

IBM DB2 Universal Database



Systemverwaltung: Implementierung

Version 7

IBM DB2 Universal Database



Systemverwaltung: Implementierung

Version 7

Anmerkung:

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen unter „Anhang M. Bemerkungen“ auf Seite 545 gelesen werden.

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

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM DB2 Universal Database Administration Guide: Implementation,
IBM Form SC09-2944-01,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 1993, 2001
© Copyright IBM Deutschland Informationssysteme GmbH 2001

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

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

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
SW TSC Germany
Kst. 2877
August 2001

Inhaltsverzeichnis

Zu diesem Handbuch	xi
Zielgruppe	xii
Aufbau dieses Handbuchs	xii
Kurzübersicht über die anderen Bände des Handbuchs zur Systemverwaltung	xiv
Systemverwaltung: Konzept	xiv
Systemverwaltung: Optimierung	xv

Teil 1. Systemverwaltung mit der Steuerzentrale **1**

Kapitel 1. Verwalten von DB2 mit den Tools für die grafische Benutzerschnittstelle. **3**

Verwaltungs-Tools	4
Gemeinsame Funktionen der Tools	7
"SQL anzeigen" und "Befehl anzeigen"	7
"Zugehörige anzeigen"	8
"DDL generieren"	9
"Filtern"	10
Hilfe	12
Steuerzentrale	12
Hauptelemente der Steuerzentrale	13
Verwenden einer angepassten Steuerzentrale in DB2 für OS/390	14
Systeme, die verwaltet werden können	15
Objekte, die verwaltet werden können	15
Anzeigen von Systemen in der Steuerzentrale	16
Verwalten von Objekten in DB2 für OS/390	17
Hinzufügen von DB2 für OS/390-Subsystemen	17
Verwalten von Gateway-Verbindungen	18
Funktionen in der Steuerzentrale	19
Erstellen neuer Objekte	20
Arbeiten mit vorhandenen Objekten	20
Suchen von Objekten (nur DB2 für OS/390)	20
Satellitenverwaltungszentrale	21
Befehlszentrale	22
Prozedurzentrale	23
Verwenden einer vorhandenen Prozedur in der Prozedurzentrale	24

Zeitliches Steuern der Ausführung einer gespeicherten Befehlsprozedur	24
Journal	25
Arbeiten mit Jobs	25
Lizenzzentrale	26
Alert-Zentrale	26
"Client-Konfiguration - Unterstützung"	27
Performance Monitor	28
Ereignismonitor	29
Verwenden der Monitor-Tools	30
Überwachen der Leistung zu einem bestimmten Zeitpunkt	32
Vordefinierte Monitore	33
Erforderliche Aktion nach der Anzeige eines Objekts in der Alert-Zentrale	35
Analysieren eines Ereignisses über einen bestimmten Zeitraum	36
Event Analyzer	36
Analysieren von SQL-Anweisungen	39
Optimieren einer Abfrage	40
Analysieren einer einfachen dynamischen SQL-Anweisung	41
Verwalten ferner Datenbanken	42
Verwalten von Benutzern	44
Erteilen und Entziehen von Berechtigungen und Zugriffsrechten	44
Versetzen von Daten	45
Verwalten von Speicher	48
Schätzen der Tabellen- und Indexgröße	48
Überprüfen des verfügbaren Speichers in einem Tabellenbereich	49
Hinzufügen von mehr Speicher für einen Tabellenbereich	50
Fehlerbehebung	51
Replizieren von Daten	52
Verwenden des Protokolls LDAP (Lightweight Directory Access Protocol)	53
Verwenden der Java-Steuerzentrale	53
Ausführen der Steuerzentrale als Java-Applet	54
Verwenden der Java-basierten Tools zur Verwaltung	54

Teil 2. Implementieren des Datenbankentwurfs 55

Kapitel 2. Vor dem Erstellen einer Datenbank 57

Vorbedingungen für das Erstellen einer Datenbank	58
Starten von DB2	58
Starten von DB2 UDB auf Windows NT.	59
Verwenden mehrerer Exemplare des Datenbankmanagers	60
Organisieren und Gruppieren von Objekten nach Schema	61
Aktivieren der Parallelität	62
Ermöglichen der Datenpartitionierung	64
Stoppen von DB2	67
Details beim Erstellen einer Datenbank	68
Entwerfen logischer und physischer Datenbankmerkmale	68
Erstellen eines Exemplars	68
Lizenzverwaltung	79
Einrichten der Umgebungsvariablen und der Profilregistrierdatenbank	79
Erstellen eines DB2-Verwaltungsservers (DAS)	89
Erstellen einer Knotenkonfigurationsdatei	108
Erstellen der Datenbankkonfigurationsdatei	111
Replizieren von Konfigurationsdaten durch Antwortdateien	112
Aktivieren der FCM-Kommunikation	112

Kapitel 3. Erstellen einer Datenbank 115

Definition der ersten Knotengruppen	117
Definition der ersten Tabellenbereiche	117
Definition von Systemkatalogtabellen	119
Definition von Datenbankverzeichnissen	120
Lokales Datenbankverzeichnis	120
Systemdatenbankverzeichnis	120
Knotenverzeichnis	121
DCE Verzeichnisservices	122
LDAP-Verzeichnisservices (Lightweight Directory Access Protocol)	122
Erstellen von Knotengruppen	123
Definition des Wiederherstellungsprotokolls der Datenbank.	124
Binden von Dienstprogrammen an die Datenbank	124
Katalogisieren einer Datenbank	125

Erstellen eines Tabellenbereichs	126
Erstellen eines temporären Systemtabellenbereichs	129
Erstellen eines temporären Benutzertabellenbereichs	130
Erstellen von Tabellenbereichen in Knotengruppen	130
Direkter Plattenzugriff (Raw I/O)	130
Erstellen eines Schemas	133
Einstellen eines Schemas	134
Erstellen und Auffüllen einer Tabelle	135
Überlegungen zu LOB-Spalten	138
Definieren von Integritätsbedingungen	140
Definieren einer generierten Spalte in einer neuen Tabelle	146
Erstellen einer benutzerdefinierten temporären Tabelle	147
Definieren einer Identitätsspalte in einer neuen Tabelle	148
Erstellen einer Sequenz	150
Vergleich von Identitätsspalten (IDENTITY) und Sequenzen	152
Erstellen einer typisierten Tabelle	153
Auffüllen einer typisierten Tabelle	153
Hierarchietabelle	153
Erstellen einer Tabelle in mehreren Tabellenbereichen	153
Erstellen einer Tabelle in einer partitionierten Datenbank	154
Erstellen eines Auslösers	156
Auslöserabhängigkeiten.	159
Erstellen einer benutzerdefinierten Funktion (UDF) oder Methode.	159
Erstellen einer Funktionszuordnung.	161
Erstellen einer Funktionsschablone	162
Erstellen eines benutzerdefinierten Datentyps (UDT).	163
Erstellen eines benutzerdefinierten einzigartigen Datentyps.	164
Erstellen eines benutzerdefinierten strukturierten Typs	165
Erstellen einer Typenzuordnung	165
Erstellen einer Sicht	166
Erstellen einer typisierten Sicht	169
Erstellen einer Übersichtstabelle	169
Erstellen eines Aliasnamens	172
Erstellen einer Oberfläche	175
Erstellen eines Servers	176

Vereinfachen der Datenquellendefinitionen und der Authentifizierungsverarbeitung durch Serveroptionen.	178
Erstellen eines Kurznamens	185
Verweisen auf Kurznamen und Datenquellenobjekte	186
Arbeiten mit Kurznamen und Datenquellenobjekten	186
Identifizieren vorhandener Kurznamen und Datenquellen.	187
Erstellen eines Index, einer Indexerweiterung oder einer Indexspezifikation	188
Verwenden eines Index	193
Verwenden der Anweisung CREATE INDEX	193
Erstellen einer benutzerdefinierten erweiterter Indexart.	196
Details zur Indexverwaltung	197
Details zur Indexsuche	197
Details zur Indexausnutzung	199
Szenario zum Definieren einer Indexerweiterung	200
Kapitel 4. Ändern einer Datenbank	203
Vor dem Ändern der Datenbank	203
Ändern der logischen und physischen Datenbankmerkmale	203
Ändern der Lizenzinformationen	203
Ändern der Exemplare	203
Ändern der Umgebungsvariablen und der Variablen der Profilregistrierdatenbank.	207
Ändern der Knotenkonfigurationsdatei	208
Ändern der Datenbankkonfiguration	208
Ändern einer Datenbank	210
Löschen einer Datenbank	210
Ändern einer Knotengruppe	211
Ändern eines Tabellenbereichs	211
Löschen eines Schemas	218
Ändern von Struktur und Inhalt einer Tabelle	218
Ändern eines benutzerdefinierten strukturierten Typs.	238
Löschen und Aktualisieren von Zeilen einer typisierten Tabelle.	239
Umbenennen einer vorhandenen Tabelle	239
Löschen einer Tabelle	240
Löschen einer benutzerdefinierten temporären Tabelle	242
Löschen eines Auslösers	242

Löschen einer benutzerdefinierten Funktion (UDF), Typenzuordnung oder Methode	243
Löschen eines benutzerdefinierten Datentyps (UDT) oder einer Typenzuordnung	244
Ändern oder Löschen einer Sicht.	244
Löschen einer Übersichtstabelle	246
Löschen einer Oberfläche	248
Ändern oder Löschen eines Servers	248
Ändern oder Löschen eines Kurznamens	249
Löschen eines Index, einer Indexerweiterung oder einer Indexspezifikation	251
Anweisungsabhängigkeiten beim Ändern von Objekten	253

Teil 3. Datenbanksicherheit 255

Kapitel 5. Steuern des Datenbankzugriffs 257

Auswählen von Benutzer-IDs und Gruppen für Ihre Installation	258
Überlegungen zur Windows NT-Plattform	260
Überlegungen zur UNIX-Plattform	260
Allgemeine Regeln	261
Auswählen einer Authentifizierungsart für den Server	261
Überlegungen zur Authentifizierung für ferne Clients	270
Überlegungen zu partitionierten Datenbanken	270
Verwenden von DCE-Sicherheitsservices zur Überprüfung von Benutzern	270
Einrichten eines DB2-Benutzers für DCE	271
Einrichten eines DB2-Servers zur Verwendung von DCE.	274
Einrichten eines DB2-Client-Exemplars zur Verwendung von DCE.	276
DB2-Einschränkungen bei der Verwendung der DCE-Sicherheit	276
Authentifizierungsverarbeitung in zusammengeschlossenen Datenbanken	278
Authentifizierungseinstellungen	278
Übergeben von Benutzer-IDs und Kennwörtern an Datenquellen	280
Beispiel für eine Authentifizierung für ein System zusammengeschlossener Datenbanken	283
Zugriffsrechte und Berechtigungen	286
Systemverwaltungsberechtigung (SYS-ADM).	288

Systemsteuerungsberechtigung (SYSC-TRL)	289
Systempflegeberechtigung (SYSMAINT)	290
Datenbankadministratorberechtigung (DBADM)	291
LOAD-Berechtigung	292
Zugriffsrechte für Datenbanken	293
Zugriffsrechte für Schemata	295
Zugriffsrechte für Tabellenbereiche	297
Zugriffsrechte für Tabellen und Sichten	297
Zugriffsrechte für Kurznamen.	300
Zugriffsrechte auf Server	302
Zugriffsrechte für Pakete	302
Zugriffsrechte für Indizes	303
Zugriffsrechte für Sequenzen	303
Steuern des Zugriffs auf Datenbankobjekte	304
Erteilen von Zugriffsrechten	304
Widerrufen von Zugriffsrechten	305
Implizite Berechtigungen durch Erstellen und Löschen von Objekten.	308
Einrichten eines Eigentumsrechts für einen Plan oder ein Paket	308
Ermöglichen indirekter Zugriffsberechtigungen durch ein Paket	309
Ermöglichen indirekter Zugriffsberechtigungen durch ein Paket mit Kurznamen	309
Steuern des Zugriffs auf Daten mit Sichten	311
Überwachen des Zugriffs auf Daten mit Hilfe der Audit-Einrichtung	314
Datenverschlüsselung	314
Aufgaben und erforderliche Berechtigungen	316
Verwenden des Systemkatalogs	317
Abrufen von Berechtigungsnamen mit erteilten Zugriffsrechten.	318
Abrufen aller Namen mit der Berechtigung DBADM	319
Abrufen der Namen mit Berechtigung zum Zugriff auf eine Tabelle	319
Abrufen aller für Benutzer erteilten Zugriffsrechte	320
Sicherheit der Systemkatalogsichten.	320
Kapitel 6. Protokollieren von DB2-Aktivitäten	323
Arbeitsweise der Prüffunktion	325
Anwendungsszenarios der Prüffunktion	327
Nachrichten der Prüffunktion.	332
Satzaufbau der Prüffunktion	332

Tipps und Verfahrensweisen für die Prüffunktion	352
Steuern der Aktivitäten der DB2-Prüffunktion	353

Teil 4. Versetzen von Daten 359

Kapitel 7. Dienstprogramme für das Versetzen von Daten 361

Teil 5. Wiederherstellung 363

Kapitel 8. Wiederherstellen einer Datenbank 365

Teil 6. Anhänge und Schlussteil 367

Anhang A. Namenskonventionen 369

Allgemeine Namenskonventionen	369
Namenskonventionen für Objektnamen	370
Zusätzliche Informationen zu Schemanamen	371
Zusätzliche Informationen zu Kennwörtern	373
Verwendung begrenzter Bezeichner in Objektnamen	374
Beibehaltung von Werten in Groß-/Kleinschreibung in einem System zusammengesetzter Datenbanken	374

Anhang B. Verwenden der DCE-Verzeichnisservices 377

Erstellen von Verzeichnisobjekten	377
Datenbankobjekte.	378
Querverweisobjekte für Datenbanken	380
Objekte für Leitweginformationen	381
Attribute der einzelnen Objektklassen	384
Einzelangaben zu den Attributen	385
Sicherheit bei den Verzeichnisservices	390
Konfigurationsparameter und Registrierungswerte	392
Befehle CATALOG und ATTACH und die Anweisung CONNECT	394
Der Befehl CATALOG GLOBAL DATABASE	394
Anweisung CONNECT	395
Der Befehl ATTACH	395

Herstellen der Verbindung eines Clients mit einer Datenbank	395	Ein Beispielszenario mit Authentifizierung auf dem Client und einer Windows NT-Client-Maschine:	431
Herstellen einer Verbindung zu Datenbanken in derselben Zelle	397	Ein Beispielszenario mit Authentifizierung auf dem Client und einer Windows 95-Client-Maschine:	432
Herstellen einer Verbindung zu Datenbanken in einer anderen Zelle.	398	Verwenden einer Sicherungs-Domänensteuereinheit mit DB2	433
Durchsuchen von Verzeichnissen.	400	Benutzerauthentifizierung mit DB2 für Windows NT.	434
Der Befehl ATTACH	400	Einschränkungen bei Benutzername und Gruppenname	434
Anweisung CONNECT	401	Sicherheitsservice von DB2 für Windows NT.	434
Temporäres Außerkraftsetzen von Informationen in DCE-Verzeichnissen	402	Installieren von DB2 auf einem Sicherungsdomänen-Controller	435
Aufgaben im Zusammenhang mit den Verzeichnisservices	402	Authentifizierung mit Gruppen und Domänensicherheit	436
Aufgaben des DCE-Administrators	403		
Aufgaben des Datenbankadministrators	403		
Aufgaben des Datenbankbenutzers	405		
Einschränkungen bei den Verzeichnisservices	406		
Anhang C. Benutzer-Exit zur Datenbankwiederherstellung	409	Anhang F. Arbeiten mit dem Windows NT-Systemmonitor	441
Anhang D. Absetzen von Befehlen an mehrere Datenbankpartitionsserver . . .	411	Registrieren von DB2 beim Windows NT-Systemmonitor	441
Befehle	411	Aktivieren des Fernzugriffs auf DB2-Leistungsinformationen.	442
Befehlsbeschreibungen	412	Anzeigen von Leistungswerten für DB2 und DB2 Connect	443
Angaben des auszuführenden Befehls	413	Zugreifen auf ferne DB2-Leistungsinformationen	444
Paralleles Ausführen von Befehlen auf Plattformen unter UNIX	414	Zurücksetzen von DB2-Leistungswerten	444
Überwachen des rah-Prozesses auf Plattformen unter UNIX	415		
Zusätzliche rah-Informationen (Run All Hosts) (nur für Solaris und AIX).	416	Anhang G. Arbeiten mit Datenbankpartitionsservern unter Windows NT oder Windows 2000	447
Präfixsequenzen	417	Auflisten der Datenbankpartitionsserver eines Exemplars	447
Angaben der Liste von Maschinen	420	Hinzufügen eines Datenbankpartitionsservers zu einem Exemplar	448
Eliminieren mehrfacher Einträge aus der Liste der Maschinen	421	Ändern der Datenbankpartition	450
Steuern des Befehls rah	422	Löschen einer Datenbankpartition aus einem Exemplar	451
\$RAHDOTFILES auf Plattformen unter UNIX	424		
Definieren des Standardumgebungsprofils unter Windows NT	425	Anhang H. Konfigurieren mehrerer logischer Knoten	453
Bestimmung von Fehlern mit rah auf Plattformen unter UNIX	425		
Anhang E. Arbeitsweise von DB2 für Windows NT mit der Windows NT-Sicherheit .	429	Anhang I. Knotenübergreifende Hochgeschwindigkeitskommunikation.	455
Ein Beispielszenario mit Authentifizierung auf dem Server:	430	Hochgeschwindigkeitsverbindungen unter TCP/IP	456

Voraussetzungen für den Einsatz eines IBM Netfinity SP Switch	457
Hochgeschwindigkeitsverbindungen mit VI	457
Hardwarekonfiguration mit Virtual Interface (VI)	458
Aktivieren von DB2 zur Ausführung mit VI	467

Anhang J. LDAP-Verzeichnisservices (Lightweight Directory Access Protocol) . 469

Unterstützen von LDAP-Client- und -Serverkonfigurationen	470
Unterstützung für Windows 2000 Active Directory.	471
Konfigurieren von DB2 zur Verwendung von Windows 2000 Active Directory	471
Konfigurieren von DB2 in einer IBM LDAP-Umgebung	472
Erstellen eines LDAP-Benutzers	473
Konfigurieren des LDAP-Benutzers für DB2-Anwendungen	474
Registrierung von DB2-Servern nach der Installation	474
Aktualisieren der Protokollinformationen für den DB2-Server	476
Katalogisieren eines Knoten-Aliasnamens für ATTACH.	477
Zurücknehmen der Registrierung des DB2-Servers	477
Registrieren von Datenbanken	478
Herstellen einer Verbindung zu einem fernen Server.	478
Zurücknehmen der Registrierung der Datenbank	479
Aktualisieren der LDAP-Einträge in lokalen Datenbank- und Knotenverzeichnissen.	480
Suchen	481
Registrieren der Host-Datenbanken	482
Einstellen der DB2-Variablen der Profilregistrierdatenbank auf Benutzerebene.	484
Aktivieren der LDAP-Unterstützung nach abgeschlossener Installation	485
Inaktivieren von LDAP-Unterstützung	486
LDAP-Unterstützung und DB2 Connect	486
Überlegungen zur Sicherheit	486
Überlegungen zur Sicherheit bei Windows 2000 Active Directory	487
Erweitern des Verzeichnisschemas mit DB2-Objektklassen und -Attributen	489

Erweitern des Verzeichnisschemas für IBM eNetwork Directory Version 2.1	489
Erweitern des Verzeichnisschemas für Windows 2000 Active Directory	490
DB2-Objekte im Windows 2000 Active Directory.	491
Von DB2 verwendete Objektklassen und Attribute.	492

Anhang K. Erweitern der Steuerzentrale 505

Überlegungen zur Leistung	505
Überlegungen zu Paketen	505
Schnittstellenbeschreibungen	505
CCExtension	506
CCObject	507
CCMenuItem	510
CCToolBarAction	511
Einsatz-Szenario	511
MyExtension.java	512
MySample.java.	512
MyDatabaseActions.java	513
MyInstance.java	514
MyDB2.java.	514
MyDatabases.java.	515
MySYSPLAN.java.	516
MyTable.java	516
MyDBUser.java	517
MyToolBarAction.java	518
MyAlterAction.java	518
MyAction.java	518
MyDropAction.java	519
MyCascadeAction.java	519
MyCreateAction.java.	520

Anhang L. Verwenden der DB2-Bibliothek 521

PDF-Dateien und gedruckte Bücher für DB2	521
Informationen zu DB2	521
Drucken der PDF-Handbücher	534
Bestellen der gedruckten Handbücher	535
DB2-Online-Dokumentation	537
Zugreifen auf die Online-Hilfefunktion	537
Anzeigen von Online-Informationen	539
Verwenden der DB2-Assistenten	541
Einrichten eines Dokument-Servers	543
Suchen nach Online-Informationen	544

Anhang M. Bemerkungen 545

Marken	548
------------------	-----

Index 551

Kontaktaufnahme mit IBM	563	Produktinformationen	563
--	------------	--------------------------------	-----

Zu diesem Handbuch

Das Handbuch zur Systemverwaltung bietet in seinen drei Bänden Informationen, die zur Verwendung und Verwaltung der Jahr 2000-konformen DB2-Produkte für Verwaltungssysteme für relationale Datenbanken (RDBMS) benötigt werden:

- Informationen zum Datenbankentwurf (im Band *Systemverwaltung: Konzept*)
- Informationen zur Implementierung und Verwaltung von Datenbanken (im Band *Systemverwaltung: Implementierung*)
- Informationen zur Konfiguration und Optimierung der Datenbankumgebung zum Zweck der Leistungsverbesserung (im Band *Systemverwaltung: Optimierung*).

Viele der in diesem Handbuch beschriebenen Operationen können mit Hilfe verschiedener Schnittstellen durchgeführt werden:

- Der **Befehlszeilenprozessor**, der Ihnen den Zugriff auf Datenbanken und deren Bearbeitung über eine grafische Schnittstelle ermöglicht. Von dieser Schnittstelle aus können Sie SQL-Anweisungen und DB2-Dienstprogrammfunktionen ausführen. Die Mehrzahl der Beispiele in diesem Handbuch zeigt die Verwendung dieser Schnittstelle. Weitere Informationen zur Verwendung des Befehlszeilenprozessors finden Sie im Handbuch *Command Reference*.
- Die **Anwendungsprogrammierschnittstelle**, die Ihnen die Ausführung von DB2-Dienstprogrammfunktionen innerhalb eines Anwendungsprogramms ermöglicht. Weitere Informationen zur Verwendung der Anwendungsprogrammierschnittstelle finden Sie im Handbuch *Administrative API Reference*.
- Die **Steuerzentrale**, die Ihnen die Ausführung administrativer Aufgaben wie das Konfigurieren des Systems, die Verwaltung von Verzeichnissen, das Sichern und Wiederherstellen des Systems, die zeitliche Festlegung von Jobs und die Verwaltung von Medien über eine grafische Schnittstelle ermöglicht. Die Steuerzentrale enthält außerdem eine Replikationsverwaltung, mit der die Replikation von Daten zwischen den Systemen mit Hilfe einer grafischen Schnittstelle eingerichtet werden kann. Darüber hinaus ermöglicht die Steuerzentrale das Ausführen von DB2-Dienstprogrammfunktionen über eine grafische Benutzerschnittstelle. Zum Aufrufen der Steuerzentrale gibt es abhängig von der Plattform verschiedene Methoden. Zum Beispiel verwenden Sie den Befehl `db2cc` in einer Befehlszeile, wählen (unter OS/2) das Symbol der Steuerzentrale im DB2-Ordner aus oder arbeiten mit dem Startmenü auf Windows-Plattformen. Wenn Sie eine einführende Hilfe

benötigen, wählen Sie **Erste Schritte** im Menü **Hilfe** des Fensters der Steuerzentrale aus. **Visual Explain** und **Performance Monitor** werden von der Steuerzentrale aus aufgerufen.

Es gibt darüber hinaus noch weitere Tools, die Sie zur Durchführung von Verwaltungsaufgaben verwenden können. Dazu gehören:

- Die Prozedurzentrale, die zum Speichern kleiner Anwendungen dient, die als Prozeduren (Scripts) bezeichnet werden. Diese Prozeduren können SQL-Anweisungen, DB2-Befehle und auch Betriebssystembefehle enthalten.
- Die Alert-Zentrale, die zur Überwachung von Nachrichten dient, die sich aus anderen DB2-Operationen ergeben.
- Das Programm Tools - Einstellungen, mit dem Sie die Einstellungen für die Steuerzentrale, die Alert-Zentrale und die Replikation ändern können.
- Das Journal, das zur zeitlichen Terminierung von Jobs dient, die im nicht-überwachten Modus ausgeführt werden sollen.
- Die Data Warehouse-Zentrale, die zur Verwaltung von Warehouse-Objekten dient.

Zielgruppe

Dieses Handbuch ist hauptsächlich für Datenbankadministratoren, Systemadministratoren, Sicherheitsadministratoren und Systembediener gedacht, die eine Datenbank für den lokalen oder fernen Zugriff entwerfen, implementieren und pflegen müssen. Es wendet sich auch an Programmierer und andere Benutzer, die Kenntnisse über die Verwaltung und Bedienung des relationalen Datenbankverwaltungssystems von DB2 benötigen.

Aufbau dieses Handbuchs

Das vorliegende Handbuch enthält Informationen zu folgenden Hauptthemen:

Systemverwaltung mit der Steuerzentrale

- Kapitel 1. Verwalten von DB2 mit den Tools für die grafische Benutzerschnittstelle, behandelt die Tools der grafischen Benutzerschnittstelle (GUI), die zur Verwaltung der Datenbank verwendet werden.

Implementieren des Datenbankentwurfs

- Kapitel 2. Vor dem Erstellen einer Datenbank, behandelt die Voraussetzungen, die vor der Erstellung einer Datenbank und der Objekte in einer Datenbank erfüllt sein müssen.
- Kapitel 3. Erstellen einer Datenbank, behandelt die Aufgaben der Erstellung einer Datenbank sowie der Objekte innerhalb einer Datenbank.

- Kapitel 4. Ändern einer Datenbank, behandelt die Voraussetzungen und Aufgaben im Zusammenhang mit dem Ändern und Löschen einer Datenbank und der Objekte innerhalb einer Datenbank.

Datenbanksicherheit

- Kapitel 5. Steuern des Datenbankzugriffs, beschreibt die Steuerung des Zugriffs auf die Datenbankressourcen.
- Kapitel 6. Protokollieren von DB2-Aktivitäten, beschreibt Methoden zur Erkennung und Überwachung unerwünschten bzw. unvorhergesehenen Zugriffs auf Daten.

Versetzen von Daten

- Kapitel 7. Dienstprogramme für das Versetzen von Daten, behandelt die Dienstprogramme LOAD, Autoloader, IMPORT und EXPORT.

Wiederherstellung

- Kapitel 8. Wiederherstellen einer Datenbank, beschreibt die Faktoren, die bei der Auswahl der Methoden zur Wiederherstellung von Datenbanken und Tabellenbereichen zu beachten sind. Wiederherstellungsfunktionen umfassen die Sicherung und Wiederherstellung einer Datenbank oder eines Tabellenbereichs sowie die aktualisierende Wiederherstellung.

Anhänge

- Anhang A. Namenskonventionen, enthält die Regeln, die bei der Benennung von Datenbanken und Objekten zu beachten sind.
- Anhang B. Verwenden der DCE-Verzeichnisservices, enthält Informationen zur Verwendung der DCE Verzeichnisservices.
- Anhang C. Benutzer-Exit zur Datenbankwiederherstellung, behandelt die Verwendung von Benutzerausgangsprogrammen mit Datenbankprotokoll-dateien und beschreibt einige Beispiele von Benutzerausgangsprogrammen.
- Anhang D. Absetzen von Befehlen an mehrere Datenbankpartitionsserver, behandelt die Verwendung der Shell-Prozeduren *db2_all* und *rah* zum Senden von Befehlen an alle Partitionen in einer partitionierten Datenbank-umgebung.
- Anhang E. Arbeitsweise von DB2 für Windows NT mit der Windows NT-Sicherheit, beschreibt, wie DB2 mit Windows NT-Sicherheit arbeitet.
- Anhang F. Arbeiten mit dem Windows NT-Systemmonitor, beschreibt die beiden in DB2 verfügbaren Leistungsmonitoren für Windows-NT-Benutzer: DB2 Performance Monitor und den Windows NT-Systemmonitor.
- Anhang G. Arbeiten mit Datenbankpartitionsservern unter Windows NT oder Windows 2000, beschreibt die Dienstprogramme, die von Windows NT und Windows 2000 zur Arbeit mit Servern partitionierter Datenbanken verwendet werden.

- Anhang H. Konfigurieren mehrerer logischer Knoten, beschreibt, wie mehrere logische Knoten in einer partitionierten Datenbankumgebung konfiguriert werden können.
- Anhang I. Knotenübergreifende Hochgeschwindigkeitskommunikation, beschreibt, wie VIA (Virtual Interface Architecture) zur Verwendung mit DB2 Enterprise - Extended Edition in der Windows NT-Umgebung eingerichtet werden kann.
- Anhang J. LDAP-Verzeichnisservices (Lightweight Directory Access Protocol), enthält Informationen zur Verwendung der LDAP-Verzeichnisservices.
- Anhang K. Erweitern der Steuerzentrale, enthält Informationen zur Erweiterung der Steuerzentrale durch Hinzufügen neuer Knöpfe in der Menüleiste, einschließlich neuer Aktionen, Hinzufügen neuer Objektdefinitionen und Hinzufügen neuer Aktionsdefinitionen.
- Anhang L. Verwenden der DB2-Bibliothek, enthält Informationen zu der Struktur der DB2-Bibliothek, einschließlich Assistenten, Online-Hilfefunktion, Nachrichten und Handbücher.

Kurzübersicht über die anderen Bände des Handbuchs zur Systemverwaltung

Systemverwaltung: Konzept

Der Band *Systemverwaltung: Konzept* behandelt den Datenbankentwurf. Er enthält Themen zum logischen und physischen Entwurf, zu verteilten Transaktionen und zur hohen Verfügbarkeit. Die einzelnen Kapitel und Anhänge des Bandes werden im Folgenden kurz vorgestellt:

DB2 Universal Database

- Das Kapitel über die Verwaltung von DB2 Universal Database enthält eine Einführung sowie eine Übersicht zu DB2 Universal Database.

Datenbankkonzepte

- Das Kapitel zu allgemeinen Konzepten relationaler Datenbanken enthält eine Übersicht über Datenbankobjekte, einschließlich Wiederherstellungsobjekte, Speicherobjekte und Systemobjekte.
- Das Kapitel über Systeme zusammenschlossener Datenbanken behandelt Datenbankverwaltungssysteme (DBMS), die Anwendungen und Benutzer unterstützen, die SQL-Anweisungen übergeben, die auf zwei oder mehr DBMSs oder Datenbanken in einer einzigen Anweisung verweisen.
- Das Kapitel über parallele Datenbanksysteme enthält eine Einführung in die Arten von Parallelität, die mit Hilfe von DB2 implementiert werden können.
- Das Kapitel über Data Warehouses enthält eine Übersicht über den Einsatz von Data Warehouses und Data Warehouse-Funktionen.

- Das Kapitel über Spatial Extender stellt Spatial Extender vor und enthält Erläuterungen zum Zweck des Produkts sowie zu den Daten, die von ihm verarbeitet werden.

Datenbankentwurf

- Das Kapitel zum Entwerfen des logischen Datenbankaufbaus behandelt die Konzepte und Richtlinien zum logischen Entwurf einer Datenbank.
- Das Kapitel zum Entwerfen der physischen Datenbank behandelt die Richtlinien zum physischen Entwurf einer Datenbank und enthält Überlegungen im Hinblick auf die Datenspeicherung.

Verteilte Transaktionsverarbeitung

- Das Kapitel zum Entwerfen verteilter Datenbanken beschreibt die Möglichkeit des Zugriffs auf mehrere Datenbanken in einer einzigen Transaktion.
- Das Kapitel zum Entwerfen für Transaktionsmanager behandelt die Verwendung von Datenbanken in einer Umgebung für verteilte Transaktionsverarbeitung wie CICS.

Systeme mit hoher Verfügbarkeit

- Das Kapitel zur Einführung in hohe Verfügbarkeit und Funktionsübernahme bietet eine Übersicht über die von DB2 bereitgestellte Unterstützung von Funktionsübernahmen zur Implementierung hoher Verfügbarkeit.

Anhänge

- Der Anhang zur Planung der Datenbankmigration enthält Informationen zur Migration von Datenbanken auf Version 7.
- Der Anhang zu Inkompatibilitäten zwischen Releases beschreibt die entstandenen Inkompatibilitäten von Release zu Release bis einschließlich Version 7.
- Der Anhang zur Unterstützung von Landessprachen beschreibt die DB2-Unterstützung von Landessprachen und enthält Informationen zu Ländern, Sprachen und Codepages.

Systemverwaltung: Optimierung

Der Band *Systemverwaltung: Optimierung* behandelt Themen zur Systemleistung. Damit sind die Themen gemeint, die das Einrichten, Testen und Optimieren der Leistung von Anwendungen sowie der Leistung des Produkts DB2 Universal Database betreffen. Die einzelnen Kapitel und Anhänge des Bandes werden im Folgenden kurz vorgestellt:

Einführung in die Optimierung

- Das Kapitel zu den Leistungselementen führt in die Konzepte und Überlegungen zur Verwaltung und Optimierung der Leistung von DB2 UDB ein.

- Die Übersicht zur Architektur und Verarbeitung stellt die zugrundeliegende Architektur und die Prozesse von DB2 Universal Database vor.

Optimieren der Anwendungsleistung

- Die Überlegungen zu den Anwendungen beschreiben einige Techniken zur Verbesserung der Datenbankleistung beim Entwurf der Anwendungen.
- Die Überlegungen zur Umgebung beschreiben einige Techniken zur Verbesserung der Datenbankleistung bei der Einrichtung der Datenbankumgebung.
- Das Kapitel zu den Systemkatalogstatistiken beschreibt einige Techniken zur Erfassung von Statistiken über die Daten und ihre Verwendung zur Gewährleistung einer optimalen Leistung.
- Das Kapitel zum SQL-Compiler beschreibt die Verarbeitung einer SQL-Anweisung durch den SQL-Compiler bei der Kompilierung.
- Das Kapitel zur SQL-EXPLAIN-Einrichtung beschreibt die Einrichtung EXPLAIN, die Ihnen ermöglicht, die Pfade und Methoden anzuzeigen, die der SQL-Compiler ausgewählt hat, um auf die Daten zuzugreifen.

Optimieren und Konfigurieren des Systems

- Das Kapitel zur Leistung bei der Ausführung gibt einen Überblick über die Verwendung von Speicher durch den Datenbankmanager und enthält Informationen zu weiteren Faktoren, die sich auf die Leistung zur Laufzeit auswirken.
- Das Kapitel zum Programm Governor bietet eine Einführung in die Verwendung des Programms Governor, mit dem einige Aspekte der Datenbankverwaltung gesteuert werden können.
- Das Kapitel zum Skalieren Ihrer Konfiguration enthält einige Informationen und Hinweise, die bei der Erweiterung des Datenbanksystems von Bedeutung sind.
- Das Kapitel zur Umverteilung von Daten auf Datenbankpartitionen behandelt die Punkte, die in einer Umgebung mit partitionierten Datenbanken bei der Neuverteilung von Daten auf die Partitionen zu beachten sind.
- Das Kapitel zu Vergleichstests (Benchmark-Tests) gibt einen Überblick über Vergleichstests und behandelt verschiedene Aspekte ihrer Durchführung.
- Das Kapitel zur DB2-Konfiguration behandelt die Dateien zur Konfiguration des Datenbankmanagers und der Datenbanken sowie die Werte für die Konfigurationsparameter.

Anhänge

- Der Anhang zu DB2-Registrierungsvariablen und DB2-Umgebungsvariablen enthält Werte für die Profilregistrierdatenbank und für Umgebungsvariablen.
- Der Anhang zu EXPLAIN-Tabellen und -Definitionen beschreibt die Tabellen, die von der DB2-EXPLAIN-Einrichtung verwendet werden, sowie die Erstellung dieser Tabellen.
- Der Anhang zu den SQL-Explain-Programmen enthält Informationen zur Verwendung der DB2-EXPLAIN-Programme: db2expln und dynexpln.
- Der Anhang zu db2exfmt - Tool für EXPLAIN-Tabellenformat - enthält Informationen dazu, wie mit dem DB2-EXPLAIN-Programm EXPLAIN-Tabellendaten formatiert werden.

Teil 1. Systemverwaltung mit der Steuerzentrale

Kapitel 1. Verwalten von DB2 mit den Tools für die grafische Benutzerschnittstelle

DB2 Universal Database bietet Tools für die grafische Benutzerschnittstelle (GUI), mit denen Sie auf einfache Weise lokale und ferne Datenbanken von einer zentralen Stelle - der Steuerzentrale - verwalten können.

In diesem Kapitel erhalten Sie eine Übersicht über die verfügbaren Tools zur Datenbankverwaltung von DB2 Universal Database und Erläuterungen zur effektiven und einfachen Anwendung dieser Tools. Außerdem wird die Java-Steuerzentrale und die Anpassung der Steuerzentrale für eigene Java-fähige Tools zusammenfassend dargestellt.

In diesem Kapitel finden Sie Informationen zu den folgenden Themen:

- „Verwaltungs-Tools“ auf Seite 4
- „Gemeinsame Funktionen der Tools“ auf Seite 7
- „Steuerzentrale“ auf Seite 12
- „Satellitenverwaltungszentrale“ auf Seite 21
- „Befehlszentrale“ auf Seite 22
- „Prozedurzentrale“ auf Seite 23
- „Journal“ auf Seite 25
- „Lizenzzentrale“ auf Seite 26
- „Alert-Zentrale“ auf Seite 26
- „Client-Konfiguration - Unterstützung“ auf Seite 27
- „Performance Monitor“ auf Seite 28
- „Verwalten ferner Datenbanken“ auf Seite 42
- „Verwalten von Benutzern“ auf Seite 44
- „Versetzen von Daten“ auf Seite 45
- „Verwalten von Speicher“ auf Seite 48
- „Fehlerbehebung“ auf Seite 51
- „Replizieren von Daten“ auf Seite 52
- „Verwenden des Protokolls LDAP (Lightweight Directory Access Protocol)“ auf Seite 53
- „Verwenden der Java-Steuerzentrale“ auf Seite 53
- „Verwenden der Java-basierten Tools zur Verwaltung“ auf Seite 54

Verwaltungs-Tools

Die Tools zur Verwaltung von DB2 sind Bestandteil von Administration Client, einer wählbaren Komponente, die in den meisten Produkten von DB2 Universal Database enthalten ist. Administration Client ist auch auf einem Satz CD-ROMs verfügbar, der die Verwaltungs-Clients für alle Betriebssysteme enthält, für die DB2 verfügbar ist. Damit können Sie Administration Client auf einer beliebigen Workstation installieren und verwenden. Hierbei spielt es keine Rolle, ob die Datenbankserver lokal oder fern installiert sind und welche Betriebssysteme auf den Datenbankservern eingesetzt werden. Mit den Tools können Sie in der grafischen Benutzerschnittstelle dieselben Funktionen ausführen wie mit dem Befehlszeilenprozessor. Zu diesen Funktionen zählen die Eingabe von DB2-Befehlen, SQL-Anweisungen oder Systembefehlen. Für den Umgang mit den grafischen Tools müssen Sie jedoch keine komplexen Anweisungen oder Befehle auswendig kennen, und Sie erhalten zusätzliche Unterstützung.

Anmerkung: Bei Administration Client handelt es sich um eine Installationsoption.

In der Menüleiste der Steuerzentrale sind die folgenden Tools verfügbar:

- Steuerzentrale. Die Steuerzentrale ist das wichtigste grafische DB2-Tool zur Verwaltung der Datenbank. Die Steuerzentrale gibt eine Übersicht über alle Systeme und Datenbankobjekte, die lokal katalogisiert sind.
- Satellitenverwaltungszentrale. Mit der Satellitenverwaltungszentrale können Sie DB2 Satellite-Server verwalten.
- Befehlszentrale. In der Befehlszentrale können Sie DB2-Datenbankbefehle, SQL-Anweisungen und Betriebssystembefehle absetzen. Außerdem können Befehle erneut aufgerufen werden, und Sie können durch die Zugriffspläne für SQL-Abfragen blättern.
- Prozedurzentrale. In der Prozedurzentrale können Sie Befehle des Betriebssystems, DB2-Befehlsprozeduren und SQL-Anweisungsprozeduren erstellen, ausführen und terminieren.
- Alert-Zentrale. In der Alert-Zentrale werden Sie informiert, wenn eingestellte Schwellen überschritten worden sind oder wenn ein Knoten innerhalb einer Umgebung mit mehreren Knoten nicht mehr antwortet.
- Journal. Im Journal können Sie den Status von Jobs anzeigen, Jobs erneut terminieren und das Wiederherstellungsprotokoll sowie das Nachrichtenprotokoll anzeigen.
- Information - Unterstützung. Mit Information - Unterstützung können sie schnell auf Informationen in den Handbüchern zu DB2-Produkten und Beispielprogrammen zugreifen. Außerdem sind andere Informationsquellen zu DB2 im Web verfügbar.

- **Lizenzzentrale.** In der Lizenzzentrale wird der Status Ihrer Lizenzen angezeigt, und Sie können hier das System für die geeignete Lizenzüberwachung konfigurieren.

Für einige Funktionen, die Sie mit den grafischen Tools ausführen können, ist ein Assistent verfügbar. Assistenten können in den Dialogfenstern der Steuerzentrale aufgerufen werden. Diese bieten eine noch weitreichendere Unterstützung, da Schritt für Schritt zur Eingabe der Daten aufgefordert wird, die für die jeweilige Aufgabe erforderlich sind. Anhand der eingegebenen Daten werden selbst Vorausberechnungen ausgeführt und Empfehlungen gegeben. Assistenten sind für neue Datenbankadministratoren oder für Personen, die nur gelegentlich eine Datenbank verwalten, sehr hilfreich.

In DB2 Universal Database gibt es die folgenden Assistenten:

- **Assistent: Datenbank sichern.** Dieser Assistent stellt dem Benutzer Fragen zu den Daten in der Datenbank, zur Verfügbarkeit der Datenbank und zu den Anforderungen an die Wiederherstellbarkeit der Daten. Anschließend wird ein Sicherungsplan vorgeschlagen und die Jobprozedur erstellt und terminiert. Sie können den **Assistent: Datenbank sichern** aufrufen, indem Sie das Symbol der zu sichernden Datenbank auswählen, mit der rechten Maustaste klicken und dann **Sichern** → **Datenbank mit Assistent** auswählen.
- **Assistent: Datenbank erstellen.** Mit diesem Assistenten können Sie eine Datenbank erstellen, Speicherplatz zuordnen und grundlegende Leistungsoptionen auswählen. Sie können den **Assistent: Datenbank erstellen** aufrufen, indem Sie das Symbol für die Datenbanken auswählen, mit der rechten Maustaste klicken und dann **Erstellen** → **Datenbank mit Assistent** auswählen.
- **Assistent: Tabelle erstellen.** Dieser Assistent unterstützt Sie beim Einrichten von Spalten mit vordefinierten Spaltenschablonen, beim Erstellen eines Primärschlüssels für die Tabelle und beim Zuordnen der Tabelle zu einem oder mehreren Tabellenbereichen. Sie können den Assistenten aufrufen, indem Sie das Symbol für Tabellen auswählen, mit der rechten Maustaste klicken und dann **Erstellen** → **Tabelle mit Assistent** auswählen.
- **Assistent: Tabellenbereich erstellen.** Mit diesem Assistenten können Sie einen neuen Tabellenbereich erstellen und grundlegende Speicher- und Leistungsoptionen definieren. Sie können diesen Assistenten aufrufen, indem Sie das Symbol für den Tabellenbereich auswählen, mit der rechten Maustaste klicken und dann **Erstellen** → **Tabellenbereich mit Assistent** auswählen.
- **Assistent: Index erstellen.** Mit diesem Assistenten können Sie bestimmen, welche Indizes für eine angegebene Gruppe von SQL-Anweisungen erstellt oder gelöscht werden sollen. Die Empfehlungen basieren auf der Auslastung, die Sie angeben. Sie können den **Assistent: Index erstellen** aufrufen,

indem Sie den Ordner 'Indizes' auswählen, mit der rechten Maustaste klicken und dann **Erstellen** —> **Index mit Assistent** auswählen.

- **Assistent: Leistungskonfiguration.** Dieser Assistent unterstützt Sie bei der Optimierung von Datenbanken und fordert Sie zur Eingabe von Informationen zur Datenbank, zu den dort gespeicherten Daten und zum Zweck des Systems auf. Es werden neue Konfigurationsparameter für die Datenbank und das Exemplar empfohlen und, falls gewünscht, automatisch angewendet. Sie können diesen Assistenten aufrufen, indem Sie das Symbol für eine Datenbank auswählen, mit der rechten Maustaste klicken und dann **Leistung mit Assistent konfigurieren** auswählen.
- **Assistent: Datenbank wiederherstellen.** Mit diesem Assistenten werden Sie Schritt für Schritt durch die Wiederherstellung einer Datenbank geführt. Sie können diesen Assistenten aufrufen, indem Sie das Symbol für eine Datenbank auswählen, mit der rechten Maustaste klicken und dann **Wiederherstellen** —> **Datenbank mit Assistent** auswählen.
- Konfigurieren von Aktualisierungen auf mehreren Systemen. Mit diesem Assistenten können Sie Datenbanken so konfigurieren, dass mit Anwendungen mehrere Systeme gleichzeitig aktualisiert werden können. Dies ist in Fällen wichtig, in denen die Daten auf allen Systemen konsistent sein müssen. Sie können diesen Assistenten aufrufen, indem Sie ein Exemplar auswählen, mit der rechten Maustaste klicken und dann **Aktualisierung auf mehreren Systemen** —> **Konfigurieren** auswählen.

Anmerkung: Für das DB2 für OS/390-Subsystem sind keine Assistenten verfügbar.

Außer den Tools mit grafischer Benutzerschnittstelle, die Sie in der Menüleiste der Steuerzentrale aufrufen können, gibt es weitere Tools mit grafischer Benutzerschnittstelle, die nicht unmittelbar über diese Menüleiste aufgerufen werden können.

- **Performance Monitor.** Der Performance Monitor ist ein Tool zur Überwachung von DB2-Objekten wie Exemplaren, Datenbanken, Tabellen, Tabellenbereichen und Verbindungen. Sie können dieses Tool zur Erkennung von Leistungsdefiziten und zur Optimierung von Datenbanken verwenden. Der Performance Monitor kann in der Steuerzentrale mit einer Option der Dialogfenster aufgerufen werden.
- **Ereignismonitor.** Der Ereignismonitor ist ein Tool, mit dem Sie die Ressourcennutzung analysieren können. Dazu wird der Status der Datenbank zurzeit des Auftretens bestimmter Ereignisse protokolliert. Sie können einen Ereignismonitor erzeugen, indem Sie in eine DB2-Befehlszeile den Befehl `db2emcrt` eingeben.

- **Event Analyzer.** Der Event Analyzer ist ein Tool, mit dem Sie die Daten analysieren können, die vom Ereignismonitor aufgezeichnet werden. Sie können einen Event Analyzer aufrufen, indem Sie in eine DB2-Befehlszeile den Befehl `db2evmon` eingeben.
- **Visual Explain.** Mit der Funktion Visual Explain können Sie den Zugriffsplan für SQL-Anweisungen als Diagramm anzeigen, so dass Sie die SQL-Abfragen entsprechend optimieren können. Vor Version 6 wurde das Tool Visual Explain verwendet, um Zugriffspläne anzuzeigen. Unter der aktuellen Version ist Visual Explain kein separates Tool mehr. Die Funktion ist jedoch in den Kontextmenüs verschiedener Datenbankobjekte innerhalb der Steuerzentrale und in der Befehlszentrale verfügbar.

Außerdem gibt es ein weiteres nützliches Tool zur Datenbankverwaltung, das nicht Bestandteil der Steuerzentrale ist - die Client-Konfiguration - Unterstützung. Die Client-Konfiguration - Unterstützung ist ein Tool, das Assistenten umfasst, die den Benutzer beim Konfigurieren von Clients für die Kommunikation mit fernen Servern unterstützen.

Diese Tools werden alle weiter unten genauer beschrieben. Im folgenden Abschnitt erhalten Sie eine Übersicht über die Funktionen dieser Tools.

Gemeinsame Funktionen der Tools

Die folgenden Funktionen sind in mehreren Tools verfügbar:

- **SQL anzeigen** und **Befehl anzeigen**
- Zugehörige anzeigen
- DDL generieren
- Filtern
- Hilfe

"SQL anzeigen" und "Befehl anzeigen"

Wenn ein Tool SQL-Anweisungen generiert, ist in der Benutzerschnittstelle des Tools der Druckknopf **SQL anzeigen** verfügbar. Ebenso ist für Tools, mit denen DB2-Befehle generiert werden, der Druckknopf **Befehl anzeigen** verfügbar. Wenn Sie einen dieser Druckknöpfe anklicken, können Sie die folgenden Operationen ausführen:

- Anzeigen der SQL-Anweisungen oder DB2-Befehle, die vom Tool aufgrund der Auswahl in der grafischen Benutzerschnittstelle generiert werden. Anhand dieser Informationen können Sie leichter erkennen, wie die Benutzerschnittstelle funktioniert.
- Speichern der Anweisungen oder Befehle als Prozedur, um diese später wiederzuverwenden. Mit dieser Funktion können Sie vermeiden, die SQL-Anweisungen oder DB2-Befehle erneut einzugeben, wenn Sie dieselben Anweisungen oder Befehle ausführen möchten. Nachdem die SQL-Anwei-

sungen oder DB2-Befehle in einer Prozedur gespeichert worden sind, können Sie die Prozedur zeitlich terminieren, bearbeiten oder ähnliche Prozeduren erstellen, ohne die Anweisungen oder Befehle erneut eingeben zu müssen.

Gehen Sie wie folgt vor, um die SQL-Anweisungen oder DB2-Befehle anzuzeigen:

1. Wechseln Sie von der Steuerzentrale in ein Fenster oder Notizbuch, um mit einem Tool zu arbeiten, das SQL-Anweisungen oder DB2-Befehle generiert. Der Druckknopf **SQL anzeigen** oder der Druckknopf **Befehl anzeigen** wird als verfügbar angezeigt.
2. Klicken Sie den Druckknopf **SQL anzeigen** oder **Befehl anzeigen** an. Das entsprechende Fenster wird geöffnet.

Das Speichern von SQL-Anweisungen und DB2-Befehlen ist besonders zweckmäßig, wenn es sich um komplexe SQL-Anweisungen oder DB2-Befehle handelt.

Wenn Sie den Druckknopf **Befehl anzeigen** oder **SQL anzeigen** verwenden, können Sie entweder neue Prozeduren erstellen, die Sie später bearbeiten, oder Sie schließen das Dialogfenster, um in das ursprüngliche Dialogfenster zurückzukehren und dort Änderungen vorzunehmen. Wenn Sie den Druckknopf 'Prozedur erstellen' anklicken, wird das Fenster 'Neue Befehlsprozedur' angezeigt. Hier können Sie die SQL-Anweisungen oder DB2-Befehle editieren, bevor Sie die Prozedur speichern.

"Zugehörige anzeigen"

Mit **Zugehörige anzeigen** werden die direkten Beziehungen zwischen Tabellen, Indizes, Sichten, Aliasnamen, Auslösern, Tabellenbereichen, benutzerdefinierten Funktionen und benutzerdefinierten Datentypen angezeigt. Wenn Sie z. B. eine Tabelle auswählen und die zugehörigen Sichten anzeigen, erhalten Sie nur Sichten, die direkt auf der Tabelle basieren. Sichten, die auf den zugehörigen Sichten basieren, werden nicht angezeigt, da diese nicht direkt aus der Tabelle erstellt worden sind.

Die Anzeige zugehöriger Objekte dient den folgenden Zwecken:

- Verständnis der Datenbankstruktur.
- Bestimmung der für eine Tabelle bereits vorhandenen Indizes.
- Bestimmung der Objekte, die in einem Tabellenbereich gespeichert sind.
- Bestimmung anderer Objekte, die zu einem Objekt gehören und auf die sich somit Ihre Aktionen auswirken. Wenn Sie z. B. eine Tabelle löschen möchten, die abhängige Sichten besitzt, werden mit **Zugehörige anzeigen** die Sichten angezeigt, die unbrauchbar werden.

Gehen Sie wie folgt vor, um die Funktion **Zugehörige anzeigen** zu verwenden:

- Wählen Sie in der Steuerzentrale ein Objekt im Inhaltsteilfenster aus und klicken Sie dann mit der rechten Maustaste.
- Wählen Sie **Zugehörige anzeigen** aus.
- Klicken Sie die Indexzunge an, um die Seite mit den gewünschten zugehörigen Objekten zu öffnen. Je nachdem, welche Indexzunge Sie auswählen, werden unterschiedliche zugehörige Objekte aufgelistet. Nur Objekte, die direkt zum ausgewählten Objekt gehören, werden angezeigt.
Sie können mit der rechten Maustaste klicken, während sich der Mauszeiger über einem zugehörigen Objekt auf der ausgewählten Seite befindet, und im Kontextmenü die Option **Zugehörige anzeigen** auswählen. Die Ausgewählte Seite ändert sich, und es werden die Objekte angezeigt, die zu Ihrer letzten Auswahl gehören. Sie können auch den Abwärtspfeil neben dem ausgewählten Objekt anklicken, um eine Liste der zuvor ausgewählten Objekte mit den zugehörigen Beziehungen anzuzeigen.
- Klicken Sie **Schließen** an, um das Notizbuch **Zugehörige anzeigen** zu schließen und in die Steuerzentrale zurückzukehren.

"DDL generieren"

Mit der Funktion **DDL generieren** können Sie die DDL- und SQL-Anweisungen und die entsprechende Statistiken erneut erstellen und in einer Prozedurdatei speichern. Dies gilt für:

- Datenbankobjekte
- Berechtigungsanweisungen
- Tabellenbereiche, Knotengruppen und Pufferpools
- Datenbankstatistiken

Dies dient den folgenden Zwecken:

- Speichern der DDL-Anweisungen, um identisch definierte Tabellen Datenbanken und Indizes in einer anderen Datenbank zu erstellen, z. B. für eine Datenbankanwendung zur Lagerverwaltung.
- Verwendung der DDL-Anweisungen zum Kopieren einer Datenbank aus der Testumgebung in eine Produktionsumgebung oder von einem System auf ein anderes.
- Editieren der DDL-Anweisungen, um ähnliche Objekte zu erstellen.

Wenn Sie den Druckknopf **DDL generieren** anklicken, wird das Fenster 'Befehl anzeigen' aufgerufen. Dieses enthält Anweisungen, die vom Dienstprogramm **db2look** generiert wurden. Im Fenster 'Befehl anzeigen' können Sie den Druckknopf **Prozedur sichern** anklicken, um die Anweisungen zu sichern. Die Anweisungen werden in eine Prozedur geschrieben. Wenn Sie den Knopf **Generieren** anklicken, wird das Fenster **Prozedur ausführen** geöffnet.

Anmerkung: Das Generieren von DDL-Anweisungen wird anders ausgeführt, wenn Sie mit der Steuerzentrale für das System 390 arbeiten. Detaillierte Informationen zu den geltenden Unterschieden enthält der zugehörige Hilfetext.

Sie können auswählen, ob DDL-Anweisungen für die ausgewählten Schemata oder für alle Schemata innerhalb der Datenbank generiert werden sollen. Anschließend können Sie die Prozedur nach Bedarf editieren, bevor Sie sie in einer Produktionsumgebung verwenden. Zur Erstellung identischer Datenbanken mit Hilfe der generierten DDL-Anweisungen verwenden Sie einfach die generierte Prozedur und führen diese in der neuen Umgebung aus.

Gehen Sie wie folgt vor, um DDL-Anweisungen zu generieren:

1. Heben Sie das Objekt hervor, für das Sie DDL-Anweisungen generieren möchten, und klicken Sie dann mit der rechten Maustaste.
2. Wählen Sie **DDL generieren** aus. Das Fenster **Prozedur ausführen** wird angezeigt.
3. Geben Sie eine Benutzer-ID und ein Kennwort ein, und klicken Sie **OK** an. Es wird ein Job anhand der Angaben für den Befehl **db2look** erstellt. In einem Nachrichtenfenster von DB2 wird die Job-ID des neuen Jobs angezeigt.
4. Klicken Sie **OK** an, um das Nachrichtenfenster zu schließen.
5. Verwenden Sie die Seite **Jobprotokoll** des Notizbuchs **Journal**, um die Ergebnisse des Jobs und den Inhalt einer zum Job gehörigen gespeicherten Prozedur anzuzeigen.
6. Wählen Sie den Job aus und klicken Sie mit der rechten Maustaste. Wählen Sie **Ergebnisse anzeigen** im Kontextmenü aus. Das Fenster **Jobergebnisse** wird geöffnet. Die Ausgabe des Befehls **db2look** wird im Teilfenster **Jobausgabe** angezeigt.
7. Wählen Sie **Prozedur erstellen** aus, um aus den Ergebnissen eine Prozedur zu erstellen. Das Fenster **Neue Befehlsprozedur** wird angezeigt.
8. Wenn Sie die Prozedur wiederverwenden wollen, speichern Sie diese.

"Filtern"

In der Steuerzentrale können Sie Informationen, die im Inhaltsteilfenster angezeigt werden, filtern. Oder Sie können Informationen filtern, die aus einer Tabelle als eine tatsächliche Ergebnismenge abgerufen werden. Sie können die Anzahl der Objekte, die angezeigt werden, oder die Anzahl der Objekte, die zurückgegeben werden, begrenzen, indem Sie für ein oder mehrere Objekte Filter erstellen. Nachdem Sie den Filter festgelegt haben, müssen Sie zur erneuten Anzeige aller Objekte im Objektbaum den Filter entfernen oder löschen.

Filtern der Anzeige

Gehen Sie wie folgt vor, um die Anzahl der im Inhaltsteilfenster angezeigten Objekte zur übersichtlicheren Verwaltung zu verringern:

1. Wählen Sie in der Menüleiste des Inhaltsteilfensters das Symbol für den Filter aus, das sich in der Steuerzentrale unten befindet, oder wählen Sie in der Menüleiste **Sicht** die Option **Filtern** aus.
2. Wählen Sie die Bedingungen aus, die zur Verringerung der Objektanzahl verwendet werden sollen.
3. Wählen Sie das Markierungsfeld **Filter aktivieren** aus, um den Filter zu aktivieren.

Wenn Sie später ein Objekt auswählen, um dessen Inhalt anzuzeigen, begrenzt der Filter, den Sie dem Objekt zugeordnet haben, die Sicht entsprechen den eingestellten Bedingungen.

Filtern abgerufener Daten

Sie können die Anzahl der in einer Abfrage zurückgegebenen Zeilen verringern und die Antwortzeit verbessern, indem Sie die Ausgabe bzw. die Ergebnismenge definieren, die bei der Objektauswahl im Inhaltsteilfenster angezeigt wird. Gehen Sie dazu wie folgt vor:

1. Wählen Sie in der Baumstruktur ein Ordnerobjekt aus und klicken Sie dann mit der rechten Maustaste.
2. Wählen Sie im Kontextmenü die Option **Filtern** aus. Das Fenster **Filtern** wird geöffnet.
3. Verwenden Sie die Funktion **Filtern**, um eine Menge von Bedingungen zu definieren, mit denen die zum betreffenden Objekt gehörigen Zeilen abgerufen werden.

Definieren eines Filters zum Abruf einer bestimmten Menge von Daten

Gehen Sie wie folgt vor, um einen Filter zum Abruf einer bestimmten Menge von Daten zu definieren:

1. Erweitern Sie in der Steuerzentrale die Ordner mit den Datenbanken oder Subsystemen (je nach Plattform).
2. Wählen Sie ein Objekt aus, für das Sie den Filter definieren wollen. Klicken Sie, während sich der Mauszeiger über dem entsprechenden Objekt befindet, mit der rechten Maustaste.
3. Wählen Sie im Kontextmenü die Option **Filtern** aus. Dadurch wird das Notizbuch **Filtern** geöffnet.
4. Geben Sie auf der Seite **Suchen** den Namen oder eine andere beschreibende Filterbedingung des ausgewählten Objekts ein. Das Ergebnis des Filterns ist die Ergebnismenge, die dem ausgewählten Objekt zugeordnet ist und die im Inhaltsteilfenster der Steuerzentrale angezeigt wird.

5. Wählen Sie auf der Seite **Suchen** einen Radioknopf aus, um anzugeben, ob alle Bedingungen oder mindestens eine der Bedingungen erfüllt werden müssen, die in den Feldern auf der Seite **Suchen** ausgewählt worden sind.
6. Verwenden Sie nach Bedarf weitere Bedingungen auf der Seite **Erweitert**, indem Sie den angezeigten Text editieren, um die Anzahl der zurückgegebenen Zeilen weiter zu begrenzen.
7. Klicken Sie **OK** an, um die definierten Filterbedingungen zu verwenden.

Sie können dieses auf der Anzahl der Zeilen basierende Filternotizbuch automatisch aufrufen, indem Sie in der Menüleiste die Option **Tools** und im Kontextmenü die Option **Tools - Einstellungen** auswählen. Mit dem Markierungsfeld **Filtern aktivieren, wenn die Anzahl der Zeilen Folgendes überschreitet** können Sie einen von der Auswahl unabhängigen Schwellenwert für zurückgegebene Zeilen vordefinieren. Sobald dieser erreicht wird, wird das Notizbuch **Filtern** angezeigt, so dass Sie die aktuelle Abfrage auf der Basis der definierten Bedingungen begrenzen können. Dies ist insbesondere zweckmäßig, wenn eine Tabelle unerwartet groß geworden ist und zuvor nicht gefiltert wurde. Je nach Plattform und Daten werden möglicherweise Millionen von Zeilen zurückgegeben, während Sie lediglich einen Teil der Zeilen benötigen.

Hilfe

Sie erhalten zu den Verwaltungs-Tools ausführliche Hilfetexte. In allen Fenstern und Notizbüchern und in der Menüleiste befindet sich der Knopf 'Hilfe'. Sie können eine erweiterte Hilfe aufrufen oder eine Hilfe zur Eingabe in Felder oder zur Ausführung von Aufgaben erhalten. Von den Hilfemenüs aus können Sie außerdem auf einen Index mit Begriffen oder auf Referenzinformationen und Informationen in Handbüchern zugreifen.

Steuerzentrale

Ihr wichtigstes Hilfsmittel ist die Steuerzentrale: In ihr können Sie Systeme, DB2-Exemplare, Datenbanken, Datenbankobjekte, wie Tabellen, Sichten und Benutzergruppen verwalten. Außerdem können Sie mit der Steuerzentrale auf Subsysteme mit DB2 für OS/390 zugreifen. Alle DB2-Datenbanken müssen katalogisiert werden, bevor Sie in der Steuerzentrale angezeigt werden. In Abb. 1 auf Seite 13 sind die wichtigsten Funktionen der Steuerzentrale dargestellt. Aufgrund von Unterschieden zwischen den Betriebssystemen unterscheidet sich die Steuerzentrale auf Ihrem System möglicherweise von der Abbildung.

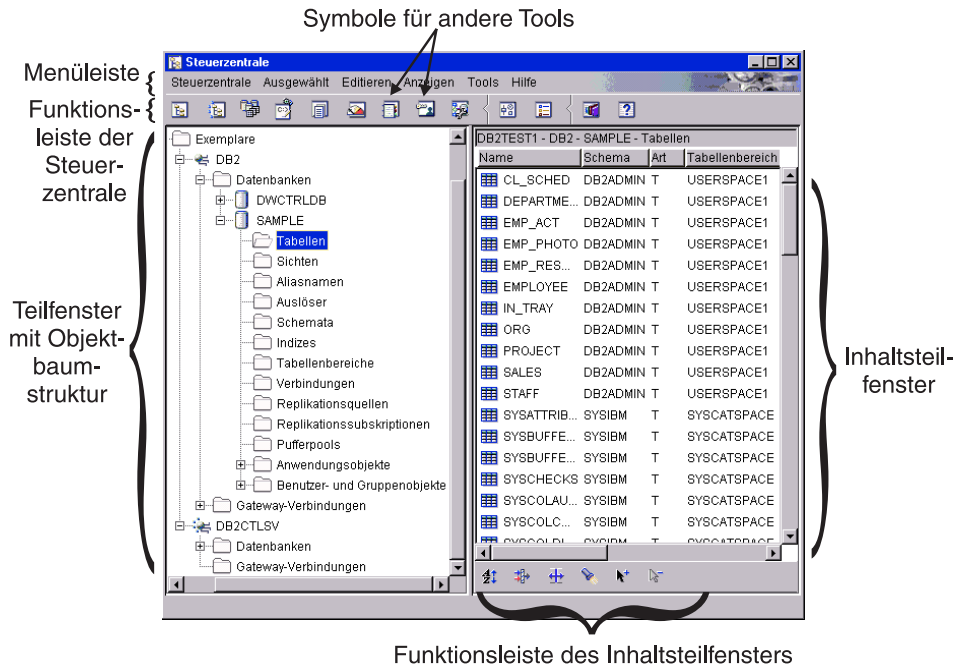


Abbildung 1. Funktionen der Steuerzentrale

Hauptelemente der Steuerzentrale

Die Steuerzentrale besteht aus den folgenden Hauptelementen:

- **Menüleiste.** Die Menüleiste befindet sich oben in der Anzeige. Wenn Sie in der Menüleiste ein Menü auswählen, können Sie viele verschiedene Funktionen aufrufen: z. B. das Beenden von DB2-Tools, der Zugriff auf Tools mit grafischer Benutzerschnittstelle und der Zugriff auf die Online-Hilfefunktion und die Produktinformation. Sie sollten sich mit diesen Funktionen vertraut machen, indem Sie in der Menüleiste die einzelnen Elemente anklicken.

- **Menüleiste der Steuerzentrale.** Die Symbole für die Steuerzentrale und andere Tools können in der Menüleiste der Steuerzentrale aufgerufen werden. Während der Mauszeiger sich auf den einzelnen Symbolen befindet, wird eine Kurzinfo angezeigt.

Sie können die Einstellungen für diese Tools ändern, indem Sie das Symbol für **Tools - Einstellungen** in der Menüleiste der Steuerzentrale auswählen.

- **Objektbaum.** Der Objektbaum befindet sich im linken Teilfenster der Anzeige. In ihm werden für alle Datenbankserver und -objekte, die Sie in der Steuerzentrale verwalten, Symbole angezeigt. Sie müssen einen fernen Datenbankserver katalogisieren, bevor dieser im Teilfenster mit dem Objekt-

baum angezeigt wird. Einige Objekte im Teilfenster mit dem Objektbaum enthalten weitere Objekte. Wenn sich links vom Objekt ein Pluszeichen (+) befindet, ist das Objekt ausgeblendet. Sie können es einblenden, indem Sie das Pluszeichen anklicken. Wenn ein Objekt einblendet ist, wird links davon ein Minuszeichen (–) angezeigt. Sie können das Objekt ausblenden, indem Sie das Minuszeichen anklicken.

- **Inhaltsteilfenster.** Das Inhaltsteilfenster befindet sich im rechten Teilfenster der Anzeige. In diesem Teilfenster werden alle Objekte angezeigt, die im ausgewählten Objekt des Objektbaumfensters enthalten sind, z. B. werden alle Tabellen im Inhaltsteilfenster angezeigt, sobald Sie im Objektbaumfenster den Tabellenordner auswählen. Wenn Sie den Datenbankordner auswählen, ändert sich der Inhalt des Inhaltsteilfensters, und alle Datenbanken werden angezeigt. Sie können die Spalten, die im Inhaltsteilfenster angezeigt werden, filtern, indem Sie in der Menüleiste des Inhaltsteilfensters das Symbol für das Filtern auswählen und die erforderlichen Daten eingeben. Oder Sie können Objekte filtern, indem Sie in der Menüleiste die Option **Tools** und anschließend **Tools - Einstellungen** auswählen. Im Inhaltsteilfenster muss das Markierungsfeld **Filter aktivieren** ausgewählt sein.
- **Menüleiste des Inhaltsteilfensters.** Diese Menüleiste befindet sich unten im Inhaltsteilfenster. Damit können Sie die Informationen im Inhaltsteilfenster anpassen. Diese Menüleiste ist ein gemeinsames Steuerelement der meisten Detailsichten innerhalb des gesamten Programms und befindet sich unter oder neben der jeweiligen Detailsicht.

Beim Arbeiten in der Steuerzentrale werden bestimmte Felder möglicherweise durch einen dicken roten Rahmen hervorgehoben. Dieser Rahmen gibt an, dass es sich bei dem Feld um ein Mußeingabefeld handelt, in das Daten eingegeben werden müssen. Nach der Auswahl bzw. Eingabe eines Wertes wird der rote Rahmen wieder ausgeblendet.

Verwenden einer angepassten Steuerzentrale in DB2 für OS/390

Sie können die **Angepasste Steuerzentrale** auf der Plattform DB2 für OS/390 als selbst definierte Zentrale verwenden, um Subsysteme, Datenbanken oder Datenbankobjekte, wie z. B. Tabellen, Sichten und Datenbankbenutzer zu verwalten. Mit der angepassten Steuerzentrale können Sie auf beliebige Objekte in DB2 für OS/390 zugreifen, die Sie definieren.

Die Hauptelemente der angepassten Steuerzentrale entsprechen den Elementen, die für die Standardsteuerzentrale aufgelistet wurden. In einer angepassten Steuerzentrale können Sie jedoch Objekte angeben, die diese enthalten soll. Dieser benutzerdefinierte Objektbaum kann zur Verwaltung von DB2-Objekten gespeichert und aufgerufen werden. Dies ist kein Ersatz für den Objektbaum in der Steuerzentrale, die die Standardeinstellung für alle Benutzer darstellt. Wenn Sie jedoch auf eine bestimmte Menge von Objekten zugreifen möchten, sobald die Steuerzentrale aufgerufen wird, ist dies zweckmäßig.

Sie können so viele Objektbäume erstellen, wie Sie benötigen. Jeder Baum kann eine andere Objektmenge enthalten und in einer beliebigen Weise geordnet werden.

Wenn Sie einen angepassten Objektbaum verwenden, müssen Sie nicht in einer festen Hierarchie von DB2-Objekten navigieren und können zugehörige Objekte gruppieren. Sie können z. B. einen Baum definieren, der nur Tabellen mit Lohnbuchhaltungsdaten enthält.

Systeme, die verwaltet werden können

Von der Steuerzentrale aus können Sie Datenbankobjekte für die Produktfamilie DB2 Universal Database für OS/2-, Windows- und UNIX-Plattformen verwalten. Plattformspezifische Installations- und Konfigurationsinformationen hierzu enthalten die Handbücher *Einstieg*.

Sie können auch Daten von DB2 für AS/400, DB2 für VSE und VM sowie DB2 für OS/390 für die Produktfamilie DB2 Universal Database replizieren. Informationen zur Replikation zwischen Produkten finden Sie im *Replikation Benutzer- und Referenzhandbuch*.

Objekte, die verwaltet werden können

Wenn Sie Objekte von der Steuerzentrale aus verwalten möchten, müssen Sie diese dem Objektbaum hinzufügen. Wenn Sie außerhalb der Steuerzentrale eine Datenbank löschen oder sie aus dem Katalog entfernen, und Sie möchten in der Steuerzentrale damit weiterarbeiten, müssen Sie diese dem Objektbaum hinzufügen.

Die folgenden Objekte in DB2 Universal Database können Sie von der Steuerzentrale aus verwalten:

- Systeme
- Exemplare
- Tabellen
- Sichten
- Indizes
- Auslöser
- Benutzerdefinierte Datentypen
- Benutzerdefinierte Funktionen
- Pakete
- Aliasnamen
- Replikationsobjekte
- Benutzer und Gruppen

Die folgenden Objekte in DB2 für OS/390 Version 5 können Sie von der Steuerzentrale aus verwalten:

- Pufferpools
- Sichten
- Katalogtabellen
- Speichergruppen
- Aliasnamen
- Synonyme
- DB2-Benutzer
- Positionen
- Anwendungsobjekte (Objektgruppen, Pakete, Pläne Prozeduren)
- Datenbanken
- Tabellen
- Tabellenbereiche
- Indizes
- Replikationsquelle
- Replikationssubskriptionen

Unter DB2 für OS/390 Version 6 können Sie zusätzlich zu den unter Version 5 dieses Produkts unterstützten Objekten die folgenden Objekte über die Steuerzentrale verwalten:

- Schemata
- Auslöser
- Benutzerdefinierte Funktionen
- Einzigartige Datentypen

Die Aktionen, die Sie mit diesen Objekten ausführen können, werden angezeigt, wenn Sie im Teilfenster mit den Objekten das Objekt auswählen und dann mit der rechten Maustaste klicken. In einem Dialogfenster werden die Funktionen angezeigt.

Anzeigen von Systemen in der Steuerzentrale

Gehen Sie wie folgt vor, um alle Systeme anzuzeigen, die auf Ihrem System katalogisiert sind und auf denen DB2 installiert ist:

1. Erweitern Sie den Objektbaum, indem Sie das Pluszeichen (+) neben **Systeme** anklicken. Symbole, die die lokale Maschine und alle fernen Maschinen darstellen, werden angezeigt. Ihr lokales System wird durch das Symbol **Lokal** dargestellt. Es wird nur angezeigt, wenn die lokale Maschine ein DB2-Server ist. Wenn Sie mit der rechten Maustaste klicken, während sich der Mauszeiger über dem Symbol **Lokal** befindet, erhalten Sie im Kontextmenü unter anderem die Option zur Herstellung einer Verbindung zum

Verwaltungsserver. Mit dem Verwaltungsserver können Sie Funktionen nutzen, wie z. B. den Performance Monitor und die zeitliche Terminierung. Er wird von den DB2-Verwaltungs-Tools verwendet, um DB2-Serviceanforderungen zu verarbeiten, und wird automatisch erstellt und gestartet. Der Standardwert für den Namen des DB2-Verwaltungsservers kann bei den einzelnen Plattformen variieren. Auf Windows- und OS/2-Plattformen wird z. B. der Name „DB2DAS00“ verwendet, unter AIX der Name „db2as“.

2. Erweitern Sie das Symbol **Lokal**. Der Exemplar von DB2 auf der lokalen Maschine wird im Baum angezeigt.

Unter OS/2, Windows und den unterstützten UNIX-basierten DB2-Systemen können Sie sich die einzelnen Kopien des Datenbankmanagercodes als separate *Exemplare* vorstellen, die in einem Verzeichnis auf Ihrer Maschine gespeichert werden. In DB2 für OS/390 wird ein Exemplar als Subsystem aufgefasst. Wenn Sie DB2 installieren wird ein lokales Standardexemplar erstellt. Auf einem einzelnen System können sich mehrere Exemplare befinden. Diese Exemplare können Sie dazu verwenden, die Entwicklungsumgebung von der Produktionsumgebung zu trennen oder sensible Daten nur einer bestimmten Gruppe von Benutzern verfügbar zu machen. Außerdem können Sie ein Exemplar für eine bestimmte Umgebung optimieren.

3. Erweitern Sie das Symbol **Exemplare**. Für die einzelnen vorhandenen Datenbanken wird jeweils ein Symbol und der entsprechende Name angezeigt.

Verwalten von Objekten in DB2 für OS/390

Mit Hilfe der Steuerzentrale können Sie viele der Funktionen ausführen, die in den vorhandenen Produkten DB2 für OS/390 Version 5 und DB2 UDB für OS/390 Version 6 möglich sind, wie z. B. das Erstellen, Ändern und Löschen von Objekten sowie die Ausführung von Dienstprogrammen zum Laden oder Reorganisieren der Daten. Bevor Sie jedoch ein DB2 für OS/390-Subsystem von der Steuerzentrale aus verwalten können, müssen Sie dieses dem Objektbaum hinzufügen, indem Sie dafür eine Verbindung konfigurieren.

Hinzufügen von DB2 für OS/390-Subsystemen

Wenn Client-Konfiguration - Unterstützung installiert ist, können Sie diese Funktion zur einfachen Konfiguration einer Verbindung zu einem DB2 für OS/390-System verwenden. Wenn Client-Konfiguration - Unterstützung nicht installiert ist, müssen Sie die Verbindung zum DB2 für OS/390-System manuell über den Befehlszeilenprozessor (CLP = Command Line Processor) konfigurieren.

Sie können Client-Konfiguration - Unterstützung verwenden, um im Netzwerk nach allen DB2 für OS/390-Systemen zu suchen, die im LAN für Ihren Client verfügbar sind. Wenn Sie eines der DB2 für OS/390-Systeme hinzufügen möchten, können Sie den **Assistent: Datenbank hinzufügen** verwenden,

um das System hinzuzufügen, eine Verbindung unter Verwendung eines Profils importieren oder die Verbindung manuell hinzufügen.

Wenn Sie sich für die Suche im Netzwerk entscheiden, muss sich im Netzwerk ein DB2 Connect-Produkt mit einer für das System definierten Verbindung befinden. Wenn Sie ein Zugriffsprofil verwenden, müssen Sie im Profil die DB2 Connect-Serververbindung auswählen, die dem System zugeordnet ist. Wenn Sie die Verbindung manuell konfigurieren wollen, benötigen Sie den Namen des Systems, das Kommunikationsprotokoll und die Parameter für das Kommunikationsprotokoll. Hierzu gehören z. B. Angaben zum Host-Namen und zur TCP/IP-Anschlussnummer oder zum symbolischen Bestimmungsnamen für SNA. Nach dem Hinzufügen des DB2 für OS/390-Systems werden die Objekte für die DB2 Connect-Serververbindungen auf dem lokalen System der Steuerzentrale angezeigt.

Wenn Sie ein DB2 für OS/390-System ab Version 5 hinzufügen, wird dieses in einem eigenen Bereich der Objektbaumstruktur für die Steuerzentrale angezeigt. Wenn Sie die Objekte von DB2 für OS/390 und andere Datenbankobjekte anzeigen wollen, die sich in einem bestimmten System befinden, müssen Sie die Sicht der Objektbaumstruktur beim DB2 für OS/390-Systemsymbol, das für Ihr DB2 für OS/390-System steht, erweitern.

Sie können die Liste der Aktionen anzeigen, die Sie mit einem bestimmten Objekt ausführen können. Dazu wählen Sie das Objekt in der Objektbaumstruktur aus und klicken dann mit der rechten Maustaste. Ein Kontextmenü wird angezeigt, in dem die verfügbaren Aktionen für das Objekt aufgeführt sind. Sie können z. B. eine Sicht erstellen, ändern oder löschen und deren Inhalt anzeigen, die Zugriffsrechte auf die Sicht ändern und eine Liste weiterer Objekte anzeigen, die zur Sicht gehören. Weitere Informationen zu den möglichen Funktionen finden Sie in der Online-Hilfefunktion für DB2 zu OS/390-Objekten.

Verwalten von Gateway-Verbindungen

Wenn ein DB2 Connect-Server katalogisiert wird, wird in der Objektbaumstruktur der Steuerzentrale unter dem Exemplarobjekt des lokalen Systems ein Ordner **Gateway-Verbindungen** angezeigt. Der Ordner **Gateway-Verbindungen** enthält eine Hierarchie von Objekten. Diese wird verwendet, um Verbindungen zu Host- und IBM AS/400-Datenbanken zu verwalten, die lokal katalogisiert werden. Mit den Aktionen, die diesen Objekten zur Verbindungsverwaltung zugeordnet sind, können Host- und IBM AS/400-Datenbankverbindungen aufgelistet, erzwungen und überwacht werden.

Der Objektbaum unterhalb des Ordners **Gateway-Verbindungen** wird zur Verwaltung von Verbindungen zu Host- und IBM AS/400-Datenbanken verwendet, und nicht für Aufgaben der Datenbankverwaltung. Wenn Sie jedoch

eine Host- oder AS/400-Datenbank auf dem lokalen System hinzufügen, ändern oder entfernen müssen, sollten Sie die Funktion Client-Konfiguration - Unterstützung verwenden.

Funktionen in der Steuerzentrale

In der Steuerzentrale können Sie folgende Aufgaben erledigen:

- Datenbankobjekte verwalten. Sie können Datenbanken, Tabellenbereiche, Tabellen, Sichten, Indizes, Auslöser und Schemata erstellen, ändern und löschen. Außerdem können Sie die Benutzer verwalten.
- Daten verwalten. Sie können Daten laden, importieren, exportieren und reorganisieren und Statistikdaten sammeln.
- Jobs zeitlich steuern. Jobs können anstehende, aktive oder beendete Ausführungen von Prozeduren sein. Sie können für Jobs Startzeiten definieren.
- Eine vorbeugende Wartung durch die Sicherung und Wiederherstellung von Datenbanken oder Tabellenbereichen ausführen.
- Die Leistung überwachen und Fehler beheben.
- Daten replizieren.
- Exemplare und Datenbanken konfigurieren und optimieren.
- Datenbankverbindungen wie z. B. DB2 Connect-Server und -Subsysteme verwalten. Anwendungen verwalten.
- Abfragen analysieren, indem Sie mit Explain SQL die Zugriffspläne anzeigen.
- Schrift für die Anzeige von Menüs und Textelementen in der Steuerzentrale ändern. Sie können eine der angezeigten Schriftarten, Schriftgrößen und Farben verwenden. Die Steuerzentrale muss erneut gestartet werden, damit die Änderungen wirksam werden.
- Weitere Tools starten. Sie können z. B. die Satellitenverwaltungszentrale oder die Befehlszentrale starten.

Sie können alle Aktionen anzeigen, die Sie mit einem Objekt ausführen können. Wählen Sie dazu im Teilfenster für die Objektbaumstruktur oder im Inhaltsteilfenster das gewünschte Objekt aus und klicken Sie anschließend mit der rechten Maustaste. Daraufhin wird ein Kontextmenü angezeigt, in dem sich alle Funktionen befinden, die Sie mit dieser Objektart ausführen können: Wenn Sie z. B. den Tabellenordner auswählen, können Sie eine neue Tabelle erstellen und dabei gegebenenfalls einen Assistenten verwenden, die Leistung der Tabellen überwachen, die im Inhaltsteilfenster angezeigten Tabellen filtern usw. Die Funktionen, die Sie ausführen können, hängen vom ausgewählten Objekt ab.

Wenn Sie mit einem bestimmten Objekt weitere Funktionen ausführen möchten, klicken Sie im Inhaltsteilfenster mit der rechten Maustaste, während sich der Mauszeiger über dem gewünschten Objekt befindet. Wenn Sie z. B. im

Inhaltsteilfenster eine der Tabellen auswählen und mit der rechten Maustaste klicken, werden in einem Kontextmenü Funktionen angezeigt, die Sie für diese Tabelle verwenden können.

Erstellen neuer Objekte

Gehen Sie wie folgt vor, um neue Objekte zu erstellen:

1. Erweitern Sie den Datenbankordner. Die Objektarten werden als Ordnersymbole angezeigt.
2. Klicken Sie, während sich der Mauszeiger auf dem Ordnersymbol für ein Objekt befindet, mit der rechten Maustaste. Klicken Sie also z. B. mit der rechten Maustaste auf dem Symbol **Tabellen**. Das Kontextmenü wird angezeigt. Für einige Objekte haben Sie zwei Möglichkeiten, eine Funktion auszuführen. Die erste Möglichkeit ist die Verwendung eines Assistenten. Nicht für alle Funktionen, die Sie ausführen können, sind Assistenten verfügbar.
3. Wählen Sie **Erstellen** aus. Da ein Assistent zur Tabellenerstellung verfügbar ist, erhalten Sie zwei Optionen. Eine der Optionen ist die Tabellenerstellung mit dem Assistenten. Wenn Sie die Methode mit dem Assistenten auswählen, werden Sie zur Eingabe von Daten aufgefordert und erhalten Empfehlungen für die Auswahl. Der Assistent ist besonders zweckmäßig für Erstbenutzer oder Personen, die selten Datenbankobjekte erstellen.

Arbeiten mit vorhandenen Objekten

Wenn Sie ein Objekt, wie z. B. den Tabellenordner im Fenster mit dem Objektbaum, anklicken, werden im Inhaltsteilfenster alle bereits vorhandenen Tabellen angezeigt. Anschließend können Sie die zu bearbeitende Tabelle auswählen und mit der rechten Maustaste klicken, um eine beliebige Funktion aufzurufen, die Sie für diese Tabelle ausführen möchten.

Weitere Informationen zur Verwendung der Steuerzentrale finden Sie in der zugehörigen Online-Hilfefunktion, die Sie in der Steuerzentrale im Menü **Hilfe** oder mit der Taste F1 aufrufen können.

Suchen von Objekten (nur DB2 für OS/390)

Sie können eine Datenbank- oder ein Subsystemobjekt auf einfache Weise suchen, indem Sie das Notizbuch **Suchen** verwenden. Dies dient den folgenden Zwecken:

- Suche nach einem Objekt, ohne durch den Objektbaum in der Steuerzentrale navigieren zu müssen. Das Objekt kann sich in einer Datenbank, einem Subsystem oder einem Tabellenbereich befinden, und die Suche kann sich auf mehrere Datenbanken, Tabellen und unterstützende Objekte erstrecken.
- Suche nach Objekten (Tabellenbereichen, Tabellen und Indizes) in mehreren Datenbanken innerhalb eines Subsystems.

Zu Angabe der Suchkriterien können Sie die Seite **Suchen** im Notizbuch **Suchen** verwenden. Auf der Seite **Erweitert** des Notizbuchs **Suchen** können Sie die Suche weiter anpassen. Sie können den Text auf der Seite **Erweitert** bearbeiten und die Suchkriterien ändern oder neue Kriterien hinzufügen.

Gehen Sie wie folgt vor, um ein Objekt zu suchen, das in einer Datenbank oder einem DB2 für OS/390-Subsystem definiert ist:

1. Klicken Sie in der Steuerzentrale mit der rechten Maustaste auf einem Objekt. Wählen Sie im Kontextmenü die Option **Suchen** aus. Das Notizbuch **Suchen** wird geöffnet.
2. Wählen Sie im Feld **Objektart** die Art des gesuchten Datenbankobjekts aus. Die Liste der verfügbaren Zielobjekte hängt vom Objekt ab, bei dem Sie mit der Suche beginnen.
3. Geben Sie auf der Seite **Suchen** die Suchkriterien ein. Sie müssen mindestens ein Suchkriterium eingeben und dürfen Platzhalterzeichen verwenden. Die Zeichen werden in Großbuchstaben umgewandelt, es sei denn, Sie verwenden gültige Begrenzer vor und nach den Kleinbuchstaben oder erweiterten Zeichen.
4. Wählen Sie auf der Seite **Suchen** einen Radioknopf aus, um anzugeben, ob alle Bedingungen oder mindestens eine der Bedingungen erfüllt werden müssen, die in den Feldern auf der Seite **Suchen** ausgewählt worden sind.
5. Klicken Sie **OK** an, um die Suchkriterien zu verwenden. Das Ergebnis der Suche wird im Fenster **Suchergebnis** angezeigt. Das Format der ausgegebenen Tabelle hängt von der gesuchten Objektart ab.
6. Wenn Sie die Suche mit den gleichen oder mit anderen Kriterien wiederholen möchten, klicken Sie **Anwenden** an.
7. Zur Anzeige eines Kontextmenüs mit weiteren Funktionen wählen Sie eine Zeile im Fenster **Suchergebnis** aus und klicken diese Zeile mit Maustaste 2 an.

Satellitenverwaltungszentrale

Die Satellitenverwaltungszentrale besteht aus einer Reihe von Tools, die über die DB2-Steuerzentrale verfügbar sind. Mit diesen Tools können Gruppen von DB2-Servern von einem zentralen Punkt aus definiert und verwaltet werden. Jeder DB2-Server, der zu einer Gruppe gehört, gilt als Satellit. Die Verwaltung der Satelliten von einem zentralen Punkt aus bedeutet, dass DB2 für jeden Benutzer eines DB2-Satelliten unsichtbar ist und somit der Lernaufwand für die Datenbankverwaltung vermieden wird.

Sie können Gruppen verwenden, um DB2-Server mit gemeinsamen Merkmalen, wie z. B. die darauf ausgeführten Anwendungen oder die Datenbank-

konfiguration für die Anwendung, zu organisieren. Die DB2-Server sind hinsichtlich ihrer Datenbankkonfiguration, ihrer Verwendung und ihres Zwecks einander ähnlich.

Wenn Sie die DB2-Server gruppieren, können Sie Gruppen von DB2-Servern verwalten, ohne jeden DB2-Server einzeln verwalten zu müssen. Sollten Sie zusätzliche DB2-Server erwerben, die dieselbe Funktion erfüllen wie die DB2-Server in einer vorhandenen Gruppe, können Sie diese der Gruppe mit Hilfe der Satellitenverwaltungszentrale hinzufügen.

In der Satellitenverwaltungszentrale können Sie Gruppen, Satelliten, Anwendungsversionen, Stapel und Berechtigungsnachweise für Authentifizierungen erstellen. Außerdem können Sie Erfolgscodes definieren und weitere Funktionen ausführen, die der Verwaltung einer Satellitenumgebung dienen. Die Daten zur Satellitenumgebung werden in einer zentralen Datenbank, der so genannten Satellite Control Database, gespeichert. In dieser Datenbank wird unter anderem aufgezeichnet, welche Satelliten sich in der Umgebung befinden, zu welcher Gruppe die einzelnen Satelliten gehören und welche Version der Endbenutzeranwendung auf dem Satelliten ausgeführt wird. Diese Datenbank befindet sich auf einem DB2-Server, dem so genannten DB2-Steuerungs-Server.

Bevor die Funktionalität der Satellitenverwaltungszentrale aktiviert werden kann, müssen Sie in der Steuerzentrale eine Satellite Control Database (SATCTLDB) katalogisieren. Wenn diese aktiviert ist, können Sie die Satellitenverwaltungszentrale verwenden, um Satelliten, Gruppen und bei der Synchronisation der Anwendungsversion ausgeführte Stapel einzurichten und zu verwalten.

Damit die Datenbankkonfiguration definiert und verwaltet wird, stellt jeder Satellit eine Verbindung zur Satellite Control Database her, um die Stapel herunterzuladen, die der Version der Endbenutzeranwendung entsprechen. Der Satellit führt diese Stapel lokal aus und sendet anschließend die Ergebnisse zurück an die Satellite Control Database. Das Herunterladen der Stapel, ihre Ausführung und die anschließende Rücksendung von Ergebnissen der Stapelverarbeitung wird als Synchronisation bezeichnet. Ein Satellit wird synchronisiert, damit er mit anderen Satelliten konsistent bleibt, die zur selben Gruppe gehören und die dieselbe Version der Endbenutzeranwendung ausführen.

Befehlszentrale

Die Befehlszentrale können Sie in der Steuerzentrale aufrufen. Dazu müssen Sie in der Menüleiste das Symbol für die Befehlszentrale anklicken.

Die Befehlszentrale dient den folgenden Zwecken:

- Anzeige des Ausgabeergebnisses einer oder mehrerer SQL-Anweisungen und DB2-Befehle in einem Ergebnisfenster. Sie können in den Ergebnissen blättern und einen Bericht generieren.
- Erstellen und Speichern von Befehlsprozeduren für die Prozedurzentrale. Sie können die Befehlsprozedur editieren, um neue Prozeduren zu erstellen. In der Prozedurzentrale kann die Befehlsprozedur anschließend zeitlich terminiert werden, so dass sie zu einer beliebigen gewünschten Zeit als Job ausgeführt wird.
- Ausführung von SQL-Anweisungen, DB2-Befehlen und Betriebssystembefehlen. Wenn Sie DB2-Befehle von der Befehlszentrale aus ausführen, müssen Sie dem Befehl nicht **db2** voranstellen. Sie können Betriebssystembefehle einer beliebigen unterstützten prozedurbasierten Sprache eines Betriebssystems (z. B. REXX) ausführen, indem Sie diesen Ausrufezeichen (!) voranstellen. Wenn Sie die Befehle und Anweisungen mit Hilfe der Befehlszentrale ausführen, können Sie mehrere Befehle gleichzeitig absetzen, anstatt diese einzeln eingeben und ausführen zu müssen.
- Schneller Zugriff auf DB2-Verwaltungs-Tools, wie z. B. die Steuerzentrale, über die Hauptmenüleiste.
- Anzeige des Zugriffsplans und der Statistik zu einer SQL-Anweisung, bevor diese ausgeführt wird.

Prozedurzentrale

Sie können die Prozedurzentrale starten, indem Sie das entsprechende Symbol in der Menüleiste der Steuerzentrale auswählen. Die Prozedurzentrale ist ein Tool, mit dem Sie Prozeduren erstellen können. Dazu geben Sie mehrere Befehle und Anweisungen ein, die Sie zu einem beliebigen gewünschten Zeitpunkt starten lassen können. Sie können vorhandene Prozeduren oder in der Befehlszentrale gespeicherte Prozeduren importieren. Aus den gespeicherten Prozeduren können Sie Prozeduren auswählen und vorhandene Prozeduren editieren, um neue Prozeduren zu erstellen, und Sie können Prozeduren kopieren oder löschen.

Sie können eine Prozedur innerhalb oder außerhalb der Prozedurzentrale editieren und dazu Ihren eigenen Editor verwenden. Wenn Sie von der Prozedurzentrale aus eine Prozedur ausführen, hat dies den Vorteil, dass die Ergebnisse im Journal protokolliert werden.

Gehen Sie wie folgt vor, um in der Prozedurzentrale Betriebssystembefehle mit einer Prozedur auszuführen:

1. Wählen Sie **Prozedur** —> **Neu** aus. Das Fenster **Neue Befehlsprozedur** wird geöffnet.
2. Wählen Sie für **Prozedurart** den Radioknopf **Betriebssystembefehl** aus.

3. Geben Sie den Namen, die Beschreibung und das Arbeitsverzeichnis für die Prozedur ein.
4. Geben Sie die Befehle ein.
5. Klicken Sie **OK** an.

In der Prozedurzentrale können Sie Daten anzeigen, wie z. B. die Beschreibung und die Prozedurart. Dies gilt für alle Befehlsprozeduren, die dem System bekannt sind. Sie können darüber hinaus die folgenden Aufgaben ausführen:

- Erstellen einer Befehlsprozedur, die DB2- und Betriebssystembefehle enthält.
- Sofortiges Ausführen einer gespeicherten Befehlsprozedur.
- Festlegen der Prozedurausführung zu einem späteren Zeitpunkt oder in regelmäßigen Intervallen. Sie können z. B. eine Prozedur erstellen, die für mehrere Tabellen Statistikdaten sammelt. Anschließend können Sie den Job über Nacht ausführen lassen. Sie können Jobs zeitlich steuern, indem Sie sie unüberwacht in definierten Intervallen ausführen lassen. Dabei können Sie Stunden, Tage, Wochen, Monate, mehrere Zeitpunkte in einer Woche oder in einem Monat für die Ausführung des Jobs angeben. Jedesmal, wenn Sie den Ablauf einer Prozedur zeitlich terminieren oder eine Prozedur sofort ausführen, wird ein Job erstellt.
- Zugriff auf das Journal von der Menüleiste aus, um die Jobs anzuzeigen, die eine bestimmte Prozedur verwenden, und um den Status aller zeitlich terminierten Jobs zu sehen.
- Editieren einer gespeicherten Befehlsprozedur.

Verwenden einer vorhandenen Prozedur in der Prozedurzentrale

Gehen Sie wie folgt vor, um in der Prozedurzentrale vorhandene Prozeduren zu verwenden, die Sie nicht in der Prozedurzentrale erstellt haben:

1. Klicken Sie in der Menüleiste **Steuerzentrale** das Symbol **Prozedurzentrale** an. Die Prozedurzentrale wird geöffnet.
2. Wählen Sie **Prozedur** —> **Importieren** aus. Das Fenster **Datei-Browser** wird geöffnet.
3. Wählen Sie eine vorhandene Prozedurdatei aus, und klicken Sie **OK** an. Das Fenster **Neue Befehlsprozedur** wird geöffnet. Die Prozedur wird im unteren Teil des Fensters im Prozedureditor angezeigt. Geben Sie die Daten in die Felder **Exemplar**, **Prozedurname**, **Prozedurbeschreibung** und **Arbeitsverzeichnis** ein, und wählen Sie eine **Prozedurart** aus.
4. Klicken Sie **OK** an. Die Prozedur wird in der Prozedurzentrale erstellt.

Zeitliches Steuern der Ausführung einer gespeicherten Befehlsprozedur

Gehen Sie wie folgt vor, um eine Prozedur zeitlich zu steuern:

1. Klicken Sie in der Menüleiste der Steuerzentrale das Symbol **Prozedurzentrale** an. Die Prozedurzentrale wird geöffnet.

2. Klicken Sie, während sich der Mauszeiger auf der gewünschten Prozedur befindet, mit der rechten Maustaste und wählen Sie im Kontextmenü die Option **Zeitplan** aus. Das Fenster **Zeitplan** wird geöffnet.
3. Wählen Sie für den Job die Häufigkeit und die Aktion bei der Beendigung aus: z. B. eine Beendigungsnachricht oder eine weitere Befehlsprozedur, die aufgerufen werden soll.
4. Klicken Sie **OK** an. Dadurch wird ein anstehender Job gestartet, den Sie im Journal verfolgen können.

Journal

Sie können das Journal starten, indem Sie das entsprechende Symbol in der Menüleiste der Steuerzentrale auswählen. Mit dem Journal können Sie Jobs überwachen und Ergebnisse überprüfen. Außerdem können Sie im Journal das Wiederherstellungsprotokoll und DB2-Nachrichten anzeigen. Das Journal dient den folgenden Zwecken:

- Überwachen anstehender Jobs, aktiver Jobs und von Job-Protokollen.
- Überprüfen der Ergebnisse
- Anzeigen von Wiederherstellungsprotokoll und Alert-Nachrichten
- Anzeigen des Protokolls mit DB2-Nachrichten

Arbeiten mit Jobs

Zum Arbeiten mit Jobs können Sie das Journal verwenden. Gehen Sie wie folgt vor, um das Journal zu öffnen:

1. Klicken Sie in der Menüleiste der Prozedurzentrale das Symbol **Journal** an. Das Journal wird geöffnet.
2. Wenn Sie die Jobs, die später ausgeführt werden sollen, sehen möchten, klicken Sie den Druckknopf **Anstehende Jobs** an. Ihren Job können Sie in der Liste anstehender Jobs sehen. Außerdem werden alle Informationen zu den Jobs angezeigt. Sie können mit einem anstehenden Job bestimmte Aktionen ausführen, wie z. B. die Neuterminierung, die Anzeige von zugeordneten Prozeduren oder die sofortige Ausführung. Wenn eine gesicherte Prozedur geändert wird, übernehmen alle davon abhängigen Jobs die Änderung.

Außerdem können Sie im Journal die Jobs, die momentan ausgeführt werden, und die Jobprotokolle anzeigen.

Darüber hinaus befinden sich im Fenster **Journal** die folgenden Seiten:

- Die Seite **Wiederherstellung**. Auf dieser Seite wird das Wiederherstellungsprotokoll (Einzeldaten der Sicherungs-, Wiederherstellungs- und Ladeoperationen) angezeigt, und Sie können hier das Wiederherstellungsprotokoll wiederherstellen.
- Die Seite **Alerts**. Auf dieser Seite werden alle Alerts angezeigt.

- Die Seite **Nachrichten**. Auf dieser Seite werden alle Nachrichten angezeigt, die von DB2-Verwaltungs-Tools ausgegeben werden.

In der Online-Hilfefunktion zum Journal erhalten Sie detaillierte Anweisungen zur Arbeit mit Jobs und Protokollen.

Lizenzzentrale

Sie können die Lizenzzentrale verwenden, um den Status der Lizenzen und die Nutzungsdaten zu DB2-Produkten anzuzeigen, die auf Ihrem System installiert sind. Außerdem können Sie die Lizenzzentrale verwenden, um Ihr System für die ordnungsgemäße Lizenzüberwachung zu konfigurieren. Die Lizenzzentrale dient den folgenden Zwecken:

- Hinzufügen einer neuen Lizenz
- Erweitern einer Probelizenz für das Produkt auf eine permanente Lizenz.
- Anzeigen der Einzeldaten zu einer Lizenz

Wenn Sie die Einzeldaten zu den Lizenzen anzeigen, erhalten Sie die folgenden Informationen:

- Produktname
- Versionsnummer
- Verfallsdatum
- Registrierte Benutzer
- Gleichzeitig angemeldete Benutzer
- Anzahl der berechtigten Benutzer
- Gleichzeitig angemeldete Benutzer
- Durchsetzungsmaßnahme
- Anzahl der Prozessoren (für DB2 Universal Database Enterprise Edition und Enterprise – Extended Edition).

Alert-Zentrale

Das Tool Alert-Zentrale überwacht das System und warnt Sie vor möglichen Fehlern. Sie können die Alert-Zentrale so einstellen, dass sie automatisch geöffnet wird und alle überwachten Objekte anzeigt, die ihre Schwelle überschritten haben und sich aufgrund dessen im Alarm- oder Warnstatus befinden. Die Schwellenwerte können Sie mit Hilfe des Performance Monitor definieren. Der Performance Monitor wird in der Steuerzentrale aufgerufen. Die Farbe des Symbols zeigt die Wertigkeit der Warnung an. Eine Alarmanzeige wird durch ein rotes Symbol gekennzeichnet. Ein gelbes Symbol bedeutet eine Warnung.

"Client-Konfiguration - Unterstützung"

Die Client-Konfiguration - Unterstützung ist ein Tool, das Assistenten enthält, die den Benutzer beim Konfigurieren von Clients für die Kommunikation mit lokalen oder fernen DB2-Servern unterstützt. Das Tool kann allerdings auch dazu verwendet werden, DB2 Connect-Server auf einfache Weise zu konfigurieren.

Die Client-Konfiguration - Unterstützung ermöglicht das Verwalten einer Liste mit Datenbanken, zu denen Ihre Anwendungen eine Verbindung herstellen können. Sie katalogisiert Knoten und Datenbanken und gleicht Ihnen gegenüber die komplexen Vorgänge aus, die zur Durchführung dieser Aufgaben erforderlich sind.

Mit der Client-Konfiguration - Unterstützung können Sie die folgenden Aufgaben ausführen:

- Hinzufügen, Modifizieren und Löschen von Datenbankverbindungseinträgen.
- Testen der Verbindung zu einer ausgewählten Datenbank.
- Konfigurieren der Konfigurationsparameter für den Datenbankmanager.
- Konfigurieren von CLI/ODBC-Einstellungen.
- Binden von DB2-Dienstprogrammen und anderen Anwendungen an eine ausgewählte Datenbank.
- Importieren und Exportieren von Konfigurationsdaten. Hierdurch können Sie die vorhandene Konfiguration auf einer bereits konfigurierten Maschine zum Konfigurieren neuer Maschinen einsetzen.
- Ändern des Kennwortes für die Benutzer-ID, die zur Herstellung der Verbindung zu einer ausgewählten Datenbank verwendet wird.

Die Client-Konfiguration - Unterstützung stellt die folgenden Methoden zur Verfügung, um Sie beim Hinzufügen neuer Datenbankverbindungseinträge zu unterstützen:

- Verwenden eines Profils. Ein Profil kann von einer bereits konfigurierten Maschine exportiert und zum Konfigurieren einer neuen Maschine verwendet werden. Server-Profile können aus der Steuerzentrale, Client- oder Server-Profile aus der Client-Konfiguration - Unterstützung exportiert werden.
- Durchsuchen des Netzwerks. Die Client-Konfiguration - Unterstützung kann das Netzwerk nach DB2-Systemen durchsuchen, auf denen ein Verwaltungsserver ausgeführt wird. Das System stellt einen (gesteuerten) Discovery-Modus SEARCH und KNOWN zur Verfügung. Beim Discovery-Modus SEARCH gelten in Bezug auf die Netzwerkkonfiguration bestimmte Einschränkungen. (Normalerweise lassen Netzwerk-Router die Übertragung von Discovery-Suchanforderungen nicht zu.) Beim Discovery-Modus KNOWN sind nur bestimmte Informationen erforderlich, um das

gewünschte Server-System zu lokalisieren. Bereits auf einem Gateway definierte Host- oder AS/400-Systeme können ebenfalls gesucht werden.

- Manuelles Konfigurieren einer Datenbankverbindung. Sie müssen alle Informationen angeben, es wird jedoch ein Assistent gestartet, um diese Aufgabe zu vereinfachen.

Performance Monitor

Der Performance Monitor liefert Daten zum Status von DB2 Universal Database und zum Status der Daten, die von DB2 Universal Database gesteuert werden. Er ist ein Dienstprogramm mit grafischer Benutzerschnittstelle, das Sie an Ihre Datenbankumgebung anpassen können. Sie können damit Schwellen oder Bereiche definieren, mit denen Warnungen oder Alarmnachrichten ausgelöst werden, sobald die vom Performance Monitor ermittelten Werte nicht innerhalb der zulässigen Bereiche liegen.

Sie können DB2-Objekte wie Exemplare, Datenbanken, Tabellen, Tabellenbereiche und Verbindungen überwachen, indem Sie das entsprechende Objekt im Teilfenster der Objektbaumstruktur oder im Inhaltsteilfenster auswählen und mit der rechten Maustaste klicken. Dort können Sie die Überwachung der Aktivität starten.

Während ein Objekt überwacht wird, wird dessen Symbol grün, gelb oder rot angezeigt, um damit den Status des Monitors anzuzeigen. Die Farben stellen die Wertigkeit des Fehlers entsprechend den von Ihnen definierten Schwellenwerten dar. Die Farbe grün bedeutet, dass der Monitor ausgeführt wird und alles in Ordnung ist. Die Farbe gelb ist eine Warnung und bedeutet, dass das überwachte Objekt sich dem von Ihnen definierten Schwellenwerten nähert. Die Farbe rot ist ein Alarmsignal. Das überwachte Objekt hat den Schwellenwert erreicht. Sie können die vordefinierten Monitore verwenden, die mit DB2 geliefert werden, oder eigene Monitore erstellen.

Zum Anzeigen der vom Performance Monitor gesammelten Daten können Sie mit der rechten Maustaste auf dem Objekt klicken und im Kontextmenü die Option **Monitoraktivität anzeigen** auswählen.

Die Informationen im Performance Monitor dienen zu folgenden Zwecken:

- Erkennen von Leistungsproblemen
- Optimieren der Leistung von Datenbanken
- Analyse von Leistungstrends
- Analyse der Leistung von Datenbankanwendungen
- Vorbeugen gegen Fehler

Mit dem Performance Monitor können Sie Trends analysieren. Dazu wird eine visuelle Darstellung der Datenbankinformationen wie z. B. der Plattenaktivität, Nutzung von Pufferpools, des Vorablesumfangs, der Nutzung von Sperren und Satzblockungen in bestimmten Intervallen erstellt.

Sie können das Tool verwenden, wenn Sie einen aufgetretenen Fehler überwachen müssen oder die Leistung des Systems untersuchen möchten. Sie können zu einem bestimmten Zeitpunkt eine Momentaufnahme der Datenbankaktivität und der Leistungsdaten erstellen. Diese Momentaufnahmen werden später zum Vergleich herangezogen. Die einzelnen Punkte im Diagramm stellen Datenwerte dar. Anweisungen zum Erstellen von Momentaufnahmen finden Sie in „Überwachen der Leistung zu einem bestimmten Zeitpunkt“ auf Seite 32. Mit diesen Informationen können Sie mögliche Probleme einfacher erkennen und analysieren oder Ausnahmebedingungen feststellen, die auf den von Ihnen definierten Schwellenwerten basieren. Wenn Sie die Leistung des Datenbankmanagers und dessen Datenbankanwendungen zu einem bestimmten Einzelzeitpunkt feststellen und die Trends über eine bestimmte Zeitdauer untersuchen möchten, können Sie das Leistungsanalyse-Tool verwenden. Verwenden Sie das Tool auch, um eine visuelle Übersicht darüber zu erhalten, welche Elemente sich im Alarmstatus befinden. Dadurch können Sie einfacher erkennen, welche Parameter optimiert werden müssen. Anschließend können Sie die für das betreffende Element definierten Parameter genau untersuchen und zur Leistungsverbesserung ändern.

Ereignismonitor

Im Gegensatz zu einer Momentaufnahme zu einem bestimmten Zeitpunkt, können Sie mit einem Ereignismonitor über einen Zeitraum hinweg Daten zu Datenbankaktivitäten sammeln. Diese gesammelten Daten ergeben eine gute Zusammenfassung der Aktivität in Bezug auf ein bestimmtes Datenbankereignis, z. B. über eine Datenbankverbindung oder eine SQL-Anweisung. Der Ereignismonitor ist ein Tool, mit dem der Status der Datenbank zurzeit des Auftretens bestimmter Ereignisse protokolliert wird. Außerdem können Sie damit die Aktivität in der Datenbank verfolgen. Der Datensätze des Ereignismonitors werden nach der Aufzeichnung der Daten gespeichert und anschließend analysiert. Sie können den Ereignismonitor verwenden, wenn Sie z. B. die Dauer einer Transaktion oder die CPU-Auslastung durch eine SQL-Anweisung feststellen möchten. Anschließend können Sie den Event Analyzer verwenden, um die vom Ereignismonitor aufgezeichneten Daten zu lesen.

Für jedes Ereignis einer Datenbankverbindung wird ein Datensatz eingetragen. Für jede Anweisung, die in dieser Verbindung ausgeführt wird, wird ein Anweisungsdatensatz aufgezeichnet. Jeder Datensatz mit einem Verbindungsereignis ist einer Zeile im Fenster **Connections View** des Event Analyzer zugeordnet. In diesem Fenster werden die folgenden Informationen zu den einzelnen Anwendungen angezeigt, die während des überwachten Zeitraums verbunden waren:

- Anwendungsname
- Ausführungs-ID
- Verbindungsdauer
- Gesamt-CPU-Zeit
- Wartezeit für Sperre
- Gesamtsortierzeit
- Gegenseitige Sperren
- DISCONNECT-Zeit
- Application ID (Anwendungs-ID)

Jede Anweisung mit einem Verbindungsereignis ist einer Zeile im Fenster **Statements View** des Event Analyzer zugeordnet.

Verwenden der Monitor-Tools

Der Performance Monitor und der Event Analyzer bieten die folgenden Vorteile:

- Umfassende, flexible Datensammlung. Über 200 Leistungsvariablen werden unterstützt: Pufferpool und E/A, Sperre und gegenseitiges Sperren, Sortieren, Kommunikation, Agenten und Protokolldaten. Es werden Daten angezeigt zu Datenbankmanagern, Datenbanken, Tabellenbereichen, Tabellen, Pufferpools, Transaktionen und SQL-Anweisungen.
- Benutzerfreundlichkeit, intuitiv verständliche Anzeige. Die Daten können in Echtzeit und in leicht verständlichen Diagrammen oder Textanzeigen angezeigt werden, die zweckmäßig in logische Gruppen gegliedert sind. Sie erhalten Einzeldaten und Zusammenfassungen und können auf genauere Informationen zugreifen.
- Solide Alert-Funktionen. Zur Leistungsmessung können Sie Ausnahmehinweise definieren, indem Sie Schwellenwerte angeben. Die Schwellenwerte werden verwendet, um visuell zu erkennen, wann eine Leistungsmessung den Schwellenwert erreicht oder überschreitet. Dazu wird die Messung im Leistungsdiagramm in einer bestimmten Zone eingezeichnet. Für das Erreichen des Schwellenwerts können Sie angeben, dass beliebige oder alle der folgenden Aktionen ausgeführt werden sollen:
 - Mitteilung über die Alert-Zentrale
 - Empfangen eines akustischen Signals
 - Ausführen eines Programms
 - Anzeigen einer Nachricht

Sie können auch angeben, dass kein Hinweis erfolgen soll.

In Abb. 2 auf Seite 31 ist dargestellt, wie die Monitore zusammenarbeiten.

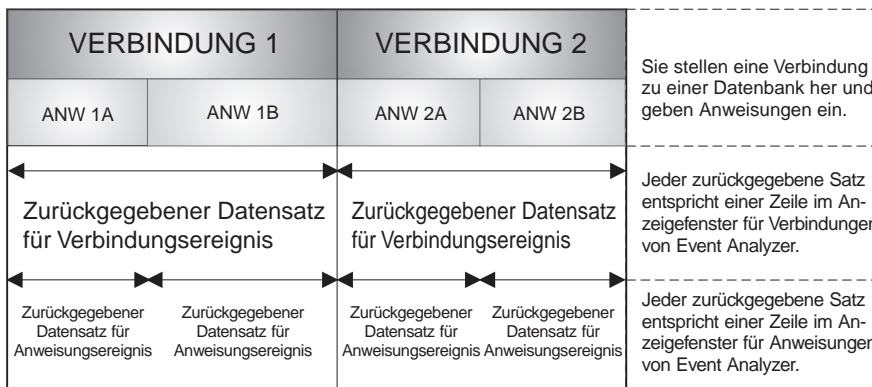
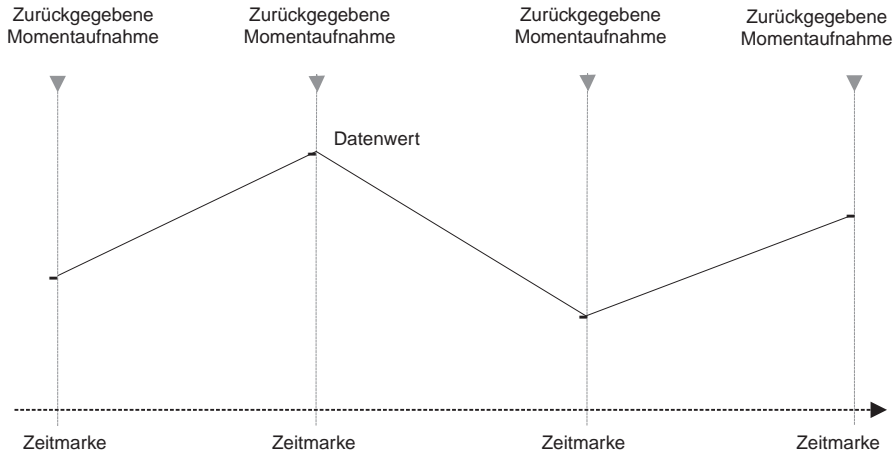


Abbildung 2. Vergleich: Momentaufnahme und Ereignismonitor. (Ereignismonitor, Event Analyzer)

Überlegungen zur Monitoraufzeichnung und zur Optimierung einer Datenbank

Bevor Sie mit der Monitoraufzeichnung und der Datenbankoptimierung beginnen, sollten Sie Folgendes beachten:

- Definieren Sie Ihre Ziele. Sie möchten möglicherweise z. B. feststellen, wie von den Anwendungen zu einem bestimmten Zeitpunkt die Ressourcen auf Exemplarebene genutzt werden, damit Sie z. B. überprüfen können, ob der gemeinsame Zugriff auf die Datenbank sich nach dem Start einer bestimmten Anwendung verringert. Oder Sie möchten untersuchen, welche Ereignisse auf Exemplarebene auftreten, wenn eine Anwendung ausgeführt wird, um zu ermitteln, ob infolge der Ausführung einer bestimmten Anwendung eine geringe Gesamtleistung verursacht wird.
- Bestimmen Sie, welche Informationen Sie analysieren. Sie möchten möglicherweise z. B. feststellen, ob die Engpässe an der Hardware liegen. Dazu erstellen Sie Momentaufnahmen, um die Aktivität der Datenbank-

verbindung, der Tabellenbereiche, der Pufferpools oder der E/A zu überwachen. Wenn Sie feststellen möchten, ob die Engpässe von der Umgebung verursacht wurden, können Sie mit dem Event Analyzer Folgendes überwachen:

- Zu hohe Anzahl der Datenbank-Tasks, die für die Spitzenzeiten terminiert werden.
- Hohe Anzahl von Benutzerverbindungen.
- Schlecht optimierte Datenbankpartitionierung (Hardwarelastausgleich).
- Verwendung des Servers für mehr als einen Datenbankserver.

Es ergeben sich z. B. die folgenden sichtbaren Auswirkungen:

- Die Abfragen/Antwortzeiten sind langsam.
- Terminierte Tasks werden nicht rechtzeitig beendet.
- Anwendungen überschreiten das Zeitlimit.
- Entscheiden Sie, ob Sie vordefinierte Monitore verwenden, die in DB2 verfügbar sind, oder ob Sie Ihre eigenen Monitore erstellen.

Im nächsten Abschnitt wird beschrieben, wie Sie Momentaufnahmen erstellen und wie Sie mit der Alert-Zentrale beliebige Leistungsprobleme verfolgen können.

Überwachen der Leistung zu einem bestimmten Zeitpunkt

Wenn Sie komplexe Daten sammeln und diese zur Erkennung möglicher Probleme analysieren möchten, können Sie den Performance Monitor verwenden, um vom System Momentaufnahmen zu erstellen und die Änderung der Leistungsdaten über eine bestimmte Zeit zu beobachten.

Dieses Tool dient den folgenden Zwecken:

- Darstellung der Leistungsdaten mit einem Diagramm
- Festlegung von Aufzeichnungshäufigkeit für die Momentaufnahmen der Leistung
- Anzeige der Ergebnisse von Leistungsberechnungen
- Definition von Schwellenwerten und zugeordneten Aktionen
- Generierung und Speicherung von Alerts
- Anzeige von Ergebnistexten (z. B. für alle Datenbanken)

Die folgenden Arten von Daten werden aufgezeichnet:

- Informationen zu einer lange andauernden Aktivität (wie z. B. der Datenbankaktivität, wenn eine Anwendung zu viel Zeit bis zur Beendigung benötigt).
- Zähler, die Informationen zum aktuellen Aktivitätsgrad aufzeichnen (wie z. B. die Anzahl geöffneter Cursor für eine Datenbank).

- Kumulative Informationen zur Datenbankaktivität, wie z. B. die Höchstzahl hergestellter Verbindungen, während ein Datenbanke exemplar aktiv ist, oder die Gesamtzahl von SQL-Anweisungen, die für eine bestimmte Datenbank ausgeführt wird.

Das Erstellen von Momentaufnahmen in vordefinierten Intervallen ergibt ein Bild des aktuellen Aktivitätsstatus innerhalb des Datenbankmanagers und seiner Anwendungen. Diese aufgezeichneten Informationen können zu folgenden Zwecken verwendet werden:

- Erkennen von Leistungsproblemen
- Analyse von Leistungstrends
- Optimieren des Datenbankmanagers und der Konfigurationsparameter der Datenbank
- Analyse der Leistung von Datenbankanwendungen

Leistungsdaten sind für die folgenden Datenbankobjekte verfügbar:

- Exemplare
- Datenbanken
- Tabellen
- Tabellenbereiche
- Datenbankverbindungen

Zu allen Objekten kann eine Reihe verschiedener Leistungsvariablen überwacht werden. In der Hilfe zu den Leistungsvariablen, die im Menü **Hilfe** eines beliebigen Fensters eines Snapshot Monitor verfügbar ist, sind alle Leistungsvariablen beschrieben. Diese Variablen werden in Kategorien eingeteilt. Es gibt die folgenden Kategorien:

- Exemplar: Agenten, Verbindungen, Sortieren
- Datenbank: Sperre und gegenseitiges Sperren, Pufferpool und E/A, Verbindungen, Sortieren, SQL-Anweisungen
- Tabelle: Tabelle
- Tabellenbereich: Pufferpool und E/A
- Datenbankverbindungen: Pufferpool und E/A, Sperre und gegenseitiges Sperren, Sortieren, SQL-Anweisungen

Genauere Informationen zum Generieren von Momentaufnahmen finden Sie in der Online-Hilfefunktion.

Vordefinierte Monitore

Der DB2 Performance Monitor enthält mehrere vordefinierte Monitore, die Sie ohne Anpassung verwenden können oder die Sie kopieren und an Ihre Anforderungen anpassen können. Die vordefinierten Monitore bieten eine umfassende Menge von Leistungsberechnungen.

Den Namen, die Gleichung oder den Beschreibungstext der von IBM gelieferten Performance Monitors können Sie nicht ändern. Sie können jedoch die Schwellenwerte und die Alert-Aktionen anpassen. Anhand der vordefinierten Monitore können Sie die Leistungsüberwachung erlernen und eigene Monitore erstellen, indem Sie einen vordefinierten Monitor kopieren und Leistungsvariablen hinzufügen oder löschen.

Zu DB2 werden die folgenden vordefinierten Monitore mitgeliefert:

- **Capacity Monitor.** Mit diesem Monitor können Sie Informationen zur Kapazität des Systems erhalten. Sie können diesen Monitor regelmäßig überprüfen, um die Gesamtnutzung des Systems über einen gewissen Zeitraum festzustellen.
- **Sort Monitor.** Mit diesem Monitor können Sie den Sortierspeicher und die Schwellenwerte für den Sortierspeicher richtig einstellen. Sie sollten diesen Monitor ausführen, wenn Sie das System zum ersten Mal starten, wenn Spitzenaktivität herrscht oder wenn sich Anwendungen ändern.
- **Locking Monitor.** Mit diesem Monitor können Sie bestimmen, wie oft im System Sperren auftreten und ob die Parameter der Sperrenliste richtig gesetzt sind.
- **Cache Tuning Monitor.** Mit diesem Monitor können Sie die Cache-Nutzung optimieren. Wenn Sie diese Werte in Spitzenzeiten überwachen, können Sie feststellen, ob Sie den Cache vergrößern müssen.
- **Buffer Pool Monitor.** Diesen Monitor können Sie für kleine Tabellen verwenden, um zu bestimmen, ob dafür separate Pufferpools erforderlich sind.
- **Deadlocks Monitor.** Mit diesem Monitor können Sie bestimmen, ob die Anwendungen sich gegenseitig sperren.
- **Fast Communication Manager (FCM) Tuning Monitor.** Mit diesem Monitor können Sie den Prozentsatz des Speichers anzeigen, der für die Übertragung von Daten zwischen den Knoten belegt wird.
- **Prefetchers Tuning Monitor.** Mit diesem Monitor können Sie bestimmen, ob für das System genügend Vorablesefunktionen definiert sind.
- **Disk Performance Monitor.** Mit diesem Monitor können Sie die Ein- und Ausgabe überwachen. Dieser Monitor enthält Leistungsvariablen, die schwerpunktmäßig für die Optimierung der Plattenleistung auf Datenbank- und Tabellenbereichsebene gedacht sind.
- **Global Memory Monitor.** Mit diesem Monitor können Sie die Speicherbelegung von Anwendungen überwachen.
- **Long Running Query Monitor.** Mit diesem Monitor können Sie auf einfachere Weise bestimmen, warum eine Abfrage lange dauert.
- **Gateway Connections Monitor.** Mit diesem Monitor können Sie die DB2 Connect-Serververbindungen überwachen.

Beispiele zur Verwendung eines vordefinierten Monitors finden Sie in der Online-Hilfefunktion zur Leistungsüberwachung.

Wenn Sie eine Liste verfügbarer Monitore anzeigen möchten, klicken Sie in der Steuerzentrale, während sich der Mauszeiger auf dem Ordner für **Systeme** befindet, mit der rechten Maustaste, und wählen im Kontextmenü die Option **List Monitors** aus. Das Fenster **List Monitors** wird geöffnet. Darin werden die Monitore aufgeführt, die auf dem JDBC-Server gespeichert sind, mit dem Sie momentan verbunden sind. Sie erhalten zu jedem Monitor den Namen, eine Beschreibung, den Status und die Information, ob der Monitor der Standardmonitor ist und wer den Monitor erstellt hat. Der „Status der Monitore“ gilt für den Status auf dem lokalen System, und nicht auf dem JDBC-Server. Unter „Standardwert für“ die Ebene wird der Standardmonitor für die Exemplar-, Datenbank-, Tabellen-, Tabellenbereichs- oder Verbindungsebene angezeigt. Für vordefinierte Monitore enthält die Spalte „Created by“ den Eintrag **NUL-LID**. Die rechte Seite des Fensters enthält Druckknöpfe, mit denen Sie mit den Monitoren verschiedene Tasks ausführen können. Weitere Informationen zum JDBC-Server finden Sie in „Ausführen der Steuerzentrale als Java-Applet“ auf Seite 54.

Sie können wählen, welcher Monitor für ein Objekt als Standardmonitor gestartet wird.

Nachdem Sie einen Performance Monitor gestartet haben, können Sie in der Menüleiste den Knopf für die Alert-Zentrale anklicken. Dadurch können Sie den Status beliebiger Objekte anzeigen, die Sie überwachen und die sich im Alert-Status befinden, da einer der Schwellenwerte erreicht worden ist. Die Objekte werden nur so lange angezeigt, wie der Schwellenwert überschritten wird.

Wenn Sie die überwachten Objekte genau untersuchen möchten, können Sie die Alert-Zentrale oder auf der Zusammenfassungsseite das Fenster **Show Monitor** geöffnet lassen und nach roten oder gelben Einträgen suchen. Außerdem können Sie die Einstellungen der Steuerzentrale so ändern, dass die Alert-Zentrale nach dem Hinzufügen einer neuen Warnung oder Alarmnachricht automatisch geöffnet wird. Darüber hinaus können Sie in der Alert-Zentrale die Alerts zeitweilig zurückstellen, während die Überwachung fortgesetzt wird.

Erforderliche Aktion nach der Anzeige eines Objekts in der Alert-Zentrale

Sie können die Alert-Zentrale so einstellen, dass sie automatisch geöffnet wird und alle überwachten Objekte anzeigt, die sich im Alarm- oder Warnstatus befinden (deren Schwellenwerte also überschritten worden sind). Die entsprechende Standardeinstellung können Sie im Fenster für **Tools - Einstellungen** ändern.

Wenn Sie in der Alert-Zentrale ein Objekt sehen, können Sie mit der rechten Maustaste auf diesem Objekt klicken und die Option **Performance Monitor** → **Show Monitor** auswählen, um die Einzeldaten der Leistung für das betreffende Datenbankobjekt anzuzeigen.

Analysieren eines Ereignisses über einen bestimmten Zeitraum

Der Event Analyzer stellt ein weiteres DB2-Leistungsanalyse-Tool dar. Sie können dieses Tool verwenden, um Diagnoseinformationen zu einem aufgetretenen Ereignis zu erhalten. Den Event Analyzer verwenden Sie in Kombination mit einem Ereignismonitor. Wenn Sie z. B. Datenbankaktivitäten, wie Verbindungen, Transaktionen, Anweisungen und gegenseitige Sperren, während die Datenbank aktiv ist, verfolgen möchten, können Sie einen Ereignismonitor verwenden. Mit einem Ereignismonitor können auch kumulative Leistungsdaten aufgezeichnet werden, die beim Trennen der Verbindung zwischen einer Anwendung und der Datenbank protokolliert werden. Nachdem der Ereignismonitor die Ereignismonitordatei erstellt hat, können Sie mit Hilfe des Event Analyzer die Leistungsdaten betrachten.

Die Ereignismonitor-Tools können Sie zu folgenden Zwecken verwenden:

- Erstellen von Ereignismonitoren, um die Arten von Datenbankereignissen zu überwachen, die für Sie relevant sind.
- Aktivieren eines Ereignismonitors, um das Sammeln von Ereignisdaten zu starten. Die Daten werden in einer Datei gespeichert.
- Stoppen der Sammlung von Ereignisdaten durch den Ereignismonitor.
- Anzeigen des Ergebnistextes in der Art einer Ablaufverfolgung (Trace), der vom Ereignismonitor generiert wurde.
- Entfernen eines Ereignismonitors, wenn Sie diesen nicht mehr benötigen. Sie können auch die Trace-Dateien bereinigen.
- Anzeigen der Liste von Ereignismonitoren, die der Datenbank zugeordnet sind.
- Anzeigen der Definition eines Ereignismonitors.

Mit dem Event Analyzer können Sie für die folgenden Ereignisarten die von einem Ereignismonitor generierten Daten anzeigen:

- Aktivität der Datenbankverbindung (Periode zwischen dem Herstellen und Trennen einer Verbindung)
- Transaktionen (Arbeitseinheiten)
- Ausführung von SQL-Anweisungen
- Erkennung des gegenseitigen Sperrens

Event Analyzer

Sie können für die folgenden Ereignisarten einen Ereignismonitor erstellen und anschließend den Event Analyzer verwenden, um die gesammelten Daten

anzuzeigen. Die Daten zu folgenden Faktoren sollten Sie allerdings mit dem Befehl `db2evmon` (siehe hierzu die Handbücher *Command Reference* und *System Monitor Guide and Reference*) anzeigen:

- Gegenseitige Sperren
- Datenbankaktivität
- Tabellenbereichsaktivität
- Tabellenaktivität
- Anweisungsaktivität

Wenn Sie mit einem Ereignismonitor und mit dem Event Analyzer Ereignisdaten analysieren möchten, befolgen Sie die unten beschriebenen Schritte. Hier ist lediglich ein Beispiel dargestellt, wie Sie einen Ereignismonitor für Verbindungs- und Anweisungsereignisse erstellen können. Gehen Sie wie folgt vor, um einen Ereignismonitor zu erstellen:

1. Geben Sie in die Befehlszeile der Befehlszentrale den Befehl **db2emcrt** ein. Das Fenster **Ereignismonitor** wird geöffnet.
2. Klicken Sie **Ereignismonitor** an, und wählen Sie im Menü die Option **Erstellen** aus. Das Fenster **Create Event Monitor** wird geöffnet.
3. Geben Sie für den Ereignismonitor, den Sie erstellen, einen Namen in das Feld ein. Dieser neue Ereignismonitor darf nicht denselben Namen tragen wie ein vorhandener Ereignismonitor. Im Namen sind Leerstellen unzulässig.
4. (Nur für DB2 Universal Database Enterprise - Extended Edition:) Wählen Sie einen Knoten, an dem sich die Dateien des Ereignismonitors befinden sollen, in der verdeckten Liste „On Node“ aus.
5. (Nur für DB2 Universal Database Enterprise Enterprise - Extended Edition:) Wählen Sie einen Bereich (Scope) für den Ereignismonitor aus. Der Standardwert für den Bereich lautet **Global**.
6. Wählen Sie ein Markierungsfeld oder mehrere aus, um die Ereignisart anzugeben, die Sie überwachen möchten. Beachten Sie, dass die Ereignisart **Deadlocks** der Standardwert ist.
7. Geben Sie an, wann dieser Monitor gestartet werden soll. Beachten Sie, dass „Start now“ der Standardwert ist.
8. Definieren Sie für Verbindungen, Anweisungen oder Transaktionen eine oder mehrere Bedingungen, mit denen die Überwachung dieser Ebenen gesteuert werden soll.
9. Geben Sie einen Pfad (Verzeichnisname) an, in den der Monitor die Ereignisdatendateien schreiben soll.
10. Klicken Sie **Options** an, um das Fenster mit den Optionen **Specifying Event Monitor File** zu öffnen. Mit diesen Optionen können Sie bestimmen, wie die Monitorausgabe bearbeitet wird. Dies kann sich auf die Leistung des Ereignismonitors auswirken.

11. Klicken Sie **OK** an, um den Monitor zu erstellen, oder klicken Sie **Abbruch** an, um das Fenster zu verlassen, ohne einen Monitor zu erstellen.
12. Schalten Sie die Ereignisüberwachung aus, indem Sie mit der rechten Maustaste auf einem Ereignismonitor klicken und im Kontextmenü die Option **Stop Event Monitoring** auswählen.

Dadurch wird erzwungen, dass der Ereignismonitor die Trace-Datei ausgibt. Wenn der Monitor ausgeschaltet ist, werden nur Daten auf die Platte geschrieben, wenn der Puffer voll ist oder alle Verbindungen beendet sind. Im Fenster **Event Monitors** können Sie die ermittelten Ereignisdaten anzeigen, indem Sie mit der rechten Maustaste auf dem von Ihnen erstellten Ereignismonitor klicken und im Kontextmenü die Option **View Event Monitor Files** auswählen. Das Fenster **Monitored Periods View** wird geöffnet.

Gehen Sie wie folgt vor, um im Event Analyzer auf Ereignisdaten zuzugreifen:

1. Geben Sie in die Befehlszeile der Befehlszentrale den Befehl **db2eva** ein, um den Event Analyzer zu starten. Das Fenster **Event Analyzer** wird geöffnet.
2. Geben Sie in das Feld für den Pfad den Namen des Verzeichnisses ein, in dem die Datendateien gespeichert sind. Wenn die Dateien inzwischen nicht verschoben worden sind, entspricht dies dem Verzeichnis, das bei der Erstellung des Ereignismonitors angegeben wurde. Falls die Dateien verschoben wurden, geben Sie das entsprechende Verzeichnis an. Sie können ... anklicken, um die vorhandenen Verzeichnisse aufzulisten.

Anmerkung: Wenn die Datendateien auf einer fernen Maschine gespeichert sind, müssen Sie die Dateien per FTP auf Ihre lokale Maschine herunterladen, um sie anzeigen zu können. Je nach Dateigröße kann diese Übertragung einige Zeit dauern. Die Dateien können in ein beliebiges lokales Verzeichnis übertragen werden. Es ist nicht erforderlich, denselben Pfad auszuwählen, der bei der Erstellung verwendet wurde.

3. Klicken Sie **OK** an, um auf die Datendateien in diesem Verzeichnis zuzugreifen, oder klicken Sie **Abbruch** an, um das Dialogfenster zu schließen. Das Fenster **Monitored Periods View** wird geöffnet.
4. Klicken Sie mit der rechten Maustaste auf einer überwachten Periode, und wählen Sie im Kontextmenü die Option **Open as —> Connections** aus. Das Fenster **Connections View** wird geöffnet. Darin wird eine Liste der Verbindungen angezeigt, die während der Ereignisüberwachungssitzung hergestellt worden sind. (Möglichweise sind mehrere Verbindungen aufgeführt. Die für Sie relevante Verbindung steht in der Liste möglicherweise nicht an erster Stelle.)

5. Klicken Sie mit der rechten Maustaste auf einer Verbindung und wählen Sie dann im Kontextmenü **Open as** → **Statements** aus. Das Fenster **SQL Statements View** wird geöffnet. Darin werden alle Anweisungen zur ausgewählten Verbindung angezeigt. Für die einzelnen Anweisungen sind in den Spalten die folgenden Informationen verfügbar:

- **Operation**
- **Paketname**
- **Ersteller**
- **Startzeit**
- **Antwortzeit**
- **Gesamt-CPU-Zeit**
- **Text**

Genaue Anweisungen zum Erstellen von Ereignismonitoren und zur Anzeige der ermittelten Ereignisdaten finden Sie in der Online-Hilfefunktion für den Ereignismonitor und den Event Analyzer.

Analysieren von SQL-Anweisungen

Sie können den Zugriffsplan für mit EXPLAIN bearbeitete SQL-Anweisungen als Diagramm anzeigen und diese Informationen zur Leistungsoptimierung der SQL-Abfragen verwenden.

Im Diagramm des Zugriffsplans sind folgende Einzeldaten enthalten:

- Tabellen (und deren zugehörige Spalten) und Indizes
- Operatoren (z. B. Tabellensuchen, Sortierungen und Verknüpfungen)
- Tabellenbereiche und Funktionen

Vor der Version 6 wurde das Tool **Visual Explain** verwendet, um Zugriffspläne anzuzeigen. Bei der aktuellen Version ist Visual Explain kein separates Tool mehr. Diese *Funktion* ist jedoch in den Kontextmenüs verschiedener Datenbankobjekte innerhalb der Steuerzentrale und in der Befehlszentrale verfügbar. In diesem Abschnitt wird diese Funktion als *Visual Explain* bezeichnet.

Visual Explain dient den folgenden Zwecken:

- Anzeigen der Statistiken, die zurzeit der Optimierung verwendet wurden. Sie können als eine Maßnahme zur Bestimmung, ob ein erneutes Binden des Pakets die Leistung verbessern könnte, diese Statistiken mit den aktuellen Katalogstatistiken vergleichen.
- Feststellen, ob ein Index zum Zugriff auf eine Tabelle verwendet wurde oder nicht. Wenn kein Index verwendet wurde, kann die Funktion Visual Explain Ihnen hilfreiche Hinweise geben, für welche Spalten ein Index möglicherweise von Vorteil wäre.

- Ermitteln der Auswirkungen verschiedener Optimierungstechniken, indem Sie die „Vorher-Nachher-Versionen“ des Zugriffsplandiagramms für eine Abfrage miteinander vergleichen.
- Anzeigen von Informationen zu jeder Operation im Zugriffsplan, einschließlich der Informationen zum Gesamtaufwand und der Anzahl von Zeilen, die abgerufen werden (Kardinalität).

Optimieren einer Abfrage

Sie können die Funktion Visual Explain zur Analyse und Optimierung von SQL-Anweisungen verwenden. Es wird eine grafische Anzeige des Zugriffsplans für mit EXPLAIN bearbeitete SQL-Anweisungen ausgegeben. Tabellen und Indizes sowie jede an ihnen ausgeführte Operation werden als Knoten und der Datenfluss als Verbindungslinien zwischen den Knoten dargestellt. Anhand der Informationen im Diagramm können Sie eine Optimierungsmethode für die SQL-Abfragen finden.

Die Funktion Visual Explain zeichnet Daten zur Art der Kompilierung von SQL-Anweisungen auf. Diese Informationen geben Aufschluss über den Plan und die potenzielle Ausführungsleistung von SQL-Anweisungen.

Diese Informationen können Sie zu folgenden Zwecken verwenden:

- Konzeption von Anwendungsprogrammen
- Entwerfen von Datenbanken
- Verständnis der Verknüpfung zweier Tabellen: verwendete Verknüpfungsmethode, Verknüpfungsreihenfolge der Tabellen, Sortierungen und Arten des Sortierens.
- Bestimmen von Methoden zur Optimierung von SQL-Anweisungen (z. B. durch die Erstellung eines Index).
- Anzeigen der Statistiken, die zurzeit der Optimierung verwendet wurden. Sie können anschließend diese Statistiken mit der aktuellen Katalogstatistik vergleichen, um einfacher bestimmen zu können, ob das erneute Binden des Pakets die Leistung erhöht. Damit können Sie außerdem einfacher bestimmen, ob mit Hilfe des Sammelns von Statistikdaten die Leistung erhöht werden kann.
- Feststellen, ob ein Index zum Zugriff auf eine Tabelle verwendet wurde oder nicht. Wenn kein Index verwendet wurde, kann die Funktion Visual Explain Ihnen hilfreiche Hinweise geben, welche Spalten dem Index hinzugefügt werden können, um die Leistung der Abfrage zu optimieren.
- Ermitteln der Auswirkungen verschiedener Leistungsoptimierungstechniken, indem Sie das Zugriffsplandiagramms für eine Abfrage vorher und nachher vergleichen.
- Anzeigen von Informationen zu jeder Operation im Zugriffsplan, einschließlich der Informationen zum Gesamtaufwand und der Anzahl von Zeilen, die abgerufen werden.

Nach der Verwendung der Funktion Visual Explain zur Erläuterung des Zugriffsplans für eine mit EXPLAIN bearbeitete SQL-Anweisung werden Sie möglicherweise feststellen, dass ein Index die Leistung dieser Abfrage verbessern könnte. Mit dem Assistent: Index erstellen können Sie empfohlene Indizes für die Abfrage abrufen. Alternativ hierzu können Sie auch den EXPLAIN-Modus RECOMMENDED_INDEXES verwenden. Weitere Informationen zum Assistenten: Index erstellen enthält die Funktion Information - Unterstützung, die über die Steuerzentrale aufgerufen werden kann.

Weitere Informationen zum EXPLAIN-Modus RECOMMENDED_INDEXES finden Sie im Handbuch *Systemverwaltung: Optimierung*.

Analysieren einer einfachen dynamischen SQL-Anweisung

In diesem Abschnitt erhalten Sie ein einfaches Einstiegsbeispiel, wie Sie eine dynamische SQL-Abfrage analysieren können.

1. Klicken Sie in der Steuerzentrale mit der rechten Maustaste auf der Beispieldatenbank, und wählen Sie im Kontextmenü die Option **Explain SQL** aus. Das Fenster **Explain SQL Statement** wird geöffnet.
2. Geben Sie in das Feld **SQL text** die folgende SQL-Anweisung ein:

```
select * from staff order by name
```
3. Klicken Sie **OK** an. Das Fenster **Access Plan Graph** wird geöffnet. Im Diagramm ist der Pfad dargestellt, den das Optimierungsprogramm als am effizientesten ermittelt hat, um die Ergebnisse für die betreffende Abfrage zu liefern.
4. Wahlfrei: Wählen Sie einen beliebigen Knoten durch Doppelklicken aus (z. B. den Operatorknoten RETURN). Das Fenster **Operator Details** wird geöffnet, in dem die Einzeldaten zum betreffenden Operator angezeigt werden.

Die mit EXPLAIN bearbeitete SQL-Anweisung wird automatisch gesichert. Gehen Sie wie folgt vor, um sie später anzuzeigen:

1. Klicken Sie in der Steuerzentrale mit der rechten Maustaste auf der Beispieldatenbank, und wählen Sie im Kontextmenü die Option **Show Explained Statements History** aus. Das Fenster **Explained Statements History** wird geöffnet.
2. Suchen Sie den gewünschten Eintrag. In der Spalte **SQL text** wird die zuvor mit EXPLAIN bearbeitete SQL-Anweisung angezeigt.
3. Klicken Sie mit der rechten Maustaste auf dem Eintrag, und wählen Sie im Kontextmenü die Option **Show Access Plan** aus. Das Fenster **Access Plan Graph** wird geöffnet.

Einzelheiten zur Bedeutung des Fensters **Access Plan Graph** und zur Optimierung von SQL-Anweisungen finden Sie in der Online-Hilfefunktion zur

Funktion Visual Explain im Menü **Help**. In der Online-Hilfefunktion erhalten Sie außerdem genaue Beispiele, wie Sie die Funktion Visual Explain verwenden können.

Verwalten ferner Datenbanken

Im folgenden Abschnitt werden die folgenden Themen behandelt:

- Hinzufügen eines fernen Systems
- Hinzufügen eines Exemplars, mit dem Sie arbeiten möchten, zum System
- Hinzufügen einer Datenbank, mit der Sie arbeiten möchten, unter diesem Exemplar

Zuerst überprüft DB2 das Knotenverzeichnis (das einen Eintrag für alle Server, mit denen ein Datenbank-Client eine Verbindung herstellen kann, und das für die Verbindung verwendete Übertragungsprotokoll enthält), um festzustellen, ob das ferne System bereits bekannt ist. Wenn das ferne System mit einem System, einem Exemplar oder einer Datenbank auf einem fernen System unbekannt ist, müssen Sie sich einen Client für das ferne System einrichten.

Nach der Installation von DB2 können Sie **Client-Konfiguration - Unterstützung** verwenden, um im Netzwerk Systeme, Exemplare und Datenbanken aufzuspüren und die Kommunikation dafür zu konfigurieren. Anschließend können Sie das ferne System hinzufügen, indem Sie es katalogisieren. Dadurch wird im Knotenverzeichnis ein Eintrag für das System erstellt, so dass dessen Exemplare und Datenbanken bekannt gegeben werden können. Anschließend müssen Sie die Exemplare und Datenbanken für das System hinzufügen, indem Sie sie katalogisieren, um für diese im Knotenverzeichnis bzw. im Datenbankverzeichnis einen Eintrag zu erstellen. Damit wird jeweils ein Eintrag im Knotenverzeichnis bzw. im Datenbankverzeichnis erstellt. Wenn die Konfiguration beendet ist, werden die fernen Systeme in der Steuerzentrale angezeigt, so dass Sie mit diesen arbeiten können.

Gehen Sie wie folgt vor, um ein fernes System hinzuzufügen:

1. Klicken Sie in der Steuerzentrale mit der rechten Maustaste auf dem Objekt **Systeme**, und wählen Sie im Kontextmenü **Hinzufügen** aus. Das Fenster **System hinzufügen** wird geöffnet.
2. Geben Sie den Systemnamen in das Feld **Systemname** ein.

Wenn der Konfigurationsparameter **Discover** für das Exemplar auf **Suchen** und der Konfigurationsparameter **discover comm** nicht leer ist, können Sie die Option **Aktualisieren** auswählen, um eine Liste ferner Systeme abzurufen. Anschließend können Sie in der Liste unterhalb des Feldes **Systemname** ein System auswählen.

3. Geben Sie den fernen Exemplarnamen in das Feld **Fernes Exemplar** ein.

4. Wählen Sie in der Liste **Betriebssystem** den Typ des Betriebssystems für das ferne System aus.
5. Wählen Sie das Protokoll aus, das für die Kommunikation mit den fernen Standorten verwendet werden soll. Für ein lokales System wird automatisch **Lokal** ausgewählt. Dies ist auch das einzige gültige Protokoll. Für die fernen Systeme sind die folgenden Protokolle möglich:
 - APPC
 - IPX/SPX
 - NetBIOS
 - TCP/IP
 - Benannte Pipe (nur unter den Betriebssystemen Windows NT und Windows 9x)

Nur die Protokolle, die auf dem Computer momentan installiert sind, werden im Listenfenster angezeigt.

6. Geben Sie die geeigneten Protokollparameter ein.
7. Geben Sie einen Kommentar ein, der dem System zugeordnet werden soll.
8. Klicken Sie **Anwenden** an, um das System dem Knotenverzeichnis hinzuzufügen.

Gehen Sie anschließend wie folgt vor, um das Exemplar hinzuzufügen, mit dem Sie auf diesem System arbeiten möchten:

1. Klicken Sie in der Steuerzentrale mit der rechten Maustaste auf dem Objekt **Exemplare**, das zum gerade hinzugefügten System gehört.
2. Wählen Sie im Kontextmenü die Option **Hinzufügen** aus. Das Fenster **Exemplar hinzufügen** wird geöffnet.
3. Geben Sie in die Felder die erforderlichen Werte ein.
4. Klicken Sie den Druckknopf **Aktualisieren** an, um eine Liste vorhandener Exemplare anzuzeigen.
5. Wählen Sie das Exemplar aus, mit dem Sie arbeiten möchten.
6. Klicken Sie den Druckknopf **Anwenden** und anschließend den Druckknopf **Schließen** an.

Gehen Sie schließlich wie folgt vor, um die Datenbank, mit der Sie arbeiten möchten, unter diesem Exemplar hinzuzufügen:

1. Klicken Sie in der Steuerzentrale mit der rechten Maustaste auf dem Objekt **Datenbanken**.
2. Wählen Sie im Kontextmenü die Option **Hinzufügen** aus. Das Fenster **Datenbank hinzufügen** wird geöffnet.

3. Geben Sie den Datenbanknamen, den Typ des Übertragungsprotokolls und, wahlfrei, einen Aliasnamen ein. Ein Aliasname ist in diesem Fall ein alternativer Name, der zur Kennzeichnung einer Datenbank dient.
4. Klicken Sie den Druckknopf **Aktualisieren** an, um eine Liste vorhandener Datenbanken für dieses Exemplar anzuzeigen.
5. Wählen Sie eine Datenbank aus.
6. Klicken Sie den Druckknopf **Anwenden** und anschließend den Druckknopf **Schließen** an.

Verwalten von Benutzern

Als Datenbankadministrator müssen Sie möglicherweise den Typ des Zugriffs, den Personen auf Daten haben, steuern oder deren Sicht auf die Daten einschränken. Im Folgenden wird beschrieben, wie Sie die Verwaltungs-Tools einsetzen können, um Datenbankberechtigungen und -zugriffsrechte auf Datenbankobjekte zu verwalten.

Berechtigungen für Datenbanken umfassen Aktionen für eine Datenbank als Ganzes. Wenn eine Datenbank erstellt wird, werden bestimmte Berechtigungen automatisch allen Personen erteilt, die auf die Datenbank zugreifen. Die Berechtigungen CONNECT, CREATETAB, BINDADD und IMPLICIT_SCHEMA werden z. B. allen Benutzern erteilt. *Zugriffsrechte* auf Datenbanken betreffen Aktionen, die sich auf bestimmte Objekte innerhalb der Datenbank beziehen. Wenn eine Datenbank erstellt wird, werden bestimmte Zugriffsrechte automatisch allen Personen erteilt, die auf die Datenbank zugreifen. Das Zugriffsrecht SELECT wird z. B. für Katalogsichten erteilt, und das Zugriffsrecht EXECUTE und BIND für alle erfolgreich gebundenen Dienstprogramme wird allen Benutzern erteilt.

Berechtigungen und Zugriffsrechte bilden in Kombination miteinander die Steuerung des Zugriffs auf ein Exemplar und seine Datenbankobjekte. Benutzer können nur auf solche Objekte zugreifen, für die sie berechtigt sind, d. h., für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigungsstufe verfügen.

Erteilen und Entziehen von Berechtigungen und Zugriffsrechten

Sie können mit Hilfe der DB2-Verwaltungs-Tools Benutzern und Gruppen Zugriffsrechte und Berechtigungen für Datenbanken, Tabellenbereiche, Tabellen, Sichten und Schemata erteilen und entziehen.

1. Klicken Sie in der Steuerzentrale mit der rechten Maustaste die Datenbanken, Tabellen, Sichten, Schemata oder Indizes an, für die Sie Zugriffsrechte erteilen oder entziehen möchten. Wählen Sie im Kontextmenü die Option **Berechtigung** oder **Zugriffsrechte** aus. Das Fenster **Berechtigungen** oder das Fenster **Zugriffsrechte** wird geöffnet.

2. Wählen Sie die Seite **Benutzer** aus, um mit Benutzerberechtigungen oder -zugriffsrechten zu arbeiten, und die Seite **Gruppe**, um mit Gruppenberechtigungen oder -zugriffsrechten zu arbeiten.
3. Wählen Sie einen oder mehrere Benutzer oder Gruppen aus. Wenn Sie der Liste einen Benutzer oder eine Gruppe hinzufügen möchten, klicken Sie den Druckknopf **Benutzer hinzufügen** oder **Gruppe hinzufügen** an.
4. Wählen Sie unten im Fenster für jede einzelne Berechtigung oder jedes einzelne Zugriffsrecht die Option **Ja**, **Nein** oder **Erteilen** aus. Die Option **Erteilen** wird nur für Objekte angezeigt, für die diese Option gültig ist.
5. Wenn Sie dies abgeschlossen haben, klicken Sie den Druckknopf **Anwenden** an.

Wenn Sie die Objekte, für die ein bestimmter Benutzer eine Berechtigung besitzt, überprüfen oder ändern möchten, können Sie einen Benutzer auswählen und mit der rechten Maustaste klicken. Anschließend können Sie die Berechtigung für ein Objekt hinzufügen, ändern oder löschen.

Versetzen von Daten

Mit DB2 werden die Dienstprogramme IMPORT, EXPORT und LOAD geliefert, mit denen Sie Daten aus vorhandenen Quellen in eine Tabelle versetzen können. Die Informationen in diesem Abschnitt geben eine kurze Übersicht über das Versetzen von Daten. Genauere Informationen zum Versetzen von Daten finden Sie im Handbuch *Versetzen von Daten Dienstprogramme und Referenz*.

Das Importdienstprogramm liest die Daten aus der Eingabedatei und fügt sie in eine Tabelle oder Sicht ein. In diesem Fall enthält die Eingabedatei Daten, die aus einer vorhandenen Datenquelle extrahiert wurden, wie z. B. aus einer Lotus 1–2–3-Datei oder einer ASCII-Datei. Außerdem können Sie mit dem Importdienstprogramm eine Tabelle oder Sicht, die mit dem Exportdienstprogramm gespeichert worden ist, neu erstellen. Im Folgenden wird erläutert, wie Sie Daten importieren können.

Sobald eine Eingabedatei in einem unterstützten Format verfügbar ist, können Sie das Notizbuch **Importieren** verwenden, um Daten aus einer Datei in eine vorhandene Tabelle einzufügen. Wenn diese Tabelle oder Sicht bereits Daten enthält, können Sie entweder die importierten Daten an die vorhandenen Daten anfügen oder die vorhandenen Daten ersetzen.

Das Notizbuch **Importieren** können Sie außerdem verwenden, um eine neue Tabelle zu erstellen, die von einer Eingabedatei aufgefüllt wird, oder vorhandene Zeilen in einer ausgewählten Tabelle löschen und mit Daten aus der Eingabedatei wieder auffüllen.

Gehen Sie wie folgt vor, um eine Datei in eine vorhandene Tabelle zu importieren:

1. Öffnen Sie die Seite **Datei** des Notizbuchs **Importieren**.
2. Wahlfrei: Geben Sie das Notizbuch **Importieren** an.
3. Wahlfrei: Wählen Sie **Große Objekte** aus.
4. Wahlfrei: Geben Sie die Spaltenimportoptionen an.
5. Klicken Sie **OK** an.

Gehen Sie wie folgt vor, um die Seite **Datei** des Notizbuchs **Importieren** zu öffnen:

1. Erweitern Sie in der Steuerzentrale den Objektbaum, bis Sie den Ordner **Tabellen** finden.
2. Klicken Sie den Ordner **Tabellen** an. Im Inhaltsteilfenster werden alle vorhandenen Tabellen angezeigt.
3. Klicken Sie im Inhaltsteilfenster mit der rechten Maustaste auf einer Tabelle, und wählen Sie im Kontextmenü die Option **Importieren** aus. Das Notizbuch **Importieren** wird geöffnet, in dem die Seite **Datei** angezeigt wird.

Gehen Sie wie folgt vor, um die Dateioptionen anzugeben:

1. Geben Sie auf der Seite **Datei** in das Feld **Importdatei** den Namen der Datei ein, die die zu importierenden Daten enthält.
2. Geben Sie den Typ der zu importierenden Datei an, indem Sie eine der folgenden Optionen auswählen:
 - **ASC (Non-Delimited ASCII Format)**
Beim ASC-Format sind die Daten in Spalten angeordnet.
 - **DEL (Delimited ASCII Format)**
Das DEL-Format ist allgemein üblich. Die Daten werden als Spaltenwerte gespeichert, die durch einen benutzerdefinierten Begrenzer wie z. B. ein Komma getrennt sind.
 - **WSF (Work Sheet Format)**
 - **IXF (Integrated Exchange Format)**
PC/IXF ist eine strukturierte Beschreibung einer Datenbanktabelle oder Sicht. Die Daten, die in das PC/IXF-Format exportiert wurden, können in ein anderes DB2 Universal Database-Produkt importiert oder geladen werden.

Weitere Informationen zu spezifischen Produkten und unterstützten Releases finden Sie in der Online-Hilfefunktion.

3. Wahlfrei: Geben Sie die Änderungswerte für den Dateityp an, indem Sie den entsprechenden Druckknopf **Optionen** anklicken. Das Fenster **Optionen** für das entsprechende Format wird geöffnet.

4. Wählen Sie einen Wert für **Importmodus** aus. Die verfügbaren Importmodi sind je nach ausgewähltem Dateityp verschieden.
5. Wahlfrei: Geben Sie in das Feld **Datensätze vor Festschreiben** die Anzahl der zu Datensätze an, die importiert werden sollen, bevor die Änderungen festgeschrieben werden.
6. Wahlfrei: Geben Sie im Feld **Erneut starten** die Anzahl der Datensätze in der Datei an, die übersprungen werden sollen, bevor das Importieren beginnt.
7. Wahlfrei: Geben Sie in das Feld **Compound-SQL** die Anzahl der SQL-Anweisungen, die ausgeführt werden sollen (in einem ausführbaren Block).
8. Wahlfrei: Wählen Sie das Markierungsfeld **Impliziertes Dezimalzeichen für Dezimaldaten einfügen (IMPLIEDDECIMAL)** aus.
9. Geben Sie im Feld **Nachrichtendatei** den Namen der Datei ein, in der Warnungen und Fehlermeldungen des Importvorgangs abgelegt werden sollen.

Sie können große Objekte aus separaten Dateien abrufen, indem Sie mit Hilfe der Seite **Große Objekte** im Notizbuch **Importieren** die großen Objekte (LOBs) aus dem Pfad oder den Pfaden abrufen, in denen die LOB-Dateien gespeichert sind.

1. Klicken Sie das Markierungsfeld **Große Objekte (LOBs) in separaten Dateien abrufen (LOBSINFIL)** an, um die Optionen auf der Seite **Große Objekte** zu aktivieren.
2. Geben Sie die Position der separaten LOB-Dateien im Listenfenster **LOB-Pfade** an, indem Sie den Druckknopf **Hinzufügen** anklicken. Diese Pfade werden nach LOB-Dateien durchsucht (in der Reihenfolge, wie sie im Listenfenster **LOB-Pfade** angezeigt sind), die in der Spalte **LOB** der Eingabedatei angegeben sind.
3. Klicken Sie **OK** an, um die Standardwerte auf den übrigen Notizbuchseiten zu übernehmen und mit dem Importieren zu beginnen.

Geben Sie die Spaltenimportoptionen an. Gehen Sie wie folgt vor, um auf der Seite **Spalten** im Notizbuch **Importieren** die Spaltenimportoptionen anzugeben:

1. Klicken Sie einen der Radioknöpfe im Fenster **Methode für Spaltenimport** an, um die Methode anzugeben, mit der die Spalten der Datendatei in die Tabelle importiert werden. Die verfügbaren Methoden sind verschieden je nach dem Dateityp und dem Modus, den Sie auf der Seite **Datei** auswählen.
2. Wahlfrei: Geben Sie die Spaltenattribute der Importdatei an oder ändern Sie diese, indem Sie den Druckknopf **Ändern** anklicken.

Diese Option ist allerdings nicht verfügbar, wenn Sie den Radioknopf **Standard (Methode S)** ausgewählt haben.

Verwalten von Speicher

Als Datenbankadministrator müssen Sie die Größe der Tabellen und Indizes schätzen und den verfügbaren Speicherplatz in einem Tabellenbereich überprüfen. Wenn dieser belegt ist, müssen Sie zum vorhandenen Speicherplatz mehr Speicherplatz hinzufügen.

In diesem Abschnitt werden die folgenden Themen behandelt:

- Schätzen der Größe von Tabellen und Indizes
- Überprüfen des verfügbaren Speicherplatzes im Tabellenbereich
- Hinzufügen von mehr Speicherplatz in einer vorhandenen Tabelle, wenn diese zunehmend belegt ist

Schätzen der Tabellen- und Indexgröße

Sie können den Speicherplatz schätzen, der für neue oder vorhandene Tabellen oder Indizes erforderlich ist, indem Sie das Dialogfenster **Größe schätzen** verwenden. Dieses Dialogfenster können Sie aufrufen, indem Sie einzelne Tabellen und Indizes auswählen und, während sich der Mauszeiger darüber befindet, mit der rechten Maustaste klicken oder indem Sie im Fenster **Tabelle erstellen** oder **Index erstellen** die Option **Größe schätzen** auswählen. Die Größe wird anhand der Definition der entsprechenden Tabelle und deren abhängiger Indizes geschätzt. Die Schätzung ergibt den voraussichtlichen Speicherbedarf für die Tabelle, wenn diese eine gegebene Anzahl von Zeilen enthält. Der Mindest- und Höchstspeicherbedarf wird ebenfalls anhand der kleinsten und größten Felder mit variabler Länge geschätzt. Wenn das Dialogfenster **Größe schätzen** für eine Tabelle oder einen Index aufgerufen wird, wird es zunächst mit den Spezifikationen der Tabelle ausgefüllt und enthält Zahlen, die sich auf die Tabelle und alle davon abhängigen Indizes beziehen. Wenn Sie den Druckknopf **Aktualisieren** anklicken, werden die geschätzte Größe, das Minimum und das Maximum anhand der Zahl aktualisiert, die Sie in den Feldern **Neue Gesamtzahl der Zeilen** und **Neue durchschnittliche Zeilenlänge** eingegeben haben.

Die Schätzung der Größe einer Tabelle oder eines Index ist zu folgenden Zwecken nützlich:

- Erstellen einer neuen Tabelle, wobei Sie ermitteln möchten, wie groß der Tabellenbereich sein muss.
- Erstellen einer neuen Tabelle auf der Basis der Schätzung einer vorhandenen Tabelle.
- Ermitteln, wie viel Speicherplatz andere Tabellen- und Indexobjekte in einem Tabellenbereich belegen, da das System bald nicht mehr ausreichend Speicherplatz besitzt.
- Schätzen der voraussichtlichen Größe einer Tabelle, bevor Daten geladen werden.

Anmerkung: Wenn Sie die Option 'Größe schätzen' bei einem DB2 Universal Database Enterprise-Extended Edition-Produkt anwenden, stellen Sie sicher, dass die Schätzungen auf der logischen Größe der Daten in der Tabelle basieren.

Wenn Sie die Statistik für die Tabelle einige Zeit nicht aktualisiert haben, können Sie den Druckknopf **Statistik ausführen** anklicken, um diese Daten für die ausgewählte Tabelle zu aktualisieren. Wenn Sie einen Index auswählen und anschließend den Knopf **Statistik ausführen** drücken, wird die Statistik der zugehörigen Tabelle aktualisiert.

Gehen Sie wie folgt vor, um die Größe einer Tabelle zu schätzen:

- Öffnen Sie das Fenster **Größe schätzen**.
- Wählen Sie für **Neue Gesamtzahl der Zeilen** einen anderen Wert aus, oder übernehmen Sie den Standardwert.
- Klicken Sie **Aktualisieren** an, um die Größenschätzungen für den neuen Wert anzuzeigen.
- Wählen Sie für **Neue durchschnittliche Zeilenlänge** einen anderen Wert aus, oder übernehmen Sie den Standardwert.
- Klicken Sie **Aktualisieren** an, um die Größenschätzungen für den neuen Wert anzuzeigen.

Überprüfen des verfügbaren Speichers in einem Tabellenbereich

Gehen Sie wie folgt vor, um den verfügbaren Speicher in einem DMS-Tabellenbereich zu überprüfen:

1. Wählen Sie in der Steuerzentrale das Symbol **Tabellenbereiche** durch Doppelklicken aus. Im Inhaltsteilfenster wird eine Liste aller Tabellenbereiche angezeigt.
2. Blättern Sie durch die Spalten mit den Überschriften **Zugeordnete Größe**, **Verwendete Größe** und **Verwendeter Speicher (Prozent)**, um die Einzeldaten zum verfügbaren Speicher in einem Tabellenbereich anzuzeigen. Der Speicherplatz wird in Seiten angegeben. Eine Seite entspricht 4 KB.

Sie können anpassen, welche Spalten in welcher Reihenfolge angezeigt werden. Dazu können Sie das Symbol **Spalten anpassen** verwenden, das sich unten im Inhaltsteilfenster befindet.

Zur Überprüfung des verfügbaren Speichers in einem SMS-Tabellenbereich können Sie die Mittel Ihres Betriebssystems für die Überwachung des verfügbaren Speichers verwenden. Dadurch können Sie sicherstellen, dass der verfügbare Speicherplatz im Verzeichnis des Tabellenbereichs nicht knapp wird.

Hinzufügen von mehr Speicher für einen Tabellenbereich

Die Kapazität für einen DMS-Tabellenbereich entspricht der Gesamtgröße der Behälter, die dem Tabellenbereich zugeordnet sind. Wenn ein DMS-Tabellenbereich seine Kapazitätsgrenze erreicht, sollten Sie mehr Speicher hinzufügen. Je nach der Nutzung des Tabellenbereichs ist 90% ein möglicher Schwellenwert. Der Datenbankmanager gleicht die Tabellen in einem DMS-Tabellenbereich zwischen den verfügbaren Behältern automatisch neu aus. Während des Ausgleichens bleiben die Daten im Tabellenbereich verfügbar.

Einem DMS-Tabellenbereich, der seine Kapazitätsgrenze erreicht hat, können Sie einen weiteren Behälter hinzufügen:

1. Klicken Sie im Inhaltsteilfenster der Steuerzentrale mit der rechten Maustaste auf dem Tabellenbereich, für den Sie einen neuen Behälter hinzufügen möchten, und wählen Sie im Kontextmenü die Option **Ändern** aus. Das Fenster **Tabellenbereich ändern** - wird geöffnet.
2. Klicken Sie **Hinzufügen** an. Das Fenster **Behälter hinzufügen** wird geöffnet.
3. Wählen Sie den Radioknopf **Datei** oder **Unformatierte Einheit** aus, und vervollständigen Sie die Felder. Weitere Informationen finden Sie in der Online-Hilfefunktion.
4. Klicken Sie **OK** an.

Im allgemeinen können Sie die Größe eines SMS-Tabellenbereichs nicht einfach erweitern, da die SMS-Kapazität vom Speicherplatz, der im Dateisystem verfügbar ist, und von der unterstützten Maximalgröße für Dateien im jeweiligen Betriebssystem abhängt. Allerdings können Sie je nach Betriebssystem möglicherweise das Dateisystem vergrößern, indem Sie die Mittel Ihres Betriebssystems einsetzen. Für einen SMS-Tabellenbereich unter einem UNIX-basierten System können Sie die Größe der Tabellenbereiche erhöhen, indem Sie die geeigneten UNIX-Systembefehle verwenden. Weitere Informationen finden Sie in der Dokumentation zu Ihrem UNIX-basierten System. Wenn das Dateisystem, in dem sich der SMS-Tabellenbereich befindet, auch Nicht-DB2-Dateien enthält, können Sie möglicherweise diese Dateien in ein anderes Dateisystem verschieben und somit für DB2 mehr Speicherplatz gewinnen. Außerdem können Sie eine umgeleitete Wiederherstellung ausführen. Dabei wird ein Tabellenbereich in einer größeren Anzahl von Behältern als zuvor wiederhergestellt. Zur umgeleiteten Wiederherstellung können Sie das Notizbuch **Datenbank wiederherstellen** verwenden. Wählen Sie von der wiederherzustellenden Datenbank aus im Kontextmenü die Optionen **Wiederherstellen** → **Datenbank** aus.

Fehlerbehebung

Zu DB2 gibt es ein Handbuch zur Fehlerbehebung, das sich an den technischen Dienst für DB2-Server und -Clients richtet. Dort finden Sie die folgenden Informationen:

- Schnelles Erkennen von Problemen oder Fehlern
- Beheben von Fehlern anhand der Symptome
- Verwenden der verfügbaren Diagnose-Tools
- Entwickeln einer Strategie zur Fehlerbehebung für den tagtäglichen DB2-Betrieb

Im Handbuch *Troubleshooting Guide* werden die folgenden grundlegenden Themen zur Fehlerbehebung behandelt:

- Gute Methoden zur Fehlerbehebung
- Fehlerbehebung auf dem Server
- Fehlerbehebung auf dem Client
- Fehlerbehebung bei der Host-Übertragung
- Fehlerbehebung bei Anwendungen
- Fehlerbehebung und Fehlerbestimmung

Im Handbuch *Troubleshooting Guide* werden die folgenden weiterführenden Themen zur Fehlerbehebung behandelt:

- DB2-Prozessmodell
- Verwenden von Protokolldaten
- Ablaufverfolgung
- Diagnose-Tools für die Betriebssysteme UNIX, OS/2 und Microsoft Windows

Aktuelle Bekanntmachungen und technische Informationen sind im World Wide Web unter der folgenden Adresse verfügbar:

<http://www.software.ibm.com/data/db2/library/>

Weitere Einzelheiten zur Kontaktaufnahme mit IBM finden Sie im entsprechenden Abschnitt am Ende dieses Handbuchs.

Replizieren von Daten

Die Replikation ist der Vorgang, bei dem im Datenbankprotokoll gespeicherte Änderungen auf dem Quellenserver auf dem Zielsystem angewendet werden. Sie können die Replikation verwenden, um Kopieroperationen für Daten im gesamten Unternehmen zu definieren, zu synchronisieren, zu automatisieren und zu verwalten. Sie können die Daten automatisch von einem Host-System auf Ziel-Systeme übertragen. Die Daten und Anwendungen können Sie z. B. an Zweigstellen, an Einzelhandelsverkaufsstellen und selbst an Notebooks von Vertriebsbeauftragten senden.

Die Replikation besteht aus den beiden Komponenten Capture-Programm und Apply-Programm. Das Capture-Programm zeichnet Änderungen an den Daten der Quellentabellen auf, die zur Replikation definiert wurden. Dazu wird das Datenbankprotokoll gelesen. Das Apply-Programm liest die geänderten Daten, die zuvor aufgezeichnet und in einer CD-Tabelle gespeichert worden sind, und wendet sie auf die Zieltabellen an.

Mit Hilfe der Steuerzentrale können Sie die für die Replikation erforderliche Konfiguration ausführen. Dazu verwenden Sie die Aktionen **Als Replikationsquelle definieren** und **Subskription definieren**. Die Replikationskomponenten Capture und Apply können außerhalb der DB2-Verwaltungs-Tools ausgeführt werden.

Replikationsadministratoren können in der Steuerzentrale die folgenden Aktionen ausführen:

- Definieren von Replikationsquellen
- Definieren von Replikationssubskriptionen
- Angeben von SQL, um die Datenübertragung während der Ausführung des Apply-Programms zu verbessern

Die allgemeinen Schritte zur Datenreplikation werden im Folgenden beschrieben. Genauere Einzelheiten finden Sie im *Replikation Benutzer- und Referenzhandbuch*.

1. Entwerfen Sie ein Replikationsszenario. (Ordnen Sie die Quellen- und Zieltabellen einander zu.)
2. Definieren Sie eine Replikationsquelle. (Dies bezieht sich auf die Aufzeichnung.)

Gehen Sie wie folgt vor, um eine Replikationsquelle zu definieren:

1. Geben Sie die Quellenspalten an, die aufzuzeichnen sind.
2. Wählen Sie die Replikationsoptionen aus.
3. Definieren Sie eine Replikationssubskription. (Dies bezieht sich auf das Anwenden.)
4. Ändern Sie die Quellentabelle mit der Option **Data Capture Changes**.

5. Starten Sie das Capture-Programm, um die Datenänderungen zu lesen und zu speichern.
6. Starten Sie das Apply-Programm, um die Änderungen in den Zieltabellen zu replizieren.

Gehen Sie wie folgt vor, um eine Replikationssubskription zu definieren:

1. Benennen Sie die Subskriptionsgruppe.
2. Geben Sie die Datenbank und die Zieltabelle an.
3. Geben Sie die Zielspalten an.
4. Geben Sie die Zeilenauswahl an.
5. Geben Sie SQL zur Laufzeitverarbeitung an.
6. Definieren Sie die Ablaufsteuerung für die Subskription.

Verwenden des Protokolls LDAP (Lightweight Directory Access Protocol)

Die Client-Konfiguration - Unterstützung kann zum Hinzufügen und Löschen von Einträgen auf einem LDAP-Server verwendet werden. Alle Datenbankexemplare, die auf dem LDAP-Server registriert sind, werden auf dem Client automatisch katalogisiert (im Cache zwischengespeichert). In der Steuerzentrale werden Sie als normale Knoten im Baum angezeigt. Diese Datenbanken können in derselben Weise verwaltet werden wie andere Datenbanken, die Sie auf Ihrer Maschine katalogisiert haben. Die Optionen ADD DATABASE und DROP DATABASE sind allerdings in diesem Release noch nicht verfügbar.

Wenn Sie eine LDAP-Datenbank verwalten möchten, wählen Sie die Datenbank aus, und klicken Sie anschließend mit der rechten Maustaste. In einem Kontextmenü werden die Funktionen aufgelistet, die Sie ausführen können. Weitere Informationen zu LDAP finden Sie im „Anhang J. LDAP-Verzeichnisservices (Lightweight Directory Access Protocol)“ auf Seite 469.

Verwenden der Java-Steuerzentrale

Sie können die Steuerzentrale als Java-Anwendung oder als Java-Applet über einen Web-Server ausführen. In beiden Fällen benötigen Sie eine unterstützte Java Virtual Machine (JVM), die auf Ihrer Maschine installiert ist und mit der die Steuerzentrale ausgeführt werden kann. Zur Ausführung der Steuerzentrale als Java-Anwendung müssen Sie außerdem die richtige Java-Umgebung (Java Runtime Environment, JRE) installiert haben. Eine JVM (Java Virtual Machine) kann für aktive Anwendungen eine Java Runtime Environment (JRE) oder für aktive Applets ein Java-fähiger Browser sein.

Java-Anwendungen werden auf Ihrer Maschine wie andere Anwendungen ausgeführt, sofern Sie die richtige JRE installiert haben.

Ausführen der Steuerzentrale als Java-Applet

Java-Applets sind Programme, die in einem Java-fähigen Browser ausgeführt werden. Der Applet-Code der Steuerzentrale kann sich auf einer fernen Maschine befinden. In diesem Fall wird der Code von einem Web-Server an den Browser des Clients gesendet. Wenn Sie die Steuerzentrale als Java-Applet ausführen, müssen Sie einen unterstützten, Java-fähigen Browser verwenden, der unter Windows (32 Bit) oder unter OS/2 läuft. Momentan sind für UNIX-Systeme keine unterstützten Browser verfügbar.

Der JDBC Applet Server der Steuerzentrale muss mit einem Benutzereintrag mit Administratorberechtigung für diejenige Maschine gestartet werden, auf der sich der Applet-Server befindet. Sie können JDBC Applet Server der Steuerzentrale beim Systemstart automatisch starten lassen.

Wenn Sie die Steuerzentrale als Java-Applet ausführen, muss auf der Maschine, auf der sich der Applet-Code und der JDBC Applet Server der Steuerzentrale befinden, ein Web-Server konfiguriert sein. Der Web-Server muss den Zugriff auf das Verzeichnis `sql11ib` gewähren. Wenn Sie ein virtuelles Verzeichnis verwenden, ersetzen Sie damit das Ausgangsverzeichnis. Wenn Sie z. B. das virtuelle Verzeichnis `temp` nennen, müssen Sie das Verzeichnis `sql11ib/temp` verwenden. DB2 unterstützt keine Installation auf einem FAT-Laufwerk für OS/2, da OS/2 FAT keine langen Dateinamen unterstützt. Diese sind jedoch für Java erforderlich. Weitere Informationen zum Installieren und Konfigurieren der Steuerzentrale als Java-Anwendung oder Java-Applet finden Sie im Handbuch *Einstieg* für die verwendete Plattform.

Verwenden der Java-basierten Tools zur Verwaltung

DB2 enthält eine Reihe von Java-Schnittstellen, mit denen Sie die Funktionen der Steuerzentrale erweitern können. Die Java-Schnittstellen können Sie zu folgenden Zwecken verwenden:

- Hinzufügen weiterer Elemente in die Menüliste, wenn Sie mit Objekten arbeiten.
- Hinzufügen von Knöpfen in der Menüleiste der Steuerzentrale.

Wenn Sie diese Funktion verwenden möchten, muss die richtige Version der Java-Software installiert sein. Weitere Informationen zur Verwendung dieser Funktion finden Sie in „Anhang K. Erweitern der Steuerzentrale“ auf Seite 505.

Teil 2. Implementieren des Datenbankentwurfs

Kapitel 2. Vor dem Erstellen einer Datenbank

Wenn der Datenbankentwurf festliegt, müssen die Datenbank und die in ihr enthaltenen Objekte erstellt werden. Zu diesen Objekten gehören Schemata, Knotengruppen, Tabellenbereiche, Tabellen, Sichten, Oberflächen, Server, Kurznamen, Datentypzuordnungen, Funktionszuordnungen, Aliasnamen, benutzerdefinierte Datentypen (UDTs), benutzerdefinierte Funktionen (UDFs), Integritätsbedingungen, Indizes und Pakete. Sie können diese Objekte mit Hilfe von SQL-Anweisungen über den Befehlszeilenprozessor, mit Hilfe der Steuerzentrale oder mit Hilfe von APIs in Anwendungen erstellen.

Informationen zu SQL-Anweisungen finden Sie im Handbuch *SQL Reference*. Informationen zu den Befehlen des Befehlszeilenprozessors enthält das Handbuch *Command Reference*. Informationen zu APIs finden Sie im Handbuch *Administrative API Reference*.

Anmerkung: Ihre Plattform unterstützt vielleicht eine Benutzerschnittstelle, mit deren Hilfe Sie Datenbankobjekte erstellen können. Diese Schnittstelle kann anstelle der SQL-Anweisungen, der Befehle des Befehlszeilenprozessors oder der APIs verwendet werden. Im Handbuch *Einstieg* für Ihre Plattform finden Sie Informationen darüber, ob Sie diese Möglichkeit haben.

Im vorliegenden Kapitel wird das Verfahren zum Abschließen von Tasks mit Hilfe der Steuerzentrale durch ein Kästchen hervorgehoben. Direkt anschließend wird ein vergleichbares Verfahren für die Befehlszeile aufgeführt, das z. T. durch entsprechende Beispiele ergänzt wird. In einigen Fällen kann für eine Task auch nur ein Verfahren aufgeführt sein. Beim Arbeiten mit der Steuerzentrale finden Sie in der Hilfe detailliertere Informationen zu den im vorliegenden Handbuch angesprochenen Themen.

Das vorliegende Kapitel konzentriert sich auf die Bereitstellung der Informationen, mit denen Sie vor dem Erstellen einer Datenbank mit den zugehörigen Objekten vertraut sein sollten. Es gibt verschiedene Konzepte und Themen, die zur Ausführung dieser Arbeitsschritte erforderlich sind, sowie mehrere Tasks, die vor dem Erstellen einer Datenbank ausgeführt werden müssen.

Das folgende Kapitel enthält eine kurze Erläuterung der verschiedenen Objekte, die Bestandteil der Implementierung Ihres Datenbankentwurfs sein können.

Das letzte Kapitel in diesem Teil enthält Themen, die vor dem Ändern einer Datenbank berücksichtigt werden müssen. Anschließend wird das Ändern und Löschen von Datenbankobjekten erläutert.

Für die Bereiche, bei denen DB2 Universal Database mit dem Betriebssystem zusammenarbeitet, werden in den im vorliegenden und den nachfolgenden Kapiteln behandelten Themen betriebssystemspezifische Unterschiede erläutert. Eventuell können Sie die betriebssystemeigenen Funktionen oder Unterschiede nutzen, die über die von DB2 Universal Database vorgesehenen hinausgehen. Detaillierte Informationen zu den bestehenden Unterschieden können den entsprechenden Handbüchern *Einstieg* sowie der Dokumentation für das Betriebssystem entnommen werden.

Zum Beispiel unterstützt Windows NT eine Art von Anwendung, die als „Service“ bekannt ist. Unter DB2 für Windows NT kann ein DB2-Exemplar als Service definiert werden. Ein Service kann automatisch beim Booten des Systems, von einem Benutzer über die Steueranwendung für Services oder durch eine auf Win32 basierende Anwendung, die die Servicefunktionen der Anwendungsprogrammierschnittstelle (API) Microsoft Win32 verwendet, gestartet werden. Services können ausgeführt werden, auch wenn kein Benutzer am System angemeldet ist.

Vorbedingungen für das Erstellen einer Datenbank

Vor dem Implementieren einer Datenbank sollten Sie sich mit den folgenden hierbei erforderlichen Aufgaben vertraut machen:

- „Starten von DB2“
- „Starten von DB2 UDB auf Windows NT“ auf Seite 59
- „Verwenden mehrerer Exemplare des Datenbankmanagers“ auf Seite 60
- „Organisieren und Gruppieren von Objekten nach Schema“ auf Seite 61
- „Aktivieren der Parallelität“ auf Seite 62
- „Ermöglichen der Datenpartitionierung“ auf Seite 64
- „Stoppen von DB2“ auf Seite 67

Starten von DB2

Das Starten und Stoppen von DB2 kann während des normalen Betriebs erforderlich werden. Beispielsweise müssen Sie ein Exemplar starten, um folgende Operationen auszuführen:

- Herstellen einer Verbindung zu einer Datenbank im Exemplar
- Vorkompilieren einer Anwendung
- Binden eines Pakets an eine Datenbank
- Zugreifen auf Host-Datenbanken

Gehen Sie wie folgt vor, um auf Ihrem System ein DB2-Exemplar zu starten:

1. Melden Sie sich mit einer Benutzer-ID oder einem Benutzernamen an, die bzw. der über die Berechtigung SYSADM, SYSCTRL oder SYSMAINT für das Exemplar verfügt. Oder melden Sie sich als Exemplareigner an.
2. Führen Sie unter UNIX-Betriebssystemen die Startprozedur wie folgt aus:
 . INSTHOME/sql1lib/db2profile (für Bourne- oder Korn-Shell)
 source INSTHOME/sql1lib/db2cshrc (für C-Shell)

Dabei ist INSTHOME das Benutzerverzeichnis des Exemplars, das Sie verwenden wollen.

3. Verwenden Sie eine der folgenden Methoden, um das Exemplar zu starten.
 - a. Gehen Sie wie folgt vor, um das Exemplar mit Hilfe der Steuerzentrale zu starten:

- 1) Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Exemplare** angezeigt wird.
- 2) Klicken Sie mit der rechten Maustaste auf dem zu startenden Exemplar und wählen Sie dann im Kontextmenü **Starten** aus.

- b. Geben Sie in der Befehlszeile Folgendes ein, um das Exemplar zu starten:

```
db2start
```

Anmerkung: Der Befehl **db2start** startet das Exemplar gemäß den in „Festlegen des aktuellen Exemplars“ auf Seite 78 angegebenen Regeln.

Starten von DB2 UDB auf Windows NT

Durch den Befehl `db2start` wird DB2 als NT-Service (Dienst) gestartet. DB2 auf Windows NT kann immer noch als Prozess ausgeführt werden, indem der Schalter `"/D"` beim Aufrufen des Befehls `DB2START` angegeben wird. DB2 kann auch als Dienst (Service) über die Systemsteuerung oder mit Hilfe des Befehls `"NET START"` gestartet werden.

Damit DB2 erfolgreich als Dienst mit `DB2START` gestartet werden kann, muss das Benutzerkonto über das richtige Zugriffsrecht verfügen, wie es vom Betriebssystem Windows NT zum Starten eines NT-Dienstes definiert wird. Das Benutzerkonto muss ein Mitglied der Gruppe der Administratorengruppe, der Server-Bediener oder der Power User sein.

Bei Ausführung in einer partitionierten Datenbankumgebung wird jeder Datenbankpartitionsserver als NT-Dienst gestartet.

Verwenden mehrerer Exemplare des Datenbankmanagers

Auf einem Server können mehrere Exemplare des Datenbankmanagers erstellt werden. Das heißt, dass verschiedene Exemplare desselben Produkts auf einem physischen System erstellt und gleichzeitig ausgeführt werden können. Dadurch haben Sie flexible Möglichkeiten bei der Einrichtung der Umgebungen.

Mehrere Exemplare können z. B. zum Erstellen der folgenden Umgebungen wünschenswert sein:

- Trennen der Entwicklungsumgebung von der Geschäftsumgebung
- Getrenntes Optimieren für spezielle Anwendungen, die ausgeführt werden sollen
- Schützen empfindlicher Informationen vor Administratoren. Es kann zum Beispiel erforderlich sein, die Lohn- und Gehaltsdaten in einem eigenen Exemplar geschützt zu speichern, so dass Eigner anderer Exemplare diese Daten nicht einsehen können.

Die DB2-Programmdateien sind physisch an einer einzigen Position auf einem bestimmten System gespeichert. Jedes Exemplar, das erstellt wird, enthält einen Zeiger auf diese Speicherposition, so dass die Programmdateien bei der Erstellung eines weiteren Exemplars nicht kopiert werden. In einem Exemplar können mehrere zusammengehörige Datenbanken gespeichert werden.

Exemplare werden entweder als lokal (local) oder als fern (remote) im Knotenverzeichnis katalogisiert. Das Standardexemplar wird durch die Umgebungsvariable DB2INSTANCE definiert. Sie können die Verbindung zu anderen Exemplaren (Befehl ATTACH) herstellen, um Wartungsaufgaben und Dienstprogramme auszuführen, die nur auf Exemplarebene ausgeführt werden können, wie z. B. Erstellen einer Datenbank, Abbrechen von Anwendungen, Monitoraufzeichnungen für eine Datenbank oder Aktualisieren der Konfiguration des Datenbankmanagers. Wenn Sie versuchen, die Verbindung zu einem Exemplar herzustellen, das nicht Ihr Standardexemplar ist, wird anhand des Knotenverzeichnisses festgestellt, wie die Kommunikation mit dem anderen Exemplar hergestellt werden kann.

Im Handbuch *Command Reference* finden Sie Informationen darüber, welche Art von Verbindung für die Ausführung des jeweiligen Befehls erforderlich ist.

Die DB2-Unterstützung für mehrere Exemplare ist je nach Betriebssystem unterschiedlich. Informationen, wie mehrere DB2-Exemplar auf einer Maschine definiert werden, finden Sie im Handbuch *Einstieg* für die entsprechende Plattform.

Zur Herstellung der Verbindung mit einem anderen Exemplar, das auch ein fernes Exemplar sein kann, verwenden Sie den Befehl ATTACH, wie im Handbuch *Command Reference* beschrieben.

Gehen Sie wie folgt vor, um die Steuerzentrale zu verwenden:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Exemplare** angezeigt wird.
2. Klicken Sie auf dem Exemplar, für das eine Verbindung hergestellt werden soll.
3. Klicken Sie mit der rechten Maustaste auf dem Namen des ausgewählten Exemplars.
4. Geben Sie im DB2-Fenster 'Verbindung herstellen' Ihre Benutzer-ID und das zugehörige Kennwort ein und klicken Sie anschließend **OK** an.

Geben Sie in der Befehlszeile Folgendes ein, um eine Verbindung für das Exemplar herzustellen:

```
db2 attach to <exemplarname>
```

Um z. B. eine Verbindung zu dem Exemplar mit dem Namen testdb2 herzustellen, das zuvor im Knotenverzeichnis katalogisiert wurde, müssen Sie folgenden Befehl eingeben:

```
db2 attach to testdb2
```

Wenn die Wartungsaktivitäten für das Exemplar testdb2 beendet sind, kann die Verbindung mit diesem Exemplar durch Ausführen des folgenden Befehls *freigegeben* werden:

```
db2 detach
```

Organisieren und Gruppieren von Objekten nach Schema

Datenbankobjektnamen bestehen aus einer einzelnen Kennung, oder sie sind *über ein Schema qualifizierte Objekte* mit zwei Kennungen. Das Schema, d. h. die höherwertige Komponente, eines dementsprechend qualifizierten Objekts ist eine Art, Objekte in der Datenbank zu klassifizieren oder zu gruppieren. Wenn ein Objekt wie eine Tabelle, eine Sicht, ein Aliasname, ein benutzerdefinierter Datentyp, eine Funktion, ein Index, ein Paket oder ein Auslöser erstellt wird, wird es einem Schema zugeordnet. Diese Zuordnung erfolgt entweder explizit oder implizit.

Explizite Verwendung des Schemas tritt auf, wenn Sie die höherwertige Komponente eines zweiteiligen Objektnamens beim Verweisen auf das Objekt in einer Anweisung verwenden. Der Benutzer A führt zum Beispiel eine Anweisung CREATE TABLE in Schema C wie folgt aus:

```
CREATE TABLE C.X (COL1 INT)
```

Implizite Verwendung des Schemas tritt auf, wenn Sie die höherwertige Komponente eines zweiteiligen Objektnamens nicht verwenden. Wenn dies eintritt, wird anhand des Sonderregisters CURRENT SCHEMA der Schemenname angegeben, mit dem die höherwertige Komponente des Objektnamens vervollständig werden soll. Der Anfangswert von CURRENT SCHEMA ist die Berechtigungs-ID des Benutzers der aktuellen Sitzung. Wenn Sie diesen Wert während der aktuellen Sitzung ändern wollen, können Sie das Sonderregister mit der Anweisung SET SCHEMA auf einen anderen Schemennamen einstellen. Weitere Informationen finden Sie im Handbuch *SQL Reference*.

Wie in „Definition von Systemkatalogtabellen“ auf Seite 119 beschrieben, werden bei der Erstellung einer Datenbank einige Objekte innerhalb bestimmter Schemata erstellt.

In dynamischen SQL-Anweisungen verwendet ein über ein Schema qualifizierter Objektname implizit den Wert des Sonderregisters CURRENT SCHEMA als Qualifikationsmerkmal für Objektnamenverweise ohne Qualifikationsmerkmal. In statischen SQL-Anweisungen gibt die Vorkompilierungs-/Bindeoption QUALIFIER implizit das Qualifikationsmerkmal für Datenbankobjektnamen ohne Qualifikationsmerkmal an.

Bevor Sie eigene Objekte erstellen, müssen Sie entscheiden, ob Sie sie im eigenen Schema erstellen wollen oder ob Sie ein anderes Schema verwenden wollen, das die Objekte logisch gruppiert. Wenn Objekte erstellt werden, die gemeinsam benutzt werden sollen, kann die Verwendung eines anderen Schemennamens sehr vorteilhaft sein. Weitere Informationen dazu, wie ein Schema explizit erstellt wird, finden Sie in „Erstellen eines Schemas“ auf Seite 133.

Aktivieren der Parallelität

Zur Nutzung der Parallelität innerhalb einer Datenbankpartition oder einer nichtpartitionierten Datenbank müssen Sie die Konfigurationsparameter ändern. Mit der partitionsinternen Parallelität können Sie z. B. mehrere Prozessoren einer SMP-Maschine (SMP = Symmetric Multi-Processor) nutzen.

Aktivieren der partitionsinternen Parallelität

Die Steuerzentrale kann zum Ermitteln oder Ändern der Werte für individuelle Einträge in einer spezifischen Datenbank oder der Konfigurationsdatei des Datenbankmanagers verwendet werden.

Zum Ermitteln der Werte für individuelle Einträge in einer bestimmten Datenbank oder der Konfigurationsdatei des Datenbankmanagers können Sie auch die Befehle GET DATABASE CONFIGURATION und GET DATABASE MANAGER CONFIGURATION verwenden. Zur Änderung einzelner Einträge für eine bestimmte Datenbank oder in der Konfigurationsdatei des

Datenbankmanagers werden die Befehle UPDATE DATABASE CONFIGURATION bzw. UPDATE DATABASE MANAGER CONFIGURATION verwendet.

Konfigurationsparameter, die sich auf die partitionsinterne Parallelität auswirken, sind die Datenbankmanagerparameter *max_querydegree* und *intra_parallel* sowie der Datenbankparameter *dft_degree*. Weitere Informationen zu Konfigurationsparametern finden Sie im Handbuch *Systemverwaltung: Optimierung*.

Aktivieren der partitionsinternen Abfrageparallelität

Zur Einrichtung der partitionsinternen Abfrageparallelität müssen Sie Konfigurationsparameter der Datenbank und des Datenbankmanagers ändern.

INTRA_PARALLEL

Konfigurationsparameter des Datenbankmanagers. Weitere Informationen zu diesem Parameter finden Sie im Handbuch *Systemverwaltung: Optimierung*.

DFT_DEGREE

Konfigurationsparameter der Datenbank. Definiert den Standardwert für die Bindeoption DEGREE und das Sonderregister CURRENT DEGREE. Weitere Informationen zu diesem Parameter finden Sie im Handbuch *Systemverwaltung: Optimierung*.

DEGREE

Option für den Vorcompiler bzw. das Binden von statischem SQL. Weitere Informationen finden Sie im Handbuch *Command Reference*.

CURRENT DEGREE

Sonderregister für dynamisches SQL. Weitere Informationen finden Sie im Handbuch *SQL Reference*.

Weitere Informationen zu den Einstellungen der Konfigurationsparameter und zur Einrichtung von Anwendungen für die Parallelverarbeitung finden Sie im Abschnitt zum Konfigurieren von DB2 im Handbuch *Systemverwaltung: Optimierung*.

Aktivieren der partitionsübergreifenden Abfrageparallelität

Partitionsübergreifende Parallelität tritt entsprechend der Anzahl von Datenbankpartitionen und der Verteilung von Daten auf diese Partitionen automatisch auf.

Aktivieren der Dienstprogrammparallelität

Dieser Abschnitt enthält eine Übersicht über die Aktivierung der partitionsinternen Parallelität für folgende Dienstprogramme:

- LOAD
- CREATE INDEX
- BACKUP DATABASE / TABLE SPACE
- RESTORE DATABASE / TABLE SPACE

Partitionsübergreifende Parallelität für diese Dienstprogramme tritt entsprechend der Anzahl von Datenbankpartitionen automatisch auf.

LOAD: Das Dienstprogramm LOAD nutzt Parallelität automatisch, Sie können aber auch die folgenden Parameter im Befehl LOAD verwenden:

- CPU_PARALLELISM
- DISK_PARALLELISM

Informationen zum Befehl LOAD finden Sie im Handbuch *Versetzen von Daten Dienstprogramme und Referenz*.

AutoLoader: Sie können mehrere Trennprozesse (SPLIT) für das Dienstprogramm AutoLoader angeben, indem Sie den Parameter MODIFIED BY ANYORDER für die Angabe LOAD in der Datei `auto1oader.cfg` angeben. Weitere Informationen finden Sie in *Versetzen von Daten Dienstprogramme und Referenz*.

CREATE INDEX: Für die Aktivierung der Parallelität bei der Erstellung eines Index gilt:

- Der Konfigurationsparameter INTRA_PARALLEL des Datenbankmanagers muss aktiviert (ON) sein.
- Die Tabelle muss ausreichend groß sein, um die Parallelität vorteilhaft nutzen zu können.
- Auf einer SMP-Maschine müssen mehrere Prozessoren aktiviert sein.

Im Handbuch *SQL Reference* finden Sie Informationen zur Anweisung CREATE INDEX.

Ermöglichen der Datenpartitionierung

Bei der Ausführung in einer partitionierten Datenbankumgebung können Sie eine Datenbank von jedem Knoten aus erstellen, der in der Datei `db2nodes.cfg` vorhanden ist. Verwenden Sie hierzu den Befehl CREATE DATABASE oder die Anwendungsprogrammierschnittstelle (API) `sqlcrea()`. Weitere Informationen finden Sie in den Handbüchern *Command Reference* und *Administrative API Reference*.

Bevor Sie eine partitionierte Datenbank erstellen, müssen Sie bestimmen, ob Sie als ein lokaler oder ein ferner Client eine Verbindung zu dem Exemplar herstellen, in dem die Datenbank erstellt werden soll. Als Nächstes müssen Sie die Verbindung (mit ATTACH) zu dem Exemplar herstellen. Darüber hinaus müssen Sie festlegen, welche Datenbankpartition der Katalogknoten für die Datenbank werden soll. Die Datenbankpartition, zu der Sie die Verbindung (mit ATTACH) herstellen und den Befehl CREATE DATABASE auswählen, wird zum *Katalogknoten* für diese spezielle Datenbank.

Der Katalogknoten ist die Datenbankpartition, in der alle Systemkatalogtabellen gespeichert werden. Jeglicher Zugriff auf die Systemtabellen muss über diese Datenbankpartition erfolgen. Alle Objekte zusammenschlossener Datenbanken (Oberflächen, Server, Kurznamen usw.) werden in den Systemkatalogtabellen auf diesem Knoten gespeichert. Wenn möglich, sollten Sie jede Datenbank in einem getrennten Exemplar erstellen. Falls dies nicht möglich ist (d. h., Sie müssen mehr als eine Datenbank pro Exemplar erstellen), sollten Sie die Katalogknoten auf die verfügbaren Datenbankpartitionen verteilen. Dadurch verringern sich Konkurrenzsituationen beim Abrufen von Katalogdaten in einer einzelnen Datenbankpartition.

Anmerkung: Sie sollten regelmäßig eine Sicherung des Katalogknotens durchführen und nach Möglichkeit vermeiden, Benutzerdaten auf ihm zu speichern, da diese Daten die für die Sicherung benötigte Zeit verlängern.

Wenn Sie eine Datenbank erstellen, wird sie automatisch in allen Datenbankpartitionen erstellt, die in der Datei `db2nodes.cfg` definiert sind.

Wenn die erste Datenbank im System erstellt wird, wird ein Systemdatenbankverzeichnis gebildet. An dieses werden Informationen zu anderen Datenbanken, die Sie erstellen, angehängt. Das Systemdatenbankverzeichnis heißt `sqlbdbir` und befindet sich im Verzeichnis `sql1ib` unter Ihrem Benutzerverzeichnis. Dieses Verzeichnis muss sich in einem gemeinsam benutzten Dateisystem (z. B. NFS auf UNIX-Plattformen) befinden, weil es nur ein Systemdatenbankverzeichnis für alle Datenbankpartitionen gibt, die zu einer partitionierten Datenbank gehören.

Ebenfalls im Verzeichnis `sqlbdbir` befindet sich die Systemintensionsdatei. Sie hat den Namen `sqldbins` und stellt sicher, dass die Datenbankpartitionen synchronisiert bleiben. Diese Datei muss ebenfalls in einem gemeinsam benutzten Dateisystem sein, da es innerhalb aller Datenbankpartitionen nur ein Verzeichnis gibt. Die Datei wird von allen Partitionen, die die Datenbank bilden, gemeinsam benutzt.

Zur Nutzung der Datenpartitionierung müssen Konfigurationsparameter geändert werden. Mit Hilfe der Befehle `GET DATABASE CONFIGURATION` und `GET DATABASE MANAGER CONFIGURATION` können Sie die Werte für einzelne Einträge in einer bestimmten Datenbank oder in der Konfigurationsdatei des Datenbankmanagers ermitteln. Zur Änderung einzelner Einträge in einer bestimmten Datenbank oder in der Konfigurationsdatei des Datenbankmanagers werden die Befehle `UPDATE DATABASE CONFIGURATION` bzw. `UPDATE DATABASE MANAGER CONFIGURATION` verwendet.

Zu den Konfigurationsparametern des Datenbankmanagers mit Auswirkung auf eine partitionierte Datenbank gehören `conn_elapse`, `fc_m_num_anchors`, `fc_m_n`

um_buffers, fcm_num_connect, fcm_num_rqb, max_connretries, max_coordagents, max_time_diff, num_poolagents und *stop_start_time*.

Weitere Informationen zu Konfigurationsparametern finden Sie im Handbuch *Systemverwaltung: Optimierung*.

BACKUP DATABASE / TABLE SPACE

Gehen Sie wie folgt vor, um die E/A-Parallelität für die Sicherung einer Datenbank oder eines Tabellenbereichs zu aktivieren:

- Verwenden Sie mehr als einen Zieldatenträger.
- Konfigurieren Sie Tabellenbereiche für parallele E/A.
- Verwenden Sie den Parameter PARALLELISM im Befehl BACKUP, um den Grad der Parallelität anzugeben.
- Verwenden Sie den Parameter WITH anzahl-puffer BUFFERS im Befehl BACKUP, damit genügend Pufferspeicher für den Grad der Parallelität vorhanden ist. Die Anzahl der Puffer sollte der Summe aus der Anzahl Ihrer Zieldatenträger und dem ausgewählten Grad der Parallelität plus einigen zusätzlichen Puffern entsprechen.

Verwenden Sie ferner einen Sicherungspuffer, für dessen Größe Folgendes gilt:

- So groß wie möglich. 4 MB oder 8 MB (1024 oder 2048 Seiten) ist eine zweckmäßige Faustregel.
- Mindestens so groß wie das größte Produkt ($\text{EXTENTSIZE} * \text{Anzahl der Behälter}$) der zu sichernden Tabellenbereiche.

Im Handbuch *Command Reference* finden Sie Informationen zum Befehl BACKUP DATABASE.

RESTORE DATABASE / TABLE SPACE

Gehen Sie wie folgt vor, um die E/A-Parallelität für das Wiederherstellen einer Datenbank oder eines Tabellenbereichs zu aktivieren:

- Verwenden Sie mehr als einen Quelledatenträger.
- Konfigurieren Sie Tabellenbereiche für parallele E/A.
- Verwenden Sie den Parameter PARALLELISM im Befehl RESTORE, um den Grad der Parallelität anzugeben.
- Verwenden Sie den Parameter WITH anzahl-puffer BUFFERS im Befehl RESTORE, damit genügend Pufferspeicher für den Grad der Parallelität vorhanden ist. Die Anzahl der Puffer sollte der Summe aus der Anzahl Ihrer Zieldatenträger und dem ausgewählten Grad der Parallelität plus einigen zusätzlichen Puffern entsprechen.

Verwenden Sie ferner einen Wiederherstellungspuffer, für dessen Größe Folgendes gilt:

- So groß wie möglich. 4 MB oder 8 MB (1024 oder 2048 Seiten) ist eine zweckmäßige Faustregel.
- Mindestens so groß wie das größte Produkt (EXTENTSIZE * Anzahl der Behälter) der wiederherzustellenden Tabellenbereiche.
- Dieselbe Größe oder ein gerades Vielfaches der Größe des Sicherungspuffers.

Im Handbuch *Command Reference* finden Sie Informationen zum Befehl RESTORE DATABASE.

Stoppen von DB2

Der Befehl **db2stop** kann nur auf dem Server ausgeführt werden. Bei Ausführung dieses Befehls sind keine Datenbankverbindungen zulässig. Wenn es Exemplarverbindungen (mit ATTACH) gibt, werden diese zwangsweise getrennt, bevor DB2 gestoppt wird.

Anmerkung: Wenn Sitzungen des Befehlszeilenprozessors einem Exemplar zugeordnet sind, müssen Sie jede dieser Sitzungen einzeln durch Ausführen des Befehls **terminate** beenden, bevor Sie den Befehl **db2stop** ausführen. Der Befehl **db2stop** stoppt das durch die Umgebungsvariable DB2INSTANCE definierte Exemplar.

Zum Stoppen eines DB2-Exemplars auf Ihrem System sind folgende Schritte erforderlich:

1. Melden Sie sich mit einer Benutzer-ID oder einem Benutzernamen an, der über die Berechtigung SYSADM, SYSCTRL oder SYSMAINT verfügt, an das Exemplar an bzw. stellen Sie die Verbindung (Attach) zu dem Exemplar her. Oder melden Sie sich als Exemplareigner an.
2. Zeigen Sie alle Anwendungen und Benutzer an, die mit der Datenbank, die Sie stoppen wollen, verbunden sind. Lassen Sie eine Liste der Anwendungen anzeigen, um sicherzugehen, dass keine wichtigen oder kritischen Anwendungen aktiv sind. Hierfür benötigen Sie die Berechtigung SYSADM, SYSCTRL oder SYSMAINT.
3. Erzwingen Sie die Trennung aller Anwendungen und Benutzer von der Datenbank. Für das Erzwingen der Trennung von Benutzern benötigen Sie die Berechtigung SYSADM oder SYSCTRL.
4. Führen Sie unter UNIX-Betriebssystemen die Startprozedur wie folgt aus:


```

      . INSTHOME/sql1lib/db2profile      (für Bourne- oder Korn-Shell)
      source INSTHOME/sql1lib/db2cshrc  (für C-Shell)
      
```

Dabei ist INSTHOME das Benutzerverzeichnis des Exemplars, das Sie verwenden wollen.

5. Verwenden Sie eine der folgenden Methoden, um das Exemplar zu stoppen:

- a. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Exemplare** angezeigt wird.
- b. Klicken Sie auf allen Exemplaren, die gestoppt werden sollen.
- c. Klicken Sie mit der rechten Maustaste auf einem der ausgewählten Exemplare und wählen Sie dann im Kontextmenü **Stoppen** aus.
- d. Klicken Sie im Fenster 'Stoppen bestätigen' auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um das Exemplar zu stoppen:

```
db2stop
```

Details beim Erstellen einer Datenbank

Vor dem Erstellen einer Datenbank sollten Sie folgende Punkte in Betracht ziehen bzw. ausführen:

- Entwerfen logischer und physischer Datenbankmerkmale
- Erstellen eines Exemplars
- Einrichten der Umgebungsvariablen und der Profilregistrierdatenbank
- Erstellen eines DB2-Verwaltungsservers (DAS)
- Erstellen einer Knotenkonfigurationsdatei
- Erstellen der Datenbankkonfigurationsdatei
- Replizieren von Konfigurationsdaten durch Antwortdateien
- Aktivieren der FCM-Kommunikation

Entwerfen logischer und physischer Datenbankmerkmale

Sie müssen bestimmte Entscheidungen zum logischen und physischen Aufbau der Datenbank treffen, bevor Sie eine Datenbank erstellen. Weitere Informationen zum logischen und physischen Datenbankdesign finden Sie im Handbuch *Systemverwaltung: Konzept*.

Erstellen eines Exemplars

Ein Exemplar ist ein logische Datenbankmanagerumgebung, in der Sie Datenbanken katalogisieren und Konfigurationsparameter setzen. Bei Bedarf können Sie mehrere Exemplare erstellen. Mehrere Exemplare können beispielsweise zu folgenden Zwecken wünschenswert sein:

- Betreiben eines Exemplars für die Entwicklungsumgebung und ein weiteres für die Geschäftsumgebung
- Anpassen eines Exemplars für eine bestimmte Umgebung
- Begrenzen des Zugriffs auf sensible Daten
- Steuern der Erteilung der Berechtigung SYSADM, SYSCTRL, und SYSMAINT für jedes Exemplar
- Optimieren der Datenbankmanagerkonfiguration für jedes Exemplar

- Begrenzen der durch den Ausfall eines Exemplars entstehenden Auswirkungen, d. h. bei einem Ausfall ist nur jeweils ein Exemplar betroffen und die übrigen Exemplare funktionieren weiterhin ordnungsgemäß.

Beim Arbeiten mit mehreren Exemplaren ergeben sich allerdings auch die folgenden geringfügigen Nachteile:

- Für jedes Exemplar sind zusätzliche Systemressourcen (virtueller Speicher und Plattenspeicherplatz) erforderlich.
- Der Verwaltungsaufwand ist größer, da mehrere Exemplare zu verwalten sind.

Im Exemplarverzeichnis werden alle Informationen für ein Datenbankexemplar gespeichert. Sobald das Exemplarverzeichnis erstellt ist, können Sie dessen Position nicht mehr ändern. Das Verzeichnis enthält Folgendes:

- Konfigurationsdatei des Datenbankmanagers
- Systemdatenbankverzeichnis
- Knotenverzeichnis
- DB2-Diagnosedatei (`db2diag.log`)
- Knotenkonfigurationsdatei (`db2nodes.cfg`)
- Alle weiteren Dateien mit Debug-Informationen, wie z. B. den Speicherausgang der Ausnahme- und Registrierdaten oder den Aufrufstapel für die DB2-Prozesse.

Unter UNIX-Betriebssystemen befindet sich das Exemplarverzeichnis im Verzeichnis `INSTHOME/sql1ib`, wobei `INSTHOME` das Benutzerverzeichnis des Exemplareigners ist.

In Systemen mit partitionierten Datenbanken wird das Exemplarverzeichnis von allen Datenbankpartitionsservern, die zu dem Exemplar gehören, gemeinsam benutzt. Darum muss das Exemplarverzeichnis auf einem gemeinsam benutzten Netzwerklaufwerk erstellt werden, auf das alle Maschinen des Exemplars zugreifen können.

Als Teil Ihres Installationsverfahrens erstellen Sie ein Anfangsexemplar von DB2 mit dem Namen „DB2“. Unter UNIX kann dem Anfangsexemplar jeder beliebige Name zugeordnet werden, der den geltenden Namenskonventionen entspricht. Der Exemplarname wird zum Definieren der Verzeichnisstruktur verwendet.

Damit dieses Exemplar sofort verwendet werden kann, wird bei der Installation Folgendes festgelegt:

- Die Umgebungsvariable DB2INSTANCE wird auf „DB2“ gesetzt.
- Die Variable der DB2-Registrierdatenbank DB2INSTDEF wird auf „DB2“ gesetzt.

Unter UNIX kann dem Standardexemplar jeder beliebige Name zugeordnet werden, der den geltenden Namenskonventionen entspricht.

Diese Einstellungen legen „DB2“ als das Standardexemplar fest. Sie können das Standardexemplar erst ändern, nachdem Sie ein weiteres Exemplar erstellt haben.

Vor dem Arbeiten mit DB2 muss die Datenbankumgebung für jeden Benutzer aktualisiert werden, so dass sie auf ein Exemplar zugreifen und die DB2-Programme ausführen kann. Dies gilt für alle Benutzer (einschließlich der Benutzer mit Verwaltungsaufgaben).

Auf UNIX-Betriebssystemen werden Beispielprozedurdateien bereitgestellt, die Sie beim Einrichten der Datenbankumgebung unterstützen. Folgende Dateien werden bereitgestellt: `db2profile` für Bourne- oder Korn-Shell und `db2cshrc` für C-Shell. Diese Prozeduren befinden sich im Unterverzeichnis `sql1lib` im Benutzerverzeichnis des Exemplareigners. Der Exemplareigner oder jeder beliebige Benutzer, der zur SYSADM-Gruppe des Exemplars gehört, kann die Prozedur für alle Benutzer eines Exemplars anpassen. Die Prozedur kann stattdessen auch kopiert und für jeden Benutzer separat angepasst werden.

Die Beispielprozedur enthält Anweisungen für folgende Aufgaben:

- Aktualisieren eines Benutzerpfads (PATH) durch Hinzufügen der folgenden Verzeichnisse zum vorhandenen Suchpfad: die Unterverzeichnisse `bin`, `adm` und `misc` unter `sql1lib` im Benutzerverzeichnis des Exemplareigners
- Setzen der Umgebungsvariablen DB2INSTANCE auf den Exemplarnamen

Automatisches Einrichten der DB2-Umgebung

Anmerkung: Diese Angaben gelten nur für Umgebungen mit UNIX-Betriebssystemen.

Standardmäßig wirken sich die Prozeduren nur für die Dauer der aktuellen Sitzung auf die Benutzerumgebung aus. Sie können die Datei `.profile` ändern, damit Sie die Prozedur `db2profile` automatisch ausführt, wenn sich der Benutzer über die Bourne- oder Korn-Shell anmeldet. Für Benutzer der C-Shell können Sie die Datei `.login` so modifizieren, dass sie die Prozedurdatei `db2shrc` ausführt.

Fügen Sie den Prozedurdateien `.profile` oder `.login` eine der folgenden Anweisungen hinzu:

- Fügen Sie für Benutzer, die eine Version der Prozedur gemeinsam benutzen, Folgendes hinzu:

```
. INSTHOME/sql1lib/db2profile    (für Bourne- oder Korn-Shell)
source INSTHOME/sql1lib/db2cshrc (für C-Shell)
```

Dabei ist `INSTHOME` das Benutzerverzeichnis des Exemplars, das Sie verwenden wollen.

- Fügen Sie für Benutzer, die eine angepasste Version der Prozedur in ihrem Benutzerverzeichnis verwenden, Folgendes hinzu:

```
. USERHOME/db2profile          (für Bourne- oder Korn-Shell)
source USERHOME/db2cshrc       (in C-Shell)
```

Dabei ist `USERHOME` das Benutzerverzeichnis des Benutzers.

Manuelles Einrichten der DB2-Umgebung

Anmerkung: Diese Angaben gelten nur für Umgebungen mit UNIX-Betriebssystemen.

Geben Sie nach der Eingabeaufforderung eine der folgenden Anweisungen ein, um das zu verwendende Exemplar auszuwählen. Der Punkt (`.`) und das Leerzeichen sind erforderliche Angaben.

- Fügen Sie für Benutzer, die eine Version der Prozedur gemeinsam benutzen, Folgendes hinzu:

```
. INSTHOME/sql1lib/db2profile    (für Bourne- oder Korn-Shell)
source INSTHOME/sql1lib/db2cshrc (für C-Shell)
```

Dabei ist `INSTHOME` das Benutzerverzeichnis des Exemplars, das Sie verwenden wollen.

- Fügen Sie für Benutzer, die eine angepasste Version der Prozedur in ihrem Benutzerverzeichnis verwenden, Folgendes hinzu:

```
. USERHOME/db2profile          (für Bourne- oder Korn-Shell)
source USERHOME/db2cshrc       (in C-Shell)
```

Dabei ist `USERHOME` das Benutzerverzeichnis des Benutzers.

Wenn Sie mehrere Exemplare zugleich verwenden wollen, führen Sie die Prozedur für jedes Exemplar, das Sie verwenden wollen, in einem separaten Fenster aus. Angenommen, Sie verfügen über die beiden Exemplare `test` und `prod` mit den Benutzerverzeichnissen `/u/test` und `/u/prod`.

Fenster 1:

- Geben Sie in einer Bourne- oder Korn-Shell Folgendes ein:
 `./u/test/sqllib/db2profile`
- Geben Sie in einer C-Shell Folgendes ein:
 `source /u/test/sqllib/db2cshrc`

Fenster 2:

- Geben Sie in einer Bourne- oder Korn-Shell Folgendes ein:
 `./u/prod/sqllib/db2profile`
- Geben Sie in einer C-Shell Folgendes ein:
 `source /u/prod/sqllib/db2cshrc`

Verwenden Sie das Fenster 1, um mit dem Exemplar `test` zu arbeiten, und das Fenster 2, um mit dem Exemplar `prod` zu arbeiten.

Anmerkung: Geben Sie den Befehl **which db2** ein, um sicherzustellen, dass Ihr Suchpfad korrekt eingerichtet ist. Dieser Befehl gibt den absoluten Pfad der ausführbaren Datei des DB2 CLPs zurück. Vergewissern Sie sich, dass sich diese Datei im Unterverzeichnis `sqllib` des Exemplars befindet.

Mehrere Exemplare auf einem System

Es ist möglich, mehr als ein Exemplar auf einem System zu betreiben. Sie können jedoch nur jeweils innerhalb eines Exemplars von DB2 gleichzeitig arbeiten.

Jedem Exemplar wird der Exemplareigner und die Systemverwaltungsgruppe (SYSADM) zugeordnet. Dies geschieht bei der Erstellung des Exemplars. Eine Benutzer-ID bzw. ein Benutzername kann nur für ein Exemplar verwendet werden. Diese Benutzer-ID bzw. dieser Benutzername wird auch als der *Exemplareigner* bezeichnet.

Jeder Exemplareigner muss über ein eindeutiges Benutzerverzeichnis verfügen. Alle erforderlichen Dateien zum Ausführen des Exemplars werden im Benutzerverzeichnis der Benutzer-ID bzw. des Benutzernamens des Exemplareigners erstellt. Falls es erforderlich wird, die Benutzer-ID bzw. den Benutzernamen des Exemplareigners aus dem System zu entfernen, könnten Sie dem Exemplar zugeordnete Dateien sowie den Zugriff auf in diesem Exemplar gespeicherte Daten verlieren. Daher ist es empfehlenswert, die Benutzer-ID bzw. den Benutzernamen eines Exemplareigners ausschließlich für die Ausführung von DB2 zu dedizieren.

Die Primärgruppe des Exemplareigners ist ebenfalls von Bedeutung. Diese Primärgruppe wird automatisch zur Systemverwaltungsgruppe des Exemplars und erhält die Berechtigung SYSADM für das Exemplar. Andere Benutzer-IDs oder Benutzernamen, die Mitglieder der Primärgruppe des Exemplars sind, erhalten ebenfalls diese Berechtigungsstufe. Aus diesem Grund kann es wünschenswert sein, die Benutzer-ID bzw. den Benutzernamen des Exemplareigners einer Primärgruppe zuzuordnen, die für die Verwaltung von Exemplaren reserviert ist. (Stellen Sie außerdem sicher, dass der Benutzer-ID bzw. dem Benutzernamen des Exemplareigners eine Primärgruppe zugeordnet ist. Andernfalls wird die Standardprimärgruppe des Systems verwendet.)

Wenn Sie bereits über eine Gruppe verfügen, die Sie als Systemverwaltungsgruppe für das Exemplar verwenden wollen, können Sie diese Gruppe einfach beim Erstellen der Benutzer-ID bzw. des Benutzernamens des Exemplareigners als Primärgruppe zuordnen. Fügen Sie andere Benutzer, denen Sie die Verwaltungsberechtigung für das Exemplar erteilen wollen, derjenigen Gruppe hinzu, die als Systemverwaltungsgruppe zugeordnet ist.

Wenn Sie die Berechtigung SYSADM für die einzelnen Exemplare voneinander trennen wollen, stellen Sie sicher, dass jede Benutzer-ID bzw. jeder Benutzername eines Exemplareigners eine andere Primärgruppe verwendet. Bei Verwendung einer allgemeinen Berechtigung SYSADM für mehrere Exemplare können Sie jedoch dieselbe Primärgruppe für mehrere Exemplare verwenden.

Hinzufügen eines Exemplars

Wenn Sie über Administratorberechtigung für OS/2 verfügen, der Administratorengruppe unter Windows NT angehören oder die Root-Berechtigung für UNIX-Plattformen besitzen, können Sie zusätzliche DB2-Exemplare hinzufügen. Die Maschine, auf der das Exemplar hinzugefügt wird, wird als Exemplareignermaschine (Knoten Null) definiert. Fügen Sie Exemplare unbedingt auf einer Maschine hinzu, auf der sich ein Verwaltungsserver befindet.

Führen Sie zum Hinzufügen eines zusätzlichen Exemplars folgende Schritte aus:

1. Melden Sie sich mit einer Benutzer-ID oder einem Benutzernamen an, die bzw. der über Administratorberechtigung verfügt oder der lokalen Administratorengruppe angehört.

2. Verwenden Sie zum Hinzufügen eines Exemplars eine der folgenden Methoden.

Gehen Sie wie folgt vor, um die Steuerzentrale zu verwenden:

- a. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Exemplare** des gewünschten Systems angezeigt wird.
- b. Klicken Sie mit der rechten Maustaste auf dem Exemplarordner und wählen Sie dann im Kontextmenü **Hinzufügen** aus.
- c. Geben Sie die erforderlichen Informationen ein und klicken Sie anschließend auf **Anwenden**.

Geben Sie in der Befehlszeile Folgendes ein, um ein Exemplar hinzuzufügen:

```
db2icrt <exemplarname>
```

3. Erstellen Sie einen Verwaltungsserver.

Bei der Verwendung des Befehls **db2icrt** zum Hinzufügen eines zusätzlichen DB2-Exemplars sollten Sie den Anmeldenamen des Exemplareigners und wahlfrei die Authentifizierungsart des Exemplars angeben. Die Authentifizierungsart gilt für alle Datenbanken, die unter diesem Exemplar erstellt werden. Die Authentifizierungsart ist eine Anweisung, wo die Authentifizierung von Benutzern stattfinden soll. Weitere Informationen zur Authentifizierung finden Sie in „Kapitel 5. Steuern des Datenbankzugriffs“ auf Seite 257.

Anmerkung: Mit dem Befehl **db2iupdt** können Sie die vorhandene Exemplarkonfiguration aktualisieren.

Mit der Umgebungsvariablen DB2INSTPROF können Sie die Position des in DB2PATH angegebenen Exemplarverzeichnis ändern. Sie benötigen dafür den Schreibzugriff für das Exemplarverzeichnis. Wenn die Verzeichnisse in einem anderen Pfad als DB2PATH erstellt werden sollen, müssen Sie DB2INSTPROF setzen, **BEVOR** Sie den Befehl **db2icrt** eingeben.

DB2 Enterprise - Extended Edition: Details zum Hinzufügen von Exemplaren: Bei Verwendung von DB2 Universal Database Enterprise - Extended Edition müssen Sie außerdem deklarieren, dass Sie ein neues Exemplar hinzufügen, bei dem es sich um ein partitioniertes Datenbanksystem handelt. Verwenden Sie dazu in der Befehlszeile den Parameter `-s eee`.

UNIX - Details zum Erstellen von Exemplaren: Beim Arbeiten mit UNIX-Betriebssystemen verfügt der Befehl **db2icrt** über die folgenden wahlfreien Parameter:

- -h oder -?

Dieser Parameter zeigt ein Hilfemenü für den Befehl an.

- -d

Dieser Parameter konfiguriert den Debug-Modus für die Fehlerbestimmung.

- -a IdentArt

Dieser Parameter gibt die Authentifizierungsart für das Exemplar an. Gültige Authentifizierungsarten sind SERVER, CLIENT, DCS und DCE. Wenn dieser Parameter nicht angegeben wird, wird standardmäßig die Authentifizierungsart SERVER verwendet, wenn ein DB2-Server installiert ist. Ansonsten wird die Option CLIENT verwendet.

Anmerkungen:

1. Die für das Exemplar angegebene Authentifizierungsart gilt für alle Datenbanken, die zu diesem Exemplar gehören.
2. Für UNIX-Betriebssysteme ist die Authentifizierungsart DCE keine zulässige Option.

- -u AbgeschirmtID

Dieser Parameter gibt den Benutzer an, unter dem die abgeschirmten benutzerdefinierten Funktionen (UDFs) und die gespeicherten Prozeduren ausgeführt werden. Dies ist nicht erforderlich, wenn Sie den DB2-Client oder den DB2 Application Development Client installieren. Für andere DB2-Produkte ist dieser Parameter erforderlich.

Anmerkung: AbgeschirmtID kann nicht „root“ oder „bin“ sein.

- -p AnschlussName

Dieser Parameter gibt an, welcher TCP/IP-Servicename bzw. welche TCP/IP-Anschlussnummer verwendet werden soll. Dieser Wert wird anschließend in der Datenbankkonfigurationsdatei für jede Datenbank des Exemplars gesetzt.

- -s ExemplarArt

Ermöglicht das Hinzufügen verschiedener Exemplararten. Zulässige Exemplararten sind: ee, eee und client.

Beispiele:

- Zum Hinzufügen eines Exemplars für einen DB2-Server können Sie folgenden Befehl verwenden:

```
db2icrt -u db2fenc1 db2inst1
```

- Wenn Sie nur DB2 Connect Enterprise Edition installiert haben, können Sie den Exemplarnamen wie folgt als abgeschirmte ID verwenden:

```
db2icrt -u db2inst1 db2inst1
```

- Zum Hinzufügen eines Exemplars für einen DB2-Client können Sie folgenden Befehl verwenden:

```
db2icrt db2inst1 -s client -u fencedID
```

DB2-Client-Exemplare werden erstellt, wenn eine Workstation mit anderen Datenbankservern verbunden werden soll und auf der Workstation selbst keine lokale Datenbank benötigt wird.

Windows NT - Details zum Erstellen von Exemplaren: Beim Arbeiten mit dem Betriebssystem Windows NT verfügt der Befehl **db2icrt** über die folgenden wahlfreien Parameter:

- `-s ExemplarArt`

Ermöglicht das Erstellen verschiedener Exemplararten. Zulässige Exemplararten sind: `ee`, `eee` und `client`.

- `/p:ExemplarProf_Pfad`

Dies ist ein wahlfreier Parameter zur Angabe eines anderen Exemplarprofilpfads. Wenn Sie den Pfad nicht angeben wird das Exemplarverzeichnis als Unterverzeichnis von `SQLLIB` erstellt und erhält den gemeinsam benutzten Namen `DB2` verknüpft mit dem Exemplarnamen. Jedem Benutzer in der Domäne werden automatisch Lese- und Schreibberechtigungen erteilt. Die Berechtigungen können geändert werden, um den Zugriff auf das Verzeichnis einzuschränken.

Wenn Sie einen anderen Exemplarprofilpfad angeben, müssen Sie ein gemeinsames Laufwerk oder Verzeichnis erstellen. Dadurch hat jeder in der Domäne Zugriff auf das Exemplarverzeichnis, solange die Berechtigungen nicht geändert werden.

- `/u:benutzername,kennwort`

Beim Erstellen einer Umgebung mit partitionierten Datenbanken müssen Sie den Anmelde- und den Kontonamen sowie das Kennwort des DB2-Services deklarieren.

- `/r:Basis_Anschluss,End_Anschluss`

Dies ist ein wahlfreier Parameter zum Angeben des TCP/IP-Anschlussbereichs für den FCM (Fast Communications Manager). Beim Angeben des TCP/IP-Anschlussbereichs müssen Sie sicherstellen, dass der Anschlussbereich auf allen Maschinen des Partitionsdatenbanksystems verfügbar ist.

Unter DB2 für Windows NT Enterprise - Extended Edition könnten Sie beispielsweise Folgendes eingeben:

```
db2icrt exempl -s eee
/p:\maschineA\db2mpp
/u:ihrname,ihrkennwort /r:9010,9015
```

Anmerkung: Der Befehl **db2icrt** erteilt dem zum Erstellen des Exemplars verwendeten Benutzernamen die Berechtigungen zum Ausführen folgender Operationen:

- Verarbeiten von Prozessen auf Betriebssystemebene
- Erstellen eines Token-Objekts
- Anheben von Quoten
- Anmelden als Service
- Ersetzen eines Tokens auf Prozessebene

Das Exemplar benötigt diese Benutzerberechtigungen zum Zugreifen auf das gemeinsame Laufwerk, zur Authentifizierung des Benutzereintrags und zum Ausführen von DB2 als Windows NT-Dienst.

Auflisten der Exemplare

Gehen Sie wie folgt vor, um mit Hilfe der Steuerzentrale eine Liste aller auf dem System verfügbaren Exemplare abzurufen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Exemplare** für das System angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Exemplarordner, und wählen Sie dann im Kontextmenü **Hinzufügen** aus.
3. Klicken Sie im Fenster „Datenbank hinzufügen“ auf **Aktualisieren**.
4. Klicken Sie auf den Pfeil für die verdeckte Liste, um eine Liste der Datenbankexemplare anzuzeigen.
5. Klicken Sie auf **Abbruch**, um das Fenster zu schließen.

Geben Sie in der Befehlszeile Folgendes ein, um alle auf dem System verfügbaren Exemplare aufzulisten:

```
db2ilist
```

Geben Sie Folgendes ein, um zu ermitteln, welches Exemplar für die aktuelle Sitzung relevant ist:

```
set db2instance
```

Anmerkung: Geben Sie unter UNIX-Betriebssystemen Folgendes ein:

```
db2 get instance
```

Festlegen des aktuellen Exemplars

Wenn Sie einen Befehl zum Starten oder Stoppen des Datenbankmanagers eines Exemplars ausführen, wendet DB2 diesen Befehl auf das aktuelle Exemplar an. DB2 ermittelt das aktuelle Exemplar wie folgt:

- Wenn die Umgebungsvariable DB2INSTANCE für die aktuelle Sitzung gesetzt ist, gibt ihr Wert das aktuelle Exemplar an. Geben Sie Folgendes ein, um die Umgebungsvariable DB2INSTANCE zu setzen:

```
set db2instance=<neuer_exemplarname>
```

- Wenn die Umgebungsvariable DB2INSTANCE für die aktuelle Sitzung nicht gesetzt ist, verwendet DB2 die Einstellung der Umgebungsvariablen DB2INSTANCE aus den Systemumgebungsvariablen. Unter Windows NT werden die Systemumgebungsvariablen in der Systemumgebung gesetzt. Unter Windows 95 werden sie in der Datei autoexec.bat gesetzt. Unter OS/2 werden sie in der Datei config.sys gesetzt.

- Wenn die Umgebungsvariable DB2INSTANCE überhaupt nicht gesetzt ist, verwendet DB2 die Variable DB2INSTDEF der Profilregistrierdatenbank. Geben Sie Folgendes ein, um die Variable DB2INSTDEF der Profilregistrierdatenbank auf der globalen Ebene der Registrierdatenbank zu setzen:

```
db2set db2instdef=<neuer_exemplarname> -g
```

Automatisches Starten von Exemplaren

Geben Sie unter UNIX-Betriebssystemen den folgenden Befehl ein, damit ein Exemplar nach jedem Neustart des Systems automatisch gestartet wird:

```
db2iauto -on ExempName
```

Dabei ist ExempName der Anmeldename des Exemplars.

Geben Sie unter UNIX-Betriebssystemen den folgenden Befehl ein, um das automatische Starten eines Exemplars nach jedem Neustart des Systems zu verhindern:

```
db2iauto -off ExempName
```

Dabei ist ExempName der Anmeldename des Exemplars.

Gleichzeitiges Ausführen mehrerer Exemplare

Sie können mehrere DB2-Exemplare starten, sofern diese die gleiche Codestufe verwenden.

Gehen Sie wie folgt vor, um mit der Steuerzentrale gleichzeitig mehrere Exemplare auszuführen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Datenbanken** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf einem Exemplar und wählen Sie dann im Kontextmenü **Starten** aus.
3. Wiederholen Sie Schritt 2 so oft, bis alle Exemplare gestartet sind, die gleichzeitig ausgeführt werden sollen.

Geben Sie in der Befehlszeile Folgendes ein, um gleichzeitig mehrere Exemplare auszuführen:

1. Setzen Sie die Variable DB2INSTANCE auf den Namen des anderen Exemplars, das Sie starten wollen, indem Sie Folgendes eingeben:

```
set db2instance=<name_anderes_exemplar>
```
2. Geben Sie den Befehl **db2start** ein, um das Exemplar zu starten.

Lizenzverwaltung

Die Verwaltung der Lizenzen für Ihre DB2-Produkte erfolgt im Wesentlichen durch die Lizenzzentrale, die Bestandteil der Steuerzentrale in der Online-Schnittstelle des jeweiligen Produkts ist. In der Lizenzzentrale können Sie Lizenzinformationen, Statistiken, registrierte Benutzer und aktuelle Benutzer für jedes installierte Produkt einsehen.

Einrichten der Umgebungsvariablen und der Profilregistrierdatenbank

Ihre Datenbankumgebung wird mit Hilfe von Umgebungsvariablen und Variablen der Profilregistrierdatenbank gesteuert.

Vor der Einführung der DB2-Profilregistrierdatenbank war die Änderung von Umgebungsvariablen auf Workstations unter Windows und OS/2 beispielsweise mit der Notwendigkeit verbunden, das System neu zu booten. Jetzt wird Ihre Umgebung mit wenigen Ausnahmen durch Variablen der Profilregistrierdatenbank gesteuert, die in den DB2-Profilregistrierdatenbanken gespeichert sind. Benutzer mit der Berechtigung zur Systemverwaltung (SYS-ADM) für ein bestimmtes Exemplar können die Registrierungswerte für das betreffende Exemplar aktualisieren. Zur Aktualisierung der Variablen der Profilregistrierdatenbank ohne Warmstart verwenden Sie den Befehl **db2set**. Diese Informationen werden unmittelbar in den Registrierdatenbanken gespeichert. Die DB2-Registrierdatenbank wendet die aktualisierten Informationen auf diejenigen DB2-Serverexemplare und DB2-Anwendungen an, die nach dem Vornehmen der Änderungen gestartet wurden.

Aktualisierungen der Registrierdatenbank wirken sich nicht auf die momentan ausgeführten DB2-Anwendungen oder -Benutzer aus. Nach der Aktualisierung gestartete Anwendungen verwenden die neuen Werte.

Anmerkung: Die DB2-Umgebungsvariablen DB2INSTANCE, DB2NODE, DB2PATH und DB2INSTPROF werden je nach Betriebssystem eventuell nicht in den DB2-Profilregistrierdatenbanken gespeichert. Zur Aktualisierung dieser Umgebungsvariablen muss der Befehl **set** verwendet werden. Diese Änderungen werden nach dem nächsten Neustart des Systems wirksam. Auf UNIX-Plattformen kann der Befehl **export** anstelle des Befehls **set** verwendet werden.

Die Verwendung der Profilregistrierdatenbank ermöglicht eine zentrale Steuerung der Umgebungsvariablen. Im Abschnitt über „DB2-Variablen der Profilregistrierdatenbank und Umgebungsvariablen“ im Handbuch *Systemverwaltung: Optimierung* sind viele der Umgebungsvariablen und Variablen der Profilregistrierdatenbank aufgelistet. Verschiedene Ebenen der Unterstützung werden nun durch verschiedene Umgebungsprofile realisiert. Eine Fernverwaltung der Umgebungsvariablen steht außerdem zur Verfügung, wenn der DB2-Verwaltungsserver (DAS) verwendet wird.

Es gibt vier Profilregistrierdatenbanken:

- Die DB2-Profilregistrierdatenbank auf Exemplarebene. Die meisten der DB2-Umgebungsvariablen werden in dieser Registrierdatenbank gespeichert. Die Einstellungen der Umgebungsvariablen für ein bestimmtes Exemplar werden in dieser Registrierdatenbank verwaltet. Die auf dieser Ebene definierten Werte überschreiben die Einstellungen der globalen Ebene.
- Die DB2-Profilregistrierdatenbank auf globaler Ebene. Wenn eine Umgebungsvariable nicht für ein bestimmtes Exemplar definiert wird, wird sie in dieser Registrierdatenbank gespeichert. Diese Registrierdatenbank enthält Umgebungsvariablen, deren Werte sich auf die gesamte Maschine beziehen. In DB2 UDB EEE gibt es auf jeder Maschine ein Profil der globalen Ebene.
- Die DB2-Profilregistrierdatenbank auf Exemplarknotenebene. In einem System, in dem die Datenbank über verschiedene Datenbankpartitionen verteilt ist, befindet sich diese Registrierdatenbank auf jedem Knoten (d. h. Maschine) und enthält Einstellungen von Umgebungsvariablen für alle Exemplare, die Daten auf dem Knoten speichern. Die auf dieser Ebene definierten Werte überschreiben vergleichbare Einstellungen auf der Exemplarebene und der globalen Ebene.
- Die DB2-Exemplarprofilregistrierdatenbank. Diese Registrierdatenbank enthält eine Liste aller Exemplarnamen, die von diesem System erkannt werden.

Benutzer können die Einstellung von Umgebungsvariablen der DB2-Exemplarprofilregistrierdatenbank für Ihre Sitzung außer Kraft setzen, indem sie die Einstellungen der Umgebungsvariablen für die Sitzung mit Hilfe des Befehls **set** (oder unter UNIX mit Hilfe des Befehls **export**) ändern.

DB2 konfiguriert die Betriebsumgebung, indem DB2 die Registrierungswerte und die Umgebungsvariablen überprüft und in folgender Reihenfolge auflöst:

1. Umgebungsvariablen, die mit dem Befehl **set** definiert wurden (oder mit dem Befehl **export** auf UNIX-Plattformen).
2. Registrierungswerte, die im Profil auf Exemplarknotenebene definiert sind (mit dem Befehl **db2set -i** mit einer Knotennummer, wie unten gezeigt).
3. Registrierungswerte, die im Profil auf Exemplebene definiert sind (mit dem Befehl **db2set -i**, wie unten gezeigt).
4. Registrierungswerte, die anhand des globalen Profils definiert sind (mit dem Befehl **db2set -g**, wie unten gezeigt).

Verwenden des Befehls **db2set**

Der Befehl **db2set** unterstützt die lokale Deklaration der Variablen der Profilregistrierdatenbank (und der Umgebungsvariablen).

Wenn Sie Hilfe für den Befehl benötigen, geben Sie Folgendes ein:

```
db2set ?
```

Zum Auflisten der vollständigen Gruppe aller unterstützten Variablen der Profilregistrierdatenbank geben Sie Folgendes ein:

```
db2set -lr
```

Verwenden Sie den folgenden Befehl, um alle definierten Variablen der Registrierdatenbank für das aktuelle oder das Standardexemplar aufzulisten:

```
db2set
```

Verwenden Sie den folgenden Befehl, um alle in der Profilregistrierdatenbank definierten Variablen der Profilregistrierdatenbank aufzulisten:

```
db2set -all
```

Zum Anzeigen des Wertes einer Variablen der Profilregistrierdatenbank geben Sie Folgendes ein:

```
db2set name_der_variablen_der_profilregistrierdatenbank
```

Zum Anzeigen des Wertes einer Variablen der Profilregistrierdatenbank auf allen Ebenen geben Sie Folgendes ein:

```
db2set name_der_variablen_der_profilregistrierdatenbank -all
```

Zum Löschen eines Variablenwerts auf einer bestimmten Ebene können Sie die gleiche Befehlssyntax wie zum Setzen der Variablen verwenden, jedoch ohne Angabe eines Variablenwerts. Geben Sie beispielsweise Folgendes ein, um die Einstellung einer Variablen auf der Knotenebene zu löschen:

```
db2set name_der_variablen_der_profilregistrierdatenbank -i exemplarname  
knotennummer
```

Geben Sie zum Löschen eines Variablenwerts und zum Begrenzen der Verwendung der Variablen Folgendes ein, wenn sie auf einer höheren Profilebene definiert ist:

```
db2set  
name_der_variablen_der_profilregistrierdatenbank= -null exemplarname
```

Dieser Befehl löscht die Einstellung für den von Ihnen angegebenen Parameter und hindert die Profile höherer Ebene (in diesem Fall das DB2-Profil der globalen Ebene) daran, den Wert dieser Variablen zu ändern. Die von Ihnen angegebene Variable kann jedoch weiterhin durch ein Profil niedrigerer Ebene (in diesem Fall das DB2-Profil der Knotenebene) gesetzt werden.

Zum Ändern einer Variablen der Profilregistrierdatenbank im aktuellen oder Standardexemplar geben Sie Folgendes ein:

```
db2set name_der_variablen_der_profilregistrierdatenbank=neuer_wert
```

Zum Ändern des Standardwerts einer Variablen der Profilregistrierdatenbank für alle Datenbanken des Exemplars geben Sie Folgendes ein:

```
db2set name_der_variablen_der_profilregistrierdatenbank=neuer_wert  
-i exemplarname
```

Zum Ändern des Standardwerts einer Variablen der Profilregistrierdatenbank für alle Exemplare im System geben Sie Folgendes ein:

```
db2set name_der_variablen_der_profilregistrierdatenbank=neuer_wert -g
```

Zum Setzen der Variablen der Profilregistrierdatenbank auf Benutzerebene geben Sie Folgendes ein:

```
db2set -u1
```

Zum Setzen der Variablen der Profilregistrierdatenbank auf Benutzerebene für einen bestimmten Benutzer geben Sie Folgendes ein:

```
db2set -u1 benutzername
```

Anmerkungen:

1. Die Parameter "-i", "-g" und "-ul" können nicht gleichzeitig im selben Befehl verwendet werden.
2. Manche Parameter verwenden standardmäßig immer das Profil der globalen Ebene. Sie können nicht in Profilen der Exemplar- oder Knotenebene gesetzt werden, z. B. db2system und db2instdef.
3. Unter UNIX müssen Sie über die Berechtigung zur Systemverwaltung (SYSADM) verfügen, um die Registrierungswerte für ein Exemplar zu ändern. Nur Benutzer mit Root-Berechtigung können Parameter in Registrierungen der globalen Ebene ändern.

Bei der Ausführung in einer LDAP-Umgebung können Sie einen Wert für eine DB2-Variable der Profilregistrierdatenbank so definieren, dass der Bereich für alle Maschinen und Benutzer, die zu einer Verzeichnispartmention oder einer Windows NT-Domäne gehören, global gültig ist. Momentan kann nur die DB2-Variable DB2LDAP_SEARCH_SCOPE der Profilregistrierdatenbank auf der globalen LDAP-Ebene definiert werden.

Verwenden Sie hierzu die Option -gl des Befehls db2set.

Anmerkung: Diese Option unterscheidet sich von der Option -g, die dazu verwendet wird, DB2-Variablen der Profilregistrierdatenbank auf globaler Maschinenebene zu definieren. Die Option -gl gilt speziell für die globale LDAP-Ebene. Darüber hinaus wird das Definieren dieser DB2-Variablen der Profilregistrierdatenbank in LDAP nur auf Windows-Plattformen unterstützt.

Zum Setzen des Suchbereichswerts von LDAP auf der globalen Ebene geben Sie Folgendes ein:

```
db2set -gl db2ldap_suchbereich = wert
```

Dabei kann *wert* „local“, „domain“ oder „global“ sein.

Zum Ändern des Standardwerts einer Variablen der Profilregistrierdatenbank für einen bestimmten Knoten eines Exemplars geben Sie Folgendes ein:

```
db2set name_der_variablen_der_profilregistrierdatenbank=neuer_wert  
-i exemplarname knotennummer
```

Zum Zurücksetzen einer Variablen der Profilregistrierdatenbank für ein Exemplar auf den in der globalen Profilregistrierdatenbank vorhandenen Standardwert geben Sie folgenden Befehl ein:

```
db2set -r name_der_variablen_der_profilregistrierdatenbank
```

Zum Zurücksetzen einer Variablen der Profilregistrierdatenbank für einen Knoten in einem Exemplar auf die in der globalen Profilregistrierdatenbank vorhandenen Standardwerte geben Sie folgenden Befehl ein:

```
db2set -r name_der_variablen_der_profilregistrierdatenbank knotennummer
```

Definieren von Umgebungsvariablen unter OS/2

Es sollte unbedingt darauf geachtet werden, alle DB2-spezifischen Variablen der Profilregistrierdatenbank in der DB2-Profilregistrierdatenbank zu definieren. Wenn die DB2-Variablen außerhalb der Registrierdatenbank festgelegt werden, ist keine Fernverwaltung dieser Variablen möglich und die Workstation muss erneut gestartet werden, damit die Variablenwerte wirksam werden.

Unter OS/2 sollten außer den Umgebungsvariablen DB2PATH und DB2INSTPROF keine DB2-Umgebungsvariablen in der Datei `config.sys` definiert sein. Alle Variablen sollten in den Profilregistrierdatenbanken mit Hilfe des Befehls **db2set** definiert werden, außer denen, die echte Umgebungsvariablen bleiben.

Die Umgebungsvariable DB2INSTANCE bleibt ebenfalls eine echte Umgebungsvariable, jedoch ist sie nicht erforderlich, wenn Sie die Variable DB2INSTDEF der Profilregistrierdatenbank verwenden. Diese Variable der Profilregistrierdatenbank definiert den Standardexemplarnamen, der verwendet wird, wenn DB2INSTANCE nicht festgelegt ist.

DB2INSTANCE und DB2PATH werden beim Installieren von DB2 festgelegt. DB2INSTPROF kann nach der Installation festgelegt werden. Die Umgebungsvariable DB2PATH muss festgelegt sein. Diese Umgebungsvariable wird bei der Installation festgelegt und sollte von Ihnen nicht geändert werden. Das Festlegen der Umgebungsvariablen DB2INSTANCE und DB2INSTPROF ist wahlfrei.

Die Einstellung einer Umgebungsvariablen lässt sich mit dem folgenden Befehl feststellen:

```
set variable
```

Zum Ändern der Einstellung einer Umgebungsvariablen geben Sie den folgenden Befehl ein:

```
set variable=wert
```

Zum Definieren der Systemumgebungsvariablen gehen Sie folgendermaßen vor: Editieren Sie die Datei `config.sys`, und starten Sie das System erneut, damit die Änderungen wirksam werden.

Die verschiedenen Profilregistrierdatenbanken befinden sich an folgenden Speicherpositionen:

- Die DB2-Profilregistrierdatenbank auf Exemplarebene befindet sich unter:
`%DB2INSTPROF%\exemplarname\PROFILE.ENV`
- Die DB2-Profilregistrierdatenbank auf globaler Ebene befindet sich unter:
`%DB2INSTPROF%\DEFAULT.ENV`
- Die DB2-Exemplarregistrierdatenbank befindet sich unter:
`%DB2INSTPROF%\PROFILES.REG`

Definieren von Umgebungsvariablen unter Windows NT und Windows 95

Es sollte unbedingt darauf geachtet werden, alle DB2-spezifischen Variablen der Profilregistrierdatenbank in der DB2-Profilregistrierdatenbank zu definieren. Wenn die DB2-Variablen außerhalb der Registrierdatenbank festgelegt werden, ist keine Fernverwaltung dieser Variablen möglich und die Workstation muss erneut gestartet werden, damit die Variablenwerte wirksam werden.

32-Bit-Windows-Betriebssysteme verfügen über eine Systemumgebungsvariable (DB2INSTANCE), die nur außerhalb der Profilregistrierdatenbank festgelegt werden kann. DB2INSTANCE muss jedoch nicht unbedingt festgelegt werden. Die DB2-Variable DB2INSTDEF der Profilregistrierdatenbank kann auf der globalen Profilebene festgelegt werden, um den Exemplarnamen anzugeben, der verwendet wird, wenn DB2INSTANCE nicht definiert ist.

Server mit DB2 Enterprise - Extended Edition unter Windows NT verfügen über die beiden Systemumgebungsvariablen DB2INSTANCE und DB2NODE, die nur außerhalb der Profilregistrierdatenbank festgelegt werden können. Sie müssen DB2INSTANCE nicht unbedingt festlegen. Die DB2-Variable DB2INSTDEF der Profilregistrierdatenbank kann auf der globalen Profilebene festgelegt werden, um den Exemplarnamen anzugeben, der verwendet wird, wenn DB2INSTANCE nicht definiert ist.

Die Umgebungsvariable DB2NODE wird zum Weiterleiten von Anforderungen an einen logischen Zielknoten innerhalb einer Maschine verwendet. Diese Umgebungsvariable muss in der Sitzung festgelegt werden, in der die Anwendung bzw. der Befehl ausgeführt wird, und nicht in der DB2-Profilregistrierdatenbank. Wenn diese Variable nicht definiert wird, wird als logischer Zielknoten standardmäßig der logische Knoten verwendet, der auf der Maschine mit Anschluss Null (0) definiert ist.

Die Einstellung einer Umgebungsvariable lässt sich mit dem Befehl **echo** feststellen. Wenn Sie beispielsweise den Wert der Umgebungsvariablen DB2PATH prüfen wollen, geben Sie Folgendes ein:

```
echo %db2path%
```

Die Systemumgebungsvariablen werden wie folgt definiert:

Unter Windows 95 und Windows 98: Editieren Sie die Datei *autoexec.bat*, und starten Sie das System erneut, damit die Änderung wirksam wird.

Unter Windows NT 4.x: Sie können die DB2-Umgebungsvariablen DB2INSTANCE, DB2PATH und DB2INSTPROF wie folgt definieren:

- Wählen Sie **Start, Einstellungen** und **Systemsteuerung** aus.
- Klicken Sie das Symbol **System** doppelt an.
- Führen Sie folgende Schritte in der Anzeige der Systemsteuerung im Abschnitt für die Umgebungsvariablen des Systems aus:
 1. Wenn die Variable DB2INSTANCE nicht vorhanden ist:
 - a. Wählen Sie eine beliebige Systemumgebungsvariable aus.
 - b. Ändern Sie den Namen im Feld *Variable* in DB2INSTANCE.
 - c. Ändern Sie das Feld *Wert* in den Exemplarnamen. Zum Beispiel db2inst.
 2. Wenn die Variable DB2INSTANCE bereits vorhanden ist, hängen Sie wie folgt einen neuen Wert an:
 - a. Wählen Sie die Umgebungsvariable DB2INSTANCE aus.
 - b. Ändern Sie das Feld *Wert* für den Exemplarnamen. Zum Beispiel db2inst.
 3. Wählen Sie **Setzen** aus.
 4. Wählen Sie **OK** aus.
 5. Starten Sie Ihr System neu, damit diese Änderungen wirksam werden.

Anmerkung: Die Umgebungsvariable DB2INSTANCE kann auch auf Sitzungsebene (Prozessebene) festgelegt werden. Wenn Sie beispielsweise ein zweites DB2-Exemplar mit dem Namen TEST starten wollen, geben Sie die folgenden Befehle in einem Befehlsfenster ein:

```
set db2instance=TEST
db2start
```

Die Profilregistrierdatenbanken befinden sich an folgenden Speicherpositionen:

- Die DB2-Profilregistrierdatenbank auf Exemplarebene in der Registrierdatenbank des Betriebssystems Windows NT mit folgendem Pfad:
`\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2\PROFILES\exemplarname`

Anmerkung: Der *exemplarname* ist für die Datenbankpartition spezifisch, in der Sie arbeiten.

- Die DB2-Profilregistrierdatenbank auf globaler Ebene in der Registrierdatenbank von Windows NT mit folgendem Pfad:

```
\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2\GLOBAL_PROFILE
```

- Die DB2-Profilregistrierdatenbank auf Exemplarknotenebene in der Registrierdatenbank von Windows NT mit folgendem Pfad:

```
...\SOFTWARE\IBM\DB2\PROFILES\exemplarname\NODES\knotennummer
```

Anmerkung: Der *exemplarname* und die *knotennummer* sind für die Datenbankpartition spezifisch, in der Sie arbeiten.

DB2 UDB ermöglicht Ihnen den Zugriff auf DB2 UDB-Variablen der Profilregistrierdatenbank auf Exemplarebene auf einer fernen Maschine. Momentan werden DB2 UDB-Variablen der Profilregistrierdatenbank auf drei verschiedenen Ebenen gespeichert: Maschinenebene (globale Ebene), Exemplarebene und Knotenebene. Die auf Exemplarebene (einschließlich der auf Knotenebene) gespeicherten Variablen der Profilregistrierdatenbank können mit DB2REMOTEPREG umgeleitet werden. Wenn DB2REMOTEPREG eingestellt ist, greift DB2 UDB von der Maschine, auf die DB2REMOTEPREG zeigt, auf die DB2 UDB-Variablen der Profilregistrierdatenbank zu. Beispiel:

```
db2set DB2REMOTEPREG=rmtwkstn
```

Dabei ist *ferne_workstation* der Name der fernen Workstation.

Anmerkung: Bei der Definition dieser Option ist Vorsicht geboten, da alle DB2-Exemplarprofile und Exemplarlisten unter dem angegebenen fernen Maschinennamen gespeichert werden.

Diese Einrichtung kann in Kombination mit dem Wert für DBINSTPROF verwendet werden, um auf ein fernes LAN-Laufwerk auf derselben Maschine zu verweisen, die die Registrierdatenbank enthält.

Definieren von Umgebungsvariablen auf UNIX-Systemen

Es sollte unbedingt darauf geachtet werden, alle DB2-spezifischen Variablen der Profilregistrierdatenbank in der DB2-Profilregistrierdatenbank zu definieren. Wenn die DB2-Variablen außerhalb der Registrierdatenbank festgelegt werden, ist keine Fernverwaltung dieser Variablen möglich. Unter UNIX-Betriebssystemen müssen Sie die Umgebungsvariable DB2INSTANCE festlegen.

Die Prozeduren **db2profile** (für Bourne- und Korn-Shell) und **db2cshrc** (für C-Shell) werden als Beispiele bereitgestellt, um Ihnen Hilfestellung beim Einrichten der Datenbankumgebung zu geben. Diese Dateien finden Sie im Verzeichnis *insthome/sqllib*, wobei *insthome* das Benutzerverzeichnis (HOME) des Exemplareigners ist.

Diese Prozeduren enthalten Anweisungen zu folgenden Operationen:

- Aktualisieren des Pfads eines Benutzers mit folgenden Verzeichnissen:
 - insthome/sqllib/bin
 - insthome/sqllib/adm
 - insthome/sqllib/misc
- Definieren der Variablen DB2INSTANCE mit dem lokalen Standardwert für den exemplarnamen für die Ausführung

Anmerkung: Alle von DB2 unterstützten Variablen außer PATH und DB2INSTANCE müssen in der DB2-Profilregistrierdatenbank festgelegt werden. Definieren Sie Variablen, die von DB2 nicht unterstützt werden, in Ihren Prozedurdateien **db2profile** und **db2cshrc**, um sie festzulegen.

Ein Exemplareigner oder Benutzer mit der Berechtigung SYSADM kann diese Prozeduren für alle Benutzer eines Exemplars anpassen. Alternativ können Benutzer eine Prozedur kopieren und anpassen und anschließend die Prozedur direkt aufrufen oder ihrer Datei .profile oder .login hinzufügen.

Zum Ändern der Umgebungsvariablen für die aktuelle Sitzung setzen Sie Befehle wie die folgenden ab:

- Für Korn-Shell:

```
db2instance=inst1
export db2instance
```
- Für Bourne-Shell oder C-Shell:

```
set db2instance inst1
```

Zur ordnungsgemäßen Verwaltung der DB2-Profilregistrierdatenbank müssen die folgenden Regeln im Hinblick auf die Zugriffsberechtigungen und Eigentumsrechte auf UNIX-Betriebssystemen eingehalten werden. (Informationen zum DB2-Verwaltungsserver (DAS) finden Sie in „Erstellen eines DB2-Verwaltungsservers (DAS)“ auf Seite 89.)

- Die DB2-Profilregistrierdatenbank auf Exemplarebene befindet sich unter:

```
INSTHOME/sqllib/profile.env
```

Die Zugriffsberechtigung und Eigentumsrechte für diese Datei sollten wie folgt definiert sein:

```
-rw-r--r-- Exemplareigner DAS_Exemplargruppe profile.env
```

INSTHOME ist der Benutzerpfad (HOME) des Exemplareigners.

- Die DB2-Profilregistrierdatenbank auf globaler Ebene befindet sich unter:

- `/var/db2/<versions_id>/default.env` für die Betriebssysteme AIX, Solaris, SINIX und NUMA-Q(Sequent) (wobei `<versions_id>` für die aktuelle Version steht).
- `/var/opt/db2/<versions_id>/default.env` für das Betriebssystem HP-UX (wobei `<versions_id>` für die aktuelle Version steht).

Die Zugriffsberechtigung und Eigentumsrechte für diese Datei sollten wie folgt definiert sein:

```
-rw-r--r-- DAS_Exemplareigner DAS_Exemplargruppe default.env
```

Zum Ändern einer globalen Variablen der Profilregistrierdatenbank muss ein Benutzer entweder als Root oder als DAS-Exemplareigner angemeldet sein. Weitere Informationen zum DB2-Verwaltungsserver finden Sie in „Erstellen eines DB2-Verwaltungsservers (DAS)“.

- Die DB2-Profilregistrierdatenbank auf Exemplarknotenebene befindet sich unter:

```
INSTHOME/sql1lib/nodes/knotennummer.env
```

Die Zugriffsberechtigung und Eigentumsrechte für dieses Verzeichnis und diese Datei sollten wie folgt definiert sein:

```
drwxrwxr-x Exemplareigner DAS_Exemplargruppe Knoten
```

```
-rw-r--r-- Exemplareigner DAS_Exemplargruppe Knotennummer.env
```

Anmerkung: Der *Exemplareigner* und der *DAS_Exemplareigner* sollten beide Mitglieder der *DAS_Exemplargruppe* sein.

INSTHOME ist der Benutzerpfad (HOME) des Exemplareigners.

- Die DB2-Exemplarregistrierdatenbank befindet sich unter:
 - `/var/db2/<versions_id>/profiles.reg` für die Betriebssysteme AIX, Solaris, SINIX und NUMA-Q(Sequent) (wobei `<versions_id>` für die aktuelle Version steht).
 - `/var/opt/db2/<versions_id>/profiles.reg` für das Betriebssystem HP-UX (wobei `<versions_id>` für die aktuelle Version steht).

Die Zugriffsberechtigung und Eigentumsrechte für diese Datei sollten wie folgt definiert sein:

```
-rw-r--r-- root system profiles.reg
```

Erstellen eines DB2-Verwaltungsservers (DAS)

Der DB2-Verwaltungsserver (DAS - DB2 Administration Server) ist ein spezieller DB2-Verwaltungssteuerelement, der nur zur Unterstützung von Verwaltungsaufgaben auf anderen DB2-Servern verwendet wird. Sie müssen über einen aktiven DAS verfügen, um die Funktion Client-Konfiguration -

Unterstützung oder die Steuerzentrale zu verwenden. Der DAS unterstützt die Steuerzentrale und die Funktion Client-Konfiguration - Unterstützung bei den folgenden Verwaltungsaufgaben:

- Aktivieren der Fernverwaltung von DB2-Servern.
- Bereitstellen der Einrichtung für Jobverwaltung, einschließlich der Fähigkeit, die Ausführung von Befehlsprozeduren für sowohl DB2 als auch das Betriebssystem zu terminieren. Diese Befehlsprozeduren sind benutzerdefiniert. Mit der Steuerzentrale können Sie den Zeitplan von Jobs definieren, die Ergebnisse beendeter Jobs anzeigen und andere Verwaltungsaufgaben für Jobs ausführen, die fern oder lokal auf dem DAS vorhanden sind.
- Bereitstellen eines Mittels zum Aufspüren von Informationen zur Konfiguration von DB2-Exemplaren, DB2-Datenbanken und anderen DB2-Verwaltungsservern in Zusammenarbeit mit dem DB2-Dienstprogramm Discovery. Mit Hilfe dieser Informationen vereinfachen und automatisieren die Funktion Client-Konfiguration - Unterstützung und die Steuerzentrale die Konfiguration der Client-Verbindungen zu DB2-Datenbanken.

Sie können nur einen Verwaltungsserver auf einer Maschine haben. Der DAS wird während der Installation so konfiguriert, dass er startet, wenn das Betriebssystem gebootet wird.

Der DAS wird verwendet, um ferne Tasks auf dem Host-System für eine Client-Anforderung von der Steuerzentrale oder von „Client-Konfiguration - Unterstützung“ durchzuführen. Der berechtigte Zugriff auf den DAS erfordert Clients mit der Berechtigung SYSADM. Alle der Clients können Teil des Konfigurationsparameters SYSADM_GROUP sein.

Für die Ausführung einiger dieser angeforderten Tasks kann eine spezielle Berechtigung erforderlich sein. Der DAS läuft unter der Kennung eines bestimmten Benutzers. Die diesem Benutzer erteilten Zugriffsrechte müssen auf die für die Tasks erforderlichen Befehle oder Operationen eingeschränkt werden, die vom Administrator ausgeführt werden sollen. Im allgemeinen sind folgende Tasks bzw. Operationen erforderlich:

- Abfragen der Konfigurationsdaten des Betriebssystems (OS)
- Abfragen des Betriebssystems nach Benutzer- und Gruppeninformationen
- Ausführen von Aktionen für andere DB2-Exemplare, um sie zu starten oder stoppen
- Ausführen terminierter Jobs
- Sammeln von Informationen für Konnektivitäts- und Protokollkonfiguration

Weitere Informationen zur Einrichtung der DAS-Kommunikation finden Sie im Handbuch *Einstieg* für Ihre Plattform.

Erstellen des DB2-Verwaltungsservers

Normalerweise erstellt das Setup-Programm während der DB2-Installation einen DAS auf der Exempleignermaschine. Wenn dies jedoch vom Setup-Programm versäumt wurde, können Sie manuell einen DAS erstellen.

Folgende Hinweise sollen Ihnen eine Übersicht der Vorgänge während des Installationsprozesses in Bezug auf DAS geben:

- Auf Plattformen unter OS/2 oder Windows NT:

Melden Sie sich an der Maschine, auf der Sie den DAS erstellen wollen, mit einem Benutzereintrag an, der über die lokale Administratorberechtigung verfügt. Wenn ein bestimmter Benutzer angegeben werden muss, erstellen Sie einen Benutzer mit lokaler Administratorberechtigung. Geben Sie den Befehl `db2admin create` ein. Wenn ein bestimmter Benutzereintrag gewünscht wird, müssen Sie „/USER:“ und „/PASSWORD:“ beim Absetzen des Befehls `db2admin create` verwenden.

Bei der Erstellung des DB2-Verwaltungsservers können Sie wahlfrei einen Benutzereintragsnamen und ein Benutzerkennwort angeben. Wenn die Angaben gültig sind, wird mit dem Benutzereintragsnamen und dem Kennwort der Eigner des DAS identifiziert. Verwenden Sie die für den DAS erstellte Benutzer-ID bzw. den dafür erstellten Benutzereintragsnamen nicht als Benutzereintrag. Setzen Sie das Kennwort für den Benutzereintragsnamen auf „Kennwort läuft nie ab“. Nach der Erstellung des DAS können Sie mit dem Befehl `db2admin setid` den Benutzereintragsnamen für den Eigner und das Benutzerkennwort festlegen bzw. ändern. Weitere Informationen zu diesem Befehl finden Sie im Handbuch *Command Reference*.

Unter DB2 UDB für Windows NT Enterprise - Extended Edition wird derjenige Datenbankpartitionsserver als Koordinatorknoten verwendet, der sich auf derselben Maschine wie der DAS befindet, wenn Sie die Konfiguration der Verbindung zu einem DB2-Server mit Hilfe der Steuerzentrale oder der Funktion Client-Konfiguration - Unterstützung automatisieren. Dies bedeutet, dass alle physischen Verbindungen vom Client zur Datenbank über den Datenbankpartitionsserver auf der Exempleignermaschine laufen, bevor sie an andere Datenbankpartitionsserver weitergeleitet werden.

Unter DB2 UDB für Windows NT Enterprise - Extended Edition ermöglicht das Erstellen zusätzlicher Verwaltungsserver auf anderen Maschinen der Steuerzentrale bzw. der Client-Konfiguration - Unterstützung das Konfigurieren anderer Systeme als Koordinatorknoten unter Verwendung von DB2 Discovery. Führen Sie dazu folgende Maßnahmen durch:

1. Melden Sie sich an der Maschine mit einem Konto an, das über die lokale Administratorberechtigung verfügt.
2. Erstellen Sie ein Windows NT-Konto mit lokaler Administratorberechtigung, der vom DAS verwendet wird. Stellen Sie sicher, dass der

Benutzername des Kontos mit den DB2-Namenskonventionen übereinstimmt. Beim Erstellen des Kontos für den DAS ist Folgendes zu beachten:

- Verwenden Sie das Konto für den DAS nicht als Benutzerkonto.
- Setzen Sie das Kennwort für den Benutzereintrag auf Kennwort läuft nie ab.

3. Führen Sie den folgenden Befehl aus:

```
db2admin create /user:benutzername  
/password:kennwort
```

Dabei sind *benutzername* und *kennwort* der Benutzername und das Kennwort für den DAS.

• Auf Plattformen unter UNIX:

1. Stellen Sie sicher, dass Sie über die Root-Berechtigung verfügen.
2. Geben Sie über eine Befehlseingabeaufforderung den folgenden Befehl vom Unterverzeichnis *exemplar* im Pfad des Exemplars von DB2 Universal Database ein:

```
dasict ASName
```

– Unter AIX:

```
/usr/lpp/db2_nn_00&/exemplar/  
dasict ASname
```

– Unter HP-UX, NUMA-Q(Sequent) oder Solaris:

```
/opt/IBMDB2/<versions_id>/exemplar/  
dasict ASname
```

– Unter Linux:

```
/usr/IBMDB2/<versions_id>/exemplar/  
dasict ASname
```

Dabei ist *ASName* der Exemplarname des Verwaltungsservers und *db2_nn_00&* oder *<versions_id>* die Kennung der aktuellen Version.

Anmerkung: Wenn Sie NIS und NIS+ verwenden, müssen Sie die Benutzer- und Gruppennamen wie folgt einrichten:

- Die Primärgruppe des DAS muss sich in der Sekundärgruppe aller Exemplare befinden.
- Die Sekundärgruppe des DAS muss die Primärgruppe aller Exemplare enthalten.

Die Listen der Sekundärgruppen werden nur automatisch geändert, wenn weder NIS noch NIS+ auf dem System aktiv ist.

Da eine Benutzer-ID nur Eigner eines einzigen Exemplars sein kann, benötigen Sie eine separate Benutzer-ID als Eigner für jeden DB2 Verwaltungs-server (DAS), den Sie erstellen.

Wenn Sie einen Verwaltungsserver erstellt haben, sollten Sie ihn zur Einrichtung von Verzeichnisstrukturen und Zugriffsberechtigungen verwenden.

Starten und Stoppen des Verwaltungsservers

Zum manuellen Starten oder Stoppen des DAS müssen Sie sich zunächst mit einem Konto oder einer Benutzer-ID mit lokaler Administratorberechtigung an der Maschine anmelden.

Unter DB2 für OS/2 oder DB2 für Windows NT müssen Sie Folgendes ausführen:

- Zum Starten des DAS geben Sie `db2admin start` ein.
- Zum Stoppen des DAS geben Sie `db2admin stop` ein.

Anmerkung: Für beide Befehle gilt unter Windows NT, dass die Person, die diese Befehle verwendet, über die Berechtigung SYSADM, SYSCTRL oder SYSMAINT verfügen muss.

Beim Arbeiten mit DB2 unter UNIX-Betriebssystemen müssen Sie Folgendes ausführen:

- Zum Starten des DAS:
 1. Melden Sie sich als DAS-Eigner an.
 2. Führen Sie die Startprozedur mit einem der folgenden Befehle aus:

```
. INSTHOME/sql1lib/db2profile    (für Bourne- oder Korn-Shell)
source INSTHOME/sql1lib/db2cshrc (für C-Shell)
```

Dabei ist INSTHOME das Benutzerverzeichnis des Exemplars.

3. Verwenden Sie zum Starten des DAS den Befehl **db2admin**:

```
db2admin start
```

Anmerkung: Der DAS wird nach jedem Warmstart des Systems automatisch gestartet.

- Zum Stoppen des DAS:
 1. Melden Sie sich als DAS-Eigner an.
 2. Führen Sie die Startprozedur mit einem der folgenden Befehle aus:

```
. INSTHOME/sql1lib/db2profile    (für Bourne- oder Korn-Shell)
source INSTHOME/sql1lib/db2cshrc (für C-Shell)
```

Dabei ist INSTHOME das Benutzerverzeichnis des Exemplars.

3. Stoppen Sie den DAS mit dem folgenden Befehl **db2admin**:

```
db2admin stop
```

Anmerkung: Für beide Befehle gilt unter UNIX, dass sich die Person, die diese Befehle ausführt, mit der Berechtigungs-ID des DAS-Eigners angemeldet haben muss.

Auflisten des Verwaltungsservers

Mit dem folgenden Befehl können Sie den Namen des DAS-Exemplars auf Ihrer Maschine feststellen:

```
db2admin
```

Dieser Befehl wird auch zum Starten und Stoppen des DAS, Erstellen eines neuen Benutzers und Kennworts, Löschen eines DAS-Exemplars und Einrichten und Ändern eines Benutzereintrags, der dem DAS-Exemplar zugeordnet ist, verwendet.

Konfigurieren des Verwaltungsservers

Die aktuellen Werte der Konfigurationsparameter für die Verwaltung, die für den DAS relevant sind, werden mit folgendem Befehl angezeigt:

```
db2 get admin cfg
```

Dieser Befehl zeigt die aktuellen Werte an, die beim Installieren des Produkts als Standardwerte angegeben wurden, oder diejenigen Werte, die bei früheren Aktualisierungen der Konfigurationsparameter angegeben wurden.

Zur Aktualisierung einzelner Einträge in der Konfigurationsdatei des Datenbankmanagers, die für den DAS relevant sind, geben Sie Folgendes ein:

```
db2 update admin cfg using ...
```

Im Handbuch *Command Reference* finden Sie weitere Informationen dazu, welche Konfigurationsparameter des Datenbankmanagers geändert werden können.

Wenn die Konfigurationsparameter auf die empfohlenen Standardwerte des Datenbankmanagers zurückgesetzt werden sollen, geben Sie Folgendes ein:

```
db2reset admin cfg
```

Änderungen an der Konfigurationsdatei des Datenbankmanagers werden erst wirksam, wenn sie in den Hauptspeicher geladen werden (d. h., wenn der Befehl `db2admin stop` gefolgt vom Befehl `db2admin start` ausgeführt wird oder wenn bei einer Windows NT-Plattform der Dienst gestoppt und wieder gestartet wird). Informationen zum Einrichten von Übertragungsprotokollen für den DAS finden Sie im Handbuch *Einstieg* für Ihre Plattform.

Sicherheitsaspekte des DB2-Verwaltungsservers

Sie müssen sich zunächst mit einem Konto oder einer Benutzer-ID, die über die lokale Administratorberechtigung verfügt, an die Maschine anmelden.

Anmerkung: Unter Windows NT sollten Sie nicht das Dienstprogramm **Dienste der Systemsteuerung** verwenden, um das Anmeldekonto für den DAS zu ändern, da dem Anmeldekonto nicht alle erforderlichen Zugriffsrechte erteilt werden. Verwenden Sie zum Festlegen oder Ändern des Anmeldekontos für den DB2-Verwaltungsserver (DAS) immer den Befehl **db2admin**.

Nach dem Erstellen des DAS können Sie das Anmeldekonto mit dem Befehl **db2admin** wie folgt festlegen oder ändern:

```
db2admin setid benutzername kennwort
```

Dabei sind *benutzername* und *kennwort* der Benutzername und das Kennwort eines Kontos, das über die lokale Administratorberechtigung verfügt.

Es wird empfohlen, der Benutzer-ID bzw. dem Benutzernamen auf jedem Server innerhalb der Umgebung die Berechtigung SYSADM zuzuordnen, so dass über diese(n) bei Bedarf andere Exemplare gestartet oder gestoppt werden können.

Aktualisieren des DB2-Verwaltungsservers

Wenn DB2 unter UNIX-Betriebssystemen durch Installieren einer vorläufigen Programmkorrektur (PTF) oder einer Codekorrektur (Patch) aktualisiert wird, sollten auch alle DB2-Verwaltungsserver (DAS) und alle vorhandenen Exemplare aktualisiert werden. Verwenden Sie zum Aktualisieren eines DAS den Befehl **dasiupdt**, der sich im Unterverzeichnis *instance* des Unterverzeichnisses für das spezifische Release der installierten DB2-Version befindet.

Sie müssen sich (unter UNIX) zunächst über ein Konto als „Root“ oder über eine Benutzer-ID mit lokaler Administratorberechtigung an der Maschine anmelden.

Der Befehl ist wie folgt anzugeben:

```
dasiupdt ExempName
```

Dabei ist *ExempName* der Anmeldenname des Exemplareigners. Für diesen Befehl gibt es außerdem wahlfreie Parameter, die (durch Leerzeichen getrennt) vor dem Parameter *ExempName* wie folgt angegeben werden können:

- **-h** oder **-?**
Zeigt ein Hilfemenü für diesen Befehl an.
- **-d**
Aktiviert den Debug-Modus, der für die Problemanalyse verwendet wird.

Entfernen des DB2-Verwaltungsservers

Sie müssen sich (unter UNIX) zunächst über ein Konto als „Root“ oder über eine Benutzer-ID mit lokaler Administratorberechtigung an der Maschine anmelden.

Der DAS wird folgendermaßen entfernt:

- Auf Plattformen unter OS/2 oder Windows NT:
 1. Stoppen Sie den DAS mit dem Befehl `db2admin stop`.
 2. Sichern Sie (falls erforderlich) alle Dateien im Unterverzeichnis `db2das00` des Unterverzeichnisses `sql1lib`. Das Exemplarverzeichnis wird durch die Variable `DB2INSTPROF` der Profilregistrierdatenbank angegeben.
 3. Löschen Sie den DAS mit dem Befehl `db2admin drop`.

Anmerkung: Unter Windows NT muss die Person, die diesen Befehl verwendet, über die Berechtigung `SYSADM`, `SYSCTRL` oder `SYSMAINT` verfügen.

- Unter UNIX-Betriebssystemen:
 1. Melden Sie sich als DAS-Eigner an.
 2. Führen Sie die Startprozedur mit einem der folgenden Befehle aus:

```
. INSTHOME/sql1lib/db2profile    (für Bourne- oder Korn-Shell)
source INSTHOME/sql1lib/db2cshrc (für C-Shell)
```

Dabei ist `INSTHOME` das Benutzerverzeichnis des Exemplars.

3. Stoppen Sie den DAS mit dem folgenden Befehl **db2admin**:

```
db2admin stop
```
4. Sichern Sie (falls erforderlich) alle Dateien im Unterverzeichnis `sql1lib` des DAS-Benutzerzeichnisses. Das Exemplarverzeichnis wird durch die Variable `DB2INSTPROF` der Profilregistrierdatenbank angegeben.
5. Melden Sie sich ab.
6. Melden Sie sich als `root` an, und entfernen Sie den DAS mit dem folgenden Befehl **dasidrop**:

```
dasidrop ASName
```

Dabei ist `ASName` der Exemplarname des Verwaltungsservers. Dieser Befehl befindet sich im Unterverzeichnis `instance` des Unterverzeichnisses für das spezifische Release der installierten DB2-Version.

Anmerkung: Der Befehl **dasidrop** entfernt das Unterverzeichnis `sql1lib` im Ausgangsverzeichnis des DB2-Verwaltungsservers (DAS).

Einrichten eines DAS mit EEE-Systemen

Die folgenden Informationen zeigen die notwendigen Schritte zum Konfigurieren von Servern unter DB2 EEE (Solaris, NT, Sequent, HP-UX und AIX) für die Fernverwaltung unter Verwendung der Steuerzentrale.

Während der Installation erstellt das Konfigurationsprogramm einen einzelnen DAS auf der Exemplareignermaschine. Es kann wünschenswert sein, weitere DAS auf anderen Maschinen zu erstellen, damit die Steuerzentrale oder die Funktion Client-Konfiguration - Unterstützung auf andere Koordinator-knoten zugreifen kann. Der Systemaufwand für die Funktionen des Koordinator-knotens kann dadurch auf mehrere Knoten eines Exemplars verteilt werden.

Gehen Sie wie folgt vor, um die Koordinatorfunktion zu verteilen:

1. Erstellen Sie auf den ausgewählten zusätzlichen Maschinen des partitionierten Datenbanksystems einen neuen DAS.
2. Katalogisieren Sie jeden DAS in der Steuerzentrale oder in der Client-Konfiguration - Unterstützung als eigenständiges System.
3. Katalogisieren Sie in jedem neuen System dasselbe Exemplar, und geben Sie dabei denselben Maschinennamen an, der beim Katalogisieren des DAS verwendet wurde.

Bei der Konfiguration sind zwei Aspekte zu berücksichtigen: die Erfordernisse für den DB2-Verwaltungsserver (DAS) und die Empfehlungen für das verwaltete DB2-Zielexemplar. In den folgenden drei Abschnitten ist den beiden Konfigurationsthemen jeweils ein Abschnitt gewidmet. Der Behandlung dieser Konfigurationsthemen geht ein Abschnitt voran, der die angenommene Umgebung beschreibt.

Beispielumgebung:

produkt/version:

DB2 UDB EEE V7.1

Installationspfad:

installationspfad

TCP-Servicedatei:

tcp_servicedatei

DB2-Exemplar:

Name: db2inst

Eigner-ID:

db2inst

Exemplarpfad:
exemplarpfad

Knoten:
3 Knoten, db2nodes.cfg:
• 0 hostA 0 hostA0switch
• 1 hostA 1 hostA1switch
• 2 hostB 0 hostBswitch

DBname:
db2instDB

DAS:

Name: db2as

Eigner/Benutzer-ID:
db2as

Exemplarpfad:
das_pfad

Installations-/Ausführungs-Host:
hostA

Anschluss für Übertragung zwischen Knoten:
16000 (nicht benutzter Anschluss für HostA und HostB)

Anmerkung: Geben Sie für die obigen Felder die standortspezifischen Werte ein. Zum Beispiel enthält die folgende Tabelle die Beispielpfadnamen für jede unterstützte EEE-Plattform:

Tabelle 1. Beispielpfadnamen für unterstützte EEE-Plattformen

Pfad	DB2 UDB EEE für AIX	DB2 UDB EEE für Solaris	DB2 UDB EEE für Windows NT
<i>installationspfad</i>	/usr/lpp/<v_r_ID>	/opt/IBMdb2/<v_r_ID>	C:\sqllib
<i>exemplarpfad</i>	/home/db2inst/sqllib	/home/db2inst/sqllib	C:\profiles\db2inst
<i>das_pfad</i>	/home/db2as/sqllib	/home/db2as/sqllib	C:\profiles\db2as
<i>TCP-Datei services</i>	/etc/services	/etc/services	C:\winnt\system32 \drivers\etc\services

In der Tabelle steht <v_r_ID> für die plattformsspezifische Version und die Release-Kennung. Bei DB2 UDB EEE für AIX in Version 5.2 gilt für <v_r_ID> der Wert db2_05_00.

Beim Installieren von DB2 UDB EEE erstellt das Konfigurationsprogramm einen DAS auf der Exemplareignermaschine.

Der Datenbankpartitionsserver befindet sich auf der gleichen Maschine wie der DAS und ist der Verbindungspunkt für das Exemplar. Das bedeutet, dieser Datenbankpartitionsserver ist der Koordinatorknoten für Anforderungen der Steuerzentrale oder der Client-Konfiguration - Unterstützung an das Exemplar.

DAS-Konfiguration: Der DAS ist ein administrativer Steuerpunkt, der bestimmte Tasks für die Steuerzentrale ausführt. Es kann nur jeweils einen DAS pro physischer Maschine geben. Im Fall eines EEE-Exemplars, das aus mehreren Maschinen besteht, muss mindestens eine der Maschinen einen DAS ausführen, damit die Steuerzentrale das EEE-Exemplar verwalten kann. Dieser DAS (db2as) „repräsentiert“ das System, das in der Baumstruktur der Steuerzentrale als Elter des DB2-Zielexemplars (db2inst) vorhanden ist.

Zum Beispiel besteht db2inst aus drei Knoten, die auf zwei physische Maschinen oder Hosts verteilt sind. Die Mindestanforderung wird durch das Ausführen von db2das auf HostA **oder** HostB erfüllt.

Anmerkungen:

1. Die Anzahl der auf HostA vorhandenen Partitionen wirkt sich nicht auf die Anzahl von DB2-Verwaltungsservern aus, die auf dem Host ausgeführt werden können. Sie können nur jeweils eine Kopie von db2as auf HostA ausführen, unabhängig von der MLN-Konfiguration (MLN - Multiple Logical Nodes) für diesen Host.
2. Es ist nicht notwendig, die DAS-ID (db2as) auf allen Hosts zu erstellen. Sie braucht nur auf dem Host vorhanden zu sein, auf dem sie aktiv ist. Zudem ist es nicht notwendig, das Benutzerverzeichnis der DAS-ID auf allen Hosts anzuhängen. Vor allem bei diesem Beispiel muss die ID db2as auf HostA vorhanden sein, sie ist nicht auf HostB erforderlich, und das Benutzerverzeichnis von db2as braucht nicht auf HostB angehängt zu werden.

Kommunikation zwischen Steuerzentrale und DAS: Serviceanschlüsse: Die Steuerzentrale überträgt mit Hilfe des TCP-Serviceanschlusses 523 Daten an den DAS. Da dieser Anschluss für exklusive Benutzung durch DB2 UDB reserviert ist, ist es nicht notwendig, neue Einträge in die TCP-Datei *services* einzufügen.

Administrative Kommunikation zwischen Knoten: Serviceanschlüsse: Für einige Verwaltungs-Tasks muss der DAS Kommunikation mit allen Knoten ausführen. Dazu muss in der TCP-Datei *services* für jeden Host, der am Exemplar teilnimmt, ein benannter TCP-Anschluss definiert werden.

Anmerkung: Windows NT EEE versucht, den TCP-Anschlusseintrag der TCP-Datei *service* hinzuzufügen.

Zum Beispiel wird db2inst über die beiden Hosts HostA und HostB definiert. Wie im Abschnitt „Beispielumgebung“ auf Seite 97 erklärt, wird der Anschluss 16000 auf beiden Hosts nicht verwendet. Daher muss für HostA und HostB die folgende Zeile in die TCP-Datei *services* eingefügt werden.

```
db2ccmsrv 16000/tcp
```

Der Anschlussname db2ccmsrv muss vorhanden sein und genau so wie oben angezeigt geschrieben werden, und auf allen Hosts muss die gleiche ausgewählte Anschlussnummer verwendet werden.

Administrative Kommunikation zwischen Knoten: UNIX DB2 EEE Server:

Nach dem Einfügen der Zeile mit dem TCP-Anschluss in die TCP/IP-Datei *services* auf HostA und HostB muss auf allen Hosts, die am Exemplar teilnehmen, ein administrativer Empfangsprozess, d. h. Dämon (db2cclst), gestartet werden. Dies kann manuell von der Befehlszeile oder durch Konfigurieren des Systems zum automatischen Aufrufen von db2cclst bei jedem Systemstart erfolgen:

Manuell:

Rufen Sie von der ID des zu verwaltenden Exemplars (db2inst) den folgenden Befehl von HostA oder HostB auf:

```
rah '<install_path>/bin/db2cclst'
```

Unter AIX zum Beispiel würde dieser Befehlsaufruf wie folgt aussehen:

```
rah '/usr/lpp/<v_r_ID>/bin/db2cclst'
```

Der Befehl rah befindet sich im Unterverzeichnis instance des Unterverzeichnisses für die Versions- und Release-Angaben. Der exakte Name des Unterverzeichnisses für die Versions- und Release-Angaben kann betriebssystemabhängig variieren. Dabei ist instance das Benutzerverzeichnis des Exemplars, das Sie verwenden wollen.

Im vorliegenden Fall steht <v_r_ID> für die plattformspezifische Versions- und Release-Kennung. Bei DB2 UDB EEE für AIX in Version 5.2 gilt für <v_r_ID> der Wert db2_05_00.

Automatisch:

Fügen Sie als Root einen entsprechenden Eintrag in der Datei */etc/inittab*

hinzu. Unter AIX zum Beispiel würde dieser Befehlsaufruf auf HostA und HostB wie folgt ausgeführt werden:

```
mkitab "db2cclst::once:su - db2inst -c  
/usr/lpp/<v_r_ID>/bin/db2cclst"
```


Bei jedem Booten einer der beiden Maschinen wird db2cclst ohne Benutzereingriff aufgerufen.

In der Tabelle steht <v_r_ID> für die plattformspezifische Version und die Release-Kennung. Bei DB2 UDB EEE für AIX in Version 5.2 gilt für <v_r_ID> der Wert db2_05_00.

Sie können durch Aufrufen des folgenden Befehls von der Exemplar-ID db2inst prüfen, ob der Empfangsdämon auf beiden Hosts aktiv ist:

```
rah 'ps -ef | grep db2cclst'
```

Wenn der Prozess db2cclst nicht auf jedem Host aktiv ist, können Sie zusätzliche Diagnoseinformationen durch Hinzufügen der folgenden Zeile zu /etc/syslog.conf auf jedem Host abrufen:

```
*.info /tmp/db2/user.info
```

Dabei kann die Datei /tmp/db2/user.info durch eine geeignetere Datei ersetzt werden.

Anmerkung: Die Datei muss vorhanden sein, und der SYSLOG-Dämon muss angewiesen werden, seine Konfigurationsdatei nach der Durchführung der Änderungen erneut zu lesen:

```
kill -1 <syslogd-PID>
```

Dabei kann syslogd-PID durch Ausführung des folgenden Befehls abgerufen werden:

```
ps -ef | grep syslogd
```

Anschließend können Sie nach dem manuellen Aufrufen des Empfangsprogramms wie oben beschrieben die Systemprotokolldatei /tmp/db2/user.info auf dem ausgefallenen Host anzeigen, um durch db2cclst generierte Fehlernachrichten einzusehen.

Administrative Kommunikation zwischen Knoten: Windows NT DB2 EEE Server: Der DB2 Remote Command Service (db2rcmd.exe) führt die administrative Kommunikation zwischen Knoten automatisch aus. Im Fall eines Fehlers werden in die Windows NT-Registrierdatenbank Diagnoseinformationen aufgenommen.

Sicherheit: Der DAS kann nur dann einige Verwaltungs-Tasks für ein Exemplar ausführen, wenn er über die entsprechende Berechtigung verfügt. Der DAS muss im Besonderen ein Systemadministrator (SYSADM) für das verwaltete Zielexemplar sein.

Dem DAS muss diese Berechtigung für alle verwalteten DB2-Exemplare erteilt werden. Kandidatenexemplare sind die auf der gleichen Maschine wie der DAS installierten Exemplare. Für ein Exemplar von DB2 EEE muss mindestens ein Datenbankpartitionsserver auf der gleichen Maschine wie der DAS vorhanden sein, um wie oben beschrieben in Frage zu kommen.

Unter UNIX zum Beispiel kann db2as die erforderliche Berechtigung zum Verwalten von db2inst zum einen erteilt werden, indem sichergestellt wird, dass die Primärgruppen von db2inst und db2as identisch sind. Es reicht zum anderen aus, die Primärgruppe von db2inst zu einer Sekundärgruppe von db2as und die Primärgruppe von db2as zu einer Sekundärgruppe von db2inst zum machen. Eine weitere Option wäre das Einstellen des Konfigurationsparameters SYSADM_GROUP für die Datenbankverwaltung für db2inst auf die Primärgruppe von db2as.

Unter Windows NT muss db2as zur Gruppe der lokalen Administratoren auf HostA und HostB gehören. Neben der Option, die db2as-ID zu erstellen und der Gruppe für lokale Administratoren auf beiden Hosts hinzuzufügen, könnten Sie eine Domänen-ID für db2as erstellen und diese Domänen-ID der Gruppe für lokale Administratoren auf jedem Host hinzufügen.

Umgebung: Die Installation für den DAS sollte bestimmte Variablen konfigurieren, die für einen ordnungsgemäßen Betrieb notwendig sind. Sie können die aktuellen Werte für diese Variablen durch Ausführen des folgenden Befehls von der DB2-Exemplar-ID db2inst oder von der DAS-ID db2das prüfen:

```
db2set -g
```

Wenigstens die folgenden Parameter müssen mit den folgenden Werten definiert werden:

```
DB2SYSTEM=hostA  
DB2ADMINSERVER=db2as
```

Stellen Sie zudem für die Kommunikation zwischen dem DAS und der Steuerzentrale sicher, dass die Variable DB2COMM der Profilregistrierdatenbank auf „TCPIP“ eingestellt ist. Sie können diese Einstellung durch Ausführen des folgenden Befehls von der DAS-ID db2as sowie auf globaler Ebene (-g) und Exemplarebene (-i) (nur eine muss eingestellt sein) prüfen:

```
db2set -a11
```

Prüfen Sie gleichermaßen, ob der Parameter DB2COMM für das DB2-Exemplar auf „TCPIP“ eingestellt ist, um die Kommunikation zwischen der Steuerzentrale und db2inst durch Absetzen des folgenden Befehls von der db2inst-ID zu aktivieren:

```
db2set -a11
```

Wenn Sie diesen Parameter für den DAS ändern, müssen Sie den DAS erneut starten, damit die Änderungen wirksam werden. Der Neustart des DB2-Exemplars ist auch erforderlich, wenn dieser Parameter für das DB2-Exemplar geändert wurde. Für `db2inst` würden Sie den Befehl `db2stop` gefolgt vom Befehl `db2start` absetzen, während für den DAS die Befehle `db2admin stop` und `db2admin start` abgesetzt würden.

Erkennen (Discovery) von Verwaltungsservern, Exemplaren und Datenbanken: Der Discovery-Modus `KNOWN` ermöglicht das Aufspüren von Exemplaren und Datenbanken auf Systemen, die Ihrem Client bekannt sind, und das Hinzufügen neuer Systeme, damit deren Exemplare und Datenbanken aufgespürt werden können. Der Discovery-Modus `SEARCH` bietet die gleichen Funktionen wie der Modus `KNOWN` sowie zusätzlich die Möglichkeit, Ihr lokales Netzwerk nach anderen DB2-Servern zu durchsuchen.

Damit ein Server den Discovery-Modus `KNOWN` unterstützt, setzen Sie den Parameter `discover` in der DAS-Konfigurationsdatei auf `KNOWN`. Damit der Server den Discovery-Modus `SEARCH` unterstützt, setzen Sie diesen Parameter auf `SEARCH`. Wenn Sie verhindern wollen, dass ein Server und alle dazugehörigen Exemplare und Datenbanken aufgespürt werden können, setzen Sie diesen Parameter auf `DISABLE`.

Anmerkung: Der Discovery-Modus `SEARCH` gibt denselben TCP/IP-Host-Namen an einen Client zurück wie Ihr DB2-Server-System, wenn Sie den Befehl `hostname` eingeben. Die diesem Host-Namen auf dem Client zugeordnete IP-Adresse wird entweder durch den auf Ihrer Client-Maschine konfigurierten TCP/IP-Domänennamensserver (DNS) oder (wenn kein DNS konfiguriert ist) durch einen Zuordnungseintrag in der Datei `hosts` des Clients festgelegt. Wenn auf Ihrem DB2-Serversystem mehrere Adapterkarten konfiguriert sind, müssen Sie sicherstellen, dass TCP/IP auf dem Server so konfiguriert ist, dass der richtige Host-Name zurückgegeben wird, und dass die Datei `hosts` des Domänennamensservers oder des lokalen Clients den Host-Namen der gewünschten IP-Adresse zuordnet.

Auf dem Client wird das Aufspüren (Discovery) ebenfalls durch den Parameter `discover` aktiviert. Allerdings wird der Parameter `discover` in diesem Fall im Client-Exemplar (oder in dem als Client eingesetzten Server) wie folgt festgelegt:

- **KNOWN**

Ermöglicht der Client-Konfiguration - Unterstützung das Aktualisieren der Liste der bekannten Systeme und das Hinzufügen neuer Systeme zu dieser Liste mit Hilfe des Druckknopfs **System hinzufügen**. Wenn der Parameter `discover` auf `KNOWN` gesetzt ist, kann die Client-Konfiguration - Unterstützung das Netzwerk nicht durchsuchen.

- **SEARCH**

Aktiviert alle Funktionen des Discovery-Modus KNOWN und zusätzlich die Netzwerksuche.

Das Symbol „Andere Systeme (Netzwerk durchsuchen)“ wird nur angezeigt, wenn die entsprechende Option ausgewählt wird. Dies ist die Standardeinstellung.

- **DISABLE**

Inaktiviert das Aufspüren (Discovery). In diesem Fall steht die Option **Netzwerk durchsuchen** im „Assistent: Datenbank hinzufügen“ nicht zur Verfügung.

Anmerkung: Der Parameter *discover* wird auf allen Client- und Serverexemplaren standardmäßig auf SEARCH gesetzt. Der Parameter *discover* wird auf allen DB2-Verwaltungsservern (DAS) standardmäßig auf SEARCH gesetzt. Dies gilt nicht für DAS, die in einer UNIX-Umgebung mit Enterprise - Extended Edition installiert sind. Dort wird *discover* standardmäßig auf KNOWN gesetzt.

Zusätzliche Einstellungen für den Discovery-Modus SEARCH: Der Discovery-Modus SEARCH erfordert, dass der Parameter *discover_comm* auf dem Server (in der Konfigurationsdatei des DB2-Verwaltungsservers) und auf dem Client (in der Konfigurationsdatei des Datenbankmanagers) definiert wird.

Der Parameter *discover_comm* steuert die Übertragungsprotokolle, über die der Server Suchanforderungen von Clients empfängt, und die Übertragungsprotokolle, die von Clients zum Übermitteln von Suchanforderungen verwendet werden. Der Parameter *discover_comm* kann auf TCP/IP oder NetBIOS eingestellt werden. Momentan werden nur diese Protokolle unterstützt.

Auf dem DAS müssen die für *discover_comm* angegebenen Werte den für DB2COMM festgelegten Werten entsprechen oder eine Untergruppe dieser Werte sein.

Anmerkung: Stellen Sie zur Vermeidung von Problemen mit der Steuerzentrale und der Client-Konfiguration - Unterstützung sicher, dass die Variable DB2COMM der Profilregistrierdatenbank in der DB2-Registrierdatenbank mit dem Befehl **db2set** festgelegt wird. Es wird nicht empfohlen, für das Festlegen der Variablen DB2COMM der Profilregistrierdatenbank eine andere Methode zu verwenden.

Auf dem Server wird der Parameter *discover_comm* in der DAS-Konfigurationsdatei festgelegt. Auf dem Client (oder auf einem als Client verwendeten Server) wird der Parameter *discover_comm* in der Konfigurationsdatei des Datenbankmanagers festgelegt.

Anmerkung: Bei Verwendung des Discovery-Modus SEARCH muss mindestens ein vom Parameter *discover_comm* auf dem Client angegebenes Protokoll mit den im Parameter *discover_comm* auf dem DAS angegebenen Protokollen übereinstimmen. Ist keine solche Übereinstimmung vorhanden, reagiert der Server nicht auf die Anforderungen des Clients.

Geben Sie folgenden Befehl ein, um die Einstellungen für die Variable DB2COMM der Profilregistrierdatenbank zu prüfen:

```
db2set db2comm
```

Darüber hinaus können zwei Variablen der DB2-Profilregistrierdatenbank dazu verwendet werden, den Discovery-Modus SEARCH über NetBIOS auf dem Client anzupassen: DB2DISCOVERYTIME und DB2NBDISCOVERYRECVBUFS. In den meisten Fällen können die Standardwerte für diese beiden Variablen der Profilregistrierdatenbank eingesetzt werden.

Die Variablen DB2DISCOVERYTIME und DB2NBDISCOVERRCVBUFS der Profilregistrierdatenbank werden auf dem Client-Exemplar (oder auf einem als Client verwendeten Server) festgelegt. Gehen Sie wie folgt vor, um die Variablen der Profilregistrierdatenbank festzulegen:

- Zum Festlegen des Registrierungswertes für DB2DISCOVERYTIME auf 60 Sekunden geben Sie den folgenden Befehl ein:

```
db2set db2discoverytime=60
```

Dieser Wert gibt an, dass der Discovery-Modus SEARCH 60 Sekunden auf eine Server-Antwort warten soll.

- Zum Festlegen des Registrierungswertes für DB2NBDISCOVERRCVBUFS auf den Wert 20 geben Sie den folgenden Befehl ein:

```
db2set db2nbdiscoverrcvbufs=20
```

Dieser Wert gibt an, wie viele NetBIOS-Puffer für gleichzeitig eingehende Antwortnachrichten von aufgespürten Servern zugeordnet werden.

Aufspüren (Discovery) von Serverexemplaren und Datenbanken inaktivieren: Möglicherweise arbeiten Sie mit mehreren Exemplaren, die jeweils mehrere Datenbanken pro Server enthalten. Es kann wünschenswert sein, einige dieser Exemplare bzw. Datenbanken vom Aufspüren (Discovery) auszuschließen.

Damit Clients die Serverexemplare auf einem System lokalisieren können, setzen Sie den Konfigurationsparameter *discover_inst* des Datenbankmanagers auf jedem Serverexemplar auf AKTIVIEREN (dies ist der Standardwert). Setzen Sie

diesen Parameter auf DISABLE, um das betreffende Exemplar und dessen Datenbanken vom Aufspüren (Discovery) auszuschließen.

Damit eine Datenbank von einem Client aufgespürt werden kann, setzen Sie den Datenbankkonfigurationsparameter *discover_db* auf AKTIVIEREN (dies ist der Standardwert). Setzen Sie diesen Parameter auf DISABLE, um die Datenbank vom Aufspüren (Discovery) auszuschließen.

Festlegen der Discovery-Parameter: Die Parameter *discover* und *discover_comm* werden in der DAS-Konfigurationsdatei auf dem Server-System und in der Konfigurationsdatei des Datenbankmanagers auf dem Client festgelegt. Gehen Sie wie folgt vor, um diese Parameter festzulegen:

- Auf dem DAS:

Aktualisieren Sie die DAS-Konfigurationsdatei mit folgender Befehlsfolge:

```
update admin cfg using discover [ DISABLE | KNOWN |  
SEARCH ]  
update admin cfg using discover_comm [ NETBIOS | TCPIP ]
```

Stoppen Sie den DAS, und starten Sie ihn erneut mit folgenden Befehlen:

```
db2admin stop  
db2admin start
```

Anmerkung: Der Discovery-Modus SEARCH funktioniert nur unter NetBIOS und TCP/IP.

- Mit Hilfe der Steuerzentrale:

1. Starten Sie die Client-Konfiguration - Unterstützung.
2. Klicken Sie den Druckknopf **Client-Einstellungen** an.
3. Wählen Sie die Indexzunge **Kommunikation** aus.
4. Wählen Sie die zu ändernden Parameter im Fenster **Parameter** aus.
5. Wählen Sie im Feld **Wert** einen Wert für den zu ändernden Parameter aus.
6. Klicken Sie den Druckknopf **OK** an, um das Fenster **Client-Einstellungen** zu schließen. Ein DB2-Nachrichtenfenster wird geöffnet.
7. Klicken Sie den Druckknopf **OK** an, und starten Sie Ihre Anwendungen erneut, damit die Änderungen wirksam werden.

Anmerkung: Wenn der Parameter *discover_comm* die Option NETBIOS beinhaltet, müssen Sie sicherstellen, dass der Parameter für den Namen der Workstation (*nname*) für den Client und den DAS festgelegt ist. Außerdem müssen Sie sicherstellen, dass die Variable DB2NBADAPTERS der Profilregistrierdatenbank auf die Adaptersnummer gesetzt ist, die Sie verwenden wollen.

Legen Sie die Parameter *discover_inst* und *discover_db* in der Steuerzentrale fest:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Exemplare** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Exemplar und wählen Sie dann im Kontextmenü **Konfigurieren** aus.
3. Wählen Sie auf der Seite „Umgebung“ den Parameter *discover_inst* aus.
4. Um die Lokalisierung des Serverexemplars durch den Client zu ermöglichen, müssen Sie **Aktivieren** auswählen und anschließend auf **OK** klicken.
5. Klicken Sie mit der rechten Maustaste in der Objektbaumstruktur auf der Datenbank und wählen Sie dann im Kontextmenü **Konfigurieren** aus.
6. Wählen Sie auf der Seite „Umgebung“ den Parameter *discover_db* aus.
7. Um die Lokalisierung der Datenbank durch den Client zu ermöglichen, müssen Sie **Aktivieren** auswählen und anschließend auf **OK** klicken.

Konfigurieren des DAS zur Verwendung der Steuerzentrale und der Client-Konfiguration - Unterstützung

Sie müssen DB2 Discovery zum Abrufen von Informationen über die in Ihrem Netzwerk enthaltenen Systeme konfigurieren. DB2 Discovery ist eine Einrichtung, die von der Client-Konfiguration - Unterstützung und der Steuerzentrale verwendet wird. Zu den für diese Einrichtung erforderlichen Konfigurationsschritten gehört möglicherweise auch das Aktualisieren der Exemplarlisten und der DAS-Konfiguration, um sicherzustellen, dass DB2 Discovery die richtigen Informationen abrufen.

Aktualisieren von Exemplarlisten: Ein DB2-Verwaltungsserver (DAS) kennt möglicherweise nicht alle Exemplare in einem partitionierten Datenbanksystem, da beim Erstellen eines Exemplars zunächst nur der DAS der Exemplareigenmaschine das Exemplar kennt.

Wenn Sie ein Exemplar auf einer Maschine erstellt haben, die nicht über einen DAS verfügt, können Sie auf dieser Maschine nachträglich einen DAS erstellen, um das Exemplar bekannt zu machen.

Führen Sie die folgenden Schritte aus, wenn Sie mehrere DAS erstellt haben und erreichen wollen, dass jeder DAS alle Exemplare Ihres partitionierten Datenbanksystems kennt:

1. Gehen Sie für jeden DAS wie folgt vor:

Führen Sie den Befehl **db2ilist** auf der Verwaltungsservermaschine aus, um eine Liste der diesem DAS bekannten Exemplare aufzurufen.

Anmerkung: Wenn die angezeigte Liste vollständig ist, können Sie die nachfolgenden Schritte überspringen und mit dem nächsten Abschnitt fortfahren.

2. **Gehen Sie für jedes Exemplar, das im vorigen Schritt in der Exemplarliste fehlt, wie folgt vor:**

Führen Sie auf der Exempleareignermaschine den Befehl **db2nlist** aus, um festzustellen, ob ein Eintrag für die Maschine vorhanden ist, die den DAS enthält. Ist kein solcher Eintrag vorhanden, führen Sie den Befehl **db2ncrt** aus, um diese Maschine dem Exemplar hinzuzufügen.

Anmerkung: Das gemeinsame Netzwerklaufwerk für das Exemplar muss auf der DAS-Maschine verfügbar sein.

Aktualisieren der Konfiguration des DB2-Verwaltungsservers

Das Konfigurationsprogramm setzt die Variable DB2SYSTEM der Profilregistrierdatenbank standardmäßig auf den Namen des Windows NT-Computers. Discovery ruft die Namen der Systeme ab, auf denen sich ein DB2-Verwaltungsserver (DAS) befindet. Discovery verwendet diese Namen beim Herstellen von Verbindungen als Koordinatorknoten.

Die DAS-Konfiguration kann auf folgende zwei Arten aktualisiert werden:

- Wenn Sie ermöglichen wollen, dass ein Koordinatorknoten aus einer Liste der DB2-Systeme ausgewählt werden kann, legen Sie in allen Konfigurationsdateien der DB2-Verwaltungsserver (die Standardeinstellung) DISCOVER=SEARCH fest.

Wenn mehrere DAS vorhanden sind, kann es vorkommen, dass ein Exemplar auf mehreren Systemen in der Schnittstelle der Client-Konfiguration - Unterstützung oder der Steuerzentrale erscheint. Jedes System verwendet jedoch einen anderen Kommunikationszugriffspfad für die Exemplare. Benutzer können verschiedene DB2-Systeme als Koordinatorknoten für die Übertragung auswählen und dadurch die Auslastung umverteilen.

- Wenn Sie nicht zulassen wollen, dass Benutzer den Koordinatorknoten auswählen können, legen Sie auf allen DAS die Einstellung DISCOVER=KNOWN fest. Definieren Sie lediglich für einen DAS der DAS-Konfiguration die Einstellung DISCOVER=SEARCH. Discovery verwendet den Datenbankpartitionsserver, auf dem sich der DAS befindet, als Koordinatorknoten zum Herstellen von Verbindungen.

Erstellen einer Knotenkonfigurationsdatei

Wenn Ihre Datenbank in einer partitionierten Datenbankumgebung arbeiten soll, müssen Sie eine Knotenkonfigurationsdatei mit dem Namen db2nodes.cfg erstellen. Diese Datei muss sich im Unterverzeichnis sqllib des Benutzerverzeichnisses (Home) für das Exemplar befinden, bevor Sie den Datenbankmanager mit der Fähigkeit zur Parallelverarbeitung in mehreren Partitionen starten können.

Die Datei enthält Konfigurationsdaten für alle Datenbankpartitionen eines Exemplars und wird von allen Datenbankpartitionen für dieses Exemplar gemeinsam benutzt.

Überlegungen zu Windows NT: Wenn Sie DB2 Enterprise - Extended Edition unter Windows NT verwenden, wird die Knotenkonfigurationsdatei beim Erstellen des Exemplars erstellt. Sie sollten nicht versuchen, eine Knotenkonfigurationsdatei manuell zu erstellen oder manuell zu ändern.

Anmerkung: Erstellen Sie keine anderen als die von DB2 erstellten Dateien oder Verzeichnisse unter dem Unterverzeichnis `sql1ib`, um Datenverlust zu vermeiden, wenn ein Exemplar gelöscht wird. Es gibt jedoch zwei Ausnahmen. Wenn Ihr System gespeicherte Prozeduren unterstützt, stellen Sie die gespeicherten Prozeduren in das Unterverzeichnis `function` im Unterverzeichnis `sql1ib`. (Informationen zu gespeicherten Prozeduren finden Sie im Abschnitt für „Gespeicherte Prozeduren“ im Handbuch *Systemverwaltung: Optimierung*.) Die andere Ausnahme betrifft die erstellten benutzerdefinierten Funktionen (UDFs). Benutzerdefinierte Funktionen können im selben Verzeichnis gespeichert werden.

Die Datei enthält für jede Datenbankpartition, die zu einem Exemplar gehört, eine Zeile. Jede Zeile hat folgendes Format:

```
knotennummer host-name [logischer-anschluss [netzname]]
```

Die Token einer Zeile sind durch Leerzeichen voneinander getrennt. Die Variablen sind:

knotennummer

Die Knotennummer (mögliche Werte: 0 - 999) definiert einen Knoten eindeutig. Knotennummern müssen in aufsteigender Reihenfolge angegeben sein. Es dürfen Sprünge in der Folge der Knotennummern auftreten. Wenn eine Knotennummer einmal zugeordnet ist, kann sie nicht mehr geändert werden. (Ansonsten könnten die Informationen in der Partitionierungszuordnung, die bestimmt, wie Daten partitioniert werden, inkonsistent werden.) Wenn Sie einen Knoten löschen, kann seine Knotennummer für jeden neuen Knoten, den Sie hinzufügen, wieder verwendet werden. Die Knotennummer wird zur Generierung eines Knotennamens im Datenbankverzeichnis verwendet. Er hat folgendes Format:

```
NODEnnnn
```

Die Ziffernfolge *nnnn* ist die Knotennummer, die links mit Nullen aufgefüllt wird. Diese Knotennummer wird außerdem in den Befehlen CREATE DATABASE und DROP DATABASE verwendet.

host-name

Der Host-Name der IP-Adresse für die partitionsübergreifende Kommunikation. (Es gibt eine Ausnahme, wenn der netzname angegeben wird. In diesem Fall wird der netzname für den Großteil der Kommunikation verwendet, während host-name nur für die Befehle DB2START, DB2STOP und db2_all verwendet wird.)

logischer-anschluss

Dieser Parameter ist wahlfrei und gibt die logische Anschlussnummer (Port) für den Knoten an. Diese Nummer wird mit dem Exemplarnamen des Datenbankmanagers verwendet, um einen TCP/IP-Serviceeintrag in der Datei etc/services zu identifizieren.

Die Kombination aus IP-Adresse und logischem Anschluss wird als allgemein bekannte Adresse verwendet und muss für alle Anwendungen eindeutig sein, um die Verbindungen zur Kommunikation zwischen Knoten zu unterstützen.

Für jeden *host-namen* muss ein *logischer-anschluss* entweder 0 (Null) oder leer (was standardmäßig einen Wert von 0 bedeutet) sein. Der Knoten, dem dieser *logische-anschluss* zugeordnet ist, ist der Standardknoten auf dem Host, zum dem Clients die Verbindung herstellen. Diese Einstellung kann mit Hilfe der Umgebungsvariablen DB2NODE in der Prozedur db2profile oder mit der API sqlesetc() überschrieben werden.

Wenn Sie mehrere Knoten auf demselben Host haben (d. h. mehr als eine *knotennummer* für einen Host), sollten Sie die Nummern für den *logischen-anschluss* den logischen Knoten in aufsteigender Reihenfolge von 0 an ohne Sprünge zuordnen.

netzname

Dieser Parameter ist wahlfrei und wird zur Unterstützung eines Hosts verwendet, der über mehr als eine aktive TCP/IP-Schnittstelle verfügt, von denen jede ihren eigenen Host-Namen hat.

Das folgende Beispiel zeigt eine mögliche Knotenkonfigurationsdatei für ein System RS/6000 SP, in dem SP2EN1 mehrere TCP/IP-Schnittstellen und zwei logische Knoten hat und SP2SW1 als Schnittstelle für DB2 Universal Database verwendet. Es zeigt außerdem die Knotennummern, beginnend bei 1 (und nicht bei 0), sowie einen Sprung in der Folge unter *knotennummer*:

<i>knotennummer</i>	<i>host-name</i>	<i>logischer-anschluss</i>	<i>netzname</i>
1	SP2EN1	0	SP2SW1
2	SP2EN1	1	SP2SW1
4	SP2EN2	0	
5	SP2EN3		

Die Datei `db2nodes.cfg` kann mit einem beliebigen Editor aktualisiert werden. (Die Ausnahme bildet hierbei, dass unter Windows NT kein Editor eingesetzt werden sollte.) Sie müssen jedoch sorgfältig darauf achten, dass die Integrität der Daten in der Datei erhalten bleibt, da für die Partitionierung vorausgesetzt wird, dass die Knotennummer nicht geändert wird. Die Knotenkonfigurationsdatei wird gesperrt, wenn der Befehl `DB2START` ausgeführt wird, und entsperrt, wenn der Befehl `DB2STOP` den Datenbankmanager beendet. Der Befehl `DB2START` kann die Datei bei Bedarf aktualisieren, während sie gesperrt ist. Sie können beispielsweise den Befehl `DB2START` mit der Option `RESTART` oder `ADDNODE` ausführen.

Anmerkung: Wenn der Befehl `DB2STOP` nicht erfolgreich ausgeführt und die Knotenkonfigurationsdatei nicht entsperrt wird, führen Sie den Befehl `DB2STOP FORCE` aus, um sie zu entsperren.

Erstellen der Datenbankkonfigurationsdatei

Für jede Datenbank wird weiterhin eine *Datenbankkonfigurationsdatei* erstellt. Diese Datei wird automatisch für Sie erstellt. Diese Datei enthält Werte für verschiedene *Konfigurationsparameter*, die sich auf die Verwendung der Datenbank auswirken, wie die folgenden:

- Parameter, die bei der Erstellung der Datenbank angegeben und/oder verwendet werden (z. B. Codepage der Datenbank, Sortierfolge, DB2-Release-Stand)
- Parameter, die den aktuellen Status der Datenbank anzeigen (z. B. Anzeiger für Sicherung anstehend, Datenbankkonsistenz, aktualisierende Wiederherstellung anstehend)
- Parameter, die die Menge an Systemressourcen definieren, die für den Betrieb der Datenbank verwendet werden sollen (z. B. Pufferpoolgröße, Datenbankprotokollierung, Sortierspeichergröße)

Eine detaillierte Beschreibung dieser Parameter finden Sie im Abschnitt zum „Konfigurieren von DB2“ im Handbuch *Systemverwaltung: Optimierung*.

Sie sollten nicht versuchen, die Parameter in der Konfigurationsdatei manuell zu ändern. Sie sollten nur die unterstützte Schnittstelle verwenden.

Hinweis zur Leistung: Viele Konfigurationsparameter verfügen zwar über Standardwerte, jedoch können sie auch aktualisiert werden, um eine optimale Leistung für Ihre Datenbank zu erzielen.

Bei mehreren Partitionen: Wenn Sie eine Datenbank haben, die auf mehr als eine Partition partitioniert ist, sollte die Konfigurationsdatei in allen Datenbankpartitionen die gleiche sein. Diese Konsistenz ist erforderlich, da der SQL-Compiler verteilte SQL-Anweisungen anhand der Informationen in der Konfigurationsdatei des lokalen Knotens kompiliert und einen Zugriffsplan erstellt, der die Anforderungen der jeweiligen SQL-Anweisung erfüllt. Wenn sich verschiedene Konfigurationsdateien in den Datenbankpartitionen befinden, kann dies je nachdem, in welcher Datenbankpartition die Anweisung vorbereitet wird, zu verschiedenen Zugriffsplänen führen. Verwenden Sie `db2_all`, um die Synchronisation der Konfigurationsdateien bei allen Datenbankpartitionen zu gewährleisten.

Replizieren von Konfigurationsdaten durch Antwortdateien

Mit dem Dienstprogramm `db2rspgn` kann eine Antwortdatei erstellt werden, die zum erneuten Installieren Ihres Systems verwendet werden kann oder zum Replizieren der Variablen der Profilregistrieredatenbank, Datenbankmanager-Konfigurationsparameter und der Konfigurationsparameter für Datenbankverwaltung des aktuellen Systems auf ein identisches System.

Nach dem Installieren eines Systems mit einem oder mehreren DB2-Produkten und dem Anpassen von Parametern für die verwendete Umgebung können Sie mit `db2rspgn` die erforderlichen Werte in einer Antwortdatei generieren. Mit Hilfe dieser Antwortdatei kann ein identisches System erstellt werden.

Die Befehlszeilensyntax gibt das Zielverzeichnis für die Antwortdatei(en) und alle unterstützenden Dateien an. Außerdem können Sie wahlfrei die zu kopierenden Exemplare angeben sowie (ebenfalls wahlfrei) das Verwaltungs-exemplar und/oder das DataLinks-Serverexemplar inaktivieren. Weitere Informationen zu Fragen der Massenimplementierung finden Sie in *Administering Satellites Guide and Reference*.

Im entsprechenden Handbuch *Einstieg* finden Sie Einzelheiten zur Syntax dieses Dienstprogramms und Angaben zur Verwendung der generierten Antwortdateien.

Aktivieren der FCM-Kommunikation

In einer partitionierten Datenbankumgebung werden der Großteil der Kommunikation zwischen Datenbankpartitionen von FCM (Fast Communications Manager) verarbeitet. Zur Aktivierung von FCM in einer Datenbankpartition und zur Ermöglichung der Kommunikation mit anderen Datenbankpartitionen müssen Sie einen Serviceeintrag in der Datei `services` des Verzeichnisses etc der Partition erstellen. Die entsprechende Vorgehensweise wird im Folgenden erläutert. FCM verwendet den angegebenen Anschluss (Port) für die Kommunikation. Wenn mehrere Partitionen auf demselben Host definiert sind, müssen Sie einen Bereich von Anschlüssen, wie unten gezeigt, definieren.

Überlegungen zu Windows NT

Wenn Sie DB2 Enterprise - Extended Edition in der Windows NT-Umgebung einsetzen, wird der TCP/IP-Anschlussbereich durch folgende Programme automatisch der Servicedatei hinzugefügt:

- Das Installationsprogramm, wenn es das Exemplar erstellt oder einen neuen Knoten hinzufügt.
- Das Dienstprogramm db2icrt, wenn es ein neues Exemplar erstellt.
- Das Dienstprogramm db2ncrt, wenn es den ersten Knoten auf der Maschine hinzufügt.

Weitere Informationen finden Sie im Handbuch *DB2 Enterprise - Extended Edition für Windows Einstieg*.

Der Serviceeintrag hat folgende Syntax:

```
DB2_exemplar anschluss/tcp #kommentar
```

DB2_exemplar

Der Wert für *exemplar* ist der Name des Datenbankmanagerexemplars. Alle Zeichen des Namens müssen in Kleinschreibung angegeben sein. Wenn der Exemplarname db2puser wäre, müssten Sie DB2_db2puser angeben.

anschluss/tcp

Der TCP/IP-Anschluss, den Sie für die Datenbankpartition reservieren wollen.

#kommentar

Ein beliebiger Kommentar, den Sie dem Eintrag hinzufügen wollen. Dem Kommentar muss ein Nummernzeichen (#) vorangestellt werden.

Wenn die Datei */etc/services* gemeinsam benutzt wird, müssen Sie sicherstellen, dass die Anzahl der in der Datei zugeordneten Anschlüsse größer oder gleich der größten Anzahl mehrerer Datenbankpartitionen in diesem Exemplar ist. Stellen Sie bei der Zuordnung von Anschlüssen außerdem sicher, dass Sie jeden möglichen Prozessor, der als Ersatz verwendet werden kann, mit berücksichtigen.

Wenn die Datei */etc/services* nicht gemeinsam benutzt wird, gelten dieselben Überlegungen, jedoch mit einem zusätzlichen Gesichtspunkt: Sie müssen sicherstellen, dass die für das DB2-Exemplar definierten Einträge in allen Dateien */etc/services* die gleichen sind (obwohl andere Einträge, die sich nicht auf Ihre partitionierte Datenbank beziehen, nicht gleich sein müssen).

Wenn Sie mehrere Datenbankpartitionen auf demselben Host in einem Exemplar haben, müssen Sie mehr als einen Anschluss für die Verwendung durch FCM definieren. Dazu müssen zwei Zeilen in die Datei `etc/services` eingefügt werden, die den Bereich von Anschlüssen, den Sie zuordnen, angeben. Die erste Zeile gibt den ersten Anschluss an, während die zweite Zeile das Ende des Blocks von Anschlüssen angibt. Im folgenden Beispiel werden fünf Anschlüsse für das Exemplar `sales` zugeordnet. Dies bedeutet, dass kein Prozessor im Exemplar mehr als fünf Datenbankpartitionen hat.

```
DB2_sales      9000/tcp
DB2_sales_END  9004/tcp
```

Anmerkung: Die Zeichenfolge `END` muss unbedingt durchgehend in Großschreibung angegeben werden. Außerdem müssen unbedingt beide Unterstreichungszeichen (`_`) vorhanden sein.

Kapitel 3. Erstellen einer Datenbank

Dieses Kapitel enthält eine kurze Erläuterung der verschiedenen Objekte, die Bestandteil der Implementierung Ihres Datenbankentwurfs sein können.

Das vorhergehende Kapitel behandelte schwerpunktmäßig die Informationen, die vor dem Erstellen einer Datenbank bekannt sein müssen. Es enthielt außerdem Informationen zu verschiedenen Themen und Aufgaben, die vor dem Erstellen einer Datenbank berücksichtigt bzw. ausgeführt werden müssen.

Das letzte Kapitel im vorliegenden Teil enthält Informationen, die vor dem Ändern einer Datenbank berücksichtigt werden müssen. Darüber hinaus erläutert das Kapitel das Ändern und Löschen von Datenbankobjekten.

Beim Erstellen einer Datenbank werden die folgenden Tasks für Sie ausgeführt:

- Definieren aller Systemkatalogtabellen, die von der Datenbank benötigt werden.
- Zuordnen des Protokolls für die Datenbankwiederherstellung.
- Erstellen der Datenbankkonfigurationsdatei und Definieren der Standardwerte.
- Binden der Datenbankdienstprogramme an die Datenbank.

Die folgenden Datenbankzugriffsrechte werden der Gruppe PUBLIC automatisch erteilt: `CREATETAB`, `BINDADD`, `CONNECT`, `IMPLICIT_SCHEMA` und `SELECT` für die Systemkatalogsichten.

Gehen Sie wie folgt vor, um eine Datenbank mit der Steuerzentrale zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Datenbanken** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Ordner **Datenbanken** und wählen Sie dann im Kontextmenü **Erstellen** → **Datenbank mit Assistent** aus.
3. Führen Sie die angegebenen Schritte aus, um die Task abzuschließen.

Mit dem folgenden Befehl des Befehlszeilenprozessors wird eine Datenbank namens `person1` an der Standardspeicherposition mit dem zusätzlichen Kommentar „Personnel DB for BSchiefer Co“ erstellt.

```
create database person1
with "Personnel DB for BSchiefer Co"
```

Wenn Sie eine Datenbank in einem anderen, möglicherweise fernen Datenbankmanagerexemplar erstellen wollen, lesen Sie die Informationen im Abschnitt „Verwenden mehrerer Exemplare des Datenbankmanagers“ auf Seite 60. Dieser Abschnitt enthält auch eine Einführung für den Befehl, den Sie verwenden müssen, wenn Sie Verwaltungsoperationen auf Exemplarebene für andere Exemplare als Ihr Standardexemplar, einschließlich der fernen Exemplare, ausführen möchten.

Anmerkung: Im Handbuch *Command Reference* finden Sie Informationen zur Standardspeicherposition von Datenbanken (Default Location) und zur Angabe einer anderen Speicherposition mit Hilfe des Befehls CREATE DATABASE.

Die bei der Erstellung einer Datenbank vom Datenbankmanager oder von Ihnen auszuführenden Operationen werden in den folgenden Abschnitten behandelt:

- „Definition der ersten Knotengruppen“ auf Seite 117
- „Definition der ersten Tabellenbereiche“ auf Seite 117
- „Definition von Systemkatalogtabellen“ auf Seite 119
- „Definition von Datenbankverzeichnissen“ auf Seite 120
- „DCE Verzeichnisservices“ auf Seite 122
- „LDAP-Verzeichnisservices (Lightweight Directory Access Protocol)“ auf Seite 122
- „Definition des Wiederherstellungsprotokolls der Datenbank“ auf Seite 124
- „Binden von Dienstprogrammen an die Datenbank“ auf Seite 124
- „Katalogisieren einer Datenbank“ auf Seite 125
- „Erstellen von Knotengruppen“ auf Seite 123
- „Erstellen eines Tabellenbereichs“ auf Seite 126
- „Erstellen eines Schemas“ auf Seite 133
- „Erstellen und Auffüllen einer Tabelle“ auf Seite 135
- „Erstellen eines Auslösers“ auf Seite 156
- „Erstellen einer benutzerdefinierten Funktion (UDF) oder Methode“ auf Seite 159
- „Erstellen eines benutzerdefinierten Datentyps (UDT)“ auf Seite 163
- „Erstellen einer Sicht“ auf Seite 166
- „Erstellen einer Übersichtstabelle“ auf Seite 169
- „Erstellen eines Aliasnamens“ auf Seite 172
- „Erstellen einer Oberfläche“ auf Seite 175

- „Erstellen eines Servers“ auf Seite 176
- „Erstellen eines Kurznamens“ auf Seite 185
- „Erstellen eines Index, einer Indexerweiterung oder einer Indexspezifikation“ auf Seite 188

Weitere Informationen zur physischen Implementierung der Datenbank finden Sie im Handbuch *Systemverwaltung: Konzept*.

Definition der ersten Knotengruppen

Wenn eine Datenbank anfangs erstellt wird, werden Datenbankpartitionen für alle in der Datei `db2nodes.cfg` angegebenen Partitionen erstellt. Andere Partitionen können mit den Befehlen `ADD NODE` und `DROP NODE` hinzugefügt bzw. gelöscht werden.

Es werden drei Knotengruppen definiert:

- `IBMCATGROUP` ist die Knotengruppe für den Tabellenbereich `SYS-CATSPACE`, in dem die Systemkatalogtabellen gespeichert werden.
- `IBMTEMPGROUP` ist die Knotengruppe für den Tabellenbereich `TEMPSPACE1`, in dem temporäre Tabellen gespeichert werden, die bei Datenbankoperationen erstellt werden.
- `IBMDEFAULTGROUP` ist die Knotengruppe für den Tabellenbereich `USERSPACE1`, der standardmäßig die Benutzertabellen und Indizes enthält.

Definition der ersten Tabellenbereiche

Bei der Erstellung einer Datenbank werden folgende drei Tabellenbereiche definiert:

- Der Bereich `SYSCATSPACE` für die Tabellen des Systemkatalogs (siehe „Definition von Systemkatalogtabellen“ auf Seite 119)
- Der Bereich `TEMPSPACE1` für temporäre Systemtabellen, die während der Datenbankverarbeitung erstellt werden
- Der Bereich `USERSPACE1` für benutzerdefinierte Tabellen und Indizes

Anmerkung: Wenn Sie eine Datenbank zum ersten Mal erstellen, wird für diese kein temporärer Benutzertabellenbereich erstellt.

Wenn im Befehl `CREATE DATABASE` keine Parameter zum Tabellenbereich angegeben werden, erstellt der Datenbankmanager diese Tabellenbereiche unter Verwendung von Verzeichnisbehältern, die vom Betriebssystem verwaltet werden (SMS). Diese Verzeichnisbehälter werden in dem Unterverzeichnis erstellt, das für die Datenbank angelegt wird (weitere Informationen zu physi-

schen Datenbankverzeichnissen enthält das Handbuch *Systemverwaltung: Konzept*). Als Wert des Parameters EXTENTSIZE für diese Tabellenbereiche wird der Standardwert verwendet.

Gehen Sie wie folgt vor, um Tabellenbereiche zum ersten Mal mit der Steuerzentrale zu definieren:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Datenbanken** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Ordner **Datenbanken** und wählen Sie dann im Kontextmenü **Erstellen** → **Datenbank mit Assistent** aus.
3. Führen Sie die angegebenen Schritte aus, um die Task abzuschließen.

Geben Sie in der Befehlszeile Folgendes ein, um Tabellenbereiche zum ersten Mal zu definieren:

```
CREATE DATABASE <name>
  CATALOG TABLESPACE
    MANAGED BY SYSTEM USING ('<pfad>')
    EXTENTSIZE <wert> PREFETCHSIZE <wert>
  USER TABLESPACE
    MANAGED BY DATABASE USING (FILE'<pfad>' 5000,
                                FILE'<pfad>' 5000)
    EXTENTSIZE <wert> PREFETCHSIZE <wert>
  TEMPORARY TABLESPACE
    MANAGED BY SYSTEM USING ('<pfad>')
  WITH "<kommentar>"
```

Wenn Sie die Standarddefinition für diese Tabellenbereiche nicht verwenden möchten, können Sie die Merkmale im Befehl CREATE DATABASE angeben. Zum Beispiel könnte der folgende Befehl zur Erstellung der Datenbank auf einem OS/2-System verwendet werden:

```
CREATE DATABASE PERSONL
  CATALOG TABLESPACE
    MANAGED BY SYSTEM USING ('d:\pcatalog','e:\pcatalog')
    EXTENTSIZE 16 PREFETCHSIZE 32
  USER TABLESPACE
    MANAGED BY DATABASE USING (FILE'd:\db2data\personl' 5000,
                                FILE'd:\db2data\personl' 5000)
    EXTENTSIZE 32 PREFETCHSIZE 64
  TEMPORARY TABLESPACE
    MANAGED BY SYSTEM USING ('f:\db2temp\personl')
  WITH "Personnel DB for BSchiefer Co"
```

In diesem Beispiel werden die Definitionen für jeden der ersten Tabellenbereiche explizit angegeben. Definitionen müssen nur für die Tabellenbereiche angegeben werden, für die die Standarddefinition nicht verwendet werden soll.

Die Codierung des Ausdrucks `MANAGED BY` im Befehl `CREATE DATABASE` folgt demselben Format wie beim Ausdruck `MANAGED BY` im Befehl `CREATE TABLESPACE`. Weitere Beispiele finden Sie in „Erstellen eines Tabellenbereichs“ auf Seite 126.

Weitere Informationen hierzu enthält auch das Handbuch *Systemverwaltung: Konzept*. Lesen Sie außerdem vor dem Erstellen Ihrer Datenbank die Informationen zum Entwerfen und Auswählen von Tabellenbereichen.

Definition von Systemkatalogtabellen

Eine Gruppe von Systemkatalogtabellen wird für jede Datenbank erstellt und verwaltet. Diese Tabellen enthalten Informationen über die Definitionen der Datenbankobjekte (z. B. Tabellen, Sichten, Indizes und Pakete) sowie Sicherheitsinformationen über die Art des Zugriffs, der Benutzern auf diese Objekte gewährt wird. Diese Tabellen werden im Tabellenbereich `SYSCATSPACE` gespeichert.

Diese Tabellen werden während des Betriebs einer Datenbank aktualisiert, zum Beispiel, wenn eine Tabelle erstellt wird. Diese Tabellen können nicht explizit erstellt oder gelöscht werden, jedoch kann ihr Inhalt abgefragt werden. Wenn eine Datenbank erstellt wird, werden neben den Objekten der Systemkatalogtabellen auch die folgenden Datenbankobjekte im Systemkatalog definiert:

- Eine Gruppe benutzerdefinierter Funktionen (UDFs) wird im Schema `SYSFUN` erstellt. Weitere Informationen zu diesen vom System erstellten Funktionen finden Sie im Handbuch *SQL Reference*.
- Eine Gruppe von Nur-Lese-Sichten für die Systemkatalogtabellen wird im Schema `SYSCAT` erstellt. Informationen zu diesen Sichten finden Sie im Abschnitt über „Katalogsichten“ des Handbuchs *SQL Reference*.
- Eine Gruppe aktualisierbarer Katalogsichten wird im Schema `SYSSTAT` erstellt. Diese aktualisierbaren Sichten ermöglichen es, bestimmte statistische Informationen zu aktualisieren, um die Leistung einer hypothetischen Datenbank zu testen oder um statistische Daten ohne Verwendung des Dienstprogramms `RUNSTATS` zu aktualisieren. Informationen zu diesen Sichten finden Sie im Abschnitt zu „Aktualisierbaren Katalogsichten“ im Handbuch *SQL Reference*.

Nach der Erstellung der Datenbank ist es vielleicht wünschenswert, den Zugriff auf die Systemkatalogsichten einzuschränken. Informationen dazu finden Sie in „Sicherheit der Systemkatalogsichten“ auf Seite 320.

Definition von Datenbankverzeichnissen

Drei Verzeichnisse werden zur Einrichtung einer neuen Datenbank verwendet.

- Lokales Datenbankverzeichnis
- Systemdatenbankverzeichnis
- Knotenverzeichnis

Lokales Datenbankverzeichnis

Eine Datei für das *lokale Datenbankverzeichnis* ist in jedem Pfad (bzw. unter anderen Betriebssystemen auf jedem „Laufwerk“) vorhanden, in dem eine Datenbank definiert wurde. Dieses Verzeichnis enthält einen Eintrag für jede Datenbank, auf die von dieser Position aus zugegriffen werden kann. Ein Eintrag enthält folgende Daten:

- Den Datenbanknamen, der mit dem Befehl CREATE DATABASE angegeben wurde
- Den Aliasnamen der Datenbank (dieser Name ist mit dem Datenbanknamen identisch, wenn kein Aliasname angegeben wird)
- Einen Kommentar zur Datenbank, der mit dem Befehl CREATE DATABASE angegeben wurde
- Den Namen des Stammverzeichnisses für die Datenbank
- Andere Systeminformationen

Wenn Sie den Inhalt dieser Datei für eine bestimmte Datenbank anzeigen wollen, geben Sie den folgenden Befehl ein, in dem *position* die Speicherposition der Datenbank bezeichnet:

```
LIST DATABASE DIRECTORY ON position
```

Systemdatenbankverzeichnis

Für jedes Exemplar des Datenbankmanagers ist eine Datei mit dem *Systemdatenbankverzeichnis* vorhanden, die einen Eintrag für jede Datenbank enthält, die für dieses Exemplar katalogisiert wurde. Datenbanken werden implizit katalogisiert, wenn der Befehl CREATE DATABASE ausgeführt wird. Sie können auch explizit mit dem Befehl CATALOG DATABASE katalogisiert werden. Informationen zur Katalogisierung von Datenbanken finden Sie in „Katalogisieren einer Datenbank“ auf Seite 125.

Für jede erstellte Datenbank wird dem Verzeichnis ein Eintrag hinzugefügt, der folgende Informationen enthält:

- Den Datenbanknamen, der mit dem Befehl CREATE DATABASE angegeben wurde
- Den Aliasnamen der Datenbank (dieser Name mit dem Datenbanknamen identisch, wenn kein Aliasname angegeben wird)
- Den Kommentar zur Datenbank, der mit dem Befehl CREATE DATABASE angegeben wurde

- Die Speicherposition des *lokalen Datenbankverzeichnis*
- Einen Anzeiger, dass die Datenbank *indirekt* ist, d. h., dass sich die Datenbank auf demselben System wie die Datei für das Systemdatenbankverzeichnis befindet
- Andere Systeminformationen

Wenn Sie den Inhalt dieser Datei anzeigen wollen, setzen Sie den Befehl LIST DATABASE DIRECTORY *ohne* Angabe der Speicherposition der Datei für das Datenbankverzeichnis ab.

In einer partitionierten Datenbankumgebung ist sicherzustellen, dass alle Datenbankpartitionen jederzeit auf dieselbe Datei des Systemdatenbankverzeichnisses `sqlbdbir` im Unterverzeichnis `sqlbdbir` des Benutzerverzeichnisses für das Exemplar zugreifen. Unvorhersehbare Fehler können auftreten, wenn entweder das Systemdatenbankverzeichnis oder die Systemintensionsdatei `sqldbins` im selben Unterverzeichnis `sqlbdbir` symbolische Verbindungen zu einer anderen Datei sind, die sich in einem gemeinsam benutzten Dateisystem befindet. Diese Dateien werden im Abschnitt „Ermöglichen der Datenpartitionierung“ auf Seite 64 beschrieben.

Knotenverzeichnis

Der Datenbankmanager erstellt das Knotenverzeichnis, wenn die erste Datenbankpartition katalogisiert wird. Zum Katalogisieren einer Datenbankpartition wird der Befehl CATALOG NODE verwendet. Zum Auflisten des Inhalts des lokalen Knotenverzeichnisses wird der Befehl LIST NODE DIRECTORY verwendet. Das Knotenverzeichnis wird auf jedem Datenbank-Client erstellt und verwaltet. Das Verzeichnis enthält einen Eintrag für jede ferne Workstation, die eine oder mehrere Datenbanken enthält, auf die der Client zugreifen kann. Der DB2-Client verwendet die Informationen über den Kommunikationsendpunkt im Knotenverzeichnis jedes Mal, wenn eine Datenbankverbindung oder eine Exemplarverbindung (mit ATTACH) angefordert wird.

Die Einträge im Verzeichnis enthalten außerdem Informationen zur Art des Übertragungsprotokolls, das für die Kommunikation zwischen dem Client und der fernen Datenbankpartition verwendet werden soll. Durch das Katalogisieren einer lokalen Datenbankpartition wird ein Aliasname für ein Exemplar erstellt, das sich auf derselben Maschine befindet. Ein lokaler Knoten sollte katalogisiert werden, wenn es mehr als ein Exemplar auf derselben Workstation gibt, auf das vom Client des Benutzers aus zugegriffen werden soll.

DCE Verzeichnisservices

DCE ist eine OSF-Architektur (Open Systems Foundation), die Tools und Services zur Verfügung stellt, um die Erstellung, Verwendung und Wartung von Anwendungen in einer verteilten, heterogenen Datenverarbeitungs Umgebung zu unterstützen. Sie stellt eine Schicht zwischen dem Betriebssystem, dem Netzwerk und einer verteilten Anwendung dar, die es Client-Anwendungen ermöglicht, auf ferne Server zuzugreifen.

Bei lokalen Verzeichnissen wird die physische Position der Zieldatenbank jeweils auf jeder Client-Workstation im Datenbankverzeichnis und im Knotenverzeichnis gespeichert. Daher kann das Aktualisieren bzw. Ändern dieser Verzeichnisse einen erheblichen Zeitaufwand für einen Datenbankadministrator mit sich bringen. Die DCE Verzeichnisservices stellen alternativ zu diesen lokalen Verzeichnissen ein zentrales Verzeichnis bereit. Dadurch ist es möglich, die Informationen zu einer Datenbank oder einem Datenbankmanagerexemplar einmal an einer zentralen Stelle zu speichern und alle Änderungen bzw. Aktualisierungen auf diese eine Stelle zu beschränken.

DCE ist keine Voraussetzung für die Verwendung von DB2. Wenn Sie jedoch in einer DCE-Umgebung arbeiten, lesen Sie die Informationen in „Anhang B. Verwenden der DCE-Verzeichnisservices“ auf Seite 377.

LDAP-Verzeichnisservices (Lightweight Directory Access Protocol)

LDAP (Lightweight Directory Access Protocol) ist eine Standardmethode zum Zugriff auf Verzeichnisservices. Bei einem Verzeichnisservice handelt es sich um ein Repository mit Ressourceninformationen zu mehreren Systemen und Services innerhalb einer verteilten Umgebung. Er stellt den Client-/Server-Zugriff auf diese Ressourcen bereit. Jedes Datenbankserverexemplar wird seine Existenz auf einem LDAP-Server veröffentlichen und dem LDAP-Verzeichnis Datenbankinformationen zur Verfügung stellen, wenn die Datenbanken erstellt werden. Wenn ein Client eine Verbindung zur Datenbank herstellt, können die Kataloginformationen für den Server aus dem LDAP-Verzeichnis abgerufen werden. Die einzelnen Clients müssen die Kataloginformationen nun nicht mehr lokal auf den verschiedenen Maschinen speichern. Client-Anwendungen durchsuchen das LDAP-Verzeichnis nach den erforderlichen Informationen für die Herstellung der Verbindung zur Datenbank.

LDAP ist keine Voraussetzung für die Verwendung von DB2. Wenn Sie jedoch in einer LDAP-Umgebung arbeiten, lesen Sie die Informationen in „Anhang J. LDAP-Verzeichnisservices (Lightweight Directory Access Protocol)“ auf Seite 469.

Erstellen von Knotengruppen

Eine Knotengruppe wird mit der Anweisung CREATE NODEGROUP erstellt. Diese Anweisung gibt eine Gruppe von Knoten an, in der sich die Behälter eines Tabellenbereichs und die Tabellendaten befinden sollen. Außerdem führt diese Anweisung folgende Operationen aus:

- Sie erstellt eine Partitionierungszuordnung für die Knotengruppe. Detaillierte Informationen zur Partitionierungszuordnung finden Sie im Handbuch *Systemverwaltung: Konzept*.
- Sie generiert eine ID für die Partitionierungszuordnung.
- Sie fügt Datensätze in die folgenden Katalogtabellen ein:
 - SYSCAT.NODEGROUPS
 - SYSCAT.PARTITIONMAPS
 - SYSCAT.NODEGROUPDEF

Gehen Sie wie folgt vor, um mit der Steuerzentrale eine Knotengruppe zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Knotengruppen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Ordner **Knotengruppen** und wählen Sie dann im Kontextmenü **Erstellen** aus.
3. Geben Sie im Fenster 'Knotengruppe erstellen' die erforderlichen Informationen ein, verwenden Sie die Pfeile, um die gewünschten Knoten vom Feld **Verfügbare Knoten** in das Feld **Ausgewählte Knoten** zu versetzen und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Knotengruppe zu erstellen:

```
CREATE NODEGROUP <name> ON NODES (<wert>,<wert>)
```

Nehmen Sie zum Beispiel an, dass Sie einige Tabellen in eine Untermenge der Datenbankpartitionen in Ihrer Datenbank laden wollen. Sie würden den folgenden Befehl verwenden, um eine Knotengruppe mit zwei Knoten (1 und 2) in einer Datenbank zu erstellen, die aus mindestens drei Knoten (0 bis 2) besteht:

```
CREATE NODEGROUP mixng12 ON NODES (1,2)
```

Weitere Informationen zum Erstellen von Knotengruppen finden Sie im Handbuch *SQL Reference*.

Mit dem Befehl CREATE DATABASE oder der API `sqlcrea()` werden die Standardknotengruppen des Systems IBMDEFAULTGROUP, IBMCATGROUP und IBMTEMPGROUP erstellt. (Weitere Informationen zu Knotengruppen finden Sie im Handbuch *Systemverwaltung: Konzept*.)

Definition des Wiederherstellungsprotokolls der Datenbank

Ein *Datenbankwiederherstellungsprotokoll* zeichnet Datensätze über alle an einer Datenbank vorgenommenen Änderungen auf, einschließlich das Hinzufügen neuer Tabellen oder Aktualisieren vorhandener Tabellen. Dieses Protokoll besteht aus einer Reihe von *Protokollspeicherbereichen*, die jeweils in einer separaten, so genannten *Protokolldatei* enthalten sind.

Das Datenbankwiederherstellungsprotokoll kann verwendet werden, um sicherzustellen, dass eine Datenbank nach einem Fehler (z. B. ein Netzstromausfall im System oder ein Anwendungsfehler) nicht in einem inkonsistenten Zustand bleibt. Im Fall eines Fehlers werden bereits durchgeführte Änderungen, die noch nicht festgeschrieben wurden, rückgängig gemacht, und alle festgeschriebenen Transaktionen, die eventuell noch nicht physisch auf die Platte geschrieben wurden, werden nachvollzogen. Durch diese Maßnahmen wird die Integrität der Datenbank gewährleistet.

Weitere Informationen finden Sie in *Versetzen von Daten Dienstprogramme und Referenz*.

Binden von Dienstprogrammen an die Datenbank

Bei der Erstellung einer Datenbank versucht der Datenbankmanager, die in der Datei `db2ubind.lst` aufgeführten Dienstprogramme an die Datenbank zu binden. Diese Datei befindet sich im Unterverzeichnis `bind` im entsprechenden Verzeichnis `sqlllib`. Das Dienstprogramm zum Binden (BIND) erstellt ein *Paket*, d. h. ein Objekt, das alle Informationen enthält, die zur Verarbeitung bestimmter SQL-Anweisungen aus einer einzelnen Quellendatei benötigt werden.

Anmerkung: Wenn diese Dienstprogramme von einem Client ausgeführt werden sollen, müssen sie explizit gebunden werden. Weitere Informationen finden Sie im Handbuch *Einstieg* für Ihre Plattform.

Wenn aus einem Grund Dienstprogramme an eine Datenbank gebunden bzw. erneut gebunden werden müssen, führen Sie dazu die folgenden Befehle über den Befehlszeilenprozessor aus:

```
connect to sample
bind @db2ubind.lst
```

Anmerkung: Sie müssen sich in dem Verzeichnis befinden, in dem diese Dateien gespeichert sind, um die Pakete in der Datenbank `sample` zu erstellen. Die Bindedateien befinden sich im Unterverzeichnis `BND` des Verzeichnisses `SQLLIB`. In diesem Beispiel ist `sample` der Name der Datenbank.

Katalogisieren einer Datenbank

Wenn Sie eine neue Datenbank erstellen, wird sie automatisch in der Datei für das Systemdatenbankverzeichnis katalogisiert. Sie können außerdem den Befehl `CATALOG DATABASE` verwenden, um eine Datenbank explizit im Systemdatenbankverzeichnis zu katalogisieren. Mit dem Befehl `CATALOG DATABASE` kann eine Datenbank mit einem anderen Aliasnamen katalogisiert werden, oder es kann ein Datenbankeintrag, der zuvor mit dem Befehl `UNCATALOG DATABASE` gelöscht wurde, katalogisiert werden.

Durch den folgenden Befehl für den Befehlszeilenprozessor wird die Datenbank `person1` als `humanres` katalogisiert:

```
catalog database person1 as humanres
with "Human Resources Database"
```

In diesem Beispiel enthält der Datenbankverzeichniseintrag den Aliasnamen `humanres` für die Datenbank, der sich vom Datenbanknamen (`person1`) unterscheidet.

Eine Datenbank kann auch in einem anderen Exemplar als dem Standardexemplar katalogisiert werden. Im folgenden Beispiel werden Verbindungen zur Datenbank `B` nun zum Exemplar `INSTANCE_C` hergestellt.

```
catalog database b as b at node instance_c
```

Anmerkung: Der Befehl `CATALOG DATABASE` wird auch auf Client-Knoten verwendet, um Datenbanken, die sich auf Datenbankserver-Maschinen befinden, zu katalogisieren. Weitere Informationen finden Sie im Handbuch *Einstieg* für Ihre Plattform.

Informationen zum DCE-Zellenverzeichnis (Distributed Computing Environment) finden Sie in „DCE Verzeichnisservices“ auf Seite 122 und in „Anhang B. Verwenden der DCE-Verzeichnisservices“ auf Seite 377.

Anmerkung: Zur Erhöhung der Leistung können Verzeichnisdateien, einschließlich des Datenbankverzeichnisses, im Cache (Hauptspeicherbereich) zwischengespeichert werden. (Informationen zum Aktivieren des Verzeichnis-Cachings finden Sie im Abschnitt über die „Unterstützung des Verzeichnis-Caches“ im Handbuch *Systemverwaltung: Optimierung*.) Wenn der Verzeichnis-Cache aktiviert ist, wird eine Änderung an einem Verzeichnis (z. B. durch den Befehl `CATALOG DATABASE` oder `UNCATALOG DATABASE`) durch eine andere Anwendung erst wirksam, wenn Ihre Anwendung erneut gestartet wird. Um den Verzeichnis-Cache, der von einer Sitzung des Befehlszeilenprozessors verwendet wird, zu aktualisieren, führen Sie den Befehl `db2 terminate` aus.

Neben dem Cache auf Anwendungsebene wird auch ein Cache auf Datenbankmanagerebene für interne Suchfunktionen des Datenbankmanagers verwendet. Um diesen „gemeinsamen“ Cache zu aktualisieren, geben Sie die Befehle `db2stop` und `db2start` ein.

Weitere Informationen zum Verzeichnis-Caching finden Sie im Abschnitt zur „Unterstützung des Verzeichnis-Caches“ im Handbuch *Systemverwaltung: Optimierung*.

Erstellen eines Tabellenbereichs

Durch das Erstellen eines Tabellenbereichs innerhalb einer Datenbank werden dem Tabellenbereich Behälter zugeordnet und die zugehörigen Definitionen und Attribute im Datenbanksystemkatalog gespeichert. Anschließend können Sie Tabellen in diesem Tabellenbereich erstellen.

Informationen zum Entwurf von Tabellenbereichen finden Sie im Handbuch *Systemverwaltung: Konzept*.

Eine detaillierte Beschreibung der Syntax der Anweisung `CREATE TABLESPACE` finden Sie im Handbuch *SQL Reference*. Weitere Informationen zu SMS- und DMS-Tabellenbereichen finden Sie im Handbuch *Systemverwaltung: Konzept*.

Gehen Sie wie folgt vor, um mit der Steuerzentrale einen Tabellenbereich zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellenbereiche** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Ordner **Tabellenbereiche** und wählen Sie dann im Kontextmenü **Erstellen** —> **Tabellenbereich mit Assistent** aus.
3. Führen Sie die im Assistenten aufgeführten Schritte aus, um die Task abzuschließen.

Geben Sie in der Befehlszeile Folgendes ein, um einen SMS-Tabellenbereich zu erstellen:

```
CREATE TABLESPACE <NAME>  
  MANAGED BY SYSTEM  
  USING ('<pfad>')
```

Geben Sie in der Befehlszeile Folgendes ein, um einen DMS-Tabellenbereich zu erstellen:

```
CREATE TABLESPACE <NAME>
  MANAGED BY DATABASE
  USING (FILE'<pfad>' <größe>)
```

Mit der folgenden SQL-Anweisung wird ein SMS-Tabellenbereich unter OS/2 oder Windows NT mit drei Verzeichnissen auf drei separaten Laufwerken erstellt:

```
CREATE TABLESPACE RESOURCE
  MANAGED BY SYSTEM
  USING ('d:\acc_tbsp', 'e:\acc_tbsp', 'f:\acc_tbsp')
```

Mit der folgenden SQL-Anweisung wird ein DMS-Tabellenbereich unter OS/2 mit zwei Dateibehältern von je 5000 Seiten erstellt:

```
CREATE TABLESPACE RESOURCE
  MANAGED BY DATABASE
  USING (FILE'd:\db2data\acc_tbsp' 5000,
        FILE'e:\db2data\acc_tbsp' 5000)
```

In den beiden gezeigten Beispielen werden explizit Namen für die Behälter angegeben. Wenn Sie jedoch relative Behälternamen angeben, wird der Behälter in dem für die Datenbank angelegten Unterverzeichnis erstellt (weitere Informationen zu physischen Datenbankverzeichnissen finden Sie im Handbuch *Systemverwaltung: Konzept*).

Wenn ein Teil des angegebenen Pfadnamens nicht existiert, wird er vom Datenbankmanager erstellt. Wird vom Datenbankmanager ein Unterverzeichnis erstellt, kann es auch wieder vom Datenbankmanager gelöscht werden, wenn der Tabellenbereich gelöscht wird.

In den obigen Beispielen wird davon ausgegangen, dass die Tabellenbereiche nicht einer bestimmten Knotengruppe zugeordnet werden. Die Standardknotengruppe IBMDEFAULTGROUP wird verwendet, wenn der folgende Parameter in der Anweisung nicht angegeben wird:

```
IN knotengruppe
```

Mit der folgenden SQL-Anweisung werden ein DMS-Tabellenbereich auf einem auf UNIX basierendem System mit drei logischen Datenträgern von je 10.000 Seiten erstellt und die zugehörigen E/A-Merkmale angegeben:

```
CREATE TABLESPACE RESOURCE
  MANAGED BY DATABASE
  USING (DEVICE '/dev/rdb1v6' 10000,
        DEVICE '/dev/rdb1v7' 10000,
        DEVICE '/dev/rdb1v8' 10000)
  OVERHEAD 24.1
  TRANSFERRATE 0.9
```

Die in der SQL-Anweisung angegebenen UNIX-Einheiten müssen bereits vorhanden sein, und der Exemplareigner und die SYSADM-Gruppe müssen Schreibzugriff auf sie haben.

Im folgenden Beispiel wird ein DMS-Tabellenbereich in einer Knotengruppe namens ODDNODEGROUP in einer partitionierten UNIX-Datenbank erstellt. Die Knotengruppe ODDNODEGROUP muss zuvor mit der Anweisung CREATE NODEGROUP erstellt worden sein. Im vorliegenden Beispiel wird angenommen, dass die Knotengruppe ODDNODEGROUP aus den Datenbankpartitionen 1, 3 und 5 besteht. In allen Datenbankpartitionen ist die Einheit (Device) /dev/hdisk0 für 10.000 4-KB-Seiten zu verwenden. Außerdem wird für jede Datenbankpartition eine Einheit von 40.000 4-KB-Seiten deklariert.

```
CREATE TABLESPACE PLANS
MANAGED BY DATABASE
USING (DEVICE '/dev/HDISK0' 10000, DEVICE '/dev/n1hd01' 40000) ON NODE 1
      (DEVICE '/dev/HDISK0' 10000, DEVICE '/dev/n3hd03' 40000) ON NODE 3
      (DEVICE '/dev/HDISK0' 10000, DEVICE '/dev/n5hd05' 40000) ON NODE 5
```

UNIX-Einheiten werden in zwei Kategorien eingeteilt: zeichenorientierte serielle Einheiten und blockorientierte Einheiten. Bei allen Dateisystemeinheiten ist normalerweise für jede blockorientierte Einheit (*Cooked Device*) eine entsprechende serielle zeichenorientierte Einheit (*Raw Device*) vorhanden. Blockorientierte Einheiten werden häufig mit Namen wie „hd0“, „fd0“ oder ähnlich bezeichnet. Zeichenorientierte serielle Einheiten erhalten typischerweise Namen wie „rhd0“, „rfd0“, „rmt0“ o. ä. Zeichenorientierte serielle Einheiten haben einen schnelleren Zugriff als blockorientierte Einheiten. Im Befehl CREATE TABLESPACE sind die Namen der zeichenorientierten seriellen Einheiten zu verwenden, und nicht die Namen der blockorientierten Einheiten.

Die Parameter OVERHEAD und TRANSFERRATE helfen bei der Feststellung des besten Zugriffspfads, der zu verwenden ist, wenn die SQL-Anweisung kompiliert wird. Informationen zu den Parametern OVERHEAD und TRANSFERRATE enthält der Abschnitt zur „Optimierung der Anwendungsleistung“ im Handbuch *Systemverwaltung: Optimierung*.

DB2 kann die Leistung sequenzieller E/A-Operationen durch die Verwendung des sequenziellen Vorabsezugriffs (Sequential Prefetching) wesentlich verbessern, da durch sie parallele E/A-Operationen durchgeführt werden. Detaillierte Informationen zu dieser Funktion finden Sie im Abschnitt über den sequenziellen Vorabsezugriff im Handbuch *Systemverwaltung: Optimierung*.

Sie können auch einen Tabellenbereich erstellen, der den Standardwert von 4 KB für die Seitengröße übersteigt. Durch die folgende SQL-Anweisung wird ein SMS-Tabellenbereich auf einem UNIX-gestützten System mit einer Seitengröße von 8 KB erstellt.

```

CREATE TABLESPACE SMS8K
  PAGESIZE 8192
  MANAGED BY SYSTEM
  USING ('FSMS_8K_1')
  BUFFERPOOL BUFFPOOL8K

```

Beachten Sie, dass der zugeordnete Pufferpool ebenfalls die Seitengröße von 8 KB aufweisen muss.

Der erstellte Tabellenbereich kann nicht verwendet werden, bis der Pufferpool, auf den er verweist, aktiviert ist.

Die SQL-Anweisung ALTER TABLESPACE kann dazu verwendet werden, einen Behälter einem DMS-Tabellenbereich hinzuzufügen und die Einstellungen für die Parameter PREFETCHSIZE, OVERHEAD und TRANSFERRATE für einen Tabellenbereich zu ändern. Die Transaktion, die die Anweisung für den Tabellenbereich absetzt, sollte so bald wie möglich festgeschrieben werden, um Konkurrenzsituationen im Systemkatalog zu vermeiden.

Anmerkung: Für den Wert für PREFETCHSIZE sollte ein Vielfaches des Werts für EXTENTSIZE verwendet werden. Wenn beispielsweise für EXTENTSIZE der Wert 10 angegeben wird, sollte der Wert für PREFETCHSIZE 20 oder 30 sein. Weitere Informationen hierzu finden Sie im Abschnitt über den sequenziellen Vorablesezugriff im Handbuch *Systemverwaltung: Optimierung*.

Erstellen eines temporären Systemtabellenbereichs

Ein temporärer Systemtabellenbereich wird zum Speichern temporärer Systemtabellen verwendet. Beim Erstellen einer Datenbank wird einer der drei standardmäßig definierten Tabellenbereiche als temporärer Systemtabellenbereich konfiguriert und erhält den Namen „TEMPSPACE1“.

Anmerkung: Eine Datenbank muss immer über mindestens einen temporären Systemtabellenbereich verfügen, da temporäre Systemtabellen nur in solch einem Tabellenbereich gespeichert werden können.

Mit der Anweisung CREATE TABLESPACE können Sie einen weiteren temporären Systemtabellenbereich erstellen. Beispiel:

```

CREATE SYSTEM TEMPORARY TABLESPACE tmp_tbsp
  MANAGED BY SYSTEM
  USING ('d:\tmp_tbsp','e:\tmp_tbsp')

```

Die einzige Knotengruppe, die beim Erstellen eines temporären Systemtabellenbereichs angegeben werden kann, ist IBMTEMPGROUP.

Erstellen eines temporären Benutzertabellenbereichs

Ein temporärer Benutzertabellenbereich wird zum Speichern deklarierter, temporärer Tabellen verwendet. Mit der Anweisung CREATE TABLESPACE können Sie einen temporären Benutzertabellenbereich erstellen:

```
CREATE USER TEMPORARY TABLESPACE usr_tbsp
MANAGED BY DATABASE
USING (FILE 'd:\db2data\user_tbsp' 5000,
FILE 'e:\db2data\user_tbsp' 5000)
```

Wie bei regulären Tabellenbereichen können temporäre Benutzertabellenbereiche in allen Knotengruppen mit Ausnahme von IBMTEMPGROUP erstellt werden. Standardmäßig wird beim Erstellen eines temporären Benutzertabellenbereichs die Knotengruppe IBMDEFAULTGROUP verwendet.

Mit der Anweisung DECLARE GLOBAL TEMPORARY TABLE werden deklarierte, temporäre Tabellen für den Einsatz in einem temporären Benutzertabellenbereich definiert.

Erstellen von Tabellenbereichen in Knotengruppen

Wenn ein Tabellenbereich in einer Knotengruppe mit mehreren Datenbankpartitionen erstellt wird, werden alle Tabellen innerhalb des Tabellenbereichs auf alle Datenbankpartitionen in der Knotengruppe verteilt oder partitioniert. Der Tabellenbereich wird in einer Knotengruppe erstellt. Wenn der Tabellenbereich in einer Knotengruppe erstellt wurde, muss er dort verbleiben und kann nicht in eine andere Knotengruppe versetzt werden. Für die Zuordnung eines Tabellenbereichs zu einer Knotengruppe wird die Anweisung CREATE TABLESPACE verwendet.

Direkter Plattenzugriff (Raw I/O)

DB2 Universal Database unterstützt den direkten Plattenzugriff (Raw I/O). Dadurch haben Sie die Möglichkeit, eine Einheit (Gerät) mit direktem Plattenzugriff (Raw) an ein DB2 Universal Database-System anzuschließen. (Als einzige Ausnahmen sind hierbei die Betriebssysteme Windows 95 und Windows 98 aufzuführen.) Die folgende Liste zeigt die physischen und logischen Verfahren, eine solche Einheit zu identifizieren:

- Unter Windows müssen Sie die folgende Syntax verwenden, um ein physisches Festplattenlaufwerk anzugeben:

```
\\.\PhysicalDriveN
```

Dabei steht N für eines der physischen Laufwerke im System. Im vorliegenden Fall kann N durch die Werte 0, 1, 2 oder eine beliebige andere positive ganze Zahl ersetzt werden:

```
\\.\PhysicalDisk5
```

- Unter Windows müssen Sie zur Angabe einer logischen unformatierten Partition (raw-Partition) die folgende Syntax verwenden:

```
\\.\N:
```

Dabei ist N: ein Buchstabe eines logischen Laufwerks im System. Die Angabe N: kann z. B. durch E: oder eine beliebige andere Angabe für den Laufwerkbuchstaben ersetzt werden.

- **Anmerkung:** Damit Protokolle auf einer Einheit geschrieben werden können, muss Windows NT Version 4.0 mit Service Pack 3 installiert sein.
- Auf Plattformen unter UNIX verwenden Sie den Namen der zeichenorientierten seriellen Einheit, z. B. /dev/rhd0.

Verwenden von unformatierter Ein-/Ausgabe unter Linux

Linux verfügt über einen Pool unformatierter Einheitenknoten, die an eine Blockeinheit gebunden werden müssen, bevor für diese Einheitenknoten eine unformatierte Ein-/Ausgabe erfolgen kann. Es gibt einen Controller für unformatierte Einheiten, der als zentrales Repository für die Bindungsinformationen von unformatierten Einheiten an die Blockeinheiten fungiert. Die Bindung erfolgt mit Hilfe eines Dienstprogramms mit dem Namen raw, das in der Regel vom Linux-Distributor zur Verfügung gestellt wird.

Bevor Sie die unformatierte Ein-/Ausgabe für Linux definieren, müssen folgende Voraussetzungen erfüllt sein:

- Mindestens eine freie IDE- oder SCSI-Plattenpartition
- Linux-Kernel ab Version 2.4.0 (Einige Linux-Varianten bieten unformatierte Ein-/Ausgabe auf Kernels der Version 2.2 an.)
- Einen Controller für unformatierte Einheiten mit dem Namen /dev/rawctl oder /dev/raw. Falls dies nicht der Fall ist, erstellen Sie eine symbolische Verbindung:

```
# ln -s /dev/your_raw_dev_ctl /dev/rawctl
```

- Das Dienstprogramm raw, das in der Regel vom Linux-Distributor zur Verfügung gestellt wird.
- DB2 ab Version 7.1 FixPak 3

Anmerkung: Bei den Versionen, die momentan unformatierte Ein-/Ausgabe unterstützen, ist die Benennung der Knoten der unformatierten Einheit unterschiedlich:

Version	Knoten	Controller
RedHat 6.2	/dev/raw/raw1 to 255	/dev/rawctl
SuSE 7.0	/dev/raw1 to 63	/dev/raw

DB2 unterstützt beide der genannten Einheiten-Controller sowie die meisten anderen Namen für Knoten unformatierter Einheiten. Unformatierte Einheiten werden von DB2 unter Linux/390 nicht unterstützt.

Gehen Sie wie folgt vor, um unformatierte Ein-/Ausgabe unter Linux zu konfigurieren:

In diesem Beispiel wird als unformatierte Partition /dev/sda5 verwendet. Diese Partition sollte keine wertvollen Daten enthalten.

Schritt 1. Berechnen Sie die Anzahl der Seiten mit 4 096-Byte in dieser Partition, und runden Sie gegebenenfalls ab. Beispiel:

```
# fdisk /dev/sda
Command (m for help): p

Disk /dev/sda: 255 heads, 63 sectors, 1106 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot   Start    End    Blocks  Id System
/dev/sda1             1     523   4200997  83 Linux
/dev/sda2             524   1106   4682947+  5 Extended
/dev/sda5             524   1106   4682947  83 Linux

Command (m for help): q
#
```

Die Anzahl der Seiten in /dev/sda5 ist

```
num_pages = floor( ((1106-524+1)*16065*512)/4096 )
num_pages = 11170736
```

Schritt 2. Binden Sie eine unbenutzte, unformatierte Einheit an diese Partition. Dieser Vorgang muss nach jedem Neustart der Maschine erneut erfolgen und erfordert Root-Zugriff. Verwenden Sie `raw -a`, um anzuzeigen, welche Knoten unformatierter Einheiten bereits verwendet werden:

```
# raw /dev/raw/raw1 /dev/sda5
/dev/raw/raw1: bound to major 8, minor 5
```

Schritt 3. Legen Sie globale Leseberechtigungen für den Controller für unformatierte Einheiten und die Plattenpartition fest. Legen Sie globale Lese- und Schreibberechtigungen für die unformatierte Einheit fest:

```
# chmod a+r /dev/rawctl
# chmod a+r /dev/sdb1
# chmod a+r /dev/raw/raw1
```

Schritt 4. Erstellen Sie den Tabellenbereich in DB2. Geben Sie dabei die unformatierte Einheit und nicht die Plattenpartition an. Beispiel:

```
CREATE TABLESPACE dms1
MANAGED BY DATABASE
USING (DEVICE '/dev/raw/raw1' 11170736)
```

Tabellenbereiche auf unformatierten Einheiten werden ebenso für alle anderen Seitengrößen unterstützt, die von DB2 unterstützt werden.

Erstellen eines Schemas

Obwohl die Daten bereits in Tabellen organisiert sind, kann es durchaus sinnvoll sein, Tabellen (und andere zugehörige Objekte) zusammen zu gruppieren. Dies kann durch Definieren eines Schemas mit Hilfe der Anweisung `CREATE SCHEMA` geschehen. Informationen zu dem Schema werden in den Systemkatalogtabellen der Datenbank gespeichert, mit der sie verbunden sind. Wenn andere Objekte erstellt werden, können sie in dieses Schema gestellt werden.

Eine detaillierte Beschreibung der Syntax der Anweisung `CREATE SCHEMA` finden Sie im Handbuch *SQL Reference*. Das neue Schema darf nicht bereits in den Systemkatalogen vorhanden sein und darf nicht mit der Zeichenfolge „SYS“ beginnen.

Wenn ein Benutzer die Berechtigung `SYSADM` oder `DBADM` hat, kann er ein Schema mit einem beliebigen gültigen Namen erstellen. Bei der Erstellung einer Datenbank wird die Berechtigung `IMPLICIT_SCHEMA` der Gruppe `PUBLIC` (d. h. allen Benutzern) erteilt.

Die Person, die Objekte definiert, die als Teil der Anweisung `CREATE SCHEMA` erstellt wurden, ist der Schemeneigner. Dieser Eigner kann Zugriffsrechte auf Schemata anderen Benutzern erteilen (`GRANT`) und entziehen (`REVOKE`).

Zur Ausführung dieser Anweisung benötigt der Benutzer die Berechtigung `DBADM`.

Schemata können auch implizit erstellt werden, wenn ein Benutzer die Berechtigung `IMPLICIT_SCHEMA` hat. Mit dieser Berechtigung erstellen Benutzer implizit ein Schema, wenn sie ein Objekt mit einem Schemennamen erstellen, der nicht bereits existiert.

Wenn Benutzer nicht über die Berechtigung `IMPLICIT_SCHEMA` verfügen, können sie nur das Schema erstellen, das den gleichen Namen hat wie ihre eigene Berechtigungs-ID.

Der Direktzugriff auf Objekte in einem Schema ist nicht zulässig, da das Schema dazu verwendet wird, für Eindeutigkeit in der Datenbank zu sorgen. Dies leuchtet ein, wenn man bedenkt, dass zwei Benutzer jeweils eine Tabelle (oder andere Objekte) mit demselben Namen erstellen könnten. Ohne ein Schema, das für Eindeutigkeit sorgt, käme es zu Problemen, wenn ein dritter Benutzer versuchen würde, die Tabelle für Abfragen zu verwenden. Es wäre ohne weitere Angaben nicht möglich festzustellen, welche Tabelle verwendet werden soll.

Um einem anderen Benutzer Zugriff auf die Tabelle ohne Eingabe eines Schemennamens als Teil der Qualifikation des Tabellennamens zu gewähren, muss für diesen Benutzer eine Sicht eingerichtet werden. Die Definition einer Sicht definiert den vollständigen Tabellennamen einschließlich des Schemas des Benutzers. Der Benutzer müsste für Abfragen nur den Namen der Sicht angeben. Die Sicht wäre durch das Schema des Benutzers als Teil der Sichtdefinition vollständig qualifiziert.

Gehen Sie wie folgt vor, um mit der Steuerzentrale ein Schema zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Schema** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Ordner **Schema** und wählen Sie dann im Kontextmenü **Erstellen** aus.
3. Geben Sie die Informationen für das neue Schema ein und klicken Sie auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um ein Schema zu erstellen:

```
CREATE SCHEMA <name> AUTHORIZATION <name>
```

Das folgende Beispiel zeigt eine Anweisung `CREATE SCHEMA`, mit der ein Schema für einen einzelnen Benutzer mit der Berechtigungs-ID „joe“ erstellt wird:

```
CREATE SCHEMA joeschema AUTHORIZATION joe
```

Einstellen eines Schemas

Es kann sinnvoll sein, ein Standardschema einzurichten, das für nicht qualifizierte Objektverweise in dynamischen SQL-Anweisungen verwendet wird, die innerhalb einer bestimmten DB2-Verbindung abgesetzt werden. Dies kann durch Einstellen des Sonderregisters `CURRENT SCHEMA` auf das Schema geschehen, das Sie als Standardschema verwenden wollen. Dieses Sonderregister kann von jedem beliebigen Benutzer eingestellt werden. Es ist keine Berechtigung erforderlich.

Eine detaillierte Beschreibung der Syntax der Anweisung `SET SCHEMA` finden Sie im Handbuch *SQL Reference*.

Das folgende Beispiel zeigt, wie das Sonderregister `CURRENT SCHEMA` eingestellt wird:

```
SET CURRENT SCHEMA = 'SCHEMA01'
```

Diese Anweisung kann innerhalb eines Anwendungsprogramms oder interaktiv abgesetzt werden. Wenn das Sonderregister CURRENT SCHEMA eingestellt ist, wird sein Wert als Qualifikationsmerkmal (Schema) für alle Objektverweise ohne Qualifikationsmerkmal in dynamischen SQL-Anweisungen (außer der Anweisung CREATE SCHEMA) verwendet, in denen ein Verweis ohne Qualifikationsmerkmal auf ein Datenbankobjekt vorhanden ist.

Der Anfangswert des Sonderregisters CURRENT SCHEMA entspricht der Berechtigungs-ID des Benutzers der aktuellen Sitzung.

Erstellen und Auffüllen einer Tabelle

Wenn festgelegt ist, wie die Daten in Tabellen organisiert werden sollen, besteht der nächste Schritt darin, diese Tabellen mit der Anweisung CREATE TABLE zu erstellen. Die Tabellenbeschreibungen werden im Systemkatalog der Datenbank, mit der Sie verbunden sind, gespeichert.

Eine detaillierte Beschreibung der Syntax der Anweisung CREATE TABLE finden Sie im Handbuch *SQL Reference*. Informationen zum Erstellen einer Übersichtstabelle finden Sie in „Erstellen einer Übersichtstabelle“ auf Seite 169. Informationen zur Benennung von Tabellen, Spalten und anderer Datenbankobjekte finden Sie in „Anhang A. Namenskonventionen“ auf Seite 369.

Durch die Anweisung CREATE TABLE werden der Tabelle ein Name, der ein qualifizierter oder nicht qualifizierter Bezeichner sein kann, und eine Definition für jede der enthaltenen Spalten gegeben. Jede Tabelle kann in einem separaten Tabellenbereich gespeichert werden, so dass ein Tabellenbereich nur eine Tabelle enthält. Wenn eine Tabelle häufig gelöscht und wieder erstellt wird, ist es effizienter, sie in einem separaten Tabellenbereich zu speichern und dann den Tabellenbereich anstelle der Tabelle zu löschen. Es können auch mehrere Tabellen in einem einzigen Tabellenbereich gespeichert werden. In einer partitionierten Datenbankumgebung definiert der ausgewählte Tabellenbereich außerdem die Knotengruppe und die Datenbankpartitionen, in denen die Tabellendaten gespeichert werden.

Zu Beginn enthält die Tabelle keine Daten. Um der Tabelle Zeilen mit Daten hinzuzufügen, haben Sie folgende Möglichkeiten:

- Die Anweisung INSERT, die im Handbuch *SQL Reference* beschrieben ist
- Den Befehl LOAD oder den Befehl IMPORT, die im Handbuch *Command Reference* beschrieben sind
- Das Dienstprogramm Autoloader (Programm für automatisches Laden), wenn Sie in einer Umgebung mit partitionierter Datenbank arbeiten. Siehe *Versetzen von Daten Dienstprogramme und Referenz*.

Einzelheiten zum Verschieben von Daten in und aus Tabellen finden Sie in *Versetzen von Daten Dienstprogramme und Referenz*.

Daten können zu einer Tabelle hinzugefügt werden, ohne die Änderungen zu protokollieren. Mit der Klausel `NOT LOGGED INITIALLY` der Anweisung `CREATE TABLE` kann verhindert werden, dass die an der Tabelle vorgenommenen Änderungen protokolliert werden. Alle Änderungen an der Tabelle durch die Operationen `INSERT`, `DELETE`, `UPDATE`, `CREATE INDEX`, `DROP INDEX` und `ALTER TABLE` in derselben Arbeitseinheit, in der die Tabelle erstellt wird, werden nicht protokolliert. Das Protokollieren beginnt in den nachfolgenden Arbeitseinheiten.

Eine Tabelle besteht aus einer oder mehreren Spaltendefinitionen. Es können maximal 500 Spalten für eine Tabelle definiert werden. Die Spalten stellen die Attribute einer Entität dar. Die Werte jeder Spalte gehören jeweils demselben Datentyp an. Weitere Informationen finden Sie im Handbuch *SQL Reference*.

Anmerkung: Der Maximalwert von 500 Spalten gilt, wenn eine Seitengröße von 4 KB verwendet wird. Bei einer Seitengröße von 8 KB, 16 KB oder 32 KB beträgt der Maximalwert 1012 Spalten.

Zu einer Spaltendefinition gehören ein *Spaltenname*, ein *Datentyp* und jedes erforderliche *Nullattribut* bzw. ein (wahlfrei vom Benutzer gewählter) Standardwert.

Der Spaltenname beschreibt die Informationen, die in der Spalte enthalten sind, und sollte leicht verständlich sein. Er muss innerhalb der Tabelle eindeutig sein. Jedoch kann derselbe Name auch in anderen Tabellen verwendet werden. Informationen zu Benennungsregeln finden Sie in „Anhang A. Namenskonventionen“ auf Seite 369.

Der Datentyp einer Spalte definiert die Länge der enthaltenen Werte und die Art von Daten, die für die Spalte gültig sind. Der Datenbankmanager unterscheidet die Datentypen für Zeichenfolgen, numerische Daten, Datum, Uhrzeit und LOB (großes Objekt). Datentypen für Grafikzeichenfolgen sind nur für Datenbankumgebungen gültig, in denen Mehrbytezeichensätze verwendet werden. Darüber hinaus können für Spalten benutzerdefinierte Datentypen (UDT - User-defined Distinct Type) definiert werden. Eine Beschreibung dieser Datentypen finden Sie in „Erstellen eines benutzerdefinierten Datentyps (UDT)“ auf Seite 163.

Mit dem Attribut `DEFAULT` wird definiert, welcher Wert zu verwenden ist, wenn kein Wert für die Spalte angegeben wird. Der Standardwert (`DEFAULT`) kann angegeben werden, oder es kann ein vom System definierter Standardwert verwendet werden. Standardwerte (`DEFAULT`-Attribute) können für Spalten mit und ohne Angabe des Nullattributs definiert werden.

Das Nullattribut gibt an, ob eine Spalte Nullwerte enthalten kann oder nicht.

Gehen Sie wie folgt vor, um mit der Steuerzentrale eine Tabelle zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Ordner **Tabellen** und wählen Sie dann im Kontextmenü **Erstellen** → **Tabellen mit Assistent** aus.
3. Führen Sie die im Assistenten aufgeführten Schritte aus, um die Tasks abzuschließen.

Geben Sie in der Befehlszeile Folgendes ein, um eine Tabelle zu erstellen:

```
CREATE TABLE <NAME>
  (<spaltenname> <datentyp> <>nullattribut>)
  IN <TABLE_SPACE_NAME>
```

Das folgende Beispiel zeigt eine Anweisung CREATE TABLE, mit der die Tabelle EMPLOYEE im Tabellenbereich RESOURCE erstellt wird. Diese Tabelle ist in der Beispieldatenbank definiert:

```
CREATE TABLE EMPLOYEE
  (EMPNO      CHAR(6)      NOT NULL PRIMARY KEY,
   FIRSTNAME  VARCHAR(12)  NOT NULL,
   MIDINIT    CHAR(1)      NOT NULL WITH DEFAULT,
   LASTNAME   VARCHAR(15)  NOT NULL,
   WORKDEPT   CHAR(3),
   PHONENO    CHAR(4),
   PHOTO      BLOB(10M)    NOT NULL)
  IN RESOURCE
```

Beim Erstellen einer Tabelle können Sie angeben, dass die Spalten der Tabelle auf den Attributen eines strukturierten Typs basieren sollen. Eine Solche Tabelle wird als „typisierte Tabelle“ bezeichnet.

Eine typisierte Tabelle kann so definiert werden, dass sie einige ihrer Spalten aus einer anderen typisierten Tabelle übernimmt. Eine solche Tabelle wird als „untergeordnete Tabelle“ bezeichnet und die Tabelle, aus der sie Spalten übernimmt, als „übergeordnete Tabelle“. Eine typisierte Tabelle mit allen untergeordneten Tabellen wird als „Tabellenhierarchie“ bezeichnet. Die oberste Tabelle in der Tabellenhierarchie (zu der es keine übergeordnete Tabelle gibt) wird als „Stammtabelle“ der Hierarchie bezeichnet.

Die folgenden Abschnitte gehen von diesem Beispiel aus, um einige andere Optionen zu behandeln, die bedacht werden sollten:

- „Überlegungen zu LOB-Spalten“ auf Seite 138
- „Definieren einer eindeutigen Integritätsbedingung“ auf Seite 140
- „Definieren einer generierten Spalte in einer neuen Tabelle“ auf Seite 146

- „Erstellen einer benutzerdefinierten temporären Tabelle“ auf Seite 147
- „Definieren einer Identitätsspalte in einer neuen Tabelle“ auf Seite 148
- „Erstellen einer Sequenz“ auf Seite 150
- „Vergleich von Identitätsspalten (IDENTITY) und Sequenzen“ auf Seite 152
- „Erstellen einer typisierten Tabelle“ auf Seite 153
- „Auffüllen einer typisierten Tabelle“ auf Seite 153
- „Hierarchietabelle“ auf Seite 153
- „Erstellen einer Tabelle in mehreren Tabellenbereichen“ auf Seite 153
- „Erstellen einer Tabelle in einer partitionierten Datenbank“ auf Seite 154

Sie können auch eine Tabelle erstellen, die auf Grundlage eines Abfrageergebnisses definiert ist. Diese Art von Tabelle wird eine *Übersichtstabelle* genannt. Weitere Informationen finden Sie in „Erstellen einer Übersichtstabelle“ auf Seite 169.

Überlegungen zu LOB-Spalten

Bevor Sie eine Tabelle erstellen, die Spalten mit großen Objekten (LOB-Spalten) enthält, müssen Sie folgende Entscheidungen treffen:

1. Sollen Änderungen an LOB-Spalten protokolliert werden?

Wenn Sie diese Änderungen nicht protokollieren wollen, müssen Sie die Protokollierung inaktivieren, indem Sie die Klausel NOT LOGGED bei der Erstellung der Tabelle angeben:

```
CREATE TABLE EMPLOYEE
  (EMPNO    CHAR(6)      NOT NULL PRIMARY KEY,
   FIRSTME  VARCHAR(12)  NOT NULL,
   MIDINIT  CHAR(1)     NOT NULL WITH DEFAULT,
   LASTNAME VARCHAR(15)  NOT NULL,
   WORKDEPT CHAR(3),
   PHONENO  CHAR(4),
   PHOTO    BLOB(10M)  NOT NULL NOT LOGGED)
IN RESOURCE
```

Wenn die LOB-Spalte Daten mit einer Größe von über 1 GB enthält, muss die Protokollierung inaktiviert werden. (Als Faustregel gilt, dass LOB-Spalten mit einer Größe über 10 MB Größe meist nicht protokolliert werden sollten.) Wie bei anderen Optionen auch, die in einer Spaltendefinition angegeben werden, kann die Einstellung der Protokolloption nur geändert werden, indem die Tabelle neu erstellt wird.

Auch wenn Sie angeben, dass die Änderungen nicht protokolliert werden sollen, werden LOB-Spalten auf bestimmte Art *gespiegelt*, damit Änderungen rückgängig gemacht werden können, wenn ein Systemfehler oder eine Anforderung durch eine Anwendung dies erforderlich macht. Dieses Spiegeln ist eine Wiederherstellungsmethode, bei der der aktuelle Inhalt der

Speicherseiten nicht überschrieben wird. Das heißt, alte, nicht geänderte Seiten werden als „Spiegelkopien“ zurückbehalten. Diese Kopien werden gelöscht, wenn sie zur Unterstützung der Zurücksetzung einer Transaktion nicht mehr benötigt werden.

Anmerkung: Bei der Wiederherstellung einer Datenbank mit den Befehlen RESTORE und ROLLFORWARD werden LOB-Daten, die nicht protokolliert („NOT LOGGED“) wurden und die seit der letzten Sicherung geschrieben wurden, *durch binäre Nullen ersetzt*.

2. Soll der für die LOB-Spalte erforderliche Speicherbereich minimiert werden?

Sie können die LOB-Spalte so klein wie möglich halten, wenn Sie die Klausel COMPACT in der Anweisung CREATE TABLE verwenden. Beispiel:

```
CREATE TABLE EMPLOYEE
  (EMPNO CHAR(6) NOT NULL PRIMARY KEY,
  FIRSTNAME VARCHAR(12) NOT NULL,
  MIDINIT CHAR(1) NOT NULL WITH DEFAULT,
  LASTNAME VARCHAR(15) NOT NULL,
  WORKDEPT CHAR(3),
  PHONENO CHAR(4),
  PHOTO BLOB(10M) NOT NULL NOT LOGGED COMPACT)
IN RESOURCE
```

Wenn Daten an eine Tabelle mit einer kompakten LOB-Spalte (COMPACT) angehängt werden, kommt es zu *Leistungseinbußen*, besonders wenn dabei die Größe der LOB-Werte erhöht wird. (Dies wird durch erforderliche Speicheranpassungen verursacht.)

Auf Plattformen wie OS/2, die die Zuordnung von Dateien mit freien Bereichen nicht unterstützen, und wo große Objekte (LOBs) in SMS-Tabellenbereichen untergebracht werden, sollten Sie die Klausel COMPACT verwenden. Die Zuordnung von Dateien mit freien Bereichen hängt mit der Art und Weise zusammen, wie ein Betriebssystem den physischen Plattenspeicher verwendet. Ein Betriebssystem, das die Zuordnung von Dateien mit freien Bereichen unterstützt, belegt im Vergleich zu Betriebssystemen, die die Zuordnung solcher Dateien nicht unterstützen, weniger physischen Speicherbereich zur Speicherung großer Objekte (LOBs). Die Option COMPACT ermöglicht noch größere „Einsparungen“ im Hinblick auf den physischen Speicher, unabhängig davon, ob die Zuordnung von Dateien mit freien Bereichen unterstützt wird oder nicht. Da Sie durch die Verwendung der Klausel COMPACT gewisse „Einsparungen“ an physischem Plattenspeicher erzielen können, sollten Sie in Betracht ziehen, die Klausel COMPACT zu verwenden, wenn Ihr Betriebssystem die Zuordnung von Dateien mit freien Bereichen nicht unterstützt.

Anmerkung: Die DB2-Systemkataloge verwenden LOB-Spalten und belegen eventuell mehr Speicher als in früheren Versionen.

3. Wünschen Sie eine Leistungssteigerung für LOB-Spalten, einschließlich jener LOB-Spalten in den DB2-Systemkatalogen?

In den Katalogtabellen gibt es Spalten mit großen Objekten (LOB-Spalten). LOB-Daten werden nicht mit anderen Daten im Pufferpool behalten, sondern jedes Mal, wenn sie benötigt werden, von der Platte gelesen. Das Lesen von der Platte verlangsamt die Leistung von DB2, wenn LOB-Spalten der Kataloge beteiligt sind. Da ein Dateisystem in der Regel über eigene Mechanismen zum Zwischenspeichern (Caching) von Daten verfügt, kann die Verwendung eines SMS-Tabellenbereichs oder eines DMS-Tabellenbereichs, der auf Dateibehältern basiert, die E/A-Operationen möglicherweise umgehen, wenn auf die LOB-Daten zuvor bereits zugegriffen wurde.

Definieren von Integritätsbedingungen

In diesem Abschnitt wird die Definition von Integritätsbedingungen behandelt:

- „Definieren einer eindeutigen Integritätsbedingung“
- „Definieren referenzieller Integritätsbedingungen“ auf Seite 141
- „Definieren von Prüfungen auf Integritätsbedingungen in Tabellen“ auf Seite 145.

Weitere Informationen zu den Integritätsbedingungen enthält der Abschnitt zur Planung der zwingenden Definition von Integritätsbedingung in den Handbüchern *Systemverwaltung: Konzept* und *SQL Reference*.

Definieren einer eindeutigen Integritätsbedingung

Eindeutige Integritätsbedingungen stellen sicher, dass jeder Wert im angegebenen Schlüssel eindeutig ist. Eine Tabelle kann mehrere Integritätsbedingungen für Eindeutigkeit haben, wobei maximal eine eindeutige Integritätsbedingung als Primärschlüssel definiert sein kann.

Eine eindeutige Integritätsbedingung wird mit der Klausel UNIQUE in der Anweisung CREATE TABLE bzw. ALTER TABLE definiert. Ein eindeutiger Schlüssel kann aus mehr als einer Spalte bestehen. In einer Tabelle ist mehr als eine eindeutige Integritätsbedingung zulässig. Eine eindeutige Integritätsbedingung kann jedoch nicht in einer untergeordneten Tabelle definiert werden.

Wenn die eindeutige Integritätsbedingung definiert ist, wird sie vom Datenbankmanager automatisch umgesetzt, wenn eine INSERT- oder UPDATE-Anweisung die Daten in der Tabelle ändert. Die eindeutige Integritätsbedingung wird mit Hilfe eines eindeutigen Index realisiert.

Wenn eine eindeutige Integritätsbedingung in einer Anweisung ALTER TABLE definiert wird und ein Index für dieselbe Gruppe von Spalten dieses eindeutigen Schlüssels existiert, wird dieser Index zum eindeutigen Index und wird von der Integritätsbedingung verwendet.

Sie können jede einzelne eindeutige Integritätsbedingung als *Primärschlüssel* verwenden. Der Primärschlüssel kann als übergeordneter Schlüssel in einer referenziellen Integritätsbedingung (zusammen mit anderen eindeutigen Integritätsbedingungen) verwendet werden. Es kann nur einen Primärschlüssel pro Tabelle geben. Ein Primärschlüssel wird mit der Klausel PRIMARY KEY in der Anweisung CREATE TABLE bzw. ALTER TABLE definiert. Der Primärschlüssel kann aus mehr als einer Spalte bestehen.

Für den Primärindex ist es zwingend erforderlich, dass der Primärschlüssel eindeutig ist. Wenn eine Tabelle mit einem Primärschlüssel erstellt wird, legt der Datenbankmanager einen Primärindex für diesen Schlüssel an.

Einige Hinweise zur Leistungsoptimierung für Indizes, die für eindeutige Integritätsbedingungen verwendet werden:

- Beim einleitenden Laden einer leeren Tabelle mit Indizes können mit LOAD bessere Leistungen erzielt werden als mit IMPORT. Dies ist von der Verwendung des Modus INSERT oder REPLACE für LOAD unabhängig.
- Beim Anhängen einer beträchtlichen Datenmenge an eine vorhandene Tabelle mit Indizes (unter Verwendung von IMPORT INSERT oder LOAD INSERT) ist LOAD etwas leistungsstärker als IMPORT.
- Wenn Sie den Befehl IMPORT für das einleitende Laden einer großen Datenmenge verwenden, erstellen Sie den eindeutigen Schlüssel nach dem Import bzw. Laden der Daten. Dadurch wird der Aufwand für die Verwaltung des Index während des Ladens der Tabelle vermieden. Darüber hinaus verwendet der Index auf diese Weise die geringste Menge an Speicher.
- Wenn Sie das Dienstprogramm zum Laden (LOAD) im Modus REPLACE verwenden, erstellen Sie den eindeutigen Schlüssel, bevor Sie die Daten laden. In diesem Fall ist die Erstellung des Index während des Ladens effizienter als die Verwendung der Anweisung CREATE INDEX nach der Ladeoperation.

Definieren referenzieller Integritätsbedingungen

Referenzielle Integrität wird durch Hinzufügen referenzieller Integritätsbedingungen zu Tabellen- und Spaltendefinitionen implementiert. Referenzielle Integritätsbedingungen werden mit der Klausel FOREIGN KEY und der Klausel REFERENCES in der Anweisung CREATE TABLE oder ALTER TABLE definiert. Weitere Informationen zu den Auswirkungen einer referenziellen Integritätsbedingung auf typisierte Tabellen oder übergeordnete typisierte Tabellen enthält das Handbuch *SQL Reference*.

Die Definition von Fremdschlüsseln implementiert Integritätsbedingungen für die Werte innerhalb der Zeilen einer Tabelle oder zwischen den Zeilen zweier Tabellen. Der Datenbankmanager prüft die in einer Tabellendefinition angegebenen Integritätsbedingungen und verwaltet die Abhängigkeitsbeziehungen entsprechend. Das Ziel besteht darin, die Integrität zu erhalten, wenn ein Datenbankobjekt auf ein anderes verweist.

Zum Beispiel enthalten sowohl der Primärschlüssel als auch der Fremdschlüssel eine Spalte für die Abteilungsnummer. In der Tabelle EMPLOYEE heißt der Spaltenname WORKDEPT, während in der Tabelle DEPARTMENT der Name DEPTNO ist. Die Abhängigkeitsbeziehung zwischen diesen beiden Tabellen wird durch die folgenden Integritätsbedingungen definiert

- Für jeden Mitarbeiter in der Tabelle EMPLOYEE gibt es nur eine Abteilungsnummer, und diese Nummer ist in der Tabelle DEPARTMENT vorhanden.
- Jede Zeile in der Tabelle EMPLOYEE verweist auf nur eine Zeile in der Tabelle DEPARTMENT. Es gibt eine eindeutige Beziehung zwischen den Tabellen.
- Jede Zeile in der Tabelle EMPLOYEE, die einen Nichtnullwert in der Spalte WORKDEPT enthält, hat eine Beziehung zu einer Zeile in der Spalte DEPTNO der Tabelle DEPARTMENT.
- Die Tabelle DEPARTMENT ist eine übergeordnete Tabelle, und die Tabelle EMPLOYEE ist die abhängige Tabelle.

Die SQL-Anweisung zur Definition der übergeordneten Tabelle DEPARTMENT sieht folgendermaßen aus:

```
CREATE TABLE DEPARTMENT
  (DEPTNO   CHAR(3)    NOT NULL,
   DEPTNAME VARCHAR(29) NOT NULL,
   MGRNO    CHAR(6),
   ADMRDEPT CHAR(3)    NOT NULL,
   LOCATION CHAR(16),
   PRIMARY KEY (DEPTNO))
IN RESOURCE
```

Die SQL-Anweisung zur Definition der abhängigen Tabelle EMPLOYEE sieht folgendermaßen aus:

```
CREATE TABLE EMPLOYEE
  (EMPNO    CHAR(6)      NOT NULL PRIMARY KEY,
   FIRSTNAME VARCHAR(12) NOT NULL,
   LASTNAME  VARCHAR(15) NOT NULL,
   WORKDEPT  CHAR(3),
   PHONENO   CHAR(4),
   PHOTO     BLOB(10m)   NOT NULL,
   FOREIGN KEY DEPT (WORKDEPT)
   REFERENCES DEPARTMENT ON DELETE NO ACTION)
IN RESOURCE
```

Durch Angeben der Spalte DEPTNO als Primärschlüssel der Tabelle DEPARTMENT und der Spalte WORKDEPT als Fremdschlüssel der Tabelle EMPLOYEE definieren Sie eine referenzielle Integritätsbedingung für die Werte der Spalte WORKDEPT. Durch diese Integritätsbedingung wird die referenzielle Integrität zwischen den Werten der beiden Tabellen implementiert. In vorliegenden Fall müssen alle Mitarbeiter (Employees), die der Tabelle EMPLOYEE hinzugefügt werden, eine Abteilungsnummer erhalten, die in der Tabelle DEPARTMENT enthalten ist.

Die Löschbedingung für die referenzielle Integritätsbedingung in der Tabelle EMPLOYEE ist NO ACTION. Das heißt, dass eine Abteilung (Department) nicht aus der Tabelle DEPARTMENT gelöscht werden kann, wenn es Mitarbeiter in dieser Abteilung gibt.

Zum Hinzufügen einer referenziellen Integritätsbedingung kann nicht nur die in den Beispielen gezeigte Anweisung CREATE TABLE, sondern auch die Anweisung ALTER TABLE verwendet werden. Siehe „Ändern von Struktur und Inhalt einer Tabelle“ auf Seite 218.

Ein weiteres Beispiel: Es werden dieselben Tabellendefinitionen wie im vorigen Beispiel verwendet. Außerdem wird die Tabelle DEPARTMENT vor der Tabelle EMPLOYEE erstellt. Jede Abteilung hat einen Manager, und dieser Manager wird in der Tabelle EMPLOYEE aufgeführt. Die Spalte MGRNO der Tabelle DEPARTMENT stellt tatsächlich einen Fremdschlüssel der Tabelle EMPLOYEE dar. Aufgrund dieses referenziellen Zyklus führt diese Integritätsbedingung zu einem kleinen Problem. Sie könnten einen Fremdschlüssel später hinzufügen (siehe „Hinzufügen von Primär- und Fremdschlüsseln“ auf Seite 223). Sie könnten außerdem die Anweisung CREATE SCHEMA verwenden, um die Tabellen EMPLOYEE und DEPARTMENT gleichzeitig zu erstellen (siehe Beispiel im Handbuch *SQL Reference*).

Klausel FOREIGN KEY: Ein Fremdschlüssel verweist auf einen Primärschlüssel oder einen eindeutigen Schlüssel in derselben oder einer anderen Tabelle. Die Zuordnung eines Fremdschlüssels zeigt an, dass die referenzielle Integrität gemäß der angegebenen referenziellen Integritätsbedingungen erhalten bleiben soll. Ein Fremdschlüssel wird mit der Klausel FOREIGN KEY in der Anweisung CREATE TABLE bzw. ALTER TABLE definiert.

Die Anzahl der Spalten im Fremdschlüssel muss mit der Anzahl der Spalten im entsprechenden Primärschlüssel oder eindeutigen Schlüssel (auch als übergeordneter Schlüssel bezeichnet) der übergeordneten Tabelle übereinstimmen. Darüber hinaus müssen sich entsprechende Teile der Schlüsselspaltendefinitionen dieselben Datentypen und Datenlängen haben. Dem Fremdschlüssel kann ein *Integritätsbedingungsname* zugeordnet werden. Wenn Sie keinen Namen zuordnen, wird der Name automatisch zugeordnet.

Zur leichteren Handhabung wird empfohlen, einen *Integritätsbedingungsnamen* zuzuordnen und nicht den vom System generierten Namen zu verwenden.

Der Wert eines zusammengesetzten Fremdschlüssels stimmt mit dem Wert eines Primärschlüssels überein, **wenn** der Wert jeder Spalte des Fremdschlüssels gleich dem Wert der entsprechenden Spalte des Primärschlüssels ist. Ein Fremdschlüssel, der Nullwerte enthält, kann nicht mit den Werten des Primärschlüssels übereinstimmen, da per Definition ein Primärschlüssel keine Nullwerte enthalten darf. Ein Nullwert eines Fremdschlüssels ist immer gültig, ungeachtet der Werte seiner Nichtnullwertspalten.

Für die Definition von Fremdschlüsseln gelten die folgenden Regeln:

- Eine Tabelle kann viele Fremdschlüssel enthalten.
- Ein Fremdschlüssel erfordert keine Eingabe (d. h. kann Nullwerte enthalten), wenn für irgendeinen seiner Teile keine Eingabe erforderlich ist (Nullwerte zulässig sind).
- Ein Fremdschlüsselwert ist ein Nullwert, wenn irgendeiner seiner Teile ein Nullwert ist.

Klausel REFERENCES: Mit der Klausel REFERENCES werden die übergeordnete Tabelle in einer Abhängigkeitsbeziehung identifiziert und die nötigen Integritätsbedingungen definiert. Diese Klausel kann in einer Spaltendefinition oder als separate Klausel, die eine Klausel FOREIGN KEY begleitet, in einer Anweisung CREATE TABLE bzw. ALTER TABLE angegeben werden.

Wenn die Klausel REFERENCES als Spaltenintegritätsbedingung angegeben wird, wird eine implizite Spaltenliste aus dem bzw. den Spaltennamen, die aufgelistet werden, gebildet. Beachten Sie, dass mehrere Spalten getrennte Klauseln REFERENCES haben können und dass eine einzelne Spalte mehr als eine Klausel REFERENCES haben kann.

In die Klausel REFERENCES ist eine Löschbedingung eingeschlossen. Im gezeigten Beispiel wird die Löschbedingung ON DELETE NO ACTION verwendet, die angibt, dass keine Abteilung gelöscht werden kann, wenn ihr Mitarbeiter zugeordnet sind. Andere Löschbedingungen sind ON DELETE CASCADE, ON DELETE SET NULL, und ON DELETE RESTRICT. Weitere Informationen zu DELETE-Regeln beim Implementieren der referenziellen Integrität finden Sie im Handbuch *Systemverwaltung: Konzept*.

Auswirkungen auf Dienstprogramme: Das Dienstprogramm zum Laden (LOAD) inaktiviert die Prüfung der Integritätsbedingungen für auf sich selbst verweisende und abhängige Tabellen und setzt diese Tabellen in einen Status, der besagt, dass die Prüfung noch ansteht. Nach Beendigung des Dienstprogramms zum Laden (LOAD) müssen Sie die Prüfung der Integritätsbedingungen für alle Tabellen, für die diese Prüfung inaktiviert wurde,

wieder aktivieren. Wenn beispielsweise die Tabellen DEPARTMENT und EMPLOYEE die einzigen Tabellen sind, die in den Status „Überprüfung anstehend“ gesetzt wurden, können Sie folgenden Befehl ausführen:

```
SET INTEGRITY FOR DEPARTMENT, EMPLOYEE IMMEDIATE CHECKED
```

Das Dienstprogramm zum Importieren (IMPORT) wird auf folgende Weise durch die referenziellen Integritätsbedingungen beeinflusst:

- Die Funktionen REPLACE und REPLACE CREATE sind nicht zulässig, wenn die Objektabelle außer sich selbst noch andere abhängige Tabellen aufweist.

Um diese Funktionen verwenden zu können, müssen erst alle Fremdschlüssel gelöscht werden, in denen die Tabelle eine übergeordnete Tabelle ist. Wenn die IMPORT-Operation abgeschlossen ist, erstellen Sie die Fremdschlüssel mit der Anweisung ALTER TABLE erneut.

- Der Erfolg eines Imports in eine Tabelle mit auf sich selbst verweisenden Integritätsbedingungen hängt von der Reihenfolge ab, in der die Zeilen importiert werden.

Definieren von Prüfungen auf Integritätsbedingungen in Tabellen

Eine Prüfung auf Integritätsbedingung in einer Tabelle (Table Check Constraint) gibt eine Suchbedingung an, die für jede Zeile der Tabelle implementiert wird, für die die Prüfung definiert ist. Sie können eine Prüfung auf Integritätsbedingung in einer Tabelle einrichten, indem Sie eine Prüfdefinition der Tabelle zuordnen, wenn die Tabelle erstellt (CREATE TABLE) bzw. geändert (ALTER TABLE) wird. Diese Prüfbedingung wird automatisch aktiviert, wenn durch eine Anweisung INSERT oder UPDATE die Daten in der Tabelle geändert werden. Eine Prüfung auf Integritätsbedingung in einer Tabelle hat keine Auswirkung auf eine Anweisung DELETE oder SELECT. Eine Prüfung auf Integritätsbedingung kann einer typisierten Tabelle zugeordnet werden.

Ein Integritätsbedingungsname kann nicht derselbe sein wie ein anderer Integritätsbedingungsname innerhalb derselben Anweisung CREATE TABLE. Wenn Sie keinen Integritätsbedingungsnamen angeben, wird vom System ein eindeutiger Bezeichner aus 18 Zeichen für die Integritätsbedingung generiert.

Eine Prüfung auf Integritätsbedingung in einer Tabelle wird zur Implementierung der Regeln für die Datenintegrität verwendet, die nicht durch die Eindeutigkeit des Schlüssels oder einer referenziellen Integritätsbedingung abgedeckt sind. In einigen Fällen kann eine Prüfbedingung verwendet werden, um Wertebereiche zu überprüfen. Die folgende Prüfung auf Integritätsbedingung in der Anweisung CREATE TABLE stellt sicher, dass das Anfangsdatum für jede Tätigkeit nicht nach dem Enddatum für dieselbe Tätigkeit liegt:

```

CREATE TABLE EMP_ACT
  (EMPNO      CHAR(6)      NOT NULL,
   PROJNO    CHAR(6)      NOT NULL,
   ACTNO     SMALLINT     NOT NULL,
   EMPTIME   DECIMAL(5,2),
   EMSTDATE  DATE,
   EMENDATE  DATE,
   CONSTRAINT ACTDATES CHECK(EMSTDATE <= EMENDATE) )
IN RESOURCE

```

Zum Hinzufügen einer Prüfung auf Integritätsbedingung kann nicht nur die im Beispiel gezeigte Anweisung `CREATE TABLE`, sondern auch die Anweisung `ALTER TABLE` verwendet werden. Siehe „Ändern von Struktur und Inhalt einer Tabelle“ auf Seite 218.

Definieren einer generierten Spalte in einer neuen Tabelle

Eine generierte Spalte wird in einer Basistabelle definiert, bei der der gespeicherte Wert mit Hilfe eines Ausdrucks ermittelt wird, anstatt mit einer Einfüge- oder Aktualisierungsoperation angegeben zu werden. Beim Erstellen einer Tabelle, bei der bestimmte Ausdrücke und Prädikate ständig verwendet werden, können eine oder mehrere generierte Tabellen zu dieser Tabelle hinzugefügt werden. Durch die Verwendung einer generierten Spalte ergibt sich die Möglichkeit, bei der Durchführung von Abfragen in den Tabellendaten eine verbesserte Leistung zu erzielen.

Es gibt z. B. zwei Situationen, bei denen die Bewertung von Ausdrücken aufwendig sein kann, wenn die Leistung wichtig ist:

1. Die Bewertung des Ausdrucks muss während einer Abfrage oft durchgeführt werden.
2. Die Berechnung ist komplex.

Um die Leistung einer Abfrage zu verbessern, können Sie eine zusätzliche Spalte definieren, die die Resultate des Ausdrucks enthält. Beim Absetzen einer Abfrage mit demselben Ausdruck kann die generierte Spalte dann direkt verwendet werden. Andernfalls kann die Komponente zum erneuten Schreiben von Abfragen, die im Optimierungsprogramm implementiert ist, den Ausdruck gegen die generierte Spalte austauschen.

Es ist außerdem möglich, für eine generierte Spalte einen nicht eindeutigen Index zu erstellen.

Wenn bei Abfragen Daten von zwei oder mehr Tabellen verknüpft werden, ermöglicht das Hinzufügen einer generierten Spalte dem Optimierungsprogramm die Auswahl potentiell besserer Verknüpfungsstrategien.

Im Folgenden ist ein Beispiel für das Definieren einer generierten Spalte für die Anweisung CREATE TABLE aufgeführt:

```
CREATE TABLE t1 (c1 INT,  
                 c2 DOUBLE,  
                 c3 DOUBLE GENERATED ALWAYS AS (c1 + c2)  
                 c4 GENERATED ALWAYS AS  
                 (CASE WHEN c1 > c2 THEN 1 ELSE NULL END))
```

Nach dem Erstellen dieser Tabelle können mit Hilfe der generierten Spalten Indizes erstellt werden. Beispiel:

```
CREATE INDEX i1 ON t1(c4)
```

Abfragen können die Vorteile generierter Spalten nutzen. Beispiel:

```
SELECT COUNT(*) FROM t1 WHERE c1 > c2
```

Diese Anweisung kann folgendermaßen angegeben werden:

```
SELECT COUNT(*) FROM t1 WHERE c4 IS NOT NULL
```

Zweites Beispiel:

```
SELECT c1 + c2 FROM t1 WHERE (c1 + c2) * c1 > 100
```

Diese Anweisung kann folgendermaßen angegeben werden:

```
SELECT c3 FROM t1 WHERE c3 * c1 > 100
```

Generierte Spalten können zur Verbesserung der Leistung von Abfragen eingesetzt werden. Aus diesem Grund werden generierte Spalten häufig nach dem Erstellen und Füllen einer Tabelle mit Werten zu dieser hinzugefügt. Weitere Informationen finden Sie in „Erstellen und Auffüllen einer Tabelle“ auf Seite 135.

Erstellen einer benutzerdefinierten temporären Tabelle

Mit der Anweisung DECLARE GLOBAL TEMPORARY TABLE werden temporäre Tabellen definiert. Die Anweisung wird innerhalb der Anwendung abgesetzt. Die benutzerdefinierte temporäre Tabelle bleibt nur so lange erhalten, bis die Anwendung die Verbindung zur Datenbank trennt.

Die Beschreibung dieser Tabelle wird im Systemkatalog nicht angezeigt, so dass sie von anderen Anwendungen nicht identifiziert und gemeinsam verwendet werden kann.

Wenn die Anwendung, die diese Tabelle verwendet, beendet wird oder die Verbindung zur Datenbank trennt, werden alle Daten in dieser Tabelle gelöscht und auch die Tabelle wird implizit gelöscht.

Im Folgenden ist ein Beispiel aufgeführt, wie eine temporäre Tabelle definiert werden kann:

```
DECLARE GLOBAL TEMPORARY TABLE gbl_temp
  LIKE emp1tab1
  ON COMMIT DELETE ROWS
  NOT LOGGED
  IN usr_tbsp
```

Diese Anweisung erstellt eine temporäre Benutzertabelle mit dem Namen `gbl_temp`. Die temporäre Benutzertabelle wurde mit Spalten definiert, die über exakt dieselben Namen und Beschreibungen verfügen wie die Spalten von `emp1tab1`. Die implizite Definition umfasst nur den Spaltennamen, den Datentyp, Angaben zur Verwendung von Nullwerten sowie die Standardwertattribute für die Spalte. Alle anderen Spaltenattribute einschließlich der eindeutigen Integritätsbedingungen, Auslöser und Indizes werden nicht definiert. Bei der Ausführung einer COMMIT-Operation werden alle Daten in der Tabelle gelöscht, wenn für die Tabelle kein WITH HOLD-Cursor geöffnet wurde. Die an der temporären Benutzertabelle vorgenommenen Änderungen werden nicht protokolliert. Die temporäre Benutzertabelle wird im angegebenen temporären Benutzertabellenbereich gespeichert. Dieser Tabellenbereich muss vorhanden sein, da andernfalls die Deklaration der Tabelle fehlschlägt.

Zusätzliche Informationen zur Anweisung `DECLARE GLOBAL TEMPORARY TABLE` finden Sie im Handbuch *SQL Reference*.

Anmerkung: Eine benutzerdefinierte temporäre Tabelle bietet keine Unterstützung für die folgenden Elemente:

- LOB-Spalten (oder Spalten eines anderen Typs, die ebenfalls auf LOB basieren)
- Benutzerdefinierte Spalten
- LONG VARCHAR-Spalten
- DATALINK-Spalten

Definieren einer Identitätsspalte in einer neuen Tabelle

Eine *Identitätsspalte* ermöglicht DB2 das automatische Generieren eines garantiert eindeutigen, numerischen Wertes für jede Zeile, die zu der Tabelle hinzugefügt wird. Beim Erstellen einer Tabelle, bei der bekannt ist, dass jede zur Tabelle hinzugefügte Zeile eindeutig identifiziert werden muss, können Sie eine Identitätsspalte zur Tabelle hinzufügen.

Nach der Erstellung können Sie die Tabellenbeschreibung nicht mehr ändern und eine Identitätsspalte einfügen.

Identitätsspalten können mit der Klausel `AS IDENTITY` der Anweisung `CREATE TABLE` angegeben werden.

Im Folgenden ist ein Beispiel für das Definieren einer Identitätsspalte für die Anweisung CREATE TABLE aufgeführt:

```
CREATE TABLE table (co11 INT,  
                    co12 DOUBLE,  
                    co13 INT NOT NULL GENERATED ALWAYS AS IDENTITY  
                    (START WITH 100, INCREMENT BY 5))
```

In diesem Beispiel wurde die dritte Spalte als Identitätsspalte definiert. Sie können darüber hinaus den in der Spalte verwendeten Wert zum eindeutigen Identifizieren aller hinzugefügten Zeilen verwenden. Im vorliegenden Beispiel wird für die erste eingegebene Zeile der Wert „100“ in der Spalte platziert. Für jede nachfolgend zu der Tabelle hinzugefügte Zeile wird der zugehörige Wert um 5 erhöht.

Außerdem können Identitätsspalten für die Zuordnung von Bestellnummern, Personalnummern, Produktnummern oder Ereignisnummern verwendet werden. Die Werte für eine Identitätsspalte können unter DB2 mit ALWAYS oder BY DEFAULT generiert werden.

Eine mit GENERATED ALWAYS definierte Identitätsspalte ist garantiert eindeutig. Die verwendeten Werte werden immer mit DB2 generiert. Für Anwendungen ist die Angabe eines expliziten Wertes nicht zulässig. Eine mit GENERATED BY DEFAULT definierte Identitätsspalte ermöglicht Anwendungen die explizite Angabe eines Wertes für die Identitätsspalte. Wenn die Anwendung keinen Wert bereitstellt, wird ein entsprechender Wert von DB2 generiert. Da der Wert von der Anwendung gesteuert wird, kann seine Eindeutigkeit durch DB2 nicht garantiert werden. Die Klausel GENERATED BY DEFAULT sollte für die Datenweitergabe eingesetzt werden, wenn der Inhalt einer vorhandenen Tabelle kopiert werden soll. Andere Einsatzbereiche sind das Entladen und das erneute Laden einer Tabelle.

Anmerkung: Identitätsspalten werden momentan in Umgebungen mit partitionierten Datenbanken nicht unterstützt.

Wenn Zeilen in eine Tabelle mit der Angabe expliziter Werte für Identitätsspalten eingefügt werden, wird der nächste intern generierte Wert nicht aktualisiert und kann Konflikte mit anderen in der Tabelle vorhandenen Werten verursachen. Doppelte Werte generieren eine Fehlermeldung.

Zusätzliche Informationen zum Definieren einer Identitätsspalte in einer neuen Tabelle finden Sie im Handbuch *SQL Reference*.

Erstellen einer Sequenz

Eine *Sequenz* ist ein Datenbankobjekt, das die automatische Generierung von Werten ermöglicht. Sequenzen sind ideal für die Aufgabe der Generierung eindeutiger Schlüsselwerte geeignet. Anwendungen können Sequenzen verwenden, um den möglichen gemeinsamen Zugriff und Leistungsprobleme wegen der Generierung eines eindeutigen Zählers außerhalb der Datenbank zu vermeiden.

Im Gegensatz zu einem Identitätsspaltenattribut ist eine Sequenz weder an eine bestimmte Tabellenspalte noch an eine eindeutige Tabellenspalte gebunden, wobei der Zugriff nur über diese Tabellenspalte erfolgen kann.

Eine Sequenz kann erstellt oder geändert werden, so dass mit einer der folgenden Methoden Werte generiert werden:

- Inkrement oder Dekrement monoton ohne Binden
- Inkrement oder Dekrement monoton bis zu einer benutzerdefinierten Begrenzung oder einem benutzerdefinierten Stopp.
- Inkrement oder Dekrement monoton bis zu einer benutzerdefinierten Begrenzung und zyklisch wieder zurück zum Anfang und Start.

Es folgt ein Beispiel für das Erstellen eines Sequenzobjekts:

```
CREATE SEQUENCE order_seq
  START WITH 1
  INCREMENT BY 1
  NOMAXVALUE
  NOCYCLE
  CACHE 24
```

In diesem Beispiel wird die Sequenz `order_seq` genannt. Die Sequenz beginnt mit 1 und wird ohne Begrenzung nach oben um jeweils 1 erhöht. Es gibt keinen Grund, zum Anfang zurückzukehren und wieder mit 1 zu beginnen, da keine obere Begrenzung zugeordnet wurde. Die dem Parameter `CACHE` zugeordnete Anzahl gibt die maximale Anzahl der Sequenzwerte an, die der Datenbankmanager vorab zuordnet und speichert.

Die generierten Sequenznummern haben folgende Merkmale:

- Die Werte können einen beliebigen exakt numerischen Datentyp (Exact Numeric) ohne Kommastellen aufweisen. Solche Datentypen sind: `SMALLINT`, `BIGINT`, `INTEGER` und `DECIMAL`.
- Nachfolgende Werte können sich um ein beliebiges angegebenes ganzzahliges Inkrement unterscheiden. Der Standardwert für das Inkrement ist 1.
- Der Wert des Zählers ist wiederherstellbar. Der Wert des Zählers wird mit Hilfe der Protokolle wiederhergestellt, wenn ein Wiederanlauf erforderlich ist.

- Werte können zwischengespeichert werden, wenn dies die Leistung verbessert. Durch das Vorabzuordnen und Speichern von Werten im Cache wird die synchrone E/A für das Protokoll verkleinert, wenn Werte für die Sequenz generiert werden. Im Falle eines Systemfehlers gelten alle zwischengespeicherten Werte, die noch nicht festgeschrieben wurden, als nicht verwendet und werden somit als verloren angesehen. Der für CACHE angegebene Wert ist die maximale Anzahl der Sequenzwerte, die verloren gehen könnten.

Wenn eine Datenbank, die mindestens eine Sequenz enthält, für einen früheren Zeitpunkt wiederhergestellt wird, kann dies zur Generierung doppelter Werte für einige Sequenzen führen. Um mögliche Doppelwerte zu vermeiden, sollte eine Datenbank mit Sequenzen nicht für einen früheren Zeitpunkt wiederhergestellt werden.

Sequenzen werden nur in Datenbanken mit Einzelknoten unterstützt.

Es gibt zwei Ausdrücke, die mit einer Sequenz verwendet werden.

Der Ausdruck PREVVAL gibt den zuletzt generierten Wert für die angegebene Sequenz für eine vorherige Anweisung innerhalb der aktuellen Sitzung zurück.

Der Ausdruck NEXTVAL gibt den nächsten Wert für die angegebene Sequenz zurück. Eine neue Sequenznummer wird generiert, wenn ein Ausdruck NEXTVAL den Namen der Sequenz angibt. Wenn es jedoch mehrere Exemplare eines Ausdrucks NEXTVAL gibt, die denselben Sequenznamen innerhalb einer Abfrage angeben, wird der Zähler für die Sequenz nur einmal für jede Zeile des Ergebnisses erhöht.

Dieselbe Sequenznummer kann als eindeutiger Schlüsselwert in zwei separaten Tabellen verwendet werden, indem auf die Sequenznummer mit einem Ausdruck NEXTVAL für die erste Tabelle und mit einem Ausdruck PREVVAL für eine weitere Tabelle verwiesen wird.

Beispiel:

```
INSERT INTO order (orderno, custno)
VALUES (NEXTVAL FOR order_seq, 123456);
INSERT INTO line_item (orderno, partno, quantity)
VALUES (PREVVAL FOR order_seq, 987654, 1)
```

Die Ausdrücke NEXTVAL und PREVVAL können an folgenden Positionen verwendet werden:

- Anweisung INSERT, Klausel VALUES
- Anweisung SELECT, Liste SELECT
- Zuordnungsanweisung SET
- Anweisung UPDATE, Klausel SET
- Anweisung VALUES und VALUES INTO

Vergleich von Identitätsspalten (IDENTITY) und Sequenzen

Zwischen Identitätsspalten und Sequenzen gibt es sowohl Gemeinsamkeiten als auch Unterschiede. Die Kenndaten von Identitätsspalten und Sequenzen können bei Analyse und Entwurf von Datenbanken und Anwendungen verwendet werden.

Eine Identitätsspalte hat folgende Kenndaten:

- Eine Identitätsspalte kann als Teil einer Tabelle definiert werden, wenn die Tabelle erstellt wird. Wenn eine Tabelle bereits erstellt ist, können Sie sie nicht ändern, um eine Identitätsspalte hinzuzufügen. (Spaltenmerkmale einer vorhandenen Identitätsspalte können jedoch geändert werden.)
- Eine Identitätsspalte erzeugt automatisch Werte für eine einzelne Tabelle.
- Wenn eine Identitätsspalte als GENERATED ALWAYS definiert ist, werden die verwendeten Werte immer vom Datenbankmanager generiert. Es ist nicht zulässig, dass Anwendungen bei der Änderung des Inhalts der Tabelle eigene Werte bereitstellen.

Ein Sequenzobjekt hat die folgenden Kenndaten:

- Ein Sequenzobjekt ist ein Datenbankobjekt, das an keine Tabelle gebunden ist.
- Ein Sequenzobjekt generiert sequenzielle Werte, die in einer beliebigen SQL-Anweisung verwendet werden können.
- Da ein Sequenzobjekt von einer beliebigen Anwendung verwendet werden kann, gibt es zwei Ausdrücke, die das Abrufen des nächsten Wertes in der angegebenen Sequenz und des Wertes, der vor der gerade ausgeführten Anweisung generiert wurde, steuern. Der Ausdruck PREVVAL gibt den zuletzt generierten Wert für die angegebene Sequenz für eine vorherige Anweisung innerhalb der aktuellen Sitzung zurück. Der Ausdruck NEXTVAL gibt den nächsten Wert für die angegebene Sequenz zurück. Die Verwendung dieser Ausdrücke ermöglicht es, denselben Wert in verschiedenen SQL-Anweisungen innerhalb mehrerer Tabellen zu verwenden.

Dies sind noch nicht alle Kenndaten dieser beiden Elemente. Die vorgestellten Kenndaten werden Sie aber dabei unterstützen, abhängig vom Datenbankentwurf und den Anwendungen, die die Datenbank verwenden, das richtige Element auszuwählen.

Erstellen einer typisierten Tabelle

Sie können eine typisierte Tabelle unter Verwendung einer Variante der Anweisung CREATE TABLE erstellen. Umfassende Informationen zu typisierten Tabellen finden Sie im Handbuch *Application Development Guide*.

Auffüllen einer typisierten Tabelle

Nach dem Erstellen der strukturierten Typen können Sie eine typisierte Tabelle mit Daten füllen. Anschließend können Sie die entsprechenden Tabellen und untergeordneten Tabellen erstellen. Umfassende Informationen zu typisierten Tabellen finden Sie im Handbuch *Application Development Guide*.

Hierarchietabelle

Eine Hierarchietabelle ist eine Tabelle, die der Implementierung einer Hierarchie typisierter Tabellen zugeordnet ist. Sie wird zum gleichen Zeitpunkt erstellt wie die Stammtabelle der Hierarchie. Umfassende Informationen zu Hierarchietabellen finden Sie im Handbuch *Application Development Guide*.

Erstellen einer Tabelle in mehreren Tabellenbereichen

Tabellendaten können im selben Tabellenbereich wie der Tabellenindex und Daten in Spalten für große Objekte (LOBs) gespeichert werden, die der Tabelle zugeordnet sind. Sie können den Index und die Daten in Spalten für große Objekte (LOBs) auch in separaten Tabellenbereichen getrennt von dem Tabellenbereich für die restlichen Tabellendaten speichern. Alle Tabellenbereiche müssen vor der Ausführung der Anweisung CREATE TABLE vorhanden sein. Die Trennung der Tabellenkomponenten ist nur bei DMS-Tabellenbereichen möglich.

Gehen Sie wie folgt vor, um eine Tabelle mit der Steuerzentrale in mehreren Tabellenbereichen zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Ordner **Tabellen** und wählen Sie dann im Kontextmenü **Erstellen** → **Tabellen mit Assistent** aus.
3. Geben Sie den Tabellennamen ein und klicken Sie dann auf **Weiter**.
4. Wählen Sie die gewünschten Spalten für Ihre Tabelle aus.
5. Klicken Sie auf der Seite **Tabellenbereich** auf **Separaten Indexbereich verwenden** und **Separaten LOB-Bereich verwenden**. Geben Sie die erforderlichen Informationen an und klicken Sie dann auf **Fertigstellen**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Tabelle in mehreren Tabellenbereichen zu erstellen:

```
CREATE TABLE <name>
  (<spaltenname> <datentyp> <>nullattribut>)
  IN <tabellebereichsname>
  INDEX IN <indexbereichsname>
  LONG IN <lob_bereichsname>
```

Das folgende Beispiel zeigt, wie die Tabelle EMP_PHOTO erstellt werden könnte, um die verschiedenen Teile der Tabelle in verschiedenen Tabellenbereichen zu speichern:

```
CREATE TABLE EMP_PHOTO
  (EMPNO          CHAR(6)          NOT NULL,
   PHOTO_FORMAT  VARCHAR(10)     NOT NULL,
   PICTURE       BLOB(100K) )
  IN RESOURCE
  INDEX IN RESOURCE_INDEXES
  LONG IN RESOURCE_PHOTO
```

In diesem Beispiel werden die Daten der Tabelle EMP_PHOTO folgendermaßen gespeichert:

- Für die Tabelle EMP_PHOTO erstellte Indizes werden im Tabellenbereich RESOURCE_INDEXES gespeichert.
- Die Daten für die Spalte PICTURE werden im Tabellenbereich RESOURCE_PHOTO gespeichert.
- Die Daten für die Spalten EMPNO und PHOTO_FORMAT werden im Tabellenbereich RESOURCE gespeichert.

Zusätzliche Überlegungen zur Verwendung mehrerer DMS-Tabellenbereiche für eine einzelne Tabelle finden Sie im Handbuch *Systemverwaltung: Konzept*.

Weitere Informationen zur Anweisung CREATE TABLE finden Sie im Handbuch *SQL Reference*.

Erstellen einer Tabelle in einer partitionierten Datenbank

Bevor Sie eine Tabelle erstellen, die physisch geteilt oder partitioniert wird, müssen Sie folgende Punkte beachten:

- Tabellenbereiche können sich über mehr als eine Datenbankpartition erstrecken. Die Anzahl der Partitionen, über die sie sich erstrecken, hängt von der Anzahl der Partitionen in der Knotengruppe ab.
- Tabellen können in einer Kollokation zusammengefasst werden, indem sie im selben Tabellenbereich gespeichert werden oder in einem anderen Tabellenbereich, der zusammen mit dem ersten Tabellenbereich derselben Knotengruppe zugeordnet ist. Weitere Informationen finden Sie in *Systemverwaltung: Konzept*.

Es gibt eine zusätzliche Option bei der Erstellung einer Tabelle in einer partitionierten Datenbankumgebung: den *Partitionierungsschlüssel*. Ein Partitionierungsschlüssel ist ein Schlüssel, der Teil der Definition einer Tabelle ist. Durch ihn wird die Partition bestimmt, in der die jeweilige Datenzeile gespeichert wird.

Die Auswahl eines geeigneten Partitionierungsschlüssels ist von großer Bedeutung, da *der Schlüssel später nicht mehr geändert werden kann*. Darüber hinaus müssen alle eindeutigen Indizes (und infolgedessen eindeutige Schlüssel bzw. Primärschlüssel) als Obermenge des Partitionierungsschlüssels definiert werden. Das heißt, wenn ein Partitionierungsschlüssel definiert wird, müssen eindeutige Schlüssel und Primärschlüssel alle die Spalten enthalten, die der Partitionierungsschlüssel enthält (sie können mehr Spalten haben).

Wenn Sie keinen Partitionierungsschlüssel explizit angeben, werden die folgenden Standardwerte verwendet. *Stellen Sie sicher, dass der Standardpartitionierungsschlüssel geeignet ist*.

- Wenn ein Primärschlüssel in der Anweisung CREATE TABLE angegeben wird, wird die erste Spalte des Primärschlüssels als Partitionierungsschlüssel verwendet.
- Wenn es keinen Primärschlüssel gibt, wird die erste Spalte verwendet, die keine Langfelddaten enthält.
- Wenn die vorhandenen Spalten die Anforderungen für einen Standardpartitionierungsschlüssel nicht erfüllen, wird die Tabelle ohne Partitionierungsschlüssel erstellt (d. h., sie ist nur in Knotengruppen mit einer Einzelpartition zulässig).

Es folgt ein Beispiel:

```
CREATE TABLE MIXREC (MIX_CNTL INTEGER NOT NULL,  
                    MIX_DESC CHAR(20) NOT NULL,  
                    MIX_CHR CHAR(9) NOT NULL,  
                    MIX_INT INTEGER NOT NULL,  
                    MIX_INTS SMALLINT NOT NULL,  
                    MIX_DEC DECIMAL NOT NULL,  
                    MIX_FLT FLOAT NOT NULL,  
                    MIX_DATE DATE NOT NULL,  
                    MIX_TIME TIME NOT NULL,  
                    MIX_TMSTMP TIMESTAMP NOT NULL)  
IN MIXTS12  
PARTITIONING KEY (MIX_INT) USING HASHING
```

In diesem Beispiel ist der Tabellenbereich MIXTS12 und der Partitionierungsschlüssel MIX_INT. Wenn der Partitionierungsschlüssel nicht explizit angegeben würde, würde die Spalte MIX_CNTL verwendet. (Wenn kein Primärschlüssel angegeben und kein Partitionierungsschlüssel definiert wird, wird die erste Spalte in der Liste, die keine Langfelddaten (LONG) enthält, als Partitionierungsschlüssel verwendet.)

Eine Zeile einer Tabelle und sämtliche Informationen zu dieser Zeile werden immer in derselben Datenbankpartition gespeichert.

Die Größenbegrenzung für eine Partition einer Tabelle ist 64 GB oder der verfügbare Plattenspeicherplatz, je nachdem, welcher Wert kleiner ist. (Hierbei wird von einer 4-KB-Seitengröße für den Tabellenbereich ausgegangen.) Die Größe der Tabelle kann auf bis zu 64 GB (oder dem verfügbaren Plattenspeicherplatz) multipliziert mit der Anzahl der Datenbankpartitionen anwachsen. Wenn die Seitengröße für den Tabellenbereich 8 KB beträgt, kann die Größe der Tabelle auf bis zu 128 GB (oder dem verfügbaren Plattenspeicherplatz) multipliziert mit der Anzahl der Datenbankpartitionen anwachsen. Wenn die Seitengröße für den Tabellenbereich 16 KB beträgt, kann die Größe der Tabelle auf bis zu 256 GB (oder den verfügbaren Plattenspeicherplatz) multipliziert mit der Anzahl der Datenbankpartitionen anwachsen. Wenn die Seitengröße für den Tabellenbereich 32 KB beträgt, kann die Größe der Tabelle auf bis zu 512 GB (oder den verfügbaren Plattenspeicherplatz) multipliziert mit der Anzahl der Datenbankpartitionen anwachsen.

Erstellen eines Auslösers

Ein Auslöser (Trigger) definiert eine Reihe von Aktionen, die in Verbindung mit bzw. ausgelöst durch eine Anweisung INSERT, UPDATE oder DELETE für eine angegebene Basistabelle bzw. die typisierte Tabelle ausgeführt werden. Auslöser können beispielsweise zu folgenden Aktionen verwendet werden:

- Überprüfen der Gültigkeit von Eingabedaten
- Generieren eines Werts für eine neu eingefügte Zeile
- Lesen aus anderen Tabellen zur Auflösung von Querverweisen
- Schreiben in andere Tabellen zur Führung von Prüfprotokollen

Auslöser können nicht mit Kurznamen verwendet werden.

Auslöser können zur Unterstützung allgemeiner Formen der Datenintegrität und Geschäftsregeln verwendet werden. Zum Beispiel kann ein Auslöser die Kreditgrenze eines Kunden überprüfen, bevor eine Bestellung entgegengenommen oder eine Tabelle mit Übersichtsdaten aktualisiert wird.

Die Vorteile der Verwendung von Auslösern lassen sich wie folgt beschreiben:

- Schnellere Anwendungsentwicklung: Da ein Auslöser in der Datenbank gespeichert wird, müssen die durch den Auslöser ausgeführten Aktionen nicht mehr in jeder Anwendung codiert werden.
- Einfachere Wartung: Wenn ein Auslöser definiert ist, wird er automatisch aufgerufen, wenn auf die Tabelle, für die er erstellt wurde, zugegriffen wird.
- Globale Implementierung von Geschäftsregeln: Wenn sich die geschäftsinternen Abläufe oder Regeln ändern, müssen lediglich die Auslöser und nicht sämtliche Anwendungsprogramme geändert werden.

Gehen Sie wie folgt vor, um mit der Steuerzentrale einen Auslöser zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Auslöser** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Ordner **Auslöser** und wählen Sie dann im Kontextmenü **Erstellen** aus.
3. Geben Sie die erforderlichen Informationen für den Auslöser an.
4. Geben Sie die Aktion an, die der Auslöser aufrufen soll und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um einen Auslöser zu erstellen:

```
CREATE TRIGGER <name>  
    <aktion> ON <tabellenname>  
    <operation>  
    <ausgelöste_aktion>
```

Mit der folgenden SQL-Anweisung wird ein Auslöser erstellt, der die Anzahl der Mitarbeiter automatisch erhöht, wenn eine neue Person angestellt wird, indem zum Wert in der Spalte für die Anzahl der Mitarbeiter (NBEMP) in der Tabelle COMPANY_STATS jedes Mal der Wert 1 addiert wird, wenn der Tabelle EMPLOYEE eine Zeile hinzugefügt wird.

```
CREATE TRIGGER NEW_HIRED
  AFTER INSERT ON EMPLOYEE
  FOR EACH ROW MODE DB2SQL
  UPDATE COMPANY_STATS SET NBEMP = NBEMP+1;
```

Eine Auslöserdefinition kann eine oder mehrere der folgenden SQL-Anweisungen enthalten: Anweisung INSERT, gezielte Anweisung UPDATE, gezielte Anweisung DELETE, Gesamtauswahlen, Anweisung SET *übergangsvariable* und Anweisung SIGNAL SQLSTATE. Der Auslöser kann vor oder nach der Anweisung INSERT, UPDATE bzw. DELETE, auf die er bezogen ist, aktiviert werden. Vollständige Syntaxinformationen zur Anweisung CREATE TRIGGER finden Sie im Handbuch *SQL Reference*. Informationen zum Erstellen und Verwenden von Auslösern finden Sie im Handbuch *Application Development Guide*.

Anmerkung: Wenn es sich bei dem Auslöser um einen Vorauslöser (BEFORE) handelt, kann als von der ausgelösten Aktion angegebener Spaltenname keine generierte Spalte außer einer Identitätsspalte angegeben werden. Dies bedeutet, dass der generierte Identitätswert für Vorauslöser sichtbar ist.

Beim Erstellen eines automatischen Auslösers ist Vorsicht beim Zeichen für das ANWEISUNGSENDE geboten. Der Datenbankmanager erkennt als Standardeinstellung ein Semikolon (;) als ANWEISUNGSENDE. Sie sollten das Zeichen für das ANWEISUNGSENDE manuell in der Prozedur für die Erstellung des automatischen Auslösers editieren, um ein anderes Zeichen als das Semikolon (;) zu verwenden. Das Semikolon (;) könnte zum Beispiel durch ein anderes Sonderzeichen wie das Nummernzeichen bzw. Doppelkreuz (#) ersetzt werden.

Anschließend müssen Sie einen der folgenden Schritte ausführen:

- Wählen Sie in der Befehlszentrale die Indexzunge für Prozeduren aus. Ändern Sie den Begrenzer unter Tools—>Einstellungsmenü für Tools, und führen Sie die Prozedur aus.
- Geben Sie im Befehlszeilenprozessor folgendes ein:
db2 -td <begrenzer> -vf <prozedur>

wobei der Begrenzer das alternative Zeichen für das ANWEISUNGSENDE und <prozedur> die geänderte Prozedur mit dem neuen Begrenzer ist.

Auslöserabhängigkeiten

Alle Abhängigkeiten eines Auslösers von einem anderen Objekt werden im Katalog SYSCAT.TRIGDEP aufgezeichnet. Ein Auslöser kann von vielen Objekten abhängig sein. Diese Objekte und die abhängigen Auslöser werden im Handbuch *SQL Reference* im Abschnitt zur Anweisung DROP detailliert erläutert.

Wenn eines dieser Objekte gelöscht wird, wird der Auslöser außer Funktion gesetzt, aber die Definition bleibt im Katalog erhalten. Um einen solchen Auslöser wieder zu aktivieren, müssen Sie die entsprechende Definition aus dem Katalog abrufen und eine neue Anweisung CREATE TRIGGER ausführen.

Wenn ein Auslöser gelöscht wird (DROP), wird die entsprechende Beschreibung aus der Katalogsicht SYSCAT.TRIGGERS gelöscht, und es werden alle zugehörigen Einträge für Abhängigkeiten aus der Katalogsicht SYSCAT.TRIGDEP gelöscht. Alle Pakete mit Abhängigkeiten über Anweisungen UPDATE, INSERT oder DELETE für den Auslöser, werden ungültig gemacht.

Wenn das abhängige Objekt eine Sicht ist und außer Funktion gesetzt wird, wird auch der Auslöser als unbrauchbar markiert. Alle Pakete, die von Auslösern abhängig sind, die als unbrauchbar markiert wurden, werden ungültig gemacht. (Weitere Informationen finden Sie in „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 253.)

Erstellen einer benutzerdefinierten Funktion (UDF) oder Methode

Benutzerdefinierte Funktionen (UDF - User-defined Functions) erweitern und ergänzen die Unterstützung, die von integrierten SQL-Funktionen zur Verfügung gestellt wird, und können an allen Stellen verwendet werden, an denen integrierte Funktionen verwendet werden können. Benutzerdefinierte Funktionen können in folgenden zwei Formen erstellt werden:

- Externe Funktionen, die in einer Programmiersprache geschrieben sind
- Quellenfunktionen, deren Implementierung von einer anderen vorhandenen Funktion übernommen wurde

Es gibt drei Arten von UDFs:

Skalar Liefert einen einzelnen Wert zurück, wenn sie aufgerufen wird. Zum Beispiel ist die integrierte Funktion SUBSTR() eine Skalarfunktion. Benutzerdefinierte Skalarfunktionen können als externe Funktion oder Quellenfunktion erstellt werden.

Spalte Liefert einen einzelnen Wert aus einer Menge von typgleichen Werten (einer Spalte). Diese Art der Funktion wird in DB2 manchmal auch als Aggregatfunktion (Aggregation) bezeichnet. Ein Beispiel für eine Spaltenfunktion ist die integrierte Funktion AVG(). Eine externe

benutzerdefinierte Spaltenfunktion kann für DB2 nicht definiert werden, aber eine benutzerdefinierte Spaltenfunktion, die aus einer integrierten Spaltenfunktion abgeleitet wird, kann definiert werden. Dies bietet Vorteile bei der Verarbeitung benutzerdefinierter Datentypen.

Wenn es zum Beispiel einen benutzerdefinierten Datentyp SHOESIZE gibt, der mit dem Basistyp INTEGER definiert wurde, kann eine benutzerdefinierte Funktion AVG(SHOESIZE) erstellt werden, die aus der integrierten Funktion AVG(INTEGER) abgeleitet wird und daher eine Spaltenfunktion ist.

Table

Liefert eine Tabelle an die SQL-Anweisung zurück, von der sie aufgerufen wird. Tabellenfunktionen können nur in der Klausel FROM einer Anweisung SELECT aufgerufen werden. Eine solche Funktion kann dazu dienen, das SQL-Verarbeitungspotenzial auf Daten anzuwenden, die keine DB2-Daten sind, oder solche Daten in eine DB2-Tabelle umzuwandeln.

Eine Tabellenfunktion kann beispielsweise eine Datei in eine Tabelle umwandeln, Musterdaten aus dem World-Wide Web tabellarisieren oder auf eine Datenbank von Lotus Notes zugreifen und Informationen zurückliefern wie z. B. Datum, Absender und Text von Nachrichten der elektronischen Post. Diese Informationen können mit anderen Tabellen in der Datenbank verknüpft werden.

Eine Tabellenfunktion kann nur eine externe Funktion sein. Sie kann keine Quellenfunktion sein.

Informationen über vorhandene benutzerdefinierte Funktionen werden in den Katalogsichten SYSCAT.FUNCTIONS und SYSCAT.FUNCPARMS gespeichert. Der Systemkatalog enthält *nicht* den ausführbaren Code der UDF. (Berücksichtigen Sie daher bei der Erstellung der Sicherungs- und Wiederherstellungspläne auch die Verwaltung der ausführbaren Dateien Ihrer benutzerdefinierten Funktionen.)

Statistische Daten über die Leistung der UDFs spielen bei der Kompilierung der SQL-Anweisungen eine wichtige Rolle. Informationen zum Aktualisieren von UDF-Statistiken im Systemkatalog finden Sie im Abschnitt über das Aktualisieren von Statistiken für benutzerdefinierte Funktionen im Handbuch *Systemverwaltung: Optimierung*.

Einzelheiten zur Verwendung der Anweisung CREATE FUNCTION, um eine geeignete UDF für Ihre spezielle Anwendung zu schreiben, finden Sie im Handbuch *Application Development Guide*. Einzelheiten zur UDF-Syntax finden Sie im Handbuch *SQL Reference*.

Erstellen einer Funktionszuordnung

Erstellen Sie in einer zusammengeschlossenen Datenbank eine Funktionszuordnung, wenn Sie eine lokale Funktion oder Funktionsschablone (beschrieben in „Erstellen einer Funktionsschablone“ auf Seite 162) in einer oder mehreren Datenquellen einer Funktion zuordnen wollen. Standard-Funktionszuordnungen für viele Datenquellenfunktionen werden zur Verfügung gestellt.

Funktionszuordnungen sind in folgenden Situationen nützlich:

- Neue integrierte Funktionen werden in einer Datenquelle verfügbar.
- Sie wollen eine benutzerdefinierte Funktion in einer Datenquelle einer lokalen Funktion zuordnen.
- Eine Anwendung erfordert eine andere Standardarbeitsweise als die Standardzuordnung.

Funktionszuordnungen, die mit Anweisungen CREATE FUNCTION MAPPING definiert wurden, werden in der zusammengeschlossenen Datenbank gespeichert.

Funktionen (oder Funktionsschablonen) müssen über die gleiche Anzahl von Eingabeparametern verfügen wie die Datenquellenfunktion. Außerdem sollten die Datentypen der Eingabeparameter auf der zusammengeschlossenen Seite mit den Datentypen der Eingabeparameter auf der Datenquellenseite kompatibel sein. Diese Voraussetzungen gelten auch für die Rückgabewerte.

Verwenden Sie die Anweisung CREATE FUNCTION MAPPING, um eine Funktionszuordnung zu erstellen. Geben Sie beispielsweise Folgendes ein, um eine Funktionszuordnung zwischen einer Oracle-Funktion AVGNEW und einem DB2-Äquivalent auf dem Server ORACLE1 zu erstellen:

```
CREATE FUNCTION MAPPING ORAVGNEW FOR SYSIBM.AVG(INT) SERVER ORACLE1
OPTIONS (REMOTE_NAME 'AVGNEW')
```

Sie müssen über die Berechtigung SYSADM oder DBADM für die zusammengeschlossene Datenbank verfügen, um diese Anweisung verwenden zu können. Attribute von Funktionszuordnungen werden in SYSCAT.FUNCMAPPINGS gespeichert.

Der Server einer zusammengeschlossenen Datenbank bindet keine Eingabe-Host-Variablen und ruft keine Ergebnisse von LOB, LONG VARCHAR/VARGRAPHIC, DATALINK sowie von einzigartigen und strukturierten Datentypen ab. Es kann keine Funktionszuordnung erstellt werden, wenn ein Eingabeparameter oder der Rückgabewert einen dieser Typen enthält.

Weitere Einzelheiten zum Verwenden und Erstellen von Funktionszuordnungen finden Sie im Handbuch *Application Development Guide*. Einzelheiten zur Syntax von CREATE FUNCTION MAPPING finden Sie im Handbuch *SQL Reference*.

Erstellen einer Funktionsschablone

In Systemen zusammengesetzter Datenbanken dienen Funktionsschablonen als „Anker“ für Funktionszuordnungen. Sie ermöglichen das Zuordnen einer Datenquellenfunktion, wenn auf dem Server einer zusammengesetzten Datenbank keine entsprechende DB2-Funktion vorhanden ist. Für eine Funktionszuordnung muss eine Funktionsschablone oder eine ähnliche Funktion unter DB2 vorhanden sein.

Die Schablone ist nur eine Funktionshülle für Name, Eingabeparameter und Rückgabewert der Funktion. Es gibt keine lokale ausführbare Datei für die Funktion.

Daher kann es vorkommen, dass ein Aufruf der Funktionsschablone fehlschlägt, obwohl die Funktion in der Datenquelle verfügbar ist. Betrachten Sie zum Beispiel die folgende Abfrage:

```
SELECT myfunc(C1)
       FROM nick1
       WHERE C2 < 'A'
```

Wenn DB2 und die Datenquelle mit dem Objekt, auf das nick1 verweist, nicht die gleiche Sortierfolge aufweisen, schlägt die Abfrage fehl, weil der Vergleich in DB2 vorgenommen werden muss, die Funktion sich jedoch in der Datenquelle befindet. Wenn die Sortierfolgen gleich wären, könnte die Vergleichsoperation an der Datenquelle stattfinden, die die zugrundeliegende Funktion enthält, auf die myfunc verweist.

Funktionen (oder Funktionsschablonen) müssen über die gleiche Anzahl von Eingabeparametern verfügen wie die Datenquellenfunktion. Die Datentypen der Eingabeparameter auf der Seite der zusammengesetzten Datenbank sollten mit den Datentypen der Eingabeparameter auf der Datenquellenseite kompatibel sein. Diese Voraussetzungen gelten auch für die Rückgabewerte.

Zum Erstellen von Funktionsschablonen können Sie die Anweisung CREATE FUNCTION mit dem Schlüsselwort AS TEMPLATE verwenden. Nach dem Erstellen der Schablone ordnen Sie die Schablone mit Hilfe der Anweisung CREATE FUNCTION MAPPING der Datenquelle zu.

Geben Sie beispielsweise Folgendes ein, um eine Funktionsschablone und eine Funktionszuordnung für die Funktion MYS1FUNC auf dem Server S1 zu erstellen:

```
CREATE FUNCTION MYFUNC(INT) RETURNS INT AS TEMPLATE
```

```
CREATE FUNCTION MAPPING S1_MYFUNC FOR MYFUNC(INT) SERVER S1 OPTIONS  
(REMOTE_NAME 'MYS1FUNC')
```

Einzelheiten zum Verwenden und Erstellen von Funktionsschablonen finden Sie im Handbuch *Application Development Guide*. Einzelheiten zur Syntax von CREATE FUNCTION finden Sie im Handbuch *SQL Reference*.

Erstellen eines benutzerdefinierten Datentyps (UDT)

Ein benutzerdefinierter Datentyp (UDT) ist ein benannter Datentyp, der in der Datenbank durch den Benutzer erstellt wird. Ein UDT kann ein einzigartiger Datentyp sein, der eine gemeinsame Darstellung mit einem integrierten Datentyp oder einem strukturierten Typ hat, der eine Folge benannter Attribute mit jeweils einem Typ aufweist. Ein strukturierter Typ kann ein untergeordneter Typ eines anderen strukturierten Typs (eines so genannten übergeordneten Typs) sein, wodurch eine Typenhierarchie definiert wird.

UDTs unterstützen Eingabe mit festgelegtem Datentyp, was bedeutet, dass sie die gleiche Darstellung wie andere Typen haben. Werte eines bestimmten UDTs sind nur mit Werten des bzw. der gleichen UDTs in der gleichen Typenhierarchie kompatibel. Mit der Katalogsicht SYSCAT.DATATYPES können die benutzerdefinierten Datentypen, die für die Datenbank definiert wurden, angezeigt werden. Diese Katalogsicht zeigt darüber hinaus die Datentypen, die bei der Erstellung der Datenbank vom Datenbankmanager definiert wurden. Eine vollständige Liste aller Datentypen finden Sie im Handbuch *SQL Reference*.

Ein benutzerdefinierter einzigartiger Typ kann für die meisten der vom System bereitgestellten bzw. integrierten Funktionen nicht als Argument verwendet werden. Zur Implementierung dieser und anderer Operationen müssen daher benutzerdefinierte Funktionen bereitgestellt werden.

Ein benutzerdefinierter Datentyp kann nur unter folgenden Bedingungen gelöscht werden:

- Er wird in *keiner* Spaltendefinition für eine vorhandene Tabelle verwendet.
- Er wird *nicht* als Typ einer vorhandenen typisierten Tabelle oder typisierten Sicht verwendet.
- Er wird in *keiner* benutzerdefinierten Funktion verwendet, die nicht gelöscht werden kann. Eine benutzerdefinierte Funktion kann nicht gelöscht werden, wenn eine Sicht, ein Auslöser, eine Prüfung auf Integritätsbedingung in einer Tabelle oder eine andere benutzerdefinierte Funktion von ihr abhängig ist.

Wenn ein benutzerdefinierter Datentyp gelöscht wird, werden alle von ihm abhängigen Funktionen ebenfalls gelöscht.

Erstellen eines benutzerdefinierten einzigartigen Datentyps

Ein benutzerdefinierter einzigartiger Datentyp ist ein von einem vorhandenen Typ wie „ganze Zahl“, „Dezimalzahl“ oder „Zeichen“ abgeleiteter Datentyp. Sie können einen einzigartigen Datentyp mit Hilfe der Anweisung `CREATE DISTINCT TYPE` erstellen.

Mit der folgenden SQL-Anweisung wird der einzigartige Datentyp `t_educ` als ein `SMALLINT`-Typ erstellt:

```
CREATE DISTINCT TYPE T_EDUC AS SMALLINT WITH COMPARISONS
```

Exemplare desselben einzigartigen Datentyps können miteinander verglichen werden, wenn die Klausel `WITH COMPARISONS` in der Anweisung `CREATE DISTINCT TYPE` angegeben wird (wie im Beispiel). Die Klausel `WITH COMPARISONS` kann nicht angegeben werden, wenn der Quellendatentyp ein großes Objekt oder `DATALINK`, `LONG VARCHAR` bzw. `LONG VARGRAPHIC` ist.

Exemplare einzigartiger Typen können nicht als Argumente von Funktionen oder als Operanden für Operationen, die für den Quellentyp definiert wurden, verwendet werden. Auch kann ein Quellendatentyp nicht in Argumenten oder Operanden verwendet werden, die zur Verwendung eines einzigartigen Datentyps definiert wurden.

Wenn ein einzigartiger Datentyp erstellt ist, kann er in Spaltendefinitionen einer Anweisung `CREATE TABLE` verwendet werden:

```
CREATE TABLE EMPLOYEE
  (EMPNO    CHAR(6)      NOT NULL,
   FIRSTNME VARCHAR(12)  NOT NULL,
   LASTNAME VARCHAR(15)  NOT NULL,
   WORKDEPT CHAR(3),
   PHONENO  CHAR(4),
   PHOTO    BLOB(10M)   NOT NULL,
   EDLEVEL  T_EDUC)
IN RESOURCE
```

Das Erstellen des einzigartigen Datentyps generiert auch Unterstützung für das Umsetzen zwischen dem einzigartigen Datentyp und dem Quellendatentyp. Daher kann ein Wert des Typs `T_EDUC` in einen `SMALLINT`-Wert und ein `SMALLINT`-Wert in einen `T_EDUC`-Wert umgesetzt werden.

Vollständige Syntaxinformationen zur Anweisung `CREATE DISTINCT TYPE` finden Sie im Handbuch *SQL Reference*. Informationen zum Erstellen und Verwenden eines einzigartigen Datentyps finden Sie im Handbuch *Application Development Guide*.

Mit Hilfe von Umsetzungen können Sie benutzerdefinierte Datentypen in Basisdatentypen und Basisdatentypen in benutzerdefinierte Datentypen umsetzen. Eine Umsetzungsfunktion kann mit der Anweisung CREATE TRANSFORM erstellt werden.

Unterstützung für Umsetzungen kann auch durch die Anweisung CREATE METHOD sowie durch Erweiterungen der Anweisung CREATE FUNCTION bereitgestellt werden. Einzelheiten zur dieser Unterstützung finden Sie im Handbuch *SQL Reference*.

Erstellen eines benutzerdefinierten strukturierten Typs

Ein strukturierter Typ ist ein benutzerdefinierter Typ, der ein oder mehrere Attribute enthält, die jeweils über einen eigenen Namen und Datentyp verfügen. Ein strukturierter Typ kann als Typ für eine Tabelle dienen, in der jede Tabellenspalte ihren Namen und Datentyp aus einem der Attribute des strukturierten Typs ableitet. Umfassende Informationen zu strukturierten Typen finden Sie im Handbuch *Application Development Guide*.

Erstellen einer Typenzuordnung

In Systemen zusammenschlossener Datenbanken können Sie mit Hilfe einer Typenzuordnung spezifische Datentypen von Datenquellentabellen und -sichten den einzigartigen DB2-Datentypen zuordnen. Eine Typenzuordnung kann für eine einzelne Datenquelle oder einen Bereich (Typ, Version) von Datenquellen gelten.

Standardzuordnungen für Datentypen werden für integrierte Datenquellentypen und integrierte DB2-Typen bereitgestellt. Neue (von Ihnen erstellte) Datentypenzuordnungen werden in der Sicht SYSCAT.TYPEMAPPINGS aufgelistet.

Typenzuordnungen können Sie mit der Anweisung CREATE TYPE MAPPING erstellen. Sie müssen über die Berechtigung SYSADM oder DBADM für die zusammenschlossene Datenbank verfügen, um diese Anweisung verwenden zu können.

Ein Beispiel für eine Typenzuordnungsanweisung:

```
CREATE TYPE MAPPING MY_ORACLE_DEC FROM SYSIBM.DECIMAL(10,2)
TO SERVER ORACLE1 TYPE NUMBER([10..38],2)
```

Für LOB, LONG VARCHAR/VARGRAPHIC, DATALINK sowie für strukturierte und einzigartige Typen können Sie keine Typenzuordnung erstellen.

Einzelheiten zum Verwenden und Erstellen von Typenzuordnungen finden Sie im Handbuch *Application Development Guide*. Einzelheiten zur Syntax von CREATE TYPE MAPPING finden Sie im Handbuch *SQL Reference*.

Erstellen einer Sicht

Sichten werden von einer oder mehreren Basistabellen, Kurznamen oder Sichten abgeleitet und können beim Abrufen von Daten frei austauschbar mit Basistabellen verwendet werden. Wenn Änderungen an den in einer Sicht gezeigten Daten vorgenommen werden, werden die Daten in der eigentlichen Tabelle geändert.

Eine Sicht kann zur Einschränkung des Zugriffs auf sensible Daten verwendet werden, während auf andere Daten ein allgemeinerer Zugriff zugelassen wird.

Beim Einfügen in eine Sicht, in der eine SELECT-Liste der Sichtdefinition direkt oder indirekt den Namen einer Identitätsspalte einer Basistabelle umfasst, gelten dieselben Regeln wie für das direkte Verweisen der INSERT-Anweisung auf die Identitätsspalte der Basistabelle. Weitere Informationen zur Anweisung INSERT finden Sie im Handbuch *SQL Reference*.

Über die bereits beschriebene Verwendung hinaus kann eine Sicht auch folgenden Zwecken dienen:

- Ändern einer Tabelle ohne Auswirkungen auf Anwendungsprogramme. Dies kann durch das Erstellen einer Sicht auf Grundlage einer zugrundeliegenden Tabelle geschehen. Anwendungen, die die zugrundeliegende Tabelle verwenden, werden durch die Erstellung der neuen Sicht nicht beeinflusst. Neue Anwendungen können die erstellte Sicht für andere Zwecke verwenden als jene Anwendungen, die die zugrundeliegende Tabelle verwenden.
- Summieren der Werte in einer Spalte, Auswählen der größten Werte oder der Durchschnittswerte
- Bereitstellen des Zugriffs auf Informationen an einer oder mehreren Datenquellen. Sie können innerhalb der Anweisung CREATE VIEW auf Kurznamen verweisen sowie Sichten für mehrere Positionen bzw. globale Sichten (die Sicht könnte Informationen aus mehreren Datenquellen auf verschiedenen Systemen verknüpfen) erstellen.

Wenn Sie eine Sicht erstellen, die mit der Standardsyntax von CREATE VIEW auf Kurznamen verweist, erscheint eine Warnung, die darauf hinweist, dass die Berechtigungs-ID der Benutzer dieser Sicht für den Zugriff auf die zugrundeliegenden Objekte an den Datenquellen verwendet werden und nicht die Berechtigungs-ID des Erstellers dieser Sicht. Diese Warnung können Sie mit dem Schlüsselwort FEDERATED unterdrücken.

Eine Alternative zur Erstellung einer Sicht ist die Verwendung eines verschachtelten oder allgemeinen Tabellenausdrucks, um das Durchsuchen von Katalogen zu reduzieren und die Leistung zu erhöhen. Weitere Informationen zu allgemeinen Tabellenausdrücken finden Sie im Handbuch *SQL Reference*.

Gehen Sie wie folgt vor, um mit der Steuerzentrale eine Sicht zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Sichten** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Ordner **Sichten** und wählen Sie dann im Kontextmenü **Erstellen** aus.
3. Geben Sie die erforderlichen Informationen ein und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Sicht zu erstellen:

```
CREATE VIEW <name> (<spalte>, <spalte>, <spalte>)  
  SELECT <spaltennamen> FROM <tabellennamen>  
  WITH CHECK OPTION
```

Die Tabelle EMPLOYEE kann beispielsweise Gehaltsdaten enthalten, die nicht unbedingt jeder Person zugänglich sein sollten. Die Telefonnummer des Mitarbeiters sollte dagegen allgemein verfügbar sein. In diesem Fall kann eine Sicht erstellt werden, die nur aus den Spalten für den Namen (LASTNAME) und für die Telefonnummer (PHONENO) besteht. Der Zugriff auf die Sicht könnte für die Berechtigung PUBLIC (allgemeiner Zugriff) erteilt werden, während der Zugriff auf die gesamte Tabelle EMPLOYEE auf die Personen beschränkt werden könnte, die die Berechtigung haben, Gehaltsdaten einzusehen. Informationen zu Sichten mit *Lesezugriff* (Read-only Views) finden Sie im Handbuch *SQL Reference*.

Mit Hilfe einer Sicht können Sie eine Untermenge der Daten einer Tabelle für ein Anwendungsprogramm verfügbar machen und die Daten auf Gültigkeit prüfen, die eingefügt oder aktualisiert werden sollen. Eine Sicht kann Spaltennamen enthalten, die sich von den Namen der entsprechenden Spalten der Originaltabellen unterscheiden.

Durch die Verwendung von Sichten gestalten sich die Möglichkeiten der Anwendungsprogramme und Endbenutzerabfragen, auf Tabellendaten zuzugreifen, wesentlich flexibler.

Mit der folgenden SQL-Anweisung wird eine Sicht von der Tabelle EMPLOYEE erstellt, die alle Mitarbeiter der Abteilung A00 mit der Personalnummer und der Telefonnummer auflistet:

```
CREATE VIEW EMP_VIEW (DA00NAME, DA00NUM, PHONENO)  
  AS SELECT LASTNAME, EMPNO, PHONENO FROM EMPLOYEE  
  WHERE WORKDEPT = 'A00'  
  WITH CHECK OPTION
```

Die erste Zeile dieser Anweisung benennt die Sicht und definiert ihre Spalten. Der Name EMP_VIEW muss innerhalb seines Schemas in SYSCAT.TABLES eindeutig sein. Der Name der Sicht erscheint als Tabellename, obwohl die Sicht keine Daten enthält. Die Sicht wird mit den Spalten DA00NAME, DA00NUM und PHONENO definiert, die den Spalten LASTNAME, EMPNO und PHONENO der Tabelle EMPLOYEE entsprechen. Die Spaltennamen werden nacheinander in der aufgeführten Reihenfolge jeweils den Spalten der mit der Anweisung SELECT angegebenen SELECT-Liste zugeordnet. Wenn Spaltennamen nicht angegeben werden, verwendet die Sicht dieselben Namen wie die Spalten der Ergebnistabelle der Anweisung SELECT.

Die zweite Zeile enthält eine Anweisung SELECT, die beschreibt, welche Werte aus der Datenbank ausgewählt werden sollen. Sie kann die Klauseln ALL, DISTINCT, FROM, WHERE, GROUP BY und HAVING enthalten. Der Name bzw. die Namen der Datenobjekte, aus denen die Spalten für die Sicht auszuwählen sind, müssen nach der Klausel FROM angegeben werden.

Die Klausel WITH CHECK OPTION gibt an, dass jede in der Sicht aktualisierte oder eingefügte Zeile an der Sichtdefinition überprüft und zurückgewiesen werden muss, wenn sie der Definition nicht entspricht. Dadurch wird die Datenintegrität erhöht, aber auch zusätzlicher Verarbeitungsaufwand verursacht. Wenn diese Klausel nicht angegeben wird, werden Einfügungen und Aktualisierungen nicht an der Sichtdefinition überprüft.

Mit der folgenden SQL-Anweisung wird dieselbe Sicht von der Tabelle EMPLOYEE erstellt, allerdings mit der Klausel SELECT AS:

```
CREATE VIEW EMP_VIEW
  SELECT LASTNAME AS DA00NAME,
         EMPNO AS DA00NUM,
         PHONENO
  FROM EMPLOYEE
  WHERE WORKDEPT = 'A00'
  WITH CHECK OPTION
```

Sie können eine Sicht erstellen, die in ihrer Definition eine benutzerdefinierte Funktion verwendet. Wenn diese Sicht jedoch aktualisiert werden soll, damit sie die neusten Funktionen enthält, muss sie gelöscht und anschließend neu erstellt werden. Wenn eine Sicht von einer benutzerdefinierten Funktion abhängig ist, kann diese Funktion nicht gelöscht werden.

Mit der folgenden SQL-Anweisung wird eine Sicht mit einer Funktion in ihrer Definition erstellt:

```
CREATE VIEW EMPLOYEE_PENSION (NAME, PENSION)
  AS SELECT NAME, PENSION(HIREDATE, BIRTHDATE, SALARY, BONUS)
  FROM EMPLOYEE
```

Die benutzerdefinierte Funktion PENSION berechnet die aktuellen Pensionsansprüche eines Mitarbeiters mit einer Formel, die mit den Werten des Einstellungsdatums (HIREDATE), des Geburtsdatums (BIRTHDATE), des Gehalts (SALARY) und der Sondervergütungen (BONUS) arbeitet.

Erstellen einer typisierten Sicht

Sie können mit Hilfe der Anweisung CREATE VIEW eine typisierte Sicht erstellen. Umfassende Informationen zu typisierten Sichten finden Sie im Handbuch *Application Development Guide*.

Erstellen einer Übersichtstabelle

Eine *Übersichtstabelle* ist eine Tabelle, deren Definition auf einem Abfrageergebnis basiert. So gesehen enthält die Übersichtstabelle in der Regel zuvor berechnete Ergebnisse von Daten, die in der bzw. den Tabelle(n) vorhanden sind, auf denen die Definition der Übersichtstabelle basiert. Wenn der SQL-Compiler feststellt, dass eine Abfrage auf eine Übersichtstabelle effektiver ausgeführt werden kann als auf die Basistabelle, wird die Abfrage auf die Übersichtstabelle ausgeführt, und das Ergebnis wird schneller ausgegeben als sonst.

Die Erstellung einer Übersichtstabelle mit der Replikationsoption kann verwendet werden, um Tabellen auf allen Knoten in einer partitionierten Datenbankumgebung zu replizieren. Sie werden als „replizierte Übersichtstabellen“ bezeichnet. Weitere Informationen zu solchen Tabellen finden Sie im Handbuch *Systemverwaltung: Konzept*.

Anmerkung: Übersichtstabellen werden nicht mit statischem SQL oder mit Kurznamen verwendet.

In der Regel wird eine Übersichtstabelle oder replizierte Übersichtstabelle zum Optimieren einer Abfrage verwendet, wenn die Isolationsstufe der Übersichtstabelle oder replizierten Übersichtstabelle größer oder gleich der Isolationsstufe der Abfrage ist. Wird eine Abfrage beispielsweise unter der Isolationsstufe für Cursorstabilität (CS) ausgeführt, werden für die Optimierung nur Übersichtstabellen und replizierte Übersichtstabellen verwendet, die unter CS oder einer höheren Isolationsstufe definiert wurden.

Sie können eine Übersichtstabelle mit Hilfe der Anweisung CREATE SUMMARY TABLE unter Angabe der Klausel AS *voll-select* und der Option IMMEDIATE oder REFRESH DEFERRED erstellen.

Sie haben die Möglichkeit, die Namen der Spalten in der Übersichtstabelle eindeutig zu identifizieren. Die Liste der Spaltennamen muss für jede Spalte der Ergebnistabelle in der vollständigen Auswahl einen Namen enthalten. Eine Liste der Spaltennamen muss angegeben werden, wenn die Ergebnistabelle für die Gesamtauswahl über doppelte Spaltennamen oder Spalten ohne Benennung verfügt. Eine nicht benannte Spalte wird von einer Konstanten, einer Funktion, einem Ausdruck oder einer definierten Operation abgeleitet, die nicht mit Hilfe der Klausel AS der SELECT-Liste benannt wurde. Wurde keine Liste mit Spaltennamen angegeben, übernehmen die Spalten der Tabelle die Namen der Spalten der Ergebnismenge der Gesamtauswahl.

Beim Erstellen der Übersichtstabelle können Sie angeben, ob die Tabelle bei Änderungen an der Basistabelle automatisch aktualisiert wird, oder ob sie mit Hilfe der Anweisung REFRESH TABLE aktualisiert wird. Geben Sie das Schlüsselwort REFRESH IMMEDIATE an, damit die Übersichtstabelle automatisch aktualisiert wird, wenn Änderungen an der Basistabelle bzw. den Basistabellen vorgenommen werden. Die sofortige Aktualisierung ist in folgenden Fällen sinnvoll:

- Bei lange dauernden Abfragen einer Basistabelle
- Bei einer oder mehreren Basistabellen, die nur selten geändert werden
- Bei Aktualisierungen mit geringem Aufwand

In diesem Fall kann die Übersichtstabelle vorausberechnete Ergebnisse zur Verfügung stellen. Wenn die Aktualisierung der Übersichtstabelle verzögert erfolgen soll, geben Sie das Schlüsselwort REFRESH DEFERRED an. Mit REFRESH DEFERRED angegebene Übersichtstabellen spiegeln Änderungen an den zugrundeliegenden Basistabellen **nicht** wider. Sie sollten Übersichtstabellen verwenden, wo dies keine Voraussetzung ist. Wenn Sie zum Beispiel DSS-Abfragen ausführen, würden Sie mit Hilfe der Übersichtstabelle übernommene Daten abrufen.

Eine mit REFRESH DEFERRED definierte Übersichtstabelle kann anstelle einer Abfrage verwendet werden, wenn Folgendes zutrifft:

- Sie erfüllt die Einschränkungen für eine Übersichtstabelle mit Gesamtauswahl oder sofortiger Neuanzeige außer den folgenden:
 - Die SELECT-Liste braucht COUNT(*) oder COUNT_BIG(*) nicht zu enthalten.
 - Die SELECT-Liste kann die Spaltenfunktion MAX und MIN enthalten.
 - Eine Klausel HAVING ist zulässig.

Das SQL-Sonderregister CURRENT REFRESH AGE SQL ist auf ANY eingestellt oder hat den Wert 9999999999999999. Die Sammlung von Neunen ist der zulässige Maximalwert in diesem Sonderregister, der ein Zeitmarkendifferenzwert mit dem Datentyp DECIMAL(20,6) ist.

Anmerkung: Mit REFRESH DEFERRED definierte Übersichtstabellen werden nicht zum Optimieren von statischem SQL verwendet.

Mit dem Sonderregister CURRENT REFRESH AGE können Sie die Zeitspanne angeben, die für die Übersichtstabelle mit verzögerter Neuanzeige für eine dynamische Abfrage verwendet werden kann, bevor sie aktualisiert werden muss. Sie können den Wert für das Sonderregister CURRENT REFRESH AGE mit Hilfe der Anweisung SET CURRENT REFRESH AGE einstellen. Weitere Informationen zum Sonderregister CURRENT REFRESH AGE und zur Anweisung SET CURRENT REFRESH AGE finden Sie im Handbuch *SQL Reference*.

Mit REFRESH IMMEDIATE definierte Übersichtstabellen gelten für statische und dynamische Abfragen und kommen ohne das Sonderregister CURRENT REFRESH AGE aus.

Anmerkung: Sie sollten beim Einstellen des Sonderregisters CURRENT REFRESH AGE auf einen anderen Wert als Null Vorsicht walten lassen. Wenn Sie zulassen, dass eine Übersichtstabelle, die die Werte der zugrundeliegenden Basistabelle eventuell nicht repräsentiert, zur Optimierung der Abfrageverarbeitung verwendet wird, gibt das Abfrageergebnis die Daten in der zugrundeliegenden Tabelle eventuell *nicht* korrekt wieder. Dies ist eventuell akzeptabel, wenn Sie wissen, dass die zugrundeliegenden Daten nicht geändert wurden oder wenn Sie die Fehlerquote in den Ergebnissen auf Grundlage Ihrer Kenntnis der Daten akzeptieren.

Da die Quelldaten durch bestimmte Vorgänge geändert werden, enthält eine Übersichtstabelle im Laufe der Zeit veraltete (inkorrekte) Daten. Sie müssen zur Aktualisierung die Anweisung REFRESH TABLE verwenden. Weitere Informationen finden Sie im Handbuch *SQL Reference*.

Wenn Sie eine neue Basistabelle erstellen wollen, die auf einer ungültigen Gesamtauswahl (*voll-select*) fußt, geben Sie das Schlüsselwort DEFINITION ONLY beim Erstellen der Tabelle an. Wenn die Operation zur Tabellenerstellung beendet ist, wird die neue Tabelle nicht als Übersichtstabelle behandelt, sondern als Basistabelle. Sie können zum Beispiel die in LOAD und SET INTEGRITY verwendeten Ausnahmetabellen wie folgt erstellen:

```
CREATE TABLE XT AS
  (SELECT T.*, CURRENT_TIMESTAMP AS TIMEAMP,CLOB(",32K)
  AS MSG FROM T) DEFINITION ONLY
```

Es folgt eine Auflistung einiger der wichtigsten Einschränkungen für Übersichtstabellen:

1. Sie können eine Übersichtstabelle nicht ändern.
2. Sie können die Länge einer Spalte für eine Basistabelle nicht ändern, wenn diese Tabelle eine Übersichtstabelle aufweist.
3. In eine Übersichtstabelle können keine Daten importiert werden.
4. Für eine Übersichtstabelle kann kein eindeutiger Index erstellt werden.
5. Es kann keine, auf dem Ergebnis einer Abfrage basierende Übersichtstabelle erstellt werden, die auf einen oder mehrere Kurznamen verweist.

Eine vollständige Abhandlung der Einschränkungen für Übersichtstabellen finden Sie im Handbuch *SQL Reference*.

Erstellen eines Aliasnamens

Ein Aliasname ist eine indirekte Methode, auf eine Tabelle, einen Kurznamen oder eine Sicht zu verweisen, so dass eine SQL-Anweisung nicht vom qualifizierten Namen dieser Tabelle oder Sicht abhängig sein muss. Es muss lediglich die Definition des Aliasnamens geändert werden, wenn die Tabelle oder Sicht umbenannt wird. Ein Aliasname kann für andere Aliasnamen erstellt werden. Ein Aliasname kann in einer Sicht- oder Auslöserdefinition und in jeder SQL-Anweisung außer in Definitionen von Prüfungen auf Integritätsbedingungen in Tabellen (Table Check Constraints) verwendet werden, in denen ein vorhandener Tabellen- oder Sichtname angegeben werden kann.

Ein Aliasname kann überall dort verwendet werden, wo ein vorhandener Tabellenname verwendet werden kann, und kann auf einen anderen Aliasnamen verweisen, sofern keine Schleifen oder sich wiederholende Verweise in der Kette der Aliasnamen auftreten.

Als Aliasname kann kein bereits vorhandener Tabellen-, Sicht- oder Aliasname angegeben werden, und der Name darf sich nur auf eine Tabelle in derselben Datenbank beziehen. Der Name einer Tabelle oder Sicht, der in einer Anweisung CREATE TABLE oder CREATE VIEW verwendet wird, darf nicht mit einem Aliasnamen im selben Schema übereinstimmen.

Sie benötigen keine besondere Berechtigung zur Erstellung eines Aliasnamens, sofern der Aliasname nicht in einem anderen Schema ist als dem, das Ihrer aktuellen Berechtigungs-ID eigen ist. Ist dies der Fall, benötigen Sie die Berechtigung DBADM.

Ein Aliasname kann für eine Tabelle, eine Sicht oder einen Aliasnamen definiert werden, die bzw. der zum Zeitpunkt der Definition noch nicht existiert. Wenn eine SQL-Anweisung, die diesen Aliasnamen enthält, kompiliert wird, muss die Tabelle, Sicht oder der Bezugsaliasname vorhanden sein.

Wenn ein Aliasname oder das Objekt, auf das ein Aliasname verweist, gelöscht wird, werden alle Pakete, die von dem Aliasnamen abhängig sind, als unbrauchbar markiert.

Gehen Sie wie folgt vor, um mit der Steuerzentrale einen Aliasnamen zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Aliasnamen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Ordner **Aliasnamen** und wählen Sie dann im Kontextmenü **Erstellen** aus.
3. Geben Sie die erforderlichen Informationen ein und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um einen Aliasnamen zu erstellen:

```
CREATE ALIAS <aliasname> FOR <tabellenname>
```

Der Aliasname wird bei der Kompilierung der Anweisung durch den entsprechenden Tabellen- oder Sichtnamen ersetzt. Wenn der Aliasname oder die Aliasnamenverkettung nicht in einen Tabellen- oder Sichtnamen aufgelöst werden kann, wird ein Fehler zurückgegeben.

Wenn zum Beispiel WORKERS ein Aliasname für die Tabelle EMPLOYEE ist, geschieht bei der Kompilierung Folgendes:

```
SELECT * FROM WORKERS
```

wird effektiv zu

```
SELECT * FROM EMPLOYEE
```

Mit der folgenden SQL-Anweisung wird ein Aliasname WORKERS für die Tabelle EMPLOYEE erstellt:

```
CREATE ALIAS WORKERS FOR EMPLOYEE
```

Anmerkung: DB2 für MVS/ESA implementiert zwei verschiedene Konzepte von Aliasnamen: ALIAS und SYNONYM. Diese Konzepte weichen auf folgende Weise von DB2 Universal Database ab:

- Für ALIASnamen in DB2 für MVS/ESA gilt:
 - Sie machen eine spezielle Berechtigung oder ein bestimmtes Zugriffsrecht für ihre Ersteller erforderlich.
 - Sie können nicht auf andere Aliasnamen verweisen.
- Für SYNONYMnamen in DB2 für MVS/ESA gilt:
 - Sie können nur von ihrem Ersteller verwendet werden.
 - Sie sind nie qualifiziert.
 - Sie werden gelöscht, wenn die Bezugstabelle gelöscht wird.
 - Sie benutzen keinen Namensbereich mit Tabellen oder Sichten gemeinsam.

Erstellen einer Oberfläche

In einer zusammengeschlossenen Datenbank registriert die Anweisung `CREATE WRAPPER` eine Oberfläche. Die Anweisung definiert, auf welche Weise ein Server einer zusammengeschlossenen Datenbank mit einer bestimmten Datenquellenkategorie interagieren kann.

Für spezifische Datenquellenarten, Versionen, Übertragungsprotokolle und Betriebssysteme müssen spezifische Bibliotheken verwendet werden. Auf AS/400- und DB2 für OS/390-Datenquellen wird zum Beispiel mit der Bibliothek `libdrda.dll` für zusammengeschlossene Datenbanken zugegriffen, die unter Windows NT-Betriebssystemen mit APPC-Kommunikation arbeiten.

Sie müssen über die Berechtigung `SYSADM` oder `DBADM` für die zusammengeschlossene Datenbank verfügen, um die Anweisung `CREATE WRAPPER` verwenden zu können.

Durch die Erstellung einer Oberfläche über die Steuerzentrale oder den Befehlszeilenprozessor wird diese in der zusammengeschlossenen Datenbank registriert.

Gehen Sie wie folgt vor, um mit der Steuerzentrale eine Oberfläche zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Objekte zusammengeschlossener Datenbanken** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Ordner **Objekte zusammengeschlossener Datenbanken** und wählen Sie dann im Kontextmenü **Oberfläche erstellen** aus.
3. Geben Sie die erforderlichen Informationen ein und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Oberfläche zu erstellen:

```
CREATE WRAPPER <oberflächenname> LIBRARY '<bibliotheksname>'
```

Die folgende SQL-Anweisung registriert die Oberfläche `ORACLE8` unter einem Windows NT-Betriebssystem:

```
CREATE WRAPPER ORACLE8 LIBRARY 'libnet8.dll'
```

Detaillierte Informationen zur Verwendung der Anweisung `CREATE WRAPPER` finden Sie im Handbuch *SQL Reference*.

Erstellen eines Servers

Erstellen Sie in einer zusammengeschlossenen Datenbank Server, um Datenquellen in DB2 zu definieren und ihre Kenndaten (zum Beispiel Name, Oberfläche, Art, Version, Position und Optionen) zu beschreiben. Diese Informationen werden dazu verwendet, spezifischen Datenverwaltungssystemen Kurznamen zuzuordnen und Informationen für das DB2-Optimierungsprogramm bereitzustellen. Die Server-Informationen befinden sich in den Katalogsichten SYSCAT.SERVERS und SYSCAT.SERVEROPTIONS.

Anmerkung: In diesem Abschnitt steht der Terminus "Server" für Datenquellen und nicht für DRDA-Server oder DB2-Server. Für den Zugriff auf andere Datenquellen (z. B. Oracle) ist DB2 Connect erforderlich.

Sie können ein Server-Objekt nur dann erstellen, wenn zuvor eine Oberfläche erstellt wurde.

Sie müssen über die Berechtigung SYSADM oder DBADM für die zusammengeschlossene Datenbank verfügen, um diese Anweisung verwenden zu können.

Sie können Benutzerzuordnungen erstellen, um die Unterschiede in der Authentifizierungsverarbeitung zwischen DB2- und Datenquellenservern auszugleichen. Benutzerzuordnungen werden im Abschnitt „Benutzerzuordnungen“ auf Seite 280 ausführlich behandelt.

Beim Löschen eines Servers werden auch alle von diesem Server abhängigen Objekte (Benutzerzuordnungen, Kurznamen, Funktionszuordnungen, Typenzuordnungen, Pläne etc.) gelöscht.

Geben Sie Serveroptionen beim Erstellen eines Servers an. Diese Optionen stellen erforderliche Einzelangaben über den Server zur Verfügung (zum Beispiel den Knotennamen). Serveroptionen können auch spezifische Leistungs- und Sicherheitswerte festlegen.

Sie können Server in der Steuerzentrale oder über den Befehlszeilenprozessor erstellen.

Gehen Sie wie folgt vor, um mit der Steuerzentrale einen Server zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Server** unter dem Ordner **Objekte zusammenschlossener Datenbanken** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf den Ordner **Server**, und wählen Sie dann im Kontextmenü **Server erstellen** aus.
3. Geben Sie die erforderlichen Informationen ein, und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um einen Server zu erstellen:

```
CREATE SERVER <server_name> TYPE <server_art>
  VERSION <server_version> WRAPPER <oberflächenname>
  OPTIONS (<server_optionsname> <zeichenfolgekonstante>)
```

Die folgende SQL-Anweisung erstellt den Oracle-Server ORA8:

```
CREATE SERVER ORA8 TYPE ORACLE VERSION 8 WRAPPER ORACLE8 OPTIONS
(NODE 'ONODE')
```

Die folgende SQL-Anweisung erstellt den DB2-Server DB2TEST:

```
CREATE SERVER DB2TEST TYPE DB2 VERSION 6.1 WRAPPER DB2UDB OPTIONS
(NODE 'DB2TEST', DBNAME 'TEST1')
```

Die Definition für NODE variiert in SQL-Anweisungen SERVER je nach Datenquelle. Wenn die Datenquelle ein DB2-Datenbankverwaltungssystem ist, verweist der Wert auf ein DB2-Exemplar, das eine oder mehrere Datenbanken enthält. Beachten Sie im vorigen Beispiel, dass die Option DBNAME den Datenbanknamen angibt. Wenn die Datenquelle ein DB2 für OS/390-Datenbankverwaltungssystem ist, verweist der Wert auf den Stationsnamen (LOCATION) des DB2 für OS/390-Systems. Wenn die Datenquelle ein Oracle-Datenbankverwaltungssystem ist, wird die Option DBNAME nicht benötigt, weil ein Oracle-Exemplar nur eine Datenbank enthält.

Weitere Einzelheiten zur Syntax der Anweisung CREATE SERVER finden Sie im Handbuch *SQL Reference*. Weitere Einzelheiten zur Verwendung der Anweisung CREATE SERVER finden Sie im Handbuch *Installation und Konfiguration Ergänzung*.

Vereinfachen der Datenquellendefinitionen und der Authentifizierungsverarbeitung durch Serveroptionen

Sie können für Variablen, die auch als *Serveroptionen* bezeichnet werden, Werte festlegen, die sich darauf auswirken, wie ein Server mit zusammengeschlossenen Datenbanken auf Datenquellen zugreift. Dieser Abschnitt behandelt folgenden Themen:

- Verwendungszwecke der Serveroptionen
- SQL-Anweisungen zum Angabens von Serveroptionen
- Die Serveroptionen und ihre Einstellungen

Verwendungszwecke der Serveroptionen

Im allgemeinen dienen Serveroptionen den folgenden Verwendungszwecken:

- Bereitstellen und Aktualisieren von Informationen zu Datenquellen. Eine Server-Referenz enthält Basisinformationen zu einer Datenquelle (zum Beispiel den Namen der Datenquelle) und Informationen, die sich im Lauf der Zeit ändern können. Manche der veränderlichen Informationen werden durch Werte übermittelt, die Serveroptionen zugeordnet sind. Der Wert der Option `cpu_ratio` gibt zum Beispiel an, ob die CPU der Datenquelle schneller oder langsamer ist als die CPU des DB2-Systems. Wenn der Prozessor des DB2-Systems aufgerüstet wird, müsste sich dieser Wert ändern.
- Vereinfachen der Authentifizierung. Sie können durch Festlegen bestimmter Serveroptionen sicherstellen, dass Benutzer-IDs und Kennwörter in der richtigen Schreibung (Groß-/Kleinschreibung) an die Datenquelle übermittelt werden. Zum Beispiel können Sie die Option `fold_id` so festlegen, dass der Server einer zusammengeschlossenen Datenbank den Benutzernamen in die für die Datenquelle erforderliche Schreibweise (Groß- oder Kleinschreibung) umwandelt, bevor er eine Benutzer-ID an die Datenquelle übermittelt. Wenn Sie die Benutzer-ID bereits auf dem Server einer zusammengeschlossenen Datenbank in der erforderlichen Schreibung definieren, können Sie durch Festlegen der Option `fold_id` verhindern, dass der Server beim Ändern der Schreibung unnötig Systemressourcen beansprucht.
- Optimieren von Abfragen. Einige Serveroptionen und die dazugehörigen Werte erleichtern die Optimierung. Beispiel: In der Anweisung `CREATE SERVER` können Sie bestimmte Leistungsstatistiken als Optionswerte angeben. Sie können die Option `cpu_ratio` auf einen Wert setzen, der die relativen Geschwindigkeiten der CPUs der Datenquelle und des Servers mit zusammengeschlossenen Datenbanken angibt. Darüber hinaus können Sie die Option `io_ratio` auf einen Wert setzen, der die relativen Übertragungsraten der E/A-Einheiten der Datenquelle und des Servers mit zusammengeschlossenen Datenbanken angibt.

Bei der Ausführung der Anweisung `CREATE SERVER` werden diese Daten der Katalogsicht `SYSCAT.SERVEROPTIONS` hinzugefügt, und das Optimierungsprogramm bezieht sie in die Entwicklung eines Zugriffsplans für die Datenquelle mit ein. Falls sich ein Statistikwert ändert (wie es beispielsweise bei einer Aufrüstung der CPU der Datenquelle möglich ist), können Sie die Katalogsicht `SYSCAT.SERVEROPTIONS` mit Hilfe der Anweisung `ALTER SERVER` mit dieser Änderung aktualisieren. Das Optimierungsprogramm verwendet dann diese aktualisierten Werte bei der Entwicklung des nächsten Zugriffsplans für die Datenquelle.

SQL für Serveroptionen

In den folgenden drei SQL-Anweisungen können Sie Werte für Serveroptionen zuordnen: `CREATE SERVER`, `ALTER SERVER` und `SET SERVER OPTION`.

Mit der Anweisung `CREATE SERVER` können Sie eine Option auf einen Wert setzen, der auf unbestimmte Zeit für mehrere Verbindungen zu einer Datenquelle bestehen bleibt. Mit dieser Anweisungen können Sie für eine Option einen anderen als den Standardwert festlegen, oder, wenn kein Standardwert für die Option besteht, einen Anfangswert festlegen.

Mit der Anweisung `ALTER SERVER` können Sie für eine Serveroption, die zuvor mit der Anweisung `CREATE SERVER` auf einen Wert gesetzt wurde, einen anderen Wert definieren, der über mehrere Verbindungen hinweg bestehen bleibt.

Verwenden Sie die Anweisung `SET SERVER OPTION` zum temporären Ändern des Wertes einer Serveroption für die Dauer einer Verbindung zur Datenbank. Anweisungen vom Typ `SET SERVER OPTION` müssen an erster Stelle der Arbeitseinheit stehen, die unmittelbar auf die Verbindung zur Datenquelle folgt.

Verwenden Sie zum Beispiel die folgende Anweisung, um für den Oracle-Server `ORASEB1` vorübergehend die Verwendung von Zugriffsplanhinweisen zu aktivieren:

```
SET SERVER OPTION plan_hints TO 'Y' FOR SERVER ORASEB1
```

Serveroptionen und zugehörige Einstellungen

Die nachfolgende Tabelle enthält die Serveroptionen mit den zugehörigen Werten. Soweit nicht anders angegeben, müssen alle Werte für Serveroptionen in einfachen Anführungszeichen angegeben werden.

Tabelle 2. Serveroptionen und zugehörige Einstellungen

Option	Gültige Einstellungen	Standard- einstellung
collating_sequence	<p>Gibt auf der Basis des codierten Zeichensatzes und der länderspezifischen Informationen an, ob die Datenquelle die gleiche Standardsortierfolge wie die zusammengeschlossene Datenbank verwendet. Wenn eine Datenquelle eine Sortierfolge besitzt, die von der Sortierfolge von DB2 abweicht, können die meisten Operationen, die von Sortierfolge von DB2 abhängig sind, nicht fern an einer Datenquelle ausgewertet werden. Ein Beispiel hierfür wäre die Ausführung von Spaltenfunktionen MAX an der Zeichenspalte für Kurznamen an einer Datenquelle mit einer anderen Sortierfolge. Da die Ergebnisse verschieden ausfallen könnten, wenn die Funktion MAX an der fernen Datenquelle ausgewertet würde, führt DB2 die Datenverbundoperation und die Funktion MAX lokal aus.</p> <p>Wenn die Abfrage ein Gleichheitszeichen enthält, ist es möglich, diesen Teil der Abfrage zu verschieben (Pushdown), auch wenn die Sortierfolgen unterschiedlich (Wert 'N') sind. Zum Beispiel könnte das Vergleichselement C1 = 'A' an eine Datenquelle verschoben werden. Natürlich können solche Abfragen nicht verschoben werden, wenn die Sortierfolge an der Datenquelle die Groß-/Kleinschreibung unterscheidet. Wenn eine Datenquelle die Groß-/Kleinschreibung nicht beachtet, sind die Ergebnisse der Vergleichselemente C1= 'A' und C1 = 'a' identisch, was in einer Umgebung mit Beachtung der Groß-/Kleinschreibung (DB2) nicht akzeptabel ist.</p> <p>Administratoren können zusammengeschlossene Datenbanken mit einer bestimmten Sortierfolge erstellen, die der Sortierfolge der Datenquelle entspricht. Diese Lösung kann die Leistung erhöhen, wenn alle Datenquellen die gleiche Sortierfolge besitzen bzw. wenn die meisten oder alle Spaltenfunktionen auf Datenquellen gerichtet sind, die die gleiche Sortierfolge verwenden.</p>	'N'
	'Y' Die Sortierfolge der Datenquelle stimmt mit der Sortierfolge der zusammengeschlossenen Datenbank überein.	
	'N' Die Sortierfolge der Datenquelle stimmt nicht mit der Sortierfolge der zusammengeschlossenen Datenbank überein.	
	'I' Die Sortierfolge der Datenquelle stimmt nicht mit der Sortierfolge der zusammengeschlossenen Datenbank überein und macht keinen Unterschied zwischen Groß- und Kleinschreibung (z. B. werden 'TOLLESON' und 'ToLLESon' als gleichwertig angesehen).	

Tabelle 2. Serveroptionen und zugehörige Einstellungen (Forts.)

Option	Gültige Einstellungen	Standard-einstellung
comm_rate	Gibt die Übertragungsgeschwindigkeit zwischen einem Server einer zusammengeschlossenen Datenbank und den zugeordneten Datenquellen an. Der Wert wird in MB pro Sekunde angegeben.	'2.0'
connectstring	Gibt die zum Herstellen einer Verbindung zu einem OLE-Datenbankbetreiber erforderlichen Initialisierungsmerkmale an. Die vollständige Syntax und Semantik der Verbindungszeichenfolge finden Sie im Abschnitt über die API für Kommunikationsverbindung der Kernkomponenten von OLE DB des Handbuchs <i>Microsoft OLE DB 2.0 Programmer's Reference and Data Access SDK, Microsoft Press, 1998</i> .	Keine
cpu_ratio	Gibt den Geschwindigkeitsunterschied zwischen der CPU einer Datenquelle und der CPU des Servers für eine zusammengeschlossene Datenbank an. Ein Wert von „1.0“ weist auf eine gleiche Übertragungsgeschwindigkeit zwischen der CPU der Datenquelle und der CPU der zusammengeschlossenen Server hin. Ein Wert <1.0 weist auf eine langsamere CPU der Datenquelle hin. Ein Wert >1.0 weist auf eine schnellere CPU der Datenquelle hin.	'1.0'
dbname	Name der Datenbank der Datenquelle, auf die der Server einer zusammengeschlossenen Datenbank zugreifen soll. Diese Angabe ist für Datenquellen der DB2-Familie erforderlich, gilt jedoch nicht für Datenquellen von Oracle.	Keine

Tabelle 2. Serveroptionen und zugehörige Einstellungen (Forts.)

Option	Gültige Einstellungen	Standard- einstellung
fold_id (Siehe Anmerkungen 1 und 4 im Anschluss an diese Tabelle.)	Gilt für Benutzer-IDs, die der Server einer zusammengeschlossenen Datenbank an Datenquellen zur Authentifizierung sendet. Gültige Werte:	Keine
	'U' Der Server einer zusammengeschlossenen Datenbank setzt die Benutzer-ID vor dem Senden an die Datenquelle in Großbuchstaben um. Dies ist eine logische Auswahl für Datenquellen der DB2-Familie und von Oracle (siehe Anmerkung 2 im Anschluss an diese Tabelle).	
	'N' Der Server einer zusammengeschlossenen Datenbank sendet die Benutzer-ID unverändert an die Datenquelle. (Siehe Anmerkung 2 im Anschluss an diese Tabelle.)	
	'L' Der Server einer zusammengeschlossenen Datenbank setzt die Benutzer-ID vor dem Senden an die Datenquelle in Kleinbuchstaben um.	
Wenn keine dieser Einstellungen verwendet wird, versucht der Server einer zusammengeschlossenen Datenbank die Benutzer-ID in Großbuchstaben an die Datenquelle zu senden. Wird die Benutzer-ID nicht akzeptiert, versucht der Server, sie in Kleinbuchstaben zu senden.		
fold_pw (Siehe Anmerkungen 1, 3 und 4 im Anschluss an diese Tabelle.)	Gilt für Kennwörter, die der Server einer zusammengeschlossenen Datenbank an Datenquellen zur Authentifizierung sendet. Gültige Werte:	Keine
	'U' Der Server einer zusammengeschlossenen Datenbank setzt das Kennwort vor dem Senden an die Datenquelle in Großbuchstaben um. Dies ist eine logische Auswahl für Datenquellen der DB2-Familie und von Oracle.	
	'N' Der Server einer zusammengeschlossenen Datenbank sendet das Kennwort unverändert an die Datenquelle.	
	'L' Der Server einer zusammengeschlossenen Datenbank setzt das Kennwort vor dem Senden an die Datenquelle in Kleinbuchstaben um.	
Wenn keine dieser Einstellungen verwendet wird, versucht der Server einer zusammengeschlossenen Datenbank das Kennwort in Großbuchstaben an die Datenquelle zu senden. Wird das Kennwort nicht akzeptiert, versucht der Server, es in Kleinbuchstaben zu senden.		

Tabelle 2. Serveroptionen und zugehörige Einstellungen (Forts.)

Option	Gültige Einstellungen	Standard-einstellung
io_ratio	Gibt den Geschwindigkeitsunterschied zwischen dem E/A-System einer Datenquelle und dem E/A-System des Servers einer zusammengeschlossenen Datenbank an. Ein Wert von „1.0“ weist auf eine gleiche Übertragungsgeschwindigkeit zwischen der CPU der Datenquelle und der CPU der zusammengeschlossenen Server hin. Ein Wert <1.0 weist auf eine langsamere CPU der Datenquelle hin. Ein Wert >1.0 weist auf eine schnellere CPU der Datenquelle hin.	'1.0'
Knoten	<p>Name, durch den eine Datenquelle als Exemplar des zugehörigen Verwaltungssystems für relationale Datenbanken definiert wird. Diese Angabe ist für alle Datenquellen erforderlich.</p> <p>Für eine Datenquelle der DB2-Familie ist dies der im DB2-Knotenverzeichnis der zusammengeschlossenen Datenbank angegebene Knoten. Dieses Verzeichnis kann mit dem Befehl db2 list node directory angezeigt werden.</p> <p>Für eine Datenquelle von Oracle ist dieser Name der Server-Name, der in der Datei <code>tnsnames.ora</code> von Oracle angegeben ist. Sie können auf der Windows NT-Plattform mit Hilfe der Option zum Anzeigen von Konfigurationsinformationen (View Configuration Information) des Oracle-Programms SQL Net Easy Configuration auf diesen Namen zugreifen.</p>	Keine
password	<p>Gibt an, ob Kennwörter an eine Datenquelle gesendet werden.</p> <p>'Y' Kennwörter werden immer an die Datenquelle gesendet und ausgewertet. Dies ist der Standardwert.</p> <p>'N' Kennwörter werden nicht an die Datenquelle gesendet (ohne Rücksicht auf mögliche Benutzerzuordnungen) und werden nicht ausgewertet.</p> <p>'ENCRYPTION' Kennwörter werden immer in verschlüsselter Form an die Datenquelle gesendet und ausgewertet. Dieser Wert ist nur für Datenquellen der DB2-Familie gültig, die verschlüsselte Kennwörter unterstützen.</p>	'Y'

Tabelle 2. Serveroptionen und zugehörige Einstellungen (Forts.)

Option	Gültige Einstellungen	Standard-einstellung
plan_hints	<p>Gibt an, ob Planhinweise (<i>plan hints</i>) aktiviert werden sollen. Planhinweise sind Anweisungsfragmente, die zusätzliche Informationen für Optimierungsprogramme von Datenquellen bereitstellen. Diese Informationen können bei einigen Abfragearten die Abfrageleistung verbessern. Die Planhinweise können das Optimierungsprogramm der Datenquelle bei der Entscheidung unterstützen, ob ein Index, und wenn ja, welcher, zu verwenden ist oder nach welcher Reihenfolge bei der Verknüpfung von Tabellen vorzugehen ist.</p> <p>'Y' Planhinweise sollen an der Datenquelle aktiviert werden, wenn die Datenquelle Planhinweise unterstützt.</p> <p>'N' Planhinweise sollen an der Datenquelle nicht aktiviert werden.</p>	'N'
pushdown	<p>'Y' DB2 zieht in Betracht, die Datenquelle Operationen auswerten zu lassen.</p> <p>'N' DB2 ruft lediglich Spalten von der fernen Datenquelle ab und lässt die Datenquelle keine anderen Operationen, wie zum Beispiel Verknüpfungen, auswerten.</p>	'Y'
varchar_no_trailing_blanks	<p>Gibt an, ob diese Datenquelle eine VARCHAR-Vergleichssemantik für nicht mit Leerzeichen aufgefüllte Zeichenfolgen verwendet. Für Zeichenfolgen variabler Länge, die keine folgenden Leerzeichen enthalten, liefert die Vergleichssemantik für nicht mit Leerzeichen aufgefüllte Zeichenfolgen einiger DBMSs die gleichen Ergebnisse wie die Vergleichssemantik von DB2. Wenn Sie sicher sind, dass alle VARCHAR-Spalten von Tabellen und Sichten an einer Datenquelle keine folgenden Leerzeichen enthalten, können Sie in Betracht ziehen, diese Serveroption auf den Wert 'Y' für eine Datenquelle zu setzen. Diese Option wird häufig für Datenquellen von Oracle verwendet. Stellen Sie sicher, dass Sie alle Objekte, die potenziell Kurznamen haben können, (einschließlich Sichten) berücksichtigen.</p> <p>'Y' Diese Datenquelle verfügt über eine Vergleichssemantik für nicht mit Leerzeichen aufgefüllte Zeichenfolgen, die der Vergleichssemantik von DB2 ähnlich ist.</p> <p>'N' Diese Datenquelle verfügt über keine mit DB2 vergleichbare Vergleichssemantik für nicht mit Leerzeichen aufgefüllte Zeichenfolgen.</p>	'N'

Hinweise zu dieser Tabelle:

1. Dieses Feld wird unabhängig von dem für die Authentifizierung (Authentication) angegebenen Wert angewendet.
2. Da DB2 die Benutzer-IDs in Großbuchstaben speichert, sind die Werte 'N' und 'U' logisch äquivalent zueinander.
3. Die Einstellung für 'fold_pw' hat keine Bedeutung, wenn die Einstellung für 'password' den Wert 'N' hat. Da kein Kennwort gesendet wird, spielt die Groß-/Kleinschreibung keine Rolle.
4. Vermeiden Sie Nullwerte für diese Optionen. Ein Nullwert könnte attraktiv erscheinen, weil DB2 mehrere Versuche unternimmt, Benutzer-IDs und Kennwörter aufzulösen. Aber die Leistung könnte dadurch beeinträchtigt werden (es ist möglich, dass DB2 bis zu viermal eine Benutzer-ID und ein Kennwort sendet, bevor die Authentifizierung an der Datenquelle erfolgreich durchgeführt wird).

Verwenden von Server-Durchgriffsitzungen

Durchgriffsitzungen ermöglichen Anwendungen die direkte Kommunikation mit einem Server unter Verwendung der eigenen Client-Zugriffsmethode und der eigenen SQL-Version.

Durchgriffsitzungen sind in folgenden Situationen nützlich:

- Anwendungen müssen in der Datenquelle Objekte erstellen bzw. INSERT-, UPDATE- oder DELETE-Operationen ausführen.
- DB2 unterstützt keine eindeutige Datenquellenoperation.

Verwenden Sie in einer Durchgriffsitzung zum Verweisen auf ein Objekt den wahren Namen (nicht den Kurznamen) des betreffenden Objekts.

Mit der Anweisung SET PASSTHRU können Sie eine Durchgriffsitzung starten und direkt auf einen Server zugreifen. Diese Anweisung muss dynamisch abgesetzt werden. Es folgt ein Beispiel für diese Anweisung:

```
SET PASSTHRU BACKEND
```

startet eine Durchgriffsitzung für die Datenquelle BACKEND.

Weitere Informationen zu SET PASSTHRU und zur SQL-Verarbeitung in Durchgriffsitzungen finden Sie im Handbuch *SQL Reference*.

Erstellen eines Kurznamens

In einer zusammengeschlossenen Datenbank sind Kurznamen Kennungen für Tabellen, Aliasnamen und Sichten von Datenquellen. Verteilte Anforderungen verweisen in der Regel auf Kurznamen und nicht auf Datenquellentabellen oder -sichten.

Kurznamen sind Bestandteil der Strategie für Positionstransparenz in DB2. Kurznamen sind beim Lokalisieren und effizienten Zugreifen auf Datenquellen von den Definitionen für Datenquellenpositionen auf dem Server abhängig. Eine Anweisung ALTER SERVER ermöglicht zum Beispiel transparentes Aktualisieren von Server-Leistungsdaten und Versionsinformationen für alle Benutzer und Anwendungen, ohne neue Kurznamen oder Änderungen im Anwendungscode erforderlich zu machen.

Kurznamen können in der Steuerzentrale oder im Befehlszeilenprozessor erstellt werden. Sie können mehrere Kurznamen für dieselbe Datenquellentabelle oder -sicht erstellen.

Kurznamen können in statischen SQL-Anweisungen nicht verwendet werden.

Führen Sie vor dem Erstellen eines Kurznamens das Äquivalent des Befehls RUNSTATS in der Datenquelle aus, und aktualisieren Sie die Statistik für Datenquellenobjekte. Statistische Informationen zu Datenquellen werden gesammelt, wenn ein Kurzname erstellt und im Katalog der zusammengesetzten Datenbank gespeichert wird. Zu diesen Katalogdaten gehören Tabellen- und Spaltendefinitionen sowie, falls verfügbar, Indexdefinitionen und Statistikdaten.

Mit der folgenden SQL-Anweisung wird der Kurzname CUSTOMER erstellt:

```
CREATE NICKNAME CUSTOMER FOR OS390A.SHAWNB.CUSTLIST
```

Sie müssen über die Berechtigung SYSADM oder DBADM, die Datenbankberechtigung IMPLICIT_SCHEMA oder die Schemenberechtigung CREATEIN (für das aktuelle Schema) in der zusammengesetzten Datenbank verfügen, um diese Anweisung verwenden zu können.

Weitere Einzelheiten zur Verwendung der Anweisung CREATE NICKNAME finden Sie im Handbuch *SQL Reference*.

Verweisen auf Kurznamen und Datenquellenobjekte

Verweise auf Datenquellenobjekte verwenden in der Regel den definierten Kurznamen. Einzige Ausnahme von dieser Regel ist ein Verweis innerhalb einer Durchgriffsitzung (weitere Informationen hierzu finden Sie in „Verwenden von Server-Durchgriffsitzungen“ auf Seite 185). Wenn Sie zum Beispiel den Kurznamen DEPT für die Datenquellentabelle DB2MVS1.PERSON.DEPT definieren, ist die Anweisung SELECT * FROM DEPT zulässig, die Anweisung SELECT * FROM DB2MVS1.PERSON.DEPT jedoch nicht.

Arbeiten mit Kurznamen und Datenquellenobjekten

Die meisten Dienstprogrammbefehle (LOAD, IMPORT, EXPORT, REORGCHK, REORGANIZE TABLE) unterstützen Kurznamen nicht.

COMMENT ON wird unterstützt. Dieser Befehl aktualisiert den Systemkatalog der zusammengeschlossenen Datenbank.

Insert-, update und delete-Operationen an Kurznamen werden nicht unterstützt.

Identifizieren vorhandener Kurznamen und Datenquellen

Nach dem Erstellen mehrerer Kurznamen können Sie anhand der folgenden Informationen feststellen, zu welcher Datenquelle ein bestimmter Kurzname gehört, oder alle Kurznamen einer bestimmten Datenquelle ermitteln.

Identifizieren eines Kurznamens und der zugehörigen Datenquelle

In diesem Beispiel wird davon ausgegangen, dass Sie zwar den Kurznamen (*PAYROLL*) und dessen Ersteller (*ACCTG*) kennen, jedoch zusätzliche Informationen zu der Datenquelle benötigen. Fragen Sie mit der folgenden SQL-Anweisung zunächst ab, unter welcher Bezeichnung *PAYROLL* in der dazugehörigen Datenquelle (*SERVER*) bekannt ist.

```
select option, setting
       from syscat.taboptions
  where tablename = 'PAYROLL'
     and tabschema = 'ACCTG'
     and option in ('SERVER', 'REMOTE_SCHEMA', 'REMOTE_TABLE');
```

Die Antwortgruppe dieser Anweisung lautet *DB2_MVS*, *FINANCE*, *DEPTJ35_PAYROLL*. Sie wissen nun, dass *PAYROLL* der Kurzname für die Tabelle *DEPTJ35_PAYROLL* des Eigners *FINANCE* auf dem Server *DB2_MVS* ist. Diese Informationen können Sie in einer weiteren Anweisung wie folgt nutzen:

```
select option, setting
       from syscat.serveroptions
  where servername = 'DB2_MVS'
     and option in ('NODE', 'DBNAME');
```

Die Antwortgruppe dieser Anweisung lautet *REGIONW* und *DB2MVSDB3*. Sie wissen nun, dass sich die Tabelle *DEPTJ35_PAYROLL* in der Tabelle *DB2MVSDB3* auf dem Knoten *REGIONW* befindet.

Anhand dieser Informationen können Sie mit dem Befehl *LIST NODE DIRECTORY* Informationen zu dem Knoten *REGIONW* abrufen, wie zum Beispiel das verwendete Übertragungsprotokoll und die verwendete Sicherheitseinstufung. Wenn der Knoten einer anderen Datenquelle als der *DB2*-Familie angehörte, müssten Sie ähnliche Informationen in den Konfigurationsdateien dieser Datenquelle suchen. Wäre der Knoten zum Beispiel eine Oracle-Datenquelle, könnten Sie ähnliche Informationen aus der Oracle-Datei *tnsnames.ora* abrufen.

Einzelheiten zu Systemkatalogsichten finden Sie im Handbuch *SQL Reference*.

Identifizieren aller unter DB2 bekannten Kurznamen

Mit der folgenden SQL-Anweisung können Sie eine Liste aller Kurznamen abrufen, die in der zusammengeschlossenen Datenbank bekannt sind, einschließlich des Schemennamens und des fernen Servers für jeden dieser Kurznamen.

```
select tabname,tabschema, setting as remote_server
       from syscat.taboptions
       where option = 'SERVER';
```

Erstellen eines Index, einer Indexerweiterung oder einer Indexspezifikation

Ein Index ist eine Liste der Speicherpositionen von Zeilen, die nach dem Inhalt einer oder mehrerer angegebener Spalten sortiert sind. In der Regel dienen Indizes zur Beschleunigung des Zugriffs auf eine Tabelle. Sie können aber auch Zwecke des logischen Datenentwurfs erfüllen. Zum Beispiel lässt ein *eindeutiger Index* nicht zu, dass in die Spalten ein Wert mehrfach eingegeben wird, wodurch gewährleistet wird, dass nicht zwei Zeilen in einer Tabelle identisch sind. Indizes können auch erstellt werden, um eine aufsteigende oder absteigende Reihenfolge der Werte in einer Spalte anzugeben.

Eine Indexerweiterung ist ein Indexobjekt, das bei Indizes verwendet wird, die über Spalten für strukturierte oder einzigartige Datentypen verfügen.

Eine Indexspezifikation ist ein Konstrukt aus Metadaten. Sie teilt dem Optimierungsprogramm mit, dass ein Index für ein Datenquellenobjekt (Tabelle oder Sicht) vorhanden ist, auf das ein Kurzname verweist. Eine Indexspezifikation enthält keine Listen mit Zeilenpositionen, sie ist nur eine Beschreibung für einen Index. Das Optimierungsprogramm verwendet die Indexspezifikation dazu, den Zugriff auf das Objekt zu verbessern, auf das der Kurzname hinweist. Bei der Ersterstellung eines Kurznamens wird eine Indexspezifikation generiert, wenn für die zugrundeliegende Tabelle ein Index in einem für DB2 erkennbaren Format in der Datenquelle vorhanden ist.

Anmerkung: Erstellen Sie, falls erforderlich, Indexspezifikationen zu Tabellenkurznamen oder Sichtkurznamen, wenn sich die Sicht auf nur eine Tabelle bezieht.

Erstellen Sie in folgenden Situationen manuell einen Index oder eine Indexspezifikation:

- Wenn dies die Leistung verbessert. Wenn Sie zum Beispiel das Optimierungsprogramm veranlassen wollen, eine bestimmte Tabelle bzw. einen bestimmten Kurznamen als innere Tabelle einer Verknüpfung mit Verschachtelungsschleife zu verwenden, erstellen Sie eine Indexspezifikation für die Verknüpfungsspalte (falls kein solcher Index vorhanden ist). Weitere Informationen zum Einsatz eines Index oder einer Indexspezifikation finden Sie im Handbuch *Systemverwaltung: Optimierung*.

- Wenn ein Index für eine Basistabelle hinzugefügt wurde, nachdem der Kurzname für diese Tabelle erstellt wurde.

Indexspezifikationen können erstellt werden, wenn kein Index für die Basistabelle vorhanden ist (DB2 sucht nicht nach dem fernen Index, wenn Sie die Anweisung CREATE INDEX ausführen). Eine Indexspezifikation erzwingt auch dann nicht die Eindeutigkeit der Zeilen, wenn das Schlüsselwort UNIQUE angegeben wird.

DB2 Index Advisor ist ein Assistent, der Sie bei der Auswahl einer optimalen Gruppe von Indizes unterstützt. Auf diesen Assistenten kann über die Steuerzentrale zugegriffen werden. Das vergleichbare Dienstprogramm heißt *db2advis*.

Ein Index wird mit Hilfe von Spalten der Basistabelle definiert. Er kann vom Ersteller einer Tabelle oder von einem Benutzer, der weiß, dass für bestimmte Spalten ein direkter Zugriff erforderlich ist, definiert werden. Ein Primärindexschlüssel wird automatisch anhand des Primärschlüssels erstellt, sofern kein benutzerdefinierter Index bereits existiert.

Für eine bestimmte Basistabelle kann eine beliebige Anzahl von Indizes definiert werden, und diese Indizes können sich positiv auf die Verarbeitung von Abfragen auswirken. Je mehr Indizes jedoch vorhanden sind, desto mehr Änderungen muss der Datenbankmanager während der Aktualisierungs-, Einfüge- und Löschoperationen durchführen. Daher kann die Erstellung einer großen Anzahl von Indizes für eine Tabelle, die häufig aktualisiert wird, die Verarbeitung von Anforderungen verlangsamen. Die Verwendung von Indizes ist also nur dann sinnvoll, wenn sich klare Vorteile für den häufigen Zugriff ergeben.

Alle Spalten, die Teil eines Indexschlüssels sind, sind auf 255 Byte begrenzt.

Anmerkung: Die Variable der Registrierdatenbank DB2_INDEX_2BYTEVARLEN kann verwendet werden, um die Angabe von Spalten mit einer Länge von über 255 Byte als Teil eines Indexschlüssels zu ermöglichen.

Die maximale Anzahl der Spalten in einem Index beträgt 16. Beim Indexieren einer typisierten Tabelle beträgt die maximale Anzahl an Spalten 15. Die maximale Länge eines Indexschlüssels beträgt 1024 Byte. Wie zuvor erwähnt, kann es durch viele Indexschlüssel in einer Tabelle zu einer Verlangsamung der Anforderungsverarbeitung kommen. Durch umfangreiche Indexschlüssel kann es ebenfalls zur Reduzierung der Verarbeitungsgeschwindigkeit bei Anforderungen kommen.

Ein *Indexschlüssel* ist eine Spalte oder eine Gruppe von Spalten, mit denen ein Index definiert wird. Die Zweckmäßigkeit eines Index hängt von seinem Schlüssel ab. Obwohl die Reihenfolge der Spalten, die einen Indexschlüssel bilden, bei der Erstellung des Indexschlüssel keine Rolle spielt, kann sie für das Optimierungsprogramm bei der Entscheidung von Bedeutung sein, ob ein Index verwendet werden soll oder nicht.

Wenn die Tabelle, für die ein Index erstellt wird, leer ist, wird der Index zwar erstellt, aber es werden erst Indexeinträge erstellt, wenn die Tabelle geladen oder Zeilen eingefügt werden. Ist die Tabelle nicht leer, erstellt der Datenbankmanager die Indexeinträge während der Verarbeitung der Anweisung `CREATE INDEX`.

Bei einem *Clustering-Index* werden neue Zeilen physisch nahe bei vorhandenen Zeilen mit ähnlichen Schlüsselwerten eingefügt. Dies führt zu einer Leistungssteigerung während der Ausführung von Abfragen, weil sich eine linearere Zugriffsweise auf die Daten und ein effektiverer Vorablesezugriff ergeben.

Wenn Sie einen Primärschlüsselindex als Clustering-Index haben wollen, sollte der Primärschlüssel nicht bei der Erstellung mit `CREATE TABLE` angegeben werden. Wenn der Primärschlüssel einmal erstellt ist, kann der zugehörige Index nicht geändert werden. Führen Sie stattdessen die Anweisung `CREATE TABLE` ohne Primärschlüsselklausel (Primary Key) aus. Führen Sie anschließend die Anweisung `CREATE INDEX` aus, in der Sie die Clustering-Attribute angeben. Verwenden Sie schließlich die Anweisung `ALTER TABLE`, um einen Primärschlüssel hinzuzufügen, der dem gerade erstellten Index entspricht. Dieser Index wird dann als der Primärschlüsselindex verwendet.

Im allgemeinen bleibt die Clusterbildung effektiver erhalten, wenn der Clustering-Index eindeutig ist.

Spaltendaten, die nicht als Teil des eindeutigen Indexschlüssels, aber im Index gespeichert/gepflegt werden, werden als *INCLUDE*-Spalten bezeichnet. `INCLUDE`-Spalten können nur für eindeutige Indizes angegeben werden. Bei der Erstellung eines Index mit `INCLUDE`-Spalten werden nur die eindeutigen Schlüsselspalten sortiert und im Hinblick auf Eindeutigkeit berücksichtigt. Die Verwendung von `INCLUDE`-Spalten verbessert die Leistung beim Abrufen von Daten, wenn es darum geht, ob ein reiner Indexzugriff erfolgen kann.

Der Datenbankmanager verwendet zum Speichern von Indizes eine B+-Baumstruktur, deren unterste Ebene aus Blattknoten besteht. Auf den Blattknoten oder -seiten sind die eigentlichen Indexschlüsselwerte gespeichert. Bei der Indexerstellung können Sie festlegen, dass diese Indexseiten online zusammengefügt oder reorganisiert werden. Durch diese Online-Reorganisation des Index wird verhindert, dass nach zahlreichen Lösch- und Aktualisierungsvorgängen auf vielen Blattseiten eines Index nur noch wenige Indexschlüssel vor-

handen sind. In einem solchen Fall könnte ohne die Online-Reorganisation der freie Speicherplatz nur durch eine Offline-Reorganisation der Daten und des Index zurückgewonnen werden. Bei der Entscheidung, ob ein Index so erstellt werden soll, dass das Reorganisieren der Indexseiten im Online-Modus möglich ist, sollte die folgende Frage berücksichtigt werden. Übersteigen die zusätzlichen Leistungsaufwände für die Prüfung auf Speicherplatz zur Ausführung von Mischoperationen beim Löschen von Schlüsseln und die tatsächlichen Aufwände zum Abschluss der Mischoperation (wenn genügend Speicherplatz verfügbar ist) die Vorteile der besseren Speicherplatznutzung für den Index. Sind diese Aufwände andererseits geringer als die reduzierten Anforderungen für die Offline-Reorganisation, um diesen Speicherplatz wieder zurückzufordern?

Anmerkung: Die durch eine Online-Reorganisation zurückgewonnenen Seiten können nur für andere Indizes in der gleichen Tabelle verwendet werden. Bei einer umfassenden Reorganisation stehen die zurückgewonnenen Seiten für andere Objekte zur Verfügung (bei Speicherplatzverwaltung durch die Datenbank) oder als Plattenspeicherplatz (bei Speicherplatzverwaltung durch das System). Außerdem werden durch die Online-Reorganisation keine Nichtblattseiten des Index zurückgewonnen, während die umfassende Reorganisation den Index auf Minimalgröße bringt, indem Nichtblattseiten und Blattseiten sowie die Anzahl der Indexstufen reduziert werden.

Weitere Informationen zum Implementieren eines Index mit Online-Reorganisation finden Sie in „Verwenden der Anweisung CREATE INDEX“ auf Seite 193.

Indizes für Tabellen in einer partitionierten Datenbank werden mit Hilfe derselben Anweisung CREATE INDEX erstellt. Sie werden anhand des Partitionierungsschlüssels der Tabelle partitioniert. Ein Index für eine Tabelle besteht aus den lokalen Indizes für die Tabelle auf dem jeweiligen Knoten in der Knotengruppe. Beachten Sie, dass in einer Mehrpartitions Umgebung der Indexschlüssel definierter eindeutiger Indizes eine Obermenge des Partitionierungsschlüssels sein muss.

Hinweis zur Leistung: Wenn Sie die folgenden Operationen durchführen wollen:

1. Tabelle erstellen
2. Tabelle laden
3. Index erstellen
4. RUNSTATS ausführen

So sollten die folgende Reihenfolge zur Ausführung der Operationen in Betracht ziehen:

1. Tabelle erstellen
2. Index erstellen
3. Tabelle mit der angeforderten Option `statistics yes` laden

Weitere Informationen zur Leistungsverbesserung bei LOAD finden Sie im Handbuch *Versetzen von Daten Dienstprogramme und Referenz*.

Indizes werden nach ihrer Erstellung gepflegt. Wenn in der Folge ein Anwendungsprogramm Zeilen in einer Tabelle aufgrund eines Schlüsselwerts ermittelt und verarbeitet, kann ein Index, der auf diesem Schlüssel basiert, verwendet werden, um direkt auf diese Zeile zuzugreifen. Dies ist wichtig, da die physische Speicherung der Zeilen einer Basistabelle nicht geordnet erfolgt. Wenn eine Zeile eingefügt wird, wird sie an der bequemsten passenden Speicherposition untergebracht, sofern nicht ein Clustering-Index definiert wurde. Wenn nach Zeilen einer Tabelle, die bestimmte Auswahlkriterien erfüllen sollen, gesucht wird und die Tabelle keinen Index hat, wird immer die gesamte Tabelle durchsucht. Ein Index optimiert dieses Abrufen von Daten, ohne dass langwierige sequenzielle Suchoperationen durchgeführt werden müssen.

Die Daten für die Indizes können im selben Tabellenbereich wie die Tabellendaten oder in einem getrennten Tabellenbereich, der nur Indexdaten enthält, gespeichert werden. Der Tabellenbereich, der zur Speicherung der Indexdaten verwendet wird, wird bei der Erstellung der Tabelle (siehe „Erstellen einer Tabelle in mehreren Tabellenbereichen“ auf Seite 153) festgelegt.

Gehen Sie wie folgt vor, um mit der Steuerzentrale einen Index zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Indizes** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Ordner **Indizes** und wählen Sie dann im Kontextmenü **Erstellen** → **Index mit Assistent** aus.
3. Führen Sie die im Assistenten aufgeführten Schritte aus, um die Task abzuschließen.

Geben Sie in der Befehlszeile Folgendes ein, um einen Index zu erstellen:

```
CREATE INDEX <name> ON <tabellenname> (<spaltenname>)
```

Die beiden folgenden Abschnitte „Verwenden eines Index“ auf Seite 193 und „Verwenden der Anweisung CREATE INDEX“ auf Seite 193 enthalten weitere Informationen zur Indexerstellung.

Verwenden eines Index

Ein Index wird nie direkt von einem Anwendungsprogramm verwendet. Die Entscheidung, ob ein Index verwendet wird und welcher der potenziell verfügbaren Indizes verwendet wird, liegt in der Zuständigkeit des Optimierungsprogramms.

Der beste Index für eine Tabelle hat folgende Merkmale:

- Er verwendet Hochgeschwindigkeitsplatten.
- Er weist eine hohe Clusterbildung auf.
- Er besteht nur aus wenigen schmalen Spalten.

Eine detaillierte Erläuterung zu den Vorteilen von Indizes finden Sie im Abschnitt über die Konzepte der Indexsuche im Handbuch *Systemverwaltung: Optimierung*.

Verwenden der Anweisung CREATE INDEX

Sie können einen Index erstellen, der mehrfach auftretende Werte zulässt (nichteindeutiger Index), um ein effizientes Abrufen über Spalten, die nicht zum Primärschlüssel gehören, zu ermöglichen, und es also erlaubt, dass doppelte Werte in der Indexspalte oder den Indexspalten vorhanden sind.

Mit der folgenden SQL-Anweisung wird ein nichteindeutiger Index namens LNAME für die Spalte LASTNAME der Tabelle LASTNAME erstellt und in aufsteigender Reihenfolge sortiert:

```
CREATE INDEX LNAME ON EMPLOYEE (LASTNAME ASC)
```

Mit der folgenden SQL-Anweisung wird ein eindeutiger Index für die Spalte mit den Telefonnummern erstellt:

```
CREATE UNIQUE INDEX PH ON EMPLOYEE (PHONENO DESC)
```

Ein eindeutiger Index stellt sicher, dass nicht zwei identische Werte in der Indexspalte oder den Indexspalten auftreten. Diese Integritätsbedingung wird am Ende der SQL-Anweisung, mit der Zeilen aktualisiert oder neue Zeilen eingefügt werden, in Kraft gesetzt. Diese Art von Index kann nicht erstellt werden, wenn die Gruppe aus einer oder mehrerer Spalten bereits doppelte Werte enthält.

Das Schlüsselwort ASC ordnet die Indexeinträge in aufsteigender Reihenfolge gemäß der Spaltenwerte an, während das Schlüsselwort DESC die Werte in absteigender Reihenfolge anordnet. Standardmäßig wird in aufsteigender Reihenfolge sortiert.

Sie können einen eindeutigen Index in zwei Spalten erstellen, wobei eine Spalte eine INCLUDE-Spalte ist. Der Primärschlüssel wird nicht in der INCLUDE-Spalte sondern in der anderen Spalte definiert. Beide werden im

Katalog als Primärschlüssel in derselben Tabelle angezeigt. In der Regel ist nur ein Primärschlüssel pro Tabelle vorhanden.

Die Klausel `INCLUDE` gibt zusätzliche Spalten an, die an das Set mit Indexschlüsselspalten angehängt werden. Alle Spalten, die mit dieser Klausel eingeschlossen werden, werden nicht verwendet, um Eindeutigkeit zu erzwingen. Die eingeschlossenen Spalten können die Leistung bei einigen Abfragen über reinen Indexzugriff verbessern. Die Spalten müssen sich von den Spalten unterscheiden, die verwendet werden, um für Eindeutigkeit zu sorgen (andernfalls wird die Fehlernachricht `SQLSTATE 42711` empfangen). Die Begrenzungen für die Anzahl der Spalten und die Summe der Längenattribute gelten für alle Spalten im eindeutigen Schlüssel und im Index.

Es wird eine Prüfung durchgeführt, um festzustellen, ob ein vorhandener Index mit der Definition des Primärschlüssels übereinstimmt (dabei werden alle `INCLUDE`-Spalten im Index ignoriert). Eine Indexdefinition stimmt überein, wenn es dasselbe Set von Spalten kennzeichnet, wobei die Spezifikationen zur Reihenfolge oder Richtung (aufsteigend oder absteigend) der Spalten keine Rolle spielt. Wenn eine übereinstimmende Indexdefinition gefunden wurde, wird die Beschreibung des Index geändert, um anzuzeigen, dass es sich - wie vom System erforderlich - um den Primärindex handelt, und der Index wird nach einer entsprechenden Überprüfung der Eindeutigkeit in eindeutig geändert, wenn er zuvor nicht eindeutig war.

Darum ist es - wie im Katalog angezeigt - möglich, mehr als einen Primärschlüssel für dieselbe Tabelle zu haben.

Beim Arbeiten mit einem strukturierten Typ kann es erforderlich sein, benutzerdefinierte Indexarten zu erstellen. Dazu ist eine Möglichkeit zum Definieren von Funktionen für Indexpflege, Indexsuche und Indexausnutzung erforderlich. Im Handbuch *SQL Reference* finden Sie Informationen zu den Voraussetzungen zum Erstellen einer Indexart.

Mit der folgenden SQL-Anweisung wird ein Clustering-Index namens `INDEX1` für die Spalte `LASTNAME` der Tabelle `EMPLOYEE` erstellt:

```
CREATE INDEX INDEX1 ON EMPLOYEE (LASTNAME) CLUSTER
```

Um den internen Speicher der Datenbank effektiv zu nutzen, verwenden Sie Clustering-Indizes mit dem Parameter `PCTFREE` der Anweisung `ALTER TABLE`, so dass neue Daten auf den richtigen Seiten eingefügt werden können. Wenn Daten auf den richtigen Seiten eingefügt werden, bleibt die Clustering-Reihenfolge erhalten. Je höher das Aufkommen an `INSERT`-Aktivitäten für die Tabelle ist, desto größer muss der `PCTFREE`-Wert (in der Tabelle) sein, der zur Erhaltung des Clusterings benötigt wird. Da dieser Index die Reihenfolge bestimmt, nach der die Daten auf den physischen Seiten gespeichert werden, kann für jede Tabelle nur ein Clustering-Index definiert werden.

Wenn andererseits die Indexschlüsselwerte dieser neuen Zeilen zum Beispiel stets neue HIGHKEY-Werte (d. h. neue Höchstwerte) sind, versucht das CLUSTERING-Attribut der Tabelle, diese Zeilen an das Ende der Tabelle zu setzen. In diesem Fall sind freie Speicherbereiche auf anderen Seiten der Erhaltung des Clusterings wenig dienlich. Daher ist es in einem solchen Fall besser, die Tabelle in den Anfügemodus (Append) zu setzen, als einen Clustering-Index zu haben und die Tabelle zum Erzielen eines großen Werts für PCTFREE zu ändern. Mit dem folgenden Befehl können Sie die Tabelle in den Anhängemodus versetzen: ALTER TABLE APPEND ON. Zusätzliche Übersichtsinformationen zu ALTER TABLE finden Sie in „Ändern von Tabellenattributen“ auf Seite 235. Zusätzliche detaillierte Informationen zu ALTER TABLE finden Sie im Handbuch *SQL Reference*.

Die obige Darstellung gilt auch für neue "Überlaufzeilen", die aus UPDATE-Operationen resultieren, die eine Zeile vergrößern.

Mit der Klausel MINPCTUSED der Anweisung CREATE INDEX wird die Schwelle für die Mindestgröße des genutzten Speicherbereichs auf einer Indexseite angegeben. Bei Verwendung dieser Klausel wird die Online-Reorganisation für diesen Index aktiviert. In diesem Fall wird anhand der folgenden Kriterien entschieden, wann die Online-Reorganisation stattfindet: Wenn nach dem Löschen eines Schlüssels auf einer Indexblattseite ein Prozentwert für benutzten Speicherplatz auf der Seite unter den angegebenen Schwellenwert absinkt, werden die angrenzenden Blattseiten daraufhin überprüft, ob die Schlüssel beider Blattseiten zu einer Indexblattseite zusammengefasst werden können.

Mit der folgenden SQL-Anweisung wird ein Index mit aktivierter Online-Reorganisation erstellt:

```
CREATE INDEX LASTN ON EMPLOYEE (LASTNAME) MINPCTUSED=20
```

Wenn ein Schlüssel aus diesem Index gelöscht wird und die verbleibenden Schlüssel auf dieser Indexseite 20 Prozent oder weniger der Indexseite belegen, wird versucht, eine Indexseite frei zu machen, indem die Schlüssel dieser Indexseite mit den Schlüsseln der angrenzenden Indexseite zusammengeführt werden. Wenn die zusammengeführten Schlüssel auf eine einzige Seite passen, wird die Reorganisation durchgeführt und eine der beiden Indexseiten gelöscht.

Die Klausel PCTFREE der Anweisung CREATE INDEX gibt den Prozentsatz jeder Indexseite an, die beim Erstellen des Index als Speicherbereich frei gelassen werden soll. Je mehr Speicherbereich Sie auf den Indexseiten frei lassen, desto weniger Seitenteilungen werden erforderlich. Dadurch brauchen Sie die Tabelle nicht mehr zu reorganisieren, um sequenzielle Indexseiten wiederzugewinnen, was den Vorablesezugriff (Prefetching) erhöht. Vorablesezugriff ist eine wichtige Komponente, die zu Leistungssteigerung führen kann. Wenn Sie

immer mit HIGHKEY-Werten zu tun haben, erwägen Sie das Senken des Werts für die Klausel PCTFREE der Anweisung CREATE INDEX. Auf diese Art wird auf jeder Indexseite so wenig Speicherbereich verschwendet wie möglich.

Wenn Sie eine replizierte Übersichtstabelle haben, muss bzw. müssen ihre Basistabelle(n) über einen eindeutigen Index verfügen, und die Spalten des Indexschlüssels müssen in der Abfrage verwendet werden, die die replizierte Übersichtstabelle definiert. Weitere Informationen zu replizierten Übersichtstabellen finden Sie im Handbuch *Systemverwaltung: Konzept*.

Bei partitionsinterner Parallelität kann die Leistung beim Erstellen von Indizes verbessert werden, indem mehrere Prozessoren zum Suchen und Sortieren von Daten während der Indexerstellung verwendet werden. Die Verwendung mehrerer Prozessoren wird durch Setzen des Parameters *intra_parallel* auf YES(1) oder ANY(-1) aktiviert. Die Anzahl von Prozessoren, die während der Indexerstellung verwendet werden, wird vom System festgelegt und wird von den Konfigurationsparametern *dft_degree* oder *max_querydegree*, vom Grad der Parallelität von Anwendungen zur Laufzeit oder vom angegebenen Grad der Parallelität bei der Kompilierung der SQL-Anweisung nicht beeinflusst. Wenn der Datenbankkonfigurationsparameter *indexsort* den Wert NO hat, verwendet die Indexerstellung nicht mehrere Prozessoren.

In Mehrpartitionsumgebungen müssen eindeutige Indizes als Obermenge des Partitionierungsschlüssels definiert werden.

Erstellen einer benutzerdefinierten erweiterten Indexart

Zur Unterstützung benutzerdefinierter Indexarten ermöglicht DB2 Universal Database das Erstellen und Anwenden eigener Logikelemente für die Primärkomponenten, die die Funktionsweise eines Index festlegen. Die folgenden Komponenten können ersetzt werden:

- **Indexverwaltung.** Diese Komponente ermöglicht die Zuordnung des Inhalts von Indexspalten zu einem bestimmten Indexschlüssel. Derartige Zuordnungen werden mit einer benutzerdefinierten Zuordnungsfunktion ausgeführt. In einem erweiterten Index kann genau eine Spalte eines strukturierten Typs verwendet werden. Anders als bei einem normalen Index kann ein erweiterter Index mehrere Indexeinträge pro Zeile umfassen. Mehrere Indexeinträge pro Zeile ermöglichen das Speichern eines Textdokuments als Objekt mit einem separaten Indexeintrag für jedes Schlüsselwort im Dokument.
- **Indexausnutzung.** Diese Komponente ermöglicht dem Anwendungsentwickler die Zuordnung von Filterbedingungen (Bereichsprädikaten) zu einer benutzerdefinierten Funktion (UDF), die andernfalls für das Optimierungsprogramm nicht transparent wäre. Hierdurch kann DB2 einen

separaten UDF-Aufruf für jede Zeile und einen Kontextwechsel zwischen Client und Server vermeiden, wodurch sich die Leistung entscheidend verbessern lässt.

Anmerkung: Die benutzerdefinierte Funktionsdefinition muss deterministisch gestaltet sein und darf die Ausführung externer Aktionen nicht zulassen, um vom Optimierungsprogramm genutzt werden zu können.

Eine wahlfreie Datenfilterfunktion kann ebenfalls angegeben werden. Das Optimierungsprogramm verwendet den Filter gegen den abgerufenen Tupel, bevor die benutzerdefinierte Funktion bewertet wird.

Nur eine Spalte mit einem strukturierten oder einzigartigen Datentyp kann die Indexerweiterung zum Erstellen einer benutzerdefinierten Indexart für diese Objekte nutzen. Der Typ des benutzerdefinierten erweiterten Index darf die folgenden Merkmale nicht aufweisen:

- Definition mit in Gruppen zusammengefassten Indizes
- Vorhandensein von INCLUDE-Spalten

Details zur Indexverwaltung

Sie definieren zwei der Komponenten, die zum Funktionieren eines Index erforderlich sind, mit der Anweisung `CREATE INDEX EXTENSION`.

Die Indexverwaltung dient zum Umsetzen des Inhalts der Indexspalte (bzw. des Quellenschlüssels) in einen Zielindexschlüssel. Der Umsetzungsprozess wird mit Hilfe einer Tabellenfunktion definiert, die zuvor in der Datenbank definiert wurde.

Die Klausel `FROM SOURCE KEY` gibt einen strukturierten oder einzigartigen Datentyp für die Quellenschlüsselspalte an, die von dieser Indexerweiterung unterstützt wird. Für die Quellenschlüsselspalte wird ein einziger Parameternamen und Datentyp angegeben und zugeordnet.

Die Klausel `GENERATE KEY USING` gibt die benutzerdefinierte Tabellenfunktion an, die zum Generieren des Indexschlüssels verwendet wird. Die von dieser Funktion erzeugte Ausgabe muss in der Klauselspezifikation `TARGET KEY` angegeben werden. Sie kann außerdem als Eingabe für die Indexfilterfunktion verwendet werden, die in der Klausel `FILTER USING` angegeben wurde.

Details zur Indexsuche

Bei der Indexsuche werden bestimmte Suchargumente einzelnen Suchbereichen zugeordnet.

Die Klausel `WITH TARGET KEY` der Anweisung `CREATE INDEX EXTENSION` gibt die Parameter für den Zielschlüssel an, bei denen es sich um die Ausgabe der benutzerdefinierten Tabellenfunktion handelt, die in der Klausel `GENERATE KEY USING` angegeben wurde. Für die Zielschlüsselspalte wird ein einziger Parameternamen und Datentyp angegeben und zugeordnet. Dieser Parameter entspricht den Spalten der Tabelle `RETURNS` der benutzerdefinierten Tabellenfunktion der Klausel `GENERATE KEY USING`.

Die Klausel `SEARCH METHODS` führt eine oder mehrere für den Index definierte Suchmethoden ein. Jede Suchmethode besteht aus einem Methodennamen, Suchargumenten, einer einen Wertebereich erzeugenden Funktion und einer wahlfreien Indexfilterfunktion. Jede Suchmethode definiert, wie Indexsuchbereiche für den zugrundeliegenden benutzerdefinierten Index von einer benutzerdefinierten Tabellenfunktion generiert werden. Darüber hinaus definiert jede Suchmethode, wie die Indexeinträge in einem bestimmten Suchbereich durch eine benutzerdefinierte Skalarfunktion weiter qualifiziert werden können, um einen einzelnen Wert zurückzugeben.

- Die Klausel `WHEN` ordnet einer Suchmethode einen Kennaussatz zu. Bei dem Kennaussatz handelt es sich um eine SQL-Kennung, die sich auf den in der Indexausnutzungsregel angegebenen Methodennamen bezieht. (Diese Regel wurde in der Klausel `PREDICATES` einer benutzerdefinierten Funktion lokalisiert.) Einer oder mehrere Parameternamen und Datentypen werden zur Verwendung als Argumente in der Bereichsfunktion und/oder der Indexfilterfunktion angegeben. Die Klausel `WHEN` gibt die Aktion an, die vom Optimierungsprogramm durchgeführt werden kann, wenn die Klausel `PREDICATES` der Anweisung `CREATE FUNCTION` mit einer eingehenden Abfrage übereinstimmt.
- Die Klausel `RANGE THROUGH` gibt die benutzerdefinierte, externe Tabellenfunktion an, die zum Generieren der Indexschlüsselbereiche eingesetzt wird. Hierdurch kann das Optimierungsprogramm das Aufrufen der zugehörigen UDF vermeiden, wenn sich die Indexschlüssel außerhalb der zulässigen Schlüsselbereiche befinden.
- Die Klausel `FILTER USING` bietet eine wahlfreie Möglichkeit zur Angabe einer benutzerdefinierten, externen Tabellenfunktion oder eines `CASE`-Ausdrucks, der zum Filtern von Indexeinträgen verwendet wird, die von der Funktion zum Generieren von Bereichen zurückgegeben werden. Wenn der von der Indexfilterfunktion oder dem `CASE`-Ausdruck zurückgegebene Wert gleich 1 ist, wird die Zeile, die dem Indexeintrag entspricht, aus der Tabelle abgerufen. Ist der zurückgegebene Wert ungleich 1, wird der Indexeintrag gelöscht. Diese Funktion ist nützlich, wenn der Aufwand für den Sekundärfilter im Vergleich zu dem Aufwand für das Bewerten der ursprünglichen Methode gering ist, und die Auswahlfrequenz des Sekundärfilters relativ niedrig ist.

Details zur Indexausnutzung

Die Indexausnutzung wird bei der Bewertung der Suchmethode ausgeführt.

Die Anweisung CREATE FUNCTION (externer Skalar) erstellt ein benutzerdefiniertes Prädikat, das mit den für die Indexerweiterung definierten Suchmethoden verwendet wird.

Die Klausel PREDICATES identifiziert mit dieser Funktion die Prädikate, die möglicherweise zur Ausnutzung der Indexerweiterungen eingesetzt werden können (und die möglicherweise die wahlfreie Klausel SELECTIVITY für die Suchbedingung des Prädikats verwenden können). Wird die Klausel PREDICATES angegeben, muss für die Funktion der Wert DETERMINISTIC mit NO EXTERNAL ACTION definiert werden.

- Die Klausel WHEN ermöglicht eine spezifische Verwendung der zu definierenden Funktion in einem Prädikat mit einem Vergleichsoperator (=, >, < etc.) und einer Konstanten bzw. einem Ausdruck (mit der Klausel EXPRESSION AS). Wenn diese Funktion von einem Prädikat mit demselben Vergleichsoperator und der angegebenen Konstanten bzw. dem angegebenen Ausdruck verwendet wird, können die Funktionen für die Filterung und die Indexausnutzung eingesetzt werden. Die Verwendung einer Konstanten dient hauptsächlich zur Verarbeitung von booleschen Ausdrücken, bei denen die Ergebnisart entweder 1 oder 0 ist. In allen anderen Fällen sollten Sie die Klausel EXPRESSION AS verwenden.
- Die Klausel FILTER USING identifiziert eine Filterfunktion, die zur Ausführung zusätzlicher Filterungsoperationen für die Ergebnistabelle eingesetzt werden kann. Es handelt sich hierbei um eine alternative und schnellere Version der definierten (und im Prädikat verwendeten) Funktion, die die Anzahl der Zeilen reduziert, für die das benutzerdefinierte Prädikat zur Feststellung übereinstimmender Zeilen ausgeführt werden muss. Sollten die durch den Index generierten Ergebnisse sich weitgehend mit den von dem benutzerdefinierten Prädikat erwarteten Ergebnissen decken, ist die Anwendung dieser Filterfunktion möglicherweise nicht erforderlich.
- Sie können wahlfrei eine Gruppe von Regeln für die einzelnen Suchmethoden einer Indexerweiterung definieren, um den Index auszunutzen. Darüber hinaus können Sie eine Suchmethode in der Indexerweiterung definieren, um die Suchziele und Suchargumente sowie die Art und Weise zu beschreiben, wie diese zur Ausführung der Indexsuche eingesetzt werden können.
 - Die Klausel SEARCH BY INDEX EXTENSION identifiziert die Indexerweiterung.
 - Die wahlfreie Klausel EXACT gibt an, dass die Indexsuchfunktion bei der Prädikatbewertung exakte Ergebnisse erzielt. Diese Klausel weist die Datenbank an, nach der Ausführung der Indexsuchfunktion nicht die ursprüngliche, vom Benutzer gestellte Prädikatfunktion oder die Filter-

funktion anzuwenden. Wird die Indexsuchfunktion nicht eingesetzt, müssen das ursprünglich bereitgestellte Prädikat sowie die ursprünglich bereitgestellten Filterfunktionen angewendet werden. Wird die Klausel EXACT nicht verwendet, wird das ursprüngliche, vom Benutzer gestellte Prädikat nach der Ausführung der Indexsuchfunktion angewendet. Das Prädikat EXACT ist nützlich, wenn die Indexsuchfunktion die gleichen Ergebnisse zurückgibt wie das Prädikat. Hierdurch wird bei der Abfrageausführung das Anwenden des benutzerdefinierten Prädikats auf die Ergebnisse der Indexsuchfunktion verhindert. Wird erwartet, dass der Index nur ein annähernd mit dem Prädikat übereinstimmendes Ergebnis erzielt, darf die Klausel EXACT nicht angegeben werden.

- Die Klausel WHEN KEY definiert das Suchziel. Für einen Schlüssel wird nur ein Suchziel angegeben. Der nach der Klausel WHEN KEY angegebene Wert identifiziert einen Parameternamen der momentan definierten Funktion. Diese Klausel wird als "wahr" (true) bewertet, wenn die Werte des benannten Parameters für Spalten stehen, die von einem Index abgedeckt werden, der auf der angegebenen Indexerweiterung basiert.
- Die Klausel USE definiert das Suchargument. Das Suchargument identifiziert, welche der in der Indexerweiterung definierte Methode verwendet werden soll. Der hier angegebene Methodenname muss mit einer Methode übereinstimmen, die in der Indexerweiterung definiert ist. Der bzw. die Parameterwerte identifizieren Parameternamen der momentan definierten Funktion. Sie dürfen mit keinem der Parameternamen übereinstimmen, die im Suchziel angegeben wurden. Die Anzahl der Parameterwerte und die zugehörigen Datentypen müssen mit den Parametern übereinstimmen, die für die Methode in der Indexerweiterung definiert wurden. Für integrierte und einzigartige Datentypen muss eine exakte Übereinstimmung vorhanden sein, wobei dieselben strukturierten Typen erforderlich sind.

Szenario zum Definieren einer Indexerweiterung

Im Folgenden ist ein Szenario zum Definieren einer Indexerweiterung aufgeführt:

1. Definieren Sie die strukturierten Typen (für Formen). Definieren Sie mit der Anweisung CREATE TYPE eine Typhierarchie, bei der die Form (shape) ein übergeordneter Typ und Nullform (nullshape), Punkt (point), Linie (line) und Vieleck (polygon) untergeordnete Typen sind. Diese strukturierten Typen dienen zum Modellieren von räumlichen Entitäten. Beispiel: Der Standort eines Lagerhauses ist ein Punkt, der Verlauf eines Flusses ist eine Linie und die Grenze eines Geschäftsbereichs ist ein Vieleck. Ein minimaler Begrenzungsrahmen (mbr) wird hierbei als Attribut verwendet. Das Attribut 'gtype' gibt an, ob die zugehörige Entität ein Punkt, eine Linie oder ein Vieleck ist. Geografische Grenzen werden mit den

Attributen 'numpart', 'numpoint' und 'geometry' modelliert. Alle anderen Attribute werden ignoriert, weil sie für das aktuelle Szenario keine Rolle spielen.

2. Erstellen Sie die Indexerweiterung.
 - Erstellen Sie mit der Anweisung CREATE FUNCTION Funktionen, die für die Schlüsselumsetzung (gridentry), Bereichserstellung (gridrange) und die Indexfilterung (checkduplicate und mbroverlap) verwendet werden.
 - Erstellen Sie mit der Anweisung CREATE INDEX EXTENSION die restlichen erforderlichen Komponenten des Index.
3. Erstellen Sie die Schlüsselumsetzung, die der Indexverwaltungs-komponente eines Indexes entspricht.

```
CREATE INDEX EXTENSION iename (parm_name datentyp, ...)
FROM SOURCE KEY (parm_name datentyp)
GENERATE KEY USING tabellenfunktionsaufruf
...
```

Mit der Klausel FROM SOURCE KEY kann der Parameter- und Datentyp der Schlüsselumsetzung festgestellt werden. Die Klausel GENERATE KEY USING dient zur Identifizierung der Funktion, die zum Zuordnen des Quellenschlüssels zu dem von der Funktion generierten Wert verwendet wird.

4. Definieren Sie die Funktionen für die Bereichserstellung und die Indexfilterung, die der Indexsuchkomponente eines Index entsprechen.

```
CREATE INDEX EXTENSION iename (parm_name datentyp, ...)
...
WITH TARGET KEY
WHEN methodenname (parm_name datentyp, ...)
RANGE THROUGH aufruf_bereichserstellungsfunktion
FILTER USING aufruf_indexfilterfunktion
```

Mit der Klausel WITH TARGET KEY wird die Suchmethodendefinition festgestellt. Die Klausel WHEN dient zur Feststellung des Methoden-namens. Die Klausel RANGE THROUGH identifiziert die Funktion, die zur Eingrenzung des Bereichs für den verwendeten Index eingesetzt wird. Die Klausel FILTER USING dient zur Feststellung der Funktion, die zum Eliminieren nicht benötigter Elemente aus den resultierenden Indexwerten dient.

Anmerkung: Die Klausel FILTER USING kann anstelle einer Indexfilterfunktion zur Identifizierung eines CASE-Ausdrucks verwendet werden.

5. Definieren Sie die Prädikate zur Nutzung der Indexerweiterung.

```
CREATE FUNCTION within (x shape, y shape)
RETURNS INTEGER
...
```

```

PREDICATES
  WHEN = 1
    FILTER USING mbrWithin (x..mbr..xmin, ...)
    SEARCH BY INDEX EXTENSION rastererweiterung
    WHEN KEY (parm_name) USE methodenname(parm_name)

```

Mit der Klausel PREDICATES können Sie ein oder mehrere Prädikate einführen, die bei jeder Klausel WHEN gestartet werden sollen. Die Klausel WHEN beginnt die Angabe für das Prädikat mit einem Vergleichsoperator, dem entweder eine Konstante oder eine Klausel EXPRESSION AS folgt. Die Klausel FILTER USING identifiziert eine Filterfunktion, die zur Ausführung zusätzlicher Filterungsoperationen für die Ergebnistabelle eingesetzt werden kann. Es handelt sich hierbei um eine günstigere Version der definierten (und im Prädikat verwendeten) Funktion, die die Anzahl der Zeilen reduziert, für die das benutzerdefinierte Prädikat zur Feststellung übereinstimmender Zeilen ausgeführt werden muss. Die Klausel SEARCH BY INDEX EXTENSION gibt an, wo die Indexausnutzung stattfindet. Die Indexausnutzung definiert die Gruppe von Regeln für die Suchmethode einer Indexerweiterung, die zur Nutzung des Indexes verwendet werden kann. Die Klausel WHEN KEY gibt die Ausnutzungsregel an. Die Ausnutzungsregel beschreibt die Suchziele und -argumente sowie die Art und Weise, wie diese Elemente zur Ausführung der Indexsuche über eine Suchmethode verwendet werden können.

6. Definieren Sie eine Filterfunktion.

```
CREATE FUNCTION mbrWithin (...)
```

Die hier definierte Funktion wird zur Verwendung im Prädikat der Indexerweiterung erstellt.

Damit das Abfrageoptimierungsprogramm Indizes, die zur Verbesserung der Abfrageleistung erstellt wurden, erfolgreich nutzen kann, steht die Option SELECTIVITY für den Funktionsaufruf zur Verfügung. Wenn Sie ungefähr abschätzen können, wie hoch der Prozentsatz der von einem Prädikat zurückgegebenen Zeilen ist, können Sie die Option SELECTIVITY beim Funktionsaufruf verwenden, um die Auswahl eines effizienteren Zugriffspfads durch das DB2-Optimierungsprogramm zu vereinfachen.

Im folgenden Beispiel berechnet die benutzerdefinierte Funktion within (auf der Basis des ersten und zweiten Parameters) das Zentrum und den Radius und erstellt eine Anweisungsfolge mit einer entsprechenden Selektivität:

```

SELECT * FROM customer
  WHERE within(loc, circle(100, 100, 10)) = 1 SELECTIVITY .05

```

In diesem Beispiel filtert das angegebene Prädikat (SELECTIVITY .05) 95 % der Zeilen in der Tabelle customer aus.

Kapitel 4. Ändern einer Datenbank

Im vorliegenden Kapitel werden die Faktoren behandelt, die vor dem Ändern einer Datenbank berücksichtigt werden müssen. Darüber hinaus enthält es Informationen zum Ändern oder Löschen von Datenbankobjekten.

Vor dem Ändern der Datenbank

Einige Zeit nach der Implementierung eines Datenbankentwurfs beabsichtigen Sie eventuell, diesen zu ändern. Daher sollten Sie die Hauptentwurfspunkte des vorherigen Entwurfs überdenken. Die folgenden Aspekte sollten besonders berücksichtigt werden:

- „Ändern der logischen und physischen Datenbankmerkmale“
- „Ändern der Lizenzinformationen“
- „Ändern der Exemplare“
- „Ändern der Umgebungsvariablen und der Variablen der Profilregistrierdatenbank“ auf Seite 207
- „Ändern der Knotenkonfigurationsdatei“ auf Seite 208
- „Ändern der Datenbankkonfiguration“ auf Seite 208

Ändern der logischen und physischen Datenbankmerkmale

Bevor Sie Änderungen mit Auswirkung auf die gesamte Datenbank vornehmen, sollten Sie alle Entscheidungen zum logischen und physischen Entwurf der Datenbank überprüfen. Wenn Sie beispielsweise einen Tabellenbereich ändern, sollten Sie Ihre Entscheidung im Hinblick auf die Verwendung von SMS- oder DMS-Speicherarten überdenken. (Weitere Informationen hierzu finden Sie im Handbuch *Systemverwaltung: Konzept*.)

Ändern der Lizenzinformationen

Im Rahmen der Lizenzverwaltung für Ihre DB2-Produkte kann es erforderlich werden, die Anzahl der Lizenzen zu erhöhen. Mit der Lizenzzentrale innerhalb der Steuerzentrale können Sie die aktuelle Nutzung der installierten Produkte prüfen und gegebenenfalls die Anzahl der Lizenzen entsprechend aufstocken.

Ändern der Exemplare

Die Exemplare wurden so gestaltet, dass sie durch das Installieren und/oder Deinstallieren anderer Produkte möglichst wenig beeinflusst werden.

In den meisten Fällen erhalten bzw. verlieren vorhandene Exemplare automatisch den Zugriff auf die Funktionen des Produkts, das installiert bzw. deinstalliert wird. Beim Installieren bestimmter Programmdateien oder Komponenten

ten übernehmen die vorhandenen Exemplare jedoch nicht automatisch die neuen Systemkonfigurationsparameter oder erhalten nicht automatisch Zugriff auf die neu hinzugekommenen Funktionen. In solchen Fällen muss das betreffende Exemplar aktualisiert werden.

Wenn DB2 durch Installieren einer vorläufigen Programmkorrektur aktualisiert wird, sollten alle DB2-Exemplare mit dem Befehl **db2iupdt** aktualisiert werden. Außerdem sollten Sie den Verwaltungsserver (DAS) mit dem Befehl **dasiupdt** aktualisieren.

Machen Sie sich zunächst mit den vorhandenen Exemplaren und Datenbankpartitionsservern vertraut, bevor Sie versuchen, ein Exemplar zu ändern oder zu löschen.

Auflisten der Exemplare

Gehen Sie wie folgt vor, um mit der Steuerzentrale eine Liste aller auf dem System verfügbaren Exemplare abzurufen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Datenbanken** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Datenbank, für die Exemplare aufgelistet werden sollen, und wählen Sie dann im Kontextmenü **Hinzufügen** aus.
3. Klicken Sie auf **Aktualisieren** und anschließend auf den Pfeil am Ende des Feldes **Datenbankname**, um die Liste der Exemplare anzuzeigen.
4. Drücken Sie auf **Abbruch**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Liste aller Exemplare anzuzeigen, die auf dem System verfügbar sind:

```
db2ilist
```

Geben Sie Folgendes ein, um festzustellen, welches Exemplar für die aktuelle Sitzung (unter OS/2 oder einer unterstützten Windows-Plattform) angewendet werden soll:

```
set db2instance
```

Aktualisieren der Exemplarkonfiguration

Der Befehl **db2iupdt** aktualisiert das angegebene Exemplar und führt dabei Folgendes aus:

- Ersetzen der Dateien im Unterverzeichnis `sql1ib` im Ausgangsverzeichnis des Exemplareigners.
- Wenn die Knotenart geändert wird, wird eine neue Konfigurationsdatei für den Datenbankmanager erstellt. Dabei werden Werte aus der vorhandenen Konfigurationsdatei des Datenbankmanagers mit der Standardkonfigurationsdatei des Datenbankmanagers für die neue Knotenart zusammenge-

fügt. Beim Erstellen einer neuen Konfigurationsdatei für den Datenbankmanager wird die alte Datei im Verzeichnis backup des Unterverzeichnisses `sql1ib` im Ausgangsverzeichnis des Exempleigners gesichert.

Der Befehl **db2iupdt** befindet sich im Unterverzeichnis `instance` des Unterverzeichnisses für die verwendete Release-Version (der genaue Name für das Verzeichnis variiert je nach Betriebssystem).

Der Befehl ist wie folgt anzugeben:

```
db2iupdt ExempName
```

Dabei ist `ExempName` der Anmelde-name des Exempleigners.

Diesem Befehl sind folgende wahlfreie Parameter zugeordnet:

- `-h` oder `-?`
Zeigt ein Hilfemenü für diesen Befehl an.
- `-d`
Dieser Parameter konfiguriert den Debug-Modus für die Fehlerbestimmung.
- `-a IdentArt`
Dieser Parameter gibt die Authentifizierungsart für das Exemplar an. Gültige Authentifizierungsarten sind `SERVER`, `CLIENT`, `DCS` und `DCE`. Wenn dieser Parameter nicht angegeben wird, wird standardmäßig die Authentifizierungsart `SERVER` verwendet, wenn ein DB2-Server installiert ist. Ansonsten wird die Option `CLIENT` verwendet. Die für das Exemplar angegebene Authentifizierungsart gilt für alle Datenbanken, die zu diesem Exemplar gehören.
Für UNIX-Betriebssysteme ist die Authentifizierungsart `DCE` keine zulässige Authentifizierungsart.
- `-e`
Ermöglicht Ihnen die Aktualisierung aller vorhandenen Exemplare. Vorhandene Exemplare können mit **db2ilist** angezeigt werden.
- `-u AbgeschirmtID`
Dieser Parameter gibt den Benutzer an, unter dem die abgeschirmten benutzerdefinierten Funktionen (UDFs) und die gespeicherten Prozeduren ausgeführt werden. Dies ist nicht erforderlich, wenn Sie den DB2-Client oder das DB2 Software Developer's Kit installieren. Für andere DB2-Produkte ist dieser Parameter erforderlich.

Anmerkung: `AbgeschirmtID` kann nicht „root“ oder „bin“ sein.

- `-k`
Dieser Parameter behält die aktuelle Exemplarart bei. Wenn Sie diesen Parameter nicht angeben, wird das aktuelle Exemplar in der nachfolgend angegebenen Reihenfolge auf die höchste verfügbare Exemplarart hochgestuft:

- Partitionierter Datenbankserver mit lokalen und fernen Clients (Standard-Exemplarart für DB2 Enterprise - Extended Edition)
- Datenbankserver mit lokalen und fernen Clients (Standard-Exemplarart für DB2 Universal Database Enterprise Edition)
- Client (Standard-Exemplarart für DB2-Client)

Beispiele:

- Wenn Sie nach dem Erstellen des Exemplars DB2 Universal Database Workgroup Edition oder DB2 Universal Database Enterprise Edition installiert haben, geben Sie den folgenden Befehl ein, um das Exemplar zu aktualisieren:

```
db2iupdt -u db2fenc1 db2inst1
```

- Wenn Sie nach dem Erstellen des Exemplars DB2 Connect Enterprise Edition installiert haben, können Sie den Exemplarnamen wie folgt als abgeschirmte ID verwenden:

```
db2iupdt -u db2inst1 db2inst1
```

- Mit dem folgenden Befehl können Sie Client-Exemplare aktualisieren:

```
db2iupdt db2inst1
```

Entfernen von Exemplaren

Gehen Sie wie folgt vor, um mit der Steuerzentrale das Exemplar zu entfernen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis das zu entfernende Exemplar angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Exemplarnamen und wählen Sie dann im Kontextmenü **Entfernen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um ein Exemplar zu entfernen:

```
db2idrop <exemplarname>
```

Zur Entfernung eines Exemplars über die Befehlszeile müssen Sie die folgenden vorbereitenden Arbeitsschritte ausführen:

1. Stoppen Sie alle Anwendungen, die momentan auf das Exemplar zugreifen.
2. Stoppen Sie den Befehlszeilenprozessor durch Ausführen des Befehls **db2 terminate** in jedem DB2-Befehlsfenster.
3. Stoppen Sie das Exemplar durch Ausführen des Befehls **db2stop**.
4. Sichern Sie das in der Variablen DB2INSTPROF der Profilregistrierdatenbank angegebene Exemplarverzeichnis. Unter UNIX-Betriebssystemen

könnte es sinnvoll sein, die Dateien im Verzeichnis `INSTHOME/sql11ib` zu sichern (dabei ist `INSTHOME` das Ausgangsverzeichnis des Exemplareigners). Beispielsweise könnten Sie beabsichtigen, die Konfigurationsdatei des Datenbankmanagers (`db2system`), die Datei `db2nodes.cfg`, die benutzerdefinierten Funktionen oder die abgeschirmten gespeicherten Prozeduren zu sichern.

5. (Nur unter UNIX-Betriebssystemen) Melden Sie sich als Exemplareigner ab.
6. (Nur unter UNIX-Betriebssystemen) Melden Sie sich als Benutzer mit Root-Berechtigung an.
7. Setzen Sie den folgenden Befehl **db2idrop** ab:
`db2idrop ExempName`

Dabei ist `ExempName` der Name des zu löschenden Exemplars.

Dieser Befehl entfernt den Eintrag für dieses Exemplar aus der Exemplarliste und löscht das Exemplarverzeichnis.

8. (Nur unter UNIX-Betriebssystemen) Entfernen Sie (wahlfrei) als Benutzer mit Root-Berechtigung die Gruppe und die Benutzer-ID des Exemplareigners (sofern sie nur für dieses Exemplar verwendet wird). Löschen Sie die Gruppe und die Benutzer-ID nicht, wenn Sie vorhaben, das Exemplar erneut zu erstellen.

Dieser Schritt ist wahlfrei, weil die Benutzer-ID und die Gruppe des Exemplareigners möglicherweise noch für andere Zwecke verwendet wird.

Der Befehl **db2idrop** entfernt den Exempleintrag aus der Exemplarliste und löscht das Unterverzeichnis `sql11ib` im Benutzerverzeichnis des Exemplareigners.

Ändern der Umgebungsvariablen und der Variablen der Profilregistrierdatenbank

Sie müssen beachten, welche Umgebungsvariablen (wenn überhaupt) auf Ihrem speziellen Betriebssystem geändert werden müssen. Wenn Umgebungsvariablen geändert werden und Sie nicht auf einer UNIX-Plattform arbeiten, müssen Sie das System erneut starten, damit die neuen Umgebungsvariablen wirksam werden. Überlegen Sie, ob Sie die Variablen der Profilregistrierdatenbank auf die Werte in der globalen Profilregistrierdatenbank zurücksetzen sollten, bevor Sie Ihre Datenbank ändern. Sie können anschließend die Variablen der Profilregistrierdatenbank auf solche Werte zurücksetzen, die am besten für die neue Datenbankumgebung geeignet sind. Wenn nur Variablen der Profilregistrierdatenbank geändert wurden, muss das System nicht erneut gestartet werden.

Ändern der Knotenkonfigurationsdatei

Wenn Sie Änderungen (Hinzufügen oder Löschen von Knoten oder Versetzen vorhandener Knoten) an Knotengruppen vornehmen wollen, sollten Sie die detaillierten Informationen zu den entsprechenden Arbeitsschritten im Abschnitt über das Skalieren Ihrer Konfiguration durch das Hinzufügen von Prozessoren im Handbuch *Systemverwaltung: Optimierung* lesen.

Ändern der Datenbankkonfiguration

Wenn Sie Änderungen an der Datenbank planen, sollten Sie die Werte der Konfigurationsparameter überprüfen. Einige Werte können von Zeit zu Zeit im Rahmen der laufenden Änderungen an der Datenbank je nach Benutzung angepasst werden.

Verwenden Sie zum Ändern der Datenbankkonfiguration den Assistent: Leistungskonfiguration in der Steuerzentrale. Dieser Assistent hilft Ihnen bei der Optimierung der Leistung und der ausgewogenen Verteilung des Speicherbedarfs für eine einzelne Datenbank pro Exemplar und macht Ihnen Vorschläge zur Änderung von Konfigurationsparametern und Empfehlungen zu geeigneten Werten.

Anmerkung: Wenn Sie Parameter ändern, werden die entsprechenden Werte erst dann aktualisiert, wenn die folgenden Bedingungen eintreten:

- Bei Datenbankparametern, wenn die erste neue Verbindung zur Datenbank hergestellt wird, nachdem alle Anwendungen getrennt waren.
- Bei Datenbankmanagerparametern, wenn das Exemplar das nächste Mal gestoppt und wieder gestartet wird.

In den meisten Fällen führen die vom Assistent: Leistungskonfiguration empfohlenen Werte zu einer besseren Leistung als die Standardwerte, weil sie sich an den Informationen zu Ihrer Art der Auslastung und zu Ihrem eigenen speziellen Server orientieren. Beachten Sie jedoch, dass die Werte die Leistung Ihres Datenbanksystems verbessern, aber nicht unbedingt auch optimieren können. Sie sollten als Ausgangspunkt für weitere Optimierungsmaßnahmen verstanden werden.

Gehen Sie wie folgt vor, um die Datenbankkonfiguration über die Steuerzentrale zu ändern:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Datenbanken** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem zu ändernden Exemplar bzw. der zu ändernden Datenbank und wählen Sie dann im Kontextmenü **Leistung mit Assistent konfigurieren** aus.
3. Klicken Sie auf die verschiedenen Seiten und nehmen Sie die erforderlichen Änderungen an den angezeigten Daten vor.
4. Klicken Sie auf der Seite **Ergebnisse**, um die durchgeführten Änderungen zu prüfen und die vorgeschlagenen Konfigurationsparameter anzuwenden.
5. Nach der Anwendung der Aktualisierungen klicken Sie auf **Fertigstellen**.

Geben Sie in der Befehlszeile Folgendes ein, um die Konfiguration des Datenbankmanagers zu ändern:

```
UPDATE DBM CFG FOR <aliasname_der_datenbank>  
USING <konfig_schlüsselwort>=<wert>
```

Mit einem einzigen Befehl können Sie eine oder mehrere Kombinationen aus <konfig_schlüsselwort>=<wert> ändern. Die meisten Änderungen an der Konfigurationsdatei des Datenbankmanagers werden erst wirksam, nachdem Sie in den Speicher geladen wurden. Bei einem Konfigurationsparameter für einen Server wird diese Operation während der Ausführung des Befehls **START DATABASE MANAGER** durchgeführt. Bei einem Konfigurationsparameter für einen Client erfolgt die Ausführung dieser Operation beim Neustart der Anwendung.

Zum Anzeigen oder Drucken der aktuellen Konfigurationsparameter für den Datenbankmanager können Sie den Befehl **GET DATABASE MANAGER CONFIGURATION** verwenden.

Einzelheiten zur Feineinstellung Ihres Systems mit Hilfe von Vergleichstests und zur Systemkonfiguration finden Sie in den Abschnitten über Vergleichstests und die DB2-Konfiguration im Handbuch *Systemverwaltung: Optimierung*.

Bei mehreren Partitionen: Wenn Sie eine Datenbank haben, die auf mehr als eine Partition partitioniert ist, sollte die Konfigurationsdatei für die Datenbank in allen Datenbankpartitionen die gleiche sein. Diese Konsistenz ist erforderlich, da der SQL-Compiler verteilte SQL-Anweisungen anhand der Informationen in der Konfigurationsdatei des Knotens kompiliert und einen Zugriffsplan erstellt, der die Anforderungen der jeweiligen SQL-Anweisung erfüllt. Wenn sich verschiedene Konfigurationsdateien in den Datenbankpartitionen befinden, kann dies je nachdem, in welcher Datenbankpartition die Anweisung vorbereitet wird, zu verschiedenen Zugriffsplänen führen. Verwenden Sie **db2_all**, um die Konfigurationsdateien bei allen Datenbankpartitionen zu verwalten.

Ändern einer Datenbank

Es gibt fast genau so viele Operationen zum Ändern von Datenbanken wie beim Erstellen von Datenbanken. Diese Operationen aktualisieren oder löschen Aspekte der zuvor erstellten Datenbank. Dazu gehören:

- „Löschen einer Datenbank“
- „Ändern einer Knotengruppe“ auf Seite 211
- „Ändern eines Tabellenbereichs“ auf Seite 211
- „Löschen eines Schemas“ auf Seite 218
- „Ändern von Struktur und Inhalt einer Tabelle“ auf Seite 218
- „Ändern eines benutzerdefinierten strukturierten Typs“ auf Seite 238
- „Löschen und Aktualisieren von Zeilen einer typisierten Tabelle“ auf Seite 239
- „Umbenennen einer vorhandenen Tabelle“ auf Seite 239
- „Löschen einer Tabelle“ auf Seite 240
- „Löschen eines Auslösers“ auf Seite 242
- „Löschen einer benutzerdefinierten Funktion (UDF), Typenzuordnung oder Methode“ auf Seite 243
- „Löschen eines benutzerdefinierten Datentyps (UDT) oder einer Typenzuordnung“ auf Seite 244
- „Ändern oder Löschen einer Sicht“ auf Seite 244
- „Löschen einer Übersichtstabelle“ auf Seite 246
- „Ändern oder Löschen eines Servers“ auf Seite 248
- „Ändern oder Löschen eines Kurznamens“ auf Seite 249
- „Löschen eines Index, einer Indexerweiterung oder einer Indexspezifikation“ auf Seite 251
- „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 253

Löschen einer Datenbank

Obwohl einige Objekte in einer Datenbank geändert werden können, kann die Datenbank selbst nicht geändert werden: sie muss gelöscht und neu erstellt werden. Das Löschen einer Datenbank kann weit reichende Auswirkungen haben, da alle Objekte der Datenbank, Behälter und zugehörige Dateien gelöscht werden. Die gelöschte Datenbank wird aus den Datenbankverzeichnissen entfernt (entkatalogisiert).

Gehen Sie wie folgt vor, um mit der Steuerzentrale eine Datenbank zu löschen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Datenbanken** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Datenbank, die gelöscht werden soll, und wählen Sie dann im Kontextmenü **Löschen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Datenbank zu löschen:

```
DROP DATABASE <name>
```

Mit dem folgenden Befehl wird die Beispieldatenbank SAMPLE gelöscht:

```
DROP DATABASE SAMPLE
```

Anmerkung: Wenn Sie beabsichtigen, das Experimentieren mit der Beispieldatenbank SAMPLE fortzusetzen, löschen Sie sie nicht. Wenn Sie die Beispieldatenbank SAMPLE gelöscht haben, sie jedoch erneut benötigen, können Sie sie erneut erstellen.

Ändern einer Knotengruppe

Detaillierte Informationen zum Ändern einer Knotengruppe finden Sie im Abschnitt zum Skalieren Ihrer Konfiguration durch das Hinzufügen von Prozessoren im Handbuch *Systemverwaltung: Optimierung*.

Wenn Sie Knoten hinzufügen oder löschen, müssen Sie die aktuellen Daten auf die neuen Knoten in der Knotengruppe verteilen. Führen Sie dazu den Befehl REDISTRIBUTE NODEGROUP aus. Informationen hierzu finden Sie im Abschnitt zum Umverteilen von Daten auf Datenbankpartitionen im Handbuch *Systemverwaltung: Optimierung* sowie im Handbuch *Command Reference*.

Ändern eines Tabellenbereichs

Wenn Sie eine Datenbank erstellen, werden mindestens drei Tabellenbereiche erstellt: ein Katalogtabellenbereich (SYSCATSPACE), ein Benutzertabellenbereich (mit dem Standardnamen USERSPACE1) und ein temporärer Systemtabellenbereich (mit dem Namen TEMPSPACE1). Sie müssen mindestens jeweils einen dieser Tabellenbereiche beibehalten. Sie können weitere Benutzertabellenbereiche und temporäre Tabellenbereiche nach Wunsch hinzufügen.

Anmerkung: Sie können den Katalogtabellenbereich SYSCATSPACE nicht löschen oder einen anderen erstellen, da immer mindestens ein temporärer Systemtabellenbereich vorhanden sein muss. Sie können andere temporäre Systemtabellenbereiche erstellen. Nach der Erstellung können auch die Seitengröße und die Speicherbereichsgröße eines Tabellenbereichs nicht mehr geändert werden.

In diesem Abschnitt werden folgende Operationen zur Änderung von Tabellenbereichen behandelt:

- „Hinzufügen eines Behälters in einem DMS-Tabellenbereich“ auf Seite 212
- „Modifizieren von Behältern in einem DMS-Tabellenbereich“ auf Seite 213
- „Hinzufügen eines Behälters in einem SMS-Tabellenbereich einer Partition“ auf Seite 214
- „Umbenennen eines Tabellenbereichs“ auf Seite 215

- „Ändern des Status eines Tabellenbereichs“ auf Seite 215
- „Löschen eines Benutzertabellenbereichs“ auf Seite 216
- „Löschen eines temporären Systemtabellenbereichs“ auf Seite 216.

Informationen zum Entwurf von Tabellenbereichen finden Sie im Handbuch *Systemverwaltung: Konzept*.

Hinzufügen eines Behälters in einem DMS-Tabellenbereich

Sie können die Größe eines DMS-Tabellenbereichs (d. h. eines mit der Klausel `MANAGED BY DATABASE` erstellten Tabellenbereichs) erhöhen, indem Sie dem Tabellenbereich einen oder mehrere Behälter hinzufügen.

Der Inhalt des Tabellenbereichs wird auf alle Behälter gleichmäßig neu verteilt. Während dieser Neuverteilung wird der Zugriff auf den Tabellenbereich nicht eingeschränkt. Wenn mehr als ein Behälter hinzugefügt werden muss, sollten diese Behälter gleichzeitig hinzugefügt werden.

Gehen Sie wie folgt vor, um mit der Steuerzentrale einen Behälter zu einem DMS-Tabellenbereich hinzuzufügen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellenbereiche** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Tabellenbereich, zu dem der Behälter hinzugefügt werden soll, und wählen Sie dann im Kontextmenü **Ändern** aus.
3. Klicken Sie auf **Hinzufügen**, geben Sie die erforderlichen Informationen ein und klicken Sie anschließend auf **OK**.
4. Wenn sich der Tabellenbereich in einer Umgebung mit partitionierten Datenbanken befindet, klicken Sie auf **Erweitert**, wenn die Leistungsparameter für den Tabellenbereich geändert werden sollen.
5. Klicken Sie auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um einen Behälter zu einem DMS-Tabellenbereich hinzuzufügen:

```
ALTER TABLESPACE <name>
  ADD (DEVICE '<pfad>' <größe>)
```

Das folgende Beispiel erläutert das Hinzufügen von zwei neuen Einheitenbehältern (mit jeweils 10 000 Seiten) zu einem Tabellenbereich auf einem UNIX-System.

```
ALTER TABLESPACE RESOURCE
  ADD (DEVICE '/dev/rhd9' 10000,
      DEVICE '/dev/rhd10' 10000)
```


Beachten Sie, dass die Anweisung ALTER TABLESPACE es ermöglicht, auch andere Merkmale des Tabellenbereichs, die sich auf die Leistung auswirken können, zu ändern. Weitere Informationen finden Sie im Abschnitt über die Auswirkung des Tabellenbereichs auf die Abfrageoptimierung im Handbuch *Systemverwaltung: Optimierung*.

Modifizieren von Behältern in einem DMS-Tabellenbereich

Sie können die Größe von Behältern in einem DMS-Tabellenbereich (d. h. einem mit der Klausel MANAGED BY DATABASE erstellten Tabellenbereich) erhöhen, indem Sie die Größe von einem oder mehreren Behältern anpassen oder einen oder mehrere dem Tabellenbereich zugeordnete Behälter erweitern. Sie sollten die Methode der Größenänderung in Betracht ziehen, wenn Sie die neue obere Begrenzung für die Größe des Behälters kennen. Sie sollten die Methode der Erweiterung in Betracht ziehen, wenn Sie die aktuelle Größe des Behälters nicht kennen (oder dies für Sie keine Bedeutung hat).

Geben Sie in der Befehlszeile Folgendes ein, um die Größe eines oder mehrerer Behälter in einem DMS-Tabellenbereich anzupassen:

```
ALTER TABLESPACE <name>
  RESIZE (DEVICE '<pfad>' <größe>)
```

Das folgende Beispiel erläutert das Erhöhen der Größe von zwei Einheitenbehältern (denen bereits 1 000 Seiten zugeordnet wurden) in einem Tabellenbereich auf einem UNIX-System:

```
ALTER TABLESPACE HISTORY
  RESIZE (DEVICE '/dev/rhd7' 2000,
         DEVICE '/dev/rhd8' 2000)
```

Durch diese Aktion wurde die Größe der Einheitenbehälter von 1 000 Seiten auf 2 000 Seiten erhöht. Ähnlich wie beim Hinzufügen neuer Behälter wird der Inhalt des Tabellenbereichs auf alle verfügbaren Behälter neu gleichmäßig verteilt. Während dieser Neuverteilung wird der Zugriff auf den Tabellenbereich nicht eingeschränkt.

Geben Sie in der Befehlszeile Folgendes ein, um einen oder mehrere Behälter in einem DMS-Tabellenbereich zu erweitern:

```
ALTER TABLESPACE <name>
  EXTEND (DEVICE '<pfad>' <größe>)
```

Das folgende Beispiel erläutert das Erhöhen der Größe von zwei Einheitenbehältern (denen bereits 1 000 Seiten zugeordnet wurden) in einem Tabellenbereich auf einem UNIX-System:

```
ALTER TABLESPACE HISTORY
  EXTEND (DEVICE '/dev/rhd11' 1000,
         DEVICE '/dev/rhd12' 1000)
```

Durch diese Aktion wurde die Größe der Einheitenbehälter von 1 000 Seiten auf 2 000 Seiten erhöht. Ähnlich wie beim Hinzufügen neuer Behälter wird der Inhalt des Tabellenbereichs auf alle verfügbaren Behälter neu gleichmäßig verteilt. Während dieser Neuverteilung wird der Zugriff auf den Tabellenbereich nicht eingeschränkt.

DMS-Behälter (sowohl Dateibehälter als auch Behälter mit unformatierten Einheiten), die während oder nach der Erstellung des Tabellenbereichs hinzugefügt wurden oder nach der Erstellung des Tabellenbereichs erweitert wurden, werden parallel über die Vorablesefunktionen ausgeführt. Um die parallele Verarbeitung der Operationen zum Erstellen oder zur Größenänderung von Behältern zu verbessern, können Sie die Anzahl der im System ausgeführten Vorablesefunktionen erhöhen. Der einzige Prozess, der nicht parallel ausgeführt werden kann, ist das Protokollieren dieser Aktionen und im Falle des Erstellens von Behältern, das Kennzeichnen der Behälter.

Anmerkung: Um die parallele Verarbeitung der Anweisungen CREATE TABLESPACE oder ALTER TABLESPACE zu maximieren (im Hinblick auf das Hinzufügen neuer Behälter zu einem vorhandenen Tabellenbereich), stellen Sie sicher, dass die Anzahl der Vorablesefunktionen größer oder gleich der Anzahl der hinzugefügten Behälter ist.

Anmerkung: Sie können die Größe der Behälter nicht reduzieren.

Beachten Sie, dass die Anweisung ALTER TABLESPACE es ermöglicht, auch andere Merkmale des Tabellenbereichs, die sich auf die Leistung auswirken können, zu ändern. Weitere Informationen finden Sie im Abschnitt über die Auswirkung des Tabellenbereichs auf die Abfrageoptimierung im Handbuch *Systemverwaltung: Optimierung*.

Hinzufügen eines Behälters in einem SMS-Tabellenbereich einer Partition
Geben Sie in der Befehlszeile Folgendes ein, um einen Behälter in einem SMS-Tabellenbereich hinzuzufügen:

```
ALTER TABLESPACE <name>  
  ADD ('<pfad>')  
  ON NODE (<partitionsnummer>)
```

Die durch die Nummer angegebene Partition und jede Partition (oder jeder Knoten) in dem Bereich der Partitionen muss in der Knotengruppe vorhanden sein, die der Tabellenbereich definiert. Eine Partitionsnummer kann nur explizit oder innerhalb eines Bereichs in genau einer Klausel ON-NODE für die Anweisung angegeben werden.

Das folgende Beispiel zeigt, wie ein neuer Behälter einer Partitionsnummer 3 der Knotengruppe hinzugefügt wird, die vom Tabellenbereich „plan“ auf einem UNIX-basiertem Basisbetriebssystem verwendet wird:

```
ALTER TABLESPACE plan
  ADD ('/dev/rhdisk0')
  ON NODE (3)
```

Umbenennen eines Tabellenbereichs

Sie können einem vorhandenen Tabellenbereich einen neuen Namen zuordnen, ohne dass sich dies auf die einzelnen Objekte innerhalb des Tabellenbereichs auswirkt. Beim Umbenennen eines Tabellenbereichs werden alle Katalogsätze, die auf diesen Tabellenbereich verweisen, geändert.

Der Tabellenbereich SYSCATSPACE kann nicht umbenannt werden.

Tabellenbereiche, die sich im Status „Aktualisierende Wiederherstellung anstehend“ oder „Aktualisierende Wiederherstellung wird ausgeführt“ befinden, können nicht umbenannt werden.

Beim Zurückschreiben eines Tabellenbereichs, der seit seiner Sicherung umbenannt wurde, müssen Sie im Befehl RESTORE DATABASE den neuen Tabellenbereichsnamen verwenden. Wenn Sie den vorherigen Tabellenbereichsnamen benutzen, kann der Tabellenbereich nicht lokalisiert werden. Wenn Sie für den Tabellenbereich mit dem Befehl ROLLFORWARD DATABASE eine aktualisierende Wiederherstellung durchführen, müssen Sie ebenfalls den neuen Namen verwenden. Wenn Sie den vorherigen Tabellenbereichsnamen benutzen, kann der Tabellenbereich nicht lokalisiert werden.

Ändern des Status eines Tabellenbereichs

Die Klausel SWITCH ONLINE der Anweisung ALTER TABLESPACE kann verwendet werden, um den Status OFFLINE von einem Tabellenbereich zu entfernen, wenn auf die Behälter, die diesem Tabellenbereich zugeordnet sind, zugegriffen werden kann. Der Status OFFLINE wird vom Tabellenbereich entfernt, während der Rest der Datenbank weiterhin aktiv ist und verwendet wird.

Eine Alternative zur Verwendung dieser Klausel ist das Trennen aller Anwendungen von der Datenbank und das anschließende erneute Verbinden der Anwendungen mit der Datenbank. Dadurch wird der Status OFFLINE vom Tabellenbereich entfernt.

Geben Sie in der Befehlszeile Folgendes ein, um den Status OFFLINE vom Tabellenbereich zu entfernen.

```
ALTER TABLESPACE <name>
  SWITCH ONLINE
```

Löschen eines Benutzertabellenbereichs

Beim Löschen eines Benutzertabellenbereichs werden alle Daten in diesem Tabellenbereich gelöscht, die Behälter freigegeben, die Katalogeinträge entfernt und alle Objekte, die in dem Tabellenbereich definiert sind, entweder gelöscht oder als ungültig markiert.

Die Behälter in einem leeren Tabellenbereich können erneut verwendet werden, indem der Tabellenbereich gelöscht wird. Allerdings muss der Befehl `DROP TABLESPACE` mit `COMMIT` festgeschrieben werden, bevor Sie versuchen, die Behälter erneut zu verwenden.

Sie können einen Benutzertabellenbereich löschen, der sämtliche Tabellendaten einschließlich Index- und LOB-Daten dieses einen Benutzertabellenbereichs enthält. Sie können auch einen Benutzertabellenbereich löschen, der Tabellen beinhaltet, die sich über mehrere Tabellenbereiche erstrecken. Dabei können sich die Tabellendaten in einem Tabellenbereich, die Indizes in einem anderen Tabellenbereich und die LOB-Daten in einem dritten Tabellenbereich befinden. Sie müssen alle drei Tabellenbereiche gleichzeitig mit einer einzigen Anweisung löschen. Wenn sich Tabellen über mehrere Tabellenbereiche erstrecken, muss die Löschanweisung alle Tabellenbereiche enthalten, auf die sich diese Tabellen erstrecken, sonst kann sie nicht erfolgreich ausgeführt werden. Einzelheiten zum Löschen von Tabellenbereichen, die verteilte Tabellendaten enthalten, finden Sie im Handbuch *SQL Reference*.

Gehen Sie wie folgt vor, um mit der Steuerzentrale einen Benutzertabellenbereich zu löschen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellenbereiche** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Tabellenbereich, der gelöscht werden soll, und wählen Sie dann im Kontextmenü **Löschen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um einen Benutzertabellenbereich zu löschen:

```
DROP TABLESPACE <name>
```

Mit der folgenden SQL-Anweisung wird der Tabellenbereich `ACCOUNTING` gelöscht:

```
DROP TABLESPACE ACCOUNTING
```

Löschen eines temporären Systemtabellenbereichs

Sie können einen temporären Systemtabellenbereich nur löschen, wenn Sie zuvor einen anderen temporären Systemtabellenbereich erstellt haben, da die

Datenbank immer über mindestens einen temporären Systemtabellenbereich verfügen muss. Wenn Sie zum Beispiel einem temporären SMS-Tabellenbereich einen Behälter hinzufügen möchten, müssen Sie zuerst einen neuen temporären Systemtabellenbereich hinzufügen und anschließend den alten temporären Systemtabellenbereich löschen.

Gehen Sie wie folgt vor, um mit der Steuerzentrale einen Systemtabellenbereich zu löschen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellenbereiche** angezeigt wird.
2. Wenn nur ein temporärer Systemtabellenbereich definiert ist, klicken Sie mit der rechten Maustaste auf dem Ordner **Tabellenbereiche** und wählen Sie dann im Kontextmenü **Erstellen** → **Tabellenbereich mit Assistent** aus. Andernfalls können Sie sofort bei Schritt 4 weiterarbeiten.
3. Führen Sie die im Assistenten aufgeführten Schritte aus, um ggf. den neuen temporären Systemtabellenbereich zu erstellen.
4. Klicken Sie erneut auf dem Ordner **Tabellenbereiche**, um eine Liste der Tabellenbereiche rechts im Fenster (Inhaltsteilfenster) anzuzeigen.
5. Klicken Sie mit der rechten Maustaste auf dem temporären Systemtabellenbereich, der gelöscht werden soll, und dann im Kontextmenü auf **Löschen**.
6. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie anschließend auf **OK**.

Wenn Sie nur über einen temporären Systemtabellenbereich verfügen, müssen Sie vor dem Löschen dieses Tabellenbereichs einen neuen erstellen. Hierzu können Sie in der Befehlszeile Folgendes eingeben:

```
CREATE SYSTEM TEMPORARY TABLESPACE <name>  
    MANAGED BY SYSTEM USING ('<einheit>')
```

Geben Sie anschließend zum Löschen eines Systemtabellenbereichs in der Befehlszeile Folgendes ein:

```
DROP TABLESPACE <name>
```

Mit der folgenden SQL-Anweisung können Sie einen neuen temporären Systemtabellenbereich mit dem Namen TEMPSPACE2 erstellen:

```
CREATE SYSTEM TEMPORARY TABLESPACE TEMPSPACE2  
    MANAGED BY SYSTEM USING ('d')
```

Nach der Erstellung von TEMPSPACE2 können Sie den ursprünglichen temporären Systemtabellenbereich TEMPSPACE1 mit dem folgenden Befehl löschen:

```
DROP TABLESPACE TEMPSPACE1
```

Die Behälter in einem leeren Tabellenbereich können erneut verwendet werden, indem der Tabellenbereich gelöscht wird. Allerdings muss der Befehl DROP TABLESPACE mit COMMIT festgeschrieben werden, bevor Sie versuchen, die Behälter erneut zu verwenden.

Löschen eines temporären Benutzertabellenbereichs

Ein temporärer Benutzertabellenbereich kann nur gelöscht werden, wenn in diesem Tabellenbereich momentan keine deklarierten temporären Tabellen definiert sind. Wenn Sie den Tabellenbereich löschen, wird nicht versucht, alle deklarierten temporären Tabellen dieses Tabellenbereichs zu löschen.

Anmerkung: Eine deklarierte temporäre Tabelle wird implizit gelöscht, wenn die Anwendung, die zur Deklaration verwendet wurde, die Verbindung zur Datenbank trennt.

Löschen eines Schemas

Bevor Sie ein Schema löschen, müssen alle Objekte, die sich in diesem Schema befinden, selbst gelöscht oder in ein anderes Schema versetzt werden. Der Schemenname muss im Katalog vorhanden sein, wenn die Anweisung DROP ausgeführt wird. Ansonsten wird ein Fehler zurückgegeben.

Gehen Sie wie folgt vor, um mit der Steuerzentrale ein Schema zu löschen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Schemata** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Schema, das gelöscht werden soll, und wählen Sie dann im Kontextmenü **Löschen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um ein Schema zu löschen:

```
DROP SCHEMA <name>
```

Im folgenden Beispiel wird das Schema „joeschma“ gelöscht:

```
DROP SCHEMA joeschma RESTRICT
```

Das Schlüsselwort RESTRICT erzwingt die Anwendung der Regel, dass im angegebenen Schema keine Objekte für das Schema definiert werden können, das aus der Datenbank gelöscht werden soll.

Ändern von Struktur und Inhalt einer Tabelle

Zum Modifizieren von Struktur und Inhalt einer Tabelle sind unter anderem folgende Tasks erforderlich:

- „Hinzufügen von Spalten in einer vorhandenen Tabelle“ auf Seite 219
- „Modifizieren einer Spaltendefinition“ auf Seite 220

- „Entfernen von Zeilen aus einer Tabelle oder Sicht“ auf Seite 221
- „Ändern einer Integritätsbedingung“ auf Seite 222
- „Definieren einer generierten Spalte in einer vorhandenen Tabelle“ auf Seite 229
- „Deklarieren einer flüchtigen Tabelle“ auf Seite 233
- „Ändern von Partitionierungsschlüsseln“ auf Seite 234
- „Ändern von Tabellenattributen“ auf Seite 235
- „Aktualisieren der Daten in einer Übersichtstabelle“ auf Seite 238

Beachten Sie, dass Sie keine Auslöser für Tabellen ändern können. Sie müssen einen Auslöser, der nicht mehr geeignet ist, löschen (siehe „Löschen eines Auslösers“ auf Seite 242) und den neuen Auslöser hinzufügen (siehe „Erstellen eines Auslösers“ auf Seite 156).

Hinzufügen von Spalten in einer vorhandenen Tabelle

Zu einer Spaltendefinition gehören ein Spaltenname, ein Datentyp und alle nötigen Integritätsbedingungen.

Wenn einer Tabelle Spalten hinzugefügt werden, werden die Spalten logisch rechts von der am weitesten rechts stehenden vorhandenen Spaltendefinition positioniert. Wenn eine neue Spalte einer vorhandenen Tabelle hinzugefügt wird, wird nur die Tabellenbeschreibung im Systemkatalog geändert, so dass die Zugriffszeit für die Tabelle nicht sofort betroffen ist. Vorhandene Datensätze werden physisch nicht geändert. Sie werden erst geändert, wenn eine Anweisung UPDATE ausgeführt wird. Beim Abrufen einer vorhandenen Zeile aus der Tabelle wird für die neue Spalte ein Nullwert oder der Standardwert bereitgestellt, je nachdem, wie die neue Spalte definiert wurde. Spalten, die nach Erstellung einer Tabelle hinzugefügt werden, können nicht mit NOT NULL definiert werden: Sie müssen entweder mit NOT NULL WITH DEFAULT oder mit der Angabe, dass Nullwerte möglich sind, definiert werden.

Gehen Sie wie folgt vor, um in der Steuerzentrale Spalten zu einer vorhandenen Tabelle hinzuzufügen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Tabelle, zu der Spalten hinzugefügt werden sollen, und wählen Sie dann im Kontextmenü **Ändern** aus.
3. Prüfen Sie die Seite **Spalten** und geben Sie die Informationen für die Spalte ein. Klicken Sie dann auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um Spalten zu einer vorhandenen Tabelle hinzuzufügen:

```
ALTER TABLE <tabellenname>  
  ADD <spaltenname> <datentyp> <>nullattribut>
```

Spalten können mit einer SQL-Anweisung hinzugefügt werden. Im folgenden Beispiel wird die Anweisung ALTER TABLE verwendet, um der Tabelle EMPLOYEE drei Spalten hinzuzufügen:

```
ALTER TABLE EMPLOYEE  
  ADD MIDINIT CHAR(1) NOT NULL WITH DEFAULT  
  ADD HIREDATE DATE  
  ADD WORKDEPT CHAR(3)
```

Modifizieren einer Spaltendefinition

Sie können die Merkmale einer Spalte durch Erhöhen der Länge einer vorhandenen VARCHAR-Spalte ändern. Die Anzahl der Zeichen kann bis zu einem Maximalwert erhöht werden, der von der verwendeten Seitengröße abhängt.

Gehen Sie wie folgt vor, um in der Steuerzentrale Spalten einer vorhandenen Tabelle zu modifizieren:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie in der Liste der Tabellen im rechten Teilfenster mit der rechten Maustaste auf der Tabelle, in der eine Spalte modifiziert werden soll und wählen Sie dann im Kontextmenü **Ändern** aus.
3. Prüfen Sie die Seite **Spalten**, wählen Sie die gewünschte Spalte aus und klicken Sie dann auf **Ändern**.
4. Geben Sie unter **Länge** die neue Byteanzahl für die Spalte ein und klicken Sie dann auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um Spalten einer vorhandenen Tabelle zu modifizieren:

```
ALTER TABLE ALTER COLUMN  
  <spaltenname> <modifikationstyp>
```

Mit der folgenden Anweisung können Sie zum Beispiel die Zeichenzahl einer Spalte auf bis zu 4.000 Zeichen erhöhen.

```
ALTER TABLE ALTER COLUMN  
  COLNAM1 SET DATA TYPE VARCHAR(4000)
```


Sie können die Spalte einer typisierten Tabelle nicht ändern. Sie können einer vorhandenen Verweisartspalte ohne definierten Bereich jedoch einen Bereich hinzufügen. Beispiel:

```
ALTER TABLE ALTER COLUMN  
COLNAMT1 ADD SCOPE TYPTAB1
```

Weitere Informationen zur Anweisung ALTER TABLE finden Sie im Handbuch *SQL Reference*.

Entfernen von Zeilen aus einer Tabelle oder Sicht

Sie können den Inhalt einer Tabelle oder einer Sicht ändern, indem Sie Zeilen löschen. Das Löschen einer Zeile aus einer Sicht hat das Löschen der Zeilen aus der Tabelle, auf der die Sicht basiert, zur Folge. Die Anweisung DELETE wird für folgende Zwecke verwendet:

- Löschen einer oder mehrerer Zeilen, die optional von einer Suchbedingung ermittelt wurden. Dies wird als *gezielter Löschvorgang* bezeichnet.
- Löschen genau einer Zeile, die durch die aktuelle Position eines Cursors ermittelt wurde. Dies wird als *positionierter Löschvorgang* bezeichnet.

Die Anweisung DELETE kann in ein Anwendungsprogramm eingebettet oder als dynamische SQL-Anweisung abgesetzt werden.

Wenn die geänderte Tabelle über referenzielle Integritätsbedingungen mit anderen Tabellen zusammenhängt, muss beim Löschen von Zeilen Folgendes bedacht werden. Wenn die angegebene Tabelle oder die Basistabelle der angegebenen Sicht ein Elter ist, dürfen die zum Löschen ausgewählten Zeilen keine abhängigen Elemente in einer Beziehung mit einer Löschbedingung RESTRICT aufweisen. Ferner darf DELETE sich nicht auf untergeordnete Zeilen auswirken, die abhängige Elemente in einer Beziehung mit einer Löschbedingung RESTRICT aufweisen.

Wenn die Löschoperation nicht von einer Löschbedingung RESTRICT verhindert wird, werden die ausgewählten Zeilen gelöscht. Weitere Informationen dazu, was mit Zeilen geschieht, die von ausgewählten Zeilen abhängig sind, finden Sie in *SQL Reference*.

Geben Sie zum Beispiel Folgendes ein, um die Abteilung (DEPTNO) „D11“ aus der Tabelle (DEPARTMENT) zu löschen:

```
DELETE FROM department WHERE deptno='D11'
```

Wenn bei der Ausführung von DELETE für mehrere Zeilen ein Fehler auftritt, werden keine Änderungen an der Tabelle vorgenommen. Wenn ein Fehler auftritt, der das Löschen aller Zeilen, die der Suchbedingung entsprechen, und aller durch vorhandene referenzielle Integritätsbedingungen erforderlichen Operationen verhindert, werden keine Änderungen an der Tabelle vorgenommen.

Wenn noch keine entsprechenden Sperren bestehen, werden während der erfolgreichen Ausführung der Anweisung DELETE eine oder mehrere exklusive Sperren eingerichtet. Sperren werden nach einer Anweisung COMMIT oder ROLLBACK freigegeben. Sperren können verhindern, dass andere Anwendungen Operationen an der Tabelle ausführen.

Modifizieren einer Identitätsspaltendefinition

Wenn Sie eine Tabelle mit einer Identitätsspalte erneut erstellen und danach eine Import- oder Ladeoperation durchgeführt wird, so wird die Identitätsspalte auf den Anfang zurückgesetzt und der Identitätswert 1 wird nach der Neuerstellung des Inhalts der Tabelle generiert. Wenn in diese erneut erstellte Tabelle neue Zeilen eingefügt werden, soll die Identitätsspalte nicht wieder mit 1 beginnen. Sie möchten keine doppelten Werte in der Identitätsspalte haben. Sie sollten folgende Schritte ausführen, um dies zu verhindern:

1. Erstellen Sie die Tabelle erneut.
2. Laden Sie die Daten mit der Klausel MODIFIED BY IDENTITYOVERRIDE in die Tabelle. Die Daten werden in die Tabelle geladen, es werden jedoch keine Identitätswerte für die Zeilen generiert.
3. Führen Sie eine Abfrage aus, um den letzten Zählerwert für die Identitätsspalte zu erhalten:

```
SELECT MAX(<identitätsspalte>)
```

Mit diesem Befehl wird der äquivalente Wert des Wertes zurückgegeben, der der Wert für die Identitätsspalte in der Tabelle gewesen wäre.

4. Verwenden Sie die Klausel RESTART der Anweisung ALTER TABLE:

```
ALTER TABLE <tabellenname> ALTER COLUMN <identitätsspalte>  
RESTART WITH <letzter Zählerwert>
```
5. Fügen Sie eine neue Zeile in der Tabelle ein. Der Wert für die Identitätsspalte wird auf Basis des Wertes generiert, der in der Klausel RESTART WITH angegeben wurde.

Ändern einer Integritätsbedingung

Integritätsbedingungen können nur durch Löschen und Hinzufügen neuer Bedingungen an ihrer Stelle geändert werden. Weitere Informationen hierzu enthalten die folgenden Abschnitte:

- „Hinzufügen einer Integritätsbedingung“ auf Seite 223
- „Löschen einer Integritätsbedingung“ auf Seite 227

Weitere Informationen zu Integritätsbedingungen finden Sie in „Definieren von Integritätsbedingungen“ auf Seite 140.

Hinzufügen einer Integritätsbedingung

Integritätsbedingungen werden mit der Anweisung ALTER TABLE hinzugefügt. Weitere Informationen zu dieser Anweisung, einschließlich Syntax, finden Sie im Handbuch *SQL Reference*.

Weitere Informationen zu Integritätsbedingungen finden Sie in „Definieren von Integritätsbedingungen“ auf Seite 140.

Hinzufügen einer eindeutigen Integritätsbedingung: Eindeutige Integritätsbedingungen können einer vorhandenen Tabelle hinzugefügt werden. Der Name der Integritätsbedingung darf nicht mit dem Namen einer anderen Integritätsbedingung innerhalb dieser Anweisung ALTER TABLE übereinstimmen, und muss innerhalb der Tabelle eindeutig sein (dies schließt auch Namen bereits definierter referenzieller Integritätsbedingungen mit ein). Vorhandene Daten werden anhand der neuen Bedingung überprüft, bevor die Anweisung erfolgreich beendet wird.

Mit der folgenden SQL-Anweisung wird der Tabelle EMPLOYEE eine eindeutige Integritätsbedingung hinzugefügt, die eine neue Methode zur eindeutigen Identifizierung von Mitarbeitern in der Tabelle darstellt:

```
ALTER TABLE EMPLOYEE
  ADD CONSTRAINT NEWID UNIQUE(EMPNO,HIREDATE)
```

Hinzufügen von Primär- und Fremdschlüsseln: Beim Hinzufügen von Integritätsbedingungen für umfangreiche Tabellen ist es effizienter, die Tabelle in den Status „Überprüfung anstehend“ (Check pending) zu setzen, die Integritätsbedingungen hinzuzufügen und anschließend die Tabelle überprüfen und eine konsolidierte Liste der die Bedingungen verletzenden Zeilen erstellen zu lassen. Verwenden Sie die Anweisung SET INTEGRITY, um den Status „Überprüfung anstehend“ explizit festzulegen: Ist die Tabelle eine übergeordnete Tabelle, wird der Status „Überprüfung anstehend“ implizit auch für alle abhängigen und untergeordneten Tabellen aktiviert.

Gehen Sie wie folgt vor, um mit der Steuerzentrale Primärschlüssel hinzuzufügen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Tabelle, die modifiziert werden soll, und wählen Sie dann im Kontextmenü **Ändern** aus.
3. Wählen Sie auf der Seite **Primärschlüssel** eine oder mehrere Spalten als Primärschlüssel aus und klicken Sie dann auf den Pfeil, um sie zu versetzen.
4. Wahlfrei: Geben Sie den Integritätsbedingungsnamen des Primärschlüssels ein.
5. Klicken Sie auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um Primärschlüssel hinzuzufügen:

```
ALTER TABLE <name>
  ADD CONSTRAINT <spaltenname>
  PRIMARY KEY <spaltenname>
```

Wenn ein Fremdschlüssel einer Tabelle hinzugefügt wird, werden Pakete und im Cache zwischengespeicherte dynamische SQL-Anweisungen, die folgende Arten von Anweisungen enthalten, eventuell als ungültig markiert:

- Anweisungen, die die Tabelle mit dem Fremdschlüssel aktualisieren oder dort Zeilen einfügen
- Anweisungen, die die übergeordnete Tabelle aktualisieren oder löschen

Weitere Informationen finden Sie in „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 253.

Gehen Sie wie folgt vor, um mit der Steuerzentrale Fremdschlüssel hinzuzufügen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Tabelle, die modifiziert werden soll, und wählen Sie dann im Kontextmenü **Ändern** aus.
3. Klicken Sie auf der Seite **Fremdschlüssel** auf **Hinzufügen**.
4. Geben Sie im Fenster **Fremdschlüssel hinzufügen** die Informationen zur übergeordneten Tabelle ein.
5. Wählen Sie eine oder mehrere Spalten als Fremdschlüssel aus und klicken Sie auf den Pfeil, um sie zu versetzen.
6. Geben Sie an, welche Aktion für die abhängige Tabelle ausgeführt werden soll, wenn in der übergeordneten Tabelle eine Zeile gelöscht oder aktualisiert wird. Sie können auch einen Integritätsbedingungsnamen für den Fremdschlüssel hinzufügen.
7. Klicken Sie auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um Fremdschlüssel hinzuzufügen:

```
ALTER TABLE <name>
  ADD CONSTRAINT <spaltenname>
    FOREIGN KEY <spaltenname>
    ON DELETE <aktionsart>
    ON UPDATE <aktionsart>
```

Die folgenden Beispiele zeigen die Verwendung der Anweisung ALTER TABLE zum Hinzufügen von Primär- und Fremdschlüsseln in einer Tabelle:

```
ALTER TABLE PROJECT
  ADD CONSTRAINT PROJECT_KEY
    PRIMARY KEY (PROJNO)
ALTER TABLE EMP_ACT
  ADD CONSTRAINT ACTIVITY_KEY
    PRIMARY KEY (EMPNO, PROJNO, ACTNO)
  ADD CONSTRAINT ACT_EMP_REF
    FOREIGN KEY (EMPNO)
    REFERENCES EMPLOYEE
    ON DELETE RESTRICT
  ADD CONSTRAINT ACT_PROJ_REF
    FOREIGN KEY (PROJNO)
    REFERENCES PROJECT
    ON DELETE CASCADE
```

Hinzufügen von Prüfungen auf Integritätsbedingungen in Tabellen: Prüfungen auf Integritätsbedingungen können einer existierenden Tabelle mit Hilfe der Anweisung ALTER TABLE hinzugefügt werden. Der Name der Integritätsbedingung darf nicht mit dem Namen einer anderen Integritätsbedingung innerhalb dieser Anweisung ALTER TABLE übereinstimmen, und muss innerhalb der Tabelle eindeutig sein (dies schließt auch Namen bereits definierter referenzieller Integritätsbedingungen mit ein). Vorhandene Daten werden anhand der neuen Bedingung überprüft, bevor die Anweisung erfolgreich beendet wird.

Beim Hinzufügen von Integritätsbedingungen für umfangreiche Tabellen ist es effizienter, die Tabelle in den Status „Überprüfung anstehend“ (Check pending) zu setzen, die Integritätsbedingungen hinzuzufügen und anschließend die Tabelle überprüfen und eine konsolidierte Liste der die Bedingungen verletzenden Zeilen erstellen zu lassen. Verwenden Sie die Anweisung SET INTEGRITY, um den Status „Überprüfung anstehend“ explizit festzulegen: Ist die Tabelle eine übergeordnete Tabelle, wird der Status „Überprüfung anstehend“ implizit auch für alle abhängigen und untergeordneten Tabellen aktiviert.

Wenn eine Prüfung auf Integritätsbedingung in einer Tabelle hinzugefügt wird, werden Pakete und im Cache zwischengespeicherte dynamische SQL-Anweisungen, die INSERT- oder UPDATE-Operationen für die Tabelle ausführen, eventuell als ungültig markiert. Der Abschnitt „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 253 enthält weitere Informationen.

Gehen Sie wie folgt vor, um mit der Steuerzentrale eine Prüfung auf Integritätsbedingungen in Tabellen hinzuzufügen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Tabelle, die modifiziert werden soll, und wählen Sie dann im Kontextmenü **Ändern** aus.
3. Klicken Sie auf der Seite **Prüfung auf Integritätsbedingung** auf **Hinzufügen**.
4. Geben Sie im Fenster **Prüfung auf Integritätsbedingung hinzufügen** die erforderlichen Informationen ein und klicken Sie dann auf **OK**.
5. Klicken Sie auf der Seite **Prüfung auf Integritätsbedingung** auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Prüfung auf Integritätsbedingungen in Tabellen hinzuzufügen:

```
ALTER TABLE <name>  
ADD CONSTRAINT <name> (<integritätsbedingung>)
```

Mit der folgenden SQL-Anweisung wird der Tabelle EMPLOYEE eine Integritätsbedingung hinzugefügt, die festlegt, dass das Gehalt (SALARY) plus Provision (COMM) für jeden Mitarbeiter mehr als 25.000 US-Dollar betragen muss:

```
ALTER TABLE EMPLOYEE
  ADD CONSTRAINT REVENUE CHECK (SALARY + COMM > 25000)
```

Löschen einer Integritätsbedingung

Integritätsbedingungen werden mit der Anweisung ALTER TABLE gelöscht. Weitere Informationen zu dieser Anweisung, einschließlich Syntax, finden Sie im Handbuch *SQL Reference*.

Weitere Informationen zu Integritätsbedingungen finden Sie in „Definieren von Integritätsbedingungen“ auf Seite 140.

Löschen einer eindeutigen Integritätsbedingung: Sie können eine eindeutige Integritätsbedingung explizit mit der Anweisung ALTER TABLE löschen. Die Namen aller eindeutigen Integritätsbedingungen für eine Tabelle befinden sich in der Systemkatalogsicht SYSCAT.INDEXES.

Mit der folgenden SQL-Anweisung wird die eindeutige Integritätsbedingung NEWID aus der Tabelle EMPLOYEE gelöscht:

```
ALTER TABLE EMPLOYEE
  DROP UNIQUE NEWID
```

Durch das Löschen dieser eindeutigen Integritätsbedingung werden alle Pakete und im Cache zwischengespeicherte dynamische SQL-Anweisungen ungültig gemacht, die diese Integritätsbedingung verwendet haben.

Löschen von Primär- und Fremdschlüsseln: Gehen Sie wie folgt vor, um mit der Steuerzentrale einen Primärschlüssel zu löschen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Tabelle, die modifiziert werden soll, und wählen Sie dann im Kontextmenü **Ändern** aus.
3. Wählen Sie auf der Seite **Primärschlüssel** rechts den zu löschenden Primärschlüssel aus und klicken Sie dann auf den Pfeil, um ihn in das Feld **Verfügbare Spalten** auf der linken Seite des Fensters zu versetzen.
4. Klicken Sie auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um einen Primärschlüssel zu löschen:

```
ALTER TABLE <name>
  DROP PRIMARY KEY
```

Wenn eine Integritätsbedingung über Fremdschlüssel gelöscht wird, werden Pakete oder im Cache zwischengespeicherte SQL-Anweisungen, die folgende Arten von Anweisungen enthalten, eventuell als ungültig markiert:

- Anweisungen, die die Tabelle mit dem Fremdschlüssel aktualisieren oder dort Zeilen einfügen
- Anweisungen, die die übergeordnete Tabelle aktualisieren oder löschen

Der Abschnitt „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 253 enthält weitere Informationen.

Gehen Sie wie folgt vor, um mit der Steuerzentrale einen Fremdschlüssel zu löschen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Tabelle, die modifiziert werden soll, und wählen Sie dann im Kontextmenü **Ändern** aus.
3. Klicken Sie auf der Seite **Fremdschlüssel** auf **Hinzufügen**.
4. Wählen Sie rechts im Fenster den zu löschenden Fremdschlüssel aus und klicken Sie dann auf den Pfeil, um ihn in das Feld **Verfügbare Spalten** auf der linken Seite des Fensters zu versetzen.
5. Klicken Sie auf der Seite **Fremdschlüssel** auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um einen Fremdschlüssel zu löschen:

```
ALTER TABLE <name>
  DROP FOREIGN KEY <fremdschlüsselname>
```

Im folgenden Beispiel werden die Klauseln DROP PRIMARY KEY und DROP FOREIGN KEY in der Anweisung ALTER TABLE verwendet, um Primär- und Fremdschlüssel zu löschen:

```
ALTER TABLE EMP_ACT
  DROP PRIMARY KEY
  DROP FOREIGN KEY ACT_EMP_REF
  DROP FOREIGN KEY ACT_PROJ_REF
ALTER TABLE PROJECT
  DROP PRIMARY KEY
```

Weitere Informationen zur Anweisung ALTER TABLE finden Sie im Handbuch *SQL Reference*.

Löschen von Prüfungen auf Integritätsbedingungen in Tabellen: Eine Prüfung auf Integritätsbedingung in einer Tabelle (Table Check Constraint) kann explizit mit Hilfe einer Anweisung ALTER TABLE gelöscht oder geändert werden, oder implizit als Ergebnis einer Anweisung DROP TABLE gelöscht werden.

Durch das Löschen einer Prüfung auf Integritätsbedingung in einer Tabelle werden alle Pakete und im Cache zwischengespeicherte dynamische SQL-Anweisungen mit INSERT- oder UPDATE-Abhängigkeiten für die Tabelle ungültig gemacht. (Weitere Informationen dazu finden Sie in „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 253.) Die Namen aller Prüfungen auf Integritätsbedingungen in einer Tabelle können mit Hilfe der Katalogsicht SYSCAT.CHECKS festgestellt werden. Vor dem Löschen einer Prüfung auf Integritätsbedingungen in einer Tabelle, die einen vom System generierten Namen hat, können Sie den Namen aus der Katalogsicht SYSCAT.CHECKS abfragen.

Gehen Sie wie folgt vor, um mit der Steuerzentrale eine Prüfung auf Integritätsbedingungen in Tabellen zu löschen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Tabelle, die modifiziert werden soll, und wählen Sie dann im Kontextmenü **Ändern** aus.
3. Wählen Sie auf der Seite **Prüfung auf Integritätsbedingung** die zu löschende Prüfung auf Integritätsbedingung aus. Klicken Sie dann auf **Entfernen** und anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Prüfung auf Integritätsbedingungen in Tabellen zu löschen:

```
ALTER TABLE <tabellenname>  
DROP CHECK <name_der_prüfung_auf_integritätsbedingung>
```

Mit der folgenden SQL-Anweisung wird die Prüfung auf Integritätsbedingung REVENUE aus der Tabelle EMPLOYEE gelöscht:

```
ALTER TABLE EMPLOYEE  
DROP CHECK REVENUE
```

Definieren einer generierten Spalte in einer vorhandenen Tabelle

Eine generierte Spalte wird in einer Basistabelle definiert, bei der der gespeicherte Wert mit Hilfe eines Ausdrucks ermittelt wird, anstatt mit einer Einfüge- oder Aktualisierungsoperation angegeben zu werden. Eine generierte Spalte kann erstellt werden, wenn eine Tabelle erstellt oder eine bereits vorhandene Tabelle modifiziert wird.

Führen Sie die folgenden Schritte aus, um eine generierte Spalte zu definieren:

1. Versetzen Sie die Tabelle in den Status 'Überprüfung anstehend'.

```
SET INTEGRITY FOR t1 OFF
```

2. Ändern Sie die Tabelle und fügen Sie eine oder mehrere generierte Spalten hinzu.

```
ALTER TABLE t1 ADD COLUMN c3 DOUBLE GENERATED ALWAYS AS (c1 + c2),  
ADD COLUMN c4 GENERATED ALWAYS AS  
(CASE WHEN c1 > c3 THEN 1 ELSE NULL END))
```

3. Zur Ausführung dieser Aufgabe gibt es mehrere Verfahren, deren Einsatz von den für die Tabelle auszuführenden Arbeitsschritten abhängt:

- Die Tabelle ist sehr umfangreich und Sie haben Zweifel, ob genügend Speicherbereich für die Protokollierung zur Verfügung steht, um die Task abzuschließen. Nach dem Laden der Daten, jedoch bevor Sie die Integritätsprüfung erneut aktivieren, muss der folgende Befehl ausgeführt werden:

```
COMMIT
```

Anschließend müssen Sie das Dienstprogramm `db2gncol` verwenden, um die Werte für die generierten Spalten festzulegen. Dieses Dienstprogramm ist im Verzeichnis `sql1ib` im Unterverzeichnis `bin` gespeichert. Verwenden Sie das Dienstprogramm folgendermaßen:

```
db2gncol -d <dbname> -s <schema> -t <tabellenname>  
-c <commit_zähler>
```

Die Variable `dbname` gibt hierbei einen Aliasnamen für die Datenbank an, in der sich die Tabelle befindet. Die Variable `schema` steht für den Schemennamen der Tabelle. Bei der Eingabe dieses Namens muss die Groß-/Kleinschreibung beachtet werden. Die Variable `tabellenname` gibt die Tabelle an, für die neue Werte für die mit Hilfe entsprechender Ausdrücke generierten Spalten berechnet werden müssen. Sowohl bei der Eingabe von Werten für `schema` als auch für `tabellenname` muss die Groß-/Kleinschreibung beachtet werden. Die Variable `commit_zähler` gibt die Anzahl der Zeilen an, die zwischen zwei internen `COMMIT`-Operationen zur Bereinigung der Protokolle verarbeitet werden sollen. Dieser Parameter beeinflusst die Größe des Speicherplatzes für die Protokollierung, der für das Generieren der Spaltenwerte benötigt wird.

Es gibt außerdem zwei wahlfreie Parameter, die im o. a. Beispiel nicht enthalten sind. Die Parameter `-u <benutzername>` und `-p <kennwort>` dienen zur Identifizierung des Benutzers sowie des zugehörigen Kennworts. Der Benutzer muss über die Berechtigung `SYSADM` oder `DBADM` verfügen. Wurden kein Benutzer und kein Kennwort identifiziert, wird die aktuelle Benutzer-ID verwendet.

Geben Sie Folgendes ein, um den Hilfetext für dieses Dienstprogramm aufzurufen:

```
db2gncol -h
```

Wenn der Hilfeparameter verwendet wird, werden alle anderen Parameter ignoriert.

Die Tabelle wird für den gesamten Prozess gesperrt, obwohl sie sich im Status 'Überprüfung anstehend' befindet. Die Tabelle wird gesperrt, weil auch andere Dienstprogramme, die auf Tabellen zugreifen können, sich im Status 'Überprüfung anstehend' befinden. Durch die Sperre können Konflikte mit diesen anderen Dienstprogrammen vermieden werden.

- Sie gehen davon aus, dass der Speicherbereich zum Protokollieren für die Aktualisierung der generierten Spalten voraussichtlich für SET INTEGRITY ausreicht. Dies ist normalerweise der Fall. Nach dem Laden der Daten berechnen Sie die Werte und ordnen diese für die generierten Spalten erneut zu. Verwenden Sie hierzu den folgenden Befehl:

```
SET INTEGRITY FOR t1 IMMEDIATE CHECKED  
FORCE GENERATED
```

Anmerkung: Zu diesem Zeitpunkt können auch Ausnahmetabellen verwendet werden.

- Die Tabelle ist sehr umfangreich und Sie haben Zweifel, ob genügend Speicherbereich für die Protokollierung zur Verfügung steht, um die Task abzuschließen. Sie haben sich nicht für die Auswahl der ersten, o. a. Methode entschieden. Nach dem Laden der Daten, jedoch bevor Sie die Integritätsprüfung erneut aktivieren, muss der folgende Befehl ausgeführt werden:
 - a. Setzen einer exklusiven Sperre für die Tabelle. Hierdurch werden für die Tabelle nur noch nicht festgeschriebene Lesevorgänge zugelassen.

```
LOCK TABLE t1
```
 - b. Versetzen der Tabelle in den Online-Status, wobei keine Datenprüfung ausgeführt wird.

```
SET INTEGRITY FOR t1 ALL IMMEDIATE UNCHECKED
```
 - c. Aktualisieren der generierten Spalten mit temporären COMMIT-Operationen und Prädikaten, um ein Auffüllen der Protokolle zu vermeiden.

```
UPDATE t1 SET (c3, c4) = (DEFAULT, DEFAULT) WHERE <prädikat>
```
 - d. Versetzen der Tabelle in den Online-Status und Prüfen ihrer Integrität.

```
SET INTEGRITY FOR t1 OFF  
SET INTEGRITY FOR t1 IMMEDIATE CHECKED
```

- e. Entsperrten der Tabelle durch Abschluss der Transaktion mit einer COMMIT-Anweisung.

COMMIT

- Sie wissen, dass die Tabelle mit der Option NOT LOGGED INITIALLY erstellt wurde. Auf diese Weise wurde das Protokollieren für die Tabelle während des Arbeitens mit den Werten der generierten Spalten inaktiviert. Hierbei sind die üblichen Konsequenzen und Risiken zu berücksichtigen.

- a. Aktivieren Sie die Option NOT LOGGED INITIALLY.

```
ALTER TABLE t1 ACTIVATE NOT LOGGED INITIALLY
```

- b. Generieren Sie die Werte.

```
SET INTEGRITY FOR t1 IMMEDIATE CHECKED FORCE GENERATED
```

- c. Inaktivieren Sie die Option NOT LOGGED INITIALLY, indem Sie die Transaktion festschreiben.

```
COMMIT
```

Die Werte für die generierten Spalten können auch einfach geprüft werden, indem der Ausdruck so angewendet wird, als ob es sich um eine Gleichheitsprüfung auf Integritätsbedingungen handelt:

```
SET INTEGRITY FOR t1 IMMEDIATE CHECKED
```

Wenn die Werte z. B. mit LOAD in einer generierten Spalte platziert wurden, und Sie wissen, dass die Werte mit dem generierten Ausdruck übereinstimmen, kann für die Tabelle der Status 'Überprüfung anstehend' inaktiviert werden, ohne die Werte zu prüfen oder zuzuordnen:

```
SET INTEGRITY FOR t1 GENERATED COLUMN IMMEDIATE UNCHECKED
```

Generierte Spalten können nur für Datentypen definiert werden, für die der Vergleichsoperator 'gleich' definiert ist. Die ausgeschlossenen Datentypen für die generierten Spalten sind die strukturierten Typen, LOBs, CLOBs, DBCLOBs, LONG VARCHAR, LONG VARGRAPHIC sowie benutzerdefinierte Typen, die unter Verwendung derselben ausgeschlossenen Datentypen definiert wurden.

Generierte Spalten können in Integritätsbedingungen, eindeutigen Indizes, referenziellen Integritätsbedingungen, Primärschlüsseln und globalen temporären Tabellen nicht verwendet werden. Eine mit LIKE erstellte Tabelle mit gespeicherten Sichten kann die Merkmale generierter Spalten nicht übernehmen. Generierte Spalten können ohne das Schlüsselwort DEFAULT nicht eingefügt oder aktualisiert werden. Beim Einfügen verhindert die Verwendung von DEFAULT, dass die Spalten in der Spaltenliste nummeriert werden müssen. Statt dessen können generierte Spalten in der Werteliste auf DEFAULT

gesetzt werden. Beim Aktualisieren ermöglicht DEFAULT die erneute Berechnung der generierten Spalten, die mit SET INTEGRITY ohne Prüfung in den Online-Modus versetzt wurden.

Die Verarbeitungsreihenfolge von Auslösern macht es erforderlich, dass Vorauslöser (BEFORE) in den Kopfdaten (vor der Aktualisierung) und im Hauptteil keine Verweise auf generierte Spalten enthalten. In der Verarbeitungsreihenfolge stehen generierte Spalten nach den Vorauslösern (BEFORE).

Das Dienstprogramm db2look kann die Prüfungen auf Integritätsbedingungen, die von einer generierten Spalte erzeugt werden, nicht erkennen.

Bei der Replikation darf die Zieltabelle in der Zuordnung keine generierten Spalten verwenden. Es gibt hierbei zwei Auswahlmöglichkeiten:

- Die Zieltabelle muss die generierte Spalte als normale Spalte definieren, d. h. nicht als generierte Spalte.
- Die Zieltabelle muss die generierte Spalte bei der Zuordnung übergehen.

Beim Arbeiten mit generierten Spalten gelten verschiedene Einschränkungen:

- Zwischen generierten Spalten dürfen keine Abhängigkeiten bestehen.
- Die Ausdrücke, die zum Erstellen der generierten Spalten verwendet werden, dürfen keine Unterabfragen enthalten. Dies schließt auch Ausdrücke ein, die READS SQL DATA-Funktionen unterstützen.
- Für generierte Spalten sind keine Prüfungen auf Integritätsbedingungen zulässig.
- Für generierte Spalten sind keine eindeutigen Indizes zulässig. Hierzu zählen auch eindeutige Integritätsbedingungen und Primärschlüssel.

Deklarieren einer flüchtigen Tabelle

Eine *flüchtige* Tabelle ist als Tabelle definiert, deren Inhalt zwischen leer und sehr umfangreich schwanken kann. Die Flüchtigkeit bzw. extreme Veränderbarkeit dieser Tabellenart macht die von RUNSTATS gesammelten Statistikdaten relativ unzuverlässig. Statistikdaten werden zu einem bestimmten Zeitpunkt gesammelt und spiegeln nur diesen Zeitpunkt wider. Das Generieren eines Zugriffsplans, der eine flüchtige Tabelle verwendet, kann zu einem fehlerhaften oder unzureichenden Zugriffsplan führen. Wenn zum Beispiel Statistikdaten gesammelt werden, während die flüchtige Tabelle leer ist, wählt das Optimierungsprogramm vorzugsweise die Tabellensuche als Zugriffsverfahren für die flüchtige Tabelle aus und nicht die Indexsuche.

Um dies zu verhindern, kann es sinnvoll sein, die Tabelle mit Hilfe der Anweisung ALTER TABLE als „flüchtig“ zu deklarieren. Wenn die Tabelle als „flüchtig“ deklariert ist, verwendet das Optimierungsprogramm vorzugsweise

die Indexsuche und nicht die Tabellensuche. Zugriffspläne, die als flüchtig deklarierte Tabellen verwenden, sind nicht von den vorhandenen Statistikdaten für diese Tabellen abhängig.

Gehen Sie wie folgt vor, um eine Tabelle mit der Steuerzentrale als 'flüchtig' zu deklarieren:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Tabelle, die modifiziert werden soll, und wählen Sie dann im Kontextmenü **Ändern** aus.
3. Wählen Sie auf der Seite **Tabelle** das Markierungsfeld **Kardinalität variiert erheblich während der Laufzeit** aus und klicken Sie dann auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Tabelle als „flüchtig“ zu deklarieren:

```
ALTER TABLE <tabellenname>  
    VOLATILE CARDINALITY
```

Ändern von Partitionierungsschlüsseln

Partitionierungsschlüssel können nur in Tabellen in Knotengruppen mit einer Einzelpartition geändert werden. Löschen Sie als Erstes den vorhandenen Partitionierungsschlüssel und erstellen Sie anschließend einen anderen.

Mit der folgenden SQL-Anweisung wird der Partitionierungsschlüssel MIX_INT aus der Tabelle MIXREC gelöscht:

```
ALTER TABLE MIXREC  
    DROP PARTITIONING KEY
```

Weitere Informationen zur Anweisung ALTER TABLE finden Sie im Handbuch *SQL Reference*.

Der Partitionierungsschlüssel einer Tabelle in einer Knotengruppe mit mehreren Partitionen kann nicht geändert werden. Wenn Sie versuchen, ihn zu löschen, wird ein Fehler zurückgegeben.

Führen Sie eine der folgenden Operationen aus, um den Partitionierungsschlüssel in Knotengruppen mit mehreren Partitionen zu ändern:

- Exportieren aller Daten in eine Knotengruppe mit einer Einzelpartition und anschließend wie bereits oben beschrieben
- Exportieren aller Daten, Löschen der Tabelle, Neuerstellen der Tabelle durch Neudefinieren des Partitionierungsschlüssels und anschließendes Importieren aller Daten.

Keine dieser Methoden ist für große Datenbanken praktikabel. Daher ist es wichtig, den geeigneten Partitionierungsschlüssel vor der Implementierung des Entwurfs großer Datenbanken zu definieren.

Ändern von Tabellenattributen

Es kann Gründe geben, aus denen Tabellenattribute wie die Option zur Datenerfassung (Data Capture), der Prozentsatz freien Speicherbereichs auf jeder Seite (PCTFREE), die Größe der Sperren (Lock Size) oder der Anfügemodus (Append) geändert werden sollten.

Die Größe des Speicherbereichs, der auf jeder Seite einer Tabelle freizulassen ist, wird durch den Wert für PCTFREE angegeben und spielt eine wichtige Rolle bei der effektiven Verwendung von Clustering-Indizes. Der Betrag, der anzugeben ist, hängt von der Art der vorhandenen Daten und von den zu erwartenden zukünftigen Daten ab. Der Wert für PCTFREE wird von den Dienstprogrammen LOAD und REORG beachtet, jedoch von INSERT-, UPDATE- und IMPORT-Aktivitäten ignoriert.

Durch Angeben eines größeren Werts für PCTFREE bleibt das Clustering länger erhalten. Allerdings wird auch mehr Plattenspeicherplatz benötigt.

Sie können die Größe (Granularität) von Sperren, die beim Zugriff auf die Tabelle verwendet wird, mit Hilfe des Parameters LOCKSIZE angeben. Standardmäßig werden bei der Erstellung der Tabelle Sperren auf Zeilenebene definiert. Durch die Verwendung von Sperren auf Tabellenebene kann die Leistung von Abfragen verbessert werden, da die Anzahl der Sperren, die aktiviert und freigegeben werden müssen, beschränkt wird.

Durch die Angabe von APPEND ON können Sie die allgemeine Leistung der Tabelle erhöhen. Diese Angabe ermöglicht schnellere INSERT-Aktivitäten, während der Aufwand für die Verwaltung von Informationen über den freien Speicherbereich vermieden wird.

Eine Tabelle mit einem Clustering-Index kann nicht zur Aktivierung des Anfügemodus (Append On) geändert werden. Entsprechend kann auch kein Clustering-Index für eine Tabelle mit aktiviertem Anfügemodus erstellt werden.

Ändern einer Identitätsspalte

Ändern Sie die Attribute einer vorhandenen Identitätsspalte mit der Anweisung ALTER TABLE. Weitere Informationen zu dieser Anweisung, einschließlich der Syntax, finden Sie im Handbuch *SQL Reference*.

Es gibt mehrere Möglichkeiten, eine Identitätsspalte so zu ändern, dass Sie einige Merkmale von Sequenzen aufweist.

Einige Aufgaben kommen nur bei der Anweisung ALTER TABLE und bei Identitätsspalten vor:

- Mit RESTART wird die Sequenz, die der Identitätsspalte zugeordnet ist, auf den implizit oder explizit beim ersten Erstellen der Spalte als Startwert zugeordneten Wert zurückgesetzt.
- Mit RESTART WITH <numerische Konstante> wird die Sequenz, die der Identitätsspalte zugeordnet ist, auf den exakten Wert der numerischen Konstante zurückgesetzt. Die numerische Konstante kann ein beliebiger positiver oder negativer Wert ohne Nullen rechts vom Dezimalzeichen sein, der der Identitätsspalte zugeordnet werden kann.

Ändern einer Sequenz

Ändern Sie die Attribute einer vorhandenen Sequenz mit der Anweisung ALTER SEQUENCE. Weitere Informationen zu dieser Anweisung, einschließlich der Syntax, finden Sie im Handbuch *SQL Reference*.

Die Attribute der Sequenz, die geändert werden können, sind:

- Ändern des Inkrements zwischen künftigen Werten
- Einrichten neuer Minimal- oder Maximalwerte
- Ändern der Anzahl zwischengespeicherter Sequenznummern
- Ändern der Anweisung, ob die Sequenz einen Zyklus ausführen soll oder nicht
- Ändern der Anweisung, ob die Sequenznummern in der Reihenfolge der Anweisung generiert werden müssen
- Neustarten der Sequenz

Es gibt zwei Aufgaben, die nicht Teil der Erstellung der Sequenz sind. Dabei handelt es sich um die folgenden Aufgaben:

- RESTART. Setzt die Sequenz auf den implizit oder explizit beim ersten Erstellen der Spalte als Startwert zugeordneten Wert zurück.
- RESTART WITH numerische Konstante. Setzt die Sequenz auf den exakten Wert der numerischen Konstante zurück. Die numerische Konstante kann ein beliebiger positiver oder negativer Wert ohne Nullen rechts vom Dezimalzeichen sein.

Nach dem Neustarten einer Sequenz oder dem Wechsel zu CYCLE können doppelte Sequenznummern auftreten. Von der Anweisung ALTER SEQUENCE sind nur zukünftige Sequenznummern betroffen.

Der Datentyp einer Sequenz kann nicht geändert werden. Sie müssen stattdessen die aktuelle Sequenz löschen und anschließend eine neue Sequenz unter Angabe des neuen Datentyps erstellen.

Alle zwischengespeicherten Sequenzwerte, die nicht von DB2 verwendet werden, gehen beim Ändern der Sequenz verloren.

Löschen einer Sequenz

Verwenden Sie die Anweisung `DROP`, um eine Sequenz zu löschen. Weitere Informationen zu dieser Anweisung, einschließlich der Syntax, finden Sie im Handbuch *SQL Reference*.

Eine bestimmte Sequenz kann wie folgt gelöscht werden:

```
DROP SEQUENCE sequenzname
```

Dabei ist `sequenzname` der Name der zu löschenden Sequenz, der den impliziten und expliziten Schemennamen umfasst, um eine vorhandene Sequenz genau zu kennzeichnen.

Vom System erstellte Sequenzen für Identitätsspalten können nicht mit der Anweisung `DROP SEQUENCE` gelöscht werden.

Sobald eine Sequenz gelöscht wurde, werden alle Zugriffsrechte für die Sequenz ebenfalls gelöscht.

Ändern von Übersichtstabellemerkmalen

Mit einigen Einschränkungen können Sie eine Übersichtstabelle in eine reguläre Tabelle bzw. eine reguläre Tabelle in eine Übersichtstabelle ändern. Bei anderen Tabellenarten als regulären und Übersichtstabellen ist eine Änderung der Tabellenart nicht möglich. Sie können z. B. keine replizierte Übersichtstabelle in eine reguläre Tabelle und umgekehrt ändern.

Nach dem Ändern einer regulären Tabelle in eine Übersichtstabelle wird die Tabelle in den Status 'Überprüfung anstehend' versetzt. Bei einer derartigen Änderung muss die Gesamtauswahl in der Definition der Übersichtstabelle mit der ursprünglichen Tabellendefinition übereinstimmen. Dies bedeutet Folgendes:

- Die Anzahl der Spalten muss gleich sein.
- Die Spaltennamen und -positionen müssen übereinstimmen.
- Die Datentypen müssen identisch sein.

Wenn die Übersichtstabelle für eine ursprüngliche Tabelle definiert ist, kann diese ursprüngliche Tabelle nicht in eine Übersichtstabelle geändert werden. Wenn für die ursprüngliche Tabelle Auslöser, Prüfungen auf Integritätsbedingungen, referenzielle Integritätsbedingungen oder definierte eindeutige Indizes definiert sind, kann diese nicht in eine Übersichtstabelle geändert werden. Wenn Sie die Tabellenmerkmale ändern, um eine Übersichtstabelle zu definieren, ist es nicht zulässig, die Tabelle innerhalb derselben Anweisung `ALTER TABLE` in irgendeiner Form zu ändern.

Beim Ändern einer regulären Tabelle in eine Übersichtstabelle kann die Gesamtauswahl der Übersichtstabelle nicht über Sichten, Aliasnamen oder andere Übersichtstabellen direkt oder indirekt auf die ursprüngliche Tabelle verweisen.

Geben Sie die folgenden Befehle ein, um eine Übersichtstabelle in eine reguläre Tabelle zu ändern:

```
ALTER TABLE sumtable  
SET SUMMARY AS DEFINITION ONLY
```

Geben Sie die folgenden Befehle ein, um eine reguläre Tabelle in eine Übersichtstabelle zu ändern:

```
ALTER TABLE regtable  
SET SUMMARY AS <gesamtauswahl>
```

Die Einschränkungen, die beim Ändern einer regulären Tabelle in eine Übersichtstabelle für die Gesamtauswahl gelten, stimmen größtenteils mit den Einschränkungen überein, die für das Erstellen einer Übersichtstabelle mit der Anweisung CREATE SUMMARY TABLE gelten.

Zusätzliche Informationen zur Anweisung CREATE SUMMARY TABLE finden Sie im Handbuch *SQL Reference*.

Aktualisieren der Daten in einer Übersichtstabelle

Sie können die Daten in einer oder mehreren Übersichtstabellen mit Hilfe der Anweisung REFRESH TABLE aktualisieren. Die Anweisung kann in ein Anwendungsprogramm eingebettet oder dynamisch abgesetzt werden. Zur Verwendung dieser Anweisung sind die Berechtigungen SYSADM oder DBADM bzw. das Zugriffsrecht CONTROL für die zu aktualisierende Tabelle erforderlich.

Das folgende Beispiel zeigt, wie Sie die Daten in einer Übersichtstabelle aktualisieren können:

```
REFRESH TABLE SUMTAB1
```

Weitere Informationen zur Anweisung REFRESH TABLE finden Sie im Handbuch *SQL Reference*.

Ändern eines benutzerdefinierten strukturierten Typs

Nach dem Erstellen eines strukturierten Typs stellen Sie eventuell fest, dass Sie diesem strukturierten Typ Attribute hinzufügen oder zugeordnete Attribute löschen müssen. Verwenden Sie dazu die Anweisung ALTER TYPE (Structured). Umfassende Informationen zu strukturierten Typen finden Sie im Handbuch *Application Development Guide*.

Löschen und Aktualisieren von Zeilen einer typisierten Tabelle

Zeilen können mit Hilfe gesuchter oder positionierter DELETE-Anweisungen aus typisierten Tabellen gelöscht werden. Zeilen können mit Hilfe gesuchter oder positionierter UPDATE-Anweisungen in typisierten Tabellen aktualisiert werden. Umfassende Informationen zu typisierten Tabellen finden Sie im Handbuch *Application Development Guide*.

Umbenennen einer vorhandenen Tabelle

Sie können eine vorhandene Tabelle mit einem neuen Namen innerhalb eines Schemas versehen und die Berechtigungen und Indizes, die für die ursprüngliche Tabelle erstellt wurden, beibehalten.

Die vorhandene, umzubenennende Tabelle kann ein Aliasnamen sein, der eine Tabelle angibt. Die vorhandene, umzubenennende Tabelle darf nicht den Namen einer Katalogtabelle, einer Übersichtstabelle, einer typisierten Tabelle haben oder ein Objekt außer Tabelle bzw. Aliasname sein.

Auf die vorhandene Tabelle darf nicht in einem der folgenden Objekte verwiesen werden:

- Sichten
- Auslöser
- Referenzielle Integritätsbedingungen
- Übersichtstabelle
- Bereich einer vorhandenen Referenzspalte

Außerdem darf es keine Prüfungen auf Integritätsbedingungen innerhalb der Tabelle und keine generierten Spalten außer der Identitätsspalte geben. Alle Pakete oder im Cache zwischengespeicherte dynamische SQL-Anweisungen, die von der ursprünglichen Tabelle abhängig sind, werden ungültig gemacht. Weiterhin werden vorhandene Aliasnamen, die auf die ursprüngliche Tabelle verweisen, nicht geändert.

Sie sollten mit Hilfe der entsprechenden Systemkatalogtabellen sicherstellen, dass die Tabelle, die Sie umbenennen, von keiner dieser Einschränkungen betroffen ist.

Pakete müssen erneut gebunden werden, wenn sie auf die Tabelle verweisen, die gerade umbenannt wurde. Pakete können unter folgenden Bedingungen implizit wieder gebunden werden:

- Eine andere Tabelle wird in den ursprünglichen Namen der Tabelle umbenannt.
- Ein Aliasname oder eine Sicht wird mit dem ursprünglichen Namen der Tabelle erstellt.

Eine dieser Optionen muss durchgeführt werden, bevor ein implizites oder explizites erneutes Binden versucht wird. Wenn keine der Optionen ausgeführt wird, schlägt das erneute Binden fehl.

Gehen Sie wie folgt vor, um mit der Steuerzentrale eine vorhandene Tabelle umzubenennen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Tabelle, die umbenannt werden soll, und wählen Sie dann im Kontextmenü **Umbenennen** aus.
3. Geben Sie den neuen Tabellennamen ein und klicken Sie dann auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine vorhandene Tabelle umzubenennen:

```
RENAME TABLE <schemenname>.<tabellenname> TO <neuer_name>
```

Mit der folgenden SQL-Anweisung wird die Tabelle EMPLOYEE innerhalb des Schemas COMPANY in EMPL umbenannt:

```
RENAME TABLE COMPANY.EMPLOYEE TO EMPL
```

Weitere Informationen zur Anweisung RENAME TABLE finden Sie im Handbuch *SQL Reference*.

Löschen einer Tabelle

Eine Tabelle kann mit der SQL-Anweisung DROP TABLE gelöscht werden.

Beim Löschen einer Tabelle wird die Zeile im Katalog SYSCAT.TABLES, die die Informationen über die Tabelle enthält, gelöscht, während alle anderen von der Tabelle abhängigen Objekte auf verschiedene Weise davon betroffen sind. Beispiel:

- Alle Spaltennamen werden gelöscht.
- Die für Spalten der Tabelle erstellten Indizes werden gelöscht.
- Alle Sichten, die auf der Tabelle basieren, werden als unbrauchbar markiert. (Weitere Informationen dazu finden Sie in „Wiederherstellen unbrauchbarer Sichten“ auf Seite 246.)
- Alle Zugriffsrechte auf die gelöschte Tabelle und abhängige Sichten werden implizit widerrufen.
- Alle referenziellen Integritätsbedingungen, in denen die Tabelle eine übergeordnete oder abhängige Tabelle ist, werden gelöscht.
- Alle Pakete und im Cache zwischengespeicherten dynamischen SQL-Anweisungen, die von der gelöschten Tabelle abhängig sind, werden als ungültig markiert und bleiben ungültig, bis die abhängigen Objekte neu

erstellt werden. Dazu gehören Pakete, die von einer zu löschenden übergeordneten Tabelle über der untergeordneten Tabelle in der Hierarchie abhängen. (Weitere Informationen dazu finden Sie in „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 253.)

- Referenzspalten, für die die gelöschte Tabelle als der Bereich der Referenz definiert ist, weisen keinen Bereich mehr auf.
- Die Definition eines Aliasnamens für die Tabelle wird nicht berührt, da ein Aliasname auch ohne Tabelle vorhanden sein kann.
- Alle von der gelöschten Tabelle abhängigen Auslöser werden als unbrauchbar markiert.
- Für alle Dateien, die über DATALINK-Spalten verbunden sind, wird die Verbindung aufgehoben. Die Operation zur Verbindungsaufhebung wird asynchron ausgeführt, was bedeutet, dass die Dateien eventuell nicht sofort für andere Operationen verfügbar sind.

Gehen Sie wie folgt vor, um mit der Steuerzentrale eine Tabelle zu löschen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Tabelle, die gelöscht werden soll, und wählen Sie dann im Kontextmenü **Löschen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Tabelle zu löschen:

```
DROP TABLE <tabellenname>
```

Mit der folgenden Anweisung wird die Tabelle DEPARTMENT gelöscht:

```
DROP TABLE DEPARTMENT
```

Eine einzelne Tabelle kann nicht gelöscht werden, wenn sie eine untergeordnete Tabelle hat. Alle Tabellen, die einer Tabellenhierarchie angehören, können jedoch mit einer einzigen Anweisung DROP TABLE HIERARCHY gelöscht werden, wie im folgenden Beispiel gezeigt:

```
DROP TABLE HIERARCHY person
```

In der Anweisung DROP TABLE HIERARCHY muss die Stammtabelle der zu löschenden Hierarchie angegeben werden.

Zwischen dem Löschen einer Tabellenhierarchie und dem Löschen einer bestimmten Tabelle bestehen folgende Unterschiede:

- DROP TABLE HIERARCHY aktiviert keine Löschauslöser wie sie von einzelnen Anweisungen DROP TABLE aktiviert würden. Beim Löschen einer

einzelnen Tabelle würden zum Beispiel Löschauslöser für die dazugehörigen übergeordneten Tabellen aktiviert.

- `DROP TABLE HIERARCHY` erstellt keine Protokolleinträge für die einzelnen Zeilen der gelöschten Tabellen. Vielmehr wird das Löschen der Hierarchie als ein einziges Ereignis protokolliert.

Weitere Informationen zur Anweisung `DROP` finden Sie im Handbuch *SQL Reference*.

Löschen einer benutzerdefinierten temporären Tabelle

Beim Löschen einer benutzerdefinierten temporären Tabelle müssen einige Aspekte berücksichtigt werden. Diese Tabellen werden mit der Anweisung `DECLARE GLOBAL TEMPORARY TABLE` erstellt.

Beim Löschen derartiger Tabellen muss der Tabellenname durch den Schemennamen `SESSION` qualifiziert werden und in einer Anwendung definiert sein, die zum Erstellen der Tabelle verwendet wurde.

Pakete dürfen nicht in einem Abhängigkeitsverhältnis zu dieser Tabellenart stehen und werden beim Löschen einer solchen Tabelle nicht ungültig.

Wird eine benutzerdefinierte temporäre Tabelle gelöscht und wurde die Erstellung dieser Tabelle vor der aktiven Arbeitseinheit bzw. dem aktuellen Sicherungspunkt durchgeführt, wird die Tabelle funktional gelöscht und die Anwendung kann nicht auf diese zugreifen. Allerdings bleibt für die Tabelle weiterhin Speicherplatz im zugehörigen Tabellenbereich reserviert. Dies verhindert, dass der temporäre Benutzertabellenbereich gelöscht wird, bevor die Arbeitseinheit festgeschrieben bzw. der Sicherungspunkt beendet ist.

Weitere Informationen zur Anweisung `DROP` finden Sie im Handbuch *SQL Reference*.

Löschen eines Auslösers

Ein Auslöserobjekt kann mit der Anweisung `DROP` gelöscht werden. Diese Prozedur hat jedoch zur Folge, dass abhängige Pakete als ungültig markiert werden, wie im Folgenden beschrieben:

- Wenn ein Auslöser zum Aktualisieren ohne explizite Spaltenliste gelöscht wird, werden Pakete mit einer Aktualisierung für die Zieltabelle ungültig gemacht.
- Wenn ein Auslöser zum Aktualisieren mit einer Spaltenliste gelöscht wird, werden Pakete mit einer Aktualisierung für die Zieltabelle nur dann ungültig gemacht, wenn das Paket auch eine Aktualisierung für mindestens eine Spalte in der Liste der Spaltennamen der Anweisung `CREATE TRIGGER` enthielt.

- Wenn ein Auslöser zum Einfügen gelöscht wird, werden Pakete, die eine Einfügung für die Zieltabelle enthalten, ungültig gemacht.
- Wenn ein Auslöser zum Löschen gelöscht wird, werden Pakete, die eine Löschung für die Zieltabelle enthalten, ungültig gemacht.

Ein Paket bleibt ungültig, bis das Anwendungsprogramm explizit gebunden bzw. neu gebunden wird oder das Anwendungsprogramm ausgeführt wird und der Datenbankmanager es automatisch erneut bindet.

Löschen einer benutzerdefinierten Funktion (UDF), Typenzuordnung oder Methode

Eine benutzerdefinierte Funktion (UDF), Funktionsschablone oder Funktionszuordnung kann mit der Anweisung DROP gelöscht werden.

Mit der Zuordnungsoption DISABLE können Sie eine Funktionszuordnung inaktivieren. Weitere Informationen hierzu finden Sie im Handbuch *SQL Reference*.

Eine benutzerdefinierte Funktion kann nicht gelöscht werden, wenn eine Sicht, ein Auslöser, eine Prüfung auf Integritätsbedingung in einer Tabelle oder eine andere benutzerdefinierte Funktion von ihr abhängig ist. Funktionen, die implizit durch die Anweisung CREATE DISTINCT TYPE generiert werden, können nicht gelöscht werden. Es ist nicht möglich, eine Funktion zu löschen, die sich im Schema SYSIBM oder im Schema SYSFUN befindet.

Andere Objekte können von einer Funktion oder Funktionsschablone abhängig sein. Alle solchen Abhängigkeiten (auch Funktionszuordnungen) müssen beseitigt werden, bevor eine Funktion gelöscht werden kann. Eine Ausnahme bilden nur Pakete, die als unbrauchbar markiert werden. Ein solches Paket wird nicht implizit erneut gebunden. Es muss entweder mit dem Befehl BIND bzw. REBIND erneut gebunden werden oder mit dem Befehl PREP vorbereitet werden. Weitere Informationen zu diesen Befehlen finden Sie im Handbuch *Command Reference*. Durch Löschen einer UDF werden alle Pakete oder im Cache zwischengespeicherte dynamische SQL-Anweisungen, die sie verwendet haben, ungültig gemacht.

Durch Löschen einer Funktionszuordnung wird ein Paket als ungültig markiert. Eine automatische erneute Bindeoperation wird ausgeführt, und das Optimierungsprogramm versucht, die lokale Funktion zu verwenden. Ist die lokale Funktion eine Schablone, so schlägt das implizite erneute Binden fehl.

(Weitere Informationen finden Sie in „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 253.)

Löschen eines benutzerdefinierten Datentyps (UDT) oder einer Typenzuordnung

Ein benutzerdefinierter Datentyp (UDT) oder eine Typenzuordnung kann mit der Anweisung DROP gelöscht werden. Ein UDT kann nicht gelöscht werden, wenn er wie folgt verwendet wird:

- In einer Spaltendefinition für eine vorhandene Tabelle oder Sicht (einzigartige Datentypen)
- Als Typ einer vorhandenen typisierten Tabelle oder typisierten Sicht (strukturierter Typ)
- Als übergeordneter Typ eines anderen strukturierten Typs

Eine Standardtypenzuordnung kann nicht gelöscht, sondern nur durch Erstellen einer anderen Typenzuordnung überschrieben werden.

Der Datenbankmanager versucht dann, alle Funktionen, die von diesem einzigartigen Typ abhängig sind, zu löschen. Wenn die benutzerdefinierten Funktionen nicht gelöscht werden können, kann auch der benutzerdefinierte Typ nicht gelöscht werden. Eine benutzerdefinierte Funktion kann nicht gelöscht werden, wenn eine Sicht, ein Auslöser, eine Prüfung auf Integritätsbedingung in einer Tabelle oder eine andere benutzerdefinierte Funktion von ihr abhängig ist. Durch Löschen eines benutzerdefinierten Datentyps werden alle Pakete oder im Cache zwischengespeicherte dynamische SQL-Anweisungen, die ihn verwendet haben, ungültig gemacht.

Wenn Sie eine Umsetzung für einen UDT erstellt haben und den UDT löschen wollen, sollten Sie erwägen (falls erforderlich), die Umsetzung zu löschen. Dies kann mit der Anweisung DROP TRANSFORM geschehen. Einzelheiten zu dieser Anweisung finden Sie im Handbuch *SQL Reference*. Beachten Sie, dass nur Umsetzungen, die von Ihnen oder von anderen Anwendungsentwicklern definiert wurden, gelöscht werden können. Integrierte Umsetzungen und die dazugehörigen Gruppenseiten können nicht gelöscht werden.

Weitere Informationen zu den benutzerdefinierten Datentypen (UDT) finden Sie in den Handbüchern *SQL Reference* und *Application Development Guide*.

Ändern oder Löschen einer Sicht

Die Anweisung ALTER VIEW ändert eine vorhandene Sicht durch Ändern einer Verweisartspalte zum Hinzufügen eines Bereichs. Für andere Änderungen an einer Sicht müssen Sie die Sicht löschen und erneut erstellen.

Beim Ändern der Sicht muss der Bereich einer vorhandenen Verweisartspalte (REF) hinzugefügt werden, für die noch kein Bereich definiert wurde. Zudem darf die Spalte nicht von einer übergeordneten Sicht übernommen werden.

Der Datentyp des Spaltennamens in der Anweisung ALTER VIEW muss REF sein (Typ des Namens der typisierten Tabelle oder des Namens der typisierten Sicht).

Andere Datenbankobjekte wie Tabellen und Indizes sind nicht betroffen, obwohl Pakete und im Cache zwischengespeicherte dynamische Anweisungen als ungültig markiert werden. Der Abschnitt „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 253 enthält weitere Informationen.

Zusätzliche Informationen zur Anweisung ALTER VIEW finden Sie im Handbuch *SQL Reference*.

Gehen Sie wie folgt vor, um mit der Steuerzentrale eine Sicht zu ändern:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Sichten** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Sicht, die modifiziert werden soll, und wählen Sie dann im Kontextmenü **Ändern** aus.
3. Geben Sie im Fenster **Sicht ändern** einen Kommentar ein oder modifizieren Sie einen vorhandenen Kommentar und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Sicht zu ändern:

```
ALTER VIEW <sichtname> ALTER <spaltenname>  
ADD SCOPE <typisierter Tabellen- oder Sichtname>
```

Gehen Sie wie folgt vor, um mit der Steuerzentrale eine Sicht zu löschen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Sichten** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Sicht, die gelöscht werden soll, und wählen Sie dann im Kontextmenü **Löschen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Sicht zu löschen:

```
DROP VIEW <sichtname>
```

Das folgende Beispiel zeigt, wie die Sicht EMP_VIEW gelöscht wird:

```
DROP VIEW EMP_VIEW
```

Alle Sichten, die von der gelöschten Sicht abhängig sind, werden als unbrauchbar markiert. (Weitere Informationen dazu finden Sie in „Wiederherstellen unbrauchbarer Sichten“ auf Seite 246.)

Wie von den Tabellenhierarchien her bekannt, kann eine ganze Sichtenhierarchie durch eine einzige Anweisung gelöscht werden, in der die Stammsicht der Hierarchie angegeben wird, wie im folgenden Beispiel gezeigt:

```
DROP VIEW HIERARCHY VPerson
```

Weitere Informationen zum Löschen und Erstellen von Sichten finden Sie im Handbuch *SQL Reference*.

Wiederherstellen unbrauchbarer Sichten

Sichten können *unbrauchbar* werden:

- Als Ergebnis eines widerrufenen Zugriffsrechts in einer zugrundeliegenden Tabelle
- Wenn eine Tabelle, ein Aliasname oder eine Funktion gelöscht wurde
- Wenn die übergeordnete Sicht unbrauchbar wurde
- Wenn die Sichten, von denen die Sicht abhängt, gelöscht wurden

Eine unbrauchbare Sicht kann auf folgende Weise wiederhergestellt werden:

1. Stellen Sie fest, mit welcher SQL-Anweisung die Sicht zu Anfang erstellt wurde. Diese Information können Sie der Spalte TEXT der Katalogsicht SYSCAT.VIEW entnehmen.
2. Erstellen Sie die Sicht erneut, indem Sie die Anweisung CREATE VIEW mit demselben Sichtnamen und derselben Definition verwenden.
3. Verwenden Sie die Anweisung GRANT, um alle Zugriffsrechte, die zuvor für die Sicht erteilt worden waren, erneut zu vergeben. (Beachten Sie, dass alle für eine unbrauchbar gewordene Sicht erteilten Zugriffsrechte widerrufen werden.)

Wenn Sie eine unbrauchbare Sicht nicht wiederherstellen möchten, können Sie sie explizit mit der Anweisung DROP VIEW löschen, oder Sie können eine neue Sicht mit demselben Namen, aber einer anderen Definition erstellen.

Eine unbrauchbare Sicht hat nur noch Einträge in den Katalogsichten SYSCAT.TABLES und SYSCAT.VIEWS. Alle Einträge in den Katalogsichten SYSCAT.VIEWDEP, SYSCAT.TABAUTH, SYSCAT.COLUMNS und SYSCAT.COLAUTH werden entfernt.

Löschen einer Übersichtstabelle

Sie können eine Übersichtstabelle nicht ändern, aber löschen.

Alle Indizes, Primärschlüssel, Fremdschlüssel und Prüfungen auf Integritätsbedingung, die auf die Tabelle verweisen, werden gelöscht. Alle Sichten und Auslöser, die auf die Tabelle verweisen, werden unbrauchbar gemacht. Alle Pakete, die von einem gelöschten oder als unbrauchbar markierten Objekt

abhängen, werden ungültig gemacht. Weitere Informationen zu Paketabhängigkeiten finden Sie in „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 253.

Gehen Sie wie folgt vor, um mit der Steuerzentrale eine Übersichtstabelle zu löschen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf der Übersichtstabelle, die gelöscht werden soll, und wählen Sie dann im Kontextmenü **Löschen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Übersichtstabelle zu löschen:

```
DROP TABLE <tabellenname>
```

Mit der folgenden SQL-Anweisung wird die Übersichtstabelle XT gelöscht:

```
DROP TABLE XT
```

Wiederherstellen unbrauchbarer Übersichtstabellen

Übersichtstabellen können als Ergebnis eines widerrufenen Zugriffsrechts SELECT in einer zugrundeliegenden Tabelle *unbrauchbar* werden.

Eine unbrauchbare Übersichtstabelle kann eventuell auf folgende Weise wiederhergestellt werden:

- Stellen Sie fest, mit welcher SQL-Anweisung die Übersichtstabelle zu Anfang erstellt wurde. Diese Information können Sie der Spalte TEXT der Katalogsicht SYSCAT.VIEW entnehmen.
- Erstellen Sie die Übersichtstabelle erneut, indem Sie die Anweisung CREATE SUMMARY TABLE mit demselben Übersichtstabellennamen und derselben Definition verwenden.
- Verwenden Sie die Anweisung GRANT, um alle Zugriffsrechte, die zuvor für die Übersichtstabelle erteilt worden waren, erneut zu erteilen. (Beachten Sie, dass alle für eine unbrauchbar gewordene Übersichtstabelle erteilten Zugriffsrechte widerrufen werden.)

Wenn Sie eine unbrauchbare Übersichtstabelle nicht wiederherstellen möchten, können Sie sie explizit mit der Anweisung DROP TABLE löschen, oder Sie können eine neue Übersichtstabelle mit demselben Namen, aber einer anderen Definition erstellen.

Eine unbrauchbare Übersichtstabelle hat nur noch Einträge in den Katalogsichten SYSCAT.TABLES und SYSCAT.VIEWS. Alle Einträge in den Katalogsichten SYSCAT.VIEWDEP, SYSCAT.TABAUTH, SYSCAT.COLUMNS und SYSCAT.COLAUTH werden entfernt.

Löschen einer Oberfläche

Mit der Anweisung DROP kann eine Oberfläche aus der Datenbank gelöscht werden. Das folgenden Beispiel zeigt, wie die Oberfläche DRDA gelöscht wird:

```
DROP WRAPPER DRDA
```

Alle Serverdefinitionen, benutzerdefinierten Funktionszuordnungen und benutzerdefinierten Datentypzuordnungen, die von der Oberfläche abhängen, werden gelöscht. Alle benutzerdefinierten Zuordnungen, Kurznamen, benutzerdefinierten Datentypzuordnungen und Benutzerzuordnungen, die von den gelöschten Serverdefinitionen abhängen, werden ebenfalls gelöscht. Alle Indexspezifikationen, die von den gelöschten Kurznamen abhängen, werden gelöscht. Alle Sichten, die von diesen Kurznamen abhängen, werden als unbrauchbar gekennzeichnet. Alle Pakete, die von den gelöschten Objekten und unbrauchbaren Sichten abhängen, werden ungültig gemacht.

Sie müssen über die Berechtigung SYSADM oder DBADM verfügen, um Oberflächen mit der Anweisung DROP löschen zu können.

Weitere Informationen zum Löschen von Oberflächen finden Sie im Handbuch *SQL Reference*.

Ändern oder Löschen eines Servers

Die Anweisung ALTER SERVER ändert eine vorhandene Server-Definition im Katalog der zusammengeschlossenen Datenbank. Mit dieser Anweisung können Sie Folgendes ausführen:

- Ändern der Definition einer bestimmten Datenquelle
- Ändern der Definition mehrerer Datenquellen eines bestimmten Typs bzw. einer bestimmten Version
- Ändern der Konfiguration einer bestimmten Datenquelle. Beispiel: Wenn das von einem bestimmten Server angegebene Datenbankverwaltungssystem auf eine neue Workstation mit schnellerem Prozessor umgestellt wird, sollten Sie die Serveroption `cpu_ratio` aktualisieren.

Diese Anweisung kann nicht zum Ändern der Serveroptionen `dbname` und `node` verwendet werden.

Das folgende Beispiel zeigt, wie der Server ORA1 geändert werden kann:

```
ALTER SERVER ORA1 OPTIONS (SET CPU_RATIO '5.0')
```

Server können aus der zusammengeschlossenen Datenbank gelöscht werden. Das folgende Beispiel zeigt, wie der Server ORALOC01 gelöscht wird:

```
DROP SERVER ORALOC01
```

Alle Kurznamen für Tabellen und Sichten, die sich auf der Datenquelle befinden, werden gelöscht. Alle Indexspezifikationen, die von diesen Kurznamen abhängen, werden gelöscht. Alle benutzerdefinierten Funktionszuordnungen, benutzerdefinierten Datentypzuordnungen und Benutzerzuordnungen, die von der gelöschten Serverdefinition abhängen, werden gelöscht. Alle Pakete, die von der gelöschten Serverdefinition, den Funktionszuordnungen, Kurznamen und Indexspezifikationen abhängen, werden ungültig gemacht.

Sie müssen über die Berechtigung SYSADM oder DBADM verfügen, um Server mit der Anweisung ALTER oder DROP ändern bzw. löschen zu können.

Weitere Informationen zum Löschen und Ändern von Servern finden Sie im Handbuch *SQL Reference*.

Ändern oder Löschen eines Kurznamens

Mit der Anweisung ALTER NICKNAME können lokal gespeicherte Informationen über eine Datenquellentabelle oder -sicht aktualisiert werden. Mit dieser Anweisung können Sie zum Beispiel den lokalen Namen einer Spalte ändern oder einen Spaltendatentyp einem anderen Datentyp zuordnen. Außerdem können Sie mit dieser Anweisung Spaltenoptionen hinzufügen. Weitere Informationen zur Syntax von ALTER NICKNAME finden Sie im Handbuch *SQL Reference*.

Beim Löschen eines Kurznamens werden für diesen Kurznamen erstellte Sichten als unbrauchbar markiert. Kurznamenspalten oder -datentypen können nicht geändert werden, wenn in einer Sicht auf den Kurznamen verwiesen wird.

Sie müssen über die Berechtigung SYSADM oder DBADM, die Datenbankberechtigung CONTROL bzw. ALL für den Kurznamen oder die Schemenberechtigung ALTERIN (für das aktuelle Schema) verfügen oder der definierende Benutzer des Kurznamens in der zusammengeschlossenen Datenbank sein, um diese Anweisung verwenden zu können.

Ändern einer Kurznamenspalte und Löschen eines Kurznamens

Das folgende Beispiel zeigt, wie der Kurzname TESTNN durch Ändern des lokalen Spaltennamens COL1 in NEWCOL geändert wird:

```
ALTER NICKNAME TESTNN ALTER COLUMN COL1 LOCAL NAME NEWCOL
```

Das folgende Beispiel zeigt, wie der Kurzname TESTNN gelöscht wird:

```
DROP NICKNAME TESTNN
```

Ändern von Kurznamenspaltenoptionen

Spalteninformationen werden in Form von Werten angegeben, die den als *Spaltenoptionen* bezeichneten Parametern zugeordnet werden. Diese Werte können in Groß- oder Kleinschreibung angegeben werden. Die nachfolgende Tabelle enthält Beschreibungen und zusätzliche Informationen zu diesen Werten.

Tabelle 3. Spaltenoptionen und zugehörige Einstellungen

Option	Gültige Einstellungen	Standard-einstellung
numeric_string	<p>'Y' Der Wert 'Y' (Yes) gibt an, dass diese Spalte nur Zeichenfolgen mit numerischen Daten enthält. WICHTIG: Wenn diese Spalte nur numerische Zeichenfolgen mit folgenden Leerzeichen enthält, ist die Angabe von 'Y' nicht zu empfehlen.</p> <p>'N' Der Wert 'N' (No) gibt an, dass diese Spalte nicht auf Zeichenfolgen mit numerischen Daten beschränkt ist.</p>	'N'

Durch Einstellen der Spaltenoption `numeric_string` auf den Wert 'Y' teilen Sie dem Optimierungsprogramm mit, dass diese Spalte keine Leerzeichen enthält, die einen störenden Einfluss auf das Sortieren der Daten dieser Spalte haben könnten. Diese Option ist in solchen Fällen nützlich, in denen die Sortierfolge einer Datenquelle von der Sortierfolge in DB2 abweicht. Mit dieser Option markierte Spalten werden nicht von der lokalen Auswertung (der Datenquelle) aufgrund unterschiedlicher Sortierfolgen ausgeschlossen.

Tabelle 3. Spaltenoptionen und zugehörige Einstellungen (Forts.)

Option	Gültige Einstellungen	Standard-einstellung
varchar_no_trailing_blanks	<p>Gibt an, ob folgende Leerzeichen in einer bestimmten Spalte des Typs VARCHAR nicht vorkommen:</p> <p>‘Y’ Der Wert ‘Y’ gibt an, dass folgende Leerzeichen in dieser VARCHAR-Spalte nicht vorkommen.</p> <p>‘N’ Der Wert ‘N’ gibt an, dass folgende Leerzeichen in dieser VARCHAR-Spalte vorkommen.</p> <p>Wenn VARCHAR-Spalten einer Datenquellen keine aufgefüllten Leerzeichen enthalten, ist die Strategie des Optimierungsprogramms für den Zugriff auf sie teilweise davon abhängig, ob die Spalten folgende Leerzeichen enthalten. Standardmäßig nimmt das Optimierungsprogramm an, dass sie tatsächlich folgende Leerzeichen enthalten. Von dieser Annahme ausgehend entwickelt es eine Zugriffsstrategie, die das Ändern von Abfragen vorsieht, so dass die aus diesen Spalten zurückgegebenen Werte auch den Erwartungen des Benutzers entsprechen. Falls eine VARCHAR-Spalte jedoch keine folgende Leerzeichen besitzt und Sie dies dem Optimierungsprogramm mitteilen, kann das Optimierungsprogramm eine effizientere Zugriffsstrategie entwickeln. Geben Sie eine Spalte in der Anweisung ALTER NICKNAME (zur Syntax siehe <i>SQL Reference</i>) an, um dem Optimierungsprogramm mitzuteilen, dass die Spalte keine folgenden Leerzeichen besitzt.</p>	‘N’

Löschen eines Index, einer Indexerweiterung oder einer Indexspezifikation

Es kann keine einzige Klausel einer Indexdefinition, Indexerweiterung oder Indexspezifikation geändert werden. Der Index bzw. die Indexerweiterung muss gelöscht und anschließend neu erstellt werden. (Das Löschen eines Index oder einer Indexspezifikation führt nicht dazu, dass andere Objekte gelöscht werden, jedoch können einige Pakete ungültig gemacht werden.)

Gehen Sie wie folgt vor, um mit der Steuerzentrale einen Index, eine Indexerweiterung oder eine Indexspezifikation zu löschen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Indizes** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf dem Index, der gelöscht werden soll, und wählen Sie dann im Kontextmenü **Löschen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie anschließend auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um einen Index, eine Indexerweiterung oder eine Indexspezifikation zu löschen:

```
DROP INDEX <indexname>
```

Mit der folgenden SQL-Anweisung wird der Index PH gelöscht:

```
DROP INDEX PH
```

Mit der folgenden SQL-Anweisung wird die Indexerweiterung IX_MAP gelöscht:

```
DROP INDEX EXTENSION ix_map RESTRICT
```

Der Name der Indexerweiterung muss sich auf eine Indexerweiterung beziehen, die im Katalog beschrieben wird. Die Klausel RESTRICT erzwingt die Anwendung der Regel, dass kein Index definiert werden darf, der in einem Abhängigkeitsverhältnis zur Indexerweiterungsdefinition steht. Wenn ein zugrundeliegender Index von dieser Indexerweiterung abhängig ist, schlägt die Löschoption fehl.

Ein Index für den Primärschlüssel oder eindeutigen Schlüssel kann nicht explizit gelöscht werden (außer wenn es sich um eine Indexspezifikation handelt). Zum Löschen eines solchen Index gibt es folgende Methoden:

- Wenn der Primärindex bzw. die eindeutige Integritätsbedingung automatisch für den Primärschlüssel oder eindeutigen Schlüssel erstellt wurde, wird der Index durch Löschen des Primärschlüssels oder eindeutigen Schlüssels gelöscht. Das Löschen des Schlüssels erfolgt mit Hilfe der Anweisung ALTER TABLE.
- Wenn der Primärindex oder die eindeutige Integritätsbedingung benutzerdefiniert ist, muss der Primärschlüssel oder eindeutige Schlüssel zuerst mit der Anweisung ALTER TABLE gelöscht werden. Nach dem Löschen des Primärschlüssels oder des eindeutigen Schlüssels wird der Index nicht länger als Primärindex oder eindeutiger Index betrachtet und kann daher explizit gelöscht werden.

Pakete und zwischengespeicherte dynamische SQL-Anweisungen, die von den gelöschten Indizes abhängen, werden als ungültig markiert. Der Abschnitt „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 253 enthält weitere Informationen. Das Anwendungsprogramm ist von Änderungen durch das Hinzufügen oder Löschen von Indizes nicht betroffen.

Anweisungsabhängigkeiten beim Ändern von Objekten

Anweisungsabhängigkeiten beziehen sich auf Pakete und im Cache zwischengespeicherte dynamische SQL-Anweisungen. Ein *Paket* (Package) ist ein Datenbankobjekt, das die Informationen enthält, die vom Datenbankmanager zum Zugriff auf Daten in der für ein bestimmtes Anwendungsprogramm effizientesten Weise benötigt werden. *Binden* (Binding) ist der Prozess, durch den das Paket erstellt wird, das der Datenbankmanager zum Zugriff auf Daten benötigt, wenn die Anwendung ausgeführt wird. Das Handbuch *Application Development Guide* enthält eine detaillierte Beschreibung dieser Paketerstellung.

Pakete und zwischengespeicherte dynamische SQL-Anweisungen können von vielen Arten von Objekten abhängig sein. Eine vollständige Liste dieser Objekte finden Sie im Handbuch *SQL Reference*.

Auf diese Objekte könnte explizit verwiesen werden, zum Beispiel, indem eine Tabelle oder eine benutzerdefinierte Funktion in einer SQL-Anweisung SELECT angegeben wird. Auf die Objekte könnte auch implizit verwiesen werden, zum Beispiel, wenn eine abhängige Tabelle überprüft werden muss, um sicherzustellen, dass keine referenziellen Integritätsbedingungen verletzt werden, wenn eine Zeile in einer übergeordneten Tabelle gelöscht wird. Pakete sind außerdem von den Zugriffsrechten abhängig, die dem Ersteller des Pakets erteilt wurden.

Wenn ein Paket oder eine im Cache zwischengespeicherte dynamische SQL-Anweisung von einem Objekt abhängig ist und dieses Objekt gelöscht wird, wird das Paket bzw. die dynamische SQL-Anweisung in den Status „ungültig“ versetzt. Wenn ein Paket von einer benutzerdefinierten Funktion abhängig ist und diese Funktion gelöscht wird, wird das Paket in den Status „unbrauchbar“ versetzt.

Eine im Cache zwischengespeicherte dynamische SQL-Anweisung, die sich im Status „ungültig“ befindet, wird bei ihrer nächsten Verwendung automatisch erneut optimiert. Wenn ein von der Anweisung benötigtes Objekt gelöscht wurde, schlägt die Ausführung der dynamischen SQL-Anweisung möglicherweise mit einer entsprechenden Fehlermeldung fehl.

Ein Paket, das sich im Status „ungültig“ befindet, wird bei der nächsten Verwendung implizit erneut gebunden. Ein solches Paket kann aber auch explizit erneut gebunden werden. Wurde ein Paket als ungültig markiert, weil ein Auslöser gelöscht wurde, kann das erneut gebundene Paket keinen Auslöser mehr aufrufen.

Ein Paket, das sich im Status „unbrauchbar“ befindet, muss explizit erneut gebunden werden, damit es verwendet werden kann. Im Handbuch *Application Development Guide* finden Sie weitere Informationen zum Binden und erneuten Binden von Paketen.

Für Objekte zusammengesetzter Datenbanken gelten ähnliche Abhängigkeiten. Durch das Löschen eines Servers werden z. B. alle diesem Server zugeordneten Pakete oder im Cache zwischengespeicherte, dynamische SQL-Anweisungen mit Verweisen auf Kurznamen ungültig gemacht.

In einigen Fällen ist es nicht möglich, das Paket erneut zu binden. Zum Beispiel, wenn eine Tabelle gelöscht und nicht wieder erstellt wurde, kann das Paket nicht erneut gebunden werden. In diesem Fall muss entweder das Objekt neu erstellt oder die Anwendung so geändert werden, dass sie nicht mehr auf das gelöschte Objekt zugreift.

In vielen anderen Fällen, zum Beispiel, wenn eine Integritätsbedingung gelöscht wurde, kann das Paket erneut gebunden werden.

Mit Hilfe der folgenden Systemkatalogsichten können Sie den Status eines Pakets und die Abhängigkeiten des Pakets feststellen:

- SYSCAT.PACKAGEAUTH
- SYSCAT.PACKAGEDEP
- SYSCAT.PACKAGES

Weitere Informationen zu Objektabhängigkeiten finden Sie in der Beschreibung der Anweisung DROP im Handbuch *SQL Reference*.

Teil 3. Datenbanksicherheit

Kapitel 5. Steuern des Datenbankzugriffs

Eine der wichtigsten Zuständigkeiten des Datenbankadministrators und des Systemadministrators ist die Datenbanksicherheit. Die Sicherheit der Datenbank umfasst folgende Bereiche:

- Verhindern unbeabsichtigter Verluste von Daten oder Datenintegrität durch technische Fehler und Systemausfälle.
- Verhindern des unbefugten Zugriffs auf wertvolle Daten. Es muss sichergestellt werden, dass nicht durch Unbefugte auf vertrauliche Daten zugegriffen wird.
- Verhindern der absichtlichen Löschung oder Manipulation von Daten durch unbefugte Personen.
- Überwachen des Zugriffs auf Daten durch Benutzer. Dies wird in „Kapitel 6. Protokollieren von DB2-Aktivitäten“ auf Seite 323 behandelt.

Die folgenden Themen werden behandelt:

- „Auswählen von Benutzer-IDs und Gruppen für Ihre Installation“ auf Seite 258
- „Auswählen einer Authentifizierungsart für den Server“ auf Seite 261
- „Überlegungen zur Authentifizierung für ferne Clients“ auf Seite 270
- „Überlegungen zu partitionierten Datenbanken“ auf Seite 270
- „Verwenden von DCE-Sicherheitservices zur Überprüfung von Benutzern“ auf Seite 270
- „Authentifizierungsverarbeitung in zusammengeschlossenen Datenbanken“ auf Seite 278
- „Zugriffsrechte und Berechtigungen“ auf Seite 286
- „Steuern des Zugriffs auf Datenbankobjekte“ auf Seite 304
- „Aufgaben und erforderliche Berechtigungen“ auf Seite 316
- „Verwenden des Systemkatalogs“ auf Seite 317.

Planung für Sicherheit: Beginnen Sie, indem Sie Ihre Objekte für einen Steuerungsplan für den Datenbankzugriff definieren und angeben, wem unter welchen Bedingungen auf welche Daten der Zugriff gewährt werden soll. Ihr Plan sollte auch beschreiben, wie diese Ziele durch Verwendung von Datenbankfunktionen, Funktionen anderer Programme und Verwaltungsprozeduren erreicht werden sollen.

Auswählen von Benutzer-IDs und Gruppen für Ihre Installation

Sicherheitsaspekte spielen für den DB2-Administrator von dem Moment an eine wichtige Rolle, wo das Produkt installiert ist. Das jeweils plattform-spezifische Handbuch *Einstieg* bietet alle Informationen, die zum Planen, Installieren und Konfigurieren von DB2 erforderlich sind.

Für die Schritte zur Ausführung der Installation von DB2 ist ein Benutzername, ein Gruppenname und ein Kennwort erforderlich. Während der Installation hat der Administrator Standardwerte für diese Anforderungen. Nachdem die Standardwerte während der Installation von DB2 verwendet wurden, sollte der Administrator vor der Erstellung von Exemplaren, in denen die Datenbanken angelegt werden, unbedingt neue Benutzernamen, Gruppennamen und Kennwörter erstellen. Die Verwendung neuer Benutzernamen, Gruppennamen und Kennwörter minimiert das Risiko, dass ein anderer Benutzer außer dem Administrator von den Standardwerten erfährt und diese in unzuträglicher Weise in Exemplaren und Datenbanken verwendet.

Kennwörter sind bei der Authentifizierung von Benutzern ausgesprochen wichtig. Wenn auf der Ebene des Betriebssystems keine Anforderungen an die Authentifizierung festgelegt wurden und die Datenbank das Betriebssystem zur Authentifizierung von Benutzern verwendet, wird den Benutzern die Erlaubnis zur Verbindung erteilt. Bei einem UNIX-Betriebssystem werden nicht definierte Kennwörter zum Beispiel wie NULL behandelt. Jeder Benutzer ohne definiertes Kennwort wird als ein Benutzer mit einem NULL-Kennwort behandelt. Aus der Perspektive des Betriebssystems ist dies eine Übereinstimmung, und der Benutzer ist somit erfolgreich überprüft worden und darf eine Verbindung zur Datenbank herstellen. Es sollte daher auf der Ebene des Betriebssystems die Verwendung von Kennwörtern erforderlich sein, wenn Sie möchten, dass das Betriebssystem die Authentifizierung von Benutzern der Datenbank übernimmt.

Eine weitere Empfehlung zur Sicherheit nach der Installation von DB2 ist die Änderung der Standardzugriffsrechte, die Benutzern erteilt sind. Während des Installationsprozesses werden SYSADM-Zugriffsrechte (System Administration) standardmäßig den folgenden Benutzern auf jedem Betriebssystem erteilt:

OS/2

Eine gültige DB2-Benutzer-ID, die zur UPM-Administratorgruppe (UPM = User Profile Management) oder lokalen Administratorgruppe gehört.

Windows 95 oder Windows 98

Alle Windows 95- oder Windows 98-Benutzer.

Windows NT oder Windows 2000

Ein gültiger DB2-Benutzername, der zur Administratorgruppe gehört

UNIX

Ein gültiger DB2-Benutzername, der zur Primärgruppe der Benutzer-ID des Exemplareigners gehört

SYSADM-Zugriffsrechte bilden die umfassendste Gruppe von Zugriffsrechten, die innerhalb von DB2 zur Verfügung steht. (Zugriffsrechte werden später in diesem Kapitel behandelt.) Es ist daher vielleicht nicht sinnvoll, dass alle diese Benutzer über SYSADM-Zugriffsrechte verfügen. DB2 stattet den Administrator mit der Möglichkeit aus, Gruppen und einzelnen Benutzer-IDs Zugriffsrechte zu erteilen und zu entziehen.

Die plattformspezifischen Informationen zur Erstellung und Zuordnung von Gruppen und Benutzer-IDs finden Sie in den verschiedenen Handbüchern *Einstieg*. Durch Aktualisieren des Parameters SYSADM_GROUP in der Datenbankmanagerkonfiguration kann der Administrator steuern, welche Gruppe als Systemadministratorgruppe mit Systemadministratorberechtigungen definiert ist. Sie müssen die im Folgenden dargestellten Richtlinien befolgen, um die Sicherheitsanforderungen für die DB2-Installation und die nachfolgende Erstellung von Exemplaren und Datenbanken zu erfüllen.

Jede Gruppe, die (durch Aktualisieren von SYSADM_GROUP) als Systemadministratorgruppe definiert ist, muss vorhanden sein. Der Name dieser Gruppe sollte eine leichte Erkennung als die Gruppe ermöglichen, die für Exemplareigner erstellt wurde. Die Benutzer-IDs und Gruppen, die dieser Gruppe angehören, verfügen über die Systemadministratorberechtigung für ihre jeweiligen Exemplare.

Es wird empfohlen, eine Benutzer-ID für den Exemplareigner zu erstellen, die als zu einem bestimmten Exemplar gehörig erkannt werden kann. Diese Benutzer-ID sollte als eine ihrer Gruppen den Namen der oben erstellten SYSADM-Gruppe enthalten. Eine weitere Empfehlung ist, dass diese Benutzer-ID des Exemplareigners nur als Mitglied der Exemplareignergruppe und in keiner anderen Gruppe verwendet wird. Dadurch wird die Zunahme der Benutzer-IDs und Gruppen gesteuert, die die Exemplarumgebung ändern können.

Der erstellten Benutzer-ID sollte immer ein Kennwort zugewiesen werden, um eine Authentifizierung vor dem Zugriff auf die Daten und Datenbanken des Exemplars zu ermöglichen. Es wird empfohlen, bei der Erstellung des Kennworts die Kennwortrichtlinien des Unternehmens einzuhalten.

Überlegungen zur Windows NT-Plattform

Beim Arbeiten unter Enterprise – Extended Edition für Windows NT kann die Systemadministratorberechtigung (SYSADM) jedem gültigen DB2-Benutzereintrag zugeordnet werden, der zur lokalen Administratorgruppe auf der Maschine gehört, auf der der Benutzereintrag definiert wurde.

Wenn ein Benutzer sich z. B. bei einem Domänenbenutzereintrag anmeldet und versucht, auf eine DB2-Datenbank zuzugreifen, greift DB2 auf den Domänen-Controller zu, um die verfügbaren Gruppen (einschließlich der Administratorgruppe) zu spezifizieren. Sie können diese Funktionsweise auf die beiden folgenden Arten ändern:

1. Definieren des Wertes 'local' für die Variable DB2_GRP_LOOKUP der Profilregistrierdatenbank und Hinzufügen der Domänenbenutzereinträge (oder globalen Gruppen) zur lokalen Administratorgruppe.
2. Aktualisieren der Konfigurationsdatei des Datenbankmanagers, so dass diese die neue Gruppe angibt. Wenn diese Gruppe auf der lokalen Maschine spezifiziert werden soll, muss auch die Variable DB2_GRP_LOOKUP der Profilregistrierdatenbank definiert werden.

Standardmäßig verfügen in einer Domänenumgebung unter Windows NT nur die Domänenbenutzer, die zur Administratorgruppe des PDC (Primary Domain Controller) gehören, über die Berechtigung SYSADM für ein Exemplar. Da DB2 die Vergabe von Berechtigungen immer auf der Maschine ausführt, auf der ein Benutzereintrag definiert wurde, wird einem Domänenbenutzer durch das Hinzufügen zur lokalen Administratorgruppe auf dem Server nicht die Berechtigung SYSADM für die Gruppe erteilt.

Um das Hinzufügen eines Domänenbenutzers zur Administratorgruppe auf dem PDC zu vermeiden, sollten Sie eine globale Gruppe erstellen und sowohl die Domänen- als auch die lokalen Benutzer hinzufügen, denen die Berechtigung SYSADM erteilt werden soll. Geben Sie dazu folgende Befehle ein:

```
DB2STOP
DB2 UPDATE DBM CFG USING SYSADM_GROUP globale_gruppe
DB2START
```

Überlegungen zur UNIX-Plattform

Auf UNIX-gestützten Plattformen muss eine Gruppe für abgeschirmte benutzerdefinierte Funktionen (Fenced UDFs) und für gespeicherte Prozeduren erstellt werden, und alle Benutzer-IDs, die abgeschirmte benutzerdefinierte Funktionen und gespeicherte Prozeduren verwenden, müssen Mitglieder dieser Gruppe sein. Wie auch bei der SYSADM-Gruppe sollte der Name der Gruppe für abgeschirmte benutzerdefinierte Funktionen und gespeicherte Prozeduren eine leichte Identifizierung ermöglichen. Die Benutzer-IDs, die zur Gruppe der abgeschirmten benutzerdefinierten Funktionen oder zur Gruppe gespeicherter Prozeduren gehören, haben standardmäßig alle Berechtigungen und Zugriffsrechte, die dieser Gruppe zugeordnet werden.

Aus Sicherheitsgründen wird empfohlen, den Exemplarnamen nicht als Namen der Gruppe für abgeschirmte Funktionen (Fenced ID) zu verwenden. Wenn Sie jedoch nicht beabsichtigen, abgeschirmte UDFs oder gespeicherte Prozeduren zu verwenden, können Sie die Fenced ID auf den Exemplarnamen setzen, anstatt eine weitere Benutzer-ID zu erstellen.

Es wird empfohlen, eine Benutzer-ID zu erstellen, die als dieser Gruppe zugeordnet erkannt werden kann. Der Benutzer für abgeschirmte UDFs und gespeicherte Prozeduren wird als Parameter der Prozedur zur Exemplarerstellung angegeben (db2icrt ... -u <AbgeschirmtID>). Dies ist nicht erforderlich, wenn Sie DB2-Clients Application Enabler oder DB2 Software Developer's Kit installieren.

Allgemeine Regeln

Im Folgenden werden die Regeln für die Benennung aller Objekte und Benutzer aufgelistet. Einige dieser Regeln gelten nur für die Plattform, auf der Sie arbeiten. Es gibt zum Beispiel eine Regel für die Verwendung von Groß- und Kleinbuchstaben in einem Namen.

- Auf UNIX-Plattformen müssen Namen in Kleinbuchstaben angegeben werden.
- Unter OS/2 müssen Namen in Großbuchstaben angegeben werden.
- Auf Windows-Plattformen sind Groß-, Kleinschreibung und gemischte Groß- und Kleinschreibung zulässig.

Im Abschnitt „Anhang A. Namenskonventionen“ auf Seite 369 finden Sie die DB2-Namenskonventionen.

Der Befehl db2icrt erstellt das Verzeichnis für die Haupt-SQL-Bibliothek (sql-lib) unter dem Benutzerverzeichnis des Exempleigners.

Auswählen einer Authentifizierungsart für den Server

Der Zugriff auf ein Exemplar oder eine Datenbank erfordert zunächst, dass der Benutzer *auf seine Identität überprüft* wird. Die *Authentifizierungsart* für das jeweilige Exemplar bestimmt, wie und wo der Benutzer überprüft wird. Die Authentifizierungsart wird in der Konfigurationsdatei des Datenbankmanagers auf dem Server gespeichert. Sie wird erstmalig bei der Erstellung des Exemplars definiert. Weitere Informationen zum Datenbankmanager-Konfigurationsparameter für die *Authentifizierung* finden Sie in „Konfigurieren von DB2“ im Handbuch *Systemverwaltung: Optimierung*. Es gibt eine Authentifizierungsart pro Exemplar, die für den Zugriff auf den zugehörigen Datenbankserver und alle Datenbanken unter seiner Steuerung gilt.

Wenn Sie auf Datenquellen von einer zusammengeschlossenen Datenbank aus zugreifen wollen, müssen Sie die Authentifizierungsverarbeitung der Datenquellen und die Definitionen für die Authentifizierungsarten zusammenge-

geschlossener Datenbanken beachten. Weitere Informationen finden Sie in „Authentifizierungsverarbeitung in zusammengeschlossenen Datenbanken“ auf Seite 278.

Die folgenden Authentifizierungsarten stehen zur Verfügung:

SERVER

Gibt an, dass die Authentifizierung auf dem Server mit Hilfe der lokalen Sicherheitseinrichtungen des Betriebssystems stattfindet. Wenn eine Benutzer-ID und ein Kennwort bei einem Versuch, die Verbindung zu einer Datenbank oder einem Exemplar herzustellen, angegeben werden, werden sie mit den gültigen Kombinationen aus Benutzer-ID und Kennwort auf dem Server verglichen, um zu bestimmen, ob dem Benutzer der Zugriff auf das Exemplar gewährt wird. Dies ist der Standardsicherheitsmechanismus.

Anmerkung: Der Server-Code erkennt, ob eine Verbindung lokal oder fern ist. Für lokale Verbindungen sind bei der Authentifizierungsart SERVER keine Benutzer-ID und kein Kennwort zur erfolgreichen Authentifizierung erforderlich.

Wenn das ferne Exemplar über die Authentifizierungsart SERVER verfügt, gibt es zwei Möglichkeiten für die Durchführung der Authentifizierung:

- Die Benutzer-ID und das Kennwort werden vom Benutzer zur Verfügung gestellt.
- Die Benutzer-ID und das Kennwort werden von DB2 abgerufen und anschließend zur Gültigkeitsprüfung an den Server übermittelt. (Der Benutzer ist bereits bei einer lokalen Maschine oder der Domäne angemeldet.)

SERVER_ENCRYPT

Gibt an, dass der Server verschlüsselte SERVER-Authentifizierungschemata annimmt. Wenn die Client-Authentifizierung nicht angegeben ist, wird der Client mit der Methode überprüft, die auf dem Server ausgewählt ist.

Wenn als Client-Authentifizierung DCS oder SERVER angegeben ist, wird der Client überprüft, indem die Benutzer-ID und das Kennwort an den Server übergeben werden. Wenn die Client-Authentifizierung DCS_ENCRYPT oder SERVER_ENCRYPT lautet, wird der Client überprüft, indem eine Benutzer-ID und ein verschlüsseltes Kennwort übergeben werden.

Wenn auf dem Client SERVER_ENCRYPT und auf dem Server SERVER angegeben ist, wird ein Fehler zurückgegeben, da die Authentifizierungsebenen voneinander abweichen.

CLIENT

Gibt an, dass die Authentifizierung mit Hilfe der Sicherheitseinrichtungen des Betriebssystems in der Datenbankpartition erfolgt, in der die Anwendung aufgerufen wird. Wenn eine Benutzer-ID und ein Kennwort bei einem Versuch, die Verbindung zu einer Datenbank oder einem Exemplar herzustellen, angegeben werden, werden sie mit den gültigen Kombinationen aus Benutzer-ID und Kennwort auf dem Client-Knoten verglichen, um zu bestimmen, ob dem Benutzer der Zugriff auf das Exemplar gewährt wird. Es findet keine weitere Authentifizierung auf dem Datenbankserver statt.

Wenn der Benutzer eine lokale Anmeldung bzw. Anmeldung auf dem Client vornimmt, ist der Benutzer nur dieser lokalen Client-Workstation bekannt.

Wenn für das ferne Exemplar die Authentifizierungsart CLIENT definiert ist, bestimmen zwei weitere Parameter die endgültige Authentifizierungsart: *trust_allclnts* und *trust_clntauth*.

Sicherheit auf CLIENT-Ebene nur für gesicherte Clients (TRUSTED):

Gesicherte Clients sind Clients, die über ein zuverlässiges lokales Sicherheitssystem verfügen. Insbesondere sind alle Clients mit Ausnahme der Windows 95- und Windows 98-Clients gesicherte Clients.

Wenn die Authentifizierungsart CLIENT ausgewählt wurde, kann eine zusätzliche Option zum Schutz vor Clients ausgewählt werden, deren Betriebsumgebung keine eigenen Sicherheitssysteme hat.

Zum Schutz gegen ungesicherte Clients kann der Administrator die Authentifizierung für gesicherte Clients auswählen, indem er den Parameter *trust_allclnts* auf den Wert NO setzt. Dies impliziert, dass alle gesicherten Plattformen den Benutzer anstelle des Servers überprüfen können. Nichtgesicherte Clients werden auf dem Server überprüft und müssen daher eine Benutzer-ID und ein Kennwort bereitstellen. Der Konfigurationsparameter *trust_allclnts* wird verwendet, um anzugeben, ob Sie den Clients „vertrauen“ (Trust). Der Standardwert für diesen Parameter ist YES.

Anmerkung: Es ist möglich, allen Clients zu „vertrauen“ (in dem Sie *trust_allclnts* auf YES setzen), auch wenn Sie einige solcher Clients haben, die über kein eigenes Sicherheitssystem zur Authentifizierung verfügen.

Vielleicht erscheint es Ihnen auch wünschenswert, die Authentifizierung auch für gesicherte Clients auf dem Server durchzuführen. Zur Festlegung, wo gesicherte Clients überprüft werden sollen, verwenden Sie den Konfigurationsparameter *trust_clntauth*. Der Standardwert für diesen Parameter ist CLIENT. Weitere Informationen zu diesem Parameter des Datenbankmanagers finden Sie in „Konfigurieren von DB2“ im Handbuch *Systemverwaltung: Optimierung*.

Anmerkung: Wenn bei gesicherten Clients beim Versuch mit CONNECT oder ATTACH eine Verbindung herzustellen, keine Benutzer-ID oder Kennwort angegeben werden, findet die Gültigkeitsprüfung auf dem Client statt. Der Parameter *trust_clntauth* wird nur dazu verwendet, festzulegen, wo die Informationen, die in den Klauseln USER/USING angegeben werden, zu überprüfen sind.

Wenn Sie den Parameter *trust_allclnts* auf DRDAONLY setzen, definieren Sie einen Schutz gegen alle Clients mit Ausnahme von DRDA-Clients von DB2 für MVS und OS/390, DB2 für VM und VSE und DB2 für OS/400. Nur diese Clients dürfen die Authentifizierung auf der Client-Seite ausführen. Alle anderen Clients müssen eine Benutzer-ID und ein Kennwort bereitstellen, deren Gültigkeit vom Server überprüft wird.

Der Parameter *trust_clntauth* wird verwendet, um zu bestimmen, wo die oben genannten Clients authentifiziert werden: Wenn *trust_clntauth* auf "client" gesetzt ist, findet die Authentifizierung auf dem Client statt. Wenn *trust_clntauth* auf "server" gesetzt ist, wird die Authentifizierung auf dem Client ausgeführt, sofern kein Kennwort angegeben ist, und auf dem Server, sofern ein Kennwort angegeben ist.

Tabelle 4. Authentifizierungsmodi mit einer Kombination der Parameter TRUST_ALLCLNTS und TRUST_CLNTAUTH

TRUST_ALLCLNTS	TRUST_CLNTAUTH	Ungesicherte Nicht-DRDA-Client-Authentifizierung ohne Kennwort	Ungesicherte Nicht-DRDA-Client-Authentifizierung mit Kennwort	Gesicherte Nicht-DRDA-Client-Authentifizierung ohne Kennwort	Gesicherte Nicht-DRDA-Client-Authentifizierung mit Kennwort	DRDA-Client-Authentifizierung ohne Kennwort	DRDA-Client-Authentifizierung mit Kennwort
YES	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT
YES	SERVER	CLIENT	SERVER	CLIENT	SERVER	CLIENT	SERVER
NO	CLIENT	SERVER	SERVER	CLIENT	CLIENT	CLIENT	CLIENT
NO	SERVER	SERVER	SERVER	CLIENT	SERVER	CLIENT	SERVER
DRDAONLY	CLIENT	SERVER	SERVER	SERVER	SERVER	CLIENT	CLIENT
DRDAONLY	SERVER	SERVER	SERVER	SERVER	SERVER	CLIENT	SERVER

DCS Dient im Wesentlichen dazu, eine Datenbank zu katalogisieren, auf die über DB2 Connect zugegriffen wird. (Der Abschnitt zur Sicherheit im *DB2 Connect Benutzerhandbuch* enthält weitere Einzelangaben zu diesem Thema.) Wenn **DCS** zur Angabe der Authentifizierungsart für ein Exemplar in der Konfigurationsdatei des Datenbankmanagers dient, bedeutet dies dasselbe wie die Authentifizierungsart **SERVER**, sofern nicht auf den Server über die DRDA-Anwendungs-Server-Architektur (DRDA - Distributed Relational Database Architecture) mit Hilfe des APPC-Protokolls (APPC - Advanced Program-To-Program Communications) zugegriffen wird. In diesem Fall gibt **DCS** an, dass die Authentifizierung auf dem Server, aber nur auf der APPC-Ebene stattfindet. Eine weitere Authentifizierung im DB2-Code wird nicht durchgeführt. Dieser Wert wird nur dann unterstützt, wenn der APPC-Parameter SECURITY für die Verbindung den Wert SAME oder PROGRAM hat.

DCS_ENCRYPT

Gibt an, dass DB2 Connect verschlüsselte SERVER-Authentifizierungsschemata annimmt. Wenn die Client-Authentifizierung nicht angegeben ist, wird der Client mit der Methode überprüft, die auf dem Server ausgewählt ist.

Wenn als Client-Authentifizierung DCS oder SERVER angegeben ist, wird der Client überprüft, indem die Benutzer-ID und das Kennwort an DB2 Connect übergeben werden. Wenn die Client-Authentifizierung DCS_ENCRYPT oder SERVER_ENCRYPT lautet, wird der Client überprüft, indem eine Benutzer-ID und ein verschlüsseltes Kennwort übergeben werden.

Wenn auf dem Client DCS_ENCRYPT und auf dem Server DCS angegeben ist, wird ein Fehler zurückgegeben, da die Authentifizierungsebenen voneinander abweichen.

DCE Gibt an, dass der Benutzer mit Hilfe der DCE-Sicherheitsservices (DCE Security Services) überprüft wird. Weitere Informationen zur DCE-Sicherheit finden Sie in „Verwenden von DCE-Sicherheitsservices zur Überprüfung von Benutzern“ auf Seite 270.

DCE_SERVER_ENCRYPT

Gibt an, dass der Server DCE-Authentifizierungen oder verschlüsselte SERVER-Authentifizierungsschemata annimmt. Wenn die Client-Authentifizierung nicht angegeben ist, wird der Client mit Hilfe der DCE-Sicherheitsservices (DCE Security Services) überprüft. Weitere Informationen zur DCE-Sicherheit finden Sie in „Verwenden von DCE-Sicherheitsservices zur Überprüfung von Benutzern“ auf Seite 270.

Wenn als Client-Authentifizierung SERVER oder DCS angegeben ist, wird der Client überprüft, indem die Benutzer-ID und das Kennwort an den Server übergeben werden. Wenn die Client-Authentifizierung SERVER_ENCRYPT oder DCS_ENCRYPT lautet, wird der Client überprüft, indem eine Benutzer-ID und ein verschlüsseltes Kennwort übergeben werden. Als Authentifizierungsart des Clients kann nicht DCE_SERVER_ENCRYPT angegeben werden. Wenn als Authentifizierungsart eines Exemplars DCE_SERVER_ENCRYPT angegeben wird, verwenden alle lokalen Anwendungen DCE als Authentifizierungsschema. Dies gilt auch für Dienstprogrammbeefehle, die keine Datenbankverbindung oder Exemplarverbindung erfordern.

Mit der Authentifizierungsart `DCE_SERVER_ENCRYPT` können nicht nur die Authentifizierungsarten `DCE` und `SERVER_ENCRYPT` kombiniert werden, sondern es werden damit auch die Einschränkungen für die Verwendung von Gruppen innerhalb von `DCE` verringert. Wenn die Authentifizierungsart auf `DCE_SERVER_ENCRYPT` gesetzt ist, wird angenommen, dass die Gruppenliste, die außerhalb der Authentifizierungszeit angefordert wird, vom Basisbetriebssystem und nicht von `DCE` gesendet wird. Sie können jedoch als Administrator auf dem Server einen Benutzer definieren, der dem kurzen `DCE`-Namen entspricht, damit die Gruppenliste außerhalb der Authentifizierungszeit unterstützt wird.

KERBEROS

Wird verwendet, wenn sowohl der `DB2-Client` als auch der Server unter Betriebssystemen installiert sind, die das Kerberos-Sicherheitsprotokoll unterstützen. Das Kerberos-Sicherheitsprotokoll führt die Authentifizierung mit Hilfe von Services anderer Hersteller aus und verwendet ein konventionelles Verschlüsselungsverfahren, um einen gemeinsam benutzten, geheimen Schlüssel zu erstellen. Dieser Schlüssel wird als Identitätsnachweis für den Benutzer definiert und zur Prüfung der Identität von Benutzern während aller Anforderungen für lokale oder Netzwerkservices verwendet. Der Schlüssel eliminiert die Notwendigkeit, den Benutzernamen sowie das zugehörige Kennwort unverschlüsselt über das Netz zu übertragen. Mit dem Kerberos-Sicherheitsprotokoll kann eine Einzelanmeldung an einem fernen `DB2-Server` durchgeführt werden.

KRB_SERVER_ENCRYPT

Gibt an, dass der Server `KERBEROS`-Authentifizierungen oder verschlüsselte `SERVER`-Authentifizierungsschemata annimmt. Wenn als Client-Authentifizierung `KERBEROS` angegeben wurde, wird der Client mit Hilfe des Kerberos-Sicherheitssystems authentifiziert. Wurde als Client-Authentifizierung nicht `KERBEROS` angegeben, entspricht die Authentifizierungsart des Systems der Einstellung `SERVER_ENCRYPT`.

Anmerkung: Die Kerberos-Authentifizierungsarten werden nur auf Clients und Servern unterstützt, die unter Windows 2000 ausgeführt werden.

Anmerkungen:

1. Die von Ihnen ausgewählte Authentifizierungsart ist nur wichtig, wenn ferne Datenbank-Clients auf die Datenbank zugreifen oder wenn Sie Funktionen für zusammengeschlossene Datenbanken verwenden. Für Benutzer, die über lokale Clients auf die Datenbank zugreifen, wird die Authentifizierung immer auf dem System durchgeführt, auf dem sich die Datenbank befindet. Eine Ausnahme kann vorkommen, wenn DCE-Sicherheitsservices verwendet werden. Weitere Informationen zur Unterstützung und Verwendung ferner Clients finden Sie im Handbuch *Einstieg*.
2. Sperren Sie sich nicht versehentlich von Ihrem Exemplar aus, wenn Sie Berechtigungsinformationen ändern, da der Zugriff auf die Konfigurationsdatei selbst durch Informationen in der Konfigurationsdatei geschützt ist. Mit den folgenden Parametern in der Konfigurationsdatei des Datenbankmanagers wird der Zugriff auf das Exemplar gesteuert:
 - AUTHENTICATION *
 - SYSADM_GROUP *
 - TRUST_ALLCLNTS
 - TRUST_CLNTAUTH
 - SYSCTRL_GROUP
 - SYSMANT_GROUP

* kennzeichnet die beiden wichtigsten Parameter, die am ehesten Probleme verursachen könnten.

Es gibt einige Maßnahmen, um dies zu verhindern: Wenn Sie sich versehentlich aus dem DB2-System aussperren, gibt es auf allen Plattformen eine Sicherheitsoption, die Ihnen ermöglicht, die normalen DB2-Sicherheitsprüfungen außer Kraft zu setzen, um die Konfigurationsdatei des Datenbankmanagers mit Hilfe eines Sicherheitsbenutzers des lokalen Betriebssystems mit hohen Zugriffsrechten zu aktualisieren. Dieser Benutzer hat **immer** das Zugriffsrecht zur Aktualisierung der Konfigurationsdatei des Datenbankmanagers und kann daher das Problem beheben. Diese Umgehung der Sicherheit ist jedoch auf eine lokale Aktualisierung der Konfigurationsdatei des Datenbankmanagers beschränkt. Sie können einen Sicherheitsbenutzer nicht fern oder für irgendeinen anderen DB2-Befehl verwenden.

Dieser Benutzer mit Sonderberechtigung wird folgendermaßen identifiziert:

- UNIX-Plattformen: der Exemplareigner
 - NT-Plattform: eine zur lokalen Gruppe der Administratoren gehörende Person
 - OS/2-Plattform: ein UPM-Administrator
 - Andere Plattformen: Da es auf anderen Plattformen keine lokale Sicherheit gibt, bestehen alle Benutzer die lokale Sicherheitsprüfung ohnehin.
3. Weitere Informationen zu Windows NT Security finden Sie im „Anhang E. Arbeitsweise von DB2 für Windows NT mit der Windows NT-Sicherheit“ auf Seite 429.

Überlegungen zur Authentifizierung für ferne Clients

Bei der Katalogisierung einer Datenbank für den fernen Zugriff, kann die Authentifizierungsart im Datenbankverzeichniseintrag angegeben werden.

Für Datenbanken, auf die über DB2 Connect zugegriffen wird, gilt: Wenn kein Wert angegeben ist, wird die Authentifizierungsart SERVER angenommen.

Für Datenbanken, auf die fern zugegriffen wird, aber nicht mit Hilfe von DB2 Connect, ist die Authentifizierungsart nicht erforderlich. Wenn sie aber nicht angegeben ist, muss der Client zuerst mit dem Server in Verbindung treten, um den Wert abzurufen, bevor der Prozess der Authentifizierung beginnen kann. Wenn sie angegeben ist, kann die Authentifizierung unverzüglich beginnen, vorausgesetzt, der angegebene Wert stimmt mit dem Wert auf dem Server überein. Wenn keine Übereinstimmung festgestellt wird, versucht DB2 eine Behebung, wodurch mehr Übertragungen zum Ausgleichen des Unterschieds entstehen bzw. ein Fehler, wenn DB2 die fehlende Übereinstimmung nicht beheben kann. Im Fall einer fehlenden Übereinstimmung wird angenommen, dass der Wert auf dem Server korrekt ist.

Überlegungen zu partitionierten Datenbanken

In einer partitionierten Datenbank muss jede Partition der Datenbank über dieselbe Menge definierter Benutzer und Gruppen verfügen. Wenn die Definitionen nicht übereinstimmen, kann der Benutzer in verschiedenen Partitionen zu verschiedenen Operationen berechtigt sein. Konsistenz über alle Partitionen hinweg ist zu empfehlen.

Verwenden von DCE-Sicherheitsservices zur Überprüfung von Benutzern

Im Hinblick auf die Sicherheit für Ihre verteilte Datenbankumgebung sind die DCE-Sicherheitsservices (DCE - Distributed Computing Environment Security Services) eine gute Wahl, da DCE folgende Funktionen bereitstellt:

- Zentralisierte Verwaltung von Benutzern und Kennwörtern
- Keine Übertragung von Kennwörtern und Benutzer-IDs im Klartext
- Eine einzige Anmeldung für Benutzer

DB2 unterstützt DCE-Standardanmeldekontexte (Default Login Contexts), Verbindungsanmeldekontexte (Connection Login Contexts) und delegierte Kontexte (Delegated Contexts). Ein *Standardanmeldekontext* wird eingerichtet, wenn ein Benutzer den Befehl `dce_login` auf einem Client ausführt. Nachfolgende DB2-Befehle haben Zugriff auf diesen Kontext und können Benutzerauthentifizierungen ohne weitere Beteiligung des Benutzers (d. h., ohne dass der Benutzer eine Benutzer-ID oder ein Kennwort angeben muss) durchführen. Ein *Verbindungsanmeldekontext* wird für eine DB2-Sitzung mit Hilfe der Benutzer-ID und dem Kennwort eingerichtet, die im Befehl `CONNECT` oder

ATTACH mit Klausel USER/USING angegeben wurden. Ein *delegierter Anmelderkontext* kommt zustande, wenn ein DB2-Client als Teil einer DCE-Server-Anwendung verwendet wird. Die DCE-Server-Anwendung (die auch ein DB2-Client ist) empfängt Anforderungen von einer DCE-Client-Anwendung, von der die ursprüngliche Identität des Benutzers definiert wird. Vorausgesetzt, der DCE-Client und der DCE-Server sind korrekt konfiguriert, um dem DCE-Server die Stellvertreterfunktion für den DCE-Client zu ermöglichen, empfängt DB2 das Stellvertreter-Token und leitet es an den DB2-Server weiter. Dadurch kann der DB2-Server zur Verarbeitung der Anforderungen die ursprüngliche Identität des DCE-Client verwenden und muss nicht die Identität des DCE-Servers verwenden. Informationen, wie ein delegierter Anmelderkontext eingerichtet werden kann, können der DCE-Dokumentation für Ihre Plattform entnommen werden.

Anmerkung: Es gibt mehrere Lieferantenprodukte, die DCE unterstützen. Für einen reibungslosen Betrieb von DB2 UDB für Windows NT mit dem DCE-Produkt von IBM im Bereich der Sicherheitsservices wurden zwei neue DLL-Dateien zur Verfügung gestellt: `db2dces.ibm` und `db2dcec.ibm`. (Diese DLL-Dateien sind nur für Windows NT gültig.) Wenn Sie das DCE-Produkt von IBM für Sicherheitsservices erwerben und verwenden, müssen diese beiden Dateien entsprechend nach `db2dces.dll` bzw. `db2dcec.dll` kopiert werden. Wenn Sie das DCE-Produkt eines anderen Lieferanten erwägen, wenden Sie sich an die Serviceorganisation des Lieferanten und an die Serviceorganisation von DB2 UDB, um zu ermitteln, ob die DCE-Implementierung des Lieferanten für alle Sicherheitsservices mit DB2 UDB funktioniert.

Einrichten eines DB2-Benutzers für DCE

Benutzer müssen in der DCE-Registrierdatenbank (Distributed Computing Environment Registry) registriert werden und über korrekte Attribute verfügen, bevor sie mit DB2 verwendet werden können. In der entsprechenden plattformspezifischen DCE-Dokumentation finden Sie Informationen zur Erstellung eines DCE-Principal.

Jeder DB2-Benutzer, der einen Server mit DCE-Authentifizierung verwenden will, muss einen DCE-Principal und einen Benutzereintrag in der DCE-Registrierdatenbank mit der Markierung CLIENT definiert haben. Dieser Principal muss außerdem einen Eintrag in seinem Abschnitt für die erweiterten Registrierungsattribute (ERA - Extended Registry Attributes) haben, der angibt, welcher Berechtigungsname für diesen Principal verwendet wird, wenn er die Verbindung zu einem bestimmten Server mit DCE-Authentifizierung herstellt.

Es kann auch wünschenswert sein, Benutzer-Principals als Mitglieder von Gruppen zu definieren, um in der Datenbank Gruppenzugriffsrechte zu verwenden. Ähnliche Informationen im Gruppen-ERA ordnen dem Gruppennamen einen DB2-Berechtigungsnamen zu. Der Berechtigungsname ist ein sekundärer Berechtigungsname, jedoch gelten dieselben Einschränkungen. Bitte entnehmen Sie Ihrer DCE-Dokumentation die weiteren Informationen zur Erstellung von Gruppen und zum Hinzufügen von Mitgliedern.

Die Informationen im ERA ordnen einen DCE-Principal eines Benutzers oder einen Gruppennamen einem DB2-Berechtigungsnamen für einen bestimmten DCE-Principal-Namen eines Servers zu. Zur Verwendung eines ERA muss ein ERA-Schema definiert werden, das das Format dieses Attributs angibt. Dies muss einmal für jede DCE-Zelle ausgeführt werden und kann durch die Ausführung der folgenden Schritte erfolgen:

1. Melden Sie sich bei DCE als gültiger DCE-Administrator an.
2. Rufen Sie `dcecp` auf, und geben Sie Folgendes ein:

```
> xattrschema create ./:/sec/xattrschema/db2map \  
> -aclmgr {{principal r m r m } {group r m r m }} \  
> -annotation {Schema entry for DB2 database access} \  
> -encoding stringarray \  
> -multivalued no \  
> -uuid 1cbe84ca-9df3-11cf-84cd-02608c2cd17b
```

Dadurch wird das ERA (Extended Registry Attribute) `db2map` erstellt.

Zum Anzeigen dieser Zuordnung geben Sie den folgenden Befehl in die `dcecp`-Eingabeaufforderung ein:

```
> xattrschema show ./:/sec/xattrschema/db2map
```

Es wird Folgendes angezeigt:

```
{axlmgr  
  {{principal {{query r} {update m} {test r} {delete m}}}  
   {group      {{query r} {update m} {test r} {delete m}}}}}  
{annotation {Schema entry for DB2 database access}}  
{applydefs no}  
{intercell rejects}  
{multivalued no}  
{reserved no}  
{scope {}}  
{trigbind {}}  
{trigtype none}  
{unique no}  
{uuid 1cbe84ca-9df3-11cf-84cd-02608c2cd17b}
```

Anmerkung: Einschränkungen für den Inhalt des Berechtigungsnamens im ERA werden von DCE nicht umgesetzt. Wenn einem DCE-Principal oder einer DCE-Gruppe ein ungültiger Berechtigungsname gegeben wird, entsteht ein Fehler, wenn von DB2 der Versuch unternommen wird, diesen Benutzer zu überprüfen. (Beachten Sie, dass die Authentifizierung bei den Befehlen CONNECT, ATTACH, DB2START bzw. bei jeder anderen Operation stattfinden kann, bei der eine Authentifizierung erforderlich ist.) Es ist außerdem sehr zu empfehlen, bei der Zuweisung von Berechtigungsnamen für DCE-Prinzipals sicherzustellen, dass die Namen eins-zu-eins und eindeutig zugewiesen werden. DCE überprüft diese Bedingungen nicht.

Wenn ein DB2-Client auf einen DB2 Universal Database-Server zugreifen soll, müssen nach Ihrer Registrierung als DCE-Principals die ERA-Informationen hinzugefügt werden, um die Zuordnung zwischen Principal-Name und Berechtigungsname herzustellen. Dies muss für jeden Benutzer bzw. jede Gruppe einmal ausgeführt werden und kann durch Ausführen der folgenden Schritte erfolgen:

- Melden Sie sich bei DCE als gültiger DCE-Administrator an.
- Rufen Sie `dcecp` auf, und geben Sie in der Eingabeaufforderung Folgendes ein:

```
> principal modify principal_name \  
> -add {db2map map_1 map_2...map_n}
```

Dabei verwendet `map_n` folgendes Format:

```
DCE_server_principal,DB2_authid
```

Hierbei ist `DCE_server_principal` ein gültiger DCE-Principal-Name für einen DB2 UDB-Server (oder das Platzhalterzeichen `*`, mit dem angezeigt wird, dass diese Zuordnung für einen beliebigen DB2-Server gültig ist, der nicht bereits in einem anderen Eintrag `map_n` angegeben ist) und `DB2_authid` ein gültiger DB2-Berechtigungsname.

Wenn eine DCE-Gruppe für ein DCE-Principal verwendet wird, muss sie auch einer DB2-Berechtigungs-ID (`authid`) zugeordnet sein, die über die korrekte Berechtigung verfügt, wie `SYSADM` oder `SYSCTRL`.

Beachten Sie, dass die Berechtigungskennung (`authid`) die im DCE-Schema angegeben wird, das für die Zuordnung eines DCE-Principal-Namens zu einer DB2-Berechtigungs-ID (`authid`) verwendet wird, in Großbuchstaben angegeben werden *muss*. Die Verwendung von Kleinbuchstaben oder gemischter Groß-/Kleinschreibung der Berechtigungs-ID (`authid`) führt zu einem Fehler.

Einrichten eines DB2-Servers zur Verwendung von DCE

Server müssen in der DCE-Registrierdatenbank (Distributed Computing Environment Registry) als Principals registriert werden und über korrekte Attribute verfügen, bevor sie mit DB2 verwendet werden können. In der entsprechenden plattformspezifischen DCE-Dokumentation finden Sie Informationen zur Erstellung eines DCE-Server-Principal.

Der DCE-Sicherheits-Client-Laufzeitcode muss installiert sein und für das Serverexemplar zugänglich sein.

Jeder DB2-Server, der DCE zur Authentifizierung verwenden will, muss sich bei DCE zum Zeitpunkt der Ausführung von DB2START registrieren. Damit dies nicht manuell geschehen muss, stellt DCE eine Methode zur Verfügung, mit der ein Server die Informationen zur eigenen Benutzer-ID und zum eigenen Kennwort (Schlüssel) in einer speziellen Datei verwaltet, die als *Chiffrierschlüsseldatei* (Keytab File) bezeichnet wird. Bei Ausführung von DB2START liest DB2 die Konfigurationsdatei des Datenbankmanagers und ruft die Authentifizierungsart für das Exemplar ab. Wenn DB2 feststellt, dass die Authentifizierungsart DCE ist, werden vom DB2-Server DCE-Aufrufe abgesetzt, um die Informationen aus der Chiffrierschlüsseldatei abzurufen. Mit diesen abgerufenen Informationen wird der Server bei DCE registriert. Diese Registrierung ermöglicht es dem Server, DCE-Token von DCE-Clients zu empfangen und sie zur Überprüfung dieser Benutzer zu verwenden.

Der Exemplaradministrator muss die Chiffrierschlüsseldatei (Keytab File) für das Exemplar mit Hilfe von DCE-Befehlen erstellen. Detaillierte Informationen zur Erstellung einer Chiffrierschlüsseldatei finden Sie in der DCE-Dokumentation für Ihre Plattform. Lesen Sie in diesem Dokument die Einzelangaben zur Chiffrierschlüsseldatei (Keytab File) und zu den Befehlen *dcecp keytab* und *rgy_edit*. Die DB2-Chiffrierschlüsseldatei muss den Namen *keytab.db2* erhalten und sich im Unterverzeichnis *security* des Verzeichnisses *sql1ib* für das Exemplar befinden. (Auf Intel-Betriebssystemen muss sich die Datei im Unterverzeichnis *security* des durch *INSTANCENAME* definierten Unterverzeichnisses im Verzeichnis *sql1ib* befinden. *INSTANCENAME* ist der Exemplarname des Exemplars, mit dem Sie arbeiten.) Sie sollte nur einen Eintrag für den Server-Principal für das angegebene Exemplar enthalten. Alle weiteren Einträge führen bei der Ausführung von DB2START zu einem Fehler. Auf UNIX-Plattformen muss diese Datei mit Dateizugriffsrechten geschützt werden, so dass nur der Exemplareigner Schreib-/Lesezugriffsrechte besitzt.

Es folgt ein Beispiel für die Erstellung der Chiffrierschlüsseldatei:

- Melden Sie sich bei DCE als gültiger DCE-Benutzer an.
- Rufen Sie `rgy_edit` auf, und geben Sie Folgendes in die Eingabeaufforderung ein:

```
> ktadd -p principal_name -pw principal_password \  
> -f keytab.db2
```

Damit DB2 nach erfolgreicher DCE-Konfiguration mit der DCE-Authentifizierung gestartet wird, muss DB2 angewiesen werden, die DCE-Authentifizierung zu verwenden, indem die Konfigurationsdatei des Datenbankmanagers durch Angeben der Authentifizierungsart „DCE“ aktualisiert wird. Dies geschieht durch Eingabe des folgenden CLP-Befehls:

```
db2 update database manager configuration using authentication DCE  
sysadm_group DCE_group_name
```

Führen Sie anschließend den Befehl `dce_login` für einen gültigen DB2-DCE-Benutzer aus, der über die Berechtigung `SYSADM` verfügt, und setzen Sie `DB2START` ab.

Anmerkung: Bevor Sie DB2 mit DCE-Authentifizierung starten, stellen Sie sicher, dass Sie einen DCE-Benutzer-Principal für die Verwendung als Ihr `SYSADM` für das Exemplar definiert haben, so dass Sie eine gültige DCE-Benutzer-ID haben, von der aus Sie das Exemplar starten, stoppen und verwalten können. Anweisungen hierzu finden Sie in „Einrichten eines DB2-Benutzers für DCE“ auf Seite 271.

Stellen Sie über diese Anweisungen hinaus sicher, dass der erstellte Principal ein Mitglied der Gruppe `SYSADM_GROUP` für dieses Exemplar ist. Standardmäßig lautet dieser Gruppenname `DB2ADMIN` für die DCE-Authentifizierung, wenn keine Gruppe explizit angegeben ist (d. h., wenn `SYSADM_GROUP` gleich null ist), jedoch kann er aktualisiert werden, bevor die Authentifizierungsart für das Exemplar in einen Gruppennamen (Berechtigungsname) Ihrer Wahl geändert wird. Die von Ihnen ausgewählte DCE-Gruppe muss ein definiertes ERA (Extended Registry Attribute) haben, das ihr den angegebenen Berechtigungsname für die Gruppe `SYSADM_GROUP` zuordnet.

Eine der Funktionen des DB2-Verwaltungsservers ist das Starten von DB2-Exemplaren. Bei AUTHENTICATION = DCE muss der DCE-Principal, der in der DB2-Chiffrierschlüsseldatei für das Exemplar verwendet wird, eine gültige Zuordnung von DCE-Principal zur DB2-Berechtigungs-ID (authid) haben. Diese Zuordnung ist für den DB2-Verwaltungsserver erforderlich, damit er das DB2-Exemplar starten kann. Die gültige Zuordnung ermöglicht dieser ID, sowohl als Client als auch als Server zu fungieren.

Einrichten eines DB2-Client-Exemplars zur Verwendung von DCE

Ein Nur-Client-Exemplar kann zur Verwendung der DCE-Authentifizierung für lokale Operationen eingerichtet werden, indem die Konfigurationsdatei des Datenbankmanagers aktualisiert und die Authentifizierungsart auf den Wert DCE gesetzt wird. Eine Chiffrierschlüsseldatei ist für ein Nur-Client-Exemplar nicht erforderlich, da es keinen Server gibt, der sich bei DCE registrieren muss. Im allgemeinen ist es nicht empfehlenswert (oder erforderlich), dass ein Nur-Client-Exemplar von DB2 die DCE-Authentifizierung verwendet. Sie wird jedoch unterstützt.

Ein Client, der auf eine ferne Datenbank mit DCE-Sicherheit zugreifen will, muss Zugriff auf das entsprechende DCE-Sicherheitsprodukt haben. Der Client kann wahlfrei die Authentifizierungsart für die Zieldatenbank im Datenbankverzeichnis katalogisieren. Wenn der Client die DCE-Authentifizierung angibt, muss auch der vollständig qualifizierte DCE-Server-Principal-Name angegeben werden. Wenn DCE-Authentifizierung im Verzeichnis nicht definiert wird, werden die Informationen zur Authentifizierung und zum Principal vom Server bei Ausführung von CONNECT abgerufen.

DB2-Einschränkungen bei der Verwendung der DCE-Sicherheit

Die Verwendung der DCE-Authentifizierung führt zu Einschränkungen bei bestimmten SQL-Funktionen, die von DB2 zur Verfügung gestellt werden und die Unterstützung von Gruppen betreffen. Folgende Einschränkungen sind bei Verwendung der DCE-Authentifizierung zu beachten:

- Bei der Verwendung der Anweisungen GRANT und REVOKE **müssen** die Schlüsselwörter USER und GROUP zur Qualifikation des angegebenen Berechtigungsnamens angegeben werden. Ansonsten wird ein Fehler gemeldet.
- Bei der Verwendung der Klausel AUTHORIZATION der Anweisung CREATE SCHEMA wird die Gruppenzugehörigkeit des angegebenen Berechtigungsnamens bei der Auswertung der Berechtigungen zur Ausführung der Anweisungen, die dieser Klausel folgen, **nicht** berücksichtigt. Dies kann zu einem Berechtigungsfehler während der Ausführung der Anweisung CREATE SCHEMA führen.
- Wenn ein Paket von einem anderen Benutzer als dem ursprünglichen Binder des Pakets erneut gebunden wird, werden die Zugriffsrechte des

ursprünglichen Binders erneut ausgewertet. In diesem Fall wird eine Gruppenzugehörigkeit des ursprünglichen Binders bei der Neuauswertung der Zugriffsrechte nicht berücksichtigt. Dies kann zu einem Berechtigungsfehler beim erneuten Binden führen.

Die DCE-Authentifizierung, wie sie von DB2 durchgeführt wird, arbeitet mit dem Fluss von DCE-Tickets, die mit Hilfe der GSSAPI-Schnittstelle (OSF DCE Generic Security Services Application Programming Interface) abgerufen werden. Sämtliche Authentifizierung für DCE-Sicherheit findet auf der Datenbankprotokollebene statt. Bestimmte Kommunikationseinrichtungen bieten eventuell eine zusätzliche Sicherheit für die Übertragungsebene, die nicht unbedingt in DCE integriert ist. In den Fällen, in denen die Authentifizierung auf Übertragungsebene vollständig unabhängig von der Authentifizierung auf der Datenbankprotokollebene erfolgen kann, gibt es keine Einschränkungen. Jedoch müssen die Kriterien sowohl der Authentifizierung auf Datenbankprotokollebene als auch der auf Übertragungsebene erfüllt werden, bevor eine Verbindung erfolgreich hergestellt werden kann. In Fällen, in denen die Funktionen zur Authentifizierung auf Datenbankprotokollebene und auf Übertragungsprotokollebene interagieren, kann ihre Verwendung eingeschränkt sein, wenn einige Kombinationen zu einer Sicherheitsbeeinträchtigung führen.

Die DCE-Authentifizierung kann in Verbindung mit der TCP/IP-SOCKS-Unterstützung verwendet werden. Die beiden Sicherheitseinrichtungen arbeiten jedoch unabhängig voneinander. Dies kann bedeuten, dass der Benutzer nicht nur einen gültigen DCE-Anmeldekontext angeben muss, sondern auch mit einer Benutzer-ID des lokalen Betriebssystems angemeldet sein muss, die die Kriterien des SOCKS-Servers erfüllt.

Die DCE-Authentifizierung kann in Verbindung mit benannten NT-Pipes (NT Named Pipes) verwendet werden. Die beiden Sicherheitseinrichtungen arbeiten jedoch unabhängig voneinander. Der Benutzer muss nicht nur einen gültigen DCE-Anmeldekontext angeben, sondern auch mit einer Benutzer-ID an der NT-Domäne angemeldet sein, die die Kriterien der Unterstützung der benannten NT-Pipes erfüllt.

Zur Vermeidung einer möglichen Verwirrung, wenn sowohl DCE-Principals als auch Benutzer-IDs des lokalen Betriebssystems zur Authentifizierung herangezogen werden, wie in den beiden obigen Beispielen, kann eine integrierte DCE-Anmeldung verwendet werden. In diesem Fall wird der Benutzer bei der Anmeldung an einem System außerdem automatisch in den entsprechenden DCE-Principal angemeldet. In der DCE-Dokumentation für Ihre Plattform finden Sie Informationen zur Verwendung dieser Einrichtung, wenn sie unterstützt wird. Beachten Sie, dass bei diesem Verfahren derselbe Name für den DCE-Principal und die ID des lokalen Betriebssystems verwendet wird. Dies

bedeutet u. U., dass derselbe Wert, der im verschlüsselten DCE-Ticket enthalten ist, auch unverschlüsselt über die Leitung der Übertragungsebene geht.

Die DCE-Authentifizierung kann nur mit der APPC-Kommunikation verwendet werden, wenn der Parameter SECURITY auf den Wert NONE gesetzt ist. Der Grund dafür ist, dass die Möglichkeit ausgeschaltet werden soll, dass ein unverschlüsselter Principal und/oder ein Kennwort auf der Übertragungsebene gesendet wird, während ein verschlüsseltes DCE-Token für denselben Principal auf der Datenbankprotokollebene verwendet wird. Zum gegenwärtigen Zeitpunkt wird die DCE-Sicherheit auf der APPC-Ebene nicht unterstützt.

Authentifizierungsverarbeitung in zusammengeschlossenen Datenbanken

Wenn die Installationsfunktion für verteilte Verknüpfungen (Distributed Join) installiert ist und wenn die Konfigurationsvariable *federated* für den Datenbankmanager auf 'YES' gesetzt ist, wird das DB2-System als System zusammengeschlossener Datenbanken betrieben. Die Einstellungen zur Datenbankauthentifizierung unterscheiden sich in einem System zusammengeschlossener Datenbanken geringfügig von Standarddefinitionen in DB2. Wichtig ist jedoch, dass Sie in einem System zusammengeschlossener Datenbanken die Authentifizierungsanforderungen für die Datenquellen berücksichtigen müssen. Allgemein werden Datenquellen (DB2, Oracle, DB2 für OS/390 usw.) so definiert, dass für sie eine Authentifizierung erforderlich ist. Dies bedeutet, dass Sie sicherstellen müssen, dass die IDs und Kennwörter, falls erforderlich, an die Datenquellen gesendet werden können. DB2 bietet mehrere Möglichkeiten zur Authentifizierung bei den Datenquellen. Diese werden im folgenden Abschnitt erläutert.

Authentifizierungseinstellungen

SERVER

Gibt an, dass Clients, die mit DB2 eine Verbindung herstellen, eine Benutzer-ID und ein Kennwort senden, um auf DB2 zuzugreifen. Dazu ist eine Benutzer-ID und ein Kennwort zur Übertragung an die Datenquellen verfügbar. Sie steuern mit Serveroptionen und Benutzerzuordnungen, was an die Datenquellen tatsächlich übertragen wird. Die Authentifizierungsdaten sind jedoch zur Übertragung an die Datenquelle verfügbar.

CLIENT

Gibt an, dass die Authentifizierung mit Hilfe der Sicherheitseinrichtungen des Betriebssystems in der Datenbankpartition erfolgt, in der die Anwendung aufgerufen wird. Zur direkten Übertragung an die Datenquellen sind keine Kennwörter verfügbar. Wenn in diesem Fall eine Datenquelle die Authentifizierung erfordert, müssen Sie eine oder mehrere Benutzerzuordnungen erstellen. Außerdem müssen Sie sicherstellen, dass die Serveroptionen korrekt definiert sind, damit die richtige Benutzer-ID und das richtige Kennwort an die Datenquelle übertragen wird.

Gehen Sie sehr sorgfältig vor, wenn Sie die Authentifizierung CLIENT verwenden. Ziehen Sie diese Authentifizierungsart nur für sichere Netzwerke in Betracht. Ein Benutzer besitzt unter den folgenden Bedingungen die Berechtigung SYSADM für eine zusammengeschlossene Datenbank:

- Die Authentifizierung ist auf CLIENT gesetzt.
- Der Benutzer besitzt auf dem Client den Status Root.
- Der Benutzer kennt den Berechtigungsnamen für SYSADM.
- Der Benutzer definiert auf dem Client einen Berechtigungsnamen, der SYSADM für DB2 entspricht.

DCS

Gibt an, dass die Authentifizierung an der Datenquelle stattfindet, und nicht in DB2. In diesem Fall wird die Standardauthentifizierung von DB2 umgangen. Benutzer-IDs und Kennwörter werden direkt an die Datenquellen übergeben, wobei die Server-Einstellungen maßgebend sind. Die Authentifizierung erfolgt nur bei den Datenquellen Oracle oder der DB2-Produktfamilie.

Seien Sie vorsichtig mit der Einstellung DCS für die Authentifizierung. Weder beim Client noch in DB2 findet eine Authentifizierung statt. Alle Benutzer, die den Authentifizierungsnamen SYSADM kennen, können die Berechtigung SYSADM für den Server einer zusammengeschlossenen Datenbank erhalten.

DCE

Wenn die Authentifizierung auf DCE gesetzt ist, ist zur Übertragung an die Datenquellen nur eine Benutzer-ID verfügbar. Ein Kennwort ist nicht verfügbar. Wenn eine Datenquelle die Authentifizierungsverarbeitung erfordert (Benutzer-ID und Kennwort), müssen Sie eine Benutzerzuordnung definieren, mit der ein Kennwort und möglicherweise auch eine Benutzer-ID an die Datenquelle übertragen wird. Wenn an der Datenquelle die DB2-Verbindung als gesichert gilt, sind keine Benutzerzuordnungen erforderlich, da die vom externen Sicherheitssystem gesendete ID an die Datenquelle übergeben werden kann.

Es sind weitere DB2-Authentifizierungseinstellungen möglich, bei denen einige in DB2 ein Kennwort zur Übertragung an die Datenquellen verfügbar machen. Wenn die Einstellungen von DB2 und der Client-Authentifizierung die Übertragung eines Kennworts an DB2 bewirken, ist dieses Kennwort für eine weitere Authentifizierungsverarbeitung an den Datenquellen verfügbar. Weitere Informationen finden Sie in Tabelle 4 auf Seite 265.

Übergeben von Benutzer-IDs und Kennwörtern an Datenquellen

Zur Steuerung der Übertragung von Authentifizierungsdaten an Datenquellen gibt es die folgenden Methoden: DB2-Authentifizierungseinstellungen, Benutzerzuordnungen, Serveroptionen und APPC-Sicherheitseinstellungen. Diese werden im Folgenden erläutert.

Authentifizierungseinstellungen

In diesem Abschnitt wird erläutert, wie sich Authentifizierungseinstellungen auf die globale Authentifizierungsverarbeitung innerhalb eines Systems zusammengesetzter Datenbanken auswirkt. Weitere Informationen zu den Definitionen für Authentifizierungseinstellungen finden Sie im vorhergehenden Abschnitt „Authentifizierungseinstellungen“ auf Seite 278. Wenn z. B. die DB2-Authentifizierung auf SERVER oder DCS gesetzt ist, sind zur Herstellung einer Verbindung eine Benutzer-ID und ein Kennwort erforderlich. Deshalb ist eine Benutzer-ID und ein Kennwort zur Übertragung an die Datenquellen verfügbar. Wenn die Authentifizierung auf DCE oder CLIENT gesetzt ist und auf dem DB2-System keine Authentifizierung erfolgt, ist zur Übertragung an die Datenquellen nur eine Benutzer-ID verfügbar. Wenn die Authentifizierungsverarbeitung an der Datenquelle ein Kennwort oder möglicherweise eine andere Benutzer-ID oder ein anderes Kennwort erfordert, müssen Sie eine Benutzerzuordnung erstellen. Wenn die Authentifizierung auf CLIENT und der Parameter *trust_clntauth* auf SERVER gesetzt ist, ist es möglich, dass ein Kennwort an DB2 gesendet wird und dieses zur Übertragung an Datenquellen verfügbar ist.

Benutzerzuordnungen

DB2 kann entweder den Berechtigungsnamen, der zur Herstellung der Verbindung mit DB2 verwendet wird, oder einen Berechtigungsnamen, der in DB2 definiert ist, senden. In Benutzerzuordnungen werden die Berechtigungsnamen gespeichert, die in DB2 definiert sind. Diese werden mit der Anweisung CREATE USER MAPPING erstellt.

Benutzerzuordnungen sind flexibel: Sie können eine ID einer neuen ID und einem Kennwort oder nur einem Kennwort zuordnen. Sie können damit fehlende Daten angeben oder IDs und Kennwörter so ändern, dass die Daten von der Datenquelle angenommen werden.

Wenn Sie eine Benutzerzuordnung erstellen oder ändern wollen, müssen Sie die Berechtigung SYSADM oder DBADM besitzen, oder Ihre Authentifizierungs-ID muss dem Berechtigungsnamen entsprechen, der für die Anweisung angegeben wurde.

Mit der folgenden Anweisung können Sie z. B. eine Benutzerzuordnung erstellen:

```
CREATE USER MAPPING FOR "SHAWN" SERVER DB21 OPTIONS (REMOTE_AUTHID "SHAWNBCA",  
    REMOTE_PASSWORD "MAPLELEAF")
```

Dabei gilt Folgendes: Eine DB2-Authentifizierungs-ID (SHAWN) wird der fernen ID SHAWNBCA und dem fernen Kennwort MAPLELEAF für einen Server mit dem Namen DB21 zugeordnet.

Wenn der einzige Unterschied zwischen dem Berechtigungsnamen (oder dem Kennwort) in DB2 und dem Berechtigungsnamen (oder dem Kennwort) an der Datenquelle in der Groß-/Kleinschreibung der übergebenen Zeichenfolge besteht, können Sie bei Bedarf mit Hilfe von Serveroptionen die Groß-/Kleinschreibung in die gewünschte Schreibung umwandeln, statt neue IDs und Kennwörter zu erstellen. Weitere Informationen finden Sie in „Serveroptionen“.

Wenn die Authentifizierungseinstellung DCE lautet und für eine Datenquelle die Authentifizierungsverarbeitung erforderlich ist (wenn ein Kennwort erwartet wird), müssen Sie eine Benutzerzuordnung erstellen. DB2 übergibt nur die DCE-Benutzer-ID an Datenquellen. Dieser Benutzer-ID muss ein Kennwort zugeordnet werden, das anschließend an die Datenquelle gesendet wird.

Serveroptionen

Mit Serveroptionen kann die Authentifizierung global unterstützt werden. Damit können Sie angeben, ob Kennwörter an die Datenquellen übergeben werden (typischerweise ja) und ob Benutzer-IDs und Kennwörter in Großschreibung oder Kleinschreibung umgewandelt werden müssen. Die Serveroptionen werden mit den Anweisungen CREATE SERVER, ALTER SERVER und SET SERVER OPTION eingestellt.

Weitere Informationen zu Serveroptionen, die die Authentifizierungsverarbeitung betreffen, werden im folgenden Abschnitt erläutert. Eine vollständigere Liste der Serveroptionen finden Sie in „Vereinfachen der Datenquellendefinitionen und der Authentifizierungsverarbeitung durch Serveroptionen“ auf Seite 178.

Serveroption für das Kennwort: Der Standardwert für *password* ist *Y*. Damit werden Kennwörter an Datenquellen gesendet. In allen Fällen, in denen eine Datenquelle eine Authentifizierung ausführt und kein verschlüsseltes Kennwort erwartet, können Sie den Wert *Y* für diesen Parameter beibehalten oder festlegen.

DB2 kann verschlüsselte Kennwörter übertragen. Wenn die Kennwörter in verschlüsselter Form an Datenquellen der DB2-Produktfamilie gesendet werden sollen können Sie die Serveroption *password* auf *ENCRYPTION* setzen. Wenn die Authentifizierungseinstellung in DB2 *DCS_ENCRYPT* oder *SERVER_ENCRYPT* lautet, sollten Sie *password* auf *ENCRYPTION* setzen.

Eine Benutzer-ID wird immer an die Datenquellen gesendet.

Umsetzungsoptionen für ID und Kennwort: Die Berechtigungsnamen und die Kennwörter müssen in bestimmten Fällen geändert werden. Unterschiedliche Datenquellen können hinsichtlich der Groß-/Kleinschreibung unterschiedliche Berechtigungsnamen und Kennwörter erfordern.

Für DB2 gibt es zwei Serveroptionen, mit denen Namensunterschiede aufgelöst werden können. Diese Parameter heißen **fold_id** und **fold_pw** und können wie folgt eingestellt werden:

- 'U' DB2 wandelt den Berechtigungsnamen oder das Kennwort in Großbuchstaben um, bevor es an die Datenquelle gesendet wird.
- 'N' DB2 wandelt den Berechtigungsnamen oder das Kennwort nicht um.
- 'L' DB2 wandelt den Berechtigungsnamen oder das Kennwort in Kleinbuchstaben um, bevor es an die Datenquelle gesendet wird.
- 'null' DB2 sendet den Berechtigungsnamen oder das Kennwort zuerst in Großbuchstaben. Wenn dies fehlschlägt, wandelt DB2 diese in Kleinbuchstaben um und sendet sie erneut.

Die Einstellung *null* mag nützlich erscheinen, da mit ihr viele Möglichkeiten abgedeckt werden. Unter dem Gesichtspunkt der Leistung ist es jedoch am besten, diese Parameter so einzustellen, dass die Herstellung einer Verbindung nur einmal versucht wird. Wenn sowohl der Parameter *fold_id* als auch *fold_pw* auf *null* eingestellt sind, versucht DB2 möglicherweise viermal, den Berechtigungsnamen und das Kennwort zu senden:

1. Den Berechtigungsnamen und das Kennwort in Großbuchstaben
2. Den Berechtigungsnamen in Großbuchstaben und das Kennwort in Kleinbuchstaben
3. Den Berechtigungsnamen in Kleinbuchstaben und das Kennwort in Großbuchstaben
4. Den Berechtigungsnamen und das Kennwort in Kleinbuchstaben

APPC-Sicherheitseinstellungen

Wenn Sie mit einer DRDA-Datenquelle über APPC eine Verbindung herstellen, für die eine Benutzer-ID und ein Kennwort erforderlich ist, oder wenn die Authentifizierungseinstellung DCS lautet und an einer DRDA-Datenquelle authentifiziert wird, muss die APPC-Sicherheitseinstellung für die Verbindung zwischen DB2 und dieser Datenquelle auf PROGRAM gesetzt sein.

Beispiel für eine Authentifizierung für ein System zusammenschlossener Datenbanken

In diesem Abschnitt erhalten Sie eine Übersicht über die Authentifizierung und die Berechtigungsschritte für ein System zusammenschlossener Datenbanken. Eine Übersicht über die Datenbankauthentifizierung und die Berechtigungsverarbeitung ist in der Abb. 3 dargestellt.

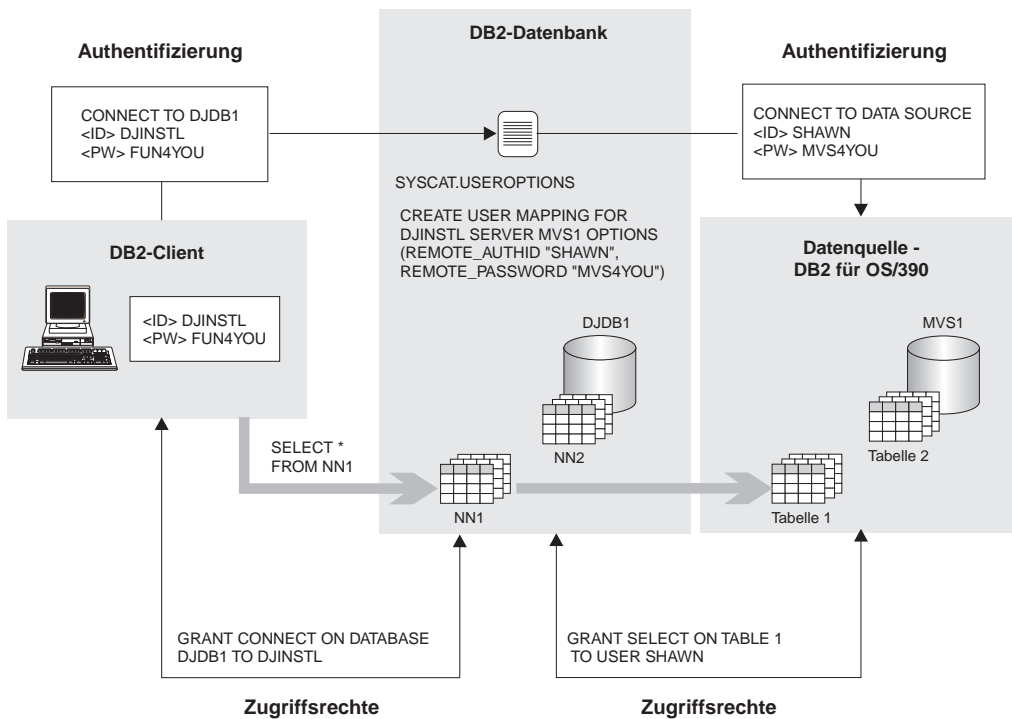


Abbildung 3. Authentifizierungs- und Berechtigungsverarbeitung in einer zusammenschlossenen Datenbank

In diesem Szenario stellt sich die Aufgabe, dem Benutzer DJINSTL eine UNION-Operation mit den zwei Kurznamen NN1 und NN2 zu ermöglichen. Die Kurznamen stellen zwei Tabellen dar. Die eine Datenquelle ist ein DB2 für OS/390-System, in dem DJINSTL eine andere Benutzer-ID und ein anderes Kennwort besitzt (siehe Abb. 3 auf Seite 283), und trägt den Namen MVS1. Zum Zugriff auf die Datenquelle MVS1 ist eine Benutzerzuordnung erforderlich. Die andere Datenquelle ist ein DB2-System, auf dem die ID und das Kennwort von DJINSTL gleich sind. Diese Datenquelle, die den Namen DB21 trägt, erfordert lediglich, dass die Benutzer-ID und das Kennwort in Großbuchstaben gesendet werden.

Die DB2-Authentifizierung ist auf SERVER gesetzt. DJINSTL greift von einem Windows NT-Client über eine TCP/IP-Verbindung auf DB2 zu. Die Verbindung von DB2 zu DB2 für OS/390 wird ebenfalls mit TCP/IP hergestellt. Der Name der zusammengeschlossenen Datenbank lautet DJDB1.

Zuerst müssen Sie sicherstellen, dass DB2 ein Kennwort erwartet und dass ein Kennwort gesendet wird. Stellen Sie außerdem sicher, dass die Authentifizierungsarten für den Client und den Server übereinstimmen. Sie können die Authentifizierungsart des DB2-Servers überprüfen, indem Sie den folgenden Befehl absetzen:

```
GET DATABASE MANAGER CONFIGURATION
```

Diesen Befehl müssen Sie vom DB2-Server aus ausführen. Sie können die Authentifizierungsart des Clients überprüfen, indem Sie den folgenden Befehl absetzen:

```
LIST DATABASE DIRECTORY
```

Diesen Befehl müssen Sie vom Client aus ausführen. In beiden Fällen ist die Authentifizierung auf SERVER gesetzt. Wenn die Einstellung Client DCS oder CLIENT lautet, können Sie diese mit den Befehlen UNCATALOG DATABASE und CATALOG DATABASE ändern.

Danach müssen Sie sicherstellen, dass die Kennwörter an die Datenquellen gesendet werden. Nach dem Herstellen der Verbindung mit der zusammengesetzten Datenbank DJDB1 können Sie die folgenden Befehle absetzen:

```
ALTER SERVER MVS1 OPTIONS (SET <kennwort> 'Y')
ALTER SERVER DB21 OPTIONS (SET <kennwort> 'Y')
```

Stellen Sie anschließend sicher, dass die Kennwörter in der richtigen Groß-/Kleinschreibung an die Datenquelle DB21 gesendet werden:

```
ALTER SERVER DB21 OPTIONS (ADD fold_id 'U')
ALTER SERVER DB21 OPTIONS (ADD fold_pw 'U')
```

Im nächsten Schritt erteilen Sie Zugriffsrechte, mit denen der Benutzer DJINSTL auf die zusammengesetzte Datenbank DJDB1 zugreifen kann, und wählen Kurznamen aus:

```
GRANT CONNECT ON DATABASE DJDB1 TO DJINSTL;
```

Nun ordnen Sie die DB2-ID und das DB2-Kennwort von DJINSTL der richtigen Benutzer-ID und dem richtigen Kennwort für den MVS1-Server zu:

```
CREATE USER MAPPING FOR "DJINSTL" SERVER MVS1 OPTIONS (REMOTE_AUTHID "SHAWN",
REMOTE_PASSWORD "MVS4YOU")
```

An dieser Stelle kann der DB2-Benutzer mit der Benutzer-ID DJINSTL Anforderungen an die Datenquellen senden. Möglicherweise sind weitere Schritte erforderlich, um auf Datenquellenobjekte zuzugreifen, auf die mit Kurznamen verwiesen wird. (Für Tabellen und Sichten, auf die mit Kurznamen verwiesen wird, sind normalerweise Zugriffsrechte erforderlich.)

Zugriffsrechte und Berechtigungen

Zugriffsrechte ermöglichen Benutzern, Datenbankressourcen zu erstellen oder auf sie zuzugreifen. *Berechtigungsstufen* bieten eine Möglichkeit zum Gruppieren von Zugriffsrechten sowie von Pflege- und Dienstprogrammoperationen des Datenbankmanagers höherer Ebene. Diese Funktionen dienen zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte. Benutzer können nur auf solche Objekte zugreifen, für die sie *berechtigt* sind, d. h., für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigungsstufe verfügen.

Es gibt die folgenden Berechtigungen:

- „Systemverwaltungs Berechtigung (SYSADM)“ auf Seite 288
- „Systemsteuerungs Berechtigung (SYSCTRL)“ auf Seite 289
- „Systempflege Berechtigung (SYSMAINT)“ auf Seite 290
- „Datenbankadministrator Berechtigung (DBADM)“ auf Seite 291
- „LOAD-Berechtigung“ auf Seite 292

Es gibt die folgenden Arten von Zugriffsrechten:

- „Zugriffsrechte für Datenbanken“ auf Seite 293
- „Zugriffsrechte für Schemata“ auf Seite 295
- „Zugriffsrechte für Tabellenbereiche“ auf Seite 297
- „Zugriffsrechte für Tabellen und Sichten“ auf Seite 297
- „Zugriffsrechte für Kurznamen“ auf Seite 300
- „Zugriffsrechte auf Server“ auf Seite 302
- „Zugriffsrechte für Pakete“ auf Seite 302
- „Zugriffsrechte für Indizes“ auf Seite 303.

In Abb. 4 auf Seite 287 ist die Abhängigkeit zwischen Berechtigungen und deren Gültigkeitsbereich (Datenbank, Datenbankmanager) dargestellt.

Berechtigungen

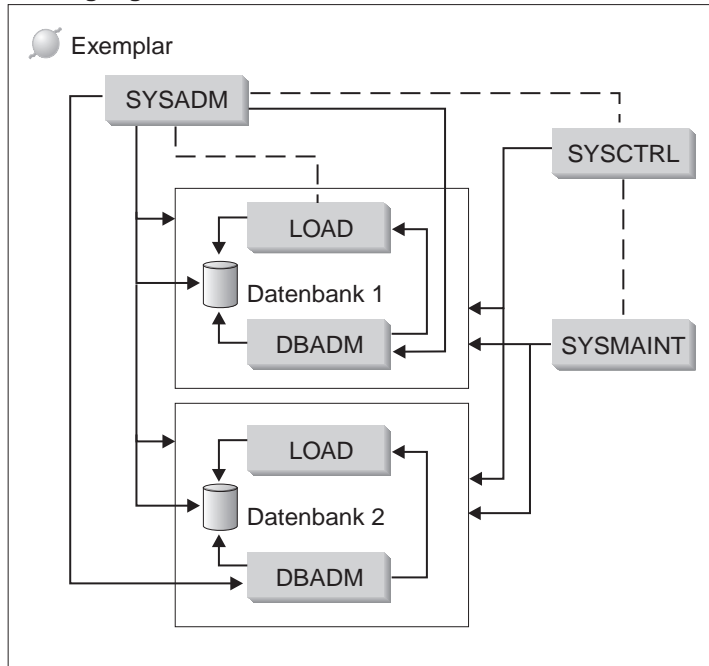


Abbildung 4. Hierarchie von Berechtigungen

Ein Benutzer oder eine Gruppe können über eine oder mehrere der nachfolgend aufgeführten Berechtigungsstufen verfügen:

- Administratorberechtigung (SYSADM oder DBADM) umfasst vollständige Zugriffsrechte für eine Menge von Objekten.
- Die Systemberechtigung (SYSCTRL oder SYSMAINT) bietet vollständige Zugriffsrechte für die Verwaltung des Systems, erlaubt aber keinen Zugriff auf die Daten.
- Die LOAD-Berechtigung (LOAD) gibt dem Dienstprogramm LOAD oder AutoLoader Zugriffsrechte, mit denen Daten in Tabellen geladen werden können.
- Das Eigentumszugriffsrecht (manchmal auch als Zugriffsrecht CONTROL bezeichnet) umfasst vollständige Zugriffsrechte für ein bestimmtes Objekt.
- Individuelle Zugriffsrechte können erteilt werden, um einem Benutzer die Ausführung bestimmter Funktionen für spezielle Objekte zu erlauben.
- Implizite Zugriffsrechte können einem Benutzer erteilt werden, der über das Zugriffsrecht zum Ausführen eines Pakets verfügt. Wenn Benutzer eine Anwendung ausführen können, benötigen sie nicht unbedingt explizite Zugriffsrechte für die im Paket verwendeten Datenobjekte. Weitere Informationen finden Sie in „Ermöglichen indirekter Zugriffsberechtigungen durch ein Paket“ auf Seite 309.

Benutzer mit Administratorberechtigung (SYSADM oder DBADM) oder Eigentumszugriffsrechten (CONTROL) können anderen Benutzern mit den Anweisungen GRANT und REVOKE Zugriffsrechte erteilen und entziehen. (Siehe „Steuern des Zugriffs auf Datenbankobjekte“ auf Seite 304.) Es ist außerdem möglich, ein Zugriffsrecht für eine Tabelle, eine Sicht oder ein Schema einem anderen Benutzer zu erteilen, wenn dieses Zugriffsrecht mit der Klausel WITH GRANT OPTION besteht. Jedoch erlaubt die Klausel WITH GRANT OPTION der Person, die das Zugriffsrecht erteilt, nicht, das erteilte Zugriffsrecht zu widerrufen. Sie müssen die Berechtigung SYSADM, DBADM oder das Zugriffsrecht CONTROL haben, um das Zugriffsrecht widerrufen zu können.

Ein Benutzer oder eine Gruppe kann für jede Kombination individueller Zugriffsrechte oder Berechtigungen berechtigt werden. Wenn ein Zugriffsrecht für eine Ressource zugeordnet wird, muss die betreffende Ressource existieren. Zum Beispiel kann einem Benutzer nur dann das Zugriffsrecht SELECT für eine Tabelle erteilt werden, wenn die Tabelle zuvor erstellt wurde.

Anmerkung: Vorsicht ist geboten, wenn einem Berechtigungsnamen Berechtigungen und Zugriffsrechte erteilt werden, und es keinen mit diesem Berechtigungsnamen erstellten Benutzer gibt. Später könnte ein Benutzer mit diesem Berechtigungsnamen erstellt werden und automatisch alle diesem Berechtigungsnamen zugeordneten Berechtigungen und Zugriffsrechte erhalten.

In den Handbüchern *Command Reference*, *Administrative API Reference* und *SQL Reference* finden Sie Informationen darüber, welche Berechtigungen für einen bestimmten Befehl, eine API oder eine SQL-Anweisung erforderlich sind.

Systemverwaltungsberechtigung (SYSADM)

Die Berechtigung SYSADM ist die höchste Stufe der Administratorberechtigung. Benutzer mit der Berechtigung SYSADM können Dienstprogramme ausführen, Befehle der Datenbank und des Datenbankmanagers ausführen und auf die Daten aller Tabellen in allen Datenbanken innerhalb des Datenbankmanagerexemplars zugreifen. Sie erteilt die Berechtigung, alle Datenbankobjekte im Exemplar zu steuern, einschließlich Datenbanken, Tabellen, Sichten, Indizes, Paketen, Schemata, Server, Aliasnamen, Datentypen, Funktionen, Prozeduren, Auslöser, Tabellenbereiche, Knotengruppen, Pufferpools und Ereignismonitoren.

Die Berechtigung SYSADM wird der Gruppe zugeordnet, die im Konfigurationsparameter *sysadm_group* angegeben ist (siehe Abschnitt zum „Konfigurieren von DB2“ im Handbuch *Systemverwaltung: Optimierung*). Die Zugehörigkeit zu dieser Gruppe wird außerhalb des Datenbankmanagers über die auf Ihrer Plattform verwendete Sicherheitseinrichtung gesteuert. Im Hand-

buch *Einstieg* finden Sie Informationen, wie Sie die Sicherheitseinrichtungen des Systems zum Erstellen, Ändern oder Löschen von SYSADM-Berechtigungen verwenden können.

Nur ein Benutzer mit der Berechtigung SYSADM kann folgende Funktionen ausführen:

- Umstellen einer Datenbank (Migration)
- Ändern der Konfigurationsdatei des Datenbankmanagers (einschließlich Angabe der Gruppen, die über die Berechtigung SYSCTRL oder SYSMaint verfügen)
- Erteilen der Berechtigung DBADM

Zusätzlich kann ein Benutzer mit der Berechtigung SYSADM die Funktionen von Benutzern mit folgenden Berechtigungen ausführen:

- „Systemsteuerungsberechtigung (SYSCTRL)“
- „Systempflegeberechtigung (SYSMAINT)“ auf Seite 290
- „Datenbankadministratorberechtigung (DBADM)“ auf Seite 291

Anmerkung: Wenn Benutzer mit der Berechtigung SYSADM Datenbanken erstellen, wird ihnen automatisch die explizite Berechtigung DBADM für die Datenbank erteilt. Wenn der Datenbankersteller aus der SYSADM-Gruppe gelöscht wird und Sie ihn auch daran hindern möchten, mit der Berechtigung DBADM auf die Datenbank zuzugreifen, müssen Sie diese Berechtigung DBADM explizit widerrufen.

Systemsteuerungsberechtigung (SYSCTRL)

Die Berechtigung SYSCTRL ist die höchste Stufe der Systemsteuerungsberechtigung. Diese Berechtigung bietet die Möglichkeit, Pflege- und Dienstprogrammoperationen für das Datenbankmanagerexemplar und seine Datenbanken auszuführen. Diese Operationen können die Systemressourcen beeinflussen, aber sie ermöglichen keinen direkten Zugriff auf Daten in den Datenbanken. Die Systemsteuerungsberechtigung ist für Benutzer gedacht, die ein Exemplar des Datenbankmanagers mit sensiblen Daten verwalten.

Die Berechtigung SYSCTRL wird der Gruppe zugeordnet, die im Konfigurationsparameter *sysctrl_group* angegeben ist (siehe Abschnitt zum „Konfigurieren von DB2“ im Handbuch *Systemverwaltung: Optimierung*). Wenn eine Gruppe angegeben wird, wird die Zugehörigkeit zu dieser Gruppe außerhalb des Datenbankmanagers durch die auf Ihrer Plattform verwendete Sicherheitseinrichtung gesteuert.

Nur ein Benutzer, der mindestens über die Berechtigung SYSCTRL verfügt, kann folgende Operationen ausführen:

- Aktualisieren einer Datenbank, eines Knotens oder eines DCS-Verzeichnisses (Distributed Connection Services)
- Benutzer zwangsweise aus dem System entfernen
- Erstellen oder Löschen einer Datenbank
- Löschen, Erstellen oder Ändern eines Tabellenbereichs
- Wiederherstellen in eine neue Datenbank

Zusätzlich kann ein Benutzer mit der Berechtigung SYSCTRL die Funktionen von Benutzern mit der „Systempflegeberechtigung (SYSMAINT)“ ausführen.

Benutzer mit der Berechtigung SYSCTRL verfügen auch über das implizite Zugriffsrecht, eine Verbindung zu einer Datenbank herzustellen.

Anmerkung: Wenn Benutzer mit der Berechtigung SYSCTRL Datenbanken erstellen, wird ihnen automatisch die explizite Berechtigung DBADM für die Datenbank erteilt. Wenn der Datenbankersteller aus der SYSCTRL-Gruppe gelöscht wird und Sie ihn auch daran hindern möchten, mit der Berechtigung DBADM auf die Datenbank zuzugreifen, müssen Sie diese Berechtigung DBADM explizit widerrufen.

Systempflegeberechtigung (SYSMAINT)

Die Berechtigung SYSMAINT ist die zweite Stufe der Systemsteuerungsberechtigung. Diese Berechtigung bietet die Möglichkeit, Pflege- und Dienstprogrammoperationen für das Datenbankmanagerexemplar und seine Datenbanken auszuführen. Diese Operationen können die Systemressourcen beeinflussen, aber sie ermöglichen keinen direkten Zugriff auf Daten in den Datenbanken. Die Systempflegeberechtigung ist für Benutzer konzipiert, die Datenbanken innerhalb eines Datenbankmanagerexemplars pflegen, das sensible Daten enthält.

Die Berechtigung SYSMAINT wird der Gruppe zugeordnet, die im Konfigurationsparameter *sysmaint_group* angegeben ist (siehe Abschnitt zum „Konfigurieren von DB2“ im Handbuch *Systemverwaltung: Optimierung*). Wenn eine Gruppe angegeben wird, wird die Zugehörigkeit zu dieser Gruppe außerhalb des Datenbankmanagers durch die auf Ihrer Plattform verwendete Sicherheitseinrichtung gesteuert.

Nur ein Benutzer mit der Berechtigung SYSMAINT oder einer höheren Systemberechtigung kann folgende Aktionen ausführen:

- Aktualisieren der Datenbankkonfigurationsdateien
- Sicherung einer Datenbank oder eines Tabellenbereichs
- Wiederherstellung in eine existierende Datenbank
- Ausführen einer aktualisierenden Wiederherstellung
- Starten oder Stoppen eines Exemplars
- Wiederherstellen eines Tabellenbereichs
- Ausführen einer Ablaufverfolgung
- Erstellen von Momentaufnahmen eines Datenbankmanagerexemplars oder seiner Datenbanken mit Hilfe des Datenbanksystemmonitors

Ein Benutzer mit der Berechtigung SYSMAINT, DBADM oder einer höheren Berechtigung kann Folgendes ausführen:

- Abfragen des Status eines Tabellenbereichs
- Aktualisieren von Protokolldateien
- Versetzen eines Tabellenbereichs in den Wartemodus (Quiesce)
- Neuorganisieren einer Tabelle
- Sammeln von Katalogstatistiken mit dem Dienstprogramm RUNSTATS

Benutzer mit der Berechtigung SYSMAINT haben auch die implizite Berechtigung, eine Verbindung zu einer Datenbank herzustellen.

Datenbankadministratorberechtigung (DBADM)

Die Berechtigung DBADM ist die zweithöchste Stufe der Administratorberechtigung. Sie bezieht sich nur auf eine bestimmte Datenbank und ermöglicht es dem Benutzer, bestimmte Dienstprogramme auszuführen, Datenbankbefehle auszuführen und auf die Daten in jeder Tabelle der Datenbank zuzugreifen. Wenn die Berechtigung DBADM erteilt wird, werden damit gleichzeitig die Zugriffsrechte BINDADD, CONNECT, CREATETAB, CREATE_NOT_FENCED und IMPLICIT_SCHEMA erteilt. Nur ein Benutzer mit der Berechtigung SYSADM kann die Berechtigung DBADM erteilen oder widerrufen. Benutzer mit der Berechtigung DBADM können anderen Benutzern Zugriffsrechte für die Datenbank erteilen und entziehen, unabhängig davon, wer diese Berechtigungen erteilt hat.

Nur ein Benutzer mit der Berechtigung DBADM oder einer höheren Berechtigung kann folgende Aktionen ausführen:

- Lesen von Protokolldateien
- Erstellen, Aktivieren und Löschen von Ereignismonitoren

Ein Benutzer mit der Berechtigung DBADM, SYSMANT oder einer höheren Berechtigung kann Folgendes ausführen:

- Abfragen des Status eines Tabellenbereichs
- Aktualisieren von Protokolldateien
- Versetzen eines Tabellenbereichs in den Wartemodus (Quiesce)
- Neuorganisieren einer Tabelle
- Sammeln von Katalogstatistiken mit dem Dienstprogramm RUNSTATS

Anmerkung: Ein Benutzer mit der Berechtigung DBADM kann die oben aufgeführten Funktionen nur für die Datenbank ausführen, für die er über die Berechtigung DBADM verfügt.

LOAD-Berechtigung

Benutzer, die über LOAD-Berechtigung auf Datenbankebene sowie INSERT-Zugriffsrechte für eine Tabelle verfügen, können den Befehl LOAD oder das Dienstprogramm AutoLoader verwenden, um Daten in eine Tabelle zu laden.

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

Wenn bei der vorangegangenen Ladeoperation Daten ersetzt wurden, muss diesem Benutzer auch das DELETE-Zugriffsrecht erteilt worden sein, damit dieser den Befehl LOAD RESTART oder LOAD TERMINATE verwenden kann.

Wenn die Ausnahmetabellen bei LOAD verwendet werden, muss der Benutzer über INSERT-Zugriffsrechte für die Ausnahmetabellen verfügen.

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

Zugriffsrechte für Datenbanken

In Abb. 5 sind die Datenbankzugriffsrechte dargestellt.

Datenbankzugriffsrechte

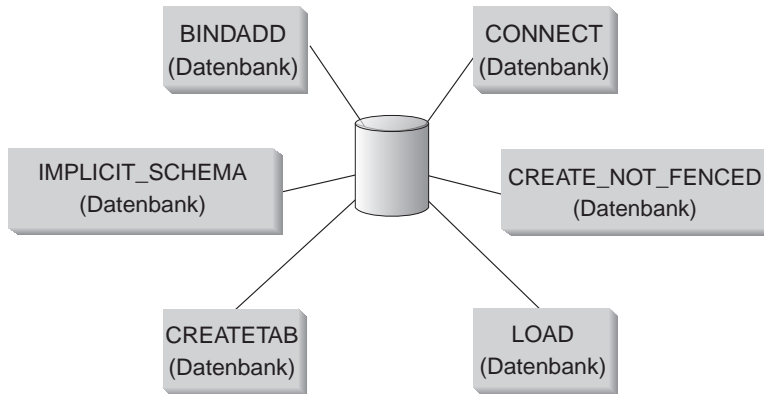


Abbildung 5. Zugriffsrechte für Datenbanken

Zugriffsrechte für Datenbanken umfassen Aktionen für eine Datenbank als Ganzes:

- CONNECT erlaubt einem Benutzer den Zugriff auf die Datenbank.
- BINDADD erlaubt einem Benutzer das Erstellen neuer Pakete in der Datenbank.
- CREATETAB erlaubt einem Benutzer das Erstellen neuer Tabellen in der Datenbank.
- CREATE_NOT_FENCED erlaubt einem Benutzer das Erstellen einer benutzerdefinierten Funktion (UDF) oder Prozedur, die nicht abgesichert („Not Fenced“) ist. Benutzerdefinierte Funktionen oder Prozeduren, die „nicht abgesichert“ sind, müssen äußerst gründlich getestet werden, weil der Datenbankmanager seinen Speicher und seine Steuerblöcke nicht vor diesen Funktionen bzw. Prozeduren schützt. (Infolgedessen kann eine fehlerhaft geschriebene und nicht gründlich getestete UDF oder Prozedur, die „nicht abgesichert“ ausgeführt wird, schwerwiegende Probleme für Ihr System verursachen.) Weitere Informationen finden Sie im Handbuch *Application Development Guide* oder *SQL Reference*.

- `IMPLICIT_SCHEMA` erlaubt jedem Benutzer das implizite Erstellen eines Schemas, indem er ein Objekt mit der Anweisung `CREATE` und einem Schemennamen, der noch nicht existiert, erstellt. `SYSIBM` wird zum Eigner des implizit erstellten Schemas, und die Gruppe `PUBLIC` erhält das Zugriffsrecht zur Erstellung von Objekten in diesem Schema.
- Mit `LOAD` kann ein Benutzer Daten in eine Tabelle laden.

Nur Benutzer mit der Berechtigung `SYSADM` oder `DBADM` können diese Zugriffsrechte anderen Benutzern erteilen und entziehen.

Anmerkung: Wenn eine Datenbank erstellt wird, werden die folgenden Zugriffsrechte automatisch der Gruppe `PUBLIC` erteilt:

- `CREATETAB`
- `BINDADD`
- `CONNECT`
- `IMPLICIT_SCHEMA`
- USE-Zugriffsrecht auf den Tabellenbereich `USERSPACE1`
- Zugriffsrecht `SELECT` für die Systemkatalogsichten

Um diese Zugriffsrechte zu widerrufen, muss ein Benutzer mit der Berechtigung `DBADM` oder `SYSADM` jedes Zugriffsrecht explizit der Gruppe `PUBLIC` entziehen.

Überlegungen zum Zugriffsrecht `IMPLICIT_SCHEMA`

Wenn eine neue Datenbank erstellt wird oder wenn eine Datenbank von einem vorigen Release-Stand umgestellt (Migration) wird, erhält die Gruppe `PUBLIC` das Zugriffsrecht `IMPLICIT_SCHEMA`. Mit dieser Berechtigung kann jeder Benutzer ein Schema erstellen, indem er ein Objekt erstellt und dabei einen Schemennamen angibt, der noch nicht existiert. `SYSIBM` wird zum Eigner des implizit erstellten Schemas, und die Gruppe `PUBLIC` erhält das Zugriffsrecht zur Erstellung von Objekten in diesem Schema.

Wenn für die Datenbank eine Kontrolle darüber erforderlich ist, wer implizit Schemenobjekte erstellen kann, sollte die Berechtigung `IMPLICIT_SCHEMA` der Gruppe `PUBLIC` entzogen werden. Wenn dies geschehen ist, gibt es nur drei Möglichkeiten zur Erstellung eines Schemenobjekts:

- Jeder Benutzer kann ein Schema erstellen, indem er seinen eigenen Berechtigungsnamen in der Anweisung `CREATE SCHEMA` verwendet.

- Jeder Benutzer mit der Berechtigung DBADM kann ein beliebiges, noch nicht vorhandenes Schema explizit erstellen und wahlfrei einen anderen Benutzer als Eigner des Schemas angeben.
- Jeder Benutzer mit der Berechtigung DBADM hat die Berechtigung IMPLICIT_SCHEMA (unabhängig von PUBLIC), so dass er implizit ein Schema mit einem beliebigen Namen bei der Erstellung anderer Datenbankobjekte erstellen kann. SYSIBM wird zum Eigner des implizit erstellten Schemas, und die Gruppe PUBLIC erhält das Zugriffsrecht zur Erstellung von Objekten in dem Schema.

Ein Benutzer hat immer die Möglichkeit, ein eigenes Schema unter Verwendung des eigenen Berechtigungsnamens explizit zu erstellen.

Zugriffsrechte für Schemata

Zugriffsrechte für Schemata gehören zur Kategorie der Zugriffsrechte für Objekte. Die Zugriffsrechte für Objekte sind in Abb. 6 auf Seite 296 dargestellt.

Objektzugriffsrechte

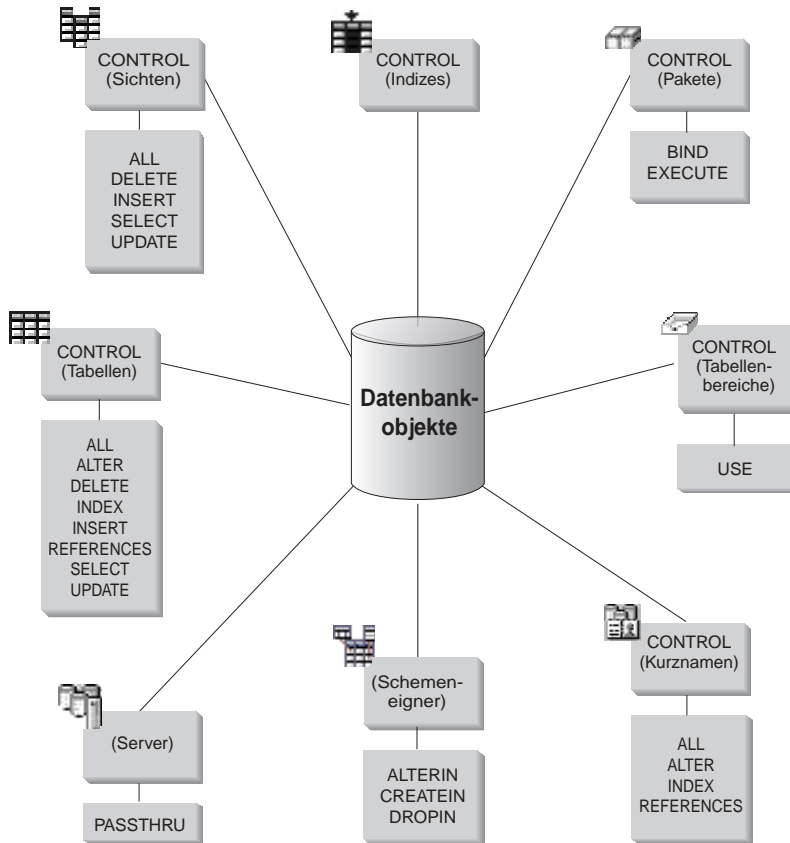


Abbildung 6. Zugriffsrechte für Objekte

Zugriffsrechte für Schemata berechtigen zu Aktionen an Schemata in einer Datenbank. Einem Benutzer kann jedes der folgenden Zugriffsrechte erteilt werden:

- CREATEIN erlaubt dem Benutzer, Objekte innerhalb des Schemas zu erstellen.
- ALTERIN erlaubt dem Benutzer, Objekte innerhalb des Schemas zu ändern.
- DROPIN erlaubt dem Benutzer, Objekte aus dem Schema zu löschen.

Der Eigner des Schemas besitzt alle diese Zugriffsrechte und die Möglichkeit, diese Zugriffsrechte anderen Benutzern zu erteilen. Zu den Objekten, die innerhalb des Schemenobjekts bearbeitet werden, gehören: Tabellen, Sichten, Indizes, Pakete, Datentypen, Funktionen, Auslöser, Prozeduren und Aliasnamen.

Zugriffsrechte für Tabellenbereiche

Mit den Zugriffsrechten auf Tabellenbereiche können Aktionen für die Tabellenbereiche einer Datenbank ausgeführt werden. Wenn einem Benutzer das Zugriffsrecht USE für einen Tabellenbereich zugeordnet wird, ist er zum Erstellen von Tabellen innerhalb des Tabellenbereichs berechtigt.

Der Eigner des Tabellenbereichs (normalerweise der Ersteller mit der Berechtigung SYSADM oder SYSCTRL) verfügt über das Zugriffsrecht USE und die Möglichkeit, dieses Zugriffsrecht auch anderen Benutzern zu erteilen. Standardmäßig wird beim Erstellen der Datenbank das Zugriffsrecht USE für den Tabellenbereich USERSPACE1 an PUBLIC erteilt. Dieses Zugriffsrecht kann jedoch auch widerrufen werden.

Das Zugriffsrecht USE kann bei SYSCATSPACE oder anderen temporären Systemtabellenbereichen nicht verwendet werden.

Zugriffsrechte für Tabellen und Sichten

Zugriffsrechte für Tabellen und Sichten umfassen Aktionen für Tabellen oder Sichten in einer Datenbank. Ein Benutzer muss über das Zugriffsrecht CONNECT für die Datenbank verfügen, um eines der folgenden Zugriffsrechte verwenden zu können:

- CONTROL erteilt dem Benutzer alle Zugriffsrechte für eine Tabelle oder Sicht, einschließlich der Erlaubnis, sie zu löschen und einzelne Zugriffsrechte für Tabellen zu erteilen oder zu widerrufen. Zur Erteilung des Zugriffsrechts CONTROL müssen Sie über die Berechtigung SYSADM oder DBADM verfügen. Der Ersteller einer Tabelle erhält automatisch das Zugriffsrecht CONTROL für die Tabelle. Der Ersteller einer Sicht erhält nur dann automatisch das Zugriffsrecht CONTROL, wenn er über das Zugriffsrecht CONTROL für alle Tabellen und Sichten verfügt, auf die in der Sichtdefinition verwiesen wird, oder wenn er über die Berechtigung SYSADM oder DBADM verfügt.

- ALTER erlaubt dem Benutzer das Hinzufügen von Spalten zu einer Tabelle, das Hinzufügen oder Ändern von Kommentaren für eine Tabelle und ihre Spalten, das Erstellen eines Primärschlüssels oder einer eindeutigen Integritätsbedingung sowie das Erstellen oder Löschen einer Prüfung auf Integritätsbedingung in einer Tabelle. Der Benutzer kann auch Auslöser für die Tabelle erstellen, allerdings wird eine zusätzliche Berechtigung für alle Objekte benötigt, auf die im Auslöser verwiesen wird (einschließlich SELECT für die Tabelle, wenn der Auslöser auf eine der Spalten in der Tabelle verweist). Ein Benutzer mit dem Zugriffsrecht ALTER für alle untergeordneten Tabellen kann einen Primärschlüssel löschen. Ein Benutzer mit dem Zugriffsrecht ALTER für die Tabelle und dem Zugriffsrecht REFERENCES für die übergeordnete Tabelle bzw. dem Zugriffsrecht REFERENCES für die entsprechenden Spalten kann einen Fremdschlüssel erstellen oder löschen. Ein Benutzer mit dem Zugriffsrecht ALTER kann außerdem die Anweisung COMMENT ON für eine Tabelle ausführen.
- DELETE erlaubt dem Benutzer das Löschen von Zeilen aus einer Tabelle oder Sicht.
- INDEX erlaubt dem Benutzer das Erstellen eines Index für eine Tabelle. Die Ersteller von Indizes verfügen automatisch über das Zugriffsrecht CONTROL für den Index. Weitere Informationen finden Sie in „Zugriffsrechte für Indizes“ auf Seite 303.
- INSERT erlaubt dem Benutzer das Einfügen einer Zeile in eine Tabelle oder Sicht und das Ausführen des Dienstprogramms IMPORT.
- REFERENCES erlaubt dem Benutzer das Erstellen und Löschen eines Fremdschlüssels unter Angabe der Tabelle als die übergeordnete Tabelle in einer Abhängigkeitsbeziehung. Der Benutzer hat dieses Zugriffsrecht möglicherweise nur für bestimmte Spalten.
- SELECT erlaubt dem Benutzer das Abrufen von Zeilen aus einer Tabelle oder Sicht, das Erstellen einer Sicht für eine Tabelle und das Ausführen des Dienstprogramms EXPORT.
- UPDATE erlaubt dem Benutzer das Ändern eines Eintrags in einer Tabelle oder Sicht bzw. für eine oder mehrere Spalten in einer Tabelle oder Sicht. Der Benutzer hat dieses Zugriffsrecht eventuell nur für bestimmte Spalten.

Das Zugriffsrecht zum Erteilen dieser Zugriffsrechte an andere Benutzer kann ebenfalls mit Hilfe der Klausel WITH GRANT OPTION in der Anweisung GRANT erteilt werden.

Anmerkung: Wenn einem Benutzer oder einer Gruppe das Zugriffsrecht CONTROL für eine Tabelle erteilt wird, werden alle anderen Zugriffsrechte für diese Tabelle automatisch mit der Klausel WITH GRANT OPTION erteilt. Wenn Sie anschließend das Zugriffsrecht CONTROL eines Benutzers für die Tabelle widerrufen, behält der Benutzer weiterhin die anderen Zugriffsrechte, die ihm automatisch erteilt wurden. Wenn Sie alle Zugriffsrechte widerrufen wollen, die zusammen mit dem Zugriffsrecht CONTROL erteilt wurden, müssen Sie entweder jedes einzelne Zugriffsrecht explizit widerrufen oder das Schlüsselwort ALL in der Anweisung REVOKE angeben wie im folgenden Beispiel:

```
REVOKE ALL
ON EMPLOYEE FROM USER HERON
```

Bei der Arbeit mit typisierten Tabellen sind folgende Punkte für die Tabellen- und Sichtzugriffsrechte zu berücksichtigen.

Anmerkung: Zugriffsrechte können auf jeder Stufe einer Tabellenhierarchie unabhängig erteilt werden. Daher kann ein Benutzer, dem ein Zugriffsrecht für eine übergeordnete Tabelle innerhalb einer Hierarchie typisierter Tabellen erteilt wurde, indirekt auch beliebige untergeordnete Tabellen beeinflussen. Ein Benutzer kann jedoch nur direkt auf eine untergeordnete Tabelle einwirken, wenn er über das nötige Zugriffsrecht für diese untergeordnete Tabelle verfügt.

Die Abhängigkeiten von über- und untergeordneten Tabellen in einer Tabellenhierarchie bedeuten, dass Operationen wie SELECT, UPDATE und DELETE die Zeilen der Zieltabelle für die Operation und alle seine untergeordneten Tabellen (sofern vorhanden) beeinflussen. Dieses Verhalten wird als „Substitutionsfähigkeit“ bezeichnet. Angenommen Sie haben z. B. eine Mitarbeitertabelle des Typs Mitarbeiter_t erstellt, zu der eine untergeordnete Tabelle des Typs Manager_t gehört. Ein Manager ist eine (besondere) Art von Mitarbeiter, wie durch die Abhängigkeit von Typ und untergeordnetem Typ für die strukturierten Typen Mitarbeiter_t und Manager_t und durch die entsprechende Abhängigkeit von Tabelle und untergeordneter Tabelle für die Tabellen Mitarbeiter und Manager angegeben wird. Die SQL-Abfrage

```
SELECT * FROM Mitarbeiter
```

gibt daher als Ergebnis dieser Abhängigkeit die Objektkennung und die Mitarbeiter_t-Attribute für Mitarbeiter und Manager zurück. Gleichmaßen gibt die Aktualisierungsoperation

```
UPDATE
Mitarbeiter SET Gehalt = Gehalt + 1000
```

eine Gehaltserhöhung von eintausend DM für Manager sowie gewöhnliche Mitarbeiter zurück.

Ein Benutzer mit dem Zugriffsrecht SELECT für die Tabelle Mitarbeiter kann diese SELECT-Operation ausführen, selbst wenn er kein explizites Zugriffsrecht SELECT für die Tabelle Manager hat. Einem Benutzer wird jedoch nicht erlaubt, eine SELECT-Operation direkt für die untergeordnete Tabelle Manager auszuführen, und daher kann er nicht auf die nicht übernommenen Spalten der Tabelle Manager zugreifen.

Gleichermaßen kann ein Benutzer mit dem Zugriffsrecht UPDATE für die Tabelle Mitarbeiter eine UPDATE-Operation für die Tabelle Manager ausführen, wodurch er gewöhnliche Mitarbeiter und Manager beeinflusst, selbst wenn er nicht über das explizite Zugriffsrecht UPDATE für die Tabelle Manager verfügt. Einem Benutzer wird jedoch nicht erlaubt, UPDATE-Operationen direkt für die untergeordnete Tabelle Manager auszuführen, und daher kann er die nicht übernommenen Spalten der Tabelle Manager nicht aktualisieren.

In den folgenden Handbüchern finden Sie Informationen über die Berechtigungen, die zum Ausführen bestimmter Befehle, APIs oder SQL-Anweisungen erforderlich sind:

- *SQL Reference*
- *Command Reference*
- *Administrative API Reference.*

Weitere Informationen zu den Berechtigungen, die zum Aktualisieren von Katalogstatistiken erforderlich sind, finden Sie im Handbuch *Systemverwaltung: Optimierung*.

Weitere Informationen zum Festlegen von Zugriffsrechten für Sichten finden Sie im Abschnitt über die Anweisung CREATE VIEW im Handbuch *SQL Reference*.

Zugriffsrechte für Kurznamen

Zugriffsrechte für Kurznamen berechtigen zu Aktionen an Kurznamen in einer Datenbank. Diese Zugriffsrechte wirken sich nicht auf Zugriffsrechte für Datenquellenobjekte aus, auf die mit Kurznamen verwiesen wird. Ein Benutzer muss über das Zugriffsrecht CONNECT für die Datenbank verfügen, um eines der folgenden Zugriffsrechte verwenden zu können:

- CONTROL erteilt dem Benutzer alle Zugriffsrechte für einen Kurznamen, einschließlich der Erlaubnis, sie zu löschen und einzelne Zugriffsrechte für Kurznamen zu erteilen oder zu widerrufen. Zur Erteilung des Zugriffsrechts CONTROL müssen Sie über die Berechtigung SYSADM oder DBADM verfügen. Der Ersteller eines Kurznamens erhält automatisch das Zugriffsrecht CONTROL für den Kurznamen.

- Mit ALTER kann der Benutzer Spaltennamen in einem Kurznamen ändern, den DB2-Typ, dem der Datentyp der Spalte zugeordnet ist, ändern oder hinzufügen und Optionen für die Kurznamenspalten festlegen.
- INDEX erlaubt dem Benutzer das Erstellen einer Indexangabe für einen Kurznamen. Die Ersteller von Indexangaben verfügen automatisch über das Zugriffsrecht CONTROL für den Index.
- REFERENCES erlaubt dem Benutzer das Erstellen und Löschen eines Fremdschlüssels unter Angabe des Kurznamens als übergeordneten Kurznamen in einer Abhängigkeitsbeziehung. Der Benutzer hat dieses Zugriffsrecht eventuell nur für bestimmte Spalten.

Das Zugriffsrecht zum Erteilen dieser Zugriffsrechte an andere Benutzer kann ebenfalls mit Hilfe der Klausel WITH GRANT OPTION in der Anweisung GRANT erteilt werden.

Anmerkung: Wenn einem Benutzer oder einer Gruppe das Zugriffsrecht CONTROL für einen Kurznamen erteilt wird, werden alle anderen Zugriffsrechte für diesen Kurznamen automatisch mit der Klausel WITH GRANT OPTION erteilt. Wenn Sie anschließend das Zugriffsrecht CONTROL eines Benutzers für den Kurznamen widerrufen, behält der Benutzer weiterhin die anderen Zugriffsrechte, die ihm automatisch erteilt wurden.

Wenn Sie auf Daten der Datenquelle zugreifen möchten, müssen Sie außerdem die geeignete Berechtigung für die Objekte an diesen Datenquellen besitzen, auf die mit den Kurznamen verwiesen wird.

Wenn ein Benutzer auf eine Sicht zugreift, auf die mit einem oder mehreren Kurznamen verwiesen wird, muss dieser Benutzer zum Zugriff auf diese Sicht und auf die Objekte berechtigt sein, auf die mit den Kurznamen an den Datenquellen verwiesen wird.

Zugriffsrechte auf Server

Es gibt ein Zugriffsrecht für Server: PASSTHRU. Mit diesem Zugriffsrecht werden die Berechtigungs-IDs gesteuert, die direkt DDL- und DML-Anweisungen an die Datenquellen senden können (Durchgriffoperationen).

DB2 unterstützt zwei SQL-Anweisungen, um Durchgriffoperationen zu steuern:

- GRANT PASSTHRU, womit die Berechtigung erteilt wird, SET PASSTHRU-Anweisungen für die Datenquelle und DML- und DDL-Durchgriffsanweisungen an diese Datenquelle abzusetzen.
- REVOKE PASSTHRU, womit die Berechtigung aufgehoben wird, SET PASSTHRU-Anweisungen für die Datenquelle und DML- und DDL-Durchgriffsanweisungen an diese Datenquelle abzusetzen.

Ein Beispiel für eine Anweisung, mit der dem Benutzer SHAWN die Berechtigung zu einer Durchgriffoperation für den Server ORACLE1 erteilt wird, kann wie folgt lauten:

```
GRANT PASSTHRU ON SERVER ORACLE1 TO USER SHAWN
```

Vollständige Informationen zur Syntax von PASSTHRU-Anweisungen finden Sie im Handbuch *SQL Reference*.

Zugriffsrechte für Pakete

Ein Paket ist ein Datenbankobjekt, das die Informationen enthält, die vom Datenbankmanager zum effizientesten Zugriff auf Daten für ein bestimmtes Anwendungsprogramm benötigt werden. Zugriffsrechte für Pakete erlauben einem Benutzer das Erstellen und Bearbeiten von Paketen. Der Benutzer muss über das Zugriffsrecht CONNECT für die Datenbank verfügen, um eines der folgenden Zugriffsrechte verwenden zu können:

- CONTROL gibt dem Benutzer die Möglichkeit, ein Paket erneut zu binden, zu löschen oder auszuführen sowie die Möglichkeit, die Zugriffsrechte auf andere Benutzer zu erweitern. Der Ersteller eines Pakets erhält dieses Zugriffsrecht automatisch. Einem Benutzer mit dem Zugriffsrecht CONTROL werden die Zugriffsrechte BIND und EXECUTE erteilt, und er kann anderen Benutzern ebenfalls die Zugriffsrechte BIND und EXECUTE erteilen. Um das Zugriffsrecht CONTROL erteilen zu können, muss der Benutzer über die Berechtigung SYSADM oder DBADM verfügen.
- BIND erlaubt dem Benutzer das erneute Binden eines existierenden Pakets.
- EXECUTE erlaubt dem Benutzer das Ausführen eines Pakets.

Zusätzlich zu diesen Zugriffsrechten für Pakete erlaubt das Zugriffsrecht BINDADD für Datenbanken Benutzern das Erstellen neuer Pakete oder das erneute Binden eines vorhandenen Pakets in der Datenbank.

Benutzer mit der Berechtigung, ein Paket mit Kurznamen auszuführen, benötigen für die Kurznamen innerhalb des Pakets keine weiteren Zugriffsrechte und keine Berechtigungsstufe. Sie müssen jedoch an denjenigen Datenquellen Authentifizierungen durchlaufen, an denen die Objekte sich befinden, auf die mit den Kurznamen verwiesen wurde. Außerdem müssen Paketbenutzer über geeignete Zugriffsrechte oder eine geeignete Berechtigungsstufe für die Datenquellenobjekte an der Datenquelle verfügen.

Für Pakete mit Kurznamen sind möglicherweise weitere Berechtigungsschritte erforderlich, da DB2 dynamisches SQL verwendet, um mit Datenquellen der DB2-Produktfamilie Daten auszutauschen. Die Berechtigungs-ID, die das Paket an der Datenquelle ausführt, muss mit der entsprechenden Berechtigung ausgestattet sein, um das Paket an dieser Datenquelle dynamisch ausführen zu können. Weitere Informationen dazu, wie DB2 statisches und dynamisches SQL verarbeitet, finden Sie im Handbuch *SQL Reference*.

Zugriffsrechte für Indizes

Der Ersteller eines Index oder einer Indexspezifikation erhält automatisch das Zugriffsrecht CONTROL für den Index. Das Zugriffsrecht CONTROL für einen Index entspricht tatsächlich der Berechtigung zum Löschen des Index. Zum Erteilen eines Zugriffsrechts CONTROL für einen Index muss ein Benutzer über die Berechtigung SYSADM oder DBADM verfügen.

Das Zugriffsrecht INDEX auf Tabellenebene erlaubt einem Benutzer das Erstellen eines Index für diese Tabelle (siehe „Zugriffsrechte für Tabellen und Sichten“ auf Seite 297).

Zugriffsrechte für Sequenzen

Der Ersteller einer Sequenz erhält das Zugriffsrecht USAGE automatisch. Das Zugriffsrecht USAGE ist für die Verwendung der Ausdrücke NEXTVAL und PREVVAL für die Sequenz erforderlich. Um anderen Benutzern die Verwendung der Ausdrücke NEXTVAL und PREVVAL zu erlauben, müssen die Sequenzzugriffsrechte für PUBLIC erteilt werden. Dadurch erhalten alle Benutzer die Erlaubnis die Ausdrücke mit der angegebenen Sequenz zu verwenden.

Steuern des Zugriffs auf Datenbankobjekte

Für die Steuerung des Datenzugriffs müssen Sie mit direkten und indirekten Zugriffsrechten, Administratorberechtigungen und Paketen vertraut sein. In diesem Abschnitt werden diese Themen behandelt und einige Beispiele gegeben.

Direkt erteilte Zugriffsrechte werden im Systemkatalog gespeichert. Eine Beschreibung der Methoden, mit denen die Implementierung des Plans zur Steuerung des Datenbankzugriffs überprüft werden kann, finden Sie in „Verwenden des Systemkatalogs“ auf Seite 317.

Berechtigungen werden auf drei Arten gesteuert:

- Die explizite Berechtigung wird über Zugriffsrechte, deren Steuerung mit den Anweisungen GRANT und REVOKE erfolgt, gesteuert.
- Die implizite Berechtigung wird durch Erstellen und Löschen von Objekten gesteuert.
- Indirekte Zugriffsrechte sind Paketen zugeordnet.

Die folgenden Themen werden behandelt:

- „Erteilen von Zugriffsrechten“
- „Widerrufen von Zugriffsrechten“ auf Seite 305
- „Implizite Berechtigungen durch Erstellen und Löschen von Objekten“ auf Seite 308
- „Ermöglichen indirekter Zugriffsberechtigungen durch ein Paket“ auf Seite 309
- „Steuern des Zugriffs auf Daten mit Sichten“ auf Seite 311
- „Überwachen des Zugriffs auf Daten mit Hilfe der Audit-Einrichtung“ auf Seite 314.

Erteilen von Zugriffsrechten

Mit der Anweisung GRANT kann ein berechtigter Benutzer Zugriffsrechte erteilen. Ein Zugriffsrecht kann einem oder mehreren Berechtigungsnamen in einer Anweisung erteilt werden. Das Zugriffsrecht kann auch der Gruppe PUBLIC erteilt werden, wodurch es allen Benutzern zur Verfügung gestellt wird. Bei einem Berechtigungsnamen kann es sich um einen Einzelbenutzer oder um eine Gruppe handeln.

Bei Betriebssystemen, unter denen Benutzer denselben Namen wie Gruppen haben können, sollten Sie angeben, ob Sie das Zugriffsrecht dem Benutzer oder der Gruppe erteilen. Die Anweisungen GRANT und REVOKE unterstützen beide die Schlüsselwörter USER und GROUP. Wenn diese wahlfreien Schlüsselwörter nicht verwendet werden, überprüft der Datenbankmanager die Sicherheitseinrichtung des Betriebssystems, um zu ermitteln, ob der

Berechtigungsname einen Benutzer oder eine Gruppe bezeichnet. Wenn es sich bei dem Berechtigungsnamen sowohl um einen Benutzer als auch um eine Gruppe handeln könnte, wird ein Fehler zurückgegeben.

Im folgenden Beispiel wird das Zugriffsrecht SELECT für die Tabelle EMPLOYEE dem Benutzer HERON erteilt:

```
GRANT SELECT
    ON EMPLOYEE TO USER HERON
```

Im folgenden Beispiel wird das Zugriffsrecht SELECT für die Tabelle EMPLOYEE der Gruppe HERON erteilt:

```
GRANT SELECT
    ON EMPLOYEE TO GROUP HERON
```

Damit ein Benutzer Zugriffsrechte für die meisten Datenbankobjekte erteilen kann, muss er über die Berechtigung SYSADM, die Berechtigung DBADM oder das Zugriffsrecht CONTROL für das betreffende Objekt verfügen. Oder der Benutzer muss das jeweilige Zugriffsrecht mit der Klausel WITH GRANT OPTION besitzen. Zugriffsrechte können nur für existierende Objekte erteilt werden. Damit ein Benutzer das Zugriffsrecht CONTROL an einen anderen Benutzer erteilen kann, muss er über die Berechtigung SYSADM oder DBADM verfügen. Zum Erteilen der Berechtigung DBADM muss der Benutzer die Berechtigung SYSADM haben.

Weitere Informationen zur Anweisung GRANT finden Sie im Handbuch *SQL Reference*.

Widerrufen von Zugriffsrechten

Die Anweisung REVOKE erlaubt es berechtigten Benutzern, Zugriffsrechte zu widerrufen, die anderen Benutzern zuvor erteilt wurden. Zum Widerrufen von Zugriffsrechten für Datenbankobjekte müssen Sie über die Berechtigung DBADM, die Berechtigung SYSADM oder das Zugriffsrecht CONTROL für das betreffende Objekt verfügen. Beachten Sie, dass es nicht ausreicht, ein Zugriffsrecht mit der Klausel WITH GRANT OPTION zu besitzen, um es widerrufen zu können. Zum Widerrufen des Zugriffsrechts CONTROL eines anderen Benutzers müssen Sie die Berechtigung SYSADM oder DBADM haben. Zum Widerrufen der Berechtigung DBADM müssen Sie über die Berechtigung SYSADM verfügen. Zugriffsrechte können nur für existierende Objekte widerrufen werden.

Anmerkung: Ein Benutzer ohne die Berechtigung DBADM oder das Zugriffsrecht CONTROL für eine Tabelle oder Sicht kann kein Zugriffsrecht widerrufen, das er durch Verwendung der Klausel WITH GRANT OPTION erteilt hat. Auch gibt es kein benutzerübergreifendes Widerrufen für die Benutzer, die Zugriffsrechte besitzen, die ihnen von der Person, deren Zugriffsrechte widerrufen werden, erteilt wurden. Weitere Informationen zur Berechtigung, die zum Widerrufen von Zugriffsrechten erforderlich ist, finden Sie im Handbuch *SQL Reference*.

Wenn ein Zugriffsrecht einem Benutzer und einer Gruppe mit demselben Namen erteilt wurde, müssen Sie beim Widerrufen des Zugriffsrechts das Schlüsselwort GROUP oder USER angeben. Im folgenden Beispiel wird das Zugriffsrecht SELECT für die Tabelle EMPLOYEE des Benutzers HERON widerrufen:

```
REVOKE SELECT
    ON EMPLOYEE FROM USER HERON
```

Im folgenden Beispiel wird das Zugriffsrecht SELECT für die Tabelle EMPLOYEE der Gruppe HERON widerrufen:

```
REVOKE SELECT
    ON EMPLOYEE FROM GROUP HERON
```

Zu beachten ist, dass beim Widerrufen eines Zugriffsrechts für eine Gruppe nicht notwendigerweise das Zugriffsrecht jedes einzelnen Angehörigen dieser Gruppe widerrufen wird. Wenn einem einzelnen Namen ein Zugriffsrecht direkt erteilt wurde, bleibt dieses Zugriffsrecht erhalten, bis es direkt widerrufen wird.

Wenn ein Zugriffsrecht auf eine Tabelle für einen Benutzer widerrufen wird, werden auch Zugriffsrechte für alle Sichten widerrufen, die von dem betreffenden Benutzer erstellt wurden und die von dem widerrufenen Zugriffsrecht auf die Tabelle abhängen. Es werden jedoch nur die vom System implizit erteilten Zugriffsrechte widerrufen. Wenn ein Zugriffsrecht für die Sicht direkt von einem anderen Benutzer erteilt wurde, bleibt dieses erhalten.

Es kann zu Situationen kommen, in denen Sie einer Gruppe ein Zugriffsrecht erteilen (GRANT) und nur für ein Mitglied dieser Gruppe widerrufen (REVOKE) möchten. Es gibt nur wenige Möglichkeiten, dies zu tun, ohne die Fehlermeldung SQL0556N zu erhalten.:

- Sie können das Mitglied aus der Gruppe entfernen oder eine neue Gruppe mit weniger Mitgliedern erstellen und dieser neuen Gruppe das Zugriffsrecht erteilen (GRANT).
- Sie können das Zugriffsrecht für die Gruppe widerrufen (REVOKE) und es anschließend einzelnen Benutzern (Berechtigungs-IDs) erteilen (GRANT).

Wenn ein explizit erteiltes Zugriffsrecht auf eine Tabelle (oder eine Sicht) für einen Benutzer mit der Berechtigung DBADM widerrufen wird, werden die Zugriffsrechte für andere Sichten, die für diese Tabelle definiert sind, **nicht** widerrufen. Der Grund hierfür ist, dass die Zugriffsrechte für die Sichten auf der Berechtigung DBADM basieren und nicht von expliziten Zugriffsrechten auf die entsprechenden Tabellen abhängig sind.

Wenn Sie eine Sicht auf eine oder mehrere zugrunde liegende Tabellen oder Sichten definiert haben und Sie das Zugriffsrecht für eine oder mehrere dieser Tabellen oder Sichten verlieren, kann die Sicht nicht verwendet werden.

Anmerkung: Wenn einem Benutzer das Zugriffsrecht CONTROL für eine Tabelle oder Sicht entzogen wird, behält der Benutzer die Möglichkeit, Zugriffsrechte anderen Benutzern zu erteilen. Durch Erhalt des Zugriffsrechts CONTROL empfängt der Benutzer auch alle anderen Zugriffsrechte mit der Klausel WITH GRANT OPTION. Wird das Zugriffsrecht CONTROL widerrufen, bleiben alle anderen Zugriffsrechte mit der Klausel WITH GRANT OPTION erhalten, bis sie explizit widerrufen werden.

Alle Pakete, die von widerrufenen Zugriffsrechten abhängig sind, werden als ungültig markiert, können jedoch wieder zu gültigen Paketen werden, wenn sie von einem Benutzer mit der entsprechenden Berechtigung erneut gebunden werden. Pakete können auch wiederhergestellt werden, wenn die Zugriffsrechte später dem Benutzer, der die Anwendung gebunden hat, wieder erteilt werden. Durch Ausführen der Anwendung wird implizit ein erfolgreiches erneutes Binden ausgelöst. Wenn Zugriffsrechte der Gruppe PUBLIC widerrufen werden, werden alle Pakete ungültig gemacht, die von Benutzern gebunden wurden, die dies nur aufgrund der Zugriffsrechte der Gruppe PUBLIC tun konnten. Wenn die Berechtigung DBADM einem Benutzer entzogen wird, werden alle von diesem Benutzer gebundenen Pakete ungültig gemacht, einschließlich der Pakete, die Dienstprogrammen für Datenbanken zugeordnet sind. Wenn versucht wird, ein Paket zu verwenden, das als ungültig markiert wurde, versucht das System, dieses erneut zu binden. Wenn dieses erneute Binden fehlschlägt, tritt ein Fehler auf (SQLCODE -727). In diesem Fall müssen die Pakete von einem Benutzer explizit neu gebunden werden, der über Folgendes verfügt:

- Die Berechtigung zum erneuten Binden der Pakete
- Entsprechende Berechtigungen für die Objekte innerhalb der Pakete

Diese Pakete sollten zu dem Zeitpunkt, an dem die Zugriffsrechte widerrufen werden, neu gebunden werden. Weitere Informationen zu den Anweisungen REVOKE und REBIND PACKAGE finden Sie im Handbuch *SQL Reference*.

Wenn Sie einen Auslöser in Abhängigkeit von einem oder mehreren Zugriffsrechten definiert haben und Sie eines oder mehrere dieser Zugriffsrechte verlieren, kann der Auslöser nicht verwendet werden.

Implizite Berechtigungen durch Erstellen und Löschen von Objekten

Der Datenbankmanager erteilt implizit bestimmte Zugriffsrechte an einen Benutzer, der eine Anweisung CREATE SCHEMA, CREATE TABLESPACE, CREATE TABLE, CREATE VIEW oder CREATE INDEX absetzt oder mit dem Befehl PREP oder BIND ein neues Paket erstellt. Es werden auch Zugriffsrechte erteilt, wenn Objekte von Benutzern mit der Berechtigung SYSADM oder DBADM erstellt werden. In ähnlicher Weise werden Zugriffsrechte entfernt, wenn ein Objekt gelöscht wird.

Wenn das erstellte Objekt ein Tabellenbereich, eine Tabelle, ein Index oder ein Paket ist, erhält der Benutzer das Zugriffsrecht CONTROL für das Objekt. Wenn das Objekt eine Sicht ist, wird das Zugriffsrecht CONTROL für die Sicht nur dann implizit erteilt, wenn der Benutzer über das Zugriffsrecht CONTROL für alle Tabellen und Sichten verfügt, auf die in der Sichtdefinition verwiesen wird.

Wenn das explizit erstellte Objekt ein Schema ist, erhält der Schemeneigner die Zugriffsrechte ALTERIN, CREATEIN und DROPIN mit der Klausel WITH GRANT OPTION. Für ein implizit erstelltes Schema wird das Zugriffsrecht CREATEIN der Gruppe PUBLIC erteilt.

Weitere Informationen zum Festlegen von Zugriffsrechten für Sichten finden Sie im Abschnitt über die Anweisung CREATE VIEW im Handbuch *SQL Reference*.

Einrichten eines Eigentumsrechts für einen Plan oder ein Paket

Mit den Befehlen BIND und PRECOMPILE kann ein Anwendungspaket erstellt oder geändert werden. Für jeden der Befehle können Sie die Option OWNER angeben, um den Eigner des generierten Pakets zu benennen. Für die Benennung des Paketeigners gibt es die folgenden einfachen Regeln:

- Jeder Benutzer kann sich selbst als Eigner benennen. Dies ist die Standardeinstellung, falls die Option OWNER nicht angegeben wird.
- Eine ID mit der Berechtigung SYSADM oder DBADM darf eine beliebige Berechtigungs-ID mit der Option OWNER als Eigner benennen.

Nicht alle Betriebssysteme, die mit DB2-Datenbankprodukten ein Paket binden können, unterstützen die Option OWNER.

Weitere Informationen zu den Befehlen BIND und PRECOMPILE finden Sie im Handbuch *Command Reference*.

Ermöglichen indirekter Zugriffsberechtigungen durch ein Paket

Der Zugriff auf Daten in einer Datenbank kann durch Anwendungsprogramme sowie durch Personen, die an einer interaktiven Workstation-Sitzung teilnehmen, angefordert werden. Ein Paket enthält Anweisungen, die es Benutzern ermöglichen, eine Vielzahl von Aktionen für viele Datenbankobjekte auszuführen. Für jede dieser Aktionen ist mindestens ein Zugriffsrecht erforderlich.

Zugriffsrechte, die Einzelbenutzern, die das Paket binden, und der Gruppe PUBLIC erteilt werden, werden beim Binden von statischem SQL zur Berechtigungsprüfung verwendet. Zugriffsrechte, die über Gruppen erteilt wurden, werden beim Binden von statischem SQL **nicht** zur Berechtigungsprüfung verwendet. Dem Benutzer mit einer gültigen *authID*, der ein Paket bindet, müssen entweder alle zum Ausführen der statischen SQL-Anweisungen im Paket erforderlichen Zugriffsrechte explizit oder implizit die erforderlichen Zugriffsrechte über PUBLIC erteilt worden sein, sofern beim Binden des Pakets nicht VALIDATE RUN angegeben wurde. Wurde während der Ausführung von BIND die Option VALIDATE RUN angegeben, schlägt BIND beim Auftreten von Berechtigungsfehlern für statische SQL-Anweisungen innerhalb des Pakets nicht fehl. Die entsprechenden SQL-Anweisungen werden dann während der Laufzeit nochmals geprüft. Zugriffsrechte über PUBLIC sowie Gruppen- und Einzelbenutzerzugriffsrechte werden **alle** bei der Überprüfung verwendet, um sicherzustellen, dass der Benutzer die nötige Berechtigung (Zugriffsrecht BIND oder BINDADD) zum Binden des Pakets besitzt.

Pakete können sowohl statisches als auch dynamisches SQL enthalten. Zum Verarbeiten eines Pakets mit statischem SQL muss ein Benutzer lediglich über das Zugriffsrecht EXECUTE für das Paket verfügen. Der Benutzer kann anschließend indirekt die Zugriffsrechte des Paketbinders für statisches SQL im Paket erhalten; dies ist allerdings nur im Rahmen der durch das Paket festgelegten Einschränkungen möglich.

Zum Verarbeiten eines Pakets mit dynamischen SQL-Anweisungen muss der Benutzer über das Zugriffsrecht EXECUTE für das Paket verfügen. Der Benutzer benötigt das Zugriffsrecht EXECUTE für das Paket und alle Zugriffsrechte, die zur Ausführung der dynamischen SQL-Anweisungen im Paket erforderlich sind. Die Berechtigungen und Zugriffsrechte des Binders werden für jedes statische SQL in dem Paket verwendet.

Ermöglichen indirekter Zugriffsberechtigungen durch ein Paket mit Kurznamen

Wenn ein Paket Verweise auf Kurznamen enthält, ist die Berechtigungsverarbeitung für Ersteller und Benutzer von Paketen etwas komplexer. Wenn ein Paketersteller Pakete, die Kurznamen enthalten, erfolgreich bindet, muss der Paketersteller keine Authentifizierung oder Überprüfung von Zugriffsrechten für Tabellen und Sichten durchlaufen, auf die an der Datenquelle mit

den Kurznamen verwiesen wird. Wer jedoch das Paket ausführt, muss eine Authentifizierung und eine Berechtigungsprüfung an den Datenquellen durchlaufen.

Nehmen Sie z. B. an, dass eine .SQL-Datei eines Paketerstellers mehrere SQL-Anweisungen enthält. Eine statische Anweisung verweist auf eine lokale Tabelle. Eine weitere dynamische Anweisung verweist auf einen Kurznamen. Wenn das Paket gebunden wird, wird die Berechtigungs-ID des Paketerstellers verwendet, um Zugriffsrechte für die lokale Tabelle zu überprüfen. Allerdings findet für die Datenquellenobjekte, die durch den Kurznamen gekennzeichnet sind, keine Überprüfung statt. Wenn ein anderer Benutzer das Paket ausführt und das Zugriffsrecht EXECUTE für dieses Paket gegeben ist, muss dieser Benutzer für die Anweisung, die auf die Tabelle verweist, keinerlei weitere Überprüfung seiner Zugriffsrechte durchlaufen. Allerdings muss der Benutzer, der das Paket ausführt, für die Anweisung mit dem Kurznamen eine Authentifizierung und eine Überprüfung der Zugriffsrechte an der Datenquelle durchlaufen.

Wenn die .SQL-Datei nur dynamische SQL-Anweisungen und eine Kombination von Tabellennamen und Kurznamen enthält, ist die DB2-Berechtigungsprüfung für lokale Objekte und Kurznamen gleich. Paketbenutzer müssen die Überprüfung der Zugriffsrechte für beliebige lokale Objekte (Tabellen, Sichten), die sich in der Anweisung befinden, durchlaufen. Außerdem werden die Zugriffsrechte für Objekte mit Kurznamen überprüft. (Paketbenutzer müssen eine Authentifizierung und eine Überprüfung der Zugriffsrechte an der Datenquelle durchlaufen, die die Objekte mit den Kurznamen enthält.) In beiden Fällen müssen die Benutzer des Pakets über das Zugriffsrecht EXECUTE verfügen.

Die ID und das Kennwort des Benutzers, der das Paket ausführt, werden für die gesamte Verarbeitung der Authentifizierungen und Zugriffsrechte verwendet. Diese Daten können geändert werden, indem eine Benutzerzuordnung erstellt wird.

Anmerkung: Beim statischen SQL können Kurznamen nicht angegeben werden. Verwenden Sie mit Paketen, die Kurznamen enthalten, nicht den Parameter DYNAMICRULES (auf BIND gesetzt).

Für Pakete mit Kurznamen sind möglicherweise weitere Berechtigungsschritte erforderlich, da DB2 dynamisches SQL verwendet, um mit Datenquellen der DB2-Produktfamilie Daten auszutauschen. Die Berechtigungs-ID, die das Paket an der Datenquelle ausführt, muss mit der entsprechenden Berechtigung ausgestattet sein, um das Paket an dieser Datenquelle dynamisch ausführen zu können. Weitere Informationen dazu, wie DB2 statisches und dynamisches SQL verarbeitet, finden Sie im Handbuch *SQL Reference*.

Steuern des Zugriffs auf Daten mit Sichten

Eine Sicht bietet eine Möglichkeit, den Zugriff auf eine Tabelle zu beschränken bzw. die Zugriffsrechte für eine Tabelle zu erweitern. Dies geschieht wie folgt:

- Es wird der Zugriff nur auf bestimmte Spalten der Tabelle ermöglicht.
Für Benutzer und Anwendungsprogramme, die nur einen Zugriff auf bestimmte Spalten einer Tabelle benötigen, kann ein berechtigter Benutzer eine Sicht erstellen, um den Zugriff auf die benötigten Spalten zu beschränken.
- Es wird der Zugriff nur auf eine Teilmenge der Tabellenzeilen ermöglicht.
Durch Angabe einer Klausel WHERE in der Unterabfrage einer Sichtdefinition kann ein berechtigter Benutzer die Zeilen, auf die über eine Sicht zugegriffen wird, beschränken.
- Es wird der Zugriff nur auf eine Teilmenge der Tabellenzeilen oder -spalten an der Datenquelle ermöglicht. Wenn Sie mit Kurznamen auf Datenquellen zugreifen, können Sie lokale DB2-Sichten erstellen, die auf Kurznamen verweisen. Diese Sichten können auf Kurznamen aus einer oder mehreren Datenquellen verweisen.

Anmerkung: Da Sie eine Sicht erstellen können, die Verweise auf Kurznamen aus mehreren Datenquellen enthalten kann, können die Benutzer von einer Sicht aus auf Daten mehrerer Datenquellen zugreifen. Diese Sichten sind *Sichten mehrerer Speicherpositionen*. Solche Sichten sind nützlich, wenn Sie Daten von Spalten mit sensiblen Tabellendaten in einer verteilten Umgebung verknüpfen oder wenn einzelne Benutzer für spezifische Objekte nicht über die erforderlichen Zugriffsrechte auf Datenquellen verfügen.

Zum Erstellen einer Sicht muss ein Benutzer über die Berechtigung SYSADM, die Berechtigung DBADM oder das Zugriffsrecht CONTROL oder SELECT für jede Tabelle oder Sicht, auf die in der Sichtdefinition verwiesen wird, verfügen. Der Benutzer muss darüber hinaus in der Lage sein, ein Objekt in dem für die Sicht angegebenen Schema zu erstellen. Das heißt, er muss über das Zugriffsrecht CREATEIN für ein vorhandenes Schema oder die Berechtigung IMPLICIT_SCHEMA für die Datenbank haben, wenn das Schema noch nicht existiert. Weitere Informationen finden Sie in „Erstellen einer Sicht“ auf Seite 166.

Wenn Sie Sichten mit Verweisen auf Kurznamen erstellen, müssen Sie über keine weitere Berechtigung auf die Datenquellen (Tabellen oder Sichten) verfügen, auf die mit den Kurznamen in der Sicht verwiesen wird. Allerdings müssen die Benutzer die Berechtigung SELECT oder eine gleichwertige Berechtigungsstufe für die zugrundeliegenden Datenquellenobjekte haben, wenn sie auf die Sicht zugreifen.

Wenn die Benutzer an der Datenquelle nicht die entsprechende Berechtigung für die zugrundeliegenden Objekte (Tabellen und Sichten) haben, können Sie wie folgt vorgehen:

1. Erstellen Sie eine Sicht auf die Datenquelle für diejenigen Spalten in der Tabelle der Datenquelle, auf die der Benutzer zureifen darf.
2. Erteilen Sie den Benutzern das Zugriffsrecht SELECT auf diese Sicht.
3. Erstellen Sie einen Kurznamen, der auf diese Sicht verweist.

Anschließend können die Benutzer auf die Spalten zugreifen, indem Sie eine SELECT-Anweisung absetzen, die den neuen Kurznamen als Verweis enthält.

Im folgenden Szenario erhalten Sie ein genaueres Beispiel, wie Sichten zur Einschränkung des Datenzugriffs verwendet werden können.

Viele Benutzer benötigen vielleicht aus unterschiedlichen Gründen Zugriff auf Informationen der Tabelle STAFF. Beispiel:

- Die Personalabteilung muss in der Lage sein, die gesamte Tabelle anzuzeigen und zu aktualisieren.
Diese Anforderung kann leicht dadurch erfüllt werden, dass der Gruppe PERSONNL die Zugriffsrechte SELECT und UPDATE für die Tabelle STAFF erteilt werden:

```
GRANT SELECT,UPDATE ON TABLE STAFF TO GROUP PERSONNL
```

- Die einzelnen Abteilungsleiter müssen in der Lage sein, Gehaltsdaten für ihre Mitarbeiter anzuzeigen.

Diese Anforderung kann erfüllt werden, indem eine Sicht für jeden Abteilungsleiter erstellt wird. Zum Beispiel kann folgende Sicht für den Leiter der Abteilung Nr. 51 erstellt werden:

```
CREATE VIEW EMP051 AS
  SELECT NAME,SALARY,JOB FROM STAFF
  WHERE DEPT=51
GRANT SELECT ON TABLE EMP051 TO JANE
```

Der Abteilungsleiter mit dem Berechtigungsnamen JANE würde die Sicht EMP051 genauso wie die Tabelle STAFF abfragen. Wenn er auf die Sicht EMP051 der Tabelle STAFF zugreift, werden dem Abteilungsleiter folgende Informationen angezeigt:

NAME	SALARY	JOB
Fraye	45150.0	Mgr
Williams	37156.5	Sales
Smith	35654.5	Sales
Lundquist	26369.8	Clerk
Wheeler	22460.0	Clerk

- Alle Benutzer müssen in der Lage sein, andere Mitarbeiter zu finden. Diese Anforderung kann erfüllt werden, indem eine Sicht auf die Spalte NAME der Tabelle STAFF und auf die Spalte LOCATION der Tabelle ORG erstellt wird und die beiden Tabellen über ihre jeweiligen Spalten DEPT und DEPTNUMB verknüpft werden:

```
CREATE VIEW EMPLOCS AS
  SELECT NAME, LOCATION FROM STAFF, ORG
  WHERE STAFF.DEPT=ORG.DEPTNUMB
GRANT SELECT ON TABLE EMPLOCS TO PUBLIC
```

Wenn Benutzer auf die Sicht EMPLOCS über die Standorte der Mitarbeiter zugreifen, werden folgende Informationen angezeigt:

NAME	LOCATION
Molinare	New York
Lu	New York
Daniels	New York
Jones	New York
Hanes	Boston
Rothman	Boston
Ngan	Boston
Kermisch	Boston
Sanders	Washington
Pernal	Washington
James	Washington
Sneider	Washington
Marenghi	Atlanta
O'Brien	Atlanta
Quigley	Atlanta
Naughton	Atlanta
Abrahams	Atlanta
Koonitz	Chicago
Plotz	Chicago
Yamaguchi	Chicago
Scoutten	Chicago
Fraye	Dallas
Williams	Dallas
Smith	Dallas

NAME	LOCATION
Lundquist	Dallas
Wheeler	Dallas
Lea	San Francisco
Wilson	San Francisco
Graham	San Francisco
Gonzales	San Francisco
Burke	San Francisco
Quill	Denver
Davis	Denver
Edwards	Denver
Gafney	Denver

Überwachen des Zugriffs auf Daten mit Hilfe der Audit-Einrichtung

Die DB2-Prüffunktion generiert ein Prüfprotokoll für eine Reihe vordefinierter Datenbankereignisse und ermöglicht Ihnen die Verwaltung des Protokolls. Auch wenn es keine Einrichtung zur Verweigerung des Datenzugriffs ist, kann die Audit-Einrichtung Versuche, auf Datenobjekte zuzugreifen bzw. diese zu ändern, überwachen und einen Protokollsatz darüber anlegen.

Zur Verwendung der Audit-Einrichtung `db2audit` ist die Berechtigung `SYS-ADM` erforderlich.

In „Kapitel 6. Protokollieren von DB2-Aktivitäten“ auf Seite 323 finden Sie eine detaillierte Beschreibung der DB2-Prüffunktion.

Datenverschlüsselung

Ein Teil Ihrer Sicherheitsplanung umfasst möglicherweise die Verschlüsselung von Daten. Sie können zu diesem Zweck die folgenden integrierten Funktionen für Verschlüsselung und Entschlüsselung verwenden: `ENCRYPT`, `DECRYPT_BIN`, `DECRYPT_CHAR` und `GETHINT`. Weitere Informationen zu diesen Funktionen einschließlich deren Syntax finden Sie im Handbuch *SQL Reference*.

Mit der Funktion `ENCRYPT` werden Daten mit Hilfe einer Methode auf Kennwortbasis verschlüsselt. Diese Funktionen ermöglichen ferner das Einbinden eines Kennworthinweises. Der Kennworthinweis ist in die verschlüsselten Daten eingebunden. Nach der Verschlüsselung können die Daten nur unter Verwendung des richtigen Kennworts wieder entschlüsselt werden. Anwendungsentwickler, die diese Funktionen verwenden möchten, sollten die Verwaltung vergessener Kennwörter und unbenutzbarer Daten in Ihre Planung mit einbeziehen.

Das Ergebnis der Verschlüsselungsfunktion (ENCRYPT) weist denselben Datentyp wie das erste Argument auf.

Es können nur variable Zeichen (VARCHAR) verschlüsselt werden.

Die vereinbarte Länge des Ergebnisses muss eine der folgenden Längen sein:

- Die Länge des Datenarguments plus 42, wenn der optionale Hinweisparameter angegeben wurde.
- Die Länge des Datenarguments plus 10, wenn der optionale Hinweisparameter nicht angegeben wurde.

Mit den Funktionen DECRYPT_BIN und DECRYPT_CHAR werden Daten mit der Entschlüsselung auf Kennwortbasis entschlüsselt.

Das Ergebnis der Funktionen DECRYPT_BIN und DECRYPT_CHAR weist denselben Datentyp wie das erste Argument auf.

Die vereinbarte Länge des Ergebnisses entspricht der Länge der Originaldaten.

Mit der Funktion GETHINT wird ein eingebundener Kennwothinweis wieder zurückgegeben. Ein Kennwothinweis ist ein Ausdruck, mit dessen Hilfe die Eigner von Daten sich wieder an die Kennwörter erinnern können. Das Wort „Ozean“ kann zum Beispiel als Hinweis für das Kennwort "Pazifik" verwendet werden.

Das Kennwort, das zum Verschlüsseln der Daten verwendet wird, wird mit einer der folgenden Methoden ermittelt:

- Kennwortargument. Das Kennwort ist eine Zeichenfolge, die beim Aufruf der Funktion ENCRYPT explizit übergeben wird. Die Daten werden mit dem angegebenen Kennwort verschlüsselt und entschlüsselt.
- Sonderregisterkennwort. Die Anweisung SET ENCRYPTION PASSWORD verschlüsselt den Kennwortwert und sendet das verschlüsselte Kennwort an den Datenbankmanager, der das Kennwort in einem Sonderregister speichert. Werden die Funktionen ENCRYPT, DECRYPT_BIN und DECRYPT_CHAR ohne Kennwortparameter aufgerufen, wird der Wert im Sonderregister ENCRYPTION PASSWORD verwendet.

Der Anfangs- oder Standardwert für das Sonderregister ist eine leere Zeichenfolge.

Gültige Werte für Kennwörter liegen zwischen 6 und 127 einschließlich dieser Werte selbst. Gültige Längen für Hinweise liegen zwischen 0 und 32 einschließlich dieser Werte selbst.

Wird das Sonderregister ENCRYPTION PASSWORD vom Client festgelegt, wird das Kennwort beim Client verschlüsselt, an den Datenbankserver gesen-

det und anschließend entschlüsselt. Um sicherzustellen, dass das Kennwort nicht lesbar bleibt, wird es auf dem Datenbankserver ebenfalls wieder verschlüsselt. Die Funktionen DECRYPT_BIN und DECRYPT_CHAR müssen das Sonderregister entschlüsseln, bevor sie es verwenden können. Der im Sonderregister ENCRYPTION PASSWORD gefundene Wert wird ebenfalls nicht in lesbarer Form belassen. Gateway-Sicherheit wird nicht unterstützt.

Aufgaben und erforderliche Berechtigungen

In den verschiedenen Unternehmen und Behörden sind die einzelnen Aufgabengebiete nicht immer in gleicher Weise verteilt. In Tabelle 5 werden einige allgemein verwendete Jobbezeichnungen, die Aufgaben, die diesen normalerweise zufallen, und die Berechtigungen bzw. Zugriffsrechte, die zum Ausführen dieser Aufgaben erforderlich sind, aufgeführt.

Tabelle 5. Allgemeine Jobbezeichnungen, Aufgaben und erforderliche Berechtigungen

JOBBEZEICHNUNG	AUFGABEN	ERFORDERLICHE BERECHTIGUNG
Abteilungsadministrator	Überwacht das Abteilungssystem; erstellt Datenbanken.	Berechtigung SYSCTRL. Berechtigung SYSADM, wenn die Abteilung über ein eigenes Exemplar verfügt.
Sicherheitsadministrator	Erteilt anderen Benutzern einige oder alle Berechtigungen oder Zugriffsrechte.	Berechtigung SYSADM oder DBADM
Datenbankadministrator	Entwirft, entwickelt, betreibt und pflegt eine oder mehrere Datenbanken und sorgt für die Sicherheit dieser Datenbank(en).	Berechtigungen DBADM und SYSMAINT für eine oder mehrere Datenbanken. In einigen Fällen Berechtigung SYSCTRL.
Systembediener	Überwacht die Datenbank und führt Sicherungen aus.	Berechtigung SYSMAINT
Anwendungsprogrammierer	Entwickelt und testet die Anwendungsprogramme für den Datenbankmanager; kann auch Tabellen mit Testdaten erstellen.	BINDADD, BIND für ein vorhandenes Paket, CONNECT und CREATETAB für eine oder mehrere Datenbanken, einige spezielle Zugriffsrechte für Schemata und eine Liste von Zugriffsrechten für einige Tabellen
Benutzeranalytiker	Definiert die Datenerfordernisse für ein Anwendungsprogramm durch Überprüfen der Systemkatalogsichten.	SELECT für die Katalogsichten; CONNECT für eine oder mehrere Datenbanken

Tabelle 5. Allgemeine Jobbezeichnungen, Aufgaben und erforderliche Berechtigungen (Forts.)

JOBBEZEICHNUNG	AUFGABEN	ERFORDERLICHE BERECHTIGUNG
Programmendbenutzer	Führt ein Anwendungsprogramm aus.	EXECUTE für das Paket; CONNECT für eine oder mehrere Datenbanken. Siehe Anmerkung im Anschluss an diese Tabelle.
Berater im Informationszentrum	Definiert die Datenerfordernisse für einen Abfragebenutzer; stellt die Daten zur Verfügung, indem er Tabellen und Sichten erstellt und den Zugriff auf Datenbankobjekte erteilt.	Berechtigung DBADM für eine oder mehrere Datenbanken
Abfragebenutzer	Führt SQL-Anweisungen zum Abrufen, Hinzufügen, Löschen oder Ändern von Daten aus; kann Ergebnisse in Form von Tabellen speichern.	CONNECT für eine oder mehrere Datenbanken; CREATEIN für das Schema der Tabellen und Sichten, die erstellt werden; und SELECT, INSERT, UPDATE, DELETE für einige Tabellen und Sichten

Anmerkung: Wenn ein Anwendungsprogramm dynamische SQL-Anweisungen enthält, muss der Programmendbenutzer möglicherweise neben EXECUTE und CONNECT über weitere Zugriffsrechte (wie SELECT, INSERT, DELETE und UPDATE) verfügen.

Verwenden des Systemkatalogs

Informationen über alle Datenbanken werden automatisch in einer Gruppe von Tabellen, dem sog. Systemkatalog, gepflegt, die beim Generieren der Datenbank erstellt werden. Dieser Systemkatalog beschreibt Tabellen, Spalten, Indizes, Programme, Zugriffsrechte und andere Objekte.

In sechs dieser Sichten werden die Zugriffsrechte von Benutzern und die Identität des Benutzers, der das Zugriffsrecht jeweils erteilt, aufgelistet:

SYSCAT.DBAUTH	Listet die Zugriffsrechte für Datenbanken auf.
SYSCAT.TABAUTH	Listet die Zugriffsrechte für Tabellen und Sichten auf.
SYSCAT.COLAUTH	Listet die Zugriffsrechte für Spalten auf.
SYSCAT.PACKAGEAUTH	Listet die Zugriffsrechte für Pakete auf.
SYSCAT.INDEXAUTH	Listet die Zugriffsrechte für Indizes auf.

SYSCAT.SCHEMAAUTH Listet die Zugriffsrechte für Schemata auf.

SYSCAT.PASSTHROUGHAUTH Listet das Zugriffsrecht für Server auf.

Zugriffsrechte, die den Benutzern vom System verliehen wurden, weisen in der Spalte GRANTOR den Wert SYSIBM auf. SYSADM, SYSMANT und SYSCAT werden nicht im Systemkatalog aufgelistet.

Durch die Anweisungen CREATE und GRANT werden Zugriffsrechte in den Systemkatalog eingetragen. Benutzer mit den Berechtigungen SYSADM und DBADM können Zugriffsrechte SELECT für die Systemkatalogsichten erteilen und widerrufen. Die folgenden Beispiele zeigen, wie durch Verwendung von SQL-Abfragen Informationen über Zugriffsrechte abgerufen werden können:

- „Abrufen von Berechtigungsnamen mit erteilten Zugriffsrechten“
- „Abrufen aller Namen mit der Berechtigung DBADM“ auf Seite 319
- „Abrufen der Namen mit Berechtigung zum Zugriff auf eine Tabelle“ auf Seite 319
- „Abrufen aller für Benutzer erteilten Zugriffsrechte“ auf Seite 320
- „Sicherheit der Systemkatalogsichten“ auf Seite 320.

Abrufen von Berechtigungsnamen mit erteilten Zugriffsrechten

Keine einzelne Systemkatalogsicht enthält Informationen über alle Zugriffsrechte. Die folgende Anweisung ruft alle Berechtigungsnamen mit Zugriffsrechten ab:

```
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'DATABASE' FROM SYSCAT.DBAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'TABLE ' FROM SYSCAT.TABAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'PACKAGE ' FROM SYSCAT.PACKAGEAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'INDEX ' FROM SYSCAT.INDEXAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'COLUMN ' FROM SYSCAT.COLAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'SCHEMA ' FROM SYSCAT.SCHEMAAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'SERVER ' FROM SYSCAT.PASSTHROUGHAUTH
ORDER BY GRANTEE, GRANTEETYPE, 3
```

Die durch diese Anweisung abgerufene Liste sollte in regelmäßigen Abständen mit Listen von Benutzer- und Gruppennamen verglichen werden, die in der Sicherheitseinrichtung des Systems definiert sind. Auf diese Weise können die Berechtigungsnamen ermittelt werden, die nicht mehr gültig sind.

Anmerkung: Wenn Sie ferne Datenbank-Clients unterstützen, ist der Berechtigungsname möglicherweise nur auf dem fernen Client und nicht auf Ihrem Datenbankserver-System definiert.

Abrufen aller Namen mit der Berechtigung DBADM

Die folgende Anweisung ruft alle Berechtigungsnamen ab, denen die Berechtigung DBADM direkt erteilt wurde:

```
SELECT DISTINCT GRANTEE FROM SYSCAT.DBAUTH
WHERE DBADMAUTH = 'Y'
```

Abrufen der Namen mit Berechtigung zum Zugriff auf eine Tabelle

Die folgende Anweisung ruft alle Berechtigungsnamen ab, die direkt zum Zugriff auf die Tabelle EMPLOYEE mit dem Qualifikationsmerkmal JAMES berechtigt sind:

```
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.TABAUTH
WHERE TABNAME = 'EMPLOYEE'
AND TABSCHEMA = 'JAMES'
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.COLAUTH
WHERE TABNAME = 'EMPLOYEE'
AND TABSCHEMA = 'JAMES'
```

Sie können herausfinden, wer die Tabelle EMPLOYEE mit dem Qualifikationsmerkmal JAMES aktualisieren kann, indem Sie die folgende Anweisung ausführen:

```
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.TABAUTH
WHERE TABNAME = 'EMPLOYEE' AND TABSCHEMA = 'JAMES' AND
(CONTROLAUTH = 'Y' OR
UPDATEAUTH = 'Y' OR UPDATEAUTH = 'G')
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.DBAUTH
WHERE DBADMAUTH = 'Y'
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.COLAUTH
WHERE TABNAME = 'EMPLOYEE' AND TABSCHEMA = 'JAMES' AND
PRIVTYPE = 'U'
```

Hierdurch werden alle Berechtigungsnamen mit der Berechtigung DBADM sowie die Namen, denen die Zugriffsrechte CONTROL oder UPDATE direkt erteilt wurden, abgerufen. Allerdings werden die Berechtigungsnamen von Benutzern, die nur über die Berechtigung SYSADM verfügen, nicht zurückgegeben.

Es ist zu beachten, dass einige Berechtigungsnamen Gruppen sein können, nicht nur Einzelbenutzer.

Abrufen aller für Benutzer erteilten Zugriffsrechte

Durch Abfragen der Systemkatalogsichten können Benutzer eine Liste der Zugriffsrechte, über die sie verfügen, und eine Liste der Zugriffsrechte, die sie anderen Benutzern erteilt haben, abrufen. Beispielsweise wird durch die folgende Anweisung eine Liste der Datenbankzugriffsrechte abgerufen, die direkt einem einzelnen Berechtigungsnamen erteilt wurden:

```
SELECT * FROM SYSCAT.DBAUTH
WHERE GRANTEE = USER AND GRANTEETYPE = 'U'
```

Mit der folgenden Anweisung wird eine Liste der Tabellenzugriffsrechte abgerufen, die direkt von einem bestimmten Benutzer erteilt wurden:

```
SELECT * FROM SYSCAT.TABAUTH
WHERE GRANTOR = USER
```

Mit der folgenden Anweisung wird eine Liste der einzelnen Spaltenzugriffsrechte abgerufen, die direkt von einem bestimmten Benutzer erteilt wurden:

```
SELECT * FROM SYSCAT.COLAUTH
WHERE GRANTOR = USER
```

Das Schlüsselwort USER in diesen Anweisungen ist immer gleich dem Wert des Berechtigungsnamens eines Benutzers. USER ist ein Sonderregister mit Lesezugriff. Weitere Informationen zu Sonderregistern finden Sie im Handbuch *SQL Reference*.

Sicherheit der Systemkatalogsichten

Während der Datenbankeinstellung wird das Zugriffsrecht SELECT für die Systemkatalogsichten der Gruppe PUBLIC erteilt. (Hinweise zu anderen Zugriffsrechten, die automatisch der Gruppe PUBLIC erteilt werden, finden Sie in „Zugriffsrechte für Datenbanken“ auf Seite 293.) In den meisten Fällen stellt dies kein Sicherheitsproblem dar. Bei sehr sensiblen Daten kann es jedoch zu Problemen kommen, da in diesen Tabellen jedes Objekt der Datenbank beschrieben wird. Wenn dies der Fall ist, kann das Zugriffsrecht SELECT für PUBLIC widerrufen werden. Anschließend kann das Zugriffsrecht SELECT nach Bedarf bestimmten Benutzern erteilt werden. Das Erteilen und Widerrufen des Zugriffsrechts SELECT für die Systemkatalogsichten erfolgt genauso wie für jede andere Sicht, aber Sie benötigen dafür die Berechtigung SYSADM oder DBADM.

Sie sollten zumindest die Beschränkung des Zugriffs auf die folgenden Katalogsichten erwägen:

- SYSCAT.DBAUTH
- SYSCAT.TABAUTH
- SYSCAT.PACKAGEAUTH
- SYSCAT.INDEXAUTH
- SYSCAT.COLAUTH
- SYSCAT.PASSTHROUGHAUTH
- SYSCAT.SCHEMAAUTH

Dadurch wird verhindert, dass Informationen über Zugriffsrechte von Benutzern für alle Benutzer mit Zugriff auf die Datenbank verfügbar werden. Mit Hilfe dieser Informationen könnte sich ein unbefugter Benutzer Zugriff auf die Datenbank verschaffen.

Sie sollten auch die Spalten überprüfen, für die die Statistikdaten gesammelt werden (siehe Abschnitt „Katalogstatistik“ im Handbuch *Systemverwaltung: Optimierung*). Einige der im Systemkatalog aufgezeichneten Statistikdaten könnten Datenwerte enthalten, die in Ihrer Umgebung sensible Informationen darstellen. Wenn diese Statistikdaten sensible Daten enthalten, können Sie das Zugriffsrecht SELECT auf die Katalogsichten SYSCAT.COLUMNS und SYSCAT.COLDIST für PUBLIC widerrufen.

Wenn Sie den Zugriff auf die Systemkatalogsichten beschränken möchten, können Sie Sichten definieren, damit jeder Berechtigungsname Informationen über seine eigenen Zugriffsrechte abrufen kann.

Zum Beispiel enthält die folgende Sicht MYSELECTS den Eigner und den Namen jeder Tabelle, für die dem Berechtigungsnamen eines Benutzers direkt das Zugriffsrecht SELECT erteilt wurde:

```
CREATE VIEW MYSELECTS AS
  SELECT TABSCHEMA, TABNAME FROM SYSCAT.TABAUTH
  WHERE GRANTEETYPE = 'U'
  AND GRANTEE = USER
  AND SELECTAUTH = 'Y'
```

Das Schlüsselwort USER in dieser Anweisung ist immer gleich dem Wert des Berechtigungsnamens. Mit der folgenden Anweisung wird die Sicht für jeden Berechtigungsnamen verfügbar gemacht:

```
GRANT SELECT ON TABLE MYSELECTS TO PUBLIC
```

Außerdem sollten Sie nicht vergessen, das Zugriffsrecht SELECT für die Basis-tabelle zu widerrufen:

```
REVOKE SELECT ON TABLE SYSCAT.TABAUTH FROM PUBLIC
```

Kapitel 6. Protokollieren von DB2-Aktivitäten

Mit Authentifizierung, Berechtigungen und Zugriffsrechten können bekannte oder erwartete Datenzugriffe gesteuert werden. Diese Maßnahmen reichen jedoch möglicherweise nicht aus, um unbekannte oder unerwartete Datenzugriffe zu verhindern. Zur Erkennung der letzteren Art von Datenzugriffen verfügt DB2 über eine Prüffunktion. Die erfolgreiche Überwachung unerwünschter Datenzugriffe mit anschließender Analyse dieser Vorgänge kann zu verbesserter Datenzugriffssteuerung und damit zur Verhinderung böswilliger oder unbedachter Datenzugriffe beitragen. Durch Überwachung von Anwendungs- und Benutzerzugriffen, einschließlich der Systemverwaltungsaktionen, kann ein fortlaufendes Protokoll aller an Ihrer Datenbank ausgeführten Aktivitäten erstellt werden.

Die DB2-Prüffunktion generiert ein Prüfprotokoll für eine Reihe vordefinierter Datenbankereignisse und ermöglicht Ihnen die Verwaltung des Protokolls. Die von dieser Funktion generierten Datensätze werden in einer Prüfprotokolldatei aufgezeichnet. Die Analyse dieser Datensätze macht Nutzungsmuster deutlich, an denen Systemmissbrauch erkennbar wird. Sobald der Systemmissbrauch identifiziert ist, können entsprechende Gegenmaßnahmen eingeleitet werden.

Die Prüffunktion arbeitet auf Exemplarebene, d. h. alle auf Exemplarebene und Datenbankebene stattfindenden Aktivitäten werden aufgezeichnet.

Beim Arbeiten in einer partitionierten Datenbankumgebung treten viele der zu protokollierenden Ereignisse in der Partition auf, mit der der Benutzer verbunden ist (Koordinator-knoten), oder auf dem Katalogknoten (sofern diese beiden Partitionen nicht identisch sind). Dies bedeutet, dass Prüfsätze von mehreren Partitionen generiert werden können. Ein Teil jedes Prüfsatzes enthält Informationen zu den Kennungen von Koordinator-knoten und Ursprungsknoten.

Das Prüfprotokoll (`db2audit.log`) und die Prüfkonfigurationsdatei (`db2audit.cfg`) befinden sich im Unterverzeichnis `security` des Exemplars. Zum Zeitpunkt der Erstellung eines Exemplars werden die Lese-/Schreibberechtigungen für diese Dateien (falls möglich) vom Betriebssystem festgelegt. Standardmäßig erhält nur der Exemplareigner Lese- und Schreibberechtigung. Es ist nicht empfehlenswert, diese Berechtigungen zu ändern.

Benutzer des Administrator-Tools der Prüffunktion (`db2audit`) müssen über die Berechtigung `SYSADM` verfügen.

Die Prüffunktion muss explizit gestoppt und gestartet werden. Beim Starten verwendet die Prüffunktion die vorhandenen Prüfkonfigurationsdaten. Da die Prüffunktion vom DB2-Server unabhängig ist, bleibt sie auch dann aktiv, wenn das Exemplar gestoppt wird. Beim Stoppen des Exemplars kann sogar ein Prüfsatz im Prüfprotokoll generiert werden.

Berechtigte Benutzer der Prüffunktion können folgende Aktionen der Prüffunktion steuern:

- Aufzeichnen von Prüfereignissen im DB2-Exemplar starten
- Aufzeichnen von Prüfereignissen im DB2-Exemplar stoppen
- Konfigurieren der Arbeitsweise der Prüffunktion, einschließlich der Auswahl von Kategorien der aufzuzeichnenden Prüfereignisse
- Anfordern einer Beschreibung der aktuellen Prüfkonfiguration
- Entfernen anstehender Prüfsätze aus dem Exemplar und Schreiben ins Prüfprotokoll
- Extrahieren von Prüfsätzen durch Formatieren und Kopieren aus dem Prüfprotokoll in eine unstrukturierte Textdatei oder in ASCII-Dateien. Das Extrahieren kann aus zwei Gründen erfolgen: Als Vorbereitung für die Analyse oder für das Abschneiden von Protokollsätzen.
- Entfernen von Prüfsätzen aus dem aktuellen Prüfprotokoll

Verschiedene Arten von Prüfsätzen können generiert werden. Beachten Sie in der Beschreibung der Kategorien möglicher Prüfereignisse (weiter unten), dass auf jeden Kategorienamen ein aus einem Wort bestehendes Schlüsselwort folgt, das die Art der Kategorie identifiziert. Folgende Kategorien von Prüfereignissen stehen zur Verfügung:

- Prüfung (AUDIT). Generiert Prüfsätze, wenn Prüfeinstellungen geändert werden oder auf das Prüfprotokoll zugegriffen wird.
- Berechtigungsprüfung (CHECKING). Generiert Prüfsätze während der Berechtigungsprüfung, wenn versucht wird, auf DB2-Objekte oder -Funktionen zuzugreifen oder sie zu manipulieren.
- Objektpflege (OBJMAINT). Generiert Prüfsätze beim Erstellen oder Löschen von Datenobjekten.
- Sicherheitspflege (SECMAINT). Generiert Prüfsätze beim Erteilen bzw. Widerrufen von Objekt- und Datenbankberechtigungen oder der Berechtigung DBADM. Prüfsätze werden auch generiert, wenn die Sicherheitskonfigurationsparameter SYSADM_GROUP, SYSCTRL_GROUP oder SYSMAINT_GROUP des Datenbankmanagers geändert werden.
- Systemverwaltung (SYSADMIN). Generiert Prüfsätze, wenn Operationen ausgeführt werden, für die die Berechtigung SYSADM, SYSMAINT oder SYSCTRL erforderlich ist.

- Benutzergültigkeitsprüfung (VALIDATE). Generiert Prüfsätze bei der Vergabe von Benutzerberechtigungen und beim Abrufen von System-sicherheitsdaten.
- Operationskontext (CONTEXT). Generiert Prüfsätze, die den Operationskontext darstellen, wenn eine Datenbankoperation ausgeführt wird. Diese Prüfkategorie erleichtert die Interpretation der Prüfprotokolldatei. Bei Verwendung zusammen mit dem Ereigniskorrelationsfeld des Protokolls kann eine Gruppe von Ereignissen auf eine einzige Datenbankoperation zurückgeführt werden. Beispiel: Eine dynamische SQL-Anweisung, eine Paketkennung für statisches SQL oder ein Indikator der ausgeführten Operationsart (z. B. CONNECT) kann den erforderlichen Kontext zum Analysieren der Prüfergebnisse liefern.

Anmerkung: Die SQL-Anweisung, die den Operationskontext liefert, kann sehr lang sein. Sie wird jedoch im CONTEXT-Prüfsatz vollständig angegeben. Dadurch kann der CONTEXT-Prüfsatz sehr umfangreich werden.

- Sie können fehlgeschlagene oder erfolgreich ausgeführte Operationen oder beide Arten prüfen.

Jede Datenbankoperation kann mehrere Prüfsätze generieren. Die tatsächliche Anzahl der generierten und in das Prüfprotokoll übertragenen Prüfsätze hängt von der Anzahl der aufzuzeichnenden Ereigniskategorien ab, die in der Konfiguration der Prüffunktion festgelegt ist. Sie ist außerdem davon abhängig, ob erfolgreiche oder fehlgeschlagene Operationen oder beide geprüft werden. Deshalb sollte gezielt ausgewählt werden, welche Ereignisse zu protokollieren sind.

Arbeitsweise der Prüffunktion

Die Prüffunktion zeichnet überprüfbare Ereignisse auf, einschließlich solcher, die sich auf Datenbankexemplare auswirken. Aus diesem Grund ist die Prüffunktion eine unabhängige DB2-Komponente, die auch ausgeführt werden kann, wenn das DB2-Exemplar gestoppt ist. Wenn die Prüffunktion aktiv ist, wird beim Starten eines gestoppten Exemplars die Protokollierung der Datenbankereignisse dieses Exemplars wieder aufgenommen.

Der zeitliche Ablauf beim Schreiben von Prüfsätzen in das Prüfprotokoll kann sich deutlich auf die Verarbeitungsleistung der Datenbanken in dem betreffenden Exemplar auswirken. Das Schreiben der Prüfsätze kann synchron oder asynchron zum Auftreten der Ereignisse erfolgen, die die Generierung dieser Prüfsätze auslösen. Der Wert des Konfigurationsparameters *audit_buf_sz* des Datenbankmanagers legt fest, wann das Schreiben der Prüfsätze stattfindet.

Wenn der Wert dieses Parameters Null (0) ist, wird der Schreibvorgang synchron ausgeführt. Das den Prüfsatz auslösende Ereignis bleibt im Wartestatus bis der Prüfsatz auf die Platte geschrieben ist. Die Wartezeit beim Schreiben der einzelnen Prüfsätze wirkt sich nachteilig auf die Verarbeitungsleistung von DB2 aus.

Wenn der Wert des Parameters *audit_buf_sz* größer als Null ist, wird der Schreibvorgang asynchron ausgeführt. Wenn der Parameterwert für *audit_buf_sz* größer als Null ist, entspricht er der Anzahl 4 KB-Seiten, die zum Erstellen eines internen Puffers verwendet wird. In dem internen Puffer werden Prüfsätze zwischengespeichert, bis sie gruppenweise auf die Platte geschrieben werden. Die Anweisung, die den aus einem Prüfereignis resultierenden Prüfsatz generiert, wartet nicht, bis der Prüfsatz auf die Platte geschrieben ist, sondern kann die Verarbeitung ohne Verzögerung fortsetzen.

Beim asynchronen Schreiben verbleiben die Prüfsätze gegebenenfalls für einige Zeit in einem nur teilweise gefüllten Puffer. Damit dies nicht über zu lange Zeiträume der Fall ist, erzwingt der Datenbankmanager das regelmäßige Schreiben der Prüfsätze auf die Platte. Ein berechtigter Benutzer der Prüffunktion kann den Prüfpuffer auch mit einer expliziten Anforderung leeren.

Je nachdem, ob der Schreibvorgang synchron oder asynchron erfolgt, ergeben sich Unterschiede beim Auftreten von Fehlern. Im asynchronen Modus gehen möglicherweise einige Datensätze verloren, da die Prüfsätze gepuffert werden, bevor sie auf die Platte geschrieben werden. Im synchronen Modus geht, wenn überhaupt, ein Datensatz verloren, da durch den Fehler höchstens ein Prüfsatz nicht geschrieben werden kann.

Die Einstellung des Prüffunktionsparameters *ERRORTYPE* legt fest, wie auftretende Fehler zwischen DB2 und der Prüffunktion behandelt werden. Wenn die Prüffunktion aktiv und der Prüffunktionsparameter *ERRORTYPE* auf *AUDIT* gesetzt ist, wird die Prüffunktion genau so behandelt wie jede andere DB2-Komponente. Ein Prüfsatz muss geschrieben werden (im synchronen Modus auf die Platte, im asynchronen Modus in den Prüfpuffer), damit ein Prüfereignis, das sich auf eine Anweisung bezieht, als erfolgreich eingestuft wird. Bei jedem in diesem Modus festgestellten Fehler wird für die Anweisung, die einen Prüfsatz ausgelöst hat, ein negativer *SQLCODE*-Wert an die Anwendung zurückgegeben. Ist der Parameter *ERRORTYPE* auf *NORMAL* gesetzt, werden alle von *db2audit* gemeldeten Fehler ignoriert und der *SQLCODE* der Operation wird zurückgegeben. Weitere Einzelheiten zum Parameter *ERRORTYPE* der Prüffunktion (und weiteren zugehörigen Parameter) finden Sie in „Anwendungsszenarios der Prüffunktion“ auf Seite 327.

Je nach der API- oder SQL-Anweisung und den Prüfeinstellungen des DB2-Exemplars können für ein bestimmtes Ereignis keine, eine oder mehrere Prüfsätze generiert werden. Beispiel: Eine SQL-Anweisung *UPDATE* mit einer

Unterabfrage SELECT kann zur Generierung eines Prüfsatzes mit den Ergebnissen der Berechtigungsprüfung für das Zugriffsrecht UPDATE einer Tabelle und zur Generierung eines weiteren Prüfsatzes mit den Ergebnissen der Berechtigungsprüfung für das Zugriffsrecht SELECT einer Tabelle führen.

Für dynamische DML-Anweisungen (DML = Data Manipulation Language) werden Prüfsätze für alle Berechtigungsprüfungen zum Zeitpunkt der Anweisungsvorbereitung (Prepare) generiert. Die erneute Verwendung dieser Anweisungen durch den selben Benutzer wird nicht erneut überprüft, weil zu diesem Zeitpunkt keine Berechtigungsprüfung stattfindet. Wenn jedoch eine der Katalogtabellen mit Berechtigungsinformationen geändert wurde, werden in der nächsten Arbeitseinheit die Berechtigungen für die zwischengespeicherten dynamischen SQL-Anweisungen erneut geprüft und ein oder mehrere neue Prüfsätze erstellt.

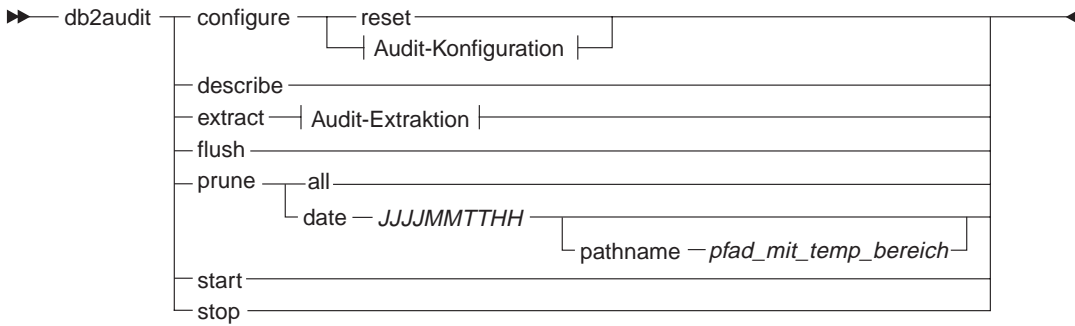
Für ein Paket, das ausschließlich statische DML-Anweisungen enthält, ist die Berechtigungsprüfung, mit der geprüft wird, ob ein Benutzer die erforderliche Berechtigung zum Ausführen des Pakets hat, das einzige prüfbare Ereignis, das einen Prüfsatz generieren könnte. Die Berechtigungsprüfung und die gegebenenfalls erforderliche Prüfsatzerstellung für die statischen SQL-Anweisungen des Pakets werden beim Vorkompilieren oder Binden des Pakets durchgeführt. Die Ausführung der statischen SQL-Anweisungen in dem Paket ist nicht prüfbar. Wird ein Paket entweder explizit vom Benutzer oder implizit vom System erneut gebunden, werden Prüfsätze für die von den statischen SQL-Anweisungen benötigten Berechtigungsprüfungen generiert.

Für Anweisungen, bei denen die Berechtigungsprüfung zum Zeitpunkt der Ausführung erfolgt (z. B. Data Definition Language (DDL) oder GRANT- und REVOKE-Anweisungen), werden bei jeder Verwendung dieser Anweisungen Prüfsätze generiert.

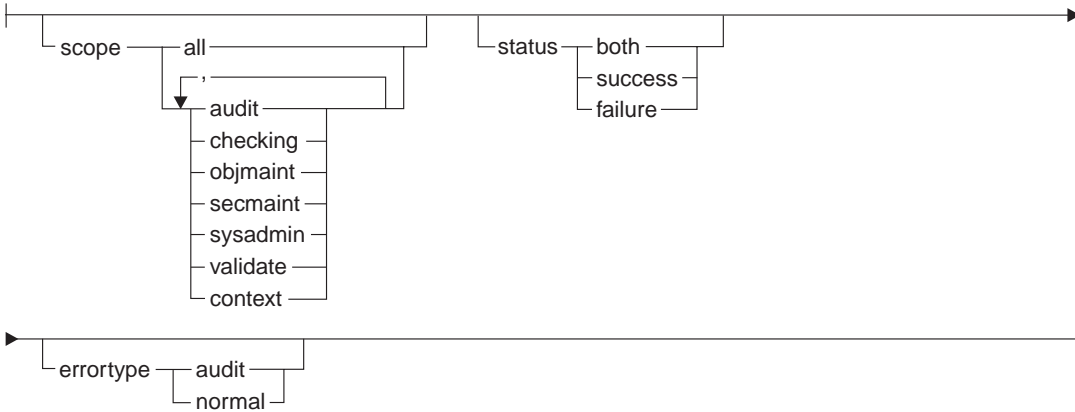
Anmerkung: Bei der Ausführung von DDL lautet die aufgezeichnete Abschnittsnummer für alle Ereignisse (ausgenommen Kontextereignisse) im Prüfsatz Null (0), und zwar unabhängig von der tatsächlichen Abschnittsnummer der Anweisung.

Anwendungsszenarios der Prüffunktion

Wenn Sie die einzelnen Abschnitte der folgenden Syntaxdiagramme genau lesen, erleichtert dies das Verständnis für die Einsatzmöglichkeiten der Prüffunktion.



Audit-Konfiguration:



Audit-Extraktion:

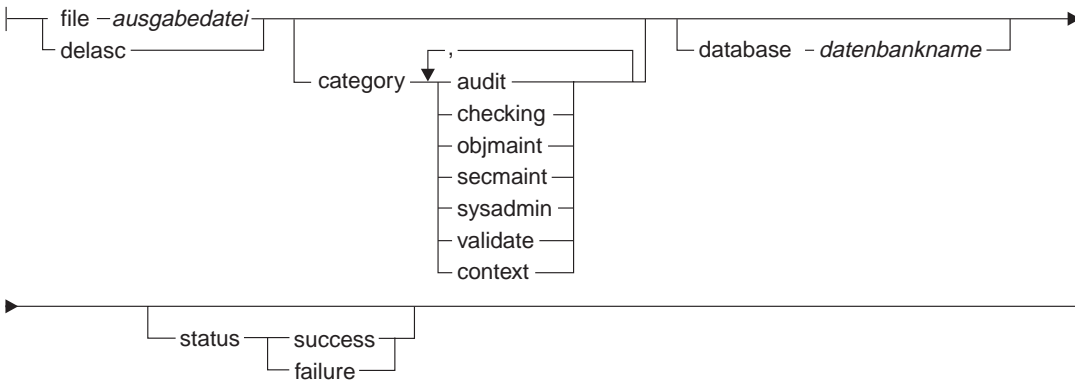


Abbildung 7. Syntax von DB2AUDIT

Es folgt eine Beschreibung der einzelnen Parameter mit impliziten Verwendungshinweisen:

configure

Dieser Parameter ermöglicht das Ändern der Konfigurationsdatei `db2audit.cfg` im Unterverzeichnis `security` des Exemplars. Diese Datei kann auch aktualisiert werden, wenn das Exemplar nicht gestartet ist. Aktualisierungen, die ausgeführt werden, während das Exemplar aktiv ist, wirken sich zugleich dynamisch auf DB2-Prüfvorgänge auf allen Partitionen aus. Die Konfigurationsaktion für die Konfigurationsdatei führt zur Erstellung eines Prüfsatzes, wenn die Prüffunktion gestartet ist und die Kategorie *audit* der prüfbaren Ereignisse geprüft wird.

Folgende Aktionen sind in der Konfigurationsdatei möglich:

- **RESET.** Durch diese Aktion wird die Anfangskonfiguration der Konfigurationsdatei wiederhergestellt (dabei beinhaltet SCOPE alle Kategorien außer CONTEXT, STATUS ist FAILURE, ERRORTYPE ist NORMAL und die Prüffunktion ist OFF). Diese Aktion erstellt eine neue Prüfkonfigurationsdatei, wenn die ursprüngliche Datei verloren gegangen oder beschädigt ist.
- **SCOPE.** Diese Aktion gibt an, welche Ereigniskategorie(n) geprüft werden soll(en). Mit dieser Aktion kann die Prüffunktion gezielt eingesetzt und das Anwachsen des Protokolls begrenzt werden. Es empfiehlt sich, die Anzahl und Art der protokollierten Ereignisse so weit wie möglich zu begrenzen, da sonst das Prüfprotokoll sehr schnell wächst.

Anmerkung: Bitte beachten Sie, dass SCOPE standardmäßig alle Kategorien außer CONTEXT umfasst und zu einer großen Anzahl generierter Prüfsätze führen kann. In Verbindung mit dem Modus (synchron oder asynchron) kann die Kategorieauswahl zu einem deutlichen Leistungsabfall und zu einem deutlich höheren Speicherplatzbedarf führen.

- **STATUS.** Diese Aktion gibt an, ob nur erfolgreiche (success) oder fehlgeschlagene (failure) Ereignisse oder beide (both) Arten protokolliert werden sollen.

Anmerkung: Kontextereignisse treten auf, bevor der Status einer Operation bekannt ist. Deshalb werden solche Ereignisse unabhängig vom Wert dieses Parameters protokolliert.

- **ERRORTYPE.** Diese Aktion gibt an, ob Prüffehler an den Benutzer zurückgegeben oder ignoriert werden. Mögliche Werte für diesen Parameter sind:

- AUDIT. Alle Fehler (auch Fehler, die in der Prüffunktion auftreten) werden von DB2 verwaltet und alle negativen SQLCODE-Werte werden an den Aufrufenden zurückgemeldet.
- NORMAL. Alle von db2audit generierten Fehler werden ignoriert und nur die SQLCODE-Werte der im Zusammenhang mit der gerade ausgeführten Operation auftretenden Fehler werden an die Anwendung zurückgegeben.

describe

Dieser Parameter zeigt die aktuellen Prüfkonfigurationsdaten und den aktuellen Prüfstatus in der Standardausgabe an.

extract Dieser Parameter ermöglicht das Verschieben von Prüfsätzen aus dem Prüfprotokoll an ein angegebenes Ziel. Wenn keine wahlfreien Klauseln angegeben sind, werden alle Prüfsätze extrahiert und in eine unstrukturierte Berichtsdatei geschrieben. Ist der Parameter „extract“ nicht angegeben, wird der Prüfsatz in eine Datei mit dem Namen *db2audit.out* geschrieben, die sich im Verzeichnis *security* befindet. Ist die *ausgabedatei* bereits vorhanden, wird eine Fehlernachricht zurückgegeben.

Folgende Optionen können beim Extrahieren verwendet werden:

- FILE. Die extrahierten Prüfsätze werden in eine Datei (*ausgabedatei*) geschrieben.
- DELASC. Die extrahierten Prüfsätze werden in begrenztem ASCII-Format dargestellt, das in relationale DB2-Tabellen geladen werden kann. Die Ausgabe wird in separate Dateien geschrieben (eine Datei für jede Kategorie). Dabei werden folgende Dateinamen verwendet:
 - audit.del
 - checking.del
 - objmaint.del
 - secmaint.del
 - sysadmin.del
 - validate.del
 - context.del

Mit der Option DELASC kann außerdem der Standardbegrenzer für die Prüfzeichenfolge („0xff“) beim Extrahieren aus dem Prüfprotokoll außer Kraft gesetzt werden. Geben Sie dazu DELASC DELIMITER, gefolgt von dem neuen Begrenzer ein, den Sie zur Vorbereitung für das Laden in eine Tabelle verwenden wollen, in die die Prüfsätze aufgenommen werden sollen. Der neue Ladebegrenzer kann ein Einzelzeichen (z. B. !) sein oder ein Vierbytezeichen, das eine Hexadezimalzahl darstellt (z. B. 0xff). Weitere Informationen finden Sie in „Tipps und Verfahrensweisen für die Prüffunktion“ auf Seite 352.

- **CATEGORY.** Die Prüfsätze für die angegebenen Prüfereigniskategorien sollen extrahiert werden. Wird diese Option nicht angegeben, können alle Kategorien extrahiert werden.
- **DATABASE.** Die Prüfsätze für eine angegebene Datenbank sollen extrahiert werden. Wird diese Option nicht angegeben, kann aus allen Datenbanken extrahiert werden.
- **STATUS.** Die Prüfsätze für den angegebenen Status sollen extrahiert werden. Wird diese Option nicht angegeben, können alle Prüfsätze extrahiert werden.

flush Dieser Parameter erzwingt, dass alle anstehenden Prüfsätze in das Prüfprotokoll geschrieben werden. Außerdem wird der Prüfstatus in der Steuerkomponente von „unable to log“ (Protokollieren nicht möglich) auf „ready to log“ (zum Protokollieren bereit) zurückgesetzt, wenn sich die Prüffunktion in einem Fehlerstatus befindet.

prune Dieser Parameter ermöglicht das Löschen von Prüfsätzen aus dem Prüfprotokoll. Wenn die Prüffunktion aktiv ist und die Ereigniskategorie „audit“ zum Prüfen angegeben ist, wird nach dem Löschen aus dem Prüfprotokoll ein Prüfsatz protokolliert.

Folgende Optionen können beim Löschen verwendet werden:

- **ALL.** Alle Prüfsätze im Prüfprotokoll sollen gelöscht werden.
- **DATE jjjmmmttss.** Der Benutzer kann festlegen, dass alle bis zu dem angegebenen Tag/Zeitpunkt generierten Prüfsätze aus dem Prüfprotokoll gelöscht werden sollen. Der Benutzer kann wahlfrei einen pfadnamen

der von der Prüffunktion beim Leeren des Protokolls als temporärer Speicherplatz verwendet wird. Dieser temporäre Speicherplatz ermöglicht das Leeren des Protokolls auch dann, wenn die Platte, auf der es sich befindet, voll ist bzw. nicht mehr genügend freier Speicherplatz für die Löschoperation verfügbar ist.

start Dieser Parameter veranlasst die Prüffunktion, mit dem Prüfen von Ereignissen (basierend auf dem Inhalt der Datei db2audit.cfg) zu beginnen. In einem partitionierten DB2-Exemplar beginnt der Prüfvorgang in allen Partitionen, wenn diese Klausel angegeben ist. Wenn die Ereigniskategorie „audit“ für den Prüfvorgang angegeben wurde, wird beim Starten der Prüffunktion ein Prüfsatz protokolliert.

stop Dieser Parameter veranlasst die Prüffunktion, das Prüfen von Ereignissen zu stoppen. In einem partitionierten DB2-Exemplar wird der Prüfvorgang in allen Partitionen gestoppt, wenn diese Klausel angegeben ist. Wenn die Ereigniskategorie „audit“ für den Prüfvorgang angegeben wurde, wird beim Stoppen der Prüffunktion ein Prüfsatz protokolliert.

Nachrichten der Prüffunktion

SQL1322N Beim Schreiben in die Prüfprotokolldatei ist ein Fehler aufgetreten.

Erläuterung: Beim Aufrufen der DB2-Prüffunktion zum Schreiben eines Prüfereignisses in die Prüfprotokolldatei wurde ein Fehler festgestellt. Das Dateisystem, in dem sich das Prüfprotokoll befindet, verfügt nicht über genügend freien Speicherbereich.

Benutzeraktion: Der Systemadministrator sollte in dem Dateisystem Speicherplatz frei machen oder durch Abschneiden die Größe des Prüfprotokolls reduzieren.

Sobald wieder Speicherplatz frei ist, löschen Sie mit db2audit die im Speicher befindlichen Daten und machen Sie die Prüffunktion wieder betriebsbereit. Stellen Sie sicher, dass die erforderlichen Extrahierungen erstellt wurden, oder dass vor dem Abschneiden eine Kopie des Protokolls erstellt wurde (gelöschte Prüfsätze können nicht wiederhergestellt werden).

sqlcode: -1322

sqlstate: 50830

SQL1323N Beim Zugriff auf die Prüfkonfigurationsdatei ist ein Fehler aufgetreten.

Erläuterung: Die Prüfkonfigurationsdatei (db2audit.cfg) konnte nicht geöffnet werden oder war ungültig. Dieser Fehler kann dadurch verursacht sein, dass die Datei db2audit.cfg nicht vorhanden ist oder beschädigt wurde.

Benutzeraktion: Führen Sie eine der folgenden Maßnahmen aus:

- Stellen Sie die Datei aus einer gespeicherten Kopie wieder her
- Setzen Sie die Konfigurationsdatei der Prüffunktion in die Grundstellung zurück, indem Sie den folgenden Befehl absetzen:
db2audit reset

sqlcode: -1323

sqlstate: 57019

Satzaufbau der Prüffunktion

Wenn ein Prüfsatz mit der Option DELASC aus dem Prüfprotokoll extrahiert wird, hat jeder Prüfsatz eines der in den folgenden Tabellen dargestellten Formate. Am Anfang jeder Tabelle ist der Inhalt eines Beispielprüfsatzes dargestellt. Die Einträge der Prüfsätze werden in der dazugehörigen Tabelle Zeile für Zeile beschrieben. Besonders wichtige Einträge sind **fett** hervorgehoben. Solche Einträge enthalten Informationen, die für Sie von besonderem Interesse sind.

Anmerkungen:

1. Nicht in allen Feldern der Beispielprüfsätze sind Werte angegeben.
2. Einige Felder, wie z. B. „Access Attempted“, sind als Bitmaps in begrenztem ASCII-Format gespeichert. In der hier gezeigten, unstrukturierten Berichtsdatei werden diese Felder jedoch durch Zeichenfolgen dargestellt, die ihre Bitmapwerte wiedergeben.

Tabelle 6. Prüfsatzaufbau für AUDIT-Ereignisse

timestamp=1998-06-24-11.54.05.151232;category=AUDIT;audit event=START; event correlator=0;event status=0; userid=boss;authid=BOSS;		
NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des AUDIT-Ereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Gültige Werte: AUDIT
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Gültige Werte: CONFIGURE, DB2AUD, EXTRACT, FLUSH, PRUNE, START, STOP und UPDATE_ADMIN_CFG
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Gibt an, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt: Erfolgreiches Ereignis > = 0 Fehlgeschlagenes Ereignis < 0
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses.

Tabelle 7. Prüfsatzaufbau für CHECKING-Ereignisse

timestamp=1998-06-24-08.42.11.622984;category=CHECKING;audit event=CHECKING_OBJECT; event correlator=2;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; object name=F00;object type=DATABASE; access approval reason=DATABASE;access attempted=CONNECT;		
NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des AUDIT-Ereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Gültige Werte: CHECKING
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Gültige Werte: CHECKING_OBJECT und CHECKING_FUNCTION

Tabelle 7. Prüfsatzaufbau für CHECKING-Ereignisse (Forts.)

<pre>timestamp=1998-06-24-08.42.11.622984;category=CHECKING;audit event=CHECKING_OBJECT; event correlator=2;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; object name=F00;object type=DATABASE; access approval reason=DATABASE;access attempted=CONNECT;</pre>		
NAME	FORMAT	BESCHREIBUNG
Event Correlator (Ereignis- korrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Gibt an, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt: Erfolgreiches Ereignis > = 0 Fehlgeschlagenes Ereignis < 0
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Exemplarebene handelt.
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses.
Origin Node Number (Nummer des Ursprungs- knotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinator- knotens)	SMALLINT	Die Knotennummer des Koordinator-knotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Application Name (Anwendungs- name)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Package Schema (Paketschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(128)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Section Number (Paketab- schnittsnummer)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.

Tabelle 7. Prüfsatzaufbau für CHECKING-Ereignisse (Forts.)

<pre>timestamp=1998-06-24-08.42.11.622984;category=CHECKING;audit event=CHECKING_OBJECT; event correlator=2;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; object name=F00;object type=DATABASE; access approval reason=DATABASE;access attempted=CONNECT;</pre>		
NAME	FORMAT	BESCHREIBUNG
Object Schema (Objektschema)	VARCHAR(128)	Das Schema des Objekts, für das das Prüfereignis generiert wurde.
Object Name (Objektname)	VARCHAR(128)	Der Name des Objekts, für das das Prüfereignis generiert wurde.
Object Type (Objektart)	VARCHAR(32)	Die Art des Objekts, für das das Prüfereignis generiert wurde. Gültige Werte: TABLE, VIEW, ALIAS, FUNCTION, INDEX, PACKAGE, DATA_TYPE, NODEGROUP, SCHEMA, STORED_PROCEDURE, BUFFERPOOL, TABLESPACE, EVENT_MONITOR, TRIGGER, DATABASE, INSTANCE, FOREIGN_KEY, PRIMARY_KEY, UNIQUE_CONSTRAINT, CHECK_CONSTRAINT, WRAPPER, SERVER, NICKNAME, USER MAPPING, SERVER OPTION, TRANSFORM, TYPE MAPPING, FUNCTION MAPPING, SUMMARY TABLES und NONE.
Access Approval Reason (Grund für Zugriffgewährung)	CHAR(18)	Gibt an, warum der Zugriff für dieses Prüfereignis gewährt wurde. Gültige Werte: Die in der ersten Liste im Anschluss an diese Tabelle angegebenen Werte.
Access Attempted (Zugriffsversuch)	CHAR(18)	Gibt an, welche Art von Zugriff versucht wurde. Gültige Werte: Die in der zweiten Liste im Anschluss an diese Tabelle angegebenen Werte.

In der folgenden Liste sind mögliche Gründe für die Zugriffsgewährung für das CHECKING-Ereignis aufgeführt:

0x0000000000000001 ACCESS DENIED

Der Zugriff wird nicht gewährt, sondern verweigert.

0x0000000000000002 SYSADM

Der Zugriff wird gewährt. Die Anwendung/der Benutzer verfügt über die Berechtigung SYSADM.

0x0000000000000004 SYSCTRL

Der Zugriff wird gewährt. Die Anwendung/der Benutzer verfügt über die Berechtigung SYSCTRL.

0x0000000000000008 SYSMANT

Der Zugriff wird gewährt. Die Anwendung/der Benutzer verfügt über die Berechtigung SYSMANT.

0x0000000000000010 DBADM

Der Zugriff wird gewährt. Die Anwendung/der Benutzer verfügt über die Berechtigung DBADM.

0x0000000000000020 DATABASE PRIVILEGE

Der Zugriff wird gewährt. Die Anwendung/der Benutzer verfügt über ein explizites Zugriffsrecht für die Datenbank.

0x0000000000000040 OBJECT PRIVILEGE

Der Zugriff wird gewährt. Die Anwendung/der Benutzer verfügt über ein explizites Zugriffsrecht für das Objekt oder die Funktion.

0x0000000000000080 DEFINER

Der Zugriff wird gewährt. Die Anwendung/der Benutzer hat das Objekt oder die Funktion definiert.

0x0000000000000100 OWNER

Der Zugriff wird gewährt. Die Anwendung/der Benutzer ist der Eigentümer des Objekts oder der Funktion.

0x0000000000000200 CONTROL

Der Zugriff wird gewährt. Die Anwendung/der Benutzer verfügt über das Zugriffsrecht CONTROL für das Objekt oder die Funktion.

0x0000000000000400 BIND

Der Zugriff wird gewährt. Die Anwendung/der Benutzer verfügt über die Bindeberechtigung für das Paket.

In der folgenden Liste sind mögliche Arten von Zugriffsversuchen auf CHECKING-Ereignisse aufgeführt:

0x0000000000000002 ALTER

Es wurde versucht ein Objekt zu ändern.

0x0000000000000004 DELETE

Es wurde versucht, ein Objekt zu löschen.

0x0000000000000008 INDEX

Es wurde versucht, einen Index zu verwenden.

0x0000000000000010 INSERT

Es wurde versucht, etwas in ein Objekt einzufügen.

0x0000000000000020 SELECT

Es wurde versucht, eine Tabelle oder Sicht abzufragen.

0x0000000000000040 UPDATE

Es wurde versucht, Daten in einem Objekt zu aktualisieren.

0x0000000000000080 REFERENCE

Es wurde versucht, referenzielle Integritätsbedingungen zwischen Objekten einzurichten.

0x0000000000000100 CREATE
Es wurde versucht, ein Objekt zu erstellen.

0x0000000000000200 DROP
Es wurde versucht, ein Objekt zu löschen.

0x0000000000000400 CREATEIN
Es wurde versucht, ein Objekt in einem anderen Schema zu erstellen.

0x0000000000000800 DROPIN
Es wurde versucht, ein in einem anderen Schema gefundenes Objekt zu löschen.

0x0000000000001000 ALTERIN
Es wurde versucht, ein in einem anderen Schema gefundenes Objekt zu ändern.

0x0000000000002000 EXECUTE
Es wurde versucht, eine Anwendung auszuführen.

0x0000000000004000 BIND
Es wurde versucht, eine Anwendung zu binden oder vorzubereiten.

0x0000000000008000 SET EVENT MONITOR
Es wurde versucht, Ereignismonitorschalter zu setzen.

0x0000000000010000 SET CONSTRAINTS
Es wurde versucht, Integritätsbedingungen für ein Objekt zu definieren.

0x0000000000020000 COMMENT ON
Es wurde versucht, Kommentare zu einem Objekt zu erstellen.

0x0000000000040000 GRANT
Es wurde versucht, einer anderen Benutzer-ID Zugriffsrechte für ein Objekt zu erteilen.

0x0000000000080000 REVOKE
Es wurde versucht, Zugriffsrechte einer Benutzer-ID für ein Objekt zu widerrufen.

0x0000000000100000 LOCK
Es wurde versucht, ein Objekt zu sperren.

0x0000000000200000 RENAME
Es wurde versucht, ein Objekt umzubenennen.

0x0000000000400000 CONNECT
Es wurde versucht, eine Verbindung zu einem Objekt herzustellen.

0x0000000000800000 Member of SYS Group
Es wurde versucht, auf ein Mitglied der SYS-Gruppe zuzugreifen oder ein Mitglied der SYS-Gruppe zu verwenden.

0x00000000100000 Access All

Es wurde versucht, eine Anweisung mit allen erforderlichen Zugriffsrechten für gespeicherte Objekte auszuführen (wird nur für DBADM/SYSADM verwendet).

0x00000000200000 Drop All

Es wurde versucht, mehrere Objekte zu löschen.

0x00000000400000 LOAD

Es wurde versucht, eine Tabelle in einem Tabellenbereich zu laden.

0x00000000800000 USE

Es wurde versucht, eine Tabelle in einem Tabellenbereich zu erstellen.

Tabelle 8. Prüfsatzaufbau für OBJMAINT-Ereignisse

<pre>timestamp=1998-06-24-08.42.41.957524;category=OBJMAINT;audit event=CREATE_OBJECT; event correlator=3;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; package schema=NULLID;package name=SQLC28A1; package section=0;object schema=BOSS;object name=AUDIT;object type=TABLE;</pre>		
NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des AUDIT-Ereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Gültige Werte: OBJMAINT
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Gültige Werte: CREATE_OBJECT, RENAME_OBJECT und DROP_OBJECT
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Gibt an, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt: Erfolgreiches Ereignis > = 0 Fehlgeschlagenes Ereignis < 0
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Exemplarebene handelt.
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses.

Tabelle 8. Prüfsatzaufbau für OBJMAINT-Ereignisse (Forts.)

<pre>timestamp=1998-06-24-08.42.41.957524;category=OBJMAINT;audit event=CREATE_OBJECT; event correlator=3;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; package schema=NULLID;package name=SQLC28A1; package section=0;object schema=BOSS;object name=AUDIT;object type=TABLE;</pre>		
NAME	FORMAT	BESCHREIBUNG
Origin Node Number (Nummer des Ursprungsknotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinator-knotens)	SMALLINT	Die Knotennummer des Koordinatorknotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Package Schema (Paketschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(128)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Section Number (Paketabschnittsnummer)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Object Schema (Objektschema)	VARCHAR(128)	Das Schema des Objekts, für das das Prüfereignis generiert wurde.
Object Name (Objektname)	VARCHAR(128)	Der Name des Objekts, für das das Prüfereignis generiert wurde.
Object Type (Objektart)	VARCHAR(32)	Die Art des Objekts, für das das Prüfereignis generiert wurde. Gültige Werte: TABLE, VIEW, ALIAS, FUNCTION, INDEX, PACKAGE, DATA_TYPE, NODEGROUP, SCHEMA, STORED_PROCEDURE, BUFFERPOOL, TABLESPACE, EVENT_MONITOR, TRIGGER, DATABASE, INSTANCE, FOREIGN_KEY, PRIMARY_KEY, UNIQUE_CONSTRAINT, CHECK_CONSTRAINT, WRAPPER, SERVER, NICKNAME, USER_MAPPING, SERVER_OPTION, TRANSFORM, TYPE_MAPPING, FUNCTION_MAPPING, SUMMARY_TABLES und NONE.

Tabelle 9. Prüfsatzaufbau für SECMAINT-Ereignisse

timestamp=1998-06-24-11.57.45.188101;category=SECMAINT;audit event=GRANT; event correlator=4;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.boss.980624155728;application name=db2bp; package schema=NULLID;package name=SQLC28A1; package section=0;object schema=BOSS;object name=T1;object type=TABLE; grantor=BOSS;grantee=WORKER;grantee type=USER;privilege=SELECT;		
NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des AUDIT-Ereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Gültige Werte: SECMAINT
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Gültige Werte: GRANT, REVOKE, IMPLICIT_GRANT, IMPLICIT_REVOKE und UPDATE_DBM_CFG.
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Gibt an, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt: Erfolgreiches Ereignis > = 0 Fehlgeschlagenes Ereignis < 0
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Exemplearebene handelt.
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses.
Origin Node Number (Nummer des Ursprungsknotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinatorknotens)	SMALLINT	Die Knotennummer des Koordinatorknotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.

Tabelle 9. Prüfsatzaufbau für SECMAINT-Ereignisse (Forts.)

<pre>timestamp=1998-06-24-11.57.45.188101;category=SECMAINT;audit event=GRANT; event correlator=4;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.boss.980624155728;application name=db2bp; package schema=NULLID;package name=SQLC28A1; package section=0;object schema=BOSS;object name=T1;object type=TABLE; grantor=BOSS;grantee=WORKER;grantee type=USER;privilege=SELECT;</pre>		
NAME	FORMAT	BESCHREIBUNG
Package Schema (Paketschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(128)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Section Number (Paketabschnittsnummer)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Object Schema (Objektschema)	VARCHAR(128)	Das Schema des Objekts, für das das Prüfereignis generiert wurde.
Object Name (Objektname)	VARCHAR(128)	Der Name des Objekts, für das das Prüfereignis generiert wurde.
Object Type (Objektart)	VARCHAR(32)	Die Art des Objekts, für das das Prüfereignis generiert wurde. Gültige Werte: TABLE, VIEW, ALIAS, FUNCTION, INDEX, PACKAGE, DATA_TYPE, NODEGROUP, SCHEMA, STORED_PROCEDURE, BUFFERPOOL, TABLESPACE, EVENT_MONITOR, TRIGGER, DATABASE, INSTANCE, FOREIGN_KEY, PRIMARY_KEY, UNIQUE_CONSTRAINT, CHECK_CONSTRAINT, WRAPPER, SERVER, NICKNAME, USER MAPPING, SERVER OPTION, TRANSFORM, TYPE MAPPING, FUNCTION MAPPING, SUMMARY TABLES und NONE.
Grantor (Berechtigter)	VARCHAR(128)	Die ID des Berechtigten.
Grantee (Berechtigter)	VARCHAR(128)	Die ID des Berechtigten, für den ein Zugriffsrecht oder eine Berechtigung erteilt oder widerrufen wurde.
Grantee Type (Art des Berechtigten)	VARCHAR(32)	Art des Berechtigten, für den eine Berechtigung erteilt oder widerrufen wurde. Gültige Werte: USER, GROUP oder BOTH.
Privilege or Authority (Zugriffsrecht oder Berechtigung)	CHAR(18)	Gibt an, welche Art von Zugriffsrecht oder Berechtigung erteilt oder widerrufen wurde. Gültige Werte: Die in der Liste im Anschluss an diese Tabelle angegebenen Werte.

In der folgenden Liste sind mögliche Zugriffsrechte und Berechtigungen für SECMAINT-Ereignisse aufgeführt:

0x0000000000000001 Control Table

Zugriffsrecht für Tabellensteuerung erteilt oder widerrufen.

0x0000000000000002 ALTER TABLE

Zugriffsrecht zum Ändern einer Tabelle erteilt oder widerrufen.

0x0000000000000004 ALTER TABLE with GRANT

Zugriffsrecht zum Ändern einer Tabelle mit der Möglichkeit zum Erteilen von Berechtigungen erteilt oder widerrufen.

0x0000000000000008 DELETE TABLE

Zugriffsrecht zum Löschen einer Tabelle erteilt oder widerrufen.

0x0000000000000010 DELETE TABLE with GRANT

Zugriffsrecht zum Löschen einer Tabelle mit der Möglichkeit zum Erteilen von Berechtigungen erteilt oder widerrufen.

0x0000000000000020 Table Index

Zugriffsrecht für einen Index erteilt oder widerrufen.

0x0000000000000040 Table Index with GRANT

Zugriffsrecht für einen Index mit der Möglichkeit zum Erteilen von Berechtigungen erteilt oder widerrufen.

0x0000000000000080 Table INSERT

Zugriffsrecht für eine Einfügung in eine Tabelle erteilt oder widerrufen.

0x0000000000000100 Table INSERT with GRANT

Zugriffsrecht für eine Einfügung in eine Tabelle mit der Möglichkeit zum Erteilen von Berechtigungen erteilt oder widerrufen.

0x0000000000000200 Table SELECT

Zugriffsrecht zum Auswählen in einer Tabelle erteilt oder widerrufen.

0x0000000000000400 Table SELECT with GRANT

Zugriffsrecht zum Auswählen in einer Tabelle mit der Möglichkeit zum Erteilen von Berechtigungen erteilt oder widerrufen.

0x0000000000000800 Table UPDATE

Zugriffsrecht zum Aktualisieren einer Tabelle erteilt oder widerrufen.

0x0000000000001000 Table UPDATE with GRANT

Zugriffsrecht zum Aktualisieren einer Tabelle mit Möglichkeit zum Erteilen von Berechtigungen erteilt oder widerrufen.

0x0000000000002000 Table REFERENCE

Zugriffsrecht zum Verweisen auf eine Tabelle erteilt oder widerrufen.

- j0x0000000000004000 Table REFERENCE with GRANT**
Zugriffsrecht zum Verweisen auf eine Tabelle mit Möglichkeit zum Erteilen von Berechtigungen erteilt oder widerrufen.
- 0x0000000000008000 Package BIND**
Zugriffsrecht BIND zum Binden eines Pakets erteilt oder widerrufen.
- 0x0000000000010000 Package EXECUTE**
Zugriffsrecht EXECUTE zum Ausführen eines Pakets erteilt oder widerrufen.
- 0x0000000000020000 CREATEIN Schema**
Zugriffsrecht CREATEIN zum Erstellen in einem Schema erteilt oder widerrufen.
- 0x0000000000040000 CREATEIN Schema with GRANT**
Zugriffsrecht CREATEIN zum Erstellen in einem Schema mit Möglichkeit zum Erteilen von Berechtigungen erteilt oder widerrufen.
- 0x0000000000080000 DROPIN Schema**
Zugriffsrecht DROPIN zum Löschen in einem Schema erteilt oder widerrufen.
- 0x0000000000100000 DROPIN Schema with GRANT**
Zugriffsrecht DROPIN zum Löschen in einem Schema mit Möglichkeit zum Erteilen von Berechtigungen erteilt oder widerrufen.
- 0x0000000000200000 ALTERIN Schema**
Zugriffsrecht ALTERIN zum Ändern in einem Schema erteilt oder widerrufen.
- 0x0000000000400000 ALTERIN Schema with GRANT**
Zugriffsrecht ALTERIN zum Ändern in einem Schema mit Möglichkeit zum Erteilen von Berechtigungen erteilt oder widerrufen.
- 0x0000000000800000 DBADM Authority**
Berechtigung DBADM erteilt oder widerrufen.
- 0x0000000001000000 CREATETAB Authority**
CREATETAB-Berechtigung erteilt oder widerrufen.
- 0x0000000002000000 BINDADD Authority**
BINDADD-Berechtigung erteilt oder widerrufen.
- 0x0000000004000000 CONNECT Authority**
CONNECT-Berechtigung erteilt oder widerrufen.
- 0x0000000008000000 Create not fenced Authority**
Berechtigung 'Create not fenced' erteilt oder widerrufen.
- 0x0000000010000000 Implicit Schema Authority**
Berechtigung 'Implicit schema' erteilt oder widerrufen.

0x0000000020000000 Server PASSTHRU

Zugriffsrecht für die Verwendung der Durchgriffsfunktion für diesen Server (Datenquelle zusammengesetzter Datenbanken) erteilt oder widerrufen.

0x0000000100000000 Table Space USE

Zugriffsrecht zum Erstellen einer Tabelle in einem Tabellenbereich erteilt oder widerrufen.

0x0000000200000000 Table Space USE with GRANT

Zugriffsrecht zum Erstellen einer Tabelle in einem Tabellenbereich mit der Möglichkeit zum Erteilen von Zugriffsrechten erteilt oder widerrufen.

0x0000000400000000 Column UPDATE

Zugriffsrecht zum Aktualisieren einer oder mehrerer bestimmter Spalten einer Tabelle erteilt oder widerrufen.

0x0000000800000000 Column UPDATE with GRANT

Zugriffsrecht zum Aktualisieren einer oder mehrerer bestimmter Spalten einer Tabelle mit Möglichkeit zum Erteilen von Zugriffsrechten erteilt oder widerrufen.

0x0000001000000000 Column REFERENCE

Zugriffsrecht zum Verweisen auf eine oder mehrere bestimmte Spalten einer Tabelle erteilt oder widerrufen.

0x0000002000000000 Column REFERENCE with GRANT

Zugriffsrecht zum Verweisen auf eine oder mehrere bestimmte Spalten einer Tabelle mit Möglichkeit zum Erteilen von Zugriffsrechten erteilt oder widerrufen.

0x0000004000000000 LOAD Authority

LOAD-Berechtigung erteilt oder widerrufen.

Tabelle 10. Prüfsatzaufbau für SYSADMIN-Ereignisse

<code>timestamp=1998-06-24-11.54.04.129923;category=SYSADMIN;audit event=DB2AUDIT; event correlator=1;event status=0; userid=boss;authid=BOSS; application id=*LOCAL.boss.980624155404;application name=db2audit;</code>		
NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des AUDIT-Ereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Gültige Werte: SYSADMIN

Tabelle 10. Prüfsatzaufbau für SYSADMIN-Ereignisse (Forts.)

<pre>timestamp=1998-06-24-11.54.04.129923;category=SYSADMIN;audit event=DB2AUDIT; event correlator=1;event status=0; userid=boss;authid=BOSS; application id=*LOCAL.boss.980624155404;application name=db2audit;</pre>		
NAME	FORMAT	BESCHREIBUNG
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Gültige Werte: Die in der Liste im Anschluss an diese Tabelle angegebenen Werte.
Event Correlator (Ereignis-korrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Gibt an, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt: Erfolgreiches Ereignis > = 0 Fehlgeschlagenes Ereignis < 0
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Exemplebarebene handelt.
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses.
Origin Node Number (Nummer des Ursprungsknotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinatorknotens)	SMALLINT	Die Knotennummer des Koordinatorknotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Package Schema (Paketeschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(128)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.

Tabelle 10. Prüfsatzaufbau für SYSADMIN-Ereignisse (Forts.)

<pre>timestamp=1998-06-24-11.54.04.129923;category=SYSADMIN;audit event=DB2AUDIT; event correlator=1;event status=0; userid=boss;authid=BOSS; application id=*LOCAL.boss.980624155404;application name=db2audit;</pre>		
NAME	FORMAT	BESCHREIBUNG
Package Section Number (Paketabschnittsnummer)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.

In der folgenden Liste sind mögliche SYSADMIN-Prüfereignisse aufgeführt:

Tabelle 11. SYSADMIN-Prüfereignisse

START_DB2	ROLLFORWARD_DB
STOP_DB2	SET_RUNTIME_DEGREE
CREATE_DATABASE	SET_TABLESPACE_CONTAINERS
DROP_DATABASE	UNCATALOG_DB
UPDATE_DBM_CFG	UNCATALOG_DCS_DB
UPDATE_DB_CFG	UNCATALOG_NODE
CREATE_TABLESPACE	UPDATE_ADMIN_CFG
DROP_TABLESPACE	UPDATE_MON_SWITCHES
ALTER_TABLESPACE	LOAD_TABLE
RENAME_TABLESPACE	DB2AUDIT
CREATE_NODEGROUP	SET_APPL_PRIORITY
DROP_NODEGROUP	CREATE_DB_AT_NODE
ALTER_NODEGROUP	KILLDBM
CREATE_BUFFERPOOL	MIGRATE_SYSTEM_DIRECTORY
DROP_BUFFERPOOL	DB2REMOT
ALTER_BUFFERPOOL	DB2AUD
CREATE_EVENT_MONITOR	MERGE_DBM_CONFIG_FILE
DROP_EVENT_MONITOR	UPDATE_CLI_CONFIGURATION
ENABLE_MULTIPAGE	OPEN_TABLESPACE_QUERY
MIGRATE_DB_DIR	SINGLE_TABLESPACE_QUERY
DB2TRC	CLOSE_TABLESPACE_QUERY
DB2SET	FETCH_TABLESPACE
ACTIVATE_DB	OPEN_CONTAINER_QUERY
ADD_NODE	FETCH_CONTAINER_QUERY
BACKUP_DB	CLOSE_CONTAINER_QUERY
CATALOG_NODE	GET_TABLESPACE_STATISTICS
CATALOG_DB	DESCRIBE_DATABASE
CATALOG_DCS_DB	ESTIMATE_SNAPSHOT_SIZE
CHANGE_DB_COMMENT	READ_ASYNC_LOG_RECORD
DEACTIVATE_DB	PRUNE_RECOVERY_HISTORY
DROP_NODE_VERIFY	UPDATE_RECOVERY_HISTORY
FORCE_APPLICATION	QUIESCE_TABLESPACE
GET_SNAPSHOT	UNLOAD_TABLE
LIST_DRDA_INDOUBT_TRANSACTIONS	UPDATE_DATABASE_VERSION
MIGRATE_DB	CREATE_INSTANCE
RESET_ADMIN_CFG	DELETE_INSTANCE
RESET_DB_CFG	SET_EVENT_MONITOR
RESET_DBM_CFG	GRANT_DBADM
RESET_MONITOR	REVOKE_DBADM
RESTORE_DB	GRANT_DB_AUTHORITIES
	REVOKE_DB_AUTHORITIES
	REDIST_NODEGROUP

Tabelle 12. Prüfsatzaufbau für VALIDATE-Ereignisse

timestamp=1998-06-24-08.42.11.527490;category=VALIDATE;audit event=CHECK_GROUP_MEMBERSHIP; event correlator=2;event status=-1092; database=F00;userid=boss;authid=BOSS;execution id=newton; application id=*LOCAL.newton.980624124210;application name=testapp; auth type=SERVER;		
NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des AUDIT-Ereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Gültige Werte: VALIDATE
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Gültige Werte sind z. B.: GET_GROUPS, GET_USERID, AUTHENTICATE_PASSWORD und VALIDATE_USER.
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Gibt an, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt: Erfolgreiches Ereignis > = 0 Fehlgeschlagenes Ereignis < 0
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Exemplarebene handelt.
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses.
Execution ID (Ausführungs-ID)	VARCHAR(1024)	Die Ausführungs-ID, die beim Auftreten des Prüfereignisses verwendet wurde.
Origin Node Number (Nummer des Ursprungsknotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinator-knotens)	SMALLINT	Die Knotennummer des Koordinator-knotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.

Tabelle 12. Prüfsatzaufbau für VALIDATE-Ereignisse (Forts.)

timestamp=1998-06-24-08.42.11.527490;category=VALIDATE;audit event=CHECK_GROUP_MEMBERSHIP; event correlator=2;event status=-1092; database=F00;userid=boss;authid=BOSS;execution id=newton; application id=*LOCAL.newton.980624124210;application name=testapp; auth type=SERVER;		
NAME	FORMAT	BESCHREIBUNG
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Authentication Type (Authentifizierungsart)	VARCHAR(32)	Die Authentifizierungsart beim Auftreten des Prüfereignisses.
Package Schema (Paketschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(128)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Section Number (Paketabschnittsnummer)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.

Tabelle 13. Prüfsatzaufbau für CONTEXT-Ereignisse

timestamp=1998-06-24-08.42.41.476840;category=CONTEXT;audit event=EXECUTE_IMMEDIATE; event correlator=3; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; package schema=NULLID;package name=SQLC28A1; package section=203;text=create table audit(c1 char(10), c2 integer);		
NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des AUDIT-Ereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Gültige Werte: CONTEXT
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Gültige Werte: Die in der Liste im Anschluss an diese Tabelle angegebenen Werte.
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Gibt an, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Exemplarebene handelt.

Tabelle 13. Prüfsatzaufbau für CONTEXT-Ereignisse (Forts.)

timestamp=1998-06-24-08.42.41.476840;category=CONTEXT;audit event=EXECUTE_IMMEDIATE; event correlator=3; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; package schema=NULLID;package name=SQLC28A1; package section=203;text=create table audit(c1 char(10), c2 integer);		
NAME	FORMAT	BESCHREIBUNG
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses.
Origin Node Number (Nummer des Ursprungsknotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinatorknotens)	SMALLINT	Die Knotennummer des Koordinatorknotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Package Schema (Paketschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(128)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Section Number (Paketabschnittsnummer)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Statement Text (statement) - Anweisungstext (Anweisung)	CLOB(32K)	Text der SQL-Anweisung, falls vorhanden. Null, wenn kein Text für die SQL-Anweisung verfügbar ist.

In der folgenden Liste sind mögliche CONTEXT-Prüfereignisse aufgeführt:

Tabelle 14. CONTEXT-Prüfereignisse

CONNECT	SET_APPL_PRIORITY
CONNECT_RESET	RESET_DB_CFG
ATTACH	GET_DB_CFG
DETACH	GET_DFLT_CFG
DARI_START	UPDATE_DBM_CFG
DARI_STOP	SET_MONITOR
BACKUP_DB	GET_SNAPSHOT
RESTORE_DB	ESTIMATE_SNAPSHOT_SIZE
ROLLFORWARD_DB	RESET_MONITOR
OPEN_TABLESPACE_QUERY	OPEN_HISTORY_FILE
FETCH_TABLESPACE	CLOSE_HISTORY_FILE
CLOSE_TABLESPACE_QUERY	FETCH_HISTORY_FILE
OPEN_CONTAINER_QUERY	SET_RUNTIME_DEGREE
CLOSE_CONTAINER_QUERY	UPDATE_AUDIT
FETCH_CONTAINER_QUERY	DBM_CFG_OPERATION
SET_TABLESPACE_CONTAINERS	DISCOVER
GET_TABLESPACE_STATISTIC	OPEN_CURSOR
READ_ASYNC_LOG_RECORD	CLOSE_CURSOR
QUIESCE_TABLESPACE	FETCH_CURSOR
LOAD_TABLE	EXECUTE
UNLOAD_TABLE	EXECUTE_IMMEDIATE
UPDATE_RECOVERY_HISTORY	PREPARE
PRUNE_RECOVERY_HISTORY	DESCRIBE
SINGLE_TABLESPACE_QUERY	BIND
LOAD_MSG_FILE	REBIND
UNQUIESCE_TABLESPACE	RUNSTATS
ENABLE_MULTIPAGE	REORG
DESCRIBE_DATABASE	REDISTRIBUTE
DROP_DATABASE	COMMIT
CREATE_DATABASE	ROLLBACK
ADD_NODE	REQUEST_ROLLBACK
FORCE_APPLICATION	IMPLICIT_REBIND

Tipps und Verfahrensweisen für die Prüffunktion

Beim Arbeiten mit CHECKING-Ereignissen wird meist das Feld für die Objektart im Prüfsatz überprüft, um festzustellen, ob die Benutzer-ID, die auf das Objekt zuzugreifen versucht, über das erforderliche Zugriffsrecht bzw. die erforderliche Berechtigung verfügt. Beispiel: Wenn ein Benutzer versucht, mit ALTER eine Tabellenspalte hinzuzufügen, weist der Prüfsatz des CHECKING-Ereignisses darauf hin, dass ein Zugriff mit „ALTER“ auf ein Objekt des Typs „TABLE“ versucht wurde (Hinweis: der Objekttyp lautet nicht COLUMN, weil hier die Tabellenzugriffsrechte geprüft werden müssen).

Muss beim Überprüfen jedoch festgestellt werden, ob die erforderliche Datenbankberechtigung vorhanden ist, mit der eine Benutzer-ID eine CREATE- oder BIND-Operation an einem Objekt ausführen oder ein Objekt löschen kann, dann erfolgt zwar eine Prüfung in der Datenbank, aber im Feld für die Objektart wird das zu erstellende, zu bindende oder zu löschende Objekt angegeben (und nicht die Datenbank).

Bei der Indexerstellung für eine Tabelle ist das Zugriffsrecht zum Erstellen eines Index erforderlich, deshalb wird im Prüfsatz des CHECKING-Ereignisses die Zugriffsart „index“ und nicht „create“ angegeben.

Beim Binden eines bereits vorhandenen Pakets wird ein OBJMAINT-Prüfsatz für die DROP-Operation des Pakets erstellt und danach ein weiterer OBJMAINT-Prüfsatz für die CREATE-Operation der neuen Paketkopie.

Die SQL-Datendefinitionssprache (Data Definition Language, DDL) kann OBJMAINT- oder SECMAINT-Ereignisse generieren, die als erfolgreich protokolliert werden. Es kann jedoch vorkommen, dass nach der Protokollierung des Ereignisses durch einen nachfolgenden Fehler eine ROLLBACK-Operation ausgelöst wird. Dadurch würde das Objekt als nicht erstellt bzw. die GRANT- oder REVOKE-Operation als unvollständig ausgewiesen. In diesem Fall erweist sich die Verwendung von CONTEXT-Ereignissen als hilfreich. Die Prüfsätze von CONTEXT-Ereignissen, insbesondere die abschließende Anweisung des Ereignisses, geben an, wie die versuchte Operation beendet wurde.

Beim Extrahieren von Prüfsätzen in begrenztem ASCII-Format, das sich zum Laden in eine relationale DB2-Datenbank eignet, ist darauf zu achten, dass im Textfeld der Anweisung der richtige Begrenzer verwendet wird. Dazu kann beim Extrahieren der begrenzten ASCII-Datei folgende Anweisung eingegeben werden:

```
db2audit extract delasc delimiter <ladebegrenzer>
```

Dabei kann der *ladebegrenzer* ein Einzelzeichen (z. B. ") sein oder ein Vierbytezeichen, das eine Hexadezimalzahl darstellt (z. B. „0xff“). Beispiele für gültige Befehle sind:

```
db2audit extract delasc
db2audit extract delasc delimiter !
db2audit extract delasc delimiter 0xff
```

Wenn beim Extrahieren nicht der Standardladebegrenzer („'") verwendet wurde, sollte die Option MODIFIED BY im Befehl LOAD verwendet werden. Es folgt ein abgekürztes Beispiel für den Befehl LOAD mit dem Begrenzer „0xff“:

```
db2 load from context.del of del modified by char del 0xff replace into ...
```

Dadurch wird der Standardzeichenfolgebegrenzer „0xff“ außer Kraft gesetzt.

Steuern der Aktivitäten der DB2-Prüffunktion

Für diese Darstellung zur Steuerung der Aktivitäten der Prüffunktion wird das folgende vereinfachte Szenario verwendet: Ein Benutzer, *newton*, führt die Anwendung *testapp* aus, die eine Verbindung aufbaut und eine Tabelle erstellt. Diese Anwendung wird in allen nachfolgend aufgeführten Beispielen verwendet.

Zunächst ein extremes Beispiel: Wenn Sie sich entschlossen haben, alle erfolgreichen und fehlgeschlagenen Prüfereignisse zu prüfen, müssten Sie die Prüffunktion wie folgt konfigurieren:

```
db2audit configure scope all status both
```

Anmerkung: Dadurch werden Prüfsätze für alle möglichen prüfbaren Ereignisse erstellt. Dies führt dazu, dass zahlreiche Prüfsätze in das Prüfprotokoll geschrieben werden, was wiederum die Verarbeitungsleistung Ihres Datenbankmanagers beeinträchtigt. Dieses extreme Beispiel ist hier nur zu Demonstrationszwecken angeführt; es empfiehlt sich nicht, die Prüffunktion mit dem oben angegebenen Befehl zu konfigurieren.

Nachdem die Prüffunktion mit dieser Konfiguration (mit Hilfe von „db2audit start“) gestartet und anschließend die Anwendung *testapp* ausgeführt wurde, werden die folgenden Datensätze generiert und in das Prüfprotokoll geschrieben. Durch Extrahieren der Prüfsätze aus dem Protokoll werden Sie feststellen, dass für die beiden von der Anwendung ausgeführten Aktionen die folgenden Datensätze generiert wurden:

Aktion Art des erstellten Datensatzes

CONNECT

```
timestamp=1998-06-24-08.42.10.555345;category=CONTEXT;  
audit event=CONNECT;event correlator=2;database=F00;  
application id=*LOCAL.newton.980624124210;  
application name=testapp;
```

```
timestamp=1998-06-24-08.42.10.944374;category=VALIDATE;  
audit event=AUTHENTICATION;event correlator=2;event status=0;  
database=F00;userid=boss;authid=BOSS;execution id=newton;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
auth type=SERVER;
```

```
timestamp=1998-06-24-08.42.11.527490;category=VALIDATE;  
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;  
event status=-1092;database=F00;userid=boss;authid=BOSS;  
execution id=newton;application id=*LOCAL.newton.980624124210;  
application name=testapp;auth type=SERVER;
```

```
timestamp=1998-06-24-08.42.11.561187;category=VALIDATE;  
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;  
event status=-1092;database=F00;userid=boss;authid=BOSS;  
execution id=newton;application id=*LOCAL.newton.980624124210;  
application name=testapp;auth type=SERVER;
```

```
timestamp=1998-06-24-08.42.11.594620;category=VALIDATE;  
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;  
event status=-1092;database=F00;userid=boss;authid=BOSS;  
execution id=newton;application id=*LOCAL.newton.980624124210;  
application name=testapp;auth type=SERVER;
```

```
timestamp=1998-06-24-08.42.11.622984;category=CHECKING;  
audit event=CHECKING_OBJECT;event correlator=2;event status=0;  
database=F00;userid=boss;authid=BOSS;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
object name=F00;object type=DATABASE;access approval reason=DATABASE;  
access attempted=CONNECT;
```

```
timestamp=1998-06-24-08.42.11.801554;category=CONTEXT;  
audit event=COMMIT;event correlator=2;database=F00;userid=boss;  
authid=BOSS;application id=*LOCAL.newton.980624124210;  
application name=testapp;
```

```
timestamp=1998-06-24-08.42.41.450975;category=CHECKING;  
audit event=CHECKING_OBJECT;event correlator=2;event status=0;  
database=F00;userid=boss;authid=BOSS;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
package schema=NULLID;package name=SQLC28A1;object schema=NULLID;  
object name=SQLC28A1;object type=PACKAGE;  
access approval reason=OBJECT;access attempted=EXECUTE;
```

CREATE TABLE

```
timestamp=1998-06-24-08.42.41.476840;category=CONTEXT;  
audit event=EXECUTE_IMMEDIATE;event correlator=3;database=F00;  
userid=boss;authid=BOSS;application id=*LOCAL.newton.980624124210;  
application name=testapp;package schema=NULLID;package name=SQLC28A1;  
package section=203;text=create table audit(c1 char(10), c2 integer);
```

```
timestamp=1998-06-24-08.42.41.539692;category=CHECKING;  
audit event=CHECKING_OBJECT;event correlator=3;event status=0;  
database=F00;userid=boss;authid=BOSS;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
package schema=NULLID;package name=SQLC28A1;package section=0;  
object schema=BOSS;object name=AUDIT;object type=TABLE;  
access approval reason=DATABASE;access attempted=CREATE;
```

```
timestamp=1998-06-24-08.42.41.570876;category=CHECKING;  
audit event=CHECKING_OBJECT;event correlator=3;event status=0;  
database=F00;userid=boss;authid=BOSS;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
package schema=NULLID;package name=SQLC28A1;package section=0;  
object name=BOSS;object type=SCHEMA;access approval reason=DATABASE;  
access attempted=CREATE;
```

```
timestamp=1998-06-24-08.42.41.957524;category=OBJMAINT;  
audit event=CREATE_OBJECT;event correlator=3;event status=0;  
database=F00;userid=boss;authid=BOSS;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
package schema=NULLID;package name=SQLC28A1;package section=0;  
object schema=BOSS;object name=AUDIT;object type=TABLE;
```

```
timestamp=1998-06-24-08.42.42.018900;category=CONTEXT;audit event=COMMIT;  
event correlator=3;database=F00;userid=boss;authid=BOSS;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
package schema=NULLID;package name=SQLC28A1;
```

Wie Sie sehen, wird bei dieser Prüfkongfiguration, die dafür sorgt, dass alle möglichen Prüfereignisse und -arten geprüft werden, eine beträchtliche Anzahl von Prüfsätzen generiert.

In den meisten Fällen werden Sie die Prüffunktion jedoch so konfigurieren, dass nur eine begrenzte bzw. zielgerichtete Auswahl von Ereignissen geprüft wird. Beispielsweise könnten Sie nur diejenigen Ereignisse prüfen, die fehlgeschlagen sind. Dazu könnte die Prüffunktion wie folgt konfiguriert werden:

```
db2audit configure scope audit,checking,objmaint,secmaint,sysadmin,  
validate status failure
```

Anmerkung: Dies ist die Grundkonfiguration der Prüffunktion, die beim Zurücksetzen der Prüfkongfiguration aktiviert wird.

Nachdem die Prüffunktion mit dieser Konfiguration gestartet und anschließend die Anwendung *testapp* ausgeführt wurde, werden die folgenden Datensätze generiert und in das Prüfprotokoll geschrieben. (Hierbei wird vorausgesetzt, dass *testapp* vorher nicht ausgeführt wurde.) Durch Extrahieren der Prüfsätze aus dem Protokoll werden Sie feststellen, dass für die beiden von der Anwendung ausgeführten Aktionen die folgenden Datensätze generiert wurden:

Aktion Art des erstellten Datensatzes

CONNECT

```
timestamp=1998-06-24-08.42.11.527490;category=VALIDATE;  
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;  
event status=-1092;database=F00;userid=boss;authid=BOSS;  
execution id=newton;application id=*LOCAL.newton.980624124210;  
application name=testapp;auth type=SERVER;
```

```
timestamp=1998-06-24-08.42.11.561187;category=VALIDATE;  
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;  
event status=-1092;database=F00;userid=boss;authid=BOSS;  
execution id=newton;application id=*LOCAL.newton.980624124210;  
application name=testapp;auth type=SERVER;
```

```
timestamp=1998-06-24-08.42.11.594620;category=VALIDATE;  
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;  
event status=-1092;database=F00;userid=boss;authid=BOSS;  
execution id=newton;application id=*LOCAL.newton.980624124210;  
application name=testapp;auth type=SERVER;
```

CREATE TABLE

(keine)

Diese Prüfkongfiguration, die die Prüfung aller möglichen Prüfereignisse (ausgenommen CONTEXT) anfordert, sofern das Prüfereignis fehlgeschlagen ist, generiert eine geringere Anzahl von Prüfsätzen. Durch Ändern der Prüfkongfiguration können Sie Art und Zusammensetzung der generierten Prüfsätze steuern.

Mit der Prüffunktion können Sie auch Prüfsätze erstellen, wenn den zu prüfenden Ereignissen erfolgreich Zugriffsrechte für ein Objekt erteilt wurden. In diesem Fall könnten Sie die Prüffunktion wie folgt konfigurieren:

```
db2audit configure scope checking status success
```


Nachdem die Prüffunktion mit dieser Konfiguration gestartet und anschließend die Anwendung *testapp* ausgeführt wurde, werden die folgenden Datensätze generiert und in das Prüfprotokoll geschrieben. (Hierbei wird vorausgesetzt, dass *testapp* vorher nicht ausgeführt wurde.) Durch Extrahieren der Prüfsätze aus dem Protokoll werden Sie feststellen, dass für die beiden von der Anwendung ausgeführten Aktionen die folgenden Datensätze generiert wurden:

Aktion Art des erstellten Datensatzes

CONNECT

```
timestamp=1998-06-24-08.42.11.622984;category=CHECKING;  
audit event=CHECKING_OBJECT;event correlator=2;event status=0;  
database=F00;userid=boss;authid=BOSS;
```

```
timestamp=1998-06-24-08.42.41.450975;category=CHECKING;  
audit event=CHECKING_OBJECT;event correlator=2;event status=0;  
database=F00;userid=boss;authid=BOSS;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
package schema=NULLID;package name=SQLC28A1;object schema=NULLID;  
object name=SQLC28A1;object type=PACKAGE;  
access approval reason=OBJECT;access attempted=EXECUTE;
```

```
timestamp=1998-06-24-08.42.41.539692;category=CHECKING;  
audit event=CHECKING_OBJECT;event correlator=3;event status=0;  
database=F00;userid=boss;authid=BOSS;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
package schema=NULLID;package name=SQLC28A1;package section=0;  
object schema=BOSS;object name=AUDIT;object type=TABLE;  
access approval reason=DATABASE;access attempted=CREATE;
```

```
timestamp=1998-06-24-08.42.41.570876;category=CHECKING;  
audit event=CHECKING_OBJECT;event correlator=3;event status=0;  
database=F00;userid=boss;authid=BOSS;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
package schema=NULLID;package name=SQLC28A1;package section=0;  
object name=BOSS;object type=SCHEMA;access approval reason=DATABASE;  
access attempted=CREATE;
```

CREATE TABLE

```
(keine)
```

Teil 4. Versetzen von Daten

Kapitel 7. Dienstprogramme für das Versetzen von Daten

Mit dem Dienstprogramm LOAD können Daten in Tabellen versetzt, vorhandene Indizes erweitert und statistische Daten generiert werden. Mit LOAD werden die Daten viel schneller als mit dem Dienstprogramm IMPORT versetzt, wenn es sich um umfangreiche Datenmengen handelt. Daten, die mit dem Dienstprogramm EXPORT aus Tabellen geladen wurden, können mit dem Dienstprogramm LOAD wieder geladen werden.

Das Dienstprogramm AutoLoader trennt umfangreiche Mengen von Daten und lädt die getrennten Daten in verschiedene Partitionen einer partitionierten Datenbank.

Mit Hilfe der Dienstprogramme IMPORT und EXPORT werden Daten zwischen einer Tabelle oder Sicht und einem anderen Datenbank- oder Tabellenkalkulationsprogramm sowie zwischen DB2-Datenbanken und Host-Datenbanken mit Hilfe von DB2 Connect versetzt.

Data Replication (bisher DataPropagator Relational (DPROPR)) ist eine Komponente von DB2 Universal Database, die das automatische Kopieren von Tabellenaktualisierungen in andere Tabellen anderer relationaler Datenbanken unter DB2 ermöglicht.

Anmerkung: Alle Informationen zu diesen Themen sowie die vergleichbaren Themen aus den Handbüchern *Command Reference* und *Administrative API Reference* wurden im Handbuch *Versetzen von Daten Dienstprogramme und Referenz* zusammengefasst.

Informationen zu diesen Themen finden Sie ausschließlich im Handbuch *Versetzen von Daten Dienstprogramme und Referenz*.

Weitere Informationen zur Replikation finden Sie in *Replikation Benutzer- und Referenzhandbuch*.

Teil 5. Wiederherstellung

Kapitel 8. Wiederherstellen einer Datenbank

Wenn ein Fehler auftritt, müssen Sie in der Lage sein, ihre Datenbank wiederherzustellen. Fehler können Stromausfälle, Fehler bei Anwendungen sowie bei Datenträgern und Speichern sein. Damit Sie in einem solchen Fall die Datenbank wiederherstellen können, benötigen Sie Sicherungen oder Kopien der gesamten Datenbank oder der Tabellenbereiche, die die Datenbank ausmachen. Diese Sicherungen können nach einem Datenbankfehler zur Wiederherstellung der Datenbank verwendet werden.

Bei der Wiederherstellung nach Systemabsturz wird automatisch versucht, die Datenbank nach einem Fehler wiederherzustellen. Die Wiederherstellung nach Systemabsturz verhindert, dass die Datenbank in einem inkonsistenten oder unbenutzbaren Zustand bleibt. Transaktionen mit der Datenbank können nach dem Auftreten eines Fehlers bei der Datenbank unvollständig bleiben. Die Wiederherstellung nach Systemabsturz macht unvollständige Transaktionen entweder rückgängig oder schreibt vollständige Transaktionen fest. Durch diese Maßnahmen wird die Datenbank konsistent und benutzbar.

Es gibt zwei andere Arten der Wiederherstellung, wenn die Datenbank beschädigt wurde, und Sie nicht sicher sind, ob der Inhalt der Datenbank in Ordnung ist. Die beiden Methoden zur Wiederherstellung einer beschädigten Datenbank sind: Versionswiederherstellung und aktualisierende Wiederherstellung. Wenn Sie mit einer Datenbank im Lesezugriff arbeiten oder es für Sie nicht wichtig ist, dass Transaktionen in der Datenbank aufgezeichnet werden, ist die Versionswiederherstellung für Sie möglicherweise ausreichend. Wenn Sie eine Sicherung der Datenbank vorgenommen haben, können Sie diese Kopie der Datenbank anwenden oder wiederherstellen. Dies wird als Versionswiederherstellung bezeichnet. Wenn Sie mit einer Datenbank arbeiten, auf die Transaktionen angewendet wurden, und Sie Wert darauf legen, dass alle diese Änderungen auf die Datenbank angewendet wurden, müssen Sie eine aktualisierende Wiederherstellung ausführen. Eine aktualisierende Wiederherstellung umfasst die Wiederherstellung einer Sicherung der Datenbank. Anschließend müssen Sie einen Datensatz mit den Protokollen, in denen die Transaktionen an der Datenbank aufgezeichnet sind, anwenden. Die Anwendung der Protokolle führt dazu, dass alle Aktivitäten mit der Datenbank wiederholt werden, so dass die Datenbank den Status kurz vor dem Fehlerpunkt erreicht. Mit dieser Wiederherstellungsmethode gehen keine Änderungen an der Datenbank verloren. Das Protokollieren ist der Schlüssel für diese Wiederherstellungsmethode.

Anmerkung: Alle Informationen zu diesen Themen sowie die vergleichbaren Themen aus den Handbüchern *Command Reference* und *Administrative API Reference* wurden im Handbuch *Data Recovery and High Availability Guide and Reference* zusammengefasst.

Informationen zu diesen Themen finden Sie ausschließlich im Handbuch *Data Recovery and High Availability Guide and Reference*.

Teil 6. Anhänge und Schlussteil

Anhang A. Namenskonventionen

Lesen Sie den Abschnitt, der die Namenskonventionen beschreibt, zu denen Sie weitere Informationen wünschen:

- „Allgemeine Namenskonventionen“
- „Namenskonventionen für Objektnamen“ auf Seite 370
- „Beibehaltung von Werten in Groß-/Kleinschreibung in einem System zusammenschlossener Datenbanken“ auf Seite 374

Allgemeine Namenskonventionen

Soweit nicht anders angegeben, können alle Namen die folgenden Zeichen enthalten:

- A bis Z. Bei den meisten Namen werden die Zeichen A bis Z von Klein- in Großschreibung umgesetzt.
- 0 - 9
- @, #, \$ und _ (Unterstreichungszeichen)

Namen dürfen nicht mit einer Ziffer oder einem Unterstreichungszeichen beginnen.

Verwenden Sie zur Benennung von Tabellen, Sichten, Spalten, Indizes und Berechtigungs-IDs keine für SQL reservierten Wörter. Eine Liste der für SQL reservierten Wörter finden Sie im Handbuch *SQL Reference*.

Es gibt weitere Sonderzeichen, die abhängig vom Betriebssystem und dem Einsatzort von DB2 möglicherweise verwendet werden können. Es gibt jedoch keine Garantie für die Verwendung dieser Zeichen. Es wird nicht empfohlen, diese weiteren Sonderzeichen bei der Benennung von Objekten in der Datenbank zu verwenden.

Namenskonventionen für Objektnamen

Für alle Objekte gelten die allgemeinen Namenskonventionen. Zusätzlich gelten für einige Objekte weitere Einschränkungen, die unten erläutert werden.

Table 15. Namenskonventionen für Datenbanknamen, Aliasnamen der Datenbank und Exemplarnamen

Objekte	Richtlinien
<ul style="list-style-type: none">• Datenbanken• Aliasnamen der Datenbank• Exemplare	<ul style="list-style-type: none">• Datenbanknamen müssen innerhalb der Position, in der sie katalogisiert werden, eindeutig sein. In auf UNIX basierenden Implementierungen von DB2 ist diese Position ein Verzeichnispfad, während in Windows-Implementierungen diese Position ein Laufwerksbuchstabe ist.• Aliasnamen der Datenbank müssen innerhalb des Systemdatenbankverzeichnisses eindeutig sein. Wenn eine neue Datenbank erstellt wird, ist der Standardwert des Aliasnamens der Datenbankname. Daher können Sie keine Datenbank mit einem Namen erstellen, der bereits als Datenbankaliasname verwendet wird, selbst wenn es keine andere Datenbank mit diesem Namen gibt.• Datenbanknamen, Aliasnamen der Datenbank und Exemplarnamen können bis zu 8 Byte umfassen.• Bei Systemen mit Windows NT und Windows 2000 darf kein Exemplar einen Servicenamen tragen. <p>Anmerkung: Zur Vermeidung von Problemen sollten Sie die Sonderzeichen @, # und \$ nicht in einem Datenbanknamen benutzen, wenn Sie beabsichtigen, die Datenbank in einer Übertragungsumgebung zu verwenden. Darüber hinaus sollten Sie diese Zeichen nicht benutzen, wenn Sie die Datenbank in einer anderen Sprache verwenden möchten, weil diese Zeichen nicht auf allen Tastaturen in gleicher Weise verfügbar sind.</p>

Tabelle 16. Namenskonventionen für Datenbankobjekte

Objekte	Richtlinien
<ul style="list-style-type: none"> • Aliasnamen • Pufferpools • Spalten • Ereignismonitore • Indizes • Methoden • Knotengruppen • Schemata • Gespeicherte Prozeduren • Tabellen • Tabellenbereiche • Auslöser • Benutzerdefinierte Funktionen (UDFs) • Benutzerdefinierte Datentypen (UDTs) • Sichten 	<p>Können bis zu 18 Byte umfassen. Es gibt folgende <i>Ausnahmen</i>:</p> <ul style="list-style-type: none"> • Tabellennamen (einschließlich Namen von Sichten, Übersichtstabellen, Aliasnamen und Korrelationsnamen), die bis zu 128 Byte umfassen können. • Spaltennamen, die bis zu 30 Byte umfassen können. • Schemennamen, die bis zu 30 Byte umfassen können. • Objektnamen können ferner folgende Zeichen enthalten: <ul style="list-style-type: none"> – Gültige diakritische Zeichen (zum Beispiel ö) – Mehrbytezeichen außer Mehrbyteleerzeichen (für Mehrbyteumgebungen)

Zusätzliche Informationen zu Schemennamen

- Tabelle mit Schemennamen, die länger als 18 Byte sind, können nicht repliziert werden.
- Benutzerdefinierte Datentypen (UDTs) dürfen keine Schemennamen aufweisen, die länger als 8 Byte sind.
- Die folgenden Schemennamen sind reservierte Wörter und dürfen nicht verwendet werden: SYSCAT, SYSFUN, SYSIBM, SYSSTAT.
- Um zukünftigen Migrationsproblemen vorzubeugen, sollten Sie keine Schemennamen verwenden, die mit der Zeichenfolge SYS beginnen. Der Datenbankmanager lässt die Erstellung von Auslösern, benutzerdefinierten Datentypen bzw. benutzerdefinierten Funktionen mit Schemennamen, die mit SYS beginnen, nicht zu.
- Es wird empfohlen, das Wort SESSION nicht als Schemennamen zu verwenden. Deklarierte temporäre Tabellen müssen mit SESSION qualifiziert werden. Daher ist es möglich, dass eine Anwendung eine temporäre Datei mit einem Namen deklariert, der mit dem einer permanenten Tabelle identisch ist. In einem solchen Fall kann die Anwendungslogik übermäßig kompliziert werden. Außer im Zusammenhang mit deklarierten temporären Tabellen sollte die Verwendung des Schemennamens SESSION also vermieden werden.

Tabelle 17. Namenskonventionen für Benutzername, Benutzer-ID und Gruppenname

Objekte	Richtlinien
<ul style="list-style-type: none"> • Gruppennamen • Benutzernamen • Benutzer-IDs 	<ul style="list-style-type: none"> • Gruppennamen können bis zu 8 Byte umfassen. • Benutzer-IDs in UNIX-basierten Systemen können bis zu 8 Zeichen enthalten. • Benutzernamen unter Windows können bis zu 30 Zeichen enthalten. Windows NT und Windows 2000 haben momentan praktisch eine Begrenzung auf 20 Zeichen. • Bei der Verwendung von DCE-Authentifizierung sind Benutzernamen auf 8 Zeichen begrenzt. • Wenn weder DCE- noch Client-Authentifizierung verwendet wird, werden Benutzernamen mit mehr als 8 Zeichen auf nicht-Windows 32-Bit-Clients, die mit Windows NT oder Windows 2000 verbunden sind, unterstützt, wenn Benutzername und Kennwort explizit angegeben werden. • Namen und IDs dürfen nicht: <ul style="list-style-type: none"> – USERS, ADMINS, GUESTS, PUBLIC, LOCAL oder wie eines der für SQL reservierten Wörter lauten, die im Handbuch <i>SQL Reference</i> aufgeführt sind. – mit IBM, SQL oder SYS beginnen. – diakritische Zeichen enthalten. <p>In auf UNIX basierenden Systemen können Gruppen und Benutzer denselben Namen haben. Bei der Anweisung GRANT müssen Sie angeben, ob eine Gruppe oder ein Benutzer gemeint ist. Bei der Anweisung REVOKE hängt die Angabe eines Benutzers oder einer Gruppe davon ab, ob für GRANTEE mehrere Zeilen in den Tabellen des Berechtigungskatalogs mit verschiedenen Werten für GRANTEETYPE enthalten sind.</p> <p>Unter Windows NT dürfen lokale und globale Gruppen und Benutzer nicht dieselben Namen haben.</p> <p>Unter OS/2 dürfen Gruppen und Benutzer nicht dieselben Namen haben.</p> <p>Anmerkungen:</p> <ol style="list-style-type: none"> 1. Bei einigen Betriebssystemen sind Benutzer-IDs und Kennwörter mit Unterscheidung von Groß-/Kleinschreibung möglich. Prüfen Sie die Dokumentation zu Ihrem Betriebssystem, um festzustellen, ob dies auf Ihr Betriebssystem zutrifft. 2. Die Berechtigungs-ID, die nach der erfolgreichen Ausführung von CONNECT oder ATTACH zurückgegeben wird, wird nach 8 Zeichen abgeschnitten. Eine Auslassung (...) wird der Berechtigungs-ID angehängt und die Felder SQLWARN enthalten Warnungen, die auf das Abschneiden hinweisen. Weitere Informationen finden Sie im Abschnitt zur Anweisung CONNECT in <i>SQL Reference</i>.

Zusätzliche Informationen zu Kennwörtern

Sie müssen möglicherweise Aufgaben zur Kennwortverwaltung ausführen. Da solche Aufgaben auf dem Server auszuführen sind und viele Benutzer mit der Arbeit in der Server-Umgebung nicht vertraut sind oder sich dabei nicht wohl fühlen, kann diese Tätigkeit eine große Herausforderung bedeuten. DB2 UDB stellt eine Möglichkeit zur Aktualisierung und Überprüfung von Kennwörtern ohne Arbeiten auf dem Server zur Verfügung. Zum Beispiel unterstützt DB2 für OS/390 Version 5 diese Methode, um das Kennwort eines Benutzers zu ändern. Wenn eine Fehlermeldung SQL1404N „Kennwort abgelaufen“ empfangen wird, ändern Sie mit der Anweisung CONNECT das Kennwort auf folgende Weise:

```
CONNECT TO <datenbank> USER <benutzer-id> USING <kennwort>
NEW <neues_kennwort> CONFIRM <neues_kennwort>
```

Das Dialogfenster „Kennwort ändern“ der DB2-Client-Konfiguration - Unterstützung kann ebenfalls zur Änderung des Kennworts verwendet werden. Weitere Informationen zu diesen Methoden zur Änderung des Kennworts finden Sie im Handbuch *SQL Reference* und in der Online-Hilfefunktion der DB2-Client-Konfiguration - Unterstützung.

Tabelle 18. Namenskonventionen für Objekte in zusammengeschlossenen Datenbanken

Objekte	Richtlinien
<ul style="list-style-type: none"> • Funktionszuordnungen • Indexspezifikationen • Kurznamen • Server • Typzuordnungen • Benutzerzuordnungen • Oberflächen 	<ul style="list-style-type: none"> • Kurznamen, Namen für Zuordnungen, Indexspezifikationen, Server und Oberflächen dürfen nicht länger als 128 Byte sein. • Optionen für Server und Kurznamen und Optionseinstellungen sind auf 255 Byte begrenzt. • Namen für Objekte in zusammengeschlossenen Datenbanken können folgende Zeichen enthalten: <ul style="list-style-type: none"> – Gültige diakritische Zeichen (zum Beispiel ö) – Ein Mehrbytezeichen außer Mehrbyteleerzeichen (für Mehrbyteumgebungen)

Verwendung begrenzter Bezeichner in Objektnamen

Schlüsselwörter dürfen verwendet werden. Bei Verwendung eines Schlüsselworts in einem Kontext, in dem es auch als SQL-Schlüsselwort interpretiert werden könnte, muss es als begrenzter Bezeichner angegeben werden.

Mit Hilfe der begrenzten Bezeichner ist es möglich, ein Objekt zu erstellen, dessen Name gegen diese Namenskonventionen verstößt. Jedoch können bei nachfolgender Verwendung eines solchen Objekts Fehler auftreten. Wenn Sie zum Beispiel eine Spalte mit einem Namen erstellen, in dem ein Pluszeichen + oder ein Minuszeichen – vorkommt, und Sie in der Folge diese Spalte in einem Index verwenden, treten Probleme auf, wenn Sie versuchen, die Tabelle zu reorganisieren. Weitere Informationen zu begrenzten Bezeichnern finden Sie im Abschnitt "SQL Identifiers" im Handbuch *SQL Reference*.

Beibehaltung von Werten in Groß-/Kleinschreibung in einem System zusammengegeschlossener Datenbanken

In verteilten Anforderungen müssen Sie manchmal Kennungen und Kennwörter angeben, deren Groß-/Kleinschreibung auf der Datenquelle unterschieden wird. Beachten Sie die folgenden Richtlinien, um sicherzustellen, dass die Schreibweise beim Übergeben an die Datenquelle korrekt ist:

- Geben Sie die IDs und Kennwörter in der erforderlichen Schreibweise (Groß-/Kleinschreibung) an, und setzen Sie sie in doppelte Anführungszeichen.
- Wenn Sie eine Benutzer-ID angeben, setzen Sie die Serveroption *fold_id* für die Datenquelle auf "n" („Nein, Groß-/Kleinschreibung nicht ändern“). Wenn Sie ein Kennwort angeben, setzen Sie die Serveroption *fold_pw* für die Datenquelle auf "n".

Es gibt eine Alternative für Benutzer-IDs und Kennwörter. Wenn bei einer Datenquelle eine Benutzer-ID in Kleinbuchstaben angegeben werden muss, können Sie sie in beliebiger Schreibweise angeben und die Serveroption *fold_id* auf "l" („Diese ID in Kleinbuchstaben an die Datenquelle senden“) setzen. Wenn bei der Datenquelle die Benutzer-ID in Großbuchstaben angegeben werden muss, können Sie sie in beliebiger Schreibweise angeben und die Serveroption *fold_id* auf "u" („Diese ID in Großbuchstaben an die Datenquelle senden“) setzen. Auf dieselbe Art und Weise können Sie, wenn bei einer Datenquelle ein Kennwort in Klein- oder in Großbuchstaben angegeben werden muss, diese Anforderung erfüllen, indem Sie die Serveroption *fold_pw* auf "l" bzw. "u" setzen.

Weitere Informationen zu Serveroptionen finden Sie in „Vereinfachen der Datenquellendefinitionen und der Authentifizierungsverarbeitung durch Serveroptionen“ auf Seite 178.

- Wenn Sie eine Kennung oder ein Kennwort mit zu beachtender Groß-/Kleinschreibung in einer Eingabeaufforderung des Betriebssystems in doppelte Anführungszeichen setzen, müssen Sie sicherstellen, dass das System die doppelten Anführungszeichen syntaktisch korrekt analysiert. Gehen Sie dazu wie folgt vor:
 - Bei einem auf UNIX basierenden Betriebssystem schließen Sie die Anweisung in einfache Anführungszeichen ein.
 - Beim Betriebssystem Windows NT stellen Sie jedem Anführungszeichen einen umgekehrten Schrägstrich voran.

Viele begrenzte Bezeichner in SB2-Datenquellen unterscheiden zum Beispiel Groß-/Kleinschreibung. Angenommen, Sie wollen einen Kurznamen KURZ1 für eine Sicht von DB2 für CS, "mein_schema"."wöch_geh", in einer Datenquelle namens NORBASE erstellen.

In die Eingabeaufforderung eines auf UNIX basierenden Systems würden Sie Folgendes eingeben:

```
db2 'create nickname kurz1 for norbase."mein_schema"."wöch_geh"'
```

In eine Eingabeaufforderung von Windows NT würden Sie Folgendes eingeben:

```
db2 create nickname kurz1 for norbase.\"mein_schema\".\"wöch_geh\"
```

Wenn Sie die Anweisung an der DB2-Eingabeaufforderung (Dialogmodus) oder in einem Anwendungsprogramm eingeben, sind die einfachen Anführungszeichen oder die umgekehrten Schrägstriche nicht notwendig. In die DB2-Eingabeaufforderung eines auf UNIX basierenden Systems oder eines Windows NT-Systems würden Sie Folgendes eingeben:

```
create nickname kurz1 for norbase."mein_schema"."wöch_geh"
```

Anhang B. Verwenden der DCE-Verzeichnisservices

DCE stellt den Cell Directory Service (CDS) und den Global Directory Service (GDS) zur Verfügung. Weitere Informationen über DCE-Konzepte und diese Services finden Sie im Handbuch *Introduction to OSF DCE*. Die DB2-Funktion für die DCE-Verzeichnisservices unterstützt nur CDS. Mit dieser Unterstützung muss der Benutzer nicht jede Datenbank, jeden Knoten und jede DCS-Datenbank auf jedem einzelnen Client erstellen. Alle diese Informationen werden von DCE Cell Directory Services zentral zur Verfügung gestellt.

In den folgenden Abschnitten wird beschrieben, wie Sie über die DCE-Verzeichnisservices eine Datenbank einrichten und auf sie zugreifen:

- Erstellen von Verzeichnisobjekten
- Attribute der einzelnen Objektklassen
- Sicherheit bei den Verzeichnisservices
- Konfigurationsparameter und Registrierungswerte
- Befehle CATALOG und ATTACH und die Anweisung CONNECT
- Herstellen der Verbindung eines Clients mit einer Datenbank
- Durchsuchen von Verzeichnissen
- Temporäres Außerkraftsetzen von Informationen in DCE-Verzeichnissen
- Aufgaben im Zusammenhang mit den Verzeichnisservices
- Einschränkungen bei den Verzeichnisservices

Die DCE-Verzeichnisservices werden möglicherweise nicht von allen DB2-Clients unterstützt. Wenn die DCE-Verzeichnisservices für einen DB2-Client unterstützt werden, finden Sie in Ihrem Handbuch *Einstieg* zusätzliche Informationen hierzu.

Erstellen von Verzeichnisobjekten

Es gibt drei Arten von Verzeichnisobjekten, die der Datenbankadministrator erstellen muss:

- „Datenbankobjekte“ auf Seite 378
- „Querverweisobjekte für Datenbanken“ auf Seite 380
- „Objekte für Leitweginformationen“ auf Seite 381

Jedes Objekt enthält Attribute. Eine vollständige Beschreibung der Attribute finden Sie in „Attribute der einzelnen Objektklassen“ auf Seite 384.

Bevor der Datenbankadministrator die Objekte erstellen kann, muss der DCE-Administrator einer CDS-Tabelle Datenbankinformationen hinzufügen und dem Datenbankadministrator Zugriffsrechte zum Erstellen (CREATE) erteilen. Weitere Einzelheiten hierzu enthält der Abschnitt „Aufgaben des DCE-Administrators“ auf Seite 403.

Datenbankobjekte

Für jede Zieldatenbank wird ein Datenbankobjekt benötigt. Das Objekt hat einen Namen, der den Zellennamen enthält, der mit dem Verzeichnisnamen und dem Namen der Datenbank verknüpft ist, zum Beispiel:

```
./.../zellename/verzname1/verzname2/OBJNAME
```

Anmerkung: Folgende Regeln werden für den Namen der Datenbank empfohlen. Der Name sollte höchstens acht Zeichen lang sein, und alle Zeichen sollten Großbuchstaben sein. Wenn der Name in Groß-/Kleinschreibung angegeben wurde oder länger als acht Zeichen ist, müssen Sie über den Befehl CATALOG GLOBAL DATABASE einen Aliasnamen zuweisen. Einzelheiten zu diesem Befehl finden Sie in „Der Befehl CATALOG GLOBAL DATABASE“ auf Seite 394.

Es folgt ein Beispiel eines Datenbankobjekts. Das im DCE-Verzeichnis gespeicherte Objekt enthält andere Informationen wie z. B. eine Zeitmarke. Der Buchstabe links neben jedem Attribut gibt an, ob das Attribut erforderlich (-R), wahlfrei (-O) oder ein Kommentar (-C) ist.

```
Object name:          /.../CELL_TORONTO/subsys/database/AIXDB1
R DB_Object_Type:      D
C DB_Product_Name:     DB2_for_AIX
C DB_Product_Release:  V7R1M000
R DB_Native_Database_Name: AIXDBASE
R DB_Database_Protocol: DB2RA
R DB_Authentication:   CLIENT
O DB_Communication_Protocol:
O DB_Database_Locator_Name: /.../CELL_TORONTO/subsys/database/AIX_INST
C DB_Comment:         Test_database_on_AIX
```

Wenn die Datenbank eine von vielen Datenbanken ist, die einem Exemplar des Datenbankmanagers zugeordnet ist, sollte das Datenbankobjekt den Namen eines Querverweisobjekts für Datenbanken (DB_Database_Locator_Name) enthalten, und das Übertragungsprotokoll (DB_Communication_Protocol) sollte leer sein. Der Name des Querverweisobjekts für Datenbanken ist der vollständig qualifizierte Name des Exemplars des Datenbankmanagers bzw. von DB2 Connect.

Es folgt ein Beispiel für die DCE-Befehle zum Erstellen des Objekts. Bevor Objekte erstellt werden können, muss der DCE-Administrator die Schritte ausführen, die im Abschnitt „Aufgaben des DCE-Administrators“ auf Seite 403 beschrieben sind.

Zunächst müssen Sie Folgendes in eine Datei mit dem Namen *cdscp.inp* eingeben:

```
create object ../subsys/database/AIXDB1

add object ../subsys/database/AIXDB1 DB_Object_Type           = D
add object ../subsys/database/AIXDB1 DB_Product_Name         = DB2_for_AIX
add object ../subsys/database/AIXDB1 DB_Product_Release      = V7R1M000
add object ../subsys/database/AIXDB1 DB_Native_Database_Name = AIXDBASE
add object ../subsys/database/AIXDB1 DB_Database_Protocol    = DB2RA
add object ../subsys/database/AIXDB1 DB_Authentication       = CLIENT
add object ../subsys/database/AIXDB1 DB_Database_Locator_Name = /...
/CELL_TORONTO/subsys/database/AIX_INST
add object ../subsys/database/AIXDB1 DB_Comment              = Test_database_on_AIX
```

Anschließend müssen Sie folgenden Befehl ausführen:

- `dcelogin principal` kennwort (unter OS/2); bzw.,
- `dce_login principal` kennwort (unter UNIX und unter Windows Betriebssystemen).

Diesem Befehl sollte folgen:

- `cdscp < cdscp.inp`

Das Objekt kann mit folgendem Befehl angezeigt werden:

```
cdscp show object ../subsys/database/AIXDB1
```

Wenn die Datenbank die einzige einem Exemplar des Datenbankmanagers zugeordnete Datenbank ist, sollte das Datenbankobjekt Werte für das Attribut für das Übertragungsprotokoll (`DB_Communication_Protocol`) enthalten, und der Name des Querverweisobjekts für Datenbanken (`DB_Database_Locator_Name`) sollte leer sein. Beispiel:

```
Object name:                /.../CELL_TORONTO/subsys/database/MVSDDB
R DB_Object type:           D
C DB_Product_Name:         DB2_for_MVS
C DB_Product_Release:      V7R1M00
R DB_Native_Database_Name: MVSDBASE
R DB_Database_Protocol:    DRDA
R DB_Authentication:       SERVER
O DB_Communication_Protocol: APPC;NET1;TARGETLU1;DB2DRDA;MODE1;PROGRAM
O DB_Database_Locator_Name:
C DB_Comment:              Test_database_on_MVS
```

Querverweisobjekte für Datenbanken

Querverweisobjekte für Datenbanken (Database Locator Objects) enthalten die Einzelheiten über alle Übertragungsprotokolle, die von einem DBMS-Exemplar (Database Management System, Datenbankverwaltungssystem) oder einem Exemplar von DB2 Connect verwendet werden. Für die folgenden Exemplare ist jeweils ein Querverweisobjekt für Datenbanken erforderlich:

- Jedes Exemplar sowohl mit Datenbankverwaltungssystem als auch mit DB2 Connect.
- Jedes DBMS-Exemplar (Database Management System, Datenbankverwaltungssystem), dem mehr als eine Datenbank, jedoch kein DB2 Connect zugeordnet ist.
- Jedes Exemplar von DB2 Connect ohne zugeordnetes Datenbankverwaltungssystem.

Das Objekt hat einen Namen, der den Zellennamen enthält, der mit dem Verzeichnisnamen und einem einteiligen Namen des Datenbankexemplars verknüpft ist, z. B.:

```
./.../zellename/verzname1/verzname2/AIX_INST
```

Anmerkung: Wenn das Exemplar als Ziel einer Anweisung ATTACH verwendet wird, darf der einteilige Name nicht länger als acht Zeichen sein und muss in Großbuchstaben angegeben werden.

Es folgt ein Beispiel eines Querverweisobjekts für Datenbanken. Das im DCE-Verzeichnis gespeicherte Objekt enthält andere Informationen wie z. B. eine Zeitmarke. Der Buchstabe links neben jedem Attribut gibt an, ob das Attribut erforderlich (- R), wahlfrei (- O) oder ein Kommentar (- C) ist.

```
Object name:          ./.../CELL_TORONTO/subsys/database/AIX_INST
R DB_Object_Type:     L
C DB_Product_Name:    DB2_for_AIX
C DB_Product_Release: V7R1M00
R DB_Communication_Protocol: TCPIP;HOSTNAME1;1234
R DB_Communication_Protocol: APPC;NET1;TARGETLU1;TPN1;MODE;PROGRAM
C DB_Comment:         Test_instance_on_AIX
```

Wenn ein Attribut sowohl im Datenbankobjekt als auch im Querverweisobjekt für Datenbanken definiert ist, wird der Wert im Datenbankobjekt verwendet.

Es folgt ein Beispiel für die DCE-Befehle zum Erstellen des Objekts. Bevor Objekte erstellt werden können, muss der DCE-Administrator die Schritte ausführen, die im Abschnitt „Aufgaben des DCE-Administrators“ auf Seite 403 beschrieben sind.

Zunächst müssen Sie Folgendes in eine Datei mit dem Namen *cdscp.inp* eingeben:

```
create object /./subsys/database/AIX_INST

add object /./subsys/database/AIX_INST DB_Object_Type           = L
add object /./subsys/database/AIX_INST DB_Product_Name         = DB2_for_AIX
add object /./subsys/database/AIX_INST DB_Product_Release     = V7R1M00
add object /./subsys/database/AIX_INST DB_Communication_Protocol = TCPIP;
HOSTNAME1;1234
add object /./subsys/database/AIX_INST DB_Communication_Protocol = APPC;NET1;
TARGETLU;TPN1;MODE;PROGRAM
add object /./subsys/database/AIX_INST DB_Comment              = Test_instance_on_AIX
```

Anschließend müssen Sie folgenden Befehl ausführen:

- dcelogin principal kennwort (unter OS/2); bzw.,
- dce_login principal kennwort (unter UNIX und unter Windows Betriebssystemen).

Diesem Befehl sollte folgen:

- cdscp < cdscp.inp

Das Objekt kann mit folgendem Befehl angezeigt werden:

```
cdscp show object /./subsys/database/AIX_INST
```

Objekte für Leitweginformationen

Objekte für Leitweginformationen werden für den Host-Zugriff benötigt. Wenn das von einem Client und das von der Zieldatenbank verwendete Datenbankprotokoll nicht übereinstimmen, teilt das Objekt für Leitweginformationen dem Client mit, welches Exemplar von DB2 Connect verwendet werden soll. Für jede Zieldatenbank existieren Attribute, die die verfügbaren Datenbankprotokolle und den Namen des Querverweisobjekts für Datenbanken für das Exemplar von DB2 Connect beinhalten. Das Objekt hat einen Namen, der den Zellennamen enthält, der mit dem Verzeichnisnamen und einem eindeutigen einteiligen Namen verknüpft ist, zum Beispiel:

```
/.../zellename/verzname1/verzname2/ROUTE1
```

Es folgt ein Beispiel eines Objekts für Leitweginformationen. Das im DCE-Verzeichnis gespeicherte Objekt enthält andere Informationen wie z. B. eine Zeitmarke. Der Buchstabe links neben jedem Attribut gibt an, ob das Attribut und jedes Token innerhalb eines Attributs erforderlich (- R), wahlfrei (- O) oder ein Kommentar (- C) ist.

Client_group_1 entspricht Client_1, Client_2 und Client_3 in Abb. 8 auf Seite 396.

```
Object name:      /.../CELL_TORONTO/subsys/database/ROUTE1
R DB_Object_Type: R
C DB_Comment:     Routing_for_client_group_1

R DB_Target_Database_Info
R Database name   = /.../CELL_TORONTO/subsys/database/MVSDB
R Outbound protocol from router = DRDA
R Inbound protocol to router   = DB2RA
R Authenticate at gateway      = 1
O Parameter string              = NOMAP,D,INTERRUPT_ENABLED
R DB_Database_Locator_Name     = /.../CELL_TORONTO/subsys/database/GW_INST

R DB_Target_Database_Info
R Database name   = *OTHERDBS
R Outbound protocol from router = DRDA
R Inbound protocol to router   = DB2RA
R Authenticate at gateway      = 0
O Parameter string              =
R DB_Database_Locator_Name     = /.../CELL_TORONTO/subsys/database/OTH_INST
```

Der Datenbankname *OTHERDBS ist ein Sonderwert und gibt einen allgemeinen Router an, der dem Zugriff auf Zieldatenbanken dient, die nicht explizit im Objekt für Leitweginformationen definiert sind.

Es folgt ein Beispiel für die DCE-Befehle zum Erstellen des Objekts. Der umgekehrte Schrägstrich (\) dient als Fortsetzungszeichen.

Bevor Objekte erstellt werden können, muss der DCE-Administrator die Schritte ausführen, die im Abschnitt „Aufgaben des DCE-Administrators“ auf Seite 403 beschrieben sind.

Zunächst müssen Sie Folgendes in eine Datei mit dem Namen *cdscp.inp* eingeben:

```
create object ././subsys/database/ROUTE1

add object ././subsys/database/ROUTE1 DB_Object_Type = R
add object ././subsys/database/ROUTE1 DB_Comment     = Routing_for_client_group_1
add object ././subsys/database/ROUTE1 DB_Target_Database_Info = \
    /.../CELL_TORONTO/subsys/database/MVSDB;\
drda;db2ra;1;NOMAP,D,INTERRUPT_ENABLE;\
    /.../CELL_TORONTO/subsys/database/GW_INST
add object ././subsys/database/ROUTE1 DB_Target_Database_Info = \
    *OTHERDBS;drda;db2ra;0;;\
    /.../CELL_TORONTO/subsys/database/OTH_INST
```

Anschließend müssen Sie folgenden Befehl ausführen:

- `dcelogin principal kennwort` (unter OS/2); bzw.,
- `dce_login principal kennwort` (unter UNIX und unter Windows Betriebssystemen).

Diesem Befehl sollte folgen:

- `cdscp < cdscp.inp`

Das Objekt kann mit folgendem Befehl angezeigt werden:

```
cdscp show object /./subsys/database/ROUTE1
```

Weitere Informationen zu den DCE-Befehlen können Sie folgenden OSF-DCE-Veröffentlichungen entnehmen:

- *OSF DCE Administration Guide*
- *OSF DCE Administration Reference*

Attribute der einzelnen Objektklassen

In der DCE-Umgebung wird jedes Objekt und jedes Objektattribut durch eine Objekt-ID (OID) gekennzeichnet. Jede OID wird aus einer Hierarchie von Zuordnungsberechtigungen ermittelt, wobei als höchste Berechtigung die International Organization for Standardization (ISO) angenommen wird.

Tabelle 19 zeigt die Attribute für jede Objektklasse und Tabelle 20 die zugehörigen Attribute.

Tabelle 19. Objektattributklassen

Objektklasse	Objekt-ID (OID)	Erforderliche Attribute	Wahlfreie Attribute
(DB) Database_Object	1.3.18.0.2.6.12	DAU, DOT, DDP, DNN	DCO, DPN, DRL, DLN, DCP, DPR
(DL) Database_Locator_Object	1.3.18.0.2.6.13	DOT, DCP	DCO, DPN, DRL
(RI) Routing_Information_Object	1.3.18.0.2.6.14	DOT, DTI	DCO, DPN, DRL

Tabelle 20. Objektklassenattribute

Attributname	OID	Minimale Länge	Maximale Länge	Syntax
(DAU) DB_Authentication	1.3.18.0.2.4.39	1	1024	Char
(DCO) DB_Comment	1.3.18.0.2.4.30	1	1024	Char
(DCP) DB_Communication_Protocol	1.3.18.0.2.4.31	1	1024	Char
(DDP) DB_Database_Protocol	1.3.18.0.2.4.32	1	1024	Char
(DLN) DB_Database_Locator_Name	1.3.18.0.2.4.33	1	1024	Char
(DNN) DB_Native_Database_Name	1.3.18.0.2.4.34	1	1024	Char
(DOT) DB_Object_Type	1.3.18.0.2.4.35	1	1	Char
(DPN) DB_Product_Name	1.3.18.0.2.4.36	1	1024	Char
(DRL) DB_Product_Release	1.3.18.0.2.4.37	1	1024	Char
(DTI) DB_Target_Database_Info	1.3.18.0.2.4.38	1	1024	Char
(DPR) DB_Principal	1.3.18.0.2.4.63	1	1024	Char
Anmerkung: Für DCP, DDP und DTI sind mehrere Werte zulässig. Für die anderen Attribute ist jeweils nur ein einziger Wert zulässig.				

Einzelangaben zu den Attributen

Im folgenden Abschnitt werden die einzelnen Attribute beschrieben.

Anmerkung: Von den DCE-Verzeichnisservices wird nicht überprüft, ob die Einträge für DB2 gültig sind. Sie müssen selbst dafür sorgen, dass Sie die benötigten Attribute eingeben und dass gültige Werte angegeben werden.

DB_Authentication (DAU)

Die für das Objekt benötigte Methode der Authentifizierung. Dieses Attribut ist für das Datenbankobjekt eines DB2-Servers erforderlich. Als Wert muss CLIENT, SERVER oder DCE angegeben werden.

DB_Principal (DPR)

Wenn für die Authentifizierungsmethode der Wert „DCE“ definiert ist, geben Sie den DCE-Principal in dieses Attribut ein.

DB_Comment (DCO)

Dieses Attribut dient nur zu Dokumentationszwecken.

DB_Communication_Protocol (DCP)

Ein aus mehreren Werten bestehendes Attribut, bei dem jeder Wert aus Token besteht, die das unterstützte Netzprotokoll beschreiben. Beispiele für die Netzprotokolle sind TCP/IP, APPC, IPX/SPX und Net-BIOS. Die Token werden durch ein Semikolon getrennt. Zwischen den Token dürfen keine Leerzeichen angegeben werden.

- Die Token für TCP/IP sind:
 1. tcpip
 2. Host-Name des Zielknotens
 3. Vom Objekt zum Empfang eingehender TCP/IP-Verbindungsanforderungen verwendete Anschlussnummer
 4. (Wahlfrei) Für die Sicherheit kann entweder NONE oder SOCKS angegeben werden.

Zum Beispiel: tcpip;HOSTNAME;1234

- Die Token für APPC sind:
 1. appc
 2. Netz-ID des Ziels, zu dem das Objekt gehört
 3. Name der LU, auf der das Ziel zu finden ist
 4. Transaktionsprogrammname (TPN) für das Objekt in der LU (unter DB2 für MVS/ESA muss DB2DRDA als TPN verwendet werden.)
 5. Modusname
 6. Vom Ziel verwendete Sicherheitseinstufung. Die Werte sind:

- NONE
- PROGRAM
- SAME

Zum Beispiel: `appc;NETID;TARGETLU;TPNAME;MODE;PROGRAM`

Anmerkung: Bei APPC muss der Client seinen lokalen Steuerpunkt (Control Point, CP) als LU-Namen verwenden.

- (Nur OS/2 und unterstützte Versionen des Betriebssystems Windows.) Die Tokens für IPX/SPX sind:
 1. `ipxspx`
 2. Name des Dateiservers
 3. Name des Objekts

Zum Beispiel: `ipxspx;SVR_NAME;OBJ_NAME`

- (Nur OS/2 und unterstützte Versionen des Betriebssystems Windows.) Die Tokens für NetBIOS sind:
 1. `netbios`
 2. Knotenname des Servers

Beispiel: `netbios;SVR_NAME`, wobei die Client-Adaptornummer entweder dem Registrierungswert `db2clientadpt` oder dem Konfigurationsparameter `dft_client_adpt` des Datenbankmanagers entnommen werden kann.

- (Nur unterstützte Versionen des Betriebssystems Windows.) Die Tokens für NPIPE sind:
 1. `NPIPE`
 2. Computername des Servers
 3. Exemplarname des Servers

Zum Beispiel: `npipe;computername;instance`

DB_Database_Protocol (DDP)

Das Datenbankprotokoll, das von der Zieldatenbank unterstützt wird oder die Datenbankprotokolle, die von der Zieldatenbank unterstützt werden. Beispiele für diese Werte sind DB2RA und DRDA. Nachfolgend sind die `cdscp`-Befehle zum Hinzufügen zweier Protokolle aufgeführt.

```
add object /./subsys/database/AIXDB1 DB_Database_Protocol db2ra
add object /./subsys/database/AIXDB1 DB_Database_Protocol drda
```

DB_Database_Locator_Name (DLN)

Der DCE-Name des Querverweisobjekts für Datenbanken. Im Datenbankobjekt bezeichnet dieser Name das DBMS-Exemplar. Im Objekt für Leitweginformationen bezeichnet dieser Name das Exemplar von DB2 Connect.

Beispiel: /.../CELL_TORONTO/subsys/database/AIX_INST

DB_Native_Database_Name (DNN)

Der Datenbankname oder Aliasname, unter dem die Datenbank innerhalb des Exemplars, das die Datenbank enthält, bekannt ist. Diesen Namen verwendet eine lokale Anwendung im Exemplar, um eine Verbindung mit der Datenbank herzustellen.

Der Name ist bei einer Datenbank unter DB2 für Universal Database max. acht Zeichen lang. Bei anderen Datenbanken kann die Länge des Namens unterschiedlich sein. Zum Beispiel kann sie bei Datenbanken unter DB2 für MVS/ESA bis zu 18 Zeichen lang sein.

DB_Object_Type (DOT)

Der Objekttyp. Dieses Attribut ist für alle Objekte erforderlich und kann einen der folgenden Werte aufweisen:

- D** Datenbankobjekt
- L** Querverweisobjekt für Datenbanken (Database Locator Object)
- R** Objekt für Leitweginformationen (Routing Information Object)

DB_Product_Name (DPN)

Die Produktkennung. Dieses Attribut dient nur zu Dokumentationszwecken.

DB_Product_Release (DRL)

Der Release-Stand des Produkts. Dieses Attribut dient nur zu Dokumentationszwecken.

DB_Target_Database_Info (DTI)

Ein aus mehreren Werten bestehendes Attribut, bei dem jeder Wert aus einer festen Anzahl von Token besteht, die durch ein Semikolon voneinander getrennt sind. Zwischen den Token dürfen keine Leerzeichen angegeben werden. Die Token müssen in folgender Reihenfolge angegeben werden:

1. Datenbankname. Der DCE-Name einer Zieldatenbank, für die der Leitwegservice zur Verfügung gestellt wird. Der Wert *OTHERDBS gibt ein Standard-Gateway für Zieldatenbanken an, die nicht explizit im Objekt für Leitweginformationen definiert sind.

2. Protokoll für vom Router abgehende Verbindungen. Das Datenbankprotokoll, das von der Zieldatenbank verwendet wird, oder das Datenbankprotokoll, das vom weiterleitenden Exemplar von DB2 Connect zur Kommunikation mit der Zieldatenbank verwendet wird. Beispielsweise kann hier DRDA angegeben werden.
3. Protokoll für zum Router hingehende Verbindungen. Das Datenbankprotokoll, das vom Objekt für das weiterleitende Exemplar von DB2 Connect akzeptiert wird. Beispielsweise kann hier DB2RA angegeben werden.
4. Authentifizierung am Gateway. Gültige Werte sind 0 oder 1. Weitere Einzelheiten hierzu enthält Tabelle 21 auf Seite 390.
5. Parameterzeichenfolge, die für den DB2 Connect-Gateway spezifische Informationen enthält. Die Zeichenfolge enthält Token, die in der nachfolgend dargestellten Reihenfolge angegeben werden müssen. Die Token werden durch Kommas getrennt. Für nicht angegebene Token werden die betreffenden Standardwerte verwendet.
 - Name der Zuordnungsdatei. Der vollständig qualifizierte Name der SQLCODE-Zuordnungsdatei, der die standardmäßige SQLCODE-Zuordnung außer Kraft setzt. Wenn die SQLCODE-Zuordnung ausgeschaltet werden soll, muss NOMAP angegeben werden.
 - D. Die Anwendung trennt die Verbindung zur DRDA-Server-Datenbank, wenn bestimmte SQLCODE-Werte zurückgegeben werden. Im *DB2 Connect Benutzerhandbuch* finden Sie Einzelheiten zu den SQLCODEs.
 - INTERRUPT_ENABLED. DB2 Connect trennt die Verbindung und macht die Arbeitseinheit rückgängig, wenn ein Client eine Unterbrechung absetzt, während er mit dem DRDA-Server verbunden ist.

Es folgen einige Beispiele:

```
NOMAP
/u/username/sqllib/map/dcs1new.map,D
/u/username/sqllib/map/dcs1new.map,D,INTERRUPT_ENABLED
```

Wenn Standardwerte verwendet werden, muss ein Komma angegeben werden, damit die Ordnung der Token erhalten bleibt. Beispiel:

```
,D
```

oder

```
,,INTERRUPT_ENABLED
```

Im *DB2 Connect Benutzerhandbuch* finden Sie Einzelheiten zur Parameterzeichenfolge.

6. Der DCE-Name des Exemplars von DB2 Connect, das den Leitwegservice zur Verfügung stellt.

Es folgt ein Beispiel einer Angabe für `DB_Target_Database_Info`:

```
.../CELL_TORONTO/subsys/database/MVSDDB;\
drda;db2ra;0;;\
.../CELL_TORONTO/subsys/database/GW_INST
```

Anmerkung: In obigem Beispiel dient der umgekehrte Schrägstrich (\) als Zeilenfortsetzungszeichen.

Sicherheit bei den Verzeichnisservices

Wenn die DCE-Verzeichnisservices in einer Umgebung ohne einen DB2 Connect-Gateway verwendet werden, erfolgt die gleiche Authentifizierung, wie sie für andere Clients, die auf Datenbank-Server zugreifen, durchgeführt wird. Weitere Informationen finden Sie in „Auswählen einer Authentifizierungsart für den Server“ auf Seite 261.

Wenn die DCE-Verzeichnisservices in einer Umgebung mit einem DB2 Connect-Gateway verwendet werden, legt der DB2 Connect-Administrator fest, wo Benutzernamen und Kennwörter überprüft werden. Bei DCE-Verzeichnissen muss Folgendes angegeben werden:

- Die Sicherheitseinstufung des Übertragungsprotokolls im Querverweisobjekt für Datenbanken (Database Locator Object), das die DB2 Connect-Workstation repräsentiert. (Wenn ein ferner Client mit einem Gateway unter DB2 Connect Extended Edition über eine APPC-Verbindung verbunden wird, geben Sie die Sicherheitseinstufung NONE im *DCE-Querverweisobjekt* des Gateway an.)
- Die Art der Authentifizierung im Datenbankobjekt
- Die Sicherheitseinstufung des Übertragungsprotokolls im Datenbankobjekt (oder dem entsprechenden Querverweisobjekt)
- Das Token für die Authentifizierung auf dem Gateway im Objekt für Leitweginformationen

Tabelle 21 zeigt die möglichen Kombinationen dieser Werte und den Ort, an dem die Gültigkeitsüberprüfung für jede Kombination bei APPC-Verbindungen ausgeführt wird. Die in dieser Tabelle gezeigten Kombinationen werden von DB2 Connect mit den DCE-Verzeichnisservices unterstützt.

Tabelle 21. Gültige Sicherheitsszenarios mit DCE über APPC-Verbindungen

Fall	Datenbankobjekt des Servers		Leitwegobjekt	Prüfung
	Authentifizierung	Sicherheit	Authentifizierung auf dem Gateway	
1	CLIENT	SAME	0	Ferner Client (oder Workstation unter DB2 Connect)
2	CLIENT	SAME	1	Workstation unter DB2 Connect
3	SERVER	PROGRAM	0	DRDA-Server
4	SERVER	PROGRAM	1	Workstation unter DB2 Connect und DRDA-Server
5	DCE	NONE	NICHT ZUTREFFEND	DCE

Tabelle 22 zeigt die möglichen Kombinationen dieser Werte und den Ort, an dem Gültigkeitsprüfung für jede dieser Kombinationen bei TCP/IP-Verbindungen durchgeführt wird. Die in dieser Tabelle gezeigten Kombinationen werden von DB2 Connect mit den DCE-Verzeichnisservices unterstützt.

Tabelle 22. Gültige Sicherheitsszenarios mit DCE über TCP/IP-Verbindungen

Fall	Authentifizierung	Authentifizierung auf dem Gateway	Prüfung
1	CLIENT	0	Client
2	CLIENT	1	Workstation unter DB2 Connect
3	SERVER	0	DRDA-Server
4	NICHT ZUTREFFEND	NICHT ZUTREFFEND	Keine
5	DCE	NICHT ZUTREFFEND	DCE

Jede Kombination gilt für APPC und TCP/IP und wird im Folgenden eingehender beschrieben:

1. Der Benutzername und das Kennwort werden nur auf dem fernen Client überprüft. (Bei einem lokalen Client werden der Benutzername und das Kennwort nur auf der DB2 Connect-Workstation überprüft.)
Es wird erwartet, dass die Identifikation des Benutzers an der Stelle geprüft wird, an der er sich das erste Mal anmeldet. Die Benutzer-ID wird über das Netzwerk gesendet, aber nicht das Kennwort. Dieser Sicherheitstyp sollte nur verwendet werden, wenn alle Client-Workstations über entsprechende Sicherheitseinrichtungen verfügen.
2. Der Benutzername und das Kennwort werden nur auf der DB2 Connect-Workstation überprüft. Das Kennwort wird über das Netzwerk von dem fernen Client an die DB2 Connect-Workstation gesendet, aber nicht an den DRDA-Server.
3. Der Benutzername und das Kennwort werden nur auf dem DRDA-Server überprüft. Das Kennwort wird über das Netzwerk von dem fernen Client an die DB2 Connect-Workstation und von der DB2 Connect-Workstation an den DRDA-Server gesendet.

4. Der Benutzername und das Kennwort werden sowohl auf der DB2 Connect-Workstation als auch auf dem DRDA-Server überprüft. Das Kennwort wird über das Netzwerk von dem fernen Client an die DB2 Connect-Workstation und von der DB2 Connect-Workstation an den DRDA-Server gesendet.

Da die Gültigkeitsprüfung an zwei Stellen ausgeführt wird, muss der gleiche Satz von Benutzernamen und Kennwörtern sowohl auf der DB2 Connect-Workstation als auch auf dem DRDA-Server gepflegt werden.

5. Ein DCE-Token wird von dem DCE-Sicherheits-Server empfangen.

Anmerkungen:

1. Bei auf AIX basierenden Systemen müssen alle Benutzer mit dem Sicherheitstyp SAME zur AIX-Systemgruppe gehören.
2. Bei auf AIX basierenden Systemen mit fernen Clients muss das Exemplar des Produkts DB2 Connect, das auf der DB2 Connect-Workstation ausgeführt wird, zur AIX-Systemgruppe gehören.
3. Der Zugriff auf einen DRDA-Server wird durch dessen eigene Sicherheitsmechanismen oder Subsysteme gesteuert, z. B. Virtual Telecommunications Access Method (VTAM) und Resource Access Control Facility (RACF). Der Zugriff auf geschützte Datenbankobjekte wird durch die SQL-Anweisungen **GRANT** und **REVOKE** gesteuert.

Konfigurationsparameter und Registrierungswerte

Folgende Konfigurationsparameter werden mit DCE-Verzeichnissen verwendet. Es wird jeweils ein Beispiel für die Werte gegeben. Einzelheiten hierzu finden Sie in „Verteilte Services“ im Kapitel „Konfigurieren von DB2“ des Handbuchs *Systemverwaltung: Optimierung*.

- *dir_obj_name* ist der Datenbankexemplarname, der mit *dir_path_name* verknüpft wird. Wenn der Exemplarname als Ziel des Befehls ATTACH verwendet wird, darf der Name nicht länger als acht Zeichen sein und muss in Großbuchstaben angegeben sein. Beispiel:

AIX_INST

- *dir_type* gibt an, ob die DCE-Verzeichnisservices verwendet werden sollen. Wenn die DCE-Verzeichnisservices aktiviert werden sollen, muss dieser Parameter auf folgenden Wert gesetzt werden:

DCE

Beachten Sie, dass auf Datenbank-Clients, die die Verwendung der DCE-Verzeichnisservices nicht unterstützen, *dir_type* auf NONE gesetzt ist und nicht aktualisiert werden kann.

- *dir_path_name* ist der Verzeichnispfadname, der vom DCE-Administrator angegeben wurde, zum Beispiel:

././subsys/database/

- *route_obj_name* ist ein wahlfreier Parameter, der den DCE-Verzeichnis-services-Namen des Objekts für Leitweginformationen angibt. Der Name kann vollständig qualifiziert sein, zum Beispiel:

././subsys/database/ROUTE1

Es kann auch ein einteiliger Name angegeben werden, der mit *dir_path_name* verknüpft wird. Beispiel:

ROUTE1

- *dft_client_comm* ist ein wahlfreier DCE-Parameter, der das vom Client verwendete Übertragungsprotokoll angibt. Beispiel:

TCPIP

Mit diesem Parameter können auch mehrere Protokolle angegeben werden. Beispiel:

TCPIP,APPC (auf UNIX basierende Plattformen)
 TCPIP,APPC,IPXSPX,NETBIOS (OS/2-Plattformen)
 TCPIP,APPC,IPXSPX,NETBIOS,NPIPE (auf unterstützten Versionen
 des Betriebssystems Windows)

- *dft_client_adpt* ist ein wahlfreier DCE-Parameter, der die Standardadapternummer des Clients für das NetBIOS-Protokoll unter OS/2 und unterstützten Versionen des Betriebssystems Windows angibt. Der gültige Bereich für die Nummern ist 0 bis 15. Wenn dieser Parameter einen nichtnumerischen Wert enthält, nimmt er standardmäßig den Wert Null (0) an. Wenn dieser Parameter einen Wert außerhalb des zulässigen Wertebereichs enthält, nimmt er standardmäßig den Wert Null (0) an.

Die Werte der folgenden Parameter können durch Variablen der Profilregistrierdatenbank außer Kraft gesetzt werden.

Konfigurationsparameter	Variable der Profilregistrierdatenbank
<i>dir_path_name</i>	DB2DIRPATHNAME
<i>route_obj_name</i>	DB2ROUTE
<i>dft_client_comm</i>	DB2CLIENTCOMM
<i>dft_client_adpt</i>	DB2CLIENTADPT

Die Regeln für die Einstellung der Variablen der Profilregistrierdatenbank sind identisch mit denen für den entsprechenden Konfigurationsparameter. Zum Beispiel ist der Wert für DB2CLIENTCOMM wie der für den Parameter *dft_client_comm* eine Zeichenfolge, die aus mehreren Werten bestehen kann, wobei die einzelnen Werte durch Kommas getrennt sind. Beispiel:

```
db2set DB2CLIENTCOMM=TCPIP,APPC
```

Befehle CATALOG und ATTACH und die Anweisung CONNECT

In den folgenden Befehlen müssen DCE-Informationen angegeben werden:

- Der Befehl CATALOG GLOBAL DATABASE
- Anweisung CONNECT
- Der Befehl ATTACH

Der Befehl CATALOG GLOBAL DATABASE

Der Befehl CATALOG GLOBAL DATABASE wird verwendet, wenn Client und Server einen unterschiedlichen Pfadnamen haben oder wenn der Datenbankname mehr als acht Zeichen oder Zeichen in Groß-/Kleinschreibung enthält. Der Datenbankadministrator gibt den DCE-Namen der Datenbank und den Verzeichnistyp DCE an.

Beispiel:

- Wenn die Pfadnamen unterschiedlich sind, z. B. wenn *dir_path_name* = *./.../CELL_TORONTO/subsys/database/*:

```
CATALOG GLOBAL DATABASE
./.../CELL_VANCOUVER/subsys/database/VMDB AS VANVMD
USING DIRECTORY DCE WITH "Kommentarzeichenfolge"
```

- Wenn der Datenbankname mehr als acht Zeichen enthält, wie z. B. der Name DB_LONGNAME:

```
CATALOG GLOBAL DATABASE
./.../CELL_VANCOUVER/subsys/database/DB_LONGNAME AS VANVMD
USING DIRECTORY DCE WITH "Kommentarzeichenfolge"
```

Anweisung CONNECT

Zum Abrufen des richtigen DCE-Verzeichnisobjekts muss dem Client der vollständig qualifizierte DCE-Name der Datenbank oder des DBMS-Exemplars bekannt sein. Im Folgenden werden einige Methoden zum Angeben des Namens in der Anweisung CONNECT beschrieben.

- Geben Sie den Aliasnamen ein, z. B.:
`CONNECT TO VANVMDB`
- Geben Sie den einteiligen Namen ein, z. B.:
`CONNECT TO VMDB`

In diesem Fall muss der am Client angegebene Pfadname mit dem am Server angegebenen Pfadnamen identisch sein. (Der Pfadname wird durch den Konfigurationsparameter *dir_path_name* oder den entsprechenden Registrierungswert definiert.)

Der Befehl ATTACH

Der effektive Pfadname des Clients muss mit dem Pfadnamen des Ziel-DBMS-Exemplars identisch sein.

Wenn der Wert für *dir_path_name* für den Client und den Server gleich ist (z. B., `./.../CELL_TORONTO/subsys/database/`) und der Wert für *dir_obj_name* auf dem Datenbank-Server AIX_INST ist, lautet der Befehl zum Herstellen einer Verbindung zum Exemplar folgendermaßen:

```
ATTACH TO AIX_INST
```

Herstellen der Verbindung eines Clients mit einer Datenbank

Abb. 8 auf Seite 396 zeigt eine Beispielkonfiguration eines Datenbanknetzwerks mit zwei DCE-Zellen (DCE Cells). `./.../CELL_TORONTO` und `./.../CELL_VANCOUVER` sind die Namen der Zellen. (Jede dieser Zellen enthält ein Verzeichnis mit dem Namen `././subsys/database/`; diese Verzeichnisse sind zwar in der Abbildung nicht dargestellt, jedoch wird in anderen Beispielen auf sie zurückgegriffen.)

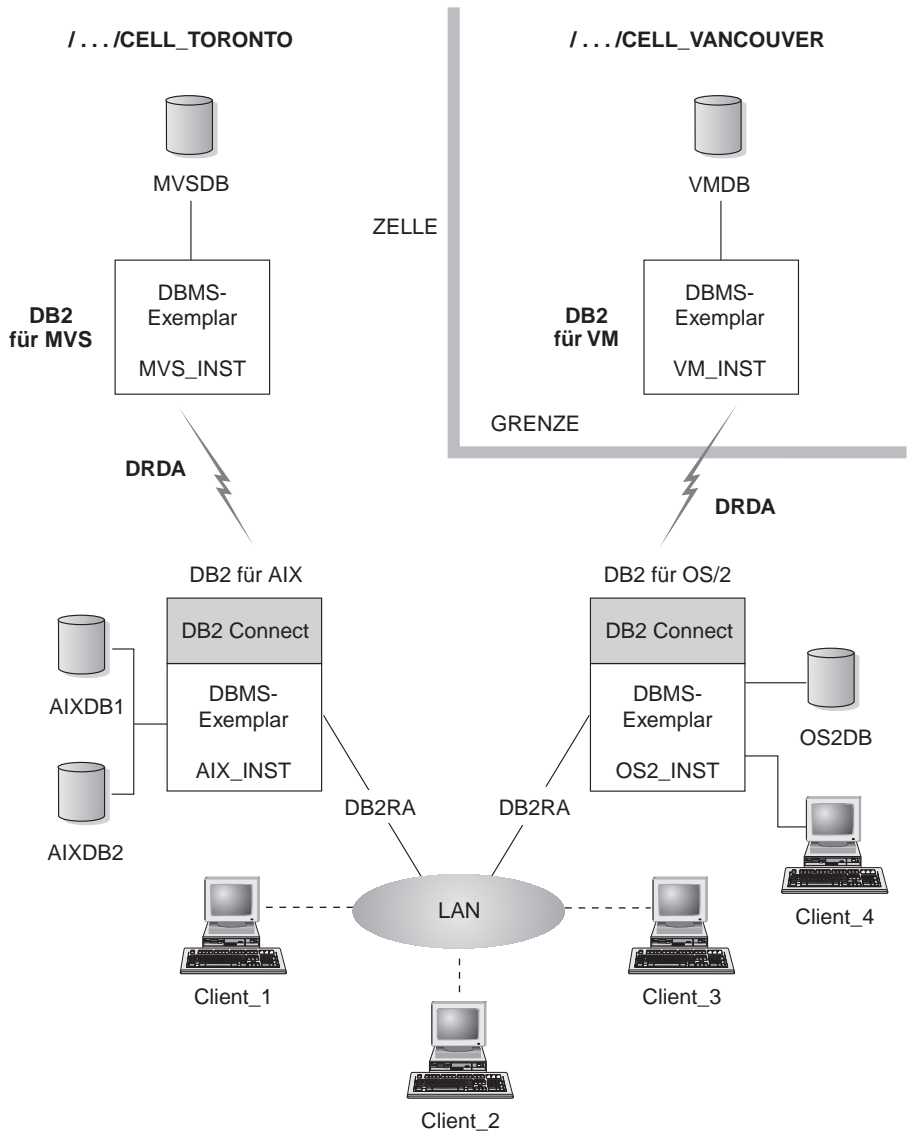


Abbildung 8. Konfiguration einer Netzwerkdatenbank

Damit die Clients in der Zelle TORONTO auf alle Datenbanken in beiden Zellen zugreifen können, müssen Werte in den Konfigurationsparametern des Datenbankmanagers angegeben und folgende Objekte erstellt werden:

- Ein Datenbankobjekt für jede Datenbank
- Ein Querverweisobjekt für die Zieldatenbanken der beiden Datenbank-Server für DB2 für AIX und für DB2 für OS/2.

- Ein einzelnes Objekt für Leitweginformationen, das allen Clients bekannt ist. Die Attribute geben an, welcher DB2 Connect-Knoten für die Datenbanken MVSDB und VMDB zu verwenden ist.

Folgende Abschnitte enthalten Beispiele dafür, wie ein Client eine Verbindung zu einer Datenbank herstellt:

- Herstellen einer Verbindung zu Datenbanken in derselben Zelle
- Herstellen einer Verbindung zu Datenbanken in einer anderen Zelle.

Diese Beispiele zeigen auch die Konfigurationsparameter des Datenbankmanagers, die angegeben werden müssen.

Herstellen einer Verbindung zu Datenbanken in derselben Zelle

Dieser Abschnitt enthält verschiedene Beispiele dafür, wie Clients eine Verbindung zu Datenbanken in derselben Zelle herstellen.

1. Client_1 stellt eine Verbindung zu AIXDB2 her. Die Datenbank verwendet den gleichen Verzeichnispfadnamen wie der Client.

Der Datenbankadministrator hat folgende Aufgaben:

- Angeben des Werts für den Verzeichnispfadnamen im Konfigurationsparameter *dir_path_name* (oder im Registrierungswert DB2DIRPATHNAME).
- Angeben von DCE als Wert für den Typ der Verzeichnisservices im Konfigurationsparameter *dir_type*.
- Angeben des Übertragungsprotokolls im Konfigurationsparameter *dft_client_comm* (oder im Registrierungswert DB2CLIENTCOMM).

Das lokale Systemdatenbankverzeichnis enthält AIXDB2 nicht, daher wird das DCE-Verzeichnis unter Verwendung des vollständig qualifizierten Namens durchsucht. Der Name wird erstellt, indem der Wert des Konfigurationsparameters *dir_path_name* (oder des Registrierungswerts DB2DIRPATHNAME) mit AIXDB2 zu einer Zeichenfolge verknüpft wird.

Dies führt zum folgenden Ablauf der Ereignisse:

- a. Client_1 ruft das Datenbankobjekt für AIXDB2 über den DCE-Namen der Datenbank `.../CELL_TORONTO/subsys/database/AIXDB2` ab.
- b. Durch dieses Objekt verfügt Client_1 über die Information, dass AIXDB2 das Datenbankprotokoll DB2RA verwendet, das auch von Client_1 benutzt wird.
- c. Da die Datenbankprotokolle übereinstimmen, liest Client_1 das DBMS-Querverweisobjekt (Database Management System) für AIX_INST, ruft den Attributwert für das Übertragungsprotokoll ab, der dem von ihm verwendeten Wert entspricht, und startet mit Hilfe dieser Informationen einen Dialog mit dem betreffenden DBMS-Exemplar.

2. Client_3 stellt eine Verbindung zu MVSDDB her. Die Datenbank verwendet den gleichen Verzeichnispfadnamen wie der Client, verwendet jedoch ein anderes Datenbankprotokoll als der Client.

Der Datenbankadministrator hat folgende Aufgaben:

- Angeben des Werts für den Verzeichnispfadnamen im Konfigurationsparameter *dir_path_name* (oder im Registrierungswert DB2DIRPATHNAME).
- Angeben von DCE als Wert für den Typ der Verzeichnisservices im Konfigurationsparameter *dir_type*.
- Angeben des Übertragungsprotokolls im Konfigurationsparameter *dft_client_comm* (oder im Registrierungswert DB2CLIENTCOMM).
- Angeben des DCE-Namens des Standardobjekts für Leitweginformationen im Konfigurationsparameter *route_obj_name* (oder im Registrierungswert DB2ROUTE).

Dies führt zum folgenden Ablauf der Ereignisse:

- a. Client_3 ruft das Datenbankobjekt für MVSDDB über den DCE-Namen der Datenbank `././CELL_TORONTO/subsys/database/MVSDDB` ab.
- b. Durch dieses Objekt stellt Client_3 fest, dass MVSDDB nur das Datenbankprotokoll DRDA verwendet, bei dem es sich nicht um das von Client_3 verwendete Protokoll handelt.
- c. Client_3 ruft anschließend das Objekt für Leitweginformationen über den Namen ab, der im Konfigurationsparameter *route_obj_name* oder im Registrierungswert DB2ROUTE definiert ist. Der Client findet die Zieldatenbankinformationen für MVSDDB.
- d. Client_3 liest das Querverweisobjekt für Datenbanken, das den Zieldatenbankinformationen für MVSDDB zugeordnet ist, ruft das Übertragungsprotokoll ab und sendet eine SQL-Anforderung CONNECT an den Router.
- e. Der Router richtet anschließend eine APPC-Verbindung mit MVSDDB ein.

Herstellen einer Verbindung zu Datenbanken in einer anderen Zelle

Dieser Abschnitt enthält ein Beispiel dafür, wie ein Client eine Verbindung zu einer Datenbank in einer anderen Zelle herstellt, wenn die Datenbankprotokolle unterschiedlich sind.

1. Client_3 wurde zuvor für die Verwendung folgender Einstellungen konfiguriert:
 - Die DCE-Verzeichnisservices (die Konfigurierung erfolgte durch Angabe des Werts DCE für den Parameter *dir_type*).

- Eine andere Zelle als CELL_VANCOUVER, wobei die Konfiguration mit dem Konfigurationsparameter *dir_path_name* vorgenommen wurde. Beispiel:

```
../../../../CELL_TORONTO/subsys/database/
```

2. Damit der Client_3 eine Verbindung zu VMDB herstellen kann, muss der Datenbankadministrator folgende Maßnahmen ergreifen:

- Explizites Katalogisieren von VMDB im lokalen Systemdatenbankverzeichnis. Zuordnung eines lokal eindeutigen Aliasnamens einer Datenbank zum DCE-Namen für VMDB und Ausführen einer Anweisung CONNECT für den Aliasnamen. Beispiel:

```
CATALOG GLOBAL DATABASE
../../../../CELL_VANCOUVER/subsys/database/VMDB AS VANVMDB
USING DIRECTORY DCE WITH "Kommentarzeichenfolge"
```

Anschließend:

```
CONNECT TO VANVMDB
```

- Angeben des Übertragungsprotokolls im Konfigurationsparameter *dft_client_comm* (oder im Registrierungswert DB2CLIENTCOMM).
- Angeben des DCE-Namens des Standardobjekts für Leitweginformationen im Konfigurationsparameter *route_obj_name* (oder im Registrierungswert DB2ROUTE).

Dies führt zum folgenden Ablauf der Ereignisse:

- a. Client_3 findet den vollständig qualifizierten DCE-Namen von VAN-VMDB in seinem Systemdatenbankverzeichnis.
- b. Client_3 ruft das Datenbankobjekt für VMDB über den DCE-Namen der Datenbank `../../../../CELL_VANCOUVER/subsys/database/VMDB` ab.
- c. Durch dieses Objekt stellt Client_3 fest, dass VMDB nur das Datenbankprotokoll DRDA verwendet, bei dem es sich nicht um das von Client_3 verwendete Protokoll handelt.
- d. Client_3 ruft anschließend das Objekt für Leitweginformationen über den Namen ab, der im Konfigurationsparameter *route_obj_name* oder im Registrierungswert DB2ROUTE definiert ist. Der Client findet die Zieldatenbankinformationen für VMDB.
- e. Client_3 liest das Querverweisobjekt für Datenbanken, das den Zieldatenbankinformationen für VMDB zugeordnet ist, ruft das Übertragungsprotokoll ab und sendet eine SQL-Anforderung CONNECT an den Router.
- f. Der Router richtet anschließend eine APPC-Verbindung mit VMDB ein.

Durchsuchen von Verzeichnissen

Wenn das DCE-Verzeichnis in einer Umgebung verwendet wird, in der alle Zieldatenbanken den gleichen Verzeichnispfadnamen verwenden, sind keine lokalen Verzeichnisse auf den Clients erforderlich.

In diesem Abschnitt wird beschrieben, in welcher Reihenfolge Verzeichnisse bei folgenden Befehlen durchsucht werden:

- Der Befehl ATTACH
- Anweisung CONNECT

Der Befehl ATTACH

Abb. 9 zeigt, wie die Verzeichnisse durchsucht werden, wenn ein Client eine Verbindung über den Befehl ATTACH zu einem DBMS-Exemplar (Database Management System) namens ABC_INST herstellt.

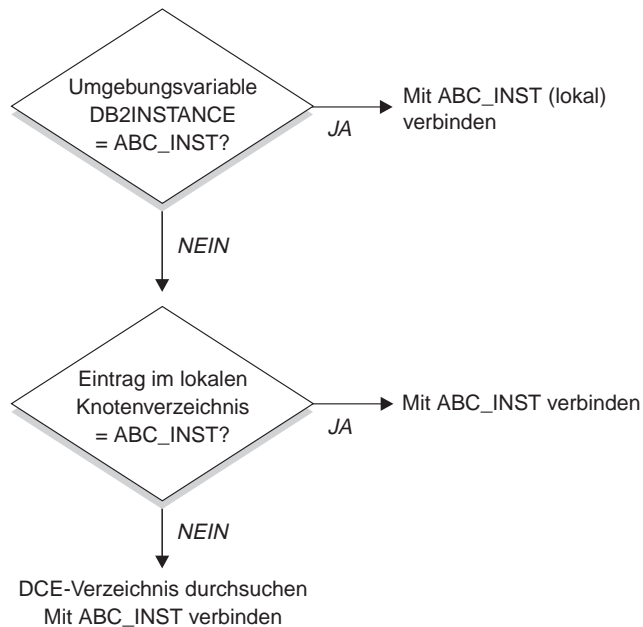


Abbildung 9. Verwenden von Verzeichnissen für die Verbindung zu einem DBMS-Exemplar

Anweisung CONNECT

Abb. 10 zeigt, wie die Verzeichnisse durchsucht werden, wenn ein Client die Verbindung zu einer Datenbank namens DBTEST herstellt.

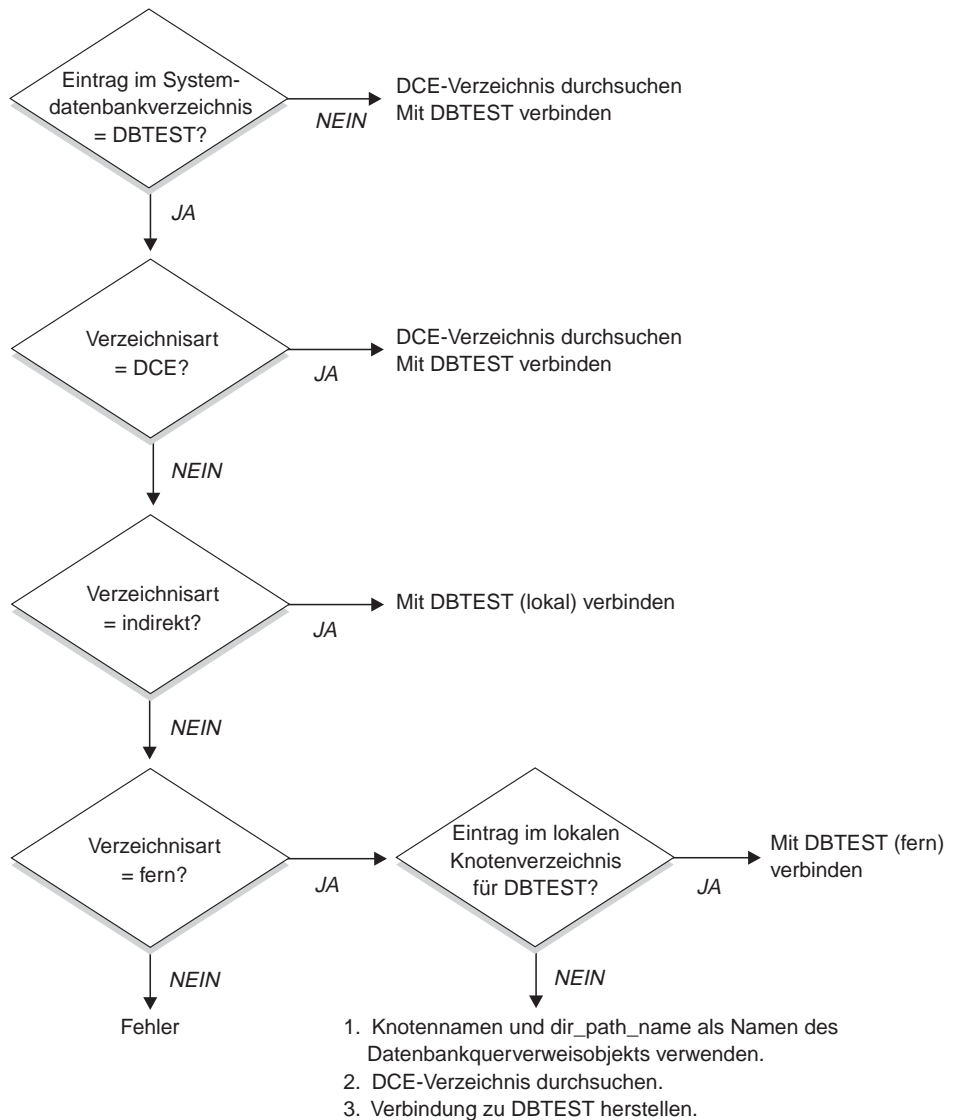


Abbildung 10. Verwenden von Verzeichnissen für die Verbindung zu einer Datenbank

Temporäres Außerkraftsetzen von Informationen in DCE-Verzeichnissen

Sie können mit dem lokalen Datenbankverzeichnis die DCE-Verzeichnisinformationen außer Kraft setzen.

Wenn beispielsweise der Befehl `CONNECT TO DBTEST` ausgeführt wird und für die Datenbank `./:/subsys/database/DBTEST` im DCE-Verzeichnis definiert ist, dass sie auf einem Host mit der Bezeichnung `JAGUAR` gespeichert wird, können Sie `DBTEST` temporär in eine andere Datenbank ändern, die sich auf einem Host mit dem Namen `STORM` befindet. Katalogisieren Sie hierfür `DBTEST` lokal als ferne Datenbank mit einem Knotenverzeichniseintrag, der auf `STORM` verweist.

Sie können einen Aliasnamen für eine Datenbank erstellen, deren DCE-Name nicht dem Verzeichnispfadnamen des Clients folgt. Einzelheiten zu diesem Befehl finden Sie in „Der Befehl `CATALOG GLOBAL DATABASE`“ auf Seite 394.

Aufgaben im Zusammenhang mit den Verzeichnisservices

Im Folgenden werden die Aufgaben aufgeführt, die zur Einrichtung und Verwendung der DCE-Verzeichnisservices durchgeführt werden müssen. Diese Aufgaben werden in den folgenden Abschnitten näher erläutert.

- Aufgaben des DCE-Administrators
Der DCE-Administrator muss das DCE-Verzeichnis aktualisieren, so dass die neuen Informationen über Datenbankressourcen hinzugefügt werden können.
- Aufgaben des Datenbankadministrators
Der Datenbankadministrator muss das DCE-Verzeichnis aktualisieren und Informationen für die Installation und Konfiguration von `DB2` bereitstellen.
- Aufgaben des Datenbankbenutzers
Der Datenbankbenutzer muss sich bei DCE anmelden und den Namen der Zieldatenbank kennen.

Zusätzlich richtet der Netzadministrator den Netzwerkzugriff für jeden Benutzerknoten ein. Weitere Informationen hierzu können Sie der Dokumentation zum Netzwerk entnehmen.

Aufgaben des DCE-Administrators

Der DCE-Administrator muss folgende Aufgaben ausführen, bevor die Verzeichnisobjekte erstellt oder gelesen werden können:

- Ordnen Sie die Unterverzeichnisstruktur für DB2 zu, z. B. `././subsys/database`.
- Erteilen Sie dem Datenbankadministrator die Zugriffsrechte zur Erstellung von Verzeichnisobjekten.
- Erteilen Sie den Datenbankbenutzern die Zugriffsrechte zum Lesen der Verzeichnisobjekte.
- Fügen Sie der *Tabelle für DCE-Attribute* die Informationen für die neuen Attribute der DCE-Verzeichnisobjekte hinzu.

Editieren Sie die Datei der CDS-Attribute (auf UNIX-Plattformen `/etc/dce/cds_attributes` bzw. unter OS/2 `X:\opt\dcelocal\etc\cds_attr`, wobei „X“ das entsprechende Laufwerk ist), und fügen Sie Folgendes an:

1.3.18.0.2.4.30	DB_Comment	char
1.3.18.0.2.4.31	DB_Communication_Protocol	char
1.3.18.0.2.4.32	DB_Database_Protocol	char
1.3.18.0.2.4.33	DB_Database_Locator_Name	char
1.3.18.0.2.4.34	DB_Native_Database_Name	char
1.3.18.0.2.4.35	DB_Object_Type	char
1.3.18.0.2.4.36	DB_Product_Name	char
1.3.18.0.2.4.37	DB_Product_Release	char
1.3.18.0.2.4.38	DB_Target_Database_Info	char
1.3.18.0.2.4.39	DB_Authentication	char
1.3.18.0.2.4.63	DB_Principal	char

- Sorgen Sie dafür, dass DCE ausgeführt wird, wenn Benutzer Zugriff auf die Datenbanken über die DCE-Verzeichnisservices benötigen.

Weitere Informationen können Sie der DCE-Dokumentation für die verwendete Plattform entnehmen.

Aufgaben des Datenbankadministrators

Der Datenbankadministrator hat folgende Aufgaben:

- Erkundigen Sie sich nach der Unterverzeichnisstruktur für die Datenbankressourcen beim DCE-Administrator. Beispiel: `././subsys/database`
- Bitten Sie bei der Installation des DB2-Datenbankmanagers den DCE-Administrator, die von DB2 benötigten neuen Attribute der DCE-Verzeichnisobjekte hinzuzufügen.
- Ordnen Sie einen eindeutigen Namen für jedes DBMS-Exemplar (Database Management System) in der DCE-Unterverzeichnisstruktur zu. Beispiel: `././subsys/database/AIX_INST`

- Geben Sie für jedes DBMS-Exemplar (Database Management System) die Konfigurationsparameter des Datenbankmanagers für DCE an.
 - *dir_type*
 - *dir_obj_name*
 - *dir_path_name*
 - *route_obj_name*
 - *dft_client_comm*
 - *dft_client_adpt*

Einige der Konfigurationsparameter können temporär durch vom Client gesetzte Registrierungsvariablen außer Kraft gesetzt werden. Weitere Informationen finden Sie in „Konfigurationsparameter und Registrierungswerte“ auf Seite 392.

- Ordnen Sie einen eindeutigen Namen für jede Datenbank in der DCE-Unterverzeichnisstruktur zu. Geben Sie den Namen im Parameter *dir_obj_name* in der Konfigurationsdatei der Datenbank an.
- Erstellen Sie die Objekte für die DCE-Verzeichnisservices mit Hilfe der *cdscp*-Befehle von DCE zum Erstellen und Anzeigen von Objekten. Die Objekte werden getrennt vom Installationsprozess des Datenbankmanagers und vom Prozess zum Starten des Datenbankmanagerexemplars erstellt. Es gibt drei Arten von Objekten.
 - Für jede Zieldatenbank wird ein Datenbankobjekt benötigt.
 - Ein Querverweisobjekt für Datenbanken wird für jedes DB2 Connect-Exemplar und jedes DBMS-Exemplar (ohne DB2 Connect) benötigt, dem mehr als eine Datenbank zugeordnet ist.
 - Objekte für Leitweginformationen werden für den Zugriff auf eine Host-Datenbank benötigt.
- Je nach Umgebung muss der Datenbankadministrator Folgendes festlegen:
 - Wie die Clients entsprechend den Datenbanken, auf die sie zugreifen, und den von ihnen verwendeten Übertragungsprotokollen in logische Gruppen eingeteilt werden.
 - Wie viele Objekte für Leitweginformationen benötigt werden.
 - Welche Zieldatenbanken in jedem Objekt für Leitweginformationen eingetragen werden sollen.
 - Welche Objekte für Leitweginformationen welcher Gruppe von Clients bekannt sein sollten.

Weitere Einzelheiten zu den Objekten finden Sie in „Erstellen von Verzeichnisobjekten“ auf Seite 377.

Aufgaben des Datenbankbenutzers

Der Datenbankbenutzer hat folgende Aufgaben:

- Erfragen Sie den Namen der Datenbank beim Datenbankadministrator. Dieser Name kann ein einfacher einteiliger Name oder ein vollständig qualifizierter DCE-Name sein.
- Geben Sie, falls erforderlich, die für die DCE-Verzeichnisservices benötigten Werte in den Registrierungsvariablen an. Vom Client definierte Registrierungsvariablen können die Konfigurationsparameter temporär außer Kraft setzen.
 - Wenn Zugriff auf die Host-Datenbank benötigt wird, erfragen Sie beim Datenbankadministrator den vollständig qualifizierten DCE-Namen des Objekts für Leitweginformationen. Wenn dieser Name nicht im Parameter *route_obj_name* angegeben ist oder wenn es sich um einen anderen Namen handelt, geben Sie diesen Namen in der Registrierungsvariablen DB2ROUTE an, bevor Sie versuchen, eine Verbindung zur Host-Datenbank herzustellen.
 - Wenn das von Ihnen bevorzugte Übertragungsprotokoll nicht im Parameter *dft_client_comm* angegeben ist oder es sich um ein anderes Protokoll handelt, geben Sie das Übertragungsprotokoll für den Client in der Registrierungsvariablen DB2CLIENTCOMM an. Es folgen **einige** Beispiele für UNIX:

```
db2set DB2CLIENTCOMM=tcpip
db2set DB2CLIENTCOMM=appc
db2set DB2CLIENTCOMM=tcpip,appc
db2set DB2CLIENTCOMM=appc,tcpip
```

Einige Beispiele für OS/2 sind:

```
db2set DB2CLIENTCOMM=ipxspx
db2set DB2CLIENTCOMM=netbios
db2set DB2CLIENTCOMM=tcpip,ipxspx,netbios
db2set DB2CLIENTCOMM=netbios,tcpip,ipxspx,appc
```

Einige Beispiele für Windows Betriebssysteme:

```
db2set DB2CLIENTCOMM=npipe
db2set DB2CLIENTCOMM=netbios
db2set DB2CLIENTCOMM=tcpip,ipxspx,netbios
db2set DB2CLIENTCOMM=netbios,tcpip,ipxspx,appc,npipe
```

Wenn mehr als ein Übertragungsprotokoll vorhanden ist, wird das erste angegebene Übertragungsprotokoll verwendet.

- Wenn eine der Datenbanken einen DCE-Namen hat, der sich nicht in dem im Konfigurationsparameter *dir_path_name* oder in der Registrierungsvariablen DB2DIRPATHNAME definierten Verzeichnispfad befindet, müssen Sie die Datenbank explizit mit dem Befehl CATALOG GLOBAL DATABASE katalogisieren. Weitere Informationen finden Sie in „Der Befehl CATALOG GLOBAL DATABASE“ auf Seite 394.
- Melden Sie sich bei DCE an, bevor Sie eine Verbindung mit der Zieldatenbank oder eine Verbindung zum Datenbanke Exemplar herstellen. Weitere Informationen über den Anmeldebefehl finden Sie im *OSF DCE Administration Guide*.

Einschränkungen bei den Verzeichnisservices

In diesem Abschnitt werden Einrichtungen beschrieben, die nicht unterstützt werden.

- Möglicherweise werden nicht alle Datenbank-Clients unterstützt. Informationen, ob die DCE-Verzeichnisservices von Ihrem DB2-Client unterstützt werden, finden Sie im Handbuch *Einstieg*. Gegenwärtig werden DB2-Clients nur für alle UNIX-Betriebssysteme, OS/2 sowie die unterstützten Versionen des Betriebssystems Windows unterstützt.
- Ein Client kann nicht über die DCE-Verzeichnisservices eine Verbindung mit einem Server unter DB2 für OS/2 Version 1 herstellen.
- Nur die Clients der unterstützten Versionen des Betriebssystems Windows können ein beliebiges oder alle der TCP/IP-, APPC-, NetBIOS-, IPX/SPX- oder NPIPE-Protokolle verwenden. Nur OS/2-Clients können ein beliebiges oder alle der Protokolle TCP/IP, APPC, NetBIOS und IPX/SPX verwenden. Alle unterstützten Clients unter UNIX können nur die Protokolle TCP/IP und APPC verwenden.
- Mit den Befehlen LIST DATABASE (oder NODE) DIRECTORY werden nur Einträge aus den lokalen Verzeichnissen, jedoch keine Einträge aus dem DCE-Verzeichnis angezeigt. Zum Anzeigen der Objekte in DCE kann der Befehl *cdscp show object* verwendet werden.

- Wenn alle folgenden Bedingungen erfüllt sind, muss der Eigner des Datenbankmanagerexemplars eine Anmeldung bei DCE durchführen, bevor der Datenbankmanager (mit dem Befehl `db2start`) gestartet wird.
 - Das Datenbankmanagerexemplar ist über den Konfigurationsparameter `dir_type` zur Unterstützung der DCE-Verzeichnisservices konfiguriert.
 - Das Objekt für die Zellenverzeichnisservices kann nur gelesen werden, indem eine explizite Anmeldung bei DCE vorgenommen wird.
 - Es muss auf das DCE-Verzeichnis zugegriffen werden, damit eines der folgenden Elemente unterstützt wird:
 - Eine durch den Konfigurationsparameter `tm_database` angegebene Transaktionsmanagerdatenbank, die sich auf einem anderen Exemplar befindet.
 - Ein Client, der die DCE-Verzeichnisservices nicht unterstützen kann oder zur Verwendung von DCE-Verzeichnisservices nicht konfiguriert ist.

Anmerkung: Bei der Ausführung der Anmeldung bei DCE sollte ein Principal verwendet werden, dessen Zugriffsberechtigung eine lange Lebensdauer hat.

- Wenn ein Gateway unter DDCS Version 2.2 (oder frühere Version) verwendet wird, um einen Client, der DCE-Verzeichnisservices verwendet, mit einem DRDA-Server zu verbinden, müssen Sie den Aliasnamen für die Datenbank im lokalen Verzeichnis des Gateways katalogisieren. Dieser Aliasname für die Datenbank muss dem Aliasnamen auf dem Client entsprechen, und er muss dieselbe Datenbank darstellen.
- Beim Einsatz von Clients für unterstützte Versionen des Betriebssystems Windows wird `DB2DCE.DLL` verwendet. Diese Datei befindet sich im Unterverzeichnis `bin` des Unterverzeichnisses `sqllib`. Wenn der DCE-Anbieter Gradient ist, ist die Datei `DB2DCE.GRD` standardmäßig äquivalent zur Datei `DB2DCE.DLL`. Wenn der DCE-Anbieter IBM ist, muss die Datei `DB2DCE.IBM` nach `DB2DCE.DLL` kopiert werden.

Anhang C. Benutzer-Exit zur Datenbankwiederherstellung

Anmerkung: Alle Informationen zu diesen Themen sowie die vergleichbaren Themen aus den Handbüchern *Command Reference* und *Administrative API Reference* wurden im Handbuch *Data Recovery and High Availability Guide and Reference* zusammengefasst.

Informationen zu diesen Themen finden Sie ausschließlich im Handbuch *Data Recovery and High Availability Guide and Reference*.

Anhang D. Absetzen von Befehlen an mehrere Datenbankpartitionsserver

In einem partitionierten Datenbanksystem kann es wünschenswert sein, Befehle absetzen zu können, die auf Maschinen im Exemplar oder auf Datenbankpartitionsservern (Knoten) ausgeführt werden. Sie haben diese Möglichkeit mit Hilfe des Befehls **rah** oder des Befehls **db2_all**. Der Befehl **rah** ermöglicht Ihnen das Absetzen von Befehlen, die Sie auf Maschinen im Exemplar ausführen wollen. Wenn Sie die Befehle auf Datenbankpartitionsservern im Exemplar ausführen wollen, verwenden Sie den Befehl **db2_all**. Dieser Abschnitt enthält eine Übersicht über diese Befehle. Die folgenden Informationen beziehen sich ausschließlich auf ein partitioniertes Datenbanksystem.

Anmerkungen:

1. Auf Plattformen unter UNIX kann Ihre Anmelde-Shell eine Korn-Shell oder eine beliebige andere Shell sein. Jedoch gibt es Unterschiede in der Art, wie verschiedene Shells Befehle behandeln, die Sonderzeichen enthalten.
2. Unter Windows NT müssen Sie mit einem Benutzerkonto angemeldet sein, dass zur Administratorengruppe gehört, um die Befehle **rah** oder **db2_all** ausführen zu können.

Informationen zum Funktionsbereich eines Befehls finden Sie im Handbuch *Command Reference*. Dieses Handbuch enthält Angaben darüber, ob ein Befehl auf einem einzelnen Datenbankpartitions-Server oder auf allen ausgeführt wird. Wenn der Befehl auf einem Datenbankpartitions-Server ausgeführt wird und Sie ihn auf allen von ihnen ausführen wollen, verwenden Sie dazu den Befehl **db2_all**. Eine Ausnahme bildet der Befehl **db2trc**, der auf allen logischen Knoten (Datenbankpartitionsservern) auf einer Maschine ausgeführt wird. Wenn Sie den Befehl **db2trc** auf allen logischen Knoten aller Maschinen ausführen wollen, verwenden Sie dazu den Befehl **rah**.

Befehle

Die Befehle können sequenziell auf einem Datenbankpartitions-Server nach dem anderen oder parallel ausgeführt werden. Wenn Sie die Befehle auf Plattformen unter UNIX parallel ausführen, können Sie wählen, ob die Ausgabe an einen Puffer gesendet werden soll, in dem sie zur späteren Anzeige gesammelt wird (Standardmodus), oder ob die Ausgabe auf der Maschine angezeigt werden soll, auf der der Befehl abgesetzt wird. Unter Windows NT wird die Ausgabe bei paralleler Ausführung der Befehle auf der Maschine angezeigt, auf der der Befehl abgesetzt wurde.

Zur Verwendung des Befehls **rah** geben Sie Folgendes ein:

```
rah befehl
```

Zur Verwendung des Befehls **db2_all** geben Sie Folgendes ein:

```
db2_all befehl
```

Hilfe zur Syntax von **rah** erhalten Sie durch folgende Eingabe:

```
rah "?"
```

Als Befehl kann fast alles angegeben werden, was in eine interaktive Eingabeaufforderung eingegeben werden könnte, einschließlich, zum Beispiel, mehrerer Befehle, die nacheinander ausgeführt werden. Auf Plattformen unter UNIX werden mehrere Befehle durch ein Semikolon (;) voneinander getrennt. Unter Windows NT werden mehrere Befehle durch ein Et-Zeichen (&) voneinander getrennt. Das Trennzeichen darf jedoch nicht nach dem letzten Befehl verwendet werden.

Das folgende Beispiel zeigt die Verwendung des Befehls **db2_all**, um die Datenbankkonfiguration auf allen Datenbankpartitionen zu ändern, die in der Knotenkonfigurationsdatei angegeben sind. Da das Semikolon (;) in den doppelten Anführungszeichen mit eingeschlossen ist, wird die Anforderung parallel ausgeführt:

```
db2_all ";UPDATE DB CFG FOR sample USING LOGFILSIZ=100"
```

Befehlsbeschreibungen

Sie können die folgenden Befehle verwenden:

Befehl	Beschreibung
rah	Führt den Befehl auf allen Maschinen aus.
db2_all	Führt den Befehl auf allen Datenbankpartitionsservern aus, die Sie angeben.
db2_kill	Stoppt abrupt alle Prozessen, die auf mehreren Datenbankpartitionsservern ausgeführt werden, und bereinigt alle Ressourcen auf allen Datenbankpartitionsservern. Durch diesen Befehl wird Ihre Datenbank inkonsistent. Führen Sie diesen Befehl <i>nicht</i> aus, es sei denn, Sie erhalten vom IBM Kundendienst eine entsprechende Anweisung.
db2_call_stack	Auf Plattformen unter UNIX werden alle Prozesse, die auf allen Datenbankpartitionsservern ausgeführt werden, veranlasst, eine Ablaufverfolgungsliste für Aufrufe in das Systemprotokoll zu schreiben.

Unter Windows NT werden alle Prozesse, die auf allen Datenbankpartitionsservern ausgeführt werden, veranlasst, eine Ablaufverfolgungsliste für Aufrufe in die Datei *Pxxxx.nnn* zu schreiben, wobei *Pxxxx* die Prozess-ID und *nnn* die Knotennummer ist.

Auf Plattformen unter UNIX führen diese Befehle **rah** mit bestimmten impliziten Einstellungen wie zum Beispiel folgenden aus:

- Parallele Ausführung auf allen Maschinen
- Puffern der Befehlsausgabe in `/tmp/$USER/db2_kill` bzw. `/tmp/$USER/db2_call_stack`

Unter Windows NT führen diese Befehle **rah** mit paralleler Ausführung auf allen Maschinen aus.

Angeben des auszuführenden Befehls

Sie können den Befehl angeben:

- In der Befehlszeile als Parameter
- Als Antwort auf die Eingabeaufforderung, wenn Sie keine Parameter angeben

Die Angabe über die Eingabeaufforderung ist zu empfehlen, wenn der Befehl die folgenden Sonderzeichen enthält:

| &
; < > () { } [] nicht ersetztes \$

Wenn Sie den Befehl als Parameter in der Befehlszeile angeben, müssen Sie ihn in doppelte Anführungszeichen setzen, wenn er eines oder mehrere der aufgeführten Sonderzeichen enthält.

Anmerkung: Auf Plattformen unter UNIX wird der Befehl Ihrem Befehlsprotokoll Form hinzugefügt, ganz so als ob Sie ihn in die Eingabeaufforderung eingegeben hätten.

Alle Sonderzeichen im Befehl können normal eingegeben werden (d. h. ohne Anführungszeichen, außer dem umgekehrten Schrägstrich `\`). Wenn Sie einen umgekehrten Schrägstrich (`\`) in Ihrem Befehl benötigen, müssen Sie zwei Schrägstriche eingeben (`\\`).

Anmerkung: Wenn auf Plattformen unter UNIX keine Korn-Shell verwendet wird, können alle Sonderzeichen im Befehl normal eingegeben werden (d. h. ohne Anführungszeichen, außer bei den Zeichen `"`, `\`, nicht ersetztes `$` und einfaches Anführungszeichen `()`). Wenn Sie eines dieser Zeichen in Ihrem Befehl benötigen, müssen Sie ihnen drei umgekehrte Schrägstriche (`\\\`) voranstellen. Wenn

Sie zum Beispiel einen umgekehrten Schrägstrich (\) in Ihrem Befehl benötigen, müssen Sie vier umgekehrte Schrägstriche (\\) eingeben.

Wenn Sie ein doppeltes Anführungszeichen (") in Ihrem Befehl benötigen, müssen Sie ihm drei umgekehrte Schrägstriche voranstellen. Beispiel: \\".

Anmerkungen:

1. Auf Plattformen unter UNIX können Sie kein einfaches Anführungszeichen (') im Befehl angeben, sofern Ihre Befehls-Shell nicht irgendeine Möglichkeit zur Eingabe eines einfachen Anführungszeichens innerhalb einer in einfache Anführungszeichen gesetzten Zeichenfolge bereitstellt.
2. Unter Windows NT können Sie kein einfaches Anführungszeichen (') im Befehl angeben, sofern Ihr Befehlsfenster nicht irgendeine Möglichkeit zur Eingabe eines einfachen Anführungszeichens innerhalb einer in einfache Anführungszeichen gesetzten Zeichenfolge bereitstellt.

Wenn Sie eine beliebige Shell-Prozedur der Korn-Shell ausführen, die Logik enthält, die von stdin im Hintergrund gelesen wird, sollten Sie stdin ausdrücklich zu einer Quelle umleiten, an der der Prozess den Lesevorgang ohne Unterbrechungen am Terminal (Nachricht SIGTTIN) durchführen kann. Zum Umleiten von stdin können Sie eine Prozedur im folgenden Format ausführen:

```
shell_script </dev/null &
```

wenn keine Eingabe bereitgestellt werden muss.

Auf ähnliche Weise sollten Sie stets </dev/null angeben, wenn db2_all im Hintergrund ausgeführt wird. Beispiel:

```
db2_all ";run_this_command" </dev/null &
```

Auf diese Weise können Sie stdin umleiten und Unterbrechungen am Terminal vermeiden.

Eine Alternative zu dieser Methode, die Sie anwenden können, wenn Sie die Ausgabe von einem fernen Befehl nicht stört, ist die Verwendung der Option „daemonize“ im Präfix db2_all:

```
db2_all ";daemonize_this_command" &
```

Paralleles Ausführen von Befehlen auf Plattformen unter UNIX

Anmerkung: Die Informationen dieses Abschnitts beziehen sich ausschließlich auf Plattformen, die auf UNIX basieren.

Standardmäßig wird der Befehl sequenziell auf jeder Maschine ausgeführt. Jedoch können Sie mit Hilfe von rshells im Hintergrund, für die Sie den Befehl mit bestimmten Präfixsequenzen präfigieren, angeben, dass die Befehle

parallel ausgeführt werden. Wenn die rshell im Hintergrund ausgeführt wird, schreibt jeder Befehl die Ausgabe in eine Pufferdatei auf der jeweiligen fernen Maschine. Dieser Prozess ruft die Ausgabe in zwei Teilen ab:

1. Nach Beendigung des fernen Befehls
2. Nach Beendigung der rshell, was später geschehen kann, wenn einige Prozesse andauern

Der Name der Pufferdatei ist standardmäßig /tmp/\$USER/rahout. Er kann aber durch die Umgebungsvariablen \$RAHBUFDIR/\$RAHBUFNAME definiert werden.

Wenn Sie angeben, dass die Befehle gleichzeitig ausgeführt werden sollen, schaltet diese Prozedur standardmäßig einen weiteren Befehl vor den Befehl, der an alle Hosts gesendet wird, um zu überprüfen, ob die Umgebungsvariablen \$RAHBUFDIR und \$RAHBUFNAME für die Pufferdatei verwendbar sind. Sie erstellt das Verzeichnis in \$RAHBUFDIR definierte Verzeichnis. Sie können dies unterdrücken, indem Sie eine Umgebungsvariable RAHCHECKBUF=no exportieren. Dies dient der Zeitersparnis, wenn Sie bereits wissen, dass das Verzeichnis existiert und verwendbar ist.

Bevor Sie **rah** verwenden, um einen Befehl gleichzeitig auf mehreren Maschinen auszuführen, sollten Sie folgende Schritte ausführen:

- Stellen Sie sicher, dass ein Verzeichnis /tmp/\$USER für Ihre Benutzer-ID auf jeder Maschine vorhanden ist. Zur Erstellung eines Verzeichnisses, falls es noch nicht vorhanden ist, führen Sie folgenden Befehl aus:

```
rah
")mkdir /tmp/$USER"
```
- Fügen Sie die folgende Zeile in Ihre Datei .kshrc (für Korn-Shell-Syntax) oder .profile ein, und geben Sie sie auch in Ihre aktuelle Sitzung ein:

```
export RAHCHECKBUF=no
```
- Stellen Sie sicher, dass jede Maschinen-ID, auf der Sie den fernen Befehl ausführen, in der zugehörigen Datei .rhosts einen Eintrag für die ID hat, auf der der Befehl **rah** ausgeführt wird, und dass die ID, auf der der Befehl **rah** ausgeführt wird, in der zugehörigen Datei .rhosts einen Eintrag für jede Maschinen-ID hat, auf der Sie den fernen Befehl ausführen.

Überwachen des rah-Prozesses auf Plattformen unter UNIX

Anmerkung: Die Informationen dieses Abschnitts beziehen sich ausschließlich auf Plattformen, die auf UNIX basieren.

Während einige ferne Befehle immer noch aktiv sind oder die gepufferte Ausgabe noch gesammelt wird, überwachen von rah gestartete Prozesse die Aktivitäten zu folgenden Zwecken:

- Schreiben von Nachrichten an die Workstation, die angeben, welche Befehle nicht ausgeführt wurden

- Abrufen der gepufferten Ausgabe

Die Informationsnachrichten werden mit einem Intervall geschrieben, das durch die Umgebungsvariable RAHWAITTIME gesteuert wird. Im Hilfetext finden Sie Einzelheiten zu ihrer Definition. Alle Informationsnachrichten können durch Exportieren von RAHWAITTIME=0 vollständig unterdrückt werden.

Der primäre Überwachungsprozess ist ein Befehl, dessen Befehlsname (wie vom Befehl ps angezeigt) **rahwaitfor** lautet. Die erste Informationsnachricht teilt Ihnen die pid (Prozess-ID) dieses Prozesses mit. Alle anderen Überwachungsprozesse erscheinen als **ksh**-Befehle, die die **rah**-Prozedur (den Namen der symbolischen Verbindung) ausführen. Falls erwünscht, können Sie alle Überwachungsprozesse durch folgenden Befehl stoppen:

```
kill <pid>
```

Dabei ist <pid> die Prozess-ID des primären Überwachungsprozesses. Geben Sie keine Signalnummer an. Lassen Sie den Standardwert auf 15. Dadurch werden die fernen Befehle in keiner Weise beeinflusst, sondern es wird das automatische Anzeigen der gepufferten Ausgabe verhindert. Beachten Sie, dass es während der Dauer einer einzelnen **rah**-Ausführung zwei oder mehr verschiedene Gruppen von Überwachungsprozessen geben kann, die zu verschiedenen Zeiten aktiv sind. Wenn Sie zu einem Zeitpunkt die aktuelle Gruppe von Überwachungsprozessen stoppen, werden keine weiteren mehr gestartet.

Wenn Ihre reguläre Anmelde-Shell keine Korn-Shell (z. B. /bin/ksh) ist, können Sie **rah** verwenden, jedoch gibt es eine Reihe leicht abweichender Regeln für die Eingabe von Befehlen, die die folgenden Sonderzeichen enthalten:

```
" nicht ersetzt  
$ '
```

Weitere Informationen erhalten Sie über die Eingabe von **rah** "?". Außerdem darf in einer auf UNIX basierenden Umgebung die Anmelde-Shell auf der ID, die die **rah**-Prozedur ausführt, keine Korn-Shell sein, wenn die Anmelde-Shell auf der ID, die die fernen Befehle ausführt, keine Korn-Shell ist. (**rah** entscheidet anhand der lokalen ID, ob die Shell der fernen ID eine Korn-Shell ist.) Die Shell darf keinerlei Substitution oder spezielle Verarbeitung an einer Zeichenfolge durchführen, die in einfache Anführungszeichen gesetzt ist. Die Zeichenfolge muss in der Form belassen werden, in der sie vorliegt.

Zusätzliche rah-Informationen (Run All Hosts) (nur für Solaris und AIX)

Zur Leistungsverbesserung wurde **rah** erweitert, um eine Baumstrukturlogik für große Systeme zu verwenden. Das heißt **rah** überprüft, wie viele Knoten in der Liste enthalten sind, und wenn diese Anzahl einen Schwellenwert überschreitet, erstellt **rah** eine Teilmenge von der Liste und sendet einen rekursiven Aufruf von sich an diese Knoten. An diesen Knoten wendet das rekursiv

aufgerufene **rah** dieselbe Logik an, bis die Liste so klein ist, dass mit der Standardlogik (nun die Logik "leaf-of-tree") die Befehle an alle Knoten in der Liste versendet werden können. Der Schwellenwert kann mit der Umgebungsvariable RAHTREETHRESH angegeben werden, oder er nimmt standardmäßig den Wert 15 an.

Im Fall eines Systems, bei dem mehrere logische Knoten auf einen physischen Knoten kommen, sendet **db2_all** den rekursiven Aufruf bevorzugt an bestimmte physische Knoten, die dann **rah** an die anderen logischen Knoten auf demselben physischen Knoten senden, so dass auch die Übertragungen zwischen den physischen Knoten reduziert werden. (Dies gilt nur für **db2_all**, nicht **rah**, da **rah** immer nur an bestimmte physische Knoten sendet.)

Präfixsequenzen

Eine Präfixsequenz besteht aus einem oder mehreren Sonderzeichen. Eine oder mehrere Präfixsequenzen müssen unmittelbar vor den Zeichen des Befehls ohne zwischengeschaltete Leerzeichen eingegeben werden. Wenn Sie mehr als eine Sequenz angeben wollen, können Sie sie in beliebiger Reihenfolge eingeben. Zeichen innerhalb einer Mehrzeichensequenz müssen jedoch in der richtigen Reihenfolge eingegeben werden. Wenn Sie Präfixsequenzen eingeben, müssen Sie den gesamten Befehl, einschließlich der Präfixsequenzen, in doppelte Anführungszeichen setzen, wie in folgenden Beispielen gezeigt:

- Auf UNIX-Plattformen:
`rah "};ps -F pid,ppid,etime,args -u $USER"`
- Unter Windows NT:
`rah "||db2 get db cfg for sample"`

Folgende Präfixsequenzen sind möglich:

Sequenz	Zweck
	Führt die Befehle nacheinander im Hintergrund aus.
&	Führt die Befehle nacheinander im Hintergrund aus und beendet den Befehl, wenn alle fernen Befehle ausgeführt sind, auch wenn einige Prozesse noch aktiv sind. Dies kann später sein, wenn zum Beispiel Kindprozesse (auf Plattformen unter UNIX) oder Hintergrundprozesse (unter Windows NT) noch aktiv sind. In diesem Fall startet der Befehl einen getrennten Hintergrundprozess, um jede ferne Ausgabe abzurufen, die nach Beendigung des Befehls generiert wurde, und schreibt diese zurück an die Ausgangsmaschine.

- Anmerkung:** Auf Plattformen unter UNIX wird durch die Angabe von & die Leistung beeinträchtigt, da mehr **rsh**-Befehle erforderlich werden.
- || Führt die Befehle parallel im Hintergrund aus.
- ||& Führt die Befehle parallel im Hintergrund aus und beendet den Befehl, wenn alle fernen Befehle ausgeführt sind, wie oben für die Sequenz |& beschrieben.
- Anmerkung:** Auf Plattformen unter UNIX wird durch die Angabe von & die Leistung beeinträchtigt, da mehr **rsh**-Befehle erforderlich werden.
- ; Hat die gleiche Funktion wie ||&. Dies ist eine kürzere alternative Form.
- Anmerkung:** Auf Plattformen unter UNIX wird durch die Angabe von ; die Leistung im Vergleich zu || herabgesetzt, da mehr **rsh**-Befehle erforderlich werden.
-] Schaltet die Punktdateiausführung des Profils des Benutzers vor die Ausführung des Befehls.
- Anmerkung:** Nur für auf UNIX basierende Plattformen verfügbar.
- } Schaltet die Punktdateiausführung der in \$RAHENV definierten Datei (wahrscheinlich .kshrc) vor die Ausführung des Befehls.
- Anmerkung:** Nur für auf UNIX basierende Plattformen verfügbar.
- }} Schaltet die Punktdateiausführung des Profils des Benutzers gefolgt von der Ausführung der in \$RAHENV definierten Datei (wahrscheinlich .kshrc) vor die Ausführung des Befehls.
- Anmerkung:** Nur für auf UNIX basierende Plattformen verfügbar.
-) Unterdrückt die Ausführung des Benutzerprofils und der in \$RAHENV definierten Datei.
- Anmerkung:** Nur für auf UNIX basierende Plattformen verfügbar.
- ' Meldet den Befehlsaufruf zurück an die Maschine.

- < Sendet an alle Maschinen außer dieser.
- <<-**nnn**< Sendet an alle außer Datenbankpartitions-Server *nnn* (d. h. alle Datenbankpartitionsserver in der Datei `db2nodes.cfg` außer dem mit der Knotennummer *nnn*. Siehe den ersten Abschnitt nach der letzten Präfixsequenz in dieser Tabelle).
- <<**+nnn**< Sendet nur an Datenbankpartitions-Server *nnn* (der Datenbankpartitions-Server in der Datei `db2nodes.cfg`, dessen Knotennummer *nnn* lautet. Siehe den ersten Abschnitt nach der letzten Präfixsequenz in dieser Tabelle).

(Leerzeichen)

Führt den fernen Befehl im Hintergrund bei geschlossenen `stdin` (Standardeingabeeinheit), `stdout` (Standardausgabeeinheit) und `stderr` (Standardfehlereinheit) aus. Diese Option ist nur gültig, wenn der Befehl im Hintergrund ausgeführt wird, d. h. nur in einer Präfixsequenz, die auch `\` oder `;` enthält. Sie ermöglicht dem Befehl eine wesentlich frühere Beendigung (sobald der ferne Befehl eingeleitet wurde). Wenn Sie dieses Präfixzeichen in der **rah**-Befehlszeile angeben, setzen Sie den Befehl entweder in einfache Anführungszeichen oder setzen Sie den Befehl in doppelte Anführungszeichen und stellen dem Präfixzeichen einen umgekehrten Schrägstrich (`\`) voran. Beispiel:

```
rah ' ; mydaemon'
```

oder

```
rah " ;\ mydaemon"
```

Bei der Ausführung als Hintergrundprozess wartet der **rah**-Befehl nie auf die Rückgabe irgendeiner Ausgabe.

- > Ersetzt die Vorkommen von `<>` durch den Maschinennamen.
- " Ersetzt die Vorkommen von `()` durch den Maschinenindex und die Vorkommen von `##` durch die Knotennummer.

Anmerkungen:

1. Der Maschinenindex ist eine Nummer, die einer Maschine innerhalb des Datenbanksystems zugeordnet ist. Wenn Sie nicht mit mehreren logischen Knoten arbeiten, entspricht der Maschinenindex für eine Maschine der Knotennummer für diese Maschine in der Knotenkonfigurationsdatei. Der Maschinenindex für eine Maschine in einer Umgebung mit mehreren logischen Knoten wird ermittelt, indem mehrere Einträge für die Maschinen, die mehrere logische Knoten ausführen, nur einfach gezählt werden. Wenn zum Beispiel

MACH1 zwei logische Knoten ausführt und MACH2 ebenfalls zwei logische Knoten ausführt, ist die Knotennummer für MACH3 in der Knotenkonfigurationsdatei 5. Der Maschinenindex für MACH3 wäre jedoch 3.

Unter Windows NT sollten Sie die Knotenkonfigurationsdatei unter keinen Umständen editieren. Verwenden Sie den Befehl **db2nlist**, wenn Sie den Maschinenindex abrufen wollen. Weitere Einzelheiten finden Sie im Handbuch *DB2 Enterprise - Extended Edition für Windows Einstieg*.

2. Wenn " angegeben wird, werden mehrfache Vorkommen aus der Liste der Maschinen nicht eliminiert. Wenn Sie mehrfache Vorkommen ausschließen wollen, siehe „Eliminieren mehrfacher Einträge aus der Liste der Maschinen“ auf Seite 421.

Bei Verwendung der Präfixsequenzen <<-nnn< und <<+nnn< ist nnn eine ein-, zwei- oder dreistellige dezimale Partitionsnummer, die mit dem Wert *knotennummer* in der Datei *db2nodes.cfg* übereinstimmen muss.

Anmerkung: Präfixsequenzen werden als Teil des Befehls behandelt. Wenn Sie Präfixsequenzen angeben, müssen Sie den gesamten Befehl, einschließlich der Präfixsequenzen, in doppelte Anführungszeichen setzen.

Angeben der Liste von Maschinen

Standardmäßig wird die Liste der Maschinen der Knotenkonfigurationsdatei *db2nodes.cfg* entnommen. Dies kann auf folgende Weise geändert werden:

- Angeben eines Pfadnamens zu der Datei, die die Liste der Maschinen enthält, indem die Umgebungsvariable *RAHOSTFILE* definiert (d. h. auf Plattformen unter UNIX exportiert bzw. unter Windows NT gesetzt) wird.
- Explizites Angeben der Liste als Zeichenfolge durch Leerzeichen getrennter Namen, indem die Umgebungsvariable *RAHOSTLIST* definiert (d. h. auf Plattformen unter UNIX exportiert bzw. unter Windows NT gesetzt) wird.

Anmerkung: Wenn beide dieser Umgebungsvariablen angegeben werden, erhält *RAHOSTLIST* den Vorrang.

Anmerkung: Unter Windows NT sollten Sie die Knotenkonfigurationsdatei auf *keinen* Fall manuell editieren, um keine Inkonsistenzen zu verursachen. Zum Abrufen der Liste der Maschinen im Exemplar verwenden Sie den Befehl **db2nlist**. Weitere Einzelheiten finden Sie im Handbuch *DB2 Enterprise - Extended Edition für Windows Einstieg*.

Eliminieren mehrfacher Einträge aus der Liste der Maschinen

Wenn Sie mit DB2 Enterprise - Extended Edition und mehreren logischen Knoten (Datenbankpartitionsservern) auf einer Maschine arbeiten, enthält Ihre Datei `db2nodes.cfg` mehrere Einträge für diese Maschine. In diesem Fall muss dem Befehl **rah** mitgeteilt werden, ob Sie den Befehl nur einmal auf jeder Maschine oder einmal für jeden in der Datei `db2nodes.cfg` aufgeführten logischen Knoten ausführen wollen. Verwenden Sie den Befehl **rah**, wenn Sie Maschinen angeben wollen. Verwenden Sie den Befehl **db2_all**, wenn Sie logische Knoten angeben wollen.

Anmerkung: Wenn Sie auf Plattformen unter UNIX Maschinen angeben, eliminiert der Befehl **rah** die doppelten Einträge normalerweise aus der Liste der Maschinen. Jedoch gibt es folgende Ausnahme: Wenn Sie logische Knoten angeben, stellt der Befehl **db2_all** Ihrem Befehl folgende Zuordnung voran:

```
export DB2NODE=nnn (für Korn-Shell-Syntax)
```

Dabei ist *nnn* die Knotennummer, die der entsprechenden Zeile in der Datei `db2nodes.cfg` entnommen ist, so dass der Befehl an den gewünschten Datenbankpartitions-Server weitergeleitet wird.

Bei der Angabe logischer Knoten können Sie die Liste mit Hilfe der Präfixsequenzen `<<-nnn<` und `<<+nnn<` auf alle logischen Knoten außer einem oder auf nur einen Datenbankpartitions-Server beschränken. Diese Möglichkeit kann nützlich sein, wenn Sie einen Befehl zuerst auf dem Katalogknoten ausführen wollen und nach Abschluss dieses Befehls denselben Befehl in allen anderen Datenbankpartitionsservern eventuell parallel ausführen wollen. Normalerweise ist diese Methode für die Ausführung des Befehls **db2 restart database** erforderlich. Für diesen Fall müssen Sie die Knotennummer des Katalogknotens kennen. Informationen zu den Präfixsequenzen finden Sie in „Präfixsequenzen“ auf Seite 417.

Wenn Sie den Befehl **db2 restart database** mit Hilfe des Befehls **rah** ausführen, werden doppelte Einträge aus der Liste der Maschinen eliminiert. Wenn Sie aber das Präfix `"` angeben, werden mehrfache Einträge nicht eliminiert, da angenommen wird, dass die Verwendung des Präfixes `"` das Senden der Befehle an alle Datenbankpartitions-Server und nicht an alle Maschinen impliziert.

Steuern des Befehls rah

Mit Hilfe der folgenden Umgebungsvariablen können Sie den Befehl **rah** steuern.

Tabelle 23.

Name	Bedeutung	Standardwert
\$RAHBUFDIR Anmerkung: Nur für auf UNIX basierende Plattformen verfügbar.	Verzeichnis für den Puffer	/tmp/\$USER
\$RAHBUFNAME Anmerkung: Nur für auf UNIX basierende Plattformen verfügbar.	Dateiname für den Puffer	rahout
\$RAHOSTFILE (auf UNIX basierende Plattformen); RAHOSTFILE (Windows NT)	Datei mit der Liste der Hosts	db2nodes.cfg
\$RAHOSTLIST (auf UNIX basierende Plattformen); RAHOSTLIST (Windows NT)	Liste der Hosts in Form einer Zeichenfolge	extrahiert aus \$RAHOSTFILE
\$RAHCHECKBUF Anmerkung: Nur für auf UNIX basierende Plattformen verfügbar.	Wenn auf "no", Prüfungen übergehen	Nicht definiert
\$RAHSLEEPTIME (auf UNIX basierende Plattformen); RAHSLEEPTIME (Windows NT)	Zeit in Sekunden, die diese Prozedur auf die erste Ausgabe parallel ausgeführter Befehle wartet	86400 Sekunden für db2_kill , 200 Sekunden für alle anderen

Tabelle 23. (Forts.)

Name	Bedeutung	Standardwert
\$RAHWAITTIME (auf UNIX basierende Plattformen); RAHWAITTIME (Windows NT)	<p>Unter Windows NT Intervall in Sekunden zwischen aufeinander folgenden Prüfungen darauf, ob ferne Jobs noch aktiv sind.</p> <p>Auf Plattformen unter UNIX das Intervall in Sekunden zwischen aufeinander folgenden Prüfungen darauf, ob ferne Jobs noch aktiv sind, und ob Nachrichten der Form rah: waiting for <pid> ... vorhanden sind.</p> <p>Bei allen Plattformen ist eine positive ganze Zahl anzugeben. Stellen Sie dem Wert eine führende Null voran, um Nachrichten zu unterdrücken. Beispiel: export RAHWAITTIME=045.</p> <p>Es ist nicht notwendig, einen niedrigen Wert anzugeben, da der Befehl rah bei der Feststellung von Jobbeendigungen nicht auf diese Prüfungen angewiesen ist.</p>	45 Sekunden
\$RAHENV Anmerkung: Nur für auf UNIX basierende Plattformen verfügbar.	Gibt den Dateinamen an, der auszuführen ist, wenn \$RAHDOTFILES=E oder K oder PE oder B ist.	\$ENV
\$RAHUSER (auf UNIX basierende Plattformen); RAHUSER (Windows NT)	<p>Auf Plattformen unter UNIX die Benutzer-ID, unter der der ferne Befehl auszuführen ist.</p> <p>Unter Windows NT das Anmeldekonto, das dem DB2-Service für ferne Befehle (Remote Command Service) zugeordnet ist.</p>	\$USER

Anmerkung: Auf Plattformen unter UNIX wird der Wert von \$RAHENV von dort, wo **rah** ausgeführt wird, verwendet, nicht der Wert (falls vorhanden), der von der fernen Shell gesetzt wurde.

\$RAHDOTFILES auf Plattformen unter UNIX

Anmerkung: Die Informationen dieses Abschnitts beziehen sich ausschließlich auf Plattformen, die auf UNIX basieren.

Im Folgenden werden die Punktdateien (·) beschrieben, die ausgeführt werden, wenn keine Präfixsequenz angegeben wird:

P	·profile
E	Datei, die in \$RAHENV definiert ist (wahrscheinlich .kshrc)
K	Wie E
PE	·profile gefolgt von der Datei, die in \$RAHENV definiert ist (wahrscheinlich .kshrc)
B	Wie PE
N	Keine (weder noch)

Anmerkung: Wenn Ihre Anmelde-Shell keine Korn-Shell ist, werden alle Punktdateien, die Sie zur Ausführung angeben, in einem Korn-Shell-Prozess ausgeführt und müssen daher der Korn-Shell-Syntax entsprechen. Wenn Ihre Anmelde-Shell beispielsweise eine C-Shell ist, dann haben Sie zwei Möglichkeiten, Ihre .cshrc-Umgebung zur Ausführung von Befehlen durch **rah** einzurichten. Entweder erstellen Sie ein Äquivalent der Korn-Shell-Datei `INSHOME/.profile` zu Ihrer Datei `.cshrc` und geben in Ihrer Datei `INSHOME/.cshrc` Folgendes an:

```
setenv RAHDOTFILES P
```

Alternativ hierzu können Sie für die Datei `.cshrc` in der Korn-Shell ein Äquivalent `INSHOME/.kshrc` erstellen und in der Datei `INSHOME/.cshrc` Folgendes angeben:

```
setenv RAHDOTFILES E
setenv RAHENV INSHOME/.kshrc
```

Außerdem ist es wichtig, dass Ihre `.cshrc` die Ausgabe nicht an die Standardausgabeeinheit (`stdout`) leitet, wenn (wie beim Aufruf durch **rsh**) kein nicht grafikfähiges Terminal vorhanden ist. Dies können Sie sicherstellen, indem Sie alle Zeilen, die an `stdout` schreiben, zum Beispiel folgendermaßen einschließen:

```
if { tty -s } then echo "executed .cshrc";
endif
```

Definieren des Standardumgebungsprofils unter Windows NT

Anmerkung: Die Informationen dieses Abschnitts beziehen sich ausschließlich auf Windows NT.

Zum Definieren des Standardumgebungsprofils für den Befehl **rah** wird eine Datei namens `db2rah.env` verwendet, die im Exemplarverzeichnis zu erstellen ist. Die Datei sollte folgendes Format aufweisen:

```
; Dies ist eine Kommentarzeile
DB2INSTANCE=exemplarname
DB2DBDFT=datenbank
; Dateiende
```

Sie können alle Umgebungsvariablen angeben, die Sie zur Initialisierung der Umgebung für den Befehl **rah** benötigen.

Bestimmung von Fehlern mit rah auf Plattformen unter UNIX

Anmerkung: Die Informationen dieses Abschnitts beziehen sich ausschließlich auf Plattformen, die auf UNIX basieren.

Im Folgenden werden Maßnahmen zur Behandlung einiger Fehler diskutiert, die bei der Ausführung von **rah** eventuell auftreten können:

1. **rah** blockiert (oder benötigt sehr lange Zeit)

Dieses Problem kann folgende Ursache haben:

- **rah** hat festgestellt, dass Ausgaben gepuffert werden müssen und `RAHCHECKBUF=no` nicht exportiert wurde. Daher sendet **rah** vor der Ausführung Ihres Befehls einen Befehl an alle Maschinen, um die Existenz des Pufferverzeichnisses zu überprüfen und es zu erstellen, falls es nicht bereits existiert.
- Eine oder mehrere der Maschinen, an die Sie Ihren Befehl senden, reagiert nicht. Der Befehl **rsh** wird schließlich sein Zeitlimit überschreiten. Das Zeitlimit ist in der Regel jedoch recht lang (in der Regel ungefähr 60 Sekunden).

2. Sie haben Nachrichten wie die folgenden empfangen:

- Anmeldung nicht korrekt (Login incorrect)
- Berechtigung verweigert (Permission denied)

Entweder ist auf einer der Maschinen die ID, auf der **rah** ausgeführt wird, nicht korrekt in der zugehörigen Datei `.hosts` definiert oder auf der ID, auf der **rah** ausgeführt wird, ist eine der Maschinen nicht korrekt in der Datei `.rhosts` definiert.

3. Obwohl bei der parallelen Ausführung von Befehlen mit rshell-Hintergrundprozessen die Befehle ausgeführt werden und in der erwarteten abgelaufenen Zeit auf den Hosts abgeschlossen werden, braucht **rah** eine lange Zeit, um dies zu erkennen und die Shell-Eingabeaufforderung anzuzeigen.

Auf der ID, auf der **rah** ausgeführt wird, ist eine der Maschinen nicht korrekt in der zugehörigen Datei `.rhosts` definiert.

4. **rah** läuft bei Ausführung über die Shell-Befehlszeile einwandfrei. Aber wenn **rah** fern mit Hilfe von rsh, zum Beispiel mit folgendem Befehl ausgeführt wird:

```
rsh somewhere -l $USER db2_kill
```

wird **rah** niemals beendet.

Dies ist normal. Der Befehl **rah** startet Überwachungsprozesse im Hintergrund, die auch nach der Beendigung weiterhin aktiv sind. Diese Prozesse bleiben in der Regel bestehen, bis alle Prozesse, die zu dem Befehl gehören, den Sie ausgeführt haben, ihrerseits beendet sind. Im Fall von **db2_kill** bedeutet dies, dass sie bis zur Beendigung aller Datenbankmanager bestehen bleiben. Sie können die Überwachungsprozesse beenden, indem Sie den Prozess auffinden, dessen Befehl **rahwaitfor** ist, und den Befehl `kill <prozess_id>` ausführen. Geben Sie keine Signalnummer an. Verwenden Sie stattdessen den Standardwert (15).

5. Die Ausgabe von **rah** wird nicht korrekt angezeigt, oder **rah** meldet fälschlicherweise, dass `$RAHBUFNAME` nicht existiert, wenn mehrere Befehle von **rah** unter derselben Benutzer-ID `$RAHUSER` abgesetzt wurden.

Dies kann eintreten, weil mehrere gleichzeitig ablaufende Ausführungen von **rah** versuchen, dieselbe Pufferdatei (z. B.

`$RAHBUFDIR/$RAHBUFNAME`) zu verwenden. Zur Vermeidung dieses Problems verwenden Sie für jeden gleichzeitig ablaufenden Befehl **rah** einen unterschiedlichen Namen für `$RAHBUFNAME`. Im Folgenden ist ein ksh-Beispiel aufgeführt:

```
export RAHBUFNAME=rahout
rah ";$befehl_1" &
export RAHBUFNAME=rah2out
rah ";$befehl_2" &
```

Alternativ hierzu können Sie eine Methode verwenden, mit der die Shell automatisch einen eindeutigen Namen auswählt. Beispiel:

```
RAHBUFNAME=rahout.$$ db2_a11 "....."
```

Unabhängig von der verwendeten Methode müssen Sie sicherstellen, dass die Pufferdateien zu einem bestimmten Zeitpunkt bereinigt werden, wenn der Plattenspeicherplatz begrenzt ist. Der Befehl **rah** löscht keine Pufferdatei am Ende der Ausführung. Eine vorhandene Pufferdatei wird jedoch beim nächsten Mal gelöscht und wiederverwendet, wenn Sie dieselbe Pufferdatei erneut angeben.

6. Sie haben Folgendes eingegeben:

```
rah "print from ()'
```

Daraufhin haben Sie die folgende Nachricht empfangen:

```
ksh: syntax error at line 1 : (' unexpected
(ksh: Syntaxfehler in Zeile 1 : (' unerwartet)
```

Voraussetzungen für die Ersetzung von () und ## sind:

- Es ist **db2_all** und nicht **rah** zu verwenden.
- Stellen Sie sicher, dass eine RAHOSTFILE-Datei verwendet wird. Dies geschieht entweder durch Exportieren der Umgebungsvariablen RAHOSTFILE oder durch die sich standardmäßig ergebene Verwendung der Datei /sql1lib/db2nodes.cfg. Ohne diese Vorbedingungen lässt **rah** die Zeichenfolgen () und ## unverändert. Sie empfangen einen Fehler, weil der Befehl **print from ()** ungültig ist.

Zur Verbesserung der Leistung bei der parallelen Ausführung der Befehle sollten Sie | anstelle von |& und || anstelle von ||& oder ; verwenden, es sei denn, die von & bereitgestellte Funktion wird unbedingt benötigt. Durch die Angabe von & werden mehr **rsh**-Befehle erforderlich und daher die Leistung herabgesetzt.

Anhang E. Arbeitsweise von DB2 für Windows NT mit der Windows NT-Sicherheit

Bei der Installation ermöglicht Windows NT die Erstellung zweier Benutzernamen für Administratoren:

- Der eine heißt „Administrator“.
- Der andere ist ein Name Ihrer Wahl. Er muss über die Administratorberechtigung verfügen und den DB2-Namenskonventionen entsprechen. Weitere Informationen zu den DB2-Namenskonventionen finden Sie in „Anhang A. Namenskonventionen“ auf Seite 369.

Der Benutzer kann sich an der lokalen Maschine anmelden, oder er kann sich, wenn die Maschine in einer Windows NT-Domäne installiert ist, an der Domäne anmelden. DB2 für Windows NT unterstützt beide Möglichkeiten. Zur Authentifizierung des Benutzers prüft DB2 zuerst die lokale Maschine, dann die Domänensteuereinheit für die aktuelle Domäne und schließlich alle der Domänensteuereinheit bekannten akzeptierten Domänen („Trusted Domains“).

Um dies zu verdeutlichen, nehmen Sie an, dass das DB2-Exemplar die Authentifizierung auf dem Server voraussetzt.

Die Konfiguration sieht folgendermaßen aus:

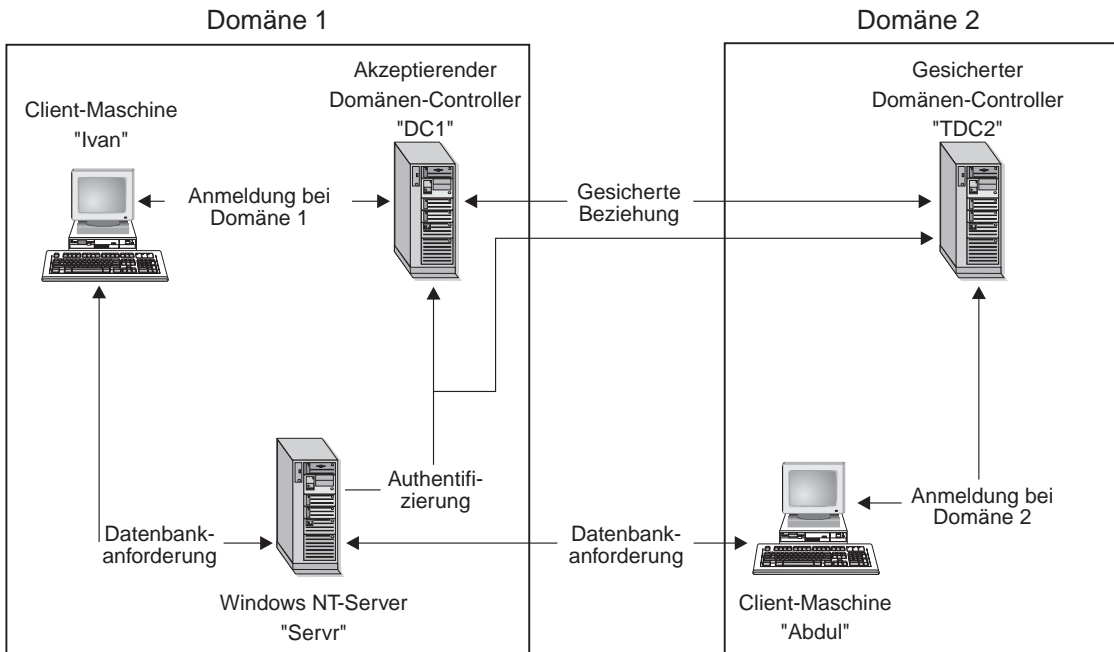


Abbildung 11. Authentifizierung in Windows NT-Domänen

Jede Maschine verfügt über eine SAM-Sicherheitsdatenbank (Security Access Management), sofern eine Client-Maschine nicht unter Windows 9x betrieben wird. Windows 9x-Maschinen haben keine SAM-Datenbank. DC1 ist die Domänensteuereinheit (Domain Controller), in der die Client-Maschine Ivan und Servr, der Server mit DB2 für Windows NT, registriert sind. TDC2 ist für DC1 eine akzeptierte ("Trusted") Domäne, und die Client-Maschine Abdul ist ein Teil der Domäne von TDC2.

Ein Beispielszenario mit Authentifizierung auf dem Server:

1. Abdul meldet sich an der Domäne TDC2 an (d. h. dieser Client ist in der SAM-Datenbank von TDC2 registriert).
2. Abdul stellt dann die Verbindung zu einer DB2-Datenbank her, die laut Katalog auf SRV3 gespeichert ist:
`db2 connect to remotedb user Abdul using fredpw`
3. SRV3 stellt fest, wo Abdul bekannt ist. Die API, die zum Abrufen dieser Informationen eingesetzt wird, durchsucht zunächst die lokale Maschine (SRV3) und anschließend die Domänensteuereinheit (DC1), bevor sie mit anderen akzeptierten Domänen fortfährt. Der Benutzername Abdul wird

auf TDC2 gefunden. Diese Suchreihenfolge setzt einen einzigen Namensbereich für Benutzer und Gruppen voraus.

4. SRV3 unternimmt dann folgende Schritte:
 - a. Er überprüft den Benutzernamen und das Kennwort mit Hilfe von TDC2.
 - b. Er ermittelt, ob Abdul ein Administrator ist, indem er TDC2 befragt.
 - c. Er zählt alle Gruppen von Abdul auf, indem er TDC2 befragt.

Ein Beispielszenario mit Authentifizierung auf dem Client und einer Windows NT-Client-Maschine:

1. Dale, der Administrator, meldet sich an SRV3 an und ändert die Authentifizierungsart für das Datenbankexemplar in Client:

```
db2 update dbm cfg using authentication client
db2stop myinst
db2start myinst
```
2. Ivan meldet sich von der Windows-Client-Maschine aus an der Domäne DC1 an (d. h., er ist in der SAM-Datenbank von DC1 registriert).
3. Ivan stellt dann die Verbindung zu einer DB2-Datenbank her, die laut Katalog auf SRV3 gespeichert ist:

```
DB2 CONNECT to remotedb user Ivan using johnpw
```
4. Ivans Maschine überprüft den Benutzernamen und das Kennwort. Die API, die zum Abrufen dieser Informationen eingesetzt wird, durchsucht zunächst die lokale Maschine (Ivan) und anschließend die Domänensteuer Einheit (DC1), bevor sie mit anderen akzeptierten Domänen fortfährt. Der Benutzername Ivan wird auf DC1 gefunden.
5. Ivans Maschine überprüft dann den Benutzernamen und das Kennwort mit Hilfe von DC1.
6. SRV3 unternimmt dann folgende Schritte:
 - a. Er stellt fest, wo Ivan registriert ist.
 - b. Er ermittelt, ob Ivan ein Administrator ist, indem er DC1 befragt.
 - c. Er zählt alle Gruppen von Ivan auf, indem er DC1 befragt.

Anmerkung: Stellen Sie sicher, dass der DB2-Sicherheitsservice gestartet wurde, bevor Sie versuchen, eine Verbindung zur DB2-Datenbank herzustellen. Der Sicherheitsdienst wird von DB2 installiert und für die Ausführung als Windows NT-Dienst konfiguriert. Er wird jedoch nicht automatisch gestartet. Geben Sie den Befehl `NET START DB2NTSECSERVER` ein, um den DB2-Sicherheitsdienst zu starten.

Ein Beispielszenario mit Authentifizierung auf dem Client und einer Windows 95-Client-Maschine:

1. Dale, der Administrator, meldet sich an SRV3 an und ändert die Authentifizierungsart für das Datenbanke Exemplar in Client:

```
db2 update dbm cfg using authentication client
db2stop myinst
db2start myinst
```
2. Ivan meldet sich von einer Windows 95-Client-Maschine aus an der Domäne DC1 an (d. h., er ist in der SAM-Datenbank von DC1 registriert).
3. Ivan stellt dann die Verbindung zu einer DB2-Datenbank her, die laut Katalog auf SRV3 gespeichert ist:

```
db2 connect to remotedb user Ivan using johnpw
```
4. Ivans Windows 95-Maschine kann den Benutzernamen und das Kennwort nicht überprüfen. Der Benutzername und das Kennwort werden daher als gültig angenommen.
5. SRV3 unternimmt dann folgende Schritte:
 - a. Er stellt fest, wo Ivan registriert ist.
 - b. Er ermittelt, ob Ivan ein Administrator ist, indem er DC1 befragt.
 - c. Er zählt alle Gruppen von Ivan auf, indem er DC1 befragt.

Anmerkung: Da ein Windows 95-Client einen verwendeten Benutzernamen und das Kennwort nicht überprüfen kann, ist eine Authentifizierung auf dem Client unter Windows 95 nicht sicher. Wenn die Windows 95-Maschine Zugriff auf einen Windows NT-Sicherheitsanbieter hat, kann ein gewisses Maß an Sicherheit erreicht werden, indem das Windows 95-System für überprüfte Durchlassanmeldungen konfiguriert wird. Einzelheiten zur Konfiguration eines Windows 95-System in dieser Weise finden Sie in der Microsoft-Dokumentation zu Windows 95.

DB2 unterstützt auch globale Gruppen. Zur Verwendung globaler Gruppen müssen Sie globale Gruppen innerhalb einer lokalen Gruppe hinzufügen, die sich auf dem Sicherheitsserver befindet. Wenn DB2 alle Gruppen aufzählt, deren Mitglied eine Person ist, führt DB2 indirekt auch die lokalen Gruppen auf, deren Mitglied der Benutzer ist (dadurch, dass er in einer globalen Gruppe ist, die selbst wiederum zu einer oder mehreren lokalen Gruppen gehört).

Verwenden einer Sicherungs-Domänensteuereinheit mit DB2

Wenn der von Ihnen für DB2 verwendete Server auch als Sicherungs-Domänensteuereinheit fungiert, können Sie die DB2-Leistung erhöhen und den Datenverkehr auf dem Netzwerk verringern, wenn Sie DB2 so konfigurieren, dass DB2 die Sicherungs-Domänensteuereinheit verwendet.

Die Sicherungs-Domänensteuereinheit wird für DB2 mit Hilfe der Registrierungsvariablen *DB2DMNBCKCTRL* definiert.

Wenn Sie den Namen der Domäne kennen, für die der DB2-Server die Sicherungs-Domänensteuereinheit ist, geben sie Folgendes an:

```
db2dmbckctlr=DOMÄNENNAME
```

Dabei muss *DOMÄNENNAME* in Großbuchstaben angegeben werden.

Wenn DB2 die Domäne bestimmen soll, für die die lokale Maschine eine Sicherungs-Domänensteuereinheit ist, geben Sie Folgendes an:

```
DB2DMNBCKCTRL=?
```

Anmerkung: DB2 verwendet eine vorhandene Sicherungs-Domänensteuereinheit nicht standardmäßig, weil die Synchronisierung einer Sicherungs-Domänensteuereinheit mit der primären Domänensteuereinheit verloren gehen und so eine Sicherheitsbeeinträchtigung verursachen könnte. Domänensteuereinheiten können ihre Synchronisierung verlieren, wenn die Sicherheitsdatenbank der primären Domänensteuereinheit aktualisiert wird, aber die Änderungen nicht an eine Sicherungs-Domänensteuereinheit weitergegeben werden. Dies kann eintreten, wenn es Netzlatenzzeiten gibt oder wenn der Browser-Dienst des Computers nicht läuft.

Benutzerauthentifizierung mit DB2 für Windows NT

Die Benutzerauthentifizierung kann für Windows NT-Benutzer aufgrund der Art und Weise, in der das Betriebssystem die Authentifizierung durchführt, problematisch sein. Im vorliegenden Abschnitt werden einige Faktoren erläutert, die für die Benutzerauthentifizierung unter DB2 für Windows NT relevant sind:

- „Einschränkungen bei Benutzername und Gruppenname“
- „Sicherheitsservice von DB2 für Windows NT“
- „Installieren von DB2 auf einem Sicherungsdomänen-Controller“ auf Seite 435
- „Authentifizierung mit Gruppen und Domänensicherheit“ auf Seite 436

Einschränkungen bei Benutzername und Gruppenname

Im Folgenden sind die für die Umgebung geltenden Einschränkungen aufgeführt:

- Die Länge von Benutzernamen ist unter DB2 auf maximal 30 Zeichen beschränkt. Für Gruppennamen sind maximal 8 Zeichen zulässig.
- Bei unter Windows NT verwendeten Benutzernamen muss die Groß-/Kleinschreibung nicht beachtet werden, jedoch bei den zugehörigen Kennwörtern.
- Benutzernamen und Gruppennamen können aus einer Kombination von Groß- und Kleinbuchstaben eingegeben werden. Allerdings werden sie für die Verwendung in DB2 normalerweise in Großbuchstaben umgesetzt. Wenn Sie z. B. eine Verbindung zu einer Datenbank herstellen und die Tabelle `schema1.table1` erstellen, wird diese Tabelle unter dem Namen `SCHEMA1.TABLE1` in der Datenbank gespeichert. (Wenn Sie Objektnamen in Kleinbuchstaben verwenden wollen, müssen Sie über den Befehlszeilenprozessor entsprechende Befehle absetzen. Hierbei müssen die Objektnamen in Anführungszeichen eingeschlossen werden. Alternativ können Sie auch ODBC-Front-End-Tools anderer Hersteller verwenden.)

Sicherheitsservice von DB2 für Windows NT

Unter DB2 Universal Database wurde die Funktion zum Authentifizieren von Benutzernamen und Kennwörtern in den DB2-System-Controller integriert. Der Sicherheitsservice wird nur für die Verbindung eines Clients zu einem Server benötigt, für den die Authentifizierungsart `CLIENT` konfiguriert wurde.

Installieren von DB2 auf einem Sicherungsdomänen-Controller

In einer Windows NT-Umgebung kann ein Benutzer entweder auf einem Primär- oder einem Sicherungs-Controller authentifiziert werden. Diese Funktion ist in umfangreichen, verteilten LANs mit einem zentralen Primärdomänen-Controller und einem oder mehreren Sicherungsdomänen-Controllern (BDC = Backup Domain Controller) an allen Standorten sehr wichtig. Die Benutzer können dann auf dem Sicherungsdomänen-Controller ihres Standortes authentifiziert werden. Auf diese Weise ist es nicht erforderlich, einen Aufruf zur Authentifizierung an den Primärdomänen-Controller (PDC) abzusetzen.

Der Vorteil von Sicherungsdomänen-Controllern liegt in diesem Fall in der schnelleren Authentifizierung der Benutzer und der geringeren Auslastung des LANs im Vergleich zu einer Konfiguration ohne Sicherungsdomänen-Controller (BDC).

Unter folgenden Bedingungen ist eine BDC-Authentifizierung möglich:

- Der DB2 für Windows NT-Server wurde auf dem Sicherungsdomänen-Controller installiert.
- Die Variable DB2DMNBCKCTRL der Profilregistrierdatenbank wurde auf einen geeigneten Wert gesetzt.

Wurde die Variable DB2DMNBCKCTRL der Profilregistrierdatenbank nicht gesetzt oder wurde ein Nullwert angegeben, führt DB2 für Windows NT die Authentifizierung auf dem Primärdomänen-Controller aus.

Die einzigen gültigen, deklarierten Einstellungen für DB2DMNBCKCTRL sind „?“ oder ein Domänenname.

Wenn für die Variable DB2DMNBCKCTRL der Profilregistrierdatenbank ein Fragezeichen (DB2DMNBCKCTRL=?) angegeben wurde, führt DB2 für Windows NT die Authentifizierung unter folgenden Bedingungen auf dem Sicherungsdomänen-Controller aus:

- Der Registrierungswert `cachedPrimaryDomain` ist auf den Namen der Domäne gesetzt, zu der diese Maschine gehört. (Sie finden diese Einstellung unter **HKEY_LOCAL_MACHINE** → **Software** → **Microsoft** → **Windows NT** → **Current Version** → **WinLogon**.)
- Der Server-Manager zeigt an, dass der Sicherungsdomänen-Controller aktiv und verfügbar ist. (D. h., das Symbol für diese Maschine wird nicht abgeblendet dargestellt.)
- Die Registrierdatenbank für den DB2 Windows NT-Server zeigt an, dass das System ein Sicherungsdomänen-Controller in einer angegebenen Domäne ist.

Unter normalen Bedingungen kann die Einstellung `DB2DMNBCKCTLR=?` verwendet werden, dies ist jedoch nicht in allen Umgebungen der Fall. Die über die in der Domäne verfügbaren Server bereitgestellten Informationen sind dynamisch und der Computer-Browser muss aktiv sein, um diese Informationen korrekt und auf dem neuesten Stand zu halten. Auf umfangreichen LANs wird der Computer-Browser möglicherweise nicht ausgeführt, so dass die Server-Manager-Informationen eventuell nicht aktuell sind. In diesem Fall gibt es ein zweites Verfahren, mit dem DB2 für Windows NT angewiesen werden kann, die Authentifizierung auf dem Sicherungsdomänen-Controller auszuführen: Setzen Sie `DB2DMNBCKCTLR=xxx`, wobei `xxx` für den Windows NT-Domänennamen des DB2-Servers steht. Mit dieser Einstellung wird die Authentifizierung auf dem Sicherungsdomänen-Controller auf der Basis der folgenden Bedingungen ausgeführt:

- Der Registrierungswert `cachedPrimaryDomain` ist auf den Namen der Domäne gesetzt, zu der diese Maschine gehört. (Sie finden diese Einstellung unter `HKEY_LOCAL_MACHINE` → **Software** → **Microsoft** → **Windows NT** → **Current Version** → **WinLogon**.)
- Die Maschine ist als Sicherungsdomänen-Controller für die angegebene Domäne konfiguriert. (Wenn die Maschine als Sicherungsdomänen-Controller für eine andere Domäne konfiguriert ist, wird durch diese Einstellung ein Fehler erzeugt.)

Authentifizierung mit Gruppen und Domänensicherheit

DB2 für Windows NT unterstützt die folgenden Gruppenarten:

- Lokale Gruppen
- Globale Gruppen
- Globale Gruppen als Member einer lokalen Gruppe

DB2 für Windows NT spezifiziert die lokalen und globalen Gruppen, zu denen der Benutzer gehört, und verwendet hierzu die Sicherheitsdatenbank, in der der Benutzer lokalisiert wurde. DB2 Universal Database stellt eine Überschreibungsfunktion zur Verfügung, durch den die Ausführung der Gruppenspezifizierung auf dem lokalen Windows NT-Server erzwungen wird, auf dem DB2 installiert ist. Hierbei spielt es keine Rolle, wo der Benutzereintrag lokalisiert wurde. Diese Überschreibung kann mit den folgenden Befehlen veranlasst werden:

- Für globale Einstellungen:
`db2set -g DB2_GRP_LOOKUP=local`
- Für Exemplareinstellungen:
`db2set -i DB2_GRP_LOOKUP=local`

Geben Sie Folgendes ein, um alle definierten Variablen der DB2-Profilregistrierdatenbank anzuzeigen:

```
db2set -all
```


Um unter DB2 für Windows NT mit der Funktion für die Domänensicherheit arbeiten zu können, müssen Sie einer lokalen Gruppe die entsprechenden Berechtigungen und Zugriffsrechte erteilen. Benutzernamen innerhalb der lokalen und globalen Gruppen MÜSSEN in derselben Domäne definiert sein wie die lokale oder globale Gruppe, damit eine korrekte Authentifizierung gewährleistet ist.

Wenn die Variable DB2_GRP_LOOKUP der Profilregistrierdatenbank auf den Wert 'local' gesetzt ist, versucht DB2 den Benutzer nur auf der lokalen Maschine zu finden. Wird der Benutzer auf der lokalen Maschine nicht gefunden oder ist er nicht als Mitglied einer lokalen oder globalen Gruppe definiert, schlägt die Authentifizierung fehl. DB2 versucht **nicht**, den Benutzer auf einer anderen Maschine in der Domäne oder auf den verfügbaren Domänen-Controllern zu finden.

Wenn die Variable DB2_GRP_LOOKUP der Profilregistrierdatenbank nicht gesetzt wurde, gilt Folgendes:

1. DB2 versucht als erstes, den Benutzer auf der selben Maschine zu finden.
2. Wurde der Benutzername lokal definiert, wird auch die Authentifizierung lokal ausgeführt.
3. Wurde der Benutzer auf dem lokalen System nicht gefunden, versucht DB2, den Benutzernamen in der zugehörigen Domäne und anschließend in den gesicherten Domänen zu lokalisieren.

Die folgenden Beispiele erläutern, wie DB2 für Windows NT die Domänensicherheit unterstützen kann. In diesem ersten Beispiel funktioniert die Verbindung problemlos, weil der Benutzername und die lokale Gruppe sich in derselben Domäne befinden. Im zweiten Beispiel schlägt die Verbindung fehl, weil der Benutzername und die lokale oder globale Gruppe sich in unterschiedlichen Domänen befinden.

Beispiel für eine erfolgreiche Verbindung: Die Verbindung im folgenden Szenario funktioniert, weil der Benutzername und die lokale oder globale Gruppe sich in derselben Domäne befinden.

Beachten Sie hierbei, dass der Benutzername und die lokale oder globale Gruppe nicht in der Domäne definiert sein müssen, in der der Datenbankserver ausgeführt wird. Sie müssen sich jedoch in derselben Domäne befinden.

Tabelle 24. Erfolgreiche Verbindung mit einem Domänencontroller

Domain1	Domain2
Zu Domain2 besteht eine gesicherte Beziehung.	<ul style="list-style-type: none"> • Zu Domain1 besteht eine gesicherte Beziehung. • Die lokale oder globale Gruppe grp2 wurde definiert. • Der Benutzername id2 wurde definiert. • Der Benutzername id2 gehört zu grp2.
Der DB2-Server wird in dieser Domäne ausgeführt. Die folgenden DB2-Befehle werden über diese Einheit abgesetzt: <pre>REVOKE CONNECT ON db FROM public GRANT CONNECT ON db TO GROUP grp2 CONNECT TO db USER id2</pre>	
Die lokale oder globale Domäne wird durchsucht, id2 kann jedoch nicht gefunden werden. Die Funktion für die Domänensicherheit wird durchsucht.	
	Der Benutzername id2 wird in dieser Domäne gefunden. DB2 erhält zusätzliche Informationen zu diesem Benutzernamen (der zu der Gruppe grp2 gehört).
Die Verbindung funktioniert, weil der Benutzername und die lokale oder globale Gruppe sich in derselben Domäne befinden.	

Beispiel für eine nicht erfolgreiche Verbindung: Die Verbindung im folgenden Szenario funktioniert nicht, weil der Benutzername in einer anderen Domäne definiert ist als die lokale oder globale Gruppe.

Tabelle 25. Nicht erfolgreiche Verbindung mit einem Domänen-Controller

Domain1	Domain2
Zu Domain2 besteht eine gesicherte Beziehung.	<ul style="list-style-type: none"> • Zu Domain1 besteht eine gesicherte Beziehung. • Die lokale oder globale Gruppe grp2 wurde definiert.
<ul style="list-style-type: none"> • Die globale Gruppe grp1 wurde definiert. • Der Benutzername id1 wurde definiert. • Der Benutzername id1 gehört zu grp1. 	
	Domäne 1\grp1 gehört zu grp2.
<p>Der DB2-Server wird in dieser Domäne ausgeführt. Die folgenden DB2-Befehle werden über diese Einheit abgesetzt:</p> <pre> REVOKE CONNECT ON db FROM public GRANT CONNECT ON db TO GROUP grp2 CONNECT TO db USER id2 </pre>	
Die lokale oder globale Gruppe wurde durchsucht und id1 wurde gefunden. DB2 erhält Informationen für diesen Benutzernamen (der Benutzername id1 gehört zu grp1 und grp1 gehört zu Domain2\grp2).	
	Die Gruppe grp2 ist in dieser Domäne definiert.
<p>Die Verbindung funktioniert nicht, weil die lokale oder globale Gruppe sich in Domain2 befindet und der aktuelle Benutzername in Domain1 definiert ist.</p> <p>Die Verbindung würde funktionieren, wenn statt dessen der folgende Befehl eingegeben würde:</p> <pre> GRANT CONNECT ON db TO GROUP grp1 </pre>	

Anhang F. Arbeiten mit dem Windows NT-Systemmonitor

Für Benutzer von DB2 für Windows NT gibt es zwei Möglichkeiten zur Leistungsüberwachung:

- **DB2 Performance Monitor**

Der DB2 Performance Monitor bietet Momentaufnahme- und Ereignisdaten nur für DB2 und DB2 Connect. (Weitere Informationen erhalten Sie, wenn Sie den Druckknopf **Hilfe** in der Steuerzentrale anklicken, um die Online-Hilfefunktion **Erste Schritte** aufzurufen.)

- **Windows NT-Systemmonitor**

Mit dem Windows NT-Systemmonitor können Sie die Datenbank- und Systemleistung überwachen und dabei Informationen von allen Leistungsdatenlieferanten abrufen, die im System registriert sind. Windows NT bietet auch Leistungsdaten zu allen Aspekten des Maschinenbetriebs, darunter:

- CPU-Auslastung
- Speicherauslastung
- Plattenaktivität
- Netzwerkaktivität

Registrieren von DB2 beim Windows NT-Systemmonitor

Das Konfigurationsprogramm registriert DB2 automatisch beim Windows NT-Systemmonitor.

Damit die Leistungsdaten von DB2 und DB2 Connect für den Windows NT-Systemmonitor verfügbar werden, müssen Sie die DLL für die Leistungszähler (Datenquellen) von DB2 für Windows NT registrieren. Dadurch können andere Windows NT-Anwendungen mit den Win32-Leistungs-APIs Leistungsdaten abrufen.

Geben Sie zum Installieren und Registrieren der Leistungszähler-DLL von DB2 für Windows NT (DB2Perf.DLL) beim Windows NT-Systemmonitor folgenden Befehl ein:

```
db2perfi -i
```

Durch das Registrieren der DLL wird ein neuer Schlüssel in der Option **Services** der Registrierungsdatenbank erstellt. Ein Eintrag gibt den Namen der DLL an, die die Zählerunterstützung bietet. Drei weitere Einträge geben Namen von Funktionen an, die in der DLL enthalten sind. Dies sind folgende Funktionen:

- **Open**
Wird aufgerufen, wenn die DLL zuerst vom System in einem Prozess geladen wird.
- **Collect**
Wird aufgerufen, um Leistungsinformationen von der DLL anzufordern.
- **Close**
Wird aufgerufen, wenn die DLL entladen wird.

Aktivieren des Fernzugriffs auf DB2-Leistungsinformationen

Wenn Ihre Workstation unter DB2 für Windows NT mit anderen Maschinen unter Windows NT vernetzt ist, können Sie die in diesem Abschnitt beschriebene Funktion verwenden.

Damit Windows NT-Leistungsobjekte von einer anderen Maschine unter DB2 für Windows NT angezeigt werden können, müssen Sie einen Administrator-Benutzernamen und ein Administrator-Kennwort in DB2 registrieren. (Der Standardbenutzername des Windows NT-Systemmonitors, **SYSTEM**, ist in DB2 ein reserviertes Wort und kann nicht verwendet werden.) Geben Sie folgenden Befehl ein, um den Namen zu registrieren:

```
db2perfr -r benutzername kennwort
```

Anmerkung: Der verwendete benutzername muss den Namenskonventionen von DB2 entsprechen.

Die Angaben benutzername und kennwort sind in einem Schlüssel in der Registrierungsdatenbank enthalten. Der Zugriff auf diese ist nur Administratoren und dem Konto SYSTEM gestattet. Die Daten sind codiert, um Sicherheitsprobleme beim Speichern eines Administratorkennworts in der Registrierungsdatenbank zu vermeiden.

Anmerkungen:

1. Sobald eine Kombination aus Benutzername und Kennwort in DB2 registriert wurde, melden sich selbst lokale Exemplare des Systemmonitors explizit mit diesem Benutzernamen und Kennwort an. Dies bedeutet, dass bei lokalen Sitzungen des Systemmonitors keine DB2-Leistungsdaten angezeigt werden, wenn die Anmeldung nicht mit den in DB2 registrierten Angaben übereinstimmen.
2. Die Kombination aus Benutzername und Kennwort muss gepflegt werden, damit die Werte für den Benutzernamen und das Kennwort den in der Sicherheitsdatenbank von Windows NT gespeicherten Werten entsprechen. Wenn der Benutzername oder das Kennwort in der Sicherheitsdatenbank von Windows NT geändert wird, muss auch die Kombination aus Benutzername und Kennwort für die ferne Leistungsüberwachung neu definiert werden.

3. Geben Sie den folgenden Befehl ein, um die Registrierung zurückzunehmen:

```
db2perfr -u <benutzername> <kennwort>
```

Anzeigen von Leistungswerten für DB2 und DB2 Connect

Wenn Sie Leistungswerte für DB2 und DB2 Connect mit dem Systemmonitor anzeigen wollen, wählen Sie einfach die Leistungszähler, deren Werte angezeigt werden sollen, im Feld **Diagramm erweitern** aus. Dieses Fenster zeigt eine Liste von Leistungsobjekten für die Leistungsdaten an. Wählen Sie ein Objekt aus, um eine Liste der Zähler anzuzeigen, die es bereitstellt.

Ein Leistungsobjekt kann auch mehrere Exemplare haben. Beispielsweise bietet das Objekt **Logische Platte** Zähler wie „% Lesezeit“ und „Bytes gelesen/s“. Es hat auch ein Exemplar (Instanz) für jedes logische Laufwerk in der Maschine, einschließlich „C:“ und „D:“.

Windows NT bietet die folgenden Leistungsobjekte:

- **DB2-Datenbankmanager (DB2 Database Manager)**

Dieses Objekt bietet allgemeine Informationen für ein Windows NT-Exemplar. Das überwachte DB2-Exemplar wird als Objektexemplar angezeigt.

Aus praktischen und Leistungsgründen können Sie Leistungsdaten jeweils nur von einem DB2-Exemplar abrufen. Das DB2-Exemplar, das im Systemmonitor angezeigt wird, wird von der Variablen `db2instance` der Profilregistrierdatenbank im Prozess des Systemmonitors gesteuert. Wenn mehrere DB2-Exemplare gleichzeitig ausgeführt werden und Sie von mehreren die Leistungsdaten anzeigen wollen, müssen Sie für jedes zu überwachende Exemplar eine separate Sitzung des Systemmonitors starten. Dabei muss `db2instance` auf den entsprechenden Wert für das zu überwachende DB2-Exemplar gesetzt sein.

Wenn Sie mit einem partitionierten Datenbanksystem arbeiten, können Leistungsdaten immer nur von einem Datenbankpartitionsserver (Knoten) abgerufen werden. Standardmäßig werden die Leistungsdaten für den Standardknoten (Knoten, dem der logische Anschluss 0 zugeordnet ist) angezeigt. Um die Leistungsdaten eines anderen Knotens anzuzeigen, müssen Sie eine separate Sitzung des Performance Monitor starten und für die Umgebungsvariable `DB2NODE` die Knotennummer des zu überwachenden Knotens definieren.

- **DB2-Datenbanken (DB2 Databases)**

Dieses Objekt bietet Informationen zu einer bestimmten Datenbank. Informationen sind für jede momentan aktive Datenbank verfügbar.

- **DB2-Anwendungen (DB2 Applications)**
Dieses Objekt bietet Informationen zu einer bestimmten DB2-Anwendung. Informationen sind für jede derzeit aktive DB2-Anwendung verfügbar.
- **DB2-DCS-Datenbanken (DB2 DCS Databases)**
Dieses Objekt bietet Informationen zu einer bestimmten DCS-Datenbank. Informationen sind für jede momentan aktive Datenbank verfügbar.
- **DB2-DCS-Anwendungen (DB2 DCS Applications)**
Dieses Objekt bietet Informationen zu einer bestimmten DB2-DCS-Anwendung. Informationen sind für jede derzeit aktive DB2-DCS-Anwendung verfügbar.

Welche dieser Objekte im Windows NT-Systemmonitor aufgelistet werden, hängt davon ab, was auf Ihrer Windows NT-Maschine installiert ist und welche Anwendungen aktiv sind. Wenn zum Beispiel DB2 UDB installiert ist und der Datenbankmanager gestartet wurde, wird das Objekt für den DB2-Datenbankmanager aufgeführt. Wenn auch einige DB2-Datenbanken und -Anwendungen auf dieser Maschine aktiv sind, werden auch die Objekte für DB2-Datenbanken und DB2-Anwendungen aufgelistet. Wenn Sie Ihr Windows NT-System als DB2 Connect-Gateway verwenden und einige DCS-Datenbanken und -Anwendungen aktiv sind, werden die Objekte für DB2-DCS-Datenbanken und DB2-DCS-Anwendungen aufgelistet.

Zugreifen auf ferne DB2-Leistungsinformationen

Die Aktivierung des Fernzugriffs auf DB2-Leistungsinformationen wurde bereits in einem früheren Abschnitt behandelt. Wählen Sie im Fenster **Diagramm erweitern** einen anderen Computer zur Überwachung aus. Dadurch wird eine Liste aller verfügbaren Leistungsobjekte auf diesem Computer angezeigt.

Damit Sie ein DB2-Leistungsobjekt auf einem fernen Computer überwachen können, muss die Stufe des auf diesem Computer installierten DB2 UDB- oder DB2 Connect-Codes Version 6 oder neuer sein.

Zurücksetzen von DB2-Leistungswerten

Wenn eine Anwendung die Überwachungs-APIs von DB2 aufruft, werden normalerweise kumulierte Werte seit dem Start des DB2-Servers zurückgegeben. Oft ist es jedoch sinnvoll, folgende Aktionen auszuführen:

- Zurücksetzen von Leistungswerten
- Ausführen eines Tests
- Erneutes Zurücksetzen der Werte
- Erneutes Ausführen des Tests

Verwenden Sie zum Zurücksetzen der Datenbankleistungswerte das Programm **db2perf**. Geben Sie folgenden Befehl ein:

```
db2perf
```

Standardmäßig werden dadurch die Leistungswerte für alle aktiven DB2-Datenbanken zurückgesetzt. Sie können jedoch auch eine Liste der Datenbanken angeben, die zurückgesetzt werden sollen. Sie können auch den Parameter `-d` verwenden, um anzugeben, dass Leistungswerte für DCS-Datenbanken zurückgesetzt werden sollen. Beispiel:

```
db2perf
db2perf dbalias1 dbalias2 ... dbaliasn

db2perf -d
db2perf -d dbalias1 dbalias2 ... dbaliasn
```

Das erste Beispiel setzt die Leistungswerte für alle aktiven DB2-Datenbanken zurück. Das zweite Beispiel setzt die Leistungswerte für bestimmte DB2-Datenbanken zurück. Das dritte Beispiel setzt die Leistungswerte für alle aktiven DB2-DCS-Datenbanken zurück. Das letzte Beispiel setzt die Leistungswerte für bestimmte DB2-DCS-Datenbanken zurück.

Das Programm **db2perf** setzt die Werte für ALLE Programme zurück, die derzeit auf Datenbankleistungsdaten für das entsprechende DB2-Serverexemplar zugreifen (d. h. das Exemplar in `db2instance` in der Sitzung, in der Sie **db2perf** ausführen).

Durch den Aufruf von **db2perf** werden auch die Werte zurückgesetzt, die bei jedem Fernzugriff auf DB2-Leistungsdaten mit dem Befehl **db2perf** angezeigt werden.

Anmerkung: Es gibt eine DB2-API, `sqlmrset`, mit der eine Anwendung nur die lokal angezeigten Werte für eine bestimmte Datenbank zurücksetzen kann. Weitere Informationen finden Sie im Handbuch *Administrative API Reference*.

Anhang G. Arbeiten mit Datenbankpartitionsservern unter Windows NT oder Windows 2000

Beim Ändern der Konfiguration in einer Windows NT- oder Windows 2000-Umgebung werden die hierbei erforderlichen Aufgaben mit speziellen Dienstprogrammen ausgeführt. In anderen Betriebssystemumgebungen werden die im Kapitel zum „Skalieren der Konfiguration durch das Hinzufügen von Prozessoren“ im Handbuch *Systemverwaltung: Optimierung* beschriebenen Methoden verwendet.

Im vorliegenden Anhang werden die folgenden Dienstprogramme vorgestellt:

- „Auflisten der Datenbankpartitionsserver eines Exemplars“
- „Hinzufügen eines Datenbankpartitionsservers zu einem Exemplar“ auf Seite 448
- „Ändern der Datenbankpartition“ auf Seite 450
- „Löschen einer Datenbankpartition aus einem Exemplar“ auf Seite 451

Auflisten der Datenbankpartitionsserver eines Exemplars

Unter Windows NT oder Windows 2000 können Sie mit dem Befehl **db2nlist** eine Liste der Datenbankpartitionsserver abrufen, die an einem Exemplar beteiligt sind.

Der Befehl ist wie folgt anzugeben:

```
db2nlist
```

Wenn Sie den Befehl wie angegeben verwenden, wird das aktuelle Exemplar (wie in der Umgebungsvariablen DB2INSTANCE festgelegt) als Standardexemplar verwendet. Mit dem folgenden Befehl können Sie ein bestimmtes Exemplar festlegen:

```
db2nlist /i:exempName
```

Dabei ist *exempName* der Name des gewünschten Exemplars.

Mit dem folgenden Befehl können Sie außerdem (wahlfrei) den Status jedes Partitionsservers abfragen:

```
db2nlist /s
```

Jeder Datenbankpartitionsserver kann einen der Statuswerte *starting*, *running*, *stopping* oder *stopped* aufweisen.

Hinzufügen eines Datenbankpartitionsservers zu einem Exemplar

Unter Windows NT oder Windows 2000 können Sie mit dem Befehl **db2ncrt** einen Datenbankpartitionsserver (Knoten) zu einem Exemplar hinzufügen.

Anmerkung: Wenn das Exemplar bereits Datenbanken enthält, darf der Befehl **db2ncrt** nicht verwendet werden. Verwenden Sie in diesem Fall stattdessen den Befehl **db2start addnode**. Dadurch wird sichergestellt, dass die Datenbank dem neuen Datenbankpartitionsserver korrekt hinzugefügt wird. **Editieren Sie AUF KEINEN FALL** die Datei `db2nodes.cfg`! Dies kann zu Inkonsistenzen im partitionierten Datenbanksystem führen.

Der Befehl verfügt über folgende erforderliche Parameter:

```
db2ncrt /n:knotennummer
        /u:benutzername,kennwort
/p:logischer_anschluss
```

- /n:

Die eindeutige Knotennummer zum Identifizieren des Datenbankpartitionsservers. Die Nummer kann aus den Werten 1 bis 999 (in aufsteigender Reihenfolge) bestehen.

- /u:

Der Name des Anmeldekontos und das Kennwort des DB2-Services.

- /p:logischer_anschluss

Die logische Anschlussnummer, die für den Datenbankpartitionsserver verwendet wird, wenn der logische Anschluss ungleich Null (0) ist. Wird hier kein Wert angegeben, wird die logische Anschlussnummer '0' zugeordnet.

Die logische Anschlussnummer ist nur dann wahlfrei, wenn der erste Knoten auf einer Maschine erstellt wird. Beim Erstellen eines logischen Knotens müssen Sie diesen Parameter angeben und eine logische Anschlussnummer auswählen, die noch nicht benutzt wird. Es gelten verschiedene Einschränkungen:

- Auf jeder Maschine muss ein Datenbankpartitionsserver mit dem logischen Anschluss '0' definiert sein.
- Die Anschlussnummer darf den für die FCM-Kommunikation reservierten Anschlussbereich in der Servicedatei im Verzeichnis `x:\winnt\system32\drivers\etc\` nicht überschreiten. Wenn Sie z. B. einen Bereich von vier Anschlüssen für das aktuelle Exemplar reservieren, beträgt die maximal zulässige Anschlussanzahl drei (Anschlüsse 1, 2 und 3; Anschluss 0 ist dem logischen Standardknoten zugeordnet). Der Anschlussbereich wird definiert, wenn der Befehl **db2icrt** mit dem Parameter `/r:basisanschluss, endanschluss` angegeben wird.

Außerdem gibt es folgende wahlfreie Parameter:

- /g:netzwerkname

Gibt den Netzwerknamen für den Datenbankpartitionsserver an. Wird dieser Parameter nicht angegeben, verwendet DB2 die erste IP-Adresse, die auf dem System festgestellt wird.

Verwenden Sie diesen Parameter, wenn auf einer Maschine mehrere IP-Adressen definiert sind und wenn Sie eine spezifische IP-Adresse für den Datenbankpartitionsserver angeben wollen. Sie können für den Parameter *netzwerkname* den Netzwerknamen oder die IP-Adresse der gewünschten Einheit angeben.

- /h:host_name

Gibt an, welcher TCP/IP-Host-Name von FCM für die interne Übertragung verwendet wird, wenn der Host-Name nicht der lokale Host-Name ist. Dieser Parameter ist erforderlich, wenn Sie den Datenbankpartitionsserver auf einer fernen Maschine hinzufügen wollen.

- /i:exemplarname

Gibt den Exemplarnamen an. Der Standardwert ist das aktuelle Exemplar.

- /m:maschinename

Der Systemname der Windows NT-Workstation, auf der sich der Knoten befindet (der Standardname ist der Systemname der lokalen Maschine).

- /o:exemplareignermaschine

Der Systemname der Maschine, die Exemplareigner ist (der Standardname ist der Name der lokalen Maschine). Dieser Parameter ist erforderlich, wenn der Befehl **db2ncrt** auf einer Maschine aufgerufen wird, die nicht die Exemplareignermaschine ist.

Wenn Sie z. B. (um mehrere logische Knoten auszuführen) einen neuen Datenbankpartitionsserver zum Exemplar TESTMPP auf der Exemplareignermaschine MYMACHIN hinzufügen wollen und diesen neuen Knoten als Knoten 2 mit dem logischen Anschluss 1 definieren wollen, müssen Sie Folgendes eingeben:

```
db2ncrt /n:2 /p:1 /u:meine_id,mein_kennwort /i:TESTMPP  
/M:TEST /o:MYMACHIN
```

Ändern der Datenbankpartition

Unter Windows NT oder Windows 2000 können Sie mit dem Befehl **db2nchg** folgende Operationen ausführen:

- Versetzen der Datenbankpartition von einer Maschine auf eine andere.
- Ändern des TCP/IP-Host-Namens der Maschine.

Wenn Sie mit mehreren Netzwerkadaptern arbeiten wollen, müssen Sie mit diesem Befehl die TCP/IP-Adresse für das Feld „netname“ in der Datei *db2nodes.cfg* angeben.

- Verwenden einer anderen logischen Anschlussnummer.
- Verwenden eines anderen Namens für den Datenbankpartitionsserver (Knoten).

Der Befehl verfügt über den folgenden erforderlichen Parameter:

```
db2nchg /n:knotennummer
```

Der Parameter */n*: gibt die Knotennummer der Konfiguration des Datenbankpartitionsservers an, die geändert werden soll. Dieser Parameter ist obligatorisch.

Der Befehl unterstützt die folgenden wahlfreien Parameter:

- */i:exemplarname*

Gibt das Exemplar an, in dem sich der aktuelle Datenbankpartitionsserver befindet. Wird dieser Parameter nicht angegeben, wird standardmäßig das aktuelle Exemplar verwendet.

- */u:benutzername,kennwort*

Ändert den Namen des Anmeldekontos und das Kennwort für den DB2-Service. Wird dieser Parameter nicht angegeben, bleiben das Anmeldekonto und das Kennwort unverändert.

- */p:logischer_anschluss*

Ändert den logischen Anschluss für den Datenbankpartitionsserver. Dieser Parameter muss angegeben werden, wenn Sie den Datenbankpartitionsserver auf eine andere Maschine versetzen wollen. Wird dieser Parameter nicht angegeben, bleibt die logische Anschlussnummer unverändert.

- */h:host_name*

Ändert den TCP/IP-Host-Namen, der von FCM für die interne Kommunikation verwendet wird. Wird dieser Parameter nicht angegeben, bleibt der Host-Name unverändert.

- /m:maschinename
Versetzt den Datenbankpartitionsserver auf eine andere Maschine. Der Datenbankpartitionsserver kann nur dann versetzt werden, wenn das Exemplar keine bereits definierten Datenbanken enthält.
- /g:netzwerkname
Ändert den Netzwerknamen des Datenbankpartitionsservers.
Verwenden Sie diesen Parameter, wenn auf einer Maschine mehrere IP-Adressen definiert sind und wenn Sie eine spezifische IP-Adresse für den Datenbankpartitionsserver verwenden wollen. Sie können für den Parameter netzwerkname den Netzwerknamen oder die IP-Adresse der gewünschten Einheit angeben.

Wenn z. B. der logische Anschluss für den Knoten 2 geändert werden soll, der an dem Exemplar TESTMPP beteiligt ist, müssen Sie folgenden Befehl eingeben, um den logischen Anschluss 3 zu verwenden:

```
db2nchg /n:2 /i:TESTMPP /p:3
```

Löschen einer Datenbankpartition aus einem Exemplar

Unter Windows NT oder Windows 2000 können Sie mit dem Befehl **db2ndrop** einen Datenbankpartitionsserver (Knoten) aus einem Exemplar löschen, in dem keine Datenbanken definiert sind. Wenn Sie einen Datenbankpartitionsserver löschen, kann seine Knotennummer für einen neuen Datenbankpartitionsserver verwendet werden.

Gehen Sie beim Löschen von Datenbankpartitionsservern aus einem Exemplar mit besonderer Sorgfalt vor. Wenn Sie den Exemplareigner-Datenbankpartitionsserver (Knoten Null) des Exemplars entfernen, wird das Exemplar dadurch unbrauchbar. Verwenden Sie zum Löschen des Exemplars den Befehl **db2idrop**.

Anmerkung: Wenn das Exemplar bereits Datenbanken enthält, darf der Befehl **db2ndrop** nicht verwendet werden. Verwenden Sie stattdessen den Befehl **db2stop drop nodenum**. Dadurch wird sichergestellt, dass die Datenbank in der Datenbankpartition korrekt gelöscht wird. **Editieren Sie AUF KEINEN FALL** die Datei `db2nodes.cfg`! Dies kann zu Inkonsistenzen im partitionierten Datenbanksystem führen.

Wenn Sie einen Knoten löschen wollen, dem über eine Maschine mit mehreren logischen Knoten der logische Anschluss 0 zugeordnet ist, müssen Sie alle anderen Knoten löschen, die den anderen logischen Anschlüssen zugeordnet sind, bevor Sie den Knoten mit dem logischen Anschluss 0 löschen können. Jeder Datenbankpartitionsserver muss über einen Knoten verfügen, dem der logische Anschluss 0 zugeordnet ist.

Der Befehl verfügt über folgende Parameter:

```
db2ndrop /n:knotennummer /i:exemplarname
```

- /n:
Die eindeutige Knotennummer zum Identifizieren des Datenbankpartitionsservers. Dieser Parameter ist erforderlich. Die Nummer kann aus den Werten 0 bis 999 (in aufsteigender Reihenfolge) bestehen. Bitte beachten Sie, dass der Knoten Null (0) die Exemplareignermaschine ist.
- /i:exemplarname
Der Exemplarname. Dies ist ein wahlfreier Parameter. Wird dieser Parameter nicht angegeben, wird standardmäßig das (mit der Variablen DB2INSTANCE der Profilregistrierdatenbank definierte) aktuelle Exemplar verwendet.

Anhang H. Konfigurieren mehrerer logischer Knoten

Normalerweise wird DB2 Enterprise - Extended Edition so konfiguriert, dass jeder Maschine ein Datenbankpartitionsserver zugeordnet ist. Es gibt jedoch verschiedene Fälle, in denen es günstiger ist, wenn auf einer Maschine mehrere Datenbankpartitionsserver ausgeführt werden. Dies bedeutet, dass die Konfiguration eine höhere Anzahl von Knoten als Maschinen umfassen kann. In diesen Fällen werden auf der betreffenden Maschine *mehrere logische Knoten* ausgeführt, wenn diese *dasselbe* Exemplar verwenden. Andernfalls werden auf der Maschine *nicht* mehrere logische Knoten ausgeführt.

Durch die Unterstützung mehrerer logischer Knoten können drei verschiedene Konfigurationsarten ausgewählt werden:

- Eine Standardkonfiguration, bei der jede Maschine über nur einen Datenbankpartitionsserver verfügt.
- Eine Konfiguration mit mehreren logischen Knoten, bei der eine Maschine über mehrere Datenbankpartitionsserver verfügt.
- Eine Konfiguration, bei der mehrere logische Knoten auf mehreren Maschinen ausgeführt werden.

Konfigurationen mit mehreren logischen Knoten eignen sich gut, wenn mit dem System Abfragen auf einer Maschine ausgeführt werden, die auf der SMP-Architektur (SMP = Symmetric Multiprocessor) basiert. Die Möglichkeit, mehrere logische Knoten auf einer Maschine zu konfigurieren, ist auch im Falle eines Maschinenausfalls von Vorteil. Wenn eine Maschine (und damit auch der bzw. die auf dieser Maschine ausgeführte(n) Datenbankpartitionsserver) ausfallen, können Sie den/die Datenbankpartitionsserver mit dem Befehl DB2START NODENUM auf einer anderen Maschine erneut starten. Hierdurch wird die kontinuierliche Verfügbarkeit der Benutzerdaten sichergestellt.

Ein weiterer Vorteil bei mehreren logischen Knoten besteht darin, dass diese die SMP-Hardwarekonfigurationen nutzen können. Darüber hinaus lassen sich durch die kleineren Datenbankpartitionen bessere Leistungswerte beim Sichern und Zurückschreiben von Datenbankpartitionen und Tabellenbereichen sowie bei der Indexerstellung erzielen.

Sie können mehrere logische Knoten auf eine der beiden folgenden Arten konfigurieren:

- Konfigurieren der logischen Knoten (Datenbankpartitionen) in der Datei `db2nodes.cfg`. In diesem Fall können Sie alle logischen und fernen Knoten mit dem Befehl DB2START bzw. der entsprechenden API starten.

Anmerkung: Bei Windows NT müssen Sie zum Hinzufügen eines Knotens *db2ncrt* verwenden, wenn auf dem System keine Datenbank vorhanden ist. Sind bereits eine oder mehrere Datenbanken angelegt, sollte stattdessen der Befehl `DB2START ADDNODE` eingegeben werden. Unter Windows NT sollte die Datei *db2nodes.cfg* niemals manuell editiert werden.

- Neustarten eines logischen Knotens auf einem anderen Prozessor, auf dem bereits andere logische Datenbankpartitionen (Knoten) ausgeführt werden. Mit dieser Methode können Sie den Host-Namen und die Anschlussnummer, die in der Datei *db2nodes.cfg* für die logische Datenbankpartition angegeben sind, außer Kraft setzen.

Zur Konfiguration einer logischen Datenbankpartition (Knoten) in der Datei *db2nodes.cfg* müssen Sie einen Eintrag in der Datei vornehmen, um dem Knoten eine logische Anschlussnummer zuzuordnen. Dabei gilt die folgende Syntax:

```
knotennummer host-name logischer-anschluss netzname
```

Anmerkung: Bei Windows NT müssen Sie zum Hinzufügen eines Knotens *db2ncrt* verwenden, wenn auf dem System keine Datenbank vorhanden ist. Sind bereits eine oder mehrere Datenbanken angelegt, sollte stattdessen der Befehl `DB2START ADDNODE` eingegeben werden. Unter Windows NT sollte die Datei *db2nodes.cfg* niemals manuell editiert werden.

Das Format der Datei *db2nodes.cfg* unter Windows NT unterscheidet sich vom Format dieser Datei auf einem UNIX-System. Unter Windows NT gilt folgendes Spaltenformat:

```
knotennummer host-name computername logischer_anschluss netzname
```

Sie müssen sicherstellen, dass in der Datei *services* des Verzeichnisses *etc* für die FCM-Kommunikation genügend Anschlüsse definiert sind.

Anhang I. Knotenübergreifende Hochgeschwindigkeitskommunikation

Beim Einsatz von DB2 Universal Database Enterprise - Extended Edition arbeiten Sie möglicherweise in einer kommunikationsintensiven Umgebung, bei der der Gesamtdurchsatz des Systems von zentraler Bedeutung für Ihr Unternehmen ist.

Es gibt zwei Netzwerktypen, die für Ihre partitionierte Umgebung eingesetzt werden können. Bei einem dieser Netzwerke wird die Datenübertragung über ein öffentliches LAN unter TCP/IP realisiert. Bei dem anderen Typ erfolgt die Kommunikation über eine dedizierte Verbindung unter TCP/IP oder VI Architecture (VI = Virtual Interface).

Die öffentliche Verbindung arbeitet mit vorhandenen TCP/IP-Komponenten. TCP/IP steht als Kommunikationsprotokoll fast überall zur Verfügung. Im vorliegenden Fall wird von einer LAN-Umgebung (LAN = Local Area Network) ausgegangen. Der Vorteil dieser Umgebung besteht darin, dass Sie Ihren Cluster direkt anschließen können, ohne dass hierzu zusätzliche lizenzpflichtige Hard- oder Software erforderlich ist. Der Nachteil dieser Umgebung liegt darin, dass der zusätzliche Datenverkehr im Cluster sich negativ auf die Qualität der im gesamten LAN zur Verfügung stehenden Services auswirkt. So kann z. B. die Datenbankaktivität im Cluster zu einem drastisch erhöhten Kommunikationsaufkommen führen, das sich auf die gesamte Kommunikation im LAN auswirkt. Darüber hinaus erschwert die restliche Kommunikation in der LAN-Umgebung die Gewährleistung einer gleich bleibenden Leistung bei der Datenbankverarbeitung innerhalb des Clusters.

Die dedizierte Verbindung wird als separates Netzwerk ausgeführt. Das Netzwerk ist möglicherweise das einzig verfügbare Netzwerk innerhalb des Clusters oder wird zusätzlich zu der LAN-Umgebung verwendet. Das Netzwerk ist für die Bereitstellung der Kommunikation zwischen den Mitgliedern des Clusters dediziert. Dieser Netzwerktyp wird als SAN (System Area Network) bezeichnet. Die Leistung der Datenbank wird durch den externen Kommunikationsverkehr nicht (wie in einer LAN-Umgebung) beeinflusst und umgekehrt. Der Nachteil dieser Umgebung liegt darin, dass möglicherweise eine separate Verwaltungsfunktion für beide Netzwerke sowie zusätzliche Kosten für Hard- und Softwarekomponenten und Protokolle für das LAN und das SAN anfallen. Als Beispiel für eine dedizierte Verbindung kann das mit 100 Mb/Sekunde arbeitende Ethernet-Netzwerk aufgeführt werden.

Möglicherweise wollen Sie Ihre bereits vorhandene LAN-Umgebung weiterhin verwenden, jedoch außerdem über die Möglichkeiten verfügen, große Datenmengen über das SAN (innerhalb Ihres Clusters) zu übertragen. Eine solche Konstellation ist günstig, wenn Sie den Kommunikationszugriff auch außerhalb der Grenzen Ihres Clusters realisieren wollen. Innerhalb einer Windows NT-Betriebsumgebung können Sie das öffentliche LAN für den Kommunikationszugriff auf den NT-Domänen-Controller verwenden. (Informationen zum Domänen-Controller finden Sie in „Anhang E. Arbeitsweise von DB2 für Windows NT mit der Windows NT-Sicherheit“ auf Seite 429.)

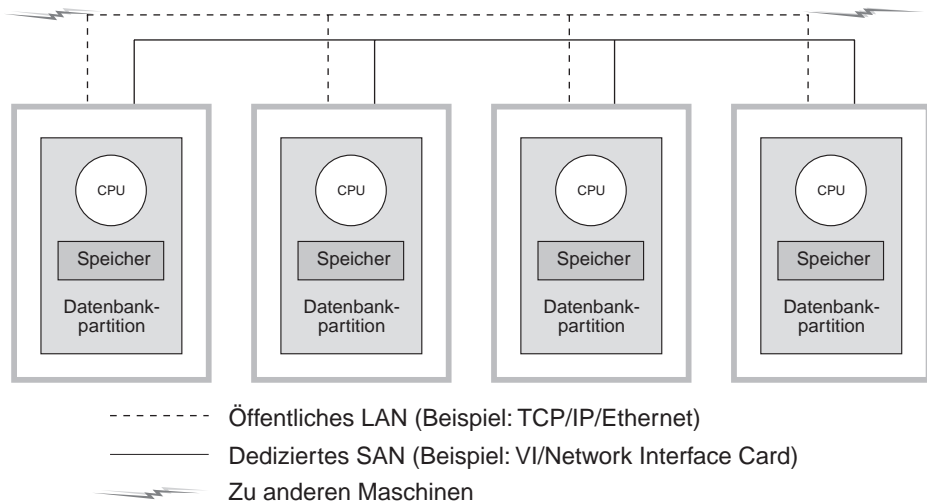


Abbildung 12. Kombination von dediziertem SAN mit einem öffentlichen LAN

In den restlichen Abschnitten werden die folgenden Themen behandelt:

- „Hochgeschwindigkeitsverbindungen unter TCP/IP“
- „Hochgeschwindigkeitsverbindungen mit VI“ auf Seite 457

Hochgeschwindigkeitsverbindungen unter TCP/IP

Im Folgenden sind Beispiele für die Voraussetzungen aufgeführt, die bei der Konfiguration der Netzwerkhardware unter TCP/IP gelten:

- Ethernet-Standardkonfiguration.

Es gibt keine besonderen Voraussetzungen in Bezug auf die Hardware, Software sowie die verwendeten Protokolle.

- IBM Netfinity SP Switch.

Die geltenden Voraussetzungen werden im nächsten Abschnitt erläutert.

Voraussetzungen für den Einsatz eines IBM Netfinity SP Switch

Informationen zu Netfinity finden Sie unter folgender URL-Adresse:
<http://www.ibm.com/pc/us/netfinity>

Zusätzliche Dokumentation und Softwareerweiterungen können Sie über die IBM Support Web Site unter folgender Adresse abrufen:

<http://www.ibm.com/pc/support>

1. Klicken Sie auf **Servers**.
2. Klicken Sie unter 'Family' auf **Clustering**.
3. Klicken Sie unter 'Technical Information' auf **Downloadable Files**, wenn Sie Softwareerweiterungen benötigen, oder klicken Sie auf **Online Publications**, um die Dokumentation abzurufen.

Suchen Sie das Thema für den IBM Netfinity SP Switch und laden Sie die benötigten Dateien herunter.

Installationsprozedur für einen IBM Netfinity SP Switch

Anweisungen zum Installieren des IBM Netfinity SP Switch finden Sie im Handbuch *IBM Netfinity SP Switch Installation and User's Guide*.

Verwenden Sie zum Installieren, Konfigurieren und Testen der verwendeten Hard- und Softwarekomponenten die Hard- und Softwarehandbücher, die zum Lieferumfang dieser Komponenten gehören. (Hierzu zählen z. B. das Server-Gehäuse, der Host-Adapter sowie die Software für den SP Switch).

DB2 kann nach der Installation den IBM Netfinity SP Switch ohne weitere Modifikationen verwenden.

Hochgeschwindigkeitsverbindungen mit VI

Virtual Interface (VI) Architecture ist das zwischen Knoten verwendete Übertragungsprotokoll, das alternativ zu TCP/IP in einer Windows NT-Konfiguration mit exklusiver Parallelverarbeitung (MPP) eingesetzt wird. VI Architecture ist eine neue Kommunikationsarchitektur, die in Zusammenarbeit von Intel, Microsoft und Compaq entwickelt wurde, um die Leistung über ein SAN (System Area Network) zu verbessern. Weitere Informationen zu VI Architecture finden Sie unter der URL-Adresse <http://www.viarch.org>.

Es gibt Produkte, die getrennt von DB2 Universal Database erhältlich sind und eine VI Architecture-fähige Netzwerkschnittstellenkarte (NIC - Network Interface Card), einen Switch sowie eine Softwaretreiberimplementierung besitzen. Verschiedene unabhängige Hardwarehersteller (IHV - Independent Hardware Vendors) bieten solche Produkte auf dem Markt an bzw. planen dies zu tun.

VI Architecture bietet im Vergleich zu TCP/IP niedrige Latenzzeiten, eine hohe Bandbreite und eine geringere CPU-Belegung. In einer kommunikationsintensiven Umgebung kann durch VI Architecture der allgemeine Systemdurchsatz verbessert werden. Je mehr Knoten im Cluster sind und je mehr Daten übertragen werden, desto größer ist der Nutzen von VI Architecture.

DB2 Universal Database unterstützt VI Architecture-Implementierungen, die der *Virtual Interface Architecture Specification, Version 1.0* und dem *Intel Virtual Interface (VI) Architecture Developers' Guide, Version 1.0* entsprechen und die „Virtual Interface Architecture Conformance Suite“ unterstützen. Die Spezifikation kann im World Wide Web unter http://www.intel.com/design/servers/vi/the_spec/specification.htm abgerufen werden. Der Developer's Guide kann im World Wide Web unter http://www.intel.com/design/servers/vi/developer/ia_imp_guide.htm abgerufen werden. Informationen zur Conformance Suite sind ebenfalls unter dieser URL-Adresse zu finden.

IBM kündigte Unterstützung für Virtual Interface (VI) Architecture ab DB2 Universal Database EEE Version 5.2 an.

Informationen zu weiteren Produkten, die VI Architecture verwenden und von DB2 Universal Database EEE unterstützt werden, erhalten Sie von der DB2 UDB-Systemunterstützung unter <http://www.software.ibm.com/data> oder unter Tel. +1-800-237-5511 (nur in den USA und Kanada).

Folgende Produkte wurden mit DB2 Universal Database getestet:

- GigaNet Interconnect (siehe „Einrichtung von GigaNet Interconnect“ auf Seite 459)
- Compaq ServerNet Interconnect (siehe „Einrichtung von ServerNet Interconnect“ auf Seite 462)
- Fujitsu Synfinity Interconnect (siehe „Konfigurationsprozedur für Synfinity-Verbindung“ auf Seite 465)

Es kann noch andere Produkte geben, die mit DB2 Universal Database zusammenarbeiten können. Fragen Sie beim Lieferanten des Produkts und bei IBM Kundendienst und Unterstützung nach, um sicherzustellen, dass das gewünschte Produkt unterstützt wird.

Hardwarekonfiguration mit Virtual Interface (VI)

Im Folgenden sind Beispiele für die Voraussetzungen aufgeführt, die bei der Konfiguration der Netzwerkhardware mit VI gelten:

- GigaNet Interconnect.
„Einrichtung von GigaNet Interconnect“ auf Seite 459 enthält einen Überblick über die Informationen zur Hard- und Software sowie zum Protokoll, die für diese Auswahl benötigt werden.

Informationen zu GigaNet-Produkten oder zur GigaNet-Kundendienst- oder -Unterstützungsfunktion finden Sie unter folgender URL-Adresse:
<http://www.giganet.com/>

- Compaq ServerNet Interconnect.

„Einrichtung von ServerNet Interconnect“ auf Seite 462 enthält einen Überblick über die Informationen zur Hard- und Software sowie zum Protokoll, die für diese Auswahl benötigt werden.

Informationen zu ServerNet-Produkten oder zur ServerNet-Kundendienst- oder -Unterstützungsfunktion finden Sie unter folgender URL-Adresse:
<http://www.servernet.com/>

- Fujitsu Synfinity Interconnect.

„Konfigurationsprozedur für Synfinity-Verbindung“ auf Seite 465 enthält einen Überblick über die Informationen zur Hard- und Software sowie zum Protokoll, die für diese Auswahl benötigt werden.

Informationen zu SynfinityNet-Produkten oder zur SynfinityNet-Kundendienst- oder -Unterstützungsfunktion bei Fujitsu System Technologies finden Sie unter folgender URL-Adresse: <http://www.fjst.com/>

Sie müssen DB2 für die Verwendung von VI konfigurieren. Informationen zur Verwendung von VI finden Sie in „Aktivieren von DB2 zur Ausführung mit VI“ auf Seite 467.

Einrichtung von GigaNet Interconnect

Die Liste der erforderlichen Hardware und Software für die Einrichtung dieser Umgebung umfasst die folgenden Produkte:

- GigaNet-Netzwerkschnittstellenkarte GNN1000
- GigaNet-Switch GNX5000
- GigaNet-Kupferverbindungskabel GNCxx11
- GigaNet cLAN-Software, Version 2.0

Die erforderlichen Schritte, mit denen sichergestellt werden kann, dass GigaNet Interconnect mit DB2 Universal Database zusammenarbeiten kann, sind unten aufgeführt. Jeder Schritt stellt nur eine Zusammenfassung dessen dar, was bei jedem Schritt erforderlich ist. Es werden hier nicht alle Einzelheiten für jeden Schritt aufgeführt. Sie sollten bei jedem Schritt die angegebene Dokumentation berücksichtigen, die die notwendigen ausführlichen Anweisungen enthält.

Jede GigaNet-Netzwerkschnittstellenkarte GNN1000 wird mit einer CD-ROM der GigaNet cLAN-Software ausgeliefert. Die CD-ROM enthält die notwendige Software zum Einrichten von GigaNet Interconnect. Darüber hinaus enthält sie auch das Software Developer's Kit für VI Architecture und Adobe Acrobat Reader. Das Software Developer's Kit für VI Architecture wird für die Entwicklung und den Test von VI-fähigen Anwendung verwendet. Der Adobe

Acrobat Reader wird zum Anzeigen von Dokumenten auf der CD-ROM verwendet, die Informationen zur Entwicklung von VI-fähigen Anwendungen enthalten.

Zusammenfassung der Schritte:

1. Adapterkarten installieren
2. Switches und Kabel installieren
3. Adaptertreiber installieren
4. cLAN Management Console installieren
5. Interconnect testen

Die Schritte im Einzelnen:

1. Installieren Sie die GigaNet-Netzwerkschnittstellenkarte GNN1000. Installationsanweisungen finden Sie im *GigaNet GNN1000 User Guide*.
2. Installieren Sie den GigaNet-Switch GNX5000 und die Kabel. Installationsanweisungen finden Sie im *GigaNet GNX5000 User Guide*.
3. Installieren Sie die Treibersoftware für den GigaNet-Adapter GNN1000 auf jedem Knoten, der an den Switch GNX5000 angeschlossen ist. Installationsanweisungen finden Sie im *GigaNet GNN1000 User Guide*. Es folgen weitere Einzeldaten für den Fall, dass Sie die Installation mit Hilfe der von GigaNet bereitgestellten Treiber durchführen:
 - a. Entfernen Sie gegebenenfalls die vorherige Version des Treibers GNN1000. Nach dem Entfernen muss der Knoten neu gestartet werden.
 - b. Installieren Sie den Treiber mit Start→Einstellungen→Systemsteuerung→Netzwerk→Netzwerkkarte→Hinzufügen.
 - c. Klicken Sie **Diskette...** an, und geben Sie das Treiberverzeichnis auf der CD-ROM an. Wenn beispielsweise F: Ihr CD-ROM-Laufwerk ist, würden Sie F:\Driver angeben.
 - d. Wählen Sie „GNN1000 NDIS Adapter“ aus, und klicken Sie **OK** an.
 - e. Konfigurieren Sie die Netzwerkprotokolle, um die Installation abzuschließen.

Die Treibersoftware für den GigaNet-Adapter kann auch von der GigaNet-Web-Site unter <http://www.giganet.com> heruntergeladen werden. Beachten Sie die Anweisungen zum Herunterladen und Installieren auf der Unterstützungsseite der GigaNet-Web-Site. Die Installation des Adaptertreibers GNN1000 bewirkt einen Neustart des Knotens.

4. Die GigaNet cLAN Management Console (GMC) kann zum Testen der Integrität von GigaNet Interconnect verwendet werden. Die GigaNet cLAN Management Console besteht aus zwei Teilen: der Konsole und dem

Agenten. Der Agent muss auf allen Knoten im Cluster installiert werden. Die Konsole kann auf jedem Netzwerkknoten installiert werden, der auf die Knoten im Cluster zugreifen kann. Die vielseitigste und daher die empfohlene Installation ist die, bei der die Konsole und der Agent auf jedem Knoten im Cluster installiert werden.

Installieren Sie die GigaNet cLAN Management Console. Installationsanweisungen und zusätzliche Informationen zur cLAN Management Console finden Sie im *GigaNet GNN1000 User Guide*. Weitere Einzelheiten zum Installationsverfahren:

- a. Legen Sie die CD mit der cLAN-Software in das CD-ROM-Laufwerk ein.
- b. Warten Sie, bis das automatische Installationsmenü der CD angezeigt wird.
- c. Klicken Sie „Install cLAN Management Console“ an.
- d. Wiederholen Sie das Installationsverfahren auf allen übrigen Knoten im Cluster.

Die Software für GigaNet cLAN Management Console kann auch von der GigaNet-Web-Site unter <http://www.giganet.com> heruntergeladen werden. Beachten Sie die Anweisungen zum Herunterladen und Installieren auf der Unterstützungsseite der GigaNet-Web-Site.

Durch die Installation der cLAN Management Console wird der Knoten eventuell neu gestartet.

5. Testen Sie, ob die GigaNet-Hardware funktioniert. Gehen Sie dazu wie folgt vor:
 - a. Öffnen Sie die GMC (Programme→GigaNet→cLAN Management Console).
 - b. Ein Dialogfenster wird geöffnet, in dem alle Maschinen im LAN angezeigt werden, auf die zugegriffen werden kann. Drücken Sie **ESC**.
 - c. Wählen Sie in der Menüleiste **Console**→**Local** aus.
 - d. Bestätigen Sie, dass alle Member im Cluster angezeigt werden und dass alle aktiv sind.
 - e. Wählen Sie in der Menüleiste **Utilities**→**VI Throughput** aus. Nun wird ein Durchsatztest durchgeführt, um zu prüfen, ob die Daten tatsächlich die Hardware durchlaufen.
 - f. Geben Sie in Großbuchstaben die Systemnamen der beiden Knoten ein, die Sie testen wollen. Geben Sie den lokalen Knoten als Ausgangsknoten an.
 - g. Klicken Sie **Start Measuring** an. Die Daten sollten mit einer Geschwindigkeit von mindestens 65 MB pro Sekunde übertragen werden.
 - h. Klicken Sie **Stop Measuring** an, um den Verbindungstest zu stoppen.

- i. Wiederholen Sie den Test für die anderen Knoten im Cluster, und messen Sie dabei den Durchsatz zwischen dem lokalen Knoten (Ausgangsknoten) und den anderen Knoten (Empfangsknoten).

Wenn der Verbindungstest nicht funktioniert, schlagen Sie nach in den Abschnitten zur Fehlerbehebung im *GigaNet GNN1000 User Guide* und *GigaNet GNX5000 User Guide*.

Informationen zum Installieren und Implementieren von DB2 Universal Database für Windows NT finden Sie im Handbuch *DB2 Enterprise - Extended Edition für Windows Einstieg*.

Einrichtung von ServerNet Interconnect

Die Liste der erforderlichen Hardware und Software für die Einrichtung dieser Umgebung umfasst die folgenden Produkte:

- ServerNet PCI Adapter Driver (SPAD), (Produkt-ID T0089), Version 1.3.5 oder neuer
- ServerNet-Switch 1
- ServerNet Area Network Manager (SANMan), (Produkt ID T0087), Version 1.1.3 oder neuer

Die erforderlichen Schritte, mit denen sichergestellt werden kann, dass ServerNet Interconnect mit DB2 Universal Database zusammenarbeiten kann, sind unten aufgeführt. Jeder Schritt stellt nur eine Zusammenfassung dessen dar, was bei jedem Schritt erforderlich ist. Es werden hier nicht alle Einzelheiten für jeden Schritt aufgeführt. Sie sollten bei jedem Schritt die angegebene Dokumentation berücksichtigen, die die notwendigen ausführlichen Anweisungen enthält.

Bei den unten aufgeführten Schritten wird davon ausgegangen, dass Sie maximal sechs (6) Knoten im Cluster verwenden. Wenn Sie mehr als sechs Knoten benötigen, wenden Sie sich an ServerNet.

Die Schritte im Einzelnen:

1. Installieren Sie die ServerNet-Netzwerkschnittstellenkarte. Installationsanweisungen finden Sie im *ServerNet-I Virtual Interface Software Release Document*, (Produkt-ID N0031).
2. Installieren Sie den ServerNet-Switch 1. Installationsanweisungen finden Sie im *ServerNet-I Virtual Interface Software Release Document*, (Produkt-ID N0031).
3. Entfernen Sie die Installation früherer ServerNet-Treiber. (Lassen Sie diesen Schritt aus, wenn Sie ServerNet zum ersten Mal installieren.)
 - a. Öffnen Sie das Objekt **Netzwerk** der Systemsteuerung (Start->Einstellungen->Systemsteuerung->Netzwerk).

- b. Klicken Sie die Indexzunge **Netzwerkkarte** an.
 - c. Entfernen Sie den **Tandem ServerNet PCI Adapter Driver**.
 - d. Klicken Sie die Indexzunge **Dienste** an.
 - e. Entfernen Sie **SANMan**.
 - f. Klicken Sie die Indexzunge **Protokolle** an.
 - g. Entfernen Sie **Tandem ServerNet-I VI Protocol**.
4. Installieren Sie den Tandem ServerNet PCI Adapter Driver. Es folgen weitere Einzeldaten für den Fall, dass Sie die Installation mit Hilfe der von ServerNet bereitgestellten Software-CD durchführen:
- a. Öffnen Sie das Objekt **Netzwerk** der Systemsteuerung (Start→Einstellungen→Systemsteuerung→Netzwerk).
 - b. Klicken Sie die Indexzunge **Netzwerkkarte** an. (Das Fenster **Netzwerkkarte** wird angezeigt.)
 - c. Stellen Sie sicher, dass der neue ServerNet-Treiber sich in einem separaten Laufwerk und/oder Verzeichnis befindet. Geben Sie dann an einer Eingabeaufforderung unter Verwendung des richtigen Laufwerks und/oder Verzeichnisses „ernnn.exe -d“ ein, um das selbst-extrahierende Programm zu starten. („ernnn.exe“ ist der Name des Engineering Release gefolgt von einer Nummer, ERnnn.EXE, welche die spezifische Version des zu installierenden ServerNet-Treibers angibt.)
 - d. Wechseln Sie in das Laufwerk und/oder Verzeichnis, in dem sich die extrahierten Dateien befinden. Wechseln Sie in das Unterverzeichnis „Spad n.n.n \ Free“ (dabei steht „n.n.n“ für die Version des Produkts). (Wenn Sie in einer Fehlerbehebungs- oder Entwicklungsumgebung arbeiten, wechseln Sie in das Unterverzeichnis „Spad n.n.n \ Checked“ statt in das Unterverzeichnis „Spad n.n.n \ Free“.)
 - e. Benennen Sie die Datei „oemsetup.multi_node“ in „oemsetup.inf“ um.
 - f. Wählen Sie auf der Indexzunge 'Netzwerkkarte' die Option **Hinzufügen** aus. (Das Fenster **Auswahl: Netzwerkkarte** wird geöffnet.)
 - g. Klicken Sie **Diskette...** an. (Das Fenster **Diskette einlegen** wird geöffnet.)
 - h. Geben Sie das Laufwerk und/oder Verzeichnis an, in dem sich die Datei „oemsetup.inf“ befindet.
 - i. Prüfen Sie, ob das Dialogfenster „Tandem ServerNet PCI Adapter Driver“ anzeigt, und klicken Sie dann **OK** an. Stellen Sie sicher, dass die Liste der Adapter jetzt den ServerNet-Adapter enthält. Klicken Sie **Schließen** an.
 - j. Wählen Sie **Ja**, um den Computer neu zu starten. Oder wählen Sie **Nein** aus, und installieren Sie SANMan und das VI Software Developer's Kit (SDK).

5. Installieren Sie SANMan. Es folgen weitere Einzeldaten für den Fall, dass Sie die Installation mit Hilfe der von ServerNet bereitgestellten Software-CD durchführen:
 - a. Öffnen Sie das Objekt **Netzwerk** der Systemsteuerung (Start→Einstellungen→Systemsteuerung→Netzwerk).
 - b. Klicken Sie die Indexzunge **Dienste** an. (Das Fenster **Dienste** wird angezeigt.)
 - c. Stellen Sie sicher, dass der neue ServerNet-Treiber sich in einem separaten Laufwerk und/oder Verzeichnis befindet. Geben Sie dann an einer Eingabeaufforderung unter Verwendung des richtigen Laufwerks und/oder Verzeichnisses „ernnn.exe -d“ ein, um das selbst-extrahierende Programm zu starten. („ernnn.exe“ ist der Name des Engineering Release gefolgt von einer Nummer, ERnnn.EXE, welche die spezifische Version des zu installierenden ServerNet-Treibers angibt.)
 - d. Wählen Sie auf der Indexzunge 'Services' die Option **Hinzufügen** aus. (Das Fenster **Auswahl: Netzwerkdienst** wird geöffnet.)
 - e. Wechseln Sie in das Laufwerk und/oder Verzeichnis, in dem sich die extrahierten Dateien befinden. Wechseln Sie in das Unterverzeichnis „SANMan n.n.n \ Free“ (dabei steht „n.n.n“ für die Version des Produkts). (Wenn Sie in einer Fehlerbehebungs- oder Entwicklungsumgebung arbeiten, wechseln Sie in das Unterverzeichnis „SANMan n.n.n \ Checked“ statt in das Unterverzeichnis „SANMan n.n.n \ Free“.)
 - f. Stellen Sie fest, ob der Switch ein X- oder Y-Switch ist. Achten Sie dabei auf die Anzeigeleuchte am Switch. Eine Anzeigeleuchte ist mit „X“, die andere mit „Y“ bezeichnet.
 - g. Wenn es sich um einen X-Switch handelt, wählen Sie X=1 und Y=0 aus. Stellen Sie sicher, dass alle Kabel auf den Netzwerkschnittstellenkarten an den X-Anschluss angeschlossen sind.
 - h. Wenn es sich um einen Y-Switch handelt, wählen Sie X=0 und Y=1 aus. Stellen Sie sicher, dass alle Kabel auf den Netzwerkschnittstellenkarten an den Y-Anschluss angeschlossen sind.
 - i. Geben Sie die Anschlussnummer des Switches an, an den die Netzwerkschnittstellenkarte der aktuellen Maschine angeschlossen ist.
 - j. Wählen Sie für alle sechs (6) Anschlüsse „PC“ aus.
6. Installieren Sie das Virtual Interface Protocol (VIP). Es folgen weitere Einzeldaten für den Fall, dass Sie die Installation mit Hilfe der von ServerNet bereitgestellten Software-CD durchführen:
 - a. Öffnen Sie das Objekt **Netzwerk** der Systemsteuerung (Start→Einstellungen→Systemsteuerung→Netzwerk).
 - b. Klicken Sie die Indexzunge **Protokolle** an. (Das Fenster **Protokolle** wird angezeigt.)

- c. Stellen Sie sicher, dass der neue ServerNet-Treiber sich in einem separaten Laufwerk und/oder Verzeichnis befindet. Geben Sie dann an einer Eingabeaufforderung unter Verwendung des richtigen Laufwerks und/oder Verzeichnisses „ernnn.exe -d“ ein, um das selbst-extrahierende Programm zu starten. („ernnn.exe“ ist der Name des Engineering Release gefolgt von einer Nummer, ERnnn.EXE, welche die spezifische Version des zu installierenden ServerNet-Treibers angibt.)
 - d. Wählen Sie auf der Indexzunge 'Protokolle' die Option **Hinzufügen** aus. (Das Fenster **Auswahl: Netzwerkprotokoll** wird geöffnet.)
 - e. Klicken Sie **Diskette...** an. (Das Fenster **Diskette einlegen** wird geöffnet.)
 - f. Geben Sie das Laufwerk und/oder Verzeichnis an, in dem sich die extrahierten Dateien befinden.
7. Testen Sie, ob die ServerNet-Hardware funktioniert. Es sind keine speziellen Testprogramme verfügbar. Verwenden Sie stattdessen einfach DB2, um die ServerNet-Hardware zu testen.

Wenn die Hardware nicht funktioniert, schlagen Sie nach im *ServerNet-I Virtual Interface Software Release Document*, (Produkt-ID N0031).

Informationen zum Installieren und Implementieren von DB2 Universal Database für Windows NT finden Sie im Handbuch *DB2 Enterprise - Extended Edition für Windows Einstieg*.

Konfigurationsprozedur für Synfinity-Verbindung

Die Liste der erforderlichen Hardware und Software für die Einrichtung dieser Umgebung umfasst die folgenden Produkte:

- Synfinity PCI Network Interface Card
- Synfinity-Switch mit sechs Anschlüssen
- Synfinity-Verbindungskabel
- Synfinity Cluster Manager Software, Version 1.10

Die erforderlichen Schritte, mit denen sichergestellt werden kann, dass Synfinity Interconnect mit DB2 Universal Database zusammenarbeiten kann, sind unten aufgeführt. Jeder Schritt stellt nur eine Zusammenfassung dessen dar, was bei jedem Schritt erforderlich ist. Es werden hier nicht alle Einzelheiten für jeden Schritt aufgeführt. Sie sollten bei jedem Schritt die angegebene Dokumentation berücksichtigen, die die notwendigen ausführlichen Anweisungen enthält.

Jedes Synfinity-System wird zusammen mit der CD-ROM für die Synfinity Cluster Manager Software, Version 1.10, ausgeliefert. Die CD-ROM enthält die

notwendige Dokumentation und Software zum Einrichten der Synfinity-Verbindung. Darüber hinaus enthält die CD-ROM auch das Handbuch *Synfinity Cluster Manager Software User Guide*.

Wenn auf Ihrem System andere VI-Hardware, -Software und -Protokolle installiert sind, kann es möglicherweise erforderlich sein, diese Komponenten zu entfernen, bevor die Synfinity-Verbindung installiert wird.

Nach der Installation wird die Synfinity-Verbindung als exotische Hardware identifiziert, die möglicherweise nicht über die Systemsteuerung von Windows NT angezeigt werden kann.

Zusammenfassung der Schritte:

1. Adapterkarten installieren
2. Synfinity Cluster Manager Software installieren
3. Switches und Kabel installieren
4. Interconnect testen

Die Schritte im einzelnen:

1. Installieren Sie die Synfinity PCI Network Interface Card. Installationsanweisungen finden Sie im *Synfinity Cluster Manager Software User Guide*.
2. Installieren Sie die Synfinity Cluster Manager Software auf einem Knoten, der mit dem Switch verbunden ist. Installationsanweisungen finden Sie im *Synfinity Cluster-Benutzerhandbuch*.

Der ausgewählte Knoten wird als Cluster-Manager definiert. Hierbei handelt es sich um den einzigen Knoten, auf dem die Software von CD installiert werden muss.

Nach der Installation sollte die Synfinity Cluster Manager-Software ausgeführt werden. Der Cluster-Manager stellt einen Cluster-Plan und schrittweise Anweisungen zum Konfigurieren des Netzwerks bereit. Darüber hinaus bietet er Empfehlungen zu den besten Routing- und Verkabelungsoptionen. Dieser Schritt sollte ausgeführt werden, bevor Kabel an die Synfinity-Switches und -Netzwerkkarten angeschlossen werden. Im Rahmen der Planungsprozeduren verwendet der Cluster-Manager den Cluster-Plan zum Erstellen installierbarer Disketten für den Einsatz auf den anderen Knoten. Dieser Arbeitsschritt umfasst auch die Erstellung der Treibersoftware für die Karten, die auf den anderen Knoten installiert sind. Umfassende Informationen zu diesem Thema finden Sie im Handbuch *Synfinity Cluster Manager Software User Guide*.

3. Installieren Sie den Synfinity Switch und die zugehörigen Kabel. Installationsanweisungen finden Sie im *Synfinity Cluster-Benutzerhandbuch*.

4. Testen Sie, ob die Synfinity-Hardware funktioniert. Gehen Sie dazu wie folgt vor:
 - a. Öffnen Sie auf einem beliebigen System im Cluster ein Fenster für die Eingabeaufforderung unter Windows NT.
 - b. Wechseln Sie ins Unterverzeichnis "utils", über das die Synfinity Cluster Manager-Software geladen wurde.
 - c. Geben Sie "vittest" ein und notieren Sie die daraufhin angezeigte Knotennummer.
 - d. Aktivieren Sie ein anderes System innerhalb des Clusters und öffnen Sie auch dort ein Fenster für die Eingabeaufforderung.
 - e. Wechseln Sie ins Unterverzeichnis "utils", über das die Synfinity Cluster Manager-Software für dieses andere System geladen wurde.
 - f. Geben Sie "vittest x" ein, wobei x für die Knotennummer steht, die in Schritt c festgestellt wurde.
 - g. Daraufhin sollte eine Nachricht angezeigt werden, in der Sie über das fehlerfreie Funktionieren der Verbindung ("CONNECTION GOOD") informiert werden.
 - h. Wird eine Nachricht angezeigt, in der Sie darüber informiert werden, dass keine Verbindung hergestellt werden konnte ("NO CONNECTION"), prüfen Sie die Verkabelung und die Hardwarekonfiguration. Weitere Informationen zur Behebung dieses Problems finden Sie im Handbuch *Synfinity Cluster Manager Software User Guide*. Darüber hinaus finden Sie technische Tipps zu Ihrer Unterstützung auf den Webseiten unter der URL-Adresse <http://www.fjst.com/>

Informationen zum Installieren und Implementieren von DB2 Universal Database für Windows NT finden Sie im Handbuch *DB2 Enterprise - Extended Edition für Windows Einstieg*.

Aktivieren von DB2 zur Ausführung mit VI

Ausführliche Installationsanweisungen finden Sie im Handbuch *DB2 Enterprise - Extended Edition für Windows Einstieg*.

Nach dem Abschluss der Installation von DB2 auf der Basis der Informationen im Handbuch *DB2 Enterprise - Extended Edition für Windows Einstieg* müssen Sie die folgenden DB2-Variablen der Profilregistrierdatenbank definieren und die folgenden Aufgaben in jeder Datenbankpartition des Exemplars ausführen:

1. Setzen Sie DB2_VI_ENABLE=ON.

Verwenden Sie den Befehl **db2set**, um den Wert für die Variable der Registrierdatenbank zu ändern. Verwenden Sie den Befehl **db2_all**, um den Befehl **db2set** auf allen Datenbankpartitionsservern des Exemplars auszuführen. Sie müssen mit einem Benutzerkonto angemeldet sein, das zur Administratorengruppe gehört, um den Befehl **db2_all** auszuführen.

Im folgenden Beispiel wurde das Zeichen ; innerhalb der doppelten Anführungszeichen platziert, damit die Anforderung gleichzeitig auf allen Datenbankpartitionsservern des Exemplars ausgeführt werden kann:

```
db2_a11 ";db2set DB2_VI_ENABLE=ON"
```

Weitere Informationen zum Befehl **db2_all** finden Sie unter "Absetzen von Befehlen an mehrere Datenbankpartitionsserver" im Handbuch *Systemverwaltung: Implementierung*.

2. Setzen Sie DB2_VI_DEVICE=nic0.

Beispiel:

```
db2_a11 ";db2set DB2_VI_DEVICE=nic0"
```

Anmerkung: Bei Synfinity Interconnect sollte für diese Variable der Wert DB2_VI_DEVICE=VINIC angegeben sein. Der Einheitenname (VINIC) muss in Großbuchstaben eingegeben werden.

3. Setzen Sie DB2_VI_VIPL=vipl.dll.

Beispiel:

```
db2_a11 ";db2set DB2_VI_VIPL=vipl.dll"
```

Anmerkung: Der im Beispiel verwendete Wert ist der Standardwert für die Variable der Registrierdatenbank. Weitere Informationen finden Sie im Handbuch *Systemverwaltung: Optimierung*.

4. Geben Sie auf dem partitionierten Exemplar db2start ein.
5. Sehen Sie sich die Datei „db2diag.log“ an. Es sollte für jede Partition eine Nachricht vorhanden sein, die angibt, dass VI aktiviert ist.
6. Möglicherweise müssen die Konfigurationsparameter für den Fast Communications Manager (FCM) aktualisiert werden. Wenn aufgrund der geltenden Ressourceneinschränkungen unter FCM Probleme auftreten, müssen Sie die Werte für die FCM-Konfigurationsparameter anheben. Wenn Sie von einer anderen Umgebung für Hochgeschwindigkeitsverbindungen wechseln, in der die Werte für die FCM-Konfigurationsparameter angehoben wurden, müssen diese Werte möglicherweise wieder gesenkt werden. Unter Windows NT kann es außerdem erforderlich sein, die Variable DB2NTMEMSIZE der Profilregistrierdatenbank so zu definieren, dass die DB2-Standardwerte überschrieben werden. Weitere Informationen zu Variablen der Registrierdatenbank finden Sie im Handbuch *Systemverwaltung: Optimierung*.

Anhang J. LDAP-Verzeichnisservices (Lightweight Directory Access Protocol)

LDAP (Lightweight Directory Access Protocol) ist eine Standardmethode zum Zugriff auf Verzeichnisservices. Bei einem Verzeichnisservice handelt es sich um ein Repository mit Ressourceninformationen zu mehreren Systemen und Services innerhalb einer verteilten Umgebung. Er stellt den Client-/Server-Zugriff auf diese Ressourcen bereit. Jedes Datenbankserverexemplar wird seine Existenz auf einem LDAP-Server veröffentlichen und dem LDAP-Verzeichnis Datenbankinformationen zur Verfügung stellen, wenn die Datenbanken erstellt werden. Wenn ein Client eine Verbindung zur Datenbank herstellt, können die Kataloginformationen für den Server aus dem LDAP-Verzeichnis abgerufen werden. Die einzelnen Clients müssen die Kataloginformationen nun nicht mehr lokal auf den verschiedenen Maschinen speichern. Client-Anwendungen durchsuchen das LDAP-Verzeichnis nach den erforderlichen Informationen für die Herstellung der Verbindung zur Datenbank.

Es gibt ein Caching-Verfahren, so dass der Client das LDAP-Verzeichnis in seinen lokalen Verzeichniskatalogen nur einmal durchsuchen muss. Sobald die Informationen abgerufen sind, werden sie auf der lokalen Maschine gespeichert bzw. im Cache zwischengespeichert. Bei nachfolgenden Zugriffsoperationen auf dieselben Informationen werden die Werte des Datenbankmanager-Konfigurationsparameters *dir_cache* und der Variablen DB2LDAPCACHE der Profilregistrierdatenbank zugrunde gelegt.

- Wurden DB2LDAPCACHE=NO und *dir_cache*=NO definiert, werden die Informationen immer über LDAP gelesen.
- Wurden DB2LDAPCACHE=NO und *dir_cache*=YES definiert, werden die Informationen einmal über LDAP gelesen und dann in den DB2-Cache eingefügt.
- Wurde DB2LDAPCACHE=YES definiert oder wurde die Variable nicht gesetzt, werden die Informationen über das LDAP-Verzeichnis gelesen und der lokale Cache wird aktualisiert, wenn die erforderlichen Informationen nicht im lokalen Cache gefunden werden können.

Anmerkung: Die Variable DB2LDAPCACHE der Profilregistrierdatenbank gilt nur für die Datenbank und die Knotenverzeichnisse.

Unterstützen von LDAP-Client- und -Serverkonfigurationen

Die folgende Tabelle enthält eine Übersicht zu den unterstützten LDAP-Client- und -Serverkonfigurationen:

Tabelle 26. Unterstützte LDAP-Client- und -Serverkonfigurationen

	IBM SecureWay Directory V3.1 und V3.1.1	Microsoft Active Directory
IBM LDAP-Client	Unterstützt	Nicht unterstützt
Microsoft LDAP/ADSI-Client	Unterstützt	Unterstützt

IBM SecureWay Directory Version 3.1 ist ein LDAP-Server der Version 3, der für Windows NT, AIX und Solaris zur Verfügung steht. IBM SecureWay Directory wird als Teil des Basisbetriebssystems für AIX und AS/400 sowie mit OS/390 Security Server geliefert.

DB2 unterstützt den IBM LDAP-Client unter AIX, Solaris, Windows NT und Windows 98.

Microsoft Active Directory ist ein LDAP-Server der Version 3 und steht als Teil des Betriebssystems Windows 2000 Server zur Verfügung.

Die Unterstützung für den Microsoft LDAP-Client ist in den folgenden Microsoft-Produkten enthalten:

- Outlook 98, Outlook 2000 oder Outlook Express

Anmerkung: Outlook Express wird als Bestandteil des Microsoft Internet Explorer installiert.

- Exchange Server
- Windows NT Server Service Pack 4
- Windows 98 Second Edition
- Windows 2000

Die Unterstützung für den Microsoft LDAP-Client ist darüber hinaus in der Komponente Active Directory Service Interface (ADSI) enthalten: Die neueste Version von ADSI kann unter folgender Adresse heruntergeladen werden:

<http://www.microsoft.com/windows2000/techinfo/howitworks/activedirectory/adsilinks.asp>

Bei der Ausführung unter den Betriebssystemen Windows 98, Windows NT und Windows 2000 unterstützt DB2 die Verwendung des IBM LDAP-Clients oder des Microsoft-LDAP-Clients, um auf IBM SecureWay Directory Server zuzugreifen. Steht der Microsoft-LDAP-Client nicht zur Verfügung, versucht DB2, den IBM LDAP-Client zu verwenden. Zur expliziten Auswahl des IBM LDAP-Clients verwenden Sie den Befehl **db2set**, mit dem die Variable DB2LDAP_CLIENT_PROVIDER der Profilregistrierdatenbank auf den Wert „IBM“ gesetzt werden kann.

Unterstützung für Windows 2000 Active Directory

DB2 nutzt Windows 2000 Active Directory folgendermaßen:

1. Die DB2-Datenbankserver werden unter Windows 2000 Active Directory als `ibm_db2Node`-Objekte veröffentlicht. Die Objektklasse `ibm_db2Node` ist eine Unterklasse der Objektklasse `ServiceConnectionPoint` (SCP). Jedes `ibm_db2Node`-Objekt enthält Protokollkonfigurationsdaten, die es Client-Anwendungen ermöglichen, eine Verbindung zum DB2-Datenbankserver herzustellen. Beim Erstellen einer neuen Datenbank wird diese in Windows 2000 Active Directory als Objekt `ibm_db2Database` unter dem `ibm_db2Node`-Objekt veröffentlicht.
2. Beim Herstellen einer Verbindung zu einer fernen Datenbank fragt der DB2-Client Windows 2000 Active Directory über die LDAP-Schnittstelle nach dem Objekt `ibm_db2Database` ab. Die Protokollkommunikation zur Herstellung der Verbindung zum Datenbankserver (Bindeinformationen) wird über das Objekt `ibm_db2Node` ermöglicht, unter dem das Objekt `ibm_db2Database` erstellt wird.

Konfigurieren von DB2 zur Verwendung von Windows 2000 Active Directory

Um auf Windows 2000 Active Directory zugreifen zu können, müssen die folgenden Bedingungen erfüllt sein:

1. Die Maschine, auf der DB2 ausgeführt wird, muss zu einer Windows 2000-Domäne gehören.
2. Der Microsoft-LDAP-Client wurde installiert. Der Microsoft-LDAP-Client ist Bestandteil des Betriebssystems Windows 2000. Bei Windows 98 oder Windows NT müssen Sie prüfen, dass `wldap32.dll` im Systemverzeichnis definiert ist.
3. Aktivieren Sie die LDAP-Unterstützung. Bei Windows 2000 wird die LDAP-Unterstützung durch das Installationsprogramm aktiviert. Bei Windows 98/NT müssen Sie LDAP explizit aktivieren. Setzen Sie hierzu mit dem Befehl **db2set** für die Variable `DB2_ENABLE_LDAP` der Profilregistrierdatenbank den Wert „YES“.
4. Melden Sie sich während der Ausführung von DB2 mit einem Domänenbenutzereintrag an, um Informationen aus Windows 2000 Active Directory zu lesen.

Konfigurieren von DB2 in einer IBM LDAP-Umgebung

Vor dem Einsatz von DB2 in einer IBM LDAP-Umgebung müssen Sie auf allen eingesetzten Maschinen Folgendes konfigurieren:

- Aktivieren Sie die LDAP-Unterstützung. Bei Windows 2000 wird die LDAP-Unterstützung durch das Installationsprogramm aktiviert. Bei Windows 98/NT müssen Sie LDAP explizit aktivieren. Setzen Sie hierzu mit dem Befehl **db2set** für die Variable `DB2_ENABLE_LDAP` der Profilregistrierdatenbank den Wert „YES“.
- TCP/IP-Host-Namen und Anschlussnummer des LDAP-Servers. Diese Werte können während einer nichtüberwachten Installation mit dem Antwortschlüsselwort `DB2LDAPHOST` eingegeben werden. Alternativ hierzu können Sie diese Werte auch zu einem späteren Zeitpunkt mit dem Befehl `DB2SET` manuell definieren:

```
db2set DB2LDAPHOST=<host_name[:anschluss]>
```

Hierbei steht `host_name` für den TCP/IP-Host-Namen des LDAP-Servers und `[:anschluss]` für die zugehörige Anschlussnummer. Wird keine Anschlussnummer angegeben, verwendet DB2 den LDAP-Standardanschluss (389).

DB2-Objekte werden unter dem registrierten LDAP-Basisnamen (`baseDN`) gespeichert. Wenn Sie die Version 3.1 des IBM SecureWay LDAP Directory Server verwenden, müssen Sie den registrierten Basisnamen nicht konfigurieren, da DB2 in der Lage ist, diese Informationen dynamisch vom Server abzurufen. Wenn Sie allerdings mit IBM eNetwork Directory Server Version 2.1 arbeiten, müssen Sie den registrierten LDAP-Basisnamen auf allen Maschinen konfigurieren. Verwenden Sie hierzu den Befehl `DB2SET` folgendermaßen:

```
db2set DB2LDAP_BASEDN=<baseDN>
```

Hierbei steht `baseDN` für den Namen des LDAP-Suffix, das auf dem LDAP-Server definiert wurde. Dieses LDAP-Suffix wird zur Speicherung von DB2-Objekten verwendet.

- Registrierter Name (DN) des LDAP-Benutzers sowie zugehöriges Kennwort. Diese Angaben sind nur erforderlich, wenn Sie LDAP zum Speichern von benutzerspezifischen DB2-Informationen verwenden wollen.

Erstellen eines LDAP-Benutzers

DB2 unterstützt das Definieren von DB2-Variablen der Profilregistrierdatenbank und die CLI-Konfiguration auf Benutzerebene. (Diese Funktion steht auf AIX- und Solaris-Plattformen nicht zur Verfügung.) Durch die Unterstützung auf Benutzerebene stehen in Mehrplatzsystemumgebungen benutzerspezifische Einstellungen zur Verfügung. Ein Beispiel ist der Windows NT Terminal Server, bei dem jeder angemeldete Benutzer seine Umgebung anpassen kann, ohne dass sich hierdurch Konflikte mit der Systemumgebung oder der Umgebung eines anderen Benutzers ergeben.

Beim Einsatz des IBM LDAP-Verzeichnisses müssen Sie einen LDAP-Benutzer definieren, bevor Informationen auf Benutzerebene unter LDAP gespeichert werden können. LDAP-Benutzer können folgendermaßen erstellt werden:

- Erstellen Sie eine LDIF-Datei, in der alle Attribute für das Benutzerobjekt enthalten sind. Führen Sie anschließend das LDIF-Importdienstprogramm aus, um das Objekt in das LDAP-Verzeichnis zu importieren. Das LDIF-Dienstprogramm für den IBM LDAP-Server trägt den Namen „LDIF2DB“.
- Verwenden Sie DMT (Directory Management Tool), das nur für IBM SecureWay LDAP Directory Server Version 3.1 zur Verfügung steht, um das Benutzerobjekt zu erstellen.

Eine LDIF-Datei mit den Attributen für einen Personenobjekt kann folgendes Format aufweisen:

```
File name: newuser.ldif
dn: cn=Mary Burnnet, ou=DB2 UDB Development, ou=Toronto, o=ibm, c=ca
objectclass: ePerson
cn: Mary Burnnet
sn: Burnnet
uid: mburnnet
userPassword: password
telephonenumber: 1-416-123-4567
facsimiletelephonenumber: 1-416-123-4568
title: Software Developer
```

Im Folgenden ist ein Beispiel für den Befehl LDIF zum Importieren einer LDIF-Datei mit dem IBM LDIF-Importdienstprogramm aufgeführt:

```
LDIF2DB -i newuser.ldif
```

Anmerkungen:

1. Der Befehl LDIF2DB muss auf der LDAP-Servermaschine ausgeführt werden.
2. Sie müssen die erforderlichen Zugriffsberechtigungen (ACL) für das LDAP-Benutzerobjekt erteilen, damit der LDAP-Benutzer sein eigenes Objekt hinzufügen, löschen, lesen und schreiben kann. Verwenden Sie das Tool LDAP Directory Server Web Administration, um für das Benutzerobjekt die Berechtigung ACL zu erteilen.

Konfigurieren des LDAP-Benutzers für DB2-Anwendungen

Beim Arbeiten mit dem IBM LDAP-Client und vor der Ausführung von DB2 müssen Sie den registrierten LDAP-Benutzernamen (DN) und das Kennwort für den aktuell angemeldeten Benutzer konfigurieren. Verwenden Sie hierzu das Dienstprogramm db2ldcfg:

```
db2ldcfg -u <userDN> -w <kennwort> -> Definieren  
des Benutzer-DN und des zugehörigen Kennworts  
-r -> Löschen des Benutzer-DN und des zugehörigen Kennworts
```

Beispiel:

```
db2ldcfg -u "cn=Mary Burnnet,ou=DB2 UDB Development,ou=Toronto,o=ibm,c=ca"  
-w kennwort
```

Registrierung von DB2-Servern nach der Installation

Jedes DB2-Serverexemplar muss in LDAP registriert werden, um die Protokollkonfigurationsdaten zu veröffentlichen, die von den Client-Anwendungen zum Herstellen einer Verbindung zu diesem DB2-Serverexemplar verwendet werden. Beim Registrieren eines Exemplars des Datenbankservers müssen Sie einen *Knotenname* angeben. Der Knotenname wird von Client-Anwendungen verwendet, wenn sie eine Verbindung zum Server herstellen. Sie können mit dem Befehl CATALOG LDAP NODE einen anderen Aliasnamen für den LDAP-Knoten katalogisieren.

Anmerkung: Wenn Sie in einer Domänenumgebung unter Windows 2000 arbeiten, wird das DB2-Serverexemplar während der Installation im Windows 2000 Active Directory automatisch mit den folgenden Informationen registriert:

```
nodename: TCP/IP hostname  
protocol type: TCP/IP
```

Wenn die Länge des TCP/IP-Host-Namens 8 Zeichen überschreitet, wird dieser auf 8 Zeichen abgeschnitten.

Der Befehl REGISTER lautet folgendermaßen:

```
db2 register db2 server in ldap
  as <ldap_knotenname>
  protocol tcpip
```

Die Klausel `protocol` gibt das Übertragungsprotokoll an, das zum Herstellen einer Verbindung zu diesem Datenbankserver verwendet wird.

Beim Erstellen eines Exemplars für DB2 Universal Database `EEE`, das mehrere physische Maschinen umfasst, muss der Befehl REGISTER für jede Maschine einmal aufgerufen werden. Verwenden Sie den Befehl `rah`, um den Befehl REGISTER auf allen Maschinen abzusetzen.

Anmerkung: In `ldap_knotenname` kann nicht derselbe Knotenname für alle Maschinen verwendet werden, da der Name in LDAP eindeutig sein muss. Sie müssen im Befehl REGISTER den Host-Namen jeder Maschine für `ldap_knotenname` verwenden. Beispiel:

```
rah ">DB2 REGISTER DB2 SERVER IN LDAP AS <> PROTOCOL TCP/IP"
```

„<>“ wird durch den Host-Namen jeder Maschine ersetzt, auf der der Befehl `rah` ausgeführt wird. Für den seltenen Fall, dass mehrere Exemplare von DB2 Universal Database `EEE` vorhanden sind, kann die Kombination aus Exemplar und Host-Index als Knotenname im Befehl `rah` verwendet werden.

Der Befehl REGISTER kann für einen fernen DB2-Server abgesetzt werden. Dazu müssen Sie den fernen Systemnamen, den Exemplarnamen und die Protokollkonfigurationsparameter beim Registrieren eines fernen Servers angeben. Der Befehl kann folgendermaßen verwendet werden:

```
db2 register db2 server in ldap
  as <ldap_knotenname>
  protocol tcpip
  hostname <host_name>
  svcname <tcpip_servicename>
  remote <ferner_systemname>
  instance <exemplarname>
```

Für den Computernamen gilt folgende Konvention:

- Wenn TCP/IP konfiguriert ist, muss der Systemname dem TCP/IP-Host-Namen entsprechen.
- Wurde APPN konfiguriert, müssen Sie als Computernamen den Namen der Partner-LU verwenden.

Beim Ausführen in einer Umgebung mit hoher Verfügbarkeit oder Funktionsübernahme (Failover) und bei Verwendung von TCP/IP als Übertragungsprotokoll muss die *Cluster-IP-Adresse* verwendet werden. Mit der Cluster-IP-Adresse kann der Client eine Verbindung zum Server auf einer beliebigen Maschine herstellen, ohne für jede Maschine einen separaten TCP/IP-Knoten katalogisieren zu müssen. Die Cluster-IP-Adresse wird mit der Klausel `hostname` wie folgt angegeben:

```
db2 register db2 server in ldap
  as <ldap_knotenname>
  protocol tcpip
  hostname n.nn.nn.nn
```

Dabei ist `n.nn.nn.nn` die Cluster-IP-Adresse.

Zusätzliche Informationen zum Befehl REGISTER finden Sie im Handbuch *Command Reference*.

Aktualisieren der Protokollinformationen für den DB2-Server

Die DB2-Serverinformationen in LDAP müssen immer aktuell sein. Änderungen an den Protokollkonfigurationsparametern oder der Server-Netzwerkadresse beispielsweise machen eine LDAP-Aktualisierung erforderlich.

Verwenden Sie den folgenden Befehl, um den DB2-Server in LDAP auf der lokalen Maschine zu aktualisieren:

```
db2 update ldap ...
```

Beispiele von aktualisierbaren Protokollkonfigurationsparametern sind unter anderem:

- TCP/IP-Host-Name und Servicename oder Anschlussnummernparameter
- IPX-Adresse
- APPC-Protokollinformationen TP-Name, Partner-LU oder Modus
- NetBIOS-Workstation-Name

Verwenden Sie den Befehl UPDATE LDAP mit der Klausel node, um die Protokollkonfigurationsparameter eines fernen DB2-Servers zu aktualisieren:

```
db2 update ldap
node <knotenname>
hostname <host_name>
svcname <tcip_servicename>
```

Weitere Informationen zum Befehl UPDATE LDAP finden Sie im Handbuch *Command Reference*.

Katalogisieren eines Knoten-Aliasnamens für ATTACH

Ein Knotenname für den DB2-Server muss beim Registrieren des Servers in LDAP angegeben werden. Anwendungen verwenden den Knotennamen zum Herstellen einer Verbindung zum Datenbankserver. Wenn ein anderer Knotenname erforderlich ist, z. B. wenn der Knotenname in einer Anwendung fest codiert ist, verwenden Sie den Befehl CATALOG LDAP NODE, um die Änderung vorzunehmen. Der Befehl würde etwa wie folgt aussehen:

```
db2 catalog ldap node <ldap_knotenname>
as <neuer_aliasname>
```

Verwenden Sie den Befehl UNCATALOG LDAP NODE, um einen LDAP-Knoten aus dem Katalog zu entfernen. Der Befehl würde etwa wie folgt aussehen:

```
db2 uncatalog ldap node <ldap_knotenname>
```

Zurücknehmen der Registrierung des DB2-Servers

Beim Zurücknehmen der Registrierung eines Exemplars aus LDAP werden auch alle Knoten- oder Aliasnamenobjekte sowie Datenbankobjekte entfernt, die auf das Exemplar verweisen.

Das Zurücknehmen der Registrierung des DB2-Servers auf einer lokalen oder fernen Maschine setzt voraus, dass der LDAP-Knotenname für den Server angegeben ist:

```
db2 deregister db2 server in ldap
node <knotenname>
```

Wenn die Registrierung des DB2-Servers zurückgenommen wird, werden alle LDAP-Knoteneinträge und LDAP-Datenbankeinträge, die auf dasselbe Exemplar im DB2-Server verweisen, ebenfalls aus dem Katalog entfernt.

Registrieren von Datenbanken

Bei der Erstellung einer Datenbank in einem Exemplar wird die Datenbank automatisch in LDAP registriert. Die Registrierung erlaubt Verbindungen ferner Clients zur Datenbank, ohne dass die Datenbank und der Knoten auf der Client-Maschine katalogisiert werden müssen. Wenn ein Client versucht, eine Verbindung zu einer Datenbank herzustellen, wird das LDAP-Verzeichnis durchsucht, wenn die Datenbank nicht im Datenbankverzeichnis auf der lokalen Maschine vorhanden ist.

Wenn der Name bereits im LDAP-Verzeichnis vorhanden ist, wird die Datenbank trotzdem auf der lokalen Maschine erstellt, allerdings mit einer Warnung bezüglich der Namensunverträglichkeit im LDAP-Verzeichnis. Aus diesem Grund können Sie eine Datenbank manuell im LDAP-Verzeichnis katalogisieren. Der Benutzer kann Datenbanken auf einem fernen Server in LDAP mit dem Befehl `CATALOG LDAP DATABASE` registrieren. Beim Registrieren einer fernen Datenbank geben Sie den Namen des LDAP-Knotens an, der den fernen Datenbankserver darstellt. Sie **müssen** den fernen Datenbankserver in LDAP mit dem Befehl `REGISTER DB2 SERVER IN LDAP` registrieren, **bevor** Sie die Datenbank registrieren.

Verwenden Sie den Befehl `CATALOG LDAP DATABASE`, um eine Datenbank manuell in LDAP zu registrieren:

```
db2 catalog ldap database <dbname>
    at node <knotenname>
    with "Meine LDAP-Datenbank"
```

Herstellen einer Verbindung zu einem fernen Server

In der LDAP-Umgebung können Sie eine Verbindung zu einem fernen Datenbankserver mit dem LDAP-Knotennamen im Befehl `ATTACH` herstellen:

```
db2 attach to <ldap_knotenname>
```

Wenn eine Client-Anwendung zum ersten Mal eine Verbindung zu einem Knoten oder einer Datenbank herstellt, durchsucht DB2 das LDAP-Verzeichnis nach dem Zielknoteneintrag, weil der Knoten im lokalen Knotenverzeichnis nicht definiert ist. Wenn der Eintrag im LDAP-Verzeichnis gefunden wird, werden die Protokollinformationen zum fernen Server abgerufen. Wenn Sie eine Verbindung zur Datenbank herstellen und wenn der Eintrag im LDAP-Verzeichnis gefunden wird, werden die Datenbankinformationen ebenfalls abgerufen. Mit Hilfe dieser Informationen katalogisiert DB2 automatisch einen Datenbankeintrag und einen Knoteneintrag auf der lokalen Maschine.

Wenn die Client-Anwendung das nächste Mal eine Verbindung zum selben Knoten oder zur selben Datenbank herstellt, werden die Informationen im lokalen Datenbankverzeichnis verwendet, so dass das LDAP-Verzeichnis nicht durchsucht werden muss.

Genauer ausgedrückt: Es gibt ein Caching-Verfahren, so dass der Client das LDAP-Verzeichnis in seinen lokalen Verzeichniskatalogen nur einmal durchsuchen muss. Sobald die Informationen abgerufen sind, werden sie auf der lokalen Maschine gespeichert bzw. im Cache zwischengespeichert. Bei nachfolgenden Zugriffsoperationen auf dieselben Informationen werden die Werte des Datenbankmanager-Konfigurationsparameters *dir_cache* und der Variablen DB2LDAPCACHE der Profilregistrierdatenbank zugrunde gelegt.

- Wurden DB2LDAPCACHE=NO und *dir_cache*=NO definiert, werden die Informationen immer über LDAP gelesen.
- Wurden DB2LDAPCACHE=NO und *dir_cache*=YES definiert, werden die Informationen einmal über LDAP gelesen und dann in den DB2-Cache eingefügt.
- Wurde DB2LDAPCACHE=YES definiert oder wurde die Variable nicht gesetzt, werden die Informationen über das LDAP-Verzeichnis gelesen und der lokale Cache wird aktualisiert, wenn die erforderlichen Informationen nicht im lokalen Cache gefunden werden können.

Anmerkung: Das Caching von LDAP-Informationen kann für CLI-Variablen oder DB2-Variablen der Profilregistrierdatenbank auf Benutzerebene nicht ausgeführt werden. Es gibt außerdem einen „speicherinternen“ Cache für die Datenbank, den Knoten und die DCS-Verzeichnisse. Für das Knotenverzeichnis steht ein solcher Cache jedoch nicht zur Verfügung.

Zurücknehmen der Registrierung der Datenbank

Die Registrierung der Datenbank in LDAP wird automatisch in den folgenden Fällen zurückgenommen:

- Die Datenbank wird gelöscht.
- Die Registrierung des Eignerexemplars in LDAP wird zurückgenommen.

Die Registrierung der Datenbank in LDAP kann mit dem folgenden Befehl manuell zurückgenommen werden:

```
db2 uncatalog ldap database <dbname>
```

Aktualisieren der LDAP-Einträge in lokalen Datenbank- und Knotenverzeichnissen

LDAP-Informationen unterliegen Änderungen, so dass es notwendig ist, die LDAP-Einträge in den lokalen Verzeichnissen und Knotenverzeichnissen zu aktualisieren. Die lokalen Datenbank- und Knotenverzeichnisse werden zum Zwischenspeichern der Einträge in LDAP verwendet.

Genauer ausgedrückt: Es gibt ein Caching-Verfahren, so dass der Client das LDAP-Verzeichnis in seinen lokalen Verzeichniskatalogen nur einmal durchsuchen muss. Sobald die Informationen abgerufen sind, werden sie auf der lokalen Maschine gespeichert bzw. im Cache zwischengespeichert. Bei nachfolgenden Zugriffsoperationen auf dieselben Informationen werden die Werte des Datenbankmanager-Konfigurationsparameters *dir_cache* und der Variablen DB2LDAPCACHE der Profilregistrierdatenbank zugrunde gelegt.

- Wurden DB2LDAPCACHE=NO und *dir_cache*=NO definiert, werden die Informationen immer über LDAP gelesen.
- Wurden DB2LDAPCACHE=NO und *dir_cache*=YES definiert, werden die Informationen einmal über LDAP gelesen und dann in den DB2-Cache eingefügt.
- Wurde DB2LDAPCACHE=YES definiert oder wurde die Variable nicht gesetzt, werden die Informationen über das LDAP-Verzeichnis gelesen und der lokale Cache wird aktualisiert, wenn die erforderlichen Informationen nicht im lokalen Cache gefunden werden können.

Anmerkung: Das Caching von LDAP-Informationen kann für CLI-Variablen oder DB2-Variablen der Profilregistrierdatenbank auf Benutzerebene nicht ausgeführt werden. Es gibt außerdem einen „speicherinternen“ Cache für die Datenbank, den Knoten und die DCS-Verzeichnisse. Für das Knotenverzeichnis steht ein solcher Cache jedoch nicht zur Verfügung.

Verwenden Sie den folgenden Befehl, um die Datenbankeinträge zu aktualisieren, die auf LDAP-Ressourcen verweisen:

```
db2 refresh ldap database directory
```

Verwenden Sie den folgenden Befehl, um die Knoteneinträge auf der lokalen Maschine zu aktualisieren, die auf LDAP-Ressourcen verweisen:

```
db2 refresh ldap node directory
```

Im Rahmen der Aktualisierung werden alle LDAP-Einträge entfernt, die in den lokalen Datenbank- und Knotenverzeichnissen gespeichert sind. Beim nächsten Zugriff der Anwendung auf die Datenbank oder den Knoten liest sie die Informationen direkt aus LDAP ab und generiert einen neuen Eintrag im lokalen Datenbank- oder Knotenverzeichnis.

Sie können wie folgt vorgehen, um eine rechtzeitige Aktualisierung sicherzustellen:

- Planen Sie eine in regelmäßigen Abständen durchgeführte Aktualisierung.
- Führen Sie den Befehl REFRESH beim Systemstart aus.
- Verwenden Sie ein verfügbares Verwaltungspaket, um den Befehl REFRESH auf allen Client-Maschinen aufzurufen.
- Definieren Sie DB2LDAPCACHE=„NO“, um das Caching von LDAP-Informationen in der Datenbank, dem Knoten und den DCS-Verzeichnissen zu verhindern.

Suchen

DB2 durchsucht die aktuelle LDAP-Verzeichnispartition oder die aktuelle Windows 2000 Active Directory-Domäne in der Windows 2000-Umgebung. In einer Umgebung mit mehreren LDAP-Verzeichnispartitionen oder -domänen können Sie den Suchbereich definieren. Wenn beispielsweise die Informationen nicht in der aktuellen Partition oder Domäne gefunden werden, kann eine automatische Durchsuchung aller anderen Partitionen oder Domänen angefordert werden. Der Suchbereich kann auch eingeschränkt werden, so dass nur die lokale Maschine durchsucht wird.

Der Suchbereich wird über die DB2-Variable DB2LDAP_SEARCH_SCOPE der Profilregistrierdatenbank gesteuert. Verwenden Sie den Parameter „-gl“, der „global in LDAP“ bedeutet, für den Befehl *db2set*, um den Wert des Suchbereichs auf der globalen Ebene in LDAP zu definieren:

```
db2set -gl db2ldap_search_scope=<wert>
```

Mögliche Werte sind: „local“, „domain“ oder „global“. Der Standardwert „domain“ grenzt den Suchbereich auf die aktuelle Verzeichnispaltung ein. Durch das Einstellen des Suchbereichs in LDAP kann der Standardsuchbereich für das gesamte Unternehmen eingestellt werden. Sie können z. B. den Suchbereich mit „global“ initialisieren, nachdem eine neue Datenbank erstellt wurde. Dadurch können Client-Maschinen alle anderen Partitionen oder Domänen durchsuchen, um eine Datenbank zu finden, die in einer bestimmten Partition oder Domäne definiert ist. Sobald der Eintrag nach der ersten Verbindung für jeden Client auf jeder Maschine aufgezeichnet ist, kann der Suchbereich in „local“ geändert werden. Sobald er in „local“ geändert ist, durchsuchen die Clients nicht mehr alle Partitionen oder Domänen.

Anmerkung: Die DB2-Variable DB2LDAP_SEARCH_SCOPE der Profilregistrierdatenbank ist die einzige Variable der Profilregistrierdatenbank, die das Setzen einer Variablen auf globaler Ebene unter LDAP unterstützt.

Registrieren der Host-Datenbanken

Beim Registrieren von Host-Datenbanken unter LDAP gibt es zwei Konfigurationsmöglichkeiten:

- Direkte Verbindung zu den Host-Datenbanken
- Verbindung zur Host-Datenbank über einen Gateway

Im ersten Fall registriert der Benutzer den Host-Server unter LDAP. Anschließend katalogisiert er die Host-Datenbank unter LDAP und gibt hierbei den Knotennamen des Host-Servers an. Im zweiten Fall registriert der Benutzer den Gateway-Server unter LDAP. Anschließend katalogisiert er die Host-Datenbank unter LDAP und gibt hierbei den Knotennamen des Gateway-Servers an.

Im Folgenden ist ein Beispiel für beide Fälle aufgeführt: Sie verfügen über die Host-Datenbank NIAGARA_FALLS. Diese Datenbank kann eingehende Verbindungen über APPN und TCP/IP akzeptieren. Wenn der Client keine direkte Verbindung zum Host herstellen kann, weil er nicht über DB2 Connect verfügt, erfolgt der Verbindungsaufbau über einen Gateway mit dem Namen „goto@niagara“.

Sie müssen die folgenden Arbeitsschritte ausführen:

1. Registrieren Sie den Host-Datenbankserver unter LDAP für die APPN-Konnektivität. Die Klauseln REMOTE und INSTANCE sind hierbei frei wählbar. Die Klausel NODETYPE wird auf den Wert „DCS“ gesetzt, um anzugeben, dass es sich bei der Einheit um einen Host-Datenbankserver handelt.

```
db2 register ldap as nfappn appn network CAIBMOML partnerlu NFLU
mode IBMRDB remote mvssys instance msvinst nodetype dcs
```

2. Registrieren Sie den Host-Datenbankserver unter LDAP für die TCP/IP-Konnektivität. Der TCP/IP-Host-Name des Servers ist „myhost“ und die Anschlussnummer ist „446“. Ähnlich wie in Schritt 1 wird die Klausel NODETYPE auf den Wert „DCS“ gesetzt, um anzugeben, dass es sich bei der Einheit um einen Host-Datenbankserver handelt.

```
db2 register ldap as nftcpip tcpip hostname myhost svcname 446
remote mvssys instance mvsinst nodetype dcs
```

3. Registrieren Sie den DB2 Connect-Gateway-Server unter LDAP für die TCP/IP-Konnektivität. Der TCP/IP-Host-Name für den Gateway-Server ist „niagara“ und die Anschlussnummer ist „50000“.

```
db2 register ldap as whasf tcpip hostname niagara svcname 50000
remote niagara instance goto nodetype server
```

4. Katalogisieren Sie die Host-Datenbank unter LDAP und verwenden Sie hierbei die TCP/IP-Konnektivität. Die Host-Datenbank trägt den Namen „NIAGARA_FALLS“, der Aliasname der Datenbank lautet „nftcpip“. Die Klausel GWNODE wird verwendet, um den Knotennamen des DB2 Connect-Gateway-Servers anzugeben.

```
db2 catalog ldap database NIAGARA_FALLS as nftcpip at node nftcpip
gwnode whasf authentication dcs
```

5. Katalogisieren Sie die Host-Datenbank unter LDAP und verwenden Sie hierbei die APPN-Konnektivität.

```
db2 catalog ldap database NIAGARA_FALLS as nfappn at node nfappn
gwnode whasf authentication dcs
```

Nach dem Abschluss der Registrierung und Katalogisierung (siehe oben) können Sie eine Verbindung zu „nftcpip“ herstellen, wenn Sie unter TCP/IP auf den Host zugreifen wollen. Wenn Sie unter APPN auf den Host zugreifen wollen, stellen Sie eine Verbindung zu „nfappn“ her. Wenn auf Ihrer Client-Workstation DB2 Connect nicht installiert ist, wird die Verbindung über den Gateway unter TCP/IP hergestellt. Abhängig davon, ob Sie „nftcpip“ oder „nfappn“ verwenden, stellt der Gateway dann unter TCP/IP oder APPN eine Verbindung zum gewünschten Host her.

Sie können Angaben zur Host-Datenbank in LDAP normalerweise manuell konfigurieren, so dass die Clients die Datenbank und den Knoten nicht lokal auf jeder Maschine katalogisieren müssen. Gehen Sie dazu wie folgt vor:

1. Registrieren Sie den Host-Datenbankserver in LDAP. Dazu müssen Sie den Namen des fernen Computers, den Exemplarnamen und die Knotenart für den Host-Datenbankserver im Befehl REGISTER angeben, wobei Sie die Klausel REMOTE, INSTANCE oder NODETYPE verwenden. Die Klausel REMOTE kann entweder auf den Host-Namen oder den LU-Namen der Host-Servermaschine gesetzt werden. Die Klausel INSTANCE kann auf eine beliebige Zeichenfolge von höchstens acht Zeichen gesetzt werden. (Der Exemplarname kann z. B. auf „DB2“ gesetzt werden.) Die Klausel NODE TYPE muss auf „DCS“ gesetzt werden, um anzugeben, dass es sich bei der Einheit um einen Host-Datenbankserver handelt.
2. Registrieren Sie die Host-Datenbank in LDAP mit dem Befehl CATALOG LDAP DATABASE. Weitere DRDA-Parameter können mit der Klausel PARMS angegeben werden. Die Datenbankauthentifizierungsart sollte auf „DCS“ gesetzt sein.

Einstellen der DB2-Variablen der Profilregistrierdatenbank auf Benutzerebene

In der LDAP-Umgebung können die DB2-Variablen der Profilregistrierdatenbank auf Benutzerebene eingestellt werden. Dadurch können Benutzer ihre eigene DB2-Umgebung anpassen. Um die DB2-Variablen der Profilregistrierdatenbank auf Benutzerebene zu setzen, verwenden Sie die Option `-ul`:

```
db2set -ul <variable>=<wert>
```

Anmerkung: Diese Funktion wird unter AIX und Solaris nicht unterstützt.

DB2 verfügt über einen Zwischenspeicherungsmechanismus (Caching). Die DB2-Variablen der Profilregistrierdatenbank auf Benutzerebene werden auf der lokalen Maschine zwischengespeichert. Wenn der Parameter `-ul` angegeben wird, liest DB2 die DB2-Variablen der Profilregistrierdatenbank immer aus dem Cache. Der Cache wird in folgenden Fällen aktualisiert:

- Sie aktualisieren eine DB2-Variable der Profilregistrierdatenbank auf Benutzerebene oder setzen sie zurück.
- Der Befehl zum Aktualisieren der LDAP-Profilvariablen auf Benutzerebene lautet wie folgt:

```
db2set -ur
```

Aktivieren der LDAP-Unterstützung nach abgeschlossener Installation

Verwenden Sie auf allen Maschinen die folgende Prozedur, um die LDAP-Unterstützung nach dem Abschluss des Installationsprozesses zu aktivieren:

- Installieren Sie die Binärdateien für die LDAP-Unterstützung. Führen Sie das Konfigurationsprogramm aus, und wählen Sie die Unterstützung für LDAP-Verzeichnisausnutzung bei der benutzerdefinierten Installation aus. Das Installationsprogramm installiert die Binärdateien und setzt die DB2-Variable `DB2_ENABLE_LDAP` der Profilregistrierdatenbank auf „YES“.

Anmerkung: Bei Windows 98/NT und auf UNIX-Plattformen müssen Sie LDAP explizit aktivieren. Definieren Sie hierzu für die Variable `DB2_ENABLE_LDAP` der Profilregistrierdatenbank mit dem Befehl **db2set** den Wert „YES“.

- (Nur UNIX-Plattformen) Deklarieren Sie mit dem folgenden Befehl den TCP/IP-Host-Namen des LDAP-Servers und (wahlfrei) die Anschlussnummer:

```
db2set  
DB2LDAPHOST=<basisdomeinname>[:anschlussnummer]
```

Hierbei steht `basisdomeinname` für den TCP/IP-Host-Namen des LDAP-Servers und `[:anschlussnummer]` für die zugehörige Anschlussnummer. Wird keine Anschlussnummer angegeben, verwendet DB2 den LDAP-Standardanschluss (389).

DB2-Objekte werden unter dem registrierten LDAP-Basisnamen (`baseDN`) gespeichert. Wenn Sie die Version 3.1 des IBM SecureWay LDAP Directory Server verwenden, müssen Sie den registrierten Basisnamen nicht konfigurieren, da DB2 in der Lage ist, diese Informationen dynamisch vom Server abzurufen. Wenn Sie allerdings mit IBM eNetwork Directory Server Version 2.1 arbeiten, müssen Sie den registrierten LDAP-Basisnamen auf allen Maschinen konfigurieren. Verwenden Sie hierzu den Befehl `DB2SET` folgendermaßen:

```
db2set DB2LDAP_BASEDN=<baseDN>
```

Hierbei steht `baseDN` für den Namen des LDAP-Suffix, das auf dem LDAP-Server definiert wurde. Dieses LDAP-Suffix wird zur Speicherung von DB2-Objekten verwendet.

- Registrieren Sie das aktuelle Exemplar des DB2-Servers in LDAP mit dem Befehl `REGISTER LDAP AS`. Beispiel:

```
db2 register ldap as <knotenname> protocol tcpip
```

- Wenn Sie Datenbanken haben, die Sie in LDAP registrieren wollen, führen Sie den Befehl `CATALOG LDAP DATABASE` aus. Beispiel:

```
db2 catalog ldap database <dbname> as
<alias_dbname>
```
- Geben Sie den registrierten Namen des LDAP-Benutzers (DN) sowie das zugehörige Kennwort ein. Diese Angaben sind nur erforderlich, wenn Sie LDAP zum Speichern von benutzerspezifischen DB2-Informationen verwenden wollen.

Inaktivieren von LDAP-Unterstützung

Verwenden Sie die folgende Prozedur, um die LDAP-Unterstützung zu inaktivieren:

- Nehmen Sie für jedes Exemplar des DB2-Servers die Registrierung in LDAP zurück:

```
db2 deregister db2 server in ldap node <knotenname>
```
- Setzen Sie die DB2-Variablen `DB2_ENABLE_LDAP` der Profilregistrierdatenbank auf den Wert „NO“.

LDAP-Unterstützung und DB2 Connect

Wenn die LDAP-Unterstützung auf dem DB2 Connect-Gateway zur Verfügung steht und die Datenbank im Gateway-Datenbankverzeichnis nicht gefunden werden kann, sucht DB2 nach LDAP und versucht, die gefundenen Informationen zu halten.

Überlegungen zur Sicherheit

Vor dem Zugriff auf Informationen im LDAP-Verzeichnis wird eine Anwendung oder ein Benutzer vom LDAP-Server authentifiziert. Die Authentifizierung wird *Binden* an den LDAP-Server genannt.

Für die im LDAP-Verzeichnis gespeicherten Informationen muss Zugriffssteuerung angewendet werden, um zu verhindern, dass anonyme Benutzer die Informationen löschen, ändern oder Informationen hinzufügen.

Die Zugriffssteuerung wird standardmäßig übernommen und kann auf Behälterebene angewendet werden. Wenn ein neues Objekt erstellt wird, übernimmt es dasselbe Sicherheitsattribut wie das Elternobjekt. Ein für den LDAP-Server verfügbares Verwaltungs-Tool kann zum Definieren der Zugriffssteuerung für das Behälterobjekt verwendet werden.

Standardmäßig wird die Zugriffssteuerung wie folgt definiert:

- Auf Datenbank- und Knoteneinträge in LDAP hat jeder (auch jeder anonyme) Benutzer Lesezugriff. Nur der Verzeichnisadministrator und der Eigentümer oder Ersteller des Objekts hat Schreib-/Lesezugriff.
- Auf Benutzerprofile haben der Profileigner und der Verzeichnisadministrator Schreib-/Lesezugriff. Ein Benutzer kann nicht auf das Profil eines anderen Benutzers zugreifen, wenn er keine Verzeichnisadministratorberechtigung hat.

Anmerkung: Die Berechtigungsprüfung wird immer vom LDAP-Server und nicht von DB2 durchgeführt. Die LDAP-Berechtigungsprüfung ist unabhängig von der DB2-Berechtigung. Ein Konto oder eine Berechtigungs-ID mit der Berechtigung SYSADM hat möglicherweise keinen Zugriff auf das LDAP-Verzeichnis.

Wird bei der Ausführung der LDAP-Befehle oder -APIs kein registrierter Bindename (bindDN) und kein Kennwort angegeben, führt DB2 eine Bindeoperation mit dem LDAP-Server durch und verwendet hierbei die Standardidentitätsnachweise, denen möglicherweise nicht die ausreichenden Berechtigungen zur Ausführung der angeforderten Befehle zugeordnet wurden. In diesem Fall wird ein Fehler zurückgegeben.

Sie können den registrierten Bindenamen (bindDN) und das Kennwort eines Benutzers mit den Klauseln USER und PASSWORD der DB2-Befehle und -APIs explizit angeben. Weitere Informationen zu DB2-Befehlen finden Sie im Handbuch *Command Reference*, weitere Informationen zu DB2-APIs enthält das Handbuch *Administrative API Reference*.

Überlegungen zur Sicherheit bei Windows 2000 Active Directory

Die DB2-Datenbank- und -Knotenobjekte werden unter dem Computerobjekt der Maschine erstellt, auf der der DB2-Server im Windows 2000 Active Directory installiert ist. Zum Registrieren eines Datenbankservers oder Katalogisieren im Windows 2000 Active Directory müssen Sie über ausreichende Zugriffsberechtigungen zum Erstellen und/oder Aktualisieren der Objekte verfügen, die unter dem Computerobjekt gespeichert sind.

Standardmäßig können die unter einem Computerobjekt gespeicherten Objekte von allen authentifizierten Benutzern gelesen und von den zugehörigen Administratoren (d. h. Benutzern, die zur Administrator-, Domänenadministrator- und Unternehmensadministratorgruppe gehören) aktualisiert werden. Um einem bestimmten Benutzer bzw. einer bestimmten Gruppe Zugriffsberechtigungen zu erteilen, verwenden Sie die Microsoft Management Console (MMC) für *Active Directory Users and Computer* folgendermaßen:

1. Starten Sie das Verwaltungs-Tool für *Active Directory Users and Computer*.
(Start—> Program—> Administration Tools—> Active Directory Users and Computer)
2. Wählen Sie unter *View* die Option *Advanced Features* aus.
3. Wählen Sie den Behälter *Computers* aus.
4. Klicken Sie mit der rechten Maustaste auf dem Computerobjekt, das für die Server-Maschine steht, auf der DB2 installiert ist, und wählen Sie dann *Properties* aus.
5. Wählen Sie die Indexzunge *Security* aus und fügen Sie für die angegebene Gruppe bzw. den angegebenen Benutzer die erforderliche Zugriffsberechtigung hinzu.

Die DB2-Variablen der Profilregistrierdatenbank und CLI-Einstellungen auf Benutzerebene werden im DB2-Merkmalobjekt unter dem Benutzerobjekt verwaltet. Um die DB2-Variablen der Profilregistrierdatenbank oder CLI-Einstellungen auf Benutzerebene zu definieren, muss ein Benutzer über ausreichende Zugriffsberechtigungen verfügen, um Objekte unter dem Benutzerobjekt zu erstellen.

Standardmäßig verfügen nur Administratoren über die Zugriffsberechtigung zum Erstellen von Objekten unter dem Benutzerobjekt. Um einem Benutzer die Zugriffsberechtigung zum Definieren von DB2-Variablen der Profilregistrierdatenbank oder CLI-Einstellungen auf Benutzerebene zu erteilen, können Sie die Microsoft Management Console (MMC) für *Active Directory Users and Computer* folgendermaßen verwenden:

1. Starten Sie das Verwaltungs-Tool für *Active Directory Users and Computer*.
(Start—> Program—> Administration Tools—> Active Directory Users and Computer)
2. Wählen Sie das Benutzerobjekt im Behälter 'Users' aus.
3. Klicken Sie mit der rechten Maustaste auf dem Benutzerobjekt und wählen Sie dann *Properties* aus.
4. Wählen Sie die Indexzunge *Security* aus.
5. Fügen Sie den Benutzernamen mit Hilfe des Knopfes 'Add' der Liste hinzu.
6. Erteilen Sie die Zugriffsberechtigungen „Write“ und „Create All Child Objects“.
7. In der erweiterten Einstellung (Advanced Setting) erlauben Sie die Anwendung auf „This object and all child objects“.
8. Wählen Sie das Markierungsfeld „Allow inheritable permissions from parent to propagate to this object“ aus.

Erweitern des Verzeichnisseschemas mit DB2-Objektklassen und -Attributen

Das LDAP-Verzeichnisseschema definiert Objektklassen und Attribute für die in den LDAP-Verzeichniseinträgen gespeicherten Informationen. Eine Objektklasse besteht aus einer Reihe von verbindlichen und wahlfreien Attributen. Jedem Eintrag im LDAP-Verzeichnis ist eine Objektklasse zugeordnet.

Bevor DB2 die Informationen in LDAP speichern kann, muss das Verzeichnisseschema für den LDAP-Server die von DB2 verwendeten Objektklassen und Attribute enthalten. Das Hinzufügen neuer Objektklassen und Attribute zum Basisschema wird Erweitern des Verzeichnisseschemas genannt.

Anmerkung: Wenn Sie IBM SecureWay LDAP Directory Version 3.1 verwenden, werden alle von DB2 erforderlichen Objektklassen und Attribute in das Basisschema aufgenommen. Sie müssen das Basisschema nicht mit DB2-Objektklassen und -Attributen erweitern.

Erweitern des Verzeichnisseschemas für IBM eNetwork Directory Version 2.1

Wenn Sie IBM eNetwork Directory Version 2.1 verwenden, müssen Sie das Basisschema mit den von DB2 verwendeten Objektklassen und Attributen erweitern.

Mit den folgenden Schritten können Sie das Basisschema für IBM eNetwork Directory Version 2.1 erweitern:

1. Kopieren Sie die DB2-Attributdefinitionsdatei, `db2.at`, und die Objektklassendefinitionsdatei, `db2.oc`, in das Verzeichnis, das die Systemattribut- und Objektklassendefinitionsdateien, `slapd.at.conf` und `slapd.oc.conf`, enthält. Die DB2-Attribut- und -Objektklassendefinitionsdateien befinden sich im Unterverzeichnis `cfg` des Unterverzeichnisses `sql1ib`. Die Systemattribut- und Objektklassendefinitionsdatei befinden sich im Unterverzeichnis `etc` des Unterverzeichnisses `%LDAPHome%`.
2. Prüfen Sie die DB2-Attribut- und -Objektklassendefinitionsdateien. Setzen Sie alle Objektklassen und Attribute auf Kommentar, die in Ihrem aktuellen LDAP-Verzeichnisseschema definiert sind.
3. Fügen Sie die folgende Zeile am Ende der Datei `slapd.oc.conf` hinzu:
`include db2.oc`
4. Fügen Sie die folgende Zeile am Ende der Datei `slapd.at.conf` hinzu:
`include db2.at`
5. Starten Sie den LDAP-Server neu.

Erweitern des Verzeichnisschemas für Windows 2000 Active Directory

Bevor DB2 Informationen im Windows 2000 Active Directory speichern kann, muss das Verzeichnisschema so erweitert werden, dass es die neuen DB2-Objektklassen und -Attribute umfasst. Das Hinzufügen neuer Objektklassen und Attribute zum Verzeichnisschema wird als *Schemaerweiterung* bezeichnet.

Vor der Erstinstallation von DB2 auf den Maschinen einer Windows 2000-Domäne müssen Sie das DB2-Programm **db2schex** für die Schemainstallation ausführen, um das Schema für Windows 2000 Active Directory zu erweitern.

Das Programm **db2schex** befindet sich auf der CD-ROM mit dem Produkt. Das Programm ist auf der CD-ROM im Verzeichnis db2 im Unterverzeichnis common gespeichert. Beispiel:

```
x:\db2\common
```

Hierbei steht x: für das CD-ROM-Laufwerk.

Der Befehl ist wie folgt anzugeben:

```
db2schex
```

Diesem Befehl sind folgende wahlfreie Klauseln zugeordnet:

- -b UserDN
Zur Angabe des registrierten Benutzernamens.
- -w Password
Zur Angabe des BIND-Kennwortes.
- -u
Zum Deinstallieren des Schemas.
- -k
Zum erzwungenen Fortsetzen der Deinstallation und Ignorieren von Fehlermeldungen.

Anmerkungen:

1. Wurden kein registrierter Benutzername (UserDN) und kein Kennwort angegeben, führt **db2schex** die Bindeoperation mit dem momentan angemeldeten Benutzer durch.
2. Die Klausel userDN kann als Windows NT-Benutzername angegeben werden.
3. Zum Aktualisieren des Schemas müssen Sie Mitglied der Schemaadministratorgruppe sein oder über die Berechtigungen zum Aktualisieren des Schemas verfügen.

Beispiele:

- Geben Sie Folgendes ein, um das DB2-Schema zu installieren:
db2schex
- Geben Sie Folgendes ein, um das DB2-Schema zu installieren und einen registrierten Bindenamen (bindDN) und ein Kennwort zu definieren:
db2schex -b "cn=A Name,dc=toronto1,dc=ibm,dc=com"
-w password

Alternativ können Sie auch Folgendes eingeben:

- db2schex -b Administrator -w password
- Geben Sie Folgendes ein, um das DB2-Schema zu deinstallieren:
db2schex -u
- Geben Sie Folgendes ein, um das DB2-Schema zu deinstallieren und Fehlermeldungen zu ignorieren:
db2schex -u -k

Das DB2-Schemainstallationsprogramm für Windows 2000 Active Directory führt die folgenden Tasks aus:

Anmerkungen:

1. Feststellen des Servers, der als Schema-Master arbeitet
2. Durchführen einer Bindeoperation zum Domänen-Controller, der als Schema-Master arbeitet
3. Sicherstellen ausreichender Benutzerberechtigungen zum Hinzufügen von Klassen und Attributen zum Schema
4. Sicherstellen der Unterstützung von Schreiboperationen auf dem Schema-Master (hierzu muss die Sicherheitssperre in der Registrierdatenbank entfernt werden)
5. Erstellen aller neuen Attribute
6. Erstellen aller neuen Objektklassen
7. Feststellen von Fehlern und ggf. Zurücksetzen (ROLLBACK) von am Schema vorgenommenen Änderungen

DB2-Objekte im Windows 2000 Active Directory

DB2 erstellt Objekte im Windows 2000 Active Directory an den beiden folgenden Positionen:

1. Die DB2-Datenbank- und -Knotenobjekte werden unter dem Computerobjekt der Maschine erstellt, auf der der DB2-Server installiert ist. Für die DB2-Server-Maschine, die nicht zur Windows NT-Domäne gehört, werden die DB2-Datenbank- und -Knotenobjekte im Behälter „System“ erstellt.
2. Die DB2-Variablen der Profilregistrierdatenbank und CLI-Einstellungen auf Benutzerebene werden in den DB2-Merkmalobjekten unter dem Benutzerobjekt gespeichert. Diese Objekte enthalten benutzerspezifische Informationen.

Von DB2 verwendete Objektklassen und Attribute

In den folgenden Tabellen werden die von DB2 verwendeten Objektklassen beschrieben:

Tabelle 27. cimManagedElement

Objektklasse	cimManagedElement
Active Directory-LDAP-Anzeigename	Nicht verfügbar
Allgemeiner Active Directory-Name (cn)	Nicht verfügbar
Beschreibung	Stellt eine Basisklasse für viele der Systemverwaltungsobjektklassen im IBM Schema bereit
SubClassOf	top
Erforderliche(s) Attribut(e)	
Wahlfreie(s) Attribut(e)	description
Art	abstract
OID (Objekt-ID)	1.3.18.0.2.6.132
GUID (globale eindeutige Kennung)	b3afd63f-5c5b-11d3-b818-002035559151

Tabelle 28. cimSetting

Objektklasse	cimSetting
Active Directory-LDAP-Anzeigename	Nicht verfügbar
Allgemeiner Active Directory-Name (cn)	Nicht verfügbar
Beschreibung	Stellt eine Basisklasse für die Konfiguration und Einstellungen im IBM Schema zur Verfügung
SubClassOf	cimManagedElement
Erforderliche(s) Attribut(e)	
Wahlfreie(s) Attribut(e)	settingID
Art	abstract
OID (Objekt-ID)	1.3.18.0.2.6.131
GUID (globale eindeutige Kennung)	b3afd64d-5c5b-11d3-b818-002035559151

Tabelle 29. eProperty

Objektklasse	eProperty
Active Directory-LDAP-Anzeigename	ibm-eProperty
Allgemeiner Active Directory-Name (cn)	ibm-eProperty
Beschreibung	Wird zur Angabe von anwendungsspezifischen Einstellungen für Benutzervorgabemerkmale verwendet

Table 29. eProperty (Forts.)

Objektklasse	eProperty
SubClassOf	cimSetting
Erforderliche(s) Attribut(e)	
Wahlfreie(s) Attribut(e)	propertyType cisPropertyType cisProperty cesPropertyType cesProperty binPropertyType binProperty
Art	structural
OID (Objekt-ID)	1.3.18.0.2.6.90
GUID (globale eindeutige Kennung)	b3afd69c-5c5b-11d3-b818-002035559151

Table 30. DB2Node

Objektklasse	DB2Node
Active Directory-LDAP-Anzeigename	ibm-db2Node
Allgemeiner Active Directory-Name (cn)	ibm-db2Node
Beschreibung	Steht für einen DB2-Server
SubClassOf	eSap / ServiceConnectionPoint
Erforderliche(s) Attribut(e)	db2nodeName
Wahlfreie(s) Attribut(e)	db2nodeAlias db2instanceName db2Type host / dNSHostName (siehe Anm. 2) protocolInformation/ServiceBindingInformation
Art	structural
OID (Objekt-ID)	1.3.18.0.2.6.116
GUID (globale eindeutige Kennung)	b3afd65a-5c5b-11d3-b818-002035559151

Tabelle 30. *DB2Node* (Forts.)

Objektklasse	DB2Node
Besondere Hinweise	<ol style="list-style-type: none"> 1. Die Klasse <i>DB2Node</i> wird von der Objektklasse <i>eSap</i> im IBM SecureWay-Verzeichnis und von der Objektklasse <i>ServiceConnectionPoint</i> im Microsoft Active Directory abgeleitet. 2. Das Attribut <i>host</i> wird in der IBM SecureWay-Umgebung verwendet. Das Attribut <i>dNSHostName</i> wird beim Microsoft Active Directory verwendet. 3. Das Attribut <i>protocolInformation</i> wird nur in der IBM SecureWay-Umgebung verwendet. Bei Microsoft Active Directory wird das Attribut <i>ServiceBindingInformation</i>, das von der Klasse <i>ServiceConnectionPoint</i> übernommen wurde, zur Speicherung der Protokollinformationen verwendet.

Das Attribut *protocolInformation* (bei IBM SecureWay Directory) oder *ServiceBindingInformation* (bei Microsoft Active Directory) im Objekt *DB2Node* enthält die Informationen zum Kommunikationsprotokoll für das Binden des DB2-Datenbankservers. Es besteht aus Token, die das unterstützte Netzwerkprotokoll beschreiben. Die Token werden durch ein Semikolon getrennt. Zwischen den Token steht kein Leerzeichen. Zur Angabe eines wahlfreien Parameters wird ein Stern (*) benutzt.

Es gibt folgende Token für TCP/IP:

- „TCPIP“
- Server-Host-Name oder IP-Adresse
- Servicename (svcname) oder Anschlussnummer (z. B. 50000)
- (Wahlfrei) Sicherheit („NONE“ oder „SOCKS“)

Es gibt folgende Token für APPN:

- „APPN“
- Netzwerk-ID
- Partner-LU
- TP-Name (TP = Transaktionsprogramm). Es werden nur Transaktionsprogramme für Anwendungen, nicht jedoch Servicetransaktionsprogramme oder Angaben für Transaktionsprogramme im Hexadezimalformat unterstützt.
- Modus
- Sicherheit (entweder „NONE“, „SAME“ oder „PROGRAM“)

- (Wahlfrei) LAN-Adapteradresse
- (Wahlfrei) LU zur Kennwortänderung

Anmerkung: Wenn auf einem DB2 für Windows NT-Client (oder unter Windows 98) die APPN-Informationen nicht im lokalen SNA-Stack konfiguriert und die LAN-Adapteradresse und die wahlfreie LU zur Kennwortänderung in LDAP lokalisiert wurden, versucht der DB2-Client, diese Informationen zum Konfigurieren des SNA-Stacks zu verwenden, wenn alle verfügbaren Angaben zum Konfigurieren des Stacks vorhanden sind. Diese Unterstützungsfunktion steht für DB2 für AIX- bzw. DB2 für Solaris-Clients nicht zur Verfügung.

Es gibt folgende Token für IPX/SPX:

- „IPXSPX“
- IPX-Adresse

Die IPX/SPX-Empfangsfunktion steht auf dem DB2-Server (nicht jedoch auf dem Client) für die Betriebssysteme AIX und Solaris zur Verfügung. NetBIOS und NPIPE werden unter AIX und Solaris nicht unterstützt.

Es gibt folgende Token für NetBIOS:

- „NETBIOS“
- Server-NetBIOS-Workstation-Name

Es gibt folgende Token für benannte Pipes:

- „NPIPE“
- Computername des Servers
- Exemplarname des Servers

Tabelle 31. DB2Database

Objektklasse	DB2Database
Active Directory-LDAP-Anzeigename	ibm-db2Database
Allgemeiner Active Directory-Name (cn)	ibm-db2Database
Beschreibung	Steht für eine DB2-Datenbank
SubClassOf	top
Erforderliche(s) Attribut(e)	db2databaseName db2nodePtr

Tabelle 31. DB2Database (Forts.)

Objektklasse	DB2Database
Wahlfreie(s) Attribut(e)	db2databaseAlias db2additionalParameters db2ARLibrary db2authenticationLocation db2gwPtr db2databaseRelease DCEPrincipalName
Art	structural
OID (Objekt-ID)	1.3.18.0.2.6.117
GUID (globale eindeutige Kennung)	b3afd659-5c5b-11d3-b818-002035559151

Tabelle 32. db2additionalParameters

Attribut	db2additionalParameters
Active Directory-LDAP-Anzeigename	ibm-db2AdditionalParameters
Allgemeiner Active Directory-Name (cn)	ibm-db2AdditionalParameters
Beschreibung	Enthält alle zusätzlichen Parameter, die zum Herstellen einer Verbindung zum Host-Datenbankserver verwendet werden
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	1024
Werte	Mit einem Wert
OID (Objekt-ID)	1.3.18.0.2.4.426
GUID (globale eindeutige Kennung)	b3afd315-5c5b-11d3-b818-002035559151

Tabelle 33. db2authenticationLocation

Attribut	db2authenticationLocation
Active Directory-LDAP-Anzeigename	ibm-db2AuthenticationLocation
Allgemeiner Active Directory-Name (cn)	ibm-db2AuthenticationLocation
Beschreibung	Gibt an, wo die Authentifizierung stattfindet
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge

Tabelle 33. db2authenticationLocation (Forts.)

Attribut	db2authenticationLocation
Maximale Länge	64
Werte	Mit einem Wert
OID (Objekt-ID)	1.3.18.0.2.4.425
GUID (globale eindeutige Kennung)	b3afd317-5c5b-11d3-b818-002035559151
Hinweise	Zulässige Werte sind: CLIENT, SERVER, DCS, DCE, KERBEROS, SVRENCRYPT oder DCSRENCRYPT

Tabelle 34. db2ARLibrary

Attribut	db2ARLibrary
Active Directory-LDAP-Anzeigename	ibm-db2ARLibrary
Allgemeiner Active Directory-Name (cn)	ibm-db2ARLibrary
Beschreibung	Name der Anwendungs-Requester-Bibliothek
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	256
Werte	Mit einem Wert
OID (Objekt-ID)	1.3.18.0.2.4.427
GUID (globale eindeutige Kennung)	b3afd316-5c5b-11d3-b818-002035559151

Tabelle 35. db2databaseAlias

Attribut	db2databaseAlias
Active Directory-LDAP-Anzeigename	ibm-db2DatabaseAlias
Allgemeiner Active Directory-Name (cn)	ibm-db2DatabaseAlias
Beschreibung	Aliasname(n) der Datenbank
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	1024
Werte	Mit mehreren Werten
OID (Objekt-ID)	1.3.18.0.2.4.422
GUID (globale eindeutige Kennung)	b3afd318-5c5b-11d3-b818-002035559151

Tabelle 36. db2databaseName

Attribut	db2databaseName
Active Directory-LDAP-Anzeigename	ibm-db2DatabaseName
Allgemeiner Active Directory-Name (cn)	ibm-db2DatabaseName
Beschreibung	Datenbankname
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	1024
Werte	Mit einem Wert
OID (Objekt-ID)	1.3.18.0.2.4.421
GUID (globale eindeutige Kennung)	b3afd319-5c5b-11d3-b818-002035559151

Tabelle 37. db2databaseRelease

Attribut	db2databaseRelease
Active Directory-LDAP-Anzeigename	ibm-db2DatabaseRelease
Allgemeiner Active Directory-Name (cn)	ibm-db2DatabaseRelease
Beschreibung	Datenbank-Release-Nummer
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	64
Werte	Mit einem Wert
OID (Objekt-ID)	1.3.18.0.2.4.429
GUID (globale eindeutige Kennung)	b3afd31a-5c5b-11d3-b818-002035559151

Tabelle 38. db2nodeAlias

Attribut	db2nodeAlias
Active Directory-LDAP-Anzeigename	ibm-db2NodeAlias
Allgemeiner Active Directory-Name (cn)	ibm-db2NodeAlias
Beschreibung	Aliasname(n) der Knoten
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	1024
Werte	Mit mehreren Werten
OID (Objekt-ID)	1.3.18.0.2.4.420
GUID (globale eindeutige Kennung)	b3afd31d-5c5b-11d3-b818-002035559151

Tabelle 39. db2nodeName

Attribut	db2nodeName
Active Directory-LDAP-Anzeigename	ibm-db2NodeName
Allgemeiner Active Directory-Name (cn)	ibm-db2NodeName
Beschreibung	Knotenname
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	64
Werte	Mit einem Wert
OID (Objekt-ID)	1.3.18.0.2.4.419
GUID (globale eindeutige Kennung)	b3afd31e-5c5b-11d3-b818-002035559151

Tabelle 40. db2nodePtr

Attribut	db2nodePtr
Active Directory-LDAP-Anzeigename	ibm-db2NodePtr
Allgemeiner Active Directory-Name (cn)	ibm-db2NodePtr
Beschreibung	Zeiger auf das Knotenobjekt (DB2Node) für den Datenbankserver, der Eigner der Datenbank ist
Syntax	Registrierter Name
Maximale Länge	1000
Werte	Mit einem Wert
OID (Objekt-ID)	1.3.18.0.2.4.423
GUID (globale eindeutige Kennung)	b3afd31f-5c5b-11d3-b818-002035559151
Besondere Hinweise	Diese Abhängigkeit erlaubt es dem Client, die erforderlichen Informationen zum Übertragungsprotokoll abzurufen, um eine Verbindung zur Datenbank herzustellen.

Tabelle 41. db2gwPtr

Attribut	db2gwPtr
Active Directory-LDAP-Anzeigename	ibm-db2GwPtr
Allgemeiner Active Directory-Name (cn)	ibm-db2GwPtr
Beschreibung	Zeiger auf das Knotenobjekt für den Gateway-Server, über den auf die Datenbank zugegriffen werden kann
Syntax	Registrierter Name
Maximale Länge	1000

Tabelle 41. db2gwPtr (Forts.)

Attribut	db2gwPtr
Werte	Mit einem Wert
OID (Objekt-ID)	1.3.18.0.2.4.424
GUID (globale eindeutige Kennung)	b3afd31b-5c5b-11d3-b818-002035559151

Tabelle 42. db2instanceName

Attribut	db2instanceName
Active Directory-LDAP-Anzeigename	ibm-db2InstanceName
Allgemeiner Active Directory-Name (cn)	ibm-db2InstanceName
Beschreibung	Name des Datenbankserverexemplars
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	256
Werte	Mit einem Wert
OID (Objekt-ID)	1.3.18.0.2.4.428
GUID (globale eindeutige Kennung)	b3afd31c-5c5b-11d3-b818-002035559151

Tabelle 43. db2Type

Attribut	db2Type
Active Directory-LDAP-Anzeigename	ibm-db2Type
Allgemeiner Active Directory-Name (cn)	ibm-db2Type
Beschreibung	Datenbankserver-Art
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	64
Werte	Mit einem Wert
OID (Objekt-ID)	1.3.18.0.2.4.418
GUID (globale eindeutige Kennung)	b3afd320-5c5b-11d3-b818-002035559151
Hinweise	Zulässige Datenbankserver-Arten: SERVER, MPP und DCS

Tabelle 44. DCEPrincipalName

Attribut	DCEPrincipalName
Active Directory-LDAP-Anzeigename	ibm-DCEPrincipalName
Allgemeiner Active Directory-Name (cn)	ibm-DCEPrincipalName

Table 44. *DCEPrincipalName* (Forts.)

Attribut	DCEPrincipalName
Beschreibung	DCE-Principal-Name
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	2048
Werte	Mit einem Wert
OID (Objekt-ID)	1.3.18.0.2.4.443
GUID (globale eindeutige Kennung)	b3afd32d-5c5b-11d3-b818-002035559151

Table 45. *cesProperty*

Attribut	cesProperty
Active Directory-LDAP-Anzeigename	ibm-cesProperty
Allgemeiner Active Directory-Name (cn)	ibm-cesProperty
Beschreibung	Werte dieses Attributs können zur Bereitstellung anwendungsspezifischer Vorgabekonfigurationsparameter genutzt werden. Ein Wert kann z. B. Daten im XML-Format enthalten. Alle Werte dieses Attributs müssen einen einheitlichen Attributwert für cesPropertyType aufweisen.
Syntax	Zeichenfolge mit genauer Beachtung der Groß-/Kleinschreibung
Maximale Länge	32700
Werte	Mit mehreren Werten
OID (Objekt-ID)	1.3.18.0.2.4.307
GUID (globale eindeutige Kennung)	b3afd2d5-5c5b-11d3-b818-002035559151

Table 46. *cesPropertyType*

Attribut	cesPropertyType
Active Directory-LDAP-Anzeigename	ibm-cesPropertyType
Allgemeiner Active Directory-Name (cn)	ibm-cesPropertyType
Beschreibung	Werte dieses Attributs können zum Beschreiben der Syntax, Semantik oder anderer Kenndaten aller Werte des Attributs cesProperty verwendet werden. Der Wert „XML“ kann z. B. verwendet werden, um anzugeben, dass alle Werte des Attributs cesProperty in der XML-Syntax codiert sind.
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge

Tabelle 46. *cesPropertyType* (Forts.)

Attribut	<i>cesPropertyType</i>
Maximale Länge	128
Werte	Mit mehreren Werten
OID (Objekt-ID)	1.3.18.0.2.4.308
GUID (globale eindeutige Kennung)	b3afd2d6-5c5b-11d3-b818-002035559151

Tabelle 47. *cisProperty*

Attribut	<i>cisProperty</i>
Active Directory-LDAP-Anzeigename	ibm-cisProperty
Allgemeiner Active Directory-Name (cn)	ibm-cisProperty
Beschreibung	Werte dieses Attributs können zur Bereitstellung anwendungsspezifischer Vorgabekonfigurationsparameter genutzt werden. Ein Wert kann z. B. eine INI-Datei enthalten. Alle Werte dieses Attributs müssen einen einheitlichen Attributwert für <i>cisPropertyType</i> aufweisen.
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	32700
Werte	Mit mehreren Werten
OID (Objekt-ID)	1.3.18.0.2.4.309
GUID (globale eindeutige Kennung)	b3afd2e0-5c5b-11d3-b818-002035559151

Tabelle 48. *cisPropertyType*

Attribut	<i>cisPropertyType</i>
Active Directory-LDAP-Anzeigename	ibm-cisPropertyType
Allgemeiner Active Directory-Name (cn)	ibm-cisPropertyType
Beschreibung	Werte dieses Attributs können zum Beschreiben der Syntax, Semantik oder anderer Kenndaten aller Werte des Attributs <i>cisProperty</i> verwendet werden. Der Wert „INI File“ kann z. B. verwendet werden, um anzugeben, dass alle Werte des Attributs <i>cisProperty</i> INI-Dateien sind.
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	128
Werte	Mit mehreren Werten

Table 48. *cisPropertyType* (Forts.)

Attribut	cisPropertyType
OID (Objekt-ID)	1.3.18.0.2.4.310
GUID (globale eindeutige Kennung)	b3afd2e1-5c5b-11d3-b818-002035559151

Table 49. *binProperty*

Attribut	binProperty
Active Directory-LDAP-Anzeigename	ibm-binProperty
Allgemeiner Active Directory-Name (cn)	ibm-binProperty
Beschreibung	Werte dieses Attributs können zur Bereitstellung anwendungsspezifischer Vorgabekonfigurationsparameter genutzt werden. Ein Wert kann z. B. eine Gruppe binär codierter Lotus 1-2-3-Merkmale enthalten. Alle Werte dieses Attributs müssen einen einheitlichen Attributwert für binPropertyType aufweisen.
Syntax	binär
Maximale Länge	250000
Werte	Mit mehreren Werten
OID (Objekt-ID)	1.3.18.0.2.4.305
GUID (globale eindeutige Kennung)	b3afd2ba-5c5b-11d3-b818-002035559151

Table 50. *binPropertyType*

Attribut	binPropertyType
Active Directory-LDAP-Anzeigename	ibm-binPropertyType
Allgemeiner Active Directory-Name (cn)	ibm-binPropertyType
Beschreibung	Werte dieses Attributs können zum Beschreiben der Syntax, Semantik oder anderer Kenndaten aller Werte des Attributs binProperty verwendet werden. Der Wert „Lotus 123“ kann z. B. verwendet werden, um anzugeben, dass alle Werte des Attributs binProperty binär codierte Lotus 1-2-3-Merkmale sind.
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	128
Werte	Mit mehreren Werten
OID (Objekt-ID)	1.3.18.0.2.4.306
GUID (globale eindeutige Kennung)	b3afd2bb-5c5b-11d3-b818-002035559151

Tabelle 51. PropertyType

Attribut	PropertyType
Active Directory-LDAP-Anzeigename	ibm-propertyType
Allgemeiner Active Directory-Name (cn)	ibm-propertyType
Beschreibung	Werte dieses Attributs beschreiben die semantischen Kenndaten des Objekts eProperty.
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	128
Werte	Mit mehreren Werten
OID (Objekt-ID)	1.3.18.0.2.4.320
GUID (globale eindeutige Kennung)	b3afd4ed-5c5b-11d3-b818-002035559151

Tabelle 52. settingID

Attribut	settingID
Active Directory-LDAP-Anzeigename	Nicht verfügbar
Allgemeiner Active Directory-Name (cn)	Nicht verfügbar
Beschreibung	Ein Benennungsattribut, das zum Identifizieren der von cimSetting abgeleiteten Objekteinträge, z. B. eProperty, dient.
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	256
Werte	Mit einem Wert
OID (Objekt-ID)	1.3.18.0.2.4.325
GUID (globale eindeutige Kennung)	b3afd596-5c5b-11d3-b818-002035559151

Anhang K. Erweitern der Steuerzentrale

In Version 7 können Sie die Steuerzentrale von DB2 Universal Database mit der neuen *Plug-In*-Architektur erweitern, um zusätzliche Funktionen zu erhalten.

Das Konzept der *Plug-In*-Architektur besteht darin, die Verfügbarkeit neuer Menüpunkte für ein bestimmtes Objekt zum Kontextmenü der Steuerzentrale und neue Druckknöpfe zur Menüleiste hinzuzufügen. Eine Gruppe von Java-Schnittstellen, die Sie implementieren müssen, werden zusammen mit den Tools geliefert. Mit diesen Schnittstellen wird der Steuerzentrale mitgeteilt, welche zusätzlichen Funktionen aufzunehmen sind.

Überlegungen zur Leistung

Die Plug-In-Erweiterungen (db2plug.zip) werden beim Systemstart der Steuerzentralen-Tools geladen. Daher kann je nach Größe der ZIP-Datei der Systemstart länger dauern. Für die meisten Benutzer dürfte die Plug-In-ZIP-Datei jedoch klein sein und die Auswirkung daher minimal sein.

Überlegungen zu Paketen

Sie müssen die Erweiterungsklassendateien gemäß den Regeln einer Java-Archivierungsdatei komprimieren. Damit die Tools der Steuerzentrale als Anwendung ausgeführt werden, muss sich die ZIP-Datei (db2plug.zip) in dem Verzeichnis befinden, auf das *classpath* verweist. Wenn die Tools der Steuerzentrale als Applet ausgeführt werden sollen, muss sich die ZIP-Datei an der Position befinden, auf die die Kennung `<code_basis>` in der HTML-Datei der Steuerzentrale zeigt.

Die ZIP-Datei sollte ohne Komprimierung erstellt werden und die relativen Pfadpositionen aller Klassendateien beibehalten (`zip -r0 db2plug.zip *.class`).

Schnittstellenbeschreibungen

Die folgenden Schnittstellen werden bereitgestellt:

- CCExtension
- CCOject
- CCMenuAction
- CCToolbarAction

Die Schnittstellen werden mit Beispielen in den folgenden Abschnitten beschrieben.

CCExtension

Mit der CCExtension-Schnittstelle können Sie die Benutzerschnittstelle der Steuerzentrale erweitern, um neue Druckknöpfe und Menüpunkte hinzuzufügen und vorhandene Menüaktionen außer Kraft zu setzen.

Die externe Schnittstelle ist wie folgt definiert:

```
public interface CCExtension
{
    /**
     * Get an array of CCOBJECT subclass objects which define
     * a list of objects to be inserted or overridden in the
     * Control Center
     * @return CCOBJECT[] CCOBJECT subclass objects array
     */
    public CCOBJECT[] getObjects();

    /**
     * Get an array of CCToolbarAction subclass objects which represent
     * a list of buttons to be added to the Control Center
     * main toolbar.
     * @return CCToolbarAction[] CCToolbarAction subclass objects array
     */
    public CCToolbarAction[] getToolbarActions();
}
```

Zur Verwendung von CCExtension erstellen Sie eine Java-Klasse, die das Paket "com.ibm.db2.tools.cc.navigator" importiert und diese Schnittstelle implementiert. Die neue Klasse muss die Implementierung der Methoden getObjects() und getToolbarActions() bieten.

Die Methode getObjects() liefert ein CCOBJECT-Feld, das die vorhandenen Objekte definiert, zu denen der Benutzer neue Menüpunkte hinzufügen oder aus denen er eine Reihe von vordefinierten Menüpunkten entfernen möchte.

Die Methode getToolbarActions() liefert ein CCToolbarAction-Feld, die zur Hauptmenüleiste der Steuerzentrale hinzugefügt wird.

Eine oder mehrere CCExtension-Unterklassendateien können verwendet werden, um die Erweiterungen der Steuerzentrale zu definieren. Damit die Steuerzentrale diese Erweiterungen verwenden kann, verwenden Sie die folgende Prozedur:

1. Erstellen Sie eine Datei "db2plug.zip", die alle CCExtension-Unterklassendateien enthält. Diese Dateien sollten nicht komprimiert sein. Wenn z. B. die CCExtension-Dateien sich im Plug-In-Paket befinden und im Plug-In-Verzeichnis enthalten sind, verwenden Sie folgenden Befehl:

```
zip -r0 db2plug.zip plugin\*.class
```

Dieser Befehl setzt alle Klassendateien des Plug-In-Pakets in die Datei "db2plug.zip" und behält die relativen Pfade bei.

2. Damit die Steuerzentrale als Applet ausgeführt wird, setzen Sie die Datei "db2plug.zip" an die Position, auf die die Kennung <code_basis> in der HTML-Datei der Steuerzentrale zeigt. Wenn die Steuerzentrale als Anwendung ausgeführt werden soll, setzen Sie die Datei "db2plug.zip" in ein Verzeichnis, auf das die Umgebungsvariable CLASSPATH zeigt.

Für Browser, die mehrere Archive unterstützen, fügen Sie einfach "db2plug.zip" zur Archivliste der HTML-Seite der Steuerzentrale hinzu. Andernfalls müssen sich alle CCExtension-, CCOBJECT-, CCToolbarAction- und CCMenuAction-Unterklassendateien in Abhängigkeit vom Paket, zu dem sie gehören, in ihren relativen Verzeichnissen befinden.

CCObject

Mit der CCOBJECT-Schnittstelle können Sie das Verhalten der Menüaktionen eines vorhandenen Objekts ändern.

Die externe Schnittstelle ist wie folgt definiert:

```
public interface CCOBJECT
{
    /**
     * The following static constants defines a list of object type
     * available to be added to the Control Center tree.
     */
    public static final int UDB_SYSTEMS_FOLDER = 0;
    public static final int UDB_SYSTEM = 1;
    public static final int UDB_INSTANCES_FOLDER = 2;
    public static final int UDB_INSTANCE = 3;
    public static final int UDB_DATABASES_FOLDER = 4;
    public static final int UDB_DATABASE = 5;
    public static final int UDB_TABLES_FOLDER = 6;
    public static final int UDB_TABLE = 7;
    public static final int UDB_TABLESPACES_FOLDER = 8;
    public static final int UDB_TABLESPACE = 9;
    public static final int UDB_VIEWS_FOLDER = 10;
    public static final int UDB_VIEW = 11;
    public static final int UDB_ALIASES_FOLDER = 12;
    public static final int UDB_ALIAS = 13;
    public static final int UDB_TRIGGERS_FOLDER = 14;
    public static final int UDB_TRIGGER = 15;
    public static final int UDB_SCHEMAS_FOLDER = 16;
    public static final int UDB_SCHEMA = 17;
    public static final int UDB_INDEXES_FOLDER = 18;
    public static final int UDB_INDEX = 19;
    public static final int UDB_CONNECTIONS_FOLDER = 20;
    public static final int UDB_CONNECTION = 21;
    public static final int UDB_REPLICATION_SOURCES_FOLDER = 22;
    public static final int UDB_REPLICATION_SOURCE = 23;
}
```

```

public static final int UDB_REPLICATION_SUBSCRIPTIONS_FOLDER      = 24;
public static final int UDB_REPLICATION_SUBSCRIPTION            = 25;
public static final int UDB_BUFFERPOOLS_FOLDER                  = 26;
public static final int UDB_BUFFERPOOL                          = 27;
public static final int UDB_APPLICATION_OBJECTS_FOLDER          = 28;
public static final int UDB_USER_DEFINED_DISTINCT_DATATYPES_FOLDER = 29;
public static final int UDB_USER_DEFINED_DISTINCT_DATATYPE      = 30;
public static final int UDB_USER_DEFINED_DISTINCT_FUNCTIONS_FOLDER = 31;
public static final int UDB_USER_DEFINED_DISTINCT_FUNCTION      = 32;
public static final int UDB_PACKAGES_FOLDER                     = 33;
public static final int UDB_PACKAGE                             = 34;
public static final int UDB_STORE_PROCEDURES_FOLDER             = 35;
public static final int UDB_STORE_PROCEDURE                     = 36;
public static final int UDB_USER_AND_GROUP_OBJECTS_FOLDER       = 37;
public static final int UDB_DB_USERS_FOLDER                     = 38;
public static final int UDB_DB_USER                             = 39;
public static final int UDB_DB_GROUPS_FOLDER                    = 40;
public static final int UDB_DB_GROUP                            = 41;
public static final int UDB_DRDA_TABLE                          = 42;

public static final int S390_SUBSYSTEMS_FOLDER                  = 43;
public static final int S390_SUBSYSTEM                          = 44;
public static final int S390_BUFFERPOOLS_FOLDER                 = 45;
public static final int S390_BUFFERPOOL                         = 46;
public static final int S390_VIEWS_FOLDER                       = 47;
public static final int S390_VIEW                               = 48;
public static final int S390_DATABASES_FOLDER                   = 49;
public static final int S390_DATABASE                           = 50;
public static final int S390_TABLESPACES_FOLDER                 = 51;
public static final int S390_TABLESPACE                         = 52;
public static final int S390_TABLES_FOLDER                      = 53;
public static final int S390_TABLE                             = 54;
public static final int S390_INDEXES_FOLDER                     = 55;
public static final int S390_INDEX                              = 56;
public static final int S390_STORAGE_GROUPS_FOLDER              = 57;
public static final int S390_STORAGE_GROUP                     = 58;
public static final int S390_ALIASES_FOLDER                     = 59;
public static final int S390_ALIAS                              = 60;
public static final int S390_SYNONYMS_FOLDER                    = 61;
public static final int S390_SYNONYM                           = 62;
public static final int S390_APPLICATION_OBJECTS_FOLDER         = 63;
public static final int S390_COLLECTIONS_FOLDER                 = 64;
public static final int S390_COLLECTION                         = 65;
public static final int S390_PACKAGES_FOLDER                    = 66;
public static final int S390_PACKAGE                            = 67;
public static final int S390_PLANS_FOLDER                       = 68;
public static final int S390_PLAN                               = 69;
public static final int S390_PROCEDURES_FOLDER                  = 70;
public static final int S390_PROCEDURE                          = 71;
public static final int S390_DB_USERS_FOLDER                    = 72;
public static final int S390_DB_USER                            = 73;
public static final int S390_LOCATIONS_FOLDER                   = 74;
public static final int S390_LOCATION                           = 75;
public static final int S390_DISTINCT_TYPES_FOLDER              = 76;
public static final int S390_DISTINCT_TYPE                      = 77;

```



```

public static final int S390_USER_DEFINED_FUNCTIONS_FOLDER      = 78;
public static final int S390_USER_DEFINED_FUNCTION             = 79;
public static final int S390_TRIGGERS_FOLDER                   = 80;
public static final int S390_TRIGGER                           = 81;
public static final int S390_SCHEMAS_FOLDER                     = 82;
public static final int S390_SCHEMA                             = 83;
public static final int S390_CATALOG_TABLES_FOLDER              = 84;
public static final int S390_CATALOG_TABLE                      = 85;

public static final int DCS_GATEWAY_CONNECTIONS_FOLDER         = 86;
public static final int DCS_GATEWAY_CONNECTION                 = 87;

/**
 * Total number of object types
 */
public static final int NUM_OBJECT_TYPES                        = 88;

/**
 * Get the name of these object
 * The function returns the name of this object. This name
 * can be of three types:
 * (1) Fully qualified name
 *     Syntax: xxxxx-yyyyy-zzzzz
 *           where xxxxx-yyyyy is the fully quality name of the
 *           parent object and zzzzz is the name of the new object.
 *     Note: Parent and child object name is separated by '-' character.
 *     If a schema name is required to identify object, the fully
 *     qualified name is represented by xxxxx-yyyyy-wwwww.zzzzz
 *           where wwwww is the schema name.
 *     Only the behavior of the object that matches this fully
 *     qualified name will be affected.
 * (2) Parent fully qualified name
 *     Syntax: xxxxx-yyyyy
 *           where xxxxx-yyyyy is the fully qualified name of the
 *           parent object.
 *     When the object type is folder (ie. DATABASES_FOLDER), the
 *     getName() should only return the fully qualified name of the
 *     folder's parent.
 *     Only the behavior of the object that match this name
 *     and the specific type return by the getType() function will be
 *     affected.
 * (3) null
 *     Syntax: null
 *     If null is return, the CCActions returns by the getActions()
 *     call will be applied to all objects of type returns by the
 *     getType() call.
 * @return String object name
 */
public String getName();

/**
 * Get the type of this object
 * @return int return one of the static type constants defined
 * in this interface
 */

```

```

public int getType();

/**
 * Get the CCMenu Action array which defines the list of menu actions
 * to be created for the selected object
 * return CCMenuAction[] CCMenuAction array
 */
public CCMenuAction[] getMenuActions();

/**
 * Check if this object is editable.
 * If not, the Alter related menu items will be removed from
 * the object's popup menu return boolean If false, the Alter
 * menu item will be removed from the object's popup menu
 */
public boolean isEditable();

/**
 * Check if this object is configurable.
 * If not, the configuration related menu items will be
 * removed from the object's popup menu return boolean If
 * false, the Configuration related menu item will be removed
 * from the object's popup menu
 */
public boolean isConfigurable();
}

```

Anmerkung: Zur Zeit sollten die beiden letzten Methoden in CCOBJECT: `isEditable()` und `isConfigurable()` immer den Wert **wahr** (True) liefern.

CCMenuItem

Mit der CCMenuAction-Schnittstelle können Sie eine neue Aktion definieren, die von einem Objekt der Steuerzentrale verwendet werden soll.

Die externe Schnittstelle ist wie folgt definiert:

```

public interface CCMenuAction
{
    /**
     * Get the name of this action
     * @return String Name text on the menu item
     */
    public String getMenuText();

    /**
     * Invoked when an action occurs. Use the getActionCommand()
     * method of the ActionEvent to get the fully qualified name of
     * the invoked Control Center object.
     * @param e Action event
     */
    public void actionPerformed(ActionEvent e);
}

```

CCToolbarAction

Mit der CCToolbarAction-Schnittstelle können Sie eine neue Aktion in der Menüleiste der Steuerzentrale definieren.

Die externe Schnittstelle ist wie folgt definiert:

```
public interface CCToolbarAction
{
    /**
     * Get the name of this action
     * @return String Name text on the menu item, or toolbar
     * button hover help
     */
    public String getHoverHelpText();

    /**
     * Get the icon for the toolbar button
     * Any toolbar CCAction should implement this function and return
     * a valid ImageIcon object. Otherwise, the button will have no icon.
     * @return ImageIcon Icon to be displayed
     */
    public ImageIcon getIcon();

    /**
     * Invoked when an action occurs.
     * @param e Action event
     */
    public void actionPerformed(ActionEvent e);
}
```

Einsatz-Szenario

Der Code im folgenden Beispiel wird folgende Aktionen durchführen:

1. Aktualisieren der Aktionen der Beispieldatenbank (siehe „MySample.java“ auf Seite 512)
2. Aktualisieren der Aktionen aller Datenbankobjekte (siehe „MyDatabaseActions.java“ auf Seite 513)
3. Hinzufügen eines neuen Exemplarobjekts (siehe „MyInstance.java“ auf Seite 514)
4. Aktualisieren der Aktionen des DB-Exemplars (siehe „MyDB2.java“ auf Seite 514)
5. Aktualisieren der Aktionen des Datenbankordners (siehe „MyDatabases.java“ auf Seite 515)
6. Aktualisieren der Aktionen der Tabelle SYSIBM.SYSPLAN (siehe „MySYSPLAN.java“ auf Seite 516)
7. Hinzufügen eines neuen Tabellenobjekts (siehe „MyTable.java“ auf Seite 516)

8. Aktualisieren der Aktionen des Objekts DB_User unter dem Anwendungsobjekt (siehe „MyDBUser.java“ auf Seite 517)
9. Hinzufügen eines Knopfs zur Menüleiste der Steuerzentrale (siehe „MyToolBarAction.java“ auf Seite 518).

Die Haupterweiterungsdatei ist MyExtension.java. Alle Klassendateien werden mit dem folgenden Befehl im Plug-In-Verzeichnis gespeichert und komprimiert:

```
zip -r0 db2plug.zip plugin
```

Die entstandene Datei "db2plug.zip" wird dann in das Verzeichnis, auf das die Umgebungsvariable CLASSPATH verweist, oder in das codebase-Verzeichnis (Codebasis) gesetzt, je nachdem, ob die Steuerzentrale als Anwendung oder Applet ausgeführt wird.

MyExtension.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyExtension implements CCExtension
{
    public CCOBJECT[] getObjects()
    {
        CCOBJECT[] objs = new CCOBJECT[10];
        objs[0] = new MySample();
        objs[1] = new MyDatabaseActions();
        objs[2] = new MyInstance();
        objs[3] = new MyDB2();
        objs[4] = new MyDatabases();
        objs[5] = new MySYSPLAN();
        objs[6] = new MyTable();
        objs[7] = new MyDBUser();
        return objs;
    }

    public CCACTION[] getActions()
    {
        CCACTION[] actions = new CCACTION[1];
        actions[0] = new MyToolBarAction();
        return actions;
    }
}
```

MySample.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MySample implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - SAMPLE";
    }
}
```

```

    }

    public int getType()
    {
        return DATABASE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[2];
        acts[0] = new MyAlterAction();
        acts[1] = new MyAction();
        return acts;
    }
}

```

MyDatabaseActions.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDatabaseActions implements CCOBJECT
{
    public String getName()
    {
        return null;
    }

    public int getType()
    {
        return DATABASE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCAction[] getActions()
    {

```

```

        CCAction[] acts = new CCAction[2];
        acts[0] = new MyDropAction();
        acts[1] = new MyAction();
        return acts;
    }
}

```

MyInstance.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyInstance implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - MyInstance";
    }

    public int getType()
    {
        return INSTANCE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return true;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[2];
        acts[0] = new MyAlterAction();
        acts[1] = new MyAction();
        return null;
    }
}

```

MyDB2.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDB2 implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2";
    }
}

```

```

public int getType()
{
    return INSTANCE;
}

public javax.swing.ImageIcon getIcon()
{
    return null;
}

public boolean isNew()
{
    return false;
}

public CCAction[] getActions()
{
    CCAction[] acts = new CCAction[3];
    acts[0] = new MyAlterAction();
    acts[1] = new MyAction();
    acts[2] = new MyCascadeAction();
    return acts;
}
}

```

MyDatabases.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDatabases implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - Databases";
    }

    public int getType()
    {
        return DATABASE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[1];
        acts[0] = new MyCreateAction();
    }
}

```

```

        return acts;
    }
}

```

MySYSPLAN.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MySYSPLAN implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - SAMPLE - SYSIBM - SYSPLAN";
    }

    public int getType()
    {
        return TABLE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCACTION[] getActions()
    {
        CCACTION[] acts = new CCACTION[2];
        acts[0] = new MyALTERACTION();
        acts[1] = new MyACTION();
        return acts;
    }
}

```

MyTable.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyTable implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - SAMPLE - SYSIBM - MyTable";
    }

    public int getType()
    {
        return TABLE;
    }
}

```



```

    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return true;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[2];
        acts[0] = new MyAlterAction();
        acts[1] = new MyAction();
        return acts;
    }
}

```

MyDBUser.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDBUser implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - TEST-DB Users";
    }

    public int getType()
    {
        return DB_USER;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[2];
        acts[0] = new MyAlterAction();
        acts[1] = new MyAction();
    }
}

```

```

        return acts;
    }
}

```

MyToolBarAction.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;
import javax.swing.*;

public class MyToolBarAction extends CCAction
{
    public MyToolBarAction()
    {
        super("MyToolBarAction");
    }

    public ImageIcon getIcon()
    {
        return <Your icon>;
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My action performed, object name = " +
                            objectName );
        return true;
    }
}

```

MyAlterAction.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyAlterAction extends CCAction
{
    public MyAlterAction()
    {
        super(0);
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My alter action performed, object name = " +
                            objectName );
        return true;
    }
}

```

MyAction.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyAction extends CCAction

```

```

{
    public MyAction()
    {
        super("MyAction");
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My action performed, object name = " +
            objectName );
        return true;
    }
}

```

MyDropAction.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDropAction extends CCAction
{
    public MyDropAction()
    {
        super(1);
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My drop action performed, object name = " +
            objectName );
        return true;
    }
}

```

MyCascadeAction.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyCascadeAction extends CCAction
{
    public MyCascadeAction()
    {
        super(11,2);
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My cascade action performed, object name = " +
            objectName );
        return true;
    }
}

```

MyCreateAction.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyCreateAction extends CCAction
{
    public MyCreateAction()
    {
        super(0);
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My create action performed, object name = " +
            objectName );
        return true;
    }
}
```

Anhang L. Verwenden der DB2-Bibliothek

Die Bibliothek für DB2 Universal Database besteht aus Online-Hilfe, Handbüchern (PDF und HTML) und Beispielprogrammen in HTML-Format. Im Folgenden wird beschrieben, welche Informationen bereitgestellt werden und wie Sie darauf zugreifen können.

Über **Information - Unterstützung** können Sie online auf die Produktinformationen zugreifen. Weitere Informationen finden Sie in „Zugreifen auf Informationen mit "Information - Unterstützung"“ auf Seite 540. Sie können sich im Web Informationen zu Tasks und zur Fehlerbehebung sowie DB2-Bücher, Beispielprogramme und DB2-Informationen anzeigen lassen.

PDF-Dateien und gedruckte Bücher für DB2

Informationen zu DB2

In der folgenden Tabelle sind die DB2-Handbücher in vier Kategorien unterteilt:

DB2-Benutzerhandbücher und -Referenzinformationen

Diese Bücher enthalten die allgemeinen DB2-Informationen für alle Plattformen.

DB2-Installations- und -Konfigurationsinformationen

Diese Bücher gelten für DB2 auf einer bestimmten Plattform. So steht beispielsweise jeweils ein separates Handbuch *Einstieg* (Quick Beginnings) für DB2 auf OS/2-, Windows- und UNIX-Plattformen zur Verfügung.

Plattformübergreifende Beispielprogramme in HTML

Bei diesen Beispielen handelt es sich um die HTML-Versionen der mit Application Development Client installierten Beispielprogramme. Sie dienen zur Information und können die Programme selbst nicht ersetzen.

Release-Informationen

Diese Dateien enthalten die neuesten Informationen, die in die DB2-Handbücher nicht mehr aufgenommen werden konnten.

Die Installationshandbücher, Release-Informationen und Lernprogramme können im HTML-Format direkt von der Produkt-CD-ROM angezeigt werden. Die meisten Handbücher stehen auf der Produkt-CD-ROM im HTML-Format zur Verfügung und können angezeigt werden. Auf der CD-ROM mit DB2-Veröffentlichungen stehen die Handbücher im PDF-Format zur Verfügung und

können mit Adobe Acrobat angezeigt und gedruckt werden. Darüber hinaus können Sie gedruckte Veröffentlichungen bei IBM bestellen. Siehe hierzu „Bestellen der gedruckten Handbücher“ auf Seite 535. Die folgende Tabelle enthält eine Liste der Bücher, die bestellt werden können.

Auf OS/2- und Windows-Plattformen können Sie die HTML-Dateien im Verzeichnis `sql11ib\doc\html` installieren. Die DB2-Informationen werden in verschiedene Sprachen übersetzt, jedoch nicht alle Informationen in alle Sprachen. Sind bestimmte Informationen in einer Sprache nicht verfügbar, wird stattdessen die englische Version dieser Informationen zur Verfügung gestellt.

Auf UNIX-Plattformen können Sie die HTML-Dateien in mehreren Sprachen installieren, und zwar in den Unterverzeichnissen `doc/%L/html`, wobei `%L` für den Code der jeweiligen Landessprache steht. Weitere Informationen finden Sie im entsprechenden Handbuch *Einstieg*.

Es gibt verschiedene Möglichkeiten, auf DB2-Bücher und -Informationen zuzugreifen:

- „Anzeigen von Online-Informationen“ auf Seite 539
- „Suchen nach Online-Informationen“ auf Seite 544
- „Bestellen der gedruckten Handbücher“ auf Seite 535
- „Drucken der PDF-Handbücher“ auf Seite 534

Tabelle 53. Informationen zu DB2

Name	Beschreibung	IBM Form PDF-Datei- name	HTML- Verzeichnis
DB2-Benutzerhandbücher und -Referenzinformationen			
<i>Systemverwaltung</i>	<p><i>Systemverwaltung: Konzept.</i> Dieses Handbuch enthält eine Übersicht über Datenbankkonzepte, Informationen zu Aspekten des Datenbankentwurfs (wie z. B. zum logischen und physischen Datenbankentwurf) sowie eine Erläuterung zur hohen Verfügbarkeit.</p> <p><i>Systemverwaltung: Implementierung.</i> Dieses Handbuch enthält Informationen zu Implementierungsaspekten, wie beispielsweise zur Implementierung des Datenbankentwurfs, zum Zugriff auf Datenbanken sowie zu Prüfungs-, Sicherungs- und Wiederherstellungsverfahren.</p> <p><i>Systemverwaltung: Optimierung.</i> Dieses Handbuch enthält Informationen zur Datenbankumgebung sowie zur Auswertung und Optimierung der Anwendungsleistung.</p>	<p>SC12-2879 db2d1g70</p> <p>SC12-2878 db2d2g70</p> <p>SC12-2877 db2d3g70</p>	db2d0
<i>Administrative API Reference</i>	<p>Dieses Handbuch enthält eine Beschreibung zu den DB2-Anwendungsprogrammierschnittstellen (APIs) und -Datenstrukturen, die Sie zum Verwalten Ihrer Datenbank verwenden können. Darüber hinaus wird in diesem Handbuch erläutert, wie Sie APIs von Ihren Anwendungen aus aufrufen können.</p>	<p>SC09-2947 db2b0e70</p>	db2b0
<i>Application Building Guide</i>	<p>Dieses Handbuch umfasst Informationen zur Umgebungskonfiguration sowie Anweisungsschritte zum Kompilieren, Verbinden und Ausführen von DB2-Anwendungen auf Windows-, OS/2- und UNIX-Plattformen.</p>	<p>SC09-2948 db2axe70</p>	db2ax

Tabelle 53. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Datei- name	HTML- Verzeichnis
<i>APPC, CPI-C, and SNA Sense Codes</i>	Dieses Handbuch enthält Basisinformationen zu APPC-, CPI-DFV- und SNA-Prüfcodes, die bei der Arbeit mit DB2 Universal Database-Produkten ausgegeben werden können. Nur im HTML-Format verfügbar.	Keine Formnummer db2ape70	db2ap
<i>Application Development Guide</i>	Dieses Handbuch enthält eine Erläuterung zur Entwicklung von Anwendungen, die mit Hilfe von eingebettetem SQL bzw. JAVA (JDBC und SQLJ) auf DB2-Datenbanken zugreifen. Unter anderem wird das Schreiben von gespeicherten Prozeduren, das Schreiben von benutzerdefinierten Funktionen, das Erstellen von benutzerdefinierten Typen, das Verwenden von Auslösern und das Entwickeln von Anwendungen in partitionierten Umgebungen oder mit Systemen zusammengesetzter Datenbanken beschrieben.	SC09-2949 db2a0e70	db2a0
<i>CLI Guide and Reference</i>	Dieses Handbuch erklärt die Entwicklung von Anwendungen, die für den Zugriff auf DB2-Datenbanken DB2 Call Level Interface verwenden, eine aufrufbare SQL-Schnittstelle, die mit der Microsoft-ODBC-Spezifikation kompatibel ist.	SC09-2950 db2l0e70	db2l0
<i>Command Reference</i>	Dieses Handbuch enthält eine Erläuterung zur Verwendung des Befehlszeilenprozessors und eine Beschreibung der DB2-Befehle für die Datenbankverwaltung.	SC09-2951 db2n0e70	db2n0

Tabelle 53. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Datei- name	HTML- Verzeichnis
<i>Konnektivität Ergänzung</i>	Dieses Handbuch enthält Konfigurations- und Referenzinformationen zur Verwendung von DB2 für AS/400, DB2 für OS/390, DB2 für MVS oder DB2 für VM als DRDA-Anwendungs-Requester mit DB2 Universal Database-Servern. Darüber hinaus enthält dieses Handbuch Informationen zur Verwendung von DRDA-Anwendungs-Servern mit DB2 Connect-Anwendungs-Requestern.	Keine Form- nummer db2h1g70	db2h1
<i>Versetzen von Daten Dienstprogramme und Referenz</i>	Dieses Handbuch enthält eine Erläuterung zur Verwendung der DB2-Dienstprogramme, wie beispielsweise IMPORT, EXPORT, LOAD, AUTOLOADER und DPROP, die das Verschieben von Daten vereinfachen.	SC12-2881 db2dmg70	db2dm
<i>Data Warehouse-Zentrale Verwaltung</i>	Dieses Handbuch enthält Informationen zur Erstellung und Verwaltung eines Data Warehouse mit Hilfe der Data Warehouse-Zentrale.	SC12-2885 db2ddg70	db2dd
<i>Data Warehouse Center Application Integration Guide</i>	Dieses Handbuch enthält Informationen, die Programmierer bei der Integration von Anwendungen in die Data Warehouse-Zentrale sowie in den Information Catalog Manager unterstützen.	SC26-9994 db2ade70	db2ad
<i>DB2 Connect Benutzer- handbuch</i>	Dieses Handbuch enthält eine Beschreibung der Konzepte der DB2 Connect-Produkte, allgemeine Informationen zur Verwendung sowie Informationen zur Programmierung dieser Produkte.	SC12-2880 db2c0g70	db2c0
<i>DB2 Query Patroller Administration Guide</i>	Dieses Handbuch enthält eine Übersicht über den Betrieb des DB2 Query Patroller-Systems, spezifische Informationen zum Systembetrieb und zur Verwaltung sowie Task-Informationen zu den GUI-Verwaltungsdienstprogrammen.	SC09-2958 db2dwe70	db2dw
<i>DB2 Query Patroller User's Guide</i>	In diesem Handbuch wird die Verwendung der Tools und Funktionen von DB2 Query Patroller beschrieben.	SC09-2960 db2wwe70	db2ww

Tabelle 53. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Datei- name	HTML- Verzeichnis
<i>Glossar</i>	Dieses Handbuch enthält Definitionen zu den in DB2 und den zugehörigen Komponenten verwendeten Begriffen. Es ist im Handbuch <i>SQL Reference</i> enthalten und steht außerdem separat im HTML-Format zur Verfügung.	Keine Formnummer db2t0g70	db2t0
<i>DB2 UDB Image, Audio und Video Extender Verwaltung und Programmierung</i>	Dieses Handbuch enthält Basisinformationen zu DB2 Extender, Informationen zur Verwaltung und Konfiguration von IAV Extender sowie Informationen zur Programmierung mit Hilfe von IAV Extender. Es enthält Referenzinformationen, Diagnoseinformationen (mit Nachrichten) und Beispiele.	SC12-2892 dmbu7g70	dmbu7
<i>Information Catalog Manager Systemverwaltung</i>	Dieses Handbuch enthält eine Anleitung zur Verwaltung von Informationskatalogen.	SC12-2886 db2dig70	db2di
<i>Information Catalog Manager Programming Guide and Reference</i>	Dieses Handbuch enthält Definitionen für die Architekturschnittstellen für Information Catalog Manager.	SC26-9997 db2bie70	db2bi
<i>Information Catalog Manager Benutzerhandbuch</i>	Dieses Handbuch enthält Informationen zur Verwendung der Information Catalog Manager-Benutzerschnittstelle.	SC12-2887 db2aig70	db2ai
<i>Installation und Konfiguration Ergänzung</i>	Dieses Handbuch enthält Anweisungen zur Planung, Installation und Konfiguration von plattformspezifischen DB2-Clients. Darüber hinaus enthält es Informationen zu Bindevorgängen, zum Einrichten der Client/Server-Kommunikation, zu DB2-GUI-Tools, zu DRDR-AS, zur verteilten Installation, zur Konfiguration von verteilten Anforderungen sowie zum Zugriff auf heterogene Datenquellen.	GC12-2864 db2iyg70	db2iy

Tabelle 53. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Datei- name	HTML- Verzeichnis
<i>Fehlernachrichten</i>	<p>Dieses Handbuch enthält eine Liste der Nachrichten und Codes, die von DB2, vom Information Catalog Manager und von der Data Warehouse-Zentrale ausgegeben werden, sowie eine Beschreibung der jeweils erforderlichen Benutzeraktionen.</p> <p>Sie können beide Bände des Handbuchs <i>Fehlernachrichten</i> in englischer Sprache in den USA und Kanada unter der Formnummer SBOF-8932 bestellen.</p>	<p>Band 1 GC12-2875</p> <p>db2m1g70 Band 2 GC12-2888</p> <p>db2m2g70</p>	db2m0
<i>OLAP Integration Server Administration Guide</i>	<p>Dieses Handbuch enthält eine Erläuterung zur Verwendung der Komponente Administration Manager von OLAP Integration Server.</p>	<p>SC27-0782</p> <p>db2dpe70</p>	n/v
<i>OLAP Integration Server Metaoutline User's Guide</i>	<p>Dieses Handbuch enthält eine Erläuterung zum Erstellen und Ausfüllen von OLAP-Metastrukturen mit Hilfe der OLAP Metaoutline-Standardschnittstelle (nicht mit Hilfe des OLAP Metaoutline Assistent).</p>	<p>SC27-0784</p> <p>db2upe70</p>	n/v
<i>OLAP Integration Server Model User's Guide</i>	<p>Dieses Handbuch enthält eine Erläuterung zum Erstellen von OLAP-Modellen mit Hilfe der OLAP Model-Standardschnittstelle (nicht mit Hilfe des OLAP Model Assistent).</p>	<p>SC27-0783</p> <p>db2lpe70</p>	n/v
<i>OLAP Konfiguration und Benutzerhandbuch</i>	<p>Dieses Handbuch enthält Informationen zur Konfiguration und Einrichtung von OLAP Starter Kit.</p>	<p>SC12-2889</p> <p>db2ipg70</p>	db2ip
<i>OLAP Tabellenkalkulations-Add-In Benutzerhandbuch für Excel</i>	<p>Dieses Handbuch enthält eine Beschreibung zur Verwendung des Tabellenkalkulationsprogramms Excel zum Analysieren von OLAP-Daten.</p>	<p>SC12-2890</p> <p>db2epg70</p>	db2ep
<i>OLAP Tabellenkalkulations-Add-In Benutzerhandbuch für Lotus 1-2-3</i>	<p>Dieses Handbuch enthält eine Beschreibung zur Verwendung des Tabellenkalkulationsprogramms Lotus 1-2-3 zum Analysieren von OLAP-Daten.</p>	<p>SC12-2891</p> <p>db2tpg70</p>	db2tp
<i>Replikation Benutzer- und Referenzhandbuch</i>	<p>Dieses Handbuch enthält Informationen zur Planung, Konfiguration, Verwaltung und Verwendung der mit DB2 gelieferten Replikations-Tools.</p>	<p>SC12-2884</p> <p>db2e0g70</p>	db2e0

Tabelle 53. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Datei- name	HTML- Verzeichnis
<i>Spatial Extender Benutzer- und Referenzhandbuch</i>	Dieses Handbuch enthält Informationen zur Installation, Konfiguration, Verwaltung, Programmierung und Fehlerbehebung für den Spatial Extender. Darüber hinaus enthält es zentrale Beschreibungen räumlicher Datenkonzepte sowie spezifische Referenzinformationen (Nachrichten und SQL) für den Spatial Extender.	SC12-2894 db2sbg70	db2sb
<i>SQL Erste Schritte</i>	Dieses Handbuch enthält eine Einführung in die SQL-Konzepte sowie Beispiele für eine Reihe von Konstrukten und Tasks.	SC12-2882 db2y0g70	db2y0
<i>SQL Reference, Band 1 und Band 2</i>	Dieses Handbuch beschreibt die Syntax, die Semantik und die Regeln von SQL. Darüber hinaus enthält das Handbuch Informationen zu Inkompatibilitäten zwischen Release-Ständen, Produkt einschränkungen und Katalogsichten. Sie können beide Bände des Handbuchs <i>SQL Reference</i> in englischer Sprache in den USA und Kanada unter der Formnummer SBOF-8933 bestellen.	Band 1 ^ SC09-2974 db2s1e70 Band 2 SC09-2975 db2s2e70	db2s0
<i>System Monitor Guide and Reference</i>	Dieses Handbuch enthält eine Beschreibung zum Sammeln unterschiedlicher Informationen zu Datenbanken und dem Datenbankmanager. In diesem Buch wird erläutert, wie Sie mit Hilfe dieser Informationen einen Einblick in Datenbankaktivitäten erhalten, die Leistung verbessern und Fehlerursachen feststellen können.	SC09-2956 db2f0e70	db2f0
<i>Text Extender Verwaltung und Programmierung</i>	Dieses Handbuch enthält Basisinformationen zu DB2 Extender, Informationen zur Verwaltung und Konfiguration von Text Extender sowie zur Programmierung mit Hilfe von Text Extender. Es bietet Referenzinformationen, Diagnoseinformationen (mit Nachrichten) und Beispiele.	SC12-2893 desu9g70	desu9

Tabelle 53. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Datei- name	HTML- Verzeichnis
<i>Troubleshooting Guide</i>	Dieses Handbuch hilft Ihnen bei der Bestimmung von Fehlerquellen, bei der Fehlerbehebung sowie bei der Verwendung von Diagnose-Tools, wenn Sie den DB2-Kundendienst in Anspruch nehmen.	GC09-2850 db2p0e70	db2p0
<i>Neue Funktionen</i>	Dieses Handbuch enthält eine Beschreibung der neuen Einrichtungen, Funktionen und Erweiterungen in DB2 Universal Database Version 7.	SC12-2883 db2q0g70	db2q0
DB2-Installations- und -Konfigurationsinformationen			
<i>DB2 Connect Enterprise Edition für OS/2 und Windows Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation und Konfiguration für DB2 Connect Enterprise Edition unter OS/2 und 32-Bit-Windows-Betriebssystemen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2863 db2c6g70	db2c6
<i>DB2 Connect Enterprise Edition für UNIX Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation, Konfiguration und Ausführung von Tasks für DB2 Connect Enterprise Edition auf UNIX-Plattformen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2862 db2cyg70	db2cy
<i>DB2 Connect Personal Edition Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation, Konfiguration und Ausführung von Tasks für DB2 Connect Personal Edition unter OS/2 und 32-Bit-Windows-Betriebssystemen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für alle unterstützten Clients.	GC12-2869 db2c1g70	db2c1
<i>DB2 Connect Personal Edition für Linux Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Migration und Konfiguration für DB2 Connect Personal Edition für alle unterstützten Linux-Varianten.	GC12-2865 db2c4g70	db2c4

Tabelle 53. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Datei- name	HTML- Verzeichnis
<i>DB2 Data Links Manager Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Konfiguration und Ausführung von Tasks für DB2 Data Links Manager unter AIX und 32-Bit-Windows-Betriebssystemen.	GC12-2868 db2z6g70	db2z6
<i>DB2 Enterprise - Extended Edition für UNIX Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation und Konfiguration für DB2 Enterprise - Extended Edition auf UNIX-Plattformen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2867 db2v3g70	db2v3
<i>DB2 Enterprise - Extended Edition für Windows Ein- stieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation und Konfiguration für DB2 Enterprise - Extended Edition unter 32-Bit-Windows-Betriebssystemen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2866 db2v6g70	db2v6
<i>DB2 für OS/2 Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation und Konfiguration von DB2 Universal Database für das Betriebssystem OS/2. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2870 db2i2g70	db2i2
<i>DB2 für UNIX Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation und Konfiguration von DB2 Universal Database auf UNIX-Plattformen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2872 db2ixg70	db2ix

Tabelle 53. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Datei- name	HTML- Verzeichnis
<i>DB2 für Windows Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Migration und Konfiguration für DB2 Universal Database unter 32-Bit-Windows-Betriebssystemen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2873 db2i6g70	db2i6
<i>DB2 Personal Edition Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Migration und Konfiguration für DB2 Universal Database Personal Edition unter OS/2 und 32-Bit-Windows-Betriebssystemen.	GC12-2871 db2i1g70	db2i1
<i>DB2 Personal Edition für Linux Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Migration und Konfiguration für DB2 Universal Database Personal Edition für alle unterstützten Linux-Varianten.	GC12-2874 db2i4g70	db2i4
<i>DB2 Query Patroller Installation Guide</i>	Dieses Handbuch enthält Installationsinformationen zu DB2 Query Patroller.	GC09-2959 db2iwe70	db2iw
<i>DB2 Warehouse Manager Installation</i>	Dieses Handbuch enthält Installationsinformationen für Warehouse-Agenten, Warehouse-Umsetzungsprogramme und den Information Catalog Manager.	GC12-2876 db2ide70	db2id
Plattformübergreifende Beispielprogramme in HTML			
Beispielprogramme in HTML	Dieses Handbuch enthält die Beispielprogramme für die Programmiersprachen auf allen von DB2 unterstützten Plattformen im HTML-Format. Die Beispielprogramme werden lediglich zu Informationszwecken zur Verfügung gestellt. Nicht alle Beispiele sind für alle Programmiersprachen verfügbar. Die HTML-Beispiele stehen nur dann zur Verfügung, wenn der DB2 Application Development Client installiert ist.	Keine Form- nummer	db2hs
	Weitere Informationen zu den Programmen finden Sie im Handbuch <i>Application Building Guide</i> .		

Tabelle 53. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Datei- name	HTML- Verzeichnis
Release-Informationen			
<i>DB2 Connect Release-Informationen</i>	Dieses Dokument enthält die neuesten Informationen, die in die DB2 Connect-Handbücher nicht mehr aufgenommen werden konnten.	Siehe Anmerkung 2.	db2cr
<i>DB2 Installationsinformationen</i>	Dieses Dokument enthält die neuesten Informationen zur Installation, die in die DB2-Handbücher nicht mehr aufgenommen werden konnten.	Nur auf der Produkt-CD-ROM verfügbar.	
<i>DB2-Release-Informationen</i>	Dieses Dokument enthält die neuesten Informationen zu allen DB2-Produkten und -Funktionen, die in die DB2-Handbücher nicht mehr aufgenommen werden konnten.	Siehe Anmerkung 2.	db2ir

Anmerkungen:

1. Das Zeichen an der sechsten Stelle des Dateinamens gibt die Landessprache eines Buchs an. So kennzeichnet der Dateiname db2d0e70 die englische Version des Handbuchs *Systemverwaltung*, der Dateiname db2d0f70 kennzeichnet die französische Version des Buchs. Folgende Buchstaben werden an der sechsten Stelle des Dateinamens verwendet, um die Landessprache für ein Handbuch anzugeben:

Sprache	Kennung
Brasilianisches Portugiesisch	b
Bulgarisch	u
Dänisch	d
Deutsch	g
Englisch	e
Finnisch	y
Französisch	f
Griechisch	a
Italienisch	i
Japanisch	j
Koreanisch	k
Niederländisch	q
Norwegisch	n
Polnisch	p
Portugiesisch	v
Russisch	r
Schwedisch	s
Slowenisch	l
Spanisch	z
Trad. Chinesisch	t
Tschechisch	x
Türkisch	m
Ungarisch	h
Vereinf. Chinesisch	c

2. Kurzfristig verfügbare Informationen, die in die DB2-Handbücher nicht mehr aufgenommen werden können, sind in den Release-Informationen enthalten, die im HTML-Format und als ASCII-Datei verfügbar sind. Die HTML-Version steht über 'Information - Unterstützung' und auf den Produkt-CD-ROMs zur Verfügung. Gehen Sie wie folgt vor, um die ASCII-Dateien anzuzeigen:
 - Rufen Sie auf UNIX-Plattformen die Datei `Release.Notes` auf. Diese Datei befindet sich im Verzeichnis `DB2DIR/Readme/%L`. Dabei ist `%L` die länderspezifische Angabe und `DB2DIR` eine der folgenden Angaben:
 - `/usr/lpp/db2_07_01` (unter AIX)
 - `/opt/IBMDB2/V7.1` (unter HP-UX, PTX, Solaris und Silicon Graphics IRIX)
 - `/usr/IBMDB2/V7.1` (unter Linux)
 - Rufen Sie auf anderen Plattformen die Datei `RELEASE.TXT` auf. Diese Datei befindet sich in dem Verzeichnis, in dem das Produkt installiert ist. Auf OS/2-Plattformen können Sie auch den Ordner **IBM DB2** und anschließend das Symbol **Release-Informationen** doppelt anklicken.

Drucken der PDF-Handbücher

Wenn Sie eine gedruckte Version der Handbücher bevorzugen, können Sie die PDF-Dateien auf der CD-ROM mit DB2-Veröffentlichungen ausdrucken. Mit Adobe Acrobat Reader können Sie entweder das gesamte Handbuch oder bestimmte Teile des Handbuchs ausdrucken. Die Namen der einzelnen Handbücher in der Bibliothek finden Sie in Tabelle 53 auf Seite 523.

Die neueste Version von Adobe Acrobat Reader finden Sie auf der Adobe-Web-Site unter <http://www.adobe.com>.

Die PDF-Dateien befinden sich auf der CD-ROM mit DB2-Veröffentlichungen und haben die Dateierweiterung PDF. Führen Sie folgende Schritte aus, um auf die PDF-Dateien zuzugreifen:

1. Legen Sie die CD-ROM mit DB2-Veröffentlichungen in das CD-ROM-Laufwerk ein. Auf UNIX-Plattformen: Hängen Sie die CD-ROM mit den DB2-Veröffentlichungen an. Das Handbuch *Einstieg* enthält Anweisungen zu den Mount-Prozeduren.
2. Starten Sie Acrobat Reader.
3. Öffnen Sie die gewünschte PDF-Datei von einer der folgenden Positionen aus:
 - Auf OS/2- und Windows-Plattformen:
Verzeichnis `x:\doc\sprache`. Dabei gibt `x` das CD-ROM-Laufwerk an, `sprache` den zweistelligen Landescode für die verwendete Sprache (z. B. EN für Englisch).

- Auf UNIX-Plattformen:
Verzeichnis `/cdrom/doc/%L` auf der CD-ROM. Dabei gibt `/cdrom` den Mount-Punkt der CD-ROM an und `%L` den Namen der gewünschten länderspezifischen Angaben.

Sie können die PDF-Dateien auch von der CD-ROM in ein lokales Laufwerk oder ein Netzlaufwerk kopieren und sie von dort aus lesen.

Bestellen der gedruckten Handbücher

Sie können die gedruckten DB2-Handbücher einzeln bestellen. In den USA und Kanada ist es außerdem möglich, mehrere Bücher als Paket unter einer SBOF-Nummer zu bestellen. Setzen Sie sich mit Ihrem IBM Vertragshändler oder Vertriebsbeauftragten in Verbindung, oder bestellen Sie die Handbücher telefonisch bei IBM Direkt unter der Nummer 0180/55 090. Darüber hinaus können Sie die Handbücher über die Web-Seite mit Veröffentlichungen unter <http://www.elink.ibm.com/pbl/pbl> bestellen.

Es sind zwei Gruppen von Handbüchern verfügbar. Die Gruppe mit der Formnummer SBOF-8935 umfasst Referenzinformationen und Informationen zur Verwendung für DB2 Warehouse Manager. Die Gruppe mit der Formnummer SBOF-8931 umfasst Referenzinformationen und Informationen zur Verwendung für alle anderen DB2 Universal Database-Produkte und -Funktionen. Der Inhalt der SBOF-Gruppen ist in der folgenden Tabelle aufgeführt.

Tabelle 54. Bestellen der gedruckten Handbücher

SBOF-Nummer	In dieser Gruppe enthaltene Handbücher	
SBOF-8931	<ul style="list-style-type: none"> • Administration Guide: Planning • Administration Guide: Implementation • Administration Guide: Performance • Administrative API Reference • Application Building Guide • Application Development Guide • CLI Guide and Reference • Command Reference • Data Movement Utilities Guide and Reference • Data Warehouse Center Administration Guide • Data Warehouse Center Application Integration Guide • DB2 Connect User's Guide • Installation and Configuration Supplement • Image, Audio, and Video Extenders Administration and Programming • Message Reference, Volumes 1 and 2 	<ul style="list-style-type: none"> • OLAP Integration Server Administration Guide • OLAP Integration Server Metaoutline User's Guide • OLAP Integration Server Model User's Guide • OLAP Integration Server User's Guide • OLAP Setup and User's Guide • OLAP Spreadsheet Add-in User's Guide for Excel • OLAP Spreadsheet Add-in User's Guide for Lotus 1-2-3 • Replication Guide and Reference • Spatial Extender Administration and Programming Guide • SQL Getting Started • SQL Reference, Volumes 1 and 2 • System Monitor Guide and Reference • Text Extender Administration and Programming • Troubleshooting Guide • What's New
SBOF-8935	<ul style="list-style-type: none"> • Information Catalog Manager Administration Guide • Information Catalog Manager User's Guide • Information Catalog Manager Programming Guide and Reference 	<ul style="list-style-type: none"> • Query Patroller Administration Guide • Query Patroller User's Guide

Zugreifen auf die Online-Hilfefunktion

Die Online-Hilfefunktion ist für alle DB2-Komponenten verfügbar. In der folgenden Tabelle werden die verschiedenen Hilfearten beschrieben.

Hilfearten	Inhalt	Zugriff
<i>Hilfe für Befehl</i>	Erklärt die Syntax von Befehlen im Befehlszeilenprozessor.	Geben Sie im interaktiven Modus des Befehlszeilenprozessors Folgendes ein: ? <i>befehl</i> Dabei stellt <i>befehl</i> ein Schlüsselwort bzw. den vollständigen Befehl dar. So kann beispielsweise durch die Eingabe von ? catalog Hilfe für alle CATALOG-Befehle angezeigt werden, während mit ? catalog database lediglich Hilfe für den Befehl CATALOG DATABASE angezeigt wird.
<i>Hilfe für Client-Konfiguration - Unterstützung</i>	Erläutert die Tasks, die Sie in einem Fenster oder Notizbuch ausführen können. Die Hilfe umfasst	Klicken Sie in einem Fenster oder in einem Notizbuch den Druckknopf Hilfe an oder drücken Sie die Taste F1 .
<i>Hilfe für die Befehlszentrale</i>	Übersichtsinformationen und unbedingt erforderliche Informationen sowie eine	
<i>Hilfe für die Steuerzentrale</i>	Beschreibung zur Verwendung der Steuerelemente im Fenster oder Notizbuch.	
<i>Hilfe für die Data Warehouse-Zentrale</i>		
Hilfe für Event Analyzer		
<i>Hilfe für Information Catalog Manager</i>		
<i>Hilfe für die Satellitenverwaltungszentrale</i>		
<i>Hilfe für die Prozedurenzentrale</i>		

Hilfearten	Inhalt	Zugriff
<i>Nachrichtenhilfe</i>	Beschreibt die Ursache von Nachrichten sowie die auszuführenden Benutzeraktionen.	<p>Geben Sie im interaktiven Modus des Befehlszeilenprozessors Folgendes ein:</p> <pre>? XXXnnnnn</pre> <p>Dabei ist <i>XXXnnnnn</i> eine gültige Nachrichten- kennung.</p> <p>Bei Eingabe von ? SQL30081 wird z. B. die Hilfe zur Nachricht SQL30081 angezeigt.</p> <p>Wenn Sie die Nachrichtenhilfe seitenweise anzeigen möchten, geben Sie den folgenden Befehl ein:</p> <pre>? XXXnnnnn more</pre> <p>Geben Sie folgenden Befehl ein, um die Nachrichtenhilfe in einer Datei zu speichern:</p> <pre>? XXXnnnnn > datei.erw</pre> <p>Dabei ist <i>datei.erw</i> die Datei, in der Sie die Nachrichtenhilfe speichern möchten.</p>
<i>Hilfe für SQL</i>	Erklärt die Syntax von SQL-Anweisungen.	<p>Geben Sie im interaktiven Modus des Befehlszeilenprozessors Folgendes ein:</p> <pre>help anweisung</pre> <p>Dabei gibt <i>anweisung</i> eine SQL-Anweisung an.</p> <p>So kann beispielsweise durch die Eingabe von <code>help SELECT</code> die Hilfe zur Anweisung <code>SELECT</code> angezeigt werden.</p> <p>Anmerkung: Die Hilfe für SQL ist auf UNIX-Plattformen nicht verfügbar.</p>
<i>SQLSTATE-Hilfe</i>	Erklärt SQLSTATE-Werte und SQL-Klassencodes.	<p>Geben Sie im interaktiven Modus des Befehlszeilenprozessors Folgendes ein:</p> <pre>? sqlstate oder ? klassencode</pre> <p>Dabei ist <i>sqlstate</i> ein gültiger, fünfstelliger SQL-Status, und <i>klassencode</i> stellt die ersten zwei Ziffern des SQL-Statuswerts dar.</p> <p>So kann beispielsweise durch die Eingabe von ? 08003 Hilfe für den SQL-Statuswert 08003 angezeigt werden, während mit ? 08 Hilfe für den Klassencode 08 angezeigt wird.</p>

Anzeigen von Online-Informationen

Die zum Lieferumfang dieses Produkts gehörenden Handbücher werden als Softcopy im HTML-Format (HTML - Hypertext Markup Language) bereitgestellt. In einer Softcopy können Sie die Informationen auf einfache Art suchen und anzeigen und über Hypertextverbindungen auf zugehörige Informationen zugreifen. Außerdem wird die gemeinsame Nutzung der Bibliothek in Ihrem gesamten Unternehmen erleichtert.

Sie können die Online-Bücher und Beispielprogramme mit jedem Browser anzeigen, der den Spezifikationen von HTML Version 3.2 entspricht.

Führen Sie die nachfolgend beschriebenen Schritte aus, um Online-Bücher oder Beispielprogramme anzuzeigen:

- Wenn Sie DB2-Verwaltungs-Tools ausführen, verwenden Sie **Information - Unterstützung**.
- Klicken Sie in einem Browser **Datei**—>**Seite öffnen** an. Die geöffnete Seite enthält eine Übersicht über die DB2-Informationen und Verbindungen (Links) zu diesen Informationen:

- Öffnen Sie auf UNIX-Plattformen die folgende Seite:

```
INSTHOME/sql11ib/doc/%L/html/index.htm
```

Dabei ist %L die länderspezifische Angabe.

- Öffnen Sie auf anderen Plattformen die folgende Seite:

```
sql11ib\doc\html\index.htm
```

Der Pfad befindet sich auf dem Laufwerk, auf dem DB2 installiert ist.

Wenn Sie **Information - Unterstützung** nicht installiert haben, können Sie die Seite öffnen, indem Sie das Symbol **DB2-Informationen** doppelt anklicken. Je nach verwendetem Betriebssystem befindet sich das Symbol im Hauptproduktordner bzw. unter Windows im Menü **Start**.

Installieren des Netscape-Browsers

Wenn Sie nicht bereits einen Web-Browser installiert haben, können Sie Netscape von der im Lieferumfang des Produkts enthaltenen Netscape-CD-ROM aus installieren. Führen Sie folgende Schritte aus, um ausführliche Informationen zur Installation zu erhalten:

1. Legen Sie die Netscape-CD-ROM ein.
2. Nur auf UNIX-Plattformen: Hängen Sie die CD-ROM an. Das Handbuch *Einstieg* enthält Anweisungen zu den Mount-Prozeduren.
3. Installationsanweisungen finden Sie in der Datei `CDNAVnn.txt`. Dabei ist *nn* die zweistellige Landeskennung. Die Datei befindet sich im Stammverzeichnis der CD-ROM.

Zugreifen auf Informationen mit "Information - Unterstützung"

Information - Unterstützung ermöglicht Ihnen den schnellen Zugriff auf DB2-Produktinformationen. **Information - Unterstützung** ist auf allen Plattformen mit DB2-Verwaltungs-Tools verfügbar.

Sie können 'Information - Unterstützung' öffnen, indem Sie das entsprechende Symbol doppelt anklicken. Abhängig vom verwendeten System befindet sich das Symbol im Hauptproduktordner im Ordner 'Information' bzw. unter Windows im Menü **Start**.

Sie können auf 'Information - Unterstützung' auch zugreifen, indem Sie die Funktionsleiste und das Menü **Hilfe** auf der DB2-Windows-Plattform verwenden.

Unter 'Information - Unterstützung' finden Sie sechs verschiedene Arten von Informationen. Klicken Sie die entsprechende Indexzunge an, um die für diese Informationsart verfügbaren Themen aufzurufen.

Funktionen Die Hauptfunktionen, die Sie mit DB2 ausführen können.

Referenz DB2-Referenzinformationen, wie beispielsweise Schlüsselwörter, Befehle und APIs.

Handbücher DB2-Handbücher.

Fehlerbehebung

Kategorien von Fehlermeldungen sowie die entsprechenden Benutzeraktionen.

Beispielprogramme

Beispielprogramme, die in DB2 Application Development Client enthalten sind. Wenn Sie DB2 Application Development Client nicht installiert haben, wird diese Indexzunge nicht angezeigt.

Web DB2-Informationen im World Wide Web. Sie müssen über Ihr System eine Verbindung zum Web herstellen können, um auf diese Informationen zugreifen zu können.

Wenn Sie einen Eintrag aus einer der Listen auswählen, startet **Information - Unterstützung** eine Funktion zum Anzeigen der Informationen. Bei der Anzeigefunktion kann es sich abhängig von der ausgewählten Informationsart um die Hilfeanzeige des Systems, einen Editor oder einen Web-Browser handeln.

In 'Information - Unterstützung' steht eine Suchfunktion zur Verfügung, mit der Sie nach einem bestimmten Thema suchen können, ohne in den Listen blättern zu müssen.

Rufen Sie über die Hypertextverbindung in 'Information - Unterstützung' das Suchformular **In DB2-Online-Informationen suchen** auf.

Der HTML-Such-Server wird normalerweise automatisch gestartet. Wenn eine Suche in HTML-Informationen fehlschlägt, müssen Sie möglicherweise mit einer der nachfolgend aufgeführten Methoden den Such-Server starten:

Unter Windows

Klicken Sie **Start** an und wählen Sie **Programme** → **IBM DB2** → **Informationen** → **HTML-Such-Server starten** aus.

Unter OS/2

Klicken Sie den Ordner **DB2 für OS/2** und anschließend das Symbol für **HTML-Such-Server starten** doppelt an.

Falls andere Probleme bei der Suche in HTML-Informationen auftreten, finden Sie möglicherweise entsprechende Hinweise in den Release-Informationen.

Anmerkung: Die Suchfunktion steht in Linux-, PTX- und Silicon Graphics IRIX-Umgebungen nicht zur Verfügung.

Verwenden der DB2-Assistenten

Assistenten unterstützen Sie bei der Ausführung bestimmter Verwaltungsaufgaben, indem sie Sie Schritt für Schritt durch jede Aufgabe führen. Assistenten stehen über die Steuerzentrale und 'Client-Konfiguration - Unterstützung' zur Verfügung. In der folgenden Tabelle sind die einzelnen Assistenten und deren Verwendungszweck aufgeführt.

Anmerkung: In Umgebungen mit partitionierten Datenbanken sind die Assistenten **Datenbank erstellen**, **Index erstellen**, **Aktualisierung auf mehreren Systemen konfigurieren** und **Leistungskonfiguration** verfügbar.

Assistent	Verwendung	Zugriff
<i>Datenbank hinzufügen</i>	Katalogisieren einer Datenbank auf einer Client-Workstation.	Klicken Sie in Client-Konfiguration - Unterstützung die Option Hinzufügen an.
<i>Datenbank sichern</i>	Festlegen, Erstellen und Terminieren eines Sicherungsplans.	Klicken Sie in der Steuerzentrale die zu sichernde Datenbank mit der rechten Maustaste an und wählen Sie Sichern → Datenbank mit Assistent aus.
<i>Aktualisierung auf mehreren Systemen konfigurieren</i>	Konfigurieren einer Aktualisierung auf mehreren Systemen, einer verteilten Transaktion oder einer zweiphasigen Festschreibung.	Klicken Sie in der Steuerzentrale den Ordner Datenbanken mit der rechten Maustaste an und wählen Sie Aktualisierung auf mehreren Systemen aus.

Assistent	Verwendung	Zugriff
<i>Datenbank erstellen</i>	Erstellen einer Datenbank und Ausführen einiger grundlegender Konfigurationsfunktionen.	Klicken Sie in der Steuerzentrale den Ordner Datenbanken mit der rechten Maustaste an und wählen Sie Erstellen → Datenbank mit Assistent aus.
<i>Tabelle erstellen</i>	Auswählen eines Basisdatentyps und Erstellen eines Primärschlüssels für die Tabelle.	Klicken Sie in der Steuerzentrale das Symbol Tabellen mit der rechten Maustaste an und wählen Sie Erstellen → Tabelle mit Assistent aus.
<i>Tabellenbereich erstellen</i>	Erstellen eines neuen Tabellenbereichs.	Klicken Sie in der Steuerzentrale das Symbol Tabellenbereiche mit der rechten Maustaste an und wählen Sie Erstellen → Tabellenbereich mit Assistent aus.
<i>Index erstellen</i>	Hinweise zum Erstellen und Löschen von Indizes für Ihre Abfragen.	Klicken Sie in der Steuerzentrale das Symbol Index mit der rechten Maustaste an und wählen Sie Erstellen → Index mit Assistent aus.
<i>Leistungs-konfiguration</i>	Optimieren der Leistung einer Datenbank durch Aktualisieren der Konfigurationsparameter, so dass sie den Anforderungen Ihres Unternehmens entsprechen.	Klicken Sie in der Steuerzentrale die Datenbank, die optimiert werden soll, mit der rechten Maustaste an und wählen Sie Leistung mit Assistent konfigurieren aus. Klicken Sie in einer Umgebung mit partitionierten Datenbanken in der Sicht für Datenbankpartitionen die erste Datenbankpartition, die optimiert werden soll, mit der rechten Maustaste an und wählen Sie Leistung mit Assistent konfigurieren aus.
<i>Datenbank wiederherstellen</i>	Wiederherstellen einer Datenbank nach einem Fehler. Dieser Assistent hilft Ihnen, zu entscheiden, welche Sicherungskopie Sie verwenden und welche Protokolle Sie erneut abarbeiten.	Klicken Sie in der Steuerzentrale die Datenbank, die wiederhergestellt werden soll, mit der rechten Maustaste an und wählen Sie Wiederherstellen → Datenbank mit Assistent aus.

Einrichten eines Dokument-Servers

Die DB2-Informationen werden standardmäßig auf Ihrem lokalen System installiert. Das bedeutet, dass alle Benutzer, die Zugriff auf DB2-Informationen benötigen, dieselben Dateien installieren müssen. Führen Sie folgende Schritte aus, um die DB2-Informationen an einer einzigen Position zu speichern:

1. Kopieren Sie alle Dateien und Unterverzeichnisse aus dem Verzeichnis `\sql11ib\doc\html` Ihres lokalen Systems auf einen Web-Server. Jedem Handbuch ist ein Unterverzeichnis zugeordnet, das alle erforderlichen HTML- und GIF-Dateien enthält, aus denen das Handbuch besteht. Stellen Sie sicher, dass die Verzeichnisstruktur erhalten bleibt.
2. Konfigurieren Sie den Web-Server so, dass er die Dateien an der neuen Speicherposition sucht. Informationen hierzu finden Sie im Anhang zu NetQuestion im Handbuch *Installation und Konfiguration Ergänzung*.
3. Wenn Sie die Java-Version von **Information - Unterstützung** verwenden, können Sie eine Basis-URL-Adresse für alle HTML-Dateien angeben. Sie sollten die URL-Adresse für das Bücherverzeichnis verwenden.
4. Wenn Sie die Buchdateien anzeigen können, ist es möglich, bei häufig aufgerufenen Themen Lesezeichen zu setzen. Es empfiehlt sich, folgende Seiten mit einem Lesezeichen zu versehen:
 - Bücherverzeichnis
 - Inhaltsverzeichnis häufig verwendeter Handbücher
 - Themen, auf die häufig verwiesen wird, wie beispielsweise zum Ändern von Tabellen
 - Suchformular

Informationen dazu, wie Sie die DB2 Universal Database-Online-Dokumentationsdateien auf einer zentralen Maschine zur Verfügung stellen können, finden Sie im Anhang zu NetQuestion im Handbuch *Installation und Konfiguration Ergänzung*.

Suchen nach Online-Informationen

Verwenden Sie eine der folgenden Methoden, um nach Informationen in den HTML-Dateien zu suchen:

- Klicken Sie im obersten Rahmen auf **Suchen**. Verwenden Sie das Suchformular, um nach einem bestimmten Thema zu suchen. Diese Funktion steht in Linux-, PIX- oder Silicon Graphics IRIX-Umgebungen nicht zur Verfügung.
- Klicken Sie im obersten Rahmen auf **Index**. Mit Hilfe des Indexes können Sie nach einem bestimmten Thema im Buch suchen.
- Rufen Sie das Inhaltsverzeichnis oder den Index der Hilfe oder des HTML-Buchs auf und verwenden Sie die Suchfunktion des Web-Browsers, um nach einem bestimmten Thema im Buch zu suchen.
- Mit Hilfe der Lesezeichenfunktion des Web-Browsers können Sie schnell zu einem bestimmten Thema zurückkehren.
- Mit Hilfe der Suchfunktion von **Information - Unterstützung** können Sie bestimmte Themen suchen. Weitere Informationen finden Sie in „Zugreifen auf Informationen mit "Information - Unterstützung"“ auf Seite 540.

Anhang M. Bemerkungen

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Dienstleistungen von IBM verwendet werden können. An Stelle der IBM Produkte, Programme oder Dienstleistungen können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb der Produkte, Programme oder Dienstleistungen in Verbindung mit Fremdprodukten und Fremddienstleistungen liegt beim Kunden, soweit nicht ausdrücklich solche Verbindungen erwähnt sind.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanfragen sind schriftlich an IBM Europe, Director of Licensing, 92066 Paris La Defense Cedex, France, zu richten. Anfragen an obige Adresse müssen auf englisch formuliert werden.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen bekanntgegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Handbuch aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt im Rahmen der Allgemeinen Geschäftsbedingungen der IBM, der Internationalen Nutzungsbedingungen der IBM für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer gesteuerten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Garantie, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Informationen über Produkte anderer Hersteller als IBM wurden von den Herstellern dieser Produkte zur Verfügung gestellt, bzw. aus von ihnen veröffentlichten Ankündigungen oder anderen öffentlich zugänglichen Quellen entnommen. IBM hat diese Produkte nicht getestet und übernimmt im Hinblick auf Produkte anderer Hersteller keine Verantwortung für einwandfreie Funktion, Kompatibilität oder andere Ansprüche. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten der IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele der IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogrammes illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden, Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHT-LIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Beispielprogramme geschrieben werden. Die in diesem Handbuch aufgeführten Beispiele sollen lediglich der Veranschaulichung und zu keinem anderen Zweck dienen. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet.

Kopien oder Teile der Beispielprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. _Jahr/Jahre angeben_. Alle Rechte vorbehalten.

Marken

Folgende Namen sind in gewissen Ländern Marken der International Business Machines Corporation.

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
IBM System AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RS/6000
DataPropagator	IBM System /370
DataRefresher	SP
DB2	SQL/DS
DB2 Connect	SQL/400
DB2 Extender	System/370
DB2 OLAP Server	IBM System /390
DB2 Universal Database	SystemView
Distributed Relational	VisualAge
Database Architecture	VM/ESA
DRDA	VSE/ESA
eNetwork	VTAM
Extended Services	WebExplorer
FFST	WIN-OS/2
First Failure Support Technology	

Folgende Namen sind in gewissen Ländern Marken oder eingetragene Marken anderer Unternehmen:

Microsoft, Windows und Windows NT sind Marken oder eingetragene Marken von Microsoft Corporation.

Java und alle auf Java basierenden Marken und Logos sowie Solaris sind in gewissen Ländern Marken von Sun Microsystems, Inc.

Tivoli und NetView sind in gewissen Ländern Marken von Tivoli Systems Inc.

UNIX ist eine eingetragene Marke und wird ausschließlich von der X/Open Company Limited lizenziert.

Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken anderer Unternehmen sein.

Index

Sonderzeichen

\$RAHBUFDIR 415
\$RAHBUFNAME 415
\$RAHCHECKBUF 415
\$RAHENV 423

A

- Abrufen von Daten
 - Index 192
- Active Directory 469
 - konfigurieren 471
 - Objekte 491
 - Sicherheit 487
 - Unterstützung 471
 - Verzeichnisschema erweitern 490
- Aggregatfunktion 159
- Aktualisieren, DAS-Konfiguration 108
- Aktualisieren, Daten in Übersichtstabelle 238
- Aktualisieren, Zeilen typisierter Tabellen 239
- Aktualisieren von Exemplarlisten 107
- Aktualisierung auf mehreren Systemen konfigurieren, Assistent 541
- ALIAS (DB2 für MVS/ESA) 174
- Aliasname
 - Berechtigung 173
 - verwenden 172
- Aliasname erstellen 172
- ALTER, Zugriffsrecht
 - Definition 298
- ALTER COLUMN 220
- ALTER NICKNAME, Anweisung, Beispiel 249
- ALTER SERVER, Anweisung, Beispiel 248
- ALTER TABLE, Anweisung
 - Hinzufügen eindeutiger Integritätsbedingung, Beispiel 223
 - Hinzufügen einer Prüfung auf Integritätsbedingung, Beispiel 227
 - Hinzufügen von Schlüsseln, Beispiel 225
- ALTER TABLE, Anweisung (*Forts.*)
 - Hinzufügen von Spalten, Beispiel 220
 - Löschen einer eindeutigen Integritätsbedingung, Beispiel 227
 - Löschen einer Prüfbedingung, Beispiel 229
 - Löschen von Schlüsseln, Beispiel 228
 - Tipps zum Hinzufügen von Integritätsbedingungen 223
- ALTER TABLESPACE, Anweisung, Beispiel 212
- ALTER VIEW Anweisung, Beispiel 245
- Ändern, benutzerdefinierter strukturierter Typ 238
- Ändern, Datenbankkonfiguration 208
- Ändern, Identitätsspalte 222
- Ändern, Integritätsbedingung 222
- Ändern, Knotengruppe 211
- Ändern, Knotenkonfigurationsdatei 208
- Ändern, Kurzname 249
- Ändern, Partitionierungsschlüssel 234
- Ändern, Server 248
- Ändern, Sicht 244
- Ändern, Spalte 220
- Ändern, Tabelle 218
- Ändern, Tabellenattribute 235
- Ändern, Tabellenbereich 211
- Ändern, Übersichtstabellenmerkmale 237
- Ändern, Umgebungsvariablen 207
- Ändern, Variablen der Profilregistrierdatenbank 207
- Anzeigen
 - Online-Informationen 539
- Anzeigen von SQL-Anweisungen 7
- Anzeigen zugehöriger Objekte 8
- Assistent
 - Datenbank wiederherstellen 542
 - Leistungskonfiguration 208
- Assistent: Datenbank erstellen 5
- Assistent: Datenbank sichern 5
- Assistent: Datenbank wiederherstellen 6
- Assistent: Index erstellen 6
- Assistent: Konfigurieren von Aktualisierungen auf mehreren Systemen 6
- Assistent: Leistungskonfiguration 6
- Assistent: Tabelle erstellen 5
- Assistent: Tabellenbereich erstellen 5
- Assistenten 5
 - Assistenten 541
 - Datenbank erstellen 5, 541
 - Datenbank hinzufügen 541, 542
 - Datenbank sichern 5, 541
 - Datenbank wiederherstellen 6
 - Index 6, 542
 - Konfigurieren von Aktualisierungen auf mehreren Systemen 6, 541
 - Leistungskonfiguration 6, 542
 - Tabelle erstellen 5, 542
 - Tabellenbereich erstellen 5, 542
 - Tasks ausführen 541
- ATTACH, Befehl
 - Angeben von DCE-Informationen (Distributed Computing Environment) 394
 - Übersicht 61
- audit_buf_sz 325
- Auffüllen, typisierte Tabelle 153
- Ausdrücke
 - NEXTVAL 151
 - PREVVAL 151
- Auslöser
 - Abhängigkeiten 159
 - erstellen 156
 - löschen 242
 - Vorteile 157
- Authentifizierung 261
 - DCE-Sicherheitservices 270
 - Definition 261
 - Distributed Computing Environment (DCE), Verzeichnisservices 390
 - Domänensicherheit 436
 - ferner Client 270
 - Gruppen 436

- Authentifizierung (*Forts.*)
 - Überlegungen zu partitionierten Datenbanken 270
 - Verarbeitung, zusammengeslossene Datenbank 278
- Authentifizierungsart 261
 - CLIENT 263
 - DCE 266
 - DCE_SERVER_ENCRYPT 266
 - DCS 265
 - DCS_ENCRYPT 266
 - KERBEROS 267
 - KRB_SERVER_ENCRYPT 267
 - SERVER 262
 - SERVER_ENCRYPT 262
- Automatische Übersichtstabelle 170
- B**
- Befehle
 - CATALOG GLOBAL DATABASE 394
 - Paralleles Ausführen 415
- Befehlszeilenprozessor
 - Binden an eine Datenbank 124
- Befehlszentrale 22
- Behälter
 - Hinzufügen (zu DMS-Tabellenbereich) 212
 - Hinzufügen zu einem SMS-Tabellenbereich 214
 - Modifizieren (zu DMS-Tabellenbereich) 213
- Beispielprogramme
 - HTML 531
 - plattformübergreifend 531
- Benutzer
 - verwalten 44
- Benutzer mit Doppelbytezeichensatz
 - Datentyp 136
- Benutzerauthentifizierung
 - Windows NT 434
- Benutzerdefinierte Funktionen (UDFs)
 - Arten 159
 - erstellen 159
 - löschen 243
 - Zugriffsrecht zum Erstellen nicht abgeschirmter Funktionen 293
- Benutzerdefinierte Spaltenfunktion (UDF) 159
- Benutzerdefinierte Tabellenfunktion 160
- Benutzerdefinierte temporäre Tabelle 147
- Benutzerdefinierte temporäre Tabellen 242
- Benutzerdefinierten einzigartigen Datentyp erstellen 164
- Benutzerdefinierten strukturierten Typ erstellen 165
- Benutzerdefinierter Datentyp (UDT)
 - erstellen 163
 - löschen 244
- Benutzertabellenbereich, temporär 130
- Benutzerzuordnungen
 - erstellen 280
- Berechtigung 288
 - Aufgaben und erforderliche Berechtigungen 316
 - Datenbankverwaltung (DBADM) 291, 294
 - Definition 286
 - Ebenen 286
 - gesicherter Client 263
 - Systempflege (SYSMAINT) 290
 - Systemsteuerung (SYSC-TRL) 289
 - Systemverwaltung (SYS-ADM) 288
 - Widerrufen von DBADM für Benutzer mit SYSADM 289
 - Widerrufen von DBADM für Benutzer mit SYSCtrl 290
- Berechtigungen
 - erteilen 44
 - widerrufen 44
- Berechtigungsnamen
 - Abrufen der Namen mit Berechtigung DBADM 319
 - Abrufen für Informationen über Zugriffsrechte 318
 - Abrufen von Namen mit Tabellenzugriffsberechtigung 319
 - Abrufen von Zugriffsrechten für 320
 - Erstellen einer Sicht für Informationen über Zugriffsrechte 321
- Bereich
 - hinzufügen 220
- Bestimmung von Fehlern mit rah 425
- BIND, Befehl
 - OWNER, Option 308
- BIND, Zugriffsrecht
 - Definition 302
- BINDADD, Zugriffsrecht
 - Definition 293
- Binden
 - Befehlszeilenprozessor 124
 - Datenbankdienstprogramme 124
 - erneutes Binden ungültiger Pakete 307
- Blockorientierte Einheiten 128
- C**
- Call Level Interface (CLI)
 - Binden an eine Datenbank 124
- CATALOG DATABASE
 - Beispiel 125
- CATALOG GLOBAL DATABASE
 - Angaben von DCE-Informationen (Distributed Computing Environment) 394
- CDS 377
- Cell Directory Service (CDS) 377
- CLIENT, Authentifizierungsart 263
- CLIENT, Sicherheitsebene 263
- Clients
 - gesichert 263
 - nicht gesichert 263
 - Verwaltung 4
- collating_sequence, Serveroption 180
- comm_rate, Serveroption 181
- CONNECT, Anweisung
 - Angaben von DCE-Informationen (Distributed Computing Environment) 394
- CONNECT, Zugriffsrecht
 - Definition 293
- connectstring, Serveroption 181
- CONTROL, Zugriffsrecht
 - Definition 297
 - implizites Ausführen 308
 - Zugriffsrechte für Pakete 302
- cpu_ratio, Serveroption 181
- CREATE ALIAS, Anweisung
 - Beispiel 174
 - verwenden 172
- CREATE DATABASE, Befehl
 - Beispiel 116
- CREATE INDEX, Anweisung
 - Beispiel 193
 - eindeutiger Index 193
 - Online-Reorganisation 191, 195
- CREATE NICKNAME 185
- CREATE_NOT_FENCED, Zugriffsrecht
 - Definition 293
- CREATE SERVER 176
- CREATE TABLE, Anweisung
 - Beispiel 137

- CREATE TABLE, Anweisung (*Forts.*)
 - Definieren referenzieller Integritätsbedingungen 142
 - Definieren von Prüfungen auf Integritätsbedingung 146
 - Verwenden mehrerer Tabellenbereiche 154
 - CREATE TABLESPACE, Anweisung
 - Beispiel 127
 - CREATE TRIGGER, Anweisung
 - Beispiel 158
 - CREATE VIEW, Anweisung
 - Ändern von Spaltennamen 168
 - Beispiel 167
 - CREATE WRAPPER 175
 - CREATETAB, Zugriffsrecht
 - Definition 293
 - CURRENT SCHEMA, Sonderregister 62, 134
- D**
- DataPropagator Relational (DPRO-PR)
 - Übersicht 361
 - Daten
 - versetzen 45
 - Verteilung ändern 211
 - Datenbank 57
 - ändern 210
 - Ändern der Verteilung von Daten 211
 - Ermöglichen der Datenpartitionierung 64
 - erstellen 115
 - Erstellen auf allen Knoten 65
 - katalogisieren 125
 - Knotengruppe ändern 211
 - löschen 210
 - Paketabhängigkeiten 253
 - Überlegungen vor dem Ändern 203
 - Überlegungen zum Erstellen 68
 - vor dem Erstellen 57
 - Wiederherstellungsprotokoll 124
 - Datenbank erstellen, Assistent 541
 - Datenbank hinzufügen, Assistent 541, 542
 - Datenbank sichern, Assistent 541
 - Datenbankadministratorberechtigung (DBADM)
 - Abrufen von Namen mit 319
 - Zugriffsrechte 291
 - Datenbanken
 - ferne verwalten 42
 - Datenbankentwurf
 - ändern 203
 - Datenbankkonfiguration
 - ändern 208
 - erstellte Datei 111
 - Datenbankmanager
 - Binden von Dienstprogrammen 124
 - Index 192
 - starten 58
 - stoppen 67
 - Zugriffssteuerung 304
 - Datenbankobjekte
 - Beispiel 378
 - erstellen 378
 - Zugriffssteuerung 304
 - Datenbankpartitionsserver
 - Absetzen von Befehlen 411
 - Windows 2000 447
 - Windows NT 447
 - Datenbankzugriff
 - steuern 257
 - Zugriffsrechte über Pakete bei SQL 309
 - Datendefinitionssprache (DDL)
 - generieren 9
 - Datenintegrität
 - eindeutiger Index 188
 - Datenreplikation 361
 - Datensätze
 - Prüfung 323
 - Datensicherheit
 - Bedeutung der 257
 - Sicherheit des Systemkatalogs 320
 - Steuerung des Datenbankzugriffs 257
 - Datentypen
 - Mehrbytezeichensatz 136
 - Spaltdefinition 136
 - Datenübertragung
 - Übersicht 361
 - Datenverschlüsselung 314
 - DAU (DB_Authentication) 385
 - DB_Authentication (DAU) 385
 - DB_Comment (DCO) 385
 - DB_Communication_Protocol (DCP) 385
 - DB_Database_Locator_Name (DLN) 387
 - DB_Database_Protocol (DDP) 386
 - DB_Native_Database_Name (DNN) 387
 - DB_Object_Type (DOT) 387
 - DB_Principal (DPR) 385
 - DB_Product_Name (DPN) 387
 - DB_Product_Release (DRL) 387
 - DB_Target_Database_Info (DTI) 387
 - DB2
 - auf Windows NT starten 59
 - db2_all 411, 412, 414
 - DB2-Bibliothek
 - Assistenten 541
 - Dokument-Server einrichten 543
 - Drucken von PDF-Handbüchern 534
 - gedruckte Handbücher bestellen 535
 - Handbücher 521
 - Information - Unterstützung 540
 - neueste Informationen 534
 - Online-Hilfefunktion 537
 - Online-Informationen anzeigen 539
 - Online-Informationen suchen 544
 - Sprachenkennung für Bücher 533
 - Struktur 521
 - db2_call_stack 412
 - DB2 Connect 361
 - DB2 für OS/390
 - Objekte verwalten 17
 - Subsysteme hinzufügen 17
 - DB2_INDEX_2BYTEVARLEN 189
 - db2_kill 412
 - DB2-Verwaltungsserver
 - aktualisieren, Konfiguration 108
 - Aktualisieren von Exemplarlisten 107
 - Client-Konfiguration - Unterstützung und Steuerzentrale verwenden 107
 - DB2-Verwaltungsserver (DAS) 97
 - administrative Kommunikation zwischen Knoten 99
 - administrative Kommunikation zwischen Knoten im partitionierten Datenbanksystem (UNIX) 100
 - administrative Kommunikation zwischen Knoten im partitionierten Datenbanksystem (Windows NT) 101
 - aktualisieren 95
 - auflisten 94
 - Discovery aktivieren 103
 - Eigentumsregeln 88
 - einrichten mit einem partitionierten Datenbanksystem 97

- DB2-Verwaltungsserver (DAS)
 - (Forts.)
 - Beispiel 97
 - entfernen 96
 - erstellen 91
 - Kommunikation 99
 - Kommunikation, Steuerzentrale 99
 - Konfiguration 99
 - konfigurieren 94
 - Serviceanschlüsse 99
 - Sicherheit 101
 - starten und stoppen 93
 - Überlegungen zu Variablen der Profilregistrierdatenbank 102
 - Überlegungen zur Sicherheit 95
 - Übersicht 89
 - Umgebung 102
 - UNIX EEE Server 100
 - Variablen der Profilregistrierdatenbank 102
 - Windows NT EEE Server 101
- db2audit 327
- db2audit.log 323
- db2dmnbckctlr
 - verwenden 433, 435
- db2gncol, Dienstprogramm 230
- db2icrt, Befehl 74
- db2idrop 206
- db2ilist 204
- DB2INSTANCE, Umgebungsvariable
 - Definieren des Standard-exemplars 60
- db2iupdt 204
- DB2LDAP_CLIENT_PROVIDER 471
- db2ldcfg, Dienstprogramm 474
- db2nchg 450
- db2ncrt 448
- db2ndrop 451
- db2nlist 447
- db2nodes.cfg, Datei 108
- db2perfc 445
- db2perfi 441
- db2perfr 442
- db2set, Befehl 79, 81
- db2start, Befehl 58
- db2stop, Befehl 67
- dbname, Serveroption 181
- DCE, Authentifizierungsart 266
- DCE-Netzwerkdatenbank
 - erstellen 396
 - verbinden 397, 398
- DCE_SERVER_ENCRYPT, Authentifizierungsart 266
- DCO (DB_Comment) 385
- DCP (DB_Communication_Protocol) 385
- DCS
 - Authentifizierungsart 265
 - Verarbeitung, zusammengesetzte Datenbank 279
- DCS_ENCRYPT, Authentifizierungsart 266
- DDP (DB_Database_Protocol) 386
- DECLARE GLOBAL TEMPORARY TABLE 147
- Dedizierte Verbindung 457, 458
- DEFAULT, Attribut 136
- Definieren
 - Standardumgebungsprofil für rah 425
- Definieren, eindeutige Integritätsbedingung 140
- Definieren, Prüfung auf Integritätsbedingung in Tabellen 145
- Definieren, referenzielle Integritätsbedingung 141
- DELETE, Zugriffsrecht
 - Definition 298
- DETACH, Befehl
 - Übersicht 61
- Direkter Plattenzugriff 130
- Discovery
 - festlegen, Parameter 106
 - inaktivieren für Serverexemplare 105
 - Konfiguration 108
- Distributed Computing Environment (DCE)
 - ATTACH, Befehl 394, 400
 - Aufgaben für Verzeichnisservices 402
 - Authentifizierung 270
 - CATALOG GLOBAL DATABASE 394
 - CDS 377
 - CONNECT, Anweisung 394, 401
 - Einrichten eines DB2-Benutzers 271
 - Einrichten eines DB2-Client-Exemplars 276
 - Einrichten eines DB2-Servers 274
 - Einschränkungen 276
 - Einschränkungen für Verzeichnisservices 406
 - GDS 377
 - Konfigurationsparameter und Variablen der Profilregistrierdatenbank 392
- Distributed Computing Environment (DCE) (Forts.)
 - Sicherheitsservices 270
 - Suchmethode für Verzeichnisse 400
 - temporäres Außerkraftsetzen von DCE-Verzeichnisservices 402
 - Überblick über Verzeichnisservices 122
 - Verwenden von Verzeichnisservices 403
- DLN (DB_Database_Locator_Name) 387
- DMS-Tabellenbereich
 - erstellen 127
- DNN (DB_Native_Database_Name) 387
- Dokument-Server einrichten 543
- Domänensicherheit
 - Authentifizierung 436
- DOT (DB_Object_Type) 387
- DPN (DB_Product_Name) 387
- DPR (DB_Principal) 385
- DPROPR 361
- DRL (DB_Product_Release) 387
- DROP DATABASE, Befehl
 - Beispiel 210
- DROP INDEX, Anweisung, Beispiel 252
- DROP NICKNAME, Anweisung, Beispiel 249
- DROP SERVER, Anweisung, Beispiel 249
- DROP TABLE, Anweisung
 - Beispiel 240
- DROP TABLESPACE, Anweisung, Beispiel 216
- DROP VIEW, Anweisung, Beispiel 245
- Drucken von PDF-Handbüchern 534
- DTI (DB_Target_Database_Info) 387
- Dynamisches SQL
 - EXECUTE, Zugriffsrecht für den Datenbankzugriff 309

E

- Eindeutige Integritätsbedingungen
 - definieren 140
 - hinzufügen 223
 - löschen 227
- Einheiten, unformatiert 128
- Einheiten, zeichenorientiert 128

- Einschränkungen
 - Windows NT-Benennung 434
- Einstellen, Schema 134
- Einstellen, VARCHAR 220
- Eliminieren mehrfacher Einträge aus der Liste der Maschinen 421
- Entwurf implementieren 57
- Entziehen von Berechtigungen und Zugriffsrechten 44
- Erstellen, Aliasname 172
- Erstellen, Auslöser 156
- Erstellen, benutzerdefinierte Funktion 159
- Erstellen, benutzerdefinierter Datentyp 163
- Erstellen, benutzerdefinierter einzigartiger Datentyp 164
- Erstellen, benutzerdefinierter strukturierter Typ 165
- Erstellen, Funktionsschablone 162
- Erstellen, Funktionszuordnung 161
- Erstellen, Index 188
- Erstellen, Indexerweiterung 188
- Erstellen, Indexspezifikation 188
- Erstellen, Kurzname 185
- Erstellen, LDAP-Benutzer 473
- Erstellen, Oberfläche 175
- Erstellen, Schema 133
- Erstellen, Server 176
- Erstellen, Sicht 166
- Erstellen, Tabelle 135
- Erstellen, Tabelle in mehreren Tabellenbereichen 153
- Erstellen, Tabellenbereich 126
- Erstellen, Typenzuordnung 165
- Erstellen, typisierte Sicht 169
- Erstellen, typisierte Tabelle 153
- Erteilen von Berechtigungen und Zugriffsrechten 44
- EXECUTE, Zugriffsrecht
 - Datenbankzugriff mit dynamischem SQL 309
 - Datenbankzugriff mit statischem SQL 309
 - Definition 302
- Exemplar
 - Definition 68
 - Eigner 72
 - entfernen 206
 - Festlegen des aktuellen 78
 - hinzufügen 73
 - standardmäßige 69
 - Verzeichnis 69
- Exemplarbenutzer
 - Einrichten der Umgebung 70
- Exemplare
 - aktualisieren 204
 - ändern 203
 - anzeigen 17
 - auflisten 77, 204
 - automatisch starten 78
 - Datenbankpartitionsserver, auflisten 447
 - erstellen 69
 - Gründe zur Verwendung 68
 - mehrere ausführen 79
 - Nachteile 69
 - partitionierte Server ändern 450
 - partitionierte Server hinzufügen 448
 - partitionierte Server löschen 451
 - starten 58
 - stoppen 67
 - Übersicht 60
- Exemplareigner 72
- Exemplarprofilregistrierdatenbank 80
- Explizite Schemaverwendung 61
- F**
 - FCM-Kommunikation 112
 - Fehlerbehebung 51
 - Fernes System 42
 - Fernverwaltung 97
 - Filter 10
 - fold_id, Serveroption 182
 - fold_pw, Serveroption 182
 - FOREIGN KEY, Klausel
 - Referenzielle Integritätsbedingungen 143
 - Regeln für die Definition von Fremdschlüsseln 143
 - Fremdschlüssel
 - DROP FOREIGN KEY, Klausel in Anweisung ALTER TABLE 227
 - hinzufügen 223
 - IMPORT, Dienstprogramm, Auswirkungen der referenziellen Integrität 145
 - Integritätsbedingungsname 143
 - Load, Dienstprogramm, Auswirkungen der referenziellen Integrität 144
 - Regeln für die Definition von Fremdschlüsseln 143
 - Zugriffsrechte zum Löschen 227
 - zusammengesetzter Schlüssel 143
- Funktionen
 - DECRYPT 315
 - ENCRYPT 314
 - GETHINT 315
- Funktionsaufruf
 - Selektivität 202
- Funktionsschablone, erstellen 162
- Funktionszuordnung, erstellen 161
- G**
 - Gateway-Verbindungen 18
 - GDS 377
 - Generierte Spalte 146, 229
 - Gesicherte Clients
 - Authentifizierung 263
 - CLIENT, Sicherheitsebene 263
 - Global Directory Service (GDS) 377
 - GRANT
 - Beispiel 305
 - GRANT, Anweisung
 - implizites Ausführen 308
 - Sicherheit 392
 - verwenden 304
 - Groß-/Kleinschreibung bei Namen, zusammengeschlossene Datenbank 374
 - Größe
 - ermitteln 48
 - Großes Objekt (LOB)
 - Hinweise zu Spalten 138
 - GUI-Tools
 - Verwaltung mit 3
- H**
 - Handbücher 521, 535
 - Hierarchietabelle 153
 - löschen 241
 - Hinzufügen, Bereich 220
 - Hinzufügen, eindeutige Integritätsbedingung 223
 - Hinzufügen, Integritätsbedingung 223
 - Hinzufügen, Prüfungen auf Integritätsbedingungen in Tabellen 226
 - Hochgeschwindigkeitskommunikation 455
 - HTML
 - Beispielprogramme 531
- I**
 - IBM eNetwork Directory
 - Objektklassen und Attribute 492
 - Verzeichnisschema erweitern 489

- IBMCATGROUP, Knoten-
gruppe 117
- IBMDEFAULTGROUP, Knoten-
gruppe 117
- IBMTEMPGROUP, Knoten-
gruppe 117
- Identifizieren von Kurznamen 187
- Identitätsspalte 148
ändern 235
- IDENTITY, Identitätsspalten 152
- IMPLICIT_SCHEMA, Berechti-
gung 133
- IMPLICIT_SCHEMA, Zugriffsrecht
Definition 294
- Implizite Schemaverwendung 62
- IMPORT, Dienstprogramm
Auswirkungen der referenziellen
Integrität 145
Binden an eine Datenbank 124
LOAD 144
- Index
benutzerdefiniert, erweitert 196
Selektivität 202
- INDEX, Zugriffsrecht
Definition 298
- Index erstellen, Assistent 542
- Indexerweiterung 188
- Indeschlüssel, Definition 189
- Indizes
ändern 251
CREATE INDEX, Anwei-
sung 193
CREATE UNIQUE INDEX,
Anweisung 193
Definition 189
DROP INDEX, Anweisung 252
eindeutiger Index 193
erstellen 188
nichteindeutiger Index 193
Nichtprimärindex 252
Online-Reorganisation 191, 195
optimale Anzahl 189
Primärindex und benutzer-
definierter Index 189
Primärschlüssel 141
verwenden 193
Zugriffsrechte 303
- Information - Unterstützung 540
- INSERT, Zugriffsrecht
Definition 298
- Installation
Netscape-Browser 539
- Integritätsbedingung
ändern 222
definieren, eindeutige 140

- Integritätsbedingung (*Forts.*)
hinzufügen 223
löschen 227
- Integritätsbedingungsname
Definieren von Fremd-
schlüsseln 144
Definieren von Prüfungen auf
Integritätsbedingungen in
Tabellen 145
- io_ratio, Serveroption 183

J

- Java-Applet 53

K

- Katalogisieren, Datenbank 125
- Katalogknoten
Beschreibung 64
- KERBEROS, Authentifizierungs-
art 267
- Kerberos-Sicherheitsprotokoll 267
- Knoten 62
ändern in Knotengruppe 211
Erstellen einer Datenbank auf
allen 65
katalogisieren 64
- Knotengruppe
Erstdefinition 117
- Knotengruppen
ändern 211
erstellen 123
IBMDEFAULTGROUP, Tabelle
standardmäßig erstellen 154
Partitionierungsschlüssel
ändern 234
Überlegungen zu Tabellen 154
- Knotenkonfigurationsdatei
ändern 208
erstellen 108
- Knotennummer 109
- Knotenübergreifende Kommunikati-
on 455
- Kommunikation
Hochgeschwindigkeit 455
- Konfigurationsparameter
Distributed Computing Environ-
ment (DCE) 392
partitionierte Datenbank 65
- Konfigurieren des LDAP-Benutzers
für Anwendungen 474
- Konfigurieren von LDAP 472
- KRB_SERVER_ENCRYPT, Authentifi-
zierungsart 267
- Kurzname
erstellen 185

- Kurznamen

- Paketzugriffsrecht, Verarbei-
tung 309
- Sichten von mehreren Daten-
quellen 311
- Zugriffsrechte 300

L

- LDAP 53, 122, 469
- LDAP-Konfigurationen
Unterstützung 470
- Leistung
Aktivieren des Fernzugriffs auf
Informationen 442
Anzeigen von Informatio-
nen 443
Katalogdaten, Verringern der
Zugriffskonkurrenz 65
Übersichtstabelle 169
Werte zurücksetzen 444
Zugriff auf ferne Informatio-
nen 444
- Leistungskonfiguration, Assis-
tent 208, 542
- Leistungszähler von DB2 für Win-
dows NT 441
- Leitweginformationen, Objekte
Beispiel 382
erstellen 381
- Lightweight Directory Access Proto-
col 122
aktivieren 485
Datenbanken registrieren 478
Datenbankregistrierung zurück-
nehmen 479
DB2 Connect 486
Einträge aktualisieren 480
Host-Datenbanken konfigurie-
ren 482
IBM eNetwork Directory 489
inaktivieren 486
Knoteneintrag katalogisie-
ren 477
Objektklassen und Attribute 492
Protokollinformationen aktuali-
sieren 476
Serverregistrierung zurückneh-
men 477
Sicherheit 486
suchen 481
Variablen der Profilregistrierda-
tenbank einstellen 484
Verbindung zu fernem Server
herstellen 478

Lightweight Directory Access Protocol (Forts.)
 Verzeichnisschema erweitern 489
 Windows 2000 Active Directory 490
 Lightweight Directory Access Protocol (LDAP) 53, 469
 Lizenzinformationen ändern 203
 Lizenzverwaltung 79
 Lizenzzentrale 26
 LOAD, Dienstprogramm
 Übersicht 361
 LOAD-Berechtigung 292
 Logische Knoten
 mehrere 453
 Lokales Datenbankverzeichnis
 Übersicht 120
 Löschen, Auslöser 242
 Löschen, benutzerdefinierte Funktion 243
 Löschen, benutzerdefinierte Tabellen 242
 Löschen, benutzerdefinierter Datentyp 244
 Löschen, Benutzertabellenbereich 216
 Löschen, Datenbank 210
 Löschen, eindeutige Integritätsbedingung 227
 Löschen, Index 251
 Löschen, Indexerweiterungen 251
 Löschen, Indexspezifikationen 251
 Löschen, Integritätsbedingung 227
 Löschen, Kurzname 249
 Löschen, Oberfläche 248
 Löschen, Prüfungen auf Integritätsbedingungen in Tabellen 229
 Löschen, Schema 218
 Löschen, Server 248
 Löschen, Sicht 244
 Löschen, Tabelle 240
 Löschen, temporärer Benutzertabellenbereich 218
 Löschen, temporärer Systemtabellenbereich 216
 Löschen, Übersichtstabelle 246
 Löschen, Zeilen aus typisierten Tabellen 239

M

Mehrere Exemplare 60
 MINPCTUSED, Klausel 195
 Modifizieren, Spalte 220

Modifizieren, Tabelle 218

N

Nachrichten
 Prüffunktion 332
 Namenskonvention
 allgemeine 369
 Namenskonventionen
 Windows NT-Einschränkungen 434
 Netscape-Browser
 installieren 539
 Neueste Informationen 534
 Neuverteilen von Daten
 auf Knoten 211
 NEXTVAL 151
 Nichteindeutige Indizes
 löschen 252
 Nichtgesicherte Clients 263
 Nichtprimärindizes
 Auswirkungen des Löschens auf Anwendungen 252
 löschen 252
 node, Serveroption 183
 Nullwert
 Spaltendefinition 136
 numeric_string, Spaltenoption 250

O

Oberfläche, erstellen 175
 Objekte
 zugehörige anzeigen 8
 Objektklassenattribute
 DB_Authentication (DAU) 385
 DB_Comment (DCO) 385
 DB_Communication_Protocol 385
 DB_Database_Locator_Name 387
 DB_Database_Protocol 386
 DB_Native_Database_Name 387
 DB_Object_Type 387
 DB_Principal (DPR) 385
 DB_Product_Name 387
 DB_Product_Release 387
 DB_Target_Database_Info 387
 Öffentliche Verbindungen 456
 Öffnen des Journals 25
 Online-Hilfefunktion 537
 Online-Informationen
 anzeigen 539
 suchen 544
 Online-Reorganisation
 Indizes 191

P

Paket
 unbrauchbar 253
 Pakete
 Abhängigkeiten 253
 Eigner 308
 löschen 252
 ungültig nach Hinzufügen eines Fremdschlüssels 224
 Widerrufen von Zugriffsrechten 307
 Zugriffsrechte 302
 Zugriffsrechte bei SQL 309
 Parallelität
 aktivieren 62
 Parallelität, partitionsintern
 aktivieren 62
 Partitionierung von Daten 64
 Partitionierungsschlüssel
 ändern 234
 Index nach Partitionierungsschlüssel partitioniert 191
 Überlegungen zu Tabellen 154
 Partitionsinterne Parallelität
 aktivieren 62
 password, Serveroption 183
 PDF 534
 plan_hints, Serveroption 184
 Präfixsequenzen 420
 PRECOMPILE, Befehl
 OWNER, Option 308
 PREVVAl 151
 Primärindex
 Eindeutigkeit für Primärschlüssel 141
 löschen 252
 Primärschlüssel
 DROP PRIMARY KEY, Klausel in Anweisung ALTER TABLE 227
 hinzufügen 223
 Primärindex 141
 Primärindex, erstellen 189
 Zeitpunkt der Erstellung 141
 Zugriffsrechte zum Löschen 227
 PRIMARY KEY, Klausel
 Einschränkungen 141
 Hinzufügen eines Primärschlüssels 223
 Profilregistrierdatenbank 79
 Profilregistrierdatenbank auf Exemplarebene 80
 Profilregistrierdatenbank auf globaler Ebene 80

Profilregistrierdatenbank auf Knoten-
ebene 80

Protokoll

Prüfung 323

Protokollieren

Einheiten, zeichenorientiert 130

Prozedurzentrale 23

vorhandene Prozedur verwenden
24

Prüfaktivitäten 323

Prüffunktion

Aktionen 324

Aktivitäten steuern 353

Arbeitsweise 325

asynchrones Schreiben von

Datensätzen 326

Beispiele 353

Berechtigungen/Zugriffsrechte 323

CONTEXT, Ereignistabelle 349

Ereignisse 324

Ereignistabellen überprüfen 333

ERRORTYPE, Parameter 326

Fehlerbehandlung 326

Nachrichten 332

OBJMAINT, Ereignistabelle 338

Parameterbeschreibung 328

Prüfereignistabelle 333

Satzaufbau 332

SECMAINT, Ereignistabelle 340

synchrones Schreiben von Daten-
sätzen 326

Syntax 327

SYSADMIN, Ereignistabelle 344

Tipps und Verfahrensweisen 352

VALIDATE, Ereignistabelle 348

Verwendungsszenarios 327

Prüfprotokoll 323

Prüfung 323

Prüfung auf Integritätsbedingung in
Tabellen

definieren 145

hinzufügen 226

löschen 229

PUBLIC

Zugriffsrechte 294

pushdown, Serveroption 184

Q

Qualifizierte Objektnamen 61

Querverweisobjekt für Datenbanken

Beispiel 380

erstellen 380

R

rah 412

Angaben der Liste von Maschi-
nen 420

Definieren des Standard-
umgebungsprofils 425

Einführung 411

Paralleles Ausführen von Befeh-
len 415

RAHDOTFILES 424

RAHOSTFILE 420

RAHOSTLIST 420

RAHWAITTIME 415

steuern 422

Überwachung von Prozes-
sen 415

Umgebungsvariablen 422

RAHCHECKBUF 415

RAHTREETHRESH 417

REFERENCES, Klausel

Hinzufügen eines Fremd-
schlüssels 223

Löschbedingungen 144

Referenzielle Integritäts-
bedingung 144

verwenden 144

REFERENCES, Zugriffsrecht

Definition 298

Referenzielle Integritätsbedingungen
definieren 141

FOREIGN KEY, Klausel in

Anweisungen CREATE/ALTER
TABLE 141

Hinzufügen in einer Tabelle 223

PRIMARY KEY, Klausel in

Anweisungen CREATE/ALTER
TABLE 141

REFERENCES, Klausel in Anwei-
sungen CREATE/ALTER
TABLE 141

Release-Informationen 534

REORG, Dienstprogramm

Binden an eine Datenbank 124

Replikation 361

Konfiguration 112

Replizieren von Daten 52

Resource Access Control Facility

(RACF) 392

REVOKE, Anweisung

Beispiel 306

implizites Ausführen 308

Sicherheit 392

verwenden 305

S

Schema

erstellen 133

löschen 218

SESSION 242

Übersicht 61

SEARCH, Discovery-Modus

zusätzliche Einstellungen 104

SELECT, Anweisung

Auswählen einer Sicht 168

SELECT, Zugriffsrecht

Definition 298

Selektivität 202

Sequenzen 152

ändern 236

erstellen 150

löschen 237

Zugriffsrechte 303

Server

erstellen 176

Zugriffsrechte 302

SERVER, Authentifizierungsart 262

SERVER_ENCRYPT

Authentifizierungsart 262

Server-eOptionen

password 183

Serveroptionen

collating_sequence 180

comm_rate 181

connectstring 181

cpu_ratio 181

dbname 181

fold_id 182, 282

fold_pw 182, 282

io_ratio 183

Knoten 183

password 282

plan_hints 184

pushdown 184

Sicherheit, Einzelangaben 281

varchar_no_trailing_blanks 184

SET ENCRYPTION PASS-

WORD 315

Sicherheit

APPC-Einstellung für Systeme

zusammengeschlossener Daten-
banken 283

Authentifizierung, zusammege-

schlossene Datenbank 278

Benutzerzuordnungen 280

CLIENT, Stufe 263

DCS-Verarbeitung, System

zusammengeschlossener Daten-
banken 279

- Sicherheit (*Forts.*)
 - Distributed Computing Environment (DCE), Verzeichnis-services 390
 - planen 257
 - Server mit zusammengeschlossenen Datenbanken, Beispiel für eine Authentifizierung 283
 - Serveroptionen 281
 - Services, Windows NT 434
 - zusammengeschlossene Datenbank, ID- und Kennwortverarbeitung 280
 - Sicherungs-Domänensteuereinheit
 - Konfiguration von DB2 433, 435
 - Sicht
 - wiederherstellen, unbrauchbare 246
 - Sichten
 - ändern 244
 - Auswirkungen des Löschens auf Systemkataloge 244
 - CHECK OPTION, Klausel, CREATE VIEW, Anweisung 168
 - Datenintegrität 166
 - Datensicherheit 166
 - Einschränkungen 244
 - erstellen 166
 - für Informationen über Zugriffsrechte 321
 - löschen 244
 - Spaltenzugriff 311
 - unbrauchbar 246
 - Zugriff auf Zeilen 311
 - Zugriffsrechte, Beispiele 312
 - Zugriffssteuerung für Tabelle 311
 - SIGTTIN 414
 - Skalare benutzerdefinierte Funktion 159
 - SMS-Tabellenbereich
 - erstellen 127
 - Hinzufügen von Behältern 214
 - Spalten
 - ändern 220
 - definieren 136
 - hinzufügen 219
 - Spaltenoptionen
 - numerische Zeichenfolge 250
 - varchar_no_trailing_blanks 251
 - Speicher
 - Verwaltung 48
 - Sprachenkennung
 - Handbücher 533
 - SQL-Anweisungen
 - anzeigen 7
 - unbrauchbar 253
 - Starten von DB2 58
 - Statisches SQL
 - EXECUTE, Zugriffsrecht für den Datenbankzugriff 309
 - stdin 414
 - Steuern des Befehls rah 422
 - Steuerzentrale
 - angepasste 14
 - Anzeigen von Systemen 16
 - Steuerzentrale als Java-Applet 54
 - Stoppen von DB2 67
 - Strukturierter Typ
 - ändern 238
 - Suchen
 - Online-Informationen 541, 544
 - Suchen von Objekten 20
 - SWITCH ONLINE, Klausel 215
 - SYNONYM (DB2 für MVS/ESA) 174
 - SYSADM, Berechtigung 288
 - Übersicht 288
 - Zugriffsrechte 288
 - SYSCAT-Sichten 317
 - SYSCATSPACE, Tabellenbereich 117
 - Übersicht 120, 121
 - Systemkatalog
 - Abrufen der Namen mit Berechtigung DBADM 319
 - Abrufen für Namen erteilter Zugriffsrechte 320
 - Abrufen von Berechtigungs-namen mit Zugriffsrechten 318
 - Abrufen von Namen mit Tabellenzugriffs-berechtigung 319
 - Auswirkungen des Löschens einer Sicht 245
 - einrichten 119
 - Hinzufügen einer neuen Spalte 219
 - Liste der Zugriffsrechte 317
 - Löschen einer Tabelle 240
 - Sicherheit 320
 - Systemkatalogtabellen
 - gespeichert auf dem Katalog-knoten der Datenbank 65
 - Systemmonitor
 - Windows NT 441
 - Systemtabellenbereich, temporär 129
- ## T
- Tabelle
 - ändern 218
 - Ändern von Attributen 235
 - Erstellen in partitionierter Datenbank 154
 - flüchtige 233
 - generierte Spalte 146, 229
 - Identitätsspalte 148
 - temporär 117
 - umbenennen 239
 - Tabelle erstellen, Assistent 542
 - Tabellen
 - Abrufen von Namen mit Zugriff 319
 - ALTER TABLE, Anweisung 220
 - Ändern von Partitionierungsschlüsseln 234
 - Benennung 135
 - CREATE TABLE, Anweisung 135
 - Definieren eindeutiger Integritätsbedingungen 140
 - Definieren einer Prüfung auf Integritätsbedingung 145
 - Definieren referenzieller Integritätsbedingungen 142
 - Hinzufügen referenzieller Integritätsbedingungen 223
 - löschen 240
 - Widerrufen von Zugriffsrechten 306
 - Zuordnen zu Knotengruppe 123
 - Tabellenbereich
 - ändern 211
 - Behälter erweitern 213
 - Behälter hinzufügen 212
 - Behältergröße anpassen 213
 - Benutzer, temporär 130
 - Dateibehälter, Beispiel 127
 - Dateisystembehälter, Beispiel 127
 - Einheitenbehälter, Beispiel 127
 - erstellen 126
 - in Knotengruppen 130
 - löschen 216
 - Standardwert bei Datenbankerstellung 117
 - System, temporär 129
 - temporär für Benutzer, löschen 218
 - temporär für System, löschen 216
 - Trennen von Datentypen, Beispiel 154

- Tabellenbereich (*Forts.*)
 - umbenennen 215
 - Zugriffsrechte 297
 - Tabellenbereich erstellen, Assistent 542
 - Tabellenbereiche
 - Festlegen des Status ONLINE 215
 - Kapazität hinzufügen 50
 - Überprüfen des verfügbaren Speichers (DMS) 49
 - TCP/IP 456
 - Temporäre Tabelle
 - benutzerdefiniert 147
 - Temporäre Tabellen
 - benutzerdefinierte löschen 242
 - TEMPSPACE1, Tabellenbereich 117
 - Typenzuordnung, erstellen 165
 - Typisierte Sicht erstellen 169
 - Typisierte Tabelle
 - Aktualisieren von Zeilen 239
 - auffüllen 153
 - erstellen 153
 - Hierarchietabelle 153
 - Typisierte Tabellen
 - Löschen von Zeilen 239
- U**
- Übergeben von IDs und Kennwörtern an Datenquellen 280
 - Übersichtstabelle
 - automatisch 170
 - Daten aktualisieren 238
 - erstellen 169
 - löschen 246
 - Merkmale ändern 237
 - Übersichtstabellen
 - wiederherstellen, unbrauchbare 247
 - Übertragungsprotokolle
 - VI Architecture 457
 - Überwachung
 - rah-Prozesse 415
 - Umbenennen, Tabelle 239
 - Umbenennen eines Tabellenbereichs 215
 - Umgebungsvariablen 79
 - ändern 207
 - einstellen unter OS/2 84
 - einstellen unter UNIX 87
 - einstellen unter Windows 95 85
 - einstellen unter Windows NT 85
 - rah 422
 - RAHDOTFILES 424
 - Umschreiben einer Abfrage
 - Übersichtstabelle 169
 - Unformatierte Ein-/Ausgabe 130
 - UPDATE, Zugriffsrecht
 - Definition 298
 - USAGE, Zugriffsrecht 303
 - USERSPACE1, Tabellenbereich 117
- V**
- varchar_no_trailing_blanks, Serveroption 184
 - varchar_no_trailing_blanks, Spaltenoption 251
 - Variablen der Profilregistrierdatenbank 79
 - Distributed Computing Environment (DCE) 392
 - Variablen der Registrierdatenbank ändern 207
 - Verschlüsselung
 - Daten 314
 - Versetzen von Daten 361
 - Verwaltung
 - mit Tools für die GUI 3
 - Verwaltungs-Tools
 - Befehlszentrale 22
 - Prozedurzentrale 23
 - Übersicht 4
 - Verwaltungsserver 16, 89
 - Verzeichnis-Cache
 - Auswirkung des Katalogisierens von Datenbanken 125
 - Verzeichnisobjekt
 - erstellen 377
 - Objektklassenattribute 384
 - Verzeichnisse
 - Knotenverzeichnis 121
 - lokales Datenbankverzeichnis 120
 - Systemdatenbankverzeichnis 120
 - VI
 - Einrichten für die Verwendung mit DB2 467
 - VI Architecture 457
 - Virtual Interface (VI) Architecture 457, 458
 - Virtual Telecommunications Access Method (VTAM) 392
- W**
- Wiederherstellen, Assistent 542
 - Wiederherstellen, unbrauchbare Sicht 246
 - Wiederherstellen, unbrauchbare Übersichtstabelle 247
 - Wiederherstellung
 - Übersicht 365, 409
 - Zuordnen eines Protokolls bei der Datenbankerstellung 124
 - Wiederherstellungsprotokoll 124
 - Windows 2000 Active Directory Objekte 491
 - Verzeichnisschema erweitern 490
 - Windows NT Active Directory
 - Objektklassen und Attribute 492
 - Windows NT Performance Monitor 441
 - DB2 registrieren 441
- Z**
- Zeichenorientierte serielle Einheiten 128
 - Zeilen
 - entfernen 221
 - Zeitplanung
 - für gespeicherte Befehlsprozeduren 24
 - Zugriffsrechte
 - Abrufen für Namen 320
 - Abrufen von Berechtigungs-namen mit 318
 - ALTER 298
 - Aufgaben und erforderliche Berechtigungen 316
 - BINDADD 293
 - CONNECT 293
 - CONTROL 297
 - CREATE_NOT_FENCED 293
 - CREATETAB 293
 - Datenbankmanager 293
 - Definition 286
 - DELETE 298
 - Eigentumszugriffsrecht (CONTROL) 287
 - Erstellen einer Sicht zu Informationszwecken 321
 - erteilen 44
 - Erteilen und Entziehen von Berechtigungen 293
 - GRANT, Anweisung 304
 - Hierarchie 286
 - IMPLICIT_SCHEMA 293
 - implizite für Pakete 287
 - INDEX 303
 - indirektes Zugriffsrechte, Kurznamen 309
 - individuelle 287

- Zugriffsrechte (*Forts.*)
 - INSERT 298
 - Kurzname 300
 - Paket 302
 - PUBLIC 294
 - REFERENCES 298
 - REVOKE, Anweisung 305
 - Schemata 295
 - SELECT 298
 - Server 302
 - Sicht 297
 - Sichten mit Kurznamen 311
 - Systemkatalogliste 317
 - Tabelle 297
 - Tabellenbereiche 297
 - USAGE 303
 - widerrufen 44
 - Zusammenfassung 286
- Zugriffssteuerung 261
 - Authentifizierung 261
 - Datenbankmanager 304
 - Datenbankobjekte 304
 - Sicht auf Tabelle 311
- Zuordnung von Dateien mit freien Bereichen 139
- Zusammengeschlossene Datenbank
 - Funktionsschablone, erstellen 162
 - Funktionszuordnung, erstellen 161
 - Indexspezifikation, erstellen 188
 - Kurzname, arbeiten mit 186
 - Kurzname, erstellen 185
 - Kurzname, identifizieren 187
 - Kurznamen, verweisen auf 186
 - Namen in Groß-/Kleinschreibung 374
 - Oberfläche, erstellen 175
 - Server, erstellen 176
 - Typenzuordnung, erstellen 165
 - Übergeben von IDs und Kennwörtern an Datenquellen 280
- Zusammengeschlossene Datenbanken
 - APPC-Einstellung 283
 - Authentifizierung 278
 - Authentifizierung, Beispiel 283
 - Benutzerzuordnungen erstellen 280
 - DCS-Einstellung 279
 - Serveroptionen, Sicherheit 281

Kontaktaufnahme mit IBM

Bei technischen Problemen lesen Sie bitte die entsprechenden Korrekturmaßnahmen im Handbuch *Troubleshooting Guide* und führen Sie diese aus, bevor Sie sich mit der IBM Kundenunterstützung in Verbindung setzen. Mit Hilfe dieses Handbuchs können Sie Informationen sammeln, die die DB2-Kundenunterstützung zur Fehlerbehebung verwenden kann.

Wenn Sie weitere Informationen benötigen oder eines der DB2 Universal Database-Produkte bestellen möchten, setzen Sie sich mit einem IBM Ansprechpartner in einer lokalen Geschäftsstelle oder einem IBM Software-Vertriebspartner in Verbindung.

Telefonische Unterstützung erhalten Sie unter der folgenden Nummer:

- Unter 0180 3/313 233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.

Produktinformationen

Telefonische Unterstützung erhalten Sie unter den folgenden Nummern:

- Unter 0180 3/313 233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0180/55 090 können Sie Handbücher telefonisch bestellen.

<http://www.ibm.com/software/data/>

Auf den DB2-World Wide Web-Seiten erhalten Sie aktuelle DB2-Informationen wie Neuigkeiten, Produktbeschreibungen, Schulungspläne und vieles mehr.

<http://www.ibm.com/software/data/db2/library/>

Mit **DB2 Product and Service Technical Library** können Sie auf häufig gestellte Fragen, Berichtigungen, Handbücher und aktuelle technische DB2-Informationen zugreifen.

Anmerkung: Diese Informationen stehen möglicherweise nur auf Englisch zur Verfügung.

<http://www.elink.ibm.com/pbl/pbl/>

Auf der Web-Site für die Bestellung internationaler Veröffentlichungen (International Publications) finden Sie Informationen zum Bestellverfahren.

<http://www.ibm.com/education/certify/>

Das 'Professional Certification Program' auf der IBM Web-Site stellt Zertifizierungstestinformationen für eine Reihe von IBM Produkten, u. a. auch DB2, zur Verfügung.

<ftp://software.ibm.com>

Melden Sie sich als *anonymous* an. Im Verzeichnis `/ps/products/db2` finden Sie Demo-Versionen, Berichtigungen, Informationen und Tools zu DB2 und vielen zugehörigen Produkten.

<comp.databases.ibm-db2>, <bit.listserv.db2-1>

Über diese Internet-Newsgroups können DB2-Benutzer Ihre Erfahrungen mit den DB2-Produkten austauschen.

Für Compuserve: GO IBMDB2

Geben Sie diesen Befehl ein, um auf IBM DB2 Family Forums zuzugreifen. Alle DB2-Produkte werden über diese Foren unterstützt.

In Anhang A des Handbuchs *IBM Software Support Handbook* finden Sie Informationen dazu, wie Sie sich mit IBM in Verbindung setzen können. Rufen Sie die folgende Web-Seite auf, um auf dieses Dokument zuzugreifen:

<http://www.ibm.com/support/>. Wählen Sie anschließend die Verbindung zum IBM Software Support Handbook am unteren Rand der Seite aus.

Anmerkung: In einigen Ländern sollten sich die IBM Vertragshändler an die innerhalb ihrer Händlerstruktur vorgesehene Unterstützung wenden, nicht an die IBM Unterstützungsfunktion.



SC12-2878-01

