

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen in Anhang F, „Bemerkungen“, auf Seite 529 gelesen werden.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM DB2 for Linux, UNIX, and Windows, Version 9 Release 7, Data Movement Utilities Guide and Reference,
IBM Form SC27-2440-01,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 1993, 2012

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:
TSC Germany
Kst. 2877
Mai 2012

Inhaltsverzeichnis

Inhalt des Handbuchs. v

Dienstprogramme und Referenzinformationen für das Versetzen von Daten . . . 1

Optionen beim Versetzen von Daten	1
Dienstprogramm EXPORT	6
Dienstprogramm EXPORT - Übersicht	6
Für das Dienstprogramm EXPORT erforderliche Zugriffsrechte und Berechtigungen	7
Exportieren von Daten	8
Referenz - Export	18
Dienstprogramm IMPORT	49
Dienstprogramm IMPORT - Übersicht	49
Für das Dienstprogramm IMPORT erforderliche Zugriffsrechte und Berechtigungen	52
Importieren von Daten	53
Weitere Aspekte des Imports	68
Referenz - Import	70
Dienstprogramm LOAD	141
LOAD - Übersicht	141
Für das Dienstprogramm LOAD erforderliche Zugriffsrechte und Berechtigungen	145
Laden von Daten	146
Weitere Aspekte des Ladens	176
Ladefunktionen zur Verwaltung der referenziellen Integrität	194
Fehlgeschlagener oder unvollständiger Ladevorgang	210
Übersicht zum Laden - Umgebungen mit partitionierten Datenbanken	216
Referenz - Laden	239
Weitere Optionen beim Versetzen von Daten	375
Versetzen von Tabellen im Online-Modus mithilfe der Prozedur ADMIN_MOVE_TABLE	375
Versetzen von Daten mit DB2 Connect	379
IBM Replikationstools nach Komponenten geordnet	381
Kopieren von Schemata	382
Ausführen eines umgeleiteten Restores mithilfe eines automatisch generierten Scripts	396
Hohe Verfügbarkeit durch Unterstützung der ausgesetzten E/A und der Onlineteilung einer Spiegeldatenbank	418
db2relocatedb - Verlagern einer Datenbank	421
db2look - DB2-Statistiktool und DDL-Extraktionstool	427
Dateiformate und Datentypen	440

Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD	440
Aspekte von Unicode beim Versetzen von Daten	498
Zeichensatz und NLS.	500
Versetzen von XML-Daten	501

Anhang A. Unterschiede zwischen den Dienstprogrammen IMPORT und LOAD. 507

Anhang B. Von den Dienstprogrammen EXPORT, IMPORT und LOAD verwendete Bindedateien 509

Anhang C. Lesen von Syntaxdiagrammen 511

Anhang D. Erfassen von Daten für Probleme beim Versetzen von Daten . 515

Anhang E. Übersicht über die technischen Informationen zu DB2. 517

Bibliothek mit technischen Informationen zu DB2 im Hardcopy- oder PDF-Format	518
Bestellen gedruckter DB2-Bücher	521
Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor	522
Zugriff auf verschiedene Versionen der DB2-Informationenzzentrale	522
Anzeigen von Themen in der gewünschten Sprache in der DB2-Informationenzzentrale	522
Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten DB2-Informationenzzentrale	523
Manuelles Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten DB2-Informationenzzentrale	525
DB2-Lernprogramme	527
Informationen zur Fehlerbehebung in DB2	527
Bedingungen	528

Anhang F. Bemerkungen 529

Index 533

Inhalt des Handbuchs

Dieses Handbuch bietet Informationen zur Verwendung der folgenden DB2 Database for Linux, UNIX and Windows-Dienstprogramme für das Versetzen von Daten:

- EXPORT und IMPORT

Die Dienstprogramme EXPORT und IMPORT versetzen Daten aus Tabellen oder Sichten in andere Datenbanken oder Tabellenkalkulationsprogramme, zwischen DB2-Datenbanken sowie zwischen DB2-Datenbanken und Hostdatenbanken mithilfe von DB2 Connect. 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.

- LOAD

Mit dem Dienstprogramm LOAD können Daten in Tabellen versetzt, vorhandene Indizes erweitert und Statistikdaten generiert werden. Das Dienstprogramm LOAD kann große Datenmengen schneller versetzen als das Dienstprogramm IMPORT. Daten, die mit dem Dienstprogramm EXPORT exportiert wurden, können mit dem Dienstprogramm LOAD geladen werden.

Wird das Dienstprogramm LOAD in einer Umgebung mit partitionierten Datenbanken eingesetzt, können große Datenmengen verteilt und in unterschiedliche Datenbankpartitionen geladen werden.

Eine vollständige Auflistung der Optionen zum Versetzen von Daten finden Sie im Abschnitt „Optionen beim Versetzen von Daten“ auf Seite 1.

Dienstprogramme und Referenzinformationen für das Versetzen von Daten

Optionen beim Versetzen von Daten

Es stehen verschiedene Optionen für das Versetzen von Daten zur Verfügung. Die folgende Tabelle enthält eine Übersicht über die verfügbaren Tools und Dienstprogramme für das Versetzen von Daten. Verwenden Sie diese Tabelle als Richtlinie bei der Ermittlung der für Ihre Anforderungen am besten geeigneten Optionen für das Versetzen von Daten.

Tabelle 1. Optionen beim Versetzen von Daten

Dienstprogrammname	Dienstprogramm LOAD
Zweck	Mit dem Dienstprogramm LOAD können große Datenmengen effizient in neu erstellte Tabellen oder in Tabellen, die bereits Daten enthalten, versetzt werden.
Plattformübergreifend kompatibel	Ja
Empfohlene Methode für die Verwendung	Dieses Dienstprogramm eignet sich am besten für Situationen, in denen die Leistung im Mittelpunkt steht. Dieses Dienstprogramm kann als Alternative zum Dienstprogramm IMPORT verwendet werden. Es arbeitet schneller als das Dienstprogramm IMPORT, da es formatierte Seiten direkt in die Datenbank schreibt, anstatt SQL-Anweisungen INSERT zu verwenden. Darüber hinaus bietet das Dienstprogramm LOAD die Möglichkeit, die Daten nicht zu protokollieren oder mit der Option COPY eine Kopie der geladenen Daten zu speichern. Ladeoperationen können die Ressourcen, wie beispielsweise CPUs und Hauptspeicherkapazitäten in SMP- und MPP-Umgebungen in vollem Umfang nutzen.
Referenzinformationen	Laden von Daten

Dienstprogrammname	db2move
Zweck	Wenn Sie das Dienstprogramm db2move mit der Option COPY verwenden, können Sie Schemaschablonen (mit oder ohne Daten) von einer Quelldatenbank in eine Zieldatenbank kopieren oder ein vollständiges Schema von einer Quelldatenbank in eine Zieldatenbank versetzen. Die Verwendung des Dienstprogramms db2move mit der Option IMPORT oder EXPORT vereinfacht das Versetzen einer großen Anzahl von Tabellen zwischen DB2-Datenbanken.
Plattformübergreifend kompatibel	Ja

Dienstprogrammname	db2move
Empfohlene Methode für die Verwendung	Wenn die Option COPY verwendet wird, müssen die Quellen- und die Zieltabelle unterschiedlich sein. Die Option COPY ist bei der Erstellung von Schemaschablonen nützlich. Verwenden Sie die Option IMPORT oder EXPORT für das Klonen von Datenbanken, wenn keine Unterstützung für plattformübergreifende Backup- und Restoreoperationen zur Verfügung steht. Die Optionen IMPORT oder EXPORT werden zusammen mit dem Befehl db2look verwendet.
Referenzinformationen	<ul style="list-style-type: none"> • „Kopieren eines Schemas“ in <i>Datenbankverwaltung - Konzepte und Konfiguration - Referenzinformationen</i> • Neuerstellung von importierten Tabellen

Dienstprogrammname	Dienstprogramm IMPORT
Zweck	Das Dienstprogramm IMPORT fügt Daten aus einer externen Datei mit einem unterstützten Dateiformat in eine Tabelle, eine Hierarchie, eine Sicht oder einen Kurznamen ein.
Plattformübergreifend kompatibel	Ja
Empfohlene Methode für die Verwendung	<p>Das Dienstprogramm IMPORT kann in den folgenden Situationen eine geeignete Alternative zum Dienstprogramm LOAD darstellen:</p> <ul style="list-style-type: none"> • Wenn es sich bei der Zieltabelle um eine Sicht handelt. • Wenn die Zieltabelle über Integritätsbedingungen verfügt und Sie nicht möchten, dass die Zieltabelle in den Status "Festlegen der Integrität anstehend" versetzt wird. • Wenn die Zieltabelle über Trigger verfügt und Sie möchten, dass diese ausgelöst werden.
Referenzinformationen	Importieren von Daten

Dienstprogrammname	Dienstprogramm EXPORT
Zweck	Das Dienstprogramm EXPORT exportiert Daten aus einer Datenbank in eines von mehreren externen Dateiformaten. Die Daten können dann zu einem späteren Zeitpunkt importiert oder geladen werden.
Plattformübergreifend kompatibel	Ja

Dienstprogrammname	Dienstprogramm EXPORT
Empfohlene Methode für die Verwendung	Dieses Dienstprogramm eignet sich am besten für Situationen, in denen Daten in einer externen Datei gespeichert werden sollen, um entweder weiter verarbeitet oder in eine andere Tabelle versetzt zu werden. High Performance Unload (HPU) ist eine Alternative hierzu, muss jedoch separat bezogen werden. Das Dienstprogramm EXPORT unterstützt XML-Spalten.
Referenzinformationen	Exportieren von Daten

Dienstprogrammname	Prozedur ADMIN_COPY_SCHEMA
Zweck	Diese Prozedur ermöglicht das Erstellen einer Kopie für alle Objekte in einem einzelnen Schema und das erneute Erstellen dieser Objekte in einem neuen Schema. Diese Kopieroperation kann innerhalb einer Datenbank mit oder ohne Daten ausgeführt werden.
Plattformübergreifend kompatibel	Ja
Empfohlene Methode für die Verwendung	<p>Dieses Dienstprogramm ist für die Erstellung von Schemaschablonen geeignet. Es ist darüber hinaus nützlich, wenn Sie mit einem Schema experimentieren möchten (z. B. neue Indizes testen), ohne dass sich dies auf die Funktionsweise des Quellschemas auswirkt. Die Hauptunterschiede zwischen der Prozedur ADMIN_COPY_SCHEMA und dem Dienstprogramm db2move sind:</p> <ul style="list-style-type: none"> • Die Prozedur ADMIN_COPY_SCHEMA wird für eine einzelne Datenbank verwendet, das Dienstprogramm db2move dagegen datenbankübergreifend. • Das Dienstprogramm db2move schlägt beim Aufruf fehl, wenn es kein physisches Objekt, wie z. B. eine Tabelle oder einen Index, erstellen kann. Die Prozedur ADMIN_COPY_SCHEMA protokolliert die Fehler und setzt die Verarbeitung fort. • Die Prozedur ADMIN_COPY_SCHEMA verwendet die Option "vom Cursor laden", um Daten von einem Schema in ein anderes zu versetzen. Das Dienstprogramm db2move verwendet eine ferne Ladeoperation, ähnlich der Option "vom Cursor laden", die die Daten aus der Quelldatenbank extrahiert.
Referenzinformationen	<ul style="list-style-type: none"> • Versetzen von Tabellen im Online-Modus mithilfe der Prozedur ADMIN_MOVE_TABLE • „Kopieren eines Schemas“ in <i>Datenbankverwaltung - Konzepte und Konfiguration - Referenzinformationen</i>

Dienstprogrammname	Prozedur ADMIN_MOVE_TABLE
Zweck	Mit dieser Prozedur können Daten in einer Tabelle in ein neues Tabellenobjekt desselben Namens (möglicherweise aber mit anderen Speicherkenndaten) versetzt werden, während die Daten online und somit zugänglich sind.
Plattformübergreifend kompatibel	Ja
Empfohlene Methode für die Verwendung	<p>Dieses Dienstprogramm automatisiert das Versetzen von Tabellendaten in ein neues Tabellenobjekt. Während der Prozedur verbleiben die Daten online und können ausgewählt, eingefügt, aktualisiert und gelöscht werden. Beim Versetzen einer Tabelle können Sie auch ein Komprimierungswörterverzeichnis erstellen.</p> <ul style="list-style-type: none"> • Vermeiden Sie es, verschiedene Tabellendaten gleichzeitig in denselben Tabellenbereich zu versetzen. • Führen Sie diese Prozedur aus, wenn kaum Aktivitäten für die Tabelle vorhanden sind. • Führen Sie eine Operation mit mehreren Schritten aus. Die Phasen INIT und COPY können jederzeit aufgerufen werden. Führen Sie die Phase REPLAY mehrmals aus, um die Größe der Staging-Tabelle gering zu halten. Setzen Sie dann während einer Zeit mit wenig Aktivität für die Tabelle den Befehl SWAP ab. • Ziehen Sie in Betracht, eine Tabelle offline zu versetzen, wenn Sie mit Tabellen ohne eindeutigen Index oder Tabellen ohne Index arbeiten.
Referenzinformationen	„ADMIN_MOVE_TABLE (Prozedur) - Onlinetabelle versetzen“ in der Veröffentlichung <i>Command Reference</i>

Dienstprogrammname	Restoredienstprogramm mit der Option REDIRECT und der Option GENERATE SCRIPT
Zweck	Mit dem Restoredienstprogramm kann eine gesamte Datenbank mithilfe eines Scripts aus einem vorhandenen Backup-Image von einem System auf ein anderes kopiert werden.
Plattformübergreifend kompatibel	Eingeschränkt. Siehe Referenzinformationen.
Empfohlene Methode für die Verwendung	Dieses Dienstprogramm eignet sich am besten für Situationen, in denen ein Backup-Image vorhanden ist.

Dienstprogrammname	Restoredienstprogramm mit der Option REDIRECT und der Option GENERATE SCRIPT
Referenzinformationen	<ul style="list-style-type: none"> • „Ausführen eines umgeleiteten Restores mithilfe eines automatisch generierten Scripts“ in <i>Datenrecovery und hohe Verfügbarkeit - Handbuch und Referenz</i> • „Backup- und Restoreoperationen zwischen unterschiedlichen Betriebssystemen und Hardwareplattformen“ in <i>Datenrecovery und hohe Verfügbarkeit - Handbuch und Referenz</i>

Dienstprogrammname	Befehl "db2relocatedb" zum Verlagern einer Datenbank
Zweck	Mit diesem Befehl können Sie eine Datenbank umbenennen oder eine Datenbank bzw. einen Teil einer Datenbank innerhalb desselben Systems oder auf ein anderes System verlagern.
Plattformübergreifend kompatibel	Nein
Empfohlene Methode für die Verwendung	<ul style="list-style-type: none"> • Dieses Dienstprogramm kann in Situationen verwendet werden, in denen Backup und Restore zu zeitaufwändig sind. • Dieses Dienstprogramm stellt eine Alternative zur Verwendung von Backup und Restore für das Versetzen von Datenbanken bzw. das Erstellen von Datenbankkopien dar. • Darüber hinaus bietet es eine zeitsparende Methode für das Klonen einer Datenbank für alternative Umgebungen, wie z. B. Testumgebungen. • Es kann für das Versetzen von Tabellenbereichscontainern in eine neue Gruppe von Speichereinheiten verwendet werden.
Referenzinformationen	„db2relocatedb - Relocate database command“ (Befehl zum Verlagern einer Datenbank) im Handbuch <i>Command Reference</i>

Dienstprogrammname	Geteilte Spiegeldatenbank
Zweck	Erstellung einer Klon-, Bereitschafts- oder Backup-Datenbank.
Plattformübergreifend kompatibel	Nein

Dienstprogrammname	Geteilte Spiegeldatenbank
Empfohlene Methode für die Verwendung	<ul style="list-style-type: none"> • Erstellen eines Bereitschaftssystems, um bei einem Ausfall des Primärsystems die Ausfallzeit zu reduzieren. • Verlagern von Backup-Operationen von einer Maschine mit laufender Produktion hin zu einer geteilten Datenbank. • Bereitstellen einer zeitsparenden Methode für das Klonen einer Datenbank für alternative Umgebungen, wie z. B. Testumgebungen.
Hinweise	<ul style="list-style-type: none"> • Nur für DMS-Tabellenbereiche kann auf der geteilten Version der Datenbank ein Backup erstellt werden. • Dieses Dienstprogramm wird normalerweise zusammen mit einer Flashcopytechnologie verwendet, die von Speichersystemen bereitgestellt wird. • Eine Alternative besteht darin, nach dem Aussetzen der Datenbank eine Dateikopieroperation auszuführen, hierdurch wird jedoch die Menge des für die Datenbank erforderlichen Speicherplatzes verdoppelt.
Referenzinformationen	„Hohe Verfügbarkeit durch Unterstützung der ausgesetzten E/A und der Onlineteilung einer Spiegeldatenbank“ in <i>Datenrecovery und hohe Verfügbarkeit - Handbuch und Referenz</i>

Dienstprogramm EXPORT

Dienstprogramm EXPORT - Übersicht

Das Dienstprogramm EXPORT extrahiert Daten mithilfe einer SQL-Anweisung SELECT oder einer XQuery-Anweisung und stellt die Informationen in eine Datei. Die Ausgabedatei können Sie zum Versetzen von Daten für eine zukünftige Import- oder Ladeoperation oder zum Bereitstellen von Daten für Analysezwecke verwenden.

Das Dienstprogramm EXPORT stellt ein ebenso einfaches wie flexibles Dienstprogramm zum Versetzen von Daten dar. Sie können das Dienstprogramm über die Steuerzentrale, durch Ausgabe des Befehls **EXPORT** über den Befehlszeilenprozessor, durch Aufruf der gespeicherten Prozedur ADMIN_CMD oder durch Aufruf der API db2Export über eine Benutzeranwendung aktivieren.

Für eine einfache Exportoperation sind folgende Angaben erforderlich:

- Pfad und Name des Betriebssystems, in dem die zu exportierenden Daten gespeichert werden sollen
- Das Format der Daten in der Eingabedatei
Das Dienstprogramm EXPORT unterstützt die Datenformate IXF, WSF und DEL für die Ausgabedateien.
- Eine nähere Angabe der zu exportierenden Daten
Für die meisten Exportoperationen müssen Sie eine Anweisung SELECT bereit-

stellen, die die für den Export abzurufenden Daten angibt. Für den Export typischer Tabellen ist eine explizite Ausgabe der Anweisung `SELECT` nicht erforderlich. Sie müssen in diesem Fall lediglich die Traversierfolge für untergeordnete Tabellen innerhalb der Hierarchie angeben.

Zum Versetzen von Daten im Format IXF können Sie das Dienstprogramm `EXPORT` mit `DB2 Connect` verwenden.

Weitere Optionen

Sie können Exportoperationen mit einer Reihe von Parametern anpassen. Änderungswerte für den Dateityp eröffnen viele Möglichkeiten und erlauben z. B. das Ändern des Datenformats und der Codepage sowie von Datums- und Zeitmarken und das Schreiben bestimmter Datentypen in separate Dateien. Mit den Parametern für `METHOD` können Sie andere Spaltennamen für die zu exportierenden Daten angeben.

Sie können Daten aus Tabellen exportieren, die Spalten mit XML-Datentyp enthalten. Mit den Parametern `XMLFILE`, `XML TO` und `XMLSAVESCHEMA` können Sie näher bestimmen, wie die exportierten Dokumente anschließend gespeichert werden sollen.

Die Leistung des Dienstprogramms `EXPORT` kann auf verschiedene Arten optimiert werden. Da es sich bei dem Dienstprogramm `EXPORT` um eine integrierte SQL-Anwendung handelt, die SQL-Abrufe intern durchführt, profitiert das Dienstprogramm `EXPORT` auch von Optimierungen, die bei den SQL-Operationen vorgenommen werden. Sinnvoll können z. B. große Pufferpools, Indexierungen und Sortierspeicher sein. Es empfiehlt sich darüber hinaus, das Risiko von Einheitenkonkurrenzsituationen bei den Ausgabedateien zu vermeiden, indem sie getrennt von den Containern und Protokolleinheiten platziert werden.

Nachrichtendatei

Das Dienstprogramm `EXPORT` schreibt Fehler- und Informationsnachrichten sowie Warnungen in ASCII-Standardtextnachrichtendateien. Abgesehen vom Befehlszeilenprozessor müssen Sie bei allen Schnittstellen den Namen dieser Dateien zuvor mit dem Parameter `MESSAGES` angeben. Wenn Sie den Befehlszeilenprozessor verwenden und keine Nachrichtendatei angeben, schreibt das Dienstprogramm `EXPORT` die Nachrichten in die Standardausgabe.

Für das Dienstprogramm `EXPORT` erforderliche Zugriffsrechte und Berechtigungen

Zugriffsrechte ermöglichen Ihnen den Zugriff auf Datenbankressourcen sowie das Erstellen, Aktualisieren und Löschen dieser Ressourcen. Berechtigungsstufen bieten die Möglichkeit, Berechtigungen zu übergeordneten Pflege- und Dienstprogrammoperationen des Datenbankmanagers zuzuordnen.

Zugriffsrechte und Berechtigungen dienen gleichermaßen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte. Sie können nur auf solche Objekte zugreifen, für die Sie zugriffsberechtigt sind, d. h., für die Sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigung verfügen.

Sie benötigen die Berechtigung `DATAACCESS` bzw. das Zugriffsrecht `CONTROL` oder `SELECT` für jede Tabelle und Sicht, die an dem Exportvorgang beteiligt ist.

Beim Export LBAC-geschützter Daten (Label-Based Access Control, kennsatzbasierte Zugriffssteuerung) muss die Berechtigungs-ID der Sitzung zum Lesen der zu exportierenden Zeilen bzw. Spalten berechtigt sein. Geschützte Zeilen, die von der Berechtigungs-ID der Sitzung nicht gelesen werden dürfen, werden nicht ex-

portiert. Schließt die Anweisung SELECT geschützte Spalten ein, die von der Berechtigungs-ID der Sitzung nicht gelesen werden dürfen, schlägt der Export fehl, und es wird ein Fehler (SQLSTATE 42512) zurückgegeben.

Exportieren von Daten

Mit dem Dienstprogramm EXPORT können Sie Daten aus einer Datenbank in eine Datei exportieren. Die Datei kann eines von mehreren externen Dateiformaten aufweisen. Sie können die zu exportierenden Daten durch eine SQL-Anweisung SELECT oder über hierarchische Informationen für typisierte Tabellen angeben.

Vorbereitende Schritte

Zum Exportieren von Daten aus einer Datenbank müssen Sie über die Berechtigung DATAACCESS, das Zugriffsrecht CONTROL oder das Zugriffsrecht SELECT für jede einzelne am Export beteiligte Tabelle oder Sicht verfügen.

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 zu der gewünschten Datenbank herzustellen. Wenn das implizite Herstellen von Verbindungen aktiviert ist, wird eine Verbindung zur Standarddatenbank hergestellt. Beim Zugriff des Dienstprogramms auf Linux-, UNIX- oder Windows-Datenbankserver von Linux-, UNIX- oder Windows-Clients aus muss es sich um eine Direktverbindung über die Steuerkomponente handeln. Eine Verbindung über ein DB2 Connect-Gateway oder eine Loopback-Umgebung ist nicht zulässig.

Da das Dienstprogramm die Anweisung COMMIT absetzt, sollten Sie vor dem Aufrufen des Dienstprogramms EXPORT alle Transaktionen beenden und alle Sperren freigeben, indem Sie eine Anweisung COMMIT oder eine Anweisung ROLLBACK absetzen. Es ist nicht erforderlich, dass andere Benutzeranwendungen, die auf die Tabelle zugreifen und hierbei eigene Verbindungen verwenden, diese Verbindungen trennen.

Tabellen, die Spalten mit strukturiertem Typ enthalten, können nicht exportiert werden.

Informationen zu diesem Vorgang

Das Dienstprogramm IMPORT kann über IBM® Data Studio, durch Angeben des Befehls **EXPORT** im Befehlszeilenprozessor (CLP) oder durch Aufrufen der Anwendungsprogrammierschnittstelle (API) db2Export von einer Clientanwendung aus aufgerufen werden. Zwar kann auch das Notizbuch 'Tabelle exportieren' in der Steuerzentrale verwendet werden, es empfiehlt sich jedoch die Verwendung von Data Studio, da die Toolfunktionen der Steuerzentrale nicht weiter unterstützt werden.

Beispiel

Befehl EXPORT über den Befehlszeilenprozessor absetzen

Einfache Exportoperationen erfordern lediglich die Angabe einer Zieldatei, eines Dateiformats und einer Quelldatei in einer Anweisung SELECT.

Geben Sie zum Exportieren von Daten über den Befehlszeilenprozessor den Befehl **EXPORT** ein:

```
db2 export to dateiname of ixf select * from tabelle
```

Dabei ist `dateiname` der Name der Ausgabedatei, die erstellt und exportiert werden soll, `ixf` das Dateiformat und `tabelle` der Name der Tabelle mit den zu kopierenden Daten.

Es kann jedoch auch sinnvoll sein, eine Nachrichtendatei, in die Warnungen und Fehlermeldungen geschrieben werden können, anzugeben. Fügen Sie dazu den Parameter **MESSAGES** und den Namen einer Nachrichtendatei (z. B. `msg.txt`) ein. In diesem Fall ergibt sich folgender Befehl:

```
db2 export to dateiname of ixf messages msg.txt select * from tabelle
```

Die vollständige Syntax sowie Informationen zur Verwendung finden Sie im Abschnitt "Befehl EXPORT".

Exportieren von XML-Daten

Beim Exportieren von XML-Daten werden die dabei generierten XDM-Instanzen (XDM = XQuery Data Model) in eine oder mehrere Dateien geschrieben, die separat von der Hauptdatendatei abgelegt werden, die die exportierten relationalen Daten enthält. Dies gilt auch dann, wenn weder die Option `XMLFILE` noch die Option `XML TO` angegeben ist.

Standardmäßig werden alle exportierten QDM-Instanzen mit derselben XML-Datei verknüpft. Mit dem Dateitypmodifikator `XMLINSEPPFILES` können Sie angeben, dass alle QDM-Instanzen in eine separate Datei geschrieben werden sollen.

Die XML-Daten werden jedoch in der Hauptdatendatei durch eine XML-Datenkennung (XDS = XML Data Specifier) dargestellt. Die XML-Datenkennung besteht aus einer Zeichenfolge, die durch einen XML-Tag mit dem Namen "XDS" dargestellt wird, und verfügt über Attribute, die Informationen zu den eigentlichen XML-Daten in der Spalte enthalten. Hierzu gehören z. B. Angaben zum Namen der Datei, in der die eigentlichen XML-Daten enthalten sind, sowie Angaben zur relativen Position und Länge der XML-Daten in dieser Datei.

Die Zielpfade und Basisdateinamen der exportierten XML-Dateien können mit den Optionen `XML TO` und `XMLFILE` angegeben werden. Bei Angabe der Option `XML TO` oder `XMLFILE` lautet das Namensformat der exportierten und im Attribut `FIL` der XDS gespeicherten XML-Dateien `xmlfilespec.xxx.xml`, wobei `xmlfilespec` für den für die Option `XMLFILE` angegebenen Wert und `xxx` für eine vom Dienstprogramm `EXPORT` generierte Folgenummer für XML-Dateien steht. Ansonsten lautet das Namensformat der exportierten XML-Dateien `exportfilename.xxx.xml`, wobei `exportfilename` für den Namen der im Befehl `EXPORT` angegebenen exportierten Ausgabedatei und `xxx` für eine vom Dienstprogramm `EXPORT` generierte Folgenummer für XML-Dateien steht.

Standardmäßig werden exportierte XML-Dateien in den Pfad geschrieben, in dem sich auch die exportierte Datendatei befindet. Der Standardwert für den Basisdateinamen stimmt bei exportierten XML-Dateien mit dem Namen der exportierten Datendatei überein, wobei eine aus drei Ziffern bestehende Folgenummer und die Erweiterung `.xml` angefügt werden.

Beispiele

Gehen Sie in den folgenden Beispielen von einer Tabelle USER.T1 aus, die vier Spalten und zwei Zeilen enthält:

```
C1 INTEGER
C2 XML
C3 VARCHAR(10)
C4 XML
```

Tabelle 2. USER.T1

C1	C2	C3	C4
2	<?xml version="1.0" encoding="UTF-8" ?><note time="12:00:00"><to>You</to><from> Me</from><heading>note1</heading><body>Hello World!</body></note>	'char1'	<?xml version="1.0" encoding="UTF-8" ?><note time="13:00:00"><to>Him</to><from> Her</from><heading>note2</heading><body>Hello World!</body></note>
4	NULL	'char2'	?xml version="1.0" encoding="UTF-8" ?><note time="14:00:00"><to>Us</to><from> Them</from><heading>note3</heading><body>Hello World!</body></note>

Beispiel 1

Mit dem folgenden Befehl können Sie den Inhalt von USER.T1 im ASCII-Format mit Begrenzern (DEL) in die Datei "/mypath/t1export.del" exportieren. Da die Optionen XML TO und XMLFILE nicht angegeben wurden, werden die in den Spalten C2 und C4 enthaltenen XML-Dokumente in denselben Pfad geschrieben wie die exportierte Hauptdatei "/mypath". Der Basisdateiname für diese Dateien lautet "t1export.del.xml". Die Option XMLSAVESCHEMA gibt an, dass die XML-Schema-Informationen während der Exportprozedur gespeichert werden.

```
EXPORT TO /mypath/t1export.del OF DEL XMLSAVESCHEMA SELECT * FROM USER.T1
```

Die exportierte Datei "/mypath/t1export.del" enthält Folgendes:

```
2,"<XDS FIL='t1export.del.001.xml' OFF='0' LEN='144' />","char1",
"<XDS FIL='t1export.del.001.xml' OFF='144' LEN='145' />"
4,,"char2","<XDS FIL='t1export.del.001.xml' OFF='289'
LEN='145' SCH='S1.SCHEMA_A' />"
```

Die exportierte XML-Datei "/mypath/t1export.del.001.xml" enthält Folgendes:

```
<?xml version="1.0" encoding="UTF-8" ?><note time="12:00:00"><to>You</to>
<from>Me</from><heading>note1</heading><body>Hello World!</body>
</note><?xml version="1.0" encoding="UTF-8" ?><note time="13:00:00"><to>Him
</to><from>Her</from><heading>note2</heading><body>Hello World!
</body></note><?xml version="1.0" encoding="UTF-8" ?><note time="14:00:00">
<to>Us</to><from>Them</from><heading>note3</heading><body>
Hello World!</body></note>
```

Beispiel 2

Mit dem folgenden Befehl können Sie den Inhalt von USER.T1 im ASCII-Format mit Begrenzern (DEL) in die Datei "t1export.del" exportieren. XML-Dokumente, die in den Spalten C2 und C4 enthalten sind, werden in den Pfad "/home/user/xmlpath" geschrieben. Den XML-Dateien wird der Basisdateiname "xmldocs" zugeordnet, und es werden mehrere exportierte XML-Dokumente in dieselbe XML-Datei geschrieben. Die Option XMLSAVESHEMA gibt an, dass die XML-Schemainformationen während der Exportprozedur gespeichert werden.

```
EXPORT TO /mypath/t1export.del OF DEL XML TO /home/user/xmlpath
XMLFILE xmldocs XMLSAVESHEMA SELECT * FROM USER.T1
```

Die exportierte DEL-Datei "/home/user/t1export.del" enthält Folgendes:

```
2,"<XDS FIL='xmldocs.001.xml' OFF='0' LEN='144' />","char1",
"<XDS FIL='xmldocs.001.xml' OFF='144' LEN='145' />"
4,,"char2","<XDS FIL='xmldocs.001.xml' OFF='289'
LEN='145' SCH='S1.SCHEMA_A' />"
```

Die exportierte XML-Datei "/home/user/xmlpath/xmldocs.001.xml" enthält Folgendes:

```
<?xml version="1.0" encoding="UTF-8" ?><note time="12:00:00"><to>You</to>
<from>Me</from><heading>note1</heading><body>Hello World!</body>
</note><?xml version="1.0" encoding="UTF-8" ?><note time="13:00:00">
<to>Him</to><from>Her</from><heading>note2</heading><body>
Hello World!</body></note><?xml version="1.0" encoding="UTF-8" ?>
<note time="14:00:00"><to>Us</to><from>Them</from><heading>
note3</heading><body>Hello World!</body></note>
```

Beispiel 3

Der folgende Befehl ist mit dem Befehl aus Beispiel 2 vergleichbar. Eine Ausnahme bildet hierbei allerdings die Tatsache, dass alle exportierten XML-Dokumente in eine separate XML-Datei geschrieben werden.

```
EXPORT TO /mypath/t1export.del OF DEL XML TO /home/user/xmlpath
XMLFILE xmldocs MODIFIED BY XMLINSEPFILES XMLSAVESHEMA
SELECT * FROM USER.T1
```

Die exportierte Datei "/mypath/t1export.del" enthält Folgendes:

```
2,"<XDS FIL='xmldocs.001.xml' />","char1","XDS FIL='xmldocs.002.xml' />"
4,,"char2","<XDS FIL='xmldocs.004.xml' SCH='S1.SCHEMA_A' />"
```

Die exportierte XML-Datei "/home/user/xmlpath/xmldocs.001.xml" enthält Folgendes:

```
<?xml version="1.0" encoding="UTF-8" ?><note time="12:00:00"><to>You</to>
<from>Me</from><heading>note1</heading><body>Hello World!</body>
</note>
```

Die exportierte XML-Datei "/home/user/xmlpath/xmldocs.002.xml" enthält Folgendes:

```
?xml version="1.0" encoding="UTF-8" ?>note time="13:00:00">to>Him/to>
from>Her/from>heading>note2/heading>body>Hello World!/body>
/note>
```

Die exportierte XML-Datei "/home/user/xmlpath/xmldocs.004.xml" enthält Folgendes:

```
<?xml version="1.0" encoding="UTF-8" ?><note time="14:00:00"><to>Us</to>
<from>Them</from><heading>note3</heading><body>Hello World!</body>
</note>
```

Beispiel 4

Mit dem folgenden Befehl können Sie die Ergebnisse einer XQuery-Operation in eine XML-Datei schreiben.

```
EXPORT TO /mypath/t1export.del OF DEL XML TO /home/user/xmlpath
XMLFILE xmldocs MODIFIED BY XMLNODEDECLARATION select
xmlquery( '$m/note/from/text()' passing by ref c4 as "m" returning sequence)
from USER.T1
```

Die exportierte DEL-Datei "/mypath/t1export.del" enthält Folgendes:

```
"<XDS FIL='xmldocs.001.xml' OFF='0' LEN='3' />"
"<XDS FIL='xmldocs.001.xml' OFF='3' LEN='4' />"
```

Die exportierte XML-Datei "/home/user/xmlpath/xmldocs.001.xml" enthält Folgendes:

```
HerThem
```

Anmerkung: Das Ergebnis dieser speziellen XQuery-Operation generiert keine korrekt formatierten XML-Dokumente. Aus diesem Grund kann die oben exportierte Datei nicht direkt in eine XML-Spalte importiert werden.

Aspekte des Exports von LBAC-geschützten Daten

Wenn Sie Daten exportieren, die durch eine kennsatzbasierte Zugriffssteuerung (LBAC, Label-Based Access Control) geschützt sind, werden nur die Daten exportiert, die Sie gemäß Ihren LBAC-Berechtigungen nachweisen lesen dürfen.

Wenn Ihre LBAC-Berechtigungen Ihnen das Lesen einer Zeile nicht ermöglichen, wird die betreffende Zeile nicht exportiert, es wird jedoch kein entsprechender Fehler zurückgegeben. Wenn Ihre LBAC-Berechtigungen Ihnen das Lesen einer Spalte nicht ermöglichen, schlägt die Ausführung des Dienstprogramms EXPORT fehl, und es wird eine Fehlermeldung SQLSTATE 42512 zurückgegeben.

Werte aus Spalten mit dem Datentyp DB2SECURITYLABEL werden als Rohdaten, eingeschlossen in Zeichenbegrenzer, exportiert. Enthalten die Originaldaten einen Zeichenbegrenzer, wird dieser Begrenzer verdoppelt. An den Byte, die zusammen den exportierten Wert ergeben, werden ansonsten keine Änderungen vorgenommen. Dies bedeutet, dass eine Datendatei, die Daten des Typs DB2SECURITYLABEL enthält, Zeichen für Zeilenumbruch, Formularvorschub und andere nicht druckbare ASCII-Zeichen enthalten kann.

Wenn Sie die Werte des Datentyps DB2SECURITYLABEL in eine für Benutzer lesbare Form exportieren möchten, können Sie dazu die Skalarfunktion SECLABEL_TO_CHAR in der Anweisung SELECT verwenden, die die Werte in das Zeichenfolgeformat für Sicherheitskennsätze konvertiert.

Beispiele

In den folgenden Beispielen erfolgt die Ausgabe im Format DEL und wird in die Datei `myfile.del` geschrieben. Die Daten werden aus einer Tabelle mit der Bezeichnung REPS exportiert, die mit der folgenden Anweisung erstellt wurde:

```
create table reps (row_label db2securitylabel,  
id integer,  
name char(30))  
security policy data_access_policy
```

Bei dem folgenden Beispiel werden die Werte der Spalte "row_label" im Standardformat exportiert:

```
db2 export to myfile.del of del select * from reps
```

Die Datendatei ist in den meisten Texteditoren nur schlecht zu entziffern, da die Werte für die Spalte "row_label" in der Regel verschiedene ASCII-Steuerzeichen enthalten.

Bei dem folgenden Beispiel werden die Werte für die Spalte "row_label" im Zeichenfolgeformat für Sicherheitskennsätze exportiert:

```
db2 export to myfile.del of del select SECLABEL_TO_CHAR(row_label,  
'DATA_ACCESS_POLICY'), id, name from reps
```

Der folgende Auszug stammt aus der Datendatei, die im vorangehenden Beispiel erstellt wurde. Beachten Sie hierbei, dass das Format des Sicherheitskennsatzes lesbar ist:

```
...  
"Secret():Epsilon 37", 2005, "Susan Liu"  
"Secret():(Epsilon 37,Megaphone,Cloverleaf)", 2006, "Johnny Cogent"  
"Secret():(Megaphone,Cloverleaf)", 2007, "Ron Imron"  
...
```

Aspekte des Tabellenexports

Standardexportoperationen beinhalten die Ausgabe ausgewählter Daten, die in vorhandene Tabellen eingefügt oder geladen werden. Es ist jedoch auch möglich, vollständige Tabellen zu exportieren, um sie anschließend mit dem Dienstprogramm IMPORT erneut zu erstellen.

Zum Exportieren einer Tabelle müssen Sie das PC/IXF-Dateiformat angeben. Sie können die gespeicherte Tabelle anschließend einschließlich zugehöriger Indizes erneut erstellen, indem Sie das Dienstprogramm IMPORT im Modus CREATE verwenden. Bestimmte Informationen werden jedoch unter folgenden Umständen in der exportierten IXF-Datei nicht gespeichert:

- Die Indexspaltennamen enthalten den Hexadezimalwert 0x2B oder 0x2D.
- Die Tabelle enthält XML-Spalten.
- Bei der Tabelle handelt es sich um eine MDC-Tabelle (Multi-Dimensionally Clustered).
- Die Tabelle enthält einen Tabellenpartitionierungsschlüssel.
- Der Indexname ist auf Grund einer Codepagekonvertierung länger als 128 Byte.
- Die Tabelle ist geschützt.
- Der Befehl **EXPORT** enthält andere Aktionszeichenfolgen als `SELECT * FROM tabelle`
- Sie geben den Parameter **METHOD N** für das Dienstprogramm EXPORT an.

Eine Liste der Tabellenattribute, die verloren gehen, enthält der Abschnitt "Aspekte des Tabellenimports". Werden Informationen nicht gespeichert, wird die Warnung SQL27984W beim erneuten Erstellen der Tabelle zurückgegeben.

Anmerkung: Der Importmodus CREATE ist veraltet. Verwenden Sie das Dienstprogramm **db2look**, um Ihre Tabellen zu erfassen und erneut zu erstellen.

Indexinformationen

Enthalten die im Index angegebenen Spaltennamen das Zeichen - oder + , werden die Indexinformationen nicht erfasst und die Warnung SQL27984W wird zurückgegeben. Das Dienstprogramm EXPORT beendet die Verarbeitung. Die exportierten Daten sind nicht betroffen, die Indexinformationen werden jedoch nicht in der IXF-Datei gespeichert. Sie müssen die Indizes deshalb mit Hilfe des Dienstprogramms **db2look** separat erstellen.

Einschränkungen durch den Speicherplatz

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

Tabellen mit anderen Dateiformaten

Wenn Sie die Exportoperation nicht mit dem IXF-Dateiformat vornehmen, enthalten die Ausgabedateien zwar keine Beschreibungen der Zieltabelle, sie enthalten jedoch die Satzdaten. Um eine Tabelle mit den zugehörigen Daten erneut zu erstellen, erstellen Sie die Zieltabelle und verwenden anschließend das Dienstprogramm LOAD oder IMPORT, um die Tabelle mit Daten zu füllen. Mit dem Dienstprogramm **db2look** können Sie die ursprünglichen Tabellendefinitionen erfassen und die entsprechende Datendefinitionssprache (DDL) generieren.

Aspekte des Exports typisierter Tabellen

Mit dem DB2 Dienstprogramm EXPORT können Sie Daten aus typisierten Tabellen versetzen, um sie zu einem späteren Zeitpunkt zu importieren. Das Versetzen von Daten aus einer hierarchischen Struktur mit typisierten Tabellen in eine andere erfolgt beim Export entsprechend einer bestimmten Reihenfolge und durch Erstellung einer temporären Flachdatei.

Bei der Arbeit mit typisierten Tabellen steuert das Dienstprogramm EXPORT die Einfügungen in die Ausgabedatei. Sie müssen lediglich den Namen der Zieltabelle und bei Bedarf eine Klausel WHERE angeben. Subselect-Anweisungen können nur dadurch ausgedrückt werden, dass der Name der Zieltabelle und die Klausel WHERE angegeben wird. Die Angabe einer Anweisung "Fullselect" oder SELECT ist beim Export von Hierarchien nicht möglich.

Beibehaltung von Hierarchien mithilfe der Traversierfolge

Typisierte Tabellen können sich in einer Hierarchie befinden. Beim Versetzen von Daten zwischen verschiedenen Hierarchien gibt es verschiedene Möglichkeiten:

- Versetzen von Daten aus einer Hierarchie in eine identische Hierarchie
- Versetzen von Daten aus einer Hierarchie in einen Unterabschnitt einer größeren Hierarchie
- Versetzen von Daten aus einem Unterabschnitt einer großen Hierarchie in eine separate Hierarchie

Die Kennzeichnung von Typen in einer Hierarchie ist von der jeweiligen Datenbank abhängig. 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.

Das für typisierte Tabellen verwendete Zuordnungsprinzip basiert auf der *Traversierfolge*. Die Traversierfolge gibt vor, dass der Durchgang durch alle übergeordneten und untergeordneten Tabellen in der Hierarchie von oben nach unten und von links nach rechts erfolgt. 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 (der Traversierfolge) alle Typen durchnummeriert werden. Abbildung 1 zeigt eine Hierarchie mit vier gültigen Traversierfolgen:

- Person, Employee, Manager, Architect, Student
- Person, Student, Employee, Manager, Architect
- Person, Employee, Architect, Manager, Student
- Person, Student, Employee, Architect, Manager

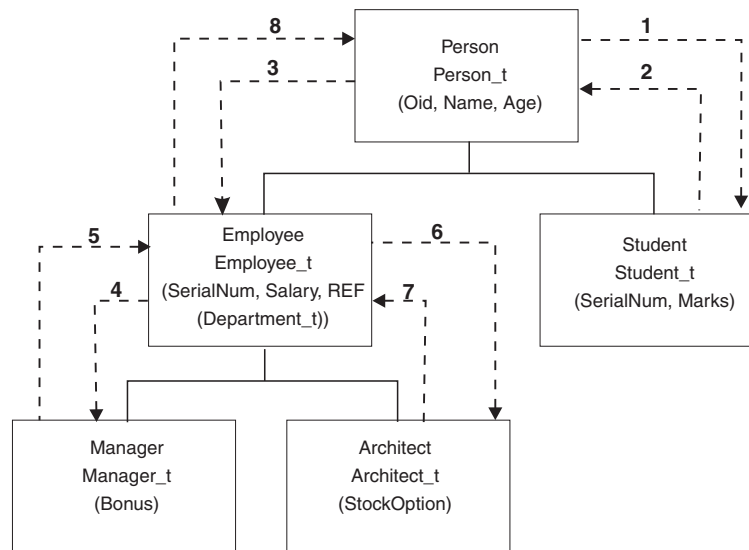


Abbildung 1. Hierarchiebeispiel

Die Traversierfolge ist wichtig beim Versetzen von Daten zwischen Tabellenhierarchien, weil dabei im Verhältnis zu anderen Daten ermittelt wird, wohin die Daten versetzt werden. Es gibt zwei Typen von Traversierfolgen: *Standardtraversierfolgen* und *benutzerdefinierte Traversierfolgen*.

Standardtraversierfolge

Bei der Standardtraversierfolge verweisen alle relevanten Typen ab einem bestimmten Startpunkt in der Hierarchie auf alle erreichbaren Typen der Hierarchie. Die Standardreihenfolge schließt alle Tabellen in der Hierarchie ein, und jede Tabelle wird nach dem Schema sortiert, das im erweiterten Sortiervergleichselement verwendet wird. Die Standardtraversierfolge in Abbildung 1, angegeben durch die gepunktete Linie, ist z. B. die Folge "Person, Student, Employee, Manager, Architect".

Die Standardtraversierfolge verhält sich bei verschiedenen Dateiformaten unterschiedlich. Beim Export von Daten in das PC/IXF-Dateiformat wird ein Datensatz mit allen relevanten Typen, ihren Definitionen und den relevanten Tabellen erzeugt. Das Dienstprogramm EXPORT nimmt außerdem die Zuordnung von Indexwerten für die einzelnen Tabellen vor. 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 führt zu Zeitdifferenzen, die die Standardtraversierfolge beim Durchgang durch die Hierarchien feststellt. 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 beim Erstellen 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

Bei der benutzerdefinierten Traversierfolge werden die zu verwendenden relevanten Typen vom Benutzer in einer Traversierfolgeliste definiert. Diese Reihenfolge gibt vor, wie die Hierarchie durchlaufen werden soll und welche untergeordneten Tabellen exportiert werden sollen, während bei der Standardtraversierfolge alle Tabellen der Hierarchie exportiert werden.

Beim Definieren der Traversierfolge können Sie zwar den Startpunkt und den Pfad durch die Hierarchie festlegen, es ist jedoch zu beachten, dass die untergeordneten Tabellen in einer *vordefinierten Reihenfolge* durchlaufen werden müssen. Jede Verzweigung der Hierarchie muss zunächst komplett durchlaufen werden, bevor mit einer neuen Verzweigung begonnen werden kann. Das Dienstprogramm EXPORT erkennt die Nichteinhaltung dieser Bedingung innerhalb der angegebenen Traversierfolge. Eine Methode, das Einhalten dieser Bedingung sicherzustellen, 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 "unmittelbar rechts angrenzende" untergeordnete Tabelle; danach zurück zur nächsthöheren übergeordneten Tabelle, weiter durch deren untergeordnete Tabellen usw.

Wenn Sie die Traversierfolge für die Hierarchien steuern möchten, müssen Sie sicherstellen, dass für die Dienstprogramme EXPORT und IMPORT die gleiche Traversierfolge verwendet wird.

Beispiel 1

Die folgenden Beispiele sind an der in Abbildung 1 dargestellten hierarchischen Struktur ausgerichtet. Geben Sie zum Exportieren der gesamten Hierarchie folgende Befehle ein:

```
DB2 CONNECT TO Source_db
DB2 EXPORT TO entire_hierarchy.ixf OF IXF HIERARCHY STARTING Person
```

Beachten Sie dabei, dass die Definition des Parameters **HIERARCHY STARTING** mit Person angibt, dass die Standardtraversierfolge mit der Tabelle PERSON beginnen soll.

Beispiel 2

Wenn Sie zwar die gesamte Hierarchie exportieren möchten, jedoch nur die Daten für Personen über 20, müssen Sie dazu die folgenden Befehle eingeben:

```
DB2 CONNECT TO Source_db
DB2 EXPORT TO entire_hierarchy.del OF DEL HIERARCHY (Person,
Employee, Manager, Architect, Student) WHERE Age>=20
```

Beachten Sie dabei, dass die Definition des Parameters **HIERARCHY** mit Person, Employee, Manager, Architect, Student auf eine benutzerdefinierte Traversierfolge verweist.

Aspekte des Identitätsspaltenexports

Mit dem Dienstprogramm EXPORT können Sie Daten aus Tabellen exportieren, die Identitätsspalten enthalten. Die Identitätsspalten schränken jedoch die Möglichkeiten bei der Auswahl der Ausgabedateiformate ein.

Hat die Anweisung SELECT, 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 die exportierte 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` für den Dateityp angegeben wird. Andernfalls werden alle Zeilen zurückgewiesen, und es wird die Fehlermeldung SQL3550W ausgegeben.

Anmerkung: Die Optionen CREATE und REPLACE_CREATE des Befehls IMPORT werden nicht weiter unterstützt und in zukünftigen Releases möglicherweise entfernt.

Aspekte des LOB-Exports

Beim Exportieren von Tabellen mit LOB-Spalten (Large Objects) besteht die Standardaktion darin, die ersten 32 KB Daten pro LOB-Wert auszuwählen und diese in dieselbe Datei wie die restlichen Spaltendaten zu stellen. Beim Export von LOB-Werten über 32 KB können Sie vermeiden, dass die Daten abgeschnitten werden, indem Sie dafür sorgen, dass die LOB-Daten in eine separate Datei geschrieben werden.

Mit dem Änderungswert `lobsinfile` für den Dateityp können Sie angeben, dass LOB-Daten in eine separate Datei geschrieben werden. Dieser Änderungswert bewirkt beim Dienstprogramm EXPORT, dass die LOB-Daten in die in der Klausel LOBS TO angegebenen Verzeichnisse gestellt werden. Die Verwendung der Optionen LOBS TO und LOBFILE impliziert eine Aktivierung des Änderungswerts für den Dateityp `lobsinfile`. Standardmäßig werden LOB-Werte in denselben Pfad geschrieben wie die exportierten relationalen Daten. Sind LOB-Pfade mit der Option LOBS TO angegeben ist, wird das Dienstprogramm EXPORT reihum zwischen den Pfaden ausgeführt, um jeden erfolgreichen LOB-Wert in die entsprechende LOB-Datei zu schreiben. Sie können mit der Option LOBFILE auch Namen für die LOB-Ausgabedateien angeben. Bei Angabe der Option LOBFILE sieht das Format des LOB-Dateinamens wie folgt aus: `lob-dateispez.xxx.lob`. Dabei stellt `lob-dateispez` den Wert dar, der für die Option LOBFILE angegeben wurde, und `xxx` ist eine Folgenummer für die vom Dienstprogramm EXPORT erstellten LOB-Dateien. Andernfalls lautet das Format des LOB-Dateinamens `exportdateiname.xxx.lob`. Dabei steht `exportdateiname` für den Namen der exportierten Ausgabedatei, die für den

Befehl EXPORT angegeben wurde, und xxx ist eine Folgenummer für die vom Dienstprogramm EXPORT erstellten LOB-Dateien.

Standardmäßig werden LOBs in eine einzige Datei geschrieben. Sie können bei Bedarf jedoch auch angeben, dass einzelne LOB-Daten in separaten Dateien gespeichert werden sollen. Das Dienstprogramm EXPORT generiert eine LOB-Positionskennung (LLS, LOB Location Specifier), um ein Speichern von mehreren LOBs in einer einzigen Datei zu ermöglichen. Bei der LOB-Positionskennung, die in die Exportausgabedatei geschrieben wird, handelt es sich um eine Zeichenfolge, die die Position der LOB-Daten in der Datei angibt. Die LOB-Positionskennung hat folgendes Format: lob-dateiname.erw.nnn.mmm/. Hierbei steht lob-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.

Wenn einzelne LOB-Daten nicht in einer Datei verknüpft werden sollen, können Sie mit dem Änderungswert lobsinsefiles für den Dateityp angeben, dass die einzelnen LOBs in eine separate Datei geschrieben werden.

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.

Beispiel 1

Das folgende Beispiel veranschaulicht den Export von LOBs in eine DEL-Datei. Die exportierten LOB-Dateien werden mit dem angegebenen Basisnamen lob1 bezeichnet.

```
db2 export to myfile.del of del lob1 to mylob1/  
lobfile lob1 modified by lobsinfile  
select * from emp_photo
```

Beispiel 2

Das folgende Beispiel veranschaulicht, wie LOBs in eine DEL-Datei exportiert werden können, wobei die einzelnen LOB-Werte jeweils in eine separate Datei und die LOB-Dateien in zwei verschiedene Verzeichnisse geschrieben werden:

```
db2 export to myfile.del of del  
lob1 to /db2exp1/, /db2exp2/ modified by lobsinfile  
select * from emp_photo
```

Referenz - Export

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.

Quick Link zu „Änderungswerte für Dateityp für das Dienstprogramm EXPORT “ auf Seite 24.

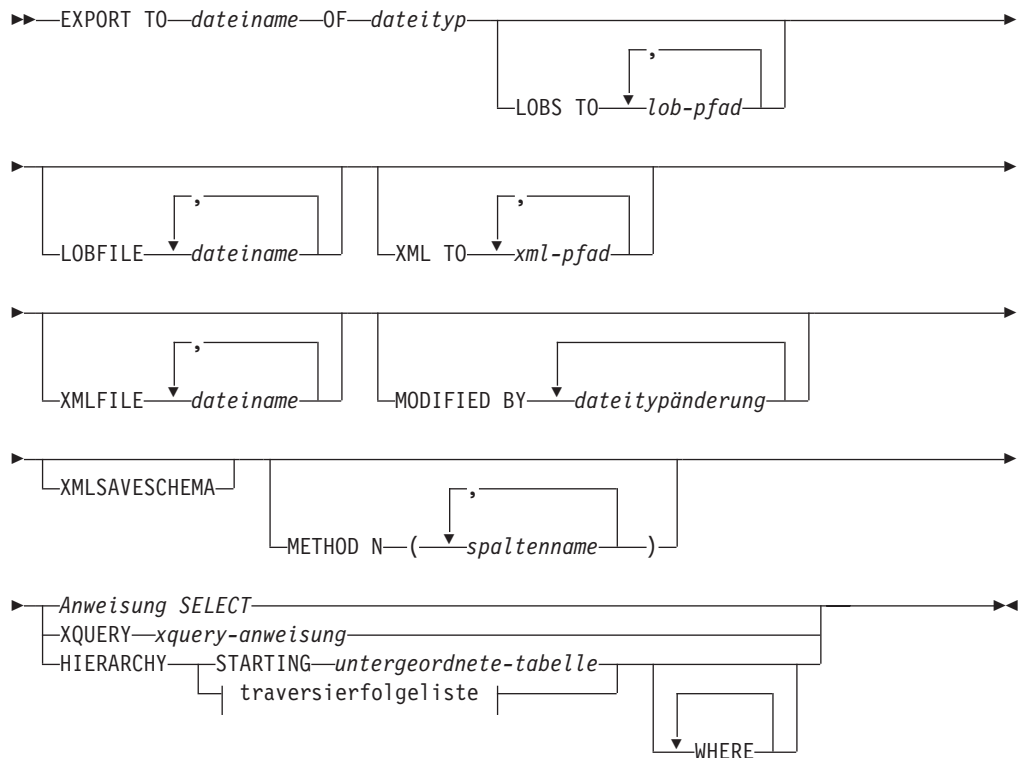
Berechtigung

Dies sind:

- Berechtigung DATAACCESS
- Zugriffsrecht CONTROL oder SELECT für jede betroffene Tabelle oder Sicht

Erforderliche Verbindung

Befehlssyntax



traversierfolgeliste:



Befehlsparameter

HIERARCHY *traversierfolgeliste*

Exportiert eine untergeordnete Hierarchie mit der angegebenen Traversierfolge. Alle untergeordneten Tabellen müssen in PRE-ORDER-Weise aufgelistet sein. Der Name der ersten untergeordneten Tabelle wird als Zieltabelle 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*.

LOBFILE *dateiname*

Gibt einen oder mehrere Basisdateinamen für die LOB-Dateien an. Wenn der Namensbereich für den ersten Namen erschöpft ist, wird der zweite Name verwendet, und so weiter. Dadurch wird implizit das Verhalten LOBSINFILE aktiviert.

Beim Erstellen von LOB-Dateien während einer Exportoperation werden die Dateinamen gebildet, indem der aktuelle Basisname der Liste an den aktuellen Pfad (*lob-pfad*) und anschließend zunächst eine dreistellige Folgenummer und die aus drei Zeichen bestehende Kennung *lob* angehängt werden. Ist der aktuelle LOB-Pfad z. B. das Verzeichnis */u/foo/lob/path/* und lautet der aktuelle LOB-Dateiname *bar*, werden die LOB-Dateien */u/foo/lob/path/bar.001.lob*, */u/foo/lob/path/bar.002.lob* etc. erstellt. Die dreistellige Folgenummer im LOB-Dateinamen wird zu einer vierstelligen Nummer, wenn Nummer 999 vergeben ist, die vierstelligen Nummer wird zu einer fünfstelligen, wenn 9999 vergeben ist 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). Es können maximal 999 Pfade angegeben werden. Dadurch wird implizit das Verhalten LOBSINFILE aktiviert.

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 *dateitypänderung*

Gibt Optionen für Änderungswerte für Dateityp an. Siehe „Änderungswerte für Dateityp für das Dienstprogramm EXPORT “ auf Seite 24.

OF *dateityp*

Gibt das Format der Daten in der Ausgabedatei an:

- DEL (ASCII-Format ohne universelle Zeilenbegrenzer) wird von verschiedenen Datenbankmanager- und Dateimanagerprogrammen verwendet.
- WSF (Work Sheet Format) wird von verschiedenen Programmen wie den folgenden verwendet:
 - Lotus 1-2-3
 - Lotus Symphony

Beim Exportieren von BIGINT- oder DECIMAL-Daten können nur Werte, die innerhalb des Bereichs des Typs DOUBLE liegen, korrekt exportiert werden. Werte, die nicht in diesem Bereich liegen, werden zwar ebenfalls exportiert, beim erneuten Importieren oder Laden dieser Werte können sich je nach Betriebssystem jedoch fehlerhafte Daten ergeben.

Anmerkung: Die Unterstützung für das WSF-Dateiformat ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.

- IXF (Integration Exchange Format, PC-Version) ist ein proprietäres Binärformat.

Anweisung SELECT

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

TO *dateiname*

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

XMLFILE *dateiname*

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

Beim Erstellen von XML-Dateien während einer Exportoperation werden die Dateinamen gebildet, indem der aktuelle Basisname aus dieser Liste an den aktuellen Pfad (*xml-pfad*) und anschließend eine dreistellige Folgenummer und die aus drei Zeichen bestehende Kennung *xml* angehängt wird. Ist der aktuelle XML-Pfad z. B. das Verzeichnis */u/foo/xml/path/* und lautet der Name der aktuellen XML-Datei *bar*, werden die XML-Dateien */u/foo/xml/path/bar.001.xml*, */u/foo/xml/path/bar.002.xml* etc. erstellt.

XML TO *xml-pfad*

Gibt einen oder mehrere Pfade zu Verzeichnissen an, in denen die XML-Dateien gespeichert werden sollen. Jeder XML-Pfad enthält mindestens eine Datei, und jede Datei enthält mindestens eine XDM-Instanz (XDM = XQuery Data Model). Wenn mehr als ein Pfad angegeben ist, werden die XDM-Instanzen gleichmäßig auf die Pfade verteilt.

XMLSAVESCHEMA

Gibt an, dass XML-Schemainformationen für alle XML-Spalten gespeichert werden sollen. Bei allen exportierten XML-Dokumenten, die beim Einfügen anhand eines XML-Schemas geprüft wurden, wird die vollständig qualifizierte SQL-Kennung des betreffenden Schemas als Attribut (SCH) in der entsprechenden XML-Datenkennung (XDS) gespeichert. Bei exportierten Dokumenten, die nicht anhand eines XML-Schemas geprüft wurden oder bei denen das Schemaobjekt nicht mehr in der Datenbank vorhanden ist, ist das Attribut SCH nicht in der entsprechenden XDS enthalten.

Die Schema- und Namensbestandteile der SQL-Kennung werden als Werte für "OBJECTSCHEMA" bzw. "OBJECTNAME" in der Katalogtabelle SYS-CAT.XSROBJECTS in der Zeile für das jeweilige XML-Schema gespeichert.

Die Option **XMLSAVESCHEMA** ist nicht mit XQuery-Sequenzen kompatibel, die keine korrekt formatierten XML-Dokumente erzeugen.

Hinweise zur Verwendung

- Achten Sie darauf, alle Tabellenoperationen abzuschließen und alle Sperren freizugeben, bevor Sie eine Exportoperation starten. Dies erreichen Sie, indem Sie ein COMMIT absetzen, nachdem Sie alle geöffneten Cursor mit WITH HOLD geschlossen haben, oder indem Sie einen ROLLBACK 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.
- 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 ist nicht erforderlich, wenn die Quellen- und die Zieldatenbank von demselben Client aus zugänglich sind.
- Mit DB2 Connect können Tabellen aus DRDA-Servern wie DB2 für OS/390, DB2 für VM und VSE und DB2 für OS/400 exportiert werden. Nur der PC/IXF-Export wird unterstützt.
- Ist der Umfang von Kennungen beim Export in das IXF-Format größer als vom IXF-Format unterstützt, verläuft die Exportoperation zwar erfolgreich, die dabei entstandene Datendatei kann jedoch nicht für nachfolgende Importoperationen im Modus CREATE verwendet werden. Die Nachricht SQL27984W wird zurückgegeben.
- Wenn die Tabelle mehr Daten enthält, als beim Exportieren unter Windows auf eine Diskette passen, fordert das System zum Einlegen einer weiteren Diskette auf. In diesem Fall werden mehrteilige PC/IXF-Dateien (auch als PC/IXF-Mehrfachdateien oder logisch geteilte PC/IXF-Dateien bezeichnet) generiert und auf separaten Disketten gespeichert. Jede dieser Dateien (mit Ausnahme der letzten) enthält einen DB2-FORTSETZUNGSSATZ (CONTINUATION RECORD oder kurz "AC"-Satz), der angibt, dass die Dateien logisch geteilt wurden und wo die nächste Datei zu finden ist. Diese Dateien können auf ein AIX-System übertragen und dort von den Dienstprogrammen IMPORT und LOAD gelesen werden. Wenn das Dienstprogramm EXPORT über ein AIX-System aufgerufen wird, werden keine mehrteiligen PC/IXF-Dateien erstellt. Ausführliche Angaben zur Verwendung finden Sie in den Abschnitten über die Befehle **IMPORT** und **LOAD**.
- 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 Subselect-Anweisungen nur dadurch ausgedrückt werden, dass der Name der Zieltabelle und die Klausel **WHERE** angegeben wird. Beim Exportieren einer Hierarchie können Fullselect und *select-anweisung* nicht angegeben werden.
- Bei anderen Dateiformaten als IXF empfiehlt es sich, die Traversierfolgeliste anzugeben. Sie teilt DB2 mit, wie die Hierarchie durchquert werden soll 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.
- Während einer Importoperation sollte die gleiche Traversierfolge verwendet werden. Das Dienstprogramm LOAD unterstützt das Laden von Hierarchien oder Unterhierarchien nicht.
- Beim Exportieren von Daten aus einer Tabelle mit geschützten Zeilen begrenzen die LBAC-Berechtigungs-ID für die Berechtigungs-ID der Sitzung möglicherweise die Zeilen, die exportiert werden. Zeilen, auf die die Berechtigungs-ID der Sitzung keinen Lesezugriff hat, werden nicht exportiert. Es wird keine Fehlermeldung oder Warnung ausgegeben.

- Wenn die LBAC-Berechtigungs nachweise das Lesen von einzelnen Spalten, die exportiert werden sollen, nicht zulässt, schlägt die Exportoperation fehl, und es wird der Fehler SQLSTATE 42512 zurückgegeben.
- Bei der Ausführung von Dienstprogrammen für das Versetzen von Daten wie **export** und **db2move** stellt der Abfragecompiler möglicherweise fest, dass die zugrunde liegende Abfrage für eine MQT (Materialized Query Table, gespeicherte Abfragetabelle) effizienter ausgeführt wird als für die Basistabelle(n). In diesem Fall wird die Abfrage für eine als REFRESH DEFERRED deklarierte MQT ausgeführt und das Ergebnis der Dienstprogramme zeigt die Daten in der zugrunde liegenden Tabelle möglicherweise nicht exakt an.
- Exportpakete werden entsprechend dem Format DATETIME ISO gebunden. Datums-, Zeit- und Zeitmarkenwerte werden dementsprechend beim Umsetzen in eine Zeichenfolgedarstellung in das ISO-Format konvertiert. Da die CLP-Pakete mit dem Format DATETIME LOC (länderspezifisches Format) gebunden werden, kann es zu inkonsistentem Verhalten zwischen CLP und Exportoperation kommen, wenn das DATETIME-Format von CLP nicht mit dem ISO-Format übereinstimmt. Dazu ein Beispiel: Die folgende Anweisung SELECT gibt die erwarteten Ergebnisse zurück:

```
db2 select col2 from tab1 where char(col2)='05/10/2005';
      COL2
-----
      05/10/2005
      05/10/2005
      05/10/2005
3 Satz/Sätze ausgewählt.
```

Ein Exportbefehl mit derselben Klausel SELECT liefert jedoch nicht die erwarteten Ergebnisse:

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

Wenn Sie das Datumsformat für die Ländereinstellung (LOCALE) durch das ISO-Format ersetzen, werden die erwarteten Ergebnisse zurückgegeben:

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

Änderungswerte für Dateityp für das Dienstprogramm EXPORT

Tabelle 3. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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.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. Wenn 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 <i>db2exp.001</i> und ist 456 Byte lang.</p> <p>Wenn Sie den Änderungswert lobsinfile bei Verwendung des Befehls EXPORT angeben, werden die LOB-Daten an den in der Klausel LOBS TO angegebenen Positionen platziert. Andernfalls werden die LOB-Daten an das Datendateiverzeichnis 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). Die Option LOBS TO bzw. LOBFILE aktiviert implizit das Verhalten LOBSINFILE.</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>
xmlinsefiles	Jede XDM-Instanz (XDM = XQuery Data Model) wird in eine separate Datei geschrieben. Standardmäßig werden mehrere Werte in derselben Datei miteinander verknüpft.
lobsinsefiles	Jeder LOB-Wert wird in eine separate Datei geschrieben. Standardmäßig werden mehrere Werte in derselben Datei miteinander verknüpft.
xmlnodeclaration	XDM-Instanzen werden ohne XML-Deklarationstag geschrieben. Standardmäßig werden XDM-Instanzen mit einem vorangestellten XML-Deklarationstag exportiert, das ein Codierungsattribut enthält.
xmlchar	XDM-Instanzen werden in der Zeichencodierung geschrieben. Dabei ist zu beachten, dass die Zeichencodierung dem Wert entspricht, der für den Änderungswert für Dateityp <i>codepage</i> angegeben ist, oder der <i>Codepage</i> der Anwendung, wenn dieser Wert nicht angegeben ist. Standardmäßig werden XDM-Instanzen in Unicode exportiert.
xmlgraphic	Bei Angabe des Änderungswerts <i>xmlgraphic</i> mit dem Befehl EXPORT wird das exportierte XML-Dokument in der <i>Codepage</i> UTF-16 codiert, unabhängig von der <i>Codepage</i> der Anwendung oder dem Änderungswert <i>codepage</i> für den Dateityp.

Tabelle 4. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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.² 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>
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 Anwendungscodepage 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. Der Änderungswert codepage kann nicht zusammen mit dem Änderungswert lobsinfile verwendet werden.</p>
coldelx	<p><i>x</i> ist ein einzelnes Zeichen, das als Spaltenbegrenzer verwendet wird. Der Standardwert ist ein Komma (.). Das angegebene Zeichen wird anstelle eines Kommas verwendet, um das Ende einer Spalte anzugeben.²</p> <p>Im folgenden Beispiel bewirkt coldel;, dass das Dienstprogramm EXPORT das Semikolon (;) als Spaltenbegrenzungszeichen für die exportierten Daten verwendet:</p> <pre>db2 "export to temp of del modified by coldel; select * from staff where dept = 20"</pre>
decplusblank	<p>Zeichen für Pluszeichen. Bewirkt, dass positiven Dezimalwerten statt eines Pluszeichens (+) ein Leerzeichen vorangestellt wird. Die Standardaktion besteht darin, positiven Dezimalwerten ein Pluszeichen voranzustellen.</p>
decptr	<p><i>x</i> ist ein einzelnes Ersatzzeichen für den Punkt als Dezimalzeichen. Der Standardwert ist ein Punkt (.). Das angegebene Zeichen wird statt eines Punkts als Dezimalzeichen verwendet.²</p>
nochardel	<p>Spaltendaten werden nicht von Zeichenbegrenzern eingeschlossen. Diese Option sollte nicht angegeben werden, wenn die Daten mithilfe von DB2 importiert oder geladen werden sollen. 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 oder nodouble del angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p>
nodouble del	<p>Unterdrückt die Erkennung von doppelten Zeichenbegrenzern.²</p>

Tabelle 4. Gültige Änderungswerte für Dateityp für das Dienstprogramm EXPORT: DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer) (Forts.)

Änderungswert	Beschreibung
timestampformat="x"	<p>x ist das Format der Zeitmarke in der Quellendatei.⁴ 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 01 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 00 bis 12 bei einem 12-Stunden-System und 00 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 00 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 00 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 - 86400; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus)</p> <p>U (1 bis 12 mal) - Bruchsekunden (die Häufigkeit von U gibt die Anzahl der Stellen mit Werten aus dem Bereich 0 bis 9 an)</p> <p>TT - Meridiananzeiger (AM oder PM)</p> <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 5. Gültige Änderungswerte für Dateityp für das Dienstprogramm EXPORT: IXF-Dateiformat

Änderungswert	Beschreibung
codepage=x	x ist eine ASCII-Zeichenfolge. Der Wert wird als Codepage der Daten im Ausgabedatensatz gewertet. Während der Exportoperation werden Zeichendaten aus dieser Codepage in die Codepage der Anwendung konvertiert. Bei reinem DBCS (Grafik), gemischtem DBCS und EUC sind Begrenzer auf den Bereich von x00 bis x3F (jeweils einschließlich) beschränkt.

Tabelle 6. Gültige Änderungswerte für Dateityp für das Dienstprogramm EXPORT: WSF-Dateiformat⁶

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

Anmerkung:

1. Das Dienstprogramm EXPORT gibt keine Warnung aus, 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.
2. *Aspekte beim Versetzen von XML-Daten - Begrenzer* listet Einschränkungen für Zeichen auf, die anstelle der standardmäßigen Begrenzer verwendet werden können.
3. 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.uuuuuuu*

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

4. 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)

5. Diese Dateien können auch einem bestimmten Produkt zugeordnet werden, indem ein L für Lotus 1-2-3 oder ein S für Symphony in der Zeichenfolge des Parameters *dateitypänderungswert* angegeben wird. Es kann nur ein einziger Wert bzw. Produktbezeichner angegeben werden. Die Unterstützung für das WSF-Dateiformat ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.
6. Das WSF-Format wird für XML-Spalten nicht unterstützt. Die Unterstützung für dieses Dateiformat ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.
7. Alle XDM-Instanzen werden in separate XML-Dateien geschrieben (nicht in die Hauptdatendatei). Dies gilt auch, wenn weder die Klausel **XMLFILE** noch die Klausel **XML TO** angegeben wird. Standardmäßig werden XML-Dateien in den Pfad der exportierten Datendatei geschrieben. Der Standardwert für den Basisdateinamen für XML-Dateien ist der Name der exportierten Datendatei mit der Erweiterung ".xml".
8. Alle XDM-Instanzen werden standardmäßig mit einer vorangestellten XML-Deklaration exportiert, die ein Codierungsattribut enthält, sofern nicht der Änderungswert für den Dateityp XMLNODEDECLARATION angegeben ist.
9. Standardmäßig werden alle XDM-Instanzen im Unicode-Format exportiert, sofern nicht der Änderungswert für den Dateityp XMLCHAR oder XMLGRAPHIC angegeben ist.
10. Der Standardpfad für XML-Daten und LOB-Daten ist der Pfad der Hauptdatendatei. Der Standardbasisname für XML-Dateien ist der Name der Hauptdatendatei. Der Standardbasisname für LOB-Dateien ist der Name der Hauptdatendatei. Bei einer Hauptdatendatei mit dem Namen
`/mypath/myfile.del`
ist der Standardpfad für XML-Daten und LOB-Daten z. B.
`/mypath"`
Der Standardbasisname für XML-Dateien ist
`myfile.del`
und der Standardbasisname für LOB-Dateien ist
`myfile.del`
Der Änderungswert für Dateityp LOBSINFILE muss angegeben sein, damit LOB-Dateien generiert werden.
11. Vom Dienstprogramm EXPORT wird eine numerische Kennung an die einzelnen LOB- bzw. XML-Dateien angehängt. Die Kennung ist ein dreistelliger, mit Nullen aufgefüllter Folgewert, der mit folgenden Zeichen Kennung beginnt:
.001

Nach der 999. LOB- bzw. XML-Datei wird die Kennung nicht mehr mit Nullen aufgefüllt (die 1000. LOB- bzw. XML-Datei hat z. B. die folgende Kennung:
.1000

Auf die numerische Kennung folgt eine aus drei Zeichen bestehende Typenkennung für den Datentyp:

.lob

oder

.xml

Eine generierte LOB-Datei wird z. B. mit einem Namen im folgenden Format gekennzeichnet:

myfile.del.001.lob

Eine generierte XML-Datei wird z. B. mit einem Namen im folgenden Format gekennzeichnet:

myfile.del.001.xml

12. Mit dem Dienstprogramm EXPORT können auch XDM-Instanzen exportiert werden, die keine korrekt formatierten Dokumente sind, wenn ein XQuery-Ausdruck angegeben wird. Sie können die exportierten Dokumente anschließend jedoch nicht direkt in eine XML-Spalte importieren bzw. laden, da XML-Spalten nur vollständige Dokumente enthalten können.

Befehl EXPORT mit Prozedur ADMIN_CMD

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.

Quick Link zu „Änderungswerte für Dateityp für das Dienstprogramm EXPORT “ auf Seite 35.

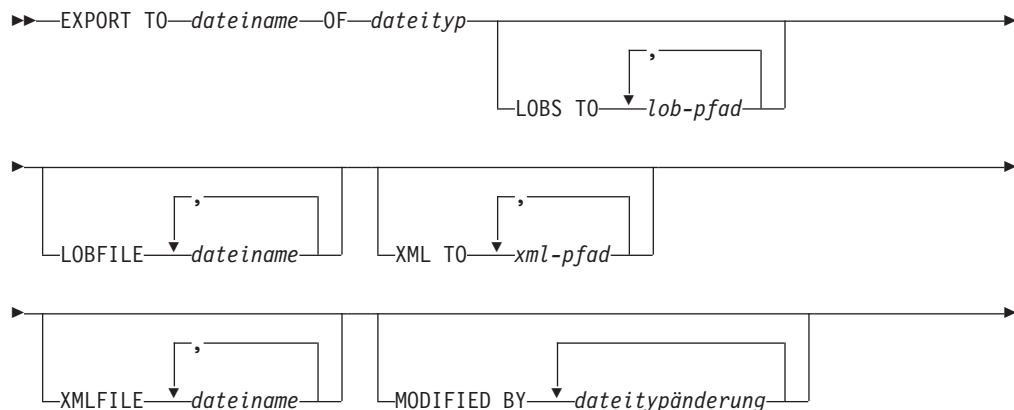
Berechtigung

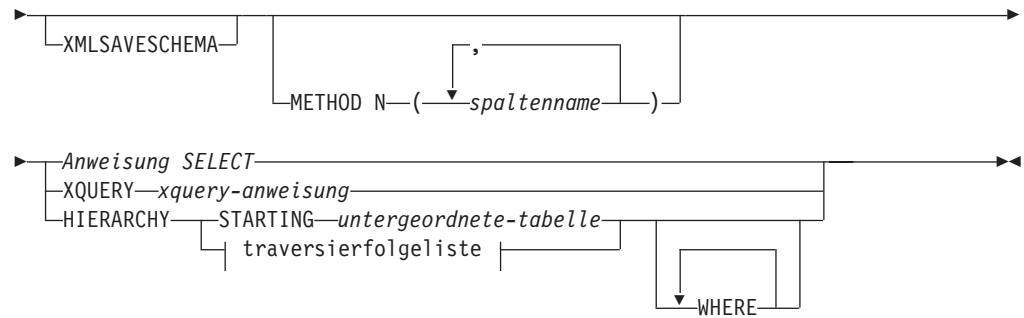
Dies sind:

- Berechtigung DATAACCESS
- Zugriffsrecht CONTROL oder SELECT für jede betroffene Tabelle oder Sicht

Erforderliche Verbindung

Befehlssyntax





traversierfolgeliste:



Befehlsparameter

HIERARCHY *traversierfolgeliste*

Exportiert eine untergeordnete Hierarchie mit der angegebenen Traversierfolge. Alle untergeordneten Tabellen müssen in PRE-ORDER-Weise aufgelistet sein. Der Name der ersten untergeordneten Tabelle wird als Zieltabelle 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*.

LOBFILE *dateiname*

Gibt einen oder mehrere Basisdateinamen für die LOB-Dateien an. Wenn der Namensbereich für den ersten Namen erschöpft ist, wird der zweite Name verwendet, und so weiter. Dadurch wird implizit das Verhalten LOBSINFILE aktiviert.

Beim Erstellen von LOB-Dateien während einer Exportoperation werden die Dateinamen gebildet, indem der aktuelle Basisname der Liste an den aktuellen Pfad (*lob-pfad*) und anschließend zunächst eine dreistellige Folgenummer und die aus drei Zeichen bestehende Kennung lob angehängt werden. Ist der aktuelle LOB-Pfad z. B. das Verzeichnis /u/foo/lob/path/ und lautet der aktuelle LOB-Dateiname bar, werden die LOB-Dateien /u/foo/lob/path/bar.001.lob, /u/foo/lob/path/bar.002.lob etc. erstellt. Die dreistellige Folgenummer im LOB-Dateinamen wird zu einer vierstelligen Nummer, wenn Nummer 999 vergeben ist, die vierstelligen Nummer wird zu einer fünfstelligen, wenn 9999 vergeben ist 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). Es können maximal 999 Pfade angegeben werden. Dadurch wird implizit das Verhalten LOBSINFILE aktiviert.

METHOD N *spaltenname*

Gibt einen oder mehrere Spaltennamen an, die in der Ausgabedatei ver-

wendet 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 *dateitypänderung*

Gibt Optionen für Änderungswerte für Dateityp an. Siehe „Änderungswerte für Dateityp für das Dienstprogramm EXPORT “ auf Seite 35.

OF *dateityp*

Gibt das Format der Daten in der Ausgabedatei an:

- DEL (ASCII-Format ohne universelle Zeilenbegrenzer) wird von verschiedenen Datenbankmanager- und Dateimanagerprogrammen verwendet.
- WSF (Work Sheet Format) wird von verschiedenen Programmen wie den folgenden verwendet:
 - Lotus 1-2-3
 - Lotus Symphony

Beim Exportieren von BIGINT- oder DECIMAL-Daten können nur Werte, die innerhalb des Bereichs des Typs DOUBLE liegen, korrekt exportiert werden. Werte, die nicht in diesem Bereich liegen, werden zwar ebenfalls exportiert, beim erneuten Importieren oder Laden dieser Werte können sich je nach Betriebssystem jedoch fehlerhafte Daten ergeben.

Anmerkung: Die Unterstützung für das WSF-Dateiformat ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.

- IXF (Integration Exchange Format, PC-Version) ist ein proprietäres Binärformat.

Anweisung SELECT

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

TO *dateiname*

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

XMLFILE *dateiname*

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

Beim Erstellen von XML-Dateien während einer Exportoperation werden die Dateinamen gebildet, indem der aktuelle Basisname aus dieser Liste an den aktuellen Pfad (*xml-pfad*) und anschließend eine dreistellige Folgenummer und die aus drei Zeichen bestehende Kennung xml angehängt wird. Ist der aktuelle XML-Pfad z. B. das Verzeichnis /u/foo/xml/path/ und lautet der Name der aktuellen XML-Datei bar, werden die XML-Dateien /u/foo/xml/path/bar.001.xml, /u/foo/xml/path/bar.002.xml etc. erstellt.

XML TO *xml-pfad*

Gibt einen oder mehrere Pfade zu Verzeichnissen an, in denen die XML-Dateien gespeichert werden sollen. Jeder XML-Pfad enthält mindestens eine Datei, und jede Datei enthält mindestens eine XDM-Instanz (XDM = XQuery Data Model). Wenn mehr als ein Pfad angegeben ist, werden die XDM-Instanzen gleichmäßig auf die Pfade verteilt.

XMLSAVESHEMA

Gibt an, dass XML-Schemainformationen für alle XML-Spalten gespeichert werden sollen. Bei allen exportierten XML-Dokumenten, die beim Einfügen anhand eines XML-Schemas geprüft wurden, wird die vollständig qualifizierte SQL-Kennung des betreffenden Schemas als Attribut (SCH) in der entsprechenden XML-Datenkennung (XDS) gespeichert. Bei exportierten Dokumenten, die nicht anhand eines XML-Schemas geprüft wurden oder bei denen das Schemaobjekt nicht mehr in der Datenbank vorhanden ist, ist das Attribut SCH nicht in der entsprechenden XDS enthalten.

Die Schema- und Namensbestandteile der SQL-Kennung werden als Werte für "OBJECTSCHEMA" bzw. "OBJECTNAME" in der Katalogtabelle SYS-CAT.XSROBJECTS in der Zeile für das jeweilige XML-Schema gespeichert.

Die Option **XMLSAVESHEMA** ist nicht mit XQuery-Sequenzen kompatibel, die keine korrekt formatierten XML-Dokumente erzeugen.

Hinweise zur Verwendung

- Achten Sie darauf, alle Tabellenoperationen abzuschließen und alle Sperren freizugeben, bevor Sie eine Exportoperation starten. Dies erreichen Sie, indem Sie ein COMMIT absetzen, nachdem Sie alle geöffneten Cursor mit WITH HOLD geschlossen haben, oder indem Sie einen ROLLBACK 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.
- 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 ist nicht erforderlich, wenn die Quellen- und die Zieldatenbank von demselben Client aus zugänglich sind.
- Mit DB2 Connect können Tabellen aus DRDA-Servern wie DB2 für OS/390, DB2 für VM und VSE und DB2 für OS/400 exportiert werden. Nur der PC/IXF-Export wird unterstützt.
- Ist der Umfang von Kennungen beim Export in das IXF-Format größer als vom IXF-Format unterstützt, verläuft die Exportoperation zwar erfolgreich, die dabei entstandene Datendatei kann jedoch nicht für nachfolgende Importoperationen im Modus CREATE verwendet werden. Die Nachricht SQL27984W wird zurückgegeben.
- Wenn die Tabelle mehr Daten enthält, als beim Exportieren unter Windows auf eine Diskette passen, fordert das System zum Einlegen einer weiteren Diskette auf. In diesem Fall werden mehrteilige PC/IXF-Dateien (auch als PC/IXF-Mehrfachdateien oder logisch geteilte PC/IXF-Dateien bezeichnet) generiert und auf separaten Disketten gespeichert. Jede dieser Dateien (mit Ausnahme der letzten) enthält einen DB2-FORTSETZUNGSSATZ (CONTINUATION RECORD oder kurz "AC"-Satz), der angibt, dass die Dateien logisch geteilt wurden und wo die

nächste Datei zu finden ist. Diese Dateien können auf ein AIX-System übertragen und dort von den Dienstprogrammen IMPORT und LOAD gelesen werden. Wenn das Dienstprogramm EXPORT über ein AIX-System aufgerufen wird, werden keine mehrteiligen PC/IXF-Dateien erstellt. Ausführliche Angaben zur Verwendung finden Sie in den Abschnitten über die Befehle **IMPORT** und **LOAD**.

- 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 Subselect-Anweisungen nur dadurch ausgedrückt werden, dass der Name der Zieltabelle und die Klausel **WHERE** angegeben wird. Beim Exportieren einer Hierarchie können Fullselect und *select-anweisung* nicht angegeben werden.
- Bei anderen Dateiformaten als IXF empfiehlt es sich, die Traversierfolgeliste anzugeben. Sie teilt DB2 mit, wie die Hierarchie durchquert werden soll 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.
- Während einer Importoperation sollte die gleiche Traversierfolge verwendet werden. Das Dienstprogramm LOAD unterstützt das Laden von Hierarchien oder Unterhierarchien nicht.
- Beim Exportieren von Daten aus einer Tabelle mit geschützten Zeilen begrenzen die LBAC-Berechtigungs-nachweise für die Berechtigungs-ID der Sitzung möglicherweise die Zeilen, die exportiert werden. Zeilen, auf die die Berechtigungs-ID der Sitzung keinen Lesezugriff hat, werden nicht exportiert. Es wird keine Fehlermeldung oder Warnung ausgegeben.
- Wenn die LBAC-Berechtigungs-nachweise das Lesen von einzelnen Spalten, die exportiert werden sollen, nicht zulässt, schlägt die Exportoperation fehl, und es wird der Fehler SQLSTATE 42512 zurückgegeben.
- Bei der Ausführung von Dienstprogrammen für das Versetzen von Daten wie **export** und **db2move** stellt der Abfragecompiler möglicherweise fest, dass die zugrunde liegende Abfrage für eine MQT (Materialized Query Table, gespeicherte Abfragetabelle) effizienter ausgeführt wird als für die Basistabelle(n). In diesem Fall wird die Abfrage für eine als REFRESH DEFERRED deklarierte MQT ausgeführt und das Ergebnis der Dienstprogramme zeigt die Daten in der zugrunde liegenden Tabelle möglicherweise nicht exakt an.
- Exportpakete werden entsprechend dem Format DATETIME ISO gebunden. Datums-, Zeit- und Zeitmarkenwerte werden dementsprechend beim Umsetzen in eine Zeichenfolgedarstellung in das ISO-Format konvertiert. Da die CLP-Pakete mit dem Format DATETIME LOC (länderspezifisches Format) gebunden werden, kann es zu inkonsistentem Verhalten zwischen CLP und Exportoperation kommen, wenn das DATETIME-Format von CLP nicht mit dem ISO-Format übereinstimmt. Dazu ein Beispiel: Die folgende Anweisung SELECT gibt die erwarteten Ergebnisse zurück:

```
db2 select col2 from tab1 where char(col2)='05/10/2005';
      COL2
-----
      05/10/2005
      05/10/2005
      05/10/2005
3 Satz/Sätze ausgewählt.
```

Ein Exportbefehl mit derselben Klausel SELECT liefert jedoch nicht die erwarteten Ergebnisse:

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

Wenn Sie das Datumsformat für die Ländereinstellung (LOCALE) durch das ISO-Format ersetzen, werden die erwarteten Ergebnisse zurückgegeben:

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

Änderungswerte für Dateityp für das Dienstprogramm EXPORT

Tabelle 7. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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. Wenn 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 <i>db2exp.001</i> und ist 456 Byte lang.</p> <p>Wenn Sie den Änderungswert lobsinfile bei Verwendung des Befehls EXPORT angeben, werden die LOB-Daten an den in der Klausel LOBS TO angegebenen Positionen platziert. Andernfalls werden die LOB-Daten an das Datendateiverzeichnis 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). Die Option LOBS TO bzw. LOBFILE aktiviert implizit das Verhalten LOBSINFILE.</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>
xmlinsefiles	Jede XDM-Instanz (XDM = XQuery Data Model) wird in eine separate Datei geschrieben. Standardmäßig werden mehrere Werte in derselben Datei miteinander verknüpft.
lobsinsefiles	Jeder LOB-Wert wird in eine separate Datei geschrieben. Standardmäßig werden mehrere Werte in derselben Datei miteinander verknüpft.
xmlnodeclaration	XDM-Instanzen werden ohne XML-Deklarationstag geschrieben. Standardmäßig werden XDM-Instanzen mit einem vorangestellten XML-Deklarationstag exportiert, das ein Codierungsattribut enthält.
xmlchar	XDM-Instanzen werden in der Zeichencodierung geschrieben. Dabei ist zu beachten, dass die Zeichencodierung dem Wert entspricht, der für den Änderungswert für Dateityp <i>codepage</i> angegeben ist, oder der Codepage der Anwendung, wenn dieser Wert nicht angegeben ist. Standardmäßig werden XDM-Instanzen in Unicode exportiert.

Tabelle 7. Gültige Änderungswerte für Dateityp für das Dienstprogramm EXPORT: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
xmlgraphic	Bei Angabe des Änderungswerts xmlgraphic mit dem Befehl EXPORT wird das exportierte XML-Dokument in der Codepage UTF-16 codiert, unabhängig von der Codepage der Anwendung oder dem Änderungswert codepage für den Dateityp.

Tabelle 8. Gültige Änderungswerte für Dateityp für das Dienstprogramm EXPORT: DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer)

Änderungswert	Beschreibung
chardelx	<p>x ist ein einzelner Zeichenfolgebegrenzer. Der Standardwert ist ein doppeltes Anführungszeichen ("). Das angegebene Zeichen wird statt doppelter Anführungszeichen verwendet, um eine Zeichenfolge einzuschließen.² Falls Sie die doppelten Anführungszeichen explizit als Zeichenfolgebegrenzer angeben wollen, müssen Sie dies folgendermaßen tun:</p> <pre>modified by charde1""</pre> <p>Auch das einfache Anführungszeichen (') kann wie folgt als Zeichenfolgebegrenzer festgelegt werden:</p> <pre>modified by charde1''</pre>
codepage=x	<p>x ist eine ASCII-Zeichenfolge. Der Wert wird als Codepage der Daten im Ausgabedatensatz gewertet. Während der Exportoperation werden Zeichendaten aus der Anwendungscodepage 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. Der Änderungswert codepage kann nicht zusammen mit dem Änderungswert lobsinfile verwendet werden.</p>
coldelx	<p>x ist ein einzelnes Zeichen, das als Spaltenbegrenzer verwendet wird. Der Standardwert ist ein Komma (.). Das angegebene Zeichen wird anstelle eines Kommas verwendet, um das Ende einer Spalte anzugeben.²</p> <p>Im folgenden Beispiel bewirkt coldel;, dass das Dienstprogramm EXPORT das Semikolon (;) als Spaltenbegrenzungszeichen für die exportierten Daten verwendet:</p> <pre>db2 "export to temp of del modified by coldel; select * from staff where dept = 20"</pre>
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	x ist ein einzelnes Ersatzzeichen für den Punkt als Dezimalzeichen. Der Standardwert ist ein Punkt (.). Das angegebene Zeichen wird statt eines Punkts als Dezimalzeichen verwendet. ²
nochardel	<p>Spaltendaten werden nicht von Zeichenbegrenzern eingeschlossen. Diese Option sollte nicht angegeben werden, wenn die Daten mithilfe von DB2 importiert oder geladen werden sollen. 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 charde1x oder nodoublede1 angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p>
nodoublede1	Unterdrückt die Erkennung von doppelten Zeichenbegrenzern. ²

Tabelle 8. Gültige Änderungswerte für Dateityp für das Dienstprogramm EXPORT: DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer) (Forts.)

Änderungswert	Beschreibung
timestampformat="x"	<p>x ist das Format der Zeitmarke in der Quellendatei.⁴ Gültige Zeitmarkenelemente 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 01 bis 12; dieses Element und das Element M schließen sich gegenseitig aus) 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) D - Tag (eine oder zwei Ziffern von 1 bis 31) DD - Tag (zwei Ziffern von 01 bis 31; dieses Element und 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) H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System) HH - Stunde (zwei Ziffern von 00 bis 12 bei einem 12-Stunden-System und 00 bis 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus) M - Minute (ein oder zwei Ziffern von 0 bis 59) MM - Minute (zwei Ziffern von 00 bis 59; dieses Element und das Minutenelement M schließen sich gegenseitig aus) S - Sekunde (ein oder zwei Ziffern von 0 bis 59) SS - Sekunde (zwei Ziffern von 00 bis 59; dieses Element und das Element S schließen sich gegenseitig aus) SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86400; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus) U (1 bis 12 mal) - Bruchsekunden (die Häufigkeit von U gibt die Anzahl der Stellen mit Werten aus dem Bereich 0 bis 9 an) TT - Meridiananzeiger (AM oder PM)</p> <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 9. Gültige Änderungswerte für Dateityp für das Dienstprogramm EXPORT: IXF-Dateiformat

Änderungswert	Beschreibung
codepage=x	x ist eine ASCII-Zeichenfolge. Der Wert wird als Codepage der Daten im Ausgabedatensatz gewertet. Während der Exportoperation werden Zeichendaten aus dieser Codepage in die Codepage der Anwendung konvertiert. Bei reinem DBCS (Grafik), gemischtem DBCS und EUC sind Begrenzer auf den Bereich von x00 bis x3F (jeweils einschließlich) beschränkt.

Tabelle 10. Gültige Änderungswerte für Dateityp für das Dienstprogramm EXPORT: WSF-Dateiformat⁶

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

Anmerkung:

1. Das Dienstprogramm EXPORT gibt keine Warnung aus, 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.
2. *Aspekte beim Versetzen von XML-Daten - Begrenzer* listet Einschränkungen für Zeichen auf, die anstelle der standardmäßigen Begrenzer verwendet werden können.
3. 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.

4. 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)

5. Diese Dateien können auch einem bestimmten Produkt zugeordnet werden, indem ein L für Lotus 1-2-3 oder ein S für Symphony in der Zeichenfolge des Parameters *dateitypänderungswert* angegeben wird. Es kann nur ein einziger Wert bzw. Produktbezeichner angegeben werden. Die Unterstützung für das WSF-Dateiformat ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.
6. Das WSF-Format wird für XML-Spalten nicht unterstützt. Die Unterstützung für dieses Dateiformat ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.
7. Alle XDM-Instanzen werden in separate XML-Dateien geschrieben (nicht in die Hauptdatendatei). Dies gilt auch, wenn weder die Klausel **XMLFILE** noch die Klausel **XML TO** angegeben wird. Standardmäßig werden XML-Dateien in den Pfad der exportierten Datendatei geschrieben. Der Standardwert für den Basisdateinamen für XML-Dateien ist der Name der exportierten Datendatei mit der Erweiterung ".xml".
8. Alle XDM-Instanzen werden standardmäßig mit einer vorangestellten XML-Deklaration exportiert, die ein Codierungsattribut enthält, sofern nicht der Änderungswert für den Dateityp XMLNODEDECLARATION angegeben ist.
9. Standardmäßig werden alle XDM-Instanzen im Unicode-Format exportiert, sofern nicht der Änderungswert für den Dateityp XMLCHAR oder XMLGRAPHIC angegeben ist.
10. Der Standardpfad für XML-Daten und LOB-Daten ist der Pfad der Hauptdatendatei. Der Standardbasisname für XML-Dateien ist der Name der Hauptdatendatei. Der Standardbasisname für LOB-Dateien ist der Name der Hauptdatendatei. Bei einer Hauptdatendatei mit dem Namen
 /mypath/myfile.del
 ist der Standardpfad für XML-Daten und LOB-Daten z. B.
 /mypath"
 Der Standardbasisname für XML-Dateien ist
 myfile.del
 und der Standardbasisname für LOB-Dateien ist
 myfile.del
 Der Änderungswert für Dateityp LOBSINFILE muss angegeben sein, damit LOB-Dateien generiert werden.
11. Vom Dienstprogramm EXPORT wird eine numerische Kennung an die einzelnen LOB- bzw. XML-Dateien angehängt. Die Kennung ist ein dreistelliger, mit Nullen aufgefüllter Folgewert, der mit folgenden Zeichen Kennung beginnt:
 .001

Nach der 999. LOB- bzw. XML-Datei wird die Kennung nicht mehr mit Nullen aufgefüllt (die 1000. LOB- bzw. XML-Datei hat z. B. die folgende Kennung:
 .1000

Auf die numerische Kennung folgt eine aus drei Zeichen bestehende Typenkennung für den Datentyp:

.lob

oder

.xml

Eine generierte LOB-Datei wird z. B. mit einem Namen im folgenden Format gekennzeichnet:

myfile.del.001.lob

Eine generierte XML-Datei wird z. B. mit einem Namen im folgenden Format gekennzeichnet:

myfile.del.001.xml

12. Mit dem Dienstprogramm EXPORT können auch XDM-Instanzen exportiert werden, die keine korrekt formatierten Dokumente sind, wenn ein XQuery-Ausdruck angegeben wird. Sie können die exportierten Dokumente anschließend jedoch nicht direkt in eine XML-Spalte importieren bzw. laden, da XML-Spalten nur vollständige Dokumente enthalten können.

db2Export - Exportieren von Daten aus einer Datenbank

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 Berechtigungen:

- Berechtigung *DATAACCESS*
- Zugriffsrecht *CONTROL* oder *SELECT* für jede betroffene Tabelle oder Sicht

LBAC (Label-Based Access Control) wird für diese Funktion erzwungen. Die exportierten Daten werden unter Umständen durch die LBAC-Berechtigungsanweisung des Aufrufers eingeschränkt, wenn die Daten durch LBAC geschützt werden.

Erforderliche Verbindung

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

API-Include-Datei

db2ApiDf.h

API- und Datenstruktursyntax

```
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;
    struct db2ExportIn *piExportInfoIn;
    struct sqlu_media_list *piXmlPathList;
    struct sqlu_media_list *piXmlFileList;
} db2ExportStruct;

typedef SQL_STRUCTURE db2ExportIn
{
    db2Uint16 *piXmlSaveSchema;
} db2ExportIn;

typedef SQL_STRUCTURE db2ExportOut
{
    db2Uint64 oRowsExported;
} db2ExportOut;

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;
    struct db2ExportIn *piExportInfoIn;
    struct sqlu_media_list *piXmlPathList;
    struct sqlu_media_list *piXmlFileList;
} db2gExportStruct;

```

Parameter der API 'db2Export'

versionNumber

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

pParmStruct

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

pSqlca

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

Parameter der db2ExportStruct-Datenstruktur

piDataFileName

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

piLobPathList

Eingabe. Ein Zeiger auf eine 'sqlu_media_list'-Struktur, wobei das Feld 'media_type' auf SQLU_LOCAL_MEDIA gesetzt ist und die 'sqlu_media_entry'-Struktur Pfade auf dem Client angibt, in denen die LOB-Dateien gespeichert werden sollen. Exportierte LOB-Daten werden gleichmäßig auf alle Pfade verteilt, die in der 'sqlu_media_entry'-Struktur aufgeführt sind.

piLobFileList

Eingabe. Ein Zeiger auf eine 'sqlu_media_list'-Struktur, wobei das Feld 'media_type' auf SQLU_CLIENT_LOCATION gesetzt ist und die 'sqlu_location_entry'-Struktur 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 'piLobPathList') und daran wiederum eine dreistellige Folgenummer und die Erweiterung .lob angefügt werden. Beispiel: Wenn der aktuelle LOB-Pfad das Verzeichnis '/u/foo/lob/path' ist, der aktuelle LOB-Dateiname 'bar' lautet und der Änderungswert für Dateityp LOBSINSEPFILLES gesetzt ist, dann heißen die erstellten LOB-Dateien '/u/foo/LOB/path/bar.001.lob', '/u/foo/LOB/path/bar.002.lob' usw. Ist der Änderungswert für Dateityp LOBSINSEPFILLES nicht gesetzt, werden alle LOB-Dokumente verknüpft und in eine Datei ('/u/foo/lob/path/bar.001.lob') gestellt.

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 interpretiert werden. Gültige Werte für diesen Parameter (gemäß Definition in der Kopfdatendatei 'sqlutil' im INCLUDE-Verzeichnis) 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 in piActionString angegebenen SELECT-Anweisung abgeleitet.

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 (gemäß Definition in der Kopfdatendatei 'sqlutil'):

SQL_DEL

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

SQL_WSF

Arbeitsblattformate (WSF = Worksheet Formats) für den Datenaustausch mit Lotus Symphony- und 1-2-3-Programmen. Die Unterstützung für dieses Dateiformat ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.

SQL_IXF

PC-Version des IXF-Formats (Integration Exchange Format). Dies ist das bevorzugte Format für den Datenexport aus einer Tabelle. In diesem Dateiformat 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 Verarbeitungsoption(en) 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. Vgl. den zugehörigen Abschnitt unten mit Informationen zu den Änderungswerten für Dateityp für das Dienstprogramm EXPORT.

piMsgFileName

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

iCallerAction

Eingabe. Eine vom aufrufenden Programm angeforderte Aktion. Gültige Werte (gemäß Definition in der Kopfdatendatei 'sqlutil' im INCLUDE-Verzeichnis) 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 Erstaufwurf zurückgegeben wurde und das Dienstprogramm eine Benutzereingabe anfordert (z. B. um auf eine Bandendebingung 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 Erstaufwurf zurückgegeben wurde und das Dienstprogramm eine Benutzereingabe anfordert (z. B. um auf eine Bandendebingung 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.

piExportInfoIn

Eingabe. Ein Zeiger auf die 'db2ExportIn'-Struktur.

piXmlPathList

Eingabe. Ein Zeiger auf eine 'sqlu_media_list'-Struktur, wobei das Feld 'media_type' auf SQLU_LOCAL_MEDIA gesetzt ist und die 'sqlu_media_entry'-Struktur Pfade auf dem Client angibt, in denen die XML-Dateien gespeichert werden sollen. Exportierte XML-Daten werden gleichmäßig auf alle Pfade verteilt, die in der 'sqlu_media_entry'-Struktur aufgeführt sind.

piXmlFileList

Eingabe. Ein Zeiger auf eine 'sqlu_media_list'-Struktur, wobei das Feld 'media_type' auf SQLU_CLIENT_LOCATION gesetzt ist und die 'sqlu_location_entry'-Struktur 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 XML-Dateien während einer Exportoperation werden die Dateinamen erzeugt, indem der aktuelle Basisdateiname aus dieser Liste an den aktuellen Pfad (aus 'piXmlFileList') und daran wiederum eine dreistellige Folgenummer und die Erweiterung .xml angefügt werden. Beispiel: Wenn der aktuelle XML-Pfad das Verzeichnis '/u/foo/xml/path' ist, der aktuelle XML-Dateiname 'bar' lautet und der Änderungswert für Dateityp XMLINSEPFILES gesetzt ist, dann heißen die erstellten XML-Dateien '/u/foo/xml/path/bar.001.xml', '/u/foo/xml/path/bar.002.xml' usw. Ist der Änderungswert für Dateityp XMLINSEPFILES nicht gesetzt, werden alle XML-Dokumente verknüpft und in eine Datei ('/u/foo/xml/path/bar.001.xml') gestellt.

Parameter der db2ExportIn-Datenstruktur

piXmlSaveSchema

Eingabe. Gibt an, dass die SQL-Kennung des XML-Schemas, das zur Prüfung der einzelnen exportierten XML-Dokumente verwendet wird, in der exportierten Datendatei gespeichert werden soll. Gültige Werte sind TRUE und FALSE.

Parameter der db2ExportOut-Datenstruktur

oRowsExported

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

Parameter der db2gExportStruct-Datenstruktur

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.

Hinweise zur Verwendung

Vor dem Start einer Exportoperation müssen alle Tabellenoperationen vollständig ausgeführt und alle Sperren freigegeben werden. Hierfür gibt es folgende Möglichkeiten:

- Schließen aller geöffneten Cursor, die mit der Klausel WITH HOLD definiert wurden, und Festschreiben der Datenänderungen durch Ausführen der Anweisung COMMIT.
- Rückgängigmachen der Datenänderungen durch Ausführen der Anweisung ROLLBACK.

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.

Gibt das Dienstprogramm EXPORT Warnungen aus, wird die Nachricht in eine Nachrichtendatei geschrieben bzw. an die Standardausgabe gesendet, sofern keine Nachrichtendatei angegeben wurde.

Eine Warnung wird ausgegeben, wenn die Anzahl der Spalten (Feld 'dcolnum' der 'sqldcol'-Struktur) in der externen Spaltennamenmatrix 'piDataDescriptor' nicht mit der Anzahl der Spalten identisch ist, die durch die Anweisung SELECT 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 .bnd-Dateien manuell gebunden werden, darf die Option 'format' für den Binder nicht verwendet werden.

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

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.

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

Indexdefinitionen für eine Tabelle sind in der PC/IXF-Datei enthalten, wenn der Inhalt einer einzelnen Datenbanktabelle in eine PC/IXF-Datei exportiert wird, wobei der Parameter 'piActionString' mit 'SELECT * FROM tabellenname' beginnt und der Parameter 'piDataDescriptor' Standardnamen angibt. Indizes werden für Sichten nicht gesichert. Gleiches gilt, wenn die Klausel SELECT von 'piActionString' einen Join enthält. Eine Klausel WHERE, eine Klausel GROUP BY oder eine Klausel HAVING im Parameter '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 Subselect-Anweisungen nur dadurch ausgedrückt werden, dass der Name der Zieltabelle und die Klausel WHERE angegeben wird. Beim Exportieren einer Hierarchie können Fullselect und select-anweisung nicht angegeben werden.

Bei anderen Dateiformaten als IXF empfiehlt es sich, die Traversierfolgeliste anzugeben. Sie teilt DB2 mit, wie die Hierarchie durchquert werden soll 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.

Syntax der API REXX

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

```
CONTINUE EXPORT
```

```
STOP EXPORT
```

Parameter der API REXX

stmt Eine REXX-Hostvariable, 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.

dateityp

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

- DEL** ASCII-Format ohne universelle Begrenzer.
- WSF** Arbeitsblattformate (Worksheet Formats). Die Unterstützung für dieses Dateiformat ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.
- IXF** PC-Version des IXF-Formats (Integration 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.

Exportsitzungen - Beispiele für CLP

Beispiel 1

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
```

Beispiel 2

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
```

Beispiel 3

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
```

Beispiel 4

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
```

Beispiel 5

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 char del'' coldel; decpt,  
select * from staff
```

Dienstprogramm IMPORT

Dienstprogramm IMPORT - Übersicht

Das Dienstprogramm IMPORT füllt Tabellen, typisierte Tabellen und Sichten mit Daten, die über eine SQL-Anweisung INSERT eingefügt werden. Wenn die Tabelle oder Sicht, die die importierten Daten aufnehmen soll, bereits Daten enthält, können diese Daten durch die importierten Daten ersetzt oder durch sie ergänzt werden.

Bei dem Dienstprogramm IMPORT handelt es sich wie bei dem Dienstprogramm EXPORT um ein einfaches Dienstprogramm zum Versetzen von Daten. Das Dienstprogramm kann über die Steuerzentrale, durch die Ausgabe von CLP-Befehlen, einen Aufruf der gespeicherten Prozedur ADMIN_CMD oder einen Aufruf der API db2Import über eine Benutzeranwendung aktiviert werden.

Das Dienstprogramm IMPORT unterstützt eine Reihe von Datenformaten und Funktionen für den Import:

- Datenformate IXF, WSF, ASC und DEL
- Verwendung von Änderungswerten für den Dateityp zum Anpassen der Importoperation
- Versetzen hierarchischer Daten und typisierter Tabellen
- Protokollieren aller Aktivitäten, Aktualisieren von Indizes, Überprüfen von Integritätsbedingungen und Auslösen von Triggern
- Angabe von Spaltennamen für die Tabellen oder Sichten, in die die Daten eingefügt werden sollen
- Verwendung mit DB2 Connect

Wichtig: Die Unterstützung für das WSF-Dateiformat ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.

Importmodus

Als Importmodus sind fünf unterschiedliche Betriebsarten verfügbar, die die beim Datenimport verwendete Methode bestimmen. Die ersten drei Betriebsarten, INSERT, INSERT_UPDATE und REPLACE, werden verwendet, wenn die Zieltabellen bereits vorhanden sind. Diese Betriebsarten unterstützen die Datenformate IXF, WSF, ASC und DEL. Mit Kurznamen können jedoch nur die Betriebsarten INSERT und INSERT_UPDATE verwendet werden.

Tabelle 11. Übersicht über die Importmodi INSERT, INSERT_UPDATE und REPLACE

Modus	Empfohlene Methode für die Verwendung
INSERT	Fügt die Eingabedaten in die Zieltabelle ein, ohne die vorhandenen Daten zu ändern.
INSERT_UPDATE	Aktualisiert Zeilen mit übereinstimmenden Primärschlüsselwerten mit den Werten von Eingabezeilen. Ist keine Zeilenentsprechung vorhanden, wird die importierte Zeile in die Tabelle eingefügt.
REPLACE	Löscht alle vorhandenen Daten und fügt importierte Daten unter Beibehaltung von Tabellen- und Indexdefinitionen ein.

Die übrigen Betriebsarten, REPLACE_CREATE und CREATE, werden verwendet, wenn es sich bei den Zieltabellen um neue Tabellen handelt. Diese Betriebsarten können nur für Eingabedateien im Format PC/IXF verwendet werden, bei dem eine strukturierte Beschreibung der jeweils zu erstellenden Tabelle enthalten ist. Importe mit diesen Betriebsarten können nicht durchgeführt werden, wenn die Objekttableau außer sich selbst noch andere abhängige Tabellen aufweist.

Anmerkung: Die Importbetriebsarten CREATE und REPLACE_CREATE sind veraltet. Verwenden Sie stattdessen das Dienstprogramm **db2look**.

Tabelle 12. Übersicht über die Importmodi REPLACE_CREATE und CREATE

Modus	Empfohlene Methode für die Verwendung
REPLACE_CREATE	Löscht alle vorhandenen Daten und fügt importierte Daten unter Beibehaltung von Tabellen- und Indexdefinitionen ein. Erstellt Zieltabelle und Index, soweit noch nicht vorhanden.
CREATE	Erstellt Zieltabelle und Index. Sie können den Namen des Tabellenbereichs für die neue Tabelle angeben.

Vorgehensweise beim Import

Die Anzahl der für den Import erforderlichen Schritte und die Dauer des Importvorgangs richtet sich nach dem Umfang der zu versetzenden Daten und den von Ihnen angegebenen Optionen. Der Importvorgang beinhaltet folgende Schritte:

1. Sperren von Tabellen
Beim Import wird, je nachdem ob Sie einen gleichzeitigen Tabellenzugriff zulassen, eine exklusive (X) oder nicht exklusive Sperre (IX) für vorhandene Zieltabellen angefordert.

2. Suchen und Abrufen von Daten
Beim Import wird die Klausel FROM zum Suchen der Eingabedaten verwendet. Wenn Sie im Befehl angegeben haben, dass XML- oder LOB-Daten vorhanden sind, werden diese Daten beim Import gesucht.
3. Einfügen von Daten
Beim Import werden vorhandene Daten ersetzt oder neue Datenzeilen an eine vorhandene Tabelle angehängt.
4. Prüfung auf Integritätsbedingungen und Auslösen von Triggern
Beim Schreiben der Daten stellt das Dienstprogramm IMPORT sicher, dass die einzelnen eingefügten Zeilen jeweils den für die Zieltabelle definierten Integritätsbedingungen entsprechen. Angaben zu zurückgewiesenen Zeilen werden in die Nachrichtendatei geschrieben. Beim Import werden darüber hinaus auch vorhandene Trigger ausgelöst.
5. Commit der Operation
Das Dienstprogramm IMPORT speichert die vorgenommenen Änderungen und gibt die Sperren der Zieltabelle frei. Sie können auch angeben, dass beim Import in regelmäßigen Abständen ein Commit durchgeführt werden soll.

Für eine einfache Importoperation sind folgende Angaben erforderlich:

- Pfad und Name der Eingabedatei
- Name oder Aliasname der Zieltabelle oder -sicht
- Format der Daten in der Eingabedatei
- Für den Datenimport zu verwendende Methode
- Traversierfolge (beim Import hierarchischer Daten)
- Liste untergeordneter Tabellen (beim Import typisierter Tabellen)

Weitere Optionen

Sie können die Importoperationen mit einer Reihe von Parametern anpassen. Für die Klausel MODIFIED BY können Sie Änderungswerte für den Dateityp angeben, um das Datenformat zu ändern, um dem Dienstprogramm anzugeben, wie mit den Daten zu verfahren ist, und um die Leistungswerte zu optimieren.

Vom Dienstprogramm IMPORT werden standardmäßig bis zum erfolgreichen Abschluss des Imports keine Commitoperationen durchgeführt. Ausgenommen sind lediglich bestimmte, mit ALLOW WRITE ACCESS definierte Importoperationen. Dies erhöht die Importgeschwindigkeit. Es kann jedoch für den gemeinsamen Zugriff, Neustartvorgänge und den aktiven Protokollspeicherbereich sinnvoll sein, eine Durchführung von Commitoperationen auch während des Imports zu definieren. Dies kann erreicht werden, indem der Parameter **COMMITCOUNT** auf "automatic" gesetzt wird. Bei Angabe dieses Werts ermittelt das Importprogramm intern, wann ein Commit durchgeführt werden sollte. Sie können jedoch stattdessen auch eine bestimmte Zahl für den Parameter **COMMITCOUNT** definieren, um anzugeben, dass ein Commit nach dem Import der angegebenen Anzahl von Datensätzen durchgeführt werden soll.

Die Importleistung kann auf verschiedene Arten optimiert werden. Da es sich bei dem Dienstprogramm IMPORT um eine integrierte SQL-Anwendung handelt, die SQL-Abrufe intern durchführt, profitiert das Dienstprogramm IMPORT auch von Optimierungen, die bei den SQL-Operationen vorgenommen werden. Mit dem Änderungswert compound für den Dateityp können Sie angeben, dass eine bestimmte Anzahl von Zeilen gleichzeitig eingefügt werden soll, anstatt beim Einfügen zeilenweise vorzugehen. Wenn Sie davon ausgehen, dass beim Import voraussichtlich eine große Anzahl von Warnungen generiert wird (mit der daraus resultierenden

Verlangsamung des Imports), können Sie den Änderungswert `norowwarnings` für den Dateityp angeben, um Warnungen zu zurückgewiesenen Zeilen zu unterdrücken.

Nachrichtendatei

Beim Import werden ASCII-Standardtextnachrichtendateien für die zugehörigen Fehlernachrichten, Informationsnachrichten oder Warnungen geschrieben. Wird das Dienstprogramm über die API `db2Import` aufgerufen, ist eine vorherige Angabe des Dateinamens für diese Dateien über den Parameter **MESSAGES** erforderlich. Andernfalls ist diese Angabe optional. Die Nachrichtendatei für eine Importoperation kann während des Imports aufgerufen werden und ermöglicht dadurch eine einfache Überwachung des Verarbeitungsfortschritts. Sollten Importoperationen fehlschlagen, kann anhand der Angabe zur letzten erfolgreich importierten Zeile in den Nachrichtendateien ermittelt werden, an welcher Stelle ein Wiederanlauf ansetzen muss.

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

Für das Dienstprogramm IMPORT erforderliche Zugriffsrechte und Berechtigungen

Zugriffsrechte ermöglichen es Benutzern, Datenbankressourcen zu erstellen und auf diese zuzugreifen. Berechtigungsstufen stellen eine Methode dar, um Berechtigungen sowie übergeordnete Pflege- und Dienstprogrammoperationen des Datenbankmanagers zusammenzufassen. Sie dienen zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte.

Benutzer können nur auf die Objekte zugreifen, für die sie zugriffsberechtigt sind, d. h. für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigung verfügen.

Benutzer mit der Berechtigung `DATAACCESS` können alle Typen von Importoperationen ausführen. In der nachfolgenden Tabelle werden weitere Berechtigungen für beim Import beteiligte Tabellen, Sichten und Kurznamen aufgeführt, die zum Ausführen des jeweiligen Importtyps erforderlich sind.

Tabelle 13. Zum Ausführen von Importoperationen erforderliche Berechtigungen

Modus	Erforderliche Berechtigung
INSERT	CONTROL oder INSERT und SELECT
INSERT_UPDATE	CONTROL oder INSERT, SELECT, UPDATE und DELETE
REPLACE	CONTROL oder INSERT, SELECT und DELETE
REPLACE_CREATE	Wenn die Zieltabelle bereits vorhanden ist: CONTROL oder INSERT, SELECT und DELETE Wenn die Zieltabelle noch nicht vorhanden ist: CREATETAB (für die Datenbank), USE (für den Tabellenbereich) und wenn das Schema nicht vorhanden ist: IMPLICIT_SCHEMA (für die Datenbank) oder wenn das Schema vorhanden ist: CREATEIN (für das Schema)

Tabelle 13. Zum Ausführen von Importoperationen erforderliche Berechtigungen (Forts.)

Modus	Erforderliche Berechtigung
CREATE	CREATETAB (für die Datenbank), USE (für den Tabellenbereich) und wenn das Schema nicht vorhanden ist: IMPLICIT_SCHEMA (für die Datenbank) oder wenn das Schema vorhanden ist: CREATEIN (für das Schema)

Anmerkung: Die Optionen **CREATE** und **REPLACE_CREATE** des Befehls **IMPORT** sind veraltet und werden in zukünftigen Releases möglicherweise entfernt. Zum Verwenden der Option **REPLACE** oder **REPLACE_CREATE** für eine Tabelle muss die Berechtigungs-ID der Sitzung über die Berechtigung zum Löschen der jeweiligen Tabelle verfügen.

Wenn Sie eine Hierarchie importieren möchten, richtet sich die erforderliche Berechtigung auch nach dem Modus. Bei vorhandenen Hierarchien reicht das Zugriffsrecht **CONTROL** für alle untergeordneten Tabellen in der Hierarchie für eine Operation **REPLACE** aus. Bei nicht vorhandenen Hierarchien reicht die Berechtigung **CONTROL** für alle untergeordneten Tabellen in der Hierarchie in Verbindung mit der Berechtigung **CREATETAB** und **USE** für eine Operation **REPLACE_CREATE** aus.

Darüber hinaus müssen beim Importieren in Tabellen mit kennsatzbasierter Zugriffssteuerung (LBAC) bestimmte Aspekte berücksichtigt werden. Um Daten in eine Tabelle importieren zu können, die geschützte Spalten enthält, muss die Berechtigungs-ID der Sitzung über LBAC-Berechtigungenachweise verfügen, die Schreibzugriffe auf alle geschützten Spalten in der Tabelle zulassen. Um Daten in eine Tabelle importieren zu können, die geschützte Zeilen enthält, muss der Berechtigungs-ID der Sitzung mit Grant ein Sicherheitskennsatz für Schreibzugriff erteilt werden, der Teil der Sicherheitsrichtlinie zum Schutz der Tabelle ist.

Importieren von Daten

Das Dienstprogramm **IMPORT** fügt Daten aus einer externen Datei mit einem unterstützten Dateiformat in eine Tabelle, eine Hierarchie, eine Sicht oder einen Kurznamen ein.

Das Dienstprogramm **LOAD** ist eine schnellere Alternative, unterstützt jedoch nicht das Laden von Daten auf Hierarchieebene.

Vorbereitende Schritte

Bevor Sie das Dienstprogramm **IMPORT** ausführen, müssen Sie mit der Datenbank, in die die Daten importiert werden sollen, verbunden bzw. in der Lage sein, implizit eine Verbindung zur Datenbank herzustellen. Wenn das implizite Herstellen von Verbindungen aktiviert ist, wird eine Verbindung zur Standarddatenbank hergestellt. Beim Zugriff des Dienstprogramms auf DB2 for Linux-, UNIX- oder Windows-Datenbankserver über DB2 for Linux-, UNIX- oder Windows-Clients muss es sich um eine Direktverbindung über die Steuerkomponente handeln. Eine Verbindung über ein DB2 Connect-Gateway oder eine Loopback-Umgebung ist nicht zulässig. Da das Dienstprogramm die Anweisung **COMMIT** oder **ROLLBACK** absetzt, beenden Sie vor dem Aufrufen von **IMPORT** alle Transaktionen und geben alle Sperren frei, indem Sie die Anweisung **COMMIT** absetzen oder eine **ROLLBACK**-Operation ausführen.

Anmerkung: Die Parameter **CREATE** und **REPLACE_CREATE** des Befehls **IMPORT** sind veraltet und werden in zukünftigen Releases möglicherweise entfernt.

Informationen zu diesem Vorgang

Für das Dienstprogramm **IMPORT** gelten die folgenden Einschränkungen:

- 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 zugrunde liegende Tabelle einer MQT (Materialized Query Table, gespeicherte 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 das Dienstprogramm **IMPORT** eigene SQL-Anweisungen generiert, kann die maximale Anweisungsgröße von 2 MB möglicherweise in einigen Fällen überschritten werden.
- Sie können eine partitionierte Tabelle oder eine MDC-Tabelle (Multidimensional Clustering Table - mehrdimensionale Clusteringtabelle) mit den Importparameter **CREATE** oder **REPLACE_CREATE** nicht erneut erstellen.
- Tabellen, die XML-Spalten enthalten, können nicht erneut erstellt werden.
- Es können keine verschlüsselten Daten importiert werden.
- Die Operation **IMPORT REPLACE** berücksichtigt die Klausel **NOT LOGGED INITIALLY** nicht. Der Parameter **REPLACE** des Befehls **IMPORT** berücksichtigt nicht die Klausel **NOT LOGGED INITIALLY (NLI)** für die Anweisung **CREATE TABLE** statement oder die Klausel **ACTIVATE NOT LOGGED INITIALLY** für die Anweisung **ALTER TABLE**. Wenn innerhalb einer Transaktion mit einer Anweisung **CREATE TABLE** oder **ALTER TABLE**, bei der die Klausel **NLI** aufgerufen wird, ein Import mit der Aktion **REPLACE** vorgenommen wird, wird die Klausel **NLI** beim Import nicht berücksichtigt. Alle Einfügungen werden protokolliert.

Ausweichmaßnahme 1: Löschen Sie den Inhalt der Tabelle mit der Anweisung **DELETE**. Rufen Sie die Importoperation dann mit der Anweisung **INSERT** auf.

Ausweichmaßnahme 2: Löschen Sie die Tabelle, und erstellen Sie sie erneut. Rufen Sie die Importoperation dann mit der Anweisung **INSERT** auf.

Die folgende Einschränkung gilt für das Dienstprogramm **IMPORT**: 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.

Vorgehensweise

Gehen Sie folgendermaßen vor, um das Dienstprogramm **IMPORT** auszuführen:

- Geben Sie den Befehl **IMPORT** im Befehlszeilenprozessor (CLP) an.

- Rufen Sie die Anwendungsprogrammierschnittstelle (API) db2Import über eine Clientanwendung auf.
- Verwenden Sie IBM Data Studio. Zwar kann auch der Notizbuch **Tabelle importieren** in der Steuerzentrale verwendet werden, es empfiehlt sich jedoch die Verwendung von IBM Data Studio, da die Toolfunktionen der Steuerzentrale nicht weiter unterstützt werden.

Beispiel

Einfache Importoperationen erfordern lediglich die Angabe der Eingabedatei, des Dateiformats, des Importmodus und der Zieldatei (oder den Namen der zu erstellenden Tabelle).

Geben Sie zum Importieren von Daten über den Befehlszeilenprozessor den Befehl **IMPORT** ein:

```
db2 import from dateiname of dateiformat importmodus into tabelle
```

Dabei ist *dateiname* der Name der Eingabedatei mit den zu importierenden Daten, *dateiformat* das Dateiformat (*ixf*), *importmodus* der Modus (*insert*) und *tabelle* der Name der Tabelle, in die die Daten eingefügt werden sollen.

Es kann jedoch auch sinnvoll sein, eine Nachrichtendatei anzugeben, in die Warnungen und Fehlernachrichten geschrieben werden können. Fügen Sie dazu den Parameter **MESSAGES** und den Namen einer Nachrichtendatei ein. Beispiel:

```
db2 import from dateiname of dateiformat messages nachrichtendatei importmodus into tabelle
```

Die vollständige Syntax sowie Informationen zur Verwendung finden Sie im Abschnitt "Befehl IMPORT".

Importieren von XML-Daten

Das Dienstprogramm IMPORT kann zum Importieren von XML-Daten in eine XML-Tabellenspalte eingesetzt werden, indem entweder der Tabellename oder ein Kurzname für ein Quelldatenobjekt von DB2 Database for Linux, UNIX and Windows verwendet wird.

Beim Importieren von Daten in eine XML-Tabellenspalte können Sie die Option XML FROM verwenden, um die Pfade der XML-Eingabedatei oder der XML-Eingabedateien anzugeben. Für die XML-Datei `"/home/user/xmlpath/xmldocs.001.xml"`, die zuvor exportiert wurde, können Sie beispielsweise den folgenden Befehl verwenden, um die Daten zurück in die Tabelle zu importieren:

```
IMPORT FROM t1export.del OF DEL XML FROM /home/user/xmlpath INSERT INTO USER.T1
```

Eingefügte Dokumente anhand von Schemata prüfen

Die Option XMLVALIDATE ermöglicht die Gültigkeitsprüfung von XML-Dokumenten mithilfe von XML-Schemata, während diese importiert werden. Im folgenden Beispiel wird die Gültigkeit eingehender XML-Dokumente auf der Basis der Schemainformationen überprüft, die während des Exports der XML-Dokumente gespeichert wurden:

```
IMPORT FROM t1export.del OF DEL XML FROM /home/user/xmlpath XMLVALIDATE
USING XDS INSERT INTO USER.T1
```

Optionen für das Parsing angeben

Mit der Option XMLPARSE können Sie angeben, ob Leerzeichen in den importierten XML-Dokumenten beim Parsing (d. h. bei der syntaktischen Analyse) beibehalten oder gelöscht werden sollen. Im folgenden Beispiel wird die Gültigkeit aller importierten XML-Dokumente auf der Basis der XML-Schemainformationen überprüft, die gespeichert wurden, als die XML-Dokumente exportiert wurden. Für diese Dokumente werden die Leerzeichen beim Parsing beibehalten.

```
IMPORT FROM tlexport.del OF DEL XML FROM /home/user/xmlpath XMLPARSE PRESERVE
WHITESPACE XMLVALIDATE USING XDS INSERT INTO USER.T1
```

Neuerstellung von importierten Tabellen

Mit dem Modus CREATE des Dienstprogramms IMPORT können Sie eine Tabelle erneut erstellen, die mit dem Dienstprogramm EXPORT gesichert wurde. Dabei sind jedoch verschiedene Einschränkungen zu beachten, da viele Attribute der Eingabetabelle nicht beibehalten werden.

Beim Import können Tabellen nur erneut erstellt werden, wenn die vorangehende Exportoperation bestimmte Anforderungen erfüllt. Die ursprüngliche Tabelle muss in eine IXF-Datei exportiert werden. Wenn Sie Dateien mit dem DEL- oder ASC-Dateiformat exportieren, enthalten die Ausgabedateien zwar die Satzdaten, sie enthalten jedoch keine Beschreibungen. 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 dem Dienstprogramm **db2look** können Sie die ursprünglichen Tabellendefinitionen erfassen und die entsprechende Datendefinitionssprache (DDL) generieren. Darüber hinaus darf die Anweisung SELECT für den Export nur bestimmte Aktionszeichenfolgen enthalten. Es dürfen z. B. keine Spaltennamen in der Klausel SELECT verwendet werden, und es ist nur die Verwendung von "SELECT *" zulässig.

Anmerkung: Der Importmodus CREATE ist veraltet. Verwenden Sie das Dienstprogramm **db2look**, um Ihre Tabellen zu erfassen und erneut zu erstellen.

Beibehaltene Attribute

In der erneut erstellten Tabelle werden folgende Attribute der Originaltabelle beibehalten:

- Name des Primärschlüssels und Definition
- Spalteninformationen, z. B.:
 - Spaltenname
 - Spaltentypen, einschließlich benutzerdefinierte einzigartige Datentypen, die als Basistyp beibehalten werden
 - Identitätsmerkmale
 - Längen (außer bei Typen lob_file)
 - Codepage (sofern zutreffend)
 - Identitätsoptionen
 - Angabe, ob die Spalte laut Definition Nullwerte enthalten darf oder nicht
 - Standardwerte für Konstanten, sofern vorhanden, jedoch keine anderen Arten von Standardwerten
- Indexinformationen, z. B.:
 - Indexname
 - Name des Indexerstellers

- Spaltennamen sowie die Angabe, ob die einzelnen Spalten in aufsteigender oder absteigender Reihenfolge sortiert sind
- 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

Anmerkung: Enthalten die im Index angegebenen Spaltennamen das Zeichen - oder +, werden keine Indexinformationen beibehalten. In diesem Fall wird die Warnung SQL27984W zurückgegeben.

Nicht beibehaltene Attribute

Die erneut erstellte Tabelle behält verschiedene Attribute der Originaltabelle nicht bei, z. B. folgende Attribute:

- Angabe, ob die Quelle eine normale Tabelle, eine MQT (Materialized Query Table, gespeicherte Abfragetabelle), eine Sicht oder eine Spaltengruppe aus einer oder allen diesen Quellen ist
- Eindeutige Integritätsbedingungen und andere Typen von Integritätsbedingungen oder Triggern (Integritätsbedingungen über Primärschlüssel ausgeschlossen)
- Tabelleninformationen, z. B.:
 - MQT-Definition (sofern zutreffend)
 - MQT-Optionen (sofern zutreffend)
 - Optionen für den Tabellenbereich (diese Informationen können jedoch mit dem Befehl **IMPORT** angegeben werden)
 - Dimensionen für mehrdimensionales Clustering (MDC)
 - Dimensionen für partitionierte Tabellen
 - Tabellenpartitionierungsschlüssel
 - Merkmal NOT LOGGED INITIALLY
 - Prüfung auf Integritätsbedingung
 - Tabellencodepage
 - Merkmale geschützter Tabellen
 - Optionen für die Komprimierung von Tabellen oder Werten
- Spalteninformationen, z. B.:
 - Gegebenenfalls vorhandene Standardwerte (außer Werte für Konstanten)
 - LOB-Optionen (sofern vorhanden)
 - XML-Merkmale
 - Verweisklausel der Anweisung CREATE TABLE (sofern vorhanden)
 - Referenzielle Integritätsbedingungen (sofern vorhanden)
 - Prüfung auf Integritätsbedingungen (sofern vorhanden)
 - Optionen für generierte Spalten (sofern vorhanden)
 - Von Datenbankbereichsfolgen abhängige Spalten
- Indexinformationen, z. B.:
 - INCLUDE-Spalten (sofern vorhanden)
 - Indexname, wenn es sich bei dem Index um einen Primärschlüsselindex handelt

- Absteigende Reihenfolge der Schlüssel, wenn es sich bei dem Index um einen Primärschlüsselindex handelt (die Standardeinstellung ist die aufsteigende Reihenfolge)
- Indexspaltennamen, die den Hexadezimalwert 0x2B oder 0x2D enthalten
- Indexnamen, die nach der Codepagekonvertierung mehr als 128 Byte enthalten
- Wert für PCTFREE2
- Eindeutige Integritätsbedingungen

Anmerkung: Diese Liste ist nicht vollständig und deshalb lediglich eine Orientierungshilfe.

Wenn die Importoperation fehlschlägt und die Nachricht SQL3311N zurückgegeben wird, können Sie die Tabelle auf jeden Fall mit dem Änderungswert `forcecreate` für den Dateityp erneut erstellen. Dieser Änderungswert ermöglicht es Ihnen, die Tabelle mit fehlenden oder beschränkt gültigen Informationen zu erstellen.

Aspekte des Imports typisierter Tabellen

Mit dem Dienstprogramm IMPORT können Daten sowohl aus typisierten Tabellen als auch in typisierten Tabellen unter Beibehaltung der vorhandenen Datenhierarchie versetzt werden. Bei Bedarf können Tabellenhierarchie und Typhierarchie beim Import auch erstellt werden.

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 während des Exports. Das Dienstprogramm IMPORT wiederum steuert Größe und Position der zu versetzenden Hierarchie mit den Parametern **CREATE**, **INTO** *tabellenname*, **UNDER** und **AS ROOT TABLE**. Das Dienstprogramm IMPORT steuert ebenfalls die Einfügungen in die Zieltabelle. Das Dienstprogramm IMPORT kann z. B. 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 einzelnen untergeordneten Tabellen versetzt.

Erneute Erstellung von Tabellen

Der verfügbare Importtyp richtet sich nach dem Dateiformat der Eingabedatei. Bei ASC-, DEL- und WSF-Datendateien muss die Zieltabelle bzw. Zielhierarchie vor dem Datenimport bereits existieren. Daten aus einer PC/IXF-Datei können jedoch importiert werden, auch wenn die Tabelle oder Hierarchie noch nicht vorhanden ist, sofern Sie die Importoperation mit der Option **CREATE** durchführen. Bei Angabe der Option **CREATE** ist jedoch zu beachten, dass die Definitionen untergeordneter Tabellen beim Import nicht geändert werden können.

Traversierfolge

Die in der Eingabedatei enthaltene Traversierfolge ermöglicht das Beibehalten der Datenhierarchien. Beim Arbeiten mit den Dienstprogrammen EXPORT und IMPORT muss dieselbe Traversierfolge verwendet werden.

Beim PC/IXF-Dateiformat muss lediglich der Name der untergeordneten Zieltabelle angegeben und die in der Datei gespeicherte Traversierfolge verwendet werden.

Werden andere Optionen als **CREATE** bei typisierten Tabellen verwendet, können Sie über die Traversierfolgeliste die Traversierfolge angeben. Diese benutzerdefinierte Traversierfolge muss mit der Folge übereinstimmen, die während der Exportopera-

tion verwendet wurde. Das exakte Versetzen von Daten in die Zieldatenbank ist bei dem Dienstprogramm IMPORT unter folgenden Bedingungen gesichert:

- 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

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. Das Dienstprogramm IMPORT erkennt die Nichteinhaltung dieser Bedingung innerhalb der angegebenen Traversierfolge.

Beispiele

Die Beispiele in diesem Abschnitt sind an der folgenden hierarchischen Struktur mit vier gültigen Traversierfolgen ausgerichtet:

- Person, Employee, Manager, Architect, Student
- Person, Student, Employee, Manager, Architect
- Person, Employee, Architect, Manager, Student
- Person, Student, Employee, Architect, Manager

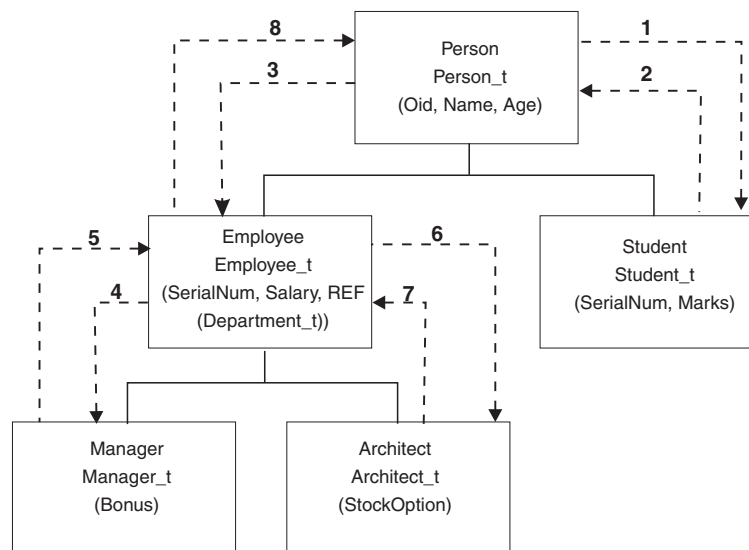


Abbildung 2. Hierarchiebeispiel

Beispiel 1

Zum erneuten Erstellen einer vollständigen Hierarchie (enthalten in der durch eine vorangehende Exportoperation erstellten Datendatei entire_hierarchy.ixf) beim Import müssen Sie folgende Befehle eingeben:

```
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, soweit 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

Zum erneuten Erstellen der gesamten Hierarchie der Quelldatenbank und Importieren dieser Hierarchie in die Zieldatenbank unter Beibehaltung der ausgewählten Daten müssen Sie folgende Befehle eingeben:

```
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 in 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 zugehörigen 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 reguläre (nicht typisierte) Tabellen. Die Datendatei enthält deshalb 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. "Object_id" wird 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.

Aspekte des Imports von LBAC-geschützten Daten

Die Voraussetzung für einen erfolgreichen Import in eine Tabelle mit geschützten Zeilen sind LBAC-Berechtigungs-nachweise (Label-Based Access Control). Darüber hinaus müssen Sie über einen gültigen Sicherheitskennsatz (oder einen Sicherheitskennsatz, der in einen gültigen Kennsatz konvertiert werden kann) für die Sicherheitsrichtlinie verfügen, die der Zieltabelle aktuell zugeordnet ist.

Wenn Sie nicht über gültige LBAC-Berechtigungs-nachweise verfügen, schlägt der Import fehl, und es wird ein Fehler (SQLSTATE 42512) zurückgegeben. Enthalten die Eingabedaten keinen Sicherheitskennsatz oder liegt der Sicherheitskennsatz nicht im zugehörigen internen Binärformat vor, können Sie die Importoperation mithilfe verschiedener Änderungswerte für den Dateityp fortsetzen.

Wenn Sie Daten in eine Tabelle mit geschützten Zeilen importieren, enthält die Zieltabelle eine einzige Spalte mit dem Datentyp DB2SECURITYLABEL. Wenn die Eingabedatenzeile keinen Wert für diese Spalte enthält, wird die Zeile zurückgewiesen, es sei denn, der Importbefehl wurde mit dem Änderungswert `usedefaults` für den Dateityp angegeben. Wurde `usedefaults` im Importbefehl angegeben, wird der Sicherheitskennsatz, über den Sie für den Schreibzugriff verfügen, aus der Sicherheitsrichtlinie, die die Tabelle schützt, verwendet. Wenn Sie keinen Sicherheitskennsatz für den Schreibzugriff haben, wird die Zeile zurückgewiesen und mit der Verarbeitung der nächsten Zeile fortgefahren.

Wenn Sie Daten in eine Tabelle mit geschützten Zeilen importieren und die Eingabedaten einen Wert für die Spalte mit dem Datentyp DB2SECURITYLABEL enthalten, gelten dieselben Regeln wie beim Einfügen von Daten in die betreffende Tabelle. Wenn der Sicherheitskennsatz, der die zu importierende Zeile schützt, also der Kennsatz in der betreffenden Zeile der Datendatei, ein Kennsatz ist, der Ihnen den Schreibzugriff ermöglicht, wird der betreffende Sicherheitskennsatz zum Schutz der Zeile verwendet. Dies bedeutet, dass der Sicherheitskennsatz in die Spalte mit dem Datentyp DB2SECURITYLABEL geschrieben wird. Wenn Sie nicht in eine Zeile schreiben können, die durch diesen Sicherheitskennsatz geschützt wird, hat dies, je nachdem wie die Sicherheitsrichtlinie zum Schutz der Quellentabelle erstellt wurde, unterschiedliche Konsequenzen:

- Wenn die Anweisung `CREATE SECURITY POLICY`, mit der die Richtlinie erstellt wurde, die Option `RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL` enthielt, schlägt das Einfügen fehl und es wird ein Fehler zurückgegeben.
- Enthielt die Anweisung `CREATE SECURITY POLICY` diese Option nicht oder stattdessen die Option `OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL`, wird der Sicherheitskennsatz in der Datendatei für die betreffende Zeile ignoriert und stattdessen zum Schutz der Zeile der Sicherheitskennsatz verwendet, den Sie für den Schreibzugriff besitzen. In diesem Fall wird weder ein Fehler noch eine Warnung ausgegeben. Wenn Sie keinen Sicherheitskennsatz für den Schreibzugriff haben, wird die Zeile zurückgewiesen und mit der Verarbeitung der nächsten Zeile fortgefahren.

Hinweise zu Begrenzern

Wenn Sie Daten in eine Spalte des Typs DB2SECURITYLABEL importieren, wird standardmäßig davon ausgegangen, dass es sich bei dem Wert in der Datendatei um die tatsächlichen Byte handelt, die die interne Darstellung dieses Sicherheitskennsatzes bilden. Einige Rohdaten enthalten jedoch möglicherweise Zeichen für Zeilenvorschub, die vom Befehl `IMPORT` fälschlicherweise als Begrenzer für die

Zeile interpretiert werden. Wenn dieses Problem vorliegt, können Sie mit dem Änderungswert `delprioritychar` für den Dateityp sicherstellen, dass der Zeichenbegrenzer Vorrang vor dem Zeilenbegrenzer hat. Wenn Sie `delprioritychar` verwenden, werden Zeilen- und Spaltenbegrenzer, die innerhalb der zwischen den Zeichenbegrenzern enthaltenen Daten vorliegen, nicht als Begrenzer erkannt. Auch wenn kein Wert ein Zeilenvorschubzeichen enthält, kann der Änderungswert `delprioritychar` für den Dateityp problemlos verwendet werden. Dies führt jedoch zu einer geringfügigen Verlangsamung der Importoperation.

Wenn die zu importierenden Daten im Format ASC (Abstract Syntax Checker) vorliegen, empfiehlt es sich unter Umständen, durch eine zusätzliche Maßnahme sicherzustellen, dass keine abschließenden Leerzeichen in die importierten Sicherheitskennsätze und Namen von Sicherheitskennsätzen eingefügt werden. Da beim ASCII-Format Spaltenpositionen als Begrenzer verwendet werden, kann dies beim Import in Felder variabler Länge vorkommen. Verwenden Sie zum Abschneiden von abschließenden Leerstellen den Änderungswert `striptblanks` für den Dateityp.

Vom Standard abweichende Sicherheitskennsatzwerte

Sie können auch Datendateien importieren, bei denen die Werte für die Sicherheitskennsätze Zeichenfolgen darstellen, die die Werte der Komponenten im jeweiligen Sicherheitskennsatz enthalten, z. B. `S:(ALPHA,BETA)`. Verwenden Sie hierzu den Änderungswert `seclabelchar` für den Dateityp. Bei Verwendung von `seclabelchar` werden Werte für Spalten mit dem Datentyp `DB2SECURITYLABEL` als Zeichenfolgekonstanten gewertet, die den Sicherheitskennsatz im Zeichenfolgeformat für Sicherheitskennsätze enthalten. Liegt eine Zeichenfolge nicht im richtigen Format vor, wird die jeweilige Zeile nicht eingefügt und eine Warnung (SQLSTATE 01H53) zurückgegeben. Stellt eine Zeichenfolge keinen gültigen Sicherheitskennsatz dar, der Bestandteil der zum Schutz der Tabelle verwendeten Sicherheitsrichtlinie ist, wird die betreffende Zeile nicht eingefügt und eine Warnung (SQLSTATE 01H53) zurückgegeben.

Sie können auch Datendateien importieren, bei denen die Werte für die Sicherheitskennsatzspalte Namen von Sicherheitskennsätzen sind. Verwenden Sie zum Importieren derartiger Dateien den Änderungswert `seclabelname` für den Dateityp. Bei Verwendung von `seclabelname` werden alle Werte für Spalten mit dem Typ `DB2SECURITYLABEL` als Zeichenfolgekonstanten gewertet, die den Namen vorhandener Sicherheitskennsätze enthalten. Existiert kein Sicherheitskennsatz mit dem angegebenen Namen für die zum Schutz der Tabelle verwendete Sicherheitsrichtlinie, wird die betreffende Zeile nicht eingefügt und eine Warnung (SQLSTATE 01H53) zurückgegeben.

Beispiele

Bei allen Beispielen liegt die Eingabedatendatei `myfile.del` im Format DEL vor. Die Daten werden jeweils in eine Tabelle mit der Bezeichnung `REPS` importiert, die mit der folgenden Anweisung erstellt wurde:

```
create table reps (row_label db2securitylabel,  
id integer,  
name char(30))  
security policy data_access_policy
```

Bei dem folgenden Beispiel wird davon ausgegangen, dass die Eingabedatei Sicherheitskennsätze im Standardformat enthält:

```
db2 import from myfile.del of del modified by delprioritychar insert into reps
```

Bei dem folgenden Beispiel wird davon ausgegangen, dass die Eingabedatei Sicherheitskennsätze im Zeichenfolgeformat für Sicherheitskennsätze enthält:

```
db2 import from myfile.del of del modified by seclabelchar insert into reps
```

Bei dem folgenden Beispiel wird davon ausgegangen, dass die Eingabedatei Namen von Sicherheitskennsätzen für die Sicherheitskennsatzspalte enthält:

```
db2 import from myfile.del of del modified by seclabelname insert into reps
```

Gepufferte INSERT-Operationen

In einer Umgebung mit partitionierten Datenbanken kann das Dienstprogramm IMPORT für die Verwendung gepufferter INSERT-Operationen aktiviert werden. Dadurch wird der Umfang der Nachrichtenübertragung beim Datenimport reduziert und die Leistung somit optimiert.

Die Option für gepufferte INSERT-Operationen sollten Sie nur aktivieren, wenn Sie auf eine Fehlerprotokollierung verzichten können, da Details zu fehlgeschlagenen gepufferten INSERT-Operationen nicht zurückgegeben werden.

Werden gepufferte INSERT-Operationen verwendet, setzt IMPORT den Wert für **WARNINGCOUNT** standardmäßig auf 1, sodass die Operation fehlschlägt, wenn eine beliebige Zeile zurückgewiesen wird. Wird ein Datensatz zurückgewiesen, setzt das Dienstprogramm die aktuelle Transaktion mittels eines Rollbacks zurück. Anhand der Anzahl der festgeschriebenen Datensätze kann ermittelt werden, welche Datensätze erfolgreich in die Datenbank eingefügt wurden. Die Anzahl der festgeschriebenen Datensätze kann nur ungleich null sein, wenn die Option **COMMITCOUNT** angegeben wurde.

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 db2uimp.bnd muss mit der Option **INSERT BUF** erneut an die Datenbank gebunden werden. Zum Beispiel:

```
db2 connect to datenbankname
db2 bind db2uimp.bnd insert buf
```

Die Funktion der gepufferten INSERT-Operationen kann nicht in Verbindung mit Importoperationen verwendet werden, bei denen der Modus **INSERT_UPDATE** angegeben ist. Diese Einschränkung wird von der Bindedatei db2uImpInsUpdate.bnd erzwungen. Diese Datei darf auf keinen Fall mit der Option **INSERT BUF** gebunden werden. Andernfalls schlagen Importoperationen im Modus **INSERT_UPDATE** fehl. Importoperationen im Modus **INSERT**, **REPLACE** oder **REPLACE_CREATE** werden durch das Binden der neuen Datei nicht beeinträchtigt.

Aspekte des Identitätsspaltenimports

Mit dem Dienstprogramm IMPORT können Sie Daten in eine Tabelle importieren, die eine Identitätsspalte enthält. Dabei spielt es keine Rolle, ob die Eingabedaten Identitätsspaltenwerte enthalten.

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.

Sie können den Import in Tabellen mit Identitätsspalte mit dem Änderungswert identitymissing für den Dateityp und mit dem Änderungswert identityignore für den Dateityp vereinfachen.

Import von Daten ohne Identitätsspalte

Der Änderungswert identitymissing vereinfacht das Importieren einer Tabelle mit einer Identitätsspalte, wenn die Eingabedatei 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))
```

Angenommen, ein Benutzer möchte 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 für den Dateityp identitymissing:

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

Import von Daten mit Identitätsspalte

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 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).

Anmerkung: Die Optionen `CREATE` und `REPLACE_CREATE` des Befehls **IMPORT** werden nicht weiter unterstützt und in zukünftigen Releases möglicherweise entfernt.

Aspekte des Imports von generierten Spalten

Mit dem Dienstprogramm **IMPORT** können Sie Daten in eine Tabelle mit generierten Spalten (Tabellen mit Identitätsspalte ausgenommen) importieren. Dabei spielt es keine Rolle, ob die Eingabedaten Werte für generierte Spalten enthalten.

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 (SQL0407W). 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.

Sie können den Import von Daten in Tabellen mit generierten Spalten mit dem Änderungswert `identitymissing` für den Dateityp und mit dem Änderungswert `identityignore` für den Dateityp vereinfachen.

Import von Daten ohne generierte Spalten

Der Änderungswert `generatedmissing` vereinfacht das Importieren 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))
```

Angenommen, ein Benutzer möchte 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 über den 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 für den Datentyp generatedmissing:

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

Import von Daten mit generierten Spalten

Der Änderungswert generatedignore stellt in gewisser Hinsicht das Gegenteil des Änderungswerts generatedmissing dar: Er weist das Dienstprogramm IMPORT an, dass die in der Eingabedatendatei 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 beschriebenen 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
```

Wenn Sie die Klausel INSERT_UPDATE verwenden, die generierte Spalte auch ein Primärschlüssel ist und der Änderungswert generatedignore angegeben ist, wird der Änderungswert generatedignore vom Befehl **IMPORT** berücksichtigt. Der Befehl

IMPORT ersetzt nicht den vom Benutzer angegebenen Wert für diese Spalte in der Klausel **WHERE** der Anweisung **UPDATE**.

Aspekte des LOB-Imports

Da die Größe von Spaltenwerten beim Dienstprogramm **IMPORT** auf 32 KB begrenzt ist, müssen beim Import von LOBs (Large Objects) besondere Aspekte beachtet werden.

Das Dienstprogramm **IMPORT** behandelt Daten in der Eingabedatei standardmäßig wie in Spalten zu ladende Daten. Sind in der Haupteingabedatei jedoch LOB-Daten gespeichert, ist der Grenzwert für die Datengröße von 32 KB relevant. Es empfiehlt sich deshalb, LOB-Daten separat von der Hauptdatei zu speichern und beim Import von LOBs den Änderungswert `lobsinfile` für den Dateityp anzugeben, um einen Verlust von Daten zu vermeiden.

Die Klausel **LOBS FROM** impliziert die Aktivierung des Änderungswerts `lobsinfile`. Die Klausel **LOBS FROM** überträgt die Liste der Pfade an das Dienstprogramm **IMPORT**, die beim Importieren der Daten nach LOB-Dateien durchsucht werden sollen. Wird die Option **LOBS FROM** nicht angegeben, wird davon ausgegangen, dass sich die zu importierenden LOB-Dateien in demselben Pfad befinden wie die Eingabedatei mit den relationalen Daten.

Angabe zur Speicherung der LOB-Daten

Mit einer LOB-Positionskenung (LOB Location Specifier - LLS) können beim Import von LOB-Informationen mehrere große Objekte in einer gemeinsamen Datei gespeichert werden. Ist `lobsinfile` angegeben, generiert und speichert das Dienstprogramm **EXPORT** die Kennung in der Ausgabedatei für die Exportoperation, sodass der Speicherort der LOB-Daten anhand der Kennung ermittelt werden kann. Werden Daten importiert, bei denen die Option "MODIFIED BY `lobsinfile`" angegeben ist, erwartet die Datenbank jeweils eine LOB-Positionskenung für die betreffenden LOB-Spalten. Falls für eine LOB-Spalte eine andere Angabe als die LOB-Positionskenung gefunden wird, führt dies bei der Datenbank zu einer Behandlung als LOB-Datei, und die Datenbank lädt die gesamte Datei als großes Objekt.

Bei einem Import im Modus **CREATE** können Sie mit der Klausel **LONG IN** angeben, dass die LOB-Daten in einem separaten Tabellenbereich erstellt und gespeichert werden sollen.

Das folgende Beispiel veranschaulicht den Import einer DEL-Datei, bei der die zugehörigen LOBs in separaten Dateien gespeichert sind:

```
IMPORT FROM inputfile.del OF DEL
  LOBS FROM /tmp/data
  MODIFIED BY lobsinfile
  INSERT INTO newtable
```

Aspekte des Imports benutzerdefinierter einzigartiger Typen

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.

Weitere Aspekte des Imports

Client/Server-Umgebungen und Import

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 ist das Aufrufen einer gespeicherten Prozedur nicht möglich:

- 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 mithilfe 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 Registrierdatenbankvariable **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 Instanzeigner hat Schreibzugriff auf das Verzeichnis.

Vom Dienstprogramm IMPORT unterstützte Tabellensperremodi

Das Dienstprogramm IMPORT unterstützt bei Tabellensperren den Offlinemodus (ALLOW NO ACCESS) und den Onlinemodus ALLOW WRITE ACCESS.

Der Modus ALLOW NO ACCESS verhindert den Zugriff auf Tabellendaten durch gleichzeitig ablaufende Anwendungen. Der Modus ALLOW WRITE ACCESS ermöglicht gleichzeitig ablaufenden Anwendungen sowohl Lese- als auch Schreibzugriffe auf die Zieltabelle der Importoperation. Ist kein Modus explizit angegeben, wird der Standardmodus ALLOW NO ACCESS beim Import verwendet. Darüber hinaus ist das Dienstprogramm IMPORT standardmäßig an die Datenbank mit der Isolationsstufe RS (Read Stability - Lesestabilität) gebunden.

Offlineimport (ALLOW NO ACCESS)

Im Modus ALLOW NO ACCESS fordert die Importoperation vor dem Einfügen von Zeilen eine exklusive Sperre (X) für die Zieltabelle an. Das Halten einer Sperre für eine Tabelle hat folgende Auswirkungen:

- Erstens wartet das Dienstprogramm IMPORT, wenn andere Anwendungen eine Tabellensperre halten oder für die Zieltabelle der Importoperation Zeilensperren

vorliegen, bis diese Anwendungen ihre Änderungen mit COMMIT festschreiben oder mit ROLLBACK rückgängig machen.

- Zweitens warten während der Ausführung der Importoperation alle anderen Anwendungen, die eine Sperre anfordern, bis die Importoperation vollständig ausgeführt wurde.

Anmerkung: Sie können ein Zeitlimit für Sperren angeben, damit die Anwendungen (das Dienstprogramm IMPORT eingeschlossen) nicht unbegrenzt auf Sperren warten.

Durch das Anfordern einer exklusiven Sperre zu Beginn der Operation verhindert das Dienstprogramm IMPORT, dass Deadlocks durch andere Anwendungen, die mit derselben Zieltabelle arbeiten und Zeilensperren halten, entstehen.

Onlineimport (ALLOW WRITE ACCESS)

Im Modus ALLOW WRITE ACCESS fordert das Dienstprogramm IMPORT eine nicht exklusive 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, wenn alle betreffenden Anwendungen ihre Änderungen mit COMMIT festgeschrieben oder mit ROLLBACK zurückgesetzt haben.
- Während der Ausführung der Importoperation warten alle anderen Anwendungen, die eine inkompatible Tabellensperre anfordern, bis die Importoperation die aktuelle Transaktion mit COMMIT festschreibt oder mit ROLLBACK zurücksetzt. Hierbei ist zu beachten, dass die Tabellensperre von IMPORT nicht über Transaktionsgrenzen hinaus bestehen bleibt. Aus diesem Grund muss beim Onlineimport nach jedem Commit 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 so lange, bis alle betreffenden Anwendungen ihre Änderungen mit COMMIT festgeschrieben oder mit ROLLBACK zurückgesetzt haben.
- Während der Ausführung der Importoperation warten alle anderen Anwendungen, die eine inkompatible Zeilensperre anfordern, bis die Importoperation die aktuelle Transaktion mit COMMIT festschreibt oder mit ROLLBACK zurücksetzt.

Um die Onlinemerkmale beizubehalten und das Risiko eines Deadlocks zu reduzieren, führt die Importoperation im Modus ALLOW WRITE ACCESS in regelmäßigen Abständen ein Commit für die aktuelle Transaktion durch und gibt alle Zeilensperren frei, bevor eine Eskalation zu einer exklusiven Tabellensperre (X) erfolgt. Wenn Sie die Häufigkeit der Commitoperationen nicht explizit definiert haben, werden die Commitoperationen beim Import so oft durchgeführt, wie es bei Angabe von COMMITCOUNT AUTOMATIC der Fall wäre. Es werden keine Commitoperationen durchgeführt, wenn die Commitzählung COMMITCOUNT auf 0 gesetzt ist.

Der Modus ALLOW WRITE ACCESS ist mit folgenden Importvarianten nicht kompatibel:

- Import im Modus REPLACE, CREATE oder REPLACE_CREATE
- Import mit gepufferten INSERT-Operationen
- Import in eine Zielsicht
- Import in eine Hierarchietabelle

- Import in eine Tabelle, bei der die Sperrgranularität auf Tabellenebene (durch Verwendung des Parameters LOCKSIZE in der Anweisung ALTER TABLE) definiert ist

Referenz - Import

IMPORT

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

Quick Link zu „Änderungswerte für Dateityp für das Dienstprogramm IMPORT “ auf Seite 84.

Berechtigung

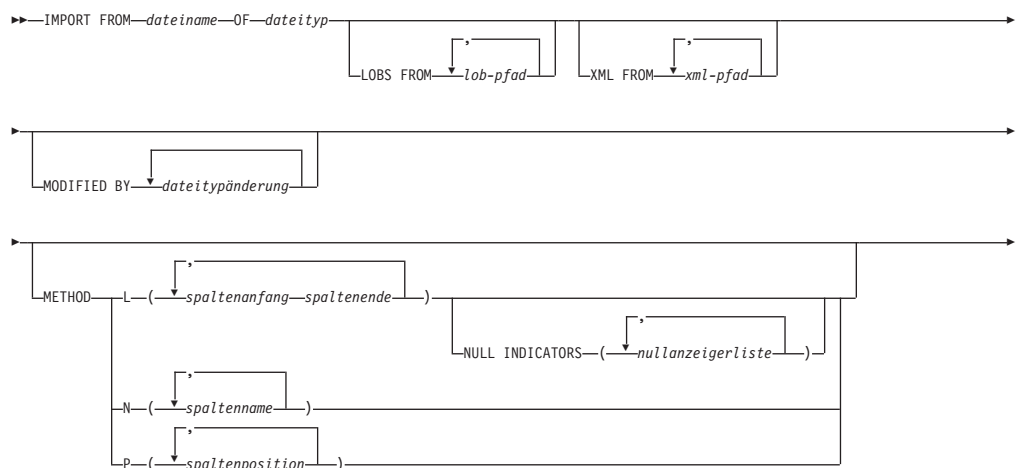
- Für die Verwendung von **IMPORT** mit der Option **INSERT** ist eine der folgenden Berechtigungen erforderlich:
 - Berechtigung DATAACCESS
 - Zugriffsrecht CONTROL für die einzelnen betroffenen Tabellen, Sichten oder Kurznamen
 - Zugriffsrechte INSERT und SELECT für jede betroffene Tabelle oder Sicht
- Für die Verwendung von **IMPORT** für eine bestehende Tabelle mit der Option **INSERT_UPDATE** ist eine der folgenden Berechtigungen erforderlich:
 - Berechtigung DATAACCESS
 - Zugriffsrecht CONTROL für die einzelnen betroffenen Tabellen, Sichten oder Kurznamen
 - Zugriffsrecht INSERT, SELECT, UPDATE und DELETE für die einzelnen betroffenen Tabellen oder Sichten
- Für die Verwendung von **IMPORT** für eine bestehende Tabelle mit der Option **REPLACE** oder **REPLACE_CREATE** ist eine der folgenden Berechtigungen erforderlich:
 - Berechtigung DATAACCESS
 - Zugriffsrecht CONTROL für die Tabelle oder 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:
 - Berechtigung DBADM
 - Berechtigung CREATETAB für die Datenbank und das Zugriffsrecht USE für den Tabellenbereich sowie eine der folgenden Berechtigungen (bzw. Zugriffsrechte):
 - Berechtigung IMPLICIT_SCHEMA für die Datenbank, wenn der implizite oder explizite Schemaname der Tabelle nicht existiert
 - Zugriffsrecht CREATEIN für das Schema, wenn sich der Schemaname der Tabelle auf ein vorhandenes Schema bezieht
- Für die Verwendung von **IMPORT** in eine nicht vorhandene Hierarchie mit der Option **CREATE** oder **REPLACE_CREATE** ist eine der folgenden Berechtigungen erforderlich:
 - Berechtigung DBADM
 - Berechtigung CREATETAB für die Datenbank und Zugriffsrecht USE für den Tabellenbereich sowie eine der folgenden Berechtigungen:

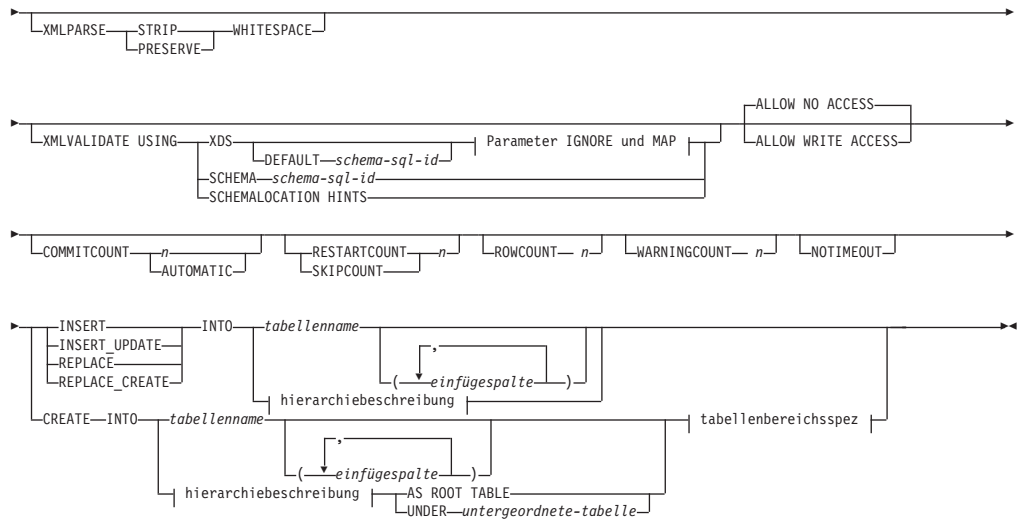
- Berechtigung `IMPLICIT_SCHEMA` für die Datenbank, falls der Name der Tabelle nicht vorhanden ist
 - Zugriffsrecht `CREATEIN` für das Schema, falls das Schema der Tabelle vorhanden ist
 - Zugriffsrecht `CONTROL` für jede untergeordnete Tabelle in der Hierarchie, wenn die Option `REPLACE_CREATE` für die gesamte Hierarchie verwendet wird
- Für die Verwendung von `IMPORT` für eine vorhandene Hierarchie mit der Option `REPLACE` ist eine der folgenden Berechtigungen erforderlich:
 - Berechtigung `DATAACCESS`
 - Zugriffsrecht `CONTROL` für alle untergeordneten Tabellen in der Hierarchie
 - Um Daten in eine Tabelle importieren zu können, die geschützte Spalten enthält, muss die Berechtigungs-ID der Sitzung über LBAC-Berechtigungs-nachweise verfügen, die Schreibzugriffe auf alle geschützten Spalten in der Tabelle zulassen. Andernfalls schlägt der Import fehl, und es wird ein Fehler (SQLSTATE 42512) zurückgegeben.
 - Um Daten in eine Tabelle laden zu können, die geschützte Zeilen enthält, muss für die Berechtigungs-ID der Sitzung ein LBAC-Berechtigungs-nachweis vorliegen, der den folgenden Kriterien entspricht:
 - Er ist Teil der Sicherheitsrichtlinie, die die Tabelle schützt
 - Er wurde der Berechtigungs-ID der Sitzung für den Schreibzugriff erteilt.

Der Kennsatz der einzufügenden Zeile, der LBAC-Berechtigungs-nachweis des Benutzers und die LBAC-Regeln bestimmen den Kennsatz der Zeile.
 - Wenn die Option `REPLACE` oder `REPLACE_CREATE` angegeben wird, muss die Berechtigungs-ID der Sitzung über die Berechtigung zum Löschen der Tabelle verfügen.
 - Um Daten in einen Kurznamen importieren zu können, muss die Berechtigungs-ID der Sitzung über das Recht zum Zugriff auf eine angegebene Datenquelle und zu deren Verwendung im Durchgriffsmodus verfügen.

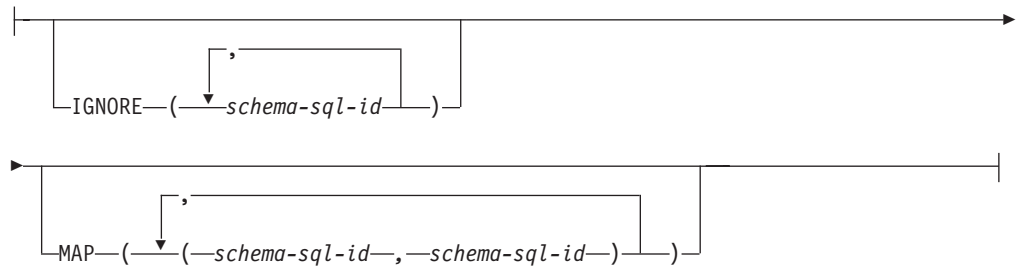
Erforderliche Verbindung

Befehlssyntax

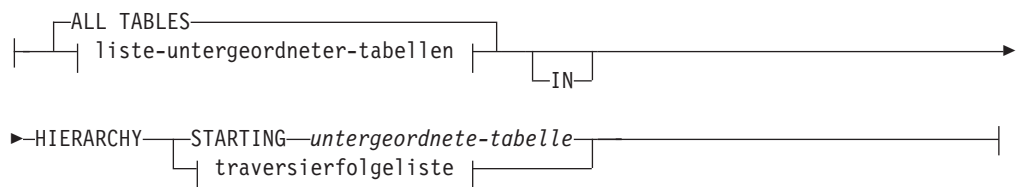




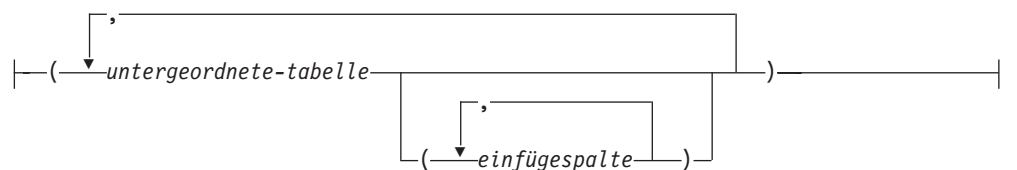
Parameter IGNORE und MAP:



hierarchiebeschreibung:



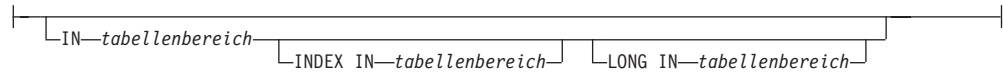
liste-untergeordneter-tabellen:



traversierfolgeliste:



tabellenbereichsspez:



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 Commits durch, um eine Sperreneskulation auf eine Tabellensperre zu vermeiden und um zu gewährleisten, dass die aktive Protokolldatei über ausreichend Speicher verfügt. Diese Commits werden auch dann durchgeführt, wenn die Option **COMMITCOUNT** nicht verwendet wurde. Bei jedem Commit verliert IMPORT die Tabellensperre IX und versucht, diese nach jedem Commit erneut anzufordern. Dieser Parameter ist erforderlich, wenn Daten in einen Kurznamen importiert werden. Für den Parameter **COMMITCOUNT** muss eine gültige Zahl angegeben werden (**AUTOMATIC** ist keine gültige Option).

AS ROOT TABLE

Erstellt eine oder mehrere untergeordnete Tabellen als eigenständige Tabellenhierarchie.

COMMITCOUNT *n* | **AUTOMATIC**

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

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

Wenn die Option **ALLOW WRITE ACCESS** angegeben und die Option **COMMITCOUNT** nicht angegeben wird, führt das Dienstprogramm IMPORT Commits genau so durch wie bei Angabe von **COMMITCOUNT AUTOMATIC**.

Die DB2-Registrierdatenbankvariable **DB2_FORCE_APP_ON_MAX_LOG** hat Auswirkungen auf die Funktionalität der Importoperation, zu gewährleisten, dass für die aktive Protokolldatei ausreichend Speicher verfügbar ist:

- Wenn die Option **DB2_FORCE_APP_ON_MAX_LOG** auf FALSE gesetzt ist und die Befehlsoption **COMMITCOUNT** AUTOMATIC angegeben ist, kann das Dienstprogramm IMPORT automatisch gewährleisten, dass für die aktive Protokolldatei ausreichend Speicher verfügbar ist.
- Wenn die Option **DB2_FORCE_APP_ON_MAX_LOG** auf FALSE gesetzt ist und die Befehlsoption **COMMITCOUNT** *n* angegeben ist, versucht das Dienstprogramm IMPORT, die Bedingung 'Protokoll voll' aufzulösen, wenn es beim Einfügen oder Aktualisieren eines Datensatzes den Fehler SQL0964C (Transaktionsprotokoll voll) feststellt. Das Dienstprogramm führt ein bedingungsloses Commit durch und wiederholt anschließend das Einfügen oder Aktualisieren des Datensatzes. Wenn das Problem auf diese Weise nicht behoben werden kann (z. B. wenn das Protokoll durch eine andere Datenbankaktivität belegt wird), schlägt der Befehl **IMPORT** wie erwartet fehl. Die Anzahl der festgeschriebenen Zeilen ist in diesem Fall jedoch möglicherweise kein Vielfaches des Werts **COMMITCOUNT** *n*. Um zu verhindern, dass beim Wiederholen der Importoperation die bereits festgeschriebenen Zeilen verarbeitet werden, können Sie die Befehlsparameter **RESTARTCOUNT** oder **SKIPCOUNT** verwenden.
- Wenn die Option **DB2_FORCE_APP_ON_MAX_LOG** auf TRUE (Standardwert) gesetzt ist, schlägt die Importoperation fehl, wenn das Dienstprogramm beim Einfügen oder Aktualisieren eines Datensatzes den Fehler SQL0964C feststellt. Die Operation kann dabei unabhängig davon fehlschlagen, ob Sie den Parameter **COMMITCOUNT** AUTOMATIC oder **COMMITCOUNT** *n* angeben.

Die Anwendung wird zwangsweise von der Datenbank abgemeldet und es wird ein Rollback der aktuellen UOW (Unit of Work) durchgeführt. Um zu verhindern, dass beim Wiederholen der Importoperation die bereits festgeschriebenen Zeilen verarbeitet werden, können Sie die Befehlsparameter **RESTARTCOUNT** oder **SKIPCOUNT** verwenden.

CREATE

Anmerkung: Der Parameter **CREATE** ist veraltet und wird in einem künftigen Release möglicherweise entfernt. Weitere Informationen finden Sie unter „Optionen **CREATE** und **REPLACE_CREATE** des Dienstprogramms IMPORT sind veraltet“.

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

Dieser Parameter ist beim Importieren in einen Kurznamen nicht zulässig.

Anmerkung: Wenn Daten aus einer MVS-Hostdatenbank exportiert wurden und Felder vom Typ LONGVAR enthalten, deren auf Basis der Seitengröße berechnete Länge größer als 254 ist, schlägt **CREATE** möglicherweise fehl, weil die Zeilen zu lang sind. Eine Liste der Einschränkungen finden Sie unter „Neuerstellung von importierten Tabellen“. In diesem Fall sollte die Tabelle manuell erstellt und **IMPORT** mit **INSERT** aufgerufen oder alternativ der Befehl **LOAD** verwendet werden.

DEFAULT *schema-sql-id*

Diese Option kann nur verwendet werden, wenn der Parameter **USING XDS** angegeben wird. Das über die Klausel **DEFAULT** angegebene Schema gibt an, welches Schema für die Prüfung verwendet werden soll, wenn die XML-Datenkennung (XDS) eines importierten XML-Dokuments kein SCH-Attribut enthält, das ein XML-Schema identifiziert.

Die Klausel **DEFAULT** hat Vorrang vor den Klauseln **IGNORE** und **MAP**. Wenn eine XDS die Klausel **DEFAULT** erfüllt, werden die Angaben **IGNORE** und **MAP** ignoriert.

FROM *dateiname***HIERARCHY**

Gibt an, dass hierarchische Daten importiert werden sollen.

IGNORE *schema-sql-id*

Diese Option kann nur verwendet werden, wenn der Parameter **USING XDS** angegeben wird. Die Klausel **IGNORE** gibt eine Liste mit mindestens einem Schema an, das ignoriert werden soll, wenn es durch ein SCH-Attribut identifiziert wird. Wenn in der XML-Datenkennung ein SCH-Attribut für ein importiertes XML-Dokument enthalten ist und das von dem SCH-Attribut identifizierte Schema in der Liste der zu ignorierenden Schemata enthalten ist, wird für das importierte XML-Dokument keine Schemaprüfung durchgeführt.

Wenn in der Klausel **IGNORE** ein Schema angegeben ist, darf dieses nicht zusätzlich im linken Teil eines Schemapaars in der Klausel **MAP** enthalten sein.

Die Klausel **IGNORE** gilt nur für die XDS. Ein von der Klausel **MAP** zugeordnetes Schema wird anschließend nicht ignoriert, wenn es von der Klausel **IGNORE** angegeben wird.

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 in der Klausel **IN** angegebene primäre Tabellenbereich 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 erstellte temporäre Tabelle, keine deklarierte temporäre Tabelle und keine Übersichtstabelle sein.

Für **INSERT**, **INSERT_UPDATE**, oder **REPLACE** kann ein Aliasname verwendet werden. Dies gilt nicht bei früheren Serverversionen, für die ein vollständig qualifizierter oder ein nicht qualifizierter Tabellenname verwendet werden sollte. Ein qualifizierter Tabellenname hat folgendes Format: *schema.tabellenname*. Dabei ist *schema* der Benutzername, mit dem die Tabelle erstellt wurde.

LOBS FROM *lob-pfad*

Die Namen der LOB-Datendateien werden in der Hauptdatendatei (ASC, DEL oder IXF) in der Spalte gespeichert, die in die LOB-Spalte geladen wird. Es können maximal 999 Pfade angegeben werden. Dadurch wird implizit das Verhalten LOBSINFILE aktiviert.

Dieser Parameter ist beim Importieren in einen Kurznamen nicht zulässig.

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 in der Klausel **IN** angegebene primäre Tabellenbereich ein DMS-Tabellenbereich ist. Der Tabellenbereich muss vorhanden und ein DMS-Tabellenbereich des Typs LARGE sein.

MAP *schema-sql-id*

Diese Option kann nur verwendet werden, wenn der Parameter **USING XDS** angegeben wird. Geben Sie über die Klausel **MAP** alternative Schemata an, die anstelle der Schemata verwendet werden sollen, die im Attribut SCH einer XML-Datenkennung (XDS) für jedes importierte XML-Dokument angegeben sind. Die Klausel **MAP** gibt eine Liste mit mindestens einem Schemapaar an, in der jedes Paar eine Zuordnung eines Schemas zu einem anderen Schema darstellt. Das erste Schema in dem Zuordnungspaar stellt ein Schema dar, auf das durch ein SCH-Attribut in einer XDS verwiesen wird. Das zweite Schema in dem Zuordnungspaar gibt an, welches Schema für die Durchführung der Schemaprüfung verwendet werden soll.

Wenn im linken Teil eines Schemapaars in der Klausel **MAP** ein Schema angegeben ist, darf dieses nicht zusätzlich in der Klausel **IGNORE** angegeben werden.

Sobald eine Schemapaarzuordnung angewendet wurde, ist das Ergebnis endgültig. Die Zuordnungsoperation ist nicht transitiv, das gewählte Schema wird deshalb nicht anschließend auf eine andere Schemapaarzuordnung angewendet.

Ein Schema kann nicht mehrmals zugeordnet werden, d. h. ein Schema kann nur bei einem einzigen Paar als erstes Schema angegeben sein.

METHOD

L Gibt die Anfangs- und Endnummer der Spalten an, aus denen Da-

ten 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 Spalten in der zu importierenden 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 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 *dateitypänderung*

Gibt Optionen für Änderungswerte für Dateityp an. Siehe „Änderungswerte für Dateityp für das Dienstprogramm IMPORT “ auf Seite 84.

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 wird. 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 einen Eintrag für jedes Datenfeld enthalten, das im Parameter **METHOD L** definiert wird. Eine Spaltennummer 0 gibt an, dass das entsprechende Datenfeld immer Daten enthält.

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

Das Zeichen für den Nullanzeiger kann über die Option **MODIFIED BY** mit dem Dateitypänderungswert nullindchar geändert werden.

OF *dateityp*

Gibt das Format der Daten in der Eingabedatei an:

- ASC (ASCII-Format mit universellen Zeilenbegrenzern)

- DEL (ASCII-Format ohne universelle Zeilenbegrenzer) wird von verschiedenen Datenbankmanager- und Dateimanagerprogrammen verwendet
- WSF (Work Sheet Format) wird von verschiedenen Programmen wie den folgenden verwendet:
 - Lotus 1-2-3
 - Lotus Symphony
- IXF (Integration Exchange Format, PC-Version) ist ein Binärformat, das ausschließlich von DB2 verwendet wird.

Wichtig: Die Unterstützung für das WSF-Dateiformat ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.

Beim Importieren von Daten in einen Kurznamen wird der Dateityp WSF nicht unterstützt.

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. 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.

Dieser Parameter ist beim Importieren in einen Kurznamen nicht zulässig.

Diese Option berücksichtigt weder die Klausel NOT LOGGED INITIALLY (NLI) der Anweisung CREATE TABLE noch die Klausel ACTIVE NOT LOGGED INITIALLY der Anweisung ALTER TABLE.

Wenn innerhalb einer Transaktion mit einer Anweisung CREATE TABLE oder ALTER TABLE, bei der die Klausel NLI aufgerufen wird, ein Import mit der Option **REPLACE** ausgeführt wird, wird die Klausel NLI beim Import nicht berücksichtigt. Alle Einfügungen werden protokolliert.

Problemumgehung 1

Löschen Sie den Inhalt der Tabelle mit der Anweisung DELETE. Rufen Sie die Importoperation dann mit der Anweisung INSERT auf.

Problemumgehung 2

Löschen Sie die Tabelle, und erstellen Sie sie erneut. Rufen Sie die Importoperation dann mit der Anweisung INSERT auf.

Diese Einschränkung gilt für DB2 Universal Database Version 7 und DB2 UDB Version 8.

REPLACE_CREATE

Anmerkung: Der Parameter **REPLACE_CREATE** ist veraltet und wird in einem künftigen Release möglicherweise entfernt. Weitere Informationen finden Sie unter „Optionen CREATE und REPLACE_CREATE des Dienstprogramms IMPORT sind veraltet“.

Löscht, wenn die Tabelle vorhanden ist, alle vorhandenen Daten in 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 *Neuerstellung von importierten Tabellen*.

Diese Option kann nur mit IXF-Dateien verwendet werden. 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.

Dieser Parameter ist beim Importieren eines Kurznamens nicht zulässig.

RESTARTCOUNT *n*

Gibt an, dass eine Importoperation bei Datensatz $n + 1$ gestartet werden soll. Die ersten n Datensätze werden übersprungen. Diese Option entspricht in ihrer Funktionalität der Option **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. Erlaubt einem Benutzer das Importieren von genau n Zeilen aus einer Datei, ab dem durch die Option **SKIPCOUNT** oder **RESTARTCOUNT** festgelegten Datensatz. Wenn die Optionen **SKIPCOUNT** und **RESTARTCOUNT** nicht angegeben sind, werden die ersten n Zeilen importiert. Wenn **SKIPCOUNT** m oder **RESTARTCOUNT** m angegeben ist, werden die Zeilen $m+1$ bis $m+n$ importiert. Bei Verwendung von Compound-INSERT-Operationen wird eine benutzerdefinierte Angabe **ROWCOUNT** n auf das erste ganzzahlige Vielfache des Compound-Zählerwerts aufgerundet.

SKIPCOUNT *n*

Gibt an, dass eine Importoperation bei Datensatz $n + 1$ gestartet werden soll. Die ersten n Datensätze werden übersprungen. Diese Option entspricht in ihrer Funktionalität der Option **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** werden in einer Liste die Namen der untergeordneten Tabellen angegeben, in die Daten importiert werden sollen.

traversierfolgeliste

Für typisierte Tabellen mit der Option **INSERT**, **INSERT_UPDATE** oder **REPLACE** wird durch eine Liste mit den Namen untergeordneter Tabellen die Traversierfolge für die zu importierenden untergeordneten Tabellen in der Hierarchie angegeben.

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 dar-

auf, 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.

XML FROM *xml-pfad*

Gibt mindestens einen Pfad mit XML-Dateien an.

XMLPARSE

Gibt an, wie die XML-Dokumente syntaktisch analysiert werden. Wenn diese Option nicht angegeben ist, wird das Parsingverhalten für XML-Dokumente durch den Wert des Sonderregisters CURRENT XMLPARSE OPTION festgelegt.

STRIP WHITESPACE

Gibt an, dass Leerzeichen bei der Syntexanalyse des XML-Dokuments entfernt werden sollen.

PRESERVE WHITESPACE

Gibt an, dass Leerzeichen bei der Syntexanalyse des XML-Dokuments nicht entfernt werden sollen.

XMLVALIDATE

Gibt an, dass XML-Dokumente anhand eines Schemas geprüft werden sollen (soweit möglich).

USING XDS

XML-Dokumente werden anhand des XML-Schemas geprüft, das über die XDS in der Hauptdatendatei angegeben ist. Wenn die Option **XMLVALIDATE** mit der Klausel **USING XDS** aufgerufen wird, wird das für die Prüfung verwendete Schema standardmäßig durch das SCH-Attribut der XDS festgelegt. Enthält die XDS kein SCH-Attribut, wird eine Schemaprüfung nur durchgeführt, wenn die Klausel **DEFAULT** ein Standardschema angibt.

Mit den Klauseln **DEFAULT**, **IGNORE** und **MAP** kann das Verhalten der Schemafestlegung geändert werden. Diese optionalen Klauseln werden nicht auf einander, sondern direkt auf die XDS-Spezifikationen angewendet. Wird ein Schema beispielsweise ausgewählt, weil es von der Klausel **DEFAULT** angegeben wurde, so wird dieses Schema nicht ignoriert, wenn es auch von der Klausel **IGNORE** angegeben wird. Wird ein Schema ausgewählt, weil es als erster Teil eines Zuordnungspaars in der Klausel **MAP** angegeben wurde, so wird dieses Schema nicht erneut zugeordnet, wenn es auch im zweiten Teil des Zuordnungspaars einer anderen Klausel **MAP** angegeben ist.

USING SCHEMA *schema-sql-id*

XML-Dokumente werden anhand des XML-Schemas mit der angegebenen SQL-ID geprüft. In diesem Fall wird das Attribut SCH der XDS für alle XML-Spalten ignoriert.

USING SCHEMALOCATION HINTS

XML-Dokumente werden anhand des XML-Schemas geprüft, das über die Angaben zur XML-Schemaposition in den XML-Quelldokumenten angegeben ist. Wenn in dem XML-Dokument kein Attribut für die Schemaposition gefunden wird, findet keine Prüfung

statt. Bei Angabe der Klausel **USING SCHEMALOCATION HINTS** wird das SCH-Attribut der XML-Datenkennung (XDS) für alle XML-Spalten ignoriert.

Siehe hierzu die Beispiele im unten stehenden Abschnitt zur Option **XMLVALIDATE**.

Hinweise zur Verwendung

Achten Sie darauf, alle Tabellenoperationen abzuschließen und alle Sperren freizugeben, bevor Sie eine Importoperation starten. Dies erreichen Sie, indem Sie ein COMMIT absetzen, nachdem Sie alle geöffneten Cursor mit WITH HOLD geschlossen haben, oder indem Sie einen ROLLBACK 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 INSERT-Anweisung fehlschlägt, hat dies eine der beiden nachstehenden 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 Fehlermeldung in die Nachrichtendatei geschrieben, und die Verarbeitung wird angehalten.

Das Dienstprogramm führt ein automatisches COMMIT aus, nachdem die alten Zeilen während einer **REPLACE**- oder **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.

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

Aktualisierungen aus dem Befehl IMPORT werden stets am Ende einer Task IMPORT festgeschrieben. Während der Ausführung des Befehls IMPORT können auch automatische Commits erfolgen, um die Größe der Sperrenliste und den Speicherbereich für aktive Protokolldateien zu verringern. Der Befehl IMPORT wird mit ROLLBACK rückgängig gemacht, wenn das aktive Protokoll während der Verarbeitung von IMPORT voll wird.

- In der Standardeinstellung werden bei der Option **INSERT** oder **INSERT_UPDATE** keine automatischen Commits ausgeführt. Dies gilt jedoch nicht, wenn der Parameter **COMMITCOUNT** ungleich Null ist.
- Beim Offlineimport werden keine automatischen Commits 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 wird standardmäßig ein automatisches Commit durchgeführt, um sowohl den Speicherbereich für die aktive Protokolldatei als auch die Sperrenliste freizugeben. Auf automatische Commits wird nur verzichtet, wenn für **COMMITCOUNT** der Wert Null angegeben ist.

Immer dann, wenn das Dienstprogramm IMPORT ein Commit 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 einem erfolgreichen Commit geschrieben. Beim erneuten Starten der Importoperation nach einem Fehler geben Sie die Anzahl der zu überspringenden Datensätze an, die Sie anhand des letzten erfolgreichen Commits 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 für eine Objektabelle nicht 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 erneuten 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.

Wenn 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 ist nicht erforderlich, wenn die Quellen- und die Zieldatenbank von demselben Client aus zugänglich sind.

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-Datei dieselbe Codepage verwenden wie die Anwendung, 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-Clients mit dem Betriebssystem AIX.

Bei Tabellenobjekten auf einer 8-KB-Seite, die den Grenzwert von 1012 Spalten fast erreicht haben, kann der Import von PC/IXF-Datendateien dazu führen, dass DB2 einen Fehler zurückgibt, weil die maximale Größe für eine 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. 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 in DRDA-Server wie DB2 für OS/390, DB2 für VM und VSE und 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. Dabei 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.

Daten können nicht in eine Systemtabelle, eine erstellte temporäre Tabelle, eine deklarierte temporäre Tabelle oder eine Übersichtstabelle importiert werden.

Mit dem Dienstprogramm **IMPORT** können keine Sichten erstellt werden.

Das Importieren einer mehrteiligen PC/IXF-Datei, deren Bestandteile von einem Windows-System auf ein AIX-System kopiert werden, wird unterstützt. Im Befehl **IMPORT** muss nur der Name der ersten Teildatei angegeben werden. Beispiel: **IMPORT FROM data.ixf OF IXF INSERT INTO TABLE1**. Die Dateien data.002 usw. müssen im selben Verzeichnis wie data.ixf verfügbar sein.

Unter Windows-Betriebssystemen:

- 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.

Sicherheitskennsätze können in ihrem internen Format Zeilenvorschubzeichen enthalten. Wenn Sie die Datei mit dem Dateiformat 'DEL' importieren, können diese Zeilenvorschubzeichen als Begrenzer interpretiert werden. Wenn dieses Problem vorliegt, geben Sie den Änderungswert `delprioritychar` für den Dateityp im Befehl **IMPORT** an, um die vorherige Standardpriorität für Begrenzer zu verwenden.

Aspekte für föderierte Datenbanken

Bei Verwendung des Befehls **IMPORT** und der Befehlsparameter **INSERT**, **UPDATE** oder **INSERT_UPDATE** muss sichergestellt werden, dass die Berechtigung **CONTROL** für den betreffenden Kurznamen vorhanden ist. Weiter muss sichergestellt werden, dass der Kurzname, der bei einer Importoperation verwendet werden soll, bereits vorhanden ist. Außerdem sollten die im Abschnitt über die Parameter des Befehls **IMPORT** beschriebenen Einschränkungen beachtet werden.

Einige Datenquellen (z. B. ODBC) unterstützen nicht den Import in Kurznamen.

Änderungswerte für Dateityp für das Dienstprogramm **IMPORT**

Tabelle 14. Gültige Änderungswerte für Dateityp für das Dienstprogramm **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 die von COMMITCOUNT angegebene Anzahl Zeilen aufzunehmen oder die Anzahl Zeilen in der Datendatei, wenn COMMITCOUNT nicht angegeben ist. Es wird daher empfohlen, die Option COMMITCOUNT anzugeben, um das Überlaufen des Transaktionsprotokolls zu vermeiden.</p> <p>Dieser Änderungswert ist nicht kompatibel mit dem Modus INSERT_UPDATE, mit hierarchischen Tabellen sowie mit den folgenden Änderungswerten: <code>usedefaults</code>, <code>identitymissing</code>, <code>identityignore</code>, <code>generatedmissing</code> und <code>generatedignore</code>.</p>

Tabelle 14. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
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.
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. Wenn 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 <i>db2exp.001</i> 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. Die Klausel LOBS FROM impliziert die Aktivierung des Verhaltens von LOBSINFILE. Die Klausel LOBS FROM überträgt die Liste der Pfade an das Dienstprogramm IMPORT, die beim Importieren der Daten nach den 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 den nachfolgenden Aufruf einer Importoperation (mit diesem Änderungswert) zur Konvertierung der Daten in einer einzelnen untergeordneten Tabelle verwendet.

Tabelle 14. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
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"> • Wenn ein Standardwert für eine Spalte angegeben werden kann, wird der Standardwert geladen. • Wenn die Spalte Nullwerte enthalten darf und für diese Spalte kein Standardwert angegeben werden kann, wird ein Nullwert geladen. • Wenn die Spalte keine Nullwerte enthalten darf und kein Standardwert angegeben werden kann, wird ein Fehler zurückgegeben, und das Dienstprogramm beendet die Verarbeitung.
norowwarnings	Unterdrückt alle Warnungen zu zurückgewiesenen Zeilen.
rowchangetimestampignore	Dieser Änderungswert informiert das Dienstprogramm IMPORT darüber, dass für die Zeitmarkenspalte für Zeilenänderung in der Datendatei Daten vorhanden sind, aber ignoriert werden sollen. Dies bewirkt, dass alle Werte für ROW CHANGE TIMESTAMP durch das Dienstprogramm generiert werden. Dieses Verhalten ist für als GENERATED ALWAYS definierte Spalten und als GENERATED BY DEFAULT definierte Spalten gleich. Dies bedeutet, dass bei als GENERATED ALWAYS definierten Spalten keine Zeilen zurückgewiesen werden. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert rowchangetimestampmissing verwendet werden.
rowchangetimestampmissing	Wird dieser Änderungswert angegeben, geht das Dienstprogramm davon aus, dass die Eingabedatendatei keine Daten (nicht einmal Nullwerte) für die Zeitmarkenspalte für Zeilenänderung enthält, und generiert daher für jede Zeile einen Wert. Dieses Verhalten ist für als GENERATED ALWAYS definierte Spalten und als GENERATED BY DEFAULT definierte Spalten gleich. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert rowchangetimestampignore verwendet werden.
seclabelchar	<p>Gibt an, dass Sicherheitskennsätze in der Eingabequellendatei im Zeichenfolgeformat für Sicherheitskennsatzwerte vorliegen (nicht im codierten numerischen Standardformat). Von IMPORT wird jeder einzelne Sicherheitskennsatz beim Laden in das interne Format konvertiert. Liegt eine Zeichenfolge nicht im richtigen Format vor, wird die jeweilige Zeile nicht geladen und eine Warnung (SQLSTATE 01H53) zurückgegeben. Stellt eine Zeichenfolge nicht einen gültigen Sicherheitskennsatz dar, der Bestandteil der zum Schutz der Tabelle verwendeten Sicherheitsrichtlinie ist, wird die jeweilige Zeile nicht geladen und eine Warnung (SQLSTATE 01H53, SQLCODE SQL3243W) zurückgegeben.</p> <p>Dieser Änderungswert kann nicht angegeben werden, wenn der Änderungswert 'seclabelname' angegeben ist. Wird diese Kombination von Änderungswerten angegeben, schlägt der Import fehl, und es wird ein Fehler (SQLCODE SQL3525N) zurückgegeben.</p>

Tabelle 14. Gültige Änderungswerte für Dateityp für das Dienstprogramm **IMPORT**: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
seclabelname	<p>Gibt an, dass Sicherheitskennsätze in der Eingabequellendatei über ihren Namen angegeben werden und nicht im codierten numerischen Standardformat vorliegen. Von IMPORT wird der jeweilige Name in den entsprechenden Sicherheitskennsatz konvertiert, soweit vorhanden. Existiert kein Sicherheitskennsatz mit dem angegebenen Namen für die zum Schutz der Tabelle verwendete Sicherheitsrichtlinie, wird die betreffende Zeile nicht geladen und eine Warnung (SQLSTATE 01H53, SQLCODE SQL3244W) zurückgegeben.</p> <p>Dieser Änderungswert kann nicht angegeben werden, wenn der Änderungswert 'seclabelchar' angegeben ist. Wird diese Kombination von Änderungswerten angegeben, schlägt der Import fehl, und es wird ein Fehler (SQLCODE SQL3525N) zurückgegeben.</p> <p>Anmerkung: Ist der Dateityp ASC, werden alle auf den Namen des Sicherheitskennsatzes folgenden Leerzeichen als Teil des Namens interpretiert. Verwenden Sie den Änderungswert für Dateityp 'striptblanks', um sicherzustellen, dass die Leerzeichen entfernt werden.</p>
usedefaults	<p>Wenn eine Quellenspalte für eine Zieltabellenspalte angegeben wurde, jedoch für mindestens eine Zeileninstanz keine Daten enthält, werden Standardwerte geladen. Beispiele für fehlende Daten sind:</p> <ul style="list-style-type: none"> • Bei DEL-Dateien: Für einen Spaltenwert werden zwei benachbarte Spaltenbegrenzer ("„") oder zwei durch eine beliebige Anzahl Leerzeichen (" , ") getrennte, benachbarte Spaltenbegrenzer angegeben. • Bei DEL/ASC/WSF-Dateien: Eine Zeile, die für die ursprüngliche Angabe zu wenig Spalten enthält oder nicht lang genug ist. <p>Anmerkung: Bei Dateien des Typs ASC werden NULL-Spaltenwerte nicht als explizit fehlend betrachtet, und es wird kein Standardwert für NULL-Spaltenwerte eingesetzt. NULL-Spaltenwerte können durch alle Leerzeichen für numerische Spalten, Datenspalten, Zeitspalten und Zeitmarkenspalten oder durch die Verwendung des Nullindikators für Spalten eines beliebigen Typs zur Angabe eines Spaltenwerts von NULL dargestellt werden.</p> <p>Ohne diese Option geschieht Folgendes, wenn eine Quellenspalte keine Daten für eine Zeileninstanz enthält:</p> <ul style="list-style-type: none"> • Für DEL/ASC/WSF-Dateien: Wenn die Spalte Nullwerte enthalten darf, wird NULL geladen. Wenn die Spalte keine Nullwerte enthalten darf, wird die Zeile vom Dienstprogramm zurückgewiesen.

Tabelle 15. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: ASCII-Dateiformate (ASC/DEL)

Änderungswert	Beschreibung
codepage=x	<p><i>x</i> ist eine ASCII-Zeichenfolge. Der Wert wird als Codepage der Daten im Eingabedatensatz gewertet. Während der Importoperation werden Zeichendaten aus dieser Codepage in die Codepage der Anwendung konvertiert.</p> <p>Es gelten folgende Regeln:</p> <ul style="list-style-type: none"> • Bei reinem DBCS (Grafik), gemischtem DBCS und EUC sind Begrenzer auf den Bereich von x00 bis x3F (jeweils einschließlich) beschränkt. • 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. <p>Anmerkung:</p> <ol style="list-style-type: none"> 1. Der Änderungswert codepage kann nicht zusammen mit dem Änderungswert lobsinfile verwendet werden. 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.
dateformat="x"	<p><i>x</i> ist das Format des Datums in der Quelldatei.² 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 01 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 01 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>
implieddecimal	<p>Die Position eines implizierten Dezimalzeichens wird durch die Spaltendefinition bestimmt. Sie wird nicht mehr am Ende des Werts angenommen. Beispielsweise wird der Wert 12345 in eine DECIMAL(8,2)-Spalte als 123,45 geladen und <i>nicht</i> als 12345,00.</p>

Tabelle 15. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
timeformat="x"	<p><i>x</i> ist das Format der Zeitangabe in der Quellendatei.² Gültige Zeitelemente sind:</p> <ul style="list-style-type: none"> H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 - 24 bei einem 24-Stunden-System) HH - Stunde (zwei Ziffern von 00 bis 12 bei einem 12-Stunden-System und 00 - 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus) M - Minute (eine oder zwei Ziffern von 0 bis 59) MM - Minute (zwei Ziffern von 00 bis 59; dieses Element und das Element M schließen sich gegenseitig aus) S - Sekunde (eine oder zwei Ziffern von 0 bis 59) SS - Sekunde (zwei Ziffern von 00 bis 59; dieses Element und das Element S schließen sich gegenseitig aus) SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86400; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus) TT - Meridiananzeiger (AM oder PM) <p>Für jedes Element, das nicht angegeben wird, wird der Standardwert 0 zugeordnet. Einige Beispiele für Zeitformate:</p> <ul style="list-style-type: none"> "HH:MM:SS" "HH.MM TT" "SSSSS"

Tabelle 15. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
timestampformat="x"	<p>x ist das Format der Zeitmarke in der Quelldatei.² Gültige Zeitmarkenelemente sind:</p> <ul style="list-style-type: none"> YYYY - Jahr (vier Ziffern von 0000 bis 9999) M - Monat (eine oder zwei Ziffern von 1 bis 12) MM - Monat (zwei Ziffern von 01 bis 12; dieses Element und das Element M schließen sich gegenseitig aus) 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) D - Tag (eine oder zwei Ziffern von 1 bis 31) DD - Tag (zwei Ziffern von 01 bis 31; dieses Element und 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) H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System) HH - Stunde (zwei Ziffern von 00 bis 12 bei einem 12-Stunden-System und 00 bis 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus) M - Minute (ein oder zwei Ziffern von 0 bis 59) MM - Minute (zwei Ziffern von 00 bis 59; dieses Element und das Minutenelement M schließen sich gegenseitig aus) S - Sekunde (ein oder zwei Ziffern von 0 bis 59) SS - Sekunde (zwei Ziffern von 00 bis 59; dieses Element und das Element S schließen sich gegenseitig aus) SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86400; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus) U (1 bis 12 mal) - Bruchsekunden (die Häufigkeit von U gibt die Anzahl der Stellen mit Werten aus dem Bereich 0 bis 9 an) TT - Meridiananzeiger (AM oder PM) <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>

Tabelle 15. Gültige Änderungswerte für Dateityp für das Dienstprogramm **IMPORT**: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
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 Recovery nach einer DROP TABLE-Operation generiert wird, wenn die wiederherzustellende Tabelle Grafikdaten enthält.</p> <p>Einschränkungen</p> <p>Der Änderungswert usegraphiccodepage DARF NICHT für DEL-Dateien angegeben werden, die mit dem Dienstprogramm EXPORT erstellt wurden, da die in diesen Dateien enthaltenen Daten nur in einer Codepage codiert sind. Der Änderungswert usegraphiccodepage wird auch von den großen Doppelbytezeichenobjekten (DBCLOBs) in Dateien ignoriert.</p>
xmlchar	<p>Gibt an, dass XML-Dokumente in der Zeichencodepage codiert sind.</p> <p>Diese Option eignet sich für die Verarbeitung von XML-Dokumenten, die in der angegebenen Zeichencodepage codiert sind, ohne eine Codierdeklaration zu enthalten.</p> <p>Bei jedem Dokument muss die Codierung, soweit eine Deklarationskennung mit Codierungsattribut existiert, der Zeichencodepage entsprechen. Ist dies nicht der Fall, wird die Zeile mit dem Dokument zurückgewiesen. Dabei ist zu beachten, dass die Zeichencodepage dem Wert entspricht, der für den Änderungswert für Dateityp codepage angegeben ist, oder der Codepage der Anwendung, wenn dieser Wert nicht angegeben ist. Die Dokumente sind standardmäßig entweder in Unicode kodiert oder enthalten eine Deklarationskennung mit Codierungsattribut.</p>
xmlgraphic	<p>Gibt an, dass XML-Dokumente in der angegebenen Grafikcodepage codiert sind.</p> <p>Diese Option eignet sich für die Verarbeitung von XML-Dokumenten, die in einer bestimmten Grafikcodepage codiert sind, ohne eine Codierdeklaration zu enthalten.</p> <p>Bei jedem Dokument muss die Codierung, soweit eine Deklarationskennung mit Codierungsattribut existiert, der Grafikcodepage entsprechen. Ist dies nicht der Fall, wird die Zeile mit dem Dokument zurückgewiesen. Dabei ist zu beachten, dass die Grafikcodepage der Grafikkomponente des Werts entspricht, der über den Änderungswert für Dateityp codepage angegeben wird, oder der Grafikkomponente der Anwendungscodpage, wenn dieser Wert nicht angegeben ist. Dokumente sind standardmäßig entweder in Unicode kodiert oder enthalten eine Deklarationskennung mit Codierungsattribut.</p> <p>Anmerkung: Bei Angabe des Änderungswerts xmlgraphic mit dem Befehl IMPORT muss das zu importierende XML-Dokument in der Codepage UTF-16 codiert sein. Andernfalls wird das XML-Dokument möglicherweise mit einem Parsing-Fehler zurückgewiesen oder mit fehlerhaften Daten in die Tabelle importiert.</p>

Tabelle 16. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: ASC-Dateiformate (ASCII-Format mit universellen Zeilenbegrenzern)

Änderungswert	Beschreibung
nochecklengths	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.
nullindchar=x	<p>x ist ein einzelnes Zeichen. Diese Option ändert das Zeichen, das einen Nullwert bezeichnet, in x. Der Standardwert für x ist \backslash.³</p> <p>Bei diesem Änderungswert muss für EBCDIC-Datendateien die Groß-/Kleinschreibung beachtet werden, außer wenn das Zeichen ein englischer Buchstabe ist. Wenn beispielsweise für das NULL-Anzeigerzeichen der Buchstabe N angegeben wird, wird auch n als NULL-Anzeiger erkannt.</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>Im folgenden Beispiel bewirkt striptblanks, dass das Dienstprogramm IMPORT abschließende Leerzeichen abschneidet:</p> <pre>db2 import from myfile.asc of asc modified by striptblanks method 1 (1 10, 12 15) messages msgs.txt insert into staff</pre> <p>Diese Option kann nicht zusammen mit striptnulls angegeben werden. Diese Optionen schließen sich gegenseitig aus. Diese Option ersetzt die veraltete Option t, die nur noch aus Gründen der Abwärtskompatibilität 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. Diese Option ersetzt die veraltete Option padwithzero, die nur noch aus Gründen der Abwärtskompatibilität unterstützt wird.</p>

Tabelle 17. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: 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 anstelle von doppelten Anführungszeichen verwendet, um eine Zeichenfolge einzuschließen.³⁴ Falls Sie die doppelten Anführungszeichen explizit als Zeichenfolgebegrenzer angeben wollen, müssen Sie dies folgendermaßen tun:</p> <pre>modified by charde1""</pre> <p>Auch das einfache Anführungszeichen (') kann als Zeichenfolgebegrenzer angegeben werden. Im folgenden Beispiel bewirkt charde1'', dass das Dienstprogramm IMPORT jedes einfache Anführungszeichen (') als Zeichenfolgebegrenzer wertet:</p> <pre>db2 "import from myfile.del of del modified by charde1'' method p (1, 4) insert into staff (id, years)"</pre>
coldelx	<p><i>x</i> ist ein einzelnes Zeichen, das als Spaltenbegrenzer verwendet wird. Der Standardwert ist ein Komma (.). Das angegebene Zeichen wird anstelle eines Kommas verwendet, um das Ende einer Spalte anzugeben.³⁴</p> <p>Im folgenden Beispiel bewirkt coldel;, 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>
decplusblank	<p>Zeichen für Pluszeichen. Bewirkt, dass positiven Dezimalwerten statt eines Pluszeichens (+) ein Leerzeichen vorangestellt wird. Die Standardaktion besteht darin, positiven Dezimalwerten ein Pluszeichen voranzustellen.</p>
decptx	<p><i>x</i> ist ein einzelnes Ersatzzeichen für den Punkt als Dezimalzeichen. Der Standardwert ist ein Punkt (.). Das angegebene Zeichen wird anstelle eines Punkts als Dezimalzeichen verwendet.³⁴</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 charde1'' 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<zeilenbegrenzer> "Vincent,<zeilenbegrenzer>, is a manager", 4005,44.37<zeilenbegrenzer></pre> <p>Wenn der Änderungswert delprioritychar angegeben wird, gibt es in dieser Datendatei nur zwei Zeilen. Der zweite <zeilenbegrenzer> wird als Teil der ersten Datenspalte der zweiten Zeile gewertet, während der erste und dritte <zeilenbegrenzer> als eigentliche Satzbegrenzer gewertet werden. Wenn dieser Änderungswert <i>nicht</i> angegeben wird, gibt es in dieser Datendatei drei Zeilen, jeweils getrennt durch einen <zeilenbegrenzer>.</p>

Tabelle 17. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer) (Forts.)

Änderungswert	Beschreibung
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	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 mithilfe von DB2 exportiert wurden (es sei denn, nochardel war beim Exportieren angegeben). Sie wird bereitgestellt, um Datendateien von Herstellern zu unterstützen, die nicht über Zeichenbegrenzer verfügen. Eine unsachgemäße Verwendung kann zu Datenverlust oder Fehlern führen. Diese Option kann nicht zusammen mit chardelx, delprioritychar oder nodoubledel angegeben werden. Diese Optionen schließen sich gegenseitig aus.
nodoubledel	Unterdrückt die Erkennung von doppelten Zeichenbegrenzern.

Tabelle 18. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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 nochecklengths angegeben ist, wird keine Überprüfung durchgeführt und nach Möglichkeit jede Zeile geladen.
indexixf	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 ist (aber das Schlüsselwort <i>indexschema</i> ist angegeben), wird die Benutzer-ID der Verbindung verwendet. Wenn das Schlüsselwort nicht angegeben ist, wird das Schema in der IXF-Datei verwendet.
nochecklengths	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.
forcecreate	Gibt an, dass die Tabelle nach der Rückgabe von SQL3311N während einer Importoperation mit potenziell fehlenden oder begrenzten Informationen erstellt werden soll.

Tabelle 19. 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.

Tabelle 19. Verhalten des Dienstprogramms IMPORT bei Verwendung von 'codepage' und 'usegraphiccodepage' (Forts.)

codepage=N	usegraphiccodepage	Verhalten von IMPORT
Vorhanden	Nicht vorhanden	Es wird angenommen, dass sich alle Daten in der Datei in der Codepage N befinden. Warnung: 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. Warnung: 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. Warnung: Grafikdaten werden beim Importieren in die Datenbank beschädigt, wenn N eine Einzelbyte-Codepage ist.

Anmerkung:

1. Das Dienstprogramm IMPORT gibt keine Warnung aus, 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 Feldebegrenzer 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 Fehlernachricht, 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. Für die Dateitypänderungswerte `chardel`, `coldel` oder `decpt` angegebene Zeichenwerte müssen 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 ...
```
4. *Aspekte beim Versetzen von XML-Daten - Begrenzer* listet Einschränkungen für Zeichen auf, die anstelle der standardmäßigen Begrenzer verwendet werden können.
5. Die folgenden Dateitypänderungswerte sind beim Import in einen Kurznamen nicht zulässig:
 - `indexixf`
 - `indexschema`
 - `dldelfiletype`
 - `nodefaults`
 - `usedefaults`
 - `no_type_idfiletype`
 - `generatedignore`
 - `generatedmissing`
 - `identityignore`
 - `identitymissing`
 - `lobsinfile`
6. Das WSF-Format wird für XML-Spalten nicht unterstützt. Die Unterstützung für dieses Dateiformat ist auch veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.
7. Der Modus **CREATE** wird für XML-Spalten nicht unterstützt.
8. Alle XML-Daten müssen sich in XML-Dateien befinden, die von der Hauptdaten-datei getrennt sind. Für jede einzelne XML-Spalte muss in der Hauptdaten-datei eine XML-Datenkennung (XDS) oder ein Nullwert (NULL) vorliegen.
9. Bei XML-Dokumenten wird davon ausgegangen, dass sie im Unicode-Format vorliegen oder eine Deklarationskennung mit Codierungsattribut enthalten, sofern nicht der Änderungswert für Dateityp `XMLCHAR` oder `XMLGRAPHIC` angegeben ist.

10. Zeilen, die Dokumente enthalten, die nicht korrekt formatiert sind, werden zurückgewiesen.
11. Wenn die Option **XMLVALIDATE** angegeben wird, werden Dokumente, die erfolgreich anhand ihres übereinstimmenden Schemas überprüft werden konnten, während des Einfügens mit Annotationen versehen, die die Schemadaten enthalten. Zeilen, die Dokumente enthalten, deren Gültigkeitsprüfung auf der Basis des zugeordneten Schemas fehlgeschlagen ist, werden zurückgewiesen. Die Prüfung kann nur erfolgreich abgeschlossen werden, wenn der Benutzer, von dem der Import gestartet wurde, mindestens über eine der folgenden Berechtigungen verfügt:
 - Berechtigung DBADM
 - Berechtigung USAGE für das bei der Prüfung verwendete XML-Schema
12. Beim Import in eine Tabelle, die eine implizit ausgeblendete Zeitmarkenspalte für Zeilenänderung enthält, wird das Merkmal **IMPLICITLY HIDDEN** der Spalte nicht berücksichtigt. Daher *muss* der Dateitypänderungswert `rowchangetimestampmissing` im Importbefehl angegeben werden, wenn die zu importierenden Daten keine Daten für die Spalte enthalten und keine explizite Spaltenliste vorhanden ist.

Befehl **IMPORT** mit Prozedur **ADMIN_CMD**

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

Quick Link zu „Änderungswerte für Dateityp für das Dienstprogramm **IMPORT**“ auf Seite 112.

Berechtigung

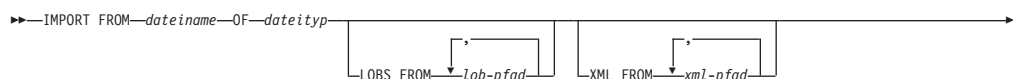
- Für die Verwendung von **IMPORT** mit der Option **INSERT** ist eine der folgenden Berechtigungen erforderlich:
 - Berechtigung DATAACCESS
 - Zugriffsrecht CONTROL für die einzelnen betroffenen Tabellen, Sichten oder Kurznamen
 - Zugriffsrechte INSERT und SELECT für jede betroffene Tabelle oder Sicht
- Für die Verwendung von **IMPORT** für eine bestehende Tabelle mit der Option **INSERT_UPDATE** ist eine der folgenden Berechtigungen erforderlich:
 - Berechtigung DATAACCESS
 - Zugriffsrecht CONTROL für die einzelnen betroffenen Tabellen, Sichten oder Kurznamen
 - Zugriffsrecht INSERT, SELECT, UPDATE und DELETE für die einzelnen betroffenen Tabellen oder Sichten
- Für die Verwendung von **IMPORT** für eine bestehende Tabelle mit der Option **REPLACE** oder **REPLACE_CREATE** ist eine der folgenden Berechtigungen erforderlich:
 - Berechtigung DATAACCESS
 - Zugriffsrecht CONTROL für die Tabelle oder 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:
 - Berechtigung DBADM

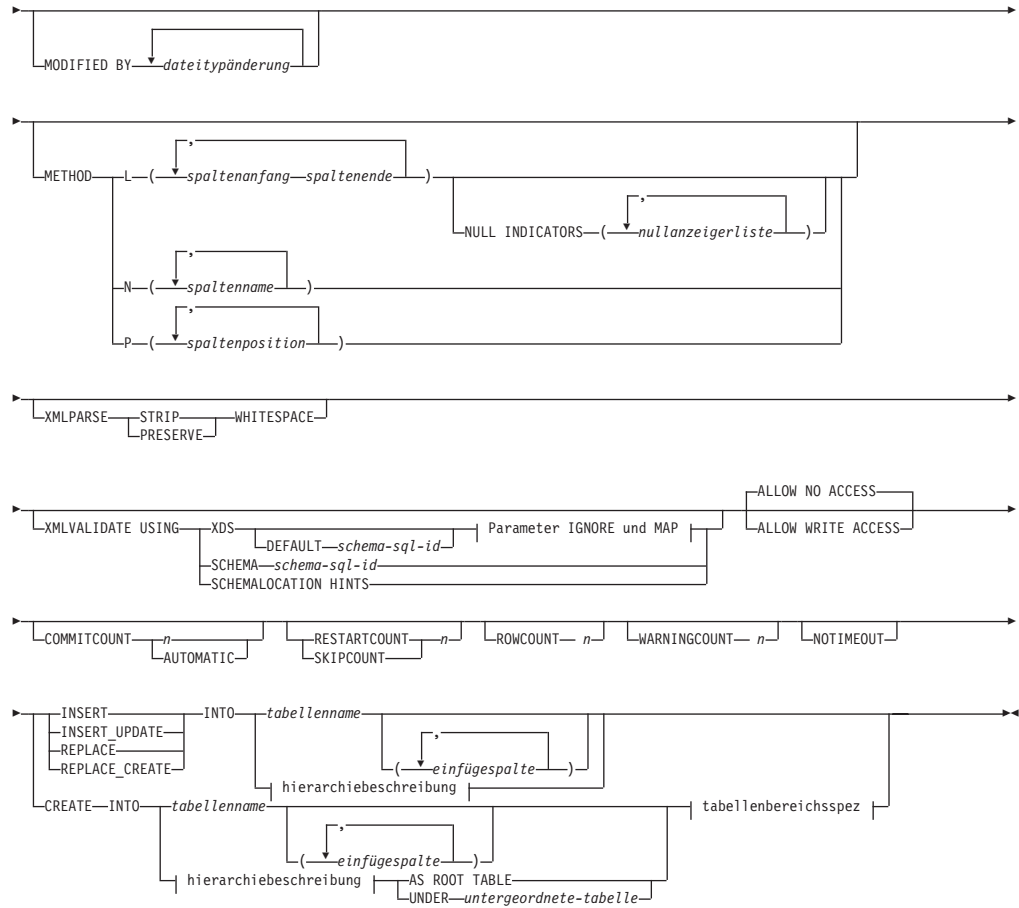
- Berechtigung **CREATETAB** für die Datenbank und das Zugriffsrecht **USE** für den Tabellenbereich sowie eine der folgenden Berechtigungen (bzw. Zugriffsrechte):
 - Berechtigung **IMPLICIT_SCHEMA** für die Datenbank, wenn der implizite oder explizite Schemaname der Tabelle nicht existiert
 - Zugriffsrecht **CREATEIN** für das Schema, wenn sich der Schemaname der Tabelle auf ein vorhandenes Schema bezieht
- Für die Verwendung von **IMPORT** in eine nicht vorhandene Hierarchie mit der Option **CREATE** oder **REPLACE_CREATE** ist eine der folgenden Berechtigungen erforderlich:
 - Berechtigung **DBADM**
 - Berechtigung **CREATETAB** für die Datenbank und Zugriffsrecht **USE** für den Tabellenbereich sowie eine der folgenden Berechtigungen:
 - Berechtigung **IMPLICIT_SCHEMA** für die Datenbank, falls der Name der Tabelle nicht vorhanden ist
 - Zugriffsrecht **CREATEIN** für das Schema, falls das Schema der Tabelle vorhanden ist
 - Zugriffsrecht **CONTROL** für jede untergeordnete Tabelle in der Hierarchie, wenn 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:
 - Berechtigung **DATAACCESS**
 - Zugriffsrecht **CONTROL** für alle untergeordneten Tabellen in der Hierarchie
- Um Daten in eine Tabelle importieren zu können, die geschützte Spalten enthält, muss die Berechtigungs-ID der Sitzung über LBAC-Berechtigungs nachweise verfügen, die Schreibzugriffe auf alle geschützten Spalten in der Tabelle zulassen. Andernfalls schlägt der Import fehl, und es wird ein Fehler (SQLSTATE 42512) zurückgegeben.
- Um Daten in eine Tabelle laden zu können, die geschützte Zeilen enthält, muss für die Berechtigungs-ID der Sitzung ein LBAC-Berechtigungs nachweis vorliegen, der den folgenden Kriterien entspricht:
 - Er ist Teil der Sicherheitsrichtlinie, die die Tabelle schützt
 - Er wurde der Berechtigungs-ID der Sitzung für den Schreibzugriff erteilt.

Der Kennsatz der einzufügenden Zeile, der LBAC-Berechtigungs nachweis des Benutzers und die LBAC-Regeln bestimmen den Kennsatz der Zeile.
- Wenn die Option **REPLACE** oder **REPLACE_CREATE** angegeben wird, muss die Berechtigungs-ID der Sitzung über die Berechtigung zum Löschen der Tabelle verfügen.
- Um Daten in einen Kurznamen importieren zu können, muss die Berechtigungs-ID der Sitzung über das Recht zum Zugriff auf eine angegebene Datenquelle und zu deren Verwendung im Durchgriffsmodus verfügen.

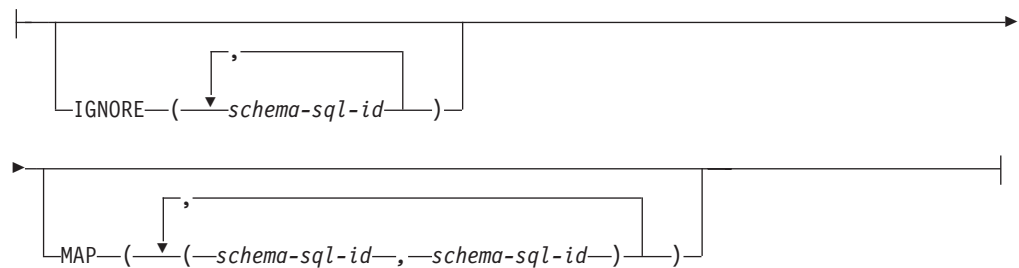
Erforderliche Verbindung

Befehlssyntax

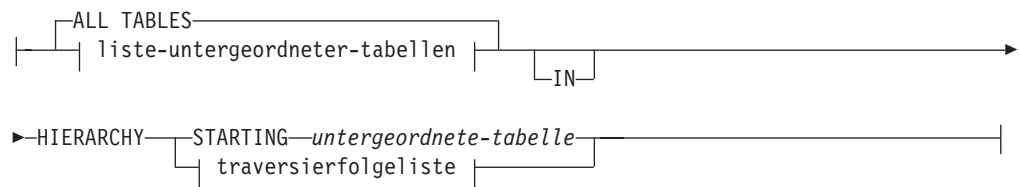




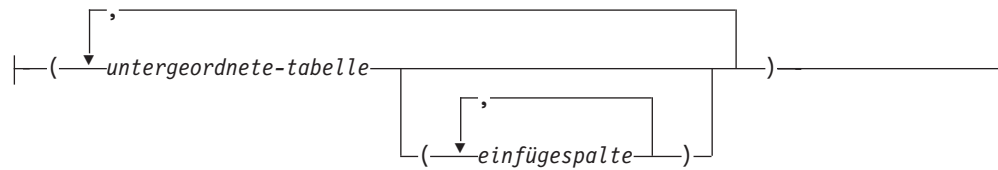
Parameter IGNORE und MAP:



hierarchiebeschreibung:



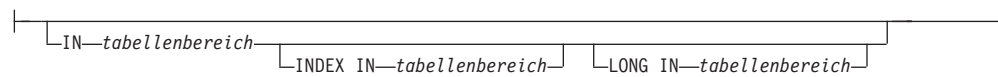
liste-untergeordneter-tabellen:



traversierfolgeliste:



tabellenbereichsspez:



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 Commits durch, um eine Sperreneskalation auf eine Tabellensperre zu vermeiden und um zu gewährleisten, dass die aktive Protokolldatei über ausreichend Speicher verfügt. Diese Commits werden auch dann durchgeführt, wenn die Option **COMMITCOUNT** nicht verwendet wurde. Bei jedem Commit verliert IMPORT die Tabellensperre IX und versucht, diese nach jedem Commit erneut anzufordern. Dieser Parameter ist erforderlich, wenn Daten in einen Kurznamen importiert werden. Für den Parameter **COMMITCOUNT** muss eine gültige Zahl angegeben werden (AUTOMATIC ist keine gültige Option).

AS ROOT TABLE

Erstellt eine oder mehrere untergeordnete Tabellen als eigenständige Tabellenhierarchie.

COMMITCOUNT *n* | AUTOMATIC

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

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

Wenn die Option **ALLOW WRITE ACCESS** angegeben und die Option **COMMITCOUNT** nicht angegeben wird, führt das Dienstprogramm IMPORT Commits genau so durch wie bei Angabe von **COMMITCOUNT AUTOMATIC**.

Die DB2-Registrierdatenbankvariable **DB2_FORCE_APP_ON_MAX_LOG** hat Auswirkungen auf die Funktionalität der Importoperation, zu gewährleisten, dass für die aktive Protokolldatei ausreichend Speicher verfügbar ist:

- Wenn die Option **DB2_FORCE_APP_ON_MAX_LOG** auf FALSE gesetzt ist und die Befehlsoption **COMMITCOUNT AUTOMATIC** angegeben ist, kann das Dienstprogramm IMPORT automatisch gewährleisten, dass für die aktive Protokolldatei ausreichend Speicher verfügbar ist.
- Wenn die Option **DB2_FORCE_APP_ON_MAX_LOG** auf FALSE gesetzt ist und die Befehlsoption **COMMITCOUNT *n*** angegeben ist, versucht das Dienstprogramm IMPORT, die Bedingung 'Protokoll voll' aufzulösen, wenn es beim Einfügen oder Aktualisieren eines Datensatzes den Fehler SQL0964C (Transaktionsprotokoll voll) feststellt. Das Dienstprogramm führt ein bedingungsloses Commit durch und wiederholt anschließend das Einfügen oder Aktualisieren des Datensatzes. Wenn das Problem auf diese Weise nicht behoben werden kann (z. B. wenn das Protokoll durch eine andere Datenbankaktivität belegt wird), schlägt der Befehl **IMPORT** wie erwartet fehl. Die Anzahl der festgeschriebenen Zeilen ist in diesem Fall jedoch möglicherweise kein Vielfaches des Werts **COMMITCOUNT *n***. Um zu verhindern, dass beim Wiederholen der Importoperation die bereits festgeschriebenen Zeilen verarbeitet werden, können Sie die Befehlsparameter **RESTARTCOUNT** oder **SKIPCOUNT** verwenden.
- Wenn die Option **DB2_FORCE_APP_ON_MAX_LOG** auf TRUE (Standardwert) gesetzt ist, schlägt die Importoperation fehl, wenn das Dienstprogramm beim Einfügen oder Aktualisieren eines Datensatzes den Fehler SQL0964C feststellt. Die Operation kann dabei unabhängig davon fehlschlagen, ob Sie den Parameter **COMMITCOUNT AUTOMATIC** oder **COMMITCOUNT *n*** angeben.

Die Anwendung wird zwangsweise von der Datenbank abgemeldet und es wird ein Rollback der aktuellen UOW (Unit of Work) durchgeführt. Um zu verhindern, dass beim Wiederholen der Importoperation die bereits festgeschriebenen Zeilen verarbeitet werden, können Sie die Befehlsparameter **RESTARTCOUNT** oder **SKIPCOUNT** verwenden.

CREATE

Anmerkung: Der Parameter **CREATE** ist veraltet und wird in einem künftigen Release möglicherweise entfernt. Weitere Informationen finden Sie unter „Optionen **CREATE** und **REPLACE_CREATE** des Dienstprogramms **IMPORT** sind veraltet“.

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

Dieser Parameter ist beim Importieren in einen Kurznamen nicht zulässig.

Anmerkung: Wenn Daten aus einer MVS-Hostdatenbank exportiert wurden und Felder vom Typ **LONGVAR** enthalten, deren auf Basis der Seitengröße berechnete Länge größer als 254 ist, schlägt **CREATE** möglicherweise fehl, weil die Zeilen zu lang sind. Eine Liste der Einschränkungen finden Sie unter „Neuerstellung von importierten Tabellen“. In diesem Fall sollte die Tabelle manuell erstellt und **IMPORT** mit **INSERT** aufgerufen oder alternativ der Befehl **LOAD** verwendet werden.

DEFAULT *schema-sql-id*

Diese Option kann nur verwendet werden, wenn der Parameter **USING XDS** angegeben wird. Das über die Klausel **DEFAULT** angegebene Schema gibt an, welches Schema für die Prüfung verwendet werden soll, wenn die XML-Datenkennung (XDS) eines importierten XML-Dokuments kein **SCH**-Attribut enthält, das ein XML-Schema identifiziert.

Die Klausel **DEFAULT** hat Vorrang vor den Klauseln **IGNORE** und **MAP**. Wenn eine XDS die Klausel **DEFAULT** erfüllt, werden die Angaben **IGNORE** und **MAP** ignoriert.

FROM *dateiname*

HIERARCHY

Gibt an, dass hierarchische Daten importiert werden sollen.

IGNORE *schema-sql-id*

Diese Option kann nur verwendet werden, wenn der Parameter **USING XDS** angegeben wird. Die Klausel **IGNORE** gibt eine Liste mit mindestens einem Schema an, das ignoriert werden soll, wenn es durch ein **SCH**-Attribut identifiziert wird. Wenn in der XML-Datenkennung ein **SCH**-Attribut für ein importiertes XML-Dokument enthalten ist und das von dem **SCH**-Attribut identifizierte Schema in der Liste der zu ignorierenden Schemata enthalten ist, wird für das importierte XML-Dokument keine Schemaprüfung durchgeführt.

Wenn in der Klausel **IGNORE** ein Schema angegeben ist, darf dieses nicht zusätzlich im linken Teil eines Schemapaars in der Klausel **MAP** enthalten sein.

Die Klausel **IGNORE** gilt nur für die XDS. Ein von der Klausel **MAP** zugeordnetes Schema wird anschließend nicht ignoriert, wenn es von der Klausel **IGNORE** angegeben wird.

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 ange-

geben 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 in der Klausel **IN** angegebene primäre Tabellenbereich 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 erstellte temporäre Tabelle, keine deklarierte temporäre Tabelle und keine Übersichtstabelle sein.

Für **INSERT**, **INSERT_UPDATE**, oder **REPLACE** kann ein Aliasname verwendet werden. Dies gilt nicht bei früheren Serverversionen, für die ein vollständig qualifizierter oder ein nicht qualifizierter Tabellename verwendet werden sollte. Ein qualifizierter Tabellename hat folgendes Format: *schema.tabellenname*. Dabei ist *schema* der Benutzername, mit dem die Tabelle erstellt wurde.

LOBS FROM *lob-pfad*

Die Namen der LOB-Datendateien werden in der Hauptdatendatei (`ASC`, `DEL` oder `IXF`) in der Spalte gespeichert, die in die LOB-Spalte geladen wird. Es können maximal 999 Pfade angegeben werden. Dadurch wird implizit das Verhalten `LOBSINFIL` aktiviert.

Dieser Parameter ist beim Importieren in einen Kurznamen nicht zulässig.

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 in der Klausel **IN** angegebene primäre Tabellenbereich ein DMS-Tabellenbereich ist. Der Tabellenbereich muss vorhanden und ein DMS-Tabellenbereich des Typs `LARGE` sein.

MAP *schema-sql-id*

Diese Option kann nur verwendet werden, wenn der Parameter **USING XDS** angegeben wird. Geben Sie über die Klausel **MAP** alternative Schemata an, die anstelle der Schemata verwendet werden sollen, die im Attribut `SCH` einer XML-Datenkennung (XDS) für jedes importierte XML-Dokument angegeben sind. Die Klausel **MAP** gibt eine Liste mit mindestens einem Sche-

mapaar an, in der jedes Paar eine Zuordnung eines Schemas zu einem anderen Schema darstellt. Das erste Schema in dem Zuordnungspaar stellt ein Schema dar, auf das durch ein SCH-Attribut in einer XDS verwiesen wird. Das zweite Schema in dem Zuordnungspaar gibt an, welches Schema für die Durchführung der Schemaprüfung verwendet werden soll.

Wenn im linken Teil eines Schemapaares in der Klausel **MAP** ein Schema angegeben ist, darf dieses nicht zusätzlich in der Klausel **IGNORE** angegeben werden.

Sobald eine Schemapaarzuordnung angewendet wurde, ist das Ergebnis endgültig. Die Zuordnungsoperation ist nicht transitiv, das gewählte Schema wird deshalb nicht anschließend auf eine andere Schemapaarzuordnung angewendet.

Ein Schema kann nicht mehrmals zugeordnet werden, d. h. ein Schema kann nur bei einem einzigen Paar als erstes Schema angegeben sein.

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 Spalten in der zu importierenden 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 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 *dateitypänderung*

Gibt Optionen für Änderungswerte für Dateityp an. Siehe „Änderungswerte für Dateityp für das Dienstprogramm IMPORT “ auf Seite 112.

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 wird. D. h., die Eingabedatei ist eine ASC-Datei. Die Nullanzei-

gerliste 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 einen Eintrag für jedes Datenfeld enthalten, das im Parameter **METHOD L** definiert wird. Eine Spaltennummer 0 gibt an, dass das entsprechende Datenfeld immer Daten enthält.

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

Das Zeichen für den Nullanzeiger kann über die Option **MODIFIED BY** mit dem Dateitypänderungswert `null indchar` geändert werden.

OF *dateityp*

Gibt das Format der Daten in der Eingabedatei an:

- ASC (ASCII-Format mit universellen Zeilenbegrenzern)
- DEL (ASCII-Format ohne universelle Zeilenbegrenzer) wird von verschiedenen Datenbankmanager- und Dateimanagerprogrammen verwendet
- WSF (Work Sheet Format) wird von verschiedenen Programmen wie den folgenden verwendet:
 - Lotus 1-2-3
 - Lotus Symphony
- IXF (Integration Exchange Format, PC-Version) ist ein Binärformat, das ausschließlich von DB2 verwendet wird.

Wichtig: Die Unterstützung für das WSF-Dateiformat ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.

Beim Importieren von Daten in einen Kurznamen wird der Dateityp WSF nicht unterstützt.

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. 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.

Dieser Parameter ist beim Importieren in einen Kurznamen nicht zulässig.

Diese Option berücksichtigt weder die Klausel NOT LOGGED INITIALLY (NLI) der Anweisung CREATE TABLE noch die Klausel ACTIVE NOT LOGGED INITIALLY der Anweisung ALTER TABLE.

Wenn innerhalb einer Transaktion mit einer Anweisung CREATE TABLE oder ALTER TABLE, bei der die Klausel NLI aufgerufen wird, ein Import mit der Option **REPLACE** ausgeführt wird, wird die Klausel NLI beim Import nicht berücksichtigt. Alle Einfügungen werden protokolliert.

Problemumgehung 1

Löschen Sie den Inhalt der Tabelle mit der Anweisung DELETE. Rufen Sie die Importoperation dann mit der Anweisung INSERT auf.

Problemumgehung 2

Löschen Sie die Tabelle, und erstellen Sie sie erneut. Rufen Sie die Importoperation dann mit der Anweisung INSERT auf.

Diese Einschränkung gilt für DB2 Universal Database Version 7 und DB2 UDB Version 8.

REPLACE_CREATE

Anmerkung: Der Parameter **REPLACE_CREATE** ist veraltet und wird in einem künftigen Release möglicherweise entfernt. Weitere Informationen finden Sie unter „Optionen CREATE und REPLACE_CREATE des Dienstprogramms IMPORT sind veraltet“.

Löscht, wenn die Tabelle vorhanden ist, alle vorhandenen Daten in 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 *Neuerstellung von importierten Tabellen*.

Diese Option kann nur mit IXF-Dateien verwendet werden. 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.

Dieser Parameter ist beim Importieren eines Kurznamens nicht zulässig.

RESTARTCOUNT *n*

Gibt an, dass eine Importoperation bei Datensatz $n + 1$ gestartet werden soll. Die ersten n Datensätze werden übersprungen. Diese Option entspricht in ihrer Funktionalität der Option **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. Erlaubt einem Benutzer das Importieren von genau n Zeilen aus einer Datei, ab dem durch die Option **SKIPCOUNT** oder **RESTARTCOUNT** festgelegten Datensatz. Wenn die Optionen **SKIPCOUNT** und **RESTARTCOUNT** nicht angegeben sind, werden die ersten n Zeilen importiert. Wenn **SKIPCOUNT** m oder **RESTARTCOUNT** m angegeben ist, werden die Zeilen $m+1$ bis $m+n$ importiert. Bei Verwendung von Compound-INSERT-Operationen wird eine benutzerdefinierte Angabe **ROWCOUNT** n auf das erste ganzzahlige Vielfache des Compound-Zählerwerts aufgerundet.

SKIPCOUNT *n*

Gibt an, dass eine Importoperation bei Datensatz $n + 1$ gestartet werden soll. Die ersten n Datensätze werden übersprungen. Diese Option entspricht in ihrer Funktionalität der Option **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** werden in einer Liste die Namen der untergeordneten Tabellen angegeben, in die Daten importiert werden sollen.

traversierfolgeliste

Für typisierte Tabellen mit der Option **INSERT**, **INSERT_UPDATE** oder **REPLACE** wird durch eine Liste mit den Namen untergeordneter Tabellen die Traversierfolge für die zu importierenden untergeordneten Tabellen in der Hierarchie angegeben.

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.

XML FROM *xml-pfad*

Gibt mindestens einen Pfad mit XML-Dateien an.

XMLPARSE

Gibt an, wie die XML-Dokumente syntaktisch analysiert werden. Wenn diese Option nicht angegeben ist, wird das Parsingverhalten für XML-Dokumente durch den Wert des Sonderregisters CURRENT XMLPARSE OPTION festgelegt.

STRIP WHITESPACE

Gibt an, dass Leerzeichen bei der Syntexanalyse des XML-Dokuments entfernt werden sollen.

PRESERVE WHITESPACE

Gibt an, dass Leerzeichen bei der Syntexanalyse des XML-Dokuments nicht entfernt werden sollen.

XMLVALIDATE

Gibt an, dass XML-Dokumente anhand eines Schemas geprüft werden sollen (soweit möglich).

USING XDS

XML-Dokumente werden anhand des XML-Schemas geprüft, das über die XDS in der Hauptdatendatei angegeben ist. Wenn die Option **XMLVALIDATE** mit der Klausel **USING XDS** aufgerufen wird, wird das für die Prüfung verwendete Schema standardmäßig durch das SCH-Attribut der XDS festgelegt. Enthält die XDS kein SCH-Attribut, wird eine Schemaprüfung nur durchgeführt, wenn die Klausel **DEFAULT** ein Standardschema angibt.

Mit den Klauseln **DEFAULT**, **IGNORE** und **MAP** kann das Verhalten der Schemafestlegung geändert werden. Diese optionalen Klauseln werden nicht auf einander, sondern direkt auf die XDS-Spezifikationen angewendet. Wird ein Schema beispielsweise ausgewählt, weil es von der Klausel **DEFAULT** angegeben wurde, so wird dieses Schema nicht ignoriert, wenn es auch von der Klausel **IGNORE** angegeben wird. Wird ein Schema ausgewählt, weil es als erster Teil eines Zuordnungspaars in der Klausel **MAP** angegeben wurde, so wird dieses Schema nicht erneut zugeordnet, wenn es auch im zweiten Teil des Zuordnungspaars einer anderen Klausel **MAP** angegeben ist.

USING SCHEMA *schema-sql-id*

XML-Dokumente werden anhand des XML-Schemas mit der angegebenen SQL-ID geprüft. In diesem Fall wird das Attribut SCH der XDS für alle XML-Spalten ignoriert.

USING SCHEMALOCATION HINTS

XML-Dokumente werden anhand des XML-Schemas geprüft, das über die Angaben zur XML-Schemaposition in den XML-Quelldokumenten angegeben ist. Wenn in dem XML-Dokument kein Attribut für die Schemaposition gefunden wird, findet keine Prüfung statt. Bei Angabe der Klausel **USING SCHEMALOCATION HINTS** wird das SCH-Attribut der XML-Datenkennung (XDS) für alle XML-Spalten ignoriert.

Siehe hierzu die Beispiele im unten stehenden Abschnitt zur Option **XMLVALIDATE**.

Hinweise zur Verwendung

Achten Sie darauf, alle Tabellenoperationen abzuschließen und alle Sperren freizugeben, bevor Sie eine Importoperation starten. Dies erreichen Sie, indem Sie ein COMMIT absetzen, nachdem Sie alle geöffneten Cursor mit WITH HOLD geschlossen haben, oder indem Sie einen ROLLBACK 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 INSERT-Anweisung fehlschlägt, hat dies eine der beiden nachstehenden 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 ein automatisches COMMIT aus, nachdem die alten Zeilen während einer **REPLACE**- oder **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.

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

Aktualisierungen aus dem Befehl **IMPORT** werden stets am Ende einer Task **IMPORT** festgeschrieben. Während der Ausführung des Befehls **IMPORT** können auch automatische Commits erfolgen, um die Größe der Sperrenliste und den Speicherbereich für aktive Protokolldateien zu verringern. Der Befehl **IMPORT** wird mit **ROLLBACK** rückgängig gemacht, wenn das aktive Protokoll während der Verarbeitung von **IMPORT** voll wird.

- In der Standardeinstellung werden bei der Option **INSERT** oder **INSERT_UPDATE** keine automatischen Commits ausgeführt. Dies gilt jedoch nicht, wenn der Parameter **COMMITCOUNT** ungleich Null ist.
- Beim Offlineimport werden keine automatischen Commits 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 wird standardmäßig ein automatisches Commit durchgeführt, um sowohl den Speicherbereich für die aktive Protokolldatei als auch die Sperrenliste freizugeben. Auf automatische Commits wird nur verzichtet, wenn für **COMMITCOUNT** der Wert Null angegeben ist.

Immer dann, wenn das Dienstprogramm **IMPORT** ein Commit 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 einem erfolgreichen Commit geschrieben. Beim erneuten Starten der Importoperation nach einem Fehler geben Sie die Anzahl der zu überspringenden Datensätze an, die Sie anhand des letzten erfolgreichen Commits 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 für eine Objekttable nicht 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 erneuten 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.

Wenn 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 ist nicht erforderlich, wenn die Quellen- und die Zieldatenbank von demselben Client aus zugänglich sind.

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-Datei dieselbe Codepage verwenden wie die Anwendung, 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-Clients mit dem Betriebssystem AIX.

Bei Tabellenobjekten auf einer 8-KB-Seite, die den Grenzwert von 1012 Spalten fast erreicht haben, kann der Import von PC/IXF-Datendateien dazu führen, dass DB2

einen Fehler zurückgibt, weil die maximale Größe für eine 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. 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 in DRDA-Server wie DB2 für OS/390, DB2 für VM und VSE und 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. Dabei 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.

Daten können nicht in eine Systemtabelle, eine erstellte temporäre Tabelle, eine deklarierte temporäre Tabelle oder eine Übersichtstabelle importiert werden.

Mit dem Dienstprogramm IMPORT können keine Sichten erstellt werden.

Das Importieren einer mehrteiligen PC/IXF-Datei, deren Bestandteile von einem Windows-System auf ein AIX-System kopiert werden, wird unterstützt. Im Befehl **IMPORT** muss nur der Name der ersten Teildatei angegeben werden. Beispiel: `IMPORT FROM data.ixf OF IXF INSERT INTO TABLE1`. Die Dateien `data.002` usw. müssen im selben Verzeichnis wie `data.ixf` verfügbar sein.

Unter Windows-Betriebssystemen:

- 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.

Sicherheitskennsätze können in ihrem internen Format Zeilenvorschubzeichen enthalten. Wenn Sie die Datei mit dem Dateiformat 'DEL' importieren, können diese Zeilenvorschubzeichen als Begrenzer interpretiert werden. Wenn dieses Problem vorliegt, geben Sie den Änderungswert `delprioritychar` für den Dateityp im Befehl **IMPORT** an, um die vorherige Standardpriorität für Begrenzer zu verwenden.

Aspekte für föderierte Datenbanken

Bei Verwendung des Befehls **IMPORT** und der Befehlsparameter **INSERT**, **UPDATE** oder **INSERT_UPDATE** muss sichergestellt werden, dass die Berechtigung **CONTROL** für den betreffenden Kurznamen vorhanden ist. Weiter muss sichergestellt werden, dass der Kurzname, der bei einer Importoperation verwendet werden soll, bereits vorhanden ist. Außerdem sollten die im Abschnitt über die Parameter des Befehls **IMPORT** beschriebenen Einschränkungen beachtet werden.

Einige Datenquellen (z. B. ODBC) unterstützen nicht den Import in Kurznamen.

Änderungswerte für Dateityp für das Dienstprogramm IMPORT

Tabelle 20. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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 die von COMMITCOUNT angegebene Anzahl Zeilen aufzunehmen oder die Anzahl Zeilen in der Datendatei, wenn COMMITCOUNT nicht angegeben ist. Es wird daher empfohlen, die Option COMMITCOUNT anzugeben, um das Überlaufen des Transaktionsprotokolls zu vermeiden.</p> <p>Dieser Änderungswert ist nicht kompatibel mit dem Modus INSERT_UPDATE, mit hierarchischen Tabellen sowie mit den folgenden Änderungswerten: 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 20. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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 Zeichenfolgenderdarstellung 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. Wenn 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 <i>db2exp.001</i> und ist 456 Byte lang.</p> <p>Die Klausel LOBS FROM gibt an, wo sich die LOB-Dateien befinden, wenn der Änderungswert „<i>lobsinfile</i>“ verwendet wird. Die Klausel LOBS FROM impliziert die Aktivierung des Verhaltens von LOBSINFILE. Die Klausel LOBS FROM überträgt die Liste der Pfade an das Dienstprogramm IMPORT, die beim Importieren der Daten nach den 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 den 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"> • Wenn ein Standardwert für eine Spalte angegeben werden kann, wird der Standardwert geladen. • Wenn die Spalte Nullwerte enthalten darf und für diese Spalte kein Standardwert angegeben werden kann, wird ein Nullwert geladen. • Wenn die Spalte keine Nullwerte enthalten darf und kein Standardwert angegeben werden kann, wird ein Fehler zurückgegeben, und das Dienstprogramm beendet die Verarbeitung.
norowwarnings	Unterdrückt alle Warnungen zu zurückgewiesenen Zeilen.
rowchangetimestampignore	Dieser Änderungswert informiert das Dienstprogramm IMPORT darüber, dass für die Zeitmarkenspalte für Zeilenänderung in der Datendatei Daten vorhanden sind, aber ignoriert werden sollen. Dies bewirkt, dass alle Werte für ROW CHANGE TIMESTAMP durch das Dienstprogramm generiert werden. Dieses Verhalten ist für als GENERATED ALWAYS definierte Spalten und als GENERATED BY DEFAULT definierte Spalten gleich. Dies bedeutet, dass bei als GENERATED ALWAYS definierten Spalten keine Zeilen zurückgewiesen werden. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert <i>rowchangetimestampmissing</i> verwendet werden.

Tabelle 20. Gültige Änderungswerte für Dateityp für das Dienstprogramm **IMPORT**: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
rowchangetimestampmissing	<p>Wird dieser Änderungswert angegeben, geht das Dienstprogramm davon aus, dass die Eingabedatendatei keine Daten (nicht einmal Nullwerte) für die Zeitmarkenspalte für Zeilenänderung enthält, und generiert daher für jede Zeile einen Wert. Dieses Verhalten ist für als GENERATED ALWAYS definierte Spalten und als GENERATED BY DEFAULT definierte Spalten gleich. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert rowchangetimestampignore verwendet werden.</p>
seclabelchar	<p>Gibt an, dass Sicherheitskennsätze in der Eingabequellendatei im Zeichenfolgeformat für Sicherheitskennsatzwerte vorliegen (nicht im codierten numerischen Standardformat). Von IMPORT wird jeder einzelne Sicherheitskennsatz beim Laden in das interne Format konvertiert. Liegt eine Zeichenfolge nicht im richtigen Format vor, wird die jeweilige Zeile nicht geladen und eine Warnung (SQLSTATE 01H53) zurückgegeben. Stellt eine Zeichenfolge nicht einen gültigen Sicherheitskennsatz dar, der Bestandteil der zum Schutz der Tabelle verwendeten Sicherheitsrichtlinie ist, wird die jeweilige Zeile nicht geladen und eine Warnung (SQLSTATE 01H53, SQLCODE SQL3243W) zurückgegeben.</p> <p>Dieser Änderungswert kann nicht angegeben werden, wenn der Änderungswert 'seclabelname' angegeben ist. Wird diese Kombination von Änderungswerten angegeben, schlägt der Import fehl, und es wird ein Fehler (SQLCODE SQL3525N) zurückgegeben.</p>
seclabelname	<p>Gibt an, dass Sicherheitskennsätze in der Eingabequellendatei über ihren Namen angegeben werden und nicht im codierten numerischen Standardformat vorliegen. Von IMPORT wird der jeweilige Name in den entsprechenden Sicherheitskennsatz konvertiert, soweit vorhanden. Existiert kein Sicherheitskennsatz mit dem angegebenen Namen für die zum Schutz der Tabelle verwendeten Sicherheitsrichtlinie, wird die betreffende Zeile nicht geladen und eine Warnung (SQLSTATE 01H53, SQLCODE SQL3244W) zurückgegeben.</p> <p>Dieser Änderungswert kann nicht angegeben werden, wenn der Änderungswert 'seclabelchar' angegeben ist. Wird diese Kombination von Änderungswerten angegeben, schlägt der Import fehl, und es wird ein Fehler (SQLCODE SQL3525N) zurückgegeben.</p> <p>Anmerkung: Ist der Dateityp ASC, werden alle auf den Namen des Sicherheitskennsatzes folgenden Leerzeichen als Teil des Namens interpretiert. Verwenden Sie den Änderungswert für Dateityp 'striptblanks', um sicherzustellen, dass die Leerzeichen entfernt werden.</p>

Tabelle 20. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
usedefaults	<p>Wenn eine Quellenspalte für eine Zieltabellenspalte angegeben wurde, jedoch für mindestens eine Zeileninstanz keine Daten enthält, werden Standardwerte geladen. Beispiele für fehlende Daten sind:</p> <ul style="list-style-type: none"> • Bei DEL-Dateien: Für einen Spaltenwert werden zwei benachbarte Spaltenbegrenzer (",,") oder zwei durch eine beliebige Anzahl Leerzeichen (" , ") getrennte, benachbarte Spaltenbegrenzer angegeben. • Bei DEL/ASC/WSF-Dateien: Eine Zeile, die für die ursprüngliche Angabe zu wenig Spalten enthält oder nicht lang genug ist. <p>Anmerkung: Bei Dateien des Typs ASC werden NULL-Spaltenwerte nicht als explizit fehlend betrachtet, und es wird kein Standardwert für NULL-Spaltenwerte eingesetzt. NULL-Spaltenwerte können durch alle Leerzeichen für numerische Spalten, Datenspalten, Zeitspalten und Zeitmarkenspalten oder durch die Verwendung des Nullindikators für Spalten eines beliebigen Typs zur Angabe eines Spaltenwerts von NULL dargestellt werden.</p> <p>Ohne diese Option geschieht Folgendes, wenn eine Quellenspalte keine Daten für eine Zeileninstanz enthält:</p> <ul style="list-style-type: none"> • Für DEL/ASC/WSF-Dateien: Wenn die Spalte Nullwerte enthalten darf, wird NULL geladen. Wenn die Spalte keine Nullwerte enthalten darf, wird die Zeile vom Dienstprogramm zurückgewiesen.

Tabelle 21. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: ASCII-Dateiformate (ASC/DEL)

Änderungswert	Beschreibung
codepage=x	<p>x ist eine ASCII-Zeichenfolge. Der Wert wird als Codepage der Daten im Eingabedatensatz gewertet. Während der Importoperation werden Zeichendaten aus dieser Codepage in die Codepage der Anwendung konvertiert.</p> <p>Es gelten folgende Regeln:</p> <ul style="list-style-type: none"> • Bei reinem DBCS (Grafik), gemischtem DBCS und EUC sind Begrenzer auf den Bereich von x00 bis x3F (jeweils einschließlich) beschränkt. • 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. <p>Anmerkung:</p> <ol style="list-style-type: none"> 1. Der Änderungswert codepage kann nicht zusammen mit dem Änderungswert lobsinfile verwendet werden. 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.

Tabelle 21. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
dateformat="x"	<p>x ist das Format des Datums in der Quelldatei.² Gültige Datumelemente 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 01 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 01 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: "D-M-YYYY" "MM.DD.YYYY" "YYYYDDD"</p>
implieddecimal	<p>Die Position eines implizierten Dezimalzeichens wird durch die Spaltendefinition bestimmt. Sie wird nicht mehr am Ende des Werts angenommen. Beispielsweise wird der Wert 12345 in eine DECIMAL(8,2)-Spalte als 123,45 geladen und <i>nicht</i> als 12345,00.</p>
timeformat="x"	<p>x ist das Format der Zeitangabe in der Quelldatei.² Gültige Zeitelemente sind:</p> <p>H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 - 24 bei einem 24-Stunden-System) HH - Stunde (zwei Ziffern von 00 bis 12 bei einem 12-Stunden-System und 00 - 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus) M - Minute (eine oder zwei Ziffern von 0 bis 59) MM - Minute (zwei Ziffern von 00 bis 59; dieses Element und das Element M schließen sich gegenseitig aus) S - Sekunde (eine oder zwei Ziffern von 0 bis 59) SS - Sekunde (zwei Ziffern von 00 bis 59; dieses Element und das Element S schließen sich gegenseitig aus) SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86400; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus) TT - Meridiananzeiger (AM oder PM)</p> <p>Für jedes Element, das nicht angegeben wird, wird der Standardwert 0 zugeordnet. Einige Beispiele für Zeitformate: "HH:MM:SS" "HH.MM TT" "SSSSS"</p>

Tabelle 21. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
timestampformat="x"	<p>x ist das Format der Zeitmarke in der Quelldatei.² Gültige Zeitmarkenelemente 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 01 bis 12; dieses Element und das Element M schließen sich gegenseitig aus) 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) D - Tag (eine oder zwei Ziffern von 1 bis 31) DD - Tag (zwei Ziffern von 01 bis 31; dieses Element und 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) H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System) HH - Stunde (zwei Ziffern von 00 bis 12 bei einem 12-Stunden-System und 00 bis 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus) M - Minute (ein oder zwei Ziffern von 0 bis 59) MM - Minute (zwei Ziffern von 00 bis 59; dieses Element und das Minutenelement M schließen sich gegenseitig aus) S - Sekunde (ein oder zwei Ziffern von 0 bis 59) SS - Sekunde (zwei Ziffern von 00 bis 59; dieses Element und das Element S schließen sich gegenseitig aus) SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86400; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus) U (1 bis 12 mal) - Bruchsekunden (die Häufigkeit von U gibt die Anzahl der Stellen mit Werten aus dem Bereich 0 bis 9 an) TT - Meridiananzeiger (AM oder PM)</p> <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: "YYYY/MM/DD HH:MM:SS.UUUUUU"</p> <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>

Tabelle 21. Gültige Änderungswerte für Dateityp für das Dienstprogramm **IMPORT**: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
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 Recovery nach einer DROP TABLE-Operation generiert wird, wenn die wiederherzustellende Tabelle Grafikdaten enthält.</p> <p>Einschränkungen</p> <p>Der Änderungswert usegraphiccodepage DARF NICHT für DEL-Dateien angegeben werden, die mit dem Dienstprogramm EXPORT erstellt wurden, da die in diesen Dateien enthaltenen Daten nur in einer Codepage codiert sind. Der Änderungswert usegraphiccodepage wird auch von den großen Doppelbytezeichenobjekten (DBCLOBs) in Dateien ignoriert.</p>
xmlchar	<p>Gibt an, dass XML-Dokumente in der Zeichencodepage codiert sind.</p> <p>Diese Option eignet sich für die Verarbeitung von XML-Dokumenten, die in der angegebenen Zeichencodepage codiert sind, ohne eine Codierdeklaration zu enthalten.</p> <p>Bei jedem Dokument muss die Codierung, soweit eine Deklarationskennung mit Codierungsattribut existiert, der Zeichencodepage entsprechen. Ist dies nicht der Fall, wird die Zeile mit dem Dokument zurückgewiesen. Dabei ist zu beachten, dass die Zeichencodepage dem Wert entspricht, der für den Änderungswert für Dateityp codepage angegeben ist, oder der Codepage der Anwendung, wenn dieser Wert nicht angegeben ist. Die Dokumente sind standardmäßig entweder in Unicode kodiert oder enthalten eine Deklarationskennung mit Codierungsattribut.</p>
xmlgraphic	<p>Gibt an, dass XML-Dokumente in der angegebenen Grafikcodepage codiert sind.</p> <p>Diese Option eignet sich für die Verarbeitung von XML-Dokumenten, die in einer bestimmten Grafikcodepage codiert sind, ohne eine Codierdeklaration zu enthalten.</p> <p>Bei jedem Dokument muss die Codierung, soweit eine Deklarationskennung mit Codierungsattribut existiert, der Grafikcodepage entsprechen. Ist dies nicht der Fall, wird die Zeile mit dem Dokument zurückgewiesen. Dabei ist zu beachten, dass die Grafikcodepage der Grafikkomponente des Werts entspricht, der über den Änderungswert für Dateityp codepage angegeben wird, oder der Grafikkomponente der Anwendungscodpage, wenn dieser Wert nicht angegeben ist. Dokumente sind standardmäßig entweder in Unicode kodiert oder enthalten eine Deklarationskennung mit Codierungsattribut.</p> <p>Anmerkung: Bei Angabe des Änderungswerts xmlgraphic mit dem Befehl IMPORT muss das zu importierende XML-Dokument in der Codepage UTF-16 codiert sein. Andernfalls wird das XML-Dokument möglicherweise mit einem Parsing-Fehler zurückgewiesen oder mit fehlerhaften Daten in die Tabelle importiert.</p>

Tabelle 22. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: ASC-Dateiformate (ASCII-Format mit universellen Zeilenbegrenzern)

Änderungswert	Beschreibung
nochecklengths	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.
nullindchar= <i>x</i>	<i>x</i> ist ein einzelnes Zeichen. Diese Option ändert das Zeichen, das einen Nullwert bezeichnet, in <i>x</i> . Der Standardwert für <i>x</i> ist Y. ³ Bei diesem Änderungswert muss für EBCDIC-Datendateien die Groß-/Kleinschreibung beachtet werden, außer wenn das Zeichen ein englischer Buchstabe ist. Wenn beispielsweise für das NULL-Anzeigerzeichen der Buchstabe N angegeben wird, wird auch n als NULL-Anzeiger erkannt.
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	Schneidet abschließende Leerzeichen ab, wenn Daten in ein Feld mit variabler Länge geladen werden. Wenn diese Option nicht angegeben wird, werden Leerstellen beibehalten. Im folgenden Beispiel bewirkt striptblanks, dass das Dienstprogramm IMPORT abschließende Leerzeichen abschneidet: <pre>db2 import from myfile.asc of asc modified by striptblanks method 1 (1 10, 12 15) messages msgs.txt insert into staff</pre> Diese Option kann nicht zusammen mit striptnulls angegeben werden. Diese Optionen schließen sich gegenseitig aus. Diese Option ersetzt die veraltete Option t, die nur noch aus Gründen der Abwärtskompatibilität unterstützt wird.
striptnulls	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. Diese Option kann nicht zusammen mit striptblanks angegeben werden. Diese Optionen schließen sich gegenseitig aus. Diese Option ersetzt die veraltete Option padwithzero, die nur noch aus Gründen der Abwärtskompatibilität unterstützt wird.

Tabelle 23. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer)

Änderungswert	Beschreibung
chardelx	<p>x ist ein einzelner Zeichenfolgebegrenzer. Der Standardwert ist ein doppeltes Anführungszeichen ("). Das angegebene Zeichen wird anstelle von doppelten Anführungszeichen verwendet, um eine Zeichenfolge einzuschließen.³⁴ Falls Sie die doppelten Anführungszeichen explizit als Zeichenfolgebegrenzer angeben wollen, müssen Sie dies folgendermaßen tun:</p> <pre>modified by charde1""</pre> <p>Auch das einfache Anführungszeichen (') kann als Zeichenfolgebegrenzer angegeben werden. Im folgenden Beispiel bewirkt charde1'', dass das Dienstprogramm IMPORT jedes einfache Anführungszeichen (') als Zeichenfolgebegrenzer wertet:</p> <pre>db2 "import from myfile.del of del modified by charde1'' method p (1, 4) insert into staff (id, years)"</pre>
coldelx	<p>x ist ein einzelnes Zeichen, das als Spaltenbegrenzer verwendet wird. Der Standardwert ist ein Komma (.). Das angegebene Zeichen wird anstelle eines Kommas verwendet, um das Ende einer Spalte anzugeben.³⁴</p> <p>Im folgenden Beispiel bewirkt coldel;, 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>
decplusblank	<p>Zeichen für Pluszeichen. Bewirkt, dass positiven Dezimalwerten statt eines Pluszeichens (+) ein Leerzeichen vorangestellt wird. Die Standardaktion besteht darin, positiven Dezimalwerten ein Pluszeichen voranzustellen.</p>
decptx	<p>x ist ein einzelnes Ersatzzeichen für den Punkt als Dezimalzeichen. Der Standardwert ist ein Punkt (.). Das angegebene Zeichen wird anstelle eines Punkts als Dezimalzeichen verwendet.³⁴</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 charde1'' 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<zeilenbegrenzer> "Vincent,<zeilenbegrenzer>, is a manager", 4005,44.37<zeilenbegrenzer></pre> <p>Wenn der Änderungswert delprioritychar angegeben wird, gibt es in dieser Datendatei nur zwei Zeilen. Der zweite <zeilenbegrenzer> wird als Teil der ersten Datenspalte der zweiten Zeile gewertet, während der erste und dritte <zeilenbegrenzer> als eigentliche Satzbegrenzer gewertet werden. Wenn dieser Änderungswert <i>nicht</i> angegeben wird, gibt es in dieser Datendatei drei Zeilen, jeweils getrennt durch einen <zeilenbegrenzer>.</p>

Tabelle 23. Gültige Änderungswerte für Dateityp für das Dienstprogramm IMPORT: DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer) (Forts.)

Änderungswert	Beschreibung
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	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 mithilfe von DB2 exportiert wurden (es sei denn, nochardel war beim Exportieren angegeben). Sie wird bereitgestellt, um Datendateien von Herstellern zu unterstützen, die nicht über Zeichenbegrenzer verfügen. Eine unsachgemäße Verwendung kann zu Datenverlust oder Fehlern führen. Diese Option kann nicht zusammen mit chardelx, delprioritychar oder nodoubleadelangegeben werden. Diese Optionen schließen sich gegenseitig aus.
nodoubledel	Unterdrückt die Erkennung von doppelten Zeichenbegrenzern.

Tabelle 24. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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 nochecklengths angegeben ist, wird keine Überprüfung durchgeführt und nach Möglichkeit jede Zeile geladen.
indexixf	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 ist (aber das Schlüsselwort <i>indexschema</i> ist angegeben), wird die Benutzer-ID der Verbindung verwendet. Wenn das Schlüsselwort nicht angegeben ist, wird das Schema in der IXF-Datei verwendet.
nochecklengths	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.
forcecreate	Gibt an, dass die Tabelle nach der Rückgabe von SQL3311N während einer Importoperation mit potenziell fehlenden oder begrenzten Informationen erstellt werden soll.

Tabelle 25. 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.

Tabelle 25. Verhalten des Dienstprogramms IMPORT bei Verwendung von 'codepage' und 'usegraphiccodepage' (Forts.)

codepage=N	usegraphiccodepage	Verhalten von IMPORT
Vorhanden	Nicht vorhanden	Es wird angenommen, dass sich alle Daten in der Datei in der Codepage N befinden. Warnung: 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. Warnung: 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. Warnung: Grafikdaten werden beim Importieren in die Datenbank beschädigt, wenn N eine Einzelbyte-Codepage ist.

Anmerkung:

1. Das Dienstprogramm IMPORT gibt keine Warnung aus, 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 Feldebegrenzer 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 Fehlernachricht, 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. Für die Dateitypänderungswerte `chardel`, `coldel` oder `decpt` angegebene Zeichenwerte müssen 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 ...
```

4. *Aspekte beim Versetzen von XML-Daten - Begrenzer* listet Einschränkungen für Zeichen auf, die anstelle der standardmäßigen Begrenzer verwendet werden können.
5. Die folgenden Dateitypänderungswerte sind beim Import in einen Kurznamen nicht zulässig:
 - `indexixf`
 - `indexschema`
 - `dldelfiletype`
 - `nodefaults`
 - `usedefaults`
 - `no_type_idfiletype`
 - `generatedignore`
 - `generatedmissing`
 - `identityignore`
 - `identitymissing`
 - `lobsinfile`
6. Das WSF-Format wird für XML-Spalten nicht unterstützt. Die Unterstützung für dieses Dateiformat ist auch veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.
7. Der Modus **CREATE** wird für XML-Spalten nicht unterstützt.
8. Alle XML-Daten müssen sich in XML-Dateien befinden, die von der Hauptdaten-datei getrennt sind. Für jede einzelne XML-Spalte muss in der Hauptdaten-datei eine XML-Datenkennung (XDS) oder ein Nullwert (NULL) vorliegen.
9. Bei XML-Dokumenten wird davon ausgegangen, dass sie im Unicode-Format vorliegen oder eine Deklarationskennung mit Codierungsattribut enthalten, sofern nicht der Änderungswert für Dateityp XMLCHAR oder XMLGRAPHIC angegeben ist.

10. Zeilen, die Dokumente enthalten, die nicht korrekt formatiert sind, werden zurückgewiesen.
11. Wenn die Option **XMLVALIDATE** angegeben wird, werden Dokumente, die erfolgreich anhand ihres übereinstimmenden Schemas überprüft werden konnten, während des Einfügens mit Annotationen versehen, die die Schemadaten enthalten. Zeilen, die Dokumente enthalten, deren Gültigkeitsprüfung auf der Basis des zugeordneten Schemas fehlgeschlagen ist, werden zurückgewiesen. Die Prüfung kann nur erfolgreich abgeschlossen werden, wenn der Benutzer, von dem der Import gestartet wurde, mindestens über eine der folgenden Berechtigungen verfügt:
 - Berechtigung DBADM
 - Berechtigung USAGE für das bei der Prüfung verwendete XML-Schema
12. Beim Import in eine Tabelle, die eine implizit ausgeblendete Zeitmarkenspalte für Zeilenänderung enthält, wird das Merkmal **IMPLICITLY HIDDEN** der Spalte nicht berücksichtigt. Daher *muss* der Dateitypänderungswert `rowchangenestampmissing` im Importbefehl angegeben werden, wenn die zu importierenden Daten keine Daten für die Spalte enthalten und keine explizite Spaltenliste vorhanden ist.

db2Import - Importieren von Daten in Tabellen, Hierarchien, Kurznamen oder Sichten

Fügt Daten aus einer externen Datei mit einem unterstützten Dateiformat in eine Tabelle, eine Hierarchie, einen Kurznamen oder eine Sicht ein. Das Dienstprogramm **LOAD** ist schneller als diese Funktion. Allerdings unterstützt das Dienstprogramm **LOAD** nicht das Laden von Daten auf Hierarchieebene bzw. nicht das Laden in einen Kurznamen.

Berechtigung

- Für die Verwendung von **IMPORT** mit der Option **INSERT** ist eine der folgenden Berechtigungen erforderlich:
 - **DATAACCESS**
 - Zugriffsrecht **CONTROL** für alle betroffenen Tabellen, Sichten oder Kurznamen
 - Zugriffsrechte **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:
 - **DATAACCESS**
 - Zugriffsrecht **CONTROL** für die Tabelle, die Sicht oder den Kurznamen
 - Zugriffsrecht **INSERT**, **SELECT**, **UPDATE** und **DELETE** für die einzelnen betroffenen Tabellen oder Sichten
- Für die Verwendung von **IMPORT** für eine bestehende Tabelle mit der Option **REPLACE** oder **REPLACE_CREATE** ist eine der folgenden Berechtigungen erforderlich:
 - **DATAACCESS**
 - Zugriffsrecht **CONTROL** für die Tabelle oder 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:
 - Berechtigung **DBADM**

- Berechtigung CREATETAB für die Datenbank und das Zugriffsrecht USE für den Tabellenbereich sowie eine der folgenden Berechtigungen (bzw. Zugriffsrechte):
 - Berechtigung IMPLICIT_SCHEMA für die Datenbank, wenn der implizite oder explizite Schemaname der Tabelle nicht existiert
 - Zugriffsrecht CREATEIN für das Schema, wenn sich der Schemaname der Tabelle auf ein vorhandenes Schema bezieht
- 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:
 - Berechtigung DBADM
 - Berechtigung CREATETAB für die Datenbank sowie eine der folgenden Berechtigungen:
 - Berechtigung IMPLICIT_SCHEMA für die Datenbank, falls der Name der Tabelle nicht vorhanden ist
 - Zugriffsrecht CREATEIN für das Schema, falls das Schema der Tabelle vorhanden ist
 - Zugriffsrecht CONTROL für jede untergeordnete Tabelle der Hierarchie, wenn 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:
 - DATAACCESS
 - 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

API- und Datenstruktursyntax

```
SQL_API_RC SQL_API_FN
db2Import (
    db2Uint32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2ImportStruct
{
    char *piDataFileName;
    struct sqlu_media_list *piLobPathList;
    struct sqldcol *piDataDescriptor;
    struct sqlchar *piActionString;
    char *piFileType;
    struct sqlchar *piFileTypeMod;
    char *piMsgFileName;
    db2int16 iCallerAction;
    struct db2ImportIn *piImportInfoIn;
    struct db2ImportOut *piImportInfoOut;
    db2int32 *piNullIndicators;
    struct sqllob *piLongActionString;
} db2ImportStruct;

typedef SQL_STRUCTURE db2ImportIn
```

```

{
    db2UInt64                iRowCount;
    db2UInt64                iRestartcount;
    db2UInt64                iSkipcount;
    db2int32                 *piCommitcount;
    db2UInt32                iWarningcount;
    db2UInt16                iNoTimeout;
    db2UInt16                iAccessLevel;
    db2UInt16 *piXmlParse;
    struct db2DMUXmlValidate *piXmlValidate;
} db2ImportIn;

typedef SQL_STRUCTURE db2ImportOut
{
    db2UInt64                oRowsRead;
    db2UInt64                oRowsSkipped;
    db2UInt64                oRowsInserted;
    db2UInt64                oRowsUpdated;
    db2UInt64                oRowsRejected;
    db2UInt64                oRowsCommitted;
} db2ImportOut;

typedef SQL_STRUCTURE db2DMUXmlMapSchema
{
    struct db2Char            iMapFromSchema;
    struct db2Char            iMapToSchema;
} db2DMUXmlMapSchema;

typedef SQL_STRUCTURE db2DMUXmlValidateXds
{
    struct db2Char *piDefaultSchema;
    db2UInt32 iNumIgnoreSchemas;
    struct db2Char *piIgnoreSchemas;
    db2UInt32 iNumMapSchemas;
    struct db2DMUXmlMapSchema *piMapSchemas;
} db2DMUXmlValidateXds;

typedef SQL_STRUCTURE db2DMUXmlValidateSchema
{
    struct db2Char *piSchema;
} db2DMUXmlValidateSchema;

typedef SQL_STRUCTURE db2DMUXmlValidate
{
    db2UInt16 iUsing;
    struct db2DMUXmlValidateXds *piXdsArgs;
    struct db2DMUXmlValidateSchema *piSchemaArgs;
} db2DMUXmlValidate;

SQL_API_RC SQL_API_FN
db2gImport (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2gImportStruct
{
    char                *piDataFileName;
    struct sqlu_media_list *piLobPathList;
    struct sqldcol *piDataDescriptor;
    struct sqlchar *piActionString;
    char *piFileType;
    struct sqlchar *piFileTypeMod;
    char                *piMsgFileName;
    db2int16 iCallerAction;
    struct db2gImportIn *piImportInfoIn;
    struct db2gImportOut *piImportInfoOut;
}

```

```

        db2int32 *piNullIndicators;
        db2Uint16                               iDataFileNameLen;
        db2Uint16 iFileTypeLen;
        db2Uint16                               iMsgFileNameLen;
    struct sqllob *piLongActionString;
} db2gImportStruct;

typedef SQL_STRUCTURE db2gImportIn
{
    db2Uint64                               iRowcount;
    db2Uint64                               iRestartcount;
    db2Uint64                               iSkipcount;
    db2int32                                *piCommitcount;
    db2int32                                iWarningcount;
    db2Uint16                               iNoTimeout;
    db2Uint16                               iAccessLevel;
    db2Uint16 *piXmlParse;
    struct db2DMUXmlValidate *piXmlValidate;
} db2gImportIn;

typedef SQL_STRUCTURE db2gImportOut
{
    db2Uint64                               oRowsRead;
    db2Uint64                               oRowsSkipped;
    db2Uint64                               oRowsInserted;
    db2Uint64                               oRowsUpdated;
    db2Uint64                               oRowsRejected;
    db2Uint64                               oRowsCommitted;
} db2gImportOut;

```

Parameter der API 'db2Import'

versionNumber

Eingabe. Gibt die Version und den Release-Level der Struktur 'pParmStruct' an, die als zweiter Parameter übergeben wird.

pParmStruct

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

pSqlca

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

Parameter der db2ImportStruct-Datenstruktur

piDataFileName

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

piLobPathList

Eingabe. Ein Zeiger auf eine 'sqlu_media_list'-Struktur, wobei das Feld 'media_type' auf SQLU_LOCAL_MEDIA gesetzt ist und die 'sqlu_media_entry'-Struktur Pfade auf dem Client angibt, in denen sich die LOB-Dateien befinden. Dieser Parameter ist beim Importieren eines Kurznamens nicht zulässig.

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 interpretiert 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 Speicherpositionspaar 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

Veraltet. Ersetzt durch piLongActionString.

piLongActionString

Eingabe. Zeiger auf eine *sqllob*-Struktur, die ein 4-Byte-Langfeld enthält, gefolgt von einer Matrix mit Zeichen, die eine Aktion mit Auswirkung auf die Tabelle angibt.

Die Zeichenmatrix 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]}
```

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 Aktua-

lisierung, 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' ist und für 'FileType' 'SQL_IXF' angegeben ist.) Ist die Tabelle noch nicht definiert, wird ein Fehler zurückgegeben.

Anmerkung: Wenn ein Fehler auftritt, nachdem die vorhandenen Daten gelöscht wurden, gehen die Daten verloren. Dieser Parameter ist beim Importieren eines Kurznamens nicht zulässig.

CREATE

Anmerkung: Der Parameter **CREATE** ist veraltet und wird in einem künftigen Release möglicherweise entfernt. Weitere Informationen finden Sie unter „Optionen CREATE und REPLACE_CREATE des Dienstprogramms IMPORT sind veraltet“.

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 mit DB2 exportiert wurde, werden außerdem Indizes erstellt. Wenn die angegebene Tabelle bereits definiert ist, wird ein Fehler zurückgegeben. Diese Option ist nur für das PC/IXF-Dateiformat gültig. Dieser Parameter ist beim Importieren in einen Kurznamen nicht zulässig.

REPLACE_CREATE

Anmerkung: Der Parameter **REPLACE_CREATE** ist veraltet und wird in einem künftigen Release möglicherweise entfernt. Weitere Informationen finden Sie unter „Optionen CREATE und REPLACE_CREATE des Dienstprogramms IMPORT sind veraltet“.

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 außerdem Indizes erstellt. Diese Option ist nur für das PC/IXF-Dateiformat gültig.

Anmerkung: Wenn ein Fehler auftritt, nachdem die vorhandenen Daten gelöscht wurden, gehen die Daten verloren. Dieser Parameter ist beim Importieren eines Kurznamens nicht zulässig.

tname Der Name der Tabelle, typisierten Tabelle, Sicht oder Objektsicht, in die die Daten eingefügt werden sollen. Für REPLACE, INSERT_UPDATE oder INSERT kann ein Aliasname verwendet werden. Dies gilt nicht bei Servern mit früheren Versionen des previ-

ous version of the DB2-Produkts, für die ein qualifizierter oder ein nicht qualifizierter Name angegeben werden sollte. Erfolgt dies in einer Sicht, darf für die Sicht nicht nur Lesezugriff bestehen.

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.

Die Parameter 'tname' und 'tspaltenliste' entsprechen den Listen 'tabellenname' und 'spaltenname' von SQL-INSERT-Anweisungen, 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 'sqldcol'-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.

Dieser Parameter ist beim Importieren eines Kurznamens nicht zulässig.

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 sowie mit vielen anderen Datenbankmanagern und Dateimanagern.

SQL_IXF

PC-Version des IXF-Formats (Integration Exchange Format). Dies ist das bevorzugte Format für den Datenexport aus einer Tabelle. In diesem Dateiformat exportierte Daten können später in dieselbe Tabelle oder in eine andere Datenbankmanagertabelle importiert werden.

SQL_WSF

Arbeitsblattformate für den Datenaustausch mit Lotus Symphony- und 1-2-3-Programmen. Beim Importieren von Daten in einen Kurznamen wird der Dateityp WSF nicht unterstützt.

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. Vgl. den nachstehenden Abschnitt mit Informationen zu den Änderungswerten für Dateityp für das Dienstprogramm 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 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 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. Darum 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 0 ist, muss die angegebene Spalte in der Datendatei entweder 'Y' oder 'N' enthalten. 'Y' gibt an, dass die Tabellenspaltdaten NULL sind, 'N' gibt an, dass die Tabellenspaltdaten nicht NULL sind.

piXmlPathList

Eingabe. Ein Zeiger auf eine 'sqlu_media_list'-Struktur, wobei das Feld 'media_type' auf SQLU_LOCAL_MEDIA gesetzt ist und die 'sqlu_media_entry'-Struktur Pfade auf dem Client angibt, in denen sich die XML-Dateien befinden.

Parameter der db2ImportIn-Datenstruktur

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' 0 ist, wird beim Importieren versucht, alle Zeilen aus der Datei zu verarbeiten.

iRestartcount

Eingabe. Die Anzahl der Datensätze, die übersprungen werden, bevor mit der Einfügung oder Aktualisierung von Datensätzen begonnen wird. Die Funktionsweise ist äquivalent zum Parameter 'iSkipcount'. Die Parameter 'iRestartcount' und 'iSkipcount' schließen sich gegenseitig aus.

iSkipcount

Eingabe. Die Anzahl der Datensätze, die übersprungen werden, bevor mit der Einfügung oder Aktualisierung von Datensätzen begonnen wird. Die Funktionsweise ist äquivalent zum Parameter '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 ein Commit durchgeführt. Ein Nullwert (NULL) gibt den Standardwert der Commitzählung an, der für den Offlineimport Null und für den Onlineimport AUTOMATIC beträgt. Die Commitzählung 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' 0 ist oder diese Option nicht angegeben ist, wird die Importoperation unabhängig von der Anzahl der ausgegebenen Warnungen fortgesetzt.

iNoTimeout

Eingabe. Gibt an, dass es für das Dienstprogramm IMPORT beim Warten auf Sperren kein Zeitlimit gibt. Diese Option überschreibt den Wert des Datenbankkonfigurationsparameters 'locktimeout'. Andere Anwendungen sind nicht betroffen. Gültige Werte sind:

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 sind:

- 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.

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 Commits durch, um eine Sperreneskalation auf eine Tabellensperre zu vermeiden und um zu gewährleisten, dass die aktive Protokolldatei über ausreichend Speicher verfügt. Diese Commits werden auch dann durchgeführt, wenn der Parameter 'piCommitCount' nicht verwendet wurde. Bei jedem Commit verliert IMPORT die Tabellensperre IX und versucht, diese nach jedem Commit erneut anzufordern. Dieser Parameter ist erforderlich,

wenn Daten in einen Kurznamen importiert werden. Für den Parameter 'piCommitCount' muss eine gültige Zahl angegeben werden (AUTOMATIC ist keine gültige Option).

piXmlParse

Eingabe. Typ der Syntexanalyse, die für XML-Dokumente ausgeführt werden soll. Gültige Werte gemäß Kopfdatei 'db2ApiDf' im INCLUDE-Verzeichnis sind:

DB2DMU_XMLPARSE_PRESERVE_WS

Leerzeichen sollen erhalten bleiben.

DB2DMU_XMLPARSE_STRIP_WS

Leerzeichen sollen entfernt werden.

piXmlValidate

Eingabe. Ein Zeiger auf die 'db2DMUXmlValidate'-Struktur. Gibt an, dass für XML-Dokumente eine XML-Schemaprüfung durchgeführt werden soll.

Parameter der db2ImportOut-Datenstruktur

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.

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.

Parameter der db2DMUXmlMapSchema-Datenstruktur

iMapFromSchema

Eingabe. Die SQL-Kennung des für die Zuordnung zu verwendenden XML-Schemas.

iMapToSchema

Eingabe. Die SQL-Kennung des für die Zuordnung zu verwendenden XML-Schemas.

Parameter der db2DMUXmlValidateXds-Datenstruktur

piDefaultSchema

Eingabe. Die SQL-Kennung des XML-Schemas, das für die Prüfung verwendet werden soll, wenn eine XDS kein SCH-Attribut enthält.

iNumIgnoreSchemas

Eingabe. Die Anzahl der XML-Schemata, die während der XML-Schemaprüfung ignoriert werden, wenn auf sie durch ein SCH-Attribut in der XDS verwiesen wird.

piIgnoreSchemas

Eingabe. Die Liste der XML-Schemata, die während der XML-Schemaprüfung ignoriert werden, wenn auf sie durch ein SCH-Attribut in der XDS verwiesen wird.

iNumMapSchemas

Eingabe. Die Anzahl der XML-Schemata, die während der XML-Schemaprüfung zugeordnet werden. Das erste Schema im Schemazuordnungspaar stellt ein Schema dar, auf das durch ein SCH-Attribut in einer XDS verwiesen wird. Das zweite Schema in dem Zuordnungspaar gibt an, welches Schema für die Durchführung der Schemaprüfung verwendet werden soll.

piMapSchemas

Eingabe. Die Liste der XML-Schemapaaare, in der jedes Paar eine Zuordnung eines Schemas zu einem anderen Schema darstellt. Das erste Schema in dem Zuordnungspaar stellt ein Schema dar, auf das durch ein SCH-Attribut in einer XDS verwiesen wird. Das zweite Schema in dem Zuordnungspaar gibt an, welches Schema für die Durchführung der Schemaprüfung verwendet werden soll.

Parameter der db2DMUXmlValidateSchema-Datenstruktur**piSchema**

Eingabe. Die SQL-Kennung des zu verwendenden XML-Schemas.

Parameter der db2DMUXmlValidate-Datenstruktur**iUsing**

Eingabe. Gibt an, was zur Durchführung der XML-Schemaprüfung verwendet werden soll. Gültige Werte gemäß Kopfdatendatei 'db2ApiDf' im INCLUDE-Verzeichnis sind:

- DB2DMU_XMLVAL_XDS

Die Prüfung soll gemäß der XDS durchgeführt werden. Dies entspricht der CLP-Klausel "XMLVALIDATE USING XDS".

- DB2DMU_XMLVAL_SCHEMA

Die Prüfung soll gemäß dem angegebenen Schema durchgeführt werden. Dies entspricht der CLP-Klausel "XMLVALIDATE USING SCHEMA".

- DB2DMU_XMLVAL_SCHEMALOC_HINTS

Die Prüfung soll gemäß den 'schemaLocation'-Hinweisen im XML-Dokument durchgeführt werden. Dies entspricht der Klausel "XMLVALIDATE USING SCHEMALOCATION HINTS".

piXdsArgs

Eingabe. Zeiger auf die 'db2DMUXmlValidateXds'-Struktur, die Argumente entsprechend der CLP-Klausel "XMLVALIDATE USING XDS" darstellt.

Dieser Parameter gilt nur dann, wenn der Parameter 'iUsing' in derselben Struktur auf DB2DMU_XMLVAL_XDS gesetzt ist.

piSchemaArgs

Eingabe. Zeiger auf die 'db2DMUXmlValidateSchema'-Struktur, die Argumente entsprechend der CLP-Klausel "XMLVALIDATE USING SCHEMA" darstellt.

Dieser Parameter gilt nur dann, wenn der Parameter 'iUsing' in derselben Struktur auf DB2DMU_XMLVAL_SCHEMA gesetzt ist.

Struktur der db2glImportStruct-Datenstruktur

iDataFileNameLen

Eingabe. Gibt die Länge des Parameters 'piDataFileName' in Byte an.

iFileTypeLen

Eingabe. Gibt die Länge des Parameters 'piFileType' in Byte an.

iMsgFileNameLen

Eingabe. Gibt die Länge des Parameters 'piMsgFileName' in Byte an.

Hinweise zur Verwendung

Vor dem Start einer Importoperation müssen alle Tabellenoperationen vollständig ausgeführt und alle Sperren freigegeben werden. Hierfür gibt es folgende Möglichkeiten:

- Schließen aller geöffneten Cursor, die mit der Klausel WITH HOLD definiert wurden, und Festschreiben der Datenänderungen durch Ausführen der Anweisung COMMIT.
- Rückgängigmachen der Datenänderungen durch Ausführen der Anweisung ROLLBACK.

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 INSERT-Anweisung fehlschlägt, hat dies eine der beiden nachstehenden 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 ein automatisches Commit 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 ein automatisches Commit für eingefügte Datensätze aus. Falls ein Systemfehler auftritt oder die Anwendung den Datenbankmanager unterbricht, nachdem ein automatisches Commit stattgefunden hat, verbleibt eine Tabelle mit teilweise vorhandenen Daten in der Datenbank. Es kann die gesamte Importoperation wiederholt werden, indem die Option REPLACE oder REPLACE_CREATE verwendet

wird, oder es kann INSERT mit dem Parameter 'iRestartcount' verwendet werden, der auf die Anzahl der erfolgreich importierten Zeilen gesetzt ist.

In der Standardeinstellung werden bei der Option INSERT oder INSERT_UPDATE keine automatischen Commits ausgeführt. Dies gilt jedoch nicht, wenn der Wert des Parameters '*piCommitcount' ungleich 0 ist. Falls das Protokoll keine weiteren Einträge mehr aufnehmen kann, wird ein Rollback ausgeführt.

Immer dann, wenn das Dienstprogramm IMPORT ein Commit 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 einem erfolgreichen Commit geschrieben. Beim erneuten Starten der Importoperation nach einem Fehler geben Sie die Anzahl der zu überspringenden Datensätze an, die Sie anhand des letzten erfolgreichen Commits 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 Objekttafel ausgeführt werden, wenn von dieser Tafel andere Tabellen als sie selbst abhängig sind. Gleiches gilt für eine Objektsicht, wenn andere Tabellen von ihrer Basistafel abhängig sind (einschließlich sie selbst). Um eine solche Tafel oder Sicht zu ersetzen, führen Sie Folgendes aus:

1. Löschen Sie alle Fremdschlüssel, in denen die Tafel als übergeordnete Tafel angegeben ist.
2. Führen Sie das Dienstprogramm IMPORT aus.
3. Ändern Sie die Tafel, 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.

Nichtstandardwerte für 'piDataDescriptor' oder das Angeben einer expliziten Liste mit Tabellenspalten in 'piLongActionString' verlangsamen das Importieren 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.

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 DB2 für AIX-Clients.

Bei Tabellenobjekten auf einer 8-KB-Seite, die den Grenzwert von 1012 Spalten fast erreicht haben, kann der Import von PC/IXF-Datendateien dazu führen, dass DB2 einen Fehler zurückgibt, weil die maximale Größe für eine 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 in DRDA-Server wie DB2 für OS/390, DB2 für VM und VSE und 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. Darum 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.

Daten können nicht in eine Systemtabelle, eine deklarierte temporäre Tabelle, eine erstellte temporäre Tabelle oder eine Übersichtstabelle importiert werden.

Mit dem Dienstprogramm IMPORT können keine Sichten erstellt werden.

Unter Windows-Betriebssystemen:

- 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.

Aspekte für föderierte Datenbanken

Bei der Verwendung der API 'db2Import' und der Parameter INSERT, UPDATE oder INSERT_UPDATE muss sichergestellt werden, dass die Berechtigung CONTROL für den betreffenden Kurznamen vorhanden ist. Weiter muss sichergestellt werden, dass der Kurzname, der bei einer Importoperation verwendet werden soll, bereits vorhanden ist.

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 Zeit "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 wurde(n) "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 wurde(n) verarbeitet. "58" Zeile(n) wurde(n) der Tabelle hinzugefügt. "0" Zeile(n) wurde(n) zurückgewiesen.

Beispiel 2

Das folgende Beispiel zeigt, wie Daten in eine Tabelle mit Identitätsspalten importiert werden können:

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 für diese Zeilen keine Identitätswerte 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 3

Das folgende Beispiel zeigt, wie Daten in eine Tabelle mit Nullanzeigern importiert werden können:

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 WWN6789   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)

```

Anmerkung:

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 bestimmte Spalte kann sich an einer beliebigen Position im Eingabedatensatz befinden. Diese Position muss jedoch angegeben werden, und der Wert Y oder N muss ebenfalls angegeben werden.

Dienstprogramm LOAD

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 XML, 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 Trigger 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. 3 auf Seite 142):

1. LOAD (Laden)

Während der LOAD-Phase werden die Daten in die Tabelle geladen. Bei Bedarf können Indexschlüssel und Tabellenstatistiken erfasst werden. *Sicherungspunkte* oder *Konsistenzpunkte* 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.

2. BUILD

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 die Option STATISTICS USE PROFILE angegeben wurde und das Profil das Erfassen von Indexstatistiken angibt). Die erfassten Statistikdaten sind denen des Befehls **RUNSTATS** ähnlich.

3. DELETE (Löschen)

Während der DELETE-Phase werden die Zeilen, die zu Verstößen gegen eindeutige Schlüssel oder Primärschlüssel geführt haben, aus der Tabelle entfernt. Diese gelöschten Zeilen werden in der LOAD-Ausnahmetabelle gespeichert, sofern diese Tabelle angegeben wurde.

4. INDEX COPY (Indexkopie)

Während der INDEX COPY-Phase werden die Indexdaten aus einem Tabellenbereich für temporäre Systemtabellen in den ursprünglichen Tabellenbereich kopiert. Dieser Vorgang findet nur dann statt, wenn bei der Indexerstellung während einer Ladeoperation, bei der die Option READ ACCESS definiert war, ein Tabellenbereich für temporäre Systemtabellen angegeben wurde.

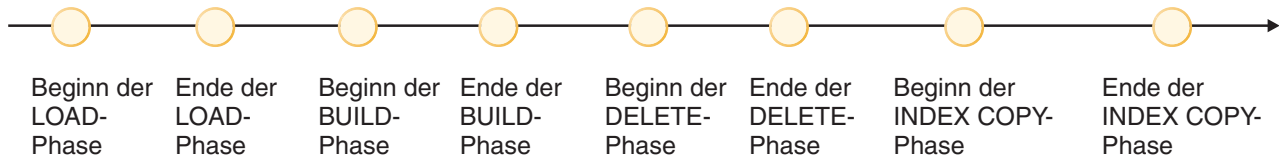


Abbildung 3. Die vier Phasen des LOAD-Prozesses: LOAD, BUILD, DELETE und INDEX COPY

Anmerkung: Nachdem das Dienstprogramm LOAD aufgerufen wurde, können Sie den Verarbeitungsfortschritt der Ladeoperation mithilfe des Befehls **LIST UTILITIES** überwachen.

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.

Lademodi

• Einfügung (INSERT)

In diesem Modus werden die Eingabedaten beim Laden an die Tabelle angefügt, ohne dass an den bereits vorhandenen Daten Änderungen vorgenommen werden.

• Ersetzung (REPLACE)

In diesem Modus werden die bereits vorhandenen Daten beim Laden aus der Tabelle gelöscht und die Tabelle wird mit den Eingabedaten gefüllt.

• Neustart (RESTART)

In diesem Modus wird ein unterbrochener Ladevorgang wieder aufgenommen. In den meisten Fällen wird der Ladevorgang mit der Phase fortgesetzt, in der

zuvor ein Fehler aufgetreten ist. Wenn es sich dabei um die LOAD-Phase gehandelt hat, wird der Ladevorgang beim letzten erfolgreich überprüften Konsistenzpunkt wieder aufgenommen.

- **Beendigung (TERMINATE)**

In diesem Modus wird eine fehlgeschlagene Ladeoperation mit einem Rollback zurückgesetzt.

Sie können die folgenden Optionen angeben:

- Die zu ladenden Daten befinden sich auf dem Client - sofern das Dienstprogramm LOAD von einem Client mit Fernverbindung aufgerufen wird. Beachten Sie, dass XML- und LOB-Daten immer aus dem Server gelesen werden, auch wenn die Option CLIENT angegeben wurde.
- Die Methode, die zum Laden der Daten verwendet werden soll: Spaltenposition, Spaltenname oder relative Spaltenposition.
- Die Häufigkeit, mit der das Dienstprogramm Konsistenzpunkte definieren soll.
- 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.
- Die Angabe, ob die Ladeoperation entweder warten soll, bis andere Dienstprogramme oder Anwendungen die Verwendung der Tabelle beendet haben, oder vor der Fortsetzung das Beenden der anderen Anwendungen erzwingen soll.
- Einen alternativen Tabellenbereich für temporäre Tabellen, in dem der Index erstellt werden soll.
- Die Pfade und die Namen der Eingabedateien, in denen LOBs gespeichert sind.

Anmerkung: Die LOB-Option COMPACT wird vom Dienstprogramm LOAD nicht unterstützt.

- Einen Nachrichtendateinamen. Bei Ladeoperationen können Sie angeben, dass Nachrichtendateien erstellt werden sollen, in denen Fehlernachrichten, Warnungen und informative Nachrichten zu diesen Operationen gespeichert werden. Geben Sie den Namen dieser Dateien mit dem Parameter MESSAGES an.

Anmerkung:

1. Sie können den Inhalt einer Nachrichtendatei erst anzeigen, wenn die Operation beendet ist. Wenn Sie die Ladenachrichten anzeigen möchten, während eine Ladeoperation ausgeführt wird, können Sie den Befehl **LOAD QUERY** verwenden.
 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.
 - Angabe, ob das Dienstprogramm die Menge des freien Speicherbereichs nach dem Laden einer Tabelle ändern soll.
 - Angabe, ob während des LOAD-Prozesses Statistikdaten zusammengestellt werden. Diese Option wird nur unterstützt, wenn die Ladeoperation im Ersetzungsmodus (REPLACE) ausgeführt wird. 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 Fehlernachricht zurückgegeben.

Wenn Daten an eine Tabelle angefügt werden, werden keine Statistikdaten erfasst. Rufen Sie zum Erfassen von aktuellen Statistikdaten zu einer angefügten Tabelle nach Fertigstellung 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 Fertigstellung des Ladeprozesses das Dienstprogramm **RUNSTATS** auf.

- Angabe, ob eine Kopie der vorgenommenen Änderungen erhalten bleiben soll. Diese wird benötigt, um eine aktualisierende Recovery der Datenbank zu ermöglichen. Diese Option wird nicht unterstützt, wenn für die Datenbank die aktualisierende Recovery inaktiviert ist, d.h. wenn die Datenbankkonfigurationsparameter *logarchmeth1* und *logarchmeth2* auf OFF gesetzt sind. Wenn keine Kopie erstellt wird und die aktualisierende Recovery aktiviert ist, verbleibt der Tabellenbereich nach Fertigstellung der Ladeoperation im Status "Backup 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 Datenbankrecovery nach einem Fehler zulässt, haben Sie folgende Möglichkeiten:

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

Wird der Datenbankkonfigurationsparameter *logindexbuild* gesetzt und wird die Ladeoperation mit der Wiederherstellbarkeitsoption COPY YES und der Indexierungsoption INCREMENTAL aufgerufen, dann werden beim Laden alle Indexänderungen protokolliert. Der Vorteil bei der Verwendung dieser Optionen liegt darin, dass bei der aktualisierenden Recovery 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. Wenn Sie eine Tabelle laden, die bereits Daten enthält, und die Datenbank nicht wiederherstellbar ist, müssen Sie sicherstellen, dass Sie über eine Backup-Kopie 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, wird die Operationsfolge schneller durchlaufen, wenn Sie jede Ladeoperation als nicht wiederherstellbar definieren und am Ende der Ladefolge ein Backup 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 Recovery wiederherzustellen. Das Dienstprogramm zur aktualisierenden Recovery 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 Fertigstellung der aktualisierenden Recovery kann eine solche Tabelle nur gelöscht werden (siehe Abb. 4 auf Seite 145). Mit dieser Option werden Tabellenbereiche nach der Ladeoperation nicht in den Status "Backup anstehend" gesetzt, und während der Ladeoperation muss keine Kopie der geladenen Daten erstellt werden.

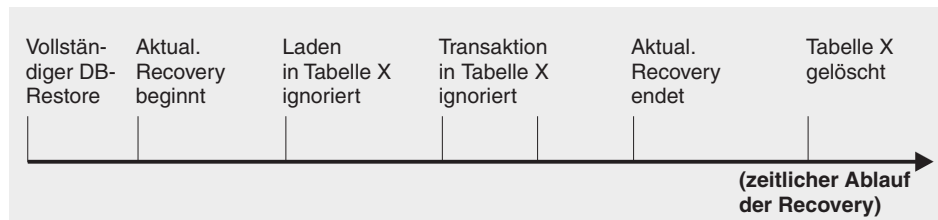


Abbildung 4. Nicht wiederherstellbare Verarbeitung bei einer aktualisierenden Recovery

- Den vollständig qualifizierten Pfad, der beim Erstellen von temporären Dateien während einer Ladeoperation verwendet werden soll. Der Name wird mit dem Parameter `TEMPFILES PATH` des Befehls **LOAD** angegeben. Der Standardwert ist der Datenbankpfad. Der Pfad befindet sich auf der Servermaschine, und der Zugriff auf diesen Pfad erfolgt ausschließlich durch die DB2-Instanz. Daher muss jede Angabe eines Pfadnamens in diesem Parameter die Verzeichnisstruktur des Servers und nicht des Clients wiedergeben. Außerdem benötigt der DB2-Instanzeigner Lese- und Schreibberechtigung für den Pfad.

Für das Dienstprogramm **LOAD** erforderliche Zugriffsrechte und Berechtigungen

Zugriffsrechte ermöglichen es Benutzern, Datenbankressourcen zu erstellen oder auf diese zuzugreifen. Berechtigungsstufen stellen eine Methode dar, um Berechtigungen sowie übergeordnete Pflege- und Dienstprogrammoperationen des Datenbankmanagers zusammenzufassen. Sie dienen zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte. Benutzer können nur auf die Objekte zugreifen, für die sie zugriffsberechtigt sind, d. h., für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigung verfügen.

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

- Berechtigung `DATAACCESS`
- Berechtigung `LOAD` oder `DBADM` 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.
 - Zugriffsrecht `SELECT` für `SYSCAT.TABLES` in einigen Fällen, in denen `LOAD` die Katalogtabellen abfragt.

Da der Instanzeigner auch Eigner aller `LOAD`-Prozesse (und im Allgemeinen auch aller DB2-Serverprozesse) ist und alle diese Prozesse die Kennung des Instanzeigners verwenden, um auf benötigte Dateien zuzugreifen, muss der Instanzeigner

über Lesezugriff auf Eingabedatendateien verfügen. Diese Eingabedatendateien müssen für den Instanzeigner unabhängig davon, wer den Befehl aufruft, lesbar sein.

Bei Angabe der Option REPLACE muss die Berechtigungs-ID der Sitzung über die Berechtigung zum Löschen der Tabelle verfügen.

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

Anmerkung:

- Um Daten in eine Tabelle laden zu können, die geschützte Spalten enthält, muss die Berechtigungs-ID der Sitzung über LBAC-Berechtigungs-nachweise verfügen, die Schreibzugriff auf alle geschützten Spalten in der Tabelle zulassen.
- Um Daten in eine Tabelle laden zu können, die geschützte Zeilen enthält, muss der Berechtigungs-ID der Sitzung mit Grant ein Sicherheitskennsatz für Schreibzugriff erteilt werden, der Teil der Sicherheitsrichtlinie zum Schutz der Tabelle ist.

Berechtigung LOAD

Benutzer, die über die Berechtigung LOAD auf Datenbankebene sowie über das Zugriffsrecht INSERT für eine Tabelle verfügen, können den Befehl **LOAD** verwenden, um Daten in eine Tabelle zu laden.

Anmerkung: Benutzer mit der Berechtigung DATAACCESS verfügen über die volle Berechtigung für die Verwendung des Befehls LOAD.

Benutzer, die über die Berechtigung LOAD auf Datenbankebene sowie über das Zugriffsrecht INSERT auf eine Tabelle verfügen, können den Befehl **LOAD RESTART** oder **LOAD TERMINATE** verwenden, wenn bei der vorangegangenen Ladeoperation Daten eingefügt (LOAD INSERT) wurden.

Benutzer, die über die Berechtigung LOAD auf Datenbankebene sowie über die Zugriffsrechte INSERT und DELETE für eine Tabelle verfügen, können den Befehl **LOAD REPLACE** verwenden.

Wenn bei der vorangegangenen Ladeoperation Daten ersetzt (LOAD REPLACE) wurden, muss diesem Benutzer auch das Zugriffsrecht DELETE erteilt werden, damit dieser den Befehl **LOAD RESTART** oder **LOAD TERMINATE** verwenden kann.

Wenn die Ausnahmetabellen im Rahmen der LOAD-Operation verwendet werden, muss der Benutzer über das Zugriffsrechte INSERT für die Ausnahmetabellen verfügen.

Der Benutzer mit dieser Berechtigung kann die Befehle **QUIESCE TABLESPACES FOR TABLE**, **RUNSTATS** und **LIST TABLESPACES** ausführen.

Laden von Daten

Mit dem Dienstprogramm LOAD können große Datenmengen effizient in neu erstellte Tabellen oder in Tabellen, die bereits Daten enthalten, versetzt werden.

Vorbereitende Schritte

Bevor Sie das Dienstprogramm LOAD ausführen, müssen Sie mit der Datenbank, in die die Daten geladen werden sollen, verbunden bzw. in der Lage sein, implizit eine Verbindung zur Datenbank herzustellen.

Da das Dienstprogramm die Anweisung COMMIT absetzt, beenden Sie vor dem Aufrufen des Dienstprogramms LOAD alle Transaktionen und geben Sie alle Sperren frei, indem Sie entweder eine Anweisung COMMIT oder eine Anweisung ROLLBACK absetzen.

Daten werden in der Reihenfolge geladen, in der sie in der Eingabedatei enthalten sind. Ausgenommen sind MDC-Tabellen (Multidimensional Clustering Table - mehrdimensionale Clusteringtabelle), partitionierte Tabellen und Ladeoperationen mit dem Änderungswert **anyorder** für den Dateityp. Falls eine bestimmte Reihenfolge gewünscht wird, müssen die Daten 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 (Clustern) zusammengefasst. Beim Laden von Daten in partitionierte Tabellen ist vor der Ladeoperation keine Sortierung erforderlich, und die Daten werden gemäß der Tabellendefinition partitioniert.

Informationen zu diesem Vorgang

Im Folgenden finden Sie einige der für das Dienstprogramm LOAD geltenden 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 und erstellte temporäre Tabellen wird nicht unterstützt.
- XML-Daten können nur serverseitig gelesen werden; wenn die XML-Dateien aus dem Client gelesen werden sollen, verwenden Sie das Dienstprogramm IMPORT.
- In einem Tabellenbereich, der sich im Status "Backup anstehend" befindet, können keine Tabellen erstellt oder gelöscht werden.
- Sie können keine Daten in eine Datenbank laden, auf die über DB2 Connect oder eine Version des Servers vor DB2 Version 2 zugegriffen wird. Optionen, die nur mit dem aktuellen Release 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 Trigger aktiviert. Geschäftsregeln, die Triggern zugeordnet sind, werden vom Dienstprogramm LOAD nicht umgesetzt.
- Das Laden verschlüsselter Daten wird nicht unterstützt.

Im Folgenden finden Sie einige der für das Dienstprogramm LOAD geltenden Einschränkungen beim Laden in eine partitionierte Tabelle:

- Konsistenzpunkte werden nicht unterstützt, wenn die Anzahl der Partitionierungsagenten größer als 1 ist.

- Das Laden von Daten in eine Untergruppe von Datenpartitionen, während die verbleibenden Datenpartitionen vollständig online bleiben, wird nicht unterstützt.
- Die von einer Ladeoperation oder einer Operation im Status "Festlegen der Integrität anstehend" verwendete Ausnahmetabelle kann nicht partitioniert werden.
- Ein eindeutiger Index kann nicht erneut erstellt werden, wenn das Dienstprogramm LOAD im Einfügemodus (INSERT) oder Neustartmodus (RESTART) ausgeführt wird und freigegebene Objekte von der Zieltabelle der Ladeoperation abhängen.

Vorgehensweise

Gehen Sie folgendermaßen vor, um das Dienstprogramm LOAD auszuführen:

- Geben Sie den Befehl **LOAD** im Befehlszeilenprozessor (CLP) an.
- Rufen Sie die Anwendungsprogrammierschnittstelle (API) db2Load über eine Clientanwendung auf.
- Verwenden Sie IBM Data Studio.

Zwar kann auch der Assistent 'Laden' in der Steuerzentrale verwendet werden, es empfiehlt sich jedoch die Verwendung von IBM Data Studio, da die Toolfunktionen der Steuerzentrale nicht weiter unterstützt werden.

Beispiel

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) gespeichert.
- Bei der Ladeoperation sind 4000 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 derselben Datenbankpartition wie die Datenbank zulässig. Es empfiehlt sich, vollständig qualifizierte Pfadnamen zu verwenden.

Nächste Schritte

Nachdem das Dienstprogramm LOAD aufgerufen wurde, können Sie den Bearbeitungsfortschritt der Ladeoperation mithilfe des Befehls **LIST UTILITIES** überwachen. Wird eine Ladeoperation im Modus **INSERT**, **REPLACE** oder **RESTART** ausge-

führt, steht die Unterstützung für eine ausführliche Überwachung des Verarbeitungsfortschritts zur Verfügung. Setzen Sie den Befehl **LIST UTILITIES** mit der Option **SHOW DETAILS** ab, um ausführliche Informationen zur aktuellen LOAD-Phase anzuzeigen. Für Ladeoperationen, die im Modus **TERMINATE** ausgeführt werden, 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.

Eine Ladeoperation erhält eindeutige Integritätsbedingungen, Bereichsvorgaben für partitionierte Tabellen, generierte Spalten sowie LBAC-Sicherheitsregeln. Bei allen anderen Integritätsbedingungen wird die Tabelle zu Beginn der Ladeoperation in den Status "Festlegen der Integrität anstehend" versetzt. Nachdem die Ladeoperation vollständig ausgeführt wurde, muss der Status "Festlegen der Integrität anstehend" für die Tabelle mithilfe der Anweisung **SET INTEGRITY** aufgehoben werden.

Laden von XML-Daten

Mit dem Dienstprogramm LOAD können große XML-Datenmengen effizient in Tabellen versetzt werden.

Beim Laden von Daten in eine XML-Tabellenspalte können Sie die Option **XML FROM** verwenden, um die Pfade der XML-Eingabedatei oder der XML-Eingabedateien anzugeben. Um beispielsweise Daten aus der Datei `/home/user/xmlpath/xmlfile1.xml` zu laden, könnten Sie den folgenden Befehl verwenden:

```
LOAD FROM data1.del OF DEL XML FROM /home/user/xmlpath INSERT INTO USER.T1
```

Die ASCII-Eingabedatei mit Begrenzern `data1.del` enthält eine XML-Datenkennung (XDS), die die Speicherposition der zu ladenden XML-Daten angibt. Die folgende XDS beispielsweise beschreibt ein XML-Dokument an der relativen Adresse (Offset) bei 123 Byte in der Datei `xmldata.ext` mit einer Länge von 456 Byte:

```
<XDS FIL='xmldata.ext' OFF='123' LEN='456' />
```

Das Laden von XML-Daten mithilfe eines deklarierten Cursors wird unterstützt. Im folgenden Beispiel wird ein Cursor deklariert und mit dem Befehl **LOAD** verwendet, um Daten aus der Tabelle **CUSTOMERS** in der Tabelle **LEVEL1_CUSTOMERS** hinzuzufügen:

```
DECLARE cursor_income_level1 CURSOR FOR
  SELECT * FROM customers
  WHERE XMLEXISTS('$DOC/customer[income_level=1]');

LOAD FROM cursor_income_level1 OF CURSOR INSERT INTO level1_customers;
```

Der Dateitypänderungswert **ANYORDER** des Befehls **LOAD** wird beim Laden von XML-Daten in eine XML-Spalte unterstützt.

Während des Ladens werden für Spalten des Typs XML keine Verteilungsstatistiken erfasst.

Laden von XML-Daten in einer Umgebung mit partitionierten Datenbanken

Bei Tabellen, die über Datenbankpartitionen verteilt sind, können Sie XML-Daten aus XML-Datendateien parallel in die Tabellen laden. Beim Laden von XML-Daten in Tabellen müssen die XML-Datendateien über Lesezugriff auf alle Datenbankpartitionen verfügen, in denen das Laden stattfindet.

Eingefügte Dokumente anhand von Schemata prüfen

Die Option XMLVALIDATE ermöglicht die Gültigkeitsprüfung von XML-Dokumenten mithilfe von XML-Schemata während des Ladens. Im folgenden Beispiel wird die Gültigkeit eingehender XML-Dokumente auf der Basis des Schemas überprüft, das durch die XDS in der ASCII-Eingabedatei mit Begrenzern data2.de1 identifiziert wird:

```
LOAD FROM data2.de1 OF DEL XML FROM /home/user/xmlpath XMLVALIDATE
USING XDS INSERT INTO USER.T2
```

In diesem Fall enthält die XDS ein Attribut vom Typ SCH mit der vollständig qualifizierten SQL-Kennung des für die Gültigkeitsprüfung zu verwendenden XML-Schemas "S1.SCHEMA_A":

```
<XDS FIL='xmldata.ext' OFF='123' LEN='456' SCH='S1.SCHEMA_A' />
```

Optionen für das Parsing angeben

Mit der Option XMLPARSE können Sie angeben, ob Leerzeichen in den geladenen XML-Dokumenten beim Parsing (d. h. bei der syntaktischen Analyse) beibehalten oder gelöscht werden sollen. Im folgenden Beispiel wird die Gültigkeit aller geladenen XML-Dokumente auf der Basis des Schemas mit der SQL-Kennung "S2.SCHEMA_A" überprüft, und für diese Dokumente werden die Leerzeichen beim Parsing beibehalten:

```
LOAD FROM data2.de1 OF DEL XML FROM /home/user/xmlpath XMLPARSE PRESERVE
WHITESPACE XMLVALIDATE USING SCHEMA S2.SCHEMA_A INSERT INTO USER.T1
```

Ladeaspekte für partitionierte Tabellen

Alle vorhandenen Ladefunktionen werden unterstützt, wenn die Zieltabelle partitioniert ist. Hierbei gelten jedoch die folgenden allgemeinen Einschränkungen:

- Konsistenzpunkte werden nicht unterstützt, wenn die Anzahl der Partitionierungsagenten größer als 1 ist.
- Das Laden von Daten in eine Untergruppe von Datenpartitionen, während die verbleibenden Datenpartitionen vollständig online bleiben, wird nicht unterstützt.
- Die von einer Ladeoperation verwendete Ausnahmetabelle kann nicht partitioniert sein.
- Es kann keine Ausnahmetabelle angegeben werden, wenn die Zieltabelle eine XML-Spalte enthält.
- Ein eindeutiger Index kann nicht erneut erstellt werden, wenn das Dienstprogramm LOAD im Einfügemodus (INSERT) oder Neustartmodus (RESTART) ausgeführt wird und freigegebene Objekte von der Zieltabelle der Ladeoperation abhängen.
- Ähnlich wie beim Laden von MDC-Tabellen bleibt die genaue Reihenfolge der Eingabedatensätze beim Laden von partitionierten Tabellen nicht erhalten. Sortierungen werden lediglich innerhalb der Zelle oder Datenpartition beibehalten.
- Ladeoperationen, die mehrere Formatierungsprogramme auf jeder Datenbankpartition verwenden, erhalten lediglich eine ungefähre Reihenfolge der Eingabedatensätze. Beim Ausführen eines einzelnen Formatierungsprogramms auf jeder Datenbankpartition werden die Eingabedatensätze nach Zellen- oder Tabellenpartitionierungsschlüssel gruppiert. Um auf jeder Datenbankpartition ein einziges Formatierungsprogramm auszuführen, muss für CPU_PARALLELISM explizit der Wert 1 angefordert werden.

Allgemeines Verhalten des Dienstprogramms LOAD

Das Dienstprogramm LOAD fügt Datensätze in die richtige Datenpartition ein. Es ist nicht erforderlich, ein externes Dienstprogramm (wie beispielsweise einen Verteilerprozess) zu verwenden, um die Eingabedaten vor dem Laden zu partitionieren.

Das Dienstprogramm LOAD greift nicht auf freigegebene oder zugeordnete Datenpartitionen zu. Daten werden lediglich in sichtbare Datenpartitionen eingefügt. Sichtbare Datenpartitionen sind weder zugeordnet noch freigegeben. Darüber hinaus werden freigegebene oder zugeordnete Datenpartitionen von LOAD REPLACE-Operationen nicht abgeschnitten. Da das Dienstprogramm LOAD Sperren für die Systemkatalogtabellen anfordert, wartet das Dienstprogramm LOAD auf alle ALTER TABLE-Transaktionen, für die noch kein Commit durchgeführt wurde. Solche Transaktionen fordern eine exklusive Sperre für die relevanten Zeilen in den Katalogtabellen an, und die exklusive Sperre muss beendet werden, bevor die Ladeoperation fortgesetzt werden kann. Dies bedeutet, dass nicht festgeschriebene Transaktionen ALTER TABLE ... ADD PARTITION, ALTER TABLE ... ATTACH PARTITION bzw. ALTER TABLE ... DETACH PARTITION nicht zulässig sind, während die Ladeoperation ausgeführt wird. Alle Eingabequellsätze, die für eine zugeordnete (ATTACHED) oder freigegebene (DETACHED) Datenpartition bestimmt sind, werden zurückgewiesen und können aus der Ausnahmetabelle abgerufen werden, sofern eine angegeben wurde. Eine Informationsnachricht wird in die Nachrichtendatei geschrieben, um anzugeben, dass einige Datenpartitionen der Zieltabelle im Status zugeordnet (ATTACHED) oder freigegeben (DETACHED) waren. Sperren für die relevanten Zeilen der Katalogtabelle, die der Zieltabelle entsprechen, verhindern, dass Benutzer die Partitionierung der Zieltabelle durch Absetzen einer Operation ALTER TABLE ... ADD PARTITION, ALTER TABLE ... ATTACH PARTITION bzw. ALTER TABLE ... DETACH PARTITION während der Ausführung des Dienstprogramms LOAD ändern.

Verarbeitung ungültiger Zeilen

Wenn das Dienstprogramm LOAD auf einen Datensatz trifft, der zu keiner der sichtbaren Datenpartitionen gehört, wird der betreffende Datensatz zurückgewiesen, und das Dienstprogramm LOAD setzt die Verarbeitung fort. Die Anzahl der Datensätze, die aufgrund einer Verletzung der Bereichsvorgabe zurückgewiesen wurden, wird nicht explizit angegeben, ist aber in der Gesamtanzahl der zurückgewiesenen Datensätze enthalten. Durch das Zurückweisen eines Datensatzes aufgrund einer Verletzung der Bereichsvorgabe wird die Anzahl der Warnungen für Zeilen nicht erhöht. Es wird eine einzige Nachricht (SQL0327N) in die Nachrichtendatei des Dienstprogramms LOAD geschrieben, die angibt, dass Bereichsverletzungen festgestellt wurden. Es wird jedoch nicht für jeden Datensatz eine eigene Nachricht protokolliert. Zusätzlich zu allen Spalten der Zieltabelle enthält die Ausnahmetabelle auch Spalten, die den Typ des Verstoßes beschreibt, der für eine bestimmte Zeile aufgetreten ist. Zeilen, die ungültige Daten enthalten (einschließlich Daten, die nicht partitioniert werden können), werden in die Speicherauszugsdatei geschrieben.

Da Einfügungen (INSERT-Operationen) in die Ausnahmetabelle ressourcenintensiv sind, können Sie steuern, welche Verstöße gegen Integritätsbedingungen in die Ausnahmetabelle eingefügt werden sollen. Das Standardverhalten des Dienstprogramms LOAD ist beispielsweise, Zeilen in die Ausnahmetabelle einzufügen, die aufgrund einer Verletzung der Bereichsvorgabe oder der eindeutigen Integritätsbedingung zurückgewiesen wurden, ansonsten jedoch gültig sind. Sie können dieses Verhalten inaktivieren, indem Sie jeweils NORANGEEXC oder NOUNIQUEEXC mit der Ausnahmebedingung FOR EXCEPTION angeben. Wenn Sie angeben, dass diese Verletzungen der Bereichsvorgabe nicht in die Ausnahmetabelle eingefügt

werden sollen, oder wenn Sie keine Ausnahmetabelle angeben, gehen alle Informationen zu Zeilen, die eine Bereichsvorgabe oder eine eindeutige Integritätsbedingung verletzen, verloren.

Protokolldatei

Bei partitionierten Zieltabellen enthält der entsprechende Eintrag in der Protokolldatei keine Liste der Tabellenbereiche, die die Zieltabelle umfasst. Eine andere Kennung für die Granularität von Operationen ('R' anstelle von 'T') gibt an, dass eine Ladeoperation für eine partitionierte Tabelle ausgeführt wurde.

Beenden einer Ladeoperation

Beim Beenden einer LOAD REPLACE-Operation werden alle sichtbaren Datenpartitionen vollständig abgeschnitten; beim Beenden einer LOAD INSERT-Operation werden alle sichtbaren Datenpartitionen auf ihre jeweilige Länge vor der Ladeoperation abgeschnitten. Beim Beenden einer ALLOW READ ACCESS-Ladeoperation, die in der LOAD COPY-Phase fehlschlug, werden die entsprechenden Indizes ungültig gemacht. Ein Index wird ebenfalls ungültig gemacht, wenn eine Ladeoperation mit der Option ALLOW NO ACCESS beendet wird, die den Index geändert hat. (Der Index wird ungültig gemacht, weil der Indexierungsmodus REBUILD (erneut erstellen) ist oder weil ein Schlüssel während der inkrementellen Verwaltung eingefügt wurde und den Index in einem inkonsistenten Status belassen hat). Das Laden von Daten in mehrere Ziele hat keinen Einfluss auf Recoveryoperationen, mit der Ausnahme, dass die Ladeoperation von einem Konsistenzpunkt, der während der LOAD-Phase erstellt wurde, nicht erneut gestartet werden kann. In diesem Fall wird die LOAD-Option SAVECOUNT ignoriert, wenn die Zieltabelle partitioniert ist. Dieses Verhalten entspricht dem Verhalten beim Laden von Daten in einer MDC-Zieltabelle.

Generierte Spalten

Befindet sich eine generierte Spalte in einem der Partitionierungs-, Dimensions- oder Verteilungsschlüssel, wird der Änderungswert generatedoverride für den Dateityp ignoriert, und das Dienstprogramm LOAD generiert Werte, als wäre der Änderungswert generatedignore für den Dateityp angegeben. Das Laden eines falschen Wertes einer generierten Spalte kann in diesem Fall dazu führen, dass der Datensatz in die falsche physische Position (wie beispielsweise in die falsche Datenpartition, den falschen MDC-Block oder die falsche Datenbankpartition) gestellt wird. Beispiel: Befindet sich ein Datensatz in einer falschen Datenpartition, muss die Operation SET INTEGRITY zum Festlegen der Integrität den Satz in eine andere physische Position versetzen, was im Verlauf von SET INTEGRITY-Operationen, die online ausgeführt werden, nicht bewerkstelligt werden kann.

Datenverfügbarkeit

Der aktuelle Algorithmus für ALLOW READ ACCESS-Ladeoperationen erstreckt sich auf partitionierte Tabellen. Eine ALLOW READ ACCESS-Ladeoperation ermöglicht den gleichzeitigen Lesezugriff auf die gesamte Tabelle. Dies schließt sowohl Ladedatenpartitionen als auch Nicht-Ladedatenpartitionen ein.

Datenpartitionsstatus

Nach einer erfolgreichen Ladeoperation können sichtbare Datenpartitionen entweder in den Tabellenstatus Set Integrity Pending oder Read Access Only oder beide wechseln. Dies hängt von bestimmten Bedingungen ab. So können Datenpartitionen in diese Status versetzt werden, wenn für die Tabelle Integritätsbedingungen bestehen, die von der Ladeoperation nicht verwaltet werden können. Dabei kann es sich z. B. um Integritätsbedingungen handeln, die Prüfungen auf Integritätsbe-

dingungen und freigegebene MQTs (Materialized Query Table) beinhalten. Schlägt eine Ladeoperation fehl, verbleiben alle sichtbaren Datenpartitionen im Tabellenstatus Load Pending ('Laden anstehend').

Fehlerisolation

Eine Fehlerisolation auf Datenpartitionesebene wird nicht unterstützt. Das Isolieren von Fehlern bedeutet, Ladeoperationen für Datenpartitionen ohne Fehler fortzusetzen und Ladeoperationen für Datenpartitionen mit Fehlern zu stoppen. Fehler können in verschiedenen Datenbankpartitionen übergreifend isoliert werden. Das Dienstprogramm LOAD kann jedoch keine Transaktionen in einer Untergruppe von sichtbaren Datenpartitionen festschreiben und in den verbleibenden sichtbaren Datenpartitionen rückgängig machen.

Sonstige Aspekte

- Eine inkrementelle Indexierung wird nicht unterstützt, wenn einer der Indizes als ungültig markiert ist. Ein Index gilt als ungültig, wenn er erneut erstellt werden muss oder wenn freigegebene abhängige Objekte mit der Anweisung SET INTEGRITY überprüft werden müssen.
- Das Laden von Daten in Tabellen, deren Partitionierung anhand einer beliebigen Kombination aus Algorithmen für das Partitionieren nach Bereich, das Verteilen nach Hash oder das Organisieren nach Dimension erfolgte, wird ebenfalls unterstützt.
- Die Größe von Protokollsätzen, die eine Liste der von der Ladeoperation betroffenen Objekt- und Tabellenbereichs-IDs umfassen (Protokollsätze LOAD START und COMMIT (PENDING LIST)) kann stark anwachsen und so den für andere Anwendungen zur Verfügung stehenden aktiven Protokollspeicher reduzieren.
- Wenn eine Tabelle sowohl partitioniert als auch verteilt ist, kann es sein, dass sich eine Ladeoperation für partitionierte Datenbanken nicht auf alle Datenbankpartitionen auswirkt. Nur die Objekte in den Ausgabedatenbankpartitionen werden geändert.
- Während einer Ladeoperation nimmt die Speicherbelegung für partitionierte Tabellen mit der Anzahl der Tabellen zu. Bitte beachten Sie, dass der Gesamtanstieg nicht linear ist, da nur ein geringer Prozentsatz des Gesamtspeicherbedarfs proportional zur Anzahl der Datenpartitionen ist.

Aspekte des Ladens von LBAC-geschützten Daten

Die Voraussetzung für eine erfolgreichen Ladeoperation in eine Tabelle mit geschützten Zeilen sind LBAC-Berechtigungsnachweise (Label-Based Access Control). Darüber hinaus müssen Sie über einen gültigen Sicherheitskennsatz (oder einen Sicherheitskennsatz, der in einen gültigen Kennsatz konvertiert werden kann) für die Sicherheitsrichtlinie verfügen, die der Zieltabelle aktuell zugeordnet ist.

Wenn Sie nicht über gültige LBAC-Berechtigungsnachweise verfügen, schlägt das Laden fehl, und es wird ein Fehler (SQLSTATE 42512) zurückgegeben. Enthalten die Eingabedaten keinen Sicherheitskennsatz oder liegt der Sicherheitskennsatz nicht im zugehörigen internen Binärformat vor, können Sie die Ladeoperation mithilfe verschiedener Änderungswerte für den Dateityp fortsetzen.

Wenn Sie Daten in eine Tabelle mit geschützten Zeilen laden, enthält die Zieltabelle eine einzige Spalte mit dem Datentyp DB2SECURITYLABEL. Wenn die Eingabedatenzeile keinen Wert für diese Spalte enthält, wird die Zeile zurückgewiesen, es sei denn, der Ladebefehl wurde mit dem Änderungswert `usedefaults` für den Dateityp angegeben. Wurde `usedefaults` im Ladebefehl angegeben, wird der Sicherheitskennsatz, über den Sie für den Schreibzugriff verfügen, aus der Sicherheits-

richtlinie, die die Tabelle schützt, verwendet. Wenn Sie keinen Sicherheitskennsatz für den Schreibzugriff haben, wird die Zeile zurückgewiesen und mit der Verarbeitung der nächsten Zeile fortgefahren.

Wenn Sie Daten in eine Tabelle mit geschützten Zeilen laden und die Eingabedaten einen Wert für die Spalte mit dem Datentyp DB2SECURITYLABEL enthalten, gelten dieselben Regeln wie beim Einfügen von Daten in die betreffende Tabelle. Wenn der Sicherheitskennsatz, der die zu ladende Zeile schützt, also der Kennsatz in der betreffenden Zeile der Datendatei, ein Kennsatz ist, der Ihnen den Schreibzugriff ermöglicht, wird der betreffende Sicherheitskennsatz zum Schutz der Zeile verwendet. Dies bedeutet, dass der Sicherheitskennsatz in die Spalte mit dem Datentyp DB2SECURITYLABEL geschrieben wird. Wenn Sie nicht in eine Zeile schreiben können, die durch diesen Sicherheitskennsatz geschützt wird, hat dies, je nachdem wie die Sicherheitsrichtlinie zum Schutz der Quellentabelle erstellt wurde, unterschiedliche Konsequenzen:

- Wenn die Anweisung CREATE SECURITY POLICY, mit der die Richtlinie erstellt wurde, die Option RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL enthielt, wird die Zeile zurückgewiesen.
- Enthielt die Anweisung CREATE SECURITY POLICY diese Option nicht oder stattdessen die Option OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL, wird der Sicherheitskennsatz in der Datendatei für die betreffende Zeile ignoriert und stattdessen zum Schutz der Zeile der Sicherheitskennsatz verwendet, den Sie für den Schreibzugriff besitzen. In diesem Fall wird weder ein Fehler noch eine Warnung ausgegeben. Wenn Sie keinen Sicherheitskennsatz für den Schreibzugriff haben, wird die Zeile zurückgewiesen und mit der Verarbeitung der nächsten Zeile fortgefahren.

Hinweise zu Begrenzern

Wenn Sie Daten in eine Spalte des Typs DB2SECURITYLABEL laden, wird standardmäßig davon ausgegangen, dass es sich bei dem Wert in der Datendatei um die tatsächlichen Byte handelt, die die interne Darstellung dieses Sicherheitskennsatzes bilden. Einige Rohdaten enthalten jedoch möglicherweise Zeichen für Zeilenvorschub, die vom Befehl **LOAD** fälschlicherweise als Begrenzer für die Zeile interpretiert werden. Wenn dieses Problem vorliegt, können Sie mit dem Änderungswert `delprioritychar` für den Dateityp sicherstellen, dass der Zeichenbegrenzer Vorrang vor dem Zeilenbegrenzer hat. Wenn Sie `delprioritychar` verwenden, werden Zeilen- und Spaltenbegrenzer, die innerhalb der zwischen den Zeichenbegrenzern enthaltenen Daten vorliegen, nicht als Begrenzer erkannt. Auch wenn kein Wert ein Zeilenvorschubzeichen enthält, kann der Änderungswert `delprioritychar` für den Dateityp problemlos verwendet werden. Dies führt jedoch zu einer geringfügigen Verlangsamung der Ladeoperation.

Wenn die zu ladenden Daten im Format ASC (Abstract Syntax Checker) vorliegen, müssen Sie eventuell durch eine zusätzliche Maßnahme sicherzustellen, dass keine abschließenden Leerzeichen in die geladenen Sicherheitskennsätze und Namen von Sicherheitskennsätzen eingefügt werden. Da beim ASCII-Format Spaltenpositionen als Begrenzer verwendet werden, kann dies beim Laden in Felder variabler Länge vorkommen. Verwenden Sie zum Abschneiden von abschließenden Leerstellen den Änderungswert `striptblanks` für den Dateityp.

Vom Standard abweichende Sicherheitskennsatzwerte

Sie können auch Datendateien laden, bei denen die Werte für die Sicherheitskennsätze Zeichenfolgen darstellen, die die Werte der Komponenten im jeweiligen Sicherheitskennsatz enthalten, z. B. `S:(ALPHA,BETA)`. Verwenden Sie hierzu den Änderungswert `seclabelchar` für den Dateityp. Bei Verwendung von `seclabelchar`

werden Werte für Spalten mit dem Datentyp DB2SECURITYLABEL als Zeichenfolgekennstanten gewertet, die den Sicherheitskennsatz im Zeichenfolgeformat für Sicherheitskennsätze enthalten. Liegt eine Zeichenfolge nicht im richtigen Format vor, wird die jeweilige Zeile nicht eingefügt und eine Warnung (SQLSTATE 01H53) zurückgegeben. Stellt eine Zeichenfolge keinen gültigen Sicherheitskennsatz dar, der Bestandteil der zum Schutz der Tabelle verwendeten Sicherheitsrichtlinie ist, wird die betreffende Zeile nicht eingefügt und eine Warnung (SQLSTATE 01H53) zurückgegeben.

Sie können auch Datendateien laden, bei denen die Werte für die Sicherheitskennsatzspalte Namen von Sicherheitskennsätzen sind. Verwenden Sie zum Laden derartiger Dateien den Änderungswert `seclabelname` für den Dateityp. Bei Verwendung von `seclabelname` werden alle Werte für Spalten mit dem Typ DB2SECURITYLABEL als Zeichenfolgekennstanten gewertet, die den Namen vorhandener Sicherheitskennsätze enthalten. Existiert kein Sicherheitskennsatz mit dem angegebenen Namen für die zum Schutz der Tabelle verwendete Sicherheitsrichtlinie, wird die betreffende Zeile nicht geladen und eine Warnung (SQLSTATE 01H53) zurückgegeben.

Zurückgewiesene Zeilen

Zeilen, die während des Ladens zurückgewiesen werden, werden entweder an eine Speicherauszugsdatei oder eine Ausnahmetabelle gesendet (wenn sie im Befehl **LOAD** angegeben sind), je nach der Ursache für das Zurückweisen der Zeilen. Zeilen, die aufgrund von Parsing-Fehlern zurückgewiesen werden, werden an die Speicherauszugsdatei gesendet. Zeilen, die die Sicherheitsrichtlinien verletzen, werden an die Ausnahmetabelle gesendet.

Anmerkung: Es kann keine Ausnahmetabelle angegeben werden, wenn die Zieltabelle eine XML-Spalte enthält.

Beispiele

Bei allen Beispielen liegt die Eingabedatendatei `myfile.del` im Format DEL vor. Die Daten werden jeweils in eine Tabelle mit der Bezeichnung REPS geladen, die mit der folgenden Anweisung erstellt wurde:

```
create table reps (row_label db2securitylabel,  
id integer,  
name char(30))  
security policy data_access_policy
```

Bei dem folgenden Beispiel wird davon ausgegangen, dass die Eingabedatei Sicherheitskennsätze im Standardformat enthält:

```
db2 load from myfile.del of del modified by delprioritychar insert into reps
```

Bei dem folgenden Beispiel wird davon ausgegangen, dass die Eingabedatei Sicherheitskennsätze im Zeichenfolgeformat für Sicherheitskennsätze enthält:

```
db2 load from myfile.del of del modified by seclabelchar insert into reps
```

Bei dem folgenden Beispiel wird davon ausgegangen, dass die Eingabedatei Namen von Sicherheitskennsätzen für die Sicherheitskennsatzspalte enthält:

```
db2 load from myfile.del of del modified by seclabelname insert into reps
```

Aspekte des Ladens von Identitätsspalten

Mit dem Dienstprogramm LOAD können Sie Daten in eine Tabelle laden, die eine Identitätsspalte enthält. Dabei spielt es keine Rolle, ob die Eingabedaten Identitätsspaltenwerte enthalten.

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.

In den meisten Fällen kann das Dienstprogramm LOAD nicht garantieren, dass Identitätsspaltenwerte den Zeilen in derselben Reihenfolge zugeordnet werden, wie diese Zeilen in der Datendatei angezeigt werden. Da die Zuordnung von Identitätsspaltenwerten vom Dienstprogramm LOAD parallel verwaltet wird, werden diese Werte in beliebiger Reihenfolge zugeordnet. Dabei gelten die folgenden Ausnahmen:

- In Einzelpartitionsdatenbanken werden Zeilen nicht parallel verarbeitet, wenn CPU_PARALLELISM auf 1 gesetzt ist. In diesem Fall werden Identitätsspaltenwerte implizit in derselben Reihenfolge zugeordnet, in der die Zeilen in der Datendatei angezeigt werden.
- In Mehrpartitionsdatenbanken werden Identitätsspaltenwerte in derselben Reihenfolge zugeordnet, in der die Zeilen in der Datendatei angezeigt werden, sofern die Identitätsspalte der Verteilungsschlüssel ist und ein einziger Partitionierungsagent vorhanden ist (wenn Sie also nicht mehrere Partitionierungsagenten bzw. den Änderungswert für den Dateityp anyorder angeben).

Wenn eine Tabelle in eine partitionierte Datenbank geladen wird und die Tabelle über eine Identitätsspalte im Partitionierungsschlüssel verfügt und der Änderungswert `identityoverride` nicht angegeben wurde, darf die Option `SAVECOUNT` nicht angegeben werden. Wenn der Partitionierungsschlüssel eine Identitätsspalte umfasst und die Identitätswerte generiert werden, dann ist zum Neustart des Ladevorgangs über die LOAD-Phase in mindestens einer Datenbankpartition der Neustart des gesamten Ladevorgangs vom Anfang der LOAD-Phase aus erforderlich. Dies bedeutet, dass keine Konsistenzpunkte vorhanden sein können.

Anmerkung: Eine Ladeoperation mit `RESTART` ist nicht zulässig, wenn alle im Folgenden aufgeführten Kriterien zutreffen:

- Die zu ladende Tabelle befindet sich in einer Umgebung mit partitionierten Datenbanken und enthält mindestens eine Identitätsspalte, die sich entweder im Verteilungsschlüssel befindet oder auf die über eine generierte Spalte verwiesen wird, die Teil des Verteilungsschlüssels ist.
- Der Änderungswert `identityoverride` ist nicht angegeben.
- Die zuvor ausgeführte, fehlgeschlagene Ladeoperation umfasste das Laden von Datenbankpartitionen, dieser Vorgang ist jedoch nach der LOAD-Phase fehlgeschlagen.

Stattdessen sollte eine Ladeoperation mit TERMINATE oder REPLACE ausgeführt werden.

Sie können das Laden von Daten in Tabellen mit Identitätsspalte mit den drei sich gegenseitig ausschließenden Änderungswerten für den Dateityp `identitymissing`, `identityignore` und `identityoverride` vereinfachen.

Laden von Daten ohne Identitätsspalten

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))
```

Wenn Sie Daten aus einer Datei (`load.del`) in die Tabelle TABLE1 laden wollen und diese Datei aus einer Tabelle exportiert wurde, die keine Identitätsspalte enthält, betrachten Sie das folgende Beispiel:

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

Eine Methode zum 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
```

Dieser Befehl führt dazu, dass die drei Spalten in der Datendatei in die Spalten `c1`, `c3` und `c4` von TABLE1 geladen werden. Für jede Zeile in `c2` wird dabei ein Wert generiert.

Laden von Daten mit Identitätsspalten

Der Änderungswert `identityignore` weist das Dienstprogramm LOAD an, dass die in der Eingabedatendatei vorhandenen Daten 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, können Sie 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
```

Laden von Daten mit vom Benutzer angegebenen Werten

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). Sie sollten außerdem beachten, dass bei Verwendung dieses Änderungswerts ein Verstoß gegen das Eindeutigkeitsmerkmal von Spalten eintreten kann, die mit `GENERATED ALWAYS` definiert sind. In dieser Situation müssen Sie eine Ladeoperation mit `TERMINATE` und anschließend eine Ladeoperation mit `INSERT` oder `REPLACE` ausführen.

Aspekte des Ladens generierter Spalten

Sie können Daten in eine Tabelle mit generierten Spalten (Tabellen mit Identitätsspalte ausgenommen) laden. Dabei spielt es keine Rolle, ob die Eingabedaten Werte für generierte Spalten enthalten. Das Dienstprogramm `LOAD` generiert die Spaltenwerte.

Sofern keine Änderungswerte für den Datentyp 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.

Sie können das Laden von Daten in Tabellen mit generierter Spalte mit den drei sich gegenseitig ausschließenden Änderungswerten für den Datentyp `generatedmissing`, `generatedignore` und `generatedoverride` vereinfachen.

Laden von Daten ohne generierte Spalten

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))
```

Wenn Sie Daten aus einer Datei (load.del) in die Tabelle TABLE1 laden wollen und diese Datei aus einer Tabelle exportiert wurde, die keine generierten Spalten enthält, betrachten Sie das folgende Beispiel:

```
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 für den Dateityp generatedmissing:

```
DB2 LOAD FROM load.del of del MODIFIED BY generatedmissing
REPLACE INTO table1
```

Dieser Befehl führt dazu, dass die drei Spalten der Datendatei in die Spalten c1, c2 und c3 von TABLE1 geladen werden. Aufgrund des Änderungswerts generatedmissing werden die Werte der Spalten g1 und g2 von TABLE1 automatisch generiert und werden keiner der Datendateispalten zugeordnet.

Laden von Daten mit generierten Spalten

Der Änderungswert generatedignore 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: Sie möchten in die oben definierte Tabelle TABLE1 die folgenden Daten aus einer Datendatei (load.del) 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), wenn keine Änderungswerte für den Dateityp verwendet werden, die den generierten Spalten zugeordnet sind. 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
```

Dieser Befehl führt dazu, dass die Spalten der Datendatei in die Spalten c1 (mit Daten 1, 2, 3), c2 (mit Daten 5, 6, 7) und c3 (mit Daten J, K, I) von TABLE1 geladen werden. Aufgrund des Änderungswerts generatedignore werden die Werte der Spalten g1 und g2 von TABLE1 automatisch generiert und die Datendateispalten (10, 11, 12 sowie 15, 16, 17) werden ignoriert.

Laden von Daten mit vom Benutzer angegebenen Werten

Der Änderungswert generatedoverride wird verwendet, um benutzerdefinierte Werte in eine Tabelle mit generierten Spalten zu laden. Dies kann bei der Migrati-

on von Daten aus einem anderen Datenbanksystem oder beim Laden einer Tabelle aus Daten nützlich sein, die mit der Option RECOVER DROPPED TABLE des Befehls **ROLLFORWARD DATABASE** wiederhergestellt wurden. Bei Verwendung dieses Änderungswerts werden alle Zeilen ohne Daten (oder mit Nulldaten) für generierte Spalten, die keine Nullwerte enthalten können, zurückgewiesen (SQL3116W).

Bei Verwendung dieses Änderungswertes wird die Tabelle nach der Ladeoperation in den Status "Festlegen der Integrität anstehend" versetzt. Um den Status "Festlegen der Integrität 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 "Festlegen der Integrität 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
```

Befindet sich eine generierte Spalte in einem der Partitionierungs-, Dimensions- oder Verteilungsschlüssel, wird der Änderungswert `generatedoverride` ignoriert, und das Dienstprogramm LOAD generiert Werte, als wäre der Änderungswert `generatedignore` angegeben. Diese Vorgehensweise wird angewendet, um ein Szenario zu vermeiden, in dem der benutzerdefinierte Spaltenwert zu einem Konflikt mit der Definition der generierten Spalte führt. Dies würde dazu führen, dass der Ergebnisdatensatz an der falschen physischen Position (wie beispielsweise in der falschen Datenpartition, dem falschen MDC-Block oder der falschen Datenbankpartition) abgelegt werden würde.

Anmerkung: In einem Fall unterstützt LOAD Werte von generierten Spalten NICHT, und zwar dann nicht, wenn einer der Ausdrücke der generierten Spalten eine benutzerdefinierte Funktion enthält, die abgeschirmt (FENCED) ist. Wird versucht, Daten in eine solche Tabelle zu laden, schlägt die Ladeoperation fehl. Sie können für diese Typen von generierten Spalten jedoch eigene Werte angeben, indem Sie den Änderungswert `generatedoverride` für den Datentyp verwenden.

Aspekte zur Verwendung eines Clients bis Version 7 mit einem Server ab Version 8

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 "Festlegen der Integrität anstehend". Falls eine Tabelle in den Status "Festlegen der Integrität 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 der folgenden Anweisung den Status "Festlegen der Integrität anstehend" für die Tabelle aufheben und die Generierung von Werten erzwingen:

```
SET INTEGRITY FOR tabellenname IMMEDIATE CHECKED FORCE GENERATED;
```

Versetzen von Daten mithilfe des Datentyps CURSOR

Durch die Angabe des Datentyps 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.

Darüber hinaus können Sie Daten aus einer weiteren Datenbank laden, indem Sie in der SQL-Abfrage auf einen Kurznamen verweisen, indem Sie die Option DATABASE der Anweisung DECLARE CURSOR verwenden oder indem Sie bei der API-Schnittstelle den Datenträgereintrag 'sqlu_remotefetch_entry' benutzen.

Für das Versetzen von Daten mithilfe des Dateityps CURSOR stehen drei Methoden zur Verfügung. Bei der ersten Methode wird der Befehlszeilenprozessor (CLP) verwendet, bei der zweiten Methode die API und bei der dritten die Prozedur ADMIN_CMD. Die Hauptunterschiede zwischen dem CLP und der Prozedur ADMIN_CMD werden in der folgenden Tabelle gezeigt.

Tabelle 26. Unterschiede zwischen dem CLP und der Prozedur ADMIN_CMD.

Unterschiede	Befehlszeilenprozessor (CLP)	Prozedur ADMIN_CMD
Syntax	Die Abfrageanweisung und die Quelldatenbank, die vom Cursor verwendet werden, werden außerhalb des Befehls LOAD mithilfe der Anweisung DECLARE CURSOR definiert.	Die Abfrageanweisung und die Quelldatenbank, die vom Cursor verwendet werden, werden innerhalb des Befehls LOAD wie folgt definiert: LOAD from (DATABASE datenbankaliasname abfrageanweisung)
Benutzerberechtigung für den Zugriff auf eine andere Datenbank	Befinden sich die Daten in einer anderen Datenbank als derjenigen, zu der momentan eine Verbindung besteht, muss das Schlüsselwort DATABASE in der Anweisung DECLARE CURSOR verwendet werden. In derselben Anweisung können auch die Benutzer-ID und das Kennwort angegeben werden. Werden Benutzer-ID und Kennwort in der Anweisung DECLARE CURSOR nicht angegeben, werden die Benutzer-ID und das Kennwort, die für die Verbindung zur Quelldatenbank explizit angegeben wurden, für den Zugriff auf die Zieldatenbank verwendet.	Befinden sich die Daten in einer anderen Datenbank als derjenigen, zu der momentan eine Verbindung besteht, muss das Schlüsselwort DATABASE im Befehl LOAD vor der Abfrageanweisung verwendet werden. Für den Zugriff auf die Zieldatenbank sind die Benutzer-ID und das Kennwort erforderlich, die für die Verbindung zur Quelldatenbank explizit angegeben wurden. Für die Quelldatenbank selbst kann keine Benutzer-ID und kein Kennwort angegeben werden. Daher kann die Prozedur ADMIN_CMD nicht zum Laden verwendet werden, wenn beim Herstellen der Verbindung zur Datenbank keine Benutzer-ID und kein Kennwort angegeben wurden oder wenn die Benutzer-ID und das Kennwort, die angegeben wurden, nicht für die Authentifizierung der Quelldatenbank verwendet werden können.

Damit eine LOAD FROM CURSOR-Operation über den Befehlszeilenprozessor (CLP) ausgeführt werden kann, muss zunächst ein Cursor für eine SQL-Abfrage deklariert werden. Sobald ein Cursor deklariert ist, 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:

1. Angenommen, eine Quellentabelle und eine Zieltabelle mit den folgenden Definitionen befinden sich beide in derselben Datenbank:

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
```

Anmerkung: 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. Definieren Sie den Wert *piSourceList* der *sqlu_media_list*-Struktur zur Verwendung der Struktur *sqlu_statement_entry* und des Datenträgertyps *SQLU_SQL_STMT*, und definieren Sie den Wert *piFileType* als *SQL_CURSOR*.

2. Angenommen, die Quellentabelle und die Zieltabelle mit den folgenden Definitionen befinden sich in unterschiedlichen Datenbanken:

Tabelle ABC.TABLE1 in Datenbank 'dbsource' enthält 3 Spalten:

- ONE INT
- TWO CHAR(10)
- THREE DATE

Tabelle ABC.TABLE2 in Datenbank 'dbtarget' enthält 3 Spalten:

- ONE VARCHAR
- TWO INT
- THREE DATE

Wenn Sie die Föderation aktiviert und die Datenquelle (dsdbsource) katalogisiert haben, können Sie einen Kurznamen für die Quelldatenbank deklarieren und anschließend einen Cursor für diesen Kurznamen deklarieren und den Befehl LOAD mit der Option FROM CURSOR aufrufen, wie in folgendem Beispiel gezeigt wird:

```
CREATE NICKNAME myschema1.table1 FOR dsdbsource.abc.table1
DECLARE mycurs CURSOR FOR SELECT TWO,ONE,THREE FROM myschema1.table1
LOAD FROM mycurs OF cursor INSERT INTO abc.table2
```

Sie haben auch die Möglichkeit, die Option DATABASE der Anweisung DECLARE CURSOR zu verwenden, wie in folgendem Beispiel gezeigt wird:

```
DECLARE mycurs CURSOR DATABASE dbsource USER dsciaraf USING mypasswd
FOR SELECT TWO,ONE,THREE FROM abc.table1
LOAD FROM mycurs OF cursor INSERT INTO abc.table2
```

Die Verwendung der Option DATABASE der Anweisung DECLARE CURSOR (bei Verwendung der API von LOAD auch als Datenträgertyp 'remotefetch' bekannt) bietet einige Vorteile gegenüber der Verwendung eines Kurznamens:

Leistung

Das Abrufen von Daten unter Verwendung des Datenträgertyps 'remotefetch' ist nahtlos in eine Ladeoperation integriert. Beim Abrufen eines Datensatzes gibt es weniger Übergangsebenen als bei der Verwendung eines Kurznamens. Ein weiterer Vorteil: Wenn die Quellentabelle und die Zieltabelle in einer Mehrpartitionsdatenbank auf identische Weise verteilt werden, kann das Dienstprogramm LOAD das Abrufen von Daten parallelisieren. Dies kann weiter zur Verbesserung der Leistung beitragen.

Hoher Bedienungskomfort

Es ist nicht erforderlich, die Funktion für Föderation zu aktivieren, eine ferne Datenquelle zu definieren oder einen Kurznamen zu deklarieren. Es ist lediglich erforderlich, die Option DATABASE (und gegebenenfalls die Optionen USER und USING) anzugeben.

Während diese Methode für katalogisierte Datenbanken verwendet werden kann, bietet die Verwendung von Kurznamen eine zuverlässige Möglichkeit, Daten aus verschiedenen Datenquellen abzurufen, die nicht ohne Weiteres katalogisiert werden können.

Zur Unterstützung dieser Funktionalität von 'remotefetch' verwendet das Dienstprogramm LOAD eine Infrastruktur, die die Funktion SOURCEUSEREXIT unterstützt. Das Dienstprogramm LOAD startet einen Prozess, der als Anwendung ausgeführt wird, um die Verbindung zur Quelldatenbank zu verwalten und den Datenabruf durchzuführen. Diese Anwendung ist einer eigenen Transaktion zugeordnet und nicht der Transaktion, unter der das Dienstprogramm LOAD ausgeführt wird.

Anmerkung:

1. Das vorstehende Beispiel zeigt, wie Daten aus einer SQL-Abfrage für eine katalogisierte Datenbank über den Befehlszeilenprozessor (CLP) unter Verwendung der Option DATABASE der Anweisung DECLARE CURSOR geladen werden können. Das Laden aus einer SQL-Abfrage für eine katalogisierte Datenbank kann jedoch auch über die API db2Load erfolgen, indem die Werte *piSourceList* und *piFileType* der *db2LoadStruct*-Struktur so definiert werden, dass der Datenträger eintrag 'sqlu_remotefetch_entry' bzw. der Datenträgertyp SQLU_REMOTEFETCH verwendet wird.
2. Wie in vorstehendem Beispiel veranschaulicht wird, müssen die Quellenspaltentypen der SQL-Abfrage nicht mit den entsprechenden Zielspaltentypen identisch sein. Kompatibilität zwischen den Spaltentypen ist jedoch erforderlich.

Einschränkungen

Beim Laden von Daten über einen Cursor, der mit der Option DATABASE definiert wurde (oder bei Verwendung des Datenträger eintrags 'sqlu_remotefetch_entry' über die API 'db2Load'), gelten die folgenden Einschränkungen:

1. Die Option SOURCEUSEREXIT kann nicht gleichzeitig angegeben werden.
2. Die Option METHOD N wird nicht unterstützt.
3. Der Änderungswert `usedefaults` für den Dateityp wird nicht unterstützt.

Weitergeben von abhängigen sofort gespeicherten Zwischenspeichertabellen

Wenn die geladene Tabelle die zugrunde 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 zugrunde liegenden Tabellen entsprechen, an die Zwischenspeichertabellen angehängt. Die inkrementelle Weitergabe beschleunigt sich, wenn große zugrunde liegende Tabellen kleine Mengen von angehängten Daten enthalten. Die Leistung wird ebenfalls verbessert, wenn die Zwischenspeichertabelle zur Aktualisierung ihrer abhängigen, verzögerten MQT 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 MQT verwendet werden. Stattdessen muss im Verwaltungsprozess für die MQT eine vollständige Aktualisierung erfolgen.

Falls eine Tabelle den 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 zugrunde liegende Tabelle der Zwischenspeichertabelle ausgeführt, oder die Option NOT LOGGED INITIALLY WITH EMPTY TABLE wurde nach der letzten Überprüfung der Integrität für die zugrunde liegende Tabelle aktiviert.
- Die abhängige MQT der Zwischenspeichertabelle oder die Zwischenspeichertabelle selbst wurde im Ersetzungsmodus oder im Einfügemodus (REPLACE oder INSERT) geladen.
- Für eine zugrunde liegende Tabelle wurde der Status "Festlegen der Integrität anstehend" aufgehoben, bevor die Zwischenspeichertabelle weitergegeben wurde (durch Verwendung der Option FULL ACCESS während der Überprüfung der Integrität).
- Die Integrität einer zugrunde liegenden Tabelle der Zwischenspeichertabelle wurde nicht inkrementell überprüft.
- Der Tabellenbereich, der die Zwischenspeichertabelle oder ihre zugrunde liegende Tabelle enthält, wurde bis zu einem bestimmten Zeitpunkt aktualisierend wiederhergestellt, und die Zwischenspeichertabelle und ihre zugrunde 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 zugrunde liegende Tabelle UT1 der Zwischenspeichertabelle G1 und ihre abhängige, verzögerte MQT 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;

```

Aktualisieren von abhängigen IMQTs (Immediate Materialized Query Tables)

Wenn die zugrunde liegende Tabelle einer aktualisierten IMQT (Immediate Materialized Query Table, sofort gespeicherte Abfragetabelle) unter Verwendung der Option INSERT geladen wird, bewirkt die Ausführung der Anweisung SET INTEGRITY für die abhängigen MQTs, für die REFRESH IMMEDIATE definiert ist, dass die MQT inkrementell aktualisiert wird.

Während einer inkrementellen Aktualisierung werden die Zeilen, die den angehängten Zeilen in den zugrunde liegenden Tabellen entsprechen, aktualisiert und in die MQTs eingefügt. Die inkrementelle Aktualisierung beschleunigt sich, wenn große zugrunde 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 MQT) verwendet wird.

Wenn die Option INCREMENTAL angegeben ist, eine inkrementelle Verarbeitung der MQT jedoch nicht möglich ist, wird in den folgenden Fällen ein Fehler zurückgegeben:

- Eine LOAD REPLACE-Operation wurde in einer zugrunde liegenden Tabelle der MQT ausgeführt, oder die Option NOT LOGGED INITIALLY WITH EMPTY TABLE wurde aktiviert, seitdem die letzte Überprüfung auf Integritätsbedingungen für die zugrunde liegende Tabelle stattgefunden hat.
- Die MQT wurde geladen (im Ersetzungsmodus oder im Einfügemodus).
- Für eine zugrunde liegende Tabelle wurde der Status "Festlegen der Integrität anstehend" aufgehoben, bevor die MQT aktualisiert wurde (durch Verwendung der Option FULL ACCESS während der Überprüfung der Integrität).
- Die Integrität einer zugrunde liegenden Tabelle der MQT wurde nicht inkrementell überprüft.
- Die MQT befand sich vor dem Upgrade im Status "Festlegen der Integrität anstehend".
- Der Tabellenbereich, der die MQT oder ihre zugrunde liegende Tabelle enthält, wurde bis zu einem bestimmten Zeitpunkt aktualisierend wiederhergestellt, und die MQT und ihre zugrunde liegende Tabelle befinden sich in unterschiedlichen Tabellenbereichen.

Wenn die MQT 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 ist, 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 zugrunde liegende Tabelle UT1 der MQT mit der Bezeichnung 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 Ta-

belle 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;
```

Aspekte des mehrdimensionalen Clustering

Für das Laden von Daten in MDC-Tabellen (Multi-Dimensional Clustering - mehrdimensionales Clustering) gelten die folgenden Einschränkungen:

- Die Option `SAVECOUNT` des Befehls `LOAD` wird nicht unterstützt.
- Der Änderungswert `total freespace` für den Dateityp wird nicht unterstützt, da diese Tabellen ihren freien Speicherbereich selbst verwalten.
- Der Änderungswert `anyorder` für den Dateityp ist für MDC-Tabellen erforderlich. Werden Daten ohne den Änderungswert `anyorder` in eine MDC-Tabelle geladen, wird dieser Änderungswert vom Dienstprogramm explizit aktiviert.

Bei Verwendung des Befehls **LOAD** mit einer MDC-Tabelle werden Verstöße gegen eindeutige Integritätsbedingungen wie folgt behandelt:

- Wenn die Tabelle vor der Ladeoperation einen eindeutigen Schlüssel enthielt und doppelte Datensätze in die Tabelle geladen werden, verbleibt der ursprüngliche Datensatz 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. Ab Version 9.5 kann der Wert für die Option `DATA BUFFER` des Befehls **LOAD** den Wert von `util_heap_sz` temporär überschreiten, falls im System mehr Speicher zur Verfügung steht. .

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 zufriedenstellende 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 temporäre Systemtabelle 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 Tabellenbereich für temporäre Tabellen groß genug ist.

Versetzen von Daten mithilfe einer angepassten Anwendung (Benutzerexit)

Die LOAD-Option SOURCEUSEREXIT stellt eine Funktion bereit, über die das Dienstprogramm LOAD ein angepasstes Script bzw. eine angepasste ausführbare Datei ausführen kann. Dieses Script bzw. diese Datei wird im vorliegenden Handbuch als *Benutzerexit* bezeichnet.

Die Aufgabe des Benutzerexits besteht darin, mindestens eine benannte Pipe mit Daten aufzufüllen, die gleichzeitig vom Dienstprogramm LOAD gelesen werden. In einer Mehrpartitionsdatenbank können mehrere Instanzen des Benutzerexits gleichzeitig aufgerufen werden, um eine parallele Verarbeitung der Eingabedaten zu erreichen.

Wie Abb. 5 auf Seite 168 zeigt, erstellt das Dienstprogramm LOAD mindestens eine benannte Pipe und startet einen Prozess zum Ausführen des angepassten ausführbaren Codes. Der Benutzerexit führt der bzw. den benannten Pipe(s) Daten zu, während das Dienstprogramm LOAD die Daten gleichzeitig liest.

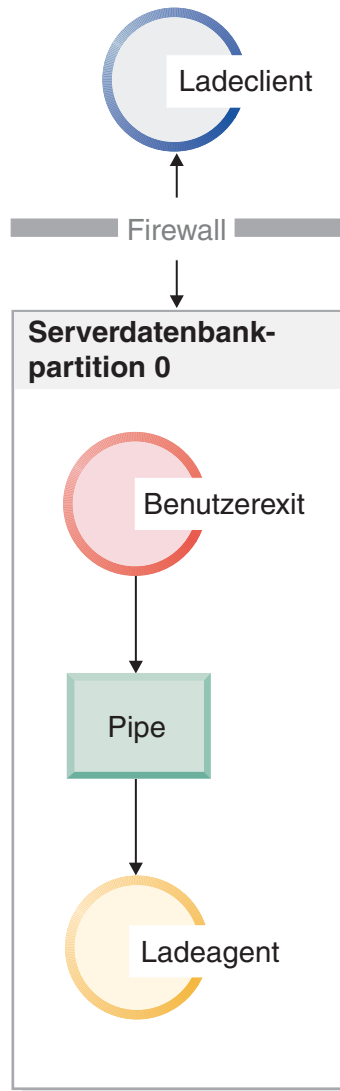


Abbildung 5. Das Dienstprogramm LOAD liest aus der Pipe und verarbeitet die ankommenden Daten.

Die Daten, die der Pipe zugeführt werden, müssen den angegebenen LOAD-Optionen entsprechen, einschließlich Dateityp und allen Änderungswerten für den Dateityp. Die angegebenen Datendateien werden vom Dienstprogramm LOAD nicht direkt gelesen. Stattdessen werden die angegebenen Datendateien als Argumente an den Benutzerexit übergeben, wenn dieser ausgeführt wird.

Aufrufen des Benutzerexits

Der Benutzerexit muss sich im Unterverzeichnis 'bin' des DB2-Installationsverzeichnisses (häufig 'sqlib') befinden. Das Dienstprogramm LOAD ruft den ausführbaren Code des Benutzerexits mit folgenden Befehlszeilenparametern auf:

```

<pipebasisdateiname> <anzahl_der_quellendatenträger>
<quellendatenträger_1> <quellendatenträger_2> ... <benutzerexit-id>
<anzahl_der_benutzerexits> <nummer_der_datenbankpartition>

```


Dabei gilt Folgendes:

<pipebasisdateiname>

Dies ist der Basisname für benannte Pipes, die das Dienstprogramm LOAD erstellt und aus denen es Daten liest. Das Dienstprogramm erstellt für jede im Befehl LOAD angegebene Quellendatei jeweils eine Pipe, und an den Namen jeder dieser Pipes wird die Erweiterung `.xxx` angehängt, wobei `xxx` der Index der angegebenen Quellendatei ist. Werden beispielsweise 2 Quellendateien im Befehl LOAD angegeben und lautet der an den Benutzerexit übergebene Wert für das Argument `<pipebasisdateiname>` `pipe123`, werden den beiden benannten Pipes `pipe123.000` und `pipe123.001` Daten vom Benutzerexit zugeführt. In Umgebung mit partitionierten Datenbanken wird die Nummer der Datenbankpartition (DBPARTITION) `.yyy` vom Dienstprogramm LOAD an den Basisnamen der Pipes angehängt, sodass die Namen der Pipes das folgende Format aufweisen: `pipe123.xxx.yyy..`

<anzahl_der_quellendatenträger>

Dies ist die Anzahl der nachfolgenden Datenträgerargumente.

<quellendatenträger_1> <quellendatenträger_2> ...

Dies ist die Liste der Quellendateien (mindestens eine), die im Befehl LOAD angegeben sind. Jede Quellendatei ist in doppelte Anführungszeichen gesetzt.

<benutzerexit_id>

Dies ist ein Sonderwert, der bei aktivierter Option PARALLELIZE nützlich ist. Dieser ganzzahlige Wert (von 1 bis N, wobei N die Gesamtanzahl der gestarteten Benutzerexits ist) identifiziert eine bestimmte Instanz eines aktiven Benutzerexits. Bei inaktivierter Option PARALLELIZE ist 1 der Standardwert.

<anzahl_der_benutzerexits>

Dies ist ein Sonderwert, der bei aktivierter Option PARALLELIZE nützlich ist. Dieser Wert stellt die Gesamtanzahl der gleichzeitig aktiven Benutzerexits dar. Bei inaktivierter Option PARALLELIZE ist 1 der Standardwert.

<nummer_der_datenbankpartition>

Dies ist ein Sonderwert, der bei aktivierter Option PARALLELIZE nützlich ist. Hierbei handelt es sich um die Nummer der Datenbankpartition (DBPARTITION), auf der der Benutzerexit ausgeführt wird. Bei inaktivierter Option PARALLELIZE ist 0 der Standardwert.

Zusätzliche Optionen und Funktionen

Im folgenden Abschnitt werden die zusätzlichen Optionen der Funktion SOURCEUSEREXIT beschrieben:

REDIRECT

Mit dieser Option können Sie Daten an die Kennung STDIN übergeben oder Daten aus den Kennungen STDOUT und STDERR des Benutzerexitprozesses erfassen.

INPUT FROM BUFFER <puffer>

Mit dieser Option können Sie Informationen direkt in den STDIN-Eingabedatenstrom Ihres Benutzerexits übergeben. Nach dem Start des Prozesses, der den Benutzerexit ausführt, fordert das Dienstprogramm LOAD den Dateideskriptor für den STDIN-Eingabedatenstrom dieses neuen Prozesses an und übergibt den angegebenen Puffer. Der Benutzerexit liest die Informationen aus STDIN. Das Dienstprogramm LOAD sendet den Inhalt von <puffer> einfach über STDIN an den Benutzerexit, ohne den Pufferinhalt zu interpretieren oder zu ändern. Wenn ein Benutzerexit beispielsweise zwei Werte aus STDIN lesen soll (eine Benutzer-ID mit 8 Byte und ein Kennwort mit 8 Byte), könnte ein in C geschriebener ausführbarer Benutzerexit-Code folgende Zeilen enthalten:

```
rc = read (stdin, pUserID, 8);  
rc = read (stdin, pPasswd, 8);
```

Ein Benutzer könnte diese Informationen mithilfe der Option INPUT FROM BUFFER übergeben, wie in folgendem Befehl LOAD gezeigt:

```
LOAD FROM myfile1 OF DEL INSERT INTO table1  
SOURCEUSEREXIT myuserexit1 REDIRECT INPUT FROM BUFFER myuseridmypasswd
```

Anmerkung: Das Dienstprogramm LOAD begrenzt die Größe von <puffer> auf die Maximalgröße eines LOB-Wertes. Innerhalb des Befehlszeilenprozessors (CLP) ist die Größe von <puffer> jedoch auf die Maximalgröße einer CLP-Anweisung begrenzt. Innerhalb von CLP wird außerdem empfohlen, für den Inhalt von <puffer> nur traditionelle ASCII-Zeichen zu verwenden. Diese Einschränkungen lassen sich umgehen, indem das Dienstprogramm LOAD mithilfe der API db2Load aufgerufen wird oder indem stattdessen die Option INPUT FROM FILE verwendet wird.

INPUT FROM FILE <dateiname>

Mit dieser Option können Sie den Inhalt einer Datei auf Clientseite direkt in den STDIN-Eingabedatenstrom des Benutzerexits übergeben. Diese Option ist beinahe mit der Option INPUT FROM BUFFER identisch, vermeidet jedoch die mögliche CLP-Einschränkung. Der Dateiname muss eine vollständig qualifizierte Datei auf der Clientseite sein und darf nicht größer sein als die Maximalgröße eines LOB-Wertes.

OUTPUT TO FILE <dateiname>

Mit dieser Option können Sie die STDOUT- und STDERR-Datenströme aus dem Benutzerexitprozess in einer Datei auf der Serverseite erfassen. Nach dem Start des Prozesses, der den ausführbaren Code des Benutzerexits ausführt, leitet das Dienstprogramm LOAD die Kennungen STDOUT und STDERR aus diesem neuen Prozess in die Datei um, deren Name angegeben wurde. Diese Option ist nützlich beim Debugging und Protokollieren von Fehlern und Aktivitäten innerhalb des Benutzerexits. Bei dem Dateinamen muss es sich um eine vollständig qualifizierte Datei auf der Serverseite handeln. Ist die Option PARALLELIZE aktiviert, ist pro Benutzerexit eine Datei vorhanden, und an jede Datei wird eine dreistellige numerische Kennung angehängt (*dateiname.000*).

PARALLELIZE

Mit dieser Option kann der Durchsatz der im Dienstprogramm LOAD eingehenden Daten erhöht werden, indem mehrere Benutzerexitprozesse gleichzeitig aufgerufen werden. Diese Option steht nur für Mehrpartitionsdatenbanken zur Verfügung. Die Anzahl der aufgerufenen Instanzen des Benutzerexits entspricht der Anzahl der Partitionierungsagenten, wenn die Daten während der Ladeoperation auf mehrere Datenbankpartitionen verteilt werden sollen. Ansonsten entspricht sie der Anzahl der Ladeagenten.

Die an die einzelnen Benutzerexits übergebenen Argumente `<benutzerexit_id>`, `<anzahl_der_benutzerexits>` und `<nummer_der_datenbankpartition>` geben jeweils die eindeutige Kennung (1 bis N), die Gesamtanzahl der Benutzerexits (N) und die Nummer der Datenbankpartition (DBPARTITION), unter der die Benutzerexitinstanz ausgeführt wird, wieder. Sie sollten sicherstellen, dass Daten, die von einem der Benutzerexitprozesse in die benannte Pipe geschrieben werden, nicht von den anderen gleichzeitig ablaufenden Prozessen dupliziert werden. Obwohl es mehrere Möglichkeiten für eine Benutzerexitanwendung gibt, dies zu erreichen, können diese Werte dabei helfen zu verhindern, dass Daten dupliziert werden. Beispiel: Wenn alle Datensätze jeweils einen eindeutigen ganzzahligen Spaltenwert enthalten, könnte eine Benutzerexitanwendung die Werte von `<benutzerexit-ID>` und `<anzahl_der_benutzerexits>` verwenden, um sicherzustellen, dass jede Benutzerexitinstanz eine eindeutige Ergebnismenge für die entsprechende benannte Pipe zurückgibt. Die Benutzerexitanwendung könnte das Merkmal **MODULUS** wie folgt verwenden:

```
i = <benutzerexit_id>
N = <anzahl_der_benutzerexits>

foreach record
{
    if ((unique-integer MOD N) == i)
    {
        write this record to my named-pipe
    }
}
```

Die Anzahl der gestarteten Benutzerexitprozesse hängt von dem Verteilungsmodus ab, der für die Datenbankpartitionierung angegeben wurde:

1. Wie Abb. 6 auf Seite 172 zeigt, wird für jeden Agenten für Partitionierungsvorbereitung jeweils ein Benutzerexitprozess gestartet, wenn `PARTITION_AND_LOAD` (Standardeinstellung) oder `PARTITION_ONLY` ohne `PARALLEL` angegeben ist.

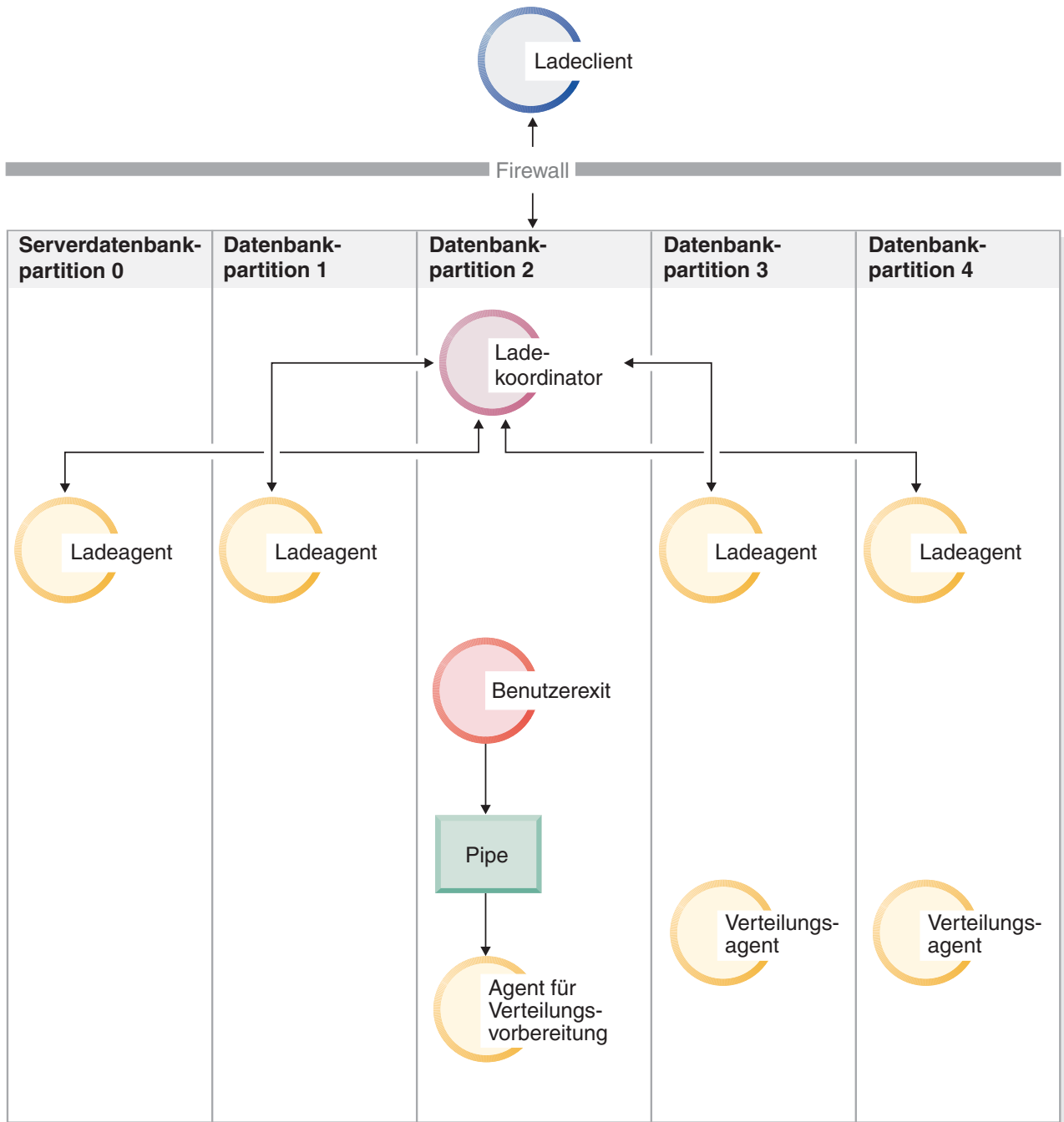


Abbildung 6. Verschiedene ausgeführte Tasks, wenn `PARTITION_AND_LOAD` (Standardeinstellung) oder `PARTITION_ONLY` ohne `PARALLEL` angegeben ist.

- Wie Abb. 7 auf Seite 173 zeigt, wird für jeden Partitionierungsagenten jeweils ein Benutzerexitprozess gestartet, wenn `PARTITION_AND_LOAD` (Standardeinstellung) oder `PARTITION_ONLY` mit der Option `PARALLEL` angegeben ist.

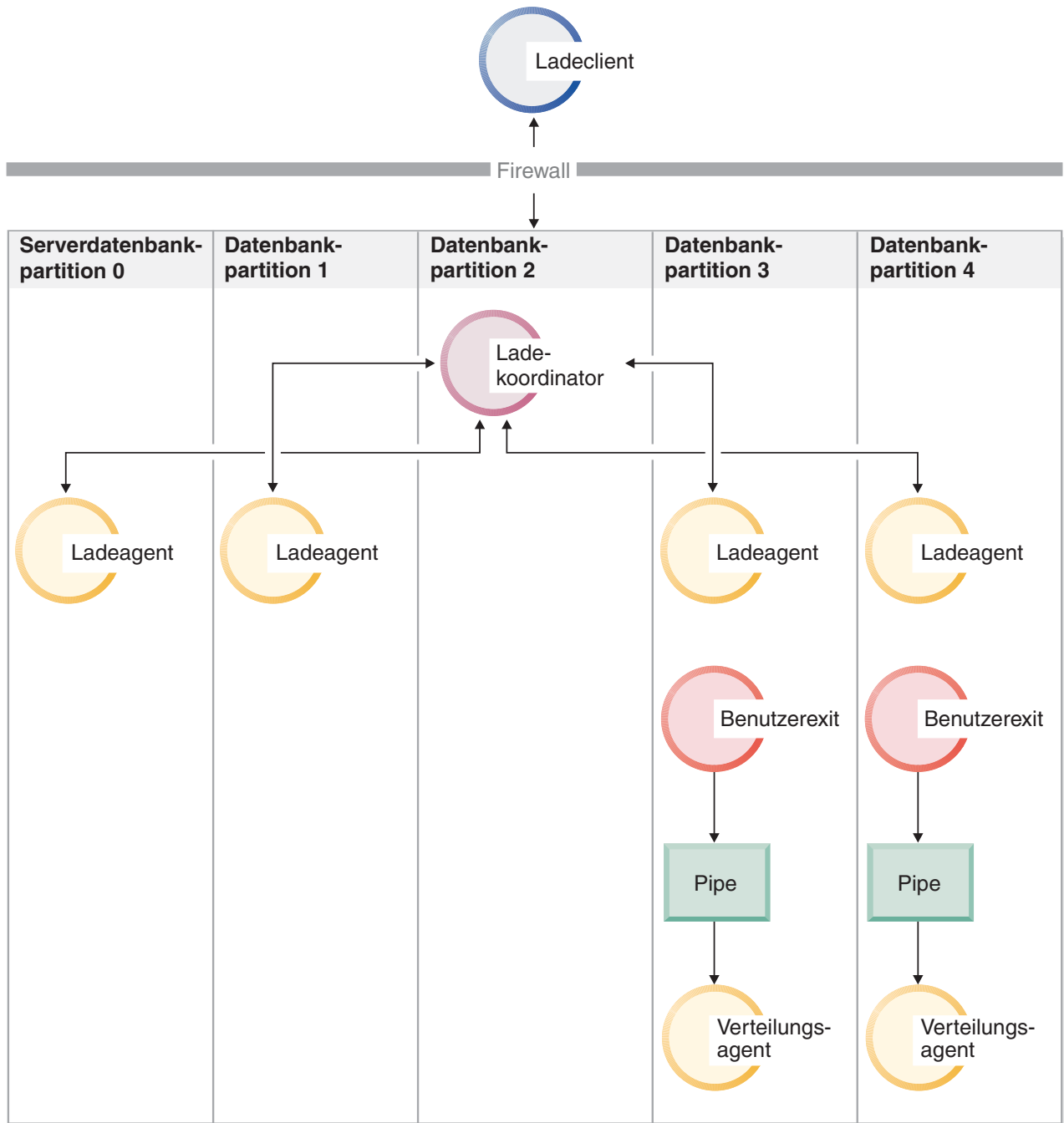


Abbildung 7. Verschiedene ausgeführte Tasks, wenn PARTITION_AND_LOAD (Standardeinstellung) oder PARTITION_ONLY mit PARALLEL angegeben ist.

- Wie Abb. 8 auf Seite 174 zeigt, wird für jeden Ladeagenten jeweils ein Benutzerexitprozess gestartet, wenn LOAD_ONLY oder LOAD_ONLY_VERIFY_PART angegeben ist.

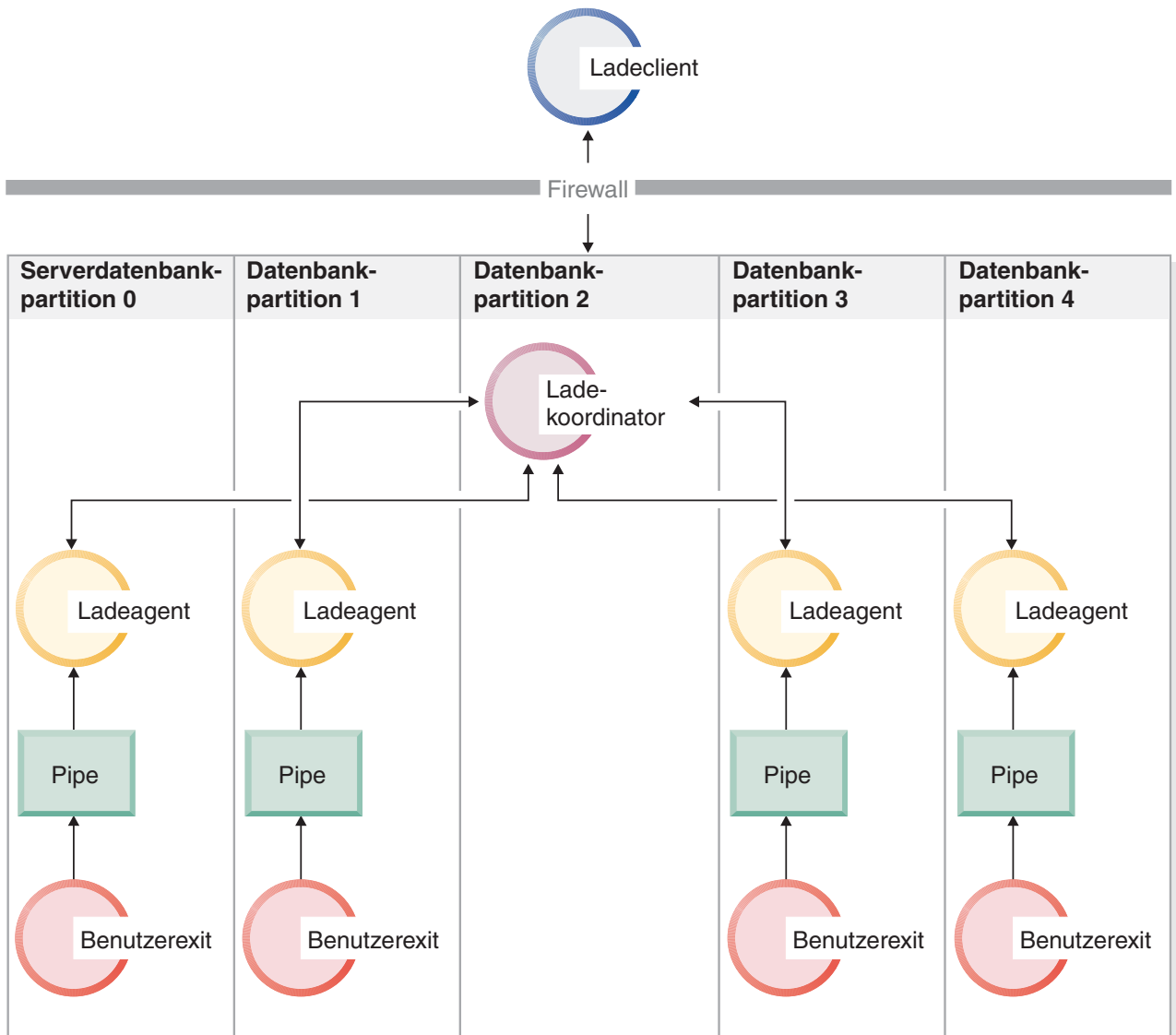


Abbildung 8. Verschiedene ausgeführte Tasks, wenn `LOAD_ONLY` oder `LOAD_ONLY_VERIFY_PART` angegeben ist.

Einschränkungen

- Die Optionen `LOAD_ONLY` und `LOAD_ONLY_VERIFY_PART` des Modus 'partitioned-db-cfg' werden nur unterstützt, wenn die `SOURCEUSEREXIT`-Option `PARALLELIZE` angegeben ist.

Beispiele

Beispiel 1: LOAD-Benutzerexitskript, das für alle Datensätze der Quelldatenträgerdatei alle Tabulatorzeichen '\t' durch Kommas ',' ersetzt. Verwenden Sie zum Aufrufen des Dienstprogramms LOAD mithilfe dieses Benutzerexitskripts einen Befehl ähnlich dem folgenden:

```
DB2 LOAD FROM /path/file1 OF DEL INSERT INTO schema1.table1
SOURCEUSEREXIT example1.pl REDIRECT OUTPUT TO FILE /path/ue_msgs.txt
```

Beachten Sie, dass der Benutzerexit im Ordner `sqllib/bin/` gespeichert sein muss und Ausführungsrechte erfordert.

example1.pl:

```
#!/bin/perl
# Dateiname: example1.pl
#
# Dieses Script ist ein einfaches Beispiel für einen Benutzerexit für die Funktion
# SOURCEUSEREXIT des Dienstprogramms LOAD. Es ersetzt für alle Datensätze der
# Quelldatenträgerdatei alle Tabulatorzeichen '\t' durch Kommas ','.
#
# Mit einem Befehl ähnlich dem folgenden wird LOAD über diesen Benutzerexit aufgerufen:
# db2 LOAD FROM /path/file1 OF DEL INSERT INTO schema1.table1
# SOURCEUSEREXIT example1.pl REDIRECT OUTPUT TO FILE /path/ue_msgs.txt
#
# Der Benutzerexit muss im Ordner sqllib/bin/ gespeichert sein und erfordert
# Ausführungsrechte.
#-----
if ($#ARGV < 5)
{
    print "Ungültige Anzahl von Argumenten:\n@ARGV\n";
    print "LOAD muss den Benutzerexit mit mindestens 5 Argumenten aufrufen:\n";
    print "<pipebasisdateiname> <anzahl_der_quelldatenträger> ";
    print "<quelldatenträger_1> <quelldatenträger_2> ... <benutzerexit-id> ";
    print "<anzahl_der_benutzerexits> <nummer_der_datenbankpartition> ";
    print "<optional:_eingabe_umleiten> \n";
    die;
}
# FIFO-Ausgabedatei öffnen (LOAD liest Daten aus dieser Pipe)
#-----
$basePipeName = $ARGV[0];
$outputPipeName = sprintf("%s.000", $basePipeName);
open(PIPETOLOAD, '>', $outputPipeName) || die "$outputPipeName
kann nicht geöffnet werden";
# Anzahl der Datenträgerdateien ermitteln
#-----
$NumMediaFiles = $ARGV[1];
# Jede Datenträgerdatei öffnen, Inhalt lesen, '\t' durch ',' ersetzen an LOAD senden
#-----
for ($i=0; $i<$NumMediaFiles; $i++)
{
    # Multimediadatei öffnen
    #-----
    $mediaFileName = $ARGV[2+$i];
    open(MEDIAFILETOREAD, '<', $mediaFileName) || die "$mediaFileName
kann nicht geöffnet werden";
    # Jeden Datensatz lesen
    #-----
    while ( $line = <MEDIAFILETOREAD> )
    {
        # '\t' durch ',' ersetzen
        #-----
        $line =~ s/\t/,/g;
        # Diesen Datensatz zur Verarbeitung an LOAD senden
        #-----
        print PIPETOLOAD $line;
    }
    # Datenträgerdatei schließen
    #-----
    close MEDIAFILETOREAD;
}
# FIFO schließen
#-----
close PIPETOLOAD;
exit 0;
```

Weitere Aspekte des Ladens

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. 9). Eine weitere Möglichkeit ist der Einsatz mehrerer Prozessoren in einer SMP-Umgebung, die eine partitionsinterne Parallelität ermöglicht (siehe Abb. 10). Beide Möglichkeiten können kombiniert verwendet werden, um ein noch schnelleres Laden der Daten zu erreichen.

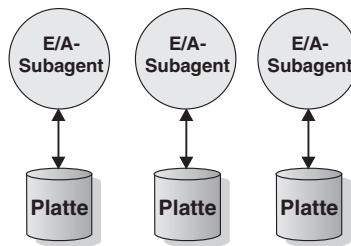


Abbildung 9. E/A-Parallelität beim Laden von Daten

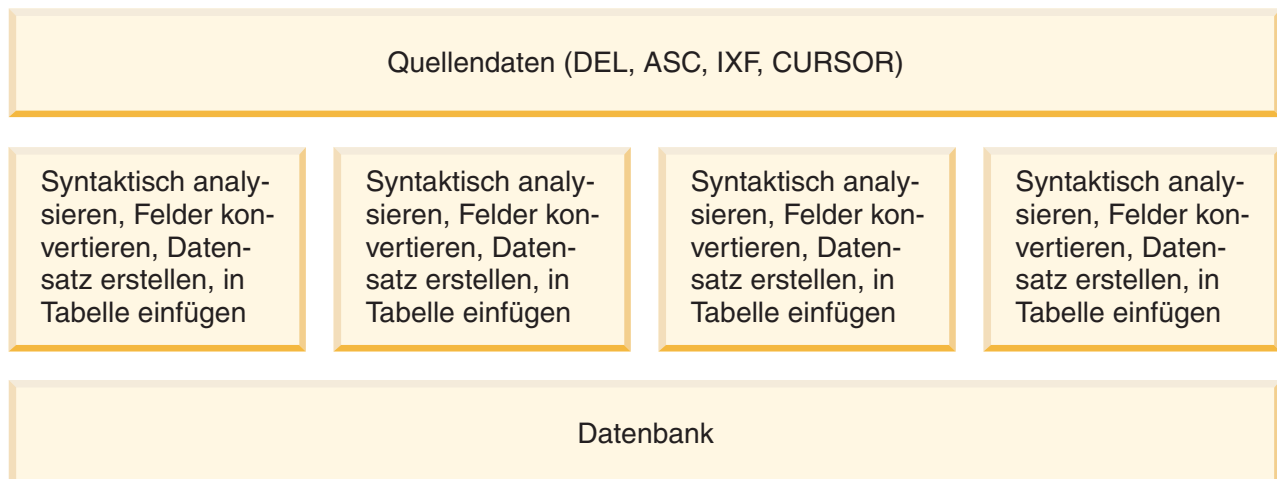


Abbildung 10. Partitionsinterne Parallelität beim Laden von Daten

Indexerstellung während einer Ladeoperation

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. AUTOSELECT ist der Standardwert. Wenn eine LOAD REPLACE-Operation ausgeführt wird, verwendet das System den Indexierungsmodus REBUILD. Andernfalls basiert der ausgewählte Indexie-

rungsmodus auf dem Verhältnis zwischen dem Volumen der in der Tabelle vorhandenen Daten und dem Volumen der neu geladenen Daten. Wenn dieses Verhältnis ausreicht, wird der Indexierungsmodus INCREMENTAL ausgewählt. Andernfalls verwendet das System den Indexierungsmodus REBUILD.

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. Die Option DEFERRED ist in folgenden Situationen nicht zulässig:
 - Wenn die Option ALLOW READ ACCESS angegeben wurde. (Bei dieser Option werden die Indizes nicht beibehalten, die Indexsuchfunktionen benötigen jedoch einen gültigen Index.)
 - Wenn für die Tabelle eindeutige Indizes definiert sind.
 - Wenn XML-Daten geladen werden (der XML-Pfadindex ist eindeutig und wird standardmäßig erstellt, wenn eine XML-Spalte zu einer Tabelle hinzugefügt wird).

Bei Ladeoperationen, für die die Option ALLOW READ ACCESS angegeben ist, 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 erneuten Erstellung für Abfragen zur Verfügung.

Gibt eine Ladeoperation im Modus ALLOW READ ACCESS die Option INDEXING MODE INCREMENTAL an, schreibt das Dienstprogramm LOAD einige Protokollsätze, die die Integrität der Indexbaumstruktur gewährleisten. 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 ALLOW NO ACCESS mit angegebener Option INDEXING MODE INCREMENTAL schreibt neben den üblichen Protokollen zur Speicherbereichszuordnung nur einen kleinen Protokollsatz.

Anmerkung: Dies gilt allerdings nur, wenn nicht COPY YES angegeben wurde und wenn für den Konfigurationsparameter *logindexrebuild* die Einstellung ON ausgewählt wurde.

Wenn eine Ladeoperation im Modus ALLOW READ ACCESS die Option INDEXING MODE REBUILD angibt, werden neue Indizes als *Schattenindizes* erstellt, und zwar entweder im selben Tabellenbereich wie der ursprüngliche Index oder in einem Tabellenbereich für temporäre Systemtabellen. 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 rückgängig gemacht wird, bleiben die ursprünglichen Indizes erhalten.

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

Wenn die Indizes in einem SMS-Tabellenbereich erstellt werden, können Sie Indexdateien 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. Werden die Indizes jedoch in einem DMS-Tabellenbereich erstellt, ist der neue Schattenindex nicht erkennbar.

Verbessern der Leistung bei der Indexerstellung

Erstellen von neuen Indizes in einem Tabellenbereich für temporäre Systemtabellen

Der neue Index kann in einem Tabellenbereich für temporäre Systemtabellen 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 Tabellenbereich für temporäre Systemtabellen erneut erstellt werden. Der Tabellenbereich für temporäre Systemtabellen kann ein SMS-Tabellenbereich oder ein DMS-Tabellenbereich sein. Die Seitengröße des Tabellenbereichs für temporäre Systemtabellen 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 `ALLOW READ ACCESS` ausgeführt wird oder wenn der Indexierungsmodus nicht kompatibel ist. Die Option `USE <tabellenbereichsname>` wird nur bei der Option `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 `ALLOW READ ACCESS` abgesetzt werden, wenn die ursprüngliche Ladeoperation nicht im Modus `ALLOW READ ACCESS` abgesetzt wurde. `LOAD TERMINATE`-Operationen führen keine Indexneuerstellung durch. Daher wird die Option `USE <tabellenbereichsname>` ignoriert.

Während der `BUILD`-Phase der Ladeoperation werden die Indizes im Tabellenbereich für temporäre Systemtabellen erstellt. Anschließend wird der Index während der `INDEX COPY`-Phase aus dem Tabellenbereich für temporäre Systemtabellen 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 erfolgt dies während der `BUILD`-Phase. 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 Tabellenbereich für temporäre Systemtabellen oder beim Tabellenbereich für den Index um einen DMS-Tabellenbereich handelt, kann das Lesen aus dem Tabellenbereich für temporäre Systemtabel-

len eine wahlfreie Ein-/Ausgabe für den Tabellenbereich für temporäre Systemtabellen bewirken und zu einer Verzögerung führen. Das Schreiben in den Tabellenbereich für den Index wird weiterhin optimiert, und die Werte DISK_PARALLELISM werden verwendet.

Aspekte zu großen Indizes

Zur Verbesserung der Leistung bei der Erstellung großer Indizes während einer Ladeoperation kann es sinnvoll sein, den Datenbankkonfigurationsparameter *sorthheap* zu optimieren. *sorthheap* ordnet die Speichermenge zu, die zum Sortieren der Indexschlüssel während einer Ladeoperation verwendet wird. Beispiel: Damit vom Dienstprogramm LOAD pro Index 4000 Hauptspeicherseiten zum Sortieren der Schlüssel verwendet werden, muss der Datenbankkonfigurationsparameter *sorthheap* auf 4000 Seiten gesetzt werden. Dann müssen alle Anwendungen von der Datenbank getrennt 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 Tabellenbereich für temporäre Tabellen gespeichert werden, der später zusammengeführt wird. Verwenden Sie das Monitorelement *sort_overflows*, um festzustellen, ob ein Sortierüberlauf stattgefunden hat. Besteht keine Möglichkeit, einen Sortierüberlauf durch ein Heraufsetzen des Parameters *sorthheap* zu verhindern, muss der Pufferpool für Tabellenbereiche für temporäre Tabellen unbedingt groß genug sein, um das Volumen der durch den Überlauf verursachten Platten-E/A zu minimieren. Es empfiehlt sich des Weiteren, Tabellenbereiche für temporäre Tabellen mit mehreren Containern zu deklarieren, die sich jeweils auf unterschiedlichen Platteneinheiten befinden, damit während der Zusammenführung der Sortierläufe eine E/A-Parallelität erzielt wird. Ist für eine Tabelle mehr als ein Index definiert, steigt die Speicherbelegung proportional an, da die Ladeoperation alle Schlüssel im Speicher hält.

Verzögern der Indexerstellung

Im Allgemeinen ist es effizienter, die Indexerstellung während der Ladeoperation zuzulassen, indem entweder der Modus REBUILD oder INCREMENTAL angegeben wird, anstatt die Indexerstellung zu verzögern. Wie in Abb. 11 auf Seite 180 dargestellt werden Tabellen normalerweise in drei Schritten erstellt: Laden von Daten, Erstellen von Indizes und Erfassen von 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 besteht darin, alle diese Aufgaben vom Dienstprogramm LOAD in einem Datendurchlauf ausführen zu lassen. Es sollte hierbei allerdings beachtet werden, dass eindeutige Indizes die Leistung von LOAD verringern, wenn Indizes doppelt vorhanden sind.

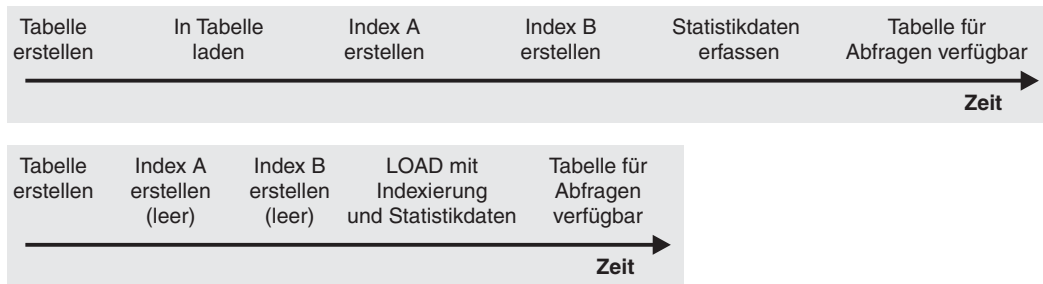


Abbildung 11. Erhöhen der Leistung von LOAD durch gleichzeitige Indexierung und Erfassung von Statistikdaten. Tabellen werden normalerweise in drei Schritten erstellt: Laden von Daten, Erstellen von Indizes und Erfassen Statistikdaten. Das führt zu mehrfacher Daten-E/A während der Ladeoperation, während 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.

In bestimmten Fällen kann die Verzögerung der Indexerstellung und der Aufruf der Anweisung CREATE INDEX zur Leistungsverbesserung beitragen. Bei der Sortierung während einer Indexneuerstellung werden maximal so viele Seiten verwendet, wie in *sortheap* angegeben wurden. Wird mehr Speicherplatz benötigt, wird der Pufferpool TEMP verwendet, und (schließlich) erfolgt ein Überlauf auf die Platte. Kommt es beim Laden zu einem Überlauf und somit zu einer Leistungsminderung, kann es ratsam sein, das Dienstprogramm **LOAD** mit der Option INDEXING MODE DEFERRED auszuführen und den Index später erneut zu erstellen. Mit CREATE INDEX wird jeweils immer nur ein Index erstellt. Dadurch wird beim mehrfachen Durchsuchen der Tabelle zum Erfassen von Schlüsseln die Speicherbelegung reduziert.

Ein weiterer Vorteil der Indexerstellung mit der Anweisung CREATE INDEX gegenüber der parallelen Erstellung mit einer Ladeoperation liegt darin, dass die Anweisung CREATE INDEX zum Sortieren von Schlüsseln mehrere Prozesse, so genannte Threads, verwenden kann. Das Erstellen des Indexes selbst wird nicht parallel ausgeführt.

Beheben von Indexierungsfehlern beim Laden von XML-Daten:

Probleme mit Ladeoperationen, die aufgrund von Indexierungsfehlern fehlschlagen, können behoben werden, indem die Protokolldatei **db2diag** zusammen mit dem Dienstprogramm IMPORT verwendet wird, um die Problemwerte in den XML-Daten zu identifizieren und zu korrigieren.

Informationen zu diesem Vorgang

Wenn eine Ladeoperation die Fehlermeldung SQL20305N (sqlcode -20305) zurückgibt, bedeutet dies, dass mindestens ein XML-Knotenwert nicht indiziert werden konnte. Die Fehlermeldung gibt den jeweiligen Ursachencode für den Fehler zurück. Geben Sie ? SQL20305N in den Befehlszeilenprozessor ein, um die Erläuterung und Benutzeraktion für den entsprechenden Ursachencode anzuzeigen.

Bei Indexierungsfehlern, die bei Einfügeoperationen auftreten, wird eine generierte XQuery-Anweisung als Ausgabe in die Protokolldatei **db2diag** geschrieben, um Sie beim Lokalisieren der fehlgeschlagenen XML-Knotenwerte im Dokument zu unterstützen. Ausführliche Informationen dazu, wie die XQuery-Anweisung verwendet wird, um die fehlerhaften XML-Knotenwerte zu lokalisieren, finden Sie in "Allgemeine Aspekte der XML-Indexierung".

Bei Indexierungsfehlern, die bei Ladeoperationen auftreten, werden die generierten XQuery-Anweisungen jedoch nicht in die Protokolldatei **db2diag** geschrieben. Um diese XQuery-Anweisungen zu generieren, muss das Dienstprogramm IMPORT für die fehlgeschlagenen, nicht geladenen Zeilen ausgeführt werden. Da die zurückgewiesenen Zeilen in der Tabelle nicht vorhanden sind, können die XQuery-Anweisungen nicht für die fehlgeschlagenen Dokumente ausgeführt werden. Um dieses Problem zu lösen, müssen Sie eine neue Tabelle mit derselben Definition ohne Indizes erstellen. Die fehlgeschlagenen Zeilen können dann in die neue Tabelle geladen werden, und die XQuery-Anweisungen können anschließend für die neue Tabelle ausgeführt werden, um die fehlerhaften XML-Knotenwerte in den Dokumenten zu lokalisieren.

Um die Indexierungsfehler zu beheben, führen Sie die folgenden Schritte aus:

Vorgehensweise

1. Ermitteln Sie anhand der Satznummern in den Ausgabeinformationen, welche Zeilen während der Ladeoperation zurückgewiesen wurden.
2. Erstellen Sie eine Datei mit der Erweiterung '.del', die ausschließlich die zurückgewiesenen Zeilen enthält.
3. Erstellen Sie eine neue Tabelle (beispielsweise T2) mit denselben Spalten wie in der Originaltabelle (T1). Erstellen Sie für die neue Tabelle keine Indizes.
4. Laden Sie die zurückgewiesenen Zeilen in die neue Tabelle (T2).
5. Führen Sie für jede zurückgewiesene Zeile in der Originaltabelle T1 Folgendes aus:
 - a. Importieren Sie die zurückgewiesenen Zeilen in T1, um die Fehlermeldung SQL20305N zu erhalten. Die Importoperation stoppt beim ersten Fehler, der auftritt.
 - b. Überprüfen Sie die Protokolldatei **db2diag** und suchen Sie die generierte XQuery-Anweisung. Um die fehlerhaften Knotenwerte im Eingabedokument zu ermitteln, suchen Sie nach der Zeichenfolge 'SQL20305N' in der Protokolldatei **db2diag** und vergleichen Sie den Ursachencode. Im Anschluss an den Ursachencode finden Sie eine Reihe von Anweisungen und danach eine generierte XQuery-Anweisung, die Sie verwenden können, um in dem Dokument denjenigen Problemwert ausfindig zu machen, der den Fehler verursacht hat.
 - c. Ändern Sie die XQuery-Anweisung entsprechend, um die neue Tabelle T2 zu verwenden.
 - d. Führen Sie die XQuery-Anweisung für T2 aus, um den Problemwert im Dokument zu lokalisieren.
 - e. Korrigieren Sie den Problemwert in der .xml-Datei, die das Dokument enthält.
 - f. Wiederholen Sie Schritt a, und importieren Sie die zurückgewiesenen Zeilen erneut in T1. Die Zeile, die zuvor zum Abbruch der Importoperation geführt hatte, sollte jetzt erfolgreich eingefügt werden. Liegt in der .del-Datei eine weitere zurückgewiesene Zeile vor, stoppt das Dienstprogramm IMPORT beim nächsten Fehler und gibt eine weitere Nachricht vom Typ SQL20305N aus. Wiederholen Sie die beschriebenen Schritte, bis die Importoperation erfolgreich ausgeführt wird.

Beispiel

In nachstehendem Beispiel wurde der Index 'BirthdateIndex' mit dem Datentyp *date* für Datumsangaben erstellt. Die Option REJECT INVALID VALUES ist angege-

ben, sodass sämtliche XML-Musterwerte von */Person/Confidential/Birthdate* für den Datentyp *date* gültig sein müssen. Sobald ein XML-Musterwert nicht in diesen Datentyp umgesetzt werden kann, wird ein Fehler zurückgegeben.

Bei Verwendung der nachstehenden XML-Dokumente sollen eigentlich fünf Zeilen geladen werden. Die erste und vierte Zeile werden jedoch zurückgewiesen, weil die Werte für das Geburtsdatum (*Birthdate*) nicht indexiert werden können. In der Datei *person1.xml* liegt der Wert *March 16, 2002* nicht im korrekten Datumsformat vor. In der Datei *person4.xml* weist der Wert *20000-12-09* in der Jahresangabe eine zusätzliche Null auf. So handelt es sich zwar um einen gültigen XML-Datumswert, der jedoch außerhalb des Bereichs liegt, der in DB2 für Jahresangaben zulässig ist (0001 bis 9999). Ein Teil der Beispielausgabe wurde bearbeitet, um das Beispiel kürzer zu gestalten.

Die folgenden fünf XML-Dateien sollen geladen werden:

person1.xml (mit ungültigem Wert für das Geburtsdatum 'Birthdate')

```
<?xml version="1.0"?>
<Person gender="Male">
  <Name>
    <Last>Cool</Last>
    <First>Joe</First>
  </Name>
  <Confidential>
    <Age unit="years">5</Age>
    <Birthdate>March 16, 2002</Birthdate>
    <SS>111-22-3333</SS>
  </Confidential>
  <Address>5224 Rose St. San Jose, CA 95123</Address>
</Person>
```

person2.xml (mit gültigem Wert für 'Birthdate')

```
<?xml version="1.0"?>
<Person gender="Male">
  <Name>
    <Last>Cool</Last>
    <First>Joe</First>
  </Name>
  <Confidential>
    <Age unit="years">5</Age>
    <Birthdate>2002-03-16</Birthdate>
    <SS>111-22-3333</SS>
  </Confidential>
  <Address>5224 Rose St. San Jose, CA 95123</Address>
</Person>
```

person3.xml (mit gültigem Wert für 'Birthdate')

```
<?xml version="1.0"?>
<Person gender="Female">
  <Name>
    <Last>McCarthy</Last>
    <First>Laura</First>
  </Name>
  <Confidential>
    <Age unit="years">6</Age>
    <Birthdate>2001-03-12</Birthdate>
    <SS>444-55-6666</SS>
  </Confidential>
  <Address>5960 Daffodil Lane, San Jose, CA 95120</Address>
</Person>
```

person4.xml (mit ungültigem Wert für 'Birthdate')

```
<?xml version="1.0"?>
<Person gender="Female">
  <Name>
    <Last>Wong</Last>
    <First>Teresa</First>
  </Name>
  <Confidential>
    <Age unit="years">7</Age>
    <Birthdate>20000-12-09</Birthdate>
    <SS>555-66-7777</SS>
  </Confidential>
  <Address>5960 Tulip Court, San Jose, CA 95120</Address>
</Person>
```

person5.xml (mit gültigem Wert für 'Birthdate')

```
<?xml version="1.0"?>
<Person gender="Male">
  <Name>
    <Last>Smith</Last>
    <First>Chris</First>
  </Name>
  <Confidential>
    <Age unit="years">10</Age>
    <Birthdate>1997-04-23</Birthdate>
    <SS>666-77-8888</SS>
  </Confidential>
  <Address>5960 Dahlia Street, San Jose, CA 95120</Address>
</Person>
```

Die Eingabedatei person.del enthält Folgendes:

```
1, <XDS FIL='person1.xml' />
2, <XDS FIL='person2.xml' />
3, <XDS FIL='person3.xml' />
4, <XDS FIL='person4.xml' />
5, <XDS FIL='person5.xml' />
```

Die DDL- und LOAD-Anweisungen sehen wie folgt aus:

```
CREATE TABLE T1 (docID INT, XMLDoc XML);

CREATE INDEX BirthdateIndex ON T1(xmlDoc)
  GENERATE KEY USING XMLPATTERN '/Person/Confidential/Birthdate' AS SQL DATE
  REJECT INVALID VALUES;

LOAD FROM person.del OF DEL INSERT INTO T1
```

Um die Indexierungsfehler zu beheben, die bei dem Versuch auftreten würden, die vorstehende XML-Dateigruppe zu laden, müssten Sie folgende Schritte ausführen:

1. Ermitteln Sie anhand der Satznummern in den Ausgabeinformationen, welche Zeilen während der Ladeoperation zurückgewiesen wurden. Aus der folgenden Ausgabe geht hervor, dass der Satz mit der Nummer 1 und der Satz mit der Nummer 4 zurückgewiesen wurden:

```
SQL20305N Ein XML-Wert kann nicht
eingefügt oder aktualisiert werden, weil beim Einfügen in den Index bzw.
beim Aktualisieren des Index, der durch "IID = 3" in Tabelle
"LEECM.T1" angegeben ist, ein Fehler festgestellt wurde. Ursachencode = "5". Bei
Ursachencodes für ein XML-Schema ist "*N" die XML-Schemakennung und "*N" der
XML-Schemadatentyp. SQLSTATE=23525
```

```
SQL3185W Der vorherige Fehler trat bei der Verarbeitung von Daten aus Zeile "F0-1"
der Eingabedatei auf.
```

SQL20305N Ein XML-Wert kann nicht eingefügt oder aktualisiert werden, weil beim Einfügen in den Index bzw. beim Aktualisieren des Index, der durch "IID = 3" in Tabelle "LEECM.T1" angegeben ist, ein Fehler festgestellt wurde. Ursachencode = "4". Bei Ursachencodes für ein XML-Schema ist "*N" die XML-Schemakennung und "*N" der XML-Schemadatentyp.
SQLSTATE=23525

SQL3185W Der vorherige Fehler trat bei der Verarbeitung von Daten aus Zeile "F0-4" der Eingabedatei auf.

SQL3227W Satztoken "F0-1" bezieht sich auf Benutzersatznummer "1".

SQL3227W Satztoken "F0-4" bezieht sich auf Benutzersatznummer "4".

SQL3107W Die Nachrichtendatei enthält mindestens eine Warnung.

Anzahl gelesener Zeilen = 5
Anzahl übersprungener Zeilen = 0
Anzahl geladener Zeilen = 3
Anzahl zurückgewiesener Zeilen = 2
Anzahl gelöschter Zeilen = 0
Anzahl festgeschriebener Zeilen = 5

- Erstellen Sie eine neue Datei namens `reject.del` mit den zurückgewiesenen Zeilen:

```
1, <XDS FIL='person1.xml' />  
4, <XDS FIL='person4.xml' />
```

- Erstellen Sie eine neue Tabelle (T2) mit denselben Spalten wie in der Originaltabelle T1. Erstellen Sie für die neue Tabelle keine Indizes.

```
CREATE TABLE T2 LIKE T1
```

- Laden Sie die zurückgewiesenen Zeilen in die neue Tabelle (T2).

```
LOAD FROM reject.del OF DEL INSERT INTO T2;
```

- Führen Sie für die erste zurückgewiesene Zeile in der Originaltabelle T1 Folgendes aus:

- Importieren Sie die zurückgewiesenen Zeilen in T1, um die Nachricht -20305 zu erhalten:

```
IMPORT FROM reject.del OF DEL INSERT INTO T1  
SQL3109N Das Dienstprogramm beginnt mit dem Laden von Daten aus der  
Datei "reject.del".
```

SQL3306N SQL-Fehler "-20305" beim Einfügen einer Zeile in die Tabelle.

SQL20305N Ein XML-Wert kann nicht eingefügt oder aktualisiert werden, weil beim Einfügen in den Index bzw. beim Aktualisieren des Index, der durch "IID = 3" in Tabelle "LEECM.T1" angegeben ist, ein Fehler festgestellt wurde. Ursachencode = "5". Bei Ursachencodes für ein XML-Schema ist "*N" die XML-Schemakennung und "*N" der XML-Schemadatentyp.
SQLSTATE=23525

SQL3110N Die Verarbeitung durch das Dienstprogramm ist abgeschlossen. "1" Zeilen aus der Eingabedatei wurden gelesen.

- Überprüfen Sie die Protokolldatei **db2diag** und suchen Sie die generierte XQuery-Anweisung.

```
FUNCTION: DB2 UDB, Xml Storage and Index Manager, xmlsDumpXQuery, probe:608  
DATA #1 : String, 36 bytes  
SQL Code: SQL20305N ; Reason Code: 5  
DATA #2 : String, 265 bytes  
To locate the value in the document that caused the error, create a  
table with one XML column and insert the failing document in the table.  
Replace the table and column name in the query below with the created
```



```

table and column name and execute the following XQuery.
DATA #3 : String, 247 bytes
xquery for $i in db2-fn:xmlcolumn(
  "LEECM.T1.XMLDOC") [/*:Person/*:Confidential/*:Birthdate="March 16, 2002"]
return
<Result>
  <ProblemDocument> {$i} </ProblemDocument>
  <ProblemValue>{$i/*:Person/*:Confidential/*:Birthdate/..} </ProblemValue>
</Result>;

```

- c. Ändern Sie die XQuery-Anweisung entsprechend, um die neue Tabelle T2 zu verwenden.

```

xquery for $i in db2-fn:xmlcolumn(
  "LEECM.T2.XMLDOC") [/*:Person/*:Confidential/*:Birthdate="March 16, 2002"]
return
<Result>
  <ProblemDocument> {$i} </ProblemDocument>
  <ProblemValue>{$i/*:Person/*:Confidential/*:Birthdate/..} </ProblemValue>
</Result>;

```

- d. Führen Sie die XQuery-Anweisung für Tabelle T2 aus, um den Problemwert im Dokument zu lokalisieren:

```

<Result><ProblemDocument><Person gender="Male">
  <Name>
    <Last>Cool</Last>
    <First>Joe</First>
  </Name>
  <Confidential>
    <Age unit="years">5</Age>
    <Birthdate>March 16, 2002</Birthdate>
    <SS>111-22-3333</SS>
  </Confidential>
  <Address>5224 Rose St. San Jose, CA 95123</Address>
</Person></ProblemDocument><ProblemValue><Confidential>
  <Age unit="years">5</Age>
  <Birthdate>March 16, 2002</Birthdate>
  <SS>111-22-3333</SS>
</Confidential></ProblemValue></Result>

```

- e. Korrigieren Sie den Problemwert in der Datei person1.xml, die das Dokument enthält. Der Wert March 16, 2002 liegt nicht im korrekten Datumsformat vor und muss daher in den Wert 2002-03-16 geändert werden:

```

<?xml version="1.0"?>
<Person gender="Male">
  <Name>
    <Last>Cool</Last>
    <First>Joe</First>
  </Name>
  <Confidential>
    <Age unit="years">5</Age>
    <Birthdate>2002-03-16</Birthdate>
    <SS>111-22-3333</SS>
  </Confidential>
  <Address>5224 Rose St. San Jose, CA 95123</Address>
</Person>

```

- f. Wiederholen Sie Schritt a., um die zurückgewiesenen Zeilen erneut in Tabelle T1 zu importieren.

6. (Erste Wiederholung von Schritt 5)

- a. Importieren Sie die zurückgewiesenen Zeilen in Tabelle T1. Die erste Zeile wurde nun erfolgreich importiert, da zwei Zeilen aus der Importdatei gelesen wurden. Ein neuer Fehler tritt bei der zweiten Zeile auf:

```

IMPORT FROM reject.del OF DEL INSERT INTO T1
SQL3109N Das Dienstprogramm beginnt mit dem Laden von Daten aus der
Datei "reject.del".

```

SQL3306N SQL-Fehler "-20305" beim Einfügen einer Zeile in die Tabelle.

SQL20305N Ein XML-Wert kann nicht eingefügt oder aktualisiert werden, weil beim Einfügen in den Index bzw. beim Aktualisieren des Index, der durch "IID = 3" in Tabelle "LEECM.T1" angegeben ist, ein Fehler festgestellt wurde. Ursachencode = "4". Bei Ursachencodes für ein XML-Schema ist "*N" die XML-Schemakennung und "*N" der XML-Schemadatentyp.
SQLSTATE=23525

SQL3110N Die Verarbeitung durch das Dienstprogramm ist abgeschlossen.
"2" Zeilen aus der Eingabedatei wurden gelesen.

- b. Überprüfen Sie die Protokolldatei **db2diag** und suchen Sie die generierte XQuery-Anweisung.

```
FUNCTION: DB2 UDB, Xml Storage and Index Manager, xmlsDumpXQuery, probe:608
DATA #1 : String, 36 bytes
SQL Code: SQL20305N ; Reason Code: 4
DATA #2 : String, 265 bytes
To locate the value in the document that caused the error, create a
table with one XML column and insert the failing document in the table.
Replace the table and column name in the query below with the created
table and column name and execute the following XQuery.
DATA #3 : String, 244 bytes
xquery for $i in db2-fn:xmlcolumn("LEECM.T1.XMLDOC")
  [/*:Person/*:Confidential/*:Birthdate="20000-12-09"]
return
<Result>
  <ProblemDocument> {$i} </ProblemDocument>
  <ProblemValue>{$i/*:Person/*:Confidential/*:Birthdate/..} </ProblemValue>
</Result>;
```

- c. Ändern Sie die XQuery-Anweisung entsprechend, um Tabelle T2 zu verwenden:

```
xquery for $i in db2-fn:xmlcolumn("LEECM.T2.XMLDOC")
  [/*:Person/*:Confidential/*:Birthdate="20000-12-09"]
return
<Result>
  <ProblemDocument> {$i} </ProblemDocument>
  <ProblemValue>{$i/*:Person/*:Confidential/*:Birthdate/..} </ProblemValue>
</Result>;
```

- d. Führen Sie die XQuery-Anweisung aus, um den Problemwert im Dokument zu lokalisieren:

```
<Result><ProblemDocument><Person gender="Female">
  <Name>
    <Last>Wong</Last>
    <First>Teresa</First>
  </Name>
  <Confidential>
    <Age unit="years">7</Age>
    <Birthdate>20000-12-09</Birthdate>
    <SS>555-66-7777</SS>
  </Confidential>
  <Address>5960 Tulip Court, San Jose, CA 95120</Address>
</Person></ProblemDocument><ProblemValue><Confidential>
  <Age unit="years">7</Age>
  <Birthdate>20000-12-09</Birthdate>
  <SS>555-66-7777</SS>
</Confidential></ProblemValue></Result>
```

- e. Korrigieren Sie den Problemwert in der Datei person4.xml, die das Dokument enthält. Der Wert 20000-12-09 weist in der Jahresangabe eine zusätzli-

che Null auf, sodass die Jahresangabe außerhalb des Bereichs liegt, der in DB2 für Jahresangaben zulässig ist (0001 bis 9999). Der Wert wird in 2000-12-09 geändert:

```
<?xml version="1.0"?>
<Person gender="Female">
  <Name>
    <Last>Wong</Last>
    <First>Teresa</First>
  </Name>
  <Confidential>
    <Age unit="years">7</Age>
    <Birthdate>2000-12-09</Birthdate>
    <SS>555-66-7777</SS>
  </Confidential>
  <Address>5960 Tulip Court, San Jose, CA 95120</Address>
</Person>
```

- f. Wiederholen Sie Schritt a, um die zurückgewiesenen Zeilen erneut in Tabelle T1 zu importieren.
7. (Zweite Wiederholung von Schritt 5)
- a. Importieren Sie die zurückgewiesenen Zeilen in Tabelle T1:

```
IMPORT FROM reject.del OF DEL INSERT INTO T1
SQL3109N Das Dienstprogramm beginnt mit dem Laden von Daten aus der
Datei "reject.del".

SQL3110N Die Verarbeitung durch das Dienstprogramm ist abgeschlossen.
"2" Zeilen aus der Eingabedatei.

SQL3221W ...COMMIT WORK beginnt. Eingabesatzzähler = "2".

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

SQL3149N "2" Zeilen aus der Eingabedatei wurden verarbeitet. "2" Zeile(n) wurden
ohne Fehler der Tabelle hinzugefügt. "0" Zeile(n) wurden zurückgewiesen.

Anzahl gelesener Zeilen           = 2
Anzahl übersprungener Zeilen     = 0
Anzahl eingefügter Zeilen        = 2
Anzahl aktualisierter Zeilen     = 0
Anzahl zurückgewiesener Zeilen   = 0
Anzahl festgeschriebener Zeilen  = 2
```

Das Problem ist nun behoben. Alle Zeilen aus der Datei person.del wurden erfolgreich in Tabelle T1 eingefügt.

Erstellen eines Komprimierungswörterverzeichnisses (Compression Dictionary) während einer Ladeoperation

Die Operationen **LOAD INSERT** und **LOAD REPLACE** lösen, wenn Sie bestimmte Bedingungen erfüllen, die automatische Wörterverzeichniserstellung (Automatic Dictionary Creation - ADC) aus. Sobald ausreichend Daten verarbeitet wurden, findet die ADC statt, wenn die Ladeoperation für eine Tabelle ausgeführt wird, bei der das Attribut **COMPRESS** aktiviert und kein Komprimierungswörterverzeichnis (Compression Dictionary) vorhanden ist.

Die Datenzeilenkomprimierung verwendet für die Komprimierung von Daten einen statischen wörterverzeichnisbasierten Komprimierungsalgorithmus. Es werden bis zu zwei separate Wörterverzeichnisse verwendet - eines für die Komprimierung der Tabellenzeilen und ein weiteres für die Komprimierung der im standardmäßigen XML-Speicherobjekt gespeicherten XML-Dokumente, sofern die Tabelle mindestens eine XML-Spalte enthält. Damit eine Komprimierung stattfinden kann, muss zunächst ein Wörterverzeichnis in der Tabelle vorhanden sein. Während einer Ladeoperation sieht das Standardverhalten (angegeben durch die Option **KEEP-**

DICTIONARY) so aus, dass entweder die vorhandenen Wörterverzeichnisse verwendet werden oder - falls keine Wörterverzeichnisse vorhanden sind - dass bei Erreichen einer bestimmten Schwelle an durchsuchten Daten die Wörterverzeichnisse generiert werden. Bitte beachten Sie, dass die Wörterverzeichnisse unabhängig voneinander erstellt werden, sobald die für sie jeweils geltenden Schwellenwerte überschritten werden. Dies geschieht normalerweise nicht gleichzeitig.

Bei Nicht-XML-Daten verwendet das Dienstprogramm LOAD die in der Zieltabelle vorhandenen Daten zum Erstellen der Wörterverzeichnisse, sofern diese bereits vorhandenen Daten repräsentativ für die Art von Daten sind, die in der Tabelle gespeichert werden sollen. In den Fällen, in denen in der Zieltabelle nicht genug Daten vorhanden sind, erstellt das Dienstprogramm LOAD die Wörterverzeichnisse, nachdem es ausreichend Eingabedaten geprüft hat. Vor DB2 Version 9.7 Fixpack 1 verwendete das Dienstprogramm LOAD sowohl die Eingabedaten als auch die vorhandenen Daten, um in dieser Situation die Wörterverzeichnisse zu erstellen. Ab Version 9.7 Fixpack 1 verwendet das Dienstprogramm LOAD nur die Eingabedaten zur Erstellung des Wörterverzeichnisses.

Bei XML-Daten prüft das Dienstprogramm LOAD lediglich die ankommenden Daten.

Wenn ADC bei bereichspartitionierten Tabellen erfolgt, wird jede Partition wie eine einzelne Tabelle behandelt. Es werden keine partitionsübergreifenden Wörterverzeichnisse erstellt und ADC erfolgt nicht in Partitionen, die bereits Wörterverzeichnisse enthalten. Bei Tabellendaten basiert das Wörterverzeichnis, das für jede Partition erstellt wird, ausschließlich auf den in der betreffenden Partition bereits vorhandenen Tabellendaten (und gegebenenfalls den geladenen Daten). Wenn die in einer Partition bereits vorhandenen Daten unter dem Mindestschwellenwert liegen, wird ab Version 9.7 Fixpack 1 das Wörterverzeichnis ausschließlich auf der Basis der geladenen Daten generiert. Bei XML-Daten basiert das für jede Partition erstellte Wörterverzeichnis auf den Daten, die in die betreffende Partition geladen werden.

Etwaige Ladeoperationen, die im Modus **INSERT** ausgeführt werden, folgen implizit dem Verhalten von **KEEPDICTIONARY**. Bei **LOAD REPLACE**-Operationen ist dies ebenfalls das Standardverhalten. Als Alternative steht zusätzlich jedoch noch die Option **RESETDICTIONARY** zur Auswahl.

LOAD REPLACE mit der Option KEEPDICTIONARY

Eine **LOAD REPLACE**-Operation, die die Option **KEEPDICTIONARY** verwendet, behält die vorhandenen Wörterverzeichnisse bei und verwendet sie zur Komprimierung der geladenen Daten, sofern für die Zieltabelle das Attribut **COMPRESS** aktiviert ist. Wenn keine Wörterverzeichnisse vorhanden sind, generiert das Dienstprogramm LOAD für Tabellen mit aktiviertem Attribut **COMPRESS** neue Wörterverzeichnisse (unter der Voraussetzung, dass die in die Tabelle geladenen Daten einen zuvor festgelegten Schwellenwert für Tabellenzeilen oder für XML-Dokumente im standardmäßigen XML-Speicherobjekt überschreiten). Da die Daten der Zieltabelle ersetzt werden, verwendet das Dienstprogramm LOAD zum Erstellen der Wörterverzeichnisse lediglich die Eingabedaten. Nach Erstellung eines Wörterverzeichnisses wird dieses in die Tabelle eingefügt, und die Ladeoperation wird fortgesetzt.

LOAD REPLACE mit der Option RESETDICTIONARY

Es sind zwei Schlüsselauswirkungen für die Verwendung der Option **RESETDICTIONARY** zu erwarten, wenn Daten in eine Tabelle geladen werden, deren **COMPRESS**-Attribut aktiviert ist. Erstens findet die Erstellung von Datenverzeichnissen statt, solange ein beliebiges Datenvolumen in der Zieltabelle vorhanden ist,

nachdem die **LOAD REPLACE**-Operation abgeschlossen ist. Mit anderen Worten: Die neuen Komprimierungswörterverzeichnisse können auch auf einer einzigen Zeile von Daten oder einem einzigen XML-Dokument beruhen. Zweitens werden die vorhandenen Wörterverzeichnisse gelöscht aber nicht ersetzt (die Zieltabelle verfügt also über keine Komprimierungswörterverzeichnisse mehr), falls eine der folgenden Situationen zutrifft:

- Die Operation wird für eine Tabelle mit inaktiviertem Attribut **COMPRESS** ausgeführt
- Es wurde nichts geladen (Nullzeilen), was bedeutet, dass ADM5591W im Benachrichtigungsprotokoll erfasst wird

Anmerkung: Wenn Sie eine **LOAD TERMINATE**-Operation nach einer **LOAD REPLACE**-Operation mit der Option **RESETDICTIONARY** durchführen, werden eventuell vorhandene Komprimierungswörterverzeichnisse gelöscht und nicht ersetzt.

Leistungseinfluss

Das Erstellen von Wörterverzeichnissen wirkt sich auf folgende Weise auf die Leistung einer Ladeoperation aus:

- Bei **LOAD INSERT**-Operationen werden sämtliche bereits vorhandenen Tabellendaten vor der Erstellung des Komprimierungswörterverzeichnis geprüft, nicht nur der Mindestschwellenwert für ADC. Daher steigt die für diese Überprüfung benötigte Zeit mit der Tabellengröße an. Diese Auswirkung gilt nicht für XML-Wörterverzeichnisse. Ab DB2 Version 9.7 Fixpack 1 besteht diese Auswirkung für keines der Wörterverzeichnisse mehr.
- Es fällt zusätzlicher Verarbeitungsaufwand für die Erstellung der Komprimierungswörterverzeichnisse an. Die Zeit, die tatsächlich zum Erstellen der Wörterverzeichnisse benötigt wird, ist minimal. Außerdem wird die ADC standardmäßig inaktiviert, sobald die Wörterverzeichnisse erstellt worden sind.

Optionen zur Verbesserung der Ladeleistung

Es gibt verschiedene Befehlsparameter, die Sie zum Optimieren der Ladeleistung verwenden können. Darüber hinaus steht eine Reihe von Änderungswerten für Dateitypen zur Verfügung, die speziell für das Dienstprogramm **LOAD** konzipiert sind und in bestimmten Fällen die Leistung dieses Dienstprogramms erheblich steigern können.

Befehlsparameter

Das Dienstprogramm **LOAD** 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. Bevor Sie versuchen, die Werte dieser Parameter für Ihre Anforderungen zu optimieren, sollten Sie prüfen, ob nicht die autonomen Einstellungen von **DISK_PARALLELISM** und **CPU_PARALLELISM** für Ihr System gut geeignet sind.

Die folgenden Informationen beziehen sich auf die Auswirkungen der verschiedenen, beim Dienstprogramm **LOAD** verfügbaren Optionen auf die Leistung:

ALLOW READ ACCESS

Mithilfe dieser Option können Sie eine Tabelle abfragen, während eine Ladeoperation ausgeführt wird. Sie können nur die Daten anzeigen, die vor der Ladeoperation bereits 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 ist eine Indexsuche erforderlich, die mit einem erheblichen Aufwand an E/A-Operationen verbunden ist. Wird die Option ALLOW READ ACCESS auch für die LOAD TERMINATE-Operation angegeben, wird für die Ein-/Ausgabe der Pufferpool verwendet.

COPY YES oder NO

Verwenden Sie diesen Parameter, um anzugeben, ob während einer Ladeoperation eine Kopie der Eingabedaten erstellt werden soll. COPY YES kann nur verwendet werden, wenn die aktualisierende Recovery aktiviert wurde. Diese Option verringert die Leistung von LOAD, weil alle Ladedaten während der Ladeoperation kopiert werden. Der Anstieg der E/A-Aktivität 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 nur verwendet werden, wenn die aktualisierende Recovery aktiviert wurde. Diese Option hat keine Auswirkungen auf die Leistung von LOAD. Allerdings werden alle Tabellenbereiche, die der geladenen Tabelle zugeordnet sind, in den Status "Backup anstehend" versetzt und müssen gesichert werden, bevor auf die Tabelle zugegriffen werden kann.

CPU_PARALLELISM

Mit diesem Parameter können Sie die Anzahl der pro Datenbankpartition ausgeführten Prozesse ausschöpfen (wenn dies 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. 12) bleibt unabhängig vom Wert dieses Parameters erhalten, wenn Folgendes gilt:

- Der Änderungswert anyorder für den Datentyp ist nicht angegeben.
- Die Option PARTITIONING_DBPARTNUMS ist nicht angegeben (und für die Partitionierung werden mehrere Partitionen verwendet).

Wenn Tabellen Daten des Typs LOB oder LONG VARCHAR enthalten, wird CPU_PARALLELISM auf den Wert 1 gesetzt. In diesem Fall wird die 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 12. Beibehaltung der Satzreihenfolge der Quelldaten bei Ausnutzung der Anzahl von pro Datenbankpartition ausgeführten Prozessen bei einer Ladeoperation

DATA BUFFER

Mit dem Parameter DATA BUFFER wird die Gesamtmenge des Speichers (in Einheiten von jeweils 4 KB) angegeben, die dem Dienstprogramm LOAD

als Puffer zugeordnet wird. Es wird empfohlen, die Größe dieses Puffers auf mehrere *Speicherbereiche* (Extents) einzustellen. Der Datenpuffer wird aus dem Zwischenspeicher der Dienstprogramme zugeordnet; er kann jedoch die Einstellung für den Datenbankkonfigurationsparameter *util_heap_sz* überschreiten, solange im System ausreichend Speicher verfügbar ist.

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 Container beim Laden von Daten ausschöpfen und die Leistung von LOAD signifikant verbessern. Der maximal zulässige Wert entspricht entweder dem Vierfachen des Wertes von CPU_PARALLELISM (der vom Dienstprogramm LOAD tatsächlich verwendet wird) oder beträgt 50. Der höhere dieser beiden Werte wird verwendet. DISK_PARALLELISM entspricht standardmäßig der Summe der Tabellenbereichscontainer in allen Tabellenbereichen, die Objekte für die zu ladende Tabelle enthalten, außer wenn dieser Wert den maximal zulässigen Wert übersteigt.

NONRECOVERABLE

Wenn die aktualisierende Recovery aktiviert wurde, verwenden Sie diesen Parameter, wenn LOAD-Transaktionen für eine Tabelle nicht wiederherstellbar sein müssen, wenn eine aktualisierende Recovery durchgeführt wird. Eine Ladeoperation mit NONRECOVERABLE und mit COPY NO weisen die gleiche Leistung auf. Allerdings besteht ein erheblicher Unterschied in Bezug auf das Risiko eines möglichen Datenverlustes. Eine Ladeoperation mit NONRECOVERABLE markiert eine Tabelle als mit einer aktualisierenden Recovery nicht wiederherstellbar, der Zugriff auf die Tabelle wird jedoch nicht eingeschränkt. Dies kann zu Problemen führen, wenn eine aktualisierende Recovery der Ladeoperation ausgeführt werden muss. In diesem Fall gehen sowohl die geladenen Daten als auch alle nachfolgend durchgeführten Aktualisierungen an der Tabelle verloren. Eine mit COPY NO durchgeführte Ladeoperation versetzt alle abhängigen Tabellenbereiche in den Status "Backup anstehend", durch den nicht mehr auf die Tabelle zugegriffen werden kann, bis der Backup durchgeführt wurde. Da nach Ausführung dieses Ladeoperationstyps ein Backup erstellt werden muss, riskieren Sie nicht den Verlust der geladenen Daten oder der nachfolgenden Aktualisierungen an der Tabelle. Dies bedeutet also, dass eine Ladeoperation mit COPY NO vollständig wiederherstellbar ist.

Anmerkung: Wenn diese LOAD-Transaktionen bei nachfolgenden Restores und aktualisierenden Recoverys vorgefunden werden, wird die Tabelle nicht aktualisiert und als ungültig (*invalid*) gekennzeichnet. Weitere Aktionen für diese Tabelle werden ignoriert. Nach Fertigstellung der aktualisierenden Recovery kann die Tabelle nur gelöscht werden.

SAVECOUNT

Verwenden Sie diesen Parameter, um ein Intervall für die Festlegung von Konsistenzpunkten **während der LOAD-Phase** einer Ladeoperation einzustellen. Die Synchronisation der Vorgänge zum Definieren eines Konsistenzpunkts 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 hoher 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 Konsistenzpunkt aus fortgesetzt. Dies gilt allerdings nur dann, wenn die LOAD RESTART-Operation bei der LOAD-Phase wieder aufgenommen wird.

STATISTICS USE PROFILE

Erfasst Statistiken, die im Tabellenstatistikprofil angegeben sind. Mit diesem Parameter können Sie Datenverteilungs- und Indexstatistikdaten effizienter erfassen als durch den Aufruf des Dienstprogramms RUNSTATS nach Fertigstellung der Ladeoperation. Die Leistung der Ladeoperation selbst wird bei Verwendung dieses Parameters allerdings verringert. (Dies gilt insbesondere dann, wenn DETAILED INDEXES ALL angegeben ist.)

Die Leistungsstärke der Anwendungen ist am höchsten, wenn sie die bestmöglichen Verteilungs- und Indexstatistikdaten 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 Befehl **BIND** erneut binden. Das Tabellenstatistikprofil wird erstellt, indem der Befehl **RUNSTATS** mit den SET PROFILE-Optionen ausgeführt wird.

Beim Laden von Daten in große Tabellen empfiehlt es sich, für den Datenbankkonfigurationsparameter *stat_heap_sz* (Größe des StatistikzwischenSpeichers) einen höheren Wert anzugeben.

USE <tabellenbereichsname>

Bei Ausführung einer Ladeoperation mit ALLOW READ ACCESS, für die der Indexierungsmodus REBUILD verwendet wird, erlaubt dieser Parameter die erneute Erstellung eines Indexes in einem Tabellenbereich für temporäre Systemtabellen und das Zurückkopieren dieses Indexes in den Indextabellenbereich während der INDEX COPY-Phase einer Ladeoperation.

Standardmäßig wird der vollständig neu erstellte Index (der auch als *Schattenindex* bezeichnet wird) im selben 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 in demselben Tabellenbereich wie der ursprüngliche Index erstellt wird, wird der ursprüngliche Index sofort durch den Schattenindex ersetzt. Wird der Schattenindex jedoch in einem Tabellenbereich für temporäre Systemtabellen erstellt, ist bei der Ladeoperation eine INDEX COPY-Phase erforderlich, die den Index aus einem Tabellenbereich für temporäre Systemtabellen 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 Tabellenbereich für temporäre Systemtabellen möglicherweise nicht sequenziell erfolgen. Während der INDEX COPY-Phase werden die in der 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. Geben Sie für den Parameter WARNINGCOUNT einen relativ niedrigen numerischen Wert an, wenn Sie lediglich einige wenige oder gar keine Warnungen erwarten. Die Ladeoperation wird nach Erreichen des Wertes für WARNINGCOUNT gestoppt. Damit haben Sie die Möglichkeit, die Daten zu korrigieren, bevor Sie versuchen, die Ladeoperation abzuschließen.

Änderungswerte für Dateityp

ANYORDER

Standardmäßig behält das Dienstprogramm LOAD die Satzreihenfolge der Quelldaten bei. Wenn das Dienstprogramm LOAD in einer SMP-Umgebung eingesetzt wird, dann müssen die Parallelverarbeitungsoperationen synchronisiert werden, um die Beibehaltung der korrekten Reihenfolge zu gewährleisten.

In einer SMP-Umgebung wird das Dienstprogramm LOAD mit dem Änderungswert `anyorder` für den Dateityp angewiesen, die Reihenfolge nicht beizubehalten. Hierdurch lässt sich die Effizienz verbessern, da die zur Beibehaltung der Reihenfolge erforderliche Synchronisation vermieden werden kann. Wenn die zu ladenden Daten jedoch bereits vorsortiert sind, beschädigt die Angabe von `anyorder` möglicherweise die Vorsortierung, und die Vorteile der Vorsortierung gehen für nachfolgende Abfragen verloren.

Anmerkung: Der Änderungswert `anyorder` für den Dateityp ist ohne Wirkung, wenn für `CPU_PARALLELISM` der Wert 1 definiert ist. Außerdem besteht keine Kompatibilität mit der Option `SAVECOUNT`.

BINARYNUMERICS, ZONEDDECIMAL und PACKEDDECIMAL

Für ASCII-Quelldaten mit fester Länge und mit universellen Zeilenbegrenzern (ASC) kann es bei der Darstellung numerischer Daten im Binärformat bei der Ausführung von Ladeoperationen zu einer Verbesserung der Leistung kommen. Wenn der Änderungswert `packeddecimal` für den Dateityp angegeben wird, werden Dezimaldaten vom Dienstprogramm LOAD als im gepackten Dezimalformat (zwei Stellen pro Byte) vorliegend interpretiert. Wenn der Änderungswert `zoneddecimal` für den Dateityp angegeben ist, werden Dezimaldaten vom Dienstprogramm LOAD als im gezonten Dezimalformat (eine Stellen pro Byte) vorliegend interpretiert. Bei allen anderen numerischen Typen werden die Daten bei Angabe des Änderungswertes `binarynumerics` für den Dateityp vom Dienstprogramm LOAD als im Binärformat vorliegend interpretiert.

Anmerkung:

- Wenn der Änderungswert `binarynumerics`, `packeddecimal` oder `zoneddecimal` für den Dateityp angegeben ist, werden numerische Daten unabhängig von der verwendeten Plattform als im Big Endian-Format (höherwertiges Byte zuerst) vorliegend interpretiert.
- Der Änderungswert `packeddecimal` für den Dateityp und der Änderungswert `zoneddecimal` für den Dateityp schließen sich gegenseitig aus.
- Der Änderungswert `packeddecimal` für den Dateityp und der Änderungswert `zoneddecimal` für den Dateityp gelten nur für die dezimalen Zielspalten. Die binären Daten müssen hierbei mit den Definitionen der Zielspalten übereinstimmen.
- Der Änderungswert `reclen` für den Dateityp muss angegeben werden, wenn der Änderungswert `binarynumerics`, `packeddecimal` oder `zoneddecimal` für den Dateityp angegeben ist.

FASTPARSE

Diese Werte sollten mit Bedacht verwendet werden. In bestimmten Situationen, in denen bekannt ist, dass die zu ladenden Daten gültig sind, ist es eventuell nicht erforderlich, dass das Dienstprogramm LOAD die Syntaxprüfung im selben Umfang durchführt wie bei Daten, die als fehlerverdächtiger eingestuft werden. Durch eine Reduzierung des Geltungsbereichs dieses Schrittes kann die Leistung des Dienstprogramms LOAD um ca. 10 - 20 Prozent gesteigert werden. Hierzu kann der Än-

derungswert `fastparse` für den Dateityp angegeben werden, mit dem der Aufwand für die Datenprüfung in vom Benutzer angegebenen Spaltenwerten aus ASC- und DEL-Dateien reduziert werden kann.

NOROWWARNINGS

Während der Ausführung einer Ladeoperation werden Warnungen zu zurückgewiesenen Zeilen in eine angegebene Datei geschrieben. Wenn vom Dienstprogramm LOAD jedoch eine große Anzahl zurückgewiesener, ungültiger oder abgeschnittener Datensätze verarbeitet werden muss, kann sich dies negativ auf die Leistung von LOAD auswirken. In Situationen, in denen mit zahlreichen Warnungen zu rechnen ist, können Sie den Änderungswert `norowwarnings` für den Dateityp verwenden, um die Aufzeichnung dieser Warnungen zu unterdrücken.

PAGEFREESPACE, INDEXFREESPACE und TOTALFREESPACE

Durch Dateneinfügings- und Datenaktualisierungsoperationen in Tabellen steigt mit der Zeit die Notwendigkeit zur Tabellen- und Indexreorganisation. Eine Lösung für dieses Problem besteht in der Erhöhung des freien Speicherbereichs für Tabellen und Indizes anhand von `pagefreespace`, `indexfreespace` und `totalfreespace`. Die ersten beiden Änderungswerte, die Priorität gegenüber dem Wert für `PCTFREE` besitzen, geben den Prozentsatz der Daten- und Indexseiten an, der als freier Speicherbereich verbleiben soll. Demgegenüber gibt `totalfreespace` den Prozentsatz der Gesamtseiten an, der als freier Speicherbereich am Ende der Tabelle angefügt werden soll.

Ladefunktionen zur Verwaltung der referenziellen Integrität

Zwar ist das Dienstprogramm LOAD normalerweise effizienter als das Dienstprogramm IMPORT, es erfordert jedoch eine Anzahl von Funktionen, um die referenzielle Integrität der geladenen Informationen sicherzustellen:

- **Tabellensperren** für die Steuerung des gemeinsamen Zugriffs und zur Verhinderung unkontrollierten Datenzugriffs während einer Ladeoperation
- **Tabellenstatus** und **Tabellenbereichsstatus**, über die entweder der Zugriff auf Daten gesteuert oder besondere Benutzeraktionen sondiert werden können
- **Ladeausnahmetabellen**, durch die sichergestellt wird, dass Zeilen mit ungültigen Daten nicht einfach ohne Ihr Wissen gelöscht werden

Überprüfen auf ungültige Integritätsbedingungen nach einer Ladeoperation

Nach einer Ladeoperation kann sich die geladene Tabelle unter Umständen im Status "Festlegen der Integrität anstehend" (im 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.
- Die Tabelle enthält generierte Spalten, und die Ladeoperation wurde mit einem Client der Version 7 oder früher eingeleitet.
- Der Tabelle sind IMQTs (Immediate Materialized Query Tables - sofort gespeicherte Abfragetabellen) oder sofort gespeicherte Zwischenspeichertabellen untergeordnet, die auf sie verweisen.
- Die Tabelle ist eine Zwischenspeichertabelle oder eine MQT (Materialized Query Table, gespeicherte Abfragetabelle).

Die Markierung STATUS des Eintrags für die geladene Tabelle in SYSCAT.TABLES gibt den Status "Festlegen der Integrität 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 SET INTEGRITY PENDING CASCADE angegeben werden, ob der Status "Festlegen der Integrität anstehend" der geladenen Tabelle direkt an die untergeordneten Tabellen weitergegeben werden soll oder nicht.

Wenn für die Tabelle sowohl Integritätsbedingungen als auch untergeordnete Fremdschlüsseltabellen, abhängige MQTs 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 SET INTEGRITY PENDING CASCADE IMMEDIATE

Die geladene Tabelle, ihre abhängigen MQTs und ihre abhängigen Zwischenspeichertabellen werden in den Status "Festlegen der Integrität anstehend" mit Lesezugriff versetzt.

INSERT, ALLOW READ ACCESS und SET INTEGRITY PENDING CASCADE DEFERRED

Nur die geladene Tabelle wird in den Status "Festlegen der Integrität anstehend" mit Lesezugriff versetzt. Untergeordnete Fremdschlüsseltabellen, untergeordnete MQTs und untergeordnete Zwischenspeichertabellen behalten ihren ursprünglichen Status bei.

INSERT, ALLOW NO ACCESS und SET INTEGRITY PENDING CASCADE IMMEDIATE

Die geladene Tabelle, ihre abhängigen MQTs und ihre abhängigen Zwischenspeichertabellen werden in den Status "Festlegen der Integrität anstehend, kein Zugriff" versetzt.

INSERT oder REPLACE, ALLOW NO ACCESS und SET INTEGRITY PENDING CASCADE DEFERRED

Nur die geladene Tabelle wird in den Status "Festlegen der Integrität anstehend, kein Zugriff" versetzt. Untergeordnete Fremdschlüsseltabellen, untergeordnete IMQTs und untergeordnete sofort gespeicherte Zwischenspeichertabellen behalten ihren ursprünglichen Status bei.

REPLACE, ALLOW NO ACCESS und SET INTEGRITY PENDING CASCADE IMMEDIATE

Die Tabelle und alle ihre untergeordneten Fremdschlüsseltabellen, untergeordneten IMQTs und untergeordneten sofort gespeicherten Zwischenspeichertabellen werden in den Status "Festlegen der Integrität 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 "Festlegen der Integrität anstehend" zu entfernen. Mit der Anweisung SET INTEGRITY wird eine Tabelle auf ungültige Integritätsbedingungen überprüft, und die Tabelle wird aus dem Status "Festlegen der Integrität anstehend" herausgenommen. Wenn alle Ladeoperationen im Einfügemodus (INSERT) ausgeführt werden, können die Integritätsbedingungen mit der Anweisung SET INTEGRITY inkrementell verarbeitet wer-

den (das heißt, 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 SET INTEGRITY PENDING CASCADE DEFERRED angegeben wird und die Anweisung SET INTEGRITY zur Überprüfung auf ungültige Integritätsbedingungen eingesetzt wird, werden die untergeordneten Tabellen in den Status "Festlegen der Integrität anstehend, kein Zugriff" versetzt. Sie müssen eine explizite Anforderung absetzen, um diesen Status für die Tabellen aufzuheben.

Wenn beim Laden einer Tabelle mit abhängigen MQTs oder abhängigen Zwischenspeichertabellen die Option INSERT angegeben wird und die Anweisung SET INTEGRITY zur Überprüfung auf ungültige Integritätsbedingungen verwendet wird, wird der Status "Festlegen der Integrität anstehend" für die Tabelle aufgehoben, und die Tabelle wird in den Status "Kein Versetzen von Daten" versetzt. Dies geschieht, um die nachfolgenden inkrementellen Aktualisierungen der abhängigen MQTs und die inkrementelle Weitergabe der abhängigen Zwischenspeichertabellen zu vereinfachen. Im Status "Kein Versetzen von Daten" sind Operationen, die das Versetzen von Zeilen innerhalb der Tabelle bewirken könnten, nicht zulässig.

Sie können den Status "Kein Versetzen von Daten" 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 MQTs statt, und die abhängigen Zwischenspeichertabellen werden in einen unvollständigen Status versetzt.

Falls die Option ALLOW READ ACCESS für eine 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 für Abfragen nur die Daten 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 "Festlegen der Integrität 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 die DB2-Datenbank 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:

- Der Tabelle werden neue Integritätsbedingungen hinzugefügt, während sie sich im Status "Festlegen der Integrität anstehend" befindet.
- Eine `LOAD REPLACE`-Operation wird ausgeführt, oder die Option `NOT LOGGED INITIALLY WITH EMPTY TABLE` wird aktiviert, nachdem die letzte Überprüfung auf Integritätsbedingungen für die Tabelle stattgefunden hat.
- Für eine übergeordnete Tabelle wird eine `LOAD REPLACE`-Operation ausgeführt, oder eine solche Tabelle wird nicht inkrementell auf Integrität überprüft.
- Vor einem Upgrade befindet sich die Tabelle im Status "Festlegen der Integrität anstehend". Wenn für die Tabelle zum ersten Mal nach dem Upgrade eine Überprüfung auf Integritätsbedingungen vorgenommen wird, ist eine vollständige Verarbeitung erforderlich.
- Der Tabellenbereich, der die Tabelle oder ihre übergeordnete Tabelle enthält, wird bis zu einem bestimmten Zeitpunkt aktualisierend wiederhergestellt, und die Tabelle sowie ihre übergeordnete Tabelle befinden sich in unterschiedlichen Tabellenbereichen.

Wenn eine Tabelle mindestens einen Wert `W` in der Spalte `CONST_CHECKED` des Katalogs `SYSCAT.TABLES` enthält und die Option `NOT INCREMENTAL` in der Anweisung `SET INTEGRITY` nicht angegeben ist, 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.

Die Anweisung `SET INTEGRITY` aktiviert keine `DELETE`-Trigger als Ergebnis des Löschens von Zeilen, die gegen Integritätsbedingungen verstoßen. Die Trigger sind jedoch aktiv, sobald für die Tabelle der Status "Festlegen der Integrität anstehend" aufgehoben wird. Wenn Sie Daten korrigieren und Zeilen aus der Ausnahmetabelle in die geladene Tabelle einfügen, werden demzufolge auch alle `INSERT`-Trigger aktiviert, die möglicherweise für die Tabelle definiert sind. Die Implikationen davon sollten berücksichtigt werden. Eine Möglichkeit wäre z. B., den `INSERT`-Trigger zu löschen, Zeilen aus der Ausnahmetabelle einzufügen und anschließend den `INSERT`-Trigger erneut zu erstellen.

Überprüfen auf ungültige Integritätsbedingungen mithilfe von SET INTEGRITY

In der Regel muss die Integritätsverarbeitung für eine Tabelle in drei Situationen manuell ausgeführt werden: nach dem Laden von Daten in eine Tabelle, beim Ändern einer Tabelle durch das Hinzufügen von Integritätsbedingungen für die Tabelle und beim Ändern einer Tabelle durch das Hinzufügen einer generierten Spalte.

Vorbereitende Schritte

- Um die Integritätsprüfung für eine Tabelle zu aktivieren und die Integritätsverarbeitung für diese auszuführen, müssen Sie eine der folgenden Voraussetzungen erfüllen:

- Zugriffsrechte CONTROL für die zu prüfenden Tabellen und, falls für einige Tabellen Ausnahmen übergeben werden, das Zugriffsrecht INSERT für die Ausnahmetabellen
- Zugriffsrecht CONTROL für alle Nachkommentabellen mit Fremdschlüsseln, alle als direkte Nachkommen identifizierten MQTs (Materialized Query Tables) sowie alle als direkte Nachkommen identifizierten Zwischenspeichertabellen, die von der Anweisung implizit in den Status 'Festlegen der Integrität anstehend' versetzt wurden.
- Berechtigung LOAD und, falls Ausnahmen an einige Tabellen übergeben wurden, Folgendes:
 - Zugriffsrechte SELECT und DELETE für jede zu überprüfende Tabelle.
 - INSERT für die Ausnahmetabellen
- Um die Integritätsprüfung für eine Tabelle zu aktivieren, ohne die Integritätsverarbeitung für diese auszuführen, müssen Sie eine der folgenden Voraussetzungen erfüllen:
 - Zugriffsrechte CONTROL für die zu prüfenden Tabellen
 - Zugriffsrecht CONTROL für alle Nachkommentabellen mit Fremdschlüsseln, alle als direkte Nachkommen identifizierten MQTs (Materialized Query Tables) sowie alle als direkte Nachkommen identifizierten Zwischenspeichertabellen, die von der Anweisung implizit in den Status 'Festlegen der Integrität anstehend' versetzt wurden.
 - Berechtigung LOAD
 - Berechtigung DATAACCESS
 - Berechtigung DBADM
- Um die Integritätsprüfung, die sofortige Aktualisierung oder die sofortige Weitergabe für Tabellen zu inaktivieren, müssen Sie die folgenden Voraussetzungen erfüllen:
 - Zugriffsrecht CONTROL für die Tabelle und alle Nachkommentabellen mit Fremdschlüsseln, alle als direkte Nachkommen identifizierten MQTs sowie alle als direkte Nachkommen identifizierten Zwischenspeichertabellen, deren Integritätsprüfung durch die Anweisung inaktiviert wurde.
 - Berechtigung LOAD

Informationen zu diesem Vorgang

Die Ladeoperation bewirkt, dass eine Tabelle automatisch in den Status 'Festlegen der Integrität anstehend' versetzt wird, wenn für die Tabelle Integritätsbedingungen definiert sind oder wenn sie abhängige Tabellen mit Fremdschlüsseln, abhängige MQTs (Materialized Query Tables, gespeicherte Abfragetabellen) oder abhängige Zwischenspeichertabellen aufweist. Sobald die Ladeoperation beendet ist, können Sie die Integrität der geladenen Daten prüfen und die Prüfung auf Integritätsbedingung für die Tabelle aktivieren. Wenn die Tabelle abhängige Tabellen mit Fremdschlüsseln, abhängige MQTs oder abhängige Zwischenspeichertabellen aufweist, werden diese automatisch in den Status 'Festlegen der Integrität anstehend' versetzt. Für jede dieser Tabellen muss eine separate Integritätsverarbeitung über das Fenster **Integrität festlegen** ausgeführt werden.

Wenn Sie eine Tabelle durch Hinzufügen eines Fremdschlüssels, einer Prüfung auf Integritätsbedingung oder einer generierten Spalte ändern, müssen Sie vor dem Ändern der Tabelle die Prüfung auf Integritätsbedingung inaktivieren. Nach dem Hinzufügen der Bedingung müssen Sie die vorhandenen Daten auf Verletzungen der neu hinzugefügten Integritätsbedingung überprüfen und die Prüfung auf Integritätsbedingung erneut aktivieren. Wenn Sie außerdem Daten in die Tabelle laden,

dürfen Sie die Prüfung auf Integritätsbedingung für die Tabelle erst dann wieder aktivieren, wenn alle Daten vollständig geladen wurden. Wenn Sie Daten in die Tabelle importieren, sollten Sie die Prüfung auf Integritätsbedingung für die Tabelle vor dem Import der Daten aktivieren.

Integritätsprüfung bezeichnet die Prüfung auf Verstöße gegen Integritätsbedingungen, ungültige Fremdschlüssel und fehlerhaft generierte Spalten. Integritätsverarbeitung bezieht sich neben der Ausführung einer Integritätsprüfung auch auf das Füllen von Identitätsspalten und generierten Spalten, auf die Aktualisierung von MQTs und die Weitergabe an Zwischenspeichertabellen.

In der Regel werden die referenzielle Integrität und Prüfung auf Integritätsbedingung für eine Tabelle automatisch aktiviert, MQTs werden automatisch sofort aktualisiert, und Zwischenspeichertabellen werden automatisch weitergegeben. In einigen Situationen muss diese Funktionsweise möglicherweise geändert werden.

Verwenden Sie zur Suche nach ungültigen Integritätsbedingungen über die Befehlszeile die Anweisung SET INTEGRITY.

Vorgehensweise

Gehen Sie wie folgt vor, um über die Steuerzentrale zu überprüfen, ob ungültige Integritätsbedingungen aufgetreten sind:

1. Öffnen Sie das Fenster **Integrität festlegen**: Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis der Ordner **Tabellen** angezeigt wird. Klicken Sie den Ordner **Tabellen** an. Alle vorhandenen Tabellen werden im Teilfenster auf der rechten Seite des Fensters angezeigt. Klicken Sie die gewünschte Tabelle mit der rechten Maustaste an, und wählen Sie **Integrität festlegen** im Kontextmenü aus. Das Fenster **Integrität festlegen** wird geöffnet.
2. Prüfen Sie den aktuellen Integritätsstatus der Tabelle, mit der Sie gerade arbeiten.
3. Gehen Sie wie folgt vor, um die Integritätsprüfung für eine Tabelle zu aktivieren, ohne die Tabellendaten zu prüfen:
 - a. Wählen Sie den Radioknopf **Sofort ohne Prüfung** aus.
 - b. Geben Sie den Integritätsverarbeitungstyp an, der aktiviert werden soll.
 - c. Wählen Sie das Optionsfeld **Vollständiger Zugriff** aus, um Datenversetzungsoperationen für die Tabelle (wie das Reorganisieren oder Umverteilen des Datenbestands) sofort auszuführen. Es sei jedoch darauf hingewiesen, dass nachfolgende Aktualisierungen für abhängige MQTs länger dauern. Ist der Tabelle eine MQT zugeordnet, sollten Sie diesen Radioknopf nicht auswählen, um den Zeitaufwand für die Aktualisierung der MQT zu reduzieren.
4. Gehen Sie wie folgt vor, um die Integritätsprüfung für eine Tabelle zu aktivieren und die vorhandenen Tabellendaten zu überprüfen:
 - a. Wählen Sie den Radioknopf **Sofort mit Prüfung** aus.
 - b. Wählen Sie den Integritätsverarbeitungstyp aus, der ausgeführt werden soll. Wenn unter **Aktueller Integritätsstatus** angezeigt wird, dass der Wert für geprüfte Integritätsbedingungen für die MQT unvollständig ist, können Sie keine inkrementelle Aktualisierung der MQT durchführen.
 - c. Optional: Wenn Sie generierte Spalten oder Identitätsspalten während der Integritätsverarbeitung füllen wollen, aktivieren Sie das Markierungsfeld **Erzwungene Generierung**.

- d. Handelt es sich bei der Tabelle nicht um eine Zwischenspeichertabelle, stellen Sie sicher, dass das Markierungsfeld **Entfernen** inaktiviert ist.
 - e. Wählen Sie den Radioknopf **Vollständiger Zugriff** aus, um Datenversetzungsoperationen für die Tabelle sofort auszuführen.
 - f. Optional: Geben Sie eine Ausnahmetabelle an. Jede Zeile, die Integritätsbedingungen verletzt, wird aus der Tabelle gelöscht und in die Ausnahmetabelle kopiert. Wenn Sie keine Ausnahmetabelle angeben, wird im Fall der Verletzung einer Integritätsbedingung nur die erste festgestellte Verletzung gemeldet. Die Tabelle verbleibt jedoch im Status 'Festlegen der Integrität anstehend'.
5. Gehen Sie wie folgt vor, um die Integritätsprüfung, die sofortige Aktualisierung oder die sofortige Weitergabe für eine Tabelle zu inaktivieren:
- a. Wählen Sie den Radioknopf **Inaktiv** aus. Die Tabelle wird in den Status 'Festlegen der Integrität anstehend' versetzt.
 - b. Mit der Option **Weitergabe** können Sie angeben, ob die Weitergabe (Kaskadierung) sofort oder erst später ausgeführt werden soll. Wenn die Weitergabe sofort durchgeführt wird, können Sie über die Markierungsfelder für **MQTs** (Materialized Query Tables), **Fremdschlüsseltabellen** und **Zwischenspeichertabellen** angeben, an welche Tabellen die Weitergabe erfolgen soll. Wenn Sie die Integritätsprüfung für eine übergeordnete Tabelle inaktivieren und angeben, dass die Änderungen an Fremdschlüsseltabellen weitergegeben werden sollen, werden die Integritätsbedingungen über Fremdschlüssel aller ihrer abhängigen Tabellen ebenfalls inaktiviert. Wenn Sie die Integritätsprüfung für eine zu Grunde liegende Tabelle inaktivieren und angeben, dass der Status **Überprüfung anstehend** an MQTs weitergegeben werden soll, werden die Merkmale der sofortigen Aktualisierung aller ihrer abhängigen MQTs ebenfalls inaktiviert. Wenn Sie die Integritätsprüfung für eine zugrunde liegende Tabelle inaktivieren und angeben, dass der Status 'Festlegen der Integrität anstehend' an Zwischenspeichertabellen weitergegeben werden soll, werden die Merkmale der sofortigen Weitergabe aller zugehörigen, abhängigen Zwischenspeichertabellen ebenfalls inaktiviert.

Beispiel

Tipp zur Fehlerbehebung

Symptom

Bei einem Versuch, eine Prüfung auf Integritätsbedingung, eine sofortige Aktualisierung oder eine sofortige Weitergabe für eine Tabelle zu aktivieren, wird die folgende Fehlermeldung angezeigt:

DB2-Nachricht

Eine abhängige Tabelle TABELLE1 kann mit der Anweisung SET INTEGRITY nicht überprüft werden, solange sich die übergeordnete oder zugrunde liegende Tabelle TABELLE2 im Status 'Festlegen der Integrität anstehend' befindet, oder wenn die Tabelle von der Anweisung SET INTEGRITY in diesen Status versetzt wird.

Dabei ist TABELLE1 die Tabelle, für die Sie die Prüfung auf Integritätsbedingung, die sofortige Aktualisierung oder die sofortige Weitergabe aktivieren möchten. Es handelt sich dabei um eine abhängige Tabelle von TABELLE2.

Mögliche Ursache

Die Integritätsprüfung, die sofortige Aktualisierung oder die sofortige Wei-

tergabe kann für eine Tabelle, die über eine übergeordnete oder zugrunde liegende Tabelle im Status 'Festlegen der Integrität anstehend' verfügt, nicht aktiviert werden.

Aktion

Heben Sie den Status 'Festlegen der Integrität anstehend' der übergeordneten oder zugrunde liegende Tabelle auf, indem Sie die Integritätsprüfung für die Tabelle aktivieren. Beginnen Sie dabei mit der Tabelle, die in der DB2-Nachricht als die übergeordnete oder zu Grunde liegende Tabelle angegeben ist. Ist diese Tabelle wiederum von einer anderen Tabelle abhängig, müssen Sie die Prüfung auf Integritätsbedingung von der Tabelle an aktivieren, von der die anderen Tabellen abhängig sind.

Achtung: Wenn die ausgewählte Tabelle eine zyklische referenzielle Integritätsbeziehung mit mindestens einer anderen Tabelle hat, können Sie das Fenster **Integrität festlegen** nicht verwenden, um die Integritätsprüfung zu aktivieren. In diesem Fall müssen Sie die Anweisung SQL SET INTEGRITY über den Befehlseditor absetzen.

Sperren von Tabellen während einer Ladeoperation

In den meisten Fällen setzt das Dienstprogramm LOAD Sperren auf Tabellenebene ein, um den Zugriff auf Tabellen einzuschränken. Die Stufe der Sperren richtet sich nach der Phase, in der sich die Ladeoperation befindet und danach, ob bei der Ladeoperation ein Lesezugriff zulässig ist oder nicht.

Eine Ladeoperation im Modus ALLOW NO ACCESS verwendet für die Dauer der Ladeoperation eine exklusive Sperre des Typs "Z" (Super Exclusive Lock) für die Tabelle.

Bevor eine Ladeoperation im Modus ALLOW READ ACCESS gestartet wird, wartet das Dienstprogramm LOAD, bis alle Anwendungen, die vor der Ladeoperation gestartet wurden, ihre Sperren für die Zieltabelle freigeben. Am Anfang einer Ladeoperation fordert das Dienstprogramm LOAD eine Aktualisierungssperre vom Typ "U-lock" (Update) für die Tabelle an. Diese Sperre wird solange beibehalten, bis die Daten festgeschrieben werden. Sobald das Dienstprogramm LOAD eine Aktualisierungssperre für die Tabelle anfordert, wartet es, bis alle Anwendungen, die vor dem Beginn der Ladeoperation Sperren für die Tabelle gehalten hatten, diese Sperren freigeben. Dies gilt auch dann, wenn es sich hierbei um kompatible Sperren handelt. Dies wird durch ein temporäres Upgrade der Aktualisierungssperre auf eine Sperre von Typ "Z" erreicht, die nicht mit neuen Anforderungen für Tabellensperren für die Zieltabelle in Konflikt steht, solange die angeforderten Sperren mit der Aktualisierungssperre der Ladeoperation kompatibel sind. Das Dienstprogramm LOAD führt beim Commit der Daten ein Upgrade der Sperre auf eine Sperre vom Typ "Z" durch, sodass sich das Commit etwas verzögern kann, 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 vor dem Laden freigeben, kann eine Zeitlimitüberschreitung eintreten. Für die Ladeoperation tritt jedoch keine Zeitlimitüberschreitung ein, während sie auf die Sperre vom Typ "Z" wartet, die für das Commit der Daten erforderlich ist.

Anwendungen mit einem Konflikt verursachenden Sperren

Verwenden Sie die Option LOCK WITH FORCE des Befehls **LOAD**, um eine Beendigung von Anwendungen, die Konflikte verursachende Sperren für die Zieltabelle besitzen, zu erzwingen, damit die Ladeoperation fortgesetzt werden kann. Bevor eine

Ladeoperation im Modus ALLOW READ ACCESS fortgesetzt werden kann, wird die Beendigung von Anwendungen, die die folgenden Sperren halten, erzwungen:

- Tabellensperren, die Konflikte mit einer Aktualisierungssperre des Typs "U" (Update Lock) für eine Tabelle verursachen (beispielsweise IMPORT oder INSERT).
- Alle Tabellensperren, die in der COMMIT-Phase der Ladeoperation vorhanden sind.

Anwendungen, die Konflikte verursachende Sperren für die Systemkatalogtabellen besitzen, werden durch das Dienstprogramm LOAD nicht zwingend beendet. Wenn die Beendigung einer Anwendung auf dem System durch das Dienstprogramm LOAD erzwungen wird, geht ihre Datenbankverbindung verloren, und das System gibt einen Fehler (SQL1224N) zurück.

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 "Backup anstehend" versetzt wird. Dieser Vorgang erfolgt unabhängig vom Zugriffsmodus. Wenn die Option LOCK WITH FORCE angegeben wird, wird die Beendigung aller Anwendungen erzwungen, die Sperren für Objekte im Tabellenbereich besitzen, die Konflikte mit einer gemeinsam benutzten Sperre des Typs "S" (Share Lock) verursachen.

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.

ALLOW NO ACCESS ist das Standardverhalten. Die Option ALLOW READ ACCESS hingegen verhindert jeglichen Schreibzugriff auf die Tabelle durch andere Anwendungen, lässt jedoch einen Lesezugriff auf bereits vorhandene 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. Betrachten Sie das folgende Beispiel:

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 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 während laufender oder nach fehlgeschlagenen Ladeoperationen - den Zugriff auf die Tabellendaten ermöglicht. Das Verhalten einer Ladeoperation im Modus 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 während der gesamten Ladeoperation möglich. Eine Ausnahme bilden hier lediglich der Anfang und das Ende der Operation.

Erster Fall: Die Ladeoperation fordert eine spezielle exklusive Sperre vom Typ "Z" für eine kurze Dauer gegen Ende ihrer SETUP-Phase an. Wenn eine Anwendung eine inkompatible Sperre für die Tabelle hält, bevor die Ladeoperation diese spezielle Z-Sperre anfordert, wartet die Ladeoperation eine bestimmte Zeit darauf, dass diese inkompatible Sperre freigegeben wird, bevor sie aufgrund einer Zeitlimitüberschreitung fehlschlägt. Die entsprechende Zeitdauer wird mit dem Datenbankkonfigurationsparameter *locktimeout* festgelegt. Bei Angabe der Option LOCK WITH FORCE erzwingt die Ladeoperation die Beendigung anderer Anwendungen, um eine Zeitlimitüberschreitung zu vermeiden. Die Ladeoperation fordert die spezielle Z-Sperre an, schreibt die Phase fest, gibt die Sperre frei und fährt anschließend mit der Ladephase (LOAD-Phase) fort. Allen Anwendungen, die für die Tabelle eine

Lesesperre anfordern, nachdem die Ladeoperation im Modus ALLOW READ ACCESS gestartet wurde, wird die Sperre mit Grant erteilt. Hierbei kommt es zu keinen Konflikten mit dieser speziellen Z-Sperre. Neue Anwendungen, die versuchen, vorhandene Daten aus der Zieltabelle zu lesen, können dies tun.

Zweiter Fall: Vor dem Commit der Daten am Ende der Ladeoperation fordert das Dienstprogramm LOAD eine exklusive Sperre des Typs "Z" für die Tabelle an. Das Dienstprogramm LOAD wartet, bis alle Anwendungen, die Sperren für die Tabelle halten, diese freigeben. Hierdurch kann sich vor dem Commit der Daten eine Verzögerung ergeben. Durch Verwendung der Option LOCK WITH FORCE wird die Beendigung von Anwendungen, die Konflikte verursachen, erzwungen und so eine Fortsetzung der Ladeoperation ohne Verzögerung ermöglicht. Normalerweise fordert eine Ladeoperation im Modus 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.

Wird das Dienstprogramm LOAD für eine Tabelle ausgeführt, die für mehrere Datenbankpartitionen definiert ist, wird das Ladeprozessmodell für jede einzelne Datenbankpartition ausgeführt. Dies bedeutet, dass Sperren unabhängig von den übrigen Datenbankpartitionen erteilt und freigegeben werden. Dies führt dazu, dass Deadlocks entstehen können, wenn eine Abfrage oder eine andere Operation zum gleichen Zeitpunkt ausgeführt wird und versucht, dieselben Sperren anzufordern. Angenommen Operation A hat eine Tabellensperre für die Datenbankpartition 0 erhalten und die Ladeoperation hat eine Tabellensperre für die Datenbankpartition 1 erhalten. Dadurch kann sich ein Deadlock ergeben, da Operation A auf die Erteilung einer Tabellensperre für die Datenbankpartition 1 wartet, während die Ladeoperation auf eine Tabellensperre für die Datenbankpartition 0 wartet. In diesem Fall führt der Deadlock-Detektor willkürlich für eine der Operationen einen Rollback durch.

Anmerkung:

1. Wenn eine Ladeoperation abgebrochen wird oder fehlschlägt, behält sie dieselbe Zugriffsebene bei, die beim Absetzen der Ladeoperation angegeben wurde. Wenn also eine Ladeoperation im Modus ALLOW NO ACCESS fehlschlägt, ist ein Zugriff auf die Tabellendaten erst dann möglich, wenn der Befehl LOAD TERMINATE oder LOAD RESTART abgesetzt wurde. Wird eine Ladeoperation im Modus ALLOW READ ACCESS abgebrochen, sind die zuvor vorhandenen Tabellendaten weiterhin für den Lesezugriff verfügbar.
2. Falls die Option ALLOW READ ACCESS für eine abgebrochene oder fehlgeschlagene Ladeoperation angegeben war, kann sie auch für die LOAD RESTART- oder LOAD TERMINATE-Operation angegeben werden. War für die abgebrochene oder fehlgeschlagene 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 vorab vorhandene Daten erst dann abgefragt werden, wenn die Ladeoperation abgeschlossen ist.
- Die Indizes wurden als ungültig gekennzeichnet, und ihre Wiederherstellung steht an. Indizes können in bestimmten Szenarios für eine aktualisierende Recovery 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 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, die sich im Status Set Integrity Pending No Access (Festlegen der Integrität anstehend, kein Zugriff) 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.

Tabellenbereichsstatus während und nach Ladeoperationen

Das Dienstprogramm LOAD setzt Angaben zum Tabellenbereichsstatus ein, um die Datenbankkonsistenz während einer Ladeoperation zu gewährleisten. Über diese Angaben zum Status ist es möglich, den Zugriff auf Daten zu steuern oder Benutzeraktionen zu sondieren.

Das Dienstprogramm LOAD versetzt die Tabellenbereiche, die von der Ladeoperation betroffen sind, nicht in den Quiescemodus (d. h. sperrt sie nicht permanent) und verwendet Statuswerte für Tabellenbereiche nur bei Ladeoperationen, bei denen Sie den Parameter **COPY NO** angegeben haben.

Mit dem Befehl **LIST TABLESPACES** können Sie die Statuswerte von Tabellenbereichen prüfen. Tabellenbereiche können sich in mehreren Status gleichzeitig befinden. Der Befehl **LIST TABLESPACES** gibt die folgenden Statuswerte zurück:

Normal

Der normale Status ist der Anfangsstatus eines Tabellenbereichs nach seiner Erstellung und gibt an, dass derzeit keine (abnormalen) Status in dem Tabellenbereich vorhanden sind.

Laden läuft

Der Status "Laden läuft" gibt an, dass für den Tabellenbereich eine Ladeoperation in Bearbeitung ist. Dieser Status verhindert, dass abhängige Tabellen während einer Ladeoperation gesichert werden. Dieser Tabellenbereichsstatus unterscheidet sich dadurch vom Tabellenstatus "Laden läuft" (der in allen Ladeoperationen verwendet wird), da das Dienstprogramm LOAD Tabellenbereiche nur dann in den Status "Laden läuft" versetzt, wenn Sie den Parameter **COPY NO** bei einer wiederherstellbaren Datenbank angegeben haben. Die Tabellenbereiche behalten diesen Status für die Dauer der Ladeoperation bei.

Backup anstehend

Wenn Sie eine Ladeoperation für eine wiederherstellbare Datenbank ausführen und den Parameter **COPY NO** angeben, werden Tabellenbereiche in den Tabellenbereichsstatus "Backup anstehend" nach der ersten Festschreibung versetzt. Ein Tabellenbereich mit dem Status "Backup anstehend" kann nicht aktualisiert werden. Sie können den Status "Backup anstehend" für den Tabellenbereich nur durch die Sicherung des Tabellenbereichs entfernen. Der Tabellenbereich verbleibt auch bei einem Abbruch der Ladeoperation im Status "Backup anstehend", weil der Status des Tabellenbereichs am Beginn der Ladeoperation geändert wird und nicht zurückgesetzt werden kann.

Restore anstehend

Wenn Sie erfolgreich eine Ladeoperation mit der Option **COPY NO** ausführen, die Datenbank wiederherstellen und bei dieser Operation eine aktualisierende Recovery ausführen, werden die zugeordneten Tabellenbereiche in den Status "Restore anstehend" versetzt. Um den Status "Restore anstehend" für Tabellenbereiche aufzuheben, muss ein Restore ausgeführt werden.

Anmerkung: DB2 LOAD versetzt Tabellenbereiche nicht in den Status **Laden anstehend** oder **Löschen anstehend**.

Beispiel für einen Tabellenbereichsstatus

Wenn Sie eine Eingabedatei (staffdata.del) wie folgt in die Tabelle NEWSTAFF laden:

```
update db cfg for sample using logretain recovery;
backup db sample;
connect to sample;
create table newstaff like staff;
load from staffdata.del of del insert into newstaff copy no;
connect reset;
```

und Sie eine weitere Sitzung öffnen und die folgenden Befehle absetzen:

```
connect to sample;
list tablespaces;
connect reset;
```

befindet sich USERSPACE1 (der Standardtabellenbereich für die Beispieldatenbank) im Status "Laden läuft" und nach der ersten Festschreibung außerdem im Status "Backup anstehend". Nachdem die Ladeoperation beendet ist, zeigt der Befehl **LIST TABLESPACES**, dass der Tabellenbereich USERSPACE1 sich nun im Status "Backup anstehend" befindet:

Tabellenbereichs-ID		= 2
Name		= USERSPACE1
Typ		= Von der Datenbank verwalteter Bereich
Inhalt		= Alle permanenten Daten. Großer Tabellenbereich.
Status		= 0x0020
Detaillierte Erläuterung:		
Backup anstehend		

Tabellenstatus während und nach Ladeoperationen

Das Dienstprogramm LOAD setzt Angaben zum Tabellenstatus ein, um die Datenbankkonsistenz während einer Ladeoperation zu gewährleisten. Über diese Angaben zum Status ist es möglich, den Zugriff auf Daten zu steuern oder Benutzeraktionen zu sondieren.

Setzen Sie zum Festlegen des Status einer Tabelle den Befehl **LOAD QUERY** ab, der außerdem den Status einer Ladeoperation prüft. Tabellen können sich in mehreren Status gleichzeitig befinden. Der Befehl **LOAD QUERY** gibt die folgenden Statuswerte zurück:

Normal

Der normale Status ist der Anfangsstatus einer Tabelle nach ihrer Erstellung und gibt an, dass derzeit keine (abnormalen) Status in der Tabelle vorhanden sind.

Lesezugriff

Wenn Sie die Option **ALLOW READ ACCESS** angeben, gilt der Status "Lesezugriff" für die Tabelle. Die Daten, die vor dem Aufruf des Befehls LOAD in der Tabelle vorhanden waren, sind während der Ladeoperation nur im Lesezugriff verfügbar.

Wenn Sie die Option **ALLOW READ ACCESS** angeben und die Ladeoperation fehlschlägt, sind die Daten, die vor der Ladeoperation in der Tabelle vorhanden waren, nach dem Fehler weiterhin im Lesezugriff verfügbar.

Laden läuft

Der Tabellenstatus "Laden läuft" gibt an, dass für die Tabelle eine Ladeoperation in Bearbeitung ist. Das Dienstprogramm LOAD entfernt diesen Übergangszustand, nachdem das Laden erfolgreich abgeschlossen wurde. Wenn die Ladeoperation jedoch fehlschlägt oder unterbrochen wird, ändert sich der Tabellenstatus in "Laden anstehend".

Umverteilung läuft

Der Tabellenstatus "Umverteilung läuft" gibt an, dass für die Tabelle eine Umverteilung in Bearbeitung ist. Das Dienstprogramm REDISTRIBUTE entfernt diesen Übergangszustand, nachdem die Verarbeitung der Tabelle erfolgreich abgeschlossen wurde. Wenn die Umverteilungsoperation jedoch fehlschlägt oder unterbrochen wird, ändert sich der Tabellenstatus in "Umverteilung anstehend".

Laden anstehend

Der Tabellenstatus "Laden anstehend" gibt an, dass eine Ladeoperation fehlgeschlagen ist oder unterbrochen wurde. Mit einem der folgenden Schritte können Sie den Status "Laden anstehend" entfernen:

- Ermitteln Sie zunächst die Fehlerursache. Wenn das Dienstprogramm LOAD beispielsweise nicht über genügend Plattenspeicherplatz verfügte, fügen Sie dem Tabellenbereich Container hinzu. Starten Sie anschließend die Ladeoperation erneut.
- Beenden Sie die Ladeoperation.
- Führen Sie eine **LOAD REPLACE**-Operation für die Tabelle aus, bei der eine Ladeoperation fehlgeschlagen ist.
- Stellen Sie Tabellenbereiche für die Ladetabelle wieder her. Verwenden Sie dazu den Befehl **RESTORE DATABASE** mit dem letzten Backup eines Tabellenbereichs oder einer Datenbank, und führen Sie danach weitere Recoveryfunktionen aus.

Umverteilung anstehend

Der Tabellenstatus "Umverteilung anstehend" gibt an, dass eine Umverteilungsoperation fehlgeschlagen ist oder unterbrochen wurde. Sie können die Operation **REDISTRIBUTE CONTINUE** oder **REDISTRIBUTE ABORT** ausführen, um den Status "Umverteilung anstehend" zu entfernen.

Für Laden nicht neu startbar

Im Status "Für Laden nicht neu startbar" wird eine Tabelle teilweise geladen und lässt ein erneutes Starten der Ladeoperation nicht zu. Es gibt zwei Situationen, in denen eine Tabelle in den Status "Für Laden nicht neu startbar" versetzt wird:

- Wenn Sie eine aktualisierende Recovery nach einer fehlgeschlagenen Ladeoperation ausführen, die Sie nicht erfolgreich neu starten oder beenden konnten
- Wenn Sie eine Restoreoperation auf der Basis eines Online-Backups ausführen, der erstellt wurde, während die Tabelle sich im Tabellenstatus "Laden läuft" oder "Laden anstehend" befand.

Für die Tabelle gilt außerdem der Status "Laden anstehend". Wenn Sie für die Tabelle den Status "Für Laden nicht neu startbar" entfernen möchten, setzen Sie den Befehl **LOAD TERMINATE** oder **LOAD REPLACE** ab.

Festlegen der Integrität anstehend

Der Status "Festlegen der Integrität anstehend" gibt an, dass für die geladene Ta-

belle Integritätsbedingungen gelten, die noch nicht überprüft wurden. Das Dienstprogramm LOAD versetzt eine Tabelle in diesen Status, sobald eine Ladeoperation für eine Tabelle mit Integritätsbedingungen beginnt. Mit der Anweisung SET INTEGRITY können Sie den Status "Festlegen der Integrität anstehend" für die Tabelle aufheben.

Indizes des Typs 1

Der Status "Indizes des Typs 1" gibt an, dass die Tabelle derzeit Indizes des Typs 1 verwendet. Indizes des Typs 1 werden nicht mehr unterstützt und müssen in Indizes des Typs 2 konvertiert werden. Sie können Indizes des Typs 1 wie folgt in den Typ 2 konvertieren: Mit der Option CONVERT im Befehl REORG INDEXES/TABLE oder mithilfe der Ausgabe des Befehls **db2IdentifyType1**. Der Befehl **db2IdentifyType1** generiert die entsprechenden Befehle für die Konvertierung der Indizes des Typs 1, die in Tabellen oder Schemata der angegebenen Datenbank gefunden werden. Weitere Informationen hierzu finden Sie im Thema „Konvertieren von Indizes des Typs 1 in Indizes des Typs 2“.

Nicht verfügbar

Eine Tabelle wird durch eine aktualisierende Recovery, die über eine nicht wiederherstellbare Ladeoperation erfolgt, in den Status "Nicht verfügbar" versetzt. In diesem Status ist die Tabelle nicht verfügbar, Sie müssen Sie freigeben oder aus einem Backup wiederherstellen.

Beispiel für eine Tabelle mit mehreren Status

Wenn Sie eine Eingabedatei (staffdata.del) mit einem großen Datenvolumen wie folgt in die Tabelle NEWSTAFF laden:

```
connect to sample;  
create table newstaff like staff;  
load from staffdata.del of del insert into newstaff allow read access;  
connect reset;
```

und Sie eine weitere Sitzung öffnen und die folgenden Befehle absetzen:

```
connect to sample;  
load query table newstaff;  
connect reset;
```

zeigt der Befehl **LOAD QUERY**, dass die Tabelle NEWSTAFF sich in den beiden Tabellenstatus "Lesezugriff" und "Laden läuft" befindet:

```
Tabellenstatus:  
  Laden läuft  
  Lesezugriff
```

LOAD-Ausnahmetabellen

Eine LOAD-Ausnahmetabelle stellt einen konsolidierten Bericht aller Zeilen zur Verfügung, die gegen eindeutige Indexregeln, Bereichsvorgaben und Sicherheitsrichtlinien verstoßen haben, während eine Ladeoperation ausgeführt wurde. Sie können eine LOAD-Ausnahmetabelle angeben, indem Sie die Klausel FOR EXCEPTION des Befehls **LOAD** verwenden.

Einschränkung: 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 kann außerdem auch nicht partitioniert werden und darf keinen eindeutigen Index haben. Darüber hinaus kann in den folgenden Fällen keine Ausnahmetabelle angegeben werden:

- Wenn die Zieltabelle Sicherheit mit kennsatzbasierter Zugriffssteuerung (LBAC) verwendet und mindestens eine XML-Spalte enthält.
- Wenn die Zieltabelle in Bereiche partitioniert ist und mindestens eine XML-Spalte enthält.

Die Ausnahmetabelle, die mit dem Dienstprogramm LOAD verwendet wird, ist identisch mit den Ausnahmetabellen, die von der Anweisung SET INTEGRITY verwendet werden. Es handelt sich dabei um eine vom Benutzer erstellte Tabelle, die die Definition der zu ladenden Tabelle wiedergibt und einige zusätzliche Spalten beinhaltet.

Eine LOAD-Ausnahmetabelle kann dem Tabellenbereich, in dem sich die zu ladende Tabelle befindet, oder einem anderen Tabellenbereich zugeordnet werden. In beiden Fällen sollten die LOAD-Ausnahmetabelle und die zu ladende Tabelle derselben Datenbankpartitionsgruppe zugeordnet werden und denselben Verteilungsschlüssel aufweisen. Stellen Sie außerdem sicher, dass die Ausnahmetabelle und die zu ladende Tabelle über dieselbe Partitionszuordnungs-ID (SYSIBM.SYSTABLES.PMAP_ID) verfügen. Diese kann sich potenziell während der Umverteilungsoperation (Operation zum Hinzufügen/Löschen von Datenbankpartitionen) unterscheiden.

Einsatzmöglichkeiten einer Ausnahmetabelle

Eine Ausnahmetabelle können Sie verwenden, um Daten zu laden, die über einen eindeutigen Index verfügen und die doppelte Datensätze enthalten können. Wenn Sie keine Ausnahmetabelle angeben und doppelte Datensätze gefunden werden, wird die Ladeoperation fortgesetzt. In diesem Fall wird lediglich eine Warnung über die gelöschten doppelten Datensätze angezeigt. Die doppelten Datensätze werden nicht protokolliert.

Nach Fertigstellung der Ladeoperation können Sie die Informationen in der Ausnahmetabelle verwenden, um fehlerhafte Daten zu korrigieren. Anschließend können die korrigierten Daten in die Tabelle eingefügt werden.

Zeilen werden an die in der Ausnahmetabelle vorhandenen Informationen angefügt. Da nicht überprüft wird, ob die Tabelle leer ist, werden neue Informationen einfach an die ungültigen Zeilen aus vorherigen Ladeoperationen angefügt. Wenn die Ausnahmetabelle nur die ungültigen Zeilen der aktuellen Ladeoperation enthalten soll, müssen Sie die bereits vorhandenen Zeilen löschen, bevor Sie das Dienstprogramm aufrufen. Bei der Definition einer Ladeoperation können Sie festlegen, dass der Zeitpunkt, zu dem eine Verletzung festgestellt wird, sowie der Name der verletzten Integritätsbedingung in der Ausnahmetabelle aufgezeichnet werden.

Da alle Löschereignisse protokolliert werden, kann das Protokoll während der DELETE-Phase der Ladeoperation stark an Umfang zunehmen, wenn sehr viele Datensätze gegen die Eindeutigkeitsbedingung verstoßen.

Zeilen, die vor der Indexerstellung aufgrund ungültiger Daten zurückgewiesen wurden, werden nicht in die Ausnahmetabelle aufgenommen.

Fehlgeschlagener oder unvollständiger Ladevorgang

Erneutes Starten einer unterbrochenen Ladeoperation

Wenn während der Ausführung einer Ladeoperation ein Fehler auftritt oder wenn diese unterbrochen wird, können Sie diese mit dem Dienstprogramm LOAD beenden, die Tabelle erneut laden oder die Ladeoperation nochmals starten.

Wenn das Dienstprogramm LOAD aufgrund eines Benutzerfehlers nicht gestartet werden kann, etwa weil eine Datendatei fehlt oder weil ungültige Spaltennamen festgestellt wurden, wird es beendet, und die Zieltabelle verbleibt in einem normalen Status.

Wenn die Ladeoperation beginnt, wird die Zieltabelle in den Tabellenstatus "Laden läuft" versetzt. Falls ein Fehler auftritt, ändert sich der Tabellenstatus in "Laden anstehend". Um die Tabelle wieder in einen anderen Status zu versetzen, können Sie **LOAD TERMINATE** eingeben, um die Operation mit einem Rollback zurückzusetzen, **LOAD REPLACE** eingeben, um die gesamte Tabelle erneut zu laden, oder **LOAD RESTART** eingeben.

Normalerweise ist es in dieser Situation zu empfehlen, dass Sie die Ladeoperation erneut starten. Dies spart Zeit, weil das Dienstprogramm LOAD die Ladeoperation ab dem zuletzt erreichten fehlerfreien Punkt innerhalb des Verarbeitungsablaufs und nicht vom Beginn der Operation an erneut startet. Wo genau der Neustart der Operation beginnt, hängt von den Parametern ab, die im ursprünglichen Befehl angegeben wurden. Wenn die Option SAVECOUNT angegeben wurde und bei der zuvor ausgeführten Ladeoperation in der LOAD-Phase ein Fehler aufgetreten ist, dann startet die Ladeoperation beim letzten erreichten Konsistenzpunkt. Andernfalls startet die Ladeoperation am Beginn der letzten Phase, die erfolgreich ausgeführt wurde (also bei der LOAD-, BUILD- oder DELETE-Phase).

Beim Laden von XML-Dokumenten ist die Vorgehensweise etwas anders. Da die Option SAVECOUNT für das Laden von XML-Daten nicht unterstützt wird, werden Ladeoperationen, bei denen in der LOAD-Phase ein Fehler auftritt, vom Beginn der Operation an erneut gestartet. Wie bei anderen Datentypen werden Indizes im Modus REBUILD erstellt, wenn bei der Ladeoperation in der BUILD-Phase ein Fehler auftritt; also wird die Tabelle durchsucht, um alle Indexschlüssel aus jeder Zeile zu erfassen. Zur Erfassung der Indexschlüssel müssen jedoch auch alle XML-Dokumente durchsucht werden. Zum Durchsuchen von XML-Dokumenten nach Schlüsseln muss eine erneute Syntexanalyse für diese Dokumente erfolgen; diese Operation ist mit einem hohen Systemaufwand verbunden. Darüber hinaus müssen zuerst die internen XML-Indizes, wie z. B. Indizes für Regionen oder Pfade, erneut erstellt werden; dazu muss außerdem das XDA-Objekt durchsucht werden.

Nachdem Sie den Fehler behoben haben, der zu dem Fehler in der Ladeoperation führte, können Sie den Ladebefehl erneut eingeben. Vergewissern Sie sich, dass Sie dabei genau die gleichen Parameter eingeben wie beim ursprünglichen Befehl, so dass das Dienstprogramm LOAD die erforderlichen temporären Dateien lokalisieren kann. Eine Ausnahme bildet hierbei der Fall, dass Sie den Lesezugriff nicht zulassen wollen. Eine Ladeoperation, in der die Option ALLOW READ ACCESS angegeben wurde, kann auch mit der Option ALLOW NO ACCESS erneut gestartet werden.

Anmerkung: Versuchen Sie nicht, temporäre Dateien, die vom Dienstprogramm LOAD erstellt wurden, zu löschen oder zu ändern.

Wenn die mit dem folgenden Befehl eingeleitete Ladeoperation fehlschlägt:

```
LOAD FROM dateiname OF dateityp
SAVECOUNT n
MESSAGES nachrichtendatei
lademethode
INTO zieltabellenname
```

dann müssen Sie sie erneut starten, indem Sie die angegebene Lademethode (lademethode) durch die Methode RESTART ersetzen:

```
LOAD FROM dateiname OF dateityp
SAVECOUNT n
MESSAGES nachrichtendatei
RESTART
INTO zieltabellenname
```

Fehlgeschlagene Ladeoperationen, die nicht erneut gestartet werden können

Fehlgeschlagene oder unterbrochene Ladeoperationen können nicht erneut gestartet werden, wenn die dabei verwendete Tabelle sich im Status "Für Laden nicht neu startbar" befindet. Tabellen werden aus den folgenden Gründen in diesen Status versetzt:

- Nach einer fehlgeschlagenen Ladeoperation, die nicht erfolgreich erneut gestartet oder beendet werden konnte, wird eine aktualisierende Recovery ausgeführt.
- Eine Restoreoperation wird auf der Basis eines Online-Backups ausgeführt, der erstellt wurde, während die Tabelle sich im Tabellenstatus "Laden läuft" oder "Laden anstehend" befand.

Geben Sie in diesem Fall entweder den Befehl **LOAD TERMINATE** oder **LOAD REPLACE** ein.

Erneutes Starten oder Beenden einer Ladeoperation mit ALLOW READ ACCESS

Eine unterbrochene oder abgebrochene Ladeoperation, in der die Option ALLOW READ ACCESS angegeben wurde, kann auch mit der Option ALLOW READ ACCESS erneut gestartet oder beendet werden. Durch die Verwendung der Option ALLOW READ ACCESS können andere Anwendungen die Tabellendaten abfragen, während die Operation zum Beenden oder erneutem Starten läuft. Wie auch bei einer Ladeoperation im Modus ALLOW READ ACCESS ("Lesezugriff zulassen") wird die Tabelle vor dem Commit der Daten exklusiv gesperrt.

Informationen zu diesem Vorgang

Falls das Indexobjekt nicht verfügbar ist oder als ungültig gekennzeichnet wurde, ist eine LOAD RESTART- oder LOAD TERMINATE-Operation im Modus ALLOW READ ACCESS nicht zulässig.

Falls die ursprüngliche Ladeoperation in der INDEX COPY-Phase unterbrochen oder abgebrochen wurde, ist eine LOAD RESTART-Operation im Modus ALLOW READ ACCESS nicht zulässig, da der Index beschädigt sein könnte.

Falls eine Ladeoperation im Modus ALLOW READ ACCESS während der LOAD-Phase unterbrochen oder abgebrochen wurde, wird sie in der LOAD-Phase erneut gestartet. Wurde sie während einer der anderen Phasen unterbrochen oder abgebrochen, findet der Neustart in der BUILD-Phase statt. Sofern die ursprüngliche Ladeoperation im Modus ALLOW NO ACCESS ("Keinen Zugriff zulassen") ausgeführt wurde, setzt die LOAD RESTART-Operation in der DELETE-Phase ein, wenn die ur-

sprüngliche Ladeoperation diesen Punkt erreicht hat 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 wird die abgebrochene oder unterbrochene Ladeoperation generell mit einer minimalen Verzögerung zurückgesetzt. Wird jedoch ein Befehl ALLOW READ ACCESS 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 auf, ob die Option ALLOW READ ACCESS für die LOAD TERMINATE-Operation angegeben wurde. Eine Verzögerung findet 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" (Not Load Restartable) lautet. Die Tabelle kann während einer aktualisierenden Recovery in den Tabellenstatus "Für Laden nicht neu startbar" versetzt werden. Dies ist beispielsweise möglich, wenn die aktualisierende Recovery bis zu einem Zeitpunkt ausgeführt wird, der vor dem Ende einer Ladeoperation lag, oder wenn eine aktualisierende Recovery über eine abgebrochene Ladeoperation hinweg, jedoch nicht bis zum Ende der LOAD TERMINATE- oder LOAD RESTART-Operation ausgeführt wird.

Recovery von Daten mithilfe der Datei mit den Angaben zur Speicherposition der Ladekopie

Anhand der Registrierdatenbankvariablen **DB2LOADREC** wird die Datei mit den Angaben zur Speicherposition der Ladekopie identifiziert. Diese Datei wird während der aktualisierenden Recovery zum Lokalisieren der Ladekopie verwendet.

DB2LOADREC enthält die folgenden 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 Recovery stattfindet.

Anmerkung:

1. In einer Mehrpartitionsdatenbank muss die Registrierdatenbankvariable **DB2LOADREC** unter Verwendung des Befehls **db2set** für alle Datenbankpartitionsserver festgelegt werden.
2. In einer Mehrpartitionsdatenbank 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 Registrierdatenbankvariable **DB2LOADREC** angegeben ist, ungültig ist, wird der ungültige Eintrag durch die 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 aufgezählte Ladekopie wiederholt. Beispiel:

```

TIMestamp      19950725182542      * Time stamp generated at load time
DBPartition    0                  * DB Partition number (OPTIONAL)
SCHema        PAYROLL             * Schema of table loaded
TABlename     EMPLOYEES           * Table name
DATabasename  DBT                 * Database name
DB2instance   toronto            * DB2INSTANCE
BUFFernumber  NULL                * Number of buffers to be used for
                                Recovery SESSionnumber NULL
                                * Number of sessions to be used
                                for Recovery TYPeofmedia L
                                * Type of media - L for local device
                                A for TSM
                                0 for other vendors

LOCationnumber 3                  * Number of locations
ENTry         /u/toronto/dbt.payroll.employes.001
ENT           /u/toronto/dbt.payroll.employes.002
ENT           /dev/rmt0
TIM           19950725192054
DBP           18
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

```

Anmerkung:

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 der Felder BUF und SES (die den Wert NULL haben können) sowie des Feldes DBP (das nicht in der Liste aufgezählt ist).

führt zu sein braucht). Falls für SES NULL definiert ist, wird der durch den Konfigurationsparameter *dft_loadrec_ses* 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 über eine Funktion zum Speichern der Ladekopiedaten verfügt.
7. Wenn Sie eine Ladeoperation aufrufen und hierbei die Option *COPY NO* oder *NONRECOVERABLE* angeben und nach Abschluss der Operation keine Backup-Kopie der Datenbank oder der betroffenen Tabellenbereiche erstellen, können Sie die Datenbank oder die Tabellenbereiche nicht in dem Zustand wiederherstellen, in dem sich diese nach der Ladeoperation befinden werden. Das heißt, dass Sie keine aktualisierende Recovery 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 Recoveryprotokolls für die Datenbank die Zeitmarke für diese spezifische Ladeoperation ermitteln. In einer Mehrpartitionsdatenbank ist die Datei des Recoveryprotokolls für jede Datenbankpartition lokal.

LOAD-Speicherauszugsdatei

Der Änderungswert *dumpfile* für den Dateityp 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 Subagenten angibt sowie die Datenbankpartitionsnummer, 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 Subagenten für Laden in Datenbankpartition fünf zurückgewiesen wurden, in einer Datei namens */u/username/dumpit.load.005* gespeichert. Zeilen, die vom Subagenten für Laden in Datenbankpartition zwei zurückgewiesen wurden, werden in einer Datei namens */u/username/dumpit.load.002* gespeichert, und Zeilen, die vom Subagenten für Partitionierung auf Datenbankpartition 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.

Temporäre Dateien für das Dienstprogramm LOAD

DB2 erstellt bei der LOAD-Verarbeitung temporäre Binärdateien. Diese Dateien werden zur Recovery einer Ladeoperation nach einem Systemabsturz, bei LOAD TERMINATE-Operationen, für Warnungen und Fehlermeldungen sowie die Laufzeitsteuerdaten verwendet.

Temporäre Dateien für das Dienstprogramm LOAD 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.

Der Pfad der temporären Dateien befindet sich auf der Servermaschine. Der Zugriff auf diesen Pfad erfolgt ausschließlich durch die DB2-Instanz. Daher muss jede Angabe eines Pfadnamens im Parameter *temp-pfadname* unbedingt die Verzeichnisstruktur auf dem Server und nicht die auf dem Client berücksichtigen. Darüber hinaus muss der Eigner der DB2-Instanz unbedingt über Lese- und Schreibberechtigung für den Pfad verfügen.

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.

Protokollsätze für das Dienstprogramm LOAD

Der Dienstprogramm-Manager erzeugt Protokollsätze für eine Reihe von DB2-Dienstprogrammen. Dazu zählt auch das Dienstprogramm LOAD.

Die folgenden Protokollsätze kennzeichnen den Anfang oder das Ende eines bestimmten Vorgangs während einer Ladeoperation:

• SETUP-Phase

- LOAD START. Dieser Protokollsatz markiert den Anfang der SETUP-Phase einer Ladeoperation.
- Commitprotokollsatz. Dieser Protokollsatz weist auf einen erfolgreichen Abschluss der SETUP-Phase hin.
- Abbruchsprotokollsatz (Abort). Dieser Protokollsatz weist darauf hin, dass die SETUP-Phase fehlgeschlagen ist. Bei Einzelpartitionsdatenbanken wird stattdessen ein Commitprotokollsatz für "Local Pending" generiert, wenn in der SETUP-Phase des Ladevorgangs ein Fehler auftritt, bevor die Tabelle physisch geändert wurde.

• LOAD-Phase

- LOAD START. Dieser Protokollsatz markiert den Anfang der LOAD-Phase einer Ladeoperation.
- Commitprotokollsatz für "Local Pending". Dieser Protokollsatz weist auf einen erfolgreichen Abschluss der LOAD-Phase hin.
- Abbruchsprotokollsatz (Abort). Dieser Protokollsatz weist darauf hin, dass die LOAD-Phase fehlgeschlagen ist.

- **DELETE-Phase**
 - **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 Löschoption 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 Recovery einer erfolgreichen Ladeoperation wiederholt.

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 Indexbaumstruktur erhalten. Die Anzahl der protokollierten Sätze ist wesentlich geringer als bei einer vollständig protokollierten Einfügung in den Index.

Übersicht zum Laden - Umgebungen mit partitionierten Datenbanken

In einer Mehrpartitionsdatenbank befinden sich große Datenmengen auf vielen verschiedenen Datenbankpartitionen. Mit Verteilungsschlüsseln können Sie die Datenbankpartition festlegen, auf der die einzelnen Datenteile jeweils gespeichert werden sollen. Die Daten müssen *verteilt* werden, bevor sie auf die richtige Datenbankpartition geladen werden können.

Beim Laden von Tabellen in Mehrpartitionsdatenbanken stellt das Dienstprogramm LOAD das folgende Funktionsspektrum zur Verfügung:

- Paralleles Verteilen von Eingabedaten
- Gleichzeitiges Laden von Daten auf entsprechende Datenbankpartitionen
- Übertragen von Daten von einem System an ein anderes System

Das Laden von Daten in eine Mehrpartitionsdatenbank erfolgt in zwei Phasen: In einer *SETUP-Phase* werden Datenbankpartitionsressourcen wie beispielsweise Tabellensperren angefordert, und in einer *LOAD-Phase* werden die Daten in die Datenbankpartitionen geladen. Mit der Option **ISOLATE_PART_ERRS** des Befehls **LOAD** können Sie auswählen, wie Fehler in diesen beiden Phasen jeweils verarbeitet werden und wie sich Fehler in einer oder mehreren der Datenbankpartitionen auf die Ladeoperation für diejenigen Datenbankpartitionen auswirken, die fehlerfrei sind.

Beim Laden von Daten in eine Mehrpartitionsdatenbank können die folgenden Modi verwendet werden:

PARTITION_AND_LOAD

Die Daten werden verteilt (möglicherweise in Parallelverarbeitung) und gleichzeitig auf die entsprechenden Datenbankpartitionen geladen.

PARTITION_ONLY

Die Daten werden verteilt (möglicherweise in Parallelverarbeitung), und die Ausgabe wird in Dateien an einer angegebenen Speicherposition auf jeder Ladedatenbankpartition geschrieben. Jede Datei enthält Partitionskopfdaten, die angeben, wie die Daten über die Datenbankpartitionen verteilt 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 über die Datenbankpartitionen verteilt sind. Der Verteilungsprozess wird übersprungen, und die Daten werden gleichzeitig auf die entsprechenden Datenbankpartitionen geladen.

LOAD_ONLY_VERIFY_PART

Es wird davon ausgegangen, dass die Daten bereits über die Datenbankpartitionen verteilt sind, aber die Datendatei enthält keine Partitionskopfdaten. Der Verteilungsprozess 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 Datenbankpartition befindet. Zeilen, die Datenbankpartitionsverstöße enthalten, werden in eine Speicherauszugsdatei gestellt, sofern der Änderungswert `dumpfile` für den Dateityp angegeben wurde. Andernfalls werden die Zeilen gelöscht. Wenn für eine bestimmte Ladedatenbankpartition Datenbankpartitionsverstöße vorliegen, wird für die betreffende Datenbankpartition eine einzige Warnung in die `LOAD`-Nachrichtendatei geschrieben.

ANALYZE

Es wird eine optimale Verteilungszuordnung mit einer gleichmäßigen Verteilung auf alle Datenbankpartitionen generiert.

Konzepte und Terminologie

Im Zusammenhang mit der Funktionsweise und der Ausführung des Dienstprogramms `LOAD` in einer Umgebung mit partitionierten Datenbanken mit mehreren Datenbankpartitionen 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` wird davon ausgegangen, dass sich die Datendatei auf dieser Datenbankpartition befindet, sofern nicht die Option `CLIENT` des Befehls `LOAD` angegeben ist. Mit der Option `CLIENT` wird angegeben, dass sich die zu ladenden Daten auf einem Client mit Fernverbindung befinden.
- In den Modi `PARTITION_AND_LOAD`, `PARTITION_ONLY` und `ANALYZE` liest der *Agent für Partitionierungsvorbereitung* die Benutzerdaten und teilt sie reihum den *Partitionierungsagenten* zu, die die Daten verteilen. Dieser Prozess findet immer auf der Koordinatorpartition statt. Bei allen Ladeoperationen ist pro Datenbankpartition maximal ein Partitionierungsagent zulässig.

- In den Modi PARTITION_AND_LOAD, LOAD_ONLY und LOAD_ONLY_VERIFY_PART werden auf jeder Ausgabedatenbankpartition *Ladeagenten* ausgeführt. Sie koordinieren das Laden der Daten auf diese Datenbankpartition.
- *Dateiladeagenten* werden während einer Ladeoperation im Modus PARTITION_ONLY auf jeder Ausgabedatenbankpartition ausgeführt. Sie empfangen Daten von Partitionierungsagenten und schreiben sie in eine Datei auf ihrer Datenbankpartition.
- Die Option SOURCEUSEREXIT stellt eine Funktion bereit, über die das Dienstprogramm LOAD ein angepasstes Script bzw. eine angepasste ausführbare Datei ausführen kann. Dieses Script bzw. diese Datei wird im vorliegenden Handbuch als *Benutzerexit* bezeichnet.

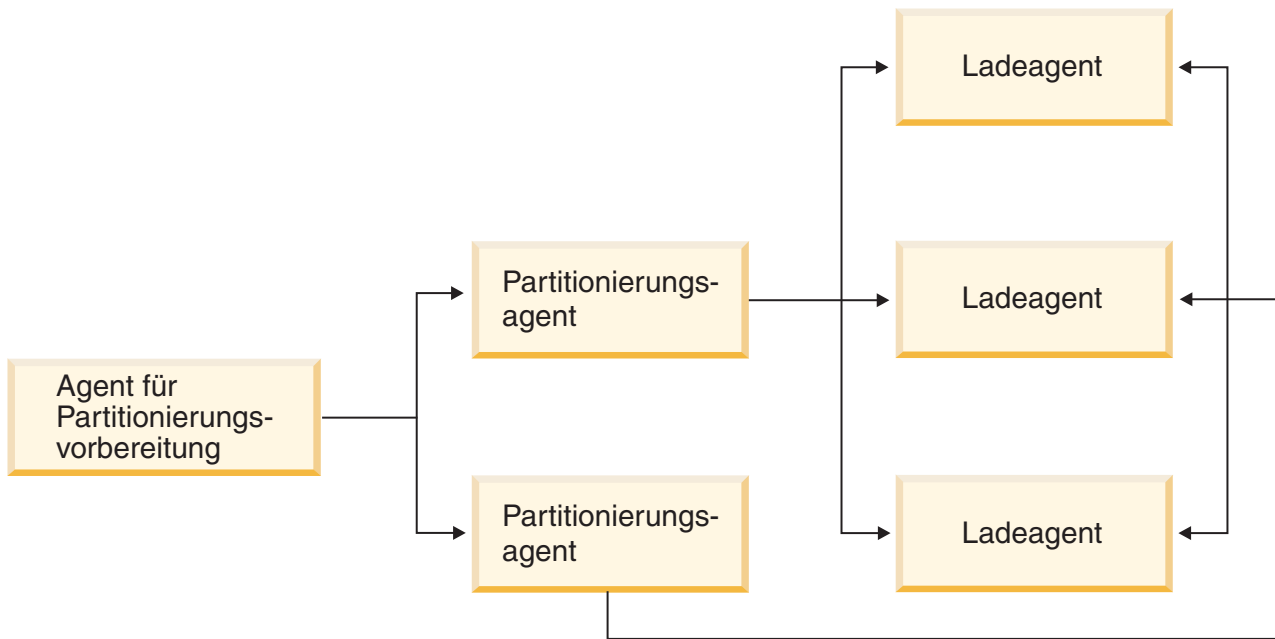


Abbildung 13. Übersicht über das Laden in partitionierte Datenbanken. Die Quelldaten werden durch den Agenten für Partitionierungsvorbereitung gelesen. Ungefähr die Hälfte der Daten wird jeweils an zwei Partitionierungsagenten gesendet, die die Daten verteilen und an eine der drei Datenbankpartitionen senden. Auf den einzelnen Datenbankpartitionen werden die Daten durch den jeweiligen Ladeagenten geladen.

Laden von Daten in einer Umgebung mit partitionierten Datenbanken

Dienstprogramm LOAD zum Laden von Daten in eine Umgebung mit partitionierten Datenbanken verwenden

Vorbereitende Schritte

Bevor Sie eine Tabelle in eine Mehrpartitionsdatenbank laden, sollten Sie die folgenden Punkte beachten:

- Stellen Sie sicher, dass der Konfigurationsparameter **svcename** des Datenbankmanagers und die Profilregistrierdatenbankvariable **DB2COMM** richtig definiert sind. Dieser Schritt ist wichtig, weil das Dienstprogramm LOAD TCP/IP verwendet, um Daten von dem Agenten für die Partitionierungsvorbereitung an die Partitionierungsagenten und von den Partitionierungsagenten an die Ladedatenbankpartitionen zu übertragen.
- Bevor Sie das Dienstprogramm LOAD ausführen, müssen Sie mit der Datenbank, in die die Daten geladen werden sollen, verbunden bzw. in der Lage sein, implizit eine Verbindung zur Datenbank herzustellen.
- Da das Dienstprogramm LOAD eine Anweisung COMMIT absetzt, beenden Sie vor dem Beginn der Ladeoperation alle Transaktionen und heben alle Sperren auf, 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 Datenbankpartition befinden. Ausgenommen sind die folgenden Fälle:
 1. Der Parameter **CLIENT** wurde angegeben. In diesem Fall müssen sich die Daten auf der Clientmaschine befinden.
 2. Der Eingabequellentyp ist CURSOR. Hierbei gibt es keine Eingabedatei.
- Führen Sie den Designadvisor aus, um die jeweils beste Datenbankpartition für die einzelnen Tabellen zu ermitteln. Weitere Informationen finden Sie unter „Designadvisor“ in der Veröffentlichung *Fehlerbehebung und Optimieren der Datenbankleistung*.

Informationen zu diesem Vorgang

Bei der Verwendung des Dienstprogramms LOAD zum Laden von Daten in eine Mehrpartitionsdatenbank gelten die folgenden Einschränkungen:

- Die Eingabedateien für die Ladeoperation dürfen nicht auf einer Bandeinheit gespeichert sein.
- Der Parameter **ROWCOUNT** wird nur dann unterstützt, wenn der Modus ANALYZE verwendet wird.
- Wenn die Zieltabelle über eine Identitätsspalte verfügt, die zur Verteilung notwendig ist, und der Änderungswert **identityoverride** für den Dateityp nicht angegeben ist oder wenn Sie mehrere Datenbankpartitionen verwenden, um die Daten zu verteilen und anschließend zu laden, wird ein größerer Wert als 0 für **SAVECOUNT** im Befehl **LOAD** nicht unterstützt.
- Wenn eine Identitätsspalte einen Teil des Verteilungsschlüssels darstellt, wird nur der Modus PARTITION_AND_LOAD unterstützt.
- Der Modus LOAD_ONLY und der Modus LOAD_ONLY_VERIFY_PART können nicht mit dem Parameter **CLIENT** des Befehls **LOAD** kombiniert werden.
- Der Modus LOAD_ONLY_VERIFY_PART kann nicht mit dem Eingabequellentyp CURSOR verwendet werden.
- Die Modi LOAD_ERRS_ONLY und SETUP_AND_LOAD_ERRS für die Isolation von Verteilungsfehlern 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 Parameter **OUTPUT_DBPARTNUMS** und

PARTITIONING_DBPARTNUMS angegebenen Datenbankpartitionen nicht überlappen. Beispiel: Wenn eine Tabelle auf den Datenbankpartitionen 0 bis 3 definiert ist, kann eine Ladeoperation Daten auf die Datenbankpartitionen 0 und 1 laden, während eine zweite Ladeoperation Daten auf die Datenbankpartitionen 2 und 3 lädt. Wenn die von den Parametern **PARTITIONING_DBPARTNUMS** angegebenen Datenbankpartitionen sich jedoch überlappen, wird für den Ladevorgang automatisch ein Parameterwert **PARTITIONING_DBPARTNUMS** ausgewählt, für den noch kein Unteragent für Partitionierung beim Laden für die Tabelle ausgeführt wird. Sind keine solchen vorhanden, schlägt der Ladevorgang fehl.

- Nur ASC-Dateien (ASCII-Format mit universellen Zeilenbegrenzern) und DEL-Dateien (ASCII-Format ohne universelle Zeilenbegrenzer) können über Tabellen auf mehreren Datenbankpartitionen verteilt werden. PC/IXF-Dateien können nicht verteilt werden, Sie können eine PC/IXF-Datei jedoch in eine über mehrere Datenbankpartitionen verteilte Tabelle laden, indem Sie die Ladeoperation im Modus **LOAD_ONLY_VERIFY_PART** verwenden.

Beispiel

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 Datenbankpartitionen: 0, 1, 2, 3 und 4. Jede Datenbankpartition verfügt über das lokale Verzeichnis `/db2/data/`. Die beiden Tabellen **TABLE1** und **TABLE2** sind auf den Datenbankpartitionen 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 Verteilen und Laden

In diesem Szenario besteht eine Verbindung zu einer Datenbankpartition, auf der die Tabelle **TABLE1** definiert sein kann oder auch nicht. Die Datendatei `load.del` befindet sich im aktuellen Arbeitsverzeichnis dieser Datenbankpartition. Um die Daten aus der Datei `load.del` auf alle Datenbankpartitionen 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 Umgebungen mit partitionierten Datenbanken die Standardwerte verwendet: Der Parameter **MODE** verwendet den Standardwert **PARTITION_AND_LOAD**, die Parameter **OUTPUT_DBPARTNUMS** nehmen alle Datenbankpartitionen als Standardwerte an, auf denen **TABLE1** definiert ist, und **PARTITIONING_DBPARTNUMS** ist standardmäßig die Gruppe der Datenbankpartitionen, die gemäß den Regeln des Befehls **LOAD** ausgewählt wurde, welche zur Auswahl von Datenbankpartitionen gelten, wenn keine angegeben sind.

Um eine Ladeoperation auszuführen, bei der Daten über die Datenbankpartitionen 3 und 4 verteilt werden, setzen Sie den folgenden Befehl ab:

```
LOAD FROM LOAD.DEL of DEL REPLACE INTO TABLE1
PARTITIONED DB CONFIG PARTITIONING_DBPARTNUMS (3,4)
```

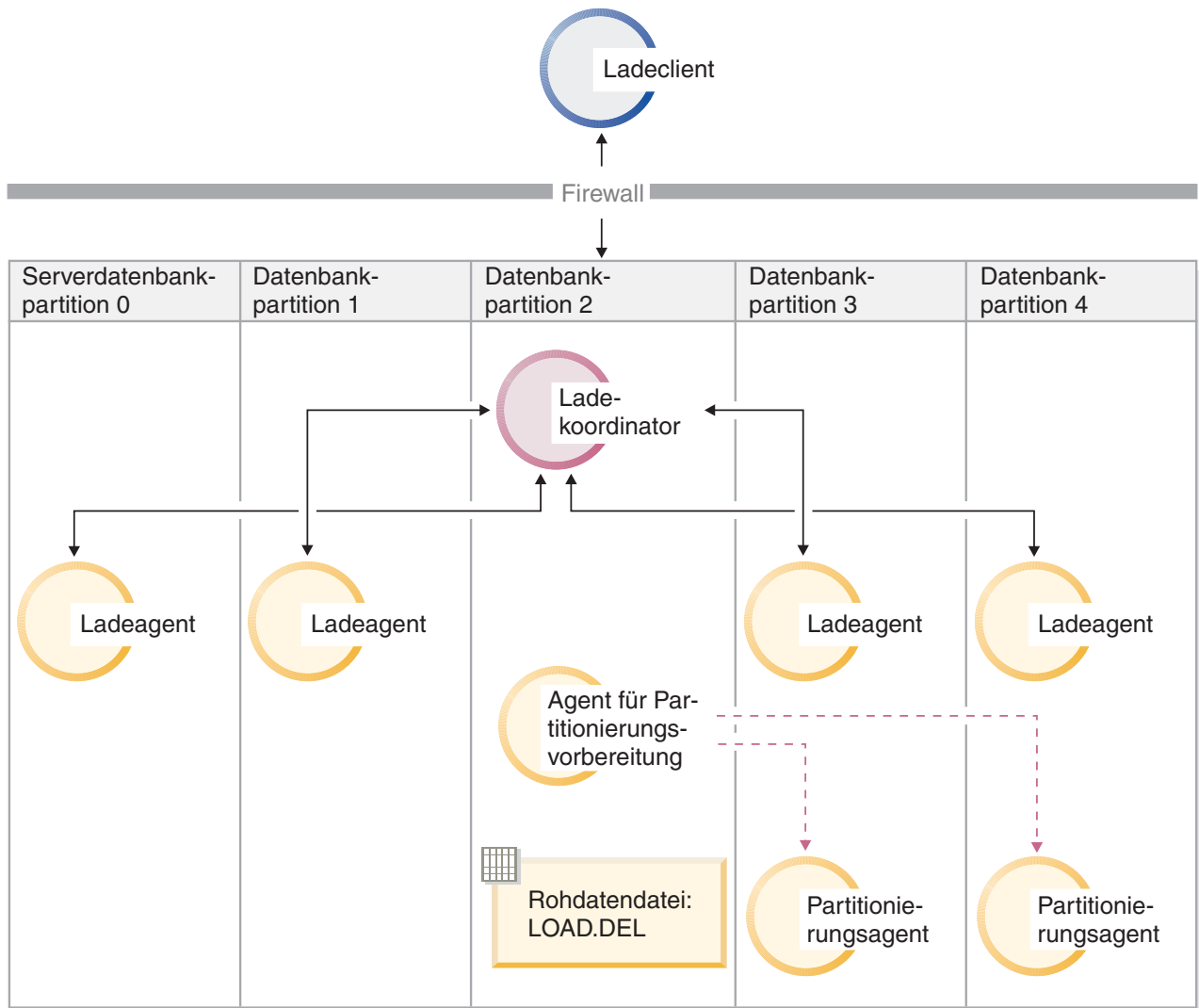


Abbildung 14. Laden von Daten in die Datenbankpartitionen 3 und 4. Diese Abbildung veranschaulicht das Verhalten, das sich bei Ausführung des vorstehenden Befehls ergibt. Die Daten werden in die Datenbankpartitionen 3 und 4 geladen.

Beispiel für Verteilen ohne Laden

In diesem Szenario besteht eine Verbindung zu einer Datenbankpartition, auf der die Tabelle TABLE1 definiert sein kann oder auch nicht. Die Datendatei load.del befindet sich im aktuellen Arbeitsverzeichnis dieser Datenbankpartition. Um die Datei load.del auf alle Datenbankpartitionen, auf denen TABLE1 definiert ist, zu verteilen (jedoch nicht zu laden) und dafür die Datenbankpartitionen 3 und 4 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 /db2/data
PARTITIONING_DBPARTNUMS (3,4)
```

Dieser Befehl bewirkt, dass eine Datei load.del.xxx im Verzeichnis /db2/data auf jeder Datenbankpartition gespeichert wird. Hierbei steht xxx für eine dreistellige Darstellung der Datenbankpartitionsnummer.

Um die Datei load.del auf die Datenbankpartitionen 1 und 3 zu verteilen und hierfür 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 /db2/data
OUTPUT_DBPARTNUMS (1,3)
```

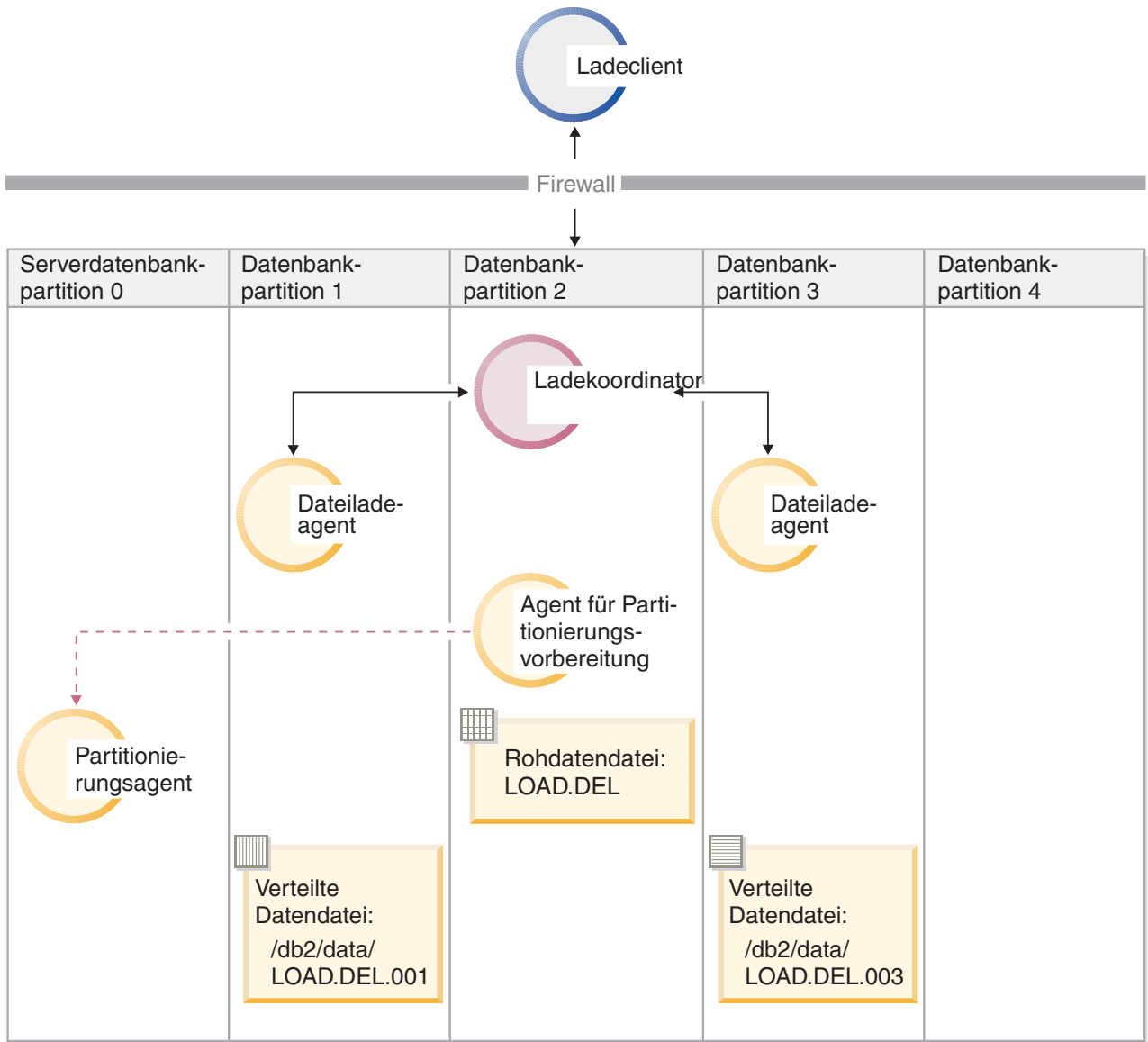


Abbildung 15. Laden von Daten in die Datenbankpartitionen 1 und 3 mithilfe eines Partitionierungsagenten. Diese Abbildung veranschaulicht das Verhalten, das sich bei Ausführung des vorstehenden Befehls ergibt. Die Daten werden auf die Datenbankpartitionen 1 und 3 geladen, wobei ein Partitionierungsagent verwendet wird, der auf Datenbankpartition 0 aktiv ist.

Beispiel für Laden ohne Verteilen

Wenn Sie bereits eine Ladeoperation im Modus **PARTITION_ONLY** ausgeführt haben und die partitionierten Dateien im Verzeichnis `/db2/data` jeder Ladedatenbankpartition auf alle Datenbankpartitionen 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 /db2/data
```

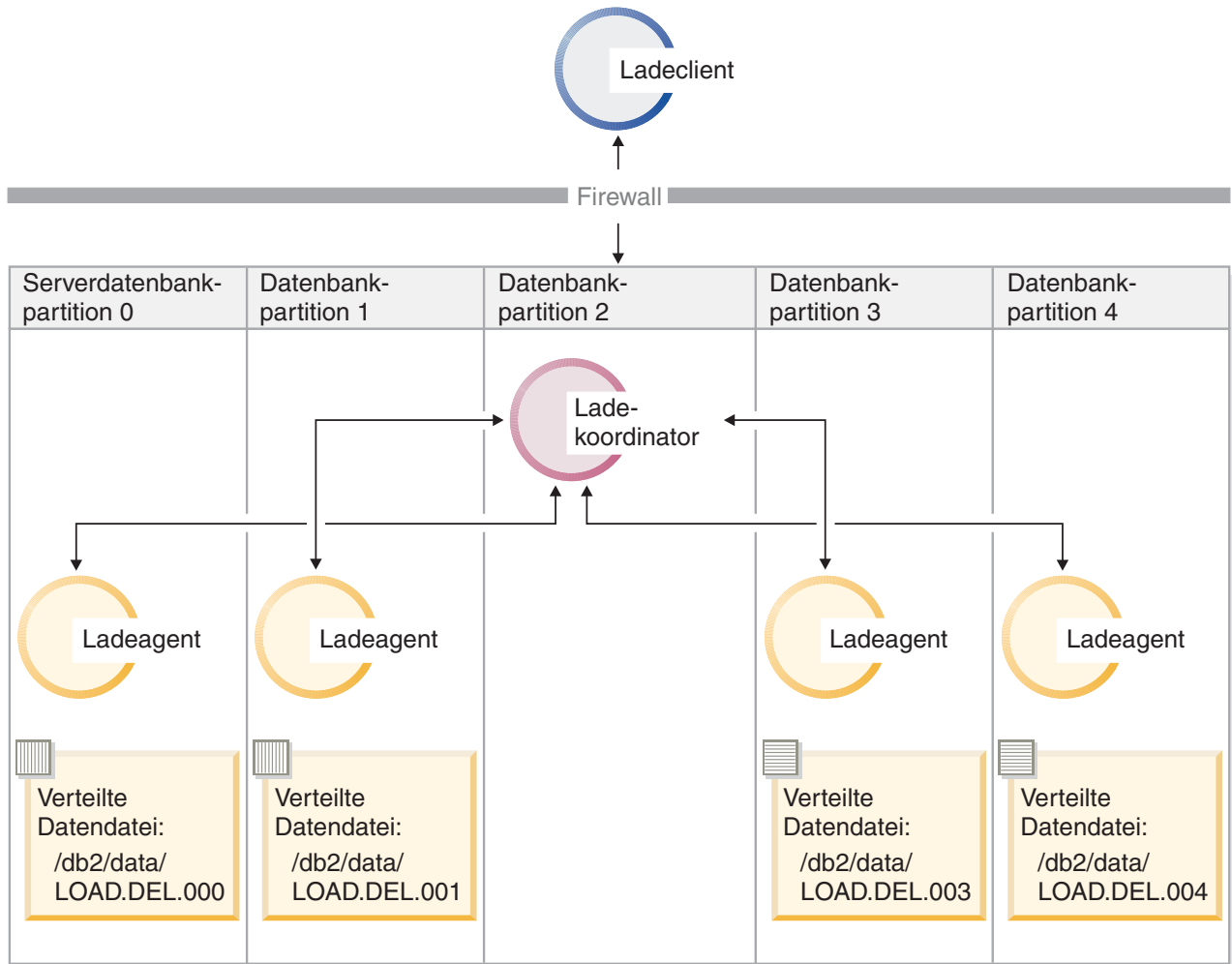


Abbildung 16. Laden von Daten in alle Datenbankpartitionen mit einer bestimmten definierten Tabelle. Diese Abbildung veranschaulicht das Verhalten, das sich bei Ausführung des vorstehenden Befehls ergibt. Die verteilten Daten werden auf alle Datenbankpartitionen geladen, auf denen TABLE1 definiert ist.

Um Daten nur auf Datenbankpartition 4 zu laden, setzen Sie den folgenden Befehl ab:

```
LOAD FROM LOAD.DEL OF DEL REPLACE INTO TABLE1
PARTITIONED DB CONFIG MODE LOAD_ONLY
PART_FILE_LOCATION /db2/data
OUTPUT_DBPARTNUMS (4)
```

Laden von vorverteilten Dateien ohne Kopfdaten für Verteilungszuordnung

Mit dem Befehl **LOAD** können Sie Datendateien ohne Verteilungskopfdaten direkt auf mehrere Datenbankpartitionen laden. Wenn die Datendateien im Verzeichnis /db2/data aller Datenbankpartitionen, auf denen TABLE1 definiert ist, bereits vorhanden sind und ihr Name load.del.xxx lautet (hierbei ist xxx die Datenbankpartitionsnummer), 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 /db2/data
```

Um die Daten nur auf Datenbankpartition 1 zu laden, setzen Sie den folgenden Befehl ab:

```
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 /db2/data
      OUTPUT_DBPARTNUMS (1)
```

Anmerkung: Zeilen, die nicht zu der Datenbankpartition gehören, von der aus sie geladen wurden, werden zurückgewiesen und in die Speicherauszugsdatei gestellt, sofern eine Speicherauszugsdatei angegeben wurde.

Laden von einem fernen Client in eine Mehrpartitionsdatenbank

Um Daten in eine Mehrpartitionsdatenbank aus einer Datei zu laden, die sich auf einem fernen Client befindet, müssen Sie mit dem Parameter **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 dem Parameter **CLIENT** verwendet werden.

Laden über einen Cursor

Wie bei Einzelpartitionsdatenbanken können Sie Daten über einen Cursor in eine Mehrpartitionsdatenbank laden. In diesem Beispiel muss beim Modus **PARTITION_ONLY** und **LOAD_ONLY** der Parameter **PART_FILE_LOCATION** einen vollständig qualifizierten Dateinamen angeben. Dieser Name ist der vollständig qualifizierte Basisdateiname der verteilten Dateien, die auf jeder Ausgabedatenbankpartition erstellt oder geladen werden. Wenn die Zieltabelle LOB-Spalten enthält, können mehrere Dateien mit dem angegebenen Basisnamen erstellt werden.

Um alle Zeilen in der Antwortgruppe der Anweisung **SELECT * FROM TABLE1** auf eine Datei zu verteilen, die sich jeweils auf allen Datenbankpartitionen namens `/db2/data/select.out.xxx` befindet (`xxx` ist die Datenbankpartitionsnummer), damit diese Zeilen später in die Tabelle **TABLE2** geladen werden können, setzen Sie die folgenden Befehle 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 /db2/data/select.out
```

Die durch die vorherigen 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 /db2/data/select.out
```

Hinweise und Tipps für das Laden von Daten in einer Umgebung mit partitionierten Datenbanken: Bevor Sie eine Tabelle in eine Mehrpartitionsdatenbank laden, sollten Sie die folgenden Punkte beachten:

- Machen Sie sich mit den Konfigurationsoptionen des Dienstprogramms LOAD 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 Ladeprozesses beibehalten möchten, darf nur eine einzige Datenbankpartition zum Verteilen verwendet werden. Bei paralleler Verteilung 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, der in einer Mehrpartitionsdatenbank ausgeführt wird, erzwingen, selbst wenn die Ladeoperation (beim Start) feststellt, dass sich einige Ladedatenbankpartitionen 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 LOAD-Konfigurationsoption `STATUS_INTERVAL` kann der Status eines Jobs überwacht werden, der in einer Mehrpartitionsdatenbank ausgeführt wird. 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 Partitionierungsvorbereitung gespeichert. Um den Inhalt dieser Datei während der Ladeoperation anzuzeigen, stellen Sie eine Verbindung zur Koordinatorpartition her und setzen den Befehl **LOAD QUERY** für die Zieltabelle ab.
- Sie können mit besserer Leistung rechnen, wenn sich die am Verteilungsprozess beteiligten Datenbankpartitionen (durch die Option `PARTITIONING_DBPARTNUMS` definiert) von den Ladedatenbankpartitionen (durch die Option `OUTPUT_DBPARTNUMS` definiert) unterscheiden, da dann weniger Konkurrenzsituationen für CPU-Zyklen auftreten. Wenn Daten in eine Mehrpartitionsdatenbank geladen werden, sollte das Dienstprogramm LOAD auf einer Datenbankpartition aufgerufen werden, die weder an der Verteilungs- noch an der 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 Datenbankpartition her und setzen einen Befehl **LOAD QUERY** für die Zieltabelle ab.
- Das Dienstprogramm LOAD wählt nur eine Ausgabedatenbankpartition für die Erfassung von Statistikdaten aus. Mit der Datenbankkonfigurationsoption `RUN_STAT_DBPARTNUM` können Sie die Datenbankpartition angeben.
- Führen Sie vor dem Laden von Daten in eine Mehrpartitionsdatenbank den Designadvisor aus, um die jeweils beste Partition für die einzelnen Tabellen zu ermitteln. Weitere Informationen hierzu finden Sie in „Designadvisor“ in *Fehlerbehebung und Optimieren der Datenbankleistung*.

Fehlerbehebung

Wenn das Dienstprogramm LOAD blockiert, haben Sie folgende Möglichkeiten:

- Mit dem Parameter `STATUS_INTERVAL` kann der Status einer Ladeoperation für Mehrpartitionsdatenbanken 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 auf den einzelnen 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 Fehlnachrichten 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.

Überwachen einer Ladeoperation in einer Umgebung mit partitionierten Datenbanken mit `LOAD QUERY`

Während einer Ladeoperation in einer Umgebung mit partitionierten Datenbanken werden von einigen Ladeprozessen Nachrichtendateien auf den Datenbankpartitionen erstellt, auf denen sie ausgeführt werden.

In den Nachrichtendateien werden alle Informationen, Warnungen und Fehlnachrichten gespeichert, die während der Ausführung der Ladeoperation 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. Der Inhalt der Nachrichtendatei steht erst nach Abschluss der Ladeoperation zur Verfügung.

Sie können während einer Ladeoperation eine Verbindung zu einzelnen Datenbankpartitionen herstellen und den Befehl `LOAD QUERY` für die Zieltabelle absetzen. Wird dieser Befehl über den Befehlszeilenprozessor (CLP) abgesetzt, zeigt er den Inhalt aller Nachrichtendateien an, die sich gegenwärtig für die im Befehl `LOAD QUERY` angegebene Tabelle auf dieser Datenbankpartition befinden.

Beispiel: Die Tabelle `TABLE1` ist auf den Datenbankpartitionen 0 bis 3 in der Datenbank `WSDB` definiert. Sie sind mit Datenbankpartition 0 verbunden und setzen 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 Datenbankpartitionen 0, 1, 2 und 3 umfasst. Außerdem wird ein Partitionierungsagent auf Datenbankpartition 1 und ein Agent für Partitionierungsvorbereitung auf Datenbankpartition 0 ausgeführt.

Datenbankpartition 0 enthält eine Nachrichtendatei für den Agenten für Partitionierungsvorbereitung und eine Nachrichtendatei für den Ladeagenten auf dieser Datenbankpartition. 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
```

Datenbankpartition 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 LOAD-Konfigurationsoption STATUS_INTERVAL generierten Nachrichten werden in der Nachrichtendatei des 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 API '**db2Load**' eingeleitet wird, muss die Nachrichtenoption (piLocalMsgFileName) angegeben werden. Dann werden die Nachrichtendateien vom Server an den Client übertragen und dort gespeichert, damit Sie die Nachrichten anzeigen können.

Bei Ladeoperationen für Mehrpartitionsdatenbanken, die über den Befehlszeilenprozessor (CLP) 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 Mehrpartitionsdatenbank abgeschlossen ist, muss die Option MESSAGES des Befehls LOAD angegeben werden. Bei Verwendung dieser Option werden die Nachrichtendateien auf jeder Datenbankpartition nach Abschluss der Ladeoperation an die Clientmaschine übertragen und dort unter dem Basisnamen in Dateien gespeichert, der in der Option MESSAGES angegeben wurde. Die folgende Liste enthält die Dateinamen, die dem Ladeprozess, der die jeweilige Datei erstellt hat, entsprechen (bei Ladeoperationen für Mehrpartitionsdatenbanken):

Prozesstyp	Dateiname
Ladeagent	<nachrichtendateiname>.LOAD. <dbpartitionsnr>
Partitionierungsagent	<nachrichtendateiname>.PART. <dbpartitionsnr>
Agent für Partitionierungsvorbereitung	<nachrichtendateiname>.PREP. <dbpartitionsnr>

Wenn in der Option MESSAGES beispielsweise der Wert /wsdb/messages/load angegeben ist, lautet der Name der Nachrichtendatei des Ladeagenten auf Datenbankpartition 2 /wsdb/messages/load.LOAD.002.

Anmerkung: Es wird dringend empfohlen, bei Ladeoperationen für Mehrpartitionsdatenbanken, die über den Befehlszeilenprozessor (CLP) eingeleitet werden, die Option MESSAGES zu verwenden.

Fortsetzen, erneutes Starten oder Beenden von Ladeoperationen in einer Umgebung mit partitionierten Datenbanken

Die Arbeitsschritte, die in einer Umgebung mit partitionierten Datenbanken nach fehlgeschlagenen Ladeoperationen ausgeführt werden müssen, hängen davon ab, wann der Fehler aufgetreten ist.

Der Ladeprozess in einer Mehrpartitionsdatenbank besteht aus zwei Phasen:

1. In der *SETUP-Phase* werden Ressourcen auf Datenbankpartitionsebene angefordert. Hierzu gehören z. B. Tabellensperren für Datenbankpartitionen, die für die Ausgabe verwendet werden.

Wenn während der *SETUP-Phase* ein Fehler auftritt, sind in der Regel keine Neustart- oder Beendigungsoperationen erforderlich. Die erforderlichen Maßnahmen hängen von dem Fehlerisolationsmodus ab, der für die fehlgeschlagene Ladeoperation angegeben wurde.

Wenn für die Ladeoperation angegeben wurde, dass Fehler während der *SETUP-Phase* nicht isoliert werden sollen, wird die gesamte Ladeoperation abgebrochen und der Status der Tabelle wird auf allen Datenbankpartitionen mittels Rollback auf den Status zurückgesetzt, der vor der Ladeoperation gültig war.

Wenn für die Ladeoperation angegeben wurde, dass Fehler während der *SETUP-Phase* isoliert werden sollen, wird die Ladeoperation auf den Datenbankpartitionen fortgesetzt, auf denen die *SETUP-Phase* erfolgreich ausgeführt werden konnte. Die Tabelle auf den Datenbankpartitionen, auf denen ein Fehler auftrat, werden mittels Rollback auf den Status zurückgesetzt, der vor der Ladeoperation gültig war. Dies bedeutet, dass eine einzige Ladeoperation in unterschiedlichen Phasen fehlschlagen kann, wenn einige Partitionen während der *SETUP-Phase* und andere während der *LOAD-Phase* fehlschlagen.

2. Die *LOAD-Phase*, während der die Daten formatiert und in Tabellen auf den Datenbankpartitionen geladen werden.

Wenn eine Ladeoperation für eine Mehrpartitionsdatenbank während der *LOAD-Phase* auf mindestens einer Datenbankpartition fehlschlägt, muss ein Befehl `LOAD RESTART` oder ein Befehl `LOAD TERMINATE` abgesetzt werden. Dies ist notwendig, weil das Laden von Daten in eine Mehrpartitionsdatenbank in einer einzigen Transaktion erfolgt.

Sie sollten sich für einen Befehl `LOAD RESTART` entscheiden, wenn Sie die Probleme, die zum Fehlschlagen der Ladeoperation führten, beheben können. Dies spart Zeit. Wenn eine `LOAD RESTART`-Operation eingeleitet wird, wird die Ladeoperation auf allen Datenbankpartitionen dort fortgesetzt, wo sie abgebrochen wurde.

Sie sollten sich für eine `LOAD TERMINATE`-Operation entscheiden, wenn die Tabelle in den Status zurückversetzt werden soll, in der sie sich vor der ursprünglichen Ladeoperation befand.

Vorgehensweise:

Ermitteln des Fehlerzeitpunkts einer Ladeoperation

Wenn Ihre Ladeoperation in einer partitionierten Umgebung fehlschlägt, müssen Sie zuerst ermitteln, auf welchen Partitionen der Fehler aufgetreten ist und in welcher Phase die Verarbeitung fehlschlug. Hierzu müssen Sie die Zusammenfassung für die Partition überprüfen. Wenn der Befehl `LOAD` über den Befehlszeilenprozessor (CLP) eingegeben wurde, wird die Partitionszusammenfassung am Ende des Ladevorgangs (siehe unten) angezeigt. Wenn der Befehl `LOAD` über die API `db2Load` eingegeben wurde, dann ist die Partitionszusammenfassung im Feld `poAgentInfoList` der Struktur `db2PartLoadOut` enthalten.

Wenn für "Agent Typ" für eine bestimmte Partition ein Eintrag "LOAD" vorhanden ist, hat diese Partition die *LOAD-Phase* erreicht. Andernfalls trat der Fehler wäh-

rend der SETUP-Phase auf. Ein negativer SQL-Code gibt an, dass ein Fehler aufgetreten ist. Im folgenden Beispiel schlug die Ladeoperation in Partition 1 während der LOAD-Phase fehl.

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.
.	.	.	.

Fortsetzen, erneutes Starten oder Beenden von fehlgeschlagenen Ladeoperationen

Nur Ladeoperationen mit der Option `ISOLATE_PART_ERRS`, für die `SETUP_ERRS_ONLY` oder `SETUP_AND_LOAD_ERRS` angegeben wurde, können während der SETUP-Phase fehlschlagen. Bei Ladeoperationen, für die auf mindestens einer Ausgabedatenbankpartition ein Fehler in dieser Phase aufgetreten ist, können Sie den Befehl **LOAD REPLACE** oder **LOAD INSERT** eingeben. Verwenden Sie die Option `OUTPUT_DBPARTNUMS`, um nur die Datenbankpartitionen anzugeben, auf denen ein Fehler aufgetreten ist.

Bei Ladeoperationen, für die auf mindestens einer Ausgabedatenbankpartition ein Fehler in der LOAD-Phase aufgetreten ist, müssen Sie den Befehl `LOAD RESTART` oder `LOAD TERMINATE` eingeben.

Bei Ladeoperationen, für die auf mindestens einer Ausgabedatenbankpartition in der SETUP-Phase und in der LOAD-Phase ein Fehler aufgetreten ist, müssen Sie zwei Ladeoperationen ausführen, um die fehlgeschlagene Ladeoperation fortzusetzen. Hierbei muss eine für die Fehler während der SETUP-Phase und eine für die Fehler während der LOAD-Phase verwendet werden. Um eine auf diese Weise fehlgeschlagene Ladeoperation rückgängig zu machen, müssen Sie den Befehl `LOAD TERMINATE` eingeben. Nach der Eingabe des Befehls müssen Sie allerdings alle Partitionen berücksichtigen, da in der Tabelle für keine der Partitionen, auf denen in der SETUP-Phase ein Fehler aufgetreten ist, Änderungen vorgenommen wurden, und weil alle Änderungen auf Partitionen rückgängig gemacht wurden, die während der LOAD-Phase fehlgeschlagen sind.

Beispiel: Die Tabelle `TABLE1` ist auf den Datenbankpartitionen 0 bis 3 in der Datenbank `WSDB` definiert. Der folgende Befehl wird abgesetzt:

```
load from load.del of del insert into table1 partitioned db config
isolate_part_errs setup_and_load_errs
```

Während der SETUP-Phase tritt auf der Ausgabedatenbankpartition 1 ein Fehler auf. Da Fehler während der SETUP-Phase isoliert werden, wird die Ladeoperation fortgesetzt, während der LOAD-Phase tritt jedoch auf Partition 3 ebenfalls ein Fehler auf. Um die Ladeoperation fortzusetzen, sollten Sie die folgenden Befehle eingeben:

```
load from load.del of del replace into table1 partitioned db config
output_dbpartnums (1)
load from load.del of del restart into table1 partitioned db config
isolate_part_errs setup_and_load_errs
```

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.

Migration und Versionskompatibilität

Die Registrierdatenbankvariable **DB2_PARTITIONEDLOAD_DEFAULT** kann verwendet werden, um in einer Datenbank mit mehreren Partitionen zum Ladeverhalten von DB2 Universal Database vor Version 8 zurückzukehren.

Anmerkung: Die Registrierdatenbankvariable **DB2_PARTITIONEDLOAD_DEFAULT** ist veraltet und wird in einem späteren Release möglicherweise entfernt.

Durch die Rückkehr zu dem Verhalten des Befehls **LOAD** von DB2 UDB vor Version 8 in einer Datenbank mit mehreren Partitionen können Sie eine Datei mit gültigen Verteilungskopfdaten in eine einzige Datenbankpartition laden, ohne hierbei zusätzliche Konfigurationsoptionen für partitionierte Datenbanken anzugeben. Hierzu können Sie den Wert von **DB2_PARTITIONEDLOAD_DEFAULT** auf **NO** setzen. Die Verwendung dieser Option ist beispielsweise sinnvoll, wenn Sie Änderungen an vorhandenen Scripts vermeiden möchten, die den Befehl **LOAD** für einzelne Datenbankpartitionen absetzen. Um beispielsweise eine Verteilungsdatei in die Datenbankpartition 3 einer Tabelle zu laden, die zu einer Datenbankpartitionsgruppe mit vier Datenbankpartitionen 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
```

In einer Mehrpartitionsdatenbank findet die Ladeoperation für alle Datenbankpartitionen, in denen die Tabelle definiert ist, statt, wenn keine **LOAD**-Konfigurationsoptionen für das Laden in Mehrpartitionsdatenbanken angegeben werden. Die Eingabedatei muss keine Verteilungskopfdaten enthalten, und die Option **MODE** nimmt standardmäßig den Wert **PARTITION_AND_LOAD** an. Um das Laden für eine einzelne Datenbankpartition auszuführen, muss die Option **OUTPUT_DBPARTNUMS** angegeben werden.

Referenz - Laden in einer partitionierten Umgebung

LOAD-Sitzungen in einer Umgebung mit partitionierten Datenbanken - CLP-Beispiele:

In den folgenden Beispielen wird das Laden von Daten in eine Mehrpartitionsdatenbank dargestellt.

Die Datenbank hat vier Datenbankpartitionen, die von 0 bis 3 nummeriert sind. Die Datenbank **WSDB** ist auf allen Datenbankpartitionen definiert. Die Tabelle **TABLE1** befindet sich in der Standarddatenbankpartitionsgruppe, die ebenfalls auf allen Datenbankpartitionen definiert ist.

Beispiel 1

Um Daten aus der Benutzerdatendatei load.del, die sich auf Datenbankpartition 0 befindet, in die Tabelle TABLE1 zu laden, stellen Sie eine Verbindung zur Datenbankpartition 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 Datenbankpartition ein Ladeagent befand, der jeweils erfolgreich ausgeführt wurde. Außerdem zeigt die Ausgabe, dass auf der Koordinatorpartition ein Agent für Partitionierungsvorbereitung ausgeführt wurde sowie ein Partitionierungsagent auf Datenbankpartition 1. Diese Prozesse wurden erfolgreich mit einem normalen SQL-Rückkehrcode 0 abgeschlossen. Die Zusammenfassung der Statistikdaten ergibt, dass der Agent für Partitionierungsvorbereitung 100.000 Zeilen gelesen hat, dass der Partitionierungsagent 100.000 Zeilen verteilt 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 verteilten Ausgabedateien werden auf allen Ausgabedatenbankpartitionen jeweils im Verzeichnis /db/data gespeichert:

```
load from load.del of del replace into table1 partitioned db config mode
partition_only part_file_location /db/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 jeder Ausgabedatenbankpartition jeweils ein Dateiladeagent aktiv war und dass diese Agenten erfolgreich ausgeführt wurden. Auf der Koordinatorpartition wurde ein Agent für Partitionierungsvorbereitung ausgeführt, und auf Datenbankpartition 1 wurde ein Partitionierungsagent ausgeführt. Die Zusammenfassung der Statistikdaten ergibt, dass der Agent für Partitionierungsvorbereitung 100.000 Zeilen erfolgreich gelesen hat und dass 100.000 Zeilen vom Partitionierungsagenten erfolgreich verteilt 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 /db/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 Ausgabedatenbankpartitionen jeweils erfolgreich ausgeführt wurden und dass insgesamt 100.000 Zeilen durch alle Ladeagenten geladen wurden. Die Zusammenfassung enthält keine Angaben zu verteilten Zeilen, da keine Verteilung erfolgte.

Beispiel 4 - Fehlgeschlagene Ladeoperation

Beim Absetzen des LOAD-Befehls

```
load from load.del of del replace into table1
```

wird eventuell die folgende Ausgabe zurückgegeben, wenn während der Ladeoperation auf einer der Ladedatenbankpartitionen 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 Zusammenfassung für die Datenbankpartitionen besagt, dass auf Datenbankpartition 1 nicht genügend Speicherbereich vorhanden war. Wird zusätzlicher Speicherbereich zu den Containern des Tabellenbereichs auf Datenbankpartition 1 hinzugefügt, können Sie die Ladeoperation wie folgt erneut starten:

```
load from load.del of del restart into table1
```

LOAD-Konfigurationsoptionen für Umgebungen mit partitionierten Datenbanken:

MODE X

Gibt den Modus an, in dem die Ladeoperation beim Laden einer Mehrpartitionsdatenbank stattfindet. PARTITION_AND_LOAD ist der Standardwert. Gültige Werte:

- PARTITION_AND_LOAD. Die Daten werden verteilt (möglicherweise in Parallelverarbeitung) und gleichzeitig auf die entsprechenden Datenbankpartitionen geladen.
- PARTITION_ONLY. Die Daten werden verteilt (möglicherweise in Parallelverarbeitung), und die Ausgabe wird in Dateien an einer angegebenen

Speicherposition auf jeder Ladedatenbankpartition geschrieben. Bei anderen Dateitypen als CURSOR lautet das Format des Namens der Ausgabedatei in der betreffenden Datenbankpartition `dateiname.xxx`, wobei `dateiname` der im Befehl LOAD angegebene Name der Eingabedatei und `xxx` die dreistellige Datenbankpartitionsnummer ist. Beim Dateityp CURSOR wird der Name der Ausgabedatei auf jeder Datenbankpartition durch die Option `PART_FILE_LOCATION` festgelegt. Der Abschnitt zur Option `PART_FILE_LOCATION` enthält weitere Informationen dazu, wie die Position der Verteilungsdatei für die betreffende Datenbankpartition angegeben wird.

Anmerkung:

1. Dieser Modus kann nicht für CLI-Ladeoperationen verwendet werden.
 2. Enthält die Tabelle eine Identitätsspalte, die für die Verteilung erforderlich ist, wird dieser Modus nicht unterstützt, es sei denn, der Änderungswert `identityoverride` für den Dateityp wird angegeben.
 3. Verteilungsdateien, die für den Dateityp CURSOR generiert werden, sind nicht mit verschiedenen DB2-Releases kompatibel. Dies bedeutet, dass Verteilungsdateien vom Dateityp CURSOR, die in einem Vorgängerrelease generiert wurden, nicht mit dem Modus `LOAD_ONLY` geladen werden können. Ebenso gilt, dass Verteilungsdateien vom Dateityp CURSOR, die im aktuellen Release generiert wurden, in einem zukünftigen Release nicht mit dem Modus `LOAD_ONLY` geladen werden können.
- `LOAD_ONLY`. Es wird davon ausgegangen, dass die Daten bereits verteilt sind. Der Verteilungsprozess wird übersprungen, und die Daten werden gleichzeitig auf die entsprechenden Datenbankpartitionen geladen. Bei anderen Dateitypen als CURSOR muss das Format des Namens der Eingabedatei für die betreffende Datenbankpartition `dateiname.xxx` lauten, wobei `dateiname` der im Befehl LOAD angegebene Name der Datei und `xxx` die dreistellige Datenbankpartitionsnummer ist. Beim Dateityp CURSOR wird der Name der Eingabedatei auf jeder Datenbankpartition durch die Option `PART_FILE_LOCATION` festgelegt. Der Abschnitt zur Option `PART_FILE_LOCATION` enthält weitere Informationen dazu, wie die Position der Verteilungsdatei für die betreffende Datenbankpartition angegeben wird.

Anmerkung:

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 Verteilung erforderlich ist, wird dieser Modus nicht unterstützt, es sei denn, der Änderungswert `identityoverride` für den Dateityp wird angegeben.
- `LOAD_ONLY_VERIFY_PART`. Es wird davon ausgegangen, dass die Daten bereits verteilt sind, aber die Datendatei enthält keine Partitionskopfdaten. Der Verteilungsprozess 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 Datenbankpartition befindet. Zeilen, die Datenbankpartitionsverstöße enthalten, werden in eine Speicherauszugsdatei gestellt, sofern der Änderungswert `dumpfile` für den Dateityp angegeben wurde. Andernfalls werden die Zeilen gelöscht. Wenn für eine bestimmte Ladedata-

tenbankpartition Datenbankpartitionsverstöße vorliegen, wird für die betreffende Datenbankpartition eine einzige Warnung in die LOAD-Nachrichtendatei geschrieben. Das Format des Namens der Eingabedatei für die betreffende Datenbankpartition muss `dateiname.xxx` lauten, wobei `dateiname` der im Befehl LOAD angegebene Name der Datei und `xxx` die dreistellige Datenbankpartitionsnummer ist. Der Abschnitt zur Option `PART_FILE_LOCATION` enthält weitere Informationen dazu, wie die Position der Verteilungsdatei für die betreffende Datenbankpartition angegeben wird.

Anmerkung:

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 Verteilung erforderlich ist, wird dieser Modus nicht unterstützt, es sei denn, der Änderungswert `identityoverride` für den Dateityp wird angegeben.
- **ANALYZE.** Es wird eine optimale Verteilungszuordnung mit einer gleichmäßigen Verteilung auf alle Datenbankpartitionen generiert.

PART_FILE_LOCATION X

In den Modi `PARTITION_ONLY`, `LOAD_ONLY` und `LOAD_ONLY_VERIFY_PART` kann mit diesem Parameter die Position der verteilten Dateien angegeben werden. Diese Position muss auf jeder Datenbankpartition 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 verteilten 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 verteilten Dateien, die auf jeder Ausgabedatenbankpartition erstellt werden (beim Modus `PARTITION_ONLY`), bzw. die Position der Dateien, aus denen für jede Datenbankpartition 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 verteilten Dateien verwendet, falls diese Option nicht angegeben ist.

OUTPUT_DBPARTNUMS X

`X` ist eine Liste mit Datenbankpartitionsnummern. Die Datenbankpartitionsnummern geben die Datenbankpartitionen an, auf denen die Ladeoperation ausgeführt werden soll. Die Datenbankpartitionsnummern müssen eine Untermenge der Datenbankpartitionen sein, auf denen die Tabelle definiert ist. Standardmäßig werden 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 Datenbankpartitionsnummern, die im Verteilungsprozess 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 für den Verteilungsprozess angegebenen Datenbankpartitio-

nen können sich von den Datenbankpartitionen, die geladen werden, unterscheiden. Wird die Option `PARTITIONING_DBPARTNUMS` nicht angegeben, ermittelt das Dienstprogramm `LOAD`, wie viele Datenbankpartitionen erforderlich sind und welche Datenbankpartitionen im Hinblick auf eine optimale Leistung verwendet werden.

Falls der Änderungswert `anyorder` für den Dateityp im Befehl `LOAD` nicht angegeben wird, wird in der `LOAD`-Sitzung nur ein Partitionierungsagent verwendet. Des Weiteren wird - wenn nur eine Datenbankpartition für die Option `OUTPUT_DBPARTNUMS` angegeben wird oder die Koordinatorpartition der Ladeoperation kein Element von `OUTPUT_DBPARTNUMS` ist - die Koordinatorpartition der Ladeoperation im Verteilungsprozess verwendet. Andernfalls wird die erste in `OUTPUT_DBPARTNUMS` angegebene Datenbankpartition (nicht die Koordinatorpartition) im Verteilungsprozess verwendet.

Ist der Änderungswert `anyorder` für den Dateityp angegeben, wird die Anzahl der im Verteilungsprozess verwendeten Datenbankpartitionen wie folgt ermittelt: (Anzahl der Partitionen in `OUTPUT_DBPARTNUMS` geteilt durch 4) plus 1.

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 Datenbankpartitionen 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 `SETUP`-Phase auf einer Datenbankpartition auftreten (beispielsweise Fehler beim Zugriff auf eine Datenbankpartition oder Probleme beim Zugriff auf einen Tabellenbereich bzw. auf eine Tabelle auf einer Datenbankpartition), wird die Ladeoperation auf der Datenbankpartition mit dem Fehler gestoppt, auf den übrigen Datenbankpartitionen jedoch fortgesetzt. Fehler, die während des Ladens von Daten auf einer Datenbankpartition auftreten, bewirken das Fehlschlagen der gesamten Operation.
- `LOAD_ERRS_ONLY`. Fehler, die während der `SETUP`-Phase auf einer Datenbankpartition auftreten, bewirken das Fehlschlagen der gesamten Ladeoperation. Wenn beim Laden der Daten ein Fehler auftritt, stoppt die Ladeoperation für die Datenbankpartition, auf der der Fehler aufgetreten ist. Die Ladeoperation wird für die verbleibenden Datenbankpartitionen fortgesetzt, bis ein Fehler auftritt oder bis alle Daten geladen wurden. Die neu geladenen Daten sind auf allen Datenbankpartitionen so lange nicht sichtbar, bis eine `LOAD RESTART`-Operation ausgeführt und erfolgreich beendet wird.

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 Datenbankpartitionsebene während der `SETUP`-Phase oder beim Laden von Daten dazu, dass die Verarbeitung nur auf den betroffenen Datenbankpartitionen gestoppt wird. Wie im Modus `LOAD_ERRS_ONLY` sind die neu geladenen Daten - falls beim Laden der Daten Partitionsfehler auftreten - auf

allen Datenbankpartitionen so lange nicht sichtbar, bis eine LOAD RESTART-Operation ausgeführt und erfolgreich beendet 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 Ladeoperation insgesamt 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 Registrierdatenbankvariable **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-Variablen **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 bei der Ein-/Ausgabe nicht auf ein Abschneiden hin überprüft.

MAP_FILE_INPUT X

X gibt den Namen der Eingabedatei für die Verteilungszuordnung an. Dieser Parameter muss angegeben werden, wenn die Verteilungszuordnung angepasst wurde, da er auf die Datei verweist, in der sich die angepasste Verteilungszuordnung befindet. Eine angepasste Verteilungszuordnung kann erstellt werden, indem die Zuordnung mit dem Programm **db2gpmap** 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 Verteilungszuordnung. Die Ausgabedatei wird auf der Datenbankpartition erstellt, die den Befehl LOAD absetzt, wobei davon ausgegangen wird, dass diese Datenbankpartition zu der Datenbankpartitionsgruppe gehört, in der die Partitionierung stattfindet. Wird der Befehl LOAD auf einer Datenbankpartition aufgerufen, die nicht an der Partitionierung beteiligt ist (wie von PARTITIONING_DBPARTNUMS definiert), wird die Ausgabedatei auf der ersten Datenbankpartition erstellt, die mit dem Parameter PARTITIONING_DBPARTNUMS definiert ist. Betrachten Sie das folgende Setup einer Umgebung mit partitionierten Datenbanken:

```
1 serv1 0
2 serv1 1
3 serv2 0
4 serv2 1
5 serv3 0
```

Bei Ausführung des folgenden LOAD-Befehls auf Server 3 (serv3) wird die Verteilungszuordnung auf Server 1 (serv1) erstellt.

```
LOAD FROM datei OF ASC METHOD L ( ...) INSERT INTO tabelle CONFIG
MODE ANALYZE PARTITIONING_DBPARTNUMS(1,2,3,4)
MAP_FILE_OUTPUT '/home/db2user/distribution.map'
```

Dieser Parameter sollte verwendet werden, wenn der Modus ANALYZE angegeben wird. Es wird eine optimale Verteilungszuordnung mit einer gleichmäßigen Verteilung auf alle Datenbankpartitionen generiert. Falls dieser Parameter nicht angegeben und der Modus ANALYZE 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 Änderungswert `reclen` für den Dateityp im Befehl LOAD angegeben wird. Bei Angabe dieser Option wird jeder Datensatz daraufhin geprüft, ob er ein Zeilenvorschubzeichen enthält. Die Satzlänge, wie im Änderungswert `reclen` für den Dateityp angegeben, wird außerdem geprüft.

DISTFILE X

Wird diese Option angegeben, generiert das Dienstprogramm LOAD eine Datenbankpartitionsverteilungsdatei mit dem angegebenen Namen. Die Datenbankpartitionsverteilungsdatei enthält 32.768 Integers: ein Integer für jeden Eintrag in der Verteilungszuordnung der Zieltabelle. Jedes Integer in der Datei stellt die Anzahl der Zeilen in den geladenen Eingabedateien dar, für die zum entsprechenden Eintrag der Verteilungszuordnung ein Hashverfahren ausgeführt wird. Diese Informationen können Ihnen nicht nur dabei helfen, ungleiche Verteilungen in Ihren Daten zu ermitteln, sondern auch zu entscheiden, ob mithilfe des Modus ANALYZE des Dienstprogramms eine neue Verteilungszuordnung 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: Bei Angabe dieser Option wird maximal ein Partitionierungsagent für die Ladeoperation verwendet. Selbst wenn Sie explizit mehrere Partitionierungsagenten anfordern, wird nur einer verwendet.

OMIT_HEADER

Gibt an, dass die Verteilungsdatei keine Kopfdaten für die Verteilungszuordnung enthalten soll. Wenn diese Option nicht angegeben wird, werden Kopfdaten generiert.

RUN_STAT_DBPARTNUM X

Falls der Parameter STATISTICS USE PROFILE im Befehl LOAD angegeben ist, werden 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 Statistikdaten auf der ersten Datenbankpartition in der Liste der Ausgabedatenbankpartitionen erfasst.

Referenz - Laden

LOAD

Lädt Daten in eine DB2-Tabelle.

Die Daten können auf dem Server in Form einer Datei, eines Bands oder einer benannten Pipe gespeichert sein. Ist das Attribut COMPRESS für die Tabelle auf YES gesetzt, werden die geladenen Daten auf allen Daten- und Datenbankpartitionen komprimiert, für die in der Tabelle ein Wörterbuch existiert, einschließlich der Daten im XML-Speicherobjekt der Tabelle.

Quick Link zu „Änderungswerte für Dateityp für das Dienstprogramm LOAD“ auf Seite 263.

Einschränkungen

Das Dienstprogramm LOAD unterstützt das Laden von Daten auf Hierarchieebene nicht. Das Dienstprogramm LOAD ist mit Bereichsclustertabellen nicht kompatibel. Das Dienstprogramm LOAD unterstützt den Parameter NOT LOGGED INITIALLY für die Anweisung CREATE TABLE oder ALTER TABLE nicht.

Geltungsbereich

Dieser Befehl kann in einer einzigen Anforderung für mehrere Datenbankpartitionen abgesetzt werden.

Berechtigung

Eine der folgenden Berechtigungen:

- DATAACCESS
- Berechtigung LOAD für die Datenbank und folgende Zugriffsrechte:
 - 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.
- Um Daten in eine Tabelle laden zu können, die geschützte Spalten enthält, muss die Berechtigungs-ID der Sitzung direkt oder indirekt über ein Gruppe oder Rolle über LBAC-Berechtigungs-nachweise verfügen, die Schreibzugriff auf alle geschützten Spalten in der Tabelle zulassen. Andernfalls schlägt das Laden fehl, und es wird ein Fehler (SQLSTATE 5U014) zurückgegeben.

- Um Daten in eine Tabelle laden zu können, die geschützte Zeilen enthält, muss für die Berechtigungs-ID der Sitzung ein Sicherheitskennsatz vorliegen, der folgenden Kriterien entspricht:
 - Der Sicherheitskennsatz ist Teil der Sicherheitsrichtlinie, die die Tabelle schützt.
 - Der Sicherheitskennsatz wurde der Berechtigungs-ID der Sitzung direkt oder indirekt über eine Gruppe oder Rolle für Schreibzugriff oder für alle Zugriffe erteilt.

Liegt für die Berechtigungs-ID der Sitzung kein derartiger Sicherheitskennsatz vor, schlägt das Laden fehl und es wird ein Fehler (SQLSTATE 5U014) zurückgegeben. Der Sicherheitskennsatz schützt geladene Zeilen, wenn die LBAC-Berechtigungs-nachweise der Berechtigungs-ID der Sitzung einen Schreibzugriff auf den Sicherheitskennsatz, der die jeweilige Zeile in den Daten schützt, nicht zulassen. Dies geschieht jedoch nicht, wenn die Sicherheitsrichtlinie zum Schutz der Tabelle mit der Option RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL der Anweisung CREATE SECURITY POLICY erstellt wurde. In diesem Fall schlägt das Laden fehl, und es wird ein Fehler (SQLSTATE 42519) zurückgegeben.

Wenn Sie Daten in eine Tabelle mit geschützten Zeilen laden, enthält die Zieltabelle eine einzige Spalte mit dem Datentyp DB2SECURITYLABEL. Wenn die Eingabedatenzeile keinen Wert für diese Spalte enthält, wird die Zeile zurückgewiesen, es sei denn, der Ladebefehl wurde mit dem Änderungswert `usedefaults` für den Dateityp angegeben. Wurde `usedefaults` im Ladebefehl angegeben, wird der Sicherheitskennsatz, über den Sie für den Schreibzugriff verfügen, aus der Sicherheitsrichtlinie, die die Tabelle schützt, verwendet. Wenn Sie keinen Sicherheitskennsatz für den Schreibzugriff haben, wird die Zeile zurückgewiesen und mit der Verarbeitung der nächsten Zeile fortgefahren.

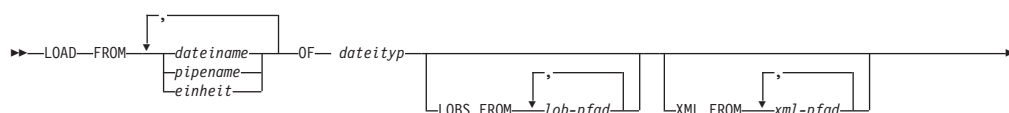
- Bei Angabe der Option REPLACE muss die Berechtigungs-ID der Sitzung über die Berechtigung zum Löschen der Tabelle verfügen.
- Bei Angabe der Option LOCK WITH FORCE ist die Berechtigung SYSADM erforderlichlich.

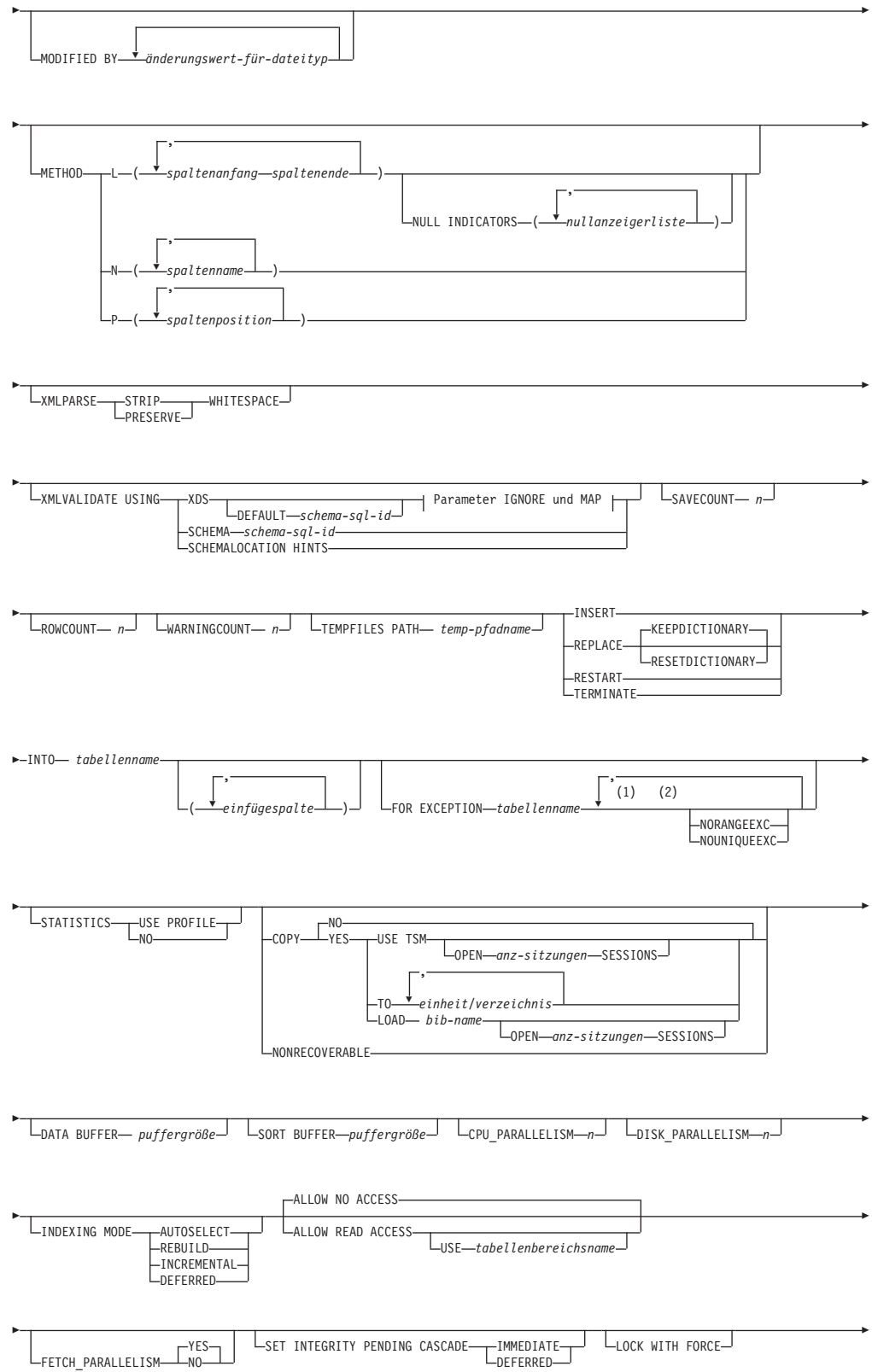
Da der Instanzeigner auch Eigner aller LOAD-Prozesse (und im Allgemeinen auch aller DB2-Serverprozesse) ist, und all diese Prozesse die Kennung des Instanzeigners verwenden, um auf benötigte Dateien zuzugreifen, muss der Instanzeigner über Lesezugriff auf Eingabedatendateien verfügen. Diese Eingabedatendateien müssen für den Instanzeigner lesbar sein (unabhängig davon, wer den Befehl aufruft).

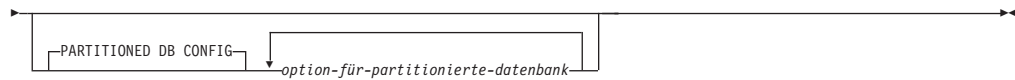
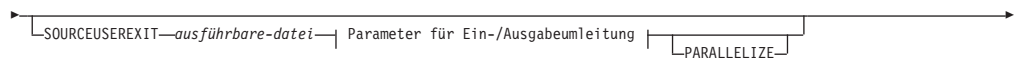
Erforderliche Verbindung

Instanz. Eine explizite Zuordnung ist nicht erforderlich. Wenn eine Verbindung zur Datenbank hergestellt wurde, wird versucht, eine implizite Zuordnung zur lokalen Instanz vorzunehmen.

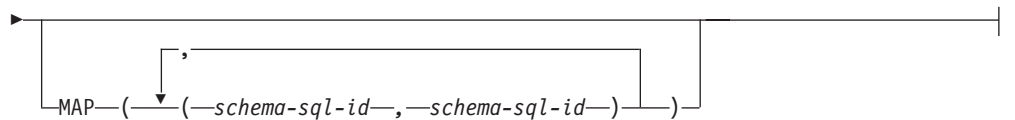
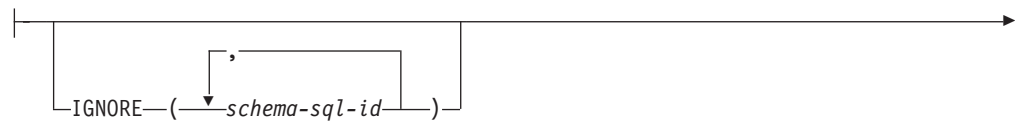
Befehlssyntax



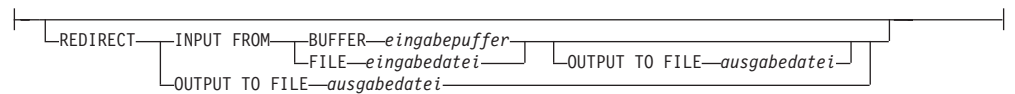




Parameter IGNORE und MAP:



Parameter für Ein-/Ausgabeumleitung:



Anmerkungen:

- 1 Diese Schlüsselwörter können in einer beliebigen Reihenfolge vorliegen.
- 2 Jedes Schlüsselwort darf nur einmal vorkommen.

Befehlsparameter

`FROM` *dateiname* | *pipename* | *einheit*

Anmerkung:

- Werden die Daten mit dem Befehl **EXPORT** und der Prozedur `ADMIN_CMD` in eine Datei exportiert, ist die ID des abgeschirmten Benutzers Eigner der Datendatei. Auf diese Datei kann der Instanzeigner im Allgemeinen nicht zugreifen. Zum Ausführen des Befehls **LOAD** über den CLP oder die Prozedur `ADMIN_CMD` muss ein Zugriff auf die Datendatei über die ID des Instanzeigners möglich sein. Dem Instanzeigner muss deshalb mit Grant Lesezugriff auf die Datendatei erteilt werden.
- 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 eine logische Datei angesehen, wenn sie alle durch einen Aufruf des Befehls **EXPORT** erstellt werden.)
- Beim Laden von XML-Daten in Tabellen in einer Umgebung mit partitionierten Datenbanken müssen die XML-Datendateien über Lesezugriff auf alle Datenbankpartitionen verfügen, in denen das Laden stattfindet.

OF *dateityp*

Gibt das Format der Daten an:

- ASC (ASCII-Format mit universellen Zeilenbegrenzern)
- DEL (ASCII-Format ohne universelle Zeilenbegrenzer)
- IXF (Integration Exchange Format, PC-Version) ist ein Binärformat, das ausschließlich von DB2-Datenbanken verwendet wird.
- CURSOR (ein Cursor, der für eine Anweisung SELECT oder VALUES deklariert wird).

Anmerkung: Wenn Sie XML-Daten mit dem Dateityp CURSOR in eine Tabelle in einer verteilten Datenbankumgebung laden, werden die Modi PARTITION_ONLY und LOAD_ONLY nicht unterstützt.

LOBS FROM *lob-pfad*

Der Pfad zu den Datendateien mit LOB-Werten, die geladen werden sollen. Der Pfad muss mit einem Schrägstrich enden. Die Namen der LOB-Datendateien werden in der Hauptdatendatei (ASC, DEL oder IXF) in der Spalte gespeichert, die in die LOB-Spalte geladen wird. Es können maximal 999 Pfade angegeben werden. Dadurch wird implizit das Verhalten **LOBSINFILE** aktiviert.

Diese Option wird ignoriert, wenn sie zusammen mit dem Dateityp CURSOR angegeben wird.

MODIFIED BY *änderungswert-für-dateityp*

Gibt Optionen für Änderungswerte für Dateityp an. Siehe „Änderungswerte für Dateityp für das Dienstprogramm LOAD“ auf Seite 263.

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. Die Nummerierung beginnt bei 1. Diese Methode kann nur mit ASC-Dateien verwendet werden und ist die einzige gültige Methode für diesen Dateityp.

NULL INDICATORS *nullanzeigerliste*

Diese Option kann nur verwendet werden, wenn der Parameter **METHOD L** angegeben wird (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 einen Eintrag für jedes Datenfeld enthalten, das im Parameter **METHOD L** definiert wird. Eine Spaltennummer 0 gibt an, dass das entsprechende Datenfeld immer Daten enthält.

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

Das Zeichen für den Nullanzeiger kann über die Option **MODIFIED BY** geändert werden.

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. 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. Diese Methode kann nur bei den Dateitypen IXF, DEL oder CURSOR verwendet werden und ist die einzige zulässige Methode für den Dateityp DEL.

XML FROM *xml-pfad*

Gibt mindestens einen Pfad mit XML-Dateien an. XML-Datenkennungen (XDSs) sind in der Hauptdatendatei (ASC, DEL oder IXF) in der Spalte enthalten, die in die XML-Spalte geladen wird.

XMLPARSE

Gibt an, wie die XML-Dokumente syntaktisch analysiert werden. Wenn diese Option nicht angegeben ist, wird das Parsingverhalten für XML-Dokumente durch den Wert des Sonderregisters CURRENT XMLPARSE OPTION festgelegt.

STRIP WHITESPACE

Gibt an, dass Leerzeichen bei der Syntaxanalyse des XML-Dokuments entfernt werden sollen.

PRESERVE WHITESPACE

Gibt an, dass Leerzeichen bei der Syntaxanalyse des XML-Dokuments nicht entfernt werden sollen.

XMLVALIDATE

Gibt an, dass XML-Dokumente anhand eines Schemas geprüft werden sollen (soweit möglich).

USING XDS

XML-Dokumente werden anhand des XML-Schemas geprüft, das über die XDS in der Hauptdatendatei angegeben ist. Wenn die Option **XMLVALIDATE** mit der Klausel **USING XDS** aufgerufen wird, wird das für die Prüfung verwendete Schema standardmäßig durch das SCH-Attribut der XDS festgelegt. Enthält die XDS kein SCH-Attribut, wird eine Schemaprüfung nur durchgeführt, wenn die Klausel **DEFAULT** ein Standardschema angibt.

Mit den Klauseln **DEFAULT**, **IGNORE** und **MAP** kann das Verhalten der Schemafestlegung geändert werden. Diese optionalen Klauseln werden nicht auf einander, sondern direkt auf die XDS-Spezifikationen angewendet. Wird ein Schema beispielsweise ausgewählt, weil es von der Klausel **DEFAULT** angegeben wurde, so wird dieses Schema nicht ignoriert, wenn es auch von der Klausel **IGNORE** angegeben wird. Wird ein Schema ausgewählt, weil es als erster Teil eines Zuordnungspaares in der Klausel **MAP** angegeben wurde, so

wird dieses Schema nicht erneut zugeordnet, wenn es auch im zweiten Teil des Zuordnungspaars einer anderen Klausel **MAP** angegeben ist.

USING SCHEMA *schema-sql-id*

XML-Dokumente werden anhand des XML-Schemas mit der angegebenen SQL-ID geprüft. In diesem Fall wird das Attribut SCH der XDS für alle XML-Spalten ignoriert.

USING SCHEMALOCATION HINTS

XML-Dokumente werden anhand des XML-Schemas geprüft, das über die Angaben zur XML-Schemaposition in den XML-Quelldokumenten angegeben ist. Wenn in dem XML-Dokument kein Attribut für die Schemaposition gefunden wird, findet keine Prüfung statt. Bei Angabe der Klausel **USING SCHEMALOCATION HINTS** wird das SCH-Attribut der XML-Datenkennung (XDS) für alle XML-Spalten ignoriert.

Siehe hierzu die Beispiele im unten stehenden Abschnitt zur Option **XMLVALIDATE**.

IGNORE *schema-sql-id*

Diese Option kann nur verwendet werden, wenn der Parameter **USING XDS** angegeben wird. Die Klausel **IGNORE** gibt eine Liste mit mindestens einem Schema an, das ignoriert werden soll, wenn es durch ein SCH-Attribut identifiziert wird. Wenn in der XML-Datenkennung ein SCH-Attribut für ein geladenes XML-Dokument enthalten ist und das von dem SCH-Attribut identifizierte Schema in der Liste der zu ignorierenden Schemata enthalten ist, wird für das geladene XML-Dokument keine Schemaprüfung durchgeführt.

Anmerkung:

Wenn in der Klausel **IGNORE** ein Schema angegeben ist, darf dieses nicht zusätzlich im linken Teil eines Schemapaars in der Klausel **MAP** enthalten sein.

Die Klausel **IGNORE** gilt nur für die XDS. Ein von der Klausel **MAP** zugeordnetes Schema wird anschließend nicht ignoriert, wenn es von der Klausel **IGNORE** angegeben wird.

DEFAULT *schema-sql-id*

Diese Option kann nur verwendet werden, wenn der Parameter **USING XDS** angegeben wird. Das über die Klausel **DEFAULT** angegebene Schema gibt an, welches Schema für die Prüfung verwendet werden soll, wenn die XML-Datenkennung (XDS) eines geladenen XML-Dokuments kein SCH-Attribut enthält, das ein XML-Schema identifiziert.

Die Klausel **DEFAULT** hat Vorrang vor den Klauseln **IGNORE** und **MAP**. Wenn eine XDS die Klausel **DEFAULT** erfüllt, werden die Angaben **IGNORE** und **MAP** ignoriert.

MAP *schema-sql-id*

Diese Option kann nur verwendet werden, wenn der Parameter **USING XDS** angegeben wird. Geben Sie über die Klausel **MAP** alternative Schemata an, die anstelle der Schemata verwendet werden sollen, die im SCH-Attribut einer XML-Datenkennung (XDS) für jedes geladene XML-Dokument angegeben sind. Die Klausel **MAP** gibt eine Liste mit mindestens einem Schemapaar an, in der jedes Paar eine Zuordnung eines Schemas zu einem ande-

ren Schema darstellt. Das erste Schema in dem Zuordnungspaar stellt ein Schema dar, auf das durch ein SCH-Attribut in einer XDS verwiesen wird. Das zweite Schema in dem Zuordnungspaar gibt an, welches Schema für die Durchführung der Schemaprüfung verwendet werden soll.

Wenn im linken Teil eines Schemapaares in der Klausel **MAP** ein Schema angegeben ist, darf dieses nicht zusätzlich in der Klausel **IGNORE** angegeben werden.

Sobald eine Schemapaarzuordnung angewendet wurde, ist das Ergebnis endgültig. Die Zuordnungsoperation ist nicht transitiv, das gewählte Schema wird deshalb nicht anschließend auf eine andere Schemapaarzuordnung angewendet.

Ein Schema kann nicht mehrmals zugeordnet werden, d. h. ein Schema kann nur bei einem einzigen Paar als erstes Schema angegeben sein.

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.

Diese Option ist nicht zulässig, wenn sie zusammen mit dem Dateityp **CURSOR** angegeben wird und wenn eine Tabelle mit einer XML-Spalte geladen wird.

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.

WARNINGCOUNT *n*

Stoppt die Ladeoperation 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 Ladedatei oder die Zieltabelle falsch angegeben, generiert das Dienstprogramm LOAD eine Warnung für jede Zeile, die es zu laden versucht, sodass die Ladeoperation 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.

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:

- 136 Byte für jede Nachricht, die vom Dienstprogramm LOAD generiert wird
- 15 KB Systemaufwand, wenn die Datendatei Langfelddaten oder LOBs enthält. Dieser Anteil kann stark zunehmen, wenn die Option **INSERT** angegeben ist und die Tabelle bereits viele Spalten mit Langfeld- oder LOB-Daten enthält.

INSERT Einer von vier Modi, in denen das Dienstprogramm LOAD ausgeführt werden kann. Fügt die geladenen Daten zur Tabelle hinzu, ohne die vorhandenen Tabellendaten zu ändern.

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.

KEEPDICTIONARY

Ein vorhandenes Komprimierungswörterbuch wird über die Operation **LOAD REPLACE** hinaus beibehalten. Wenn das Tabellenattribut **COMPRESS** auf **YES** gesetzt ist, werden die neu ersetzten Daten mithilfe des Wörterbuchs komprimiert, das vor dem Aufrufen des Befehls **LOAD** vorhanden war. Wenn vorher kein Wörterbuch in der Tabelle vorhanden war, wird anhand der Daten, die in die Tabelle ersetzt werden, ein neues Wörterbuch erstellt, sofern das Tabellenattribut **COMPRESS** auf **YES** gesetzt ist. Welches Datenvolumen in diesem Fall zum Erstellen des Komprimierungswörterbuchs erforderlich ist, hängt von den Richtlinien für die automatische Erstellung von Komprimierungswörterbüchern (ADC) ab. Diese Daten werden in nicht komprimierter Form in die Tabelle eingetragen. Sobald das Wörterbuch in die Tabelle eingefügt ist, werden die übrigen zu ladenden Daten mithilfe dieses Wörterbuchs komprimiert. Dies ist der Standardparameter. Eine Zusammenfassung finden Sie in der nachfolgenden Tabelle.

Tabelle 27. **LOAD REPLACE KEEPDICTIONARY**

Komprimieren	Wörterverzeichnis der Tabellenzeilendaten ist vorhanden	Wörterverzeichnis der XML-Speicherobjekte ist vorhanden ¹	Komprimierungswörterverzeichnis	Datenkomprimierung
JA	JA	JA	Wörterverzeichnis der Tabellenzeilendaten und XML-Wörterverzeichnisse beibehalten.	Zu ladende Daten werden komprimiert.

Tabelle 27. LOAD REPLACE KEEPDICTIONARY (Forts.)

Komprimieren	Wörterverzeichnis der Tabellenzeilendaten ist vorhanden	Wörterverzeichnis der XML-Speicherobjekte ist vorhanden ¹	Komprimierungswörterverzeichnis	Datenkomprimierung
JA	JA	NEIN	Wörterverzeichnis für Tabellenzeilendaten beibehalten und neues XML-Wörterverzeichnis erstellen.	Zu ladende Tabellenzeilendaten werden komprimiert. Nach dem Erstellen des XML-Wörterverzeichnisses werden verbleibende XML-Daten, die geladen werden sollen, komprimiert.
JA	NEIN	JA	Wörterverzeichnis der Tabellenzeilendaten erstellen und XML-Wörterverzeichnis beibehalten.	Nach dem Erstellen des Wörterverzeichnisses der Tabellenzeilendaten werden verbleibende Tabellenzeilendaten, die geladen werden sollen, komprimiert. Zu ladende XML-Daten werden komprimiert.
JA	NEIN	NEIN	Wörterverzeichnis der Tabellenzeilendaten und XML-Wörterverzeichnis erstellen.	Nach dem Erstellen der Wörterverzeichnisse werden verbleibende Daten, die geladen werden sollen, komprimiert.
NEIN	JA	JA	Wörterverzeichnis der Tabellenzeilendaten und XML-Wörterverzeichnisse beibehalten.	Zu ladende Daten werden nicht komprimiert.
NEIN	JA	NEIN	Wörterverzeichnis der Tabellenzeilendaten beibehalten.	Zu ladende Daten werden nicht komprimiert.
NEIN	NEIN	JA	Keine Auswirkung auf das Wörterverzeichnis der Tabellenzeilendaten. XML-Wörterverzeichnis beibehalten.	Zu ladende Daten werden nicht komprimiert.
NEIN	NEIN	NEIN	Keine Auswirkung.	Zu ladende Daten werden nicht komprimiert.

Anmerkung:

1. Für das XML-Speicherobjekt einer Tabelle kann nur ein Komprimierungswörterverzeichnis erstellt werden, wenn die XML-Spalten der Tabelle in DB2 Version 9.7 oder einer höheren Version hinzugefügt werden bzw. wenn die Tabelle mithilfe einer Onlineoperation zum Versetzen von Tabellen migriert wird.
2. Wenn die Operation **LOAD REPLACE KEEPDICTIONARY** unterbrochen wird, kann das Dienstprogramm LOAD eine Recovery durchführen, nachdem entweder der Befehl **LOAD RESTART** oder der Befehl **LOAD TERMINATE** abgesetzt wird. Das vorhandene Wörterbuch für XML-Speicherobjekte kann möglicherweise nach einer Recovery nach einer unterbrochenen Operation **LOAD**

REPLACE KEEPDICTIONARY nicht beibehalten werden. Bei Verwendung des Befehls **LOAD RESTART** wird ein neues Wörterbuch für XML-Speicherobjekte erstellt.

RESETDICTIONARY

Diese Anweisung bewirkt, dass bei der Verarbeitung von **LOAD REPLACE** ein neues Wörterbuch für das Tabellendatenobjekt erstellt wird, sofern das Tabellenattribut COMPRESS auf YES gesetzt ist. Wenn das Attribut COMPRESS auf NO gesetzt ist und die Tabelle bereits ein Wörterbuch enthält, wird das vorhandene Wörterbuch entfernt und kein neues Wörterbuch in die Tabelle eingefügt. Ein Komprimierungswörterbuch kann mithilfe eines einzigen Benutzerdatensatzes erstellt werden. Wenn die Größe der geladenen Datei null ist und bereits ein Wörterbuch vorhanden ist, wird das Wörterbuch nicht beibehalten. Die erforderliche Datenmenge zum Erstellen eines Wörterbuchs mit dieser Anweisung hängt nicht von den Richtlinien für die automatische Erstellung von Komprimierungswörterbüchern (ADC) ab. Eine Zusammenfassung finden Sie in der nachfolgenden Tabelle.

Tabelle 28. LOAD REPLACE RESETDICTIONARY

Komprimieren	Wörterverzeichnis der Tabellenzeilendaten ist vorhanden	Wörterverzeichnis der XML-Speicherobjekte ist vorhanden ¹	Komprimierungswörterverzeichnis	Datenkomprimierung
JA	JA	JA	Neue Wörterverzeichnis erstellen. ² Wird die Option DATA CAPTURE CHANGES in den Anweisungen CREATE TABLE oder ALTER TABLE aktiviert, wird das aktuelle Wörterverzeichnis für Tabellenzeilendaten beibehalten (und als <i>Protokollwörterverzeichnis der Komprimierung</i> bezeichnet).	Nach dem Erstellen der Wörterverzeichnisse werden verbleibende Daten, die geladen werden sollen, komprimiert.
JA	JA	NEIN	Neue Wörterverzeichnis erstellen. ² Wird die Option DATA CAPTURE CHANGES in den Anweisungen CREATE TABLE oder ALTER TABLE aktiviert, wird das aktuelle Wörterverzeichnis für Tabellenzeilendaten beibehalten (und als <i>Protokollwörterverzeichnis der Komprimierung</i> bezeichnet).	Nach dem Erstellen der Wörterverzeichnisse werden verbleibende Daten, die geladen werden sollen, komprimiert.
JA	NEIN	JA	Neue Wörterverzeichnis erstellen.	Nach dem Erstellen der Wörterverzeichnisse werden verbleibende Daten, die geladen werden sollen, komprimiert.

Tabelle 28. LOAD REPLACE RESETDICTIONARY (Forts.)

Komprimieren	Wörterverzeichnis der Tabellenzeilendaten ist vorhanden	Wörterverzeichnis der XML-Speicherobjekte ist vorhanden ¹	Komprimierungswörterverzeichnis	Datenkomprimierung
JA	NEIN	NEIN	Neue Wörterverzeichnisse erstellen.	Nach dem Erstellen der Wörterverzeichnisse werden verbleibende Daten, die geladen werden sollen, komprimiert.
NEIN	JA	JA	Wörterverzeichnisse löschen.	Zu ladende Daten werden nicht komprimiert.
NEIN	JA	NEIN	Wörterverzeichnis der Tabellenzeilendaten entfernen.	Zu ladende Daten werden nicht komprimiert.
NEIN	NEIN	JA	Wörterverzeichnis der XML-Speicherobjekte entfernen.	Zu ladende Daten werden nicht komprimiert.
NEIN	NEIN	NEIN	Keine Auswirkung.	Alle Tabellendaten werden nicht komprimiert.

Hinweise:

1. Für das XML-Speicherobjekt einer Tabelle kann nur ein Komprimierungswörterverzeichnis erstellt werden, wenn die XML-Spalten der Tabelle in DB2 Version 9.7 oder einer höheren Version hinzugefügt werden bzw. wenn die Tabelle mithilfe einer Onlineoperation zum Versetzen von Tabellen migriert wird.
2. Ist ein Wörterbuch vorhanden und das Komprimierungsattribut ist aktiviert, aber es sind keine Datensätze zum Laden in die Tabellenpartition vorhanden, so kann kein neues Wörterbuch erstellt werden, und die Operation **RESETDICTIONARY** behält das vorhandene Wörterbuch nicht bei.

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, sodass beim nächsten Zugriff automatisch eine Indexwiederherstellung erfolgt). Wenn es sich bei der zu beendenden Ladeoperation um **LOAD REPLACE** handelt, wird die Tabelle nach der Operation **LOAD TERMINATE** auf eine leere Tabelle reduziert. Handelt es sich bei der zu beendenden Ladeoperation um **LOAD INSERT**, bleiben nach der Operation **LOAD TERMINATE** alle ursprünglichen Datensätze der Tabelle erhalten. Eine Zusammenfassung der Wörterbuchverwaltung finden Sie in der nachfolgenden Tabelle.

Die Option **LOAD TERMINATE** hebt keinen Status für anstehendes Backup in Tabellenbereichen auf.

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. Eine Zusammenfassung der Wörterbuchverwaltung finden Sie in der nachfolgenden Tabelle.

INTO *tabellenname*

Gibt die Datenbanktabelle an, in die die Daten geladen werden sollen. Diese Tabelle darf keine Systemtabelle, keine deklarierte temporäre Tabelle und keine erstellte 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.

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:

Dieser Befehl schlägt aufgrund der Spalte Int 4 fehl. Die Lösung besteht darin, solche Spaltennamen in doppelte Anführungszeichen einzuschließen.

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. Wenn ein Tabellename 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 partitionierter Datenbank muss eine Ausnahmetabelle für die Datenbankpartitionen definiert werden, auf denen die Ladetabelle definiert ist. Andernfalls enthält die Speicherauszugsdatei Zeilen, die nicht geladen werden können, weil sie ungültig sind oder Syntaxfehler enthalten.

Wenn Sie XML-Daten laden, wird die Angabe der Ladeausnahmetabelle mit der Klausel **FOR EXCEPTION** in den folgenden Fällen nicht unterstützt:

- Bei Verwendung von kennsatzbasierter Zugriffssteuerung (LBAC)
- Beim Laden von Daten in eine partitionierte Tabelle

NORANGEEXC

Gibt an, dass eine wegen einer Bereichsverletzung zurückgewiesene Zeile nicht in die Ausnahmetabelle eingefügt wird.

NOUNIQUEEXC

Gibt an, dass eine Zeile, die zurückgewiesen wird, weil sie eine eindeutige Integritätsbedingung verletzt, nicht in die Ausnahmetabelle eingefügt wird.

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. Das Profil wird durch den Befehl **RUNSTATS** erstellt. 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.

Während des Ladens werden für Spalten des Typs XML keine Verteilungstatistiken 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.

COPY NO

Gibt an, dass der Tabellenbereich, in dem sich die Tabelle befindet, in den Status 'Backup anstehend' versetzt wird, wenn die aktualisierende Recovery aktiviert ist (dass heißt, **logretain** oder **userexit** ist aktiv). Die Option **COPY NO** versetzt den Tabellenbereich außerdem in den Status 'Laden läuft'. Dies ist ein vorübergehender Status, der aufgehoben wird, sobald das Laden abgebrochen wird oder abgeschlossen ist. Die Daten in allen Tabellen des Tabellenbereichs können erst dann aktualisiert oder gelöscht werden, nachdem ein Backup des Tabellenbereichs oder ein Datenbankgesamtbackup durchgeführt wurde. Der Zugriff auf die Daten in allen Tabellen ist jedoch durch die SELECT-Anweisung möglich.

Der Befehl **LOAD** mit der Option **COPY NO** für eine wiederherstellbare Datenbank versetzt die Tabellenbereiche in den Status 'Backup anstehend'. Beispielsweise ist nach Ausführung des Befehls **LOAD** mit den Optionen **COPY NO** und **INDEXING MODE DEFERRED** eine Aktualisierung der Indizes erforderlich. 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 'Backup anstehend' hat. In diesem Fall ist der Zugriff auf die Tabelle nicht zulässig, bis ein Backup ausgeführt wurde. Die Indexaktualisierung wird automatisch durch die Datenbank ausgeführt, wenn eine Abfrage auf den Index zugreift. Wenn keine der Optionen **COPY NO**, **COPY YES** oder **NONRECOVERABLE** angegeben wird und die Datenbank wiederherstellbar ist (**logretain** oder **logarchmeth1** ist aktiviert), wird **COPY NO** als Standardwert verwendet.

COPY YES

Gibt an, dass eine Kopie der geladenen Daten gesichert wird. Diese Option ist ungültig, wenn die aktualisierende Recovery inaktiviert ist.

USE TSM

Gibt an, dass die Kopie mithilfe 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 Kopienimage erstellt wird.

LOAD *bib-name*

Der Name der gemeinsam genutzten Bibliothek (DLL unter Windows-Betriebssystemen), in der die zu verwendenden E/A-Funktionen für Backup und Restore des Herstellers enthalten sind. Der Name kann den vollständigen Pfad enthalten. Wenn der vollständige Pfad nicht angegeben wird, wird standardmäßig der Pfad verwendet, in dem sich die Benutzerexitprogramme befinden.

NONRECOVERABLE

Gibt an, dass eine Ladetransaktion als nicht wiederherstellbar gekennzeichnet werden soll und dass sie nicht durch eine nachfolgende aktualisieren-

de Recovery wiedergestellt werden kann. Das Dienstprogramm für aktualisierende Recovery 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 aktualisierenden Recovery kann eine solche Tabelle nur nach einem Commitpunkt, der auf den Abschluss der nicht wiederherstellbaren Ladeoperation folgt, gelöscht oder aus einem Backup (Gesamtbackup oder Backup des Tabellenbereichs) wiederhergestellt werden.

Mit dieser Option werden Tabellenbereiche nach der Ladeoperation nicht in den Status 'Backup anstehend' versetzt, und während der Ladeoperation muss keine Kopie der geladenen Daten erstellt werden. Wenn keine der Optionen **COPY NO**, **COPY YES** oder **NONRECOVERABLE** angegeben wird und die Datenbank nicht wiederherstellbar ist (weder **logretain** noch **logarchmeth1** ist aktiviert), wird **NONRECOVERABLE** als Standardwert verwendet.

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'.

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 für Dienstprogramme zugeordnet, dessen Größe durch den Datenbankkonfigurationsparameter **util_heap_sz** geändert werden kann. Ab Version 9.5 kann der Wert für die Option **DATA BUFFER** des Befehls **LOAD** den Wert von **util_heap_sz** temporär überschreiten, falls im System mehr Speicher zur Verfügung steht. In dieser Situation wird der Zwischenspeicher für Dienstprogramme entsprechend dem Bedarf dynamisch erhöht, bis der Grenzwert für **datenbankspeicher** erreicht ist. Dieser Speicher wird freigegeben, sobald die Ladeoperation abgeschlossen ist.

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 Instanzerstellung des Ladeprogramms im Zwischenspeicher der Dienstprogramme verfügbar ist, sowie auf einigen Merkmalen der Tabelle.

SORT BUFFER *puffergröße*

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 **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.

CPU_PARALLELISM *n*

Gibt die Anzahl von Prozessen bzw. Threads an, die vom Dienstprogramm LOAD zur Syntaxanalyse, Konvertierung und Formatierung von Datensätzen verwendet werden. Dieser Parameter soll die Nutzung der pro Datenbankpartition aktiven Prozesse 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.

Anmerkung:

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 der Wert für **CPU_PARALLELISM** größer als eins ist, erfolgen die Flushoperationen asynchron und das Ladeprogramm kann die CPU nutzen. Wenn **CPU_PARALLELISM** auf den Wert eins gesetzt ist, wartet das Ladeprogramm an Konsistenzpunkten auf E/A-Daten. Eine Ladeoperation mit dem Wert 2 für **CPU_PARALLELISM** und dem Wert 10 000 für **SAVECOUNT** wird schneller abgeschlossen als die gleiche Operation mit dem Wert 1 für **CPU_PARALLELISM**, obwohl nur eine CPU vorhanden ist.

DISK_PARALLELISM *n*

Gibt die Anzahl von Prozessen bzw. Threads an, die vom Dienstprogramm LOAD erstellt werden, um Daten in die Tabellenbereichscontainer zu schreiben. Wenn kein Wert angegeben wird, wählt das Dienstprogramm einen sinnvollen Standardwert auf Basis der Anzahl der Tabellenbereichscontainer und der Merkmale der Tabelle aus.

INDEXING MODE

Gibt an, ob das Dienstprogramm LOAD Indizes erneut erstellen oder sie inkrementell erweitern soll. Gültige Werte sind:

AUTOSELECT

Das Dienstprogramm LOAD wählt automatisch den Modus REBUILD oder INCREMENTAL aus. Die Entscheidung richtet sich nach dem Umfang der zu ladenden Daten und der Tiefe der Indexstruktur. Informationen zur Tiefe der Indexstruktur sind im Indexobjekt gespeichert. **RUNSTATS** ist nicht erforderlich, um diese Informationen zu füllen. **AUTOSELECT** ist der Standardmodus für die Indexierung.

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 unterstützt, wenn das Indexobjekt am Anfang einer Ladeoperation gültig und erreichbar ist

(beispielsweise ist es unmittelbar nach einer Ladeoperation, in der der Modus DEFERRED angegeben war, nicht gültig). Wenn dieser Modus angegeben ist, aber aufgrund des Indexstatus nicht unterstützt wird, wird eine Warnung zurückgegeben und die Ladeoperation wird im Modus REBUILD fortgesetzt. Außerdem wird der Modus INCREMENTAL nicht unterstützt, wenn eine LOAD RESTART-Operation in der BUILD-Phase der Ladeoperation eingeleitet wird.

DEFERRED

Das Dienstprogramm LOAD erstellt keine Indizes, wenn dieser Modus angegeben wird. Die Indizes werden als aktualisierungsbedürftig gekennzeichnet. Bei dem ersten Zugriff auf solche Indizes, der nicht im Rahmen einer Ladeoperation erfolgt, wird möglicherweise eine erneute Erstellung erzwungen, oder die Indizes werden erneut erstellt, wenn die Datenbank erneut gestartet wird. Dieser Ansatz erfordert genug Sortierspeicherplatz für alle Schlüsselteile für den größten Index. Die im Anschluss daran für die Indexerstellung benötigte Gesamtzeit ist länger als die im Modus REBUILD benötigte Zeit. 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.

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 zulässige Option für **LOAD REPLACE**.

Wenn für die Tabelle Integritätsbedingungen gelten, wird der Tabellenstatus auf 'Festlegen der Integrität anstehend' sowie 'Laden läuft' gesetzt. Die Anweisung SET INTEGRITY muss verwendet werden, um den Status 'Festlegen der Integrität anstehend' 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 Deltateil 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 der Ladeoperation verfügbar. **LOAD TERMINATE** oder **LOAD RESTART** in einer Ladeoperation mit **ALLOW READ ACCESS** kann diese Option verwenden. **LOAD TERMINATE** oder **LOAD RESTART** in einer Ladeoperation mit **ALLOW NO ACCESS** kann diese Option nicht verwenden. 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 'Festlegen der Integrität anstehend' sowie 'Laden läuft' und 'Lesezu-

griff' gesetzt. Am Ende der Ladeoperation wird der Tabellenstatus 'Laden läuft' aufgehoben. Die Statuswerte 'Festlegen der Integrität anstehend' und 'Lesezugriff' für die Tabelle bleiben jedoch gültig. Die Anweisung SET INTEGRITY muss verwendet werden, um den Status 'Festlegen der Integrität anstehend' für die Tabelle aufzuheben. Während sich die Tabelle im Status 'Festlegen der Integrität anstehend' und 'Lesezugriff' 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 Ladeoperationen 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*

Wenn die Indizes erneut erstellt werden, wird eine Spiegelkopie des Indexes im Tabellenbereich *tabellenbereichsname* erstellt und am Ende der Ladeoperation während einer 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 während 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 vorkommen, 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. Sie wird auch ignoriert, wenn **INDEXING MODE AUTOSELECT** ausgewählt ist und das Dienstprogramm LOAD eine inkrementelle Aktualisierung des Indexes einleitet.

FETCH_PARALLELISM YES | NO

Wird eine Ladeoperation über einen Cursor ausgeführt, der mit dem Schlüsselwort **DATABASE** deklariert wurde, oder über den API-Datenträger eintrag `sqlu_remotefetch_entry`, der auf YES gesetzt ist, versucht das Dienstprogramm LOAD (falls möglich), Daten parallel aus der fernen Datenquelle abzurufen. Ist die Option auf NO gesetzt, erfolgt kein paralleles Abrufen. Der Standardwert ist YES. Weitere Informationen hierzu finden Sie unter „Versetzen von Daten mithilfe des Dateityps CURSOR“.

SET INTEGRITY PENDING CASCADE

Wenn **LOAD** die Tabelle in den Status 'Festlegen der Integrität anstehend' versetzt, kann der Benutzer über die Option **SET INTEGRITY PENDING**

CASCADE angeben, ob dieser Status der geladenen Tabelle direkt an alle untergeordneten Elemente (einschließlich der untergeordneten Fremdschlüsseltabellen, IMQs und sofort gespeicherten Zwischenspeichertabellen) weitergegeben werden soll.

IMMEDIATE

Gibt an, dass der Status 'Festlegen der Integrität anstehend' unmittelbar auf alle untergeordneten Fremdschlüsseltabellen, untergeordneten IMQs und untergeordneten sofort gespeicherten Zwischenspeichertabellen ausgeweitet wird. Bei einer **LOAD INSERT**-Operation wird der Status 'Festlegen der Integrität anstehend' nicht an untergeordnete Fremdschlüsseltabellen weitergegeben, selbst wenn die Option **IMMEDIATE** angegeben wird.

Bei der späteren Prüfung der geladenen Tabelle auf ungültige Integritätsbedingungen (mit der Option **IMMEDIATE CHECKED** der Anweisung **SET INTEGRITY**), werden untergeordnete Fremdschlüsseltabellen, die den Status 'Festlegen der Integrität anstehend, Lesezugriff' aufweisen, in den Status 'Festlegen der Integrität anstehend, Kein Zugriff' versetzt.

DEFERRED

Gibt an, dass nur die geladene Tabelle in den Status 'Festlegen der Integrität anstehend' versetzt wird. Die Statuswerte von untergeordneten Fremdschlüsseltabellen, untergeordneten IMQs und untergeordneten sofort gespeicherten Zwischenspeichertabellen werden nicht geändert.

Untergeordnete Fremdschlüsseltabellen werden später möglicherweise implizit in den Status 'Festlegen der Integrität anstehend' versetzt, wenn die dazugehörigen übergeordneten Tabellen auf ungültige Integritätsbedingungen geprüft werden (mit der Option **IMMEDIATE CHECKED** der Anweisung **SET INTEGRITY**). Untergeordnete IMQs und untergeordnete sofort gespeicherte Zwischenspeichertabellen werden implizit in den Status 'Festlegen der Integrität anstehend' versetzt, sobald eine der ihnen zu Grunde liegenden Tabellen auf ungültige Integritätsbedingungen überprüft wird. Eine Abfrage einer Tabelle im Status 'Festlegen der Integrität anstehend' kann erfolgreich sein, wenn die Abfrage nicht auf die angegebene Tabelle zugreift, sondern auf eine infrage kommende MQI, die sich nicht im Status 'Festlegen der Integrität anstehend' befindet. Eine Warnung (SQLSTATE 01586) wird ausgegeben, um anzugeben, dass untergeordnete Tabellen in den Status 'Festlegen der Integrität anstehend' versetzt wurden. Angaben dazu, wann diese untergeordneten Tabellen in den Status 'Festlegen der Integrität anstehend' versetzt werden, finden Sie im Abschnitt mit den Hinweisen zur Anweisung **SET INTEGRITY** im Handbuch 'SQL Reference'.

Wenn die Option **SET INTEGRITY PENDING CASCADE** nicht angegeben ist, gilt Folgendes:

- Nur die geladene Tabelle wird in den Status 'Festlegen der Integrität anstehend' versetzt. Die Statuswerte von untergeordneten Fremdschlüsseltabellen, untergeordneten IMQs und untergeordneten sofort gespeicherten Zwischenspeichertabellen bleiben unverändert. Diese Tabellen können später implizit in den Status 'Festlegen der Integrität anstehend' versetzt werden, sobald die geladene Tabelle auf ungültige Integritätsbedingungen hin überprüft wird.

Wenn **LOAD** die Zieltabelle nicht in den Status 'Festlegen der Integrität anstehend' versetzt, wird die Option **SET INTEGRITY PENDING CASCADE** ignoriert.

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 benötigt wie für den Befehl **FORCE APPLICATIONS** (SYSADM oder SYSCTRL).

Ladeoperationen mit ALLOW NO ACCESS können die Beendigung von Anwendungen mit Konflikte verursachenden Sperren am Anfang der Ladeoperation erzwingen. Am Anfang der Ladeoperation kann das Dienstprogramm die Beendigung von Anwendungen erzwingen, die versuchen, die Tabelle entweder abzufragen oder zu ändern.

Ladeoperationen mit **ALLOW READ ACCESS** können die Beendigung von Anwendungen mit Konflikte verursachenden Sperren am Anfang oder Ende der Ladeoperation erzwingen. Am Anfang der Ladeoperation kann das Dienstprogramm **LOAD** die Beendigung von Anwendungen erzwingen, die versuchen, die Tabelle zu ändern. Am Ende der Ladeoperation kann das Dienstprogramm **LOAD** die Beendigung von Anwendungen erzwingen, die versuchen, die Tabelle abzufragen oder zu ändern.

SOURCEUSEREXIT *ausführbare-datei*

Gibt den Namen einer ausführbaren Datei an, die aufgerufen werden soll, um dem Dienstprogramm Daten zuzuführen.

REDIRECT

INPUT FROM

BUFFER *eingabepuffer*

Der Bytestrom, der in *eingabepuffer* angegeben ist, wird in den Dateideskriptor STDIN des Prozesses übergeben, in dem die gegebene ausführbare Datei ausgeführt wird.

FILE *ingabedatei*

Der Inhalt dieser clientseitigen Datei wird in den Dateideskriptor STDIN des Prozesses übergeben, in dem die gegebene ausführbare Datei ausgeführt wird.

OUTPUT TO

FILE *ausgabedatei*

Die Dateideskriptoren STDOUT und STDERR werden in der angegebenen vollständig qualifizierten serverseitigen Datei erfasst.

PARALLELIZE

Erhöht den Durchsatz der im Dienstprogramm **LOAD** eingehenden

Daten, indem mehrere Benutzerexitprozesse gleichzeitig aufgerufen werden. Diese Option gilt nur für Umgebungen mit partitionierter Datenbank und wird in Umgebungen mit Einzelpartitionsdatenbank ignoriert.

Weitere Informationen hierzu finden Sie unter „Versetzen von Daten mithilfe einer angepassten Anwendung (Benutzerexit)“.

PARTITIONED DB CONFIG *option-für-partitionierte-db*

Dieser Parameter ermöglicht das Laden von Daten in eine Tabelle, die auf mehrere Datenbankpartitionen verteilt ist. Der Parameter **PARTITIONED DB CONFIG** ermöglicht das Angeben von spezifischen Konfigurationsoptionen für partitionierte Datenbanken. Die *option-für-partitionierte-db* kann folgende Werte annehmen:

```
PART_FILE_LOCATION x
OUTPUT_DBPARTNUMS x
PARTITIONING_DBPARTNUMS x
MODE x
MAX_NUM_PART_AGENTS x
ISOLATE_PART_ERRS x
STATUS_INTERVAL x
PORT_RANGE x
CHECK_TRUNCATION
MAP_FILE_INPUT x
MAP_FILE_OUTPUT x
TRACE x
NEWLINE
DISTFILE x
OMIT_HEADER
RUN_STAT_DBPARTNUM x
```

Ausführliche Beschreibungen dieser Optionen finden Sie unter „LOAD-Konfigurationsoptionen für Umgebungen mit partitionierten Datenbanken“.

RESTARTCOUNT

Veraltet.

USING *verzeichnis*

Veraltet.

Beispiele für das Laden von Daten aus XML-Dokumenten

XML-Daten laden

Beispiel 1

Vom Benutzer wurde eine Datendatei mit XDS-Feldern zum Beschreiben der Dokumente erstellt, die in die Tabelle eingefügt werden sollen. Die Datei sieht wie folgt aus:

```
1, "<XDS FIL=""datei1.xml"" />"
2, "<XDS FIL='datei2.xml' OFF='23' LEN='45' />"
```

In der ersten Zeile wird das XML-Dokument durch die Datei namens `datei1.xml` identifiziert. Da das Anführungszeichen als Zeichenbegrenzer verwendet wird und die XDS Anführungszeichen enthält, werden die Anführungszeichen in der XDS gedoppelt. In der zweiten Zeile wird das XML-Dokument durch die Datei namens `datei2.xml` identifiziert. Es beginnt an der relativen Byteadresse 23 und ist 45 Byte lang.

Beispiel 2

Der Benutzer setzt einen Befehl **LOAD** für die XML-Spalte ohne Optionen für Syntaxanalyse oder Prüfung ab, und die Daten werden erfolgreich geladen:

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

Laden von XML-Daten vom CURSOR

Das Laden von Daten vom Cursor entspricht dem Laden von einem regulären relationalen Spaltentyp. Der Benutzer verfügt über zwei Tabellen (T1 und T2) die jeweils aus einer einzigen XML-Spalte namens C1 bestehen. Zum Laden von T1 in T2 muss der Benutzer zunächst einen Cursor deklarieren:

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

Im nächsten Schritt kann der Benutzer einen Befehl **LOAD** absetzen, der den Cursortyp verwendet:

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

Das Anwenden der XML-spezifischen **LOAD**-Optionen auf den Cursortyp entspricht dem Laden aus einer Datei.

Hinweise 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. Wenn die Reihenfolge der Quelldaten nicht beibehalten werden muss, können Sie auch die Verwendung des Dateitypänderungswerts **ANYORDER** in Betracht ziehen, der im nachfolgenden Abschnitt „Änderungswerte für Dateityp für das Dienstprogramm **LOAD**“ beschrieben wird.
- Das Dienstprogramm **LOAD** erstellt Indizes gemäß den vorhandenen Definitionen. Mithilfe von Ausnahmetabellen werden doppelt vorhandene Werte für eindeutige Schlüssel verarbeitet. Das Dienstprogramm nimmt für MQT (Materialized Query Tables), 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 'Festlegen der Integrität anstehend' versetzt. Übersichtstabellen, für die das Attribut **REFRESH IMMEDIATE** definiert ist und die von geladenen Tabellen abhängig sind, werden ebenfalls in den Status 'Festlegen der Integrität anstehend' versetzt. Durch Absetzen der Anweisung **SET INTEGRITY** können Sie den Status 'Festlegen der Integrität anstehend' für die Tabellen aufheben. Die Ausführung von Ladeoperationen für replizierte MQT 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.
- Wenn Sie beim Laden von Daten in eine geschützte Tabelle eine Ausnahmetabelle angeben, werden alle Zeilen, die durch ungültige Sicherheitskennsätze geschützt sind, in diese Tabelle übertragen. Dies ermöglicht es Benutzern, die Zugriff auf die Ausnahmetabelle haben, auf Daten zuzugreifen, für die ihnen eigentlich die Zugriffsberechtigung fehlt. Aus Sicherheitsgründen empfiehlt es sich, die Benutzer, die mit Grant Zugriff auf die Ausnahmetabelle erhalten, sorgfältig auszuwählen, die einzelnen Zeilen, sobald sie repariert und in die zu ladende Tabelle kopiert sind, sofort zu löschen und die Ausnahmetabelle selbst sofort zu löschen, sobald sie nicht mehr benötigt wird.

- Sicherheitskennsätze können in ihrem internen Format Zeilenvorschubzeichen enthalten. Wenn Sie die Datei mit dem Dateiformat DEL laden, können diese Zeilenvorschubzeichen als Begrenzer interpretiert werden. Wenn dieses Problem vorliegt, geben Sie den Änderungswert **delprioritychar** für den Dateityp im Befehl **LOAD** an, um die vorherige Standardpriorität für Begrenzer zu verwenden.
- Zum Ausführen einer Ladeoperation für den Dateityp CURSOR, bei der das Schlüsselwort DATABASE in der Anweisung DECLARE CURSOR angegeben wurde, wird die Benutzer-ID mit Kennwort zur Authentifizierung in der Datenbank, zu der momentan eine Verbindung (für die Ladeoperation) besteht, für die Authentifizierung in der Quelldatenbank verwendet, die in der Option DATABASE der Anweisung DECLARE CURSOR angegeben ist. Wurde keine Benutzer-ID und/oder kein Kennwort für die Verbindung zur ladenden Datenbank angegeben, müssen bei der Anweisung DECLARE CURSOR eine Benutzer-ID und ein Kennwort für die Quelldatenbank angegeben werden.
- Das Laden einer mehrteiligen PC/IXF-Datei, deren Bestandteile von einem Windows-System auf ein AIX-System kopiert werden, wird unterstützt. Die Namen aller Dateien müssen im Befehl **LOAD** angegeben werden. Beispiel: **LOAD FROM DATA.IXF, DATA.002 OF IXF INSERT INTO TABLE1**. Das Laden von logisch geteilten PC/IXF-Dateien in das Windows-Betriebssystem wird nicht unterstützt.
- Beim erneuten Starten einer fehlgeschlagenen **LOAD**-Operation wird das bestehende Verhalten insofern beibehalten, als in der BUILD-Phase die Verwendung des REBUILD-Modus für Indizes erzwungen wird.
- Das Laden von XML-Dokumenten von einer Datenbank auf andere Datenbanken wird nicht unterstützt und führt zur Ausgabe der Fehlermeldungen SQL1407N.
- Das Dienstprogramm **LOAD** bietet keine Unterstützung für das Laden in Tabellen, in denen Spalten enthalten sind, die auf abgeschirmte Prozeduren verweisen. Wenn Sie für eine solche Tabelle den Befehl **LOAD** absetzen, erhalten Sie die Fehlermeldung SQL1376N. Sie können diese Einschränkung umgehen, indem Sie die Routine neu definieren, sodass sie nicht abgeschirmt ist, oder das Dienstprogramm IMPORT verwenden.
- Der Befehl STATISTICS YES ist nur eingeschränkt anwendbar und wird möglicherweise in einem zukünftigen Release entfernt.
- Die Optionen STATISTICS lassen sich nur mit der Option **LOAD REPLACE** und keinen anderen Befehlsoptionen des Typs **LOAD** anwenden.

Zusammenfassung der **LOAD TERMINATE**- und **LOAD RESTART**-Wörterbuchverwaltung

Das folgende Diagramm enthält eine Zusammenfassung der Verwaltung von Komprimierungswörterbüchern für die **LOAD**-Verarbeitung im Rahmen der Anweisung **TERMINATE**.

Tabelle 29. **LOAD TERMINATE**-Wörterbuchverwaltung

Tabellenattribut COMPRESS	Wörterverzeichnis der Tabellenzeilendaten ist vor der LOAD -Operation vor- .r handen	Wörterverzeichnis der XML-Speicherobjekte ist vor der LOAD -Operation vorhanden ¹	TERMINATE: LOAD REPLACE KEEPDICTIONARY oder LOAD INSERT	TERMINATE: LOAD REPLACE RESETDICTIONARY
JA	JA	JA	Vorhandene Wörterverzeichnisse beibehalten.	Kein Wörterverzeichnis wird beibehalten. ²
JA	JA	NEIN	Vorhandenes Wörterbuch beibehalten.	Es wird nichts beibehalten. ²

Tabelle 29. LOAD TERMINATE-Wörterbuchverwaltung (Forts.)

Tabellenattribut COMPRESS	Wörterverzeichnis der Tabellen- zeilendaten ist vor der LOAD- Operation vor- .r handen	Wörterverzeichnis der XML- Speicherobjekte ist vor der LOAD-Opera- tion vorhanden ¹	TERMINATE: LOAD REPLACE KEEPDICTIONARY oder LOAD INSERT	TERMINATE: LOAD REPLACE RESETDICTIONARY
JA	NEIN	JA	Vorhandenes Wörterbuch beibehalten.	Es wird nichts beibehal- ten.
JA	NEIN	NEIN	Es wird nichts beibehalten.	Es wird nichts beibehal- ten.
NEIN	JA	JA	Vorhandene Wörterver- zeichnisse beibehalten.	Es wird nichts beibehal- ten.
NEIN	JA	NEIN	Vorhandenes Wörterbuch beibehalten.	Es wird nichts beibehal- ten.
NEIN	NEIN	JA	Vorhandenes Wörterbuch beibehalten.	Es wird nichts beibehal- ten.
NEIN	NEIN	NEIN	Nichts ausführen.	Nichts ausführen.

Anmerkung:

1. Für das XML-Speicherobjekt einer Tabelle kann nur ein Komprimierungswörterverzeichnis erstellt werden, wenn die XML-Spalten der Tabelle in DB2 Version 9.7 oder einer höheren Version hinzugefügt werden bzw. wenn die Tabelle mithilfe einer Onlineoperation zum Versetzen von Tabellen migriert wird.
2. In den Sonderfällen, in denen die Datenerfassung für die Tabelle aktiviert ist, wird das Wörterverzeichnis für die Tabellenzeilendaten beibehalten.

LOAD RESTART verkürzt eine Tabelle bis zum letzten erreichten Konsistenzpunkt. Im Rahmen der **LOAD RESTART**-Verarbeitung existiert ein Komprimierungswörterbuch in der Tabelle, wenn es bereits beim Erreichen des letzten **LOAD**-Konsistenzpunkts in der Tabelle vorhanden war. In diesem Fall erstellt **LOAD RESTART** kein neues Wörterbuch. Eine Zusammenfassung der möglichen Bedingungen finden Sie in der nachfolgenden Tabelle.

Tabelle 30. LOAD RESTART-Wörterbuchverwaltung

Tabellenattribut COMPRESS	Wörterverzeichnis der Tabellen- zeilendaten ist vor dem LOAD- Synchronisations- punkt vorhanden ¹	Wörterverzeichnis der XML- Speicherobjekte ist vor der letzten LOAD- Operation vorhan- den ²	RESTART: LOAD REPLACE KEEPDICTIONARY oder LOAD INSERT	RESTART: LOAD REPLACE RESETDICTIONARY
JA	JA	JA	Vorhandene Wörterver- zeichnisse beibehalten.	Vorhandene Wörterver- zeichnisse beibehalten.
JA	JA	NEIN	Wörterverzeichnis der Tabellenzeilendaten beibe- halten und XML-Wörterver- zeichnis auf der Basis von ADC erstellen.	Wörterverzeichnis der Tabellenzeilendaten beibe- halten und XML-Wörter- verzeichnis erstellen.
JA	NEIN	JA	Wörterverzeichnis der Tabellenzeilendaten auf der Basis von ADC erstellen. Vorhandenes XML-Wörter- verzeichnis beibehalten.	Wörterverzeichnis der Tabellenzeilendaten er- stellen. Vorhandenes XML-Wörterverzeichnis beibehalten.

Tabelle 30. LOAD RESTART-Wörterbuchverwaltung (Forts.)

Tabellenattribut COMPRESS	Wörterverzeichnis der Tabellen- zeilendaten ist vor dem LOAD- Synchronisations- punkt vorhanden ¹	Wörterverzeichnis der XML- Speicherobjekte ist vor der letzten LOAD- Operation vorhan- den ²	RESTART: LOAD REPLACE KEEPDICTIONARY oder LOAD INSERT	RESTART: LOAD REPLACE RESETDICTIONARY
JA	NEIN	NEIN	Wörterverzeichnis der Tabellenzeilendaten und XML-Wörterverzeichnis auf der Basis von ADC erstel- len.	Wörterverzeichnis der Tabellenzeilendaten und XML-Wörterverzeichnis erstellen.
NEIN	JA	JA	Vorhandene Wörterver- zeichnisse beibehalten.	Vorhandene Wörterverzeichnisse löschen.
NEIN	JA	NEIN	Vorhandenes Wörterver- zeichnis der Tabellenzeilendaten beibe- halten.	Vorhandenes Wörterver- zeichnis der Tabellenzeilendaten ent- fernen.
NEIN	NEIN	JA	Vorhandenes XML-Wörter- verzeichnis beibehalten.	Vorhandenes XML-Wör- terverzeichnis entfernen.
NEIN	NEIN	NEIN	Nichts ausführen.	Nichts ausführen.

Hinweise:

1. Die Option **SAVECOUNT** ist für das Laden von XML-Daten nicht zulässig. Ladeoperationen, bei denen in der LOAD-Phase ein Fehler auftritt, werden vom Beginn der Operation an erneut gestartet.
2. Für das XML-Speicherobjekt einer Tabelle kann nur ein Komprimierungswörterverzeichnis erstellt werden, wenn die XML-Spalten der Tabelle in DB2 Version 9.7 oder einer höheren Version hinzugefügt werden bzw. wenn die Tabelle mithilfe einer Onlineoperation zum Versetzen von Tabellen migriert wird.

Änderungswerte für Dateityp für das Dienstprogramm LOAD

Tabelle 31. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: Alle Dateiformate

Änderungswert	Beschreibung
anyorder	Dieser Änderungswert wird zusammen mit dem Parameter cpu_parallelism verwendet. Er gibt an, dass die Beibehaltung der Quelldatenreihenfolge nicht erforderlich ist, was zu erheblichen Leistungssteigerungen auf SMP-Systemen führt. Wenn der Wert von cpu_parallelism 1 ist, wird diese Option ignoriert. Diese Option wird nicht unterstützt, wenn SAVECOUNT > 0, da die Recovery 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 generatedmissing oder dem Änderungswert generatedoverride verwendet werden.

Tabelle 31. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
generatedmissing	<p>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 generatedignore oder dem Änderungswert generatedoverride verwendet werden.</p>
generatedoverride	<p>Dieser Änderungswert weist das Dienstprogramm LOAD an (entgegen der normalen Regeln für diese Spalentyten), 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 RECOVER DROPPED TABLE des Befehls ROLLFORWARD DATABASE 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). Bei Verwendung dieses Änderungswerts wird die Tabelle in den Status 'Festlegen der Integrität anstehend' versetzt. Um den Status 'Festlegen der Integrität 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 <i>tabellenname</i> GENERATED COLUMN IMMEDIATE UNCHECKED</pre> <p>Um den Status 'Festlegen der Integrität 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 <i>tabellenname</i> IMMEDIATE CHECKED.</pre> <p>Wird dieser Änderungswert angegeben und gibt es in einem der partitionierten Schlüssel, Dimensionsschlüssel oder Verteilungsschlüssel eine generierte Spalte, konvertiert der Befehl LOAD den Änderungswert automatisch in generatedignore und fährt mit dem Laden fort. Dies bewirkt, dass alle generierten Spaltenwerte erneut generiert werden.</p> <p>Dieser Änderungswert kann nicht zusammen mit dem Änderungswert generatedmissing oder dem Änderungswert generatedignore verwendet werden.</p>
identityignore	<p>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 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 oder dem Änderungswert identityoverride verwendet werden.</p>
identitymissing	<p>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.</p>

Tabelle 31. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
identityoverride	<p>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. Wird diese Option verwendet, versucht das Dienstprogramm LOAD nicht, die Eindeutigkeit der Werte in der Identitätsspalte der Tabelle zu verwalten oder zu prüfen.</p>
indexfreespace=x	<p><i>x</i> 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. Beim Laden mit INDEXING MODE INCREMENTAL wird diese Option ignoriert. Der erste Eintrag auf einer Seite wird ohne Einschränkung hinzugefügt; nachfolgende Einträge werden mit der Maßgabe hinzugefügt, dass der Schwellenwert für den Prozentsatz des freizuhaltenden Speicherbereichs eingehalten wird. Der Standardwert ist der zum Zeitpunkt der Indexerstellung (CREATE INDEX) verwendete Wert.</p> <p>Dieser Wert hat Vorrang vor dem in der Anweisung CREATE INDEX angegebenen Wert für PCTFREE. Die Option indexfreespace wirkt sich nur auf Indexblattseiten aus.</p>
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. Die Klausel LOBS FROM impliziert die Aktivierung des Verhaltens von lobsinfile. Die Klausel LOBS FROM überträgt die Liste der Pfade an das Dienstprogramm LOAD, die beim Laden der Daten nach den 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</i>.<i>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. Wenn 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 <i>db2exp.001</i> 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>

Tabelle 31. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
noheader	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.) 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.
norowwarnings	Unterdrückt alle Warnungen zu zurückgewiesenen Zeilen.
pagefreespace=x	x 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. wenn für eine Zeile, die mindestens 3.000 Byte lang ist, für x der Wert 50 angegeben ist), wird die Zeile auf einer neuen Seite platziert. Wenn der Wert 100 angegeben wird, wird jede Zeile auf eine neue Seite gestellt. 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.
rowchangetimestampignore	Dieser Änderungswert informiert das Dienstprogramm LOAD darüber, dass für die Zeitmarkenspalte für Zeilenänderung in der Datendatei Daten vorhanden sind, aber ignoriert werden sollen. Dies bewirkt, dass alle Wert für ROW CHANGE TIMESTAMP durch das Dienstprogramm generiert werden. Dieses Verhalten ist für als GENERATED ALWAYS definierte Spalten und als GENERATED BY DEFAULT definierte Spalten gleich. Dies bedeutet, dass bei als GENERATED ALWAYS definierten Spalten keine Zeilen zurückgewiesen werden. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert rowchangetimestampmissing oder rowchangetimestampoverride verwendet werden.
rowchangetimestampmissing	Wird dieser Änderungswert angegeben, geht das Dienstprogramm davon aus, dass die Eingabedatendatei keine Daten (nicht einmal Nullwerte) für die Zeitmarkenspalte für Zeilenänderung enthält, und generiert daher für jede Zeile einen Wert. Dieses Verhalten ist für als GENERATED ALWAYS definierte Spalten und als GENERATED BY DEFAULT definierte Spalten gleich. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert rowchangetimestampignore oder rowchangetimestampoverride verwendet werden.
rowchangetimestampoverride	Dieser Änderungswert sollte nur dann verwendet werden, wenn die zu ladende Tabelle eine als GENERATED ALWAYS definierte Zeitmarkenspalte für Zeilenänderung enthält. Er weist das Dienstprogramm an, entgegen der normalen Regeln für diese Typen von Zeitmarkenspalten für Zeilenänderung, 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 Spalte ROW CHANGE TIMESTAMP zurückgewiesen (SQL3116W). Dieser Änderungswert kann nicht zusammen mit dem Änderungswert rowchangetimestampmissing oder rowchangetimestampignore verwendet werden. Wird diese Option verwendet, versucht das Dienstprogramm LOAD nicht, die Eindeutigkeit der Werte in der Zeitmarkenspalte für Zeilenänderung zu verwalten oder zu prüfen.

Tabelle 31. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
seclabelchar	<p>Gibt an, dass Sicherheitskennsätze in der Eingabequellendatei im Zeichenfolgeformat für Sicherheitskennsatzwerte vorliegen (nicht im codierten numerischen Standardformat). Von LOAD wird jeder einzelne Sicherheitskennsatz beim Laden in das interne Format konvertiert. Liegt eine Zeichenfolge nicht im richtigen Format vor, wird die betreffende Zeile nicht geladen und eine Warnung (SQLSTATE 01H53, SQLCODE SQL3242W) zurückgegeben. Stellt eine Zeichenfolge nicht einen gültigen Sicherheitskennsatz dar, der Bestandteil der zum Schutz der Tabelle verwendeten Sicherheitsrichtlinie ist, wird die betreffende Zeile nicht geladen und eine Warnung (SQLSTATE 01H53, SQLCODE SQL3243W) zurückgegeben.</p> <p>Dieser Änderungswert kann nicht angegeben werden, wenn der Änderungswert seclabelname angegeben ist. Wird diese Kombination von Änderungswerten angegeben, schlägt das Laden fehl, und es wird ein Fehler (SQLCODE SQL3525N) zurückgegeben.</p> <p>Wenn Sie über eine Tabelle verfügen, die eine einzige Spalte DB2SECURITYLABEL enthält, sieht die Datendatei möglicherweise wie folgt aus:</p> <pre>"CONFIDENTIAL:ALPHA:G2" "CONFIDENTIAL;SIGMA:G2" "TOP SECRET:ALPHA:G2"</pre> <p>Zum Laden und Importieren dieser Daten muss der Änderungswert für den Dateityp seclabelchar verwendet werden:</p> <pre>LOAD FROM input.del OF DEL MODIFIED BY SECLABELCHAR INSERT INTO t1</pre>
seclabelname	<p>Gibt an, dass Sicherheitskennsätze in der Eingabequellendatei über ihren Namen angegeben werden und nicht im codierten numerischen Standardformat vorliegen. Von LOAD wird der jeweilige Name in den entsprechenden Sicherheitskennsatz konvertiert, soweit vorhanden. Existiert kein Sicherheitskennsatz mit dem angegebenen Namen für die zum Schutz der Tabelle verwendete Sicherheitsrichtlinie, wird die betreffende Zeile nicht geladen und eine Warnung (SQLSTATE 01H53, SQLCODE SQL3244W) zurückgegeben.</p> <p>Dieser Änderungswert kann nicht angegeben werden, wenn der Änderungswert seclabelchar angegeben ist. Wird diese Kombination von Änderungswerten angegeben, schlägt das Laden fehl, und es wird ein Fehler (SQLCODE SQL3525N) zurückgegeben.</p> <p>Wenn Sie über eine Tabelle verfügen, die eine einzige Spalte DB2SECURITYLABEL enthält, besteht die Datendatei möglicherweise aus Sicherheitskennsatznamen wie den folgenden:</p> <pre>"LABEL1" "LABEL1" "LABEL2"</pre> <p>Zum Laden und Importieren dieser Daten muss der Änderungswert für den Dateityp seclabelname verwendet werden:</p> <pre>LOAD FROM input.del OF DEL MODIFIED BY SECLABELNAME INSERT INTO t1</pre> <p>Anmerkung: Ist der Dateityp ASC, werden alle auf den Namen des Sicherheitskennsatzes folgenden Leerzeichen als Teil des Namens interpretiert. Verwenden Sie den Änderungswert für Dateityp 'striptblanks', um sicherzustellen, dass die Leerzeichen entfernt werden.</p>

Tabelle 31. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
total freespace=x	<p>x 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 x 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. Wenn zwei Ladeoperationen mit dieser Option ausgeführt werden, wird der zusätzliche Speicherbereich, der am Ende durch die erste Ladeoperation angehängt wurde, von der zweiten Ladeoperation nicht erneut verwendet.</p>
usedefaults	<p>Wenn eine Quellenspalte für eine Zieltabellenspalte angegeben wurde, jedoch für mindestens eine Zeileninstanz keine Daten enthält, werden Standardwerte geladen. Beispiele für fehlende Daten sind:</p> <ul style="list-style-type: none"> • Bei DEL-Dateien: Für einen Spaltenwert werden zwei benachbarte Spaltenbegrenzer (",,") oder zwei durch eine beliebige Anzahl Leerzeichen (" , ") getrennte, benachbarte Spaltenbegrenzer angegeben. • Bei DEL/ASC/WSF-Dateien: Eine Zeile, die für die ursprüngliche Angabe zu wenig Spalten enthält oder nicht lang genug ist. Bei Dateien des Typs ASC werden NULL-Spaltenwerte nicht als explizit fehlend betrachtet, und es wird kein Standardwert für NULL-Spaltenwerte eingesetzt. NULL-Spaltenwerte können durch alle Leerzeichen für numerische Spalten, Datenspalten, Zeitspalten und Zeitmarkenspalten oder durch die Verwendung des Nullindikators für Spalten eines beliebigen Typs zur Angabe eines Spaltenwerts von NULL dargestellt werden. <p>Ohne diese Option geschieht Folgendes, wenn eine Quellenspalte keine Daten für eine Zeileninstanz enthält:</p> <ul style="list-style-type: none"> • Für DEL/ASC/WSF-Dateien: Wenn die Spalte Nullwerte enthalten darf, wird NULL geladen. Wenn die Spalte keine Nullwerte enthalten darf, wird die Zeile vom Dienstprogramm zurückgewiesen.

Tabelle 32. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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>Es gelten folgende Regeln:</p> <ul style="list-style-type: none"> • Bei reinem DBCS (Grafik), gemischtem DBCS und EUC sind Begrenzer auf den Bereich von x00 bis x3F (jeweils einschließlich) beschränkt. • Bei DEL-Daten, die in einer EBCDIC-Codepage angegeben werden, sind die Begrenzer nicht unbedingt mit den DBCS-Startzeichen und -Endezeichen identisch. • 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. <p>Die Verwendung dieser Option in Verbindung mit dem Dateityp CURSOR wird nicht unterstützt.</p>

Tabelle 32. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
dateformat="x"	<p>x ist das Format des Datums in der Quellendatei.¹ 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 01 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 01 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: "D-M-YYYY" "MM.DD.YYYY" "YYYYDDD"</p>
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 Instanzeigner. Die Standarddateiberechtigungen können mit dem Änderungswert für Dateityp dumpfileaccessall überschrieben werden.</p> <p>Anmerkung:</p> <ol style="list-style-type: none"> 1. In einer Umgebung mit partitionierter Datenbank muss der Pfad für die ladende Datenbankpartition lokal sein, damit gleichzeitig aktive Ladeoperationen nicht versuchen, in dieselbe Datei zu schreiben. 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. 3. Wenn die angegebene Datei bereits vorhanden ist, wird sie nicht erneut erstellt, sondern abgeschnitten.
dumpfileaccessall	<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"> 1. wenn er zusammen mit dem Änderungswert für Dateityp dumpfile verwendet wird, 2. der Benutzer über das Zugriffsrecht SELECT für die Ladezieltabelle verfügt, 3. wenn er in einer DB2-Serverdatenbankpartition abgesetzt wird, die sich auf einem UNIX-Betriebssystem befindet. <p>Wenn die angegebene Datei bereits vorhanden ist, werden die für sie bestehenden Berechtigungen nicht geändert.</p>

Tabelle 32. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
fastparse	Diese Werte sollten mit Bedacht verwendet werden. Reduziert die Syntaxprüfung für Spaltenwerte, die vom Benutzer angegeben wurden, und verbessert die Leistung. Die korrekte Architektur der Tabellen wird sichergestellt (die vom Dienstprogramm vorgenommene Datenprüfung reicht aus, um Segmentierungsverstöße und Traps zu verhindern), aber die Datenkohärenz wird nicht geprüft. Verwenden Sie diese Option nur, wenn Sie sicher sind, dass Ihre Daten kohärent und korrekt sind. Enthalten die vom Benutzer angegebenen Daten beispielsweise den ungültigen Wert :1>0-00-20-07.11.12.000000 für die Zeitmarkenspalte, wird dieser Wert in die Tabelle eingefügt, wenn fastparse angegeben ist, und zurückgewiesen, wenn fastparse nicht angegeben ist.
implieddecimal	Die Position eines implizierten Dezimalzeichens wird durch die Spaltendefinition bestimmt. Sie wird nicht mehr am Ende des Werts angenommen. Beispielsweise wird der Wert 12345 in eine DECIMAL(8,2)-Spalte als 123,45 geladen und <i>nicht</i> als 12345,00. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert packeddecimal verwendet werden.
timeformat="x"	x ist das Format der Zeitangabe in der Quelldatei. ¹ Gültige Zeitelemente sind: H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 - 24 bei einem 24-Stunden-System) HH - Stunde (zwei Ziffern von 00 bis 12 bei einem 12-Stunden-System und 00 - 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus) M - Minute (eine oder zwei Ziffern von 0 bis 59) MM - Minute (zwei Ziffern von 00 bis 59; dieses Element und das Element M schließen sich gegenseitig aus) S - Sekunde (eine oder zwei Ziffern von 0 bis 59) SS - Sekunde (zwei Ziffern von 00 bis 59; dieses Element und das Element S schließen sich gegenseitig aus) SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86400; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus) TT - Meridiananzeiger (AM oder PM) Für jedes Element, das nicht angegeben wird, wird der Standardwert 0 zugeordnet. Einige Beispiele für Zeitformate: "HH:MM:SS" "HH.MM TT" "SSSSS"

Tabelle 32. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
timestampformat="x"	<p>x ist das Format der Zeitmarke in der Quellendatei.¹ 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 01 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 00 bis 12 bei einem 12-Stunden-System und 00 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 00 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 00 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 - 86400; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus)</p> <p>U (1 bis 12 mal)</p> <p>- Bruchsekunden (die Häufigkeit von U gibt die Anzahl der Stellen mit Werten aus dem Bereich 0 bis 9 an)</p> <p>TT - Meridiananzeiger (AM oder PM)</p>

Tabelle 32. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
<p>timestampformat="x" (Fortsetzung)</p>	<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>Wenn der Änderungswert timestampformat nicht angegeben wird, verwendet das Dienstprogramm LOAD für das Zeitmarkenfeld eines von zwei möglichen Formaten:</p> <pre>YYYY-MM-DD-HH.MM.SS YYYY-MM-DD HH:MM:SS</pre> <p>Das Dienstprogramm LOAD wählt das Format entsprechend dem Trennzeichen zwischen DD und HH. Ist das Trennzeichen ein Gedankenstrich '-', wird vom Dienstprogramm LOAD das Format mit Gedankenstrichen und Punkten verwendet (YYYY-MM-DD-HH.MM.SS). Ist das Trennzeichen ein Leerzeichen, erwartet das Dienstprogramm LOAD einen Doppelpunkt ':' zwischen den Angaben HH, MM und SS.</p> <p>In beiden Formaten erwartet das Dienstprogramm LOAD den Punkt '.' als Trennzeichen, wenn Sie das Feld für Mikrosekunden (UUUUUU) einschließen. YYYY-MM-DD-HH.MM.SS.UUUUUU oder YYYY-MM-DD HH:MM:SS.UUUUUU ist zulässig.</p> <p>Das folgende Beispiel veranschaulicht, wie Daten mit benutzerdefinierten Datums- und Zeitformaten in eine Tabelle namens SCHEDULE geladen werden:</p> <pre>db2 load from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>
<p>usegraphiccodepage</p>	<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 Recovery nach einer DROP TABLE-Operation generiert wird, wenn die wiederherzustellende Tabelle Grafikdaten enthält.</p> <p>Einschränkungen</p> <p>Der Änderungswert usegraphiccodepage DARF NICHT für DEL-Dateien angegeben werden, die mit dem Dienstprogramm EXPORT erstellt wurden, da die in diesen Dateien enthaltenen Daten nur in einer Codepage codiert sind. Der Änderungswert usegraphiccodepage wird auch von den großen Doppelbytezeichenobjekten (DBCLOBs) in Dateien ignoriert.</p>

Tabelle 32. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
xmlchar	<p>Gibt an, dass XML-Dokumente in der Zeichencodepage codiert sind.</p> <p>Diese Option eignet sich für die Verarbeitung von XML-Dokumenten, die in der angegebenen Zeichencodepage codiert sind, ohne eine Codierdeklaration zu enthalten.</p> <p>Bei jedem Dokument muss die Codierung, soweit eine Deklarationskennung mit Codierungsattribut existiert, der Zeichencodepage entsprechen. Ist dies nicht der Fall, wird die Zeile mit dem Dokument zurückgewiesen. Dabei ist zu beachten, dass die Zeichencodepage dem Wert entspricht, der für den Änderungswert für Dateityp codepage angegeben ist, oder der Codepage der Anwendung, wenn dieser Wert nicht angegeben ist. Die Dokumente sind standardmäßig entweder in Unicode kodiert oder enthalten eine Deklarationskennung mit Codierungsattribut.</p>
xmlgraphic	<p>Gibt an, dass XML-Dokumente in der angegebenen Grafikcodepage codiert sind.</p> <p>Diese Option eignet sich für die Verarbeitung von XML-Dokumenten, die in einer bestimmten Grafikcodepage codiert sind, ohne eine Codierdeklaration zu enthalten.</p> <p>Bei jedem Dokument muss die Codierung, soweit eine Deklarationskennung mit Codierungsattribut existiert, der Grafikcodepage entsprechen. Ist dies nicht der Fall, wird die Zeile mit dem Dokument zurückgewiesen. Dabei ist zu beachten, dass die Grafikcodepage der Grafikkomponente des Werts entspricht, der über den Änderungswert für Dateityp codepage angegeben wird, oder der Grafikkomponente der Anwendungscodepage, wenn dieser Wert nicht angegeben ist. Dokumente sind standardmäßig entweder in Unicode kodiert oder enthalten eine Deklarationskennung mit Codierungsattribut.</p>

Tabelle 33. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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.</p> <p>Es gelten folgende Regeln:</p> <ul style="list-style-type: none"> • Es wird keine Konvertierung zwischen Datentypen durchgeführt, mit Ausnahme von BIGINT, INTEGER und SMALLINT. • Die Datenlänge muss mit der jeweiligen Zielspaltendefinition übereinstimmen. • FLOAT-Werte müssen im IEEE-Gleitkommaformat vorliegen. • Binärdaten in der Ladequellendatei werden als Big-Endian gewertet, unabhängig von der Plattform, auf der die Ladeoperation ausgeführt wird. <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>

Tabelle 33. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASC-Dateiformate (ASCII-Format mit universellen Zeilenbegrenzern) (Forts.)

Änderungswert	Beschreibung
nochecklengths	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.
nullindchar=x	<p>x ist ein einzelnes Zeichen. Ändert das Zeichen, das einen Nullwert bezeichnet, in x. Der Standardwert für x ist Y.²</p> <p>Bei diesem Änderungswert muss für EBCDIC-Datendateien die Groß-/Kleinschreibung beachtet werden, außer wenn das Zeichen ein englischer Buchstabe ist. Wenn beispielsweise für das NULL-Anzeigerzeichen der Buchstabe N angegeben wird, wird auch n als NULL-Anzeiger erkannt.</p>
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.</p> <p>Unterstützte Werte für das Vorzeichenhalbbyte sind:</p> <p style="margin-left: 40px;">+ = 0xC 0xA 0xE 0xF - = 0xD 0xB</p> <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>Binärdaten in der Ladequellendatei werden als Big-Endian gewertet, unabhängig von der Serverplattform, auf der die Ladeoperation ausgeführt wird. Bei Verwendung dieses Änderungswerts unter Windows-Betriebssystemen darf die Byteanordnung jedoch nicht umgekehrt 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. Diese Option ersetzt die veraltete Option t, die nur noch aus Gründen der Abwärtskompatibilität 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. Diese Option ersetzt die veraltete Option padwithzero, die nur noch aus Gründen der Abwärtskompatibilität unterstützt wird.</p>

Tabelle 33. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASC-Dateiformate (ASCII-Format mit universellen Zeilenbegrenzern) (Forts.)

Änderungswert	Beschreibung
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.</p> <p>Die folgenden Halbbytevorzeichenwerte sind möglich: + = 0xC 0xA 0xE 0xF 0x3 - = 0xD 0xB 0x7</p> <p>Für Ziffern werden Werte von 0x0 bis 0x9 unterstützt.</p> <p>Für Zonen werden die Werte 0x3 und 0xF unterstützt.</p>

Tabelle 34. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: DEL-Dateiformate (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 anstelle von doppelten Anführungszeichen verwendet, um eine Zeichenfolge einzuschließen.²³ Falls Sie die doppelten Anführungszeichen explizit als Zeichenfolgebegrenzer angeben wollen, müssen Sie dies folgendermaßen tun: modified by chardel""</p> <p>Auch das einfache Anführungszeichen (') kann wie folgt als Zeichenfolgebegrenzer festgelegt werden: modified by chardel''</p>
coldelx	<p><i>x</i> ist ein einzelnes Zeichen, das als Spaltenbegrenzer verwendet wird. Der Standardwert ist ein Komma (,). Das angegebene Zeichen wird anstelle eines Kommas verwendet, um das Ende einer Spalte anzugeben.²³</p>
decplusblank	<p>Zeichen für Pluszeichen. Bewirkt, dass positiven Dezimalwerten statt eines Pluszeichens (+) ein Leerzeichen vorangestellt wird. Die Standardaktion besteht darin, positiven Dezimalwerten ein Pluszeichen voranzustellen.</p>
decptx	<p><i>x</i> ist ein einzelnes Ersatzzeichen für den Punkt als Dezimalzeichen. Der Standardwert ist ein Punkt (.). Das angegebene Zeichen wird anstelle eines Punkts als Dezimalzeichen verwendet.²³</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: db2 load ... modified by delprioritychar ...</p> <p>Beispiel anhand der folgenden DEL-Datendatei: "Smith, Joshua",4000,34.98<zeilenbegrenzer> "Vincent,<zeilenbegrenzer>, is a manager", 4005,44.37<zeilenbegrenzer></p> <p>Wenn der Änderungswert delprioritychar angegeben wird, gibt es in dieser Datendatei nur zwei Zeilen. Der zweite <zeilenbegrenzer> wird als Teil der ersten Datenspalte der zweiten Zeile gewertet, während der erste und dritte <zeilenbegrenzer> als eigentliche Satzbegrenzer gewertet werden. Wenn dieser Änderungswert <i>nicht</i> angegeben wird, gibt es in dieser Datendatei drei Zeilen, jeweils getrennt durch einen <zeilenbegrenzer>.</p>

Tabelle 34. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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>
nocharde1	<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 mithilfe eines DB2-Datenbanksystems exportiert wurden (es sei denn, nocharde1 war beim Exportieren angegeben). 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 nodoublede1 angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p>
nodoublede1	Unterdrückt die Erkennung von doppelten Zeichenbegrenzern.

Tabelle 35. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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>

Anmerkung:

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.

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 \).

2. Für die Dateitypänderungswerte **chardel**, **coldel** oder **decpt** angegebene Zeichenwerte müssen 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 ...
```

3. „Aspekte beim Versetzen von XML-Daten - Begrenzer“ listet Einschränkungen für Zeichen auf, die anstelle der standardmäßigen Begrenzer verwendet werden können.
4. Das Dienstprogramm LOAD gibt keine Warnung aus, 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.
5. Beim Import in eine Tabelle, die eine implizit ausgeblendete Zeitmarkenspalte für Zeilenänderung enthält, wird das Merkmal **IMPLICITLY HIDDEN** der Spalte nicht berücksichtigt. Daher *muss* der Dateitypänderungswert **rowchangetimestampmissing** im Befehl **IMPORT** angegeben werden, wenn die zu importierenden Daten keine Daten für die Spalte enthalten und keine explizite Spaltenliste vorhanden ist.

Tabelle 36. 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 wird.

Tabelle 36. Verhalten des Dienstprogramms LOAD bei Verwendung von 'codepage' und 'usegraphiccodepage' (Forts.)

codepage=N	usegraphiccodepage	Verhalten von LOAD
Vorhanden	Nicht vorhanden	Es wird angenommen, dass sich alle Daten in der Datei in der Codepage N befinden. Warnung: Grafikdaten werden beim Laden in die Datenbank beschädigt, wenn N eine Einzelbyte-Codepage ist.
Nicht vorhanden	Vorhanden	Es wird angenommen, dass Zeichendaten in der Datenbankcodepage vorliegen, selbst wenn die Option CLIENT angegeben wird. Es wird angenommen, dass Grafikdaten in der Codepage der Datenbankgrafikdaten vorliegen, selbst wenn die Option CLIENT angegeben wird. Wenn die Datenbankcodepage eine Einzelbyte-Codepage ist, wird angenommen, dass sich alle Daten in der Datenbankcodepage befinden. Warnung: 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. Warnung: Grafikdaten werden beim Laden in die Datenbank beschädigt, wenn N eine Einzelbyte-Codepage ist.

Befehl LOAD mit Prozedur ADMIN_CMD

Lädt Daten in eine DB2-Tabelle.

Die Daten können auf dem Server in Form einer Datei, eines Bands oder einer benannten Pipe gespeichert sein. Ist das Attribut COMPRESS für die Tabelle auf YES gesetzt, werden die geladenen Daten auf allen Daten- und Datenbankpartitionen komprimiert, für die in der Tabelle ein Wörterbuch existiert, einschließlich der Daten im XML-Speicherobjekt der Tabelle.

Quick Link zu „Änderungswerte für Dateityp für das Dienstprogramm LOAD“ auf Seite 303.

Einschränkungen

Das Dienstprogramm LOAD unterstützt das Laden von Daten auf Hierarchieebene nicht. Das Dienstprogramm LOAD ist mit Bereichsclustertabellen nicht kompatibel. Das Dienstprogramm LOAD unterstützt den Parameter NOT LOGGED INITIALLY für die Anweisung CREATE TABLE oder ALTER TABLE nicht.

Geltungsbereich

Dieser Befehl kann in einer einzigen Anforderung für mehrere Datenbankpartitionen abgesetzt werden.

Berechtigung

Eine der folgenden Berechtigungen:

- DATAACCESS
- Berechtigung LOAD für die Datenbank und folgende Zugriffsrechte:
 - 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.
- Um Daten in eine Tabelle laden zu können, die geschützte Spalten enthält, muss die Berechtigungs-ID der Sitzung direkt oder indirekt über ein Gruppe oder Rolle über LBAC-Berechtigungs nachweise verfügen, die Schreibzugriff auf alle geschützten Spalten in der Tabelle zulassen. Andernfalls schlägt das Laden fehl, und es wird ein Fehler (SQLSTATE 5U014) zurückgegeben.
- Um Daten in eine Tabelle laden zu können, die geschützte Zeilen enthält, muss für die Berechtigungs-ID der Sitzung ein Sicherheitskennsatz vorliegen, der folgenden Kriterien entspricht:
 - Der Sicherheitskennsatz ist Teil der Sicherheitsrichtlinie, die die Tabelle schützt.
 - Der Sicherheitskennsatz wurde der Berechtigungs-ID der Sitzung direkt oder indirekt über eine Gruppe oder Rolle für Schreibzugriff oder für alle Zugriffe erteilt.

Liegt für die Berechtigungs-ID der Sitzung kein derartiger Sicherheitskennsatz vor, schlägt das Laden fehl und es wird ein Fehler (SQLSTATE 5U014) zurückgegeben. Der Sicherheitskennsatz schützt geladene Zeilen, wenn die LBAC-Berechtigungs nachweise der Berechtigungs-ID der Sitzung einen Schreibzugriff auf den Sicherheitskennsatz, der die jeweilige Zeile in den Daten schützt, nicht zulassen. Dies geschieht jedoch nicht, wenn die Sicherheitsrichtlinie zum Schutz der Tabelle mit der Option RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL der Anweisung CREATE SECURITY POLICY erstellt wurde. In diesem Fall schlägt das Laden fehl, und es wird ein Fehler (SQLSTATE 42519) zurückgegeben.

Wenn Sie Daten in eine Tabelle mit geschützten Zeilen laden, enthält die Zieltabelle eine einzige Spalte mit dem Datentyp DB2SECURITYLABEL. Wenn die Eingabedatenzeile keinen Wert für diese Spalte enthält, wird die Zeile zurückgewiesen, es sei denn, der Ladebefehl wurde mit dem Änderungswert `usedefaults` für den Dateityp angegeben. Wurde `usedefaults` im Ladebefehl angegeben, wird der Sicherheitskennsatz, über den Sie für den Schreibzugriff verfügen, aus der Sicherheitsrichtlinie, die die Tabelle schützt, verwendet. Wenn Sie keinen Sicherheitskennsatz für den Schreibzugriff haben, wird die Zeile zurückgewiesen und mit der Verarbeitung der nächsten Zeile fortgefahren.

- Bei Angabe der Option REPLACE muss die Berechtigungs-ID der Sitzung über die Berechtigung zum Löschen der Tabelle verfügen.

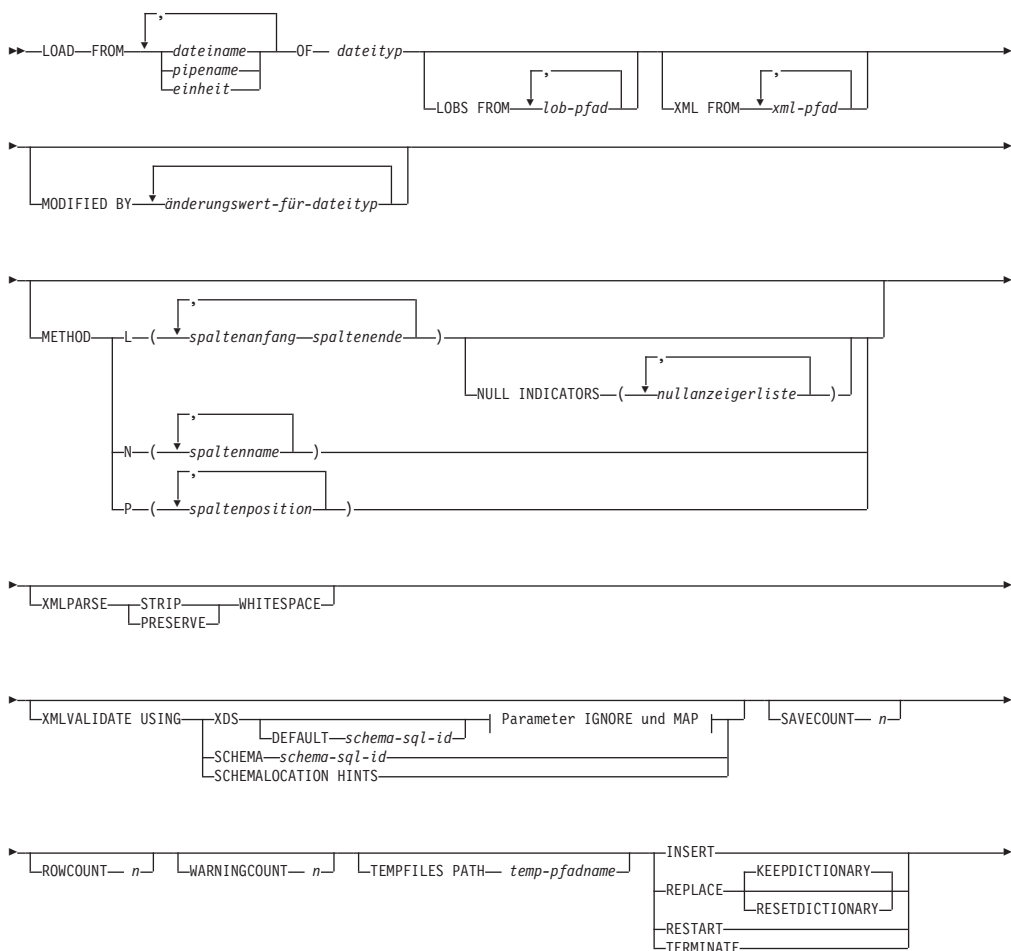
- Bei Angabe der Option LOCK WITH FORCE ist die Berechtigung SYSADM erforderlich.

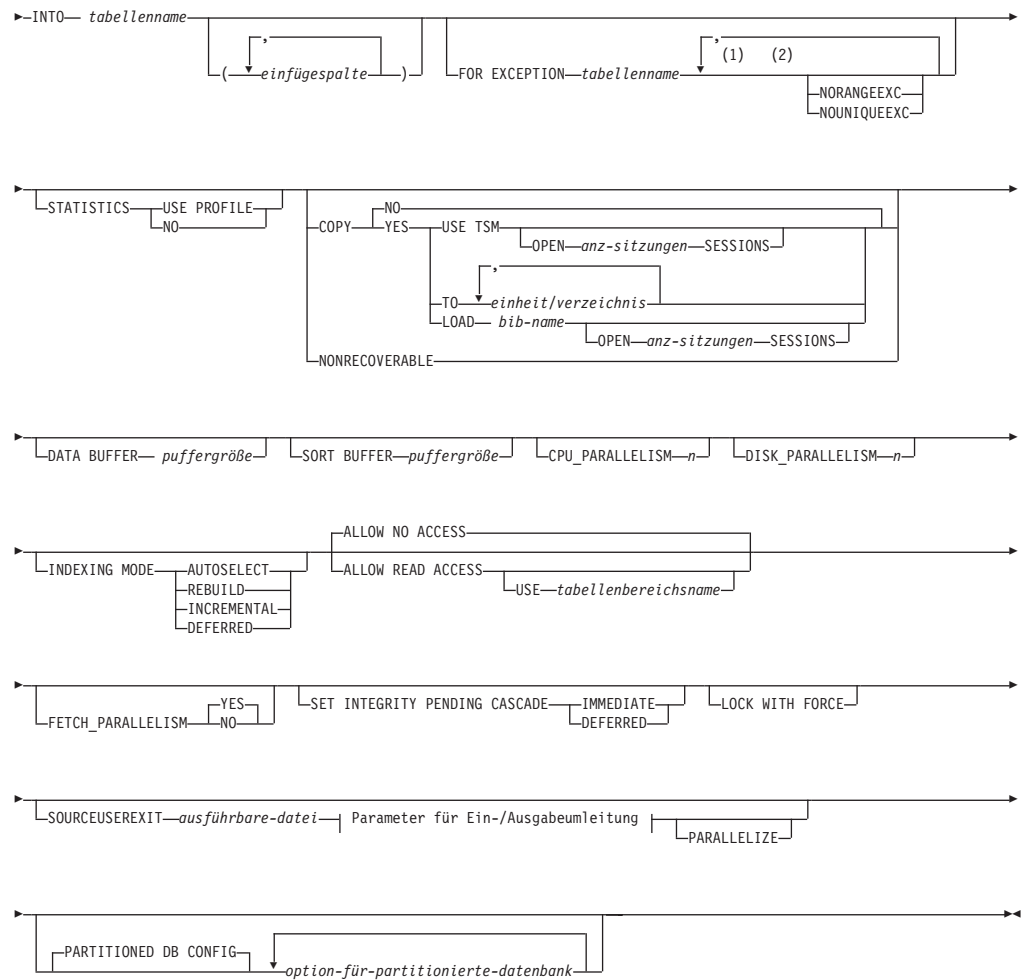
Da der Instanzeigner auch Eigner aller LOAD-Prozesse (und im Allgemeinen auch aller DB2-Serverprozesse) ist, und all diese Prozesse die Kennung des Instanzeigners verwenden, um auf benötigte Dateien zuzugreifen, muss der Instanzeigner über Lesezugriff auf Eingabedatendateien verfügen. Diese Eingabedatendateien müssen für den Instanzeigner lesbar sein (unabhängig davon, wer den Befehl aufruft).

Erforderliche Verbindung

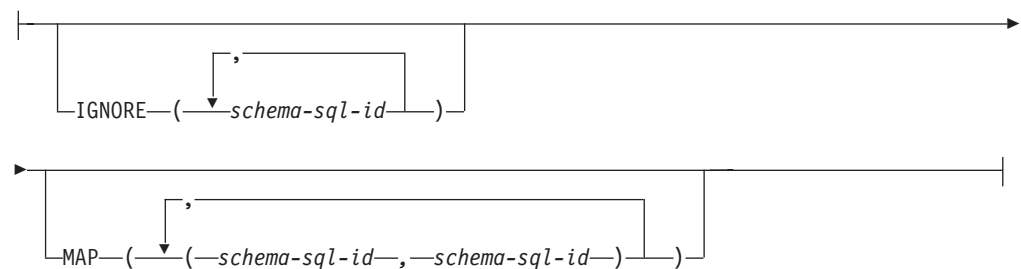
Instanz. Eine explizite Zuordnung ist nicht erforderlich. Wenn eine Verbindung zur Datenbank hergestellt wurde, wird versucht, eine implizite Zuordnung zur lokalen Instanz vorzunehmen.

Befehlssyntax

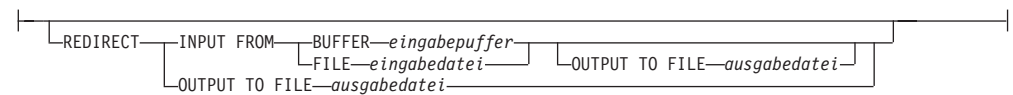




Parameter IGNORE und MAP:



Parameter für Ein-/Ausgabeumleitung:



Anmerkungen:

- 1 Diese Schlüsselwörter können in einer beliebigen Reihenfolge vorliegen.
- 2 Jedes Schlüsselwort darf nur einmal vorkommen.

Befehlsparameter

FROM *dateiname* | *pipename* | *einheit*

Anmerkung:

- Werden die Daten mit dem Befehl **EXPORT** und der Prozedur ADMIN_CMD in eine Datei exportiert, ist die ID des abgeschirmten Benutzers Eigner der Datendatei. Auf diese Datei kann der Instanzeigner im Allgemeinen nicht zugreifen. Zum Ausführen des Befehls **LOAD** über den CLP oder die Prozedur ADMIN_CMD muss ein Zugriff auf die Datendatei über die ID des Instanzeigners möglich sein. Dem Instanzeigner muss deshalb mit Grant Lesezugriff auf die Datendatei erteilt werden.
- 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 eine logische Datei angesehen, wenn sie alle durch einen Aufruf des Befehls **EXPORT** erstellt werden.)
- Beim Laden von XML-Daten in Tabellen in einer Umgebung mit partitionierten Datenbanken müssen die XML-Datendateien über Lesezugriff auf alle Datenbankpartitionen verfügen, in denen das Laden stattfindet.

OF *dateityp*

Gibt das Format der Daten an:

- ASC (ASCII-Format mit universellen Zeilenbegrenzern)
- DEL (ASCII-Format ohne universelle Zeilenbegrenzer)
- IXF (Integration Exchange Format, PC-Version) ist ein Binärformat, das ausschließlich von DB2-Datenbanken verwendet wird.
- CURSOR (ein Cursor, der für eine Anweisung SELECT oder VALUES deklariert wird).

Anmerkung: Wenn Sie XML-Daten mit dem Dateityp CURSOR in eine Tabelle in einer verteilten Datenbankumgebung laden, werden die Modi PARTITION_ONLY und LOAD_ONLY nicht unterstützt.

LOBS FROM *lob-pfad*

Der Pfad zu den Datendateien mit LOB-Werten, die geladen werden sollen. Der Pfad muss mit einem Schrägstrich enden. Die Namen der LOB-Datendateien werden in der Hauptdatendatei (ASC, DEL oder IXF) in der Spalte gespeichert, die in die LOB-Spalte geladen wird. Es können maximal 999 Pfade angegeben werden. Dadurch wird implizit das Verhalten **LOBSINFILE** aktiviert.

Diese Option wird ignoriert, wenn sie zusammen mit dem Dateityp CURSOR angegeben wird.

MODIFIED BY *änderungswert-für-dateityp*

Gibt Optionen für Änderungswerte für Dateityp an. Siehe „Änderungswerte für Dateityp für das Dienstprogramm LOAD“ auf Seite 303.

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. Die Nummerierung beginnt bei 1. Diese Methode kann nur mit ASC-Dateien verwendet werden und ist die einzige gültige Methode für diesen Dateityp.

NULL INDICATORS *nullanzeigerliste*

Diese Option kann nur verwendet werden, wenn der Parameter **METHOD L** angegeben wird (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 einen Eintrag für jedes Datenfeld enthalten, das im Parameter **METHOD L** definiert wird. Eine Spaltennummer 0 gibt an, dass das entsprechende Datenfeld immer Daten enthält.

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

Das Zeichen für den Nullanzeiger kann über die Option **MODIFIED BY** geändert werden.

- 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. 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. Diese Methode kann nur bei den Dateitypen IXF, DEL oder CURSOR verwendet werden und ist die einzige zulässige Methode für den Dateityp DEL.

XML FROM *xml-pfad*

Gibt mindestens einen Pfad mit XML-Dateien an. XML-Datenkennungen (XDSs) sind in der Hauptdatendatei (ASC, DEL oder IXF) in der Spalte enthalten, die in die XML-Spalte geladen wird.

XMLPARSE

Gibt an, wie die XML-Dokumente syntaktisch analysiert werden. Wenn diese Option nicht angegeben ist, wird das Parsingverhalten für XML-Dokumente durch den Wert des Sonderregisters CURRENT XMLPARSE OPTION festgelegt.

STRIP WHITESPACE

Gibt an, dass Leerzeichen bei der Syntaxanalyse des XML-Dokuments entfernt werden sollen.

PRESERVE WHITESPACE

Gibt an, dass Leerzeichen bei der Syntaxanalyse des XML-Dokuments nicht entfernt werden sollen.

XMLVALIDATE

Gibt an, dass XML-Dokumente anhand eines Schemas geprüft werden sollen (soweit möglich).

USING XDS

XML-Dokumente werden anhand des XML-Schemas geprüft, das über die XDS in der Hauptdatendatei angegeben ist. Wenn die Option **XMLVALIDATE** mit der Klausel **USING XDS** aufgerufen wird, wird das für die Prüfung verwendete Schema standardmäßig durch das SCH-Attribut der XDS festgelegt. Enthält die XDS kein SCH-Attribut, wird eine Schemaprüfung nur durchgeführt, wenn die Klausel **DEFAULT** ein Standardschema angibt.

Mit den Klauseln **DEFAULT**, **IGNORE** und **MAP** kann das Verhalten der Schemafestlegung geändert werden. Diese optionalen Klauseln werden nicht auf einander, sondern direkt auf die XDS-Spezifikationen angewendet. Wird ein Schema beispielsweise ausgewählt, weil es von der Klausel **DEFAULT** angegeben wurde, so wird dieses Schema nicht ignoriert, wenn es auch von der Klausel **IGNORE** angegeben wird. Wird ein Schema ausgewählt, weil es als erster Teil eines Zuordnungspaars in der Klausel **MAP** angegeben wurde, so wird dieses Schema nicht erneut zugeordnet, wenn es auch im zweiten Teil des Zuordnungspaars einer anderen Klausel **MAP** angegeben ist.

USING SCHEMA *schema-sql-id*

XML-Dokumente werden anhand des XML-Schemas mit der angegebenen SQL-ID geprüft. In diesem Fall wird das Attribut SCH der XDS für alle XML-Spalten ignoriert.

USING SCHEMALOCATION HINTS

XML-Dokumente werden anhand des XML-Schemas geprüft, das über die Angaben zur XML-Schemaposition in den XML-Quelldokumenten angegeben ist. Wenn in dem XML-Dokument kein Attribut für die Schemaposition gefunden wird, findet keine Prüfung statt. Bei Angabe der Klausel **USING SCHEMALOCATION HINTS** wird das SCH-Attribut der XML-Datenkennung (XDS) für alle XML-Spalten ignoriert.

Siehe hierzu die Beispiele im unten stehenden Abschnitt zur Option **XMLVALIDATE**.

IGNORE *schema-sql-id*

Diese Option kann nur verwendet werden, wenn der Parameter **USING XDS** angegeben wird. Die Klausel **IGNORE** gibt eine Liste mit mindestens einem Schema an, das ignoriert werden soll, wenn es durch ein SCH-Attribut identifiziert wird. Wenn in der XML-Datenkennung ein SCH-Attribut für ein geladenes XML-Dokument enthalten ist und das von dem SCH-Attribut identifizierte Schema in der Liste der zu ignorierenden Schemata enthalten ist, wird für das geladene XML-Dokument keine Schemaprüfung durchgeführt.

Anmerkung:

Wenn in der Klausel **IGNORE** ein Schema angegeben ist, darf dieses nicht zusätzlich im linken Teil eines Schemapaars in der Klausel **MAP** enthalten sein.

Die Klausel **IGNORE** gilt nur für die XDS. Ein von der Klausel **MAP** zugeordnetes Schema wird anschließend nicht ignoriert, wenn es von der Klausel **IGNORE** angegeben wird.

DEFAULT *schema-sql-id*

Diese Option kann nur verwendet werden, wenn der Parameter **USING XDS** angegeben wird. Das über die Klausel **DEFAULT** angegebene Schema gibt an, welches Schema für die Prüfung verwendet werden soll, wenn die XML-Datenkennung (XDS) eines geladenen XML-Dokuments kein SCH-Attribut enthält, das ein XML-Schema identifiziert.

Die Klausel **DEFAULT** hat Vorrang vor den Klauseln **IGNORE** und **MAP**. Wenn eine XDS die Klausel **DEFAULT** erfüllt, werden die Angaben **IGNORE** und **MAP** ignoriert.

MAP *schema-sql-id*

Diese Option kann nur verwendet werden, wenn der Parameter **USING XDS** angegeben wird. Geben Sie über die Klausel **MAP** alternative Schemata an, die anstelle der Schemata verwendet werden sollen, die im SCH-Attribut einer XML-Datenkennung (XDS) für jedes geladene XML-Dokument angegeben sind. Die Klausel **MAP** gibt eine Liste mit mindestens einem Schemapaar an, in der jedes Paar eine Zuordnung eines Schemas zu einem anderen Schema darstellt. Das erste Schema in dem Zuordnungspaar stellt ein Schema dar, auf das durch ein SCH-Attribut in einer XDS verwiesen wird. Das zweite Schema in dem Zuordnungspaar gibt an, welches Schema für die Durchführung der Schemaprüfung verwendet werden soll.

Wenn im linken Teil eines Schemapaars in der Klausel **MAP** ein Schema angegeben ist, darf dieses nicht zusätzlich in der Klausel **IGNORE** angegeben werden.

Sobald eine Schemapaarzuordnung angewendet wurde, ist das Ergebnis endgültig. Die Zuordnungsoperation ist nicht transitiv, das gewählte Schema wird deshalb nicht anschließend auf eine andere Schemapaarzuordnung angewendet.

Ein Schema kann nicht mehrmals zugeordnet werden, d. h. ein Schema kann nur bei einem einzigen Paar als erstes Schema angegeben sein.

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.

Diese Option ist nicht zulässig, wenn sie zusammen mit dem Dateityp **CURSOR** angegeben wird und wenn eine Tabelle mit einer XML-Spalte geladen wird.

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.

WARNINGCOUNT *n*

Stoppt die Ladeoperation 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 Ladedatei oder die Zieltabelle falsch angegeben, generiert das Dienstprogramm LOAD eine Warnung für jede Zeile, die es zu laden versucht, sodass die Ladeoperation 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.

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:

- 136 Byte für jede Nachricht, die vom Dienstprogramm LOAD generiert wird
- 15 KB Systemaufwand, wenn die Datendatei Langfelddaten oder LOBs enthält. Dieser Anteil kann stark zunehmen, wenn die Option **INSERT** angegeben ist und die Tabelle bereits viele Spalten mit Langfeld- oder LOB-Daten enthält.

INSERT Einer von vier Modi, in denen das Dienstprogramm LOAD ausgeführt werden kann. Fügt die geladenen Daten zur Tabelle hinzu, ohne die vorhandenen Tabellendaten zu ändern.

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.

KEEPDICTIONARY

Ein vorhandenes Komprimierungswörterbuch wird über die Operation **LOAD REPLACE** hinaus beibehalten. Wenn das Tabellenattribut **COMPRESS** auf YES gesetzt ist, werden die neu ersetzten Daten mithilfe des Wörterbuchs komprimiert, das vor dem Aufrufen des Befehls LOAD vorhanden war. Wenn vorher kein Wörterbuch in der Tabelle vorhanden war, wird anhand der Daten, die in die Tabelle ersetzt werden, ein neues Wörterbuch erstellt, sofern das Ta-

Wörterbuchattribut COMPRESS auf YES gesetzt ist. Welches Datenvolumen in diesem Fall zum Erstellen des Komprimierungswörterbuchs erforderlich ist, hängt von den Richtlinien für die automatische Erstellung von Komprimierungswörterbüchern (ADC) ab. Diese Daten werden in nicht komprimierter Form in die Tabelle eingetragen. Sobald das Wörterbuch in die Tabelle eingefügt ist, werden die übrigen zu ladenden Daten mithilfe dieses Wörterbuchs komprimiert. Dies ist der Standardparameter. Eine Zusammenfassung finden Sie in der nachfolgenden Tabelle.

Tabelle 37. LOAD REPLACE KEEPDICTIONARY

Komprimieren	Wörterverzeichnis der Tabellenzeilendaten ist vorhanden	Wörterverzeichnis der XML-Speicherobjekte ist vorhanden ¹	Komprimierungswörterverzeichnis	Datenkomprimierung
JA	JA	JA	Wörterverzeichnis der Tabellenzeilendaten und XML-Wörterverzeichnis beibehalten.	Zu ladende Daten werden komprimiert.
JA	JA	NEIN	Wörterverzeichnis für Tabellenzeilendaten beibehalten und neues XML-Wörterverzeichnis erstellen.	Zu ladende Tabellenzeilendaten werden komprimiert. Nach dem Erstellen des XML-Wörterverzeichnisses werden verbleibende XML-Daten, die geladen werden sollen, komprimiert.
JA	NEIN	JA	Wörterverzeichnis der Tabellenzeilendaten erstellen und XML-Wörterverzeichnis beibehalten.	Nach dem Erstellen des Wörterverzeichnisses der Tabellenzeilendaten werden verbleibende Tabellenzeilendaten, die geladen werden sollen, komprimiert. Zu ladende XML-Daten werden komprimiert.
JA	NEIN	NEIN	Wörterverzeichnis der Tabellenzeilendaten und XML-Wörterverzeichnis erstellen.	Nach dem Erstellen der Wörterverzeichnisse werden verbleibende Daten, die geladen werden sollen, komprimiert.
NEIN	JA	JA	Wörterverzeichnis der Tabellenzeilendaten und XML-Wörterverzeichnis beibehalten.	Zu ladende Daten werden nicht komprimiert.
NEIN	JA	NEIN	Wörterverzeichnis der Tabellenzeilendaten beibehalten.	Zu ladende Daten werden nicht komprimiert.
NEIN	NEIN	JA	Keine Auswirkung auf das Wörterverzeichnis der Tabellenzeilendaten. XML-Wörterverzeichnis beibehalten.	Zu ladende Daten werden nicht komprimiert.
NEIN	NEIN	NEIN	Keine Auswirkung.	Zu ladende Daten werden nicht komprimiert.

Anmerkung:

1. Für das XML-Speicherobjekt einer Tabelle kann nur ein Komprimierungswörterverzeichnis erstellt werden, wenn die XML-Spalten der Tabelle in DB2 Version 9.7 oder einer höheren Version hinzugefügt werden bzw. wenn die Tabelle mithilfe einer Onlineoperation zum Versetzen von Tabellen migriert wird.
2. Wenn die Operation **LOAD REPLACE KEEPDICTIONARY** unterbrochen wird, kann das Dienstprogramm LOAD eine Recovery durchführen, nachdem entweder der Befehl **LOAD RESTART** oder der Befehl **LOAD TERMINATE** abgesetzt wird. Das vorhandene Wörterbuch für XML-Speicherobjekte kann möglicherweise nach einer Recovery nach einer unterbrochenen Operation **LOAD REPLACE KEEPDICTIONARY** nicht beibehalten werden. Bei Verwendung des Befehls **LOAD RESTART** wird ein neues Wörterbuch für XML-Speicherobjekte erstellt.

RESETDICTIONARY

Diese Anweisung bewirkt, dass bei der Verarbeitung von **LOAD REPLACE** ein neues Wörterbuch für das Tabellendatenobjekt erstellt wird, sofern das Tabellenattribut COMPRESS auf YES gesetzt ist. Wenn das Attribut COMPRESS auf NO gesetzt ist und die Tabelle bereits ein Wörterbuch enthält, wird das vorhandene Wörterbuch entfernt und kein neues Wörterbuch in die Tabelle eingefügt. Ein Komprimierungswörterbuch kann mithilfe eines einzigen Benutzerdatensatzes erstellt werden. Wenn die Größe der geladenen Datei null ist und bereits ein Wörterbuch vorhanden ist, wird das Wörterbuch nicht beibehalten. Die erforderliche Datenmenge zum Erstellen eines Wörterbuchs mit dieser Anweisung hängt nicht von den Richtlinien für die automatische Erstellung von Komprimierungswörterbüchern (ADC) ab. Eine Zusammenfassung finden Sie in der nachfolgenden Tabelle.

Tabelle 38. **LOAD REPLACE RESETDICTIONARY**

Komprimieren	Wörterverzeichnis der Tabellenzeilendaten ist vorhanden	Wörterverzeichnis der XML-Speicherobjekte ist vorhanden ¹	Komprimierungswörterverzeichnis	Datenkomprimierung
JA	JA	JA	Neue Wörterverzeichnisse erstellen. ² Wird die Option DATA CAPTURE CHANGES in den Anweisungen CREATE TABLE oder ALTER TABLE aktiviert, wird das aktuelle Wörterverzeichnis für Tabellenzeilendaten beibehalten (und als <i>Protokollwörterverzeichnis der Komprimierung</i> bezeichnet).	Nach dem Erstellen der Wörterverzeichnisse werden verbleibende Daten, die geladen werden sollen, komprimiert.

Tabelle 38. LOAD REPLACE RESETDICTIONARY (Forts.)

Komprimieren	Wörterverzeichnis der Tabellenzeilendaten ist vorhanden	Wörterverzeichnis der XML-Speicherobjekte ist vorhanden ¹	Komprimierungswörterverzeichnis	Datenkomprimierung
JA	JA	NEIN	Neue Wörterverzeichnis erstellen. ² Wird die Option DATA CAPTURE CHANGES in den Anweisungen CREATE TABLE oder ALTER TABLE aktiviert, wird das aktuelle Wörterverzeichnis für Tabellenzeilendaten beibehalten (und als <i>Protokollwörterverzeichnis der Komprimierung</i> bezeichnet).	Nach dem Erstellen der Wörterverzeichnisse werden verbleibende Daten, die geladen werden sollen, komprimiert.
JA	NEIN	JA	Neue Wörterverzeichnis erstellen.	Nach dem Erstellen der Wörterverzeichnisse werden verbleibende Daten, die geladen werden sollen, komprimiert.
JA	NEIN	NEIN	Neue Wörterverzeichnis erstellen.	Nach dem Erstellen der Wörterverzeichnisse werden verbleibende Daten, die geladen werden sollen, komprimiert.
NEIN	JA	JA	Wörterverzeichnisse löschen.	Zu ladende Daten werden nicht komprimiert.
NEIN	JA	NEIN	Wörterverzeichnis der Tabellenzeilendaten entfernen.	Zu ladende Daten werden nicht komprimiert.
NEIN	NEIN	JA	Wörterverzeichnis der XML-Speicherobjekte entfernen.	Zu ladende Daten werden nicht komprimiert.
NEIN	NEIN	NEIN	Keine Auswirkung.	Alle Tabellendaten werden nicht komprimiert.

Hinweise:

1. Für das XML-Speicherobjekt einer Tabelle kann nur ein Komprimierungswörterverzeichnis erstellt werden, wenn die XML-Spalten der Tabelle in DB2 Version 9.7 oder einer höheren Version hinzugefügt werden bzw. wenn die Tabelle mithilfe einer Onlineoperation zum Versetzen von Tabellen migriert wird.
2. Ist ein Wörterbuch vorhanden und das Komprimierungsattribut ist aktiviert, aber es sind keine Datensätze zum Laden in die Tabellenpartition vorhanden, so kann kein neues Wörterbuch erstellt werden, und die Operation **RESETDICTIONARY** behält das vorhandene Wörterbuch nicht bei.

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, sodass beim nächsten Zugriff automatisch eine Indexwiederherstellung erfolgt). Wenn es sich bei der zu beendenden Ladeoperation um **LOAD REPLACE** handelt, wird die Tabelle nach der Operation **LOAD TERMINATE** auf eine leere Tabelle reduziert. Handelt es sich bei der zu beendenden Ladeoperation um **LOAD INSERT**, bleiben nach der Operation **LOAD TERMINATE** alle ursprünglichen Datensätze der Tabelle erhalten. Eine Zusammenfassung der Wörterbuchverwaltung finden Sie in der nachfolgenden Tabelle.

Die Option **LOAD TERMINATE** hebt keinen Status für anstehendes Backup in Tabellenbereichen auf.

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. Eine Zusammenfassung der Wörterbuchverwaltung finden Sie in der nachfolgenden Tabelle.

INTO *tabellenname*

Gibt die Datenbanktabelle an, in die die Daten geladen werden sollen. Diese Tabelle darf keine Systemtabelle, keine deklarierte temporäre Tabelle und keine erstellte 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.

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:

Dieser Befehl schlägt aufgrund der Spalte Int 4 fehl. Die Lösung besteht darin, solche Spaltennamen in doppelte Anführungszeichen einzuschließen.

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. Wenn ein Tabellename 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 partitionierter Datenbank muss eine Ausnahmetabelle für die Datenbankpartitionen definiert werden, auf denen die Ladetabelle definiert ist. Andernfalls enthält die Speicherauszugsdatei Zeilen, die nicht geladen werden können, weil sie ungültig sind oder Syntaxfehler enthalten.

Wenn Sie XML-Daten laden, wird die Angabe der Ladeausnahmetabelle mit der Klausel **FOR EXCEPTION** in den folgenden Fällen nicht unterstützt:

- Bei Verwendung von kennsatzbasierter Zugriffssteuerung (LBAC)
- Beim Laden von Daten in eine partitionierte Tabelle

NORANGEEXC

Gibt an, dass eine wegen einer Bereichsverletzung zurückgewiesene Zeile nicht in die Ausnahmetabelle eingefügt wird.

NOUNIQUEEXC

Gibt an, dass eine Zeile, die zurückgewiesen wird, weil sie eine eindeutige Integritätsbedingung verletzt, nicht in die Ausnahmetabelle eingefügt wird.

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. Das Profil wird durch den Befehl **RUNSTATS** erstellt. 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.

Während des Ladens werden für Spalten des Typs XML keine Verteilungsstatistiken 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.

COPY NO

Gibt an, dass der Tabellenbereich, in dem sich die Tabelle befindet, in den Status 'Backup anstehend' versetzt wird, wenn die aktualisierende Recovery aktiviert ist (dass heißt, **logretain** oder **userexit** ist aktiv). Die Option **COPY NO** versetzt den Tabellenbereich außerdem in den Status 'Laden läuft'. Dies ist ein vorübergehender Status, der aufgehoben wird, sobald das Laden abgebrochen wird oder abgeschlossen ist. Die Daten in allen Tabellen des Tabellenbereichs können erst dann aktualisiert oder gelöscht werden, nachdem ein Backup des Tabellenbereichs oder ein Datenbankgesamtbackup durchgeführt wurde. Der Zugriff auf die Daten in allen Tabellen ist jedoch durch die SELECT-Anweisung möglich.

Der Befehl **LOAD** mit der Option **COPY NO** für eine wiederherstellbare Datenbank versetzt die Tabellenbereiche in den Status 'Backup anstehend'. Beispielsweise ist nach Ausführung des Befehls **LOAD** mit den Optionen **COPY NO** und **INDEXING MODE DEFERRED** eine Aktualisierung der Indizes erforderlich. 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 'Backup anstehend' hat. In diesem Fall ist der Zugriff auf die Tabelle nicht zulässig, bis ein Backup ausgeführt wurde. Die Indexaktualisierung wird automatisch durch die Datenbank ausgeführt, wenn eine Abfrage auf den Index zugreift. Wenn keine der Optionen **COPY NO**, **COPY YES** oder **NONRECOVERABLE** angegeben wird und die Datenbank wiederherstellbar ist (**logretain** oder **logarchmeth1** ist aktiviert), wird **COPY NO** als Standardwert verwendet.

COPY YES

Gibt an, dass eine Kopie der geladenen Daten gesichert wird. Diese Option ist ungültig, wenn die aktualisierende Recovery inaktiviert ist.

USE TSM

Gibt an, dass die Kopie mithilfe 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 Kopienimage erstellt wird.

LOAD *bib-name*

Der Name der gemeinsam genutzten Bibliothek (DLL unter Windows-Betriebssystemen), in der die zu verwendenden E/A-Funktionen für Backup und Restore des Herstellers enthalten sind. Der Name kann den vollständigen Pfad enthalten. Wenn der vollständige Pfad nicht angegeben wird, wird standardmäßig der Pfad verwendet, in dem sich die Benutzerexitprogramme befinden.

NONRECOVERABLE

Gibt an, dass eine Ladetransaktion als nicht wiederherstellbar gekennzeichnet werden soll und dass sie nicht durch eine nachfolgende aktualisierende Recovery wiedergestellt werden kann. Das Dienstprogramm für aktualisierende Recovery 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 aktualisierenden Recovery kann eine solche Tabelle nur nach einem Commitpunkt, der auf den Abschluss der nicht wiederherstellbaren Ladeoperation folgt, gelöscht oder aus einem Backup (Gesamtbackup oder Backup des Tabellenbereichs) wiederhergestellt werden.

Mit dieser Option werden Tabellenbereiche nach der Ladeoperation nicht in den Status 'Backup anstehend' versetzt, und während der Ladeoperation muss keine Kopie der geladenen Daten erstellt werden. Wenn keine der Optionen **COPY NO**, **COPY YES** oder **NONRECOVERABLE** angegeben wird und die Datenbank nicht wiederherstellbar ist (weder **logretain** noch **logarchmeth1** ist aktiviert), wird **NONRECOVERABLE** als Standardwert verwendet.

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'.

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 für Dienstprogramme zugeordnet, dessen Größe durch den Datenbankkonfigurationsparameter **util_heap_sz** geändert werden kann. Ab Version 9.5 kann der Wert für die Option **DATA BUFFER** des Befehls **LOAD** den Wert von **util_heap_sz** temporär überschreiten, falls im System mehr Speicher zur Verfügung steht. In dieser Situation wird der Zwischenspeicher für Dienstprogramme

entsprechend dem Bedarf dynamisch erhöht, bis der Grenzwert für **datenbankspeicher** erreicht ist. Dieser Speicher wird freigegeben, sobald die Ladeoperation abgeschlossen ist.

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 Instanzerstellung des Ladeprogramms im Zwischenspeicher der Dienstprogramme verfügbar ist, sowie auf einigen Merkmalen der Tabelle.

SORT BUFFER *puffergröße*

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 **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.

CPU_PARALLELISM *n*

Gibt die Anzahl von Prozessen bzw. Threads an, die vom Dienstprogramm LOAD zur Syntaxanalyse, Konvertierung und Formatierung von Datensätzen verwendet werden. Dieser Parameter soll die Nutzung der pro Datenbankpartition aktiven Prozesse 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.

Anmerkung:

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 der Wert für **CPU_PARALLELISM** größer als eins ist, erfolgen die Flushoperationen asynchron und das Ladeprogramm kann die CPU nutzen. Wenn **CPU_PARALLELISM** auf den Wert eins gesetzt ist, wartet das Ladeprogramm an Konsistenzpunkten auf E/A-Daten. Eine Ladeoperation mit dem Wert 2 für **CPU_PARALLELISM** und dem Wert 10 000 für **SAVECOUNT** wird schneller abgeschlossen als die gleiche Operation mit dem Wert 1 für **CPU_PARALLELISM**, obwohl nur eine CPU vorhanden ist.

DISK_PARALLELISM *n*

Gibt die Anzahl von Prozessen bzw. Threads an, die vom Dienstprogramm LOAD erstellt werden, um Daten in die Tabellenbereichscontainer zu schreiben. Wenn kein Wert angegeben wird, wählt das Dienstprogramm einen sinnvollen Standardwert auf Basis der Anzahl der Tabellenbereichscontainer und der Merkmale der Tabelle aus.

INDEXING MODE

Gibt an, ob das Dienstprogramm LOAD Indizes erneut erstellen oder sie inkrementell erweitern soll. Gültige Werte sind:

AUTOSELECT

Das Dienstprogramm LOAD wählt automatisch den Modus REBUILD oder INCREMENTAL aus. Die Entscheidung richtet sich nach dem Umfang der zu ladenden Daten und der Tiefe der Indexstruktur. Informationen zur Tiefe der Indexstruktur sind im Indexobjekt gespeichert. **RUNSTATS** ist nicht erforderlich, um diese Informationen zu füllen. AUTOSELECT ist der Standardmodus für die Indexierung.

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 unterstützt, wenn das Indexobjekt am Anfang einer Ladeoperation gültig und erreichbar ist (beispielsweise ist es unmittelbar nach einer Ladeoperation, in der der Modus DEFERRED angegeben war, nicht gültig). Wenn dieser Modus angegeben ist, aber aufgrund des Indexstatus nicht unterstützt wird, wird eine Warnung zurückgegeben und die Ladeoperation wird im Modus REBUILD fortgesetzt. Außerdem wird der Modus INCREMENTAL nicht unterstützt, wenn eine LOAD RESTART-Operation in der BUILD-Phase der Ladeoperation eingeleitet wird.

DEFERRED

Das Dienstprogramm LOAD erstellt keine Indizes, wenn dieser Modus angegeben wird. Die Indizes werden als aktualisierungsbedürftig gekennzeichnet. Bei dem ersten Zugriff auf solche Indizes, der nicht im Rahmen einer Ladeoperation erfolgt, wird möglicherweise eine erneute Erstellung erzwungen, oder die Indizes werden erneut erstellt, wenn die Datenbank erneut gestartet wird. Dieser Ansatz erfordert genug Sortierspeicherplatz für alle Schlüsselteile für den größten Index. Die im Anschluss daran für die Indexerstellung benötigte Gesamtzeit ist länger als die im Modus REBUILD benötigte Zeit. 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.

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 zulässige Option für **LOAD REPLACE**.

Wenn für die Tabelle Integritätsbedingungen gelten, wird der Tabellenstatus auf 'Festlegen der Integrität anstehend' sowie 'Laden läuft' gesetzt. Die

Anweisung SET INTEGRITY muss verwendet werden, um den Status 'Festlegen der Integrität anstehend' 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 der Ladeoperation verfügbar. **LOAD TERMINATE** oder **LOAD RESTART** in einer Ladeoperation mit **ALLOW READ ACCESS** kann diese Option verwenden. **LOAD TERMINATE** oder **LOAD RESTART** in einer Ladeoperation mit **ALLOW NO ACCESS** kann diese Option nicht verwenden. 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 'Festlegen der Integrität anstehend' sowie 'Laden läuft' und 'Lesezugriff' gesetzt. Am Ende der Ladeoperation wird der Tabellenstatus 'Laden läuft' aufgehoben. Die Statuswerte 'Festlegen der Integrität anstehend' und 'Lesezugriff' für die Tabelle bleiben jedoch gültig. Die Anweisung SET INTEGRITY muss verwendet werden, um den Status 'Festlegen der Integrität anstehend' für die Tabelle aufzuheben. Während sich die Tabelle im Status 'Festlegen der Integrität anstehend' und 'Lesezugriff' 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 Ladeoperationen 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*

Wenn die Indizes erneut erstellt werden, wird eine Spiegelkopie des Indexes im Tabellenbereich *tabellenbereichsname* erstellt und am Ende der Ladeoperation während einer 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 während 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 vorkommen, 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. Sie wird auch ignoriert, wenn **INDEXING MODE AUTOSELECT** ausgewählt ist und das Dienstprogramm LOAD eine inkrementelle Aktualisierung des Indexes einleitet.

FETCH_PARALLELISM YES | NO

Wird eine Ladeoperation über einen Cursor ausgeführt, der mit dem Schlüsselwort **DATABASE** deklariert wurde, oder über den API-Datenträgeintrag `sqlu_remotefetch_entry`, der auf YES gesetzt ist, versucht das Dienstprogramm LOAD (falls möglich), Daten parallel aus der fernen Datenquelle abzurufen. Ist die Option auf NO gesetzt, erfolgt kein paralleles Abrufen. Der Standardwert ist YES. Weitere Informationen hierzu finden Sie unter „Versetzen von Daten mithilfe des Dateityps CURSOR“.

SET INTEGRITY PENDING CASCADE

Wenn **LOAD** die Tabelle in den Status 'Festlegen der Integrität anstehend' versetzt, kann der Benutzer über die Option **SET INTEGRITY PENDING CASCADE** angeben, ob dieser Status der geladenen Tabelle direkt an alle untergeordneten Elemente (einschließlich der untergeordneten Fremdschlüsseltabellen, IMQs und sofort gespeicherten Zwischenspeichertabellen) weitergegeben werden soll.

IMMEDIATE

Gibt an, dass der Status 'Festlegen der Integrität anstehend' unmittelbar auf alle untergeordneten Fremdschlüsseltabellen, untergeordneten IMQs und untergeordneten sofort gespeicherten Zwischenspeichertabellen ausgeweitet wird. Bei einer **LOAD INSERT**-Operation wird der Status 'Festlegen der Integrität anstehend' nicht an untergeordnete Fremdschlüsseltabellen weitergegeben, selbst wenn die Option **IMMEDIATE** angegeben wird.

Bei der späteren Prüfung der geladenen Tabelle auf ungültige Integritätsbedingungen (mit der Option **IMMEDIATE CHECKED** der Anweisung **SET INTEGRITY**), werden untergeordnete Fremdschlüsseltabellen, die den Status 'Festlegen der Integrität anstehend, Lesezugriff' aufweisen, in den Status 'Festlegen der Integrität anstehend, Kein Zugriff' versetzt.

DEFERRED

Gibt an, dass nur die geladene Tabelle in den Status 'Festlegen der Integrität anstehend' versetzt wird. Die Statuswerte von untergeordneten Fremdschlüsseltabellen, untergeordneten IMQs und untergeordneten sofort gespeicherten Zwischenspeichertabellen werden nicht geändert.

Untergeordnete Fremdschlüsseltabellen werden später möglicherweise implizit in den Status 'Festlegen der Integrität anstehend' versetzt, wenn die dazugehörigen übergeordneten Tabellen auf ungültige Integritätsbedingungen geprüft werden (mit der Option **IMMEDIATE CHECKED** der Anweisung **SET INTEGRITY**). Untergeordnete IMQs und untergeordnete sofort gespeicherte Zwischenspeichertabellen werden implizit in den Status 'Festlegen der Integrität anstehend' versetzt, sobald eine der ihnen zu Grunde liegenden Tabellen auf ungültige Integritätsbedingungen überprüft

wird. Eine Abfrage einer Tabelle im Status 'Festlegen der Integrität anstehend' kann erfolgreich sein, wenn die Abfrage nicht auf die angegebene Tabelle zugreift, sondern auf eine infrage kommende MQT, die sich nicht im Status 'Festlegen der Integrität anstehend' befindet. Eine Warnung (SQLSTATE 01586) wird ausgegeben, um anzugeben, dass untergeordnete Tabellen in den Status 'Festlegen der Integrität anstehend' versetzt wurden. Angaben dazu, wann diese untergeordneten Tabellen in den Status 'Festlegen der Integrität anstehend' versetzt werden, finden Sie im Abschnitt mit den Hinweisen zur Anweisung SET INTEGRITY im Handbuch 'SQL Reference'.

Wenn die Option **SET INTEGRITY PENDING CASCADE** nicht angegeben ist, gilt Folgendes:

- Nur die geladene Tabelle wird in den Status 'Festlegen der Integrität anstehend' versetzt. Die Statuswerte von untergeordneten Fremdschlüssel-tabellen, untergeordneten IMQTs und untergeordneten sofort gespeicherten Zwischenspeichertabellen bleiben unverändert. Diese Tabellen können später implizit in den Status 'Festlegen der Integrität anstehend' versetzt werden, sobald die geladene Tabelle auf ungültige Integritätsbedingungen hin überprüft wird.

Wenn **LOAD** die Zieltabelle nicht in den Status 'Festlegen der Integrität anstehend' versetzt, wird die Option **SET INTEGRITY PENDING CASCADE** ignoriert.

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 benötigt wie für den Befehl **FORCE APPLICATIONS** (SYSADM oder SYSCTRL).

Ladeoperationen mit ALLOW NO ACCESS können die Beendigung von Anwendungen mit Konflikte verursachenden Sperren am Anfang der Ladeoperation erzwingen. Am Anfang der Ladeoperation kann das Dienstprogramm die Beendigung von Anwendungen erzwingen, die versuchen, die Tabelle entweder abzufragen oder zu ändern.

Ladeoperationen mit **ALLOW READ ACCESS** können die Beendigung von Anwendungen mit Konflikte verursachenden Sperren am Anfang oder Ende der Ladeoperation erzwingen. Am Anfang der Ladeoperation kann das Dienstprogramm LOAD die Beendigung von Anwendungen erzwingen, die versuchen, die Tabelle zu ändern. Am Ende der Ladeoperation kann das Dienstprogramm LOAD die Beendigung von Anwendungen erzwingen, die versuchen, die Tabelle abzufragen oder zu ändern.

SOURCEUSEREXIT *ausführbare-datei*

Gibt den Namen einer ausführbaren Datei an, die aufgerufen werden soll, um dem Dienstprogramm Daten zuzuführen.

REDIRECT

INPUT FROM

BUFFER *eingabepuffer*

Der Bytestrom, der in *eingabepuffer* angegeben ist, wird in den Dateideskriptor STDIN des Prozesses übergeben, in dem die gegebene ausführbare Datei ausgeführt wird.

FILE *eingabedatei*

Der Inhalt dieser clientseitigen Datei wird in den Dateideskriptor STDIN des Prozesses übergeben, in dem die gegebene ausführbare Datei ausgeführt wird.

OUTPUT TO

FILE *ausgabedatei*

Die Dateideskriptoren STDOUT und STDERR werden in der angegebenen vollständig qualifizierten serverseitigen Datei erfasst.

PARALLELIZE

Erhöht den Durchsatz der im Dienstprogramm LOAD eingehenden Daten, indem mehrere Benutzerexitprozesse gleichzeitig aufgerufen werden. Diese Option gilt nur für Umgebungen mit partitionierter Datenbank und wird in Umgebungen mit Einzelpartitionsdatenbank ignoriert.

Weitere Informationen hierzu finden Sie unter „Versetzen von Daten mithilfe einer angepassten Anwendung (Benutzerexit)“.

PARTITIONED DB CONFIG *option-für-partitionierte-db*

Dieser Parameter ermöglicht das Laden von Daten in eine Tabelle, die auf mehrere Datenbankpartitionen verteilt ist. Der Parameter **PARTITIONED DB CONFIG** ermöglicht das Angeben von spezifischen Konfigurationsoptionen für partitionierte Datenbanken. Die *option-für-partitionierte-db* kann folgende Werte annehmen:

```
PART_FILE_LOCATION x
OUTPUT_DBPARTNUMS x
PARTITIONING_DBPARTNUMS x
MODE x
MAX_NUM_PART_AGENTS x
ISOLATE_PART_ERRS x
STATUS_INTERVAL x
PORT_RANGE x
CHECK_TRUNCATION
MAP_FILE_INPUT x
MAP_FILE_OUTPUT x
TRACE x
NEWLINE
DISTFILE x
OMIT_HEADER
RUN_STAT_DBPARTNUM x
```

Ausführliche Beschreibungen dieser Optionen finden Sie unter „LOAD-Konfigurationsoptionen für Umgebungen mit partitionierten Datenbanken“.

RESTARTCOUNT

Veraltet.

USING *verzeichnis*
Veraltet.

Beispiele für das Laden von Daten aus XML-Dokumenten

XML-Daten laden

Beispiel 1

Vom Benutzer wurde eine Datendatei mit XDS-Feldern zum Beschreiben der Dokumente erstellt, die in die Tabelle eingefügt werden sollen. Die Datei sieht wie folgt aus:

```
1, "<XDS FIL=""datei1.xml"" />"  
2, "<XDS FIL='datei2.xml' OFF='23' LEN='45' />"
```

In der ersten Zeile wird das XML-Dokument durch die Datei namens `datei1.xml` identifiziert. Da das Anführungszeichen als Zeichenbegrenzer verwendet wird und die XDS Anführungszeichen enthält, werden die Anführungszeichen in der XDS gedoppelt. In der zweiten Zeile wird das XML-Dokument durch die Datei namens `datei2.xml` identifiziert. Es beginnt an der relativen Byteadresse 23 und ist 45 Byte lang.

Beispiel 2

Der Benutzer setzt einen Befehl **LOAD** für die XML-Spalte ohne Optionen für Syntaxanalyse oder Prüfung ab, und die Daten werden erfolgreich geladen:

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

Laden von XML-Daten vom CURSOR

Das Laden von Daten vom Cursor entspricht dem Laden von einem regulären relationalen Spaltentyp. Der Benutzer verfügt über zwei Tabellen (T1 und T2) die jeweils aus einer einzigen XML-Spalte namens C1 bestehen. Zum Laden von T1 in T2 muss der Benutzer zunächst einen Cursor deklarieren:

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

Im nächsten Schritt kann der Benutzer einen Befehl **LOAD** absetzen, der den Cursortyp verwendet:

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

Das Anwenden der XML-spezifischen **LOAD**-Optionen auf den Cursortyp entspricht dem Laden aus einer Datei.

Hinweise 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. Wenn die Reihenfolge der Quelldaten nicht beibehalten werden muss, können Sie auch die Verwendung des Dateitypänderungswerts **ANYORDER** in Betracht ziehen, der im nachfolgenden Abschnitt „Änderungswerte für Dateityp für das Dienstprogramm LOAD“ beschrieben wird.
- Das Dienstprogramm **LOAD** erstellt Indizes gemäß den vorhandenen Definitionen. Mithilfe von Ausnahmetabellen werden doppelt vorhandene Werte für eindeutige Schlüssel verarbeitet. Das Dienstprogramm nimmt für MQT (Materia-

lized Query Tables), 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 'Festlegen der Integrität anstehend' versetzt. Übersichtstabellen, für die das Attribut REFRESH IMMEDIATE definiert ist und die von geladenen Tabellen abhängig sind, werden ebenfalls in den Status 'Festlegen der Integrität anstehend' versetzt. Durch Absetzen der Anweisung SET INTEGRITY können Sie den Status 'Festlegen der Integrität anstehend' für die Tabellen aufheben. Die Ausführung von Ladeoperationen für replizierte MQT 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.
- Wenn Sie beim Laden von Daten in eine geschützte Tabelle eine Ausnahmetabelle angeben, werden alle Zeilen, die durch ungültige Sicherheitskennsätze geschützt sind, in diese Tabelle übertragen. Dies ermöglicht es Benutzern, die Zugriff auf die Ausnahmetabelle haben, auf Daten zuzugreifen, für die ihnen eigentlich die Zugriffsberechtigung fehlt. Aus Sicherheitsgründen empfiehlt es sich, die Benutzer, die mit Grant Zugriff auf die Ausnahmetabelle erhalten, sorgfältig auszuwählen, die einzelnen Zeilen, sobald sie repariert und in die zu ladende Tabelle kopiert sind, sofort zu löschen und die Ausnahmetabelle selbst sofort zu löschen, sobald sie nicht mehr benötigt wird.
- Sicherheitskennsätze können in ihrem internen Format Zeilenvorschubzeichen enthalten. Wenn Sie die Datei mit dem Dateiformat DEL laden, können diese Zeilenvorschubzeichen als Begrenzer interpretiert werden. Wenn dieses Problem vorliegt, geben Sie den Änderungswert **delprioritychar** für den Dateityp im Befehl **LOAD** an, um die vorherige Standardpriorität für Begrenzer zu verwenden.
- Zum Ausführen einer Ladeoperation für den Dateityp CURSOR, bei der das Schlüsselwort DATABASE in der Anweisung DECLARE CURSOR angegeben wurde, wird die Benutzer-ID mit Kennwort zur Authentifizierung in der Datenbank, zu der momentan eine Verbindung (für die Ladeoperation) besteht, für die Authentifizierung in der Quelldatenbank verwendet, die in der Option DATABASE der Anweisung DECLARE CURSOR angegeben ist. Wurde keine Benutzer-ID und/oder kein Kennwort für die Verbindung zur ladenden Datenbank angegeben, müssen bei der Anweisung DECLARE CURSOR eine Benutzer-ID und ein Kennwort für die Quelldatenbank angegeben werden.
- Das Laden einer mehrteiligen PC/IXF-Datei, deren Bestandteile von einem Windows-System auf ein AIX-System kopiert werden, wird unterstützt. Die Namen aller Dateien müssen im Befehl **LOAD** angegeben werden. Beispiel: **LOAD FROM DATA.IXF, DATA.002 OF IXF INSERT INTO TABLE1**. Das Laden von logisch geteilten PC/IXF-Dateien in das Windows-Betriebssystem wird nicht unterstützt.
- Beim erneuten Starten einer fehlgeschlagenen **LOAD**-Operation wird das bestehende Verhalten insofern beibehalten, als in der BUILD-Phase die Verwendung des REBUILD-Modus für Indizes erzwungen wird.
- Das Laden von XML-Dokumenten von einer Datenbank auf andere Datenbanken wird nicht unterstützt und führt zur Ausgabe der Fehlermeldungen SQL1407N.
- Das Dienstprogramm **LOAD** bietet keine Unterstützung für das Laden in Tabellen, in denen Spalten enthalten sind, die auf abgeschirmte Prozeduren verweisen. Wenn Sie für eine solche Tabelle den Befehl **LOAD** absetzen, erhalten Sie die Fehlermeldung SQL1376N. Sie können diese Einschränkung umgehen, indem Sie die Routine neu definieren, sodass sie nicht abgeschirmt ist, oder das Dienstprogramm **IMPORT** verwenden.
- Der Befehl **STATISTICS YES** ist nur eingeschränkt anwendbar und wird möglicherweise in einem zukünftigen Release entfernt.

- Die Optionen STATISTICS lassen sich nur mit der Option **LOAD REPLACE** und keinen anderen Befehlsoptionen des Typs **LOAD** anwenden.

Zusammenfassung der **LOAD TERMINATE**- und **LOAD RESTART**-Wörterbuchverwaltung

Das folgende Diagramm enthält eine Zusammenfassung der Verwaltung von Komprimierungswörterbüchern für die **LOAD**-Verarbeitung im Rahmen der Anweisung **TERMINATE**.

Tabelle 39. **LOAD TERMINATE**-Wörterbuchverwaltung

Tabellenattribut COMPRESS	Wörterverzeichnis der Tabellen- zeilendaten ist vor der LOAD - Operation vor- handen	Wörterverzeichnis der XML- Speicherobjekte ist vor der LOAD -Opera- tion vorhanden ¹	TERMINATE: LOAD REPLACE KEEPDICTIONARY oder LOAD INSERT	TERMINATE: LOAD REPLACE RESETDICTIONARY
JA	JA	JA	Vorhandene Wörterver- zeichnisse beibehalten.	Kein Wörterverzeichnis wird beibehalten. ²
JA	JA	NEIN	Vorhandenes Wörterbuch beibehalten.	Es wird nichts beibehal- ten. ²
JA	NEIN	JA	Vorhandenes Wörterbuch beibehalten.	Es wird nichts beibehal- ten.
JA	NEIN	NEIN	Es wird nichts beibehalten.	Es wird nichts beibehal- ten.
NEIN	JA	JA	Vorhandene Wörterver- zeichnisse beibehalten.	Es wird nichts beibehal- ten.
NEIN	JA	NEIN	Vorhandenes Wörterbuch beibehalten.	Es wird nichts beibehal- ten.
NEIN	NEIN	JA	Vorhandenes Wörterbuch beibehalten.	Es wird nichts beibehal- ten.
NEIN	NEIN	NEIN	Nichts ausführen.	Nichts ausführen.

Anmerkung:

1. Für das XML-Speicherobjekt einer Tabelle kann nur ein Komprimierungswörterverzeichnis erstellt werden, wenn die XML-Spalten der Tabelle in DB2 Version 9.7 oder einer höheren Version hinzugefügt werden bzw. wenn die Tabelle mithilfe einer Onlineoperation zum Versetzen von Tabellen migriert wird.
2. In den Sonderfällen, in denen die Datenerfassung für die Tabelle aktiviert ist, wird das Wörterverzeichnis für die Tabellenzeilendaten beibehalten.

LOAD RESTART verkürzt eine Tabelle bis zum letzten erreichten Konsistenzpunkt. Im Rahmen der **LOAD RESTART**-Verarbeitung existiert ein Komprimierungswörterbuch in der Tabelle, wenn es bereits beim Erreichen des letzten **LOAD**-Konsistenzpunkts in der Tabelle vorhanden war. In diesem Fall erstellt **LOAD RESTART** kein neues Wörterbuch. Eine Zusammenfassung der möglichen Bedingungen finden Sie in der nachfolgenden Tabelle.

Tabelle 40. LOAD RESTART-Wörterbuchverwaltung

Tabellenattribut COMPRESS	Wörterverzeichnis der Tabellen- zeilendaten ist vor dem LOAD- Synchronisations- punkt vorhanden ¹	Wörterverzeichnis der XML- Speicherobjekte ist vor der letzten LOAD- Operation vorhan- den ²	RESTART: LOAD REPLACE KEEPDICTIONARY oder LOAD INSERT	RESTART: LOAD REPLACE RESETDICTIONARY
JA	JA	JA	Vorhandene Wörterver- zeichnisse beibehalten.	Vorhandene Wörterver- zeichnisse beibehalten.
JA	JA	NEIN	Wörterverzeichnis der Tabellenzeilendaten beibe- halten und XML-Wörterver- zeichnis auf der Basis von ADC erstellen.	Wörterverzeichnis der Tabellenzeilendaten beibe- halten und XML-Wörter- verzeichnis erstellen.
JA	NEIN	JA	Wörterverzeichnis der Tabellenzeilendaten auf der Basis von ADC erstellen. Vorhandenes XML-Wörter- verzeichnis beibehalten.	Wörterverzeichnis der Tabellenzeilendaten er- stellen. Vorhandenes XML-Wörterverzeichnis beibehalten.
JA	NEIN	NEIN	Wörterverzeichnis der Tabellenzeilendaten und XML-Wörterverzeichnis auf der Basis von ADC erstel- len.	Wörterverzeichnis der Tabellenzeilendaten und XML-Wörterverzeichnis erstellen.
NEIN	JA	JA	Vorhandene Wörterver- zeichnisse beibehalten.	Vorhandene Wörterverzeichnisse löschen.
NEIN	JA	NEIN	Vorhandenes Wörterver- zeichnis der Tabellenzeilendaten beibe- halten.	Vorhandenes Wörterver- zeichnis der Tabellenzeilendaten ent- fernen.
NEIN	NEIN	JA	Vorhandenes XML-Wörter- verzeichnis beibehalten.	Vorhandenes XML-Wörter- verzeichnis entfernen.
NEIN	NEIN	NEIN	Nichts ausführen.	Nichts ausführen.

Hinweise:

1. Die Option **SAVECOUNT** ist für das Laden von XML-Daten nicht zulässig. Ladeoperationen, bei denen in der LOAD-Phase ein Fehler auftritt, werden vom Beginn der Operation an erneut gestartet.
2. Für das XML-Speicherobjekt einer Tabelle kann nur ein Komprimierungswörterverzeichnis erstellt werden, wenn die XML-Spalten der Tabelle in DB2 Version 9.7 oder einer höheren Version hinzugefügt werden bzw. wenn die Tabelle mithilfe einer Onlineoperation zum Versetzen von Tabellen migriert wird.

Änderungswerte für Dateityp für das Dienstprogramm LOAD

Tabelle 41. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: Alle Dateiformate

Änderungswert	Beschreibung
anyorder	Dieser Änderungswert wird zusammen mit dem Parameter cpu_parallelism verwendet. Er gibt an, dass die Beibehaltung der Quelldatenreihenfolge nicht erforderlich ist, was zu erheblichen Leistungssteigerungen auf SMP-Systemen führt. Wenn der Wert von cpu_parallelism 1 ist, wird diese Option ignoriert. Diese Option wird nicht unterstützt, wenn SAVECOUNT > 0, da die Recovery 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 generatedmissing oder dem Änderungswert generatedoverride 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 generatedignore oder dem Änderungswert generatedoverride 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 RECOVER DROPPED TABLE des Befehls ROLLFORWARD DATABASE 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). Bei Verwendung dieses Änderungswerts wird die Tabelle in den Status 'Festlegen der Integrität anstehend' versetzt. Um den Status 'Festlegen der Integrität 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 <i>tabellenname</i> GENERATED COLUMN IMMEDIATE UNCHECKED</pre> <p>Um den Status 'Festlegen der Integrität 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 <i>tabellenname</i> IMMEDIATE CHECKED.</pre> <p>Wird dieser Änderungswert angegeben und gibt es in einem der partitionierten Schlüssel, Dimensionsschlüssel oder Verteilungsschlüssel eine generierte Spalte, konvertiert der Befehl LOAD den Änderungswert automatisch in generatedignore und fährt mit dem Laden fort. Dies bewirkt, dass alle generierten Spaltenwerte erneut generiert werden.</p> <p>Dieser Änderungswert kann nicht zusammen mit dem Änderungswert generatedmissing oder dem Änderungswert generatedignore verwendet werden.</p>

Tabelle 41. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
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 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 oder dem Änderungswert identityoverride 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 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. 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	<p>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. Beim Laden mit INDEXING MODE INCREMENTAL wird diese Option ignoriert. Der erste Eintrag auf einer Seite wird ohne Einschränkung hinzugefügt; nachfolgende Einträge werden mit der Maßgabe hinzugefügt, dass der Schwellenwert für den Prozentsatz des freizuhaltenden Speicherbereichs eingehalten wird. Der Standardwert ist der zum Zeitpunkt der Indexerstellung (CREATE INDEX) verwendete Wert.</p> <p>Dieser Wert hat Vorrang vor dem in der Anweisung CREATE INDEX angegebenen Wert für PCTFREE. Die Option indexfreespace wirkt sich nur auf Indexblattseiten aus.</p>

Tabelle 41. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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. Die Klausel LOBS FROM impliziert die Aktivierung des Verhaltens von lobsinfile. Die Klausel LOBS FROM überträgt die Liste der Pfade an das Dienstprogramm LOAD, die beim Laden der Daten nach den 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. Wenn 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 <i>db2exp.001</i> 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>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.
pagefreespace=x	<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. wenn für eine Zeile, die mindestens 3.000 Byte lang ist, für <i>x</i> der Wert 50 angegeben ist), wird die Zeile auf einer neuen Seite platziert. Wenn der Wert 100 angegeben wird, wird jede Zeile auf eine neue Seite gestellt. 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>

Tabelle 41. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
rowchangetimestampignore	Dieser Änderungswert informiert das Dienstprogramm LOAD darüber, dass für die Zeitmarkenspalte für Zeilenänderung in der Datendatei Daten vorhanden sind, aber ignoriert werden sollen. Dies bewirkt, dass alle Wert für ROW CHANGE TIMESTAMP durch das Dienstprogramm generiert werden. Dieses Verhalten ist für als GENERATED ALWAYS definierte Spalten und als GENERATED BY DEFAULT definierte Spalten gleich. Dies bedeutet, dass bei als GENERATED ALWAYS definierten Spalten keine Zeilen zurückgewiesen werden. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert rowchangetimestampmissing oder rowchangetimestampoverride verwendet werden.
rowchangetimestampmissing	Wird dieser Änderungswert angegeben, geht das Dienstprogramm davon aus, dass die Eingabedatendatei keine Daten (nicht einmal Nullwerte) für die Zeitmarkenspalte für Zeilenänderung enthält, und generiert daher für jede Zeile einen Wert. Dieses Verhalten ist für als GENERATED ALWAYS definierte Spalten und als GENERATED BY DEFAULT definierte Spalten gleich. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert rowchangetimestampignore oder rowchangetimestampoverride verwendet werden.
rowchangetimestampoverride	Dieser Änderungswert sollte nur dann verwendet werden, wenn die zu ladende Tabelle eine als GENERATED ALWAYS definierte Zeitmarkenspalte für Zeilenänderung enthält. Er weist das Dienstprogramm an, entgegen der normalen Regeln für diese Typen von Zeitmarkenspalten für Zeilenänderung, 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 Spalte ROW CHANGE TIMESTAMP zurückgewiesen (SQL3116W). Dieser Änderungswert kann nicht zusammen mit dem Änderungswert rowchangetimestampmissing oder rowchangetimestampignore verwendet werden. Wird diese Option verwendet, versucht das Dienstprogramm LOAD nicht, die Eindeutigkeit der Werte in der Zeitmarkenspalte für Zeilenänderung zu verwalten oder zu prüfen.
seclabelchar	<p>Gibt an, dass Sicherheitskennsätze in der Eingabequellendatei im Zeichenfolgeformat für Sicherheitskennsatzwerte vorliegen (nicht im codierten numerischen Standardformat). Von LOAD wird jeder einzelne Sicherheitskennsatz beim Laden in das interne Format konvertiert. Liegt eine Zeichenfolge nicht im richtigen Format vor, wird die betreffende Zeile nicht geladen und eine Warnung (SQLSTATE 01H53, SQLCODE SQL3242W) zurückgegeben. Stellt eine Zeichenfolge nicht einen gültigen Sicherheitskennsatz dar, der Bestandteil der zum Schutz der Tabelle verwendeten Sicherheitsrichtlinie ist, wird die betreffende Zeile nicht geladen und eine Warnung (SQLSTATE 01H53, SQLCODE SQL3243W) zurückgegeben.</p> <p>Dieser Änderungswert kann nicht angegeben werden, wenn der Änderungswert seclabelname angegeben ist. Wird diese Kombination von Änderungswerten angegeben, schlägt das Laden fehl, und es wird ein Fehler (SQLCODE SQL3525N) zurückgegeben.</p> <p>Wenn Sie über eine Tabelle verfügen, die eine einzige Spalte DB2SECURITYLABEL enthält, sieht die Datendatei möglicherweise wie folgt aus:</p> <pre>"CONFIDENTIAL:ALPHA:G2" "CONFIDENTIAL;SIGMA:G2" "TOP SECRET:ALPHA:G2"</pre> <p>Zum Laden und Importieren dieser Daten muss der Änderungswert für den Dateityp seclabelchar verwendet werden:</p> <pre>LOAD FROM input.del OF DEL MODIFIED BY SECLABELCHAR INSERT INTO t1</pre>

Tabelle 41. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
seclabelname	<p>Gibt an, dass Sicherheitskennsätze in der Eingabequellendatei über ihren Namen angegeben werden und nicht im codierten numerischen Standardformat vorliegen. Von LOAD wird der jeweilige Name in den entsprechenden Sicherheitskennsatz konvertiert, soweit vorhanden. Existiert kein Sicherheitskennsatz mit dem angegebenen Namen für die zum Schutz der Tabelle verwendete Sicherheitsrichtlinie, wird die betreffende Zeile nicht geladen und eine Warnung (SQLSTATE 01H53, SQLCODE SQL3244W) zurückgegeben.</p> <p>Dieser Änderungswert kann nicht angegeben werden, wenn der Änderungswert seclabelchar angegeben ist. Wird diese Kombination von Änderungswerten angegeben, schlägt das Laden fehl, und es wird ein Fehler (SQLCODE SQL3525N) zurückgegeben.</p> <p>Wenn Sie über eine Tabelle verfügen, die eine einzige Spalte DB2SECURITYLABEL enthält, besteht die Datendatei möglicherweise aus Sicherheitskennsatznamen wie den folgenden:</p> <pre>"LABEL1" "LABEL1" "LABEL2"</pre> <p>Zum Laden und Importieren dieser Daten muss der Änderungswert für den Dateityp seclabelname verwendet werden:</p> <pre>LOAD FROM input.del OF DEL MODIFIED BY SECLABELNAME INSERT INTO t1</pre> <p>Anmerkung: Ist der Dateityp ASC, werden alle auf den Namen des Sicherheitskennsatzes folgenden Leerzeichen als Teil des Namens interpretiert. Verwenden Sie den Änderungswert für Dateityp 'striptblanks', um sicherzustellen, dass die Leerzeichen entfernt werden.</p>
total freespace=x	<p>x 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 x 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. Wenn zwei Ladeoperationen mit dieser Option ausgeführt werden, wird der zusätzliche Speicherbereich, der am Ende durch die erste Ladeoperation angehängt wurde, von der zweiten Ladeoperation nicht erneut verwendet.</p>

Tabelle 41. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: Alle Dateiformate (Forts.)

Änderungswert	Beschreibung
usedefaults	<p>Wenn eine Quellenspalte für eine Zieltabellenspalte angegeben wurde, jedoch für mindestens eine Zeileninstanz keine Daten enthält, werden Standardwerte geladen. Beispiele für fehlende Daten sind:</p> <ul style="list-style-type: none"> • Bei DEL-Dateien: Für einen Spaltenwert werden zwei benachbarte Spaltenbegrenzer (",,") oder zwei durch eine beliebige Anzahl Leerzeichen (" , ") getrennte, benachbarte Spaltenbegrenzer angegeben. • Bei DEL/ASC/WSF-Dateien: Eine Zeile, die für die ursprüngliche Angabe zu wenig Spalten enthält oder nicht lang genug ist. Bei Dateien des Typs ASC werden NULL-Spaltenwerte nicht als explizit fehlend betrachtet, und es wird kein Standardwert für NULL-Spaltenwerte eingesetzt. NULL-Spaltenwerte können durch alle Leerzeichen für numerische Spalten, Datenspalten, Zeitspalten und Zeitmarkenspalten oder durch die Verwendung des Nullindicators für Spalten eines beliebigen Typs zur Angabe eines Spaltenwerts von NULL dargestellt werden. <p>Ohne diese Option geschieht Folgendes, wenn eine Quellenspalte keine Daten für eine Zeileninstanz enthält:</p> <ul style="list-style-type: none"> • Für DEL/ASC/WSF-Dateien: Wenn die Spalte Nullwerte enthalten darf, wird NULL geladen. Wenn die Spalte keine Nullwerte enthalten darf, wird die Zeile vom Dienstprogramm zurückgewiesen.

Tabelle 42. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASCII-Dateiformate (ASC/DEL)

Änderungswert	Beschreibung
codepage=<i>x</i>	<p><i>x</i> 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>Es gelten folgende Regeln:</p> <ul style="list-style-type: none"> • Bei reinem DBCS (Grafik), gemischtem DBCS und EUC sind Begrenzer auf den Bereich von x00 bis x3F (jeweils einschließlich) beschränkt. • Bei DEL-Daten, die in einer EBCDIC-Codepage angegeben werden, sind die Begrenzer nicht unbedingt mit den DBCS-Startzeichen und -Endezeichen identisch. • 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. <p>Die Verwendung dieser Option in Verbindung mit dem Dateityp CURSOR wird nicht unterstützt.</p>

Tabelle 42. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
dateformat ="x"	<p>x ist das Format des Datums in der Quelldatei.¹ 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 01 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 01 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: "D-M-YYYY" "MM.DD.YYYY" "YYYYDDD"</p>
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 Instanzeigner. Die Standarddateiberechtigungen können mit dem Änderungswert für Dateityp dumpfileaccessall überschrieben werden.</p> <p>Anmerkung:</p> <ol style="list-style-type: none"> 1. In einer Umgebung mit partitionierter Datenbank muss der Pfad für die ladende Datenbankpartition lokal sein, damit gleichzeitig aktive Ladeoperationen nicht versuchen, in dieselbe Datei zu schreiben. 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. 3. Wenn die angegebene Datei bereits vorhanden ist, wird sie nicht erneut erstellt, sondern abgeschnitten.
dumpfileaccessall	<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"> 1. wenn er zusammen mit dem Änderungswert für Dateityp dumpfile verwendet wird, 2. der Benutzer über das Zugriffsrecht SELECT für die Ladezieltabelle verfügt, 3. wenn er in einer DB2-Serverdatenbankpartition abgesetzt wird, die sich auf einem UNIX-Betriebssystem befindet. <p>Wenn die angegebene Datei bereits vorhanden ist, werden die für sie bestehenden Berechtigungen nicht geändert.</p>

Tabelle 42. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
fastparse	Diese Werte sollten mit Bedacht verwendet werden. Reduziert die Syntaxprüfung für Spaltenwerte, die vom Benutzer angegeben wurden, und verbessert die Leistung. Die korrekte Architektur der Tabellen wird sichergestellt (die vom Dienstprogramm vorgenommene Datenprüfung reicht aus, um Segmentierungsverstöße und Traps zu verhindern), aber die Datenkohärenz wird nicht geprüft. Verwenden Sie diese Option nur, wenn Sie sicher sind, dass Ihre Daten kohärent und korrekt sind. Enthalten die vom Benutzer angegebenen Daten beispielsweise den ungültigen Wert :1>0-00-20-07.11.12.000000 für die Zeitmarkenspalte, wird dieser Wert in die Tabelle eingefügt, wenn fastparse angegeben ist, und zurückgewiesen, wenn fastparse nicht angegeben ist.
implieddecimal	Die Position eines implizierten Dezimalzeichens wird durch die Spaltendefinition bestimmt. Sie wird nicht mehr am Ende des Werts angenommen. Beispielsweise wird der Wert 12345 in eine DECIMAL(8,2)-Spalte als 123,45 geladen und <i>nicht</i> als 12345,00. Dieser Änderungswert kann nicht zusammen mit dem Änderungswert packeddecimal verwendet werden.
timeformat="x"	x ist das Format der Zeitangabe in der Quelldatei. ¹ Gültige Zeitelemente sind: H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 - 24 bei einem 24-Stunden-System) HH - Stunde (zwei Ziffern von 00 bis 12 bei einem 12-Stunden-System und 00 - 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus) M - Minute (eine oder zwei Ziffern von 0 bis 59) MM - Minute (zwei Ziffern von 00 bis 59; dieses Element und das Element M schließen sich gegenseitig aus) S - Sekunde (eine oder zwei Ziffern von 0 bis 59) SS - Sekunde (zwei Ziffern von 00 bis 59; dieses Element und das Element S schließen sich gegenseitig aus) SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86400; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus) TT - Meridiananzeiger (AM oder PM) Für jedes Element, das nicht angegeben wird, wird der Standardwert 0 zugeordnet. Einige Beispiele für Zeitformate: "HH:MM:SS" "HH.MM TT" "SSSSS"

Tabelle 42. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
timestampformat="x"	<p>x ist das Format der Zeitmarke in der Quellendatei.¹ 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 01 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 00 bis 12 bei einem 12-Stunden-System und 00 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 00 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 00 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 - 86400; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus)</p> <p>U (1 bis 12 mal)</p> <p>- Bruchsekunden (die Häufigkeit von U gibt die Anzahl der Stellen mit Werten aus dem Bereich 0 bis 9 an)</p> <p>TT - Meridiananzeiger (AM oder PM)</p>

Tabelle 42. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
<p>timestampformat="x" (Fortsetzung)</p>	<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>Wenn der Änderungswert timestampformat nicht angegeben wird, verwendet das Dienstprogramm LOAD für das Zeitmarkenfeld eines von zwei möglichen Formaten:</p> <pre> YYYY-MM-DD-HH.MM.SS YYYY-MM-DD HH:MM:SS </pre> <p>Das Dienstprogramm LOAD wählt das Format entsprechend dem Trennzeichen zwischen DD und HH. Ist das Trennzeichen ein Gedankenstrich '-', wird vom Dienstprogramm LOAD das Format mit Gedankenstrichen und Punkten verwendet (YYYY-MM-DD-HH.MM.SS). Ist das Trennzeichen ein Leerzeichen, erwartet das Dienstprogramm LOAD einen Doppelpunkt ':' zwischen den Angaben HH, MM und SS.</p> <p>In beiden Formaten erwartet das Dienstprogramm LOAD den Punkt '.' als Trennzeichen, wenn Sie das Feld für Mikrosekunden (UUUUUU) einschließen. YYYY-MM-DD-HH.MM.SS.UUUUUU oder YYYY-MM-DD HH:MM:SS.UUUUUU ist zulässig.</p> <p>Das folgende Beispiel veranschaulicht, wie Daten mit benutzerdefinierten Datums- und Zeitformaten in eine Tabelle namens SCHEDULE geladen werden:</p> <pre> db2 load from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule </pre>
<p>usegraphiccodepage</p>	<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 Recovery nach einer DROP TABLE-Operation generiert wird, wenn die wiederherzustellende Tabelle Grafikdaten enthält.</p> <p>Einschränkungen</p> <p>Der Änderungswert usegraphiccodepage DARF NICHT für DEL-Dateien angegeben werden, die mit dem Dienstprogramm EXPORT erstellt wurden, da die in diesen Dateien enthaltenen Daten nur in einer Codepage codiert sind. Der Änderungswert usegraphiccodepage wird auch von den großen Doppelbytezeichenobjekten (DBCLOBs) in Dateien ignoriert.</p>

Tabelle 42. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASCII-Dateiformate (ASC/DEL) (Forts.)

Änderungswert	Beschreibung
xmlchar	<p>Gibt an, dass XML-Dokumente in der Zeichencodepage codiert sind.</p> <p>Diese Option eignet sich für die Verarbeitung von XML-Dokumenten, die in der angegebenen Zeichencodepage codiert sind, ohne eine Codierdeklaration zu enthalten.</p> <p>Bei jedem Dokument muss die Codierung, soweit eine Deklarationskennung mit Codierungsattribut existiert, der Zeichencodepage entsprechen. Ist dies nicht der Fall, wird die Zeile mit dem Dokument zurückgewiesen. Dabei ist zu beachten, dass die Zeichencodepage dem Wert entspricht, der für den Änderungswert für Dateityp codepage angegeben ist, oder der Codepage der Anwendung, wenn dieser Wert nicht angegeben ist. Die Dokumente sind standardmäßig entweder in Unicode kodiert oder enthalten eine Deklarationskennung mit Codierungsattribut.</p>
xmlgraphic	<p>Gibt an, dass XML-Dokumente in der angegebenen Grafikcodepage codiert sind.</p> <p>Diese Option eignet sich für die Verarbeitung von XML-Dokumenten, die in einer bestimmten Grafikcodepage codiert sind, ohne eine Codierdeklaration zu enthalten.</p> <p>Bei jedem Dokument muss die Codierung, soweit eine Deklarationskennung mit Codierungsattribut existiert, der Grafikcodepage entsprechen. Ist dies nicht der Fall, wird die Zeile mit dem Dokument zurückgewiesen. Dabei ist zu beachten, dass die Grafikcodepage der Grafikkomponente des Werts entspricht, der über den Änderungswert für Dateityp codepage angegeben wird, oder der Grafikkomponente der Anwendungscodepage, wenn dieser Wert nicht angegeben ist. Dokumente sind standardmäßig entweder in Unicode kodiert oder enthalten eine Deklarationskennung mit Codierungsattribut.</p>

Tabelle 43. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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.</p> <p>Es gelten folgende Regeln:</p> <ul style="list-style-type: none"> • Es wird keine Konvertierung zwischen Datentypen durchgeführt, mit Ausnahme von BIGINT, INTEGER und SMALLINT. • Die Datenlänge muss mit der jeweiligen Zielspaltendefinition übereinstimmen. • FLOAT-Werte müssen im IEEE-Gleitkommaformat vorliegen. • Binärdaten in der Ladequellendatei werden als Big-Endian gewertet, unabhängig von der Plattform, auf der die Ladeoperation ausgeführt wird. <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>

Tabelle 43. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASC-Dateiformate (ASCII-Format mit universellen Zeilenbegrenzern) (Forts.)

Änderungswert	Beschreibung
nochecklengths	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.
nullindchar=x	<p>x ist ein einzelnes Zeichen. Ändert das Zeichen, das einen Nullwert bezeichnet, in x. Der Standardwert für x ist \backslash.</p> <p>Bei diesem Änderungswert muss für EBCDIC-Datendateien die Groß-/Kleinschreibung beachtet werden, außer wenn das Zeichen ein englischer Buchstabe ist. Wenn beispielsweise für das NULL-Anzeigerzeichen der Buchstabe N angegeben wird, wird auch n als NULL-Anzeiger erkannt.</p>
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.</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>Binärdaten in der Ladequellendatei werden als Big-Endian gewertet, unabhängig von der Serverplattform, auf der die Ladeoperation ausgeführt wird. Bei Verwendung dieses Änderungswerts unter Windows-Betriebssystemen darf die Byteanordnung jedoch nicht umgekehrt 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. Diese Option ersetzt die veraltete Option t, die nur noch aus Gründen der Abwärtskompatibilität 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. Diese Option ersetzt die veraltete Option padwithzero, die nur noch aus Gründen der Abwärtskompatibilität unterstützt wird.</p>

Tabelle 43. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: ASC-Dateiformate (ASCII-Format mit universellen Zeilenbegrenzern) (Forts.)

Änderungswert	Beschreibung
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.</p> <p>Die folgenden Halbbytevorzeichenwerte sind möglich: + = 0xC 0xA 0xE 0xF 0x3 - = 0xD 0xB 0x7</p> <p>Für Ziffern werden Werte von 0x0 bis 0x9 unterstützt. Für Zonen werden die Werte 0x3 und 0xF unterstützt.</p>

Tabelle 44. Gültige Änderungswerte für Dateityp für das Dienstprogramm LOAD: DEL-Dateiformate (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 anstelle von doppelten Anführungszeichen verwendet, um eine Zeichenfolge einzuschließen.²³ Falls Sie die doppelten Anführungszeichen explizit als Zeichenfolgebegrenzer angeben wollen, müssen Sie dies folgendermaßen tun: modified by chardel""</p> <p>Auch das einfache Anführungszeichen (') kann wie folgt als Zeichenfolgebegrenzer festgelegt werden: modified by chardel''</p>
coldelx	<p><i>x</i> ist ein einzelnes Zeichen, das als Spaltenbegrenzer verwendet wird. Der Standardwert ist ein Komma (.). Das angegebene Zeichen wird anstelle eines Kommas verwendet, um das Ende einer Spalte anzugeben.²³</p>
decplusblank	<p>Zeichen für Pluszeichen. Bewirkt, dass positiven Dezimalwerten statt eines Pluszeichens (+) ein Leerzeichen vorangestellt wird. Die Standardaktion besteht darin, positiven Dezimalwerten ein Pluszeichen voranzustellen.</p>
decptx	<p><i>x</i> ist ein einzelnes Ersatzzeichen für den Punkt als Dezimalzeichen. Der Standardwert ist ein Punkt (.). Das angegebene Zeichen wird anstelle eines Punkts als Dezimalzeichen verwendet.²³</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: db2 load ... modified by delprioritychar ...</p> <p>Beispiel anhand der folgenden DEL-Datendatei: "Smith, Joshua",4000,34.98<zeilenbegrenzer> "Vincent,<zeilenbegrenzer>, is a manager", 4005,44.37<zeilenbegrenzer></p> <p>Wenn der Änderungswert delprioritychar angegeben wird, gibt es in dieser Datendatei nur zwei Zeilen. Der zweite <zeilenbegrenzer> wird als Teil der ersten Datenspalte der zweiten Zeile gewertet, während der erste und dritte <zeilenbegrenzer> als eigentliche Satzbegrenzer gewertet werden. Wenn dieser Änderungswert <i>nicht</i> angegeben wird, gibt es in dieser Datendatei drei Zeilen, jeweils getrennt durch einen <zeilenbegrenzer>.</p>

Tabelle 44. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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>
nocharde1	<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 mithilfe eines DB2-Datenbanksystems exportiert wurden (es sei denn, nocharde1 war beim Exportieren angegeben). 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 nodoublede1 angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p>
nodoublede1	Unterdrückt die Erkennung von doppelten Zeichenbegrenzern.

Tabelle 45. Gültige Änderungswerte für Dateityp für das Dienstprogramm 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>

Anmerkung:

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.

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 \).

2. Für die Dateitypänderungswerte **chardel**, **coldel** oder **decpt** angegebene Zeichenwerte müssen 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 ...
```

3. „Aspekte beim Versetzen von XML-Daten - Begrenzer“ listet Einschränkungen für Zeichen auf, die anstelle der standardmäßigen Begrenzer verwendet werden können.
4. Das Dienstprogramm LOAD gibt keine Warnung aus, 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.
5. Beim Import in eine Tabelle, die eine implizit ausgeblendete Zeitmarkenspalte für Zeilenänderung enthält, wird das Merkmal **IMPLICITLY HIDDEN** der Spalte nicht berücksichtigt. Daher *muss* der Dateitypänderungswert **rowchangetimestampmissing** im Befehl **IMPORT** angegeben werden, wenn die zu importierenden Daten keine Daten für die Spalte enthalten und keine explizite Spaltenliste vorhanden ist.

Tabelle 46. 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 wird.

Tabelle 46. Verhalten des Dienstprogramms LOAD bei Verwendung von 'codepage' und 'usegraphiccodepage' (Forts.)

codepage=N	usegraphiccodepage	Verhalten von LOAD
Vorhanden	Nicht vorhanden	Es wird angenommen, dass sich alle Daten in der Datei in der Codepage N befinden. Warnung: Grafikdaten werden beim Laden in die Datenbank beschädigt, wenn N eine Einzelbyte-Codepage ist.
Nicht vorhanden	Vorhanden	Es wird angenommen, dass Zeichendaten in der Datenbankcodepage vorliegen, selbst wenn die Option CLIENT angegeben wird. Es wird angenommen, dass Grafikdaten in der Codepage der Datenbankgrafikdaten vorliegen, selbst wenn die Option CLIENT angegeben wird. Wenn die Datenbankcodepage eine Einzelbyte-Codepage ist, wird angenommen, dass sich alle Daten in der Datenbankcodepage befinden. Warnung: 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. Warnung: Grafikdaten werden beim Laden in die Datenbank beschädigt, wenn N eine Einzelbyte-Codepage ist.

db2Load - Laden von Daten in eine Tabelle

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 ist zwar schneller als das Dienstprogramm IMPORT, unterstützt jedoch nicht das Laden von Daten auf Hierarchieebene bzw. nicht das Laden in einen Kurznamen.

Berechtigung

Eine der folgenden Berechtigungen:

- *DATAACCESS*
- 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.

Bei Angabe der Option FORCE ist die Berechtigung SYSADM erforderlich.

Anmerkung: In der Regel sind alle Ladeprozesse und alle DB2-Serverprozesse Eigentum des Instanzeigners. Alle diese Prozesse verwenden die Identifikation des Instanzeigners, um auf die benötigten Dateien zuzugreifen. Unabhängig davon, wer den Befehl aufruft, muss daher der Instanzeigner einen Lesezugriff auf die Eingabedateien haben.

Erforderliche Verbindung

Datenbank. Wenn das implizite Herstellen von Verbindungen aktiviert ist, wird eine Verbindung zur Standarddatenbank hergestellt. Beim Zugriff des Dienstprogramms auf Linux-, UNIX- oder Windows-Datenbankserver von Linux-, UNIX- oder Windows-Clients aus muss es sich um eine Direktverbindung über die Steuerkomponente handeln. Eine Verbindung über ein DB2 Connect-Gateway oder eine Loopback-Umgebung ist nicht zulässig.

Instanz. Eine explizite Zuordnung ist nicht erforderlich. Wenn eine Verbindung zur Datenbank hergestellt wurde, wird versucht, eine implizite Zuordnung zur lokalen Instanz vorzunehmen.

API-Include-Datei

db2ApiDf.h

API- und Datenstruktursyntax

```
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;
    struct sqlu_media_list *piXmlPathList;
    struct sqllob *piLongActionString;
} db2LoadStruct;

typedef SQL_STRUCTURE db2LoadUserExit
{
    db2Char iSourceUserExitCmd;
```

```

        struct db2Char *piInputStream;
        struct db2Char *piInputFileName;
        struct db2Char *piOutputFileName;
        db2UInt16 *piEnableParallelism;
    } db2LoadUserExit;

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;
    db2UInt16 *piXmlParse;
    db2DMUxmlValidate *piXmlValidate;
    db2UInt16 iSetIntegrityPending;
    struct db2LoadUserExit *piSourceUserExit;
} 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;

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;
    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;
    struct sqlu_media_list *piXmlPathList;
    struct sqllob *piLongActionString;
} 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          iUseTablesapceLen;
        db2UInt16 iSetIntegrityPending;
        db2UInt16 *piXmlParse;
        db2DMUxmlValidate *piXmlValidate;
        struct db2LoadUserExit *piSourceUserExit;
    } db2gLoadIn;

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;
    void *piReserved1;
    db2UInt16          iHostnameLen;
    db2UInt16          iFileTransferLen;
    db2UInt16          iPartFileLocLen;
    db2UInt16          iMapFileInputLen;
    db2UInt16          iMapFileOutputLen;
    db2UInt16          iDistfileLen;
} db2gPartLoadIn;

```

```

/* Definitionen für den Wert iUsing der Struktur db2DMUxmlValidate */
#define DB2DMU_XMLVAL_XDS 1 /* XDS verwenden */
#define DB2DMU_XMLVAL_SCHEMA 2 /* Angegebenes Schema verwenden */
#define DB2DMU_XMLVAL_SCHEMALOC_HINTS 3 /* schemaLocation-Hinweise
verwenden */
#define DB2DMU_XMLVAL_ORIGSCHEMA 4 /* Schema verwenden, an dem das
Dokument ursprünglich geprüft
wurde (nur aus Cursor laden) */

```

Parameter der API 'db2Load'

versionNumber

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

pParmStruct

Eingabe. Ein Zeiger auf die 'db2LoadStruct'-Struktur.

pSqlca

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

Parameter der db2LoadStruct-Datenstruktur

piSourceList

Eingabe. Ein Zeiger auf eine 'sqlu_media_list'-Struktur, die verwendet wird, um eine Liste mit Quelldateien, Einheiten, Lieferanten, Pipes oder SQL-Anweisungen bereitzustellen.

Die in dieser Struktur bereitgestellten Daten hängen vom Wert des Feldes 'media_type' ab. Gültige Werte (gemäß Definition in der Kopfdatendatei 'sqlutil' im INCLUDE-Verzeichnis) 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 genutzten 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.

SQLU_REMOTEFETCH

Wenn das Feld 'media_type' auf diesen Wert gesetzt ist, stellt das

aufrufende Programm über eine Struktur 'sqlu_remotefetch_entry' Informationen bereit. Das Feld 'sessions' muss auf den Wert 1 gesetzt sein.

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 (gemäß Definition in der Kopfdatendatei 'sqlutil' im INCLUDE-Verzeichnis) 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 genutzten 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.

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 'piFileType' auf SQL_ASC gesetzt ist, muss das Feld 'dcolmeth' dieser Struktur auf SQL_METH_L gesetzt werden. Der Benutzer gibt die Anfangs- und Endpositionen für jede zu ladende Spalte an.

Lautet der Dateityp 'SQL_DEL', kann 'dcolmeth' entweder 'SQL_METH_P' oder 'SQL_METH_D' sein. Lautet der Wert 'SQL_METH_P', muss der Benutzer die Position der Quellenspalte angeben. Bei 'SQL_METH_D' wird die erste Spalte in der Datei in die erste Spalte der Tabelle geladen usw.

Lautet der Dateityp 'SQL_IXF', 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

Veraltet. Ersetzt durch piLongActionString.

piLongActionString

Eingabe. Zeiger auf eine *sqllob*-Struktur, die ein 4-Byte-Langfeld enthält, gefolgt von einer Matrix mit Zeichen, die eine Aktion mit Auswirkung auf die Tabelle angibt.

Die Zeichenmatrix besitzt folgendes Format:

```
"INSERT|REPLACE KEEPDICTIONARY|REPLACE RESETDICTIONARY|RESTART|TERMINATE  
INTO tname [(spaltenliste)]  
[FOR EXCEPTION ausnahmetbname]"
```

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, sodass 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.

Die Option LOAD TERMINATE hebt keinen Status für anstehendes Backup in Tabellenbereichen auf.

tname

Der Name der Tabelle, in die die Daten geladen werden sollen. Die Tabelle darf keine Systemtabelle, keine deklarierte temporäre Tabelle und keine erstellte 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.tabellename'. 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.

FOR EXCEPTION ausnahmetbname

Gibt die Ausnahmetabelle an, in die fehlerhafte Zeilen kopiert werden. Die Ausnahmetabelle wird verwendet, um Kopien der Zeilen zu speichern, die die Regeln eines eindeutigen Indexes, Bereichsvorgaben und Sicherheitsrichtlinien verletzen.

NORANGEEXC

Gibt an, dass eine wegen einer Bereichsverletzung zurückgewiesene Zeile nicht in die Ausnahmetabelle eingefügt wird.

NOUNIQUEEXC

Gibt an, dass eine Zeile, die zurückgewiesen wird, weil sie eine eindeutige Integritätsbedingung verletzt, nicht in die Ausnahmetabelle eingefügt wird.

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 sowie mit vielen anderen Datenbankmanagern und Dateimanagern.

SQL_IXF

PC-Version des IXF-Formats (Integration Exchange Format). Dies ist das bevorzugte Format für den Datenexport aus einer Tabelle. In diesem Dateiformat exportierte Daten können später in dieselbe Tabelle oder in eine andere Datenbankmanagertabelle geladen werden.

SQL_CURSOR

Eine SQL-Abfrage. Die durch den Parameter 'piSourceList' übergebene Struktur 'sqlu_media_list' hat entweder den Typ SQLU_SQL_STMT oder SQLU_REMOTEFETCH und verweist auf eine SQL-Abfrage oder einen Tabellennamen.

piFileTypeMod

Eingabe. Ein Zeiger auf die 'sqlchar'-Struktur, gefolgt von einer Matrix mit Zeichen, die eine oder mehrere Verarbeitungsoption(en) 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. Vgl. zugehörigen Link: "Änderungswerte für Dateityp für das Dienstprogramm LOAD"

piLocalMsgFileName

Eingabe. Eine Zeichenfolge, die den Namen einer lokalen Datei enthält, in die Ausgabenachrichten geschrieben werden sollen.

piTempFilePath

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 Ko-

pierimage erstellt werden soll) verwendet wird, um eine Liste mit Zielpfaden, Einheiten oder eine gemeinsam genutzte Bibliothek bereitzustellen, in die das Kopierimage geschrieben werden soll.

Die in dieser Struktur bereitgestellten Werte hängen vom Wert des Feldes 'media_type' ab. Gültige Werte für diesen Parameter (gemäß Definition in der Kopfdatei 'sqlutil' im INCLUDE-Verzeichnis) 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 anderen 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 genutzten Bibliothek des Produkts des betreffenden Lieferanten. Es darf nur ein 'sqlu_vendor'-Eintrag angegeben werden, 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 unterschiedlichen Folgenummern, jedoch mit denselben Daten in dem einen 'sqlu_vendor'-Eintrag.

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. Daher muss die Anzahl der Elemente mit dem Feld 'dcolnum' des Parameters 'piDataDescriptor' 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 Position in der Datendatei entweder 'Y' oder 'N' enthalten. 'Y' gibt an, dass die Tabellenspaltdaten NULL sind, 'N' gibt an, dass die Tabellenspaltdaten nicht NULL sind.

piLoadInfoIn

Eingabe. Ein Zeiger auf die 'db2LoadIn'-Struktur.

poLoadInfoOut

Ausgabe. 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 (gemäß Definition in der Kopfdatendatei 'sqlutil' im INCLUDE-Verzeichnis) sind:

SQLU_INITIAL

Erstauf Ruf. Dieser Wert (oder SQLU_NOINTERRUPT) muss beim ersten Aufruf der API verwendet werden.

SQLU_NOINTERRUPT

Erstauf Ruf. 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 Bandendebingung 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. Bewirkt die vorzeitige Beendigung des Dienstprogramms LOAD, wobei die zu ladenden Tabellenbereiche im Status LOAD_PENDING (Laden anstehend) bleiben. Diese Option sollte angegeben werden, wenn keine weitere Verarbeitung der Daten erfolgen soll.

SQLU_ABORT

Verarbeitung beenden. Bewirkt die vorzeitige Beendigung des Dienstprogramms LOAD, wobei die zu ladenden Tabellenbereiche im Status LOAD_PENDING (Laden anstehend) bleiben. 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.

piXmlPathList

Eingabe. Zeiger auf eine Struktur 'sqlu_media_list', wobei das Feld 'media_type' auf SQLU_LOCAL_MEDIA gesetzt ist und die Struktur 'sqlu_media_entry' Pfade auf dem Client angibt, in denen sich die XML-Dateien befinden.

Parameter der db2LoadUserExit-Datenstruktur

iSourceUserExitCmd

Eingabe. Der vollständig qualifizierte Name einer ausführbaren Funktion,

die verwendet wird, um dem Dienstprogramm Daten zuzuführen. Aus Sicherheitsgründen muss die ausführbare Funktion in das Verzeichnis `sql/lib/bin` auf dem Server gestellt werden. Dieser Parameter ist obligatorisch, wenn die `'piSourceUserExit'`-Struktur ungleich NULL ist.

Die Felder `'piInputStream'`, `'piInputFileName'`, `'piOutputFileName'` und `'piEnableParallelism'` sind optional.

piInputStream

Eingabe. Ein generischer Bytestrom, der über STDIN direkt an die Benutzerexitanwendung übergeben wird. Sie haben die vollständige Kontrolle darüber, welche Daten in welchem Format in diesem Bytestrom enthalten sein sollen. Das Dienstprogramm LOAD überträgt diesen Bytestrom lediglich an den Server und übergibt ihn an die Benutzerexitanwendung, indem der Standardeingabe STDIN des Prozesses Daten zugeführt werden (der Bytestrom wird weder geändert, noch wird für ihn eine Codepagekonvertierung durchgeführt). Ihre Benutzerexitanwendung würde die Argumente aus STDIN lesen und die Daten wie beabsichtigt verwenden.

Ein wichtiges Merkmal dieser Funktion ist die Möglichkeit, sensible Informationen (wie beispielsweise Benutzer-IDs und Kennwörter) zu verdecken.

piInputFileName

Eingabe. Enthält den Namen einer vollständig qualifizierten clientseitigen Datei, deren Inhalt durch Zuführung von Daten an die Standardeingabe STDIN des Prozesses an die Benutzerexitanwendung übergeben wird.

piOutputFileName

Eingabe. Der vollständig qualifizierte Name einer serverseitigen Datei. Die STDOUT- und STDERR-Datenströme des Prozesses, der die Benutzerexitanwendung ausführt, werden in diese Datei geleitet. Wenn `'piEnableParallelism'` auf TRUE gesetzt ist, werden mehrere Dateien erstellt (eine für jede `'user-exit'`-Instanz). An jeden Dateinamen wird eine dreistellige numerische Knotennummer wie `<dateiname>.000` angehängt.

piEnableParallelism

Eingabe. Eine Markierung, die angibt, dass das Dienstprogramm versuchen soll, den Aufruf der Benutzerexitanwendung zu parallelisieren.

Parameter der db2LoadIn-Datenstruktur

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. Wenn die Indizes erneut erstellt werden, wird eine Spiegelkopie des Indexes im Tabellenbereich `'iUseTablespaceName'` erstellt und am Ende der Ladeoperation 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 auch ignoriert, wenn `INDEXING MODE AUTOSELECT` ausgewählt ist und das Dienstprogramm `LOAD` die inkrementelle Aktualisierung des Indexes einleitet.

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 Konsistenzpunkt 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 für 'savecount' nicht groß genug ist, wirkt sich die Synchronisation der Aktivitäten, die an jedem Konsistenzpunkt ausgeführt wird, auf die Leistung aus.

Der Standardwert ist 0, d. h., dass Konsistenzpunkte 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 des Dienstprogramms 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 Instanzerstellung 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 zu laden versucht, sodass die Ladeoperation fehlschlägt. Wenn 'warningcnt' 0 ist oder diese Option nicht angegeben ist, wird die Ladeoperation unabhängig von der Anzahl der ausgegebenen 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 Quiescemo-
dus: EXCLUSIVE" belassen soll, bzw. auf FALSE, wenn dies nicht erwünscht ist.

iCpuParallelism

Eingabe. Gibt die Anzahl der Prozesse oder Threads an, die vom Dienstprogramm LOAD zur Syntaxanalyse, Konvertierung und Formatierung von Datensätzen verwendet werden. 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. Gibt die Anzahl der Prozesse oder Threads an, die vom Dienstprogramm LOAD erstellt werden, um Daten in die Tabellenbereichscontainer zu schreiben. Wenn kein Wert angegeben wird, wählt das Dienstprogramm einen sinnvollen Standardwert auf Basis der Anzahl der Tabellenbereichscontainer 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 Recovery wiederherzustellen. Das Dienstprogramm zur aktualisierenden Recovery 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 Fertigstellung der aktualisierenden Recovery kann eine solche Tabelle nur gelöscht werden. Mit dieser Option werden Tabellenbereiche nach der Ladeoperation nicht in den Status 'Backup 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 (gemäß Definition in der Kopfdatendatei 'sqlutil' im INCLUDE-Verzeichnis) 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 sind:

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 benötigt wie für den Befehl FORCE APPLICATIONS (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 modifizieren.

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 modifizieren. Am Ende des Ladevorgangs kann das Dienstprogramm LOAD die Beendigung von Anwendungen erzwingen, die versuchen, die Tabelle entweder abzufragen oder zu modifizieren.

iCheckPending

Dieser Parameter wird ab Version 9.1 nicht weiter unterstützt. Verwenden Sie stattdessen den Parameter 'iSetIntegrityPending'.

iRestartphase

Eingabe. Reserviert. Gültiger Wert ist ein einzelnes Leerzeichen (' ').

iStatsOpt

Eingabe. Granularität der zu erfassenden Statistikdaten. Gültige Werte sind:

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 mithilfe des Befehls RUNSTATS erstellt werden. Ist für die aktuelle Tabelle kein Profil vorhanden, wird eine Warnung zurückgegeben und es werden keine Statistiken erfasst.

Während des Ladens werden für Spalten des Typs XML keine Verteilungsstatistiken erfasst.

iSetIntegrityPending

Eingabe. Gibt an, dass die Tabelle in den Status 'Festlegen der Integrität anstehend' versetzt werden soll. Falls der Wert SQLU_SI_PENDING_CASCADE_IMMEDIATE angegeben ist, wird der Status 'Festlegen der Integrität anstehend' sofort an alle abhängigen und untergeordneten Tabellen weitergegeben. Ist der Wert SQLU_SI_PENDING_CASCADE_DEFERRED angegeben, wird die Weitergabe des Statuswerts 'Festlegen der Integrität anstehend' an abhängige Tabellen so lange verzögert, bis die Zieltabelle auf Integritätsverletzungen hin überprüft wurde. SQLU_SI_PENDING_CASCADE_DEFERRED ist der Standardwert, wenn die Option nicht angegeben wird.

piSourceUserExit

Eingabe. Ein Zeiger auf die Struktur 'db2LoadUserExit'.

piXmlParse

Eingabe. Typ der Syntaxanalyse, die für XML-Dokumente ausgeführt werden soll. Gültige Werte gemäß der Kopfdatendatei 'db2ApiDf' im INCLUDE-Verzeichnis sind:

DB2DMU_XMLPARSE_PRESERVE_WS

Leerzeichen sollen erhalten bleiben.

DB2DMU_XMLPARSE_STRIP_WS

Leerzeichen sollen entfernt werden.

piXmlValidate

Eingabe. Ein Zeiger auf die 'db2DMUXmlValidate'-Struktur. Gibt an, dass für XML-Dokumente eine XML-Schemaprüfung durchgeführt werden soll.

```
/* XML-Prüfstruktur */
typedef SQL_STRUCTURE db2DMUXmlValidate
{
    db2Uint16          iUsing;      /* Angaben zum Ausführen */
                                /* der Prüfung */
    struct db2DMUXmlValidateXds *piXdsArgs; /* Argumente für */
                                /* XMLVALIDATE USING XDS */
    struct db2DMUXmlValidateSchema *piSchemaArgs; /* Argumente für */
                                /* XMLVALIDATE USING SCHEMA */
} db2DMUXmlValidate;
```

Parameter der db2LoadOut-Datenstruktur

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.

Parameter der db2PartLoadIn-Datenstruktur**piHostname**

Eingabe. Der Hostname für den Parameter 'iFileTransferCmd'. Bei Angabe von NULL nimmt der Hostname standardmäßig den Wert "nohost" an. Dieser Parameter ist veraltet.

piFileTransferCmd

Eingabe. Parameter für den Dateiübertragungsbefehl. Falls er nicht erforderlich ist, muss er auf NULL gesetzt sein. Dieser Parameter ist veraltet. Verwenden Sie stattdessen den Parameter 'piSourceUserExit'.

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 Datenbankpartitionen 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 Ausgabedatenbankpartition erstellt werden (beim Modus PARTITION_ONLY), bzw. die Position der Dateien, die aus jeder Datenbankpartition 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 Ausgabedatenbankpartitionen 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.

piMode

Eingabe. Gibt den Lademodus für partitionierte Datenbanken an. Gültige Werte (gemäß Definition in der Kopfdatendatei 'db2ApiDf' im INCLUDE-Verzeichnis) sind:

- DB2LOAD_PARTITION_AND_LOAD

Die Daten werden verteilt (möglicherweise in Parallelverarbeitung) und gleichzeitig auf die entsprechenden Datenbankpartitionen geladen.

- DB2LOAD_PARTITION_ONLY

Die Daten werden verteilt (möglicherweise in Parallelverarbeitung),

und die Ausgabe wird in Dateien an einer angegebenen Speicherposition auf jeder Ladedatenbankpartition geschrieben. Für andere Dateitypen als SQL_CURSOR hat der Name der Ausgabedatei in jeder Datenbankpartition das Format 'dateiname.xxx'. Hierbei ist 'dateiname' der Name der ersten Eingabedatei, die von 'piSourceList' angegeben wird, und 'xxx' ist die Datenbankpartitionsnummer. Für den Dateityp SQL_CURSOR wird der Name der Ausgabedatei in jeder Datenbankpartition vom Parameter 'piPartFileLocation' bestimmt. Weitere Informationen zum Angeben der Speicherposition der Datenbankpartitionsdatei in jeder Datenbankpartition finden Sie im Abschnitt zum Parameter 'piPartFileLocation'.

Anmerkung: Dieser Modus kann für CLI LOAD nicht verwendet werden.

DB2LOAD_LOAD_ONLY

Es wird davon ausgegangen, dass die Daten bereits verteilt sind. Der Verteilungsprozess wird übersprungen, und die Daten werden gleichzeitig auf die entsprechenden Datenbankpartitionen geladen. Für andere Dateitypen als SQL_CURSOR wird der Eingabedateiname in jeder Datenbankpartition im Format 'dateiname.xxx' erwartet. Hierbei ist 'dateiname' der Name der ersten Datei, die von 'piSourceList' angegeben wird, und 'xxx' ist die 3-stellige Datenbankpartitionsnummer. Für den Dateitypen SQL_CURSOR wird der Name der Eingabedatei in jeder Datenbankpartition vom Parameter 'piPartFileLocation' bestimmt. Weitere Informationen zum Angeben der Speicherposition der Datenbankpartitionsdatei in jeder Datenbankpartition finden Sie im Abschnitt zum 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 verteilt sind, aber die Datendatei enthält keine Datenbankpartitionskopfdaten. Der Verteilungsprozess 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 Datenbankpartition befindet. Zeilen, die Datenbankpartitionsverstöße enthalten, werden in eine Speicherausgabedatei gestellt, sofern der Dateitypänderungswert 'dumpfile' angegeben ist. Andernfalls werden die Zeilen gelöscht. Wenn für eine bestimmte Ladedatenbankpartition Datenbankpartitionsverstöße vorliegen, wird für die betreffende Datenbankpartition eine einzige Warnung in die LOAD-Nachrichtendatei geschrieben. Der Eingabedateiname in jeder Datenbankpartition wird im Format 'dateiname.xxx' erwartet. Hierbei ist 'dateiname' der Name der ersten Datei, die von 'piSourceList' angegeben wird, und 'xxx' ist die 3-stellige Datenbankpartitionsnummer.

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_ANALYZE

Es wird eine optimale Verteilungszuordnung 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 Datenbankpartitionen auftreten. Gültige Werte (gemäß Definition in der Kopfdatendatei 'db2ApiDf' im INCLUDE-Verzeichnis) sind:

DB2LOAD_SETUP_ERRS_ONLY

In diesem Modus wird bei Fehlern, die während der SETUP-Phase in einer Datenbankpartition auftreten (beispielsweise Fehler beim Zugriff auf eine Datenbankpartition oder Probleme beim Zugriff auf einen Tabellenbereich bzw. auf eine Tabelle in einer Datenbankpartition), die Ladeoperation in der Datenbankpartition mit dem Fehler gestoppt, in den übrigen Datenbankpartitionen jedoch fortgesetzt. Fehler, die in einer Datenbankpartition auftreten, während Daten geladen werden, bewirken das Fehlschlagen der gesamten Operation und den Rollback zum letzten Konsistenzpunkt in jeder Datenbankpartition.

DB2LOAD_LOAD_ERRS_ONLY

In diesem Modus bewirken Fehler, die während der SETUP-Phase in einer Datenbankpartition auftreten, das Fehlschlagen der gesamten Ladeoperation. Wenn beim Laden der Daten ein Fehler auftritt, werden die Datenbankpartitionen mit Fehlern auf ihren letzten Konsistenzpunkt zurückgesetzt. Die Ladeoperation wird für die verbleibenden Datenbankpartitionen fortgesetzt, bis ein Fehler auftritt oder bis alle Daten geladen wurden. Auf den Datenbankpartitionen, auf denen alle Daten geladen wurden, sind die Daten nach der Ladeoperation nicht sichtbar. Aufgrund der Fehler auf den anderen Datenbankpartitionen wird die Transaktion abgebrochen. Die Daten bleiben auf allen Datenbankpartitionen so lange nicht sichtbar, bis eine LOAD RESTART-Operation ausgeführt wird. Dies macht die neu geladenen Daten auf den Datenbankpartitionen sichtbar, auf denen die Ladeoperation vollständig ausgeführt wurde. Auf Datenbankpartitionen 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 Datenbankpartitionsebene während der SETUP-Phase oder beim Laden von Daten dazu, dass die Verarbeitung nur auf den betroffenen Datenbankpartitionen gestoppt wird. Wie im Modus DB2LOAD_LOAD_ERRS_ONLY bleiben die Daten - falls beim Laden der Daten Datenbankpartitionsfehler auftreten - in allen Datenbankpartitionen weiterhin 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 Verteilungszuordnung. 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 Verteilungszuordnung. 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 Datenbankpartitionsverteilungsdatei an. Bei Angabe von NULL wird der Wert DISTFILE als Standardwert verwendet.

piOmitHeader

Eingabe. Gibt an, dass die Datenbankpartitionsdatei keine Kopfdaten für die Verteilungszuordnung 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 Ausgabe-datenbankpartitionsliste.

Parameter der db2LoadNodeList-Datenstruktur**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.

Parameter der db2LoadPortRange-Datenstruktur**iPortMin**

Eingabe. Niedrigste Portnummer.

iPortMax

Eingabe. Höchste Portnummer.

Parameter der db2PartLoadOut-Datenstruktur**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 im Handbuch "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 Datenbankpartition, in der der Agent ausgeführt wurde.

oSqlcode

Der endgültige SQLCODE-Wert, der aus der Verarbeitung des Agenten resultierte.

oTableState

Der endgültige Status der Tabelle in der Datenbankpartition, in 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 wird. 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 Datenbankpartitionen.

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.

Parameter der db2LoadAgentInfo-Datenstruktur**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. Die möglichen Werte sind:

DB2LOADQUERY_NORMAL

Gibt an, dass der Ladevorgang in der Datenbankpartition 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 'Festlegen der Integrität 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 aufgrund eines Fehlers abgebrochen wurde, jedoch den Status der Tabelle in der Datenbankpartition noch nicht geändert hat. Daher ist der entsprechende Status gültig, der vor dem Aufruf von 'db2Load' vorlag. Es ist nicht erforderlich,

für solche Datenbankpartitionen eine LOAD RESTART- oder LOAD TERMINATE-Operation auszuführen.

DB2LOADQUERY_LOADPENDING

Gibt an, dass der Ladejob während der Verarbeitung abgebrochen wurde, die Tabelle in der Datenbankpartition jedoch im Status "Laden anstehend" belassen wurde. Dies gibt an, dass der Ladejob in der entsprechenden Datenbankpartition entweder beendet oder erneut gestartet werden muss.

oNodeNum

Ausgabe. Die Nummer der Datenbankpartition, in der der Agent ausgeführt wurde.

oAgentType

Ausgabe. Der Agenttyp. Gültige Werte (gemäß Definition in der Kopfdatei 'db2ApiDf' im INCLUDE-Verzeichnis) sind:

- DB2LOAD_LOAD_AGENT
- DB2LOAD_PARTITIONING_AGENT
- DB2LOAD_PRE_PARTITIONING_AGENT
- DB2LOAD_FILE_TRANSFER_AGENT
- DB2LOAD_LOAD_TO_FILE_AGENT

Parameter der db2gLoadStruct-Datenstruktur

iFileTypeLen

Eingabe. Gibt die Länge des Parameters 'iFileType' in Byte an.

iLocalMsgFileLen

Eingabe. Gibt die Länge des Parameters 'iLocalMsgFileName' in Byte an.

iTempFilesPathLen

Eingabe. Gibt die Länge des Parameters 'iTempFilesPath' in Byte an.

piXmlPathList

Eingabe. Zeiger auf eine Struktur 'sqlu_media_list', wobei das Feld 'media_type' auf SQLU_LOCAL_MEDIA gesetzt ist und die Struktur 'sqlu_media_entry' Pfade auf dem Client angibt, in denen sich die XML-Dateien befinden.

Parameter der db2gLoadIn-Datenstruktur

iUseTablespaceLen

Eingabe. Die Länge des Parameters 'piUseTablespace' in Byte.

piXmlParse

Eingabe. Typ der Syntexanalyse, die für XML-Dokumente ausgeführt werden soll. Gültige Werte gemäß der Kopfdatei 'db2ApiDf' im INCLUDE-Verzeichnis sind:

DB2DMU_XMLPARSE_PRESERVE_WS

Leerzeichen sollen erhalten bleiben.

DB2DMU_XMLPARSE_STRIP_WS

Leerzeichen sollen entfernt werden.

piXmlValidate

Eingabe. Ein Zeiger auf die 'db2DMUXmlValidate'-Struktur. Gibt an, dass für XML-Dokumente eine XML-Schemaprüfung durchgeführt werden soll.

```

/* XML-Prüfstruktur */
typedef SQL_STRUCTURE db2DMUXmlValidate
{
    db2Uuint16          iUsing;      /* Angaben zum Ausführen */
                                /* der Prüfung */
    struct db2DMUXmlValidateXds *piXdsArgs; /* Argumente für */
                                /* XMLVALIDATE USING XDS */
    struct db2DMUXmlValidateSchema *piSchemaArgs; /* Argumente für */
                                /* XMLVALIDATE USING SCHEMA */
} db2DMUXmlValidate;

```

Parameter der db2gPartLoadIn-Datenstruktur

piReserved1

Zur zukünftigen Verwendung reserviert.

iHostnameLen

Eingabe. Die Länge des Parameters 'piHostname' in Byte.

iFileTransferLen

Eingabe. Die Länge des Parameters 'piFileTransferCmd' in Byte.

iPartFileLocLen

Eingabe. Die Länge des Parameters 'piPartFileLocation' in Byte.

iMapFileInputLen

Eingabe. Die Länge des Parameters 'piMapFileInput' in Byte.

iMapFileOutputLen

Eingabe. Die Länge des Parameters 'piMapFileOutput' in Byte.

iDistfileLen

Eingabe. Die Länge des Parameters 'piDistfile' in Byte.

Hinweise 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. Mithilfe 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 'Festlegen der Integrität anstehend' versetzt. Übersichtstabellen, für die das Attribut REFRESH IMMEDIATE definiert ist und die von geladenen Tabellen abhängig sind, werden ebenfalls in den Status 'Festlegen der Integrität anstehend' versetzt. Durch Absetzen der Anweisung SET INTEGRITY können Sie den Status 'Festlegen der Integrität 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.

CLP-Beispiele für LOAD-Sitzungen

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 2 und 3  QQY  XXN
Testdaten 4,5 und 6 WWN6789  Y

```

Mit dem folgenden Befehl wird die Tabelle aus der Datei geladen:

```

db2 load from ascfile1 of asc modified by striptblanks reclen=40
      method L (1 20, 21 22, 24 27, 28 31)
      null indicators (0,0,23,32)
insert into table1 (col1, col5, col2, col3)

```

Anmerkung:

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 `reclen=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 `COL4` nicht in der Eingabedatei bereitgestellt wird, wird sie in `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 bestimmte Spalte kann sich an einer beliebigen Position im Eingabedatensatz befinden. Diese Position 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 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
```

Anmerkung:

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 4 (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)
```


Anmerkung:

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.

SET INTEGRITY

Die Anweisung SET INTEGRITY wird für folgende Zwecke verwendet:

- Für mindestens eine Tabelle den Status 'Festlegen der Integrität anstehend' (früher als Status 'Überprüfung anstehend' bezeichnet) aufheben, indem die erforderliche Integritätsverarbeitung für diese Tabellen ausgeführt wird.
- Für mindestens eine Tabelle den Status 'Festlegen der Integrität anstehend' aufheben, ohne die erforderliche Integritätsverarbeitung für diese Tabellen auszuführen.
- Mindestens eine Tabelle in den Status 'Festlegen der Integrität anstehend' versetzen.
- Mindestens eine Tabelle in den Status für uneingeschränkten Zugriff versetzen.
- Den Inhalt mindestens einer Zwischenspeichertabelle bereinigen.

Wenn die Anweisung zum Ausführen der Integritätsverarbeitung für eine Tabelle verwendet wird, nachdem die Tabelle geladen oder zugeordnet wurde, kann die Tabelle vom System teilweise verarbeitet werden, indem nur der angefügte Teil auf ungültige Integritätsbedingungen überprüft wird. Wenn die Subjekttable eine MQT- oder eine Zwischenspeichertabelle ist und LOAD-, ATTACH- oder DETACH-Operationen an ihren zu Grunde liegenden Tabellen ausgeführt werden, kann das System die MQT inkrementell aktualisieren oder inkrementell an die Zwischenspeichertabelle weitergeben (d. h. nur mit den geänderten Teilen der zu Grunde liegenden Tabellen). In manchen Situationen kann das System jedoch keine solchen Optimierungen vornehmen. Stattdessen wird die vollständige Integritätsverarbeitung durchgeführt, um die Datenintegrität zu gewährleisten. Bei der vollständigen Integritätsverarbeitung wird die gesamte Tabelle auf ungültige Integritätsbedingungen geprüft, die Definition einer MQT erneut berechnet oder eine Zwischenspeichertabelle als inkonsistent markiert. Die zuletzt genannte Variante setzt voraus, dass eine vollständige Aktualisierung der zugehörigen MQT erforderlich ist. Denkbar ist auch eine Situation, in der Sie die inkrementelle Verarbeitung durch Angeben der Option INCREMENTAL explizit anfordern.

Die Anweisung SET INTEGRITY unterliegt der Transaktionssteuerung.

Aufruf

Diese Anweisung kann in ein Anwendungsprogramm integriert oder mithilfe dynamischer SQL-Anweisungen abgesetzt werden. Sie ist eine ausführbare Anweisung, die nur dynamisch vorbereitet werden kann, wenn das Ausführungsverhalten DYNAMICRULES für das Paket wirksam ist (SQLSTATE 42509).

Berechtigung

Die erforderlichen Zugriffsrechte für die Ausführung der Anweisung SET INTEGRITY hängen vom Verwendungszweck der Anweisung ab (wie nachfolgend beschrieben).

- Den Status 'Festlegen der Integrität anstehend' für Tabellen aufheben und die erforderliche Integritätsverarbeitung ausführen.

Die Berechtigungs-ID der Anweisung muss mindestens eines der folgenden Zugriffsrechte aufweisen:

- Zugriffsrecht CONTROL für:
 - Die Tabellen, für die Integritätsverarbeitung durchgeführt wird und (wenn Ausnahmetabellen für mindestens eine dieser Tabellen bereitgestellt werden) das Zugriffsrecht INSERT für die Ausnahmetabellen
 - Alle untergeordneten Fremdschlüsseltabellen, untergeordneten IMQs und untergeordneten, sofort gespeicherten Zwischenspeichertabellen, die von der Anweisung implizit in den Status 'Festlegen der Integrität anstehend' versetzt werden
- Berechtigung LOAD (mit Bedingungen). Alle nachfolgenden Bedingungen müssen erfüllt sein, damit die Berechtigung LOAD als mit gültigen Zugriffsrechten ausgestattet angesehen werden kann:
 - Die erforderliche Integritätsverarbeitung beinhaltet nicht die folgenden Aktionen:
 - Aktualisieren einer MQT
 - Weitergeben einer Zwischenspeichertabelle
 - Aktualisieren einer generierten Spalte oder Identitätsspalte
 - Wenn Ausnahmetabellen für mindestens eine Tabelle bereitgestellt werden, wird für die Dauer der Integritätsverarbeitung der erforderliche Zugriff für die Tabellen erteilt, an denen die Integritätsverarbeitung durchgeführt wird, und für die zugeordneten Ausnahmetabellen. Das heißt:
 - Zugriffsrechte SELECT und DELETE für jede Tabelle, für die Integritätsverarbeitung ausgeführt wird, und
 - INSERT für die Ausnahmetabellen

– Berechtigung DATAACCESS

- Den Status 'Festlegen der Integrität anstehend' für Tabellen aufheben, ohne die erforderliche Integritätsverarbeitung für diese Tabellen auszuführen.

Die Berechtigungs-ID der Anweisung muss mindestens eines der folgenden Zugriffsrechte aufweisen:

- Zugriffsrecht CONTROL für alle zu verarbeitenden Tabellen, für alle untergeordneten Fremdschlüsseltabellen, untergeordneten IMQs und als direkt untergeordnet identifizierten Zwischenspeichertabellen, die von der Anweisung implizit in den Status 'Festlegung der Integrität anstehend' versetzt werden
- Berechtigung LOAD
- Berechtigung DATAACCESS
- Berechtigung DBADM

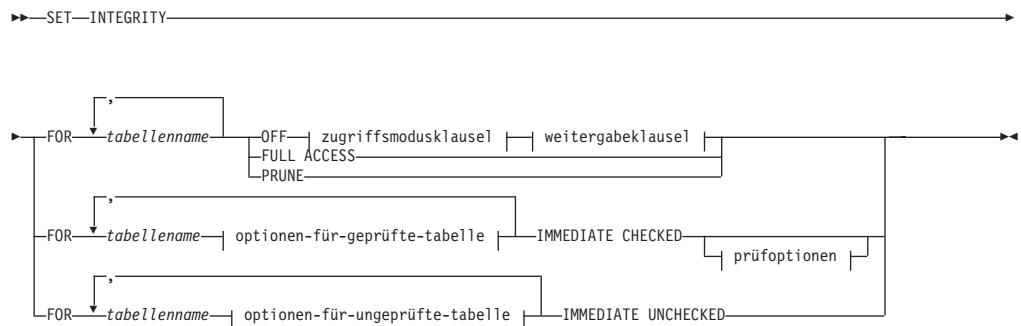
- Versetzen von Tabellen in den Status 'Festlegen der Integrität anstehend'.

Die Berechtigungs-ID der Anweisung muss mindestens eines der folgenden Zugriffsrechte aufweisen:

- Zugriffsrecht CONTROL für:
 - Die angegebenen Tabellen und

- Untergeordnete Fremdschlüsseltabellen, die von der Anweisung in den Status 'Festlegen der Integrität anstehend' versetzt werden, und
- Untergeordnete IMQTs, die von der Anweisung in den Status 'Festlegen der Integrität anstehend' versetzt werden, und
- Untergeordnete, sofort gespeicherte Zwischenspeichertabellen, die von der Anweisung in den Status 'Festlegen der Integrität anstehend' versetzt werden
- Berechtigung LOAD
- Berechtigung DATAACCESS
- Berechtigung DBADM
- Versetzen einer Tabelle in den Status für uneingeschränkten Zugriff.
Die Berechtigungs-ID der Anweisung muss mindestens eines der folgenden Zugriffsrechte aufweisen:
 - Zugriffsrecht CONTROL für die Tabellen, die in den Status für uneingeschränkten Zugriff versetzt werden
 - Berechtigung LOAD
 - Berechtigung DATAACCESS
 - Berechtigung DBADM
- Bereinigen einer Zwischenspeichertabelle.
Die Berechtigungs-ID der Anweisung muss mindestens eines der folgenden Zugriffsrechte aufweisen:
 - Zugriffsrecht CONTROL für die zu bereinigende Tabelle
 - Berechtigung DATAACCESS

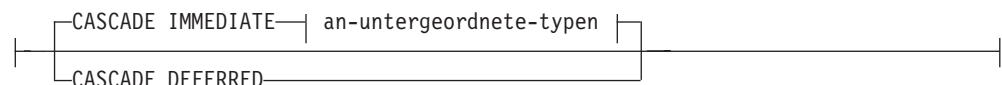
Syntax



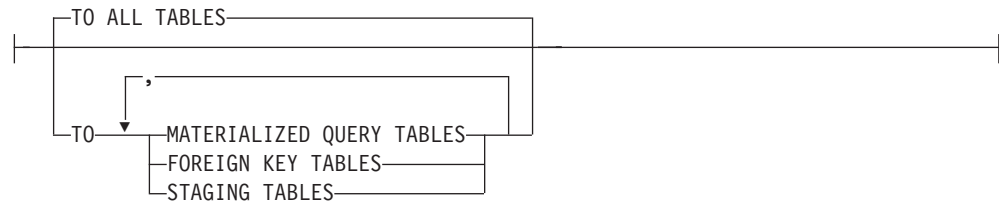
zugriffsmodusklausel:



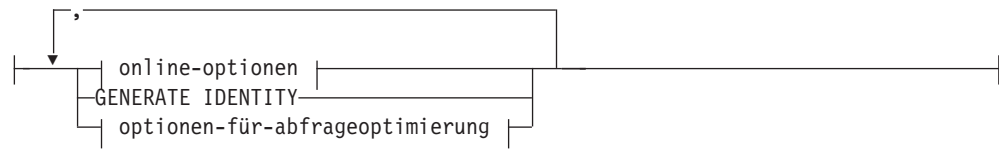
weitergabeklausel:



an-untergeordnete-typen:



optionen-für-geprüfte-tabelle:



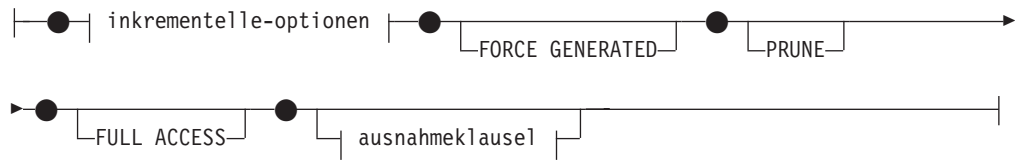
online-optionen:



optionen-für-abfrageoptimierung:



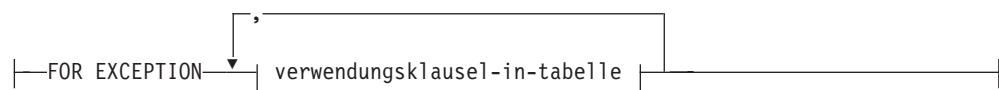
prüfoptionen:



inkrementelle-optionen:



ausnahmeklausel:



an-untergeordnete-typen

Gibt den Typ der untergeordneten Tabellen an, an die der Status 'Festlegen der Integrität anstehend' direkt weitergegeben wird.

TO ALL TABLES

Gibt an, dass der Status 'Festlegen der Integrität anstehend' direkt an alle untergeordneten Tabellen der in der Aufrufliste angegebenen Tabellen weitergegeben werden soll. Zu den untergeordneten Tabellen gehören alle untergeordneten Fremdschlüsseltabellen, sofort gespeicherten Zwischenspeichertabellen und IMQTs, die den in der Aufrufliste angegebenen Tabellen untergeordnet sind, oder untergeordnete Elemente der untergeordneten Fremdschlüsseltabellen.

Die Angabe TO ALL TABLES entspricht dem Angeben von TO FOREIGN KEY TABLES, TO MATERIALIZED QUERY TABLES und TO STAGING TABLES in einer einzigen Anweisung.

TO MATERIALIZED QUERY TABLES

Wenn nur TO MATERIALIZED QUERY TABLES angegeben ist, wird der Status 'Festlegen der Integrität anstehend' nur an untergeordnete IMQTs direkt weitergegeben. Andere untergeordnete Tabellen werden bei Bedarf möglicherweise später in den Status 'Festlegen der Integrität anstehend' versetzt, wenn der Status 'Festlegen der Integrität anstehend' der Tabelle aufgehoben wird. Wenn sowohl TO FOREIGN KEY TABLES als auch TO MATERIALIZED QUERY TABLES angegeben ist, wird der Status 'Festlegen der Integrität anstehend' direkt an alle untergeordneten Fremdschlüsseltabellen weitergegeben, an alle untergeordneten IMQTs der Tabellen in der Aufrufliste und an alle IMQTs, die untergeordnete Elemente der untergeordneten Fremdschlüsseltabellen sind.

TO FOREIGN KEY TABLES

Gibt an, dass der Status 'Festlegen der Integrität anstehend' direkt an untergeordnete Fremdschlüsseltabellen weitergegeben wird. Andere untergeordnete Tabellen werden bei Bedarf möglicherweise später in den Status 'Festlegen der Integrität anstehend' versetzt, wenn der Status 'Festlegen der Integrität anstehend' der Tabelle aufgehoben wird.

TO STAGING TABLES

Gibt an, dass der Status 'Festlegen der Integrität anstehend' direkt an untergeordnete Zwischenspeichertabellen weitergegeben wird. Andere untergeordnete Tabellen werden bei Bedarf möglicherweise später in den Status 'Festlegen der Integrität anstehend' versetzt, wenn der Status 'Festlegen der Integrität anstehend' der Tabelle aufgehoben wird. Wenn sowohl TO FOREIGN KEY TABLES als auch TO STAGING TABLES angegeben ist, wird der Status 'Festlegen der Integrität anstehend' direkt an alle untergeordneten Fremdschlüsseltabellen weitergegeben, an alle untergeordneten, sofort gespeicherten Zwischenspeichertabellen der Tabellen in der Aufrufliste, und an alle sofort gespeicherten Zwischenspeichertabellen, die untergeordnete Elemente der untergeordneten Fremdschlüsseltabellen sind.

CASCADE DEFERRED

Gibt an, dass nur die Tabellen in der Aufrufliste in den Status 'Festlegen der Integrität anstehend' versetzt werden sollen. Die Status der untergeordneten Tabellen bleiben unverändert. Untergeordnete Fremdschlüsseltabellen können später implizit in den Status 'Festlegen der Integrität anstehend' versetzt werden, wenn die dazugehörigen übergeordneten Tabellen auf ungültige Integritätsbedingungen geprüft werden. Untergeordnete IMQTs

und untergeordnete, sofort gespeicherte Zwischenspeichertabellen können implizit in den Status 'Festlegen der Integrität anstehend' versetzt werden, wenn eine ihrer zu Grunde liegenden Tabellen auf Integritätsverletzungen geprüft wird. Eine Abfrage einer Tabelle im Status 'Festlegen der Integrität anstehend' kann erfolgreich sein, wenn die Abfrage nicht auf die angegebene Tabelle zugreift, sondern auf eine infrage kommende MQT, die sich nicht im Status 'Festlegen der Integrität anstehend' befindet.

Wenn *weitergabeklausel* nicht angegeben ist, wird der Status 'Festlegen der Integrität anstehend' direkt an alle untergeordneten Tabellen weitergegeben.

IMMEDIATE CHECKED

Gibt an, dass der Status 'Festlegen der Integrität anstehend' für die Tabelle aufgehoben werden soll, wenn die erforderliche Integritätsverarbeitung an der Tabelle durchgeführt wird. Dies erfolgt in Übereinstimmung mit den in den Spalten STATUS und CONST_CHECKED der Katalogsicht SYSCAT.TABLES festgelegten Informationen. Das heißt:

- Der Wert in der Spalte STATUS muss 'C' sein (die Tabelle befindet sich im Status 'Festlegen der Integrität anstehend'), oder es wird ein Fehler zurückgegeben. Es sei denn, die Tabelle ist eine untergeordnete Fremdschlüsseltabelle, untergeordnete MQT oder untergeordnete Zwischenspeichertabelle einer in der Liste angegebenen Tabelle, die sich im Status 'Festlegen der Integrität anstehend' befindet, und deren direkte Vorgänger ebenfalls in der Liste enthalten sind.
- Wenn sich die zu prüfende Tabelle im Status 'Festlegen der Integrität anstehend' befindet, gibt der Wert in CONST_CHECKED an, welche Integritätsoptionen geprüft werden sollen.

Wird der Status 'Festlegen der Integrität anstehend' der Tabelle aufgehoben, so werden ihre untergeordneten Tabellen in den Status 'Festlegen der Integrität anstehend' versetzt (falls erforderlich). Eine Warnung weist darauf hin, dass untergeordnete Tabellen in den Status 'Festlegen der Integrität anstehend' versetzt wurden (SQLSTATE 01586).

Ist die Tabelle eine vom System verwaltete MQT, werden die Daten gegen die Abfrage abgeglichen und gegebenenfalls aktualisiert. (IMMEDIATE CHECKED kann für vom Benutzer verwaltete MQTs nicht verwendet werden.) Ist die Tabelle eine Zwischenspeichertabelle, werden die Daten gegen die zugehörige Abfragedefinition abgeglichen und nach Bedarf weitergegeben.

Wenn die Integrität einer untergeordneten Tabelle geprüft wird, gilt Folgendes:

- Keines der übergeordneten Objekte kann in den Status 'Festlegen der Integrität anstehend' versetzt werden oder
- Jedes übergeordnete Objekt muss in derselben Anweisung SET INTEGRITY auf ungültige Integritätsbedingungen geprüft werden

Wenn eine IMQT aktualisiert wird oder Deltas an die Zwischenspeichertabelle weitergegeben werden, gilt Folgendes:

- Keine der zu Grunde liegenden Tabellen kann in den Status 'Festlegen der Integrität anstehend' versetzt werden oder
- Jede der zu Grunde liegenden Tabellen muss in derselben Anweisung SET INTEGRITY geprüft werden

Andernfalls wird ein Fehler zurückgegeben (SQLSTATE 428A8).

optionen-für-geprüfte-tabelle

online-optionen

Gibt die Zugriffsmöglichkeiten auf die Tabelle während ihrer Verarbeitung an.

ALLOW NO ACCESS

Gibt an, dass keine anderen Benutzer auf die Tabelle zugreifen können, während sie verarbeitet wird, es sei denn, sie arbeiten mit der Isolationsstufe UR (Uncommitted Read).

ALLOW READ ACCESS

Gibt an, dass andere Benutzer Lesezugriff auf die Tabelle haben, während sie verarbeitet wird.

ALLOW WRITE ACCESS

Gibt an, dass andere Benutzer Lese- und Schreibzugriff auf die Tabelle haben, während sie verarbeitet wird.

GENERATE IDENTITY

Gibt an, dass die Werte von der Anweisung SET INTEGRITY generiert werden, wenn die Tabelle eine Identitätsspalte enthält. Wenn die Option GENERATE IDENTITY angegeben ist, werden standardmäßig nur die Identitätsspaltenwerte der zugeordneten Zeile von der Anweisung SET INTEGRITY generiert. Die Option NOT INCREMENTAL muss zusammen mit der Option GENERATE IDENTITY angegeben werden, damit die Anweisung SET INTEGRITY Identitätsspaltenwerte für alle Zeilen in der Tabelle (einschließlich zugeordnete, geladene und vorhandene Zeilen) generiert. Ist die Option GENERATE IDENTITY nicht angegeben, bleiben die aktuellen Identitätsspaltenwerte aller Zeilen in der Tabelle unverändert.

optionen-für-abfrageoptimierung

Gibt die Abfrageoptimierungsoptionen für die Verwaltung von MQTs an, die als REFRESH DEFERRED deklariert sind.

ALLOW QUERY OPTIMIZATION USING REFRESH DEFERRED TABLES WITH REFRESH AGE ANY

Gibt Folgendes an: Wenn das Sonderregister CURRENT REFRESH AGE auf ANY gesetzt ist, können bei der Verwaltung von *tabellenname* als REFRESH DEFERRED deklarierte MQTs verwendet werden, um die Abfrage zum Verwalten von *tabellenname* zu optimieren. Ist *tabellenname* keine als REFRESH DEFERRED deklarierte MQT, wird ein Fehler zurückgegeben (SQLSTATE 428FH). Als REFRESH IMMEDIATE deklarierte Abfragetabellen werden bei der Abfrageoptimierung immer berücksichtigt.

prüfoptionen

inkrementelle-optionen

INCREMENTAL

Gibt an, dass Integritätsverarbeitung auf den angehängten Teil der Tabelle (falls vorhanden) angewendet wird. Kann eine solche Anforderung nicht erfüllt werden (z. B. wenn das System feststellt, dass die ganze Tabelle auf Datenintegrität geprüft werden muss), wird ein Fehler zurückgegeben (SQLSTATE 55019).

NOT INCREMENTAL

Gibt an, dass Integritätsverarbeitung auf die ganze Tabelle angewendet wird. Ist die Tabelle eine MQT, wird die MQT-Definition erneut berechnet. Ist für die Tabelle mindestens eine Integritätsbedingung definiert, so bewirkt diese Option die vollständige Verar-

beitung der untergeordneten Fremdschlüsseltabellen und der untergeordneten IMQs. Ist die Tabelle eine Zwischenspeichertabelle, wird sie in einen inkonsistenten Status versetzt.

Ist die Klausel *inkrementelle-optionen* nicht angegeben, wird vom System ermittelt, ob inkrementelle Verarbeitung möglich ist. Ist dies nicht der Fall, wird die ganze Tabelle geprüft.

FORCE GENERATED

Wenn die Tabelle Spalten mit GENERATED BY-Ausdruck enthält, werden die Werte auf der Basis des Ausdrucks berechnet und in der Spalte gespeichert. Wenn diese Option nicht angegeben ist, werden die aktuellen Werte mit dem berechneten Wert des Ausdrucks verglichen (wie bei einer aktiven Integritätsbedingung für Gleichheitsprüfung). Bei inkrementeller Integritätsverarbeitung der Tabelle werden nur für den angefügten Teil generierte Spalten berechnet.

PRUNE

Diese Option kann nur für Zwischenspeichertabellen angegeben werden. Gibt an, dass der Inhalt der Zwischenspeichertabelle bereinigt und die Zwischenspeichertabelle in einen inkonsistenten Status versetzt werden soll. Ist eine Tabelle in der Liste *tabellenname* keine Zwischenspeichertabelle, wird ein Fehler zurückgegeben (SQLSTATE 428FH). Ist die Prüfoption INCREMENTAL ebenfalls angegeben, wird ein Fehler zurückgegeben (SQLSTATE 428FH).

FULL ACCESS

Gibt an, dass die Tabelle nach Ausführung der Anweisung SET INTEGRITY in den Status für uneingeschränkten Zugriff versetzt werden soll.

Bei inkrementeller Verarbeitung einer zu Grunde liegenden Tabelle (die über abhängige IMQs oder abhängige, sofort gespeicherte Zwischenspeichertabellen verfügt) aus der Aufrufliste, wird die zu Grunde liegende Tabelle nach Ausführung der Anweisung SET INTEGRITY bei Bedarf in den Status 'Kein Versetzen von Daten' versetzt. Wenn der Status 'Festlegen der Integrität anstehend' für alle inkrementell aktualisierbaren, abhängigen IMQs und Zwischenspeichertabellen aufgehoben wird, wird die zu Grunde liegende Tabelle automatisch aus dem Status 'Kein Versetzen von Daten' in den Status für uneingeschränkten Zugriff versetzt. Wird die Option FULL ACCESS mit der Option IMMEDIATE CHECKED angegeben, wird die zu Grunde liegende Tabelle direkt in den Status für uneingeschränkten Zugriff versetzt (unter Umgehung des Status 'Kein Versetzen von Daten'). In DB2 Version 9.7 Fixpack 1 und späteren Fixpacks wird durch Angabe der Option FULL ACCESS nur die Abhängigkeit zwischen den abhängigen Tabellen und der zu Grunde liegenden Tabelle entfernt. Die zu Grunde liegende Tabelle ist erst dann wieder verfügbar, wenn der DETACH-Prozess von der Task zur asynchronen Aufhebung von Partitionszuordnungen abgeschlossen wurde.

Abhängige IMQs, die nicht aktualisiert wurden, werden möglicherweise in der nachfolgenden Anweisung REFRESH TABLE vollständig neu berechnet, und abhängige, sofort gespeicherte Zwischenspeichertabellen, an die die angefügten Teile der Tabelle nicht weitergegeben wurden, werden möglicherweise als inkonsistent markiert.

Wenn für eine zu Grunde liegende Tabelle in der Aufrufliste vollständige Verarbeitung erforderlich ist, oder die Tabelle keine abhängigen IMQs oder keine abhängigen, sofort gespeicherten Zwischenspeicherta-

ellen hat, wird die zu Grunde liegende Tabelle nach Ausführung der Anweisung SET INTEGRITY direkt in den Status für vollständigen Zugriff versetzt (unabhängig davon, ob die Option FULL ACCESS angegeben wurde).

ausnahmeklausel

FOR EXCEPTION

Gibt an, dass jede Zeile, die gegen eine zu prüfende Integritätsbedingung verstößt, in eine Ausnahmetabelle versetzt werden soll. Selbst wenn Fehler festgestellt werden, wird der Status 'Festlegen der Integrität anstehend' für die Tabelle aufgehoben. Eine Warnung weist darauf hin, dass mindestens eine Zeile in die Ausnahmetabelle versetzt wurde (SQLSTATE 01603).

Wenn die Option FOR EXCEPTION nicht angegeben ist und gegen Integritätsbedingungen verstoßen wird, wird nur der erste festgestellte Verstoß zurückgegeben (SQLSTATE 23514). Tritt in einer Tabelle ein Verstoß auf, bleiben alle Tabellen im Status 'Festlegen der Integrität anstehend'.

Es wird empfohlen, bei der Prüfung auf ungültige Integritätsbedingungen immer die Option FOR EXCEPTION zu verwenden, um das Rollback der Anweisung SET INTEGRITY zu verhindern, wenn ein Verstoß gefunden wird.

IN *tabellenname*

Gibt an, aus welcher Tabelle Zeilen, die gegen Integritätsbedingungen verstoßen, versetzt werden sollen. Für jede zu prüfende Tabelle muss eine Ausnahmetabelle angegeben werden. Diese Klausel kann nicht für eine MQT oder eine Zwischenspeichertabelle angegeben werden (SQLSTATE 428A7).

USE *tabellenname*

Gibt die Ausnahmetabelle an, in die fehlerhafte Zeilen versetzt werden sollen.

FULL ACCESS

Wenn die Option FULL ACCESS als einzige Operation der Anweisung angegeben ist, wird die Tabelle ohne erneute Prüfung auf Integritätsverstöße in den Status für uneingeschränkten Zugriff versetzt. Für abhängige IMQs, die nicht aktualisiert wurden, ist in nachfolgenden REFRESH TABLE-Anweisungen möglicherweise eine vollständige Neuberechnung erforderlich, und abhängige, sofort gespeicherte Zwischenspeichertabellen, an die die Deltas der Tabelle nicht weitergegeben wurden, werden möglicherweise in einen unvollständigen Status versetzt. Diese Option kann nur für eine Tabelle angegeben werden, die sich im Status 'Kein Versetzen von Daten' oder im Nicht-Zugriffs-Status befindet, jedoch nicht für eine Tabelle im Status 'Festlegen der Integrität anstehend' (SQLSTATE 428FH).

PRUNE

Diese Option kann nur für Zwischenspeichertabellen angegeben werden. Gibt an, dass der Inhalt der Zwischenspeichertabelle bereinigt und die Zwischenspeichertabelle in einen inkonsistenten Status versetzt werden soll. Ist eine Tabelle in der Liste *tabellenname* keine Zwischenspeichertabelle, wird ein Fehler zurückgegeben (SQLSTATE 428FH).

optionen-für-ungeprüfte-tabelle

integritätsoptionen

Definiert die Typen der erforderlichen Integritätsverarbeitung, die umgangen werden sollen, wenn der Status 'Festlegen der Integrität anstehend' für die Tabelle aufgehoben wird.

ALL

Für die Tabelle wird der Status 'Festlegen der Integrität anstehend' sofort aufgehoben, ohne die erforderliche Integritätsverarbeitung für die Tabelle auszuführen.

FOREIGN KEY

Beim Aufheben des Status 'Festlegen der Integrität anstehend' für die Tabelle wird die erforderliche Prüfung auf Fremdschlüsselintegritätsbedingungen nicht ausgeführt.

CHECK

Beim Aufheben des Status 'Festlegen der Integrität anstehend' für die Tabelle wird die erforderliche Prüfung auf Integritätsbedingungen nicht ausgeführt.

MATERIALIZED QUERY

Beim Aufheben des Status 'Festlegen der Integrität anstehend' für die Tabelle wird die erforderliche Aktualisierung einer MQT nicht ausgeführt.

GENERATED COLUMN

Beim Aufheben des Status 'Festlegen der Integrität anstehend' für die Tabelle wird die erforderliche Prüfung auf Integritätsbedingungen für generierte Spalten nicht ausgeführt.

STAGING

Beim Aufheben des Status 'Festlegen der Integrität anstehend' für die Tabelle wird die erforderliche Datenweitergabe an eine Zwischenspeichertabelle nicht ausgeführt.

Wenn keine anderen Typen der Integritätsverarbeitung für die Tabelle erforderlich sind, nachdem ein bestimmter Typ der Integritätsverarbeitung als umgangen markiert wurde, wird der Status 'Festlegen der Integrität anstehend' für die Tabelle sofort aufgehoben.

FULL ACCESS

Gibt an, dass die Tabellen nach Ausführung der Anweisung SET INTEGRITY in den Status für uneingeschränkten Zugriff versetzt werden sollen.

Bei inkrementeller Verarbeitung einer zu Grunde liegenden Tabelle aus der Aufrufliste, die über abhängige MQTs oder abhängige, sofort gespeicherte Zwischenspeichertabellen verfügt, wird die zu Grunde liegende Tabelle nach Ausführung der Anweisung SET INTEGRITY bei Bedarf in den Status 'Kein Versetzen von Daten' versetzt. Wenn der Status 'Festlegen der Integrität anstehend' für alle inkrementell aktualisierbaren, abhängigen IMQTs und Zwischenspeichertabellen aufgehoben wurde, wird die zu Grunde liegende Tabelle automatisch aus dem Status 'Kein Versetzen von Daten' in den Status für uneingeschränkten Zugriff versetzt. Wird die Option FULL ACCESS mit der Option IMMEDIATE UNCHECKED angegeben, wird die zu Grunde liegende Tabelle direkt in den Status für uneingeschränkten Zugriff versetzt (unter Umgehung des Status 'Kein Versetzen von Daten'). Abhängige IMQTs, die nicht aktualisiert wurden, werden möglicherweise in der nachfolgenden Anweisung REFRESH TABLE vollständig neu berech-

net, und abhängige, sofort gespeicherte Zwischenspeichertabellen, an die die angefügten Teile der Tabelle nicht weitergegeben wurden, werden möglicherweise als inkonsistent markiert.

In DB2 Version 9.7 Fixpack 1 und späteren Fixpacks wird durch Angabe der Option FULL ACCESS nur die Abhängigkeit zwischen den abhängigen Tabellen und der zu Grunde liegenden Tabelle entfernt. Die zu Grunde liegende Tabelle ist erst dann wieder verfügbar, wenn der DETACH-Prozess von der Task zur asynchronen Aufhebung von Partitionszuordnungen abgeschlossen wurde.

Wenn für eine zu Grunde liegende Tabelle in der Aufrufliste vollständige Verarbeitung erforderlich ist oder die Tabelle keine abhängigen IMQs oder abhängigen, sofort gespeicherten Zwischenspeichertabellen hat, wird die zu Grunde liegende Tabelle nach Ausführung der Anweisung SET INTEGRITY direkt in den Status für uneingeschränkten Zugriff versetzt (unabhängig davon, ob die Option FULL ACCESS angegeben wurde).

Wenn die Option FULL ACCESS zusammen mit der Option IMMEDIATE UNCHECKED angegeben wurde und die Anweisung den Status 'Festlegen der Integrität anstehend' für die Tabelle nicht aufhebt, wird ein Fehler zurückgegeben (SQLSTATE 428FH).

IMMEDIATE UNCHECKED

Gibt eine der folgenden Möglichkeiten an:

- Für die Tabelle soll der Status 'Festlegen der Integrität anstehend' sofort aufgehoben werden, ohne die erforderliche Integritätsverarbeitung auszuführen.
- Für die Tabelle soll mindestens ein Typ der erforderlichen Integritätsverarbeitung umgangen werden, wenn der Status 'Festlegen der Integrität anstehend' für die Tabelle durch eine nachfolgende Anweisung SET INTEGRITY mit der Option IMMEDIATE CHECKED aufgehoben wird.

Beachten Sie die Auswirkungen auf die Datenintegrität, wenn Sie diese Option verwenden. Siehe hierzu den nachfolgenden Abschnitt „Hinweise“.

Hinweise

- Auswirkungen auf Tabellen in einem der eingeschränkten Status für Festlegen der Integrität (SET INTEGRITY):
 - Die Verwendung von INSERT, UPDATE oder DELETE ist nicht zulässig, wenn sich die Tabelle im Lesezugriffsstatus oder im Nicht-Zugriffs-Status befindet. Außerdem wird jede Anweisung zurückgewiesen, die diesen Änderungstyp für eine Tabelle erfordert, die sich in einem solchen Status befindet. Beispielsweise ist das Löschen einer Zeile in einer übergeordneten Tabelle, die an eine abhängige Tabelle im Nicht-Zugriffs-Status weitergegeben wird, nicht zulässig.
 - SELECT kann nicht für eine Tabelle ausgeführt werden, die sich im Nicht-Zugriffs-Status befindet. Außerdem wird jede Anweisung zurückgewiesen, die Lesezugriff für eine Tabelle im Nicht-Zugriffs-Status erfordert.
 - Neu hinzugefügte Integritätsbedingungen für eine Tabelle werden normalerweise sofort umgesetzt. Befindet sich die Tabelle jedoch im Status 'Festlegen der Integrität anstehend', erfolgt die Prüfung der neuen Integritätsbedingungen erst, wenn der Status 'Festlegen der Integrität anstehend' für die Tabelle aufgehoben wird. Befindet sich die Tabelle im Status 'Festlegen der Integrität anstehend', wird sie beim Hinzufügen einer neuen Integritätsbedingung in den Status 'Festlegen der Integrität anstehend, kein Zugriff' versetzt, weil die Datenintegrität gefährdet ist.

- Die Anweisung CREATE INDEX kann nicht auf eine Tabelle verweisen, die sich im Lesezugriffsstatus oder im Nicht-Zugriffs-Status befindet. Ebenso kann eine Anweisung ALTER TABLE zum Hinzufügen eines Primärschlüssels oder einer eindeutigen Integritätsbedingung nicht auf eine Tabelle verweisen, die sich im Lesezugriffsstatus oder im Nicht-Zugriffs-Status befindet.
- Das Dienstprogramm IMPORT kann nicht auf eine Tabelle angewendet werden, die sich im Lesezugriffsstatus oder im Nicht-Zugriffs-Status befindet.
- Das Dienstprogramm EXPORT kann nicht auf eine Tabelle angewendet werden, die sich im Nicht-Zugriffs-Status befindet, es kann jedoch auf eine Tabelle im Lesezugriffsstatus angewendet werden. Wenn sich eine Tabelle im Lesezugriffsstatus befindet, exportiert das Dienstprogramm EXPORT nur die Daten aus dem nicht angefügten Teil der Tabelle.
- Operationen wie REORG, REDISTRIBUTE, update distribution key, update multidimensional clustering key, update range clustering key, update table partitioning key usw., die das Versetzen von Daten innerhalb einer Tabelle erfordern können, sind für eine Tabelle, die sich in einem der folgenden Status befindet, nicht zulässig: Lesezugriff, Nicht-Zugriff oder kein Versetzen von Daten.
- Die Dienstprogramme LOAD, BACKUP, RESTORE, UPDATE STATISTICS, RUNSTATS, REORGCHK, LIST HISTORY, und ROLLFORWARD sind für eine Tabelle zulässig, die sich in einem der folgenden Status befindet: uneingeschränkter Zugriff, Lesezugriff, kein Zugriff oder kein Versetzen von Daten.
- Die Anweisungen ALTER TABLE, COMMENT, DROP TABLE, CREATE ALIAS, CREATE TRIGGER, CREATE VIEW, GRANT, REVOKE und SET INTEGRITY können auf eine Tabelle verweisen, die sich in einem der folgenden Status befindet: uneingeschränkter Zugriff, Lesezugriff, kein Zugriff oder kein Versetzen von Daten. Diese Anweisungen können jedoch dazu führen, dass die Tabelle in den Nicht-Zugriffs-Status versetzt wird.
- Pakete, Sichten und andere Objekte, die von einer Tabelle im Nicht-Zugriffs-Status abhängen, geben einen Fehler zurück, wenn während der Ausführungszeit auf die Tabelle zugegriffen wird. Pakete, die von einer Tabelle mit Lesezugriffsstatus abhängen, geben einen Fehler zurück, wenn während der Ausführungszeit eine Operation INSERT, UPDATE oder DELETE für die Tabelle eingeleitet wird.

Das Entfernen von Zeilen mit Integritätsverstößen durch die Anweisung SET INTEGRITY ist keine DELETE-Operation. Darum kann eine Anweisung SET INTEGRITY keine Trigger auslösen. Ebenso werden beim Aktualisieren generierter Spalten durch die Option FORCE GENERATED keine Trigger aktiviert.

- Vorsicht bei Verwendung der Klausel IMMEDIATE UNCHECKED:
 - Diese Klausel ist für die Verwendung durch Dienstprogramme vorgesehen, sie sollte nicht in Anwendungsprogrammen verwendet werden. Enthält die Tabelle Daten, die nicht den für die Tabelle definierten Integritätsspezifikationen entsprechen, und wird die Option IMMEDIATE UNCHECKED verwendet, kann dies zu fehlerhaften Abfrageergebnissen führen.

Das Aufheben des Status 'Festlegen der Integrität anstehend' für die Tabelle ohne Ausführen der erforderlichen Integritätsverarbeitung wird im Katalog aufgezeichnet (das entsprechende Byte in der Spalte CONST_CHECKED der Sicht SYSCAT.TABLES wird auf 'U' gesetzt). Dies bedeutet, dass der Benutzer die Verantwortung für die Datenintegrität im Hinblick auf die spezifischen Integritätsbedingungen übernommen hat. Dieser Wert bleibt unverändert, bis eines der folgenden Ereignisse eintritt:

- Die Tabelle wird wieder in den Status 'Festlegen der Integrität anstehend' versetzt (durch einen Verweis auf die Tabelle in einer Anweisung SET IN-

TEGRITY mit der Option OFF). Zu diesem Zeitpunkt wird der Wert 'U' in der Spalte CONST_CHECKED in 'W' geändert, um anzugeben, dass der Benutzer zuvor die Verantwortung für die Datenintegrität übernommen hatte, und dass die Daten jetzt vom System geprüft werden müssen.

- Alle nicht geprüften Integritätsbedingungen für die Tabelle werden gelöscht.

Der Status 'W' unterscheidet sich vom Status 'N' dadurch, dass die Datenintegrität vorher zwar vom Benutzer geprüft wurde, aber nicht vom System. Wenn der Benutzer die Anweisung SET INTEGRITY ... IMMEDIATE CHECKED mit der Option NOT INCREMENTAL absetzt, prüft das System erneut die Datenintegrität der ganzen Tabelle (oder führt eine vollständige Aktualisierung für eine MQT durch) und ändert anschließend den Status 'W' in den Status 'Y'. Ist IMMEDIATE UNCHECKED angegeben, oder ist NOT INCREMENTAL nicht angegeben, wird der Status 'W' in den Status 'U' geändert, um zu protokollieren, dass manche Daten noch nicht vom System geprüft wurden. Im zuletzt genannten Fall (wenn NOT INCREMENTAL nicht angegeben ist) wird eine Warnung zurückgegeben (SQLSTATE 01636).

Wenn die Integrität einer zu Grunde liegenden Tabelle mit der Klausel IMMEDIATE UNCHECKED geprüft wurde, werden die Werte 'U' in der Spalte CONST_CHECKED der zu Grunde liegenden Tabelle an die entsprechende Spalte CONST_CHECKED der folgenden Tabellen weitergegeben:

- Abhängige IMQTs (Immediate Materialized Query Tables)
- Abhängige, verzögerte MQTs (Materialized Query Tables)
- Abhängige Zwischenspeichertabellen

Für eine abhängige IMQT erfolgt diese Weitergabe, wenn für die zu Grunde liegende Tabelle der Status 'Festlegen der Integrität anstehend' aufgehoben oder die MQT aktualisiert wird. Für eine abhängige, verzögerte MQT erfolgt diese Weitergabe, wenn die MQT aktualisiert wird. Für abhängige Zwischenspeichertabellen erfolgt diese Weitergabe, wenn der Status 'Festlegen der Integrität anstehend' für die zu Grunde liegende Tabelle aufgehoben wird. Diese weitergegebenen 'U'-Werte in den CONST_CHECKED-Spalten der abhängigen MQTs und Zwischenspeichertabellen geben an, dass diese MQTs und Zwischenspeichertabellen von einer zu Grunde liegenden Tabelle abhängig sind, deren erforderliche Integritätsverarbeitung durch die Option IMMEDIATE UNCHECKED umgangen wurde.

Für eine MQT bleibt der Wert 'U' in der Spalte CONST_CHECKED, der von der zu Grunde liegenden Tabelle weitergegeben wurde, erhalten, bis die MQT vollständig aktualisiert wird und keine ihrer zu Grunde liegenden Tabellen einen Wert 'U' in der entsprechenden Spalte CONST_CHECKED enthält. Nach einer solchen Aktualisierung wird der Wert 'U' in der Spalte CONST_CHECKED für die MQT in 'Y' geändert.

Für eine Zwischenspeichertabelle bleibt der Wert 'U' in der Spalte CONST_CHECKED, der von der zu Grunde liegenden Tabelle weitergegeben wurde, erhalten, bis die entsprechende verzögerte MQT der Zwischenspeichertabelle aktualisiert wird. Nach einer solchen Aktualisierung wird der Wert 'U' in der Spalte CONST_CHECKED für die Zwischenspeichertabelle in 'Y' geändert.

- Wenn eine untergeordnete Tabelle und ihre übergeordnete Tabelle in derselben Anweisung SET INTEGRITY mit der Option IMMEDIATE CHECKED geprüft werden und die übergeordnete Tabelle eine vollständige Prüfung ihrer Integritätsbedingungen erfordert, werden die Fremdschlüsselintegritätsbedingungen der untergeordneten Tabelle geprüft (unabhängig davon, ob die untergeordnete Tabelle einen Wert 'U' in der Spalte CONST_CHECKED für Fremdschlüsselintegritätsbedingungen enthält).

- Nach dem Anfügen von Daten mit LOAD INSERT oder ALTER TABLE ATTACH prüft die Anweisung SET INTEGRITY mit der Option IMMEDIATE CHECKED die Tabelle auf ungültige Integritätsbedingungen. Das System ermittelt, ob die inkrementelle Verarbeitung für die Tabelle möglich ist. Ist dies der Fall, wird nur der angefügte Teil auf Integritätsverletzungen geprüft. Andernfalls prüft das System die ganze Tabelle auf Integritätsverletzungen.

- Betrachten Sie die folgende Anweisung:

SET INTEGRITY FOR T IMMEDIATE CHECKED

In den folgenden Szenarios wird weder die Prüfoption INCREMENTAL für T noch eine inkrementelle Aktualisierung von T (wenn T eine MQT oder eine Zwischenspeichertabelle ist) unterstützt:

- T wurden neue Integritätsbedingungen hinzugefügt, während die Tabelle sich im Status "Festlegen der Integrität anstehend" befand.
 - Wenn eine Operation LOAD REPLACE für T, die übergeordneten Elemente von T oder die zu Grunde liegenden Tabellen von T ausgeführt wurde
 - Wenn die Option NOT LOGGED INITIALLY WITH EMPTY TABLE aktiviert wurde, nachdem die letzte Prüfung auf Integritätsbedingungen für T, die zugehörigen übergeordneten Elemente oder die zugehörigen zu Grunde liegenden Tabellen ausgeführt wurde
 - Der Weitergabeeffekt der vollständigen Verarbeitung, wenn ein übergeordnetes Element von T (oder eine zu Grunde liegende Tabelle, wenn T eine MQT oder eine Zwischenspeichertabelle ist) ohne inkrementelle Verarbeitung auf Integritätsbedingungen geprüft wurde
 - Wenn der Tabellenbereich, der die Tabelle oder ihr übergeordnetes Element enthält (bzw. die zu Grunde liegende Tabelle einer MQT oder Zwischenspeichertabelle) bis zu einem bestimmten Zeitpunkt aktualisierend wiederhergestellt wurde und sich die Tabelle und ihr übergeordnetes Element (bzw. die zu Grunde liegende Tabelle, wenn die Tabelle eine MQT oder eine Zwischenspeichertabelle ist) in verschiedenen Tabellenbereichen befinden
 - T ist eine MQT, und nach der letzten Aktualisierung wurde eine Operation LOAD REPLACE oder LOAD INSERT direkt in T ausgeführt.
- Die inkrementelle Verarbeitung wird immer verwendet, wenn die Situation dies zulässt, da sie besonders effizient ist. In den meisten Fällen ist die Option INCREMENTAL nicht erforderlich. Sie wird jedoch benötigt, um sicherzustellen, dass Integritätsprüfungen tatsächlich inkrementell verarbeitet werden. Stellt das System fest, dass eine vollständige Verarbeitung erforderlich ist, um die Datenintegrität zu gewährleisten, wird ein Fehler zurückgegeben (SQLSTATE 55019).
 - Sind die im vorherigen Listenpunkt beschriebenen Bedingungen für vollständige Verarbeitung nicht erfüllt, versucht das System, nur den angefügten Teil auf Integritätsbedingungen zu prüfen oder eine inkrementelle Aktualisierung auszuführen (wenn es sich um eine MQT handelt), wenn der Benutzer die Option NOT INCREMENTAL für die Anweisung SET INTEGRITY FOR T IMMEDIATE CHECKED nicht angibt.
 - Tritt während der Integritätsverarbeitung ein Fehler auf, werden alle Verarbeitungsschritte (auch das Löschen aus den Quellentabelle und das Einfügen in die Ausnahmetabellen) rückgängig gemacht.
 - Schlägt eine Anweisung SET INTEGRITY mit angegebener Option FORCE GENERATED durch einen Mangel an Protokollspeicherbereich fehl, vergrößern Sie den verfügbaren aktiven Protokollspeicherbereich, und setzen Sie die Anweisung SET INTEGRITY erneut ab. Alternativ können Sie die Anweisung SET INTEGRITY mit den Optionen GENERATED COLUMN und IMMEDIATE UNCHECKED verwenden, um die Prüfung der generierten Spalten für die Tabelle zu umgehen.

Setzen Sie anschließend eine Anweisung SET INTEGRITY mit der Option IMMEDIATE CHECKED und ohne die Option FORCE GENERATED ab, um die Tabelle auf andere Integritätsverletzungen zu prüfen (falls zutreffend) und den Status 'Festlegen der Integrität anstehend' für die Tabelle aufzuheben. Nachdem der Status 'Festlegen der Integrität anstehend' für die Tabelle aufgehoben wurde, können die generierten Spalten mit den zugehörigen Standardwerten (generierten Werten) aktualisiert werden, indem diese dem Schlüsselwort DEFAULT in einer Anweisung UPDATE zugeordnet werden. Dies kann mithilfe von mehreren Anweisungen für die Aktualisierung mit Suche auf der Basis von Bereichen (auf die jeweils ein COMMIT folgt) erreicht werden, oder mit einer Cursormethode, die nicht unterbrechungsfreie COMMIT-Operationen verwendet. Ein Cursor „with hold“ sollte verwendet werden, wenn die Sperren nach nicht unterbrechungsfreien COMMIT-Operationen mithilfe der Cursormethode erhalten bleiben sollen.

- Für eine Tabelle, die mit der Option CASCADE DEFERRED in der Anweisung SET INTEGRITY oder im Befehl LOAD in den Status 'Festlegen der Integrität anstehend' versetzt wurde, oder durch die Anweisung ALTER TABLE mit der Klausel ATTACH, und die mit der Option IMMEDIATE CHECKED der Anweisung SET INTEGRITY auf Integritätsverletzungen geprüft wird, werden die zugehörigen untergeordneten Fremdschlüsselspalten, die untergeordneten IMQs und die untergeordneten, sofort gespeicherten Zwischenspeichertabellen bei Bedarf wie folgt in den Status 'Festlegen der Integrität anstehend' versetzt:
 - Wenn die ganze Tabelle auf Integritätsverletzungen geprüft wird, werden ihre untergeordneten Fremdschlüsseltabellen, untergeordneten IMQs und untergeordneten, sofort gespeicherten Zwischenspeichertabellen in den Status 'Festlegen der Integrität anstehend' versetzt.
 - Wenn die Tabelle inkrementell auf Integritätsverletzungen geprüft wird, werden ihre untergeordneten IMQs und Zwischenspeichertabellen in den Status 'Festlegen der Integrität anstehend' versetzt, und die ursprünglichen Status ihrer untergeordneten Fremdschlüsseltabellen bleiben unverändert erhalten.
 - Wenn die Tabelle keine Prüfung erfordert, bleiben die ursprünglichen Status ihrer untergeordneten IMQs, untergeordneten Zwischenspeichertabellen und untergeordneten Fremdschlüsseltabellen unverändert erhalten.
- Für eine Tabelle, die mit der Option CASCADE DEFERRED (in der Anweisung SET INTEGRITY oder im Befehl LOAD) in den Status 'Festlegen der Integrität anstehend' versetzt wurde, und deren Status 'Festlegen der Integrität anstehend' mit der Option IMMEDIATE UNCHECKED der Anweisung SET INTEGRITY aufgehoben wird, werden die zugehörigen untergeordneten Fremdschlüsseltabellen, untergeordneten IMQs und untergeordneten, sofort gespeicherten Zwischenspeichertabellen bei Bedarf wie folgt in den Status 'Festlegen der Integrität anstehend' versetzt:
 - Wenn die Tabelle im Modus REPLACE geladen wurde, werden ihre untergeordneten Fremdschlüsseltabellen, untergeordneten IMQs und untergeordneten, sofort gespeicherten Zwischenspeichertabellen in den Status 'Festlegen der Integrität anstehend' versetzt.
 - Wenn die Tabelle im Modus INSERT geladen wurde, werden ihre untergeordneten IMQs und Zwischenspeichertabellen in den Status 'Festlegen der Integrität anstehend' versetzt und die ursprünglichen Status ihrer untergeordneten Fremdschlüsseltabellen bleiben unverändert erhalten.
 - Wenn die Tabelle nicht geladen wurde, bleiben die ursprünglichen Status ihrer untergeordneten IMQs, untergeordneten Zwischenspeichertabellen und untergeordneten Fremdschlüsseltabellen unverändert erhalten.
- Die Ausführung der Anweisung SET INTEGRITY dauert in der Regel lange. Um das Risiko eines Rollbacks der gesamten Anweisung aufgrund einer Überschrei-

tung der Sperrzeit zu reduzieren, können Sie die Anweisung SET CURRENT LOCK TIMEOUT mit der Option WAIT absetzen, bevor die Anweisung SET INTEGRITY ausgeführt wird, und nach der Festschreibung der Transaktion das Sonderregister auf seinen vorherigen Wert zurücksetzen. Dabei ist jedoch zu beachten, dass sich das Sonderregister CURRENT LOCK TIMEOUT nur auf eine bestimmte Gruppe von Sperrtypen auswirkt.

- Stellen Sie bei Verwendung der Option ALLOW QUERY OPTIMIZATION USING REFRESH DEFERRED TABLES WITH REFRESH AGE ANY sicher, dass die Reihenfolge der Verwaltung für MQTs mit REFRESH DEFERRED korrekt ist. Angenommen, Sie verfügen über zwei MQTs (MQT1 und MQT2), deren gespeicherte Abfragen dieselben zu Grunde liegenden Tabellen gemeinsam nutzen. Die gespeicherte Abfrage für MQT2 kann mithilfe von MQT1 (anstelle der zu Grunde liegenden Tabellen) berechnet werden. Werden diese beiden MQTs durch separate Anweisungen verwaltet, und wird MQT2 zuerst verwaltet, entscheidet sich das System möglicherweise dafür, den Inhalt von MQT1 (der noch nicht verwaltet wurde) zum Verwalten von MQT2 zu verwenden. In diesem Fall enthält MQT1 aktuelle Daten, aber MQT2 enthält möglicherweise veraltete Daten, obwohl beide fast zur gleichen Zeit verwaltet wurden. Bei Verwendung von zwei SET INTEGRITY-Anweisungen anstelle von einer besteht die korrekte Verwaltungsreihenfolge darin, MQT1 zuerst zu verwalten.
- Bei Verwendung der Anweisung SET INTEGRITY zum Ausführen der Integritätsverarbeitung für eine Basistabelle, die geladen oder angefügt wurde, wird empfohlen, die abhängigen, als REFRESH IMMEDIATE deklarierten MQTs und die zugehörigen, als PROPAGATE IMMEDIATE deklarierten Zwischenspeichertabellen in derselben SET INTEGRITY-Anweisung zu verarbeiten. Dadurch wird vermieden, dass diese abhängigen Tabellen nach Beendigung der SET INTEGRITY-Verarbeitung in den Status 'Festlegen der Integrität anstehend, kein Zugriff' versetzt werden. Beachten Sie, dass es bei Basistabellen mit einer großen Anzahl abhängiger, als REFRESH IMMEDIATE deklarierten MQTs und als PROPAGATE IMMEDIATE deklarierten Zwischenspeichertabellen möglicherweise aufgrund von Speicherbegrenzungen nicht möglich ist, alle abhängigen Tabellen in derselben Anweisung wie die Basistabelle zu verarbeiten.
- Wenn die Option FORCE GENERATED oder GENERATE IDENTITY angegeben ist, und die generierte Spalte Teil eines eindeutigen Indexes ist, gibt die Anweisung SET INTEGRITY einen Fehler (SQLSTATE 23505) zurück, und wird rückgängig gemacht, wenn doppelte Schlüssel in dem eindeutigen Index festgestellt werden. Dieser Fehler wird auch zurückgegeben, wenn für die zu verarbeitende Tabelle eine Ausnahmetabelle vorhanden ist.

Dieser Fall kann unter den folgenden Umständen eintreten:

- Die Anweisung SET INTEGRITY wird nach einem Befehl LOAD für die Tabelle ausgeführt, und während der Ladeoperation wird der Dateitypänderungswert GENERATEDOVERRIDE oder IDENTITYOVERRIDE angegeben. Um dies zu verhindern, sollten Sie den Dateitypänderungswert GENERATEDIGNORE oder GENERATEDMISSING anstelle von GENERATEDOVERRIDE verwenden und den Änderungswert IDENTITYIGNORE oder IDENTITYMISSING anstelle von IDENTITYOVERRIDE. Bei Verwendung dieser empfohlenen Änderungswerte ist während der Ausführung der SET INTEGRITY-Anweisung keine Verarbeitung einer Spalte mit GENERATED BY-Ausdruck oder einer Identitätsspalte erforderlich.
- Die SET INTEGRITY-Anweisung wird nach einer ALTER TABLE-Anweisung ausgeführt, die den Ausdruck einer Spalte mit GENERATED BY-Ausdruck ändert.

Gehen Sie wie folgt vor, um den Status 'Festlegen der Integrität anstehend' für eine Tabelle aufzuheben, wenn ein solches Szenario vorliegt:

- Verwenden Sie nicht die Option FORCE GENERATED oder GENERATE IDENTITY, um die Spaltenwerte neu zu generieren. Verwenden Sie stattdessen die Option IMMEDIATE CHECKED zusammen mit der Option FOR EXCEPTION, um alle Zeilen, die gegen den Ausdruck der generierten Spalte verstoßen, in eine Ausnahmetabelle zu versetzen. Fügen Sie die Zeilen anschließend aus der Ausnahmetabelle wieder in die Tabelle ein, um den korrekten Ausdruck zu generieren und die Prüfung auf eindeutige Schlüssel auszuführen. Auf diese Weise wird die erneute Verarbeitung der gesamten Tabelle vermieden, weil nur Zeilen erneut verarbeitet werden müssen, die gegen den generierten Spaltenausdruck verstoßen.
- Wenn die zu verarbeitende Tabelle angefügte Partitionen enthält, heben Sie die Zuordnung dieser Partitionen auf, bevor Sie die im vorherigen Listenelement beschriebenen Aktionen ausführen. Fügen Sie die Partitionen anschließend erneut an, und führen Sie eine SET INTEGRITY-Anweisung aus, um die Integritätsverarbeitung für die angefügten Partitionen separat vorzunehmen.
- Wird für die SET INTEGRITY-Anweisung eine geschützte Tabelle und eine Ausnahmetabelle angegeben, müssen alle nachfolgenden Tabellenkriterien erfüllt sein. Andernfalls wird ein Fehler zurückgegeben (SQLSTATE 428A5):
 - Die Tabellen müssen durch dieselbe Sicherheitsrichtlinie geschützt werden.
 - Hat eine Spalte in der geschützten Tabelle den Datentyp DB2SECURITYLABEL, muss die entsprechende Spalte in der Ausnahmetabelle ebenfalls den Datentyp DB2SECURITYLABEL aufweisen.
 - Wird eine Spalte in der geschützten Tabelle durch einen Sicherheitskennsatz geschützt, muss die entsprechende Spalte in der Ausnahmetabelle durch denselben Sicherheitskennsatz geschützt werden.
- *Syntaxalternativen:* Die folgenden Alternativen werden aus Gründen der Kompatibilität mit früheren Versionen von DB2 und mit anderen Datenbankprodukten unterstützt. Diese Alternativen stellen keinen Standard dar und sollten daher nicht verwendet werden.
 - SET CONSTRAINTS kann anstelle von SET INTEGRITY angegeben werden
 - SUMMARY kann anstelle von MATERIALIZED QUERY angegeben werden

Beispiele

Beispiel 1: Die folgende Beispielabfrage liefert Informationen zum Status 'Festlegen der Integrität anstehend' und die Status für SET INTEGRITY-Zugriffsbeschränkungen. Mit SUBSTR werden einzelne Byte aus der Spalte CONST_CHECKED von SYSCAT.TABLES extrahiert. Das erste Byte steht für Fremdschlüsselintegritätsbedingungen, das zweite Byte steht für Prüfungen auf Integritätsbedingungen, das fünfte Byte steht für die Integrität der MQT, das sechste Byte steht für Integritätsbedingungen der generierten Spalten, das siebte Byte steht für die Integrität der Zwischenspeichertabelle und das achte Byte steht für Integritätsbedingungen der Datenpartitionierung. STATUS aktiviert den Status 'Festlegen der Integrität anstehend' und ACCESS_MODE aktiviert den SET INTEGRITY-Status für Zugriffsbeschränkung.

```
SELECT TABNAME, STATUS, ACCESS_MODE,
       SUBSTR(CONST_CHECKED,1,1) AS FK_CHECKED,
       SUBSTR(CONST_CHECKED,2,1) AS CC_CHECKED,
       SUBSTR(CONST_CHECKED,5,1) AS MQT_CHECKED,
       SUBSTR(CONST_CHECKED,6,1) AS GC_CHECKED,
       SUBSTR(CONST_CHECKED,7,1) AS STG_CHECKED,
       SUBSTR(CONST_CHECKED,8,1) AS DP_CHECKED
FROM SYSCAT.TABLES
```

Beispiel 2: Die übergeordnete in den Status 'Festlegen der Integrität anstehend, kein Zugriff' versetzen, und den Status 'Festlegen der Integrität anstehend' direkt an die untergeordneten Elemente weitergeben.

```
SET INTEGRITY FOR PARENT OFF  
NO ACCESS CASCADE IMMEDIATE
```

Beispiel 3: Die übergeordnete Tabelle in den Status 'Festlegen der Integrität anstehend, Lesezugriff' versetzen, ohne den Status 'Festlegen der Integrität anstehend' direkt an die untergeordneten Elemente weiterzugeben.

```
SET INTEGRITY FOR PARENT OFF  
READ ACCESS CASCADE DEFERRED
```

Beispiel 4: Integritätsprüfung für eine Tabelle mit dem Namen FACT_TABLE. Wenn keine Integritätsverletzungen festgestellt werden, wird der Status 'Festlegen der Integrität anstehend' für die Tabelle aufgehoben. Werden Integritätsverletzungen festgestellt, wird die gesamte Anweisung rückgängig gemacht, und die Tabelle bleibt im Status 'Festlegen der Integrität anstehend'.

```
SET INTEGRITY FOR FACT_TABLE IMMEDIATE CHECKED
```

Beispiel 5: Integrität der Tabellen SALES und PRODUCTS prüfen und Zeilen mit Integritätsverletzungen in Ausnahmetabellen mit den Namen SALES_EXCEPTIONS und PRODUCTS_EXCEPTIONS versetzen. Für die beiden Tabellen SALES und PRODUCTS wird der Status 'Festlegen der Integrität anstehend' aufgehoben (unabhängig davon, ob Integritätsverletzungen vorliegen).

```
SET INTEGRITY FOR SALES, PRODUCTS IMMEDIATE CHECKED  
FOR EXCEPTION IN SALES USE SALES_EXCEPTIONS,  
IN PRODUCTS USE PRODUCTS_EXCEPTIONS
```

Beispiel 6: Die Prüfung auf Integritätsbedingung über Fremdschlüssel in der Tabelle MANAGER aktivieren und die Integritätsprüfung über CHECK in der Tabelle EMPLOYEE, sowie die Umgehung dieser Prüfungen mit der Option IMMEDIATE UNCHECKED.

```
SET INTEGRITY FOR MANAGER FOREIGN KEY,  
EMPLOYEE CHECK IMMEDIATE UNCHECKED
```

Beispiel 7: Mit zwei ALTER TABLE-Anweisungen eine Prüfung auf Integritätsbedingung und einen Fremdschlüssel zur Tabelle EMP_ACT hinzufügen. Durch die Anweisung SET INTEGRITY mit der Option OFF wird die Tabelle in den Status 'Festlegen der Integrität anstehend' versetzt, damit die Integritätsbedingungen nicht unmittelbar bei der Ausführung der beiden ALTER TABLE-Anweisungen geprüft werden. Mit einer Anweisung SET INTEGRITY mit der Option IMMEDIATE CHECKED werden beide hinzugefügten Integritätsbedingungen in einem einzigen Durchlauf der Tabelle geprüft.

```
SET INTEGRITY FOR EMP_ACT OFF;  
ALTER TABLE EMP_ACT ADD CHECK  
(EMSTDATE <= EMENDATE);  
ALTER TABLE EMP_ACT ADD FOREIGN KEY  
(EMPNO) REFERENCES EMPLOYEE;  
SET INTEGRITY FOR EMP_ACT IMMEDIATE CHECKED  
FOR EXCEPTION IN EMP_ACT USE EMP_ACT_EXCEPTIONS
```

Beispiel 8: Generierte Spalten mit den korrekten Werten aktualisieren.

```
SET INTEGRITY FOR SALES IMMEDIATE CHECKED  
FORCE GENERATED
```

Beispiel 9: Anfügen (mit LOAD INSERT) aus verschiedenen Quellen an eine zu Grunde liegende Tabelle (SALES) einer als REFRESH IMMEDIATE deklarierten

MQT (SALES_SUMMARY). Inkrementelle Prüfung der Datenintegrität der Tabelle SALES und inkrementelle Aktualisierung der Tabelle SALES_SUMMARY. In diesem Szenario erfolgt die Integritätsprüfung für SALES und die Aktualisierung von SALES_SUMMARY inkrementell, weil das System die inkrementelle Verarbeitung auswählt. Für die Tabelle SALES wird die Option ALLOW READ ACCESS verwendet, um gleichzeitig ablaufende Leseoperationen für die vorhandenen Daten zu ermöglichen, während die Integritätsprüfung des geladenen Teils der Tabelle ausgeführt wird.

```
LOAD FROM 2000_DATA.DEL OF DEL
INSERT INTO SALES ALLOW READ ACCESS;
LOAD FROM 2001_DATA.DEL OF DEL
INSERT INTO SALES ALLOW READ ACCESS;
SET INTEGRITY FOR SALES ALLOW READ ACCESS IMMEDIATE CHECKED
FOR EXCEPTION IN SALES USE SALES_EXCEPTIONS;
REFRESH TABLE SALES_SUMMARY;
```

Beispiel 10: Anfügen einer neuen Partition an eine Tabelle SALES mit partitionierten Daten. Inkrementelle Prüfung auf ungültige Integritätsbedingungen in den angefügten Daten der Tabelle SALES und inkrementelle Aktualisierung der abhängigen Tabelle SALES_SUMMARY. Die Option ALLOW WRITE ACCESS wird für beide Tabellen verwendet, um gleichzeitig ablaufende Aktualisierungen zuzulassen, während die Integritätsprüfung ausgeführt wird.

```
ALTER TABLE SALES
ATTACH PARTITION STARTING (100) ENDING (200)
FROM SOURCE;
SET INTEGRITY FOR SALES ALLOW WRITE ACCESS, SALES_SUMMARY ALLOW WRITE ACCESS
IMMEDIATE CHECKED FOR EXCEPTION IN SALES
USE SALES_EXCEPTIONS;
```

Beispiel 11: Aufheben der Zuordnung einer Partition zu einer partitionierten Datentabelle SALES. Inkrementelle Aktualisierung der abhängigen Tabelle SALES_SUMMARY.

```
ALTER TABLE SALES
DETACH PARTITION 2000_PART INTO ARCHIVE_TABLE;
SET INTEGRITY FOR SALES_SUMMARY
IMMEDIATE CHECKED;
```

Beispiel 12: Aufheben des Status 'Festlegen der Integrität anstehend' für eine neue, vom Benutzer verwaltete MQT.

```
CREATE TABLE YEARLY_SALES
AS (SELECT YEAR, SUM(SALES)AS SALES
FROM FACT_TABLE GROUP BY YEAR)
DATA INITIALLY DEFERRED REFRESH DEFERRED MAINTAINED BY USER

SET INTEGRITY FOR YEARLY_SALES
ALL IMMEDIATE UNCHECKED
```

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.

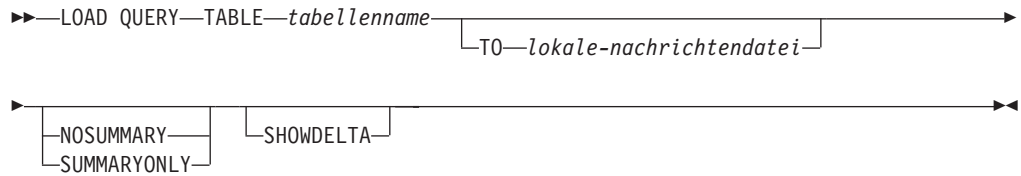
Berechtigung

Keine

Erforderliche Verbindung

Datenbank

Befehlssyntax



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 Tabellename 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 die *nachrichtendatei* sein, die für den Befehl **LOAD** angegeben wurde. Wenn die Datei bereits vorhanden ist, werden die Informationen vom Dienstprogramm LOAD an sie angefügt.

Beispiele

Ein Benutzer, der ein umfangreiches Datenvolumen in die Tabelle STAFF der Datenbank BILLYBOB lädt, möchte den Status der Ladeoperation prüfen. Der Benutzer kann Folgendes angeben:

```
db2 connect to billybob
db2 load query table staff to /u/mydir/staff.tempmsg
```

Die Ausgabedatei */u/mydir/staff.tempmsg* könnte folgendermaßen aussehen:

```
SQL3501W Der oder die Tabellenbereich, die die Tabelle enthalten,
werden nicht in den Status 'Backup anstehend' versetzt,
da die aktualisierende Recovery 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

```

Hinweise zur Verwendung

Neben Sperrern verwendet das Dienstprogramm LOAD Statuswerte für Tabellen, um den Zugriff auf die Tabelle einzuschränken. Mit dem Befehl **LOAD QUERY** kann der Tabellenstatus ermittelt werden. Der Befehl **LOAD QUERY** kann auch für Tabellen verwendet werden, die momentan nicht geladen werden. Bei einer partitionierten Tabelle handelt es sich bei dem gemeldeten Status für die entsprechenden sichtbaren Datenpartitionen um den Status mit den meisten Einschränkungen. Wenn sich beispielsweise eine Datenpartition im Status "Lesezugriff" befindet und alle anderen Datenpartitionen im Status "Normal", gibt die Operation **LOAD QUERY** den Status "Lesezugriff" zurück. Bei einer Ladeoperation wird keine Untergruppe von Datenpartitionen in einem Status belassen, der sich von dem der restlichen Tabelle unterscheidet. Folgende Tabellenstatus werden von **LOAD QUERY** beschrieben:

Normal

Eine Tabelle befindet sich im Status "Normal", wenn sie keinen anderen (abnormalen) Tabellenstatus aufweist. ++Der Status "Normal" ist der Anfangsstatus einer Tabelle nach ihrer Erstellung.

Festlegen der Integrität anstehend

Für die Tabelle gelten Integritätsbedingungen, die noch nicht überprüft wurden. Mit der Anweisung **SET INTEGRITY** können Sie den Status "Festlegen der Integrität anstehend" für die Tabelle aufheben. Das Dienstpro-

gramm LOAD versetzt eine Tabelle in den Status "Festlegen der Integrität anstehend", sobald eine Ladeoperation für eine Tabelle mit Integritätsbedingungen beginnt.

Laden läuft

Dies ist ein vorübergehender Status, der nur während einer Ladeoperation wirksam ist.

Laden anstehend

Für diese Tabelle war eine Ladeoperation aktiv, die jedoch abgebrochen wurde, bevor die Daten festgeschrieben werden konnten. Setzen Sie einen Befehl **LOAD TERMINATE**, **LOAD RESTART** oder **LOAD REPLACE** ab, um die Tabelle aus diesem Status herauszuholen.

Lesezugriff

Eine Tabelle befindet sich während einer Ladeoperation in diesem Status, wenn die Option **ALLOW READ ACCESS** angegeben wurde. "Lesezugriff" ist ein vorübergehender Status, der anderen Anwendungen und Dienstprogrammen Lesezugriff auf Daten gewährt, die schon vor der Ladeoperation vorhanden waren.

Reorganisation anstehend

Eine empfohlene Anweisung ALTER TABLE im Befehl **REORG** wurde für die Tabelle ausgeführt. Eine klassische **REORG**-Operation muss durchgeführt werden, bevor die Tabelle wieder zugänglich ist.

Nicht verfügbar

Die Tabelle ist nicht verfügbar. Die Tabelle kann lediglich gelöscht oder aus einem Backup wiederhergestellt werden. Eine Tabelle wird durch eine aktualisierende Recovery, 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". Setzen Sie einen Befehl **LOAD TERMINATE** oder **LOAD REPLACE** ab, um die Tabelle aus dem Status "Für Laden nicht neu startbar" herauszuholen. Eine Tabelle wird in diesen Status versetzt, wenn eine aktualisierende Recovery nach einer fehlgeschlagenen Ladeoperation ausgeführt wird, die nicht erfolgreich erneut gestartet oder beendet wurde, bzw. wenn eine Restore-Operation mit einem Online-Backup ausgeführt wird, das 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.

Unbekannt

Der Befehl **LOAD QUERY** kann den Tabellenstatus nicht ermitteln.

Derzeit werden mindestens 25 Tabellen- oder Tabellenbereichsstatus von dem IBM DB2-Datenbankprodukt unterstützt. Diese Status steuern den Zugriff auf Daten unter bestimmten Umständen, oder sie sondieren bei Bedarf bestimmte Benutzeraktionen, um die Integrität der Datenbank zu schützen. Sie resultieren meist aus Ereignissen im Zusammenhang mit den Operationen eines der DB2-Datenbankdienstprogramme (z. B. des Dienstprogramms LOAD, BACKUP oder RESTORE).

Obwohl abhängige Tabellenbereiche vor einer Ladeoperation nicht länger in den Quiescemodus versetzt werden (der Quiescemodus ist eine permanente Sperre), verhindert der Tabellenbereichsstatus "Laden läuft" das Backup abhängiger Tabellen während einer Ladeoperation. Der Tabellenbereichsstatus "Laden läuft" unterscheidet sich von dem Tabellenstatus "Laden läuft": Alle Ladeoperationen verwenden den Tabellenstatus "Laden läuft", aber Ladeoperationen für eine wiederherstellbare Datenbank mit angegebener Option **COPY NO** verwenden ebenfalls den Tabellenbereichsstatus "Laden läuft".

Die folgende Tabelle beschreibt die einzelnen unterstützten Tabellenstatus. Die Tabelle enthält außerdem praxisnahe Beispiele, die im Detail zeigen, wie Status interpretiert und behandelt werden können, die beim Verwalten Ihrer Datenbank auftreten können. Die Beispiele stammen aus Befehlsscripts, die unter AIX ausgeführt wurden. Sie können diese Scripts kopieren, einfügen und selbst ausführen. Wenn Sie das DB2-Datenbankprodukt auf einem System ausführen, das nicht auf UNIX basiert, stellen Sie sicher, dass alle Pfadnamen das richtige Format für Ihr System aufweisen. Die meisten Beispiele basieren auf Tabellen aus der Beispieldatenbank SAMPLE, die im Lieferumfang des DB2-Datenbankprodukts enthalten ist. Einige wenige Beispiele basieren auf Szenarios, die nicht Bestandteil der Beispieldatenbank SAMPLE sind, aber Sie können eine Verbindung zur Beispieldatenbank SAMPLE als Ausgangspunkt verwenden.

Tabelle 47. Unterstützte Tabellenstatus

Status	Beispiele
Laden anstehend	<p>Erstellen Sie für die vorgegebene Ladeeingabedatei <code>staffdata.del</code> mit einem großen Datenvolumen (z. B. 20.000 oder mehr Datensätze) einen kleinen Tabellenbereich, der die Zieltabelle der Ladeoperation (eine neue Tabelle mit dem Namen NEWSTAFF) enthält:</p> <pre>connect to sample; create tablespace ts1 managed by database using (file '/home/melnyk/melnyk/NODE0000 /SQL00001/ts1c1' 256); create table newstaff like staff in ts1; load from staffdata.del of del insert into newstaff; load query table newstaff; load from staffdata.del of del terminate into newstaff; load query table newstaff; connect reset;</pre> <p>Die vom Befehl LOAD QUERY zurückgegebenen Informationen zeigen, dass sich die Tabelle NEWSTAFF im Status "Laden anstehend" befindet. Nach einer LOAD TERMINATE-Operation befindet sich die Tabelle im Status "Normal".</p>
Laden läuft	<p>Führen Sie für die vorgegebene Ladeeingabedatei <code>staffdata.del</code> mit einem großen Datenvolumen (z. B. 20.000 oder mehr Datensätze) Folgendes aus:</p> <pre>connect to sample; create table newstaff like staff; load from staffdata.del of del insert into newstaff;</pre> <p>Führen Sie das folgende Script aus einer anderen Sitzung aus, während die Ladeoperation aktiv ist:</p> <pre>connect to sample; load query table newstaff; connect reset;</pre> <p>Die vom Befehl LOAD QUERY zurückgegebenen Informationen zeigen, dass sich die Tabelle NEWSTAFF im Status "Laden läuft" befindet.</p>

Tabelle 47. Unterstützte Tabellenstatus (Forts.)

Status	Beispiele
Normal	<pre>connect to sample; create table newstaff like staff; load query table newstaff;</pre> <p>Die vom Befehl LOAD QUERY zurückgegebenen Informationen zeigen, dass sich die Tabelle NEWSTAFF im Status "Normal" befindet.</p>
Für Laden nicht neu startbar	<pre>Führen Sie für die vorgegebene Ladeeingabedatei staffdata.del mit einem großen Datenvolumen (z. B. 20.000 oder mehr Datensätze) Folgendes aus: update db cfg for sample using logretain recovery; backup db sample; connect to sample; create tablespace ts1 managed by database using (file '/home/melnk/melnk/NODE0000 /SQL00001/ts1c1' 256); create table newstaff like staff in ts1; connect reset;backup db sample;</pre> <p>Die Zeitmarke für dieses Backup-Image ist: 20040629205935.</p> <pre>connect to sample; load from staffdata.del of del insert into newstaff copy yes to /home/melnk/backups; connect reset;restore db sample taken at 20040629205935; rollforward db sample to end of logs and stop; connect to sample; load query table newstaff; connect reset;</pre> <p>Die vom Befehl LOAD QUERY zurückgegebenen Informationen zeigen, dass sich die Tabelle NEWSTAFF im Status "Für Laden nicht neu startbar" und "Laden anstehend" befindet.</p> <pre>connect to sample; load from staffdata.del of del terminate into newstaff copy yes to /home/melnk/backups; load query table newstaff; connect reset;</pre> <p>Die vom Befehl LOAD QUERY zurückgegebenen Informationen zeigen, dass sich die Tabelle NEWSTAFF jetzt im Status "Normal" befindet.</p>
Lesezugriff	<pre>Führen Sie für die vorgegebene Ladeeingabedatei staffdata.del mit einem großen Datenvolumen (z. B. 20.000 oder mehr Datensätze) Folgendes aus: connect to sample; export to st_data.del of del select * from staff; create table newstaff like staff; import from st_data.del of del insert into newstaff; load from staffdata.del of del insert into newstaff allow read access;</pre> <p>Führen Sie das folgende Script aus einer anderen Sitzung aus, während die Ladeoperation aktiv ist:</p> <pre>connect to sample; load query table newstaff; select * from newstaff; connect reset;</pre> <p>Die vom Befehl LOAD QUERY zurückgegebenen Informationen zeigen, dass sich die Tabelle NEWSTAFF im Status "Lesezugriff" und "Laden anstehend" befindet. Die Abfrage gibt nur den exportierten Inhalt der Tabelle STAFF zurück (Daten, die vor der Ladeoperation in der Tabelle NEWSTAFF vorhanden waren).</p>

Tabelle 47. Unterstützte Tabellenstatus (Forts.)

Status	Beispiele
Festlegen der Integrität anstehend	<p>Vorgegeben ist eine Ladeeingabedatei <code>staff_data.del</code> mit folgendem Inhalt:</p> <pre>11,"Melnyk",20,"Sales",10,70000,15000: connect to sample; alter table staff add constraint max_salary check (100000 - salary > 0); load from staff_data.del of del insert into staff; load query table staff;</pre> <p>Die vom Befehl LOAD QUERY zurückgegebenen Informationen zeigen, dass sich die Tabelle STAFF im Status "Festlegen der Integrität anstehend" befindet.</p>
Nicht verfügbar	<p>Vorgegeben ist eine Ladeeingabedatei <code>staff_data.del</code> mit folgendem Inhalt:</p> <pre>11,"Melnyk",20,"Sales",10,70000,15000: update db cfg for sample using logretain recovery; backup db sample;</pre> <p>Die Zeitmarke für dieses Backup-Image ist: 20040629182012.</p> <pre>connect to sample; load from staff_data.del of del insert into staff nonrecoverable; connect reset;restore db sample taken at 20040629182012; rollforward db sample to end of logs and stop; connect to sample; load query table staff; connect reset;</pre> <p>Die vom Befehl LOAD QUERY zurückgegebenen Informationen zeigen, dass sich die Tabelle STAFF im Status "Nicht verfügbar" befindet.</p>

Der Fortschritt einer Ladeoperation kann auch mit dem Befehl **LIST UTILITIES** überwacht werden.

LIST TABLESPACES

Listet Tabellenbereiche und zugehörige Informationen für die aktuelle Datenbank auf.

Wichtig: Dieser Befehl bzw. diese API gilt als veraltet und wird möglicherweise in einem zukünftigen Release entfernt. Stattdessen können Sie die Tabellenfunktionen `MON_GET_TABLESPACE` und `MON_GET_CONTAINER` verwenden, die zusätzliche Informationen zurückgeben. Weitere Informationen hierzu finden Sie im Abschnitt „Die Befehle `LIST TABLESPACES` und `LIST TABLESPACE CONTAINERS` gelten als veraltet“ in der Veröffentlichung *Neuerungen in DB2 Version 9.7*.

Die in diesem Befehl angezeigten Informationen sind auch in der Tabellenbereichsmomentaufnahme verfügbar.

Geltungsbereich

Dieser Befehl gibt nur Informationen für die Datenbankpartition zurück, auf der er ausgeführt wird.

Berechtigung

Eine der folgenden Berechtigungen:

- SYSADM

- SYSCTRL
- SYSMANT
- SYSMON
- DBADM
- Berechtigung LOAD

Erforderliche Verbindung

Datenbank

Befehlssyntax

```
▶—LIST TABLESPACES—┐
                       └SHOW DETAIL┘▶
```

Befehlsparameter

SHOW DETAIL

Wenn diese Option nicht angegeben wird, werden nur die folgenden Basisinformationen zu jedem Tabellenbereich zur Verfügung gestellt:

- Tabellenbereichs-ID
- Name
- Typ (vom System verwalteter oder vom Datenbankmanager verwalteter Bereich)
- Inhalt (beliebige Daten, lange Daten bzw. Indexdaten oder temporäre Daten)
- Status - ein Hexadezimalwert, der den aktuellen Tabellenbereichsstatus angibt. Der extern angezeigte Status eines Tabellenbereichs besteht aus der Hexadezimalsumme bestimmter Statuswerte. Lautet der Status beispielsweise "Im Quiesce: EXCLUSIVE" und "Laden anstehend", ist der Wert 0x0004 + 0x0008, was 0x000c entspricht. Über den Befehl **db2tbst** zum Abrufen des Tabellenbereichsstatus erhalten Sie den Tabellenbereichsstatus, der einem bestimmten Hexadezimalwert zugeordnet ist. Die folgenden Bitdefinitionen sind in `sqlutil.h` aufgelistet:

0x0	Normal
0x1	Im Quiesce: SHARE
0x2	Im Quiesce: UPDATE
0x4	Im Quiesce: EXCLUSIVE
0x8	Laden anstehend
0x10	Löschen anstehend
0x20	Backup anstehend
0x40	Aktualisierende Recovery wird ausgeführt
0x80	Aktualisierende Recovery anstehend
0x100	Restore anstehend
0x100	Recovery anstehend (nicht verwendet)
0x200	Inaktivierung anstehend
0x400	Reorganisation wird ausgeführt
0x800	Backup wird ausgeführt
0x1000	Speicher muss definiert werden
0x2000	Restore wird ausgeführt
0x4000	Offline und kein Zugriff
0x8000	Löschen anstehend
0x10000	Schreiben aussetzen
0x20000	Laden läuft
0x2000000	Speicher kann definiert werden
0x4000000	Speicherdefinition ist abgeschlossen
0x8000000	Speicherdefinition vor aktualisierender Recovery geändert

```

0x10000000 DMS-Ausgleichsfunktion ist aktiv
0x20000000 Tabellenbereich wird gelöscht
0x40000000 Tabellenbereich wird erstellt

```

Anmerkung: DB2 LOAD versetzt Tabellenbereiche nicht in den Status Laden anstehend oder Löschen anstehend.

Wenn diese Option angegeben wird, werden die folgenden zusätzlichen Informationen zu jedem Tabellenbereich zur Verfügung gestellt:

- Gesamtanzahl Seiten
- Anzahl verwendbarer Seiten
- Anzahl verwendeter Seiten
- Anzahl freier Seiten
- Obere Grenze (Seiten)
- Seitengröße (in Byte)
- Speicherbereichsgröße (in Seiten)
- Vorablesezugriffsgröße (in Seiten)
- Anzahl Container
- Mindestrecoveryzeitpunkt (frühester Zeitpunkt, für den eine aktualisierende Recovery des Tabellenbereichs durchgeführt werden kann; Zeitmarke wird in koordinierter Weltzeit ausgedrückt und nur angezeigt, wenn nicht null)
- Tabellenbereichs-ID für Statusänderung (nur angezeigt, wenn der Tabellenbereichsstatus "Laden anstehend" oder "Löschen anstehend" ist)
- Objekt-ID für Statusänderung (nur angezeigt, wenn der Tabellenbereichsstatus "Laden anstehend" oder "Löschen anstehend" ist)
- Anzahl der Personen, die Quiesce durchführen (nur angezeigt, wenn der Tabellenbereichsstatus "Im Quiesce: SHARE", "Im Quiesce: UPDATE" oder "Im Quiesce: EXCLUSIVE" ist)
- Tabellenbereichs-ID und Objekt-ID für jede Person, die Quiesce durchführt (nur angezeigt, wenn die Anzahl Personen, die Quiesce durchführen größer als Null ist).

Beispiele

Die beiden folgenden Beispiele zeigen Ausgaben von **LIST TABLESPACES SHOW DETAIL**.

```

          Tabellenbereiche für aktuelle Datenbank
Tabellenbereichs-ID          = 0
Name                         = SYSCATSPACE
Typ                          = Von der Datenbank verwalteter Bereich
Inhalt                       = Beliebige Daten
Status                       = 0x0000
  Detaillierte Erläuterung:
    Normal
Gesamtanzahl Seiten          = 895
Verwendbare Seiten           = 895
Verwendete Seiten            = 895
Freie Seiten                  = Nicht zutreffend
Obere Grenze (Seiten)        = Nicht zutreffend
Seitengröße (Byte)           = 4096
Speicherbereichsgröße (Seiten) = 32
Vorablesezugriffsgröße (Seiten) = 32
Anzahl Container              = 1

```

```

Tabellenbereichs-ID           = 1
Name                           = TEMPSPACE1
Typ                             = Vom System verwalteter Bereich
Inhalt                          = Temporäre Daten
Status                          = 0x0000
    Detaillierte Erläuterung:
        Normal
Gesamtanzahl Seiten             = 1
Verwendbare Seiten              = 1
Verwendete Seiten               = 1
Freie Seiten                     = Nicht zutreffend
Obere Grenze (Seiten)           = Nicht zutreffend
Seitengröße (Byte)              = 4096
Speicherbereichsgröße (Seiten) = 32
Vorabesezugriffsgröße (Seiten) = 32
Anzahl Container                 = 1

    Tabellenbereichs-ID           = 2
Name                           = USERSPACE1
Typ                             = Von der Datenbank verwalteter Bereich
Inhalt                          = Beliebige Daten
Status                          = 0x000c
    Detaillierte Erläuterung:
        Im Quiesce: EXCLUSIVE    Laden anstehend
Gesamtanzahl Seiten             = 337
Verwendbare Seiten              = 337
Verwendete Seiten               = 337
Freie Seiten                     = Nicht zutreffend
Obere Grenze (Seiten)           = Nicht zutreffend
Seitengröße (Byte)              = 4096
Speicherbereichsgröße (Seiten) = 32
Vorabesezugriffsgröße (Seiten) = 32
Anzahl Container                 = 1
Statusänderung: Tabellenbereichs-ID = 2
Statusänderung: Objekt-ID        = 3
Anz. Personen, die Quiesce durchf. = 1
    Person 1, die Quiesce durchführt:
        Tabellenbereichs-ID       = 2
        Objekt-ID                  = 3

```

DB21011I In einer Umgebung mit partitionierten Datenbankservern werden nur die Tabellenbereiche auf dem aktuellen Knoten aufgelistet.

```

    Tabellenbereiche für aktuelle Datenbank
Tabellenbereichs-ID           = 0
Name                           = SYSCATSPACE
Typ                             = Vom System verwalteter Bereich
Inhalt                          = Beliebige Daten
Status                          = 0x0000
    Detaillierte Erläuterung:
        Normal
Gesamtanzahl Seiten             = 1200
Verwendbare Seiten              = 1200
Verwendete Seiten               = 1200
Freie Seiten                     = Nicht zutreffend
Obere Grenze (Seiten)           = Nicht zutreffend
Seitengröße (Byte)              = 4096
Speicherbereichsgröße (Seiten) = 32
Vorabesezugriffsgröße (Seiten) = 32
Anzahl Container                 = 1

Tabellenbereichs-ID           = 1
Name                           = TEMPSPACE1
Typ                             = Vom System verwalteter Bereich
Inhalt                          = Temporäre Daten
Status                          = 0x0000
    Detaillierte Erläuterung:
        Normal

```

```

Gesamtanzahl Seiten           = 1
Verwendbare Seiten           = 1
Verwendete Seiten            = 1
Freie Seiten                  = Nicht zutreffend
Obere Grenze (Seiten)        = Nicht zutreffend
Seitengröße (Byte)           = 4096
Speicherbereichsgröße (Seiten) = 32
Vorablesezugriffsgröße (Seiten) = 32
Anzahl Container              = 1

Tabellenbereichs-ID          = 2
Name                          = USERSPACE1
Typ                            = Vom System verwalteter Bereich
Inhalt                         = Beliebige Daten
Status                         = 0x0000
    Detaillierte Erläuterung:
        Normal
Gesamtanzahl Seiten           = 1
Verwendbare Seiten           = 1
Verwendete Seiten            = 1
Freie Seiten                  = Nicht zutreffend
Obere Grenze (Seiten)        = Nicht zutreffend
Seitengröße (Byte)           = 4096
Speicherbereichsgröße (Seiten) = 32
Vorablesezugriffsgröße (Seiten) = 32
Anzahl Container              = 1

Tabellenbereichs-ID          = 3
Name                          = DMS8K
Typ                            = Von der Datenbank verwalteter Bereich
Inhalt                         = Beliebige Daten
Status                         = 0x0000
    Detaillierte Erläuterung:
        Normal
Gesamtanzahl Seiten           = 2000
Verwendbare Seiten           = 1952
Verwendete Seiten            = 96
Freie Seiten                  = 1856
Obere Grenze (Seiten)        = 96
Seitengröße (Byte)           = 8192
Speicherbereichsgröße (Seiten) = 32
Vorablesezugriffsgröße (Seiten) = 32
Anzahl Container              = 2

Tabellenbereichs-ID          = 4
Name                          = TEMP8K
Typ                            = Vom System verwalteter Bereich
Inhalt                         = Temporäre Daten
Status                         = 0x0000
    Detaillierte Erläuterung:
        Normal
Gesamtanzahl Seiten           = 1
Verwendbare Seiten           = 1
Verwendete Seiten            = 1
Freie Seiten                  = Nicht zutreffend
Obere Grenze (Seiten)        = Nicht zutreffend
Seitengröße (Byte)           = 8192
Speicherbereichsgröße (Seiten) = 32
Vorablesezugriffsgröße (Seiten) = 32
Anzahl Container              = 1
DB21011I In einer Umgebung mit partitionierten Datenbankservern werden nur die
Tabellenbereiche auf dem aktuellen Knoten aufgelistet.

```

Hinweise zur Verwendung

In einer Umgebung mit partitionierten Datenbanken gibt dieser Befehl nicht alle Tabellenbereiche in der Datenbank zurück. Fragen Sie `SYSCAT.TABLESPACES` ab, um eine Liste aller Tabellenbereiche abzurufen.

Wenn der Befehl `LIST TABLESPACES SHOW DETAIL` abgesetzt wird, versucht er, alle anstehenden freien Speicherbereiche im Tabellenbereich freizugeben. Werden die anstehenden freien Speicherbereiche erfolgreich freigegeben, wird ein Datensatz protokolliert.

Während des Neuausgleichs eines Tabellenbereichs schließt die Anzahl der verwendbaren Seiten auch Seiten für den neu hinzugefügten Container ein. Diese neuen Seiten werden jedoch erst nach Fertigstellung des Neuausgleichs in der Anzahl der freien Seiten berücksichtigt. Findet kein Neuausgleich des Tabellenbereichs statt, entspricht die Anzahl der verwendbaren Seiten der Anzahl der verwendeten Seiten plus der Anzahl der freien Seiten.

Weitere Optionen beim Versetzen von Daten

Versetzen von Tabellen im Online-Modus mithilfe der Prozedur `ADMIN_MOVE_TABLE`

Mithilfe der Prozedur `ADMIN_MOVE_TABLE` können Sie Tabellen online oder offline versetzen. Versetzen Sie Tabellen nicht im Offline-Modus, sondern im Online-Modus, wenn Verfügbarkeit bei Ihnen einen höheren Stellenwert hat als Kosten, Speicherplatz, Leistung beim Versetzen und Transaktionsaufwand.

Vorbereitende Schritte

Stellen Sie sicher, dass der Plattenspeicherplatz für die Kopien von Tabelle und Index, die Zwischenspeichertabelle und zusätzliche Protokolleinträge ausreicht.

Informationen zu diesem Vorgang

Sie können eine Tabelle online versetzen, indem Sie die gespeicherte Prozedur einmalig oder mehrmals (für jede einzelne von der Prozedur auszuführende Operation) aufrufen. Ein mehrmaliges Aufrufen eröffnet Ihnen zusätzliche Möglichkeiten. So können Sie in diesem Fall z. B. das Versetzen abrechnen oder steuern, wann die Zieltabelle für eine Aktualisierung offline genommen wird.

Wenn Sie die Prozedur `SYSPROC.ADMIN_MOVE_TABLE` aufrufen, wird eine Spiegelkopie der Quellentabelle erstellt. Während der Kopierphase werden Änderungen (Aktualisierungen, Einfügungen und Löschungen) an der Quellentabelle mit Triggern erfasst und in eine Zwischenspeichertabelle gestellt. Im Anschluss an die Kopierphase werden die in der Zwischenspeichertabelle erfassten Änderungen auf die Spiegelkopie angewendet. Anschließend wird die Quellentabelle von der gespeicherten Prozedur für eine kurze Zeit offline genommen, um der Spiegelkopie und den zugehörigen Indizes den Tabellennamen und die Indexnamen der Quellentabelle zuzuordnen. Die Spiegeltabelle wird anschließend erneut online genommen und tritt an die Stelle der Quellentabelle. Die Quellentabelle wird standardmäßig gelöscht, Sie können jedoch bei Bedarf mithilfe der Option `KEEP` sicherstellen, dass sie unter einem anderen Namen beibehalten wird.

Vermeiden Sie es, Tabellen ohne Indizes, insbesondere ohne eindeutige Indizes, online zu versetzen. Wenn Sie eine Tabelle ohne einen eindeutigen Index online versetzen, kann dies zu Deadlocks führen und eine komplizierte bzw. kostenintensive Wiederholung von Operationen erforderlich machen.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Tabelle online zu versetzen:

1. Rufen Sie die Prozedur ADMIN_MOVE_TABLE auf eine der folgenden Arten auf:
 - Rufen Sie die Prozedur ADMIN_MOVE_TABLE einmalig auf und geben Sie dabei zumindest den Schemanamen der Quellentabelle, den Namen der Quellentabelle und den Operationstyp MOVE an. Verwenden Sie beispielsweise die folgende Syntax, um die Daten in eine vorhandene Tabelle in demselben Tabellenbereich zu versetzen:

```
CALL SYSPROC.ADMIN_MOVE_TABLE (  
  'schemaname',  
  'quellentabelle',  
  '',  
  '',  
  '',  
  '',  
  '',  
  '',  
  '',  
  '',  
  '',  
  '',  
  'MOVE')
```
 - Rufen Sie die Prozedur ADMIN_MOVE_TABLE mehrmals auf (einmal für jede Operation) und geben Sie dabei zumindest den Schemanamen der Quellentabelle, den Namen der Quellentabelle und einen Operationsnamen an. Verwenden Sie beispielsweise die folgende Syntax, um die Daten in eine neue Tabelle in demselben Tabellenbereich zu versetzen:

```
CALL SYSPROC.ADMIN_MOVE_TABLE (  
  'schemaname',  
  'quellentabelle',  
  '',  
  '',  
  '',  
  '',  
  '',  
  '',  
  '',  
  '',  
  'operationsname')
```

Dabei steht *operationsname* für die folgenden Werte: INIT, COPY, REPLAY, VERIFY oder SWAP. Sie müssen die Prozedur in dieser Operationsreihenfolge aufrufen. Beim ersten Aufruf muss z. B. INIT als Operationsname angegeben werden.

Anmerkung: Die Operation VERIFY ist kostenintensiv. Führen Sie diese Operation nur aus, wenn sie für das Versetzen der Tabelle benötigt wird.

2. Führen Sie die Prozedur erneut aus, wenn das Versetzen im Online-Modus fehlschlägt:
 - a. Beheben Sie den Fehler, der dazu führt, dass das Versetzen der Tabelle fehlschlägt.

- b. Bestimmen Sie die Phase, in der das Versetzen fehlgeschlagen ist, indem Sie den Prozedurstatus in der Protokolltabelle SYSTOOLS.ADMIN_MOVE_TABLE abfragen.
- c. Rufen Sie die gespeicherte Prozedur erneut auf und geben Sie dabei eine der folgenden Optionen an:
 - Verwenden Sie die Option INIT, wenn die Prozedur den Status INIT aufweist.
 - Verwenden Sie die Option COPY, wenn die Prozedur den Status COPY aufweist.
 - Verwenden Sie die Option REPLAY oder SWAP, wenn die Prozedur den Status REPLAY aufweist.
 - Verwenden Sie die Option CLEANUP, wenn die Prozedur den Status CLEANUP aufweist.

Wenn der Status beim Versetzen einer Tabelle im Online-Modus nicht COMPLETED oder CLEANUP lautet, können Sie das Versetzen abbrechen, indem Sie die Option CANCEL für die gespeicherte Prozedur angeben.

Beispiele

Beispiel 1: Versetzen Sie die Tabelle T1 im Schema SVALENTI in den Tabellenbereich ACCOUNTIN, ohne T1 dazu offline zu nehmen. Geben Sie die Tabellenbereiche DATA, INDEX und LONG an, um die Tabelle in einen neuen Tabellenbereich zu versetzen.

```
CALL SYSPROC.ADMIN_MOVE_TABLE(
'SVALENTI',
'T1',
'ACCOUNTING',
'ACCOUNTING',
'ACCOUNTING',
'',
'',
'',
'',
'',
'',
'MOVE')
```

Beispiel 2: Versetzen Sie die Tabelle T1 im Schema EBABANI in den Tabellenbereich ACCOUNTIN, ohne T1 dazu offline zu nehmen und behalten Sie eine Kopie der ursprünglichen Tabelle nach dem Versetzen. Verwenden Sie die Optionen COPY_USE_LOAD und LOAD_MSGPATH um den Pfad der Nachrichtendatei für Ladeoperationen festzulegen. Geben Sie die Tabellenbereiche DATA, INDEX und LONG an, um die Tabelle in einen neuen Tabellenbereich zu versetzen. Die ursprüngliche Tabelle bleibt mit einem Namen, der 'EBABANI' ähnlich ist, erhalten. T1AAAAXo'.

```
CALL SYSPROC.ADMIN_MOVE_TABLE(
'EBABANI',
'T1',
'ACCOUNTING',
'ACCOUNTING',
'ACCOUNTING',
'',
'',
'',
'',
'',
'KEEP, COPY_USE_LOAD,LOAD_MSGPATH "/home/ebabani"',
'MOVE')
```



```

MAXVALUE IN TS3 INDEX IN TS3 LONG IN TS3)',
',
',
',
'MOVE')"
```

Versetzen von Daten mit DB2 Connect

Vorbereitende Schritte

Wenn Sie in einer komplexen Umgebung arbeiten, in der Sie Daten zwischen einem Hostdatenbanksystem und einer Workstation versetzen müssen, können Sie DB2 Connect, das Gateway für die Datenübertragung zwischen Host und Workstation, verwenden (siehe Abb. 17).

Informationen zu diesem Vorgang

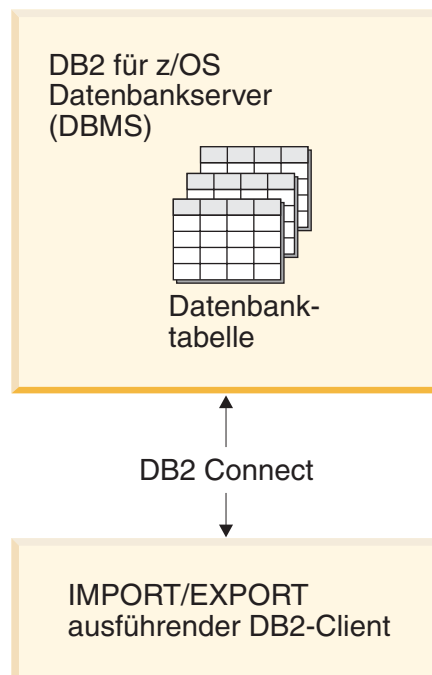


Abbildung 17. Import/Export über DB2 Connect

Mit den DB2-Dienstprogrammen IMPORT und EXPORT können Daten aus einer Datenbank auf einem IBM Mainframe-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 IBM Mainframe-Server in eine PC/IXF-Datei exportieren und diese Daten anschließend in eine DB2 Database for Linux, UNIX and Windows-Datenbank importieren.

Sie können Export- und Importfunktionen von einem Datenbankclient oder von der DB2 Connect-Workstation aus ausführen.

Anmerkung:

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 Abfragen verwendet werden. Geben Sie beim Dienstprogramm IMPORT den Änderungswert `compound` für den Dateityp an, um eine angegebene Anzahl von Abfrageanweisungen zu einem Block zusammenzufassen. Hierdurch können der Systemaufwand für das Netz verringert und die Antwortzeiten verbessert werden.

Bei DB2 Connect müssen Export- und Importoperationen die folgenden Bedingungen erfüllen:

- Der Dateityp muss PC/IXF sein.
- Es muss eine Zieltabelle auf dem Zielsystem 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, System z, IBM Power Systems, VM und VSE), markieren DBCS-Startzeichen und DBCS-Endzeichen 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

Gehen Sie wie folgt vor, um Daten in eine Datenbank auf einem Host- oder System i-Server zu versetzen:

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.

Gehen Sie wie folgt vor, um Daten aus einer Datenbank auf dem Host-Server auf eine Workstation zu versetzen:

Vorgehensweise

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

Beispiel

Das folgende Beispiel veranschaulicht, wie Sie Daten von einer Workstation in eine Datenbank auf einem Host- oder System i-Server versetzen.

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
```

Setzen Sie den folgenden Befehl ab, um eine DRDA-Verbindung zur DB2-Zieldatenbank herzustellen:

```
db2 connect to cbc664 user admin using xxx
```

Erstellen Sie die Zieltabelle auf der DB2-Zieldatenbankinstanz, wenn sie dort 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))
```

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.

Nächste Schritte

Ausführliche Informationen enthält das Handbuch "Moving Data Across the DB2 Family", eine Veröffentlichung aus der Reihe der IBM Redbooks. Diese Redbooks-Veröffentlichung steht unter der URL <http://www.redbooks.ibm.com/redbooks/SG246905> zur Verfügung.

IBM Replikationstools nach Komponenten geordnet

IBM bietet zwei primäre Replikationslösungen an: Q Replication und SQL Replication.

Die Hauptkomponenten von Q Replication sind das Q Capture-Programm und das Q Apply-Programm. Die Hauptkomponenten von SQL Replication sind das Capture-Programm und das Apply-Programm. Beide Replikationstypen nutzen gemeinsam den Replikationsalertmonitor. Sie können diese Replikationskomponenten mithilfe der Replikationszentrale und des Befehlszeilenprogramms ASNCLP konfigurieren und verwalten.

In der folgenden Liste werden diese Replikationskomponenten kurz zusammengefasst:

Q Capture-Programm

Sucht im DB2-Recoveryprotokoll nach Änderungen an DB2-Quellentabellen und setzt festgeschriebene Quellendaten in WebSphere MQ-Nachrichten um, die im XML-Format für eine subscribierende Anwendung veröffentlicht oder in einem komprimierten Format in das Q Apply-Programm repliziert werden können.

Q Apply-Programm

Ruft WebSphere MQ-Nachrichten aus einer Warteschlange ab, setzt die Nachrichten in SQL-Anweisungen um und aktualisiert eine Zieltabelle oder gespeicherte Prozedur. Zu den unterstützten Zielen gehören DB2-Datenbanken oder -Subsysteme sowie Informix und Microsoft SQL Server-Datenbanken, auf die über Kurznamen von Servern mit föderierten Datenbanken zugegriffen wird.

Capture-Programm

Sucht im DB2-Recoveryprotokoll nach Änderungen an registrierten Quellentabellen oder Sichten und speichert anschließend festgeschriebene Transaktionsdaten in relationalen Tabellen (so genannten CD-Tabellen) so lange zwischen, bis das Zielsystem bereit ist, diese Daten zu kopieren. SQL Replication stellt außerdem Capture-Trigger zur Verfügung, die eine Zwischenspeichertabelle (eine so genannte CCD-Tabelle) mit Datensätzen zu Änderungen an Nicht-DB2-Quellentabellen auffüllt.

Apply-Programm

Liest Daten aus Zwischenspeichertabellen und nimmt die erforderlichen Änderungen an Zieltabellen vor. Bei Nicht-DB2-Datenquellen liest das Apply-Programm die CCD-Tabelle mithilfe des Kurznamens dieser Tabelle in der föderierten Datenbank und nimmt die erforderlichen Änderungen an der Zieltabelle vor.

Replikationsalertmonitor

Ein Dienstprogramm, das überprüft, ob das Q Capture-, Q Apply-, Capture- und Apply-Programm ordnungsgemäß funktioniert. Es sucht nach Situationen, in denen ein Programm beendet wird, eine Warnung oder Fehlernachricht ausgibt, einen bestimmten Grenzwert erreicht oder eine bestimmte Aktion ausführt, und gibt anschließend Benachrichtigungen an einen E-Mail-Server, einen Pager oder die z/OS-Konsole aus.

Verwenden Sie die Replikationszentrale für folgende Zwecke:

- Definieren von Registrierungen, Subskriptionen, Veröffentlichungen, Warteschlangenmasken, Alertbedingungen und anderen Objekten
- Starten, Stoppen, Aussetzen, Wiederaufnehmen und Reinitialisieren des Replikationsprogramms
- Festlegen des zeitlichen Ablaufs der automatisierten Kopiervorgänge
- Angeben von SQL-Erweiterungen für die Daten
- Definieren von Beziehungen zwischen den Quellen- und Zieltabellen

Kopieren von Schemata

Das Dienstprogramm **db2move** und die Prozedur **ADMIN_COPY_SCHEMA** ermöglichen Ihnen die rasche Erstellung von Kopien eines Datenbankschemas. Nach der Einrichtung eines Modellschemas können Sie dieses als Vorlage für die Erstellung neuer Versionen verwenden.

Informationen zu diesem Vorgang

Zum Kopieren eines einzelnen Schemas innerhalb derselben Datenbank können Sie die Prozedur **ADMIN_COPY_SCHEMA** verwenden. Zum Kopieren eines oder mehrerer Schemata von einer Quellen- in eine Zieldatenbank können Sie das Dienstprogramm **db2move** mit der Aktion **COPY -co** verwenden. Die meisten Datenbankobjek-

te aus dem Quellschema werden unter das neue Schema in der Zieldatenbank kopiert.

Tipps zur Fehlerbehebung

Sowohl die Prozedur **ADMIN_COPY_SCHEMA** als auch das Dienstprogramm **db2move** rufen den Befehl **LOAD** auf. Während der Verarbeitung des Ladevorgangs werden die Tabellenbereiche, in denen sich die Zieldatenbankobjekte befinden, in den Status 'Backup anstehend' versetzt.

Prozedur ADMIN_COPY_SCHEMA

Durch die Verwendung dieser Prozedur mit der Option **COPYNO** werden die Tabellenbereiche, in denen sich das Zielobjekt befindet, in den Status 'Backup anstehend' versetzt, wie in der obigen Anmerkungen beschrieben. Um den Tabellenbereich aus dem Status 'Festlegen der Integrität anstehend' herauszunehmen, setzt diese Prozedur eine Anweisung **SET INTEGRITY** ab. In Situationen, in denen ein Zieltabellenobjekt definierte referenzielle Integritätsbedingungen besitzt, wird die Zieltabelle ebenfalls in den Status 'Festlegen der Integrität anstehend' versetzt. Da sich die Tabellenbereiche bereits im Status 'Backup anstehend' befinden, schlägt die Ausführung der Anweisung **SET INTEGRITY** der Prozedur **ADMIN_COPY_SCHEMA** fehl.

Zur Behebung dieses Problems müssen Sie einen Befehl **BACKUP DATABASE** absetzen, um den Status 'Backup anstehend' für die betroffenen Tabellenbereiche aufzuheben. Als Nächstes müssen Sie die Spalte **Statement_text** der Fehlertabelle prüfen, die von dieser Prozedur generiert wurde, um eine Liste der Tabellen im Status 'Festlegen der Integrität anstehend' zu finden. Führen Sie anschließend die Anweisung **SET INTEGRITY** für jede in der Liste aufgeführte Tabelle aus, um jede einzelne Tabelle aus dem Status 'Festlegen der Integrität anstehend' herauszunehmen.

Dienstprogramm 'db2move'

Dieses Dienstprogramm versucht, alle zulässigen Schemaobjekte mit Ausnahme der folgenden Typen zu kopieren:

- Tabellenhierarchie
- Zwischenspeichertabellen (vom Dienstprogramm **LOAD** in Umgebungen mit mehreren Datenbankpartitionen nicht unterstützt)
- JAR-Dateien (Java Routine Archives)
- Kurznamen
- Pakete
- Sichthierarchien
- Objektzugriffsrechte (Alle neuen Objekte werden mit Standardberechtigungen erstellt.)
- Statistiken (Neue Objekte enthalten keine statistischen Informationen.)
- Indexerweiterungen (in Verbindung mit benutzerdefinierten strukturierten Typen)
- Benutzerdefinierte strukturierte Typen und ihre Umsetzungsfunktionen

Fehler aufgrund nicht unterstützter Typen

Wenn ein Objekt eines der nicht unterstützten Typen im Quellschema angetroffen wird, wird ein Eintrag in einer Fehlerdatei protokolliert, der darauf hinweist, dass ein nicht unterstützter Objekttyp gefunden wurde. Die **COPY**-Operation wird trotzdem erfolgreich ausgeführt. Der protokollierte Eintrag soll Sie über Objekte informieren, die durch diese Operation nicht kopiert wurden.

Nicht mit Schemata gekoppelte Objekte

Objekte, die nicht mit einem Schema gekoppelt sind, wie zum Beispiel Tabellenbereiche und Ereignismonitore, werden bei einer Schemakopieroperation nicht berücksichtigt. Sie müssen sie in der Zieldatenbank vor dem Aufrufen der Schemakopieroperation erstellen.

Replizierte Tabellen

Wenn eine replizierte Tabelle kopiert wird, ist die neue Kopie der Tabelle nicht für die Replikation eingerichtet. Die Tabelle wird als reguläre Tabelle erneut erstellt.

Unterschiedliche Instanzen

Die Quelldatenbank muss katalogisiert werden, wenn sie sich nicht in derselben Instanz wie die Zieldatenbank befindet.

Option SCHEMA_MAP

Wenn die Option SCHEMA_MAP zur Angabe eines anderen Schemanamens in der Zieldatenbank verwendet wird, führt die Schemakopieroperation nur eine minimale Analyse der Objektdefinitionsanweisungen durch, um den ursprünglichen Schemanamen durch den neuen Schemanamen zu ersetzen. Zum Beispiel werden alle Vorkommen des ursprünglichen Schemanamens, die im Inhalt einer SQL-Prozedur auftreten, nicht durch den neuen Schemanamen ersetzt. Aus diesem Grund kann die Erstellung dieser Objekte durch die Schemakopieroperation fehlschlagen. Weitere mögliche Beispiele: Zwischenspeichertabelle, Ergebnistabelle, MQT (Materialized Query Table). Sie können die Anweisungen der Datendefinitionssprache (DDL) in der Fehlerdatei verwenden, um diese nicht erstellten Objekte nach Abschluss der Kopieroperation erneut zu erstellen.

Gegenseitige Abhängigkeiten zwischen Objekten

Die Schemakopieroperation versucht, Objekte in einer Reihenfolge erneut zu erstellen, die die gegenseitigen Abhängigkeiten zwischen diesen Objekten berücksichtigt. Wenn zum Beispiel eine Tabelle T1 eine Spalte enthält, die eine benutzerdefinierte Funktion U1 referenziert, erstellt die Operation die Funktion U1 erneut, bevor sie die Tabelle T1 erneut erstellt. Informationen zu Abhängigkeiten von Prozeduren sind jedoch in den Katalogen nicht einfach verfügbar. Bei der erneuten Erstellung von Prozeduren versucht die Schemakopieroperation daher zunächst, alle Prozeduren erneut zu stellen. Anschließend wiederholt sie den Versuch für die Prozeduren, deren erste Neuerstellung fehlgeschlagen ist. (Dies geschieht in der Annahme, dass sich Prozeduren, die von einer Prozedur abhängig sind, die beim vorherigen Versuch erfolgreich erstellt wurde, bei einem nachfolgenden Versuch erfolgreich neu erstellen lassen.) Die Operation setzt die Versuche, diese Prozeduren, deren Neuerstellung fehlgeschlagen ist, erneut zu erstellen, so lange fort, wie sie in der Lage ist, mindestens eine weitere Prozedur bei einem nachfolgenden Versuch erneut zu erstellen. Bei jedem Versuch, eine Prozedur erneut zu erstellen, wird ein Fehler (mit zugehörigen DDL-Anweisungen) in der Fehlerdatei protokolliert. Möglicherweise sehen Sie in der Fehlerdatei viele Einträge für dieselben Prozeduren. Diese Prozeduren können dennoch bei einem nachfolgenden Versuch erfolgreich erneut erstellt worden sein. Sie sollten nach Abschluss der Schemakopieroperation die Tabelle SYSCAT.PROCEDURES abfragen, um festzustellen, ob diese in der Fehlerdatei aufgeführten Prozeduren erfolgreich erneut erstellt wurden.

Weitere Informationen finden Sie in den Beschreibungen zur Prozedur **ADMIN_COPY_SCHEMA** und zum Dienstprogramm **db2move**.

Beispiele für das Kopieren eines Schemas mit dem Dienstprogramm 'db2move'

Zum Kopieren eines oder mehrerer Schemata aus einer Quelldatenbank in eine Zieldatenbank können Sie das Dienstprogramm **db2move** mit der Aktion 'COPY -co' verwenden. Nach der Einrichtung eines Modellschemas können Sie dieses als Vorlage für die Erstellung neuer Versionen verwenden.

Beispiel 1: Verwenden der Optionen 'COPY -c'

Im folgenden Beispiel für das Dienstprogramm **db2move** mit der Option COPY -co wird das Schema BAR aus der Datenbank 'sample' in die Zieldatenbank ('target') kopiert und in FOO umbenannt:

```
db2move sample COPY -sn BAR -co target_db target schema_map
"((BAR,FOO))" -u benutzer-id -p kennwort
```

Die neuen Zielschemaobjekte werden mit den gleichen Objektnamen wie die Objekte im Quellschema, jedoch mit dem Qualifikationsmerkmal 'target' erstellt. Es ist möglich, Kopien von Tabellen mit oder ohne Daten aus der Quellentabelle zu erstellen. Die Quelldatenbank und die Zieldatenbank können sich auf verschiedenen Systemen befinden.

Beispiel 2: Angeben der Namenszuordnungen für Tabellenbereiche bei der COPY-Operation

Das folgende Beispiel zeigt, wie bestimmte Namenszuordnungen für Tabellenbereiche angegeben werden, die bei einer COPY-Operation mit dem Dienstprogramm **db2move** anstelle der Tabellenbereiche aus dem Quellsystem verwendet werden sollen. Sie können das Schlüsselwort SYS_ANY angeben, um festzulegen, dass der Zieltabellenbereich mithilfe des Standardauswahlalgorithmus für Tabellenbereiche ausgewählt werden soll. In diesem Fall wählt das Dienstprogramm **db2move** alle verfügbaren Tabellenbereiche zur Verwendung als Zielobjekte aus:

```
db2move sample COPY -sn BAR -co target_db target schema_map
"((BAR,FOO))" tablespace_map "(SYS_ANY)" -u benutzer-id -p kennwort
```

Das Schlüsselwort SYS_ANY kann für alle Tabellenbereiche verwendet werden. Alternativ können Sie auch bestimmte Zuordnungen für einige der Tabellenbereiche und den Standardauswahlalgorithmus für die restlichen Tabellenbereiche angeben:

```
db2move sample COPY -sn BAR -co target_db target schema_map "
((BAR,FOO))" tablespace_map "((TS1, TS2),(TS3, TS4), SYS_ANY)"
-u benutzer-id -p kennwort
```

In diesem Beispiel wird angegeben, dass Tabellenbereich TS1 dem Tabellenbereich TS2 und Tabellenbereich TS3 dem Tabellenbereich TS4 zugeordnet werden, die restlichen Tabellenbereiche jedoch den Standardauswahlalgorithmus für Tabellenbereiche verwenden.

Beispiel 3: Ändern der Objekteigner nach der COPY-Operation

Nach einer erfolgreichen COPY-Operation können Sie den Eigner der neuen Objekte ändern, die im Zielschema erstellt wurden. Der Standardeigner der Zielobjekte ist der Benutzer, der die Verbindung hergestellt hat. Wenn diese Option angegeben wird, wird das Eigentumsrecht an einen neuen Eigner wie folgt übertragen:

```
db2move sample COPY -sn BAR -co target_db target schema_map
"((BAR,FOO))" tablespace_map "(SYS_ANY)" owner jrichards
-u benutzer-id -p kennwort
```

Der neue Eigner der Zielobjekte ist 'jrichards'.

EXPORT

Exportiert alle Tabellen, die die Filterkriterien in *optionen* erfüllen. Sind keine *optionen* angegeben, werden alle Tabellen exportiert. Interne Zwischenspeicherinformationen werden in der Datei `db2move.lst` gespeichert.

IMPORT

Importiert alle Tabellen, die in der internen Zwischenspeicherdatei `db2move.lst` aufgeführt sind. Bestimmte IMPORT-Aktionen können mit der Option `-io` angegeben werden.

LOAD

Lädt alle Tabellen, die in der internen Zwischenspeicherdatei `db2move.lst` aufgeführt sind. Bestimmte LOAD-Aktionen können mit der Option `-lo` angegeben werden.

COPY Dupliziert Schemata in eine Zieldatenbank. Die Zieldatenbank muss eine lokale Datenbank sein. Ein Schema bzw. mehrere Schemata wird/werden mit der Option `-sn` angegeben. Siehe die Option `-co` für spezielle **COPY**-Optionen. Mit der Option `-tn` oder `-tf` können Sie Tabellen im Modus **LOAD_ONLY** filtern. Wenn die gespeicherte Prozedur `ADMIN_COPY_SCHEMA()` verwendet wird oder das Dienstprogramm 'db2move' mit der Option `-COPY` verwendet wird, müssen Sie einen Tabellenbereich mit der Bezeichnung `SYS-TOOLS` verwenden.

Die nachstehende Liste nennt die Dateien, die während der einzelnen Aktionen jeweils generiert werden.

`-tc` *tabellendefinierer*

Standardmäßig werden alle definierenden Benutzer ausgewählt.

Diese Option gilt nur für die Aktion EXPORT. Wenn sie angegeben wird, werden nur die Tabellen exportiert, die von den mit dieser Option angegebenen Benutzern definiert wurden. Wird sie nicht angegeben, werden standardmäßig alle definierenden Benutzer verwendet. Bei Angabe mehrerer definierender Benutzer müssen diese Angaben jeweils durch ein Komma getrennt werden. Leerzeichen sind zwischen den IDs der definierenden Benutzer nicht zulässig. Diese Option kann zusammen mit der Option `-tn` *tabellennamen* verwendet werden, um die Tabellen zum Exportieren auszuwählen.

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 oder COPY.

Bei Angabe zusammen mit der Aktion EXPORT werden nur die Tabellen exportiert, deren Namen mit den in der angegebenen Zeichenfolge enthaltenen Namen 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. Tabellennamen sollten ohne Qualifikationsmerkmal aufgelistet werden, und die Option `-sn` sollte zum Filtern von Schemata verwendet werden.

Beim Export kann ein Stern (*) an einer beliebigen Stelle in der Zeichenfolge als Platzhalterzeichen angegeben werden.

Bei Angabe zusammen mit der Aktion COPY muss **-co "MODE" LOAD_ONLY kopieroption** ebenfalls angegeben werden, und nur die angegebenen Tabellen werden in der Zieldatenbank erneut aufgefüllt. Die Tabellennamen sollten mit ihrem Schemaqualifikationsmerkmal im Format "schema"."tabelle" aufgelistet werden.

-sn *schemanamen*

Standardmäßig werden bei EXPORT (nicht bei COPY) alle Schemata verwendet.

Falls angegeben, werden nur diejenigen Tabellen exportiert oder kopiert, deren Schemanamen übereinstimmen. Bei der Angabe mehrerer Schemanamen müssen diese Angaben durch ein Komma getrennt werden. Leerzeichen sind zwischen Schemanamen nicht zulässig. Schemanamen mit weniger als 8 Zeichen werden auf 8 Zeichen Länge aufgefüllt.

Das Platzhalterzeichen Stern (*) in den Schemanamen wird beim Exportieren in ein Prozentzeichen (%) geändert, und der Tabellename (mit Prozentzeichen) wird im Prädikat LIKE der Klausel WHERE verwendet. Wird sie nicht angegeben, werden standardmäßig alle Schemata verwendet. Wenn **db2move** mit der Option **-tn** oder **-tc** verwendet wird, wirkt sich der Befehl nur auf diejenigen Tabellen aus, deren Schemata mit den angegebenen Schemanamen und deren definierende Benutzer mit den angegebenen definierenden Benutzern übereinstimmen. Ein Schemaname fred muss als **-sn fr*d*** angegeben werden (nicht als **-sn fr*d**), 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. Tabellenbereichsnamen mit weniger als 8 Zeichen werden auf 8 Zeichen Länge aufgefüllt. Beispielsweise muss der Tabellenbereichsname meintb als **-ts meine*b*** angegeben werden (nicht als **-sn meine*b**), wenn der Stern verwendet wird.

-tf *dateiname*

Bei Angabe zusammen mit der Aktion EXPORT werden nur die Tabellen exportiert, deren Namen genau mit den in der angegebenen Datei enthaltenen Namen übereinstimmen. Wird sie nicht angegeben, werden standardmäßig alle Benutzertabellen verwendet. Pro Zeile darf nur eine Tabelle aufgelistet werden, und jede Tabelle muss vollständig qualifiziert sein. Platzhalterzeichen sind in den Zeichenfolgen nicht zulässig. Es folgt ein Beispiel für den Inhalt einer Datei:

```
"SCHEMA1"."TABLE NAME1"  
"SCHEMA NAME77"."TABLE155"
```

Bei Angabe zusammen mit der Aktion COPY muss **-co "MODE" LOAD_ONLY kopieroption** ebenfalls angegeben werden, und nur die in der Datei angege-

benen Tabellen werden in der Zieldatenbank erneut aufgefüllt. Die Tabellennamen sollten mit ihrem Schemaqualifikationsmerkmal im Format "schema"."tabelle" aufgelistet werden.

-io *importoption*

Der Standardwert ist REPLACE_CREATE. Weitere Informationen zu Einschränkungen der CREATE-Funktion von IMPORT finden Sie unter „Optionen CREATE und REPLACE_CREATE des Befehls IMPORT sind veraltet“.

Gültige Werte sind INSERT, INSERT_UPDATE, REPLACE, CREATE und REPLACE_CREATE.

-lo *ladeoption*

Der Standardwert ist INSERT.

Gültige Optionen sind INSERT und REPLACE.

-co Ist COPY die Aktion des Befehls **db2move**, stehen für **-co** die nachstehenden Optionen für Folgeaktionen zur Verfügung:

"TARGET_DB db-name [USER benutzer-id USING kennwort]"

Ermöglicht es Benutzern, den Namen der Zieldatenbank und den Benutzer/das Kennwort anzugeben. (Der Benutzer und das Kennwort der Quelldatenbank können mithilfe der vorhandenen Optionen **-p** und **-u** angegeben werden). Die Klausel USER/USING ist optional. Wird für **USER** eine Benutzer-ID angegeben, muss das Kennwort nach der Klausel **USING** angegeben werden. Wird das Kennwort nicht angegeben, wird der Benutzer von **db2move** aufgefordert, es anzugeben. Das Kennwort wird aus nachstehend erläuterten Sicherheitsgründen abgefragt. **TARGET_DB** ist eine verbindliche Option für die Aktion COPY. Die mit **TARGET_DB** angegebene Zieldatenbank darf nicht mit der Quelldatenbank übereinstimmen. Zum Kopieren von Schemata innerhalb derselben Datenbank kann die Prozedur ADMIN_COPY_SCHEMA verwendet werden. Die Aktion COPY erfordert die Angabe mindestens eines Schemas (**-sn**) bzw. einer Tabelle (**-tn** oder **-tf**).

Werden zum Kopieren von Schemata von einer Datenbank in eine andere mehrere **db2move**-Befehle gleichzeitig ausgeführt, kommt es zu Deadlocks. Es darf daher immer nur ein Befehl **db2move** abgesetzt werden. Änderungen an Tabellen im Quellschema während der Verarbeitung der Kopieraktion können dazu führen, dass die Daten im Zielschema nach dem Kopieren nicht mit denen im Quellschema übereinstimmen.

"MODE"

DDL_AND_LOAD

Erstellt alle unterstützten Objekte aus dem Quellschema und füllt die Tabellen mit den Daten der Quellentabellen auf. Dies ist die Standardoption.

DDL_ONLY

Erstellt alle unterstützten Objekte aus dem Quellschema, füllt die Tabellen jedoch nicht erneut auf.

LOAD_ONLY

Lädt alle angegebenen Tabellen aus der Quelldatenbank in die Zieldatenbank. Die Tabellen müssen im Ziel bereits

vorhanden sein. Der Modus `LOAD_ONLY` erfordert die Eingabe mindestens einer Tabelle mit der Option `-tn` oder `-tf`.

Dies ist eine wahlfreie Option, die nur für die Aktion `COPY` verwendet wird.

“`SCHEMA_MAP`”

Ermöglicht es Benutzern, ein Schema beim Kopieren in ein Ziel umzubenennen. Stellt eine Liste der Zuordnungen zwischen Quellen- und Zielschemata bereit, wobei in jedem Zuordnungspaar das Quellen- und Zielschema durch ein Komma werden und in runden Klammern stehen. Beispiel: `schema_map ((s1, t1), (s2, t2))`. In diesem Fall werden Objekte aus Schema `s1` in das Schema `t1` im Ziel kopiert, und Objekte aus Schema `s2` werden in das Schema `t2` im Ziel kopiert. Standardmäßig entspricht der Name des Zielschemas dem Namen des Quellschemas. Dies wird auch empfohlen. Der Grund hierfür ist, dass der Befehl `db2move` nicht versucht, das Schema für irgendwelche qualifizierten Objekte innerhalb von Objekthauptteilen zu ändern. Daher kann die Verwendung eines unterschiedlichen Zielschmanamens zu Problemen führen, wenn sich im Objekthauptteil qualifizierte Objekte befinden.

Beispiel:`create view F00.v1 as 'select c1 from F00.t1'`

In diesem Fall wird beim Kopieren des Schemas `FOO` nach `BAR` `v1` wie folgt generiert:`create view BAR.v1 as 'select c1 from F00.t1'`

Dies schlägt entweder fehl, weil das Schema `FOO` in der Zieldatenbank nicht vorhanden ist, oder es kommt zu einem unerwarteten Ergebnis, da `FOO` sich von `BAR` unterscheidet. Wird derselbe Schemaname wie für die Quelle verwendet, können solche Probleme vermieden werden. Gibt es zwischen den Schemata übergreifende Abhängigkeiten, müssen alle voneinander abhängigen Schemata kopiert werden, da ansonsten Fehler beim Kopieren der Objekte mit den übergreifenden Abhängigkeiten auftreten.

Beispiel:`create view F00.v1 as 'select c1 from BAR.t1'`

In diesem Fall schlägt das Kopieren von `v1` entweder fehl, wenn `BAR` nicht ebenfalls kopiert wird, oder es kommt zu unerwarteten Ergebnissen, wenn `BAR` im Ziel sich von `BAR` in der Quelle unterscheidet. Der Befehl `db2move` versucht nicht, Abhängigkeiten zwischen Schemata zu ermitteln.

Dies ist eine wahlfreie Option, die nur für die Aktion `COPY` verwendet wird.

Wenn bereits ein Zielschema vorhanden ist, schlägt das Dienstprogramm fehl. Löschen Sie mithilfe der Prozedur `ADMIN_DROP_SCHEMA` das Schema und alle ihm zugeordneten Objekte.

“`NONRECOVERABLE`”

Diese Option ermöglicht dem Benutzer, das Standardverhalten der Ladeoperation außer Kraft zu setzen, die zusammen mit `COPY-NO` ausgeführt wird. Beim Standardverhalten wird der Benutzer gezwungen, Backups aller Tabellenbereiche zu erstellen, in die geladen wurde. Bei Angabe dieses Schlüsselworts `NONRECOVERABLE` wird der Benutzer nicht gezwungen, sofort Backups der Tabellenbereiche zu erstellen. Es wird jedoch dringend empfohlen,

die Backups so bald wie möglich durchzuführen, um sicherzustellen, dass die neu erstellten Tabellen bei Bedarf ordnungsgemäß wiederhergestellt werden können. Dies ist eine wahlfreie Option, die nur für die Aktion COPY verfügbar ist.

“OWNER”

Diese Option ermöglicht es Benutzern, den Eigner jedes neuen Objekts zu ändern, das im Zielschema nach einer erfolgreichen COPY-Aktion erstellt wird. Der Standardeigner der Zielobjekte ist der Benutzer, der die Verbindung hergestellt hat. Wenn diese Option angegeben ist, wird das Eigentumsrecht auf den neuen Eigner übertragen. Dies ist eine wahlfreie Option, die nur für die Aktion COPY verfügbar ist.

“TABLESPACE_MAP”

Der Benutzer kann Namenszuordnungen für Tabellenbereiche angeben, die bei einer Operation COPY anstelle der Tabellenbereiche aus dem Quellensystem verwendet werden sollen. Dies ist eine Matrix mit Tabellenbereichszuordnungen, die in Klammern eingeschlossen ist. Beispiel: `tablespace_map ((TS1, TS2), (TS3, TS4))`. In diesem Fall werden Objekte aus dem Tabellenbereich TS1 in den Tabellenbereich TS2 der Zieldatenbank kopiert und Objekte aus dem Tabellenbereich TS3 werden in den Tabellenbereich TS4 im Ziel kopiert. Bei einer Matrix `((T1, T2), (T2, T3))` werden alle Objekte in T1 in der Quelldatenbank in T2 in der Zieldatenbank erneut erstellt, und alle Objekte in T2 in der Quelldatenbank werden in T3 in der Zieldatenbank erneut erstellt. Standardmäßig wird derselbe Tabellenbereichsname wie in der Quelle verwendet. In diesem Fall ist die Eingabezuordnung für diesen Tabellenbereich nicht erforderlich. Wenn der angegebene Tabellenbereich nicht vorhanden ist, schlägt das Kopieren der Objekte, die diesen Tabellenbereich verwenden, fehl und wird in der Fehlerdatei protokolliert.

Der Benutzer kann außerdem das Schlüsselwort `SYS_ANY` angeben, um festzulegen, dass der Zieltabellenbereich mithilfe des Standardauswahlalgorithmus für Tabellenbereiche ausgewählt werden soll. In diesem Fall kann `db2move` alle verfügbaren Tabellenbereiche zur Verwendung als Zielobjekte auswählen. Das Schlüsselwort `SYS_ANY` kann für alle Tabellenbereiche verwendet werden (Beispiel: `tablespace_map SYS_ANY`). Außerdem kann der Benutzer für einige Tabellenbereiche bestimmte Zuordnungen angeben, und für die restlichen Tabellenbereiche den Standardauswahlalgorithmus. Beispiel: `tablespace_map ((TS1, TS2), (TS3, TS4), SYS_ANY)`. In diesem Beispiel wird angegeben, dass der Tabellenbereich TS1 dem Tabellenbereich TS2 zugeordnet wird und der Tabellenbereich TS3 dem Tabellenbereich TS4. Die restlichen Tabellenbereiche verwenden jedoch ein Standardtabellenbereichsziel. Das Schlüsselwort `SYS_ANY` wird verwendet, da ein Tabellenbereich nicht mit "SYS" beginnen darf.

Dies ist eine wahlfreie Option, die nur für die Aktion COPY verfügbar ist.

-I *lob-pfade*

Bei Angabe dieser Option für IMPORT und EXPORT wird sie auch für XML-Pfade verwendet. Der Standardwert ist das aktuelle Verzeichnis.

Diese Option gibt die absoluten Pfadnamen an, in denen LOB- oder XML-Dateien erstellt werden (bei EXPORT) bzw. gesucht werden (bei IMPORT)

oder LOAD). Bei Angabe mehrerer Pfade muss jeder Pfad durch ein Komma getrennt werden. Leerzeichen sind zwischen Pfaden nicht zulässig. Werden mehrere Pfade angegeben, verwendet EXPORT sie mit Round-Robin im Umlauf. So wird ein LOB-Dokument in den ersten Pfad geschrieben, eines in den zweiten Pfad usw. bis zum letzten Pfad und anschließend zurück zum ersten. Das Gleiche gilt für XML-Dokumente. Werden im ersten Pfad keine Dateien gefunden (bei Ausführung von IMPORT oder LOAD), wird der zweite Pfad verwendet usw.

-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. Wird die Option **-p** ohne Kennwort angegeben, fordert **db2move** den Benutzer zur Eingabe des Kennworts auf. Dies erfolgt aus Sicherheitsgründen. Die Eingabe des Kennworts über die Befehlszeile kann zu Sicherheitsproblemen führen. Mit dem Befehl `ps -ef` beispielsweise würde das Kennwort angezeigt werden. Wird **db2move** jedoch über ein Script aufgerufen, müssen die Kennwörter angegeben werden. Wenn der Befehl auf einem Client abgesetzt 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 ist, werden Tabellen, für die beim Exportieren Warnungen ausgegeben werden, nicht in die Datei `db2move.lst` aufgenommen, obwohl die Dateien `.ixf` und `.msg` für die betreffende Tabelle generiert werden. In manchen Szenarios (z. B. Abschneiden der Daten) möchte der Benutzer möglicherweise zulassen, dass solche Tabellen in die Datei `db2move.lst` aufgenommen werden. Bei Angabe dieser Option können Tabellen, für die beim Exportieren Warnungen ausgegeben werden, in die Datei `.lst` aufgenommen werden.

Beispiele

- Der folgende Befehl kann verwendet werden, um alle Tabellen in der Beispieldatenbank SAMPLE unter Verwendung der Standardwerte für alle Optionen zu exportieren:

```
db2move sample export
```

- Der folgende Befehl kann verwendet werden, um alle Tabellen zu exportieren, die von `userid1` oder Benutzer-IDs wie `us*rid2` und mit den Namen `tname1` oder Tabellennamen wie `%tname2` erstellt wurden:

```
db2move sample export -tc userid1,us*rid2 -tn tname1,*tname2
```

- Setzen Sie den folgenden Befehl ab, um alle Tabellen in der Beispieldatenbank SAMPLE zu importieren (die LOB-Pfade `D:\LOBPATH1` und `C:\LOBPATH2` werden nach LOB-Dateien durchsucht; dieses Beispiel gilt nur für Windows-Betriebssysteme):

```
db2move sample import -l D:\LOBPATH1,C:\LOBPATH2
```


- Setzen Sie den folgenden Befehl ab, um alle Tabellen aus der Beispieldatenbank SAMPLE zu laden (die Unterverzeichnisse /home/userid/lobpath und tmp werden nach LOB-Dateien durchsucht; dieses Beispiel gilt nur für Linux- und UNIX-Systeme):

```
db2move sample load -l /home/userid/lobpath,/tmp
```

- Der folgende Befehl kann verwendet werden, um alle Tabellen in die Beispieldatenbank SAMPLE im Ersetzungsmodus (REPLACE) zu importieren und dabei die angegebene Kombination aus Benutzer-ID (userid) und Kennwort (password) zu verwenden:

```
db2move sample import -io replace -u userid -p password
```

- Der folgende Befehl kann verwendet werden, um das Schema schema1 aus der Quelldatenbank dbsrc in der Zieldatenbank dbtgt zu duplizieren:

```
db2move dbsrc COPY -sn schema1 -co TARGET_DB dbtgt USER myuser1 USING mypass1
```

- Setzen Sie den folgenden Befehl ab, um das Schema schema1 aus der Quelldatenbank dbsrc in die Zieldatenbank dbtgt zu duplizieren, das Schema in der Zieldatenbank in newschema1 umzubenennen und den Quellentabellenbereich ts1 dem Zieltabellenbereich ts2 zuzuordnen:

```
db2move dbsrc COPY -sn schema1 -co TARGET_DB dbtgt USER myuser1 USING mypass1
```

Hinweise zur Verwendung

- Beim Kopieren von einem oder mehreren Schemas in eine Zieldatenbank müssen die Schemas unabhängig voneinander sein. Ist dies nicht der Fall, werden möglicherweise einige Objekte nicht erfolgreich in die Zieldatenbank kopiert.
- Das Laden von Daten in Tabellen mit XML-Spalten wird nur für die Aktion **LOAD**, nicht aber für die Aktion **COPY**, unterstützt. Als Ausweichlösung kann der Befehl **IMPORT** oder **EXPORT** manuell abgesetzt werden, oder es können die Verhaltnisse **db2move Export** und **db2move Import** verwendet werden. Wenn diese Tabellen zudem Identitätsspalten des Typs GENERATED ALWAYS enthalten, können keine Daten in die Tabellen importiert werden.
- Eine Aktion **db2move EXPORT**, gefolgt von einer Aktion **db2move IMPORT/LOAD**, erleichtert das Versetzen von Tabellendaten. Alle anderen den Tabellen zugeordneten Datenbankobjekte (z. B. Aliasnamen, Sichten oder Auslöser) sowie Objekte, von denen diese Tabellen möglicherweise abhängen (z. B. benutzerdefinierte Typen oder Funktionen), müssen manuell versetzt werden.
- Wenn die Aktion **IMPORT** mit der Option **CREATE** oder **REPLACE_CREATE** (beide Optionen sind veraltet und werden in einem zukünftigen Release möglicherweise entfernt) verwendet wird, um die Tabellen in der Zieldatenbank zu erstellen, gelten die in „Neuerstellung von importierten Tabellen“ beschriebenen Einschränkungen. Werden während der Importphase von **db2move** bei Verwendung der Option **REPLACE_CREATE** unerwartete Fehler festgestellt, sollten Sie die entsprechende Nachrichtendatei tabnnn.msg prüfen und überlegen, ob die Fehler auf die Einschränkungen der Tabellenerstellung zurückzuführen sind.
- Tabellen mit als GENERATED ALWAYS definierten Identitätsspalten können nicht mit **db2move** importiert oder geladen werden. Diese Tabellen können jedoch manuell importiert oder geladen werden. Weitere Informationen hierzu finden Sie unter „Aspekte des Ladens von Identitätsspalten“ oder „Aspekte des Identitätsspaltenimports“.
- Wenn die APIs EXPORT, IMPORT oder LOAD von **db2move** aufgerufen werden, wird der Parameter **FileTypeMod** auf 'lobsinfile' gesetzt. Dies bedeutet, dass LOB-Daten für jede Tabelle in separaten Dateien (nicht in der PC/IXF-Datei) erfasst werden.
- Der Befehl **LOAD** muss lokal auf der Maschine ausgeführt werden, auf der sich die Datenbank und die Datendatei befindet.

- Wenn **db2move LOAD** verwendet wird und logretain für die Datenbank aktiviert ist (d. h. die Datenbank ist wiederherstellbar), gilt Folgendes:
 - Ist die Option **NONRECOVERABLE** nicht angegeben, so ruft **db2move** die API db2Load mit der Standardoption NO COPY auf, und die Tabellenbereiche, in denen sich die geladenen Tabellen befinden, werden bei Beendigung des Dienstprogramms in den Status "Backup anstehend" versetzt. (Ein vollständiges Backup ist erforderlich, um den Status "Backup anstehend" für die Tabellenbereiche aufzuheben.)
 - Ist die Option **NONRECOVERABLE** nicht angegeben, werden die Tabellenbereiche nicht in den Status "Backup anstehend" versetzt. Wird jedoch später eine aktualisierende Recovery durchgeführt, so wird die Tabelle als nicht zugänglich markiert und muss gelöscht werden. Weitere Informationen zu den Wiederherstellbarkeitsoption für LOAD finden Sie unter „Optionen zur Verbesserung der Ladeleistung“.
- Die Leistung des Befehls **db2move** mit den Aktionen **IMPORT** oder **LOAD** kann durch Ändern des Standardpufferpools IBMDEFAULTBP und durch Aktualisieren der Konfigurationsparameter **sortheap**, **util_heap_sz**, **logfilsiz** und **logprimary** verbessert werden.
- Bei der Ausführung von Dienstprogrammen für das Versetzen von Daten wie **export** und **db2move** stellt der Abfragecompiler möglicherweise fest, dass die zugrunde liegende Abfrage für eine MQT (Materialized Query Table, gespeicherte Abfragetabelle) effizienter ausgeführt wird als für die Basistabelle(n). In diesem Fall wird die Abfrage für eine als REFRESH DEFERRED deklarierte MQT ausgeführt und das Ergebnis der Dienstprogramme zeigt die Daten in der zugrunde liegenden Tabelle möglicherweise nicht exakt an.
- Der Befehl **db2move** ist bei DB2-Clients nicht verfügbar. Wenn Sie den Befehl **db2move** auf einer Clientmaschine absetzen, erhalten Sie eine Fehlermeldung, die angibt, dass db2move nicht als interner oder externer Befehl, ausführbares Programm oder Batchdatei erkannt wird. Sie können diesen Fehler vermeiden, indem Sie den Befehl **db2move** direkt auf dem Server absetzen.

Bei Verwendung von EXPORT erforderliche/generierte Dateien:

- Eingabe: Keine.
- Ausgabe:

EXPORT.out

Das zusammengefasste Ergebnis der Aktion EXPORT.

db2move.lst

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 **db2move** verwendet.

tabnnn.ixf

Die exportierte PC/IXF-Datei einer bestimmten Tabelle.

tabnnn.msg

Die exportierte Nachrichtendatei der entsprechenden Tabelle.

tabnnnc.yyy

Die exportierten LOB-Dateien einer bestimmten Tabelle.

"nnn" ist die Tabellenummer. "c" ist ein Buchstabe des Alphabets. „yyy“ ist eine Zahl im Bereich 001 bis 999.

Diese Dateien werden nur erstellt, sofern die Tabelle, die exportiert wird, LOB-Daten enthält. Wenn sie erstellt werden, werden diese LOB-Dateien in den LOB-Pfadverzeichnissen (Wert für "lob-pfade") gespeichert. Für die LOB-Dateien sind insgesamt 26.000 Namen möglich.

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.

Bei Verwendung von COPY erforderliche/generierte Dateien:

- Eingabe: Keine

- Ausgabe:

COPYSCHEMA.msg

Eine Ausgabedatei mit Nachrichten, die während der Operation COPY generiert wurden.

COPYSCHEMA.err

Eine Ausgabedatei mit einer Fehlermeldung für jeden Fehler, der wäh-

rend der Operation COPY festgestellt wurde (einschließlich DDL-Anweisungen für jedes Objekt, das in der Zieldatenbank nicht erneut erstellt werden konnte).

LOADTABLE.msg

Eine Ausgabedatei mit Nachrichten, die bei den einzelnen Aufrufen des Dienstprogramms LOAD generiert wurden, das zum erneuten Auffüllen der Daten in der Zieldatenbank verwendet wurde.

LOADTABLE.err

Eine Ausgabedatei mit den Namen der Tabellen, für die bei der Ladeoperation ein Fehler aufgetreten ist, oder die in der Zieldatenbank noch gefüllt werden müssen. Weitere Details finden Sie im Thema „Erneutes Starten einer fehlgeschlagenen Operation zum Kopieren eines Schemas“.

Diese Dateien werden mit einer Zeitmarke versehen, wobei alle Dateien, die im Verlauf derselben Ausführung von COPY generiert werden, auch dieselbe Zeitmarke erhalten.

Ausführen eines umgeleiteten Restores mithilfe eines automatisch generierten Scripts

Wenn Sie eine umgeleitete Restoreoperation durchführen, müssen Sie die Positionen physischer Container angeben, die im Backup-Image gespeichert sind, und vollständige Informationen zur Gruppe von Containern für jeden Tabellenbereich bereitstellen, den Sie ändern wollen. Verwenden Sie die folgende Prozedur, um ein Script für den umgeleiteten Restore auf der Basis eines vorhandenen Backup-Images zu generieren, das Script zu modifizieren und es anschließend zur Durchführung der umgeleiteten Restoreoperation auszuführen.

Vorbereitende Schritte

Sie können einen umgeleiteten Restore nur dann ausführen, wenn die Datenbank zuvor mit dem DB2-Backup-Dienstprogramm gesichert wurde.

Informationen zu diesem Vorgang

- Wenn die Datenbank vorhanden ist, müssen Sie eine Verbindung zu ihr herstellen können, um das Script zu generieren. Daher gilt: Wenn für die Datenbank ein Upgrade oder eine Recovery nach Systemabsturz erforderlich ist, muss dieses Upgrade bzw. diese Recovery ausgeführt werden, bevor Sie versuchen, ein Script für den umgeleiteten Restore zu generieren.
- Wenn Sie in einer Umgebung mit partitionierten Datenbanken arbeiten und die Zieldatenbank nicht vorhanden ist, können Sie den Befehl zur Generierung des Scripts für den umgeleiteten Restore nicht gleichzeitig in allen Datenbankpartitionen ausführen. Stattdessen muss der Befehl zur Generierung des Scripts für den umgeleiteten Restore jeweils nur in einer Datenbankpartition gleichzeitig, beginnend mit der Katalogpartition, ausgeführt werden.
Alternativ können Sie zuerst eine Pseudodatenbank mit demselben Namen wie Ihre Zieldatenbank erstellen. Nach der Erstellung der Pseudodatenbank können Sie das Script für den umgeleiteten Restore in allen Datenbankpartitionen gleichzeitig generieren.
- Auch wenn Sie die Option REPLACE EXISTING bei der Ausführung des Befehls RESTORE zur Generierung des Scripts angeben, wird die Option REPLACE EXISTING in dem Script auf Kommentar gesetzt.
- Aus Sicherheitsgründen wird Ihr Kennwort nicht in das generierte Script eingetragen. Sie müssen das Kennwort manuell einfügen.

- Sie können das Script für einen umgeleiteten Restore nicht über den Restoreassistenten in der Steuerzentrale generieren.

Gehen Sie zur Durchführung eines umgeleiteten Restores mithilfe eines Scripts wie folgt vor:

Vorgehensweise

1. Generieren Sie mithilfe des Restoredienstprogramms ein Script für den umgeleiteten Restore. Das Restoredienstprogramm kann über den Befehlszeilenprozessor (CLP) oder die Anwendungsprogrammierschnittstelle **db2Restore** aufgerufen werden. Das folgende Beispiel zeigt einen Befehl **RESTORE DATABASE** mit der Option **REDIRECT** und der Option **GENERATE SCRIPT**:

```
db2 restore db test from /home/jseifert/backups taken at 20050304090733
      redirect generate script test_node0000.clp
```

Dieser Befehl erstellt ein Script für den umgeleiteten Restore auf dem Client mit dem Namen `test_node0000.clp`.

2. Öffnen Sie das Script für den umgeleiteten Restore in einem Texteditor, um alle erforderlichen Modifikationen vorzunehmen. Folgende Angaben können modifiziert werden:
 - **RESTORE**-Optionen
 - Pfade für dynamischen Speicher
 - Layout und Pfade von Containern
3. Führen Sie das modifizierte Script für den umgeleiteten Restore aus. Beispiel:

```
db2 -tvf test_node0000.clp
```

RESTORE DATABASE

Der Befehl **RESTORE DATABASE** erstellt eine beschädigte Datenbank erneut, die mit dem DB2-Backup-Dienstprogramm gesichert wurde. Die wiederhergestellte Datenbank weist den gleichen Zustand auf, den sie beim Erstellen der Backup-Kopie hatte.

Dieses Dienstprogramm kann die folgenden Operationen durchführen:

- Es kann eine Datenbank mit einem anderen Image überschreiben oder die Backup-Kopie in einer neuen Datenbank wiederherstellen.
- Es kann für einen Restore von Backup-Images in DB2 Version 9.7 verwendet werden, die mit DB2 Universal Database Version 8, DB2 Version 9.1 oder DB2 Version 9.5 erstellt wurden.
 - Wenn ein Datenbankupgrade erforderlich sein sollte, wird die Upgradeoperation nach dem Restore automatisch aufgerufen.
- Wenn die Datenbank zum Zeitpunkt der Backup-Operation für aktualisierende Recovery aktiviert war, kann die Datenbank wieder in den vorherigen Zustand versetzt werden, indem nach erfolgreicher Beendigung einer Restoreoperation das Dienstprogramm für aktualisierende Recovery aufgerufen wird.
- Es kann ein Restore eines Tabellenbereichs durchgeführt werden.
- Sätze von Tabellenbereichen und SQL-Schemas können mithilfe der Option **TRANSPORT** vom Datenbankbackup-Image in eine Datenbank transportiert werden (ab DB2 Version 9.7 Fixpack 2).

Informationen zu den von DB2-Datenbanksystemen unterstützten Wiederherstellungsoperationen zwischen unterschiedlichen Betriebssystemen und Hardwareplattformen enthält der Abschnitt „Backup- und Restoreoperationen zwischen

unterschiedlichen Betriebssystemen und Hardwareplattformen“ in *Datenrecovery und hohe Verfügbarkeit Handbuch und Referenz*.

Inkrementelle Images und Images, die nur die Unterschiede seit dem Zeitpunkt der letzten Aufzeichnung erfassen („Delta-Image“), können nicht wiederhergestellt werden, wenn sich die Betriebssysteme oder die Wortgröße (32-Bit oder 64-Bit) unterscheiden.

Nach einer erfolgreichen Wiederherstellungsoperation von einer Umgebung in eine andere sind inkrementellen Backups oder Delta-Backups erst wieder zulässig, nachdem ein nicht inkrementelles Backup erstellt wurde. (Diese Einschränkung gilt nicht nach Wiederherstellungsoperationen innerhalb derselben Umgebung.)

Auch bei erfolgreichen Wiederherstellungsoperationen von einer Umgebung in eine andere sind einige Dinge zu beachten: Pakete müssen vor der Verwendung erneut gebunden werden (mit dem Befehl **BIND**, dem Befehl **REBIND** oder dem Dienstprogramm **db2rbind**), SQL-Prozeduren müssen gelöscht und erneut erstellt werden, und alle externen Bibliotheken müssen auf der neuen Plattform erneut erstellt werden. (Diese Überlegungen gelten nicht für die Wiederherstellung in derselben Umgebung.)

Eine Restoreoperation, die für eine vorhandene Datenbank und vorhandene Container ausgeführt wird, verwendet dieselben Container und dieselbe Tabellenbereichszuordnung wieder.

Eine Restoreoperation, die für eine neue Datenbank ausgeführt wird, ordnet alle Container neu zu und stellt eine optimierte Tabellenbereichszuordnung wieder her. Eine Restoreoperation, die für eine vorhandene Datenbank mit mindestens einem fehlenden Container ausgeführt wird, ordnet ebenfalls alle Container neu zu und stellt eine optimierte Tabellenbereichszuordnung wieder her.

Geltungsbereich

Dieser Befehl wirkt sich nur auf den Knoten aus, auf dem er ausgeführt wird.

Berechtigung

Zum Wiederherstellen in eine vorhandene Datenbank eine der folgenden Berechtigungen:

- SYSADM
- SYSCtrl
- SYSMaint

Zum Wiederherstellen in eine neue Datenbank eine der folgenden Berechtigungen:

- SYSADM
- SYSCtrl

Erforderliche Verbindung

Die erforderliche Verbindung variiert je nach Typ der Wiederherstellungsaktion:

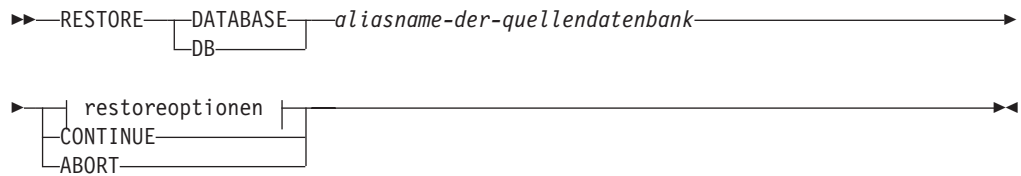
- Für die Wiederherstellung in eine vorhandene Datenbank ist eine Datenbankverbindung erforderlich. Dieser Befehl stellt automatisch eine exklusive Verbindung zu der angegebenen Datenbank her.

- Für die Wiederherstellung in eine neue Datenbank ist eine Instanz und eine Datenbankverbindung erforderlich. Der Instanzanschluss wird zum Erstellen der Datenbank benötigt.

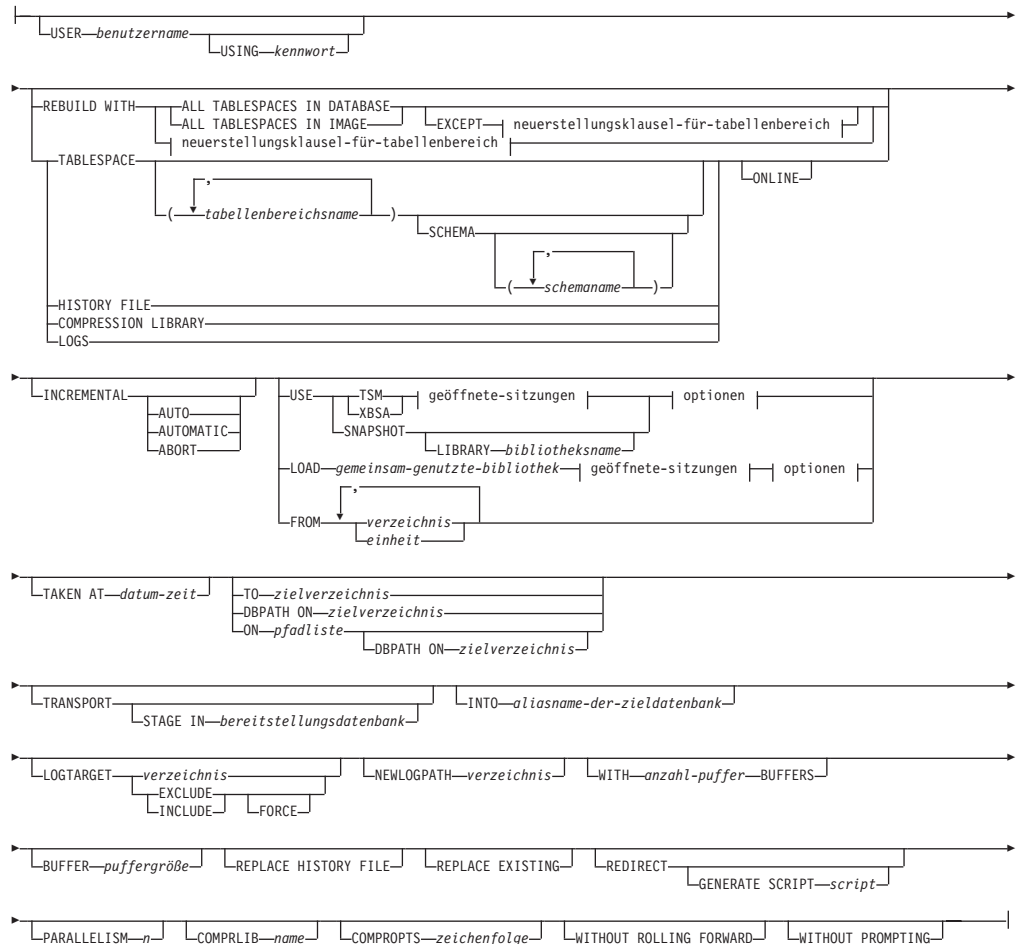
Beim Restore in eine neue Datenbank in einer anderen als der aktuellen Instanz muss zunächst eine Verbindung zu der Instanz hergestellt werden, in der sich die neue Datenbank befinden soll. Die neue Instanz kann eine lokale oder ferne Instanz sein. Die aktuelle Instanz wird durch den Wert der Umgebungsvariablen DB2INSTANCE definiert.

- Für den Momentaufnahme-restore sind *Instanz-* und *Datenbankverbindungen* erforderlich.

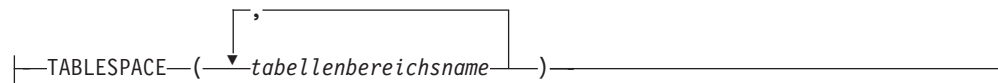
Befehlssyntax



restoreoptionen:



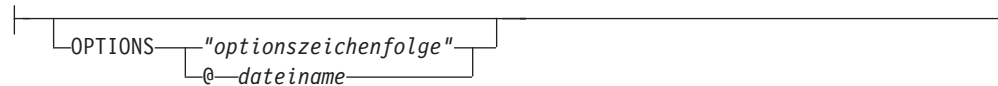
neuerstellungsklausel-für-tabellenbereich:



geöffnete-sitzungen:



optionen:



Befehlsparameter

DATABASE *aliasname-der-quellendatenbank*

Der Aliasname der Quelldatenbank, aus der das Backup stammt.

CONTINUE

Gibt an, dass die Container neu definiert wurden und der abschließende Schritt einer umgeleiteten Restoreoperation ausgeführt werden sollte.

ABORT

Dieser Parameter:

- Stoppt eine umgeleitete Restoreoperation. Dies ist hilfreich, wenn ein Fehler aufgetreten ist, der die Wiederholung mindestens eines Schrittes erforderlich macht. Nach dem Absetzen von **RESTORE DATABASE** mit der Option **ABORT** muss jeder Schritt einer umgeleiteten Restoreoperation wiederholt werden, einschließlich **RESTORE DATABASE** mit der Option **RE-DIRECT**.
- Beendet eine inkrementelle Restoreoperation, bevor sie abgeschlossen ist.

USER *benutzername*

Gibt an, unter welchem Benutzernamen die Datenbank wiederhergestellt werden soll.

USING *kennwort*

Das zur Authentifizierung des Benutzernamens verwendete Kennwort. Wenn das Kennwort übergangen wird, wird der Benutzer aufgefordert, es einzugeben.

REBUILD WITH ALL TABLESPACES IN DATABASE

Stellt die Datenbank mit allen Tabellenbereichen wieder her, die der Datenbank zum Zeitpunkt des wiederherzustellenden Images bekannt waren. Bei dieser Restoreoperation wird eine bereits vorhandene Datenbank überschrieben.

REBUILD WITH ALL TABLESPACES IN DATABASE EXCEPT *neuerstellungsklausel-für-tabellenbereich*

Stellt die Datenbank mit allen Tabellenbereichen wieder her, die der Datenbank zum Zeitpunkt des wiederherzustellenden Images bekannt waren, mit Ausnahme der in der Liste angegebenen Tabellenbereiche. Bei dieser Restoreoperation wird eine bereits vorhandene Datenbank überschrieben.

REBUILD WITH ALL TABLESPACES IN IMAGE

Stellt die Datenbank nur mit den Tabellenbereichen aus dem wiederherzustellenden Image wieder her. Bei dieser Restoreoperation wird eine bereits vorhandene Datenbank überschrieben.

REBUILD WITH ALL TABLESPACES IN IMAGE EXCEPT *neuerstellungsklausel-für-tabellenbereich*

Stellt die Datenbank nur mit den Tabellenbereichen aus dem wiederherzustellenden Image wieder her, mit Ausnahme der in der Liste angegebenen Tabellenbereiche. Bei dieser Restoreoperation wird eine bereits vorhandene Datenbank überschrieben.

REBUILD WITH *neuerstellungsklausel-für-tabellenbereich*

Stellt die Datenbank nur mit den in der Liste aufgeführten Tabellenbereichen wieder her. Bei dieser Restoreoperation wird eine bereits vorhandene Datenbank überschrieben.

TABLESPACE *tabellenbereichsname*

Eine Namensliste zum Angeben der Tabellenbereiche, die wiederhergestellt werden sollen.

Wenn die Option **TRANSPORT** festgelegt ist, müssen Tabellenbereichsnamen angegeben werden.

SCHEMA *schemaname*

Eine Namensliste zum Angeben der Schemas, die wiederhergestellt werden sollen.

Wenn die Option **TRANSPORT** festgelegt ist, müssen Schemanamen angegeben werden. Die Option **SCHEMA** ist nur gültig, wenn die Option **TRANSPORT** angegeben wird.

ONLINE

Mit diesem Schlüsselwort (nur anwendbar für Restoreoperationen auf Tabellenbereichsebene) kann angegeben werden, dass ein Backup-Image online wiederhergestellt werden darf. Dies bedeutet, dass andere Agenten Verbindung zu dieser Datenbank herstellen können, während das Backup-Image wiederhergestellt wird, und dass die Daten in anderen Tabellenbereichen zur Verfügung stehen, während die angegebenen Tabellenbereiche wiederhergestellt werden.

HISTORY FILE

Dieses Schlüsselwort gibt an, dass nur die Protokolldatei aus dem Backup-Image wiederhergestellt werden soll.

COMPRESSION LIBRARY

Dieses Schlüsselwort gibt an, dass nur das Komprimierungswörterbuch aus dem Backup-Image wiederhergestellt werden soll. Wenn das Objekt im Backup-Image enthalten ist, wird es in das Datenbankverzeichnis wiederhergestellt. Ist das Objekt nicht im Backup-Image enthalten, schlägt die Restoreoperation fehl.

LOGS Dieses Schlüsselwort gibt an, dass nur die im Backup-Image enthaltenen Protokolldateien wiederhergestellt werden sollen. Enthält das Backup-Image keine Protokolldateien, schlägt die Restoreoperation fehl. Wenn diese Option angegeben wird, muss die Option **LOGTARGET** ebenfalls angegeben werden.

INCREMENTAL

Ohne weitere Parameter bezeichnet **INCREMENTAL** eine manuelle kumulative Restoreoperation. Beim manuellen Restore muss der Benutzer jeden Resto-

rebefehl für jedes an der Restoreoperation beteiligte Image manuell absetzen. Verwenden Sie dabei diese Reihenfolge: Letztes, erstes, zweites, drittes usw. bis zum letzten Image (einschließlich).

INCREMENTAL AUTOMATIC/AUTO

Gibt eine automatische kumulative Restoreoperation an.

INCREMENTAL ABORT

Gibt an, dass eine laufende, manuelle kumulative Restoreoperation abgebrochen werden soll.

USE

TSM Gibt an, dass die Datenbank aus Ausgabedaten, die von Tivoli Storage Manager verwaltet werden, wiederhergestellt werden soll.

XBSA Gibt an, dass die XBSA-Schnittstelle verwendet werden soll. XBSA (Backup Services APIs) ist eine offene Anwendungsprogrammierschnittstelle für Anwendungen oder Funktionen, die Datenspeicherungsmanagement für Backup- oder Archivierungszwecke benötigen.

SNAPSHOT

Gibt an, dass die Daten aus einem Momentaufnahmebackup wiederhergestellt werden sollen.

Der Parameter **SNAPSHOT** kann nicht zusammen mit einem der folgenden Parameter verwendet werden:

- INCREMENTAL
- TO
- ON
- DBPATH ON
- INTO
- NEWLOGPATH
- WITH *anzahl-puffer* BUFFERS
- BUFFER
- REDIRECT
- REPLACE HISTORY FILE
- COMPRESSION LIBRARY
- PARALLELISM
- COMPRLIB
- OPEN *anzahl-sitzungen* SESSIONS
- HISTORY FILE
- LOGS

Außerdem kann der Parameter **SNAPSHOT** nicht in Verbindung mit einer Restoreoperation verwendet werden, die eine Tabellenbereichsliste enthält (dies schließt die Option **REBUILD WITH** ein).

Das Standardverhalten bei der Wiederherstellung von Daten aus einem Momentaufnahmebackup-Image ist ein FULL DATABASE OFFLINE-Restore aller Pfade, die zu der Datenbank gehören, einschließlich aller Container, des lokalen Datenträgerverzeichnis und des Datenbankpfads (DBPATH). Standardmäßig werden die Protokolle von dem Momentaufnahme-restore ausgeschlossen (**LOGTARGET EXCLUDE** ist der Standardwert für alle Momentaufnahme-

merestores, es sei denn, **LOGTARGET INCLUDE** ist explizit angegeben). Wenn eine Zeitmarke angegeben ist, wird das entsprechende Momentaufnahmebackup-Image wiederhergestellt.

LIBRARY *bibliotheksname*

In IBM Data Server ist ein DB2 ACS-API-Treiber für die folgende Speicherhardware integriert:

- IBM TotalStorage SAN Volume Controller
- IBM Enterprise Storage Server Modell 800
- IBM System Storage DS6000
- IBM System Storage DS8000
- IBM System Storage N Series
- NetApp V-series
- NetApp FAS

Wenn Sie über andere Speicherhardware sowie über einen DB2 ACS-API-Treiber für diese Hardware verfügen, können Sie den DB2 ACS-API-Treiber mit dem Parameter **LIBRARY** angeben.

Der Wert für den Parameter **LIBRARY** ist der vollständig qualifizierte Name der Bibliotheksdatei.

OPTIONS

"optionszeichenfolge"

Gibt an, welche Optionen für die Restoreoperation verwendet werden sollen. Die Zeichenfolge wird genauso übergeben, wie sie eingegeben wurde (ohne die doppelten Anführungszeichen).

@dateiname

Gibt an, dass die für die Restoreoperation zu verwendenden Optionen in einer Datei enthalten sind, die sich auf dem DB2-Server befindet. Die Zeichenfolge wird an die Unterstützungsbibliothek des Herstellers übergeben. Der Dateiname muss ein vollständig qualifizierter Dateiname sein.

Mit dem Datenbankkonfigurationsparameter **VENDOROPT** können keine herstellerspezifischen Optionen für Momentaufnahmeoperationen angegeben werden. Zu diesem Zweck müssen Sie den Parameter **OPTIONS** der Restoredienstprogramme verwenden.

OPEN *anzahl-sitzungen* **SESSIONS**

Die Anzahl E/A-Sitzungen, die mit TSM oder einem Programm eines anderen Anbieters verwendet werden soll.

FROM *verzeichnis/einheit*

Der vollständig qualifizierte Pfadname des Verzeichnisses oder der Einheit, in dem bzw. auf der sich das Backup-Image befindet. Wenn die Optionen **USE TSM**, **FROM** und **LOAD** fehlen, wird standardmäßig das aktuelle Arbeitsverzeichnis der Clientmaschine verwendet. Dieses Zielverzeichnis bzw. diese Zieleinheit muss auf dem Zielsystem bzw. der Zielinstanz vorhanden sein.

Wenn mehrere Elemente angegeben werden und das letzte Element ein Bandlaufwerk ist, wird der Benutzer aufgefordert, ein anderes Band einzulegen. Gültige Antwortoptionen sind:

- c** Fortsetzen. Verwenden der Einheit fortsetzen, die die Warnung generiert hat (zum Beispiel, wenn ein neues Band eingelegt wurde).

- d** Einheit beenden. Verwendung *nur* der Einheit beenden, die die Warnung generiert hat (zum Beispiel, wenn keine Bänder mehr vorhanden sind).
- t** Beenden. Restoreoperation abbrechen, nachdem der Benutzer eine vom Dienstprogramm angeforderte Aktion nicht ausgeführt hat.

LOAD *gemeinsam-genutzte-bibliothek*

Der Name der gemeinsam genutzten Bibliothek (DLL unter Windows-Betriebssystemen), in der die zu verwendenden E/A-Funktionen für Backup und Restore des Herstellers enthalten sind. Der Name kann einen vollständigen Pfad enthalten. Wenn der vollständige Pfad nicht angegeben wird, wird standardmäßig der Pfad verwendet, in dem sich das Benutzerexitprogramm befindet.

TAKEN AT *datum-zeit*

Die Zeitmarke des Datenbank-Backup-Images. Die Zeitmarke wird nach erfolgreicher Beendigung der Backupoperation angezeigt; sie ist Bestandteil des Pfadnamens für das Backup-Image. Die Zeitmarke hat das Format *jjjmmthhmmss*. Die Zeitmarke kann auch teilweise angegeben werden. Sind beispielsweise zwei verschiedene Backup-Images mit den Zeitmarken 20021001010101 und 20021002010101 vorhanden, wird bei Angabe von 20021002 das Image mit der Zeitmarke 20021002010101 verwendet. Ist kein Wert für diesen Parameter angegeben, dürfen die Quelldatenträger nur ein Backup-Image enthalten.

TO *zielverzeichnis*

Dieser Parameter gibt das Zieldatenbankverzeichnis an. Dieser Parameter wird ignoriert, wenn die Restoreoperation in eine vorhandene Datenbank erfolgt. Das von Ihnen angegebene Laufwerk und Verzeichnis muss lokal vorhanden sein. Wenn das Backup-Image eine Datenbank mit aktiviertem dynamischem Speicher enthält, wird nur das Datenbankverzeichnis geändert, die zugeordneten Speicherpfade der Datenbank werden beibehalten.

DBPATH ON *zielverzeichnis*

Dieser Parameter gibt das Zieldatenbankverzeichnis an. Dieser Parameter wird ignoriert, wenn die Restoreoperation in eine vorhandene Datenbank erfolgt. Das von Ihnen angegebene Laufwerk und Verzeichnis muss lokal vorhanden sein. Wenn das Backup-Image eine Datenbank mit aktiviertem dynamischem Speicher enthält und der Parameter ON nicht angegeben wird, ist dieser Parameter synonym mit dem Parameter TO, d. h. nur das Datenbankverzeichnis wird geändert, die zugeordneten Speicherpfade der Datenbank werden beibehalten.

ON *pfadliste*

Dieser Parameter definiert die Speicherpfade erneut, die einer Datenbank mit dynamischem Speicher zugeordnet sind. Die Verwendung dieses Parameters mit einer Datenbank ohne aktivierten dynamischen Speicher verursacht einen Fehler (SQL20321N). Die im Backup-Image definierten Speicherpfade werden nicht länger verwendet, und die Tabellenbereiche des dynamischen Speichers werden automatisch in die neuen Pfade umgeleitet. Wenn dieser Parameter für eine Datenbank mit dynamischem Speicher nicht angegeben ist, werden die im Backup-Image definierten Speicherpfade unverändert übernommen.

Wenn mehrere Pfade angegeben werden, sind diese durch Kommas zu trennen. Jeder Pfad muss über einen absoluten Pfadnamen verfügen und lokal vorhanden sein. Wenn die Datenbank auf dem Datenträger noch

nicht vorhanden ist und der Parameter **DBPATH ON** nicht angegeben wird, wird der erste Pfad als Zieldatenbankverzeichnis verwendet.

Für eine Mehrpartitionsdatenbank kann die Option **ON pfadliste** nur in der Katalogpartition angegeben werden. Die Katalogpartition muss vor anderen Partitionen wiederhergestellt werden, wenn die Option **ON** verwendet wird. Durch die Restoreoperation der Katalogpartition mit neuen Speicherpfaden werden alle Knoten, die sich nicht im Katalog befinden, in den Status **RESTORE_PENDING** versetzt. Danach können die nicht im Katalog enthaltenen Knoten parallel wiederhergestellt werden, ohne die Klausel **ON** im Befehl **RESTORE** anzugeben.

Im Allgemeinen müssen für jede Partition einer Mehrpartitionsdatenbank dieselben Speicherpfade verwendet werden. Diese Speicherpfade müssen vor dem Ausführen des Befehls **RESTORE DATABASE** vorhanden sein. Diese Regel gilt nicht bei Verwendung von Datenbankpartitionsausdrücken innerhalb des Speicherpfads. Dieses Verfahren bietet die Möglichkeit, die Datenbankpartitionsnummer in die Speicherpfade einzufügen, sodass der resultierende Pfadname für jede Partition unterschiedlich ist.

Der Parameter **ON** kann nicht für die erneute Definition von Speicherpfaden für den Schematransport verwendet werden. Für den Schematransport werden bereits vorhandene Speicherpfade in der Zieldatenbank verwendet.

INTO *aliasname-der-zieldatenbank*

Der Aliasname der Zieldatenbank. Wenn die Zieldatenbank nicht vorhanden ist, wird sie erstellt.

Beim Restore eines Datenbankbackups in eine vorhandene Datenbank übernimmt die wiederhergestellte Datenbank den Aliasnamen und den Datenbanknamen der vorhandenen Datenbank. Beim Restore eines Datenbankbackups in eine nicht vorhandene Datenbank wird die neue Datenbank mit dem von Ihnen angegebenen Aliasnamen und Datenbanknamen erstellt. Dieser neue Datenbankname muss auf dem Zielsystem der Restoreoperation eindeutig sein.

TRANSPORT INTO *aliasname-der-zieldatenbank*

Gibt den vorhandenen Aliasnamen der Zieldatenbank für eine Transportoperation an. Die Tabellenbereiche und Schemas, die transportiert werden, werden der Datenbank hinzugefügt.

Die Optionen **TABLESPACE** und **SCHEMA** müssen Tabellenbereichs- und Schemanamen angeben, die eine gültige transportierbare Gruppe definieren. Andernfalls schlägt die Transportoperation fehl. **SQLCODE=SQL2590N rc=1**

Die vom System erstellten Tabellenbereiche können nicht transportiert werden. **SQLCODE=SQL2590N rc=4**.

Nachdem die Schemas mithilfe des Befehls **RESTORE** geprüft wurden, werden die Systemkatalogeinträge, die die Objekte in den transportierten Tabellenbereichen beschreiben, in der Zieldatenbank erstellt. Nach Abschluss der erneuten Erstellung der Schemas übernimmt die Zieldatenbank das Eigentumsrecht an den physischen Tabellenbereichscontainern.

Die in den wiederherzustellenden Tabellenbereichen enthaltenen physischen und logischen Objekte werden in der Zieldatenbank erneut erstellt und die Tabellenbereichsdefinitionen und -container werden zur Zieldatenbank hinzugefügt. Das Fehlschlagen einer Objekterstellung oder der Wiedergabe der DDL führt zu einem Fehler.

STAGE IN *bereitstellungsdatenbank*

Gibt den Namen einer temporären Bereitstellungsdatenbank für das Backup-Image an, die die Quelle für die Transportoperation ist. Wird die Option **STAGE IN** angegeben, wird die temporäre Datenbank nach Abschluss der Transportoperation nicht gelöscht. Die Datenbank wird jedoch nach dem Transport nicht mehr benötigt und kann vom Datenbankadministrator gelöscht werden.

Folgendes trifft zu, wenn die Option **STAGE IN** nicht angegeben ist:

- Der Datenbankname weist das Format **SYSTGxxx** auf, wobei *xxx* einen Ganzzahlwert darstellt.
- Die temporäre Bereitstellungsdatenbank wird nach Abschluss der Transportoperation gelöscht.

LOGTARGET *verzeichnis*

Restoreoperationen ohne Momentaufnahme:

Der absolute Pfadname eines vorhandenen Verzeichnisses auf dem Datenbankserver, das als Zielverzeichnis zum Extrahieren von Protokolldateien aus einem Backup-Image verwendet werden soll. Wenn diese Option angegeben wird, werden alle in dem Backup-Image enthaltenen Protokolldateien in das Zielverzeichnis extrahiert. Wenn diese Option nicht angegeben ist, werden die in einem Backup-Image enthaltenen Protokolldateien nicht extrahiert. Geben Sie die Option **LOGS** an, um nur die Protokolldateien aus dem Backup-Image zu extrahieren.

Restoreoperationen aus Momentaufnahmen:

INCLUDE

Protokollverzeichnisdatenträger aus dem Momentaufnahme-Image wiederherstellen. Wenn diese Option angegeben ist und das Backup-Image Protokollverzeichnisse enthält, werden diese wiederhergestellt. Vorhandene Protokollverzeichnisse und Protokolldateien auf dem Datenträger bleiben erhalten, wenn sie nicht mit den Protokollverzeichnissen im Backup-Image in Konflikt stehen. Stehen vorhandene Protokollverzeichnisse in Konflikt mit den Protokollverzeichnissen im Backup-Image, wird ein Fehler zurückgegeben.

EXCLUDE

Protokollverzeichnisdatenträger nicht wiederherstellen. Bei Angabe dieser Option werden keine Protokollverzeichnisse aus dem Backup-Image wiederhergestellt. Vorhandene Protokollverzeichnisse und Protokolldateien auf dem Datenträger bleiben erhalten, wenn sie nicht mit den Protokollverzeichnissen im Backup-Image in Konflikt stehen. Wird beim Restore eines zur Datenbank gehörenden Pfads implizit ein Protokollverzeichnis wiederhergestellt und dabei ein Protokollverzeichnis überschrieben, wird ein Fehler zurückgegeben.

FORCE

Erlaubt beim Wiederherstellen eines Momentaufnahme-Images das Überschreiben und Ersetzen vorhandener Protokollverzeichnisse in der aktuellen Datenbank. Ohne diese Option schlägt die Restoreoperation fehl, wenn Konflikte zwischen vorhandenen Protokollverzeichnissen und -dateien auf dem Datenträger und Protokollverzeichnissen im Momentaufnahme-Image bestehen. Geben Sie diese Option an, um zuzulassen, dass die vorhandenen Protokollverzeichnisse bei der Restoreoperation überschrieben und ersetzt werden können.

Anmerkung: Verwenden Sie diese Option mit Vorsicht, und stellen Sie dabei sicher, dass alle Protokolle gesichert und archiviert sind, die für die Recovery erforderlich sein können.

Anmerkung: Wenn **LOGTARGET** für Restoreoperationen mit Momentaufnahme nicht angegeben ist, gilt **LOGTARGET EXCLUDE** als standardmäßiges **LOGTARGET**-Verzeichnis.

Wenn die Option **TRANSPORT** angegeben ist und für die Funktion zum Transportieren von Schemas ein Online-Backup-Image verwendet wird, ist die Option **LOGTARGET** obligatorisch.

NEWLOGPATH *verzeichnis*

Der absolute Pfadname eines Verzeichnisses, das nach der Restoreoperation für aktive Protokolldateien verwendet wird. Dieser Parameter hat dieselbe Funktion wie der Datenbankkonfigurationsparameter **newlogpath**, er gilt jedoch nur für die Restoreoperation, in der er angegeben wird. Der Parameter kann verwendet werden, wenn der Protokollpfad im Backup-Image nach erfolgter Restoreoperation nicht weiter verwendet werden kann (z. B. weil der Pfad ungültig ist oder von einer anderen Datenbank verwendet wird).

WITH *anzahl-puffer* **BUFFERS**

Die Anzahl der Puffer, die verwendet werden sollen. Das DB2-Datenbanksystem wählt automatisch einen optimalen Wert für diesen Parameter. Eine größere Anzahl Puffer kann verwendet werden, um die Leistung zu verbessern, wenn aus mehreren Quellen gelesen wird oder wenn der Wert für **PARALLELISM** erhöht wurde.

BUFFER *puffergröße*

Die Größe des für die Restoreoperation verwendeten Puffers in Seiten. Das DB2-Datenbanksystem wählt automatisch einen optimalen Wert für diesen Parameter. Der Mindestwert für diesen Parameter ist 8 Seiten.

Die Größe der Restorepuffer muss eine positive ganze Zahl und ein Vielfaches der Backup-Puffergröße sein, die bei der Backup-Operation angegeben wurde. Wird eine nicht korrekte Puffergröße angegeben, erhalten die zugeordneten Puffer die kleinstmögliche Größe.

REPLACE HISTORY FILE

Gibt an, dass die Restoreoperation die Protokolldatei auf dem Datenträger durch die Protokolldatei aus dem Backup-Image ersetzen soll.

REPLACE EXISTING

Wenn bereits eine Datenbank mit demselben Aliasnamen wie die Zieldatenbank vorhanden ist, gibt dieser Parameter an, dass das Restoredienstprogramm die vorhandene Datenbank durch die wiederhergestellte Datenbank ersetzen soll. Dies ist hilfreich für Scripts, die das Restoredienstprogramm aufrufen, weil der Befehlszeilenprozessor den Benutzer nicht auffordert, das Löschen einer vorhandenen Datenbank zu prüfen. Wenn der Parameter **WITHOUT PROMPTING** angegeben wird, muss **REPLACE EXISTING** nicht angegeben werden, aber in diesem Fall schlägt die Operation fehl, wenn Ereignisse auftreten, die normalerweise einen Benutzereingriff erfordern.

REDIRECT

Gibt eine umgeleitete Restoreoperation an. Um eine umgeleitete Restoreoperation abzuschließen, sollte auf diesen Befehl mindestens ein Befehl **SET TABLESPACE CONTAINERS** folgen und anschließend ein Befehl **RESTORE**

DATABASE mit der Option **CONTINUE**. Alle Befehle, die einer einzigen, umgeleiteten Restoreoperation zugeordnet sind, müssen im selben Fenster oder in derselben CLP-Sitzung aufgerufen werden.

GENERATE SCRIPT *script*

Erstellt ein Script für umgeleiteten Restore mit dem angegebenen Dateinamen. Der Scriptname kann relativ oder absolut sein, und das Script wird auf der Clientseite generiert. Wenn die Datei nicht auf der Clientseite erstellt werden kann, wird eine Fehlernachricht (SQL9304N) zurückgegeben. Wenn die Datei bereits vorhanden ist, wird sie überschrieben. Weitere Informationen zur Verwendung finden Sie in den nachfolgenden Beispielen.

WITHOUT ROLLING FORWARD

Gibt an, dass die Datenbank nicht in den Status für anstehende aktualisierende Recovery versetzt werden soll, nachdem sie erfolgreich wiederhergestellt wurde.

Wenn sich die Datenbank nach einer erfolgreichen Restoreoperation im Status für anstehende aktualisierende Recovery befindet, muss der Befehl **ROLLFORWARD** aufgerufen werden, bevor die Datenbank wieder verwendet werden kann.

Wenn diese Option für eine Restoreoperation aus einem Online-Backup-Image angegeben wird, wird der Fehler SQL2537N zurückgegeben.

Wenn das Backup-Image eine wiederherstellbare Datenbank enthält, darf **WITHOUT ROLLING FORWARD** nicht mit der Option **REBUILD** angegeben werden.

PARALLELISM *n*

Gibt an, wie viele Puffermanipulatoren während der Restoreoperation erstellt werden sollen. Das DB2-Datenbanksystem wählt automatisch einen optimalen Wert für diesen Parameter.

COMPRLIB *name*

Gibt den Namen der Bibliothek an, die für die Dekomprimierung verwendet werden soll (z. B. db2compr.dll für Windows- oder libdb2compr.so für Linux/UNIX-Systeme). Der Name muss ein vollständig qualifizierter Pfad sein, der auf eine Datei auf dem Server verweist. Wenn dieser Parameter nicht angegeben wird, versucht das DB2-Datenbanksystem, die in dem Image gespeicherte Bibliothek zu verwenden. Wenn das Backup nicht komprimiert war, wird der Wert dieses Parameters ignoriert. Wenn die angegebene Bibliothek nicht geladen werden kann, schlägt die Restoreoperation fehl.

COMPROPTS *zeichenfolge*

Beschreibt einen Block mit Binärdaten, der an die Initialisierungsroutine der Dekomprimierungsbibliothek übergeben wird. Das DB2-Datenbanksystem leitet diese Zeichenfolge direkt vom Client zum Server weiter, d. h. alle Probleme durch Bytefolgeumkehrung oder Codepagekonvertierung werden von der Dekomprimierungsbibliothek verarbeitet. Wenn das erste Zeichen des Datenblocks „@“ ist, werden die übrigen Daten vom DB2-Datenbanksystem als der Name einer Datei auf dem Server angesehen. Das DB2-Datenbanksystem ersetzt den Inhalt von *zeichenfolge* durch den Inhalt dieser Datei und gibt den neuen Wert an die Initialisierungsroutine weiter. Die maximal zulässige Länge für die Zeichenfolge beträgt 1.024 Byte.

WITHOUT PROMPTING

Gibt an, dass die Restoreoperation nicht überwacht ausgeführt werden soll. Bei allen Aktionen, die normalerweise einen Benutzereingriff erfordern, wird eine Fehlernachricht zurückgegeben. Bei Verwendung eines aus-

tauschbaren Datenträgers wie Band oder Diskette wird der Benutzer zum Wechseln des Datenträgers aufgefordert, selbst wenn diese Option angegeben ist.

Beispiele

1. Im folgenden Beispiel ist die Datenbank WSDB auf allen 4 Datenbankpartitionen mit den Nummern 0 bis 3 definiert. Der Pfad /dev3/backup ist von allen Datenbankpartitionen zugänglich. Die folgenden Offline-Backup-Images sind in /dev3/backup verfügbar:

```
wsdb.0.db2inst1.NODE0000.CATN0000.20020331234149.001
wsdb.0.db2inst1.NODE0001.CATN0000.20020331234427.001
wsdb.0.db2inst1.NODE0002.CATN0000.20020331234828.001
wsdb.0.db2inst1.NODE0003.CATN0000.20020331235235.001
```

Um zuerst die Katalogpartition und anschließend alle anderen Partitionen der Datenbank WSDB aus dem Verzeichnis /dev3/backup wiederherzustellen, setzen Sie den folgenden Befehl in einer der Datenbankpartitionen ab:

```
db2_all '<<+0< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331234149
INTO wsdb REPLACE EXISTING'
db2_all '<<+1< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331234427
INTO wsdb REPLACE EXISTING'
db2_all '<<+2< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331234828
INTO wsdb REPLACE EXISTING'
db2_all '<<+3< db2 RESTORE DATABASE wsdb FROM /dev3/backup
TAKEN AT 20020331235235
INTO wsdb REPLACE EXISTING'
```

Das Dienstprogramm **db2_all** setzt den Restorebefehl an jede angegebene Datenbankpartition ab. Beim Ausführen einer Restoreoperation mit **db2_all** sollten Sie unbedingt **REPLACE EXISTING** und/oder **WITHOUT PROMPTING** angeben. Andernfalls scheint die Anwendung zu blockieren, wenn Eingabeaufforderungen ausgegeben werden. Dies liegt daran, dass **db2_all** keine Benutzereingabeaufforderung unterstützt.

2. Es folgt ein typisches Beispielszenario für den umgeleiteten Restore einer Datenbank mit dem Aliasnamen MEINEDB:

- a. Setzen Sie einen Befehl **RESTORE DATABASE** mit der Option **REDIRECT** ab.

```
restore db meinedb replace existing redirect
```

Nach der erfolgreichen Ausführung von Schritt 1 und vor der Vollendung von Schritt 3 kann die Restoreoperation durch Absetzen des folgenden Befehls abgebrochen werden:

```
restore db meinedb abort
```

- b. Setzen Sie einen Befehl **SET TABLESPACE CONTAINERS** für jeden Tabellenbereich ab, dessen Container erneut definiert werden müssen. Beispiel:

```
set tablespace containers for 5 using
(file 'f:\ts3con1' 20000, file 'f:\ts3con2' 20000)
```

Setzen Sie den Befehl **LIST TABLESPACE CONTAINERS** ab, um sicherzustellen, dass es sich bei den Containern der wiederhergestellten Datenbank um die in diesem Schritt angegebenen Container handelt.

- c. Nach der erfolgreichen Durchführung der Schritte 1 und 2 setzen Sie den folgenden Befehl ab:

```
restore db meinedb continue
```

Dies ist der letzte Schritt des umgeleiteten Restores.

- d. Falls Schritt 3 fehlschlägt oder die Restoreoperation abgebrochen wurde, kann der umgeleitete Restore erneut gestartet werden, indem Sie wieder bei Schritt 1 beginnen.
3. Im Folgenden sehen Sie ein Beispiel für eine wöchentliche inkrementelle Backup-Strategie für eine wiederherstellbare Datenbank. Diese Strategie umfasst ein wöchentliches Gesamtbackup der Datenbank, ein tägliches nicht kumulatives Backup (Delta-Backup) sowie ein kumulatives Backup (inkrementell) in der Wochenmitte:

```
(So) backup db meinedb use tsm
(Mo) backup db meinedb online incremental delta use tsm
(Di) backup db meinedb online incremental delta use tsm
(Mi) backup db meinedb online incremental use tsm
(Do) backup db meinedb online incremental delta use tsm
(Fr) backup db meinedb online incremental delta use tsm
(Sa) backup db meinedb online incremental use tsm
```

Setzen Sie für eine automatische Wiederherstellung der Datenbank aus den am Freitagmorgen erstellten Images den folgenden Befehl ab:

```
restore db meinedb incremental automatic taken at (Fr)
```

Setzen Sie für eine manuelle Wiederherstellung der Datenbank aus den am Freitagmorgen erstellten Images den folgenden Befehl ab:

```
restore db meinedb incremental taken at (Fr)
restore db meinedb incremental taken at (So)
restore db meinedb incremental taken at (Mi)
restore db meinedb incremental taken at (Do)
restore db meinedb incremental taken at (Fr)
```

4. Geben Sie Folgendes ein, um ein Backup-Image, das Protokolle einschließt, für die Übertragung auf ein fernes System zu erstellen:

```
backup db sample online to /dev3/backup include logs
```

Um dieses Backup-Image wiederherzustellen, geben Sie einen LOGTARGET-Pfad an, und geben Sie diesen Pfad innerhalb von **ROLLFORWARD** an:

```
restore db sample from /dev3/backup logtarget /dev3/logs
rollforward db sample to end of logs and stop overflow log path /dev3/logs
```

5. Geben Sie Folgendes ein, um nur die Protokolldateien aus einem Backup-Image, das Protokolle einschließt, abzurufen:

```
restore db sample logs from /dev3/backup logtarget /dev3/logs
```

6. Mit den Schlüsselwörtern **USE TSM OPTIONS** können die TSM-Informationen angegeben werden, die für die Restoreoperation verwendet werden sollen. Lassen Sie auf Windows-Plattformen die Option **-fromowner** weg.

- Angabe einer Zeichenfolge mit Begrenzer:

```
restore db sample use TSM options '"-fromnode=bar -fromowner=dmcinnis"'
```

- Angabe eines vollständig qualifizierten Dateinamens:

```
restore db sample use TSM options @/u/dmcinnis/meineoptionen.txt
```

Die Datei `meineoptionen.txt` enthält die folgenden Informationen:
-fromnode=bar -fromowner=dmcinnis

7. Das folgende Beispiel enthält eine Restoreoperation für eine Mehrpartitionsdatenbank mit dynamischem Speicher und neuen Speicherpfaden. Die Datenbank wurde ursprünglich mit einem Speicherpfad (/meinPfad0:) erstellt.

- Setzen Sie in der Katalogpartition den folgenden Befehl ab: `restore db meinedb on /meinPfad1,/meinPfad2`

- Setzen Sie in allen Nichtkatalogpartitionen den folgenden Befehl ab: restore db meinedb
8. Eine Scriptausgabe auf einer Datenbank ohne dynamischen Speicher für den Befehl

```
restore db sample from /home/jseifert/backups taken at 20050301100417 redirect
generate script SAMPLE_NODE0000.clp
```

würde wie folgt aussehen:

```
-- *****
-- ** Automatisch erstelltes Umleitungs-Restore-Script
-- *****
UPDATE COMMAND OPTIONS USING S ON Z ON SAMPLE_NODE0000.out V ON;
SET CLIENT ATTACH_DBPARTITIONNUM 0;
SET CLIENT CONNECT_DBPARTITIONNUM 0;
-- *****
-- ** Umgeleiteten Restore initialisieren
-- *****
RESTORE DATABASE SAMPLE
-- USER '<benutzername>'
-- USING '<kennwort>'
FROM '/home/jseifert/backups'
TAKEN AT 20050301100417
-- DBPATH ON '<zielverzeichnis>'
  INTO SAMPLE
-- NEWLOGPATH '/home/jseifert/jseifert/NODE0000/SQL00001/SQLQDIR/'
  -- WITH <pufferanzahl> BUFFERS
  -- BUFFER <puffergröße>
  -- REPLACE HISTORY FILE
  -- REPLACE EXISTING
REDIRECT
  -- PARALLELISM <n>
  -- WITHOUT ROLLING FORWARD
  -- WITHOUT PROMPTING
;
-- *****
-- ** Tabellenbereichsdefinition
-- *****
-- *****
-- ** Tablespace name = SYSCATSPACE
-- ** Tablespace ID = 0
-- ** Tablespace Type = System managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = No
-- ** Total number of pages = 5572
-- *****
SET TABLESPACE CONTAINERS FOR 0
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  PATH 'SQLT0000.0'
);
-- *****
-- ** Tablespace name = TEMPSPACE1
-- ** Tablespace ID = 1
-- ** Tablespace Type = System managed space
-- ** Tablespace Content Type = System Temporary data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = No
-- ** Total number of pages = 0
-- *****
SET TABLESPACE CONTAINERS FOR 1
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
```

```

    PATH 'SQLT0001.0'
);
-- *****
-- ** Tablespace name = USERSPACE1
-- ** Tablespace ID = 2
-- ** Tablespace Type = System managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = No
-- ** Total number of pages = 1
-- *****
SET TABLESPACE CONTAINERS FOR 2
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
    PATH 'SQLT0002.0'
);
-- *****
-- ** Tablespace name = DMS
-- ** Tablespace ID = 3
-- ** Tablespace Type = Database managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = No
-- ** Auto-resize enabled = No
-- ** Total number of pages = 2000
-- ** Number of usable pages = 1960
-- ** High water mark (pages) = 96
-- *****
SET TABLESPACE CONTAINERS FOR 3
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
    FILE /tmp/dms1 1000
    , FILE /tmp/dms2 1000
);
-- *****
-- ** Tablespace name = RAW
-- ** Tablespace ID = 4
-- ** Tablespace Type = Database managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = No
-- ** Auto-resize enabled = No
-- ** Total number of pages = 2000
-- ** Number of usable pages = 1960
-- ** High water mark (pages) = 96
-- *****
SET TABLESPACE CONTAINERS FOR 4
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
    DEVICE '/dev/hdb1' 1000
    , DEVICE '/dev/hdb2' 1000
);
-- *****
-- ** Umgeleiteten Restore starten
-- *****
RESTORE DATABASE SAMPLE CONTINUE;--
-- ** end of file
-- *****

```

9. Eine Scriptausgabe auf einer Datenbank mit dynamischem Speicher für den Befehl

```

restore db test from /home/jseifert/backups taken at 20050304090733 redirect
generate script TEST_NODE0000.clp

```

würde wie folgt aussehen:

```
-- *****
-- ** Automatisch erstelltes Umleitungs-Restore-Script
-- *****
UPDATE COMMAND OPTIONS USING S ON Z ON TEST_NODE0000.out V ON;
SET CLIENT ATTACH_DBPARTITIONNUM 0;
SET CLIENT CONNECT_DBPARTITIONNUM 0;
-- *****
-- ** Umgeleiteten Restore initialisieren
-- *****
RESTORE DATABASE TEST
-- USER '<benutzername>'
-- USING '<kennwort>'
FROM '/home/jseifert/backups'
TAKEN AT 20050304090733
ON '/home/jseifert'
-- DBPATH ON <zielverzeichnis>
INTO TEST
-- NEWLOGPATH '/home/jseifert/jseifert/NODE0000/SQL00002/SQLLOGDIR/'
-- WITH <pufferanzahl> BUFFERS
-- BUFFER <puffergröße>
-- REPLACE HISTORY FILE
-- REPLACE EXISTING
REDIRECT
-- PARALLELISM <n>
-- WITHOUT ROLLING FORWARD
-- WITHOUT PROMPTING
;
-- *****
-- ** Tabellenbereichsdefinition
-- *****
-- *****
-- ** Tablespace name = SYSCATSPACE
-- ** Tablespace ID = 0
-- ** Tablespace Type = Database managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 4
-- ** Using automatic storage = Yes
-- ** Auto-resize enabled = Yes
-- ** Total number of pages = 6144
-- ** Number of usable pages = 6140
-- ** High water mark (pages) = 5968
-- *****
-- *****
-- ** Tablespace name = TEMPSPACE1
-- ** Tablespace ID = 1
-- ** Tablespace Type = System managed space
-- ** Tablespace Content Type = System Temporary data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = Yes
-- ** Total number of pages = 0
-- *****
-- *****
-- ** Tablespace name = USERSPACE1
-- ** Tablespace ID = 2
-- ** Tablespace Type = Database managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = Yes
-- ** Auto-resize enabled = Yes
-- ** Total number of pages = 256
-- ** Number of usable pages = 224
-- ** High water mark (pages) = 96
-- *****
```

```

-- *****
-- ** Tablespace name = DMS
-- ** Tablespace ID = 3
-- ** Tablespace Type = Database managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = No
-- ** Auto-resize enabled = No
-- ** Total number of pages = 2000
-- ** Number of usable pages = 1960
-- ** High water mark (pages) = 96
-- *****
SET TABLESPACE CONTAINERS FOR 3
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
    FILE '/tmp/dms1' 1000
    , FILE '/tmp/dms2' 1000
);
-- *****
-- ** Tablespace name = RAW
-- ** Tablespace ID = 4
-- ** Tablespace Type = Database managed space
-- ** Tablespace Content Type = Any data
-- ** Tablespace Page size (bytes) = 4096
-- ** Tablespace Extent size (pages) = 32
-- ** Using automatic storage = No
-- ** Auto-resize enabled = No
-- ** Total number of pages = 2000
-- ** Number of usable pages = 1960
-- ** High water mark (pages) = 96
-- *****
SET TABLESPACE CONTAINERS FOR 4
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
    DEVICE '/dev/hdb1' 1000
    , DEVICE '/dev/hdb2' 1000
);
-- *****
-- ** Umgeleiteten Restore starten
-- *****
RESTORE DATABASE TEST CONTINUE;
-- *****
-- ** end of file
-- *****

```

10. Es folgen Beispiele für den Befehl **RESTORE DB** mit der Option **SNAPSHOT**:
 Protokollverzeichnisdatenträger aus dem Momentaufnahme-Image ohne Benutzereingabeaufforderung wiederherstellen.
- ```
db2 restore db sample use snapshot LOGTARGET INCLUDE without prompting
```
- Keine Protokollverzeichnisdatenträger wiederherstellen und keine Benutzereingabeaufforderung verwenden.
- ```
db2 restore db sample use snapshot LOGTARGET EXCLUDE without prompting
```
- Keine Protokollverzeichnisdatenträger wiederherstellen und keine Benutzereingabeaufforderung verwenden. Wenn **LOGTARGET** nicht angegeben ist, wird der Standardwert **LOGTARGET EXCLUDE** verwendet.
- ```
db2 restore db sample use snapshot without prompting
```
- Beim Wiederherstellen des Momentaufnahme-Images, das in Konflikt stehende Protokollverzeichnisse enthält, dürfen vorhandene Protokollverzeichnisse in der aktuellen Datenbank ohne Benutzereingabeaufforderung überschrieben und ersetzt werden.
- ```
db2 restore db sample use snapshot LOGTARGET EXCLUDE FORCE without prompting
```

Beim Wiederherstellen des Momentaufnahme-Images, das in Konflikt stehende Protokollverzeichnisse enthält, dürfen vorhandene Protokollverzeichnisse in der aktuellen Datenbank ohne Benutzereingabeaufforderung überschrieben und ersetzt werden.

```
db2 restore db sample use snapshot LOGTARGET INCLUDE FORCE without prompting
```

Wenn mit dem Parameter **AT DBPARTITIONNUM** eine Datenbankpartition neu erstellt wird, die (aufgrund einer Beschädigung) gelöscht worden war, befindet sich die Datenbank in dieser Datenbankpartition im Status 'Restore anstehend'. Nach der Neuerstellung der Datenbankpartition muss die Datenbank in dieser Datenbankpartition sofort neu erstellt werden.

Hinweise zur Verwendung

- Ein Befehl **RESTORE DATABASE** im Format `db2 restore db <name>` führt einen vollständigen Datenbankrestore mit einem Datenbankimage durch und eine Restoreoperation für die Tabellenbereiche, die in einem Tabellenbereichs-Image gefunden werden. Ein Befehl **RESTORE DATABASE** im Format `db2 restore db <name> tabellenbereich` führt eine Restoreoperation für die Tabellenbereiche durch, die in dem Image gefunden werden. Wird in einem solchen Befehl zusätzlich eine Liste mit Tabellenbereichen angegeben, werden die explizit aufgelisteten Tabellenbereiche wiederhergestellt.
- Im Anschluss an die Restoreoperation eines Online-Backups müssen Sie eine aktualisierende Recovery ausführen.
- Mit dem Parameter **OPTIONS** können Sie Restoreoperationen in TSM-Umgebungen aktivieren, die Proxy-Knoten unterstützen. Weitere Informationen finden Sie im Thema „Konfigurieren eines Tivoli Storage Manager-Clients“.
- Wenn ein Backup-Image komprimiert ist, erkennt das DB2-Datenbanksystem die Komprimierung und dekomprimiert die Daten automatisch, bevor die Restoreoperation ausgeführt wird. Wenn in einer db2Restore-API eine Bibliothek angegeben ist, wird sie zum Dekomprimieren der Daten verwendet. Andernfalls wird geprüft, ob in dem Backup-Image eine Bibliothek gespeichert ist. Ist dies der Fall, wird die Bibliothek verwendet. Wenn in dem Backup-Image keine Bibliothek gespeichert ist, können die Daten nicht dekomprimiert werden, und die Restoreoperation schlägt fehl.
- Soll die Komprimierungsbibliothek aus einem Backup-Image wiederhergestellt werden (entweder explizit durch Angeben der Option **COMPRESSION LIBRARY** oder implizit durch einen regulären Restore aus einem komprimierten Backup), muss die Restoreoperation auf derselben Plattform und unter demselben Betriebssystem erfolgen wie die Erstellung des Backups. Wurde das Backup nicht auf derselben Plattform erstellt, auf der die Restoreoperation erfolgt, schlägt die Restoreoperation fehl, auch wenn das DB2-Datenbanksystem normalerweise plattformübergreifende Restoreoperationen unterstützt, an denen zwei Systeme beteiligt sind.
- Ein gesicherter SMS-Tabellenbereich kann nur in einem SMS-Tabellenbereich wiederhergestellt werden. Er kann nicht in einem DMS-Tabellenbereich wiederhergestellt werden oder umgekehrt.
- Zum Wiederherstellen von Protokolldateien aus dem Backup-Image, in dem sie enthalten sind, muss die Option **LOGTARGET** mit dem vollständig qualifizierten und gültigen Pfad angegeben werden, der auf dem DB2-Server vorhanden ist. Wenn diese Bedingungen erfüllt sind, schreibt das Restoredienstprogramm die Protokolldateien aus dem Image in den Zielpfad. Wird **LOGTARGET** bei der Wiederherstellung eines Backup-Images angegeben, das keine Protokolle enthält, gibt die Restoreoperation einen Fehler zurück, bevor versucht wird, Tabellenbe-

reichsdaten wiederherzustellen. Eine Restoreoperation schlägt ebenfalls mit einer Fehlermeldung fehl, wenn ein ungültiger oder schreibgeschützter Protokollziel-pfad angegeben ist.

- Sind im Protokollziel-pfad zu dem Zeitpunkt, an dem der Befehl **RESTORE WDATABASE** abgesetzt wird, bereits Protokoll-dateien vorhanden, wird eine Warnung an den Benutzer zurückgegeben. Diese Warnung wird nicht zurückgegeben, wenn **WITHOUT PROMPTING** angegeben ist.
- Kann bei einer Restoreoperation, in der **LOGTARGET** angegeben ist, keine Protokoll-datei extrahiert werden, schlägt die Restoreoperation fehl und gibt einen Fehler zurück. Trägt eine der aus dem Backup-Image extrahierten Protokoll-dateien denselben Namen wie eine im Protokollziel-pfad vorhandene Datei, schlägt die Restoreoperation fehl, und es wird ein Fehler zurückgegeben. Das Daten-bankrestoredienstprogramm überschreibt keine vorhandenen Protokoll-dateien im Protokollzielverzeichnis.
- Es ist auch möglich, nur die gespeicherten Protokoll-dateien aus einem Backup-Image wiederherzustellen. Um anzugeben, dass nur die Protokoll-dateien wiederhergestellt werden sollen, geben Sie zusätzlich zum Protokollziel-pfad die Option **LOGS** an. Das Angeben der Option **LOGS** ohne einen Protokollziel-pfad führt zu einem Fehler. Wenn beim Wiederherstellen von Protokoll-dateien in dieser Betriebs-art Probleme auftreten, wird die Restoreoperation sofort abgebrochen und ein Fehler zurückgegeben.
- Bei einer automatischen inkrementellen Restoreoperation werden nur die Protokoll-dateien aus dem Backup-Image abgerufen, die sich im Zielimage der Restoreoperation befinden. Protokolle, die sich in temporären Images befinden, auf die während des inkrementellen Restores verwiesen wird, werden nicht aus diesen temporären Backup-Images extrahiert. Bei einer manuellen inkrementellen Restoreoperation sollte der Protokollziel-pfad nur in dem endgültigen Restorebe-fehl angegeben werden, der abgesetzt werden soll.
- Offline-Backups der gesamten Datenbank sowie inkrementelle Offline-Daten-bankbackups können im Gegensatz zu Online-Backups in einer späteren Daten-bankversion wiederhergestellt werden. Bei Mehrpartitionsdatenbanken muss die Katalogpartition zuerst einzeln wiederhergestellt werden. Anschließend können die übrigen Datenbankpartitionen parallel oder seriell wiederhergestellt werden. Das von der Restoreoperation durchgeführte implizite Datenbankupgrade kann jedoch fehlschlagen. Bei einer Mehrpartitionsdatenbank kann sie auf einer oder mehreren Datenbankpartitionen fehlschlagen. In diesem Fall können Sie nach dem Befehl **RESTORE DATABASE** den Befehl **UPGRADE DATABASE** einmalig von der Katalogpartition absetzen, um die Datenbank erfolgreich aufzurüsten.
- Die Option **TRANSPORT** wird nur dann unterstützt, wenn die Codepages von Cli-ent und Datenbank gleich sind.

Momentaufnahmerestore

Wie bei einem traditionellen Restore (nicht aus einer Momentaufnahme), werden auch beim Restore eines Momentaufnahmebackup-Images standardmäßig keine Protokollverzeichnisse wiederhergestellt (**LOGTARGET EXCLUDE**).

Stellt der DB2-Datenbankmanager fest, dass die Gruppen-ID eines Protokollver-zeichnisses von anderen wiederherzustellenden Pfaden gemeinsam genutzt wird, wird ein Fehler zurückgegeben. In diesem Fall muss **LOGTARGET INCLUDE** oder **LOGTARGET INCLUDE FORCE** angegeben werden, da die Protokollverzeichnisse Teil der Restoreoperation sein müssen.

Der DB2-Datenbankmanager versucht mit allen Mitteln, vorhandene Protokollverzeichnisse (Primär-, Spiegel- und Überlaufprotokolle) zu sichern, bevor die Restoreoperation der Pfade aus dem Backup-Image ausgeführt wird.

Sollen die Protokollverzeichnisse wiederhergestellt werden und erkennt der DB2-Datenbankmanager, dass die vorhandenen Protokollverzeichnisse auf dem Datenträger in Konflikt mit den Protokollverzeichnissen im Backup-Image stehen, meldet der DB2-Datenbankmanager einen Fehler. Wenn Sie in einem solchen Fall **LOGTARGET INCLUDE FORCE** angegeben haben, wird dieser Fehler unterdrückt, und die Protokollverzeichnisse aus dem Image werden wiederhergestellt (dabei werden bereits vorhandene Protokolle gelöscht).

Es gibt einen Sonderfall, in dem die Option **LOGTARGET EXCLUDE** angegeben ist und sich ein Protokollverzeichnispfad im Datenbankverzeichnis (beispielsweise /NODExxxx/SQLxxxxx/SQLOGDIR/) befindet. In diesem Fall würde das Protokollverzeichnis bei der Restoreoperation dennoch überschrieben, während der Datenbankpfad mit allen untergeordneten Elementen wiederhergestellt würde. Wenn der DB2-Datenbankmanager dieses Szenario feststellt und dieses Protokollverzeichnis Protokolldateien enthält, wird ein Fehler gemeldet. Wenn Sie **LOGTARGET EXCLUDE FORCE** angegeben haben, wird dieser Fehler unterdrückt, und die Protokollverzeichnisse aus dem Backup-Image überschreiben die in Konflikt stehenden Protokollverzeichnisse auf dem Datenträger.

Transport von Tabellenbereichen und Schemas

Es muss die vollständige Liste der Tabellenbereiche und Schemas angegeben werden.

Die Zieldatenbank muss zum Zeitpunkt des Transports aktiv sein.

Wenn ein Online-Backup-Image verwendet wird, wird die Bereitstellungsdatenbank zum Endpunkt des Backups aktualisierend wiederhergestellt. Wird ein Offline-Backup-Image verwendet, findet keine aktualisierende Wiederherstellung durchgeführt.

Eine Bereitstellungsdatenbank, die aus dem Systemkatalog-Tabellenbereich des Backup-Images besteht, wird unter dem durch den Datenbankparameter **dftdbpath** angegebenen Pfad erstellt. Diese Datenbank wird nach Abschluss des Befehls **RESTORE DATABASE** gelöscht. Die Bereitstellungsdatenbank wird benötigt, um die DDL (Data Definition Language) zu extrahieren, die zur erneuten Generierung der Objekte in der transportierten Tabelle verwendet wird.

Beim Transport von Tabellenbereichen versucht der DB2-Datenbankmanager, den ersten verfügbaren Pufferpool mit der übereinstimmenden Seitengröße dem Tabellenbereich zuzuweisen, der transportiert wird. Wenn die Zieldatenbank über keine Pufferpools mit entsprechender Seitengröße für transportierte Tabellenbereiche verfügt, wird möglicherweise ein verdeckter Pufferpool zugewiesen. Verdeckte Pufferpools sind temporäre Platzhalter für transportierte Tabellenbereiche. Sie können die den transportierten Tabellenbereichen zugewiesenen Pufferpools nach Abschluss des Transports prüfen. Zum Aktualisieren der Pufferpools können Sie den Befehl **ALTER TABLESPACE** absetzen.

Wenn die aktualisierende Recovery einer Datenbank einen Protokolleintrag für ein transportiertes Tabellenbereichsschema findet, wird der entsprechende Tabellenbereich offline gesetzt und in den Status "Löschen anstehend" versetzt. Der Grund dafür ist, dass die Datenbank nicht über vollständige Protokolle zu transportierten

Tabellenbereichen verfügt, um diese sowie deren Inhalte erneut zu erstellen. Sie können nach Beendigung des Transports einen vollständiges Backup der Zieldatenbank durchführen, damit bei der nachfolgenden aktualisierenden Recoveryoperation der Punkt des Schematransports im Protokollstrom nicht übergangen wird.

Hohe Verfügbarkeit durch Unterstützung der ausgesetzten E/A und der Onlineteilung einer Spiegeldatenbank

Durch die Unterstützung der ausgesetzten E/A von IBM Data Server können Sie gespiegelte Kopien Ihrer Primärdatenbank teilen, ohne die Datenbank in den Offlinestatus versetzen zu müssen. Sie können diese Funktion verwenden, um rasch eine Bereitschaftsdatenbank zu erstellen, die die Arbeit übernimmt, wenn die Primärdatenbank fehlschlägt.

Die Plattenspiegelung ist ein Vorgang, bei dem Daten gleichzeitig auf zwei unterschiedliche Festplatten geschrieben werden. Die eine Kopie der Daten wird als der "Spiegel" der anderen Kopie bezeichnet. Das Teilen einer solchen Spiegelung bedeutet, dass die beiden Kopien voneinander getrennt werden.

Sie können mit der Plattenspiegelung eine Kopie Ihrer Primärdatenbank verwalten. Sie können die IBM Data Server-Funktionalität der ausgesetzten E/A dazu verwenden, die primären und sekundären gespiegelten Kopien der Datenbank zu teilen, ohne die Datenbank in den Offlinestatus versetzen zu müssen. Sobald die primären und sekundären Kopien der Datenbank geteilt sind, kann die sekundäre Datenbank die Operationen übernehmen, falls die Primärdatenbank ausfällt.

Wenn Sie eine große Datenbank lieber nicht mit dem Backup-Dienstprogramm von IBM Data Server sichern möchten, können Sie Kopien von einem Spiegelimage erstellen, indem Sie dazu die ausgesetzte E/A und die Funktion zur Erstellung einer Spiegeldatenbank verwenden. Außerdem erreichen Sie mit dieser Methode Folgendes:

- Sie vermeiden hohen Backup-Aufwand auf der Produktionsmaschine.
- Sie verfügen über eine schnelle Methode, Systeme zu klonen.
- Sie verfügen über eine schnelle Implementierung der Funktionsübernahme aus dem Bereitschaftsmodus. Ein einleitender Restore findet nicht statt und sollte eine aktualisierende Recovery sich als zu langsam erweisen oder sollten Fehler auftreten, ist die erneute Initialisierung sehr schnell.

Mit dem Befehl **db2inidb** wird die geteilte Spiegeldatenbank initialisiert, sodass Sie sie wie folgt einsetzen können:

- Als Klondatenbank
- Als Bereitschaftsdatenbank
- Als Backup-Image

Dieser Befehl kann nur für eine geteilte Spiegeldatenbank abgesetzt werden, und die geteilte Spiegeldatenbank kann erst nach Ausführung des Befehls verwendet werden.

In einer Umgebung mit partitionierten Datenbanken müssen Sie E/A-Schreibvorgänge nicht auf allen Datenbankpartitionen gleichzeitig aussetzen. Sie können die E/A für eine Untergruppe aus einer oder mehreren Datenbankpartitionen aussetzen, um zum Ausführen von Offline-Backups geteilte Spiegeldatenbanken zu erstellen. Wenn in dieser Untergruppe die Katalogpartition enthalten ist, muss das Aussetzen für diese Datenbankpartition zuletzt erfolgen.

Befehlsparameter

aliasname_der_datenbank

Gibt den Aliasnamen der zu initialisierenden Datenbank an.

SNAPSHOT

Gibt an, dass die Spiegeldatenbank als Klon der Primärdatenbank initialisiert werden soll.

STANDBY

Gibt an, dass die Datenbank in den Status "Aktualisierende Recovery anstehend" versetzt wird. Neue Protokolle aus der Primärdatenbank können abgerufen und auf die Bereitschaftsdatenbank angewendet werden. Anschließend kann die Bereitschaftsdatenbank anstelle der Primärdatenbank verwendet werden, falls diese ausfällt.

ACS

Gibt an, dass der Befehl **db2inidb** für eine ACS-Momentaufnahmekopie der Datenbank verwendet werden muss, um die Aktion **STANDBY** durchzuführen. Diese Option ist erforderlich, da der Befehl **db2inidb** nur für Momentaufnahmen von geteilten Spiegeldatenbanken abgesetzt werden kann, die mit dem Befehl **SET WRITE SUSPEND | RESUME** erstellt wurden.

Die Verwendung der Option **ACS STANDBY** sorgt dafür, dass die ACS-Momentaufnahmekopie in den Status 'Aktualisierende Recovery anstehend' versetzt wird, sodass der DB2-Befehl **BACKUP** erfolgreich für das Momentaufnahme-Image abgesetzt werden kann. Ohne diese Option wird bei jedem Versuch, eine Verbindung zum Momentaufnahme-Image herzustellen, diese Kopie in den Status 'Restore anstehend' versetzt, wodurch sie bei zukünftigen Recoveryoperationen nicht mehr als Backup-Kopie verwendet werden kann.

Diese Funktion wurde speziell für das Einrichten von Schnittstellen mit Speichermanagern wie IBM Tivoli Storage FlashCopy Manager mit dem Zweck eingeführt, ein ausgelagertes DB2-Backup zu erstellen, das auf einer ACS-Momentaufnahme basiert. Die Verwendung dieser Option zu anderen Zwecken, beispielsweise zum Bereitstellen oder Ändern des Inhalts einer ACS-Momentaufnahmekopie oder zur Generierung eines DB2-Backups, kann zu einem späteren Zeitpunkt zu einem nicht definierten Verhalten führen.

MIRROR Gibt an, dass die Spiegeldatenbank als Backup-Image zum Wiederherstellen der Primärdatenbank verwendet werden soll.

RELOCATE USING *konfigurationsdatei*

Gibt an, dass die Datenbankdateien anhand der Informationen aus der angegebenen *konfigurationsdatei* verlagert werden, bevor die Datenbank als Momentaufnahme-, Bereitschafts- oder Spiegeldatenbank initialisiert wird. Das Format für die *konfigurationsdatei* ist in „db2relocatedb - Verlagern einer Datenbank“ auf Seite 421 beschrieben.

Hinweise zur Verwendung

Führen Sie die Operation `db2 connect to aliasname_der_datenbank` nicht vor dem Absetzen des Befehls `db2inidb aliasname_der_datenbank as mirror` aus. Bei dem Versuch, eine Verbindung zu einer geteilten Spiegeldatenbank herzustellen, bevor diese initialisiert wurde, werden die für die aktualisierende Recovery benötigten Protokolldateien gelöscht. Beim Verbindungsaufbau wird Ihre Datenbank wieder in den Status versetzt, den sie hatte, als sie ausgesetzt wurde. Wurde die Datenbank

bei ihrer Aussetzung als konsistent markiert, geht das DB2-Datenbanksystem davon aus, dass keine Recovery nach Systemabsturz erforderlich ist, und löscht den Inhalt der Protokolldateien, damit sie später wieder verwendet werden können. Wenn der Inhalt der Protokolldateien gelöscht wurde, wird bei dem Versuch einer aktualisierenden Recovery die Fehlermeldung SQL4970N zurückgegeben.

In einer Umgebung mit partitionierten Datenbanken muss der Befehl **db2inidb** auf jeder Datenbankpartition ausgeführt werden, bevor der geteilte Spiegel irgendeiner Datenbankpartition verwendet werden kann. **db2inidb** kann auf allen Datenbankpartitionen gleichzeitig ausgeführt werden, wenn der Befehl **db2_all** verwendet wird.

Wenn Sie jedoch die Option **RELOCATE USING** verwenden, können Sie den Befehl **db2_all** nicht verwenden, um **db2inidb** auf allen Partitionen gleichzeitig auszuführen. Für jede Partition muss eine eigene Konfigurationsdatei angegeben werden, die den Wert NODENUM der zu ändernden Datenbankpartition enthält. Wenn beispielsweise der Name der Datenbank geändert wird, sind alle Datenbankpartitionen von dieser Änderung betroffen, und der Befehl **db2relocatedb** muss auf jeder Datenbankpartition mit einer eigenen Konfigurationsdatei ausgeführt werden. Wenn Container versetzt werden, die zu einer einzelnen Datenbankpartition gehören, muss der Befehl **db2relocatedb** nur einmal auf dieser Datenbankpartition ausgeführt werden.

Wird der Parameter **RELOCATE USING** *konfigurationsdatei* angegeben und die Datenbank erfolgreich verlagert, dann wird die angegebene *konfigurationsdatei* in das Datenbankverzeichnis kopiert und in *db2path.cfg* umbenannt. Bei einer späteren Recovery nach Systemabsturz oder aktualisierenden Recovery wird diese Datei während der Verarbeitung von Protokolldateien verwendet, um Containerpfade umzubenenen.

Beim Initialisieren einer Klondatenbank wird die angegebene *konfigurationsdatei* nach Beendigung einer Recovery nach Systemabsturz automatisch aus dem Datenbankverzeichnis entfernt.

Beim Initialisieren einer Bereitschafts- oder Spiegeldatenbank wird die angegebene *konfigurationsdatei* automatisch aus dem Datenbankverzeichnis entfernt, nachdem eine aktualisierende Recovery beendet oder abgebrochen wurde. Neue Containerpfade können zur Datei *db2path.cfg* hinzugefügt werden, nachdem der Befehl **db2inidb** ausgeführt wurde. Dies wird erforderlich, wenn CREATE- oder ALTER TABLESPACE-Operationen an der ursprünglichen Datenbank ausgeführt werden und in der Bereitschaftsdatenbank andere Pfade verwendet werden müssen.

Verwenden Sie beim Durchführen der Initialisierung einer geteilten Spiegeldatenbank aus einer HADR-Primär- oder Bereitschaftsdatenbank den Parameter **STANDBY**, wenn eine der folgenden Situationen vorliegt:

- Die neue Datenbank wird in einem HADR-Paar verwendet, und die HADR-Konfigurationseinstellungen des neuen Paares stimmen nicht mit den Einstellungen des ursprünglichen Paares überein.
- Die Datenbank wird als eigenständige Datenbank initialisiert.

db2relocatedb - Verlagern einer Datenbank

Dieser Befehl benennt eine Datenbank um oder verlagert eine Datenbank oder einen Teil einer Datenbank (z. B. den Container und das Protokollverzeichnis), wie

in der durch den Benutzer bereitgestellten Konfigurationsdatei angegeben ist. Dieses Tool nimmt die erforderlichen Änderungen an den Unterstützungsdateien der DB2-Instanz und der Datenbank vor.

Die Zieldatenbank muss offline sein, damit der Befehl **db2relocatedb** zum Ändern der Steuerdateien und Metadaten der Zieldatenbank ausgeführt werden kann.

Die mit dem Befehl **db2relocatedb** an den Dateien und Steuerstrukturen einer Datenbank vorgenommenen Änderungen werden nicht protokolliert und sind daher nicht wiederherstellbar. Es empfiehlt sich, nach der Ausführung des Befehls für eine Datenbank ein Gesamtbackup auszuführen, insbesondere wenn die Datenbank wiederherstellbar ist und Protokolldateien beibehalten werden.

Berechtigung

Keine

Befehlssyntax

```
►► db2relocatedb -f konfigurationsdateiname ◀◀
```

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=alteInstanz,neueInstanz
NODENUM=knotenNummer
LOG_DIR=alterVerzPfad,neuerVerzPfad
CONT_PATH=alterBehältPfad1,neuerBehältPfad1
CONT_PATH=alterBehältPfad2,neuerBehältPfad2
...
STORAGE_PATH=alterSpeicherpfad1,neuerSpeicherpfad1
STORAGE_PATH=alterSpeicherpfad2,neuerSpeicherpfad2
...
FAILARCHIVE_PATH=neuerVerzPfad
LOGARCHMETH1=neuerVerzPfad
LOGARCHMETH2=neuerVerzPfad
MIRRORLOG_PATH=neuerVerzPfad
OVERFLOWLOG_PATH=neuerVerzPfad
...
```

Dabei gilt Folgendes:

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 ursprünglichen Pfad der Datenbank an, die verlagert wird. Wenn der Datenbankpfad geändert wird, müssen der alte und der neue Pfad angegeben werden. Dies ist ein erforderliches Feld.

INSTANCE

Gibt die Instanz an, in dem die Datenbank vorhanden ist. Wenn die Datenbank in eine neue Instanz versetzt wird, müssen die alte und die neue Instanz 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 sowohl der alte als auch 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 Tabellenbereichscontainer an. Der alte und der neue Containerpfad müssen angegeben werden. Mehrere **CONT_PATH**-Zeilen können angegeben werden, wenn mehrere Containerpfadänderungen erforderlich sind. Diese Spezifikation ist optional, wenn sich der Containerpfad unter dem Datenbankpfad befindet. In diesem Fall werden die Pfade automatisch aktualisiert. Wenn Sie Änderungen an mehr als einem Container 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.

STORAGE_PATH

Diese Option gilt nur für Datenbanken mit aktivierter Funktion für dynamischen Speicher. Sie gibt eine Änderung an der Speicherposition eines der Speicherpfade für die Datenbank an. Sowohl der alte als auch der neue Speicherpfad müssen angegeben werden. Mehrere **STORAGE_PATH**-Zeilen können angegeben werden, wenn mehrere Speicherpfadänderungen erforderlich sind.

FAILARCHIVE_PATH

Gibt eine neue Speicherposition zum Archivieren der Protokolldateien an, wenn der Datenbankmanager die Protokolldateien weder an der primären noch an der sekundären Archivposition archivieren kann. Geben Sie dieses Feld nur an, wenn in der zu verlagernden Datenbank der Konfigurationsparameter **failarchpath** definiert ist.

LOGARCHMETH1

Gibt eine neue primäre Archivposition an. Geben Sie dieses Feld nur an, wenn in der zu verlagernden Datenbank der Konfigurationsparameter **logarchmeth1** definiert ist.

LOGARCHMETH2

Gibt eine neue sekundäre Archivposition an. Geben Sie dieses Feld nur an, wenn in der zu verlagernden Datenbank der Konfigurationsparameter **logarchmeth2** definiert ist.

MIRRORLOG_PATH

Gibt eine neue Speicherposition für den Spiegelprotokollpfad an. Diese Zeichenfolge muss auf einen Pfadnamen verweisen, der ein vollständig qualifizierter Pfadname und kein relativer Pfadname

ist. Geben Sie dieses Feld nur an, wenn in der zu verlagernden Datenbank der Konfigurationsparameter **mirrorlogpath** definiert ist.

OVERFLOWLOG_PATH

Gibt eine neue Speicherposition an, an der die Protokolldateien gesucht werden sollen, die für eine ROLLFORWARD-Operation benötigt werden, an der die aktiven Protokolldateien gespeichert werden sollen, die aus dem Archiv abgerufen werden, und an der die Protokolldateien gesucht und gespeichert werden sollen, die für die API db2ReadLog benötigt werden. Geben Sie dieses Feld nur an, wenn in der zu verlagernden Datenbank der Konfigurationsparameter **overflowlogpath** definiert ist.

Leere Zeilen oder Zeilen, die mit einem Kommentarzeichen (#) beginnen, werden ignoriert.

Beispiele

Beispiel 1

Erstellen Sie die folgende Konfigurationsdatei, um den Namen der Datenbank TESTDB in der Instanz 'db2inst1', die 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 der Instanz 'jsmith' im Pfad /dbpath in die Instanz '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 in der Instanz 'inst1' im Pfad /databases/PRODDB vorhanden. Die Speicherposition von zwei Tabellenbereichscontainern muss wie folgt geändert werden:

- Der SMS-Container /data/SMS1 muss nach /DATA/NewSMS1 versetzt werden.
- Der DMS-Container /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 in der Instanz 'db2inst1' vorhanden und wurde im Pfad /databases/TESTDB erstellt. Danach wurden Tabellenbereiche mit den folgenden Containern 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. Die Instanz 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 Container als Teil des Versetzens verlagert werden. (Die ersten drei Container werden relativ zum Datenbankverzeichnis angegeben und die anderen beiden relativ zum Datenbankpfad.) Weil sich diese Container im Datenbankverzeichnis bzw. im Datenbankpfad befinden, müssen sie nicht in der Konfigurationsdatei aufgelistet werden. Wenn die restlichen beiden Container an andere Speicherpositionen auf dem neuen System versetzt werden sollen, müssen sie in der Konfigurationsdatei aufgelistet werden.

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 Datenbankpartitionen auf den Datenbankpartitionsservern 10 und 20. Die Instanz 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
NODENUM=10
```

Auf Datenbankpartitionsserver 20 wird die folgende Konfigurationsdatei verwendet:

```
DB_NAME=TESTDB,SERVDB
DB_PATH=/home/servinst,/databases
INSTANCE=servinst
NODENUM=20
LOG_DIR=/testdb_logdir,/servdb_logdir
```

Beispiel 6

Die Datenbank MAINDB ist in der Instanz maininst im Pfad /home/maininst vorhanden. Die Speicherposition von vier Tabellenbereichscontainern 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.

Eine ähnliche Änderung wird in allen Containern vorgenommen, d. h. /maininst_files/allconts/ wird durch /MAINDB/ ersetzt, sodass ein einziger Eintrag mit dem Platzhalterzeichen verwendet werden kann:

```
DB_NAME=MAINDB
DB_PATH=/home/maininst
INSTANCE=maininst
NODENUM=0      CONT_PATH=/maininst_files/allconts/*, /MAINDB/*
```

Hinweise zur Verwendung

Wenn eine Instanz geändert wird, zu der eine Datenbank gehört, müssen Sie die folgenden Schritte ausführen, bevor Sie diesen Befehl ausführen, um sicherzustellen, dass Änderungen an der Instanz und an den Datenbankunterstützungsdateien vorgenommen werden:

- Wenn eine Datenbank in eine andere Instanz versetzt wird, erstellen Sie die neue Instanz. Die neue Instanz muss dasselbe Release-Level aufweisen wie die Instanz, zu der die Datenbank gehört.
- Wenn die neue Instanz einen anderen Eigentümer als die aktuelle Instanz aufweist, erteilen Sie dem neuen Instanzeigner die entsprechenden Zugriffsrechte.
- Kopieren Sie die Dateien und Einheiten der Datenbank, die kopiert wird, auf das System, auf dem sich die neue Instanz 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 sqlbdir 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 Restore-Operation ist eine Alternative zu **db2relocatedb**.

- Ändern Sie die Berechtigungen der kopierten Dateien/Einheiten, damit der Instanzeigner Eigner dieser Dateien/Einheiten wird.

Wenn Sie eine Datenbank aus einem Datenbankpfad versetzen, in dem sich mehrere Datenbanken befinden, müssen Sie das Verzeichnis `sqlbdir` in diesem Datenbankpfad kopieren, nicht versetzen. Dieses Verzeichnis wird von DB2 weiterhin an der alten Position benötigt, um die nicht versetzten Datenbanken zu finden. Nach dem Kopieren des Verzeichnisses `sqlbdir` an die neue Speicherposition werden die nicht versetzten Datenbanken mit dem Befehl `LIST DB DIRECTORY ON neuerPfad` aufgelistet. Diese Verweise können nicht entfernt werden und es können keine neuen Datenbanken mit solchen Namen in demselben Pfad erstellt werden. In einem anderen Pfad können jedoch Datenbanken mit einem solchen Namen erstellt werden.

Mit dem Befehl **db2relocatedb** können keine vorhandene, von Benutzern erstellte Container für einen Tabellenbereich versetzt werden, der mit der Anweisung `ALTER TABLESPACE MANAGED BY AUTOMATIC STORAGE` für die Verwendung von dynamischem Speicher konvertiert wurde.

Wenn die Instanz geändert wird, muss der Befehl vom neuen Instanzeigner ausgeführt werden.

In einer Umgebung mit partitionierten Datenbanken muss dieses Tool für alle Datenbankpartitionen ausgeführt werden, auf denen Änderungen erforderlich sind. Für jede Datenbankpartition muss eine eigene Konfigurationsdatei angegeben werden, die den Wert `NODENUM` der zu ändernden Datenbankpartition enthält. Wenn beispielsweise der Name der Datenbank geändert wird, sind alle Datenbankpartitionen von dieser Änderung betroffen, und der Befehl **db2relocatedb** muss auf jeder Datenbankpartition mit einer eigenen Konfigurationsdatei ausgeführt werden. Wenn Container versetzt werden, die zu einer einzelnen Datenbankpartition gehören, muss der Befehl **db2relocatedb** nur einmal auf dieser Datenbankpartition ausgeführt werden.

Der Befehl **db2relocatedb** kann nicht zum Verlagern einer Datenbank verwendet werden, für die eine Ladeoperation aktiv ist, oder die auf die Beendigung eines Befehls `LOAD RESTART` oder `LOAD TERMINATE` wartet.

Einschränkung: In einer Umgebung mit partitionierten Datenbanken können Sie keinen vollständigen Knoten verlagern, wenn dieser Knoten eine von zwei oder mehr logischen Partitionen repräsentiert, die sich in derselben Einheit befinden.

db2look - DB2-Statistiktool und DDL-Extraktionstool

Extrahiert die DDL-Anweisungen (DDL = Data Definition Language), die zum Reproduzieren der Datenbankobjekte einer Produktionsdatenbank in einer Testdatenbank erforderlich sind. Der Befehl **db2look** generiert die DDL-Anweisungen nach Objekttyp. Beachten Sie, dass dieser Befehl außer benutzerdefinierte Funktionen und gespeicherte Prozeduren alle Objekte unter dem Schema `SYSTOOLS` ignoriert.

Oft ist es vorteilhaft, über ein Testsystem zu verfügen, das eine Untermenge der Daten eines Produktionssystems enthält. Jedoch sind die Zugriffspläne, die auf einem solchen Testsystem ausgewählt werden, nicht unbedingt dieselben wie die, die für das Produktionssystem ausgewählt würden. Mit dem Tool **db2look** können Sie jedoch eine Testdatenbank erstellen, deren Zugriffspläne den auf dem Produktions-

system verwendeten Plänen ähneln. Mit diesem Tool können Sie die UPDATE-Anweisungen generieren, die erforderlich sind, um die Katalogstatistiken für die Objekte einer Produktionsdatenbank in einer Testdatenbank zu replizieren. Außerdem können Sie mit diesem Tool die Befehle **UPDATE DATABASE CONFIGURATION**, **UPDATE DATABASE MANAGER CONFIGURATION** und **db2set** generieren, damit die Werte der auf das Abfrageoptimierungsprogramm bezogenen Konfigurationsparameter und der Registrierdatenbankvariablen der Testdatenbank denen einer Produktionsdatenbank entsprechen.

Die vom Befehl **db2look** generierten DDL-Anweisungen sollten überprüft werden, da sie möglicherweise nicht alle Merkmale der ursprünglichen SQL-Objekte reproduzieren. Bei Tabellenbereichen in Umgebungen mit partitionierter Datenbank sind die DDL-Anweisungen möglicherweise nicht vollständig, wenn manche Datenbankpartitionen nicht aktiv sind. Stellen Sie mit dem Befehl **ACTIVATE DATABASE** sicher, dass alle Datenbankpartitionen aktiv sind.

Berechtigung

Zugriffsrecht SELECT für die Systemkatalogtabellen.

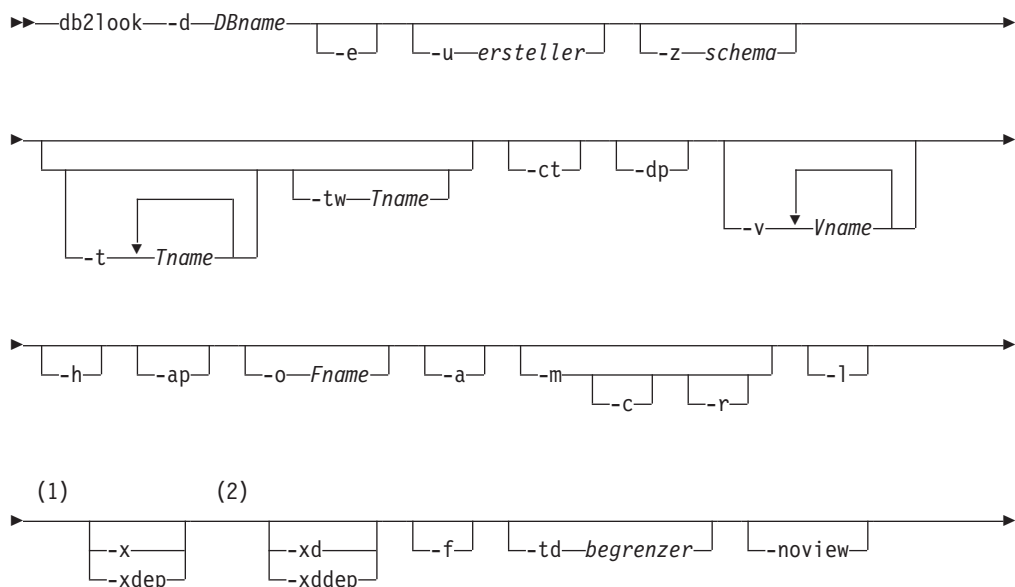
In manchen Fällen, z. B. bei der DDL-Generierung für Tabellenbereichscontainer, benötigen Sie eine der folgenden Berechtigungen:

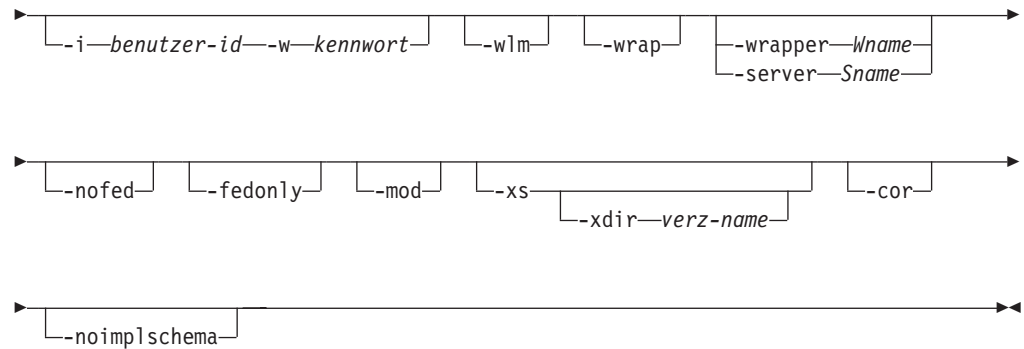
- SYSADM
- SYSCTRL
- SYSMANT
- SYSMON
- DBADM

Erforderliche Verbindung

Keine

Befehlssyntax





Anmerkungen:

- 1 Sie können den Parameter `-x` und den Parameter `-xdep` nicht gleichzeitig angeben.
- 2 Sie können den Parameter `-xd` und den Parameter `-xdep` nicht gleichzeitig angeben.

Befehlsparameter

`-d DBname`

Aliasname der Produktionsdatenbank, die abgefragt werden soll. *DBname* kann der Name einer DB2 Database for Linux, UNIX and Windows- oder DB2 Version 9.1 für z/OS-Datenbank sein. Wenn *DBname* eine DB2 for z/OS-Datenbank ist, werden mit dem Befehl `db2look` die folgenden Anweisungen für OS/390- und z/OS-Objekte generiert:

- DDL-Anweisungen für Tabellen, Indizes, Sichten und benutzerdefinierte einzigartige Typen
- UPDATE-Statistikanweisungen für Tabellen, Spalten, Spaltenverteilungen und Indizes

Diese DDL- und UPDATE-Statistikanweisungen können auf eine DB2 Database for Linux, UNIX and Windows-Datenbank angewendet werden und nicht auf eine DB2 for z/OS-Datenbank. Diese Anweisungen sind hilfreich, wenn Sie OS/390- und z/OS-Objekte extrahieren und in einer DB2 Database for Linux, UNIX and Windows-Datenbank erneut erstellen möchten.

`-e` Extrahiert DDL-Anweisungen für folgende Datenbankobjekte:

- Aliasnamen
- Prüfrichtlinien
- Prüfungen auf Integritätsbedingungen
- Funktionszuordnungen
- Funktionsschablonen
- Globale Variablen
- Indizes
- Indexspezifikationen
- MQTs (Materialized Query Tables)
- Kurznamen
- Integritätsbedingungen über Primärschlüssel
- Referenzielle Integritätsbedingungen
- Rollen

- Schemata
- Sicherheitskennsätze
- Sicherheitskennsatzkomponenten
- Sicherheitsrichtlinien
- Sequenzen
- Server
- Gespeicherte Prozeduren
- Tabellen

Anmerkung: Werte aus der Spalte STATISTICS_PROFILE in der Katalogtabelle SYSIBM.SYSTABLES sind nicht enthalten.

- Auslöser
- Gesicherte Kontexte
- Typenzuordnungen
- Benutzerzuordnungen
- Benutzerdefinierte einzigartige Typen
- Benutzerdefinierte Funktionen
- Benutzerdefinierte Methoden
- Benutzerdefinierte strukturierte Typen
- Benutzerdefinierte Umsetzungen
- Sichten
- Wrapper

Wenn Sie mithilfe von DDL-Anweisungen, die mit dem Befehl **db2look** erstellt wurden, eine benutzerdefinierte Funktion neu erstellen möchten, muss der Quellcode, auf den die Funktion verweist (z. B. die Klausel EXTERNAL NAME) verfügbar sein, damit die Funktion verwendet werden kann.

-u *ersteller*

Generiert DDL-Anweisungen für Objekte, die mit der angegebenen Ersteller-ID erstellt wurden. Begrenzt die Ausgabe auf Objekte, die mit der angegebenen Ersteller-ID erstellt wurden. Die Ausgabe enthält keine funktionsunfähigen Objekte. Verwenden Sie zum Anzeigen der funktionsunfähigen Objekte den Parameter **-a**. Wenn Sie den Parameter **-a** angeben, wird der Parameter **-u** ignoriert.

-z *schema*

Generiert DDL-Anweisungen für Objekte, die den angegebenen Schemanamen haben. Begrenzt die Ausgabe auf Objekte, die den angegebenen Schemanamen haben. Die Ausgabe enthält keine funktionsunfähigen Objekte. Verwenden Sie zum Anzeigen der funktionsunfähigen Objekte den Parameter **-a**. Wenn Sie den Parameter **-z** nicht angeben, werden Objekte mit allen Schemanamen extrahiert. Wenn Sie den Parameter **-a** angeben, wird der Parameter **-z** ignoriert. Dieser Parameter wird auch bei föderierten DDL-Anweisungen ignoriert.

-t *Tname1 Tname2 ... TnameN*

Generiert DDL-Anweisungen für die angegebenen Tabellen und die von ihnen abhängigen Objekte. Begrenzt die Ausgabe auf die in der Tabellenliste angegebenen Tabellen und generiert die DDL-Anweisungen für alle abhängigen Objekte aller benutzerdefinierten Tabellen. Die maximale Tabellenanzahl ist 30. Geben Sie die Liste wie folgt an:

- Trennen Sie an mit einem Leerzeichen.
- Schließen Sie Namen, bei denen die Groß-/Kleinschreibung zu beachten ist, sowie Doppelbytezeichensatz-Namen mit einem umgekehrten Schrägstrich (\) und mit Anführungszeichen (" ") ein (z. B. \
MeineTabelle \").
- Schließen Sie Tabellennamen, die aus mehreren Wörtern bestehen, mit einem umgekehrten Schrägstrich und zwei Sets von Anführungszeichen ein (z. B. \
"Meine Tabelle\").), damit sie vom Befehlszeilenprozessor (CLP) nicht wie einzelne Wörter behandelt werden. Wenn Sie nur ein einziges Set von Anführungszeichen verwenden (z. B. "Meine Tabelle"), werden alle Wörter in Großbuchstaben umgewandelt und der Befehl **db2look** sucht nach einem Tabellennamen in Großbuchstaben (z. B. MEI-NE TABELLE).

Wenn Sie den Parameter **-t** mit dem Parameter **-l** angeben, werden parti-
onierete Tabellen unterstützt.

Mithilfe von zweiteiligen Tabellennamen im Format *schema.tabelle* können Sie einen Tabellennamen ohne Verwendung des Parameters **-z schema** voll-
ständig qualifizieren. Verwenden Sie zweiteilige Tabellennamen, wenn die
Tabelle abhängige Objekte besitzt, die ein anderes Schema als das Tabellen-
schema aufweisen, und Sie für die abhängigen Objekte DDL-Anweisungen
generieren müssen. Wenn Sie das Schema mit dem Parameter **-z schema** an-
geben, schließt der Parameter abhängige Objekte aus, die nicht dasselbe
übergeordnete Schema aufweisen, was die Generierung von DDL-Anwei-
sungen für die abhängigen Objekte verhindert.

-tw *Tname*

Generiert DDL-Anweisungen für die Tabellen, deren Namen mit dem Mus-
ter übereinstimmen, das Sie mit *Tname* angeben, und generiert die DDL-
Anweisungen für alle Objekte, die von diesen Tabellen abhängig sind. *Tna-
me* darf nur einen einzigen Wert enthalten. Das Unterstrichungszeichen ()
in *Tname* steht für ein beliebiges einzelnes Zeichen. Das Prozentzeichen (%)
steht für eine Zeichenfolge aus null oder mehr Zeichen. Wenn **-tw** angege-
ben ist, wird die Option **-t** ignoriert.

Mithilfe von zweiteiligen Tabellennamen im Format *schema.tabelle* können Sie einen Tabellennamen ohne Verwendung des Parameters **-z schema** voll-
ständig qualifizieren. Verwenden Sie zweiteilige Tabellennamen, wenn die
Tabelle abhängige Objekte besitzt, die ein anderes Schema als das Tabellen-
schema aufweisen, und Sie für die abhängigen Objekte DDL-Anweisungen
generieren müssen. Wenn Sie das Schema mit dem Parameter **-z schema** an-
geben, schließt der Parameter abhängige Objekte aus, die nicht dasselbe
übergeordnete Schema aufweisen, was die Generierung von DDL-Anwei-
sungen für die abhängigen Objekte verhindert.

-ct

Generiert DDL-Anweisungen nach dem Zeitpunkt der Objekterstellung.
Die DDL-Anweisungen der Objekte werden möglicherweise nicht in der
richtigen Abhängigkeitsreihenfolge angezeigt. Bei Angabe des Parameters
-ct unterstützt der Befehl **db2look** nur die folgenden zusätzlichen Parame-
ter: **-e**, **-a**, **-u**, **-z**, **-t**, **-tw**, **-v**, **-l**, **-noview** und **-wlm**. Bei Verwendung des
Parameters **-ct** mit den Parametern **-z** und **-t** generiert der Befehl **db2look**
die erforderlichen UPDATE-Anweisungen zum Replizieren der Statistiken
für Tabellen, Statistiksichten, Spalten und Indizes.

-dp

Generiert vor einer Anweisung CREATE eine Anweisung DROP. Die An-
weisung DROP funktioniert möglicherweise nicht, wenn ein Objekt von
dem gelöschten Objekt abhängig ist. Sie können beispielsweise kein Sche-

ma löschen, wenn eine vorhandene Tabelle von diesem Schema abhängig ist, und Sie können keinen benutzerdefinierten Typ oder keine benutzerdefinierte Funktion löschen, wenn ein anderes Element (Typ, Funktion, Auslöser oder Tabelle) davon abhängig ist. Bei typisierten Tabellen wird die Anweisung DROP TABLE HIERARCHY nur für die Stammtabelle generiert. Für Indizes, Primär- und Fremdschlüssel sowie für Integritätsbedingungen wird keine Anweisung DROP generiert, da sie beim Löschen der Tabelle ebenfalls gelöscht werden. Eine Tabelle, die über das Attribut RESTRICT ON DROP verfügt, kann nicht gelöscht werden.

-v *Vname1 Vname2 ... VnameN*

Generiert DDL-Anweisungen für die angegebenen Sichten, jedoch nicht für die von ihnen abhängigen Objekte. Die maximale Anzahl für Sichten ist 30. Die Regeln für Tabellennamen, bei denen die Groß-/Kleinschreibung zu beachten ist, Doppelbytezeichensatz-Tabellennamen und Tabellennamen, die aus mehreren Wörtern bestehen, gelten auch für Sichtnamen. Wenn Sie den Parameter **-t** angeben, wird der Parameter **-v** ignoriert.

Sie können einen zweiteiligen Sichtnamen im Format *schema.sicht* verwenden, um eine Sicht vollständig zu qualifizieren.

-h Hilfetext anzeigen. Wenn Sie diesen Parameter angeben, werden alle übrigen Parameter ignoriert.

-ap Generiert die Anweisungen AUDIT USING, die erforderlich sind, um Prüfrichtlinien für andere Datenbankobjekte zuzuordnen.

-o *Fname*

Schreibt die Ausgabe in die Datei *Fname*. Wenn Sie keine Erweiterung angeben, wird die Erweiterung *.sql* verwendet. Wird dieser Parameter nicht angegeben, wird die Ausgabe in die Standardausgabe geschrieben.

-a Generiert DDL-Anweisungen für Objekte, die von einem beliebigen Benutzer erstellt wurden, einschließlich funktionsunfähiger Objekte. Wenn Sie diesen Parameter beispielsweise mit dem Parameter **-e** angeben, werden DDL-Anweisungen für alle Objekte in der Datenbank extrahiert. Wenn Sie diesen Parameter mit dem Parameter **-m** angeben, werden UPDATE-Anweisungen für die Statistiken aller von Benutzern erstellten Tabellen und Indizes in der Datenbank extrahiert.

Wenn Sie weder **-u** noch **-a** als Parameter auswählen, wird die Umgebungsvariable USER verwendet. Auf UNIX-Betriebssystemen brauchen Sie diese Variable nicht explizit festzulegen. Auf Windows-Betriebssystemen gibt es jedoch keinen Standardwert für die Umgebungsvariable USER. Daher müssen Sie bei SYSTEM-Variablen eine Benutzervariable festlegen oder für die Sitzung den Befehl **set USER=benutzername** absetzen.

-m Generiert die erforderlichen UPDATE-Anweisungen zum Replizieren der Statistiken für Tabellen, Statistiksichten, Spalten und Indizes. Die Verwendung des Parameters **-m** wird als Ausführung im Nachahnungsmodus (mimic) bezeichnet.

-c Wenn Sie diese Option angeben, generiert der Befehl **db2look** keine COMMIT-, CONNECT- und CONNECT RESET-Anweisungen. Standardmäßig werden diese Anweisungen generiert. Diese Option wird ignoriert, außer Sie geben außerdem den Parameter **-m** oder **-e** an.

-r Wenn Sie diese Option zusammen mit dem Parameter **-m** angeben, generiert der Befehl **db2look** den Befehl **RUNSTATS** nicht. Standardmäßig wird der Befehl **RUNSTATS** generiert.

Wichtig: Wenn Sie das Befehlsprozessorscript, das mit dem Befehl **db2look** erstellt wird, mit dem Parameter **-m** auf einer anderen Datenbank ausführen möchten (beispielsweise um die Katalogstatistik der Testdatenbank mit der in der Produktionsdatenbank abzugleichen), müssen beide Datenbanken denselben codierten Zeichensatz und dasselbe Gebietsschema verwenden.

- l Generiert DDL-Anweisungen für folgende Datenbankobjekte:
 - Benutzerdefinierte Tabellenbereiche
 - Benutzerdefinierte Datenbankpartitionsgruppen
 - Benutzerdefinierte Pufferpools
- x Generiert DDL-Berechtigungsanweisungen, z. B. GRANT-Anweisungen.

Die folgenden Berechtigungen werden unterstützt:

- Spalten: UPDATE, REFERENCES
- Datenbanken: ACCESSCTRL, BINDADD, CONNECT, CREATETAB, CREATE_EXTERNAL_ROUTINE, CREATE_NOT_FENCED_ROUTINE, DATAACCESS, DBADM, EXPLAIN, IMPLICIT_SCHEMA, LOAD, QUIESCE_CONNECT, SECADM, SQLADM, WLMADM
- Freistellungen
- Globale Variablen
- Indizes: CONTROL
- Pakete: CONTROL, BIND, EXECUTE
- Rollen
- Schemata: CREATEIN, DROPIN, ALTERIN
- Sicherheitskennsätze
- Reihenfolgen: USAGE, ALTER
- Gespeicherte Prozeduren: EXECUTE
- Tabellen: ALTER, SELECT, INSERT, DELETE, UPDATE, INDEX, REFERENCE, CONTROL
- Sichten: SELECT, INSERT, DELETE, UPDATE, CONTROL
- Benutzerdefinierte Funktionen (UDFs): EXECUTE
- Benutzerdefinierte Methoden: EXECUTE
- Tabellenbereiche: USE
- Workloads: USAGE

Anmerkung: Dieser Parameter generiert keine DDL-Berechtigungsanweisungen für abhängige Objekte, wenn er mit dem Parameter **-t** oder **-tw** verwendet wird. Verwenden Sie zum Generieren von DDL-Berechtigungsanweisungen für übergeordnete und abhängige Objekte den Parameter **-xdep**.

- xdep Generiert DDL-Berechtigungsanweisungen, z. B. GRANT-Anweisungen, für übergeordnete und abhängige Objekte. Dieser Parameter wird ignoriert, wenn der Parameter **-t** oder **-tw** nicht angegeben wird. Die folgenden Berechtigungen werden unterstützt:
 - Spalten: UPDATE, REFERENCES
 - Indizes: CONTROL
 - Gespeicherte Prozeduren: EXECUTE
 - Tabellen: ALTER, SELECT, INSERT, DELETE, UPDATE, INDEX, REFERENCE, CONTROL

- Tabellenbereiche: USE
 - Benutzerdefinierte Funktionen (UDFs): EXECUTE
 - Benutzerdefinierte Methoden: EXECUTE
 - Sichten: SELECT, INSERT, DELETE, UPDATE, CONTROL
- xd** Generiert DDL-Berechtigungsanweisungen, einschließlich DDL-Berechtigungsanweisungen für Objekte, deren Berechtigungen zum Zeitpunkt der Objekterstellung von SYSIBM erteilt wurden. Der Parameter generiert keine Autorisierungs-DLLs für Systemkatalogtabellen und Katalogsichten.
- Anmerkung:** Er generiert keine DDL-Berechtigungsanweisungen für abhängige Objekte, wenn er mit dem Parameter **-t** oder **-tw** verwendet wird. Verwenden Sie zum Generieren von DDL-Berechtigungsanweisungen für übergeordnete und abhängige Objekte den Parameter **-xddep**.
- xddep** Generiert alle DDL-Berechtigungsanweisungen für übergeordnete und abhängige Objekte, einschließlich DDL-Berechtigungsanweisungen für Objekte, deren Berechtigungen zum Zeitpunkt der Objekterstellung von SYSIBM erteilt wurden. Dieser Parameter wird ignoriert, wenn der Parameter **-t** oder **-tw** nicht angegeben wird.
- f** Extrahiert die Konfigurationsparameter und Registrierdatenbankvariablen, die das Abfrageoptimierungsprogramm betreffen.
- td** *begrenzer*
Gibt den Anweisungsbegrenzer für SQL-Anweisungen an, die mit dem Befehl **db2look** generiert werden. Der Standardbegrenzer ist ein Semikolon (;). Verwenden Sie diesen Parameter, wenn Sie den Parameter **-e** angeben, weil die extrahierten Objekte möglicherweise Auslöser oder SQL-Routinen enthalten.
- noview**
Gibt an, dass CREATE VIEW-DDL-Anweisungen nicht extrahiert werden.
- i** *benutzer-id*
Gibt die Benutzer-ID an, mit der sich der Befehl **db2look** an einem fernen System anmeldet. Wenn Sie diesen Parameter und den Parameter **-w** angeben, kann der Befehl **db2look** für eine Datenbank auf einem fernen System ausgeführt werden. Die lokale und die ferne Datenbank müssen dieselbe DB2-Version verwenden.
- w** *kennwort*
Gibt das Kennwort an, mit dem sich der Befehl **db2look** an einem fernen System anmeldet. Wenn Sie diesen Parameter und den Parameter **-i** angeben, kann der Befehl **db2look** für eine Datenbank auf einem fernen System ausgeführt werden. Die lokale und die ferne Datenbank müssen dieselbe DB2-Version verwenden.
- wlm** Generiert eine WLM-spezifische DDL-Ausgabe, mit der CREATE- und ALTER-Anweisungen für folgende Elemente generiert werden können:
- Histogramme
 - Serviceklassen
 - Schwellenwerte
 - WLM-Ereignismonitore
 - Workloads
 - Arbeitsaktionssets

- Arbeitsklassensets
- wrap** Generiert verschlüsselte Versionen von DDL-Anweisungen für Routinen, Trigger, Sichten und PL/SQL-Pakete.
- wrapper** *Wname*
Generiert DDL-Anweisungen für föderierte Objekte, die für den angegebenen Wrapper gelten. Die folgenden föderierten DDL-Anweisungen können generiert werden:
- CREATE FUNCTION ... AS TEMPLATE
 - CREATE FUNCTION MAPPING
 - CREATE INDEX SPECIFICATION
 - CREATE NICKNAME
 - CREATE SERVER
 - CREATE TYPE MAPPING
 - CREATE USER MAPPING
 - CREATE WRAPPER
 - GRANT (Zugriffsrechte für Kurznamen, Server, Indizes)
- Wenn Sie keinen oder mehrere Wrappernamen angeben, wird ein Fehler zurückgegeben.
- server** *Sname*
Generiert DDL-Anweisungen für föderierte Objekte, die für den angegebenen Server gelten. Die folgenden föderierten DDL-Anweisungen können generiert werden:
- CREATE FUNCTION ... AS TEMPLATE
 - CREATE FUNCTION MAPPING
 - CREATE INDEX SPECIFICATION
 - CREATE NICKNAME
 - CREATE SERVER
 - CREATE TYPE MAPPING
 - CREATE USER MAPPING
 - CREATE WRAPPER
 - GRANT (Zugriffsrechte für Kurznamen, Server, Indizes)
- Wenn Sie keinen oder mehrere Servernamen angeben, wird ein Fehler zurückgegeben.
- nofed** Gibt an, dass keine föderierten DDL-Anweisungen generiert werden. Wenn Sie diesen Parameter angeben, werden die Parameter **-wrapper** und **-server** ignoriert.
- fedonly**
Gibt an, dass nur föderierte DDL-Anweisungen generiert werden.
- mod** Generiert DDL-Anweisungen für die einzelnen Module und für alle in den einzelnen Modulen definierten Objekte.
- xs** Exportiert alle Dateien, die zum Registrieren von XML-Schemata und DTDs in der Zieldatenbank erforderlich sind, und generiert entsprechende Befehle für ihre Registrierung. Die Gruppe der XSR-Objekte, die exportiert werden, wird durch die Parameter **-u**, **-z** und **-a** gesteuert.

- xdir** *verz-name*
Exportiert XML-bezogene Dateien in den angegebenen Pfad. Wird dieser Parameter nicht angegeben, werden alle XML-bezogenen Dateien in das aktuelle Verzeichnis exportiert.
- cor** Generiert DDL-Anweisungen mit der Klausel CREATE OR REPLACE unabhängig davon, ob die Anweisungen diese Klausel ursprünglich enthalten haben.
- noimplschema**
Gibt an, dass Anweisungen des Typs CREATE SCHEMA DDL für implizit erstellte Schemas nicht generiert werden. Wenn Sie diesen Parameter festlegen, muss auch der Parameter **-e** angegeben werden.

Beispiele

Die folgenden Beispiele zeigen, wie der Befehl **db2look** zu verwenden ist:

- Generieren der DDL-Anweisungen für Objekte, die vom Benutzer `walid` in der Datenbank DEPARTMENT erstellt wurden. Die Ausgabe wird an die Datei `db2look.sql` gesendet.
`db2look -d department -u walid -e -o db2look.sql`
- Generieren der DDL-Anweisungen für Objekte, die den Schemanamen `ianhe` aufweisen, der vom Benutzer `walid` in der Datenbank DEPARTMENT erstellt wurden. Die Ausgabe wird an die Datei `db2look.sql` gesendet.
`db2look -d department -u walid -z ianhe -e -o db2look.sql`
- Generieren der UPDATE-Anweisungen zum Replizieren der Statistiken für die Datenbankobjekte, die vom Benutzer `walid` in der Datenbank DEPARTMENT erstellt wurden. Die Ausgabe wird an die Datei `db2look.sql` gesendet.
`db2look -d department -u walid -m -o db2look.sql`
- Generieren der DDL-Anweisungen für die vom Benutzer `walid` erstellten Objekte und der UPDATE-Anweisungen zum Replizieren der Statistikdaten zu den vom gleichen Benutzer erstellten Datenbankobjekten. Die Ausgabe wird an die Datei `db2look.sql` gesendet.
`+db2look -d department -u walid -e -m -o db2look.sql`
- Generieren der DDL-Anweisungen für die von allen Benutzern in der Datenbank DEPARTMENT erstellten Objekte. Die Ausgabe wird an die Datei `db2look.sql` gesendet.
`db2look -d department -a -e -o db2look.sql`
- Generieren der DDL-Anweisungen für alle benutzerdefinierten Datenbankpartitionsgruppen, Pufferpools und Tabellenbereiche. Die Ausgabe wird an die Datei `db2look.sql` gesendet.
`db2look -d department -l -o db2look.sql`
- Generieren der UPDATE-Anweisungen für die auf das Optimierungsprogramm bezogenen Konfigurationsparameter der Datenbank und des Datenbankmanagers sowie der **db2set**-Befehle für auf das Optimierungsprogramm bezogene Registrierdatenbankvariablen in der Datenbank DEPARTMENT. Die Ausgabe wird an die Datei `db2look.sql` gesendet.
`db2look -d department -f -o db2look.sql`
- Generieren der **db2set**-Befehle für auf das Optimierungsprogramm bezogene Registrierdatenbankvariablen und der folgenden Anweisungen für die Datenbank DEPARTMENT:
 - DDL-Anweisungen für alle Datenbankobjekte

- UPDATE-Anweisungen zum Replizieren der Statistiken für alle Tabellen und Indizes
- GRANT-Berechtigungsanweisungen
- UPDATE-Anweisungen für die auf das Optimierungsprogramm bezogenen Konfigurationsparameter der Datenbank und des Datenbankmanagers
- **db2set**-Befehle für auf das Optimierungsprogramm bezogene Registrierdatenbankvariablen
- DDL-Anweisungen für alle benutzerdefinierten Datenbankpartitionsgruppen, Pufferpools und Tabellenbereiche

Die Ausgabe wird an die Datei db2look.sql gesendet.

```
db2look -d department -a -e -m -l -x -f -o db2look.sql
```

- Generieren aller DDL-Anweisungen zur Autorisierung für alle Objekte in der Datenbank DEPARTMENT, einschließlich der vom ursprünglichen Ersteller erstellten Objekte. (In diesem Fall wurden die Autorisierungen zum Zeitpunkt der Objekterstellung von SYSIBM erteilt.) Die Ausgabe wird an die Datei db2look.sql gesendet.

```
db2look -d department -xd -o db2look.sql
```

- Generieren der DDL-Anweisungen für die von allen Benutzern in der Datenbank DEPARTMENT erstellten Objekte. Die Ausgabe wird an die Datei db2look.sql gesendet.

```
db2look -d department -a -e -td % -o db2look.sql
```

Die Ausgabe kann anschließend vom Befehlszeilenprozessor gelesen werden:

```
db2 -td% -f db2look.sql
```

- Generieren der DDL-Anweisungen für Objekte in der Datenbank DEPARTMENT, ausgenommen die CREATE VIEW-Anweisungen. Die Ausgabe wird an die Datei db2look.sql gesendet.

```
db2look -d department -e -noview -o db2look.sql
```

- Generieren der DDL-Anweisungen für Objekte in der Datenbank DEPARTMENT, die sich auf die angegebenen Tabellen beziehen. Die Ausgabe wird an die Datei db2look.sql gesendet.

```
db2look -d department -e -t tabl "\"Meine TaBe11E2\""" -o db2look.sql
```

- Generieren der DDL-Anweisungen für alle Objekte (föderierte und nicht föderierte) in der föderierten Datenbank FEDDEPART. Föderierte DDL-Anweisungen werden nur generiert, soweit sie für den angegebenen Wrapper (FEDWRAP) gelten. Die Ausgabe wird an die Standardausgabe gesendet.

```
db2look -d feddepart -e -wrapper fedwrap
```

- Generieren einer Scriptdatei, die nur nicht föderierte DDL-Anweisungen enthält. Der folgende Systembefehl kann für eine föderierte Datenbank (FEDDEPART) ausgeführt werden und dennoch nur Ausgabedaten wie bei der Ausführung für eine nicht föderierte Datenbank erzeugen. Die Ausgabe wird an die Datei out.sql gesendet.

```
db2look -d feddepart -e -nofed -o out
```

- Generieren der DDL-Anweisungen für Objekte in der Datenbank DEPARTMENT, die den Schemanamen valid aufweisen. Die erforderlichen Dateien zum Registrieren der enthaltenen XML-Schemata und DTDs werden in das aktuelle Verzeichnis exportiert. Die Ausgabe wird an die Datei db2look.sql gesendet.

```
db2look -d department -z valid -e -xs -o db2look.sql
```

- Generieren der DDL-Anweisungen für die von allen Benutzern in der Datenbank DEPARTMENT erstellten Objekte. Die zum Registrieren der enthaltenen XML-

Schemata und DTDs erforderlichen Dateien werden in das Verzeichnis /home/ofer/ofer/ exportiert. Die Ausgabe wird an die Standardausgabe gesendet.

```
db2look -d department -a -e -xs -xdir /home/ofer/ofer/
```

- Generieren von ausschließlich WLM-spezifischen DDL-Anweisungen für die Datenbank DEPARTMENT:

```
db2look -d department -wlm
```

- Generieren der DDL-Anweisungen für alle Objekte in der Datenbank DEPARTMENT:

```
db2look -d department -wlm -e -l
```

- Generieren der DDL-Anweisungen für die übergeordnete Tabelle TAB1 im Schema TABLES und der abhängigen Sicht von TAB1 mit der Bezeichnung VIEW1 im Schema VIEWS der Datenbank DB1. Die Ausgabe wird an die Datei db2look.sql gesendet.

```
db2look -d DB1 -t TABLES.TAB1 -e -o db2look.sql
```

- Generieren der DDL-Anweisungen und der DDL-Berechtigungsanweisungen für die übergeordnete Tabelle TAB1 im Schema TABLES und der abhängigen Sicht von TAB1 mit der Bezeichnung VIEW1 im Schema VIEWS der Datenbank DB1. Die Ausgabe wird an die Datei db2look.sql gesendet.

```
db2look -d DB1 -t TABLES.TAB1 -e -xdep -o db2look.sql
```

- Generieren der RUNSTATS DDL-Anweisungen für die Tabelle TABLE1 im Abbilmodus. Die Ausgabe wird an die Datei db2look.sql gesendet. Nicht alle verfügbaren RUNSTATS-Klauseln des Befehls werden unterstützt.

```
db2look -d DB1 -t TABLE1 -m -e -o db2look.sql
```

Hinweise zur Verwendung

Auf Windows-Betriebssystemen müssen Sie den Befehl **db2look** über ein DB2-Befehlsfenster absetzen.

Standardmäßig verfügt der Instanzeigner über die Berechtigung EXECUTE für **db2look**-Pakete. Damit andere Benutzer den Befehl **db2look** ausführen können, muss der Instanzeigner die Berechtigung EXECUTE für **db2look**-Pakete erteilen. Um die **db2look**-Paketnamen zu ermitteln, kann der Befehl **db2bfd** folgendermaßen verwendet werden:

```
cd ../sqllib/bnd
db2bfd -b db2look.bnd
db2bfd -b db21kfun.bnd
db2bfd -b db21ksp.bnd
```

Zum Erstellen von DDL-Anweisungen für föderierte Objekte müssen Sie in der Datenbankmanagerkonfiguration die Verwendung föderierter Systeme aktivieren. Nachdem die Scriptdatei von dem Befehl **db2look** generiert wurde, müssen Sie den Konfigurationsparameter **federated** auf YES setzen, bevor das Script ausgeführt wird. In einer föderierten Umgebung werden die folgenden Parameter des Befehls **db2look** unterstützt:

-ap

Generiert AUDIT USING-Anweisungen.

-e

Generiert DDL-Anweisungen für föderierte Objekte.

-f

Extrahiert föderationsbezogene Informationen aus der Datenbankmanagerkonfiguration.

- m Extrahiert Statistiken für Kurznamen.
- x Generiert GRANT-Anweisungen zur Erteilung von Zugriffsrechten für föderierte Objekte.
- xd
Generiert DDL-Anweisungen, um Zugriffsrechte, die vom System erteilt wurden, zu föderierten Objekten hinzuzufügen.
- wlm
Generiert WLM-spezifische DDL-Anweisungen.

Wenn die Kurznamenspalte und die Spalte der fernen Tabelle unterschiedliche Datentypen aufweisen, wird mit dem Befehl **db2look** eine Anweisung ALTER COLUMN für die Kurznamenspalte generiert.

Sie müssen das Ausgabescript modifizieren, um die fernen Kennwörter für die CREATE USER MAPPING-Anweisungen hinzuzufügen.

Sie müssen das Ausgabescript des Befehls **db2look** modifizieren, indem Sie AUTHORIZATION und PASSWORD zu den verwendeten CREATE SERVER-Anweisungen hinzufügen, um eine Instanz der DB2-Produktfamilie als Datenquelle zu definieren.

Die Option **-tw** wird wie folgt verwendet:

- Zum Generieren der DDL-Anweisungen für Objekte in der Datenbank DEPARTMENT, die Tabellen zugeordnet sind, deren Namen mit abc beginnen, und zum Senden der Ausgabe an die Datei db2look.sql:


```
db2look -d department -e -tw abc% -o db2look.sql
```
- Zum Generieren der DDL-Anweisungen für Objekte in der Datenbank DEPARTMENT, die Tabellen zugeordnet sind, deren Namen ein d als zweites Zeichen enthalten, und zum Senden der Ausgabe an die Datei db2look.sql:


```
db2look -d department -e -tw _d% -o db2look.sql
```
- Der Befehl **db2look** verwendet das Prädikat LIKE, um zu bewerten, welche Tabellennamen mit dem im Argument *Tname* angegebenen Muster übereinstimmen. Wenn bei Verwendung des Prädikats LIKE das Zeichen _ oder % im Tabellennamen enthalten ist, muss unmittelbar vor diesem Zeichen der umgekehrte Schrägstrich angegeben werden. In diesem Fall kann weder das Zeichen _ noch das Zeichen % als Platzhalterzeichen in *Tname* verwendet werden. Geben Sie beispielsweise Folgendes ein, um die DDL-Anweisungen für Objekte in der Datenbank DEPARTMENT zu generieren, denen Tabellen zugeordnet sind, deren Namen weder als erstes noch als letztes Zeichen ein Prozentzeichen enthalten:


```
db2look -d department -e -tw zeichenfolge\%string
```
- Tabellen- und Sichtnamen, bei denen Groß-/Kleinschreibung zu beachten ist, Doppelbytezeichensatz-Tabellen- und -Sichtnamen sowie Tabellen- und Sichtnamen, die aus mehreren Wörtern bestehen, müssen in einen umgekehrten Schrägstrich und Anführungszeichen eingeschlossen werden. Beispiel:


```
\\"Meine Tabelle\"
```

Wenn ein Mehrbytezeichensatz- oder Doppelbytezeichensatz-Name nicht in das Begrenzungszeichen aus umgekehrtem Schrägstrich und Anführungszeichen eingeschlossen ist und dasselbe Byte enthält wie der Kleinbuchstabe, wird er in Großbuchstaben umgewandelt und der Befehl **db2look** sucht nach einem Datenbankobjekt mit dem umgewandelten Namen. Dies führt dazu, dass die DDL-Anweisung nicht extrahiert wird.

- Die Option **-tw** kann zusammen mit der Option **-x** (zum Generieren von GRANT-Zugriffsrechten) verwendet werden sowie mit der Option **-m** (zum Zurückgeben von Tabellen- und Spaltenstatistiken) und mit der Option **-l** (zum Generieren der DDL-Anweisungen für benutzerdefinierte Tabellenbereiche, Partitionsgruppen und Pufferpools). Wenn die Option **-t** zusammen mit der Option **-tw** angegeben wird, wird die Option **-t** (und das dazugehörige Argument *Tname*) ignoriert.
- Die Option **-tw** kann nicht zum Generieren der DDL-Anweisungen für Tabellen (und für deren zugehörige Objekte) verwendet werden, die sich in föderierten Datenquellen oder in DB2 Universal Database for z/OS and OS/390, in DB2 for i oder in DB2 Server for VSE & VM befinden.
- Die Option **-tw** wird nur über den Befehlszeilenprozessor unterstützt.

Wenn Sie versuchen, DDL-Anweisungen auf Systemen in einer Umgebung mit partitionierten Datenbanken zu generieren, wird anstelle der DDL-Anweisungen für Tabellenbereiche, die sich in inaktiven Datenbankpartitionen befinden, eine Warnung angezeigt. Um sicherzustellen, dass für alle Tabellenbereiche korrekte DDL-Anweisungen erstellt werden, müssen Sie alle Datenbankpartitionen aktivieren.

Der Befehl **db2look** wird nur in Datenbanken mit DB2 Version 9.5 oder höher unterstützt, Sie können ihn jedoch auf DB2-Clients früherer Versionen absetzen. Beispiel: Das Absetzen dieses Befehls von einem Client mit Version 9.1 für eine Datenbank mit Version 9.5 wird unterstützt. Das Absetzen dieses Befehls von einem Client mit Version 9.5 für eine Datenbank mit Version 9.1 wird dagegen nicht unterstützt.

Wenn Sie das Dienstprogramm db2look aufrufen, generiert der Befehl **db2look** die DDL-Anweisungen für alle Objekte, die mithilfe einer nicht festgeschriebene Transaktion erstellt wurden.

Beim Extrahieren einer DDL-Anweisung für eine Sicherheitskennsatzkomponente des Typs 'array' generiert die extrahierte Anweisung möglicherweise keine Komponente, deren interne Darstellung (Codierung von Elementen in dem Array) mit der Darstellung dieser Komponente in der Datenbank übereinstimmt, für die Sie den Befehl **db2look** abgesetzt haben. Diese Diskrepanz kann auftreten, wenn Sie die Sicherheitskennsatzkomponente durch Hinzufügen von einem oder mehreren Elementen geändert haben. In diesem Fall sind für Daten, die aus einer Tabelle extrahiert und in eine andere, aus einer **db2look**-Ausgabe erstellte Tabelle versetzt werden, keine entsprechenden Sicherheitskennsatzwerte vorhanden, sodass die neue Tabelle möglicherweise nicht umfassend geschützt ist.

Zugehörige Informationen:

Namen für Kurznamenspalten und Indizes

Migrationsänderungen mit Auswirkungen auf Anwendungen

Dateiformate und Datentypen

Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD

Es werden fünf Betriebssystemdateiformate beschrieben, die von den DB2-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 (Integration 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 (WSF). Dieses Format ermöglicht den Datenaustausch mit Produkten wie Lotus 1-2-3 und Symphony. Dieses Dateiformat wird vom Dienstprogramm LOAD nicht unterstützt.

Dieses Format ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt.

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. Der einzigartige Datentyp (UDT, User-Defined Distinct Typ) muss jedoch definiert sein. Andernfalls erhalten Sie eine Fehlermeldung zu einem Fehler durch einen nicht definierten Namen (SQL0204N). Ebenso werden UDTs beim Exportieren in das PC/IXF-Datendateiformat in der Ausgabedatei 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.

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-Familie herzustellen, erstellt das Dienstprogramm EXPORT Dateien mit numerischen Daten im Intel-Format, während das Dienstprogramm IMPORT die Daten in diesem Format erwartet.

Abhängig von der Hardwareplattform konvertieren DB2-Produkte numerische Werte zwischen Intel- und Nicht-Intel-Formaten (mittels Bytefolgeumkehrung) sowohl während der Export- als auch der Importoperationen.

UNIX-basierte Implementierungen von DB2-Datenbanken 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 unter UNIX mit dem DB2-Dienstprogramm EXPORT generiert wurden, können von DB2-Datenbanken für Windows erstellt 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
 - Textdateien von UNIX-Betriebssystemen verwenden ein Zeilenvorschubzeichen.
 - Textdateien von anderen Betriebssystemen verwenden eine Folge aus Rücklauf- und Zeilenvorschubzeichen.
- Dateiendezeichen
 - Textdateien von UNIX-Betriebssystemen weisen kein Dateiendezeichen auf.
 - Textdateien von anderen Betriebssystemen 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 (Arbeitsblattformat)

Numerische Daten in Dateien im WSF-Format (Arbeitsblattformat) werden im Intel-Maschinenformat gespeichert. Dieses Format ermöglicht es, Lotus-WSF-Dateien in andere Lotus-Betriebsumgebungen (z. B. auf Intel und UNIX-basierten Systemen) zu übertragen und dort zu verwenden.

Anmerkung: Die Unterstützung für dieses Dateiformat ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.

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.

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^a

Spaltenbegrenzer ::= Standardwert: ASCII-Komma (,)^b

```
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 (")^c

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 angegeben ist

Dateieindezeichen ::= Hex '1A' (nur Windows-Betriebssysteme)

ASCII-Darstellung eines numerischen Werts^d ::= 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 (.)^e

- ^a Als Satzbegrenzer wird das Zeilenvorschubzeichen, ASCII-Wert x0A, angenommen. Unter einem Windows-Betriebssystem 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 geladen werden, indem der Änderungswert codepage für den Dateityp mit dem Befehl **LOAD** verwendet wird).
- ^b Der Spaltenbegrenzer kann mit dem Änderungswert coldel für den Dateityp angegeben werden.
- ^c Der Zeichenfolgebegrenzer kann mit dem Änderungswert chardel für den Dateityp angegeben werden.

Anmerkung: Die Standardpriorität der Begrenzer ist:

1. Satzbegrenzer
2. Zeichenbegrenzer
3. Spaltenbegrenzer

- ^d 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.
- ^e Das Dezimalzeichen kann mit dem Änderungswert `decpt` für den Datentyp angegeben werden.

Vom Dienstprogramm EXPORT wird jedes Zeichenfolgebegrenzerbyte (der Standardwert ist ein doppeltes Anführungszeichen oder `x22`), das in Spaltendaten eingebettet ist, durch zwei Zeichenfolgebegrenzerbyte ersetzt (das Zeichen wird also verdoppelt). Dies ist erforderlich, damit die Parsingroutinen beim Import zwischen einem Zeichenfolgebegrenzerbyte, das Anfang und Ende einer Spalte definiert, und einem Zeichenfolgebegrenzerbyte unterscheiden können, das in den Spaltendaten enthalten ist. Dies ist bei der Verwendung einer exportierten DEL-Datei für einige andere Anwendungen als das Dienstprogramm EXPORT zu beachten. Dieselbe Zeichenverdoppelung erfolgt bei Zeichenfolgebegrenzern innerhalb von mit 'FOR BIT' markierten binären Spaltendaten.

DEL-Datentypbeschreibungen:

Tabelle 48. Gültige Datentypformate für das DEL-Dateiformat

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
BIGINT	Ganzzahlige Konstante (INTEGER) im Bereich von -9 223 372 036 854 775 808 bis 9 223 372 036 854 775 807.	ASCII-Darstellung eines numerischen Werts im Bereich von -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 aufgefüllt (X'20') und so an die Breite der Datenbankspalte angeglichen.

Tabelle 48. Gültige Datentypformate für das DEL-Dateiformat (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
DATE	<p><i>jjjmmmtt</i> (Jahr Monat Tag) ohne Spaltenbegrenzer. Beispiel: 19931029</p> <p>Alternativ kann die Option DATESISO verwendet werden, um festzulegen, dass alle Datumswerte im ISO-Format exportiert werden sollen.</p>	<p>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>.</p>
DBCLOB (nur DBCS)	<p>Grafikdaten werden als begrenzte Zeichenfolge exportiert.</p>	<p>Eine begrenzte oder nicht begrenzte Zeichenfolge (Länge ist eine gerade Anzahl Byte). Die Zeichenfolge wird als Wert der Datenbankspalte verwendet.</p>
DBCLOB_FILE (nur DBCS)	<p>Die Zeichendaten für jede DBCLOB-Spalte werden in einzelnen Dateien gespeichert, und der Dateiname ist in Zeichenbegrenzer eingeschlossen.</p>	<p>Der begrenzte oder nicht begrenzte Name der Datei, die die Daten enthält.</p>
DB2SECURITYLABEL	<p>Spaltendaten werden als Rohdaten in Anführungszeichen (") exportiert. Verwenden Sie die Skalarfunktion SECLABEL_TO_CHAR in der Anweisung SELECT, um den Wert in das Zeichenfolgeformat des Sicherheitskennsatzes zu konvertieren.</p>	<p>Es wird standardmäßig davon ausgegangen, dass es sich beim Wert in der Datendatei um die tatsächlichen Byte handelt, die die interne Darstellung dieses Sicherheitskennsatzes bilden. Es wird davon ausgegangen, dass der Wert durch Anführungszeichen (") begrenzt ist.</p>
DECIMAL	<p>Eine DECIMAL-Konstante mit der Genauigkeit und der Anzahl der Kommastellen des Feldes, das exportiert wird. Mit dem Änderungswert <i>decplusblank</i> für den Dateityp kann angegeben werden, dass positiven Dezimalwerten ein Leerzeichen statt eines Pluszeichens (+) vorangestellt werden soll.</p>	<p>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.</p>
FLOAT(long)	<p>Eine FLOAT-Konstante im Bereich -10E307 bis 10E307.</p>	<p>ASCII-Darstellung eines numerischen Wertes im Bereich -10E307 bis 10E307.</p>

Tabelle 48. Gültige Datentypformate für das DEL-Dateiformat (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
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	Ganzzahlige Konstante (INTEGER) im Bereich von -2 147 483 648 bis 2 147 483 647.	ASCII-Darstellung eines numerischen Werts im Bereich von -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	Ganzzahlige Konstante (INTEGER) im Bereich von -32 768 bis 32 767.	ASCII-Darstellung eines numerischen Werts im Bereich von -32 768 bis 32 767. Dezimal- und Gleitkommazahlen werden ganzzahlig abgeschnitten.
TIME	ss.mm.ss (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	jjjj-mm-tt-ss.mm.ss.nnnnnn (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.

Tabelle 48. Gültige Datentypformate für das DEL-Dateiformat (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
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.

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
```

Anmerkung:

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, sind die Begrenzer nicht unbedingt mit den DBCS-Startzeichen und -Endezeichen identisch.
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.

Aspekte beim Versetzen von XML-Daten- Begrenzer:

Beim Versetzen von ASCII-Dateien (DEL) muss sichergestellt werden, dass die Daten beim Versetzen nicht versehentlich durch Fehler bei der Erkennung von Begrenzerzeichen geändert werden. Um derartige Fehler zu vermeiden, gibt DB2 verschiedene einschränkende Bedingungen vor und stellt eine Anzahl von Änderungswerten für den Dateityp bereit.

Einschränkungen für Begrenzer

Verschiedene Einschränkungen verhindern, dass das gewählte Begrenzerzeichen als Teil der zu versetzenden Daten behandelt wird. Zunächst gilt, dass Begrenzer sich gegenseitig ausschließen müssen. Zweitens dürfen Begrenzer keine binäre Nullen, Zeilenvorschubzeichen, Rücklaufzeichen oder Leerzeichen sein. Auch das Standarddezimalzeichen (.) darf nicht als Zeichenfolgebegrenzer verwendet werden. Und abschließend wäre zu nennen, dass das Pipe-Zeichen (|) in einer DBCS-Umgebung nicht als Zeichenbegrenzer unterstützt wird.

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:
 - Zeichenfolgebegrenzer: " (0x22, doppeltes Anführungszeichen)
 - Spaltenbegrenzer: , (0x2c, Komma)
- Standardbegrenzer für Datendateien in EBCDIC-SBCS-Codepages sind:
 - Zeichenfolgebegrenzer: " (0x7F, doppeltes Anführungszeichen)
 - Spaltenbegrenzer: , (0x6B, Komma)
- 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 ...
```

Probleme mit Begrenzern beim Versetzen von Daten

Doppelte Zeichenbegrenzer

Standardmäßig werden bei zeichenbasierten Feldern in DEL-Dateien alle Vorkommen von in einem Feld gefundenen Zeichenbegrenzern durch doppelte Zeichenbegrenzer dargestellt. Wird z. B. das doppelte Anführungszeichen als Zeichenbegrenzer verwendet, wird der exportierte Text `I am 6" tall.` im Ausgabertext in der DEL-Datei folgendermaßen wiedergegeben: `"I am 6"" tall."`. Auf der anderen Seite gilt, dass der Eingabetext in einer DEL-Datei `"What a ""nice"" day!"` als `What a "nice" day!` importiert wird.

nodoubledel

Das Verhalten bei doppelten Zeichenbegrenzern kann für die Dienstprogramme `IMPORT`, `EXPORT` und `LOAD` durch Angabe des Änderungswerts für den Dateityp `nodoubledel` inaktiviert werden. Dabei ist jedoch zu bedenken, dass das Verhalten bei doppelten Zeichenbegrenzern eingesetzt wird, um Parsing-Fehler zu vermeiden. Wenn Sie den Änderungswert `nodoubledel` beim Export verwenden, werden vorhandene Zeichenbegrenzer in Zeichenfeldern nicht verdoppelt. Wenn Sie `nodoubledel` beim Import und beim Laden verwenden, werden doppelte Zeichenbegrenzer nicht als Literalvorkommen von Zeichenbegrenzern interpretiert.

nochardel

Wenn Sie den Änderungswert `nochardel` für den Dateityp beim Export verwenden, werden die Zeichenfelder nicht in Zeichenbegrenzer eingeschlossen. Wird `nochardel` beim Importieren oder Laden verwendet, werden die Zeichenbegrenzer nicht als Sonderzeichen, sondern als echte Daten interpretiert.

chardel

Es gibt weitere Änderungswerte für den Dateityp, die manuell angewendet werden können, um eine Verwechslung von Standardbegrenzern und Daten zu verhindern. Der Änderungswert `chardel` für den Dateityp gibt an, dass anstelle des Standardbegrenzers in Form von doppelten Anführungszeichen das einzelne Zeichen `x` als Zeichenfolgebegrenzer verwendet werden soll.

coldel

Gleichermaßen können Sie mit dem Änderungswert `coldel`, der angibt, dass das einzelne Zeichen `x` als Begrenzer für Spaltendaten verwendet werden soll, die standardmäßige Verwendung des Kommas als Spaltenbegrenzer verhindern.

delprioritychar

Ein weiterer beim Versetzen von DEL-Dateien zu beachtender Aspekt ist, dass der Vorrang der einzelnen Begrenzer eingehalten werden muss. Die Standardpriorität für Begrenzer lautet: Zeile, Zeichen, Spalte. Bei einigen Anwendungen ist jedoch folgende Priorität wichtig: Zeichen, Zeile, Spalte. Mit dieser Standardpriorität würde z. B. die DEL-Datendatei

```
"Vincent <zeilenbegrenzer> is a manager",<zeilenbegrenzer>
```

als eine Datei mit zwei Zeilen interpretiert (Vincent, und is a manager), da der `<zeilenbegrenzer>` Vorrang vor dem Zeichenbegrenzer (`"`) hat. Bei Verwendung von `delprioritychar` hat der Zeichenbegrenzer (`"`) Vorrang vor dem Zeilenbegrenzer (`<zeilenbegrenzer>`), sodass dieselbe DEL-Datei in diesem Fall richtig als Datei mit einer einzigen Zeile interpretiert wird: `Vincent is a manager.`

ASC-Dateiformat (ASCII-Format mit universellen Zeilenbegrenzern)

Das ASCII-Format mit universellen Zeilenbegrenzern, das die Dienstprogramme `IMPORT` und `LOAD` als `ASC` kennen, wird in zwei Varianten zur Verfügung gestellt: mit fester Länge und mit flexibler Länge. Beim `ASC`-Format mit fester Länge

haben alle Datensätze eine feste Länge. Beim ASC-Format mit flexibler Länge sind die Datensätze durch einen Zeilenbegrenzer (immer ein Zeilenumbruch) begrenzt. Der Begriff *universell* bedeutet im Zusammenhang mit dem ASCII-Format mit universellen Zeilenbegrenzern, dass Spaltenwerte nicht durch Begrenzer voneinander getrennt werden.

Beim Importieren oder Laden von ASC-Daten wird durch den Änderungswert `reclen` für den Dateityp angegeben, dass die Datendatei im ASC-Format mit fester Länge vorliegt. Wird dieser Änderungswert nicht angegeben, bedeutet dies, dass die Datendatei im ASC-Format mit flexibler Länge vorliegt.

Das ASCII-Format mit universellen Zeilenbegrenzern 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 (angegeben über **IMPORT**-Parameter) definiert. Jedes Paar stellt Speicherpositionen innerhalb einer Zeile dar, die als Bytepositionen angegeben werden. Die erste Position innerhalb einer Zeile ist die Byteposition 1. Das erste Element jedes Positionspaares 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-Zeilenvorschubfolgea
```

- ^a Als Satzbegrenzer wird das Zeilenvorschubzeichen, ASCII-Wert `x0A`, angenommen. Unter einem Windows-Betriebssystem 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 geladen werden, indem der Änderungswert `codepage` für den Dateityp mit dem Befehl **LOAD** verwendet wird). Der Satzbegrenzer wird nie als Teil eines Feldes mit Daten gewertet.

ASC-Datentypbeschreibungen:

Tabelle 49. 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 von -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	<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>
BLOB_FILE, CLOB_FILE, DBCLOB_FILE (nur DBCS)	<p>Der begrenzte oder nicht begrenzte Name der Datei, die die Daten enthält.</p>
CHAR	<p>Eine Zeichenfolge. Die Zeichenfolge wird nach Bedarf rechts abgeschnitten oder mit Leerzeichen aufgefüllt und so an die Breite der Zielspalte angeglichen.</p>
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)	<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 und so an die maximal zulässige Länge der Zielspalte angeglichen.</p>
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>

Tabelle 49. Gültige Datentypformate für das ASC-Dateiformat (Forts.)

Datentyp	Für das Dienstprogramm IMPORT zulässiges Format
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 von -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>
LONG VARGRAPHIC (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 und so an die maximal zulässige Länge der Zielspalte angeglichen.</p>
SMALLINT	<p>Eine Konstante eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich von -32 768 bis 32 767 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>

Tabelle 49. Gültige Datentypformate für das ASC-Dateiformat (Forts.)

Datentyp	Für das Dienstprogramm IMPORT zulässiges Format
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.

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
```

Anmerkung:

1. Bei ASC-Dateien wird davon ausgegangen, dass sie keine Spaltennamen enthalten.
2. Zeichenfolgen sind *nicht* in Begrenzer eingeschlossen. Der Datentyp einer Spalte in der ASC-Datei wird durch den Datentyp der Zielspalte in der Datenbanktafel festgelegt.
3. 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 Endpositionspaar 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 enthalten.
 - Die LOAD-Option NULL INDICATORS wird verwendet, und die Nullanzeiger-spalte 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.

PC-Version des IXF-Dateiformats

Die PC-Version des IXF-Dateiformats PC/IXF ist eine Anpassung der IXF-Datenaustauscharchitektur (Integration Exchange Format) für Datenbankmanager. 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 Anwendungssätze vom Satztyp A enthalten. Diese Datensätze sind in PC/IXF-Dateien zulässig, sodass 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 werden, 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 mithilfe 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.

Der Eintrag des D-Datensatzes für jede XML-Spalte enthält zwei Byte Little Endian-Daten zur Angabe der Länge der XML-Datenkennung (XDS), gefolgt von der XDS selbst.

Beispiel: Die XDS

```
XDS FIL="a.xml" OFF="1000" LEN="100" SCH="RENATA.SCHEMA" />
```

wird von den folgenden Byte in einem D-Datensatz dargestellt:

```
0x3D 0x00 XDS FIL="a.xml" OFF="1000" LEN="100" SCH="RENATA.SCHEMA" />
```

PC/IXF-Dateisätze bestehen aus Feldern, die Zeichendaten enthalten. Die Dienstprogramme IMPORT und EXPORT interpretieren diese Zeichendaten mithilfe 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, IXFTCNT, IXFTNAML,
IXFCLENG, IXFCDRID, IXFCPOSN, IXFCNAML, IXFCTYPE,
IXFCSBCP, IXFCBCP, IXFCNDIM, IXFCDSIZ, IXFDRID
```

Anmerkung: Das PC/IXF-Dateiformat des Datenbankmanagers ist nicht identisch mit dem Format von IBM System/370.

PC/IXF-Satztypen:

DB2 verwendet fünf PC/IXF-Basissatztypen und sieben Subtypen von Anwendungssätzen.

Es gibt fünf PC/IXF-Basissatztypen:

- Kopfdatensatz
- Tabellensatz
- Spaltendeskriptorsatz
- Datensatz
- Anwendungssatz

Außerdem werden von DB2 sieben Subtypen von Anwendungssätzen verwendet:

- Indexsatz
- Hierarchiesatz
- Satz für untergeordnete Tabelle
- Fortsetzungssatz
- Beendigungssatz
- Identitätssatz
- DB2-SQL-Kommunikationsbereich

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
IXFHDBCP	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 sechs 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 sechs 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 Identifikations-

zwecken 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 *jjjmmmtt*.

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
IXFTQULL	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 sechs 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 sechs 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.

IXFTCCNT

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 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

FELDNAME	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
IXFCKPOS	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
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 sechs 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 sechs 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 IXFCNAME-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 IXFCSB-CP-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. Ab Version 9.7 enthält dieses Feld beim Datentyp für Zeitmarken (TIMESTAMP) die rechtsbündige Zeichendarstellung einer ganzen Zahl (Integer), die die Genauigkeit der Zeitmarke angibt.

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.

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 muss null sein), hat dieses Feld die Länge null und ist eigentlich gar nicht vorhanden.

DATENSATZ

FELDNAME	LÄNGE	TYP	KOMMENTARE
IXFDRECL	06 BYTE	CHARACTER	Satzlänge
IXFDRECT	01 BYTE	CHARACTER	Satztyp = "D"
IXFDRID	03 BYTE	CHARACTER	Satzkennung "D"
IXDFIL1	04 BYTE	CHARACTER	reserviert
IXFDCOLS	variabel	variabel	Spaltendaten

Die folgenden Felder sind in Datensätzen enthalten:

IXFDRECL

Der Satzlängenanzeiger. Es handelt sich dabei um eine sechs 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 sechs 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 Satzkennung, 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.

IXDFFIL1

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'-Datensatzes) besteht aus mindestens einem Spalteneintrag. Es gibt einen Spalteneintrag für jeden Spaltendeskriptorsatz, der die gleiche 'D'-Satzkennung hat wie der 'D'-Datensatz. Im 'D'-Datensatz wird die Anfangsposition der Spalteneinträge durch den IXFCPOSN-Wert in 'C'-Datensätzen 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 zwei Byte große ganze Zahlen in einem Format, das vom IXFT-MFRM-Feld festgelegt wird.

Die Länge des Datenbereichs eines 'D'-Datensatzes darf 32.771 Byte nicht überschreiten.

ANWENDUNGSSATZ

FELDNAME	LÄNGE	TYP	KOMMENTARE
IXFARECL	06 BYTE	CHARACTER	Satzlänge
IXFARECT	01 BYTE	CHARACTER	Satztyp = "A"
IXFAAPPID	12 BYTE	CHARACTER	Anwendungskennzeichen
IXFADATA	variabel	variabel	anwendungsspezifische Daten

Die folgenden Felder sind in Anwendungssätzen enthalten:

IXFARECL

Der Satzlängenanzeiger. Es handelt sich dabei um eine sechs 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 sechs 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 Anwendungssatz 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 sechs 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 sechs 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 sechs 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 Anwendungssatz 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, von der dieser A-Datensatz erstellt wird.

IXFAITYP

Gibt an, dass dies der Subtyp 'I' der DB2-Anwendungsdatensätze 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

FELDNAME	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 Datensätze für untergeordnete Tabellen (siehe nachfolgende Liste) 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 sechs 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 sechs 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 Anwendungssatz 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, von der dieser 'A'-Datensatz erstellt wird.

IXFAXTYP

Gibt an, dass dies der Subtyp 'X' der DB2-Anwendungsdatensätze 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.

IXFAYSTR

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

FELDNAME	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 Tabelle 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 Anwendungssatz 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, von der dieser 'A'-Datensatz erstellt wird.

IXFAYTYP

Gibt an, dass dies der Subtyp 'Y' der DB2-Anwendungsdatensätze 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

FELDNAME	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 verteil-

ten 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 sechs 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 sechs 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 Anwendungssatz 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, von der dieser 'A'-Datensatz erstellt wird.

IXFACTYP

Gibt an, dass dies der Subtyp 'C' der DB2-Anwendungsdatensätze 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.

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 Dateiondezeichen, das sich am Ende einer IXF-Datei befindet. DB2-Beendigungssätze enthalten die folgenden Felder:

IXFARECL

Der Satzlängenanzeiger. Es handelt sich dabei um eine sechs 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 sechs 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 Anwendungssatz 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, von der dieser 'A'-Datensatz erstellt wird.

IXFAETYP

Gibt an, dass dies der Subtyp 'E' der DB2-Anwendungsdatensätze 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

FELDDNAME	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
IXFAMINX	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 sechs 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 sechs 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 Anwendungssatz 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, von der dieser A-Datensatz erstellt wird.

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.

DB2-SQLCA-DATENSATZ

FELDNAME	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 = 'A'
IXFADATE	008 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Daten
IXFATIME	006 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Zeit
IXFASLCA	136 BYTE	VARIABLE	sqlca - SQL-Kommunikationsbereich

Um anzuzeigen, dass mit der IXF-Datei die Tabelle in einer nachfolgenden Importoperation nicht erneut erstellt werden kann, wird ein Datensatz diesen Typs verwendet. Weitere Informationen finden Sie in der in IXFASLCA zurückgegebenen Nachricht mit dem entsprechenden Ursachencode.

DB2-SQLCA-Datensätze enthalten die folgenden Felder:

IXFARECL

Der Satzlängenanzeiger. Es handelt sich dabei um eine sechs 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 sechs 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 anzuzeigen, dass es sich um einen Anwendungssatz 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, von der dieser 'A'-Datensatz erstellt wird.

IXFAITYP

Gibt an, dass dies der Subtyp 'A' der DB2-Anwendungsdatensätze 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.

IXFASLCA

Der SQL-Kommunikationsbereich, der die Warnung SQL27984W mit einem Ursachencode enthält, die erläutert, warum die IXF-Datei nicht alle Informationen aufweist, die der Befehl **IMPORT** für das erneute Erstellen der Tabelle benötigt.

PC/IXF-Datentypen:

Tabelle 50. PC/IXF-Datentypen

Name	IXFCTYPE-Wert	Beschreibung
BIGINT	492	Eine 8 Byte große ganze Zahl in dem von IXFTMFRM festgelegten Format. Dieser Wert stellt eine ganze Zahl im Bereich 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. IXFLENG wird nicht verwendet und muss Leerzeichen enthalten.
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 und DBCLOB_LOCATION_SPECIFIER	960, 964, 968	<p>Ein Feld fester Länge, das nicht größer als 255 Byte sein kann. Die LOB-Positionskenung (LLS) liegt in der Codepage vor, die in IXFCSBCP angegeben ist. Wenn IXFCSBCP null ist, besteht die LOB-Positionskenung 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-Positionskenung 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-Positionskenung erstellt wird.</p>

Tabelle 50. 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 01 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 50. 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 $(P+2)/2$. 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>
DECFLOAT	996	<p>Bei einem dezimalen Gleitkommawert handelt es sich um eine Zahl mit Dezimalzeichen gemäß IEEE-Standard 754r. Die Position des Dezimalzeichens wird jeweils im dezimalen Gleitkommawert gespeichert. Dezimale Gleitkommazahlen liegen im Bereich der Zahlen mit einer Dezimalstellengenauigkeit von 16 oder 34 Stellen, und der Exponentenbereich liegt bei 10^{-383} bis 10^{+384} bzw. entsprechend bei 10^{-6143} bis 10^{+6144}. Die Speicherlänge des 16-stelligen Werts beträgt 8 Byte, die Speicherlänge des 34-stelligen Werts 16 Byte.</p>

Tabelle 50. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
FLOATING POINT	480	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.
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. Dieser Wert stellt eine ganze Zahl im Bereich 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.

Tabelle 50. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
LONG VARGRAPHIC	472	Eine Zeichenfolge variabler Länge aus Doppelbytezeichen. Das IXFCLENG-Feld im Spaltendeskriptorsatz gibt die maximale Anzahl von Doppelbytezeichen für die 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.
SMALLINT	500	Eine 2 Byte große ganze Zahl in dem von IXFTMFRM festgelegten Format. Dieser Wert 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 24-Stunden-Zeiteinteilung. Jede Zeitangabe ist eine 8 Byte große Zeichenfolge im ISO-Format: <i>ss.mm.ss</i> . Der Wertebereich für die Stunde liegt zwischen 00 und 24 und der Wertebereich für Minuten und Sekunden zwischen 00 und 59. Bei Stunde 24 sind die Minuten- und Sekundenangaben gleich 00. 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.

Tabelle 50. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
TIMESTAMP	392	Das Datum und die Uhrzeit mit der Genauigkeit von Bruchsekunden. Jede Zeitmarke ist eine Zeichenfolge im Format <i>jjjj-mm-tt-ss.mm.ss.mmmmm</i> (Jahr Monat Tag Stunde Minuten Sekunden Bruchsekunden). Ab Version 9.7 ist die Genauigkeit für Zeitmarken im Feld IXFLENG des Spaltendeskriptorsatzes enthalten und darf 12 nicht überschreiten. Vor Version 9.7 wird IXFLENG 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 IXFLENG-Feld des Spaltendeskriptorsatzes 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.
VARGRAPHIC	464	Eine Zeichenfolge variabler Länge aus Doppelbytezeichen. Das IXFLENG-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-Endezeichen.

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 51. 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)
Anmerkung: x und y sind nicht 0.		

PC/IXF-Datentypbeschreibungen:

Tabelle 52. Gültige Datentypformate für das PC/IXF-Dateiformat

Datentyp	Format in vom Dienstprogramm EX- PORT erstellten Da- teien	Für das Dienstprogramm IMPORT zulässi- ges 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 von -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.	Eine PC/IXF-Spalte des Typs CHAR, VARCHAR, LONG VARCHAR, BLOB, BLOB_FILE oder BLOB_LOCATION_SPECIFIER ist in folgenden Fällen zulässig: <ul style="list-style-type: none"> • Die Datenbankspalte weist die Markierung FOR BIT DATA auf. • 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.

Tabelle 52. Gültige Datentypformate für das PC/IXF-Dateiformat (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
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"> • Die Datenbankspalte weist die Markierung FOR BIT DATA auf. • 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. <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>
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.	<p>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.</p>
DATE	Eine DATE-Spalte wird erstellt, die mit der Datenbankspalte identisch ist.	<p>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.</p>

Tabelle 52. Gültige Datentypformate für das PC/IXF-Dateiformat (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
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.
DECFLOAT	Eine DECFLOAT-Spalte, die mit der Datenbankspalte identisch ist, wird erstellt. Die Anzahl der Kommastellen in der Spalte werden im Spaltendeskriptorsatz gespeichert.	Spalten des Typs SMALLINT, INTEGER, BIGINT (nur in DECFLOAT(34)), DECIMAL, FLOAT, REAL, DOUBLE oder DECFLOAT(16) (nur in DECFLOAT(34)) werden akzeptiert. Andere numerische Spaltentypen sind für DECFLOAT gültig, übersteigt die Anzahl der Kommastellen jedoch die Dezimalgenauigkeit der Zielspalte, wird der betreffende Wert gerundet.
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.
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 von -2 147 483 648 bis 2 147 483 647 liegen.

Tabelle 52. Gültige Datentypformate für das PC/IXF-Dateiformat (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
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"> • Die Datenbankspalte weist die Markierung FOR BIT DATA auf. • 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. <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.
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 von -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.

Tabelle 52. Gültige Datentypformate für das PC/IXF-Dateiformat (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
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 = 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"> • Die Datenbankspalte weist die Markierung FOR BIT DATA auf. • 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 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.
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 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.

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 (IXF-HID = "IXF"). IXF-Dateien anderer Formate können nicht importiert werden.
- Das Dienstprogramm IMPORT weist eine PC/IXF-Datei mit mehr als 1024 Spalten zurück.
- Ist der Umfang von Kennungen bei einem Export in das IXF-Format größer, als es das IXF-Format unterstützt, ist die Exportoperation zwar erfolgreich, die bei diesem Export entstandene Datendatei kann jedoch nicht für nachfolgende Importoperationen im Modus CREATE verwendet werden. Es wird die Nachricht SQL27984W zurückgegeben.

Anmerkung: Die Optionen CREATE und REPLACE_CREATE des Befehls IMPORT werden nicht weiter unterstützt und in zukünftigen Releases möglicherweise entfernt.

- 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 Tabellen
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 Tabellen
Der Import einer PC/IXF-Datei in eine *vorhandene* Tabelle wird durch die Schlüsselwörter INSERT, INSERT_UPDATE, REPLACE oder REPLACE_CREATE 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 nächsten PC/IXF-Zeile fortgesetzt.

Datentypbezogene 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 $254 \leq n \leq 4001$ gilt) in eine PC/IXF-Spalte LONG VARCHAR mit 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 $127 \leq n \leq 2001$ 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 53 und Tabelle 54 bieten eine Übersicht über den Import von PC/IXF-Dateien in neue oder vorhandene Datenbanktabellen ohne Verwendung der Option FORCEIN.

Tabelle 53. Übersicht über den Import von PC/IXF-Dateien ohne Verwendung der Option FORCEIN - numerische Datentypen

PC/IXF- SPALTENDATENTYP	SPALTENDATENTYP DER DATENBANK					
	SMALL INT	INT	BIGINT	DEC	DFP	FLT
-SMALLINT	N					
	V	V	V	V ^a	V	V
-INTEGER		N				
	V ^a	V	V	V ^a	V	V
-BIGINT			N			
	V ^a	V ^a	V	V ^a	V	V
-DECIMAL				N		
	V ^a	V ^a	V ^a	V ^a	V	V
-DECFLOAT						
	V ^a	V ^a	V ^a	V ^a	V	V ^a
-FLOAT						N
	V ^a	V ^a	V ^a	V ^a	V	V

^a Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich für den numerischen Zieldatentyp liegen.

Tabelle 54. Übersicht über den Import von PC/IXF-Dateien ohne Verwendung der Option FORCEIN - Zeichendatentypen, grafische Datentypen und Datentypen für Uhrzeit/Datum

PC/IXF- SPALTENDATENTYP	SPALTENDATENTYP DER DATENBANK						
	(0,0)	(SBCS, 0) ^d	(SBCS, DBCS) ^b	GRAPH ^b	DATE	TIME	TIME STAMP
-(0,0)	N						
	V				V ^c	V ^c	V ^c
-(SBCS,0)		N	N				

Tabelle 54. Übersicht über den Import von PC/IXF-Dateien ohne Verwendung der Option FORCEIN - Zeichendatentypen, grafische Datentypen und Datentypen für Uhrzeit/Datum (Forts.)

PC/IXF- SPALTENDATENTYP	SPALTENDATENTYP DER DATENBANK						
	(0,0)	(SBCS, 0) ^d	(SBCS, DBCS) ^b	GRAPH ^b	DATE	TIME	TIME STAMP
	V	V	V		V ^c	V ^c	V ^c
-(SBCS, DBCS)			N		V ^c	V ^c	V ^c
	V		V				
-GRAPHIC				N			
	V			V			
-DATE					N		
					V		
-TIME						N	
						V	
-TIME STAMP							N
							V

^b Datentyp ist nur in DBCS-Umgebungen verfügbar.

^c Einzelne Werte werden zurückgewiesen, wenn sie keine gültigen Datums- oder Uhrzeitwerte darstellen.

^d Datentyp ist in DBCS-Umgebungen nicht verfügbar.

Anmerkung:

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).
2. Zeichenfolgedatentypen werden anhand von Codepage-Attributen unterschieden. Diese Attribute werden als geordnetes Paar (SBCS,DBCS) angezeigt, wobei Folgendes gilt:
 - SBCS ist entweder null oder gibt einen Wert ungleich null eines Einzelbyte-Codepage-Attributs des Zeichendatentyps an.
 - DBCS ist entweder null oder gibt einen Wert ungleich null eines Doppelbyte-Codepage-Attributs des Zeichendatentyps an.
3. Wenn die Tabelle angibt, dass eine PC/IXF-Zeichenspalte in eine Datenbankzeichenspalte importiert werden kann, entsprechen die Werte der jeweiligen Codepage-Attributepaare den Regeln, die für die Gleichheit von Codepages gelten.

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.
- Anwendungssätze (A) sind nach dem H-Datensatz in einer PC/IXF-Datei zulässig. Sie werden nicht gezählt, wenn der Wert des Feldes IXFHHCNT 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.

Option FORCEIN:

Der Änderungswert `forcein` für den Dateityp 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 Semantikregeln für `forcein`

Für die Verwendung des Änderungswerts für den Dateityp `forcein` in einer SBCS- oder DBCS-Umgebung gilt die folgende allgemeine Semantik:

- Der Änderungswert `forcein` für den Dateityp 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 dem Änderungswert `lobsinfile` für den Dateityp exportiert und die Dateien auf einen anderen Client mit einer anderen Codepage ver-

setzt 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.

Codepage-Semantikregeln für forcein

Für die Verwendung des Änderungswerts `forcein` für den Dateityp in einer SBCS- oder DBCS-Umgebung gilt die folgende Codepage-Semantik:

- Der Änderungswert `forcein` für den Dateityp 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.
- `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.
- `forcein` impliziert keine Codepage-Umsetzung.
Werte mit Datentypen, auf die sich der Änderungswert `forcein` für den Dateityp auswirkt, 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 `forcein` in eine *vorhandene* Tabelle importiert werden, gilt Folgendes:
 - Der Codepage-Wert der Zieldatenbanktabelle und -spalten ist immer vorrangig.
 - Der Codepage-Wert der PC/IXF-Datei und -Spalten wird ignoriert.Diese Regel gilt unabhängig davon, ob `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 von `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 IXFCBCP = "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.

An den folgenden Tabellen können Sie Folgendes ablesen:

- 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 55. Übersicht über die Codepage-Semantik des Dienstprogramms IMPORT (neue Tabelle) für SBCS. 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 von forcein funktionieren.

CODEPAGE-ATTRIBUTE DES PC/IXF- DATENTYPS	CODEPAGE-ATTRIBUTE DER DATENBANK- TABELLEN- SPALTE	
	Ohne forcein	Mit forcein
(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)

Anmerkung:
1. Siehe Anmerkungen zu Tabelle 56.

Tabelle 56. Übersicht über die Codepage-Semantik des Dienstprogramms IMPORT (neue Tabelle) für DBCS. In dieser Tabelle wird davon ausgegangen, dass es keine Konvertierungstabelle zwischen a und x gibt.

CODEPAGE-ATTRIBUTE DES PC/IXF-DATENTYPS	CODEPAGE-ATTRIBUTE DER DATENBANKTABELLENSPALTE	
	Ohne forcein	Mit forcein
(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)

Tabelle 56. Übersicht über die Codepage-Semantik des Dienstprogramms IMPORT (neue Tabelle) für DBCS (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 DATENBANKTABELLENSPALTE	
	Ohne forcein	Mit forcein
(x,b)	zurückweisen	(a,b)
(0,b)	(-,b)	(-,b)
(0,y)	zurückweisen	(-,b)

Anmerkung:

- 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. Das Zeichen '-' steht für einen undefinierten Codepage-Wert.
- 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.
- Nur Zeichen- und Grafikdatentypen sind von der forcein-Codepage-Semantik betroffen.
- 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.
- Der Datentyp des Ergebnisses wird durch die im Abschnitt "Datentypsemantik für forcein" beschriebenen Regeln bestimmt.
- Das Ergebnis zurückweisen ist eine Folge der Regeln für ungültige oder inkompatible Datentypen.

An den folgenden Tabellen können Sie Folgendes ablesen:

- 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 57. Übersicht über die Codepage-Semantik des Dienstprogramms IMPORT (vorhandene Tabelle) für SBCS. In dieser Tabelle wird davon ausgegangen, dass es keine Konvertierungstabelle zwischen a und x gibt.

CODEPAGE-ATTRIBUTE DES PC/IXF- DATENTYPS	CODEPAGE-ATTRIBUTE DER ZIELDATEN- BANKSPALTE	ERGEBNISSE DES IMPORTS	
		Ohne forcein	Mit forcein
(0,0)	(0,0)	akzeptieren	akzeptieren
(a,0)	(0,0)	akzeptieren	akzeptieren

Tabelle 57. Übersicht über die Codepage-Semantik des Dienstprogramms *IMPORT* (vorhandene Tabelle) für *SBCS* (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 ZIELDATENBANKSPALTE	ERGEBNISSE DES IMPORTS	
		Ohne forcein	Mit forcein
(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

Anmerkung:

1. Siehe Anmerkungen zu Tabelle 55 auf Seite 492.
2. Das Ergebnis Null oder zurückweisen ist eine Folge der Regeln für ungültige oder inkompatible Datentypen.

Tabelle 58. Übersicht über die Codepage-Semantik des Dienstprogramms *IMPORT* (vorhandene Tabelle) für *DBCS*. In dieser Tabelle wird davon ausgegangen, dass es keine Konvertierungstabelle zwischen a und x gibt.

CODEPAGE-ATTRIBUTE DES PC/IXF-DATENTYPS	CODEPAGE-ATTRIBUTE DER ZIELDATENBANKSPALTE	ERGEBNISSE DES IMPORTS	
		Ohne forcein	Mit forcein
(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

Tabelle 58. Übersicht über die Codepage-Semantik des Dienstprogramms *IMPORT* (vorhandene Tabelle) für *DBCS* (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 ZIELDATENBANKSPALTE	ERGEBNISSE DES IMPORTS	
		Ohne <i>forcein</i>	Mit <i>forcein</i>
(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
(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

Anmerkung:

1. Siehe Anmerkungen zu Tabelle 55 auf Seite 492.
2. Das Ergebnis Null oder zurückweisen ist eine Folge der Regeln für ungültige oder inkompatible Datentypen.

Datentypsemantikregeln für *forcein*

Der Änderungswert *forcein* für den Datentyp ermöglicht den Import bestimmter PC/IXF-Spalten in Zieldatenbankspalten mit ungleichen oder anderweitig inkom-

patiblen Datentypen. Für die Verwendung von `forcein` in einer SBCS- oder DBCS-Umgebung gilt die folgende Datentypsemantik (sofern nicht anders angegeben):

- In SBCS-Umgebungen ermöglicht `forcein` folgende Importvarianten:
 - 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 dem Änderungswert `forcein` für den Dateityp 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 `forcein` für den Import ungültig.
- In DBCS-Umgebungen ermöglicht `forcein` folgende Importvarianten:
 - 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 59 bietet eine Übersicht des Imports von PC/IXF-Dateien in neue oder vorhandene Datenbanktabellen unter Angabe von `forcein`.

Tabelle 59. Übersicht des Imports von PC/IXF-Dateien mit `forcein`

PC/IXF-SPALTEN-DATENTYP	SPALTENDATENTYP DER DATENBANK											
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) ^e	(SBCS, DBCS) ^b	GRAPH ^b	DATE	TIME	TIME STAMP
-SMALLINT	N											
	V	V	V	V ^a	V							
-INTEGER		N										
	V ^a	V	V	V ^a	V							
-BIGINT			N									
	V ^a	V ^a	V	V ^a	V							
-DECIMAL				N								
	V ^a	V ^a	V ^a	V ^a	V							
-FLOAT					N							
	V ^a	V ^a	V ^a	V ^a	V							
-(0,0)						N						
						V	V mit F	V mit F	V mit F	V ^c	V ^c	V ^c

Tabelle 59. Übersicht des Imports von PC/IXF-Dateien mit forcein (Forts.)

PC/IXF-SPALTEN-DATENTYP	SPALTENDATENTYP DER DATENBANK											
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) ^e	(SBCS, DBCS) ^b	GRAPH ^b	DATE	TIME	TIME STAMP
-(SBCS,0)							N	N				
						V	V	V		V ^c	V ^c	V ^c
-(SBCS, DBCS)							N mit F ^d	N		V ^c	V ^c	V ^c
						V	V mit F	V				
-GRAPHIC							N mit F ^d		N			
						V			V			
-DATE										N		
										V		
-TIME											N	
											V	
-TIME STAMP												N
												V

Anmerkung: Wenn eine a PC/IXF-Spalte nur mit der 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. Der Änderungswert forcein für den Dateityp wirkt sich nur auf die Kompatibilität von Zeichen- und Grafikdatentypen aus.

^a Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich für den numerischen Zieltyp liegen.

^b Datentyp ist nur in DBCS-Umgebungen verfügbar.

^c Einzelne Werte werden zurückgewiesen, wenn sie keine gültigen Datums- oder Uhrzeitwerte darstellen.

^d Gilt nur, wenn der PC/IXF-Quellendatentyp nicht von der Zieldatenbank unterstützt wird.

^e Datentyp ist in DBCS-Umgebungen nicht verfügbar.

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 Arbeitsblattdatensätze, die für alle Lotus-Produkte identisch sind. Dies bedeutet, dass für die vom Datenbankmanager unterstützten Releases der Lotus 1-2-3- und Symphony-Produkte alle Dateinamen mit einer beliebigen Erweiterung aus drei Zeichen akzeptiert werden (z. B. WKS, WK1, WRK, WR1, WJ2).

Anmerkung: Die Unterstützung für dieses Dateiformat ist veraltet und wird in einem zukünftigen Release möglicherweise entfernt. Beginnen Sie mit der Verwendung eines unterstützten Dateiformats anstelle einer WSF-Datei, bevor die Unterstützung nicht mehr gewährleistet ist.

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 möglicherweise einige Daten verloren gehen.

WSF-Dateien verwenden eine Lotus-Codepunktzuordnung, die nicht notwendigerweise mit den vorhandenen Codepages übereinstimmt, die von DB2-Datenbanken 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.

Aspekte von Unicode beim Versetzen von Daten

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.

Die DEL-, ASC- und PC/IXF-Dateiformate werden - wie in diesem Abschnitt noch erläutert wird - bei einer Unicode-Datenbank unterstützt. Das WSF-Format wird nicht unterstützt.

Beim Exportieren von Daten aus einer Unicode-Datenbank in eine ASCII-Datei ohne universelle Zeilenbegrenzer 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 Unicode-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. Sie können die Codepage der Datendatei angeben, indem Sie die Registrierdatenbankvariable **DB2CODEPAGE** auf die Codepage der ASCII-Datendatei setzen oder indem Sie den Änderungswert `codepage` für den Dateityp verwenden.

Der Bereich der gültigen ASCII-Begrenzer für SBCS- und MBCS-Clients entspricht dem Bereich, der gegenwärtig durch IBM DB2 V9.1 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 Unicode-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 Unicode-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. 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 Unicode-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.

Besonderheiten für die Codepages 1394, 1392 und 5488

Mit den Dienstprogrammen `IMPORT`, `EXPORT` und `LOAD` können Daten aus der Codepage GB18030 für Chinesisch (Codepage-IDs 1392 und 5488) und der Codepage ShiftJISX 0213 für Japanisch (Codepage-ID 1394) an Unicode-Datenbanken von DB2 übertragen werden. Zusätzlich können mit dem Dienstprogramm `EXPORT` Daten aus Unicode-Datenbanken von DB2 an Daten übertragen werden, die die Codepage GB18030 oder ShiftJIS X0213 verwenden.

Der folgende Befehl lädt beispielsweise die Datendatei `u/jp/user/x0213/data.del` mit dem Format Shift JIS X0213, 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.

Da nur Verbindungen zwischen einem Unicode-Client und einem Unicode-Server unterstützt werden, müssen Sie entweder einen Unicode-Client verwenden oder die DB2-Registrierdatenbankvariable `DB2CODEPAGE` auf den Wert 1208 setzen, bevor Sie das Dienstprogramm `LOAD`, `IMPORT` oder `EXPORT` verwenden.

Eine Konvertierung aus der Codepage 1394 in Unicode kann zu einer Vergrößerung des Datenumfangs führen. 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-Byte aufnehmen zu können.

Inkompatibilität

Bei Anwendungen, die mit einer Unicode-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 Unicode verwenden, liegen die Grafikzeichenfolgedaten entweder in der DBCS-Codepage der Anwendung vor oder sind nicht zulässig, falls die Anwendungscodpage SBCS ist. Beispiel: Wenn ein 932-Client mit einer Datenbank verbunden ist, die ein anderes Format als Unicode für Japanisch verwendet, liegen die Grafikzeichenfolgedaten in der Codepage 301 vor. Bei Anwendungen des 932-Clients, die mit einer Unicode-Datenbank verbunden sind, liegen die Grafikzeichenfolgedaten in UCS-2-Codierung vor.

Zeichensatz und NLS

Die DB2-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 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 für den Dateityp angegeben ist, wird vorab kein Vergleich durchgeführt, und die Daten werden nach Möglichkeit importiert bzw. geladen. Wenn die Daten nach der Umsetzung zu lang sind, wird die entsprechende Zeile zurückgewiesen. Ansonsten werden die Daten importiert bzw. geladen.

Versetzen von XML-Daten

Unterstützung für das Versetzen von XML-Daten wird durch die Dienstprogramme LOAD, IMPORT und EXPORT bereitgestellt. Unterstützung für das Versetzen von Tabellen, die XML-Spalten enthalten, ohne in den Offlinestatus zu wechseln, wird durch die gespeicherte Prozedur ADMIN_MOVE_TABLE bereitgestellt.

Importieren von XML-Daten

Das Dienstprogramm IMPORT kann zum Einfügen von XML-Dokumenten in eine reguläre, relationale Tabelle verwendet werden. Nur korrekt formatierte XML-Dokumente können importiert werden.

Verwenden Sie die Option XML FROM des Befehls IMPORT zum Angeben der Speicherposition der XML-Dokumente, die importiert werden sollen. Die Option XMLVALIDATE gibt an, wie die importierten Dokumente auf ihre Gültigkeit überprüft werden sollen. Sie können auswählen, ob die importierten XML-Daten auf der Basis des im Befehl IMPORT angegebenen Schemas, auf der Basis eines Schemapositionshinweises im XML-Quelldokument oder auf der Basis eines Schemas überprüft werden sollen, das von der XML-Datenkennung in der Hauptdatendatei angegeben wird. Sie können auch die Option XMLPARSE verwenden, um anzugeben, wie Leerzeichen beim Importieren des XML-Dokuments verarbeitet werden sollen. Die Dateitypänderungswerte xmlchar und xmlgraphic ermöglichen Ihnen die Angabe der Codierungsmerkmale für die importierten XML-Daten.

Laden von XML-Daten

Das Dienstprogramm LOAD bietet eine effiziente Methode zum Einfügen großer XML-Datenvolumina in eine Tabelle. Darüber hinaus können mit diesem Dienstprogramm auch bestimmte Optionen verwendet werden, die beim Dienstprogramm IMPORT nicht verfügbar sind, wie beispielsweise die Möglichkeit zum Laden von Daten aus einem benutzerdefinierten Cursor.

Ebenso wie beim Befehl IMPORT können Sie mit dem Befehl LOAD die Speicherposition der zu ladenden XML-Daten, die Prüfungsoptionen für die XML-Daten sowie die Art der Leerzeichenverarbeitung angeben. Und ebenso wie bei IMPORT können Sie die Dateitypänderungswerte `xmlchar` und `xmlgraphic` verwenden, um die Codierungsmerkmale für die geladenen XML-Daten anzugeben.

Exportieren von XML-Daten

Daten können aus Tabellen exportiert werden, die eine oder auch mehrere Spalten mit dem XML-Datentyp umfassen. Exportierte XML-Daten werden separat von der Hauptdatendatei gespeichert, die die exportierten relationalen Daten enthält. Informationen zu den einzelnen exportierten XML-Dokumenten werden in der Hauptdatei der exportierten Daten durch eine XML-Datenkennung (XDS) dargestellt. Die XML-Datenkennung besteht aus einer Zeichenfolge, in der der Name der Systemdatei angegeben ist, in der das XML-Dokument gespeichert ist. Darüber hinaus enthält diese Zeichenfolge auch die exakte Position und Länge des XML-Dokuments innerhalb dieser Datei sowie das XML-Schema, das zur Gültigkeitsprüfung des XML-Dokuments verwendet wird.

Sie können die Parameter `XMLFILE`, `XML TO` und `XMLSAVESCHEMA` des Befehls EXPORT verwenden, um detaillierte Informationen zur Speicherung exportierter XML-Dokumente anzugeben. Die Dateitypänderungswerte `xmlinsepfiles`, `xmlno-declaration`, `xmlchar` und `xmlgraphic` ermöglichen Ihnen die Angabe weiterführender Details zur Speicherposition und zur Codierung der exportierten XML-Daten.

Versetzen von Tabellen im Onlinestatus

Mithilfe der gespeicherten Prozedur `ADMIN_MOVE_TABLE` werden die Daten in einer aktiven Tabelle in ein neues Tabellenobjekt mit demselben Namen versetzt, wobei die Daten online und im Zugriff bleiben. Die Tabelle kann eine oder auch mehrere Spalten mit dem XML-Datentyp enthalten. Versetzen Sie Tabellen online statt offline, wenn die Verfügbarkeit für Sie eine höhere Priorität hat als Kosten, Speicherplatz, Versetzungsleistung und Transaktionsaufwand.

Sie können die Prozedur einmal oder mehrmals aufrufen, wobei die Prozedur für jede Operation einen eigenen Aufruf ausführt. Wenn Sie mehrere Aufrufe verwenden, stehen Ihnen weitere Optionen zur Verfügung. Sie können beispielsweise das Versetzen abbrechen oder bestimmen, wann die Zieltabelle zur Aktualisierung in den Offlinestatus wechseln soll.

Versetzen von XML-Daten - zentrale Aspekte

Beim Importieren oder Exportieren von XML-Daten ist eine Reihe von Einschränkungen, Voraussetzungen und Erinnerungen zu berücksichtigen. Prüfen Sie diese Aspekte, bevor Sie XML-Daten importieren oder exportieren.

Berücksichtigen Sie die folgenden Aspekte, wenn Sie XML-Daten exportieren oder importieren:

- Exportierte XML-Daten werden immer separat von der Hauptdatendatei gespeichert, die die exportierten relationalen Daten enthält.
- Standardmäßig schreibt das Dienstprogramm EXPORT XML-Daten im Unicode-Format. Verwenden Sie den Dateitypänderungswert `xmlchar`, um XML-Daten in der Zeichencodepage zu schreiben, oder den Dateitypänderungswert `xmlgraphic`, um XML-Daten in UTF-16 (der grafischen Codepage) zu schreiben, unabhängig von der Anwendungscodepage.

- XML-Daten können in Nicht-Unicode-Datenbanken gespeichert werden und die in eine XML-Spalte einzufügenden Daten werden vor dem Einfügen aus der Datenbankcodepage in UTF-8 konvertiert. Um während des XML-Parsings die Einführung von Substitutionszeichen zu vermeiden, sollten einzufügende Zeichendaten ausschließlich aus Codepunkten bestehen, die Teil der Datenbankcodepage sind. Bei Einstellung des Konfigurationsparameters `enable_xmlchar` auf `no` wird das Einfügen von Zeichendatentypen während des XML-Parsings blockiert und so das Einfügen auf Datentypen beschränkt, die keiner Codepagekonvertierung unterzogen werden, beispielsweise BIT DATA, BLOB und XML.
- Beim Importieren oder Laden von XML-Daten wird davon ausgegangen, dass die XML-Daten im Unicode-Format codiert sind, sofern das zu importierende XML-Dokument keinen Deklarationstag enthält, der ein Codierungsattribut umfasst. Sie können den Dateitypänderungswert `xmlchar` verwenden, um anzugeben, dass die zu importierenden XML-Dokumente in der Zeichencodepage codiert werden, während der Dateitypänderungswert `xmlgraphic` angibt, dass die zu importierenden XML-Dokumente in UTF-16 codiert werden.
- Zeilen, die nicht korrekt formatierte Dokumente enthalten, werden von den Dienstprogrammen `IMPORT` und `LOAD` zurückgewiesen.
- Wenn für die Dienstprogramme `IMPORT` und `LOAD` die Option `XMLVALIDATE` angegeben wird, werden Dokumente, die erfolgreich anhand ihres übereinstimmenden Schemas überprüft werden konnten, während des Einfügens in eine Tabelle mit Annotationen versehen, die die Schemadaten enthalten. Zeilen, die Dokumente enthalten, deren Gültigkeitsprüfung auf der Basis des zugeordneten Schemas fehlgeschlagen ist, werden zurückgewiesen.
- Wenn für ein Dienstprogramm `LOAD` oder `IMPORT` die Option `XMLVALIDATE` angegeben wird und mehrere XML-Schemata zur Prüfung von XML-Dokumenten verwendet werden, müssen Sie unter Umständen den Konfigurationsparameter `catalogcache_sz` für die Katalogcachegröße erhöhen. Ist die Erhöhung von `catalogcache_sz` nicht möglich, können Sie den integrierten Import- oder Ladebefehl in mehrere Befehle aufteilen, die weniger Schemadokumente verwenden.
- Wenn Sie beim Exportieren von XML-Daten eine XQuery-Anweisung angeben, können Sie XDM-Instanzen (XDM = XQuery and XPath Data Model; XQuery- und XPath-Datenmodell) exportieren, bei denen es sich nicht um korrekt formatierte XML-Dokumente handelt. Exportierte XML-Dokumente, die nicht korrekt formatiert sind, können nicht direkt in eine XML-Spalte importiert werden, weil Spalten, für die der XML-Datentyp definiert wurde, nur vollständige, korrekt formatierte XML-Dokumente enthalten dürfen.
- Die Einstellung `CPU_PARALLELISM` wird während einer Ladeoperation auf den Wert 1 reduziert, wenn Statistikdaten erfasst werden.
- Für eine XML-Ladeoperation muss gemeinsam genutzter Sortierspeicher verwendet werden, um weiterarbeiten zu können. Aktivieren Sie den Parameter `SHEAPTHRES_SHR` oder `INTRA_PARALLEL` oder den Verbindungskonzentrator. Der Parameter `SHEAPTHRES_SHR` wird standardmäßig eingestellt, damit gemeinsam genutzter Sortierspeicher für die Standardkonfiguration verfügbar ist.
- Sie können die Option `SOURCEUSEREXIT` oder den Parameter `SAVECOUNT` des Befehls `LOAD` nicht angeben, wenn Sie eine Tabelle laden, die eine XML-Spalte enthält.
- Ebenso wie LOB-Dateien müssen auch XML-Dateien bei der Verwendung des Befehls `LOAD` auf der Serverseite gespeichert werden.
- Wenn Sie XML-Daten in einer Umgebung mit partitionierten Datenbanken auf mehrere Datenbankpartitionen laden, müssen alle Datenbankpartitionen Zugriff

auf die Dateien haben, die die XML-Daten enthalten. Sie können den Zugriff auf die Dateien beispielsweise ermöglichen, indem Sie die Dateien kopieren oder einen NFS-Mount erstellen.

Verhalten von LOB- und XML-Dateien hinsichtlich IMPORT und EXPORT

LOB- und XML-Dateien weisen im Hinblick auf ihr Verhalten und die Kompatibilität bestimmte Gemeinsamkeiten auf, die beim Importieren (IMPORT) und Exportieren (EXPORT) von Daten genutzt werden können.

Export Wenn beim Exportieren von Daten mit der Option LOBS TO mindestens ein LOB-Pfad angegeben wird, dann schreibt das Dienstprogramm EXPORT reihum die aufeinander folgenden LOB-Werte in die entsprechenden LOB-Dateien. Wird mit der Option XML TO mindestens ein XML-Pfad angegeben, dann schreibt das Dienstprogramm EXPORT ebenfalls reihum alle aufeinander folgenden XDM-Instanzen (XDM = XQuery- und XPath-Datenmodell) in die entsprechenden XML-Dateien. Standardmäßig werden LOB-Werte und XDM-Instanzen in den Pfad geschrieben, in dem auch die exportierten relationalen Daten abgelegt werden. Sofern nicht der Dateitypmodifikator OBSINSEPPFILES oder XMLINSEPPFILES definiert wurde, sind sowohl für LOB- als auch für XML-Dateien mehrere verknüpfte Werte für dieselbe Datei zulässig.

Die Option LOBFILE bietet eine Möglichkeit zur Angabe des Basisdateinamens der LOB-Dateien, die vom Dienstprogramm EXPORT erstellt wurden. In ähnlicher Weise bietet die Option XMLFILE die Möglichkeit, den Basisdateinamen der XML-Dateien anzugeben, die vom Dienstprogramm EXPORT generiert wurden. Der Standardbasisdateiname der LOB-Datei stimmt mit dem Namen der exportierten Datendatei überein, verfügt jedoch über die Erweiterung `.lob`. Der Standardbasisdateiname der XML-Datei stimmt mit dem Namen der exportierten Datendatei überein, verfügt jedoch über die Erweiterung `.xml`. Der vollständige Name der exportierten LOB- oder XML-Datei besteht deshalb aus dem Basisdateinamen und einer numerischen Erweiterung, die auf drei Ziffern aufgefüllt wird, sowie der Erweiterung `.lob` oder `.xml`.

Import

Beim Importieren von Daten ist ein LLS (LOB Location Specifier) mit einer XML-Zielspalte und eine XML-Datenkennung (XDS) mit einer LOB-Zielspalte kompatibel. Wird die Option LOBS FROM nicht angegeben, dann geht das System davon aus, dass die zu importierenden LOB-Dateien sich im selben Pfad wie die relationale Eingabedatendatei befinden. Wenn die Option XML FROM nicht angegeben wurde, geht das System davon aus, dass die zu importierenden XML-Dateien sich im selben Pfad wie die relationalen Eingabedatendatei befinden.

Beispiele zum Exportieren

In dem folgenden Beispiel werden alle LOB-Werte in die Datei `/mypath/t1export.del.001.lob` geschrieben, und alle XDM-Instanzen werden in die Datei `/mypath/t1export.del.001.xml` geschrieben:

```
EXPORT TO /mypath/t1export.del OF DEL MODIFIED BY LOBSINFILE
SELECT * FROM USER.T1
```

In dem folgenden Beispiel wird der erste LOB-Wert in die Datei `/lob1/t1export.del.001.lob` und der zweite in die Datei `/lob2/t1export.del.002.lob` geschrieben, während der dritte an die Datei

/lob1/tlexport.del.001.lob und der vierte an die Datei /lob2/tlexport.del.002.lob angehängt wird usw.:

```
EXPORT TO /mypath/tlexport.del OF DEL LOBS TO /lob1,/lob2
MODIFIED BY LOBSINFILE SELECT * FROM USER.T1
```

In dem folgenden Beispiel wird die erste XDM-Instanz in die Datei /xml1/xmlbase.001.xml, die zweite in die Datei /xml2/xmlbase.002.xml, die dritte in die Datei /xml1/xmlbase.003.xml und die vierte in die Datei /xml2/xmlbase.004.xml geschrieben usw.:

```
EXPORT TO /mypath/tlexport.del OF DEL XML TO /xml1,/xml2 XMLFILE xmlbase
MODIFIED BY XMLINSEPFILS SELECT * FROM USER.T1
```

Beispiele zum Importieren

Für eine Tabelle mit dem Namen "mytable", die eine einzige XML-Spalte enthält, und den unten aufgeführten Befehl IMPORT gilt Folgendes:

```
IMPORT FROM myfile.del of del LOBS FROM /lobpath XML FROM /xmlpath
MODIFIED BY LOBSINFILE XMLCHAR replace into mytable
```

Wenn "myfile.del" die folgenden Daten enthält:

```
mylobfile.001.lob.123.456/
```

Das Dienstprogramm IMPORT versucht in diesem Fall, ein XML-Dokument aus der Datei /lobpath/mylobfile.001.lob zu importieren. Hierbei wird an der relativen Dateiposition (Offset) 123 begonnen und mit einer Länge von 456 Byte gearbeitet.

Es wird davon ausgegangen, dass sich die Datei "mylobfile.001.lob" in LOB-Pfad und nicht im XML-Pfad befindet, da auf den Wert mithilfe eines LLS (LOB Location Specifier) und nicht mit einer XML-Datenkennung verwiesen wird.

Außerdem geht das System davon aus, dass das Dokument in der Zeichencodierung codiert ist, da der Dateitypmodifikator XMLCHAR angegeben wurde.

XML-Datenkennung

XML-Daten, die mit den Dienstprogrammen EXPORT, IMPORT und LOAD versetzt werden, müssen in Dateien gespeichert werden, die separat von der Hauptdatendatei abgelegt sind. In der Hauptdatendatei werden die XML-Daten durch eine XML-Datenkennung (XDS = XML Data Specifier) dargestellt.

Die XML-Datenkennung besteht aus einer Zeichenfolge, die durch einen XML-Tag mit dem Namen "XDS" dargestellt wird, und verfügt über Attribute, die Informationen zu den eigentlichen XML-Daten in der Spalte enthalten. Hierzu gehören z. B. Angaben zum Namen der Datei, in der die eigentlichen XML-Daten enthalten sind, sowie Angaben zur relativen Position und Länge der XML-Daten in dieser Datei. Die Attribute der XML-Datenkennung werden im Folgenden beschrieben.

- FIL** Der Name der Datei (File), die die XML-Daten enthält. Sie können keine benannte Pipe angeben. Das Importieren oder Laden von XML-Dokumenten aus einer benannten Pipe wird nicht unterstützt.
- OFF** Die relative Byteadresse (Offset) der XML-Daten in der Datei, die im Attribut FIL angegeben wurde. Hierbei beginnt die relative Adresse bei 0.
- LEN** Die Länge (Length) der XML-Daten, die sich in der im Attribut FIL angegebenen Datei befinden, in Byte.
- SCH** Die vollständig qualifizierte SQL-Kennung des XML-Schemas, das zur Gült-

tigkeitsprüfung dieses XML-Dokuments verwendet wird. Die Schema- und Namenskomponenten der SQL-Kennung werden als Werte für "OBJECT-SCHEMA" und "OBJECTNAME" der Zeile in der Katalogtabelle SYS-CAT.XSROBJECTS gespeichert, die diesem XML-Schema zugeordnet ist.

Die XML-Datenkennung (XDS) wird als Zeichenfeld in der Datendatei interpretiert und unterliegt dem Parsingverhalten für Zeichenspalten des Dateiformats. Beim ASCII-Dateiformat mit Begrenzern (DEL) muss der Zeichenbegrenzer z. B. verdoppelt werden, sofern dieser in der XML-Datenkennung vorhanden ist. Die Sonderzeichen (<, >, &, ', ") in den Attributwerten müssen immer mit einem Escapezeichen versehen werden. Objektnamen, bei denen die Groß-/Kleinschreibung zu beachten ist, müssen zwischen Zeicheneinheiten vom Typ " eingeschlossen werden.

Beispiele

Gehen Sie von einem Attribut vom Typ FIL mit dem Wert abc&"def".del aus. Um diese XMS-Datenkennung in eine ASCII-Datei mit Begrenzern einzufügen, wobei als Zeichenbegrenzer das doppelte Anführungszeichen (") verwendet wird, müssen die doppelten Anführungszeichen (") verdoppelt und Sonderzeichen mit einem Escapezeichen versehen werden:

```
<XDS FIL=""abc&amp;&quot;def&quot;.del"" />
```

Das nachstehende Beispiel zeigt eine XML-Datenkennung (XDS), die in einer ASCII-Datendatei mit Begrenzern angegeben sein kann. XML-Daten werden in der Datei xmldocs.xml.001 gespeichert, wobei bei der relativen Byteadresse 100 begonnen und eine Länge von 300 Byte verwendet wird. Da sich diese XML-Datenkennung in einer ASCII-Datei befindet, bei der als Begrenzer doppelte Anführungszeichen verwendet werden, müssen die doppelten Anführungszeichen innerhalb des XDS-Tags verdoppelt werden.

```
"<XDS FIL = ""xmldocs.xml.001"" OFF=""100"" LEN=""300"" />"
```

Das nachstehende Beispiel zeigt die vollständig qualifizierte SQL-Kennung ANTHONY.purchaseOrderTest. Der Teil der Kennung, bei dem die Groß-/Kleinschreibung zu beachten ist, muss in der XML-Datenkennung zwischen Zeicheneinheiten vom Typ " eingeschlossen werden:

```
"<XDS FIL='/home/db2inst1/xmlload/a.xml' OFF='0' LEN='6758'  
SCH='ANTHONY.&quot;purchaseOrderTest&quot;' />"
```

Abfrage- und XPath-Datenmodell

Auf XML-Daten in einer Datenbank kann entweder über die XQuery-Funktionen, die in SQL bereitgestellt werden, oder durch direktes Aufrufen von XQuery zugegriffen werden. Eine Instanz des XQuery- und XPath-Datenmodells (XDM) kann ein korrekt formatiertes XML-Dokument, eine Folge von Knoten oder atomaren Werten oder eine beliebige Kombination aus Knoten und atomaren Werten sein.

Einzelne XDM-Instanzen können mithilfe des Befehls EXPORT in eine XML-Datei oder auch in mehrere XML-Dateien geschrieben werden.

Anhang A. Unterschiede zwischen den Dienstprogrammen IMPORT und LOAD

Die folgende Tabelle fasst die wichtigen 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 eingeschränkt genutzt. Partitionsinterne Parallelität kann nur durch gleichzeitige Aufrufe des Importdienstprogramms im Modus ALLOW WRITE ACCESS erreicht werden.	Partitionsinterne Parallelität wird genutzt. 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 MQTs (Materialized Query Tables, gespeicherte Abfragetabellen) wird nicht unterstützt.	Das Laden von Daten in MQTs 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 für den Datentyp <code>generatedoverride</code> und <code>identityoverride</code> möglich.
Das Importieren von Daten in Tabellen, Sichten und Kurznamen wird unterstützt.	Daten können nur in Tabellen geladen werden.
Alle Zeilen werden protokolliert.	Es erfolgt nur eine minimale Protokollierung.
Trigger werden unterstützt.	Trigger werden nicht unterstützt.
Wenn die Importoperation unterbrochen wird und ein Wert für <code>commitcount</code> angegeben wurde, ist die Tabelle verwendbar und enthält die Zeilen, die bis zum letzten Commit 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 <code>savecount</code> 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 Backup-Images wiederhergestellt wird.

Dienstprogramm IMPORT	Dienstprogramm LOAD
Der erforderliche Speicherbereich entspricht in etwa dem größten Index plus 10%. Dieser Speicherbereich wird den Tabellenbereichen für temporäre Tabellen 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.
Sie können Daten über DB2 Connect in eine Hostdatenbank importieren.	Es ist nicht möglich, Daten in eine Host-Datenbank zu laden.
Importdateien müssen auf dem Client vorhanden sein, auf dem das Dienstprogramm IMPORT aufgerufen wird.	In Abhängigkeit von den angegebenen Optionen können sich die zu ladenden Dateien oder Pipes entweder auf der/den Datenbankpartition(en) befinden, die die Datenbank enthält/enthalten, oder auf dem Client mit Fernverbindung, von dem aus das Dienstprogramm LOAD aufgerufen wird. Anmerkung: LOBs und XML-Daten können nur serverseitig gelesen werden.
Ein Backup-Image ist nicht erforderlich. Da das Dienstprogramm IMPORT SQL-INSERT-Anweisungen verwendet, wird die Aktivität protokolliert, und es sind keine Backups erforderlich, um diese Operationen bei Fehlern wiederherzustellen.	Ein Backup-Image kann während der Ladeoperation erstellt werden.

Anhang B. Von den Dienstprogrammen 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 Abfrage.
db2uimpm.bnd (RS)	IMPORT. Einfügen von Daten aus der Quellendatendatei in die Zieltabelle, wenn die Option INSERT, REPLACE oder REPLACE_CREATE verwendet wird. Anmerkung: Die Optionen CREATE und REPLACE_CREATE des Befehls IMPORT werden nicht weiter unterstützt und in zukünftigen Releases möglicherweise entfernt.
db2uipkg.bnd (CS)	IMPORT. Prüfen von Bindeoptionen.
db2ucltb.bnd (CS)	LOAD. Ausführen allgemeiner Initialisierungsprozesse für eine Ladeoperation.
db2ulxld.bnd (CS)	LOAD. Verarbeiten der 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.
db2uqtpd.bnd (RR)	IMPORT/EXPORT. Ausführen der Verarbeitung für hierarchische Tabellen.
db2uimtb.bnd (RS)	IMPORT. Ausführen allgemeiner Initialisierungsprozesse für eine Importoperation.
db2uImpInsUpdate.bnd (RS)	IMPORT. Einfügen von Daten aus der Quellendatendatei in die Zieltabelle, wenn die Option INSERT_UPDATE verwendet wird. Kann nicht mit der Option INSERT BUF gebunden werden.
db2uiDescribe.bnd (RS)	IMPORT/EXPORT. Abfragen von Informationen zu der Definition und den Merkmalen einer Tabelle, die verarbeitet wird. Beispielsweise ein Wrappermodul zum Ausführen einer SQL-Operation DESCRIBE für eine Tabelle, die verarbeitet wird.

Anhang C. Lesen von Syntaxdiagrammen

Zur Beschreibung der Syntax wird die folgende Struktur verwendet:

Lesen Sie die Syntaxdiagramme von links nach rechts sowie von oben nach unten, indem Sie dem Pfad des Zeilenverlaufs folgen.

Das Symbol \blacktriangleright kennzeichnet den Beginn eines Syntaxdiagramms.

Das Symbol \longrightarrow gibt an, dass die Syntax in der nächsten Zeile fortgesetzt wird.

Das Symbol \blacktriangleright gibt an, dass die Syntax von der vorherigen Zeile hier fortgesetzt wird.

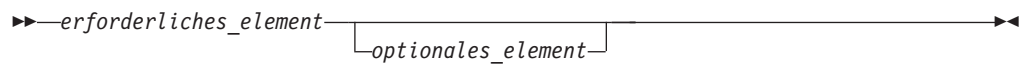
Das Symbol $\longrightarrow\blacktriangleleft$ kennzeichnet das Ende eines Syntaxdiagramms.

Syntaxfragmente beginnen mit dem Symbol \mid und Enden mit dem Symbol \mid .

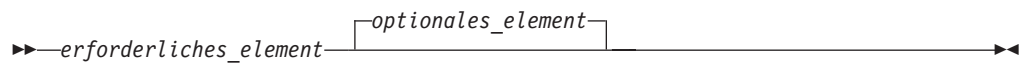
Erforderliche Elemente werden auf der horizontalen Linie (dem Hauptpfad) angegeben.



Optionale Elemente werden unterhalb des Hauptpfades angegeben.

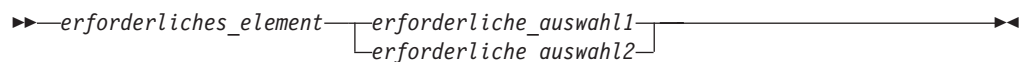


Wird ein optionales Element oberhalb des Hauptpfades angezeigt, hat dieses Element keinen Einfluss auf die Ausführung, sondern wird nur der einfacheren Lesbarkeit halber angegeben.

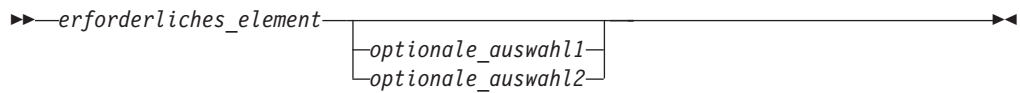


Ist eine Auswahl aus zwei oder mehr Elementen möglich, werden die Optionen in einer Stapelgruppe angezeigt.

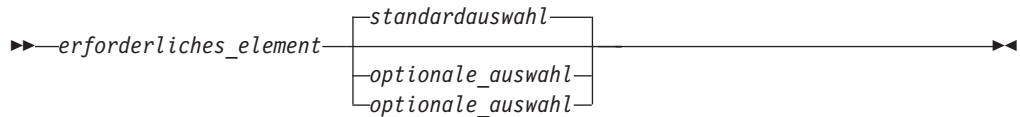
Wenn eines der Elemente ausgewählt werden muss, wird ein Element des Stapels im Hauptpfad angegeben.



Ist die Auswahl eines der Elemente optional, wird der gesamte Stapel unterhalb des Hauptpfades angegeben.



Ist eines der Elemente die Standardauswahl, wird das betreffende Element oberhalb des Hauptpfads angezeigt, und die verbleibenden Auswahlmöglichkeiten werden unterhalb des Hauptpfads angegeben.



Ein nach links zurückweisender Pfeil oberhalb des Hauptpfads gibt ein Element an, das wiederholt werden kann. In diesem Fall müssen wiederholte Elemente durch mindestens ein Leerzeichen voneinander getrennt werden.



Enthält der Wiederholungspfeil ein Komma, müssen die wiederholten Elemente durch ein Komma voneinander getrennt werden.



Ein Wiederholungspfeil oberhalb einer Stapelgruppe gibt an, dass aus den gestapelten Elementen mehr als eine Option ausgewählt werden kann oder dass eine ausgewählte Einzeloption wiederholt werden kann.

Schlüsselwörter werden in Großbuchstaben angegeben (beispielsweise FROM). Ihre Schreibweise muss genau der angegebenen Schreibweise entsprechen. Variablen werden in Kleinbuchstaben angegeben (beispielsweise spal tenname). Sie stehen für Namen oder Werte in der Syntax, die vom Benutzer angegeben werden.

Werden Interpunktionszeichen, runde Klammern, arithmetische Operatoren oder ähnliche Symbole angezeigt, müssen diese als Teil der Syntax eingegeben werden.

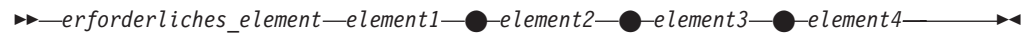
In einigen Fällen steht eine einzelne Variable für ein größeres Fragment der Syntax. Im folgenden Diagramm beispielsweise steht die Variable `parameterblock` für das gesamte Syntaxfragment, das mit **parameterblock** gekennzeichnet ist:



parameterblock:



Benachbarte Segmente zwischen „großen Listenpunkten“ (●) können in beliebiger Reihenfolge angegeben werden.



Das vorstehende Diagramm zeigt, dass 'element2' und 'element3' in beiden möglichen Reihenfolgen angegeben werden können. Die folgenden beiden Reihenfolgen sind gültig:

```
erforderliches_element element1 element2 element3 element4  
erforderliches_element element1 element3 element2 element4
```

Anhang D. Erfassen von Daten für Probleme beim Versetzen von Daten

Wenn bei der Ausführung von Befehlen für das Versetzen von Daten Fehler auftreten und die Ursache des Problems nicht festgestellt werden kann, können Sie Diagnosedaten erfassen, die Sie selbst oder die Mitarbeiter der IBM Unterstützungsfunktion dazu verwenden können, das Problem zu diagnostizieren und zu beheben.

Führen Sie für Ihre Situation geeignete Anweisungen der folgenden Liste zum Erfassen von Daten aus:

- Zum Erfassen von Daten für Probleme beim Befehl **db2move** wechseln Sie in das Verzeichnis, in dem Sie den Befehl abgesetzt haben. Suchen Sie die folgende(n) Datei(en), abhängig von der im Befehl angegebenen Aktion:
 - Suchen Sie für die Aktion COPY nach Dateien mit dem Namen `COPY.zeitmarke.ERR` und `COPYSCHEMA.zeitmarke.MSG`. Wenn Sie darüber hinaus den Modus `LOAD_ONLY` oder `DDL_AND_LOAD` angegeben haben, suchen Sie außerdem nach einer Datei mit dem Namen `LOADTABLE.zeitmarke.MSG`.
 - Suchen Sie für die Aktion EXPORT nach der Datei `EXPORT.out`.
 - Suchen Sie für die Aktion IMPORT nach der Datei `IMPORT.out`.
 - Suchen Sie für die Aktion LOAD nach der Datei `LOAD.out`.
- Zum Erfassen von Daten für Probleme bei den Befehlen **EXPORT**, **IMPORT** oder **LOAD** stellen Sie fest, ob der Befehl den Parameter `MESSAGES` enthielt. War dies der Fall, erfassen Sie die Daten der entsprechenden Ausgabedatei. Diese Dienstprogramme verwenden das aktuelle Verzeichnis und das Standardlaufwerk als Ziel, falls keine anderen Angaben gemacht werden.
- Suchen Sie zum Erfassen von Daten für Probleme beim Befehl **REDISTRIBUTE** nach der Datei `datenbankname.name_der_datenbankpartitionsgruppe.zeitmarke` (Linux und UNIX) bzw. `datenbankname.datenbankname.name_der_datenbankpartitionsgruppe.datum.uhrzeit` (Windows). Sie befindet sich im Verzeichnis `$HOME/sql1lib/db2dump` bzw. `$DB2PATH\sql1lib\redist`, wobei `$HOME` das Ausgangsverzeichnis des Instanzeigners ist.

Anhang E. Übersicht über die technischen Informationen zu DB2

Die technischen Informationen zu DB2 stehen über die folgenden Tools und Methoden zur Verfügung:

- DB2-Informationszentrale
 - Themen (zu Tasks, Konzepten und Referenzinformationen)
 - Hilfe für DB2-Tools
 - Beispielprogramme
 - Lernprogramme
- DB2-Bücher
 - PDF-Dateien (für den Download verfügbar)
 - PDF-Dateien (auf der DB2-PDF-DVD)
 - Gedruckte Bücher
- Befehlszeilenhilfe
 - Hilfe für Befehle
 - Hilfe für Nachrichten

Anmerkung: Die Themen der DB2-Informationszentrale werden häufiger aktualisiert als die PDF- und Hardcopybücher. Um stets die neuesten Informationen zur Verfügung zu haben, sollten Sie die Dokumentationsaktualisierungen installieren, sobald diese verfügbar sind, oder die DB2-Informationszentrale unter ibm.com aufrufen.

Darüber hinaus können Sie auf zusätzliche technische Informationen zu DB2, wie beispielsweise technische Hinweise (Technotes), White Papers und IBM Redbooks, online über ibm.com zugreifen. Rufen Sie die Website 'DB2 Information Management - Software - Library' unter <http://www.ibm.com/software/data/sw-library/> auf.

Feedback zur Dokumentation

Senden Sie uns Ihr Feedback zur DB2-Dokumentation! Wenn Sie Anregungen zur Verbesserung der DB2-Dokumentation haben, senden Sie eine E-Mail an db2docs@ca.ibm.com. Das DB2-Dokumentationsteam bearbeitet das gesamte Feedback, kann jedoch nicht im Einzelnen auf Ihre E-Mails antworten. Nennen Sie uns, wenn möglich, konkrete Beispiele, sodass wir die Problemstellung besser beurteilen können. Wenn Sie uns Feedback zu einem bestimmten Thema oder einer bestimmten Hilfedatei senden, geben Sie den entsprechenden Titel sowie die URL an.

Verwenden Sie diese E-Mail-Adresse nicht, wenn Sie sich an die DB2-Kundenunterstützung wenden möchten. Wenn ein technisches Problem bei DB2 vorliegt, das Sie mithilfe der Dokumentation nicht beheben können, fordern Sie beim zuständigen IBM Service-Center Unterstützung an.

Bibliothek mit technischen Informationen zu DB2 im Hardcopy- oder PDF-Format

Die folgenden Tabellen enthalten eine Beschreibung der DB2-Bibliothek, die im IBM Publications Center unter www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss zur Verfügung steht. Englische Handbücher zu Version 9.7 im PDF-Format können über die folgende Adresse heruntergeladen werden: www.ibm.com/support/docview.wss?uid=swg27015148. Übersetzte DB2-Handbücher im PDF-Format können über die folgende Adresse heruntergeladen werden: www.ibm.com/support/docview.wss?uid=swg27015149.

In den Tabellen sind die Bücher, die in gedruckter Form zur Verfügung stehen, gekennzeichnet; möglicherweise sind diese in Ihrem Land oder Ihrer Region jedoch nicht verfügbar.

Die Formnummer wird bei jeder Aktualisierung eines Handbuchs erhöht. Anhand der nachfolgenden Liste können Sie sicherstellen, dass Sie die jeweils neueste Version des Handbuchs lesen.

Anmerkung: Die *DB2-Informationszentrale* wird häufiger aktualisiert als die PDF- und Hardcopybücher.

Tabelle 60. Technische Informationen zu DB2

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
<i>Administrative API Reference</i>	SC27-2435-03	Ja	Juli 2012
<i>Administrative Routines and Views</i>	SC27-2436-03	Nein	Juli 2012
<i>Call Level Interface Guide and Reference, Volume 1</i>	SC27-2437-03	Ja	Juli 2012
<i>Call Level Interface Guide and Reference, Volume 2</i>	SC27-2438-03	Ja	Juli 2012
<i>Command Reference</i>	SC27-2439-03	Ja	Juli 2012
<i>Dienstprogramme für das Versetzen von Daten - Handbuch und Referenz</i>	SC12-4281-01	Ja	Juli 2012
<i>Datenrecovery und hohe Verfügbarkeit - Handbuch und Referenz</i>	SC12-4282-03	Ja	Juli 2012
<i>Datenbankverwaltung - Konzepte und Konfiguration - Referenzinformationen</i>	SC12-4283-03	Ja	Juli 2012
<i>Datenbanküberwachung - Handbuch und Referenz</i>	SC12-4287-03	Ja	Juli 2012
<i>Datenbanksicherheit</i>	SC12-4285-02	Ja	Juli 2012
<i>DB2 Text Search</i>	SC12-4288-03	Ja	Juli 2012

Tabelle 60. Technische Informationen zu DB2 (Forts.)

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-2444-02	Ja	Juli 2012
<i>Developing Embedded SQL Applications</i>	SC27-2445-02	Ja	Juli 2012
<i>Developing Java Applications</i>	SC27-2446-03	Ja	Juli 2012
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-2447-02	Nein	Juli 2012
<i>Developing User-defined Routines (SQL and External)</i>	SC27-2448-02	Ja	Juli 2012
<i>Getting Started with Database Application Development</i>	GI11-9410-02	Ja	Juli 2012
<i>Installation und Verwaltung von DB2 unter Linux und Windows - Erste Schritte</i>	GI11-3220-00	Ja	August 2009
<i>Globalisierung</i>	SC12-4279-00	Ja	August 2009
<i>DB2-Server - Installation</i>	GC12-4276-03	Ja	Juli 2012
<i>IBM Data Server-Clients - Installation</i>	GC12-4275-02	Nein	Juli 2012
<i>Fehlernachrichten, Band 1</i>	SC12-4295-01	Nein	August 2009
<i>Fehlernachrichten, Band 2</i>	SC12-4296-01	Nein	August 2009
<i>Net Search Extender - Verwaltung und Benutzerhandbuch</i>	SC12-4298-02	Nein	September 2010
<i>Partitionierung und Clustering</i>	SC12-4286-02	Ja	Juli 2012
<i>pureXML - Handbuch</i>	SC12-4293-02	Ja	Juli 2012
<i>Query Patroller - Verwaltung und Benutzerhandbuch</i>	SC12-4304-00	Nein	August 2009
<i>Spatial Extender und Geodetic Data Management Feature - Benutzer- und Referenzhandbuch</i>	SC12-4299-02	Nein	Juli 2012
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-2470-03	Ja	Juli 2012
<i>SQL Reference, Volume 1</i>	SC27-2456-03	Ja	Juli 2012
<i>SQL Reference, Volume 2</i>	SC27-2457-03	Ja	Juli 2012

Tabelle 60. Technische Informationen zu DB2 (Forts.)

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
<i>Fehlerbehebung und Optimieren der Datenbankleistung</i>	SC12-4289-03	Ja	Juli 2012
<i>Upgrade auf DB2 Version 9.7</i>	SC12-4274-03	Ja	Juli 2012
<i>Lernprogramm für Visual Explain</i>	SC12-4290-00	Nein	August 2009
<i>Neuerungen in DB2 Version 9.7</i>	SC12-4291-03	Ja	Juli 2012
<i>Workload-Manager - Handbuch und Referenz</i>	SC12-4292-03	Ja	Juli 2012
<i>XQuery - Referenz</i>	SC12-4294-01	Nein	November 2009

Tabelle 61. Technische Informationen zu DB2 Connect

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
<i>DB2 Connect Personal Edition - Installation und Konfiguration</i>	SC12-4277-03	Ja	Juli 2012
<i>DB2 Connect-Server - Installation und Konfiguration</i>	SC12-4278-03	Ja	Juli 2012
<i>DB2 Connect - Benutzerhandbuch</i>	SC12-4280-02	Ja	Juli 2012

Tabelle 62. Technische Informationen zu Information Integration

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
<i>Information Integration: Föderierte Systeme - Verwaltung</i>	SC12-3759-02	Ja	August 2009
<i>Information Integration: ASNCLP Program Reference for Replication and Event Publishing</i>	SC19-1018-04	Ja	August 2009
<i>Information Integration: Konfiguration föderierter Datenquellen</i>	SC12-3777-02	Nein	August 2009
<i>Information Integration: SQL Replication - Handbuch und Referenz</i>	SC12-3782-02	Ja	August 2009
<i>Information Integration: Replikation und Event Publishing - Einführung</i>	GC12-3779-02	Ja	August 2009

Bestellen gedruckter DB2-Bücher

Informationen zu diesem Vorgang

Gedruckte DB2-Bücher können Sie in den meisten Ländern oder Regionen online bestellen. Das Bestellen gedruckter DB2-Bücher ist stets über den zuständigen IBM Ansprechpartner möglich. Beachten Sie hierbei bitte, dass einige Softcopybücher auf der DVD mit der *DB2-PDF-Dokumentation* nicht in gedruckter Form verfügbar sind. So sind beispielsweise die beiden Bände des Handbuchs *DB2 Fehlernachrichten* nicht in gedruckter Form erhältlich.

Gedruckte Versionen vieler DB2-Bücher, die auf der DVD mit der DB2-PDF-Dokumentation verfügbar sind, können gegen eine Gebühr bei IBM bestellt werden. Abhängig vom jeweiligen Land bzw. der jeweiligen Region können Sie Bücher möglicherweise online über das IBM Publications Center bestellen. Ist im jeweiligen Land bzw. der jeweiligen Region keine Onlinebestellung möglich, können Sie gedruckte DB2-Bücher stets über den zuständigen IBM Ansprechpartner bestellen. Nicht alle Bücher, die auf der DVD mit der DB2-PDF-Dokumentation verfügbar sind, können in gedruckter Form bestellt werden.

Anmerkung: Über <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7> haben Sie Zugriff auf die DB2-Informationszentrale, wo Sie die neueste und umfassendste DB2-Dokumentation finden.

Gehen Sie wie folgt vor, um gedruckte DB2-Bücher zu bestellen:

Vorgehensweise

- Informationen dazu, ob in Ihrem Land oder Ihrer Region die Bestellung von gedruckten DB2-Büchern möglich ist, finden Sie auf der Website mit dem IBM Publications Center unter <http://www.ibm.com/shop/publications/order>. Wählen Sie ein Land, eine Region oder eine Sprache aus, um die Bestellinformationen für Veröffentlichungen aufzurufen, und führen Sie dann die entsprechenden Schritte des Bestellverfahrens für Ihr Land bzw. Ihre Region aus.
- Gehen Sie wie folgt vor, um gedruckte DB2-Bücher beim zuständigen IBM Ansprechpartner zu bestellen:
 1. Kontaktinformationen zum zuständigen Ansprechpartner finden Sie auf einer der folgenden Websites:
 - IBM Verzeichnis weltweiter Kontakte unter www.ibm.com/planetwide.
 - Website mit IBM Veröffentlichungen unter <http://www.ibm.com/shop/publications/order>. Wählen Sie das gewünschte Land, die gewünschte Region oder die gewünschte Sprache aus, um auf die entsprechende Homepage mit Veröffentlichungen Ihres Landes bzw. Ihrer Region zuzugreifen. Folgen Sie auf dieser Seite dem Link für Informationen zu dieser Site ("About this Site").
 2. Geben Sie bei Ihrem Anruf an, dass Sie eine DB2-Veröffentlichung bestellen möchten.
 3. Teilen Sie dem zuständigen Ansprechpartner die Titel und Formularnummern der Bücher mit, die Sie bestellen möchten. Titel und Formularnummern finden Sie unter „Bibliothek mit technischen Informationen zu DB2 im Hardcopy- oder PDF-Format“ auf Seite 518.

Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor

DB2-Produkte geben für Bedingungen, die aufgrund 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 Starten der Hilfe für SQL-Statuswerte müssen Sie den Befehlszeilenprozessor öffnen und Folgendes eingeben:

? *SQL-Status* oder ? *Klassencode*

Hierbei steht *SQL-Status* für einen gültigen fünfstelligen SQL-Statuswert und *Klassencode* für die ersten beiden Ziffern dieses Statuswerts.

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.

Zugriff auf verschiedene Versionen der DB2-Informationenzzentrale

Informationen zu diesem Vorgang

Für Themen aus DB2 Version 9.8 lautet die URL der *DB2-Informationenzzentrale* <http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/>.

Für Themen aus DB2 Version 9.7 lautet die URL der *DB2-Informationenzzentrale* <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>.

Für Themen aus DB2 Version 9.5 lautet die URL der *DB2-Informationenzzentrale* <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>.

Für Themen aus DB2 Version 9.1 lautet die URL der *DB2-Informationenzzentrale* <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>.

Für Themen aus DB2 Version 8 lautet die URL der *DB2-Informationenzzentrale (Version 8, 'Information - Unterstützung')* <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>.

Anzeigen von Themen in der gewünschten Sprache in der DB2-Informationenzzentrale

Informationen zu diesem Vorgang

In der DB2-Informationenzzentrale 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 der DB2-Informationenzzentrale 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 die DB2-Informationszentrale in der gewünschten Sprache anzuzeigen.
- Um Themen in der gewünschten Sprache in einem Firefox- oder Mozilla-Browser anzuzeigen, gehen Sie wie folgt vor:
 1. Wählen Sie den Knopf im Bereich **Languages** des Dialogfensters **Tools** —> **Options** —> **Advanced** 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 zur Liste hinzufügen möchten, klicken Sie den Knopf **Add...** 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 **Move Up** aus, bis die Sprache an erster Stelle in der Liste steht.
 3. Aktualisieren Sie die Seite, um die DB2-Informationszentrale in der gewünschten Sprache anzuzeigen.

Ergebnisse

Bei einigen Kombinationen aus Browser und Betriebssystem müssen Sie auch die Ländereinstellungen des Betriebssystems in die gewünschte Locale und Sprache ändern.

Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten DB2-Informationszentrale

Eine lokal installierte DB2-Informationszentrale muss regelmäßig aktualisiert werden.

Vorbereitende Schritte

Eine DB2-Informationszentrale der Version 9.7 muss bereits installiert sein. Einzelheiten hierzu finden Sie unter „Installation der DB2-Informationszentrale mit dem DB2-Installationsassistenten“ in *DB2-Server - Installation*. Alle für die Installation der Informationszentrale geltenden Voraussetzungen und Einschränkungen gelten auch für die Aktualisierung der Informationszentrale.

Informationen zu diesem Vorgang

Eine vorhandene DB2-Informationszentrale kann automatisch oder manuell aktualisiert werden:

- Automatische Aktualisierungen. Verwenden Sie diese Aktualisierungsmethode zur Aktualisierung vorhandener Komponenten und Sprachen der Informationszentrale. Ein zusätzlicher Vorteil von automatischen Aktualisierungen ist, dass die Informationszentrale während der Aktualisierung nur für einen sehr kurzen Zeitraum nicht verfügbar ist. Darüber hinaus können automatische Aktualisierungen so konfiguriert werden, dass sie als Teil anderer, regelmäßig ausgeführter Stapeljobs ausgeführt werden.
- Manuelle Aktualisierungen. Verwenden Sie diese Aktualisierungsmethode, wenn Sie während des Aktualisierungsprozesses Komponenten oder Sprachen hinzufügen möchten. Beispiel: Eine lokale Informationszentrale wurde ursprünglich sowohl mit englischer als auch mit französischer Sprachunterstützung installiert; nun soll auch die deutsche Sprachunterstützung installiert werden. Bei einer manuellen Aktualisierung werden sowohl eine Installation der deutschen Sprachunterstützung als auch eine Aktualisierung der vorhandenen Komponenten und Sprachen der Informationszentrale durchgeführt. Sie müssen jedoch bei einer manuellen Aktualisierung die Informationszentrale manuell stoppen, aktualisieren und erneut starten. Die Informationszentrale ist während des gesamten Aktualisierungsprozesses nicht verfügbar.

Dieser Abschnitt enthält Details zum Prozess der automatischen Aktualisierung. Anweisungen zur manuellen Aktualisierung finden Sie im Abschnitt „Manuelles Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten DB2-Informationszentrale“.

Vorgehensweise

Gehen Sie wie folgt vor, um die auf Ihrem Computer bzw. Intranet-Server installierte DB2-Informationszentrale automatisch zu aktualisieren:

1. Unter Linux:
 - a. Navigieren Sie zu dem Pfad, in dem die Informationszentrale installiert ist. Standardmäßig ist die DB2-Informationszentrale im Verzeichnis `/opt/ibm/db2ic/V9.7` installiert.
 - b. Navigieren Sie vom Installationsverzeichnis in das Verzeichnis `doc/bin`.
 - c. Führen Sie das Script `update-ic` aus:
`update-ic`
2. Unter Windows:
 - a. Öffnen Sie ein Befehlsfenster.
 - b. Navigieren Sie zu dem Pfad, in dem die Informationszentrale installiert ist. Standardmäßig ist die DB2-Informationszentrale im Verzeichnis `<Programme>\IBM\DB2 Information Center\Version 9.7` installiert, wobei `<Programme>` das Verzeichnis der Programmdateien (Program Files) angibt.
 - c. Navigieren Sie vom Installationsverzeichnis in das Verzeichnis `doc\bin`.
 - d. Führen Sie die Datei `update-ic.bat` aus:
`update-ic.bat`

Ergebnisse

Die DB2-Informationszentrale wird automatisch erneut gestartet. Standen Aktualisierungen zur Verfügung, zeigt die Informationszentrale die neuen und aktualisierten Abschnitte an. Waren keine Aktualisierungen für die Informationszentrale verfügbar, wird eine entsprechende Nachricht zum Protokoll hinzugefügt. Die Protokolldatei befindet sich im Verzeichnis `doc\eclipse\configuration`. Der Name der Protokolldatei ist eine Zufallszahl. Beispiel: `1239053440785.log`.

Manuelles Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten DB2-Informationszentrale

Wenn Sie die DB2-Informationszentrale lokal installiert haben, können Sie Dokumentationsaktualisierungen von IBM abrufen und installieren.

Informationen zu diesem Vorgang

Zur manuellen Aktualisierung der lokal installierten *DB2-Informationszentrale* sind die folgenden Schritte erforderlich:

1. Stoppen Sie die *DB2-Informationszentrale* auf Ihrem Computer und starten Sie die Informationszentrale im Standalone-Modus erneut. Die Ausführung der Informationszentrale im Standalone-Modus verhindert, dass andere Benutzer in Ihrem Netz auf die Informationszentrale zugreifen, und ermöglicht das Anwenden von Aktualisierungen. Die Workstationversion der DB2-Informationszentrale wird stets im Standalone-Modus ausgeführt.
2. Verwenden Sie die Aktualisierungsfunktion, um zu prüfen, welche Aktualisierungen verfügbar sind. Falls Aktualisierungen verfügbar sind, die Sie installieren müssen, können Sie die Aktualisierungsfunktion verwenden, um diese abzurufen und zu installieren.

Anmerkung: Wenn es in der verwendeten Umgebung erforderlich ist, die Aktualisierungen für die *DB2-Informationszentrale* auf einer Maschine zu installieren, die nicht über eine Verbindung zum Internet verfügt, spiegeln Sie die Aktualisierungssite auf ein lokales Dateisystem und verwenden Sie dabei eine Maschine, die mit dem Internet verbunden ist und auf der die *DB2-Informationszentrale* installiert ist. Wenn viele Benutzer Ihres Netzes die Dokumentationsaktualisierungen installieren sollen, können Sie die Zeit, die jeder einzelne Benutzer für die Aktualisierungen benötigt, reduzieren, indem Sie die Aktualisierungssite lokal spiegeln und ein Proxy dafür erstellen. Ist dies der Fall, verwenden Sie die Aktualisierungsfunktion, um die Pakete abzurufen. Die Aktualisierungsfunktion ist jedoch nur im Standalone-Modus verfügbar.

3. Stoppen Sie die im Standalone-Modus gestartete Informationszentrale und starten Sie die *DB2-Informationszentrale* auf Ihrem Computer erneut.

Anmerkung: Unter Windows 2008 und Windows Vista (und neueren Versionen) müssen die in diesem Abschnitt aufgeführten Befehle mit Administratorberechtigung ausgeführt werden. Zum Öffnen einer Eingabeaufforderung oder eines Grafiktools mit vollen Administratorberechtigungen klicken Sie mit der rechten Maustaste die Verknüpfung an und wählen Sie **Als Administrator ausführen** aus.

Vorgehensweise

Gehen Sie wie folgt vor, um die auf Ihrem Computer bzw. Intranet-Server installierte *DB2-Informationszentrale* zu aktualisieren:

1. Stoppen Sie die *DB2-Informationszentrale*.
 - Unter Windows klicken Sie **Start > Einstellungen > Systemsteuerung > Verwaltung > Dienste** an. Klicken Sie mit der rechten Maustaste die **DB2-Informationszentrale** an und wählen Sie **Beenden** aus.
 - Unter Linux: Geben Sie den folgenden Befehl ein:

```
/etc/init.d/db2icdv97 stop
```
2. Starten Sie die Informationszentrale im Standalone-Modus.

- Unter Windows:
 - a. Öffnen Sie ein Befehlsfenster.
 - b. Navigieren Sie zu dem Pfad, in dem die Informationszentrale installiert ist. Standardmäßig ist die *DB2-Informationszentrale* im Verzeichnis *Programme\IBM\DB2 Information Center\Version 9.7* installiert, wobei *Programme* das Verzeichnis der Programmdateien (Program Files) angibt.
 - c. Navigieren Sie vom Installationsverzeichnis in das Verzeichnis *doc\bin*.
 - d. Führen Sie die Datei *help_start.bat* aus:


```
help_start.bat
```
- Unter Linux:
 - a. Navigieren Sie zu dem Pfad, in dem die Informationszentrale installiert ist. Standardmäßig ist die *DB2-Informationszentrale* im Verzeichnis */opt/ibm/db2ic/V9.7* installiert.
 - b. Navigieren Sie vom Installationsverzeichnis in das Verzeichnis *doc/bin*.
 - c. Führen Sie das Script *help_start* aus:


```
help_start
```

Der standardmäßig auf dem System verwendete Web-Browser wird geöffnet und zeigt die Standalone-Informationszentrale an.

3. Klicken Sie den Aktualisierungsknopf (🔄) an. (JavaScript muss im verwendeten Browser aktiviert sein.) Klicken Sie im rechten Fenster der Informationszentrale den Knopf für die Suche nach Aktualisierungen an. Eine Liste der Aktualisierungen für die vorhandene Dokumentation wird angezeigt.
4. Wählen Sie zum Initiieren des Installationsprozesses die gewünschten Aktualisierungen aus und klicken Sie anschließend den Knopf für die Installation der Aktualisierungen an.
5. Klicken Sie nach Abschluss des Installationsprozesses **Fertigstellen** an.
6. Stoppen Sie die im Standalone-Modus gestartete Informationszentrale:
 - Unter Windows: Navigieren Sie in das Verzeichnis *doc\bin* des Installationsverzeichnisses und führen Sie die Datei *help_end.bat* aus:


```
help_end.bat
```

Anmerkung: Die Stapeldatei *help_end* enthält die Befehle, die erforderlich sind, um die Prozesse, die mit der Stapeldatei *help_start* gestartet wurden, ordnungsgemäß zu stoppen. Verwenden Sie nicht die Tastenkombination *Strg+C* oder eine andere Methode, um *help_start.bat* zu stoppen.
 - Unter Linux: Navigieren Sie in das Verzeichnis *doc/bin* des Installationsverzeichnisses und führen Sie das Script *help_end* aus:


```
help_end
```

Anmerkung: Das Script *help_end* enthält die Befehle, die erforderlich sind, um die Prozesse, die mit dem Script *help_start* gestartet wurden, ordnungsgemäß zu stoppen. Verwenden Sie keine andere Methode, um das Script *help_start* zu stoppen.
7. Starten Sie die *DB2-Informationszentrale* erneut.
 - Unter Windows klicken Sie **Start > Einstellungen > Systemsteuerung > Verwaltung > Dienste** an. Klicken Sie mit der rechten Maustaste die **DB2-Informationszentrale** an und wählen Sie **Start** aus.
 - Unter Linux: Geben Sie den folgenden Befehl ein:


```
/etc/init.d/db2icdv97 start
```

Ergebnisse

In der aktualisierten *DB2-Informationen* werden die neuen und aktualisierten Themen angezeigt.

DB2-Lernprogramme

Die DB2-Lernprogramme unterstützen Sie dabei, sich mit den unterschiedlichen Aspekten der DB2-Produkte vertraut zu machen. Die Lerneinheiten bieten eine in einzelne Schritte unterteilte Anleitung.

Vorbereitungen

Die XHTML-Version des Lernprogramms kann über die Informationszentrale unter <http://publib.boulder.ibm.com/infocenter/db2help/> angezeigt werden.

In einigen der Lerneinheiten werden Beispieldaten und Codebeispiele verwendet. Informationen zu bestimmten Voraussetzungen für die Ausführung der Tasks finden Sie in der Beschreibung des Lernprogramms.

DB2-Lernprogramme

Klicken Sie zum Anzeigen des Lernprogramms den Titel an.

„pureXML“ in *pureXML - Handbuch*

Einrichten einer DB2-Datenbank, um XML-Daten zu speichern und Basisoperationen mit dem nativen XML-Datenspeicher auszuführen.

„Visual Explain“ in *Lernprogramm für Visual Explain*

Analysieren, Optimieren und Anpassen von SQL-Anweisungen zur Leistungsverbesserung mithilfe 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-Datenbankprodukten zu unterstützen.

DB2-Dokumentation

Informationen zur Fehlerbehebung stehen im Handbuch *Fehlerbehebung und Optimieren der Datenbankanleistung* oder im Abschnitt mit grundlegenden Informationen zu Datenbanken in der *DB2-Informationen* zur Verfügung. Die Fehlerbehebungsinformationen enthalten Themen, die Sie beim Eingrenzen und Identifizieren von Problemen mithilfe der Diagnosetools und -dienstprogramme von DB2 unterstützen. Darüber hinaus finden Sie hier Lösungen für einige der häufigsten Probleme sowie Hinweise zur Lösung von Problemen, die in den verwendeten DB2-Datenbankprodukten auftreten können.

IBM Support Portal

Im IBM Support Portal finden Sie Informationen zu Problemen und den möglichen Ursachen und Fehlerbehebungsmaßnahmen. Die Website mit technischer Unterstützung enthält Links zu den neuesten DB2-Veröffentlichungen, technischen Hinweisen (TechNotes), APARs (Authorized Program Analysis Reports) und Fehlerkorrekturen, Fixpacks sowie weiteren Ressourcen. Sie können diese Wissensbasis nach möglichen Lösungen für aufgetretene Probleme durchsuchen.

Sie können auf das IBM Support Portal über die folgende Website zugreifen: http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_for_Linux,_UNIX_and_Windows.

Bedingungen

Die Berechtigungen zur Nutzung dieser Veröffentlichungen werden Ihnen auf der Basis der folgenden Bedingungen gewährt.

Persönliche Nutzung: Sie dürfen diese Veröffentlichungen für Ihre persönliche, nicht kommerzielle Nutzung unter der Voraussetzung vervielfältigen, dass alle Eigentumsvermerke erhalten bleiben. Sie dürfen diese Veröffentlichungen oder Teile dieser Veröffentlichungen ohne ausdrückliche Genehmigung von IBM nicht weitergeben, anzeigen oder abgeleitete Werke davon erstellen.

Kommerzielle Nutzung: Sie dürfen diese Veröffentlichungen nur innerhalb Ihres Unternehmens und unter der Voraussetzung, dass alle Eigentumsvermerke erhalten bleiben, vervielfältigen, weitergeben und anzeigen. Sie dürfen diese Veröffentlichungen oder Teile dieser Veröffentlichungen ohne ausdrückliche Genehmigung von IBM außerhalb Ihres Unternehmens nicht vervielfältigen, weitergeben, anzeigen oder abgeleitete Werke davon erstellen.

Abgesehen von den hier gewährten Berechtigungen erhalten Sie keine weiteren Berechtigungen, Lizenzen oder Rechte (veröffentlicht oder stillschweigend) in Bezug auf die Veröffentlichungen oder darin enthaltene Informationen, Daten, Software oder geistiges Eigentum.

IBM behält sich das Recht vor, die in diesem Dokument gewährten Berechtigungen nach eigenem Ermessen zurückzuziehen, wenn sich die Nutzung der Veröffentlichungen für IBM als nachteilig erweist oder wenn die obigen Nutzungsbestimmungen nicht genau befolgt werden.

Sie dürfen diese Informationen nur in Übereinstimmung mit allen anwendbaren Gesetzen und Vorschriften, einschließlich aller US-amerikanischen Exportgesetze und Verordnungen, herunterladen und exportieren.

IBM übernimmt keine Gewährleistung für den Inhalt dieser Informationen. Diese Veröffentlichungen werden auf der Grundlage des gegenwärtigen Zustands (auf "as-is"-Basis) und ohne eine ausdrückliche oder stillschweigende Gewährleistung für die Handelüblichkeit, die Verwendungsfähigkeit oder die Freiheit der Rechte Dritter zur Verfügung gestellt.

Anhang F. Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. Die Informationen über Produkte anderer Hersteller als IBM basieren auf den zum Zeitpunkt der ersten Veröffentlichung dieses Dokuments verfügbaren Informationen und können geändert werden.

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 zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

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 Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuauflage veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt 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 unabhängig voneinander 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
U59/3600
3600 Steeles Avenue East
Markham, Ontario L3R 9Z7
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 Dokument aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten 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 Gewährleistung, 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 davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung kann Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes enthalten. Sie sollen nur die Funktionen des Lizenzprogramms 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 Musteranwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Musterprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Musterprogramme 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. Die Musterprogramme werden ohne Wartung (auf "as-is"-Basis) und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM haftet nicht für Schäden, die durch Verwendung der Musterprogramme entstehen.

Kopien oder Teile der Musterprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Musterprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. *_Jahr/Jahre angeben_*. Alle Rechte vorbehalten.

Marken

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der IBM Corporation in den USA und/oder anderen Ländern. Weitere Produkt- oder Servicennamen können Marken von oder anderen Herstellern sein. IBM oder anderen Herstellern sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" unter www.ibm.com/legal/copytrade.shtml.

Die folgenden Namen sind Marken oder eingetragene Marken anderer Unternehmen.

- Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.
- Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken von Oracle und/oder ihren verbundenen Unternehmen.
- UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.
- Intel, das Intel-Logo, Intel Inside, das Intel Inside-Logo, Intel Centrino, das Intel Centrino-Logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium und Pentium sind Marken oder eingetragene Marken der Intel Corporation oder deren Tochtergesellschaften in den USA oder anderen Ländern.
- Microsoft, Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Weitere Unternehmens-, Produkt- oder Servicennamen können Marken anderer Hersteller sein.

Index

A

ADMIN_CMD, Prozedur
 Befehle
 EXPORT 30
 IMPORT 97
 LOAD 278
Aktualisierungen
 DB2-Informationszentrale 523, 525
Änderungswerte für Dateityp
 EXPORT (Dienstprogramm) 18
 Export-API 41
 IMPORT (Befehl) 70
 Import-API 124
 Lade-API 318
 LOAD (Befehl) 239
 Speicherauszugsdatei 214
Anwendungssätze
 PC/IXF 457
anyorder (Änderungswert für Dateityp) 239, 318
APIs
 db2Export 41
 db2Import 124
 db2Load 318
 sqluexpr 41
 sqluimpr 124
ASC (Importdateityp) 70
ASC-Dateien
 Beispiel 454
 Format 450
ASC-Datentypbeschreibungen 451
ASCII-Dateiformat mit universellen Zeilenbegrenzern (ASC) 450
ASCII-Format ohne universelle Zeilenbegrenzer (DEL)
 Daten zwischen Plattformen versetzen 442
 Übersicht 443
Ausgesetzte E/A
 Übersicht 418
Ausnahmetabellen
 LOAD, Dienstprogramm 208
 SET INTEGRITY (Anweisung) 345
Automatische Wörterverzeichniserstellung (ADC)
 Versetzen von Daten 187

B

Bedingungen
 Veröffentlichungen 528
Beendigung
 Ladeoperationen
 ALLOW READ ACCESS 211
 Umgebungen mit partitionierten Datenbanken 228
 PC/IXF-Sätze 457
Befehle
 db2inidb 419
 db2look
 Details 427
 db2move 386
 db2relocatedb 422
 EXPORT 18, 30
 IMPORT 70, 97

Befehle (*Forts.*)
 LIST TABLESPACES 370
 LOAD 239, 278
 LOAD QUERY 364
 RESTORE DATABASE 397
Begrenzer
 Einschränkungen beim Versetzen von Daten 449
 modifizieren 449
 Zeichenfolge 448
Beispiele
 Dateien
 ASC 454
 DEL 448
Bemerkungen 529
Benutzerdefinierte Datentypen (UDTs)
 einzigartige Typen
 importieren 67
Benutzerexitprogramme
 Daten versetzen 167
Berechtigung LOAD
 Details 146
Berechtigungen
 LOAD 146
Bestellen von DB2-Büchern 521
binarynumerics (Änderungswert für Dateityp)
 db2load (API) 318
 LOAD (Befehl) 239
Bindedateien
 von Dienstprogrammen verwendete 509
Bücher
 bestellen 521

C

chardel (Änderungswert für Dateityp)
 API 'db2Export' 41
 API 'db2Import' 124
 db2Load (API) 318
 EXPORT (Befehl) 18
 IMPORT (Befehl) 70
 LOAD (Befehl) 239
codepage (Änderungswert für Dateityp) 239, 318
Codepages
 API 'db2Export' 41
 API 'db2Import' 124
 Dienstprogramm IMPORT 500
 Dienstprogramm LOAD 500
 EXPORT (Befehl) 18
 IMPORT (Befehl) 70
 konvertieren
 Dateien 485
 PC/IXF-Daten 485
coldel (Änderungswert für Dateityp)
 API 'db2Export' 41
 API 'db2Import' 124
 db2Load (API) 318
 EXPORT (Befehl) 18
 IMPORT (Befehl) 70
 LOAD (Befehl) 239
compound (Änderungswert für Dateityp)
 API 'db2Import' 124

compound (Änderungswert für Dateityp) (Forts.)
 IMPORT (Befehl) 70
CURSOR, Dateityp
 Daten versetzen 160

D

dateformat (Änderungswert für Dateityp)

 API 'db2Import' 124
 db2Load (API) 318
 IMPORT (Befehl) 70
 LOAD (Befehl) 239

Dateiformate

 ASC (ASCII-Format mit universellen Zeilenbegren-
 zern) 450
 CURSOR 160
 Datei in Tabelle importieren 70
 DEL (ASCII-Format ohne universelle Zeilenbegrenzer) 443
 PC/IXF (PC-Version von IXF) 455
 Tabelle in Datei exportieren 18
 WSF (Arbeitsblattformat) 497

Daten

 exportieren 8
 Import 53
 kennsatzbasierte Zugriffssteuerung (LBAC)
 exportieren 7
 übertragen
 zwischen Hosts und Workstations 379
 zwischen Plattformen 442
 versetzen
 Tools 1
 Verteilung
 Daten versetzen 167

Datenbanken

 Daten in Tabellen laden 239
 erneut erstellen
 RESTORE DATABASE (Befehl) 397
 Restore durchführen 397
 von Datei in Tabelle importieren
 API 'db2Import' 124
 IMPORT (Befehl) 70
 von Tabelle in Datei exportieren
 API 'db2Export' 41
 EXPORT (Befehl) 18

Datensatz

 kennsatzbasierte Zugriffssteuerung (LBAC)
 laden 145

Datensätze

 Typen
 PC/IXF 457

Datensatztyp

 PC/IXF 457

Datentypen

 ASC 451
 DEL 445
 PC/IXF 473, 480

DB2 Connect

 Daten versetzen 379

DB2-Informationszentrale

 Aktualisierung 523, 525
 Sprachen 522
 Versionen 522

DB2-Statistiktool und DDL-Extraktionstool (Befehl) 427

db2inidb, Befehl

 Details 419
 Übersicht 418

db2Load (API) 318

DB2LOADREC, Registrierdatenbankvariable
 Recovery von Daten 212

db2look (Befehl)

 Details 427

db2move, Befehl

 Schema kopieren, Beispiele 385

db2move (Befehl)

 Details 386
 Übersicht 1

db2relocatedb (Befehl)

 Details 422
 Übersicht 1

DB2SECURITYLABEL (Datentyp)

 exportieren 12
 Import 61
 laden 153

decplusblank (Änderungswert für Dateityp)

 EXPORT (Befehl) 18
 IMPORT (Befehl) 70
 LOAD (Befehl) 239

decpt (Änderungswert für Dateityp)

 EXPORT (Befehl) 18
 IMPORT (Befehl) 70
 LOAD (Befehl) 239

DEL-Datei

 Beispiel 448
 Format 443

DEL-Datentypbeschreibungen 445

delprioritychar (Änderungswert für Dateityp)

 IMPORT (Befehl) 70
 LBAC-geschützte Daten importieren 61
 LBAC-geschützte Daten laden 153
 LOAD (Befehl) 239

Diagnoseinformationen

 Probleme beim Versetzen von Daten 515

Dienstprogramm EXPORT

 Dateiformate 440
 Daten zwischen Hosts und Workstations übertragen 379
 Einschränkungen 8
 erforderliche Berechtigungen 7
 erforderliche Zugriffsrechte 7
 Identitätsspalten 17
 Leistung 6
 LOBs 17
 Optionen 6
 Tabelle erneut erstellen 13
 Übersicht 1, 6
 Vorbedingungen 8

Dienstprogramm IMPORT

 ALLOW NO ACCESS 68
 ALLOW WRITE ACCESS 68
 benutzerdefinierte einzigartige Datentypen (UDTs) 67
 Client/Server 68
 Codepages 500
 Dateiformate 440
 Daten zwischen Host und Workstation übertragen 379
 Einschränkungen 53
 erforderliche Berechtigungen 52
 erforderliche Zugriffsrechte 52
 ferne Datenbanken 68
 Generierte Spalten 65
 Gepufferte INSERT-Operationen 63
 Identitätsspalten 64
 LOBs 67
 Neuerstellung von exportierten Tabellen 56
 Tabellen sperren 68
 Übersicht 1, 49

- Dienstprogramm IMPORT (*Forts.*)
 - Vergleich mit Dienstprogramm LOAD 507
 - Vorbedingungen 53
- Dienstprogramm LOAD
 - Codepages 500
 - Dateiformate 440
 - Einschränkungen 147
 - Optionen beim Versetzen von Daten 1
 - Protokollsätze 215
 - Vergleich mit Dienstprogramm IMPORT 507
 - Vorbedingungen 147
- Dienstprogramme
 - Dateiformate 440
- Dienstprogramme für das Versetzen von Daten Handbuch und Referenz
 - Übersicht v
- Dokumentation
 - gedruckt 518
 - Nutzungsbedingungen 528
 - PDF-Dateien 518
 - Übersicht 517
- dumpfile (Änderungswert für Dateityp) 239

E

- Export
 - Daten
 - Änderungswerte für Dateityp 18
 - EXPORT (Befehl) 18
- EXPORT (Befehl)
 - Details
 - mit ADMIN_CMD 30
 - ohne ADMIN_CMD 18
- Export-API 41
- Exportieren
 - Daten
 - Änderungswerte für Dateityp 41
 - API 'db2Export' 41
 - Beispiele 48
 - LBAC-geschützte 12
 - Übersicht zum Dienstprogramm EXPORT 6
 - Vorgehensweise 8
 - XML 9
- Exportierte Tabellen
 - erneut erstellen 56

F

- fastparse (Änderungswert für Dateityp)
 - db2Load (API) 318
 - LOAD (Befehl) 239
- Fehlerbehebung
 - Diagnosedaten
 - Versetzen von Daten 515
 - Lernprogramme 527
 - Onlineinformationen 527
- Fehlerbestimmung
 - Lernprogramme 527
 - verfügbare Informationen 527
- Festlegen der Integrität anstehend, Status
 - SET INTEGRITY (Anweisung) 345
- forcein (Änderungswert für Dateityp)
 - API 'db2Import' 124
 - db2Load (API) 318
 - Details 490
 - IMPORT (Befehl) 70

- forcein (Änderungswert für Dateityp) (*Forts.*)
 - LOAD (Befehl) 239
- Fortsetzungssatztyp
 - PC/IXF 457

G

- generatedignore (Änderungswert für Dateityp)
 - API 'db2Import' 124
 - db2load (API) 318
 - IMPORT (Befehl) 70
 - LOAD (Befehl) 239
 - Spalten importieren 65
- generatedmissing (Änderungswert für Dateityp)
 - API 'db2Import' 124
 - db2load (API) 318
 - IMPORT (Befehl) 70
 - LOAD (Befehl) 239
 - Spalten importieren 65
- generatedoverride (Änderungswert für Dateityp)
 - db2load (API) 318
 - LOAD (Befehl) 239
- Generierte Spalten
 - Dienstprogramm IMPORT 65
 - LOAD, Dienstprogramm 158
- Gepufferte INSERT-Operationen
 - Dienstprogramm IMPORT 63
- Geteilte Spiegeldatenbanken
 - Handhabung 418
 - Übersicht 1
- Große Objekte (LOBs)
 - Export 504
 - Import 504

H

- Hierarchiesätze 457
- Hilfe
 - Konfiguration der Sprache 522
 - SQL-Anweisungen 522

I

- IBM Replikationstools für relationale Daten 381
- Identitätssätze 457
- Identitätsspalten
 - Daten exportieren 17
 - Dienstprogramm IMPORT 64
 - LOAD, Dienstprogramm 156
- identityignore (Änderungswert für Dateityp)
 - API 'db2Import' 124
 - db2Load (API) 318
 - IMPORT, Befehl 64
 - IMPORT (Befehl) 70
 - LOAD (Befehl) 239
- identitymissing (Änderungswert für Dateityp)
 - API 'db2Import' 124
 - db2Load (API) 318
 - IMPORT, Befehl 64
 - IMPORT (Befehl) 70
 - LOAD (Befehl) 239
- identityoverride (Änderungswert für Dateityp)
 - db2Load (API) 318
 - LOAD (Befehl) 239
- implieddecimal (Änderungswert für Dateityp)
 - API 'db2Import' 124

implieddecimal (Änderungswert für Dateityp) (Forts.)
db2Load (API) 318
IMPORT (Befehl) 70
LOAD (Befehl) 239

Import

Änderungswerte für Dateityp 124
Datei in Datenbanktabelle 124
Daten 53, 70
Einschränkungen 124
in ferne Datenbank 124
in nicht vorhandene Hierarchie 124
PC/IXF-Dateien
mehrteilig 124

IMPORT (Befehl)

Details
ohne ADMIN_CMD 70
unter Verwendung von ADMIN_CMD 97

Import-API 124

importieren

LBAC-Schutz 52

Importieren

Daten 61
Datenbankzugriff über DB2 Connect 124
Hinweise zu Codepages 124
in nicht vorhandene Tabelle 124
in typisierte Tabellen 124
PC/IXF-Dateien
allgemeine Regeln 485
datentypspezifische Regeln 486
Option FORCEIN 490
Übersicht 49
XML-Daten 55

indexfreespace (Änderungswert für Dateityp)

API 'db2Import' 318
LOAD (Befehl) 239

indexixf (Änderungswert für Dateityp)

API 'db2Import' 124
IMPORT (Befehl) 70

indexschema (Änderungswert für Dateityp)

API 'db2Import' 124
IMPORT (Befehl) 70

Indizes

erneut erstellen 176
erstellen
Leistung nach Ladeoperationen verbessern 176
Modi 176
PC/IXF-Satz 457
XML-Daten
Fehler beim Laden 180

Initialisieren einer Spiegeldatenbank, Befehl 419

Integration Exchange Format (IXF) 455

Integritätsbedingungen

überprüfen
nach Ladeoperationen 194
Verstöße 197

Integritätsprüfung 194

K

keepblanks (Änderungswert für Dateityp)

API 'db2Import' 124
db2Load (API) 318
IMPORT (Befehl) 70
LOAD (Befehl) 239

Kennsatzbasierte Zugriffssteuerung (LBAC)

Daten exportieren 7, 12
Daten importieren 52, 61

Kennsatzbasierte Zugriffssteuerung (LBAC) (Forts.)

Daten laden 145, 153

Komprimierung

Tabellen
Laden von Daten 187

Komprimierungswörterverzeichnis (Compression Dictionary)

KEEPDICTIONARY (Option) 187
RESETDICTIONARY (Option) 187

Kopfdatensatz

PC/IXF 457

L

Lade-API 318

Ladekopie, Datei mit Angaben zur Speicherposition 212

Laden

Änderungswerte für Dateityp 239
Beispiele
Übersicht 341
Umgebungen mit partitionierten Datenbanken 230
BUILD-Phase 176
Datei in Datenbanktabelle 239
Datenbankpartitionen 216, 224
erforderliche Angaben 141
Komprimierte Tabellen 187
Konfigurationsoptionen 233
LBAC-geschützte Daten 153
MDC-Tabellen (MDC = mehrdimensionales Clustering) 166
mit CURSOR 160
Partitionierte Tabellen 150
Tabellenzugriffsoptionen 202
Übersicht 141
Umgebungen mit partitionierten Datenbanken 233
XML-Daten 149
Zugriffsoptionen 202

Ladeoperationen

erneut starten
Mehrpartitionsdatenbanken 228
Modus "Lesezugriff zulassen" 211
Übersicht 210

Leistung

Dienstprogramm LOAD 189

Lernprogramme

Fehlerbehebung 527
Fehlerbestimmung 527
Liste 527
Visual Explain 527

LIST TABLESPACES (Befehl)

Details 370

LOAD, Befehl

Partitionierte Datenbanken, Umgebungen mit 230

LOAD (Befehl)

Details
ohne ADMIN_CMD 239
unter Verwendung von ADMIN_CMD 278
Umgebungen mit partitionierten Datenbanken 219

LOAD, Dienstprogramm

Änderungswerte für Dateityp 189
Ausnahmetabellen 208
Berechtigungen 145
BUILD-Phase 141
Daten mit SOURCEUSEREXIT versetzen 167
Datenbankrecovery 141
DELETE-Phase 141
Funktionen für referenzielle Integrität
Tabellenbereichsstatus 205

- LOAD, Dienstprogramm (Forts.)
 - Funktionen für referenzielle Integrität (Forts.)
 - Tabellenstatus 206
 - Übersicht 194
 - Generierte Spalten 158
 - Identitätsspalten 156
 - INDEX COPY-Phase 141
 - Indexerstellung 176
 - Ladeoperationen mit Status "Für Laden nicht neu startbar" 210
 - Leistungsoptimierung 189
 - LOAD-Phase 141
 - Parallelität 176
 - Recovery nach Fehlern 210
 - Speicherauszugsdatei 214
 - Tabellen sperren 201
 - Tabellenbereichsstatus 205
 - Tabellenstatus 206
 - temporäre Dateien
 - Übersicht 215
 - Übersicht 141
 - XML-Daten 180
 - Zugriffsrechte 145
 - zurückgewiesene Zeilen 214
- LOAD (Dienstprogramm)
 - Änderungswerte für Dateityp 318
 - temporäre Dateien
 - Speicherbedarf 239
- LOAD DELETE START COMPENSATION, Protokollsatz 215
- LOAD QUERY, Befehl
 - Details 364
 - Partitionierte Datenbanken, Umgebungen mit 226
- LOAD START, Protokollsatz
 - Übersicht 215
- LOB-Positionskenung 455
- LOBs
 - exportieren 17
 - importieren 67
- lobsinfile (Änderungswert für Dateityp)
 - API 'db2Export' 41
 - API 'db2Import' 124
 - db2Load (API) 318
 - EXPORT (Befehl) 18
 - exportieren 17
 - IMPORT (Befehl) 70
 - LOAD (Befehl) 239
- lobsinsefiles (Änderungswert für Dateityp) 17

M

- Materialized Query Table (MQT, Gespeicherte Abfragetabelle)
 - abhängige sofort gespeicherte 165
 - Daten aktualisieren 165
 - Status "Festlegen der Integrität anstehend" 165
- MDC-Tabellen (MDC = mehrdimensionales Clustering)
 - Laden 166

N

- Nachrichten
 - Dienstprogramm EXPORT 6
 - Dienstprogramm IMPORT 49
 - Dienstprogramm LOAD 141
- Nicht als Identitätsspalten generierte Spalten 65, 158
- Nicht wiederherstellbare Datenbanken
 - Ladeoptionen 141

- nochecklengths (Änderungswert für Dateityp)
 - API 'db2Import' 124
 - db2Load (API) 318
 - IMPORT (Befehl) 70
 - LOAD (Befehl) 239
- nodefaults (Änderungswert für Dateityp)
 - API 'db2Import' 124
 - IMPORT (Befehl) 70
- nodoubledel (Änderungswert für Dateityp)
 - API 'db2Export' 41
 - API 'db2Import' 124
 - db2Load (API) 318
 - EXPORT (Befehl) 18
 - IMPORT (Befehl) 70
 - LOAD (Befehl) 239
- noeofchar (Änderungswert für Dateityp)
 - API 'db2Import' 124
 - db2Load (API) 318
 - IMPORT (Befehl) 70
 - LOAD (Befehl) 239
- noheader (Änderungswert für Dateityp)
 - db2Load (API) 318
 - LOAD (Befehl) 239
- norowwarnings (Änderungswert für Dateityp)
 - db2Load (API) 318
 - LOAD (Befehl) 239
- notypeid (Änderungswert für Dateityp)
 - API 'db2Import' 124
 - IMPORT (Befehl) 70
- nullindchar (Änderungswert für Dateityp)
 - API 'db2Import' 124
 - db2Load (API) 318
 - IMPORT (Befehl) 70
 - LOAD (Befehl) 239

P

- packeddecimal (Änderungswert für Dateityp) 239
- pagefreespace (Änderungswert für Dateityp) 239
- Parallelität
 - Dienstprogramm LOAD 176
- Partitionierte Datenbanken, Umgebungen mit
 - Daten laden
 - Migration 230
 - Übersicht 224
 - Überwachung 226
 - Versionskompatibilität 230
 - Migration 230
 - Versionskompatibilität 230
- Partitionierte Tabellen
 - Laden 150
- PC/IXF
 - Codepagekonvertierungsdateien 485
 - Dateiimport
 - allgemeine Regeln 485
 - datentypspezifische Regeln 486
 - forcein (Änderungswert für Dateityp) 490
 - Daten zwischen Plattformen versetzen 442
 - Datentypen
 - gültige 473, 480
 - ungültige 473, 485
 - Inkompatible Spalten 485
 - Satztypen 457
 - System/370 IXF, Vergleich 489
 - Übersicht 455
 - ungültige Spaltenwerte 485

Protokollsätze
Dienstprogramm LOAD 215
Prozedur ADMIN_COPY_SCHEMA
Übersicht 1

R

reclen (Änderungswert für Dateityp)
API 'db2Import' 124
db2Load (API) 318
IMPORT (Befehl) 70
LOAD (Befehl) 239
Recovery
Datenbanken
RESTORE DATABASE (Befehl) 397
ohne aktualisierende Wiederherstellung 397
Registrierdatenbankvariablen
DB2LOADREC 212
REMOTEFETCH, Datenträgertyp 160
Replikation
Tools 381
RESTORE DATABASE (Befehl)
Details 397
Restore durchführen
ältere Versionen von DB2-Datenbanken 397
Restoredienstprogramm
GENERATE SCRIPT (Option) 1
REDIRECT (Option) 1
ROLLFORWARD, Dienstprogramm
Datei mit Angaben zur Speicherposition der Ladekopie 212

S

Sätze untergeordneter Tabellen
PC/IXF 457
Schemata
Kopieren 382
Tipps zur Fehlerbehebung 382
seclabelchar (Änderungswert für Dateityp)
Importieren von Daten 61
Laden von Daten 153
seclabelname (Änderungswert für Dateityp)
Importieren von Daten 61
Laden von Daten 153
SELECT (Anweisung)
EXPORT (Befehl) 18
SET CONSTRAINTS (Anweisung) 345
SET INTEGRITY, Anweisung
Prüfen auf ungültige Integritätsbedingungen 197
SET INTEGRITY (Anweisung)
Details 345
SOURCEUSEREXIT, Option 167
Spalten
Importieren 124
LBAC-geschützte
exportieren 7, 12
importieren 61
laden 153
Werte
ungültige 485
Spaltendeskriptorsatz 457
Speicher
XML-Datenkennung 505
Speicherauszugsdateien
Dienstprogramm LOAD 214

Sperren
Dienstprogramm IMPORT 68
Tabellenebene 201
SQL-Anweisungen
Hilfe
anzeigen 522
SET CONSTRAINTS 345
SET INTEGRITY 345
sqlexpr (API) 41
sqlimpr (API) 124
striptblanks (Änderungswert für Dateityp)
IMPORT (Befehl) 70
Importieren LBAC-geschützter Daten 61
LBAC-geschützte Daten laden 153
LOAD (Befehl) 239
striptnulls (Änderungswert für Dateityp)
IMPORT (Befehl) 70
LOAD (Befehl) 239
subtableconvert (Änderungswert für Dateityp) 239
Syntaxdiagramme
lesen 511
System/370 IXF
im Gegensatz zu PC/IXF 489
im Gegensatz zu System/370 IXF 489

T

Tabellen
Ausnahme 345
Dateien importieren 70, 124
exportierte Tabellen erneut erstellen 56
in Dateien exportieren 18, 41
laden 239
online versetzen
ADMIN_MOVE_TABLE, Prozedur 375
sperren 201
Tabellenbereiche
Statusangaben 205
Tabellenbereichsstatus
Ladeoperationen 205
Tabellensatz
PC/IXF 457
Tabellenstatus
Ladeoperationen 206
TABLE LOAD DELETE START, Protokollsatz 215
temporäre Dateien
LOAD (Dienstprogramm)
Speicherbedarf 239
Temporäre Dateien
Dienstprogramm LOAD
Übersicht 215
timeformat (Änderungswert für Dateityp)
API 'db2Import' 124
db2Load (API) 318
IMPORT (Befehl) 70
LOAD (Befehl) 239
timestampformat (Änderungswert für Dateityp)
API 'db2import' 124
db2load (API) 318
IMPORT (Befehl) 70
LOAD (Befehl) 239
Tool zum Versetzen von Daten, Befehl 386
totalreespace (Änderungswert für Dateityp)
db2Load (API) 318
LOAD (Befehl) 239
Typisierte Tabellen
Daten versetzen zwischen 14, 58

Typisierte Tabellen (*Forts.*)
erneut erstellen 58
exportieren 14
importieren 58
Traversierfolge 14, 58

U

Übersichtstabellen
Importbeschränkung 53
Umgebungen mit partitionierten Datenbanken
Daten laden
Einschränkungen 219
Übersicht 216
Umgeleitete Restores
mit generiertem Script 396
Unicode-Codierung UCS-2
Versetzen von Daten 498
usedefaults (Änderungswert für Dateityp)
API 'db2Import' 124
db2Load (API) 318
IMPORT (Befehl) 70
Importe LBAC-geschützter Daten 61
LBAC-geschützte Daten laden 153
LOAD (Befehl) 239

V

Verlagern einer Datenbank (Befehl) 422
Versetzen von Daten
Begrenzer, Einschränkungen 449
DB2 Connect 379
Dienstprogramm EXPORT 6
Dienstprogramm IMPORT 49
Dienstprogramm LOAD 141
XML 502
zwischen Datenbanken 70, 124
Verteilungsschlüssel
Daten laden 216

W

Wiederherstellbare Datenbanken
Ladeoptionen 141
Worksheet Format (WSF)
Daten zwischen Plattformen versetzen 442
Details 497

X

XML-Daten
Abfrage- und XPath-Datenmodell 506
exportieren 9
importieren 55
laden 149
versetzen 501, 502
XML-Datentyp
Export 504
Import 504
XQuery-Anweisungen
XQuery- und XPath-Datenmodell 506

Z

Zeichenfolgen
Begrenzer 448
Zeilen
Daten in LBAC-geschützte Zeilen laden 153
in LBAC-geschützte importieren 61
LBAC-geschützte Daten exportieren 7, 12
zoneddecimal (Änderungswert für Dateityp)
db2Load (API) 318
LOAD (Befehl) 239
Zugriffsrechte
Dienstprogramm EXPORT 7
Dienstprogramm IMPORT 52
Dienstprogramm LOAD 145
Zusatzspeicherobjekte
XML-Datenkennung 505
Zwischenspeichertabellen
abhängige sofort gespeicherte 164
weitergeben 164



SC12-4281-01



Spine information:

DB2 for Linux, UNIX and Windows **Version 9 Release 7**

Dienstprogramme für das Versetzen von Daten - Handbuch und Referenz

