sam" sdf
```

Das Dienstprogramm lädt "sam" in die dritte Spalte der Tabelle, und die Zeichen "sdf" werden in einer Warnung markiert. Der Datensatz wird nicht zurückgewiesen. Anderes Beispiel:

```
22 3, 34,"bob"
```

Das Dienstprogramm lädt 22,34,"bob" und generiert eine Warnung, dass einige Daten in Spalte eins nach 22 ignoriert wurden. Der Datensatz wird nicht zurückgewiesen.

## Beispiel 3 (Laden von DATALINK-Daten)

Mit dem folgenden Befehl wird die Tabelle MOVIE TABLE aus der Eingabedatei delfile1 geladen, die Daten im DEL-Format enthält:

```
db2 load from delfile1 of del  
modified by dldel  
insert into movietable (actorname, description, url_making_of,  
url_movie) datalink specification (dl_url_default_prefix  
"http://narang"), (dl_url_replace_prefix "http://bomdel"  
dl_url_suffix ".mpeg") for exception excptab
```

### Anmerkungen:

1. Die Tabelle enthält vier Spalten:

actorname	VARCHAR(n)
description	VARCHAR(m)
url_making_of	DATALINK (mit LINKTYPE URL)
url_movie	DATALINK (mit LINKTYPE URL)

2. Die DATALINK-Daten in der Eingabedatei verwenden den vertikalen Balken (|) als Teilfeldbegrenzer.

3. Wenn ein Spaltenwert für url\_making\_of keine Präfixzeichenfolge aufweist, wird "http://narang" verwendet.

4. Jeder Spaltenwert ungleich NULL für url\_movie erhält "http://bomdel" als Präfix. Vorhandene Werte werden ersetzt.

5. An jeden Spaltenwert ungleich NULL für url\_movie wird ".mpeg" an den Pfad angefügt. Beispiel: Wenn ein Spaltenwert für url\_movie "http://server1/x/y/z" ist, wird er als "http://bomdel/x/y/z.mpeg" gespeichert; ist der Wert "/x/y/z", wird er als "http://bomdel/x/y/z.mpeg" gespeichert.
6. Wenn Verletzungen des eindeutigen Indexes oder DATALINK-Verletzungen auftreten, während die Tabelle geladen wird, werden die betroffenen Datensätze aus der Tabelle gelöscht und in die Ausnahmetabelle excptab gestellt.

#### Beispiel 4 (Laden einer Tabelle mit einer Identitätsspalte)

TABLE1 enthält 4 Spalten:

- C1 VARCHAR(30)
- C2 INT GENERATED BY DEFAULT AS IDENTITY
- C3 DECIMAL(7,2)
- C4 CHAR(1)

TABLE2 entspricht TABLE1, wobei jedoch C2 eine als GENERATED ALWAYS definierte Identitätsspalte ist.

Die Datensätze in DATAFILE1 (DEL-Format) lauten:

```
"Liszt"
"Holst",,187.43, H
"Grieg",100, 66.34, G
"Satie",101, 818.23, I
```

Die Datensätze in DATAFILE2 (DEL-Format) lauten:

```
"Liszt", 74.49, A
"Holst", 0.01, H
"Grieg", 66.34, G
"Satie", 818.23, I
```

#### Anmerkungen:

1. Der folgende Befehl generiert Identitätswerte für die Zeilen 1 und 2, da in der Datei DATAFILE1 für diese Zeilen keine Identitätswerte zur Verfügung gestellt werden. Den Zeilen 3 und 4 wird jedoch der benutzerdefinierte Identitätswert 100 bzw. 101 zugeordnet.

```
db2 load from datafile1.del of del replace into table1
```
2. Um die Datei DATAFILE1 so in die Tabelle TABLE1 zu laden, dass für alle Zeilen Identitätswerte generiert werden, setzen Sie einen der folgenden Befehle ab:

```
db2 load from datafile1.del of del method P(1, 3, 4)
  replace into table1 (c1, c3, c4)
db2load from datafile1.del of del modified by identityignore
  replace into table1
```
3. Um die Datei DATAFILE2 so in die Tabelle TABLE1 zu laden, dass für alle Zeilen Identitätswerte generiert werden, setzen Sie einen der folgenden Befehle ab:

```
db2 load from datafile2.del of del replace into table1 (c1, c3, c4)
db2 load from datafile2.del of del modified by identitymissing
  replace into table1
```
4. Um die Datei DATAFILE1 so in die Tabelle TABLE2 zu laden, dass den Zeilen 3 und 4 die Identitätswerte 100 bzw. 101 zugeordnet werden, setzen Sie den folgenden Befehl ab:

```
db2 load from datafile1.del of del modified by identityoverride
  replace into table2
```

In diesem Fall werden die Zeilen 1 und 2 zurückgewiesen, da das Dienstprogramm angewiesen wurde, die durch das System generierten Identitätswerte mit den benutzerdefinierten Werten zu überschreiben. Sind jedoch keine benutzerdefinierten Werte vorhanden, muss die Zeile zurückgewiesen werden, da Identitätsspalten implizit nicht NULL sind.

5. Wird die Datei DATAFILE1 in die Tabelle TABLE2 geladen, ohne dass einer der identitätsbezogenen Änderungswerte für den Dateityp verwendet wird, werden die Zeilen 1 und 2 geladen, die Zeilen 3 und 4 jedoch zurückgewiesen, da diese Zeilen eigene Werte, die nicht NULL sind, liefern und die Identitätsspalte als GENERATED ALWAYS definiert wurde.

### Beispiel 5 (Laden über CURSOR)

Die Tabelle MY.TABLE1 enthält 3 Spalten:

- ONE INT
- TWO CHAR(10)
- THREE DATE

Die Tabelle MY.TABLE2 enthält 3 Spalten:

- ONE INT
- TWO CHAR(10)
- THREE DATE

Der Cursor MYCURSOR ist wie folgt definiert:

```
declare mycursor cursor for select * from my.table1
```

Mit dem folgenden Befehl werden alle Daten aus der Tabelle MY.TABLE1 in die Tabelle MY.TABLE2 geladen:

```
load from mycursor of cursor method P(1,2,3) insert into  
my.table2(one,two,three)
```

### Anmerkungen:

1. In einem einzelnen Befehl LOAD kann nur ein Cursorname angegeben werden. Dies bedeutet, dass die Angabe load from mycurs1, mycurs2 of cursor... nicht zulässig ist.
2. Die einzigen gültigen Werte für METHOD beim Laden über einen Cursor sind die Werte P und N.
3. Im vorstehenden Beispiel könnten die Angabe METHOD P und die Spaltenliste für das Einfügen (1,2,3) übergangen werden, weil sie Standardwerte darstellen.
4. MY.TABLE1 kann eine Tabelle, eine Sicht, ein Aliasname oder ein Kurzname sein.

### Zugehörige Konzepte:

- „LOAD - Übersicht“ auf Seite 82

### Zugehörige Referenzen:

- „LOAD QUERY“ auf Seite 137
- „Beispiele für LOAD-Sitzungen für partitionierte Datenbanken“ auf Seite 219
- „Befehl LOAD“ auf Seite 112

---

## Kapitel 4. Laden von Daten in einer Umgebung mit partitionierten Datenbanken

Dieses Kapitel beschreibt das Laden von Daten in einer Umgebung mit partitionierten Datenbanken.

Die folgenden Themen werden behandelt:

- „Übersicht über das Laden in partitionierten Datenbanken“
- „Verwenden von LOAD in einer Umgebung mit partitionierten Datenbanken“ auf Seite 203
- „Überwachen des Ladens in partitionierte Datenbanken mit LOAD QUERY“ auf Seite 209
- „Erneutes Starten oder Beenden einer Ladeoperation in einer Umgebung mit partitionierten Datenbanken“ auf Seite 211
- „LOAD-Konfigurationsoptionen für partitionierte Datenbanken“ auf Seite 213
- „Beispiele für LOAD-Sitzungen für partitionierte Datenbanken“ auf Seite 219
- „Migration und Kompatibilität zu früheren Versionen“ auf Seite 222
- „Hinweise und Tipps für das Laden von Daten in einer Umgebung mit partitionierten Datenbanken“ auf Seite 224

---

### Übersicht über das Laden in partitionierten Datenbanken

In einer partitionierten Datenbank werden große Datenmengen partitionsübergreifend angeordnet. Mit Partitionierungsschlüsseln können Sie die Datenbankpartition bestimmen, auf der die einzelnen Datenteile gespeichert sind. Die Daten müssen *partitioniert* werden, bevor sie auf die richtige Datenbankpartition geladen werden können. Beim Laden von Tabellen in eine Umgebung mit partitionierten Datenbanken stellt das Dienstprogramm LOAD das folgende Funktionsspektrum zur Verfügung:

- Paralleles Partitionieren von Eingabedaten
- Gleichzeitiges Laden von Daten auf entsprechende Datenbankpartitionen
- Übertragen von Daten von einem System an ein anderes System

Ladeoperationen für partitionierte Datenbanken bestehen aus zwei Phasen: In einer Setupphase werden Partitionsressourcen wie beispielsweise Tabellensperren angefordert, und in einer LOAD-Phase werden die Daten auf die Partitionen geladen. Mit der Option `ISOLATE_PART_ERRS` des Befehls `LOAD` können Sie auswählen, wie Fehler in diesen beiden Phasen behandelt werden und wie sich Fehler auf einer oder mehreren Partitionen auf die Ladeoperation für diejenigen Partitionen auswirken, die fehlerfrei sind.

Beim Laden von Daten auf eine partitionierte Datenbank können die folgenden Modi verwendet werden:

- `PARTITION_AND_LOAD`. Die Daten werden partitioniert (vielleicht in Parallelverarbeitung) und gleichzeitig auf die entsprechenden Datenbankpartitionen geladen.

- **PARTITION\_ONLY.** Die Daten werden partitioniert (vielleicht in Parallelverarbeitung), und die Ausgabe wird in Dateien an einer angegebenen Speicherposition auf jeder Ladepartition geschrieben. Jede Datei enthält Partitionskopfdaten, die angeben, wie die Daten partitioniert wurden, und dass die Datei unter Verwendung des Modus' `LOAD_ONLY` in die Datenbank geladen werden kann.
- **LOAD\_ONLY.** Es wird davon ausgegangen, dass die Daten bereits partitioniert sind. Der Partitionierungsprozess wird übersprungen, und die Daten werden gleichzeitig auf die entsprechenden Datenbankpartitionen geladen.
- **LOAD\_ONLY\_VERIFY\_PART.** Es wird davon ausgegangen, dass die Daten bereits partitioniert sind, aber die Datendatei enthält keine Partitionskopfdaten. Der Partitionierungsprozess wird übersprungen, und die Daten werden gleichzeitig auf die entsprechenden Datenbankpartitionen geladen. Während der Ladeoperation wird für jede Zeile geprüft, ob sie sich auf der korrekten Partition befindet. Zeilen, die ungültige Partitionierungen enthalten, werden in eine Speicherauszugsdatei gestellt, sofern der Änderungswert für den Dateityp *dumpfile* angegeben wurde. Andernfalls werden die Zeilen gelöscht. Wenn für eine bestimmte Ladepartition eine Partitionsungültigkeit vorliegt, wird für diese Partition eine Warnung in die `LOAD`-Nachrichtendatei geschrieben.
- **ANALYZE.** Es wird eine optimale Partitionierungszuordnung mit einer gleichmäßigen Datenverteilung auf alle Datenbankpartitionen generiert.

### Konzepte und Terminologie

Im Zusammenhang mit der Funktionsweise und der Ausführung des Dienstprogramms `LOAD` in einer Umgebung mit partitionierten Datenbanken wird die folgende Terminologie verwendet:

- Die *Koordinatorpartition* ist die Datenbankpartition, zu der der Benutzer eine Verbindung herstellt, um die Ladeoperation auszuführen. In den Modi `PARTITION_AND_LOAD`, `PARTITION_ONLY` und `ANALYZE` müssen sich die Datendateien auf dieser Partition befinden, sofern nicht die Option `CLIENT` des Befehls `LOAD` angegeben wird. Mit der Option `CLIENT` des Befehls `LOAD` wird angegeben, dass sich die zu ladenden Daten auf einem fernen verbundenen Client befinden.
- In den Modi `PARTITION_AND_LOAD`, `PARTITION_ONLY` und `ANALYZE` liest der *Agent für die Partitionierungsvorbereitung* die Benutzerdaten und verteilt sie im Round-Robin-Verfahren an die *Partitionierungsagenten*, die die Daten partitionieren. Dieser Prozess findet immer auf der Koordinatorpartition statt. Bei jeder Ladeoperation ist pro Partition maximal ein Partitionierungsagent zulässig.
- In den Modi `PARTITION_AND_LOAD`, `LOAD_ONLY` und `LOAD_ONLY_VERIFY_PART` werden auf jeder Ausgabepartition *Ladeagenten* ausgeführt. Sie koordinieren das Laden der Daten auf diese Partition.
- *Dateiladeagenten* werden während einer Ladeoperation im Modus `PARTITION_ONLY` auf jeder Ausgabepartition ausgeführt. Sie empfangen Daten von Partitionierungsagenten und schreiben sie in eine Datei auf ihrer Partition.
- Auf der Koordinatorpartition wird ein *Agent für Dateiübertragungsbefehle* ausgeführt. Er ist für die Ausführung eines Dateiübertragungsbefehls zuständig.

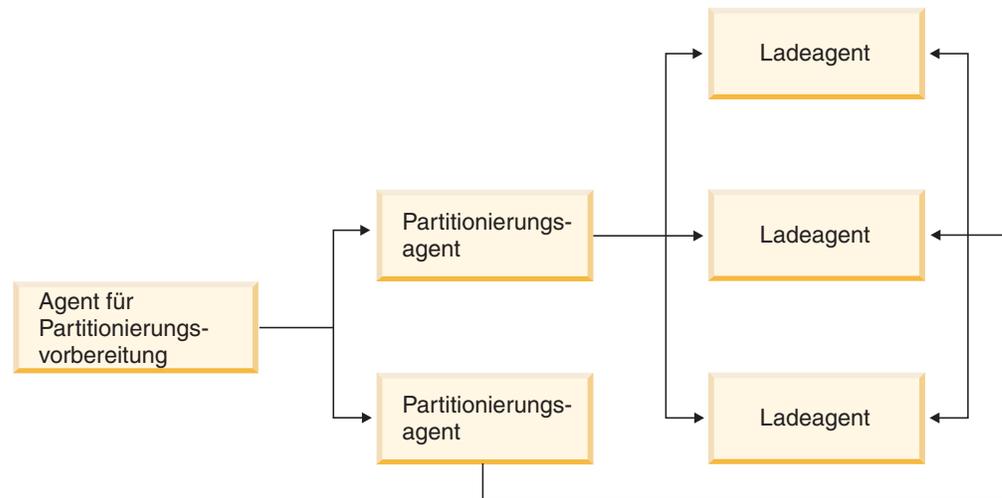


Abbildung 7. Übersicht über das Laden in partitionierte Datenbanken. Die Quelldaten werden durch den Agenten für die Partitionierungsvorbereitung gelesen. Ungefähr die Hälfte der Daten wird jeweils an zwei Partitionierungsagenten gesendet, die die Daten partitionieren und an eine der drei Datenbankpartitionen senden. Auf den einzelnen Partitionen werden die Daten durch den jeweiligen Ladeagenten geladen.

#### Zugehörige Konzepte:

- „Hinweise und Tipps für das Laden von Daten in einer Umgebung mit partitionierten Datenbanken“ auf Seite 224
- „Überwachen des Ladens in partitionierte Datenbanken mit LOAD QUERY“ auf Seite 209
- „Erneutes Starten oder Beenden einer Ladeoperation in einer Umgebung mit partitionierten Datenbanken“ auf Seite 211

#### Zugehörige Referenzen:

- „LOAD-Konfigurationsoptionen für partitionierte Datenbanken“ auf Seite 213

## Verwenden von LOAD in einer Umgebung mit partitionierten Datenbanken

#### Vorbedingungen:

Bevor Sie eine Tabelle in eine Umgebung mit partitionierten Datenbanken laden, sollten Sie die folgenden Punkte beachten:

1. Stellen Sie sicher, dass der Konfigurationsparameter *svcename* des Datenbankmanagers und die Profilregistrierdatenbank-Variable **DB2COMM** richtig definiert sind. Dies ist wichtig, weil das Dienstprogramm LOAD TCP/IP verwendet, um Daten von den Agenten für die Partitionierungsvorbereitung an die Partitionierungsagenten und von den Partitionierungsagenten an die Ladepartitionen zu übertragen.
2. Bevor Sie das Dienstprogramm LOAD ausführen, müssen Sie mit der Datenbank, in die die Daten geladen werden, verbunden bzw. in der Lage sein, implizit eine Verbindung zur Datenbank herzustellen. Da das Dienstprogramm LOAD eine Anweisung COMMIT absetzt, sollten Sie vor dem Beginn der Ladeoperation alle Transaktionen beenden und alle Sperren aufheben, indem Sie eine Anweisung COMMIT oder ROLLBACK absetzen.

Wird der Modus PARTITION\_AND\_LOAD, PARTITION\_ONLY oder ANALYZE verwendet, muss sich die geladene Datendatei auf dieser Partition befinden. Ausgenommen sind die folgenden Fälle:

- a. Die Option CLIENT wurde angegeben. In diesem Fall müssen sich die Daten auf der Clientmaschine befinden.
  - b. Der Eingabequellentyp ist CURSOR. Hierbei gibt es keine Eingabedatei.
3. Es kann von Nutzen sein, den Designadvisor auszuführen, um die jeweils beste Partition für die einzelnen Tabellen zu ermitteln. Weitere Informationen finden Sie unter Der Designadvisor.

### Einschränkungen:

Bei der Verwendung des Dienstprogramms LOAD zum Laden von Daten in einer Umgebung mit partitionierten Datenbanken gelten die folgenden Einschränkungen:

- Die Eingabedateien für die Ladeoperation dürfen nicht auf einer Bandeinheit gespeichert sein.
- Die Option ROWCOUNT wird nur dann unterstützt, wenn der Modus ANALYZE verwendet wird.
- Wenn die Zieltabelle über eine Identitätsspalte verfügt, die zur Partitionierung notwendig ist und der Änderungswert *identityoverride* nicht angegeben ist oder wenn Sie mehrere Datenbankpartitionen zur Partitionierung verwenden und anschließend die Daten laden, wird ein größerer Wert als 0 für SAVECOUNT im Befehl LOAD nicht unterstützt.
- Wenn eine Identitätsspalte einen Teil des Partitionierungsschlüssels darstellt, wird nur der Modus PARTITION\_AND\_LOAD unterstützt.
- Der Modus LOAD\_ONLY und der Modus LOAD\_ONLY\_VERIFY\_PART kann nicht mit der Option CLIENT des Befehls LOAD kombiniert werden.
- Der Modus LOAD\_ONLY\_VERIFY\_PART kann nicht mit dem Eingabequellentyp CURSOR verwendet werden.
- Die Werte LOAD\_ERRS\_ONLY und SETUP\_AND\_LOAD\_ERRS für den Fehlerisolationsmodus von Partitionen können nicht zusammen mit den Optionen ALLOW READ ACCESS und COPY YES des Befehls LOAD verwendet werden.
- Mehrere Ladeoperationen können gleichzeitig Daten in dieselbe Tabelle laden, wenn sich die durch die Optionen OUTPUT\_DBPARTNUMS und PARTITIONING\_DBPARTNUMS angegebenen Partitionen nicht überlappen. Beispiel: Wenn eine Tabelle auf den Partitionen 0 bis 3 definiert ist, kann eine Ladeoperation Daten auf die Partitionen 0 und 1 laden, während eine zweite Ladeoperation Daten auf die Partitionen 2 und 3 lädt.
- Nur ASC-Dateien (ASCII-Format mit universellen Zeilenbegrenzern) und DEL-Dateien (ASCII-Format ohne universelle Zeilenbegrenzer) können partitioniert werden. PC/IXF-Dateien können nicht partitioniert werden. Um eine PC/IXF-Datei in eine Mehrpartitionstabelle zu laden, können Sie sie zunächst in eine Einzelpartitionstabelle laden und dann eine Ladeoperation mit dem Dateityp CURSOR ausführen, um die Daten in eine Mehrpartitionstabelle zu versetzen. Sie können auch eine PC/IXF-Datei in eine Mehrpartitionstabelle laden, indem Sie LOAD im Modus LOAD\_ONLY\_VERIFY\_PART verwenden.

### Prozedur:

Die folgenden Beispiele veranschaulichen, wie Sie mit dem Befehl LOAD unterschiedliche Typen von Ladeoperationen einleiten können. Die in den folgenden Beispielen verwendete Datenbank enthält fünf Partitionen (0, 1, 2, 3 und 4). Jede Partition verfügt über ein lokales Verzeichnis namens /udb/data/. Die beiden Tabellen TABLE1 und TABLE2 sind auf den Partitionen 0, 1, 3 und 4 definiert. Beim Laden von einem Client aus greift der Benutzer auf einen fernen Client zu, bei dem es sich nicht um eine der Datenbankpartitionen handelt.

### Laden von einer Serverpartition

#### Beispiel für Partitionieren und Laden

In diesem Szenario besteht eine Verbindung zu einer Partition, auf der die Tabelle TABLE1 definiert sein kann oder auch nicht. Die Datendatei load.del befindet sich im aktuellen Arbeitsverzeichnis dieser Partition. Um die Daten aus der Datei load.del auf alle Partitionen zu laden, auf denen die Tabelle TABLE1 definiert ist, setzen Sie den folgenden Befehl ab: LOAD FROM LOAD.DEL of DEL REPLACE INTO TABLE1

**Anmerkung:** In diesem Beispiel werden für alle Konfigurationsparameter für die partitionierte Datenbank die Standardwerte verwendet: Der Parameter MODE verwendet den Standardwert PARTITION\_AND\_LOAD, die Optionen OUTPUT\_DBPARTNUMS nehmen alle Knoten als Standardwert an, auf denen TABLE1 definiert ist, und PARTITIONING\_DBPARTNUMS ist standardmäßig die Knotengruppe, die gemäß den Regeln des Befehls LOAD ausgewählt wurde, welche zur Auswahl von Partitionierungsknoten gelten, wenn keine Knoten angegeben sind.

Mit dem folgenden Befehl führen Sie eine Ladeoperation aus, die die Knoten 3 und 4 als Partitionierungsknoten verwendet:

```
LOAD FROM LOAD.DEL of DEL REPLACE INTO TABLE1
PARTITIONED DB CONFIG PARTITIONING_DBPARTNUMS (3,4)
```

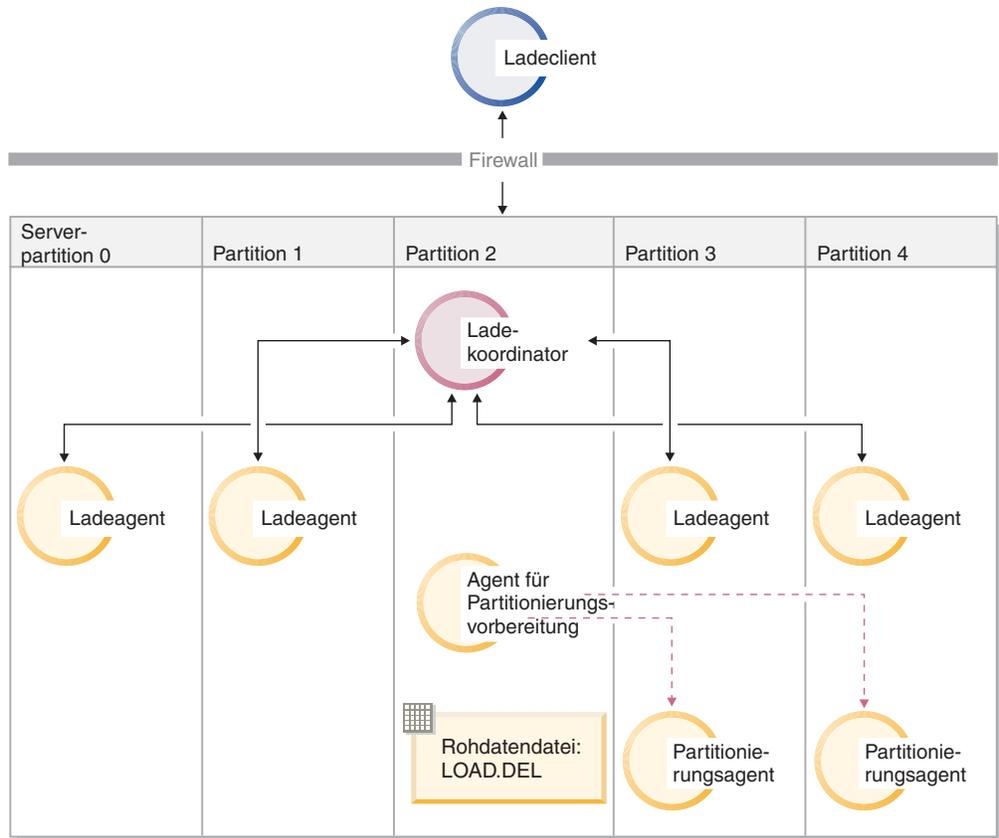


Abbildung 8. . Diese Abbildung veranschaulicht das Verhalten, das aus dem oben angegebenen Befehl resultiert. Die Daten werden in die Partitionen 3 und 4 geladen.

### Beispiel für Partitionieren ohne Laden

In diesem Szenario besteht eine Verbindung zu einer Partition, auf der die Tabelle TABLE1 definiert sein kann oder auch nicht. Die Datendatei load.del befindet sich im aktuellen Arbeitsverzeichnis dieser Partition. Um die Datei load.del auf allen Datenbankpartitionen, auf denen TABLE1 definiert ist, zu partitionieren (jedoch nicht zu laden) und die Partitionen 3 und 4 als Partitionierungsknoten zu verwenden, setzen Sie den folgenden Befehl ab:

```
LOAD FROM LOAD.DEL of DEL REPLACE INTO TABLE1
PARTITIONED DB CONFIG MODE PARTITION_ONLY
PART_FILE_LOCATION /udb/data
PARTITIONING_DBPARTNUMS (3,4)
```

Dieser Befehl bewirkt, dass eine Datei load.del.xxx im Verzeichnis /udb/data jeder Partition gespeichert wird. Hierbei steht xxx für eine dreistellige Darstellung der Partitionsnummer.

Um die Datei load.del auf den Partitionen 1 und 3 zu partitionieren und nur einen auf der Partition 0 aktiven Partitionierungsagenten zu verwenden (Standardwert für PARTITIONING\_DBPARTNUMS), verwenden Sie den folgenden Befehl:

```
LOAD FROM LOAD.DEL OF DEL REPLACE INTO TABLE1
PARTITIONED DB CONFIG MODE PARTITION_ONLY
PART_FILE_LOCATION /udb/data
OUTPUT_DBPARTNUMS (1,3)
```

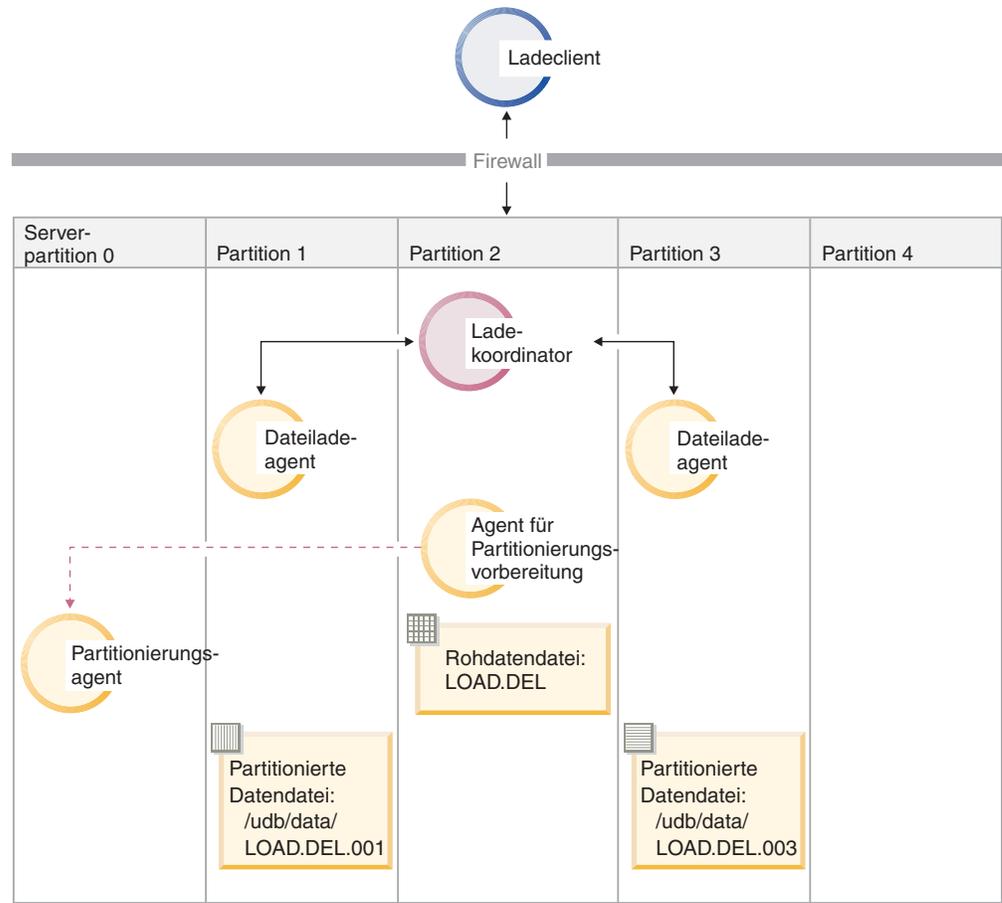


Abbildung 9. . Diese Abbildung veranschaulicht das Verhalten, das aus dem oben angegebenen Befehl resultiert. Die Daten werden auf die Partitionen 1 und 3 geladen, wobei ein auf Partition 0 aktiver Partitionierungsagent verwendet wird.

### Beispiel für Laden ohne Partitionieren

Wenn Sie bereits eine Ladeoperation im Modus PARTITION\_ONLY ausgeführt haben und die partitionierten Dateien im Verzeichnis /udb/data jeder Ladepartition auf alle Partitionen laden wollen, auf denen TABLE1 definiert ist, setzen Sie den folgenden Befehl ab:

```
LOAD FROM LOAD.DEL OF DEL REPLACE INTO TABLE1
PARTITIONED DB CONFIG MODE LOAD_ONLY
PART_FILE_LOCATION /udb/data
```

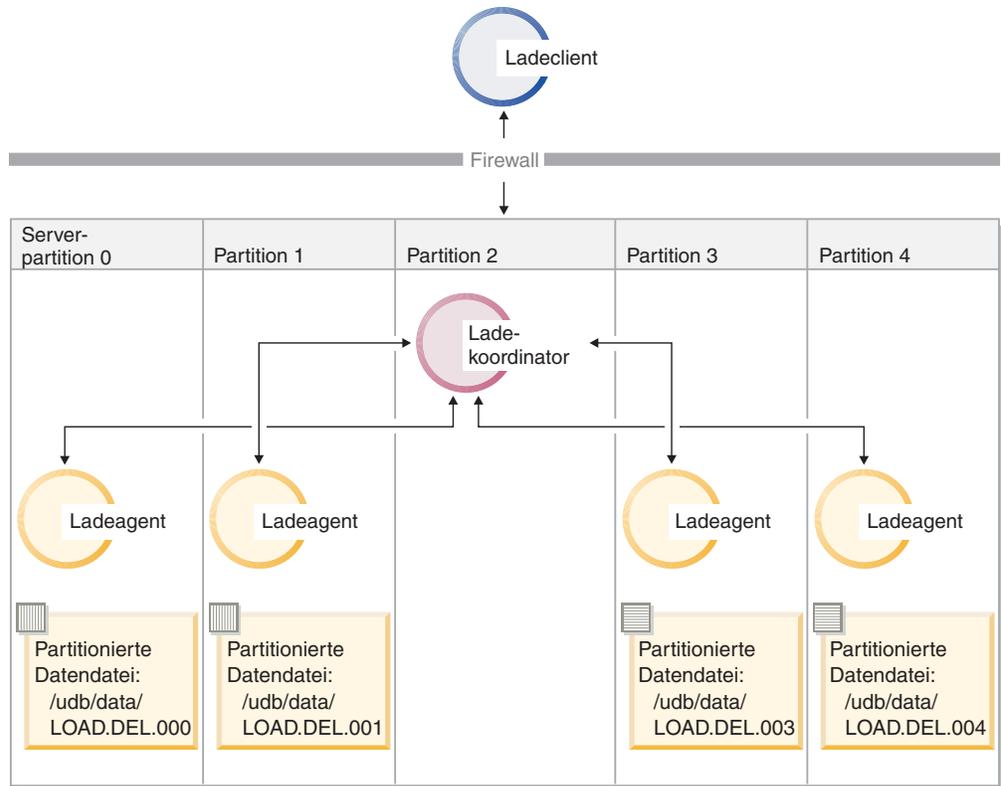


Abbildung 10. . Diese Abbildung veranschaulicht das Verhalten, das aus dem oben angegebenen Befehl resultiert. Die partitionierten Daten werden auf alle Partitionen geladen, auf denen TABLE1 definiert ist.

Mit dem folgenden Befehl werden Daten nur auf Partition 4 geladen:

```
LOAD FROM LOAD.DEL OF DEL REPLACE INTO TABLE1
PARTITIONED DB CONFIG MODE LOAD_ONLY
PART_FILE_LOCATION /udb/data
OUTPUT_DBPARTNUMS (4)
```

### Laden von vorpartitionierten Dateien ohne Kopfdaten für Partitionszuordnung

Mit dem Befehl LOAD können Sie Datendateien ohne Partitionskopfdaten direkt auf mehrere Datenbankpartitionen laden. Wenn die Datendateien im Verzeichnis /udb/data aller Partitionen, auf denen TABLE1 definiert ist, vorhanden sind und ihr Name load.del.xxx lautet (hierbei ist xxx die Partitionsnummer) können die Dateien mit dem folgenden Befehl geladen werden:

```
LOAD FROM LOAD.DEL OF DEL modified by dumpfile=rejected.rows
REPLACE INTO TABLE1
PARTITIONED DB CONFIG MODE LOAD_ONLY_VERIFY_PART
PART_FILE_LOCATION /udb/data
```

Mit dem folgenden Befehl werden die Daten nur auf Partition 1 geladen:

```
LOAD FROM LOAD.DEL OF DEL modified by dumpfile=rejected.rows
REPLACE INTO TABLE1
PARTITIONED DB CONFIG MODE LOAD_ONLY_VERIFY_PART
PART_FILE_LOCATION /udb/data
OUTPUT_DBPARTNUMS (1)
```

**Anmerkung:** Zeilen, die nicht zu der Partition gehören, von der aus sie geladen werden, werden zurückgewiesen und in die Speicherauszugsdatei gestellt, sofern eine Speicherauszugsdatei angegeben wurde.

## Laden von einem fernen Client in eine partitionierte Datenbank

Um Daten in eine partitionierte Datenbank aus einer Datei zu laden, die sich auf einem fernen Client befindet, müssen Sie mit der Option CLIENT des Befehls LOAD angeben, dass sich die Datendatei nicht auf einer Serverpartition befindet. Beispiel:

```
LOAD CLIENT FROM LOAD.DEL OF DEL REPLACE INTO TABLE1
```

**Anmerkung:** Der Modus LOAD\_ONLY oder LOAD\_ONLY\_VERIFY\_PART kann nicht zusammen mit der Option CLIENT verwendet werden.

## Laden über einen Cursor

Wie bei Einzelpartitionsdatenbanken können Sie über einen Cursor Daten in eine Mehrpartitionsdatenbank laden. In diesem Beispiel muss beim Modus PARTITION\_ONLY und LOAD\_ONLY die Option PART\_FILE\_LOCATION einen vollständig qualifizierten Dateinamen angeben. Dieser Name ist der vollständig qualifizierte Basisdateiname der partitionierten Dateien, die auf jeder Ausgabepartition erstellt oder geladen werden. Wenn die Zieldatenbank LOB-Spalten enthält, werden möglicherweise mehrere Dateien mit dem angegebenen Basisnamen erstellt.

Um alle Zeilen in der Antwortgruppe der Anweisung SELECT \* FROM TABLE1 in einer Datei auf jeder Partition namens /udb/data/select.out.xxx zu partitionieren (xxx ist die Knotennummer), damit sie später in die Tabelle TABLE2 geladen werden können, setzen Sie die folgenden Befehle über den DB2-Befehlszeilenprozessor ab:

```
DECLARE C1 CURSOR FOR SELECT * FROM TABLE1

LOAD FROM C1 OF CURSOR REPLACE INTO TABLE2
      PARTITIONED DB CONFIG MODE PARTITION_ONLY
      PART_FILE_LOCATION /udb/data/select.out
```

Die durch die vorstehende Operation erzeugten Datendateien können anschließend durch Absetzen des folgenden Befehls LOAD geladen werden:

```
LOAD FROM C1 OF CURSOR REPLACE INTO TABLE2
      PARTITIONED CB CONFIG MODE LOAD_ONLY
      PART_FILE_LOCATION /udb/data/select.out
```

### Zugehörige Konzepte:

- „Der Designadvisor“ in *Systemverwaltung: Optimierung*
- „Versetzen von Daten mit Dateityp CURSOR“ auf Seite 257

### Zugehörige Referenzen:

- „Lade-API "db2Load"“ auf Seite 139

---

## Überwachen des Ladens in partitionierte Datenbanken mit LOAD QUERY

### Während des Ladens in partitionierte Datenbanken erzeugte Nachrichtendateien

Während einer Ladeoperation werden von einigen Ladeprozessen Nachrichtendateien auf den Partitionen erstellt, auf denen sie ausgeführt werden. In diesen Dateien werden alle Informationen, Warnungen und Fehlermeldungen gespeichert, die während der Prozessausführung erzeugt wurden. Bei den Ladeprozessen, die

vom Benutzer anzeigbare Nachrichtendateien erzeugen, handelt es sich um den Ladeagenten, den Agenten für die Partitionierungsvorbereitung und den Partitionierungsagenten.

Benutzer können während einer Ladeoperation eine Verbindung zu einzelnen Partitionen herstellen und den Befehl LOAD QUERY für die Zieltabelle absetzen. Wird dieser Befehl über den Befehlszeilenprozessor abgesetzt, zeigt er den Inhalt aller Nachrichtendateien an, die sich gegenwärtig auf dieser Partition für die im Befehl LOAD QUERY angegebene Tabelle befinden.

Beispiel: Die Tabelle TABLE1 ist auf den Partitionen 0 bis 3 in der Datenbank WSDb definiert. Ein Benutzer ist mit Partition 0 verbunden und setzt den folgenden Befehl LOAD ab:

```
load from load.del of del replace into table1 partitioned db config
partitioning_dbpartnums (1)
```

Dieser Befehl leitet eine Ladeoperation ein, die das Ausführen von Ladeagenten auf den Partitionen 0, 1, 2 und 3 umfasst. Außerdem wird ein Partitionierungsagent auf Partition 1 und ein Agent für die Partitionierungsvorbereitung auf Partition 0 ausgeführt.

Partition 0 enthält eine Nachrichtendatei für den Agenten für die Partitionierungsvorbereitung und eine Datei für den Ladeagenten auf dieser Partition. Um den Inhalt dieser Dateien gleichzeitig anzuzeigen, starten Sie eine neue Sitzung, und setzen Sie die folgenden Befehle am Befehlszeilenprozessor ab:

```
set client connect_node 0
connect to wsdb
load query table table1
```

Partition 1 enthält eine Datei für den Ladeagenten und eine Datei für den Partitionierungsagenten. Um den Inhalt dieser Dateien anzuzeigen, starten Sie eine neue Sitzung, und setzen Sie die folgenden Befehle am Befehlszeilenprozessor ab:

```
set client connect_node 1
connect to wsdb
load query table table1
```

**Anmerkung:** Die durch die Konfigurationsoption STATUS\_INTERVAL generierten Nachrichten werden in der Nachrichtendatei für den Agenten für die Partitionierungsvorbereitung angezeigt. Um diese Nachrichten während einer Ladeoperation anzuzeigen, müssen Sie eine Verbindung zur Koordinatorpartition herstellen und den Befehl LOAD QUERY absetzen.

### Sichern des Inhalts von Nachrichtendateien

Wenn eine Ladeoperation über die Lade-API (**db2Load**) eingeleitet wird, muss die Nachrichtenoption (piLocalMsgFileName) angegeben werden. Dann werden die Nachrichtendateien vom Server an den Client übertragen und dort gespeichert, damit der Benutzer die Nachrichten anzeigen kann.

Bei Ladeoperationen für partitionierte Datenbanken, die über den Befehlszeilenprozessor eingeleitet werden, werden die Nachrichtendateien nicht auf der Konsole angezeigt oder beibehalten. Um den Inhalt dieser Dateien zu sichern oder anzuzeigen, nachdem das Laden in eine partitionierte Datenbank abgeschlossen wurde, muss die Option MESSAGES des Befehls LOAD angegeben werden. Bei Verwendung dieser Option werden die Nachrichtendateien auf jeder Partition nach dem

Abschluss der Ladeoperation an die Clientmaschine übertragen und dort in Dateien mit dem Basisnamen gespeichert, der mit der Option MESSAGES angegeben wurde. Die folgende Liste enthält die Dateinamen, die dem Ladeprozess, der die jeweilige Datei erstellt hat, entsprechen (bei Ladeoperationen für partitionierte Datenbanken):

Prozesstyp	Dateiname
Ladeagent	<nachrichtendateiname>.LOAD. <partitionsnummer>
Partitionierungsagent	<nachrichtendateiname>.PART. <partitionsnummer>
Agent für Partitionierungsvorbereitung	<nachrichtendateiname>.PREP. <partitionsnummer>

Wenn in der Option MESSAGES beispielsweise der Wert /wsdb/messages/load angegeben ist, lautet der Name der Nachrichtendatei für den Ladeagenten auf Partition 2 /wsdb/messages/load.LOAD.002.

**Anmerkung:** Es wird dringend empfohlen, die Option MESSAGES bei Ladeoperationen für partitionierte Datenbanken zu verwenden, die über den Befehlszeilenprozessor eingeleitet werden.

**Zugehörige Referenzen:**

- „Abfrage-API "db2LoadQuery" für den Status des Ladevorgangs" auf Seite 163

## Erneutes Starten oder Beenden einer Ladeoperation in einer Umgebung mit partitionierten Datenbanken

In einer Umgebung mit partitionierten Datenbanken besteht der Ladeprozess aus zwei Phasen: Während der SETUP-Phase werden Ressourcen auf Partitionesebene wie z. B. Tabellensperren für Ausgabepartitionen angefordert. In der LOAD-Phase werden die Daten formatiert und in die Tabellen auf den Partitionen geladen. Die vier Fehlerisolationsmodi für Partitionen (LOAD\_ERRS\_ONLY, SETUP\_ERRS\_ONLY, SETUP\_AND\_LOAD\_ERRS und NO\_ISOLATION) wirken sich auf das Verhalten der LOAD RESTART- und LOAD TERMINATE-Operationen aus, wenn während einer oder beiden Phasen Fehler auftreten. Wenn während der SETUP-Phase ein Fehler auftritt, sind in der Regel keine Neustart- oder Beendigungsoperationen erforderlich. Findet jedoch während der LOAD-Phase ein Fehler statt, muss eine LOAD RESTART- oder eine LOAD TERMINATE-Operation für alle von der Ladeoperation betroffenen Partitionen ausgeführt werden.

### Fehler in der SETUP-Phase

Wenn eine Ladeoperation während der SETUP-Phase auf mindestens einer Partition fehlschlägt und die Fehler der SETUP-Phase nicht isoliert werden (der Fehlerisolationsmodus also entweder LOAD\_ERRS\_ONLY oder NO\_ISOLATION lautet), wird die gesamte Ladeoperation abgebrochen, und der Status der Tabelle wird auf allen Partitionen auf den Status zurückgesetzt, der vor der Ladeoperation gültig war. In diesem Fall ist es nicht erforderlich, einen Befehl LOAD RESTART oder LOAD TERMINATE abzusetzen.

Wenn eine Ladeoperation während der einleitenden SETUP-Phase auf mindestens einer Partition fehlschlägt und die Fehler der SETUP-Phase isoliert werden (der

Fehlerisolationsmodus also entweder `SETUP_ERRS_ONLY` oder `SETUP_AND_LOAD_ERRS` lautet), wird die Ladeoperation auf den Partitionen fortgesetzt, auf denen die `SETUP`-Phase erfolgreich ausgeführt wurde. Auf Partitionen mit Fehlern wird der Status der Tabelle jedoch jeweils auf den Status zurückgesetzt, der vor der Ladeoperation gültig war. In diesem Fall muss nur dann eine `LOAD RESTART`- oder `LOAD TERMINATE`-Operation ausgeführt werden, wenn auch während der `LOAD`-Phase ein Fehler aufgetreten ist.

Um den Ladeprozess auf den Partitionen zu vervollständigen, auf denen die Ladeoperation während der `SETUP`-Phase fehlgeschlagen ist, setzen Sie einen Befehl `LOAD REPLACE` oder `LOAD INSERT` mit der Option `OUTPUT_DBPARTNUMS` ab, um nur die Partitionsnummern derjenigen Partitionen anzugeben, auf denen während der eigentlichen Ladeoperation Fehler auftraten.

Beispiel: Die Tabelle `TABLE1` ist auf den Partitionen 0 bis 3 in der Datenbank `WSDB` definiert. Der folgende Befehl wird abgesetzt:

```
load from load.del of del replace into table1 partitioned db config
  isolate_part_errs setup_and_load_errs
```

Während der `SETUP`-Phase der Ladeoperation tritt auf den Partitionen 1 und 3 ein Fehler auf. Da Fehler der `SETUP`-Phase isoliert werden, wird die Ladeoperation erfolgreich abgeschlossen, und auf die Partitionen 0 und 2 werden Daten geladen. Mit dem folgenden Befehl können Sie die Ladeoperation durch das Laden von Daten auf die Partitionen 1 und 3 vollständig abschließen:

```
load from load.del of del replace into table1 partitioned db config
  output_dbpartnums (1, 3)
```

### Fehler in der `LOAD`-Phase

Wenn eine Ladeoperation für eine partitionierte Datenbank während der `LOAD`-Phase auf mindestens einer Partition fehlschlägt, muss ein Befehl `LOAD RESTART` oder `LOAD TERMINATE` für alle von der Ladeoperation betroffenen Partitionen abgesetzt werden, und zwar unabhängig davon, ob auf diesen Partitionen beim Laden der Daten ein Fehler aufgetreten ist. Dies ist notwendig, weil das Laden von Daten in einer Umgebung mit partitionierten Datenbanken in einer einzigen Transaktion erfolgt. Wenn eine `LOAD RESTART`-Operation eingeleitet wird, wird das Laden auf allen Partitionen dort fortgesetzt, wo es beendet wurde.

Beispiel: Die Tabelle `TABLE1` ist auf den Partitionen 0 bis 3 in der Datenbank `WSDB` definiert. Der folgende Befehl wird abgesetzt:

```
load from load1.del of del replace into table1 partitioned db config
  isolate_part_errs no_isolation
```

Während der `LOAD`-Phase der Ladeoperation tritt auf den Partitionen 1 und 3 ein Fehler auf. Um die Ladeoperation wieder aufzunehmen, muss im Befehl `LOAD RESTART` dieselbe Gruppe von Ausgabepartitionen wie im ursprünglichen Befehl angegeben werden, da die Ladeoperation auf allen Partitionen erneut gestartet werden muss:

```
load from load.del of del restart into table1 partitioned db config
  isolate_part_errs no_isolation
```

**Anmerkung:** Bei `LOAD RESTART`-Operationen gelten die im Befehl `LOAD RESTART` angegebenen Optionen. Es ist daher wichtig, in diesem Befehl dieselben Optionen wie im ursprünglichen Befehl `LOAD` anzugeben.

Wenn ein Befehl `LOAD TERMINATE` nach einer Ladeoperation für eine Umgebung mit partitionierten Datenbanken verwendet wird, die während der `LOAD`-Phase fehlgeschlagen ist, gehen alle in der vorherigen Ladeoperation ausgeführten Aktionen verloren, und die Tabelle wird auf allen Partitionen auf den Status zurückgesetzt, der vor der ersten Ladeoperation gültig war.

Beispiel: Die Tabelle `TABLE1` ist auf den Partitionen 0 bis 3 in der Datenbank `WSDB` definiert. Der folgende Befehl wird abgesetzt:

```
load from load.del of del replace into table1 partitioned db config
isolate_part_errs no_isolation
```

Wenn während der `LOAD`-Phase ein Fehler auftritt, kann die Ladeoperation durch einen Befehl `LOAD TERMINATE` beendet werden, der dieselben Ausgabeparameter wie der Originalbefehl angibt:

```
load from load.del of del terminate into table1 partitioned db config
isolate_part_errs no_isolation
```

#### Zugehörige Konzepte:

- „Erneutes Starten einer unterbrochenen Ladeoperation“ auf Seite 109
- „Übersicht über das Laden in partitionierten Datenbanken“ auf Seite 201

---

## LOAD-Konfigurationsoptionen für partitionierte Datenbanken

### HOSTNAME X

Diese Option ist nur dann relevant, wenn sie zusammen mit der Option `FILE_TRANSFER_CMD` verwendet wird. `X` ist der Name der fernen Maschine, auf der sich die Datendatei befindet. Dies kann ein `z/OS`-Host oder eine andere Workstation sein. Wenn diese Option bei Verwendung der Option `FILE_TRANSFER_CMD` nicht angegeben wird, wird für den Hostnamen der Wert `nohost` verwendet.

### FILE\_TRANSFER\_CMD X

Diese Option gibt eine ausführbare Datei, eine Stapeldatei oder eine Prozedur an, die aufgerufen wird, bevor Daten auf Partitionen geladen werden. Der angegebene Wert muss ein vollständig qualifizierter Pfad sein. Der gesamte Pfad darf inklusive Programmdateiname nicht länger als 254 Zeichen sein. Der Befehl wird mit der folgenden Syntax aufgerufen:

```
<BEFEHL> <protokollpfad> <hostname>
<pipebasisdateiname> <anzahl_datenträger>
<quellendatenträgerliste>
```

Hierbei gilt:

#### <BEFEHL>

Der durch den Änderungswert `FILE:TRANSFER:CMD` angegebene Befehl.

#### <protokollpfad>

Der Protokollpfad für die Datei, aus der die Daten geladen werden. In diesen Pfad können diagnostische oder temporäre Daten geschrieben werden.

#### <hostname>

Der Wert der Option `HOSTNAME`.

#### <pipebasisdateiname>

Der Basisdateiname für benannte Pipes, die durch die Ladeoperation erstellt werden und von denen sie Daten erwartet. Für

jede Quelldatei im Befehl LOAD wird eine eigene Pipe erstellt. Jede dieser Dateien endet auf .xxx. Hierbei ist xxx der Index der Quelldatei für den Befehl LOAD. Gibt es beispielsweise 2 Quelldateien für den Befehl LOAD und lautet der Wert für <pipebasisdateiname> pipe123 würden zwei benannte Pipes namens pipe123.000 und pipe123.001 erstellt. Die mit <BEFEHL> angegebene Datei stellt Benutzerdaten in diese benannten Pipes.

**<anzahl\_datenträger>**

Gibt die Anzahl der nachfolgenden Datenträgerargumente an.

**<quellendatenträgerliste>**

Die Liste der Quelldateien, die im Befehl LOAD angegeben sind. Jede Quelldatei muss in doppelte Anführungszeichen gesetzt werden.

**PART\_FILE\_LOCATION X**

In den Modi PARTITION\_ONLY, LOAD\_ONLY und LOAD\_ONLY\_VERIFY\_PART kann mit diesem Parameter die Position der partitionierten Dateien angegeben werden. Diese Position muss auf jeder Partition vorhanden sein, die mit der Option OUTPUT\_DBPARTNUMS angegeben wurde. Handelt es sich bei der angegebenen Position um den Namen eines relativen Pfads, wird der Pfad an das aktuelle Verzeichnis angehängt, um die Position für die partitionierten Dateien zu erstellen.

Beim Dateityp CURSOR muss diese Option angegeben werden, und die Position muss sich auf einen vollständig qualifizierten Dateinamen beziehen. Dieser Name ist der vollständig qualifizierte Basisdateiname der partitionierten Dateien, die auf jeder Ausgabepartition erstellt werden (beim Modus PARTITION\_ONLY), bzw. die Position der Dateien, aus denen für jede Partition gelesen werden soll (beim Modus LOAD\_ONLY). Bei Verwendung des Modus PARTITION\_ONLY können mehrere Dateien mit dem angegebenen Basisnamen erstellt werden, wenn die Zieltabelle LOB-Spalten enthält.

Bei anderen Dateitypen als CURSOR wird das aktuelle Verzeichnis für die partitionierten Dateien verwendet, falls diese Option nicht angegeben ist.

**OUTPUT\_DBPARTNUMS X**

X ist eine Liste mit Partitionsnummern. Die Partitionsnummern geben die Datenbankpartitionen an, auf denen die Ladeoperation ausgeführt werden soll. Die Partitionsnummern müssen eine Untermenge der Datenbankpartitionen sein, auf denen die Tabelle definiert ist. In der Standardeinstellung sind alle Datenbankpartitionen ausgewählt. Die Liste muss in runde Klammern gesetzt werden. Die einzelnen Listeneinträge müssen mit einem Komma voneinander abgegrenzt werden. Die Angabe von Bereichen ist zulässig (Beispiel: (0, 2 to 10, 15)).

**PARTITIONING\_DBPARTNUMS X**

X ist eine Liste der Partitionsnummern, die im Partitionierungsprozess verwendet werden. Die Liste muss in runde Klammern gesetzt werden. Die einzelnen Listeneinträge müssen mit einem Komma voneinander abgegrenzt werden. Die Angabe von Bereichen ist zulässig (Beispiel: (0, 2 to 10, 15)). Die angegebenen Partitionierungsknoten können sich von den Datenbankpartitionen, die geladen werden, unterscheiden. Wenn diese Option nicht angegeben wird, ermittelt der Befehl LOAD, wie viele Partitionen erforderlich sind und welche Partitionen im Hinblick auf eine optimale Leistung verwendet werden.

Falls der Änderungswert ANYORDER im Befehl LOAD nicht angegeben wird, wird in der LOAD-Sitzung nur ein Partitionsagent verwendet. Des Weiteren wird - sofern nur eine Partition für die Option OUTPUT\_DBPARTNUMS angegeben wird oder der Koordinatorknoten der Ladeoperation kein Element von OUTPUT:DBPARTNUMS ist - der Koordinatorknoten der Ladeoperation als Partitionierungspartition verwendet. Andernfalls wird die erste in OUTPUT\_DBPARTNUMS angegebene Partition (nicht der Koordinatorknoten) als Partitionierungspartition verwendet.

Wenn der Änderungswert ANYORDER angegeben ist, wird die Anzahl der Partitionierungsknoten wie folgt ermittelt: (Anzahl der Partitionen in OUTPUT\_DBPARTNUMS) geteilt durch 4 plus 1. Anschließend wird die Anzahl der Partitionierungsknoten aus OUTPUT\_DBPARTNUMS ausgewählt, wobei die zum Laden der Daten verwendete Partition ausgenommen ist.

## MODE X

Gibt den Modus an, in dem die Ladeoperation beim Laden einer partitionierten Datenbank stattfindet. Der Standardwert lautet PARTITION\_AND\_LOAD. Gültige Werte:

- PARTITION\_AND\_LOAD. Die Daten werden partitioniert (möglicherweise in Parallelverarbeitung) und gleichzeitig auf die entsprechenden Datenbankpartitionen geladen.
- PARTITION\_ONLY. Die Daten werden partitioniert (möglicherweise in Parallelverarbeitung), und die Ausgabe wird in Dateien an einer angegebenen Speicherposition auf jeder Ladepartition geschrieben. Bei anderen Dateitypen als CURSOR lautet das Format des Namens der Ausgabe-datei in der betreffenden Partition `dateiname.xxx`, wobei `dateiname` der im Befehl LOAD angegebene Name der Eingabedatei und `xxx` die dreistellige Partitionsnummer ist. Beim Dateityp CURSOR wird der Name der Ausgabedatei auf jeder Partition durch die Option PART\_FILE\_LOCATION festgelegt. Der Abschnitt zur Option PART\_FILE\_LOCATION enthält weitere Informationen dazu, wie die Position der Partitionsdatei für die betreffende Partition angegeben wird.

### Anmerkungen:

1. Dieser Modus kann nicht für CLI-Ladeoperationen verwendet werden.
  2. Enthält die Tabelle eine Identitätsspalte, die für die Partitionierung erforderlich ist, dann wird dieser Modus nicht unterstützt sofern der Änderungswert *identityoverride* nicht angegeben wird.
- LOAD\_ONLY. Es wird davon ausgegangen, dass die Daten bereits partitioniert sind. Der Partitionierungsprozess wird übersprungen, und die Daten werden gleichzeitig auf die entsprechenden Datenbankpartitionen geladen. Bei anderen Dateitypen als CURSOR lautet das Format des Namens der Eingabedatei für die betreffende Partition `dateiname.xxx`, wobei `dateiname` der im Befehl LOAD angegebene Name der Datei und `xxx` die dreistellige Partitionsnummer ist. Beim Dateityp CURSOR wird der Name der Eingabedatei auf jeder Partition durch die Option PART\_FILE\_LOCATION festgelegt. Der Abschnitt zur Option PART\_FILE\_LOCATION enthält weitere Informationen dazu, wie die Position der Partitionsdatei für die betreffende Partition angegeben wird.

### Anmerkungen:

1. Dieser Modus kann nicht für CLI-Ladeoperationen verwendet werden und nicht, wenn die Option CLIENT des Befehls LOAD angegeben ist.

2. Enthält die Tabelle eine Identitätsspalte, die für die Partitionierung erforderlich ist, dann wird dieser Modus nicht unterstützt sofern der Änderungswert *identityoverride* nicht angegeben wird.

- **LOAD\_ONLY\_VERIFY\_PART.** Es wird davon ausgegangen, dass die Daten bereits partitioniert sind, aber die Datendatei enthält keine Partitionskopfdaten. Der Partitionierungsprozess wird übersprungen, und die Daten werden gleichzeitig auf die entsprechenden Datenbankpartitionen geladen. Während der Ladeoperation wird für jede Zeile geprüft, ob sie sich auf der korrekten Partition befindet. Zeilen, die ungültige Partitionierungen enthalten, werden in eine Speicherauszugsdatei gestellt, sofern der Änderungswert für den Dateityp *dumpfile* angegeben wurde. Andernfalls werden die Zeilen gelöscht. Wenn für eine bestimmte Ladepartition eine Partitionsungültigkeit vorliegt, wird für diese Partition eine Warnung in die LOAD-Nachrichtendatei geschrieben. Das Format des Namens der Eingabedatei für die betreffende Partition lautet *dateiname.xxx*, wobei *dateiname* der im Befehl LOAD angegebene Name der Datei und *xxx* die dreistellige Partitionsnummer ist. Der Abschnitt zur Option **PART\_FILE\_LOCATION** enthält weitere Informationen dazu, wie die Position der Partitionsdatei für die betreffende Partition angegeben wird.

#### **Anmerkungen:**

1. Dieser Modus kann nicht für CLI-Ladeoperationen verwendet werden und nicht, wenn die Option **CLIENT** des Befehls **LOAD** angegeben ist.
  2. Enthält die Tabelle eine Identitätsspalte, die für die Partitionierung erforderlich ist, dann wird dieser Modus nicht unterstützt sofern der Änderungswert *identityoverride* nicht angegeben wird.
- **ANALYZE.** Es wird eine optimale Partitionierungszuordnung mit einer gleichmäßigen Verteilung auf alle Datenbankpartitionen generiert.

#### **MAX\_NUM\_PART\_AGENTS X**

Gibt an, wie viele Partitionierungsagenten in einer LOAD-Sitzung maximal verwendet werden. Der Standardwert ist 25.

#### **ISOLATE\_PART\_ERRS X**

Gibt an, wie die Ladeoperation auf Fehler reagiert, die auf einzelnen Partitionen auftreten. Der Standardwert ist **LOAD\_ERRS\_ONLY**, sofern nicht sowohl die Option **ALLOW READ ACCESS** als auch **COPY YES** des Befehls **LOAD** angegeben ist. In diesem Fall ist der Standardwert **NO\_ISOLATION**. Gültige Werte:

- **SETUP\_ERRS\_ONLY.** Bei Fehlern, die während der Setupphase auf einer Partition auftreten (beispielsweise Fehler beim Zugriff auf eine Partition oder Probleme beim Zugriff auf einen Tabellenbereich bzw. auf eine Tabelle auf einer Partition), wird die Ladeoperation auf der Partition mit dem Fehler gestoppt, auf den übrigen Partitionen jedoch fortgesetzt. Fehler, die auf einer Partition auftreten, während Daten geladen werden, bewirken das Fehlschlagen der gesamten Operation und das Zurücksetzen in den letzten Konsistenzzustand auf jeder Partition.
- **LOAD\_ERRS\_ONLY.** Fehler, die während der Setupphase auf einer Partition auftreten, bewirken das Fehlschlagen der gesamten Ladeoperation. Wenn beim Laden der Daten ein Fehler auftritt, werden die Partitionen mit Fehlern auf ihren letzten Konsistenzzustand zurückgesetzt. Die Ladeoperation wird für die verbleibenden Partitionen fortgesetzt, bis ein Fehler auftritt oder bis alle Daten geladen wurden. Auf den Partitionen, auf denen alle Daten geladen wurden, sind die Daten nach der Lade-

operation nicht erkennbar. Aufgrund der Fehler auf den anderen Partitionen wird die Transaktion abgebrochen. Die Daten bleiben auf allen Partitionen nicht erkennbar, bis eine LOAD RESTART-Operation ausgeführt wird. Dies macht die neu geladenen Daten auf den Partitionen erkennbar, auf denen die Ladeoperation vollständig ausgeführt wurde. Auf Partitionen mit Fehlern wird die Ladeoperation hierdurch wieder aufgenommen.

**Anmerkung:** Dieser Modus kann nicht verwendet werden, wenn sowohl die Option ALLOW READ ACCESS als auch COPY YES des Befehls LOAD angegeben ist.

- **SETUP\_AND\_LOAD\_ERRS.** In diesem Modus führen Fehler auf Partitionsebene während der Setupphase oder dem Laden von Daten dazu, dass die Verarbeitung nur auf den betroffenen Partitionen gestoppt wird. Wie im Modus LOAD\_ERRS\_ONLY bleiben die Daten - falls beim Laden der Daten Partitionsfehler auftreten - auf allen Partitionen nicht erkennbar, bis eine LOAD RESTART-Operation ausgeführt wird.

**Anmerkung:** Dieser Modus kann nicht verwendet werden, wenn sowohl die Option ALLOW READ ACCESS als auch COPY YES des Befehls LOAD angegeben ist.

- **NO\_ISOLATION.** In diesem Modus bewirkt jeder Fehler während der Ladeoperation, dass die Transaktion fehlschlägt.

#### **STATUS\_INTERVAL X**

X gibt an, wie häufig der Benutzer eine Benachrichtigung über den Umfang der gelesenen Daten empfängt. Die Maßeinheit ist Megabyte (MB). Der Standardwert 100 MB. Gültige Werte sind ganze Zahlen von 1 bis 4000.

#### **PORT\_RANGE X**

X gibt den Bereich von TCP-Ports an, die zur Erstellung von Sockets für die interne Kommunikation verwendet werden. Der Standardbereich erstreckt sich von 6000 bis 6063. Wenn die DB2-Registrierdatenbank-Variable DB2ATLD\_PORTS zum Zeitpunkt des Aufrufs definiert ist, überschreibt ihr Wert den Wert, der für die LOAD-Konfigurationsoption PORT\_RANGE angegeben ist. Der Bereich für die Registrierdatenbank-Variable DB2ATLD\_PORTS sollte in dem folgenden Format angegeben werden:

```
<niedrige_portnummer>:<hohe_portnummer>
```

Das Format für den Befehlszeilenprozessor lautet:

```
( niedrige_portnummer, hohe_portnummer )
```

#### **CHECK\_TRUNCATION**

Gibt an, dass das Programm bei der Ein-/Ausgabe prüfen soll, ob Datensätze abgeschnitten werden. In der Standardeinstellung werden die Daten nicht auf ein Abschneiden bei der Ein-/Ausgabe überprüft.

#### **MAP\_FILE\_INPUT X**

X gibt den Namen der Eingabedatei für die Partitionierungszuordnung an. Dieser Parameter muss angegeben werden, wenn die Partitionierungszuordnung angepasst wurde, da er auf die Datei verweist, in der sich die angepasste Partitionierungszuordnung befindet. Eine angepasste Partitionierungszuordnung kann erstellt werden, indem die Zuordnung mit dem Programm **db2gpmmap** aus der Tabelle mit dem Datenbanksystemkatalog extrahiert wird oder indem mit dem Modus ANALYZE des Befehls LOAD eine optimale Zuordnung generiert wird. Die mit dem Modus ANALYZE

generierte Zuordnung muss auf alle Datenbankpartitionen in der Datenbank versetzt werden, bevor die Ladeoperation fortgesetzt werden kann.

#### **MAP\_FILE\_OUTPUT X**

X ist der Name der Ausgabedatei für die Partitionierungszuordnung. Dieser Parameter sollte verwendet werden, wenn der Modus ANALYZE angegeben wird. Es wird eine optimale Partitionierungszuordnung mit einer gleichmäßigen Verteilung auf alle Datenbankpartitionen generiert. Falls dieser Änderungswert verwendet wird und der Modus ANALYZE nicht angegeben ist, wird das Programm mit einem Fehler beendet.

#### **TRACE X**

Gibt die Anzahl der Datensätze an, für die ein Trace erstellt werden soll, wenn ein Speicherauszug des Datenkonvertierungsprozesses überprüft werden soll und eine Ausgabe der Hash-Werte erforderlich ist. Der Standardwert ist 0.

#### **NEWLINE**

Diese Option wird verwendet, wenn es sich bei der Eingabedatendatei um eine ASC-Datei handelt, in der jeder Datensatz durch ein Zeilenvorschubzeichen begrenzt wird, und der Parameter RecLen des Befehls LOAD angegeben wird. Wenn sie angegeben wird, wird jeder Datensatz darauf geprüft, ob er ein Zeilenvorschubzeichen enthält. Außerdem wird die im Parameter RecLen angegebene Satzlänge geprüft.

#### **DISTFILE X**

Wird diese Option angegeben, generiert das Dienstprogramm LOAD eine Partitionsverteilungsdatei mit dem angegebenen Namen. Die Partitionsverteilungsdatei enthält 4096 Integer: Ein Integer für jeden Eintrag in der Partitionszuordnung der Zieltabelle. Jedes Integer in der Datei stellt die Anzahl der Zeilen in den Eingabedateien, die geladen werden, für die zum entsprechenden Eintrag der Partitionierungszuordnung ein Hashverfahren ausgeführt wird. Diese Informationen können Ihnen dabei helfen, Abweichungen in Ihren Daten anzugeben und kann Ihnen dabei helfen zu entscheiden, ob eine neue Partitionierungszuordnung mit Hilfe des Modus ANALYZE des Dienstprogramms für die Tabelle generiert werden soll. Wenn diese Option nicht angegeben wird, ist es das Standardverhalten des Dienstprogramms LOAD, die Verteilungsdatei nicht zu generieren.

**Anmerkung:** Wenn diese Option angegeben wird, wird ein Maximum eines Partitionierungsagenten für die Ladeoperation verwendet. Werden vom Benutzer explizit mehrere Partitionierungsagenten angefordert, wird nur einer verwendet.

#### **OMIT\_HEADER**

Gibt an, dass die Partitionsdatei keine Kopfdaten für die Partitionszuordnung enthalten soll. Wenn diese Option nicht angegeben wird, werden Kopfdaten generiert.

#### **RUN\_STAT\_DBPARTNUM X**

Falls der Parameter STATISTICS YES im Befehl LOAD angegeben wurde, werden die Statistikdaten nur auf einer Datenbankpartition erfasst. Dieser Parameter gibt an, auf welcher Datenbankpartition Statistikdaten erfasst werden sollen. Wenn dieser Wert -1 lautet oder gar nicht angegeben wird, werden die Statistikdaten auf der ersten Datenbankpartition in der Ausgabepartitionsliste erfasst.

### Zugehörige Tasks:

- „Verwenden von LOAD in einer Umgebung mit partitionierten Datenbanken“ auf Seite 203

### Zugehörige Referenzen:

- „REDISTRIBUTE DATABASE PARTITION GROUP Command“ in *Command Reference*
- „Befehl LOAD“ auf Seite 112

---

## Beispiele für LOAD-Sitzungen für partitionierte Datenbanken

In den folgenden Beispielen enthält die Datenbank 4 Partitionen, die mit 0 bis 4 nummeriert sind. Die Datenbank WSDB ist auf allen Partitionen definiert. Die Tabelle TABLE1 befindet sich in der Standardknotengruppe, die ebenfalls auf allen Partitionen definiert ist.

### Beispiel 1

Um Daten aus der Benutzerdatendatei load.del, die sich auf Partition 0 befindet, in die Tabelle TABLE1 zu laden, stellen Sie eine Verbindung zu Partition 0 her, und setzen Sie anschließend den folgenden Befehl ab:

```
load from load.del of del replace into table1
```

Wenn die Ladeoperation erfolgreich durchgeführt wird, erhalten Sie die folgende Ausgabe:

Agent Typ	Knoten	SQL-Code	Ergebnis
LOAD	000	+00000000	Erfolg.
LOAD	001	+00000000	Erfolg.
LOAD	002	+00000000	Erfolg.
LOAD	003	+00000000	Erfolg.
PARTITION	001	+00000000	Erfolg.
PRE_PARTITION	000	+00000000	Erfolg.
ERGEBNISSE:	4 von 4 LOAD-Operationen wurden erfolgreich beendet.		

```
Zusammenfassung für Partitionierungsagenten
Gelesene Zeilen           = 100000
Zurückgewiesene Zeilen   = 0
Partitionierte Zeilen     = 100000
```

```
Zusammenfassung für LOAD-Agenten:
Anzahl gelesener Zeilen   = 100000
Anzahl übersprungener Zeilen = 0
Anzahl geladener Zeilen   = 100000
Anzahl zurückgewiesener Zeilen = 0
Anzahl gelöschter Zeilen  = 0
Anzahl festgeschriebener Zeilen = 100000
```

Die Ausgabe gibt an, dass sich auf jeder Partition ein Ladeagent befand, der jeweils erfolgreich ausgeführt wurde. Außerdem zeigt sie, dass auf der Koordinatorpartition ein Agent für die Partitionierungsvorbereitung ausgeführt wurde sowie ein Partitionierungsagent auf Partition 1. Diese Prozesse wurden erfolgreich mit

einem normalen SQL-Rückkehrcode 0 abgeschlossen. Die Zusammenfassung der Statistikdaten ergibt, dass der Agent für die Partitionierungsvorbereitung 100.000 Zeilen gelesen hat, dass der Partitionierungsagent 100.000 Zeilen partitioniert hat und dass durch die Ladeagenten insgesamt 100.000 Zeilen geladen wurden.

### Beispiel 2

Im folgenden Beispiel werden die Daten in die Tabelle TABLE1 geladen, wobei der Modus PARTITION\_ONLY verwendet wird. Die partitionierten Ausgabedateien werden auf der jeweiligen Ausgabepartition im Verzeichnis /udb/data gespeichert:

```
load from load.del of del replace into table1 partitioned db config mode
partition_only part_file_location /udb/data
```

Der Befehl LOAD hat die folgende Ausgabe:

Agent Typ	Knoten	SQL-Code	Ergebnis
LOAD_TO_FILE	000	+00000000	Erfolg.
LOAD_TO_FILE	001	+00000000	Erfolg.
LOAD_TO_FILE	002	+00000000	Erfolg.
LOAD_TO_FILE	003	+00000000	Erfolg.
PARTITION	001	+00000000	Erfolg.
PRE_PARTITION	000	+00000000	Erfolg.

```
Zusammenfassung für Partitionierungsagenten
Gelesene Zeilen           = 100000
Zurückgewiesene Zeilen    = 0
Partitionierte Zeilen     = 100000
```

Die Ausgabe macht deutlich, dass auf jedem Ausgabeknoten ein Dateiladeagent aktiv war und dass diese Agenten erfolgreich ausgeführt wurden. Auf der Koordinatorpartition wurde ein Agent für die Partitionierungsvorbereitung ausgeführt, und auf dem Knoten 1 wurde ein Partitionierungsagent ausgeführt. Die Zusammenfassung der Statistikdaten ergibt, dass der Agent für die Partitionierungsvorbereitung 100.000 Zeilen erfolgreich gelesen hat und dass 100.000 Zeilen erfolgreich vom Partitionierungsagenten partitioniert wurden. Da keine Zeilen in die Tabelle geladen wurden, wird keine Zusammenfassung über die Anzahl der geladenen Zeilen angezeigt.

### Beispiel 3

Mit dem folgenden Befehl können Sie die Dateien laden, die mit der vorstehenden Ladeoperation im Modus PARTITION\_ONLY generiert wurden:

```
load from load.del of del replace into table1 partitioned db config mode
load_only part_file_location /udb/data
```

Der Befehl LOAD hat die folgende Ausgabe:

Agent Typ	Knoten	SQL-Code	Ergebnis
LOAD	000	+00000000	Erfolg.
LOAD	001	+00000000	Erfolg.
LOAD	002	+00000000	Erfolg.

LOAD	003	+00000000	Erfolg.
ERGEBNISSE: 4 von 4 LOAD-Operationen wurden erfolgreich beendet.			

Zusammenfassung für LOAD-Agenten:

Anzahl gelesener Zeilen = 100000  
 Anzahl übersprungener Zeilen = 0  
 Anzahl geladener Zeilen = 100000  
 Anzahl zurückgewiesener Zeilen = 0  
 Anzahl gelöschter Zeilen = 0  
 Anzahl festgeschriebener Zeilen = 100000

Die Ausgabe besagt, dass die Ladeagenten auf allen Ausgabepartitionen erfolgreich ausgeführt wurden und dass insgesamt 100.000 Zeilen durch alle Ladeagenten geladen wurden. Die Zusammenfassung enthält keine Angaben zu partitionierten Zeilen, da keine Partitionierung erfolgte.

#### Beispiel 4 - Fehlgeschlagene Ladeoperation

Beim Absetzen des LOAD-Befehls

```
load from load.del of del replace into table1
```

wird die folgende Ausgabe zurückgegeben, wenn während der Ladeoperation auf einer der Ladepartitionen nicht genügend Speicherbereich im Tabellenbereich verfügbar ist:

```
SQL0289N Dem Tabellenbereich "DMS4KT" konnten keine neuen Seiten
zugeordnet werden.
SQLSTATE=57011
```

Agent Typ	Knoten	SQL-Code	Ergebnis
LOAD	000	+00000000	Erfolg.
LOAD	001	-00000289	Fehler. Möglicherweise Neustart (RESTART) erforderlich.
LOAD	002	+00000000	Erfolg.
LOAD	003	+00000000	Erfolg.
PARTITION	001	+00000000	Erfolg.
PRE_PARTITION	000	+00000000	Erfolg.
ERGEBNISSE: 3 von 4 LOAD-Operationen wurden erfolgreich beendet.			

Zusammenfassung für Partitionierungsagenten

Gelesene Zeilen = 0  
 Zurückgewiesene Zeilen = 0  
 Partitionierte Zeilen = 0

Zusammenfassung für LOAD-Agenten:

Anzahl gelesener Zeilen = 0  
 Anzahl übersprungener Zeilen = 0  
 Anzahl geladener Zeilen = 0  
 Anzahl zurückgewiesener Zeilen = 0  
 Anzahl gelöschter Zeilen = 0  
 Anzahl festgeschriebener Zeilen = 0

Die Ausgabe gibt an, dass die Ladeoperation den Fehler SQL0289 zurückgegeben hat. Die Partitionszusammenfassung besagt, dass auf der Partition 1 nicht genügend Speicherbereich vorhanden war. Da der Standardwert für den Fehlerisolationsmodus NO\_ISOLATION lautet, wird die Ladeoperation auf allen Partitionen abgebrochen, und es muss eine LOAD RESTART- oder LOAD TERMINATE-Operation aufgerufen werden. Falls zusätzlicher Speicherbereich zu den Behältern des Tabellenbereichs auf Partition 1 hinzugefügt wird, können Sie die Ladeoperation wie folgt erneut starten:

```
load from load.del of del restart into table1
```

#### Zugehörige Tasks:

- „Verwenden von LOAD in einer Umgebung mit partitionierten Datenbanken“ auf Seite 203

#### Zugehörige Referenzen:

- „Befehl LOAD“ auf Seite 112

---

## Migration und Kompatibilität zu früheren Versionen

### Verwenden des Befehls LOAD in einer Umgebung mit partitionierten Datenbanken

In Version 8 wurden am Befehl LOAD Änderungen vorgenommen, die die Funktionalität des Dienstprogramms **db2atld** ersetzen. Infolgedessen werden Sie einige Unterschiede hinsichtlich der verwendeten Terminologie feststellen.

Im Zusammenhang mit dem Dienstprogramm **db2atld** wurde von einem *Verteilerprozess* und einer *Teildatei* (= Ausgabe des Verteilerprozesses) gesprochen. Die entsprechenden Begriffe bei der Verwendung des Befehls LOAD sind *Partitionierungsgang* und *Partitionsdatei*. Außerdem wird bei Verwendung des Befehls LOAD davon gesprochen, dass eine Datendatei *partitioniert* und **nicht geteilt** wird. Die Moduswerte und Konfigurationsoptionen des Befehls LOAD wurden entsprechend umbenannt, um diese Änderung wiederzugeben. Beispielsweise entsprechen dem Modus SPLIT\_ONLY und der Option SPLIT\_NODES von Autoloader (Dienstprogramm für automatisches Laden) jetzt der Modus PARTITION\_ONLY und die Option PARTITIONING\_DBPARTNUMS des Befehls LOAD.

Die Entsprechung für die Autoloader-Option FORCE ist die Konfigurationsoption ISOLATE\_PART\_ERRS des Befehls LOAD. Mit Hilfe dieser Option können Benutzer vier Typen von Partitionsfehlerisolationen angeben. Einer dieser Typen (SETUP\_AND\_LOAD\_ERRS) hat Ähnlichkeit mit der Fehlerisolation, die durch die Option FORCE YES geboten wurde.

### Verwenden von db2atld

Das Dienstprogramm **db2atld** ist weiterhin verfügbar. Es akzeptiert eine Konfigurationsdatei mit derselben Syntax wie vor Version 8. Bei der Verwendung von **db2atld** müssen Sie Folgendes beachten:

1. **db2atld** stellt jetzt nur eine Verbindung her, wenn Daten in Mehrpartitionsdatenbanken geladen werden. Vor Version 8 stellte das Dienstprogramm **db2atld** eine Verbindung pro Ausgabepartition her.
2. Beim Einsatz in einer Umgebung mit partitionierten Datenbanken verwendet **db2atld** das gleiche Einzeltransaktionsmodell wie das Dienstprogramm LOAD.

3. Der im erweiterten Dienstprogramm LOAD gebotene Modus LOAD\_ONLY\_VERIFY\_PART und andere neue Funktionen sind nicht verfügbar, wenn ein Ladevorgang für eine partitionierte Datenbank über die ausführbare Datei **db2atld** eingeleitet wird.
4. Alle zuvor unterstützten Konfigurationsparameter von Autoloader (Dienstprogramm für automatisches Laden) werden erkannt.
5. In der Konsole werden jetzt etwas andere Informationen durch **db2atld** angezeigt. Beispielsweise werden die Partitionierungsstatistik und die Ladestatistik im Modus SPLIT\_AND\_LOAD jetzt immer angezeigt. Früher wurden diese Statistiken nur dann angezeigt, wenn eine Diskrepanz zwischen den Statistikdaten vorlag. Des Weiteren werden Statusintervalldaten während einer Ladeoperation jetzt nicht als Speicherauszug in der Konsole ausgegeben, sondern stattdessen als Speicherauszug in die Nachrichtendatei des Agenten für die Partitionierungsvorbereitung gestellt. Der Name dieser Datei lautet <nachrichtendatei>.PREP.<knotennummer>. Statusinformationen über die Ergebnisse der Verteilerprozesse und der Ladeprozesse sind jetzt nicht mehr verfügbar.
6. Die ausführbare Datei **db2atld** muss künftig nicht mehr über einen Pfad gestartet werden, der partitionsübergreifend angehängt ist.
7. Beim Laden von Daten in eine Umgebung mit partitionierten Datenbanken ist 999 die maximale Anzahl an Eingabequellen.
8. Vor Version 8: Bei Verwendung des Modus LOAD\_ONLY des Dienstprogramms **db2atld** konnten die Namen der Teildateien der jeweiligen Partition die Erweiterung .00xxx oder .xxx haben, wobei xxx die Partitionsnummer ist. Version 8 oder höher: Die Erweiterung .00xxx wird nicht mehr unterstützt. Sie können lediglich dreistellige Erweiterungen (.xxx) angeben.

### Zurücksetzen auf das Verhalten von LOAD vor Version 8 mit der DB2®-Registrierdatenbank-Variablen PARTITIONEDLOAD\_DEFAULT

Vor Version 8 konnten Sie mit dem Dienstprogramm LOAD eine einzelne Partition einer Mehrpartitionsdatenbank laden, sofern die Eingabedatendatei Partitionskopfdaten enthielt, die für die geladene Partition gültig waren. Diese Kopfdaten wurden normalerweise im Modus SPLIT\_ONLY von Autoloader (Dienstprogramm für automatisches Laden) oder über das Dienstprogramm **db2split** erstellt. Die Partitionskopfdaten wurden durch das Dienstprogramm LOAD geprüft, bevor die übrigen Daten geladen wurden. Bei Tabellen, die in einer EEE-Umgebung zu einer Partitionsgruppe einer Einzelpartitionsdatenbank gehören, konnte eine Datendatei ohne Partitionskopfdaten geladen werden, sofern der Änderungswert für den Dateityp noheader angegeben war.

Das Verhalten des Befehls LOAD hat sich seither geändert. In einer Umgebung mit partitionierten Datenbanken wird jetzt davon ausgegangen, dass die Ladeoperation für alle Partitionen, in denen die Tabelle definiert ist, stattfindet, wenn keine Konfigurationsoptionen für das Laden in partitionierte Datenbanken angegeben werden. Die Eingabedatei muss keine Partitionskopfdaten enthalten, und die Option MODE ist standardmäßig auf PARTITION\_AND\_LOAD gesetzt. Um das Laden für eine Einzelpartition auszuführen, muss die Option OUTPUT\_DBPARTNUMS angegeben werden.

Es besteht die Möglichkeit, das vor Version 8 geltende Verhalten des Befehls LOAD in einer Umgebung mit partitionierten Datenbanken zu erhalten. Auf diese Weise könnten Sie eine Datei mit gültigen Partitionskopfdaten in eine einzige Datenbankpartition laden, ohne hierbei zusätzliche Konfigurationsoptionen für partitionierte

Datenbanken anzugeben. Dies erreichen Sie, indem Sie den Wert der Registrierdatenbankvariablen `DB2_PARTITIONEDLOAD_DEFAULT` auf `NO` setzen. Die Verwendung dieser Option ist beispielsweise dann sinnvoll, wenn Sie Änderungen an vorhandenen Prozeduren vermeiden möchten, die den Befehl `LOAD` für einzelne Datenbankpartitionen absetzen. Um beispielsweise eine partitionierte Datei in die Partition 3 einer Tabelle zu laden, die zu einer Knotengruppe mit 4 Partitionen gehört, setzen Sie den folgenden Befehl ab:

```
db2set DB2_PARTITIONEDLOAD_DEFAULT=NO
```

Anschließend setzen Sie die folgenden Befehle über den DB2-Befehlszeilenprozessor (CLP) ab:

```
CONNECT RESET
```

```
SET CLIENT CONNECT_NODE 3
```

```
CONNECT TO DB MYDB
```

```
LOAD FROM LOAD.DEL OF DEL REPLACE INTO TABLE1
```

#### Zugehörige Referenzen:

- „Befehl `LOAD`“ auf Seite 112

---

## Hinweise und Tipps für das Laden von Daten in einer Umgebung mit partitionierten Datenbanken

Bevor Sie eine Tabelle in eine Umgebung mit partitionierten Datenbanken laden, sollten Sie die folgenden Punkte beachten:

- Machen Sie sich mit den Konfigurationsoptionen der Funktion für das Laden von partitionierten Daten vertraut, indem Sie das Dienstprogramm mit kleinen Datenmengen verwenden.
- Wenn die Eingabedaten bereits sortiert sind oder in einer bestimmten Reihenfolge vorliegen und Sie diese Reihenfolge während des Ladevorgangs beibehalten wollen, darf nur eine einzige Datenbankpartition zum Partitionieren verwendet werden. Durch paralleles Partitionieren kann nicht sichergestellt werden, dass die Daten in derselben Reihenfolge geladen werden, in der sie empfangen wurden. Das Dienstprogramm `LOAD` wählt standardmäßig einen einzigen Partitionierungsagenten aus, wenn der Änderungswert `anyorder` im Befehl `LOAD` nicht angegeben wird.
- Wenn große Objekte (LOBs) aus separaten Dateien geladen werden (wenn Sie also den Änderungswert `lobsinfile` über das Dienstprogramm `LOAD` verwenden), müssen alle Datenbankpartitionen, auf denen Ladevorgänge stattfinden, Lesezugriff auf alle Verzeichnisse mit LOB-Dateien haben. Der `LOAD`-Parameter *lob-pfad* muss beim Arbeiten mit LOBs vollständig qualifiziert sein.
- Sie können die Fortsetzung eines Jobs für partitionierte Datenbanken erzwingen, selbst wenn die Ladeoperation (beim Start) feststellt, dass sich einige Ladepartitionen oder zugeordnete Tabellenbereiche oder Tabellen im Offlinemodus befinden. Hierzu setzen Sie die Option `ISOLATE_PART_ERRS` auf `SETUP_ERRS_ONLY` oder `SETUP_AND_LOAD_ERRS`.
- Mit der Konfigurationsoption `STATUS_INTERVAL` der Funktion für das Laden von partitionierten Daten kann der Status eines Ladejobs für partitionierte Datenbanken überwacht werden. Die Ladeoperation gibt in den angegebenen Intervallen Nachrichten zurück, aus denen ersichtlich ist, wie viele Megabyte Daten durch den Agenten für die Partitionierungsvorbereitung gelesen wurden. Diese Nachrichten werden in der Nachrichtendatei des Agenten für die Partitio-

nierungsvorbereitung gespeichert. Um den Inhalt dieser Datei während der Ladeoperation anzuzeigen, stellen Sie eine Verbindung zur Koordinatorpartition her, und setzen Sie einen Befehl LOAD QUERY für die Zieltabelle ab.

- Sie können mit besserer Leistung rechnen, wenn sich die Partitionierungsknoten (durch die Option PARTITIONING\_DBPARTNUMS definiert) von den Ladepartitionen (durch die Option OUTPUT\_DBPARTNUMS definiert) unterscheiden, da es weniger Konkurrenzsituationen für CPU-Zyklen gibt. Wenn Daten in eine partitionierte Datenbank geladen werden, sollte das Dienstprogramm LOAD selbst auf einer Datenbankpartition aufgerufen werden, die nicht an der Partitionierungs- oder Ladeoperation beteiligt ist.
- Bei Angabe des Parameters MESSAGES im Befehl LOAD werden die Nachrichtendateien der Agenten für die Partitionierungsvorbereitung, der Partitionierungs- und der Ladeagenten am Ende der Ladeoperation zu Referenzzwecken gespeichert. Um den Inhalt dieser Dateien während einer Ladeoperation anzuzeigen, stellen Sie eine Verbindung zur gewünschten Partition her, und setzen Sie einen Befehl LOAD QUERY für die Zieltabelle ab.
- Das Dienstprogramm LOAD wählt nur eine Ausgabedatenbankpartition für die Erfassung von Statistikdaten aus. Mit der Konfigurationsoption RUN\_STAT\_DBPARTNUM der Funktion für das Laden in partitionierte Datenbanken können Sie diese Partition angeben.
- Vor dem Laden von Daten in einer Umgebung mit partitionierten Datenbanken kann es von Nutzen sein, den Designadvisor auszuführen, um die jeweils beste Partition für die einzelnen Tabellen zu ermitteln. Weitere Informationen hierzu finden Sie in Der Designadvisor.

## Fehlerbehebung

Wenn das Dienstprogramm LOAD blockiert, haben Sie folgende Möglichkeiten:

- Mit dem Parameter STATUS\_INTERVAL kann der Status einer Ladeoperation für partitionierte Datenbanken überwacht werden. Die Statusintervalldaten werden in der Nachrichtendatei des Agenten für die Partitionierungsvorbereitung auf der Koordinatorpartition gespeichert.
- Überprüfen Sie in der Nachrichtendatei des Partitionierungsagenten den Status der Prozesse des Partitionierungsagenten für die einzelnen Partitionierungen von Datenbankpartitionen. Wenn das Laden ohne Fehler fortgesetzt wird und die Option TRACE definiert wurde, sollten diese Nachrichtendateien Tracenachrichten für eine Reihe von Datensätzen enthalten.
- Überprüfen Sie, ob die LOAD-Nachrichtendatei Fehlernachrichten enthält.

**Anmerkung:** Damit diese Dateien vorhanden sind, müssen Sie die Option MESSAGES des Befehls LOAD angeben.

- Unterbrechen Sie die aktuelle Ladeoperation, wenn Sie Fehler finden, die darauf schließen lassen, dass bei einem der Ladeprozesse Fehler aufgetreten sind.

## Zugehörige Referenzen:

- „LOAD QUERY“ auf Seite 137
- „Befehl LOAD“ auf Seite 112



---

## Kapitel 5. Versetzen von DB2 Data Links Manager-Daten

In diesem Kapitel wird beschrieben, wie die DB2-Dienstprogramme EXPORT, IMPORT und LOAD verwendet werden, um DB2 Data Links Manager-Daten zu versetzen.

Informationen zu den Dateiformaten, die Sie mit diesen Dienstprogrammen verwenden können, finden Sie unter „Dateiformate der Dienstprogramme EXPORT, IMPORT und LOAD“ auf Seite 275.

Die folgenden Themen werden behandelt:

- „Versetzen von DB2 Data Links Manager-Daten mit EXPORT - Konzepte“
- „Versetzen von DB2 Data Links Manager-Daten mit EXPORT“ auf Seite 230
- „Versetzen von DB2 Data Links Manager-Daten mit IMPORT“ auf Seite 231
- „Versetzen von DB2 Data Links Manager-Daten mit LOAD“ auf Seite 232.

---

### Versetzen von DB2 Data Links Manager-Daten mit EXPORT - Konzepte

Da sich die Tabellendaten in der Datenbank und die Dateien, auf die DATALINK-Spalten verweisen, auf Data Links-Servern befinden, muss das Dienstprogramm EXPORT sowohl die Datenbankdaten als auch die Datendateien auf den entsprechenden Data Links-Servern versetzen. Dazu erstellt das Dienstprogramm EXPORT eine Steuerdatei für jeden Data Links-Server. Der Name der erstellten Steuerdatei entspricht dem Namen des Data Links-Servers. Die Steuerdateien werden in einem neu erstellten Verzeichnis namens `d1fm/JJJJMMTT/SSMMSS` erstellt. Dabei steht `JJJJMMTT` für *JahrMonatTag* und `SSMMSS` für *StundeMinuteSekunde*. Dieses Verzeichnis wird unter dem gleichen Verzeichnis erstellt, in dem auch die EXPORT-Datendatei erstellt wird. Eine Steuerdatei listet die Namen der entsprechenden DB2 Data Links Manager-Dateien auf, auf die die zu exportierenden DATALINK-Spaltenzeilen verweisen.

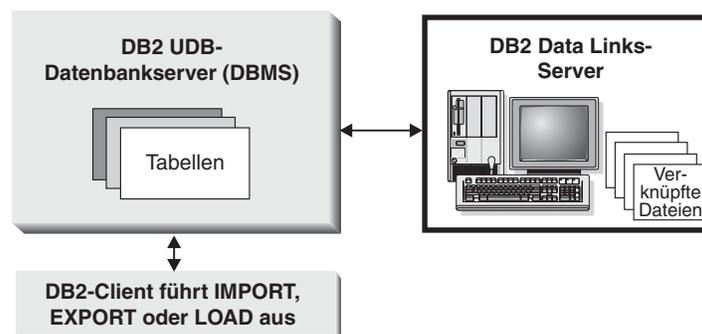


Abbildung 11. Versetzen von DB2 Data Links Manager-Daten. Da sich die Tabellendaten in der Datenbank und die Dateien, auf die DATALINK-Spalten verweisen, auf Data Links-Servern befinden, müssen die Dienstprogramme EXPORT, IMPORT und LOAD sowohl die Datenbankdaten als auch die Datendateien auf den entsprechenden Data Links-Servern versetzen.

Bei Windows<sup>®</sup>-Betriebssystemen erstellt das Dienstprogramm EXPORT nur eine einzige Steuerdatei für alle Data Links-Server. Der Name dieser Steuerdatei lautet `ctrlfile.lst`. Sie wird in einem neu erstellen Verzeichnis namens

`d1fm\JJJJMMTT\SSMMSS` erstellt. Dieses Verzeichnis wird unter dem gleichen Verzeichnis erstellt, in dem auch die EXPORT-Datendatei erstellt wird. Die Steuerdatei listet die URL-Adressen aller DB2 Data Links Manager-Dateien auf, auf die in den DATALINK-Spalten der exportierten Zeilen verwiesen wird.

DATALINK-Werte mit dem Merkmal NO LINK CONTROL werden nicht in die Steuerdatei aufgenommen.

Die Steuerdateien müssen auf die jeweiligen Data Links-Server übertragen werden. Bei Windows-Betriebssystemen muss die Steuerdatei auf alle verwendeten Data Links-Server übertragen werden. Das Dienstprogramm `d1fm_export` muss unter Angabe des Steuerdateinamens auf jedem Data Links-Server ausgeführt werden. Dieses Dienstprogramm erstellt ein Archiv der Dateien, die in der Steuerdatei für den betreffenden Data Links-Server aufgelistet sind. Legen Sie die Registrierdatenbank-Variable `DLFM_FS_ENVIRONMENT` entsprechend fest, bevor Sie das Dienstprogramm `d1fm_export` ausführen.

Das Dienstprogramm EXPORT wird als SQL-Anwendung ausgeführt. Die Zeilen und Spalten, die die Bedingungen der SELECT-Anweisung erfüllen, werden aus der Datenbank extrahiert. Bei DATALINK-Spalten darf in der SELECT-Anweisung keine Skalarfunktion angegeben werden.

Vom Dienstprogramm EXPORT werden die folgenden Dateien generiert:

- Die Exportdatendatei. Ein DATALINK-Spaltenwert in dieser Datei hat das gleiche Format wie das von den Dienstprogrammen LOAD und IMPORT verwendete. Wenn der DATALINK-Spaltenwert NULL ist, wird er auf die gleiche Art behandelt wie für andere NULL-Spalten.
- Steuerdateien für jeden Data Links-Server. Die Steuerdatei listet den vollständigen Pfad und die Namen aller Dateien auf, die von diesem Data Links-Server exportiert werden sollen. Bei Windows-Betriebssystemen gibt es nur eine einzige Steuerdatei für alle Data Links-Server, auf die von DATALINK-Spaltenwerten verwiesen wird.

Exportieren Sie die Dateien wie folgt mit dem Dienstprogramm `d1fm_export` von einem oder mehreren Data Links-Servern:

```
d1fm_export steuerdateiname archivdateiname
```

Dabei ist *steuerdateiname* der Name der durch Ausführen des Dienstprogramms EXPORT auf dem DB2®-Client generierten Steuerdatei, und *archivdateiname* ist der Name der Archivdatei, die generiert wird. Der Standardwert für *archivdateiname* ist `export.tar` im aktuellen Arbeitsverzeichnis.

Mit dem bereitgestellten Dienstprogramm `d1fm_import` können Dateien aus dem von `d1fm_export` generierten Archiv abgerufen und wiederhergestellt werden. Dieses Dienstprogramm muss verwendet werden, und zwar unabhängig davon, ob die Archivdateien auf dem gleichen oder auf einem anderen Data Links-Server wiederhergestellt werden.

Mit dem Dienstprogramm `d1fm_import` können Sie wie folgt Dateien aus dem Archiv abrufen:

```
d1fm_import archivdateiname [LISTFILES]
```

Dabei ist *archivdateiname* der Name der Archivdatei, die zum Wiederherstellen der Dateien verwendet wird. Der Standardwert für *archivdateiname* ist `export.tar`. LISTFILES ist ein wahlfreies Schlüsselwort. Seine Angabe bewirkt, dass das Dienst-

programm eine Liste der im Archiv enthaltenen Dateien zurückgibt. Führen Sie das Dienstprogramm **dlfm\_import** mit Rootberechtigung auf jedem Data Links-Server aus, weil Sie die Archivdateien möglicherweise auf einem anderen Data Links-Server wiederherstellen möchten, der über eine abweichende Verzeichnisstruktur und andere Benutzer-IDs verfügt als der Data Links-Server, auf dem das Dienstprogramm **dlfm\_export** ausgeführt wurde. Legen Sie die Registrierdatenbankvariable **DLFM\_FS\_ENVIRONMENT** entsprechend fest, bevor Sie das Dienstprogramm **dlfm\_import** ausführen.

**Anmerkung:** Wird das Dienstprogramm **dlfm\_import** nicht auf demselben Data Links-Server ausgeführt, auf dem das Dienstprogramm **dlfm\_export** ausgeführt wurde, werden die Dateien in den richtigen Pfaden wiederhergestellt. Der Eigner der Dateien ist Root, falls einige der Benutzer-IDs auf der Maschine, auf der die Dateien importiert werden, nicht vorhanden sind. Stellen Sie vor dem Einfügen dieser Dateien in eine Datenbank sicher, dass alle Dateien die richtigen Berechtigungen aufweisen und die richtigen Benutzer-IDs ihre Eigner sind.

Die folgende Tabelle zeigt das Exportieren der DB2-Daten und der Dateien, auf die von der Datenbank namens SystemA zur Datenbank SystemB verwiesen wird. SystemA verwendet die Data Links-Server DLFM1 und DLFM2. SystemB verwendet die Data Links-Server DLFMX und DLFMY. Die Dateien auf DLFM1 werden nach DLFMX exportiert, und die Dateien auf DLFM2 werden nach DLFMY exportiert.

Datenbank SystemA mit Data Links-Servern DLFM1 und DLFM2			Schritt
DB2-Daten in Datei	Datei1 mit Dateinamen für DLFM1	Datei2 mit Dateinamen für DLFM2	1) Führen Sie den Befehl <b>dlfm_export</b> (mit Root-Berechtigung) auf beiden Data Links-Servern aus. Dadurch wird auf beiden Data Links-Servern ein Archiv erstellt.
Datenbank SystemB mit Data Links-Servern DLFMX und DLFMY			
	Bei DLFMX Wiederherstellen vom Archiv	Bei DLFMY Wiederherstellen vom Archiv	2) Führen Sie den Befehl <b>dlfm_import</b> (mit Root-Berechtigung) auf beiden Data Links-Servern aus.
			3) Führen Sie den Befehl <b>IMPORT</b> auf SystemB mit dem Parameter <b>DL_URL_REPLACE_PREFIX</b> aus, um den entsprechenden Data Links-Server für jede exportierte Datei anzugeben.
Bei der Ausführung des Befehls <b>IMPORT</b> auf SystemB werden die Daten von SystemA und alle Dateien, auf die von DATALINK-Spalten verwiesen wird, importiert.			

**Zugehörige Tasks:**

- „Verwenden von EXPORT“ auf Seite 3
- „Versetzen von DB2 Data Links Manager-Daten mit EXPORT“ auf Seite 230

**Zugehörige Referenzen:**

- „Export-API "db2Export"“ auf Seite 14

---

## Versetzen von DB2 Data Links Manager-Daten mit EXPORT

### Prozedur:

So können Sie sicherstellen, dass eine konsistente Kopie der Tabelle und der entsprechenden Dateien, auf die in den DATALINK-Spalten verwiesen wird, kopiert wird:

1. Setzen Sie den folgenden Befehl ab, um sicherzustellen, dass während der Ausführung der Exportoperation keine Aktualisierungstransaktionen ablaufen:  

```
db2 quiesce tablespaces for table tabellenname share
```
2. Rufen Sie das Dienstprogramm EXPORT auf.
3. Führen Sie das Dienstprogramm **dlfm\_export** mit Root-Berechtigung auf jedem Data Links-Server aus. Damit können Dateien archiviert werden, auf die der Data Links-Dateimanageradministrator keinen Zugriff hat. Die Dienstprogramm erfasst nicht die ACL-Informationen der archivierten Dateien. Geben Sie als Eingabe für **dlfm\_export** den Namen der Steuerdatei an, die durch das Dienstprogramm EXPORT generiert wurde.
4. Setzen Sie den folgenden Befehl ab, um die Tabelle für Aktualisierungen verfügbar zu machen:

```
db2 quiesce tablespaces for table tabellenname reset
```

### Zugehörige Konzepte:

- „EXPORT - Übersicht“ auf Seite 1
- „Versetzen von DB2 Data Links Manager-Daten mit EXPORT - Konzepte“ auf Seite 227

### Zugehörige Tasks:

- „Versetzen von DB2 Data Links Manager-Daten mit IMPORT“ auf Seite 231
- „Versetzen von DB2 Data Links Manager-Daten mit LOAD“ auf Seite 232

## Versetzen von DB2 Data Links Manager-Daten mit IMPORT

Da sich die Tabellendaten in der Datenbank und die Dateien, auf die DATALINK-Spalten verweisen, auf Data Links-Servern befinden, muss das Dienstprogramm IMPORT sowohl die Datenbankdaten als auch die Datendateien auf den entsprechenden Data Links-Servern versetzen.

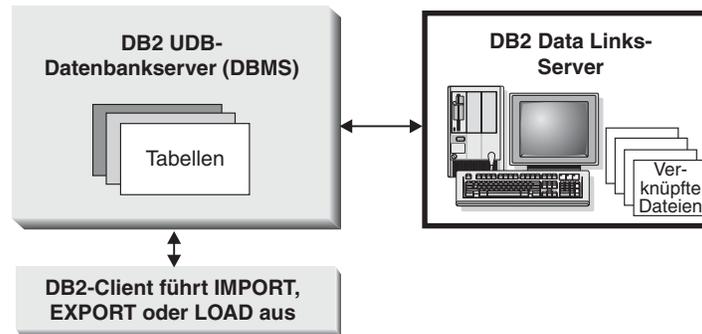


Abbildung 12. Versetzen von DB2 Data Links Manager-Daten. Da sich die Tabellendaten in der Datenbank und die Dateien, auf die DATALINK-Spalten verweisen, auf Data Links-Servern befinden, müssen die Dienstprogramme EXPORT, IMPORT und LOAD sowohl die Datenbankdaten als auch die Datendateien auf den entsprechenden Data Links-Servern versetzen.

### Prozedur:

Vor der Ausführung des Dienstprogramms IMPORT für die Zieldatenbank sollten Sie die folgenden Schritte ausführen:

1. Kopieren Sie die Dateien, auf die verwiesen wird, auf die entsprechenden Data Links-Server. Mit dem Dienstprogramm **dlfm\_import** können Dateien aus einem Archiv extrahiert werden, das vom Dienstprogramm **dlfm\_export** erstellt wurde.
2. Definieren Sie auf den Data Links-Dateimanagern oder den Data Links-Servern den bzw. die Präfixnamen. (Sie müssen eventuell auch andere Verwaltungsaufgaben ausführen, wie das Registrieren der Datenbank.)
3. Aktualisieren Sie die Informationen zu den Data Links-Servern in den URL-Adressen (der DATALINK-Spalten) aus den exportierten Daten für die SQL-Tabelle (falls erforderlich). (Wenn die Data Links-Server der ursprünglichen Konfiguration mit den Servern an der Zielposition übereinstimmen, müssen die Namen der Data Links-Server nicht aktualisiert werden.)
4. Definieren Sie die Data Links-Server in der Zielkonfiguration in der DB2 Data Links Manager-Konfigurationsdatei.

Wenn das Dienstprogramm IMPORT für die Zieldatenbank ausgeführt wird, werden Dateien, auf die in DATALINK-Spalten verwiesen wird, mit den entsprechenden Data Links-Servern verknüpft.

### Zugehörige Konzepte:

- „IMPORT - Übersicht“ auf Seite 27

### Zugehörige Tasks:

- „Versetzen von DB2 Data Links Manager-Daten mit EXPORT“ auf Seite 230
- „Versetzen von DB2 Data Links Manager-Daten mit LOAD“ auf Seite 232

---

## Versetzen von DB2 Data Links Manager-Daten mit LOAD

### Prozedur:

Wenn Sie Daten in eine Tabelle mit einer DATALINK-Spalte laden, die mit FILE LINK CONTROL definiert ist, müssen Sie die folgenden Schritte ausführen, bevor Sie das Dienstprogramm LOAD aufrufen. (Wenn alle DATALINK-Spalten mit NO LINK CONTROL definiert sind, sind diese Schritte nicht erforderlich.)

1. Stellen Sie sicher, dass DB2 Data Links Manager auf den Data Links-Servern, auf die die DATALINK-Spaltenwerte verweisen, installiert ist.
2. Stellen Sie sicher, dass die Datenbank im DB2 Data Links Manager eingetragen ist.
3. Kopieren Sie alle Dateien, die als DATALINK-Werte eingefügt werden, auf die entsprechenden Data Links-Server.
4. Definieren Sie für die DB2 Data Links Manager auf den Data Links-Servern den bzw. die Präfixnamen.
5. Registrieren Sie die Data Links-Server, auf die in den (zu ladenden) DATALINK-Daten verwiesen wird, in der DB2 Data Links Manager-Konfigurationsdatei.

Während der Ladeoperation fehlgeschlagene Verbindungen werden als eine Art von Datenintegritätsverletzung eingestuft und wie Verletzungen eindeutiger Indizes gehandhabt. Es wurde daher eine besondere Ausnahmebedingung für das Laden von Tabellen definiert, die mindestens eine DATALINK-Spalte enthalten.

Die folgenden Funktionen des Dienstprogramms LOAD werden beim Laden von Tabellen mit DATALINK-Spalten nicht unterstützt:

- CPU\_PARALLELISM (der Wert wird auf 1 gesetzt)
- LOAD REPLACE
- LOAD TERMINATE
- LOAD COPY

### Zugehörige Referenzen:

- „Befehl LOAD“ auf Seite 112

---

## Kapitel 6. Versetzen von Daten zwischen Systemen

In diesem Kapitel wird beschrieben, wie die DB2-Dienstprogramme EXPORT, IMPORT und LOAD verwendet werden, um Daten zwischen verschiedenen Plattformen sowie in und aus iSeries-Hostdatenbanken zu übertragen. Auch DB2 DataPropagator, eine weitere Methode zum Versetzen von Daten innerhalb eines Unternehmens, wird beschrieben. Dieses Kapitel enthält außerdem eine Einführung in die Data Warehouse-Zentrale, mit deren Hilfe Daten aus Betriebsdatenbanken in eine Warehouse-Datenbank versetzt werden können.

Die folgenden Themen werden behandelt:

- „Versetzen von Daten zwischen Plattformen - Hinweise zum Dateiformat“
- „Versetzen von Daten mit DB2 Connect“ auf Seite 235
- „Befehl "db2move" - Tool zum Versetzen von Daten“ auf Seite 237
- „Befehl "db2relocatedb" zum Verlagern einer Datenbank“ auf Seite 242
- „Versetzen von Daten zwischen typisierten Tabellen“ auf Seite 248
- „Verwenden der Replikation zum Versetzen von Daten“ auf Seite 253
- „Versetzen von Daten mit der Data Warehouse-Zentrale“ auf Seite 255.
- „Versetzen von Daten mit Dateityp CURSOR“ auf Seite 257.

---

### Versetzen von Daten zwischen Plattformen - Hinweise zum Dateiformat

Kompatibilität ist beim Exportieren, Importieren oder Laden von Daten zwischen verschiedenen Plattformen wichtig. In den folgenden Abschnitten finden Sie Hinweise zu den PC/IXF-, WSF- und DEL-Dateiformaten (ASCII-Format ohne universelle Zeilenbegrenzer) beim Versetzen von Daten zwischen verschiedenen Betriebssystemen.

#### PC/IXF-Dateiformat

PC/IXF ist das bevorzugte Dateiformat zum Versetzen von Daten zwischen Plattformen. PC/IXF-Dateien ermöglichen es dem Dienstprogramm LOAD oder IMPORT, numerische Daten, die normalerweise von der Maschine abhängig sind, maschinenunabhängig zu verarbeiten. Numerische Daten werden beispielsweise von der Intel-Architektur anders als von anderen Hardwarearchitekturen gespeichert und verwaltet.

Um die Kompatibilität der PC/IXF-Dateien für alle Produkte der DB2<sup>®</sup>-Familie herzustellen, erstellt das Dienstprogramm EXPORT Dateien mit numerischen Daten im Intel-Format, während das Dienstprogramm IMPORT Daten dieses Formats erwartet.

Abhängig von der Hardwareplattform konvertieren DB2-Produkte numerische Werte zwischen Intel- und Nicht-Intel-Formaten (mittels Bytefolgeumkehrung) während der Export- und Importoperationen.

Auf UNIX<sup>®</sup> basierende Implementierungen von DB2 erstellen während des Exports keine mehrteiligen PC/IXF-Dateien. Der Import einer mehrteiligen PC/IXF-Datei ist dagegen zulässig, sofern die betreffende Datei von DB2 erstellt wurde. Wenn eine solche Datei importiert wird, müssen sich alle zugehörigen Teildateien im selben Verzeichnis befinden. Andernfalls meldet das Dienstprogramm einen Fehler.

Einteilige PC/IXF-Dateien, die durch auf UNIX basierende Implementierungen des DB2-Dienstprogramms EXPORT erstellt wurden, können mit DB2 für Windows® importiert werden.

## DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer)

DEL-Dateien weisen betriebssystemspezifische Unterschiede auf, je nachdem, unter welchem Betriebssystem sie erstellt wurden. Es gibt folgende Unterschiede:

- Zeilentrennzeichen
  - Auf UNIX basierende Textdateien verwenden ein Zeilenvorschubzeichen.
  - Nicht auf UNIX basierende Textdateien verwenden eine Folge aus Rücklauf- und Zeilenvorschubzeichen.
- Dateiendezeichen
  - Auf UNIX basierende Textdateien verwenden kein Dateiendezeichen.
  - Nicht auf UNIX basierende Textdateien verwenden ein Dateiendezeichen (X'1A').

Da DEL-Exportdateien Textdateien sind, können sie von einem Betriebssystem auf ein anderes übertragen werden. Dateiübertragungsprogramme können betriebssystemabhängige Dateien unterschiedlich verarbeiten, wenn Sie die Dateien im Textmodus übertragen. Zeilentrennzeichen und Dateiendezeichen werden im Binärmodus nicht konvertiert.

**Anmerkung:** Wenn die Zeichendatenfelder Zeilentrennzeichen enthalten, werden diese ebenfalls während der Dateiübertragung konvertiert. Diese Konvertierung führt zu unvorhersehbaren Änderungen der Daten. Aus diesem Grund sollten Sie keine DEL-Exportdateien verwenden, um Daten zwischen Plattformen zu versetzen. Verwenden Sie stattdessen das PC/IXF-Dateiformat.

## WSF-Dateiformat

Numerische Daten in Dateien im WSF-Format werden im Intel-Maschinenformat gespeichert. Dieses Format ermöglicht es, Lotus®-WSF-Dateien in andere Lotus-Betriebsumgebungen (z. B. auf Intel und auf UNIX basierende Systeme) zu übertragen und dort zu verwenden.

Als Folge dieser Konsistenz der internen Formate können WSF-Dateien, die aus DB2-Produkten exportiert wurden, von Lotus 1-2-3® oder Symphony auf anderen Plattformen verwendet werden. Außerdem können DB2-Produkte auch WSF-Dateien importieren, die auf anderen Plattformen erstellt wurden.

Verwenden Sie bei der Übertragung von WSF-Dateien zwischen Betriebssystemen stets den Binärmodus, nicht den Textmodus.

**Anmerkung:** Verwenden Sie das WSF-Dateiformat nicht zur Übertragung von Daten zwischen DB2-Datenbanken auf verschiedenen Plattformen, da dies zu Datenverlust führen kann. Verwenden Sie stattdessen das PC/IXF-Dateiformat.

### Zugehörige Referenzen:

- „Dateiformate der Dienstprogramme EXPORT, IMPORT und LOAD“ auf Seite 275

## Versetzen von Daten mit DB2 Connect

Wenn Sie in einer komplexen Umgebung arbeiten, in der Sie Daten zwischen einem Hostdatenbanksystem und einer Workstation versetzen müssen, können Sie DB2 Connect, den Gateway für die Datenübertragung zwischen Host und Workstation, verwenden (siehe Abb. 13).

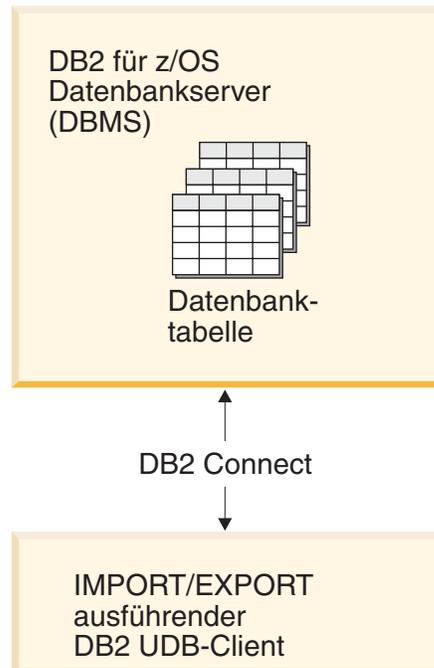


Abbildung 13. Import/Export über DB2 Connect

Mit den DB2-Dienstprogrammen EXPORT und IMPORT können Daten aus einer Datenbank auf einem Host- oder iSeries-Server in eine Datei auf der DB2 Connect-Workstation und umgekehrt versetzt werden. Anschließend können diese Daten mit jeder anderen Anwendung oder jedem anderen Verwaltungssystem für relationale Datenbanken verwendet werden, von der/dem das betreffende Import-/Exportformat unterstützt wird. Beispielsweise können Sie Daten aus einer Datenbank auf einem Host- oder iSeries-Server in eine PC/IXF-Datei exportieren und diese Daten anschließend in eine DB2 für Windows-Datenbank importieren.

Sie können Export- und Importfunktionen von einem Datenbankclient oder von der DB2 Connect-Workstation aus ausführen.

### Anmerkungen:

1. Die zu importierenden oder exportierenden Daten dürfen die Beschränkungen der beiden Datenbanken hinsichtlich der Größe und Datentypen nicht verletzen.
2. Zur Verbesserung der Leistung beim Importieren können zusammengesetzte SQL-Anweisungen verwendet werden. Geben Sie beim Dienstprogramm IMPORT den Änderungswert für den Dateityp compound an, um eine angegebene Anzahl von SQL-Anweisungen zu einem Block zusammenzufassen. Hierdurch können der Systemaufwand für das Netzwerk verringert und die Antwortzeiten verbessert werden.

### Einschränkungen:

Bei DB2 Connect müssen Export- und Importoperationen die folgenden Bedingungen erfüllen:

- Der Dateityp muss PC/IXF sein.
- Es muss eine Zieltabelle auf dem Zielsever erstellt werden, deren Attribute mit denen der Daten kompatibel sind, bevor der Importvorgang gestartet werden kann. Mit dem Dienstprogramm **db2look** können Sie die Attribute der Quellentabelle abrufen. Beim Import durch DB2 Connect kann keine Tabelle erstellt werden, weil INSERT die einzige unterstützte Option ist.

Wenn eine dieser Bedingungen nicht erfüllt ist, schlägt die Operation fehl, und eine Fehlermeldung wird zurückgegeben.

**Anmerkung:** Indexdefinitionen werden weder beim Export gespeichert noch beim Import verwendet.

Beim Importieren und Exportieren gemischter Daten (d. h. Spalten, die sowohl SBCS- als auch DBCS-Daten enthalten) muss Folgendes beachtet werden:

- Auf Systemen, die Daten im EBCDIC-Format speichern (MVS, OS/390, OS/400, VM und VSE), markieren DBCS-Startzeichen und DBCS-Endezeichen den Anfang und das Ende der DBCS-Daten. Bei der Definition von Spaltenlängen für die Datenbanktabellen muss daher genügend Platz für diese Zeichen gelassen werden.
- Wenn die Daten in einer Spalte kein einheitliches Muster haben, werden Spalten für Zeichendaten variabler Länge empfohlen.

### Versetzen von Daten von einer Workstation auf einen Host-Server:

So versetzen Sie Daten in eine Datenbank auf einem Host- oder AS/400- bzw. iSeries-Server:

1. Exportieren Sie die Daten aus einer DB2-Tabelle in eine PC/IXF-Datei.
2. Importieren Sie die PC/IXF-Datei mit der Option INSERT in eine kompatible Tabelle der Datenbank auf dem Host-Server.

So versetzen Sie Daten aus einer Datenbank auf dem Host-Server auf eine Workstation:

1. Exportieren Sie die Daten aus der Tabelle der Datenbank auf dem Host-Server in eine PC/IXF-Datei.
2. Importieren Sie die PC/IXF-Datei in eine DB2-Tabelle.

### Beispiel

Das folgende Beispiel veranschaulicht, wie Sie Daten von einer Workstation in eine Datenbank auf einem Host- oder AS/400- bzw. iSeries-Server versetzen.

1. Exportieren Sie die Daten in ein externes IXF-Format, indem Sie den folgenden Befehl absetzen:

```
db2 export to staff.ixf of ixf select * from userid.staff
```

2. Setzen Sie den folgenden Befehl ab, um eine DRDA-Verbindung zum DB2 UDB-Zielserver herzustellen:

```
db2 connect to cbc664 user admin using xxx
```

3. Erstellen Sie die Zieltabelle auf dem DB2 UDB-Zielserver, wenn sie noch nicht vorhanden ist:

```
CREATE TABLE mydb.staff (ID SMALLINT NOT NULL, NAME VARCHAR(9),  
DEPT SMALLINT, JOB CHAR(5), YEARS SMALLINT, SALARY DECIMAL(7,2),  
COMM DECIMAL(7,2))
```

4. Setzen Sie den folgenden Befehl ab, um die Daten zu importieren:

```
db2 import from staff.ixf of ixf insert into mydb.staff
```

Jede Datenzeile wird aus der Datei im IXF-Format gelesen, und eine SQL-Anweisung INSERT wird abgesetzt, um die betreffende Zeile in die Tabelle 'mydb.staff' einzufügen. Es werden so lange einzelne Zeilen eingefügt, bis alle Daten in die Zieltabelle versetzt wurden.

Ausführliche Informationen enthält das folgende IMB Redbook: Moving Data Across the DB2 Family. Dieses Redbook steht unter folgender URL zur Verfügung: <http://www.redbooks.ibm.com/redbooks/SG246905.html>.

#### Zugehörige Konzepte:

- „Versetzen von Daten zwischen Plattformen - Hinweise zum Dateiformat“ auf Seite 233

#### Zugehörige Referenzen:

- „Befehl EXPORT“ auf Seite 9
- „Befehl IMPORT“ auf Seite 39

---

## Befehl "db2move" - Tool zum Versetzen von Daten

Dieses Tool kann Sie bei der Übertragung großer Mengen von Tabellen zwischen DB2-Datenbanken auf Workstations unterstützen. Das Tool fragt die Systemkatalogtabellen nach einer bestimmten Datenbank ab und stellt eine Liste aller Benutzertabellen zusammen. Anschließend exportiert es diese Tabellen im PC/IXF-Format. Die PC/IXF-Dateien können in andere lokale DB2-Datenbanken auf demselben System importiert oder geladen werden oder auf eine andere Workstation-Plattform übertragen und auf dieser Plattform in eine DB2-Datenbank importiert oder geladen werden.

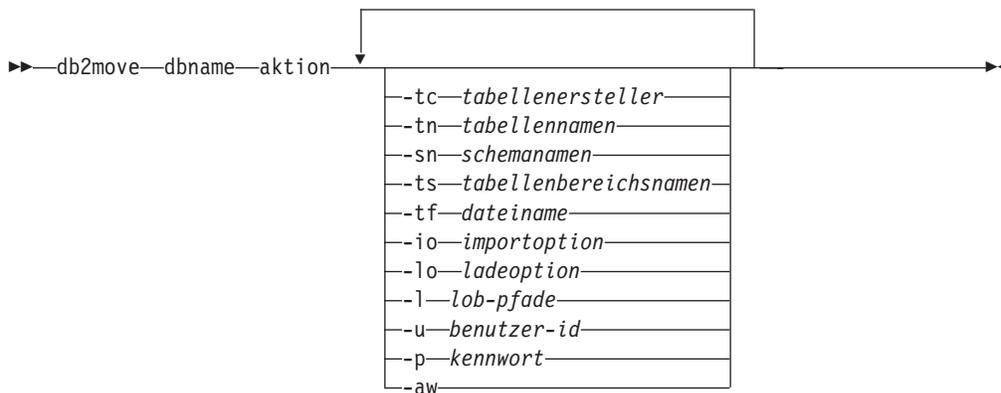
**Anmerkung:** Tabellen mit Spalten strukturierten Typs werden bei Verwendung dieses Tools nicht versetzt.

#### Berechtigung:

Dieses Tool ruft, je nach der vom Benutzer angeforderten Aktion, die Export-, Import- oder Lade-API von DB2 auf. Daher muss die anfordernde Benutzer-ID über die richtigen Berechtigungen für die APIs verfügen, sonst schlägt die Anforderung fehl.

## Tool "db2move" zum Versetzen von Daten

### Befehlssyntax:



### Befehlsparameter:

#### **dbname**

Der Name der Datenbank.

**aktion** Muss EXPORT, IMPORT oder LOAD sein.

**-tc** Tabellenersteller. Standardmäßig werden alle Ersteller ausgewählt.

Diese Option gilt nur für die Aktion EXPORT. Wenn sie angegeben wird, werden nur die von den mit dieser Option angegebenen Erstellern erstellten Tabellen exportiert. Wird sie nicht angegeben, werden standardmäßig alle Ersteller verwendet. Bei Angabe mehrerer Ersteller müssen diese Angaben durch ein Komma getrennt werden. Leerzeichen sind zwischen Ersteller-IDs nicht zulässig. Die maximale Anzahl von Erstellern, die angegeben werden kann, ist 10. Diese Option kann zusammen mit der Option "-tn" (Tabellennamen) zur Auswahl der zu exportierenden Tabellen verwendet werden.

Der Stern (\*) kann an einer beliebigen Stelle in der Zeichenfolge als Platzhalterzeichen angegeben werden.

**-tn** Tabellennamen. Standardmäßig werden alle Benutzertabellen ausgewählt.

Diese Option gilt nur für die Aktion EXPORT. Wenn sie angegeben wird, werden nur die Tabellen exportiert, deren Namen exakt mit denen in der angegebenen Zeichenfolge übereinstimmen. Wird sie nicht angegeben, werden standardmäßig alle Benutzertabellen verwendet. Bei der Angabe mehrerer Tabellennamen müssen diese Angaben durch ein Komma getrennt werden. Leerzeichen sind zwischen Tabellennamen nicht zulässig. Die maximale Anzahl von Tabellennamen, die angegeben werden kann, ist 10. Diese Option kann zusammen mit der Option "-tc" (Tabellenersteller) zur Auswahl der zu exportierenden Tabellen verwendet werden. **db2move** exportiert nur die Tabellen, deren Namen mit den angegebenen Tabellennamen und deren Ersteller mit den angegebenen Erstellern übereinstimmen.

Der Stern (\*) kann an einer beliebigen Stelle in der Zeichenfolge als Platzhalterzeichen angegeben werden.

**-sn** schemanamen. Standardmäßig werden alle Schemas ausgewählt.

Wenn dieser angegeben wird, werden nur die Tabellen exportiert, deren Schemanamen genau übereinstimmen. Wenn das Platzhalterzeichen Stern (\*) in den Schemanamen verwendet wird, wird dieser in ein Prozent-

zeichen (%) geändert und der Tabellename (mit Prozentzeichen) wird im Prädikat LIKE der Klausel WHERE verwendet. Wird sie nicht angegeben, werden standardmäßig alle Schemas verwendet. Bei der Angabe mehrerer Schemanamen müssen diese Angaben durch ein Komma getrennt werden. Leerzeichen sind zwischen Schemanamen nicht zulässig. Die maximale Anzahl von Schemanamen, die angegeben werden kann, ist 10. Wenn **db2move** mit der Option -tn oder -tc verwendet wird, werden nur jene Tabellen exportiert, deren Schemata mit dem angegebenen Schemanamen und deren Eigner mit den angegebenen Eignern übereinstimmen.

**Anmerkung:** Schemanamen mit weniger als 8 Zeichen werden auf 8 Zeichen Länge aufgefüllt. Beispielsweise muss ein Schemaname 'fred' als "-sn fr\*d\*" statt "-sn fr\*d" angegeben werden, wenn ein Stern verwendet wird.

**-ts** tabellenbereichsnamen. Standardmäßig werden alle Tabellenbereiche ausgewählt.

Diese Option gilt nur für die Aktion EXPORT. Wenn diese Option angegeben wird, werden nur jene Tabellen exportiert, die sich im angegebenen Tabellenbereich befinden. Wenn das Platzhalterzeichen Stern (\*) in den Tabellenbereichsnamen verwendet wird, wird dieser in ein Prozentzeichen (%) geändert und der Tabellename (mit Prozentzeichen) wird im Prädikat LIKE in der Klausel WHERE verwendet. Wird die Option -ts nicht angegeben, werden standardmäßig alle Tabellenbereiche verwendet. Bei der Angabe mehrerer Tabellenbereichsnamen müssen diese Angaben durch ein Komma getrennt werden. Leerzeichen sind zwischen Tabellenbereichsnamen nicht zulässig. Die maximale Anzahl von Tabellenbereichsnamen, die angegeben werden kann, ist 10.

**Anmerkung:** Tabellenbereichsnamen mit weniger als 8 Zeichen werden auf 8 Zeichen Länge aufgefüllt. Beispielsweise muss ein Tabellenbereichsname 'mytb' als "-ts my\*b\*" statt "-sn my\*b" angegeben werden, wenn ein Stern verwendet wird.

**-tf** dateiname

Diese Option gilt nur für die Aktion EXPORT. Wird sie angegeben, werden nur die Tabellen exportiert, die in der Datei aufgelistet werden. Pro Zeile darf nur eine Tabelle aufgelistet werden, und jede Tabelle muss vollständig qualifiziert sein. Es folgt ein Beispiel für den Inhalt einer Datei:

```
"SCHEMA1"."TABLE NAME1"  
"SCHEMA NAME77"."TABLE155"
```

**-io** Importoption. Der Standardwert ist REPLACE\_CREATE.

Gültige Werte sind INSERT, INSERT\_UPDATE, REPLACE, CREATE und REPLACE\_CREATE.

**-lo** Ladeoption. Der Standardwert ist INSERT.

Gültige Optionen sind INSERT und REPLACE.

**-l** LOB-Pfade. Der Standardwert ist das aktuelle Verzeichnis.

Diese Option gibt die absoluten Pfadnamen an, in denen LOB-Dateien (bei EXPORT) erstellt bzw. (bei IMPORT oder LOAD) gesucht werden. Bei der Angabe mehrerer LOB-Pfade muss jede Angabe durch ein Komma getrennt werden. Leerzeichen sind zwischen LOB-Pfaden nicht zulässig. Wenn im

## Tool "db2move" zum Versetzen von Daten

ersten Pfad kein Speicherplatz mehr verfügbar ist (bei EXPORT) oder die Dateien nicht gefunden werden (bei IMPORT oder LOAD), wird der zweite Pfad verwendet usw.

Wenn als Aktion EXPORT angegeben wurde und LOB-Pfade angegeben werden, werden alle Dateien in den LOB-Pfadverzeichnissen gelöscht, die Verzeichnisse entfernt und neue Verzeichnisse erstellt. Wenn diese Option nicht angegeben wird, wird das aktuelle Verzeichnis als LOB-Pfad verwendet.

- u** Benutzer-ID. Der Standardwert ist die angemeldete Benutzer-ID.  
Sowohl die Benutzer-ID als auch das Kennwort sind wahlfrei. Wenn jedoch eines von beiden angegeben wird, müssen beide angegeben werden. Wenn der Befehl auf einem Client ausgeführt wird, der eine Verbindung zu einem fernen Server herstellt, sollten die Benutzer-ID und das Kennwort angegeben werden.
- p** Kennwort. Standardwert ist das bei der Anmeldung verwendete Kennwort.  
Sowohl die Benutzer-ID als auch das Kennwort sind wahlfrei. Wenn jedoch eines von beiden angegeben wird, müssen beide angegeben werden. Wenn der Befehl auf einem Client ausgeführt wird, der eine Verbindung zu einem fernen Server herstellt, sollten die Benutzer-ID und das Kennwort angegeben werden.
- aw** Warnungen zulassen. Wenn '-aw' nicht angegeben wird, sind Tabellen nicht in der Datei db2move.lst enthalten, bei deren Export Warnungen vorkommen (obwohl die Datei .ixf und .msg der Tabelle dennoch generiert werden). In einigen Szenarien (wie zum Beispiel Abschneiden von Daten) kann es vorkommen, dass der Benutzer möchte, dass solche Tabellen in der Datei db2move.lst enthalten sind. Wenn diese Option angegeben wird, können Tabellen, die beim Export Warnungen empfangen, in der Datei .lst enthalten sein.

### Beispiele:

- db2move sample export  
Mit diesem Befehl werden alle Tabellen in der Beispieldatenbank SAMPLE exportiert. Für alle Optionen werden die Standardwerte verwendet.
- db2move sample export -tc userid1,us\*rid2 -tn tname1,\*tname2  
Mit diesem Befehl werden alle Tabellen exportiert, die von "userid1" oder Benutzer-IDs wie "us%rid2" und mit dem Namen "tname1" oder Tabellennamen wie "%tname2" erstellt wurden.
- db2move sample import -l D:\LOBPATH1,C:\LOBPATH2  
Dieses Beispiel gilt nur für das Betriebssystem Windows. Mit diesem Befehl werden alle Tabellen in SAMPLE importiert, zugehörige LOB-Dateien werden in den LOB-Pfaden "D:\LOBPATH1" und "C:\LOBPATH2" gesucht.
- db2move sample load -l /home/userid/lobpath,/tmp  
Dieses Beispiel gilt nur für auf UNIX basierende Systeme. Mit diesem Befehl werden alle Tabellen in die Beispieldatenbank SAMPLE geladen, zugehörige LOB-Dateien werden im Unterverzeichnis /home/userid/lobpath und im Unterverzeichnis tmp gesucht.
- db2move sample import -io replace -u userid -p password  
Mit diesem Befehl werden alle Tabellen in die Beispieldatenbank SAMPLE im Ersetzungsmodus (REPLACE) importiert, und die angegebene Kombination aus Benutzer-ID und Kennwort wird verwendet.

### Anmerkungen zur Verwendung:

Dieses Tool exportiert, importiert oder lädt von Benutzern erstellte Tabellen. Wenn die Datenbank von einem Betriebssystem in ein anderes Betriebssystem dupliziert werden soll, vereinfacht **db2move** das Versetzen der Tabellen. Sie müssen auch daran denken, alle anderen Objekte, die den Tabellen zugeordnet sind, zu versetzen. Hierzu gehören Aliasnamen, Sichten, Auslöser, benutzerdefinierte Funktionen usw. Wird zum Erstellen der Tabellen in der Zieldatenbank das Dienstprogramm IMPORT mit der Option REPLACE\_CREATE verwendet, gelten die unter Erneutes Erstellen einer exportierten Tabelle mit IMPORT beschriebenen Einschränkungen. Treten bei Verwendung der Option REPLACE\_CREATE während der Importphase von **db2move** unerwartete Fehler auf, untersuchen Sie die entsprechende Nachrichtendatei tabnnn.msg und bedenken Sie, dass die Fehler aus den Einschränkungen bei der Erstellung von Tabellen resultieren können.

Wenn die Export-, Import- oder Lade-API von **db2move** aufgerufen wird, wird der Parameter FileTypeMod auf lobsinfile gesetzt. Das heißt, LOB-Daten werden in Dateien gespeichert, die von PC/IXF-Dateien getrennt sind. Es sind 26.000 Dateinamen für LOB-Dateien verfügbar.

Die Aktion LOAD muss lokal auf der Maschine ausgeführt werden, auf der sich die Datenbank und die Datendatei befinden. Wenn die Lade-API von **db2move** aufgerufen wird, wird der Parameter CopyTargetList auf den Wert NULL gesetzt, d. h. es erfolgt kein Kopiervorgang. Wenn der Parameter *logretain* aktiviert ist, kann die Ladeoperation später nicht aktualisierend wiederhergestellt werden. Der Tabellenbereich, in dem die geladene Tabelle gespeichert ist, wird in den Status "Sicherung anstehend" versetzt und ist für den Zugriff nicht verfügbar. Es ist eine vollständige Datenbanksicherung oder eine Tabellenbereichssicherung erforderlich, um den Status "Sicherung anstehend" für den Tabellenbereich aufzuheben.

**Anmerkung:** Die Leistung von 'db2move import' kann verbessert werden, indem der standardmäßige Pufferpool, IBMDEFAULTBP geändert wird und indem die Konfigurationsparameter *sorthheap*, *util\_heap\_sz*, *logfilsz* und *logprimary* aktualisiert werden.

### Bei Verwendung von EXPORT erforderliche/generierte Dateien:

- Eingabe: Keine.
- Ausgabe:

<b>EXPORT.out</b>	Das zusammengefasste Ergebnis der Aktion EXPORT.
<b>db2move.lst</b>	Die Liste der Originaltabellennamen, ihrer entsprechenden PC/IXF-Dateinamen (tabnnn.ixf) sowie der Nachrichtendateinamen (tabnnn.msg). Diese Liste, die exportierten PC/IXF-Dateien und die LOB-Dateien (tabnnnc.yyy) werden als Eingabe für die Aktion IMPORT oder LOAD von <b>db2move</b> verwendet.
<b>tabnnn.ixf</b>	Die exportierte PC/IXF-Datei einer bestimmten Tabelle.
<b>tabnnn.msg</b>	Die exportierte Nachrichtendatei der entsprechenden Tabelle.
<b>tabnnnc.yyy</b>	Die exportierten LOB-Dateien einer bestimmten Tabelle. "nnn" ist die Tabellenummer. "c" ist ein Buchstabe des Alphabets. "yyy" ist eine Ziffer aus dem Bereich von 001 bis 999.  Diese Dateien werden nur erstellt, sofern die Tabelle, die exportiert wird, LOB-Daten enthält. Wenn sie erstellt werden, werden

## Tool "db2move" zum Versetzen von Daten

diese LOB-Dateien in den LOB-Pfadverzeichnissen (Wert für "lob-pfade") gespeichert. Es sind 26.000 Dateinamen für LOB-Dateien verfügbar.

**system.msg** Die Nachrichtendatei mit den Systemnachrichten zu den Befehlen für das Erstellen oder Löschen von Dateien oder Verzeichnissen. Diese Datei wird nur verwendet, sofern die Aktion EXPORT ist und ein LOB-Pfad angegeben wird.

### Bei Verwendung von IMPORT erforderliche/generierte Dateien:

- Eingabe:

**db2move.lst** Eine Ausgabedatei der Aktion EXPORT.

**tabnnn.ixf** Eine Ausgabedatei der Aktion EXPORT.

**tabnnnc.yyy** Eine Ausgabedatei der Aktion EXPORT.

- Ausgabe:

**IMPORT.out** Das zusammengefasste Ergebnis der Aktion IMPORT.

**tabnnn.msg** Die Datei mit IMPORT-Nachrichten für die entsprechende Tabelle.

### Bei Verwendung von LOAD erforderliche/generierte Dateien:

- Eingabe:

**db2move.lst** Eine Ausgabedatei der Aktion EXPORT.

**tabnnn.ixf** Eine Ausgabedatei der Aktion EXPORT.

**tabnnnc.yyy** Eine Ausgabedatei der Aktion EXPORT.

- Ausgabe:

**LOAD.out** Das zusammengefasste Ergebnis der Aktion LOAD.

**tabnnn.msg** Die Datei mit LOAD-Nachrichten für die entsprechende Tabelle.

### Zugehörige Referenzen:

- „db2look - DB2 Statistics and DDL Extraction Tool Command“ in *Command Reference*

---

## Befehl "db2relocatedb" zum Verlagern einer Datenbank

| Benennt eine Datenbank um, oder verlagert eine Datenbank oder einen Teil einer  
| Datenbank (z. B. den Behälter und das Protokollverzeichnis) wie in der durch den  
| Benutzer bereitgestellten Konfigurationsdatei angegeben. Dieses Tool nimmt die  
| erforderlichen Änderungen am DB2-Exemplar und an den Datenbankunter-  
| stützungsdateien vor.

### Berechtigung:

Keine

### Befehlssyntax:

►► db2relocatedb -f konfigurationsdateiname ◀◀

## Befehl "db2relocatedb" zum Verlagern einer Datenbank

### Befehlsparameter:

#### **-f konfigurationsdateiname**

Gibt den Namen der Datei mit den Konfigurationsdaten an, die zum Verlagern der Datenbank benötigt werden. Dies kann ein relativer oder ein absoluter Dateiname sein. Die Konfigurationsdatei hat folgendes Format:

```
DB_NAME=alterName,neuerName
DB_PATH=alterPfad,neuerPfad
INSTANCE=altesExemplar,neuesExemplar
NODENUM=knotenNummer
LOG_DIR=alterVerzPfad,neuerVerzPfad
CONT_PATH=alterBehältPfad1,neuerBehältPfad1
CONT_PATH=alterBehältPfad2,neuerBehältPfad2
...
```

Hierbei gilt:

#### **DB\_NAME**

Gibt den Namen der Datenbank an, die verlagert wird. Wenn der Name der Datenbank geändert wird, müssen der alte und der neue Name angegeben werden. Dies ist ein erforderliches Feld.

#### **DB\_PATH**

Gibt den Pfad der Datenbank an, die verlagert wird. Dies ist der Pfad in dem die Datenbank ursprünglich erstellt wurde. Wenn der Datenbankpfad geändert wird, müssen der alte und der neue Pfad angegeben werden. Dies ist ein erforderliches Feld.

#### **INSTANCE**

Gibt das Exemplar an, in dem die Datenbank vorhanden ist. Wenn die Datenbank in ein neues Exemplar versetzt wird, müssen das alte und das neue Exemplar angegeben werden. Dies ist ein erforderliches Feld.

#### **NODENUM**

Gibt die Knotennummer für den zu ändernden Datenbankknoten an. Der Standardwert ist 0.

#### **LOG\_DIR**

Gibt eine Änderung der Speicherposition des Protokollpfads an. Wenn der Protokollpfad geändert wird, müssen der alte und der neue Pfad angegeben werden. Diese Spezifikation ist optional, wenn sich der Protokollpfad unter dem Datenbankpfad befindet. In diesem Fall wird der Pfad automatisch aktualisiert.

#### **CONT\_PATH**

Gibt eine Änderung der Speicherposition der Tabellenbereichsbehälter an. Der alte und der neue Behälterpfad müssen angegeben werden. Sie können mehrere Zeilen CONT\_PATH angeben, wenn mehrere Änderungen am Behälterpfad vorgenommen werden müssen. Diese Spezifikation ist optional, wenn sich der Behälterpfad unter dem Datenbankpfad befindet. In diesem Fall werden die Pfade automatisch aktualisiert. Wenn Sie Änderungen an mehr als einem Behälter vornehmen, wobei derselbe alte Pfad durch einen gemeinsamen neuen Pfad ersetzt wird, können Sie einen einzigen Eintrag CONT\_PATH verwenden. In diesem Fall kann ein Stern (\*) als Platzhalterzeichen für den alten und neuen Pfad verwendet werden.

## Befehl "db2relocatedb" zum Verlagern einer Datenbank

**Anmerkung:** Leere Zeilen oder Zeilen, die mit einem Kommentarzeichen (#) beginnen, werden ignoriert.

### Hinweise:

Wenn ein Exemplar geändert wird, zu dem eine Datenbank gehört, müssen Sie die folgenden Schritte ausführen, bevor Sie diesen Befehl ausführen, um sicherzustellen, dass Änderungen am Exemplar und an den Datenbankunterstützungsdateien vorgenommen werden:

- Wenn eine Datenbank in ein anderes Exemplar versetzt wird, erstellen Sie das neue Exemplar.
- Kopieren Sie die Dateien/Einheiten der Datenbank, die kopiert wird, auf das System, auf dem sich das neue Exemplar befindet. Falls erforderlich müssen die Pfadnamen angepasst werden. Sind jedoch bereits Datenbanken in dem Verzeichnis vorhanden, in das die Datenbankdateien versetzt werden, kann es vorkommen, dass die bestehende Datei `sqlbdbir` versehentlich überschrieben wird und dadurch die Verweise auf die bestehenden Datenbanken entfernt werden. In diesem Szenario kann das Dienstprogramm **db2relocatedb** nicht verwendet werden. Eine umgeleitete Wiederherstellung ist eine Alternative zu **db2relocatedb**.
- Ändern Sie die Berechtigungen der kopierten Dateien/Einheiten, damit der Exemplareigner Eigner dieser Dateien/Einheiten wird.

Wenn das Exemplar geändert wird, muss das Tool vom neuen Exemplareigner ausgeführt werden.

In einer Umgebung mit partitionierten Datenbanken muss dieses Tool für alle Partitionen ausgeführt werden, auf denen Änderungen erforderlich sind. Für jede Partition muss eine eigene Konfigurationsdatei angegeben werden, die den Wert `NODENUM` der zu ändernden Partition enthält. Wenn z. B. der Name der Datenbank geändert wird, sind alle Partitionen betroffen, und der Befehl **db2relocatedb** muss auf jeder Partition mit einer eigenen Konfigurationsdatei ausgeführt werden. Wenn Behälter versetzt werden, die zu einer einzelnen Datenbankpartition gehören, muss der Befehl **db2relocatedb** nur einmal auf dieser Partition ausgeführt werden.

### Beispiele:

#### Beispiel 1

Erstellen Sie die folgende Konfigurationsdatei, um den Namen der Datenbank `TESTDB` im Exemplar `'db2inst1'`, das sich im Pfad `/home/db2inst1` befindet, in `PRODDB` zu ändern:

```
DB_NAME=TESTDB,PRODDB
DB_PATH=/home/db2inst1
INSTANCE=db2inst1
NODENUM=0
```

Speichern Sie die Konfigurationsdatei als `relocate.cfg`, und verwenden Sie den folgenden Befehl, um die Änderungen an den Datenbankdateien vorzunehmen:

```
db2relocatedb -f relocate.cfg
```

### Beispiel 2

Gehen Sie wie folgt vor, um die Datenbank DATAB1 aus dem Exemplar 'jsmith' im Pfad /dbpath in das Exemplar 'prodinst' zu versetzen:

1. Versetzen Sie die Dateien im Verzeichnis /dbpath/jsmith in das Verzeichnis /dbpath/prodinst.
2. Verwenden Sie die folgende Konfigurationsdatei mit dem Befehl **db2relocatedb**, um die Änderungen an den Datenbankdateien vorzunehmen:

```
DB_NAME=DATAB1
DB_PATH=/dbpath
INSTANCE=jsmith,prodinst
NODENUM=0
```

### Beispiel 3

Die Datenbank PRODDB ist im Exemplar 'inst1' im Pfad /databases/PRODDB vorhanden. Die Speicherposition von zwei Tabellenbereichsbehältern muss wie folgt geändert werden:

- Der SMS-Behälter /data/SMS1 muss nach /DATA/NewSMS1 versetzt werden.
- Der DMS-Behälter /data/DMS1 muss nach /DATA/DMS1 versetzt werden.

Nachdem die physischen Verzeichnisse und Dateien an die neuen Speicherpositionen versetzt wurden, kann die folgende Konfigurationsdatei mit dem Befehl **db2relocatedb** verwendet werden, um die Änderungen an den Datenbankdateien vorzunehmen, damit sie die neuen Speicherpositionen erkennen:

```
DB_NAME=PRODDB
DB_PATH=/databases/PRODDB
INSTANCE=inst1
NODENUM=0
CONT_PATH=/data/SMS1,/DATA/NewSMS1
CONT_PATH=/data/DMS1,/DATA/DMS1
```

### Beispiel 4

Die Datenbank TESTDB ist im Exemplar 'db2inst1' vorhanden und wurde im Pfad /databases/TESTDB erstellt. Danach wurden Tabellenbereiche mit den folgenden Behältern erstellt:

```
TS1
TS2_Cont0
TS2_Cont1
/databases/TESTDB/TS3_Cont0
/databases/TESTDB/TS4/Cont0
/Data/TS5_Cont0
/dev/rTS5_Cont1
```

TESTDB soll auf ein neues System versetzt werden. Das Exemplar auf dem neuen System ist 'newinst', und die Speicherposition der Datenbank ist /DB2.

Beim Versetzen der Datenbank müssen alle Dateien, die im Verzeichnis /databases/TESTDB/db2inst1 vorhanden sind, in das Verzeichnis /DB2/newinst versetzt werden. Das bedeutet, dass die ersten fünf Behälter als Teil des Versetzens verlagert werden. (Die ersten drei Behälter werden relativ zum Datenbankverzeichnis angegeben und die anderen beiden relativ zum Datenbankpfad.) Weil sich diese Behälter im Datenbankverzeichnis bzw. im Datenbankpfad befinden, müssen sie nicht in der Konfigurationsdatei aufgelistet werden. Wenn die restlichen beiden Behälter an andere Speicherpositionen auf dem neuen System versetzt werden sollen, müssen sie in der Konfigurationsdatei aufgelistet werden.

## Befehl "db2relocatedb" zum Verlagern einer Datenbank

Nachdem die physischen Verzeichnisse und Dateien an ihre neuen Speicherpositionen versetzt wurden, kann die folgende Konfigurationsdatei mit dem Befehl **db2relocatedb** verwendet werden, um Änderungen an den Datenbankdateien vorzunehmen, damit sie die neuen Speicherpositionen erkennen:

```
DB_NAME=TESTDB
DB_PATH=/databases/TESTDB,/DB2
INSTANCE=db2inst1,newinst
NODENUM=0
CONT_PATH=/Data/TS5_Cont0,/DB2/TESTDB/TS5_Cont0
CONT_PATH=/dev/rTS5_Cont1,/dev/rTESTDB_TS5_Cont1
```

### Beispiel 5

Die Datenbank TESTDB hat zwei Partitionen auf den Datenbankpartitionsservern 10 und 20. Das Exemplar ist 'servinst', und der Datenbankpfad auf beiden Datenbankpartitionsservern ist /home/servinst. Der Name der Datenbank wird in SERVDB und der Datenbankpfad auf beiden Datenbankpartitionsservern in /databases geändert. Außerdem wird das Protokollverzeichnis auf dem Datenbankpartitionsserver 20 von /testdb\_logdir in /servdb\_logdir geändert.

Weil an beiden Datenbankpartitionen Änderungen vorgenommen werden, muss für jede Datenbankpartition eine Konfigurationsdatei erstellt werden. Der Befehl **db2relocatedb** muss auf jedem Datenbankpartitionsserver mit der entsprechenden Konfigurationsdatei ausgeführt werden.

Auf Datenbankpartitionsserver 10 wird die folgende Konfigurationsdatei verwendet:

```
DB_NAME=TESTDB,SERVDB
DB_PATH=/home/servinst,/databases
INSTANCE=servinst
NODE_NUM=10
```

Auf Datenbankpartitionsserver 20 wird die folgende Konfigurationsdatei verwendet:

```
DB_NAME=TESTDB,SERVDB
DB_PATH=/home/servinst,/databases
INSTANCE=servinst
NODE_NUM=20
LOG_DIR=/testdb_logdir,/servdb_logdir
```

### Beispiel 6

Die Datenbank MAINDB ist im Exemplar maininst im Pfad /home/maininst vorhanden. Die Speicherposition von vier Tabellenbereichsbehältern muss wie folgt geändert werden:

```
/maininst_files/allconts/C0 muss nach /MAINDB/C0 versetzt werden
/maininst_files/allconts/C1 muss nach /MAINDB/C1 versetzt werden
/maininst_files/allconts/C2 muss nach /MAINDB/C2 versetzt werden
/maininst_files/allconts/C3 muss nach /MAINDB/C3 versetzt werden
```

Nachdem die physischen Verzeichnisse und Dateien an die neuen Speicherpositionen versetzt wurden, kann die folgende Konfigurationsdatei mit dem Befehl **db2relocatedb** verwendet werden, um die Änderungen an den Datenbankdateien vorzunehmen, damit sie die neuen Speicherpositionen erkennen.

## Befehl "db2relocatedb" zum Verlagern einer Datenbank

**Anmerkung:** An allen Behältern wird die gleiche Änderung vorgenommen. Das heißt, /maininst\_files/allconts/ wird durch /MAINDB/ ersetzt. Deshalb kann ein einziger Eintrag mit einem Platzhalterzeichen verwendet werden:

```
DB_NAME=MAINDB
DB_PATH=/home/maininst
INSTANCE=maininst
NODE_NUM=0
CONT_PATH=/maininst_files/allconts/*, /MAINDB/*
```

### Zugehörige Referenzen:

- „db2inidb - Initialize a Mirrored Database Command“ in *Command Reference*

---

## Einschränkungen für Begrenzer beim Versetzen von Daten

### Einschränkungen für Begrenzer:

Der Benutzer muss darauf achten, dass das gewählte Begrenzerzeichen nicht Teil der zu versetzenden Daten ist. Andernfalls können unvorhersehbare Fehler auftreten. Beim Versetzen von Daten gelten für Spalten-, Zeichenfolge-, DATALINK- und Dezimalzeichenbegrenzer die folgenden Einschränkungen:

- Begrenzer schließen sich gegenseitig aus.
- Unzulässige Begrenzer sind binäre Nullen, Zeilenvorschubzeichen, Rücklauf- oder Leerzeichen.
- Das Standarddezimalzeichen (.) kann nicht als Zeichenfolgebegrenzer verwendet werden.
- Die folgenden Zeichen werden durch eine ASCII- und EBCDIC-Codepage anders angegeben:
  - Das DBCS-Endezeichen (0x0F) und das DBCS-Startzeichen (0x0E) können nicht als Begrenzer für eine EBCDIC-MBCS-Datendatei verwendet werden.
  - Begrenzer für MBCS-, EUC- oder DBCS-Codepages dürfen nicht größer sein als 0x40. Hiervon ausgenommen ist das Standarddezimalzeichen für EBCDIC-MBCS-Daten (0x4b).
  - Standardbegrenzer für Datendateien in ASCII-Codepages oder EBCDIC-MBCS-Codepages sind:
    - " (0x22, doppeltes Anführungszeichen; Zeichenfolgebegrenzer)
    - , (0x2c, Komma; Spaltenbegrenzer)
  - Standardbegrenzer für Datendateien in EBCDIC-SBCS-Codepages sind:
    - " (0x7F, doppeltes Anführungszeichen; Zeichenfolgebegrenzer)
    - , (0x6B, Komma; Spaltenbegrenzer)
  - Das Standarddezimalzeichen für ASCII-Datendateien ist 0x2e (Punkt).
  - Das Standarddezimalzeichen für EBCDIC-Datendateien ist 0x4B (Punkt).
  - Wenn sich die Codepage des Servers von der des Clients unterscheidet, sollte die hexadezimale Darstellung von Begrenzern, die von den Standardwerten abweichen, angegeben werden. Beispiel:

```
db2 load from ... modified by charde10x0C colde1X1e ...
```

## Befehl "db2relocatedb" zum Verlagern einer Datenbank

Die folgenden Informationen zur Unterstützung für die Erkennung von doppelten Zeichenbegrenzern in DEL-Dateien gelten für die Dienstprogramme EXPORT, IMPORT und LOAD:

- Zeichenbegrenzer sind innerhalb der zeichenorientierten Felder einer DEL-Datei zulässig. Dies gilt für Felder des Typs CHAR, VARCHAR, LONG VARCHAR oder CLOB (außer, wenn lobsinfile angegeben ist). Jedes Paar von Zeichenbegrenzern, das sich zwischen den eigentlichen Zeichenbegrenzern befindet, wird in die Datenbank importiert oder geladen. Beispiel:

```
"What a "nice" day!"
```

wird importiert als:

```
What a "nice" day!
```

Bei einem Export gelten die Regeln umgekehrt. Beispiel:

```
I am 6" tall.
```

wird in eine DEL-Datei exportiert als:

```
"I am 6"" tall."
```

- In einer DBCS-Umgebung wird das Pipe-Zeichen (|) nicht als Zeichenbegrenzer unterstützt.

---

## Versetzen von Daten zwischen typisierten Tabellen

Mit den DB2<sup>®</sup>-Dienstprogrammen EXPORT und IMPORT können Daten in typisierte Tabellen bzw. aus typisierten Tabellen heraus versetzt werden. Typisierte Tabellen können sich in einer Hierarchie befinden. Das Versetzen von Daten zwischen Hierarchien kann Folgendes beinhalten:

- Versetzen von Daten aus einer Hierarchie in eine identische Hierarchie
- Versetzen von Daten aus einer Hierarchie in einen Teilbereich einer größeren Hierarchie
- Versetzen von Daten aus einem Teilbereich einer großen Hierarchie in eine separate Hierarchie

Die Option IMPORT CREATE ermöglicht es, sowohl die Tabellen- als auch die Typhierarchie zu erstellen.

Die Kennzeichnung von Typen innerhalb einer Hierarchie ist datenbankabhängig. Dies bedeutet, dass derselbe Typ in verschiedenen Datenbanken unterschiedliche Kennungen hat. Darum muss beim Versetzen von Daten zwischen diesen Datenbanken ein Typenabgleich erfolgen, um sicherzustellen, dass die Daten korrekt versetzt werden.

Bevor die einzelnen typisierten Zeilen bei einer Exportoperation herausgeschrieben werden, wird eine Kennung in einen Indexwert umgesetzt. Dieser Indexwert kann eine beliebige Zahl von 1 bis zur Anzahl der relevanten Typen in der Hierarchie sein. Indexwerte werden gebildet, indem beim Durchgang durch die Hierarchie in einer bestimmten Reihenfolge alle Typen durchnummeriert werden. Die Reihenfolge wird als "Traversierfolge" bezeichnet. Dabei erfolgt der Durchgang durch alle übergeordneten und untergeordneten Tabellen in der Hierarchie von oben nach unten und von links nach rechts. Die Traversierfolge ist wichtig beim Versetzen von Daten zwischen Tabellenhierarchien, weil dabei im Verhältnis zu anderen Daten ermittelt wird, wohin die Daten versetzt werden.

Eine Methode ist der Durchgang von der obersten Stufe der Hierarchie (oder der Stammtabelle) durch die untergeordneten Stufen (untergeordnete Tabellen) bis zur letzten untergeordneten Tabelle; anschließend zurück zu ihrer übergeordneten Tabelle, weiter durch die nächste(n) "unmittelbar rechts angrenzende(n)" untergeordnete(n) Tabelle(n); danach zurück zur nächsthöheren übergeordneten Tabelle, weiter durch deren untergeordnete Tabellen usw.

Die folgende Abbildung zeigt eine Hierarchie mit vier Traversierfolgen:

- Person, Employee, Manager, Architect, Student
- Person, Student, Employee, Manager, Architect (diese Traversierfolge ist mit einer gestrichelten Linie gekennzeichnet)
- Person, Employee, Architect, Manager, Student
- Person, Student, Employee, Architect, Manager

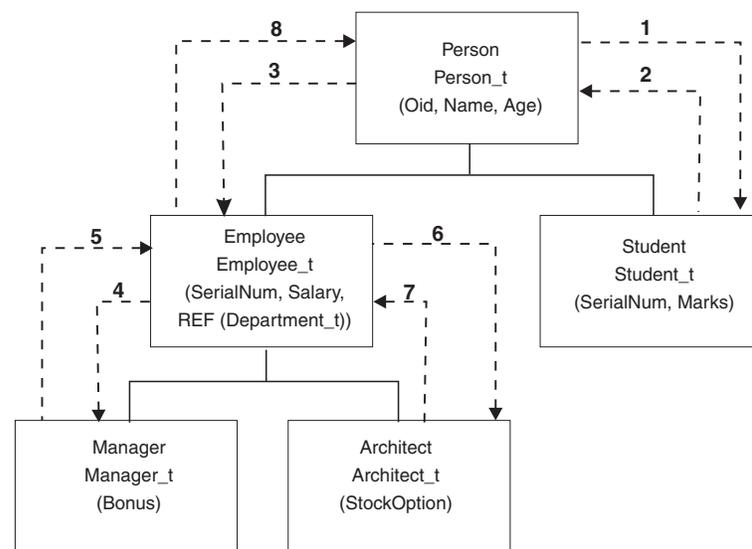


Abbildung 14.

#### Zugehörige Konzepte:

- „EXPORT - Übersicht“ auf Seite 1
- „IMPORT - Übersicht“ auf Seite 27

---

## Versetzen von Daten zwischen typisierten Tabellen - Details

### Traversierfolge

Es gibt eine Standardtraversierfolge, in der alle relevanten Typen ab einem bestimmten Startpunkt in der Hierarchie auf alle erreichbaren Typen der Hierarchie verweisen. Die Standardreihenfolge schließt alle Tabellen in der Hierarchie ein, und jede Tabelle wird nach dem Schema sortiert, das im erweiterten Sortiervergleichselement verwendet wird. Außerdem gibt es eine benutzerdefinierte Traversierfolge, bei der die zu verwendenden relevanten Typen vom Benutzer in einer Traversierfolgeliste definiert werden. Beim Arbeiten mit den Dienstprogrammen EXPORT und IMPORT muss dieselbe Traversierfolge verwendet werden.

Beachten Sie beim Angeben der Traversierfolge, dass der Durchgang durch die untergeordneten Tabellen der vorgegebenen Anordnung (PRE-ORDER) folgen

muss (d. h. ein neuer Zweig der hierarchischen Struktur wird erst angefangen, nachdem der vorangegangene Zweig bis zum Ende abgearbeitet ist).

### **Standardtraversierfolge**

Die Standardtraversierfolge verhält sich bei verschiedenen Dateiformaten unterschiedlich. In den nachfolgend dargestellten Fällen wird vorausgesetzt, dass identische Tabellenhierarchien und Typenbeziehungen vorliegen.

Beim Export von Daten in das PC/IXF-Dateiformat wird ein Datensatz mit allen relevanten Typen, ihren Definitionen und den relevanten Tabellen erzeugt. EXPORT nimmt außerdem die Zuordnung von Indexwerten für die einzelnen Tabellen vor. Beim Import wird anhand dieser Zuordnung sichergestellt, dass die Daten korrekt in die Zieldatenbank versetzt werden. Beim Arbeiten mit dem PC/IXF-Dateiformat ist es ratsam, die Standardtraversierfolge zu verwenden.

Beim Arbeiten mit dem ASC-, DEL- und WSF-Dateiformat kann die Erstellungsreihenfolge der typisierten Zeilen und typisierten Tabellen unterschiedlich sein, auch wenn die Quellen- und Zielhierarchien möglicherweise dieselbe Struktur aufweisen. Dies würde zu Zeitdifferenzen führen, die die Standardtraversierfolge beim Durchgang durch die Hierarchien feststellen würde. Die Erstellungszeit jedes Typs bestimmt, in welcher Reihenfolge der Durchgang durch die Quellen- und die Zielhierarchie erfolgt, wenn die Standardtraversierfolge verwendet wird. Sie müssen sicherstellen, dass sowohl die Reihenfolge der Erstellung der einzelnen Typen in Quellen- und Zielhierarchie als auch die hierarchische Struktur von Quelle und Ziel identisch sind. Wählen Sie eine benutzerdefinierte Standardtraversierfolge aus, wenn diese Bedingungen nicht erfüllt werden können.

### **Benutzerdefinierte Traversierfolge**

Wenn Sie die Traversierfolge für die Hierarchien steuern möchten, müssen Sie sicherstellen, dass für die Dienstprogramme EXPORT und IMPORT die gleiche Traversierfolge verwendet wird. Angenommen, die folgenden Bedingungen sind erfüllt:

- Eine identische Definition von untergeordneten Tabellen in den Quellen- und Zieldatenbanken
- Eine identische hierarchische Beziehung zwischen untergeordneten Tabellen in den Quellen- und Zieldatenbanken
- Eine identische Traversierfolge

In diesem Fall ist die exakte Versetzung von Daten in die Zieldatenbank durch das Dienstprogramm IMPORT sichergestellt.

Beim Definieren der Traversierfolge legen Sie zwar den Startpunkt und den Pfad durch die Hierarchie fest, aber jede Verzweigung der Hierarchie muss zunächst komplett verarbeitet werden, bevor mit der nächsten Verzweigung begonnen werden kann. Die Dienstprogramme EXPORT und IMPORT erkennen die Nichteinhaltung dieser Bedingung innerhalb der angegebenen Traversierfolge und melden gegebenenfalls einen Fehler.

#### **Zugehörige Referenzen:**

- „DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer)“ auf Seite 276
- „ASC-Dateiformat (ASCII-Format mit universellen Zeilenbegrenzern)“ auf Seite 281
- „PC-Version des IXF-Dateiformats“ auf Seite 285
- „WSF-Dateiformat (Arbeitsblattformat)“ auf Seite 329

## Datenauswahl beim Versetzen von Daten

Das Versetzen von Daten aus einer hierarchischen Struktur mit typisierten Tabellen in eine andere erfolgt entsprechend einer bestimmten Traversierfolge und durch Erstellung einer temporären Flachdatei. Über das Dienstprogramm EXPORT (in Kombination mit der Traversierfolge) kann gesteuert werden, welche Daten aus der Quelldatenbank in die temporäre Datei übernommen werden. Sie brauchen nur den Namen der Zieltabelle zusammen mit der Klausel WHERE anzugeben. Das Dienstprogramm EXPORT verwendet diese Auswahlkriterien, um eine entsprechende temporäre Datei zu erstellen.

Das Dienstprogramm IMPORT steuert, was in die Zieltabelle gestellt wird. Sie können, falls gewünscht, am Ende jedes Namens einer untergeordneten Tabelle eine Attributliste angeben, um festzulegen, dass nur bestimmte Attribute in die Zieldatenbank versetzt werden sollen. Wenn keine Attributliste angegeben ist, werden alle Spalten der jeweiligen untergeordneten Tabelle versetzt.

Das Dienstprogramm IMPORT steuert Größe und Position der zu versetzenden Hierarchie mit den Parametern CREATE, INTO tabellenname, UNDER und AS ROOT TABLE.

### Zugehörige Referenzen:

- „Befehl IMPORT“ auf Seite 39

## Beispiele zum Versetzen von Daten zwischen typisierten Tabellen

Die Beispiele in diesem Abschnitt basieren auf der folgenden hierarchischen Struktur:

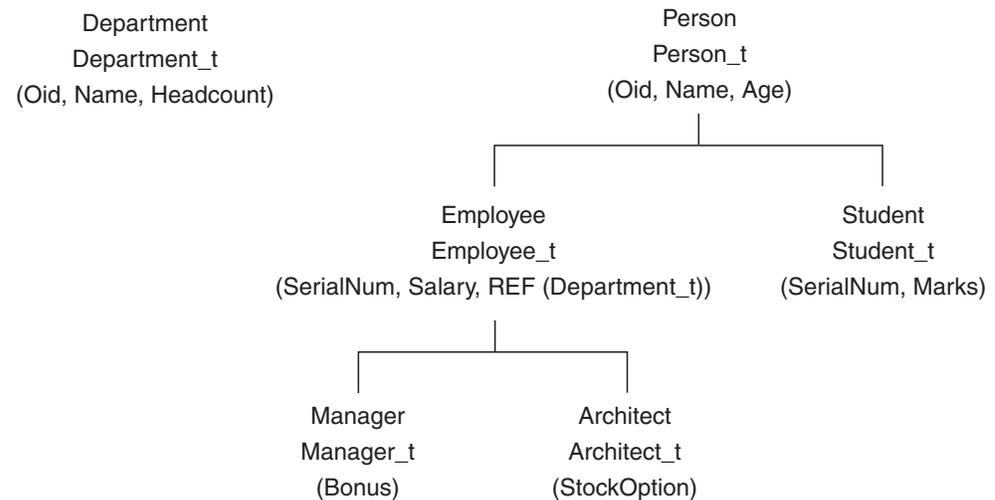


Abbildung 15.

## Beispiel 1

Importieren einer vollständigen Hierarchie und erneutes Erstellen dieser Hierarchie durch eine Importoperation:

```
DB2® CONNECT TO Source_db
DB2 EXPORT TO entire_hierarchy.ixf OF IXF HIERARCHY STARTING Person
DB2 CONNECT TO Target_db
DB2 IMPORT FROM entire_hierarchy.ixf OF IXF CREATE INTO
HIERARCHY STARTING Person AS ROOT TABLE
```

Alle in der Hierarchie enthaltenen Typen werden erstellt, wenn sie noch nicht vorhanden sind. Wenn diese Typen bereits vorhanden sind, müssen sie in der Zieldatenbank die gleichen Definitionen aufweisen wie in der Quelldatenbank. Wenn die Definitionen nicht identisch sind, wird ein SQL-Fehler (SQL20013N) zurückgegeben. Da eine neue Hierarchie erstellt wird, gilt außerdem, dass keine der untergeordneten Tabellen vorhanden sein darf, die in der in die Zieldatenbank (Target\_db) zu versetzenden Datendatei definiert sind. Alle angegebenen Tabellen aus der Hierarchie der Quelldatenbank werden erstellt. Zuletzt werden die Daten aus der Quelldatenbank in die richtigen untergeordneten Tabellen der Zieldatenbank importiert.

## Beispiel 2

Anhand eines etwas komplexeren Beispiels wird gezeigt, wie die gesamte Hierarchie der Quelldatenbank exportiert und anschließend in die Zieldatenbank importiert wird. Zwar werden alle Daten zu Personen, die über 20 Jahre alt sind, aus der Quelldatenbank exportiert, doch werden nur bestimmte ausgewählte Daten in die Zieldatenbank importiert.

```
DB2 CONNECT TO Source_db
DB2 EXPORT TO entire_hierarchy.del OF DEL HIERARCHY (Person,
Employee, Manager, Architect, Student) WHERE Age>=20
DB2 CONNECT TO Target_db
DB2 IMPORT FROM entire_hierarchy.del OF DEL INSERT INTO (Person,
Employee(Salary), Architect) IN HIERARCHY (Person, Employee,
Manager, Architect, Student)
```

Die Zieltabellen Person, Employee und Architect müssen vorhanden sein. Daten werden in die untergeordneten Tabellen Person, Employee und Architect importiert. Es wird also Folgendes importiert:

- Alle Spalten aus der Tabelle Person in die Tabelle Person
- Alle Spalten aus der Tabelle Person und die Spalte Salary aus der Tabelle Employee in die Tabelle Employee
- Alle Spalten aus der Tabelle Person und die Spalte Salary aus der Tabelle Employee sowie alle Spalten aus der Tabelle Architect in die Tabelle Architect

Die Spalten SerialNum und REF(Employee\_t) werden weder in die Tabelle Employee noch in die dazugehörigen untergeordneten Tabellen importiert (hier ist Architect die einzige untergeordnete Tabelle, in die Daten importiert werden).

**Anmerkung:** Da Architect eine untergeordnete Tabelle von Employee ist und als einzige Importspalte für Employee die Spalte Salary angegeben ist, wird Salary als einzige spezifische Spalte der Tabelle Employee in die Tabelle Architect importiert. Dies bedeutet, dass weder die Spalte SerialNum noch die Spalte REF(Employee\_t) in die Zeilen Employee oder Architect importiert werden.

Es werden keine Daten in die Tabellen Manager und Student importiert.

### Beispiel 3

Dieses Beispiel zeigt, wie Daten aus einer regulären Tabelle exportiert und als eine einzige untergeordnete Tabelle in eine Hierarchie importiert werden. Der Befehl EXPORT verarbeitet eine reguläre (nicht typisierte) Tabelle, d. h. die Datendatei enthält keine Spalte Type\_id. Der Änderungswert no\_type\_id gibt diese Information an das Dienstprogramm IMPORT weiter, damit es nicht als erste Spalte die Spalte Type\_id erwartet.

```
DB2 CONNECT TO Source_db
DB2 EXPORT TO Student_sub_table.del OF DEL SELECT * FROM
Regular_Student
DB2 CONNECT TO Target_db
DB2 IMPORT FROM Student_sub_table.del OF DEL METHOD P(1,2,3,5,4)
MODIFIED BY NO_TYPE_ID INSERT INTO HIERARCHY (Student)
```

In diesem Beispiel muss die Zieltabelle Student vorhanden sein. Da es sich bei Student um eine untergeordnete Tabelle handelt, wird durch den Änderungswert no\_type\_id angegeben, dass die erste Spalte keine Type\_id enthält. Sie müssen jedoch sicherstellen, dass die Tabelle Student zusätzlich zu allen anderen Attributen der Tabelle eine Spalte Object\_id enthält. Außerdem wird Object\_id als erste Spalte in jeder Zeile erwartet, die in die Tabelle Student importiert wird. Die Klausel METHOD kehrt die Reihenfolge der beiden letzten Attribute um.

#### Zugehörige Konzepte:

- „Versetzen von Daten zwischen typisierten Tabellen“ auf Seite 248

---

## Verwenden der Replikation zum Versetzen von Daten

Die Replikation ermöglicht es Ihnen, Daten regelmäßig auf mehrere ferne Datenbanken zu kopieren. Wenn Sie Aktualisierungen einer Hauptdatenbank automatisch auf andere Datenbanken kopieren müssen, können Sie die DB2<sup>®</sup>-Replikationskomponenten verwenden, um anzugeben, welche Daten kopiert werden sollen, in welche Datenbanktabellen die Daten zu kopieren sind und wie oft die Aktualisierungen kopiert werden sollen. Die DB2-Replikationskomponenten sind Teil einer größeren IBM<sup>®</sup> Lösung für die Replikation von Daten in kleinen und großen Unternehmen.

Die IBM Replikationstools sind eine Gruppe von DB2 DataPropagator<sup>™</sup>-Programmen und DB2 Universal Database<sup>™</sup>-Tools, die Daten zwischen Verwaltungssystemen für verteilte relationale Datenbanken kopieren:

- Zwischen DB2 Universal Database-Plattformen
- Zwischen DB2 Universal Database-Plattformen und Host-Datenbanken, die die DRDA-Konnektivität (Distributed Relational Database Architecture<sup>™</sup> - Architektur einer verteilten relationalen Datenbank) unterstützen
- Zwischen Host-Datenbanken, die die DRDA<sup>®</sup>-Konnektivität unterstützen

Mit Hilfe von DB2 DataJoiner<sup>®</sup> können Daten auch auf Verwaltungssysteme für relationale Datenbanken von anderen Herstellern als IBM repliziert werden.

Sie können die IBM Replikationstools verwenden, um Kopieroperationen von einem einzigen Steuerpunkt aus für Daten im gesamten Unternehmen zu definieren, zu synchronisieren, zu automatisieren und zu verwalten. Die Replikationstools in DB2 Universal Database ermöglichen eine Replikation zwischen relationalen Datenbanken. Außerdem arbeiten sie mit IMS<sup>™</sup> DataPropagator (vormals

DPropNR) zusammen, um IMS- und VSAM-Daten zu replizieren, sowie mit Lotus® NotesPump, um Replikationen an und von Lotus Notes®-Datenbanken vorzunehmen.

Die Replikation versetzt Sie in die Lage, Endbenutzern und Anwendungen den Zugriff auf Produktionsdaten zu ermöglichen, ohne die Produktionsdatenbank weiter zu belasten. Sie können die Daten auf eine Datenbank kopieren, die für einen Benutzer oder eine Anwendung lokal ist, so dass sie nicht fern auf die Daten zugreifen müssen. Ein typisches Replikationsszenario wäre eine Quellentabelle mit Kopien in mindestens einer fernen Datenbank, wie es zum Beispiel bei der Zentrale einer Bank und ihren Zweigstellen vorstellbar ist. Zu vordefinierten Zeiten erfolgen automatische Aktualisierungen der DB2-Datenbanken, und alle Änderungen an der Quelldatenbank werden in die Zieldatenbanktabellen kopiert.

Die Replikationstools ermöglichen es Ihnen, die Struktur der Kopiertabelle anzupassen. Sie können beim Kopieren in die Zieldatenbank SQL verwenden, um die zu kopierenden Daten auszuwählen, zu bearbeiten oder auf andere Weise aufzubereiten. Sie können Kopien für den Lesezugriff erstellen, die die Quellentabelle duplizieren, und haben die Möglichkeit, Daten zu einem bestimmten Zeitpunkt zu erfassen, ein Protokoll der Änderungen bereitzustellen oder Daten, die zu weiteren Zieltabellen kopiert werden sollen, zwischenspeichern. Darüber hinaus können Sie Kopien für den Schreib-/Lesezugriff erstellen, die von Endbenutzern oder Anwendungen aktualisiert werden können und deren Änderungen zurück in die Haupttabelle repliziert werden. Sie können Sichten von Quellentabellen und Sichten von Kopien replizieren. Ereignisgesteuerte Replikation ist ebenfalls möglich.

Die Replikation von Daten zwischen DB2-Datenbanken ist auf den folgenden Plattformen möglich: AIX®, AS/400®, HP-UX, Linux, Windows®, OS/390®, SCO UnixWare, Solaris™ Operating Environment, Sequent®, VM und VSE. Außerdem können (mit DB2 DataJoiner) Daten zwischen DB2 und den folgenden anderen Datenbankprodukten repliziert werden: Informix®, Microsoft® Jet, Microsoft SQL Server, Oracle®, Sybase und Sybase SQLAnywhere. In Kombination mit anderen IBM Produkten können Sie DB2-Daten auf bzw. von IMS, VSAM oder Lotus Notes replizieren. Schließlich können Sie Daten ferner an DB2 Everywhere auf Einheiten mit dem Windows CE- oder Palm-Betriebssystem replizieren.

#### **Zugehörige Konzepte:**

- „IBM Replikationstools nach Komponenten“ auf Seite 254

---

## **IBM Replikationstools**

### **IBM Replikationstools nach Komponenten**

Die IBM® Replikationstool-Lösung setzt sich aus zwei Komponenten zusammen: dem Capture-Programm und dem Apply-Programm. Sie können diese Komponenten mit der DB2®-Steuerzentrale einrichten. Der Betrieb und die Überwachung dieser Komponenten erfolgt außerhalb der Steuerzentrale.

Das Capture-Programm erfasst Änderungen, die an den Quellentabellen vorgenommen wurden. Eine Quellentabelle kann Folgendes sein:

- Eine externe Tabelle, die SQL-Daten aus einem Dateisystem oder einem nichtrelationalen Datenbankmanager enthält, die außerhalb von DB2 DataPropagator™ geladen werden.
- Eine vorhandene Tabelle in der Datenbank.

- Eine Tabelle, die zuvor vom Apply-Programm aktualisiert wurde, das das Kopieren von Änderungen zurück in die Quellentabelle oder in andere Zieltabellen ermöglicht.

Die Änderungen werden in eine CD-Tabelle kopiert, in der sie gespeichert werden, bis das Zielsystem bereit ist, sie zu kopieren. Anschließend ruft das Programm DPROPR Apply die Änderungen aus der CD-Tabelle ab und kopiert sie in die Zieltabellen.

Verwenden Sie die Steuerzentrale für folgende Zwecke:

- Einrichten der Replikationsumgebung
- Definieren von Quellen- und Zieltabellen
- Festlegen des zeitlichen Ablaufs der automatisierten Kopiervorgänge
- Angeben von SQL-Erweiterungen für die Daten
- Definieren von Beziehungen zwischen den Quellen- und Zieltabellen

#### Zugehörige Tasks:

- „Planen einer SQL Replication-Umgebung“ in *IBM DB2 Information Integrator SQL Replication Handbuch und Referenz*

---

## Versetzen von Daten mit der Data Warehouse-Zentrale

Mit Hilfe der Data Warehouse-Zentrale können Daten aus Betriebsdatenbanken in eine Warehouse-Datenbank versetzt werden, die von Benutzern als Entscheidungshilfe abgefragt werden kann. Außerdem können Sie die Data Warehouse-Zentrale verwenden, um die Struktur der Betriebsdatenbanken, den so genannten *Quellen* zu definieren. Anschließend können Sie angeben, wie die Betriebsdaten versetzt und für das Warehouse umgesetzt werden sollen. Sie können die Struktur der Tabellen in der Warehouse-Datenbank (die so genannten *Ziele*) modellieren oder die Tabellen automatisch im Rahmen des Definitionsprozesses für die Operationen der Datenversetzung erstellen.

Die Data Warehouse-Zentrale setzt die folgenden DB2<sup>®</sup>-Funktionen zum Versetzen und Umsetzen von Daten ein:

- SQL

Mit Hilfe von SQL können Sie Daten aus Quellen auswählen und die Daten in Ziele einfügen. Außerdem können Sie SQL zum Umsetzen der Daten in das entsprechende Warehouse-Format verwenden. SQL können Sie mit der Data Warehouse-Zentrale generieren. Alternativ können Sie auch Ihr eigenes SQL schreiben.

- Dienstprogramme LOAD und EXPORT

Mit diesen DB2-Dienstprogrammen können Sie Daten aus einer Quelle exportieren und die Daten anschließend in ein Ziel laden. Diese Dienstprogramme sind nützlich, wenn Sie große Datenmengen versetzen müssen. Die Data Warehouse-Zentrale unterstützt die folgenden Arten von Lade- und Exportoperationen:

#### **DB2 Daten exportieren**

Exportiert Daten aus einer lokalen DB2-Datenbank in eine begrenzte Datei.

#### **Daten mit ODBC exportieren**

Wählt Daten aus einer Tabelle in einer Datenbank, die in ODBC registriert ist, aus und schreibt die Daten anschließend in eine begrenzte Datei.

### **DB2 Laden**

Lädt Daten aus einer begrenzten Datei in eine DB2-Tabelle.

### **DB2 Laden in DB2 UDB ESE-Datenbank (nur AIX)**

Lädt Daten aus einer begrenzten Datei in eine Datenbank von DB2 Universal Database™ Enterprise Server Edition und ersetzt hierbei in einer Tabelle vorhandene Daten durch neue Daten. Diese Operation fordert die Zielpartitionierungszuordnung für die Datenbank an, partitioniert die Eingabedatei so, dass jede Datei auf eine Datenbankpartition geladen werden kann, und führt anschließend für alle Partitionen eine ferne Ladeoperation durch.

- **Replikation**

Zum Kopieren großer Datenmengen von Warehouse-Quellen in ein Warehouse-Ziel können Sie auch die Replikation einsetzen und anschließend alle nachfolgenden Änderungen der Quelldaten erfassen. Die Data Warehouse-Zentrale unterstützt die folgenden Replikationsarten:

#### **Basisergebnistabelle**

Erstellt eine Zieltabelle, die Ergebnisdaten für eine Benutzertabelle enthält und in bestimmten Intervallen angehängt wird.

#### **CA-Tabelle**

Erstellt eine Zieltabelle, die Ergebnisdaten enthält und auf den Änderungen basiert, die für eine Quellentabelle aufgezeichnet werden.

#### **Zeitpunkt**

Erstellt eine Zieltabelle, die mit der Quellentabelle identisch ist, und fügt der Zieltabelle eine Zeitmarkenspalte hinzu.

#### **Zwischenspeichertabelle**

Erstellt eine CDD-Tabelle, die als Quelle für aktualisierte Daten, die für mehrere Zieltabellen bestimmt sind, verwendet werden kann.

#### **Benutzerkopie**

Erstellt eine Zieltabelle, die mit der Quellentabelle zum Zeitpunkt der Kopiererstellung identisch ist.

Diese Operationen werden in jeder Workstation-Betriebsumgebung von DB2 Universal Database - DB2 Universal Database für OS/390®, DB2 für AS/400® und DataJoiner® - unterstützt.

- **Umsetzungsprogramm für gespeicherte Prozeduren**

Mit der Data Warehouse-Zentrale können Sie Daten in eine OLAP-Datenbank (Online Analytical Processing - Online-Analyseverarbeitung) versetzen. Nachdem die Daten in das Warehouse versetzt wurden, können Sie das Umsetzungsprogramm für gespeicherte Prozeduren verwenden, um die Daten zu bereinigen und anschließend in Fakt- und Dimensionstabellen zusammenfassen. Außerdem können Sie mit den Umsetzungsprogrammen statistische Daten generieren. Nachdem die Daten bereinigt und umgesetzt wurden, können Sie sie in OLAP-Kuben laden oder auf Abteilungs-Server replizieren, die manchmal auch als *Datamarts* bezeichnet werden. Die Umsetzungsprogramme sind nicht in allen DB2-Angeboten enthalten. Weitere Informationen erhalten Sie bei Ihrem IBM® Ansprechpartner.

#### **Zugehörige Konzepte:**

- „Lösungen durch Data Warehousing“ in *Data Warehouse-Zentrale Verwaltung*

---

## Versetzen von Daten mit Dateityp CURSOR

Durch die Angabe des Dateityps CURSOR bei Verwendung des Befehls LOAD können Sie die Ergebnisse einer SQL-Abfrage direkt in eine Zieltabelle laden, ohne eine temporäre exportierte Datei zu erstellen. Durch einen Verweis auf einen Kurznamen in der SQL-Abfrage kann das Dienstprogramm LOAD auch Daten aus einer anderen Datenbank in einem einzigen Schritt laden.

Damit eine LOAD FROM CURSOR-Operation über den Befehlszeilenprozessor (CLP) möglich ist, muss zunächst für eine SQL-Abfrage ein Cursor deklariert werden. Sobald dies ausgeführt wurde, können Sie den Befehl LOAD absetzen. Hierbei verwenden Sie den Namen des deklarierten Cursors als Wert für *cursorname* und die Angabe CURSOR für den Dateityp.

Beispiel:

Die Tabelle ABC.TABLE1 enthält 3 Spalten:

- ONE INT
- TWO CHAR(10)
- THREE DATE

Die Tabelle ABC.TABLE2 enthält 3 Spalten:

- ONE VARCHAR
- TWO INT
- THREE DATE

Durch Ausführung der folgenden CLP-Befehle werden alle Daten aus der Tabelle ABC.TABLE1 in die Tabelle ABC.TABLE2 geladen:

```
DECLARE mycurs CURSOR FOR SELECT TWO,ONE,THREE FROM abc.table1
LOAD FROM mycurs OF cursor INSERT INTO abc.table2
```

### Anmerkungen:

1. Das vorstehende Beispiel zeigt, wie Daten aus einer SQL-Abfrage über den Befehlszeilenprozessor geladen werden. Das Laden aus einer SQL-Abfrage kann jedoch auch über die API **db2Load** erfolgen, indem die Werte *piSourceList* und *piFileType* der *db2LoadStruct*-Struktur entsprechend definiert werden.
2. Wie oben dargestellt müssen die Quellenspalentypen der SQL-Abfrage nicht mit den entsprechenden Zielspalentypen identisch sein. Kompatibilität zwischen den Spalentypen ist jedoch erforderlich.

### Zugehörige Konzepte:

- „Kurznamen und Datenquellenobjekte“ in *Systeme zusammenschlossener Datenbanken*

### Zugehörige Referenzen:

- „Befehl LOAD“ auf Seite 112
- „Assignments and comparisons“ in *SQL Reference, Volume 1*



---

## Anhang A. Lesen von Syntaxdiagrammen

Ein Syntaxdiagramm zeigt, wie ein Befehl angegeben werden muss, damit das Betriebssystem die Eingabe richtig auswerten kann.

Lesen Sie ein Syntaxdiagramm von links nach rechts und von oben nach unten. Folgen Sie dabei der horizontalen Linie (dem Hauptpfad). Wenn eine Linie mit einer Pfeilspitze endet, wird die Befehlssyntax fortgesetzt, und die nächste Linie beginnt mit einer Pfeilspitze. Ein vertikaler Balken kennzeichnet das Ende der Befehlssyntax.

Bei der Eingabe von Daten eines Syntaxdiagramms müssen Sie auch Zeichen wie Anführungszeichen und Gleichheitszeichen eingeben.

Parameter sind als Schlüsselwörter oder Variablen klassifiziert:

- Schlüsselwörter sind Konstanten und werden in Großbuchstaben dargestellt. An der Eingabeaufforderung können sie jedoch in beliebiger Schreibweise eingegeben werden. Ein Befehlsname ist ein Beispiel für ein Schlüsselwort.
- Variablen sind Namen oder Werte, die vom Benutzer angegeben werden, und werden in Kleinbuchstaben dargestellt. An der Eingabeaufforderung können sie jedoch in beliebiger Schreibweise eingegeben werden, sofern dies nicht ausdrücklich anders angegeben ist. Ein Dateiname ist ein Beispiel für eine Variable.

Ein Parameter kann eine Kombination aus einem Schlüsselwort und einer Variablen sein.

Erforderliche Parameter werden im Hauptpfad angegeben:

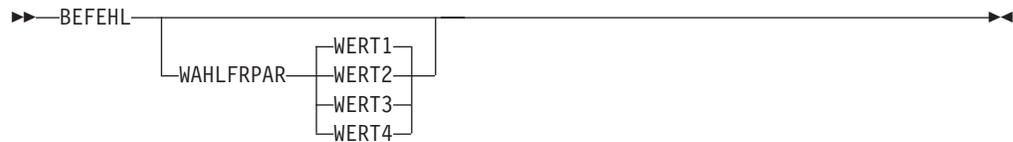
►►—BEFEHL—*erforderlicher-parameter*—►►

Wahlfreie Parameter werden unterhalb des Hauptpfads angegeben:

►►—BEFEHL—  
└─*wahlfreier-parameter*—┘—►►

## Lesen von Syntaxdiagrammen

Der Standardwert eines Parameters wird oberhalb des Pfad angegeben:



Ein Stapel von Parametern (erster Parameter im Hauptpfad angegeben) gibt an, dass einer der Parameter gewählt werden muss:

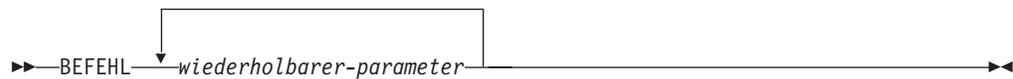


Ein Stapel von Parametern, bei dem der erste Parameter unterhalb des Hauptpfads angegeben ist, gibt an, dass einer der Parameter gewählt werden kann:



Ein nach links zurückweisender Pfeil oberhalb des Pfads gibt an, dass Elemente gemäß den folgenden Konventionen wiederholt werden können:

- Wenn der Pfeil durchgängig ist, kann das Element in einer Liste wiederholt werden, wobei die Elemente durch Leerstellen zu trennen sind:

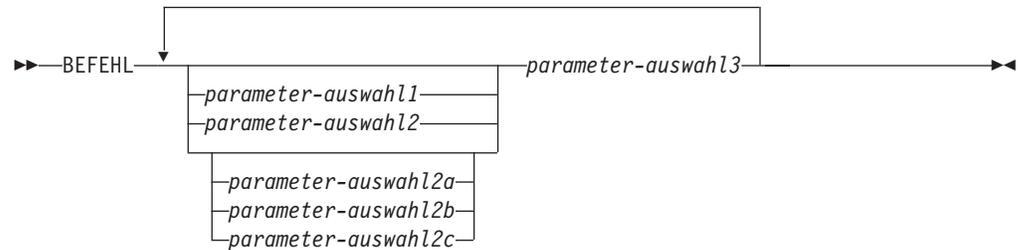


- Wenn der Pfeil ein Komma enthält, kann das Element in einer Liste wiederholt werden, wobei die Elemente durch Kommas zu trennen sind:



Elemente aus Parameterstapeln können gemäß den Stapelkonventionen für die zuvor erläuterten erforderlichen und wahlfreien Parameter wiederholt werden.

Manche Syntaxdiagramme enthalten Parameterstapel innerhalb von anderen Parameterstapeln. Elemente aus Stapeln können nur gemäß den zuvor erläuterten Konventionen wiederholt werden. Das heißt, wenn über einem inneren Stapel kein nach links zurückweisender Pfeil ist, über einem äußeren Stapel jedoch schon, kann nur ein einziger Parameter aus dem inneren Stapel gewählt und mit einem beliebigen Parameter des äußeren Stapels kombiniert werden; diese Kombination kann dann wiederholt werden. Das folgende Diagramm zeigt beispielsweise, dass der Parameter *parameter-auswahl2a* mit dem Parameter *parameter-auswahl2* kombiniert und diese Kombination dann wiederholt werden kann (*parameter-auswahl2* plus *parameter-auswahl2a*):

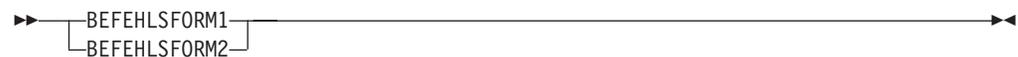


Einigen Befehlen ist ein wahlfreier Pfadparameter vorangestellt:



Wenn dieser Parameter nicht angegeben ist, sucht das System den Befehl im aktuellen Verzeichnis. Wenn der Befehl nicht gefunden wird, sucht das System weiterhin in allen Verzeichnissen in den Pfaden, die in der Datei `.profile` bzw. in der Anweisung `PATH` angegeben sind.

Einige Befehle weisen syntaktische Varianten auf, die funktionell identisch sind:



## Lesen von Syntaxdiagrammen

## Anhang B. Unterschiede zwischen den Dienstprogrammen IMPORT und LOAD

Die folgende Tabelle fasst wichtige Unterschiede zwischen den DB2-Dienstprogrammen LOAD und IMPORT zusammen.

Dienstprogramm IMPORT	Dienstprogramm LOAD
Langsam beim Versetzen von großen Datenmengen.	Schneller als das Dienstprogramm IMPORT beim Versetzen großer Datenmengen, da das Dienstprogramm LOAD formatierte Seiten direkt in die Datenbank schreibt.
Partitionsinterne Parallelität wird beschränkt ausgenutzt.	Partitionsinterne Parallelität wird ausgenutzt. Dazu sind in der Regel symmetrische Mehrprozessormaschinen (SMP-Maschinen) erforderlich.
FASTPARSE wird nicht unterstützt.	FASTPARSE wird unterstützt und bietet eine reduzierte Datenprüfung der vom Benutzer bereitgestellten Daten.
Hierarchische Daten werden unterstützt.	Hierarchische Daten werden nicht unterstützt.
Das Erstellen von Tabellen, Hierarchien und Indizes wird beim PC/IXF-Format unterstützt.	Tabellen und Indizes müssen vorhanden sein.
Das Importieren von Daten in gespeicherte Abfragetabellen wird nicht unterstützt.	Das Laden von Daten in gespeicherte Abfragetabellen wird unterstützt.
WSF-Format wird unterstützt.	WSF-Format wird nicht unterstützt.
BINARYNUMERICS wird nicht unterstützt.	BINARYNUMERICS wird unterstützt.
PACKEDDECIMAL wird nicht unterstützt.	PACKEDDECIMAL wird unterstützt.
ZONEDDECIMAL wird nicht unterstützt.	ZONEDDECIMAL wird unterstützt.
Das Überschreiben von Spalten, die als GENERATED ALWAYS definiert sind, ist nicht möglich.	Das Überschreiben von Spalten, die als GENERATED ALWAYS definiert sind, ist durch Verwendung der Änderungswerte GENERATEDOVERRIDE und IDENTITYOVERRIDE für den Dateityp möglich.
Das Importieren von Daten in Tabellen und Sichten wird unterstützt.	Daten können nur in Tabellen geladen werden.
Alle Zeilen werden protokolliert.	Es erfolgt nur eine minimale Protokollierung.
Auslöser werden unterstützt.	Auslöser werden nicht unterstützt.
Wenn die Importoperation unterbrochen wird und ein Wert für <i>commitcount</i> angegeben wurde, ist die Tabelle verwendbar und enthält die Zeilen, die bis zur letzten COMMIT-Operation geladen wurden. Der Benutzer kann dann die Importoperation erneut starten oder die Tabelle so akzeptieren, wie sie vorliegt.	Wenn die Ladeoperation unterbrochen wird und ein Wert für <i>savecount</i> angegeben wurde, verbleibt die Tabelle im Status "Laden anstehend" und kann nicht verwendet werden, bis die Ladeoperation erneut gestartet und fortgesetzt wird oder der Tabellenbereich anhand eines vor der Ladeoperation erstellten Sicherungsabbilds wiederhergestellt wird.

Dienstprogramm IMPORT	Dienstprogramm LOAD
Der erforderliche Speicherbereich entspricht in etwa dem größten Index plus 10%. Dieser Speicherbereich wird den temporären Tabellenbereichen in der Datenbank entnommen.	Der erforderliche Speicherbereich entspricht in etwa der Summe der Größen aller Indizes, die für die Tabelle definiert sind, und kann u. U. bis doppelt so groß werden. Dieser Speicherbereich wird dem temporären Speicherplatz in der Datenbank entnommen.
Alle Integritätsbedingungen werden während der Importoperation ausgewertet.	Das Dienstprogramm LOAD überprüft die Eindeutigkeit und berechnet Werte für generierte Spalten. Alle anderen Integritätsbedingungen müssen jedoch mit der Anweisung SET INTEGRITY überprüft werden.
Die Schlüsselwerte werden während einer Importoperation einzeln in den Index eingefügt.	Nach dem Laden der Daten werden die Schlüsselwerte sortiert, und der Index wird erstellt.
Wenn aktualisierte Statistikdaten erforderlich sind, muss das Dienstprogramm RUNSTATS nach einer Importoperation ausgeführt werden.	Statistikdaten können während der Ladeoperation zusammengestellt werden, wenn die Daten in der Tabelle ersetzt (REPLACE) werden.
Über DB2 Connect können Sie Daten in eine Host-Datenbank importieren.	Es ist nicht möglich, Daten in eine Host-Datenbank zu laden.
Importdateien müssen sich auf dem Knoten befinden, auf dem das Dienstprogramm IMPORT aufgerufen wird.	In einer Umgebung mit partitionierten Datenbanken müssen sich die zu ladenden Dateien oder Pipes auf dem Knoten befinden, der die Datenbank enthält. In einer Umgebung ohne partitionierte Datenbanken können sich die zu ladenden Dateien oder Pipes auf dem Knoten befinden, der die Datenbank enthält, oder auf dem Client mit Fernverbindung, von dem aus das Dienstprogramm LOAD aufgerufen wurde.
Ein Sicherungsabbild ist nicht erforderlich. Da vom Dienstprogramm IMPORT SQL-INSERT-Anweisungen verwendet werden, protokolliert DB2 die Aktivität, und es sind keine Sicherungen erforderlich, um diese Operationen bei Fehlern wiederherzustellen.	Ein Sicherungsabbild kann während der Ladeoperation erstellt werden.

#### Zugehörige Konzepte:

- „IMPORT - Übersicht“ auf Seite 27
- „LOAD - Übersicht“ auf Seite 82

#### Zugehörige Referenzen:

- „Befehl IMPORT“ auf Seite 39
- „Befehl LOAD“ auf Seite 112

---

## Anhang C. API-Beispielprogramm für EXPORT-, IMPORT- und LOAD-Sitzungen

Das folgende Beispielprogramm zeigt, wie folgende Aufgaben ausgeführt werden:

- Daten in eine Datei exportieren
- Daten in eine Tabelle importieren
- Daten in eine Tabelle laden
- Status einer Ladeoperation überprüfen

Die Quellendatei für dieses Beispielprogramm (tbmove.sqc) befindet sich im Verzeichnis \sql1ib\samples\c. Sie enthält sowohl DB2-API-Aufrufe als auch Aufrufe mit eingebettetem SQL. Die Prozedurdatei b1dapp.cmd, die sich im gleichen Verzeichnis befindet, enthält die Befehle zur Erstellung dieses und anderer Beispielprogramme.

Geben Sie tbmove ein, um das Beispielprogramm (ausführbare Datei) zu starten. Es kann unter Umständen nützlich sein, einige der generierten Dateien näher zu betrachten, etwa die Nachrichtendatei und die ASCII-Datendatei mit begrenzter Satzlänge.

```
/******  
** Licensed Materials - Property of IBM  
**  
** Governed under the terms of the International  
** License Agreement for Non-Warranted Sample Code.  
**  
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2002  
** All Rights Reserved.  
**  
** US Government Users Restricted Rights - Use, duplication or  
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
*****  
**  
** SOURCE FILE NAME: tbmove.sqc  
**  
** SAMPLE: How to move table data  
**  
** DB2 APIs USED:  
**     db2Export -- Export  
**     db2Import -- Import  
**     sqluvqdp -- Quiesce Table Spaces for Table  
**     db2Load -- Load  
**     db2LoadQuery -- Load Query  
**  
** SQL STATEMENTS USED:  
**     PREPARE  
**     DECLARE CURSOR  
**     OPEN  
**     FETCH  
**     CLOSE  
**     CREATE TABLE  
**     DROP  
**  
** OUTPUT FILE: tbmove.out (available in the online documentation)  
*****  
**  
** For more information on the sample programs, see the README file.  
**
```

```

** For information on developing C applications, see the Application
** Development Guide.
**
** For information on using SQL statements, see the SQL Reference.
**
** For information on DB2 APIs, see the Administrative API Reference.
**
** For the latest information on programming, building, and running DB2
** applications, visit the DB2 application development website:
**   http://www.software.ibm.com/data/db2/udb/ad
**
*****/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sqlenv.h>
#include <sqlutil.h>
#include <db2ApiDf.h>
#include "utilemb.h"

int DataExport(char *);
int TbImport(char *);
int TbLoad(char *);
int TbLoadQuery(void);

/* support function */
int ExportedDataDisplay(char *);
int NewTableDisplay(void);

EXEC SQL BEGIN DECLARE SECTION;
char strStmt[256];
short deptnumb;
char deptname[15];
EXEC SQL END DECLARE SECTION;

int main(int argc, char *argv[])
{
    int rc = 0;
    char dbAlias[SQL_ALIAS_SZ + 1];
    char user[USERID_SZ + 1];
    char pswd[PSWD_SZ + 1];
    char dataFileName[256];

    /* check the command line arguments */
    rc = CmdLineArgsCheck1(argc, argv, dbAlias, user, pswd);
    if (rc != 0)
    {
        return rc;
    }

    printf("\nTHIS SAMPLE SHOWS HOW TO MOVE TABLE DATA.\n");

    /* connect to database */
    rc = DbConn(dbAlias, user, pswd);
    if (rc != 0)
    {
        return rc;
    }

    #if defined(DB2NT)
        sprintf(dataFileName, "%s%stbmove.DEL", getenv("DB2PATH"), PATH_SEP);
    #else /* UNIX */
        sprintf(dataFileName, "%s%stbmove.DEL", getenv("HOME"), PATH_SEP);
    #endif

    rc = DataExport(dataFileName);
    rc = TbImport(dataFileName);

```

```

rc = TblLoad(dataFileName);
rc = TblLoadQuery();

/* disconnect from the database */
rc = DbDisconn(dbAlias);
if (rc != 0)
{
    return rc;
}

return 0;
} /* main */

int ExportedDataDisplay(char *dataFileName)
{
    struct sqlca sqlca;
    FILE *fp;
    char buffer[100];
    int maxChars = 100;
    int numChars;
    int charNb;

    fp = fopen(dataFileName, "r");
    if (fp == NULL)
    {
        return 1;
    }

    printf("\n The content of the file '%s' is:\n", dataFileName);
    printf(" ");
    numChars = fread(buffer, 1, maxChars, fp);
    while (numChars > 0)
    {
        for (charNb = 0; charNb < numChars; charNb++)
        {
            if (buffer[charNb] == '\n')
            {
                printf("\n");
                if (charNb < numChars - 1)
                {
                    printf(" ");
                }
            }
            else
            {
                printf("%c", buffer[charNb]);
            }
        }
        numChars = fread(buffer, 1, maxChars, fp);
    }

    if (ferror(fp))
    {
        fclose(fp);
        return 1;
    }
    else
    {
        fclose(fp);
    }

    return 0;
} /* ExportedDataDisplay */

int NewTableDisplay(void)
{
    struct sqlca sqlca;

```

```

printf("\n SELECT * FROM newtable\n");
printf("  DEPTNUMB DEPTNAME      \n");
printf("  ----- \n");

strcpy(strStmt, "SELECT * FROM newtable");

EXEC SQL PREPARE stmt FROM :strStmt;
EMB_SQL_CHECK("statement -- prepare");

EXEC SQL DECLARE c0 CURSOR FOR stmt;

EXEC SQL OPEN c0;
EMB_SQL_CHECK("cursor -- open");

EXEC SQL FETCH c0 INTO :deptnumb, :deptname;
EMB_SQL_CHECK("cursor -- fetch");

while (sqlca.sqlcode != 100)
{
printf("   %8d %-s\n", deptnumb, deptname);

EXEC SQL FETCH c0 INTO :deptnumb, :deptname;
EMB_SQL_CHECK("cursor -- fetch");
}

EXEC SQL CLOSE c0;

return 0;
} /* NewTableDisplay */

int DataExport(char *dataFileName)
{
int rc = 0;
struct sqlca sqlca;
struct sqldcol dataDescriptor;
char actionString[256];
struct sqllob *pAction;
char msgFileName[128];
struct db2ExportOut outputInfo;
struct db2ExportStruct exportParmStruct;

printf("\n-----");
printf("\nUSE THE DB2 API:\n");
printf(" db2Export -- Export\n");
printf("TO EXPORT DATA TO A FILE.\n");

printf("\n Be sure to complete all table operations and release\n");
printf(" all locks before starting an export operation. This\n");
printf(" can be done by issuing a COMMIT after closing all\n");
printf(" cursors opened WITH HOLD, or by issuing a ROLLBACK.\n");
printf(" Please refer to the 'Administrative API Reference'\n");
printf(" for the details.\n");

/* export data */
dataDescriptor.dcolmeth = SQL_METH_D;
strcpy(actionString, "SELECT deptnumb, deptname FROM org");
pAction = (struct sqllob *)malloc(sizeof(sqluint32) +
                                sizeof(actionString) + 1);
pAction->length = strlen(actionString);
strcpy(pAction->data, actionString);
strcpy(msgFileName, "tbexport.MSG");

exportParmStruct.piDataFileName = dataFileName;
exportParmStruct.piLobPathList = NULL;
exportParmStruct.piLobFileList = NULL;
exportParmStruct.piDataDescriptor = &dataDescriptor;

```

```

exportParmStruct.piActionString    = pAction;
exportParmStruct.piFileType        = SQL_DEL;
exportParmStruct.piFileTypeMod     = NULL;
exportParmStruct.piMsgFileName     = msgFileName;
exportParmStruct.iCallerAction     = SQLU_INITIAL;
exportParmStruct.poExportInfoOut   = &outputInfo;

printf("\n Export data.\n");
printf("  client destination file name: %s\n", dataFileName);
printf("  action                       : %s\n", actionString);
printf("  client message file name      : %s\n", msgFileName);

/* export data */
db2Export(db2Version820,
          &exportParmStruct,
          &sqlca);

DB2_API_CHECK("data -- export");

/* free memory allocated */
free(pAction);

/* display exported data */
rc = ExportedDataDisplay(dataFileName);

return 0;
} /* DataExport */

int TbImport(char *dataFileName)
{
    int rc = 0;
    struct sqlca sqlca;
    struct sqldcol dataDescriptor;
    char actionString[256];
    struct sqlchar *pAction;
    char msgFileName[128];
    struct db2ImportIn inputInfo;
    struct db2ImportOut outputInfo;
    struct db2ImportStruct importParmStruct;
    long commitcount = 10;

    printf("\n-----");
    printf("\nUSE THE DB2 API:\n");
    printf(" db2Import -- Import\n");
    printf("TO IMPORT DATA TO A TABLE.\n");

    /* create new table */
    printf("\n CREATE TABLE newtable(deptnumb SMALLINT NOT NULL,");
    printf("\n                               deptname VARCHAR(14))\n");

    EXEC SQL CREATE TABLE newtable(deptnumb SMALLINT NOT NULL,
                                     deptname VARCHAR(14));
    EMB_SQL_CHECK("new table -- create");

    /* import table */
    dataDescriptor.dcolmeth = SQL_METH_D;
    strcpy(actionString, "INSERT INTO newtable");
    pAction = (struct sqlchar *)malloc(sizeof(short) +
                                       sizeof(actionString) + 1);
    pAction->length = strlen(actionString);
    strcpy(pAction->data, actionString);
    strcpy(msgFileName, "tbimport.MSG");

    /* Setup db2ImportIn structure */
    inputInfo.iRowcount = inputInfo.iRestartcount = 0;
    inputInfo.iSkipcount = inputInfo.iWarningcount = 0;
    inputInfo.iNoTimeout = 0;

```

```

inputInfo.iAccessLevel = SQLU_ALLOW_NO_ACCESS;
inputInfo.piCommitcount = &commitcount;

printf("\n Import table.\n");
printf(" client source file name : %s\n", dataFileName);
printf(" action : %s\n", actionString);
printf(" client message file name: %s\n", msgFileName);

ImportParmStruct.piDataFileName = dataFileName;
importParmStruct.piLobPathList = NULL;
importParmStruct.piDataDescriptor = &dataDescriptor;
importParmStruct.piActionString = pAction;
importParmStruct.piFileType = SQL_DEL;
importParmStruct.piFileTypeMod = NULL;
importParmStruct.piMsgFileName = msgFileName;
importParmStruct.piImportInfoIn = &inputInfo;
importParmStruct.piImportInfoOut = &outputInfo;
importParmStruct.piNullIndicators = NULL;
importParmStruct.iCallerAction = SQLU_INITIAL;

/* import table */
db2Import(db2Version820,
          &importParmStruct,
          &sqlca);

DB2_API_CHECK("table -- import");

/* free memory allocated */
free(pAction);

/* display import info */
printf("\n Import info.\n");
printf(" rows read : %ld\n", (int)outputInfo.oRowsRead);
printf(" rows skipped : %ld\n", (int)outputInfo.oRowsSkipped);
printf(" rows inserted : %ld\n", (int)outputInfo.oRowsInserted);
printf(" rows updated : %ld\n", (int)outputInfo.oRowsUpdated);
printf(" rows rejected : %ld\n", (int)outputInfo.oRowsRejected);
printf(" rows committed: %ld\n", (int)outputInfo.oRowsCommitted);

/* display content of the new table */
rc = NewTableDisplay();

/* drop new table */
printf("\n DROP TABLE newtable\n");

EXEC SQL DROP TABLE newtable;
EMB_SQL_CHECK("new table -- drop");

return 0;
} /* TbImport */

int TbLoad(char *dataFileName)
{
    int rc = 0;
    struct sqlca sqlca;

    struct db2LoadStruct paramStruct;
    struct db2LoadIn inputInfoStruct;
    struct db2LoadOut outputInfoStruct;

    struct sqlu_media_list mediaList;
    struct sqldcol dataDescriptor;
    char actionString[256];
    struct sqlchar *pAction;
    char localMsgFileName[128];

    printf("\n-----");

```

```

printf("\nUSE THE DB2 API:\n");
printf(" sqluvqdp -- Quiesce Table Spaces for Table\n");
printf(" db2Load -- Load\n");
printf("TO LOAD DATA INTO A TABLE.\n");

/* create new table */
printf("\n CREATE TABLE newtable(deptnumb SMALLINT NOT NULL,");
printf("\n                                deptname VARCHAR(14))\n");

EXEC SQL CREATE TABLE newtable(deptnumb SMALLINT NOT NULL,
                                deptname VARCHAR(14));
EMB_SQL_CHECK("new table -- create");

/* quiesce table spaces for table */
printf("\n Quiesce the table spaces for 'newtable'.\n");

EXEC SQL COMMIT;
EMB_SQL_CHECK("transaction -- commit");

/* quiesce table spaces for table */
sqluvqdp("newtable", SQLU QUIESCEMODE_RESET_OWNED, NULL, &sqlca);
DB2_API_CHECK("tablespaces for table -- quiesce");

/* load table */
mediaList.media_type = SQLU_CLIENT_LOCATION;
mediaList.sessions = 1;
mediaList.target.location =
    (struct sqlu_location_entry *)malloc(sizeof(struct sqlu_location_entry) *
                                        mediaList.sessions);
strcpy(mediaList.target.location->location_entry, dataFileName);

dataDescriptor.dcolmeth = SQL_METH_D;

strcpy(actionString, "INSERT INTO newtable");
pAction = (struct sqlchar *)malloc(sizeof(short) +
                                   sizeof(actionString) + 1);
pAction->length = strlen(actionString);
strcpy(pAction->data, actionString);

strcpy(localMsgFileName, "tblload.MSG");

/* Setup the input information structure */
inputInfoStruct.piUseTablespace = NULL;
inputInfoStruct.iSavecount = 0; /* consistency points */
/* as infrequently as possible */
inputInfoStruct.iRestartcount = 0; /* start at row 1 */
inputInfoStruct.iRowcount = 0; /* load all rows */
inputInfoStruct.iWarningcount = 0; /* don't stop for warnings */
inputInfoStruct.iDataBufferSize = 0; /* default data buffer size */
inputInfoStruct.iSortBufferSize = 0; /* def. warning buffer size */
inputInfoStruct.iHoldQuiesce = 0; /* don't hold the quiesce */
inputInfoStruct.iRestartphase = ' '; /* ignored anyway */
inputInfoStruct.iStatsOpt = SQLU_STATS_NONE; /* don't bother with them */
inputInfoStruct.iIndexingMode = SQLU_INX_AUTOSELECT; /* let load choose */
/* indexing mode */

inputInfoStruct.iCpuParallelism = 0;
inputInfoStruct.iNonrecoverable = SQLU_NON_RECOVERABLE_LOAD;
inputInfoStruct.iAccessLevel = SQLU_ALLOW_NO_ACCESS;
inputInfoStruct.iLockWithForce = SQLU_NO_FORCE;
inputInfoStruct.iCheckPending = SQLU_CHECK_PENDING_CASCADE_DEFERRED;

/* Setup the parameter structure */
paramStruct.piSourceList = &mediaList;
paramStruct.piLobPathList = NULL;
paramStruct.piDataDescriptor = &dataDescriptor;
paramStruct.piActionString = pAction;
paramStruct.piFileType = SQL_DEL;

```

```

paramStruct.piFileTypeMod = NULL;
paramStruct.piLocalMsgFileName = localMsgFileName;
paramStruct.piTempFilesPath = NULL;
paramStruct.piVendorSortWorkPaths = NULL;
paramStruct.piCopyTargetList = NULL;
paramStruct.piNullIndicators = NULL;
paramStruct.piLoadInfoIn = &inputInfoStruct;
paramStruct.poLoadInfoOut = &outputInfoStruct;
paramStruct.piPartLoadInfoIn = NULL;
paramStruct.poPartLoadInfoOut = NULL;
paramStruct.iCallerAction = SQLU_INITIAL;

printf("\n Load table.\n");
printf(" client source file name : %s\n", dataFileName);
printf(" action : %s\n", actionString);
printf(" client message file name: %s\n", localMsgFileName);

/* load table */
db2Load (db2Version810, /* Database version number */
        &paramStruct, /* In/out parameters */
        &sqlca); /* SQLCA */

DB2_API_CHECK("table -- load");

/* free memory allocated */
free(pAction);

/* display load info */
printf("\n Load info.\n");
printf(" rows read : %d\n", (int)outputInfoStruct.oRowsRead);
printf(" rows skipped : %d\n", (int)outputInfoStruct.oRowsSkipped);
printf(" rows loaded : %d\n", (int)outputInfoStruct.oRowsLoaded);
printf(" rows deleted : %d\n", (int)outputInfoStruct.oRowsDeleted);
printf(" rows rejected : %d\n", (int)outputInfoStruct.oRowsRejected);
printf(" rows committed: %d\n", (int)outputInfoStruct.oRowsCommitted);

/* display content of the new table */
rc = NewTableDisplay();

/* drop new table */
printf("\n DROP TABLE newtable\n");

EXEC SQL DROP TABLE newtable;
EMB_SQL_CHECK("new table -- drop");

return 0;
} /* TbLoad */

int TbLoadQuery(void)
{
int rc = 0;
struct sqlca sqlca;
char tableName[128];
char loadMsgFileName[128];
db2LoadQueryStruct loadQueryParameters;
db2LoadQueryOutputStruct loadQueryOutputStructure;

printf("\n-----");
printf("\nUSE THE DB2 API:\n");
printf(" db2LoadQuery -- Load Query\n");
printf("TO CHECK THE STATUS OF A LOAD OPERATION.\n");

/* Initialize structures */
memset(&loadQueryParameters, 0, sizeof(db2LoadQueryStruct));
memset(&loadQueryOutputStructure, 0, sizeof(db2LoadQueryOutputStruct));

/* Set up the tablename to query. */

```

```

loadQueryParameters.iStringType = DB2LOADQUERY_TABLENAME;
loadQueryParameters.piString = tableName;

/* Specify that we want all LOAD messages to be reported. */
loadQueryParameters.iShowLoadMessages = DB2LOADQUERY_SHOW_ALL_MSGS;

/* LOAD summary information goes here. */
loadQueryParameters.poOutputStruct = &loadQueryOutputStructure;

/* Set up the local message file. */
loadQueryParameters.piLocalMessageFile = loadMsgFileName;

/* call the DB2 API */
strcpy(tableName, "ORG");
strcpy(loadMsgFileName, "tbldqry.MSG");

/* load query */
db2LoadQuery(db2Version810, &loadQueryParameters, &sqlca);
printf("\n Note: the table load for '%s' is NOT in progress.\n", tableName);
printf(" So an empty message file '%s' will be created,\n", loadMsgFileName);
printf(" and the following values will be zero.\n");
DB2_API_CHECK("status of load operation -- check");

printf("\n Load status has been written to local file %s.\n",
        loadMsgFileName);

printf("    Number of rows read          = %d\n",
        loadQueryOutputStructure.oRowsRead);

printf("    Number of rows skipped        = %d\n",
        loadQueryOutputStructure.oRowsSkipped);

printf("    Number of rows loaded         = %d\n",
        loadQueryOutputStructure.oRowsLoaded);

printf("    Number of rows rejected       = %d\n",
        loadQueryOutputStructure.oRowsRejected);

printf("    Number of rows deleted        = %d\n",
        loadQueryOutputStructure.oRowsDeleted);

printf("    Number of rows committed     = %d\n",
        loadQueryOutputStructure.oRowsCommitted);

printf("    Number of warnings            = %d\n",
        loadQueryOutputStructure.oWarningCount);

return 0;
} /* TbLoadQuery */

```



---

## Anhang D. Dateiformate

---

### Dateiformate der Dienstprogramme EXPORT, IMPORT und LOAD

Es werden fünf Betriebssystemdateiformate beschrieben, die von den DB2<sup>®</sup>-Dienstprogrammen EXPORT, IMPORT und LOAD unterstützt werden:

**DEL** ASCII-Format ohne universelle Zeilenbegrenzer. Dieses Format ermöglicht den Datenaustausch zwischen einer Vielzahl von Datenbankmanagern und Dateimanagern. Bei diesem allgemeinen Ansatz zum Speichern von Daten werden spezielle Zeichenbegrenzer verwendet, um Spaltenwerte zu trennen.

**ASC** ASCII-Format mit universellen Zeilenbegrenzern. Dieses Format dient zum Importieren oder Laden von Daten anderer Anwendungen, die Flachdateien mit ausgerichteten Spaltendaten erstellen.

**PC/IXF**

PC-Version des IXF-Formats (Integrated Exchange Format). Dies ist das bevorzugte Format für den Austausch von Daten innerhalb des Datenbankmanagers. PC/IXF ist eine strukturierte Beschreibung einer Datenbanktabelle, die eine externe Darstellung der internen Tabelle enthält.

**WSF** Arbeitsblattformat. Dieses Format ermöglicht den Datenaustausch mit Produkten wie Lotus<sup>®</sup> 1-2-3<sup>®</sup> and Symphony. Dieses Dateiformat wird vom Dienstprogramm LOAD nicht unterstützt.

**CURSOR**

Ein Cursor, der für eine SQL-Abfrage deklariert wird. Dieser Dateityp wird nur durch das Dienstprogramm LOAD unterstützt.

Bei Verwendung des DEL-, WSF- oder ASC-Datendateiformats müssen Sie die Tabelle, einschließlich der Spaltennamen und Datentypen, definieren, bevor Sie die Datei importieren. Die Datentypen in den Feldern der Betriebssystemdatei werden in den entsprechenden Datentyp in der Datenbanktabelle umgesetzt. Das Dienstprogramm IMPORT akzeptiert Daten mit geringfügigen Inkompatibilitätsproblemen, einschließlich Zeichendaten, die beim Import möglicherweise aufgefüllt bzw. abgeschnitten werden, und numerischer Daten, die in unterschiedliche Typen numerischer Felder importiert werden.

Bei Verwendung des Datendateiformats PC/IXF muss die Tabelle beim Starten der Importoperation noch nicht vorhanden sein. Benutzerdefinierte einzigartige Typen (UDTs - User-defined Types) werden nicht in die Spaltentypen der neuen Tabelle aufgenommen. Stattdessen wird der Basistyp verwendet. Auf ähnliche Weise werden UDTs beim Exportieren in das PC/IXF-Datendateiformat in der PC/IXF-Datei als Basisdatentypen gespeichert.

Bei Verwendung des Dateityps CURSOR muss die Tabelle inklusive Spaltennamen und Datentypen definiert werden, bevor die Ladeoperation gestartet wird. Die Spaltentypen der SQL-Abfrage müssen mit den entsprechenden Spaltentypen in der Zieltabelle kompatibel sein. Es ist nicht erforderlich, dass der angegebene Cursor geöffnet ist, bevor die Ladeoperation gestartet wird. Das Dienstprogramm LOAD verarbeitet das gesamte Ergebnis der Abfrage, die dem angegebenen Cursor zugeordnet ist, und zwar unabhängig davon, ob der Cursor zum Abrufen von Zeilen verwendet wurde oder nicht.

### Zugehörige Konzepte:

- „Queries and table expressions“ in *SQL Reference, Volume 1*

### Zugehörige Referenzen:

- „DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer)“ auf Seite 276
- „ASC-Dateiformat (ASCII-Format mit universellen Zeilenbegrenzern)“ auf Seite 281
- „PC-Version des IXF-Dateiformats“ auf Seite 285
- „Casting between data types“ in *SQL Reference, Volume 1*
- „Assignments and comparisons“ in *SQL Reference, Volume 1*

### Zugehörige Beispiele:

- „dtformat.out -- HOW TO LOAD AND IMPORT DATA FORMAT EXTENSIONS (C)“
- „dtformat.sqc -- Load and import data format extensions (C)“

---

## DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer)

Eine DEL-Datei (Delimited ASCII - ASCII-Format ohne universelle Zeilenbegrenzer) ist eine sequenzielle ASCII-Datei mit Zeilen- und Spaltenbegrenzungszeichen. Jede DEL-Datei ist ein Datenstrom aus ASCII-Zeichen, der sich aus Zellenwerten zusammensetzt, die zuerst nach Zeilen und dann nach Spalten geordnet sind. Die Zeilen im Datenstrom werden durch Zeilenbegrenzungszeichen getrennt. Die einzelnen Zellenwerte innerhalb einer jeden Zeile werden durch Spaltenbegrenzungszeichen getrennt.

Die folgende Tabelle beschreibt das Format von DEL-Dateien, die importiert oder als Ergebnis einer Exportaktion generiert werden können.

```
DEL-Datei ::= Zeile 1 Daten || Zeilenbegrenzer ||  
            Zeile 2 Daten || Zeilenbegrenzer ||  
            .  
            .  
            .  
            Zeile n Daten || Wahlfreier Zeilenbegrenzer
```

```
Zeile i Daten ::= Zellenwert(i,1) || Spaltenbegrenzer ||  
                Zellenwert(i,2) || Spaltenbegrenzer ||  
                .  
                .  
                .  
                Zellenwert(i,m)
```

Zeilenbegrenzer ::= ASCII-Zeilenvorschubfolge<sup>a</sup>

Spaltenbegrenzer ::= Standardwert: ASCII-Komma (,)<sup>b</sup>

```
Zellenwert(i,j) ::= Führende Leerzeichen  
                  || ASCII-Darstellung eines numerischen Werts  
                  (Integer, Dezimal oder Gleitkomma)  
                  || Begrenzte Zeichenfolge  
                  || Nicht begrenzte Zeichenfolge  
                  || Nachgestellte Leerzeichen
```

Nicht begrenzte Zeichenfolge ::= Ein Satz beliebiger Zeichen außer eines Zeilen- oder Spaltenbegrenzers

Begrenzte Zeichenfolge ::= Ein Zeichenfolgebegrenzer ||

Eine erweiterte Zeichenfolge ||  
Ein Zeichenfolgebegrenzer ||  
Nicht verwertbarer Rest

Nicht verwertbarer Rest ::= Ein Satz beliebiger Zeichen außer eines Zeilen- oder Spaltenbegrenzers

Zeichenfolgebegrenzer ::= Standardwert: Doppelte ASCII-Anführungszeichen (")<sup>c</sup>

Erweiterte Zeichenfolge ::= || Ein Satz beliebiger Zeichen außer eines Zeilen- oder Zeichenfolgebegrenzers, wenn der Änderungswert NODOUBLEDEL angegeben ist  
|| Ein Satz beliebiger Zeichen außer eines Zeilen- oder Zeichenfolgebegrenzers, wenn die Zeichenfolge nicht Teil zweier aufeinander folgender Zeichenfolgebegrenzer ist  
|| Ein Satz beliebiger Zeichen außer eines Zeichenfolgebegrenzers, wenn der Zeichenfolgebegrenzer nicht Teil zweier aufeinander folgender Zeichenfolgebegrenzer ist und der Änderungswert DELPRIORITYCHAR angegeben ist

Dateieindezeichen ::= Hex '1A' (nur Windows-Betriebssysteme)

ASCII-Darstellung eines numerischen

Werts<sup>d</sup> ::= Wahlfreies

Vorzeichen "+" oder "-"

|| 1 bis 31 Dezimalziffern mit wahlfreiem Dezimalzeichen vor,  
nach oder zwischen zwei Ziffern  
|| Wahlfreier Exponent

Exponent ::= Zeichen "E" oder "e"

|| Wahlfreies Zeichen "+" oder "-"  
|| 1 bis 3 Dezimalziffern ohne Dezimalzeichen

Dezimalziffer ::= Beliebiges Zeichen "0", "1", ... "9"

Dezimalzeichen ::= Standardwert: ASCII-Punkt (.)<sup>e</sup>

- <sup>a</sup> Als Satzbegrenzer wird das Zeilenvorschubzeichen, ASCII-Wert x0A, angenommen. Unter den Windows-Betriebssystemen generierte Daten können den 2-Byte-Standardwert für Rücklauf/Zeilenvorschub, 0x0D0A, verwenden. Daten in EBCDIC-Codepages müssen das EBCDIC-Zeichen für Zeilenvorschub (0x25) als Satzbegrenzer verwenden (EBCDIC-Daten können mit der Option CODEPAGE des Befehls LOAD geladen werden).
- <sup>b</sup> Das Spaltenbegrenzungszeichen kann mit der Option COLDEL angegeben werden.
- <sup>c</sup> Das Zeichenfolgebegrenzungszeichen kann mit der Option CHARDEL angegeben werden.

**Anmerkung:** Die Standardpriorität der Begrenzer ist:

1. Satzbegrenzer
2. Zeichenbegrenzer
3. Spaltenbegrenzer

- <sup>d</sup> Wenn die ASCII-Darstellung eines numerischen Wertes einen Exponenten aufweist, handelt es sich um eine FLOAT-Konstante. Wenn sie ein Dezimalzeichen,

jedoch keinen Exponenten aufweist, handelt es sich um eine DECIMAL-Konstante. Wenn sie kein Dezimalzeichen und keinen Exponenten aufweist, handelt es sich um eine INTEGER-Konstante.

- <sup>e</sup> Das Dezimalzeichen kann mit der Option DECPT angegeben werden.

#### Zugehörige Referenzen:

- „Beschreibungen von DEL-Datentypen“ auf Seite 279

---

## Beispiel und Datentypbeschreibungen

### Beispiel für DEL-Datei

Es folgt ein Beispiel für eine DEL-Datei. Jede Zeile endet mit einer Zeilenvorschubfolge (unter den Windows-Betriebssystemen endet jede Zeile mit einer Folge aus einem Rücklauf- und einem Zeilenvorschubzeichen).

```
"Smith, Bob",4973,15.46  
"Jones, Bill",12345,16.34  
"Williams, Sam",452,193.78
```

Das folgende Beispiel zeigt die Verwendung von nicht begrenzten Zeichenfolgen. Das Spaltenbegrenzungszeichen wurde durch ein Semikolon ersetzt, weil die Zeichendaten ein Komma enthalten.

```
Smith, Bob;4973;15.46  
Jones, Bill;12345;16.34  
Williams, Sam;452;193.78
```

#### Anmerkungen:

1. Ein Leerzeichen (X'20') ist als Begrenzer nie zulässig.
2. Leerzeichen vor dem ersten oder nach dem letzten Zeichen eines Zellenwertes werden beim Import gelöscht. Leerzeichen, die in einen Zellenwert eingebettet sind, werden nicht gelöscht.
3. Ein Punkt (.) ist kein gültiges Zeichenfolgebegrenzungszeichen, da dies zu Konflikten mit den Punkten in den Werten für Zeitmarken führen würde.
4. Bei reinem DBCS (Grafik), gemischtem DBCS und EUC sind Begrenzer auf den Bereich von x00 bis x3F (jeweils einschließlich) beschränkt.
5. Bei DEL-Daten, die in einer EBCDIC-Codepage angegeben werden, dürfen die Begrenzer nicht mit den DBCS-Startzeichen und -Endezeichen identisch sein.
6. Unter den Windows-Betriebssystemen gibt das erste Dateiendezeichen (X'1A'), das sich nicht innerhalb von Zeichenbegrenzern befindet, das Dateiende an. Darauf folgende Daten werden nicht importiert.
7. Ein Nullwert wird durch das Fehlen eines Zellenwerts an einer Stelle, an der sich normalerweise ein Zellenwert befände, oder durch eine Folge aus Leerzeichen gekennzeichnet.
8. Da einige andere Produkte die Länge von Zeichenfeldern auf 254 oder 255 Byte begrenzen, gibt das Dienstprogramm EXPORT eine Warnung aus, wenn eine Zeichenspalte mit einer Länge von mehr als 254 Zeichen für den Export ausgewählt wird. Das Dienstprogramm IMPORT verarbeitet Felder, die so lang wie die längste LONG VARCHAR- und LONG VARGRAPHIC-Spalte sind.

## Beschreibungen von DEL-Datentypen

Tabelle 16. Gültige Datentypformate für das DEL-Dateiformat

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
BIGINT	Eine INTEGER-Konstante im Bereich -9 223 372 036 854 775 808 bis 9 223 372 036 854 775 807.	ASCII-Darstellung eines numerischen Wertes im Bereich -9 223 372 036 854 775 808 bis 9 223 372 036 854 775 807. Dezimal- und Gleitkommazahlen werden ganzzahlig abgeschnitten.
BLOB, CLOB	Zeichendaten, die in Zeichenbegrenzer (z. B. doppelte Anführungszeichen) eingeschlossen sind.	Eine begrenzte oder nicht begrenzte Zeichenfolge. Die Zeichenfolge wird als Wert der Datenbankspalte verwendet.
BLOB_FILE, CLOB_FILE	Die Zeichendaten für jede BLOB-/CLOB-Spalte werden in einzelnen Dateien gespeichert, und der Dateiname ist in Zeichenbegrenzer eingeschlossen.	Der begrenzte oder nicht begrenzte Name der Datei, die die Daten enthält.
CHAR	Zeichendaten, die in Zeichenbegrenzer (z. B. doppelte Anführungszeichen) eingeschlossen sind.	Eine begrenzte oder nicht begrenzte Zeichenfolge. Die Zeichenfolge wird nach Bedarf abgeschnitten oder mit Leerzeichen (X'20') aufgefüllt und so an die Breite der Datenbankspalte angeglichen.
DATE	<i>jjjmmmtt</i> (Jahr Monat Tag) ohne Spaltenbegrenzer. Beispiel: 19931029  Alternativ kann die Option DATESISO verwendet werden, um festzulegen, dass alle Datumswerte im ISO-Format exportiert werden sollen.	Eine begrenzte oder nicht begrenzte Zeichenfolge, die einen Datumswert in einem ISO-Format enthält, das mit dem Gebietscode der Zieldatenbank konsistent ist, oder eine nicht begrenzte Zeichenfolge im Format <i>jjjmmmtt</i> .
DBCLOB (nur DBCS)	Grafikdaten werden als begrenzte Zeichenfolge exportiert.	Eine begrenzte oder nicht begrenzte Zeichenfolge (Länge ist eine gerade Anzahl Byte). Die Zeichenfolge wird als Wert der Datenbankspalte verwendet.
DBCLOB_FILE (nur DBCS)	Die Zeichendaten für jede DBCLOB-Spalte werden in einzelnen Dateien gespeichert, und der Dateiname ist in Zeichenbegrenzer eingeschlossen.	Der begrenzte oder nicht begrenzte Name der Datei, die die Daten enthält.

Tabelle 16. Gültige Datentypformate für das DEL-Dateiformat (Forts.)

<b>Datentyp</b>	<b>Format in vom Dienstprogramm EXPORT erstellten Dateien</b>	<b>Für das Dienstprogramm IMPORT zulässiges Format</b>
DECIMAL	Eine DECIMAL-Konstante mit der Genauigkeit und der Anzahl der Kommastellen des Feldes, das exportiert wird. Die Option DECPLUSBLANK kann verwendet werden, um anzugeben, dass positiven Dezimalwerten ein Leerzeichen statt eines Pluszeichens (+) vorangestellt werden soll.	ASCII-Darstellung eines numerischen Werts, der nicht außerhalb des Bereichs der Datenbankspalte liegt, in die das Feld importiert wird. Wenn der Eingabewert mehr Ziffern nach dem Dezimalzeichen enthält, als in der Datenbankspalte Platz finden, werden die überzähligen Ziffern abgeschnitten.
FLOAT(long)	Eine FLOAT-Konstante im Bereich -10E307 bis 10E307.	ASCII-Darstellung eines numerischen Wertes im Bereich -10E307 bis 10E307.
GRAPHIC (nur DBCS)	Grafikdaten werden als begrenzte Zeichenfolge exportiert.	Eine begrenzte oder nicht begrenzte Zeichenfolge (Länge ist eine gerade Anzahl Byte). Die Zeichenfolge wird nach Bedarf abgeschnitten oder mit Doppelbyteleerzeichen (X'8140') aufgefüllt und so an die Breite der Datenbankspalte angeglichen.
INTEGER	Eine INTEGER-Konstante im Bereich -2 147 483 648 bis 2 147 483 647.	ASCII-Darstellung eines numerischen Werts im Bereich -2 147 483 648 bis 2 147 483 647. Dezimal- und Gleitkommazahlen werden ganzzahlig abgeschnitten.
LONG VARCHAR	Zeichendaten, die in Zeichenbegrenzer (z. B. doppelte Anführungszeichen) eingeschlossen sind.	Eine begrenzte oder nicht begrenzte Zeichenfolge. Die Zeichenfolge wird als Wert der Datenbankspalte verwendet.
LONG VARGRAPHIC (nur DBCS)	Grafikdaten werden als begrenzte Zeichenfolge exportiert.	Eine begrenzte oder nicht begrenzte Zeichenfolge (Länge ist eine gerade Anzahl Byte). Die Zeichenfolge wird als Wert der Datenbankspalte verwendet.
SMALLINT	Eine INTEGER-Konstante im Bereich -32 768 bis 32 767.	ASCII-Darstellung eines numerischen Werts im Bereich -32 768 bis 32 767. Dezimal- und Gleitkommazahlen werden ganzzahlig abgeschnitten.

Tabelle 16. Gültige Datentypformate für das DEL-Dateiformat (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
TIME	<i>ss.mm.ss</i> (Stunde Minuten Sekunden). Ein Zeitwert im ISO-Format, in Zeichenbegrenzer eingeschlossen. Beispiel: "09.39.43"	Eine begrenzte oder nicht begrenzte Zeichenfolge, die einen Zeitwert in einem Format enthält, das mit dem Gebietscode der Zieldatenbank konsistent ist.
TIMESTAMP	<i>jjjj-mm-tt-ss.mm.ss.nnnnnn</i> (Jahr Monat Tag Stunde Minuten Sekunden Mikrosekunden). Eine Zeichenfolge, die ein Datum und eine Uhrzeit darstellt, in Zeichenbegrenzer eingeschlossen.	Eine begrenzte oder nicht begrenzte Zeichenfolge, die einen Zeitmarkenwert enthält, dessen Speicherung in einer Datenbank zulässig ist.
VARCHAR	Zeichendaten, die in Zeichenbegrenzer (z. B. doppelte Anführungszeichen) eingeschlossen sind.	Eine begrenzte oder nicht begrenzte Zeichenfolge. Die Zeichenfolge wird nach Bedarf abgeschnitten und so an die maximal zulässige Breite der Datenbankspalte angeglichen.
VARGRAPHIC (nur DBCS)	Grafikdaten werden als begrenzte Zeichenfolge exportiert.	Eine begrenzte oder nicht begrenzte Zeichenfolge (Länge ist eine gerade Anzahl Byte). Die Zeichenfolge wird nach Bedarf abgeschnitten und so an die maximal zulässige Breite der Datenbankspalte angeglichen.

## ASC-Dateiformat (ASCII-Format mit universellen Zeilenbegrenzern)

Eine ASC-Datei ist eine sequenzielle ASCII-Datei mit Zeilenbegrenzungszeichen. Sie kann für den Austausch von Daten mit beliebigen ASCII-Produkten verwendet werden, die Daten im Spaltenformat verarbeiten können (einschließlich Textverarbeitungsprogrammen). Jede ASC-Datei ist ein Datenstrom aus ASCII-Zeichen, der sich aus Datenwerten zusammensetzt, die in Zeilen und Spalten angeordnet sind. Die Zeilen innerhalb des Datenstroms werden durch Zeilenbegrenzungszeichen voneinander getrennt. Jede Spalte innerhalb einer Zeile wird durch ein Wertepaar bestehend aus Anfangsposition-Endposition (mit IMPORT-Parametern angegeben) definiert. Jedes Paar stellt Speicherpositionen innerhalb einer Zeile dar, die als Bytepositionen angegeben werden. Die erste Position innerhalb einer Zeile ist Byteposition 1. Das erste Element jedes Positionspaars ist das Byte in der Zeile, bei dem die Spalte beginnt, und das zweite Element ist das Byte, bei dem die Spalte endet. Die Spalten können sich möglicherweise überlappen. Jede Zeile in einer ASC-Datei hat die gleiche Spaltendefinition.

Eine ASC-Datei ist durch Folgendes definiert:

```
ASC-Datei ::= Zeile 1 Daten || Zeilenbegrenzer ||  
           Zeile 2 Daten || Zeilenbegrenzer ||  
           .  
           .  
           .  
           Zeile n Daten
```

Zeile i Daten ::= ASCII-Zeichen || Zeilenbegrenzer

Zeilenbegrenzer ::= ASCII-Zeilenvorschubfolge<sup>a</sup>

- <sup>a</sup> Als Satzbegrenzer wird das Zeilenvorschubzeichen, ASCII-Wert x0A, angenommen. Unter den Windows-Betriebssystemen generierte Daten können den 2-Byte-Standardwert für Rücklauf/Zeilenvorschub, 0x0D0A, verwenden. Daten in EBCDIC-Codepages müssen das EBCDIC-Zeichen für Zeilenvorschub (0x25) als Satzbegrenzer verwenden (EBCDIC-Daten können mit der Option CODEPAGE des Befehls LOAD geladen werden). Der Satzbegrenzer wird nie als Teil eines Feldes mit Daten gewertet.

#### Zugehörige Referenzen:

- „Beschreibungen von ASC-Datentypen“ auf Seite 283

---

## Beispiel und Datentypbeschreibungen

### Beispiel für ASC-Datei

Es folgt ein Beispiel für eine ASC-Datei. Jede Zeile endet mit einer Zeilenvorschubfolge (unter den Windows-Betriebssystemen endet jede Zeile mit einer Folge aus einem Rücklauf- und einem Zeilenvorschubzeichen).

Smith, Bob	4973	15.46
Jones, Suzanne	12345	16.34
Williams, Sam	452123	193.78

#### Anmerkungen:

- Bei ASC-Dateien wird davon ausgegangen, dass sie keine Spaltennamen enthalten.
- Zeichenfolgen sind *nicht* in Begrenzer eingeschlossen. Der Datentyp einer Spalte in der ASC-Datei wird durch den Datentyp der Zielspalte in der Datenbanktabelle festgelegt.
- NULL wird unter folgenden Bedingungen in einer Datenbankspalte, die Nullwerte enthalten darf, importiert:
  - Ein Feld mit Leerzeichen ist für eine numerische Datenbankspalte oder eine Datenbankspalte des Typs DATE, TIME oder TIMESTAMP bestimmt.
  - Es wird ein Feld ohne Anfangs- und Endpositionsangabe angegeben.
  - Es wird ein Positionspaar mit Anfangs- und Endpositionen gleich Null angegeben.
  - Eine Datenzeile ist zu kurz, um einen gültigen Wert für die Zielspalte zu erhalten.
  - Die LOAD-Option NULL INDICATORS wird verwendet und die Nullanzeigerspalte enthält N (oder einen anderen vom Benutzer angegebenen Wert).

4. Wenn die Zielspalte keinen Nullwert enthalten darf, führt der Versuch, ein Feld mit Leerzeichen in eine numerische Spalte oder eine Spalte des Typs DATE, TIME oder TIMESTAMP zu importieren, zum Zurückweisen der Zeile.
5. Wenn die Eingabedaten nicht mit der Zielspalte kompatibel sind und diese Spalte Nullwerte enthalten darf, wird eine Null importiert oder die Zeile zurückgewiesen, je nachdem, wo der Fehler erkannt wird. Wenn die Spalte keine Nullwerte enthalten darf, wird die Zeile zurückgewiesen. Nachrichten mit Angaben zu den festgestellten Inkompatibilitäten werden in die Nachrichtendatei geschrieben.

## Beschreibungen von ASC-Datentypen

Tabelle 17. Gültige Datentypformate für das ASC-Dateiformat

Datentyp	Für das Dienstprogramm IMPORT zulässiges Format
BIGINT	<p>Eine Konstante eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich -9 223 372 036 854 775 808 bis 9 223 372 036 854 775 807 liegen. Dezimalzahlen werden ganzzahlig abgeschnitten. Ein Komma, Punkt oder Doppelpunkt gilt als Dezimalzeichen. Tausendertrennzeichen sind nicht zulässig.</p> <p>Die Anfangs- und Endposition müssen ein Feld festlegen, dessen Breite 50 Byte nicht überschreitet. Ganze Zahlen, Dezimalzahlen und die Mantissen von Gleitkommazahlen dürfen nicht mehr als 31 Stellen haben. Die Exponenten von Gleitkommazahlen dürfen nicht mehr als 3 Stellen haben.</p>
BLOB/CLOB	Eine Zeichenfolge. Die Zeichenfolge wird nach Bedarf rechts abgeschnitten und so an die maximal zulässige Länge der Zielspalte angeglichen. Wenn die ASC-Option zum <i>Abschneiden von Leerzeichen</i> wirksam ist, werden abschließende Leerzeichen von der ursprünglichen oder der abgeschnittenen Zeichenfolge entfernt.
BLOB_FILE, CLOB_FILE, DBCLOB_FILE (nur DBCS)	Der begrenzte oder nicht begrenzte Name der Datei, die die Daten enthält.
CHAR	Eine Zeichenfolge. Die Zeichenfolge wird nach Bedarf rechts abgeschnitten oder mit Leerzeichen aufgefüllt und so an die Breite der Zielspalte angeglichen.
DATE	<p>Eine Zeichenfolge, die einen Datumswert in einem Format enthält, das mit dem Gebietscode der Zieldatenbank konsistent ist.</p> <p>Die Anfangs- und Endposition müssen eine Feldbreite festlegen, die im Bereich der externen Darstellung eines Datums liegt.</p>
DBCLOB (nur DBCS)	Eine Zeichenfolge mit einer geraden Anzahl Byte. Eine Zeichenfolge mit einer ungeraden Anzahl Byte ist ungültig und wird abgelehnt. Eine gültige Zeichenfolge wird nach Bedarf rechts abgeschnitten und so an die maximal zulässige Länge der Zielspalte angeglichen.

Tabelle 17. Gültige Datentypformate für das ASC-Dateiformat (Forts.)

Datentyp	Für das Dienstprogramm IMPORT zulässiges Format
DECIMAL	<p>Eine Konstante eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich der Datenbankspalte liegen, in die sie importiert werden. Wenn der Eingabewert mehr Ziffern nach dem Dezimalzeichen enthält, als in der Datenbankspalte Platz finden, werden die überzähligen Ziffern abgeschnitten. Ein Komma, Punkt oder Doppelpunkt gilt als Dezimalzeichen. Tausendertrennzeichen sind nicht zulässig.</p> <p>Die Anfangs- und Endposition müssen ein Feld festlegen, dessen Breite 50 Byte nicht überschreitet. Ganze Zahlen, Dezimalzahlen und die Mantissen von Gleitkommazahlen dürfen nicht mehr als 31 Stellen haben. Die Exponenten von Gleitkommazahlen dürfen nicht mehr als 3 Stellen haben.</p>
FLOAT(long)	<p>Eine Konstante eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Alle Werte sind gültig. Ein Komma, Punkt oder Doppelpunkt gilt als Dezimalzeichen. Ein E wird unabhängig von der Groß-/Kleinschreibung als Anfang des Exponenten einer FLOAT-Konstante akzeptiert.</p> <p>Die Anfangs- und Endposition müssen ein Feld festlegen, dessen Breite 50 Byte nicht überschreitet. Ganze Zahlen, Dezimalzahlen und die Mantissen von Gleitkommazahlen dürfen nicht mehr als 31 Stellen haben. Die Exponenten von Gleitkommazahlen dürfen nicht mehr als 3 Stellen haben.</p>
GRAPHIC (nur DBCS)	<p>Eine Zeichenfolge mit einer geraden Anzahl Byte. Eine Zeichenfolge mit einer ungeraden Anzahl Byte ist ungültig und wird abgelehnt. Eine gültige Zeichenfolge wird nach Bedarf rechts abgeschnitten oder mit Doppelbyteleerzeichen (0x8140) aufgefüllt und so an die maximal zulässige Länge der Zielspalte angeglichen.</p>
INTEGER	<p>Eine Konstante eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich -2 147 483 648 bis 2 147 483 647 liegen. Dezimalzahlen werden ganzzahlig abgeschnitten. Ein Komma, Punkt oder Doppelpunkt gilt als Dezimalzeichen. Tausendertrennzeichen sind nicht zulässig.</p> <p>Die Anfangs- und Endposition müssen ein Feld festlegen, dessen Breite 50 Byte nicht überschreitet. Ganze Zahlen, Dezimalzahlen und die Mantissen von Gleitkommazahlen dürfen nicht mehr als 31 Stellen haben. Die Exponenten von Gleitkommazahlen dürfen nicht mehr als 3 Stellen haben.</p>
LONG VARCHAR	<p>Eine Zeichenfolge. Die Zeichenfolge wird nach Bedarf rechts abgeschnitten und so an die maximal zulässige Länge der Zielspalte angeglichen. Wenn die ASC-Option zum <i>Abschneiden von Leerzeichen</i> wirksam ist, werden abschließende Leerzeichen von der ursprünglichen oder der abgeschnittenen Zeichenfolge entfernt.</p>

Tabelle 17. Gültige Datentypformate für das ASC-Dateiformat (Forts.)

Datentyp	Für das Dienstprogramm IMPORT zulässiges Format
LONG VARGRAPHIC (nur DBCS)	Eine Zeichenfolge mit einer geraden Anzahl Byte. Eine Zeichenfolge mit einer ungeraden Anzahl Byte ist ungültig und wird abgelehnt. Eine gültige Zeichenfolge wird nach Bedarf rechts abgeschnitten und so an die maximal zulässige Länge der Zielspalte angeglichen.
SMALLINT	Eine Konstante eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich -32 768 bis 32 767 liegen. Dezimalzahlen werden ganzzahlig abgeschnitten. Ein Komma, Punkt oder Doppelpunkt gilt als Dezimalzeichen. Tausendertrennzeichen sind nicht zulässig.  Die Anfangs- und Endposition müssen ein Feld festlegen, dessen Breite 50 Byte nicht überschreitet. Ganze Zahlen, Dezimalzahlen und die Mantissen von Gleitkommazahlen dürfen nicht mehr als 31 Stellen haben. Die Exponenten von Gleitkommazahlen dürfen nicht mehr als 3 Stellen haben.
TIME	Eine Zeichenfolge, die einen Zeitwert in einem Format enthält, das mit dem Gebietscode der Zieldatenbank konsistent ist.  Die Anfangs- und Endposition müssen eine Feldbreite festlegen, das im Bereich der externen Darstellung eines Zeitwerts liegt.
TIMESTAMP	Eine Zeichenfolge, die einen Zeitmarkenwert enthält, dessen Speicherung in einer Datenbank zulässig ist.  Die Anfangs- und Endposition müssen eine Feldbreite festlegen, die im Bereich der externen Darstellung einer Zeitmarke liegt.
VARCHAR	Eine Zeichenfolge. Die Zeichenfolge wird nach Bedarf rechts abgeschnitten und so an die maximal zulässige Länge der Zielspalte angeglichen. Wenn die ASC-Option zum <i>Abschneiden von Leerzeichen</i> wirksam ist, werden abschließende Leerzeichen von der ursprünglichen oder der abgeschnittenen Zeichenfolge entfernt.
VARGRAPHIC (nur DBCS)	Eine Zeichenfolge mit einer geraden Anzahl Byte. Eine Zeichenfolge mit einer ungeraden Anzahl Byte ist ungültig und wird abgelehnt. Eine gültige Zeichenfolge wird nach Bedarf rechts abgeschnitten und so an die maximal zulässige Länge der Zielspalte angeglichen.

## PC-Version des IXF-Dateiformats

Die PC-Version des IXF-Dateiformats (PC/IXF) ist eine Datenbankmanageranpassung der IXF-Datenaustauscharchitektur (Integration Exchange Format). Die IXF-Architektur ist speziell auf den Austausch von Strukturen und Daten relationaler Datenbanken ausgelegt. Die PC/IXF-Architektur ermöglicht es dem Datenbankmanager, eine Datenbank zu exportieren, ohne dass die Anforderungen und Eigenarten eines Zielprodukts bekannt sein müssen. Auch ein Produkt, in das eine PC/IXF-Datei importiert wird, muss nur die PC/IXF-Architektur beherrschen. Die Merkmale des Produkts, von dem die Datei exportiert wurde, sind nicht von Bedeutung. Die PC/IXF-Dateiarchitektur wahrt die Unabhängigkeit sowohl der exportierenden als auch der importierenden Datenbanksysteme.

Die IXF-Architektur stellt ein generisches Austauschformat für relationale Datenbanken dar, das eine Vielzahl von Typen relationaler Daten unterstützt, zu denen auch einige Typen zählen, die von bestimmten Produkten für relationale Datenbanken möglicherweise nicht unterstützt werden. Das PC/IXF-Dateiformat wahrt diese Flexibilität. Beispielsweise unterstützt die PC/IXF-Architektur sowohl SBCS-Datentypen (Einzelbytezeichenfolge) als auch DBCS-Datentypen (Doppelbytezeichenfolge). Nicht alle Implementierungen unterstützen alle PC/IXF-Datentypen. Selbst eingeschränkte Implementierungen erlauben jedoch die Erkennung und Disposition von nicht unterstützten Datentypen beim Import.

Im allgemeinen besteht eine PC/IXF-Datei aus einer ununterbrochenen Folge von Datensätzen mit variabler Länge. Die Datei enthält die folgenden Datensatztypen in der gezeigten Reihenfolge:

- Einen Kopfdatensatz der Satzart H
- Einen Tabellensatz der Satzart T
- Mehrere Spaltendeskriptorsätze der Satzart C (ein Datensatz für jede Spalte in der Tabelle)
- Mehrere Datensätze der Satzart D (jede Zeile in der Tabelle wird durch einen oder mehrere D-Datensätze dargestellt)

Eine PC/IXF-Datei kann an einer beliebigen Stelle hinter dem H-Satz auch Anwendungsdatensätze vom Satztyp A enthalten. Diese Datensätze sind in PC/IXF-Dateien zulässig, so dass eine Anwendung zusätzliche Daten, die im PC/IXF-Format nicht definiert sind, in eine PC/IXF-Datei einfügen kann. A-Datensätze werden von jedem Programm ignoriert, das eine PC/IXF-Datei liest und keine Informationen zum Datenformat und Inhalt besitzt, das bzw. der durch die Anwendungskennung im A-Datensatz impliziert wird.

Jeder Datensatz in einer PC/IXF-Datei beginnt mit einem Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große rechtsbündige Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Programme, die PC/IXF-Dateien lesen, müssen diese Satzlängen verwenden, um das Ende des aktuellen Datensatzes und den Anfang des nächsten Datensatzes zu bestimmen. H-, T- und C-Datensätze müssen groß genug für alle definierten Felder sein, und die Satzlängenfelder müssen selbstverständlich mit den Ist-Längen übereinstimmen. Wenn jedoch zusätzliche Daten (z. B. ein *neues* Feld) am Ende eines dieser Datensätze angefügt wird, müssen vorhandene Programme, die PC/IXF-Dateien lesen, die zusätzlichen Daten ignorieren und lediglich eine Warnung generieren. Programme hingegen, die PC/IXF-Dateien schreiben, müssen H-, T- und C-Datensätze mit genau der für alle definierten Felder erforderlichen Länge schreiben.

Wenn eine PC/IXF-Datei Spalten mit LOB-Positionskennungen (so genannte LLS-Spalten) enthält, muss jede LLS-Spalte einen eigenen D-Datensatz haben. D-Datensätze werden automatisch durch das Dienstprogramm EXPORT erstellt. Werden die PC/IXF-Dateien jedoch mit Hilfe eines Fremdherstellertools erstellt, müssen Sie diese Datensätze manuell erstellen. Des Weiteren wird für jede LOB-Spalte in einer Tabelle (auch für LOB-Spalten mit einem Nullwert) eine LOB-Positionskennung benötigt. Für eine LOB-Spalte mit einem Nullwert müssen Sie eine LOB-Positionskennung mit einem LOB-Nullwert erstellen.

PC/IXF-Dateisätze bestehen aus Feldern, die Zeichendaten enthalten. Die Dienstprogramme IMPORT und EXPORT interpretieren diese Zeichendaten mit Hilfe der CPGID der Zieldatenbank, wobei es zwei Ausnahmen gibt:

- Das IXFADATA-Feld von A-Datensätzen.  
Die Codepage-Umgebung von Zeichendaten in einem IXFADATA-Feld wird von der Anwendung festgelegt, die einen bestimmten A-Datensatz erstellt und verarbeitet. Das bedeutet, dass die Umgebung je nach Implementierung variiert.
- Das IXFDCOLS-Feld von D-Datensätzen.  
Die Codepage-Umgebung von Zeichendaten in einem IXFDCOLS-Feld ist eine Funktion der im C-Datensatz enthaltenen Informationen. Dieser Datensatz definiert eine bestimmte Spalte und ihre Daten.

Numerische Felder in H-, T- und C-Datensätzen sowie im Präfixteil von D- und A-Datensätzen müssen rechtsbündige Einzelbytezeichendarstellungen von ganzzahligen Werten sein, die mit führenden Nullen oder Leerzeichen aufgefüllt sind. Nullwerte müssen mit mindestens einem (rechtsbündigen) Nullzeichen, nicht mit Leerzeichen, angegeben werden. Wenn eines dieser numerischen Felder, z. B. IXFCLENG, dessen Länge vom Datentyp impliziert wird, nicht verwendet wird, muss es mit Leerzeichen aufgefüllt werden. Diese numerischen Felder sind:

```
IXFHRECL, IXFTRECL, IXFCRECL, IXFDRECL, IXFARECL,
IXFHHCNT, IXFHSBCP, IXFHDBCP, IXFTCCNT, IXFTNAML,
IXFCLENG, IXFCDRID, IXFCPOSN, IXFCNAML, IXFCTYPE,
IXFCSBCP, IXFCDBCP, IXFCNDIM, IXFCDSIZ, IXFDRID
```

**Anmerkung:** Das PC/IXF-Dateiformat des Datenbankmanagers ist nicht identisch mit dem Format von IBM System/370.

#### Zugehörige Referenzen:

- „PC/IXF-Satztypen“ auf Seite 287
- „PC/IXF-Datentypen“ auf Seite 304
- „Allgemeine Regeln für den Import von PC/IXF-Dateien in Datenbanken“ auf Seite 315
- „Datentypspezifische Regeln für den Import von PC/IXF-Dateien in Datenbanken“ auf Seite 317
- „Option FORCEIN“ auf Seite 320
- „Unterschiede zwischen PC/IXF und Version 0 System/370 IXF“ auf Seite 328
- „Beschreibungen von PC/IXF-Datentypen“ auf Seite 311

---

## PC-Version des IXF-Dateiformats - Details

### PC/IXF-Satztypen

Es gibt fünf PC/IXF-Basissatztypen:

- Kopfdatensatz
- Tabellensatz
- Spaltendeskriptorsatz
- Datensatz
- Anwendungssatz

Außerdem werden von DB2 UDB sechs Untertypen von Anwendungssätzen verwendet:

- Indexsatz
- Hierarchiesatz

- Satz für untergeordnete Tabelle
- Fortsetzungssatz
- Beendigungssatz
- Identitätssatz

Jeder PC/IXF-Satztyp ist als eine Folge von Feldern definiert. Diese Felder sind erforderlich und müssen in der gezeigten Reihenfolge vorkommen.

#### KOPFDATEN

FELDDNAME	LÄNGE	TYP	KOMMENTARE
IXFHRECL	06 BYTE	CHARACTER	Satzlänge
IXFHRECT	01 BYTE	CHARACTER	Satztyp = "H"
IXFHID	03 BYTE	CHARACTER	IXF-Kennung
IXFHVERS	04 BYTE	CHARACTER	IXF-Version
IXFHPROD	12 BYTE	CHARACTER	Produkt
IXFHDATE	08 BYTE	CHARACTER	Erstellungsdatum
IXFHTIME	06 BYTE	CHARACTER	Erstellungszeit
IXFHHCNT	05 BYTE	CHARACTER	Anzahl vorhergehender Datensätze
IXFHSBCP	05 BYTE	CHARACTER	Einzelbyte-Codepage
IXFHDBC	05 BYTE	CHARACTER	Doppelbyte-Codepage
IXFHFIL1	02 BYTE	CHARACTER	reserviert

Die folgenden Felder sind in den Kopfdaten enthalten:

#### IXFHRECL

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der H-Datensatz muss lang genug sein für alle darin definierten Felder.

#### IXFHRECT

Der IXF-Satztyp, der für diesen Datensatz auf H gesetzt ist.

#### IXFHID

Die Dateiformatkennung, die für diese Datei auf IXF gesetzt ist.

#### IXFHVERS

Die Version des PC/IXF-Formats, die bei der Erstellung der Datei verwendet wurde, ist auf "0002" gesetzt.

#### IXFHPROD

Ein Feld, das von dem Programm, das die Datei erstellt, zu Identifikationszwecken verwendet werden kann. Wenn dieses Feld ausgefüllt wird, dienen die ersten sechs Byte dazu, das Produkt anzugeben, von dem die Datei erstellt wurde. Die letzten sechs Byte dienen dazu, die Version bzw. das Release des Produkts anzugeben. Der Datenbankmanager verwendet dieses Feld, um das Vorhandensein von datenbankmanagerspezifischen Daten anzuzeigen.

#### IXFHDATE

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjjmmtt*.

#### IXFHTIME

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld ist wahlfrei und kann leergelassen werden.

### IXFHHCNT

Die Anzahl der H-, T- und C-Datensätze in dieser Datei, die vor dem ersten Datensatz kommen. A-Datensätze sind bei diesem Wert nicht berücksichtigt.

### IXFHSBCP

Einzelbyte-Codepage-Feld, das eine Einzelbytezeichendarstellung einer SBCS-CPGID oder "00000" enthält.

Das Dienstprogramm EXPORT setzt dieses Feld mit der SBCS-CPGID der exportierten Datenbanktabelle gleich. Beispiel: Wenn die SBCS-CPGID den Wert 850 hat, enthält dieses Feld "00850".

### IXFHDBCP

Doppelbyte-Codepage-Feld, das eine Einzelbytezeichendarstellung einer DBCS-CPGID oder "00000" enthält.

Das Dienstprogramm EXPORT setzt dieses Feld mit der DBCS-CPGID der exportierten Datenbanktabelle gleich. Beispiel: Wenn die DBCS-CPGID den Wert 301 hat, enthält dieses Feld "00301".

### IXFHFIL1

Zusatzfeld, das zwei Leerzeichen enthält. Dient als Entsprechung zu einem reservierten Feld in Host-IXF-Dateien.

### TABELLENSATZ

FELDDNAME	LÄNGE	TYP	KOMMENTARE
-----	-----	-----	-----
IXFTRECL	006 BYTE	CHARACTER	Satzlänge
IXFTRECT	001 BYTE	CHARACTER	Satztyp = "T"
IXFTNAML	003 BYTE	CHARACTER	Namenslänge
IXFTNAME	256 BYTE	CHARACTER	Name der Daten
IXFTQUALL	003 BYTE	CHARACTER	Länge Qualifikationsmerkmal
IXFTQUAL	256 BYTE	CHARACTER	Qualifikationsmerkmal
IXFTSRC	012 BYTE	CHARACTER	Datenquelle
IXFTDATA	001 BYTE	CHARACTER	Datenkonvention = "C"
IXFTFORM	001 BYTE	CHARACTER	Datenformat = "M"
IXFTMFRM	005 BYTE	CHARACTER	Maschinenformat = "PC"
IXFTLOC	001 BYTE	CHARACTER	Datenspeicherposition = "I"
IXFTCCNT	005 BYTE	CHARACTER	"C"-Satzzählung
IXFTFIL1	002 BYTE	CHARACTER	reserviert
IXFTDESC	030 BYTE	CHARACTER	Datenbeschreibung
IXFTPKNM	257 BYTE	CHARACTER	Primärschlüsselname
IXFTDSPC	257 BYTE	CHARACTER	reserviert
IXFTISPC	257 BYTE	CHARACTER	reserviert
IXFTLSPC	257 BYTE	CHARACTER	reserviert

Die folgenden Felder sind im Tabellensatz enthalten:

### IXFTRECL

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der T-Datensatz muss lang genug sein für alle darin definierten Felder.

### IXFTRECT

Der IXF-Satztyp, der für diesen Datensatz auf T gesetzt ist.

### IXFTNAML

Die Länge des Tabellennamens im IXFTNAME-Feld in Byte.

**IXFTNAME**

Der Name der Tabelle. Wenn jede Datei nur eine Tabelle enthält, dient dieses Feld nur zur Information. Dieses Feld wird vom Datenbankmanager beim Importieren von Daten nicht verwendet. Beim Schreiben einer PC/IXF-Datei schreibt der Datenbankmanager den DOS-Dateinamen (und möglicherweise Pfadinformationen) in dieses Feld.

**IXFTQULL**

Die Länge des Qualifikationsmerkmals für den Tabellennamen im IXFTQUAL-Feld in Byte.

**IXFTQUAL**

Qualifikationsmerkmal für den Tabellennamen, das den Ersteller einer Tabelle in einem relationalen System angibt. Dieses Feld dient nur zur Information. Wenn ein Programm, das eine Datei schreibt, keine Daten in dieses Feld schreiben muss, wird es vorzugsweise mit Leerzeichen gefüllt. Programme, die eine Datei lesen, können dieses Feld möglicherweise drucken oder anzeigen oder es in einem Informationsfeld speichern. Vom Inhalt dieses Feldes dürfen jedoch keine Berechnungen abhängen.

**IXFTSRC**

Wird verwendet, um die ursprüngliche Quelle der Daten anzugeben. Dieses Feld dient nur zur Information. Wenn ein Programm, das eine Datei schreibt, keine Daten in dieses Feld schreiben muss, wird es vorzugsweise mit Leerzeichen gefüllt. Programme, die eine Datei lesen, können dieses Feld möglicherweise drucken oder anzeigen oder es in einem Informationsfeld speichern. Vom Inhalt dieses Feldes dürfen jedoch keine Berechnungen abhängen.

**IXFTDATA**

Konvention, die zum Beschreiben der Daten verwendet wird. Dieses Feld muss für den Import und Export auf C gesetzt sein, um anzugeben, dass einzelne Spaltenattribute in den folgenden Spaltendeskriptorsätzen (C) beschrieben werden und dass Daten den PC/IXF-Konventionen entsprechen.

**IXFTFORM**

Konvention, die zum Speichern von numerischen Daten verwendet wird. Dieses Feld muss auf M gesetzt sein, um anzugeben, dass numerische Daten in den Datensätzen (D) in dem (internen) Maschinenformat gespeichert sind, das im IXFTMFRM-Feld angegeben ist.

**IXFTMFRM**

Das Format beliebiger Maschinendaten in der PC/IXF-Datei. Der Datenbankmanager liest oder schreibt Dateien nur, wenn dieses Feld auf PCbbb gesetzt ist, wobei *b* ein Leerzeichen ist und PC angibt, dass Daten in der PC/IXF-Datei in einem IBM PC-Maschinenformat vorliegen.

**IXFTLOC**

Die Speicherposition der Daten. Der Datenbankmanager unterstützt nur den Wert I, was bedeutet, dass sich die Daten in dieser Datei befinden.

**IXFTCNT**

Die Anzahl von C-Datensätzen in dieser Tabelle. Hierbei handelt es sich um eine rechtsbündige Zeichendarstellung eines ganzzahligen Wertes.

**IXFTFIL1**

Zusatzfeld, das zwei Leerzeichen enthält. Dient als Entsprechung zu einem reservierten Feld in Host-IXF-Dateien.

## IXFTDESC

Beschreibende Daten zur Tabelle. Dieses Feld dient nur zur Information. Wenn ein Programm, das eine Datei schreibt, keine Daten in dieses Feld schreiben muss, wird es vorzugsweise mit Leerzeichen gefüllt. Programme, die eine Datei lesen, können dieses Feld möglicherweise drucken oder anzeigen oder es in einem Informationsfeld speichern. Vom Inhalt dieses Feldes dürfen jedoch keine Berechnungen abhängen. Dieses Feld enthält NOT NULL WITH DEFAULT, wenn für die Spalte NOT NULL WITH DEFAULT angegeben war und der Tabellename nicht aus einer Workstation-Datenbank stammt.

## IXFTPKNM

Der Name des für die Tabelle definierten Primärschlüssels (sofern vorhanden). Der Name wird in Form einer auf auf Null endenden Zeichenfolge gespeichert.

## IXFTDSPC

Dieses Feld ist zur zukünftigen Verwendung reserviert.

## IXFTISPC

Dieses Feld ist zur zukünftigen Verwendung reserviert.

## IXFTLSPC

Dieses Feld ist zur zukünftigen Verwendung reserviert.

## SPALTENDESKRIPTORSATZ

FELDDNAME	LÄNGE	TYP	KOMMENTARE
-----	-----	-----	-----
IXFCRECL	006 BYTE	CHARACTER	Satzlänge
IXFTRECT	001 BYTE	CHARACTER	Satztyp = "C"
IXFCNAML	003 BYTE	CHARACTER	Länge des Spaltennamens
IXFCNAME	256 BYTE	CHARACTER	Spaltenname
IXFCNULL	001 BYTE	CHARACTER	Spalte lässt Nullen zu
IXFCDEF	001 BYTE	CHARACTER	Spalte hat Standardwerte
IXFCSLCT	001 BYTE	CHARACTER	Spaltenauswahlmarkierung
IXFCCKPOS	002 BYTE	CHARACTER	Position im Primärschlüssel
IXFCCLAS	001 BYTE	CHARACTER	Datenklasse
IXFCTYPE	003 BYTE	CHARACTER	Datentyp
IXFCSBCP	005 BYTE	CHARACTER	Einzelbyte-Codepage
IXFCDBC	005 BYTE	CHARACTER	Doppelbyte-Codepage
IXFCLENG	005 BYTE	CHARACTER	Spaltendatenlänge
IXFCDRID	003 BYTE	CHARACTER	Satzkennung "D"
IXFCPOSN	006 BYTE	CHARACTER	Spaltenposition
IXFCDESC	030 BYTE	CHARACTER	Spaltenbeschreibung
IXFCLOBL	020 BYTE	CHARACTER	Länge der LOB-Spalte
IXFCUDTL	003 BYTE	CHARACTER	Länge des UDT-Namens
IXFCUDTN	256 BYTE	CHARACTER	UDT-Name
IXFCDEFL	003 BYTE	CHARACTER	Länge des Standardwerts
IXFCDEFV	254 BYTE	CHARACTER	Standardwert
IXFCDLPR	010 BYTE	CHARACTER	DATALINK-Merkmale
IXFCREF	001 BYTE	CHARACTER	Verweistyp
IXFCNDIM	002 BYTE	CHARACTER	Anzahl Dimensionen
IXFCDSIZ	variabel	CHARACTER	Größe jeder Dimension

Die folgenden Felder sind in Spaltendeskriptorsätzen enthalten:

## IXFCRECL

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der C-Datensatz muss lang genug sein für alle darin definierten Felder.

**IXFCRECT**

Der IXF-Satztyp, der für diesen Datensatz auf C gesetzt ist.

**IXFCNAML**

Die Länge des Spaltennamens im IXFTNAME-Feld in Byte.

**IXFCNAME**

Der Name der Spalte.

**IXFCNULL**

Gibt an, ob Nullen in dieser Spalte zulässig sind. Gültige Werte sind Y oder N.

**IXFCDEF**

Gibt an, ob für dieses Feld ein Standardwert definiert ist. Gültige Werte sind Y oder N.

**IXFCSLCT**

Ein veraltetes Feld, das die Auswahl einer Untermenge von Spalten in der Tabelle ermöglichen sollte. Programme, die PC/IXF-Dateien schreiben, müssen in diesem Feld stets Y speichern. Programme, die PC/IXF-Dateien lesen, müssen dieses Feld ignorieren.

**IXFCKPOS**

Die Position der Spalte als Bestandteil des Primärschlüssels. Gültige Werte sind 01 bis 16 bzw. N, falls die Spalte nicht Bestandteil des Primärschlüssels ist.

**IXFCCLAS**

Die Klasse der Datentypen, die im IXFCTYPE-Feld verwendet werden sollen. Der Datenbankmanager unterstützt nur relationale Typen (R).

**IXFCTYPE**

Der Datentyp der Spalte.

**IXFCSBCP**

Enthält eine Einzelbytezeichendarstellung einer SBCS-CPGID. Dieses Feld legt die CPGID für Einzelbytezeichendaten fest, die im IXFDCOLS-Feld der D-Datensätze für diese Spalte vorkommen.

Die Semantik dieses Feldes variiert mit dem Datentyp für die Spalte (im IXFCTYPE-Feld angegeben).

- Für eine Zeichenfolgespalte sollte dieses Feld normalerweise einen Wert ungleich Null enthalten, der mit dem Wert des IXFHSBCP-Feldes im H-Datensatz übereinstimmt. Andere Werte sind jedoch erlaubt. Wenn dieser Wert Null ist, wird angenommen, dass die Spalte Bitfolgenddaten enthält.
- Für eine numerische Spalte hat dieses Feld keine Bedeutung. Es wird vom Dienstprogramm EXPORT auf Null gesetzt und vom Dienstprogramm IMPORT ignoriert.
- Für eine Datums- oder Uhrzeitspalte hat dieses Feld keine Bedeutung. Es wird vom Dienstprogramm EXPORT auf den Wert des Feldes IXFHSBCP gesetzt und vom Dienstprogramm IMPORT ignoriert.
- Für eine Grafikspalte muss dieses Feld Null sein.

**IXFCDBCP**

Enthält eine Einzelbytezeichendarstellung einer DBCS-CPGID. Dieses Feld legt die CPGID für Doppelbytezeichendaten fest, die im IXFDCOLS-Feld der D-Datensätze für diese Spalte vorkommen.

Die Semantik dieses Feldes variiert mit dem Datentyp für die Spalte (im IXFCTYPE-Feld angegeben).

- Für eine Zeichenfolgespalte muss dieses Feld Null sein oder einen Wert enthalten, der mit dem Wert des IXFHDBCP-Feldes im H-Datensatz übereinstimmt. Andere Werte sind jedoch erlaubt. Wenn der Wert im IXFCSBCP-Feld Null ist, muss der Wert in diesem Feld Null sein.
- Für eine numerische Spalte hat dieses Feld keine Bedeutung. Es wird vom Dienstprogramm EXPORT auf Null gesetzt und vom Dienstprogramm IMPORT ignoriert.
- Für eine Datums- oder Uhrzeitspalte hat dieses Feld keine Bedeutung. Es wird vom Dienstprogramm EXPORT auf Null gesetzt und vom Dienstprogramm IMPORT ignoriert.
- Für eine Grafikspalte muss dieses Feld einen Wert haben, der mit dem Wert des IXFHDBCP-Feldes identisch ist.

### **IXFCLENG**

Bietet Informationen zur Größe der beschriebenen Spalte. Für einige Datentypen wird dieses Feld nicht verwendet und muss Leerzeichen enthalten. Für andere Datentypen enthält dieses Feld die rechtsbündige Zeichendarstellung einer ganzen Zahl, die die Spaltenlänge festlegt. Für andere Datentypen ist dieses Feld in zwei Teilfenster unterteilt: 3 Byte für die Genauigkeit und 2 Byte für die Anzahl der Kommastellen. Beide Teilfelder sind rechtsbündige Zeichendarstellungen von ganzen Zahlen.

### **IXFCDRID**

Die D-Satzkennung. Dieses Feld enthält die rechtsbündige Zeichendarstellung eines ganzzahligen Wertes. Es ist möglich, mehrere D-Datensätze für die einzelnen Datenzeilen in der PC/IXF-Datei zu verwenden. Dieses Feld gibt an, welcher D-Datensatz (einer von mehreren D-Datensätzen, die eine Datenzeile bilden) die Daten für die Spalte enthält. Der Wert Eins (z. B. 001) gibt an, dass sich die Daten für eine Spalte im ersten D-Datensatz in einer Datenzeile befinden. Der erste C-Datensatz muss den IXFCDRID-Wert Eins haben. Alle nachfolgenden C-Datensätze müssen einen IXFCDRID-Wert haben, der mit dem Wert des vorherigen C-Datensatzes übereinstimmt oder um Eins größer ist.

### **IXFCPOSN**

Der Wert dieses Feldes wird verwendet, um die Daten für die Spalte innerhalb eines der D-Datensätze zu lokalisieren, die eine Zeile der Tabellendaten darstellen. Es handelt sich um den Anfangspunkt der Daten für diese Spalte innerhalb des IXFDCOLS-Feldes des D-Datensatzes. Wenn die Spalte Nullwerte enthalten darf, zeigt IXFCPOSN auf den Nullanzeiger, andernfalls auf die Daten selbst. Wenn eine Spalte Daten mit variabler Länge enthält, beginnen die Daten selbst mit dem aktuellen Längenzeiger. Der IXFCPOSN-Wert für das erste Byte im IXFDCOLS-Feld des D-Datensatzes ist Eins (nicht Null). Wenn sich eine Spalte in einem neuen D-Datensatz befindet, muss der Wert von IXFCPOSN Eins sein. Andernfalls müssen die IXFCPOSN-Werte von Spalte zu Spalte so zunehmen, dass sich die Datenwerte nicht überschneiden.

### **IXFCDESC**

Beschreibende Informationen zu der Spalte. Dieses Feld dient nur zur Information. Wenn ein Programm, das in eine Datei schreibt, keine Daten in dieses Feld schreiben muss, wird es vorzugsweise mit Leerzeichen gefüllt. Programme, die eine Datei lesen, können dieses Feld möglicherweise drucken oder anzeigen oder es in einem Informationsfeld speichern. Vom Inhalt dieses Feldes dürfen jedoch keine Berechnungen abhängen.

**IXFCLOBL**

Die Länge des in dieser Spalte definierten großen Objekts (LOB) in Byte. Handelt es sich nicht um eine LOB-Spalte, lautet der Wert in diesem Feld 000.

**IXFCUDTL**

Die Länge des Namens des benutzerdefinierten Typs (UDT) im IXFCUDTN-Feld in Byte. Handelt es sich nicht um eine Spalte des Typs UDT, hat dieses Feld den Wert 000.

**IXFCUDTN**

Der Name des benutzerdefinierten Typs, der als Datentyp für diese Spalte verwendet wird.

**IXFCDEFL**

Die Länge des Standardwerts im IXFCDEFV-Feld in Byte. Wenn diese Spalte keinen Standardwert hat, lautet der Wert in diesem Feld 000.

**IXFCDEFV**

Gibt den Standardwert für diese Spalte an, sofern ein solcher definiert wurde.

**IXFCDLPR**

Handelt es sich bei der Spalte um eine DATALINK-Spalte, beschreibt dieses Feld die folgenden Merkmale:

- Das erste Zeichen steht für den Verbindungstyp. Es hat den Wert U.
- Das zweite Zeichen stellt den Verbindungssteuerungstyp dar. Gültige Werte sind N (= keine Steuerung) und F (= Dateisteuerung).
- Das dritte Zeichen stellt die Integritätsstufe dar und hat den Wert A. Hiermit wird angegeben, dass der Datenbankmanager alle DATALINK-Werte steuert.
- Das vierte Zeichen steht für die Leseberechtigung. Gültige Werte sind D (= Berechtigung wird durch die Datenbank festgestellt) und F (= Berechtigung wird durch Dateisystem festgestellt).
- Das fünfte Zeichen gibt die Schreibberechtigung an. Gültige Werte sind B (= geblockter Zugriff) und F (Berechtigung wird durch Dateisystem festgestellt).
- Das sechste Zeichen stellt die Wiederherstellungsoptionen dar. Gültige Werte sind Y (= DB2 unterstützt Zeitpunkt wiederherstellung von Dateien, auf die in dieser Spalte verwiesen wird) und N (= keine Unterstützung).
- Das siebte Zeichen stellt die Aktion dar, die ausgeführt werden soll, wenn die Verbindung der Datendatei aufgehoben wird. Gültige Werte sind R (= Wiederherstellen) und D (= Datei löschen).

**IXFCREF**

Wenn die Spalte Teil einer Hierarchie ist, gibt dieses Feld an, ob es sich bei der Spalte um eine Datenspalte (D) oder um eine Referenzspalte (R) handelt.

**IXFCNDIM**

Die Anzahl der Dimensionen in der Spalte. Matrizen werden in dieser Version von PC/IXF nicht unterstützt. Dieses Feld muss daher eine Zeichen-darstellung eines ganzzahligen Nullwerts enthalten.

**IXFCDSIZ**

Die Größe bzw. der Bereich jeder Dimension. Die Länge dieses Feldes beträgt 5 Byte pro Dimension. Da Matrizen nicht unterstützt werden

(d. h. die Anzahl der Dimensionen Null sein muss), hat dieses Feld die Länge Null und ist eigentlich gar nicht vorhanden.

#### DATENSATZ

FELDDNAME	LÄNGE	TYP	KOMMENTARE
-----	-----	-----	-----
IXFDRECL	06 BYTE	CHARACTER	Satzlänge
IXFDRECT	01 BYTE	CHARACTER	Satztyp = "D"
IXFDRID	03 BYTE	CHARACTER	Satzkennung "D"
IXFDFIL1	04 BYTE	CHARACTER	reserviert
IXFDCOLS	variabel	variabel	Spaltendaten

Die folgenden Felder sind in Datensätzen enthalten:

#### IXFDRECL

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Jeder D-Datensatz muss lang genug für alle signifikanten Daten für das aktuelle Vorkommen der letzten im Datensatz gespeicherten Datenspalte sein.

#### IXFDRECT

Der IXF-Satztyp, der für diesen Datensatz auf D gesetzt ist, um anzugeben, dass der Datensatz Datenwerte für die Tabelle enthält.

#### IXFDRID

Die Satzkenkung, die einen bestimmten D-Datensatz in der Folge von mehreren D-Datensätzen angibt, die eine Datenzeile bilden. Für den ersten D-Datensatz in einer Datenzeile enthält dieses Feld den Wert Eins. Für den zweiten D-Datensatz in einer Datenzeile enthält dieses Feld den Wert Zwei usw. In jeder Datenzeile müssen alle D-Satzkennungen, die in den C-Datensätzen gesendet werden, auch vorhanden sein.

#### IXFDFIL1

Zusatzfeld, das vier Leerzeichen enthält. Dient als Entsprechung zu reservierten Feldern in Host-IXF-Dateien und bietet Platz für ein mögliches DBCS-Startzeichen.

#### IXFDCOLS

Der Bereich für Spaltendaten. Der Datenbereich eines Datensatzes (D-Satzes) besteht aus mindestens einem Spalteneintrag. Es gibt einen Spalteneintrag für jeden Spaltendeskriptorsatz, der die gleiche D-Satzkennung hat wie der D-Satz. Im D-Satz wird die Anfangsposition der Spalteneinträge durch den IXFCPOSN-Wert in C-Sätzen angegeben.

Das Format der Spalteneingabedaten hängt davon ab, ob die Spalte Nullwerte enthalten darf:

- Wenn die Spalte Nullwerte enthalten darf (das IXFCNULL-Feld auf Y gesetzt ist), enthalten die Spalteneingabedaten einen Nullanzeiger. Wenn die Spalte nicht Null ist, folgen auf den Anzeiger datentypspezifische Informationen, einschließlich des eigentlichen Datenbankwertes. Der Nullanzeiger ist ein 2 Byte großer Wert, der für nicht Null auf x'0000' und für Null auf x'FFFF' gesetzt wird.
- Wenn die Spalte keine Nullwerte enthalten darf, enthalten die Spalteneingabedaten nur datentypspezifische Informationen, einschließlich des tatsächlichen Datenbankwertes.

Für Datentypen mit variabler Länge schließen die datentypspezifischen Informationen einen aktuellen Längenanzeiger ein. Die aktuellen Längenanzeiger sind 2 Byte große ganze Zahlen in einem Format, das vom IXFTMFRM-Feld festgelegt wird.

Die Länge des Datenbereichs eines D-Datensatzes darf 32 771 Byte nicht überschreiten.

#### ANWENDUNGSDATENSATZ

FELDDNAME	LÄNGE	TYP	KOMMENTARE
-----	-----	-----	-----
IXFARECL	06 BYTE	CHARACTER	Satzlänge
IXFARECT	01 BYTE	CHARACTER	Satztyp = "A"
IXFAPPID	12 BYTE	CHARACTER	Anwendungskennzeichen
IXFADATA	variabel	variabel	anwendungsspezifische Daten

Die folgenden Felder sind in Anwendungsdatensätzen enthalten:

#### IXFARECL

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muss lang genug sein für das gesamte IXFAPPID-Feld.

#### IXFARECT

Der IXF-Satztyp, der für diesen Datensatz auf A gesetzt ist, um anzugeben, dass es sich um einen Anwendungsdatensatz handelt. Diese Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

#### IXFAPPID

Das Anwendungskennzeichen, das die Anwendung angibt, die den A-Datensatz erstellt. Vom Datenbankmanager erstellte PC/IXF-Dateien können A-Datensätze enthalten, in denen die ersten sechs Zeichen dieses Feldes auf eine Konstante gesetzt sind, die den Datenbankmanager angibt, während die letzten 6 Zeichen das Release oder die Version des Datenbankmanagers oder einer anderen Anwendung angeben, der bzw. die den A-Datensatz schreibt.

#### IXFADATA

Dieses Feld enthält anwendungsabhängige Ergänzungsdaten, deren Format und Inhalt nur dem Programm bekannt ist, das den A-Datensatz erstellt, sowie anderen Anwendungen, die den A-Datensatz wahrscheinlich verarbeiten.

#### DB2-INDEXSATZ

FELDDNAME	LÄNGE	TYP	KOMMENTARE
-----	-----	-----	-----
IXFARECL	006 BYTE	CHARACTER	Satzlänge
IXFARECT	001 BYTE	CHARACTER	Satztyp = "A"
IXFAPPID	012 BYTE	CHARACTER	Anwendungskennzeichen = 'DB2 02.00'
IXFAITYP	001 BYTE	CHARACTER	Anwendungsspezifischer Datentyp = 'I'
IXFADATE	008 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Daten
IXFATIME	006 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Zeit
IXFANDXL	002 BYTE	SHORT INT	Länge des Indexnamens
IXFANDXN	256 BYTE	CHARACTER	Name des Indexes
IXFANCL	002 BYTE	SHORT INT	Länge des Namens des Indexerstellers

IXFANCN	256 BYTE	CHARACTER	Name des Indexerstellers
IXFATABL	002 BYTE	SHORT INT	Länge des Tabellennamens
IXFATABN	256 BYTE	CHARACTER	Tabellenname
IXFATCL	002 BYTE	SHORT INT	Länge des Namens des Tabellenerstellers
IXFATCN	256 BYTE	CHARACTER	Name des Tabellenerstellers
IXFAUNIQ	001 BYTE	CHARACTER	Eindeutige Regel
IXFACCNT	002 BYTE	CHARACTER	Spaltenanzahl
IXFAREVS	001 BYTE	CHARACTER	Markierung für Rückwärtsdurchsuchen zulassen
IXFAPCTF	002 BYTE	CHARACTER	Wert für PCTFREE
IXFAPCTU	002 BYTE	CHARACTER	Wert für MINPCTUSED (Prozentsatz des auf Indexseiten freizuhaltenen Minimums an verwendetem Speicherplatz)
IXFAEXTI	001 BYTE	CHARACTER	reserviert
IXFACNML	002 BYTE	SHORT INT	Länge des Namens der Spalten
IXFACOLN	variabel	CHARACTER	Name der Spalten im Index

Für jeden benutzerdefinierten Index wird 1 Datensatz diesen Typs angegeben. Die Position dieses Datensatzes befindet sich nach allen C-Datensätzen für die Tabelle. DB2-Indexsätze enthalten die folgenden Felder:

#### **IXFARECL**

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muss lang genug sein für das gesamte IXFAPPID-Feld.

#### **IXFARECT**

Der IXF-Satztyp, der für diesen Datensatz auf A gesetzt ist, um anzugeben, dass es sich um einen Anwendungsdatensatz handelt. Diese Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

#### **IXFAPPID**

Das Anwendungskennzeichen, das DB2 als die Anwendung angibt, die diesen A-Datensatz erstellt.

#### **IXFAITYP**

Gibt an, dass dies der Subtyp "I" von DB2-Anwendungssätzen ist.

#### **IXFADATE**

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjjmmtt*. Dieses Feld muss denselben Wert wie das IXFHDATE-Feld aufweisen.

#### **IXFATIME**

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld muss denselben Wert wie das IXFHTIME-Feld aufweisen.

#### **IXFANDXL**

Die Länge des Indexnamens im IXFANDXN-Feld in Byte.

#### **IXFANDXN**

Der Name des Indexes.

#### **IXFANCL**

Die Länge des Namens des Indexerstellers im IXFANCN-Feld in Byte.

#### **IXFANCN**

Der Name des Indexerstellers.

#### **IXFATABL**

Die Länge des Tabellennamens im IXFATABN-Feld in Byte.

**IXFATABN**

Der Name der Tabelle.

**IXFATCL**

Die Länge des Namens des Tabellenerstellers im IXFATCN-Feld in Byte.

**IXFATCN**

Der Name des Tabellenerstellers.

**IXFAUNIQ**

Gibt den Indextyp an. Gültige Werte sind P (= Primärschlüssel), U (= eindeutiger Index) und D (= nicht eindeutiger Index).

**IXFACCNT**

Gibt die Anzahl der Spalten in der Indexdefinition an.

**IXFAREVS**

Gibt an, ob das Rückwärtsdurchsuchen dieses Indexes zulässig ist. Gültige Werte sind Y (= zulässig) und N (= nicht zulässig).

**IXFAPCTF**

Gibt den Prozentsatz der freizuhaltenden Indexseiten an. Gültige Werte liegen zwischen -1 und 99. Wenn der Wert -1 oder Null angegeben wird, wird der Systemstandardwert verwendet.

**IXFAPCTU**

Gibt den Mindestprozentsatz für den Speicherplatz an, der auf Indexseiten freigehalten werden muss, bevor zwei Indexseiten zusammengeführt werden können. Gültige Werte liegen zwischen 00 und 99.

**IXFAEXTI**

Zur zukünftigen Verwendung reserviert.

**IXFACNML**

Die Länge der Spaltennamen im IXFACOLN-Feld in Byte.

**IXFACOLN**

Die Namen der Spalten, die Bestandteil dieses Indexes sind. Gültige Werte haben das Format *+name-name...*. Hierbei gibt + eine aufsteigende Sortierung der Spalte und - eine absteigende Sortierung der Spalte an.

**DB2-HIERARCHIESATZ**

FELDDNAME	LÄNGE	TYP	KOMMENTARE
-----	-----	-----	-----
IXFARECL	006 BYTE	CHARACTER	Satzlänge
IXFARECT	001 BYTE	CHARACTER	Satztyp = "A"
IXFAPPID	012 BYTE	CHARACTER	Anwendungskennzeichen = 'DB2 02.00'
IXFAXTYP	001 BYTE	CHARACTER	Anwendungsspezifischer Datentyp = 'X'
IXFADATE	008 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Daten
IXFATIME	006 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Zeit
IXFAYCNT	010 BYTE	CHARACTER	"Y"-Satzzählung für diese Hierarchie
IXFAYSTR	010 BYTE	CHARACTER	Anfangsspalte dieser Hierarchie

Zur Beschreibung einer Hierarchie wird 1 Datensatz diesen Typs verwendet. Alle Sätze für untergeordnete Tabellen (siehe unten) müssen unmittelbar auf den Hierarchiesatz folgen. Die Hierarchiesätze müssen auf die C-Datensätze für die Tabelle folgen. DB2-Hierarchiesätze enthalten die folgenden Felder:

**IXFARECL**

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des

PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muss lang genug sein für das gesamte IXFAPPID-Feld.

#### IXFARECT

Der IXF-Satztyp, der für diesen Datensatz auf A gesetzt ist, um anzugeben, dass es sich um einen Anwendungsdatensatz handelt. Diese Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

#### IXFAPPID

Das Anwendungskennzeichen, das DB2 als die Anwendung angibt, die diesen A-Datensatz erstellt.

#### IXFAXTYP

Gibt an, dass dies der Subtyp "X" von DB2-Anwendungssätzen ist.

#### IXFADATE

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjjmmtt*. Dieses Feld muss denselben Wert wie das IXFHDATE-Feld aufweisen.

#### IXFATIME

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld muss denselben Wert wie das IXFHTIME-Feld aufweisen.

#### IXFAYCNT

Gibt die Anzahl der Sätze für untergeordnete Tabellen an, die nach diesem Hierarchiesatz erwartet werden.

#### IXFAWSTR

Gibt den Index der Sätze für untergeordnete Tabellen am Anfang der exportierten Daten an. Wurde der Export einer Hierarchie von einer untergeordneten Tabelle aus gestartet, die nicht die Stammtabelle ist, werden alle übergeordneten Tabellen dieser untergeordneten Tabelle exportiert. Die Position dieser untergeordneten Tabelle innerhalb der IXF-Datei wird ebenfalls in diesem Feld gespeichert. Der erste X-Datensatz stellt die Spalte mit einem Index 0 dar.

#### DB2-SATZ FÜR UNTERGEORDNETE TABELLE

FELDDNAME	LÄNGE	TYP	KOMMENTARE
IXFARECL	006 BYTE	CHARACTER	Satzlänge
IXFARECT	001 BYTE	CHARACTER	Satztyp = "A"
IXFAPPID	012 BYTE	CHARACTER	Anwendungskennzeichen = 'DB2 02.00'
IXFAYTYP	001 BYTE	CHARACTER	Anwendungsspezifischer Datentyp = 'Y'
IXFADATE	008 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Daten
IXFATIME	006 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Zeit
IXFASCHL	003 BYTE	CHARACTER	Länge des Typschemennamens
IXFASCHN	256 BYTE	CHARACTER	Typschemenname
IXFATYPL	003 BYTE	CHARACTER	Länge des Typnamens
IXFATYPN	256 BYTE	CHARACTER	Typname
IXFATABL	003 BYTE	CHARACTER	Länge des Tabellennamens
IXFATABN	256 BYTE	CHARACTER	Tabellenname
IXFAPNDX	010 BYTE	CHARACTER	Index der untergeordneten Tabelle für übergeordnete Tabelle
IXFASNDX	005 BYTE	CHARACTER	Beginn Spaltenindex der aktuellen Tabelle
IXFAENDX	005 BYTE	CHARACTER	Ende Spaltenindex der aktuellen Tabelle

Zur Beschreibung einer untergeordneten Tabelle als Teil einer Hierarchie wird 1 Datensatz diesen Typs verwendet. Alle Sätze für untergeordnete Tabellen, die zu einer Hierarchie gehören, müssen zusammen gespeichert werden und unmittelbar auf den entsprechenden Hierarchiesatz folgen. Eine untergeordnete Tabelle besteht aus einer oder mehreren Spalten. Jede Spalte wird in einem Spaltensatz beschrieben. Jede Spalte in einer untergeordneten Tabelle muss durch eine Reihe von aufeinander folgenden C-Datensätzen beschrieben werden. DB2-Sätze für untergeordnete Tabellen enthalten die folgenden Felder:

#### **IXFARECL**

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muss lang genug sein für das gesamte IXFAPPID-Feld.

#### **IXFARECT**

Der IXF-Satztyp, der für diesen Datensatz auf A gesetzt ist, um anzugeben, dass es sich um einen Anwendungsdatensatz handelt. Diese Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

#### **IXFAPPID**

Das Anwendungskennzeichen, das DB2 als die Anwendung angibt, die diesen A-Datensatz erstellt.

#### **IXFAYTYP**

Gibt an, dass dies der Subtyp "Y" von DB2-Anwendungssätzen ist.

#### **IXFADATE**

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjjmmtt*. Dieses Feld muss denselben Wert wie das IXFHDATE-Feld aufweisen.

#### **IXFATIME**

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld muss denselben Wert wie das IXFHTIME-Feld aufweisen.

#### **IXFASCHL**

Die Länge des Schemennamens für die untergeordnete Tabelle im IXFASCHN-Feld in Byte.

#### **IXFASCHN**

Der Name des Schemas für die untergeordnete Tabelle.

#### **IXFATYPL**

Die Länge des Namens der untergeordneten Tabelle im IXFATYPN-Feld in Byte.

#### **IXFATYPN**

Der Name der untergeordneten Tabelle.

#### **IXFATABL**

Die Länge des Tabellennamens im IXFATABN-Feld in Byte.

#### **IXFATABN**

Der Name der Tabelle.

#### **IXFAPNDX**

Der Eintragsindex für die untergeordnete Tabelle der übergeordneten Tabelle. Bildet diese untergeordnete Tabelle den Stamm einer Hierarchie, enthält dieses Feld den Wert -1.

## IXFASNDX

Startindex der Spaltensätze, aus denen diese untergeordnete Tabelle besteht.

## IXFAENDX

Endindex der Spaltensätze, aus denen diese untergeordnete Tabelle besteht.

### DB2-FORTSETZUNGSSATZ

FELDDNAME	LÄNGE	TYP	KOMMENTARE
IXFARECL	006 BYTE	CHARACTER	Satzlänge
IXFARECT	001 BYTE	CHARACTER	Satztyp = "A"
IXFAPPID	012 BYTE	CHARACTER	Anwendungskennzeichen = 'DB2 02.00'
IXFACTYP	001 BYTE	CHARACTER	Anwendungsspezifischer Datentyp = "C"
IXFADATE	008 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Daten
IXFATIME	006 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Zeit
IXFALAST	002 BYTE	SHORT INT	Datenträgernummer der ersten Diskette
IXFATHIS	002 BYTE	SHORT INT	Datenträgernummer dieser Diskette
IXFANEXT	002 BYTE	SHORT INT	Datenträgernummer der nächsten Diskette

Dieser Datensatz befindet sich am Ende jeder Datei, die Bestandteil einer auf mehrere Datenträger verteilten IXF-Datei ist, sofern es sich bei dem aktuellen Datenträger nicht um den letzten Datenträger handelt. Außerdem ist dieser Datensatz am Beginn jeder Datei zu finden, die Bestandteil einer auf mehrere Datenträger verteilten IXF-Datei ist, sofern es sich bei dem aktuellen Datenträger nicht um den ersten Datenträger handelt. Der Zweck dieses Datensatzes ist die Einhaltung der Dateireihenfolge. DB2-Fortsetzungssätze enthalten die folgenden Felder:

### IXFARECL

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muss lang genug sein für das gesamte IXFAPPID-Feld.

### IXFARECT

Der IXF-Satztyp, der für diesen Datensatz auf A gesetzt ist, um anzugeben, dass es sich um einen Anwendungsdatensatz handelt. Diese Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

### IXFAPPID

Das Anwendungskennzeichen, das DB2 als die Anwendung angibt, die diesen A-Datensatz erstellt.

### IXFACTYP

Gibt an, dass dies der Subtyp "C" von DB2-Anwendungssätzen ist.

### IXFADATE

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjmmmtt*. Dieses Feld muss denselben Wert wie das IXFHDATE-Feld aufweisen.

### IXFATIME

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld muss denselben Wert wie das IXFHTIME-Feld aufweisen.

### IXFALAST

Dieses Feld ist ein Feld für Binärdaten im Little-Endian-Format. Der Wert sollte um 1 kleiner als der Wert im IXFATHIS-Feld sein.

### IXFATHIS

Dieses Feld ist ein Feld für Binärdaten im Little-Endian-Format. Der Wert in diesem Feld sollte auf Fortsetzungsdatenträgern ebenfalls fortlaufend sein. Der erste Datenträger hat den Wert 1.

### IXFANEXT

Dieses Feld ist ein Feld für Binärdaten im Little-Endian-Format. Der Wert sollte um 1 größer als der Wert im IXFATHIS-Feld sein, es sei denn, der Datensatz befindet sich am Dateianfang. In diesem Fall sollte der Wert 0 lauten.

### DB2-BEENDIGUNGSSATZ

FELDNAME	LÄNGE	TYP	KOMMENTARE
IXFARECL	006 BYTE	CHARACTER	Satzlänge
IXFARECT	001 BYTE	CHARACTER	Satztyp = "A"
IXFAPPID	012 BYTE	CHARACTER	Anwendungskennzeichen = 'DB2 02.00'
IXFAETYP	001 BYTE	CHARACTER	Anwendungsspezifischer Datentyp = 'E'
IXFADATE	008 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Daten
IXFATIME	006 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Zeit

Dieser Datensatz ist das Dateiendezeichen, das sich am Ende einer IXF-Datei befindet. DB2-Beendigungssätze enthalten die folgenden Felder:

### IXFARECL

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muss lang genug sein für das gesamte IXFAPPID-Feld.

### IXFARECT

Der IXF-Satztyp, der für diesen Datensatz auf A gesetzt ist, um anzugeben, dass es sich um einen Anwendungsdatensatz handelt. Diese Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

### IXFAPPID

Das Anwendungskennzeichen, das DB2 als die Anwendung angibt, die diesen A-Datensatz erstellt.

### IXFAETYP

Gibt an, dass dies der Subtyp "E" von DB2-Anwendungssätzen ist.

### IXFADATE

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjjmmtt*. Dieses Feld muss denselben Wert wie das IXFHDATE-Feld aufweisen.

### IXFATIME

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld muss denselben Wert wie das IXFHTIME-Feld aufweisen.

### DB2-IDENTITÄTSSATZ

FELDNAME	LÄNGE	TYP	KOMMENTARE
IXFARECL	06 BYTE	CHARACTER	Satzlänge
IXFARECT	01 BYTE	CHARACTER	Satztyp = "A"
IXFAPPID	12 BYTE	CHARACTER	Anwendungskennzeichen
IXFATYPE	01 BYTE	CHARACTER	anwendungsspezifischer Satztyp = "S"

IXFADATE	08 BYTE	CHARACTER	Erstellungsdatum des Anwendungssatzes
IXFATIME	06 BYTE	CHARACTER	Erstellungszeit des Anwendungssatzes
IXFACOLN	06 BYTE	CHARACTER	Spaltennummer der Identitätsspalte
IXFAITYP	01 BYTE	CHARACTER	GENERATED ALWAYS ("Y" oder "N")
IXFASTRT	33 BYTE	CHARACTER	START AT-Wert für Identitätsspalte
IXFAINCR	33 BYTE	CHARACTER	INCREMENT BY-Wert für Identitätsspalte
IXFACACH	10 BYTE	CHARACTER	CACHE-Wert für Identitätsspalte
IXFAMINV	33-BYTE	CHARACTER	MINVALUE-Wert für Identitätsspalte
IXFAMAXV	33-BYTE	CHARACTER	MAXVALUE-Wert für Identitätsspalte
IXFACYCL	01-BYTE	CHARACTER	CYCLE-Wert für Identitätsspalte ('Y' oder 'N')
IXFAORDR	01-BYTE	CHARACTER	ORDER-Wert für Identitätsspalte ('Y' oder 'N')
IXFARMRL	03-BYTE	CHARACTER	Länge der Anmerkung für Identitätsspalte
IXFARMRK	254-BYTE	CHARACTER	Wert der Anmerkung für Identitätsspalte

DB2-Identitätssätze enthalten die folgenden Felder:

#### **IXFARECL**

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muss lang genug sein für das gesamte IXFAPPID-Feld.

#### **IXFARECT**

Der IXF-Satztyp, der für diesen Datensatz auf A gesetzt ist, um anzugeben, dass es sich um einen Anwendungsdatensatz handelt. Diese Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

#### **IXFAPPID**

Das Anwendungskennzeichen, das DB2 als die Anwendung angibt, die diesen A-Datensatz erstellt.

#### **IXFATYPE**

Gibt den anwendungsspezifischen Satztyp an. Dieses Feld sollte immer den Wert "S" haben.

#### **IXFADATE**

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjjmmtt*. Dieses Feld muss denselben Wert wie das IXFHDATE-Feld aufweisen.

#### **IXFATIME**

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld muss denselben Wert wie das IXFHTIME-Feld aufweisen.

#### **IXFACOLN**

Die Spaltennummer der Identitätsspalte in der Tabelle.

#### **IXFAITYP**

Der Typ der Identitätsspalte. Der Wert "Y" gibt an, dass die Identitätsspalte als GENERATED ALWAYS definiert ist, d. h. in jedem Fall generiert wird. Alle anderen Werte werden so interpretiert, dass die Spalte als GENERATED BY DEFAULT definiert ist.

#### **IXFASTRT**

Der START AT-Wert für die Identitätsspalte, der zum Zeitpunkt der Tabellenerstellung an die Anweisung CREATE TABLE übergeben wurde.

#### **IXFAINCR**

Der INCREMENT BY-Wert für die Identitätsspalte, der zum Zeitpunkt der Tabellenerstellung an die Anweisung CREATE TABLE übergeben wurde.

### IXFACACH

Der CACHE-Wert für die Identitätsspalte, der zum Zeitpunkt der Tabellenerstellung an die Anweisung CREATE TABLE übergeben wurde. Der Wert "1" entspricht der Option NO CACHE.

### IXFAMINV

Der MINVALUE-Wert für die Identitätsspalte, der zum Zeitpunkt der Tabellenerstellung an die Anweisung CREATE TABLE übergeben wurde.

### IXFAMAXV

Der MAXVALUE-Wert für die Identitätsspalte, der zum Zeitpunkt der Tabellenerstellung an die Anweisung CREATE TABLE übergeben wurde.

### IXFACYCL

Der CYCLE-Wert für die Identitätsspalte, der zum Zeitpunkt der Tabellenerstellung an die Anweisung CREATE TABLE übergeben wurde. Der Wert "Y" entspricht der Option CYCLE. Alle anderen Werte entsprechen NO CYCLE.

### IXFAORDR

Der ORDER-Wert für die Identitätsspalte, der zum Zeitpunkt der Tabellenerstellung an die Anweisung CREATE TABLE übergeben wurde. Der Wert "Y" entspricht der Option ORDER. Alle anderen Werte entsprechen NO ORDER.

### IXFARMRL

Die Länge (in Byte) der Anmerkung im Feld IXFARMRK.

### IXFARMRK

Dies ist die vom Benutzer eingegebene Anmerkung, die der Identitätsspalte zugeordnet ist. Dieses Feld dient nur zur Information. Dieses Feld wird vom Datenbankmanager beim Importieren von Daten nicht verwendet.

#### Zugehörige Referenzen:

- „PC/IXF-Datentypen“ auf Seite 304
- „Beschreibungen von PC/IXF-Datentypen“ auf Seite 311

## PC/IXF-Datentypen

Tabelle 18. PC/IXF-Datentypen

Name	IXFCTYPE-Wert	Beschreibung
BIGINT	492	Eine 8 Byte große ganze Zahl in dem von IXFTMFRM festgelegten Format. Stellt eine ganze Zahl zwischen -9 223 372 036 854 775 808 und 9 223 372 036 854 775 807 dar. IXFCSBCP und IXFCDBCP sind nicht signifikant und müssen Null sein. IXFCLENG wird nicht verwendet und muss Leerzeichen enthalten.

Tabelle 18. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
BLOB, CLOB	404, 408	<p>Eine Zeichenfolge variabler Länge. Die maximale Länge der Zeichenfolge ist im IXFLENG-Feld des Spaltendeskriptorsatzes enthalten, und sie darf 32 767 Byte nicht überschreiten. Der Zeichenfolge selbst wird ein aktueller Längenanzeiger vorangestellt, bei dem es sich um eine 4 Byte große ganze Zahl handelt, die die Länge der Zeichenfolge in Byte angibt. Die Zeichenfolge liegt in der Codepage vor, die in IXFCSBCP angegeben ist.</p> <p>Folgendes gilt nur für BLOBs: Wenn IXFCSBCP Null ist, besteht die Zeichenfolge aus Bitdaten und darf von keinem Umsetzungsprogramm umgesetzt werden.</p> <p>Folgendes gilt nur für CLOBs: Wenn IXFCDBCP ungleich Null ist, kann die Zeichenfolge auch Doppelbytezeichen in der Codepage enthalten, die in IXFCDBCP angegeben ist.</p>
BLOB_LOCATION_SPECIFIER, BLOB_LOCATION_SPECIFIER und DBCLOB_LOCATION_SPECIFIER	960, 964, 968	<p>Ein Feld fester Länge, das nicht größer als 255 Byte sein kann. Die LOB-Positionskennung (LLS) liegt in der Codepage vor, die in IXFCSBCP angegeben ist. Wenn IXFCSBCP Null ist, besteht die LOB-Positionskennung aus Bitdaten und darf von keinem Umsetzungsprogramm umgesetzt werden. Wenn IXFCDBCP ungleich Null ist, kann die Zeichenfolge auch Doppelbytezeichen in der Codepage enthalten, die in IXFCDBCP angegeben ist.</p> <p>Da die Länge der LOB-Positionskennung in IXFLENG gespeichert ist, geht die tatsächliche Länge des ursprünglichen großen Objekts verloren. PC/IXF-Dateien mit Spalten dieses Typs dürfen nicht zur erneuten Erstellung des LOB-Feldes verwendet werden, weil das LOB-Feld mit der Länge der LOB-Positionskennung erstellt wird.</p>

Tabelle 18. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
BLOB_FILE, CLOB_FILE, DBCLOB_FILE	916, 920, 924	<p>Ein Feld fester Länge, das eine SQLFILE-Struktur enthält, bei der die Felder <i>name_length</i> und <i>name</i> ausgefüllt sind. Die Länge der Struktur ist im IXFCLENG-Feld des Spaltendeskriptorsatzes enthalten und darf 255 Byte nicht überschreiten. Der Dateiname liegt in der Codepage vor, die in IXFCSBCP angegeben ist. Wenn IXFCDBCP ungleich Null ist, kann der Dateiname auch Doppelbytezeichen in der Codepage enthalten, die in IXFCDBCP angegeben ist. Wenn IXFCSBCP Null ist, besteht der Dateiname aus Bitdaten und darf von keinem Umsetzungsprogramm umgesetzt werden.</p> <p>Da die Länge der Struktur in IXFCLENG gespeichert ist, geht die eigentliche Länge des ursprünglichen LOBs verloren. IXF-Dateien mit Spalten des Typs BLOB_FILE, CLOB_FILE oder DBCLOB_FILE dürfen zur erneuten Erstellung des LOB-Feldes nicht verwendet werden, da das LOB-Feld mit der Länge <i>sql_lobfile_len</i> erstellt wird.</p>
CHAR	452	<p>Eine Zeichenfolge mit fester Länge. Die Länge der Zeichenfolge ist im IXFCLENG-Feld des Spaltendeskriptorsatzes enthalten, und sie darf 254 Byte nicht überschreiten. Die Zeichenfolge liegt in der Codepage vor, die in IXFCSBCP angegeben ist. Wenn IXFCDBCP ungleich Null ist, kann die Zeichenfolge auch Doppelbytezeichen in der Codepage enthalten, die in IXFCDBCP angegeben ist. Wenn IXFCSBCP Null ist, besteht die Zeichenfolge aus Bitdaten und darf von keinem Umsetzungsprogramm umgesetzt werden.</p>
DATE	384	<p>Ein Zeitpunkt gemäß gregorianischem Kalender. Jedes Datum ist eine 10 Byte große Zeichenfolge im ISO-Format: <i>jjjj-mm-tt</i>. Der Wertebereich für das Jahr liegt zwischen 0001 und 9999. Der Wertebereich für den Monat liegt zwischen 1 und 12. Der Bereich für den Tag ist 01 bis <i>n</i>, wobei <i>n</i> vom Monat abhängt. Dabei gelten die üblichen Regeln für die Tage eines Monats und Schaltjahre. Führende Nullen können bei keinem Teil weggelassen werden. IXFCLENG wird nicht verwendet und muss Leerzeichen enthalten. Gültige Zeichen für DATE sind in allen PC-ASCII-Codepages gleich. Daher sind IXFCSBCP und IXFCDBCP nicht signifikant und müssen Null sein.</p>

Tabelle 18. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
DBCLOB	412	<p>Eine Zeichenfolge variabler Länge aus Doppelbytezeichen. Das IXFCLENG-Feld im Spaltendeskriptorsatz gibt die maximale Anzahl von Doppelbytezeichen in der Zeichenfolge an, die 16 383 nicht überschreiten darf. Der Zeichenfolge selbst wird ein aktueller Längenanzeiger vorangestellt, bei dem es sich um eine 4 Byte große ganze Zahl handelt, die die Länge der Zeichenfolge in Doppelbytezeichen angibt (d. h. der Wert dieser ganzen Zahl ist die halbe Länge der Zeichenfolge in Byte). Die Zeichenfolge liegt in der DBCS-Codepage vor, wie mit IXFCDBCP im C-Datensatz angegeben. Da die Zeichenfolge nur Doppelbytezeichen enthält, muss IXFCSBCP Null sein. Es gibt keine DBCS-Startzeichen oder DBCS-Endezeichen.</p>
DECIMAL	484	<p>Eine gepackte Dezimalzahl mit der Genauigkeit P (wie durch die ersten drei Byte von IXFCLENG im Spaltendeskriptorsatz festgelegt) und der Anzahl S der Kommastellen (wie durch die letzten beiden Byte von IXFCLENG festgelegt). Die Länge einer gepackten Dezimalzahl in Byte ist <math>(P+2)/2</math>. Die Genauigkeit muss eine ungerade Zahl zwischen 1 und 31 (jeweils einschließlich) sein. Die gepackte Dezimalzahl liegt im internen Format vor, das mit IXFTMFRM angegeben wird, während die gepackte Dezimalzahl für den PC so definiert ist, dass sie mit der gepackten Dezimalzahl für das IBM System/370 identisch ist. IXFCSBCP und IXFCDBCP sind nicht signifikant und müssen Null sein.</p>
FLOATING POINT	480	<p>Entweder eine lange (8 Byte) oder kurze (4 Byte) Gleitkommazahl. Dies hängt davon ab, ob IXFCLENG auf 8 oder 4 gesetzt ist. Die Daten liegen im internen Maschinenformat vor, wie mit IXFTMFRM angegeben. IXFCSBCP und IXFCDBCP sind nicht signifikant und müssen Null sein. 4 Byte große Gleitkommazahlen werden vom Datenbankmanager nicht unterstützt.</p>

Tabelle 18. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
GRAPHIC	468	Eine Zeichenfolge fester Länge aus Doppelbytezeichen. Das IXFCLENG-Feld im Spaltendeskriptorsatz gibt die Anzahl von Doppelbytezeichen in der Zeichenfolge an und darf maximal den Wert 127 enthalten. Die eigentliche Länge der Zeichenfolge ist das Doppelte des Werts des IXFCLENG-Feldes in Byte. Die Zeichenfolge liegt in der DBCS-Codepage vor, wie mit IXFCDBCP im C-Datensatz angegeben. Da die Zeichenfolge nur Doppelbytezeichen enthält, muss IXFCSBCP Null sein. Es gibt keine DBCS-Startzeichen oder DBCS-Endezeichen.
INTEGER	496	Eine 4 Byte große ganze Zahl in dem von IXFTMFRM festgelegten Format. Stellt eine ganze Zahl zwischen -2 147 483 648 und +2 147 483 647 dar. IXFCSBCP und IXFCDBCP sind nicht signifikant und müssen Null sein. IXFCLENG wird nicht verwendet und muss Leerzeichen enthalten.
LONGVARCHAR	456	Eine Zeichenfolge variabler Länge. Die maximale Länge der Zeichenfolge ist im IXFCLENG-Feld des Spaltendeskriptorsatzes enthalten, und sie darf 32 767 Byte nicht überschreiten. Der Zeichenfolge selbst wird ein aktueller Längenanzeiger vorangestellt, bei dem es sich um eine 2 Byte große ganze Zahl handelt, die die Länge der Zeichenfolge in Byte angibt. Die Zeichenfolge liegt in der Codepage vor, die in IXFCSBCP angegeben ist. Wenn IXFCDBCP ungleich Null ist, kann die Zeichenfolge auch Doppelbytezeichen in der Codepage enthalten, die in IXFCDBCP angegeben ist. Wenn IXFCSBCP Null ist, besteht die Zeichenfolge aus Bitdaten und darf von keinem Umsetzungsprogramm umgesetzt werden.
LONG VARGRAPHIC	472	Eine Zeichenfolge variabler Länge aus Doppelbytezeichen. Das IXFCLENG-Feld im Spaltendeskriptorsatz gibt die maximale Anzahl von Doppelbytezeichen in der Zeichenfolge an, die 16 383 nicht überschreiten darf. Der Zeichenfolge selbst wird ein aktueller Längenanzeiger vorangestellt, bei dem es sich um eine 2 Byte große ganze Zahl handelt, die die Länge der Zeichenfolge in Doppelbytezeichen angibt (d. h. der Wert dieser ganzen Zahl ist die halbe Länge der Zeichenfolge in Byte). Die Zeichenfolge liegt in der DBCS-Codepage vor, wie mit IXFCDBCP im C-Datensatz angegeben. Da die Zeichenfolge nur Doppelbytezeichen enthält, muss IXFCSBCP Null sein. Es gibt keine DBCS-Startzeichen oder DBCS-Endezeichen.

Tabelle 18. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
SMALLINT	500	Eine 2 Byte große ganze Zahl in dem von IXFTMFRM festgelegten Format. Stellt eine ganze Zahl zwischen -32 768 und +32 767 dar. IXFCSBCP und IXFCDBCP sind nicht signifikant und müssen Null sein. IXFCLENG wird nicht verwendet und muss Leerzeichen enthalten.
TIME	388	Ein Zeitpunkt gemäß gregorianischem Kalender. Jede Zeitangabe ist eine 8 Byte große Zeichenfolge im ISO-Format: <i>ss.mm.ss</i> . Der Wertebereich für die Stunde liegt zwischen 0 und 24 und der Wertebereich für Minuten und Sekunden zwischen 0 und 59. Bei Stunde 24 sind die Minuten- und Sekundenangaben gleich Null. Der kleinste Zeitwert ist 00.00.00, der größte 24.00.00. Führende Nullen können bei keinem Teil weggelassen werden. IXFCLENG wird nicht verwendet und muss Leerzeichen enthalten. Gültige Zeichen für TIME sind in allen PC-ASCII-Codepages gleich. Daher sind IXFCSBCP und IXFCDBCP nicht signifikant und müssen Null sein.
TIMESTAMP	392	Das Datum und die Uhrzeit mit der Genauigkeit von Mikrosekunden. Jede Zeitmarke ist eine Zeichenfolge im Format <i>jjjj-mm-tt-ss.mm.ss.nnnnnn</i> (Jahr Monat Tag Stunde Minuten Sekunden Mikrosekunden). IXFCLENG wird nicht verwendet und muss Leerzeichen enthalten. Gültige Zeichen für TIMESTAMP sind in allen PC-ASCII-Codepages gleich. Daher sind IXFCSBCP und IXFCDBCP nicht signifikant und müssen Null sein.
VARCHAR	448	Eine Zeichenfolge variabler Länge. Die maximale Länge der Zeichenfolge in Byte ist im IXFCLENG-Feld des Spaltendes- kriptorsatzes enthalten und darf 254 Byte nicht überschreiten. Der Zeichenfolge selbst wird ein aktueller Längenanzeiger vorangestellt, bei dem es sich um eine 2 Byte große ganze Zahl handelt, die die Länge der Zeichenfolge in Byte angibt. Die Zeichenfolge liegt in der Codepage vor, die in IXFCSBCP angegeben ist. Wenn IXFCDBCP ungleich Null ist, kann die Zeichenfolge auch Doppelbytezeichen in der Codepage enthalten, die in IXFCDBCP angegeben ist. Wenn IXFCSBCP Null ist, besteht die Zeichenfolge aus Bitdaten und darf von keinem Umsetzungsprogramm umgesetzt werden.

Tabelle 18. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
VARGRAPHIC	464	Eine Zeichenfolge variabler Länge aus Doppelbytezeichen. Das IXFCLENG-Feld im Spaltendeskriptorsatz gibt die maximale Anzahl von Doppelbytezeichen in der Zeichenfolge an, die 127 nicht überschreiten darf. Der Zeichenfolge selbst wird ein aktueller Längenanzeiger vorangestellt, bei dem es sich um eine 2 Byte große ganze Zahl handelt, die die Länge der Zeichenfolge in Doppelbytezeichen angibt (d. h. der Wert dieser ganzen Zahl ist die halbe Länge der Zeichenfolge in Byte). Die Zeichenfolge liegt in der DBCS-Codepage vor, wie mit IXFCDBCP im C-Datensatz angegeben. Da die Zeichenfolge nur Doppelbytezeichen enthält, muss IXFCSBCP Null sein. Es gibt keine DBCS-Startzeichen oder DBCS-Endzeichen.

Nicht alle Kombinationen von IXFCSBCP- und IXFCDBCP-Werten für die PC/IXF-Zeichen- oder -Grafikspalten sind gültig. Eine PC/IXF-Zeichen- oder -Grafikspalte mit einer ungültigen Kombination aus IXFCSBCP und IXFCDBCP ist ein ungültiger Datentyp.

Tabelle 19. Gültige PC/IXF-Datentypen

PC/IXF-Datentyp	Gültige Paare (IXFCSBCP,IXFCDBCP)	Ungültige Paare (IXFCSBCP,IXFCDBCP)
CHAR, VARCHAR oder LONG VARCHAR	(0,0), (x,0) oder (x,y)	(0,y)
BLOB	(0,0)	(x,0), (0,y) oder (x,y)
CLOB	(x,0), (x,y)	(0,0), (0,y)
GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC oder DBCLOB	(0,y)	(0,0), (x,0) oder (x,y)
<b>Anmerkung:</b> x und y sind nicht 0.		

**Zugehörige Referenzen:**

- „PC/IXF-Satztypen“ auf Seite 287
- „Option FORCEIN“ auf Seite 320
- „Beschreibungen von PC/IXF-Datentypen“ auf Seite 311

## Beschreibungen von PC/IXF-Datentypen

Tabelle 20. Gültige Datentypformate für das PC/IXF-Dateiformat

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
BIGINT	Eine BIGINT-Spalte wird erstellt, die mit der Datenbankspalte identisch ist.	Eine Spalte eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich -9 223 372 036 854 775 808 bis 9 223 372 036 854 775 807 liegen.
BLOB	Eine PC/IXF-Spalte des Typs BLOB wird erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spaltendeskriptorsatz kopiert.	<p>Eine PC/IXF-Spalte des Typs CHAR, VARCHAR, LONG VARCHAR, BLOB, BLOB_FILE oder BLOB_LOCATION_SPECIFIER ist in folgenden Fällen zulässig:</p> <ul style="list-style-type: none"> <li>• Die Datenbankspalte weist die Markierung FOR BIT DATA auf.</li> <li>• Der Einzelbyte-Codepage-Wert der PC/IXF-Spalte stimmt mit der SBCS-CPGID der Datenbankspalte überein, und der Doppelbyte-Codepage-Wert der PC/IXF-Spalte entspricht Null oder der DBCS-CPGID der Datenbankspalte. Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC BLOB ist ebenfalls zulässig. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muss ihre Länge mit der maximalen Länge der Datenbankspalte kompatibel sein.</li> </ul>
CHAR	Eine PC/IXF-Spalte des Typs CHAR wird erstellt. Die Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den PC/IXF-Spaltendeskriptorsatz kopiert.	<p>Eine PC/IXF-Spalte des Typs CHAR, VARCHAR oder LONG VARCHAR ist in folgenden Fällen zulässig:</p> <ul style="list-style-type: none"> <li>• Die Datenbankspalte weist die Markierung FOR BIT DATA auf.</li> <li>• Der Einzelbyte-Codepage-Wert der PC/IXF-Spalte stimmt mit der SBCS-CPGID der Datenbankspalte überein, und der Doppelbyte-Codepage-Wert der PC/IXF-Spalte entspricht Null oder der DBCS-CPGID der Datenbankspalte.</li> </ul> <p>Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC ist ebenfalls zulässig, wenn die Datenbankspalte mit FOR BIT DATA markiert ist. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muss ihre Länge in jedem Fall mit der Länge der Datenbankspalte kompatibel sein. Die Daten werden bei Bedarf auf der rechten Seite mit Einzelbyteleerzeichen (x'20') aufgefüllt.</p>

Tabelle 20. Gültige Datentypformate für das PC/IXF-Dateiformat (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
CLOB	Eine PC/IXF-Spalte des Typs CLOB wird erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spaltendeskriptorsatz kopiert.	Eine PC/IXF-Spalte des Typs CHAR, VARCHAR, LONG VARCHAR, CLOB, CLOB_FILE oder CLOB_LOCATION_SPECIFIER ist zulässig, wenn der Einzelbyte-Codepage-Wert der PC/IXF-Spalte mit der SBCS-CPGID der Datenbankspalte identisch ist und der Doppelbyte-Codepage-Wert der PC/IXF-Spalte entweder Null ist oder der DBCS-CPGID der Datenbankspalte entspricht. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muss ihre Länge mit der maximalen Länge der Datenbankspalte kompatibel sein.
DATE	Eine DATE-Spalte wird erstellt, die mit der Datenbankspalte identisch ist.	Eine PC/IXF-Spalte des Typs DATE ist die übliche Eingabe. Das Dienstprogramm IMPORT versucht auch, Spalten in einem beliebigen Zeichentyp anzunehmen, sofern sie keine inkompatiblen Längen aufweisen. Die Zeichenspalte in der PC/IXF-Datei muss Datumswerte in einem Format enthalten, das mit dem Gebietscode der Zieldatenbank konsistent ist.
DBCLOB	Eine PC/IXF-Spalte des Typs DBCLOB wird erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spaltendeskriptorsatz kopiert.	Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC, DBCLOB, DBCLOB_FILE oder DBCLOB_LOCATION_SPECIFIER ist zulässig, wenn der Doppelbyte-Codepage-Wert der PC/IXF-Spalte mit dem der Datenbankspalte übereinstimmt. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muss ihre Länge mit der maximalen Länge der Datenbankspalte kompatibel sein.
DECIMAL	Eine DECIMAL-Spalte, die mit der Datenbankspalte identisch ist, wird erstellt. Die Genauigkeit und die Anzahl der Kommastellen werden im Spaltendeskriptorsatz gespeichert.	Eine Spalte eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich der DECIMAL-Spalte liegen, in die sie importiert werden.
FLOAT	Eine FLOAT-Spalte wird erstellt, die mit der Datenbankspalte identisch ist.	Eine Spalte eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Alle Werte liegen innerhalb des Bereichs.

Tabelle 20. Gültige Datentypformate für das PC/IXF-Dateiformat (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
GRAPHIC (nur DBCS)	Eine PC/IXF-Spalte des Typs GRAPHIC wird erstellt. Die Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spaltendeskriptorsatz kopiert.	Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC ist zulässig, wenn der Doppelbyte-Codepage-Wert der PC/IXF-Spalte mit dem der Datenbankspalte übereinstimmt. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muss ihre Länge mit der Länge der Datenbankspalte kompatibel sein. Die Daten werden bei Bedarf auf der rechten Seite mit Doppelbyteleerzeichen (x'8140') aufgefüllt.
INTEGER	Eine Spalte des Typs INTEGER wird erstellt, die mit der Datenbankspalte identisch ist.	Eine Spalte eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich -2 147 483 648 bis 2 147 483 647 liegen.
LONG VARCHAR	Eine PC/IXF-Spalte des Typs LONG VARCHAR wird erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spaltendeskriptorsatz kopiert.	<p>Eine PC/IXF-Spalte des Typs CHAR, VARCHAR oder LONG VARCHAR ist in folgenden Fällen zulässig:</p> <ul style="list-style-type: none"> <li>• Die Datenbankspalte weist die Markierung FOR BIT DATA auf.</li> <li>• Der Einzelbyte-Codepage-Wert der PC/IXF-Spalte stimmt mit der SBCS-CPGID der Datenbankspalte überein, und der Doppelbyte-Codepage-Wert der PC/IXF-Spalte entspricht Null oder der DBCS-CPGID der Datenbankspalte.</li> </ul> <p>Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC ist ebenfalls zulässig, wenn die Datenbankspalte mit FOR BIT DATA markiert ist. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muss ihre Länge in jedem Fall mit der maximalen Länge der Datenbankspalte kompatibel sein.</p>
LONG VARGRAPHIC (nur DBCS)	Eine PC/IXF-Spalte des Typs LONG VARGRAPHIC wird erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spaltendeskriptorsatz kopiert.	Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC ist zulässig, wenn der Doppelbyte-Codepage-Wert der PC/IXF-Spalte mit dem der Datenbankspalte übereinstimmt. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muss ihre Länge mit der maximalen Länge der Datenbankspalte kompatibel sein.

Tabelle 20. Gültige Datentypformate für das PC/IXF-Dateiformat (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
SMALLINT	Eine Spalte des Typs SMALLINT wird erstellt, die mit der Datenbankspalte identisch ist.	Eine Spalte eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich -32 768 bis 32 767 liegen.
TIME	Eine Spalte des Typs TIME wird erstellt, die mit der Datenbankspalte identisch ist.	Eine PC/IXF-Spalte des Typs TIME ist die übliche Eingabe. Das Dienstprogramm IMPORT versucht auch, Spalten in einem beliebigen Zeichentyp anzunehmen, sofern sie keine inkompatiblen Längen aufweisen. Die Zeichenspalte in der PC/IXF-Datei muss Zeitwerte in einem Format enthalten, das mit dem Gebietscode der Zieldatenbank konsistent ist.
TIMESTAMP	Eine Spalte des Typs TIMESTAMP wird erstellt, die mit der Datenbankspalte identisch ist.	Eine PC/IXF-Spalte des Typs TIMESTAMP ist die übliche Eingabe. Das Dienstprogramm IMPORT versucht auch, Spalten in einem beliebigen Zeichentyp anzunehmen, sofern sie keine inkompatiblen Längen aufweisen. Die Zeichenspalte in der PC/IXF-Datei muss Daten im Eingabeformat für Zeitmarken enthalten.
VARCHAR	Wenn die maximale Länge der Datenbankspalte $\leq$ 254 ist, wird eine PC/IXF-Spalte des Typs VARCHAR erstellt. Wenn die maximale Länge der Datenbankspalte $>$ 254 ist, wird eine PC/IXF-Spalte des Typs LONG VARCHAR erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spaltendeskriptorsatz kopiert.	Eine PC/IXF-Spalte des Typs CHAR, VARCHAR oder LONG VARCHAR ist in folgenden Fällen zulässig: <ul style="list-style-type: none"> <li>• Die Datenbankspalte weist die Markierung FOR BIT DATA auf.</li> <li>• Der Einzelbyte-Codepage-Wert der PC/IXF-Spalte stimmt mit der SBCS-CPGID der Datenbankspalte überein, und der Doppelbyte-Codepage-Wert der PC/IXF-Spalte entspricht Null oder der DBCS-CPGID der Datenbankspalte.</li> </ul> Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC ist ebenfalls zulässig, wenn die Datenbankspalte mit FOR BIT DATA markiert ist. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muss ihre Länge in jedem Fall mit der maximalen Länge der Datenbankspalte kompatibel sein.

Tabelle 20. Gültige Datentypformate für das PC/IXF-Dateiformat (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
VARGRAPHIC (nur DBCS)	Wenn die maximale Länge der Datenbankspalte <= 127 ist, wird eine PC/IXF-Spalte des Typs VARGRAPHIC erstellt. Wenn die maximale Länge der Datenbankspalte > 127 ist, wird eine PC/IXF-Spalte des Typs LONG VARGRAPHIC erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spaltendescriptorsatz kopiert.	Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC ist zulässig, wenn der Doppelbyte-Codepage-Wert der PC/IXF-Spalte mit dem der Datenbankspalte übereinstimmt. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muss ihre Länge mit der maximalen Länge der Datenbankspalte kompatibel sein.

**Zugehörige Referenzen:**

- „PC/IXF-Satztypen“ auf Seite 287
- „PC/IXF-Datentypen“ auf Seite 304

## Allgemeine Regeln für den Import von PC/IXF-Dateien in Datenbanken

Das Dienstprogramm IMPORT des Datenbankmanagers wendet beim Importieren einer PC/IXF-Datei in einer SBCS- oder DBCS-Umgebung die folgenden allgemeinen Regeln an:

- Das Dienstprogramm IMPORT akzeptiert nur Dateien im PC/IXF-Format (IXFHID = "IXF"). IXF-Dateien anderer Formate können nicht importiert werden.
- Das Dienstprogramm IMPORT weist eine PC/IXF-Datei mit mehr als 1024 Spalten zurück.
- Der Wert von IXFHSBCP im PC/IXF-H-Datensatz muss mit der SBCS-CPGID übereinstimmen, oder es muss eine Konvertierungstabelle zwischen IXFHSBCP/IXFHDBCP und der SBCS/DBCS-CPGID der Zieldatenbank geben. Der Wert von IXFHDBCP muss entweder "00000" sein oder dem Wert der DBCS-CPGID der Zieldatenbank entsprechen. Wenn eine dieser Bedingungen nicht erfüllt ist, weist das Dienstprogramm IMPORT die PC/IXF-Datei zurück, sofern nicht die Option FORCEIN angegeben ist.
- Ungültige Datentypen — neue Tabelle  
Der Import einer PC/IXF-Datei in eine *neue* Tabelle wird durch die Schlüsselwörter CREATE oder REPLACE\_CREATE des Befehls IMPORT angegeben. Wenn eine PC/IXF-Spalte mit einem ungültigen Datentyp für den Import in

eine neue Tabelle ausgewählt wird, wird das Dienstprogramm IMPORT beendet. Die gesamte PC/IXF-Datei wird zurückgewiesen, es wird keine Tabelle erstellt, und es werden keine Daten importiert.

- Ungültige Datentypen — vorhandene Tabelle

Der Import einer PC/IXF-Datei in eine *vorhandene* Tabelle wird durch die Schlüsselwörter INSERT oder INSERT\_UPDATE des Befehls IMPORT angegeben. Wenn eine PC/IXF-Spalte mit einem ungültigen Datentyp für den Import in eine vorhandene Tabelle ausgewählt wird, ist eine von zwei Aktionen möglich:

- Wenn die Zieltabellenspalte Nullwerte enthalten darf, werden alle Werte der ungültigen PC/IXF-Spalte ignoriert, und die Tabellenspaltenwerte werden auf NULL gesetzt.
- Wenn die Zieltabellenspalte keine Nullwerte enthalten darf, wird das Dienstprogramm IMPORT beendet. Die gesamte PC/IXF-Datei wird zurückgewiesen, und es werden keine Daten importiert. Die vorhandene Tabelle bleibt unverändert.

- Beim Import in eine neue Tabelle bewirken PC/IXF-Spalten, die Nullwerte enthalten dürfen, die Generierung von Datenbankspalten, die Nullwerte enthalten dürfen. PC/IXF-Spalten, die keine Nullwerte enthalten dürfen, bewirken die Generierung von Datenbankspalten, die keine Nullwerte enthalten dürfen.
- Eine PC/IXF-Spalte, die keine Nullwerte enthalten darf, kann in eine Datenbankspalte, die Nullwerte enthalten darf, importiert werden.
- Eine PC/IXF-Spalte, die Nullwerte enthalten darf, kann in eine Datenbankspalte, die keine Nullwerte enthalten darf, importiert werden. Wenn ein NULL-Wert in der PC/IXF-Spalte festgestellt wird, weist das Dienstprogramm IMPORT die Werte aller Spalten in der PC/IXF-Zeile zurück, die den NULL-Wert enthält (die ganze Zeile wird zurückgewiesen), und die Verarbeitung wird bei der nächsten PC/IXF-Zeile fortgesetzt. Das bedeutet, dass keine Daten aus einer PC/IXF-Zeile, die einen NULL-Wert enthält, importiert werden, wenn eine Zieltabellenspalte keine Nullwerte enthalten darf.
- Inkompatible Spalten — neue Tabelle

Wenn beim Import in eine *neue* Datenbanktabelle eine PC/IXF-Spalte ausgewählt wird, die mit der Zieldatenbankspalte inkompatibel ist, wird das Dienstprogramm IMPORT beendet. Die gesamte PC/IXF-Datei wird zurückgewiesen, es wird keine Tabelle erstellt, und es werden keine Daten importiert.

**Anmerkung:** Die IMPORT-Option FORCEIN erweitert den Bereich kompatibler Spalten.

- Inkompatible Spalten — vorhandene Tabelle

Wenn beim Import in eine *vorhandene* Datenbanktabelle eine PC/IXF-Spalte ausgewählt wird, die zur Zieldatenbankspalte inkompatibel ist, ist eine von zwei Aktionen möglich:

- Wenn die Zieltabellenspalte Nullwerte enthalten darf, werden alle Werte der PC/IXF-Spalte ignoriert, und die Tabellenspaltenwerte werden auf NULL gesetzt.
- Wenn die Zieltabellenspalte keine Nullwerte enthalten darf, wird das Dienstprogramm IMPORT beendet. Die gesamte PC/IXF-Datei wird zurückgewiesen, und es werden keine Daten importiert. Die vorhandene Tabelle bleibt unverändert.

**Anmerkung:** Die IMPORT-Option FORCEIN erweitert den Bereich kompatibler Spalten.

- Ungültige Werte  
Wenn beim Import ein PC/IXF-Spaltenwert festgestellt wird, der für die Zieldatenbankspalte ungültig ist, weist das Dienstprogramm IMPORT die Werte aller Spalten in der PC/IXF-Zeile mit dem ungültigen Wert zurück, d. h. die gesamte Zeile wird zurückgewiesen, und die Verarbeitung wird bei der nächste PC/IXF-Zeile fortgesetzt.
- Das Importieren oder Laden von PC/IXF-Dateien, die DBCS-Daten enthalten, erfordert, dass die entsprechenden Konvertierungsdateien (in `sqllib\conv`) auf der Client-Maschine installiert sind. Die Namen dieser Konvertierungsdateien enthalten die Nummern der Quellen- und der Ziel-Codepage. Die Erweiterung ist stets `.cnv`. Die Datei `09320943.cnv` beispielsweise enthält die Tabelle zur Konvertierung von Codepage 932 in 943.  
Wenn die Client-Maschine nicht über die entsprechenden Konvertierungsdateien verfügt, können sie von einer Server-Maschine in das Verzeichnis `sqllib\conv` auf der Client-Maschine kopiert werden. Achten Sie darauf, die Dateien von einer kompatiblen Plattform zu kopieren. Wird der Client beispielsweise unter einem UNIX-Betriebssystem ausgeführt, müssen Sie die Dateien von einem Server kopieren, der ebenfalls unter einem UNIX-Betriebssystem ausgeführt wird.

#### Zugehörige Referenzen:

- „PC/IXF-Datentypen“ auf Seite 304
- „Option FORCEIN“ auf Seite 320

## Datentypspezifische Regeln für den Import von PC/IXF-Dateien in Datenbanken

- Eine gültige numerische PC/IXF-Spalte kann in jede beliebige kompatible numerische Datenbankspalte importiert werden. PC/IXF-Spalten, die 4 Byte große Gleitkommadata enthalten, werden nicht importiert, weil dies ein ungültiger Datentyp ist.
- Datum-/Uhrzeitspalten der Datenbank können Werte aus entsprechenden PC/IXF-Datums-/Uhrzeitspalten (DATE, TIME und TIMESTAMP) sowie aus anderen PC/IXF-Zeichenspalten (CHAR, VARCHAR und LONG VARCHAR) akzeptieren, wobei jedoch die Einschränkungen hinsichtlich der Spaltenlänge und der Wertekompatibilität gelten.
- Eine gültige PC/IXF-Zeichenspalte (CHAR, VARCHAR oder LONG VARCHAR) kann immer in eine *vorhandene* Datenbankzeichenspalte mit der Markierung FOR BIT DATA importiert werden. Ansonsten gilt Folgendes:
  - IXFCSBCP und die SBCS-CPGID müssen übereinstimmen.
  - Es muss eine Konvertierungstabelle für IXFCSBCP/IXFCDBCP und SBCS/DBCS vorhanden sein.
  - Ein Satz muss ausschließlich aus Nullen bestehen (FOR BIT DATA).

Wenn IXFCSBCP nicht Null ist, muss der Wert von IXFCDBCP entweder Null sein oder mit der DBCS-CPGID der Zieldatenbankspalte übereinstimmen.

Wenn eine dieser Bedingungen nicht erfüllt ist, sind die PC/IXF-Spalten und die Datenbankspalten inkompatibel.

Beim Importieren einer gültigen PC/IXF-Zeichenspalte in eine *neue* Datenbanktabelle muss der Wert von IXFCSBCP Null sein oder mit der SBCS-CPGID der Datenbank übereinstimmen, oder es ist eine Konvertierungstabelle erforderlich. Wenn IXFCSBCP Null ist, muss auch IXFCDBCP Null sein (andernfalls hat die PC/IXF-Spalte einen ungültigen Datentyp). Von IMPORT wird eine Zeichenspalte mit der Markierung FOR BIT DATA in der neuen Tabelle erstellt. Wenn

IXFCSBCP nicht Null ist und mit der SBCS-CPGID der Datenbank übereinstimmt, muss der Wert von IXFCDBCP entweder Null sein oder mit der DBCS-CPGID der Datenbank übereinstimmen. In diesem Fall wird vom Dienstprogramm eine Zeichenspalte in der neuen Tabelle erstellt, wobei die SBCS- und DBCS-CPGID-Werte mit denen der Datenbank übereinstimmen. Wenn diese Bedingungen nicht erfüllt sind, sind die PC/IXF-Spalten und die Datenbankspalten inkompatibel.

Die Option FORCEIN kann verwendet werden, um Codepage-Gleichheitsüberprüfungen zu überschreiben. Eine PC/IXF-Zeichenspalte mit IXFCSBCP gleich Null und IXFCDBCP ungleich Null ist jedoch ein ungültiger Datentyp, der nicht importiert werden kann, auch dann nicht, wenn FORCEIN angegeben ist.

- Eine gültige PC/IXF-Grafikspalte (GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC) kann immer in eine *vorhandene* Datenbankzeichenspalte mit der Markierung FOR BIT DATA importiert werden, ist jedoch mit allen anderen Datenbankspalten inkompatibel. Die Option FORCEIN kann verwendet werden, um die Einschränkung abzuschwächen. Eine PC/IXF-Grafikspalte mit IXFCSBCP ungleich Null und IXFCDBCP gleich Null ist jedoch ein ungültiger Datentyp, der nicht importiert werden kann, auch dann nicht, wenn FORCEIN angegeben ist.

Beim Import einer gültigen PC/IXF-Grafikspalte in eine Datenbankgrafikspalte muss der Wert von IXFCDBCP mit der DBCS-CPGID der Zieldatenbankspalte übereinstimmen (das heißt, die Doppelbyte-Codepages der beiden Spalten müssen übereinstimmen).

- Wenn beim Import einer PC/IXF-Datei in eine vorhandene Datenbanktabelle eine Zeichenfolgespalte mit fester Länge (CHAR oder GRAPHIC) ausgewählt wird, deren Länge die maximale Länge der Zielspalte überschreitet, sind die Spalten inkompatibel.
- Wenn beim Import einer PC/IXF-Datei in eine vorhandene Datenbanktabelle eine Zeichenfolgespalte mit variabler Länge (VARCHAR, LONG VARCHAR, VARGRAPHIC oder LONG VARGRAPHIC) ausgewählt wird, deren Länge die maximale Länge der Zielspalte überschreitet, *sind* die Spalten kompatibel. Einzelne Werte werden anhand der Kompatibilitätsregeln verarbeitet, die für die INSERT-Anweisung des Datenbankmanagers gelten. PC/IXF-Werte, die zu lang für die Zieldatenbankspalte sind, sind ungültig.
- PC/IXF-Werte, die in eine *Zeichenspalte* der Datenbank mit fester Länge (d. h. eine CHAR-Spalte) importiert werden, werden nach Bedarf rechts mit Einzelbyteleerzeichen (0x20) aufgefüllt, um Werte zu erhalten, deren Länge mit der der Datenbankspalte übereinstimmt. PC/IXF-Werte, die in eine *Grafikspalte* der Datenbank mit fester Länge (d. h. eine GRAPHIC-Spalte) importiert werden, werden nach Bedarf rechts mit Doppelbyteleerzeichen (0x8140) aufgefüllt, um Werte zu erhalten, deren Länge mit der der Datenbankspalte übereinstimmt.
- Da PC/IXF-Spalten des Typs VARCHAR maximal 254 Byte lang sein dürfen, muss eine VARCHAR-Spalte der Datenbank mit der maximalen Länge  $n$  (wobei  $254 < n < 4001$  gilt) in eine PC/IXF-Spalte LONG VARCHAR der maximalen Länge  $n$  exportiert werden.
- Obwohl PC/IXF-Spalten des Typs LONG VARCHAR maximal 32 767 Byte lang sein dürfen und LONG VARCHAR-Spalten der Datenbank maximal 32 700 Byte lang sein dürfen, sind PC/IXF-Spalten des Typs LONG VARCHAR, die länger sind als 32 700 Byte (aber kürzer als 32 768 Byte), dennoch gültig und können in LONG VARCHAR-Spalten der Datenbank importiert werden. Allerdings können hierbei Daten verloren gehen.

- Da PC/IXF-Spalten des Typs VARGRAPHIC maximal 127 Byte lang sein dürfen, muss eine VARGRAPHIC-Spalte der Datenbank mit der maximalen Länge  $n$  (wobei  $127 < n < 2001$  gilt) in eine PC/IXF-Spalte LONG VARGRAPHIC der maximalen Länge  $n$  exportiert werden.
- Obwohl PC/IXF-Spalten des Typs LONG VARGRAPHIC maximal 16 383 Byte lang sein dürfen und LONG VARGRAPHIC-Spalten der Datenbank maximal 16 350 Byte lang sein dürfen, sind PC/IXF-Spalten des Typs LONG VARGRAPHIC, die länger sind als 16 350 Byte (aber kürzer als 16 384 Byte), dennoch gültig und können in LONG VARGRAPHIC-Spalten der Datenbank importiert werden. Allerdings können hierbei Daten verloren gehen.

Tabelle 21 bietet eine Übersicht des Imports von PC/IXF-Dateien in neue oder vorhandene Datenbanktabellen ohne Verwendung der Option FORCEIN.

Tabelle 21. Übersicht des Imports von PC/IXF-Dateien ohne Verwendung der Option FORCEIN

PC/IXF- SPALTEN- DATENTYP	SPALTENDATENTYP DER DATENBANK											
	NUMERIC					CHARACTER			GRAPH	DATETIME		
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) <sup>d</sup>	(SBCS, DBCS) <sup>b</sup>	<sup>b</sup>	DATE	TIME	TIME STAMP
NUMERIC (Numerisch)												
-SMALLINT	N											
	V	V	V	V <sup>a</sup>	V							
-INTEGER		N										
	V <sup>a</sup>	V	V	V <sup>a</sup>	V							
-BIGINT			N									
	V <sup>a</sup>	V <sup>a</sup>	V	V <sup>a</sup>	V							
-DECIMAL				N								
	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V							
-FLOAT					N							
	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V							
CHARACTER (Zeichen)												
-(0,0)						N						
						V				V <sup>c</sup>	V <sup>c</sup>	V <sup>c</sup>
-(SBCS,0)							N	N				
						V	V	V		V <sup>c</sup>	V <sup>c</sup>	V <sup>c</sup>
-(SBCS, DBCS)								N		V <sup>c</sup>	V <sup>c</sup>	V <sup>c</sup>
						V		V				
GRAPH (Grafik)												
									N			
						V			V			
DATETIME (Datum und Uhrzeit)												
-DATE										N		
										V		
-TIME											N	
											V	
-TIME STAMP												N
												V

Tabelle 21. Übersicht des Imports von PC/IXF-Dateien ohne Verwendung der Option FORCEIN (Forts.)

PC/IXF- SPALTEN- DATENTYP	SPALTENDATENTYP DER DATENBANK											
	NUMERIC					CHARACTER			GRAPH	DATETIME		
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) <sup>d</sup>	(SBCS, DBCS) <sup>b</sup>	<sup>b</sup>	DATE	TIME	TIME STAMP
<b>Anmerkungen:</b>												
<p>1. Die Tabelle ist eine Matrix aller gültigen PC/IXF- und Datenbankmanagerdatentypen. Wenn eine PC/IXF-Spalte in eine Datenbankspalte importiert werden kann, wird in der Matrixzelle am Schnittpunkt der Matrixzeile des PC/IXF-Datentyps und der Matrixspalte des Datenbankmanager-Datentyps ein Buchstabe angezeigt. Ein "N" gibt an, dass das Dienstprogramm eine neue Datenbanktabelle erstellt (es wird eine Datenbankspalte des angegebenen Datentyps erstellt). Ein "V" gibt an, dass das Dienstprogramm Daten in eine vorhandene Datenbanktabelle importiert (eine Datenbankspalte des angegebenen Datentyps ist ein gültiges Ziel).</p> <p>2. Zeichenfolgedatentypen werden anhand von Codepage-Attributen unterschieden. Diese Attribute werden als geordnetes Paar (SBCS,DBCS) angezeigt, wobei Folgendes gilt:</p> <ul style="list-style-type: none"> <li>• SBCS ist entweder Null oder gibt einen Wert ungleich Null eines Einzelbyte-Codepage-Attributs des Zeichendatentyps an.</li> <li>• DBCS ist entweder Null oder gibt einen Wert ungleich Null eines Doppelbyte-Codepage-Attributs des Zeichendatentyps an.</li> </ul> <p>3. Wenn die Tabelle angibt, dass eine PC/IXF-Zeichenspalte in eine Datenbankzeichenspalte importiert werden kann, entsprechen die Werte der jeweiligen Codepage-Attribute den Regeln, die für die Gleichheit von Codepages gelten.</p> <p><sup>a</sup> Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich für den numerischen Zieldatentyp liegen.</p> <p><sup>b</sup> Datentyp ist nur in DBCS-Umgebungen verfügbar.</p> <p><sup>c</sup> Einzelne Werte werden zurückgewiesen, wenn sie keine gültigen Datums- oder Uhrzeitwerte darstellen.</p> <p><sup>d</sup> Datentyp ist in DBCS-Umgebungen nicht verfügbar.</p>												

## Option FORCEIN

Die Option FORCEIN ermöglicht den Import einer PC/IXF-Datei trotz Codepage-Unterschieden zwischen Daten in der PC/IXF-Datei und der Zieldatenbank. Sie bietet zusätzliche Flexibilität bei der Definition kompatibler Spalten.

### Allgemeine FORCEIN-Semantik

Für die Verwendung der Option FORCEIN in einer SBCS- oder DBCS-Umgebung gilt die folgende allgemeine Semantik:

- Die Option FORCEIN sollte mit Bedacht verwendet werden. Es ist zumeist ratsam, zu versuchen, einen Import ohne Aktivierung dieser Option durchzuführen. Aufgrund der generischen Beschaffenheit der PC/IXF-Datenaustauscharchitektur können einige PC/IXF-Dateien jedoch möglicherweise Datentypen oder Werte enthalten, die nicht ohne Eingriff importiert werden können.
- Der Import mit FORCEIN in eine *neue* Tabelle kann unter Umständen zu einem anderen Ergebnis führen als der Import in eine vorhandene Tabelle. Eine vorhandene Tabelle enthält vordefinierte Zieldatentypen für jeden PC/IXF-Datentyp.
- Wenn LOB-Daten mit der Option LOBSINFILE exportiert und die Dateien auf einen anderen Client mit einer anderen Codepage versetzt werden, werden die CLOBS und DBCLOBS in den separaten Dateien anders als andere Daten beim Importieren oder Laden in eine Datenbank nicht in die Client-Codepage konvertiert.

## FORCEIN-Codepage-Semantik

Für die Verwendung der Option FORCEIN in einer SBCS- oder DBCS-Umgebung gilt die folgende Codepage-Semantik:

- Die Option FORCEIN inaktiviert alle Codepage-Vergleiche des Dienstprogramms IMPORT.

Diese Regel gilt für Codepage-Vergleiche auf Spalten- und auch auf Dateiebene, wenn in eine neue oder vorhandene Datenbanktabelle importiert wird. Auf der Spaltenebene (z. B. Datentyp) gilt diese Regel nur für die folgenden Datenbankmanager- und PC/IXF-Datentypen: Zeichen (CHAR, VARCHAR und LONG VARCHAR) und Grafik (GRAPHIC, VARGRAPHIC und LONG VARGRAPHIC). Die Einschränkung ergibt sich daraus, dass die Codepage-Attribute anderer Datentypen für die Auswertung der Datentypwerte nicht von Bedeutung sind.

- Die Option FORCEIN inaktiviert nicht die Prüfung der Codepage-Attribute zur Ermittlung der Datentypen.

Beispiel: Der Datenbankmanager erlaubt die Deklaration einer CHAR-Spalte mit dem Attribut FOR BIT DATA. Eine solche Deklaration setzt sowohl die SBCS-CPGID als auch die DBCS-CPGID der Spalte auf Null. Es ist der Nullwert dieser CPGIDs, der die Spaltenwerte als Bitfolgen (statt als Zeichenfolgen) angibt.

- Die Option FORCEIN impliziert keine Codepage-Umsetzung.

Werte mit Datentypen, die von der Option FORCEIN betroffen sind, werden "unverändert" kopiert. Es werden keine Codepunktzuordnungen angewendet, um eine Änderung der Codepage-Umgebungen zu berücksichtigen. Das Auffüllen des importierten Werts mit Leerzeichen kann im Falle von Zielspalten mit fester Länge erforderlich sein.

- Wenn Daten mit der Option FORCEIN in eine *vorhandene* Tabelle importiert werden, gilt Folgendes:

- Der Codepage-Wert der Zieldatenbanktabelle und -spalten ist immer vorrangig.
- Der Codepagewert der PC/IXF-Datei und -Spalten wird ignoriert.

Diese Regel gilt unabhängig davon, ob die Option FORCEIN verwendet wird. Der Datenbankmanager lässt nach der Erstellung der Datenbank keine Änderungen an einem Codepage-Wert einer Datenbank oder Spalte zu.

- Beim Importieren in eine *neue* Tabelle unter Verwendung der Option FORCEIN gilt Folgendes:

- Der Codepage-Wert der Zieldatenbank ist immer vorrangig.
- PC/IXF-Zeichenspalten mit IXFCSBCP = IXFCDBCP = 0 generieren Tabellenspalten mit dem Attribut FOR BIT DATA.
- Alle anderen PC/IXF-Zeichenspalten generieren Tabellenspalten mit SBCS- und DBCS-CPGID-Werten, die mit denen der Datenbank identisch sind.
- PC/IXF-Grafikspalten generieren Tabellengrafikspalten mit der SBCS-CPGID "undefiniert" und einer DBCS-CPGID, die mit der der Datenbank identisch ist (nur DBCS-Umgebung).

## Beispiel für FORCEIN

Angenommen, Sie haben eine PC/IXF-Spalte des Typs CHAR mit IXFCBCP = "00897" und IXFCDBC = "00301". Diese Spalte soll in eine Datenbankspalte des Typs CHAR mit SBCS-CPGID = "00850" und DBCS-CPGID = "00000" importiert werden. Ohne FORCEIN wird das Dienstprogramm beendet, und es werden keine Daten importiert, oder aber die PC/IXF-Spaltenwerte werden ignoriert und die Datenbankspalte enthält NULL-Werte (wenn die Datenbankspalte Nullwerte enthalten darf). Mit FORCEIN wird das Dienstprogramm fortgesetzt, und Codepage-Inkompatibilitäten werden ignoriert. Wenn keine weiteren Datentypinkompatibilitäten (wie beispielsweise die Länge) auftreten, werden die Werte der PC/IXF-Spalte "unverändert" importiert und stehen zur Auswertung unter der Codepage-Umgebung der Datenbank zur Verfügung.

Die folgende Tabelle zeigt:

- Die Codepage-Attribute einer Spalte, die in einer *neuen* Datenbanktabelle erstellt wird, wenn aus der PC/IXF-Datei ein Datentyp mit angegebenen Codepage-Attributen importiert wird.
- Dass das Dienstprogramm IMPORT die PC/IXF-Datentypen zurückweist, wenn sie ungültig oder inkompatibel sind.

*Tabelle 22. Übersicht über die Codepage-Semantik des Dienstprogramms IMPORT (neue Tabelle).* In dieser Tabelle wird davon ausgegangen, dass es keine Konvertierungstabelle zwischen a und x gibt. Gäbe es eine solche, würden die Elemente 3 und 4 auch ohne Verwendung der Option FORCEIN funktionieren.

CODEPAGE-ATTRIBUTE DES PC/IXF-DATENTYPS	CODEPAGE-ATTRIBUTE DER DATENBANKTABELLEN- SPALTE	
	Ohne FORCEIN	Mit FORCEIN
SBCS		
(0,0)	(0,0)	(0,0)
(a,0)	(a,0)	(a,0)
(x,0)	zurückweisen	(a,0)
(x,y)	zurückweisen	(a,0)
(a,y)	zurückweisen	(a,0)
(0,y)	zurückweisen	(0,0)
DBCS		
(0,0)	(0,0)	(0,0)
(a,0)	(a,b)	(a,b)
(x,0)	zurückweisen	(a,b)
(a,b)	(a,b)	(a,b)
(x,y)	zurückweisen	(a,b)
(a,y)	zurückweisen	(a,b)
(x,b)	zurückweisen	(a,b)
(0,b)	(-,b)	(-,b)
(0,y)	zurückweisen	(-,b)

Tabelle 22. Übersicht über die Codepage-Semantik des Dienstprogramms *IMPORT* (neue Tabelle) (Forts.). In dieser Tabelle wird davon ausgegangen, dass es keine Konvertierungstabelle zwischen a und x gibt. Gäbe es eine solche, würden die Elemente 3 und 4 auch ohne Verwendung der Option *FORCEIN* funktionieren.

CODEPAGE-ATTRIBUTE DES PC/IXF-DATENTYPS	CODEPAGE-ATTRIBUTE DER DATENBANKTABELLEN- SPALTE	
	Ohne <i>FORCEIN</i>	Mit <i>FORCEIN</i>
<b>Anmerkungen:</b>		
<ol style="list-style-type: none"> <li>Codepage-Attribute eines PC/IXF-Datentyps werden als geordnetes Paar gezeigt, wobei x für einen Einzelbyte-Codepage-Wert ungleich Null und y für einen Doppelbyte-Codepage-Wert ungleich Null steht. Ein "-" steht für einen nicht definierten Codepage-Wert.</li> <li>Die Verwendung verschiedener Buchstaben in verschiedenen Codepage-Attributpaaren ist willkürlich. Verschiedene Buchstaben implizieren verschiedene Werte. Beispiel: Wenn ein PC/IXF-Datentyp als (x,y) gezeigt wird und die Datenbankspalte als (a,y), entspricht x nicht a, die PC/IXF-Datei und die Datenbank haben jedoch den gleichen Doppelbyte-Codepage-Wert y.</li> <li>Nur Zeichen- und Grafikdatentypen sind von der <i>FORCEIN</i>-Codepage-Semantik betroffen.</li> <li>Es wird angenommen, dass die Datenbank, die die neue Tabelle enthält, die Codepage-Attribute (a,0) hat. Demzufolge müssen alle Zeichenspalten in der neuen Tabelle die Codepage-Attribute (0,0) oder (a,0) haben.  In einer DBCS-Umgebung wird angenommen, dass die Datenbank, die die neue Tabelle enthält, die Codepage-Attribute (a,b) hat. Demzufolge müssen alle Grafikspalten in der neuen Tabelle die Codepage-Attribute (-,b) und alle Zeichenspalten die Codepage-Attribute (a,b) haben. Die SBCS-CPGID wird als '-' gezeigt, weil sie für Grafikdatentypen nicht definiert ist.</li> <li>Der Datentyp des Ergebnisses wird durch die in „<i>FORCEIN</i>-Datentypsemantik“ auf Seite 326 beschriebenen Regeln bestimmt.</li> <li>Das Ergebnis zurückweisen ist eine Folge der Regeln für ungültige oder inkompatible Datentypen.</li> </ol>		

Die folgende Tabelle zeigt:

- Dass das Dienstprogramm *IMPORT* PC/IXF-Datentypen mit verschiedenen Codepage-Attributen in einer *vorhandenen* Tabellenspalte (der *Zielspalte*) mit den angegebenen Codepage-Attributen akzeptiert.
- Dass das Dienstprogramm *IMPORT* den Import eines PC/IXF-Datentyps mit bestimmten Codepage-Attributen in eine *vorhandene* Tabellenspalte mit den gezeigten Codepage-Attributen nicht zulässt. Das Dienstprogramm weist die PC/IXF-Datentypen zurück, wenn sie ungültig oder inkompatibel sind.

Tabelle 23. Übersicht über die Codepage-Semantik des Dienstprogramms *IMPORT* (vorhandene Tabelle). In dieser Tabelle wird davon ausgegangen, dass es keine Konvertierungstabelle zwischen a und x gibt.

CODEPAGE-ATTRIBUTE DES PC/IXF-DATENTYPS	CODEPAGE-ATTRIBUTE DER ZIEL-DATENBANK-SPALTE	ERGEBNISSE DES IMPORTS	
		Ohne FORCEIN	Mit FORCEIN
SBCS			
(0,0)	(0,0)	akzeptieren	akzeptieren
(a,0)	(0,0)	akzeptieren	akzeptieren
(x,0)	(0,0)	akzeptieren	akzeptieren
(x,y)	(0,0)	akzeptieren	akzeptieren
(a,y)	(0,0)	akzeptieren	akzeptieren
(0,y)	(0,0)	akzeptieren	akzeptieren
(0,0)	(a,0)	Null oder zurückweisen	akzeptieren
(a,0)	(a,0)	akzeptieren	akzeptieren
(x,0)	(a,0)	Null oder zurückweisen	akzeptieren
(x,y)	(a,0)	Null oder zurückweisen	akzeptieren
(a,y)	(a,0)	Null oder zurückweisen	akzeptieren
(0,y)	(a,0)	Null oder zurückweisen	Null oder zurückweisen
DBCS			
(0,0)	(0,0)	akzeptieren	akzeptieren
(a,0)	(0,0)	akzeptieren	akzeptieren
(x,0)	(0,0)	akzeptieren	akzeptieren
(a,b)	(0,0)	akzeptieren	akzeptieren
(x,y)	(0,0)	akzeptieren	akzeptieren
(a,y)	(0,0)	akzeptieren	akzeptieren
(x,b)	(0,0)	akzeptieren	akzeptieren
(0,b)	(0,0)	akzeptieren	akzeptieren
(0,y)	(0,0)	akzeptieren	akzeptieren
(0,0)	(a,b)	Null oder zurückweisen	akzeptieren
(a,0)	(a,b)	akzeptieren	akzeptieren
(x,0)	(a,b)	Null oder zurückweisen	akzeptieren
(a,b)	(a,b)	akzeptieren	akzeptieren
(x,y)	(a,b)	Null oder zurückweisen	akzeptieren

Tabelle 23. Übersicht über die Codepage-Semantik des Dienstprogramms *IMPORT* (vorhandene Tabelle) (Forts.). In dieser Tabelle wird davon ausgegangen, dass es keine Konvertierungstabelle zwischen a und x gibt.

CODEPAGE-ATTRIBUTE DES PC/IXF-DATENTYPS	CODEPAGE-ATTRIBUTE DER ZIEL-DATENBANK-SPALTE	ERGEBNISSE DES IMPORTS	
		Ohne FORCEIN	Mit FORCEIN
(a,y)	(a,b)	Null oder zurückweisen	akzeptieren
(x,b)	(a,b)	Null oder zurückweisen	akzeptieren
(0,b)	(a,b)	Null oder zurückweisen	Null oder zurückweisen
(0,y)	(a,b)	Null oder zurückweisen	Null oder zurückweisen
(0,0)	(-,b)	Null oder zurückweisen	akzeptieren
(a,0)	(-,b)	Null oder zurückweisen	Null oder zurückweisen
(x,0)	(-,b)	Null oder zurückweisen	Null oder zurückweisen
(a,b)	(-,b)	Null oder zurückweisen	Null oder zurückweisen
(x,y)	(-,b)	Null oder zurückweisen	Null oder zurückweisen
(a,y)	(-,b)	Null oder zurückweisen	Null oder zurückweisen
(x,b)	(-,b)	Null oder zurückweisen	Null oder zurückweisen
(0,b)	(-,b)	akzeptieren	akzeptieren
(0,y)	(-,b)	Null oder zurückweisen	akzeptieren

**Anmerkungen:**

1. Siehe Anmerkungen zu Tabelle 22 auf Seite 322.
2. Das Ergebnis Null oder zurückweisen ist eine Folge der Regeln für ungültige oder inkompatible Datentypen.

## FORCEIN-Datentypsemantik

Die Option FORCEIN ermöglicht den Import bestimmter PC/IXF-Spalten in Ziel-datenbankspalten mit ungleichen oder anderweitig inkompatiblen Datentypen. Für die Verwendung der Option FORCEIN in einer SBCS- oder DBCS-Umgebung gilt die folgende allgemeine Datentypsemantik (sofern nicht anders angegeben):

- In SBCS-Umgebungen kann mit der Option FORCEIN Folgendes importiert werden:
  - Ein PC/IXF-Datentyp BIT (IXFCSBCP = 0 = IXFCDBCP für eine PC/IXF-Zeichenspalte) in eine Datenbankzeichenspalte (SBCS-CPGID ungleich Null und DBCS-CPGID gleich Null); nur vorhandene Tabellen
  - Ein PC/IXF-Datentyp MIXED (IXFCSBCP und IXFCDBCP ungleich Null) in eine Datenbankzeichenspalte; neue und vorhandene Tabellen
  - Ein PC/IXF-Datentyp GRAPHIC in eine Datenbankspalte mit dem Attribut FOR BIT DATA (SBCS-CPGID = 0 = DBCS-CPGID); nur neue Tabellen (dies ist für vorhandene Tabellen immer zulässig)
- Mit der Option FORCEIN wird der Bereich gültiger PC/IXF-Datentypen nicht erweitert.  
PC/IXF-Spalten mit Datentypen, die nicht als gültige PC/IXF-Datentypen definiert sind, sind mit oder ohne die Option FORCEIN für den Import ungültig.
- In DBCS-Umgebungen kann mit der Option FORCEIN Folgendes importiert werden:
  - Ein PC/IXF-Datentyp BIT in eine Datenbankzeichenspalte
  - Ein PC/IXF-Datentyp BIT in eine Grafikspalte der Datenbank. Wenn die PC/IXF-Spalte des Typs BIT eine feste Länge besitzt, muss die Länge ein gerader Wert sein. Eine PC/IXF-Spalte des Typs BIT mit fester Länge und einem ungeraden Längenwert ist mit einer Grafikspalte der Datenbank nicht kompatibel. Eine PC/IXF-Spalte des Typs BIT mit variabler Länge *ist* kompatibel, unabhängig davon, ob der Längenwert gerade oder ungerade ist, obwohl ein ungerader Längenwert aus einer Spalte mit variabler Länge ein ungültiger Wert für den Import in eine Grafikspalte der Datenbank ist.
  - Ein PC/IXF-Datentyp MIXED in eine Datenbankzeichenspalte

Tabelle 24 bietet eine Übersicht des Imports von PC/IXF-Dateien in neue oder vorhandene Datenbanktabellen unter Verwendung der Option FORCEIN.

Tabelle 24. Übersicht des Imports von PC/IXF-Dateien mit Verwendung der Option FORCEIN

PC/IXF- SPALTEN- DATENTYP	SPALTENDATENTYP DER DATENBANK											
	NUMERIC					CHARACTER			GRAPH	DATETIME		
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) <sup>e</sup>	(SBCS, DBCS) <sup>b</sup>	<sup>b</sup>	DATE	TIME	TIME STAMP
NUMERIC (Numerisch)												
-SMALLINT	N											
	V	V	V	V <sup>a</sup>	V							
-INTEGER		N										
	V <sup>a</sup>	V	V	V <sup>a</sup>	V							
-BIGINT			N									
	V <sup>a</sup>	V <sup>a</sup>	V	V <sup>a</sup>	V							
-DECIMAL				N								
	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V							
-FLOAT					N							

Tabelle 24. Übersicht des Imports von PC/IXF-Dateien mit Verwendung der Option FORCEIN (Forts.)

PC/IXF- SPALTEN- DATENTYP	SPALTENDATENTYP DER DATENBANK											
	NUMERIC					CHARACTER			GRAPH	DATETIME		
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) <sup>e</sup>	(SBCS, DBCS) <sup>b</sup>	<sup>b</sup>	DATE	TIME	TIME STAMP
	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V							
CHARACTER (Zeichen)												
-(0,0)						N						
						V	V mit F	V mit F	V mit F	V <sup>c</sup>	V <sup>c</sup>	V <sup>c</sup>
-(SBCS,0)							N	N				
						V	V	V		V <sup>c</sup>	V <sup>c</sup>	V <sup>c</sup>
-(SBCS, DBCS)							N mit F <sup>d</sup>	N		V <sup>c</sup>	V <sup>c</sup>	V <sup>c</sup>
						V	V mit F	V				
GRAPH (Grafik)												
						N mit F <sup>d</sup>			N			
						V			V			
DATETIME (Datum und Uhrzeit)												
-DATE										N		
										V		
-TIME											N	
											V	
-TIME STAMP												N
												V
<p><b>Anmerkung:</b> Wenn eine a PC/IXF-Spalte nur mit der Option FORCEIN in eine Datenbankspalte importiert werden kann, wird die Zeichenfolge "mit F" zusammen mit "N" oder "V" gezeigt. Ein "N" gibt an, dass das Dienstprogramm eine neue Datenbanktabelle erstellt. Ein "V" gibt an, dass das Dienstprogramm Daten in eine vorhandene Datenbanktabelle importiert. Die Option FORCEIN wirkt sich nur auf die Kompatibilität von Zeichen- und Grafikdatentypen aus.</p> <p><sup>a</sup> Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich für den numerischen Zieldatentyp liegen.</p> <p><sup>b</sup> Datentyp ist nur in DBCS-Umgebungen verfügbar.</p> <p><sup>c</sup> Einzelne Werte werden zurückgewiesen, wenn sie keine gültigen Datums- oder Uhrzeitwerte darstellen.</p> <p><sup>d</sup> Gilt nur, wenn der PC/IXF-Quellendatentyp nicht von der Zieldatenbank unterstützt wird.</p> <p><sup>e</sup> Datentyp ist in DBCS-Umgebungen nicht verfügbar.</p>												

**Zugehörige Referenzen:**

- „PC/IXF-Datentypen“ auf Seite 304
- „Allgemeine Regeln für den Import von PC/IXF-Dateien in Datenbanken“ auf Seite 315

## Unterschiede zwischen PC/IXF und Version 0 System/370 IXF

Im Folgenden werden die Unterschiede zwischen PC/IXF, das vom Datenbankmanager verwendet wird, und Version 0 System/370 IXF, das von mehreren Host-Datenbankprodukten verwendet wird, beschrieben:

- PC/IXF-Dateien orientieren sich an ASCII und nicht an EBCDIC. PC/IXF-Dateien haben eine erheblich erweiterte Codepage-Kennzeichnung, einschließlich neuer Codepage-Kennungen im H-Datensatz und der Verwendung von Codepage-Werten in den Spaltendeskriptorsätzen. Es gibt auch einen Mechanismus zur Markierung von Spalten mit Zeichendaten als FOR BIT DATA. FOR BIT DATA-Spalten sind von besonderer Bedeutung, weil Umsetzungen, die ein PC/IXF-Dateiformat in ein anderes IXF- oder Datenbankdateiformat oder umgekehrt konvertieren, keine Codepage-Umsetzung der Werte durchführen können, die in FOR BIT DATA-Spalten enthalten sind.
- Es ist nur das Maschinendatenformat zulässig, d. h. das Feld IXFTFORM muss immer den Wert M enthalten. Weiterhin müssen die Maschinendaten in PC-Formaten vorliegen. Das bedeutet, dass das Feld IXFTMFRM den Wert PC enthalten muss. Das bedeutet, dass ganze Zahlen, Gleitkommazahlen und Dezimalzahlen in Datenteilen von PC/IXF-Datensätzen PC-Formate aufweisen müssen.
- Anwendungsdatensätze (A) sind nach dem H-Datensatz in einer PC/IXF-Datei zulässig. Sie werden nicht gezählt, wenn der Wert des Feldes IXFHCNT berechnet wird.
- Jeder PC/IXF-Datensatz beginnt mit einem Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des PC/IXF-Datensatzes in Byte enthält (ohne Einschluss des Satzlängenanzeigers), d. h. die gesamte Satzlänge minus 6 Byte. Der Zweck des Satzlängenfeldes besteht darin, PC-Programmen die Erkennung von Satzbegrenzungen zu ermöglichen.
- PC/IXF unterstützt keine Version 0 IXF X-Sätze (jedoch D-Satzkennungen), um die kompakte Speicherung von Daten mit variabler Länge zu erleichtern und eine komplexe Verarbeitung zu vermeiden, wenn ein Feld in mehrere Sätze aufgeteilt ist. Wenn ein Feld mit variabler Länge oder ein Feld, das Nullwerte enthalten darf, das letzte Feld in einem D-Datensatz ist, ist es nicht nötig, die gesamte maximale Länge des Feldes in die PC/IXF-Datei zu schreiben.

---

## WSF-Dateiformat (Arbeitsblattformat)

Die Produkte Lotus 1-2-3 und Symphony verwenden dasselbe Basisformat, wobei in jedem neuen Release zusätzliche Funktionen hinzugefügt werden. Der Datenbankmanager unterstützt eine Untergruppe der Arbeitsblatt Datensätze, die für alle Lotus-Produkte identisch sind. Dies bedeutet, dass für die vom Datenbankmanager unterstützten Releases von Lotus 1-2-3 und Symphony alle Dateinamen mit einer beliebigen Erweiterung aus drei Zeichen akzeptiert werden. Zum Beispiel: WKS, WK1, WRK, WR1, WJ2.

Jede WSF-Datei repräsentiert ein Arbeitsblatt. Der Datenbankmanager verwendet folgende Konventionen, um die Arbeitsblätter zu interpretieren und um die Konsistenz der im Rahmen der Exportoperationen generierten Arbeitsblätter zu gewährleisten:

- Zellen in der ersten Zeile (ROW-Wert 0) werden für die beschreibenden Informationen zum gesamten Arbeitsblatt reserviert. Alle Daten in dieser Zeile sind wahlfrei. Sie wird beim Import ignoriert.
- Zellen in der zweiten Zeile (ROW-Wert 1) werden für die Spaltenkennsätze verwendet.
- Die verbleibenden Zeilen sind Datenzeilen (Datensätze oder Zeilen mit Daten der Tabelle).
- Zellenwerte unter Spaltenüberschriften sind Werte für diese Spalte oder für dieses Feld.
- Ein NULL-Wert wird durch das Fehlen eines realen Datensatzes für den Zelleninhalt (z. B. ganze Zahl, Zahl, Kennsatz oder Formeldatensatz) in einer bestimmten Spalte innerhalb einer Zeile mit Datensätzen für den Zelleninhalt dargestellt.

**Anmerkung:** Eine Zeile, die nur NULL-Werte enthält, wird nicht exportiert oder importiert.

Beim Erstellen einer Datei im WSF-Format während einer Exportoperation können einige Daten verloren gehen.

WSF-Dateien verwenden eine Lotus-Codepunktzuordnung, die nicht notwendigerweise mit den Codepages übereinstimmt, die von DB2 unterstützt werden. Infolgedessen werden beim Importieren einer WSF-Datei Daten aus den Lotus-Codepunkten in die Codepunkte konvertiert, die von der Codepage der Anwendung unterstützt werden. Beim Exportieren erfolgt die Konvertierung in umgekehrter Richtung. DB2 unterstützt die Konvertierung zwischen Lotus-Codepunkten und Codepunkten, die durch die Codepages 437, 819, 850, 860, 863 und 865 definiert sind.

**Anmerkung:** Für Benutzer von Mehrbytezeichensätzen wird keine Konvertierung ausgeführt.



---

## Anhang E. Hinweise zu Unicode bei Verwendung der Dienstprogramme EXPORT/IMPORT/LOAD

Die Dienstprogramme EXPORT, IMPORT und LOAD werden nicht unterstützt, wenn sie zusammen mit einem Unicode-Client eingesetzt werden, der mit einer Datenbank verbunden ist, die nicht das Unicode-Format verwendet. Dateien von Unicode-Clients werden nur dann unterstützt, wenn der Unicode-Client mit einer Unicode-Datenbank verbunden ist.

Die DEL-, ASC- und PC/IXF-Dateiformate werden - wie in diesem Abschnitt noch erläutert wird - bei einer UCS-2-Datenbank unterstützt. Das WSF-Format wird nicht unterstützt.

Beim Exportieren von Daten aus einer UCS-2-Datenbank in eine ASCII-Datei ohne universelle Zeilenbegrenzer (DEL) werden alle Zeichendaten in die Codepage der Anwendung konvertiert. Sowohl Zeichenfolgedaten als auch Grafikzeichenfolgedaten werden in dieselbe SBCS- oder MBCS-Codepage des Clients konvertiert. Dies ist das erwartete Verhalten beim Export von Datenbanken. Es kann nicht geändert werden, weil die gesamte ASCII-Datei ohne universelle Zeilenbegrenzer nur eine Codepage haben kann. Daher werden beim Export in eine solche ASCII-Datei nur diejenigen UCS-2-Zeichen gesichert, die in der Codepage der Anwendung enthalten sind. Andere Zeichen werden durch das Standardsubstitutionszeichen für die Codepage der Anwendung ersetzt. Bei UTF-8-Clients (Codepage 1208) gehen keine Daten verloren, weil alle UCS-2-Zeichen durch UTF-8-Clients unterstützt werden.

Beim Importieren von Daten aus einer ASCII-Datei (mit dem DEL- oder ASC-Format) in eine UCS-2-Datenbank werden Zeichenfolgedaten aus der Codepage der Anwendung in UTF-8 und Grafikzeichenfolgedaten aus der Codepage der Anwendung in UCS-2 konvertiert. Es gehen keine Daten verloren. Wenn Sie ASCII-Daten importieren wollen, die unter einer anderen Codepage gesichert wurden, sollten Sie die Codepage der Datendatei ändern, bevor Sie den Befehl IMPORT absetzen. Dies erreichen Sie beispielsweise, indem Sie für DB2CODEPAGE die Codepage der ASCII-Datendatei angeben.

Der Bereich der gültigen ASCII-Begrenzer für SBCS- und MBCS-Clients entspricht dem Bereich, der gegenwärtig durch DB2<sup>®</sup> UDB für diese Clients unterstützt wird. Der Bereich der gültigen Begrenzer für UTF-8-Clients erstreckt sich von X'01' bis X'7F', wobei die üblichen Einschränkungen gelten.

Beim Exportieren von Daten aus einer UCS-2-Datenbank in eine PC/IXF-Datei werden Zeichenfolgedaten in die SBCS/MBCS-Codepage des Clients konvertiert. Grafikzeichenfolgedaten werden nicht konvertiert, sondern in UCS-2 (Codepage 1200) gespeichert. Es gehen keine Daten verloren.

Beim Importieren von Daten aus einer PC/IXF-Datei in eine UCS-2-Datenbank müssen Zeichenfolgedaten in der in den PC/IXF-Kopfdaten gespeicherten SBCS/MBCS-Codepage vorliegen und Grafikzeichenfolgedaten in der DBCS-Codepage, die in den PC/IXF-Kopfdaten gespeichert ist. Zeichenfolgedaten werden durch das Dienstprogramm IMPORT aus der in den PC/IXF-Kopfdaten angegebenen Codepage in die Codepage des Clients konvertiert. Anschließend werden sie durch die Anweisung INSERT von der Client-Codepage in UTF-8 konvertiert.

Grafikzeichenfolgedaten werden durch das Dienstprogramm IMPORT aus der DBCS-Codepage, die in den PC/IXF-Kopfdaten angegeben ist, direkt in UCS-2 (Codepage 1200) konvertiert.

Das Dienstprogramm LOAD stellt die Daten direkt in die Datenbank und geht standardmäßig davon aus, dass Daten in ASC- oder DEL-Dateien in der Codepage der Datenbank vorliegen. Daher findet für ASCII-Dateien standardmäßig keine Codepagekonvertierung statt. Wenn die Codepage für die Datendatei explizit angegeben wurde (mit dem Änderungswert `codepage`), verwendet das Dienstprogramm LOAD diese Informationen, um vor dem Laden der Daten eine Konvertierung aus der angegebenen Codepage in die Datenbankcodepage vorzunehmen. Bei PC/IXF-Dateien konvertiert das Dienstprogramm LOAD die in den IXF-Kopfdaten angegebenen Codepages immer in die Datenbankcodepage (Codepage 1208 bei CHAR und Codepage 1200 bei GRAPHIC).

Die Codepage für DBCLOB-Dateien ist bei UCS-2 immer 1200. Für CLOB-Dateien wird die gleiche Codepage wie für die importierten, geladenen oder exportierten Daten verwendet. Wenn beispielsweise Daten mit dem PC/IXF-Format geladen oder importiert werden, wird davon ausgegangen, dass die CLOB-Datei die in den PC/IXF-Kopfdaten angegebene Codepage verwendet. Falls die DBCLOB-Datei im ASC- oder DEL-Format vorliegt, geht das Dienstprogramm LOAD davon aus, dass die CLOB-Daten die Codepage der Datenbank verwenden (sofern nicht explizit eine andere Codepage durch den Änderungswert `codepage` angegeben ist). Das Dienstprogramm IMPORT geht hingegen davon aus, dass die Codepage der Clientanwendung verwendet wird.

Der Änderungswert `nochecklengths` wird aus den folgenden Gründen immer bei einer UCS-2-Datenbank angegeben:

- SBCS kann mit einer Datenbank verbunden sein, für die keine DBCS-Codepage vorhanden ist.
- Zeichenfolgen im Format UTF-8 haben in der Regel eine andere Länge als Zeichenfolgen in Client-Codepages.

---

## Einschränkungen für Codepages 1394, 1392 und 5488

Mit den Dienstprogrammen IMPORT, EXPORT und LOAD können jetzt Daten aus der neuen Codepage GB 18030 für Chinesisch (Codepage-IDs 1392 und 5488) und der neuen Codepage ShiftJISX 0213 für Japanisch (Codepage-ID 1394) an Unicode-Datenbanken von DB2 UDB übertragen werden. Zusätzlich können mit dem Dienstprogramm EXPORT Daten aus Unicode-Datenbanken von DB2 UDB an Daten übertragen werden, die die Codepage GB 18030 oder ShiftJIS X0213 verwenden.

Der folgende Befehl lädt beispielsweise die Datendatei `u/jp/user/x0213/data.del` mit dem Format `Shift_JISX0213`, die sich auf einem fernen verbundenen Client befindet, in die Tabelle MYTABLE:

```
db2 load client from /u/jp/user/x0213/data.del
of del modified by codepage=1394 insert into mytable
```

Hierbei befindet sich die Tabelle MYTABLE in einer Unicode-Datenbank von DB2 UDB.

Da nur Verbindungen zwischen einem Unicode-Client und einem Unicode-Server unterstützt werden, müssen Sie entweder einen Unicode-Client verwenden oder die DB2-Registrierdatenbank-Variablen DB2CODEPAGE auf den Wert 1208 setzen, bevor Sie das Dienstprogramm LOAD, IMPORT oder EXPORT verwenden.

Eine Konvertierung aus der Codepage 1394, 1392 oder 5488 in Unicode kann eine Vergrößerung des Datenumfangs nach sich ziehen. Beispiel: Ein 2-Byte-Zeichen kann in Spalten mit dem Datentyp GRAPHIC in Form von zwei 16-Bit-Unicode-Zeichen gespeichert werden. Sie müssen gewährleisten, dass die Zielspalten in der Unicode-Datenbank groß genug sind, um alle vergrößerten Unicode-Bytes aufnehmen zu können.

---

## Inkompatibilität

Bei Anwendungen, die mit einer UCS-2-Datenbank verbunden sind, liegen Grafikzeichenfolgedaten immer im Format UCS-2 (Codepage 1200) vor. Bei Anwendungen, die mit Datenbanken verbunden sind, welche ein anderes Format als UCS-2 verwenden, liegen die Grafikzeichenfolgedaten entweder in der DBCS-Codepage der Anwendung vor oder sind nicht zulässig, falls die Anwendungscodepage SBCS ist. Beispiel: Wenn ein 932-Client mit einer Datenbank verbunden ist, die ein anderes Format als UCS-2 für Japanisch verwendet, liegen die Grafikzeichenfolgedaten in der Codepage 301 vor. Bei Anwendungen des 932-Clients, die mit einer UCS-2-Datenbank verbunden sind, liegen die Grafikzeichenfolgedaten in UCS-2 vor.

### Zugehörige Referenzen:

- „Beschreibungen von DEL-Datentypen“ auf Seite 279
- „ASC-Dateiformat (ASCII-Format mit universellen Zeilenbegrenzern)“ auf Seite 281
- „PC-Version des IXF-Dateiformats“ auf Seite 285



## Anhang F. Durch Dienstprogramme EXPORT, IMPORT und LOAD verwendete Bindedateien

Die folgende Tabelle listet Bindedateien mit ihren jeweiligen Standardisolationsstufen auf. Außerdem ist angegeben, welche Dienstprogramme diese Dateien einsetzen und zu welchem Zweck sie verwendet werden.

Bindedatei (Standardisolationsstufe)	Dienstprogramm/Zweck
db2ueiwi.bnd (CS)	IMPORT/EXPORT. Abfragen von Informationen zu Tabellenspalten und Indizes.
db2uexpm.bnd (CS)	EXPORT. Abrufen von Daten aus der für die Exportoperation angegebenen SQL-Abfrage.
db2uimpm.bnd (RS)	IMPORT. Einfügen von Daten aus der Quellendatendatei in die Zieltabelle.
db2uipkg.bnd (CS)	IMPORT. Prüfen von Bindeoptionen.
db2uiici.bnd (RR)	IMPORT. Erstellen von Indizes bei angegebener Option IXF CREATE.
db2ucktb.bnd (CS)	LOAD. Ausführen allgemeiner Initialisierungsprozesse für eine Ladeoperation.
db2ulxld.bnd (CS)	LOAD. Verarbeiten der SQL-Abfrage, die während einer LOAD FROM CURSOR-Operation bereitgestellt wird.
db2uigsi.bnd (RS auf UNIX-basierten Systemen, RR auf allen anderen Plattformen)	IMPORT/EXPORT. Löschen von Indizes und Prüfen von referenziellen Integritätsbedingungen für eine IMPORT REPLACE-Operation. Außerdem Abrufen von Informationen zu Identitätsspalten beim Export von IXF-Dateien.
db2uiict.bnd (RR)	IMPORT. Erstellen von Tabellen bei angegebener Option IXF CREATE.
db2uqtpd.bnd (RR)	IMPORT/EXPORT. Ausführen der Verarbeitung für hierarchische Tabellen.
db2uqtnm.bnd (RR)	IMPORT. Ausführen der Verarbeitung für hierarchische Tabellen bei angegebener Option IXF CREATE.
db2uimtb.bnd (RS)	IMPORT. Ausführen allgemeiner Initialisierungsprozesse für eine Importoperation.



---

## Anhang G. Warnungen, Fehler- und Beendigungsnachrichten

Die von den verschiedenen Dienstprogrammen generierten Nachrichten sind in den SQL-Nachrichten enthalten. Diese Nachrichten werden vom Datenbankmanager generiert, wenn eine Warnungs- oder Fehlerbedingung erkannt wurde. Jede Nachricht besitzt eine Nachrichten-ID, die aus einem Präfix (SQL) und einer vier- oder fünfstelligen Nachrichtennummer zusammengesetzt ist. Es gibt Informationsnachrichten, Warnungen und Nachrichten über kritische Systemfehler. Nachrichten-IDs, die mit N enden, sind Fehlernachrichten. IDs, die mit W enden, sind Warnungen oder Informationsnachrichten. Nachrichten-IDs, die mit C enden, weisen auf kritische Systemfehler hin.

Die Nachrichtennummer wird auch als *SQLCODE-Wert* bezeichnet. Der *SQLCODE*-Wert wird als positive oder negative Zahl an die Anwendung weitergegeben, je nach Nachrichtentyp (N, W oder C). N und C ergeben negative Werte, W ergibt positive Werte. DB2 gibt den *SQLCODE*-Wert an die Anwendung zurück, und die Anwendung kann die dem *SQLCODE*-Wert zugeordnete Nachricht abrufen. DB2 gibt auch einen *SQLSTATE*-Wert bei Bedingungen zurück, die das Ergebnis einer SQL-Anweisung sein könnten. Einigen *SQLCODE*-Werten sind *SQLSTATE*-Werte zugeordnet.

Sie können die Informationen in diesen Buch verwenden, um einen Fehler oder ein Problem einzugrenzen und das Problem zu beheben, indem Sie die entsprechende Wiederherstellungsmaßnahme ausführen. Diese Informationen können auch verwendet werden, um sich darüber zu informieren, wo Nachrichten generiert und protokolliert werden.

SQL-Nachrichten und der Nachrichtentext, die den *SQLSTATE*-Werten zugeordnet sind, können auch über die Befehlszeile des Betriebssystems aufgerufen werden. Geben Sie Folgendes an der Eingabeaufforderung des Betriebssystems ein, um die Hilfsfunktion für diese Fehlernachrichten aufzurufen:

```
db2 ? SQLnnnnn
```

Dabei steht *nnnnn* für die Nachrichtennummer. Auf Systemen mit UNIX-Basis empfiehlt es sich, Begrenzer in Form von doppelten Anführungszeichen zu verwenden. Hierdurch werden Probleme verhindert, die auftreten können, wenn das Verzeichnis Dateinamen enthält, die aus einem Einzelbuchstaben bestehen:

```
db2 "? SQLnnnnn"
```

Die Nachrichten-ID, die als Parameter des Befehls **db2** akzeptiert wird, kann in beliebiger Schreibweise angegeben werden, und das letzte Zeichen ist nicht erforderlich. Demzufolge führen die folgenden Befehle alle zum gleichen Ergebnis:

```
db2 ? SQL0000N
db2 ? sql0000
db2 ? SQL0000n
```

Wenn der Nachrichtentext zu lang für den Bildschirm ist, können Sie den folgenden Befehl verwenden (auf UNIX-basierten Basisbetriebssystemen und anderen, die den Pipe-Befehl "more" unterstützen):

```
db2 ? SQLnnnnn | more
```

Sie können die Ausgabe auch in eine Datei umleiten, die dann angezeigt werden kann.

Die Hilfe kann auch im interaktiven Eingabemodus aufgerufen werden. Geben Sie Folgendes an der Eingabeaufforderung des Betriebssystems ein, um diesen Modus zu aktivieren:

```
db2
```

Geben Sie Folgendes an der Eingabeaufforderung db2 => ein, um die Hilfe zu DB2-Nachrichten im interaktiven Eingabemodus aufzurufen:

```
? SQLnnnnn
```

Der Nachrichtentext, der SQLSTATE-Werten zugeordnet ist, kann durch Absetzen folgender Befehle abgerufen werden:

```
db2 ? nnnnn  
oder  
db2 ? nn
```

Dabei ist *nnnnn* ein fünfstelliger SQLSTATE-Wert (alphanumerisch) und *nn* ein zweistelliger SQLSTATE-Klassencode (die ersten beiden Stellen des SQLSTATE-Werts).

---

## Anhang H. Technische Informationen zu DB2 Universal Database

---

### DB2-Dokumentation und Hilfe

Die technischen Informationen zu DB2<sup>®</sup> stehen über die folgenden Tools und Methoden zur Verfügung:

- DB2 Information - Unterstützung
  - Themen
  - Hilfe für DB2-Tools
  - Beispielprogramme
  - Lernprogramme
- Für den Download verfügbare PDF-Dateien, PDF-Dateien auf CD und gedruckte Bücher
  - Handbücher
  - Referenzhandbücher
- Befehlszeilenhilfe
  - Hilfe für Befehle
  - Hilfe für Nachrichten
  - Hilfe für SQL-Anweisungen
- Installierter Quellcode
  - Beispielprogramme

Darüber hinaus können Sie auf zusätzliche technische Informationen zu DB2 Universal Database<sup>™</sup>, wie beispielsweise technische Hinweise (Technotes), White Papers und Redbooks<sup>™</sup>, online über [ibm.com](http://ibm.com)<sup>®</sup> zugreifen. Rufen Sie die Website 'DB2 Information Management - Library' unter [www.ibm.com/software/data/pubs/](http://www.ibm.com/software/data/pubs/) auf.

### Aktualisierungen der DB2-Dokumentation

In bestimmten Fällen stellt IBM<sup>®</sup> in regelmäßigen Abständen Dokumentations-Fix-Paks und andere Dokumentationsaktualisierungen für 'DB2 Information - Unterstützung' zur Verfügung. Wenn Sie über <http://publib.boulder.ibm.com/infocenter/db2help/> auf 'DB2 Information - Unterstützung' zugreifen, erhalten Sie stets die neuesten Informationen. Falls Sie 'DB2 Information - Unterstützung' lokal installiert haben, müssen Sie alle Aktualisierungen manuell installieren, bevor Sie sie anzeigen können. Diese Dokumentationsaktualisierungen ermöglichen Ihnen, die Informationen, die Sie von der CD mit *DB2 Information - Unterstützung* installiert haben, auf den neuesten Stand zu bringen, sobald neue Informationen verfügbar sind.

'DB2 Information - Unterstützung' wird häufiger aktualisiert als die PDF- und Hardcopy-Bücher. Um stets die jeweils neuesten technischen Informationen zu DB2 zur Verfügung zu haben, sollten Sie die Dokumentationsaktualisierungen installieren, sobald sie verfügbar sind, oder 'DB2 Information - Unterstützung' über die Website [www.ibm.com](http://www.ibm.com) aufrufen.

### Zugehörige Tasks:

- „Aufrufen der Kontexthilfe über ein DB2-Tool“ auf Seite 359

### Zugehörige Referenzen:

- „DB2-Dokumentation in PDF-Format und gedrucktem Format“ auf Seite 352

---

## DB2 Information - Unterstützung

Die DB2<sup>®</sup>-Komponente 'DB2 Information - Unterstützung' bietet Ihnen die Möglichkeit, auf alle Informationen zuzugreifen, die Sie zur optimalen Nutzung der Produkte innerhalb der DB2-Produktfamilie, wie z. B. DB2 Universal Database<sup>™</sup>, DB2 Connect<sup>™</sup>, DB2 Information Integrator und DB2 Query Patroller<sup>™</sup>, benötigen. 'DB2 Information - Unterstützung' dokumentiert auch die wichtigsten DB2-Funktionen und -Komponenten, einschließlich der Funktionen für die Replikation, das Data Warehousing und die DB2 Extender.

Wenn Sie für die Anzeige von 'DB2 Information - Unterstützung' Mozilla ab Version 1.0 oder Microsoft<sup>®</sup> Internet Explorer ab Version 5.5 verwenden, stehen Ihnen die folgenden Funktionen zur Verfügung. Für bestimmte Funktionen muss die JavaScript<sup>™</sup>-Unterstützung aktiviert werden :

### Flexible Installationsoptionen

Wählen Sie für die Anzeige der DB2-Dokumentation die Option, die Ihren Anforderungen am besten entspricht:

- Stellen Sie ohne großen Aufwand sicher, dass Ihre Dokumentation stets auf dem neuesten Stand ist, indem Sie auf die gesamte Dokumentation direkt über 'DB2 Information - Unterstützung' auf der IBM<sup>®</sup> Website unter <http://publib.boulder.ibm.com/infocenter/db2help/> zugreifen.
- Reduzieren Sie den Aktualisierungsaufwand auf ein Minimum und begrenzen Sie den Datenaustausch auf Ihr Intranet, indem Sie die DB2-Dokumentation auf einem einzigen Server innerhalb Ihres Intranets installieren.
- Erzielen Sie maximale Flexibilität und reduzieren Sie die Abhängigkeit von Netzwerkverbindungen, indem Sie die DB2-Dokumentation auf dem eigenen Computer installieren.

### Suchen

Sie können alle Themen in 'DB2 Information - Unterstützung' durchsuchen, indem Sie einen Suchbegriff im Textfeld **Suchen** eingeben. Schließen Sie Begriffe in Anführungszeichen ein, wenn Sie nach exakten Übereinstimmungen suchen möchten. Mit Hilfe von Platzhalterzeichen (\*, ?) und Booleschen Operatoren (AND, NOT, OR) können Sie die Suche eingrenzen.

### Aufgabenorientiertes Inhaltsverzeichnis

Die Themen in der DB2-Dokumentation können über ein zentrales Inhaltsverzeichnis lokalisiert werden. Das Inhaltsverzeichnis ist primär auf der Basis übergeordneter Aufgabenbereiche aufgebaut, enthält jedoch auch Einträge für Produktübersichten, Ziele, Referenzinformationen sowie einen Index und ein Glossar.

- Produktübersichten beschreiben die Beziehung zwischen den in der DB2-Produktfamilie verfügbaren Produkten sowie die von den einzelnen Produkten bereitgestellten Funktionen und enthalten darüber hinaus die neuesten Release-Informationen für diese Produkte.
- Aufgabenkategorien, wie z. B. Installation, Verwaltung und Entwicklung, umfassen Themen, mit deren Hilfe Sie die einzelnen Aufgaben schnell

ausführen und sich außerdem genauere Kenntnisse über die Hintergrundinformationen zu diesen Aufgaben verschaffen können.

- In den Referenzthemen finden Sie detaillierte Informationen zu einem Thema, einschließlich der Anweisungs- und Befehlssyntax, der Hilfetexte zu Nachrichten und der Konfigurationsparameter.

#### **Anzeigen des aktuellen Themas im Inhaltsverzeichnis**

Wenn Sie sehen möchten, welchem Bereich des Inhaltsverzeichnisses das aktuelle Thema zugeordnet ist, klicken Sie den Knopf **Aktualisieren / aktuelles Thema anzeigen** im Teilfenster des Inhaltsverzeichnisses oder den Knopf **Im Inhaltsverzeichnis anzeigen** im Inhaltsteilfenster an. Diese Funktion ist zum Beispiel dann von Nutzen, wenn Sie mehreren Links zu zugehörigen Themen in verschiedenen Dateien gefolgt sind oder ein Thema über das Ergebnis einer Suche aufgerufen haben.

**Index** Über den Index können Sie auf die gesamte Dokumentation zugreifen. Der Index ist alphabetisch nach Indexeinträgen sortiert.

#### **Glossar**

Im Glossar finden Sie Definitionen zu Termini, die in der DB2-Dokumentation verwendet werden. Das Glossar ist alphabetisch nach Glossareinträgen sortiert.

#### **Integrierte übersetzte Informationen**

Die Informationen in 'DB2 Information - Unterstützung' werden in der Sprache angezeigt, die Sie in den Benutzervorgaben des verwendeten Browsers festgelegt haben. Ist ein Thema nicht in der bevorzugten Sprache verfügbar, wird die englische Version des Themas angezeigt.

Technische Informationen zu iSeries™ finden Sie im Informationszentrum von IBM eServer™ iSeries unter [www.ibm.com/eserver/series/infocenter/](http://www.ibm.com/eserver/series/infocenter/).

#### **Zugehörige Konzepte:**

- „DB2 Information - Unterstützung: Installationsszenarios“ auf Seite 341

#### **Zugehörige Tasks:**

- „Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung'“ auf Seite 350
- „Anzeigen von Themen in der gewünschten Sprache in 'DB2 Information - Unterstützung'“ auf Seite 351
- „Aufrufen von 'DB2 Information - Unterstützung'“ auf Seite 349
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (UNIX)“ auf Seite 344
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (Windows)“ auf Seite 347

---

## **DB2 Information - Unterstützung: Installationsszenarios**

Je nach Arbeitsumgebung kann es unterschiedliche Anforderungen hinsichtlich des Zugriffs auf DB2®-Informationen geben. Sie können auf 'DB2 Information - Unterstützung' entweder auf der IBM® Website zugreifen oder auf einem Server im unternehmensinternen Netzwerk oder auf eine auf dem lokalen Computer installierte Version. In allen drei Fällen befindet sich die Dokumentation in 'DB2 Information - Unterstützung', einem strukturierten System themenbasierter Informationen, die über einen Browser angezeigt werden können. Standardmäßig greifen DB2-Produkte auf 'DB2 Information - Unterstützung' auf der IBM Website zu.

Wenn Sie jedoch auf 'DB2 Information - Unterstützung' auf einem Intranet-Server oder auf dem eigenen Computer zugreifen möchten, müssen Sie 'DB2 Information - Unterstützung' mit Hilfe der entsprechenden CD installieren, die sich im Programmpaket des Produkts befindet. Anhand der nachfolgenden Übersicht über die verfügbaren Optionen für den Zugriff auf die DB2-Dokumentation und mit Hilfe der drei Installationsszenarios können Sie ermitteln, welche Methode für den Zugriff auf 'DB2 Information - Unterstützung' für Ihre Anforderungen und Arbeitsumgebung am besten geeignet ist und welche Aspekte Sie bei der Installation berücksichtigen müssen.

### Übersicht über die verfügbaren Optionen für den Zugriff auf die DB2-Dokumentation:

Die folgende Tabelle enthält Empfehlungen hinsichtlich der für Ihre Arbeitsumgebung geeigneten Optionen für den Zugriff auf die DB2-Produktdokumentation in 'DB2 Information - Unterstützung'.

Internetzugriff	Intranetzugriff	Empfehlung
Ja	Ja	Greifen Sie entweder über die IBM Website auf 'DB2 Information - Unterstützung' zu oder auf die auf einem Intranet-Server installierte Version von 'DB2 Information - Unterstützung'.
Ja	Nein	Greifen Sie über die IBM Website auf 'DB2 Information - Unterstützung' zu.
Nein	Ja	Greifen Sie auf die auf einem Intranet-Server installierte Version von 'DB2 Information - Unterstützung' zu.
Nein	Nein	Greifen Sie auf die auf einem lokalen Computer installierte Version von 'DB2 Information - Unterstützung' zu.

### Szenario: Zugriff auf 'DB2 Information - Unterstützung' auf Ihrem Computer:

Tsu-Chen besitzt eine Fabrik in einer Kleinstadt, in der es vor Ort keinen Anbieter für einen Internetzugang gibt. Für die Verwaltung des Lagerbestands, der Produktbestellungen, der Betriebsausgaben und seines Bankkontos hat Tsu-Chen DB2 Universal Database™ gekauft. Da er zuvor noch nie ein DB2-Produkt verwendet hat, muss er anhand der DB2-Produktdokumentation lernen, wie die Verwaltung funktioniert.

Nachdem er DB2 Universal Database mit der Option für die Standardinstallation auf seinem Computer installiert hat, versucht Tsu-Chen, auf die DB2-Dokumentation zuzugreifen. Sein Browser zeigt jedoch eine Fehlermeldung mit der Information an, dass die Seite, die geöffnet werden sollte, nicht gefunden werden kann. Tsu-Chen überprüft das Installationshandbuch für sein DB2-Produkt und findet heraus, dass er 'DB2 Information - Unterstützung' zunächst installieren muss, um auf seinem Computer auf die DB2-Dokumentation zugreifen zu können. Im Programmpaket findet er die *CD für DB2 Information - Unterstützung* und installiert sie.

Über das Programm zum Aufrufen von Anwendungen für sein Betriebssystem hat Tsu-Chen nun Zugriff auf 'DB2 Information - Unterstützung', um sich mit der Verwendung seines DB2-Produkts vertraut zu machen und so einen wertvollen Beitrag zum Erfolg seines Unternehmens leisten.

### Szenario: Zugriff auf 'DB2 Information - Unterstützung' über die IBM Website:

Colin ist IT-Berater bei einer Schulungsfirma. Er ist auf Datenbanktechnologie und SQL spezialisiert und hält Seminare zu diesen Themen für Unternehmen aus ganz Nordamerika ab. Hierfür verwendet er DB2 Universal Database. Im Rahmen seiner Seminare verwendet Colin die DB2-Dokumentation als Unterrichtsmaterial. Für SQL-Kurse beispielsweise verwendet Colin die DB2-Dokumentation zu SQL, um die grundlegende und erweiterte Syntax für Datenbankabfragen zu unterrichten.

Die meisten Unternehmen, bei denen Colin unterrichtet, verfügen über einen Internetzugang. Aus diesem Grund entschied sich Colin, seinen tragbaren Computer für den Zugriff auf 'DB2 Information - Unterstützung' über die Website von IBM zu konfigurieren, als er die letzte Version von DB2 Universal Database installiert hat. Diese Konfiguration ermöglicht es Colin, während seiner Seminare online auf die neueste DB2-Dokumentation zuzugreifen.

Wenn er auf Reisen ist, hat Colin bisweilen allerdings keinen Internetzugang. Dieser Umstand war für ihn recht problematisch, insbesondere dann, wenn er Zugriff auf die DB2-Dokumentation benötigte, um sich auf seine Seminare vorzubereiten. Um Situationen wie diese zu vermeiden, installierte Colin eine Kopie von 'DB2 Information - Unterstützung' auf seinem tragbaren Computer.

Auf diese Weise hat Colin nun jederzeit eine Kopie der DB2-Dokumentation zur Verfügung und ist dadurch wesentlich flexibler. Mit dem Befehl **db2set** kann Colin ohne Schwierigkeiten die Registrierdatenbankvariablen auf seinem tragbaren Computer so konfigurieren, dass er den jeweiligen Umständen entsprechend entweder über die Website von IBM oder über seinen tragbaren Computer auf 'DB2 Information - Unterstützung' zugreifen kann.

### Szenario: Zugriff auf 'DB2 Information - Unterstützung' über einen Intranet-Server:

Eva arbeitet als leitende Datenbankadministratorin für eine Lebensversicherung. In ihre Zuständigkeit fallen auch das Installieren und Konfigurieren der neuesten Version von DB2 Universal Database auf den UNIX<sup>®</sup>-basierten Datenbankservern des Unternehmens. Vor Kurzem hat das Unternehmen seine Mitarbeiter darüber informiert, dass sie aus Sicherheitsgründen während der Arbeitszeit keinen Internetzugang erhalten würden. Da ihr Unternehmen in einer Netzwerkumgebung arbeitet, beschließt Eva, eine Kopie von 'DB2 Information - Unterstützung' auf einem Intranet-Server zu installieren, damit alle Mitarbeiter, die das Data Warehouse des Unternehmens regelmäßig verwenden (Vertriebsbeauftragte, Vertriebsleiter und Geschäftsanalysten), Zugriff auf die DB2-Dokumentation haben.

Eva weist ihr Datenbankteam an, die neueste Version von DB2 Universal Database auf allen Computern der Mitarbeiter mit Hilfe einer Antwortdatei zu installieren, um sicherzustellen, dass die Konfiguration des Zugriffs auf 'DB2 Information - Unterstützung' auf allen Computern mit dem Hostnamen und der Portnummer des Intranet-Servers erfolgt.

Durch ein Missverständnis installiert jedoch Migual, ein Datenbankadministrator in Evas Team, eine Kopie von 'DB2 Information - Unterstützung' auf mehreren Mitarbeitercomputern, anstatt DB2 Universal Database für den Zugriff auf 'DB2 Information - Unterstützung' über den Intranet-Server zu konfigurieren. Um diesen Fehler zu korrigieren, weist Eva Migual an, mit dem Befehl **db2set** die Registrierdatenbankvariablen von 'DB2 Information - Unterstützung' (DB2\_DOCHOST für den Hostnamen und DB2\_DOCPORT für die Portnummer) auf allen entsprechen-

den Computern zu ändern. Anschließend haben nun alle erforderlichen Computer im Netzwerk Zugriff auf 'DB2 Information - Unterstützung', und die Mitarbeiter können mit Hilfe der DB2-Dokumentation Antworten auf ihre Fragen zu DB2 finden.

**Zugehörige Konzepte:**

- „DB2 Information - Unterstützung“ auf Seite 340

**Zugehörige Tasks:**

- „Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung'“ auf Seite 350
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (UNIX)“ auf Seite 344
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (Windows)“ auf Seite 347

**Zugehörige Referenzen:**

- „db2set - DB2 Profile Registry Command“ in *Command Reference*

---

## Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (UNIX)

Es gibt drei Möglichkeiten, auf die DB2-Produktdokumentation zuzugreifen: auf der IBM Website, auf einem Intranet-Server oder auf eine auf dem lokalen Computer installierte Version. Standardmäßig greifen DB2-Produkte auf die DB2-Dokumentation auf der IBM Website zu. Wenn Sie jedoch auf die DB2-Dokumentation auf einem Intranet-Server oder auf dem eigenen Computer zugreifen möchten, müssen Sie die Dokumentation von der CD 'DB2 Information - Unterstützung' aus installieren. Mit dem DB2-Installationsassistenten können Sie Ihre Installationseinstellungen definieren und 'DB2 Information - Unterstützung' auf einem Computer installieren, der das Betriebssystem UNIX verwendet.

**Voraussetzungen:**

Dieser Abschnitt erläutert die Voraussetzungen für Hardware, Betriebssystem, Software und Kommunikation zum Installieren von 'DB2 Information - Unterstützung' auf UNIX-Computern.

• **Hardwarevoraussetzungen**

Sie benötigen einen der folgenden Prozessoren:

- PowerPC (AIX)
- HP 9000 (HP-UX)
- Intel 32-Bit (Linux)
- Solaris UltraSPARC-Computer (Solaris-Betriebsumgebung)

• **Betriebssystemvoraussetzungen**

Sie benötigen eines der folgenden Betriebssysteme:

- IBM AIX 5.1 (auf PowerPC)
- HP-UX 11i (auf HP 9000)
- Red Hat Linux 8.0 (auf Intel 32-Bit)
- SuSE Linux 8.1 (auf Intel 32-Bit)

- Sun Solaris Version 8 (auf UltraSPARC-Computern in der Solaris-Betriebsumgebung)

**Anmerkung:** 'DB2 Information - Unterstützung' kann unter einem Teil der UNIX-Betriebssysteme ausgeführt werden, unter denen DB2-Clients unterstützt werden. Daher wird empfohlen, entweder über die IBM Website auf 'DB2 Information - Unterstützung' zuzugreifen oder 'DB2 Information - Unterstützung' auf einem Intranet-Server zu installieren und dort auf die Dokumentation zuzugreifen.

- **Softwarevoraussetzungen**

- Unterstützte Browser:

- Mozilla Version 1.0 oder höher

- Beim DB2-Installationsassistenten handelt es sich um ein grafisches Installationsprogramm. Um den DB2-Installationsassistenten auf Ihrem Computer ausführen zu können, benötigen Sie eine Implementierung der X Window System-Software zur Wiedergabe einer grafischen Benutzerschnittstelle (GUI). Bevor Sie den DB2-Installationsassistenten ausführen können, müssen Sie die entsprechende Anzeigefunktion (DISPLAY) unbedingt ordnungsgemäß exportieren. Geben Sie hierzu beispielsweise den folgenden Befehl an der Eingabeaufforderung ein:

```
export DISPLAY=9.26.163.144:0.
```

- **Kommunikationsvoraussetzungen**

- TCP/IP

**Vorgehensweise:**

Um 'DB2 Information - Unterstützung' mit Hilfe des DB2-Installationsassistenten zu installieren, gehen Sie wie folgt vor:

1. Melden Sie sich am System an.
2. Legen Sie die Produkt-CD von 'DB2 Information - Unterstützung' in das CD-Laufwerk ein, und hängen Sie die CD an Ihr System an.
3. Wechseln Sie in das Verzeichnis, in dem die CD angehängt ist. Geben Sie hierzu den folgenden Befehl ein:

```
cd /cd
```

Hierbei steht `/cd` für den Mountpunkt der CD.

4. Geben Sie den Befehl **`./db2setup`** ein, um den DB2-Installationsassistenten zu starten.
5. Die IBM DB2-Klickstartleiste wird geöffnet. Um direkt mit der Installation von 'DB2 Information - Unterstützung' fortzufahren, klicken Sie **Produkt installieren** an. Die Onlinehilfe enthält Informationen, die Sie durch die verbleibenden Schritte der Installation führen. Um die Onlinehilfe aufzurufen, klicken Sie **Hilfe** an. Sie können jederzeit **Abbrechen** anklicken, um die Installation zu beenden.
6. Klicken Sie im Fenster **Wählen Sie das zu installierende Produkt aus** den Knopf **Weiter** an.
7. Klicken Sie **Weiter** im Fenster **Willkommen beim DB2-Installationsassistenten** an. Der DB2-Installationsassistent leitet Sie durch die erforderlichen Schritte zum Installieren des Programms.
8. Um mit der Installation fortfahren zu können, müssen Sie die Lizenzvereinbarung akzeptieren. Wählen Sie auf der Seite **Lizenzvereinbarung** die Option **Bedingungen in der Lizenzvereinbarung anerkennen** aus, und klicken Sie **Weiter** an.

9. Wählen Sie **DB2 Information - Unterstützung auf diesem Computer installieren** auf der Seite **Installationsaktion auswählen** aus. Wenn Sie 'DB2 Information - Unterstützung' zu einem späteren Zeitpunkt auf diesem Computer oder anderen Computern mit Hilfe einer Antwortdatei installieren möchten, wählen Sie **Ihre Einstellungen in einer Antwortdatei speichern** aus. Klicken Sie **Weiter** an.
10. Wählen Sie auf der Seite **Zu installierende Sprachen auswählen** die Sprachen aus, in denen 'DB2 Information - Unterstützung' installiert werden soll. Klicken Sie den Knopf **Weiter** an.
11. Konfigurieren Sie 'DB2 Information - Unterstützung' auf der Seite **Port von DB2 Information - Unterstützung angeben** für eingehende Kommunikation. Klicken Sie **Weiter** an, um mit der Installation fortzufahren.
12. Überprüfen Sie auf der Seite **Kopieren der Dateien starten** noch einmal die von Ihnen ausgewählten Installationseinstellungen. Wenn Sie die Einstellungen ändern möchten, klicken Sie **Zurück** an. Klicken Sie **Installieren** an, um die Dateien von 'DB2 Information - Unterstützung' auf Ihren Computer zu kopieren.

Sie können 'DB2 Information - Unterstützung' auch mit Hilfe einer Antwortdatei installieren.

Die Installationsprotokolldateien db2setup.his, db2setup.log und db2setup.err befinden sich standardmäßig im Verzeichnis /tmp.

Die Datei db2setup.log erfasst alle Installationsinformationen zu DB2-Produkten, einschließlich Fehlern. Die Datei db2setup.his zeichnet alle DB2-Produktinstallationen auf Ihrem Computer auf. DB2 hängt die Datei db2setup.log an die Datei db2setup.his an. Die Datei db2setup.err erfasst die gesamte Fehlerausgabe, die von Java zurückgegeben wird, wie beispielsweise Informationen zu Ausnahmere Bedingungen und Traps.

Nach Abschluss der Installation ist 'DB2 Information - Unterstützung' je nach UNIX-Betriebssystem in einem der folgenden Verzeichnisse installiert:

- AIX: /usr/opt/db2\_08\_01
- HP-UX: /opt/IBM/db2/V8.1
- Linux: /opt/IBM/db2/V8.1
- Solaris-Betriebsumgebung: /opt/IBM/db2/V8.1

#### **Zugehörige Konzepte:**

- „DB2 Information - Unterstützung: Installationsszenarios“ auf Seite 341

#### **Zugehörige Tasks:**

- „Anzeigen von Themen in der gewünschten Sprache in 'DB2 Information - Unterstützung'“ auf Seite 351
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (Windows)“ auf Seite 347

---

## Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (Windows)

Es gibt drei Möglichkeiten, auf die DB2-Produktdokumentation zuzugreifen: auf der IBM Website, auf einem Intranet-Server oder auf eine auf dem lokalen Computer installierte Version. Standardmäßig greifen DB2-Produkte auf die DB2-Dokumentation auf der IBM Website zu. Wenn Sie jedoch auf die DB2-Dokumentation auf einem Intranet-Server oder auf dem eigenen Computer zugreifen möchten, müssen Sie die DB2-Dokumentation von der CD 'DB2 Information - Unterstützung' aus installieren. Mit dem DB2-Installationsassistenten können Sie Ihre Installationseinstellungen definieren und 'DB2 Information - Unterstützung' auf einem Computer installieren, der ein Windows-Betriebssystem verwendet.

### Voraussetzungen:

Dieser Abschnitt erläutert die Voraussetzungen für Hardware, Betriebssystem, Software und Kommunikation zum Installieren von 'DB2 Information - Unterstützung' unter Windows.

- **Hardwarevoraussetzungen**

Sie benötigen einen der folgenden Prozessoren:

- 32-Bit-Computer: eine Pentium- oder mit Pentium kompatible CPU

- **Betriebssystemvoraussetzungen**

Sie benötigen eines der folgenden Betriebssysteme:

- Windows 2000
- Windows XP

**Anmerkung:** 'DB2 Information - Unterstützung' kann unter einem Teil der Windows-Betriebssysteme ausgeführt werden, unter denen DB2-Clients unterstützt werden. Daher wird empfohlen, entweder über die IBM Website auf 'DB2 Information - Unterstützung' zuzugreifen oder 'DB2 Information - Unterstützung' auf einem Intranet-Server zu installieren und dort auf die Dokumentation zuzugreifen.

- **Softwarevoraussetzungen**

- Unterstützte Browser:
  - Mozilla 1.0 oder höher
  - Internet Explorer Version 5.5 oder 6.0 (Version 6.0 für Windows XP)

- **Kommunikationsvoraussetzungen**

- TCP/IP

### Einschränkungen:

- Sie benötigen einen Benutzereintrag mit Administratorberechtigung, um 'DB2 Information - Unterstützung' zu installieren.

### Vorgehensweise:

Um 'DB2 Information - Unterstützung' mit Hilfe des DB2-Installationsassistenten zu installieren, gehen Sie wie folgt vor:

1. Melden Sie sich mit dem für die Installation von 'DB2 Information - Unterstützung' definierten Benutzereintrag am System an.

2. Legen Sie die CD in das Laufwerk ein. Die IBM DB2 Setup-Klickstartleiste wird von der Funktion für automatische Ausführung gestartet, sofern diese Funktion aktiviert ist.
3. Der DB2-Installationsassistent ermittelt die Systemsprache und startet das Installationsprogramm für diese Sprache. Wenn Sie das Installationsprogramm nicht in Englisch ausführen möchten oder wenn beim automatischen Starten des Programms ein Fehler aufgetreten ist, können Sie den DB2-Installationsassistenten auch manuell starten.  
Um den DB2-Installationsassistenten manuell zu starten, gehen Sie wie folgt vor:
  - a. Klicken Sie **Start** an, und wählen Sie die Option **Ausführen** aus.
  - b. Geben Sie im Feld **Öffnen** den folgenden Befehl ein:  

```
x:\setup.exe /i zweistellige sprachenkennung
```

Hierbei steht *x*: für das CD-Laufwerk und *zweistellige sprachenkennung* für die Sprache, in der das Installationsprogramm ausgeführt werden soll.
  - c. Klicken Sie **OK** an.
4. Die IBM DB2-Klickstartleiste wird geöffnet. Um direkt mit der Installation von 'DB2 Information - Unterstützung' fortzufahren, klicken Sie **Produkt installieren** an. Die Onlinehilfe enthält Informationen, die Sie durch die verbleibenden Schritte der Installation führen. Um die Onlinehilfe aufzurufen, klicken Sie **Hilfe** an. Sie können jederzeit **Abbrechen** anklicken, um die Installation zu beenden.
5. Klicken Sie im Fenster **Wählen Sie das zu installierende Produkt** aus den Knopf **Weiter** an.
6. Klicken Sie **Weiter** im Fenster **Willkommen beim DB2-Installationsassistenten** an. Der DB2-Installationsassistent leitet Sie durch die erforderlichen Schritte zum Installieren des Programms.
7. Um mit der Installation fortfahren zu können, müssen Sie die Lizenzvereinbarung akzeptieren. Wählen Sie auf der Seite **Lizenzvereinbarung** die Option **Bedingungen in der Lizenzvereinbarung anerkennen** aus, und klicken Sie **Weiter** an.
8. Wählen Sie **DB2 Information - Unterstützung auf diesem Computer installieren** auf der Seite **Installationsaktion auswählen** aus. Wenn Sie 'DB2 Information - Unterstützung' zu einem späteren Zeitpunkt auf diesem Computer oder anderen Computern mit Hilfe einer Antwortdatei installieren möchten, wählen Sie **Ihre Einstellungen in einer Antwortdatei speichern** aus. Klicken Sie **Weiter** an.
9. Wählen Sie auf der Seite **Zu installierende Sprachen auswählen** die Sprachen aus, in denen 'DB2 Information - Unterstützung' installiert werden soll. Klicken Sie den Knopf **Weiter** an.
10. Konfigurieren Sie 'DB2 Information - Unterstützung' auf der Seite **Port von DB2 Information - Unterstützung angeben** für eingehende Kommunikation. Klicken Sie **Weiter** an, um mit der Installation fortzufahren.
11. Überprüfen Sie auf der Seite **Kopieren der Dateien starten** noch einmal die von Ihnen ausgewählten Installationseinstellungen. Wenn Sie die Einstellungen ändern möchten, klicken Sie **Zurück** an. Klicken Sie **Installieren** an, um die Dateien von 'DB2 Information - Unterstützung' auf Ihren Computer zu kopieren.

Sie haben die Möglichkeit, 'DB2 Information - Unterstützung' mit Hilfe einer Antwortdatei zu installieren. Sie können auch den Befehl **db2rspgn** verwenden, um eine Antwortdatei auf der Grundlage einer vorhandenen Installation zu generieren.

Die Dateien `db2.log` und `db2wi.log` im Verzeichnis 'Eigene Dateien'\DB2LOG\ enthalten Informationen zu Fehlern, die während der Installation aufgetreten sind. Die Position des Verzeichnisses 'Eigene Dateien' hängt von den Einstellungen Ihres Computers ab.

Die Datei `db2wi.log` erfasst die neuesten DB2-Installationsinformationen. Die Datei `db2.log` erfasst die Protokollinformationen von DB2-Produktinstallationen.

#### Zugehörige Konzepte:

- „DB2 Information - Unterstützung: Installationsszenarios“ auf Seite 341

#### Zugehörige Tasks:

- „Anzeigen von Themen in der gewünschten Sprache in 'DB2 Information - Unterstützung'“ auf Seite 351
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (UNIX)“ auf Seite 344

---

## Aufrufen von 'DB2 Information - Unterstützung'

'DB2 Information - Unterstützung' bietet Ihnen die Möglichkeit, auf alle Informationen zuzugreifen, die Sie zur Verwendung der DB2-Produkte für die Betriebssysteme Linux, UNIX und Windows, wie z. B. DB2 Universal Database, DB2 Connect, DB2 Information Integrator und DB2 Query Patroller, benötigen.

Rufen Sie 'DB2 Information - Unterstützung' auf eine der folgenden Arten auf:

- Von einem Computer aus, auf dem ein DB2 UDB-Client oder -Server installiert ist
- Von einem Intranet-Server oder einem lokalen Computer aus, auf dem 'DB2 Information - Unterstützung' installiert ist
- Über die IBM Website

#### Voraussetzungen:

Führen Sie vor dem Aufrufen von 'DB2 Information - Unterstützung' folgende Schritte aus:

- *Optional*: Konfigurieren des Browsers für die Anzeige der Themen in der gewünschten Landessprache
- *Optional*: Konfigurieren des DB2-Clients für die Verwendung der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung'

#### Vorgehensweise:

Gehen Sie wie folgt vor, um 'DB2 Information - Unterstützung' auf einem Computer aufzurufen, auf dem ein DB2 UDB-Client oder -Server installiert ist:

- Wählen Sie (unter Windows) **Start → Programme → IBM DB2 → Information → DB2 Information - Unterstützung** aus.
- Geben Sie in der Befehlszeile Folgendes ein:
  - Unter Linux und UNIX: Geben Sie den Befehl **db2icdocs** ein.
  - Unter Windows: Geben Sie den Befehl **db2icdocs.exe** ein.

Gehen Sie wie folgt vor, um die auf einem Intranet-Server oder lokalen Computer installierte Komponente 'DB2 Information - Unterstützung' in einem Webbrowser zu öffnen:

- Öffnen Sie die Webseite unter <http://<hostname>:<portnummer>/>. Dabei stellt <hostname> den Namen des Hosts dar und <portnummer> die Nummer des Ports, an dem 'DB2 Information - Unterstützung' verfügbar ist.

Gehen Sie wie folgt vor, um 'DB2 Information - Unterstützung' auf der IBM Website in einem Webbrowser zu öffnen:

- Öffnen Sie die Webseite unter [publib.boulder.ibm.com/infocenter/db2help/](http://publib.boulder.ibm.com/infocenter/db2help/).

#### **Zugehörige Konzepte:**

- „DB2 Information - Unterstützung“ auf Seite 340

#### **Zugehörige Tasks:**

- „Anzeigen von Themen in der gewünschten Sprache in 'DB2 Information - Unterstützung'“ auf Seite 351
- „Aufrufen der Kontexthilfe über ein DB2-Tool“ auf Seite 359
- „Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung'“ auf Seite 350
- „Aufrufen der Hilfe für Nachrichten über den Befehlszeilenprozessor“ auf Seite 360
- „Aufrufen der Hilfe für Befehle über den Befehlszeilenprozessor“ auf Seite 360
- „Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor“ auf Seite 361

---

## **Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung'**

Die Komponente 'DB2 Information - Unterstützung', auf die Sie über <http://publib.boulder.ibm.com/infocenter/db2help/> zugreifen können, wird in regelmäßigen Abständen durch neue oder geänderte Dokumentationen aktualisiert. IBM stellt in bestimmten Fällen auch Aktualisierungen von 'DB2 Information - Unterstützung' zum Download bereit, die Sie auf Ihrem Computer oder Intranet-Server installieren können. Durch die Aktualisierung von 'DB2 Information - Unterstützung' werden keine DB2-Client- oder -Serverprodukte aktualisiert.

#### **Voraussetzungen:**

Sie benötigen Zugriff auf einen Computer, der über eine Verbindung zum Internet verfügt.

#### **Vorgehensweise:**

Gehen Sie wie folgt vor, um die auf Ihrem Computer bzw. Intranet-Server installierte Komponente 'DB2 Information - Unterstützung' zu aktualisieren:

1. Öffnen Sie 'DB2 Information - Unterstützung' auf der IBM Website unter <http://publib.boulder.ibm.com/infocenter/db2help/>.

2. Klicken Sie im Downloadbereich der Eingangsseite den Link **DB2 Universal Database-Dokumentation** unter der Überschrift für Service und Unterstützung an.
3. Stellen Sie fest, ob die Version der installierten Komponente 'DB2 Information - Unterstützung' veraltet ist, indem Sie die Stufe des neuesten aktualisierten Dokumentationsimage mit der installierten Dokumentationsstufe vergleichen. Die installierte Dokumentationsstufe ist auf der Eingangsseite von 'DB2 Information - Unterstützung' aufgeführt.
4. Wenn eine neuere Version von 'DB2 Information - Unterstützung' verfügbar ist, laden Sie das neueste aktualisierte Image für *DB2 Information - Unterstützung* für das von Ihnen verwendete Betriebssystem herunter.
5. Befolgen Sie zur Installation des aktualisierten Image für *DB2 Information - Unterstützung* die Anweisungen auf der Webseite.

#### Zugehörige Konzepte:

- „DB2 Information - Unterstützung: Installationsszenarios“ auf Seite 341

#### Zugehörige Tasks:

- „Aufrufen von 'DB2 Information - Unterstützung'“ auf Seite 349
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (UNIX)“ auf Seite 344
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (Windows)“ auf Seite 347

---

## Anzeigen von Themen in der gewünschten Sprache in 'DB2 Information - Unterstützung'

In 'DB2 Information - Unterstützung' werden Themen, wenn möglich, in der Sprache angezeigt, die in den Vorgaben Ihres Browsers angegeben ist. Falls ein Thema nicht in die gewünschte Sprache übersetzt wurde, wird es in 'DB2 Information - Unterstützung' in Englisch angezeigt.

#### Vorgehensweise:

Um Themen in der gewünschten Sprache im Browser 'Internet Explorer' anzuzeigen, gehen Sie wie folgt vor:

1. Klicken Sie im Internet Explorer **Extras** —> **Internetoptionen...** —> **Sprachen...** an. Das Fenster **Spracheinstellung** wird geöffnet.
2. Stellen Sie sicher, dass die gewünschte Sprache als erster Eintrag in der Liste angegeben ist.
  - Klicken Sie den Knopf **Hinzufügen...** an, um eine neue Sprache zur Liste hinzuzufügen.

**Anmerkung:** Das Hinzufügen einer Sprache bedeutet nicht zwangsläufig, dass der Computer über die erforderlichen Schriftarten verfügt, um die Themen in der gewünschten Sprache anzuzeigen.

- Um eine Sprache an den Anfang der Liste zu verschieben, wählen Sie zunächst die gewünschte Sprache und anschließend den Knopf **Nach oben** aus, bis die Sprache an erster Stelle in der Liste steht.
3. Aktualisieren Sie die Seite, um 'DB2 Information - Unterstützung' in der gewünschten Sprache anzuzeigen.

Um Themen in der gewünschten Sprache im Browser 'Mozilla' anzuzeigen, gehen Sie wie folgt vor:

1. Wählen Sie in Mozilla **Bearbeiten** —> **Einstellungen** —> **Sprachen** aus. Die Anzeige für die Auswahl der Sprache wird im Fenster mit den Einstellungen aufgerufen.
2. Stellen Sie sicher, dass die gewünschte Sprache als erster Eintrag in der Liste angegeben ist.
  - Wenn Sie eine neue Sprache hinzufügen möchten, klicken Sie den Knopf **Hinzufügen...** an, um eine Sprache im entsprechenden Fenster auszuwählen.
  - Um eine Sprache an den Anfang der Liste zu verschieben, wählen Sie zunächst die gewünschte Sprache und anschließend den Knopf **Nach oben** aus, bis die Sprache an erster Stelle in der Liste steht.
3. Aktualisieren Sie die Seite, um 'DB2 Information - Unterstützung' in der gewünschten Sprache anzuzeigen.

---

## DB2-Dokumentation in PDF-Format und gedrucktem Format

In den folgenden Tabellen sind die offiziellen Buchtitel, Formularnummern und PDF-Dateinamen aufgeführt. Zum Bestellen von Hardcopybüchern benötigen Sie den offiziellen Buchtitel. Zum Drucken der PDF-Version benötigen Sie den PDF-Dateinamen.

Die DB2-Dokumentation ist in die folgenden Kategorien unterteilt:

- DB2-Kerninformationen
- Verwaltungsinformationen
- Informationen zur Anwendungsentwicklung
- Informationsmanagement
- Informationen zu DB2 Connect
- Einführungsinformationen
- Lernprogramminformationen
- Informationen zu Zusatzkomponenten
- Release-Informationen

In den folgenden Tabellen wird für die einzelnen Bücher der DB2-Bibliothek beschrieben, welche Informationen zum Bestellen von Hardcopies bzw. zum Drucken oder Anzeigen der PDF-Versionen erforderlich sind. Eine vollständige Beschreibung der in der DB2-Bibliothek verfügbaren Bücher finden Sie im IBM Publications Center unter folgender Adresse:  
[www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order).

### DB2-Kerninformationen

Diese Bücher enthalten grundlegende Informationen für alle DB2-Benutzer. Diese Informationen sind sowohl für Programmierer als auch für Datenbankadministratoren geeignet und unterstützen Sie bei der Arbeit mit DB2 Connect, DB2 Warehouse Manager und anderen DB2-Produkten.

Tabelle 25. DB2-Kerninformationen

Name	IBM Form	PDF-Dateiname
IBM DB2 Universal Database Command Reference	SC09-4828	db2n0e81
IBM DB2 Universal Database Glossar	Keine Formnummer	db2t0g81
IBM DB2 Universal Database Fehlernachrichten, Band 1	GC12-3043, nicht als Hardcopy verfügbar	db2m1g81
IBM DB2 Universal Database Fehlernachrichten, Band 2	GC12-3042, nicht als Hardcopy verfügbar	db2m2g81
IBM DB2 Universal Database Neue Funktionen	SC12-3044	db2q0g81

## Verwaltungsinformationen

Die Informationen in diesen Büchern umfassen die Themen, die zum effektiven Entwerfen, Implementieren und Verwalten von DB2-Datenbanken, Data Warehouses und Systemen zusammengesetzter Datenbanken erforderlich sind.

Tabelle 26. Verwaltungsinformationen

Name	IBM Form	PDF-Dateiname
IBM DB2 Universal Database Systemverwaltung: Konzept	SC12-3057	db2d1g81
IBM DB2 Universal Database Systemverwaltung: Implementierung	SC12-3059	db2d2g81
IBM DB2 Universal Database Systemverwaltung: Optimierung	SC12-3058	db2d3g81
IBM DB2 Universal Database Administrative API Reference	SC09-4824	db2b0e81
IBM DB2 Universal Database Dienstprogramme für das Versetzen von Daten Handbuch und Referenz	SC12-3055	db2dmg81
IBM DB2 Universal Database Datenwiederherstellung und hohe Verfügbarkeit Handbuch und Referenz	SC12-3054	db2hag81
IBM DB2 Universal Database Data Warehouse-Zentrale Verwaltung	SC12-3068	db2ddg81
IBM DB2 Universal Database SQL Reference, Volume 1	SC09-4844	db2s1e81
IBM DB2 Universal Database SQL Reference, Volume 2	SC09-4845	db2s2e81
IBM DB2 Universal Database System Monitor Guide and Reference	SC09-4847	db2f0e81

## Informationen zur Anwendungsentwicklung

Die Informationen in diesen Büchern sind besonders für Anwendungsentwickler und Programmierer von Interesse, die mit DB2 Universal Database (DB2 UDB) arbeiten. Sie finden hier Informationen zu den unterstützten Programmiersprachen und Compilern sowie die Dokumentation, die für den Zugriff auf DB2 UDB über die verschiedenen unterstützten Programmierschnittstellen, z. B. eingebettetes SQL, ODBC, JDBC, SQLJ und CLI, erforderlich ist. Wenn Sie die Komponente 'DB2 Information - Unterstützung' verwenden, können Sie auch auf HTML-Versionen des Quellcodes für die Beispielprogramme zugreifen.

*Tabelle 27. Informationen zur Anwendungsentwicklung*

<b>Name</b>	<b>IBM Form</b>	<b>PDF-Dateiname</b>
<i>IBM DB2 Universal Database Application Development Guide: Building and Running Applications</i>	SC09-4825	db2axe81
<i>IBM DB2 Universal Database Application Development Guide: Programming Client Applications</i>	SC09-4826	db2a1e81
<i>IBM DB2 Universal Database Application Development Guide: Programming Server Applications</i>	SC09-4827	db2a2e81
<i>IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1</i>	SC09-4849	db2l1e81
<i>IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2</i>	SC09-4850	db2l2e81
<i>IBM DB2 Universal Database Data Warehouse Center Application Integration Guide</i>	SC27-1124	db2ade81
<i>IBM DB2 XML Extender Verwaltung und Programmierung</i>	SC12-3062	db2sxxg81

## Informationsmanagement

Die Informationen in diesen Büchern beschreiben den Einsatz von Komponenten, mit denen Sie die Data Warehousing- und Analysefunktionen von DB2 Universal Database erweitern können.

*Tabelle 28. Informationsmanagement*

<b>Name</b>	<b>IBM Form</b>	<b>PDF-Dateiname</b>
<i>IBM DB2 Warehouse Manager Standard Edition Informationskatalogzentrale Verwaltung</i>	SC12-3070	db2dig81
<i>IBM DB2 Warehouse Manager Standard Edition Installation</i>	GC12-3069	db2idg81
<i>IBM DB2 Warehouse Manager Standard Edition Managing ETI Solution Conversion Programs with DB2 Warehouse Manager</i>	SC18-7727	iwhe1mste80

## Informationen zu DB2 Connect

Die Informationen in dieser Kategorie beschreiben den Zugriff auf Daten auf großen und mittleren Serversystemen mit Hilfe von DB2 Connect Enterprise Edition oder DB2 Connect Personal Edition.

*Tabelle 29. Informationen zu DB2 Connect*

Name	IBM Form	PDF-Dateiname
<i>IBM Konnektivität Ergänzung</i>	Keine Formnummer	db2h1g81
<i>IBM DB2 Connect Enterprise Edition Einstieg</i>	GC12-3051	db2c6g81
<i>IBM DB2 Connect Personal Edition Einstieg</i>	GC12-3049	db2c1g81
<i>IBM DB2 Connect Benutzerhandbuch</i>	SC12-3048	db2c0g81

## Einführungsinformationen

Die Informationen in dieser Kategorie unterstützen Sie beim Installieren und Konfigurieren von Servern, Clients und anderen DB2-Produkten.

*Tabelle 30. Einführungsinformationen*

Name	IBM Form	PDF-Dateiname
<i>IBM DB2 Universal Database für DB2-Clients Einstieg</i>	GC12-3052, nicht als Hardcopy verfügbar	db2itg81
<i>IBM DB2 Universal Database für DB2-Server Einstieg</i>	GC12-3047	db2isg81
<i>IBM DB2 Universal Database Personal Edition Einstieg</i>	GC12-3045	db2i1g81
<i>IBM DB2 Universal Database Installation und Konfiguration Ergänzung</i>	GC12-3046, nicht als Hardcopy verfügbar	db2iyg81
<i>IBM DB2 Universal Database Data Links Manager Einstieg</i>	GC12-3056	db2z6g81

## Lernprogramminformationen

In den Lernprogramminformationen werden DB2-Funktionen vorgestellt. Darüber hinaus wird die Ausführung verschiedener Tasks beschrieben.

*Tabelle 31. Lernprogramminformationen*

Name	IBM Form	PDF-Dateiname
<i>Lernprogramm für das Informationsmanagement: Data Warehouse - Einführung</i>	Keine Formnummer	db2tug81
<i>Lernprogramm für das Informationsmanagement: Data Warehouse - Weiterführende Informationen</i>	Keine Formnummer	db2tag81
<i>Lernprogramm für die Informationskatalogzentrale</i>	Keine Formnummer	db2aig81
<i>Video Central für e-business Lernprogramm</i>	Keine Formnummer	db2twg81

Tabelle 31. Lernprogramminformationen (Forts.)

Name	IBM Form	PDF-Dateiname
Lernprogramm für Visual Explain	Keine Formnummer	db2tv81

## Informationen zu Zusatzkomponenten

Die Informationen in dieser Kategorie beschreiben das Arbeiten mit den DB2-Zusatzkomponenten.

Tabelle 32. Informationen zu Zusatzkomponenten

Name	IBM Form	PDF-Dateiname
IBM DB2 Cube Views Handbuch und Referenz	n/v	db2aag81
IBM DB2 Query Patroller-Handbuch: Installation, Verwaltung und Verwendung	GC12-3225	db2dwg81
IBM DB2 Spatial Extender und Geodetic Extender Benutzer- und Referenzhandbuch	SC12-3063	db2sbg81
IBM DB2 Universal Database Data Links Manager Administration Guide and Reference	SC27-1221	db2z0e82
DB2 Net Search Extender Verwaltung und Benutzerhandbuch	SH12-3021	n/v

**Anmerkung:** Die HTML-Version dieses Dokuments wird nicht von der HTML-Dokumentations-CD installiert.

## Release-Informationen

Die Release-Informationen enthalten zusätzliche Informationen für das verwendete Produktrelease und die verwendete FixPak-Stufe. Die Release-Informationen enthalten außerdem Zusammenfassungen der Dokumentationsaktualisierungen in den verschiedenen Releases, Aktualisierungen und FixPaks.

Tabelle 33. Release-Informationen

Name	IBM Form	PDF-Dateiname
DB2 Release-Informationen	Siehe Anmerkung.	Siehe Anmerkung.
DB2 Installationsinformationen	Nur auf der Produkt-CD-ROM verfügbar.	n/v

**Anmerkung:** Die Release-Informationen stehen in den folgenden Formaten zur Verfügung:

- XHTML und Textformat auf den Produkt-CDs
- PDF-Format auf der CD mit der PDF-Dokumentation

Darüber hinaus sind die Abschnitte zu *bekanntten Problemen und Fehlerumgehungen* sowie zur *Inkompatibilität zwischen einzelnen Releases*, die Teil der Release-Informationen sind, auch über 'DB2 Information - Unterstützung' verfügbar.

Informationen zum Anzeigen der Release-Informationen in Textformat auf UNIX-Plattformen finden Sie in der Datei `Release.Notes`. Diese Datei befindet sich im Verzeichnis `DB2DIR/Readme/%L`. Hierbei steht `%L` für die länderspezifische Angabe und `DB2DIR` für eine der folgenden Angaben:

- Für AIX-Betriebssysteme: `/usr/opt/db2_08_01`
- Für alle anderen UNIX-Betriebssysteme: `/opt/IBM/db2/V8.1`

#### **Zugehörige Konzepte:**

- „DB2-Dokumentation und Hilfe“ auf Seite 339

#### **Zugehörige Tasks:**

- „Drucken von DB2-Büchern mit PDF-Dateien“ auf Seite 357
- „Bestellen gedruckter DB2-Bücher“ auf Seite 358
- „Aufrufen der Kontexthilfe über ein DB2-Tool“ auf Seite 359

---

## **Drucken von DB2-Büchern mit PDF-Dateien**

DB2-Bücher können mit Hilfe der PDF-Dateien auf der CD mit der *DB2-PDF-Dokumentation* gedruckt werden. Mit Adobe Acrobat Reader können Sie entweder das gesamte Handbuch oder bestimmte Seitenbereiche des Handbuchs ausdrucken.

#### **Voraussetzungen:**

Stellen Sie sicher, dass Adobe Acrobat Reader installiert ist. Falls Sie Adobe Acrobat Reader noch nicht installiert haben, finden Sie das Produkt auf der Adobe-Website unter folgender Adresse: [www.adobe.com](http://www.adobe.com)

#### **Vorgehensweise:**

Gehen Sie wie folgt vor, um ein DB2-Buch mit einer PDF-Datei auszudrucken:

1. Legen Sie die CD mit der *DB2-PDF-Dokumentation* in das CD-ROM-Laufwerk ein. Hängen Sie unter UNIX-Betriebssystemen die CD mit der *DB2-PDF-Dokumentation* an. Informationen zum Anhängen einer CD unter UNIX-Betriebssystemen finden Sie im Handbuch *Einstieg* für das jeweilige Betriebssystem.
2. Öffnen Sie `index.htm`. Die Datei wird in einem Browserfenster geöffnet.
3. Klicken Sie den Titel der PDF an, die Sie aufrufen möchten. Die PDF wird in Acrobat Reader geöffnet.
4. Wählen Sie **Datei** → **Drucken** aus, um einen beliebigen Teil des gewünschten Buches zu drucken.

#### **Zugehörige Konzepte:**

- „DB2 Information - Unterstützung“ auf Seite 340

#### **Zugehörige Tasks:**

- „Anhängen der CD-ROM (AIX)“ in *DB2 Universal Database für DB2-Server Einstieg*
- „Anhängen der CD-ROM (HP-UX)“ in *DB2 Universal Database für DB2-Server Einstieg*
- „Anhängen der CD-ROM (Linux)“ in *DB2 Universal Database für DB2-Server Einstieg*
- „Bestellen gedruckter DB2-Bücher“ auf Seite 358

- „Anhängen der CD-ROM (Solaris-Betriebsumgebung)“ in *DB2 Universal Database für DB2-Server Einstieg*

**Zugehörige Referenzen:**

- „DB2-Dokumentation in PDF-Format und gedrucktem Format“ auf Seite 352

---

## Bestellen gedruckter DB2-Bücher

Wenn Sie die Hardcopyversion der Bücher bevorzugen, können Sie sie auf eine der nachfolgend aufgeführten Arten bestellen.

**Vorgehensweise:**

In bestimmten Ländern oder Regionen können gedruckte Bücher bestellt werden. Auf der Website mit IBM Veröffentlichungen für das jeweilige Land bzw. die jeweilige Region finden Sie Informationen darüber, ob dieser Service im betreffenden Land bzw. in der betreffenden Region angeboten wird. Wenn die Veröffentlichungen bestellt werden können, haben Sie folgende Möglichkeiten:

- Wenden Sie sich an den zuständigen IBM Vertragshändler oder Vertriebsbeauftragten. Informationen zum lokalen IBM Ansprechpartner finden Sie im globalen IBM Verzeichnis für Kontakte unter folgender Adresse: [www.ibm.com/planetwide](http://www.ibm.com/planetwide).
- Weitere Informationen enthält das IBM Publications Center unter <http://www.ibm.com/shop/publications/order>. Die Möglichkeit, Bücher über das IBM Publications Center zu bestellen, besteht möglicherweise nicht in allen Ländern.

Die gedruckten Bücher sind zu dem Zeitpunkt, an dem das DB2-Produkt verfügbar gemacht wird, identisch mit den PDF-Versionen auf der CD mit der *DB2-PDF-Dokumentation*. Darüber hinaus stimmt der Inhalt der gedruckten Bücher mit den entsprechenden Informationen auf der CD für *DB2 Information - Unterstützung* überein. Diese CD enthält jedoch zusätzliche Informationen, die in den PDF-Büchern nicht enthalten sind (wie beispielsweise SQL-Verwaltungsroutinen und HTML-Beispiele). Nicht alle Bücher, die auf der CD mit der DB2-PDF-Dokumentation verfügbar sind, können als Hardcopy bestellt werden.

**Anmerkung:** 'DB2 Information - Unterstützung' wird häufiger aktualisiert als die PDF- oder die Hardcopyversion der Bücher. Installieren Sie die Dokumentationsupdates, sobald diese verfügbar sind, oder greifen Sie über 'DB2 Information - Unterstützung' unter <http://publib.boulder.ibm.com/infocenter/db2help/> auf die neuesten Informationen zu.

**Zugehörige Tasks:**

- „Drucken von DB2-Büchern mit PDF-Dateien“ auf Seite 357

**Zugehörige Referenzen:**

- „DB2-Dokumentation in PDF-Format und gedrucktem Format“ auf Seite 352

---

## Aufrufen der Kontexthilfe über ein DB2-Tool

Die Kontexthilfe bietet Informationen zu den Tasks bzw. Steuerelementen, die einem bestimmten Fenster, Notizbuch, Assistenten oder Advisor zugeordnet sind. Die Kontexthilfe steht in allen DB2-Verwaltungs- und -entwicklungstools zur Verfügung, die über eine grafische Benutzerschnittstelle verfügen. Zwei Arten der Kontexthilfe stehen zur Verfügung:

- Die über den Knopf **Hilfe** aufgerufenen Hilfetexte, der in jedem Fenster bzw. Notizbuch zur Verfügung steht.
- Die Kurzhilfe. Hierbei handelt es sich um Informationsfenster, die angezeigt werden, wenn sich der Mauszeiger auf einem Feld oder Steuerelement befindet oder wenn bei der Auswahl eines Feldes oder Steuerelements in einem Fenster, Notizbuch, Assistenten oder Advisor die Taste F1 gedrückt wird.

Über den Knopf **Hilfe** können Sie auf Übersichtsinformationen, Informationen zu Voraussetzungen sowie Informationen zu Tasks zugreifen. In der Kurzhilfe werden die einzelnen Felder und Steuerelemente beschrieben.

### Vorgehensweise:

Gehen Sie wie folgt vor, um Kontexthilfe aufzurufen:

- Hilfe zu Fenstern und Notizbüchern können Sie anzeigen, indem Sie eines der DB2-Tools aufrufen und anschließend ein beliebiges Fenster oder Notizbuch öffnen. Klicken Sie den Knopf **Hilfe** in der rechten unteren Ecke des Fensters bzw. Notizbuchs an, um die Kontexthilfe aufzurufen.

Zugriff auf die Kontexthilfe besteht darüber hinaus über den Menüpunkt **Hilfe** am oberen Rand jeder Zentrale der DB2-Tools.

Innerhalb von Assistenten und Advisorfunktionen klicken Sie den Link für die Taskübersicht auf der ersten Seite an, um die Kontexthilfe aufzurufen.

- Kurzhilfe zu einzelnen Steuerelementen eines Fensters oder Notizbuchs können Sie aufrufen, indem Sie das gewünschte Steuerelement anklicken und anschließend **F1** drücken. Die Kurzhilfeinformationen mit Details zum jeweiligen Steuerelement werden in einem gelben Fenster angezeigt.

**Anmerkung:** Wenn die Kurzhilfe angezeigt werden soll, sobald sich der Mauszeiger auf einem Feld oder Steuerelement befindet, wählen Sie das Markierungsfeld **Kurzhilfe automatisch anzeigen** auf der Seite **Dokumentation** des Notizbuchs 'Tools - Einstellungen' aus.

Ähnlich wie die Kurzhilfe sind auch Dialogfenster mit Diagnoseinformationen eine Form der kontextbezogenen Hilfe; sie enthalten Regeln für die Dateneingabe. Diese Diagnoseinformationen werden in einem violetten Fenster angezeigt, das aufgerufen wird, wenn die eingegebenen Daten nicht gültig oder nicht ausreichend sind. Die Kontexthilfe mit Diagnoseinformationen kann für folgende Felder angezeigt werden:

- Musseingabefelder
- Felder, in denen die Daten einem bestimmten Format entsprechen müssen, wie z. B. Datumsfelder

**Zugehörige Tasks:**

- „Aufrufen von 'DB2 Information - Unterstützung'" auf Seite 349
- „Aufrufen der Hilfe für Nachrichten über den Befehlszeilenprozessor" auf Seite 360
- „Aufrufen der Hilfe für Befehle über den Befehlszeilenprozessor" auf Seite 360
- „Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor" auf Seite 361

---

## Aufrufen der Hilfe für Nachrichten über den Befehlszeilenprozessor

Die Hilfe für Nachrichten beschreibt die Ursache von Nachrichten und die Aktionen, die der Benutzer zur Behebung des aufgetretenen Fehlers ausführen sollte.

**Vorgehensweise:**

Zum Aufrufen der Hilfe für Nachrichten müssen Sie den Befehlszeilenprozessor öffnen und Folgendes eingeben:

`? XXXnnnnn`

Dabei ist `XXXnnnnn` eine gültige Nachrichtenennung.

So kann beispielsweise durch die Eingabe von `? SQL30081` die Hilfe zur Nachricht `SQL30081` angezeigt werden.

**Zugehörige Tasks:**

- „Aufrufen der Kontexthilfe über ein DB2-Tool" auf Seite 359
- „Aufrufen von 'DB2 Information - Unterstützung'" auf Seite 349
- „Aufrufen der Hilfe für Befehle über den Befehlszeilenprozessor" auf Seite 360
- „Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor" auf Seite 361

**Zugehörige Referenzen:**

- „db2 - Command Line Processor Invocation Command" in *Command Reference*

---

## Aufrufen der Hilfe für Befehle über den Befehlszeilenprozessor

Die Hilfe für Befehle erläutert die Syntax von Befehlen im Befehlszeilenprozessor.

**Vorgehensweise:**

Zum Aufrufen der Hilfe für Befehle müssen Sie den Befehlszeilenprozessor öffnen und Folgendes eingeben:

`? command`

Dabei stellt `command` ein Schlüsselwort bzw. den vollständigen Befehl dar.

So kann beispielsweise durch die Eingabe von `? catalog` Hilfe für alle CATALOG-Befehle angezeigt werden, während mit `? catalog database` nur Hilfe für den Befehl `CATALOG DATABASE` angezeigt wird.

#### Zugehörige Tasks:

- „Aufrufen der Kontexthilfe über ein DB2-Tool“ auf Seite 359
- „Aufrufen von 'DB2 Information - Unterstützung'“ auf Seite 349
- „Aufrufen der Hilfe für Nachrichten über den Befehlszeilenprozessor“ auf Seite 360
- „Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor“ auf Seite 361

#### Zugehörige Referenzen:

- „db2 - Command Line Processor Invocation Command“ in *Command Reference*

---

## Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor

DB2 Universal Database gibt für Bedingungen, die auf Grund einer SQL-Anweisung generiert werden können, einen SQLSTATE-Wert zurück. Die SQLSTATE-Hilfe erläutert die Bedeutung der SQL-Statuswerte und der SQL-Statusklassencodes.

#### Vorgehensweise:

Zum Aufrufen der Hilfe für SQL-Statuswerte müssen Sie den Befehlszeilenprozessor öffnen und Folgendes eingeben:

```
? sqlstate oder ? klassencode
```

Hierbei steht *sqlstate* für einen gültigen fünfstelligen SQL-Statuswert und *klassencode* für die ersten beiden Ziffern dieses Statuswertes.

So kann beispielsweise durch die Eingabe von ? 08003 Hilfe für den SQL-Statuswert 08003 angezeigt werden, durch die Eingabe von ? 08 Hilfe für den Klassencode 08.

#### Zugehörige Tasks:

- „Aufrufen von 'DB2 Information - Unterstützung'“ auf Seite 349
- „Aufrufen der Hilfe für Nachrichten über den Befehlszeilenprozessor“ auf Seite 360
- „Aufrufen der Hilfe für Befehle über den Befehlszeilenprozessor“ auf Seite 360

---

## DB2-Lernprogramme

Die Lernprogramme von DB2<sup>®</sup> unterstützen Sie bei der Einarbeitung in die verschiedenen Themenbereiche von DB2 Universal Database. Sie umfassen Übungen mit in einzelne Arbeitsschritte untergliederten Anweisungen zum Entwickeln von Anwendungen, Optimieren der SQL-Abfrageleistung, Arbeiten mit Data Warehouses, Verwalten von Metadaten und Entwickeln von Webservices mit Hilfe von DB2.

#### Vorbereitungen:

Die XHTML-Version der Lernprogramme kann über 'DB2 Information - Unterstützung' unter <http://publib.boulder.ibm.com/infocenter/db2help/> angezeigt werden.

In einigen der Lernprogrammübungen werden Beispieldaten und Codebeispiele verwendet. Informationen zu den spezifischen Voraussetzungen zur Ausführung der Tasks finden Sie in der Beschreibung des jeweiligen Lernprogramms.

### **Lernprogramme von DB2 Universal Database:**

Klicken Sie einen der Lernprogrammtitel in der folgenden Liste an, um das entsprechende Lernprogramm aufzurufen.

*Lernprogramm für das Informationsmanagement: Data Warehouse - Einführung*  
Ausführung grundlegender Data Warehousing-Tasks mit Hilfe der Data Warehouse-Zentrale.

*Lernprogramm für das Informationsmanagement: Data Warehouse - Weiterführende Informationen*  
Ausführung weiterführender Data Warehousing-Tasks mit Hilfe der Data Warehouse-Zentrale.

*Lernprogramm für die Informationskatalogzentrale*  
Erstellen und Verwalten eines Informationskatalogs zum Lokalisieren und Verwenden von Metadaten mit Hilfe der Informationskatalogzentrale.

*Lernprogramm für Visual Explain*  
Analysieren, Optimieren und Anpassen von SQL-Anweisungen zur Leistungsverbesserung mit Hilfe von Visual Explain.

---

## **Informationen zur Fehlerbehebung in DB2**

Eine breite Palette verschiedener Informationen zur Fehlerbestimmung und Fehlerbehebung steht zur Verfügung, um Sie bei der Verwendung von DB2<sup>®</sup>-Produkten zu unterstützen.

### **DB2-Dokumentation**

Informationen zur Fehlerbehebung stehen in der gesamten Komponente 'DB2 Information - Unterstützung' sowie in den PDF-Büchern der DB2-Bibliothek zur Verfügung. Folgen Sie der Verzweigung 'Unterstützung und Fehlerbehebung' in der Navigationsbaumstruktur von 'DB2 Information - Unterstützung' (im linken Teilfenster des Browserfensters), um eine umfassende Liste der DB2-Dokumentationen zur Fehlerbehebung aufzurufen.

### **DB2-Website mit technischer Unterstützung**

Auf der DB2-Website mit technischer Unterstützung finden Sie Informationen zu Problemen und den möglichen Ursachen und Fehlerbehebungsmaßnahmen. Die DB2-Website mit technischer Unterstützung stellt Links zu den neuesten DB2-Veröffentlichungen, technischen Hinweisen (TechNotes), APARs (Authorized Program Analysis Reports), FixPaks, den neuesten Listen mit internen DB2-Fehlercodes sowie weiteren Ressourcen zur Verfügung. Sie können diese Wissensbasis nach möglichen Lösungen für aufgetretene Probleme durchsuchen.

Rufen Sie die DB2-Website mit technischer Unterstützung unter <http://www.ibm.com/software/data/db2/udb/winos2unix/support> auf.

## DB2-Lernprogramme zur Fehlerbestimmung

Auf der Website mit den DB2-Lernprogrammen zur Fehlerbestimmung finden Sie Informationen dazu, wie Sie Fehler, die bei der Verwendung von DB2-Produkten möglicherweise auftreten, rasch identifizieren und beheben können. Eines der Lernprogramme bietet eine Einführung in die verfügbaren DB2-Einrichtungen und -Tools zur Fehlerbestimmung sowie Entscheidungshilfen für deren Verwendung. Andere Lernprogramme befassen sich mit zugehörigen Themen, wie beispielsweise der Fehlerbestimmung für die Datenbanksteuerkomponente, der Fehlerbestimmung für die Leistung und der Fehlerbestimmung für Anwendungen.

Die vollständige Liste der DB2-Lernprogramme zur Fehlerbestimmung finden Sie auf der DB2-Website mit technischer Unterstützung unter <http://www.ibm.com/software/data/support/pdm/db2tutorials.html>.

### Zugehörige Konzepte:

- „DB2 Information - Unterstützung“ auf Seite 340
- „Einführung in die Fehlerbestimmung - Lernprogramm für die technische Unterstützung in DB2“ im Handbuch *Fehlerbehebung*

---

## Eingabehilfen

Eingabehilfen unterstützen Benutzer mit körperlichen Behinderungen, wie z. B. eingeschränkter Bewegungsfähigkeit oder Sehkraft, beim erfolgreichen Einsatz von Softwareprodukten. Im Folgenden sind die wichtigsten Eingabehilfen aufgeführt, die in den Produkten von DB2<sup>®</sup> Version 8 zur Verfügung stehen:

- Die gesamte DB2-Funktionalität kann sowohl über die Maus als auch über die Tastatur gesteuert werden. Weitere Informationen hierzu finden Sie unter „Tastatureingabe und Navigation“.
- Sie können die Größe und Farbe der verwendeten Schriftarten in den DB2-Schnittstellen anpassen. Weitere Informationen hierzu finden Sie unter „Eingabehilfen für Bildschirme“ auf Seite 364.
- DB2-Produkte unterstützen Anwendungen mit Eingabehilfen, die mit der Java<sup>™</sup> Accessibility API arbeiten. Weitere Informationen hierzu finden Sie unter „Kompatibilität mit Unterstützungseinrichtungen“ auf Seite 364.
- Die DB2-Dokumentation steht in behindertengerechtem Format zur Verfügung. Weitere Informationen hierzu finden Sie unter „Dokumentation im behindertengerechten Format“ auf Seite 364.

## Tastatureingabe und Navigation

### Tastatureingabe

Die verfügbaren DB2-Tools können unter ausschließlicher Benutzung der Tastatur verwendet werden. Mit entsprechenden Tasten oder Tastenkombinationen können Operationen ausgeführt werden, die auch über die Maus verfügbar sind. Die Standardtastenkombinationen des Betriebssystems werden für die entsprechenden Standardoperationen des Betriebssystems verwendet.

Weitere Informationen zur Verwendung von Tasten oder Tastenkombinationen für die Ausführung von Operationen finden Sie unter " 'Direktaufrufe über die Tastatur: Gemeinsame GUI - Hilfe'.

## **Navigation über die Tastatureingabe**

Sie können in den Benutzerschnittstellen der DB2-Tools mit Hilfe von Tasten oder Tastenkombinationen navigieren.

Weitere Informationen zur Navigation in den DB2-Tools mit Hilfe der Tastatureingabe finden Sie unter " 'Direktaufrufe über die Tastatur: Gemeinsame GUI - Hilfe'.

## **Tastatureingabebereich**

Unter UNIX®-Betriebssystemen ist der Bereich des aktiven Fensters, in dem die Tastatureingabe wirksam ist, hervorgehoben.

## **Eingabehilfen für Bildschirme**

Die DB2-Tools stellen Funktionen bereit, mit denen sehbehinderten Benutzern verbesserten Eingabehilfen zur Verfügung stehen. Diese Eingabehilfen umfassen die Unterstützung individuell anpassbarer Schriftarteigenschaften.

### **Schriftarteneinstellungen**

Über das Notizbuch 'Tools - Einstellungen' können Sie die Farbe, Größe und Schriftart des Textes in Menüs und Dialogfenstern auswählen.

Weitere Informationen zur Angabe von Schriftarteneinstellungen finden Sie unter " 'Ändern der Schriftarten für Menüs und Text: Gemeinsame GUI - Hilfe'.

### **Unabhängigkeit von Farben**

Zur Verwendung der Funktionen des vorliegenden Produkts ist es nicht erforderlich, zwischen unterschiedlichen Farben differenzieren zu können.

## **Kompatibilität mit Unterstützungseinrichtungen**

Die Schnittstellen der DB2-Tools unterstützen die Java Accessibility API. Hierdurch wird der Einsatz von Sprachausgabeprogrammen und anderen Unterstützungseinrichtungen für Personen mit Behinderungen mit den DB2-Produkten ermöglicht.

## **Dokumentation im behindertengerechten Format**

Die Dokumentation für DB2 steht im Format XHTML 1.0 zur Verfügung, das mit den meisten Webbrowsern geöffnet werden kann. XHTML ermöglicht das Aufrufen der Dokumentation mit den Anzeigeeinstellungen, die Sie in Ihrem Browser definiert haben. Darüber hinaus ist der Einsatz von Sprachausgabeprogrammen und anderen Unterstützungseinrichtungen möglich.

Syntaxdiagramme stehen in der Schreibweise mit Trennzeichen zur Verfügung. Dieses Format ist nur dann verfügbar, wenn Sie mit Hilfe eines Sprachausgabeprogramms auf die Onlinedokumentation zugreifen.

### **Zugehörige Konzepte:**

- „Syntaxdiagramme in der Schreibweise mit Trennzeichen“ auf Seite 365

---

## Syntaxdiagramme in der Schreibweise mit Trennzeichen

Syntaxdiagramme stehen für Benutzer, die mit Hilfe eines Sprachausgabe-  
programms auf 'DB2 Information - Unterstützung' zugreifen, in der Schreibweise  
mit Trennzeichen zur Verfügung.

In der Schreibweise mit Trennzeichen steht jedes Syntaxelement in einer separaten  
Zeile. Wenn zwei oder mehr Syntaxelemente stets gemeinsam angegeben (oder  
nicht angegeben) werden müssen, können sie in derselben Zeile stehen, da sie als  
ein zusammengesetztes Syntaxelement betrachtet werden können.

Jede Zeile beginnt mit einer Zahl in der Schreibweise mit Trennzeichen, zum Bei-  
spiel 3 oder 3.1 oder 3.1.1. Um diese Zahlen korrekt zu hören, müssen Sie sicher-  
stellen, dass das Sprachausgabeprogramm so konfiguriert ist, dass die Interpunk-  
tion angesagt wird. Alle Syntaxelemente mit derselben Zahl in der Schreibweise  
mit Trennzeichen (z. B. alle Syntaxelemente mit der Zahl 3.1) stellen Alternativen  
dar, die sich gegenseitig ausschließen. Wenn Sie die Zeilen '3.1 USERID' und '3.1  
SYSTEMID' hören, wissen Sie, dass die Syntax entweder USERID oder SYSTEMID  
enthalten kann, nicht jedoch beides.

Die Nummerierung bei der Schreibweise mit Trennzeichen gibt den Grad der Aus-  
gliederung an. Beispiel: Wenn auf das Syntaxelement mit der Zahl 3 in der Schreib-  
weise mit Trennzeichen eine Reihe von Syntaxelementen mit der Zahl 3.1 folgt,  
sind alle Syntaxelemente mit der Zahl 3.1 dem Syntaxelement mit der Zahl 3 unter-  
geordnet.

Bestimmte Wörter und Symbole werden zusätzlich zu den Zahlen in der Schreib-  
weise mit Trennzeichen verwendet, um weitere Informationen zu den Syntax-  
elementen anzugeben. In manchen Fällen können diese Wörter und Symbole am  
Anfang des Elements selbst stehen. Zur einfacheren Identifizierung wird dem Wort  
oder Symbol ein umgekehrter Schrägstrich (\) vorangestellt, wenn es Teil des  
Syntaxelements ist. Das Symbol \* (Stern) kann zusätzlich zu einer Zahl in der  
Schreibweise mit Trennzeichen verwendet werden, um anzugeben, dass das  
Syntaxelement wiederholt wird. Beispiel: Das Syntaxelement \*FILE mit der Zahl 3  
in der Schreibweise mit Trennzeichen erhält das Format 3 \\* FILE. Format 3\* FILE  
gibt an, dass das Syntaxelement FILE wiederholt wird. Format 3\* \\* FILE gibt an,  
dass das Syntaxelement \* FILE wiederholt wird.

Zeichen wie beispielsweise Kommas, die bei einer Folge von Syntaxelementen als  
Trennzeichen verwendet werden, werden in der Syntax unmittelbar vor den Ele-  
menten dargestellt, die sie trennen. Diese Zeichen können in derselben Zeile stehen  
wie das jeweilige Element oder in einer separaten Zeile mit derselben Zahl in der  
Schreibweise mit Trennzeichen, die auch dem betreffenden Element zugeordnet ist.  
Die Zeile kann auch ein weiteres Symbol enthalten, das Informationen zu den  
Syntaxelementen angibt. So bedeuten z. B. die Zeilen 5.1\*, 5.1 LASTRUN und 5.1  
DELETE, dass, wenn Sie mehr als eines der Elemente LASTRUN und DELETE ver-  
wenden, diese Elemente durch Kommas voneinander getrennt werden müssen.  
Wenn kein Trennzeichen angegeben wird, verwendet das System zum Trennen der  
einzelnen Syntaxelemente ein Leerzeichen.

Wenn einem Syntaxelement das Symbol % vorangestellt ist, gibt dies einen Verweis  
an, der an anderer Stelle definiert ist. Die Zeichenfolge, die auf das Symbol %  
folgt, ist der Name eines Syntaxfragments und kein Literal. So gibt die Zeile 2.1  
%OP1 beispielsweise einen Verweis auf das separate Syntaxfragment OP1 an.

Die nachfolgend aufgeführten Wörter und Symbole werden zusätzlich zu den Zahlen in der Schreibweise mit Trennzeichen verwendet:

- ? stellt ein optionales Syntaxelement dar. Eine Zahl in der Schreibweise mit Trennzeichen, auf die das Symbol ? (Fragezeichen) folgt, gibt an, dass alle Syntaxelemente mit einer entsprechenden Zahl in der Schreibweise mit Trennzeichen sowie alle untergeordneten Syntaxelemente optional sind. Ist nur ein Syntaxelement mit einer Zahl in der Schreibweise mit Trennzeichen vorhanden, wird das Symbol ? in derselben Zeile angezeigt wie das Syntaxelement (zum Beispiel 5? NOTIFY). Sind mehrere Syntaxelemente mit einer Zahl in der Schreibweise mit Trennzeichen vorhanden, wird das Symbol ? in einer separaten Zeile angezeigt, gefolgt von den optionalen Syntaxelementen. Wenn Sie beispielsweise die Zeilen 5 ?, 5 NOTIFY und 5 UPDATE hören, wissen Sie, dass die Syntaxelemente NOTIFY und UPDATE optional sind; das bedeutet, Sie können eines oder keines dieser Elemente auswählen. Das Symbol ? entspricht einer Umgehungslinie in einem Pfeildiagramm.
- ! stellt ein Standardsyntaxelement dar. Eine Zahl in der Schreibweise mit Trennzeichen, auf die das Symbol ! (Ausrufezeichen) und ein Syntaxelement folgen, gibt an, dass es sich bei diesem Syntaxelement um die Standardoption für alle Syntaxelemente handelt, denen dieselbe Zahl in der Schreibweise mit Trennzeichen zugeordnet ist. Nur für eines der Syntaxelemente, denen dieselbe Zahl in der Schreibweise mit Trennzeichen zugeordnet ist, darf das Symbol ! angegeben werden. Wenn Sie beispielsweise die Zeilen 2? FILE, 2.1! (KEEP) und 2.1 (DELETE) hören, wissen Sie, dass (KEEP) die Standardoption für das Schlüsselwort FILE ist. Wenn Sie in diesem Beispiel das Schlüsselwort FILE verwenden, jedoch keine Option angeben, wird die Standardoption KEEP verwendet. Eine Standardoption ist auch für die nächsthöhere Zahl in der Schreibweise mit Trennzeichen gültig. In diesem Beispiel bedeutet das: Wenn das Schlüsselwort FILE weggelassen wird, wird der Standardwert FILE(KEEP) verwendet. Wenn Sie jedoch die Zeilen 2? FILE, 2.1, 2.1.1! (KEEP) und 2.1.1 (DELETE) hören, gilt die Standardoption KEEP nur für die nächsthöhere Zahl in der Schreibweise mit Trennzeichen, 2.1 (der kein Schlüsselwort zugeordnet ist), nicht jedoch für 2? FILE. Wird das Schlüsselwort FILE weggelassen, wird kein Wert verwendet.
- \* stellt ein Syntaxelement dar, das keinmal, einmal oder mehrmals wiederholt werden kann. Eine Zahl in der Schreibweise mit Trennzeichen, auf die das Symbol \* (Stern) folgt, gibt an, dass dieses Syntaxelement keinmal, einmal oder mehrmals verwendet werden kann, d. h., es ist optional und kann wiederholt werden. Wenn Sie beispielsweise die Zeile 5.1\* Datenbereich hören, wissen Sie, dass Sie einen, mehrere oder keinen Datenbereich angeben können. Hören Sie die Zeilen 3\*, 3 HOST und 3 STATE, wissen Sie, dass Sie HOST, STATE, beide oder keines der Elemente angeben können.

#### **Anmerkungen:**

1. Wenn neben einer Zahl in der Schreibweise mit Trennzeichen ein Stern (\*) angezeigt wird und nur ein Element mit dieser Zahl vorhanden ist, können Sie dieses Element mehrmals wiederholen.
2. Wenn neben einer Zahl in der Schreibweise mit Trennzeichen ein Stern angezeigt wird und diese Zahl mehreren Elementen zugeordnet ist, können Sie mehrere Elemente aus der Liste verwenden, jedes davon jedoch nur einmal. Im vorhergehenden Beispiel könnten Sie HOST STATE angeben, nicht jedoch HOST HOST.
3. Das Symbol \* entspricht einer zum Ausgangspunkt zurück führenden Linie in einem Pfeildiagramm.

- + stellt ein Syntaxelement dar, das mindestens einmal angegeben werden muss. Eine Zahl in der Schreibweise mit Trennzeichen, auf die das Symbol + (Pluszeichen) folgt, gibt an, dass dieses Syntaxelement mindestens einmal angegeben werden muss und wiederholt werden kann. Wenn Sie beispielsweise die Zeile 6.1+ Datenbereich hören, müssen sie mindestens einen Datenbereich angeben. Wenn Sie die Zeilen 2+, 2 HOST und 2 STATE hören, wissen Sie, dass Sie HOST, STATE oder beides angeben müssen. Wie auch für das Symbol \* gilt hier, dass mit dem Pluszeichen ein bestimmtes Element nur dann wiederholt werden kann, wenn es sich um das einzige Element mit dieser Zahl in der Schreibweise mit Trennzeichen handelt. Das Symbol + entspricht wie das Symbol \* einer zum Ausgangspunkt zurück führenden Linie in einem Pfeildiagramm.

**Zugehörige Referenzen:**

- „How to read the syntax diagrams“ in *SQL Reference, Volume 2*

---

## **Common Criteria-Zertifizierung von DB2 Universal Database-Produkten**

Für DB2 Universal Database läuft momentan der Bewertungsprozess für die Zertifizierung entsprechend der Richtlinien von Common Criteria Evaluation Assurance Level 4 (EAL4). Weitere Informationen zu Common Criteria finden Sie auf der Common Criteria-Website unter: <http://niap.nist.gov/cc-scheme/>.



---

## Anhang I. Bemerkungen

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Dienstleistungen von IBM verwendet werden können. An Stelle der IBM Produkte, Programme oder Dienstleistungen können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb der Produkte, Programme oder Dienstleistungen in Verbindung mit Fremdprodukten und Fremddienstleistungen liegt beim Kunden, soweit nicht ausdrücklich solche Verbindungen erwähnt sind.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Europe, Director of Licensing, 92066 Paris La Defense Cedex, France.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen bekanntgegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Canada Limited  
Office of the Lab Director  
8200 Warden Avenue  
Markham, Ontario  
L6G 1C7  
CANADA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Handbuch aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt im Rahmen der Allgemeinen Geschäftsbedingungen der IBM, der Internationalen Nutzungsbedingungen der IBM für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer gesteuerten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Garantie, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Informationen über Produkte anderer Hersteller als IBM wurden von den Herstellern dieser Produkte zur Verfügung gestellt, bzw. aus von ihnen veröffentlichten Ankündigungen oder anderen öffentlich zugänglichen Quellen entnommen. IBM hat diese Produkte nicht getestet und übernimmt im Hinblick auf Produkte anderer Hersteller keine Verantwortung für einwandfreie Funktion, Kompatibilität oder andere Ansprüche. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten der IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele der IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogrammes illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden, Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

#### COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Kopien oder Teile der Beispielprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. *Jahr/Jahre angeben*. Alle Rechte vorbehalten.

---

## Marken

Folgende Namen sind in gewissen Ländern Marken der International Business Machines Corporation und wurden in mindestens einem der Dokumente in der DB2 UDB-Dokumentationsbibliothek verwendet:

ACF/VTAM	iSeries
AISPO	LAN Distance
AIX	MVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
IBM System AS/400	NetView
BookManager	OS/390
C Set++	OS/400
C/370	PowerPC
CICS	pSeries
Database 2	QBIC
DataHub	QMF
DataJoiner	RACF
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/400
DB2 Extenders	SQL/DS
DB2 OLAP Server	System/370
DB2 Information Integrator	IBM System /390
DB2 Query Patroller	SystemView
DB2 Universal Database	Tivoli
Distributed Relational Database Architecture	VisualAge
DRDA	VM/ESA
eServer	VSE/ESA
Extended Services	VTAM
FFST	WebExplorer
First Failure Support Technology	WebSphere
IBM	WIN-OS/2
IMS	z/OS
IMS/ESA	zSeries

Folgende Namen sind in gewissen Ländern Marken oder eingetragene Marken anderer Unternehmen und wurden in mindestens einem der Dokumente in der DB2 UDB-Dokumentationsbibliothek verwendet.

Microsoft, Windows, Windows NT und das Windows-Logo sind in gewissen Ländern Marken der Microsoft Corporation.

Intel und Pentium sind in gewissen Ländern Marken der Intel Corporation.

Java und alle auf Java basierenden Marken sind in gewissen Ländern Marken von Sun Microsystems, Inc.

UNIX ist in gewissen Ländern eine eingetragene Marke von The Open Group.

Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken anderer Unternehmen sein.



# Index

## A

Abfrage-API für Status des Ladevorgangs 163  
Aktualisieren  
  HTML-Dokumentation 350  
Aktualisierende Wiederherstellung (Dienstprogramm)  
  Datei mit Angaben zur Speicherposition der Ladekopie verwenden 110  
Änderungswerte  
  Dateityp  
    EXPORT (Befehl) 9  
    IMPORT (Befehl) 39  
    LOAD (Befehl) 112  
Änderungswerte für Dateityp  
  Dienstprogramm EXPORT 9, 14  
  Export-API 14  
  für Dienstprogramm IMPORT 53  
  IMPORT (Befehl) 39  
  Import-API 53  
  Lade-API 139  
  LOAD (Befehl) 112  
Anwendungsdatensatz (PC/IXF) 287  
anyorder (Änderungswert für Dateityp) 112, 139  
Anzeiger für Satzlänge 285  
APIs  
  db2Load 139  
  db2LoadQuery 163  
  sqluexpr 14  
  sqluimpr 53  
Arbeitsblatt  
  Dateiformat (WSF) 329  
ASC (Importdateityp) 39  
ASC-Datei  
  Beispiel 282  
  Format 281  
ASC-Datentypbeschreibungen 283  
ASCII-Dateiformat mit universellen Zeilenbegrenzern (ASC) 281  
ASCII-Format ohne universelle Zeilenbegrenzer (DEL) 276  
  Daten zwischen Plattformen versetzen 233  
Aufrufen  
  Hilfe für Befehle 360  
  Hilfe für Nachrichten 360  
  Hilfe zu SQL-Anweisungen 361  
Ausnahmetabellen  
  LOAD (Dienstprogramm) 183

## B

Beendigung  
  Ladeoperationen  
    in partitionierten Datenbanken 211  
    Modus "Lesezugriff zulassen" 109  
Beendigungsnachrichten 337

Beendigungssatz (PC/IXF) 287  
Befehle  
  db2move 237  
  db2relocatedb (Befehl) 242  
  EXPORT 9  
  IMPORT 39  
  LOAD 112  
  LOAD QUERY (Befehl) 137  
Befehlssyntax  
  interpretieren 259  
Begrenzer für Zeichenfolgen 278  
Beispiele  
  Dateien  
    ASC 282  
    DEL 278  
Benutzerdefinierte Datentypen (UDTs)  
  einzigartige Typen  
  Importieren 37  
Bestellen von DB2-Büchern 358  
binarynumerics (Änderungswert für Dateityp) 112, 139  
Bindedateien  
  durch EXPORT, IMPORT und LOAD verwendete 335

## C

chardel (Änderungswert für Dateityp)  
  Export 9, 14  
  Import 39, 53  
  Laden 112, 139  
codepage (Änderungswert für Dateityp) 112, 139  
Codepages  
  EXPORT (Befehl) 9  
  Export-API 14  
  Hinweise für Dienstprogramm IMPORT 76  
  Hinweise für Dienstprogramm LOAD 189  
  IMPORT (Befehl) 39  
  Import-API 53  
  konvertieren  
    beim Importieren oder Laden von PC/IXF-Daten 315  
    Dateien 315  
coldel (Änderungswert für Dateityp)  
  Export 9, 14  
  Import 39, 53  
  Laden 112, 139  
compound (Änderungswert für Dateityp) 39, 53  
CURSOR (Dateityp)  
  Versetzen von Daten 257

## D

Data Links Manager  
  Daten versetzen 230

Data Warehouse-Zentrale  
  Daten versetzen 255  
  Übersicht 255  
dateformat (Änderungswert für Dateityp) 39, 53, 112, 139  
Datei mit Angaben zur Speicherposition der Ladekopie bei aktualisierender Wiederherstellung 110  
Dateiformate  
  ASC (ASCII-Format mit universellen Zeilenbegrenzern) 281  
  Datei in Tabelle importieren 39  
  DEL (ASCII-Format ohne universelle Zeilenbegrenzer) 276  
  PC/IXF (PC-Version von IXF) 285  
  Tabelle in Datei exportieren 9  
  WSF (Arbeitsblattformat) 329  
Daten  
  zwischen Plattformen versetzen 233  
Daten laden  
  Partitionen 224  
Datenbanken  
  Datei in Tabelle importieren 39, 53  
  Datei in Tabelle laden 112  
  Ladeoptionen für nicht wiederherstellbare Datenbanken 82  
  Ladeoptionen für wiederherstellbare Datenbanken 82  
  Tabelle in Datei exportieren 9, 14  
  Warehouse 255  
Datensatz (PC/IXF) 287  
Datentypbeschreibungen  
  ASC 283  
  DEL 279  
  PC/IXF 311  
Datentypen  
  PC/IXF 304  
Datentypen für große Objekte (LOBs)  
  Exportieren 5  
  Importieren 36  
Datenübertragung  
  zwischen Host und Workstation 235  
  zwischen Plattformen 233  
datesiso (Änderungswert für Dateityp) 9, 14, 39, 53, 112, 139  
DB2-Bücher  
  PDF-Dateien drucken 357  
DB2 Data Links Manager  
  Dienstprogramm EXPORT 227  
  Dienstprogramm IMPORT 231  
  Dienstprogramm LOAD 232  
  zwischen Exemplaren exportieren 227  
DB2 Information - Unterstützung 340  
  aufrufen 349  
  installieren 341, 344, 347  
DB2-Lernprogramme 361  
db2Load (API) 139  
db2LoadQuery (API) 163  
DB2LOADREC (Registrierdatenbank-Variante) 110

db2move (Befehl) 237  
 db2relocatedb (Befehl) 242  
 decplusblank (Änderungswert für Dateityp) 9, 14, 39, 53, 112, 139  
 decpt (Änderungswert für Dateityp) 9, 14, 39, 53, 112, 139  
 DEL-Datei  
   Beispiel 278  
   Format 276  
 DEL-Datentypbeschreibungen 279  
 delprioritychar (Änderungswert für Dateityp) 39, 53, 112, 139  
 Dienstprogramme  
   Dateiformate 275  
 Direktaufrufe über die Tastatur  
   Unterstützung 363  
 dldel (Änderungswert für Dateityp) 9, 14, 39, 53, 112, 139  
 Dokumentation  
   anzeigen 349  
 Drucken  
   PDF-Dateien 357  
 dumpfile (Änderungswert für Dateityp) 112, 139

## E

Erneutes Starten einer Ladeoperation  
   Ladeoperationen bei partitionierten Datenbanken 211  
   Modus "Lesezugriff zulassen" 109  
 Erstellen von Indizes 96  
 EXPORT (Befehl) 9  
 EXPORT (Dienstprogramm)  
   Dateiformate 275  
   Daten zwischen Host und Workstation übertragen 235  
   DB2 Data Links Manager 227  
   Einschränkungen 3  
   exportierte Tabelle erneut erstellen 4  
   große Objekte (LOBS) 5  
   Identitätsspalten 4  
   paralleles Exportieren mit db2batch 6  
   Übersicht 1  
   zur Verwendung erforderliche Berechtigungen und Zugriffsrechte 3  
 Export-API 14  
 EXPORT-Nachrichtendateien 1, 27, 82  
 Exportieren  
   Änderungswerte für Dateityp 9, 14  
   Dateien mit Datenbanktabellen 9, 14  
   Hinweise zu DB2 Data Links Manager 9  
   Spaltennamen angeben 14  
 Exportierte Tabellen  
   mit Dienstprogramm EXPORT erneut erstellen 4  
   mit Dienstprogramm IMPORT erneut erstellen 35  
   mit in IXF-Datei gespeicherten Tabellenattributen erneut erstellen 35  
   ohne in IXF-Datei gespeicherte Tabellenattribute erneut erstellen 35

Exportoperationen, Data Warehouse-Zentrale 255

## F

fastparse (Änderungswert für Dateityp) 112, 139  
 Fehlerbehebung  
   Lernprogramme 362  
   Onlineinformationen 362  
 Fehlerbestimmung  
   Lernprogramme 362  
   Onlineinformationen 362  
 Fehlernachrichten  
   Übersicht 337  
 forcein (Änderungswert für Dateityp) 39, 53, 112, 139, 320  
 Fortsetzungssatztyp PC/IXF 287

## G

Gedruckte Bücher bestellen 358  
 generatedignore (Änderungswert für Dateityp) 39, 53, 112, 139  
 generatedmissing (Änderungswert für Dateityp) 39, 53, 112, 139  
 generatedoverride (Änderungswert für Dateityp) 112, 139  
 Generierte Spalten  
   beim Dienstprogramm IMPORT 34  
   beim Dienstprogramm LOAD 100  
 Gepufferte INSERT-Operationen  
   Dienstprogramm IMPORT 31  
 Gespeicherte Abfragetabellen (MQT)  
   abhängige sofort gespeicherte 105  
   aktualisieren 105  
   Status "Überprüfung anstehend" 105  
 Gespeicherte Prozeduren  
   Umsetzungsprogramm 255  
 Gültige PC/IXF-Datentypen 304

## H

Hierarchiesatz (PC/IXF) 287  
 Hilfe  
   anzeigen 349, 351  
   für Befehle  
     aufrufen 360  
   für Nachrichten  
     aufrufen 360  
   für SQL-Anweisungen  
     aufrufen 361  
 Hilfe für Befehle  
   aufrufen 360  
 Hilfe für Nachrichten  
   aufrufen 360  
 Hilfe zu SQL-Anweisungen  
   aufrufen 361  
 HTML-Dokumentation  
   aktualisieren 350

## I

IBM Replikationstools für relationale Daten  
   Komponenten 254  
   Übersicht 253  
 Identitätssatz (PC/IXF) 287  
 Identitätsspalten 4  
   beim Dienstprogramm IMPORT 32  
   beim Dienstprogramm LOAD 98  
 identityignore (Änderungswert für Dateityp) 39, 53, 112, 139  
 identitymissing (Änderungswert für Dateityp) 39, 53, 112, 139  
 identityoverride (Änderungswert für Dateityp) 112, 139  
 implieddecimal (Änderungswert für Dateityp) 39, 53, 112, 139  
 IMPORT (Befehl) 39  
 IMPORT (Dienstprogramm)  
   benutzerdefinierte einzigartige Datentypen (UDTs) 37  
   Berechtigungen 29  
   Client/Server 30  
   Dateiformate 275  
   Daten zwischen Host und Workstation übertragen 235  
   DB2 Data Links Manager 231  
   Einschränkungen 29  
   exportierte Tabelle erneut erstellen 35  
   ferne Datenbank 30  
   generierte Spalten 34  
   gepufferte INSERT-Operationen 31  
   große Objekte (LOBS) 36  
   Hinweise zu Codepages 76  
   Identitätsspalten 32  
   im Vergleich zum Dienstprogramm LOAD 263  
   Leistung 27  
   Leistung optimieren 27  
   Tabellen sperren 37  
   Übersicht 27  
   Zugriffsrechte 29  
 Import-API 53  
 IMPORT-Nachrichtendateien 1, 27, 82  
 Importieren  
   allgemeine Regeln für PC/IXF-Dateien 315  
   Änderungswerte für Dateityp 53  
   Datei in Datenbanktabelle 53  
   Daten 39  
   Datenbankzugriff über DB2 Connect 53  
   datentypspezifische Regeln für PC/IXF-Dateien 317  
   Einschränkungen 53  
   Hinweise zu Codepages 53  
   Hinweise zu DB2 Data Links Manager 53  
   in ferne Datenbank 53  
   in nicht vorhandene Tabelle oder Hierarchie 53  
   keine typisierten Tabellen 53  
   mehnteilige PC/IXF-Dateien 53  
   von PC/IXF-Dateien mit FORCEIN 320

indexfreespace (Änderungswert für Dateityp) 112, 139  
indexixf (Änderungswert für Dateityp) 39, 53  
Indexsatz (PC/IXF) 287  
indexschema (Änderungswert für Dateityp) 39, 53  
Indizes  
  erstellen 96  
Inkompatible Spalten 315  
Installieren  
  DB2 Information - Unterstützung 341, 344, 347  
Integration Exchange Format (IXF) 285  
Integritätsbedingungen  
  überprüfen  
  nach Ladeoperationen 102  
Integritätsprüfung 102

## K

keepblanks (Änderungswert für Dateityp) 39, 53, 112, 139  
Kopfdatensatz (PC/IXF) 287

## L

Lade-API 139  
Laden  
  Änderungswerte für Dateityp 112  
  Datei in Datenbanktabelle 112  
  Daten  
  Partitionen 201  
Ladeoperationen, Data Warehouse-Zentrale 255  
Leistung  
  Dienstprogramm LOAD 191  
  Importieren 27  
Lernprogramme 361  
  Fehlerbehebung und Fehlerbestimmung 362  
LOAD (Befehl) 112  
  in einer Umgebung mit partitionierten Datenbanken 203, 222  
LOAD (Dienstprogramm)  
  Änderungswerte für Dateityp 139  
  Ausnahmetabelle 183  
  BUILD-Phase 82  
  Dateiformate 275  
  Datenbankwiederherstellung 82  
  DB2 Data Links Manager 232  
  DELETE-Phase 82  
  Einschränkungen 91  
  geänderte Syntax und Funktionsweise 82  
  generierte Spalten 100  
  Hinweise zu Codepages 189  
  Identitätsspalten 98  
  im Vergleich zum Dienstprogramm  
  IMPORT 263  
  INDEX COPY-Phase 82  
  Leistung optimieren 191  
  LOAD-Phase 82  
  Parallelität 89  
  Protokollsätze 185  
  Speicherausgangsdatei 184

LOAD (Dienstprogramm) (Forts.)  
  Tabellen sperren 186  
  Tabellenstatus 186  
  temporäre Dateien 112, 184  
  Übersicht 82  
  Wiederherstellung über Prozess 82  
  Wiederherstellung nach Fehler 109  
  zur Verwendung erforderliche Berechtigungen und Zugriffsrechte 90  
LOAD DELETE START COMPENSATION (Protokollsatz) 185  
LOAD-Nachrichtendateien 1, 27, 82  
LOAD PENDING LIST (Protokollsatz) 185  
LOAD QUERY (Befehl) 137  
  in einer Umgebung mit partitionierten Datenbanken 209  
LOAD START (Protokollsatz) 185  
LOB (großes Objekt), Datentypen  
  Exportieren 5  
  Importieren 36  
LOB-Positionskennung 285  
lobsinfile (Änderungswert für Dateityp) 9, 39, 53, 112, 139  
  Export-API 14

## M

Mehrdimensionales Clustering (MDC)  
  Hinweise 108

## N

Nachrichten  
  Übersicht 337  
Nachrichtendateien  
  EXPORT, IMPORT und LOAD 1, 27, 82  
Nicht als Identitätsspalten generierte Spalten 34, 100  
Nicht wiederherstellbare Datenbank  
  Ladeoptionen 82  
nochecklengths (Änderungswert für Dateityp) 39, 53, 112, 139  
nodefaults (Änderungswert für Dateityp) 39, 53  
nodoubleled (Änderungswert für Dateityp) 9, 14, 39, 53, 112, 139  
noeofchar (Änderungswert für Dateityp) 39, 53, 112, 139  
noheader (Änderungswert für Dateityp) 112, 139  
norowwarnings (Änderungswert für Dateityp) 112, 139  
notypeid (Änderungswert für Dateityp) 39, 53  
nullindchar (Änderungswert für Dateityp) 39, 53, 112, 139

## O

Online  
  Zugriff auf Hilfe 359  
Optionen  
  FORCEIN 320

## P

packeddecimal (Änderungswert für Dateityp) 112, 139  
pagefreespace (Änderungswert für Dateityp) 112, 139  
Paralleles Exportieren mit db2batch 6  
Parallelität  
  Dienstprogramm LOAD 89  
Parameter  
  Syntax 259  
Partitionierte Datenbanken  
  Einschränkungen für Laden 203  
Partitionierung von Daten  
  Daten laden 201  
Partitionierungsschlüssel  
  Daten laden 201  
PC/IXF  
  Codepagekonvertierungsdateien 315  
  Datentypen 311  
  gültige 304  
  im Gegensatz zu System370 IXF 328  
  Satztypen 287  
  Spaltenwerte, ungültige 315  
  ungültig  
  Datentypen 304, 315  
  Spaltenwerte 315  
PC/IXF-Datei importieren  
  datentypspezifische Regeln 317  
  mit FORCEIN 320  
  Regeln 315, 317  
PC/IXF-Dateiformat  
  Beschreibung 285  
  Daten zwischen Plattformen versetzen 233  
PC-Version des IXF-Dateiformats (PC/IXF) 285  
Protokollsätze  
  Dienstprogramm LOAD 185

## R

reclen (Änderungswert für Dateityp) 39  
  Importieren 53  
  Lade-API 139  
  Laden 112  
Registrierdatenbank-Variablen  
  DB2LOADREC 110  
Replikation  
  Data Warehouse-Zentrale  
  unterstützte Typen 255

## S

Satz für untergeordnete Tabelle (PC/IXF)  
  Übersicht 287  
Satzlängenanzeiger 285  
Satztyp (PC/IXF)  
  Anwendungssatz 287  
  Beendigungssatz 287  
  Daten 287  
  Fortsetzungssatz 287  
  Hierarchiesatz 287  
  Identitätssatz 287  
  Indexsatz 287  
  Kopfdatensatz 287  
  Liste 285

Satztyp (PC/IXF) (Forts.)  
 Satz für untergeordnete Tabelle 287  
 Spaltendeskriptorsatz 287  
 Tabellensatz 287  
 Schlüsselwörter  
 Syntax 259  
 SELECT-Anweisung  
 im EXPORT-Befehl 9  
 Semantik  
 FORCEIN, allgemein 320  
 FORCEIN, Codepage 320  
 FORCEIN, Datentyp 320  
 Spalten  
 inkompatible 315  
 Werte, ungültige 315  
 zum Importieren angeben 53  
 Spaltendeskriptorsatz (PC/IXF) 287  
 Speicherauszugsdateien  
 Dienstprogramm LOAD 184  
 Sperren  
 Dienstprogramm IMPORT 37  
 Tabellenebene 186  
 SQL-Nachrichten 337  
 SQLCODE  
 Übersicht 337  
 SQLSTATE  
 Übersicht 337  
 sqluexpr (API) 14  
 sqluimpr (API) 53  
 Statusangaben  
 Laden anstehend 190  
 Löschen anstehend 190  
 Sicherung anstehend 190  
 Überprüfung anstehend 190  
 Statusangaben für anstehende Aktionen 190  
 striptblanks (Änderungswert für Dateityp) 39, 53, 112, 139  
 striptnulls (Änderungswert für Dateityp) 39, 53, 112, 139  
 Struktur  
 ASC-Dateien (ASCII-Format mit universellen Zeilenbegrenzern) 281  
 DEL-Dateien (ASCII-Format ohne universelle Zeilenbegrenzer) 276  
 subtableconvert (Änderungswert für Dateityp) 112  
 Syntax  
 Änderungen, Dienstprogramm LOAD 82  
 Syntaxdiagramm  
 lesen 259  
 Syntaxdiagramme (Schreibweise mit Trennzeichen) 365  
 System370 IXF  
 im Gegensatz zu PC/IXF 328  
 im Gegensatz zu System370 328

## T

Tabellen  
 Dateien importieren 39, 53  
 Dateien laden in 112  
 exportierte Tabellen erneut erstellen 35  
 in Dateien exportieren 9, 14  
 Sperren 186

Tabellen (Forts.)  
 Statusangaben 186  
 Tabellenbereiche  
 Statusangaben 186  
 Tabellensatz (PC/IXF) 287  
 TABLE LOAD DELETE START (Protokollsatz) 185  
 Temporäre Dateien  
 Dienstprogramm LOAD 184  
 LOAD (Befehl) 112  
 timeformat (Änderungswert für Dateityp) 39, 53, 112, 139  
 timestampformat (Änderungswert für Dateityp) 39, 53, 112, 139  
 Tool zum Versetzen von Daten, Befehl 237  
 totalreespace (Änderungswert für Dateityp) 112, 139  
 Traversierfolge  
 benutzerdefiniert 249  
 Standard 249  
 typisierte Tabellen 27, 249  
 Typisierte Tabellen  
 beim Versetzen von Daten auswählen 251  
 Beispiele für das Versetzen von Daten 251  
 Daten versetzen zwischen 248  
 Exportieren 248  
 Importieren 248  
 Traversierfolge 27, 249

## U

Übersichtstabellen  
 Importbeschränkung 29  
 Umgebungen mit partitionierten Datenbanken  
 Daten laden 222, 224  
 Ladeoperationen überwachen 209  
 Umsetzungsprogramme  
 gespeicherte Prozeduren 255  
 Unicode (UCS-2)  
 Hinweise für das Versetzen von Daten 331  
 Unverfügbarkeit 363  
 usedefaults (Änderungswert für Dateityp) 39, 53, 112, 139

## V

Variablen  
 Syntax 259  
 Verlagern von Datenbanken, Befehl 242  
 Versetzen von Daten  
 zwischen Datenbanken 39, 53

## W

Warnungen  
 Übersicht 337  
 Wiederherstellbare Datenbanken  
 Ladeoptionen 82  
 WSF-Dateiformat (Arbeitsblattformat)  
 Beschreibung 329

WSF-Dateiformat (Arbeitsblattformat) (Forts.)  
 Daten zwischen Plattformen versetzen 233

## Z

Zeichenfolgen  
 Begrenzer 278  
 zoned decimal (Änderungswert für Dateityp) 112, 139  
 Zugriffsmöglichkeit  
 Funktionen 363  
 Syntaxdiagramme (Schreibweise mit Trennzeichen) 365  
 Zugriffsrechte  
 Export 3  
 Import 29  
 LOAD 90  
 Zwischenspeichertabellen  
 abhängige sofort gespeicherte 106  
 weitergeben 106

---

## Kontaktaufnahme mit IBM

Telefonische Unterstützung erhalten Sie über folgende Nummern:

- Unter 0180 3 313233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0190 7 72243 erreichen Sie die DB2 Helpline, wo Sie Antworten zu DB2-spezifischen Problemen erhalten.

Informationen zur nächsten IBM Niederlassung in Ihrem Land oder Ihrer Region finden Sie im IBM Verzeichnis für weltweite Kontakte, das Sie im Web unter <http://www.ibm.com/planetwide> abrufen können.

---

## Produktinformationen

Informationen zu DB2 Universal Database-Produkten erhalten Sie telefonisch oder im World Wide Web unter <http://www.ibm.com/software/data/db2/udb>.

Diese Site enthält die neuesten Informationen zur technischen Bibliothek, zum Bestellen von Büchern, zu Produktdownloads, Newsgroups, FixPaks, Neuerungen und Links auf verfügbare Webressourcen.

Telefonische Unterstützung erhalten Sie über folgende Nummern:

- Unter 0180 3 313233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0180 5 5090 können Sie Handbücher telefonisch bestellen.

Informationen dazu, wie Sie sich mit IBM in Verbindung setzen können, finden Sie auf der globalen IBM Internet-Seite unter folgender Adresse:  
[www.ibm.com/planetwide](http://www.ibm.com/planetwide)







SC12-3055-01

