

IBM DB2 Universal Database



# Systemverwaltung: Implementierung

*Version 8.2*



IBM DB2 Universal Database



# Systemverwaltung: Implementierung

*Version 8.2*

#### Hinweis

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

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

Diese Veröffentlichung ist eine Übersetzung des Handbuchs  
*IBM DB2 Universal Database Administration Guide: Implementation Version 8.2*,  
IBM Form SC09-4820-01,  
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 1993-2004  
© Copyright IBM Deutschland GmbH 2004

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

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

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:  
SW TSC Germany  
Kst. 2877  
April 2004

# Inhaltsverzeichnis

<b>Zu diesem Handbuch</b> . . . . .	<b>ix</b>
Zielgruppe . . . . .	x
Aufbau dieses Handbuchs . . . . .	x
Kurzübersicht über die anderen Bände des Handbuchs zur Systemverwaltung . . . . .	xii
Systemverwaltung: Konzept . . . . .	xii
Systemverwaltung: Optimierung . . . . .	xiii

## Teil 1. Implementieren des Datenbankentwurfs . . . . . 1

### Kapitel 1. Vor dem Erstellen einer Datenbank . . . . . 3

Arbeiten mit Exemplaren . . . . .	4
Starten von DB2 UDB unter UNIX . . . . .	4
Starten von DB2 UDB unter Windows . . . . .	5
Mehrere Exemplare des Datenbankmanagers . . . . .	6
Herstellen der Verbindung zu einem anderen Exemplar des Datenbankmanagers . . . . .	7
Gruppieren von Objekten nach Schema . . . . .	8
Parallelität . . . . .	9
Stoppen eines Exemplars unter UNIX . . . . .	15
Stoppen eines Exemplars unter Windows . . . . .	16
Vorbereiten der Erstellung einer Datenbank . . . . .	17
Entwerfen logischer und physischer Datenbankmerkmale . . . . .	17
Exemplarerstellung . . . . .	18
Automatisches Einrichten der DB2 UDB-Umgebung unter UNIX . . . . .	20
Manuelles Einrichten der DB2 UDB-Umgebung unter UNIX . . . . .	20
Mehrere Exemplare unter einem UNIX-Betriebssystem . . . . .	21
Mehrere Exemplare unter einem Windows-Betriebssystem . . . . .	22
Erstellen zusätzlicher Exemplare . . . . .	23
UNIX - Details beim Erstellen von Exemplaren . . . . .	24
Windows - Details beim Erstellen von Exemplaren . . . . .	25
Hinzufügen eines Exemplars . . . . .	26
Auflisten von Exemplaren . . . . .	27
Festlegen des aktuellen Exemplars . . . . .	27
Automatisches Starten von Exemplaren . . . . .	28
Gleichzeitiges Ausführen mehrerer Exemplare . . . . .	29
Lizenzverwaltung . . . . .	29
Umgebungsvariablen und die Profilregistrierdatenbank . . . . .	29
Deklarieren von Registrier- und Umgebungsvariablen . . . . .	33
Definieren von Umgebungsvariablen unter Windows . . . . .	35
Definieren von Umgebungsvariablen auf UNIX-Systemen . . . . .	37
Erstellen einer Knotenkonfigurationsdatei . . . . .	39

Erstellen der Datenbankkonfigurationsdatei . . . . .	42
FCM-Kommunikation . . . . .	43

### | Kapitel 2. Erstellen und Verwenden des DB2-Verwaltungsservers (DAS) . . . . . 45

DB2-Verwaltungsserver . . . . .	45
Erstellen eines DB2-Verwaltungsservers . . . . .	47
Starten und Stoppen des DB2-Verwaltungsservers . . . . .	48
Auflisten des DB2-Verwaltungsservers . . . . .	49
Konfigurieren des DB2-Verwaltungsservers . . . . .	50
Einrichten und Konfigurieren der Toolskatalogdatenbank und des DAS-Scheduler . . . . .	50
Einrichten und Konfigurieren der Benachrichtigung und der Kontaktliste . . . . .	55
Einrichten der virtuellen Java-Maschine für den DB2-Verwaltungsserver . . . . .	56
Sicherheitsaspekte des DB2-Verwaltungsservers unter Windows . . . . .	57
Aktualisieren des DB2-Verwaltungsservers unter UNIX . . . . .	57
Entfernen des DB2-Verwaltungsservers . . . . .	58
Einrichten des DB2-Verwaltungsservers mit Enterprise Server Edition (ESE)-Systemen . . . . .	59
DAS-Konfiguration auf Enterprise Server Edition (ESE)-Systemen . . . . .	61
Erkennen (Discovery) von Verwaltungsservern, Exemplaren und Datenbanken . . . . .	63
Ausschließen von Serverexemplaren und Datenbanken aus Discovery-Erkennung . . . . .	64
Einstellen von Discovery-Parametern . . . . .	65
Konfigurieren des DAS zur Verwendung des Konfigurationsassistenten und der Steuerzentrale . . . . .	66
Aktualisieren der DAS-Konfiguration für Discovery FFDC (First Failure Data Capture) durch den DB2-Verwaltungsserver . . . . .	67

### Kapitel 3. Erstellen einer Datenbank . . . . . 69

Erstellen einer Datenbank . . . . .	69
Definition der ersten Datenbankpartitionsgruppen . . . . .	70
Definieren der ersten Tabellenbereiche . . . . .	71
Erstellen eines Pufferpools . . . . .	72
Definition von Systemkatalogtabellen . . . . .	73
Definition von Datenbankverzeichnissen . . . . .	74
Lokales Datenbankverzeichnis . . . . .	74
Systemdatenbankverzeichnis . . . . .	75
Angeben eines alternativen Servers für eine Datenbank . . . . .	75
Anzeigen des Inhalts der Datei des lokalen oder des Systemdatenbankverzeichnisses . . . . .	76
Knotenverzeichnis . . . . .	76
LDAP-Verzeichnisservice (Lightweight Directory Access Protocol) . . . . .	77
Erstellen von Datenbankpartitionsgruppen (Knotengruppen) . . . . .	78

	Definition des Datenbankwiederherstellungsprotokolls . . . . .	79
	Implementierung der automatischen Clientweiterleitung . . . . .	79
	Binden von Dienstprogrammen an die Datenbank	80
	Katalogisieren einer Datenbank . . . . .	81
	Aktualisieren der Verzeichnisse mit Informationen über ferne Datenbankservermaschinen . . . . .	82
	Erstellen eines Tabellenbereichs . . . . .	83
	Erstellen bestimmter Typen von Tabellenbereichen	86
	Erstellen eines Tabellenbereichs für temporäre Systemtabellen . . . . .	86
	Erstellen eines Tabellenbereichs für temporäre Benutzertabellen . . . . .	87
	Erstellen von Tabellenbereichen in Datenbankpartitionierungsgruppen . . . . .	88
	Angabe eines direkten E/A-Zugriffs (Raw I/O)	88
	Einrichten eines direkten E/A-Zugriffs (Raw I/O) unter Linux . . . . .	90
	Erstellen eines Schemas . . . . .	92
	Details zum Erstellen von Schemata . . . . .	93
	Einstellen eines Schemas . . . . .	93
	<b>Kapitel 4. Erstellen von Tabellen und anderen zugehörigen Tabellenobjekten . 95</b>	
	Erstellen und Füllen einer Tabelle . . . . .	95
	Details zum Erstellen und Füllen einer Tabelle.	97
	Einführung in die Speicherplatzkomprimierung für Tabellen . . . . .	97
	Speicherplatzkomprimierung für neue Tabellen	98
	Überlegungen zu Spalten für große Objekte (LOB) . . . . .	98
	Definieren von Integritätsbedingungen . . . . .	100
	Definieren einer Prüfung auf Integritätsbedingung in Tabellen . . . . .	105
	Definieren einer informativen Integritätsbedingung . . . . .	106
	Definieren einer generierten Spalte in einer neuen Tabelle . . . . .	107
	Erstellen einer benutzerdefinierten temporären Tabelle . . . . .	108
	Definieren einer Identitätsspalte in einer neuen Tabelle . . . . .	109
	Erstellen einer Sequenz . . . . .	111
	Vergleich von IDENTITY-Spalten und Sequenzen	112
	Beispiele für Bereichsclustertabellen . . . . .	113
	Funktionsweise des SQL-Compilers bei Bereichsclustertabellen . . . . .	115
	Richtlinien zur Verwendung von Bereichsclustertabellen . . . . .	116
	Definieren von Dimensionen für eine Tabelle	116
	Erstellen einer Hierarchietabelle oder einer typisierten Tabelle . . . . .	118
	Füllen einer typisierten Tabelle . . . . .	119
	Erstellen einer Tabelle in mehreren Tabellenbereichen . . . . .	119
	Erstellen einer Tabelle in einer partitionierten Datenbank . . . . .	120
	Erstellen eines Auslösers . . . . .	122
	Auslöserabhängigkeiten . . . . .	124

	Verwenden von Auslösern zum Aktualisieren von Sichtinhalten . . . . .	125
	Erstellen einer benutzerdefinierten Funktion (UDF) oder Methode . . . . .	126
	Details zum Erstellen einer benutzerdefinierten Funktion (UDF) oder Methode . . . . .	127
	Erstellen einer Funktionszuordnung . . . . .	127
	Erstellen einer Funktionsschablone . . . . .	128
	Benutzerdefinierter Datentyp (UDT) . . . . .	129
	Details zum Erstellen eines benutzerdefinierten Datentyps (UDT) . . . . .	130
	Erstellen eines benutzerdefinierten einzigartigen Datentyps . . . . .	130
	Erstellen eines benutzerdefinierten strukturierten Typs . . . . .	132
	Erstellen einer Typenzuordnung . . . . .	132
	Erstellen einer Sicht . . . . .	133
	Details zum Erstellen einer Sicht . . . . .	136
	Erstellen einer typisierten Sicht . . . . .	136
	Erstellen einer gespeicherten Abfragetabelle . . . . .	136
	Erstellen einer benutzerverwalteten gespeicherten Abfragetabelle . . . . .	139
	Füllen einer benutzerverwalteten gespeicherten Abfragetabelle . . . . .	140
	Erstellen einer Zwischenspeichertabelle . . . . .	141
	Erstellen eines Aliasnamens . . . . .	143
	Index, Indexerweiterung oder Indexspezifikation	145
	Details zum Erstellen eines Index, einer Indexerweiterung oder einer Indexspezifikation . . . . .	148
	Erstellen eines Index . . . . .	148
	Verwenden eines Index . . . . .	149
	Optionen der Anweisung CREATE INDEX . . . . .	150
	Erstellen eines benutzerdefinierten erweiterten Indextyps . . . . .	155
	Details zum Erstellen eines benutzerdefinierten erweiterten Indextyps . . . . .	156
	Details zur Indexverwaltung . . . . .	156
	Details zur Indexsuche . . . . .	156
	Details zur Indexausnutzung . . . . .	157
	Ein Szenario zum Definieren einer Indexerweiterung . . . . .	159
	Aufrufen des Konfigurationsadvisors über den Befehlszeilenprozessor . . . . .	161

## **Kapitel 5. Ändern einer Datenbank 163**

	Ändern eines Exemplars . . . . .	163
	Ändern von Exemplaren (nur UNIX) . . . . .	163
	Details zum Ändern von Exemplaren . . . . .	164
	Ändern der Knoten- und Datenbankkonfigurationsdatei . . . . .	168
	Ändern der Datenbankkonfiguration über mehrere Partitionen . . . . .	169
	Ändern einer Datenbank . . . . .	170
	Löschen einer Datenbank . . . . .	170
	Ändern einer Datenbankpartitionierungsgruppe . . . . .	171
	Ändern eines Tabellenbereichs . . . . .	171
	Details zum Ändern eines Tabellenbereichs . . . . .	172
	Löschen eines Schemas . . . . .	182
	Ändern eines Pufferpools . . . . .	183

<b>Kapitel 6. Ändern von Tabellen und anderen zugehörigen Tabellenobjekten</b>	<b>185</b>
Modifizieren vorhandener Tabellen und der zugehörigen Tabellenobjekte	185
Speicherplatzkomprimierung für vorhandene Tabellen	185
Ändern einer Tabelle mit Hilfe einer gespeicherten Prozedur	186
Hinzufügen von Spalten in einer vorhandenen Tabelle	188
Modifizieren einer Spaltendefinition	189
Entfernen von Zeilen aus einer Tabelle oder Sicht	191
Modifizieren des Merkmals GENERATED oder IDENTITY einer Spalte	192
Modifizieren einer Identitätsspalte	193
Ändern einer Integritätsbedingung	193
Hinzufügen einer Integritätsbedingung	193
Löschen einer eindeutigen Integritätsbedingung	197
Definieren einer generierten Spalte in einer vorhandenen Tabelle	200
Deklarieren einer flüchtigen Tabelle	203
Ändern von Partitionierungsschlüsseln	203
Ändern von Tabellenattributen	204
Ändern einer Identitätsspalte	205
Ändern einer Sequenz	205
Löschen einer Sequenz	206
Ändern der Merkmale einer gespeicherten Abfragetabelle	207
Aktualisieren der Daten in einer gespeicherten Abfragetabelle	208
Ändern eines benutzerdefinierten strukturierten Typs	208
Löschen und Aktualisieren von Zeilen einer typisierten Tabelle	209
Umbenennen einer vorhandenen Tabelle oder eines Index	209
Aktualisieren von Tabellen- und Sichtinhalten mit der Anweisung MERGE	210
Löschen einer Tabelle	212
Löschen einer benutzerdefinierten temporären Tabelle	213
Löschen eines Auslösers	214
Löschen einer benutzerdefinierten Funktion (UDF), Funktionszuordnung oder Methode	214
Löschen eines benutzerdefinierten Datentyps (UDT) oder einer Typenzuordnung	215
Ändern oder Löschen einer Sicht	216
Wiederherstellen unbrauchbarer Sichten	218
Löschen einer gespeicherten Abfragetabelle oder einer Zwischenspeichertabelle	218
Wiederherstellen unbrauchbarer Übersichtstabellen	219
Löschen eines Index, einer Indexerweiterung oder einer Indexspezifikation	220
Anweisungsabhängigkeiten beim Ändern von Objekten	221

## Teil 2. Datenbanksicherheit . . . . 225

<b>Kapitel 7. Steuern des Datenbankzugriffs</b>	<b>227</b>
Sicherheitsaspekte bei der Installation von DB2	
Universal Database	227
Abrufen von Windows-Benutzergruppeninformationen mit einem Zugriffstoken	229
Details zur Sicherheit auf der Basis des Betriebssystems	231
Windows NT-Plattform: Sicherheitsaspekte für Benutzer	232
Unterstützung für das lokale Systemkonto unter Windows	233
UNIX-Plattform: Sicherheitsaspekte für Benutzer	233
Position des Exemplarverzeichnisses	233
Sicherheits-Plug-ins	234
Authentifizierungsmethoden für den Server	234
Authentifizierungsaspekte für ferne Clients	240
Authentifizierungsaspekte bei partitionierten Datenbanken	240
Kerberos-Authentifizierung - Details	241
Kerberos - Beschreibung und Einführung	241
Kerberos - Einrichtung	242
Kerberos und Client-Principals	242
Kerberos und Zuordnung von Berechtigungs-IDs	243
Kerberos und Server-Principals	243
Kerberos-Chiffrierschlüsseldateien	244
Kerberos und Gruppen	244
Aktivieren der Kerberos-Authentifizierung auf dem Client	244
Aktivieren der Kerberos-Authentifizierung auf dem Server	244
Erstellen eines Kerberos-Plug-ins	245
Zugriffsrechte, Berechtigungsstufen und Datenbankberechtigungen	245
Objekterstellung, Eigentumsrechte und Zugriffsrechte	249
Details zu Zugriffsrechten und Berechtigungen	250
Systemverwaltungsberechtigung (SYSADM)	250
Systemsteuerungsberechtigung (SYSCTRL)	251
Systempflegeberechtigung (SYSMAINT)	252
Datenbankadministratorberechtigung (DBADM)	253
Systemmonitorberechtigung (SYSMON)	253
LOAD-Berechtigung	254
Datenbankberechtigungen	255
Überlegungen zur Berechtigung IMPLICIT_SCHEMA	257
Zugriffsrechte für Schemata	257
Zugriffsrechte für Tabellenbereiche	259
Zugriffsrechte für Tabellen und Sichten	259
Zugriffsrechte für Pakete	261
Zugriffsrechte für Indizes	262
Zugriffsrechte für Sequenzen	263
Zugriffsrechte für Routinen	263
Steuern des Zugriffs auf Datenbankobjekte	263
Details zur Steuerung des Zugriffs auf Datenbankobjekte	264
Erteilen von Zugriffsrechten	264
Widerrufen von Zugriffsrechten	265
Verwalten impliziter Berechtigungen durch Erstellen und Löschen von Objekten	267

Einrichten eines Eigentumsrechts für ein Paket	268
Indirekte Zugriffsrechte durch ein Paket	268
Indirekte Zugriffsrechte durch ein Paket mit Kurznamen	269
Steuern des Zugriffs auf Daten mit Sichten	270
Überwachen des Datenzugriffs mit der Prüffunktion	273
Datenverschlüsselung	273
Aufgaben und erforderliche Berechtigungen	274
Verwenden des Systemkatalogs zu Sicherheitszwecken	276
Details zur Verwendung des Systemkatalogs in Bezug auf die Sicherheit	277
Abrufen von Berechtigungsnamen mit erteilten Zugriffsrechten	277
Abrufen aller Namen mit der Berechtigung DBADM	277
Abrufen der Namen mit Zugriffsberechtigung für eine Tabelle	278
Abrufen aller Benutzern erteilter Zugriffsrechte	278
Schützen der Systemkatalogsicht	279
Einführung in die Firewallunterstützung	280
Screening-Router-Firewalls	281
Proxy-Firewalls der Anwendungsebene	281
Circuit-Level-Firewalls	282
SMLI-Firewalls (Stateful Multi-Layer Inspection)	282

## Kapitel 8. Prüfen von Aktivitäten in DB2 Universal Database (DB2 UDB). . . . . 283

Einführung in die Prüffunktion von DB2 Universal Database (DB2 UDB)	283
Funktionsweise der Prüffunktion	285
Verwendung der Prüffunktion	287
Arbeiten mit DB2-Prüfdaten in DB2-Tabellen	290
Arbeiten mit DB2-Prüfdaten in DB2-Tabellen	291
Erstellen von Tabellen zur Aufnahme der DB2-Prüfdaten	291
Erstellen von DB2-Prüfdatendateien	294
Laden von DB2-Prüfdaten in Tabellen	296
Auswählen von DB2-Prüfdaten aus Tabellen	298
Nachrichten der Prüffunktion	300
Aufbau der Datensätze der Prüffunktion (Einführung)	300
Details zu den Datensatzlayouts der Prüffunktion	301
Prüfsatzaufbau für AUDIT-Ereignisse	301
Prüfsatzaufbau für CHECKING-Ereignisse	301
Prüfsatzobjekttypen	303
Liste möglicher Gründe für die Zugriffsgewährung bei CHECKING-Ereignissen	304
Liste der möglichen Typen von Zugriffsversuchen bei CHECKING-Ereignissen	305
Prüfsatzaufbau für OBJMAINT-Ereignisse	307
Prüfsatzaufbau für SECMAINT-Ereignisse	309
Liste der möglichen Zugriffsrechte oder Berechtigungen für SECMAINT-Ereignisse	310
Prüfsatzaufbau für SYSADMIN-Ereignisse	313
Liste der möglichen SYSADMIN-Prüfereignisse	315
Prüfsatzaufbau für VALIDATE-Ereignisse	316
Prüfsatzaufbau für CONTEXT-Ereignisse	317
Liste der möglichen CONTEXT-Prüfereignisse	319
Tipps und Techniken für die Prüffunktion	319

Steuern der Aktivitäten der DB2 UDB-Prüffunktion	321
--	-----

## Teil 3. Anhänge und Schlussteil 325

### Anhang A. Namenskonventionen . . . . . 327

Allgemeine Namenskonventionen	327
Namenskonventionen für DB2 UDB-Objekte	327
Namen von begrenzten Bezeichnern und Objekten	329
Namenskonventionen für Benutzer, Benutzer-IDs und Gruppen	330
Namenskonventionen für Objekte in zusammengesetzten Datenbanken	331
Weitere Einschränkungen und Empfehlungen zur Verwendung von Schemanamen	331
Kennwortverwaltung auf Servern	331
Namenskonventionen für Workstations	332
Namenskonventionen in einer NLS-Umgebung	333
Namenskonventionen in einer Unicode-Umgebung	334

### Anhang B. Verwenden der automatischen Clientweiterleitung . . . . . 335

Automatische Clientweiterleitung - Beschreibung und Einrichtung	335
Automatische Clientweiterleitung - Einschränkungen	336
Automatische Clientweiterleitung - Beispiele	338

### Anhang C. Verwenden der LDAP-Verzeichnisservices (Lightweight Directory Access Protocol) . . . . . 341

Einführung in Lightweight Directory Access Protocol (LDAP)	341
Unterstützte LDAP-Client- und -Serverkonfigurationen	342
Unterstützung für Active Directory	343
Konfigurieren von DB2 zur Verwendung von Active Directory	344
Konfigurieren von DB2 in einer IBM LDAP-Umgebung	345
Erstellen eines LDAP-Benutzers	346
Konfigurieren des LDAP-Benutzers für DB2-Anwendungen	347
Registrierung von DB2-Servern nach der Installation	347
Aktualisieren der Protokollinformationen für den DB2-Server	349
Weiterleiten von LDAP-Clients an andere Server	349
Katalogisieren eines Knoten-Aliasnamens für ATTACH	350
Zurücknehmen der Registrierung des DB2-Servers	351
Registrierung von Datenbanken im LDAP-Verzeichnis	351
Herstellen einer ATTACH-Verbindung zu einem fernen Server in der LDAP-Umgebung	352
Zurücknehmen der Registrierung der Datenbank aus dem LDAP-Verzeichnis	353
Aktualisieren der LDAP-Einträge in lokalen Datenbank- und Knotenverzeichnissen	353



Durchsuchen der LDAP-Verzeichnispartitionen oder -Domänen . . . . .	354
Registrieren von Hostdatenbanken in LDAP . . . . .	355
Definieren von DB2-Registriervariablen auf Benutzerebene in der LDAP-Umgebung . . . . .	357
Aktivieren der LDAP-Unterstützung nach Abschluss der Installation . . . . .	357
Inaktivieren der LDAP-Unterstützung . . . . .	358
LDAP-Unterstützung und DB2 Connect . . . . .	359
Sicherheitsaspekte in einer LDAP-Umgebung . . . . .	359
Sicherheitsaspekte für Active Directory . . . . .	360
Erweitern des LDAP-Verzeichnisseschemas mit DB2-Objektklassen und -Attributen . . . . .	361
Erweitern des Verzeichnisseschemas für Active Directory . . . . .	361
DB2-Objekte im Active Directory . . . . .	363
Netscape-Unterstützung und Attributdefinitionen für das LDAP-Verzeichnis . . . . .	363
Erweitern des Verzeichnisseschemas für IBM SecureWay Directory Server . . . . .	365
Erweitern des Verzeichnisseschemas für Sun One Directory Server . . . . .	367
Von DB2 verwendete LDAP-Objektklassen und Attribute . . . . .	369

**Anhang D. Absetzen von Befehlen an mehrere Datenbankpartitionen . . . . . 381**

Absetzen von Befehlen in einer partitionierten Datenbankumgebung . . . . .	381
Übersicht über die Befehle rah und db2_all . . . . .	381
Beschreibungen der Befehle rah und db2_all . . . . .	382
Angaben der Befehle rah und db2_all . . . . .	383
Paralleles Ausführen von Befehlen auf Plattformen unter UNIX . . . . .	384
Überwachen von rah-Prozessen auf Plattformen unter UNIX . . . . .	385
Zusätzliche Informationen zum Befehl rah (nur Solaris und AIX) . . . . .	386
Präfixsequenzen für den Befehl rah . . . . .	387
Angaben der Liste von Maschinen in einer partitionierten Umgebung . . . . .	389
Eliminieren doppelter Einträge aus einer Liste von Maschinen in einer partitionierten Umgebung . . . . .	390
Steuern des Befehls rah . . . . .	391
Verwenden von \$RAHDOTFILES auf UNIX-Plattformen . . . . .	392
Einstellen des Standardumgebungsprofils für rah unter Windows NT . . . . .	393
Bestimmen von Fehlern mit rah auf Plattformen unter UNIX . . . . .	394

**Anhang E. Verwenden der Unterstützung für Windows Management Instrumentation (WMI) . . . . . 397**

Einführung in Windows Management Instrumentation (WMI) . . . . .	397
Integration von DB2 Universal Database in Windows Management Instrumentation . . . . .	398

**Anhang F. Verwenden der Windows NT-Sicherheit . . . . . 401**

Einführung in DB2 für Windows NT und in die Windows NT-Sicherheit . . . . .	401
Ein DB2 für Windows NT-Szenario mit Serverauthentifizierung . . . . .	402
Ein DB2 für Windows NT-Szenario mit Clientauthentifizierung und einer Windows NT-Clientmaschine . . . . .	403
Ein DB2 für Windows NT-Szenario mit Clientauthentifizierung und einer Windows 9x-Clientmaschine . . . . .	404
Unterstützung globaler Gruppen (unter Windows) . . . . .	404
Verwenden eines Sicherungsdomänencontrollers mit DB2 UDB . . . . .	405
Benutzerauthentifizierung mit DB2 für Windows NT . . . . .	405
Einschränkungen für DB2 für Windows NT-Benutzer- und Gruppennamen . . . . .	405
Gruppen- und Benutzerauthentifizierung unter Windows NT . . . . .	406
Vertrauensstellungen zwischen Domänen unter Windows NT . . . . .	407
DB2 für Windows NT-Sicherheitsservice . . . . .	407
Installieren von DB2 auf einem Sicherungsdomänencontroller . . . . .	408
DB2 für Windows NT-Authentifizierung mit Gruppen und Domänensicherheit . . . . .	409
Authentifizierung mit einer geordneten Domänenliste . . . . .	410
DB2 für Windows NT-Unterstützung für die Domänensicherheit . . . . .	411

**Anhang G. Verwenden des Windows-Systemmonitors . . . . . 413**

Windows-Systemmonitor - Einführung . . . . .	413
Registrieren von DB2 im Windows-Systemmonitor . . . . .	413
Aktivieren des Fernzugriffs auf DB2-Leistungsinformationen . . . . .	414
Anzeigen von Leistungswerten für DB2 UDB und DB2 Connect . . . . .	415
Windows-Leistungsobjekte . . . . .	415
Zugreifen auf ferne DB2 UDB-Leistungsinformationen . . . . .	416
Zurücksetzen von DB2-Leistungswerten . . . . .	417

**Anhang H. Verwenden von Windows-Datenbankpartitionsservern . . . . . 419**

Auflisten der Datenbankpartitionsserver eines Exemplars . . . . .	419
Hinzufügen eines Datenbankpartitionsservers zu einem Exemplar (Windows) . . . . .	420
Ändern der Datenbankpartition (Windows) . . . . .	421
Löschen einer Datenbankpartition aus einem Exemplar (Windows) . . . . .	423

**Anhang I. Konfigurieren mehrerer logischer Knoten . . . . . 425**

Möglichkeiten zur Verwendung mehrerer logischer Knoten . . . . .	425	Anzeigen von Themen in der gewünschten Sprache in 'DB2 Information - Unterstützung' . . . . .	462
Konfigurieren mehrerer logischer Knoten . . . . .	425	DB2-Dokumentation in PDF-Format und gedrucktem Format . . . . .	462
<b>Anhang J. Erweitern der Steuerzentrale . . . . .</b>	<b>427</b>	DB2-Kerninformationen . . . . .	463
Einführung der Plug-in-Architektur für die Steuerzentrale . . . . .	427	Verwaltungsinformationen . . . . .	463
Richtlinien für Entwickler von Plug-ins für die Steuerzentrale . . . . .	427	Informationen zur Anwendungsentwicklung	464
Kompilieren und Ausführen der Beispiel-Plug-ins	428	Informationsmanagement . . . . .	465
Schreiben von Plug-ins als Erweiterungen der Steuerzentrale . . . . .	429	Informationen zu DB2 Connect . . . . .	465
Beschreibungen von Plug-in-Tasks . . . . .	430	Einführungsinformationen . . . . .	465
Erstellen eines Plug-ins, das einen Symbolleistenknopf hinzufügt . . . . .	430	Lernprogramminformationen . . . . .	466
Erstellen eines Plug-ins, das dem Datenbankobjekt neue Menüelemente hinzufügt . . . . .	431	Informationen zu Zusatzkomponenten . . . . .	466
Erstellen eines Plug-ins, das Plug-in-Objekte unter dem Datenbankobjekt in der Baumstruktur hinzufügt. . . . .	436	Release-Informationen . . . . .	467
Inaktivieren von Konfigurationsfunktionen mit isConfigurable() . . . . .	446	Drucken von DB2-Büchern mit PDF-Dateien . . . . .	468
Inaktivieren der Möglichkeit zum Ändern von Objekten mit isEditable() . . . . .	446	Bestellen gedruckter DB2-Bücher . . . . .	468
Inaktivieren der Standardwertknöpfe in Konfigurationsdialogen mit hasConfigurationDefaults() . . . . .	447	Aufrufen der Kontexthilfe über ein DB2-Tool . . . . .	469
<b>Anhang K. Technische Informationen zu DB2 Universal Database . . . . .</b>	<b>449</b>	Aufrufen der Hilfe für Nachrichten über den Befehlszeilenprozessor . . . . .	470
DB2-Dokumentation und Hilfe . . . . .	449	Aufrufen der Hilfe für Befehle über den Befehlszeilenprozessor . . . . .	471
Aktualisierungen der DB2-Dokumentation . . . . .	449	Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor . . . . .	471
DB2 Information - Unterstützung. . . . .	450	DB2-Lernprogramme . . . . .	472
DB2 Information - Unterstützung: Installations-szenarios . . . . .	452	Informationen zur Fehlerbehebung in DB2 . . . . .	472
Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (UNIX) . . . . .	454	Eingabehilfen . . . . .	473
Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (Windows) . . . . .	457	Tastatureingabe und Navigation . . . . .	474
Aufrufen von 'DB2 Information - Unterstützung' . . . . .	459	Eingabehilfen für Bildschirme . . . . .	474
Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung' . . . . .	461	Kompatibilität mit Unterstützungseinrichtungen	474
		Dokumentation im behindertengerechten Format. . . . .	474
		Syntaxdiagramme in der Schreibweise mit Trennzeichen . . . . .	475
		Common Criteria-Zertifizierung von DB2 Universal Database-Produkten . . . . .	477
		<b>Anhang L. Bemerkungen . . . . .</b>	<b>479</b>
		Marken . . . . .	481
		<b>Index . . . . .</b>	<b>483</b>
		<b>Kontaktaufnahme mit IBM . . . . .</b>	<b>493</b>
		Produktinformationen . . . . .	493

---

## Zu diesem Handbuch

Das Handbuch zur Systemverwaltung bietet in seinen drei Bänden Informationen, die zur Verwendung und Verwaltung der 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** ermöglicht Ihnen den Zugriff auf Datenbanken und deren Bearbeitung über eine grafische Schnittstelle. 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** ermöglicht Ihnen die Ausführung von DB2-Dienstprogrammfunktionen innerhalb eines Anwendungsprogramms. Weitere Informationen zur Verwendung der Anwendungsprogrammierschnittstelle finden Sie im Handbuch *Administrative API Reference*.
- Die **Steuerzentrale** ermöglicht Ihnen die Ausführung von Verwaltungsaufgaben, z. B. das Konfigurieren des Systems, die Verwaltung von Verzeichnissen, das Sichern und Wiederherstellen des Systems, die zeitliche Terminierung von Jobs und die Verwaltung von Medien, über eine grafische Schnittstelle. Die Steuerzentrale enthält außerdem eine Replikationsverwaltung, mit der die Replikation von Daten zwischen den Systemen eingerichtet werden kann. Darüber hinaus ermöglicht die Steuerzentrale das Ausführen von DB2-Dienstprogrammfunktionen über eine grafische Benutzerschnittstelle. Je nach Plattform gibt es unterschiedliche Möglichkeiten, die Steuerzentrale aufzurufen. Geben Sie zum Beispiel den Befehl db2cc in der Befehlszeile ein, wählen Sie das Symbol der Steuerzentrale im DB2-Ordner aus oder verwenden Sie auf Windows-Plattformen das Menü **Start**. Wenn Sie eine einführende Hilfe benötigen, wählen Sie **Erste Schritte** im Menü **Hilfe** des Fensters der Steuerzentrale aus. Das Tool **Visual Explain** wird über die Steuerzentrale aufgerufen.

Die Steuerzentrale ist in drei Sichten verfügbar:

- Basissicht. Diese Sicht zeigt die zentralen DB2 UDB-Funktionen für wesentliche Objekte wie Datenbanken, Tabellen und gespeicherte Prozeduren.
- Erweiterte Sicht. Diese Sicht enthält alle verfügbaren Objekte und Aktionen. Verwenden Sie diese Sicht, wenn Sie in einer Unternehmensumgebung arbeiten und eine Verbindung zu DB2 für z/OS oder IMS herstellen wollen.
- Angepasste Sicht. Diese Sicht gibt Ihnen die Möglichkeit, die Objektbaumstruktur und die Objektaktionen individuell anzupassen.

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

- Der Befehlseditor, der die Befehlszentrale ersetzt und zum Generieren, Editieren, Ausführen und Manipulieren von SQL-Anweisungen, IMS- und DB2-Befehle,

sowie zum Arbeiten mit der resultierenden Ausgabe und zum Anzeigen einer grafischen Darstellung des Zugriffsplans für mit EXPLAIN bearbeitete SQL-Anweisungen dient.

- Die Entwicklungszentrale, die eine Unterstützung für native gespeicherte SQL-Prozeduren des Persistent Storage Module (PSM), für gespeicherte Java-Prozeduren für iSeries Version 5 Release 3 und spätere Versionen sowie für benutzerdefinierte Funktionen (UDFs) und strukturierte Typen bereitstellt.
- Die Diagnosezentrale, die ein Tool zur Verfügung stellt, das Datenbankadministratoren (DBA) bei der Lösung von Problemen mit der Leistung und der Ressourcenzuordnung unterstützt.
- Das Programm „Tools - Einstellungen“, das zur Änderung von Einstellungen für die Steuerzentrale, die Diagnosezentrale und die Replikationszentrale dient.
- 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 in erster Linie für Datenbankadministratoren, Systemadministratoren, Sicherheitsadministratoren und Systembediener gedacht, die eine Datenbank für den Zugriff durch lokale oder ferne Clients 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 Universal Database™ (DB2 UDB) benötigen.

---

## Aufbau dieses Handbuchs

Das vorliegende Handbuch enthält Informationen zu folgenden Hauptthemen:

### Implementieren des Datenbankentwurfs

- Kapitel 1, „Vor dem Erstellen einer Datenbank“, beschreibt die Voraussetzungen für die Erstellung einer Datenbank und der Objekte innerhalb einer Datenbank.
- Kapitel 2, „Erstellen und Verwenden des DB2-Verwaltungsservers (DAS)“, erläutert, was ein DAS ist, wie er erstellt wird und wie er verwendet wird.
- Kapitel 3, „Erstellen einer Datenbank“, beschreibt die Aufgaben im Zusammenhang mit der Erstellung einer Datenbank und der Objekte innerhalb einer Datenbank.
- Kapitel 4, „Erstellen von Tabellen und anderen zugehörigen Tabellenobjekten“, beschreibt die Erstellung von Tabellen mit spezifischen Merkmalen bei der Implementierung eines Datenbankentwurfs.
- Kapitel 5, „Ändern einer Datenbank“, beschreibt die Voraussetzungen und Aufgaben im Zusammenhang mit dem Ändern oder Löschen einer Datenbank und der Objekte innerhalb einer Datenbank.
- Kapitel 6, „Ändern von Tabellen und anderen zugehörigen Tabellenobjekten“, beschreibt das Löschen von Tabellen bzw. das Ändern der diesen Tabellen zugeordneten spezifischen Merkmale. Das Löschen und Ändern zusammengehöriger Tabellenobjekte wird ebenfalls an dieser Stelle erläutert.

## Datenbanksicherheit

- Kapitel 7, „Steuern des Datenbankzugriffs“, beschreibt die Steuerung des Zugriffs auf die Datenbankressourcen.
- Kapitel 8, „Prüfen von Aktivitäten in DB2 Universal Database™ (DB2 UDB)“, beschreibt Methoden zur Erkennung und Überwachung unerwünschten bzw. unvorhergesehenen Zugriffs auf Daten.

## Anhänge

- Anhang A, „Namenskonventionen“, enthält die Regeln, die bei der Benennung von Datenbanken und Objekten zu beachten sind.
- Anhang B, „Verwenden der automatischen Clientweiterleitung“, erläutert die automatische Weiterleitung von Clientanwendungen sowie die Aktivierung dieser Unterstützung.
- Anhang C, „Verwenden der LDAP-Verzeichnisservices (Lightweight Directory Access Protocol)“, enthält Informationen zur Verwendung der LDAP-Verzeichnisservices.
- Anhang D, „Absetzen von Befehlen an mehrere Datenbankpartitionen“, behandelt die Verwendung der Shellprozeduren *db2\_all* und *rah* zum Senden von Befehlen an alle Partitionen in Umgebungen mit partitionierten Datenbanken.
- Anhang E, „Verwenden der Unterstützung für Windows Management Instrumentation (WMI)“, enthält Informationen darüber, wie DB2 mit WMI verwaltet werden kann.
- Anhang F, „Verwenden der Windows NT-Sicherheit“, beschreibt, wie DB2 mit den Windows-Sicherheitsfunktionen arbeitet.
- Anhang G, „Verwenden des Windows-Systemmonitors“, beschreibt die Verwendung des Windows-Systemmonitors zur Erfassung von DB2-Leistungsdaten.
- Anhang H, „Verwenden von Windows-Datenbankpartitionsservern“, beschreibt die Dienstprogramme, die von Windows zur Arbeit mit Servern partitionierter Datenbanken verwendet werden.
- Anhang I, „Konfigurieren mehrerer logischer Knoten“, beschreibt, wie mehrere logische Knoten in einer partitionierten Datenbankumgebung konfiguriert werden können.
- Anhang J, „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.

Ein Kapitel mit dem Titel „Dienstprogramme für das Versetzen von Daten“ wurde aus dem Handbuch *Systemverwaltung: Implementierung* herausgenommen.

**Anmerkung:** Alle Informationen zu den DB2-Dienstprogrammen für das Versetzen von Daten sowie die vergleichbaren Themen aus den Handbüchern *Command Reference* und *Administrative API Reference* wurden in *Dienstprogramme für das Versetzen von Daten Handbuch und Referenz* zusammengefasst.

Informationen zu diesen Themen finden Sie ausschließlich in *Dienstprogramme für das Versetzen von Daten Handbuch und Referenz*.

Weitere Informationen zur Replikation von Daten finden Sie in *IBM DB2 Information Integrator SQL Replication Handbuch und Referenz*.

Ein Kapitel mit dem Titel „Wiederherstellen einer Datenbank“ wurde aus dem Handbuch *Systemverwaltung: Implementierung* herausgenommen.

**Anmerkung:** Alle Informationen über die Methoden und Tools zur Sicherung und Wiederherstellung von Daten sowie die vergleichbaren Themen aus den Handbüchern *Command Reference* und *Administrative API Reference* wurden in *Datenwiederherstellung und hohe Verfügbarkeit Handbuch und Referenz* zusammengefasst.

Informationen zu diesen Themen finden Sie ausschließlich in *Datenwiederherstellung und hohe Verfügbarkeit Handbuch und Referenz*.

---

## 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 sowie zu verteilten Transaktionen. Die einzelnen Kapitel und Anhänge des Bandes werden im Folgenden kurz vorgestellt:

#### Datenbankkonzepte

- Das Kapitel zu allgemeinen Konzepten relationaler Datenbanken enthält eine Übersicht über Datenbankobjekte, einschließlich Wiederherstellungsobjekten, Speicherobjekten und Systemobjekten.
- 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.

#### Datenbankentwurf

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

#### Anhänge

- Der Anhang über Inkompatibilitäten zwischen Releases stellt die Inkompatibilitäten dar, die von Version 7 und Version 8 eingeführt werden, und weist auf zukünftige Inkompatibilitäten hin, auf die geachtet werden sollte.
- Der Anhang zur Unterstützung von Landessprachen beschreibt die DB2-Unterstützung von Landessprachen und enthält Informationen zu Gebieten, Sprachen und Codepages.
- Der Anhang zum Aktivieren der Unterstützung großer Seiten in einer 64-Bit-Umgebung (AIX) erläutert die Unterstützung für die Seitengröße von 16 MB sowie die Aktivierung dieser Unterstützung.

## 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 "Einführung in die Optimierung" führt in die Konzepte und Überlegungen zur Verwaltung und Optimierung der Leistung von DB2 UDB ein.
- Das Kapitel "Architektur und Prozesse" stellt die zugrunde liegende 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 gibt 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 Umgebungen mit partitionierten Datenbanken bei der Neuverteilung von Daten auf die Partitionen zu beachten sind.
- Das Kapitel zu Vergleichstests (Benchmark-Tests) enthält eine Übersicht über Vergleichstests und behandelt verschiedene Aspekte ihrer Durchführung.
- Das Kapitel zur DB2-Konfiguration behandelt die Konfigurationsdateien für den Datenbankmanager und die Datenbanken sowie die Werte für die Konfigurationsparameter für den Datenbankmanager, die Datenbanken und den DB2-Verwaltungsserver (DAS).

### Anhänge

- Der Anhang zu DB2-Registriervariablen und DB2-Umgebungsvariablen beschreibt 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, und die Erstellung dieser Tabellen.

- Der Anhang zu den SQL-Explain-Programmen beschreibt die Verwendung der DB2-EXPLAIN-Programme: db2expln und dynexpln.
- Der Anhang zu db2exfmt - dem Formatierungstool für EXPLAIN-Tabellen - beschreibt, wie mit diesem DB2-EXPLAIN-Tool EXPLAIN-Tabellendaten formatiert werden.



---

## **Teil 1. Implementieren des Datenbankentwurfs**



---

## Kapitel 1. 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, Datenbankpartitionsgruppen, Tabellenbereiche, Tabellen, Sichten, Wrapper, 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 und mit Hilfe von SQL in Anwendungen erstellen.

Informationen zu SQL-Anweisungen finden Sie im Handbuch *SQL Reference*. Informationen zu Befehlen des Befehlszeilenprozessors finden Sie im Handbuch *Command Reference*. Informationen zu den Anwendungsprogrammierschnittstellen (Application Programming Interfaces - APIs) finden Sie im Handbuch *Administrative API Reference*.

Eine weitere Möglichkeit zur Erstellung von Datenbankobjekten besteht über die Steuerzentrale. Die Steuerzentrale kann anstelle der SQL-Anweisungen, der Befehle des Befehlszeilenprozessors oder der APIs verwendet werden.

Im vorliegenden Kapitel wird das Verfahren zur Durchführung von Tasks mit Hilfe der Steuerzentrale durch einen Rahmen hervorgehoben. Daran unmittelbar 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. Bei der Arbeit mit der Steuerzentrale finden Sie in der dortigen Hilfefunktion detailliertere Informationen als in der Übersicht des vorliegenden Handbuchs.

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 vorausgesetzt werden, 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. Genaue Informationen zu diesen Unterschieden finden Sie im Handbuch *Einstieg* sowie in der Dokumentation zum jeweiligen Betriebssystem.

Zum Beispiel unterstützt Windows eine Art von Anwendung, die als „Dienst“ (Service) bezeichnet wird. Unter DB2 für Windows wird jedes DB2-Exemplar als Dienst definiert. Ein Dienst kann automatisch beim Booten des Systems, von einem Benutzer über das Applet „Dienste“ der Systemsteuerung oder durch eine 32-Bit-Anwendung, die die Servicefunktionen der 32-Bit-Anwendungsprogrammierschnittstelle (API) von Microsoft Windows verwendet, gestartet werden. Dienste können ausgeführt werden, auch wenn kein Benutzer am System angemeldet ist.

Sofern nichts anderes speziell angegeben wird, bezieht sich die Bezeichnung Windows 9x auf Windows 98 und Windows ME. Die Bezeichnung Windows NT bezieht sich auf Windows NT, Windows 2000, Windows XP und Windows Server 2003. Die Bezeichnung Windows allein bezieht sich auf alle unterstützten Windows-Betriebssysteme.

---

## Arbeiten mit Exemplaren

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

- „Starten von DB2 UDB unter UNIX“
- „Starten von DB2 UDB unter Windows“ auf Seite 5
- „Mehrere Exemplare des Datenbankmanagers“ auf Seite 6
- „Gruppieren von Objekten nach Schema“ auf Seite 8
- „Parallelität“ auf Seite 9
- „Aktivieren der Datenpartitionierung in einer Datenbank“ auf Seite 13
- „Stoppen eines Exemplars unter UNIX“ auf Seite 15

## Starten von DB2 UDB unter UNIX

Das Starten und Stoppen von DB2 Universal Database™ (DB2 UDB) 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 Hostdatenbanken

### Voraussetzungen:

Gehen Sie wie folgt vor, um auf Ihrem System ein DB2 UDB-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 das Startscript wie folgt aus:

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

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

### Vorgehensweise:

Starten Sie das Exemplar mit einer der folgenden Methoden:

1. Gehen Sie wie folgt vor, um das Exemplar über die 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 das zu startenden Exemplar und wählen Sie im Kontextmenü **Starten** aus.

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

`db2start`

### Zugehörige Tasks:

- „Stoppen eines Exemplars unter UNIX“ auf Seite 15
- „Festlegen des aktuellen Exemplars“ auf Seite 27
- „Starten von DB2 UDB unter Windows“ auf Seite 5

## Starten von DB2 UDB unter Windows

Das Starten und Stoppen von DB2 Universal Database™ (DB2 UDB) 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 Hostdatenbanken

### Voraussetzungen:

Damit DB2 UDB 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 Windows-Dienstes definiert wird. Das Benutzerkonto muss ein Mitglied der Gruppe der Administratoren, der Serveroperatoren oder der Hauptbenutzer sein.

### Vorgehensweise:

Starten Sie das Exemplar mit einer der folgenden Methoden:

1. Gehen Sie wie folgt vor, um das Exemplar über die 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 das zu startenden Exemplar und wählen Sie im Kontextmenü **Starten** aus.

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

`db2start`

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

Bei Ausführung in einer partitionierten Datenbankumgebung wird jeder Datenbankpartitionsserver als Windows-Dienst gestartet. In einer partitionierten Datenbankumgebung können Sie den Schalter "/D" nicht verwenden, um DB2 als Prozess zu starten.

#### Zugehörige Tasks:

- „Starten von DB2 UDB unter UNIX“ auf Seite 4
- „Stoppen eines Exemplars unter UNIX“ auf Seite 15
- „Stoppen eines Exemplars unter Windows“ auf Seite 16

## Mehrere 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 von Umgebungen.

Mehrere Exemplare können z. B. zum Erstellen von Umgebungen zu folgenden Zwecken wünschenswert sein:

- Trennen der Entwicklungsumgebung von der Geschäftsumgebung.
- Getrenntes Optimieren für spezielle Anwendungen, die ausgeführt werden sollen.
- Schützen sensibler 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.

**Anmerkung:** (Nur für UNIX<sup>®</sup>-Betriebssysteme:) Zur Vermeidung von Umgebungskonflikten zwischen zwei oder mehr Exemplaren sollten Sie sicherstellen, dass jedes Exemplar über ein eigenes Ausgangsdateisystem verfügt. Eine gemeinsame Nutzung der Ausgangsdateisysteme führt zu Fehlern.

Die Programmdateien von DB2<sup>®</sup> Universal Database (DB2 UDB) 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 herzustellen ist.

#### Zugehörige Konzepte:

- „Mehrere Exemplare unter einem UNIX-Betriebssystem“ auf Seite 21
- „Mehrere Exemplare unter einem Windows-Betriebssystem“ auf Seite 22

#### Zugehörige Tasks:

- „Erstellen zusätzlicher Exemplare“ auf Seite 23

#### Zugehörige Referenzen:

- „ATTACH Command“ in *Command Reference*

## Herstellen der Verbindung zu einem anderen Exemplar des Datenbankmanagers

Zur Herstellung der Verbindung mit einem anderen Exemplar, das auch ein fernes Exemplar sein kann, verwenden Sie den Befehl **ATTACH**.

#### Voraussetzungen:

Es muss bereits mehr als ein Exemplar vorhanden sein.

#### Vorgehensweise:

Gehen Sie wie folgt vor, um eine Verbindung zu einem anderen Exemplar des Datenbankmanagers über die Steuerzentrale herzustellen:

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

Geben Sie in der Befehlszeile Folgendes ein, um eine Verbindung zu einem Exemplar herzustellen:

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

Um zum Beispiel eine Verbindung zu einem Exemplar mit dem Namen testdb2 herzustellen, das zuvor im Knotenverzeichnis katalogisiert wurde, müssen Sie folgenden Befehl eingeben:

```
db2 attach to testdb2
```

Nach Beendigung der Wartungsaktivitäten für das Exemplar testdb2 kann die Verbindung zu diesem Exemplar durch Ausführen des Befehls **DETACH** wie folgt getrennt werden:

```
db2 detach
```

#### Zugehörige Referenzen:

- „ATTACH Command“ in *Command Reference*
- „DETACH Command“ in *Command Reference*

## 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 derart qualifizierten Objekts stellt eine Methode bereit, 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.

Eine explizite Verwendung des Schemas liegt vor, 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)
```

Eine implizite Verwendung des Schemas liegt vor, wenn Sie die höherwertige Komponente eines zweiteiligen Objektnamens nicht verwenden. Wenn dies der Fall ist, wird anhand des Sonderregisters CURRENT SCHEMA der Schemaname ermittelt, der als höherwertige Komponente des Objektnamens ergänzt wird. 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 Schemanamen einstellen.

Bei der Erstellung der Datenbank werden einige Objekte innerhalb bestimmter Schemata erstellt und in den Systemkatalogtabellen gespeichert.

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 Schemanamens sehr vorteilhaft sein.

### Zugehörige Konzepte:

- „Definition von Systemkatalogtabellen“ auf Seite 73

### Zugehörige Tasks:

- „Erstellen eines Schemas“ auf Seite 92

### Zugehörige Referenzen:

- „SET SCHEMA statement“ in *SQL Reference, Volume 2*
- „CURRENT SCHEMA special register“ in *SQL Reference, Volume 1*



## Parallelität

Zur Nutzung der Parallelität innerhalb einer Datenbankpartition oder einer nicht partitionierten Datenbank müssen Sie die Konfigurationsparameter ändern. Durch die partitionsinterne Parallelität können Sie zum Beispiel mehrere Prozessoren einer SMP-Maschine (SMP = Symmetric Multiprocessor) nutzen.

### Aktivieren der partitionsübergreifenden Abfrageparallelität

#### Vorgehensweise:

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

#### Zugehörige Konzepte:

- „Partitions- und Prozessorumgebungen“ in *Systemverwaltung: Konzept*
- „Datenpartitionierung“ in *Systemverwaltung: Konzept*
- „Aufbau von Datenbankpartitionsgruppen“ in *Systemverwaltung: Konzept*
- „Partitionen in einer partitionierten Datenbank“ in *Systemverwaltung: Optimierung*

#### Zugehörige Tasks:

- „Aktivieren der partitionsinternen Parallelität für Abfragen“ auf Seite 9
- „Aktivieren der Datenpartitionierung in einer Datenbank“ auf Seite 13
- „Umverteilen von Daten auf Partitionen“ in *Systemverwaltung: Optimierung*

### Aktivieren der partitionsinternen Parallelität für Abfragen

#### Vorgehensweise:

Die Steuerzentrale kann zum Ermitteln oder Ändern der Werte für individuelle Einträge in einer bestimmten 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** und **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*.

Zur Einrichtung der partitionsinternen Abfrageparallelität müssen Sie einen oder mehrere Konfigurationsparameter der Datenbank und des Datenbankmanagers, Vorkompilierungs- und Bindeoptionen oder ein Sonderregister ändern.

### *intra\_parallel*

Dieser Konfigurationsparameter des Datenbankmanagers gibt an, ob der Datenbankmanager partitionsinterne Parallelität verwenden kann. Der Standardwert gibt an, dass die partitionsinterne Parallelität nicht verwendet werden kann.

### *max\_querydegree*

Dieser Konfigurationsparameter des Datenbankmanagers gibt den maximalen Grad partitionsinterner Parallelität an, der für SQL-Anweisungen verwendet wird, die auf diesem Exemplar ausgeführt werden. Eine SQL-Anweisung führt nicht mehr als diese Anzahl paralleler Operationen innerhalb einer Partition aus. Der Konfigurationsparameter *intra\_parallel* muss zudem auf den Wert „YES“ gesetzt sein, damit der Wert des Parameters *max\_querydegree* verwendet wird. Der Standardwert für diesen Konfigurationsparameter ist -1. Dieser Wert bedeutet, dass das System den vom Optimierungsprogramm festgelegten Parallelitätsgrad verwendet. Andernfalls wird der vom Benutzer angegebene Wert verwendet.

### *dft\_degree*

Konfigurationsparameter der Datenbank. Definiert den Standardwert für die Bindeoption DEGREE und das Sonderregister CURRENT DEGREE. Der Standardwert ist 1. Der Wert ANY bedeutet, dass das System den vom Optimierungsprogramm festgelegten Parallelitätsgrad verwendet.

### **DEGREE**

Option für den Vorcompiler bzw. das Binden von statischem SQL.

### **CURRENT DEGREE**

Sonderregister für dynamisches SQL.

### **Zugehörige Konzepte:**

- „Parallelverarbeitung für Anwendungen“ in *Systemverwaltung: Optimierung*
- „Informationen zur Parallelverarbeitung“ in *Systemverwaltung: Optimierung*

### **Zugehörige Tasks:**

- „Konfigurieren von DB2 mit Konfigurationsparametern“ in *Systemverwaltung: Optimierung*

### **Zugehörige Referenzen:**

- „max\_querydegree - Maximaler Grad der Parallelität bei Abfragen“ in *Systemverwaltung: Optimierung*
- „intra\_parallel - Partitionsinterne Parallelität aktivieren“ in *Systemverwaltung: Optimierung*
- „dft\_degree - Grad der Parallelität“ in *Systemverwaltung: Optimierung*
- „BIND Command“ in *Command Reference*
- „PRECOMPILE Command“ in *Command Reference*
- „CURRENT DEGREE special register“ in *SQL Reference, Volume 1*

## Aktivieren der partitionsinternen Parallelität für Dienstprogramme

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

- LOAD
- CREATE INDEX
- BACKUP DATABASE oder TABLESPACE
- RESTORE DATABASE oder TABLESPACE

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

**Aktivieren der Parallelität zum Laden von Daten:** Das Dienstprogramm LOAD nutzt die Parallelität automatisch, und Sie können die folgenden Parameter im Befehl **LOAD** verwenden:

- CPU\_PARALLELISM
- DISK\_PARALLELISM

In einer Umgebung mit partitionierten Datenbanken erfolgt die partitionsübergreifende Parallelität zum Laden von Daten automatisch, wenn die Zieltabelle für mehrere Partitionen definiert ist. Die partitionsübergreifende Parallelität zum Laden von Daten kann durch die Angabe OUTPUT\_DBPARTNUMBS überschrieben werden. Das Dienstprogramm LOAD macht außerdem intelligenten Gebrauch von der Parallelität durch die Datenpartitionierung in Abhängigkeit von der Größe der Zielpartitionen. Der Parameter MAX\_NUM\_PART\_AGENTS kann zur Steuerung des maximalen Grads der Parallelität verwendet werden, der von dem Dienstprogramm LOAD ausgewählt wird. Die Parallelität durch Datenpartitionierung kann durch Angeben von PARTITIONING\_DBPARTNUMS überschrieben werden, wenn außerdem ANYORDER angegeben wird.

### Zugehörige Konzepte:

- „LOAD - Übersicht“ in *Dienstprogramme für das Versetzen von Daten Handbuch und Referenz*
- „Übersicht über das Laden in partitionierten Datenbanken“ in *Dienstprogramme für das Versetzen von Daten Handbuch und Referenz*

**Aktivieren der Parallelität beim Erstellen von Indizes:** 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.

### Zugehörige Referenzen:

- „intra\_parallel - Partitionsinterne Parallelität aktivieren“ in *Systemverwaltung: Optimierung*
- „CREATE INDEX statement“ in *SQL Reference, Volume 2*

**Aktivieren der E/A-Parallelität zum Sichern einer Datenbank oder eines Tabellenbereichs:** 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, indem Sie mehrere Behälter definieren, oder verwenden Sie einen einzigen Behälter mit mehreren Platten, und verwenden Sie die Registriervariable `DB2_PARALLEL_IO` entsprechend. Wenn Sie eine parallele E/A nutzen wollen, müssen Sie bereits vor dem Definieren von Behältern die erforderlichen Vorbedingungen berücksichtigen und entsprechend vorgehen. Diese Vorbedingungen können nicht erst bei Erkennung eines Bedarfs erfüllt werden, sondern müssen eingeplant werden, bevor Sie den Punkt erreichen, an dem Sie Ihre Datenbank oder Ihren Tabellenbereich sichern müssen.
- Geben Sie mit dem Parameter `PARALLELISM` im Befehl **BACKUP** den Grad der Parallelität an.
- Verwenden Sie den Parameter `WITH anzahl-puffer BUFFERS` im Befehl **BACKUP**, um sicherzustellen, dass genügend Puffer für den Grad der Parallelität verfügbar sind. 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 (`EXTENTSIZE * Anzahl der Behälter`) der zu sichernden Tabellenbereiche.

#### **Zugehörige Referenzen:**

- „BACKUP DATABASE Command“ in *Command Reference*

**Aktivieren der E/A-Parallelität zum Wiederherstellen einer Datenbank oder eines Tabellenbereichs:** 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 Quelldatenträger.
- Konfigurieren Sie Tabellenbereiche für parallele E/A. Sie müssen die Entscheidung für die Verwendung dieser Option treffen, bevor Sie Ihre Behälter definieren. Die entsprechenden Vorbedingungen können nicht erst bei Erkennung eines Bedarfs erfüllt werden, sondern müssen eingeplant werden, bevor Sie den Punkt erreichen, an dem Sie Ihre Datenbank oder Ihren Tabellenbereich wiederherstellen müssen.
- Geben Sie mit dem Parameter `PARALLELISM` im Befehl **RESTORE** den Grad der Parallelität an.
- Verwenden Sie den Parameter `WITH anzahl-puffer BUFFERS`, um sicherzustellen, dass genügend Puffer für den Grad der Parallelität verfügbar sind. 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.

#### Zugehörige Referenzen:

- „RESTORE DATABASE Command“ in *Command Reference*

### Aktivieren der Datenpartitionierung in einer Datenbank

Die Entscheidung, eine Datenbank in einer partitionierten Umgebung zu nutzen, muss vor der Erstellung der betreffenden Datenbank getroffen werden. Im Rahmen Ihrer Entscheidungen beim Datenbankentwurf müssen Sie festlegen, ob die Leistungsverbesserungen durch eine Partitionierung der Datenbank genutzt werden sollen.

Einige der für die Entscheidung zur Erstellung einer partitionierten Datenbank relevanten Gesichtspunkte werden in diesem Abschnitt behandelt.

#### Vorgehensweise:

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 dazu den Befehl **CREATE DATABASE** oder die Anwendungsprogrammierschnittstelle (API) `sqlcrea()`.

Vor der Erstellung einer partitionierten Datenbank müssen Sie die Datenbankpartition auswählen, die als Katalogknoten für die Datenbank fungieren soll. Anschließend können Sie die Datenbank direkt von dieser Partition oder von einem fernen Client aus erstellen, der mit dieser Partition verbunden ist. Die Datenbankpartition, zu der Sie die Verbindung (mit `ATTACH`) herstellen, um den Befehl **CREATE DATABASE** auszuführen, 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 zusammengesetzter Datenbanken (Wrapper, 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. Unter UNIX heißt das Systemdatenbankverzeichnis `sqldbdir` und befindet sich im Verzeichnis `sql1lib` unter Ihrem Ausgangsverzeichnis oder unter dem Verzeichnis, in dem DB2 Universal Database™ (DB2 UDB) installiert wurde. Dieses Verzeichnis muss sich unter UNIX 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. Unter Windows befindet sich das Systemdatenbankverzeichnis im Exemplarverzeichnis.

Ebenfalls im Verzeichnis `sqldbdir` 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 gespeichert 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`, `fcu_num_anchors`, `fcu_num_buffers`, `fcu_num_connect`, `fcu_num_rqb`, `max_connretries`, `max_coordagents`, `max_time_diff`, `num_poolagents` und `stop_start_time`.

#### **Zugehörige Tasks:**

- „Konfigurieren von DB2 mit Konfigurationsparametern“ in *Systemverwaltung: Optimierung*

#### **Zugehörige Referenzen:**

- „sqlcrea - Create Database“ in *Administrative API Reference*
- „CREATE DATABASE Command“ in *Command Reference*

## Stoppen eines Exemplars unter UNIX

Es ist möglich, dass Sie das aktuelle Exemplar des Datenbankmanagers stoppen müssen.

### Voraussetzungen:

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

1. Melden Sie sich mit einer Benutzer-ID oder einem Benutzernamen, 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 sich eine Liste der Anwendungen anzeigen, um sicherzugehen, dass keine wichtigen oder kritischen Anwendungen aktiv sind. Dazu 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.

### Einschränkungen:

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

**Anmerkung:** Wenn Sitzungen des Befehlszeilenprozessors mit einem Exemplar verbunden 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.

### Vorgehensweise:

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

1. Gehen Sie wie folgt vor, um das Exemplar über die Steuerzentrale zu stoppen:

- |   |
|---|
| <ol style="list-style-type: none"><li>1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner <b>Exemplare</b> angezeigt wird.</li><li>2. Klicken Sie auf alle Exemplare, die gestoppt werden sollen.</li><li>3. Klicken Sie mit der rechten Maustaste auf eines der ausgewählten Exemplare und wählen Sie im Kontextmenü <b>Stoppen</b> aus.</li><li>4. Klicken Sie im Fenster <b>Stoppen bestätigen</b> auf <b>OK</b>.</li></ol> |
|---|

2. Geben Sie in der Befehlszeile Folgendes ein, um das Exemplar zu stoppen:  
`db2stop`

Sie können den Befehl `db2stop` zum Stoppen oder Löschen einzelner Partitionen innerhalb einer partitionierten Datenbankumgebung verwenden. Wenn Sie in einer partitionierten Datenbank versuchen, eine logische Partition mit dem folgenden Befehl zu löschen:

```
db2stop drop nodenum <0>
```

müssen Sie sicherstellen, dass momentan keine Benutzer versuchen, auf die Datenbank zuzugreifen. Wenn dies dennoch der Fall ist, empfangen Sie eine Fehlermeldung SQL6030N.

#### Zugehörige Referenzen:

- „`db2stop` - Stop DB2 Command“ in *Command Reference*
- „`TERMINATE` Command“ in *Command Reference*

## Stoppen eines Exemplars unter Windows

Es ist möglich, dass Sie das aktuelle Exemplar des Datenbankmanagers stoppen müssen.

#### Voraussetzungen:

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

1. Das Benutzerkonto, das den Service von DB2 Universal Database™ (DB2 UDB) stoppt, muss über das richtige im Windows-Betriebssystem definierte Zugriffsrecht verfügen. Das Benutzerkonto muss ein Mitglied der Gruppe der Administratoren, der Serveroperatoren oder der Hauptbenutzer sein.
2. Zeigen Sie alle Anwendungen und Benutzer an, die mit der Datenbank, die Sie stoppen wollen, verbunden sind. Lassen Sie sich eine Liste der Anwendungen anzeigen, um sicherzugehen, dass keine wichtigen oder kritischen Anwendungen aktiv sind. Dazu 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`.

#### Einschränkungen:

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

**Anmerkung:** Wenn Sitzungen des Befehlszeilenprozessors mit einem Exemplar verbunden 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.



### Vorgehensweise:

Stoppen Sie ein Exemplar auf Ihrem System mit einer der folgenden Methoden:

- **db2stop**
- Stoppen des Service über die Steuerzentrale

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Exemplare** angezeigt wird.
2. Klicken Sie auf alle Exemplare, die gestoppt werden sollen.
3. Klicken Sie mit der rechten Maustaste auf eines der ausgewählten Exemplare und wählen Sie im Kontextmenü **Stoppen** aus.
4. Klicken Sie im Fenster **Stoppen bestätigen** auf **OK**.

- Stoppen über den Befehl „NET STOP“
- Stoppen des Exemplars aus einer Anwendung heraus

Berücksichtigen Sie, dass bei der Verwendung von DB2 UDB in einer partitionierten Datenbankumgebung jeder Datenbankpartitionsserver als Service (Dienst) gestartet wird. Jeder dieser Services muss gestoppt werden.

### Zugehörige Referenzen:

- „db2stop - Stop DB2 Command“ in *Command Reference*

---

## Vorbereiten der Erstellung einer Datenbank

Vor der Erstellung einer Datenbank gibt es zahlreiche Konzepte und Tasks, die Sie im Rahmen der erforderlichen Vorarbeiten zu diesem Zweck berücksichtigen sollten. Zu diesen Konzepten und Tasks gehören das Entwerfen der Datenbank sowie die Einrichtung des Exemplars, der Verzeichnisse und der anderen Unterstützungsdateien, die zur Arbeit mit einer Datenbank benötigt werden. Im Folgenden werden die folgenden Themen behandelt:

- Entwerfen logischer und physischer Datenbankmerkmale
- Exemplarerstellung
- Umgebungsvariablen und die Profilregistrierdatenbank
- DB2-Verwaltungsserver
- Erstellen einer Knotenkonfigurationsdatei
- Erstellen der Datenbankkonfigurationsdatei
- 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*.

## Exemplarerstellung

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

- Betreiben eines Exemplars für die Entwicklungsumgebung und eines weiteren für die Geschäftsumgebung.
- Anpassen eines Exemplars für eine bestimmte Umgebung.
- Begrenzen des Zugriffs auf sensible Daten.
- Steuern der Erteilung der Berechtigung SYSADM, SYSCTRL und SYSMANT für jedes Exemplar.
- Optimieren der Datenbankmanagerkonfiguration für jedes Exemplar.
- Begrenzen der durch den Ausfall eines Exemplars entstehenden Auswirkungen. Im Fall eines Ausfalls ist jeweils nur ein Exemplar betroffen. Die übrigen Exemplare funktionieren weiterhin ordnungsgemäß.

Mehrere Exemplare bringen allerdings auch einige kleinere Nachteile mit sich:

- 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 folgende Objekte:

- Die Konfigurationsdatei des Datenbankmanagers
- Das Systemdatenbankverzeichnis
- Das Knotenverzeichnis
- Die Knotenkonfigurationsdatei (`db2nodes.cfg`)
- Alle weiteren Dateien mit Debuginformationen, wie z. B. den Speicherauszug der Ausnahme- und Registrierdaten oder den Aufrufstapel für die Prozesse von DB2<sup>®</sup> Universal Database (DB2 UDB).

Unter UNIX<sup>®</sup>-Betriebssystemen befindet sich das Exemplarverzeichnis im Verzeichnis `INSTHOME/sql1ib`, wobei `INSTHOME` das Ausgangsverzeichnis des Exemplareigners ist.

Unter Windows<sup>®</sup>-Betriebssystemen befindet sich das Exemplarverzeichnis im Unterverzeichnis `/sql1ib` des Verzeichnisses, in dem DB2 UDB installiert wurde.

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.

Im Rahmen Ihres Installationsverfahrens erstellen Sie ein Anfangsexemplar von DB2 UDB 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, werden bei der Installation folgende Konfigurationswerte festgelegt:

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

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

Unter Windows ist der Exemplarname mit dem Namen des Service (Dienstes) identisch, so dass keine Konflikte auftreten sollten. Sie müssen über die richtige Berechtigung verfügen, um einen Service zu erstellen.

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

Vor der Verwendung von DB2 UDB muss die Datenbankumgebung für jeden Benutzer aktualisiert werden, so dass sie auf ein Exemplar zugreifen und die DB2 UDB-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 die Bourne- oder Korn-Shell und db2cshrc für die C-Shell. Diese Prozeduren befinden sich im Unterverzeichnis sql1ib im Ausgangsverzeichnis 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 alternativ 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 im Unterverzeichnis sql1ib im Ausgangsverzeichnis des Exemplareigners
- Setzen der Umgebungsvariablen DB2INSTANCE auf den Exemplarnamen

#### **Zugehörige Konzepte:**

- „Mehrere Exemplare unter einem UNIX-Betriebssystem“ auf Seite 21
- „Mehrere Exemplare unter einem Windows-Betriebssystem“ auf Seite 22

#### **Zugehörige Tasks:**

- „Hinzufügen eines Exemplars“ auf Seite 26
- „UNIX - Details beim Erstellen von Exemplaren“ auf Seite 24
- „Windows - Details beim Erstellen von Exemplaren“ auf Seite 25
- „Festlegen des aktuellen Exemplars“ auf Seite 27
- „Automatisches Starten von Exemplaren“ auf Seite 28
- „Gleichzeitiges Ausführen mehrerer Exemplare“ auf Seite 29
- „Auflisten von Exemplaren“ auf Seite 27
- „Erstellen zusätzlicher Exemplare“ auf Seite 23

## Automatisches Einrichten der DB2 UDB-Umgebung unter UNIX

Standardmäßig wirken sich die Prozeduren, durch die die Datenbankumgebung eingerichtet wird, wenn Sie ein Exemplar erstellen, 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.

### Vorgehensweise:

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

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

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

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

- Fügen Sie für Benutzer, die eine angepasste Version der Prozedur in ihrem Ausgangsverzeichnis verwenden, folgende Anweisungen hinzu:

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

Dabei ist `USERHOME` das Ausgangsverzeichnis des Benutzers.

### Zugehörige Tasks:

- „Manuelles Einrichten der DB2 UDB-Umgebung unter UNIX“ auf Seite 20

## Manuelles Einrichten der DB2 UDB-Umgebung unter UNIX

### Vorgehensweise:

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, folgende Anweisungen hinzu:

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

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

- Fügen Sie für Benutzer, die eine angepasste Version der Prozedur in ihrem Ausgangsverzeichnis verwenden, folgende Anweisungen hinzu:

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

Dabei ist `USERHOME` das Ausgangsverzeichnis 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 Ausgangsverzeichnissen /u/test und /u/prod .

Fenster 1:

- Geben Sie in einer Bourne- oder Korn-Shell Folgendes ein:  
`. /u/test/sql1lib/db2profile`
- Geben Sie in einer C-Shell Folgendes ein:  
`source /u/test/sql1lib/db2cshrc`

Fenster 2:

- Geben Sie in einer Bourne- oder Korn-Shell Folgendes ein:  
`. /u/prod/sql1lib/db2profile`
- Geben Sie in einer C-Shell Folgendes ein:  
`source /u/prod/sql1lib/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 CLP-Datei zurück. Vergewissern Sie sich, dass sich diese Datei im Unterverzeichnis sql1lib des Exemplars befindet.

**Zugehörige Tasks:**

- „Automatisches Einrichten der DB2 UDB-Umgebung unter UNIX“ auf Seite 20

## Mehrere Exemplare unter einem UNIX-Betriebssystem

Es ist möglich, mehr als ein Exemplar unter einem UNIX<sup>®</sup>-Betriebssystem zu betreiben. Sie können jedoch nur jeweils innerhalb eines Exemplars von DB2<sup>®</sup> Universal Database (DB2 UDB) gleichzeitig arbeiten.

**Anmerkung:** Zur Vermeidung von Umgebungskonflikten zwischen zwei oder mehr Exemplaren sollten Sie sicherstellen, dass jedes Exemplar über ein eigenes Ausgangsdateisystem verfügt. Eine gemeinsame Nutzung der Ausgangsdateisysteme führt zur Rückgabe von Fehlern.

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 Ausgangsverzeichnis verfügen. Alle erforderlichen Dateien zum Ausführen des Exemplars werden im Ausgangsverzeichnis 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 UDB 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.

#### **Zugehörige Tasks:**

- „UNIX - Details beim Erstellen von Exemplaren“ auf Seite 24

## **Mehrere Exemplare unter einem Windows-Betriebssystem**

Es ist möglich, mehrere Exemplare von DB2<sup>®</sup> Universal Database (DB2 UDB) auf derselben Maschine zu betreiben. Jedes Exemplar von DB2 UDB enthält eigene Datenbanken und besitzt eigene Konfigurationsparameter für den Datenbankmanager.

Ein Exemplar von DB2 UDB besteht aus folgenden Komponenten:

- Ein Windows<sup>®</sup>-Service (Dienst), der das Exemplar darstellt. Der Name des Service ist mit dem Namen des Exemplars identisch. Der Anzeigename des Service (im Fenster „Dienste“) ist der Exemplarname, dem das Präfix „DB2 - “ vorangestellt ist. Zum Beispiel gibt es für ein Exemplar mit dem Namen DB2 einen Windows-Service mit dem Namen „DB2“, der als „DB2 - DB2“ angezeigt wird.

**Anmerkung:** Für Windows 98, Windows ME oder für Clientexemplare wird kein Windows-Service erstellt.

- Ein Exemplarverzeichnis. Dieses Verzeichnis enthält die Konfigurationsdateien des Datenbankmanagers, das Systemdatenbankverzeichnis, das Knotenverzeichnis, das DCS-Datenbankverzeichnis, alle Diagnoseprotokolldateien sowie alle Dateien, die dem Exemplar zugeordnet sind. Das Exemplarverzeichnis ist standardmäßig ein Unterverzeichnis im Verzeichnis SQLLIB und hat den gleichen Namen wie das Exemplar. Für das Exemplar „DB2“ ist dieses Verzeichnis zum Beispiel C:\SQLLIB\DB2, wobei C:\SQLLIB das Verzeichnis ist, in dem DB2 UDB installiert ist. Sie können mit Hilfe der Registriervariablen DB2INSTPROF die Standardposition des Exemplarverzeichnisses ändern. Wenn die Registriervariable DB2INSTPROF eine andere Position angibt, wird das Exemplarverzeichnis

unter dem Verzeichnis erstellt, auf das die Variable DB2INSTPROF zeigt. Wenn zum Beispiel DB2INSTPROF=D:\DB2PROFS definiert würde, wäre das Exemplarverzeichnis D:\DB2PROFS\DB2.

- Ein Registrierschlüssel unter HKEY\_LOCAL\_MACHINE\SOFTWARE\IBM\DB2\PROFILES\

Sie können mehrere DB2 UDB-Exemplare gleichzeitig ausführen. Zur Arbeit mit einem Exemplar müssen Sie die Umgebungsvariable DB2INSTANCE auf den Namen des gewünschten Exemplars setzen, bevor Sie Befehle für das Exemplar ausführen.

Damit ein Exemplar nicht auf die Datenbanken eines anderen Exemplars zugreift, werden die Datenbankdateien für ein Exemplar unter einem Verzeichnis erstellt, das den gleichen Namen wie das Exemplar besitzt. Wenn zum Beispiel eine Datenbank auf Laufwerk C: für das Exemplar DB2 erstellt wird, werden die Datenbankdateien in einem Verzeichnis mit dem Namen C:\DB2 erstellt. Analog werden bei der Erstellung einer Datenbank auf Laufwerk C: für das Exemplar TEST die Datenbankdateien in einem Verzeichnis mit dem Namen C:\TEST erstellt.

#### Zugehörige Konzepte:

- „Hohe Verfügbarkeit“ in *Datenwiederherstellung und hohe Verfügbarkeit Handbuch und Referenz*

#### Zugehörige Tasks:

- „Windows - Details beim Erstellen von Exemplaren“ auf Seite 25

## Erstellen zusätzlicher Exemplare

| Obwohl im Rahmen der Installation von DB2 Universal Database™ (DB2 UDB) ein  
| Exemplar erstellt wird, können Ihre Geschäftsanforderungen eine Erstellung  
| zusätzlicher Exemplare erforderlich machen.

#### Voraussetzungen:

Wenn Sie der Administratorengruppe unter Windows angehören oder die Rootberechtigung für UNIX-Plattformen besitzen, können Sie zusätzliche DB2 UDB-Exemplare hinzufügen. Die Maschine, auf der das Exemplar hinzugefügt wird, wird als Exemplareignermaschine (Knoten 0) definiert. Fügen Sie Exemplare unbedingt auf einer Maschine hinzu, auf der sich ein Verwaltungsserver befindet.

#### Vorgehensweise:

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

```
db2icrt <exemplarname>
```

Bei der Verwendung des Befehls **db2icrt** zum Hinzufügen eines zusätzlichen DB2 UDB-Exemplars sollten Sie den Anmeldenamen des Exemplareigners und optional den Authentifizierungstyp des Exemplars angeben. Der Authentifizierungstyp gilt für alle Datenbanken, die unter diesem Exemplar erstellt werden. Der Authentifizierungstyp ist eine Anweisung, wo die Authentifizierung von Benutzern stattfinden soll.

Mit der Umgebungsvariablen DB2INSTPROF können Sie die Position des in DB2PATH angegebenen Exemplarverzeichnisses ändern. Sie benötigen dafür

Schreibzugriff auf das Exemplarverzeichnis. Wenn die Verzeichnisse in einem anderen Pfad als DB2PATH erstellt werden sollen, müssen Sie die Variable DB2INSTPROF definieren, *bevor* Sie den Befehl **db2icrt** eingeben.

Für DB2 Universal Database Enterprise - Server Edition (ESE) müssen Sie außerdem deklarieren, dass Sie ein neues Exemplar hinzufügen, das ein partitioniertes Datenbanksystem ist. Wenn Sie darüber hinaus mit einem ESE-Exemplar arbeiten, das mehr als eine Partition hat und Fast Communication Manager (FCM) verwenden, können Sie mehrere Verbindungen zwischen Partitionen haben, indem Sie bei der Erstellung des Exemplars mehrere TCP/IP-Ports definieren. Für Windows-Betriebssysteme verwenden Sie dazu zum Beispiel den Befehl **db2icrt** mit dem Parameter **-r <portbereich>**. Der Portbereich wird wie folgt angegeben:

```
-r:<anfangsport, endport>
```

Dabei ist 'anfangsport' der erste Port und 'endport' der letzte Port in einem Bereich von Portnummern, die von FCM verwendet werden können.

#### Zugehörige Konzepte:

- „Authentifizierungsmethoden für den Server“ auf Seite 234
- „Authentifizierungsaspekte für ferne Clients“ auf Seite 240

#### Zugehörige Referenzen:

- „db2icrt - Create Instance Command“ in *Command Reference*

## UNIX - Details beim Erstellen von Exemplaren

Auf UNIX-Betriebssystemen verfügt der Befehl **db2icrt** über die folgenden optionalen Parameter:

- **-h** oder **-?**

Dieser Parameter zeigt ein Hilfemenü für den Befehl an.

- **-d**

Dieser Parameter konfiguriert den Debugmodus für die Fehlerbestimmung.

- **-a AuthTyp**

Dieser Parameter gibt den Authentifizierungstyp für das Exemplar an. Gültige Authentifizierungstypen sind **SERVER**, **SERVER\_ENCRYPT** und **CLIENT**. Wenn dieser Parameter nicht angegeben wird, wird standardmäßig der Authentifizierungstyp **SERVER** verwendet, wenn ein DB2 Universal Database™-Server (DB2 UDB-Server) installiert ist. Ansonsten wird die Option **CLIENT** verwendet.

#### Anmerkungen:

1. Der für das Exemplar angegebene Authentifizierungstyp gilt für alle Datenbanken, die zu diesem Exemplar gehören.
2. Für UNIX-Betriebssysteme ist der Authentifizierungstyp **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 UDB-Client oder den DB2 UDB Application Development Client installieren. Für andere DB2 UDB-Produkte ist dieser Parameter erforderlich.

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



- `-p PortName`  
Dieser Parameter gibt an, welcher TCP/IP-Servicename bzw. welche TCP/IP-Portnummer verwendet werden soll. Dieser Wert wird anschließend in der Datenbankkonfigurationsdatei für jede Datenbank des Exemplars definiert.
- `-s ExemplarTyp`  
Ermöglicht das Erstellen verschiedener Exemplartypen. Gültige Exemplartypen sind: `ese`, `wse`, `client` und `standalone`.

Beispiele:

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

```
db2icrt -u db2fenc1 db2inst1
```

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

```
db2icrt -u db2inst1 db2inst1
```

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

```
db2icrt db2inst1 -s client -u abgeschirmtID
```

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

#### Zugehörige Referenzen:

- „db2icrt - Create Instance Command“ in *Command Reference*

## Windows - Details beim Erstellen von Exemplaren

Auf Windows-Betriebssystemen verfügt der Befehl **db2icrt** über die folgenden optionalen Parameter:

- `-s ExemplarTyp`  
Ermöglicht das Erstellen verschiedener Exemplartypen. Gültige Exemplartypen sind: `ese`, `wse`, `client` und `standalone`.
- `-p:ExemplarProf_Pfad`  
Dies ist ein optionaler 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 Benutzer 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 Kontonamen in der Form `Domäne/Benutzer` sowie das Kennwort des DB2 Universal Database-Services deklarieren.

- `-r:Anfangs_Port,End_Port`

Dies ist ein optionaler Parameter zur Angabe des TCP/IP-Portbereichs für den FCM (Fast Communications Manager). Bei der Angabe des TCP/IP-Portbereichs müssen Sie sicherstellen, dass der Portbereich auf allen Maschinen des partitionierten Datenbanksystems verfügbar ist.

Das folgende Beispiel kann unter DB2 Universal Database (DB2 UDB) Enterprise Server Edition für Windows verwendet werden:

```
db2icrt inst1 -s ese
-p:\\maschineA\db2mpp
-u:<benutzerkontoname>,<kennwort> -r:9010,9015
```

**Anmerkung:** Wenn Sie das Servicekonto (Dienstkonto) ändern, d. h., wenn Sie den bei der Erstellung des ersten Exemplars im Rahmen der Installation des Produkts erstellten Standardservice ändern, müssen Sie dem Kontonamen Domäne/Benutzer, der zur Erstellung des Exemplars verwendet wird, die folgenden erweiterten Zugriffsberechtigungen erteilen:

- Verarbeiten von Prozessen als Teil des Betriebssystems
- Erstellen eines Tokenobjekts
- Anheben von Quoten
- Anmelden als Service (Dienst)
- Ersetzen eines Tokens auf Prozessebene
- Sperren einer Seite im Speicher

Das Exemplar benötigt diese Benutzerberechtigungen zum Zugreifen auf das gemeinsame Laufwerk, zur Authentifizierung des Benutzerkontos sowie zum Ausführen von DB2 UDB als Windows-Dienst. Die Berechtigung zum „Sperren einer Seite im Speicher“ wird für die AWE-Unterstützung (AWE = Address Windowing Extensions) benötigt.

**Zugehörige Referenzen:**

- „db2icrt - Create Instance Command“ in *Command Reference*

## Hinzufügen eines Exemplars

**Vorgehensweise:**

Nach der Erstellung eines zusätzlichen Exemplars müssen Sie einen Datensatz über dieses Exemplar innerhalb der Steuerzentrale hinzufügen, um mit diesem Exemplar über die Steuerzentrale arbeiten zu können.

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, wenn Sie die Steuerzentrale verwenden:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Exemplare** des gewünschten Systems angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf den Exemplarordner und wählen Sie im Kontextmenü **Hinzufügen** aus.
3. Geben Sie die erforderlichen Informationen ein und klicken Sie auf **Anwenden**.

**Zugehörige Konzepte:**

- „Exemplarerstellung“ auf Seite 18

**Zugehörige Tasks:**

- „Auflisten von Exemplaren“ auf Seite 27

## Auflisten von Exemplaren

**Vorgehensweise:**

Gehen Sie wie folgt vor, um eine Liste aller auf dem System verfügbaren Exemplare über die Steuerzentrale abzurufen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Exemplare** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf den Exemplarordner, und wählen Sie im Kontextmenü **Hinzufügen** aus.
3. Klicken Sie im Fenster **Exemplar 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 eine Liste aller Exemplare anzuzeigen, die auf dem System verfügbar sind:

```
db2ilist
```

Geben Sie Folgendes ein, um festzustellen, welches Exemplar für die aktuelle Sitzung (auf unterstützten Windows-Plattformen) gilt:

```
set db2instance
```

**Zugehörige Referenzen:**

- „db2ilist - List Instances Command“ in *Command Reference*

## Festlegen des aktuellen Exemplars

**Vorgehensweise:**

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

- Wenn die Umgebungsvariable DB2INSTANCE für die aktuelle Sitzung definiert ist, gibt ihr Wert das aktuelle Exemplar an.

Geben Sie Folgendes ein, um die Umgebungsvariable DB2INSTANCE zu definieren:

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

- Wenn die Umgebungsvariable DB2INSTANCE für die aktuelle Sitzung nicht definiert ist, verwendet DB2 UDB die Einstellung der Umgebungsvariablen DB2INSTANCE aus den Systemumgebungsvariablen. Unter Windows NT werden die Systemumgebungsvariablen auf der Registerkarte **Umgebung** im Applet **System** Systemsteuerung definiert. Unter Windows 9x werden sie in der Datei `autoexec.bat` definiert.

- Wenn die Umgebungsvariable DB2INSTANCE überhaupt nicht definiert ist, verwendet DB2 UDB die Variable DB2INSTDEF der Registrierdatenbank.

Geben Sie folgenden Befehl ein, um die Variable DB2INSTDEF der Registrierdatenbank auf der globalen Ebene der Registrierdatenbank zu definieren:

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

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

```
db2 get instance
```

#### Zugehörige Tasks:

- „Deklarieren von Registrier- und Umgebungsvariablen“ auf Seite 33

## Automatisches Starten von Exemplaren

#### Vorgehensweise:

Auf Windows-Betriebssystemen wird standardmäßig das bei der Installation definierte DB2 Universal Database™-Exemplar (DB2 UDB-Exemplar) auf automatische Startart eingestellt. Ein Exemplar, das mit Hilfe des Befehls **db2icrt** erstellt wird, wird auf manuelle Startart eingestellt. Zum Ändern der Startart müssen Sie dieses Merkmal des DB2 UDB-Service im Fenster **Dienste** ändern.

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

```
db2iauto -on <exemplarname>
```

Dabei ist <exemplarname> der Anmeldename des Exemplars.

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

```
db2iauto -off <exemplarname>
```

Dabei ist <exemplarname> der Anmeldename des Exemplars.

#### Zugehörige Konzepte:

- „Exemplarerstellung“ auf Seite 18

#### Zugehörige Referenzen:

- „db2iauto - Auto-start Instance Command“ in *Command Reference*

## Gleichzeitiges Ausführen mehrerer Exemplare

### Vorgehensweise:

Gehen Sie wie folgt vor, um über die 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 ein Exemplar und wählen Sie im Kontextmenü **Starten** aus.
3. Wiederholen Sie Schritt 2 so oft, bis alle Exemplare gestartet sind, die gleichzeitig ausgeführt werden sollen.

(Nur unter Windows:) Führen Sie folgende Schritte über die Befehlszeile aus, um mehrere Exemplare gleichzeitig auszuführen:

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

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

### Zugehörige Konzepte:

- „Mehrere Exemplare des Datenbankmanagers“ auf Seite 6

### Zugehörige Tasks:

- „UNIX - Details beim Erstellen von Exemplaren“ auf Seite 24
- „Windows - Details beim Erstellen von Exemplaren“ auf Seite 25
- „Erstellen zusätzlicher Exemplare“ auf Seite 23

## Lizenzverwaltung

Die Verwaltung der Lizenzen für Ihre DB2® Universal Database-Produkte (DB2 UDB) erfolgt im Wesentlichen über die Lizenzzentrale, die Bestandteil der Steuerzentrale in der Onlineschnittstelle des jeweiligen Produkts ist. In der Lizenzzentrale können Sie Lizenzinformationen, Statistiken, registrierte Benutzer und aktuelle Benutzer für jedes installierte Produkt einsehen.

Wenn die Steuerzentrale nicht verfügbar ist, können Sie grundlegende Lizenzfunktionen mit Hilfe des Befehls **db2licm** des Tools zur Lizenzverwaltung ausführen. Mit diesem Befehl können Sie auf Ihrem lokalen System installierte Lizenzen und Richtlinien hinzufügen, entfernen, auflisten und modifizieren.

### Zugehörige Referenzen:

- „db2licm - License Management Tool Command“ in *Command Reference*

## Umgebungsvariablen und die Profilregistrierdatenbank

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

Sie können den Konfigurationsassistenten (**db2ca**) zur Konfiguration von Konfigurationsparametern und Registriervariablen verwenden.

Vor der Einführung der DB2<sup>®</sup> Universal Database-Profilregistrierdatenbank (DB2 UDB-Profilregistrierdatenbank) war die Änderung von Umgebungsvariablen auf Workstations unter Windows<sup>®</sup> beispielsweise mit der Notwendigkeit verbunden, das System neu zu starten. Jetzt wird Ihre Umgebung mit wenigen Ausnahmen durch Registriervariablen gesteuert, die in den DB2 UDB-Profilregistrierdatenbanken gespeichert sind. Benutzer auf UNIX<sup>®</sup>-Betriebssystemen mit der Berechtigung zur Systemverwaltung (SYSADM) für ein bestimmtes Exemplar können die Registrierungswerte für das betreffende Exemplar aktualisieren. Windows-Benutzer benötigen keine SYSADM-Berechtigung, um Registrierungswerte aktualisieren zu können. Zur Aktualisierung der Variablen der Profilregistrierdatenbank ohne Systemneustart verwenden Sie den Befehl **db2set**. Diese Informationen werden sofort wirksam in den Profilregistrierdatenbanken gespeichert. Die DB2 UDB-Registrierdatenbank wendet die aktualisierten Informationen auf diejenigen DB2 UDB-Serverexemplare und DB2 UDB-Anwendungen an, die nach der Durchführung der Änderungen gestartet werden.

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

**Anmerkung:** Die DB2 UDB-Umgebungsvariablen DB2INSTANCE und DB2NODE werden eventuell nicht in den DB2 UDB-Profilregistrierdatenbanken gespeichert. Bei einigen Betriebssystemen muss der Befehl **set** zur Aktualisierung dieser Umgebungsvariablen 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. Verschiedene Ebenen der Unterstützung werden nun durch verschiedene Profile realisiert. Eine Fernverwaltung der Umgebungsvariablen steht außerdem zur Verfügung, wenn der DB2-Verwaltungsserver (DAS) verwendet wird.

Es gibt vier Profilregistrierdatenbanken:

- Die DB2 UDB-Profilregistrierdatenbank auf Exemplarebene. Die meisten der DB2 UDB-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 UDB-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 ESE gibt es auf jeder Maschine ein Profil der globalen Ebene.
- Die DB2 UDB-Profilregistrierdatenbank auf Exemplarknotenebene. Diese Registrierdatenbankebene enthält Variableneinstellungen, die für eine Partition (einen Knoten) in einer Umgebung mit mehreren Partitionen spezifisch sind. Die auf dieser Ebene definierten Werte überschreiben die entsprechenden Einstellungen auf der Exemplarebene und der globalen Ebene.
- Die DB2 UDB-Exemplarprofilregistrierdatenbank. Diese Registrierdatenbank enthält eine Liste aller Exemplarnamen, die von diesem System erkannt werden. Sie können die vollständige Liste aller auf dem System verfügbaren Exemplare mit Hilfe des Befehls **db2ilist** anzeigen.

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

1. Umgebungsvariablen, die mit dem Befehl `set` definiert wurden (bzw. mit dem Befehl `export` auf UNIX-Plattformen).
2. Registrierungswerte, die im Profil auf Exemplarknotenebene definiert wurden (mit dem Befehl `db2set -i <exemplarname> <knotennummer>`).
3. Registrierungswerte, die im Profil auf Exemplarebene definiert wurden (mit dem Befehl `db2set -i`).
4. Registrierungswerte, die im Profil auf globaler Ebene definiert wurden (mit dem Befehl `db2set -g`).

### Profilregistrierdatenbank auf Exemplarebene

Bei der Arbeit mit einer partitionierten Datenbankumgebung sind einige Unterschiede zwischen UNIX und Windows zu beachten. Diese Unterschiede werden im folgenden Beispiel gezeigt.

Nehmen Sie an, eine partitionierte Datenbankumgebung besteht aus drei physischen Knoten, die als „red“, „white“ und „blue“ angesprochen werden. Der Exemplareigner führt nun auf UNIX-Plattformen die folgenden Befehle auf einem beliebigen der Knoten aus:

```
db2set -i F00=BAR
```

oder

```
db2set F00=BAR      ('-i' ist impliziert)
```

In diesem Fall wird der Wert von FOO für alle Knoten des aktuellen Exemplars sichtbar (d. h. für „red“, „white“ und „blue“).

Auf UNIX-Plattformen wird die Profilregistrierdatenbank der Exemplarebene in einer Textdatei im Verzeichnis `sql1ib` gespeichert. In Umgebungen mit partitionierten Datenbanken befindet sich das Verzeichnis `sql1ib` in dem Dateisystem, das von allen physischen Knoten gemeinsam benutzt wird.

Wenn der Benutzer den gleichen Befehl vom Knoten „red“ auf Windows-Plattformen ausführt, wird der Wert von FOO nur auf dem Knoten „red“ des aktuellen Exemplars sichtbar. DB2 UDB speichert die Profilregistrierdatenbank der Exemplarebene in der Windows-Registrierdatenbank. Eine gemeinsame Nutzung durch mehrere physische Knoten findet nicht statt. Sollen die Registriervariablen auf allen physischen Maschinen definiert werden, verwenden Sie den Befehl „rah“ wie folgt:

```
rah db2set -i F00=BAR
```

Durch den Befehl „rah“ wird der Befehl „db2set“ auf den Knoten „red“, „white“ und „blue“ fern ausgeführt.

Es ist möglich, mit Hilfe der Umgebungsvariablen `DB2REMOTEPREG` die Registriervariablen auf Maschinen, die nicht Exemplareigner sind, so zu konfigurieren, dass sie auf die Registriervariablen von Exemplareignermaschinen verweisen. Dadurch wird eine Umgebung erstellt, in der die Registriervariablen auf der Exemplareignermaschine von allen Maschinen im Exemplar effektiv gemeinsam benutzt werden.

Im oben gezeigten Beispiel und unter der Annahme, dass „red“ die Exemplareignermaschine ist, müsste die Umgebungsvariable DB2REMOTEPREG auf den Maschinen „white“ und „blue“ wie folgt definiert werden, damit die Registriervariablen auf „red“ gemeinsam genutzt würden:

```
(auf Maschine red) Keine Aktion
(auf den Maschinen white und blue) db2set DB2REMOTEPREG=\\red
```

Die Einstellung für DB2REMOTEPREG darf nicht mehr geändert werden, wenn sie definiert wurde.

Die Umgebungsvariable DB2REMOTEPREG funktioniert wie folgt:

Wenn DB2 UDB die Registriervariablen unter Windows liest, wird zuerst der Wert von DB2REMOTEPREG gelesen. Wenn DB2REMOTEPREG definiert ist, wird die Registrierdatenbank auf der fernen Maschine geöffnet, deren Name in der Variablen DB2REMOTEPREG angegeben ist. Nachfolgende Lese- und Aktualisierungsoperationen für die Registriervariablen werden an die angegebene ferne Maschine weitergeleitet.

Der Zugriff auf die ferne Registrierdatenbank setzt voraus, dass auf der Zielmaschine der Service (Dienst) für die Fernregistrierung (Remote Registry Service) aktiv ist. Außerdem müssen das Benutzeranmeldekonto und alle DB2 UDB-Serviceanmeldekonto über ausreichende Zugriffsrechte für die ferne Registrierdatenbank verfügen. Bei der Verwendung der Variablen DB2REMOTEPREG sollten Sie daher in einer Windows-Domänenumgebung operieren, so dass der erforderliche Registrierungszugriff dem Domänenkonto erteilt werden kann.

Für Microsoft® Cluster Server (MSCS) sind ebenfalls zwei Punkte zu beachten. In einer MSCS-Umgebung sollten Sie die Variable DB2REMOTEPREG nicht verwenden. Bei der Nutzung einer MSCS-Konfiguration, in der alle Maschinen zum gleichen MSCS-Cluster gehören, werden die Registriervariablen in der Clusterregistrierdatenbank gepflegt. Daher haben alle Maschinen im gleichen MSCS-Cluster bereits Zugriff auf sie, so dass die Variable DB2REMOTEPREG in diesem Fall nicht erforderlich ist.

Bei der Arbeit in einer Umgebung mit mehreren Partitionen und Funktionsübernahmekonfiguration (Failover), in der sich Partitionen über mehrere MSCS-Cluster erstrecken, können Sie nicht mit der Variablen DB2REMOTEPREG auf die Exemplareignermaschine verweisen, weil sich die Registriervariablen der Exemplareignermaschine in der Clusterregistrierdatenbank befinden.

#### **Zugehörige Konzepte:**

- „Registrierdatenbank- und Umgebungsvariablen von DB2“ in *Systemverwaltung: Optimierung*

#### **Zugehörige Tasks:**

- „Deklarieren von Registrier- und Umgebungsvariablen“ auf Seite 33



## Deklarieren von Registrier- und Umgebungsvariablen

### Vorgehensweise:

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 Registriervariablen aufzulisten:

```
db2set -all
```

Zum Anzeigen des Wertes einer Registriervariablen geben Sie Folgendes ein:

```
db2set name_der_registriervariablen
```

Zum Anzeigen des Wertes einer Registriervariablen auf allen Ebenen geben Sie Folgendes ein:

```
db2set name_der_registriervariablen -all
```

Zum Ändern einer Registriervariablen im aktuellen oder Standardexemplar geben Sie Folgendes ein:

```
db2set name_der_registriervariablen=neuer_wert
```

Zum Ändern des Standardwerts einer Registriervariablen für alle Datenbanken des Exemplars geben Sie Folgendes ein:

```
db2set name_der_registriervariablen=neuer_wert  
-i exemplarname
```

Zum Ändern des Standardwerts einer Registriervariablen für eine bestimmte Partition eines Exemplars geben Sie Folgendes ein:

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

Zum Ändern des Standardwerts einer Registriervariablen für alle Exemplare im System geben Sie Folgendes ein:

```
db2set name_der_registriervariablen=neuer_wert -g
```

Wenn Sie mit Lightweight Directory Access Protocol (LDAP) arbeiten, können Sie Registriervariablen in LDAP wie folgt definieren:

- Zum Definieren von Registriervariablen auf Benutzerebene in LDAP geben Sie Folgendes ein:

```
db2set -ul
```

- Zum Definieren von Registriervariablen auf globaler Ebene in LDAP geben Sie Folgendes ein:

```
db2set -gl benutzername
```

Bei der Ausführung in einer LDAP-Umgebung können Sie einen Wert für eine DB2 Universal Database™-Registriervariable (DB2 UDB-Registriervariable) so definieren, dass der Bereich für alle Maschinen und Benutzer, die zu einer Verzeichnispartmention oder einer Windows NT-Domäne gehören, global gültig ist. Momentan sind nur zwei DB2 UDB-Registriervariablen definiert, die auf globaler LDAP-Ebene gesetzt werden können: DB2LDAP\_KEEP\_CONNECTION und DB2LDAP\_SEARCH\_SCOPE.

Verwenden Sie, um den Suchbereichswert in LDAP global zu setzen, folgenden Befehl:

```
db2set -gl db2ldap_search_scope = wert
```

Dabei kann als *wert* „local“, „domain“ oder „global“ angegeben werden.

#### Anmerkungen:

1. Wenn die DB2 UDB-Datei 'profile.env' durch den Befehl 'db2set' simultan aktualisiert wird (d. h. gleichzeitig oder nahezu gleichzeitig), wird die Größe der Datei 'profile.env' auf null reduziert. Darüber hinaus zeigt die Ausgabe des Befehls `db2set -all` inkonsistente Werte.
2. Es besteht ein Unterschied zwischen der Option `-g`, die zur Definition von DB2 UDB-Registriervariablen auf der globalen Ebene der Maschine dient, und der Option `-gl`, die für die globale Ebene von LDAP spezifisch ist.
3. Die Registriervariable der Benutzerebene wird nur unter Windows unterstützt, wenn in einer LDAP-Umgebung gearbeitet wird.
4. Variableneinstellungen auf der Benutzerebene enthalten benutzerspezifische Variablenwerte. Alle Änderungen an der Benutzerebene werden in das LDAP-Verzeichnis geschrieben.
5. Die Parameter „-i“, „-g“, „-gl“ und „-ul“ können nicht gleichzeitig im selben Befehl verwendet werden.
6. Manche Variablen beziehen sich standardmäßig immer das Profil der globalen Ebene. Sie können nicht in Profilen der Exemplar- oder Knotenebene definiert werden, z. B. DB2SYSTEM und DB2INSTDEF.
7. Unter UNIX müssen Sie über die Berechtigung zur Systemverwaltung (SYS-ADM) verfügen, um die Registrierungswerte für ein Exemplar zu ändern. Nur Benutzer mit Rootberechtigung können Parameter in Registrierdatenbanken der globalen Ebene ändern.

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

```
db2set -r name_der_registriervariablen
```

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

```
db2set -r name_der_registriervariablen knotennummer
```

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 folgenden Befehl ein, um die Einstellung einer Variablen auf der Knotenebene zu löschen:

```
db2set name_der_registriervariablen -i exemplarname
      knotennummer
```

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

```
db2set name_der_registriervariablen= -null exemplarname
```

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

#### **Zugehörige Konzepte:**

- „Registrierdatenbank- und Umgebungsvariablen von DB2“ in *Systemverwaltung: Optimierung*

#### **Zugehörige Tasks:**

- „Definieren von Umgebungsvariablen unter Windows“ auf Seite 35
- „Definieren von Umgebungsvariablen auf UNIX-Systemen“ auf Seite 37
- „Durchsuchen der LDAP-Verzeichnispartitionen oder -Domänen“ auf Seite 354
- „Definieren von DB2-Registriervariablen auf Benutzerebene in der LDAP-Umgebung“ auf Seite 357

## **Definieren von Umgebungsvariablen unter Windows**

### **Vorgehensweise:**

Es wird ausdrücklich empfohlen, alle spezifischen Registriervariablen in der Profilregistrierdatenbank von DB2 Universal Database™ (DB2 UDB) zu definieren. Wenn die DB2 UDB-Variablen außerhalb der Registrierdatenbank festgelegt werden, ist keine Fernverwaltung dieser Variablen möglich und die Workstation muss erneut gestartet werden, damit die Variablenwerte wirksam werden.

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

Server mit DB2 UDB Enterprise Server Edition unter Windows verfügen über die beiden Systemumgebungsvariablen DB2INSTANCE und DB2NODE, die nur außerhalb der Profilregistrierdatenbank festgelegt werden können. Sie müssen DB2INSTANCE nicht unbedingt festlegen. Die Variable DB2INSTDEF der DB2 UDB-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 UDB-Profilregistrierdatenbank. Wenn diese Variable nicht definiert wird, wird als logischer Zielknoten standardmäßig der logische Knoten verwendet, der auf der Maschine mit der Nummer Null (0) definiert ist.

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

```
echo %db2path%
```

Die Systemumgebungsvariablen werden wie folgt definiert:

**Windows 9x:** Editieren Sie die Datei autoexec.bat und starten Sie das System erneut, um die Änderung in Kraft zu setzen.

**Windows:** Sie können die DB2 UDB-Umgebungsvariablen DB2INSTANCE und DB2NODE (in der folgenden Beschreibung an DB2INSTANCE erläutert) wie folgt definieren:

- (Windows NT und Windows 2000) Wählen Sie **Start, Einstellungen, Systemsteuerung** aus. (Windows XP und Windows Server 2003) Wählen Sie **Start** —> **Systemsteuerung** aus.
- (Windows NT und Windows 2000) Klicken Sie das Symbol **System** doppelt an. (Windows XP und Windows Server 2003) Je nach Windows-Thema und dem momentan ausgewählten Ansichtstyp müssen Sie eventuell zunächst **Leistung** (Performance) und **Wartung** (Maintenance) auswählen, bevor Sie das Symbol **System** auswählen können.
- (Windows NT) Gehen Sie auf der Seite für Umgebungsvariablen des Systems im Applet **System** der Systemsteuerung folgendermaßen vor: (Windows 2000, Windows XP und Windows Server 2003) Im Fenster der Systemeigenschaften müssen Sie die Registerkarte **Erweitert** auswählen und anschließend den Knopf **Umgebungsvariablen** anklicken und folgende Schritte ausführen:
  1. Wenn die Variable DB2INSTANCE nicht vorhanden ist:
    - a. (Windows NT) Wählen Sie eine beliebige Systemumgebungsvariable aus. (Windows 2000, Windows XP und Windows Server 2003) Klicken Sie den Knopf **Neu** an.
    - b. (Windows NT) Ändern Sie im Feld *Variable* den Namen in DB2INSTANCE. (Windows 2000, Windows XP und Windows Server 2003) Geben Sie in das Feld *Variablenname* den Namen DB2INSTANCE ein.
    - c. (Windows NT) Ändern Sie das Feld *Wert* in den Namen des Exemplars, zum Beispiel db2inst. (Windows 2000, Windows XP und Windows Server 2003) Geben Sie in das Feld *Variablenwert* den Namen des Exemplars ein, 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* in den Namen des Exemplars, zum Beispiel db2inst.
  3. (Windows NT) Wählen Sie **Setzen** aus. (Windows 2000, Windows XP und Windows Server 2003) Wählen Sie **OK** 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 UDB-Exemplar mit dem Namen TEST starten wollen, geben Sie die folgenden Befehle in einem Befehlsfenster ein:

```
set DB2INSTANCE=TEST
db2start
```

Bei Verwendung einer C-Shell geben Sie die folgenden Befehle in einem Befehlsfenster ein:

```
setenv DB2INSTANCE TEST
```

Die Profilregistrierdatenbanken befinden sich an folgenden Positionen:

- Die DB2 UDB-Profilregistrierdatenbank auf Exemplarebene in der Registrierdatenbank des Windows-Betriebssystems mit folgendem Pfad:

```
\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2\PROFILES\exemplarname
```

**Anmerkung:** Der Wert *exemplarname* ist der Name des DB2 UDB-Exemplars.

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

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

- Die DB2 UDB-Profilregistrierdatenbank auf Exemplarknotenebene in der Registrierdatenbank von Windows 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.

- Es ist keine DB2 UDB-Profilregistrierdatenbank auf Exemplarebene erforderlich. Für jedes der DB2 UDB-Exemplare im System wird ein Schlüssel in folgendem Pfad erstellt:

```
\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2\PROFILES\exemplarname
```

Die Liste der Exemplare kann durch Zählen der Schlüssel unter dem Schlüssel PROFILES ermittelt werden.

#### Zugehörige Konzepte:

- „DB2-Verwaltungsserver“ auf Seite 45

#### Zugehörige Tasks:

- „Definieren von Umgebungsvariablen auf UNIX-Systemen“ auf Seite 37

## Definieren von Umgebungsvariablen auf UNIX-Systemen

### Vorgehensweise:

Es wird ausdrücklich empfohlen, alle spezifischen Registervariablen in der DB2 UDB-Profilregistrierdatenbank zu definieren. Wenn die DB2 UDB-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 Korn-Shell) und `db2cshrc` (für Bourne-Shell oder 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 Ausgangsverzeichnis 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 exemplarnamen des lokal auszuführenden Standardexempars.

**Anmerkung:** Alle unterstützten Variablen außer `PATH` und `DB2INSTANCE` müssen in der DB2 UDB-Profilregistrierdatenbank festgelegt werden. Definieren Sie Variablen, die von DB2 UDB nicht unterstützt werden, in Ihren Prozedurdateien `userprofile` und `usercshrc`.

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:

```
export DB2INSTANCE=<inst1>
```
- Für C-Shell:

```
setenv DB2INSTANCE <inst1>
```

Zur ordnungsgemäßen Verwaltung der DB2 UDB-Profilregistrierdatenbank müssen die folgenden Regeln im Hinblick auf die Zugriffsberechtigungen und Eigentumsrechte auf UNIX-Betriebssystemen eingehalten werden.

- Die DB2 UDB-Profilregistrierdatenbank auf Exemplarebene befindet sich unter:

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

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

```
-rw-rw-r-- <db2inst1> <db2iadm1> profile.env
```

Dabei ist `<db2inst1>` der Exemplareigner und `<db2iadm1>` die Gruppe des Exemplareigners.

`INSTHOME` ist der Ausgangspfad (`HOME`) des Exemplareigners.

- Die DB2 UDB-Profilregistrierdatenbank auf globaler Ebene befindet sich unter:
  - `/var/db2/<version_id>/default.env` unter den Betriebssystemen AIX, Solaris Operating Environment und Linux (wobei `<version_id>` für die aktuelle Version steht).
  - `/var/opt/db2/<versions_id>/default.env` unter dem Betriebssystem HP-UX (wobei `<versions_id>` für die aktuelle Version steht)

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

```
-rw-rw-r-- <exemplareigner> <exemplareignergruppe> default.env
```

Zum Ändern einer globalen Variablen der Profilregistrierdatenbank muss ein Benutzer als Root angemeldet sein.

- Die DB2 UDB-Profilregistrierdatenbank auf Exemplarknotenebene befindet sich unter:

```
INSTHOME/sqlllib/nodes/<knotennummer>.env
```

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

```
drwxrwsr-w <exemplareigner> <exemplareignergruppe> nodes
```

```
-rw-rw-r-- <exemplareigner> <exemplareignergruppe> <knotennummer>.env
```

INSTHOME ist der Ausgangspfad (HOME) des Exemplareigners.

- Die DB2 UDB-Exemplarregistrierdatenbank befindet sich unter:
  - /var/db2/<versions\_id>/profiles.reg unter den Betriebssystemen AIX, Solaris Operating Environment und Linux (wobei <versions\_id> für die aktuelle Version steht)
  - /var/opt/db2/<versions\_id>/profiles.reg unter dem 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
```

#### Zugehörige Konzepte:

- „DB2-Verwaltungsserver“ auf Seite 45

#### Zugehörige Tasks:

- „Definieren von Umgebungsvariablen unter Windows“ auf Seite 35

## Erstellen einer Knotenkonfigurationsdatei

### Vorgehensweise:

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 `sqlllib` des Ausgangsverzeichnisses (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

Wenn Sie DB2 Universal Database™ (DB2 UDB) Enterprise - Server Edition unter Windows verwenden, wird die Knotenkonfigurationsdatei beim Erstellen des Exemplars erstellt. Sie sollten nicht versuchen, eine Knotenkonfigurationsdatei manuell zu erstellen oder manuell zu ändern. Mit Hilfe des Befehls **db2ncrt** können Sie einem Exemplar einen Datenbankpartitionsserver hinzufügen. Mit Hilfe des Befehls **db2ndrop** können Sie einen Datenbankpartitionsserver für ein Exemplar löschen. Mit Hilfe des Befehls **db2nchg** können Sie die Konfiguration der Datenbankpartitionsserver ändern. Dies umfasst das Versetzen des Datenbankpartitionsservers

von einer Maschine auf eine andere, das Ändern des TCP/IP-Hostnamens oder das Auswählen eines anderen logischen Portnamens oder Netzwerknamens.

**Anmerkung:** Erstellen Sie keine anderen als die von DB2 UDB 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 Prozeduranwendungen in das Unterverzeichnis `function` im Unterverzeichnis `sql1ib`. Die andere Ausnahme betrifft eventuell erstellte benutzerdefinierte Funktionen (UDFs). Benutzerdefinierte Funktionen können im selben Verzeichnis gespeichert werden.

Die Datei enthält eine Zeile für jede Datenbankpartition, die zu einem Exemplar gehört. Jede Zeile hat folgendes Format:

```
dbpartitionsnummer hostname [logischer-port [netzname]]
```

Die Token einer Zeile sind durch Leerzeichen voneinander getrennt. Die Variablen sind:

#### **dbpartitionsnummer**

Die Datenbankpartitionsnummer (mögliche Werte: 0 - 999) definiert einen Knoten eindeutig. Datenbankpartitionsnummern müssen in aufsteigender Reihenfolge angegeben sein. Es dürfen Sprünge in der Folge der Nummern auftreten.

Wenn eine Datenbankpartitionsnummer 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 Datenbankpartitionsnummer für jeden neuen Knoten, den Sie hinzufügen, wieder verwendet werden.

>Die Datenbankpartitionsnummer wird zur Generierung eines Knotennamens im Datenbankverzeichnis verwendet. Er hat folgendes Format:

```
NODEnnnn
```

Die Ziffernfolge `nnnn` ist die Datenbankpartitionsnummer, die links mit Nullen aufgefüllt wird. Diese Datenbankpartitionsnummer wird außerdem durch die Befehle **CREATE DATABASE** und **DROP DATABASE** verwendet.

#### **hostname**

Der Hostname der IP-Adresse für die partitionsübergreifende Kommunikation. Verwenden Sie den vollständig qualifizierten Namen für den Hostnamen. In der Datei `/etc/hosts` sollte ebenfalls der vollständig qualifizierte Name verwendet werden. Wenn der vollständig qualifizierte Name in der Datei `'db2nodes.cfg'` nicht verwendet wird, in der Datei `/etc/hosts` jedoch verwendet wird, empfangen Sie eventuell die Fehlermeldung `SQL30082N RC=3`.

(Es gibt eine Ausnahme, wenn der `netzname` angegeben wird. In diesem Fall wird der `netzname` für den Großteil der Kommunikation verwendet, während `hostname` nur für die Befehle **db2start**, **db2stop** und **db2\_all** verwendet wird.)



### logischer-port

Dieser Parameter ist optional und gibt die logische Portnummer für den Knoten an. Diese Nummer wird mit dem Exemplarnamen des Datenbankmanagers verwendet, um einen TCP/IP-Servicenamenseintrag in der Datei `etc/services` anzugeben.

Die Kombination aus IP-Adresse und logischem Port wird als allgemein bekannte Adresse verwendet und muss für alle Anwendungen eindeutig sein, um die Verbindungen zur Kommunikation zwischen Knoten zu unterstützen.

Für jeden hostnamen muss ein logischer-port entweder 0 (Null) oder leer sein (was standardmäßig 0 bedeutet). Der Knoten, dem dieser logische-port 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 `dbpartitionsnummer` für einen Host), sollten Sie die Nummern für den logischen-port den logischen Knoten von 0 an ohne Lücken zuordnen. Die Reihenfolge ist hierbei nicht relevant.

Die folgende Konfiguration ist z. B. zulässig:

```
| 0 cpaiss43.mach1.xxx.com 1
| 1 cpaiss43.mach1.xxx.com 0
| 2 cpaiss43.mach1.xxx.com 2
| 3 cpaiss44.mach1.xxx.com 0
```

### netzname

Dieser Parameter ist optional und wird zur Unterstützung eines Hosts verwendet, der über mehr als eine aktive TCP/IP-Schnittstelle verfügt, von denen jede ihren eigenen Hostnamen hat.

Das folgende Beispiel zeigt eine mögliche Knotenkonfigurationsdatei für ein System RS/6000 SP, in dem SP2EN1 mehrere TCP/IP-Schnittstellen und zwei logische Partitionen hat und SP2SW1 als DB2 UDB-Schnittstelle verwendet. Es zeigt außerdem die Partitionsnummern, beginnend bei 1 (und nicht bei 0), sowie einen Sprung in der Folge der `dbpartitionsnummern`:

Tabelle 1. Eine Tabelle mit einem Beispiel für Datenbankpartitionsnummern

dbpartitionsnummer	hostname	logischer-port	netzname
1	SP2EN1.mach1.xxx.com	0	SP2SW1
2	SP2EN1.mach1.xxx.com	1	SP2SW1
4	SP2EN2.mach1.xxx.com	0	
5	SP2EN3.mach1.xxx.com		

Die Datei `db2nodes.cfg` kann mit einem beliebigen Editor aktualisiert werden. (Die Ausnahme bildet hierbei, dass unter Windows 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 Datenbankpartitionsnummer nicht geändert wird. Die Knotenkonfigurationsdatei wird gesperrt, wenn der Befehl **db2start** ausgeführt wird, und entsperrt, wenn der Befehl **db2stop** den Datenbankmanager beendet hat.

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.

**Zugehörige Konzepte:**

- „Richtlinien für gespeicherte Prozeduren“ in *Systemverwaltung: Optimierung*

**Zugehörige Referenzen:**

- „db2start - Start DB2 Command“ in *Command Reference*
- „db2stop - Stop DB2 Command“ in *Command Reference*
- „CREATE DATABASE Command“ in *Command Reference*
- „DROP DATABASE Command“ in *Command Reference*
- „db2nchg - Change Database Partition Server Configuration Command“ in *Command Reference*
- „db2ncrt - Add Database Partition Server to an Instance Command“ in *Command Reference*
- „db2ndrop - Drop Database Partition Server from an Instance Command“ in *Command Reference*

## Erstellen der Datenbankkonfigurationsdatei

**Vorgehensweise:**

Für jede Datenbank wird 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 oder verwendet werden (z. B. Codepage der Datenbank, Sortierfolge, DB2 UDB-Release-Level)
- 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)

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 den Befehl **db2\_all**, um die Synchronisation der Konfigurationsdateien bei allen Datenbankpartitionen zu gewährleisten.

**Zugehörige Konzepte:**

- „Absetzen von Befehlen in einer partitionierten Datenbankumgebung“ auf Seite 381

**Zugehörige Tasks:**

- „Konfigurieren von DB2 mit Konfigurationsparametern“ in *Systemverwaltung: Optimierung*

## FCM-Kommunikation

In einer partitionierten Datenbankumgebung wird ein 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 Port für die Kommunikation. Wenn mehrere Partitionen auf demselben Host definiert sind, müssen Sie einen Bereich von Ports, wie unten gezeigt, definieren.

**Überlegungen zu Windows®**

Wenn Sie DB2® Universal Database (DB2 UDB) Enterprise - Server Edition in der Windows-Umgebung verwenden, wird der TCP/IP-Portbereich 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.

Der Serviceeintrag hat folgende Syntax:

```
DB2_exemplar port/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.

**port/tcp**

Der TCP/IP-Port, 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 `services` des Verzeichnisses `etc` gemeinsam benutzt wird, müssen Sie sicherstellen, dass die Anzahl der in der Datei zugeordneten Ports größer oder gleich der größten Anzahl mehrerer Datenbankpartitionen in diesem Exemplar ist. Stellen Sie bei der Zuordnung von Ports außerdem sicher, dass Sie jeden möglichen Prozessor, der als Ersatz verwendet werden kann, mit berücksichtigen.

Wenn die Datei `services` im Verzeichnis `etc` nicht gemeinsam benutzt wird, gelten dieselben Überlegungen, jedoch mit einem zusätzlichen Gesichtspunkt: Sie müssen sicherstellen, dass die für das DB2 UDB-Exemplar definierten Einträge in allen Dateien `services` des Verzeichnisses `etc` 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 Port für die Verwendung durch FCM definieren. Dazu müssen zwei Zeilen in die Datei `services` des Verzeichnisses `etc` eingefügt werden, die den Bereich von Ports angeben, den Sie zuordnen. Die erste Zeile gibt den ersten Port an, während die zweite Zeile das Ende des Blocks von Ports angibt. Im folgenden Beispiel werden fünf Ports für das Exemplar `sales` zugeordnet. Dies bedeutet, dass kein Prozessor im Exemplar mehr als fünf Datenbankpartitionen hat. Beispiel:

```
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.

#### **Zugehörige Konzepte:**

- „Partitions- und Prozessorumgebungen“ in *Systemverwaltung: Konzept*

---

## Kapitel 2. Erstellen und Verwenden des DB2-Verwaltungs-servers (DAS)

Der DB2-Verwaltungsserver (DAS) dient zur Unterstützung von DB2-Servertasks.

---

### DB2-Verwaltungsserver

Der DB2<sup>®</sup>-Verwaltungsserver (DAS - DB2 Administration Server) ist ein Steuerpunkt, der nur zur Unterstützung von Operationen auf DB2 Universal Database<sup>™</sup>-Servern (DB2 UDB-Servern) verwendet wird. Sie müssen über einen aktiven DAS verfügen, wenn Sie solche verfügbaren Tools wie den Konfigurationsassistenten, die Steuerzentrale oder die Entwicklungszentrale verwenden wollen. Der DAS unterstützt die Steuerzentrale und den Konfigurationsassistenten bei den folgenden Verwaltungsaufgaben:

- Aktivieren der Fernverwaltung von DB2 UDB-Servern.
- Bereitstellen der Einrichtung für Jobverwaltung, einschließlich der Fähigkeit, die Ausführung von Befehlsprozeduren sowohl für DB2 UDB als auch für das Betriebssystem zu terminieren. Diese Befehlsprozeduren sind benutzerdefiniert.
- Definieren von Zeitplänen zur Ausführung von Jobs, Anzeigen der Ergebnisse ausgeführter Jobs und Durchführen von Verwaltungstasks an zum DAS fernen oder lokalen Jobs über die Taskzentrale.
- Bereitstellen eines Mechanismus zum Aufspüren von Informationen zur Konfiguration von DB2 UDB-Exemplaren, DB2 UDB-Datenbanken und anderen DB2-Verwaltungsservern in Zusammenarbeit mit dem DB2 UDB-Dienstprogramm Discovery. Diese Informationen dienen dem Konfigurationsassistenten und der Steuerzentrale zur Vereinfachung und Automatisierung der Konfiguration von Clientverbindungen zu DB2 UDB-Datenbanken.

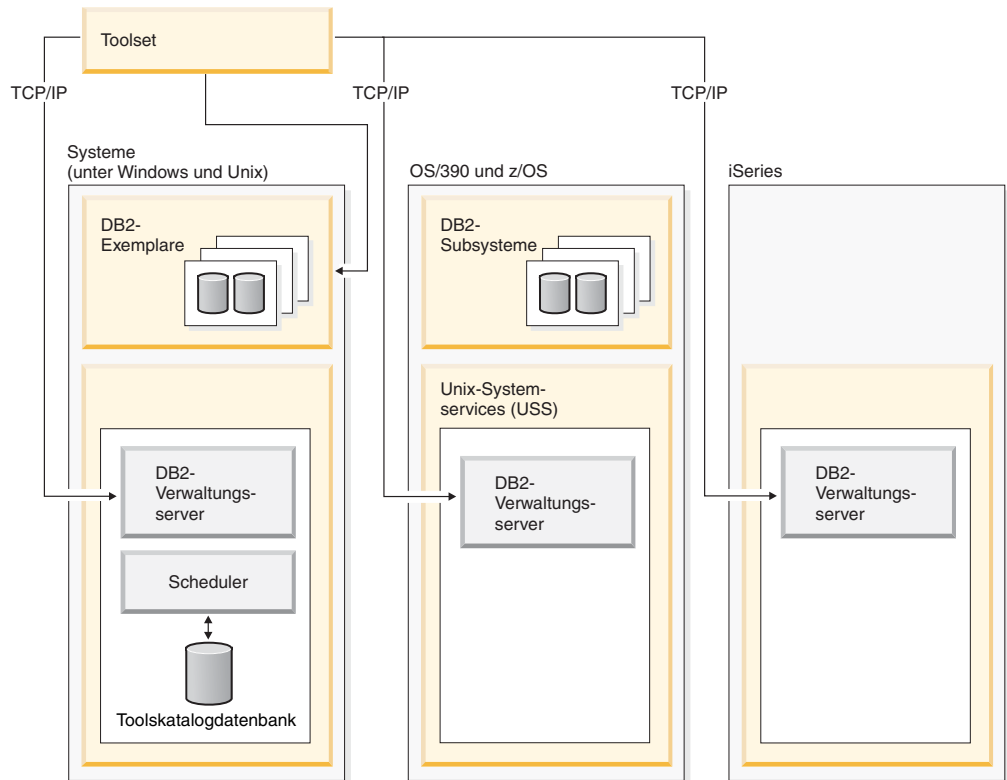


Abbildung 1. Einsatzbereiche des DB2-Verwaltungsservers (DAS)

Sie können nur einen DB2-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 Serversystem und dem Hostsystem für eine Clientanforderung von der Steuerzentrale oder vom Konfigurationsassistenten bzw. von einem der anderen verfügbaren Tools durchzuführen.

Der DAS ist für alle unterstützten Windows<sup>®</sup>- und UNIX<sup>®</sup>-Plattformen sowie für die zSeries<sup>®</sup>-Plattformen (nur OS/390 und z/OS<sup>™</sup>) verfügbar. Der DAS unter zSeries dient zur Unterstützung der Steuerzentrale, der Entwicklungszentrale und der Replikationszentrale bei Verwaltungstasks.

Der DB2-Verwaltungsserver für zSeries-Plattformen (nur OS/390 und z/OS) wird im Paket und als Teil des DB2-Managementclients-Feature von DB2 UDB geliefert. Produkte, für die der DAS erforderlich ist, wie die Steuerzentrale, die Replikationszentrale und die Entwicklungszentrale, setzen die Installation der DAS-Funktion voraus. Informationen zur Verfügbarkeit des DAS für Ihr Betriebssystem erhalten Sie von Ihrem IBM<sup>®</sup> Ansprechpartner.

Der DAS für Windows und UNIX enthält einen Scheduler zur Ausführung von Tasks (z. B. Befehlsprozeduren für DB2 UDB und das Betriebssystem), die über die Taskzentrale definiert werden. Taskinformationen wie die Befehle für die Ausführung, für Zeitplanfunktionen, für die Benachrichtigung und für die der Task zugeordneten Aktionen zur Ausführung nach der Beendigung der Task sowie für die Ausgabe von Ergebnissen werden in einer Reihe von Tabellen und Sichten einer DB2 UDB-Datenbank gespeichert, die als Toolskatalog bezeichnet wird. Der Toolskatalog wird im Rahmen der Installation erstellt. Er kann außerdem über die

Steuerzentrale oder über den Befehl **CREATE TOOLS CATALOG** des Befehlszeilenprozessors erstellt und aktiviert werden.

Obwohl für zSeries (nur OS/390 und z/OS) kein Scheduler bereitgestellt wird, können Sie über die in der Steuerzentrale verfügbaren Funktionen **JCL erzeugen** und **JCL erstellen** JCL (Jobsteuersprache - Job Control Language) generieren, die in partitionierten Dateien gespeichert wird und mit Hilfe der Zeitplanfunktion (Scheduler) des Systems ausgeführt werden kann.

#### **Zugehörige Konzepte:**

- „Sicherheitsaspekte des DB2-Verwaltungsservers unter Windows“ auf Seite 57
- „DAS-Konfiguration auf Enterprise Server Edition (ESE)-Systemen“ auf Seite 61
- „Erkennen (Discovery) von Verwaltungsservern, Exemplaren und Datenbanken“ auf Seite 63
- „FFDC (First Failure Data Capture) durch den DB2-Verwaltungsserver“ auf Seite 67

#### **Zugehörige Tasks:**

- „Erstellen eines DB2-Verwaltungsservers“ auf Seite 47
- „Starten und Stoppen des DB2-Verwaltungsservers“ auf Seite 48
- „Auflisten des DB2-Verwaltungsservers“ auf Seite 49
- „Konfigurieren des DB2-Verwaltungsservers“ auf Seite 50
- „Aktualisieren des DB2-Verwaltungsservers unter UNIX“ auf Seite 57
- „Entfernen des DB2-Verwaltungsservers“ auf Seite 58
- „Einrichten des DB2-Verwaltungsservers mit Enterprise Server Edition (ESE)-Systemen“ auf Seite 59
- „Ausschließen von Serverexemplaren und Datenbanken aus Discovery-Erkennung“ auf Seite 64
- „Einstellen von Discovery-Parametern“ auf Seite 65
- „Konfigurieren des DAS zur Verwendung des Konfigurationsassistenten und der Steuerzentrale“ auf Seite 66
- „Aktualisieren der DAS-Konfiguration für Discovery“ auf Seite 67
- „Einrichten und Konfigurieren der Toolskatalogdatenbank und des DAS-Schedulers“ auf Seite 50
- „Einrichten und Konfigurieren der Benachrichtigung und der Kontaktliste“ auf Seite 55
- „Einrichten der virtuellen Java-Maschine für den DB2-Verwaltungsserver“ auf Seite 56

---

## **Erstellen eines DB2-Verwaltungsservers**

Der DB2-Verwaltungsserver (DAS) stellt Unterstützungsservices für DB2 Universal Database™-Tools (DB2 UDB-Tools) wie die Steuerzentrale und den Konfigurationsassistenten bereit.

#### **Voraussetzungen:**

Für die Erstellung eines DAS müssen Sie auf UNIX-Plattformen über Rootberechtigung verfügen oder ein Konto verwenden, das über die richtige Berechtigung zur Erstellung eines Service (Dienst) verfügt.

Wenn unter Windows ein bestimmter Benutzer angegeben werden muss, erstellen Sie einen Benutzer mit lokaler Administratorberechtigung. Geben Sie den Befehl **db2admin create** ein. Wenn ein bestimmtes Konto gewünscht wird, müssen Sie die Parameter „/USER:“ und „ /PASSWORD:“ beim Absetzen des Befehls **db2admin create** verwenden.

#### Vorgehensweise:

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

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

- Auf Windows-Plattformen:  
Melden Sie sich an der Maschine, auf der Sie den DAS erstellen wollen, mit einem Konto an, das über die richtige Berechtigung zur Erstellung eines Dienstes (Service) verfügt.

Bei der Erstellung des DB2-Verwaltungsservers können Sie optional ein Benutzerkonto und ein Benutzerkennwort angeben. Wenn die Angaben gültig sind, wird mit dem Benutzerkonto und dem Kennwort der Eigner des DAS identifiziert. Verwenden Sie die für den DAS erstellte Benutzer-ID bzw. den entsprechenden Kontonamen nicht als Benutzerkonto. Setzen Sie das Kennwort für den Kontonamen auf „Kennwort läuft nie ab“. Nach der Erstellung des DAS können Sie das Eigentumsrecht für den DAS durch Angabe eines Benutzerkontos und eines Kennworts im Befehl **db2admin setid** festlegen bzw. ändern.

- Auf UNIX-Plattformen:
  1. Stellen Sie sicher, dass Sie über die Rootberechtigung verfügen.
  2. Setzen Sie über eine Befehlseingabeaufforderung den folgenden Befehl aus dem Unterverzeichnis `instance` im DB2 UDB-Installationspfad ab:

```
dascrt -u <DASBenutzer>
```

<DASBenutzer> ist der Benutzername des DAS-Benutzers, den Sie bei der Erstellung von Benutzern und Gruppen für DB2 UDB erstellt haben.

- Unter AIX:

```
/usr/opt/db2_08_01/instance/  
dascrt -u <DASBenutzer>
```

- Unter HP-UX, Solaris Operating Environment oder Linux:

```
/opt/IBM/db2/V8.1/instance/  
dascrt -u <DASBenutzer>
```

#### Zugehörige Referenzen:

- „db2admin - DB2 Administration Server Command“ in *Command Reference*

---

## Starten und Stoppen des DB2-Verwaltungsservers

#### Vorgehensweise:

Zum manuellen Starten des DB2-Verwaltungsservers (DAS) unter Windows müssen Sie sich zunächst an der Maschine mit einem Konto bzw. einer Benutzer-ID anmelden, die zu der Gruppe der Administratoren, Serveroperatoren oder Hauptbenutzer gehört. Zum manuellen Starten oder Stoppen des DAS unter UNIX muss das Konto bzw. die Benutzer-ID der Gruppe `dasadm_group` hinzugefügt werden. Die Gruppe `dasadm_group` wird in den DAS-Konfigurationsparametern angegeben.



Zum Starten oder Stoppen des DAS unter Windows verwenden Sie den Befehl **db2admin start** bzw. **db2admin stop**.

Beim Arbeiten mit DB2 Universal Database™ (DB2 UDB) unter einem der UNIX-Betriebssysteme müssen Sie wie folgt vorgehen:

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

```
. DASHOME/das/dasprofile    (für Bourne- oder Korn-Shell)
source DASHOME/das/dascshrc (für C-Shell)
```

Dabei ist DASHOME das Ausgangsverzeichnis des DB2-Verwaltungsservers.

3. Geben Sie zum Starten des DAS den folgenden **db2admin**-Befehl ein:

```
db2admin start
```

**Anmerkung:** Der DAS wird nach jedem Neustart des Systems automatisch gestartet. Die Standardstartfunktionsweise des DAS kann über den Befehl **dasauto** geändert werden.

- Stoppen des DAS:
  1. Melden Sie sich mit einem Konto bzw. einer Benutzer-ID an, die zur Gruppe *dasadm\_group* gehört.
  2. Stoppen Sie den DAS mit Hilfe des **db2admin**-Befehls wie folgt:

```
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. Der Benutzer muss zur Gruppe *dasadm\_group* gehören, um den Befehl **db2admin start** bzw. **db2admin stop** ausführen zu können.

**Zugehörige Referenzen:**

- „db2admin - DB2 Administration Server Command“ in *Command Reference*
- „dasadm\_group - DASADM-Gruppenname“ in *Systemverwaltung: Optimierung*

---

## Auflisten des DB2-Verwaltungsservers

**Vorgehensweise:**

Mit dem folgenden Befehl können Sie den Namen des DB2-Verwaltungsservers (DAS) auf Ihrer Maschine feststellen:

```
db2admin
```

Dieser Befehl dient darüber hinaus auch zum Starten und Stoppen des DAS, Erstellen eines neuen Benutzers und Kennworts, Löschen des DAS sowie zum Einrichten und Ändern eines Benutzerkontos, das dem DAS zugeordnet ist.

**Zugehörige Referenzen:**

- „db2admin - DB2 Administration Server Command“ in *Command Reference*

---

## Konfigurieren des DB2-Verwaltungsservers

### Vorgehensweise:

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.

Um die DAS-Konfigurationsdatei mit Hilfe des Befehlszeilenprozessors (CLP) und des Befehls UPDATE ADMIN CONFIG zu aktualisieren, müssen Sie den CLP von einem Exemplar aus verwenden, das sich auf derselben Installationsstufe befindet wie DAS. Zum Aktualisieren einzelner Einträge in der DAS-Konfigurationsdatei, geben Sie folgenden Befehl ein:

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

Wenn die Konfigurationsparameter auf die empfohlenen Standardwerte zurückgesetzt werden sollen, geben Sie folgenden Befehl ein:

```
db2 reset admin cfg
```

In einigen Fällen werden Änderungen an der DAS-Konfigurationsdatei erst wirksam, wenn sie in den Hauptspeicher geladen werden (d. h., wenn der Befehl **db2admin stop** gefolgt von dem Befehl **db2admin start** ausgeführt wird oder, bei einer Windows-Plattform, wenn der Dienst gestoppt und wieder gestartet wird). In anderen Fällen sind die Konfigurationsparameter online konfigurierbar (d. h., der DAS muss nicht neu gestartet werden, damit die neuen Werte wirksam werden).

### Zugehörige Tasks:

- „Konfigurieren von DB2 mit Konfigurationsparametern“ in *Systemverwaltung: Optimierung*

### Zugehörige Referenzen:

- „UPDATE ADMIN CONFIGURATION Command“ in *Command Reference*

---

## Einrichten und Konfigurieren der Toolskatalogdatenbank und des DAS-Scheduler

| Die Toolskatalogdatenbank (Tools Catalog) enthält die Taskinformationen, die von  
| der Taskzentrale und der Steuerzentrale erstellt werden. Diese Tasks werden durch  
| den Scheduler (Zeitplanfunktion) des DB2-Verwaltungsservers (DAS) ausgeführt.  
| Der Scheduler und die Toolskatalogdatenbank arbeiten immer zusammen. Der eine  
| kann ohne die andere nicht funktionieren. Der Scheduler ist eine spezifische Kom-  
| ponente der DB2-Verwaltungsservers, die als Agent zum Lesen der Toolskatalogda-  
| tenbank fungiert und die Tasks zu den jeweils festgelegten Zeiten ausführt.

### Voraussetzungen:

Der DB2-Verwaltungsserver muss installiert sein.

### Vorgehensweise:

Der Zweck besteht darin, die Toolskatalogdatenbank und den DAS-Scheduler einzurichten und zu konfigurieren.

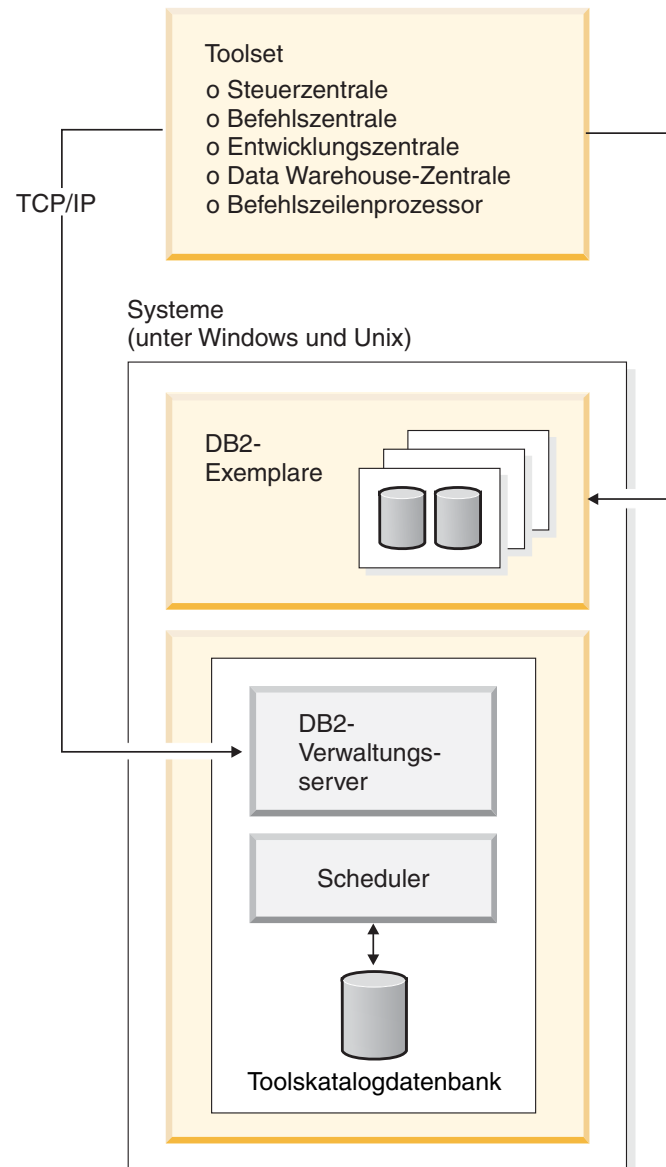


Abbildung 2. Beziehung des DAS zu anderen Komponenten von DB2 UDB

Der Konfigurationsprozess des DB2-Verwaltungsservers teilt dem Scheduler die Position der Toolskatalogdatenbank mit und gibt an, ob der Scheduler aktiviert werden soll oder nicht. Standardmäßig wird bei der Erstellung einer Toolskatalogdatenbank die entsprechende DAS-Konfiguration aktualisiert. Das heißt, der Scheduler wird konfiguriert und steht für die Verwendung des neuen Toolskatalogs bereit. Es besteht keine Notwendigkeit, den DAS erneut zu starten.

Die Toolskatalogdatenbank kann auf einem zum Scheduler-System fernen oder lokalen Server erstellt werden. Wenn die Toolskatalogdatenbank auf einem fernen Server erstellt wird, muss sie im Exemplar der Scheduler-Toolskatalogdatenbank

(TOOLSCAT\_INST) katalogisiert werden. Darüber hinaus muss die Scheduler-Benutzer-ID mit Hilfe des Befehls **db2admin setschedid** definiert werden, so dass der Scheduler eine Verbindung zum fernen Katalog herstellen und eine Authentifizierung durchführen kann. Die vollständige Syntax des Befehls **db2admin** finden Sie im Handbuch *Command Reference*.

Der DAS-Scheduler erfordert eine virtuelle Java-Maschine (JVM) für den Zugriff auf die Informationen im Toolskatalog. Die JVM-Informationen werden mit Hilfe des Konfigurationsparameters `jdk_path` für den DB2-Verwaltungsserver angegeben.

Der Konfigurationsparameter `jdk_64_path` ist erforderlich, wenn Sie einen Toolskatalog für ein 64-Bit-Exemplar auf einer der Plattformen erstellen, die 32- und 64-Bit-Exemplare unterstützen (AIX, Sun und HP-UX).

Die Steuerzentrale und die Taskzentrale greifen direkt vom Client auf die Toolskatalogdatenbank zu. Die Toolskatalogdatenbank muss daher auf dem Client katalogisiert werden, bevor sie von der Steuerzentrale verwendet werden kann. Die Steuerzentrale enthält einen Mechanismus zum automatischen Abrufen der Informationen über die Toolskatalogdatenbank sowie zum Erstellen der erforderlichen Verzeichniseinträge im lokalen Knotenverzeichnis und im Datenbankverzeichnis. Das einzige für diese automatische Katalogfunktion unterstützte Protokoll ist TCP/IP.

Einer der DAS-Konfigurationsparameter heißt *exec\_exp\_task*. Dieser Parameter gibt an, ob der Scheduler die Tasks ausführt, die in der Vergangenheit terminiert wurden, jedoch noch nicht ausgeführt wurden. Der Scheduler erkennt verfallene Tasks nur, wenn er gestartet wird.

Wenn Sie zum Beispiel einen Job so terminiert haben, dass er jeden Samstag ausgeführt wird, und der Scheduler am Freitag ausgeschaltet und am folgenden Montag wieder eingeschaltet wird, ist der für Samstag terminierte Job nun ein in der Vergangenheit terminierter verfallener Job. Wenn der Konfigurationsparameter *exec\_exp\_task* aktiviert („Yes“) ist, wird der für Samstag terminierte Job ausgeführt, wenn der Scheduler erneut gestartet wird.

Die anderen DAS-Konfigurationsparameter, die für den Scheduler erforderlich sind, identifizieren die Toolskatalogdatenbank und den SMTP-Server (Simple Mail Transfer Protocol), der zur Benachrichtigung verwendet werden soll.

Die folgenden Beispiele erläutern die Verwendung dieser Parameter:

- Eine Beispieleinrichtung auf einem Windows-Server:

1. Die Toolskatalogdatenbank kann einen beliebigen Namen haben. In diesem Beispiel heißt die Toolskatalogdatenbank „CCMD“ und wird unter dem Exemplar von DB2 Universal Database™ (DB2 UDB) auf der Servermaschine Host1 (TCP/IP-Hostname Host1) erstellt. Zur eindeutigen Identifizierung der Toolskatalogdatenbank wird ein Schemaname verwendet. Für dieses Beispiel wird der Schemaname „CCADMIN“ angenommen.
2. Das Exemplar „DB2“ wird für die TCP/IP-Kommunikation über Portnummer 50000 wie folgt eingerichtet:

```
db2set -i DB2 DB2COMM=TCPIP
db2 update dbm cfg using svcname db2cDB2
db2stop
db2start
```

3. Der Servicename db2cDB2 wird in der Datei %SystemRoot%\system32\drivers\etc\services definiert. Das heißt, in der Datei services wäre folgende Zeile zu finden:
 

```
db2cDB2      50000/tcp      #connection port for the DB2 instance DB2
```
4. Der Toolskatalog wird durch den Befehl CREATE TOOLS CATALOG erstellt. Durch diesen Befehl werden die Tabellen und Sichten des Toolskatalogs mit einem Schemanamen erstellt, der dem Katalognamen in der angegebenen Datenbank entspricht. Die Konfigurationsparameter des DB2-Verwaltungsservers werden automatisch aktualisiert und der Scheduler aktiviert und gestartet.
5. Nehmen Sie an, dass sich der SMTP-Server, der für E-Mail-Benachrichtigungen verwendet wird, auf der Maschine Host2 (TCP/IP-Hostname Host2) befindet. Diese Informationen werden dem DB2-Verwaltungsserver mit folgendem Befehl angegeben:

```
db2 update admin cfg using smtp_server Host2
```

Dies kann während des Installationsprozesses erfolgen. Wenn es später erfolgt, müssen die Informationen dem DAS manuell mit Hilfe des oben gezeigten Befehls des Befehlszeilenprozessors von DB2 UDB Version 8 angegeben werden.

6. IBM Software Development Kit (SDK) für Java ist unter Windows im Verzeichnis %DB2PATH%\java\jdk installiert. Dies sollte dem DAS bereits angegeben sein. Ob dies der Fall ist, kann überprüft und bei Bedarf mit folgendem Befehl durchgeführt werden:

```
db2 update admin cfg using jdk_path c:\SQLLIB\java\jdk
```

Dieser Befehl geht davon aus, dass DB2 UDB unter C:\SQLLIB installiert ist.

**Anmerkung:** Wenn der DAS durch den Befehl db2admin create erstellt wird, stellen Sie sicher, dass die Optionen /USER und /PASSWORD verwendet werden. Das USER-Konto wird vom Schedulerprozess verwendet. Ohne diese Angabe wird der Scheduler nicht ordnungsgemäß gestartet. Das USER-Konto sollte über die Berechtigung SYSADM für das Exemplar des Toolskatalogs verfügen.

Wenn der DAS durch den Befehl db2admin create erstellt werden soll und die Optionen /USER und /PASSWORD zu diesem Zeitpunkt nicht angegeben werden sollen, können Sie die USER-Informationen zu einem späteren Zeitpunkt aktualisieren. Diese Aktualisierung erfolgt auf dem DAS durch Ausführen der folgenden Befehle:

```
db2admin stop
db2admin setid <benutzerkonto-ID> <kennwort>
db2admin start
```

- Eine Beispieleinrichtung auf einem Windows-Client:
  1. Dieses Beispiel geht davon aus, dass die Steuerzentrale auf der Clientmaschine C1 (TCP/IP-Hostname C1) ausgeführt wird.
  2. Der DAS wird als Verwaltungsserverknoten im lokalen Knotenverzeichnis entweder über den Konfigurationsassistenten oder die Steuerzentrale bzw. durch den folgenden Befehl katalogisiert:
 

```
db2 catalog admin tcpip node Host1 remote Host1 system Host1 ostype NT
```
  3. Wenn die Taskzentrale gestartet und das System Host1 ausgewählt wird, versucht die Taskzentrale, die Toolskatalogdatenbank im lokalen Verzeichnis zu finden. (Anstelle der Taskzentrale könnte die Steuerzentrale verwendet

werden.) Falls dies keinen Erfolg hat, versucht sie, den Knoten und die Datenbank mit folgendem Befehl zu katalogisieren:

```
db2 catalog tcpip node <eindeutiger-knotenname>
remote Host1 server 50000
remote_instance DB2 system Host1 ostype NT
db2 catalog db CCMD as <eindeutiger-db-alias> at node <eindeutiger-knotenname>
```

Wenn die automatische Katalogisierung erfolglos ist, kann die Datenbank über den Konfigurationsassistenten oder die Steuerzentrale katalogisiert werden. Anschließend wird die Datenbank von der Taskzentrale erkannt und verwendet.

- Eine Beispieleinrichtung auf einem AIX-Server:

1. Die Toolskatalogdatenbank kann einen beliebigen Namen haben. In diesem Beispiel heißt die Toolskatalogdatenbank „CCMD“ und wird unter dem Exemplar „db2inst1“ auf der Servermaschine Host1 (TCP/IP-Hostname Host1) erstellt. Zur eindeutigen Identifizierung der Toolskatalogdatenbank wird ein Schemaname verwendet. Für dieses Beispiel wird der Schemaname „CCADMIN“ angenommen.

2. Das Exemplar „db2inst1“ wird für die TCP/IP-Kommunikation über Portnummer 50000 wie folgt eingerichtet:

```
db2set -i DB2 DB2COMM=TCPIP
db2 update dbm cfg using svcename xdb2inst
db2stop
db2start
```

3. Der Servicename xdb2inst wird in der Datei /etc/services definiert. Das heißt, in der Datei services wäre folgende Zeile zu finden:

```
xdb2inst1 50000/tcp #connection port for the DB2 instance db2inst1
```

4. Der Toolskatalog wird durch den Befehl CREATE TOOLS CATALOG erstellt. Durch diesen Befehl werden die Tabellen und Sichten des Toolskatalogs mit einem Schemanamen erstellt, der dem Katalognamen in der angegebenen Datenbank entspricht. Die Konfigurationsparameter des DB2-Verwaltungsservers werden automatisch aktualisiert und der Scheduler aktiviert und gestartet.

5. Nehmen Sie an, dass sich der SMTP-Server, der für E-Mail-Benachrichtigungen verwendet wird, auf der Maschine Host2 (TCP/IP-Hostname Host2) befindet. Diese Informationen werden dem DB2-Verwaltungsserver mit folgendem Befehl angegeben:

```
db2 update admin cfg using smtp_server Host2
```

Dies kann während des Installationsprozesses erfolgen. Wenn es später erfolgt, müssen die Informationen dem DAS manuell mit Hilfe des oben gezeigten Befehls des Befehlszeilenprozessors von DB2 UDB Version 8 angegeben werden.

6. IBM Software Developer's Kit für Java (SDK) Version 1.3.1 ist unter AIX im Verzeichnis /sql1ib/java/jdk installiert. Dies sollte dem DAS bereits angegeben sein. Ob dies der Fall ist, kann überprüft und bei Bedarf mit folgendem Befehl durchgeführt werden:

```
db2 update admin cfg using jdk_path /sql1ib/java/jdk
```

- Eine Beispieleinrichtung auf einem AIX-Client:

1. Dieses Beispiel geht davon aus, dass die Steuerzentrale auf der Clientmaschine C1 (TCP/IP-Hostname C1) ausgeführt wird.
2. Der DAS wird als Verwaltungsserverknoten im lokalen Knotenverzeichnis entweder über den Konfigurationsassistenten oder die Steuerzentrale durch

den folgenden Befehl katalogisiert:

```
db2 catalog admin tcpip node Host1 remote Host1 system Host1
ostype AIX
```

3. Wenn die Taskzentrale gestartet und das System Host1 ausgewählt wird, versucht die Taskzentrale, die Toolskatalogdatenbank im lokalen Verzeichnis zu finden. (Anstelle der Taskzentrale könnte die Steuerzentrale verwendet werden.) Falls dies keinen Erfolg hat, versucht sie, den Knoten und die Datenbank mit folgendem Befehl zu katalogisieren:

```
db2 catalog tcpip node <eindeutiger-knotenname>
remote Host1 server 50000
remote_instance DB2 system Host1 ostype AIX
db2 catalog db CCMD as <eindeutiger-db-alias> at node <eindeutiger-knotenname>
```

Wenn die automatische Katalogisierung erfolglos ist, kann die Datenbank über den Konfigurationsassistenten oder die Steuerzentrale katalogisiert werden. Anschließend wird die Datenbank von der Taskzentrale erkannt und verwendet.

#### Zugehörige Referenzen:

- „svcname - TCP/IP-Servicename“ in *Systemverwaltung: Optimierung*
- „sched\_enable - Schedulermodus“ in *Systemverwaltung: Optimierung*
- „toolscat\_inst - Exemplar der Toolskatalogdatenbank“ in *Systemverwaltung: Optimierung*
- „toolscat\_db - Toolskatalogdatenbank“ in *Systemverwaltung: Optimierung*
- „toolscat\_schema - Schema der Toolskatalogdatenbank“ in *Systemverwaltung: Optimierung*
- „smtp\_server - SMTP-Server“ in *Systemverwaltung: Optimierung*
- „jdk\_path - Installationspfad für Software Developer's Kit für Java auf DAS“ in *Systemverwaltung: Optimierung*
- „exec\_exp\_task - Verfallene Tasks ausführen“ in *Systemverwaltung: Optimierung*

---

## Einrichten und Konfigurieren der Benachrichtigung und der Kontaktliste

Benachrichtigungen per E-Mail und Pager vom DB2-Verwaltungsserver (DAS) können lokal oder fern erfolgen. Eine Kontaktliste ist erforderlich, um sicherzustellen, dass Benachrichtigungen an den richtigen Hostnamen gesendet werden.

#### Vorgehensweise:

Zwei DAS-Konfigurationsparameter dienen zur Aktivierung von Benachrichtigungen durch den Scheduler oder den Diagnosemonitor.

Mit dem DAS-Konfigurationsparameter *smtp\_server* wird der SMTP-Server (Simple Mail Transfer Protocol) angegeben, der vom Scheduler zum Senden von Benachrichtigungen per E-Mail oder Pager im Rahmen der Aktionen, die über die Taskzentrale als nach Abschluss der Taskausführung auszuführend definiert sind, oder vom Diagnosemonitor zum Senden von Warnnachrichten per E-Mail oder Pager verwendet wird.

Mit dem DAS-Konfigurationsparameter *contact\_host* wird die Position angegeben, an der Kontaktinformationen gespeichert sind, die vom Scheduler bzw. vom Diagnosemonitor zur Benachrichtigung verwendet werden. Die Position wird in Form des TCP/IP-Hostnamens eines DB2-Verwaltungsservers definiert. Dadurch,

dass sich der durch den Parameter *contact\_host* angegebene Kontakthost auf einem fernen DAS befinden kann, wird eine gemeinsame Verwendung der Kontaktliste durch mehrere DB2-Verwaltungsserver unterstützt. Dieser Parameter sollte für Umgebungen mit partitionierten Datenbanken definiert werden, um sicherzustellen, dass für alle Partitionen eine gemeinsame Kontaktliste verwendet wird. Die Kontaktliste wird in einer unstrukturierten Datei unter dem DAS-Verzeichnis gespeichert. Wenn der Parameter *contact\_host* nicht definiert ist, nimmt der DAS an, dass die Kontaktinformationen lokal vorliegen.

**Zugehörige Referenzen:**

- „smtp\_server - SMTP-Server“ in *Systemverwaltung: Optimierung*
- „contact\_host - Speicherposition der Liste mit Ansprechpartnern“ in *Systemverwaltung: Optimierung*

---

## Einrichten der virtuellen Java-Maschine für den DB2-Verwaltungsserver

**Vorgehensweise:**

Der Konfigurationsparameter *jdk\_path* gibt das Verzeichnis an, unter dem das Produkt IBM Software Developer's Kit (SDK) für Java installiert ist, das zur Ausführung von Funktionen des DB2-Verwaltungsservers (DAS) zu verwenden ist. Die Umgebungsvariablen, die durch den Java-Interpreter verwendet werden, werden aus dem Wert dieses Parameters ermittelt.

Der Scheduler benötigt eine virtuelle Java-Maschine (JVM), um die Toolskatalogdatenbank verwenden zu können. Daher ist es erforderlich, die JVM einzurichten, bevor der Scheduler erfolgreich gestartet werden kann.

Auf UNIX-Plattformen ist für diesen Parameter kein Standardwert vorgesehen. Sie sollten einen Wert für diesen Parameter definieren, wenn Sie IBM Software Developer's Kit (SDK) für Java installieren.

IBM Software Developer's Kit (SDK) für Java wird unter Windows im Verzeichnis %DB2PATH%\java\jdk installiert (was gleichzeitig der Standardwert für diesen Parameter auf Windows-Plattformen ist). Dies sollte dem DAS bereits angegeben sein. Sie können den Wert für *jdk\_path* mit folgendem Befehl überprüfen:

```
db2 get admin cfg
```

Dieser Befehl zeigt die Werte der Konfigurationsdatei für den DB2-Verwaltungsserver an, wobei *jdk\_path* einer der Konfigurationsparameter ist. Bei Bedarf kann ein Wert für den Parameter mit folgendem Befehl definiert werden:

```
db2 update admin cfg using jdk_path 'C:\Program Files\IBM\SQLLIB'
```

Dieser Befehl geht davon aus, dass DB2 Universal Database™ (DB2 UDB) im Verzeichnis 'C:\Program Files\IBM\SQLLIB' installiert ist.

IBM Software Developer's Kit (SDK) für Java wird unter AIX im Verzeichnis /usr/java130 installiert. Bei Bedarf kann ein Wert für den Parameter mit folgendem Befehl definiert werden:

```
db2 update admin cfg using jdk_path /usr/java130
```

**Anmerkung:** Wenn Sie einen Toolskatalog für ein 64-Bit-Exemplar auf einer der Plattformen, die 32- und 64-Bit-Exemplare unterstützen (AIX, Sun



oder HP-UX), erstellen oder verwenden, verwenden Sie den Konfigurationsparameter `jdk_64_path` anstelle des Parameters `jdk_path`. Dieser Konfigurationsparameter gibt das Verzeichnis an, unter dem die 64-Bit-Version von IBM Software Developer's Kit (SDK) für Java installiert ist.

**Zugehörige Referenzen:**

- „GET ADMIN CONFIGURATION Command“ in *Command Reference*
- „UPDATE ADMIN CONFIGURATION Command“ in *Command Reference*
- „jdk\_path - Installationspfad für Software Developer's Kit für Java auf DAS“ in *Systemverwaltung: Optimierung*

---

## Sicherheitsaspekte des DB2-Verwaltungsservers unter Windows

Sie müssen eventuell die Benutzer-ID ändern, unter der der DAS-Dienst unter Windows<sup>®</sup> ausgeführt wird.

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. Vor dem Ausführen dieses Befehls müssen Sie sich an einer Maschine mit einem Konto oder einer Benutzer-ID anmelden, die über die lokale Administratorberechtigung verfügt.

**Anmerkung:** Denken Sie daran, dass bei Kennwörtern die Groß-/Kleinschreibung beachtet werden muss. Eine Mischung aus Groß- und Kleinbuchstaben ist zulässig, was bedeutet, dass die exakte Schreibweise des Kennworts sehr wichtig ist.

**Anmerkung:** Unter Windows 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<sup>®</sup>-Verwaltungsserver (DAS) immer den Befehl **db2admin**.

**Zugehörige Referenzen:**

- „db2admin - DB2 Administration Server Command“ in *Command Reference*

---

## Aktualisieren des DB2-Verwaltungsservers unter UNIX

**Vorgehensweise:**

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

Sie müssen sich zunächst mit Superuserberechtigung, in der Regel als „root“, an der Maschine anmelden.

Der Befehl ist wie folgt anzugeben:

```
dasupdt
```

Für diesen Befehl gibt es außerdem optionale Parameter:

- `-h` oder `-?`  
Zeigt ein Hilfemenü für diesen Befehl an.
- `-d`  
Aktiviert den Debugmodus, der für die Problemanalyse verwendet wird.
- `-D`  
Versetzt DAS von einer höheren Codestufe auf einem Pfad auf eine niedrigere Codestufe, die auf einem anderen Pfad installiert ist.

**Anmerkung:** Unter Windows gehört die Aktualisierung des DAS zum Installationsprozess. Es sind keine Aktionen seitens des Benutzers erforderlich.

#### Beispiele:

DAS führt Version 8.1.2-Code im Installationspfad der Version 8 aus. Wenn FixPak 3 im Installationspfad von Version 8 installiert wurde, aktualisiert der folgende Befehl DAS zu FixPak 3. Dieser Befehl wird im Installationspfad von Version 8 aufgerufen:

```
dasupdt
```

DAS führt Version 8.1.2-Code in einem alternativen Installationspfad aus. Wenn FixPak 1 in einem anderen alternativen Installationspfad installiert wurde, aktualisiert der folgende Befehl, der im alternativen Installationspfad von FixPak 1 aufgerufen wird, DAS auf FixPak 1, indem er im alternativen Installationspfad von FixPak 1 ausgeführt wird:

```
dasupdt -D
```

#### Zugehörige Konzepte:

- „DB2-Verwaltungsserver“ auf Seite 45
- „Sicherheitsaspekte des DB2-Verwaltungsservers unter Windows“ auf Seite 57

---

## Entfernen des DB2-Verwaltungsservers

### Vorgehensweise:

Der DB2-Verwaltungsserver (DAS) wird folgendermaßen entfernt:

- Unter Windows-Betriebssystemen:
  1. Melden Sie sich an der Maschine mit einem Konto oder einer Benutzer-ID an, die über die richtige Berechtigung zum Entfernen eines Dienstes (Service) verfügt.
  2. Stoppen Sie den DAS mit dem Befehl **db2admin stop**.
  3. Sichern Sie (falls erforderlich) alle Dateien im Unterverzeichnis `db2das00` des Unterverzeichnisses `sql1ib`.

**Anmerkung:** Dieses Beispiel geht davon aus, dass `db2das00` der Name des zu entfernenden DAS ist. Es ist möglich, einen DAS mit einem anderen Namen als `DB2DAS00` zu haben, falls ein Benutzer ein DB2 Universal Database™-Exemplar (DB2 UDB-Exemplar) erstellt hat, das den Namen `DB2DAS00` besitzt. In einem solchen Fall erhält der DAS den Namen `DB2DAS01` (oder, falls dieser bereits benutzt wird, den Namen `DB2DAS02` usw.). Suchen Sie nach dem Dienst mit dem Präfix „DB2DAS“, um den speziellen DAS aus der Liste mehrerer DAS zu ermitteln, die eventuell vorhanden sein können. Zum Auflisten aller DAS können Sie den Befehl **db2admin** ohne Optionen verwenden.

4. Löschen Sie den DAS mit dem Befehl **db2admin drop**.
- Unter UNIX-Betriebssystemen:
    1. Melden Sie sich als Benutzer mit `DASADM`-Berechtigung an.
    2. Führen Sie die Startprozedur mit einem der folgenden Befehle aus:

```
. DASHOME/das/dasprofile    (für Bourne- oder Korn-Shell)
source DASHOME/das/dascshrc  (für C-Shell)
```

Dabei ist `DASHOME` das Ausgangsverzeichnis des DAS-Eigners.

3. Stoppen Sie den DAS mit Hilfe des **db2admin**-Befehls wie folgt:

```
db2admin stop
```
4. Sichern Sie (falls erforderlich) alle Dateien im Unterverzeichnis `das` des DAS-Ausgangsverzeichnisses.
5. Melden Sie sich ab.
6. Melden Sie sich als `root` an, und entfernen Sie den DAS mit dem Befehl **dasdrop** wie folgt:

```
dasdrop
```

Der Befehl **dasdrop** befindet sich im Unterverzeichnis `instance` des Unterverzeichnisses für das spezifische Release der installierten DB2 UDB-Version.

**Anmerkung:** Der Befehl **dasdrop** entfernt das Unterverzeichnis `das` unter dem Ausgangsverzeichnis des DB2-Verwaltungsservers (DAS).

#### Zugehörige Referenzen:

- „db2admin - DB2 Administration Server Command“ in *Command Reference*
- „dasdrop - Remove a DB2 Administration Server Command“ in *Command Reference*

---

## Einrichten des DB2-Verwaltungsservers mit Enterprise Server Edition (ESE)-Systemen

### Vorgehensweise:

Die folgenden Informationen beschreiben die erforderlichen Schritte zur Konfiguration Servern mit DB2 Universal Database™ (DB2 UDB) ESE (Linux, Solaris Operating Environment, Windows NT, Windows 2000, Windows Server 2003, HP-UX und AIX) zur Fernverwaltung über die Steuerzentrale.

Während der Installation erstellt das Konfigurationsprogramm einen einzelnen DB2-Verwaltungsserver (DAS) auf der Exempleignermaschine. Sie müssen weitere DAS auf anderen Maschinen zu erstellen, damit die Steuerzentrale oder der

Konfigurationsassistent auf andere Koordinatorknoten zugreifen kann. Der Systemaufwand für die Funktionen des Koordinatorknotens kann dadurch auf mehrere Partitionen eines Exemplars verteilt werden. Das Installationsprogramm erstellen den DAS auf allen Knoten, auf denen es ausgeführt wird. Nur wenn Sie **db2setup** nicht verwenden, müssen Sie dies manuell erledigen.

Die im Folgenden erläuterten Anweisungen gelten nur für eine partitionierte ESE-Umgebung. Wenn Sie ein ESE-System mit nur einer Partition betreiben, treffen die Anweisungen auf Ihre Umgebung nicht zu.

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 im Konfigurationsassistenten als separates 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 UDB-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 ESE V8.1

**Installationspfad:**

installationspfad

**TCP-Servicedatei:**

services

**DB2 UDB-Exemplar:**

**Name:** db2inst

**Eigner-ID:**

db2inst

**Exemplarpfad:**

exemplarpfad

**Knoten:**

3 Knoten, db2nodes.cfg:

- 0 hostA 0 hostAswitch
- 1 hostA 1 hostAswitch
- 2 hostB 0 hostBswitch

**DBname:**

db2instDB

**DAS:**

**Name:** db2as00

**Eigner/Benutzer-ID:**

db2as

**Exemplarpfad:**

das\_pfad

**Installations-/Ausführungshost:**

hostA

**Port für die Kommunikation zwischen Knoten:**

16000 (nicht benutzter Port für HostA und HostB)

**Anmerkung:** Geben Sie für die obigen Felder die standortspezifischen Werte ein. Zum Beispiel enthält die folgende Tabelle Beispielpfadnamen für einige unterstützte ESE-Plattformen:

*Tabelle 2. Beispielpfadnamen für unterstützte ESE-Plattformen*

Pfad	DB2 UDB ESE für AIX	DB2 UDB ESE für Solaris Operating Environment	DB2 UDB ESE für Windows
<i>installationspfad</i>	/usr/opt/<v_r_ID>	/opt/IBM/db2/<v_r_ID>	C:\sqllib
<i>exemplarpfad</i>	/home/db2inst/sqllib	/home/db2inst/sqllib	C:\profiles\db2inst
<i>das_pfad</i>	/home/db2as/das	/home/db2as/das	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 plattformspezifische Version und die Release-Kennung. Zum Beispiel entspricht <v\_r\_ID> bei DB2 UDB ESE für AIX in Version 8 dem Wert db2\_08\_01.

Beim Installieren von DB2 UDB ESE erstellt das Konfigurationsprogramm einen DAS auf der Exempleignermaschine. Der Datenbankpartitionsserver befindet sich auf der gleichen Maschine wie der DAS und ist der Verbindungspunkt für das Exemplar. Das bedeutet, dieser Datenbankpartitionsserver ist der Koordinator-knoten für Anforderungen der Steuerzentrale oder des Konfigurationsassistenten an das Exemplar.

Wenn auf jeder physischen Maschine ein DAS installiert wird, kann jede Maschine als Koordinatorknoten fungieren. Jede physische Maschine erscheint in der Steuerzentrale oder im Konfigurationsassistenten als separates DB2SYSTEM. Wenn verschiedene Clients verschiedene Systeme zur Herstellung der Verbindung zu einem Server einer partitionierten Datenbank nutzen, wird dadurch die Funktionalität des Koordinatorknotens und die Arbeitslast für eingehende Verbindungen verteilt.

**Zugehörige Konzepte:**

- „DB2-Verwaltungsserver“ auf Seite 45
- „DAS-Konfiguration auf Enterprise Server Edition (ESE)-Systemen“ auf Seite 61

---

## DAS-Konfiguration auf Enterprise Server Edition (ESE)-Systemen

Der DAS ist ein administrativer Steuerpunkt, der bestimmte Tasks für die Tools ausführt. Es kann nur jeweils einen DAS pro physischer Maschine geben. Im Fall eines ESE-Exemplars, das aus mehreren Maschinen besteht, müssen alle Maschinen einen DAS ausführen, damit die Steuerzentrale das ESE-Exemplar verwalten kann. Dieser DAS (db2as) wird von dem System repräsentiert, das in der Baumstruktur

der Steuerzentrale als übergeordnetes Objekt (Elter) des Zielexemplars (db2inst) von DB2<sup>®</sup> Universal Database (DB2 UDB) vorhanden ist.

Zum Beispiel besteht db2inst aus drei Knoten, die auf zwei physische Maschinen oder Hosts verteilt sind. Die Mindestanforderung kann durch das Ausführen des Befehls **db2as** auf HostA und HostB erfüllt werden.

#### **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 des DAS auf HostA ausführen, unabhängig von der MLN-Konfiguration (MLN - Multiple Logical Nodes) für diesen Host.
2. Auf jeder Maschine bzw. jedem physischen Knoten ist ein DAS erforderlich, der jeweils mit dem Befehl **dascrt** erstellt werden muss. Der DAS auf jeder Maschine bzw. jedem physischen Knoten muss aktiv sein, damit die Taskzentrale und die Steuerzentrale ordnungsgemäß funktionieren. Die ID db2as muss auf HostA und auf HostB vorhanden sein. Das Ausgangsverzeichnis der ID db2as darf nicht gegenseitig zwischen den beiden Systemen angehängt werden. Alternativ können verschiedene Benutzer-IDs zur Erstellung des DAS auf HostA und HostB verwendet werden.

Unter DB2 Universal Database<sup>™</sup> (DB2 UDB) Enterprise Server Edition for Windows<sup>®</sup> wird derjenige Datenbankpartitionsserver als Koordinatorknoten verwendet, der sich auf derselben Maschine wie der DAS befindet, wenn Sie die Konfiguration der Verbindung zu einem DB2 UDB-Server mit Hilfe des Konfigurationsassistenten oder der Steuerzentrale automatisieren. Dies bedeutet, dass alle physischen Verbindungen vom Client zur Datenbank an den Koordinatorknoten geleitet werden, bevor sie an andere Datenbankpartitionsserver weitergeleitet werden.

Unter DB2 Universal Database (DB2 UDB) Enterprise Server Edition for Windows haben Sie, wenn Sie zusätzliche DB2-Verwaltungsserver auf anderen Maschinen erstellen, die Möglichkeit, andere Systeme über den Konfigurationsassistenten oder die Steuerzentrale mit Hilfe von DB2 Discovery als Koordinatorknoten zu konfigurieren.

Beim Arbeiten unter DB2 Universal Database (DB2 UDB) Enterprise Server Edition for Windows führt der DB2 UDB Remote Command Service (**db2rcmd.exe**) die Verwaltungskommunikation zwischen den Knoten automatisch durch.

Die Steuerzentrale kommuniziert mit dem DAS über den TCP-Service-Port 523. Dieser Port ist für die ausschließliche Benutzung durch DB2 UDB reserviert. Aus diesem Grund ist das Einfügen neuer Einträge in die TCP-Servicedatei nicht erforderlich.

#### **Zugehörige Tasks:**

- „Erstellen eines DB2-Verwaltungsservers“ auf Seite 47

#### **Zugehörige Referenzen:**

- „db2admin - DB2 Administration Server Command“ in *Command Reference*

---

## Erkennen (Discovery) von Verwaltungsservern, Exemplaren und Datenbanken

Zur Konfiguration von Verbindungen zu einer fernen Maschine stehen zwei Methoden zur Verfügung: die Verwendung des in den Konfigurationsassistenten integrierten Discovery-Service oder die Verwendung eines vorhandenen Verzeichnisservice wie Lightweight Directory Access Protocol (LDAP).

Der Discovery-Service ist in den Konfigurationsassistenten und den DB2<sup>®</sup>-Verwaltungsserver (DAS) integriert. Zur Konfiguration einer Verbindung zu einer fernen Maschine muss sich der Benutzer an der Clientmaschine anmelden und den Konfigurationsassistenten (CA) ausführen. Der Konfigurationsassistent sendet ein Broadcastsignal an alle Maschinen im Netzwerk. Jede Maschine, auf der ein DAS installiert und für Discovery konfiguriert ist, antwortet auf das Broadcastsignal vom Konfigurationsassistenten, indem sie ein Paket zurücksendet, das alle Exemplar- und Datenbankinformationen dieser Maschine enthält. Anschließend verwendet der Konfigurationsassistent die Informationen in diesem Paket zur Konfiguration der Clientkonnektivität. Mit Hilfe der Discovery-Methode können Kataloginformationen für einen fernen Server automatisch im lokalen Datenbank- und Knotenverzeichnis generiert werden.

Die Discovery-Methode setzt voraus, dass Sie sich an jeder Clientmaschine anmelden und den Konfigurationsassistenten ausführen. Wenn Sie eine Umgebung mit einer sehr großen Anzahl von Clients verwalten, kann dies sehr schwierig und zeitaufwendig sein. In diesem Fall können Sie alternativ einen Verzeichnisservice wie LDAP verwenden.

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 ermittelt werden können. Der Discovery-Modus SEARCH bietet die gleichen Funktionen wie der Modus KNOWN sowie zusätzlich die Möglichkeit, Ihr lokales Netzwerk nach anderen DB2 Universal Database<sup>™</sup>-Servern (DB2 UDB-Servern) zu durchsuchen.

Damit ein System den Discovery-Modus KNOWN unterstützt, setzen Sie den Parameter *discover* in der DAS-Konfigurationsdatei auf den Wert KNOWN. Damit ein System beide Discovery-Modi, d. h. KNOWN und SEARCH, unterstützt, setzen Sie den Parameter *discover* in der DAS-Konfigurationsdatei auf den Wert SEARCH (Standardwert). Wenn Sie verhindern wollen, dass ein System und alle dazugehörigen Exemplare und Datenbanken ermittelt werden können, setzen Sie diesen Parameter auf DISABLE. Durch das Setzen des Parameters *discover* in der DAS-Konfigurationsdatei auf den Wert DISABLE wird verhindert, dass das System vom Discovery-Service erkannt wird.

**Anmerkung:** Der Discovery-Modus SEARCH gibt denselben TCP/IP-Hostnamen an einen Client zurück wie Ihr DB2 UDB-Serversystem, wenn Sie den Befehl **hostname** eingeben. Die diesem Hostnamen auf dem Client zugeordnete IP-Adresse wird entweder durch den auf Ihrer Clientmaschine 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 UDB-Serversystem mehrere Adapterkarten konfiguriert sind, müssen Sie sicherstellen, dass TCP/IP auf dem Server so konfiguriert ist,

dass der richtige Hostname zurückgegeben wird und dass der DNS oder die Datei *hosts* des lokalen Clients den Hostnamen 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 Clientexemplar (oder in dem als Client eingesetzten Server) wie folgt festgelegt:

- **KNOWN**

Der Discovery-Modus KNOWN wird vom Konfigurationsassistenten und der Steuerzentrale zum Abrufen von Exemplar- und Datenbankinformationen verwendet, die Systemen zugeordnet sind, die Ihrem lokalen System bereits bekannt sind. Neue Systeme können über die Funktionalität **Systeme hinzufügen** in den Tools hinzugefügt werden. Wenn der Parameter *discover* auf den Wert KNOWN gesetzt ist, können Sie das Netzwerk nicht durchsuchen.

- **SEARCH**

Aktiviert alle Funktionen des Discovery-Modus KNOWN und zusätzlich die lokale Netzwerksuche. Dies bedeutet, dass jede Suche auf das lokale Netzwerk beschränkt ist.

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* hat auf allen DB2-Verwaltungsservern (DAS) den Standardwert SEARCH.

**Zugehörige Konzepte:**

- „LDAP-Verzeichnisservice (Lightweight Directory Access Protocol)“ auf Seite 77

**Zugehörige Tasks:**

- „Ausschließen von Serverexemplaren und Datenbanken aus Discovery-Erkennung“ auf Seite 64
- „Einstellen von Discovery-Parametern“ auf Seite 65

---

## Ausschließen von Serverexemplaren und Datenbanken aus Discovery-Erkennung

**Vorgehensweise:**

Möglicherweise arbeiten Sie mit mehreren Exemplaren mit jeweils mehreren Datenbanken auf einem Serversystem. Es kann wünschenswert sein, einige dieser Exemplare bzw. Datenbanken von der Erkennung durch Discovery auszuschließen.

Damit Clients die Serverexemplare auf einem System erkennen können, setzen Sie den Konfigurationsparameter *discover\_inst* des Datenbankmanagers auf jedem Serverexemplar auf den Wert ENABLE (Standardwert). Setzen Sie diesen Parameter auf DISABLE, um das betreffende Exemplar und dessen Datenbanken von der Erkennung durch Discovery auszuschließen.



Damit eine Datenbank von einem Client aufgespürt werden kann, setzen Sie den Datenbankkonfigurationsparameter *discover\_db* auf den Wert ENABLE (Standardwert). Setzen Sie diesen Parameter auf den Wert DISABLE, um die Datenbank von der Erkennung durch Discovery auszuschließen.

**Anmerkung:** Wenn ein Exemplar von Discovery erkannt werden soll, muss außerdem der Parameter *discover* in der DAS-Konfigurationsdatei auf den Wert KNOWN oder SEARCH gesetzt werden. Wenn eine Datenbank von Discovery erkannt werden soll, muss der Parameter *discover\_inst* im Serverexemplar aktiviert (ENABLE) werden.

**Zugehörige Referenzen:**

- „discover\_inst - Discovery-Serverexemplar“ in *Systemverwaltung: Optimierung*
- „discover\_db - Discovery-Unterstützung für diese Datenbank“ in *Systemverwaltung: Optimierung*

---

## Einstellen von Discovery-Parametern

**Vorgehensweise:**

Der Parameter *discover* wird in der DAS-Konfigurationsdatei auf dem Serversystem und in der Datenbankmanagerkonfigurationsdatei auf dem Client eingestellt. Stellen Sie die Konfigurationsparameter für den Datenbankmanager über den Konfigurationsassistenten oder die Steuerzentrale ein: *discover*, *discover\_inst*, *discover\_db*. Gehen Sie wie folgt vor, um diese Parameter zu definieren:

- Auf dem DAS:

Aktualisieren Sie den Parameter *discover* (als Beispiel) in der DAS-Konfigurationsdatei mit Hilfe des Befehlszeilenprozessors:

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

Der DAS-Konfigurationsparameter *discover* ist online konfigurierbar. Das heißt, der DAS braucht nicht gestoppt und erneut gestartet zu werden, um die Änderung in Kraft zu setzen.

**Anmerkung:** Der Discovery-Modus SEARCH funktioniert nur über TCP/IP.

- Über den Konfigurationsassistenten:

Starten Sie den Konfigurationsassistenten durch Eingabe von **db2ca** in der Befehlszeile (alle Plattformen) oder über das Menü Start (unter Windows): Klicken Sie hierzu auf **Start** → **Programme** → **IBM DB2** → **Installations- und Konfigurationstools** → **Konfigurationsassistent**.

Gehen Sie wie folgt vor, um mit dem Konfigurationsassistenten die Konfigurationsparameter des Datenbankmanagers zu setzen:

1. Klicken Sie auf **Konfigurieren** → **DBM-Konfiguration**.
2. Klicken Sie auf das Schlüsselwort, das modifiziert werden soll.
3. Klicken Sie in der Spalte für den Wert auf einen Wert für das Schlüsselwort, das Sie modifizieren möchten, und klicken Sie anschließend auf **OK**.
4. Klicken Sie erneut auf **OK**. Daraufhin wird eine Nachricht angezeigt. Klicken Sie **Schließen** an.

Legen Sie die Parameter *discover\_inst* und *discover\_db* über die Steuerzentrale fest.

Sie können außerdem den Konfigurationsassistenten zur Aktualisierung von Konfigurationsparametern verwenden.

**Zugehörige Referenzen:**

- „discover\_inst - Discovery-Serverexemplar“ in *Systemverwaltung: Optimierung*
- „discover\_db - Discovery-Unterstützung für diese Datenbank“ in *Systemverwaltung: Optimierung*
- „UPDATE ADMIN CONFIGURATION Command“ in *Command Reference*
- „discover - DAS-Discovery-Modus“ in *Systemverwaltung: Optimierung*

---

## Konfigurieren des DAS zur Verwendung des Konfigurationsassistenten und der Steuerzentrale

**Voraussetzungen:**

Sie müssen den Parameter **discover** zum Abrufen von Informationen zu Systemen in Ihrem Netzwerk konfigurieren.

**Einschränkungen:**

In jeder physischen Partition muss sich ein DAS befinden. Wenn ein DAS in der Partition erstellt wird, wird DB2SYSTEM mit dem TCP/IP-Hostnamen konfiguriert und der Parameter **discover** auf den Standardwert SEARCH gesetzt.

**Vorgehensweise:**

DB2 Discovery ist eine Einrichtung, die vom Konfigurationsassistenten und von der Steuerzentrale verwendet wird. Zu den für diese Einrichtung erforderlichen Konfigurationsschritten gehört möglicherweise das Aktualisieren der Konfiguration des DB2-Verwaltungsservers (DAS) sowie der Konfiguration des Datenbankmanagers eines Exemplars um sicherzustellen, dass DB2 Discovery die richtigen Informationen abrufen.

Wenn ein Client eine Discovery-Anforderung vom Konfigurationsassistenten bzw. von der Steuerzentrale aus absetzt, antwortet jeder DAS, dessen Discovery aktiviert ist. In einer Umgebung mit partitionierten Datenbanken antwortet jede physische Partition als getrennter DB2SYSTEM-Name. Die Exemplare, die tatsächlich verwaltet werden können, hängen von den Exemplaren ab, die der jeweiligen physischen Partition bekannt sind. Da sich Exemplare über mehrere Partitionen verteilen können, kann dasselbe Exemplar potenziell über verschiedene Systemnamen verwaltet werden. Diese Möglichkeit kann Ihnen bei der Verteilung der Arbeitslast auf das Serverexemplar behilflich sein. Wenn zum Beispiel ein Exemplar „A“ über System „S1“ und System „S2“ verfügbar ist, können einige Benutzer eine Datenbank mit Hilfe von „S1“ katalogisieren und einige Benutzer die gleiche Datenbank mit Hilfe von „S2“. Jeder Benutzer könnte die Verbindung zum Server über eine andere Koordinatorpartition herstellen.

**Zugehörige Referenzen:**

- „db2ilist - List Instances Command“ in *Command Reference*
- „db2ncrt - Add Database Partition Server to an Instance Command“ in *Command Reference*
- „discover - DAS-Discovery-Modus“ in *Systemverwaltung: Optimierung*

---

## Aktualisieren der DAS-Konfiguration für Discovery

### Einschränkungen:

In jeder physischen Partition muss sich ein DAS befinden. Wenn ein DAS in der Partition erstellt wird, wird DB2SYSTEM mit dem TCP/IP-Hostnamen konfiguriert und der Parameter *discover* auf den Standardwert SEARCH gesetzt.

### Vorgehensweise:

Discovery ruft die Namen der Systeme ab, auf denen sich ein DB2-Verwaltungsserver (DAS) befindet. Discovery verwendet diese Namen beim Herstellen von Verbindungen als Koordinator-knoten.

Wenn Sie eine DAS-Konfiguration aktualisieren und dabei die Auswahl eines Koordinator-knotens aus einer Liste von DB2 Universal Database™-Systemen (DB2 UDB-Systemen) haben wollen, definieren Sie den Parameter *discover=SEARCH* (Standardwert) in der Konfigurationsdatei jedes DB2-Verwaltungsservers.

Wenn mehr als ein DAS in einer Serverumgebung mit partitionierten Datenbanken vorhanden sind, kann es vorkommen, dass ein Exemplar auf mehreren Systemen in der Schnittstelle des Konfigurationsassistenten oder der Steuerzentrale angezeigt wird. Jedes System verwendet jedoch einen anderen Kommunikationszugriffspfad für die Exemplare. Benutzer können verschiedene DB2 UDB-Systeme als Koordinator-knoten für die Kommunikation auswählen und dadurch die Auslastung umverteilen.

### Zugehörige Referenzen:

- „Verschiedene Variablen“ in *Systemverwaltung: Optimierung*

---

## FFDC (First Failure Data Capture) durch den DB2-Verwaltungsserver

First Failure Data Capture (FFDC), was in etwa „Erfassung erster Fehlerdaten“ bedeutet, ist eine allgemeine Bezeichnung für den Satz von Diagnoseinformationen, die vom DB2®-Verwaltungsserver automatisch erfasst werden, wenn Fehler auftreten. Diese Informationen verringern die Notwendigkeit, Fehler zur Beschaffung von Diagnoseinformationen zu reproduzieren. Die Diagnoseinformationen werden an einer Position gespeichert.

Zu den Informationen, die von der FFDC-Erfassung des DB2-Verwaltungsservers aufgezeichnet werden, gehören folgende:

- Verwaltungsbenachrichtigungsprotokolle  
Wenn ein Ereignis auftritt, schreibt der DB2-Verwaltungsserver Informationen in die DAS-Protokolldatei *db2dasdiag.log*.
- Speicherauszugsdateien (Dump files)  
Für einige Fehlerbedingungen werden zusätzliche Informationen in externen binären Auszugsdateien gespeichert, die nach der fehlerhaften Prozess-ID benannt sind. Diese Dateien sind zur Verwendung durch die Kundenunterstützung für DB2 Universal Database™ (DB2 UDB) vorgesehen.

- Trapdateien

Der DB2-Verwaltungsserver generiert eine Trapdatei, falls er die Verarbeitung aufgrund einer Fehlerunterbrechung (Trap), einer Segmentierungsverletzung oder einer Ausnahmebedingung nicht fortsetzen kann. Trapdateien enthalten einen Funktionsablauf der letzten Schritte, die ausgeführt wurden, bevor ein Problem auftrat.

### Position der FFDC-Informationen des DB2-Verwaltungsservers

Standardmäßig werden die FFDC-Informationen des DB2-Verwaltungsserver an folgenden Positionen abgelegt:

- Auf Windows®-Systemen:

Wenn die Umgebungsvariable DB2INSTPROF nicht definiert ist:

```
db2path\DB2DAS00\dump
```

Dabei ist db2path der Pfad, der in der Umgebungsvariablen DB2PATH angegeben ist, und DB2DAS00 ist der Name des DAS-Service (Dienst). Der DAS-Name kann angezeigt werden, indem der Befehl **db2admin** ohne Argumente eingegeben wird.

Wenn die Umgebungsvariable DB2INSTPROF definiert ist:

```
x:\db2instprof\DB2DAS00\dump
```

Dabei ist x: das Laufwerk, das in der Umgebungsvariablen DB2PATH angegeben ist, db2instprof ist das Exemplarprofilverzeichnis und DB2DAS00 ist der Name des DAS-Service.

- Auf UNIX®-Systemen:

```
$DASHOME/das/dump
```

Dabei ist \$DASHOME das Ausgangsverzeichnis des DAS-Benutzers.

**Anmerkung:** Sie sollten das Auszugsverzeichnis regelmäßig bereinigen, um zu verhindern, dass es zu sehr anwächst.

### Interpretieren des Protokolls des DB2-Verwaltungsservers

Das Format der Protokolldatei des DB2-Verwaltungsservers (db2dasdiag.log) ist dem Format der DB2-FFDC-Protokolldatei db2diag.log ähnlich. Informationen darüber, wie die Datei db2dasdiag.log zu interpretieren ist, finden Sie in den Abschnitten zum Interpretieren der Verwaltungsprotokolle in den Fehlerbehebungsthemen.

#### Zugehörige Konzepte:

- „DB2-Verwaltungsserver“ auf Seite 45

---

## 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.

---

### Erstellen einer Datenbank

#### Voraussetzungen:

Vor der Erstellung einer Datenbank sollten Sie genügend Zeit auf die Planung des Inhalts, des Layouts, des potenziellen Anwachsens sowie auf die gewünschte Nutzung der Datenbank verwenden.

#### Vorgehensweise:

Beim Erstellen einer Datenbank werden die folgenden Tasks für Sie ausgeführt:

- Einrichten 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 über die 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 den Ordner **Datenbanken** und wählen Sie im Kontextmenü **Erstellen** → **Standard** oder **Erstellen** → **Mit automatischer Verwaltung** aus.
3. Führen Sie die Task gemäß den angegebenen Schritten aus.

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"
```

Bei der Erstellung einer Datenbank können Sie auch die Verwendung des **Konfigurationsadvisors** anfordern, der bei der Konfiguration der Datenbank hilft, anstatt für alle Konfigurationsparameter die Standardwerte zu übernehmen. Geben Sie zu diesem Zweck die Option **AUTOCONFIGURE** im Befehl **CREATE DATABASE** an:

```
CREATE DATABASE <datenbankname>
    AUTOCONFIGURE
```

Die Klausel **AUTOCONFIGURE** unterstützt mehrere Optionen. Sie können die Klausel **AUTOCONFIGURE** jedoch nicht verwenden, wenn Sie eine Datenbank in einer partitionierten Umgebung erstellen.

Bei der Erstellung einer Datenbank wird gleichzeitig auch ein detaillierter Ereignismonitor für gegenseitige Sperren erstellt. Wie bei jedem Monitor fällt für diesen Ereignismonitor etwas Systemaufwand an. Wenn der detaillierte Ereignismonitor für gegenseitige Sperren nicht aktiviert sein soll, kann er mit dem folgenden Befehl gelöscht werden:

```
DROP EVENT MONITOR db2detaildeadlock
```

Zur Begrenzung der Größe des Plattenspeicherplatzes, den dieser Ereignismonitor beansprucht, wird der Ereignismonitor inaktiviert und eine Nachricht wird in das Protokoll mit Benachrichtigungen für die Systemverwaltung geschrieben, wenn die maximale Anzahl von Ausgabedateien erreicht ist. Durch das Entfernen der Ausgabedateien, die nicht mehr benötigt werden, kann der Ereignismonitor bei der nächsten Datenbankaktivierung erneut aktiviert werden.

Sie haben die Möglichkeit, eine Datenbank in einem anderen, möglicherweise fernem Datenbankmanagerexemplar zu erstellen. In dieser Art von Umgebung können Sie Verwaltungsoperationen auf Exemplarebene an einem anderem Exemplar als Ihrem Standardexemplar, einschließlich ferner Exemplare, ausführen.

#### **Zugehörige Konzepte:**

- „Festlegen der in einer Datenbank aufzuzeichnenden Daten“ in *Systemverwaltung: Konzept*
- „Mehrere Exemplare des Datenbankmanagers“ auf Seite 6
- „Datenbankberechtigungen“ auf Seite 255
- „Zusätzliche Überlegungen zum Datenbankentwurf“ in *Systemverwaltung: Konzept*

#### **Zugehörige Referenzen:**

- „CREATE DATABASE Command“ in *Command Reference*

---

## **Definition der ersten Datenbankpartitionsgruppen**

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 DBPARTITIONNUM** und **DROP DBPARTITIONNUM VERIFY** hinzugefügt bzw. gelöscht werden.

Es werden drei Datenbankpartitionsgruppen definiert:

- **IBMCATGROUP** ist die Datenbankpartitionsgruppe für den Tabellenbereich **SYSCATSPACE**, in dem die Systemkatalogtabellen gespeichert werden.

- IBMTEMPGROUP ist die Datenbankpartitionsgruppe für den Tabellenbereich TEMPSPACE1, in dem temporäre Tabellen gespeichert werden, die bei Datenbankoperationen erstellt werden.
- IBMDEFAULTGROUP ist die Datenbankpartitionsgruppe für den Tabellenbereich USERSPACE1, der standardmäßig die Benutzertabellen und Indizes enthält.

**Zugehörige Konzepte:**

- „Datenbankpartitionsgruppen“ in *Systemverwaltung: Konzept*

**Zugehörige Referenzen:**

- „ADD DBPARTITIONNUM Command“ in *Command Reference*
- „DROP DBPARTITIONNUM VERIFY Command“ in *Command Reference*

## Definieren der ersten Tabellenbereiche

Bei der Erstellung einer Datenbank werden folgende drei Tabellenbereiche definiert:

- Der Bereich SYSCATSPACE für die Systemkatalogtabellen
- 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. Als Wert des Parameters EXTENTSIZE für diese Tabellenbereiche wird der Standardwert verwendet.

**Voraussetzungen:**

Die Datenbank muss erstellt sein, und Sie müssen über die Berechtigung zur Erstellung von Tabellenbereichen verfügen.

**Vorgehensweise:**

Gehen Sie wie folgt vor, um die ersten Tabellenbereiche über die 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 den Ordner **Datenbanken** und wählen Sie im Kontextmenü **Erstellen** → **Standard** oder **Erstellen** → **Mit automatischer Verwaltung** aus.
3. Führen Sie die Task gemäß den angegebenen Schritten aus.

Geben Sie in der Befehlszeile Folgendes ein, um erste Tabellenbereiche 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 unter Windows 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.

**Anmerkung:** In einer Umgebung mit partitionierten Datenbanken können Sie keine Behälter erstellen oder bestimmten Partitionen zuordnen. Sie müssen die Datenbank zunächst mit den Standardtabellenbereichen für Benutzertabellen und temporäre Tabellen erstellen. Anschließend können Sie mit Hilfe der Anweisung **CREATE TABLESPACE** die erforderlichen Tabellenbereiche erstellen. Im letzten Schritt können Sie die Standardtabellenbereiche löschen.

Für die Codierung des Ausdrucks **MANAGED BY** im Befehl **CREATE DATABASE** gilt dasselbe Format wie für den Ausdruck **MANAGED BY** im Befehl **CREATE TABLESPACE**.

#### Zugehörige Konzepte:

- „Definition von Systemkatalogtabellen“ auf Seite 73
- „Aufbau von Tabellenbereichen“ in *Systemverwaltung: Konzept*

#### Zugehörige Tasks:

- „Erstellen eines Tabellenbereichs“ auf Seite 83

#### Zugehörige Referenzen:

- „CREATE DATABASE Command“ in *Command Reference*

---

## Erstellen eines Pufferpools

Sie können neue Pufferpools erstellen, die vom Datenbankmanager verwendet werden. Diese Pufferpools verbessern die Leistung des Datenbanksystems sofort.



Die für die Tabellenbereiche angegebene Seitengröße sollte die Seitengröße festlegen, die für die Pufferpools ausgewählt wird. Die Auswahl der Seitengröße, die für einen Pufferpool angegeben wird, ist wichtig, da die Seitengröße nach der Erstellung des Pufferpools nicht mehr geändert werden kann.

#### **Voraussetzungen:**

Die Berechtigungs-ID der Anweisung muss über die Berechtigung SYSCTRL oder SYSADM verfügen.

Vor der Erstellung eines neuen Pufferpools müssen die folgenden Fragen geklärt werden:

- Welcher Pufferpoolname soll verwendet werden?
- Soll der Pufferpool sofort oder erst im Anschluss an die nächste Inaktivierung und erneute Aktivierung der Datenbank erstellt werden?
- Soll der Pufferpool einer Untergruppe aller Datenbankpartitionen zugeordnet werden, aus denen die Datenbank besteht?
- Welche Werte sollen den Parametern zugeordnet werden, die zur Steuerung der Pufferpoolgröße dienen? Hierbei muss auch die Seitengröße und die Gesamtgröße des Pufferpools auf der Basis der Seitenanzahl berücksichtigt werden.
- Soll die Funktion für den erweiterten Speicher, die Unterstützungsfunktion für die blockbasierte Ein-/Ausgabe oder keine der Funktionen eingesetzt werden?

#### **Vorgehensweise:**

Gehen Sie wie folgt vor, um einen neuen Pufferpool zu erstellen:

1. Verwenden Sie die Anweisung `SELECT BPNAME FROM SYSCAT.BUFFERPOOLS`, um die Liste der Pufferpoolnamen abzurufen, die in der Datenbank bereits vorhanden sind.
2. Wählen Sie einen Pufferpoolnamen aus, der sich momentan nicht in der Ergebnisliste befindet. Der Name darf nicht mit den Zeichen „SYS“ oder „IBM“ beginnen.
3. Legen Sie die Merkmale des zu erstellenden Pufferpools fest.
4. Vergewissern Sie sich, dass die korrekte Berechtigungs-ID zum Ausführen der Anweisung `CREATE BUFFERPOOL` vorhanden ist.
5. Führen Sie die Anweisung `CREATE BUFFERPOOL` aus.

#### **Zugehörige Tasks:**

- „Ändern eines Pufferpools“ auf Seite 183

#### **Zugehörige Referenzen:**

- „CREATE BUFFERPOOL statement“ in *SQL Reference, Volume 2*

---

## **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. Sie können diese Tabellen nicht explizit erstellen oder löschen, jedoch können Sie ihren Inhalt abfragen. Bei der Erstellung einer Datenbank werden neben den Objekten der Systemkatalogtabellen auch die folgenden Datenbankobjekte im Systemkatalog definiert:

- Eine Gruppe von Routinen (Funktionen und Prozeduren) wird in den Schemata SYSIBM, SYSFUN und SYSPROC erstellt.
- Eine Gruppe von Sichten für die Systemkatalogtabellen mit reinem Lesezugriff wird im Schema SYSCAT erstellt.
- 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 untersuchen oder um statistische Daten ohne Verwendung des Dienstprogramms **RUNSTATS** zu aktualisieren.

Nach der Erstellung der Datenbank ist es vielleicht sinnvoll, den Zugriff auf die Systemkatalogsichten einzuschränken.

#### **Zugehörige Konzepte:**

- „User-defined functions“ in *SQL Reference, Volume 1*
- „Catalog views“ in *SQL Reference, Volume 1*
- „Functions overview“ in *SQL Reference, Volume 1*

#### **Zugehörige Tasks:**

- „Schützen der Systemkatalogsicht“ auf Seite 279

#### **Zugehörige Referenzen:**

- „Functions“ in *SQL Reference, Volume 1*

---

## **Definition von Datenbankverzeichnissen**

Drei Verzeichnisse werden zur Einrichtung einer neuen Datenbank verwendet.

- Lokales Datenbankverzeichnis
- Systemdatenbankverzeichnis
- Knotenverzeichnis

### **Lokales Datenbankverzeichnis**

In jedem Pfad (bzw. „Laufwerk“ unter Windows®-Betriebssystemen), in dem eine Datenbank definiert wurde, ist eine Datei für das *lokal Datenbankverzeichnis* vorhanden. 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

#### **Zugehörige Referenzen:**

- „CREATE DATABASE Command“ in *Command Reference*

## Systemdatenbankverzeichnis

Für jedes Exemplar des Datenbankmanagers ist eine Datei mit einem *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.

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 ist 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 Datenbankverzeichnisses
- Einen Anzeiger, dass die Datenbank *indirekt* ist, d. h., dass sich die Datenbank im aktuellen Datenbankmanagerexemplar befindet
- Andere Systeminformationen

Auf UNIX<sup>®</sup>-Plattformen und in einer partitionierten Datenbankumgebung müssen Sie sicherstellen, dass alle Datenbankpartitionen jederzeit auf dieselbe Datei des Systemdatenbankverzeichnisses `sqlbdir` im Unterverzeichnis `sqlbdir` des Ausgangsverzeichnisses für das Exemplar zugreifen. Unvorhersehbare Fehler können auftreten, wenn entweder das Systemdatenbankverzeichnis oder die Systemintensionsdatei `sqlbins` im selben Unterverzeichnis `sqlbdir` symbolische Verbindungen zu einer anderen Datei sind, die sich in einem gemeinsam benutzten Dateisystem befindet.

### Zugehörige Tasks:

- „Aktivieren der Datenpartitionierung in einer Datenbank“ auf Seite 13
- „Katalogisieren einer Datenbank“ auf Seite 81

### Zugehörige Referenzen:

- „CREATE DATABASE Command“ in *Command Reference*

## Angeben eines alternativen Servers für eine Datenbank

Wenn ein Server ausfällt, empfängt jeder mit diesem Server verbundene Client einen Kommunikationsfehler, der die Verbindung beendet und einen Anwendungsfehler verursacht. In Fällen, in denen die Verfügbarkeit eine wichtige Rolle spielt, sollten Sie entweder eine redundante Konfiguration oder die Möglichkeit zur Übernahme der Serverfunktion durch einen Bereitschaftsknoten implementiert haben. In beiden Fällen versucht der Clientcode von DB2 Universal Database<sup>™</sup> (DB2 UDB) die Verbindung zu dem ursprünglichen Server, der jetzt auf einem Übernahmeknoten (die IP-Adresse wird ebenfalls übernommen) ausgeführt wird, oder zu einem neuen Server wiederherzustellen.

### Vorgehensweise:

Verwenden Sie den Befehl **UPDATE ALTERNATE SERVER FOR DATABASE**, um einen neuen bzw. alternativen Server zu definieren. Dieser Befehl aktualisiert die Informationen über den alternativen Server für einen Aliasnamen einer Datenbank im Systemdatenbankverzeichnis.

#### Zugehörige Konzepte:

- „Implementierung der automatischen Clientweiterleitung“ auf Seite 79
- „Automatische Clientweiterleitung - Beschreibung und Einrichtung“ auf Seite 335

## Anzeigen des Inhalts der Datei des lokalen oder des Systemdatenbankverzeichnisses

Vielleicht wollen Sie einige der Informationen zu den Datenbanken prüfen, die Sie auf Ihrem System haben.

#### Voraussetzungen:

Sie können den Inhalt der Dateien des lokalen bzw. des Systemdatenbankverzeichnisses erst anzeigen, wenn Sie ein Exemplar und eine Datenbank erstellt haben.

#### Vorgehensweise:

Wenn Sie den Inhalt der Datei des lokalen Datenbankverzeichnisses anzeigen wollen, geben Sie den folgenden Befehl ein, in dem <position> die Speicherposition der Datenbank bezeichnet:

```
LIST DATABASE DIRECTORY ON <position>
```

Wenn Sie den Inhalt der Datei des Systemdatenbankverzeichnisses anzeigen wollen, setzen Sie den Befehl **LIST DATABASE DIRECTORY** ohne Angabe der Speicherposition der Datei für das Datenbankverzeichnis ab.

#### Zugehörige Referenzen:

- „LIST DATABASE DIRECTORY Command“ in *Command Reference*

## 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 Datenbankclient 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<sup>®</sup>-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 zum Typ 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.

#### Zugehörige Referenzen:

- „CATALOG TCPIP NODE Command“ in *Command Reference*
- „LIST NODE DIRECTORY Command“ in *Command Reference*

- „CATALOG NETBIOS NODE Command“ in *Command Reference*
- „CATALOG LOCAL NODE Command“ in *Command Reference*
- „CATALOG NAMED PIPE NODE Command“ in *Command Reference*

---

## LDAP-Verzeichnisservice (Lightweight Directory Access Protocol)

Bei einem Verzeichnisservice handelt es sich um ein Repository mit Ressourceninformationen zu mehreren Systemen und Services innerhalb einer verteilten Umgebung. Er stellt den Client- und Serverzugriff auf diese Ressourcen bereit. Clients und Server verwenden den Verzeichnisservice in der Regel, um festzustellen, wie sie auf andere Ressourcen zugreifen können. Informationen zu diesen anderen Ressourcen in der verteilten Umgebung müssen in das Repository des Verzeichnisservice eingegeben werden.

LDAP (Lightweight Directory Access Protocol) ist eine Standardmethode zum Zugriff auf Verzeichnisservices. Jedes Datenbankserverexemplar veröffentlicht Informationen über seine Existenz auf einem LDAP-Server und stellt dem LDAP-Verzeichnis Datenbankinformationen zur Verfügung, 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. Clientanwendungen durchsuchen das LDAP-Verzeichnis nach den erforderlichen Informationen für die Herstellung der Verbindung zur Datenbank.

Als Administrator eines DB2<sup>®</sup> UDB-Systems können Sie einen Verzeichnisservice einrichten und verwalten. Zur Pflege dieses Verzeichnisservice können Sie den Konfigurationsassistenten oder die Steuerzentrale zu Hilfe nehmen. Der Verzeichnisservice wird für DB2 UDB durch LDAP-Verzeichnisservices (Lightweight Directory Access Protocol) verfügbar gemacht. Zur Verwendung von LDAP-Verzeichnisservices muss zunächst ein LDAP-Server vorhanden sein, der von DB2 unterstützt wird, so dass Verzeichnisisinformationen in ihm gespeichert werden können.

**Anmerkung:** In einer Domänenumgebung unter Windows<sup>®</sup> 2000 steht bereits ein LDAP-Server zur Verfügung, da er in Windows 2000 Active Directory integriert ist. Infolgedessen kann jede Maschine, die mit Windows 2000 arbeitet, LDAP verwenden.

Ein LDAP-Verzeichnis ist in einer Unternehmensumgebung nützlich, in der es aufgrund einer großen Anzahl von Clients schwierig ist, lokale Verzeichniskataloge auf jeder Clientmaschine zu aktualisieren. In einer solchen Situation wird empfohlen, die Verzeichniseinträge auf einem LDAP-Server zu speichern, so dass die Verwaltung von Katalogeinträgen an nur einer Stelle erfolgen kann: auf dem LDAP-Server. Die Kosten der Anschaffung und Pflege eines LDAP-Servers können beträchtlich sein, so dass ein solcher Server nur in Betracht kommt, wenn er durch eine ausreichend hohe Anzahl von Clients gerechtfertigt wird.

### Zugehörige Konzepte:

- „Erkennen (Discovery) von Verwaltungsservern, Exemplaren und Datenbanken“ auf Seite 63
- „Einführung in Lightweight Directory Access Protocol (LDAP)“ auf Seite 341

---

## Erstellen von Datenbankpartitionsgruppen (Knotengruppen)

Eine Datenbankpartitionsgruppe wird mit der Anweisung CREATE DATABASE PARTITION GROUP erstellt. Diese Anweisung gibt eine Gruppe von Datenbankpartitionen 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 Datenbankpartitionsgruppe.
- Sie generiert eine ID für die Partitionierungszuordnung.
- Sie fügt Datensätze in die folgenden Katalogtabellen ein:
  - SYSCAT.DBPARTITIONGROUPS
  - SYSCAT.PARTITIONMAPS
  - SYSCAT.DBPARTITIONGROUPDEF

### Voraussetzungen:

Die Maschinen und Systeme müssen verfügbar sein und eine Umgebung mit partitionierten Datenbanken verarbeiten können. Sie haben DB2 Universal Database Enterprise - Server Edition erworben und installiert. Die Datenbank muss vorhanden sein.

### Vorgehensweise:

Gehen Sie wie folgt vor, um eine Datenbankpartitionsgruppe über die Steuerzentrale zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Datenbankpartitionsgruppen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf den Ordner **Datenbankpartitionsgruppen** und wählen Sie im Kontextmenü **Erstellen** aus.
3. Geben Sie im Fenster **Datenbankpartitionsgruppen erstellen** die erforderlichen Informationen ein, versetzen Sie mit Hilfe der Pfeile Knoten aus dem Feld **Verfügbare Partitionen** in das Feld **Ausgewählte Partitionen** und klicken Sie auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine Datenbankpartitionsgruppe zu erstellen:

```
CREATE DATABASE PARTITION GROUP <name> ON PARTITIONS (<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 Datenbankpartitionsgruppe mit zwei Datenbankpartitionen (1 und 2) in einer Datenbank zu erstellen, die aus mindestens drei Datenbankpartitionen (0 bis 2) besteht:

```
CREATE DATABASE PARTITION GROUP mixng12 ON PARTITIONS (1,2)
```

Der Befehl **CREATE DATABASE** bzw. die API `sqlcrea()` erstellen auch die Standarddatenbankpartitionsgruppen des Systems **IBMDEFAULTGROUP**, **IBMCATGROUP** und **IBMTEMPGROUP**.

**Zugehörige Konzepte:**

- „Datenbankpartitionsgruppen“ in *Systemverwaltung: Konzept*
- „Partitionierungszuordnungen“ in *Systemverwaltung: Konzept*

**Zugehörige Referenzen:**

- „CREATE DATABASE PARTITION GROUP statement“ in *SQL Reference, Volume 2*
- „sqlcrea - Create Database“ in *Administrative API Reference*
- „CREATE DATABASE Command“ in *Command Reference*

---

## Definition des Datenbankwiederherstellungsprotokolls

Ein *Datenbankwiederherstellungsprotokoll* zeichnet Datensätze über alle an einer Datenbank vorgenommenen Änderungen auf, wozu auch das Hinzufügen neuer Tabellen oder das Aktualisieren vorhandener Tabellen gehören. 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.

**Zugehörige Konzepte:**

- „Wiederherstellungsprotokolle“ in *Datenwiederherstellung und hohe Verfügbarkeit Handbuch und Referenz*

---

## Implementierung der automatischen Clientweiterleitung

Wenn eine Clientanwendung von DB2<sup>®</sup> Universal Database (DB2 UDB) die Kommunikationsverbindung zu einem DB2 UDB-Server verliert, wäre es wünschenswert, wenn der Client in der Lage wäre, einen solchen Verlust wettzumachen, ohne dass Sie oder ein anderer Administrator eingreifen müssten. DB2 UDB unterstützt die Wiederherstellung der Clientverbindung zu einem DB2 UDB-Server. Vor einem Kommunikationsausfall muss jedoch ein alternativer Standort eingerichtet werden, zu dem die Clientverbindung hergestellt werden kann.

Der Befehl `UPDATE ALTERNATE SERVER FOR DATABASE` dient zur Definition der alternativen Serverposition in einer bestimmten Datenbank. Der alternative Hostname und die entsprechende Portnummer werden als Teil des Befehls angegeben. Die Position wird in der Systemdatenbankverzeichnisdatei auf dem Server gespeichert.

Wenn Sie die Position des alternativen Servers in einer bestimmten Datenbank auf dem Serverexemplar angegeben haben, werden die Daten über diese alternative Serverposition an den Client im Rahmen des Prozesses zum Verbindungsaufbau zurückgegeben. Wenn die Kommunikation zwischen dem Client und dem Server aus irgendeinem Grund verloren geht, versucht der DB2 UDB-Clientcode die Verbindung unter Verwendung der Daten über den alternativen Server wiederherzustellen. Der DB2 UDB-Client versucht, abwechselnd eine Verbindung zum

ursprünglichen Server und zum alternativen Server herzustellen. Der Zeitraum zwischen diesen Versuchen beginnt mit sehr kurzen Intervallen, die anschließend im Laufe der weiteren Versuche schrittweise verlängert werden.

Wenn eine Verbindung erfolgreich hergestellt werden kann, wird der SQLCODE-Wert -30108 zurückgegeben, um anzuzeigen, dass eine erneute Datenbankverbindung nach einem Kommunikationsausfall hergestellt wurde. Der Hostname bzw. die IP-Adresse und der Servicenname bzw. die Portnummer werden zurückgegeben. Der Clientcode gibt nur den Fehler des ursprünglichen Kommunikationsausfalls an die Anwendung zurück, wenn die erneute Herstellung der Clientkommunikation mit dem ursprünglichen oder dem alternativen Server nicht möglich ist.

**Zugehörige Konzepte:**

- „Automatische Clientweiterleitung - Beschreibung und Einrichtung“ auf Seite 335
- „Automatische Clientweiterleitung - Einschränkungen“ auf Seite 336

**Zugehörige Referenzen:**

- „Automatische Clientweiterleitung - Beispiele“ auf Seite 338

---

## 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 `bnd` im entsprechenden Verzeichnis `sql1ib`.

Durch das Binden (BIND) eines Dienstprogramms wird ein *Paket* (d. h. ein Objekt) erstellt, das alle Informationen enthält, die zur Verarbeitung bestimmter SQL-Anweisungen aus einer einzelnen Quelldatei benötigt werden.

**Anmerkung:** Wenn diese Dienstprogramme von einem Client ausgeführt werden sollen, müssen sie explizit gebunden werden.

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 `sql1ib`. In diesem Beispiel ist `sample` der Name der Datenbank.

**Zugehörige Tasks:**

- „Erstellen einer Datenbank“ auf Seite 69

**Zugehörige Referenzen:**

- „BIND Command“ in *Command Reference*



---

## 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** können Sie eine Datenbank mit einem anderen Aliasnamen oder auch einen Datenbankeintrag katalogisieren, der zuvor mit dem Befehl **UNCATALOG DATABASE** gelöscht wurde.

### Voraussetzungen:

Obwohl Datenbanken bei ihrer Erstellung automatisch katalogisiert werden, kann es erforderlich werden, eine Datenbank zu katalogisieren. Wenn dieser Fall eintritt, muss die Datenbank vorhanden sein.

### Vorgehensweise:

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 `INSTNC_C` hergestellt. Das Exemplar `instnc_c` muss bereits als lokaler Knoten katalogisiert sein, bevor dieser Befehl ausgeführt wird.

```
CATALOG DATABASE b as b_on_ic AT NODE instnc_c
```

**Anmerkung:** Der Befehl **CATALOG DATABASE** wird auch auf Clientknoten verwendet, um Datenbanken zu katalogisieren, die sich auf Datenbankservermaschinen befinden.

**Anmerkung:** Die Standardverzeichnisdateien, einschließlich des Datenbankverzeichnisses, werden im Hauptspeicher unter Verwendung des Konfigurationsparameters „Verzeichniscacheunterstützung (*dir\_cache*)“ zwischengespeichert. Wenn der Verzeichniscache aktiviert ist, wird eine Änderung an einem Verzeichnis (z. B. durch den Befehl **CATALOG DATABASE** oder **UNCATALOG DATABASE**) durch eine andere Anwendung möglicherweise erst wirksam, wenn Ihre Anwendung erneut gestartet wird. Um den Verzeichniscache, der von einer Sitzung des Befehlszeilenprozessors verwendet wird, zu aktualisieren, führen Sie den Befehl **db2 terminate** aus.

In einer partitionierten Datenbank wird in jeder Datenbankpartition ein Cache für Verzeichnisdateien erstellt.

Neben dem Cache auf Anwendungsebene wird auch ein Cache auf Datenbankmanagerebene für interne Suchfunktionen des Datenbankmanagers verwendet. Um diesen „gemeinsamen“ Cache zu aktualisieren, führen Sie die Befehle **db2stop** und **db2start** aus.

### Zugehörige Tasks:

- „Aktualisieren der Verzeichnisse mit Informationen über ferne Datenbankservermaschinen“ auf Seite 82

### Zugehörige Referenzen:

- „dir\_cache - Verzeichniscacheunterstützung“ in *Systemverwaltung: Optimierung*
- „CATALOG DATABASE Command“ in *Command Reference*
- „TERMINATE Command“ in *Command Reference*
- „UNCATALOG DATABASE Command“ in *Command Reference*

---

## Aktualisieren der Verzeichnisse mit Informationen über ferne Datenbankservermaschinen

### Vorgehensweise:

Sie können den „Assistenten: Datenbank hinzufügen“ des Interpreters des Konfigurationsassistenten (CA) verwenden, um Katalogeinträge zu erstellen. Wenn Sie über DB2 Application Development Client verfügen, können Sie außerdem ein Anwendungsprogramm zum Katalogisieren von Einträgen erstellen.

**Anmerkung:** Zum Katalogisieren einer Datenbank müssen Sie die Berechtigung SYSADM oder SYSCTRL haben. Ist dies nicht der Fall, muss der Konfigurationsparameter *catalog\_noauth* auf den Wert YES gesetzt sein.

Gehen Sie wie folgt vor, um die Verzeichnisse über den Befehlszeilenprozessor zu aktualisieren:

1. Verwenden Sie einen der folgenden Befehle, um das Knotenverzeichnis zu aktualisieren:

- Für einen Knoten mit APPC-Verbindung:

```
db2 CATALOG APPC NODE <knotenname>  
    REMOTE <symbolischer_zielname> SECURITY <sicherheitstyp>
```

Zum Beispiel:

```
db2 CATALOG APPC NODE DB2NODE REMOTE DB2CPIC SECURITY PROGRAM
```

- Für eine Datenbank unter DB2 Universal Database für z/OS und OS/390 Version 5.1 (oder spätere Version) oder DB2 Universal Database für AS/400 Version 4.2 (oder spätere Version) mit einer TCP/IP-Verbindung:

```
db2 CATALOG TCPIP NODE <knotenname>  
    REMOTE <hostname> oder <IP-adresse>  
    SERVER <servicename> oder <portnummer>  
    SECURITY <sicherheitstyp>
```

Zum Beispiel:

```
db2 CATALOG TCPIP NODE MVSIPNOD REMOTE MVSHOST SERVER DB2INSTC
```

Der Standardport für TCP/IP-Verbindungen unter DB2 für OS/390 und z/OS ist 446.

2. Wenn Sie mit DB2 Connect arbeiten, müssen Sie eine Aktualisierung des DCS-Verzeichnisses mit Hilfe des Befehls CATALOG DCS DATABASE in Betracht ziehen.

Wenn Sie ferne Clients haben, müssen Sie außerdem die Verzeichnisse auf jedem fernen Client aktualisieren.

**Zugehörige Konzepte:**

- „Werte des Systemdatenbankverzeichnisses“ in *DB2 Connect Benutzerhandbuch*
- „Werte des DCS-Verzeichnisses“ in *DB2 Connect Benutzerhandbuch*

**Zugehörige Referenzen:**

- „CATALOG DATABASE Command“ in *Command Reference*
- „CATALOG DCS DATABASE Command“ in *Command Reference*
- „CATALOG APPC NODE Command“ in *Command Reference*
- „CATALOG TCPIP NODE Command“ in *Command Reference*
- „CATALOG NETBIOS NODE Command“ in *Command Reference*
- „CATALOG APPN NODE Command“ in *Command Reference*

---

## Erstellen eines Tabellenbereichs

Tabellenbereiche stellen eine Beziehung zwischen den physischen Speichereinheiten, die von Ihrem Datenbanksystem verwendet werden, und den logischen Behältern bzw. Tabellen her, die zum Speichern von Daten verwendet werden.

**Voraussetzungen:**

Sie müssen die Einheiten- bzw. Dateinamen der Behälter kennen, auf die Sie bei der Erstellung Ihrer Tabellenbereiche verweisen wollen. Sie müssen wissen, wie viel Speicherplatz auf einer Einheit bzw. in einer Datei, die dem Tabellenbereich zugeordnet werden soll, zur Verfügung steht.

**Vorgehensweise:**

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.

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 den Ordner **Tabellenbereiche** und wählen Sie im Kontextmenü **Erstellen** —> **Tabellenbereich mit Assistent** aus.
3. Befolgen Sie die im Assistenten aufgeführten Schritte zur Durchführung der Task.

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 unter Windows 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 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.

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 Datenbankpartitionsgruppe zugeordnet werden. Die Standarddatenbankpartitionsgruppe IBMDEFAULTGROUP wird verwendet, wenn der folgende Parameter in der Anweisung nicht angegeben wird:

```
IN database_partitionsgruppenname
```

Mit der folgenden SQL-Anweisung werden ein DMS-Tabellenbereich auf einem auf UNIX-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 12.67
TRANSFERRATE 0.18
```

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 Datenbankpartitionsgruppe mit dem Namen ODDGROUP in einer partitionierten UNIX-Datenbank erstellt. Die Datenbankpartitionsgruppe ODDGROUP muss zuvor mit der Anweisung CREATE DATABASE PARTITION GROUP erstellt worden sein. Im vorliegenden Beispiel wird angenommen, dass die Datenbankpartitionsgruppe ODDGROUP aus den Datenbankpartitionen 1, 3 und 5 besteht. In allen Datenbankpartitionen ist die Einheit /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 IN ODDGROUP
MANAGED BY DATABASE
USING (DEVICE '/dev/HDISK0' 10000, DEVICE '/dev/n1hd01' 40000)
      ON DBPARTITIONNUM 1
      (DEVICE '/dev/HDISK0' 10000, DEVICE '/dev/n3hd03' 40000)
      ON DBPARTITIONNUM 3
      (DEVICE '/dev/HDISK0' 10000, DEVICE '/dev/n5hd05' 40000)
      ON DBPARTITIONNUM 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. Die aktuellen Standardwerte sind folgende:

- OVERHEAD 12.67 ms
- TRANSFERRATE 0.18 ms

DB2 UDB kann die Leistung sequenzieller E/A-Operationen durch die Verwendung des sequenziellen Vorabsezugriffs (Sequential Prefetching) wesentlich verbessern, da dieser mit parallelen E/A-Operationen arbeitet.

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
PAGE SIZE 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, Behälter in einem SMS-Tabellenbereich hinzuzufügen, zu löschen oder in der Größe zu ändern sowie die Einstellungen für die Parameter PREFETCHSIZE, OVERHEAD und TRANSFERRATE für einen Tabellenbereich zu modifizieren. Die Transaktion, die die Anweisung für den Tabellenbereich absetzt, sollte so bald wie möglich festgeschrieben werden, um Konkurrenzsituationen beim Zugriff auf den 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 PREFETCH-

SIZE 20 oder 30 sein. Sie sollten zur manuellen Einstellung des Wertes für PREFETCHSIZE bei der Erstellung eines Tabellenbereichs die folgende Gleichung verwenden:

$$\text{PREFETCHSIZE} = (\text{Anzahl Behälter}) \times (\text{Anzahl physischer Spindeln pro Behälter}) \times \text{EXTENTSIZE}$$

Sie sollten zudem in Betracht ziehen, den Wert für PREFETCHSIZE von DB2 UDB automatisch bestimmen zu lassen.

**Zugehörige Konzepte:**

- „Aufbau von Tabellenbereichen“ in *Systemverwaltung: Konzept*
- „Vom Betriebssystem verwalteter Speicherbereich (SMS)“ in *Systemverwaltung: Konzept*
- „Von der Datenbank verwalteter Speicherbereich (DMS)“ in *Systemverwaltung: Konzept*
- „Sequenzieller Vorabesezugriff“ in *Systemverwaltung: Optimierung*

**Zugehörige Tasks:**

- „Unterstützung großer Seiten in einer 64-Bit-Umgebung aktivieren (AIX)“ in *Systemverwaltung: Konzept*

**Zugehörige Referenzen:**

- „ALTER TABLESPACE statement“ in *SQL Reference, Volume 2*
- „CREATE TABLESPACE statement“ in *SQL Reference, Volume 2*

---

## Erstellen bestimmter Typen von Tabellenbereichen

Es gibt verschiedene Typen von Tabellenbereichen, die vom Datenbankmanager verwendet werden oder zur Verwendung durch Anwendungen und Benutzer vorgesehen sind.

### Erstellen eines Tabellenbereichs für temporäre Systemtabellen

Obwohl standardmäßig ein Tabellenbereich für temporäre Systemtabellen bei der Erstellung einer Datenbank erstellt wird, kann es sinnvoll sein, einen getrennten Tabellenbereich für Sortieroperationen des Systems zuzuordnen.

**Voraussetzungen:**

Die Behälter, die dem Tabellenbereich für temporäre Systemtabellen zugeordnet werden sollen, müssen vorhanden sein.

**Einschränkungen:**

Eine Datenbank muss immer über mindestens einen temporären Systemtabellenbereich verfügen, da temporäre Systemtabellen nur in solch einem Tabellenbereich gespeichert werden können.

**Vorgehensweise:**

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“.

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')
```

Sie sollten mindestens einen Tabellenbereich für jede Seitengröße haben.

Die einzige Datenbankpartitionsgruppe, die beim Erstellen eines temporären Systemtabellenbereichs angegeben werden kann, ist IBMTEMPGROUP.

#### Zugehörige Tasks:

- „Erstellen eines Tabellenbereichs für temporäre Benutzertabellen“ auf Seite 87

#### Zugehörige Referenzen:

- „CREATE TABLESPACE statement“ in *SQL Reference, Volume 2*

## Erstellen eines Tabellenbereichs für temporäre Benutzertabellen

Temporäre Benutzertabellenbereiche werden nicht standardmäßig bei der Erstellung einer Datenbank erstellt. Temporäre Tabellen können im Rahmen der Operationen, die Ihre Anwendungsprogramme an den Daten der Datenbank ausführen, erforderlich werden. Wenn dies der Fall ist, müssen Sie eine temporäre Benutzertabelle erstellen.

#### Vorgehensweise:

Ein temporärer Benutzertabellenbereich wird zum Speichern als temporär deklarierter 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 Datenbankpartitionsgruppen mit Ausnahme von IBMTEMPGROUP erstellt werden. Standardmäßig wird beim Erstellen eines temporären Benutzertabellenbereichs die Datenbankpartitionsgruppe IBMDEFAULTGROUP verwendet.

Mit der Anweisung DECLARE GLOBAL TEMPORARY TABLE werden als temporär deklarierte Tabellen für die Verwendung in einem temporären Benutzertabellenbereich definiert.

#### Zugehörige Tasks:

- „Erstellen einer benutzerdefinierten temporären Tabelle“ auf Seite 108

#### Zugehörige Referenzen:

- „CREATE TABLESPACE statement“ in *SQL Reference, Volume 2*
- „DECLARE GLOBAL TEMPORARY TABLE statement“ in *SQL Reference, Volume 2*

## Erstellen von Tabellenbereichen in Datenbankpartitionsgruppen

Wenn ein Tabellenbereich in einer Datenbankpartitionsgruppe mit mehreren Datenbankpartitionen erstellt wird, werden alle Tabellen innerhalb des Tabellenbereichs auf alle Datenbankpartitionen in der Datenbankpartitionsgruppe verteilt oder partitioniert. Der Tabellenbereich wird in einer Datenbankpartitionsgruppe erstellt. Wenn der Tabellenbereich in einer Datenbankpartitionsgruppe erstellt wurde, muss er dort verbleiben und kann nicht in eine andere Datenbankpartitionsgruppe versetzt werden. Für die Zuordnung eines Tabellenbereichs zu einer Datenbankpartitionsgruppe wird die Anweisung CREATE TABLESPACE verwendet.

### Zugehörige Referenzen:

- „CREATE TABLESPACE statement“ in *SQL Reference, Volume 2*

## Angeben eines direkten E/A-Zugriffs (Raw I/O)

Wenn Behälter zum Speichern von Daten verwendet werden, unterstützt DB2 Universal Database einen direkten Plattenzugriff (Raw I/O). Diese Art der Unterstützung gibt Ihnen die Möglichkeit, eine Einheit (Gerät) mit direktem Plattenzugriff (Raw) an ein DB2 Universal Database-System anzuschließen. (Ausnahme: Systeme mit dem Betriebssystem Windows 9x.)

### Voraussetzungen:

Sie müssen die Einheiten- bzw. Dateinamen der Behälter kennen, auf die Sie bei der Erstellung Ihrer Tabellenbereiche verweisen wollen. Sie müssen wissen, wie viel Speicherplatz auf einer Einheit bzw. in einer Datei, die dem Tabellenbereich zugeordnet werden soll, zur Verfügung steht.

Sie benötigen die richtigen Berechtigungen für den Schreib- und Lesezugriff auf den Behälter.

### Vorgehensweise:

Die folgende Liste zeigt die physischen und logischen Verfahren, eine Einheit mit einem direkten Zugriffstyp anzugeben:

- Unter Windows geben Sie ein physisches Festplattenlaufwerk mit folgender Syntax an:

```
\\.\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:

```
\\.\PhysicalDrive5
```

- Unter Windows geben Sie ein logisches Laufwerk (d. h. eine unformatierte Partition) mit folgender Syntax an:

```
\\.\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. Um die Einschränkung zu überwinden, die durch die Verwendung eines Buchstabens zur Angabe des Laufwerks gegeben ist, können Sie eine global eindeutige ID (GUID) für das logische Laufwerk verwenden.

- **Anmerkung:** Um auf eine Einheit (Gerät) schreiben zu können, müssen Sie Windows NT Version 4.0 mit installiertem Service Pack 3 oder einem späteren Service Pack haben.
- Auf Plattformen auf UNIX-Basis kann ein logischer Datenträger Benutzern und Anwendungen als einzelne, zusammenhängende und erweiterbarer Datenträger erscheinen. Trotzdem kann er sich auf nicht zusammenhängenden physischen Partitionen und sogar mehr als einem physischen Datenträger befinden. Der logische Datenträger muss außerdem in einer einzelnen Datenträgergruppe enthalten sein. Es besteht eine Begrenzung auf 256 logische Datenträger pro Datenträgergruppe. Es besteht eine Begrenzung auf 32 physische Datenträger pro Datenträgergruppe. Sie können mit Hilfe des Befehls **mklv** zusätzliche logische Datenträger erstellen. Mit Hilfe dieses Befehls können Sie den Namen des logischen Datenträgers angeben und seine Merkmale einschließlich der Anzahl und Position der logischen Partitionen definieren, die für ihn zugeordnet werden sollen.

Nachdem Sie einen logischen Datenträger erstellt haben, können Sie seinen Namen und seine Merkmale mit Hilfe des Befehls **chlv** ändern sowie die Anzahl der ihm zugeordneten logischen Partitionen mit dem Befehl **extendlv** ändern. Die standardmäßige Maximalgröße für einen logischen Datenträger bei der Erstellung sind 512 logische Partitionen sofern sie nicht größer angegeben wurde. Mit Hilfe des Befehls **chlv** kann diese Begrenzung außer Kraft gesetzt werden.

In AIX wird die Gruppe der Betriebssystembefehle, Bibliothekssubroutinen und anderen Tools, mit denen Sie logischen Datenträgerspeicher aufbauen und steuern können, Logical Volume Manager (LVM) genannt. LVM steuert Datenträgerressourcen, indem Daten zwischen einer einfacheren und flexiblen logischen Sicht des Speicherbereichs und den tatsächlichen physischen Datenträgern zugeordnet werden.

Weitere Informationen zu **mklv** und weiteren logischen Datenträgerbefehlen und LVM finden Sie im Handbuch *AIX 5L Version 5.2 System Management Concepts: Operating System and Devices*.

Für Windows 2000 und spätere Betriebssystemversionen ist eine neue Methode zur Angabe unformatierter DMS-Tabellenbereichsbehälter zu verwenden. Datenträgern (d. h. Partitionen auf Basisfestplatten) wird bei ihrer Erstellung eine global eindeutige ID (GUID) zugeordnet. Die GUID kann als Einheiten-ID bei der Angabe der Behälter in einer Tabellenbereichsdefinition verwendet werden. Die GUIDs sind über Systeme hinweg eindeutig. Das heißt, dass sie in einer Datenbankkonfiguration mit mehreren Partitionen für jede Partition unterschiedlich sind, selbst wenn die Definitionen der Festplattenpartitionen übereinstimmen.

Es steht ein Tool mit dem Namen *db2listvolumes.exe* (nur für Windows-Betriebssysteme) zur Verfügung, das die Anzeige der GUIDs für alle Plattendatenträger vereinfacht, die auf einem Windows-System definiert sind. Dieses Tool erstellt zwei Dateien in dem Verzeichnis, in dem es ausgeführt wird. Eine Datei mit dem Namen *volumes.xml* enthält in XML codierte Informationen über jeden Plattendatenträger, die einfach mit einem beliebigen XML-fähigen Browser angezeigt werden können. Die zweite Datei mit dem Namen *tablespace.ddl* enthält die erforderliche Syntax zur Angabe der Tabellenbereichsbehälter. Diese Datei muss aktualisiert werden, indem die noch ausstehenden Informationen eingefügt

werden, die für eine Tabellenbereichsdefinition erforderlich sind. Das Tool *db2listvolumes* benötigt keine Befehlszeilenargumente.

**Zugehörige Tasks:**

- „Einrichten eines direkten E/A-Zugriffs (Raw I/O) unter Linux“ auf Seite 90

## Einrichten eines direkten E/A-Zugriffs (Raw I/O) unter Linux

Wenn Behälter zum Speichern von Daten verwendet werden, unterstützt DB2 Universal Database einen direkten Plattenzugriff (Raw I/O). Diese Art der Unterstützung gibt Ihnen die Möglichkeit, eine Einheit (Gerät) mit direktem Plattenzugriff (Raw) an ein DB2 Universal Database-System anzuschließen. Für eine Linux-Umgebung sind jedoch bestimmte Informationen zu beachten.

**Voraussetzungen:**

Sie müssen die Einheiten- bzw. Dateinamen der Behälter kennen, auf die Sie bei der Erstellung Ihrer Tabellenbereiche verweisen wollen. Sie müssen wissen, wie viel Speicherplatz auf einer Einheit bzw. in einer Datei, die dem Tabellenbereich zugeordnet werden soll, zur Verfügung steht.

Bevor Sie die unformatierte Ein-/Ausgabe für Linux definieren, müssen folgende Voraussetzungen erfüllt sein:

- Mindestens eine freie IDE- oder SCSI-Plattenpartition
- 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_ctrl /dev/rawctl
```
- Das Dienstprogramm `raw`, das in der Regel mit der Linux-Variante zur Verfügung gestellt wird.

**Anmerkung:** Bei den Versionen, die momentan unformatierte Ein-/Ausgabe unterstützen, ist die Benennung der Knoten unformatierter Einheiten unterschiedlich:

*Tabelle 3. Linux-Varianten mit Unterstützung eines unformatierten (direkten) E/A-Zugriffs*

Variante	Knoten für unformatierte Einheiten	Controller für unformatierte Einheiten
RedHat oder TurboLinux	<code>/dev/raw/raw1</code> bis 255	<code>/dev/rawctl</code>
SuSE	<code>/dev/raw1</code> bis 63	<code>/dev/raw</code>

DB2 unterstützt beide der genannten Einheitencontroller sowie die meisten anderen Namen für Knoten unformatierter Einheiten. Unformatierte Einheiten werden von DB2 unter Linux/390 nicht unterstützt.

**Vorgehensweise:**

Linux verfügt über einen Pool von Knoten für unformatierte Einheiten, 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 Informationen zu den Bindungen 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.

Gehen Sie wie folgt vor, um eine 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
```

*Tabelle 4. Berechnungen für unformatierte E/A unter Linux*

Einheitenstart	Anfang	Ende	Blöcke	ID	System
<code>/dev/sda1</code>	1	523	4200997	83	Linux
<code>/dev/sda2</code>	524	1106	4682947+	5	Extended
<code>/dev/sda5</code>	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 geeignete Leseberechtigungen für den Controller für unformatierte Einheiten und die Plattenpartition fest. Legen Sie geeignete Lese- und Schreibberechtigungen für die unformatierte Einheit fest.

**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 auch für alle anderen Seitengrößen unterstützt, die von DB2 unterstützt werden.

#### Zugehörige Tasks:

- „Angabe eines direkten E/A-Zugriffs (Raw I/O)“ auf Seite 88

---

## Erstellen eines Schemas

Während Ihre Daten in Tabellen organisiert werden, kann es außerdem vorteilhaft sein, Tabellen und andere zugehörige Objekte zusammen zu gruppieren. Dies geschieht durch Definieren eines Schemas mit Hilfe der Anweisung `CREATE SCHEMA`. Informationen zu dem Schema werden in den Systemkatalogtabellen der Datenbank gespeichert, mit der Sie verbunden sind. Weitere Objekte können in diesem Schema angelegt werden.

### Voraussetzungen:

Die Datenbanktabellen und andere zugehörige Objekte, die zusammengruppiert werden sollen, müssen vorhanden sein.

### Einschränkungen:

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 Schemanamen erstellen, der nicht bereits existiert.

Wenn Benutzer nicht über die Berechtigung `IMPLICIT_SCHEMA` verfügen, können sie nur ein Schema erstellen, das den gleichen Namen wie ihre eigene Berechtigungs-ID hat.

Ein Zugriff ohne Angabe eines Schemas als Qualifikationsmerkmal auf Objekte in einem Schema ist nicht zulässig, da durch das Schema die Eindeutigkeit in der Datenbank sichergestellt wird. Angesichts der Möglichkeit, dass zwei Benutzer zwei Tabellen (oder andere Objekte) mit demselben Namen erstellen könnten, leuchtet dies ein. Ohne Schema, das für Eindeutigkeit sorgt, gäbe es Mehrdeutigkeiten, wenn ein dritter Benutzer versuchte, die Tabelle abzufragen. Ohne weitere Qualifikationsmerkmale ist es unmöglich, die zu verwendende Tabelle zu ermitteln.

Das neue Schema darf nicht bereits in den Systemkatalogen vorhanden sein und darf nicht mit der Zeichenfolge „SYS“ beginnen.

### Vorgehensweise:

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.

Der Benutzer, der durch die Anweisung `CREATE SCHEMA` erstellte Objekte definiert hat, ist der Eigner des Schemas. Dieser Eigner kann Zugriffsrechte auf Schemata anderen Benutzern erteilen (`GRANT`) und entziehen (`REVOKE`).

Um einem anderen Benutzer Zugriff auf eine Tabelle ohne Angabe eines Schemanamens als Teil des Qualifikationsmerkmals des Tabellennamens zu gewähren, muss für diesen Benutzer eine Sicht erstellt 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 ein Schema über die Steuerzentrale zu erstellen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Schema** in der Datenbank angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf den Ordner **Schema**, und wählen Sie dann **Erstellen** aus.
3. Geben Sie die Informationen für das neue Schema ein, und klicken Sie dann 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
```

#### Zugehörige Konzepte:

- „Gruppieren von Objekten nach Schema“ auf Seite 8
- „Überlegungen zur Berechtigung IMPLICIT\_SCHEMA“ auf Seite 257
- „Zugriffsrechte für Schemata“ auf Seite 257

#### Zugehörige Tasks:

- „Einstellen eines Schemas“ auf Seite 93

#### Zugehörige Referenzen:

- „CREATE SCHEMA statement“ in *SQL Reference, Volume 2*

---

## Details zum Erstellen von Schemata

Schemata dienen zur Organisation der Objekteigentumsrechte innerhalb der Datenbank.

### Einstellen eines Schemas

Wenn mehrere Schemata vorhanden sind, kann es wünschenswert sein, ein Schema als Standardschema zur Verwendung für Objektverweise festzulegen, die ohne Qualifikationsmerkmal in dynamischen SQL-Anweisungen auftreten, die aus einer bestimmten DB2-Verbindung heraus abgesetzt werden.

#### Vorgehensweise:

Die Festlegung eines Standardschemas erfolgt durch Einstellen des Sonderregisters CURRENT SCHEMA auf das Schema, das Sie als Standardschema verwenden wollen. Dieses Sonderregister kann von jedem beliebigen Benutzer eingestellt werden. Es ist keine Berechtigung erforderlich.

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, in der ein Verweis ohne Qualifikationsmerkmal auf ein Datenbankobjekt vorhanden ist) verwendet.

Der Anfangswert des Sonderregisters CURRENT SCHEMA entspricht der Berechtigungs-ID des Benutzers der aktuellen Sitzung.

**Zugehörige Konzepte:**

- „Schemas“ in *SQL Reference, Volume 1*

**Zugehörige Referenzen:**

- „SET SCHEMA statement“ in *SQL Reference, Volume 2*
- „Reserved schema names and reserved words“ in *SQL Reference, Volume 1*
- „CURRENT SCHEMA special register“ in *SQL Reference, Volume 1*

---

## Kapitel 4. Erstellen von Tabellen und anderen zugehörigen Tabellenobjekten

Dieses Kapitel beschreibt die Erstellung von Tabellen mit spezifischen Merkmalen bei der Implementierung eines Datenbankentwurfs.

---

### Erstellen und Füllen einer Tabelle

Tabellen sind das Hauptrepository (d. h. „Hauptlager“) von Daten in Datenbanken. Das Erstellen von Tabellen und die Eingabe von Daten zum Füllen der Tabellen gehören zur Erstellung einer neuen Datenbank.

#### Voraussetzungen:

Sie müssen sich die Zeit nehmen, die Tabellen, die Ihre Daten enthalten sollen, zu entwerfen und zu organisieren.

#### Vorgehensweise:

Wenn Sie festgelegt haben, wie Ihre 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.

Durch die Anweisung `CREATE TABLE` werden der Tabelle ein Name, der ein Bezeichner mit oder ohne Qualifikationsmerkmal 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 Datenbankpartitionsgruppe 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 Befehle `LOAD` oder `IMPORT`
- Das Dienstprogramm Autoloader (Programm für automatisches Laden), wenn Sie in einer Umgebung mit partitionierter Datenbank arbeiten

Daten können einer Tabelle hinzugefügt werden, ohne die Änderungen zu protokollieren. Mit der Klausel `NOT LOGGED INITIALLY` der Anweisung `CREATE TABLE` lässt sich verhindern, 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.

**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.

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.

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 eine Tabelle über die Steuerzentrale 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 den Ordner **Tabellen**, und wählen Sie dann **Erstellen** aus.
3. Befolgen Sie die im Assistenten aufgeführten Schritte, um die Tasks durchzuführen.

Geben Sie in der Befehlszeile Folgendes ein, um eine Tabelle zu erstellen:

```
CREATE TABLE <NAME>
  (<spaltenname> <datentyp> <>nullattribut>)
  IN <TABELLENBEREICHNAME>
```

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,
  FIRSTNME 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.

Mit Hilfe der Anweisung `DECLARE GLOBAL TEMPORARY TABLE` können Sie eine globale temporäre Tabelle deklarieren.

Sie können auch eine Tabelle erstellen, die auf Grundlage eines Abfrageergebnisses definiert ist. Dieser Typ von Tabelle wird als *gespeicherte Abfragetabelle* bezeichnet.

#### **Zugehörige Konzepte:**

- „Import - Übersicht“ in *Dienstprogramme für das Versetzen von Daten Handbuch und Referenz*
- „LOAD - Übersicht“ in *Dienstprogramme für das Versetzen von Daten Handbuch und Referenz*
- „Versetzen von Daten zwischen Plattformen - Hinweise zum Dateiformat“ in *Dienstprogramme für das Versetzen von Daten Handbuch und Referenz*
- „Benutzerdefinierter Datentyp (UDT)“ auf Seite 129

#### **Zugehörige Tasks:**

- „Erstellen einer gespeicherten Abfragetabelle“ auf Seite 136

#### **Zugehörige Referenzen:**

- „CREATE TABLE statement“ in *SQL Reference, Volume 2*
- „INSERT statement“ in *SQL Reference, Volume 2*
- „DECLARE GLOBAL TEMPORARY TABLE statement“ in *SQL Reference, Volume 2*
- „IMPORT Command“ in *Command Reference*
- „LOAD Command“ in *Command Reference*

---

## **Details zum Erstellen und Füllen einer Tabelle**

Alle Ihre Daten sind in Tabellen enthalten. Bei der Erstellung von Tabellen und der Anordnung von Daten in diesen Tabellen sind zahlreiche Gesichtspunkte zu beachten.

### **Einführung in die Speicherplatzkomprimierung für Tabellen**

Tabellen können in zwei Fällen mit geringerem Speicherplatzbedarf auf Datenträgern gespeichert werden:

- Wenn der Spaltenwert Null ist und die definierte und feste Speicherplatzgröße nicht reserviert wird.
- Wenn der Spaltenwert leicht ermittelt oder bestimmt werden kann (z. B. Standardwerte) und wenn der Wert bei der Formatierung von Datensätzen sowie bei der Spaltenextraktion für den Datenbankmanager verfügbar ist.

DB2<sup>®</sup> Universal Database (DB2 UDB) verfügt über ein optionales Datensatzformat, das diese Art von Speicherplatzeinsparungen ermöglicht. Speicherplatzeinsparungen lassen sich sowohl auf Tabellenebene als auch auf Spaltenebene realisieren.

**Zugehörige Konzepte:**

- „Speicherbedarf für Datenbankobjekte“ in *Systemverwaltung: Konzept*
- „Speicherplatzkomprimierung für neue Tabellen“ auf Seite 98
- „Speicherplatzkomprimierung für vorhandene Tabellen“ auf Seite 185

## Speicherplatzkomprimierung für neue Tabellen

Bei der Erstellung einer Tabelle können Sie mit der optionalen Klausel VALUE COMPRESSION angeben, dass die Tabelle das platzsparende Zeilenformat auf Tabellenebene und möglicherweise auf Spaltenebene verwendet.

Wenn die Klausel VALUE COMPRESSION verwendet wird, werden Nullwerte (NULL) und Daten der Länge 0, die Datentypen mit variabler Länge (VARCHAR, VARGRAPHICS, LONG VARCHAR, LONG VARGRAPHIC, BLOB, CLOB und DBCLOB) zugeordnet wurden, nicht auf der Platte gespeichert. Lediglich Werte für den mit diesen Datentypen verbundenen Systemaufwand belegen Plattenspeicherplatz.

Bei Verwendung der Klausel VALUE COMPRESSION kann auch die Option COMPRESS SYSTEM DEFAULT verwendet werden, um die Plattenspeichernutzung weiter zu verringern. Eingefügte oder aktualisierte Werte, die mit dem Systemstandardwert für den Datentyp ihrer Spalte übereinstimmen, beanspruchen nur minimalen Plattenspeicherplatz. Der Standardwert selbst wird nicht auf der Platte gespeichert. Alle numerischen Datentypen, Zeichenfolgetypen mit fester Länge und Grafikzeichenfolgen mit fester Länge für Spalten unterstützen die Option COMPRESS SYSTEM DEFAULT. Das heißt, dass Nullen und Leerzeichen komprimiert werden können.

**Zugehörige Referenzen:**

- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

## Überlegungen zu Spalten für große Objekte (LOB)

Bevor Sie eine Tabelle erstellen, die Spalten für große Objekte (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,
 FIRSTNME 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,
   FIRSTNME 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 angehängt werden, kommt es zu *Leistungseinbußen*, insbesondere wenn dabei die Größe der LOB-Werte erhöht wird. (Dies wird durch erforderliche Speicheranpassungen verursacht.)

Auf Plattformen, die die Zuordnung von Dateien mit freien Bereichen nicht unterstützen und auf denen 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<sup>®</sup>-Systemkataloge verwenden LOB-Spalten und belegen eventuell mehr Speicher als in früheren Versionen.

3. Wünschen Sie eine bessere Leistung für LOB-Spalten, einschließlich der 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.

**Zugehörige Konzepte:**

- „Speicherbedarf für LOB-Daten“ in *Systemverwaltung: Konzept*

**Zugehörige Referenzen:**

- „CREATE TABLE statement“ in *SQL Reference, Volume 2*
- „Large objects (LOBs)“ in *SQL Reference, Volume 1*

## Definieren von Integritätsbedingungen

In diesem Abschnitt wird die Definition von Integritätsbedingungen behandelt:

- „Definieren einer eindeutigen Integritätsbedingung“
- „Definieren referenzieller Integritätsbedingungen“ auf Seite 101
- „Definieren einer Prüfung auf Integritätsbedingung in Tabellen“ auf Seite 105
- „Definieren einer informativen Integritätsbedingung“ auf Seite 106.

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.

**Einschränkungen:**

Eine eindeutige Integritätsbedingung kann nicht in einer untergeordneten Tabelle definiert werden.

Es kann nur einen Primärschlüssel pro Tabelle geben.

**Vorgehensweise:**

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.

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. 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.

Der Primärindex sorgt zwingend dafür, 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.

#### **Zugehörige Konzepte:**

- „Keys“ in *SQL Reference, Volume 1*
- „Constraints“ in *SQL Reference, Volume 1*

#### **Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

## **Definieren referenzieller Integritätsbedingungen**

Referenzielle Integrität wird durch Hinzufügen referenzieller Integritätsbedingungen zu Tabellen- und Spaltendefinitionen implementiert. Wenn referenzielle Integritätsbedingungen für den Datenbankmanager definiert sind, werden Änderungen an Daten in den Tabellen und Spalten an der definierten Integritätsbedingung überprüft. Die Durchführung der angeforderten Aktion hängt von dem Ergebnis der Prüfung der Integritätsbedingung ab.

## Vorgehensweise:

Referenzielle Integritätsbedingungen werden mit der Klausel FOREIGN KEY und der Klausel REFERENCES in der Anweisung CREATE TABLE oder ALTER TABLE definiert. Es gibt Auswirkungen einer referenziellen Integritätsbedingung auf eine typisierte Tabelle bzw. auf eine übergeordnete Tabelle, die eine typisierte Tabelle ist, die Sie vor der Erstellung einer referenziellen Integritätsbedingung berücksichtigen sollten.

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.

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. Sie könnten außerdem die Anweisung CREATE SCHEMA verwenden, um die Tabellen EMPLOYEE und DEPARTMENT gleichzeitig zu erstellen.

#### **Zugehörige Konzepte:**

- „Klausel FOREIGN KEY“ auf Seite 103
- „Klausel REFERENCES“ auf Seite 104

#### **Zugehörige Tasks:**

- „Hinzufügen von Fremdschlüsseln“ auf Seite 195

#### **Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „CREATE SCHEMA statement“ in *SQL Reference, Volume 2*
- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

## **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 *Integritätsbedingungsname* 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.

#### **Zugehörige Tasks:**

- „Definieren einer eindeutigen Integritätsbedingung“ auf Seite 100
- „Definieren referenzieller Integritätsbedingungen“ auf Seite 101

#### **Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

## **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 Löschregel eingeschlossen. Im gezeigten Beispiel wird die Löschregel ON DELETE NO ACTION verwendet, die angibt, dass keine Abteilung gelöscht werden kann, wenn ihr Mitarbeiter zugeordnet sind. Andere Löschregeln sind ON DELETE CASCADE, ON DELETE SET NULL und ON DELETE RESTRICT.

#### **Zugehörige Konzepte:**

- „Klausel FOREIGN KEY“ auf Seite 103

#### **Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „CREATE TABLE statement“ in *SQL Reference, Volume 2*



## Auswirkungen auf Dienstprogrammoperationen

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 die folgende Anweisung 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 Objekttablette 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.

### Zugehörige Konzepte:

- „Import - Übersicht“ in *Dienstprogramme für das Versetzen von Daten Handbuch und Referenz*
- „LOAD - Übersicht“ in *Dienstprogramme für das Versetzen von Daten Handbuch und Referenz*
- „Überprüfen auf ungültige Integritätsbedingungen“ in *Dienstprogramme für das Versetzen von Daten Handbuch und Referenz*

### Zugehörige Referenzen:

- „SET INTEGRITY statement“ in *SQL Reference, Volume 2*

## Definieren einer Prüfung auf Integritätsbedingung 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. Wenn Prüfungen auf Integritätsbedingungen in Tabellen für den Datenbankmanager definiert sind, werden eingefügte oder aktualisierte Daten in den Tabellen an der definierten Integritätsbedingung überprüft. Die Durchführung der angeforderten Aktion hängt von dem Ergebnis der Prüfung der Integritätsbedingung ab.

### Vorgehensweise:

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.

#### **Zugehörige Konzepte:**

- „Constraints“ in *SQL Reference, Volume 1*

#### **Zugehörige Tasks:**

- „Hinzufügen einer Prüfung auf Integritätsbedingung in einer Tabelle“ auf Seite 196

#### **Zugehörige Referenzen:**

- „CREATE TABLE statement“ in *SQL Reference, Volume 2*
- „ALTER SERVER statement“ in *SQL Reference, Volume 2*

## **Definieren einer informativen Integritätsbedingung**

Eine *informative Integritätsbedingung* ist eine Regel, die vom SQL-Compiler verwendet werden kann, jedoch nicht durch den Datenbankmanager angewendet wird. Der SQL-Compiler durchläuft eine Phase des Umschreibens der Abfrage, die SQL-Anweisungen in Formen umsetzt, die optimiert werden und den Zugriffsplan für die angeforderten Daten verbessern können. Der Zweck der Integritätsbedingung besteht nicht darin, den Datenbankmanager zu zusätzlichen Datenprüfungen zu veranlassen, sondern darin, die Abfrageleistung zu verbessern.

#### **Vorgehensweise:**

Informative Integritätsbedingungen werden mit Hilfe der Anweisungen CREATE TABLE oder ALTER TABLE definiert. In diesen Anweisungen fügen Sie referenzielle Integritätsbedingungen bzw. Prüfungen auf Integritätsbedingungen in Tabellen

hinzu. Anschließend ordnen Sie Attribute für die Integritätsbedingungen zu, indem Sie angeben, ob der Datenbankmanager die Integritätsbedingung anwenden (ENFORCE) soll oder nicht und ob die Integritätsbedingung für die Abfrageoptimierung genutzt werden soll oder nicht.

**Zugehörige Konzepte:**

- „Constraints“ in *SQL Reference, Volume 1*
- „Der SQL-Compilerprozess“ in *Systemverwaltung: Optimierung*
- „Methoden zum Umschreiben von Abfragen und Beispiele“ in *Systemverwaltung: Optimierung*

**Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

## 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.

**Vorgehensweise:**

Beim Erstellen einer Tabelle, bei der bestimmte Ausdrücke und Vergleichselemente 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 zum Beispiel zwei Fälle, in denen die Auswertung von Ausdrücken aufwendig sein kann, wenn die Leistung wichtig ist:

1. Die Auswertung des Ausdrucks muss während einer Abfrage häufig 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 Umschreiben 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 potenziell besserer Verknüpfungsstrategien.

Das folgende Beispiel zeigt die Definition einer generierten Spalte in der Anweisung CREATE TABLE:

```
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 wie folgt umformuliert 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 wie folgt umformuliert 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 hinzugefügt.

#### Zugehörige Tasks:

- „Definieren einer generierten Spalte in einer vorhandenen Tabelle“ auf Seite 200

#### Zugehörige Referenzen:

- „CREATE INDEX statement“ in *SQL Reference, Volume 2*
- „CREATE TABLE statement“ in *SQL Reference, Volume 2*
- „SELECT statement“ in *SQL Reference, Volume 2*

## Erstellen einer benutzerdefinierten temporären Tabelle

Eine benutzerdefinierte temporäre Tabelle wird von Anwendungen benötigt, die Sie zur Bearbeitung von Daten in der Datenbank schreiben. Ergebnisse aus der Bearbeitung der Daten müssen temporär in einer Tabelle gespeichert werden.

#### Voraussetzungen:

Ein Tabellenbereich für temporäre Benutzertabellen muss vorhanden sein, bevor eine benutzerdefinierte temporäre Tabelle erstellt wird.

#### Einschränkungen:

Die Beschreibung dieser Tabelle wird im Systemkatalog nicht gespeichert, 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.

Eine benutzerdefinierte temporäre Tabelle bietet keine Unterstützung für die folgenden Elemente:

- LOB-Spalten (oder Spalten eines anderen Typs, die ebenfalls auf LOB basieren)
- Spalten mit benutzerdefinierten Datentypen
- LONG VARCHAR-Spalten
- DATALINK-Spalten

#### Vorgehensweise:

Eine temporäre Tabellen wird mit der Anweisung DECLARE GLOBAL TEMPORARY TABLE 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. Das folgende Beispiel zeigt, 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.

Wenn bei der Erstellung dieser Tabelle ROLLBACK oder ROLLBACK TO SAVEPOINT angegeben wird, können Sie entweder angeben, dass alle Zeilen in der Tabelle gelöscht werden sollen (DELETE ROWS, Standardwert), oder Sie können angeben, dass die Zeilen der Tabelle erhalten bleiben sollen (PRESERVE ROWS).

#### Zugehörige Tasks:

- „Erstellen eines Tabellenbereichs für temporäre Benutzertabellen“ auf Seite 87

#### Zugehörige Referenzen:

- „ROLLBACK statement“ in *SQL Reference, Volume 2*
- „SAVEPOINT statement“ in *SQL Reference, Volume 2*
- „DECLARE GLOBAL TEMPORARY TABLE statement“ in *SQL Reference, Volume 2*

## Definieren einer Identitätsspalte in einer neuen Tabelle

Eine *Identitätsspalte* ermöglicht DB2 das automatische Generieren eines eindeutigen numerischen Wertes für jede Zeile, die der Tabelle hinzugefügt wird. Beim Erstellen einer Tabelle, bei der bekannt ist, dass jede der Tabelle hinzugefügte Zeile eindeutig identifiziert werden muss, können Sie der Tabelle eine Identitätsspalte hin-

zufügen. Um einen eindeutigen numerischen Wert für jede Zeile sicherzustellen, die der Tabelle hinzugefügt wird, sollten Sie einen eindeutigen Index für die Identitätsspalte definieren oder die Spalte als Primärschlüssel deklarieren.

#### **Einschränkungen:**

Nach der Erstellung können Sie die Tabellenbeschreibung nicht mehr ändern, um eine Identitätsspalte einzufügen.

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 führen zu einer Fehlermeldung, wenn die Eindeutigkeit der Werte in der Identitätsspalte durch einen Primärschlüssel oder einen für die Identitätsspalte definierten eindeutigen Index überprüft wird.

#### **Vorgehensweise:**

Identitätsspalten können mit der Klausel AS IDENTITY der Anweisung CREATE TABLE angegeben werden.

Das folgende Beispiel zeigt die Definition einer Identitätsspalte in der Anweisung CREATE TABLE:

```
CREATE TABLE table (col1 INT,  
                    col2 DOUBLE,  
                    col3 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 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 erhält immer von DB2 generierte Werte. 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.

#### **Zugehörige Konzepte:**

- „Vergleich von IDENTITY-Spalten und Sequenzen“ auf Seite 112

#### **Zugehörige Referenzen:**

- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

## Erstellen einer Sequenz

Eine *Sequenz* ist ein Datenbankobjekt, das die automatische Generierung von Werten ermöglicht. Sequenzen eignen sich ideal für die Aufgabe der Generierung eindeutiger Schlüsselwerte. Anwendungen können Sequenzen verwenden, um mögliche Probleme in Bezug auf den gemeinsamen Zugriff und die Leistung infolge der Generierung eines eindeutigen Zählers außerhalb der Datenbank zu vermeiden.

### Einschränkungen:

Im Gegensatz zu einem Identitätsspaltenattribut ist eine Sequenz weder an eine bestimmte Tabellenspalte noch an eine eindeutige Tabellenspalte gebunden, die gleichzeitig die einzige Zugriffsmöglichkeit darstellt.

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.

Die Positionen, an denen NEXTVAL- oder PREVVAL-Ausdrücke verwendet werden können, unterliegen verschiedenen Einschränkungen.

### Vorgehensweise:

Eine Sequenz kann erstellt oder geändert werden, so dass mit einer der folgenden Methoden Werte generiert werden:

- Monoton erhöhen (INCREMENT) oder vermindern (DECREMENT) ohne Grenze
- Monoton erhöhen oder vermindern bis zu einer benutzerdefinierten Begrenzung oder einem benutzerdefinierten Stopp
- Monoton erhöhen oder vermindern bis zu einer benutzerdefinierten Begrenzung und zyklisch wieder zurück an den Anfang zum erneuten Start

Das folgende Beispiel zeigt das Erstellen eines Sequenzobjekts:

```
CREATE SEQUENCE order_seq
  START WITH 1
  INCREMENT BY 1
  NOMAXVALUE
  NOCYCLE
  CACHE 24
```

In diesem Beispiel heißt die Sequenz `order_seq`. Die Sequenz beginnt mit 1 und wird ohne Obergrenze 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 exakten 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 eine Wiederherstellung erforderlich ist.
- Werte können in einem Cache zwischengespeichert werden, wenn dies die Leistung verbessert. Durch das Vorabzuordnen und Speichern von Werten im Cache wird die synchrone E/A für das Protokoll verringert, wenn Werte für die Sequenz generiert werden. Im Fall eines Systemfehlers werden alle zwischengespeicherten Werte, die noch nicht festgeschrieben wurden, nicht verwendet und gelten als verloren. Der für CACHE angegebene Wert ist die maximale Anzahl der Sequenzwerte, die verloren gehen könnten.

Für eine Sequenz können zwei Ausdrücke verwendet werden.

Der Ausdruck PREVVAL gibt den zuletzt generierten Wert für die angegebene Sequenz für eine vorherige Anweisung innerhalb des aktuellen Anwendungsprozesses 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 und alle Exemplare des NEXTVAL-Ausdrucks geben den gleichen Wert für eine Zeile des Ergebnisses zurück.

Dieselbe Sequenznummer kann als eindeutiger Schlüsselwert in zwei separaten Tabellen verwendet werden, indem auf die Sequenznummer mit einem Ausdruck NEXTVAL für die erste Zeile und mit einem Ausdruck PREVVAL für alle weiteren Zeilen 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)
```

**Zugehörige Konzepte:**

- „Vergleich von IDENTITY-Spalten und Sequenzen“ auf Seite 112

**Zugehörige Referenzen:**

- „CREATE SEQUENCE statement“ in *SQL Reference, Volume 2*

## Vergleich von IDENTITY-Spalten und Sequenzen

Zwischen Identitätsspalten und Sequenzen gibt es sowohl Gemeinsamkeiten als auch Unterschiede. Die Merkmale von Identitätsspalten und Sequenzen können bei Analyse und Entwurf von Datenbanken und Anwendungen verwendet werden.

Eine Identitätsspalte hat folgende Merkmale:

- 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 generiert 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 folgende Merkmale:

- Ein Sequenzobjekt ist ein Datenbankobjekt, das an keine bestimmte 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 nicht alle Merkmale dieser beiden Elemente. Die vorgestellten Merkmale werden Sie jedoch dabei unterstützen, abhängig vom Datenbankentwurf und den Anwendungen, die auf die Datenbank zugreifen, das geeignete Element auszuwählen.

#### **Zugehörige Tasks:**

- „Definieren einer generierten Spalte in einer neuen Tabelle“ auf Seite 107
- „Erstellen einer Sequenz“ auf Seite 111
- „Definieren einer generierten Spalte in einer vorhandenen Tabelle“ auf Seite 200

## **Beispiele für Bereichsclustertabellen**

Die beiden folgenden Beispiele sind einfach und veranschaulichen die Vorgehensweise zum Erstellen einer Bereichsclustertabelle. Die Beispiele zeigen, wie Sie entweder eine einzelne Spalte oder mehrere Spalten als Schlüssel einer Tabelle einsetzen können. Darüber hinaus wird in den Beispielen erläutert, wie eine Tabelle erstellt werden kann, in der ein Datenüberlauf zulässig bzw. nicht zulässig ist.

Das erste Beispiel zeigt eine Bereichsclustertabelle, die zum Lokalisieren eines Kursteilnehmers über seine ID (STUDENT\_ID) verwendet wird. Jeder Kursteilnehmerdatensatz umfasst die folgenden Informationen:

- ID der Schule
- ID des Kurses
- Kursteilnehmernummer
- Kursteilnehmer-ID
- Vorname des Kursteilnehmers
- Nachname des Kursteilnehmers
- Durchschnittliche Leistung des Kursteilnehmers

Im vorliegenden Fall werden die Kursteilnehmerdatensätze nur auf der Basis der IDs (STUDENT\_ID) eingesetzt. Hierbei wird STUDENT\_ID zum Hinzufügen, Aktualisieren und Löschen von Kursteilnehmerdatensätzen verwendet.

**Anmerkung:** Weitere Indizes können zu einem späteren Zeitpunkt separat hinzugefügt werden. Im Rahmen des vorliegenden Beispiels wird die Struktur der Tabelle und die Art und Weise des Zugriffs auf die Tabellendaten jedoch während der Tabellenerstellung definiert.

Im Folgenden ist die Syntax aufgeführt, die für diese Tabelle erforderlich ist:

```
CREATE TABLE STUDENTS
(SCHOOL_ID      INT NOT NULL,
PROGRAM_ID     INT NOT NULL,
STUDENT_NUM    INT NOT NULL,
STUDENT_ID     INT NOT NULL,
FIRST_NAME     CHAR(30),
LAST_NAME      CHAR(30),
GPA            FLOAT)
ORGANIZE BY KEY SEQUENCE
(STUDENT_ID    STARTING FROM 1 ENDING AT 1000000)
ALLOW OVERFLOW
;
```

Die Größe der einzelnen Datensätze ergibt sich aus der Summe der zugehörigen Spalten. Im vorliegenden Fall wird ein 10 Byte großer Header + 4 + 4 + 4 + 4 + 30 + 30 + 8 + 3 (für Nullwertspalten) verwendet, wodurch sich eine Größe von 97 Byte ergibt. Mit einer Seitengröße von 4 KB (d. h. 4096 Byte) verbleiben unter Berücksichtigung des Systemaufwands 4038 Byte, also genug Speicherplatz für 42 Datensätze pro Seite. Wird von einer Anzahl von 1 Million Kursteilnehmerdatensätzen ausgegangen, dann ergibt sich für die Anzahl der erforderlichen Seiten der Wert 23809.5. Dieser wird ermittelt durch die Division von 1 Million durch 42 Datensätze pro Seite. Aufgerundet kann also von einem Wert von 23810 erforderlichen Seiten ausgegangen werden. Für die Tabellen werden vier und für die Bereichszuordnung werden weitere drei Seiten einkalkuliert. Daraus ergibt sich ein Wert von 23817 Seiten mit jeweils 4 KB Größe, die vorab zugeordnet werden müssen. (Bei der Bereichszuordnung wird davon ausgegangen, dass die Tabelle in einem einzigen Behälter gespeichert wird. Für jeden Behälter sollten drei Seiten verfügbar sein.)

Im zweiten Beispiel, das eine Variante des ersten Beispiels darstellt, wird ein Schulgremium behandelt. In diesem Schulgremium sind 200 Schulen vertreten, wobei jede 20 Schulungsräume mit einer Kapazität von 35 Kursteilnehmern umfasst. Dieses Schulgremium ist für maximal 140.000 Kursteilnehmer eingerichtet.

Im vorliegenden Fall soll mit Kursteilnehmerdatensätzen auf der Basis von drei Faktoren gearbeitet werden: Schul-ID (SCHOOL\_ID), Kurs-ID (CLASS\_ID) und Kursteilnehmernummer (STUDENT\_NUM). Jede dieser drei Spalten verfügt über eindeutige Werte und wird zusammen mit den anderen Spalten zum Hinzufügen, Aktualisieren und Löschen von Kursteilnehmerdatensätzen verwendet.

**Anmerkung:** Wie im vorherigen Beispiel können auch hier weitere Indizes zu einem späteren Zeitpunkt separat hinzugefügt werden.

Im Folgenden ist die Syntax aufgeführt, die für diese Tabelle erforderlich ist:

```
CREATE TABLE STUDENTS
(SCHOOL_ID      INT NOT NULL,
CLASS_ID       INT NOT NULL,
STUDENT_NUM    INT NOT NULL,
```

```

        STUDENT_ID    INT NOT NULL,
        FIRST_NAME    CHAR(30),
        LAST_NAME     CHAR(30),
        GPA           FLOAT)
    ORGANIZE BY KEY SEQUENCE
    (SCHOOL_ID        STARTING FROM 1 ENDING AT 200,
     CLASS_ID         STARTING FROM 1 ENDING AT 20,
     STUDENT_NUM      STARTING FROM 1 ENDING AT 35)
    DISALLOW OVERFLOW
;

```

In diesem Fall ist ein Überlauf nicht zulässig. Dies ist sinnvoll, da voraussichtlich eine Richtlinie des Schulgremiums existiert, die die Anzahl der Kursteilnehmer innerhalb eines Kurses beschränkt. Im vorliegenden Beispiel ist die Kursgröße auf maximal 35 Teilnehmer begrenzt. Wenn Sie diesen Faktor in Beziehung zu den jeweiligen örtlichen Beschränkungen durch die Anzahl der Kursräume und Schulen setzen, ist nachvollziehbar, warum ein Überlauf bei der Anzahl der Kursteilnehmer vom Schulgremium als unzulässig eingestuft wird.

Die Schulen verfügen möglicherweise über eine nicht einheitliche Anzahl von Kursräumen. In diesem Fall muss beim Festlegen des Bereichs für die Anzahl der Schulungsräume (mit Hilfe von CLASS\_ID) als Obergrenze die größte Anzahl an Schulungsräumen verwendet werden, wenn alle Schulen berücksichtigt werden. Dies hat zur Folge, dass einige kleinere Schulen (d. h. Schulen mit weniger Schulungsräumen als die größte Schule) Platz für Kursteilnehmerdatensätze haben, der möglicherweise nie verwendet wird (sofern die Kapazität der betreffenden Schule nicht z. B. durch mobile Schulungsräume aufgestockt wird).

Wenn Sie dieselbe Seitengröße von 4 KB und dieselbe Größe für die Kursteilnehmerdatensätze verwenden wie im vorherigen Beispiel, dann können 42 Datensätze pro Seite gespeichert werden. Bei 140.000 Kursteilnehmerdatensätzen werden also 3333,3 Seiten (bzw. aufgerundet 3334 Seiten) benötigt. Unter Berücksichtigung von zwei Seiten für Tabelleninformationen und drei Seiten für die Bereichszuordnung ergibt sich hierbei ein Wert von 3339 Seiten mit jeweils 4 KB Größe, die vorab zugeordnet werden müssen.

#### Zugehörige Referenzen:

- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

## Funktionsweise des SQL-Compilers bei Bereichsclustertabellen

Der SQL-Compiler verarbeitet Bereichsclustertabellen (RCT = Range-Clustered Table) in ähnlicher Weise wie reguläre Tabellen, die über eine sekundäre B+-Indexstruktur verfügen. Anstatt eine B+-Indexstruktur zu durchlaufen, um die Speicherposition oder die Satzkenung (RID = Record Identifier) eines Datensatzes festzustellen, wird in Bereichsclustertabellen eine funktionale Suchroutine eingesetzt, die auf der Basis der Datensatz-Schlüsselwerte und des Algorithmus aus der Bereichsdefinition arbeitet. Hierbei wird ähnlich vorgegangen wie bei einem Index, da ein Schlüsselwert verwendet wird, um die RID schnell abrufen zu können.

Wenn der beste Zugriffspfad zu den erforderlichen Daten festgestellt werden soll, verwendet der SQL-Compiler Statistikdaten, die zu den Tabellen gespeichert werden. Die Indexstatistikdaten werden während der Tabellensuche erfasst, wenn der Befehl RUNSTATS ausgegeben wird. Bei einer Bereichsclustertabelle wird die Tabelle selbst als reguläre Tabelle und der Index als funktionsbasierter Index aufgebaut.

Die Reihenfolge der Datensätze in der Tabelle kann nicht mit Sicherheit vorausbestimmt werden, wenn beim Erstellen einer Bereichsclustertabelle der Überlauf als zulässig definiert wird.

**Zugehörige Konzepte:**

- „Bereichsclustertabellen“ in *Systemverwaltung: Konzept*
- „Richtlinien zur Verwendung von Bereichsclustertabellen“ auf Seite 116

## Richtlinien zur Verwendung von Bereichsclustertabellen

Beim Arbeiten mit DB2<sup>®</sup> Universal Database und Bereichsclustertabellen sind die folgenden Richtlinien zu beachten:

- Beim Definieren des Bereichs der Schlüsselwerte ist der Mindestwert optional. Wird dieser Wert nicht angegeben, wird als Standardwert 1 verwendet. Negative Werte sind als Mindest- und Maximalwerte zulässig. Beim Arbeiten mit negativen Werten muss der Mindestwert explizit angegeben werden. Beispiel: ORGANIZE BY KEY SEQUENCE (F1 STARTING FROM -100 ENDING AT -10)
- Das Erstellen eines regulären Indexes für dieselben Schlüsselwerte, die zum Definieren der Bereichsclustertabelle verwendet wurden, ist nicht zulässig.
- Einige ALTER TABLE-Optionen können bei Bereichsclustertabellen nicht verwendet werden. Wenn die Option die physische Struktur der Tabelle nicht beeinflusst, ist die Option zulässig.
- Da beim Erstellen einer Bereichsclustertabelle der erforderliche Platten Speicherplatz vorab zugeordnet wird, muss dieser Speicherplatz verfügbar sein, da andernfalls die Tabellenerstellung fehlschlägt.

**Zugehörige Konzepte:**

- „Bereichsclustertabellen“ in *Systemverwaltung: Konzept*
- „Beispiele für Bereichsclustertabellen“ auf Seite 113

## Definieren von Dimensionen für eine Tabelle

Eine *Dimension* ist ein Clustering-Schlüssel für eine Tabelle. Für eine Tabelle können eine oder mehrere Dimensionen ausgewählt werden. Wenn Sie mehr als eine Dimension in einer Tabelle haben, wird diese Tabelle als Tabelle mit mehrdimensionalem Clustering (MDC - Multidimensional Clustering) betrachtet. Eine solche Tabelle wird mit Hilfe der Klausel ORGANIZE BY DIMENSIONS in der Anweisung CREATE TABLE erstellt.

**Einschränkungen:**

Die Gruppe von Spalten, die in der Klausel ORGANIZE BY [DIMENSIONS] angegeben wird, muss den Regeln für die Anweisung CREATE INDEX entsprechen. Die Spalten werden als Schlüssel behandelt, die zur Beibehaltung der physischen Reihenfolge der Daten im Speicher dienen.

**Vorgehensweise:**

Jede Dimension wird in der Anweisung CREATE TABLE durch die Klausel ORGANIZE BY [DIMENSIONS] und eine oder mehrere Spalten spezifiziert. Klammern in der Dimensionsliste dienen zur Gruppierung von Spalten, die einer einzigen Dimension zugeordnet werden sollen.

Die Daten werden physisch in Gruppen (so genannten Clustern) nach einer oder mehreren Dimensionen zusammengefasst und zwar jeweils nach so vielen Dimensionen, wie gleichzeitig angegeben werden. Die Daten werden in EXTENTSIZE großen Speicherbereichen bzw. „Blöcken“ entlang der Dimensionsgrenzen organisiert. Bei Abfragen von Daten mit Dimensionsvergleichselementen kann die Suche auf diejenigen Speicherbereiche der Tabelle eingeschränkt werden, die die betroffenen Dimensionswerte enthalten. Darüber hinaus kann für diese Datensuchen ein sehr effizienter Vorabsezugriff durchgeführt werden, da die EXTENTSIZE großen Speicherbereiche Gruppen sequenzieller Seiten auf dem Datenträger sind.

Während eine Tabelle mit einem einzelnen Clusterindex mit der Zeit die Clusterbildung einbüßen kann, je mehr der Speicherbereich in der Tabelle mit Daten gefüllt wird, ist eine Tabelle mit mehreren Dimensionen in der Lage, die Clusterbildung der Daten über alle Dimensionen automatisch und kontinuierlich beizubehalten. Infolgedessen erübrigt sich die Notwendigkeit einer Reorganisation der Tabelle, um die sequenzielle Reihenfolge der Daten wiederherzustellen.

Für jede angegebene Dimension wird automatisch ein Dimensionsblockindex erstellt. Der Dimensionsblockindex ermöglicht den Zugriff auf Daten entlang einer Dimension. Ein Dimensionsblockindex zeigt auf EXTENTSIZE-Speicherbereiche und nicht auf einzelne Zeilen, so dass Dimensionsblockindizes wesentlich kleiner als reguläre Indizes sind. Solche Dimensionsblockindizes können verwendet werden, um rasch nur auf diejenigen Speicherbereiche der Tabelle zuzugreifen, die bestimmte Dimensionswerte enthalten.

Ein zusammengesetzter Blockindex, der alle Dimensionsschlüsselspalten enthält, wird automatisch erstellt. Der zusammengesetzte Blockindex dient zur Beibehaltung der Clusterbildung der Daten bei Einfüge- und Aktualisierungsaktivitäten. Der zusammengesetzte Blockindex wird bei der Abfrageverarbeitung dazu verwendet, auf Daten in der Tabelle zuzugreifen, die bestimmte Dimensionswerte besitzen.

**Anmerkung:** Die Reihenfolge der Schlüsselbestandteile im zusammengesetzten Blockindex kann seine Verwendung oder Anwendbarkeit bei der Abfrageverarbeitung beeinflussen. Die Reihenfolge der Schlüsselbestandteile wird durch die Reihenfolge der Spalten festgelegt, die in der gesamten zur Erstellung der MDC-Tabelle verwendeten Klausel ORGANIZE BY [DIMENSIONS] vorgefunden wird. Eine Tabelle wird zum Beispiel mit folgender Anweisung erstellt:

```
CREATE TABLE t1 (c1 int, c2 int, c3 int, c4 int)
  ORGANIZE BY DIMENSIONS (c1, c4, (c3,c1), c2)
```

In diesem Fall wird der zusammengesetzte Blockindex für die Spalten (c1,c4,c3,c2) erstellt. Die Spalte c1 wird zwar zweimal in der Dimensionsklausel angegeben, jedoch nur einmal als Schlüsselbestandteil für den zusammengesetzten Blockindex und dies in der Reihenfolge ihres ersten Auftretens verwendet. Die Reihenfolge der Schlüsselbestandteile im zusammengesetzten Blockindex macht für die Verarbeitung von Einfügeoperationen keinen Unterschied, kann sich hingegen auf die Verarbeitung von Abfragen auswirken. Wenn es daher eher wünschenswert ist, den zusammengesetzten Blockindex mit der Spaltenfolge (c1,c2,c3,c4) zu haben, sollte die Tabelle mit folgender Anweisung erstellt werden:

```
CREATE TABLE t1 (c1 int, c2 int, c3 int, c4 int)
  ORGANIZE BY DIMENSIONS (c1, c2, (c3,c1), c4)
```

Ein zusammengesetzter Blockindex wird nicht erstellt, wenn eine angegebene Dimension bereits alle Spalten enthält, die auch der zusammengesetzte Blockindex enthielte. Zum Beispiel würde für die folgende Tabelle kein zusammengesetzter Blockindex erstellt:

```
CREATE TABLE t1 (c1 int, c2 int)
  ORGANIZE BY DIMENSIONS (c1,(c2,c1))
```

**Zugehörige Konzepte:**

- „Tabellen mit mehrdimensionalem Clustering“ in *Systemverwaltung: Konzept*

**Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

## Erstellen einer Hierarchietabelle oder einer typisierten Tabelle

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.

Bei der Einrichtung einer strukturierten Typenhierarchie erstellen Sie typisierte Tabellen. Typisierte Tabellen dienen zum Speichern von Objektexemplaren, deren Merkmale mit der Anweisung CREATE TYPE definiert werden.

**Voraussetzungen:**

Der Typ, auf dessen Basis die Hierarchietabelle oder die typisierte Tabelle erstellt wird, muss vorhanden sein.

**Vorgehensweise:**

Sie können eine Hierarchietabelle oder typisierte Tabelle mit einer Variante der Anweisung CREATE TABLE erstellen.

**Zugehörige Konzepte:**

- „Typed tables“ in *Application Development Guide: Programming Server Applications*

**Zugehörige Tasks:**

- „Füllen einer typisierten Tabelle“ auf Seite 119
- „Erstellen einer typisierten Sicht“ auf Seite 136
- „Creating a structured type hierarchy“ in *Application Development Guide: Programming Server Applications*
- „Dropping typed tables“ in *Application Development Guide: Programming Server Applications*
- „Creating typed tables“ in *Application Development Guide: Programming Server Applications*

**Zugehörige Referenzen:**

- „CREATE TABLE statement“ in *SQL Reference, Volume 2*
- „CREATE TYPE (Structured) statement“ in *SQL Reference, Volume 2*

## Füllen einer typisierten Tabelle

Bei der Einrichtung einer strukturierten Typenhierarchie erstellen Sie typisierte Tabellen. Typisierte Tabellen dienen zum Speichern von Objekten, deren Merkmale mit der Anweisung CREATE TYPE definiert werden. Nach der Erstellung müssen Sie die typisierte Tabelle mit Daten füllen.

### Voraussetzungen:

Die typisierte Tabelle muss vorhanden sein.

### Vorgehensweise:

Sie können eine typisierte Tabelle mit Daten füllen, nachdem Sie die strukturierten Typen und anschließend die entsprechenden Tabellen und untergeordneten Tabellen erstellt haben.

### Zugehörige Konzepte:

- „Substitutability in typed tables“ in *Application Development Guide: Programming Server Applications*
- „Typed tables“ in *Application Development Guide: Programming Server Applications*

### Zugehörige Tasks:

- „Erstellen einer Hierarchietabelle oder einer typisierten Tabelle“ auf Seite 118
- „Storing objects in typed table rows“ in *Application Development Guide: Programming Server Applications*
- „Dropping typed tables“ in *Application Development Guide: Programming Server Applications*
- „Creating typed tables“ in *Application Development Guide: Programming Server Applications*

### Zugehörige Referenzen:

- „CREATE TYPE (Structured) statement“ in *SQL Reference, Volume 2*

## 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.

### Voraussetzungen:

Alle Tabellenbereiche müssen vor der Ausführung der Anweisung CREATE TABLE vorhanden sein.

### Einschränkungen:

Die Trennung der Tabellenkomponenten ist nur bei DMS-Tabellenbereichen möglich.

### Vorgehensweise:

Gehen Sie wie folgt vor, um eine Tabelle über die 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 den Ordner **Tabellen**, und wählen Sie dann im Kontextmenü **Erstellen** aus.
3. Geben Sie den Tabellennamen ein, und klicken Sie 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 auf **Fertig stellen**.

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.

### Zugehörige Referenzen:

- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

## Erstellen einer Tabelle in einer partitionierten Datenbank

Durch die Erstellung einer Tabelle in mehreren Partitionen einer partitionierten Datenbank lassen sich Leistungsvorteile erzielen. Die Arbeitslast, die mit dem Abruf von Daten verbunden ist, kann auf die Datenbankpartitionen verteilt werden.

### Voraussetzungen:

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 Datenbankpartitionsgruppe 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 Datenbankpartitionsgruppe zugeordnet ist.

### Einschränkungen:

Sie müssen bei der Auswahl eines geeigneten Partitionierungsschlüssels mit großer Sorgfalt vorgehen, 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).

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.

### Vorgehensweise:

Die Erstellung einer Tabelle, die zu mehreren Datenbankpartitionen gehören soll, wird bei der Erstellung der Tabelle angegeben. Bei der Erstellung einer Tabelle in einer partitionierten Datenbankumgebung gibt es eine zusätzliche Option: 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.

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 Datenbankpartitionsgruppen 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 heißen 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.

#### Zugehörige Konzepte:

- „Datenbankpartitionsgruppen“ in *Systemverwaltung: Konzept*
- „Aufbau von Datenbankpartitionsgruppen“ in *Systemverwaltung: Konzept*
- „Tabellenkollokation“ in *Systemverwaltung: Konzept*

#### Zugehörige Referenzen:

- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

---

## 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 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 geschäftsinernen Abläufe oder Regeln ändern, müssen lediglich die Auslöser und nicht sämtliche Anwendungsprogramme geändert werden.

### Einschränkungen:

Auslöser können nicht mit Kurznamen verwendet werden.

Wenn es sich bei dem Auslöser um einen Vorauslöser (BEFORE) handelt, darf der von der ausgelösten Aktion angegebene Spaltenname keine generierte Spalte außer einer Identitätsspalte bezeichnen. Dies bedeutet, dass der generierte Identitätswert für Vorauslöser sichtbar ist.

Beim Erstellen eines ganzheitlichen Auslösers (ATOMIC) ist Vorsicht beim Zeichen für das Anweisungsende geboten. Der Datenbankmanager erkennt standardmäßig ein Semikolon (;) als Markierung für das Anweisungsende. Sie sollten das Zeichen für das Anweisungsende manuell in der Prozedur für die Erstellung des ganzheitlichen Auslösers editieren, um ein anderes Zeichen als das Semikolon (;) zu verwenden. Das Semikolon könnte zum Beispiel durch ein anderes Sonderzeichen wie das Nummernzeichen (#) ersetzt werden.

Anschließend müssen Sie einen der folgenden Schritte ausführen:

- Wählen Sie im Befehlseditor (der die Befehlszentrale ersetzt) 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>
```

Dabei ist der Begrenzer das alternative Zeichen für das Anweisungsende und <prozedur> die modifizierte Prozedur mit dem neuen Begrenzer.

### Vorgehensweise:

Gehen Sie wie folgt vor, um einen Auslöser über die Steuerzentrale 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 den Ordner **Auslöser** und wählen Sie 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 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
UPDATE COMPANY_STATS SET NBEMP = NBEMP+1;

```

Der Hauptteil einer 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.

#### Zugehörige Konzepte:

- „Auslöserabhängigkeiten“ auf Seite 124
- „INSERT, UPDATE, and DELETE triggers“ in *Application Development Guide: Programming Server Applications*
- „Triggers in application development“ in *Application Development Guide: Programming Server Applications*
- „Trigger creation guidelines“ in *Application Development Guide: Programming Server Applications*
- „Verwenden von Auslösern zum Aktualisieren von Sichtinhalten“ auf Seite 125

#### Zugehörige Tasks:

- „Löschen eines Auslösers“ auf Seite 214
- „Creating triggers“ in *Application Development Guide: Programming Server Applications*
- „Defining business rules using triggers“ in *Application Development Guide: Programming Server Applications*
- „Defining actions using triggers“ in *Application Development Guide: Programming Server Applications*

#### Zugehörige Referenzen:

- „CREATE TRIGGER statement“ in *SQL Reference, Volume 2*

---

## 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 in den Informationen zur Anweisung DROP detailliert erläutert.

Wenn eines dieser Objekte gelöscht wird, wird der Auslöser außer Funktion gesetzt, während die Definition jedoch im Katalog erhalten bleibt. 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.

#### Zugehörige Konzepte:

- „Verwenden von Auslösern zum Aktualisieren von Sichtinhalten“ auf Seite 125

#### Zugehörige Tasks:

- „Erstellen eines Auslösers“ auf Seite 122
- „Löschen eines Auslösers“ auf Seite 214

#### Zugehörige Referenzen:

- „CREATE TRIGGER statement“ in *SQL Reference, Volume 2*
- „DROP statement“ in *SQL Reference, Volume 2*

---

## Verwenden von Auslösern zum Aktualisieren von Sichtinhalten

INSTEAD OF-Auslöser können verwendet werden, um eine Lösch-, Einfüge- oder Aktualisierungsanforderung für eine Sicht durchzuführen, die an sich nicht aktualisierbar ist. Anwendungen, die diesen Typ von Auslöser nutzen, können Aktualisierungsoperationen für Sichten enthalten, so als wäre die Sicht eine Tabelle.

Zum Beispiel könnten Sie eine Sicht mit folgenden SQL-Anweisungen erstellen:

```
CREATE VIEW EMPV(EMPNO, FIRSTNME, MIDINIT, LASTNAME, PHONENO, HIREDATE,
DEPTNAME)
AS SELECT EMPNO, FIRSTNME, MIDINIT, LASTNAME, PHONENO, HIREDATE, DEPTNAME
FROM EMPLOYEE, DEPARTMENT WHERE EMPLOYEE.WORKDEPT = DEPARTMENT.DEPTNO
```

Aufgrund der Verknüpfung im Hauptteil der Definition der Sicht EMPV wäre es nicht möglich, die Sicht zur Aktualisierung von Daten in den zugrunde liegenden Tabellen zu verwenden, wenn nicht die folgenden Anweisungen hinzugefügt würden:

```
| CREATE TRIGGER EMPV_INSERT INSTEAD OF INSERT ON EMPV
| REFERENCING NEW AS NEWEMP DEFAULTS NULL FOR EACH ROW
| INSERT INTO EMPLOYEE (EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT,
| PHONENO, HIREDATE)
| VALUES(EMPNO, FIRSTNME, MIDINIT, LASTNAME,
| COALESCE((SELECT DEPTNO FROM DEPARTMENT AS D
| WHERE D.DEPTNAME = NEWEMP.DEPTNAME),
| RAISE_ERROR('70001', 'Unknown department name')),
| PHONENO, HIREDATE)
```

Diese Anweisung CREATE TRIGGER ermöglicht die Ausführung von INSERT-Anforderungen für die Sicht EMPV.

```
| CREATE TRIGGER EMPV_DELETE INSTEAD OF DELETE ON EMPV
| REFERENCING OLD AS OLDEMP FOR EACH ROW
| DELETE FROM EMPLOYEE AS E WHERE E.EMPNO = OLDEMP.EMPNO
```

Diese Anweisung CREATE TRIGGER ermöglicht die Ausführung von DELETE-Anforderungen für die Sicht EMPV.

```
| CREATE TRIGGER EMPV_UPDATE INSTEAD OF UPDATE ON EMPV
| REFERENCING NEW AS NEWEMP
| OLD AS OLDEMP
| DEFAULTS NULL FOR EACH ROW
| BEGIN ATOMIC
```

```

VALUES(CASE WHEN NEWEMP.EMPNO = OLDEMP.EMPNO THEN 0
        ELSE RAISE_ERROR('70002', 'Must not change EMPNO') END);
UPDATE EMPLOYEE AS E
  SET (FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE) =
      (NEWEMP.FIRSTNME, NEWEMP.MIDINIT, NEWEMP.LASTNAME,
       COALESCE((SELECT DEPTNO FROM DEPARTMENT AS D
                  WHERE D.DEPTNAME = NEWEMP.DEPTNAME),
                RAISE_ERROR('70001', 'Unknown department name')),
       NEWEMP.PHONENO, NEWEMP.HIREDATE)
  WHERE NEWEMP.EMPNO = E.EMPNO;
END

```

Diese Anweisung CREATE TRIGGER ermöglicht die Ausführung von UPDATE-Anforderungen für die Sicht EMPV.

#### Zugehörige Tasks:

- „Erstellen eines Auslösers“ auf Seite 122

#### Zugehörige Referenzen:

- „CREATE TRIGGER statement“ in *SQL Reference, Volume 2*

---

## 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) zurück. Diese Art der Funktion wird in DB2<sup>®</sup> 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.

#### Tabelle

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 E-Mail-Nachrichten. 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.

#### **Zugehörige Konzepte:**

- „Scalar functions“ in *SQL Reference, Volume 1*
- „User-defined functions“ in *SQL Reference, Volume 1*
- „Table functions“ in *SQL Reference, Volume 1*
- „Erstellen einer Funktionszuordnung“ auf Seite 127
- „Statistiken für benutzerdefinierte Funktionen“ in *Systemverwaltung: Optimierung*
- „Allgemeine Regeln zum manuellen Aktualisieren von Katalogstatistiken“ in *Systemverwaltung: Optimierung*
- „DB2 User-Defined Functions and Methods“ in *Application Development Guide: Programming Client Applications*

#### **Zugehörige Tasks:**

- „Erstellen einer Funktionsschablone“ auf Seite 128

#### **Zugehörige Referenzen:**

- „Functions“ in *SQL Reference, Volume 1*
- „CREATE FUNCTION statement“ in *SQL Reference, Volume 2*

---

## **Details zum Erstellen einer benutzerdefinierten Funktion (UDF) oder Methode**

Diese folgenden Abschnitte enthalten Informationen zu Gesichtspunkten für zusammengesessene Datenbanken, die bei der Erstellung benutzerdefinierter Funktionen oder Methoden zu beachten sind.

### **Erstellen einer Funktionszuordnung**

Erstellen Sie in einer zusammengesessenen Datenbank eine Funktionszuordnung, wenn Sie eine lokale Funktion oder Funktionsschablone in einer oder mehreren Datenquellen einer Funktion zuordnen wollen. Für viele Datenquellenfunktionen werden Standardfunktionszuordnungen zur Verfügung gestellt.

Funktionszuordnungen sind in folgenden Situationen nützlich:

- Neue integrierte Funktionen werden in einer Datenquelle verfügbar.
- Sie müssen eine benutzerdefinierte Funktion in einer Datenquelle einer lokalen Funktion zuordnen.
- Eine Anwendung erfordert eine andere als die durch die Standardzuordnung bereitgestellte Standardarbeitsweise.

Funktionszuordnungen, die mit Anweisungen CREATE FUNCTION MAPPING definiert wurden, werden in der zusammengeschlossenen Datenbank gespeichert.

Funktionen (oder Funktionsschablonen) müssen die gleiche Anzahl von Eingabeparametern wie die Datenquellenfunktion haben. Außerdem sollten die Datentypen der Eingabeparameter auf der Seite der zusammengeschlossenen Datenbank mit den Datentypen der Eingabeparameter auf der Datenquellenseite kompatibel sein. Diese Voraussetzungen gelten auch für die Rückgabewerte.

Verwenden Sie zur Erstellung einer Funktionszuordnung die Anweisung CREATE FUNCTION MAPPING. Geben Sie beispielsweise Folgendes ein, um eine Funktionszuordnung zwischen einer Oracle-Funktion AVGNEW und einer äquivalenten DB2<sup>®</sup>-Funktion 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-Hostvariablen und ruft keine Ergebnisse der Typen LOB, LONG VARCHAR/VARGRAPHIC und DATALINK sowie einzigartiger und strukturierter Datentypen ab. Es kann keine Funktionszuordnung erstellt werden, wenn ein Eingabeparameter oder der Rückgabewert einen dieser Typen enthält.

#### **Zugehörige Konzepte:**

- „Host language program mappings with transform functions“ in *Application Development Guide: Programming Server Applications*

#### **Zugehörige Tasks:**

- „Erstellen einer Funktionsschablone“ auf Seite 128

#### **Zugehörige Referenzen:**

- „CREATE FUNCTION MAPPING statement“ in *SQL Reference, Volume 2*

## **Erstellen einer Funktionsschablone**

In Systemen zusammengeschlossener Datenbanken dienen Funktionsschablonen als „Anker“ für Funktionszuordnungen. Sie ermöglichen das Zuordnen einer Datenquellenfunktion, wenn auf dem Server einer zusammengeschlossenen 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.

#### **Einschränkungen:**

Es ist keine lokale ausführbare Datei für die Funktion vorhanden, so dass es möglich ist, 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 die gleiche Anzahl von Eingabeparametern wie die Datenquellenfunktion haben. Die Datentypen der Eingabeparameter auf der Seite der zusammengeschlossenen Datenbank sollten mit den Datentypen der Eingabeparameter auf der Datenquellenseite kompatibel sein. Diese Voraussetzungen gelten auch für die Rückgabewerte.

#### **Vorgehensweise:**

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')
```

#### **Zugehörige Konzepte:**

- „Erstellen einer Funktionszuordnung“ auf Seite 127

#### **Zugehörige Referenzen:**

- „CREATE FUNCTION (Sourced or Template) statement“ in *SQL Reference, Volume 2*

---

## **Benutzerdefinierter Datentyp (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 eine strenge Typentrennung, was bedeutet, dass sie zwar die gleiche Darstellung wie andere Typen verwenden, Werte eines bestimmten UDTs jedoch nur mit Werten des bzw. der gleichen UDTs in der gleichen Typenhierarchie als kompatibel betrachtet werden.

Über die Katalogsicht SYSCAT.DATATYPES können die benutzerdefinierten Datentypen angezeigt werden, die für die Datenbank definiert wurden. Diese Katalogsicht zeigt darüber hinaus die Datentypen, die bei der Erstellung der Datenbank vom Datenbankmanager definiert wurden.

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.

**Zugehörige Konzepte:**

- „Erstellen eines benutzerdefinierten strukturierten Typs“ auf Seite 132

**Zugehörige Tasks:**

- „Erstellen eines benutzerdefinierten einzigartigen Datentyps“ auf Seite 130

**Zugehörige Referenzen:**

- „User-defined types“ in *SQL Reference, Volume 1*
- „Data types“ in *SQL Reference, Volume 1*

---

## Details zum Erstellen eines benutzerdefinierten Datentyps (UDT)

Die folgenden Abschnitte behandeln einzigartige Datentypen und die Definitionen strukturierter Typen sowie Typzuordnungen für zusammengeslossene Datenbanken.

### Erstellen eines benutzerdefinierten einzigartigen Datentyps

Ein benutzerdefinierter einzigartiger Datentyp ist ein von einem vorhandenen Typ wie „ganze Zahl“ (INTEGER), „Dezimalzahl“ (DECIMAL) oder „Zeichen“ (CHAR) abgeleiteter Datentyp. Sie können einen einzigartigen Datentyp mit Hilfe der Anweisung CREATE DISTINCT TYPE erstellen.

### Einschränkungen:

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 Typ LOB, DATA LINK, 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.

### Vorgehensweise:

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
```

Wenn ein einzigartiger Datentyp erstellt ist, kann er in Spaltendefinitionen einer Anweisung CREATE TABLE verwendet werden:

```
CREATE TABLE EMPLOYEE
(EMPNO      CHAR(6)      NOT NULL,
 FIRSTNAME  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 eine 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.

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.

Eine Unterstützung für Umsetzungen kann auch durch die Anweisung CREATE METHOD sowie durch Erweiterungen der Anweisung CREATE FUNCTION bereitgestellt werden.

### Zugehörige Konzepte:

- „Benutzerdefinierter Datentyp (UDT)“ auf Seite 129

### Zugehörige Referenzen:

- „CREATE DISTINCT TYPE statement“ in *SQL Reference, Volume 2*
- „CREATE TRANSFORM statement“ in *SQL Reference, Volume 2*
- „CREATE METHOD statement“ in *SQL Reference, Volume 2*
- „CREATE FUNCTION (Sourced or Template) statement“ in *SQL Reference, Volume 2*

- „User-defined types“ in *SQL Reference, Volume 1*
- „Data types“ in *SQL Reference, Volume 1*

## Erstellen eines benutzerdefinierten strukturierten Typs

Ein strukturierter Typ ist ein benutzerdefinierter Datentyp, der ein oder mehrere benannte Attribute enthält. Jedem Attribut ist ein Name und ein eigener Datentyp zugeordnet. Bei Attributen handelt es sich um Merkmale, die ein Exemplar eines bestimmten Typs beschreiben. 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.

### Zugehörige Konzepte:

- „User-defined structured types“ in *Application Development Guide: Programming Server Applications*
- „Structured type hierarchies“ in *Application Development Guide: Programming Server Applications*

### Zugehörige Tasks:

- „Creating structured types“ in *Application Development Guide: Programming Server Applications*
- „Creating a structured type hierarchy“ in *Application Development Guide: Programming Server Applications*

### Zugehörige Referenzen:

- „CREATE TYPE (Structured) statement“ in *SQL Reference, Volume 2*
- „User-defined types“ in *SQL Reference, Volume 1*

## Erstellen einer Typenzuordnung

In Systemen zusammenschlossener Datenbanken können Sie mit Hilfe einer Typenzuordnung spezifische Datentypen von Datenquellentabellen und -sichten 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.

### Einschränkungen:

Für die Typen LOB, LONG VARCHAR/VARGRAPHIC, DATALINK sowie für strukturierte und einzigartige Typen können Sie keine Typenzuordnung erstellen.

### Vorgehensweise:

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.

Das folgende Beispiel zeigt eine Anweisung zur Erstellung einer Typenzuordnung:

```
CREATE TYPE MAPPING MY_ORACLE_DEC FROM SYSIBM.DECIMAL(10,2)
TO SERVER ORACLE1 TYPE_NUMBER([10..38],2)
```

### Zugehörige Referenzen:

- „CREATE TYPE MAPPING statement“ in *SQL Reference, Volume 2*
- „Data Type Mappings between DB2 and OLE DB“ in *Application Development Guide: Programming Client Applications*

---

## Erstellen einer Sicht

Sichten werden aus 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 eine 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 wenn die INSERT-Anweisung direkt auf die Identitätsspalte der Basistabelle verweisen würde.

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 Berechnen 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 zugrunde liegenden 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.

### Voraussetzungen:

Die Basistabelle, der Kurzname bzw. die Sicht, auf der die Sicht basieren soll, muss vorhanden sein, bevor die Sicht erstellt werden kann.

### Einschränkungen:

Sie können eine Sicht erstellen, die in ihrer Definition eine benutzerdefinierte Funktion verwendet. Wenn diese Sicht jedoch aktualisiert werden soll, damit sie die neuesten 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 Sonderversgütungen (BONUS) arbeitet.

### Vorgehensweise:

Gehen Sie wie folgt vor, um eine Sicht über die Steuerzentrale 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 den Ordner **Sichten** und wählen Sie im Kontextmenü **Erstellen** aus.
3. Geben Sie die erforderlichen Informationen ein und klicken Sie 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.

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

```

#### **Zugehörige Konzepte:**

- „Views“ in *SQL Reference, Volume 1*
- „Zugriffsrechte für Tabellen und Sichten“ auf Seite 259
- „Steuern des Zugriffs auf Daten mit Sichten“ auf Seite 270
- „Verwenden von Auslösern zum Aktualisieren von Sichtinhalten“ auf Seite 125

#### **Zugehörige Tasks:**

- „Erstellen einer typisierten Sicht“ auf Seite 136
- „Entfernen von Zeilen aus einer Tabelle oder Sicht“ auf Seite 191
- „Ändern oder Löschen einer Sicht“ auf Seite 216
- „Wiederherstellen unbrauchbarer Sichten“ auf Seite 218

#### **Zugehörige Referenzen:**

- „CREATE VIEW statement“ in *SQL Reference, Volume 2*
- „INSERT statement“ in *SQL Reference, Volume 2*

---

## Details zum Erstellen einer Sicht

Eine typisierte Sicht basiert auf einem vordefinierten strukturierten Typ.

### Erstellen einer typisierten Sicht

#### Vorgehensweise:

Sie können mit Hilfe der Anweisung CREATE VIEW eine typisierte Sicht erstellen.

#### Zugehörige Konzepte:

- „Typed views“ in *Application Development Guide: Programming Server Applications*

#### Zugehörige Tasks:

- „Creating typed views“ in *Application Development Guide: Programming Server Applications*
- „Altering typed views“ in *Application Development Guide: Programming Server Applications*
- „Dropping typed views“ in *Application Development Guide: Programming Server Applications*

#### Zugehörige Referenzen:

- „CREATE VIEW statement“ in *SQL Reference, Volume 2*

---

## Erstellen einer gespeicherten Abfragetabelle

Eine *gespeicherte Abfragetabelle* (engl. materialized query table) ist eine Tabelle, deren Definition auf einem Abfrageergebnis basiert. Das heißt, die gespeicherte Abfragetabelle enthält in der Regel zuvor berechnete Ergebnisse auf der Grundlage von Daten, die in der bzw. den Tabellen vorhanden sind, auf denen die Definition der gespeicherten Abfragetabelle basiert. Wenn der SQL-Compiler feststellt, dass eine Abfrage auf eine gespeicherte Abfragetabelle effektiver ausgeführt werden kann als auf die Basistabelle bzw. Basistabellen, wird die Abfrage auf die gespeicherte Abfragetabelle ausgeführt, und Sie erhalten das Ergebnis schneller als mit der anderen Methode.

#### Einschränkungen:

Mit REFRESH DEFERRED definierte gespeicherte Abfragetabellen werden nicht zum Optimieren von statischem SQL verwendet.

Sie sollten beim Einstellen des Sonderregisters CURRENT REFRESH AGE auf einen anderen Wert als Null Vorsicht walten lassen. Wenn Sie zulassen, dass eine gespeicherte Abfragetabelle, die die Werte der zugrunde liegenden Basistabelle eventuell nicht repräsentiert, zur Optimierung der Abfrageverarbeitung verwendet wird, gibt das Abfrageergebnis die Daten in der zugrunde liegenden Tabelle eventuell *nicht* korrekt wieder. Dies ist eventuell akzeptabel, wenn Sie wissen, dass die zugrunde liegenden Daten nicht geändert wurden oder wenn Sie die Abweichungen in den Ergebnissen aufgrund Ihrer Kenntnisse der Daten akzeptieren.

Wenn Sie eine neue Basistabelle erstellen wollen, die auf einer beliebigen gültigen Gesamtauswahl ( *voll-select*) beruht, geben Sie das Schlüsselwort DEFINITION ONLY beim Erstellen der Tabelle an. Wenn die Operation zur Tabellenerstellung beendet ist, wird die neue Tabelle nicht wie eine gespeicherte Abfragetabelle, son-



dern wie eine Basistabelle behandelt. Sie können zum Beispiel die in LOAD und SET INTEGRITY verwendeten Ausnahmetabellen wie folgt erstellen:

```
CREATE TABLE XT AS
(SELECT T.*, CURRENT_TIMESTAMP AS TIMESTAMP,CLOB(",32K)
AS MSG FROM T) DEFINITION ONLY
```

Es folgt eine Übersicht über eine Reihe der wichtigsten Einschränkungen für gespeicherte Abfragetabellen:

1. Sie können eine gespeicherte Abfragetabelle nicht ändern.
2. Sie können die Länge einer Spalte für eine Basistabelle nicht ändern, wenn es für diese Tabelle eine gespeicherte Abfragetabelle gibt.
3. In eine gespeicherte Abfragetabelle können keine Daten importiert werden.
4. Für eine gespeicherte Abfragetabelle kann kein eindeutiger Index erstellt werden.
5. Es kann keine gespeicherte Abfragetabelle erstellt werden, die auf dem Ergebnis einer Abfrage basiert, die auf einen oder mehrere Kurznamen verweist.

#### **Vorgehensweise:**

Die Erstellung einer gespeicherten Abfragetabelle mit der Replikationsoption kann verwendet werden, um Tabellen auf allen Knoten in einer partitionierten Datenbankumgebung zu replizieren. Solche Tabellen werden als „replizierte gespeicherte Abfragetabellen“ bezeichnet.

In der Regel wird eine gespeicherte Abfragetabelle oder replizierte gespeicherte Abfragetabelle zum Optimieren einer Abfrage verwendet, wenn die Isolationsstufe der gespeicherten Abfragetabelle bzw. der replizierten gespeicherten Abfragetabelle größer oder gleich der Isolationsstufe der Abfrage ist. Wird eine Abfrage beispielsweise unter der Isolationsstufe für Cursorstabilität (CS) ausgeführt, werden für die Optimierung nur solche gespeicherten Abfragetabellen bzw. replizierten gespeicherten Abfragetabellen verwendet, die unter CS oder einer höheren Isolationsstufe definiert wurden.

Sie können eine gespeicherte Abfragetabelle mit Hilfe der Anweisung CREATE TABLE mit der Klausel AS *voll-select* und der Option REFRESH IMMEDIATE oder DEFERRED erstellen.

Sie haben die Möglichkeit, die Namen der Spalten in der gespeicherten Abfragetabelle eindeutig anzugeben. Die Liste der Spaltennamen muss für jede Spalte der Ergebnistabelle in der Gesamtauswahl (*voll-select*) einen Namen enthalten. Eine Liste der Spaltennamen muss angegeben werden, wenn die Ergebnistabelle für die Gesamtauswahl doppelte Spaltennamen oder Spalten ohne Benennung enthält. 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.

Bei der Erstellung einer gespeicherten Abfragetabelle haben Sie die Möglichkeit, anzugeben, ob das System oder der Benutzer die gespeicherte Abfragetabelle pflegen soll. Die Standarddefinition sieht die Pflege durch das System vor, die auch explizit mit der Klausel MAINTAINED BY SYSTEM angegeben werden kann. Vom Benutzer verwaltete gespeicherte Abfragetabellen werden mit der Klausel MAINTAINED BY USER angegeben.

Wenn Sie eine vom System verwaltete gespeicherte Abfragetabelle erstellen, haben Sie ferner die Option, anzugeben, ob die gespeicherte Abfragetabelle automatisch zu aktualisieren ist, wenn die Basistabelle geändert wird, oder ob sie durch die Anweisung REFRESH TABLE aktualisiert werden soll. Geben Sie das Schlüsselwort REFRESH IMMEDIATE an, um die gespeicherte Abfragetabelle automatisch aktualisieren zu lassen, wenn Änderungen an der Basistabelle bzw. den Basistabellen vorgenommen werden. Die sofortige Aktualisierung ist in folgenden Fällen sinnvoll:

- Ihre Abfragen müssen sicherstellen, dass die Daten, auf die sie zugreifen, auf neuestem Stand sind.
- Die Basistabelle bzw. die Basistabellen werden nur selten geändert.
- Die Aktualisierung erfordert nur geringen Aufwand.

In diesem Fall kann die gespeicherte Abfragetabelle vorausberechnete Ergebnisse zur Verfügung stellen. Wenn die Aktualisierung der gespeicherten Abfragetabelle verzögert erfolgen soll, geben Sie das Schlüsselwort REFRESH DEFERRED an. Mit REFRESH DEFERRED angegebene gespeicherte Abfragetabellen spiegeln Änderungen an den zugrunde liegenden Basistabellen **nicht** wider. Sie sollten gespeicherte Abfragetabellen verwenden, bei denen dies keine Voraussetzung ist. Wenn Sie zum Beispiel DSS-Abfragen ausführen, könnte die gespeicherte Abfragetabelle aus einer älteren Version übernommene Daten enthalten.

Eine mit REFRESH DEFERRED definierte gespeicherte Abfragetabelle kann anstelle einer Abfrage verwendet werden, wenn Folgendes zutrifft:

- Sie erfüllt die Einschränkungen für eine Gesamtauswahl einer mit REFRESH IMMEDIATE definierten gespeicherten Abfragetabelle mit folgenden Ausnahmen:
  - 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.

Mit dem Sonderregister CURRENT REFRESH AGE können Sie die Zeitdauer angeben, für die die mit REFRESH DEFERRED definierte gespeicherte Abfragetabelle 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.

Das Sonderregister kann auf den Wert ANY oder auf den Wert 99999999999999999999 gesetzt werden, um verzögert aktualisierte (REFRESH DEFERRED) gespeicherte Abfragetabellen in einer dynamischen Abfrage verwenden zu können. Die Folge von Neunen ist der zulässige Maximalwert in diesem Sonderregister, das einen Zeitmarkendifferenzwert mit dem Datentyp DECIMAL(20,6) darstellt. Der Wert null (0) gibt an, dass nur mit REFRESH IMMEDIATE definierte gespeicherte Abfragetabellen zur Optimierung der Verarbeitung einer Abfrage verwendet werden dürfen. In diesem Fall werden gespeicherte Abfragetabellen, die mit REFRESH DEFERRED definiert wurden, nicht zur Optimierung verwendet.

Mit REFRESH IMMEDIATE definierte gespeicherte Abfragetabellen gelten für statische und dynamische Abfragen und kommen ohne das Sonderregister CURRENT REFRESH AGE aus.

Abfragen werden an gespeicherte Abfragetabellen geleitet, wenn die Tabelle mit der Klausel ENABLE QUERY OPTIMIZATION definiert wurde und wenn, im Fall einer mit REFRESH DEFERRED definierten gespeicherten Abfragetabelle, das

Sonderregister CURRENT REFRESH AGE auf den Wert ANY gesetzt wurde. Jedoch ist für benutzerverwaltete gespeicherte Abfragetabellen die Verwendung des Sonderregisters CURRENT REFRESH AGE nicht die beste Methode zur Steuerung der Weiterleitung von Abfragen. Das Sonderregister CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION gibt an, welche Arten im Cache zwischengespeicherter Daten für die Weiterleitung verfügbar sind.

Bei Aktivitäten, die sich auf die Quelldaten beziehen, enthält eine gespeicherte Abfragetabelle nach einem gewissen Zeitraum keine exakten Daten. Sie müssen zur Aktualisierung die Anweisung REFRESH TABLE verwenden.

#### Zugehörige Konzepte:

- „Isolation levels“ in *SQL Reference, Volume 1*

#### Zugehörige Tasks:

- „Ändern der Merkmale einer gespeicherten Abfragetabelle“ auf Seite 207
- „Aktualisieren der Daten in einer gespeicherten Abfragetabelle“ auf Seite 208
- „Löschen einer gespeicherten Abfragetabelle oder einer Zwischenspeichertabelle“ auf Seite 218

#### Zugehörige Referenzen:

- „CREATE TABLE statement“ in *SQL Reference, Volume 2*
- „REFRESH TABLE statement“ in *SQL Reference, Volume 2*
- „SET CURRENT REFRESH AGE statement“ in *SQL Reference, Volume 2*
- „CURRENT REFRESH AGE special register“ in *SQL Reference, Volume 1*
- „CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION special register“ in *SQL Reference, Volume 1*
- „SET CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION statement“ in *SQL Reference, Volume 2*

---

## Erstellen einer benutzerverwalteten gespeicherten Abfragetabelle

Benutzerverwaltete gespeicherte Abfragetabellen (MQTs) sind für Datenbanksysteme nützlich, in denen Tabellen von Übersichtsdaten bereits vorhanden sind. Angepasste Anwendungen, die solche Übersichtstabellen verwalten, werden häufig eingesetzt. Die Angabe vorhandener Übersichtstabellen als benutzerverwaltete gespeicherte Abfragetabellen veranlasst das Abfrageoptimierungsprogramm, die vorhandene Übersichtstabelle zur Berechnung der Ergebnismengen für Abfragen auf die Basistabelle zu verwenden.

**Anmerkung:** Das Abfrageoptimierungsprogramm verwendet keine benutzerverwalteten gespeicherten Abfragetabellen, wenn ein Zugriffsplan für statische SQL-Abfragen ausgewählt wird.

#### Einschränkungen:

Wenn Sie eine vom Benutzer verwaltete gespeicherte Abfragetabelle erstellen, gelten dieselben Einschränkungen wie für eine vom System verwaltete gespeicherte Abfragetabelle, jedoch mit folgenden Ausnahmen:

- INSERT-, UPDATE- und DELETE-Operationen sind an der gespeicherten Abfragetabelle zulässig. Jedoch erfolgt keine Gültigkeitsprüfung an den zugrunde liegenden Basistabellen. Sie sind für die Richtigkeit der Daten selbst verantwortlich.

- Die Dienstprogramme LOAD, EXPORT, IMPORT sowie die Datenreplikation funktionieren für diese Art einer gespeicherten Abfragetabelle, jedoch wird keine Gültigkeitsprüfung durchgeführt.
- Die Anweisung REFRESH TABLE darf für diese Art einer gespeicherten Abfragetabelle nicht verwendet werden.
- Die Anweisung SET INTEGRITY ... IMMEDIATE CHECKED darf für diese Art einer gespeicherten Abfragetabelle nicht verwendet werden.
- Vom Benutzer verwaltete gespeicherte Abfragetabellen müssen mit REFRESH DEFERRED definiert werden.

Im Abschnitt „Erstellen einer gespeicherten Abfragetabelle“ finden Sie Informationen zu weiteren Einschränkungen.

#### **Vorgehensweise:**

Sie können eine gespeicherte Abfragetabelle mit Hilfe der Anweisung CREATE TABLE mit der Klausel AS *voll-select* und der Option REFRESH IMMEDIATE oder DEFERRED erstellen.

Bei der Erstellung einer gespeicherten Abfragetabelle haben Sie die Möglichkeit, anzugeben, ob das System oder der Benutzer die gespeicherte Abfragetabelle verwalten soll. Die Standarddefinition sieht die Verwaltung durch das System vor, die auch explizit mit der Klausel MAINTAINED BY SYSTEM angegeben werden kann. Vom Benutzer verwaltete gespeicherte Abfragetabellen werden mit der Klausel MAINTAINED BY USER angegeben.

In großen Datenbankumgebungen oder in Data Warehouse-Umgebungen sind in vielen Fällen angepasste Anwendungen vorhanden, die benutzerverwaltete gespeicherte Abfragetabellen pflegen und mit Daten füllen.

**Anmerkung:** Damit das Optimierungsprogramm eine benutzerverwaltete gespeicherte Abfragetabelle in Betracht zieht, muss der Abfrageoptimierungsgrad auf 2 oder auf einen Grad höher oder gleich 5 gesetzt werden.

#### **Zugehörige Tasks:**

- „Füllen einer benutzerverwalteten gespeicherten Abfragetabelle“ auf Seite 140

#### **Zugehörige Referenzen:**

- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

---

## **Füllen einer benutzerverwalteten gespeicherten Abfragetabelle**

Wenn Sie die Tabelle zur Aufnahme der Übersichtsdaten erstellt haben, soll die gespeicherte Abfragetabelle (MQT) sicherlich mit den Übersichtsdaten gefüllt werden, die vom Optimierungsprogramm zur Bestimmung von Ergebnismengen zu verwenden sind.

#### **Voraussetzungen:**

Stellen Sie sicher, dass die Tabelle zur Aufnahme der Übersichtsdaten vorhanden ist.

### Vorgehensweise:

Sie können benutzerverwaltete gespeicherte Abfragetabellen mit Hilfe von Auslösern und Einfügeoperationen sowie mit den Dienstprogrammen LOAD, IMPORT und DB2 DataPropagator füllen. Bei der Ausführung des ersten Füllens einer benutzerverwalteten gespeicherten Abfragetabelle können Sie den Protokollieraufwand vermeiden, indem Sie das Dienstprogramm LOAD oder IMPORT verwenden.

Die folgenden Schritte stellen ein typisches Verfahren zum Füllen einer benutzerverwalteten gespeicherten Abfragetabelle dar:

- Aktivieren Sie den Schreibschutz für die Basistabellen, um zu verhindern, dass neue Datensätze erstellt oder vorhandene Datensätze modifiziert werden.
- Extrahieren Sie die benötigten Daten aus den Basistabellen und schreiben Sie sie in eine externe Datei.
- Importieren oder laden Sie die Daten aus der externen Datei in die gespeicherte Abfragetabelle. Sie können die Dienstprogramme LOAD und IMPORT für eine Tabelle im Status CHECK PENDING NO ACCESS (Prüfung anstehend, kein Zugriff) verwenden.

**Anmerkung:** Wenn Sie die gespeicherte Abfragetabelle mit SQL-Einfügeoperationen füllen wollen, müssen Sie den Status 'Anstehend, kein Zugriff' (PENDING NO ACCESS) ändern. Das Optimierungsprogramm muss jedoch zuerst durch die Option DISABLE QUERY OPTIMIZATION in der Klausel SET MATERIALIZED QUERY der Anweisung ALTER TABLE inaktiviert werden, um sicherzustellen, dass eine dynamische SQL-Abfrage nicht zufällig mit Zugriff auf diese gespeicherte Abfragetabelle optimiert wird, während die Daten noch hergestellt werden. Wenn die gespeicherte Abfragetabelle gefüllt ist, muss die Optimierung über die Option ENABLE QUERY OPTIMIZATION in der Klausel SET MATERIALIZED QUERY der Anweisung ALTER TABLE aktiviert werden.

- Setzen Sie den Status CHECK PENDING NO ACCESS zurück, um SQL-Abfragen auf die neue gespeicherte Abfrage auszuführen. Dadurch geben Sie an, dass Sie die Verantwortung für die Datenintegrität der gespeicherten Abfragetabelle übernommen haben. Die Anweisung dazu sieht wie folgt aus:

```
DB2 SET INTEGRITY FOR example ALL IMMEDIATE UNCHECKED
```

- Aktivieren Sie den Schreib-/Lesezugriff auf die Basistabellen.

**Anmerkung:** Damit das Optimierungsprogramm eine benutzerverwaltete gespeicherte Abfragetabelle in Betracht zieht, muss der Abfrageoptimierungsgrad auf 2 oder auf einen Grad höher oder gleich 5 gesetzt werden.

### Zugehörige Referenzen:

- „IMPORT Command“ in *Command Reference*
- „LOAD Command“ in *Command Reference*

---

## Erstellen einer Zwischenspeichertabelle

Eine *Zwischenspeichertabelle* (engl. staging table) ermöglicht eine Unterstützung von Teilaktualisierungen (inkrementelle Pflege) für gespeicherte Abfragetabellen, die mit REFRESH DEFERRED definiert sind. Die Zwischenspeichertabelle sammelt Änderungen, die auf die gespeicherte Abfragetabelle angewendet werden müssen, um diese mit dem Inhalt der zugrunde liegenden Tabellen zu synchronisieren.

Durch die Nutzung von Zwischenspeichertabellen wird die starke Sperrkonkurrenz beseitigt, die durch sofort zu aktualisierende Daten verursacht wird, wenn eine unverzügliche Aktualisierung (REFRESH IMMEDIATE) der gespeicherten Abfragetabelle angefordert ist. Außerdem brauchen gespeicherte Abfragetabellen nicht länger völlig neu generiert zu werden, wenn ein Befehl REFRESH TABLE ausgeführt wird.

Gespeicherte Abfragetabellen sind eine leistungsfähige Methode zum Verbessern der Antwortzeit komplexer Abfragen, insbesondere solcher Abfragen, die eventuell eine der folgenden Operationen erfordern:

- Ermitteln von Ergebnisdaten aus einer oder mehreren Dimensionen
- Verknüpfungen und Ergebnisberechnung von Daten einer Gruppe von Tabellen
- Verarbeiten von Daten aus einer häufig verwendeten Teilmenge von Daten
- Neupartitionieren von Daten aus einer Tabelle bzw. einem Teil einer Tabelle in einer partitionierten Datenbankumgebung

#### **Einschränkungen:**

Es folgt eine Übersicht über einige der wichtigsten Einschränkungen für Zwischenspeichertabellen:

1. Die Abfrage, die zur Definition der Zwischenspeichertabelle verwendet wird, muss inkrementell aktualisierbar sein. Das bedeutet, sie muss den gleichen Regeln wie eine gespeicherte Abfragetabelle mit der Option zur sofortigen Aktualisierung (REFRESH IMMEDIATE) genügen.
2. Eine unterstützende Zwischenspeichertabelle ist nur für eine verzögerte Aktualisierung (REFRESH DEFERRED) möglich. Die Abfrage definiert auch die gespeicherte Abfragetabelle, der die Zwischenspeichertabelle zugeordnet wird. Die gespeicherte Abfragetabelle muss mit der Option REFRESH DEFERRED definiert sein.
3. Bei der Aktualisierung über die Zwischenspeichertabellen wird nur eine Aktualisierung bis zum aktuellen Zeitpunkt unterstützt.

#### **Vorgehensweise:**

Eine Zwischenspeichertabelle in einem inkonsistenten, unvollständigen oder schwebenden (pending) Status kann nicht zur inkrementellen Aktualisierung (Teilaktualisierung) der zugeordneten gespeicherten Abfragetabelle verwendet werden, sofern nicht einige weitere Operationen stattfinden. Diese Operationen sorgen dafür, dass der Inhalt der Zwischenspeichertabelle mit der ihr zugeordneten gespeicherten Abfragetabelle und deren zugrunde liegenden Tabellen konsistent wird, und nehmen die Zwischenspeichertabelle aus dem schwebenden Status heraus. Nach der Aktualisierung einer gespeicherten Abfragetabelle wird der Inhalt der zugehörigen Zwischenspeichertabelle gelöscht und die Zwischenspeichertabelle in den Normalstatus versetzt. Eine Zwischenspeichertabelle kann mit Hilfe der Anweisung SET INTEGRITY mit den entsprechenden Optionen auch vorsätzlich abgeschnitten werden (PRUNE). Das Abschneiden versetzt die Zwischenspeichertabelle in einen inkonsistenten Status. Zum Beispiel erzwingt die folgende Anweisung das Abschneiden einer Zwischenspeichertabelle mit dem Namen STAGTAB1:

```
SET INTEGRITY FOR STAGTAB1 PRUNE;
```

Eine Zwischenspeichertabelle wird bei ihrer Erstellung in einen schwebenden Status versetzt und besitzt einen Anzeiger, der darauf hinweist, dass die Tabelle inkonsistent und unvollständig im Hinblick auf den Inhalt der zugrunde liegenden Tabellen und der zugeordneten gespeicherten Abfragetabelle ist. Die Zwischenspeichertabelle muss aus dem schwebenden und inkonsistenten Status herausge-

nommen werden, um mit der Erfassung der Änderungen aus den zugrunde liegenden Tabellen beginnen zu können. Während des schwebenden Status schlagen alle Versuche, Änderungen an einer beliebigen der zugrunde liegenden Tabellen der Zwischenspeichertabelle vorzunehmen, ebenso fehl wie alle Versuche, die zugeordnete gespeicherte Abfragetabelle zu aktualisieren.

Eine Zwischenspeichertabelle kann durch mehrere Methoden aus einem schwebenden Status herausgenommen werden. Zum Beispiel:

- SET INTEGRITY FOR <zwischenstichtabelle> STAGING IMMEDIATE UNCHECKED
- SET INTEGRITY FOR <zwischenstichtabelle> IMMEDIATE CHECKED

#### **Zugehörige Tasks:**

- „Erstellen einer gespeicherten Abfragetabelle“ auf Seite 136
- „Ändern der Merkmale einer gespeicherten Abfragetabelle“ auf Seite 207
- „Aktualisieren der Daten in einer gespeicherten Abfragetabelle“ auf Seite 208
- „Löschen einer gespeicherten Abfragetabelle oder einer Zwischenspeichertabelle“ auf Seite 218

#### **Zugehörige Referenzen:**

- „SET INTEGRITY statement“ in *SQL Reference, Volume 2*

---

## **Erstellen eines Aliasnamens**

Ein Aliasname ist eine indirekte Methode, auf eine Tabelle, einen Kurznamen oder eine Sicht zu verweisen, so dass eine SQL-Anweisung vom qualifizierten Namen dieser Tabelle oder Sicht unabhängig sein kann. 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.

#### **Voraussetzungen:**

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.

#### **Einschränkungen:**

Ein Aliasname kann überall dort verwendet werden, wo ein vorhandener Tabellenname verwendet werden kann, und kann auf einen anderen Aliasnamen verweisen, sofern keine rückbezüglichen oder sich wiederholenden 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 sich nicht in einem anderen Schema als dem befindet, das Ihrer aktuellen Berechtigungs-ID eigen ist. In diesem Fall benötigen Sie die Berechtigung DBADM.

Wenn ein Aliasname oder das Objekt, auf das ein Aliasname verweist, gelöscht wird, werden alle von dem Aliasnamen abhängigen Pakete als ungültig und alle von dem Aliasnamen abhängigen Sichten und Auslöser als unbrauchbar markiert.

#### Vorgehensweise:

Gehen Sie wie folgt vor, um einen Aliasnamen über die Steuerzentrale 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 den Ordner **Aliasnamen** und wählen Sie im Kontextmenü **Erstellen** aus.
3. Geben Sie die erforderlichen Informationen ein und klicken Sie 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 Aliasnamenskette 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 OS/390 oder z/Series 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 OS/390 oder z/Series gilt:
  - Sie machen eine spezielle Berechtigung oder ein bestimmtes Zugriffsrecht für ihre Ersteller erforderlich.
  - Sie können nicht auf andere Aliasnamen verweisen.
- Für SYNONYMe in DB2 für OS/390 oder z/Series gilt:
  - Sie können nur von ihrem Ersteller verwendet werden.
  - Sie sind immer ohne Qualifikationsmerkmal.
  - Sie werden gelöscht, wenn die Bezugstabelle gelöscht wird.
  - Sie benutzen keinen Namensbereich mit Tabellen oder Sichten gemeinsam.

#### Zugehörige Konzepte:

- „Aliases“ in *SQL Reference, Volume 1*

#### Zugehörige Referenzen:

- „CREATE ALIAS statement“ in *SQL Reference, Volume 2*



---

## Index, Indexerweiterung oder 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, sondern ist lediglich eine Beschreibung eines 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).
- 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 *db2adv*.

Ein Index wird durch 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.

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 zu einer Verlangsamung 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 Indexschlüssels 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, jedoch 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 *Clusterindex* 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 Vorabesezugriff ergeben.

Wenn Sie einen Primärschlüsselindex als Clusterindex 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 Clusterindex eindeutig ist.

Spaltendaten, die nicht Teil des eindeutigen Indexschlüssels sind, jedoch im Index gespeichert/gepflegt werden sollen, 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 geprüft. Die Verwendung von *INCLUDE*-Spalten verbessert die Leistung beim Abrufen von Daten, wenn ein Indexzugriff stattfindet.

Der Datenbankmanager verwendet zum Speichern von Indizes eine B+-Baumstruktur, deren unterste Ebene aus so genannten Blattknoten (engl. leaf nodes) besteht. Auf den Blattknoten oder -seiten sind die eigentlichen Indexschlüsselwerte gespeichert. Bei der Indexerstellung können Sie festlegen, dass diese Indexblattseiten online zusammengefügt werden können. Durch diese Onlinedefragmentierung des Index wird verhindert, dass nach zahlreichen Löscho- und Aktualisierungsvorgängen auf vielen Blattseiten eines Index nur noch wenige Indexschlüssel vorhanden sind. In einem solchen Fall könnte ohne die Onlineindexdefragmentierung der freie Speicherplatz nur durch eine Reorganisation der Daten mit oder ohne Index zurückgewonnen werden. Bei der Entscheidung, ob ein Index so erstellt werden soll, dass eine Onlinedefragmentierung der Indexseiten möglich ist, sollte die folgende Fragestellung berücksichtigt werden: Sind die zusätzlichen Aufwände zur Prüfung auf vorhandenen Speicherplatz für das Zusammenfügen von Seiten, jedesmal wenn ein Schlüssel physisch aus einer Blattseite entfernt wird, sowie die tatsächlichen Aufwände zur Durchführung der Zusammenfügung, wenn genügend Speicherplatz verfügbar ist, größer als die Vorteile für den Index, die sich aus der besseren Speicherplatznutzung ergeben, und kleiner als die geringere Notwendigkeit zur Durchführung einer Reorganisation, um Speicherplatz zurückzugewinnen?

#### **Anmerkungen:**

1. Die durch eine Onlineindexdefragmentierung zurückgewonnenen Seiten können nur für andere Indizes in der gleichen Tabelle verwendet werden. Bei einer vollständigen Reorganisation stehen die zurückgewonnenen Seiten für andere Objekte (bei DMS-Speicher) oder als Plattenspeicherplatz (bei SMS-Speicher) zur Verfügung. Außerdem werden durch die Onlineindexdefragmentierung keine Nichtblattseiten des Index zurückgewonnen, während die vollständige Reorganisation den Index auf Minimalgröße bringt, indem Nichtblattseiten und Blattseiten sowie die Anzahl der Indexstufen reduziert werden.
2. Bei Indizes, die mit einer Version vor Version 8 erstellt wurden, wird ein Schlüssel im Rahmen der Löschung oder Aktualisierung einer Tabellenzeile physisch aus einer Blattseite entfernt. Bei Typ-2-Indizes werden Schlüssel lediglich als gelöscht markiert, wenn eine Zeile gelöscht oder aktualisiert wird. Sie werden von einer Seite erst dann physisch entfernt, wenn eine Bereinigung durchgeführt wird, nachdem die Löschung oder Aktualisierung festgeschrieben wurde. Eine solche Bereinigung kann durch eine nachfolgende Transaktion durchgeführt werden, welche die Seite ändert, auf der ein Schlüssel als gelöscht markiert ist. Die Bereinigung kann explizit durch die Option CLEANUP ONLY [ALL | PAGES] des Dienstprogramms REORG INDEXES ausgelöst werden.

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 Datenbankpartitionsgruppe. Beachten Sie, dass in einer Mehrpartitions Umgebung der Indexschlüssel definierter eindeutiger Indizes eine Obermenge des Partitionierungsschlüssels sein muss.

#### **Zugehörige Konzepte:**

- „Indexes“ in *SQL Reference, Volume 1*
- „Verwenden eines Index“ auf Seite 149
- „Optionen der Anweisung CREATE INDEX“ auf Seite 150
- „Erstellen eines benutzerdefinierten erweiterten Indextyps“ auf Seite 155
- „Zugriffsrechte für Indizes“ auf Seite 262

#### Zugehörige Tasks:

- „Aktivieren der Parallelität beim Erstellen von Indizes“ auf Seite 11
- „Erstellen eines Index“ auf Seite 148
- „Umbenennen einer vorhandenen Tabelle oder eines Index“ auf Seite 209
- „Löschen eines Index, einer Indexerweiterung oder einer Indexspezifikation“ auf Seite 220

#### Zugehörige Referenzen:

- „CREATE INDEX statement“ in *SQL Reference, Volume 2*
- „CREATE INDEX EXTENSION statement“ in *SQL Reference, Volume 2*

---

## Details zum Erstellen eines Index, einer Indexerweiterung oder einer Indexspezifikation

Sie können mit den Indizes arbeiten, die vom Datenbankmanager verwaltet werden oder auch einen eigenen Index angeben.

### Erstellen eines Index

Ein *Index* ist eine Gruppe aus einem oder mehreren Schlüsseln, die jeweils auf Zeilen in einer Tabelle zeigen. Ein Index ermöglicht einen effizienteren Zugriff auf Zeilen einer Tabelle, indem er einen direkten Pfad zu den Daten mit Hilfe von Zeigern erstellt.

#### Vorgehensweise:

**Hinweis zur Leistung:** Wenn Sie die folgende Abfolge von Operationen durchführen wollen:

1. Erstellen einer Tabelle
2. Laden von Daten in Tabelle
3. Erstellen eines Index (ohne Option COLLECT STATISTICS)
4. Ausführen von RUNSTATS

Oder wenn Sie die folgende Abfolge von Operationen durchführen wollen:

1. Erstellen einer Tabelle
2. Laden von Daten in Tabelle
3. Erstellen eines Index (mit Option COLLECT STATISTICS)

In diesen Fällen sollten Sie die folgende Reihenfolge zur Durchführung der Operationen in Betracht ziehen:

1. Erstellen der Tabelle
2. Erstellen des Index
3. Laden von Daten in Tabelle mit angeforderter Option `statistics yes`

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.

Bei der Erstellung einer Tabelle können Sie eine MDC-Tabelle (Multidimensional Clustering) erstellen. Durch die Erstellung dieses Typs von Tabelle werden Blockindizes erstellt. Normale Indizes zeigen auf einzelne Zeilen. Blockindizes hingegen verweisen auf Datenblöcke oder Datenbereiche und sind wesentlich kleiner als normale Indizes. Blockindizes werden zusammen mit normalen Indizes im gleichen Tabellenbereich gespeichert.

Wenn eine Zeile eingefügt wird, wird sie an der bequemsten passenden Speicherposition untergebracht, sofern nicht ein Clusterindex definiert wurde. Wenn nach Zeilen einer Tabelle gesucht wird, die bestimmte Auswahlbedingungen erfüllen sollen, und die Tabelle keinen Index hat, wird 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 festgelegt.

Gehen Sie wie folgt vor, um einen Index über die Steuerzentrale 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 den Ordner **Indizes** und wählen Sie im Kontextmenü **Erstellen** → **Index mit Assistent** aus.
3. Befolgen Sie die im Assistenten aufgeführten Schritte zur Durchführung der Task.

Geben Sie in der Befehlszeile Folgendes ein, um einen Index zu erstellen:

```
CREATE INDEX <name> ON <tabellenname> (<spaltenname>)
```

#### Zugehörige Konzepte:

- „Optimieren der Leistung von LOAD“ in *Dienstprogramme für das Versetzen von Daten Handbuch und Referenz*
- „Verwenden eines Index“ auf Seite 149
- „Optionen der Anweisung CREATE INDEX“ auf Seite 150
- „Zugriffsrechte für Indizes“ auf Seite 262

#### Zugehörige Tasks:

- „Umbenennen einer vorhandenen Tabelle oder eines Index“ auf Seite 209
- „Löschen eines Index, einer Indexerweiterung oder einer Indexspezifikation“ auf Seite 220

#### Zugehörige Referenzen:

- „CREATE INDEX statement“ in *SQL Reference, Volume 2*

## 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.
- Er verwendet Spalten mit hoher Kardinalität.

**Zugehörige Konzepte:**

- „Tipps zur Indexplanung“ in *Systemverwaltung: Optimierung*
- „Tipps zur Indexleistung“ in *Systemverwaltung: Optimierung*
- „Datenzugriff über Indexsuchen“ in *Systemverwaltung: Optimierung*
- „Tabellen- und Indexverwaltung für Standardtabellen“ in *Systemverwaltung: Optimierung*
- „Tabellen- und Indexverwaltung für MDC-Tabellen“ in *Systemverwaltung: Optimierung*

## Optionen der Anweisung CREATE INDEX

Sie können einen Index erstellen, der mehrfach auftretende Werte zulässt (nicht eindeutiger Index), um ein effizientes Abrufen über Spalten, die nicht zum Primärschlüssel gehören, zu ermöglichen, und es also erlaubt, dass doppelte Werte in der indexierten Spalte bzw. den indexierten Spalten vorhanden sind.

Mit der folgenden SQL-Anweisung wird ein nicht eindeutiger Index namens LNAME für die Spalte LASTNAME der Tabelle EMPLOYEE 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 mit Spalte definiert, die nicht die INCLUDE-Spalte ist. 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 die Gruppe der Indexschlüsselspalten angehängt werden. Alle Spalten, die mit dieser Klausel eingeschlossen werden, werden nicht zur Erhaltung der Eindeutigkeit verwendet. Die eingeschlossenen Spalten können die Leistung bei einigen Abfragen durch einen reinen Indexzugriff verbessern. Die Spalten dürfen keine Spalten sein, die zur Gewährleistung der Eindeutigkeit verwendet werden (andernfalls empfangen Sie

die Fehlermeldung SQLSTATE 42711). 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 sie dieselbe Gruppe von Spalten angibt, wobei die Spezifikationen zur Reihenfolge oder Richtung (aufsteigend oder absteigend) der Spalten keine Rolle spielen. Wenn eine übereinstimmende Indexdefinition gefunden wurde, wird die Beschreibung des Index geändert, um anzuzeigen, dass es sich um den vom System angeforderten Primärindex handelt, und der Index wird nach einer entsprechenden Überprüfung der Eindeutigkeit in einen eindeutigen Index geändert, wenn er zuvor nicht eindeutig war.

Darum ist es möglich, mehr als einen Primärschlüssel für dieselbe Tabelle zu haben, wie dies im Katalog angegeben wird.

Beim Arbeiten mit einem strukturierten Typ kann es erforderlich sein, benutzerdefinierte Indextypen zu erstellen. Dazu ist eine Möglichkeit zum Definieren von Funktionen für Indexpflege, Indexsuche und Indexausnutzung erforderlich.

Mit der folgenden SQL-Anweisung wird ein Clusterindex 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 Clusterindizes 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 Clusterindex definiert werden.

Wenn die Indexschlüsselwerte dieser neuen Zeilen zum Beispiel stets neue HIGH-KEY-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 Clusterindex 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.

Die obige Darstellung gilt auch für neue "Überlaufzeilen", die aus UPDATE-Operationen resultieren, die eine Zeile vergrößern.

Ein einzelner Index, der mit dem Parameter ALLOW REVERSE SCANS in der Anweisung CREATE INDEX erstellt wurde, kann vorwärts und rückwärts durchsucht werden. Das heißt, solche Indizes unterstützen Suchoperationen in der Richtung, die bei der Erstellung des Index definiert wurde, und Suchoperationen in entgegengesetzter oder umgekehrter Richtung. Die Anweisung könnte ungefähr wie folgt aussehen:

```
CREATE INDEX iname ON tname (cname DESC) ALLOW REVERSE SCANS
```

In diesem Fall wird der Index (iname) nach den absteigenden Werten (DESC) der angegebenen Spalte (cname) gebildet. Durch Zulassen von umgekehrten Suchoperationen kann eine Suche in aufsteigender Folge (umgekehrter Folge) durchgeführt werden, obwohl der Index für die Spalte für Suchoperationen in absteigender Folge definiert ist. Die tatsächliche Verwendung des Index in beiden Richtungen wird nicht von Ihnen, sondern vom Optimierungsprogramm bei der Erstellung und Auswahl von Zugriffsplänen gesteuert.

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 Onlineindexdefragmentierung für diesen Index aktiviert. In diesem Fall wird anhand der folgenden Kriterien entschieden, ob eine Onlineindexdefragmentierung stattfindet: Wenn nach dem physischen Entfernen eines Schlüssels aus einer Blattseite dieses Index der Prozentwert für den 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 Onlineindexdefragmentierung erstellt:

```
CREATE INDEX LASTN ON EMPLOYEE (LASTNAME) MINPCTUSED 20
```

Wenn ein Schlüssel aus einer Seite dieses Index physisch entfernt 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 einer angrenzenden Indexseite zusammengefasst werden. Wenn die zusammengefassten Schlüssel auf eine einzige Seite passen, wird die Zusammenfassung durchgeführt und eine der beiden Indexseiten gelöscht.

Mit der Anweisung CREATE INDEX können Sie den Index erstellen, während gleichzeitig der Schreib- und Lesezugriff auf die zugehörige Tabelle und alle bereits vorhandenen Indizes zugelassen wird. Wenn Sie den Zugriff auf die Tabelle während der Erstellung des Indexes einschränken wollen, können Sie die Tabelle vor der Erstellung des Indexes mit der Anweisung LOCK TABLE sperren. Der neue Index wird erstellt, indem die zugrunde liegende Tabelle durchsucht wird. Alle Änderungen, die während der Erstellung des Indexes an der Tabelle durchgeführt werden, werden protokolliert. Nach der Erstellung des neuen Indexes werden die Änderungen auf diesen angewendet. Damit die protokollierten Änderungen während der Indexerstellung schneller angewendet werden können, wird eine separate Kopie der Änderungen im Speicherpufferbereich gehalten, der bei Bedarf aus dem Zwischenspeicher des Dienstprogramms zugeordnet wird. Dadurch können die Änderungen bei der Indexerstellung verarbeitet werden, indem sie zunächst direkt aus dem Speicher gelesen werden. Gegebenenfalls werden anschließend auch die Protokolle gelesen, jedoch erst zu einem viel späteren Zeitpunkt. Wenn alle Änderungen auf den Index übertragen sind, wird die Tabelle in den Wartemodus versetzt, während der neue Index sichtbar gemacht wird.

Bei der Erstellung eines eindeutigen Index müssen Sie sicherstellen, dass keine doppelten Schlüssel in der Tabelle vorhanden sind und dass auch durch gleichzeitige Einfügeoperationen während der Indexerstellung keine doppelten Schlüssel eingeführt werden. Bei der Indexerstellung werden doppelte Schlüssel mit Hilfe eines verzögert angewandten Eindeutigkeitsschemas ermittelt, so dass doppelte Schlüssel erst am Ende der Indexerstellung erkannt werden. Wenn doppelte Schlüssel vorliegen, schlägt die Indexerstellung an diesem Punkt fehl.



Die Klausel PCTFREE der Anweisung CREATE INDEX gibt den Prozentsatz jeder Indexseite an, die beim Erstellen des Index als Speicherbereich frei zu lassen ist. 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) verbessert. Vorablesezugriff ist eine wichtige Komponente, die zu Leistungssteigerung führen kann. Wenn immer HIGHKEY-Werte vorliegen, sollten Sie ein Senken des Werts für die Klausel PCTFREE der Anweisung CREATE INDEX in Erwägung ziehen. Auf diese Art wird auf jeder Indexseite so wenig Speicherbereich wie möglich verschwendet.

Mit der Klausel LEVEL2 PCTFREE wird das System zur Beibehaltung eines angegebenen Prozentsatzes an freiem Speicherbereich auf jeder Seite in der zweiten Ebene eines Indexes angewiesen. Sie geben einen Prozentsatz an freiem Speicherbereich bei der Indexerstellung an, um zukünftige Einfügungen und Aktualisierungen zu berücksichtigen. Die zweite Ebene ist die Ebene, die direkt oberhalb der Blattebene angeordnet ist. Standardmäßig werden mindestens 10 und der PCTFREE-Wert in allen Nichtblattseiten beibehalten. Der Parameter LEVEL2 PCTFREE ermöglicht das Überschreiben des Standardwertes. Wenn Sie die Integeroption LEVEL2 PCTFREE in der Anweisung CREATE INDEX verwenden, bleibt der Integerprozentwert an freiem Speicherbereich auf temporären Seiten der Stufe 2 erhalten. Ein Minimum von 10 und ein Integerprozentwert an freiem Speicherbereich bleibt auch auf temporären Seiten der Stufe 3 und höher erhalten. Durch die Beibehaltung eines höheren Umfangs an freiem Speicherbereich auf Stufe 2 wird die Anzahl der Seitenteilungen auf Stufe 2 des Indexes reduziert.

Die Klauseln PAGE SPLIT SYMMETRIC, PAGE SPLIT HIGH und PAGE SPLIT LOW erlauben die Auswahl des Seitenteilungsverhaltens beim Einfügen in einen Index.

Die Klausel PAGE SPLIT SYMMETRIC definiert ein Standardverhalten bei der Seitenteilung, bei der eine Teilung etwa in der Mitte der Indexseite erfolgt. Die Verwendung dieses Standardverhaltens eignet sich am besten für wahlfreie Einfügungen in einen Index oder wenn die Einfügung nicht nach einem der Muster ausgeführt wird, die in den Klauseln PAGE SPLIT HIGH und PAGE SPLIT LOW verwendet werden.

Das PAGE SPLIT HIGH-Verhalten ist sinnvoll, wenn der verwendete Index eine kontinuierlich steigende Anzahl von Bereichen umfasst. Eine steigende Anzahl von Bereichen kann im Index unter folgenden Bedingungen auftreten:

- Auf dem System ist ein Index mit mehreren Schlüsselkomponenten definiert, und es gibt viele Werte (mit einem Umfang von mehreren Indexseiten), wobei allen mit Ausnahme der letzten Schlüsselkomponente der gleiche Wert zugeordnet ist.
- Alle Einfügungen in die Tabelle beinhalten einen neuen Wert, der mit Ausnahme der letzten Schlüsselkomponente mit dem Wert eines bereits vorhandenen Schlüssels übereinstimmt.
- Die letzte Schlüsselkomponente des eingefügten Werts ist größer als die der vorhandenen Schlüssel.

Beispiel: Der Index umfasst die folgenden Schlüsselwerte:

(1,1), (1,2), (1,3), ... (1,n),  
(2,1), (2,2), (2,3), ... (2,n),  
...  
(m,1), (m,2), (m,3), ... (m,n)

In diesem Fall weist der nächste einzufügende Schlüssel den Wert (x,y) auf, wobei  $1 \leq x \leq m$  und  $y > n$  ist. Wenn die Einfügungen nach einem derartigen Muster ausgeführt werden, kann die Klausel `PAGE SPLIT HIGH` verwendet werden, damit die Seitenteilungen nicht zu einer hohen Anzahl von Seiten führen, die zu 50 Prozent leer sind.

Analog hierzu kann `PAGE SPLIT LOW` verwendet werden, wenn in einem Index immer weniger Bereiche vorhanden sind. Dadurch kann ebenfalls vermieden werden, dass Seiten zu 50 Prozent leer sind.

**Anmerkung:** Wenn Sie einen Primär- oder einen eindeutigen Schlüssel hinzufügen wollen, und wenn im zugehörigen Index `SPLIT HIGH`, `SPLIT LOW`, `PCTFREE`, `LEVEL2 PCTFREE`, `MINPCTUSED`, `CLUSTER` oder `ALLOW REVERSE SCANS` verwendet werden soll, müssen Sie als Erstes einen Index erstellen und die gewünschten Schlüssel und Parameter angeben. Verwenden Sie anschließend die Anweisung `ALTER TABLE`, um den Primärschlüssel oder den eindeutigen Schlüssel hinzuzufügen. Die Anweisung `ALTER TABLE` verwendet den bereits erstellten Index.

Sie können Indexstatistiken im Rahmen der Erstellung des Index erfassen. Zu dem Zeitpunkt, zu dem Sie die Anweisung `CREATE INDEX` ausführen, stehen die Schlüsselwertstatistiken und die physischen Statistiken zur Verfügung. Durch die Erfassung der Indexstatistiken mit der Anweisung `CREATE INDEX` brauchen Sie das Dienstprogramm `RUNSTATS` nicht unmittelbar im Anschluss an die Anweisung `CREATE INDEX` auszuführen.

Mit der folgenden SQL-Anweisung werden zum Beispiel die grundlegenden Indexstatistiken im Rahmen der Erstellung eines Index erfasst:

```
CREATE INDEX IDX1 ON TABL1 (COL1) COLLECT STATISTICS
```

Wenn Sie eine replizierte gespeicherte Abfragetabelle 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.

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.

In Mehrpartitionsumgebungen müssen eindeutige Indizes als Obermenge des Partitionierungsschlüssels definiert werden.

#### **Zugehörige Konzepte:**

- „Tipps zur Indexleistung“ in *Systemverwaltung: Optimierung*
- „Indexreorganisation“ in *Systemverwaltung: Optimierung*
- „Tabellen- und Indexverwaltung für Standardtabellen“ in *Systemverwaltung: Optimierung*
- „Defragmentierung des Onlineindex“ in *Systemverwaltung: Optimierung*

- „Tabellen- und Indexverwaltung für MDC-Tabellen“ in *Systemverwaltung: Optimierung*

**Zugehörige Tasks:**

- „Ändern von Tabellenattributen“ auf Seite 204

**Zugehörige Referenzen:**

- „max\_querydegree - Maximaler Grad der Parallelität bei Abfragen“ in *Systemverwaltung: Optimierung*
- „intra\_parallel - Partitionsinterne Parallelität aktivieren“ in *Systemverwaltung: Optimierung*
- „dft\_degree - Grad der Parallelität“ in *Systemverwaltung: Optimierung*
- „CREATE INDEX statement“ in *SQL Reference, Volume 2*

## Erstellen eines benutzerdefinierten erweiterten Indextyps

Zur Unterstützung benutzerdefinierter Indextypen 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. Im Unterschied zu 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 (Bereichsvergleichselementen) 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 erheblich verbessern lässt.

**Anmerkung:** Die benutzerdefinierte Funktionsdefinition muss deterministisch sein und darf die Ausführung externer Aktionen nicht zulassen, um vom Optimierungsprogramm genutzt werden zu können.

Eine optionale Datenfilterfunktion kann ebenfalls angegeben werden. Das Optimierungsprogramm wendet den Filter auf den abgerufenen Tupel an, bevor die benutzerdefinierte Funktion ausgewertet wird.

Nur eine Spalte mit einem strukturierten oder einzigartigen Datentyp kann die Indexerweiterung zum Erstellen eines benutzerdefinierten Indextyps für diese Objekte nutzen. Der Typ des benutzerdefinierten erweiterten Index darf die folgenden Merkmale nicht aufweisen:

- Definition mit Clusterindizes
- Vorhandensein von INCLUDE-Spalten

**Zugehörige Konzepte:**

- „Details zur Indexverwaltung“ auf Seite 156
- „Details zur Indexsuche“ auf Seite 156

- „Details zur Indexausnutzung“ auf Seite 157
- „Ein Szenario zum Definieren einer Indexerweiterung“ auf Seite 159

---

## Details zum Erstellen eines benutzerdefinierten erweiterten Indextyps

In diesem Abschnitt werden die verschiedenen Gesichtspunkte behandelt, die bei der Erstellung eines eigenen erweiterten Indextyps zu beachten sind.

### Details zur Indexverwaltung

Sie definieren zwei der Komponenten, die zum Funktionieren eines Index erforderlich sind, mit Hilfe der Anweisung `CREATE INDEX EXTENSION`.

Die Indexverwaltung ist der Prozess des Umsetzens des Inhalts der Indexpalte (bzw. des Quellschlüssels) 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 Quellschlüsselspalte an, die von dieser Indexerweiterung unterstützt wird. Für die Quellschlüsselspalte 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 wird.

#### Zugehörige Konzepte:

- „Erstellen eines benutzerdefinierten erweiterten Indextyps“ auf Seite 155

#### Zugehörige Referenzen:

- „`CREATE INDEX EXTENSION` statement“ in *SQL Reference, Volume 2*

### 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, welche die Ausgabe der benutzerdefinierten Tabellenfunktion bilden, die in der Klausel `GENERATE KEY USING` angegeben wird. Für die Zielschlüsselspalte wird ein einziger Parameternamen und Datentyp angegeben und zugeordnet. Dieser Parameter entspricht den Spalten der `RETURNS`-Tabelle 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 zugrunde liegenden 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 eine Verweismarke zu. Diese Verweismarke ist eine SQL-Kennung, die sich auf den in der Indexausnutzungsregel angegebenen Methodennamen bezieht. (Diese Regel befindet sich in der Klausel `PREDICATES` einer benutzerdefinierten Funktion.) Einer oder mehrere Parameternamen und Datentypen werden zur Verwendung als Argumente in der Bereichsfunktion mit oder ohne 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. Dadurch 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 Auswerten der ursprünglichen Methode gering ist, und die Selektivität des Sekundärfilters relativ niedrig ist.

#### Zugehörige Konzepte:

- „Erstellen eines benutzerdefinierten erweiterten Indextyps“ auf Seite 155
- „Details zur Indexverwaltung“ auf Seite 156
- „Details zur Indexausnutzung“ auf Seite 157

#### Zugehörige Referenzen:

- „`CREATE INDEX EXTENSION` statement“ in *SQL Reference, Volume 2*

## Details zur Indexausnutzung

Die Indexausnutzung findet bei der Auswertung der Suchmethode statt.

Die Anweisung `CREATE FUNCTION` (externe Skalarfunktion) erstellt ein benutzerdefiniertes Vergleichselement, das mit den für die Indexerweiterung definierten Suchmethoden verwendet wird.

Die Klausel `PREDICATES` identifiziert die Vergleichselemente mit dieser Funktion, die möglicherweise die Indexerweiterungen ausnutzen können (und die möglicherweise die optionale Klausel `SELECTIVITY` für die Suchbedingung des Vergleichselements 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 Vergleichselement mit einem Vergleichsoperator (`=`, `>`, `<` etc.) und einer Konstanten bzw. einem Ausdruck (mit der Klausel `EXPRESSION AS`). Wenn diese Funktion von einem Vergleichselement 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 der Ergebnistyp 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 Vergleichselement verwendeten) Funktion, die die Anzahl der Zeilen reduziert, für die das benutzerdefinierte Vergleichselement zur Feststellung übereinstimmender Zeilen ausgeführt werden muss. Sollten sich die durch den Index generierten Ergebnisse weitgehend mit den vom benutzerdefinierten Vergleichselement 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` gibt die Indexerweiterung an.
  - Die optionale Klausel `EXACT` gibt an, dass die Indexsuchfunktion bei der Auswertung der Vergleichselemente exakte Ergebnisse erzielt. Diese Klausel weist die Datenbank an, nach der Ausführung der Indexsuchfunktion nicht die ursprüngliche, vom Benutzer erstellte Vergleichselementfunktion oder die Filterfunktion anzuwenden. Wird die Indexsuchfunktion nicht eingesetzt, müssen das ursprüngliche Vergleichselement sowie die ursprünglichen Filterfunktionen angewendet werden. Wird die Klausel `EXACT` nicht verwendet, wird das ursprüngliche, vom Benutzer erstellte Vergleichselement nach der Ausführung der Indexsuchfunktion angewendet. Die Klausel `EXACT` ist nützlich, wenn die Indexsuchfunktion die gleichen Ergebnisse wie das Vergleichselement zurückgibt. Sie verhindert, dass bei der Abfrageausführung das benutzerdefinierte Vergleichselement auf die Ergebnisse der Indexsuchfunktion angewendet wird. Wenn zu erwarten ist, dass der Index nur ein annähernd mit dem Vergleichselement 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 gibt einen Parameternamen der Funktion an, die gerade definiert wird. Diese Klausel wird als "wahr" (`true`) ausgewertet, wenn die Werte des benannten Parameters Spalten sind, die von einem Index abgedeckt werden, der auf der angegebenen Indexerweiterung basiert.
  - Die Klausel `USE` definiert das Suchargument. Das Suchargument gibt die zu verwendende Methode an, die in der Indexerweiterung definiert ist. Der hier angegebene Methodename muss mit einer Methode übereinstimmen, die in der Indexerweiterung definiert ist. Der bzw. die Parameterwerte identifizieren Parameternamen der Funktion, die gerade definiert wird. 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. Die Übereinstimmung muss für integrierte und einzigartige Datentypen exakt sein und innerhalb derselben strukturierten Typen liegen.

### Zugehörige Konzepte:

- „Erstellen eines benutzerdefinierten erweiterten Indextyps“ auf Seite 155
- „Details zur Indexverwaltung“ auf Seite 156
- „Details zur Indexsuche“ auf Seite 156
- „Ein Szenario zum Definieren einer Indexerweiterung“ auf Seite 159

### Zugehörige Referenzen:

- „CREATE FUNCTION (External Scalar) statement“ in *SQL Reference, Volume 2*

## Ein Szenario zum Definieren einer Indexerweiterung

Das folgende Szenario veranschaulicht das Definieren einer Indexerweiterung:

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 dieses 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 Indexverwaltungskomponente eines Indexes entspricht.

```
CREATE INDEX EXTENSION iename (parm_name datatype, ...)
FROM SOURCE KEY (parm_name datatype)
GENERATE KEY USING tabellenfunktionsaufruf
...
```

Die Klausel `FROM SOURCE KEY` gibt den Parameter und den Datentyp der Schlüsselumsetzung an. 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 datatype, ...)
...
WITH TARGET KEY
WHEN methodenname (parm_name datatype, ...)
RANGE THROUGH aufruf_bereichserstellungsfunktion
FILTER USING aufruf_indexfilterfunktion
```

Mit der Klausel `WITH TARGET KEY` wird die Suchmethodendefinition angegeben. Die Klausel `WHEN` dient zur Angabe des Methodennamens. Die Klausel `RANGE THROUGH` identifiziert die Funktion, die zur Eingrenzung des Bereichs des zu verwendenden Index eingesetzt wird. Die Klausel `FILTER`

USING dient zur Angabe der Funktion, die zum Eliminieren nicht benötigter Elemente aus den resultierenden Indexwerten dient.

**Anmerkung:** Die Klausel FILTER USING kann zur Angabe eines CASE-Ausdrucks anstelle einer Indexfilterfunktion verwendet werden.

5. Definieren Sie die Vergleichselemente zur Ausnutzung 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 Vergleichselemente einführen, die bei jeder Klausel WHEN gestartet werden sollen. Die Klausel WHEN beginnt die Angabe für das Vergleichselement 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 Vergleichselement verwendeten) Funktion, die die Anzahl der Zeilen reduziert, für die das benutzerdefinierte Vergleichselement zur Ermittlung 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 Vergleichselement 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 durch das Vergleichselement 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 Anweisungszeichenfolge mit einer geeigneten Selektivität:

```
SELECT * FROM customer
  WHERE within(loc, circle(100, 100, 10)) = 1 SELECTIVITY .05
```

In diesem Beispiel filtert das angegebene Vergleichselement (SELECTIVITY .05) 95 % der Zeilen in der Tabelle customer aus.



**Zugehörige Konzepte:**

- „Erstellen eines benutzerdefinierten erweiterten Indextyps“ auf Seite 155
- „Details zur Indexverwaltung“ auf Seite 156
- „Details zur Indexsuche“ auf Seite 156
- „Details zur Indexausnutzung“ auf Seite 157

**Zugehörige Referenzen:**

- „CREATE INDEX EXTENSION statement“ in *SQL Reference, Volume 2*
- „CREATE FUNCTION (External Scalar) statement“ in *SQL Reference, Volume 2*

---

## Aufrufen des Konfigurationsadvisors über den Befehlszeilenprozessor

**Voraussetzungen:**

Die Datenbank wurde bereits erstellt.

**Vorgehensweise:**

Nach der Erstellung der Datenbank können Sie mit Hilfe des Befehls AUTOCONFIGURE den Konfigurationsadvisor aufrufen. Diese Vorgehensweise kann auch dann verwendet werden, wenn Sie bei der Datenbankerstellung die Option AUTOCONFIGURE ausgewählt haben.

Mit den verfügbaren Optionen von AUTOCONFIGURE können Sie Werte für verschiedene Konfigurationsparameter definieren und den Anwendungsbereich dieser Parameter festlegen. Der Anwendungsbereich (SCOPE) kann NONE sein, was bedeutet, dass keiner der Werte angewendet wird. DB ONLY bedeutet, dass nur die Werte für die Datenbankkonfiguration und den Pufferpool angewendet werden. Und DB AND DBM bedeutet, dass alle Parameter und ihre Werte angewendet werden.

**Zugehörige Konzepte:**

- „Konfigurationsparameter“ in *Systemverwaltung: Optimierung*

**Zugehörige Referenzen:**

- „AUTOCONFIGURE Command“ in *Command Reference*



---

## Kapitel 5. 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.

---

### Ändern eines Exemplars

Einige Zeit nach der Implementierung eines Datenbankentwurfs beabsichtigen Sie eventuell, diesen zu ändern. Daher sollten Sie die Hauptentwurfspunkte des vorherigen Entwurfs überdenken.

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-Speichertypen überdenken.

Im Rahmen der Lizenzverwaltung für Ihre DB2 Universal Database™-Produkte (DB2 UDB) kann es erforderlich werden, die Anzahl der Lizenzen zu erhöhen. Über die Lizenzzentrale innerhalb der Steuerzentrale können Sie die aktuelle Nutzung der installierten Produkte prüfen und gegebenenfalls die Anzahl der Lizenzen entsprechend aufstocken.

Die folgenden Aspekte sollten besonders berücksichtigt werden:

- „Ändern von Exemplaren (nur UNIX)“
- „Ändern der Knoten- und Datenbankkonfigurationsdatei“ auf Seite 168

### Ändern von Exemplaren (nur UNIX)

Die Exemplare wurden so gestaltet, dass sie durch ein nachfolgendes Installieren und 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 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® Universal Database (DB2 UDB) durch Installieren einer vorläufigen Programmkorrektur (PTF - Program Temporary Fix) aktualisiert wird, sollten alle vorhandenen DB2 UDB-Exemplare mit Hilfe des Befehls **db2iupdt** aktualisiert werden.

Machen Sie sich zunächst mit den vorhandenen Exemplaren und Datenbankpartitionsservern vertraut, bevor Sie versuchen, ein Exemplar zu ändern oder zu löschen.

#### Zugehörige Konzepte:

- „Exemplarerstellung“ auf Seite 18

#### Zugehörige Tasks:

- „Aktualisieren der Exemplarkonfiguration unter UNIX“ auf Seite 164
- „Entfernen von Exemplaren“ auf Seite 167

#### Zugehörige Referenzen:

- „db2iupdt - Update Instances Command“ in *Command Reference*

## Details zum Ändern von Exemplaren

Vor dem Ändern eines Exemplars sollten Sie alle vorhandenen Exemplare auflisten.

### Auflisten von Exemplaren

#### Vorgehensweise:

Gehen Sie wie folgt vor, um eine Liste aller auf dem System verfügbaren Exemplare über die Steuerzentrale abzurufen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Exemplare** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf den Exemplarordner, und wählen Sie im Kontextmenü **Hinzufügen** aus.
3. Klicken Sie im Fenster **Exemplar 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 eine Liste aller Exemplare anzuzeigen, die auf dem System verfügbar sind:

```
db2ilist
```

Geben Sie Folgendes ein, um festzustellen, welches Exemplar für die aktuelle Sitzung (auf unterstützten Windows-Plattformen) gilt:

```
set db2instance
```

### Aktualisieren der Exemplarkonfiguration unter UNIX

Der Befehl **db2iupdt** aktualisiert das angegebene Exemplar durch folgende Operationen:

- Die Dateien im Unterverzeichnis `sql1ib` im Ausgangsverzeichnis des Exemplareigners werden ersetzt.
- Wenn der Knotentyp 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 den neuen Knotentyp zusammengefügt. Beim Erstellen einer neuen Konfigurationsdatei für den Datenbankmanager wird die alte Datei im Verzeichnis `backup` des Unterverzeichnisses `sql1ib` im Ausgangsverzeichnis des Exemplareigners gesichert.

## Vorgehensweise:

Unter AIX befindet sich der Befehl **db2iupdt** im Verzeichnis /usr/opt/db2\_08\_01/instance/. Unter HP-UX, Solaris Operating Environment oder Linux befindet sich der Befehl **db2iupdt** im Verzeichnis /opt/IBM/db2/V8.1/instance/.

Der Befehl ist wie folgt anzugeben:

```
db2iupdt ExempName
```

Dabei ist ExempName der Anmeldename des Exemplareigners.

Für diesen Befehl sind weitere optionale Parameter verfügbar:

- -h oder -?  
Zeigt ein Hilfemenü für diesen Befehl an.
- -d  
Dieser Parameter konfiguriert den Debugmodus für die Fehlerbestimmung.
- -a AuthTyp  
Dieser Parameter gibt den Authentifizierungstyp für das Exemplar an. Gültige Authentifizierungstypen sind SERVER, SERVER\_ENCRYPT und CLIENT. Wenn dieser Parameter nicht angegeben wird, wird standardmäßig der Authentifizierungstyp SERVER verwendet, wenn ein DB2-Server installiert ist. Ansonsten wird die Option CLIENT verwendet. Der für das Exemplar angegebene Authentifizierungstyp gilt für alle Datenbanken, die zu diesem Exemplar gehören.
- -e  
Ermöglicht Ihnen die Aktualisierung aller vorhandenen Exemplare. Vorhandene Exemplare können mit **db2ilist** angezeigt werden.
- -u Abgeschirmt-ID  
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:** Abgeschirmt-ID kann nicht „root“ oder „bin“ sein.
- -k  
Dieser Parameter behält den aktuellen Exemplartyp bei. Wenn Sie diesen Parameter nicht angeben, wird das aktuelle Exemplar in der nachfolgend angegebenen Reihenfolge auf den höchsten verfügbaren Exemplartyp hochgestuft:
  - Partitionierter Datenbankserver mit lokalen und fernen Clients (Standardexemplartyp für DB2 Enterprise - Extended Edition)
  - Datenbankserver mit lokalen und fernen Clients (Standardexemplartyp für DB2 Universal Database Enterprise Server Edition)
  - Client (Standardexemplartyp für DB2-Client)

Beispiele:

- Wenn Sie nach dem Erstellen des Exemplars DB2 Universal Database Workgroup Server Edition oder DB2 Universal Database Enterprise Server 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 auch als

abgeschirmte ID verwenden:

```
db2iupdt -u db2inst1 db2inst1
```

- Mit dem folgenden Befehl können Sie Clientexemplare aktualisieren:

```
db2iupdt db2inst1
```

#### Zugehörige Tasks:

- „Entfernen von Exemplaren“ auf Seite 167

#### Zugehörige Referenzen:

- „db2ilist - List Instances Command“ in *Command Reference*
- „db2iupdt - Update Instances Command“ in *Command Reference*

### Aktualisieren der Exemplarkonfiguration unter Windows

Der Befehl **db2iupdt** aktualisiert das angegebene Exemplar durch folgende Operationen:

- Die Dateien im Unterverzeichnis `sql1ib` im Ausgangsverzeichnis des Exemplareigners werden ersetzt.
- Wenn der Knotentyp 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 den neuen Knotentyp zusammengefügt. Beim Erstellen einer neuen Konfigurationsdatei für den Datenbankmanager wird die alte Datei im Verzeichnis `backup` des Unterverzeichnisses `sql1ib` im Ausgangsverzeichnis des Exemplareigners gesichert.

#### Vorgehensweise:

Der Befehl **db2iupdt** befindet sich im Verzeichnis `\sql1ib\bin`.

Der Befehl ist wie folgt anzugeben:

```
db2iupdt ExempName
```

Dabei ist `ExempName` der Anmeldename des Exemplareigners.

Für diesen Befehl sind weitere optionale Parameter verfügbar:

- `/h: hostname`  
Überschreibt den Standard-TCP/IP-Hostnamen, wenn es einen oder mehrere TCP/IP-Hostnamen für die aktuelle Maschine gibt.
- `/p: exemplarprofilpfad`  
Gibt den neuen Exemplarprofilpfad für das aktualisierte Exemplar an.
- `/r: anfangsport,endport`  
Gibt den Bereich der TCP/IP-Ports an, die von dem partitionierten Datenbankexemplar bei der Ausführung mit mehreren Partitionen verwendet werden.
- `/u: benutzername,kennwort`  
Gibt den Kontonamen und das Kennwort für den DB2-Service (Dienst) an.

#### Zugehörige Tasks:

- „Auflisten von Exemplaren“ auf Seite 27
- „Aktualisieren der Exemplarkonfiguration unter UNIX“ auf Seite 164
- „Entfernen von Exemplaren“ auf Seite 167

## Entfernen von Exemplaren

### Vorgehensweise:

Gehen Sie wie folgt vor, um ein Exemplar über die Steuerzentrale 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 den Exemplarnamen und wählen Sie im Kontextmenü **Entfernen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie auf **OK**.

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

```
db2idrop <exemplarname>
```

Zum Entfernen 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 Registriervariablen DB2INSTPROF angegebene Exemplarverzeichnis.

Unter UNIX-Betriebssystemen könnte es sinnvoll sein, die Dateien im Verzeichnis `INSTHOME/sql1ib` 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 Rootberechtigung 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 (optional) als Benutzer mit Rootberechtigung 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 optional, weil die Benutzer-ID und die Gruppe des Exemplareigners weiterhin für andere Zwecke verwendet werden können.

Der Befehl **db2idrop** entfernt den Exempleintrag aus der Exemplarliste und löscht das Unterverzeichnis `sql1ib` im Ausgangsverzeichnis des Exemplareigners.

**Anmerkung:** Wenn Sie unter UNIX-Betriebssystemen versuchen, ein Exemplar mit Hilfe des Befehls **db2idrop** zu löschen, wird eine Nachricht generiert, die besagt, dass das Unterverzeichnis `sql1ib` nicht gelöscht werden kann, und im Unterverzeichnis `adm` werden verschiedene Dateien mit

der Erweiterung `.nfs` generiert. Das Unterverzeichnis `adm` ist ein über NFS angehängtes System, und die Dateien werden auf dem Server gesteuert. Sie müssen die Dateien `*.nfs` von der Position auf dem Dateiserver löschen, an der das Verzeichnis angehängt ist. Anschließend können Sie das Unterverzeichnis `sql1ib` entfernen.

**Zugehörige Referenzen:**

- „`db2stop` - Stop DB2 Command“ in *Command Reference*
- „`TERMINATE` Command“ in *Command Reference*
- „`STOP DATABASE MANAGER` Command“ in *Command Reference*
- „`db2idrop` - Remove Instance Command“ in *Command Reference*
- „`db2ilist` - List Instances Command“ in *Command Reference*

## Ändern der Knoten- und Datenbankkonfigurationsdatei

Verwenden Sie zum Aktualisieren der Datenbankkonfigurationsdatei den Konfigurationsadvisor der Steuerzentrale, oder führen Sie den Befehl `db2 autoconfigure` mit den entsprechenden Optionen aus. Der Konfigurationsadvisor hilft Ihnen bei der Optimierung der Leistung und der ausgewogenen Verteilung des Speicherbedarfs für eine einzelne Datenbank pro Exemplar und macht 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 Ereignisse stattfinden:

- 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 Konfigurationsadvisor empfohlenen Werte zu einer besseren Leistung als die Standardwerte, weil sie auf den Informationen zur Auslastung Ihres Systems und zu Ihrem speziellen Server basieren. Beachten Sie jedoch, dass die Werte die Leistung Ihres Datenbanksystems verbessern, aber nicht unbedingt optimieren können. Sie sollten deshalb als Ausgangspunkt für weitere Optimierungsmaßnahmen verstanden werden.

**Voraussetzungen:**

Wenn Sie Änderungen an bestimmten Datenbankpartitionsgruppen (Hinzufügen oder Löschen von Partitionen oder Versetzen vorhandener Partitionen) planen, muss die Knotenkonfigurationsdatei aktualisiert werden.

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.

**Vorgehensweise:**

Gehen Sie wie folgt vor, um die Datenbankkonfiguration über die Steuerzentrale zu aktualisieren:



1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Datenbanken** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf das zu ändernde Exemplar bzw. die zu ändernde Datenbank, und wählen Sie dann **Konfigurationsadvisor** aus.
3. Klicken Sie auf die einzelnen Seiten, und nehmen Sie die erforderlichen Änderungen an den Informationen vor.
4. Klicken Sie auf die Seite **Ergebnisse**, um die vorgeschlagenen Änderungen an den Konfigurationsparametern anzuzeigen.
5. Wenn Sie zur Anwendung oder Sicherung der Aktualisierungen bereit sind, klicken Sie auf **Fertig stellen**.

Wenn Sie den Konfigurationsadvisor über die Befehlszeile verwenden wollen, geben Sie den Befehl AUTOCONFIGURE ein.

Geben Sie in der Befehlszeile Folgendes ein, um einzelne Parameter in der Konfiguration des Datenbankmanagers zu aktualisieren:

```
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 Ausgeben der aktuellen Konfigurationsparameter für den Datenbankmanager können Sie den Befehl GET DATABASE MANAGER CONFIGURATION verwenden.

#### Zugehörige Konzepte:

- „Vergleichstests“ in *Systemverwaltung: Optimierung*

#### Zugehörige Tasks:

- „Ändern der Datenbankkonfiguration über mehrere Partitionen“ auf Seite 169
- „Konfigurieren von DB2 mit Konfigurationsparametern“ in *Systemverwaltung: Optimierung*

#### Zugehörige Referenzen:

- „GET DATABASE MANAGER CONFIGURATION Command“ in *Command Reference*
- „UPDATE DATABASE MANAGER CONFIGURATION Command“ in *Command Reference*

## Ändern der Datenbankkonfiguration über mehrere Partitionen

### Vorgehensweise:

Wenn Sie eine Datenbank haben, die in mehrere Partitionen 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 Knotenkonfigurationsdatei

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 den Befehl **db2\_all**, um die Konfigurationsdateien in allen Datenbankpartitionen zu verwalten.

**Zugehörige Konzepte:**

- „Absetzen von Befehlen in einer partitionierten Datenbankumgebung“ auf Seite 381

**Zugehörige Tasks:**

- „Ändern der Knoten- und Datenbankkonfigurationsdatei“ auf Seite 168

---

## Ändern einer Datenbank

Es gibt fast ebenso viele Operationen zum Ändern von Datenbanken wie zum Erstellen von Datenbanken. Diese Operationen aktualisieren oder löschen Objekte der zuvor erstellten Datenbank. Dazu gehören:

- „Löschen einer Datenbank“
- „Ändern einer Datenbankpartitionsgruppe“ auf Seite 171
- „Ändern eines Tabellenbereichs“ auf Seite 171
- „Löschen eines Schemas“ auf Seite 182
- Kapitel 6, „Ändern von Tabellen und anderen zugehörigen Tabellenobjekten“, auf Seite 185
- „Ändern eines benutzerdefinierten strukturierten Typs“ auf Seite 208
- „Löschen und Aktualisieren von Zeilen einer typisierten Tabelle“ auf Seite 209
- „Umbenennen einer vorhandenen Tabelle oder eines Index“ auf Seite 209
- „Löschen einer Tabelle“ auf Seite 212
- „Löschen einer benutzerdefinierten temporären Tabelle“ auf Seite 213
- „Löschen eines Auslösers“ auf Seite 214
- „Löschen einer benutzerdefinierten Funktion (UDF), Funktionszuordnung oder Methode“ auf Seite 214
- „Löschen eines benutzerdefinierten Datentyps (UDT) oder einer Typenzuordnung“ auf Seite 215
- „Ändern oder Löschen einer Sicht“ auf Seite 216
- „Wiederherstellen unbrauchbarer Sichten“ auf Seite 218
- „Löschen einer gespeicherten Abfragetabelle oder einer Zwischenspeichertabelle“ auf Seite 218
- „Wiederherstellen unbrauchbarer Übersichtstabellen“ auf Seite 219
- „Löschen eines Index, einer Indexerweiterung oder einer Indexspezifikation“ auf Seite 220
- „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 221

## Löschen einer Datenbank

**Vorgehensweise:**

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 eine Datenbank über die Steuerzentrale 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 die Datenbank, die gelöscht werden soll, und wählen Sie im Kontextmenü **Löschen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie 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, die Beispieldatenbank zum weiteren Experimentieren zu nutzen, löschen Sie sie nicht. Wenn Sie die Beispieldatenbank SAMPLE gelöscht haben, sie jedoch erneut benötigen, können Sie sie erneut erstellen.

**Zugehörige Referenzen:**

- „GET SNAPSHOT Command“ in *Command Reference*
- „DROP DATABASE Command“ in *Command Reference*
- „LIST ACTIVE DATABASES Command“ in *Command Reference*

## Ändern einer Datenbankpartitionsgruppe

**Vorgehensweise:**

Wenn Sie Partitionen hinzufügen oder löschen, müssen Sie die aktuellen Daten auf die neuen Partitionen in der Datenbankpartitionsgruppe verteilen. Führen Sie dazu den Befehl REDISTRIBUTE DATABASE PARTITION GROUP aus.

**Zugehörige Konzepte:**

- „Umverteilung von Daten“ in *Systemverwaltung: Optimierung*
- „Verwalten der Datenbankserverkapazität“ in *Systemverwaltung: Optimierung*

**Zugehörige Tasks:**

- „Umverteilen von Daten auf Partitionen“ in *Systemverwaltung: Optimierung*

**Zugehörige Referenzen:**

- „REDISTRIBUTE DATABASE PARTITION GROUP Command“ in *Command Reference*

## Ändern eines Tabellenbereichs

**Vorgehensweise:**

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 weder löschen noch einen anderen erstellen. Außerdem muss immer mindestens ein temporärer Systemtabellenbereich mit einer Seitengröße von 4 KB vorhanden sein. Sie können andere temporäre Systemtabellenbereiche erstellen. Nach der Erstellung können auch die Seitengröße und der EXTENTSIZE-Wert eines Tabellenbereichs nicht mehr geändert werden.

#### **Zugehörige Tasks:**

- „Hinzufügen eines Behälters zu einem DMS-Tabellenbereich“ auf Seite 172
- „Modifizieren von Behältern in einem DMS-Tabellenbereich“ auf Seite 173
- „Hinzufügen eines Behälters in einem SMS-Tabellenbereich einer Partition“ auf Seite 177
- „Umbenennen eines Tabellenbereichs“ auf Seite 178
- „Löschen eines Benutzertabellenbereichs“ auf Seite 179
- „Löschen eines temporären Systemtabellenbereichs“ auf Seite 180
- „Löschen eines temporären Benutzertabellenbereichs“ auf Seite 182

#### **Zugehörige Referenzen:**

- „ALTER TABLESPACE statement“ in *SQL Reference, Volume 2*

## **Details zum Ändern eines Tabellenbereichs**

Dieser Abschnitt behandelt die Tasks, die mit der Änderungen von Tabellenbereichen verbunden sind.

### **Hinzufügen eines Behälters zu einem DMS-Tabellenbereich**

#### **Vorgehensweise:**

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.

Wenn einem Tabellenbereich neue Behälter hinzugefügt oder vorhandene Behälter erweitert werden, kann ein Neuausgleich der Tabellenbereichsdaten stattfinden. Der Prozess des Neuausgleichs beinhaltet ein Versetzen von `EXTENTSIZE` großen Teilen von Tabellenbereichen von einer Position an eine andere. Während dieses Prozesses wird versucht, die einheitenübergreifende Speicherung (Striping) der Daten innerhalb des Tabellenbereichs beizubehalten. Der Neuausgleich findet nicht unbedingt über alle Behälter hinweg statt, sondern hängt von vielen Faktoren ab, zu denen die vorhandene Behälterkonfiguration, die Größe der neuen Behälter und die Füllungsgrad des Tabellenbereichs zählen.

Wenn Behälter einem vorhandenen Tabellenbereich hinzugefügt werden, können sie so hinzugefügt werden, dass sie nicht in Stripe 0 (d. h. im ersten einheitenübergreifend gespeicherten Datenblock) beginnen. An welcher Stelle in der Zuordnung sie beginnen, wird durch den Datenbankmanager festgelegt und hängt von der Größe der hinzugefügten Behälter ab. Wenn der Behälter, der hinzugefügt wird, nicht ausreichend groß ist, wird er so positioniert, dass er im letzten Stripe der Zuordnung endet. Wenn er groß genug ist, wird er so positioniert, dass er in Stripe 0 beginnt.

Wenn Sie neue Behälter hinzufügen und ein neues Stripeset erstellen, findet kein Neuausgleich statt. Ein neues Stripeset wird mit der Klausel `BEGIN NEW STRIPE SET` in der Anweisung `ALTER TABLESPACE` erstellt. Außerdem können Sie Behälter vorhandenen Stripesets hinzufügen, indem Sie die Klausel `ADD TO STRIPE SET` in der Anweisung `ALTER TABLESPACE` verwenden.

Während des Neuausgleichs ist 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 einem DMS-Tabellenbereich einen Behälter über die Steuerzentrale hinzuzufügen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellenbereiche** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf den Tabellenbereich, dem der Behälter hinzugefügt werden soll, und wählen Sie im Kontextmenü **Ändern** aus.
3. Klicken Sie auf **Hinzufügen**, geben Sie die erforderlichen Informationen ein und klicken Sie auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um einem DMS-Tabellenbereich einen Behälter hinzuzufügen:

```
ALTER TABLESPACE <name>
  ADD (DEVICE '<pfad>' <größe>, FILE '<dateiname>' <größe>)
```

Das folgende Beispiel zeigt, wie einem Tabellenbereich auf einem UNIX-System zwei neue Einheitenbehälter (mit jeweils 10 000 Seiten) hinzugefügt werden.

```
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 zu ändern, die sich auf die Leistung auswirken können.

#### Zugehörige Konzepte:

- „Auswirkung von Tabellenbereichen auf die Abfrageoptimierung“ in *Systemverwaltung: Optimierung*
- „Hinzufügen und Erweitern von Behältern in DMS-Tabellenbereichen“ in *Systemverwaltung: Konzept*

#### Zugehörige Tasks:

- „Hinzufügen eines Behälters in einem SMS-Tabellenbereich einer Partition“ auf Seite 177

#### Zugehörige Referenzen:

- „ALTER TABLESPACE statement“ in *SQL Reference, Volume 2*

## Modifizieren von Behältern in einem DMS-Tabellenbereich

#### Einschränkungen:

Jede unformatierte Einheit (raw) kann lediglich als ein Behälter verwendet werden. Die unformatierte Einheit ist nach ihrer Erstellung festgelegt. Wenn Sie beabsichti-

gen, einen unformatierten Einheitenbehälter mit Hilfe der Option zur Änderung der Größe bzw. zur Erweiterung zu vergrößern, sollten Sie die Größe der unformatierten Einheit zunächst überprüfen, um sicherzugehen, dass Sie nicht versuchen, den Einheitenbehälter über die Größe der unformatierten Einheit hinaus zu vergrößern.

### Vorgehensweise:

Sie können die Größe eines DMS-Tabellenbereichs (d. h. eines mit der Klausel `MANAGED BY DATABASE` erstellten Tabellenbereichs) ändern.

Gehen Sie wie folgt vor, um einen oder mehrere Behälter in einem DMS-Tabellenbereich über die Steuerzentrale zu vergrößern:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellenbereiche** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf den Tabellenbereich, dem der Behälter hinzugefügt werden soll, und wählen Sie im Kontextmenü **Ändern** aus.
3. Klicken Sie auf **Größe ändern**, geben Sie die erforderlichen Informationen ein und klicken Sie auf **OK**.

Sie haben auch die Möglichkeit, vorhandene Behälter aus einem DMS-Tabellenbereich zu löschen, die Größe vorhandener Behälter in einem DMS-Tabellenbereich zu verringern und einem DMS-Tabellenbereich neue Behälter hinzuzufügen, ohne dass ein Neuausgleich der Daten über alle Behälter durchgeführt wird.

Das Löschen vorhandener Tabellenbereichsbehälter sowie das Verkleinern vorhandener Behälter ist nur zulässig, wenn die Anzahl von `EXTENTSIZE` großen Speicherbereichen, die gelöscht oder verkleinert werden sollen, kleiner als oder gleich der Anzahl der freien `EXTENTSIZE` großen Speicherbereiche oberhalb der oberen Grenze im Tabellenbereich ist. Die obere Grenze ist die Seitennummer der höchsten zugeordneten Seite im Tabellenbereich. Diese Grenze ist nicht identisch mit dem Wert für verwendete Seiten im Tabellenbereich, weil einige der Speicherbereiche unterhalb der oberen Grenze zur Wiederverwendung verfügbar gemacht worden sein können.

Die Anzahl der freien `EXTENTSIZE` großen Speicherbereiche oberhalb der oberen Grenze spielt eine wichtige Rolle, da sich alle Speicherbereiche bis zur oberen Grenze (einschließlich) an der gleichen logischen Position innerhalb des Tabellenbereichs befinden müssen. Der sich ergebende Tabellenbereich muss ausreichend Platz haben, um alle Daten aufnehmen zu können. Wenn nicht genügend freier Speicherplatz verfügbar ist, wird eine Fehlermeldung (`SQL20170N, SQLSTATE 57059`) zurückgegeben.

Zum Löschen von Behältern wird die Option `DROP` in der Anweisung `ALTER TABLESPACE` verwendet. Beispiel:

```
ALTER TABLESPACE TS1 DROP (FILE 'date1', DEVICE '/dev/rdisk1')
```

Zum Verkleinern vorhandener Behälter können Sie entweder die Option `RESIZE` oder die Option `REDUCE` verwenden. Bei Verwendung der Option `RESIZE` müssen alle Behälter, die in der Anweisung aufgelistet werden, entweder vergrößert oder verkleinert werden. Sie können nicht innerhalb derselben Anweisung einige Behälter vergrößern und andere Behälter verkleinern. Sie sollten die Methode der Größenänderung (`RESIZE`) in Betracht ziehen, wenn Sie die neue untere Begrenzung für die Größe des Behälters kennen. Sie sollten die Methode der Verringerung

(REDUCE) in Betracht ziehen, wenn Sie die aktuelle Größe des Behälters nicht kennen (oder diese für Sie keine Rolle spielt).

Geben Sie Folgendes in eine Befehlszeile ein, um einen oder mehrere Behälter in einem DMS-Tabellenbereich zu verkleinern:

```
ALTER TABLESPACE <name>
  REDUCE (FILE '<dateiname>' <größe>)
```

Das folgende Beispiel zeigt, wie ein Dateibehälter (der bereits mit einer Größe von 1 000 Seiten vorhanden ist) in einem Tabellenbereich auf einem Windows-System verkleinert wird:

```
ALTER TABLESPACE PAYROLL
  REDUCE (FILE 'd:\hldr\finance' 200)
```

Durch diese Aktion wird die Datei von 1 000 Seiten auf die Größe von 800 Seiten verkleinert.

Geben Sie Folgendes in eine Befehlszeile ein, um einen oder mehrere Behälter in einem DMS-Tabellenbereich zu vergrößern:

```
ALTER TABLESPACE <name>
  RESIZE (DEVICE '<pfad>' <größe>)
```

Das folgende Beispiel zeigt, wie zwei Einheitenbehälter (die bereits mit einer Größe von 1 000 Seiten vorhanden sind) in einem Tabellenbereich auf einem UNIX-System vergrößert werden:

```
ALTER TABLESPACE HISTORY
  RESIZE (DEVICE '/dev/rhd7' 2000,
         DEVICE '/dev/rhd8' 2000)
```

Durch diese Aktion werden die beiden Einheitenbehälter von 1 000 Seiten auf 2 000 Seiten vergrößert. Der Inhalt des Tabellenbereichs kann über die Behälter neu ausgeglichen werden. Während des Neuausgleichs ist 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 (FILE '<dateiname>' <größe>)
```

Das folgende Beispiel zeigt, wie Dateibehälter (die jeweils in einer Größe von 1 000 Seiten bereits vorhanden sind) in einem Tabellenbereich auf einem Windows-System vergrößert werden:

```
ALTER TABLESPACE PERSNEL
  EXTEND (FILE 'e:\wrkhist1' 200
         FILE 'f:\wrkhist2' 200)
```

Durch diese Aktion werden die beiden Dateien von 1 000 Seiten auf 1 200 Seiten vergrößert. Der Inhalt des Tabellenbereichs kann über die Behälter neu ausgeglichen werden. Während dieses Neuausgleichs 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. Die Anzahl der Vorablesefunktionen wird über den Datenbankkonfigurationsparameter *num\_ioservers* gesteuert. Die Datenbank muss gestoppt werden, damit der neue Parameterwert in Kraft treten kann. Das heißt, dass alle Anwendungen und Benutzer die Verbindung zur Datenbank trennen müssen, damit die Änderung wirksam werden kann.

Beachten Sie, dass die Anweisung ALTER TABLESPACE es ermöglicht, auch andere Merkmale des Tabellenbereichs zu ändern, die sich auf die Leistung auswirken können.

#### Zugehörige Referenzen:

- „ALTER TABLESPACE statement“ in *SQL Reference, Volume 2*

### Automatische Anpassung von PREFETCHSIZE nach Hinzufügen oder Löschen von Behältern

Wenn Sie mit der Möglichkeit rechnen, dass Sie eventuell vergessen, den Wert für PREFETCHSIZE eines Tabellenbereichs nach dem Hinzufügen oder Löschen von Behältern anzupassen, sollten Sie in Betracht ziehen, den Wert für PREFETCHSIZE durch den Datenbankmanager automatisch bestimmen zu lassen. Wenn Sie die Anpassung des PREFETCHSIZE-Werts vergessen, kann es zu merklichen Leistungseinbußen in der Datenbank kommen.

DB2® Universal Database (DB2 UDB) ist so konfiguriert, dass die automatische Bestimmung des Werts für PREFETCHSIZE die Standardeinstellung für alle Tabellenbereiche ist, die unter Version 8.2 (und späteren Versionen) erstellt werden. Der Datenbankmanager arbeitet nach der folgenden Formel, um den Wert für PREFETCHSIZE für einen Tabellenbereich zu berechnen:

$$\text{PREFETCHSIZE} = (\text{Anzahl Behälter}) \times (\text{Anzahl physischer Spindeln pro Behälter}) \times \text{EXTENTSIZ}$$

Es gibt drei Möglichkeiten, den PREFETCHSIZE-Wert für einen Tabellenbereich nicht automatisch bestimmen zu lassen:

- Erstellen Sie den Tabellenbereich mit einem spezifischen Wert für PREFETCHSIZE. Durch die manuelle Auswahl eines Werts für PREFETCHSIZE geben Sie an, dass Sie daran denken werden, den Wert für PREFETCHSIZE bei Bedarf anzupassen, wenn eine Änderung der dem Tabellenbereich zugeordneten Anzahl von Behältern erfolgt.
- Verwenden Sie keinen Wert für PREFETCHSIZE bei der Erstellung des Tabellenbereichs und setzen Sie den Wert des Datenbankkonfigurationsparameters *dft\_prefetch\_sz* auf einen anderen Wert als AUTOMATIC. DB2 UDB überprüft diesen Parameter, wenn kein expliziter Wert für PREFETCHSIZE bei der Erstellung eines Tabellenbereichs angegeben wird. Wenn ein anderer Wert als AUTOMATIC festgelegt wird, wird dieser als Standardwert für PREFETCHSIZE verwendet.



In diesem Fall müssen Sie daran denken, bei Bedarf den Wert für PREFETCHSIZE anzupassen, wenn eine Änderung der dem Tabellenbereich zugeordneten Anzahl von Behältern erfolgt.

- Ändern Sie den Wert für PREFETCHSIZE manuell mit Hilfe einer Anweisung ALTER TABLESPACE.

### Verwendung der Registriervariablen DB2\_PARALLEL\_IO

Vorablesezugriffsanforderungen werden in mehrere kleinere Vorablesezugriffsanforderungen gemäß der Parallelität eines Tabellenbereichs und vor der Übergabe an die Vorablesewarteschlangen aufgeteilt. Die Registriervariable DB2\_PARALLEL\_IO dient zur Definition der Anzahl physischer Spindeln pro Behälter sowie zur Beeinflussung der parallelen Ein-/Ausgabe für den Tabellenbereich. Bei inaktivierter paralleler E/A entspricht die Parallelität eines Tabellenbereichs der Anzahl von Behältern. Bei aktivierter paralleler E/A ist die Parallelität eines Tabellenbereichs gleich der Anzahl von Behältern multipliziert mit dem Wert, der in der Registriervariablen DB2\_PARALLEL\_IO angegeben ist. (Dies entspricht anders formuliert der Aussage, dass die Parallelität des Tabellenbereichs gleich dem Wert für PREFETCHSIZE dividiert durch den Wert für EXTENTSIZE des Tabellenbereichs ist.)

Im Folgenden wird die Beeinflussung des Werts für PREFETCHSIZE durch die Registriervariable DB2\_PARALLEL\_IO anhand einiger Beispiele erläutert. (Nehmen Sie an, dass alle Tabellenbereiche mit dem Wert AUTOMATIC für PREFETCHSIZE definiert wurden.)

- DB2\_PARALLEL\_IO=\*
  - Alle Tabellenbereiche verwenden den Standardwert, bei dem die Anzahl von Spindeln für jeden Behälter gleich 6 ist. Der Wert für PREFETCHSIZE wird bei aktivierter paralleler E/A 6-mal größer sein.
  - Für alle Tabellenbereiche wird die parallele E/A aktiviert. Die Vorablesezugriffsanforderung wird in mehrere kleinere Anforderungen unterteilt, jeweils auf den Wert für PREFETCHSIZE dividiert durch den Wert für EXTENTSIZE (bzw. auf die Anzahl Behälter multipliziert mit der Anzahl Spindeln).
- DB2\_PARALLEL\_IO=\*:3
  - Alle Tabellenbereiche verwenden 3 als Anzahl Spindeln pro Behälter.
  - Für alle Tabellenbereiche wird die parallele E/A aktiviert.
- DB2\_PARALLEL\_IO=\*:3,1:1
  - Alle Tabellenbereiche verwenden 3 als Anzahl Spindeln pro Behälter, mit Ausnahme von Tabellenbereich 1, der den Wert 1 verwendet.
  - Für alle Tabellenbereiche wird die parallele E/A aktiviert.

#### Zugehörige Tasks:

- „Ändern eines Tabellenbereichs“ auf Seite 171
- „Hinzufügen eines Behälters zu einem DMS-Tabellenbereich“ auf Seite 172
- „Modifizieren von Behältern in einem DMS-Tabellenbereich“ auf Seite 173

### Hinzufügen eines Behälters in einem SMS-Tabellenbereich einer Partition

#### Einschränkungen:

Sie können einen Behälter nur einem SMS-Tabellenbereich in einer Partition hinzufügen, der zurzeit keine Behälter besitzt.

### Vorgehensweise:

Geben Sie in der Befehlszeile Folgendes ein, um einem SMS-Tabellenbereich einen Behälter hinzuzufügen:

```
ALTER TABLESPACE <name>
  ADD ('<pfad>')
  ON DBPARTITIONNUM (<partitionsnummer>)
```

Die durch die Nummer angegebene Partition und jede Partition (oder jeder Knoten) in dem Bereich der Partitionen muss in der Datenbankpartitionsgruppe vorhanden sein, in der der Tabellenbereich definiert ist. Eine Partitionsnummer kann nur explizit oder innerhalb eines Bereichs in genau einer Klausel ON DBPARTITIONNUM für die Anweisung angegeben werden.

Das folgende Beispiel zeigt, wie ein neuer Behälter der Partitionsnummer 3 der Datenbankpartitionsgruppe hinzugefügt wird, die vom Tabellenbereich „plans“ auf einem UNIX-Betriebssystem verwendet wird:

```
ALTER TABLESPACE plans
  ADD ('/dev/rhdisk0')
  ON DBPARTITIONNUM (3)
```

### Zugehörige Tasks:

- „Hinzufügen eines Behälters zu einem DMS-Tabellenbereich“ auf Seite 172
- „Modifizieren von Behältern in einem DMS-Tabellenbereich“ auf Seite 173

### Zugehörige Referenzen:

- „ALTER TABLESPACE statement“ in *SQL Reference, Volume 2*

## Umbenennen eines Tabellenbereichs

### Einschränkungen:

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.

### Vorgehensweise:

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 geändert, die auf diesen Tabellenbereich verweisen.

### Zugehörige Referenzen:

- „RENAME TABLESPACE statement“ in *SQL Reference, Volume 2*

## Umschalten des Status eines Tabellenbereichs

### Vorgehensweise:

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.

```
db2 ALTER TABLESPACE <name>
    SWITCH ONLINE
```

### Zugehörige Referenzen:

- „ALTER TABLESPACE statement“ in *SQL Reference, Volume 2*

## Löschen eines Benutzertabellenbereichs

### Vorgehensweise:

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.

Gehen Sie wie folgt vor, um einen Benutzertabellenbereich über die Steuerzentrale 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 den Tabellenbereich, der gelöscht werden soll, und wählen Sie im Kontextmenü **Löschen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie 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
```

#### **Zugehörige Tasks:**

- „Löschen eines temporären Systemtabellenbereichs“ auf Seite 180
- „Löschen eines temporären Benutzertabellenbereichs“ auf Seite 182

#### **Zugehörige Referenzen:**

- „COMMIT statement“ in *SQL Reference, Volume 2*
- „DROP statement“ in *SQL Reference, Volume 2*

## **Löschen eines temporären Systemtabellenbereichs**

#### **Einschränkungen:**

Sie können einen temporären Systemtabellenbereich, der eine Seitengröße von 4 KB hat, nur löschen, wenn Sie zuvor einen anderen temporären Systemtabellenbereich erstellt haben. Der neue temporäre Systemtabellenbereich muss eine Seitengröße von 4 KB haben, weil die Datenbank immer über mindestens einen temporären Systemtabellenbereich verfügen muss, der eine Seitengröße von 4 KB hat. Wenn Sie zum Beispiel einen einzigen temporären Systemtabellenbereich mit einer Seitengröße von 4 KB haben und diesem einen Behälter hinzufügen wollen und es sich um einen SMS-Tabellenbereich handelt, müssen Sie zunächst einen neuen temporären Systemtabellenbereich mit einer Seitengröße von 4 KB mit der entsprechenden Anzahl von Behältern hinzufügen, und anschließend den alten temporären Systemtabellenbereich löschen. (Wenn Sie mit DMS-Tabellenbereichen arbeiten, können Sie einen Behälter hinzufügen, ohne den Tabellenbereich löschen und erneut erstellen zu müssen.)

#### **Vorgehensweise:**

Die Standardseitengröße für Tabellenbereiche beträgt 4 KB.

Gehen Sie wie folgt vor, um einen Systemtabellenbereich über die Steuerzentrale 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 den Ordner **Tabellenbereiche** und wählen Sie im Kontextmenü **Erstellen** —> **Tabellenbereich mit Assistent** aus. Andernfalls können Sie sofort mit Schritt 4 fortfahren.
3. Führen Sie die im Assistenten aufgeführten Schritte aus, um ggf. den neuen temporären Systemtabellenbereich zu erstellen.
4. Klicken Sie erneut auf den Ordner **Tabellenbereiche**, um eine Liste der Tabellenbereiche rechts im Fenster (Inhaltsteilfenster) anzuzeigen.
5. Klicken Sie mit der rechten Maustaste auf den temporären Systemtabellenbereich, der gelöscht werden soll, und klicken Sie im Kontextmenü auf **Löschen**.
6. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie auf **OK**.

Die folgende Anweisung dient zur Erstellung eines temporären Systemtabellenbereichs:

```
CREATE SYSTEM TEMPORARY TABLESPACE <name>  
    MANAGED BY SYSTEM USING ('<verzeichnisse>')
```

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:\systemp2')
```

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.

#### **Zugehörige Tasks:**

- „Löschen eines Benutzertabellenbereichs“ auf Seite 179
- „Löschen eines temporären Benutzertabellenbereichs“ auf Seite 182

#### **Zugehörige Referenzen:**

- „CREATE TABLESPACE statement“ in *SQL Reference, Volume 2*
- „DROP statement“ in *SQL Reference, Volume 2*

## Löschen eines temporären Benutzertabellenbereichs

### Vorgehensweise:

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.

### Zugehörige Tasks:

- „Löschen eines Benutzertabellenbereichs“ auf Seite 179
- „Löschen eines temporären Systemtabellenbereichs“ auf Seite 180

### Zugehörige Referenzen:

- „DROP statement“ in *SQL Reference, Volume 2*

## Löschen eines Schemas

### Vorgehensweise:

Bevor Sie ein Schema löschen, müssen alle Objekte, die sich in diesem Schema befinden, selbst gelöscht oder in ein anderes Schema versetzt werden. Der Schemaname muss im Katalog vorhanden sein, wenn die Anweisung DROP ausgeführt wird. Ansonsten wird ein Fehler zurückgegeben.

Gehen Sie wie folgt vor, um ein Schema über die Steuerzentrale 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 das Schema, das gelöscht werden soll, und wählen Sie im Kontextmenü **Löschen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie 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.

### Zugehörige Referenzen:

- „DROP statement“ in *SQL Reference, Volume 2*

## Ändern eines Pufferpools

Sie müssen möglicherweise eine der folgenden Tasks ausführen, wenn Sie mit einem bereits vorhandenen Pufferpool arbeiten:

- Ändern der Größe des Pufferpools in allen Partitionen oder in einer einzelnen Partition.
- Aktivieren oder Inaktivieren der Verwendung von erweitertem Speicher.
- Hinzufügen dieser Pufferpooldefinition zu einer neuen Datenbankpartitionsgruppe.
- Ändern des Blockbereichs des Pufferpools zur Unterstützung der blockbasierten Ein-/Ausgabe.

### Voraussetzungen:

Die Berechtigungs-ID der Anweisung muss über die Berechtigung SYSCTRL oder SYSADM verfügen.

### Vorgehensweise:

1. Verwenden Sie die Anweisung `SELECT BPNAME FROM SYSCAT.BUFFERPOOLS`, um die Liste der Pufferpoolnamen abzurufen, die in der Datenbank bereits vorhanden sind.
2. Wählen Sie den Pufferpoolnamen in der Ergebnisliste aus.
3. Legen Sie fest, welche Änderungen vorgenommen werden müssen.
4. Vergewissern Sie sich, dass die korrekte Berechtigungs-ID zum Ausführen der Anweisung `ALTER BUFFERPOOL` vorhanden ist.

**Anmerkung:** Bei den Parametern `IMMEDIATE` und `DEFERRED` handelt es sich um Schlüsselparameter. Wenn Sie den Parameter `IMMEDIATE` angeben, wird die Größe des Pufferpools sofort geändert. Wenn nicht ausreichend reservierter Speicherbereich im gemeinsam verwendeten Speicher der Datenbank vorhanden ist, um neuen Speicherbereich zuzuordnen, wird die Anweisung verzögert ausgeführt.

Wenn Sie die Option `DEFERRED` verwenden, wird der Pufferpool im Cache zwischengespeichert, wenn die Datenbank nach dem Trennen der Verbindung dieser Anwendungen erneut aktiviert wird. Reservierter Speicherbereich ist nicht erforderlich. DB2 UDB ordnet den erforderlichen Speicher bei der Aktivierung automatisch zu.

5. Mit der Anweisung `ALTER BUFFERPOOL` können Sie ein einzelnes Merkmal des Pufferpoolobjekts ändern.

### Zugehörige Tasks:

- „Erstellen eines Pufferpools“ auf Seite 72

### Zugehörige Referenzen:

- „ALTER BUFFERPOOL statement“ in *SQL Reference, Volume 2*





---

## Kapitel 6. Ändern von Tabellen und anderen zugehörigen Tabellenobjekten

Zum Modifizieren der Struktur und des Inhalts einer Tabelle und ihrer zugehörigen Tabellenobjekte sind unter anderem folgende Tasks erforderlich:

- „Speicherplatzkomprimierung für vorhandene Tabellen“
- „Hinzufügen von Spalten in einer vorhandenen Tabelle“ auf Seite 188
- „Modifizieren einer Spaltendefinition“ auf Seite 189
- „Entfernen von Zeilen aus einer Tabelle oder Sicht“ auf Seite 191
- „Aktualisieren von Tabellen- und Sichtinhalten mit der Anweisung MERGE“ auf Seite 210
- „Modifizieren einer Identitätsspaltendefinition“ auf Seite 193
- „Ändern einer Integritätsbedingung“ auf Seite 193
- „Definieren einer generierten Spalte in einer vorhandenen Tabelle“ auf Seite 200
- „Deklarieren einer flüchtigen Tabelle“ auf Seite 203
- „Ändern von Partitionierungsschlüsseln“ auf Seite 203
- „Ändern von Tabellenattributen“ auf Seite 204
- „Ändern der Merkmale einer gespeicherten Abfragetabelle“ auf Seite 207
- „Aktualisieren der Daten in einer gespeicherten Abfragetabelle“ auf Seite 208

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 214) und den neuen Auslöser hinzufügen (siehe „Erstellen eines Auslösers“ auf Seite 122).

---

### Modifizieren vorhandener Tabellen und der zugehörigen Tabellenobjekte

#### Speicherplatzkomprimierung für vorhandene Tabellen

Eine vorhandene Tabelle kann in ein Datensatzformat geändert werden, das eine Speicherplatzkomprimierung ermöglicht. Die Summe der Byteanzahlen der Spalten im Datensatzformat, das eine Speicherplatzkomprimierung ermöglicht, kann die Summe der Byteanzahlen der Spalten im ursprünglichen Datensatzformat (das keine Speicherplatzkomprimierung zulässt) überschreiten, sofern die Summe der Byteanzahlen nicht die zulässige Zeilenlänge der Tabelle im Tabellenbereich überschreitet. In einem Tabellenbereich mit einer Seitengröße von 4 KB beträgt die zulässige Zeilenlänge zum Beispiel 4005 Byte. Falls die zulässige Zeilenlänge überschritten wird, wird Fehlermeldung SQL0670N zurückgegeben. Die Formel für die Byteanzahl ist im Rahmen der Informationen über die Anweisung CREATE TABLE dokumentiert.

Umgekehrt kann eine vorhandene Tabelle auch aus einem Datensatzformat, das Speicherplatzkomprimierung ermöglicht, in ein Datensatzformat geändert werden, das dies nicht zulässt. Hinsichtlich der Summe der Byteanzahlen der Spalten gilt dieselbe Bedingung, ebenso wie auch die Fehlermeldung SQL0670N zurückgegeben wird, falls erforderlich.

Bei der Ermittlung, ob eine Speicherplatzkomprimierung für eine Tabelle in Betracht kommt, sollte berücksichtigt werden, dass eine Tabelle, deren Werte mehrheitlich mit den Systemstandardwerten übereinstimmen oder NULL sind, Vorteile durch das neue Zeilenformat gewinnen würde. Wenn zum Beispiel eine Tabelle eine INTEGER-Spalte enthält und 90 % der Spalte den Wert 0 (d. h. den Standardwert für den Datentyp INTEGER) oder NULL haben, würde eine Komprimierung dieser Tabelle plus dieser Spalte zu einem Vorteil aufgrund des neuen Zeilenformats führen und eine erhebliche Einsparung an Plattenspeicherplatz bedeuten.

Bei der Änderung einer Tabelle können Sie mit der Klausel `VALUE COMPRESSION` angeben, dass die Tabelle das Komprimierungszeilenformat auf Tabellenebene und möglicherweise auf Spaltenebene verwendet. Mit `ACTIVATE VALUE COMPRESSION` geben Sie an, dass die Tabelle die Speichereinspartechniken nutzen soll, während Sie mit `DEACTIVATE VALUE COMPRESSION` angeben, dass die Tabelle nicht länger mit den Speicherspartechniken für Daten in der Tabelle arbeiten soll.

Wenn Sie `DEACTIVATE VALUE COMPRESSION` verwenden, inaktiviert dies implizit alle Optionen `COMPRESS SYSTEM DEFAULT`, die Spalten in dieser Tabelle zugeordnet sind.

Nach der Modifizierung der Tabelle in ein neues Zeilenformat erhalten alle nachfolgenden Zeilen, die eingefügt, geladen oder aktualisiert werden, das neue Zeilenformat. Damit jede Zeile in das neue Zeilenformat geändert wird, sollten Sie eine Reorganisation der Tabelle oder eine Aktualisierungsoperation an vorhandenen Zeilen ausführen, bevor Sie das Zeilenformat ändern.

**Zugehörige Konzepte:**

- „Speicherplatzkomprimierung für neue Tabellen“ auf Seite 98

**Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

## Ändern einer Tabelle mit Hilfe einer gespeicherten Prozedur

Alle Daten für Ihr Unternehmen werden in Tabellen gespeichert. Bevor Sie Ihre Datenbank erstellen, müssen Sie sich über den Typ und die Organisation der Daten Gedanken machen, die Sie in der Datenbank verwalten wollen. Umfangreiche Planung ist erforderlich, um sicherzustellen, dass Sie die Verwendung und Bearbeitung aller relevanten Daten bedacht haben, die Ihr Unternehmen benötigen könnte. Dinge unterliegen jedoch ständigem Wandel. Trotz aller guten Planung können sich neue Anforderungen oder Unternehmensänderungen ergeben, die Änderungen an den Tabellen in der Datenbank erfordern.

Es ist möglich, dass Sie Änderungen einer oder mehrerer der folgenden Arten innerhalb Ihrer Tabellen vornehmen müssen:

- Umbenennen von Spalten
- Entfernen von Spalten
- Ändern eines Spaltentyps und Umsetzen vorhandener Daten mit Hilfe von SQL-Skalarfunktionen
- Vergrößern oder Verkleinern von Spalten
- Ändern des Standardwerts für Spalten

- Ändern einer Spalte von NOT NULL in NULLABLE
- Ändern der Genauigkeit und der Kommastellenanzahl für Dezimalwerte

Bei der Durchführung dieser Arten von Änderungen müssen Sie das Risiko eines Verlusts der ursprünglichen Tabellendaten minimieren. DB2® Universal Database (DB2 UDB) stellt eine Benutzerschnittstelle und eine gespeicherte Prozedur zur Verfügung, die Ihnen die Änderung einer Tabelle ermöglichen. Die ursprüngliche Tabelle und die zugehörigen Daten werden erst gelöscht, wenn Sie explizit angeben, dass alle gewünschten Änderungsarbeiten an der Tabelle abgeschlossen sind.

Jeder Aufruf der gespeicherten Prozedur, der über die Benutzerschnittstelle erfolgt, führt eine Reihe von Aktionen aus, wie zum Beispiel Löschen, erneutes Erstellen und Laden von Daten, um die oben aufgelisteten Änderungsarten zu realisieren.

Die Möglichkeiten zu Änderungen in der Tabellen sind jedoch eingeschränkt. Solche Einschränkungen sind:

- Änderungen an gespeicherten Abfragetabellen (MQTs) werden nicht unterstützt. Jedoch gibt es eine Unterstützung zur Änderung einer Tabelle, die eine gespeicherte Abfragetabelle hat. Außerdem werden gespeicherte Abfragetabellen, die für eine Basistabelle, die geändert wird, definiert sind, durch den ALTER TABLE-Prozess nicht aktualisiert (mit Daten gefüllt). In einer gespeicherten Abfragetabelle gehen bei der Änderung der zugehörigen Basistabelle durch die gespeicherte Prozedur ALTOBJ() alle Spalten verloren, die nicht Teil des SELECT-Ergebnisses aus der Basistabelle sind, weil der Inhalt der gespeicherten Abfragetabelle aus der neuen Basistabelle völlig neu gebildet wird.
- Änderungen an typisierten Tabellen bzw. an einer Tabelle, die der Bereich einer vorhandenen Tabelle mit einer Verweistypspalte ist, werden nicht unterstützt.
- Änderungen an einer fernen Tabelle über einen Kurznamen werden nicht unterstützt.
- Die Spaltenreihenfolge innerhalb der Tabelle kann nicht geändert werden.
- Die Spaltenaktionen ADD und RENAME schließen sich mit der Spaltenaktion DROP aus.  
Das heißt, diese Spaltenaktionen können nicht gleichzeitig in einem einzigen ALTER TABLE-Aufruf angegeben werden.
- Der Datentyp DATALINK wird nicht unterstützt.
- Die Definition der Objekte kann sich zwischen Aufrufen der Prozedur ALTOBJ() ändern, weil es keine Objektsperren gibt, die durchgehend gehalten werden.
- Tabellenprofile, wie zum Beispiel ein RUNSTATS-Profil, die dem Tabellenpaketdeskriptor zugeordnet sind, gehen nach dem Durchlauf des ALTER TABLE-Prozesses verloren.
- Nur eine Sequenz von ALTER TABLE-Aufrufen der gespeicherten Prozedur wird pro Tabelle zu einem gegebenen Zeitpunkt unterstützt. Das heißt, wenn die gespeicherte Prozedur ALTOBJ() einmal aufgerufen ist, sollte sie beendet oder rückgängig gemacht werden, bevor eine weitere Anweisung ALTER TABLE für die gleiche Tabelle gestartet werden kann. Die gleichzeitige Änderung mehrerer Tabellen mit Hilfe der gespeicherten Prozedur ALTOBJ() wird unterstützt, solange die Tabellenabhängigkeiten nicht in Konflikt geraten.

Es gibt verschiedene Bestandteile, aus denen sich die verfügbaren Optionen bei der Verwendung der gespeicherten Prozedur zusammensetzen und die die ALTER TABLE-Aktionen ausführen. Diese Teile sind:

- ALTER\_OBJ('GENERATE', '<sql-anweisung>', 0, ?)

Diese Prozedur generiert alle SQL-Anweisungen und speichert sie in einer Metadatatabelle.

**Anmerkung:** Im Modus GENERATE kann der Parameter für die SQL-Anweisung nicht null sein. Außerdem wird eine Änderungs-ID, falls angegeben, ignoriert.

- ALTER\_OBJ('VALIDATE',NULL,123,?)

Diese Prozedur prüft das generierte SQL, schließt jedoch kein Versetzen von Daten ein. Die Ausführung der Prozeduren zur Gültigkeitsprüfung erfolgt unter der angegebenen Benutzer-ID „123“. Die Ergebnisse der Überprüfung werden in die Metadatatabelle eingefügt (die auch die anderen Informationen aus der Tabelle enthält, die geändert wird).

- ALTER\_OBJ('APPLY\_CONTINUE\_ON\_ERROR',NULL,123,?)

Diese Prozedur führt alle SQL-Anweisungen unter der angegebenen ID aus und schreibt die Ergebnisse in die Metadatatabellen. Die SQL-Anweisungen umfassen Aktionen wie die Erstellung der neuen Tabelle, die Erstellung etwaiger abhängiger Objekte und das Füllen der neuen Tabelle mit Daten. Sie können die alten Definitionen mit Hilfe des Modus UNDO zurückholen (siehe unten).

Eine SQLCODE-Warnung wird für die gespeicherte Prozedur im SQL-Kommunikationsbereich (SQLCA) eingetragen, und die Transaktionen in der gespeicherten Prozedur werden beendet.

- ALTER\_OBJ('APPLY\_STOP\_ON\_ERROR',NULL,123,?)

Diese Prozedur führt jede der SQL-Anweisungen einzeln unter der angegebenen Benutzer-ID aus und stoppt, wenn Fehler auftreten.

Ein SQLCODE-Fehlerwert wird für die gespeicherte Prozedur im SQL-Kommunikationsbereich (SQLCA) eingetragen, und die Transaktionen werden automatisch rückgängig gemacht.

- ALTER\_OBJ('UNDO',NULL,123,?)

Führt die Prozedur aus, die alle Änderungen enthält, die durch die ALTER TABLE-Aktionen unter der angegebenen Benutzer-ID ausgeführt wurden. Alle diese Änderungen werden rückgängig gemacht.

**Anmerkung:** Bei der Verwendung von ALTOBJ\_UNDO kann der ID-Parameter nicht null sein.

- ALTER\_OBJ('FINISH',NULL,123,?)

Diese Prozedur löscht die ursprüngliche Tabelle und bereinigt alle Einträge in der Metadatatabelle unter der angegebenen Benutzer-ID.

**Anmerkung:** Dieser Modus kann nur getrennt von allen anderen Modi aufgerufen werden.

#### Zugehörige Referenzen:

- „Supported functions and SQL administrative routines“ in *SQL Reference, Volume 1*
- „ALTOBJ procedure“ in *SQL Administrative Routines*

## Hinzufügen von Spalten in einer vorhandenen Tabelle

#### Vorgehensweise:

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 (NULLABLE), definiert werden.

Gehen Sie wie folgt vor, um einer vorhandenen Tabelle Spalten über die Steuerzentrale hinzuzufügen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf die Tabelle, der Spalten hinzugefügt werden sollen, und wählen Sie im Kontextmenü **Ändern** aus.
3. Prüfen Sie die Seite **Spalten** und geben Sie die Informationen für die Spalte ein. Klicken Sie 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)
```

#### Zugehörige Tasks:

- „Modifizieren einer Spaltendefinition“ auf Seite 189

#### Zugehörige Referenzen:

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*

## Modifizieren einer Spaltendefinition

### Vorgehensweise:

Sie können die Merkmale einer Spalte durch Erhöhen des Werts für die Länge einer vorhandenen VARCHAR- oder VARCHAR2-Spalte ändern. Die Anzahl der Zeichen kann bis zu einem Maximalwert erhöht werden, der von der verwendeten Seitengröße abhängt.

Sie können den einer Spalte zugeordneten Standardwert modifizieren. Wenn Sie den neuen Standardwert definiert haben, wird der neue Wert bei allen nachfolgenden SQL-Operationen mit angegebener Verwendung des Standardwerts für die

Spalte verwendet. Der neue Wert muss den Regeln für die Zuordnung entsprechen und unterliegt ebenfalls den Einschränkungen, die unter der Anweisung CREATE TABLE dokumentiert sind.

**Anmerkung:** Der Standardwert generierter Spalten kann mit Hilfe dieser Anweisung nicht geändert werden.

Gehen Sie wie folgt vor, um die Länge einer Spalte einer vorhandenen Tabelle über die Steuerzentrale 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 die Tabelle, in der eine Spalte modifiziert werden soll, und wählen Sie im Kontextmenü **Ändern** aus.
3. Prüfen Sie die Seite **Spalten**, wählen Sie die gewünschte Spalte aus und klicken Sie auf **Ändern**.
4. Geben Sie unter **Länge** die neue Byteanzahl für die Spalte ein und klicken Sie auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um die Länge und den Typ einer Spalte einer vorhandenen Tabelle zu modifizieren:

```
ALTER TABLE <tabellenname>  
  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 t1  
  ALTER COLUMN spalte1  
  SET DATA TYPE VARCHAR(4000)
```

Das folgende Beispiel zeigt, wie Sie eine SQL-Anweisung wie die folgende verwenden, um einer Spalte einen neuen VARGRAPHIC-Wert zuzuordnen:

```
ALTER TABLE t1  
  ALTER COLUMN spalte2  
  SET DATA TYPE VARGRAPHIC(2000)
```

Sie können die Spalte einer typisierten Tabelle nicht ändern. Sie können einer vorhandenen Verweistypspalte ohne definierten Bereich jedoch einen Bereich hinzufügen. Beispiel:

```
ALTER TABLE t1  
  ALTER COLUMN spaltet1  
  ADD SCOPE typtab1
```

Geben Sie in der Befehlszeile Folgendes ein, um den Standardwert einer Spalte einer vorhandenen Tabelle zu modifizieren:

```
ALTER TABLE <tabellenname>  
  ALTER COLUMN <spaltenname>  
  SET DEFAULT 'neuer_standardwert'
```

Mit einer Anweisung wie der folgenden können Sie zum Beispiel den Standardwert für eine Spalte ändern:

```
ALTER TABLE t1  
  ALTER COLUMN spalte1  
  SET DEFAULT '123'
```

#### Zugehörige Tasks:

- „Modifizieren einer Identitätsspaltendefinition“ auf Seite 193

#### Zugehörige Referenzen:

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*

## Entfernen von Zeilen aus einer Tabelle oder Sicht

#### Vorgehensweise:

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 eine übergeordnete Tabelle (Elter) ist, dürfen die zum Löschen ausgewählten Zeilen keine abhängigen Elemente in einer Beziehung mit der Löschregel RESTRICT aufweisen. Ferner darf DELETE sich nicht auf untergeordnete Zeilen auswirken, die abhängige Elemente in einer Beziehung mit der Löschregel RESTRICT aufweisen.

Wenn die Löschoperation nicht von der Löschregel RESTRICT verhindert wird, werden die ausgewählten Zeilen gelöscht.

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, sowie alle durch vorhandene referenzielle Integritätsbedingungen erforderlichen Operationen verhindert, werden keine Änderungen an der Tabelle vorgenommen.

Wenn noch keine entsprechenden Sperrungen bestehen, werden während der erfolgreichen Ausführung der Anweisung DELETE eine oder mehrere exklusive Sperrungen eingerichtet. Sperrungen werden nach einer Anweisung COMMIT oder ROLLBACK wieder freigegeben. Sperrungen können verhindern, dass andere Anwendungen Operationen an der Tabelle ausführen.

#### **Zugehörige Konzepte:**

- „Sperrern und Steuerung des gemeinsamen Zugriffs“ in *Systemverwaltung: Optimierung*
- „Sperrern und Leistung“ in *Systemverwaltung: Optimierung*
- „Faktoren, die sich auf das Sperrern auswirken“ in *Systemverwaltung: Optimierung*
- „Richtlinien zum Sperrern“ in *Systemverwaltung: Optimierung*

#### **Zugehörige Referenzen:**

- „DELETE statement“ in *SQL Reference, Volume 2*

## **Modifizieren des Merkmals GENERATED oder IDENTITY einer Spalte**

Das Merkmal GENERATED oder IDENTITY einer Spalte in einer Tabelle kann mit Hilfe der Klausel ALTER COLUMN der Anweisung ALTER TABLE hinzugefügt oder gelöscht werden.

Sie können eine der folgenden Aktionen ausführen:

- Wenn Sie mit einer vorhandenen, nicht generierten Spalte (ohne das Merkmal GENERATED) arbeiten, können Sie ein Attribut mit einem GENERATED-Ausdruck hinzufügen. Die modifizierte Spalte wird dann zu einer generierten Spalte.
- Wenn Sie mit einer vorhandenen generierten Spalte (mit dem Merkmal GENERATED) arbeiten, können Sie das Attribut mit dem GENERATED-Ausdruck löschen. Die modifizierte Spalte wird dann zu einer normalen, nicht generierten Spalte.
- Wenn Sie mit einer vorhandenen Nicht-Identitätsspalte (ohne das Merkmal IDENTITY) arbeiten, können Sie ein IDENTITY-Attribut hinzufügen. Die modifizierte Spalte wird dann zu einer Identitätsspalte.
- Wenn Sie mit einer vorhandenen Identitätsspalte arbeiten, können Sie das IDENTITY-Attribut löschen. Die modifizierte Spalte wird dann zu einer normalen, nicht generierten Nicht-Identitätsspalte.
- Wenn Sie mit einer vorhandenen generierten Spalte arbeiten, können Sie die generierte Spalte vom Attribut GENERATED ALWAYS in das Attribut GENERATED BY DEFAULT ändern. Der umgekehrte Fall ist ebenfalls möglich. Das heißt, Sie können eine generierte Spalte vom Attribut GENERATED BY DEFAULT in das Attribut GENERATED ALWAYS ändern. Dies ist nur bei der Arbeit mit einer generierten Spalte möglich.
- Sie können das DEFAULT-Attribut aus einer benutzerdefinierten Standardspalte löschen. Wenn Sie dies tun, ist der neue Standardwert null.
- Sie können innerhalb derselben Anweisung ALTER COLUMN das vorhandene DEFAULT-, IDENTITY- oder GENERATED-Attribut löschen und anschließend ein neues DEFAULT-, IDENTITY- oder GENERATED-Attribut definieren.
- Für die Anweisungen CREATE TABLE und ALTER TABLE ist das Wort „ALWAYS“ in der GENERATED-Klausel optional. Das heißt, in der Anweisung ALTER TABLE ist der Ausdruck GENERATED ALWAYS mit dem Ausdruck GENERATED äquivalent.

#### **Zugehörige Tasks:**

- „Definieren einer generierten Spalte in einer neuen Tabelle“ auf Seite 107
- „Definieren einer Identitätsspalte in einer neuen Tabelle“ auf Seite 109



## Modifizieren einer Identitätsspaltendefinition

### Vorgehensweise:

Wenn Sie eine Tabelle mit einer Identitätsspalte (IDENTITY-Spalte) erneut erstellen und danach eine Import- oder Ladeoperation durchgeführt wird, wird die Identitätsspalte zurückgesetzt, um nach der Neuerstellung des Inhalts der Tabelle die Generierung des Identitätswerts bei 1 erneut zu beginnen. 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.

### Zugehörige Referenzen:

- „MAX aggregate function“ in *SQL Reference, Volume 1*
- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „LOAD Command“ in *Command Reference*

## 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“
- „Löschen einer eindeutigen Integritätsbedingung“ auf Seite 197

## Hinzufügen einer Integritätsbedingung

Integritätsbedingungen werden mit der Anweisung `ALTER TABLE` hinzugefügt. Weitere Informationen zu dieser Anweisung, einschließlich Syntax, finden Sie im Handbuch *SQL Reference*.

### Hinzufügen einer eindeutigen Integritätsbedingung

#### Vorgehensweise:

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)
```

#### Zugehörige Tasks:

- „Definieren einer eindeutigen Integritätsbedingung“ auf Seite 100
- „Löschen einer eindeutigen Integritätsbedingung“ auf Seite 197

#### Zugehörige Referenzen:

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*

## Hinzufügen von Primärschlüsseln

#### Vorgehensweise:

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 Primärschlüssel über die Steuerzentrale hinzuzufügen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf die Tabelle, die modifiziert werden soll, und wählen Sie im Kontextmenü **Ändern** aus.
3. Wählen Sie auf der Seite **Primärschlüssel** eine oder mehrere Spalten als Primärschlüssel aus und klicken Sie auf den Pfeil, um sie zu versetzen.
4. Optional: 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>
```

#### Zugehörige Tasks:

- „Hinzufügen von Fremdschlüsseln“ auf Seite 195

#### Zugehörige Referenzen:

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „SET INTEGRITY statement“ in *SQL Reference, Volume 2*

## Hinzufügen von Fremdschlüsseln

### Vorgehensweise:

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

Gehen Sie wie folgt vor, um Fremdschlüssel über die Steuerzentrale hinzuzufügen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf die Tabelle, die modifiziert werden soll, und wählen Sie 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 <aktionstyp>
  ON UPDATE <aktionstyp>
```

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
```

### Zugehörige Konzepte:

- „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 221

### Zugehörige Tasks:

- „Hinzufügen von Primärschlüsseln“ auf Seite 194

### Zugehörige Referenzen:

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*

## Hinzufügen einer Prüfung auf Integritätsbedingung in einer Tabelle

### Vorgehensweise:

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.

Gehen Sie wie folgt vor, um eine Prüfung auf Integritätsbedingungen in Tabellen über die Steuerzentrale hinzuzufügen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf die Tabelle, die modifiziert werden soll, und wählen Sie 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 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)
```

#### Zugehörige Konzepte:

- „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 221

#### Zugehörige Referenzen:

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „SET INTEGRITY statement“ in *SQL Reference, Volume 2*

## Löschen einer eindeutigen Integritätsbedingung

Integritätsbedingungen werden mit der Anweisung ALTER TABLE gelöscht. Weitere Informationen zu dieser Anweisung, einschließlich Syntax, finden Sie im Handbuch *SQL Reference*.

## Löschen einer eindeutigen Integritätsbedingung

#### Vorgehensweise:

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.

#### Zugehörige Referenzen:

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*

## Löschen von Primärschlüsseln

#### Vorgehensweise:

Gehen Sie wie folgt vor, um einen Primärschlüssel über die Steuerzentrale 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 die 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 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

**Zugehörige Konzepte:**

- „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 221

**Zugehörige Tasks:**

- „Löschen eines Fremdschlüssels“ auf Seite 198

**Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*

## Löschen eines Fremdschlüssels

**Vorgehensweise:**

Gehen Sie wie folgt vor, um einen Fremdschlüssel über die Steuerzentrale 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 die Tabelle, die modifiziert werden soll, und wählen Sie 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 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
```

**Zugehörige Konzepte:**

- „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 221

**Zugehörige Tasks:**

- „Löschen von Primärschlüsseln“ auf Seite 197

**Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*

## Löschen einer Prüfung auf Integritätsbedingung in einer Tabelle

### Vorgehensweise:

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. 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 eine Prüfung auf Integritätsbedingungen in einer Tabelle über die Steuerzentrale 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 die Tabelle, die modifiziert werden soll, und wählen Sie 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
```

### Zugehörige Konzepte:

- „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 221

### Zugehörige Tasks:

- „Hinzufügen einer Prüfung auf Integritätsbedingung in einer Tabelle“ auf Seite 196

### Zugehörige Referenzen:

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*

## Definieren einer generierten Spalte in einer vorhandenen Tabelle

Eine generierte Spalte wird in einer Basistabelle definiert, wobei der gespeicherte Wert mit Hilfe eines Ausdrucks berechnet und nicht durch eine Einfüge- oder Aktualisierungsoperation angegeben wird. Eine generierte Spalte kann erstellt werden, wenn eine Tabelle erstellt oder eine bereits vorhandene Tabelle modifiziert wird.

### Voraussetzungen:

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 VARCHARIC sowie benutzerdefinierte Typen, die unter Verwendung derselben ausgeschlossenen Datentypen definiert wurden.

Generierte Spalten können nicht in Integritätsbedingungen, eindeutigen Indizes, referenziellen Integritätsbedingungen, Primärschlüsseln und globalen temporären Tabellen verwendet werden. Eine mit LIKE erstellte Tabelle mit gespeicherten Sichten kann die Merkmale generierter Spalten nicht übernehmen.

### Einschränkungen:

Werte für 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 aufgezählt 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 Onlinemodus 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 Funktionen zum Lesen von SQL-Daten (READS SQL DATA) enthalten.
- Für generierte Spalten sind keine Prüfungen auf Integritätsbedingungen zulässig.



### Vorgehensweise:

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. Ordnen Sie den generierten Spalten die korrekten Werte zu. Dies kann durch eine der folgenden Methoden geschehen:

- Berechnen Sie die Werte für die generierten Spalten und ordnen Sie sie erneut zu, indem Sie folgenden Befehl ausführen:

```
SET INTEGRITY FOR t1 IMMEDIATE CHECKED FORCE GENERATED
```

Wenn die Ausführung dieser Anweisung SET INTEGRITY wegen Protokollspeichermangels fehlschlägt, vergrößern Sie den verfügbaren Speicherbereich für aktive Protokolle und setzen Sie die Anweisung SET INTEGRITY erneut ab.

#### Anmerkung:

An diesem Punkt können auch Ausnahmetabellen verwendet werden.

- Wenn es nicht möglich ist, den verfügbaren Speicherbereich für aktive Protokolle zu vergrößern, verwenden Sie gezielte UPDATE-Anweisungen, um den generierten Spalten ihre Standardwerte zuzuordnen.
  - a. Setzen Sie eine exklusiven Sperre für die Tabelle. Dadurch wird Transaktionen jeder Zugriff auf die Tabelle mit Ausnahme des nicht festgeschriebenen Lesezugriffs (Uncommitted Read) verwehrt. Beachten Sie, dass die Tabellensperre durch die erste sporadische Festschreibung freigegeben wird, so dass für andere Transaktionen Zeilen mit generierten Spalten sichtbar werden, denen ihre Standardwerte noch nicht zugeordnet wurden.

```
LOCK TABLE t1
```

- b. Übergehen Sie die Prüfung der generierten Spalten:

```
SET INTEGRITY FOR t1 GENERATED COLUMN IMMEDIATE UNCHECKED
```

- c. Prüfen Sie die Tabelle auf andere Integritätsverletzungen (falls zutreffend) und nehmen Sie sie aus dem Status 'Überprüfung anstehen':

```
SET INTEGRITY FOR t1 IMMEDIATE CHECKED
```

- d. Aktualisieren der generierten Spalten unter Verwendung sporadischer COMMIT-Operationen und Vergleichselementen, um ein Auffüllen der Protokolle zu vermeiden.

```
UPDATE t1 SET (c3, c4) = (DEFAULT, DEFAULT) WHERE <vergleichselement>
```

- e. Entsperrern Sie die Tabelle durch Abschließen der Transaktion mit einer COMMIT-Anweisung.

```
COMMIT
```

- Es kann auch ein cursorbasiertes Verfahren angewendet werden, wenn es nicht möglich ist, den verfügbaren Speicherbereich für aktive Protokolle zu vergrößern:
  - a. Deklarieren Sie einen FOR UPDATE-Cursor für die Tabelle. Wenn Sperren nach den sporadischen COMMIT-Operationen beibehalten werden sollen, sollte die Option WITH HOLD verwendet werden.

```
DECLARE C1 CURSOR WITH HOLD FOR S1
```

Datei ist S1 wie folgt definiert:

```
SELECT '0' FROM t1 FOR UPDATE OF C3, C4
```

b. Öffnen Sie den Cursor.

```
OPEN C1
```

c. Übergehen Sie die Prüfung der generierten Spalten:

```
SET INTEGRITY FOR t1 GENERATED COLUMN IMMEDIATE UNCHECKED
```

d. Prüfen Sie die Tabelle auf andere Integritätsverletzungen (falls zutreffend) und nehmen Sie sie aus dem Status 'Überprüfung anstehend':

```
SET INTEGRITY FOR t1 IMMEDIATE CHECKED
```

e. Rufen Sie alle Zeilen in der Tabelle durch eine Schleife ab und führen Sie für jede abgerufene Zeile Folgendes aus, um den generierten Spalten ihre Standardwerte zuzuordnen. Es ist wichtig, sicherzustellen, dass der erste Abruf sofort erfolgt, wenn die Tabelle aus dem Status 'Überprüfung anstehend' genommen wurde, um zu gewährleisten, dass die Tabelle für die Dauer des Cursors gesperrt wird.

```
UPDATE t1 SET (C3, C4) = (DEFAULT, DEFAULT) WHERE CURRENT OF C1
```

Führen Sie sporadische Festschreibungen (COMMIT) durch, um ein Auffüllen der Protokolle zu vermeiden.

f. Schließen Sie den Cursor und führen Sie eine Festschreibung durch, um die Tabelle zu entsperren:

```
CLOSE C1  
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. Dabei 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 GENERATION
```

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
```

#### Zugehörige Tasks:

- „Definieren einer generierten Spalte in einer neuen Tabelle“ auf Seite 107

#### Zugehörige Referenzen:

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „COMMIT statement“ in *SQL Reference, Volume 2*
- „LOCK TABLE statement“ in *SQL Reference, Volume 2*

- „SET INTEGRITY statement“ in *SQL Reference, Volume 2*
- „UPDATE statement“ in *SQL Reference, Volume 2*
- „db2look - DB2 Statistics and DDL Extraction Tool Command“ in *Command Reference*

## Deklarieren einer flüchtigen Tabelle

### Vorgehensweise:

Eine *flüchtige* Tabelle ist als Tabelle definiert, deren Inhalt bei der Verarbeitung zwischen leer und sehr umfangreich schwanken kann. Die Flüchtigkeit bzw. extreme Veränderlichkeit 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 über die 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 die Tabelle, die modifiziert werden soll, und wählen Sie 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 <tabellename>
VOLATILE CARDINALITY
```

### Zugehörige Referenzen:

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*

## Ändern von Partitionierungsschlüsseln

### Vorgehensweise:

Partitionierungsschlüssel können nur in Tabellen in Datenbankpartitionsgruppen mit einer Einzelpartition geändert werden. Löschen Sie zunächst 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
```

Der Partitionierungsschlüssel einer Tabelle in einer Datenbankpartitionsgruppe mit mehreren Partitionen kann nicht geändert werden. Wenn Sie versuchen, ihn zu löschen, wird ein Fehler zurückgegeben.

Gehen Sie zum Ändern des Partitionierungsschlüssels von Datenbankpartitionsgruppen mit mehreren Partitionen nach einer der folgenden Methoden vor:

- Exportieren Sie alle Daten in eine Datenbankpartitionsgruppe mit einer Einzelpartition und gehen Sie anschließend wie oben beschrieben vor.
- Exportieren Sie alle Daten, löschen Sie die Tabelle und erstellen Sie die Tabelle neu, indem Sie den Partitionierungsschlüssel neu definieren und anschließend alle Daten importieren.

Keine dieser Methoden ist für große Datenbanken praktikabel. Daher ist es wichtig, den geeigneten Partitionierungsschlüssel vor der Implementierung des Entwurfs großer Datenbanken zu definieren.

**Zugehörige Konzepte:**

- „Partitionierungsschlüssel“ in *Systemverwaltung: Konzept*

**Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*

## Ändern von Tabellenattributen

**Vorgehensweise:**

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 Clusterindizes. 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 die Clusterbildung 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 einer 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-Operationen, wobei gleichzeitig der Aufwand für die Verwaltung von Informationen über den freien Speicherbereich vermieden wird.

Eine Tabelle mit einem Clusterindex kann nicht zur Aktivierung des Anfügemodus (Append On) geändert werden. Entsprechend kann auch kein Clusterindex für eine Tabelle mit aktiviertem Anfügemodus erstellt werden.

**Zugehörige Konzepte:**

- „Sperrern und Steuerung des gemeinsamen Zugriffs“ in *Systemverwaltung: Optimierung*
- „Sperrattribute“ in *Systemverwaltung: Optimierung*
- „Sperrern und Leistung“ in *Systemverwaltung: Optimierung*
- „Faktoren, die sich auf das Sperrern auswirken“ in *Systemverwaltung: Optimierung*
- „Richtlinien zum Sperrern“ in *Systemverwaltung: Optimierung*

**Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*

## Ändern einer Identitätsspalte

**Vorgehensweise:**

Ändern Sie die Attribute einer vorhandenen Identitätsspalte mit der Anweisung ALTER TABLE.

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 gesetzt. Die numerische Konstante kann ein beliebiger positiver oder negativer Wert sein, der keine anderen Ziffern außer Nullen rechts von einem etwaigen Dezimalzeichen aufweisen darf und der einer Identitätsspalte zugeordnet werden könnte.

**Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*

## Ändern einer Sequenz

**Vorgehensweise:**

Ändern Sie die Attribute einer vorhandenen Sequenz mit der Anweisung ALTER SEQUENCE.

Die folgenden Attribute der Sequenz können geändert werden:

- 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 Anforderungsreihenfolge generiert werden müssen
- Neustarten der Sequenz

Es gibt zwei Aufgaben, die nicht Teil der Erstellung der Sequenz sind. Diese sind:

- **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 Konstanten zurück. Die numerische Konstante kann ein beliebiger positiver oder negativer Wert, der von einem etwaigen Dezimalzeichen keine anderen Ziffern als Nullen aufweisen darf.

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.

#### **Zugehörige Tasks:**

- „Löschen einer Sequenz“ auf Seite 206

#### **Zugehörige Referenzen:**

- „ALTER SEQUENCE statement“ in *SQL Reference, Volume 2*

## **Löschen einer Sequenz**

### **Vorgehensweise:**

Verwenden Sie zum Löschen einer Sequenz die Anweisung **DROP**.

Eine bestimmte Sequenz kann wie folgt gelöscht werden:

```
DROP SEQUENCE sequenzname
```

Dabei ist *sequenzname* der Name der zu löschenden Sequenz, der den impliziten bzw. expliziten Schemanamen enthält, um eine vorhandene Sequenz genau anzugeben.

Vom System erstellte Sequenzen für Identitätsspalten können nicht mit der Anweisung **DROP SEQUENCE** gelöscht werden.

Wenn eine Sequenz gelöscht wird, werden alle Zugriffsrechte für die Sequenz ebenfalls gelöscht.

#### Zugehörige Tasks:

- „Ändern einer Sequenz“ auf Seite 205

#### Zugehörige Referenzen:

- „DROP statement“ in *SQL Reference, Volume 2*

## Ändern der Merkmale einer gespeicherten Abfragetabelle

#### Vorgehensweise:

Mit einigen Einschränkungen können Sie eine gespeicherte Abfragetabelle in eine reguläre Tabelle bzw. eine reguläre Tabelle in eine gespeicherte Abfragetabelle ändern. Bei anderen Tabellentypen als regulären Tabellen und gespeicherten Abfragetabellen ist eine Änderung des Tabellentyps nicht möglich. Sie können z. B. keine replizierte gespeicherte Abfragetabelle in eine reguläre Tabelle oder umgekehrt ändern.

Nach dem Ändern einer regulären Tabelle in eine gespeicherte Abfragetabelle wird die Tabelle in den Status 'Überprüfung anstehend' versetzt. Bei einer derartigen Änderung muss die Gesamtauswahl in der Definition der gespeicherten Abfragetabelle 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 gespeicherte Abfragetabelle für eine ursprüngliche Tabelle definiert wird, kann diese ursprüngliche Tabelle nicht selbst in eine gespeicherte Abfragetabelle geändert werden. Wenn für die ursprüngliche Tabelle Auslöser, Prüfungen auf Integritätsbedingungen, referenzielle Integritätsbedingungen oder definierte eindeutige Indizes definiert sind, kann diese nicht in eine gespeicherte Abfragetabelle geändert werden. Wenn Sie die Tabellenmerkmale ändern, um eine gespeicherte Abfragetabelle 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 gespeicherte Abfragetabelle kann die Gesamtauswahl der Definition der gespeicherten Abfragetabelle nicht über Sichten, Aliasnamen oder andere gespeicherte Abfragetabellen direkt oder indirekt auf die ursprüngliche Tabelle verweisen.

Geben Sie Folgendes ein, um eine gespeicherte Abfragetabelle in eine reguläre Tabelle zu ändern:

```
ALTER TABLE sumtable
SET SUMMARY AS DEFINITION ONLY
```

Geben Sie Folgendes ein, um eine reguläre Tabelle in eine gespeicherte Abfragetabelle zu ändern:

```
ALTER TABLE regtable
SET SUMMARY AS <gesamtauswahl>
```

Die Einschränkungen, die beim Ändern einer regulären Tabelle in eine gespeicherte Abfragetabelle für die Gesamtauswahl gelten, stimmen größtenteils mit den Einschränkungen überein, die für das Erstellen einer Übersichtstabelle mit der Anweisung CREATE SUMMARY TABLE gelten.

**Zugehörige Tasks:**

- „Erstellen einer gespeicherten Abfragetabelle“ auf Seite 136
- „Aktualisieren der Daten in einer gespeicherten Abfragetabelle“ auf Seite 208
- „Löschen einer gespeicherten Abfragetabelle oder einer Zwischenspeichertabelle“ auf Seite 218

**Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

## Aktualisieren der Daten in einer gespeicherten Abfragetabelle

**Vorgehensweise:**

Sie können die Daten in einer oder mehreren gespeicherten Abfragetabellen 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 gespeicherten Abfragetabelle aktualisieren können:

```
REFRESH TABLE SUMTAB1
```

**Zugehörige Tasks:**

- „Erstellen einer gespeicherten Abfragetabelle“ auf Seite 136
- „Ändern der Merkmale einer gespeicherten Abfragetabelle“ auf Seite 207

**Zugehörige Referenzen:**

- „REFRESH TABLE statement“ in *SQL Reference, Volume 2*

---

## Ändern eines benutzerdefinierten strukturierten Typs

**Vorgehensweise:**

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).

**Zugehörige Konzepte:**

- „User-defined structured types“ in *Application Development Guide: Programming Server Applications*
- „Structured type hierarchies“ in *Application Development Guide: Programming Server Applications*

**Zugehörige Tasks:**

- „Creating structured types“ in *Application Development Guide: Programming Server Applications*

**Zugehörige Referenzen:**

- „ALTER TYPE (Structured) statement“ in *SQL Reference, Volume 2*



---

## Löschen und Aktualisieren von Zeilen einer typisierten Tabelle

Zeilen können mit Hilfe gezielter oder positionierter DELETE-Anweisungen aus typisierten Tabellen gelöscht werden. Zeilen können mit Hilfe gezielter oder positionierter UPDATE-Anweisungen in typisierten Tabellen aktualisiert werden.

### Zugehörige Konzepte:

- „Typed tables“ in *Application Development Guide: Programming Server Applications*

### Zugehörige Referenzen:

- „DELETE statement“ in *SQL Reference, Volume 2*
- „UPDATE statement“ in *SQL Reference, Volume 2*

---

## Umbenennen einer vorhandenen Tabelle oder eines Index

Sie können einer vorhandenen Tabelle oder einem Index einen neuen Namen innerhalb eines Schemas geben und die Berechtigungen und Indizes, die für die ursprüngliche Tabelle erstellt wurden, beibehalten.

### Voraussetzungen:

Die vorhandene, umzubenennende Tabelle bzw. der Index kann ein Aliasname sein, der eine Tabelle bzw. einen Index angibt.

### Einschränkungen:

Die vorhandene, umzubenennende Tabelle bzw. der Index darf nicht den Namen einer Katalogtabelle bzw. eines Katalogindex, einer Übersichtstabelle bzw. eines Übersichtsindex, einer typisierten Tabelle, einer deklarierten globalen temporären Tabelle, eines Kurznamens oder eines anderen Objekts als einer Tabelle, einer Sicht oder eines Aliasnamens haben.

Auf die vorhandene Tabelle bzw. den Index 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 bzw. der Index, die oder den Sie umbenennen, von keiner dieser Einschränkungen betroffen ist.

### Vorgehensweise:

Gehen Sie wie folgt vor, um eine vorhandene Tabelle oder einen Index über die Steuerzentrale umzubenennen:

1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner **Tabellen** oder **Sichten** angezeigt wird.
2. Klicken Sie mit der rechten Maustaste auf die Tabelle oder Sicht, die umbenannt werden soll, und wählen Sie im Kontextmenü **Umbenennen** aus.
3. Geben Sie den neuen Namen für die Tabelle oder Sicht ein und klicken Sie auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine vorhandene Tabelle umzubenennen:

```
RENAME TABLE <schemaname>.<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
```

Geben Sie in der Befehlszeile Folgendes ein, um einen vorhandenen Index umzubenennen:

```
RENAME INDEX <schemaname>.<indexname> TO <neuer_name>
```

Mit der folgenden SQL-Anweisung wird der Index EMPIND im Schema COMPANY in MSTRIND umbenannt:

```
RENAME INDEX COMPANY.EMPIND TO MSTRIND
```

Pakete werden ungültig gemacht und müssen erneut gebunden werden, wenn sie auf die Tabelle oder den Index verweisen, die bzw. der gerade umbenannt wurde. Die Pakete werden implizit erneut gebunden, unabhängig davon, ob ein anderer Index mit dem gleichen Namen vorhanden ist. Sofern keine bessere Wahl möglich ist, verwendet das Paket den gleichen Index wie zuvor, jedoch unter seinem neuen Namen.

### Zugehörige Referenzen:

- „RENAME statement“ in *SQL Reference, Volume 2*

---

## Aktualisieren von Tabellen- und Sichtinhalten mit der Anweisung MERGE

DB2 Universal Database bietet die Möglichkeit, eine Tabelle oder Sicht mit Daten aus einer anderen Quelle, in der Regel dem Ergebnis eines Tabellenverweises, zu aktualisieren. Diese Art der Aktualisierung wird mit Hilfe der Anweisung MERGE ausgeführt.

Zeilen in der Zieltabelle, die mit der Quelle übereinstimmen, können durch entsprechende Angaben in der Anweisung MERGE gelöscht oder aktualisiert werden. Zeilen, die in der Zieltabelle nicht vorhanden sind, können eingefügt werden.

Aktualisierungen, Löschungen oder Einfügungen von Zeilen in einer Sicht bewirken entsprechende Aktualisierungen, Löschungen oder Einfügungen in der Tabelle, auf der die Sicht basiert.

### Einschränkungen:

Die Berechtigungs-ID, die der Anweisung MERGE zugeordnet ist, muss über die erforderlichen Zugriffsrechte zur Ausführung der drei möglichen Aktionen verfügen: Aktualisieren (UPDATE), Löschen (DELETE) oder Einfügen (INSERT) für die Tabelle bzw. die zugrunde liegende Tabelle der Sicht. Die Berechtigungs-ID muss außerdem über die erforderlichen Zugriffsrechte für die Tabelle bzw. die zugrunde liegende Tabelle der Sicht in der Unterabfrage verfügen.

Wenn ein Fehler in der Anweisung MERGE auftritt, wird die gesamte Gruppe von Operationen, die der MERGE-Anweisung zugeordnet ist, rückgängig gemacht.

Es ist nicht möglich, eine Zeile in der Zieltabelle bzw. der zugrunde liegenden Tabelle der Sicht zu aktualisieren, die vor der Ausführung der Anweisung MERGE noch nicht vorhanden war. Das heißt, es ist nicht zulässig, eine Zeile zu aktualisieren, die im Rahmen der MERGE-Anweisung eingefügt wurde.

Wenn als Ziel der MERGE-Anweisung eine Sicht angegeben wird, sollten entweder keine INSTEAD OF-Auslöser für die Sicht definiert werden oder es sollte ein INSTEAD OF-Auslöser für jede einzelne UPDATE-, DELETE- und INSERT-Operation definiert werden.

### Vorgehensweise:

Geben Sie folgenden Befehl in eine Eingabeaufforderung ein, um eine UPDATE-, DELETE- oder INSERT-Aktion bzw. eine beliebige Kombination dieser Aktionen an der Zieltabelle auszuführen:

```
MERGE INTO <tabellen- oder sichtname>
  USING <tabellenverweis> ON <suchbedingung>
  WHEN <übereinstimmungsbedingung> THEN <änderungsoperation oder signal-anweisung>
```

Die Änderungsoperationen und SIGNAL-Anweisungen können in einer MERGE-Anweisung mehrmals angegeben werden. Jede Zeile in der Zieltabelle oder -sicht kann innerhalb einer MERGE-Anweisung nur einmal bearbeitet werden. Dies bedeutet, dass eine Zeile in der Zieltabelle oder -sicht nur mit einer Zeile in der Ergebnistabelle des Tabellenverweises als übereinstimmend (MATCHED) erkannt werden kann.

Betrachten Sie ein Beispiel mit zwei Tabellen: 'shipment' und 'inventory'. Mischen Sie mit Hilfe der Anweisung MERGE die Tabelle 'shipment' in die Tabelle 'inventory'. Erhöhen Sie für übereinstimmende Zeilen die Menge ('quantity') in der Tabelle 'inventory' um die Menge in der Tabelle 'shipment'. Fügen Sie anderenfalls die neue Artikelnummer ('partno') in die Tabelle 'inventory' ein.

```
MERGE INTO inventory AS in
  USING (SELECT partno, description, count FROM shipment
  WHERE shipment. partno IS NOT NULL) AS sh
  ON (in.partno = sh.partno)
  WHEN MATCHED THEN
    UPDATE SET
      description = sh.description
      quantity = in.quantity + sh.count
  WHEN NOT MATCHED THEN
    INSERT
      (partno, description, quantity)
    VALUES (sh.partno, sh.description, sh.count)
```

In diesem Beispiel wird die DELETE-Option nicht verwendet. Mit einer etwas komplexeren Übereinstimmungsbedingung könnte eine DELETE-Option eingefügt werden. Darüber hinaus sind weitere Optionen verfügbar, wie zum Beispiel die Verwendung der SIGNAL-Anweisung oder der ELSE-Klausel, die nicht an dieser Stelle, sondern im Handbuch 'SQL Reference' dokumentiert sind.

**Zugehörige Referenzen:**

- „MERGE-Anweisung“ in *SQL Reference, Volume 2*

---

## Löschen einer Tabelle

**Vorgehensweise:**

Eine Tabelle kann mit der SQL-Anweisung DROP TABLE gelöscht werden.

Beim Löschen einer Tabelle wird die Zeile im Katalog SYSCAT.TABLES gelöscht, die die Informationen über die Tabelle enthält, während alle anderen von der Tabelle abhängigen Objekte auf verschiedene Weise davon betroffen sind. Zum 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.
- 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.
- 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 eine Tabelle über die Steuerzentrale zu löschen:

- |  |
|--|
| <ol style="list-style-type: none"><li>1. Erweitern Sie die Sicht der Objektbaumstruktur so lange, bis der Ordner <b>Tabellen</b> angezeigt wird.</li><li>2. Klicken Sie mit der rechten Maustaste auf die Tabelle, die gelöscht werden soll, und wählen Sie im Kontextmenü <b>Löschen</b> aus.</li><li>3. Wählen Sie das Markierungsfeld <b>Bestätigung</b> aus und klicken Sie auf <b>OK</b>.</li></ol> |
|--|

Geben Sie in der Befehlszeile Folgendes ein, um eine Tabelle zu löschen:

```
DROP TABLE <tablenname>
```

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.

**Zugehörige Konzepte:**

- „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 221

**Zugehörige Tasks:**

- „Löschen einer benutzerdefinierten temporären Tabelle“ auf Seite 213
- „Wiederherstellen unbrauchbarer Sichten“ auf Seite 218

**Zugehörige Referenzen:**

- „DROP statement“ in *SQL Reference, Volume 2*

---

## Löschen einer benutzerdefinierten temporären Tabelle

Eine benutzerdefinierte temporäre Tabelle wird mit der Anweisung DECLARE GLOBAL TEMPORARY TABLE erstellt.

**Voraussetzungen:**

Beim Löschen derartiger Tabellen muss der Tabellename durch den Schemanamen SESSION qualifiziert werden und in der Anwendung definiert sein, welche die Tabelle erstellt hat.

**Einschränkungen:**

Pakete können von dieser Tabellenart nicht abhängig sein und werden daher beim Löschen einer solchen Tabelle nicht ungültig.

**Vorgehensweise:**

Wenn eine benutzerdefinierte temporäre Tabelle gelöscht wird und die Erstellung dieser Tabelle vor der aktiven Arbeitseinheit bzw. dem aktuellen Sicherungspunkt stattfand, wird die Tabelle funktional gelöscht und die Anwendung kann nicht auf sie zugreifen. Allerdings bleibt für die Tabelle weiterhin Speicherplatz im zugehöri-

gen Tabellenbereich reserviert. Dies verhindert, dass der temporäre Benutzer-tabellenbereich gelöscht wird, bevor die Arbeitseinheit festgeschrieben bzw. der Sicherungspunkt beendet ist.

**Zugehörige Tasks:**

- „Erstellen einer benutzerdefinierten temporären Tabelle“ auf Seite 108

**Zugehörige Referenzen:**

- „DROP statement“ in *SQL Reference, Volume 2*
- „SET SCHEMA statement“ in *SQL Reference, Volume 2*

---

## Löschen eines Auslösers

**Vorgehensweise:**

Ein Auslöserobjekt kann mit der Anweisung DROP gelöscht werden. Dieses Verfahren 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.

**Zugehörige Tasks:**

- „Erstellen eines Auslösers“ auf Seite 122

**Zugehörige Referenzen:**

- „DROP statement“ in *SQL Reference, Volume 2*

---

## Löschen einer benutzerdefinierten Funktion (UDF), Funktionszuordnung oder Methode

Eine benutzerdefinierte Funktion (UDF), Funktionsschablone oder Funktionszuordnung kann mit der Anweisung DROP gelöscht werden.

**Voraussetzungen:**

Andere Objekte können von einer Funktion oder Funktionsschablone abhängig sein. Alle solchen Abhängigkeiten (auch Funktionszuordnungen) müssen entfernt werden, bevor eine Funktion gelöscht werden kann. Eine Ausnahme bilden Pakete, die als unbrauchbar markiert werden.

### **Einschränkungen:**

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.

### **Vorgehensweise:**

Mit der Zuordnungsoption DISABLE können Sie eine Funktionszuordnung inaktivieren.

Pakete, die als unbrauchbar markiert werden, werden nicht implizit erneut gebunden. Das Paket muss entweder mit dem Befehl BIND bzw. REBIND erneut gebunden werden oder mit dem Befehl PREP vorbereitet werden. 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.

### **Zugehörige Referenzen:**

- „DROP statement“ in *SQL Reference, Volume 2*
- „BIND Command“ in *Command Reference*
- „PRECOMPILE Command“ in *Command Reference*
- „REBIND Command“ in *Command Reference*

---

## **Löschen eines benutzerdefinierten Datentyps (UDT) oder einer Typenzuordnung**

Ein benutzerdefinierter Datentyp (UDT) oder eine Typenzuordnung kann mit der Anweisung DROP gelöscht werden.

### **Einschränkungen:**

Ein UDT kann nicht gelöscht werden, wenn er wie folgt verwendet wird:

- In einer Spaltendefinition für eine vorhandene Tabelle oder Sicht (einzige 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 außer Kraft gesetzt werden.

Der Datenbankmanager versucht dann, alle Funktionen zu löschen, die von diesem einzigartigen Typ abhängig sind. 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.

Beachten Sie, dass nur Umsetzungen, die von Ihnen oder von anderen Anwendungsentwicklern definiert wurden, gelöscht werden können. Integrierte Umsetzungen und die dazugehörigen Gruppdefinitionen können nicht gelöscht werden.

**Vorgehensweise:**

Sie können Ihren benutzerdefinierten Datentyp mit der Anweisung DROP löschen.

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.

**Zugehörige Konzepte:**

- „Benutzerdefinierter Datentyp (UDT)“ auf Seite 129

**Zugehörige Tasks:**

- „Erstellen eines benutzerdefinierten einzigartigen Datentyps“ auf Seite 130
- „Erstellen einer Typenzuordnung“ auf Seite 132

**Zugehörige Referenzen:**

- „DROP statement“ in *SQL Reference, Volume 2*

---

## Ändern oder Löschen einer Sicht

Die Anweisung ALTER VIEW ändert eine vorhandene Sichtdefinition durch Ändern einer Verweistypspalte zum Hinzufügen eines Bereichs. Die Anweisung DROP löscht eine Sicht.

**Voraussetzungen:**

Beim Ändern der Sicht muss der Bereich (Scope) einer vorhandenen Verweistypspalte (REF) hinzugefügt werden, für die noch kein Bereich definiert wurde. Zudem darf die Spalte nicht von einer übergeordneten Sicht übernommen werden.

**Einschränkungen:**

Für Änderungen, die Sie an dem Inhalt vornehmen, der einer Sicht zugrunde liegt, benötigen Sie Auslöser. Andere Änderungen an einer Sicht können Sie nur durchführen, indem Sie die Sicht löschen und neu erstellen.

**Vorgehensweise:**

Der Datentyp des Spaltennamens in der Anweisung ALTER VIEW muss REF sein (Typ des Namens der typisierten Tabelle oder des Namens der typisierten Sicht). Sie können den Inhalt einer Sicht außerdem über INSTEAD OF-Auslöser modifizieren.



Andere Datenbankobjekte wie Tabellen und Indizes sind nicht betroffen, obwohl Pakete und im Cache zwischengespeicherte dynamische Anweisungen als ungültig markiert werden.

Gehen Sie wie folgt vor, um die Definition einer Sicht über die Steuerzentrale 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 die Sicht, die modifiziert werden soll, und wählen Sie 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 <name einer typisierten tabelle oder sicht>
```

Gehen Sie wie folgt vor, um eine Sicht über die Steuerzentrale 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 die Sicht, die gelöscht werden soll, und wählen Sie im Kontextmenü **Löschen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie 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.

Ebenso wie bei Tabellenhierarchien kann eine gesamte Sichtenhierarchie in einer einzigen Anweisung gelöscht werden, in der die Stammsicht der Hierarchie angegeben wird, wie im folgenden Beispiel gezeigt:

```
DROP VIEW HIERARCHY VPerson
```

#### **Zugehörige Konzepte:**

- „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 221

#### **Zugehörige Tasks:**

- „Erstellen eines Auslösers“ auf Seite 122
- „Erstellen einer Sicht“ auf Seite 133
- „Wiederherstellen unbrauchbarer Sichten“ auf Seite 218

#### **Zugehörige Referenzen:**

- „ALTER VIEW statement“ in *SQL Reference, Volume 2*
- „DROP statement“ in *SQL Reference, Volume 2*

---

## Wiederherstellen unbrauchbarer Sichten

### Vorgehensweise:

Sichten können *unbrauchbar* werden:

- Infolge eines widerrufenen Zugriffsrechts für eine zugrunde liegende Tabelle.
- Wenn eine Tabelle, ein Aliasname oder eine Funktion gelöscht wurde.
- Wenn die übergeordnete Sicht unbrauchbar wird. (Eine übergeordnete Sicht ist eine typisierte Sicht, auf der eine andere typisierte Sicht, d. h. eine untergeordnete Sicht, basiert.)
- Wenn die Sichten, von denen die Sichten abhängen, gelöscht werden.

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 waren, erneut zu erteilen. (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.

### Zugehörige Tasks:

- „Ändern oder Löschen einer Sicht“ auf Seite 216

### Zugehörige Referenzen:

- „CREATE VIEW statement“ in *SQL Reference, Volume 2*
- „DROP statement“ in *SQL Reference, Volume 2*
- „GRANT (Table, View, or Nickname Privileges) statement“ in *SQL Reference, Volume 2*
- „SYSCAT.VIEWS catalog view“ in *SQL Reference, Volume 1*

---

## Löschen einer gespeicherten Abfragetabelle oder einer Zwischenspeichertabelle

### Vorgehensweise:

Sie können eine gespeicherte Abfrage- oder Zwischenspeichertabelle nicht ändern, sondern nur löschen.

Alle Indizes, Primärschlüssel, Fremdschlüssel und Prüfungen auf Integritätsbedingungen, 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.

Gehen Sie wie folgt vor, um eine gespeicherte Abfragetabelle über die Steuerzentrale 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 die gespeicherte Abfrage- oder Zwischenspeichertabelle, die gelöscht werden soll, und wählen Sie im Kontextmenü **Löschen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie auf **OK**.

Geben Sie in der Befehlszeile Folgendes ein, um eine gespeicherte Abfrage- oder Zwischenspeichertabelle zu löschen:

```
DROP TABLE <tablename>
```

Mit der folgenden SQL-Anweisung wird die gespeicherte Abfragetabelle XT gelöscht:

```
DROP TABLE XT
```

Eine gespeicherte Abfragetabelle kann explizit mit der Anweisung DROP TABLE oder implizit durch das Löschen einer der zugrunde liegenden Tabellen gelöscht werden.

Eine Zwischenspeichertabelle kann explizit mit der Anweisung DROP TABLE oder implizit durch das Löschen ihrer zugeordneten gespeicherten Abfragetabelle gelöscht werden.

#### **Zugehörige Konzepte:**

- „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 221

#### **Zugehörige Tasks:**

- „Erstellen einer gespeicherten Abfragetabelle“ auf Seite 136
- „Erstellen einer Zwischenspeichertabelle“ auf Seite 141

#### **Zugehörige Referenzen:**

- „DROP statement“ in *SQL Reference, Volume 2*

---

## **Wiederherstellen unbrauchbarer Übersichtstabellen**

### **Vorgehensweise:**

Übersichtstabellen können infolge eines widerrufenen Zugriffsrechts SELECT in einer zugrunde liegenden 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 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.VIEWDEF, SYSCAT.TABAUTH, SYSCAT.COLUMNS und SYSCAT.COLAUTH werden entfernt.

**Zugehörige Referenzen:**

- „CREATE TABLE statement“ in *SQL Reference, Volume 2*
- „DROP statement“ in *SQL Reference, Volume 2*
- „GRANT (Table, View, or Nickname Privileges) statement“ in *SQL Reference, Volume 2*
- „SYSCAT.VIEWS catalog view“ in *SQL Reference, Volume 1*

## Löschen eines Index, einer Indexerweiterung oder einer Indexspezifikation

**Einschränkungen:**

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.)

Der Name der Indexerweiterung muss sich auf eine Indexerweiterung beziehen, die im Katalog beschrieben ist. Die Klausel RESTRICT erzwingt die Anwendung der Regel, dass kein Index definiert werden darf, der von der Indexerweiterungsdefinition abhängig ist. Wenn ein zugrundeliegender Index von dieser Indexerweiterung abhängig ist, schlägt die Löschoperation 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 Primär-

schlüssels oder des eindeutigen Schlüssels wird der Index nicht länger als Primärindex oder eindeutiger Index betrachtet und kann daher explizit gelöscht werden.

#### Vorgehensweise:

Gehen Sie wie folgt vor, um einen Index, eine Indexerweiterung oder eine Indexspezifikation über die Steuerzentrale 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 den Index, der gelöscht werden soll, und wählen Sie im Kontextmenü **Löschen** aus.
3. Wählen Sie das Markierungsfeld **Bestätigung** aus und klicken Sie 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
```

Pakete und zwischengespeicherte dynamische SQL-Anweisungen, die von den gelöschten Indizes abhängen, werden als ungültig markiert. Das Anwendungsprogramm ist von Änderungen durch das Hinzufügen oder Löschen von Indizes nicht betroffen.

#### Zugehörige Konzepte:

- „Anweisungsabhängigkeiten beim Ändern von Objekten“ auf Seite 221

#### Zugehörige Referenzen:

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „DROP statement“ in *SQL Reference, Volume 2*

---

## Anweisungsabhängigkeiten beim Ändern von Objekten

Anweisungsabhängigkeiten gelten für 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.

Pakete und zwischengespeicherte dynamische SQL-Anweisungen können von vielen Arten von Objekten abhängig sein.

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 Bei-

spiel, 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 sind.

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“ (engl. invalid) versetzt. Wenn ein Paket von einer benutzerdefinierten Funktion abhängig ist und diese Funktion gelöscht wird, wird das Paket in den Status „unbrauchbar“ (engl. inoperative) 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.

Für Objekte zusammenschlossener 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

#### **Zugehörige Konzepte:**

- „Package Creation Using the BIND Command“ in *Application Development Guide: Programming Client Applications*
- „Application, Bind File, and Package Relationships“ in *Application Development Guide: Programming Client Applications*
- „Package Rebinding“ in *Application Development Guide: Programming Client Applications*

**Zugehörige Referenzen:**

- „DROP statement“ in *SQL Reference, Volume 2*
- „SYSCAT.PACKAGEAUTH catalog view“ in *SQL Reference, Volume 1*
- „SYSCAT.PACKAGEDEP catalog view“ in *SQL Reference, Volume 1*
- „SYSCAT.PACKAGES catalog view“ in *SQL Reference, Volume 1*
- „BIND Command“ in *Command Reference*
- „REBIND Command“ in *Command Reference*





---

## Teil 2. Datenbanksicherheit



---

## Kapitel 7. 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 8, „Prüfen von Aktivitäten in DB2 Universal Database™ (DB2 UDB)“, auf Seite 283 behandelt.

Die folgenden Themen werden behandelt:

- „Sicherheitsaspekte bei der Installation von DB2 Universal Database“
- „Authentifizierungsmethoden für den Server“ auf Seite 234
- „Authentifizierungsaspekte für ferne Clients“ auf Seite 240
- „Authentifizierungsaspekte bei partitionierten Datenbanken“ auf Seite 240
- „Einführung in die Firewallunterstützung“ auf Seite 280
- „Zugriffsrechte, Berechtigungsstufen und Datenbankberechtigungen“ auf Seite 245
- „Steuern des Zugriffs auf Datenbankobjekte“ auf Seite 263
- „Aufgaben und erforderliche Berechtigungen“ auf Seite 274
- „Verwenden des Systemkatalogs zu Sicherheitszwecken“ auf Seite 276.

**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.

---

### | Sicherheitsaspekte bei der Installation von DB2 Universal Database

Sicherheitsaspekte spielen für den DB2®-Administrator von dem Moment der Installation des Produkts an eine wichtige Rolle.

| Zur Ausführung der Installation von DB2 Universal Database™ (DB2 UDB) sind  
| eine Benutzer-ID, ein Gruppenname und ein Kennwort erforderlich. Das grafisch  
| orientierte DB2 UDB-Installationsprogramm erstellt Standardwerte für verschie-  
| dene Benutzer-IDs und die Gruppe. Abhängig davon, ob Sie auf UNIX®- oder Win-  
| dows®-Plattformen installieren, werden unterschiedliche Standardwerte erstellt:

- Auf UNIX-Plattformen erstellt das DB2 UDB-Installationsprogramm verschie-  
| dene Standardbenutzer für den DAS (dasusr), den Exemplareigner (db2inst)  
| und den abgeschirmten Benutzer (db2fenc).

| Das DB2 UDB-Installationsprogramm hängt eine Nummer von 1 bis 99 an den  
| Standardbenutzernamen an, bis eine Benutzer-ID erstellt wird, die nicht bereits

vorhanden ist. Wenn zum Beispiel die Benutzer db2inst1 und db2inst2 bereits vorhanden sind, erstellt das DB2 UDB-Installationsprogramm den Benutzernamen db2inst3. Wenn eine Nummer über 10 verwendet wird, wird der Buchstabenteil des Namens in der Standardbenutzer-ID abgeschnitten. Wenn zum Beispiel die Benutzer-ID db2fenc9 bereits vorhanden ist, schneidet das DB2 UDB-Installationsprogramm das c der Benutzer-ID ab und hängt dann den Wert 10 an (also db2fen10). Dieses Abschneiden findet nicht statt, wenn der numerische Wert an die Standard-DAS-Benutzer-ID angehängt wird (z. B. dasusr24).

- Auf Windows-Plattformen erstellt das DB2 UDB-Installationsprogramm die Standardbenutzer-ID db2admin für den DAS-Benutzer, den Exemplareigner und abgeschirmte Benutzer. Im Unterschied zu UNIX-Plattformen wird kein numerischer Wert an die Benutzer-ID angehängt.

Zur Minimierung des Risikos, dass ein anderer Benutzer als der Administrator von den Standardwerten erfährt und sie in unlauterer Weise in Datenbanken und Exemplaren verwendet, sollten Sie die Standardwerte bei der Installation in eine neue oder vorhandene Benutzer-ID Ihrer Wahl ändern.

**Anmerkung:** Installationen mit Antwortdateien arbeiten nicht mit Standardwerten für Benutzer-IDs oder Gruppennamen. Diese Werte müssen in der Antwortdatei angegeben werden.

Kennwörter spielen bei der Authentifizierung von Benutzern eine wichtige Rolle. 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 Berechtigung zur Verbindung erteilt. Unter UNIX-Betriebssystemen werden nicht definierte Kennwörter beispielsweise wie NULL behandelt. In diesem Fall wird jeder Benutzer ohne definiertes Kennwort wie ein Benutzer mit dem Kennwort NULL behandelt. Aus der Perspektive des Betriebssystems ist dies eine Übereinstimmung, und der Benutzer erhält eine gültige Berechtigung und kann eine Verbindung zur Datenbank herstellen. Verwenden Sie Kennwörter auf Betriebssystemebene, wenn das Betriebssystem für die Authentifizierung von Benutzern Ihrer Datenbank zuständig sein soll.

**Anmerkung:** Nicht definierte Kennwörter sind nicht zulässig, wenn Ihre Datenbankumgebung den Anforderungen der Common Criteria-Zertifizierung entsprechen soll.

Nach der Installation von DB2 Universal Database sollten Sie darüber hinaus die Standardzugriffsrechte, die Benutzern erteilt wurden, prüfen und (falls erforderlich) ändern. Standardmäßig erteilt der Installationsprozess den folgenden Benutzern unter den einzelnen Betriebssystemen die Zugriffsrechte eines Systemadministrators (SYSADM):

**Windows 9x**

Jedem Windows 98- oder Windows ME-Benutzer

**Andere Windows-Umgebungen**

Unter Windows NT<sup>®</sup>, Windows 2000, Windows XP oder Windows Server 2003 einem gültigen DB2 UDB-Benutzernamen, der zur Administratorengruppe gehört.

**UNIX-Plattformen**

Einem gültigen DB2 UDB-Benutzernamen, der zur Primärgruppe des Exemplareigners gehört.

SYSADM-Zugriffsrechte bilden die mächtigste Gruppe von Zugriffsrechten, die innerhalb von DB2 Universal Database zur Verfügung steht. Es ist daher vielleicht

nicht sinnvoll, dass alle diese Benutzer über SYSADM-Zugriffsrechte verfügen. DB2 UDB stattet den Administrator mit der Möglichkeit aus, Gruppen und einzelnen Benutzer-IDs Zugriffsrechte zu erteilen und zu entziehen.

Durch Aktualisieren des Konfigurationsparameters *sysadm\_group* des Datenbankmanagers kann der Administrator steuern, welche Gruppe von Benutzern SYSADM-Zugriffsrechte besitzt. Sie müssen die im Folgenden dargestellten Hinweise befolgen, um die Sicherheitsanforderungen für die DB2 UDB-Installation sowie die nachfolgende Erstellung von Exemplaren und Datenbanken zu erfüllen.

Jede Gruppe, die (durch Aktualisieren des Parameters *sysadm\_group*) als Systemadministratorgruppe definiert ist, muss vorhanden sein. Der Name sollte diese Gruppe leicht als die Gruppe zu erkennen geben, die für Exemplareigner erstellt wurde. Die Benutzer-IDs und Gruppen, die dieser Gruppe angehören, verfügen über die Systemadministratorberechtigung für ihre jeweiligen Exemplare.

Der Administrator sollte in Betracht ziehen, eine Benutzer-ID für den Exemplareigner zu erstellen, die sich leicht als zu einem bestimmten Exemplar gehörig erkennen lässt. Diese Benutzer-ID sollte als eine ihrer Gruppen den Namen der oben erstellten SYSADM-Gruppe enthalten. Eine weitere Empfehlung ist, diese Benutzer-ID des Exemplareigners nur als Mitglied der Exemplareignergruppe und in keiner anderen Gruppe zu verwenden. Dadurch wird die Verbreitung von Benutzer-IDs und Gruppen begrenzt, die das Exemplar oder ein Objekt innerhalb des Exemplars modifizieren können.

Der erstellten Benutzer-ID muss 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 jeweiligen Unternehmens einzuhalten.

#### **Zugehörige Konzepte:**

- „Namenskonventionen in einer NLS-Umgebung“ auf Seite 333
- „Namenskonventionen in einer Unicode-Umgebung“ auf Seite 334
- „Windows NT-Plattform: Sicherheitsaspekte für Benutzer“ auf Seite 232
- „UNIX-Plattform: Sicherheitsaspekte für Benutzer“ auf Seite 233
- „Authentifizierung“ in *Systemverwaltung: Konzept*
- „Berechtigung“ in *Systemverwaltung: Konzept*
- „Position des Exemplarverzeichnisses“ auf Seite 233
- „Allgemeine Namenskonventionen“ auf Seite 327
- „Namenskonventionen für Benutzer, Benutzer-IDs und Gruppen“ auf Seite 330

---

## **Abrufen von Windows-Benutzergruppeninformationen mit einem Zugriffstoken**

Ein Zugriffstoken ist ein Objekt, das den Sicherheitskontext eines Prozesses oder Threads beschreibt. Die Informationen in einem Zugriffstoken enthalten die Identität und die Zugriffsrechte des Benutzerkontos, das dem Prozess oder Thread zugeordnet ist.

Wenn Sie sich anmelden, überprüft das System Ihr Kennwort, indem es dieses mit den in einer Sicherheitsdatenbank gespeicherten Informationen vergleicht. Wenn

das Kennwort authentifiziert wird, erstellt das System ein Zugriffstoken. Jeder Prozess, der in Ihrem Namen ausgeführt wird, verwendet eine Kopie dieses Zugriffstokens.

Ein Zugriffstoken kann außerdem auf der Grundlage eines im Cache gespeicherten Berechtigungsnachweises empfangen werden. Wenn Sie einmal am System authentifiziert sind, wird Ihr Berechtigungsnachweis vom Betriebssystem in den Cache gestellt. Auf das Zugriffstoken der letzten Anmeldung kann im Cache zurückgegriffen werden, wenn es nicht möglich ist, den Domänencontroller zu kontaktieren.

Das Zugriffstoken enthält Informationen über alle Gruppen, denen Sie angehören: lokale Gruppen und verschiedene Domänengruppen (globale Gruppen, lokale Domänengruppen, universelle Gruppen).

**Anmerkung:** Eine Gruppensuche (Lookup) unter Verwendung der Clientauthentifizierung wird über eine Remoteverbindung nicht unterstützt, selbst wenn die Unterstützung für Zugriffstoken aktiviert ist.

Zur Aktivierung der Unterstützung für Zugriffstoken müssen Sie die Registriervariable `DB2_GRP_LOOKUP` mit Hilfe des Befehls `db2set` aktualisieren. Folgende Auswahlmöglichkeiten stehen zur Aktualisierung dieser Registriervariablen zur Verfügung:

- `TOKEN`

Diese Auswahl aktiviert die Unterstützung für Zugriffstoken zur Suche nach allen Gruppen, zu denen der Benutzer gehört, an dem Standort, an dem das Benutzerkonto definiert ist. Dieser Standort ist in der Regel entweder die Domäne oder das lokale System des DB2 Universal Database™-Servers (DB2 UDB-Servers).
- `TOKENLOCAL`

Diese Auswahl aktiviert die Unterstützung für Zugriffstoken zur Suche nach allen lokalen Gruppen, zu denen der Benutzer gehört, auf dem DB2 UDB-Server.
- `TOKENDOMAIN`

Diese Auswahl aktiviert die Unterstützung für Zugriffstoken zur Suche nach allen Domänengruppen, zu denen der Benutzer gehört, in der Domäne.

Wenn Sie die Unterstützung für Zugriffstoken aktivieren, sind einige Einschränkungen zu beachten, die sich auf die Infrastruktur Ihrer Kontenverwaltung beziehen. Wenn diese Unterstützung aktiviert ist, sammelt DB2 UDB Gruppeninformationen über den Benutzer, der die Verbindung zur Datenbank herstellt. Nachfolgende Operationen nach einer erfolgreichen `CONNECT`- oder `ATTACH`-Anforderung, die Abhängigkeiten von anderen Berechtigungs-IDs haben, verwenden weiterhin die konventionelle Gruppenaufzählung. Die Vorteile von Zugriffstoken verschachtelter globaler Gruppen, lokaler Domänengruppen und im Cache gespeicherter Berechtigungsnachweise stehen hier nicht zur Verfügung. Wenn zum Beispiel nach Herstellung einer Verbindung der Befehl `SET SESSION_USER` verwendet wird, um zu einer anderen Berechtigungs-ID zu wechseln, wird nur die konventionelle Gruppenaufzählung verwendet, um die Rechte zu überprüfen, die der neuen Berechtigungs-ID für die Sitzung erteilt werden. Sie müssen weiterhin einzelnen, in DB2 UDB bekannten Berechtigungs-IDs explizite Zugriffsrechte erteilen oder entziehen, anstatt Gruppen, zu denen die Berechtigungs-IDs gehören, Zugriffsrechte erteilen und entziehen zu können.

Wenn Sie beabsichtigen, den Berechtigungen `SYSADM`, `SYSMAINT` oder `SYSCTRL` Gruppen zuzuordnen, müssen Sie sicherstellen, dass die zugeordneten Gruppen

weder verschachtelte globale Gruppen noch lokale Domänengruppen sind. Anschließend wird die Funktion zur Speicherung von Berechtigungs nachweisen im Cache nicht benötigt.

Sie sollten in Betracht ziehen, die Registriervariable DB2\_GRP\_LOOKUP zu verwenden und den Standort für die Gruppensuche anzugeben, um DB2 UDB mitzuteilen, wo Gruppen mit Hilfe der konventionellen Gruppensuchmethode zu suchen sind. Beispiel:

```
db2set DB2_GRP_LOOKUP=LOCAL,TOKENLOCAL
```

Dieser Befehl aktiviert die Unterstützung für Zugriffstoken zur Aufzählung lokaler Gruppen. Die Gruppensuche nach einer anderen Berechtigungs-ID als der des verbundenen Benutzers erfolgt auf dem DB2 UDB-Server.

```
db2set DB2_GRP_LOOKUP=,TOKEN
```

Dieser Befehl aktiviert die Unterstützung für Zugriffstoken zur Aufzählung von Gruppen an dem Standort, an dem die Benutzer-ID definiert ist. Die Gruppensuche nach einer anderen Berechtigungs-ID als der des verbundenen Benutzers erfolgt dort, wo die Benutzer-ID definiert ist.

```
db2set DB2_GRP_LOOKUP=DOMAIN,TOKENDOMAIN
```

Dieser Befehl aktiviert die Unterstützung für Zugriffstoken zur Aufzählung von Domänengruppen. Die Gruppensuche nach einer anderen Berechtigungs-ID als der des verbundenen Benutzers erfolgt dort, wo die Benutzer-ID definiert ist.

Anwendungen, die mit dynamischem SQL in einem Paket arbeiten, das mit der Option DYNAMICRULES RUN (der Standardoption) gebunden wurde, werden unter den Zugriffsrechten der ausführenden Person ausgeführt. In diesem Fall gelten die zuvor erwähnten Einschränkungen nicht. Hierzu gehören auch Anwendungen, die zur Verwendung von Java Database Connectivity (JDBC) und DB2 Call Level Interface (CLI) geschrieben wurden.

Die Unterstützung für Zugriffstoken kann bei allen Authentifizierungstypen außer der Clientauthentifizierung (CLIENT) aktiviert werden.

**Anmerkung:** Für Windows® NT 4.0-Benutzer unterstützt die Unterstützung für Zugriffstoken nur den Sicherheitskontext auf Prozessebene, wenn Ihre DB2 UDB-Anwendungen mit impliziten lokalen Verbindungen arbeiten. Das heißt, alle Threads in den Anwendungen werden so behandelt, als würden sie unter dem Sicherheitskontext des Benutzers ausgeführt, der die Anwendungen ausführt. Wenn Sie verschiedene Benutzersicherheitskontexte für unterschiedliche Threads benötigen, sollten Sie eine Migration auf Windows 2000 oder eine spätere Version in Betracht ziehen. Oder ändern Sie Ihre DB2 UDB-Anwendungen so, dass sie explizite Verbindungen verwenden.

#### Zugehörige Konzepte:

- „Sicherheitsaspekte bei der Installation von DB2 Universal Database“ auf Seite 227

---

## Details zur Sicherheit auf der Basis des Betriebssystems

Jedes Betriebssystem stellt Möglichkeiten zur Verwaltung der Sicherheit bereit. Dieser Abschnitt behandelt einige der mit den Betriebssystemen verbundenen Sicherheitsaspekte.

**Anmerkung:** Die kryptografischen Routinen, die von DB2 Universal Database™ (DB2 UDB) zur Verschlüsselung von Benutzer-IDs und Kennwörtern beim Authentifizierungstyp SERVER\_ENCRYPT sowie von Benutzer-IDs, Kennwörtern und Benutzerdaten beim Authentifizierungstyp DATA\_ENCRYPT verwendet werden, sind mit der Spezifikation FIPS 140-2 konform.

Die kryptografischen Routinen werden von IBM Crypto for C (ICC) Version 1.2.1 zur Verfügung gestellt. Das FIPS 140-2 Validation Certificate No. 384 for ICC finden Sie auf der NIST-Website unter folgender Adresse: <http://csrc.nist.gov/cryptval/140-1/140crt/140crt384.pdf>

Die Sicherheitsrichtlinie (Security Policy) finden Sie ebenfalls auf der NIST-Website unter:  
<http://csrc.nist.gov/cryptval/140-1/140sp/140sp384.pdf>

Die Sicherheitsrichtlinie (Security Policy) hat eine Anleitung, die befolgt werden muss, um DB2 UDB in einer FIPS 140-2-konformen Weise zu installieren. Abschnitt 5.3.2 enthält weitere Informationen, die Sie beachten sollten.

Die Einhaltung der FIPS 140-2-Spezifikation ist nur für die folgenden Systeme möglich: AIX, Microsoft Windows, Solaris Operating Environment, Linux und HP-UX. Dem Prüfungszertifikat (Validation Certificate) und der Sicherheitsrichtlinie (Security Policy) sind die vollständigen Details zu unterstützten Systemen zu entnehmen.

## Windows NT-Plattform: Sicherheitsaspekte für Benutzer

Die Systemadministratorberechtigung (SYSADM) wird jedem gültigen DB2® Universal Database-Benutzerkonto (DB2 UDB-Benutzerkonto) erteilt, das zur lokalen Administratorgruppe auf der Maschine gehört, auf der das Konto definiert ist.

Standardmäßig verfügen in einer Windows®-Domänenumgebung nur die Domänenbenutzer, die zur Administratorgruppe des Domänencontrollers gehören, über die Berechtigung SYSADM für ein Exemplar. Da DB2 UDB die Vergabe von Berechtigungen immer auf der Maschine ausführt, auf der ein Konto definiert wurde, wird einem Domänenbenutzer durch das Hinzufügen zur lokalen Administratorgruppe auf dem Server nicht die Berechtigung SYSADM für die Gruppe erteilt.

**Anmerkung:** In einer solchen Domänenumgebung, wie sie in Windows NT® anzutreffen ist, authentifiziert DB2 UDB nur die ersten 64 Gruppen, welche die Anforderungen und Einschränkungen erfüllen und zu denen eine Benutzer-ID gehört. Sie können mehr als 64 Gruppen haben.

Um das Hinzufügen eines Domänenbenutzers zur Administratorgruppe auf dem primären Domänencontroller (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
```



#### Zugehörige Konzepte:

- „UNIX-Plattform: Sicherheitsaspekte für Benutzer“ auf Seite 233

## Unterstützung für das lokale Systemkonto unter Windows

Auf Windows®-Plattformen unterstützt DB2® Universal Database (DB2 UDB) Anwendungen, die unter dem Kontext des lokalen Systemkontos (LSA) mit impliziter lokaler Verbindung ausgeführt werden. Entwickler, die Anwendungen zur Ausführung unter diesem Konto schreiben, müssen sich über die Einschränkungen in DB2 UDB für Objekte mit Schemanamen, die mit „SYS“ beginnen, im Klaren sein. Wenn Ihre Anwendungen DDL-Anweisungen enthalten, durch die DB2 UDB-Objekte erstellt werden, sollten diese daher so geschrieben sein, dass sie folgende Merkmale erfüllen:

- Für statisches SQL sollten sie mit einem anderen Wert für die QUALIFIER-Optionen als dem Standardwert gebunden werden.
- Für dynamisches SQL sollten die zu erstellenden Objekte explizit mit einem von DB2 UDB unterstützten Schemanamen qualifiziert werden, oder das Register CURRENT SCHEMA muss auf ein Schema gesetzt werden, das von DB2 UDB unterstützt wird.

Gruppeninformationen für das lokale Systemkonto (LSA) werden beim ersten Suchen nach Gruppen (Lookup) nach dem Start des DB2 UDB-Exemplars erfasst und erst aktualisiert, wenn das Exemplar erneut gestartet wird.

#### Zugehörige Konzepte:

- „Sicherheitsaspekte bei der Installation von DB2 Universal Database“ auf Seite 227

## UNIX-Plattform: Sicherheitsaspekte für Benutzer

DB2® Universal Database (DB2 UDB) unterstützt keine direkte Funktion der Rootberechtigung als Datenbankadministrator. Sie sollten den Befehl **su - <exemplareigner>** als Datenbankadministrator verwenden.

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.

#### Zugehörige Konzepte:

- „Windows NT-Plattform: Sicherheitsaspekte für Benutzer“ auf Seite 232

## Position des Exemplarverzeichnisses

Unter UNIX® erstellt der Befehl **db2icrt** das Hauptverzeichnis für die SQL-Bibliothek (`sqllib`) unter dem Ausgangsverzeichnis des Exemplareigners.

Unter Windows®-Betriebssystemen befindet sich das Exemplarverzeichnis im Unterverzeichnis /sql1lib des Verzeichnisses, in dem DB2® installiert wurde.

**Zugehörige Konzepte:**

- „Exemplarerstellung“ auf Seite 18

**Zugehörige Tasks:**

- „Erstellen zusätzlicher Exemplare“ auf Seite 23

## Sicherheits-Plug-ins

Die Authentifizierung in DB2® Universal Database (DB2 UDB) wird mit Hilfe von *Sicherheits-Plug-ins* ausgeführt. Weitere Informationen finden Sie unter "Sicherheits-Plug-ins" in *Application Development Guide: Programming Client Applications*.

---

## Authentifizierungsmethoden für den Server

Der Zugriff auf ein Exemplar oder eine Datenbank erfordert zunächst, dass der Benutzer *authentifiziert*, d. h. auf seine Identität hin überprüft, wird. Der *Authentifizierungstyp* für das jeweilige Exemplar bestimmt, wie und wo der Benutzer überprüft wird. Der Authentifizierungstyp wird in der Konfigurationsdatei des Datenbankmanagers auf dem Server gespeichert. Sie wird erstmalig bei der Erstellung des Exemplars definiert. Es gibt einen Authentifizierungstyp pro Exemplar, der 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 Authentifizierungstypen zusammengeschlossener Datenbanken beachten.

Die folgenden Authentifizierungstypen 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.

**Anmerkungen:**

1. Der Servercode erkennt, ob eine Verbindung lokal oder fern ist. Für lokale Verbindungen sind beim Authentifizierungstyp SERVER keine Benutzer-ID und kein Kennwort zur erfolgreichen Authentifizierung erforderlich.
2. Wenn Sie DB2® Universal Database (DB2 UDB) zur Einrichtung einer Common Criteria-zertifizierten Konfiguration installieren, müssen Sie SERVER angeben.

### SERVER\_ENCRYPT

Gibt an, dass der Server verschlüsselte SERVER-Authentifizierungsschemata akzeptiert. Wenn die Clientauthentifizierung nicht angegeben ist, wird der Client mit der Methode überprüft, die auf dem Server ausgewählt ist.

Wenn als Clientauthentifizierung SERVER angegeben ist, wird der Client überprüft, indem die Benutzer-ID und das Kennwort an den Server übergeben werden. Wenn die Clientauthentifizierung SERVER\_ENCRYPT definiert ist, wird der Client überprüft, indem eine verschlüsselte 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 Clientknoten verglichen, um zu bestimmen, ob dem Benutzer der Zugriff auf das Exemplar gewährt wird. Es findet keine weitere Authentifizierung auf dem Datenbankserver statt. Dies wird manchmal als Einzelanmeldung (engl. single sign-on) bezeichnet.

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 der Authentifizierungstyp CLIENT definiert ist, bestimmen zwei weitere Parameter den endgültigen Authentifizierungstyp: *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 von Clients mit Windows® 9x-Betriebssystemen gesicherte Clients.

Wenn der Authentifizierungstyp 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 außerdem 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.

**Anmerkung:** Wenn bei gesicherten Clients beim Versuch mit CONNECT oder ATTACH eine Verbindung herzustellen, keine Benutzer-ID oder Kennwort explizit 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 bzw. USING angegeben werden, zu überprüfen sind.

Zum Schutz gegen alle Clients außer DRDA®-Clients von DB2 für OS/390® und z/OS™, DB2 für VM und VSE sowie DB2 für iSeries™ setzen Sie den Parameter *trust\_allclnts* auf den Wert DRDAONLY. Nur diese Clients dürfen die Authentifizierung auf der Clientseite 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 den Wert CLIENT gesetzt ist, findet die Authentifizierung auf dem Client statt. Wenn *trust\_clntauth* auf den Wert 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 5. Authentifizierungsmodi mit einer Kombination der Parameter TRUST\_ALLCLNTS und TRUST\_CLNTAUTH

TRUST_ALLCLNTS	TRUST_CLNTAUTH	Unge-sicherte Nicht-DRDA-Client-authentifizierung ohne Kennwort	Unge-sicherte 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

## KERBEROS

Wird verwendet, wenn sowohl der DB2 UDB-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 Services 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 (engl. single sign-on) an einem fernen DB2 UDB-Server durchgeführt werden. Der Authentifizierungstyp KERBEROS wird nur auf Clients und Servern unterstützt, die unter den Betriebssystemen Windows 2000, AIX® und Solaris Operating Environment ausgeführt werden.

Die Kerberos-Authentifizierung funktioniert wie folgt:

1. Ein Benutzer, der sich an der Clientmaschine mit einem Domänenkonto anmeldet, wird in der Kerberos-Schlüsselverteilungszentrale (KDC - Key Distribution Center) auf dem Domänencontroller authentifiziert. Die Schlüsselverteilungszentrale gibt ein so genanntes Ticket-granting Ticket (TGT) an den Client aus.
2. Während der ersten Phase der Verbindungsherstellung sendet der Server den Ziel-Principal-Namen, bei dem es sich um den Servicekontonamen für den Service des DB2 UDB-Servers handelt, an den Client. Mit Hilfe des Ziel-Principal-Namens des Servers und dem Ticket-granting Ticket fordert der Client ein Service-Ticket vom Ticket erteilenden Service (TGS - Ticket-Granting Service) an, der sich auf dem Domänencontroller befindet. Wenn sowohl das Ticket-granting Ticket des Clients als auch der Ziel-Principal-Name des Servers gültig sind, gibt der TGS ein Service-Ticket an den Client aus. Der Name des Principals, der im Datenbankverzeichnis aufgezeichnet wird, kann jetzt als 'name/exemplar@REALM' angegeben werden. (Dieses Format ist zusätzlich zu den aktuellen Formaten 'DOMÄNE\benutzerID' und 'benutzerID@xxx.xxx.xxx.com' möglich, die unter Windows 2000 von DB2 UDB Version 7.1 und späteren Versionen akzeptiert werden.)
3. Der Client sendet dieses Service-Ticket über den Kommunikationskanal (bei dem es sich beispielsweise um TCP/IP handeln kann) an den Server.
4. Der Server prüft die Gültigkeit des Server-Tickets des Clients. Wenn das Service-Ticket des Clients gültig ist, wird die Authentifizierung abgeschlossen.

Es ist möglich, die Datenbanken auf der Clientmaschine zu katalogisieren und den Authentifizierungstyp Kerberos explizit zusammen mit dem Ziel-Principal-Namen des Servers anzugeben. Auf diese Weise lässt sich die erste Phase der Verbindungsherstellung überspringen.

Wenn eine Benutzer-ID und ein Kennwort angegeben werden, fordert der Client das Ticket-granting Ticket für dieses Benutzerkonto an und verwendet es zur Authentifizierung.

#### **KRB\_SERVER\_ENCRYPT**

Gibt an, dass der Server Schemata der KERBEROS-Authentifizierung oder einer verschlüsselten SERVER-Authentifizierung akzeptiert. Wenn als Clientauthentifizierung KERBEROS angegeben wurde, wird der Client mit Hilfe des Kerberos-Sicherheitssystems authentifiziert. Wenn als Clientauthentifizierung SERVER\_ENCRYPT angegeben ist, wird der Client mit Hilfe einer Benutzer-ID und eines verschlüsselten Kennworts authentifiziert. Wenn die Clientauthentifizierung nicht angegeben ist, arbeitet der Client mit Kerberos, falls verfügbar, oder mit Kennwortverschlüsselung. Für andere Clientauthentifizierungstypen wird ein Authentifizierungsfehler zurückgegeben. Als Authentifizierungstyp des Clients kann nicht KRB\_SERVER\_ENCRYPT angegeben werden.

**Anmerkung:** Die Kerberos-Authentifizierungstypen werden nur auf Clients und Servern unterstützt, die unter den Betriebssystemen Windows 2000, Windows XP, Windows Server 2003 und AIX sowie in der Solaris-Betriebsumgebung ausgeführt werden. Darüber hinaus müssen sowohl die Client- als auch die Servermaschine entweder zur gleichen Windows-Domäne oder zu gesicherten (vertrauten) Domänen gehören. Dieser

Authentifizierungstyp sollte verwendet werden, wenn der Server Kerberos unterstützt und einige, jedoch nicht alle, Clientmaschinen die Kerberos-Authentifizierung unterstützen.

#### **DATA\_ENCRYPT**

Der Server akzeptiert verschlüsselte SERVER-Authentifizierungsschemata und die Verschlüsselung von Benutzerdaten. Die Authentifizierung funktioniert in exakt gleicher Weise wie für den Authentifizierungstyp SERVER\_ENCRYPT erläutert. Weitere Informationen finden Sie dort.

Die folgenden Benutzerdaten werden bei Verwendung dieses Authentifizierungstyps verschlüsselt:

- SQL-Anweisungen
- Daten von SQL-Programmvariablen
- Ausgabedaten aus der Serververarbeitung einer SQL-Anweisung, einschließlich einer Beschreibung der Daten
- Einige oder alle Antwortgruppendaten, die aus einer Abfrage resultieren
- LOB-Datenströme (große Objekte)
- SQLDA-Deskriptoren

#### **DATA\_ENCRYPT\_CMP**

Der Server akzeptiert verschlüsselte SERVER-Authentifizierungsschemata und die Verschlüsselung von Benutzerdaten. Darüber hinaus bietet dieser Authentifizierungstyp Kompatibilität mit Produkten früherer Versionen, die den Authentifizierungstyp DATA\_ENCRYPT nicht unterstützen. Diese Produkte erhalten die Möglichkeit, die Verbindung mit dem Authentifizierungstyp SERVER\_ENCRYPT und ohne Verschlüsselung von Benutzerdaten herzustellen. Produkte, die den neuen Authentifizierungstyp unterstützen, müssen ihn verwenden. Dieser Authentifizierungstyp ist nur in der Konfigurationsdatei des Datenbankmanagers des Servers, jedoch nicht im Befehl CATALOG DATABASE gültig.

#### **GSSPLUGIN**

Gibt an, dass der Server ein GSS-API-Plug-in zur Ausführung der Authentifizierung verwendet. Wenn die Clientauthentifizierung nicht angegeben ist, gibt der Server eine Liste der vom Server unterstützten Plug-ins, einschließlich aller Kerberos-Plug-ins, die im Konfigurationsparameter *srvcon\_gssplugin\_list* des Datenbankmanagers aufgelistet sind, an den Client zurück. Der Client wählt das erste Plug-in in der Liste aus, das im Plug-in-Verzeichnis des Clients zu finden ist. Wenn der Client kein Plug-in der Liste unterstützt, wird der Client mit dem Kerberos-Authentifizierungsschema authentifiziert (falls dies zurückgegeben wird). Wenn der Clientauthentifizierungstyp das Authentifizierungsschema GSSPLUGIN angibt, wird der Client mit dem ersten unterstützten Plug-in der Liste authentifiziert.

#### **GSS\_SERVER\_ENCRYPT**

Gibt an, dass der Server Plug-in-Authentifizierungsschemata oder verschlüsselte SERVER-Authentifizierungsschemata akzeptiert. Wenn die Clientauthentifizierung über ein Plug-in erfolgt, wird der Client mit dem ersten vom Client unterstützten Plug-in in der Liste der vom Server unterstützten Plug-ins authentifiziert.

Wenn die Clientauthentifizierung nicht angegeben ist und eine implizite Verbindung hergestellt wird (d. h., der Client gibt keine Benutzer-ID und kein Kennwort bei der Herstellung der Verbindung an), gibt der Server eine Liste der vom Server unterstützten Plug-ins, das Kerberos-Authentifi-

zierungsschema (wenn eines der Plug-ins in der Liste auf Kerberos basiert) und das verschlüsselte SERVER-Authentifizierungsschema zurück. Der Client wird mit dem ersten unterstützten Plug-in in der Liste authentifiziert, das im Plug-in-Verzeichnis des Clients zu finden ist. Wenn der Client keines der Plug-ins der Liste unterstützt, wird der Client mit dem Kerberos-Authentifizierungsschema authentifiziert. Wenn der Client das Kerberos-Authentifizierungsschema nicht unterstützt, wird der Client mit dem verschlüsselten SERVER-Authentifizierungsschema authentifiziert und die Verbindung schlägt fehl, weil das Kennwort fehlt. Ein Client unterstützt das Kerberos-Authentifizierungsschema, wenn für das Betriebssystem ein von DB2 UDB geliefertes Kerberos-Plug-in vorhanden ist oder ein Kerberos-basiertes Plug-in im Konfigurationsparameter *srocon\_gssplugin\_list* des Datenbankmanagers angegeben ist.

Wenn die Clientauthentifizierung nicht angegeben ist und eine explizite Verbindung hergestellt wird (d. h., die Benutzer-ID und das Kennwort werden angegeben), ist der Authentifizierungstyp mit SERVER\_ENCRYPT äquivalent.

#### **Anmerkungen:**

1. 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 UDB-System aussperren, gibt es auf allen Plattformen eine Sicherheitsoption, die Ihnen ermöglicht, die normalen DB2 UDB-Sicherheitsprüfungen außer Kraft zu setzen, um die Konfigurationsdatei des Datenbankmanagers über einen Sicherheitsbenutzer des lokalen Betriebssystems, der über Zugriffsrechte einer hohen Berechtigungsstufe verfügt, 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 UDB-Befehl verwenden. Dieser Benutzer mit Sonderberechtigung wird folgendermaßen identifiziert:

- UNIX<sup>®</sup>-Plattformen: der Exemplareigner.
- NT-Plattform: eine zur lokalen Gruppe der Administratoren gehörende Person.
- Andere Plattformen: Da es auf anderen Plattformen keine lokalen Sicherheitseinrichtungen gibt, bestehen alle Benutzer lokale Sicherheitsprüfungen ohnehin.

**Zugehörige Konzepte:**

- „Authentifizierungsaspekte für ferne Clients“ auf Seite 240
- „Authentifizierungsaspekte bei partitionierten Datenbanken“ auf Seite 240
- „Einführung in DB2 für Windows NT und in die Windows NT-Sicherheit“ auf Seite 401

**Zugehörige Referenzen:**

- „authentication - Authentifizierungstyp“ in *Systemverwaltung: Optimierung*
- „trust\_allclnts - Alle Clients akzeptieren“ in *Systemverwaltung: Optimierung*
- „trust\_clntauth - Authentifizierung gesicherter Clients“ in *Systemverwaltung: Optimierung*

---

## Authentifizierungsaspekte für ferne Clients

Bei der Katalogisierung einer Datenbank für den fernen Zugriff, kann der Authentifizierungstyp im Datenbankverzeichniseintrag angegeben werden.

Für Datenbanken, auf die über DB2<sup>®</sup> Connect zugegriffen wird, gilt: Wenn kein Wert angegeben ist, wird der Authentifizierungstyp SERVER angenommen.

Der Authentifizierungstyp ist nicht erforderlich. Wenn er nicht angegeben ist, nimmt der Client den Standardwert SERVER\_ENCRYPT an. Wenn der Server jedoch den Authentifizierungstyp SERVER\_ENCRYPT nicht unterstützt, versucht es der Client mit einem vom Server unterstützten Wert erneut. Wenn der Server mehrere Authentifizierungstypen unterstützt, trifft der Client keine Auswahl unter diesen, sondern gibt stattdessen einen Fehler zurück. Die Rückgabe eines Fehlers soll sicherstellen, dass der korrekte Authentifizierungstyp verwendet wird. In diesem Fall muss der Client die Datenbank mit einem unterstützten Authentifizierungstyp katalogisieren. Wenn ein Authentifizierungstyp angegeben wird, kann die Authentifizierung unverzüglich beginnen, sofern der angegebene Wert mit dem auf dem Server übereinstimmt. Wenn keine Übereinstimmung erkannt wird, unternimmt DB2 Universal Database<sup>™</sup> (DB2 UDB) einen Behebungsversuch. Dieser Behebungsversuch kann zum Ausgleich des Unterschieds zu mehr Übertragungen führen oder einen Fehler verursachen, wenn DB2 UDB keine Behebung möglich ist. Im Fall einer fehlenden Übereinstimmung wird angenommen, dass der Wert auf dem Server korrekt ist.

**Zugehörige Konzepte:**

- „Authentifizierungsmethoden für den Server“ auf Seite 234

---

## Authentifizierungsaspekte bei 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.

**Zugehörige Konzepte:**

- „Authentifizierungsmethoden für den Server“ auf Seite 234



---

## Kerberos-Authentifizierung - Details

DB2® Universal Database (DB2 UDB) bietet Unterstützung für das Kerberos-Authentifizierungsprotokoll für die Betriebssysteme AIX® Version 5.2, Solaris Operating Environment Version 8, Red Hat Enterprise Linux Server Advanced Server 2.1 (32-Bit-Intel) und Windows® 2000 (und späteren Versionen).

Die Kerberos-Unterstützung wird in Form eines GSS-API-Sicherheits-Plug-ins mit dem Namen „IBMkrb5“ zur Verfügung gestellt, das als Authentifizierungs-Plug-in sowohl auf dem Server als auch auf dem Client verwendet wird. Die Bibliothek befindet sich unter UNIX® und Linux in den Verzeichnissen `sqllib/security{32|64}/plugin/IBM/{client|server}` und unter Windows in den Verzeichnissen `sqllib/security/plugin/IBM{client|server}`.

**Anmerkung:** Für 64-Bit-Windows hat die Plug-in-Bibliothek den Namen `IBMkrb564.dll`. Darüber hinaus ist der tatsächliche Plug-in-Quellcode `IBMkrb5.C` für das UNIX- und Linux-Plug-in im Verzeichnis `sqllib/samples/security/plugins` verfügbar.

Es wird dringend empfohlen, sich gute Kenntnisse über die Verwendung und Konfiguration von Kerberos anzueignen, bevor Sie versuchen, die Kerberos-Authentifizierung mit DB2 UDB zu verwenden.

## Kerberos - Beschreibung und Einführung

Kerberos ist ein Netzwerkauthentifizierungsprotokoll eines Fremdanbieters, das mit einem System gemeinsamer geheimer Schlüssel (Shared Secret Keys) zur sicheren Authentifizierung eines Benutzers in einer ungesicherten Netzwerkumgebung arbeitet. In den späten 80er Jahren (1980er) am MIT entwickelt, wird die aktuelle Revision Kerberos 5 im RFC 1510 der Internet Engineering Task Force (IETF) beschrieben. (Die URL-Adresse der IETF lautet: <http://www.ietf.org>. RFC 1510 trägt den Titel „The Kerberos Network Authentication Service (V5)“.) Kerberos implementiert ein dreistufiges System, in dem verschlüsselte Tickets (die von einem getrennten Server bereitgestellt werden, der als Kerberos Key Distribution Center (Kerberos-Schlüsselverteilungszentrale), bzw. kurz KDC bezeichnet wird) zwischen dem Anwendungsserver und dem Client anstelle von Paaren aus Benutzer-ID und Kennwort im Textformat ausgetauscht werden. Diese verschlüsselten Service-Tickets (auch als *Credentials* - Identitätsnachweis bezeichnet) haben eine endliche Gültigkeitsdauer und werden nur vom Client und dem Server verstanden. Dies verringert das Sicherheitsrisiko, selbst wenn das Ticket im Netzwerk abgefangen wird. Jeder Benutzer, bzw. in der Kerberos-Terminologie *Principal*, besitzt einen privaten Chiffrierschlüssel, der mit dem KDC gemeinsam verwendet wird. Zusammengefasst werden die Gruppe von Principals und Computern, die bei einem KDC registriert sind, als *Realm* bezeichnet.

Eine Schlüsselfunktion von Kerberos besteht darin, eine Einzelanmeldungs-umgebung (Single Sign-on) zu ermöglichen, in der ein Benutzer seine Identität den Ressourcen innerhalb des Kerberos-Realms gegenüber nur einmal nachweisen muss. Für die Arbeit mit DB2 UDB bedeutet dies, dass ein Benutzer in der Lage ist, eine Verbindung zu einem DB2 UDB-Server herzustellen (CONNECT oder ATTACH), ohne eine Benutzer-ID und ein Kennwort anzugeben. Ein weiterer Vorteil ist der, dass die Verwaltung von Benutzer-IDs vereinfacht wird, da ein zentrales Repository für Principals verwendet wird. Und schließlich unterstützt Kerberos eine gegenseitige Authentifizierung, so dass der Client auch die Identität des Servers überprüfen kann.

## Kerberos - Einrichtung

DB2 UDB und die implementierte Unterstützung von Kerberos beruhen darauf, dass die Kerberos-Schicht auf allen beteiligten Maschinen vor der Einbindung von DB2 UDB ordnungsgemäß installiert und konfiguriert ist. Dies umfasst unter anderem die Erfüllung der folgenden Voraussetzungen:

1. Die Client- und Servermaschinen sowie die Principals müssen zum gleichen Realm oder ansonsten zu vertrauten (gesicherten) Realms (bzw. vertrauten Domänen in der Windows-Terminologie) gehören.
2. Es müssen geeignete Principals erstellt werden.
3. Es müssen Serverchiffrierschlüsseldateien (keytab) nach Bedarf erstellt werden.
4. Auf allen beteiligten Maschinen müssen die Systemuhren synchronisiert werden (Kerberos lässt in der Regel einen Zeitunterschied von fünf Minuten zu, ansonsten kann ein Vorauthentifizierungsfehler beim Empfang des Identitätsnachweises auftreten).

Detaillierte Informationen zur Installation und Konfiguration von Kerberos finden Sie in der Dokumentation des installierten Kerberos-Produkts.

Für DB2 UDB ist allein wichtig, ob der Kerberos-Sicherheitskontext auf der Basis der von der Anwendung, die die Verbindung herstellt, bereitgestellten Identitätsnachweise erfolgreich erstellt wird (d. h., ob die Authentifizierung erfolgreich ist). Andere Kerberos-Funktionen, wie zum Beispiel die Signierung oder Verschlüsselung von Nachrichten, werden nicht verwendet. Darüber hinaus wird, sofern verfügbar, die gegenseitige Authentifizierung unterstützt.

Für Kerberos gelten folgende Voraussetzungen:

- AIX Version 5.2 mit IBM® Network Authentication Service (NAS) Toolkit 1.3
- Solaris Operating Environment Version 8 mit SEAM (Sun Enterprise Authentication Mechanism) und IBM NAS Toolkit 1.3
- Red Hat Enterprise Linux Advanced Server 2.1 mit den Dateien krb5-libs und krb5-workstation
- Windows 2000 Server (und spätere Serverbetriebssysteme)

## Kerberos und Client-Principals

Der Principal kann entweder in einem zweiteiligen oder mehrteiligen Format vorliegen (d. h. in der Form *name@REALM* oder *name/exemplar@REALM*). Da der Wert für „name“ in der Zuordnung der Berechtigungs-ID (AUTHID) verwendet wird, muss der Name den DB2 UDB-Namenskonventionen entsprechen. Dies bedeutet, dass der Name bis zu 30 Zeichen lang sein darf und die vorhandenen Einschränkungen bei der Auswahl der Zeichen berücksichtigt. (Die AUTHID-Zuordnung wird in einem späteren Abschnitt behandelt.)

**Anmerkung:** Windows ordnet den Kerberos-Principal direkt einem Domänenbenutzer zu. Daraus folgt, dass die Kerberos-Authentifizierung für Windows-Maschinen, die nicht zu einer Domäne oder einem Realm gehören, nicht zur Verfügung steht. Darüber hinaus unterstützt Windows lediglich zweiteilige Namen (d. h. der Form *name@domäne*).

Der Principal selbst muss in der Lage sein, abgehende Identitätsnachweise abzurufen, mit denen er Service-Tickets für die Zieldatenbank anfordern und empfangen kann. Dies geschieht normalerweise mit Hilfe des Befehls **kinit** unter UNIX oder Linux und erfolgt implizit bei der Anmeldung an Windows.

## Kerberos und Zuordnung von Berechtigungs-IDs

Im Unterschied zu Benutzer-IDs von Betriebssystemen, deren Existenzbereich im Normalfall auf eine einzige Maschine beschränkt ist (mit der besonderen Ausnahme von NIS), bieten Kerberos-Principals die Möglichkeit einer Authentifizierung auch in anderen Realms als dem eigenen. Das potenzielle Problem doppelter Namen von Principals wird dadurch vermieden, dass der Principal mit dem Realmnamen vollständig qualifiziert wird. In Kerberos hat ein vollständig qualifizierter Principal die Form `name/exemplar@REALM`. Dabei kann das Exemplarfeld tatsächlich sogar mehrere, durch „/“ getrennte Exemplare enthalten, d. h. `name/exemplar1/exemplar2@REALM`, oder es kann ganz weggelassen werden. Dies unterliegt natürlich der Einschränkung, dass der Realmname innerhalb aller im Netzwerk definierten Realms eindeutig sein muss. Das Problem für DB2 UDB ist, dass zur Bereitstellung einer einfachen Zuordnung zwischen Principal und Berechtigungs-ID (AUTHID) eine Eins-zu-eins-Zuordnung zwischen dem Namen des Principals, das heißt dem Wert für „name“ im vollständig qualifizierten Principal, und der Berechtigungs-ID wünschenswert ist. Eine einfache Zuordnung ist erforderlich, da die Berechtigungs-ID als Standardschema in DB2 UDB verwendet wird und daher problemlos und logisch ableitbar sein sollte. Infolgedessen muss der Datenbankadministrator mit den folgenden potenziellen Problemen rechnen:

- Principals aus verschiedenen Realms, jedoch mit dem gleichen Namen, werden derselben Berechtigungs-ID (AUTHID) zugeordnet.
- Principals mit dem gleichen Namen, jedoch verschiedenen Exemplaren, werden derselben Berechtigungs-ID (AUTHID) zugeordnet.

In Anbetracht der oben genannten Umstände ergeben sich die folgenden Empfehlungen:

- Pflegen Sie einen eindeutigen Namensbereich (Namespace) für den Namen innerhalb aller vertrauten (gesicherten) Realms, von denen aus auf den DB2 UDB-Server zugegriffen wird.
- Alle Principals mit dem gleichen Namen sollten unabhängig vom Exemplar demselben Benutzer gehören.

## Kerberos und Server-Principals

Unter UNIX oder Linux wird als Name des Server-Principals für das DB2 UDB-Exemplar ein Name der Form `<exemplarname>/<vollständig qualifizierter hostname>@REALM` angenommen. Dieser Principal muss in der Lage sein, Kerberos-Sicherheitskontexte zu akzeptieren, und muss vorhanden sein, bevor das DB2 UDB-Exemplar gestartet wird, da der Servername durch das Plug-in bei der Initialisierung an DB2 UDB gemeldet wird.

Unter Windows wird als Principal für den Server das Domänenkonto angenommen, unter dem der DB2 UDB-Service gestartet wurde. Eine Ausnahme bildet der Fall, dass das Exemplar durch das lokale Systemkonto gestartet wird. In diesem Fall wird der Name des Server-Principals in der Form `host/<hostname>` gemeldet. Dies ist nur gültig, wenn sowohl der Client als auch der Server zu Windows-Domänen gehören.

Windows unterstützt keine Namen, die aus mehr als zwei Teilen bestehen. Dies stellt ein Problem dar, wenn ein Windows-Client versucht, eine Verbindung zu einem UNIX-Server aufzubauen. Daher muss eventuell eine Zuordnung zwischen dem Kerberos-Principal und dem Windows-Konto in der Windows-Domäne definiert werden, wenn Interoperabilität mit UNIX-Kerberos erforderlich ist. (Anleitungen zu diesem Thema finden Sie in der entsprechenden Microsoft®-Dokumentation.)

## Kerberos-Chiffrierschlüsseldateien

Jeder Kerberos-Service unter UNIX oder Linux, der Sicherheitskontextanforderungen akzeptieren soll, muss seinen Identitätsnachweis in einer Chiffrierschlüsseldatei (*keytab*) hinterlegen. Dies gilt für die Principals, die als Server-Principals von DB2 UDB verwendet werden. Der Schlüssel des Servers wird nur in der Standardchiffrierschlüsseldatei gesucht. Anweisungen zum Hinzufügen eines Schlüssels zur Chiffrierschlüsseldatei finden Sie in der Dokumentation zu Ihrem Kerberos-Produkt.

Das Konzept einer Chiffrierschlüsseldatei ist unter Windows unbekannt. Das System sorgt automatisch für das Speichern und Abrufen der Kennungen für Identitätsnachweise für einen Principal.

## Kerberos und Gruppen

Kerberos ist ein Authentifizierungsprotokoll, das kein Gruppenkonzept besitzt. Infolgedessen greift DB2 UDB auf das lokale Betriebssystem zurück, um eine Gruppenliste für den Kerberos-Principal abzurufen. Für UNIX oder Linux setzt dies voraus, dass ein äquivalentes Systemkonto für jeden Principal vorhanden sein muss. Für den Principal `name@REALM` sammelt DB2 UDB zum Beispiel Gruppeninformationen, indem das lokale Betriebssystem nach den Namen aller Gruppen abgefragt wird, zu denen der Betriebssystembenutzer *name* gehört. Wenn kein Betriebssystembenutzer vorhanden ist, wird die Berechtigungs-ID (AUTHID) nur der Gruppe PUBLIC zugeordnet. Windows hingegen ordnet einem Kerberos-Principal automatisch ein Domänenkonto zu und der zusätzliche Schritt zur Erstellung eines getrennten Betriebssystemkontos entfällt.

## Aktivieren der Kerberos-Authentifizierung auf dem Client

Der Konfigurationsparameter `clnt_krb_plugin` des Datenbankmanagers muss mit dem Namen des verwendeten Kerberos-Plug-ins aktualisiert werden. Auf den unterstützten Plattformen muss dieser Parameter auf den Wert `IBMkrb5` gesetzt werden. Dieser Parameter teilt DB2 UDB mit, dass Kerberos zum Verbindungsaufbau und zu lokalen Aktionen auf Exemplarebene verwendet werden kann, wenn der Parameter `AUTHENTICATION` auf den Wert `KERBEROS` oder `KRB_SERVER_ENCRYPT` gesetzt ist. Ansonsten wird keine clientseitige Kerberos-Unterstützung angenommen.

**Anmerkung:** Es werden keine Prüfungen durchgeführt, um festzustellen, ob die Kerberos-Unterstützung verfügbar ist.

Optional kann beim Katalogisieren einer Datenbank auf dem Client der Authentifizierungstyp wie folgt angegeben werden:

```
db2 catalog db testdb at node testnode authentication kerberos target
principal service/host@REALM
```

Wenn die Authentifizierungsinformationen jedoch nicht angegeben werden, sendet der Server dem Client den Namen des Server-Principals.

## Aktivieren der Kerberos-Authentifizierung auf dem Server

Der Konfigurationsparameter `srvcon_gssplugin_list` des Datenbankmanagers muss mit dem Namen des Kerberos-Plug-ins des Servers aktualisiert werden. Obwohl dieser Parameter eine Liste von unterstützten Plug-ins enthalten kann, darf nur ein Kerberos-Plug-in angegeben werden. Wenn dieses Feld jedoch leer ist und `AUTHENTICATION` auf den Wert `KERBEROS` oder `KRB_SERVER_ENCRYPT` gesetzt ist, wird das Kerberos-Standard-Plug-in (`IBMkrb5`) angenommen und ver-

wendet. Entweder der Parameter AUTHENTICATION oder der Parameter SVR-CON\_AUTH muss auf den Wert KERBEROS oder KRB\_SERVER\_ENCRYPT gesetzt werden, wenn die Kerberos-Authentifizierung abhängig davon verwendet werden soll, ob sie für alle Verbindungen oder nur für ankommende Verbindungen zu verwenden ist.

## Erstellen eines Kerberos-Plug-ins

Für die Erstellung eines Kerberos-Plug-ins sind verschiedene Aspekte zu beachten:

- Schreiben Sie ein Kerberos-Plug-in als GSS-API-Plug-in mit der besonderen Ausnahme, dass der Parameter *plugintype* im Funktionszeigerfeld, das an DB2 UDB in der Initialisierungsfunktion zurückgegeben wird, auf den Wert DB2SEC\_PLUGIN\_TYPE\_KERBEROS gesetzt werden muss.
- Unter bestimmten Bedingungen kann der Name des Server-Principals durch den Server an den Client gemeldet werden. Als solcher sollte der Name des Principals nicht im GSS\_C\_NT\_HOSTBASED\_SERVICE-Format (service@host) angegeben werden, da DRDA<sup>®</sup> festlegt, dass der Name des Principals im GSS\_C\_NT\_USER\_NAME-Format (server/host@REALM) anzugeben ist.
- In einer typischen Situation kann die Standardchiffrierschlüsseldatei (keytab) durch die Umgebungsvariable KRB5\_KTNAME angegeben werden. Da das Server-Plug-in jedoch innerhalb eines DB2 UDB-Engineprozesses ausgeführt wird, ist vielleicht kein Zugriff auf diese Umgebungsvariable möglich.

### Zugehörige Konzepte:

- „Authentifizierungsmethoden für den Server“ auf Seite 234

---

## Zugriffsrechte, Berechtigungsstufen und Datenbankberechtigungen

*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 auf höherer Ebene. *Datenbankberechtigungen* ermöglichen Benutzern die Ausführung von Aktivitäten auf der Datenbankebene. Zugriffsrechte, Berechtigungsstufen und Datenbankberechtigungen können zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte verwendet werden. Benutzer können nur auf die Objekte zugreifen, für die sie über das erforderliche Zugriffsrecht, die Berechtigungsstufe oder Datenbankberechtigung verfügen. Dies wird von DB2<sup>®</sup> Universal Database (DB2 UDB) bei der Ausführung einer Berechtigungsprüfung für einen authentifizierten Benutzer festgestellt.

Der Datenbankmanager setzt voraus, dass jeder Benutzer entweder implizit oder explizit zur Verwendung der jeweiligen Datenbankfunktion speziell berechtigt wird, um eine bestimmte Task auszuführen. *Explizite* Berechtigungen oder Zugriffsrechte werden dem Benutzer erteilt (GRANTEETYPE hat den Wert U in den Datenbankkatalogen). *Implizite* Berechtigungen oder Zugriffsrechte werden einer Gruppe erteilt, der der Benutzer angehört (GRANTEETYPE hat den Wert G in den Datenbankkatalogen). Das heißt, zur Erstellung einer Tabelle muss ein Benutzer berechtigt sein, Tabellen zu erstellen, zur Änderung einer Tabelle muss ein Benutzer berechtigt sein, die Tabelle zu ändern, usw.

In Abb. 3 sind die Abhängigkeiten zwischen Berechtigungen und deren Gültigkeitsbereichen (Datenbank, Datenbankmanager) dargestellt.

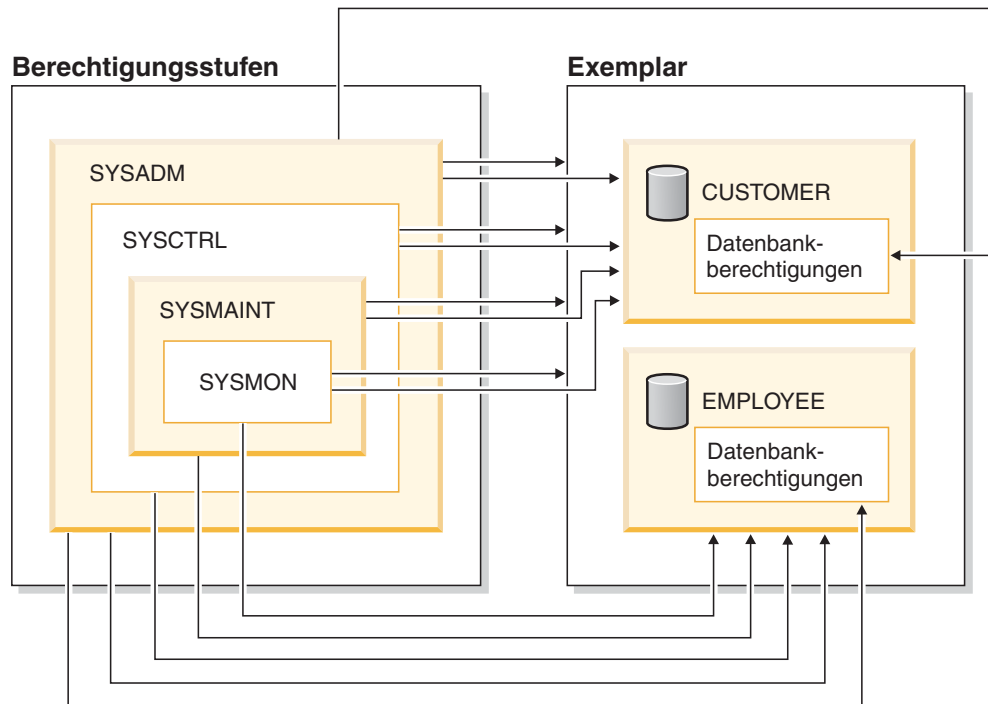


Abbildung 3. Hierarchie der Berechtigungen

Ein Benutzer oder eine Gruppe können über eine oder mehrere der nachfolgend aufgeführten Berechtigungen (bzw. Zugriffsrechte) verfügen:

- Administratorberechtigung:

- SYSADM (Systemadministrator)

Die Berechtigungsstufe SYSADM berechtigt zur Steuerung aller Ressourcen, die vom Datenbankmanager erstellt oder verwaltet werden. Der Systemadministrator besitzt sämtliche Berechtigungen, d. h. DBADM, SYSCTRL, SYSMOINT und SYSMON, sowie die Berechtigung, die Berechtigung DBADM zu erteilen oder zu widerrufen.

Der Benutzer, der die Berechtigung SYSADM besitzt, ist sowohl für die Steuerung des Datenbankmanagers als auch für die Gewährleistung der Sicherheit und Integrität der Daten zuständig. Die Berechtigung SYSADM erteilt implizite Zugriffsrechte für alle Objekte in der Datenbank und ermöglicht die Steuerung, welche Benutzer auf den Datenbankmanager zugreifen können und in welchem Umfang sie dies können. Weitere Informationen zur Berechtigung SYSADM finden Sie unter "Systemverwaltungsberechtigung (SYSADM)".

- DBADM (Datenbankadministrator)

Die Datenbankberechtigung DBADM erteilt Verwaltungsberechtigung über eine einzelne Datenbank. Dieser Datenbankadministrator besitzt die Zugriffsrechte, die zur Erstellung von Objekten, zur Ausführung von Datenbankbefehlen und zum Zugriff auf Tabellendaten erforderlich sind. Der Datenbankadministrator kann außerdem das Zugriffsrecht CONTROL und Einzelzugriffsrechte erteilen oder widerrufen. Weitere Informationen finden Sie unter "Datenbankadministratorberechtigung (DBADM)".

|

|

- Systemsteuerungsberechtigung:

- SYSCTRL (Systemsteuerung)

| Die Berechtigungsstufe SYSCTRL berechtigt zur Steuerung von Operationen, die sich auf Systemressourcen beziehen. Zum Beispiel kann ein Benutzer mit der Berechtigung SYSCTRL eine Datenbank erstellen, aktualisieren, stoppen oder löschen. Ein solcher Benutzer kann darüber hinaus ein Exemplar stoppen, jedoch nicht auf Tabellendaten zugreifen. Benutzer mit der Berechtigung SYSCTRL verfügen auch über die Berechtigung SYSMON. Weitere Informationen finden Sie unter "Systemsteuerungsberechtigung (SYSCTRL)".

- SYSMAINT (Systempflege)

| Die Berechtigungsstufe SYSMAINT erteilt die Berechtigung, die zur Ausführung von Pflegeoperationen an allen zu einem Exemplar gehörenden Datenbanken erforderlich ist. Ein Benutzer mit der Berechtigung SYSMAINT kann die Datenbankkonfiguration aktualisieren, eine Datenbank oder einen Tabellenbereich sichern, eine vorhandene Datenbank wiederherstellen und eine Datenbank überwachen. Ebenso wie SYSCTRL erteilt SYSMAINT keinen Zugriff auf Tabellendaten. Benutzer mit der Berechtigung SYSMAINT verfügen auch über die Berechtigung SYSMON. Weitere Informationen finden Sie unter "Systempflegeberechtigung (SYSMAINT)".

- SYSMON (Systemmonitorberechtigung)

| Die Berechtigungsstufe SYSMON erteilt die Berechtigung, die zur Verwendung des Datenbanksystemmonitors erforderlich ist. Weitere Informationen finden Sie unter "Systemmonitorberechtigung (SYSMON)".

- Datenbankberechtigungen

| Zur Ausführung solcher Aktivitäten wie das Erstellen einer Tabelle oder einer Routine oder das Laden von Daten in eine Tabelle sind bestimmte Datenbankberechtigungen erforderlich. Weitere Informationen finden Sie unter "Datenbankberechtigungen".

- Zugriffsrechte:

| Zugriffsrechte sind erforderlich, um Aktivitäten an Datenbankobjekten auszuführen (z. B. zum Erstellen und Löschen eines Index). Zugriffsrechte sind exakte Definitionen der Tasks, die ein Benutzer ausführen kann. Zum Beispiel kann ein Benutzer das Zugriffsrecht haben, einen Index für eine Tabelle erstellen, jedoch nicht das Zugriffsrecht, einen Auslöser für die gleiche Tabelle zu erstellen.

- CONTROL (Zugriffsrecht)

| Der Besitz des Zugriffsrechts CONTROL für ein Objekt berechtigt einen Benutzer zum Zugriff auf dieses Datenbankobjekt sowie zum Erteilen und Widerrufen von Zugriffsrechten für andere Benutzer in Bezug auf dieses Objekt.

| **Anmerkung:** Das Zugriffsrecht CONTROL gilt nur für Tabellen, Sichten, Kurznamen und Pakete.

| Wenn ein anderer Benutzer das Zugriffsrecht CONTROL für dieses Objekt benötigt, muss ein Benutzer mit der Berechtigung SYSADM oder DBADM das Zugriffsrecht CONTROL für dieses Objekt erteilen. Das Zugriffsrecht CONTROL kann dem Objekteigner nicht entzogen werden.

| In einigen Fällen erhält der Ersteller eines Objekts automatisch das Zugriffsrecht CONTROL für das Objekt. Weitere Informationen finden Sie unter "Objekterstellung, Eigentumsrechte und Zugriffsrechte".

- Individuelle Zugriffsrechte können erteilt werden, um einem Benutzer die Ausführung bestimmter Tasks für spezielle Objekte zu erlauben.  
Benutzer mit einer Administratorberechtigung (SYSADM oder DBADM) oder dem Zugriffsrecht CONTROL können anderen Benutzern Zugriffsrechte erteilen und entziehen.  
Einzelne Zugriffsrechte und Datenbankberechtigungen erlauben eine bestimmte Funktion, jedoch beinhalten sie nicht das Recht, die gleichen Zugriffsrechte oder Berechtigungen anderen Benutzern zu erteilen. Das Recht, Zugriffsrechte für Tabellen, Sichten, Schemata, Pakete, Routinen und Sequenzen zu erteilen kann durch die Klausel WITH GRANT OPTION in der Anweisung GRANT auf andere Benutzer übertragen werden. Jedoch erlaubt die Klausel WITH GRANT OPTION der Person, die das Zugriffsrecht erteilt, nicht, das erteilte Zugriffsrecht zu widerrufen. Sie müssen die Berechtigung SYSADM bzw. DBADM oder das Zugriffsrecht CONTROL haben, um das Zugriffsrecht widerrufen zu können.  
Zugriffsrechte können auch der Gruppe PUBLIC erteilt werden. Zugriffsrechte der Gruppe PUBLIC gelten für alle Benutzer (Berechtigungsnamen), einschließlich zukünftiger Benutzer, unabhängig davon, ob einzelne Benutzer zuvor das Zugriffsrecht bereits erhalten haben.
- 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.

Ein Benutzer oder eine Gruppe kann für jede Kombination individueller Zugriffsrechte oder Berechtigungen berechtigt werden. Wenn ein Zugriffsrecht einem Objekt zugeordnet wird, muss dieses Objekt vorhanden sein. 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.

Die Anweisung REVOKE dient zum Widerrufen zuvor erteilter Zugriffsrechte. In DB2 UDB wird durch das Widerrufen eines Zugriffsrechts für einen Berechtigungsnamen das von allen Berechtigungsnamen erteilte Zugriffsrecht widerrufen.

Durch das Widerrufen eines Zugriffsrechts für einen Berechtigungsnamen wird das gleiche Zugriffsrecht nicht für andere Berechtigungsnamen widerrufen, die das Zugriffsrecht von diesem Berechtigungsnamen erhalten haben. Nehmen Sie zum Beispiel an, dass die Benutzerin CLAIRE dem Benutzer RICK das Zugriffsrecht SELECT WITH GRANT OPTION erteilt, und anschließend der Benutzer RICK das Zugriffsrecht SELECT den Benutzern BOBBY und CHRIS erteilt. Wenn nun die Benutzerin CLAIRE das Zugriffsrecht SELECT für RICK widerruft, behalten die Benutzer BOBBY und CHRIS das Zugriffsrecht SELECT.

#### **Zugehörige Konzepte:**

- „Systemverwaltungsberechtigung (SYSADM)“ auf Seite 250
- „Systemsteuerungsberechtigung (SYSCTRL)“ auf Seite 251
- „Systempflegeberechtigung (SYSMAINT)“ auf Seite 252



- „Datenbankadministratorberechtigung (DBADM)“ auf Seite 253
- „LOAD-Berechtigung“ auf Seite 254
- „Datenbankberechtigungen“ auf Seite 255
- „Zugriffsrechte für Schemata“ auf Seite 257
- „Zugriffsrechte für Tabellenbereiche“ auf Seite 259
- „Zugriffsrechte für Tabellen und Sichten“ auf Seite 259
- „Zugriffsrechte für Pakete“ auf Seite 261
- „Zugriffsrechte für Indizes“ auf Seite 262
- „Zugriffsrechte für Sequenzen“ auf Seite 263
- „Steuern des Zugriffs auf Datenbankobjekte“ auf Seite 263
- „Indirekte Zugriffsrechte durch ein Paket“ auf Seite 268
- „Zugriffsrechte für Routinen“ auf Seite 263
- „Objekterstellung, Eigentumsrechte und Zugriffsrechte“ auf Seite 249
- „Systemmonitorberechtigung (SYSMON)“ auf Seite 253

---

## Objekterstellung, Eigentumsrechte und Zugriffsrechte

Wenn ein Objekt erstellt wird, wird einem Berechtigungsnamen das *Eigentumsrecht* an diesem Objekt zugeordnet. Eigentumsrecht bedeutet, dass der Benutzer berechtigt ist, in jeder beliebigen SQL-Anweisung auf das Objekt zu verweisen.

Wenn ein Objekt innerhalb eines Schemas erstellt wird, muss die Berechtigungs-ID der Anweisung über das erforderliche Zugriffsrecht verfügen, um Objekte in dem implizit oder explizit angegebenen Schema erstellen zu können. Das bedeutet, der Berechtigungsname muss entweder der Eigner des Schemas sein oder das Zugriffsrecht CREATEIN für das Schema besitzen.

**Anmerkung:** Diese Voraussetzung gilt nicht für die Erstellung von Tabellenbereichen, Pufferpools oder Datenbankpartitionsgruppen. Solche Objekte werden nicht in Schemata erstellt.

Wenn ein Objekt erstellt wird, ist die Berechtigungs-ID der Anweisung der Eigner dieses Objekts.

**Anmerkung:** Es gibt eine Ausnahme. Wenn die Option AUTHORIZATION in der Anweisung CREATE SCHEMA angegeben wird, ist jedes andere Objekt, das im Rahmen der CREATE SCHEMA-Operation erstellt wird, der Berechtigungs-ID eigen, die in der Option AUTHORIZATION angegeben wird. Alle Objekte, die in dem Schema nach der ersten CREATE SCHEMA-Operation erstellt werden, sind jedoch der Berechtigungs-ID eigen, die der bestimmten CREATE-Anweisung zugeordnet wird.

Zum Beispiel erstellt die Anweisung CREATE SCHEMA SCOTTSTUFF AUTHORIZATION SCOTT CREATE TABLE T1 (C! INT) das Schema SCOTTSTUFF und die Tabelle SCOTTSTUFF.T1, deren beider Eigner SCOTT ist. Nehmen Sie an, der Benutzer BOBBY erhält das Zugriffsrecht CREATEIN für das Schema SCOTTSTUFF und erstellt einen Index für die Tabelle SCOTTSTUFF.T1. Da der Index nach dem Schema erstellt wird, ist BOBBY der Eigner des Index für die Tabelle SCOTTSTUFF.T1.

Zugriffsrechte werden dem Objekteigner auf der Grundlage des Typs des erstellten Objekts erteilt:

- Das Zugriffsrecht CONTROL wird für neu erstellte Tabellen, Indizes und Pakete implizit erteilt. Dieses Zugriffsrecht berechtigt den Objektersteller zum Zugriff auf das Datenbankobjekt sowie zum Erteilen und Widerrufen von Zugriffsrechten für andere Benutzer in Bezug auf dieses Objekt. Wenn ein anderer Benutzer das Zugriffsrecht CONTROL für dieses Objekt benötigt, muss ein Benutzer mit der Berechtigung SYSADM oder DBADM das Zugriffsrecht CONTROL für dieses Objekt erteilen. Das Zugriffsrecht CONTROL kann nicht durch den Objekteigner entzogen werden.
- Das Zugriffsrecht CONTROL wird implizit für neu erstellte Sichten erteilt, wenn der Objekteigner über das Zugriffsrecht CONTROL für alle Tabellen, Sichten und Kurznamen verfügt, die in der Sichtdefinition angegeben sind.
- Anderen Objekten wie Auslösern, Routinen, Sequenzen, Tabellenbereichen und Pufferpools ist kein Zugriffsrecht CONTROL zugeordnet. Der Objekteigner erhält jedoch alle einem Objekt zugeordneten Zugriffsrechte automatisch (und kann diese Zugriffsrechte mit Hilfe der Klausel WITH GRANT OPTION der Anweisung GRANT anderen Benutzern erteilen, wo dies unterstützt wird). Darüber hinaus kann der Objekteigner das Objekt ändern und löschen oder ihm einen Kommentar hinzufügen. Diese Berechtigungen gelten für den Objekteigner implizit und können nicht widerrufen werden.

#### Zugehörige Konzepte:

- „Zugriffsrechte, Berechtigungsstufen und Datenbankberechtigungen“ auf Seite 245
- „Zugriffsrechte für Schemata“ auf Seite 257
- „Zugriffsrechte für Tabellenbereiche“ auf Seite 259
- „Zugriffsrechte für Tabellen und Sichten“ auf Seite 259
- „Zugriffsrechte für Pakete“ auf Seite 261
- „Zugriffsrechte für Indizes“ auf Seite 262
- „Zugriffsrechte für Sequenzen“ auf Seite 263
- „Zugriffsrechte für Routinen“ auf Seite 263

---

## Details zu Zugriffsrechten und Berechtigungen

Dieser Abschnitt behandelt die einzelnen Arten von Berechtigungen und Zugriffsrechten.

### Systemverwaltungs Berechtigung (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, Servern, Aliasnamen, Datentypen, Funktionen, Prozeduren, Auslösern, Tabellenbereichen, Datenbankpartitionsgruppen, Pufferpools und Ereignismonitoren.

Die Berechtigung SYSADM wird der Gruppe zugeordnet, die im Konfigurationsparameter *sysadm\_group* angegeben ist. Die Zugehörigkeit zu dieser Gruppe wird außerhalb des Datenbankmanagers über die auf Ihrer Plattform verwendete Sicherheitseinrichtung gesteuert.

Nur ein Benutzer mit der Berechtigung SYSADM kann folgende Funktionen ausführen:

- Migrieren einer Datenbank
- Ändern der Konfigurationsdatei des Datenbankmanagers (einschließlich Angabe der Gruppen, die über die Berechtigung SYSCTRL, SYSMANT oder SYSMON verfügen)
- Erteilen der Berechtigung DBADM

**Anmerkung:** Wenn Benutzer mit der Berechtigung SYSADM eine Datenbank erstellt, wird diesem Benutzer automatisch die explizite Berechtigung DBADM für die Datenbank erteilt. Wenn der Datenbankersteller aus der SYSADM-Gruppe entfernt wird und Sie diesen Benutzer auch daran hindern möchten, mit der Berechtigung DBADM auf die Datenbank zuzugreifen, müssen Sie die Berechtigung DBADM des Benutzers explizit widerrufen.

#### Zugehörige Konzepte:

- „Systemsteuerungsberechtigung (SYSCTRL)“ auf Seite 251
- „Systempflegeberechtigung (SYSMANT)“ auf Seite 252
- „Datenverschlüsselung“ auf Seite 273
- „Systemmonitorberechtigung (SYSMON)“ auf Seite 253

## 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. 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 Systempflegeberechtigung (SYSMANT) und Systemmonitorberechtigung (SYSMON) 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.

**Zugehörige Konzepte:**

- „Systempflegeberechtigung (SYSMAINT)“ auf Seite 252
- „Datenbankadministratorberechtigung (DBADM)“ auf Seite 253
- „Systemmonitorberechtigung (SYSMON)“ auf Seite 253

## 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. 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 verfügen auch über das implizite Zugriffsrecht, eine Verbindung zu einer Datenbank herzustellen, und sie können die Funktionen von Benutzern mit Systemmonitorberechtigung (SYSMON) ausführen.

### Zugehörige Konzepte:

- „Datenbankadministratorberechtigung (DBADM)“ auf Seite 253
- „Systemmonitorberechtigung (SYSMON)“ auf Seite 253

## 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 Datenbankberechtigungen BINDADD, CONNECT, CREATETAB, CREATE\_EXTERNAL\_ROUTINE, CREATE\_NOT\_FENCED\_ROUTINE, IMPLICIT\_SCHEMA, QUIESCE\_CONNECT und LOAD 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.

### Zugehörige Konzepte:

- „Systemverwaltungsberechtigung (SYSADM)“ auf Seite 250
- „Systemsteuerungsberechtigung (SYSCTRL)“ auf Seite 251
- „Systempflegeberechtigung (SYSMAINT)“ auf Seite 252
- „LOAD-Berechtigung“ auf Seite 254
- „Datenbankberechtigungen“ auf Seite 255
- „Überlegungen zur Berechtigung IMPLICIT\_SCHEMA“ auf Seite 257

## Systemmonitorberechtigung (SYSMON)

Die Berechtigung SYSMON ermöglicht die Verwendung des Datenbanksystemmonitors zur Erstellung von Momentaufnahmen (Snapshots) eines Datenbankmanagerexemplars oder seiner Datenbanken. Die Berechtigung SYSMON wird der Gruppe zugeordnet, die im Konfigurationsparameter *sysmon\_group* angegeben ist. Wenn eine Gruppe angegeben wird, wird die Zugehörigkeit zu dieser Gruppe außerhalb des Datenbankmanagers durch die auf Ihrer Plattform verwendete Sicherheitseinrichtung gesteuert.

Die Berechtigung SYSMON gibt dem Benutzer die Möglichkeit, die folgenden Befehle auszuführen:

- GET DATABASE MANAGER MONITOR SWITCHES
- GET MONITOR SWITCHES
- GET SNAPSHOT
- LIST ACTIVE DATABASES
- LIST APPLICATIONS
- LIST DCS APPLICATIONS
- RESET MONITOR
- UPDATE MONITOR SWITCHES

Die Berechtigung SYSMON gibt dem Benutzer die Möglichkeit, die folgenden APIs zu verwenden:

- db2GetSnapshot - Momentaufnahme erstellen
- db2GetSnapshotSize - Größe des erforderlichen Ausgabepuffers für db2GetSnapshot() schätzen
- db2MonitorSwitches - Monitorschalter abrufen/aktualisieren
- db2ResetMonitor - Monitor zurücksetzen

Die Berechtigung SYSMON gibt dem Benutzer die Möglichkeit, die folgenden SQL-Tabellenfunktionen zu verwenden:

- Alle Tabellenfunktionen zur Erstellung von Momentaufnahmen, ohne zuvor SYSPROC.SNAPSHOT\_FILEW auszuführen.  
SYSPROC.SNAPSHOT\_FILEW erfasst eine Momentaufnahme und speichert ihren Inhalt in einer Datei. Wenn eine der Tabellenfunktionen für Momentaufnahmen mit null Eingabeparametern aufgerufen wird, wird der Inhalt der Datei anstelle einer Echtzeitmomentaufnahme des Systems zurückgegeben.

Benutzer mit der Berechtigungsstufe SYSADM, SYSCTRL oder SYSMAINT besitzen auch die Berechtigung SYSMON.

#### Zugehörige Referenzen:

- „sysmon\_group - Berechtigungsgruppenname für Systemmonitor (Konfigurationsparameter)“ in *Systemverwaltung: Optimierung*

## LOAD-Berechtigung

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

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

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

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

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

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

#### Zugehörige Konzepte:

- „Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von LOAD“ in *Dienstprogramme für das Versetzen von Daten Handbuch und Referenz*
- „Zugriffsrechte für Tabellen und Sichten“ auf Seite 259

#### Zugehörige Referenzen:

- „RUNSTATS Command“ in *Command Reference*
- „QUIESCE TABLESPACES FOR TABLE Command“ in *Command Reference*
- „LIST TABLESPACES Command“ in *Command Reference*
- „LOAD Command“ in *Command Reference*

## Datenbankberechtigungen

Abb. 4 zeigt die Datenbankberechtigungen.

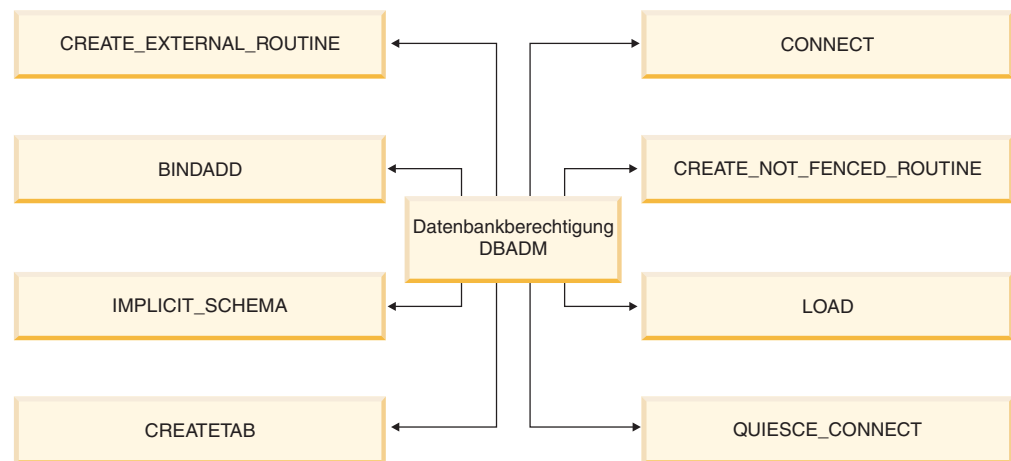


Abbildung 4. Datenbankberechtigungen

Datenbankberechtigungen umfassen Aktionen für eine Datenbank als Ganzes. Jeder Benutzer mit der Berechtigung DBADM besitzt die vollständige Gruppe der folgenden Datenbankberechtigungen:

- **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\_EXTERNAL\_ROUTINE ermöglicht einem Benutzer das Erstellen einer Prozedur zur Verwendung durch Anwendungen und andere Benutzer der Datenbank.
- CREATE\_NOT\_FENCED\_ROUTINE erlaubt einem Benutzer das Erstellen einer benutzerdefinierten Funktion (UDF) oder Prozedur, die nicht abgeschirmt („Not Fenced“) ist. Benutzerdefinierte Funktionen oder Prozeduren, die „nicht abgeschirmt“ 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 abgeschirmt“ ausgeführt wird, schwerwiegende Probleme für Ihr System verursachen.)

**Anmerkung:** CREATE\_EXTERNAL\_ROUTINE wird jedem Benutzer automatisch erteilt, dem CREATE\_NOT\_FENCED\_ROUTINE erteilt wird.

- IMPLICIT\_SCHEMA erlaubt jedem Benutzer das implizite Erstellen eines Schemas, indem er ein Objekt mit der Anweisung CREATE und einem Schemanamen, 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.
- QUIESCE\_CONNECT ermöglicht einem Benutzer den Zugriff auf die Datenbank, während sie im Wartemodus ist.

Nur Benutzer mit der Berechtigung SYSADM oder DBADM können diese Datenbankberechtigungen anderen Benutzern erteilen und entziehen.

**Anmerkung:** Wenn eine Datenbank erstellt wird, werden die folgenden Datenbankberechtigungen automatisch der Gruppe PUBLIC erteilt:

- Datenbankberechtigung CREATETAB
- Datenbankberechtigung BINDADD
- Datenbankberechtigung CONNECT
- Datenbankberechtigung IMPLICIT\_SCHEMA
- Zugriffsrecht USE für den Tabellenbereich USERSPACE1
- Zugriffsrecht SELECT für die Systemkatalogsichten

Zum Widerrufen dieser Datenbankberechtigungen muss ein Benutzer mit der Berechtigung DBADM oder SYSADM jede Datenbankberechtigung der Gruppe PUBLIC explizit entziehen.

#### **Zugehörige Tasks:**

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265



## Überlegungen zur Berechtigung IMPLICIT\_SCHEMA

Wenn eine neue Datenbank erstellt wird, erhält die Gruppe PUBLIC die Datenbankberechtigung IMPLICIT\_SCHEMA. Mit dieser Berechtigung kann jeder Benutzer ein Schema erstellen, indem er ein Objekt erstellt und dabei einen Schemanamen 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 Schemaobjekte 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 optional 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.

### Zugehörige Tasks:

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265

## Zugriffsrechte für Schemata

Zugriffsrechte für Schemata gehören zur Kategorie der Zugriffsrechte für Objekte. Die Zugriffsrechte für Objekte sind in Abb. 5 auf Seite 258 dargestellt.

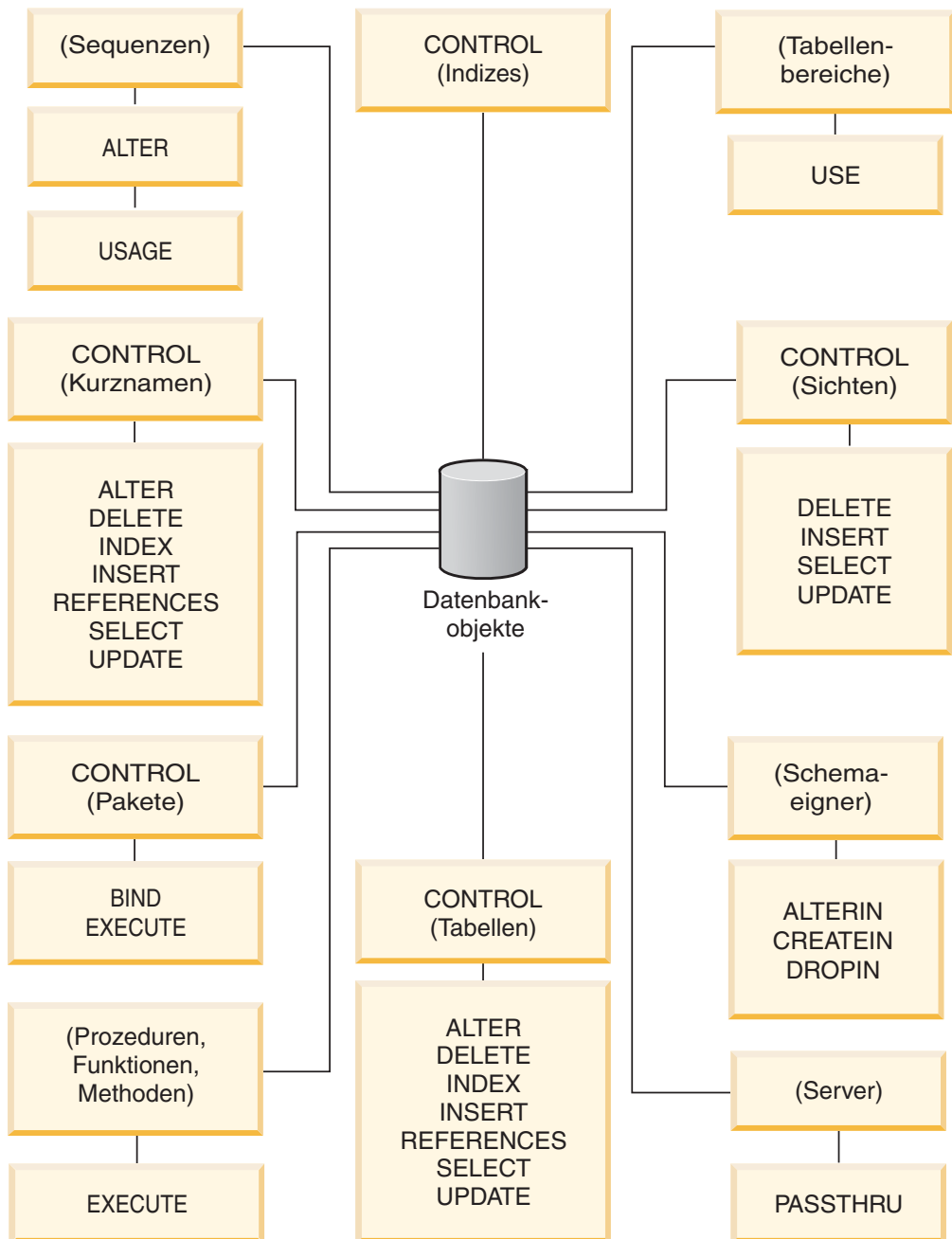


Abbildung 5. Zugriffsrechte für Objekte

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.

**Zugehörige Tasks:**

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265

**Zugehörige Referenzen:**

- „ALTER SEQUENCE statement“ in *SQL Reference, Volume 2*

## Zugriffsrechte für Tabellenbereiche

Mit den Zugriffsrechten für 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 USER-SPACE1 der Gruppe PUBLIC erteilt. Dieses Zugriffsrecht kann jedoch auch widerrufen werden.

Das Zugriffsrecht USE kann bei SYSCATSPACE oder anderen temporären Systemtabellenbereichen nicht verwendet werden.

**Zugehörige Tasks:**

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265

**Zugehörige Referenzen:**

- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

## Zugriffsrechte für Tabellen und Sichten

Zugriffsrechte für Tabellen und Sichten umfassen Aktionen für Tabellen oder Sichten in einer Datenbank. Ein Benutzer muss über die Berechtigung 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, Sichten und Kurznamen verfügt, auf die in der Sichtdefinition verwiesen wird, oder wenn er über die Berechtigung SYSADM oder DBADM verfügt.
- Das Zugriffsrecht ALTER ermöglicht dem Benutzer, eine Tabelle zu modifizieren, zum Beispiel, der Tabelle Spalten oder eine eindeutige Integritätsbedingung hinzuzufügen. Ein Benutzer mit dem Zugriffsrecht ALTER kann außerdem die Anweisung COMMENT ON für eine Tabelle bzw. für Spalten der Tabelle angeben. Informationen zu den möglichen Modifikationen, die an einer Tabelle vorgenommen werden können, finden Sie in den Beschreibungen der Anweisungen ALTER TABLE und COMMENT.

- 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.
- 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 ihre untergeordneten Tabellen (sofern vorhanden) beeinflussen. Dieses Verhalten wird als *Substitutionsfähigkeit* bezeichnet. Nehmen Sie zum Beispiel an, Sie haben eine Arbeitertabelle des Typs Employee\_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 Employee\_t und Manager\_t und durch die entsprechende Abhängigkeit von

Tabelle und untergeordneter Tabelle für die Tabellen Employee und Manager angegeben wird. Die SQL-Abfrage `SELECT * FROM Employee` gibt daher als Ergebnis dieser Abhängigkeit die Objektkennung und die Employee\_t-Attribute für Mitarbeiter und Manager zurück. Gleichmaßen definiert die Aktualisierungsoperation `UPDATE Employee SET Salary = Salary + 1000` eine Gehaltserhöhung von 1000 ? sowohl für Manager als auch für reguläre Mitarbeiter.

Ein Benutzer mit dem Zugriffsrecht `SELECT` für die Tabelle Employee 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. Daher kann er nicht auf die nicht übernommenen Spalten der Tabelle Manager zugreifen.

Gleichmaßen kann ein Benutzer mit dem Zugriffsrecht `UPDATE` für die Tabelle Employee eine `UPDATE`-Operation für die Tabelle Manager ausführen, was sich auf gewöhnliche Mitarbeiter und Manager auswirkt, 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.

#### **Zugehörige Konzepte:**

- „Zugriffsrechte für Indizes“ auf Seite 262

#### **Zugehörige Tasks:**

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265

#### **Zugehörige Referenzen:**

- „ALTER TABLE statement“ in *SQL Reference, Volume 2*
- „CREATE VIEW statement“ in *SQL Reference, Volume 2*
- „SELECT statement“ in *SQL Reference, Volume 2*

## **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 die Berechtigung `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 auch anderen Benutzern diese Zugriffsrechte mit Hilfe der Anweisung `GRANT` erteilen. (Wenn ein Zugriffsrecht mit der Klausel `WITH GRANT OPTION` erteilt wird, kann ein Benutzer, der das Zugriffsrecht `BIND` oder `EXECUTE` erhält, seinerseits dieses Zugriffsrecht anderen Benutzern erteilen.) Um das Zugriffsrecht `CONTROL` erteilen zu können, muss der Benutzer über die Berechtigung `SYSADM` oder `DBADM` verfügen.

- Das Zugriffsrecht BIND für ein Paket ermöglicht dem Benutzer, dieses Paket zu binden oder erneut zu binden sowie neue Paketversionen des gleichen Paketnamens mit dem gleichen Ersteller hinzuzufügen.
- EXECUTE ermöglicht dem Benutzer, ein Paket auszuführen.

**Anmerkung:** Alle Paketzugriffsrechte gelten für sämtliche VERSIONEN, die den gleichen Paketnamen und den gleichen Ersteller haben.

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.

Objekte, auf die über Kurznamen verwiesen wird, müssen Authentifizierungsprüfungen an den Datenquellen durchlaufen, die die Objekte enthalten. 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® Universal Database (DB2 UDB) 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.

**Zugehörige Konzepte:**

- „Datenbankberechtigungen“ auf Seite 255

**Zugehörige Tasks:**

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265

## 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 effektiv 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 ermöglicht einem Benutzer das Erstellen eines Index für diese Tabelle.

Das Zugriffsrecht INDEX auf Kurznamenebene ermöglicht einem Benutzer das Erstellen eines Index für diesen Kurznamen.

**Zugehörige Konzepte:**

- „Zugriffsrechte für Tabellen und Sichten“ auf Seite 259

**Zugehörige Tasks:**

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265

## Zugriffsrechte für Sequenzen

Der Ersteller einer Sequenz erhält automatisch die Zugriffsrechte USAGE und ALTER für die Sequenz. Das Zugriffsrecht USAGE ist zur Verwendung der Ausdrücke NEXT VALUE und PREVIOUS VALUE für die Sequenz erforderlich. Um anderen Benutzern die Verwendung der Ausdrücke NEXT VALUE und PREVIOUS VALUE zu erlauben, müssen die Sequenzzugriffsrechte der Gruppe PUBLIC erteilt werden. Dadurch erhalten alle Benutzer die Erlaubnis die Ausdrücke mit der angegebenen Sequenz zu verwenden.

Das Zugriffsrecht ALTER für die Sequenz erlaubt dem Benutzer solche Tasks wie das erneute Starten der Sequenz oder das Ändern des Inkrements für zukünftige Sequenzwerte auszuführen. Der Ersteller der Sequenz kann das Zugriffsrecht ALTER anderen Benutzern erteilen. Wenn dazu die Klausel WITH GRANT OPTION verwendet wird, können diese Benutzer selbst diese Zugriffsrechte wieder anderen Benutzern erteilen.

### Zugehörige Tasks:

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265

### Zugehörige Referenzen:

- „ALTER SEQUENCE statement“ in *SQL Reference, Volume 2*

## Zugriffsrechte für Routinen

Zugriffsrechte zum Ausführen (EXECUTE) beinhalten Aktionen an allen Arten von Routinen, wie zum Beispiel Funktionen, Prozeduren und Methoden, innerhalb einer Datenbank. Wenn ein Benutzer das Zugriffsrecht EXECUTE besitzt, kann er die entsprechende Routine aufrufen, eine Quellenfunktion erstellen, die von dieser Routine abgeleitet ist (gilt nur für Funktionen), und auf die Routine in einer beliebigen DDL-Anweisung wie CREATE VIEW oder CREATE TRIGGER verweisen.

Der Benutzer, der die extern gespeicherte Prozedur, Funktion oder Methode definiert, erhält das Zugriffsrecht EXECUTE WITH GRANT. Wenn das Zugriffsrecht EXECUTE einem anderen Benutzer über die Klausel WITH GRANT OPTION erteilt wird, kann dieser Benutzer selbst das Zugriffsrecht EXECUTE wieder einem anderen Benutzer erteilen.

### Zugehörige Tasks:

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265

---

## 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.

Berechtigungen werden auf drei Arten gesteuert:

- Die explizite Berechtigung wird über Zugriffsrechte gesteuert, deren Steuerung mit den Anweisungen GRANT und REVOKE erfolgt.
- Die implizite Berechtigung wird durch Erstellen und Löschen von Objekten gesteuert.
- Indirekte Zugriffsrechte sind Paketen zugeordnet.

**Anmerkung:** Ein Datenbankgruppenname darf höchstens acht Zeichen lang sein, wenn er in einer Anweisung GRANT oder REVOKE bzw. in der Steuerzentrale verwendet wird. Auch wenn ein Datenbankgruppenname akzeptiert wird, der länger als acht Zeichen ist, führt der längere Name zu einer Fehlermeldung, wenn Benutzer, die zu der Gruppe gehören, auf Datenbankobjekte zugreifen.

**Zugehörige Konzepte:**

- „Verwenden des Systemkatalogs zu Sicherheitszwecken“ auf Seite 276

**Zugehörige Tasks:**

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265

---

## Details zur Steuerung des Zugriffs auf Datenbankobjekte

Die Zugriffssteuerung für Datenbankobjekte erfolgt durch die Verwendung der Anweisungen GRANT und REVOKE. Darüber hinaus behandelt dieser Abschnitt auch die implizite Zugriffserteilung und indirekte Zugriffsrechte.

### Erteilen von Zugriffsrechten

**Einschränkungen:**

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 über die 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.

**Vorgehensweise:**

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 optionalen 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
```

#### **Zugehörige Konzepte:**

- „Steuern des Zugriffs auf Datenbankobjekte“ auf Seite 263

#### **Zugehörige Tasks:**

- „Widerrufen von Zugriffsrechten“ auf Seite 265

#### **Zugehörige Referenzen:**

- „GRANT (Database Authorities) statement“ in *SQL Reference, Volume 2*
- „GRANT (Index Privileges) statement“ in *SQL Reference, Volume 2*
- „GRANT (Package Privileges) statement“ in *SQL Reference, Volume 2*
- „GRANT (Schema Privileges) statement“ in *SQL Reference, Volume 2*
- „GRANT (Table, View, or Nickname Privileges) statement“ in *SQL Reference, Volume 2*
- „GRANT (Server Privileges) statement“ in *SQL Reference, Volume 2*
- „GRANT (Table Space Privileges) statement“ in *SQL Reference, Volume 2*
- „GRANT (Sequence Privileges) statement“ in *SQL Reference, Volume 2*
- „GRANT (Routine Privileges) statement“ in *SQL Reference, Volume 2*

## **Widerrufen von Zugriffsrechten**

Die Anweisung REVOKE erlaubt es berechtigten Benutzern, Zugriffsrechte zu widerrufen, die anderen Benutzern zuvor erteilt wurden.

#### **Einschränkungen:**

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 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 Zugriffs-  
| rechte widerrufen werden, erteilt wurden.

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.

#### **Vorgehensweise:**

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).

**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.

Wenn Sie einen Auslöser oder eine SQL-Funktion in Abhängigkeit von einem oder mehreren Zugriffsrechten definieren und Sie eines oder mehrere dieser Zugriffsrechte verlieren, kann der Auslöser bzw. die SQL-Funktion nicht verwendet werden.

#### **Zugehörige Tasks:**

- „Erteilen von Zugriffsrechten“ auf Seite 264

#### **Zugehörige Referenzen:**

- „REVOKE (Database Authorities) statement“ in *SQL Reference, Volume 2*
- „REVOKE (Index Privileges) statement“ in *SQL Reference, Volume 2*
- „REVOKE (Package Privileges) statement“ in *SQL Reference, Volume 2*
- „REVOKE (Schema Privileges) statement“ in *SQL Reference, Volume 2*
- „REVOKE (Table, View, or Nickname Privileges) statement“ in *SQL Reference, Volume 2*
- „REVOKE (Server Privileges) statement“ in *SQL Reference, Volume 2*
- „REVOKE (Table Space Privileges) statement“ in *SQL Reference, Volume 2*
- „REVOKE (Routine Privileges) statement“ in *SQL Reference, Volume 2*

## **Verwalten impliziter Berechtigungen durch Erstellen und Löschen von Objekten**

#### **Vorgehensweise:**

Der Datenbankmanager erteilt einem Benutzer, der ein Datenbankobjekt, wie zum Beispiel eine Tabelle oder ein Paket, erstellt, implizit bestimmte Zugriffsrechte. 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 eine Tabelle, ein Kurzname, 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, Sichten und Kurznamen verfügt, auf die in der Sichtdefinition verwiesen wird.

Wenn das explizit erstellte Objekt ein Schema ist, erhält der Schemaeigner 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.

#### Zugehörige Tasks:

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265

## Einrichten eines Eigentumsrechts für ein Paket

#### Vorgehensweise:

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 Universal Database™-Datenbankprodukten (DB2 UDB) ein Paket binden können, unterstützen die Option OWNER.

#### Zugehörige Referenzen:

- „BIND Command“ in *Command Reference*
- „PRECOMPILE Command“ in *Command Reference*

## Indirekte Zugriffsrechte 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-An-

weisungen 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.

Wenn das Paket dynamisches SQL enthält, hängen die erforderlichen Zugriffsrechte von dem Wert ab, der beim Vorkompilieren (PRECOMPILE) oder Binden (BIND) des Pakets für DYNAMICRULES angegeben wurde. Weitere Informationen finden Sie im Abschnitt zur Wirkung von DYNAMICRULES auf dynamisches SQL.

#### Zugehörige Konzepte:

- „Indirekte Zugriffsrechte durch ein Paket mit Kurznamen“ auf Seite 269
- „Effect of DYNAMICRULES bind option on dynamic SQL“ in *Application Development Guide: Programming Client Applications*

#### Zugehörige Referenzen:

- „BIND Command“ in *Command Reference*

## Indirekte Zugriffsrechte 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 .SQC-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 und den Kurznamen zu überprüfen. Allerdings findet für die Datenquellenobjekte, die durch den Kurznamen angegeben werden, 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 .SQC-Datei nur dynamische SQL-Anweisungen und eine Kombination von Tabellen- und Kurznamenverweisen enthält, ist die Berechtigungsprüfung durch DB2<sup>®</sup> Universal Database (DB2 UDB) für lokale Objekte und Kurznamen ähnlich. Paketbenutzer müssen die Überprüfung der Zugriffsrechte für beliebige lokale Objekte (Tabellen, Sichten), die sich in der Anweisung befinden, durchlaufen. Außerdem werden die Zugriffsrechte für Kurznamenobjekte überprüft. (Paketbenutzer müssen eine Authentifizierung und eine Überprüfung der Zugriffsrechte

| an der Datenquelle durchlaufen, die die durch die Kurznamen angegebenen  
| Objekte 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:** In statischem SQL können Kurznamen nicht angegeben werden. Verwenden Sie mit Paketen, die Kurznamen enthalten, nicht die Option DYNAMICRULES (auf BIND gesetzt).

Für Pakete mit Kurznamen sind möglicherweise weitere Berechtigungsschritte erforderlich, da DB2 UDB 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.

#### **Zugehörige Konzepte:**

- „Indirekte Zugriffsrechte durch ein Paket“ auf Seite 268

## **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 beschränken, auf die über eine Sicht zugegriffen wird.
- 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 Sichten in DB2<sup>®</sup> Universal Database (DB2 UDB) 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, jede Sicht oder jeden Kurznamen verfügen, auf die in der Sichtdefinition  
| verwiesen wird. 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.

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 der Sicht die Berechtigung SELECT oder eine gleichwertige Berechtigungsstufe für die zugrunde liegenden Datenquellenobjekte haben, wenn sie auf die Sicht zugreifen.

Wenn die Benutzer an der Datenquelle nicht die entsprechende Berechtigung für die zugrunde liegenden 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. Zum 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
Lundquist	Dallas
Wheeler	Dallas
Lea	San Francisco
Wilson	San Francisco
Graham	San Francisco
Gonzales	San Francisco
Burke	San Francisco
Quill	Denver



NAME	LOCATION
Davis	Denver
Edwards	Denver
Gafney	Denver

#### Zugehörige Tasks:

- „Erstellen einer Sicht“ auf Seite 133
- „Erteilen von Zugriffsrechten“ auf Seite 264

## Überwachen das Datenzugriffs mit der Prüffunktion

Die Prüffunktion von DB2® Universal Database (DB2 UDB) generiert ein Prüfprotokoll für eine Reihe vordefinierter Datenbankereignisse und ermöglicht Ihnen die Verwaltung des Protokolls. Wenngleich es sich um keine Einrichtung handelt, mit der Datenzugriffe verhindert werden können, lassen sich mit der Prüffunktion Versuche, auf Datenobjekte zuzugreifen bzw. diese zu ändern, überwachen und aufzeichnen.

Zur Verwendung der Prüffunktion **db2audit** ist die Berechtigung SYSADM erforderlich.

#### Zugehörige Konzepte:

- „Einführung in die Prüffunktion von DB2 Universal Database (DB2 UDB)“ auf Seite 283

## 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.

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 ENCRYPT-Funktionen sind Daten des Typs VARCHAR FOR BIT DATA (mit einer Begrenzung von 32.631).

Nur Daten der Typen CHAR, VARCHAR und FOR BIT DATA können verschlüsselt werden.

Mit den Funktionen DECRYPT\_BIN und DECRYPT\_CHAR werden Daten mit der Entschlüsselung auf Kennwortbasis entschlüsselt.

DECRYPT\_BIN gibt immer Daten des Typs VARCHAR FOR BIT DATA zurück, während DECRYPT\_CHAR immer Daten des Typs VARCHAR zurückgibt. Da das

erste Argument CHAR FOR BIT DATA oder VARCHAR FOR BIT DATA sein kann, gibt es Fälle, in denen das Ergebnis nicht mit dem ersten Argument übereinstimmt.

Die Länge des Ergebnisses hängt von den Byte bis zur nächsten 8-Byte-Grenze ab. Die Länge des Ergebnisses könnte die Länge des Datenarguments plus 40 plus die Anzahl von Byte bis zur nächsten 8-Byte-Grenze sein, wenn der optionale Hinweisparameter angegeben wird. Oder die Länge des Ergebnisses könnte die Länge des Datenarguments plus 8 plus die Anzahl von Byte bis zur nächsten 8-Byte-Grenze sein, wenn der optionale Hinweisparameter nicht angegeben wird.

Mit der Funktion GETHINT wird ein eingebundener Kennworthinweis wieder zurückgegeben. Ein Kennworthinweis ist ein Ausdruck, mit dessen Hilfe die Eigentümer 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.
- Sonderregister für Verschlüsselungskennwort. 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. Das Sonderregister ENCRYPTION PASSWORD wird nur in verschlüsselter Form gespeichert. Der Anfangs- oder Standardwert für das Sonderregister ist eine leere Zeichenfolge.

Gültige Längen für Kennwörter liegen zwischen 6 und 127 einschließlich dieser Längen selbst. Gültige Längen für Hinweise liegen zwischen 0 und 32 einschließlich.

#### Zugehörige Referenzen:

- „SET ENCRYPTION PASSWORD statement“ in *SQL Reference, Volume 2*
- „DECRYPT\_BIN and DECRYPT\_CHAR scalar functions“ in *SQL Reference, Volume 1*
- „ENCRYPT scalar function“ in *SQL Reference, Volume 1*
- „GETHINT scalar function“ in *SQL Reference, Volume 1*

---

## Aufgaben und erforderliche Berechtigungen

In den verschiedenen Unternehmen und Behörden sind die einzelnen Aufgabengebiete nicht immer in gleicher Weise verteilt. In Tabelle 6 auf Seite 275 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 6. 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 SYSMANT für eine oder mehrere Datenbanken. In einigen Fällen Berechtigung SYSCTRL.
Systembediener	Überwacht die Datenbank und führt Sicherungen aus.	Berechtigung SYSMANT
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.  CREATE_EXTERNAL_ROUTINE kann ebenfalls erforderlich sein.
Benutzeranalytiker	Definiert die Datenerfordernisse für ein Anwendungsprogramm durch Überprüfen der Systemkatalogsichten.	SELECT für die Katalogsichten; CONNECT für eine oder mehrere Datenbanken
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.

**Zugehörige Konzepte:**

- „Systemverwaltungsberechtigung (SYSADM)“ auf Seite 250
- „Systemsteuerungsberechtigung (SYSCTRL)“ auf Seite 251
- „Systempflegeberechtigung (SYSMAINT)“ auf Seite 252

- „Datenbankadministratorberechtigung (DBADM)“ auf Seite 253
- „LOAD-Berechtigung“ auf Seite 254
- „Datenbankberechtigungen“ auf Seite 255

**Zugehörige Tasks:**

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265

---

## Verwenden des Systemkatalogs zu Sicherheitszwecken

Informationen über alle Datenbanken werden automatisch in einer Gruppe von Tabellen, dem so genannten Systemkatalog, gepflegt, die beim Generieren der Datenbank erstellt werden. Dieser Systemkatalog beschreibt Tabellen, Spalten, Indizes, Programme, Zugriffsrechte und andere Objekte.

In diesen Sichten werden die Zugriffsrechte von Benutzern und die Identität des Benutzers, der das Zugriffsrecht jeweils erteilt, aufgelistet:

<b>SYSCAT.DBAUTH</b>	Listet die Zugriffsrechte für Datenbanken auf.
<b>SYSCAT.TABAUTH</b>	Listet die Zugriffsrechte für Tabellen und Sichten auf.
<b>SYSCAT.COLAUTH</b>	Listet die Zugriffsrechte für Spalten auf.
<b>SYSCAT.PACKAGEAUTH</b>	Listet die Zugriffsrechte für Pakete auf.
<b>SYSCAT.INDEXAUTH</b>	Listet die Zugriffsrechte für Indizes auf.
<b>SYSCAT.SCHEMAAUTH</b>	Listet die Zugriffsrechte für Schemata auf.
<b>SYSCAT.PASSTHROUGHAUTH</b>	Listet das Zugriffsrecht für Server auf.
<b>SYSCAT.ROUTINEAUTH</b>	Listet die Zugriffsrechte für Routinen (Funktionen, Methoden und gespeicherte Prozeduren) auf.

Zugriffsrechte, die den Benutzern vom System verliehen wurden, weisen in der Spalte GRANTOR den Wert SYSIBM auf. SYSADM, SYSMAINT und SYSCTRL 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.

**Zugehörige Tasks:**

- „Abrufen von Berechtigungsnamen mit erteilten Zugriffsrechten“ auf Seite 277
- „Abrufen aller Namen mit der Berechtigung DBADM“ auf Seite 277
- „Abrufen der Namen mit Zugriffsberechtigung für eine Tabelle“ auf Seite 278
- „Abrufen aller Benutzern erteilter Zugriffsrechte“ auf Seite 278
- „Schützen der Systemkatalogsicht“ auf Seite 279

**Zugehörige Referenzen:**

- „SYSCAT.COLAUTH catalog view“ in *SQL Reference, Volume 1*
- „SYSCAT.DBAUTH catalog view“ in *SQL Reference, Volume 1*
- „SYSCAT.INDEXAUTH catalog view“ in *SQL Reference, Volume 1*

- „SYSCAT.PACKAGEAUTH catalog view“ in *SQL Reference, Volume 1*
- „SYSCAT.SCHEMAAUTH catalog view“ in *SQL Reference, Volume 1*
- „SYSCAT.TABAUTH catalog view“ in *SQL Reference, Volume 1*
- „SYSCAT.PASSTHROUGH AUTH catalog view“ in *SQL Reference, Volume 1*
- „SYSCAT.ROUTINEAUTH catalog view“ in *SQL Reference, Volume 1*

---

## Details zur Verwendung des Systemkatalogs in Bezug auf die Sicherheit

Dieser Abschnitt behandelt einige Möglichkeiten zur Ermittlung von Personen und ihrer Zugriffsrechte innerhalb der Datenbank.

### Abrufen von Berechtigungsnamen mit erteilten Zugriffsrechten

#### Vorgehensweise:

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.PASSTHROUGH AUTH
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 Datenbankclients unterstützen, ist der Berechtigungsname möglicherweise nur auf dem fernen Client und nicht auf Ihrem Datenbankserversystem definiert.

#### Zugehörige Konzepte:

- „Verwenden des Systemkatalogs zu Sicherheitszwecken“ auf Seite 276

### Abrufen aller Namen mit der Berechtigung DBADM

#### Vorgehensweise:

Die folgende Anweisung ruft alle Berechtigungsnamen ab, denen die Berechtigung DBADM direkt erteilt wurde:

```
SELECT DISTINCT GRANTEE FROM SYSCAT.DBAUTH
WHERE DBADMAUTH = 'Y'
```

### Zugehörige Konzepte:

- „Datenbankadministratorberechtigung (DBADM)“ auf Seite 253
- „Verwenden des Systemkatalogs zu Sicherheitszwecken“ auf Seite 276

## Abrufen der Namen mit Zugriffsberechtigung für eine Tabelle

### Vorgehensweise:

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'
```

Mit Hilfe der folgenden Anweisung können Sie herausfinden, wer die Tabelle EMPLOYEE mit dem Qualifikationsmerkmal JAMES aktualisieren kann:

```
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'
```

Diese Anweisung ruft alle Berechtigungsnamen mit der Berechtigung DBADM sowie die Namen ab, denen die Zugriffsrechte CONTROL oder UPDATE direkt erteilt wurden. 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.

### Zugehörige Konzepte:

- „Zugriffsrechte für Tabellen und Sichten“ auf Seite 259
- „Verwenden des Systemkatalogs zu Sicherheitszwecken“ auf Seite 276

## Abrufen aller Benutzern erteilter Zugriffsrechte

### Vorgehensweise:

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. Zum Beispiel 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 für Lesezugriff.

#### **Zugehörige Konzepte:**

- „Zugriffsrechte, Berechtigungsstufen und Datenbankberechtigungen“ auf Seite 245
- „Datenbankberechtigungen“ auf Seite 255
- „Verwenden des Systemkatalogs zu Sicherheitszwecken“ auf Seite 276

#### **Zugehörige Tasks:**

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265

## **Schützen der Systemkatalogsicht**

#### **Vorgehensweise:**

Während der Datenbankeinstellung wird das Zugriffsrecht SELECT für die Systemkatalogsichten der Gruppe PUBLIC erteilt. 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, jedoch benötigen Sie 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 wurden. 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 Basistabelle zu widerrufen:

```
REVOKE SELECT ON TABLE SYSCAT.TABAUTH FROM PUBLIC
```

#### **Zugehörige Konzepte:**

- „Katalogstatistiken“ in *Systemverwaltung: Optimierung*
- „Datenbankberechtigungen“ auf Seite 255
- „Verwenden des Systemkatalogs zu Sicherheitszwecken“ auf Seite 276

#### **Zugehörige Tasks:**

- „Erteilen von Zugriffsrechten“ auf Seite 264
- „Widerrufen von Zugriffsrechten“ auf Seite 265

---

## **Einführung in die Firewallunterstützung**

Eine *Firewall* ist eine Gruppe zusammengehöriger Programme, die sich auf einem Gateway-Netzwerkserver befinden und dazu dienen, unbefugten Zugriff auf ein System oder ein Netzwerk zu verhindern.

Es gibt vier Arten von Firewalls:

1. Paketfilterungsfirewalls, Screening Router Firewalls oder Firewalls der Netzwerkebene
2. Klassische Proxy-Firewalls der Anwendungsebene
3. Transparente Proxy- oder Circuit-Level-Firewalls
4. SMLI-Firewalls (Stateful Multi-Layer Inspection)



Es sind Firewallprodukte verfügbar, die eine der oben genannten Firewallarten implementieren. Viele andere Firewallprodukte implementieren jedoch eine Kombination aus diesen Arten.

**Zugehörige Konzepte:**

- „Screening-Router-Firewalls“ auf Seite 281
- „Proxy-Firewalls der Anwendungsebene“ auf Seite 281
- „Circuit-Level-Firewalls“ auf Seite 282
- „SMLI-Firewalls (Stateful Multi-Layer Inspection)“ auf Seite 282

---

## Screening-Router-Firewalls

Dieser Typ von Firewall wird auch als Netzwerk- oder Paketfilterungsfirewall bezeichnet. Die Funktionsweise einer solchen Firewall beruht auf der Überwachung (Screening) ankommender Datenpakete und der Prüfung von Protokollattributen. Zu den getesteten Protokollattributen gehören die Quellen- und Zieladresse, der Typ des Protokolls, der Quellen- und Zielpport sowie einige andere protokollspezifische Attribute.

Für alle Firewall-Lösungen (außer SOCKS) müssen Sie sicherstellen, dass alle von DB2<sup>®</sup> Universal Database (DB2 UDB) verwendeten Ports für ankommende und abgehende Pakete geöffnet sind. DB2 UDB verwendet den Port 523 für den DB2-Verwaltungsserver (DAS), der von den DB2 UDB-Tools verwendet wird. Bestimmen Sie die Ports, die von allen Ihren Serverexemplaren verwendet werden, indem Sie mit Hilfe der Datei 'services' den Servicenamen in der Konfigurationsdatei des Datenbankmanagers auf dem Server mit seiner Portnummer abgleichen.

**Zugehörige Konzepte:**

- „Einführung in die Firewallunterstützung“ auf Seite 280

---

## Proxy-Firewalls der Anwendungsebene

Ein Proxy bzw. ein Proxy-Server ist eine Technik, die als Vermittlungsstelle zwischen einem Webclient und einem Webserver fungiert. Eine Proxy-Firewall fungiert als Gateway für Anforderungen, die von Clients ankommen. Wenn Clientanforderungen durch die Firewall empfangen werden, wird die endgültige Zieladresse durch die Proxysoftware ermittelt. Der Anwendungsproxy übersetzt die Adresse, führt weitere Prüfungen zur Zugriffssteuerung und Protokollfunktionen nach Bedarf aus und stellt die Verbindung zum Server für den Client her.

Das Produkt DB2<sup>®</sup> Connect auf einer Firewallmaschine kann als Proxy zum Zielserver eingesetzt werden. Darüber hinaus funktioniert ein DB2 Universal Database<sup>™</sup>-Server (DB2 UDB-Server) auf der Firewallmaschine, der als Hop-Server eingesetzt wird, wie ein Anwendungsproxy.

**Zugehörige Konzepte:**

- „Einführung in die Firewallunterstützung“ auf Seite 280

---

## Circuit-Level-Firewalls

Dieser Typ von Firewall wird auch als transparente Proxy-Firewall bezeichnet. Eine transparente Proxy-Firewall modifiziert die Anforderung bzw. die Antwort nicht über das hinaus, was für die Proxy-Authentifizierung und -Identifikation erforderlich ist. Ein Beispiel für eine transparente Proxy-Firewall ist SOCKS.

DB2<sup>®</sup> Universal Database (DB2 UDB) unterstützt SOCKS Version 4.

### Zugehörige Konzepte:

- „Einführung in die Firewallunterstützung“ auf Seite 280

---

## SMLI-Firewalls (Stateful Multi-Layer Inspection)

Dieser Typ von Firewall bietet eine fortgeschrittene Form der Paketfilterung, bei der alle sieben Schichten des OSI-Modells (Open System Interconnection) untersucht werden. Jedes Paket wird untersucht und mit bekannten Status freundlicher Pakete verglichen. Während Screening-Router-Firewalls nur den Paketheader untersuchen, prüfen SMLI-Firewalls das gesamte Paket einschließlich der Daten.

### Zugehörige Konzepte:

- „Einführung in die Firewallunterstützung“ auf Seite 280

---

## Kapitel 8. Prüfen von Aktivitäten in DB2 Universal Database™™ (DB2 UDB)

---

### Einführung in die Prüffunktion von DB2 Universal Database (DB2 UDB)

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® Universal Database (DB2 UDB) 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 UDB-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 Administratortools 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 UDB-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:

- Starten der Aufzeichnung von Prüfeignissen im DB2 UDB-Exemplar.
- Stoppen der Aufzeichnung von Prüfeignissen im DB2 UDB-Exemplar.
- Konfigurieren der Arbeitsweise der Prüffunktion, einschließlich der Auswahl von Kategorien der aufzuzeichnenden Prüfeignisse.
- Anfordern einer Beschreibung der aktuellen Prüfkongfiguration.
- 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.

**Anmerkung:** Stellen Sie sicher, dass die Prüffunktion mit Hilfe des Befehls `DB2AUDIT START` aktiviert wird, bevor Sie die AUDIT-Dienstprogramme verwenden.

Verschiedene Arten von Prüfsätzen können generiert werden. Beachten Sie in der Beschreibung der Kategorien möglicher Prüfeignisse (weiter unten), dass auf jeden Kategorienamen ein aus einem Wort bestehendes Schlüsselwort folgt, das die Art der Kategorie identifiziert. Folgende Kategorien von Prüfeignissen 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 UDB-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 Systemsicherheitsdaten.
- 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 des ausgeführten Operationstyps (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.

#### Zugehörige Konzepte:

- „Funktionsweise der Prüffunktion“ auf Seite 285
- „Aufbau der Datensätze der Prüffunktion (Einführung)“ auf Seite 300
- „Tipps und Techniken für die Prüffunktion“ auf Seite 319

#### Zugehörige Tasks:

- „Steuern der Aktivitäten der DB2 UDB-Prüffunktion“ auf Seite 321

#### Zugehörige Referenzen:

- „Verwendung der Prüffunktion“ auf Seite 287
- „Nachrichten der Prüffunktion“ auf Seite 300

---

## Funktionsweise der Prüffunktion

Die Prüffunktion zeichnet überprüfbare Ereignisse auf, einschließlich solcher, die sich auf Datenbankexemplare auswirken. Aus diesem Grund ist die Prüffunktion eine unabhängige Komponente von DB2<sup>®</sup> Universal Database (DB2 UDB), die auch ausgeführt werden kann, wenn das DB2 UDB-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 UDB 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 (FLUSH) 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 UDB 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 UDB-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.

Je nach der API bzw. der SQL-Anweisung und den Prüfeinstellungen des DB2 UDB-Exemplars können für ein bestimmtes Ereignis kein Prüfsatz, ein Prüfsatz 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 denselben 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 ist die aufgezeichnete Abschnittsnummer für alle Ereignisse (ausgenommen Kontextereignisse) im Prüfsatz null (0), und zwar unabhängig von der tatsächlichen Abschnittsnummer der Anweisung.

**Zugehörige Konzepte:**

- „Einführung in die Prüffunktion von DB2 Universal Database (DB2 UDB)“ auf Seite 283

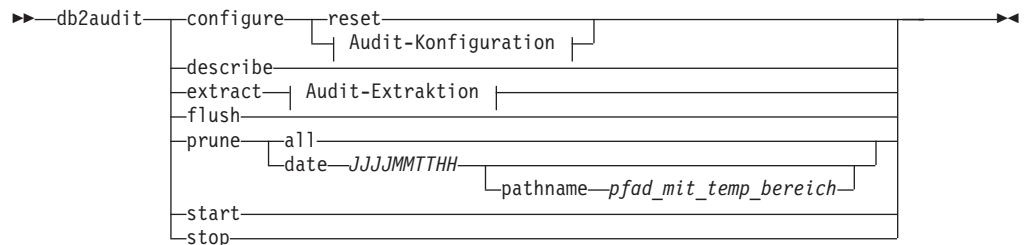
**Zugehörige Referenzen:**

- „audit\_buf\_sz - Prüfpuffergröße“ in *Systemverwaltung: Optimierung*
- „Verwendung der Prüffunktion“ auf Seite 287

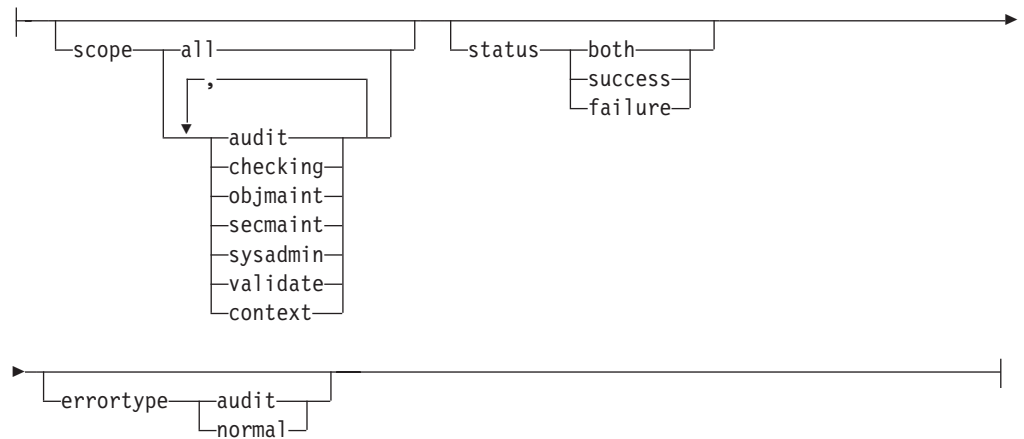
---

## Verwendung der Prüffunktion

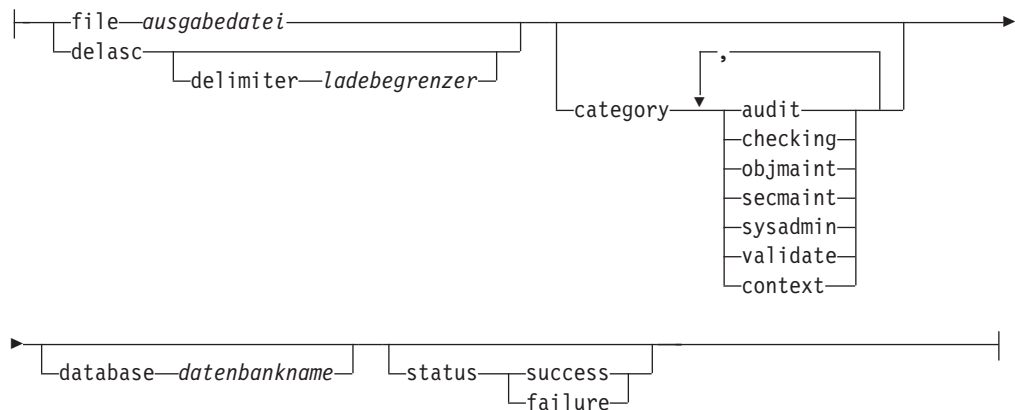
Wenn Sie die einzelnen Abschnitte der folgenden Syntaxdiagramme genau lesen, erleichtert dies das Verständnis für die Einsatzmöglichkeiten der Prüffunktion.



**Audit-Konfiguration:**



**Audit-Extraktion:**



Es folgt eine Beschreibung der einzelnen Parameter mit Hinweisen auf ihre implizierte Verwendung:

### 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 Prüfvorgänge von DB2 Universal Database™ (DB2 UDB) in 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 anwächst.

**Anmerkung:** Bitte beachten Sie, dass SCOPE standardmäßig alle Kategorien außer CONTEXT umfasst und rasch zu einer großen Anzahl generierter Prüfsätze führen kann. In Verbindung mit dem Modus (synchron oder asynchron) kann die Kategorienausswahl 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 UDB 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 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. Wenn kein Dateiname angegeben wird, werden die Datensätze in die Datei *db2audit.out* im Unterverzeichnis *security* des Verzeichnisses *sql1ib* geschrieben. Wenn kein Verzeichnis angegeben wird, wird die *ausgabedatei* in das aktuelle Arbeitsverzeichnis geschrieben.
- **DELASC.** Die extrahierten Prüfsätze werden im ASCII-Format mit Begrenzern ausgegeben, das in relationale DB2 UDB-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*

Diese Dateien werden immer in das Unterverzeichnis *security* des Verzeichnisses *sql1ib* geschrieben.

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).

- **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, wird aus allen Datenbanken extrahiert.

- STATUS. Die Prüfsätze für den angegebenen Status sollen extrahiert werden. Wird diese Option nicht angegeben, werden alle Prüfsätze extrahiert.

**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 für den Parameter PRUNE verwendet werden:

- ALL. Alle Prüfsätze im Prüfprotokoll sollen gelöscht werden.
- DATE jjjmmthh. Der Benutzer kann festlegen, dass alle bis zu dem angegebenen Tag/Zeitpunkt generierten Prüfsätze aus dem Prüfprotokoll gelöscht werden sollen. Der Benutzer kann optional den Parameter `pathname`

mit einem Pfad für einen temporären Speicherbereich angeben, der von der Prüffunktion beim Leeren des Protokolls 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 UDB-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 UDB-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.

#### Zugehörige Konzepte:

- „Einführung in die Prüffunktion von DB2 Universal Database (DB2 UDB)“ auf Seite 283
- „Tipps und Techniken für die Prüffunktion“ auf Seite 319

#### Zugehörige Referenzen:

- „db2audit - Audit Facility Administrator Tool Command“ in *Command Reference*

---

## Arbeiten mit DB2-Prüfdaten in DB2-Tabellen

Die folgenden Abschnitte beschreiben die Erstellung von DB2-Prüfdaten, die Erstellung von Tabellen zur Aufnahme dieser Daten, das Füllen von DB2-Prüfdaten in die Tabellen sowie die Auswahl der DB2-Prüfdaten aus den Tabellen.

## Arbeiten mit DB2-Prüfdaten in DB2-Tabellen

Wenn Sie die DB2-Prüffunktion zur Verwaltung eines Prüfprotokolls der Datenbankaktivitäten verwenden, legt die Prüffunktion die Prüfsätze standardmäßig in einer Protokolldatei ab. Wenn Sie dies wünschen, können Sie die Protokolleinträge aus der Protokolldatei in eine Textdatei schreiben, oder Sie können die Protokolleinträge aus der Protokolldatei in Dateien des ASCII-Formats mit Begrenzern schreiben und anschließend den Inhalt der ASCII-Dateien in DB2-Tabellen laden. Wenn sich die Prüfdaten in DB2-Tabellen befinden, können Sie die Daten aus den Tabellen auswählen, um Antworten auf Fragen zu erhalten, die Sie zur Aktivität Ihres DB2-Exemplars vielleicht haben.

### Vorgehensweise:

Gehen Sie wie folgt vor, um mit Prüfdaten in DB2-Tabellen zu arbeiten:

1. Erstellen Sie Tabellen zur Aufnahme der DB2-Prüfdaten.
2. Erstellen die DB2-Prüfdatendateien.
3. Laden Sie die Daten mit Hilfe des Dienstprogramms LOAD in die Tabellen.
4. Wählen Sie die Tabellendaten aus.

### Zugehörige Konzepte:

- „Funktionsweise der Prüffunktion“ auf Seite 285
- „Tipps und Techniken für die Prüffunktion“ auf Seite 319

### Zugehörige Tasks:

- „Erstellen von DB2-Prüfdatendateien“ auf Seite 294
- „Erstellen von Tabellen zur Aufnahme der DB2-Prüfdaten“ auf Seite 291
- „Laden von DB2-Prüfdaten in Tabellen“ auf Seite 296
- „Auswählen von DB2-Prüfdaten aus Tabellen“ auf Seite 298

### Zugehörige Referenzen:

- „Verwendung der Prüffunktion“ auf Seite 287

## Erstellen von Tabellen zur Aufnahme der DB2-Prüfdaten

Bevor Sie mit Prüfdaten in Tabellen arbeiten können, müssen Sie die Tabellen erstellen, in denen die Daten gespeichert werden sollen. Sie sollten in Betracht ziehen, diese Tabelle in einem separaten Schema zu erstellen, um die Daten in den Tabellen von unbefugten Benutzern zu isolieren.

### Voraussetzungen:

- Die zur Erstellung eines Schemas erforderlichen Berechtigungen und Zugriffsrechte finden Sie in den Informationen zur Anweisung CREATE SCHEMA.
- Die zur Erstellung einer Tabelle erforderlichen Berechtigungen und Zugriffsrechte finden Sie in den Informationen zur Anweisung CREATE TABLE.
- Legen Sie fest, welcher Tabellenbereich zum Speichern der Tabellen verwendet werden soll. (Dieser Abschnitt behandelt keine Erstellung von Tabellenbereichen.)

### Vorgehensweise:

In den folgenden Beispielen wird gezeigt, wie Tabellen erstellt werden, die alle Datensätze aus sämtlichen ASCII-Dateien aufnehmen können. Falls erwünscht, können Sie ein separates Schema zum Speichern dieser Tabellen erstellen.

Wenn Sie nicht alle Daten verwenden wollen, die in den Dateien enthalten sind, können Sie je nach Bedarf Spalten aus den Tabellendefinitionen weglassen oder die Erstellung von Tabellen übergehen. Wenn Sie Spalten aus den Tabellendefinitionen weglassen, müssen Sie die Befehle modifizieren, die Sie zum Laden von Daten in diese Tabellen verwenden.

1. Führen Sie den Befehl **db2** aus, um ein DB2-Befehlsfenster zu öffnen.
2. Optional. Erstellen Sie ein Schema zum Speichern der Tabellen. Führen Sie den folgenden Befehl aus. In diesem Beispiel erhält das Schema den Namen AUDIT:

```
CREATE SCHEMA AUDIT
```

3. Optional. Wenn Sie das Schema AUDIT erstellt haben, wechseln Sie zu diesem Schema, bevor Sie Tabellen erstellen. Führen Sie den folgenden Befehl aus:

```
SET CURRENT SCHEMA = 'AUDIT'
```

4. Zur Erstellung der Tabelle, in der Datensätze aus der Datei audit.del gespeichert werden, führen Sie die folgende SQL-Anweisung aus:

```
CREATE TABLE AUDIT (TIMESTAMP CHAR(26),  
                    CATEGORY CHAR(8),  
                    EVENT VARCHAR(32),  
                    CORRELATOR INTEGER,  
                    STATUS INTEGER,  
                    USERID VARCHAR(1024),  
                    AUTHID VARCHAR(128))
```

5. Zur Erstellung der Tabelle, in der Datensätze aus der Datei checking.del gespeichert werden, führen Sie die folgende SQL-Anweisung aus:

```
CREATE TABLE CHECKING (TIMESTAMP CHAR(26),  
                      CATEGORY CHAR(8),  
                      EVENT VARCHAR(32),  
                      CORRELATOR INTEGER,  
                      STATUS INTEGER,  
                      DATABASE CHAR(8),  
                      USERID VARCHAR(1024),  
                      AUTHID VARCHAR(128),  
                      NODENUM SMALLINT,  
                      COORDNUM SMALLINT,  
                      APPID VARCHAR(255),  
                      APPNAME VARCHAR(1024),  
                      PKGSCHEMA VARCHAR(128),  
                      PKGNAME VARCHAR(128),  
                      PKGSECNUM SMALLINT,  
                      OBJSCHEMA VARCHAR(128),  
                      OBJNAME VARCHAR(128),  
                      OBJTYPE VARCHAR(32),  
                      ACCESSAPP CHAR(18),  
                      ACCESSATT CHAR(18),  
                      PKGVER VARCHAR(64))
```

6. Zur Erstellung der Tabelle, in der Datensätze aus der Datei objmaint.del gespeichert werden, führen Sie die folgende SQL-Anweisung aus:

```
CREATE TABLE OBJMAINT (TIMESTAMP CHAR(26),  
                      CATEGORY CHAR(8),  
                      EVENT VARCHAR(32),  
                      CORRELATOR INTEGER,  
                      STATUS INTEGER,  
                      DATABASE CHAR(8),  
                      USERID VARCHAR(1024),  
                      AUTHID VARCHAR(128),  
                      NODENUM SMALLINT,  
                      COORDNUM SMALLINT,  
                      APPID VARCHAR(255),  
                      APPNAME VARCHAR(1024),  
                      PKGSCHEMA VARCHAR(128),
```

```
PKGNAME VARCHAR(128),
PKGSECNUM SMALLINT,
OBJSCHEMA VARCHAR(128),
OBJNAME VARCHAR(128),
OBJTYPE VARCHAR(32),
PACKVER VARCHAR(64)
```

7. Zur Erstellung der Tabelle, in der Datensätze aus der Datei secmaint.del gespeichert werden, führen Sie die folgende SQL-Anweisung aus:

```
CREATE TABLE SECMAINT (TIMESTAMP CHAR(26),
CATEGORY CHAR(8),
EVENT VARCHAR(32),
CORRELATOR INTEGER,
STATUS INTEGER,
DATABASE CHAR(8),
USERID VARCHAR(1024),
AUTHID VARCHAR(128),
NODENUM SMALLINT,
COORDNUM SMALLINT,
APPID VARCHAR(255),
APPNAME VARCHAR(1024),
PKGSHEMA VARCHAR(128),
PKGNAME VARCHAR(128),
PKGSECNUM SMALLINT,
OBJSCHEMA VARCHAR(128),
OBJNAME VARCHAR(128),
OBJTYPE VARCHAR(32),
GRANTOR VARCHAR(128),
GRANTEE VARCHAR(128),
GRANTEETYPE VARCHAR(32),
PRIVAUTH CHAR(18),
PKGVER VARCHAR(64))
```

8. Zur Erstellung der Tabelle, in der Datensätze aus der Datei sysadmin.del gespeichert werden, führen Sie die folgende SQL-Anweisung aus:

```
CREATE TABLE SYSADMIN (TIMESTAMP CHAR(26),
CATEGORY CHAR(8),
EVENT VARCHAR(32),
CORRELATOR INTEGER,
STATUS INTEGER,
DATABASE CHAR(8),
USERID VARCHAR(1024),
AUTHID VARCHAR(128),
NODENUM SMALLINT,
COORDNUM SMALLINT,
APPID VARCHAR(255),
APPNAME VARCHAR(1024),
PKGSHEMA VARCHAR(128),
PKGNAME VARCHAR(128),
PKGSECNUM SMALLINT,
PKGVER VARCHAR(64))
```

9. Zur Erstellung der Tabelle, in der Datensätze aus der Datei validate.del gespeichert werden, führen Sie die folgende SQL-Anweisung aus:

```
CREATE TABLE VALIDATE (TIMESTAMP CHAR(26),
CATEGORY CHAR(8),
EVENT VARCHAR(32),
CORRELATOR INTEGER,
STATUS INTEGER,
DATABASE CHAR(8),
USERID VARCHAR(1024),
AUTHID VARCHAR(128),
EXECID VARCHAR(1024),
NODENUM SMALLINT,
COORDNUM SMALLINT,
APPID VARCHAR(255),
APPNAME VARCHAR(1024),
```

```
AUHTYPE VARCHAR(32),
PKGSHEMA VARCHAR(128),
PKGNAME VARCHAR(128),
PKGSECNUM SMALLINT,
PKGVER VARCHAR(64)
PLUGINNAME VARCHAR(32))
```

10. Zur Erstellung der Tabelle, in der Datensätze aus der Datei `context.del` gespeichert werden, führen Sie die folgende SQL-Anweisung aus:

```
CREATE TABLE CONTEXT (TIMESTAMP CHAR(26),
CATEGORY CHAR(8),
EVENT VARCHAR(32),
CORRELATOR INTEGER,
DATABASE CHAR(8),
USERID VARCHAR(1024),
AUTHID VARCHAR(128),
NODENUM SMALLINT,
COORDNUM SMALLINT,
APPID VARCHAR(255),
APPNAME VARCHAR(1024),
PKGSHEMA VARCHAR(128),
PKGNAME VARCHAR(128),
PKGSECNUM SMALLING,
STMTTEXT CLOB(2M),
PKGVER VARCHAR(64))
```

11. Nach der Erstellung der Tabellen führen Sie die Anweisung `COMMIT` aus, um sicherzustellen, dass die Tabellendefinitionen auf die Platte geschrieben werden.
12. Wenn Sie die Tabellen erstellt haben, können Sie die Prüfsätze aus der Datei `'db2audit.log'` in Dateien im ASCII-Format mit Begrenzern zu extrahieren.

#### Zugehörige Tasks:

- „Einstellen eines Schemas“ auf Seite 93

#### Zugehörige Referenzen:

- „CREATE SCHEMA statement“ in *SQL Reference, Volume 2*
- „CREATE TABLE statement“ in *SQL Reference, Volume 2*

## Erstellen von DB2-Prüfdatendateien

Standardmäßig schreibt die DB2-Prüffunktion Prüfdaten in die Datei `db2audit.log`. Die Datensätze in dieser Datei können nicht in Tabellen geladen werden. Sie müssen die Prüfsätze in Dateien im ASCII-Format mit Begrenzern extrahieren, die Sie anschließend zum Füllen von Tabellen verwenden können.

#### Voraussetzungen:

Zum Ausführen des Befehls `db2audit` ist die Berechtigung `SYSADM` erforderlich.

#### Vorgehensweise:

Gehen Sie wie folgt vor, um Datensätze der Prüffunktion in Dateien im ASCII-Format mit Begrenzern zu schreiben:

1. Legen Sie mit Hilfe der Informationen im Abschnitt zur Verwendung der Prüffunktion fest, welche Art von DB2-Aktivitäten geprüft werden sollen.

Wenn die für die Prüffunktion festgelegte Konfiguration Ihren Vorstellungen entspricht, führen Sie den folgenden Befehl aus, um mit der Aufzeichnung der Prüfdaten zu beginnen:

```
db2audit start
```

2. Geben Sie den folgenden Befehl ein, um sicherzustellen, dass alle Prüfsätze aus dem Speicher gelöscht und in der Datei `db2audit.log` gespeichert werden:

```
db2audit flush
```

3. Geben Sie den folgenden Befehl ein, um die Prüfsätze aus der Datei `db2audit.log` in Dateien im ASCII-Format mit Begrenzern zu versetzen:

```
db2audit extract delasc
```

Die folgenden Dateien werden im Unterverzeichnis `security` des Verzeichnisses `sql1ib` erstellt. Wenn Sie einen bestimmten Typ von Ereignis nicht prüfen, wird die Datei für den Ereignistyp erstellt, jedoch bleibt die Datei leer.

- `audit.del`
- `checking.del`
- `objmaint.del`
- `secmaint.del`
- `sysadmin.del`
- `validate.del`
- `context.del`

4. Geben Sie den folgenden Befehl ein, um die Prüfsätze aus der Datei `db2audit.log` zu löschen, aus der Sie gerade extrahiert haben:

```
db2audit prune date JJJJMMTHH
```

Dabei steht `JJJJMMTHH` für den aktuellen Zeitpunkt, der sich aus Jahr, Monat, Tag und Stunde zusammensetzt. Notieren Sie sich den Wert, den Sie verwenden, da Sie diese Informationen im nächsten Schritt benötigen, wenn Sie die Tabellen mit den Prüfdaten füllen.

Die Prüffunktion fährt mit dem Schreiben neuer Prüfsätze in die Datei `db2audit.log` fort, die nun eine Zeitmarke eines Zeitpunkts nach `JJJJMMTHH` erhalten. Durch das Entfernen von Datensätzen, die Sie bereits aus der Datei `db2audit.log` extrahiert haben, können Sie vermeiden, dieselben Datensätze ein zweites Mal zu extrahieren. Alle Prüfsätze, die nach `JJJJMMTHH` geschrieben werden, werden bei der nächsten Extraktion der Prüfdaten in die `.del`-Dateien geschrieben.

5. Nach der Erstellung der Prüfdatendateien besteht der nächste Schritt darin, die Tabellen mit Hilfe des Dienstprogramms `LOAD` mit den Prüfdaten zu füllen.

#### Zugehörige Referenzen:

- „db2audit - Audit Facility Administrator Tool Command“ in *Command Reference*
- „Verwendung der Prüffunktion“ auf Seite 287
- „Prüfsatzaufbau für AUDIT-Ereignisse“ auf Seite 301
- „Prüfsatzaufbau für CHECKING-Ereignisse“ auf Seite 301
- „Prüfsatzaufbau für OBJMAINT-Ereignisse“ auf Seite 307
- „Prüfsatzaufbau für SECMAINT-Ereignisse“ auf Seite 309
- „Prüfsatzaufbau für SYSADMIN-Ereignisse“ auf Seite 313
- „Prüfsatzaufbau für VALIDATE-Ereignisse“ auf Seite 316
- „Prüfsatzaufbau für CONTEXT-Ereignisse“ auf Seite 317

## Laden von DB2-Prüfdaten in Tabellen

Wenn Sie die Tabellen zur Aufnahme der Prüfdaten erstellt haben, können Sie die Daten anschließend aus den ASCII-Dateien in die Tabellen laden.

### Voraussetzungen:

Weitere Informationen hierzu finden Sie im Abschnitt zu den Zugriffsrechten und Berechtigungen, die zur Verwendung des Dienstprogramms LOAD erforderlich sind.

### Vorgehensweise:

Verwenden Sie das Dienstprogramm LOAD, um die Daten in die Tabellen zu laden. Führen Sie für jede Tabelle einen getrennten LOAD-Befehl aus. Wenn Sie eine oder mehrere Spalten aus den Tabellendefinitionen weggelassen haben, müssen Sie die Version des LOAD-Befehls modifizieren, um die Daten erfolgreich zu laden. Wenn Sie ein anderes Begrenzungszeichen als das Standardzeichen (0xff) beim Extrahieren der Prüfdaten angegeben haben, müssen Sie die von Ihnen verwendete Version des LOAD-Befehls ebenfalls modifizieren (siehe weitere Informationen unter "Dateitypwerte für LOAD").

1. Führen Sie den Befehl **db2** aus, um ein DB2-Befehlsfenster zu öffnen.
2. Zum Laden von Daten in die Tabelle AUDIT führen Sie den folgenden Befehl aus:

```
LOAD FROM audit.del OF del MODIFIED BY CHARDEL0xff INSERT INTO schema.AUDIT
```

**Anmerkung:** Verwenden Sie zur Angabe des Dateinamen den vollständig qualifizierten Pfadnamen. Zum Beispiel müsste bei einer Installation von DB2 UDB auf Laufwerk C: eines Windows-basierten Computers der vollständig qualifizierte Name der Datei `audit.del` wie folgt angegeben werden: `C:\Program Files\IBM\SQLLIB\instance\security\audit.del`

Nach dem Laden von Daten in die Tabelle AUDIT führen Sie die folgende DELETE-Anweisung aus, um sicherzustellen, dass Sie beim nächsten Laden von Daten keine doppelten Zeilen in die Tabelle laden. Als Sie die Prüfsätze aus der Datei `db2audit.log` extrahierten, wurden alle Datensätze in die `.del`-Dateien geschrieben. Mit einiger Wahrscheinlichkeit enthielten die `.del`-Dateien jedoch Datensätze, die nach dem Zeitpunkt geschrieben wurden, bis zu dem das Prüfprotokoll nachfolgend gelöscht wurde (weil der Befehl **db2audit prune** Datensätze nur bis zu einem angegebenen Zeitpunkt löscht). Wenn Sie das nächste Mal Prüfsätze extrahieren, werden die neuen `.del`-Dateien Datensätze enthalten, die zuvor extrahiert, jedoch nicht durch den Befehl **db2audit prune** gelöscht wurden (weil sie nach dem für die PRUNE-Operation angegebenen Zeitpunkt geschrieben wurden). Durch das Löschen von Zeilen aus der Tabelle bis zu eben dem Zeitpunkt, bis zu dem die Datensätze aus der Datei `db2audit.log` gelöscht wurden, lässt sich sicherstellen, dass die Tabelle keine doppelten Zeilen enthält und keine Prüfsätze verloren gehen.

```
DELETE FROM schema.AUDIT WHERE TIMESTAMP > TIMESTAMP('JJJMMTTHH0000')
```

Dabei ist `JJJMMTTHH` der Wert, den Sie angegeben haben, als Sie Datensätze durch eine PRUNE-Operation aus der Datei `db2audit.log` entfernt haben.



Da die DB2-Prüffunktion das Schreiben von Datensätzen in die Datei db2audit.log nach dem Entfernen von Datensätzen fortsetzt, müssen Sie den Wert 0000 für Minuten und Sekunden angeben, um sicherzustellen, dass Prüf-sätze, die nach dem Entfernen von Prüfsätzen aus der Datei db2audit.log geschrieben wurden, nicht aus der Tabelle gelöscht werden.

3. Zum Laden von Daten in die Tabelle CHECKING führen Sie den folgenden Befehl aus:

```
LOAD FROM checking.del OF del MODIFIED BY CHARDEL0xff INSERT INTO
schema.CHECKING
```

Nach dem Laden von Daten in die Tabelle CHECKING führen Sie die folgende SQL-Anweisung aus, um sicherzustellen, dass Sie beim nächsten Laden von Daten keine doppelten Zeilen in die Tabelle laden:

```
DELETE FROM schema.CHECKING WHERE TIMESTAMP > TIMESTAMP('JJJMMTTHH0000')
```

Dabei ist JJJMMTTHH der Wert, den Sie angegeben haben, als Sie Datensätze aus der Protokolldatei entfernten.

4. Zum Laden von Daten in die Tabelle OBJMAINT führen Sie den folgenden Befehl aus:

```
LOAD FROM objmaint.del OF del MODIFIED BY CHARDEL0xff INSERT INTO
schema.OBJMAINT
```

Nach dem Laden von Daten in die Tabelle OBJMAINT führen Sie die folgende SQL-Anweisung aus, um sicherzustellen, dass Sie beim nächsten Laden von Daten keine doppelten Zeilen in die Tabelle laden:

```
DELETE FROM schema.OBJMAINT WHERE TIMESTAMP > TIMESTAMP('JJJMMTTHH0000')
```

Dabei ist JJJMMTTHH der Wert, den Sie angegeben haben, als Sie Datensätze aus der Protokolldatei entfernten.

5. Zum Laden von Daten in die Tabelle SECMAINT führen Sie den folgenden Befehl aus:

```
LOAD FROM secmaint.del OF del MODIFIED BY CHARDEL0xff INSERT INTO
schema.SECMAINT
```

Nach dem Laden von Daten in die Tabelle SECMAINT führen Sie die folgende SQL-Anweisung aus, um sicherzustellen, dass Sie beim nächsten Laden von Daten keine doppelten Zeilen in die Tabelle laden:

```
DELETE FROM schema.SECMAINT WHERE TIMESTAMP > TIMESTAMP('JJJMMTTHH0000')
```

Dabei ist JJJMMTTHH der Wert, den Sie angegeben haben, als Sie Datensätze aus der Protokolldatei entfernten.

6. Zum Laden von Daten in die Tabelle SYSADMIN führen Sie den folgenden Befehl aus:

```
LOAD FROM sysadmin.del OF del MODIFIED BY CHARDEL0xff INSERT INTO
schema.SYSADMIN
```

Nach dem Laden von Daten in die Tabelle SYSADMIN führen Sie die folgende SQL-Anweisung aus, um sicherzustellen, dass Sie beim nächsten Laden von Daten keine doppelten Zeilen in die Tabelle laden:

```
DELETE FROM schema.SYSADMIN WHERE TIMESTAMP > TIMESTAMP('JJJMMTTHH0000')
```

Dabei ist JJJMMTTHH der Wert, den Sie angegeben haben, als Sie Datensätze aus der Protokolldatei entfernten.

7. Zum Laden von Daten in die Tabelle VALIDATE führen Sie den folgenden Befehl aus:

```
LOAD FROM validate.del OF del MODIFIED BY CHARDEL0xff INSERT INTO
schema.VALIDATE
```

Nach dem Laden von Daten in die Tabelle VALIDATE führen Sie die folgende SQL-Anweisung aus, um sicherzustellen, dass Sie beim nächsten Laden von Daten keine doppelten Zeilen in die Tabelle laden:

```
DELETE FROM schema.VALIDATE WHERE TIMESTAMP > TIMESTAMP('JJJMMTTHH0000')
```

Dabei ist JJJMMTTHH der Wert, den Sie angegeben haben, als Sie Datensätze aus der Protokolldatei entfernten.

8. Zum Laden von Daten in die Tabelle CONTEXT führen Sie den folgenden Befehl aus:

```
LOAD FROM context.del OF del MODIFIED BY CHARDEL0xff INSERT INTO  
schema.CONTEXT
```

Nach dem Laden von Daten in die Tabelle CONTEXT führen Sie die folgende SQL-Anweisung aus, um sicherzustellen, dass Sie beim nächsten Laden von Daten keine doppelten Zeilen in die Tabelle laden:

```
DELETE FROM schema.CONTEXT WHERE TIMESTAMP > TIMESTAMP('JJJMMTTHH0000')
```

Dabei ist JJJMMTTHH der Wert, den Sie angegeben haben, als Sie Datensätze aus der Protokolldatei entfernten.

9. Nach Abschluss des Ladens der Daten in die Tabellen, löschen Sie die .del-Dateien aus dem Unterverzeichnis security des Verzeichnisses sql11b.
10. Wenn Sie die Prüfdaten in die Tabellen geladen haben, können Sie Daten aus diesen Tabellen auswählen.

Wenn Sie die Tabellen bereits ein erstes Mal gefüllt haben und dies wiederholen wollen, verwenden Sie die Option INSERT, um die neuen Tabellendaten den vorhandenen Tabellendaten hinzuzufügen. Wenn Sie die Datensätze der vorigen Operation **db2audit extract** aus den Tabellen entfernen wollen, laden Sie die neuen Daten mit der Option REPLACE in die Tabellen. In beiden Fällen sollten Sie darauf achten, die Prüfsätze in die Datei db2audit.log mit einer FLUSH-Operation auszulesen, bevor Sie die Datensätze in die .del-Dateien extrahieren, und die Datensätze nach der Extraktion aus der Datei db2audit.log mit einer PRUNE-Operation zu entfernen, um zu vermeiden, dieselben Datensätze mehrmals in die Tabellen zu laden.

#### Zugehörige Tasks:

- „Arbeiten mit DB2-Prüfdaten in DB2-Tabellen“ auf Seite 291

## Auswählen von DB2-Prüfdaten aus Tabellen

Wenn Sie die Prüfdaten erfolgreich in die Tabellen geladen haben, können Sie Daten zur weiteren Analyse aus diesen Tabellen auswählen.

#### Voraussetzungen:

Informationen zu den Berechtigungen und Zugriffsrechten, die zum Auswählen von Daten aus einer Tabelle erforderlich sind, finden Sie im Abschnitt zur Anweisung SELECT.

### Vorgehensweise:

Gehen Sie wie folgt vor, um alle Zeilen einer Tabelle auszuwählen:

1. Führen Sie den Befehl **db2** aus, um ein DB2-Befehlsfenster zu öffnen.
2. Führen Sie eine SQL-Anweisung der folgenden Form für jede Tabelle aus, aus der Sie Prüfdaten auswählen wollen:

```
SELECT * FROM schema.tabelle
```

Um zum Beispiel alle Daten aus der Tabelle CHECKING im Schema AUDIT auszuwählen, verwenden Sie die folgende Anweisung:

```
SELECT * FROM AUDIT.CHECKING
```

Die SELECT-Anweisung, die Sie ausführen, sollte den Typ von Analyse berücksichtigen, die Sie an den Daten ausführen wollen. Zum Beispiel können Sie Datensätze nach einer Berechtigungs-ID (authid) auswählen, um festzustellen, welche Art von Aktivitäten durch diese Berechtigungs-ID ausgeführt wurden:

```
SELECT * FROM AUDIT.CHECKING WHERE AUTHID = berechtigungs-id
```

Dabei stellt *berechtigungs-id* die Benutzer-ID dar, für die Sie die Daten analysieren wollen.

Eine Beschreibung der Werte, die in Prüfdaten enthalten sein können, finden Sie in den Informationen über den Prüfsatzaufbau der entsprechenden Tabelle sowie in der Liste der möglichen Rückgabewerte für die Tabelle.

### Zugehörige Referenzen:

- „Subselect“ in *SQL Reference, Volume 1*
- „SELECT statement“ in *SQL Reference, Volume 2*
- „Prüfsatzaufbau für AUDIT-Ereignisse“ auf Seite 301
- „Prüfsatzaufbau für CHECKING-Ereignisse“ auf Seite 301
- „Liste möglicher Gründe für die Zugriffsgewährung bei CHECKING-Ereignissen“ auf Seite 304
- „Liste der möglichen Typen von Zugriffsversuchen bei CHECKING-Ereignissen“ auf Seite 305
- „Prüfsatzaufbau für OBJMAINT-Ereignisse“ auf Seite 307
- „Prüfsatzaufbau für SECMAINT-Ereignisse“ auf Seite 309
- „Liste der möglichen Zugriffsrechte oder Berechtigungen für SECMAINT-Ereignisse“ auf Seite 310
- „Prüfsatzaufbau für SYSADMIN-Ereignisse“ auf Seite 313
- „Liste der möglichen SYSADMIN-Prüfereignisse“ auf Seite 315
- „Prüfsatzaufbau für VALIDATE-Ereignisse“ auf Seite 316
- „Prüfsatzaufbau für CONTEXT-Ereignisse“ auf Seite 317
- „Liste der möglichen CONTEXT-Prüfereignisse“ auf Seite 319

---

## Nachrichten der Prüffunktion

---

### SQL1322N Beim Schreiben in die Prüfprotokolldatei ist ein Fehler aufgetreten.

| **Erläuterung:** Beim Aufrufen der Prüffunktion von DB2 Universal Database™ (DB2 UDB) 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

#### Zugehörige Konzepte:

- „Einführung in die Prüffunktion von DB2 Universal Database (DB2 UDB)“ auf Seite 283

### 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

---

## Aufbau der Datensätze der Prüffunktion (Einführung)

Jeder Prüfsatz, der mit der Option DELASC aus dem Prüfprotokoll extrahiert wird, hat 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.
3. Ein neues Feld mit dem Namen „Package Version“ (Paketversion) wurde dem Prüfsatzlayout für die Ereignisse der Typen CHECKING, OBJMAINT, SECMAINT, SYSADMIN, VALIDATE und CONTEXT hinzugefügt.

#### Zugehörige Referenzen:

- „Prüfsatzaufbau für AUDIT-Ereignisse“ auf Seite 301
- „Prüfsatzaufbau für CHECKING-Ereignisse“ auf Seite 301
- „Prüfsatzaufbau für OBJMAINT-Ereignisse“ auf Seite 307
- „Prüfsatzaufbau für SECMAINT-Ereignisse“ auf Seite 309
- „Prüfsatzaufbau für SYSADMIN-Ereignisse“ auf Seite 313
- „Prüfsatzaufbau für VALIDATE-Ereignisse“ auf Seite 316
- „Prüfsatzaufbau für CONTEXT-Ereignisse“ auf Seite 317

## Details zu den Datensatzlayouts der Prüffunktion

In diesem Abschnitt werden die verschiedenen Datensatzlayouts der Prüffunktion beschrieben.

### Prüfsatzaufbau für AUDIT-Ereignisse

Tabelle 7. 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.

#### Zugehörige Konzepte:

- „Aufbau der Datensätze der Prüffunktion (Einführung)“ auf Seite 300

### Prüfsatzaufbau für CHECKING-Ereignisse

Tabelle 8. 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; package schema=NULLID;package name=SYSSH200; package section=0;object schema=GSTAGER;object name=NONE;object type=REOPT_VALUES; access approval reason=DBADM;access attempted=STORE;		
NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des AUDIT-Ereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Gültige Werte:  CHECKING

Tabelle 8. 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; package schema=NULLID;package name=SYSSH200; package section=0;object schema=GSTAGER;object name=NONE;object type=REOPT_VALUES; access approval reason=DBADM;access attempted=STORE;</pre>		
NAME	FORMAT	BESCHREIBUNG
<b>Audit Event (Prüfereignis)</b>	VARCHAR(32)	Spezifisches Prüfereignis.  Gültige Werte: CHECKING_OBJECT und CHECKING_FUNCTION
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Gibt an, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
<b>Event Status (Ereignisstatus)</b>	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 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 (Objekttyp)	VARCHAR(32)	Der Typ des Objekts, für das das Prüfereignis generiert wurde. Gültige Werte: Die im Thema mit dem Titel „Prüfsatzobjekttypen“ angegebenen Werte.

Tabelle 8. 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; package schema=NULLID;package name=SYSSH200; package section=0;object schema=GSTAGER;object name=NONE;object type=REOPT_VALUES; access approval reason=DBADM;access attempted=STORE;</pre>		
NAME	FORMAT	BESCHREIBUNG
Access Approval Reason (Grund für Zugriffsgewährung)	CHAR(18)	Gibt an, warum der Zugriff für dieses Prüfereignis gewährt wurde. Gültige Werte: Die im Thema mit dem Titel „Liste möglicher Gründe für die Zugriffsgewährung bei CHECKING-Ereignissen“ angegebenen Werte.
Access Attempted (Zugriffsversuch)	CHAR(18)	Gibt an, welche Art von Zugriff versucht wurde. Gültige Werte: Die im Thema mit dem Titel „Liste der möglichen Typen von Zugriffsversuchen bei CHECKING-Ereignissen“ angegebenen Werte.
Package Version (Paketversion)	VARCHAR (64)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.

**Zugehörige Konzepte:**

- „Aufbau der Datensätze der Prüffunktion (Einführung)“ auf Seite 300

**Zugehörige Referenzen:**

- „Liste möglicher Gründe für die Zugriffsgewährung bei CHECKING-Ereignissen“ auf Seite 304
- „Liste der möglichen Typen von Zugriffsversuchen bei CHECKING-Ereignissen“ auf Seite 305
- „Prüfsatzobjekttypen“ auf Seite 303

## Prüfsatzobjekttypen

Tabelle 9. Prüfsatzobjekttypen auf der Basis der Prüfereignisse

Objekttyp	CHECKING-Ereignisse	OBJMAINT-Ereignisse	SECMAINT-Ereignisse
NONE	X	X	X
TABLE	X	X	X
VIEW	X	X	X
ALIAS	X	X	
FUNCTION	X	X	X
INDEX	X	X	X
INDEX EXTENSION		X	
PACKAGE	X	X	X
PACKAGE CACHE	X		
DATA_TYPE		X	
NODEGROUP	X	X	
SCHEMA	X	X	X
STORED_PROCEDURE	X	X	X
METHOD_BODY	X	X	X
BUFFERPOOL	X	X	
SEQUENCE	X	X	

Tabelle 9. Prüfsatzobjekttypen auf der Basis der Prüfereignisse (Forts.)

Objekttyp	CHECKING-Ereignisse	OBJMAINT-Ereignisse	SECMAINT-Ereignisse
TABLESPACE	X	X	X
EVENT_MONITOR	X	X	
TRIGGER		X	
DATABASE	X		X
INSTANCE	X		
FOREIGN_KEY		X	
PRIMARY_KEY		X	
UNIQUE_CONSTRAINT		X	
CHECK_CONSTRAINT		X	
WRAPPER	X	X	
SERVER	X	X	X
NICKNAME	X	X	X
USER MAPPING	X	X	
SERVER OPTION	X	X	
TYPE&TRANSFORM	X	X	
TYPE MAPPING	X	X	
FUNCTION MAPPING	X	X	
SUMMARY TABLES	X	X	X
JAR_FILE		X	
ALL	X		
REOPT_VALUES	X		

**Zugehörige Referenzen:**

- „Prüfsatzaufbau für CHECKING-Ereignisse“ auf Seite 301
- „Prüfsatzaufbau für OBJMAINT-Ereignisse“ auf Seite 307
- „Prüfsatzaufbau für SECMAINT-Ereignisse“ auf Seite 309

## Liste möglicher Gründe für die Zugriffsgewährung bei CHECKING-Ereignissen

In der folgenden Liste sind mögliche Gründe für die Zugriffsgewährung für CHECKING-Ereignisse 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 Eigner 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.

**0x0000000000000800 SYSQUIESCE**

Der Zugriff wird gewährt. Wenn das Exemplar oder die Datenbank im Wartemodus ist, kann die Anwendung bzw. der Benutzer eine Verbindung (CONNECT oder ATTACH) herstellen.

**0x0000000000001000 SYSMON**

Der Zugriff wird gewährt. Die Anwendung/der Benutzer verfügt über die Berechtigung SYSMON.

**Zugehörige Referenzen:**

- „Prüfsatzaufbau für CHECKING-Ereignisse“ auf Seite 301
- „Liste der möglichen Typen von Zugriffsversuchen bei CHECKING-Ereignissen“ auf Seite 305

## Liste der möglichen Typen von Zugriffsversuchen bei CHECKING-Ereignissen

In der folgenden Liste sind mögliche Arten von Zugriffsversuchen für 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 oder eine Routine aufzurufen, eine Funktion aus der Routine als Quelle (gilt nur für Funktionen) zu erstellen oder auf eine Routine in einer DDL-Anweisung zu verweisen.

**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.

**0x000000000800000 Member of SYS Group**

Es wurde versucht, auf ein Mitglied der SYS-Gruppe zuzugreifen oder ein Mitglied der SYS-Gruppe zu verwenden.

**0x000000001000000 Access All**

Es wurde versucht, eine Anweisung mit allen erforderlichen Zugriffsrechten für gespeicherte Objekte auszuführen (wird nur für DBADM/SYSADM verwendet).

**0x000000002000000 Drop All**

Es wurde versucht, mehrere Objekte zu löschen.

**0x000000004000000 LOAD**

Es wurde versucht, eine Tabelle in einen Tabellenbereich zu laden.

**0x000000008000000 USE**

Es wurde versucht, eine Tabelle in einem Tabellenbereich zu erstellen.

**0x0000000010000000 SET SESSION\_USER**

Es wurde versucht, eine Anweisung SET SESSION\_USER auszuführen.

**0x0000000020000000 FLUSH**

Es wurde versucht, eine Anweisung FLUSH auszuführen.

**0x0000000040000000 STORE**

Es wurde versucht, die Werte einer reoptimierten Anweisung in der Tabelle EXPLAIN\_PREDICATE anzuzeigen.

**Zugehörige Referenzen:**

- „Prüfsatzaufbau für CHECKING-Ereignisse“ auf Seite 301
- „Liste möglicher Gründe für die Zugriffsgewährung bei CHECKING-Ereignissen“ auf Seite 304

## Prüfsatzaufbau für OBJMAINT-Ereignisse

Tabelle 10. Prüfsatzaufbau für OBJMAINT-Ereignisse

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;		
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.

Tabelle 10. 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
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 Exemplebene 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 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 (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 (Objekttyp)	VARCHAR(32)	Der Typ des Objekts, für das das Prüfereignis generiert wurde. Gültige Werte: Die im Abschnitt „Prüfsatzobjekttypen“ angegebenen Werte.
Package Version (Paketversion)	VARCHAR (64)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.

**Zugehörige Konzepte:**

- „Einführung in die Prüffunktion von DB2 Universal Database (DB2 UDB)“ auf Seite 283

**Zugehörige Referenzen:**

- „Prüfsatzobjekttypen“ auf Seite 303

## Prüfsatzaufbau für SECMAINT-Ereignisse

Tabelle 11. 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, SET_SESSION_USER 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 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 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.
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.

Tabelle 11. 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 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 (Objekttyp)	VARCHAR(32)	Die Art des Objekts, für das das Prüfereignis generiert wurde. Gültige Werte: Die im Thema mit dem Titel „Prüfsatzobjekttypen“ angegebenen Werte.
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 im ersten Thema mit dem Titel „Liste möglicher Zugriffsrechte oder Berechtigungen für SECMAINT-Ereignisse“ angegebenen Werte.
Package Version (Paketversion)	VARCHAR (64)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.

**Zugehörige Konzepte:**

- „Aufbau der Datensätze der Prüffunktion (Einführung)“ auf Seite 300

**Zugehörige Referenzen:**

- „Liste der möglichen Zugriffsrechte oder Berechtigungen für SECMAINT-Ereignisse“ auf Seite 310
- „Prüfsatzobjekttypen“ auf Seite 303

## Liste der möglichen Zugriffsrechte oder Berechtigungen für SECMAINT-Ereignisse

In der folgenden Liste sind mögliche Zugriffsrechte und Berechtigungen für SECMAINT-Ereignisse aufgeführt:

**0x0000000000000001 Control Table**

Zugriffsrecht CONTROL für eine Tabelle oder Sicht 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 Zugriffsrechten erteilt oder widerrufen.

0x0000000000000008 **DELETE TABLE**  
 Zugriffsrecht zum Löschen einer Tabelle oder Sicht erteilt oder widerrufen.

0x0000000000000010 **DELETE TABLE with GRANT**  
 Zugriffsrecht zum Löschen einer Tabelle mit der Möglichkeit zum Erteilen von Zugriffsrechten 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 Zugriffsrechten erteilt oder widerrufen.

0x0000000000000080 **Table INSERT**  
 Zugriffsrecht zum Einfügen in eine Tabelle oder Sicht erteilt oder widerrufen.

0x0000000000000100 **Table INSERT with GRANT**  
 Zugriffsrecht zum Einfügen in eine Tabelle mit der Möglichkeit zum Erteilen von Zugriffsrechten 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 Zugriffsrechten erteilt oder widerrufen.

0x0000000000000800 **Table UPDATE**  
 Zugriffsrecht zum Aktualisieren einer Tabelle oder Sicht erteilt oder widerrufen.

0x0000000000001000 **Table UPDATE with GRANT**  
 Zugriffsrecht zum Aktualisieren einer Tabelle oder Sicht mit Möglichkeit zum Erteilen von Zugriffsrechten 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 Zugriffsrechten 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 Zugriffsrechten 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 Zugriffsrechten erteilt oder widerrufen.

0x0000000000200000 **ALTERIN Schema**  
 Zugriffsrecht ALTERIN zum Ändern in einem Schema erteilt oder widerrufen.

- 0x000000000400000 ALTERIN Schema with GRANT**  
Zugriffsrecht ALTERIN zum Ändern in einem Schema mit Möglichkeit zum Erteilen von Zugriffsrechten erteilt oder widerrufen.
- 0x000000000800000 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' (nicht abgeschirmte erstellen) 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 zusammenschlossener 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.
- 0x0000008000000000 Package BIND**  
Zugriffsrecht BIND zum Binden eines Pakets erteilt oder widerrufen.
- 0x0000010000000000 Package BIND with GRANT**  
Zugriffsrecht BIND für ein Paket mit der Möglichkeit zum Erteilen von Zugriffsrechten erteilt oder widerrufen.



**0x0000020000000000 EXECUTE**

Zugriffsrecht EXECUTE für ein Paket oder eine Routine erteilt oder widerrufen.

**0x0000040000000000 EXECUTE with GRANT**

Zugriffsrecht EXECUTE für ein Paket oder eine Routine mit der Möglichkeit zum Erteilen von Zugriffsrechten erteilt oder widerrufen.

**0x0000080000000000 EXECUTE IN SCHEMA**

Zugriffsrecht EXECUTE für alle Routinen in einem Schema erteilt oder widerrufen.

**0x0000100000000000 EXECUTE IN SCHEMA with GRANT**

Zugriffsrecht EXECUTE für alle Routinen in einem Schema mit der Möglichkeit zum Erteilen von Zugriffsrechten erteilt oder widerrufen.

**0x0000200000000000 EXECUTE IN TYPE**

Zugriffsrecht EXECUTE für alle Routinen in einem Typ erteilt oder widerrufen.

**0x0000400000000000 EXECUTE IN TYPE with GRANT**

Zugriffsrecht EXECUTE für alle Routinen in einem Typ mit der Möglichkeit zum Erteilen von Zugriffsrechten erteilt oder widerrufen.

**0x0000800000000000 CREATE EXTERNAL ROUTINE**

Zugriffsrecht CREATE EXTERNAL ROUTINE erteilt oder widerrufen.

**0x0001000000000000 QUIESCE\_CONNECT**

Zugriffsrecht QUIESCE\_CONNECT erteilt oder widerrufen.

**Zugehörige Referenzen:**

- „Prüfsatzaufbau für SECMAINT-Ereignisse“ auf Seite 309

## Prüfsatzaufbau für SYSADMIN-Ereignisse

Tabelle 12. Prüfsatzaufbau für SYSADMIN-Ereignisse

<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
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des AUDIT-Ereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Gültige Werte:  SYSADMIN
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.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt:  Erfolgreiches Ereignis > = 0 Fehlgeschlagenes Ereignis < 0

Tabelle 12. Prüfsatzaufbau für SYSADMIN-Ereignisse (Forts.)

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;		
NAME	FORMAT	BESCHREIBUNG
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 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 (Paket- name)	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.
Package Version (Paketversion)	VARCHAR (64)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.

**Zugehörige Konzepte:**

- „Aufbau der Datensätze der Prüffunktion (Einführung)“ auf Seite 300

**Zugehörige Referenzen:**

- „Liste der möglichen SYSADMIN-Prüfereignisse“ auf Seite 315

## Liste der möglichen SYSADMIN-Prüfereignisse

In der folgenden Liste sind mögliche SYSADMIN-Prüfereignisse aufgeführt:

Tabelle 13. 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

### Zugehörige Referenzen:

- „Prüfsatzaufbau für SYSADMIN-Ereignisse“ auf Seite 313

## Prüfsatzaufbau für VALIDATE-Ereignisse

Tabelle 14. 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.
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Authentication Type (Authentifizierungstyp)	VARCHAR(32)	Der Authentifizierungstyp beim Auftreten des Prüfereignisses.
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 14. Prüfsatzaufbau für VALIDATE-Ereignisse (Forts.)

<pre>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;</pre>		
NAME	FORMAT	BESCHREIBUNG
Package Section Number (Paketabschnittsnummer)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Version (Paketversion)	VARCHAR (64)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Plug-in Name (Plug-in-Name)	VARCHAR(32)	Der Name des Plug-ins, das beim Auftreten des Prüfereignisses verwendet wurde.

**Zugehörige Konzepte:**

- „Aufbau der Datensätze der Prüffunktion (Einführung)“ auf Seite 300

## Prüfsatzaufbau für CONTEXT-Ereignisse

Tabelle 15. Prüfsatzaufbau für CONTEXT-Ereignisse

<pre>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);</pre>		
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.
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.

Tabelle 15. Prüfsatzaufbau für CONTEXT-Ereignisse (Forts.)

<pre>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);</pre>		
NAME	FORMAT	BESCHREIBUNG
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 (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 (2M)	Text der SQL-Anweisung, falls vorhanden. Null, wenn kein Text für die SQL-Anweisung verfügbar ist.
Package Version (Paketversion)	VARCHAR (64)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.

**Zugehörige Konzepte:**

- „Aufbau der Datensätze der Prüffunktion (Einführung)“ auf Seite 300

**Zugehörige Referenzen:**

- „Liste der möglichen CONTEXT-Prüfereignisse“ auf Seite 319

## Liste der möglichen CONTEXT-Prüfereignisse

In der folgenden Liste sind mögliche CONTEXT-Prüfereignisse aufgeführt:

Tabelle 16. 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

### Zugehörige Referenzen:

- „Prüfsatzaufbau für CONTEXT-Ereignisse“ auf Seite 317

---

## Tipps und Techniken für die Prüffunktion

Beim Arbeiten mit CHECKING-Ereignissen wird meist das Feld für den Objekttyp 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 den Objekttyp 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 der Zugriffstyp „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 Datenbank von DB2<sup>®</sup> Universal Database (DB2 UDB) 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 del0xff replace into ...
```

Dadurch wird der Standardzeichenfolgebegrenzer „0xff“ außer Kraft gesetzt.

#### **Zugehörige Konzepte:**

- „Aufbau der Datensätze der Prüffunktion (Einführung)“ auf Seite 300

#### **Zugehörige Referenzen:**

- „Verwendung der Prüffunktion“ auf Seite 287



---

## Steuern der Aktivitäten der DB2 UDB-Prüffunktion

### Vorgehensweise:

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 Prüfkfiguration, die dafür sorgt, dass alle möglichen Prüfereignisse und -arten geprüft werden, eine beträchtliche Anzahl von Prüfungen generiert.

In den meisten Fällen werden Sie die Prüffunktion jedoch so konfigurieren, dass nur eine begrenzte bzw. gezielte 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

(none)

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

(none)

#### Zugehörige Konzepte:

- „Aufbau der Datensätze der Prüffunktion (Einführung)“ auf Seite 300

#### Zugehörige Referenzen:

- „Verwendung der Prüffunktion“ auf Seite 287

---

## **Teil 3. Anhänge und Schlussteil**



---

## Anhang A. Namenskonventionen

---

### Allgemeine Namenskonventionen

Für die Benennung aller Objekte und Benutzer gelten Regeln. Einige dieser Regeln sind für die Plattform spezifisch, auf der Sie arbeiten. Es gibt zum Beispiel eine Regel für die Verwendung von Groß- und Kleinbuchstaben in einem Namen.

- Auf UNIX<sup>®</sup>-Plattformen müssen Namen in Kleinbuchstaben angegeben werden.
- Auf Windows<sup>®</sup>-Plattformen sind Groß-, Kleinschreibung und gemischte Groß- und Kleinschreibung zulässig.

Wenn nichts anderes angegeben ist, dürfen alle Namen die folgenden Zeichen enthalten:

- A bis Z. In den meisten Namen werden die Zeichen A bis Z von Kleinbuchstaben in Großbuchstaben umgesetzt.
- 0 bis 9.
- ! % ( ) { } . - ^ ~ \_ (Unterstreichungszeichen) @, #, \$ und Leerzeichen.
- \ (umgekehrter Schrägstrich (Backslash)).

Namen dürfen nicht mit einer Ziffer oder dem Unterstreichungszeichen beginnen.

Für SQL reservierte Wörter dürfen nicht als Namen von Tabellen, Sichten, Spalten, Indizes oder Berechtigungs-IDs verwendet werden.

Abhängig vom verwendeten Betriebssystem und dem Ort, an dem Sie mit DB2<sup>®</sup> Universal Database (DB2 UDB) arbeiten, sind möglicherweise weitere Sonderzeichen verfügbar, die ebenfalls verwendet werden können. Diese Zeichen können funktionieren, jedoch gibt es keine Garantie, dass sie funktionieren. Daher wird empfohlen, beim Benennen von Objekten in der Datenbank keine anderen als die oben aufgelisteten Sonderzeichen zu verwenden.

Darüber hinaus müssen Sie auch die Namenskonventionen der einzelnen Objekte, die Namenskonventionen für die Workstation, die Namenskonventionen in einer NLS-Umgebung und die Namenskonventionen in einer Unicode-Umgebung beachten.

#### Zugehörige Konzepte:

- „Namenskonventionen für DB2 UDB-Objekte“ auf Seite 327
- „Namenskonventionen für Workstations“ auf Seite 332
- „Namenskonventionen für Benutzer, Benutzer-IDs und Gruppen“ auf Seite 330
- „Namenskonventionen für Objekte in zusammengesetzten Datenbanken“ auf Seite 331

---

### Namenskonventionen für DB2 UDB-Objekte

Für alle Objekte müssen die allgemeinen Namenskonventionen beachtet werden. Darüber hinaus gelten für bestimmte Objekte zusätzliche Einschränkungen, die den folgenden Tabellen zu entnehmen sind.

Tabelle 17. Namenskonventionen für Datenbanken, Aliasnamen von Datenbanken und Exemplare

Objekte	Richtlinien
<ul style="list-style-type: none"> <li>• Datenbanken</li> <li>• Aliasnamen von Datenbanken</li> <li>• Exemplare</li> </ul>	<ul style="list-style-type: none"> <li>• Datenbanknamen müssen an der Position, an der sie katalogisiert werden, eindeutig sein. Bei UNIX<sup>®</sup>-Implementierungen von DB2<sup>®</sup> Universal Database (DB2 UDB) ist diese Position ein Verzeichnispfad. Bei Windows<sup>®</sup>-Implementierungen ist sie eine logische Platte.</li> <li>• Aliasnamen der Datenbanken müssen innerhalb des Systemdatenbankverzeichnisses eindeutig sein. Beim Erstellen einer neuen Datenbank wird der Aliasname der Datenbank standardmäßig so definiert, dass er mit dem Datenbanknamen identisch ist. Daher können Sie keine Datenbank mit einem Namen erstellen, der bereits als Aliasname einer Datenbank verwendet wird, auch wenn noch keine Datenbank mit diesem Namen vorhanden ist.</li> <li>• Datenbanknamen, Aliasnamen der Datenbanken und Exemplarnamen dürfen bis zu 8 Byte lang sein.</li> <li>• Auf Systemen unter Windows NT<sup>®</sup>, Windows 2000, Windows XP und Windows Server 2003 dürfen Exemplare keine Namen haben, die als Namen von Diensten verwendet werden.</li> </ul> <p><b>Anmerkung:</b> Um mögliche Probleme zu vermeiden, empfiehlt es sich, die Sonderzeichen @, # und \$ nicht in Datenbanknamen zu verwenden, wenn die Datenbank in einer Übertragungsumgebung verwendet werden soll. Darüber hinaus sollten Sie diese Zeichen sowie Umlaute nicht benutzen, wenn Sie die Datenbank in einer anderen Sprache verwenden wollen, da diese Zeichen nicht auf allen Tastaturen in gleicher Weise verfügbar sind.</p>

Tabelle 18. Namenskonventionen für Datenbankobjekte

Objekte	Richtlinien
<ul style="list-style-type: none"> <li>• Aliasnamen</li> <li>• Pufferpools</li> <li>• Spalten</li> <li>• Ereignismonitore</li> <li>• Indizes</li> <li>• Methoden</li> <li>• Knotengruppen</li> <li>• Pakete</li> <li>• Paketversionen</li> <li>• Schemata</li> <li>• Gespeicherte Prozeduren</li> <li>• Tabellen</li> <li>• Tabellenbereiche</li> <li>• Auslöser</li> <li>• Benutzerdefinierte Funktionen</li> <li>• Benutzerdefinierte Typen</li> <li>• Sichten</li> </ul>	<p>Namen dürfen bis zu 18 Byte enthalten, <i>außer</i> in den folgenden Fällen:</p> <ul style="list-style-type: none"> <li>• Tabellennamen (einschließlich Sichtnamen, Namen von Übersichtstabellen, Aliasnamen und Korrelationsnamen) dürfen bis zu 128 Byte enthalten.</li> <li>• Spaltennamen dürfen bis zu 30 Byte enthalten.</li> <li>• Paketnamen dürfen bis zu 8 Byte enthalten.</li> <li>• Schemanamen dürfen bis zu 30 Byte enthalten.</li> <li>• Paketversionen dürfen bis zu 64 Byte enthalten</li> <li>• Objektnamen dürfen auch die folgenden Elemente enthalten: <ul style="list-style-type: none"> <li>– Gültige Zeichen mit Akzent und Umlaute (wie beispielsweise ö)</li> <li>– Mehrbytezeichen mit Ausnahme von Mehrbyteleerzeichen (in Mehrbyteumgebungen)</li> </ul> </li> <li>• Paketnamen und Paketversionen dürfen auch Punkte (.), Bindestriche (-) und Doppelpunkte (:) enthalten.</li> </ul>



Tabelle 19. Namenskonventionen für Objekte in zusammengesetzten Datenbanken

Objekte	Richtlinien
<ul style="list-style-type: none"> <li>• Funktionszuordnungen</li> <li>• Indexspezifikationen</li> <li>• Kurznamen</li> <li>• Server</li> <li>• Typenzuordnungen</li> <li>• Benutzerzuordnungen</li> <li>• Wrapper</li> </ul>	<ul style="list-style-type: none"> <li>• Kurznamen sowie Namen von Zuordnungen, Indexspezifikationen, Servern und Wrappern dürfen nicht länger als 128 Byte sein.</li> <li>• Optionen und Optionseinstellungen für Server und Kurznamen sind auf 255 Byte beschränkt.</li> <li>• Namen für Objekte in zusammengesetzten Datenbanken dürfen auch die folgenden Elemente enthalten:             <ul style="list-style-type: none"> <li>– Gültige Zeichen mit Akzent und Umlaute (wie beispielsweise ö)</li> <li>– Mehrbytezeichen mit Ausnahme von Mehrbyteleerzeichen (in Mehrbyteumgebungen)</li> </ul> </li> </ul>

**Namen von begrenzten Bezeichnern und Objekten:**

Schlüsselwörter dürfen verwendet werden. Wird ein Schlüsselwort in einem Kontext verwendet, in dem es auch als SQL-Schlüsselwort interpretiert werden kann, 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. Bei der späteren Verwendung eines solchen Objekts können jedoch Fehler auftreten. Wenn Sie zum Beispiel eine Spalte mit einem Namen erstellt haben, in dem ein Pluszeichen (+) oder ein Minuszeichen (-) vorkommt, und Sie diese Spalte später in einem Index verwenden, treten Probleme auf, wenn Sie versuchen, die Tabelle zu reorganisieren.

**Weitere Informationen zu Schemanamen:**

- Benutzerdefinierte Typen (UDTs) dürfen keine Schemanamen verwenden, die länger als 8 Byte sind.
- Die folgenden Schemanamen sind reservierte Wörter und dürfen nicht verwendet werden: SYSCAT, SYSFUN, SYSIBM, SYSSTAT.
- Um mögliche Probleme bei einer zukünftigen Migration auszuschließen, sollten Sie keine Schemanamen verwenden, die mit der Zeichenfolge SYS beginnen. Der Datenbankmanager lässt die Erstellung von Auslösern, benutzerdefinierten Typen oder benutzerdefinierten Funktionen, die einen mit SYS beginnenden Schemanamen verwenden, nicht zu.
- Es wird empfohlen, das Wort SESSION nicht als Schemanamen zu verwenden. Deklarierte temporäre Tabelle müssen durch SESSION qualifiziert werden. Daher kann es vorkommen, dass eine Anwendung eine temporäre Tabelle mit einem Namen deklariert, der mit dem einer persistenten Tabelle identisch ist. In diesem Fall kann die Anwendungslogik zu komplex werden. Vermeiden Sie die Verwendung des Schemas SESSION, außer wenn Sie mit deklarierten temporären Tabellen arbeiten.

**Zugehörige Konzepte:**

- „Allgemeine Namenskonventionen“ auf Seite 327

---

## Namen von begrenzten Bezeichnern und Objekten

Schlüsselwörter dürfen verwendet werden. Wird ein Schlüsselwort in einem Kontext verwendet, in dem es auch als SQL-Schlüsselwort interpretiert werden kann, 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. Bei der späteren Verwendung eines solchen Objekts können jedoch Fehler auftreten. Wenn Sie zum Beispiel eine Spalte mit einem Namen erstellt haben, in dem ein Pluszeichen (+) oder ein Minuszeichen (-) vorkommt, und Sie diese Spalte später in einem Index verwenden, treten Probleme auf, wenn Sie versuchen, die Tabelle zu reorganisieren.

**Zugehörige Konzepte:**

- „Allgemeine Namenskonventionen“ auf Seite 327

---

## Namenskonventionen für Benutzer, Benutzer-IDs und Gruppen

*Tabelle 20. Namenskonventionen für Benutzer, Benutzer-IDs und Gruppen*

Objekte	Richtlinien
<ul style="list-style-type: none"> <li>• Gruppennamen</li> <li>• Benutzernamen</li> <li>• Benutzer-IDs</li> </ul>	<ul style="list-style-type: none"> <li>• Gruppennamen dürfen bis zu 30 Zeichen enthalten.</li> <li>• Benutzer-IDs auf Linux- und UNIX-Systemen dürfen bis zu 8 Zeichen enthalten.</li> <li>• Benutzernamen unter Windows® dürfen bis zu 30 Zeichen enthalten. Für Windows NT®, Windows 2000, Windows XP und Windows Server 2003 gilt momentan eine praktische Begrenzung auf 20 Zeichen.</li> <li>• Wird keine Clientauthentifizierung verwendet, werden Clients unter einem anderen als einem 32-Bit-Windows-Betriebssystem, die auf Windows NT, Windows 2000, Windows XP und Windows Server 2003 zugreifen und Benutzernamen verwenden, die länger als 8 Zeichen sind, unterstützt, wenn der Benutzername und das Kennwort explizit angegeben werden.</li> <li>• Für Namen und IDs gilt Folgendes:             <ul style="list-style-type: none"> <li>– USERS, ADMINS, GUESTS, PUBLIC, LOCAL oder für SQL reservierte Wörter sind nicht zulässig.</li> <li>– Sie dürfen nicht mit IBM, SQL oder SYS beginnen.</li> <li>– Sie dürfen keine Zeichen mit Akzent oder Umlaute enthalten.</li> </ul> </li> </ul>

**Anmerkungen:**

1. Für bestimmte Betriebssysteme muss bei Benutzer-IDs und Kennwörtern die Groß-/Kleinschreibung beachtet werden. Informationen hierzu enthält die Dokumentation des Betriebssystems.
2. Die von einer erfolgreichen CONNECT- oder ATTACH-Operation zurückgegebene Berechtigungs-ID wird nach 8 Zeichen abgeschnitten. An die Berechtigungs-ID werden drei Punkte (...) angefügt, und die SQLWARN-Felder enthalten Warnungen, die auf die Abtrennung der übrigen Zeichen hinweisen.
3. Folgende Leerzeichen werden aus Benutzer-IDs und Kennwörtern entfernt.

**Zugehörige Konzepte:**

- „Allgemeine Namenskonventionen“ auf Seite 327
- „Namenskonventionen für Objekte in zusammengesetzten Datenbanken“ auf Seite 331

---

## Namenskonventionen für Objekte in zusammengesetzten Datenbanken

Tabelle 21. Namenskonventionen für Objekte in zusammengesetzten Datenbanken

Objekte	Richtlinien
<ul style="list-style-type: none"><li>• Funktionszuordnungen</li><li>• Indexspezifikationen</li><li>• Kurznamen</li><li>• Server</li><li>• Typenzuordnungen</li><li>• Benutzerzuordnungen</li><li>• Wrapper</li></ul>	<ul style="list-style-type: none"><li>• Kurznamen sowie Namen von Zuordnungen, Indexspezifikationen, Servern und Wrappern dürfen nicht länger als 128 Byte sein.</li><li>• Optionen und Optionseinstellungen für Server und Kurznamen sind auf 255 Byte beschränkt.</li><li>• Namen für Objekte in zusammengesetzten Datenbanken dürfen auch die folgenden Elemente enthalten:<ul style="list-style-type: none"><li>– Gültige Zeichen mit Akzent und Umlaute (wie beispielsweise ö)</li><li>– Mehrbytezeichen mit Ausnahme von Mehrbyteleerzeichen (in Mehrbyteumgebungen)</li></ul></li></ul>

### Zugehörige Konzepte:

- „Allgemeine Namenskonventionen“ auf Seite 327

---

## Weitere Einschränkungen und Empfehlungen zur Verwendung von Schemanamen

- Benutzerdefinierte Typen (UDTs) dürfen keine Schemanamen verwenden, die länger als 8 Byte sind.
- Die folgenden Schemanamen sind reservierte Wörter und dürfen nicht verwendet werden: SYSCAT, SYSFUN, SYSIBM, SYSSTAT.
- Um mögliche Probleme bei einer zukünftigen Migration auszuschließen, sollten Sie keine Schemanamen verwenden, die mit der Zeichenfolge SYS beginnen. Der Datenbankmanager lässt die Erstellung von Auslösern, benutzerdefinierten Typen oder benutzerdefinierten Funktionen, die einen mit SYS beginnenden Schemanamen verwenden, nicht zu.
- Es wird empfohlen, das Wort SESSION nicht als Schemanamen zu verwenden. Deklarierte temporäre Tabellen müssen durch SESSION qualifiziert werden. Daher kann es vorkommen, dass eine Anwendung eine temporäre Tabelle mit einem Namen deklariert, der mit dem einer persistenten Tabelle identisch ist. In diesem Fall kann die Anwendungslogik zu komplex werden. Vermeiden Sie die Verwendung des Schemas SESSION, außer wenn Sie mit deklarierten temporären Tabellen arbeiten.

### Zugehörige Konzepte:

- „Allgemeine Namenskonventionen“ auf Seite 327

---

## Kennwortverwaltung auf Servern

Es kann erforderlich werden, Tasks für die Kennwortverwaltung auszuführen. Da diese Tasks auf dem Server ausgeführt werden müssen und viele Benutzer nicht mit der Arbeit in der Serverumgebung vertraut sind, kann das Ausführen dieser Tasks eine schwierige Herausforderung darstellen. DB2<sup>®</sup> Universal Database (DB2 UDB) stellt eine Möglichkeit zur Verfügung, mit der Kennwörter aktualisiert und überprüft werden können, ohne sich dabei am Server befinden zu müssen.

DB2 für OS/390® Version 5 unterstützt beispielsweise diese Möglichkeit zum Ändern des Kennworts eines Benutzers. Wird die Fehlermeldung SQL1404N „Das Kennwort ist abgelaufen“ empfangen, können Sie die Anweisung CONNECT wie folgt verwenden, um das Kennwort zu ändern:

```
CONNECT TO <datenbank> USER <benutzer-id> USING <kennwort>  
NEW <new_kennwort> CONFIRM <neues_kennwort>
```

Der Dialog **Kennwort ändern** im DB2 UDB-Konfigurationsassistenten kann ebenfalls verwendet werden, um das Kennwort zu ändern.

#### Zugehörige Konzepte:

- „Allgemeine Namenskonventionen“ auf Seite 327
- „Namenskonventionen für DB2 UDB-Objekte“ auf Seite 327
- „Namenskonventionen für Workstations“ auf Seite 332
- „Namenskonventionen für Benutzer, Benutzer-IDs und Gruppen“ auf Seite 330
- „Namenskonventionen für Objekte in zusammengesetzten Datenbanken“ auf Seite 331
- „Namen von begrenzten Bezeichnern und Objekten“ auf Seite 329
- „Weitere Einschränkungen und Empfehlungen zur Verwendung von Schemanamen“ auf Seite 331

---

## Namenskonventionen für Workstations

Der *Workstationname* wird verwendet, um den NetBIOS-Namen für einen Datenbankserver, einen Datenbankclient oder DB2® Universal Database (DB2 UDB) Personal Edition anzugeben, der bzw. das sich auf der lokalen Workstation befindet. Dieser Name wird in der Konfigurationsdatei des Datenbankmanagers gespeichert. Der Workstationname wird auch als *nname* der Workstation bezeichnet.

Der von Ihnen angegebene Name muss außerdem die folgenden Merkmale aufweisen:

- Er darf aus 1 bis 8 Zeichen bestehen.
- Er darf nicht die Zeichen &, # und @ enthalten.
- Er muss im Netzwerk eindeutig sein.

In einem partitionierten Datenbanksystem gibt es nur einen Workstationnamen (*nname*), der das gesamte partitionierte Datenbanksystem darstellt; dafür verfügt jeder Knoten über einen eigenen abgeleiteten NetBIOS-Namen (*nname*), der eindeutig ist.

Der Workstationname (*nname*), der das partitionierte Datenbanksystem darstellt, ist in der Konfigurationsdatei des Datenbankmanagers für den Datenbankpartitionsserver gespeichert, der Eigner des Exemplars ist.

Jeder eindeutige Knotenname (*nname*) besteht aus einer Kombination des Workstationnamens (*nname*) und der Knotennummer.

Ist ein Knoten nicht Eigner eines Exemplars, wird der NetBIOS-Name (*nname*) wie folgt abgeleitet:

1. Das erste Zeichen des Workstationnamens (*nname*) der Exempleigermaschine wird als erstes Zeichen des NetBIOS-Namens (*nname*) für den Knoten verwendet.

2. Die nächsten 1 bis 3 Zeichen stellen die Knotennummer dar. Diese kann einen Wert aus dem Bereich 1 bis 999 annehmen.
3. Die übrigen Zeichen werden dem Workstationnamen (*nname*) der Exemplareignermaschine entnommen. Die Anzahl der übrigen Zeichen richtet sich nach der Länge des Workstationnamens (*nname*) der Exemplareignermaschine. Die Anzahl kann zwischen 0 bis 4 Zeichen betragen.

Beispiel:

Workstationname ( <i>nname</i> ) der Exemplareignermaschine	Knotennummer	Abgeleiteter NetBIOS-Name ( <i>nname</i> ) des Knotens
GEORGE	3	G3ORGE
A	7	A7
B2	94	B942
N0076543	21	N216543
GEORGE5	1	G1RGE5

Wurde der Standardname der Workstation (*nname*) während der Installation geändert, müssen die letzten 4 Zeichen des Workstationnamens (*nname*) im NetBIOS-Netz eindeutig sein, um die Wahrscheinlichkeit zu minimieren, dass ein unzulässiger NetBIOS-Name (*nname*) abgeleitet wird.

**Zugehörige Konzepte:**

- „Allgemeine Namenskonventionen“ auf Seite 327

## Namenskonventionen in einer NLS-Umgebung

Der Basiszeichensatz, der in Datenbanknamen verwendet werden kann, besteht aus den Buchstaben A - Z in Groß- und Kleinschreibung (d. h. ohne Umlaute), den Ziffern 0 - 9 und dem Unterstrichungszeichen (\_). Diese Liste wird um drei Sonderzeichen (#, @ und \$) erweitert, um die Kompatibilität mit Hostdatenbankprodukten zu gewährleisten. Verwenden Sie die Sonderzeichen #, @ und \$ in einer NLS-Umgebung mit Vorsicht, da sie nicht Teil des unveränderlichen NLS-Hostzeichensatzes (EBCDIC) sind. Abhängig von der verwendeten Codepage können auch Zeichen des erweiterten Zeichensatzes verwendet werden. Wird die Datenbank in einer Umgebung mit mehreren Codepages verwendet, muss sichergestellt werden, dass alle Codepages alle verwendeten Elemente des erweiterten Zeichensatzes unterstützen.

Beim Benennen von Datenbankobjekten (wie beispielsweise Tabellen und Sichten) Programmbezeichnungen, Hostvariablen und Cursors können auch Elemente des erweiterten Zeichensatzes (wie beispielsweise Umlaute) verwendet werden. Welche Zeichen verfügbar sind, hängt von der verwendeten Codepage ab.

**Definition des erweiterten Zeichensatzes für DBCS-Bezeichner:**

In DBCS-Umgebungen (DBCS = Doppelbytezeichensatz) besteht der erweiterte Zeichensatz aus allen Zeichen des Basiszeichensatzes sowie den folgenden Zeichen:

- Alle Doppelbytezeichen der einzelnen DBCS-Codepages mit Ausnahme des Doppelbyteleerzeichens sind gültige Buchstaben.
- Das Doppelbyteleerzeichen ist ein Sonderzeichen.

- Die in den einzelnen gemischten Codepages verfügbaren Einzelbytezeichen werden wie folgt verschiedenen Kategorien zugeordnet:

Kategorie	Gültige Codepunkte innerhalb der einzelnen gemischten Codepage
Ziffern	x30-39
Buchstaben	x23-24, x40-5A, x61-7A, xA6-DF (A6-DF nur für Codepage 932 und 942)
Sonderzeichen	Alle anderen gültigen Codepunkte für Einzelbytezeichen

#### Zugehörige Konzepte:

- „Allgemeine Namenskonventionen“ auf Seite 327
- „Namenskonventionen für DB2 UDB-Objekte“ auf Seite 327
- „Namenskonventionen für Workstations“ auf Seite 332

---

## Namenskonventionen in einer Unicode-Umgebung

In einer UCS-2-Datenbank sind alle Bezeichner in Mehrbyte-UTF-8 codiert. Daher ist es möglich, in Bezeichnern beliebige UCS-2-Zeichen zu verwenden, sofern die Verwendung eines Zeichens des erweiterten Zeichensatzes (wie beispielsweise eines Zeichens mit Akzent oder eines Mehrbytezeichens) von DB2<sup>®</sup> Universal Database (DB2 UDB) zugelassen wird.

Clients können beliebige in ihrer Umgebung unterstützte Zeichen eingeben; alle Zeichen der Bezeichner werden vom Datenbankmanager in UTF-8 umgesetzt. Bei der Verwendung von Zeichen der Landessprache in Bezeichnern für eine UCS-2-Datenbank müssen jedoch die folgenden beiden Punkte beachtet werden:

- Für jedes Zeichen, das nicht Teil des ASCII-Zeichensatzes ist, sind 2 bis 4 Byte erforderlich. Daher kann ein Bezeichner von  $n$  Byte Länge nur zwischen  $n/4$  und  $n$  Zeichen enthalten. Der genaue Wert hängt vom Verhältnis der ASCII- zu den Nicht-ASCII-Zeichen ab. Enthält der Bezeichner nur ein oder zwei Nicht-ASCII-Zeichen (beispielsweise Zeichen mit Akzent), liegt die mögliche Länge näher bei  $n$  Zeichen, während bei einem vollständig aus Nicht-ASCII-Zeichen bestehenden Bezeichner (beispielsweise in japanischer Sprache) nur  $n/4$  bis  $n/3$  Zeichen verwendet werden können.
- Sollen Bezeichner von unterschiedlichen Clientumgebungen aus eingegeben werden, müssen sie so definiert werden, dass sie nur Zeichen enthalten, die Teil der gemeinsamen Untermenge der für diese Clients verfügbaren Zeichen sind. Soll beispielsweise auf eine UCS-2-Datenbank von Umgebungen mit Latin-1, Arabisch und Japanisch aus zugegriffen werden, sollten alle Bezeichner unter normalen Umständen nur Zeichen des ASCII-Zeichensatzes enthalten.

#### Zugehörige Konzepte:

- „Allgemeine Namenskonventionen“ auf Seite 327
- „Namenskonventionen für DB2 UDB-Objekte“ auf Seite 327
- „Namenskonventionen für Workstations“ auf Seite 332

---

## Anhang B. Verwenden der automatischen Clientweiterleitung

In diesem Anhang wird die Funktionsweise der automatischen Clientweiterleitung beschrieben und erläutert, wie Sie die korrekte Funktionsweise in Ihrer Umgebung sicherstellen können.

---

### Automatische Clientweiterleitung - Beschreibung und Einrichtung

Wenn ein Server ausfällt, empfängt jeder mit diesem Server verbundene Client einen Kommunikationsfehler, der die Verbindung beendet und einen Anwendungsfehler verursacht. In Fällen, in denen die Verfügbarkeit eine wichtige Rolle spielt, sollten Sie entweder eine redundante Konfiguration oder die Möglichkeit zur Übernahme der Serverfunktion durch einen Bereitschaftsknoten implementiert haben. In beiden Fällen versucht der Clientcode von DB2<sup>®</sup> Universal Database (DB2 UDB) die Verbindung zu dem ursprünglichen Server, der jetzt auf einem Übernahmeknoten (die IP-Adresse wird ebenfalls übernommen) ausgeführt wird, oder zu einem neuen Server wiederherzustellen.

Wenn die Verbindung wiederhergestellt ist, empfängt die Anwendung eine Fehlermeldung, die sie über den Transaktionsfehler informiert, jedoch kann die Anwendung mit der nächsten Transaktion fortfahren.

Das Hauptziel der Funktion zur automatischen Clientweiterleitung besteht darin, einer DB2 UDB-Clientanwendung die Behebung eines Kommunikationsverlusts zu ermöglichen, so dass die Anwendung ihre Arbeit nach minimaler Unterbrechung fortsetzen kann. Wie der Name erkennen lässt, spielt die Weiterleitung eine zentrale Rolle bei der Unterstützung eines unterbrechungsfreien Betriebs. Die Weiterleitung ist jedoch nur möglich, wenn es einen alternativen Standort gibt, der der Clientverbindung bekannt ist.

Die Funktion zur automatischen Clientweiterleitung könnte in den folgenden konfigurierbaren Umgebungen genutzt werden:

1. Enterprise Server Edition (ESE) mit dem Data Partitioning Feature (DPF)
2. Data Propagator (DPROPR)-artige Replikation
3. High Availability Cluster Multiprocessor (HACMP)
4. High Availability Disaster Recovery (HADR)

Die automatische Clientweiterleitung funktioniert in Verbindung mit HADR, um einer Clientanwendung die Fortsetzung ihrer Arbeit bei minimaler Unterbrechung nach einer Funktionsübernahme (Failover) der Datenbank zu ermöglichen, auf die zugegriffen wird.

Um DB2 UDB die Möglichkeit zu geben, einen Verlust der Kommunikationsverbindung zu beheben, muss ein alternativer Serverstandort angegeben werden, bevor der Verlust der Kommunikation auftritt. Der alternative Server kann mit dem Befehl `UPDATE ALTERNATE SERVER FOR DATABASE` angegeben werden. Um sicherzustellen, dass der angegebene alternative Serverstandort für alle Clients gilt, muss der alternative Serverstandort auf der Serverseite angegeben werden. Der alternative Server wird ignoriert, wenn er im Clientexemplar angegeben wird.

Nehmen Sie zum Beispiel an, dass sich eine Datenbank auf dem Knoten „N1“ (mit dem Hostnamen XXX und der Portnummer YYY) befindet. Der Datenbank-

administrator möchte den alternativen Serverstandort mit dem Hostnamen = AAA und der Portnummer 123 angeben. Dazu würde der Datenbankadministrator den folgenden Befehl auf Knoten N1 (im Serverexemplar) ausführen:

```
db2 update alternate server for database db2 using hostname AAA port 123
```

Wenn der Datenbankadministrator den alternativen Serverstandort für eine bestimmte Datenbank im Serverexemplar angegeben hat, wird dieser beim Aufbau einer Verbindung an den Client zurückgegeben. Wenn die Kommunikation aus einem irgendeinem Grund verloren geht, ist der DB2 UDB-Clientcode in der Lage, die Verbindung mit Hilfe der Informationen über den alternativen Server, die vom Server zurückgegeben wurden, wiederherzustellen. Wenn der alternative Server nicht auf dem Server angegeben ist, wird die automatische Clientweiterleitung nicht verarbeitet.

Im Allgemeinen wird bei angegebenem alternativen Server die automatische Clientweiterleitung aktiviert, wenn ein Kommunikationsfehler (SQLCODE-Wert -30081) oder ein SQLCODE-Wert -1224 erkannt wird. In einer HADR-Umgebung (High Availability Disaster Recovery) wird sie außerdem aktiviert, wenn SQLCODE-Wert -1776 vom HADR-Bereitschaftsserver zurückgegeben wird.

**Zugehörige Konzepte:**

- „Automatische Clientweiterleitung - Einschränkungen“ auf Seite 336

**Zugehörige Referenzen:**

- „Automatische Clientweiterleitung - Beispiele“ auf Seite 338

---

## Automatische Clientweiterleitung - Einschränkungen

Die Verwendung der Funktion zur automatischen Clientweiterleitung unterliegt einigen Einschränkungen:

- Die automatische Clientweiterleitung wird nur unterstützt, wenn TCP/IP als Kommunikationsprotokoll zur Herstellung der Verbindung zum DB2<sup>®</sup> Universal Database-Server (DB2 UDB-Server) bzw. zum DB2 Connect<sup>™</sup>-Server verwendet wird. Das bedeutet, dass die Funktion zur automatischen Clientweiterleitung nicht aktiviert wird, wenn die Verbindung mit einem anderen Protokoll als TCP/IP arbeitet. Auch wenn DB2 UDB mit Loopback-Funktion konfiguriert ist, muss das Kommunikationsprotokoll TCP/IP verwendet werden, um die Funktion der automatischen Clientweiterleitung nutzen zu können.
- Wenn die Verbindung zum alternativen Serverstandort wiederhergestellt wird, wird jede neue Verbindung zu dem gleichen Aliasnamen der Datenbank zum alternativen Serverstandort hergestellt. Wenn Sie wollen, dass neue Verbindungen zum ursprünglichen Standort hergestellt werden, nachdem das Problem am ursprünglichen Standort behoben ist, steht eine Reihe von Optionen dazu zur Auswahl:
  - Sie müssen den alternativen Server offline nehmen, so dass die Verbindungen zum ursprünglichen Server wiederhergestellt werden. (Dies setzt voraus, dass der ursprüngliche Server mit dem Befehl UPDATE ALTERNATE SERVER katalogisiert wurde, so dass er als alternativer Standort für den alternativen Server definiert ist.)
  - Sie könnten einen neuen Aliasnamen der Datenbank katalogisieren, der von den neuen Verbindungen zu verwenden ist.
  - Sie könnten den Datenbankeintrag entkatalogisieren und erneut katalogisieren.



- Unter DB2 UDB für Linux, UNIX® und Windows®-Betriebssysteme wird die Funktion zur automatischen Clientweiterleitung sowohl auf der Client- als auch auf der Serverseite unterstützt. Andere DB2 UDB-Familien unterstützen diese Funktion zurzeit nicht.
- Eine SYSPLEX-Konfiguration zur gemeinsamen Datennutzung unter DB2 UDB für z/OS™ kann eine Liste von Servern zurückgeben, die zur Herstellung einer Verbindung verfügbar sind. Allerdings wird die Liste nur im Arbeitsspeicher verwaltet, wenn ein Kommunikationsfehler auftritt. In diesem Fall verwendet der DB2 UDB-Client die Liste, um den Standort des entsprechenden alternativen Servers zum Verbindungsaufbau zu bestimmen.
- Der DB2 UDB-Server, der auf dem alternativen Server installiert ist, muss dieselbe Version wie der ursprüngliche DB2 UDB-Host-Server haben (jedoch kann die FixPak-Version höher sein).
- Unabhängig davon, ob Sie die Berechtigung zum Aktualisieren des Datenbankverzeichnisses auf dem Clientsystem haben, werden die Informationen über den alternativen Server immer im Arbeitsspeicher verwaltet. Das heißt mit anderen Worten, dass andere Anwendungen, wenn Sie keine Berechtigung zum Aktualisieren des Datenbankverzeichnisses hatten (oder weil es sich um ein schreibgeschütztes Datenbankverzeichnis handelt), nicht in der Lage sind, den alternativen Server zu bestimmen und zu verwenden, weil der Arbeitsspeicher nicht von Anwendungen gemeinsam benutzt wird.
- Für alle alternativen Standorte wird die gleiche Authentifizierung ausgeführt. Das bedeutet, dass der Client keine neue Datenbankverbindung herstellen kann, wenn der alternative Standort mit einem anderen Authentifizierungstyp als der ursprüngliche Standort arbeitet.
- Wenn es zu einem Kommunikationsausfall kommt, gehen alle Sitzungsressourcen wie zum Beispiel globale temporäre Tabellen, Identitätswerte, Sequenzen, Cursor, Serveroptionen (SET SERVER OPTION) zur Verarbeitung in zusammengesetzten Umgebungen und Sonderregister verloren. Die Anwendung ist dafür zuständig, die Sitzungsressourcen neu einzurichten, um die Verarbeitung fortzusetzen. Sie brauchen nach Wiederherstellung der Verbindung keine der Anweisungen für Sonderregister auszuführen, weil DB2 UDB die Anweisungen für Sonderregister automatisch noch einmal nachvollzieht, die vor dem Verbindungsfehler ausgeführt wurden. Allerdings werden einige Sonderregister nicht neu eingestellt. Dies sind:
  - SET ENCRYPTPW
  - SET EVENT MONITOR STATE
  - SET SESSION AUTHORIZATION
  - SET TRANSFORM GROUP

**Anmerkung:** Wenn der Client mit CLI, JCC Type 2- oder Type 4-Treibern, arbeitet, werden nach der Wiederherstellung der Verbindung die SQL-Anweisungen, die mit dem ursprünglichen Server bereits vorbereitet wurden, implizit mit dem neuen Server erneut vorbereitet. Eingebettete SQL-Routinen (z. B. SQC- oder SQX-Anwendungen) werden jedoch nicht erneut vorbereitet.

Eine alternative Methode zur automatischen Clientweiterleitung ist die Verwendung des DNS-Eintrags zur Angabe einer alternativen IP-Adresse für einen DNS-Eintrag. Die Methode besteht darin, eine zweite IP-Adresse (einen alternativen Serverstandort) in dem DNS-Eintrag anzugeben, so dass der Client zwar keinen alternativen Server kennt, DB2 UDB jedoch beim Verbindungsaufbau die IP-Adressen für den DNS-Eintrag abwechselnd verwendet.

### Zugehörige Konzepte:

- „Implementierung der automatischen Clientweiterleitung“ auf Seite 79
- „Automatische Clientweiterleitung - Beschreibung und Einrichtung“ auf Seite 335

### Zugehörige Tasks:

- „Angabe eines alternativen Servers für eine Datenbank“ auf Seite 75

### Zugehörige Referenzen:

- „Automatische Clientweiterleitung - Beispiele“ auf Seite 338

---

## Automatische Clientweiterleitung - Beispiele

Das folgende Beispiel zeigt eine automatische Clientweiterleitung für eine Clientanwendung (nur in Pseudocode dargestellt):

```
int checkpoint = 0;

check_sqlca(unsigned char *str, struct sqlca *sqlca)
{
    if (sqlca->sqlcode == -30081)
    {
        // Bei Kommunikationsverlust Anwendung sofort beenden
        exit(1);
    }
    else
    {
        // Fehler ausgeben
        printf(...);
    }

    if (sqlca->sqlcode == -30108)
    {
        // Bei erneutem Verbindungsaufbau fehlgeschlagene Transaktion wiederholen
        if (checkpoint == 0)
        {
            goto checkpt0;
        }
        else if (checkpoint == 1)
        {
            goto checkpt1;
        }
        else if (checkpoint == 2)
        {
            goto checkpt2;
        }
        ....
        exit;
    }
}

main()
{
    connect to mydb;
    check_sqlca("connect failed", &sqlca);

    checkpt0:
    EXEC SQL set current schema XXX;
    check_sqlca("set current schema XXX failed", &sqlca);

    EXEC SQL create table t1...;
    check_sqlca("create table t1 failed", &sqlca);
}
```

```

EXEC SQL commit;
check_sqlca("commit failed", &sqlca);

if (sqlca.sqlcode == 0)
{
    checkpoint = 1;
}

checkpt1:
EXEC SQL set current schema YYY;
check_sqlca("set current schema YYY failed", &sqlca);

EXEC SQL create table t2...;
check_sqlca("create table t2 failed", &sqlca);

EXEC SQL commit;
check_sqlca("commit failed", &sqlca);

if (sqlca.sqlcode == 0)
{
    checkpoint = 2;
}
...
}

```

Auf dem Clientsystem wird die Datenbank mit dem Namen „mydb“ katalogisiert, die auf einen Knoten „hornet“ verweist, wobei „hornet“ ebenfalls im Knotenverzeichnis (Hostname „hornet“ mit Portnummer 456) katalogisiert wird.

### Beispiel 1 (mit einer Nicht-HADR-Datenbank)

Auf dem Server „hornet“ (Hostname ist gleich „hornet“ mit einer Portnummer) wird eine Datenbank mit dem Namen „mydb“ erstellt. Darüber hinaus wird die Datenbank „mydb“ auch auf dem alternativen Server (Hostname „montero“ mit Portnummer 456) erstellt. Sie müssen außerdem den alternativen Server für die Datenbank „mydb“ auf dem Server „hornet“ wie folgt aktualisieren:

```
db2 update alternate server for database mydb using hostname montero port 456
```

In der obigen Beispielanwendung und bei nicht eingerichteter Funktion zur automatischen Clientweiterleitung wird die Anwendung im Fall eines Kommunikationsfehlers in der Anweisung `create table t1` beendet. Bei eingerichteter Funktion zur automatischen Clientweiterleitung versucht DB2 UDB die Verbindung zum Host „hornet“ (mit Port 456) erneut herzustellen. Wenn dieser noch nicht wieder funktioniert, versucht DB2 UDB den alternativen Serverstandort (Hostname „montero“ mit Port 456). Unter der Voraussetzung, dass es keinen Kommunikationsfehler in der Verbindung zum alternativen Serverstandort gibt, kann die Anwendung anschließend mit der Ausführung nachfolgender Anweisungen fortfahren (und die fehlgeschlagene Transaktion erneut ausführen).

### Beispiel 2 (mit einer HADR-Datenbank)

Auf dem Server „hornet“ (Hostname ist gleich „hornet“ mit einer Portnummer) wird die Primärdatenbank „mydb“ erstellt. Eine Sekundärdatenbank wird außerdem auf Host „montero“ mit Port 456 erstellt. Informationen zur Konfiguration von HADR für die Primär- und die Sekundärdatenbank finden Sie in *Datenwiederherstellung und hohe Verfügbarkeit Handbuch und Referenz*. Sie müssen außerdem den alternativen Server für die Datenbank „mydb“ wie folgt aktualisieren:

```
db2 update alternate server for database mydb using hostname montero port 456
```

| In der obigen Beispielanwendung und bei nicht eingerichteter Funktion zur auto-  
| matischen Clientweiterleitung wird die Anwendung im Fall eines Kommunikati-  
| onsfehlers in der Anweisung create table t1 beendet. Bei eingerichteter Funktion  
| zur automatischen Clientweiterleitung versucht DB2 UDB die Verbindung zum  
| Host „hornet“ (mit Port 456) erneut herzustellen. Wenn dieser noch nicht wieder  
| funktioniert, versucht DB2 UDB den alternativen Serverstandort (Hostname „mon-  
| tero“ mit Port 456). Unter der Voraussetzung, dass es keinen Kommunikations-  
| fehler in der Verbindung zum alternativen Serverstandort gibt, kann die Anwen-  
| dung anschließend mit der Ausführung nachfolgender Anweisungen fortfahren  
| (und die fehlgeschlagene Transaktion erneut ausführen).

| **Zugehörige Konzepte:**

- | • „Automatische Clientweiterleitung - Beschreibung und Einrichtung“ auf Seite  
| 335

| **Zugehörige Tasks:**

- | • „Angabe eines alternativen Servers für eine Datenbank“ auf Seite 75

---

## Anhang C. Verwenden der LDAP-Verzeichnisservices (Lightweight Directory Access Protocol)

---

### Einführung in Lightweight Directory Access Protocol (LDAP)

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- und Serverzugriff auf diese Ressourcen bereit. Jedes Datenbankserverexemplar veröffentlicht Informationen über seine Existenz auf einem LDAP-Server und stellt dem LDAP-Verzeichnis Datenbankinformationen zur Verfügung, 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. Clientanwendungen durchsuchen das LDAP-Verzeichnis nach den erforderlichen Informationen für die Herstellung der Verbindung zur Datenbank.

Es gibt einen Cachingmechanismus, so dass der Client den LDAP-Verzeichnisserver nur einmal durchsuchen muss. Sobald die Informationen aus dem LDAP-Verzeichnisserver abgerufen sind, werden sie auf der lokalen Maschine nach Maßgabe der Werte des Konfigurationsparameters *dir\_cache* des Datenbankmanagers und der Registriervariablen DB2LDAPCACHE gespeichert bzw. im Cache abgelegt. Der Konfigurationsparameter *dir\_cache* des Datenbankmanagers dient zum Speichern von Datenbank-, Knoten- und DCS-Verzeichnisdateien in einem Speichercache. Der Verzeichniscache wird von einer Anwendung genutzt, bis die Anwendung geschlossen wird. Die Registriervariable DB2LDAPCACHE dient zum Speichern von Datenbank-, Knoten- und DCS-Verzeichnisdateien in einem lokalen Plattencache.

- Sind DB2LDAPCACHE=NO und *dir\_cache*=NO definiert, werden die Informationen immer aus LDAP gelesen.
- Sind DB2LDAPCACHE=NO und *dir\_cache*=YES definiert, werden die Informationen einmal aus LDAP gelesen und in den DB2®-Cache eingefügt.
- Ist DB2LDAPCACHE=YES oder diese Registriervariable überhaupt nicht definiert, werden die Informationen einmal aus dem LDAP gelesen und in den Cache für lokale Datenbank-, Knoten- und DCS-Verzeichnisse gestellt.

**Anmerkung:** Die Registriervariable DB2LDAPCACHE gilt nur für die Datenbank- und die Knotenverzeichnisse.

#### Zugehörige Konzepte:

- „LDAP-Verzeichnisservice (Lightweight Directory Access Protocol)“ auf Seite 77
- „Unterstützung für Active Directory“ auf Seite 343
- „LDAP-Unterstützung und DB2 Connect“ auf Seite 359
- „Sicherheitsaspekte in einer LDAP-Umgebung“ auf Seite 359
- „Sicherheitsaspekte für Active Directory“ auf Seite 360
- „Registrierdatenbank- und Umgebungsvariablen von DB2“ in *Systemverwaltung: Optimierung*

- „Erweitern des LDAP-Verzeichnisseschemas mit DB2-Objektklassen und -Attributen“ auf Seite 361

**Zugehörige Tasks:**

- „Konfigurieren von DB2 zur Verwendung von Active Directory“ auf Seite 344
- „Konfigurieren von DB2 in einer IBM LDAP-Umgebung“ auf Seite 345
- „Erstellen eines LDAP-Benutzers“ auf Seite 346
- „Konfigurieren des LDAP-Benutzers für DB2-Anwendungen“ auf Seite 347
- „Registrierung von DB2-Servern nach der Installation“ auf Seite 347
- „Aktualisieren der Protokollinformationen für den DB2-Server“ auf Seite 349
- „Katalogisieren eines Knoten-Aliasnamens für ATTACH“ auf Seite 350
- „Zurücknehmen der Registrierung des DB2-Servers“ auf Seite 351
- „Registrierung von Datenbanken im LDAP-Verzeichnis“ auf Seite 351
- „Herstellen einer ATTACH-Verbindung zu einem fernen Server in der LDAP-Umgebung“ auf Seite 352
- „Zurücknehmen der Registrierung der Datenbank aus dem LDAP-Verzeichnis“ auf Seite 353
- „Aktualisieren der LDAP-Einträge in lokalen Datenbank- und Knotenverzeichnissen“ auf Seite 353
- „Durchsuchen der LDAP-Verzeichnispaktionen oder -Domänen“ auf Seite 354
- „Registrieren von Hostdatenbanken in LDAP“ auf Seite 355
- „Definieren von DB2-Registriervariablen auf Benutzerebene in der LDAP-Umgebung“ auf Seite 357
- „Aktivieren der LDAP-Unterstützung nach Abschluss der Installation“ auf Seite 357
- „Inaktivieren der LDAP-Unterstützung“ auf Seite 358
- „Erweitern des Verzeichnisseschemas für Active Directory“ auf Seite 361

**Zugehörige Referenzen:**

- „Unterstützte LDAP-Client- und -Serverkonfigurationen“ auf Seite 342
- „DB2-Objekte im Active Directory“ auf Seite 363
- „Von DB2 verwendete LDAP-Objektklassen und Attribute“ auf Seite 369
- „Netscape-Unterstützung und Attributdefinitionen für das LDAP-Verzeichnis“ auf Seite 363

---

## Unterstützte LDAP-Client- und -Serverkonfigurationen

Die folgende Tabelle enthält eine Übersicht zu den unterstützten LDAP-Client- und -Serverkonfigurationen:

*Tabelle 22. Unterstützte LDAP-Client- und -Serverkonfigurationen*

	IBM SecureWay Directory	Microsoft Active Directory	Netscape LDAP-Server
IBM LDAP-Client	Unterstützt	Unterstützt	Unterstützt
Microsoft LDAP/ADSI-Client	Unterstützt	Unterstützt	Unterstützt

IBM SecureWay Directory Version 3.1 ist ein LDAP-Server der Version 3, der für Windows NT, AIX, Solaris Operating Environment und HP-UX zur Verfügung steht. IBM SecureWay Directory wird als Teil des Basisbetriebssystems für AIX und iSeries (AS/400) sowie mit OS/390 Security Server geliefert.

Die 32-Bit-Version von DB2 Universal Database™ (DB2 UDB) unterstützt den IBM LDAP-Client unter AIX, Solaris Operating Environment, HP-UX 11.11, Windows, Linux IA32 und Linux/390.

Microsoft Active Directory ist ein LDAP-Server der Version 3 und steht als Teil des Betriebssystems Windows 2000 Server zur Verfügung.

Der Microsoft LDAP-Client ist in das Windows-Betriebssystem integriert.

Bei der Ausführung unter Windows-Betriebssystemen unterstützt DB2 die Verwendung des IBM LDAP-Clients oder des Microsoft-LDAP-Clients, um auf IBM SecureWay Directory Server zuzugreifen. Zur expliziten Auswahl des IBM LDAP-Clients verwenden Sie den Befehl **db2set**, mit dem die Registriervariable **DB2LDAP\_CLIENT\_PROVIDER** auf den Wert „IBM“ gesetzt werden kann.

#### Zugehörige Konzepte:

- „Einführung in Lightweight Directory Access Protocol (LDAP)“ auf Seite 341
- „Unterstützung für Active Directory“ auf Seite 343

---

## Unterstützung für Active Directory

DB2® Universal Database (DB2 UDB) nutzt Windows Active Directory folgendermaßen:

1. Die DB2 UDB-Datenbankserver werden in Active Directory als **ibm\_db2Node**-Objekte bereitgestellt. Die Objektklasse **ibm\_db2Node** ist eine Unterklasse der Objektklasse **ServiceConnectionPoint** (SCP). Jedes **ibm\_db2Node**-Objekt enthält Protokollkonfigurationsdaten, die es Clientanwendungen ermöglichen, eine Verbindung zum DB2 UDB-Datenbankserver herzustellen. Beim Erstellen einer neuen Datenbank wird diese in Active Directory als **ibm\_db2Database**-Objekt unter dem **ibm\_db2Node**-Objekt bereitgestellt.
2. Beim Herstellen einer Verbindung zu einer fernen Datenbank fragt der DB2-Client Windows 2000 Active Directory über die LDAP-Schnittstelle nach dem **ibm\_db2Database**-Objekt ab. Die Daten zur Protokollkommunikation zur Herstellung der Verbindung zum Datenbankserver (Bindeinformationen) werden aus dem **ibm\_db2Node**-Objekt abgerufen, unter dem das **ibm\_db2Database**-Objekt erstellt ist.

Eigenschaftenseiten für die **ibm\_db2Node**- und **ibm\_db2Database**-Objekte können über das Snap-in *Active Directory-Benutzer und -Computer* der Managementkonsole (MMC) auf einem Domänencontroller angezeigt werden. Zur Einrichtung der Eigenschaftenseite führen Sie den Befehl **regsvr32** wie folgt aus, um die Eigenschaftenseiten für die DB2-Objekte zu registrieren:

```
regsvr32 %DB2PATH%\bin\db2ads.dll
```

Sie können die Objekte über das Snap-in *Active Directory-Benutzer und -Computer* der Managementkonsole (MMC) auf einem Domänencontroller anzeigen. Auf dieses Verwaltungstool können Sie über **Start**—> **Programme**—> **Verwaltung**—> **Active Directory-Benutzer und -Computer** zugreifen.

**Anmerkung:** Sie müssen *Benutzer, Gruppen und Computer als Container* im Menü **Ansicht** auswählen, um die DB2-Objekte unterhalb der Computerobjekte anzuzeigen.

**Anmerkung:** Wenn DB2 UDB nicht auf dem Domänencontroller installiert ist, können Sie die Eigenschaftenseiten von DB2 UDB-Objekten anzeigen, indem Sie die Datei `db2ads.dll` aus dem Verzeichnis `%DB2PATH%\bin` und die Ressourcen-DLL-Datei `db2adsr.dll` aus dem Verzeichnis `%DB2PATH%\msg\locale-name` in ein lokales Verzeichnis auf dem Domänencontroller kopieren. (Das Verzeichnis, in dem Sie diese beiden kopierten Dateien ablegen, muss eines der Verzeichnisse sein, die in der Benutzer-/Systemumgebungsvariablen `PATH` definiert sind. Anschließend führen Sie den Befehl `regsvr32` in dem lokalen Verzeichnis aus, um die DLL-Datei zu registrieren.

**Zugehörige Konzepte:**

- „Sicherheitsaspekte für Active Directory“ auf Seite 360

**Zugehörige Tasks:**

- „Konfigurieren von DB2 zur Verwendung von Active Directory“ auf Seite 344
- „Erweitern des Verzeichnisschemas für Active Directory“ auf Seite 361

**Zugehörige Referenzen:**

- „DB2-Objekte im Active Directory“ auf Seite 363

---

## Konfigurieren von DB2 zur Verwendung von Active Directory

**Vorgehensweise:**

Um auf Windows Active Directory zugreifen zu können, müssen die folgenden Bedingungen erfüllt sein:

1. Die Maschine, auf der DB2 Universal Database™ (DB2 UDB) ausgeführt wird, muss zu einer Windows 2000- oder Windows Server 2003-Domäne gehören.
2. Der Microsoft-LDAP-Client wurde installiert. Der Microsoft®-LDAP-Client ist Bestandteil der Betriebssysteme Windows 2000, Windows XP und Windows Server 2003. Bei Windows 98, Windows NT oder Windows Me müssen Sie sicherstellen, dass die Active Directory-Clienterweiterung (`wldap32.dll`) unter dem Systemverzeichnis vorhanden ist.
3. Aktivieren Sie die LDAP-Unterstützung. Bei Windows 2000, Windows XP oder Windows Server 2003 wird die LDAP-Unterstützung durch das Installationsprogramm aktiviert. Bei Windows 98, Windows NT oder Windows Me müssen Sie LDAP explizit aktivieren. Setzen Sie dazu mit dem Befehl `db2set` die Registervariable `DB2_ENABLE_LDAP` auf den Wert „YES“.
4. Melden Sie sich während der Ausführung von DB2 UDB mit einem Domänenbenutzerkonto an, um Informationen aus Active Directory zu lesen.

**Zugehörige Konzepte:**

- „Unterstützung für Active Directory“ auf Seite 343
- „Registrierdatenbank- und Umgebungsvariablen von DB2“ in *Systemverwaltung: Optimierung*

**Zugehörige Tasks:**

- „Konfigurieren des LDAP-Benutzers für DB2-Anwendungen“ auf Seite 347



---

## Konfigurieren von DB2 in einer IBM LDAP-Umgebung

### Vorgehensweise:

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 oder Windows NT müssen Sie LDAP explizit aktivieren. Setzen Sie dazu die Registryvariable `DB2_ENABLE_LDAP` mit dem Befehl `db2set` auf den Wert „YES“. Unter allen Windows-Betriebssystemen ist der LDAP-Client von Microsoft die Standardeinstellung. Wenn Sie den IBM LDAP-Client verwenden wollen, müssen Sie die Registryvariable `DB2LDAP_CLIENT_PROVIDER` mit Hilfe des Befehls `db2set` auf den Wert „IBM“ setzen.
- TCP/IP-Hostnamen und Portnummer 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=<hostname[:port]>
```

Dabei steht `hostname` für den TCP/IP-Hostnamen des LDAP-Servers und `[:port]` für die zugehörige Portnummer. Wird keine Portnummer angegeben, verwendet DB2 den LDAP-Standardport (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 dazu den Befehl `DB2SET` folgendermaßen:

```
db2set DB2LDAP_BASEDN=<baseDN>
```

Dabei 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.

### Zugehörige Konzepte:

- „Registrierdatenbank- und Umgebungsvariablen von DB2“ in *Systemverwaltung: Optimierung*

### Zugehörige Tasks:

- „Konfigurieren von DB2 zur Verwendung von Active Directory“ auf Seite 344
- „Erstellen eines LDAP-Benutzers“ auf Seite 346

### Zugehörige Referenzen:

- „db2set - DB2 Profile Registry Command“ in *Command Reference*

---

## Erstellen eines LDAP-Benutzers

### Vorgehensweise:

DB2 unterstützt das Definieren von DB2-Registriervariablen 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 dadurch 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 hat 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: kennwort
telephonenumber: 1-416-123-4567
facsimiletelephonenumber: 1-416-123-4568
title: Software Developer
```

Das folgende Beispiel zeigt den Befehl LDIF zum Importieren einer LDIF-Datei mit dem IBM LDIF-Importdienstprogramm:

```
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 ACL-Zugriffsberechtigungen zu erteilen.

### Zugehörige Tasks:

- „Konfigurieren von DB2 in einer IBM LDAP-Umgebung“ auf Seite 345
- „Konfigurieren des LDAP-Benutzers für DB2-Anwendungen“ auf Seite 347

---

## Konfigurieren des LDAP-Benutzers für DB2-Anwendungen

### Vorgehensweise:

Bei der Verwendung des Microsoft LDAP-Clients stimmt der LDAP-Benutzer mit dem Benutzerkonto des Betriebssystems überein. Beim Arbeiten mit dem IBM LDAP-Client müssen Sie jedoch vor der Verwendung von DB2 den registrierten LDAP-Benutzernamen (DN) und das Kennwort für den aktuell angemeldeten Benutzer konfigurieren. Verwenden Sie hierzu das Dienstprogramm `db2ldcfg`:

```
db2ldcfg -u <benutzerDN> -w <kennwort> -> Benutzer-DN u. Kennwort definieren
-r                                           -> Benutzer-DN u. Kennwort löschen
```

### Beispiel:

```
db2ldcfg -u "cn=Mary Burnnet,ou=DB2 UDB Development,ou=Toronto,o=ibm,c=ca"
-w kennwort
```

### Zugehörige Tasks:

- „Erstellen eines LDAP-Benutzers“ auf Seite 346

### Zugehörige Referenzen:

- „db2ldcfg - Configure LDAP Environment Command“ in *Command Reference*

---

## Registrierung von DB2-Servern nach der Installation

### Vorgehensweise:

Jedes DB2-Serverexemplar muss in LDAP registriert werden, um die Protokollkonfigurationsdaten zu veröffentlichen, die von den Clientanwendung 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 Clientanwendungen 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 Active Directory automatisch mit den folgenden Informationen registriert:

```
Knotenname: TCP/IP-Hostname
Protokolltyp: TCP/IP
```

Wenn die Länge des TCP/IP-Hostnamens 8 Zeichen überschreitet, wird dieser auf 8 Zeichen abgeschnitten.

Der Befehl **REGISTER** sieht wie folgt aus:

```
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 Enterprise Server Edition, das mehrere physische Maschinen umfasst, muss der Befehl **REGISTER** für

jede Maschine einmal aufgerufen werden. Verwenden Sie den Befehl **rah**, um den Befehl **REGISTER** an alle 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 Hostnamen jeder Maschine anstelle von `ldap_knotenname` verwenden. Zum Beispiel:

```
rah ">DB2 REGISTER DB2 SERVER IN LDAP AS <> PROTOCOL TCPIP"
```

„<>“ wird durch den Hostnamen jeder Maschine ersetzt, auf der der Befehl **rah** ausgeführt wird. Für den seltenen Fall, dass mehrere Exemplare von DB2 Universal Database Enterprise Server Edition vorhanden sind, kann die Kombination aus Exemplar und Hostindex 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 <hostname>
svcname <tcpip_servicename>
remote <remote_computername>
instance <exemplarname>
```

Für den Computernamen gilt folgende Konvention:

- Wenn TCP/IP konfiguriert ist, muss der Systemname mit dem TCP/IP-Hostnamen übereinstimmen.
- Wenn APPN konfiguriert ist, müssen Sie als Computernamen den Namen der Partner-LU verwenden.

Beim Betrieb 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.

#### Zugehörige Konzepte:

- „Übersicht über die Befehle `rah` und `db2_all`“ auf Seite 381

#### Zugehörige Tasks:

- „Aktualisieren der Protokollinformationen für den DB2-Server“ auf Seite 349
- „Katalogisieren eines Knoten-Aliasnamens für ATTACH“ auf Seite 350
- „Zurücknehmen der Registrierung des DB2-Servers“ auf Seite 351
- „Herstellen einer ATTACH-Verbindung zu einem fernen Server in der LDAP-Umgebung“ auf Seite 352

**Zugehörige Referenzen:**

- „REGISTER Command“ in *Command Reference*
- „CATALOG LDAP NODE Command“ in *Command Reference*

---

## Aktualisieren der Protokollinformationen für den DB2-Server

**Vorgehensweise:**

Die DB2-Serverinformationen in LDAP müssen immer aktuell sein. Änderungen an den Protokollkonfigurationsparametern oder der Server-Netzwerkadresse beispielsweise machen eine LDAP-Aktualisierung erforderlich.

Aktualisieren Sie den DB2-Server in LDAP auf der lokalen Maschine mit dem folgenden Befehl:

```
db2 update ldap ...
```

Folgende Protokollkonfigurationsparameter können zum Beispiel aktualisiert werden:

- Die Parameter für TCP/IP-Hostname und Servicename oder Portnummer
- APPC-Protokollinformationen wie TP-Name, Partner-LU oder Modus
- NetBIOS-Workstation-Name

Zum Aktualisieren der Protokollkonfigurationsparameter eines fernen DB2-Servers verwenden Sie den Befehl UPDATE LDAP mit der Klausel node:

```
db2 update ldap
  node <knotenname>
  hostname <hostname>
  svcename <tcpip_servicename>
```

**Zugehörige Tasks:**

- „Registrierung von DB2-Servern nach der Installation“ auf Seite 347
- „Katalogisieren eines Knoten-Aliasnamens für ATTACH“ auf Seite 350
- „Herstellen einer ATTACH-Verbindung zu einem fernen Server in der LDAP-Umgebung“ auf Seite 352

**Zugehörige Referenzen:**

- „UPDATE LDAP NODE Command“ in *Command Reference*

---

## Weiterleiten von LDAP-Clients an andere Server

Die Möglichkeit zur Weiterleitung von Clients steht ebenso wie bei Systemausfällen auch beim Einsatz von LDAP zur Verfügung.

**Voraussetzungen:**

Die Registriervariable DB2\_ENABLE\_LDAP ist auf den Wert „Yes“ gesetzt.

**Vorgehensweise:**

In einer LDAP-Umgebung werden sämtliche Datenbank- und Knotenverzeichnisdaten auf einem LDAP-Server verwaltet. Der Client ruft Daten aus dem LDAP-

Verzeichnis ab. Diese Daten werden in den lokalen Datenbank- und Knotenverzeichnissen aktualisiert, wenn die Registriervariable DB2LDAPCACHE auf den Wert „Yes“ gesetzt ist.

Verwenden Sie den Befehl UPDATE ALTERNATE SERVER FOR LDAP DATABASE, um den alternativen Server für eine Datenbank zu definieren, die DB2 Universal Database™ (DB2 UDB) in LDAP darstellt.

Wenn die Daten über diesen alternativen Server eingerichtet sind, werden sie an den Client zurückgegeben, wenn eine Verbindung hergestellt wird.

**Anmerkung:** Wenn Sie den Befehl UPDATE ALTERNATE SERVER FOR DATABASE (beachten Sie, dass es hier nicht „FOR LDAP DATABASE“ heißt) auf dem Serverexemplar eingeben oder wenn die LDAP-Unterstützung auf dem Server aktiviert (DB2\_ENABLE\_LDAP=Yes) ist und die LDAP-Benutzer-ID und das Kennwort im Cache gespeichert werden (der Befehl **db2ldcfg** wurde zuvor ausgeführt), dann wird der alternative Server für die Datenbank automatisch bzw. implizit auf dem LDAP-Server aktualisiert. Dies funktioniert genau so, wie bei einer expliziten Eingabe des Befehls UPDATE ALTERNATE SERVER FOR LDAP DATABASE.

#### Zugehörige Konzepte:

- „Implementierung der automatischen Clientweiterleitung“ auf Seite 79
- „Automatische Clientweiterleitung - Beschreibung und Einrichtung“ auf Seite 335

---

## Katalogisieren eines Knoten-Aliasnamens für ATTACH

#### Vorgehensweise:

Beim Registrieren eines DB2-Servers in LDAP muss ein Knotenname für den Server 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 sieht etwa wie folgt aus:

```
db2 catalog ldap node <ldap_knotenname>  
as <neuer_aliasname>
```

Zum Entfernen eines LDAP-Knotens aus dem Katalog verwenden Sie den Befehl UNCATALOG LDAP NODE. Der Befehl sieht etwa wie folgt aus:

```
db2 uncatalog ldap node <ldap_knotenname>
```

#### Zugehörige Tasks:

- „Registrierung von DB2-Servern nach der Installation“ auf Seite 347
- „Herstellen einer ATTACH-Verbindung zu einem fernen Server in der LDAP-Umgebung“ auf Seite 352

#### Zugehörige Referenzen:

- „CATALOG LDAP NODE Command“ in *Command Reference*
- „UNCATALOG LDAP NODE Command“ in *Command Reference*

---

## Zurücknehmen der Registrierung des DB2-Servers

### Vorgehensweise:

Beim Zurücknehmen der Registrierung eines Exemplars aus LDAP werden auch alle Knoten- oder Aliasnamenobjekte sowie Datenbankobjekte entfernt, die auf das Exemplar verweisen.

Zum Zurücknehmen der Registrierung des DB2-Servers auf einer lokalen oder fernen Maschine muss der LDAP-Knotenname für den Server angegeben werden:

```
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.

### Zugehörige Tasks:

- „Registrierung von DB2-Servern nach der Installation“ auf Seite 347

### Zugehörige Referenzen:

- „DEREGISTER Command“ in *Command Reference*

---

## Registrierung von Datenbanken im LDAP-Verzeichnis

### Vorgehensweise:

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 Clientmaschine 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"
```

### Zugehörige Tasks:

- „Registrierung von DB2-Servern nach der Installation“ auf Seite 347
- „Zurücknehmen der Registrierung der Datenbank aus dem LDAP-Verzeichnis“ auf Seite 353

### Zugehörige Referenzen:

- „CATALOG LDAP DATABASE Command“ in *Command Reference*

---

## Herstellen einer ATTACH-Verbindung zu einem fernen Server in der LDAP-Umgebung

### Vorgehensweise:

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 Clientanwendung zum ersten Mal eine Verbindung zu einem Knoten oder einer Datenbank herstellt, durchsucht der Datenbankmanager 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 der Eintrag im LDAP-Verzeichnis gefunden wird, werden die Datenbankinformationen ebenfalls abgerufen. Mit Hilfe dieser Informationen katalogisiert der Datenbankmanager automatisch einen Datenbankeintrag und einen Knoteneintrag auf der lokalen Maschine. Wenn die Clientanwendung 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.

Es gibt einen Cachingmechanismus, der dafür sorgt, dass der Client den LDAP-Server nur einmal durchsucht. Sobald die Informationen abgerufen sind, werden sie auf der lokalen Maschine nach Maßgabe der Werte des Konfigurationsparameters *dir\_cache* des Datenbankmanagers und der Registriervariablen DB2LDAPCACHE gespeichert bzw. im Cache abgelegt.

- Sind DB2LDAPCACHE=NO und *dir\_cache*=NO definiert, werden die Informationen immer aus LDAP gelesen.
- Sind DB2LDAPCACHE=NO und *dir\_cache*=YES definiert, werden die Informationen einmal aus LDAP gelesen und dann in den DB2-Cache eingefügt.
- Ist DB2LDAPCACHE=YES oder diese Registriervariable überhaupt nicht definiert, werden die Informationen einmal aus dem LDAP-Server gelesen und in den Cache für lokale Datenbank-, Knoten- und DCS-Verzeichnisse gestellt.

**Anmerkung:** Das Caching von LDAP-Informationen kann für CLI-Variablen oder DB2-Variablen der Profilregistrierdatenbank auf Benutzerebene nicht ausgeführt werden.

### Zugehörige Konzepte:

- „Registrierdatenbank- und Umgebungsvariablen von DB2“ in *Systemverwaltung: Optimierung*

### Zugehörige Tasks:

- „Registrierung von DB2-Servern nach der Installation“ auf Seite 347
- „Aktualisieren der Protokollinformationen für den DB2-Server“ auf Seite 349
- „Katalogisieren eines Knoten-Aliasnamens für ATTACH“ auf Seite 350
- „Registrierung von Datenbanken im LDAP-Verzeichnis“ auf Seite 351



#### Zugehörige Referenzen:

- „ATTACH Command“ in *Command Reference*

---

## Zurücknehmen der Registrierung der Datenbank aus dem LDAP-Verzeichnis

#### Vorgehensweise:

Die Registrierung der Datenbank in LDAP wird automatisch in den folgenden Fällen zurückgenommen:

- Die Datenbank wird gelöscht.
- Die Registrierung des Eigenerexemplars in LDAP wird zurückgenommen.

Die Registrierung der Datenbank in LDAP kann mit dem folgenden Befehl manuell zurückgenommen werden:

```
db2 uncatalog ldap database <dbname>
```

#### Zugehörige Tasks:

- „Registrierung von Datenbanken im LDAP-Verzeichnis“ auf Seite 351

#### Zugehörige Referenzen:

- „UNCATALOG LDAP DATABASE Command“ in *Command Reference*

---

## Aktualisieren der LDAP-Einträge in lokalen Datenbank- und Knotenverzeichnissen

#### Vorgehensweise:

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.

Es gibt einen Cachingmechanismus, der dafür sorgt, dass der Client den LDAP-Server nur einmal durchsucht. Sobald die Informationen abgerufen sind, werden sie auf der lokalen Maschine nach Maßgabe der Werte des Konfigurationsparameters *dir\_cache* des Datenbankmanagers und der Registriervariablen DB2LDAPCACHE gespeichert bzw. im Cache abgelegt.

- Sind DB2LDAPCACHE=NO und *dir\_cache*=NO definiert, werden die Informationen immer aus LDAP gelesen.
- Sind DB2LDAPCACHE=NO und *dir\_cache*=YES definiert, werden die Informationen einmal aus LDAP gelesen und dann in den DB2-Cache eingefügt.
- Ist DB2LDAPCACHE=YES oder diese Registriervariable überhaupt nicht definiert, werden die Informationen einmal aus dem LDAP-Server gelesen und in den Cache für lokale Datenbank-, Knoten- und DCS-Verzeichnisse gestellt.

**Anmerkung:** Das Caching von LDAP-Informationen kann für CLI-Variablen oder DB2-Variablen der Profilregistrieredatenbank auf Benutzerebene nicht ausgeführt werden.

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 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 Clientmaschinen aufzurufen.
- Definieren Sie DB2LDAPCACHE=„NO“, um das Caching von LDAP-Informationen in der Datenbank, dem Knoten und den DCS-Verzeichnissen zu verhindern.

#### Zugehörige Konzepte:

- „Registrierdatenbank- und Umgebungsvariablen von DB2“ in *Systemverwaltung: Optimierung*

#### Zugehörige Referenzen:

- „dir\_cache - Verzeichniscacheunterstützung“ in *Systemverwaltung: Optimierung*
- „REFRESH LDAP Command“ in *Command Reference*

---

## Durchsuchen der LDAP-Verzeichnispartitionen oder -Domänen

### Vorgehensweise:

DB2 UDB durchsucht die aktuelle LDAP-Verzeichnispartition oder die aktuelle 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 UDB-Profilregistriervariable DB2LDAP\_SEARCH\_SCOPE 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 Verzeichnispartition 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 Clientmaschinen 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 UDB-Profilregistriervariablen DB2LDAP\_KEEP\_CONNECTION und DB2LDAP\_SEARCH\_SCOPE sind die einzigen Registriervariablen, die das Setzen einer Variablen auf globaler Ebene unter LDAP unterstützen.

**Zugehörige Konzepte:**

- „Registrierdatenbank- und Umgebungsvariablen von DB2“ in *Systemverwaltung: Optimierung*

**Zugehörige Tasks:**

- „Deklarieren von Registrier- und Umgebungsvariablen“ auf Seite 33

---

## Registrieren von Hostdatenbanken in LDAP

**Vorgehensweise:**

Beim Registrieren von Hostdatenbanken unter LDAP gibt es zwei Konfigurationsmöglichkeiten:

- Direkte Verbindung zu den Hostdatenbanken
- Verbindung zur Hostdatenbank über ein Gateway

Im ersten Fall registriert der Benutzer den Host-Server unter LDAP. Anschließend katalogisiert er die Hostdatenbank unter LDAP, wobei er den Knotennamen des Host-Servers angibt. Im zweiten Fall registriert der Benutzer den Gateway-Server unter LDAP. Anschließend katalogisiert er die Hostdatenbank unter LDAP, wobei er den Knotennamen des Gateway-Servers angibt.

Wenn die LDAP-Unterstützung auf dem DB2® Connect-Gateway zur Verfügung steht und die Datenbank im Gateway-Datenbankverzeichnis nicht gefunden werden kann, sucht DB2 UDB über LDAP und versucht, die gefundenen Informationen zu speichern.

Das folgende ein Beispiel veranschaulicht beide Fälle: Nehmen Sie an, Sie verfügen über die Hostdatenbank 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 ein Gateway mit dem Namen „goto@niagara“.

Sie müssen die folgenden Arbeitsschritte ausführen:

1. Registrieren Sie den Hostdatenbankserver 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 Hostdatenbankserver handelt.

```
db2 register ldap as nfappn appn network CAIBMOML partnerlu NFLU
mode IBMRDB remote mvssys instance msvinst nodetype dcs
```

2. Registrieren Sie den Hostdatenbankserver unter LDAP für die TCP/IP-Konnektivität. Der TCP/IP-Hostname des Servers ist „myhost“ und die Portnummer

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 Hostdatenbankserver 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-Hostname für den Gateway-Server ist „niagara“ und die Portnummer ist „50000“.

```
db2 register ldap as whasf tcpip hostname niagara svcname 50000
remote niagara instance goto nodetype server
```

4. Katalogisieren Sie die Hostdatenbank unter LDAP und verwenden Sie hierbei die TCP/IP-Konnektivität. Die Hostdatenbank 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 Hostdatenbank in LDAP unter Verwendung der 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 das Gateway unter TCP/IP hergestellt. Abhängig davon, ob Sie „nftcpip“ oder „nfappn“ verwenden, stellt das Gateway dann unter TCP/IP oder APPN eine Verbindung zum gewünschten Host her.

Sie können Angaben zur Hostdatenbank 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 Hostdatenbankserver in LDAP. Dazu müssen Sie den Namen des fernen Computers, den Exemplarnamen und den Knotentyp für den Hostdatenbankserver im Befehl REGISTER angeben, wobei Sie die Klausel REMOTE, INSTANCE und NODETYPE verwenden. Die Klausel REMOTE kann entweder auf den Hostnamen oder den LU-Namen der Hostservermaschine 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 NODETYPE muss auf „DCS“ gesetzt werden, um anzugeben, dass es sich bei der Einheit um einen Hostdatenbankserver handelt.
2. Registrieren Sie die Hostdatenbank in LDAP mit dem Befehl CATALOG LDAP DATABASE. Weitere DRDA-Parameter können mit der Klausel PARMS angegeben werden. Der Datenbankauthentifizierungstyp sollte auf „DCS“ gesetzt sein.

#### Zugehörige Referenzen:

- „REGISTER Command“ in *Command Reference*
- „CATALOG LDAP DATABASE Command“ in *Command Reference*

---

## Definieren von DB2-Registriervariablen auf Benutzerebene in der LDAP-Umgebung

### Vorgehensweise:

In der LDAP-Umgebung können die DB2-Variablen der Profilregistrierdatenbank auf Benutzerebene definiert werden. Dadurch können Benutzer ihre eigene DB2-Umgebung anpassen. Um die DB2-Variablen der Profilregistrierdatenbank auf Benutzerebene zu definieren, verwenden Sie die Option `-ul`:

```
db2set -ul <variable>=<wert>
```

**Anmerkung:** Diese Funktion wird unter AIX und Solaris Operating Environment nicht unterstützt.

DB2 verfügt über ein Caching-Verfahren. Die DB2-Variablen der Profilregistrierdatenbank auf Benutzerebene werden auf der lokalen Maschine zwischengespeichert. Wenn der Parameter `-ul` angegeben wird, liest DB2 die DB2-Registriervariablen immer aus dem Cache. Der Cache wird in folgenden Fällen aktualisiert:

- Sie aktualisieren eine DB2-Registriervariable auf Benutzerebene oder setzen sie zurück.
- Der Befehl zum Aktualisieren der LDAP-Profilvariablen auf Benutzerebene lautet wie folgt:

```
db2set -ur
```

### Zugehörige Tasks:

- „Deklariere von Registrier- und Umgebungsvariablen“ auf Seite 33

### Zugehörige Referenzen:

- „db2set - DB2 Profile Registry Command“ in *Command Reference*

---

## Aktivieren der LDAP-Unterstützung nach Abschluss der Installation

### Vorgehensweise:

Gehen Sie auf allen Maschinen wie folgt vor, 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, Windows NT und UNIX-Plattformen müssen Sie LDAP explizit aktivieren. Setzen Sie dazu die Registriervariable `DB2_ENABLE_LDAP` mit dem Befehl `db2set` auf den Wert „YES“.

- (Nur UNIX-Plattformen) Deklarieren Sie mit dem folgenden Befehl den TCP/IP-Hostnamen des LDAP-Servers und (optional) die Portnummer:

```
db2set DB2LDAPHOST=<basistdomänenname>[:portnummer]
```

Dabei steht `basistdomänenname` für den TCP/IP-Hostnamen des LDAP-Servers und `[:portnummer]` für die zugehörige Portnummer. Wird keine Portnummer angegeben, verwendet DB2 den LDAP-Standardport (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>
```

Dabei 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. Zum 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. Zum Beispiel:

```
db2 catalog ldap database <dbname> as <db_aliasname>
```

- 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.

#### **Zugehörige Konzepte:**

- „Registrierdatenbank- und Umgebungsvariablen von DB2“ in *Systemverwaltung: Optimierung*

#### **Zugehörige Tasks:**

- „Inaktivieren der LDAP-Unterstützung“ auf Seite 358

#### **Zugehörige Referenzen:**

- „REGISTER Command“ in *Command Reference*
- „db2set - DB2 Profile Registry Command“ in *Command Reference*
- „CATALOG LDAP DATABASE Command“ in *Command Reference*

---

## **Inaktivieren der LDAP-Unterstützung**

#### **Vorgehensweise:**

Verwenden Sie 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“.

#### **Zugehörige Tasks:**

- „Deklarieren von Registrier- und Umgebungsvariablen“ auf Seite 33
- „Aktivieren der LDAP-Unterstützung nach Abschluss der Installation“ auf Seite 357

#### **Zugehörige Referenzen:**

- „DEREGISTER Command“ in *Command Reference*

---

## LDAP-Unterstützung und DB2 Connect

Wenn die LDAP-Unterstützung auf dem DB2<sup>®</sup> Connect-Gateway zur Verfügung steht und die Datenbank im Gateway-Datenbankverzeichnis nicht gefunden werden kann, sucht DB2 über LDAP und versucht, die gefundenen Informationen zu behalten.

### Zugehörige Konzepte:

- „Einführung in Lightweight Directory Access Protocol (LDAP)“ auf Seite 341
- „Sicherheitsaspekte in einer LDAP-Umgebung“ auf Seite 359

---

## Sicherheitsaspekte in einer LDAP-Umgebung

Vor dem Zugriff auf Informationen im LDAP-Verzeichnis wird eine Anwendung oder ein Benutzer vom LDAP-Server authentifiziert. Die Authentifizierung wird als *Binden* an den LDAP-Server bezeichnet.

Für die im LDAP-Verzeichnis gespeicherten Informationen muss eine Zugriffssteuerung angewendet werden, um zu verhindern, dass anonyme Benutzer Informationen löschen, ändern oder hinzufügen.

Die Zugriffssteuerung wird standardmäßig übernommen und kann auf Behälterebene angewendet werden. Wenn ein neues Objekt erstellt wird, übernimmt es dasselbe Sicherheitsattribut wie das übergeordnete Objekt. Ein für den LDAP-Server verfügbares Verwaltungstool 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 Eigner oder Ersteller des Objekts haben 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<sup>®</sup> durchgeführt. Die LDAP-Berechtigungsprüfung ist unabhängig von der DB2-Berechtigung. Ein Konto oder eine Berechtigung-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 dabei die Standardidentitätsnachweise, denen möglicherweise nicht ausreichende 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.

### Zugehörige Konzepte:

- „Sicherheitsaspekte für Active Directory“ auf Seite 360

---

## Sicherheitsaspekte für Active Directory

Die DB2<sup>®</sup>-Datenbank- und -Knotenobjekte werden unter dem Computerobjekt der Maschine erstellt, auf der der DB2-Server im Active Directory installiert ist. Zum Registrieren eines Datenbankservers oder Katalogisieren im Active Directory müssen Sie über ausreichende Zugriffsberechtigungen zum Erstellen 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 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 das Snap-in **Active Directory-Benutzer und -Computer** der Microsoft Management Console (MMC) wie folgt:

1. Starten Sie das Verwaltungstool **Active Directory-Benutzer und -Computer**. (Start—> Programme—> Verwaltung—> Active Directory-Benutzer und -Computer)
2. Wählen Sie unter **Ansicht** die Option **Erweiterte Funktionen** aus.
3. Wählen Sie den Container **Computers** aus.
4. Klicken Sie mit der rechten Maustaste auf das Computerobjekt, das die Servermaschine darstellt, auf der DB2 installiert ist, und wählen Sie **Eigenschaften** aus.
5. Wählen Sie die Registerkarte **Sicherheitseinstellungen** aus und fügen Sie für die angegebene Gruppe bzw. den angegebenen Benutzer die erforderliche Zugriffsberechtigungen hinzu.

Die DB2-Registriervariablen und CLI-Einstellungen auf Benutzerebene werden im Objekt für DB2-Eigenschaften unter dem Benutzerobjekt verwaltet. Um die DB2-Registriervariablen 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-Registriervariablen oder CLI-Einstellungen auf Benutzerebene zu erteilen, verwenden Sie das Snap-in **Active Directory-Benutzer und -Computer** der Microsoft Management Console (MMC) wie folgt:

1. Starten Sie das Verwaltungstool **Active Directory-Benutzer und -Computer**. (Start—> Programme—> Verwaltung—> Active Directory-Benutzer und -Computer)
2. Wählen Sie das Benutzerobjekt im Container **Users** aus.
3. Klicken Sie mit der rechten Maustaste auf das Benutzerobjekt und wählen Sie **Eigenschaften** aus.
4. Wählen Sie die Registerkarte **Sicherheitseinstellungen** aus.
5. Fügen Sie den Benutzername mit Hilfe des Knopfes **Hinzufügen** der Liste hinzu.
6. Erteilen Sie die Zugriffsberechtigungen „Schreiben“ und „Alle untergeordneten Objekte erstellen“.



7. In den erweiterten Einstellungen (Knopf **Erweitert**) wählen Sie im Feld **Übernehmen für** den Eintrag „Dieses und alle untergeordneten Objekte“ aus.
8. Wählen Sie das Markierungsfeld „Vererbte übergeordnete Berechtigungen übernehmen“ aus.

**Zugehörige Konzepte:**

- „Sicherheitsaspekte in einer LDAP-Umgebung“ auf Seite 359

---

## Erweitern des LDAP-Verzeichnisschemas mit DB2-Objektklassen und -Attributen

Das LDAP-Verzeichnisschema definiert Objektklassen und Attribute für die in den LDAP-Verzeichniseinträgen gespeicherten Informationen. Eine Objektklasse besteht aus einer Reihe von verbindlichen und optionalen Attributen. Jedem Eintrag im LDAP-Verzeichnis ist eine Objektklasse zugeordnet.

Bevor DB2<sup>®</sup> die Informationen in LDAP speichern kann, muss das Verzeichnisschema für den LDAP-Server die von DB2 verwendeten Objektklassen und Attribute aufnehmen. Das Hinzufügen neuer Objektklassen und Attribute zum Basisschema wird als Erweitern des Verzeichnisschemas bezeichnet.

**Anmerkung:** Wenn Sie IBM<sup>®</sup> SecureWay<sup>®</sup> LDAP Directory Version 3.1 verwenden, sind alle für DB2 UDB Version 8.1 und früheren Versionen erforderlichen Objektklassen und Attribute im Basisschema enthalten. In diesem Fall müssen Sie das Basisschema nicht um DB2-Objektklassen und -Attribute erweitern. Zwei neue Attribute für DB2 UDB Version 8.2 sind jedoch im Basisschema nicht enthalten. In diesem Fall müssen Sie das Basisschema um die beiden neuen DB2 UDB-Attribute erweitern.

**Zugehörige Konzepte:**

- „Erweitern des Verzeichnisschemas für IBM SecureWay Directory Server“ auf Seite 365

**Zugehörige Tasks:**

- „Erweitern des Verzeichnisschemas für Active Directory“ auf Seite 361

---

## Erweitern des Verzeichnisschemas für Active Directory

**Vorgehensweise:**

Bevor DB2 Universal Database<sup>™</sup> (DB2 UDB) Informationen im Active Directory speichern kann, muss das Verzeichnisschema so erweitert werden, dass es die neuen DB2 UDB-Objektklassen und -Attribute umfasst. Das Hinzufügen neuer Objektklassen und Attribute zum Verzeichnisschema wird als *Schemaerweiterung* bezeichnet.

Vor der Erstinstallation von DB2 UDB auf den Maschinen einer Windows-Domäne müssen Sie das Programm **db2schex** für die DB2 UDB-Schemainstallation ausführen, um das Schema für Active Directory zu erweitern.

Das Programm **db2schex** befindet sich auf der Produkt-CD-ROM. Das Programm ist auf der CD-ROM im Verzeichnis db2 im Unterverzeichnis windows und utilities gespeichert. Zum Beispiel:

```
x:\db2\windows\utilities\
```

Dabei steht x: für das CD-ROM-Laufwerk.

Der Befehl ist wie folgt anzugeben:

```
db2schex
```

Diesem Befehl sind folgende wahlfreie Klauseln zugeordnet:

- -b BenutzerDN  
Zur Angabe des registrierten Benutzernamens
- -w Kennwort  
Zur Angabe des BIND-Kennwortes
- -u  
Zum Deinstallieren des Schemas
- -k  
Zum zwingenden Fortsetzen der Deinstallation und Ignorieren von Fehlermeldungen

#### Anmerkungen:

1. Wurden kein registrierter Benutzername (BenutzerDN) und kein Kennwort angegeben, führt **db2schex** die Bindeoperation mit dem momentan angemeldeten Benutzer durch.
2. Die Klausel BenutzerDN 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.

Zur Erweiterung des Verzeichnisschemas müssen Sie den Befehl **db2schex.exe** ausführen, der im Produkt DB2 UDB Version 8.2 enthalten ist.

Wenn Sie den Befehl **db2schex.exe** aus der vorigen Version des DB2 UDB-Produkts für Windows ausgeführt haben und den gleichen Befehl aus DB2 UDB Version 8.2 noch einmal ausführen, werden die beiden folgenden optionalen Attribute der ibm-db2Database-Klasse hinzugefügt:

```
ibm-db2AltGwPtr  
ibm-db2NodePtr
```

Wenn Sie den Befehl **db2schex.exe** aus der vorigen Version des DB2 UDB-Produkts für Windows nicht ausgeführt haben und diesen Befehl aus DB2 Version 8.2 ausführen, werden alle Klassen und Attribute für die LDAP-Unterstützung von DB2 UDB hinzugefügt.

Beispiele:

- Geben Sie Folgendes ein, um das DB2 UDB-Schema zu installieren:  

```
db2schex
```
- Geben Sie Folgendes ein, um das DB2 UDB-Schema zu installieren und einen registrierten Bindnamen (bindDN) und ein Kennwort zu definieren:  

```
db2schex -b "cn=A Name,dc=toronto1,dc=ibm,dc=com"  
-w kennwort
```

Alternativ können Sie auch Folgendes eingeben:

```
db2schex -b Administrator -w kennwort
```

- Geben Sie Folgendes ein, um das DB2 UDB-Schema zu deinstallieren:  

```
db2schex -u
```
- Geben Sie Folgendes ein, um das DB2 UDB-Schema zu deinstallieren und Fehlermeldungen zu ignorieren:  

```
db2schex -u -k
```

Das DB2 UDB-Schemainstallationsprogramm für Active Directory führt die folgenden Tasks aus:

**Anmerkungen:**

1. Feststellen des Servers, der als Schema-Master fungiert
2. Binden an den Domänencontroller, der als Schema-Master fungiert
3. Sicherstellen ausreichender Benutzerberechtigungen zum Hinzufügen von Klassen und Attributen zum Schema
4. Sicherstellen des Schreibzugriffs auf den Schema-Master (dazu wird die Sicherheitsperre in der Registrierdatenbank entfernt)
5. Erstellen aller neuen Attribute
6. Erstellen aller neuen Objektklassen
7. Feststellen von Fehlern und ggf. Rückgängigmachen am Schema vorgenommener Änderungen

**Zugehörige Konzepte:**

- „Erweitern des LDAP-Verzeichnisschemas mit DB2-Objektklassen und -Attributen“ auf Seite 361

---

## DB2-Objekte im Active Directory

DB2 erstellt Objekte im 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-Servermaschine, die nicht zur Windows NT-Domäne gehört, werden die DB2-Datenbank- und -Knotenobjekte im Container „System“ erstellt.
2. Die DB2-Registriervariablen und CLI-Einstellungen auf Benutzerebene werden in den DB2-Eigenschaftsobjekten unter dem Benutzerobjekt gespeichert. Diese Objekte enthalten benutzerspezifische Informationen.

**Zugehörige Referenzen:**

- „Von DB2 verwendete LDAP-Objektklassen und Attribute“ auf Seite 369

---

## Netscape-Unterstützung und Attributdefinitionen für das LDAP-Verzeichnis

Die unterstützte Stand für Netscape LDAP Server ist Version 4.12 oder eine spätere Version.

In Netscape LDAP Server Version 4.12 (oder später) ermöglicht der Netscape Directory Server Anwendungen, das Schema durch Hinzufügen von Attribut-

und Objektklassendefinitionen in den beiden Dateien `slapd.user_oc.conf` und `slapd.user_at.conf` zu erweitern. Diese beiden Dateien befinden sich im folgenden Verzeichnis:

```
<Netscape_installationspfad>\slapd-<maschinenname>\config
```

**Anmerkung:** Wenn Sie iPlan Directory Server 5.0 einsetzen, müssen Sie die Dokumentation zu diesem Produkt konsultieren, um detaillierte Anweisungen zur Erweiterung des Schemas zu erhalten.

Folgende DB2-Attribute müssen der Datei `slapd.user_at.conf` hinzugefügt werden:

```
#####
#
# IBM DB2 Universal Database
# Attributdefinitionen
#
# bin -> binär
# ces -> Zeichenfolge in exakter Groß-/Kleinschreibung
# cis -> Zeichenfolge ohne Unterscheidung der Groß-/Kleinschreibung
# dn -> registrierter Name (Distinguished Name)
#
#####

attribute binProperty                1.3.18.0.2.4.305    bin
attribute binPropertyType            1.3.18.0.2.4.306    cis
attribute cesProperty                1.3.18.0.2.4.307    ces
attribute cesPropertyType            1.3.18.0.2.4.308    cis
attribute cisProperty                1.3.18.0.2.4.309    cis
attribute cisPropertyType            1.3.18.0.2.4.310    cis
attribute propertyType                1.3.18.0.2.4.320    cis
attribute systemName                 1.3.18.0.2.4.329    cis
attribute db2nodeName                1.3.18.0.2.4.419    cis
attribute db2nodeAlias                1.3.18.0.2.4.420    cis
attribute db2instanceName            1.3.18.0.2.4.428    cis
attribute db2Type                     1.3.18.0.2.4.418    cis
attribute db2databaseName            1.3.18.0.2.4.421    cis
attribute db2databaseAlias           1.3.18.0.2.4.422    cis
attribute db2nodePtr                 1.3.18.0.2.4.423    dn
attribute db2gwPtr                   1.3.18.0.2.4.424    dn
attribute db2additionalParameters    1.3.18.0.2.4.426    cis
attribute db2ARLibrary                1.3.18.0.2.4.427    cis
attribute db2authenticationLocation  1.3.18.0.2.4.425    cis
attribute db2databaseRelease         1.3.18.0.2.4.429    cis
attribute DCEPrincipalName           1.3.18.0.2.4.443    cis
```

Folgende DB2-Objektklassen müssen der Datei `slapd.user_oc.conf` hinzugefügt werden:

```
#####
#
# IBM DB2 Universal Database
# Objektklassendefinitionen
#
#####

objectclass eProperty
    oid 1.3.18.0.2.6.90
    requires
        objectClass
    allows
        cn,
        propertyType,
        binProperty,
        binPropertyType,
```

```

        cesProperty,
        cesPropertyType,
        cisProperty,
        cisPropertyType

objectclass eApplicationSystem
    oid 1.3.18.0.2.6.84
    requires
        objectClass,
        systemName

objectclass DB2Node
    oid 1.3.18.0.2.6.116
    requires
        objectClass,
        db2nodeName
    allows
        db2nodeAlias,
        host,
        db2instanceName,
        db2Type,
        description,
        protocolInformation

objectclass DB2Database
    oid 1.3.18.0.2.6.117
    requires
        objectClass,
        db2databaseName,
        db2nodePtr
    allows
        db2databaseAlias,
        description,
        db2gwPtr,
        db2additionalParameters,
        db2authenticationLocation,
        DCEPrincipalName,
        db2databaseRelease,
        db2ARLibrary

```

Nach dem Hinzufügen der DB2-Schemadefinition muss der Directory Server erneut gestartet werden, um alle Änderungen in Kraft zu setzen.

#### Zugehörige Referenzen:

- „Von DB2 verwendete LDAP-Objektklassen und Attribute“ auf Seite 369

---

## Erweitern des Verzeichnisschemas für IBM SecureWay Directory Server

Wenn Sie IBM® SecureWay® LDAP Directory Server Version 3.1 oder eine spätere Version verwenden, sind alle für DB2® Universal Database (DB2 UDB) vor Version 8.2 erforderlichen Objektklassen und Attribute im Basisschema enthalten. Führen Sie den folgenden Befehl aus, um das Basisschema um die neuen, in Version 8.2 eingeführten DB2 UDB-Attributen zu erweitern:

```
ldapmodify -c -h <maschinename>:389 -D <dn> -w <kennwort> -f altgwnode.ldif
```

Die Datei `altgwnode.ldif` besitzt folgenden Inhalt:

```

dn: cn=schema
changeType: modify
add: attributeTypes (
1.3.18.0.2.4.3092
NAME 'db2altgwpPtr'
DESC 'DN pointer to DB2 alternate gateway (node) object'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
-

```

```

add: ibmattributeTypes
ibmattributeTypes: (
1.3.18.0.2.4.3092
DBNAME ('db2altgwpPtr' 'db2altgwpPtr')
ACCESS-CLASS NORMAL
LENGTH 1000)

```

```

dn: cn=schema
changeType: modify
add: attributeTypes (
1.3.18.0.2.4.3093
NAME 'db2altnodePtr'
DESC 'DN pointer to DB2 alternate node object'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
-

```

```

add: ibmattributeTypes
ibmattributeTypes: (
1.3.18.0.2.4.3093
DBNAME ('db2altnodePtr' 'db2altnodePtr')
ACCESS-CLASS NORMAL
LENGTH 1000)

```

```

dn: cn=schema
changeType: modify
replace: objectClasses (
1.3.18.0.2.6.117
NAME 'DB2Database'
DESC 'DB2 database'
SUP ctmSetting
MUST ( db2databaseName $ db2nodePtr )
MAY ( db2additionalParameters $ db2altgwpPtr $ db2altnodePtr $ db2authenticationLocation $ db2databaseAlias $ db2databaseRelease $ db2gwpPtr $ DCEPrincipalName ) )

```

Die Dateien `altgwnode.ldif` und `altgwnode.readme` stehen unter folgender URL-Adresse zur Verfügung: `ftp://ftp.software.ibm.com/ps/products/db2/tools/ldap`

Nach dem Hinzufügen der DB2-Schemadefinition muss der Directory Server erneut gestartet werden, um alle Änderungen in Kraft zu setzen.

**Zugehörige Konzepte:**

- „Erweitern des LDAP-Verzeichnisschemas mit DB2-Objektklassen und -Attributen“ auf Seite 361
- „Erweitern des Verzeichnisschemas für Sun One Directory Server“ auf Seite 367

**Zugehörige Tasks:**

- „Erweitern des Verzeichnisschemas für Active Directory“ auf Seite 361

---

## Erweitern des Verzeichnisschemas für Sun One Directory Server

Sun One Directory Server wird auch als Netscape- oder iPlanet-Verzeichnisserver bezeichnet.

Zur funktionsfähigen Einrichtung von Sun One Directory Server in Ihrer Umgebung fügen Sie die Datei `60ibmdb2.ldif` dem folgenden Verzeichnis hinzu:

Unter Windows<sup>®</sup>, wenn iPlanet im Verzeichnis `C:\iPlanet\Servers` installiert ist, fügen Sie die oben genannte Datei dem Verzeichnis `.\slapd-  
<maschinenname>\config\schema` hinzu.

Unter UNIX<sup>®</sup>, wenn iPlanet im Verzeichnis `/usr/iplanet/servers` installiert ist, fügen Sie die oben genannte Datei dem Verzeichnis `./slapd-  
<maschinenname>/config/schema` hinzu.

Die Datei besitzt folgenden Inhalt:

```
#####
# IBMRegtm; DB2&regtm; Universal Database
#####
dn: cn=schema
#####
# Attribute Definitions (Before V8.2)
#####
attributeTypes: ( 1.3.18.0.2.4.305 NAME 'binProperty'
attributeTypes: ( 1.3.18.0.2.4.306 NAME 'binPropertyType'
attributeTypes: ( 1.3.18.0.2.4.307 NAME 'cesProperty'
attributeTypes: ( 1.3.18.0.2.4.308 NAME 'cesPropertyType'
attributeTypes: ( 1.3.18.0.2.4.309 NAME 'cisProperty'
attributeTypes: ( 1.3.18.0.2.4.310 NAME 'cisPropertyType'
attributeTypes: ( 1.3.18.0.2.4.320 NAME 'propertyType'
attributeTypes: ( 1.3.18.0.2.4.329 NAME 'systemName'
attributeTypes: ( 1.3.18.0.2.4.419 NAME 'db2nodeAlias'
attributeTypes: ( 1.3.18.0.2.4.420 NAME 'db2nodeAlias'
attributeTypes: ( 1.3.18.0.2.4.428 NAME 'db2instanceName'
attributeTypes: ( 1.3.18.0.2.4.418 NAME 'db2Type'
attributeTypes: ( 1.3.18.0.2.4.421 NAME 'db2databaseName'
attributeTypes: ( 1.3.18.0.2.4.422 NAME 'db2databaseAlias'
attributeTypes: ( 1.3.18.0.2.4.425 NAME 'db2additionalParameters'
attributeTypes: ( 1.3.18.0.2.4.427 NAME 'db2ARLibrary'
attributeTypes: ( 1.3.18.0.2.4.425 NAME 'db2authenticationLocation'
attributeTypes: ( 1.3.18.0.2.4.429 NAME 'db2databaseRelease'
attributeTypes: ( 1.3.18.0.2.4.443 NAME 'DCEPPrincipalName'
attributeTypes: ( 1.3.18.0.2.4.423 NAME 'db2nodePtr'
attributeTypes: ( 1.3.18.0.2.4.424 NAME 'db2owPtr'
#####
# Attribute Definitions (V8.2)
#####
attributeTypes: ( 1.3.18.0.2.4.3092 NAME 'db2altgwPtr'
attributeTypes: ( 1.3.18.0.2.4.3093 NAME 'db2altnodePtr'
#####
# Object Class Definitions
#####
# DB2Database for V8.2 has the above two new optional attributes.
#####
objectClasses: ( 1.3.18.0.2.6.90 NAME 'eProperty' SUP top STRUCTURAL
MAY ( cn $ propertyType $ binPropertyType $ cesPropertyType $ cisPropertyType $ cisPropertyType $ cisPropertyType )
X-ORIGIN 'IBM DB2' )
objectClasses: ( 1.3.18.0.2.6.84 NAME 'eApplicationSystem' SUP top STRUCTURAL MUST systemName
X-ORIGIN 'IBM DB2' )
objectClasses: ( 1.3.18.0.2.6.116 NAME 'DR2Node' SUP top STRUCTURAL MUST db2nodeName
MAY ( db2instanceName $ db2nodeAlias $ db2Type $ description $ host $ protocolInformation )
X-ORIGIN 'IBM DB2' )
objectClasses: ( 1.3.18.0.2.6.117 NAME 'DR2Database' SUP top STRUCTURAL MUST (db2authenticationLocation $ db2nodePtr )
MAY ( db2additionalParameters $ db2altgwPtr $ db2altnodePtr $ db2authenticationLocation $ db2databaseName $ db2databaseRelease $ db2owPtr $ DCEPrincipalName $ description )
X-ORIGIN 'IBM DB2' )
```



Die Dateien 60ibmdb2.ldif und 60ibmdb2.readme stehen unter folgender URL-Adresse zur Verfügung: ftp://ftp.software.ibm.com/ps/products/db2/tools/ldap

Nach dem Hinzufügen der DB2-Schemadefinition muss der Directory Server erneut gestartet werden, um alle Änderungen in Kraft zu setzen.

**Zugehörige Konzepte:**

- „Erweitern des LDAP-Verzeichnisschemas mit DB2-Objektklassen und -Attributen“ auf Seite 361
- „Erweitern des Verzeichnisschemas für IBM SecureWay Directory Server“ auf Seite 365

**Zugehörige Tasks:**

- „Erweitern des Verzeichnisschemas für Active Directory“ auf Seite 361

---

## Von DB2 verwendete LDAP-Objektklassen und Attribute

In den folgenden Tabellen werden die von DB2 verwendeten Objektklassen beschrieben:

*Tabelle 23. 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.
Unterklasse von (SubClassOf)	top
Erforderliche(s) Attribut(e)	
Optionale(s) Attribut(e)	description
Typ	abstract
OID (Objekt-ID)	1.3.18.0.2.6.132
GUID (globale eindeutige Kennung)	b3afd63f-5c5b-11d3-b818-002035559151

*Tabelle 24. 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.
Unterklasse von (SubClassOf)	cimManagedElement
Erforderliche(s) Attribut(e)	
Optionale(s) Attribut(e)	settingID
Typ	abstract
OID (Objekt-ID)	1.3.18.0.2.6.131
GUID (globale eindeutige Kennung)	b3afd64d-5c5b-11d3-b818-002035559151

Table 25. 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.
Unterklasse von (SubClassOf)	cimSetting
Erforderliche(s) Attribut(e)	
Optionale(s) Attribut(e)	propertyType cisPropertyType cisProperty cesPropertyType cesProperty binPropertyType binProperty
Typ	structural
OID (Objekt-ID)	1.3.18.0.2.6.90
GUID (globale eindeutige Kennung)	b3afd69c-5c5b-11d3-b818-002035559151

Table 26. DB2Node

Objektklasse	DB2Node
Active Directory-LDAP-Anzeigename	ibm-db2Node
Allgemeiner Active Directory-Name (cn)	ibm-db2Node
Beschreibung	Stellt einen DB2-Server dar.
Unterklasse von (SubClassOf)	eSap / ServiceConnectionPoint
Erforderliche(s) Attribut(e)	db2nodeName
Optionale(s) Attribut(e)	db2nodeAlias db2instanceName db2Type host / dNSHostName (siehe Anm. 2) protocolInformation/ServiceBindingInformation
Typ	structural
OID (Objekt-ID)	1.3.18.0.2.6.116
GUID (globale eindeutige Kennung)	b3afd65a-5c5b-11d3-b818-002035559151

Tabelle 26. DB2Node (Forts.)

Objektklasse	DB2Node
Anmerkungen	<ol style="list-style-type: none"> <li>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.</li> <li>2. Das Attribut <i>host</i> wird in der IBM SecureWay-Umgebung verwendet. Das Attribut <i>dNSHostName</i> wird im Microsoft Active Directory verwendet.</li> <li>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.</li> </ol>

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-Datenbank-servers. Es besteht aus Token, die das unterstützte Netzwerkprotokoll beschreiben. Die Token werden jeweils 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-Hostname oder IP-Adresse
- Servicename (svcname) oder Portnummer (z. B. 50000)
- (Optional) 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“)
- (Optional) LAN-Adapteradresse
- (Optional) LU zur Kennwortänderung

**Anmerkung:** Wenn auf einem DB2 UDB für Windows NT-Client die APPN-Informationen nicht im lokalen SNA-Stack konfiguriert und die LAN-Adapteradresse und die optionale LU zur Kennwortänderung in LDAP lokalisiert wurden, versucht der DB2 UDB-Client, diese Informationen zum Konfigurieren des SNA-Stacks zu verwenden, wenn alle verfügbaren Angaben zum Konfigurieren des Stacks vorhanden sind.

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 27. DB2Database*

Objektklasse	DB2Database
Active Directory-LDAP-Anzeigename	ibm-db2Database
Allgemeiner Active Directory-Name (cn)	ibm-db2Database
Beschreibung	Stellt eine DB2-Datenbank dar.
Unterklasse von (SubClassOf)	top
Erforderliche(s) Attribut(e)	db2databaseName db2nodePtr
Optionale(s) Attribut(e)	db2databaseAlias db2additionalParameters db2ARLibrary db2authenticationLocation db2gwPtr db2databaseRelease DCEPrincipalName db2altgwPtr db2altnodePtr
Typ	structural
OID (Objekt-ID)	1.3.18.0.2.6.117
GUID (globale eindeutige Kennung)	b3afd659-5c5b-11d3-b818-002035559151

*Tabelle 28. 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 Hostdatenbankserver 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 29. 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
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
Anmerkungen	Zulässige Werte sind: CLIENT, SERVER, DCS, DCE, KERBEROS, SVRENCRYPT oder DCSENCRIPT

*Tabelle 30. 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 31. 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 32. 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

Table 32. db2databaseName (Forts.)

Attribut	db2databaseName
OID (Objekt-ID)	1.3.18.0.2.4.421
GUID (globale eindeutige Kennung)	b3afd319-5c5b-11d3-b818-002035559151

Table 33. 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

Table 34. 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

Table 35. 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

Table 36. db2nodePtr

Attribut	db2nodePtr
Active Directory-LDAP-Anzeigename	ibm-db2NodePtr
Allgemeiner Active Directory-Name (cn)	ibm-db2NodePtr

Table 36. db2nodePtr (Forts.)

Attribut	db2nodePtr
Beschreibung	Zeiger auf das Knotenobjekt (DB2Node) für den Datenbankserver, der Eigner der Datenbank ist
Syntax	Registrierter Name (DN)
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
Anmerkungen	Diese Abhängigkeit erlaubt es dem Client, die erforderlichen Informationen zum Übertragungsprotokoll abzurufen, um eine Verbindung zur Datenbank herzustellen.

Table 37. db2altnodePtr

Attribut	db2altnodePtr
Active Directory-LDAP-Anzeigename	ibm-db2AltNodePtr
Allgemeiner Active Directory-Name (cn)	ibm-db2AltNodePtr
Beschreibung	Zeiger auf das Knotenobjekt (DB2Node), das den alternativen Datenbankserver darstellt
Syntax	Registrierter Name (DN)
Maximale Länge	1000
Werte	Mit mehreren Werten
OID (Objekt-ID)	1.3.18.0.2.4.3093
GUID (globale eindeutige Kennung)	5694e266-2059-4e32-971e-0778909e0e72

Table 38. 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 (DN)
Maximale Länge	1000
Werte	Mit einem Wert
OID (Objekt-ID)	1.3.18.0.2.4.424
GUID (globale eindeutige Kennung)	b3afd31b-5c5b-11d3-b818-002035559151

Table 39. db2altgwPtr

Attribut	db2altgwPtr
Active Directory-LDAP-Anzeigename	ibm-db2AltGwPtr
Allgemeiner Active Directory-Name (cn)	ibm-db2AltGwPtr
Beschreibung	Zeiger auf das Knotenobjekt, das den alternativen Gateway-Server darstellt
Syntax	Registrierter Name (DN)

Tabelle 39. db2altgwPtr (Forts.)

Attribut	db2altgwPtr
Maximale Länge	1000
Werte	Mit mehreren Werten
OID (Objekt-ID)	1.3.18.0.2.4.3092
GUID (globale eindeutige Kennung)	70ab425d-65cc-4d7f-91d8-084888b3a6db

Tabelle 40. 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 41. db2Type

Attribut	db2Type
Active Directory-LDAP-Anzeigename	ibm-db2Type
Allgemeiner Active Directory-Name (cn)	ibm-db2Type
Beschreibung	Datenbankservertyp
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
Anmerkungen	Zulässige Datenbankservertypen: SERVER, MPP und DCS

Tabelle 42. DCEPrincipalName

Attribut	DCEPrincipalName
Active Directory-LDAP-Anzeigename	ibm-DCEPrincipalName
Allgemeiner Active Directory-Name (cn)	ibm-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 43. 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 44. 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 Merkmale 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
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

Table 45. cisProperty

Attribut	cisProperty
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 cisPropertyType 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

Table 46. *cisPropertyType*

Attribut	cisPropertyType
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 Merkmale aller Werte des Attributs cisProperty verwendet werden. Der Wert „INI File“ kann z. B. verwendet werden, um anzugeben, dass alle Werte des Attributs cisProperty INI-Dateien sind.
Syntax	Von Groß-/Kleinschreibung unabhängige Zeichenfolge
Maximale Länge	128
Werte	Mit mehreren Werten
OID (Objekt-ID)	1.3.18.0.2.4.310
GUID (globale eindeutige Kennung)	b3afd2e1-5c5b-11d3-b818-002035559151

Table 47. *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 48. *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 Merkmale 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

Table 48. binPropertyType (Forts.)

Attribut	binPropertyType
GUID (globale eindeutige Kennung)	b3afd2bb-5c5b-11d3-b818-002035559151

Table 49. PropertyType

Attribut	PropertyType
Active Directory-LDAP-Anzeigename	ibm-propertyType
Allgemeiner Active Directory-Name (cn)	ibm-propertyType
Beschreibung	Werte dieses Attributs beschreiben die semantischen Merkmale des eProperty-Objekts.
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

Table 50. 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

**Zugehörige Konzepte:**

- „Einführung in Lightweight Directory Access Protocol (LDAP)“ auf Seite 341



---

## Anhang D. Absetzen von Befehlen an mehrere Datenbankpartitionen

---

### Absetzen von Befehlen in einer partitionierten Datenbankumgebung

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<sup>®</sup> kann Ihre Anmeldeshell 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, das 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 Datenbankpartitionsserver oder auf allen ausgeführt wird. Wenn der Befehl auf einem Datenbankpartitionsserver 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**.

#### Zugehörige Konzepte:

- „Übersicht über die Befehle **rah** und **db2\_all**“ auf Seite 381
- „Angaben der Befehle **rah** und **db2\_all**“ auf Seite 383

#### Zugehörige Referenzen:

- „Beschreibungen der Befehle **rah** und **db2\_all**“ auf Seite 382

---

### Übersicht über die Befehle **rah** und **db2\_all**

Die Befehle können sequenziell auf einem Datenbankpartitionsserver nach dem anderen oder parallel ausgeführt werden. Wenn Sie die Befehle auf Plattformen unter UNIX<sup>®</sup> 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 in allen Datenbankpartitionen zu ändern, die in der Knotenkonfigurationsdatei angegeben sind. Da das Semikolon (;) in die doppelten Anführungszeichen mit eingeschlossen ist, wird die Anforderung parallel ausgeführt:

```
db2_all ";UPDATE DB CFG FOR sample USING LOGFILSIZ 100"
```

#### Zugehörige Konzepte:

- „Absetzen von Befehlen in einer partitionierten Datenbankumgebung“ auf Seite 381
- „Angaben der Befehle rah und db2\_all“ auf Seite 383

#### Zugehörige Referenzen:

- „Beschreibungen der Befehle rah und db2\_all“ auf Seite 382

---

## Beschreibungen der Befehle rah und db2\_all

Sie können die folgenden Befehle verwenden:

<b>Befehl</b>	<b>Beschreibung</b>
<b>rah</b>	Führt den Befehl auf allen Maschinen aus.
<b>db2_all</b>	Führt den Befehl auf allen Datenbankpartitionsservern aus, die Sie angeben.
<b>db2_kill</b>	Stoppt abrupt alle Prozesse, die auf mehreren Datenbankpartitionsservern ausgeführt werden, und bereinigt alle Ressourcen auf allen Datenbankpartitionsservern. Durch diesen Befehl werden Ihre Datenbanken inkonsistent. Führen Sie diesen Befehl <i>nicht</i> ohne entsprechende Anweisung durch den IBM Kundendienst aus.
<b>db2_call_stack</b>	<p>Auf Plattformen unter UNIX werden alle Prozesse, die auf allen Datenbankpartitionsservern ausgeführt werden, veranlasst, eine Ablaufverfolgungsliste für Aufrufe in das Systemprotokoll zu schreiben.</p> <p>Unter Windows NT werden alle Prozesse, die auf allen Datenbankpartitionsservern ausgeführt werden, veranlasst, eine Ablauf-</p>

verfolgungsliste für Aufrufe in die Datei *Pxxx.mnn* zu schreiben, wobei *Pxxx* die Prozess-ID und *mnn* 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.

#### Zugehörige Konzepte:

- „Übersicht über die Befehle rah und db2\_all“ auf Seite 381
- „Angabe der Befehle rah und db2\_all“ auf Seite 383
- „Paralleles Ausführen von Befehlen auf Plattformen unter UNIX“ auf Seite 384

---

## Angeben der Befehle rah und db2\_all

Sie können den Befehl wie folgt 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<sup>®</sup> wird der Befehl Ihrem Befehlsprotokoll so hinzugefügt, als hätten Sie ihn in die Eingabeaufforderung eingegeben.

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 Angabe der 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 Befehlsshell 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 Shellprozedur 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:

```
shellprozedur </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 ";diesen_befehl_ausfuehren" </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_diesen_befehl" &
```

**Zugehörige Konzepte:**

- „Paralleles Ausführen von Befehlen auf Plattformen unter UNIX“ auf Seite 384
- „Zusätzliche Informationen zum Befehl `rah` (nur Solaris und AIX)“ auf Seite 386

**Zugehörige Tasks:**

- „Einstellen des Standardumgebungsprofils für `rah` unter Windows NT“ auf Seite 393

**Zugehörige Referenzen:**

- „Beschreibungen der Befehle `rah` und `db2_all`“ auf Seite 382
- „Präfixsequenzen für den Befehl `rah`“ auf Seite 387
- „Steuern des Befehls `rah`“ auf Seite 391

---

## Paralleles Ausführen von Befehlen auf Plattformen unter UNIX

**Anmerkung:** Die Informationen dieses Abschnitts beziehen sich ausschließlich auf Plattformen, die auf UNIX<sup>®</sup> 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 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-Shellsyntax) 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, unter der der Befehl **rah** ausgeführt wird, und dass die ID, unter 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.

#### Zugehörige Konzepte:

- „Zusätzliche Informationen zum Befehl rah (nur Solaris und AIX)“ auf Seite 386

#### Zugehörige Tasks:

- „Überwachen von rah-Prozessen auf Plattformen unter UNIX“ auf Seite 385

#### Zugehörige Referenzen:

- „Präfixsequenzen für den Befehl rah“ auf Seite 387
- „Bestimmen von Fehlern mit rah auf Plattformen unter UNIX“ auf Seite 394

---

## Überwachen von rah-Prozessen auf Plattformen unter UNIX

### Vorgehensweise:

**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 (bzw. 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 Anmeldeshell 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 Anmeldeshell unter der ID, die die **rah**-Prozedur ausführt, keine Korn-Shell sein, wenn die Anmeldeshell unter 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.

#### Zugehörige Konzepte:

- „Paralleles Ausführen von Befehlen auf Plattformen unter UNIX“ auf Seite 384
- „Zusätzliche Informationen zum Befehl rah (nur Solaris und AIX)“ auf Seite 386

---

## Zusätzliche Informationen zum Befehl rah (nur 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 der Liste und sendet einen rekursiven Aufruf von sich selbst 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 kön-

nen. 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 für **rah**, da **rah** immer nur an bestimmte physische Knoten sendet.)

#### Zugehörige Konzepte:

- „Paralleles Ausführen von Befehlen auf Plattformen unter UNIX“ auf Seite 384

#### Zugehörige Tasks:

- „Überwachen von rah-Prozessen auf Plattformen unter UNIX“ auf Seite 385

---

## Präfixsequenzen für den Befehl rah

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) 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.  <b>Anmerkung:</b> Auf Plattformen unter UNIX wird durch die Angabe von & die Leistung beeinträchtigt, da mehr <b>rsh</b> -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.

	<b>Anmerkung:</b> Auf Plattformen unter UNIX wird durch die Angabe von & die Leistung beeinträchtigt, da mehr <b>rsh</b> -Befehle erforderlich werden.
;	Hat die gleiche Funktion wie   &. Dies ist eine kürzere alternative Form.
	<b>Anmerkung:</b> Auf Plattformen unter UNIX wird durch die Angabe von ; die Leistung im Vergleich zu    herabgesetzt, da mehr <b>rsh</b> -Befehle erforderlich werden.
]	Schaltet die Punktdateiausführung des Profils des Benutzers vor die Ausführung des Befehls.
	<b>Anmerkung:</b> Nur für auf UNIX basierende Plattformen verfügbar.
}	Schaltet die Punktdateiausführung der in \$RAHENV definierten Datei (wahrscheinlich .kshrc) vor die Ausführung des Befehls.
	<b>Anmerkung:</b> 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.
	<b>Anmerkung:</b> Nur für auf UNIX basierende Plattformen verfügbar.
)	Unterdrückt die Ausführung des Benutzerprofils und der in \$RAHENV definierten Datei.
	<b>Anmerkung:</b> Nur für auf UNIX basierende Plattformen verfügbar.
'	Meldet den Befehlsaufruf zurück an die Maschine.
<	Sendet an alle Maschinen außer dieser.
<<- <i>nnn</i> <	Sendet an alle außer Datenbankpartitionsserver <i>nnn</i> (d. h. alle Datenbankpartitionsserver in der Datei db2nodes.cfg außer dem mit der Knotennummer <i>nnn</i> . Siehe den ersten Abschnitt nach der letzten Präfixsequenz in dieser Tabelle).
<<+ <i>nnn</i> <	Sendet nur an Datenbankpartitionsserver <i>nnn</i> (der Datenbankpartitionsserver in der Datei db2nodes.cfg, dessen Knotennummer <i>nnn</i> lautet. Siehe den ersten Abschnitt nach der letzten Präfixsequenz in dieser Tabelle).
<b>(Leerzeichen)</b>	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 <b>rah</b> -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: <pre>rah ';' mydaemon'</pre> 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.

2. Wenn " angegeben wird, werden mehrfache Vorkommen aus der Liste der Maschinen nicht eliminiert.

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.

#### **Zugehörige Konzepte:**

- „Angaben der Befehle rah und db2\_all“ auf Seite 383
- „Paralleles Ausführen von Befehlen auf Plattformen unter UNIX“ auf Seite 384

#### **Zugehörige Referenzen:**

- „Beschreibungen der Befehle rah und db2\_all“ auf Seite 382

---

## **Angaben der Liste von Maschinen in einer partitionierten Umgebung**

#### **Vorgehensweise:**

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**.

**Zugehörige Tasks:**

- „Eliminieren doppelter Einträge aus einer Liste von Maschinen in einer partitionierten Umgebung“ auf Seite 390

---

## Eliminieren doppelter Einträge aus einer Liste von Maschinen in einer partitionierten Umgebung

**Vorgehensweise:**

Wenn Sie mit DB2 Universal Database Enterprise Server 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-Shellsyntax)
```

Dabei ist *nnn* die Knotennummer, die der entsprechenden Zeile in der Datei `db2nodes.cfg` entnommen ist, so dass der Befehl an den gewünschten Datenbankpartitionsserver 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 Datenbankpartitionsserver 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.

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 Datenbankpartitionsserver und nicht an alle Maschinen impliziert.

**Zugehörige Tasks:**

- „Angeben der Liste von Maschinen in einer partitionierten Umgebung“ auf Seite 389

**Zugehörige Referenzen:**

- „RESTART DATABASE Command“ in *Command Reference*
- „Präfixsequenzen für den Befehl rah“ auf Seite 387

---

## Steuern des Befehls rah

Mit Hilfe der folgenden Umgebungsvariablen können Sie den Befehl **rah** steuern.

*Tabelle 51.*

Name	Bedeutung	Standardwert
\$RAHBUFDIR <b>Anmerkung:</b> Nur für UNIX-Plattformen verfügbar.	Verzeichnis für den Puffer	/tmp/\$USER
\$RAHBUFNAME <b>Anmerkung:</b> Nur für UNIX-Plattformen verfügbar.	Dateiname für den Puffer	rahout
\$RAHOSTFILE (auf UNIX-Plattformen); RAHOSTFILE (Windows NT)	Datei mit der Liste der Hosts	db2nodes.cfg
\$RAHOSTLIST (auf UNIX-Plattformen); RAHOSTLIST (Windows NT)	Liste der Hosts in Form einer Zeichenfolge	extrahiert aus \$RAHOSTFILE
\$RAHCHECKBUF <b>Anmerkung:</b> Nur für UNIX-Plattformen verfügbar.	Wenn auf "no", Prüfungen übergehen	Nicht definiert
\$RAHSLEEPTIME (auf UNIX-Plattformen); RAHSLEEPTIME (Windows NT)	Zeit in Sekunden, die diese Prozedur auf die erste Ausgabe parallel ausgeführter Befehle wartet	86400 Sekunden für <b>db2_kill</b> , 200 Sekunden für alle anderen

Tabelle 51. (Forts.)

Name	Bedeutung	Standardwert
\$RAHWAITTIME (auf UNIX-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 &lt;pid&gt; ... 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 <b>rah</b> bei der Feststellung von Jobbeendigungen nicht auf diese Prüfungen angewiesen ist.</p>	45 Sekunden
\$RAHENV <b>Anmerkung:</b> Nur für UNIX-Plattformen verfügbar.	Gibt den Dateinamen an, der auszuführen ist, wenn \$RAHDOTFILES=E oder K oder PE oder B ist.	\$ENV
\$RAHUSER (auf UNIX-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> <p><b>Anmerkung:</b> Auf Plattformen unter UNIX wird der Wert von \$RAHENV von dort, wo <b>rah</b> ausgeführt wird, verwendet, nicht der Wert (falls vorhanden), der von der fernen Shell definiert wurde.</p> <p><b>Zugehörige Referenzen:</b></p> <ul style="list-style-type: none"> <li>• „Verwenden von \$RAHDOTFILES auf UNIX-Plattformen“ auf Seite 392</li> </ul>	\$USER

## Verwenden von \$RAHDOTFILES auf UNIX-Plattformen

**Anmerkung:** Die Informationen dieses Abschnitts beziehen sich ausschließlich auf Plattformen, die auf UNIX basieren.

Im Folgenden werden die Punktdateien (.dateien) 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 Anmeldeshell keine Korn-Shell ist, werden alle Punktdateien, die Sie zur Ausführung angeben, in einem Korn-Shellprozess ausgeführt und müssen daher der Korn-Shell-Syntax entsprechen. Wenn Ihre Anmeldeshell 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-Shelldatei INSTHOME/.profile zu Ihrer Datei .cshrc und geben in Ihrer Datei INSTHOME/.cshrc Folgendes an:

```
setenv RAHDOTFILES P
```

Alternativ können Sie für die Datei .cshrc in der Korn-Shell ein Äquivalent INSTHOME/.kshrc erstellen und in der Datei INSTHOME/.cshrc Folgendes angeben:

```
setenv RAHDOTFILES E
setenv RAHENV INSTHOME/.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 (TTY) 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
```

#### Zugehörige Referenzen:

- „Steuern des Befehls rah“ auf Seite 391

---

## Einstellen des Standardumgebungsprofils für rah unter Windows NT

#### Vorgehensweise:

**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 haben:

```
; 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.

#### Zugehörige Konzepte:

- „Angabe der Befehle rah und db2\_all“ auf Seite 383

---

## Bestimmen 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 dass `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_all "....."
```

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.

#### Zugehörige Referenzen:

- „Steuern des Befehls rah“ auf Seite 391



---

## Anhang E. Verwenden der Unterstützung für Windows Management Instrumentation (WMI)

---

### Einführung in Windows Management Instrumentation (WMI)

Eine Initiative der Industrie bemüht sich um die Einrichtung von Standards der Managementinfrastruktur und stellt eine Methode bereit, Informationen aus verschiedenen Hard- und Softwareverwaltungssystemen zu kombinieren. Diese Initiative trägt den Namen Web-Based Enterprise Management (WBEM). WBEM basiert auf dem CIM-Schema (Common Information Model), das ein von der Desktop Management Task Force (DMTF) propagierter Industriestandard ist.

Microsoft® Windows® Management Instrumentation (WMI) stellt eine Implementierung der WBEM-Initiative für unterstützte Windows-Plattformen dar. WMI ist nützlich in einem Windows-Unternehmensnetzwerk, in dem WMI den Wartungsaufwand und die Kosten zur Verwaltung von Netzwerkkomponenten des Unternehmens verringert. WMI stellt Folgendes bereit:

- Ein konsistentes Modell für den Betrieb, die Konfiguration und den Status von Windows
- Eine COM-API für den Zugriff auf Managementinformationen
- Die Möglichkeit zur Arbeit mit anderen Windows-Managementservices
- Eine flexible und erweiterbare Architektur, die Lieferanten die Möglichkeit gibt, weitere WMI-Provider zu schreiben, um neue Geräte, Anwendungen und andere Weiterentwicklungen zu unterstützen
- Die WMI Query Language (WQL) zur Erstellung detaillierter Abfragen der Informationen
- Eine API für Entwickler von Verwaltungsanwendungen zum Schreiben von Visual Basic- oder Windows Scripting Host-Skripts (WSH-Skripts)

Die WMI-Architektur besteht aus zwei Teilen:

1. Eine Managementinfrastruktur, die den CIM-Objektmanager (CIMOM - CIM Object Manager) und einen zentralen Speicherbereich für Verwaltungsdaten enthält, der als CIMOM-Objektrepository bezeichnet wird. CIMOM ermöglicht Anwendungen eine einheitliche Methode zum Zugriff auf Managementdaten.
2. WMI-Provider. Die so genannten WMI-Provider sind Vermittlungselemente zwischen CIMOM und verwalteten Objekten. Über die WMI-APIs versorgen WMI-Provider CIMOM mit Daten aus verwalteten Objekten, verarbeiten Anforderungen im Auftrag von Verwaltungsanwendungen und generieren Ereignisbenachrichtigungen.

WMI-Provider sind Standard-COM- oder DCOM-Server, die als Mittler zwischen verwalteten Objekten und dem CIM-Objektmanager fungieren. Wenn der CIMOM eine Anforderung aus einer Verwaltungsanwendung für Daten, die im CIMOM-Objektrepository nicht verfügbar sind, oder für Ereignisse empfängt, leitet der CIMOM die Anweisung an die WMI-Provider weiter. WMI-Provider liefern Daten und Ereignisbenachrichtigungen für verwaltete Objekte, die für ihren speziellen Einsatzbereich spezifisch sind.

#### Zugehörige Konzepte:

- „Integration von DB2 Universal Database in Windows Management Instrumentation“ auf Seite 398

---

## Integration von DB2 Universal Database in Windows Management Instrumentation

Auf die Überwachungsprogramme (Monitore) für Momentaufnahmen kann durch Windows<sup>®</sup> Management Instrumentation (WMI) über DB2<sup>®</sup>-Leistungsdatenquellen und mit Hilfe des integrierten PerfMon-Providers zugegriffen werden.

Auf die DB2-Profilregistriervariablen kann durch WMI über den integrierten Registrierprovider zugegriffen werden.

Das WMI Software Development Kit (WMI SDK) enthält verschiedene integrierte Provider:

- PerfMon-Provider
- Registrierereignisprovider
- Registrierprovider
- Windows NT<sup>®</sup>-Ereignisprotokollprovider
- Win32-Provider
- WDM-Provider

Auf die DB2-Fehler, die in den Ereignisprotokollen aufgezeichnet werden, kann durch WMI über den integrierten Windows NT-Ereignisprotokollprovider zugegriffen werden.

DB2 Universal Database<sup>™</sup> (UDB) verfügt über einen DB2-WMI-Verwaltungsprovider sowie über WMI-Beispielscriptdateien, die einen Zugriff auf die folgenden verwalteten Objekte ermöglichen:

1. Exemplare des Datenbankservers, einschließlich partitionierter Exemplare. Die folgenden Operationen können ausgeführt werden:
  - Aufzählen von Exemplaren
  - Konfigurieren von Datenbankmanagerparametern
  - Starten, Stoppen und Abfragen des Status des DB2-Serverdienstes
  - Konfigurieren oder Einrichten der Kommunikation
2. Datenbanken. Die folgenden Operationen können ausgeführt werden:
  - Aufzählen von Datenbanken
  - Konfigurieren von Datenbankparametern
  - Erstellen und Löschen von Datenbanken
  - Sichern, Wiederherstellen und aktualisierendes Wiederherstellen von Datenbanken

Sie müssen den DB2-WMI-Provider im System registrieren, bevor Sie WMI-Anwendungen ausführen. Die Registrierung geschieht durch die Eingabe der folgenden Befehle:

- `mofcomp %DB2PATH%\bin\db2wmi.mof`  
Dieser Befehl lädt die Definition des DB2-WMI-Schemas in das System.
- `regsvr %DB2PATH%\bin\db2wmi.dll`  
Dieser Befehl registriert die COM-DLL-Datei für den DB2-WMI-Provider unter Windows.

In beiden Befehlen ist %DB2PATH% der Pfad, in dem DB2 installiert ist. Die Datei db2wmi.mof ist die .MOF-Datei, die die DB2-WMI-Schemadefinition enthält.

Die Integration in die WMI-Infrastruktur hat verschiedene Vorteile:

1. Mit Hilfe des durch WMI bereitgestellten Tools können Sie problemlos Scripts zur Verwaltung von DB2-Servern in einer Windows-basierten Umgebung schreiben. Visual Basic-Beispielscripts (VBS-Scripts) werden zur Verfügung gestellt, um einfache Aufgaben wie das Auflisten von Exemplaren, Erstellen und Löschen von Datenbanken und das Aktualisieren von Konfigurationsparametern auszuführen. Die Beispielscripts sind im Produkt DB2 Application Development für Windows enthalten.
2. Sie können leistungsstarke Verwaltungsanwendungen erstellen, die viele Aufgaben über WMI ausführen. Solche Aufgaben könnten zum Beispiel folgende sein:
  - Anzeigen von Systeminformationen
  - Überwachen der DB2-Leistung
  - Überwachen der Systemressourcennutzung durch DB2

Durch Überwachen sowohl von Systemereignissen als auch von DB2-Ereignissen mit Hilfe dieser Art von Verwaltungsanwendung können Sie eine Datenbank besser verwalten.

3. Sie können vorhandene Kenntnisse und Fähigkeiten der COM- und Visual Basic-Programmierung nutzen. Durch die Bereitstellung einer COM- bzw. Visual Basic-Schnittstelle können Ihre Programmierer bei der Entwicklung von Verwaltungsanwendungen für das Unternehmen Zeit sparen.

**Zugehörige Konzepte:**

- „Einführung in Windows Management Instrumentation (WMI)“ auf Seite 397





---

## Anhang F. Verwenden der Windows NT-Sicherheit

---

### Einführung in DB2 für Windows NT und in die Windows NT-Sicherheit

Eine Windows<sup>®</sup> NT-Domäne ist eine Anordnung von Client- und Servercomputern, die über einen bestimmten und eindeutigen Namen angesprochen werden und eine einzige Benutzerkontendatenbank gemeinsam verwenden, die als Security Access Manager (SAM) bezeichnet wird. Einer der Computer in der Domäne ist der Domänencontroller. Der Domänencontroller verwaltet alle Aspekte der Interaktionen zwischen den Benutzern und der Domäne. Mit Hilfe der Informationen in der Benutzerkontendatenbank der Domäne authentifiziert der Domänencontroller Benutzer, die sich über Domänenkonten anmelden. Für jede Domäne ist ein Domänencontroller der primäre Domänencontroller (PDC - Primary Domain Controller). Innerhalb der Domäne kann es außerdem Sicherungsdomänencontroller (BDC - Backup Domain Controller) geben, welche die Authentifizierung von Benutzerkonten durchführen, wenn kein primärer Domänencontroller vorhanden oder der primäre Domänencontroller nicht verfügbar ist. Sicherungsdomänencontroller enthalten eine Kopie der SAM-Datenbank, die regelmäßig mit der Masterkopie auf dem primären Domänencontroller synchronisiert wird.

Benutzerkonten, Benutzer-IDs und Kennwörter müssen nur auf dem primären Domänencontroller definiert werden, um auf die Ressourcen der Domäne zugreifen zu können.

Im Rahmen des Setup-Programms bei der Installation eines Windows NT<sup>®</sup>-Servers können Sie folgende Erstellungsoptionen auswählen:

- Primärer Domänencontroller in einer neuen Domäne
- Sicherungsdomänencontroller in einer bekannten Domäne
- Alleinstehender Server in einer bekannten Domäne

Durch die Auswahl „Controller“ in einer neuen Domäne wird dieser Server als primärer Domänencontroller eingerichtet.

Der Benutzer kann sich an der lokalen Maschine oder, wenn die Maschine in einer Windows NT-Domäne installiert ist, an der Domäne anmelden. DB2<sup>®</sup> für Windows NT unterstützt beide Möglichkeiten. Zur Authentifizierung des Benutzers prüft DB2 zuerst die lokale Maschine, dann den Domänencontroller für die aktuelle Domäne und schließlich alle dem Domänencontroller bekannten gesicherten (vertrauten) Domänen.

Nehmen Sie zur Veranschaulichung einmal an, dass ein DB2-Exemplar die Authentifizierung auf dem Server voraussetzt. Die Konfiguration sieht folgendermaßen aus:

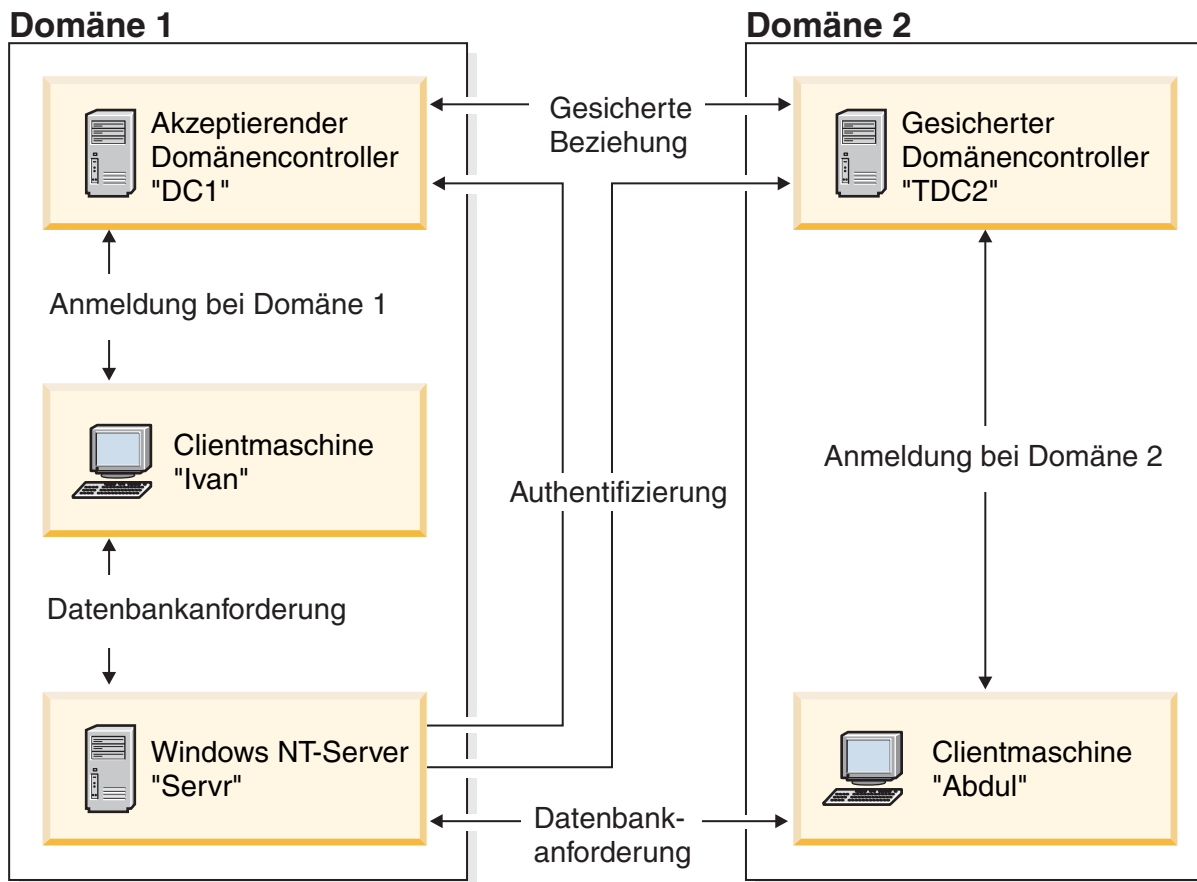


Abbildung 6. Authentifizierung in Windows NT-Domänen

Jede Maschine verfügt über eine SAM-Sicherheitsdatenbank (Security Access Management), sofern es sich nicht um eine Clientmaschine unter Windows 9x handelt. Windows 9x-Maschinen haben keine SAM-Datenbank. DC1 ist der Domänencontroller, auf dem die Clientmaschine Ivan sowie Servr, der Server mit DB2 für Windows NT, registriert sind. TDC2 ist für DC1 eine gesicherte ("Trusted") Domäne, und die Clientmaschine Abdul ist ein Mitglied der Domäne von TDC2.

#### Zugehörige Konzepte:

- „Gruppen- und Benutzerauthentifizierung unter Windows NT“ auf Seite 406

#### Zugehörige Tasks:

- „Verwenden eines Sicherungsdomänencontrollers mit DB2 UDB“ auf Seite 405
- „Installieren von DB2 auf einem Sicherungsdomänencontroller“ auf Seite 408
- „DB2 für Windows NT-Authentifizierung mit Gruppen und Domänensicherheit“ auf Seite 409

## Ein DB2 für Windows NT-Szenario mit Serverauthentifizierung

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 den Domänencontroller (DC1), bevor sie mit anderen gesicherten Domänen fortfährt. Der Benutzername Abdul wird auf TDC2 gefunden. Diese Suchreihenfolge setzt einen einzigen Namensbereich (Namespace) 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.

**Zugehörige Konzepte:**

- „Einführung in DB2 für Windows NT und in die Windows NT-Sicherheit“ auf Seite 401

## Ein DB2 für Windows NT-Szenario mit Clientauthentifizierung und einer Windows NT-Clientmaschine

1. Dale, der Administrator, meldet sich an SRV3 an und ändert den Authentifizierungstyp für das Datenbankexemplar in CLIENT:
 

```
db2 update dbm cfg using authentication client
db2stop
db2start
```
2. Ivan meldet sich von der Windows-Clientmaschine 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 Domänensteuereinheit (DC1), bevor sie mit anderen gesicherten 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 Sicherheitsservice wird im Rahmen der Windows-Installation installiert. DB2 wird anschließend installiert und als Windows NT-Dienst (Service) „registriert“, jedoch nicht automatisch gestartet. Geben Sie den Befehl NET START DB2NTSECSERVER ein, um den DB2-Sicherheitsdienst zu starten.

**Zugehörige Konzepte:**

- „Einführung in DB2 für Windows NT und in die Windows NT-Sicherheit“ auf Seite 401

---

## Ein DB2 für Windows NT-Szenario mit Clientauthentifizierung und einer Windows 9x-Clientmaschine

1. Dale, der Administrator, meldet sich an SRV3 an und ändert den Authentifizierungstyp für das Datenbankexemplar in CLIENT:  

```
db2 update dbm cfg using authentication client
db2stop
db2start
```
2. Ivan meldet sich von einer Windows 9x-Clientmaschine 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 9x-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 9x-Client einen verwendeten Benutzernamen und das Kennwort nicht überprüfen kann, ist eine Authentifizierung auf dem Client unter Windows 9x als solche nicht sicher. Wenn die Windows 9x-Maschine Zugriff auf einen Windows NT-Sicherheitsanbieter hat, kann ein gewisses Maß an Sicherheit erreicht werden, indem das Windows 9x-System für überprüfte Durchlassanmeldungen (Pass-through) konfiguriert wird. Einzelheiten zur Konfiguration eines Windows 9x-Systems in dieser Weise finden Sie in der Microsoft-Dokumentation zu Windows 9x.

### Zugehörige Konzepte:

- „Einführung in DB2 für Windows NT und in die Windows NT-Sicherheit“ auf Seite 401
- „Unterstützung globaler Gruppen (unter Windows)“ auf Seite 404

---

## Unterstützung globaler Gruppen (unter Windows)

DB2<sup>®</sup> Universal Database (DB2 UDB) unterstützt auch globale Gruppen. Zur Nutzung globaler Gruppen müssen Sie die gewünschten globalen Gruppen als Mitglieder in eine lokale Gruppe einfügen. Wenn DB2 UDB alle Gruppen aufzählt, deren Mitglied eine Person ist, führt DB2 UDB 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).

Globale Gruppen können auf zwei Arten verwendet werden:

- Als Mitglieder in einer lokalen Gruppe. Berechtigungen müssen dieser lokalen Gruppe erteilt werden.
- Als eigenständige Gruppe auf einem Domänencontroller. Berechtigungen müssen der globalen Gruppe erteilt werden.

### Zugehörige Konzepte:

- „Gruppen- und Benutzerauthentifizierung unter Windows NT“ auf Seite 406

---

## Verwenden eines Sicherungsdomänencontrollers mit DB2 UDB

### Vorgehensweise:

Wenn der von Ihnen für DB2 Universal Database™ (DB2 UDB) verwendete Server auch als Sicherungsdomänencontroller fungiert, können Sie die DB2 UDB-Leistung erhöhen und den Datenverkehr auf dem Netzwerk verringern, wenn Sie DB2 UDB zur Verwendung des Sicherungsdomänencontrollers konfigurieren.

Der Sicherungsdomänencontroller wird für DB2 UDB mit Hilfe der Registriervariablen DB2DMNBCKCTLR definiert.

Wenn Sie den Namen der Domäne kennen, für die der DB2 UDB-Server als Sicherungsdomänencontroller fungiert, verwenden Sie folgende Angabe:

```
db2dmnbckctlr=<domänenname>
```

Dabei muss domänenname in Großbuchstaben angegeben werden.

Wenn DB2 UDB die Domäne bestimmen soll, für die die lokale Maschine ein Sicherungsdomänencontroller ist, geben Sie Folgendes an:

```
DB2DMNBCKCTLR=?
```

**Anmerkung:** DB2 UDB verwendet einen vorhandenen Sicherungsdomänencontroller nicht standardmäßig, weil die Synchronisierung eines Sicherungsdomänencontrollers mit dem primären Domänencontroller verloren gehen und so eine Sicherheitslücke entstehen könnte. Domänencontroller können ihre Synchronisierung verlieren, wenn die Sicherheitsdatenbank des primären Domänencontrollers aktualisiert wird, aber die Änderungen nicht an einen Sicherungsdomänencontroller weitergegeben werden. Dies kann eintreten, wenn es Netzlatenzzeiten gibt oder wenn der Computersuchdienst nicht aktiv ist.

### Zugehörige Tasks:

- „Installieren von DB2 auf einem Sicherungsdomänencontroller“ auf Seite 408

---

## 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 für DB2 für Windows NT-Benutzer- und Gruppennamen“
- „DB2 für Windows NT-Sicherheitsservice“ auf Seite 407
- „Installieren von DB2 auf einem Sicherungsdomänencontroller“ auf Seite 408
- „DB2 für Windows NT-Authentifizierung mit Gruppen und Domänensicherheit“ auf Seite 409

## Einschränkungen für DB2 für Windows NT-Benutzer- und Gruppennamen

Im Folgenden sind die für diese Umgebung geltenden Einschränkungen aufgeführt:

- Die Länge von Benutzernamen und Gruppennamen ist unter DB2 Universal Database™ (DB2 UDB) auf maximal 30 Zeichen beschränkt.

- 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 UDB 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. Dabei müssen die Objektnamen in Anführungszeichen eingeschlossen werden. Alternativ können Sie auch ODBC-Front-End-Tools anderer Hersteller verwenden.)
- Ein Benutzer kann nicht mehr als 64 Gruppen angehören.
- DB2 UDB unterstützt einen einzigen Namensbereich (Namespace). Das heißt für eine Umgebung mit gesicherten (vertrauten) Domänen, dass Sie kein Benutzerkonto desselben Namens haben sollten, das in mehreren Domänen vorhanden ist oder das in der lokalen SAM-Datenbank der Servermaschine und in einer anderen Domäne vorhanden ist.

**Zugehörige Konzepte:**

- „Gruppen- und Benutzerauthentifizierung unter Windows NT“ auf Seite 406
- „Vertrauensstellungen zwischen Domänen unter Windows NT“ auf Seite 407

## Gruppen- und Benutzerauthentifizierung unter Windows NT

Benutzer werden unter Windows<sup>®</sup> NT definiert, indem Benutzerkonten über ein Windows NT<sup>®</sup>-Verwaltungstool mit dem Namen „Benutzer-Manager“ erstellt werden.

Eine Gruppe ist ein Konto, das andere Konten enthält, die in diesem Fall als Mitglieder bezeichnet werden. Gruppen bieten Windows NT-Administratoren die Möglichkeit, Benutzern einer Gruppe gleichzeitig Zugriffsrechte und Berechtigungen zu erteilen, ohne jeden Benutzer einzeln verwalten zu müssen. Gruppen werden ebenso wie Benutzerkonten in der SAM-Datenbank (SAM - Security Access Manager) definiert und gepflegt.

Es gibt zwei Arten von Gruppen:

- Lokale Gruppen. Eine lokale Gruppe kann Benutzerkonten enthalten, die in der lokalen Kontendatenbank erstellt wurden. Wenn sich die lokale Gruppe auf einer Maschine befindet, die zu einer Domäne gehört, kann die lokale Gruppe außerdem Domänenkonten und Gruppen aus der Windows NT-Domäne enthalten. Wenn die lokale Gruppe auf einer Workstation erstellt wird, ist sie für diese Workstation spezifisch.
- Globale Gruppen. Eine globale Gruppe ist nur auf einem Domänencontroller vorhanden und enthält Benutzerkonten der SAM-Datenbank der Domäne. Das heißt, eine globale Gruppe kann nur Benutzerkonten aus der Domäne enthalten, in der sie erstellt wurde. Sie kann keine anderen Gruppen als Mitglieder enthalten. Eine globale Gruppe kann in Servern und Workstations der eigenen Domäne und auch in vertrauenden Domänen verwendet werden.

**Zugehörige Konzepte:**

- „Vertrauensstellungen zwischen Domänen unter Windows NT“ auf Seite 407
- „Unterstützung globaler Gruppen (unter Windows)“ auf Seite 404

**Zugehörige Tasks:**

- „DB2 für Windows NT-Authentifizierung mit Gruppen und Domänensicherheit“ auf Seite 409

**Zugehörige Referenzen:**

- „Einschränkungen für DB2 für Windows NT-Benutzer- und Gruppennamen“ auf Seite 405

## Vertrauensstellungen zwischen Domänen unter Windows NT

Vertrauensstellungen stellen eine Verwaltungs- und Kommunikationsverbindung zwischen zwei Domänen dar. Eine Vertrauensstellung zwischen zwei Domänen stellt die Möglichkeit bereit, Benutzerkonten und globale Gruppen in einer anderen Domäne zu verwenden als der, in der die Konten definiert sind. Konteninformationen werden gemeinsam verwendet, um die Gültigkeit von Zugriffsrechten und Berechtigungen von Benutzerkonten und globalen Gruppen ohne erneute Authentifizierung zu bestätigen. Vertrauensstellungen vereinfachen die Benutzerverwaltung, indem sie zwei oder mehr Domänen zu einer administrativen Einheit verbinden.

In einer Vertrauensstellung befinden sich zwei Domänen:

- Die vertrauende Domäne. Diese Domäne vertraut der anderen Domäne in Bezug auf die Authentifizierung von Benutzern für beide Domänen.
- Die vertraute (gesicherte) Domäne. Diese Domäne authentifiziert Benutzer im Auftrag (im Vertrauen) einer anderen Domäne.

Vertrauensstellungen sind nicht transitiv. Dies bedeutet, dass in beide Richtungen zwischen den Domänen explizite Vertrauensstellungen eingerichtet werden müssen. Zum Beispiel muss die vertrauende Domäne nicht unbedingt auch eine vertraute Domäne sein.

**Zugehörige Konzepte:**

- „Gruppen- und Benutzerauthentifizierung unter Windows NT“ auf Seite 406
- „Unterstützung globaler Gruppen (unter Windows)“ auf Seite 404

**Zugehörige Referenzen:**

- „Einschränkungen für DB2 für Windows NT-Benutzer- und Gruppennamen“ auf Seite 405

## DB2 für Windows NT-Sicherheitsservice

In DB2<sup>®</sup> Universal Database (DB2 UDB) wurde die Funktion zum Authentifizieren von Benutzernamen und Kennwörtern in den DB2-Systemcontroller integriert. Der Sicherheitsservice wird nur für die Verbindung eines Clients zu einem Server benötigt, für den der Authentifizierungstyp CLIENT konfiguriert wurde.

**Zugehörige Konzepte:**

- „Einführung in DB2 für Windows NT und in die Windows NT-Sicherheit“ auf Seite 401

## Installieren von DB2 auf einem Sicherungsdomänencontroller

### Vorgehensweise:

In einer Windows NT 4.0-Umgebung kann ein Benutzer entweder auf einem primären Domänencontroller oder einem Sicherungsdomänencontroller authentifiziert werden. Diese Funktion ist in umfangreichen, verteilten LANs mit einem zentralen primären Domänencontroller (PDC) und einem oder mehreren Sicherungsdomänencontrollern (BDC = Backup Domain Controller) an allen Standorten sehr wichtig. Die Benutzer können dann auf dem Sicherungsdomänencontroller ihres Standortes authentifiziert werden. Auf diese Weise ist es nicht erforderlich, einen Aufruf zur Authentifizierung an den primären Domänencontroller abzusetzen.

Der Vorteil von Sicherungsdomänencontrollern liegt in diesem Fall in der schnelleren Authentifizierung der Benutzer und der geringeren Belastung des LANs im Vergleich zu einer Konfiguration ohne Sicherungsdomänencontroller.

Unter folgenden Bedingungen ist eine Authentifizierung durch den Sicherungsdomänencontroller möglich:

- Der DB2 für Windows NT-Server wurde auf dem Sicherungsdomänencontroller installiert.
- Die Variable DB2DMNBCKCTRL der Profilregistrierdatenbank wurde auf den entsprechenden Wert gesetzt.

Wurde die Variable DB2DMNBCKCTRL der Profilregistrierdatenbank nicht gesetzt oder wurde ein Nullwert angegeben, führt DB2 für Windows NT die Authentifizierung auf dem primären Domänencontroller 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 Sicherungsdomänencontroller 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 Servermanager zeigt an, dass der Sicherungsdomänencontroller aktiv und verfügbar ist. (D. h., das Symbol für diese Maschine wird nicht abgeblendet dargestellt.)
- Die Registrierdatenbank für den DB2 Windows NT-Server gibt an, dass das System ein Sicherungsdomänencontroller in der angegebenen Domäne ist.

Unter normalen Bedingungen kann die Einstellung DB2DMNBCKCTRL=? 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 Computersuchdienst muss aktiv sein, um diese Informationen korrekt und auf dem neuesten Stand zu halten. Auf umfangreichen LANs wird der Computersuchdienst möglicherweise nicht ausgeführt, so dass die Servermanager-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 Sicherungsdomänencontroller auszuführen: Setzen Sie



DB2DMNBCKCTRL=xxx, wobei xxx für den Windows NT-Domänennamen des DB2-Servers steht. Mit dieser Einstellung wird die Authentifizierung auf dem Sicherungsdomänencontroller unter 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 Sicherungsdomänencontroller für die angegebene Domäne konfiguriert. (Wenn die Maschine als Sicherungsdomänencontroller für eine andere Domäne konfiguriert ist, führt diese Einstellung zu einem Fehler.)

#### Zugehörige Tasks:

- „Verwenden eines Sicherungsdomänencontrollers mit DB2 UDB“ auf Seite 405

## DB2 für Windows NT-Authentifizierung mit Gruppen und Domänensicherheit

#### Vorgehensweise:

DB2 Universal Database™ (DB2 UDB) gibt Ihnen die Möglichkeit, beim Erteilen von Zugriffsrechten oder Definieren von Berechtigungsstufen eine lokale Gruppe oder eine globale Gruppe anzugeben. Ein Benutzer wird als Mitglied einer Gruppe erkannt, wenn das Konto des Benutzers explizit in der lokalen bzw. globalen Gruppe oder implizit durch seine Mitgliedschaft in einer globalen Gruppe definiert ist, die wiederum als Mitglied einer lokalen Gruppe definiert ist.

DB2 für Windows NT unterstützt die folgenden Gruppenarten:

- Lokale Gruppen
  - Globale Gruppen
  - Globale Gruppen als Mitglieder einer lokalen Gruppe
- DB2 für Windows NT listet die lokalen und globalen Gruppen auf, zu denen der Benutzer gehört, und verwendet hierzu die Sicherheitsdatenbank, in der der Benutzer lokalisiert wurde. DB2 UDB stellt eine Überschreibungsfunktion zur Verfügung, durch die die Ausführung der Gruppenuflistung auf dem lokalen Windows NT-Server erzwungen wird, auf dem DB2 UDB installiert ist. Hierbei spielt es keine Rolle, wo das Benutzerkonto lokalisiert wurde. Diese Überschreibung kann mit den folgenden Befehlen veranlasst werden:

– Für globale Einstellungen:

```
db2set -g DB2_GRP_LOOKUP=local
```

– Für Exemplareinstellungen:

```
db2set -i <exemplarname> DB2_GRP_LOOKUP=local
```

Nach der Ausführung dieses Befehls müssen Sie das DB2 UDB-Exemplar stoppen und erneut starten, um die Änderung in Kraft zu setzen. Erstellen Sie anschließend lokale Gruppen und fügen Sie Domänenkonten oder globale Gruppen in die lokale Gruppe als Mitglieder ein.

Geben Sie Folgendes ein, um alle definierten Variablen der DB2-Profilregistrierdatenbank anzuzeigen:

```
db2set -all
```

Wenn die Variable DB2\_GRP\_LOOKUP der Profilregistrierdatenbank auf den Wert 'local' gesetzt ist, versucht DB2 UDB 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 UDB versucht **nicht**, den Benutzer auf einer anderen Maschine in der Domäne oder auf den verfügbaren Domänencontrollern zu finden.

Wenn die Variable DB2\_GRP\_LOOKUP der Profilregistrierdatenbank nicht gesetzt wurde, gilt Folgendes:

1. DB2 UDB versucht zunächst, 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 UDB, den Benutzernamen in der zugehörigen Domäne und anschließend in den gesicherten (vertrauten) Domänen zu lokalisieren.

Wenn DB2 UDB auf einer Maschine ausgeführt wird, die als primärer Domänencontroller (PDC) oder Sicherungsdomänencontroller (BDC) in der Ressourcendomäne eingesetzt ist, kann DB2 UDB jeden Domänencontroller in jeder beliebigen gesicherten (vertrauten) Domäne lokalisieren. Der Grund dafür ist der, dass die Namen der Domänen von Sicherungsdomänencontrollern in gesicherten Domänen einer Maschine nur bekannt sind, wenn sie ein Domänencontroller ist.

Wenn DB2 UDB nicht auf einem Domänencontroller ausgeführt wird, geben Sie folgenden Befehl ein:

```
db2set -g DB2_GRP_LOOKUP=DOMAIN
```

Dieser Befehl weist DB2 UDB an, einen Domänencontroller in der eigenen Domäne zu verwenden, um den Namen eines Domänencontrollers in der Kontendomäne ausfindig zu machen. Das heißt, wenn DB2 UDB ermittelt, dass ein bestimmtes Benutzerkonto in Domäne x definiert ist, versucht DB2 nicht, einen Domänencontroller für Domäne x zu finden, sondern sendet die entsprechende Anforderung an einen Domänencontroller in der eigenen Domäne. Der Name des Domänencontrollers in der Kontendomäne wird gefunden und an die Maschine zurückgegeben, auf der DB2 UDB ausgeführt wird. Diese Methode bieten zwei Vorteile:

1. Es wird ein Sicherungsdomänencontroller gefunden, wenn der primäre Domänencontroller nicht verfügbar ist.
2. Es wird ein Sicherungsdomänencontroller gefunden, der sich in der Nähe befindet, wenn sich der primäre Domänencontroller an einem geographisch fernen Ort befindet.

#### **Zugehörige Konzepte:**

- „Gruppen- und Benutzerauthentifizierung unter Windows NT“ auf Seite 406

## **Authentifizierung mit einer geordneten Domänenliste**

Benutzer-IDs können mehr als einmal in einer gesicherten (vertrauten) Domänengesamtstruktur definiert sein. Eine gesicherte Domänengesamtstruktur ist eine Gruppe von Domänen, die durch ein Netzwerk miteinander verbunden sind. Es ist möglich, dass ein Benutzer in einer Domäne die gleiche Benutzer-ID wie ein anderer Benutzer in einer anderen Domäne besitzt. Dies kann beim Versuch einer der folgenden Aktionen zu Schwierigkeiten führen:

- Authentifizierung mehrerer Benutzer, die die gleiche Benutzer-ID, jedoch in verschiedenen Domänen haben

- Gruppensuche zum Zweck des Erteilens und Widerrufens von Zugriffsrechten für Gruppen
- Überprüfung von Kennwörtern
- Steuerung des Datenaustauschs im Netzwerk

#### **Vorgehensweise:**

Zur Vermeidung von Schwierigkeiten aufgrund der möglichen Existenz mehrerer Benutzer mit der gleichen Benutzer-ID in einer Domänengesamtstruktur sollten Sie eine geordnete Domänenliste verwenden, wie sie in der Registriervariablen DB2DOMAINLIST mit Hilfe des Befehls **db2set** definiert werden kann. Bei der Definition der Reihenfolge müssen die Domänen, die in die Liste aufgenommen werden sollen, durch Kommas voneinander getrennt werden. Sie müssen eine bewusste Entscheidung im Hinblick auf die Reihenfolge treffen, in der die Domänen bei der Authentifizierung von Benutzern durchsucht werden.

Die Benutzer-IDs, die sich in Domänen weiter hinten in der Liste befinden, müssen von Ihnen umbenannt werden, wenn Sie für den Zugriff authentifiziert werden sollen.

Eine Steuerung des Zugriffs kann über die Domänenliste erfolgen. Wenn zum Beispiel die Domäne eines Benutzers nicht in der Liste enthalten ist, erhält der Benutzer keine Berechtigung zur Herstellung einer Verbindung.

**Anmerkung:** Die Registriervariable DB2DOMAINLIST ist nur wirksam, wenn in der Datenbankmanagerkonfiguration die Clientauthentifizierung (CLIENT) aktiviert ist, und sie wird benötigt, wenn eine einmalige Anmeldung (Single Sign-on) von einem Windows NT-Desktop aus in einer Windows NT-Domänenumgebung erforderlich ist.

#### **Zugehörige Konzepte:**

- „Einführung in DB2 für Windows NT und in die Windows NT-Sicherheit“ auf Seite 401

## **DB2 für Windows NT-Unterstützung für die Domänensicherheit**

Die folgenden Beispiele erläutern, wie DB2 für Windows NT die Domänensicherheit unterstützen kann. In diesem ersten Beispiel funktioniert die Verbindung problemlos, weil sich der Benutzername und die lokale Gruppe in derselben Domäne befinden. Im zweiten Beispiel schlägt die Verbindung fehl, weil sich der Benutzername und die lokale oder globale Gruppe in unterschiedlichen Domänen befinden.

**Beispiel für eine erfolgreiche Verbindung:** Die Verbindung im folgenden Szenario funktioniert, weil sich der Benutzername und die lokale oder globale Gruppe 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.

Tabella 52. Erfolgreiche Verbindung mit einem Domänencontroller

Domäne1	Domäne2
Zu Domäne2 besteht eine Vertrauensstellung.	<ul style="list-style-type: none"> <li>• Zu Domäne1 besteht eine Vertrauensstellung.</li> <li>• Die lokale oder globale Gruppe grp2 wurde definiert.</li> <li>• Der Benutzername id2 wurde definiert.</li> <li>• Der Benutzername id2 gehört zu grp2.</li> </ul>
<p>Der DB2-Server wird in dieser Domäne ausgeführt. Die folgenden DB2-Befehle werden über dieses System 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 Domäne wird durchsucht, id2 kann jedoch nicht gefunden werden. Die Einrichtung 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 sich der Benutzername und die lokale oder globale Gruppe in derselben Domäne befinden.	

**Zugehörige Konzepte:**

- „Gruppen- und Benutzerauthentifizierung unter Windows NT“ auf Seite 406

**Zugehörige Tasks:**

- „DB2 für Windows NT-Authentifizierung mit Gruppen und Domänensicherheit“ auf Seite 409

---

## Anhang G. Verwenden des Windows-Systemmonitors

---

### Windows-Systemmonitor - Einführung

Wenn Sie mit DB2® Universal Database (DB2 UDB) für Windows® arbeiten, stehen Ihnen Tools zur Überwachung der Leistung zur Verfügung:

- **DB2 Performance Expert**

DB2 Performance Expert for Multiplatforms Version 1.1 dient zur Konsolidierung, Dokumentation, Analyse und Empfehlung von Änderungen zur Selbstverwaltung und zur Ressourcenoptimierung auf der Grundlage leistungsbezogener DB2 UDB-Informationen.

- **DB2 UDB-Diagnosezentrale**

Die Funktionen der Diagnosezentrale stellen Ihnen verschiedene Methoden zur Arbeit mit leistungsbezogenen Informationen zur Verfügung. Diese Funktionen ersetzen in gewissen Rahmen die Funktionen des Performance Monitor der Steuerzentrale.

- **Windows-Systemmonitor**

Mit dem Windows-Systemmonitor können Sie die Datenbank- und Systemleistung überwachen und dabei Informationen von allen Leistungsdatenquellen abrufen, die im System registriert sind. Windows bietet auch Leistungsinformationen zu allen Aspekten des Maschinenbetriebs, darunter:

- CPU-Auslastung
- Speicherauslastung
- Plattenaktivität
- Netzwerkaktivität

**Zugehörige Tasks:**

- „Registrieren von DB2 im Windows-Systemmonitor“ auf Seite 413
- „Aktivieren des Fernzugriffs auf DB2-Leistungsinformationen“ auf Seite 414
- „Anzeigen von Leistungswerten für DB2 UDB und DB2 Connect“ auf Seite 415
- „Zugreifen auf ferne DB2 UDB-Leistungsinformationen“ auf Seite 416
- „Zurücksetzen von DB2-Leistungswerten“ auf Seite 417

**Zugehörige Referenzen:**

- „Windows-Leistungsobjekte“ auf Seite 415

---

### Registrieren von DB2 im Windows-Systemmonitor

**Vorgehensweise:**

Das Konfigurationsprogramm registriert DB2 automatisch im Windows-Systemmonitor.

Um die Leistungsdaten von DB2 Universal Database™ (DB2 UDB) und DB2 Connect für den Windows-Systemmonitor verfügbar zu machen, müssen Sie die DLL für die Leistungszähler (Datenquellen) von DB2 für Windows registrieren. Dadurch können andere Windows-Anwendungen mit den Win32-Leistungs-APIs Leistungsdaten abrufen.

Geben Sie zum Installieren und Registrieren der Leistungszähler-DLL von DB2 für Windows (DB2Perf.DLL) im Windows-Systemmonitor folgenden Befehl ein:

```
db2perfi -i
```

Durch das Registrieren der DLL wird ein neuer Schlüssel in der Option **Services** der Registrierdatenbank 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.

**Zugehörige Referenzen:**

- „db2perfi - Performance Counters Registration Utility Command“ in *Command Reference*

---

## Aktivieren des Fernzugriffs auf DB2-Leistungsinformationen

**Vorgehensweise:**

Wenn Ihre Workstation unter DB2 für Windows mit anderen Maschinen unter Windows vernetzt ist, können Sie die in diesem Abschnitt beschriebene Funktion verwenden.

Damit Windows-Leistungsobjekte von einer anderen Maschine unter DB2 für Windows angezeigt werden können, müssen Sie einen Administratorbenutzernamen und ein Administratorkennwort in DB2 Universal Database™ (DB2 UDB) registrieren. (Der Standardbenutzername des Windows-Systemmonitors, **SYSTEM**, ist in DB2 UDB 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 UDB entsprechen.

Die Angaben benutzername und kennwort sind in einem Schlüssel in der Registrierdatenbank 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 Registrierdatenbank zu vermeiden.

**Anmerkungen:**

1. Sobald eine Kombination aus Benutzername und Kennwort in DB2 UDB registriert wurde, melden sich auch lokale Exemplare des Systemmonitors explizit mit diesem Benutzernamen und Kennwort an. Dies bedeutet, dass bei lokalen Sitzungen des Systemmonitors keine DB2 UDB-Leistungsdaten angezeigt werden, wenn die Anmeldung nicht mit den in DB2 UDB registrierten Angaben übereinstimmen.
2. Die Kombination aus Benutzername und Kennwort muss so gepflegt werden, dass die Werte für den Benutzernamen und das Kennwort den in der

Sicherheitsdatenbank von Windows gespeicherten Werten entsprechen. Wenn der Benutzername oder das Kennwort in der Sicherheitsdatenbank von Windows 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>

**Zugehörige Konzepte:**

- „Allgemeine Namenskonventionen“ auf Seite 327

**Zugehörige Referenzen:**

- „db2perfr - Performance Monitor Registration Tool Command“ in *Command Reference*

---

## Anzeigen von Leistungswerten für DB2 UDB und DB2 Connect

**Vorgehensweise:**

Wenn Sie Leistungswerte für DB2 Universal Database™ (DB2 UDB) und DB2 Connect mit dem Systemmonitor anzeigen wollen, wählen Sie einfach die Leistungszähler (Datenquellen), 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 Datenquellen 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 zudem ein Exemplar (Instanz) für jedes logische Laufwerk in der Maschine, einschließlich „C:“ und „D:“.

**Zugehörige Konzepte:**

- „Windows-Systemmonitor - Einführung“ auf Seite 413

**Zugehörige Referenzen:**

- „Windows-Leistungsobjekte“ auf Seite 415

---

## Windows-Leistungsobjekte

Windows stellt die folgenden Leistungsobjekte bereit:

- **DB2-Datenbankmanager (DB2 Database Manager)**

Dieses Objekt bietet allgemeine Informationen für ein Windows-Exemplar. Das überwachte Exemplar von DB2 Universal Database™ (DB2 UDB) wird als Objektexemplar angezeigt.

Aus praktischen und Leistungsgründen können Sie Leistungsdaten jeweils nur von einem DB2 UDB-Exemplar abrufen. Das DB2 UDB-Exemplar, das im Systemmonitor angezeigt wird, wird von der Registriervariablen db2instance im Prozess des Systemmonitors gesteuert. Wenn mehrere DB2 UDB-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 UDB-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 Port 0 zugeordnet ist) angezeigt. Um die Leistungsdaten eines anderen Knotens anzuzeigen, müssen Sie eine separate Sitzung des Systemmonitors starten und für die Umgebungsvariable DB2NODE die Knotennummer des zu überwachenden Knotens definieren.

- **DB2 UDB-Datenbanken (DB2 UDB 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 UDB-Anwendung. Informationen sind für jede derzeit aktive DB2 UDB-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-Systemmonitor aufgelistet werden, hängt davon ab, was auf Ihrer Windows-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 UDB-Datenbanken und -Anwendungen auf dieser Maschine aktiv sind, werden auch die Objekte für DB2-Datenbanken und DB2-Anwendungen aufgelistet. Wenn Sie Ihr Windows-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.

**Zugehörige Konzepte:**

- „Registrierdatenbank- und Umgebungsvariablen von DB2“ in *Systemverwaltung: Optimierung*

---

## Zugreifen auf ferne DB2 UDB-Leistungsinformationen

**Vorgehensweise:**

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 höher sein.

**Zugehörige Konzepte:**

- „Windows-Systemmonitor - Einführung“ auf Seite 413



---

## Zurücksetzen von DB2-Leistungswerten

### Vorgehensweise:

Wenn eine Anwendung die Überwachungs-APIs von DB2 aufruft, werden normalerweise kumulierte Werte seit dem Start des DB2 Universal Database™-Servers (DB2 UDB-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 UDB-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. Zum Beispiel:

```
db2perf
db2perf dbalias1 dbalias2 ... dbaliasn

db2perf -d
db2perf -d dbalias1 dbalias2 ... dbaliasn
```

Das erste Beispiel setzt die Leistungswerte für alle aktiven DB2 UDB-Datenbanken zurück. Das zweite Beispiel setzt die Leistungswerte für bestimmte DB2 UDB-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 UDB-Serverexemplar zugreifen (d. h. das in DB2INSTANCE definierte Exemplar in der Sitzung, in der Sie **db2perf** ausführen).

Durch den Aufruf von **db2perf** werden auch die Werte zurückgesetzt, die bei jedem Fernzugriff auf DB2 UDB-Leistungsdaten mit dem Befehl **db2perf** angezeigt werden.

**Anmerkung:** Es gibt eine DB2 UDB-API, `sqlmrset`, mit der eine Anwendung nur die lokal (nicht global) angezeigten Werte für bestimmte Datenbanken zurücksetzen kann.

### Zugehörige Referenzen:

- „db2ResetMonitor - Reset Monitor“ in *Administrative API Reference*
- „db2perf - Reset Database Performance Values Command“ in *Command Reference*



---

## Anhang H. Verwenden von Windows-Datenbankpartitionsservern

Beim Ändern der Konfiguration in einer Windows-Umgebung werden die dazu erforderlichen Tasks mit speziellen Dienstprogrammen ausgeführt.

Im vorliegenden Anhang werden die folgenden Dienstprogramme vorgestellt:

- „Auflisten der Datenbankpartitionsserver eines Exemplars“
- „Hinzufügen eines Datenbankpartitionsservers zu einem Exemplar (Windows)“ auf Seite 420
- „Ändern der Datenbankpartition (Windows)“ auf Seite 421
- „Löschen einer Datenbankpartition aus einem Exemplar (Windows)“ auf Seite 423

---

### Auflisten der Datenbankpartitionsserver eines Exemplars

#### Vorgehensweise:

Unter Windows 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. Sie können in dem Befehl auch ein bestimmtes Exemplar angeben:

```
db2nlist /i:exempName
```

Dabei ist `exempName` der Name des gewünschten Exemplars.

Mit dem folgenden Befehl können Sie außerdem (optional) den Status jedes Partitionsservers abfragen:

```
db2nlist /s
```

Jeder Datenbankpartitionsserver kann einen der folgenden Statuswerte aufweisen: `starting`, `running`, `stopping` oder `stopped`.

#### Zugehörige Tasks:

- „Hinzufügen eines Datenbankpartitionsservers zu einem Exemplar (Windows)“ auf Seite 420
- „Ändern der Datenbankpartition (Windows)“ auf Seite 421
- „Löschen einer Datenbankpartition aus einem Exemplar (Windows)“ auf Seite 423

---

## Hinzufügen eines Datenbankpartitionsservers zu einem Exemplar (Windows)

### Vorgehensweise:

Unter Windows können Sie mit dem Befehl **db2ncrt** einem Exemplar einen Datenbankpartitionsserver (Knoten) 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_port
```

- /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\_port  
Die logische Portnummer, die für den Datenbankpartitionsserver verwendet wird, wenn der logische Port ungleich null (0) ist. Wird hier kein Wert angegeben, wird die logische Portnummer '0' zugeordnet.

Die logische Portnummer ist nur dann optional, wenn der erste Knoten auf einer Maschine erstellt wird. Beim Erstellen eines logischen Knotens müssen Sie diesen Parameter angeben und eine logische Portnummer auswählen, die noch nicht benutzt wird. Es gelten verschiedene Einschränkungen:

- Auf jeder Maschine muss ein Datenbankpartitionsserver mit dem logischen Port '0' definiert sein.
- Die Portnummer darf den für die FCM-Kommunikation reservierten Portbereich in der Servicedatei im Verzeichnis `%SystemRoot%\system32\drivers\etc` nicht überschreiten. Wenn Sie zum Beispiel einen Bereich von vier Ports für das aktuelle Exemplar reservieren, ist die höchste zulässige Portnummer 3 (Ports 1, 2 und 3; Port 0 ist dem logischen Standardknoten zugeordnet). Der Portbereich wird definiert, wenn der Befehl **db2icrt** mit dem Parameter `/r:anfangsport, endport` angegeben wird.

Außerdem gibt es folgende optionale 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:hostname

Gibt an, welcher TCP/IP-Hostname von FCM für die interne Übertragung verwendet wird, wenn der Hostname nicht der lokale Hostname 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-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. dem Exemplar TESTMPP einen neuen Datenbankpartitionsserver auf der Exemplareignermaschine MYMACHIN hinzufügen wollen (um mehrere logische Knoten auszuführen) und diesen neuen Knoten als Knoten 2 mit dem logischen Port 1 definieren wollen, müssen Sie Folgendes eingeben:

```
db2ncrt /n:2 /p:1 /u:meine_id,mein_kennwort /i:TESTMPP  
/M:TEST /o:MYMACHIN
```

#### Zugehörige Referenzen:

- „db2start - Start DB2 Command“ in *Command Reference*
- „db2icrt - Create Instance Command“ in *Command Reference*
- „db2ncrt - Add Database Partition Server to an Instance Command“ in *Command Reference*

---

## Ändern der Datenbankpartition (Windows)

### Vorgehensweise:

Unter Windows können Sie mit dem Befehl **db2nchg** folgende Operationen ausführen:

- Versetzen der Datenbankpartition von einer Maschine auf eine andere.
- Ändern des TCP/IP-Hostnamens 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 Portnummer.
- 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 optionalen 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 Service von DB2 Universal Database™ (DB2 UDB). Wird dieser Parameter nicht angegeben, bleiben das Anmeldekonto und das Kennwort unverändert.

- /p:logischer\_port

Ändert den logischen Port 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 Portnummer unverändert.

- /h:hostname

Ändert den TCP/IP-Hostnamen, der von FCM für die interne Kommunikation verwendet wird. Wird dieser Parameter nicht angegeben, bleibt der Hostname 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 zum Beispiel der logische Port für den Knoten 2 geändert werden soll, der an dem Exemplar TESTMPP beteiligt ist, müssen Sie folgenden Befehl eingeben, um den logischen Port 3 zu verwenden:

```
db2nchg /n:2 /i:TESTMPP /p:3
```

DB2 UDB ermöglicht Ihnen den Zugriff auf DB2 UDB-Registriervariablen auf Exemplarebene auf einer fernen Maschine. Momentan werden DB2 UDB-Registriervariablen auf drei verschiedenen Ebenen gespeichert: Maschinenebene (globale Ebene), Exemplarebene und Knotenebene. Die auf Exemplarebene (einschließlich der auf Knotenebene) gespeicherten Registriervariablen können mit DB2REMOtepREG umgeleitet werden.

Wenn DB2REMOTEPREG definiert ist, greift DB2 UDB auf die DB2 UDB-Registriervariablen auf der Maschine zu, auf die DB2REMOTEPREG zeigt. Der Befehl `db2set` sähe folgendermaßen aus:

```
db2set DB2REMOTEPREG=<ferne_workstation>
```

Dabei ist `<ferne_workstation>` der Name der fernen Workstation.

**Anmerkung:** Bei der Definition dieser Option ist Vorsicht geboten, da alle DB2 UDB-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.

**Zugehörige Konzepte:**

- „Registrierdatenbank- und Umgebungsvariablen von DB2“ in *Systemverwaltung: Optimierung*

**Zugehörige Referenzen:**

- „`db2nchg` - Change Database Partition Server Configuration Command“ in *Command Reference*

---

## Löschen einer Datenbankpartition aus einem Exemplar (Windows)

**Vorgehensweise:**

Unter Windows 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 0) des Exemplars entfernen, wird das Exemplar dadurch unbrauchbar. Verwenden Sie zum Löschen des Exemplars den Befehl **db2idrop**.

**Anmerkung:** Wenn das Exemplar 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 Port 0 zugeordnet ist, müssen Sie alle anderen Knoten löschen, die den anderen logischen Ports zugeordnet sind, bevor Sie den Knoten mit dem logischen Port 0 löschen können. Jeder Datenbankpartitionsserver muss über einen Knoten verfügen, dem der logische Port 0 zugeordnet ist.

Der Befehl verfügt über folgende Parameter:

```
db2ndrop /n:knotennummer /i:exemplarname
```

- /n:  
Die eindeutige Knotennummer zum Identifizieren des Datenbankpartitions-servers. Dieser Parameter ist erforderlich. Die Nummer kann aus den Werten 0 bis 999 (in aufsteigender Reihenfolge) bestehen. Beachten Sie, dass der Knoten null (0) die Exempleignermaschine ist.
- /i:exemplarname  
Der Exemplarname. Dies ist ein optionaler Parameter. Wird dieser Parameter nicht angegeben, wird standardmäßig das (mit der Registriervariablen DB2INSTANCE definierte) aktuelle Exemplar verwendet.

**Zugehörige Konzepte:**

- „Registrierdatenbank- und Umgebungsvariablen von DB2“ in *Systemverwaltung: Optimierung*

**Zugehörige Referenzen:**

- „db2stop - Stop DB2 Command“ in *Command Reference*
- „db2idrop - Remove Instance Command“ in *Command Reference*
- „db2ndrop - Drop Database Partition Server from an Instance Command“ in *Command Reference*



---

## Anhang I. Konfigurieren mehrerer logischer Knoten

---

### Möglichkeiten zur Verwendung mehrerer logischer Knoten

Normalerweise wird DB2<sup>®</sup> Universal Database (DB2 UDB) Enterprise Server 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 mit SMP-Architektur (SMP = Symmetric Multiprocessor) ausgeführt werden. 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. Dies stellt die kontinuierliche Verfügbarkeit der Benutzerdaten sicher.

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.

#### Zugehörige Tasks:

- „Konfigurieren mehrerer logischer Knoten“ auf Seite 425

#### Zugehörige Referenzen:

- „db2start - Start DB2 Command“ in *Command Reference*

---

## Konfigurieren mehrerer logischer Knoten

### Vorgehensweise:

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` keinesfalls 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 Hostnamen und die Portnummer, 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 Portnummer zuzuordnen. Dabei gilt die folgende Syntax:

```
knotennummer hostname logischer-port 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` keinesfalls 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 hostname computername logischer_port netzname
```

Verwenden Sie den vollständig qualifizierten Namen für den Hostnamen. In der Datei `/etc/hosts` sollte ebenfalls der vollständig qualifizierte Name verwendet werden. Wenn der vollständig qualifizierte Name in der Datei `'db2nodes.cfg'` nicht verwendet wird, in der Datei `/etc/hosts` jedoch verwendet wird, empfangen Sie eventuell die Fehlermeldung `SQL30082N RC=3`.

Sie müssen sicherstellen, dass in der Datei `services` des Verzeichnisses `etc` für die FCM-Kommunikation genügend Ports definiert sind.

#### Zugehörige Konzepte:

- „Möglichkeiten zur Verwendung mehrerer logischer Knoten“ auf Seite 425

#### Zugehörige Tasks:

- „Ändern der Knoten- und Datenbankkonfigurationsdatei“ auf Seite 168
- „Erstellen einer Knotenkonfigurationsdatei“ auf Seite 39

#### Zugehörige Referenzen:

- „db2start - Start DB2 Command“ in *Command Reference*
- „db2ncrt - Add Database Partition Server to an Instance Command“ in *Command Reference*

---

## Anhang J. Erweitern der Steuerzentrale

---

### Einführung der Plug-in-Architektur für die Steuerzentrale

Sie können die Steuerzentrale von DB2 Universal Database durch die neue *Plug-In*-Architektur erweitern, um zusätzliche Funktionen bereitzustellen.

Das Konzept der *Plug-In*-Architektur besteht darin, die Möglichkeit zu schaffen, dem Kontextmenü der Steuerzentrale neue Menüpunkte für ein bestimmtes Objekt hinzuzufügen, der Baumstruktur der Steuerzentrale Objekte hinzuzufügen sowie der Funktionsleiste neue Knöpfe hinzuzufügen. Eine Gruppe von Java<sup>™</sup>-Schnittstellen, die Sie implementieren müssen, wird zusammen mit den Tools geliefert. Mit diesen Schnittstellen wird der Steuerzentrale mitgeteilt, welche zusätzlichen Funktionen aufzunehmen sind.

Die Plug-In-Erweiterungen (db2plug.zip) werden beim Starten der Tools der Steuerzentrale geladen. Dadurch kann abhängig von der Größe der ZIP-Datei der Start der Tools länger dauern. Für die meisten Benutzer ist die ZIP-Datei für die Plug-ins jedoch klein und hat deswegen nur minimale Auswirkungen.

#### Zugehörige Konzepte:

- „Kompilieren und Ausführen der Beispiel-Plug-ins“ auf Seite 428
- „Schreiben von Plug-ins als Erweiterungen der Steuerzentrale“ auf Seite 429
- „Richtlinien für Entwickler von Plug-ins für die Steuerzentrale“ auf Seite 427

---

### Richtlinien für Entwickler von Plug-ins für die Steuerzentrale

Da in der Datei db2plug.zip mehrere Plug-ins enthalten sein können, sollten Plug-in-Entwickler bei der Erstellung eines Plug-ins für die Steuerzentrale die folgenden Richtlinien befolgen:

- Verwenden Sie Java<sup>™</sup>-Pakete, um sicherzustellen, dass ihre Plug-in-Klassen eindeutige Namen haben. Befolgen Sie dabei die Benennungskonventionen für Java-Pakete. Stellen Sie Ihren Paketnamen als Präfix den umgekehrten Namen Ihrer Internetdomäne (z. B. com.firmenname) voran. Alle Paketnamen oder zumindest die eindeutigen Präfixe sollten in Kleinbuchstaben geschrieben sein.
- Die Datei db2plug.zip sollte im Verzeichnis tools unter dem Verzeichnis sql1ib installiert werden. Vor Version 8 musste die Datei db2plug.zip im Verzeichnis cc unter dem Verzeichnis sql1ib installiert werden.
- Wenn Sie ein Plug-in für die Steuerzentrale erstellen und bereits eine Datei db2plug.zip vorhanden ist, sollten Sie Ihre Plug-in-Klassen der vorhandenen Datei db2plug.zip hinzufügen. Sie sollten die vorhandene Datei db2plug.zip nicht durch eine eigene Datei db2plug.zip überschreiben. Verwenden Sie den folgenden ZIP-Befehl, um Ihr Plug-in einer vorhandenen Datei db2plug.zip hinzuzufügen:

```
zip -r0 db2plug.zip com\firmenname\meinplugin\*.class
```

Datei ist der Paketname Ihres Plug-ins com.firmenname.meinplugin.

- Alle Klassen in der Datei db2plug.zip werden beim Starten der Steuerzentrale geladen. Die Datei db2plug.zip sollte alle Ihre Dateien für CCExtension-Klassen und Klassen enthalten, die Klassen im Paket com.ibm.db2.tools.cc.navigator

erweitern (extend) oder implementieren. Andere Klassen, die nicht direkt von diesen Klassen verwendet werden, brauchen nicht in der Datei db2plug.zip enthalten zu sein. Sie können in einer getrennten JAR-Datei gespeichert werden, um die Leistungseinbußen beim Starten der Steuerzentrale zu minimieren. Dies ist vorteilhaft, wenn eine große Anzahl zusätzlicher Klassen vorhanden ist. Sie sollten die JAR-Datei im Verzeichnis tools unter dem Verzeichnis sql1ib speichern. Die JAR-Datei wird automatisch in den Klassenpfad (*classpath*) aufgenommen, wenn Sie den Befehl **db2cc** zum Starten der Steuerzentrale verwenden.

- Plug-in-Klassen, die CObject implementieren, sollten einen argumentlosen Standardkonstruktor bereitstellen, der Aufrufe der Methode Class.newInstance() durch die Steuerzentrale ermöglicht.
- Vermeiden Sie nach Möglichkeit die Verwendung innerer Klassen. Im Allgemeinen sollten Plug-in-Klassen, die CCTreeObject zur Erstellung neuer Plug-in-Objekte in der Steuerzentrale implementieren, nicht als innere Klassen deklariert werden. Dadurch würde verhindert, dass die Steuerzentrale Instanzen dieser Klassen erstellen könnte.
- Testen Sie, ob Ihr Plug-in korrekt geladen wird, indem Sie den Befehl **db2cc -tf dateiname** verwenden. Dadurch werden Trace-Informationen zur Steuerzentrale in der angegebenen Datei aufgezeichnet. Wenn Sie keinen vollständigen Pfadnamen angeben, wird die Tracedatei in das Verzeichnis tools im Verzeichnis sql1ib geschrieben. Traceangaben, die sich auf Plug-ins beziehen, enthalten das Wort „Plugin“. Sie können prüfen, ob Ihre Klassen geladen werden, indem Sie nach Zeilen mit dem Text „PluginLoader“ suchen.

#### Zugehörige Konzepte:

- „Kompilieren und Ausführen der Beispiel-Plug-ins“ auf Seite 428
- „Schreiben von Plug-ins als Erweiterungen der Steuerzentrale“ auf Seite 429

#### Zugehörige Referenzen:

- „db2cc - Start Control Center Command“ in *Command Reference*

---

## Kompilieren und Ausführen der Beispiel-Plug-ins

Die Plug-in-Funktion der Steuerzentrale wird in den folgenden Abschnitten erläutert und an den entsprechenden Plug-in-Beispielprogrammen demonstriert: Example1.java, Example2.java, Example3.java, Example3Folder.java und Example3Child.java. Diese Java-Beispieldateien werden zusammen mit dem DB2® Application Development-Client installiert. Auf Windows®-Plattformen befinden sich diese Beispielprogramme im Verzeichnis `laufwerk:\sql1ib\samples\java\plugin`, wobei `laufwerk:` das Laufwerk darstellt, auf dem DB2 installiert ist. Auf UNIX®-Plattformen befinden sich diese Beispiele im Verzeichnis `/u/db2inst1/sql1ib/samples/java/plugin`, wobei `/u/db2inst1` das Verzeichnis darstellt, in dem DB2 installiert ist.

**Anmerkung:** Die Plug-in-Beispielprogramme können Aktualisierungen enthalten, die an dieser Stelle noch nicht berücksichtigt sind. Der Beispielcode und die Java-Dokumentation sollten als die aktuelleren Informationen betrachtet werden, wenn es Diskrepanzen zu den folgenden Ausführungen gibt.

Zur Ausführung der Beispiel-Plug-ins müssen Sie die Dateien mit der Erweiterung `class` gemäß den Regeln einer Java™-Archivdatei (JAR-Datei) in einer ZIP-Datei speichern. Die ZIP-Datei (db2plug.zip) muss sich in dem durch *classpath* definierten Klassenpfad befinden. Speichern Sie unter Windows-Betriebssystemen die Datei

db2plug.zip im Verzeichnis laufwerk:\sqllib\tools, wobei laufwerk: das Laufwerk darstellt, auf dem DB2 installiert ist. Auf UNIX-Plattformen speichern Sie die Datei db2plug.zip im Verzeichnis /u/db2inst1/sqllib/tools, wobei /u/db2inst1 das Verzeichnis darstellt, in dem DB2 installiert ist.

**Anmerkung:** Der Befehl **db2cc** definiert die Umgebungsvariable *classpath* so, dass sie auf die Datei db2plug.zip im Verzeichnis tools zeigt.

Die Beispiele (außer Example3, Example3Folder und Example3Child, die zusammengehören) sollten nicht in derselben Datei db2plug.zip zusammen gespeichert werden, da sie Konflikte untereinander verursachen können.

Zum Kompilieren jeder dieser Java-Beispieldateien müssen folgende JAR-Dateien ebenfalls in der Umgebungsvariablen *classpath* angegeben werden:

- Auf Windows-Plattformen:
  - laufwerk: \sqllib\java\Common.jar
  - laufwerk: \sqllib\tools\db2navplug.jar

Dabei stellt laufwerk das Laufwerk dar, auf dem DB2 installiert ist.

- Auf UNIX-Plattformen:
  - /u/db2inst1/sqllib/java/Common.jar
  - /u/db2inst1/sqllib/tools/db2navplug.jar

Dabei stellt /u/db2inst1 das Verzeichnis dar, in dem DB2 installiert ist.

Erstellen Sie die Datei db2plug.zip so, dass sie alle Klassen enthält, die durch Kompilieren der Java-Beispieldatei generiert wurden. Die Datei sollte ohne Komprimierung erstellt werden. Geben Sie zum Beispiel den folgenden Befehl ein:

```
zip -r0 db2plug.zip *.class
```

Dieser Befehl speichert alle class-Dateien in der Datei db2plug.zip und behält die relativen Pfadangaben bei.

#### **Zugehörige Konzepte:**

- „Schreiben von Plug-ins als Erweiterungen der Steuerzentrale“ auf Seite 429
- „Richtlinien für Entwickler von Plug-ins für die Steuerzentrale“ auf Seite 427

#### **Zugehörige Referenzen:**

- „db2cc - Start Control Center Command“ in *Command Reference*

---

## **Schreiben von Plug-ins als Erweiterungen der Steuerzentrale**

Der erste Schritt beim Schreiben eines Plug-ins besteht darin, eine Klasse zu definieren, die die Schnittstelle CCExtension implementiert. Diese Klasse enthält eine Liste von Plug-in-Klassen, die von der Steuerzentrale geladen werden müssen. Wenn Sie den Standardobjekten der Steuerzentrale, wie Datenbanken und Tabellen, Menüelemente hinzufügen oder eigene Objekte in der Baumstruktur anzeigen wollen, erstellen Sie Klassen, die die Schnittstelle CCOBJECT implementieren und geben ein Datenfeld (Array) dieser CCOBJECT-Objekte in der Methode getObjects() zurück. Wenn Sie einen Symbolleistenknopf hinzufügen wollen, implementieren Sie die Schnittstelle CCToolbarAction und geben ein Datenfeld von CCToolbarAction-Aktionen in der Methode getToolbarActions() zurück.

Jede dieser Schnittstellen ist in folgenden Verzeichnissen dokumentiert:

- Auf Windows®-Plattformen: LAUFWERK:\sql11ib\samples\java\plugin\doc, wobei LAUFWERK das Laufwerk darstellt, auf dem DB2® installiert ist.
- Auf UNIX®-Plattformen: /u/db2inst1/sql11ib/samples/java/plugin/doc, wobei /u/db2inst1 das Verzeichnis darstellt, in dem DB2 installiert ist.

#### Zugehörige Tasks:

- „Erstellen eines Plug-ins, das einen Symbolleistenknopf hinzufügt“ auf Seite 430
- „Erstellen einer Basismenüaktion“ auf Seite 432
- „Positionieren des Menüelements“ auf Seite 433
- „Erstellen eines Separators für Basismenüaktionen“ auf Seite 434
- „Erstellen von Untermenüs“ auf Seite 435
- „Hinzufügen eines Menüelements zu nur einem Objekt mit einem bestimmten Namen“ auf Seite 436
- „Hinzufügen des Ordners zur Aufnahme mehrerer Objekte in der Baumstruktur“ auf Seite 436
- „Hinzufügen eines Beispiellobjekts unter dem Ordner“ auf Seite 438
- „Einstellen von Attributen für ein Plug-in-Baumstrukturobjekt“ auf Seite 440
- „Hinzufügen der Erstellungsaktion“ auf Seite 442
- „Hinzufügen der Löschaktion mit Unterstützung für Mehrfachauswahl“ auf Seite 443
- „Hinzufügen der Änderungsaktion“ auf Seite 445
- „Inaktivieren von Konfigurationsfunktionen mit isConfigurable()“ auf Seite 446
- „Inaktivieren der Möglichkeit zum Ändern von Objekten mit isEditable()“ auf Seite 446
- „Inaktivieren der Standardwertknöpfe in Konfigurationsdialogen mit hasConfigurationDefaults()“ auf Seite 447

---

## Beschreibungen von Plug-in-Tasks

Die folgenden Plug-in-Tasks werden behandelt:

1. Erstellen eines Plug-ins, das einen Symbolleistenknopf hinzufügt
2. Erstellen eines Plug-ins, das dem Datenbankobjekt neue Menüelemente hinzufügt
3. Erstellen eines Plug-ins, das Plug-in-Objekte unter dem Datenbankobjekt in der Baumstruktur hinzufügt
4. Inaktivieren von Konfigurationsfunktionen mit isConfigurable()
5. Inaktivieren der Möglichkeit zum Ändern von Objekten mit isEditable()
6. Inaktivieren der Standardwertknöpfe in Konfigurationsdialogen mit hasConfigurationDefaults()

### Erstellen eines Plug-ins, das einen Symbolleistenknopf hinzufügt

#### Vorgehensweise:

In diesem Beispiel wird lediglich ein Symbolleistenknopf wie folgt hinzugefügt, so dass die Methode getObject() ein Nulldatenfeld (Array) zurückliefern sollte:

```
import com.ibm.db2.tools.cc.navigator.*;
import java.awt.event.*;
import javax.swing.*;
```

```

public class Example1 implements CCExtension {

    public CCOBJECT[] getObjects () {
        return null;
    }

}

```

Beachten Sie, dass das Paket `com.ibm.db2.tools.cc.navigator` importiert wird. Diese Klasse implementiert die Schnittstelle `CCToolbarAction`, die wiederum das Implementieren der drei Methoden `getHoverHelpText()`, `getIcon()` und `actionPerformed()` erfordert. Die Steuerzentrale verwendet die Methode `getHoverHelpText()` zum Anzeigen eines Textkästchens, das erscheint, wenn ein Benutzer den Mauszeiger über dem Symbolleistenknopf anhält. Das Symbol für den Knopf geben Sie mit der Methode `getIcon()` an. Die Steuerzentrale ruft die Methode `actionPerformed()` auf, wenn ein Benutzer auf den Knopf klickt. Das folgende Beispiel fügt einen Knopf mit dem Namen X hinzu, der eine Nachricht an die Konsole schreibt, wenn Sie darauf klicken. Als Symbol wird das Refresh-Symbol aus der `Bildrepository`-Klasse der Steuerzentrale verwendet.

```

class Example1ToolbarAction implements CCToolbarAction {

    public String getHoverHelpText() { return "X"; }

    public ImageIcon getIcon() {
        return CommonImageRepository.getCommonIcon(CommonImageRepository.WC_NV_
REFRESH);
    }

    public void actionPerformed(ActionEvent e) {
        System.out.println("Ich wurde angeklickt");
    }

}

```

Der letzte Schritt besteht darin, die Methode `getToolBarActions()` in `Example1` wie folgt zu implementieren, um eine Instanz der neuen Klasse zurückzugeben:

```

public CCToolbarAction[] getToolBarActions () {
    return new CCToolbarAction[] { new Example1ToolbarAction() };
}

```

#### Zugehörige Konzepte:

- „Kompilieren und Ausführen der Beispiel-Plug-ins“ auf Seite 428

## Erstellen eines Plug-ins, das dem Datenbankobjekt neue Menüelemente hinzufügt

Die folgende Prozedur gibt eine Übersicht über die Erstellung eines Plug-ins, das dem Datenbankobjekt neue Menüelemente hinzufügt:

1. Erstellen einer Basismenüaktion
2. Positionieren des Menüelements
3. Erstellen eines Separators für eine Basismenüaktion
4. Erstellen von Untermenüs
5. Hinzufügen eines Menüelements zu nur einem Objekt mit einem bestimmten Namen

## Erstellen einer Basismenüaktion

### Vorgehensweise:

Dieser Abschnitt behandelt das etwas fortgeschrittenere Thema des Einfügens neuer Befehle in das Kontextmenü des Datenbankobjekts (Database).

Ebenso wie in Beispiel 1 besteht der erste Schritt darin, eine Klasse zu schreiben, welche die Schnittstelle CCExtension implementiert.

```
import com.ibm.db2.tools.cc.navigator.*;
import java.awt.event.*;
import javax.swing.*;

public class Example2 implements CCExtension {

    public CCToolbarAction[] getToolbarActions () {
        return null;
    }

}
```

Im zweiten Schritt wird ein CCOBJECT-Objekt für das Datenbankobjekt in der Baumstruktur wie folgt erstellt:

```
class CCDatabase implements CCOBJECT {

    public String getName () { return null; }
    public boolean isEditable () { return true; }
    public boolean isConfigurable () { return true; }

    public int getType () { return UDB_DATABASE; }

}
```

Da in diesem Beispiel keine anderen Merkmale als die Möglichkeit zum Hinzufügen von Menüelementen in integrierten Objekten der Steuerzentrale (hier zum Beispiel das Datenbankobjekt) genutzt werden, werden die meisten Funktionen so implementiert, dass sie den Wert „null“ oder „true“ zurückgeben. Um festzulegen, dass dieses Objekt ein DB2 UDB-Datenbankobjekt darstellen soll, muss der Typ in Form der CCOBJECT-Konstanten UDB\_DATABASE angegeben werden. In diesem Beispiel erhält die Klasse den Namen „CCDatabase“, jedoch sollten Sie Ihre Klassennamen so eindeutig wie möglich wählen, da sich Plug-ins anderer Lieferanten in der gleichen ZIP-Datei wie Ihr Plug-in befinden können. Sie sollten Java-Pakete verwenden, um die Sicherstellung eindeutiger Klassennamen zu vereinfachen.

Die Methode getObjects() Ihres CCExtension-Objekts sollte wie folgt ein Datenfeld (Array) zurückliefern, das eine Instanz des CCDatabase-Objekts enthält:

```
public CCOBJECT[] getObjects () {
    return new CCOBJECT[] { new CCDatabase() };
}
```

Sie können mehrere CCOBJECT-Unterklassen mit dem Typ UDB\_DATABASE erstellen. Wenn jedoch die Werte, die von den Methoden isEditable() oder isConfigurable() zurückgegeben werden, widersprüchlich sind, überschreiben die Objekte, die den Wert „false“ zurückgeben, diejenigen, die den Wert „true“ zurückgeben.

Nun braucht nur noch die Methode getMenuActions() implementiert zu werden. Diese Methode liefert ein Datenfeld mit Menüaktionen (CCMenuActions), so dass zunächst eine Klasse geschrieben wird, die diese Schnittstelle implementiert.



Es sind zwei Methoden zu implementieren: getMenuText() und actionPerformed(). Der im Menü angezeigte Text wird mit der Methode getMenuText() abgerufen. Klickt ein Benutzer auf das Menüelement, bewirkt das ausgelöste Ereignis einen Aufruf der Methode actionPerformed().

Die folgende Beispielklasse zeigt ein Menüelement "Example2a Action" an, wenn ein einzelnes Datenbankobjekt ausgewählt wird. Wenn der Benutzer dieses Menüelement anklickt, wird die Nachricht "Example2a menu item actionPerformed" an die Konsole geschrieben.

```
class Example2AAction implements CCMenuAction {  
  
    public String getMenuText () { return "Example2a Action"; }  
  
    public void actionPerformed (ActionEvent e) {  
        System.out.println("Example2a menu item actionPerformed");  
    }  
  
}
```

Schließlich wird dieses Menüelement dem CCOBJECT-Objekt des Typs UDB\_DATABASE durch den folgenden Code hinzugefügt.

```
public CCMenuAction[] getMenuActions () {  
    return new CCMenuAction[] { new Example2AAction() };  
}
```

#### **Zugehörige Konzepte:**

- „Kompilieren und Ausführen der Beispiel-Plug-ins“ auf Seite 428

#### **Zugehörige Tasks:**

- „Positionieren des Menüelements“ auf Seite 433
- „Erstellen eines Separators für Basismenüaktionen“ auf Seite 434
- „Erstellen von Untermenüs“ auf Seite 435
- „Hinzufügen eines Menüelements zu nur einem Objekt mit einem bestimmten Namen“ auf Seite 436

## **Positionieren des Menüelements**

#### **Vorgehensweise:**

Bei der Erstellung des Basismenüs wurde die Position des Menüelements innerhalb des Menüs nicht angegeben. Die Standardfunktionsweise beim Hinzufügen von Plug-in-Menüelementen zu einem Menü fügt die Elemente am Ende des Menüs, jedoch vor möglichen Menüelementen mit Aktualisierungs- und Filterfunktionen (Refresh und Filter), hinzu.

Sie können diese Funktionsweise außer Kraft setzen, um jede beliebige Positionsnummer von 0 bis zur Anzahl von Elementen im Menü (die Aktualisierungs- und Filtermenüelemente nicht eingerechnet) anzugeben. Ändern Sie zu diesem Zweck die Unterklasse CCMenuAction so, dass sie Positionable implementiert und implementieren Sie anschließend die Methode getPosition() wie folgt:

```
class Example2BAction implements CCMenuAction, Positionable {  
  
    public String getMenuText () { return "Example2B Action"; }  
  
    public void actionPerformed (ActionEvent e) {  
        System.out.println("Example2B menu item actionPerformed");  
    }  
  
}
```

```

    public int getPosition() {
        return 0;
    }
}

```

Durch die Angabe der Positionsnummer 0 wird das Menüelement als erstes Element in die Liste eingefügt. Durch die Angabe der Positionsnummer, die der Anzahl von Elementen im Menü ohne Zählung des eigenen Plug-in-Menüelements entspricht, wird das Element am Ende des Menüs, jedoch vor möglicherweise vorhandenen Aktualisierungs- und Filtermenüelementen platziert. Sie können auch den Wert `Positionable.POSITION_BOTTOM` zurückgeben, um die Standardfunktionsweise zu erhalten, das heißt, um das Menüelement am Ende des Menüs vor möglichen Aktualisierungs- und Filtermenüelementen zu platzieren. Wenn mehrere `CCObject`-Objekte des Typs `UDB_DATABASE` mit Menüelementen vorhanden sind, die mit dem Wert `POSITION_BOTTOM` positioniert sind, werden die Menüelemente in der Reihenfolge angeordnet, in der die `CCObject`-Objekte des Typs `UDB_DATABASE` von der Methode `getObjects()` in der Schnittstelle `CCExtension` zurückgegeben werden.

Ändern Sie das Objekt `CCDatabase` wie folgt, um `Example2BAction` dem Menü hinzuzufügen:

```

    public CCMenuAction[] getMenuActions () {
        return new CCMenuAction[] { new Example2AAction(),
                                    new Example2BAction() };
    }
}

```

#### Zugehörige Tasks:

- „Erstellen einer Basismenüaktion“ auf Seite 432
- „Erstellen eines Separators für Basismenüaktionen“ auf Seite 434
- „Erstellen von Untermenüs“ auf Seite 435
- „Hinzufügen eines Menüelements zu nur einem Objekt mit einem bestimmten Namen“ auf Seite 436

## Erstellen eines Separators für Basismenüaktionen

#### Vorgehensweise:

Erstellen Sie eine Aktion „`CCMenuAction`“, die die Schnittstelle „`Separator`“ implementiert, um einen Separator hinzuzufügen. Alle anderen Methoden (außer `getPosition()`, wenn Sie „`Positionable`“ implementieren) werden ignoriert.

```

class Example2CSeparator implements CCMenuAction, Separator, Positionable {

    public String getMenuText () { return null; }

    public void actionPerformed (ActionEvent e) {}

    public int getPosition() {
        return 1;
    }
}

public CCMenuAction[] getMenuActions () {
    return new CCMenuAction[] { new Example2AAction(),
                                new Example2BAction(),
                                new Example2CSeparator() };
}

```

### Zugehörige Tasks:

- „Erstellen einer Basismenüaktion“ auf Seite 432
- „Positionieren des Menüelements“ auf Seite 433
- „Erstellen von Untermenüs“ auf Seite 435
- „Hinzufügen eines Menüelements zu nur einem Objekt mit einem bestimmten Namen“ auf Seite 436

## Erstellen von Untermenüs

### Vorgehensweise:

Ein Untermenü ist lediglich ein Feld von Menüaktionen (CCMenuAction). Damit ein Menüelement Untermenüs enthalten kann, muss es die Schnittstelle SubMenuParent implementieren. Dann erstellen Sie jeweils eine Implementierung von CCMenuAction für jedes Untermenüelement und geben diese in einem Datenfeld aus der Methode getSubMenuActions() der Schnittstelle SubMenuParent zurück. Das Hinzufügen von Menüelementen in Untermenüs, die selbst nicht als Plug-ins hinzugefügt wurden, wird nicht unterstützt. Beachten Sie außerdem, dass SubMenuParent-Objekte keine Aktionsereignisse (ActionEvent) aus der Steuerzentrale empfangen. Es folgt ein Beispiel:

```
class Example2DAction implements CCMenuAction, SubMenuParent {  
    public String getMenuText () { return "Example2D Action"; }  
    public void actionPerformed (ActionEvent e) {}  
    public CCMenuAction[] getSubMenuActions() {  
        return new CCMenuAction[] { new Example2DSubMenuAction() };  
    }  
}  
  
class Example2DSubMenuAction implements CCMenuAction {  
    public String getMenuText () { return "Example2D Sub-Menu Action"; }  
    public void actionPerformed (ActionEvent e) {  
        System.out.println("Example2D sub-menu menu item actionPerformed");  
    }  
}
```

Dieses neue Menüelement muss wiederum dem CCDatabase-Objekt hinzugefügt werden.

```
public CCMenuAction[] getMenuActions () {  
    return new CCMenuAction[] { new Example2AAction(),  
                                new Example2BAction(),  
                                new Example2CSeparator(),  
                                new Example2DAction() };  
}
```

### Zugehörige Tasks:

- „Erstellen einer Basismenüaktion“ auf Seite 432
- „Positionieren des Menüelements“ auf Seite 433
- „Erstellen eines Separators für Basismenüaktionen“ auf Seite 434
- „Hinzufügen eines Menüelements zu nur einem Objekt mit einem bestimmten Namen“ auf Seite 436

## Hinzufügen eines Menüelements zu nur einem Objekt mit einem bestimmten Namen

### Vorgehensweise:

Bisher würde jede Datenbank, die Sie in der Steuerzentrale anzeigen, die Plug-in-Menüelemente anzeigen, die Sie geschrieben haben. Sie können diese Menüelemente auf eine Datenbank mit einem bestimmten Namen beschränken, indem Sie diesen Namen in der Methode `getName()` des Objekts `CCDatabase` zurückgeben. Dabei muss es sich um einen vollständig qualifizierten Namen handeln. In diesem Fall müssen, da sich der Name auf eine Datenbank bezieht, die Namen des Systems, des Exemplars und der Datenbank in dem Namen enthalten sein, der in der Methode `getName()` zurückgegeben wird. Diese Namen werden durch " – " getrennt. Das folgende Beispiel zeigt die Methode `getName()` mit dem zurückgegebenen Namen für ein System namens `MYSYSTEM`, ein Exemplar namens `DB2` und eine Datenbank namens `SAMPLE`.

```
class CCDatabase implements CCOBJECT {
    ...
    public String getName () { return "MYSYSTEM – DB2 – SAMPLE"; }
    ...
}
```

### Zugehörige Tasks:

- „Erstellen einer Basismenüaktion“ auf Seite 432
- „Positionieren des Menüelements“ auf Seite 433
- „Erstellen eines Separators für Basismenüaktionen“ auf Seite 434
- „Erstellen von Untermenüs“ auf Seite 435

## Erstellen eines Plug-ins, das Plug-in-Objekte unter dem Datenbankobjekt in der Baumstruktur hinzufügt

Die folgende Prozedur skizziert die Erstellung eines Plug-ins, das Plug-in-Objekte unter einem Datenbankobjekt in der Baumstruktur hinzufügt:

1. Hinzufügen des Ordners zur Aufnahme mehrerer Objekte in der Baumstruktur
2. Hinzufügen eines Beispielobjekts unter dem Ordner
3. Einstellen von Attributen für ein Plug-in-Baumstrukturobjekt
4. Hinzufügen der Erstellungsaktion
5. Hinzufügen der Löschkaktion
6. Hinzufügen der Änderungsaktion

## Hinzufügen des Ordners zur Aufnahme mehrerer Objekte in der Baumstruktur

### Vorgehensweise:

In diesem Beispiel wird anstelle der Schnittstelle `CCObject` die Schnittstelle `CCTreeObject` implementiert, so dass Plug-in-Objekte in die Baumstruktur der Steuerzentrale unter dem Datenbankobjekt (`Database`) eingefügt werden können. Zunächst muss eine `CCTreeObject`-Implementierung für dieses Objekt erstellt werden. Es ist üblich, einen Ordner zu erstellen, wenn mehrere Objekte in die Baumstruktur eingefügt werden sollen, anstatt alle Objekte direkt unter dem Datenbankobjekt anzulegen. Das folgende Beispiel zeigt eine erste Version eines Ordners:

```
public class Example3Folder implements CCTreeObject {
    private String parentName = null;
    public boolean isEditable () { return false; }
```

```

public boolean isConfigurable () { return false; }
public CCTableObject getChildren () { return null; }
public void setParentName(String name)
{
    parentName = name;
}
public CCColumn[] getColumns () { return null; }
public boolean isLeaf () { return false; }
public CCMenuAction[] getMenuActions () { return null; }

public String getName () { return "Example3 Folder"; }

public void getData (Object[] data) {
    data[0] = this;
}

public int getType () { return CCTypeFactory.getTypeNumber
(this.getClass().getName()); }

public Icon getIcon (int iconState) {
    switch (iconState) {
        case CLOSED_FOLDER:
            return CommonImageRepository.getScaledIcon(CommonImageRepository.NV_CLOSED_
FOLDER);

            case OPEN_FOLDER:
            return CommonImageRepository.getScaledIcon(CommonImageRepository.NV_OPEN_
FOLDER);

            default:
            return CommonImageRepository.getScaledIcon(CommonImageRepository.NV_CLOSED_
FOLDER);
    }
}
}

```

Beachten Sie, dass die Methode `getType()` eine Klasse `CCTypeFactory` nutzt. Die Klasse `CCTypeFactory` hat den Zweck zu verhindern, dass zwei Objekte die gleiche Typennummer verwenden, so dass die Plug-ins durch die Steuerzentrale anhand eindeutiger Typen identifiziert werden können. Der neue Ordner besitzt keinen der integrierten `CC`-Objekttypen, sondern einen neuen Typ und muss dementsprechend eine neue Typennummer erhalten, die weder mit denen anderer neuer Typen, die Sie möglicherweise erstellen, noch mit denen der vorhandenen integrierten Typen übereinstimmen darf.

Die Methode `getIcon()` empfängt den Parameter `iconState`, über den mitgeteilt wird, ob der Ordner geöffnet oder geschlossen ist. Dadurch können Sie das Symbol dem Status des Ordners entsprechend anpassen.

Um den Ordner in der Detailansicht und nicht nur in der Baumstruktur zu zeigen, wenn die Datenbank ausgewählt ist, muss die Methode `getData()` eine einzige Spalte zurückliefern, deren Wert das Plug-in-Objekt selbst ist. Die Methode `getData()` ordnet dem ersten Element im Datenfeld den Verweis `this` zu. Dadurch können sowohl das Symbol als auch der Name in der gleichen Spalte der Detailansicht angezeigt werden. Durch das Zurückliefern der Unterklasse `CCTableObject` wird der Steuerzentrale mitgeteilt, dass sie die Methoden `getIcon()` und `getName()` für die Klasse `Example3Folder` aufrufen kann.

Der nächste Schritt ist die Erstellung einer Klasse CCDatabase, um CCTreeObject wie folgt zu implementieren und aus der zugehörigen Methode getChildren() ein CCTableObject-Datenfeld zurückzugeben, das eine Instanz der Klasse Example3Folder enthält:

```
import java.util.*;

class CCDatabase implements CCTreeObject {
    private String parentName = null;
    private Vector childVector;

    public CCDatabase() {
        childVector = new Vector();
        childVector.addElement(new Example3Folder());
    }

    public CCTableObject[] getChildren() {
        CCTableObject[] children = new CCTableObject[childVector.size()];
        childVector.copyInto(children);
        return children;
    }

    public void setParentName(String name)
    {
        parentName = name;
    }

    public String getName () { return null; }
    public boolean isEditable () { return false; }
    public boolean isConfigurable () { return false; }
    public void getData (Object[] data) { }
    public CCColumn[] getColumns () { return null; }
    public boolean isLeaf () { return false; }
    public int getType () { return UDB_DATABASE; }
    public Icon getIcon (int iconState) { return null; }
    public CCMenuAction[] getMenuActions () { return null; }
}
```

#### **Zugehörige Konzepte:**

- „Kompilieren und Ausführen der Beispiel-Plug-ins“ auf Seite 428

#### **Zugehörige Tasks:**

- „Hinzufügen eines Beispielobjekts unter dem Ordner“ auf Seite 438
- „Einstellen von Attributen für ein Plug-in-Baumstrukturobjekt“ auf Seite 440
- „Hinzufügen der Erstellungsaktion“ auf Seite 442
- „Hinzufügen der Löschaktion mit Unterstützung für Mehrfachauswahl“ auf Seite 443
- „Hinzufügen der Änderungsaktion“ auf Seite 445

### **Hinzufügen eines Beispielobjekts unter dem Ordner**

#### **Vorgehensweise:**

Der erste Schritt besteht darin, eine CCOBJECT-Implementierung für das untergeordnete Objekt (Child) wie folgt zu erstellen:

```
class Example3Child implements CCTableObject {
    private String parentName = null;
    public String getName () { return null; }
    public boolean isEditable () { return false; }
    public boolean isConfigurable () { return false; }
    public void getData (Object[] data) { }
    public CCColumn[] getColumns () { return null; }
}
```

```

public Icon getIcon (int iconState) { return null; }
public CCMenuAction[] getMenuActions () { return null; }
public void setParentName(String name)
{
    parentName = name;
}
public int getType () { return CTypeFactory.getTypeNumber
(this.getClass().getName()); }
}

```

Im nächsten Schritt muss die Klasse Example3Folder wie folgt modifiziert werden, so dass sie einen Vektor dieser Exercise3Child-Objekte verwaltet:

```

public class Example3Folder implements CCOBJECT {
    private String parentName = null;
    private Vector childVector;
    ...

    public Example3Folder() {
        childVector = new Vector();
    }

    ...
    public CCTableObject[] getChildren () {
        CCTableObject[] children = new CCTableObject[childVector.size()];
        childVector.copyInto(children);
        return children;
    }
    public void setParentName(String name)
    {
        parentName = name;
    }
    ...
}

```

Aus Gründen der Einfachheit gibt in diesem Beispiel der Befehl getChildren einen Bereich von untergeordneten Objekten zurück, die im "childVector" genannten Vektor gespeichert werden.

Ein echtes Plug-in kann die untergeordneten Objekte wiederherstellen, wenn "getChildren" aufgerufen wird. Dadurch wird die Liste aktualisiert, so dass diese neue oder geänderte untergeordnete Objekte enthalten kann, die außerhalb der Steuerzentrale erstellt oder geändert worden sein können seit diese Liste zuletzt angezeigt wurde. Untergeordnete Objekte können in persistentem Speicher gespeichert und aus diesem gelesen werden damit diese nicht verloren gehen.

Außerdem ist in einem echten Plug-in die Liste der untergeordneten Objekte, die von "getChildren" zurückgegeben werden davon abhängig, welche Objekte in der Baumstruktur der Steuerzentrale die übergeordneten Objekte dieses Objekts sind. Die übergeordneten Informationen befinden sich in der Zeichenfolge "parentName", die durch den Aufruf der Steuerzentrale an die Methode "setParentName" bereitgestellt wird.

**Anmerkung:** In diesem Beispiel geht, wenn in der Steuerzentrale im Datenbankobjekt oder an einer höheren Stelle in der Baumstruktur eine Aktualisierung ausgeführt wird, die Liste der untergeordneten Objekte unter dem Ordner Example3 verloren. Der Grund dafür ist, dass ein neuer Example3Folder von der Steuerzentrale erstellt wird, wenn die Aktualisierung ausgeführt wird. Würde dieser Beispielcode die untergeordneten Objekte aus persistentem Speicher einlesen, würden die untergeordneten Objekte nicht verloren gehen. Aus Gründen der Einfachheit wurde darauf verzichtet.

### Zugehörige Konzepte:

- „Kompilieren und Ausführen der Beispiel-Plug-ins“ auf Seite 428

### Zugehörige Tasks:

- „Hinzufügen des Ordners zur Aufnahme mehrerer Objekte in der Baumstruktur“ auf Seite 436
- „Einstellen von Attributen für ein Plug-in-Baumstrukturobjekt“ auf Seite 440
- „Hinzufügen der Erstellungsaktion“ auf Seite 442
- „Hinzufügen der Löschaktion mit Unterstützung für Mehrfachauswahl“ auf Seite 443
- „Hinzufügen der Änderungsaktion“ auf Seite 445

## Einstellen von Attributen für ein Plug-in-Baumstrukturobjekt

### Vorgehensweise:

Wenn Sie die Baumstruktur zu Ihrem Plug-in-Ordner erweitern und den Ordner auswählen, bemerken Sie, dass das Detailfenster keine Spalten enthält. Dies liegt daran, dass die `Example3Child`-Implementierung der Methode `getColumns()` den Wert `null` zurückgibt. Um dies zu ändern, erstellen Sie zunächst eine Reihe von `CCColumn`-Implementierungen. In diesem Beispiel werden zwei Spalten erstellt, weil ein späteres Beispiel zeigen wird, wie der Wert einer dieser Spalten zur Laufzeit geändert wird, und jedes Objekt zudem eine Spalte haben soll, die sich nie ändert. Die unveränderliche Spalte erhält den Namen „Name“ und die veränderliche Spalte den Namen „State“.

```
class NameColumn implements CCColumn {
    getName() { return "Name"; }
    getColumnClass { return CCTableObject.class; }
}

class StateColumn implements CCColumn {
    getName() { return "State"; }
    getColumnClass { return String.class; }
}
```

Zu den unterstützten Klassentypen gehören die Klassenäquivalente der Java-Basis-klassen (z. B. `java.lang.Integer`), die Klasse `java.util.Date` und die Klasse `CCTableObject`.

Ändern Sie die Methode `getColumns()` von `Example3Child`, um diese beiden Spalten einzufügen.

```
class Example3Child implements CCTableObject {
    ...
    public CCColumn[] getColumns () {
        return new CCColumn[] { new NameColumn(),
                                new StateColumn() };
    }
    ...
}
```

Sie müssen außerdem den übergeordneten Ordner ändern, um die gleichen Spalten hinzuzufügen.

```
class Example3Folder implements CCTableObject {
    ...
    public CCColumn[] getColumns () {
        return new CCColumn[] { new NameColumn(),
```



```

        new StateColumn() };
    }
    ...
}

```

Im nächsten Schritt müssen Sie die Werte festlegen, die für jede Zeile in der Detailansicht angezeigt werden. Dies geschieht, indem Sie die Elemente des Object-Datenfelds festlegen, die an die Methode `getData()` übergeben werden. Die Klasse der Daten jeder Spalte muss mit der Klasse übereinstimmen, die von der Methode `getColumnClass()` für die entsprechende Spalte zurückgegeben wird.

```

class Example3Child implements CCTableObject {
    ...
    private String name;
    private String state;

    public Exampe3Child(String name, String state) {
        this.name = name;
        this.state = state;
    }
    ...
    public void getData (Object[] data) {
        data[0] = this;
        data[1] = state;
    }
    ...
}

```

In diesem Fall erhält die erste Spalte, die den Klassentyp `CCTableObject` besitzt, den Wert „this“. Dadurch kann die Steuerzentrale sowohl den Text, der von der Methode `getName()` zurückgeliefert wird, als auch das Symbol wiedergeben, das von der Methode `getIcon()` zurückgeliefert wird. Der nächste Schritt besteht nun darin, diese Methoden zu implementieren. In diesem Beispiel wird dasselbe Aktualisierungssymbol (REFRESH) als Symbolleistenknopf wie in Beispiel 1 (Example1) verwendet.

```

class Example3Child implements CCTableObject {
    ...
    public String getName () {
        return name;
    }
    public Icon getIcon () {
        return CommonImageRepository.getScaledIcon(CommonImageRepository.WC_NV_
REFRESH);
    }
    ...
}

```

Um sich die Ergebnisse der bisher durchgeführten Schritte anzusehen, können Sie ein untergeordnetes Beispielobjekt erstellen, das in der nächsten Übung wieder entfernt wird. Fügen Sie eine Instanz der Klasse `Example3Child` dem Objekt `Example3Folder` hinzu, wenn der Vektor `childVector` konstruiert wird.

```

public class Example3Folder implements CCTreeObject {
    ...
    public Example3Folder() {
        childVector = new Vector();
        childVector.addElement(new Example3Child("Plugin1", "State1"));
    }
    ...
}

```

#### Zugehörige Konzepte:

- „Kompilieren und Ausführen der Beispiel-Plug-ins“ auf Seite 428

### Zugehörige Tasks:

- „Hinzufügen des Ordners zur Aufnahme mehrerer Objekte in der Baumstruktur“ auf Seite 436
- „Hinzufügen eines Beispielobjekts unter dem Ordner“ auf Seite 438
- „Hinzufügen der Erstellungsaktion“ auf Seite 442
- „Hinzufügen der Änderungsaktion“ auf Seite 445

## Hinzufügen der Erstellungsaktion

### Vorgehensweise:

Wenn Ihre Benutzer die Möglichkeit haben sollen, zur Laufzeit Objekte unter Ihrem Ordner zu erstellen, müssen Sie nur den Vektor (Vector) aktualisieren, Ihre Klasse als Observable festlegen und die Methode `notifyObservers()` aufrufen, wenn der Benutzer ein Ereignis auslöst. Die Steuerzentrale registriert sich automatisch selbst als „Überwacher“ (Observer) für alle `CCTableObject`-Objekte, die überwachbar (Observable) sind.

Zunächst muss dem Objekt `Example3Folder` eine Methode hinzugefügt werden, die ihrem Vektor für untergeordnete Objekte (Child-Objekte) ein untergeordnetes Objekt hinzufügt.

```
public class Example3Folder implements CCTreeObject, Observable {
    ...
    public void addChild(Example3Child child) {
        childVector.addElement(child);
        setChanged();
        notifyObservers(new CCOBJECTCollectionEvent(this,
            CCOBJECTCollectionEvent.OBJECT_ADDED,
            child));
    }
    ...
}
```

In dem oben gezeigten Code wird eine neue Klasse mit dem Namen `CCObjectCollectionEvent` als Argument für die Methode `notifyObservers()` verwendet. Ein `CCObjectCollectionEvent`-Ereignis ist ein Ereignis, das eine Änderung in einer Kollektion von `CCObject`-Objekten, wie zum Beispiel einem Ordner in der Baumstruktur der Steuerzentrale darstellt. Die Steuerzentrale überwacht alle `CCObject`-Objekte, die die Klasse `Observable` erweitern, und reagiert auf `CCObjectCollectionEvent`-Ereignisse, indem sie die Baumstruktur und die Detailansicht aktualisiert. Es gibt drei Typen von Ereignissen: Hinzufügen (ADD), Entfernen (REMOVE) und Ändern (ALTER).

Ein `CCObjectCollectionEvent`-Ereignis empfängt drei Argumente. Das erste Argument ist das Objekt, das das Ereignis ausgelöst hat. Das zweite Argument ist der Ereignistyp, der die Werte `OBJECT_ADDED`, `OBJECT_ALTERED` oder `OBJECT_REMOVED` haben kann. Das letzte Argument ist das neue Objekt, das erstellt wird.

Im nächsten Schritt fügen Sie dem Ordner ein Menüelement hinzu, das es dem Benutzer ermöglicht, einen Aufruf an Ihre neue Methode `addChild()` auszulösen.

```
class CreateAction implements CCMenuAction {
    private int pluginNumber = 0;
    public String getMenuText () { return "Create"; }

    public void actionPerformed (ActionEvent e) {
```

```

Example3Folder folder = (Example3Folder)((Vector)e.getSource()).elementAt(0);
    folder.addChild(new Example3Child("Plugin " + ++pluginNumber, "State1"));
}
}

```

Das ActionEvent-Argument enthält immer einen Vektor aller derjenigen Objekte, für die die Aktion aufgerufen wurde. Da diese Aktion nur für einen Ordner Example3Folder aufgerufen wird und es nur einen Ordner geben kann, wird das erste Objekt einfach in den Vektor (Vector) umgesetzt und die Methode addChild() für den Ordner aufgerufen.

Im letzten Schritt wird die Menüaktion dem Ordner hinzugefügt. Jetzt können Sie auch das Beispielobjekt entfernen, das Sie zuvor hinzugefügt haben.

```

public class Example3Folder extends Observable implements CCTreeObject {
    private CCMenuAction[] menuActions =
        new CCMenuAction[] { new CreateChildAction(); }
    ...
    public Example3Folder() {
        childVector = new Vector();
    }
    ...
    public CCMenuAction[] getMenuActions () {
        return menuActions;
    }
    ...
}

```

#### Zugehörige Konzepte:

- „Kompilieren und Ausführen der Beispiel-Plug-ins“ auf Seite 428

#### Zugehörige Tasks:

- „Hinzufügen des Ordners zur Aufnahme mehrerer Objekte in der Baumstruktur“ auf Seite 436
- „Hinzufügen eines Beispielobjekts unter dem Ordner“ auf Seite 438
- „Einstellen von Attributen für ein Plug-in-Baumstrukturobjekt“ auf Seite 440
- „Hinzufügen der Löschaktion mit Unterstützung für Mehrfachauswahl“ auf Seite 443
- „Hinzufügen der Änderungsaktion“ auf Seite 445

### Hinzufügen der Löschaktion mit Unterstützung für Mehrfachauswahl

#### Vorgehensweise:

Nun, da Ihre Benutzer die Möglichkeit haben, beliebig viele Instanzen Ihres Plugins zu erstellen, ist es auch sinnvoll, ihnen auch die Möglichkeit zu geben, diese wieder zu löschen. Dazu fügen Sie dem Objekt Example3Folder zunächst eine Methode zum Entfernen (remove) des untergeordneten Objekts (child) sowie zur Benachrichtigung (notify) der Steuerzentrale hinzu.

```

public class Example3Folder extends Observable implements CCTreeObject {

    public void removeChild(Example3Child child) {
        childVector.removeElement(child);
        setChanged();
        notifyObservers(new CCollectionEvent(this,

```

```

        CCollectionEvent.OBJECT_REMOVED,
        child));
    }
}

```

Im nächsten Schritt wird dem Example3Child-Objekt eine Menüaktion hinzugefügt. Außerdem wird diese Menüaktion so eingerichtet, dass sie die Schnittstelle MultiSelectable implementiert, so dass die Benutzer mehrere Objekte gleichzeitig entfernen können. Da die Quelle dieser Aktion ein Vektor von Example3Child-Objekten und nicht ein Example3Folder-Objekt ist, sollte das Example3Folder-Objekt auf andere Art an die Menüaktion übergeben werden, zum Beispiel im Konstruktor.

```

class RemoveAction implements CCMenuAction, MultiSelectable {
    private Example3Folder folder;

    public RemoveAction(Example3Folder folder) {
        this.folder = folder;
    }

    public String getMenuText () { return "Remove"; }

    public int getSelectionMode () { return MultiSelectable.MULTI_HANDLE_ONE; }

    public void actionPerformed (ActionEvent e){
        Vector childrenVector = (Vector)e.getSource();
        for (int i = 0; i < childrenVector.size(); i++) {
            folder.removeChild((Example3Child)childrenVector.elementAt(i));
        }
    }
}

```

Die Implementierung von MultiSelectable erfordert auch die Implementierung der Methode getSelectionMode(). In diesem Fall gibt sie den Wert MULTI\_HANDLE\_ONE zurück, der bewirkt, dass dieses Menüelement auch dann im Menü erscheint, wenn mehrere Objekte ausgewählt sind, und dass für alle ausgewählten Objekte nur ein Aufruf an die Methode actionPerformed() erfolgt.

Fügen Sie die neue Menüaktion nun dem Example3Child-Objekt hinzu. Dazu gehört das Hinzufügen eines neuen Parameters im Example3Child-Konstruktor, der im Ordner übergeben wird.

```

class Example3Child implements CCTableObject {
    ...
    private CCMenuAction[] menuActions;

    public Example3Child(Example3Folder folder, String name, String state) {
        ...
        menuActions = new CCMenuAction[] { new RemoveAction(folder) };
    }
    ...
    public CCMenuAction[] getMenuActions () {
        return menuActions;
    }
}

```

Vergessen Sie nicht, die Menüaktion CreateAction zu ändern, so dass sie den neuen Konstruktor verwendet.

```

class CreateAction implements CCMenuAction {
    ...
    public void actionPerformed (ActionEvent e) {
        ...
    }
}

```

```

        folder.addChild(new Example3Child(folder, "Plugin " + ++pluginNumber,
"State 1"));
    }
}

```

### Zugehörige Konzepte:

- „Kompilieren und Ausführen der Beispiel-Plug-ins“ auf Seite 428

### Zugehörige Tasks:

- „Hinzufügen des Ordners zur Aufnahme mehrerer Objekte in der Baumstruktur“ auf Seite 436
- „Hinzufügen eines Beispielobjekts unter dem Ordner“ auf Seite 438
- „Einstellen von Attributen für ein Plug-in-Baumstrukturobjekt“ auf Seite 440
- „Hinzufügen der Erstellungsaktion“ auf Seite 442
- „Hinzufügen der Änderungsaktion“ auf Seite 445

## Hinzufügen der Änderungsaktion

### Vorgehensweise:

Der letzte Typ von Ereignis, für den die Steuerzentrale in Bezug auf Plug-ins empfangsbereit ist, heißt OBJECT\_ALTERED. Im vorigen Beispiel wurde eine Spalte „State“ erstellt. Diese Einrichtung soll nun in diesem Beispiel demonstriert werden. Der Wert der Spalte „State“ wird erhöht, wenn die Änderungsaktion (AlterAction) aufgerufen wird.

Im ersten Schritt wird dazu eine Methode zum Ändern des Status (State) geschrieben, wobei sich die Aktion diesmal auf das Example3Child-Objekt und nicht auf den Ordner bezieht. In diesem Fall ist sowohl das erste als auch das dritte Argument das Example3Child-Objekt. Es darf nicht vergessen werden, die Klasse Observable zu erweitern.

```

class Example3Child extends Observable implements CCTableObject {
    ...
    public void setState(String state) {
        this.state = state;
        setChanged();
        notifyObservers(new CCObservableCollectionEvent(this,
            CCObservableCollectionEvent.OBJECT_ALTERED, this));
    }
    ...
}

```

Im folgenden Schritt erstellen Sie eine Menüaktion für das Ändern („Alter“) und fügen diese dem CCMenuAction-Datenfeld im Example3Child-Objekt hinzu. Die Klasse AlterAction implementiert außerdem die Schnittstelle CCDefaultMenuAction, um ‚Alter‘ als Standardaktion zu definieren, die aufgerufen wird, wenn der Benutzer ein Example3Child-Objekt in der Steuerzentrale doppelt anklickt.

```

class AlterAction implements CCMenuAction, CCDefaultMenuAction {
    private int stateNumber = 1;
    public String getMenuText () { return "Alter"; }

    public void actionPerformed (ActionEvent e) {
        ((Example3Child)((Vector)e.getSource()).elementAt(0)).setState("State "
            + ++stateNumber);
    }
}

class Example3Child implements CCTableObject {

```

```

...
public Example3Child(Example3Folder folder, String name, String state) {
    ...
    menuActions = new CCMenuAction[] { new AlterAction(),
                                        new RemoveAction(folder) };
    }
    ...
}

```

#### Zugehörige Konzepte:

- „Kompilieren und Ausführen der Beispiel-Plug-ins“ auf Seite 428

#### Zugehörige Tasks:

- „Hinzufügen des Ordners zur Aufnahme mehrerer Objekte in der Baumstruktur“ auf Seite 436
- „Hinzufügen eines Beispielobjekts unter dem Ordner“ auf Seite 438
- „Einstellen von Attributen für ein Plug-in-Baumstrukturobjekt“ auf Seite 440
- „Hinzufügen der Erstellungsaktion“ auf Seite 442
- „Hinzufügen der Löschaktion mit Unterstützung für Mehrfachauswahl“ auf Seite 443

## Inaktivieren von Konfigurationsfunktionen mit isConfigurable()

#### Vorgehensweise:

Wenn in der Methode isConfigurable() eines CCOBJECT-Objekts des Typs UDB\_DATABASE (Datenbank) oder UDB\_INSTANCE (Exemplar) der Wert „false“ zurückgegeben wird, wird das Menüelement „Konfigurieren“ (Configure) aus dem Kontextmenü für eine Datenbank bzw. ein Exemplar entfernt.

#### Zugehörige Konzepte:

- „Kompilieren und Ausführen der Beispiel-Plug-ins“ auf Seite 428

## Inaktivieren der Möglichkeit zum Ändern von Objekten mit isEditable()

#### Vorgehensweise:

Für alle Objekte der Steuerzentrale, die eine Änderungsaktion (Alter) unterstützen, kann diese Aktion entfernt werden, indem ein Plug-in für dieses Objekt erstellt wird, das in der Methode isEditable() den Wert „false“ zurückgibt. Sie können die Änderungsaktion auch nur für ein Objekt entfernen, dessen vollständig qualifizierter Name mit dem Wert übereinstimmt, der von der Methode getName() für dieses Objekt zurückgegeben wird.

Eine Zusatzeinrichtung ist die Möglichkeit, eine Aktion zum Durchsuchen (Browse) hinzuzufügen. Diese Möglichkeit besteht jedoch nur für Tabellen, Sichten und Indizes von DB2 UDB. Die Browse-Aktion wird hinzugefügt, indem eine Datei mit dem Namen BrowseTable, BrowseView oder BrowseIndex der Datei db2plug.zip hinzugefügt wird. Die Dateien können leer sein, jedoch müssen ihre Namen unbedingt BrowseTable, BrowseView oder BrowseIndex lauten. Das Hinzufügen eines Knopfs für die Browse-Aktion (Durchsuchen) lässt sich nicht auf einen bestimmten Namen begrenzen.

**Zugehörige Konzepte:**

- „Kompilieren und Ausführen der Beispiel-Plug-ins“ auf Seite 428

## **Inaktivieren der Standardwertknöpfe in Konfigurationsdialogen mit `hasConfigurationDefaults()`**

**Vorgehensweise:**

Die Konfigurationsdialoge für UDB-Datenbanken und -Exemplare enthalten Knöpfe, über die die Werte auf die DB2-Standardwerte zurückgesetzt werden können. Wenn Sie einen eigenen Satz von Standardwerten haben und nicht wünschen, dass Benutzer die Knöpfe für die Standardwerte versehentlich auswählen, können Sie die Knöpfe mit Hilfe eines Plug-ins inaktivieren. Erstellen Sie ein Objekt, das `CCObject` implementiert, und setzen Sie den Typ auf den Wert `UDB_DATABASE` oder `UDB_INSTANCE`. Das Objekt muss außerdem die Schnittstelle `Defaultable` implementieren. Diese Schnittstelle enthält eine Methode mit dem Namen `hasConfigurationDefaults()`. Definieren Sie diese Methode so, dass sie den Wert „false“ zurückgibt. Dadurch werden alle Standardwertknöpfe für den Konfigurationsdialog inaktiviert. Das Beispiel inaktiviert die Standardwertknöpfe für die UDB-Datenbankkonfiguration.

**Zugehörige Konzepte:**

- „Kompilieren und Ausführen der Beispiel-Plug-ins“ auf Seite 428





---

## Anhang K. Technische Informationen zu DB2 Universal Database

---

### DB2-Dokumentation und Hilfe

Die technischen Informationen zu DB2<sup>®</sup> stehen über die folgenden Tools und Methoden zur Verfügung:

- DB2 Information - Unterstützung
  - Themen
  - Hilfe für DB2-Tools
  - Beispielprogramme
  - Lernprogramme
- Für den Download verfügbare PDF-Dateien, PDF-Dateien auf CD und gedruckte Bücher
  - Handbücher
  - Referenzhandbücher
- Befehlszeilenhilfe
  - Hilfe für Befehle
  - Hilfe für Nachrichten
  - Hilfe für SQL-Anweisungen
- Installierter Quellcode
  - Beispielprogramme

Darüber hinaus können Sie auf zusätzliche technische Informationen zu DB2 Universal Database<sup>™</sup>, wie beispielsweise technische Hinweise (Technotes), White Papers und Redbooks<sup>™</sup>, online über [ibm.com](http://ibm.com)<sup>®</sup> zugreifen. Rufen Sie die Website 'DB2 Information Management - Library' unter [www.ibm.com/software/data/pubs/](http://www.ibm.com/software/data/pubs/) auf.

### Aktualisierungen der DB2-Dokumentation

In bestimmten Fällen stellt IBM<sup>®</sup> in regelmäßigen Abständen Dokumentations-Fix-Paks und andere Dokumentationsaktualisierungen für 'DB2 Information - Unterstützung' zur Verfügung. Wenn Sie über <http://publib.boulder.ibm.com/infocenter/db2help/> auf 'DB2 Information - Unterstützung' zugreifen, erhalten Sie stets die neuesten Informationen. Falls Sie 'DB2 Information - Unterstützung' lokal installiert haben, müssen Sie alle Aktualisierungen manuell installieren, bevor Sie sie anzeigen können. Diese Dokumentationsaktualisierungen ermöglichen Ihnen, die Informationen, die Sie von der CD mit *DB2 Information - Unterstützung* installiert haben, auf den neuesten Stand zu bringen, sobald neue Informationen verfügbar sind.

'DB2 Information - Unterstützung' wird häufiger aktualisiert als die PDF- und Hardcopy-Bücher. Um stets die jeweils neuesten technischen Informationen zu DB2 zur Verfügung zu haben, sollten Sie die Dokumentationsaktualisierungen installieren, sobald sie verfügbar sind, oder 'DB2 Information - Unterstützung' über die Website [www.ibm.com](http://www.ibm.com) aufrufen.

#### Zugehörige Konzepte:

- „CLI sample programs“ in *CLI Guide and Reference, Volume 1*
- „Java sample programs“ in *Application Development Guide: Building and Running Applications*

- „DB2 Information - Unterstützung“ auf Seite 450

#### Zugehörige Tasks:

- „Aufrufen der Kontexthilfe über ein DB2-Tool“ auf Seite 469
- „Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung'“ auf Seite 461
- „Aufrufen der Hilfe für Nachrichten über den Befehlszeilenprozessor“ auf Seite 470
- „Aufrufen der Hilfe für Befehle über den Befehlszeilenprozessor“ auf Seite 471
- „Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor“ auf Seite 471

#### Zugehörige Referenzen:

- „DB2-Dokumentation in PDF-Format und gedrucktem Format“ auf Seite 462

---

## DB2 Information - Unterstützung

Die DB2<sup>®</sup>-Komponente 'DB2 Information - Unterstützung' bietet Ihnen die Möglichkeit, auf alle Informationen zuzugreifen, die Sie zur optimalen Nutzung der Produkte innerhalb der DB2-Produktfamilie, wie z. B. DB2 Universal Database<sup>™</sup>, DB2 Connect<sup>™</sup>, DB2 Information Integrator und DB2 Query Patroller<sup>™</sup>, benötigen. 'DB2 Information - Unterstützung' dokumentiert auch die wichtigsten DB2-Funktionen und -Komponenten, einschließlich der Funktionen für die Replikation, das Data Warehousing und die DB2 Extender.

Wenn Sie für die Anzeige von 'DB2 Information - Unterstützung' Mozilla ab Version 1.0 oder Microsoft<sup>®</sup> Internet Explorer ab Version 5.5 verwenden, stehen Ihnen die folgenden Funktionen zur Verfügung. Für bestimmte Funktionen muss die JavaScript<sup>™</sup>-Unterstützung aktiviert werden :

#### Flexible Installationsoptionen

Wählen Sie für die Anzeige der DB2-Dokumentation die Option, die Ihren Anforderungen am besten entspricht:

- Stellen Sie ohne großen Aufwand sicher, dass Ihre Dokumentation stets auf dem neuesten Stand ist, indem Sie auf die gesamte Dokumentation direkt über 'DB2 Information - Unterstützung' auf der IBM<sup>®</sup> Website unter <http://publib.boulder.ibm.com/infocenter/db2help/> zugreifen.
- Reduzieren Sie den Aktualisierungsaufwand auf ein Minimum und begrenzen Sie den Datenaustausch auf Ihr Intranet, indem Sie die DB2-Dokumentation auf einem einzigen Server innerhalb Ihres Intranets installieren.
- Erzielen Sie maximale Flexibilität und reduzieren Sie die Abhängigkeit von Netzwerkverbindungen, indem Sie die DB2-Dokumentation auf dem eigenen Computer installieren.

#### Suchen

Sie können alle Themen in 'DB2 Information - Unterstützung' durchsuchen, indem Sie einen Suchbegriff im Textfeld **Suchen** eingeben. Schließen Sie Begriffe in Anführungszeichen ein, wenn Sie nach exakten Übereinstimmungen suchen möchten. Mit Hilfe von Platzhalterzeichen (\*, ?) und Booleschen Operatoren (AND, NOT, OR) können Sie die Suche eingrenzen.

#### Aufgabenorientiertes Inhaltsverzeichnis

Die Themen in der DB2-Dokumentation können über ein zentrales Inhaltsverzeichnis lokalisiert werden. Das Inhaltsverzeichnis ist primär auf der

Basis übergeordneter Aufgabenbereiche aufgebaut, enthält jedoch auch Einträge für Produktübersichten, Ziele, Referenzinformationen sowie einen Index und ein Glossar.

- Produktübersichten beschreiben die Beziehung zwischen den in der DB2-Produktfamilie verfügbaren Produkten sowie die von den einzelnen Produkten bereitgestellten Funktionen und enthalten darüber hinaus die neuesten Release-Informationen für diese Produkte.
- Aufgabenkategorien, wie z. B. Installation, Verwaltung und Entwicklung, umfassen Themen, mit deren Hilfe Sie die einzelnen Aufgaben schnell ausführen und sich außerdem genauere Kenntnisse über die Hintergrundinformationen zu diesen Aufgaben verschaffen können.
- In den Referenzthemen finden Sie detaillierte Informationen zu einem Thema, einschließlich der Anweisungs- und Befehlssyntax, der Hilfetexte zu Nachrichten und der Konfigurationsparameter.

#### **Anzeigen des aktuellen Themas im Inhaltsverzeichnis**

Wenn Sie sehen möchten, welchem Bereich des Inhaltsverzeichnisses das aktuelle Thema zugeordnet ist, klicken Sie den Knopf **Aktualisieren / aktuelles Thema anzeigen** im Teilfenster des Inhaltsverzeichnisses oder den Knopf **Im Inhaltsverzeichnis anzeigen** im Inhaltsteilfenster an. Diese Funktion ist zum Beispiel dann von Nutzen, wenn Sie mehreren Links zu zugehörigen Themen in verschiedenen Dateien gefolgt sind oder ein Thema über das Ergebnis einer Suche aufgerufen haben.

**Index** Über den Index können Sie auf die gesamte Dokumentation zugreifen. Der Index ist alphabetisch nach Indexeinträgen sortiert.

#### **Glossar**

Im Glossar finden Sie Definitionen zu Termini, die in der DB2-Dokumentation verwendet werden. Das Glossar ist alphabetisch nach Glossareinträgen sortiert.

#### **Integrierte übersetzte Informationen**

Die Informationen in 'DB2 Information - Unterstützung' werden in der Sprache angezeigt, die Sie in den Benutzervorgaben des verwendeten Browsers festgelegt haben. Ist ein Thema nicht in der bevorzugten Sprache verfügbar, wird die englische Version des Themas angezeigt.

Technische Informationen zu iSeries™ finden Sie im Informationszentrum von IBM eServer™ iSeries unter [www.ibm.com/eserver/series/infocenter/](http://www.ibm.com/eserver/series/infocenter/).

#### **Zugehörige Konzepte:**

- „DB2 Information - Unterstützung: Installationsszenarios“ auf Seite 452

#### **Zugehörige Tasks:**

- „Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung'“ auf Seite 461
- „Anzeigen von Themen in der gewünschten Sprache in 'DB2 Information - Unterstützung'“ auf Seite 462
- „Aufrufen von 'DB2 Information - Unterstützung'“ auf Seite 459
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (UNIX)“ auf Seite 454
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (Windows)“ auf Seite 457

## DB2 Information - Unterstützung: Installationsszenarios

Je nach Arbeitsumgebung kann es unterschiedliche Anforderungen hinsichtlich des Zugriffs auf DB2<sup>®</sup>-Informationen geben. Sie können auf 'DB2 Information - Unterstützung' entweder auf der IBM<sup>®</sup> Website zugreifen oder auf einem Server im unternehmensinternen Netzwerk oder auf eine auf dem lokalen Computer installierte Version. In allen drei Fällen befindet sich die Dokumentation in 'DB2 Information - Unterstützung', einem strukturierten System themenbasierter Informationen, die über einen Browser angezeigt werden können. Standardmäßig greifen DB2-Produkte auf 'DB2 Information - Unterstützung' auf der IBM Website zu. Wenn Sie jedoch auf 'DB2 Information - Unterstützung' auf einem Intranet-Server oder auf dem eigenen Computer zugreifen möchten, müssen Sie 'DB2 Information - Unterstützung' mit Hilfe der entsprechenden CD installieren, die sich im Programmpaket des Produkts befindet. Anhand der nachfolgenden Übersicht über die verfügbaren Optionen für den Zugriff auf die DB2-Dokumentation und mit Hilfe der drei Installationsszenarios können Sie ermitteln, welche Methode für den Zugriff auf 'DB2 Information - Unterstützung' für Ihre Anforderungen und Arbeitsumgebung am besten geeignet ist und welche Aspekte Sie bei der Installation berücksichtigen müssen.

### Übersicht über die verfügbaren Optionen für den Zugriff auf die DB2-Dokumentation:

Die folgende Tabelle enthält Empfehlungen hinsichtlich der für Ihre Arbeitsumgebung geeigneten Optionen für den Zugriff auf die DB2-Produktdokumentation in 'DB2 Information - Unterstützung'.

Internetzugriff	Intranetzugriff	Empfehlung
Ja	Ja	Greifen Sie entweder über die IBM Website auf 'DB2 Information - Unterstützung' zu oder auf die auf einem Intranet-Server installierte Version von 'DB2 Information - Unterstützung'.
Ja	Nein	Greifen Sie über die IBM Website auf 'DB2 Information - Unterstützung' zu.
Nein	Ja	Greifen Sie auf die auf einem Intranet-Server installierte Version von 'DB2 Information - Unterstützung' zu.
Nein	Nein	Greifen Sie auf die auf einem lokalen Computer installierte Version von 'DB2 Information - Unterstützung' zu.

### Szenario: Zugriff auf 'DB2 Information - Unterstützung' auf Ihrem Computer:

Tsu-Chen besitzt eine Fabrik in einer Kleinstadt, in der es vor Ort keinen Anbieter für einen Internetzugang gibt. Für die Verwaltung des Lagerbestands, der Produktbestellungen, der Betriebsausgaben und seines Bankkontos hat Tsu-Chen DB2 Universal Database<sup>™</sup> gekauft. Da er zuvor noch nie ein DB2-Produkt verwendet hat, muss er anhand der DB2-Produktdokumentation lernen, wie die Verwaltung funktioniert.

Nachdem er DB2 Universal Database mit der Option für die Standardinstallation auf seinem Computer installiert hat, versucht Tsu-Chen, auf die DB2-Dokumentation zuzugreifen. Sein Browser zeigt jedoch eine Fehlermeldung mit der Information an, dass die Seite, die geöffnet werden sollte, nicht gefunden werden kann. Tsu-Chen überprüft das Installationshandbuch für sein DB2-Produkt und findet

heraus, dass er 'DB2 Information - Unterstützung' zunächst installieren muss, um auf seinem Computer auf die DB2-Dokumentation zugreifen zu können. Im Programmpaket findet er die *CD für DB2 Information - Unterstützung* und installiert sie.

Über das Programm zum Aufrufen von Anwendungen für sein Betriebssystem hat Tsu-Chen nun Zugriff auf 'DB2 Information - Unterstützung', um sich mit der Verwendung seines DB2-Produkts vertraut zu machen und so einen wertvollen Beitrag zum Erfolg seines Unternehmens leisten.

#### **Szenario: Zugriff auf 'DB2 Information - Unterstützung' über die IBM Website:**

Colin ist IT-Berater bei einer Schulungsfirma. Er ist auf Datenbanktechnologie und SQL spezialisiert und hält Seminare zu diesen Themen für Unternehmen aus ganz Nordamerika ab. Hierfür verwendet er DB2 Universal Database. Im Rahmen seiner Seminare verwendet Colin die DB2-Dokumentation als Unterrichtsmaterial. Für SQL-Kurse beispielsweise verwendet Colin die DB2-Dokumentation zu SQL, um die grundlegende und erweiterte Syntax für Datenbankabfragen zu unterrichten.

Die meisten Unternehmen, bei denen Colin unterrichtet, verfügen über einen Internetzugang. Aus diesem Grund entschied sich Colin, seinen tragbaren Computer für den Zugriff auf 'DB2 Information - Unterstützung' über die Website von IBM zu konfigurieren, als er die letzte Version von DB2 Universal Database installiert hat. Diese Konfiguration ermöglicht es Colin, während seiner Seminare online auf die neueste DB2-Dokumentation zuzugreifen.

Wenn er auf Reisen ist, hat Colin bisweilen allerdings keinen Internetzugang. Dieser Umstand war für ihn recht problematisch, insbesondere dann, wenn er Zugriff auf die DB2-Dokumentation benötigte, um sich auf seine Seminare vorzubereiten. Um Situationen wie diese zu vermeiden, installierte Colin eine Kopie von 'DB2 Information - Unterstützung' auf seinem tragbaren Computer.

Auf diese Weise hat Colin nun jederzeit eine Kopie der DB2-Dokumentation zur Verfügung und ist dadurch wesentlich flexibler. Mit dem Befehl **db2set** kann Colin ohne Schwierigkeiten die Registrierdatenbankvariablen auf seinem tragbaren Computer so konfigurieren, dass er den jeweiligen Umständen entsprechend entweder über die Website von IBM oder über seinen tragbaren Computer auf 'DB2 Information - Unterstützung' zugreifen kann.

#### **Szenario: Zugriff auf 'DB2 Information - Unterstützung' über einen Intranet-Server:**

Eva arbeitet als leitende Datenbankadministratorin für eine Lebensversicherung. In ihre Zuständigkeit fallen auch das Installieren und Konfigurieren der neuesten Version von DB2 Universal Database auf den UNIX<sup>®</sup>-basierten Datenbankservern des Unternehmens. Vor Kurzem hat das Unternehmen seine Mitarbeiter darüber informiert, dass sie aus Sicherheitsgründen während der Arbeitszeit keinen Internetzugang erhalten würden. Da ihr Unternehmen in einer Netzwerkumgebung arbeitet, beschließt Eva, eine Kopie von 'DB2 Information - Unterstützung' auf einem Intranet-Server zu installieren, damit alle Mitarbeiter, die das Data Warehouse des Unternehmens regelmäßig verwenden (Vertriebsbeauftragte, Vertriebsleiter und Geschäftsanalysten), Zugriff auf die DB2-Dokumentation haben.

Eva weist ihr Datenbankteam an, die neueste Version von DB2 Universal Database auf allen Computern der Mitarbeiter mit Hilfe einer Antwortdatei zu installieren, um sicherzustellen, dass die Konfiguration des Zugriffs auf 'DB2 Information - Unterstützung' auf allen Computern mit dem Hostnamen und der Portnummer des Intranet-Servers erfolgt.

Durch ein Missverständnis installiert jedoch Migual, ein Datenbankadministrator in Evas Team, eine Kopie von 'DB2 Information - Unterstützung' auf mehreren Mitarbeitercomputern, anstatt DB2 Universal Database für den Zugriff auf 'DB2 Information - Unterstützung' über den Intranet-Server zu konfigurieren. Um diesen Fehler zu korrigieren, weist Eva Migual an, mit dem Befehl **db2set** die Registrierdatenbankvariablen von 'DB2 Information - Unterstützung' (DB2\_DOCHOST für den Hostnamen und DB2\_DOCPORT für die Portnummer) auf allen entsprechenden Computern zu ändern. Anschließend haben nun alle erforderlichen Computer im Netzwerk Zugriff auf 'DB2 Information - Unterstützung', und die Mitarbeiter können mit Hilfe der DB2-Dokumentation Antworten auf ihre Fragen zu DB2 finden.

#### **Zugehörige Konzepte:**

- „DB2 Information - Unterstützung“ auf Seite 450

#### **Zugehörige Tasks:**

- „Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung'“ auf Seite 461
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (UNIX)“ auf Seite 454
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (Windows)“ auf Seite 457
- „Festlegen der Speicherposition für den Zugriff auf 'DB2 Information - Unterstützung': Gemeinsame GUI - Hilfe“

#### **Zugehörige Referenzen:**

- „db2set - DB2 Profile Registry Command“ in *Command Reference*

---

## **Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (UNIX)**

Es gibt drei Möglichkeiten, auf die DB2-Produktdokumentation zuzugreifen: auf der IBM Website, auf einem Intranet-Server oder auf eine auf dem lokalen Computer installierte Version. Standardmäßig greifen DB2-Produkte auf die DB2-Dokumentation auf der IBM Website zu. Wenn Sie jedoch auf die DB2-Dokumentation auf einem Intranet-Server oder auf dem eigenen Computer zugreifen möchten, müssen Sie die Dokumentation von der CD 'DB2 Information - Unterstützung' aus installieren. Mit dem DB2-Installationsassistenten können Sie Ihre Installationseinstellungen definieren und 'DB2 Information - Unterstützung' auf einem Computer installieren, der das Betriebssystem UNIX verwendet.

#### **Voraussetzungen:**

Dieser Abschnitt erläutert die Voraussetzungen für Hardware, Betriebssystem, Software und Kommunikation zum Installieren von 'DB2 Information - Unterstützung' auf UNIX-Computern.

- **Hardwarevoraussetzungen**

Sie benötigen einen der folgenden Prozessoren:

- PowerPC (AIX)
- HP 9000 (HP-UX)
- Intel 32-Bit (Linux)
- Solaris UltraSPARC-Computer (Solaris-Betriebsumgebung)

- **Betriebssystemvoraussetzungen**

Sie benötigen eines der folgenden Betriebssysteme:

- IBM AIX 5.1 (auf PowerPC)
- HP-UX 11i (auf HP 9000)
- Red Hat Linux 8.0 (auf Intel 32-Bit)
- SuSE Linux 8.1 (auf Intel 32-Bit)
- Sun Solaris Version 8 (auf UltraSPARC-Computern in der Solaris-Betriebsumgebung)

**Anmerkung:** 'DB2 Information - Unterstützung' kann unter einem Teil der UNIX-Betriebssysteme ausgeführt werden, unter denen DB2-Clients unterstützt werden. Daher wird empfohlen, entweder über die IBM Website auf 'DB2 Information - Unterstützung' zuzugreifen oder 'DB2 Information - Unterstützung' auf einem Intranet-Server zu installieren und dort auf die Dokumentation zuzugreifen.

- **Softwarevoraussetzungen**

- Unterstützte Browser:

- Mozilla Version 1.0 oder höher

- Beim DB2-Installationsassistenten handelt es sich um ein grafisches Installationsprogramm. Um den DB2-Installationsassistenten auf Ihrem Computer ausführen zu können, benötigen Sie eine Implementierung der X Window System-Software zur Wiedergabe einer grafischen Benutzerschnittstelle (GUI). Bevor Sie den DB2-Installationsassistenten ausführen können, müssen Sie die entsprechende Anzeigefunktion (DISPLAY) unbedingt ordnungsgemäß exportieren. Geben Sie hierzu beispielsweise den folgenden Befehl an der Eingabeaufforderung ein:

```
export DISPLAY=9.26.163.144:0.
```

- **Kommunikationsvoraussetzungen**

- TCP/IP

**Vorgehensweise:**

Um 'DB2 Information - Unterstützung' mit Hilfe des DB2-Installationsassistenten zu installieren, gehen Sie wie folgt vor:

1. Melden Sie sich am System an.
2. Legen Sie die Produkt-CD von 'DB2 Information - Unterstützung' in das CD-Laufwerk ein, und hängen Sie die CD an Ihr System an.
3. Wechseln Sie in das Verzeichnis, in dem die CD angehängt ist. Geben Sie hierzu den folgenden Befehl ein:

```
cd /cd
```

Hierbei steht /cd für den Mountpunkt der CD.

4. Geben Sie den Befehl **./db2setup** ein, um den DB2-Installationsassistenten zu starten.

5. Die IBM DB2-Klickstartleiste wird geöffnet. Um direkt mit der Installation von 'DB2 Information - Unterstützung' fortzufahren, klicken Sie **Produkt installieren** an. Die Onlinehilfe enthält Informationen, die Sie durch die verbleibenden Schritte der Installation führen. Um die Onlinehilfe aufzurufen, klicken Sie **Hilfe** an. Sie können jederzeit **Abbrechen** anklicken, um die Installation zu beenden.
6. Klicken Sie im Fenster **Wählen Sie das zu installierende Produkt aus** den Knopf **Weiter** an.
7. Klicken Sie **Weiter** im Fenster **Willkommen beim DB2-Installationsassistenten** an. Der DB2-Installationsassistent leitet Sie durch die erforderlichen Schritte zum Installieren des Programms.
8. Um mit der Installation fortfahren zu können, müssen Sie die Lizenzvereinbarung akzeptieren. Wählen Sie auf der Seite **Lizenzvereinbarung** die Option **Bedingungen in der Lizenzvereinbarung anerkennen** aus, und klicken Sie **Weiter** an.
9. Wählen Sie **DB2 Information - Unterstützung auf diesem Computer installieren** auf der Seite **Installationsaktion auswählen** aus. Wenn Sie 'DB2 Information - Unterstützung' zu einem späteren Zeitpunkt auf diesem Computer oder anderen Computern mit Hilfe einer Antwortdatei installieren möchten, wählen Sie **Ihre Einstellungen in einer Antwortdatei speichern** aus. Klicken Sie **Weiter** an.
10. Wählen Sie auf der Seite **Zu installierende Sprachen auswählen** die Sprachen aus, in denen 'DB2 Information - Unterstützung' installiert werden soll. Klicken Sie den Knopf **Weiter** an.
11. Konfigurieren Sie 'DB2 Information - Unterstützung' auf der Seite **Port von DB2 Information - Unterstützung angeben** für eingehende Kommunikation. Klicken Sie **Weiter** an, um mit der Installation fortzufahren.
12. Überprüfen Sie auf der Seite **Kopieren der Dateien starten** noch einmal die von Ihnen ausgewählten Installationseinstellungen. Wenn Sie die Einstellungen ändern möchten, klicken Sie **Zurück** an. Klicken Sie **Installieren** an, um die Dateien von 'DB2 Information - Unterstützung' auf Ihren Computer zu kopieren.

Sie können 'DB2 Information - Unterstützung' auch mit Hilfe einer Antwortdatei installieren.

Die Installationsprotokolldateien db2setup.his, db2setup.log und db2setup.err befinden sich standardmäßig im Verzeichnis /tmp.

Die Datei db2setup.log erfasst alle Installationsinformationen zu DB2-Produkten, einschließlich Fehlern. Die Datei db2setup.his zeichnet alle DB2-Produktinstallationen auf Ihrem Computer auf. DB2 hängt die Datei db2setup.log an die Datei db2setup.his an. Die Datei db2setup.err erfasst die gesamte Fehlerausgabe, die von Java zurückgegeben wird, wie beispielsweise Informationen zu Ausnahmereingungen und Traps.

Nach Abschluss der Installation ist 'DB2 Information - Unterstützung' je nach UNIX-Betriebssystem in einem der folgenden Verzeichnisse installiert:

- AIX: /usr/opt/db2\_08\_01
- HP-UX: /opt/IBM/db2/V8.1
- Linux: /opt/IBM/db2/V8.1
- Solaris-Betriebsumgebung: /opt/IBM/db2/V8.1



### Zugehörige Konzepte:

- „DB2 Information - Unterstützung“ auf Seite 450
- „DB2 Information - Unterstützung: Installationsszenarios“ auf Seite 452

### Zugehörige Tasks:

- „Installieren von DB2 mit Hilfe einer Antwortdatei (UNIX)“ in *Installation und Konfiguration Ergänzung*
- „Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung'“ auf Seite 461
- „Anzeigen von Themen in der gewünschten Sprache in 'DB2 Information - Unterstützung'“ auf Seite 462
- „Aufrufen von 'DB2 Information - Unterstützung'“ auf Seite 459
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (Windows)“ auf Seite 457

---

## Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (Windows)

Es gibt drei Möglichkeiten, auf die DB2-Produktdokumentation zuzugreifen: auf der IBM Website, auf einem Intranet-Server oder auf eine auf dem lokalen Computer installierte Version. Standardmäßig greifen DB2-Produkte auf die DB2-Dokumentation auf der IBM Website zu. Wenn Sie jedoch auf die DB2-Dokumentation auf einem Intranet-Server oder auf dem eigenen Computer zugreifen möchten, müssen Sie die DB2-Dokumentation von der CD 'DB2 Information - Unterstützung' aus installieren. Mit dem DB2-Installationsassistenten können Sie Ihre Installationseinstellungen definieren und 'DB2 Information - Unterstützung' auf einem Computer installieren, der ein Windows-Betriebssystem verwendet.

### Voraussetzungen:

Dieser Abschnitt erläutert die Voraussetzungen für Hardware, Betriebssystem, Software und Kommunikation zum Installieren von 'DB2 Information - Unterstützung' unter Windows.

#### • Hardwarevoraussetzungen

Sie benötigen einen der folgenden Prozessoren:

- 32-Bit-Computer: eine Pentium- oder mit Pentium kompatible CPU

#### • Betriebssystemvoraussetzungen

Sie benötigen eines der folgenden Betriebssysteme:

- Windows 2000
- Windows XP

**Anmerkung:** 'DB2 Information - Unterstützung' kann unter einem Teil der Windows-Betriebssysteme ausgeführt werden, unter denen DB2-Clients unterstützt werden. Daher wird empfohlen, entweder über die IBM Website auf 'DB2 Information - Unterstützung' zuzugreifen oder 'DB2 Information - Unterstützung' auf einem Intranet-Server zu installieren und dort auf die Dokumentation zuzugreifen.

#### • Softwarevoraussetzungen

- Unterstützte Browser:
  - Mozilla 1.0 oder höher
  - Internet Explorer Version 5.5 oder 6.0 (Version 6.0 für Windows XP)

- **Kommunikationsvoraussetzungen**

- TCP/IP

**Einschränkungen:**

- Sie benötigen einen Benutzereintrag mit Administratorberechtigung, um 'DB2 Information - Unterstützung' zu installieren.

**Vorgehensweise:**

Um 'DB2 Information - Unterstützung' mit Hilfe des DB2-Installationsassistenten zu installieren, gehen Sie wie folgt vor:

1. Melden Sie sich mit dem für die Installation von 'DB2 Information - Unterstützung' definierten Benutzereintrag am System an.
2. Legen Sie die CD in das Laufwerk ein. Die IBM DB2 Setup-Klickstartleiste wird von der Funktion für automatische Ausführung gestartet, sofern diese Funktion aktiviert ist.
3. Der DB2-Installationsassistent ermittelt die Systemsprache und startet das Installationsprogramm für diese Sprache. Wenn Sie das Installationsprogramm nicht in Englisch ausführen möchten oder wenn beim automatischen Starten des Programms ein Fehler aufgetreten ist, können Sie den DB2-Installationsassistenten auch manuell starten.

Um den DB2-Installationsassistenten manuell zu starten, gehen Sie wie folgt vor:

- a. Klicken Sie **Start** an, und wählen Sie die Option **Ausführen** aus.
- b. Geben Sie im Feld **Öffnen** den folgenden Befehl ein:

```
x:\setup.exe /i zweistellige sprachenkennung
```

Hierbei steht *x*: für das CD-Laufwerk und *zweistellige sprachenkennung* für die Sprache, in der das Installationsprogramm ausgeführt werden soll.

- c. Klicken Sie **OK** an.
4. Die IBM DB2-Klickstartleiste wird geöffnet. Um direkt mit der Installation von 'DB2 Information - Unterstützung' fortzufahren, klicken Sie **Produkt installieren** an. Die Onlinehilfe enthält Informationen, die Sie durch die verbleibenden Schritte der Installation führen. Um die Onlinehilfe aufzurufen, klicken Sie **Hilfe** an. Sie können jederzeit **Abbrechen** anklicken, um die Installation zu beenden.
  5. Klicken Sie im Fenster **Wählen Sie das zu installierende Produkt aus** den Knopf **Weiter** an.
  6. Klicken Sie **Weiter** im Fenster **Willkommen beim DB2-Installationsassistenten** an. Der DB2-Installationsassistent leitet Sie durch die erforderlichen Schritte zum Installieren des Programms.
  7. Um mit der Installation fortfahren zu können, müssen Sie die Lizenzvereinbarung akzeptieren. Wählen Sie auf der Seite **Lizenzvereinbarung** die Option **Bedingungen in der Lizenzvereinbarung anerkennen** aus, und klicken Sie **Weiter** an.
  8. Wählen Sie **DB2 Information - Unterstützung auf diesem Computer installieren** auf der Seite **Installationsaktion auswählen** aus. Wenn Sie 'DB2 Information - Unterstützung' zu einem späteren Zeitpunkt auf diesem Computer oder anderen Computern mit Hilfe einer Antwortdatei installieren möchten, wählen Sie **Ihre Einstellungen in einer Antwortdatei speichern** aus. Klicken Sie **Weiter** an.

9. Wählen Sie auf der Seite **Zu installierende Sprachen auswählen** die Sprachen aus, in denen 'DB2 Information - Unterstützung' installiert werden soll. Klicken Sie den Knopf **Weiter** an.
10. Konfigurieren Sie 'DB2 Information - Unterstützung' auf der Seite **Port von DB2 Information - Unterstützung angeben** für eingehende Kommunikation. Klicken Sie **Weiter** an, um mit der Installation fortzufahren.
11. Überprüfen Sie auf der Seite **Kopieren der Dateien starten** noch einmal die von Ihnen ausgewählten Installationseinstellungen. Wenn Sie die Einstellungen ändern möchten, klicken Sie **Zurück** an. Klicken Sie **Installieren** an, um die Dateien von 'DB2 Information - Unterstützung' auf Ihren Computer zu kopieren.

Sie haben die Möglichkeit, 'DB2 Information - Unterstützung' mit Hilfe einer Antwortdatei zu installieren. Sie können auch den Befehl **db2rspgn** verwenden, um eine Antwortdatei auf der Grundlage einer vorhandenen Installation zu generieren.

Die Dateien `db2.log` und `db2wi.log` im Verzeichnis 'Eigene Dateien'\DB2LOG\ enthalten Informationen zu Fehlern, die während der Installation aufgetreten sind. Die Position des Verzeichnisses 'Eigene Dateien' hängt von den Einstellungen Ihres Computers ab.

Die Datei `db2wi.log` erfasst die neuesten DB2-Installationsinformationen. Die Datei `db2.log` erfasst die Protokollinformationen von DB2-Produktinstallationen.

#### **Zugehörige Konzepte:**

- „DB2 Information - Unterstützung“ auf Seite 450
- „DB2 Information - Unterstützung: Installationsszenarios“ auf Seite 452

#### **Zugehörige Tasks:**

- „Installieren eines DB2-Produkts mit Hilfe einer Antwortdatei (Windows)“ in *Installation und Konfiguration Ergänzung*
- „Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung'“ auf Seite 461
- „Anzeigen von Themen in der gewünschten Sprache in 'DB2 Information - Unterstützung'“ auf Seite 462
- „Aufrufen von 'DB2 Information - Unterstützung'“ auf Seite 459
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (UNIX)“ auf Seite 454

#### **Zugehörige Referenzen:**

- „db2rspgn - Response File Generator Command (Windows)“ in *Command Reference*

---

## **Aufrufen von 'DB2 Information - Unterstützung'**

'DB2 Information - Unterstützung' bietet Ihnen die Möglichkeit, auf alle Informationen zuzugreifen, die Sie zur Verwendung der DB2-Produkte für die Betriebssysteme Linux, UNIX und Windows, wie z. B. DB2 Universal Database, DB2 Connect, DB2 Information Integrator und DB2 Query Patroller, benötigen.

Rufen Sie 'DB2 Information - Unterstützung' auf eine der folgenden Arten auf:

- Von einem Computer aus, auf dem ein DB2 UDB-Client oder -Server installiert ist
- Von einem Intranet-Server oder einem lokalen Computer aus, auf dem 'DB2 Information - Unterstützung' installiert ist
- Über die IBM Website

#### Voraussetzungen:

Führen Sie vor dem Aufrufen von 'DB2 Information - Unterstützung' folgende Schritte aus:

- *Optional:* Konfigurieren des Browsers für die Anzeige der Themen in der gewünschten Landessprache
- *Optional:* Konfigurieren des DB2-Clients für die Verwendung der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung'

#### Vorgehensweise:

Gehen Sie wie folgt vor, um 'DB2 Information - Unterstützung' auf einem Computer aufzurufen, auf dem ein DB2 UDB-Client oder -Server installiert ist:

- Wählen Sie (unter Windows) **Start** → **Programme** → **IBM DB2** → **Information** → **DB2 Information - Unterstützung** aus.
- Geben Sie in der Befehlszeile Folgendes ein:
  - Unter Linux und UNIX: Geben Sie den Befehl **db2icdocs** ein.
  - Unter Windows: Geben Sie den Befehl **db2icdocs.exe** ein.

Gehen Sie wie folgt vor, um die auf einem Intranet-Server oder lokalen Computer installierte Komponente 'DB2 Information - Unterstützung' in einem Webbrowser zu öffnen:

- Öffnen Sie die Webseite unter <http://<hostname>:<portnummer>/>. Dabei stellt <hostname> den Namen des Hosts dar und <portnummer> die Nummer des Ports, an dem 'DB2 Information - Unterstützung' verfügbar ist.

Gehen Sie wie folgt vor, um 'DB2 Information - Unterstützung' auf der IBM Website in einem Webbrowser zu öffnen:

- Öffnen Sie die Webseite unter [publib.boulder.ibm.com/infocenter/db2help/](http://publib.boulder.ibm.com/infocenter/db2help/).

#### Zugehörige Konzepte:

- „DB2 Information - Unterstützung“ auf Seite 450
- „DB2 Information - Unterstützung: Installationsszenarios“ auf Seite 452

#### Zugehörige Tasks:

- „Aufrufen der Kontexthilfe über ein DB2-Tool“ auf Seite 469
- „Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung'“ auf Seite 461
- „Aufrufen der Hilfe für Befehle über den Befehlszeilenprozessor“ auf Seite 471
- „Festlegen der Speicherposition für den Zugriff auf 'DB2 Information - Unterstützung': Gemeinsame GUI - Hilfe“

#### Zugehörige Referenzen:

- „HELP Command“ in *Command Reference*

---

## Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten Komponente 'DB2 Information - Unterstützung'

Die Komponente 'DB2 Information - Unterstützung', auf die Sie über <http://publib.boulder.ibm.com/infocenter/db2help/> zugreifen können, wird in regelmäßigen Abständen durch neue oder geänderte Dokumentationen aktualisiert. IBM stellt in bestimmten Fällen auch Aktualisierungen von 'DB2 Information - Unterstützung' zum Download bereit, die Sie auf Ihrem Computer oder Intranet-Server installieren können. Durch die Aktualisierung von 'DB2 Information - Unterstützung' werden keine DB2-Client- oder -Serverprodukte aktualisiert.

### Voraussetzungen:

Sie benötigen Zugriff auf einen Computer, der über eine Verbindung zum Internet verfügt.

### Vorgehensweise:

Gehen Sie wie folgt vor, um die auf Ihrem Computer bzw. Intranet-Server installierte Komponente 'DB2 Information - Unterstützung' zu aktualisieren:

1. Öffnen Sie 'DB2 Information - Unterstützung' auf der IBM Website unter <http://publib.boulder.ibm.com/infocenter/db2help/>.
2. Klicken Sie im Downloadbereich der Eingangsseite den Link **DB2 Universal Database-Dokumentation** unter der Überschrift für Service und Unterstützung an.
3. Stellen Sie fest, ob die Version der installierten Komponente 'DB2 Information - Unterstützung' veraltet ist, indem Sie die Stufe des neuesten aktualisierten Dokumentationsimage mit der installierten Dokumentationsstufe vergleichen. Die installierte Dokumentationsstufe ist auf der Eingangsseite von 'DB2 Information - Unterstützung' aufgeführt.
4. Wenn eine neuere Version von 'DB2 Information - Unterstützung' verfügbar ist, laden Sie das neueste aktualisierte Image für *DB2 Information - Unterstützung* für das von Ihnen verwendete Betriebssystem herunter.
5. Befolgen Sie zur Installation des aktualisierten Image für *DB2 Information - Unterstützung* die Anweisungen auf der Webseite.

### Zugehörige Konzepte:

- „DB2 Information - Unterstützung: Installationsszenarios“ auf Seite 452

### Zugehörige Tasks:

- „Aufrufen von 'DB2 Information - Unterstützung'“ auf Seite 459
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (UNIX)“ auf Seite 454
- „Installation von 'DB2 Information - Unterstützung' mit dem DB2-Installationsassistenten (Windows)“ auf Seite 457

---

## Anzeigen von Themen in der gewünschten Sprache in 'DB2 Information - Unterstützung'

In 'DB2 Information - Unterstützung' werden Themen, wenn möglich, in der Sprache angezeigt, die in den Vorgaben Ihres Browsers angegeben ist. Falls ein Thema nicht in die gewünschte Sprache übersetzt wurde, wird es in 'DB2 Information - Unterstützung' in Englisch angezeigt.

### Vorgehensweise:

Um Themen in der gewünschten Sprache im Browser 'Internet Explorer' anzuzeigen, gehen Sie wie folgt vor:

1. Klicken Sie im Internet Explorer **Extras** —> **Internetoptionen...** —> **Sprachen...** an. Das Fenster **Spracheinstellung** wird geöffnet.
2. Stellen Sie sicher, dass die gewünschte Sprache als erster Eintrag in der Liste angegeben ist.
  - Klicken Sie den Knopf **Hinzufügen...** an, um eine neue Sprache zur Liste hinzuzufügen.

**Anmerkung:** Das Hinzufügen einer Sprache bedeutet nicht zwangsläufig, dass der Computer über die erforderlichen Schriftarten verfügt, um die Themen in der gewünschten Sprache anzuzeigen.

- Um eine Sprache an den Anfang der Liste zu verschieben, wählen Sie zunächst die gewünschte Sprache und anschließend den Knopf **Nach oben** aus, bis die Sprache an erster Stelle in der Liste steht.
3. Aktualisieren Sie die Seite, um 'DB2 Information - Unterstützung' in der gewünschten Sprache anzuzeigen.

Um Themen in der gewünschten Sprache im Browser 'Mozilla' anzuzeigen, gehen Sie wie folgt vor:

1. Wählen Sie in Mozilla **Bearbeiten** —> **Einstellungen** —> **Sprachen** aus. Die Anzeige für die Auswahl der Sprache wird im Fenster mit den Einstellungen aufgerufen.
2. Stellen Sie sicher, dass die gewünschte Sprache als erster Eintrag in der Liste angegeben ist.
  - Wenn Sie eine neue Sprache hinzufügen möchten, klicken Sie den Knopf **Hinzufügen...** an, um eine Sprache im entsprechenden Fenster auszuwählen.
  - Um eine Sprache an den Anfang der Liste zu verschieben, wählen Sie zunächst die gewünschte Sprache und anschließend den Knopf **Nach oben** aus, bis die Sprache an erster Stelle in der Liste steht.
3. Aktualisieren Sie die Seite, um 'DB2 Information - Unterstützung' in der gewünschten Sprache anzuzeigen.

### Zugehörige Konzepte:

- „DB2 Information - Unterstützung“ auf Seite 450

---

## DB2-Dokumentation in PDF-Format und gedrucktem Format

In den folgenden Tabellen sind die offiziellen Buchtitel, Formularnummern und PDF-Dateinamen aufgeführt. Zum Bestellen von Hardcopybüchern benötigen Sie den offiziellen Buchtitel. Zum Drucken der PDF-Version benötigen Sie den PDF-Dateinamen.

Die DB2-Dokumentation ist in die folgenden Kategorien unterteilt:

- DB2-Kerninformationen
- Verwaltungsinformationen
- Informationen zur Anwendungsentwicklung
- Informationsmanagement
- Informationen zu DB2 Connect
- Einführungsinformationen
- Lernprogramminformationen
- Informationen zu Zusatzkomponenten
- Release-Informationen

In den folgenden Tabellen wird für die einzelnen Bücher der DB2-Bibliothek beschrieben, welche Informationen zum Bestellen von Hardcopies bzw. zum Drucken oder Anzeigen der PDF-Versionen erforderlich sind. Eine vollständige Beschreibung der in der DB2-Bibliothek verfügbaren Bücher finden Sie im IBM Publications Center unter folgender Adresse:  
[www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order).

## DB2-Kerninformationen

Diese Bücher enthalten grundlegende Informationen für alle DB2-Benutzer. Diese Informationen sind sowohl für Programmierer als auch für Datenbankadministratoren geeignet und unterstützen Sie bei der Arbeit mit DB2 Connect, DB2 Warehouse Manager und anderen DB2-Produkten.

*Tabelle 53. DB2-Kerninformationen*

Name	IBM Form	PDF-Dateiname
<i>IBM DB2 Universal Database Command Reference</i>	SC09-4828	db2n0e81
<i>IBM DB2 Universal Database Glossar</i>	Keine Formnummer	db2t0g81
<i>IBM DB2 Universal Database Fehlermeldungen, Band 1</i>	GC12-3043, nicht als Hardcopy verfügbar	db2m1g81
<i>IBM DB2 Universal Database Fehlermeldungen, Band 2</i>	GC12-3042, nicht als Hardcopy verfügbar	db2m2g81
<i>IBM DB2 Universal Database Neue Funktionen</i>	SC12-3044	db2q0g81

## Verwaltungsinformationen

Die Informationen in diesen Büchern umfassen die Themen, die zum effektiven Entwerfen, Implementieren und Verwalten von DB2-Datenbanken, Data Warehouses und Systemen zusammenschlossener Datenbanken erforderlich sind.

*Tabelle 54. Verwaltungsinformationen*

Name	IBM Form	PDF-Dateiname
<i>IBM DB2 Universal Database Systemverwaltung: Konzept</i>	SC12-3057	db2d1g81
<i>IBM DB2 Universal Database Systemverwaltung: Implementierung</i>	SC12-3059	db2d2g81

*Tabelle 54. Verwaltungsinformationen (Forts.)*

<b>Name</b>	<b>IBM Form</b>	<b>PDF-Dateiname</b>
<i>IBM DB2 Universal Database Systemverwaltung: Optimierung</i>	SC12-3058	db2d3g81
<i>IBM DB2 Universal Database Administrative API Reference</i>	SC09-4824	db2b0e81
<i>IBM DB2 Universal Database Dienstprogramme für das Versetzen von Daten Handbuch und Referenz</i>	SC12-3055	db2dmg81
<i>IBM DB2 Universal Database Datenwiederherstellung und hohe Verfügbarkeit Handbuch und Referenz</i>	SC12-3054	db2hag81
<i>IBM DB2 Universal Database Data Warehouse-Zentrale Verwaltung</i>	SC12-3068	db2ddg81
<i>IBM DB2 Universal Database SQL Reference, Volume 1</i>	SC09-4844	db2s1e81
<i>IBM DB2 Universal Database SQL Reference, Volume 2</i>	SC09-4845	db2s2e81
<i>IBM DB2 Universal Database System Monitor Guide and Reference</i>	SC09-4847	db2f0e81

## Informationen zur Anwendungsentwicklung

Die Informationen in diesen Büchern sind besonders für Anwendungsentwickler und Programmierer von Interesse, die mit DB2 Universal Database (DB2 UDB) arbeiten. Sie finden hier Informationen zu den unterstützten Programmiersprachen und Compilern sowie die Dokumentation, die für den Zugriff auf DB2 UDB über die verschiedenen unterstützten Programmierschnittstellen, z. B. eingebettetes SQL, ODBC, JDBC, SQLJ und CLI, erforderlich ist. Wenn Sie die Komponente 'DB2 Information - Unterstützung' verwenden, können Sie auch auf HTML-Versionen des Quellcodes für die Beispielprogramme zugreifen.

*Tabelle 55. Informationen zur Anwendungsentwicklung*

<b>Name</b>	<b>IBM Form</b>	<b>PDF-Dateiname</b>
<i>IBM DB2 Universal Database Application Development Guide: Building and Running Applications</i>	SC09-4825	db2axe81
<i>IBM DB2 Universal Database Application Development Guide: Programming Client Applications</i>	SC09-4826	db2a1e81
<i>IBM DB2 Universal Database Application Development Guide: Programming Server Applications</i>	SC09-4827	db2a2e81
<i>IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1</i>	SC09-4849	db2l1e81



*Tabelle 55. Informationen zur Anwendungsentwicklung (Forts.)*

<b>Name</b>	<b>IBM Form</b>	<b>PDF-Dateiname</b>
<i>IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2</i>	SC09-4850	db2l2e81
<i>IBM DB2 Universal Database Data Warehouse Center Application Integration Guide</i>	SC27-1124	db2ade81
<i>IBM DB2 XML Extender Verwaltung und Programmierung</i>	SC12-3062	db2sxs81

## Informationsmanagement

Die Informationen in diesen Büchern beschreiben den Einsatz von Komponenten, mit denen Sie die Data Warehousing- und Analysefunktionen von DB2 Universal Database erweitern können.

*Tabelle 56. Informationsmanagement*

<b>Name</b>	<b>IBM Form</b>	<b>PDF-Dateiname</b>
<i>IBM DB2 Warehouse Manager Standard Edition Informationskatalogzentrale Verwaltung</i>	SC12-3070	db2dig81
<i>IBM DB2 Warehouse Manager Standard Edition Installation</i>	GC12-3069	db2idg81
<i>IBM DB2 Warehouse Manager Standard Edition Managing ETI Solution Conversion Programs with DB2 Warehouse Manager</i>	SC18-7727	iwhe1mste80

## Informationen zu DB2 Connect

Die Informationen in dieser Kategorie beschreiben den Zugriff auf Daten auf großen und mittleren Serversystemen mit Hilfe von DB2 Connect Enterprise Edition oder DB2 Connect Personal Edition.

*Tabelle 57. Informationen zu DB2 Connect*

<b>Name</b>	<b>IBM Form</b>	<b>PDF-Dateiname</b>
<i>IBM Konnektivität Ergänzung</i>	Keine Formnummer	db2h1g81
<i>IBM DB2 Connect Enterprise Edition Einstieg</i>	GC12-3051	db2c6g81
<i>IBM DB2 Connect Personal Edition Einstieg</i>	GC12-3049	db2c1g81
<i>IBM DB2 Connect Benutzerhandbuch</i>	SC12-3048	db2c0g81

## Einführungsinformationen

Die Informationen in dieser Kategorie unterstützen Sie beim Installieren und Konfigurieren von Servern, Clients und anderen DB2-Produkten.

Tabelle 58. Einführungsinformationen

Name	IBM Form	PDF-Dateiname
IBM DB2 Universal Database für DB2-Clients Einstieg	GC12-3052, nicht als Hardcopy verfügbar	db2itg81
IBM DB2 Universal Database für DB2-Server Einstieg	GC12-3047	db2isg81
IBM DB2 Universal Database Personal Edition Einstieg	GC12-3045	db2i1g81
IBM DB2 Universal Database Installation und Konfiguration Ergänzung	GC12-3046, nicht als Hardcopy verfügbar	db2iyg81
IBM DB2 Universal Database Data Links Manager Einstieg	GC12-3056	db2z6g81

## Lernprogramminformationen

In den Lernprogramminformationen werden DB2-Funktionen vorgestellt. Darüber hinaus wird die Ausführung verschiedener Tasks beschrieben.

Tabelle 59. Lernprogramminformationen

Name	IBM Form	PDF-Dateiname
Lernprogramm für das Informationsmanagement: Data Warehouse - Einführung	Keine Formnummer	db2tug81
Lernprogramm für das Informationsmanagement: Data Warehouse - Weiterführende Informationen	Keine Formnummer	db2tag81
Lernprogramm für die Informationskatalogzentrale	Keine Formnummer	db2aig81
Video Central für e-business Lernprogramm	Keine Formnummer	db2twg81
Lernprogramm für Visual Explain	Keine Formnummer	db2tv81

## Informationen zu Zusatzkomponenten

Die Informationen in dieser Kategorie beschreiben das Arbeiten mit den DB2-Zusatzkomponenten.

Tabelle 60. Informationen zu Zusatzkomponenten

Name	IBM Form	PDF-Dateiname
IBM DB2 Cube Views Handbuch und Referenz	n/v	db2aag81
IBM DB2 Query Patroller-Handbuch: Installation, Verwaltung und Verwendung	GC12-3225	db2dwg81
IBM DB2 Spatial Extender und Geodetic Extender Benutzer- und Referenzhandbuch	SC12-3063	db2sbg81

Tabelle 60. Informationen zu Zusatzkomponenten (Forts.)

Name	IBM Form	PDF-Dateiname
IBM DB2 Universal Database Data Links Manager Administration Guide and Reference	SC27-1221	db2z0e82
DB2 Net Search Extender Verwaltung und Benutzerhandbuch	SH12-3021	n/v

**Anmerkung:** Die HTML-Version dieses Dokuments wird nicht von der HTML-Dokumentations-CD installiert.

## Release-Informationen

Die Release-Informationen enthalten zusätzliche Informationen für das verwendete Produktrelease und die verwendete FixPak-Stufe. Die Release-Informationen enthalten außerdem Zusammenfassungen der Dokumentationsaktualisierungen in den verschiedenen Releases, Aktualisierungen und FixPaks.

Tabelle 61. Release-Informationen

Name	IBM Form	PDF-Dateiname
DB2 Release-Informationen	Siehe Anmerkung.	Siehe Anmerkung.
DB2 Installationsinformationen	Nur auf der Produkt-CD-ROM verfügbar.	n/v

**Anmerkung:** Die Release-Informationen stehen in den folgenden Formaten zur Verfügung:

- XHTML und Textformat auf den Produkt-CDs
- PDF-Format auf der CD mit der PDF-Dokumentation

Darüber hinaus sind die Abschnitte zu *bekanntem Problemen und Fehlerumgehungen* sowie zur *Inkompatibilität zwischen einzelnen Releases*, die Teil der Release-Informationen sind, auch über 'DB2 Information - Unterstützung' verfügbar.

Informationen zum Anzeigen der Release-Informationen in Textformat auf UNIX-Plattformen finden Sie in der Datei `Release.Notes`. Diese Datei befindet sich im Verzeichnis `DB2DIR/Readme/%L`. Hierbei steht `%L` für die länderspezifische Angabe und `DB2DIR` für eine der folgenden Angaben:

- Für AIX-Betriebssysteme: `/usr/opt/db2_08_01`
- Für alle anderen UNIX-Betriebssysteme: `/opt/IBM/db2/V8.1`

### Zugehörige Konzepte:

- „DB2-Dokumentation und Hilfe“ auf Seite 449

### Zugehörige Tasks:

- „Drucken von DB2-Büchern mit PDF-Dateien“ auf Seite 468
- „Bestellen gedruckter DB2-Bücher“ auf Seite 468
- „Aufrufen der Kontexthilfe über ein DB2-Tool“ auf Seite 469

---

## Drucken von DB2-Büchern mit PDF-Dateien

DB2-Bücher können mit Hilfe der PDF-Dateien auf der CD mit der *DB2-PDF-Dokumentation* gedruckt werden. Mit Adobe Acrobat Reader können Sie entweder das gesamte Handbuch oder bestimmte Seitenbereiche des Handbuchs ausdrucken.

### Voraussetzungen:

Stellen Sie sicher, dass Adobe Acrobat Reader installiert ist. Falls Sie Adobe Acrobat Reader noch nicht installiert haben, finden Sie das Produkt auf der Adobe-Website unter folgender Adresse: [www.adobe.com](http://www.adobe.com)

### Vorgehensweise:

Gehen Sie wie folgt vor, um ein DB2-Buch mit einer PDF-Datei auszudrucken:

1. Legen Sie die CD mit der *DB2-PDF-Dokumentation* in das CD-ROM-Laufwerk ein. Hängen Sie unter UNIX-Betriebssystemen die CD mit der DB2-PDF-Dokumentation an. Informationen zum Anhängen einer CD unter UNIX-Betriebssystemen finden Sie im Handbuch *Einstieg* für das jeweilige Betriebssystem.
2. Öffnen Sie `index.htm`. Die Datei wird in einem Browserfenster geöffnet.
3. Klicken Sie den Titel der PDF an, die Sie aufrufen möchten. Die PDF wird in Acrobat Reader geöffnet.
4. Wählen Sie **Datei** → **Drucken** aus, um einen beliebigen Teil des gewünschten Buches zu drucken.

### Zugehörige Konzepte:

- „DB2 Information - Unterstützung“ auf Seite 450

### Zugehörige Tasks:

- „Anhängen der CD-ROM (AIX)“ in *DB2 Universal Database für DB2-Server Einstieg*
- „Anhängen der CD-ROM (HP-UX)“ in *DB2 Universal Database für DB2-Server Einstieg*
- „Anhängen der CD-ROM (Linux)“ in *DB2 Universal Database für DB2-Server Einstieg*
- „Bestellen gedruckter DB2-Bücher“ auf Seite 468
- „Anhängen der CD-ROM (Solaris-Betriebsumgebung)“ in *DB2 Universal Database für DB2-Server Einstieg*

### Zugehörige Referenzen:

- „DB2-Dokumentation in PDF-Format und gedrucktem Format“ auf Seite 462

---

## Bestellen gedruckter DB2-Bücher

Wenn Sie die Hardcopyversion der Bücher bevorzugen, können Sie sie auf eine der nachfolgend aufgeführten Arten bestellen.

### Vorgehensweise:

In bestimmten Ländern oder Regionen können gedruckte Bücher bestellt werden. Auf der Website mit IBM Veröffentlichungen für das jeweilige Land bzw. die jeweilige Region finden Sie Informationen darüber, ob dieser Service im betreffenden

Land bzw. in der betreffenden Region angeboten wird. Wenn die Veröffentlichungen bestellt werden können, haben Sie folgende Möglichkeiten:

- Wenden Sie sich an den zuständigen IBM Vertragshändler oder Vertriebsbeauftragten. Informationen zum lokalen IBM Ansprechpartner finden Sie im globalen IBM Verzeichnis für Kontakte unter folgender Adresse:  
[www.ibm.com/planetwide](http://www.ibm.com/planetwide).
- Weitere Informationen enthält das IBM Publications Center unter <http://www.ibm.com/shop/publications/order>. Die Möglichkeit, Bücher über das IBM Publications Center zu bestellen, besteht möglicherweise nicht in allen Ländern.

Die gedruckten Bücher sind zu dem Zeitpunkt, an dem das DB2-Produkt verfügbar gemacht wird, identisch mit den PDF-Versionen auf der CD mit der *DB2-PDF-Dokumentation*. Darüber hinaus stimmt der Inhalt der gedruckten Bücher mit den entsprechenden Informationen auf der CD für *DB2 Information - Unterstützung* überein. Diese CD enthält jedoch zusätzliche Informationen, die in den PDF-Büchern nicht enthalten sind (wie beispielsweise SQL-Verwaltungsroutinen und HTML-Beispiele). Nicht alle Bücher, die auf der CD mit der DB2-PDF-Dokumentation verfügbar sind, können als Hardcopy bestellt werden.

**Anmerkung:** 'DB2 Information - Unterstützung' wird häufiger aktualisiert als die PDF- oder die Hardcopyversion der Bücher. Installieren Sie die Dokumentationsupdates, sobald diese verfügbar sind, oder greifen Sie über 'DB2 Information - Unterstützung' unter <http://publib.boulder.ibm.com/infocenter/db2help/> auf die neuesten Informationen zu.

**Zugehörige Tasks:**

- „Drucken von DB2-Büchern mit PDF-Dateien“ auf Seite 468

**Zugehörige Referenzen:**

- „DB2-Dokumentation in PDF-Format und gedrucktem Format“ auf Seite 462

---

## Aufrufen der Kontexthilfe über ein DB2-Tool

Die Kontexthilfe bietet Informationen zu den Tasks bzw. Steuerelementen, die einem bestimmten Fenster, Notizbuch, Assistenten oder Advisor zugeordnet sind. Die Kontexthilfe steht in allen DB2-Verwaltungs- und -entwicklungstools zur Verfügung, die über eine grafische Benutzerschnittstelle verfügen. Zwei Arten der Kontexthilfe stehen zur Verfügung:

- Die über den Knopf **Hilfe** aufgerufenen Hilfetexte, der in jedem Fenster bzw. Notizbuch zur Verfügung steht.
- Die Kurzhilfe. Hierbei handelt es sich um Informationsfenster, die angezeigt werden, wenn sich der Mauszeiger auf einem Feld oder Steuerelement befindet oder wenn bei der Auswahl eines Feldes oder Steuerelements in einem Fenster, Notizbuch, Assistenten oder Advisor die Taste F1 gedrückt wird.

Über den Knopf **Hilfe** können Sie auf Übersichtsinformationen, Informationen zu Voraussetzungen sowie Informationen zu Tasks zugreifen. In der Kurzhilfe werden die einzelnen Felder und Steuerelemente beschrieben.

**Vorgehensweise:**

Gehen Sie wie folgt vor, um Kontexthilfe aufzurufen:

- Hilfe zu Fenstern und Notizbüchern können Sie anzeigen, indem Sie eines der DB2-Tools aufrufen und anschließend ein beliebiges Fenster oder Notizbuch öffnen. Klicken Sie den Knopf **Hilfe** in der rechten unteren Ecke des Fensters bzw. Notizbuchs an, um die Kontexthilfe aufzurufen.

Zugriff auf die Kontexthilfe besteht darüber hinaus über den Menüpunkt **Hilfe** am oberen Rand jeder Zentrale der DB2-Tools.

Innerhalb von Assistenten und Advisorfunktionen klicken Sie den Link für die Taskübersicht auf der ersten Seite an, um die Kontexthilfe aufzurufen.

- Kurzhilfe zu einzelnen Steuerelementen eines Fensters oder Notizbuchs können Sie aufrufen, indem Sie das gewünschte Steuerelement anklicken und anschließend **F1** drücken. Die Kurzhilfeinformationen mit Details zum jeweiligen Steuerelement werden in einem gelben Fenster angezeigt.

**Anmerkung:** Wenn die Kurzhilfe angezeigt werden soll, sobald sich der Mauszeiger auf einem Feld oder Steuerelement befindet, wählen Sie das Markierungsfeld **Kurzhilfe automatisch anzeigen** auf der Seite **Dokumentation** des Notizbuchs 'Tools - Einstellungen' aus.

Ähnlich wie die Kurzhilfe sind auch Dialogfenster mit Diagnoseinformationen eine Form der kontextbezogenen Hilfe; sie enthalten Regeln für die Dateneingabe. Diese Diagnoseinformationen werden in einem violetten Fenster angezeigt, das aufgerufen wird, wenn die eingegebenen Daten nicht gültig oder nicht ausreichend sind. Die Kontexthilfe mit Diagnoseinformationen kann für folgende Felder angezeigt werden:

- Musseingabefelder
- Felder, in denen die Daten einem bestimmten Format entsprechen müssen, wie z. B. Datumsfelder

#### Zugehörige Tasks:

- „Aufrufen von 'DB2 Information - Unterstützung'“ auf Seite 459
- „Aufrufen der Hilfe für Nachrichten über den Befehlszeilenprozessor“ auf Seite 470
- „Aufrufen der Hilfe für Befehle über den Befehlszeilenprozessor“ auf Seite 471
- „Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor“ auf Seite 471
- „Zugriff auf 'DB2 Information - Unterstützung': Konzepthilfe“
- „Verwenden der DB2 UDB-Hilfe: Gemeinsame GUI - Hilfe“
- „Festlegen der Speicherposition für den Zugriff auf 'DB2 Information - Unterstützung': Gemeinsame GUI - Hilfe“
- „Einrichten des Zugriffs auf DB2-Kontexthilfe und -Dokumentation: Gemeinsame GUI - Hilfe“

---

## Aufrufen der Hilfe für Nachrichten über den Befehlszeilenprozessor

Die Hilfe für Nachrichten beschreibt die Ursache von Nachrichten und die Aktionen, die der Benutzer zur Behebung des aufgetretenen Fehlers ausführen sollte.

#### Vorgehensweise:

Zum Aufrufen der Hilfe für Nachrichten müssen Sie den Befehlszeilenprozessor öffnen und Folgendes eingeben:

? XXXnnnnn

Dabei ist *XXXnnnnn* eine gültige Nachrichtenkennung.

So kann beispielsweise durch die Eingabe von `? SQL30081` die Hilfe zur Nachricht SQL30081 angezeigt werden.

**Zugehörige Konzepte:**

- „Nachrichten - Einführung“ in *Fehlernachrichten Band 1*

**Zugehörige Referenzen:**

- „db2 - Command Line Processor Invocation Command“ in *Command Reference*

---

## Aufrufen der Hilfe für Befehle über den Befehlszeilenprozessor

Die Hilfe für Befehle erläutert die Syntax von Befehlen im Befehlszeilenprozessor.

**Vorgehensweise:**

Zum Aufrufen der Hilfe für Befehle müssen Sie den Befehlszeilenprozessor öffnen und Folgendes eingeben:

`? command`

Dabei stellt *command* ein Schlüsselwort bzw. den vollständigen Befehl dar.

So kann beispielsweise durch die Eingabe von `? catalog` Hilfe für alle CATALOG-Befehle angezeigt werden, während mit `? catalog database` nur Hilfe für den Befehl CATALOG DATABASE angezeigt wird.

**Zugehörige Tasks:**

- „Aufrufen der Kontexthilfe über ein DB2-Tool“ auf Seite 469
- „Aufrufen von 'DB2 Information - Unterstützung'“ auf Seite 459
- „Aufrufen der Hilfe für Nachrichten über den Befehlszeilenprozessor“ auf Seite 470
- „Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor“ auf Seite 471

**Zugehörige Referenzen:**

- „db2 - Command Line Processor Invocation Command“ in *Command Reference*

---

## Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor

DB2 Universal Database gibt für Bedingungen, die auf Grund einer SQL-Anweisung generiert werden können, einen SQLSTATE-Wert zurück. Die SQLSTATE-Hilfe erläutert die Bedeutung der SQL-Statuswerte und der SQL-Statusklassencodes.

**Vorgehensweise:**

Zum Aufrufen der Hilfe für SQL-Statuswerte müssen Sie den Befehlszeilenprozessor öffnen und Folgendes eingeben:

`? sqlstate` oder `? klassencode`

Hierbei steht *sqlstate* für einen gültigen fünfstelligen SQL-Statuswert und *klassencode* für die ersten beiden Ziffern dieses Statuswertes.

So kann beispielsweise durch die Eingabe von ? 08003 Hilfe für den SQL-Statuswert 08003 angezeigt werden, durch die Eingabe von ? 08 Hilfe für den Klassencode 08.

**Zugehörige Tasks:**

- „Aufrufen von 'DB2 Information - Unterstützung'“ auf Seite 459
- „Aufrufen der Hilfe für Nachrichten über den Befehlszeilenprozessor“ auf Seite 470
- „Aufrufen der Hilfe für Befehle über den Befehlszeilenprozessor“ auf Seite 471

---

## DB2-Lernprogramme

Die Lernprogramme von DB2® unterstützen Sie bei der Einarbeitung in die verschiedenen Themenbereiche von DB2 Universal Database. Sie umfassen Übungen mit in einzelne Arbeitsschritte untergliederten Anweisungen zum Entwickeln von Anwendungen, Optimieren der SQL-Abfrageleistung, Arbeiten mit Data Warehouses, Verwalten von Metadaten und Entwickeln von Webservices mit Hilfe von DB2.

**Vorbereitungen:**

Die XHTML-Version der Lernprogramme kann über 'DB2 Information - Unterstützung' unter <http://publib.boulder.ibm.com/infocenter/db2help/> angezeigt werden.

In einigen der Lernprogrammübungen werden Beispieldaten und Codebeispiele verwendet. Informationen zu den spezifischen Voraussetzungen zur Ausführung der Tasks finden Sie in der Beschreibung des jeweiligen Lernprogramms.

**Lernprogramme von DB2 Universal Database:**

Klicken Sie einen der Lernprogrammtitel in der folgenden Liste an, um das entsprechende Lernprogramm aufzurufen.

*Lernprogramm für das Informationsmanagement: Data Warehouse - Einführung*  
Ausführung grundlegender Data Warehousing-Tasks mit Hilfe der Data Warehouse-Zentrale.

*Lernprogramm für das Informationsmanagement: Data Warehouse - Weiterführende Informationen*  
Ausführung weiterführender Data Warehousing-Tasks mit Hilfe der Data Warehouse-Zentrale.

*Lernprogramm für die Informationskatalogzentrale*  
Erstellen und Verwalten eines Informationskatalogs zum Lokalisieren und Verwenden von Metadaten mit Hilfe der Informationskatalogzentrale.

*Lernprogramm für Visual Explain*  
Analysieren, Optimieren und Anpassen von SQL-Anweisungen zur Leistungsverbesserung mit Hilfe von Visual Explain.

---

## Informationen zur Fehlerbehebung in DB2

Eine breite Palette verschiedener Informationen zur Fehlerbestimmung und Fehlerbehebung steht zur Verfügung, um Sie bei der Verwendung von DB2®-Produkten zu unterstützen.



### DB2-Dokumentation

Informationen zur Fehlerbehebung stehen in der gesamten Komponente 'DB2 Information - Unterstützung' sowie in den PDF-Büchern der DB2-Bibliothek zur Verfügung. Folgen Sie der Verzweigung 'Unterstützung und Fehlerbehebung' in der Navigationsbaumstruktur von 'DB2 Information - Unterstützung' (im linken Teilfenster des Browserfensters), um eine umfassende Liste der DB2-Dokumentationen zur Fehlerbehebung aufzurufen.

### DB2-Website mit technischer Unterstützung

Auf der DB2-Website mit technischer Unterstützung finden Sie Informationen zu Problemen und den möglichen Ursachen und Fehlerbehebungsmaßnahmen. Die DB2-Website mit technischer Unterstützung stellt Links zu den neuesten DB2-Veröffentlichungen, technischen Hinweisen (TechNotes), APARs (Authorized Program Analysis Reports), FixPaks, den neuesten Listen mit internen DB2-Fehlercodes sowie weiteren Ressourcen zur Verfügung. Sie können diese Wissensbasis nach möglichen Lösungen für aufgetretene Probleme durchsuchen.

Rufen Sie die DB2-Website mit technischer Unterstützung unter <http://www.ibm.com/software/data/db2/udb/winos2unix/support> auf.

### DB2-Lernprogramme zur Fehlerbestimmung

Auf der Website mit den DB2-Lernprogrammen zur Fehlerbestimmung finden Sie Informationen dazu, wie Sie Fehler, die bei der Verwendung von DB2-Produkten möglicherweise auftreten, rasch identifizieren und beheben können. Eines der Lernprogramme bietet eine Einführung in die verfügbaren DB2-Einrichtungen und -Tools zur Fehlerbestimmung sowie Entscheidungshilfen für deren Verwendung. Andere Lernprogramme befassen sich mit zugehörigen Themen, wie beispielsweise der Fehlerbestimmung für die Datenbanksteuerkomponente, der Fehlerbestimmung für die Leistung und der Fehlerbestimmung für Anwendungen.

Die vollständige Liste der DB2-Lernprogramme zur Fehlerbestimmung finden Sie auf der DB2-Website mit technischer Unterstützung unter <http://www.ibm.com/software/data/support/pdm/db2tutorials.html>.

### Zugehörige Konzepte:

- „DB2 Information - Unterstützung“ auf Seite 450
- „Einführung in die Fehlerbestimmung - Lernprogramm für die technische Unterstützung in DB2“ in *Troubleshooting Guide*

---

## Eingabehilfen

Eingabehilfen unterstützen Benutzer mit körperlichen Behinderungen, wie z. B. eingeschränkter Bewegungsfähigkeit oder Sehkraft, beim erfolgreichen Einsatz von Softwareprodukten. Im Folgenden sind die wichtigsten Eingabehilfen aufgeführt, die in den Produkten von DB2<sup>®</sup> Version 8 zur Verfügung stehen:

- Die gesamte DB2-Funktionalität kann sowohl über die Maus als auch über die Tastatur gesteuert werden. Weitere Informationen hierzu finden Sie unter „Tastatureingabe und Navigation“ auf Seite 474.
- Sie können die Größe und Farbe der verwendeten Schriftarten in den DB2-Schnittstellen anpassen. Weitere Informationen hierzu finden Sie unter „Eingabehilfen für Bildschirme“ auf Seite 474.
- DB2-Produkte unterstützen Anwendungen mit Eingabehilfen, die mit der Java<sup>™</sup> Accessibility API arbeiten. Weitere Informationen hierzu finden Sie unter „Kompatibilität mit Unterstützungseinrichtungen“ auf Seite 474.

- Die DB2-Dokumentation steht in behindertengerechtem Format zur Verfügung. Weitere Informationen hierzu finden Sie unter „Dokumentation im behindertengerechten Format“.

## Tastatureingabe und Navigation

### Tastatureingabe

Die verfügbaren DB2-Tools können unter ausschließlicher Benutzung der Tastatur verwendet werden. Mit entsprechenden Tasten oder Tastenkombinationen können Operationen ausgeführt werden, die auch über die Maus verfügbar sind. Die Standardtastenkombinationen des Betriebssystems werden für die entsprechenden Standardoperationen des Betriebssystems verwendet.

Weitere Informationen zur Verwendung von Tasten oder Tastenkombinationen für die Ausführung von Operationen finden Sie unter " 'Direktaufrufe über die Tastatur: Gemeinsame GUI - Hilfe'.

### Navigation über die Tastatureingabe

Sie können in den Benutzerschnittstellen der DB2-Tools mit Hilfe von Tasten oder Tastenkombinationen navigieren.

Weitere Informationen zur Navigation in den DB2-Tools mit Hilfe der Tastatureingabe finden Sie unter " 'Direktaufrufe über die Tastatur: Gemeinsame GUI - Hilfe'.

### Tastatureingabebereich

Unter UNIX<sup>®</sup>-Betriebssystemen ist der Bereich des aktiven Fensters, in dem die Tastatureingabe wirksam ist, hervorgehoben.

## Eingabehilfen für Bildschirme

Die DB2-Tools stellen Funktionen bereit, mit denen sehbehinderten Benutzern verbesserten Eingabehilfen zur Verfügung stehen. Diese Eingabehilfen umfassen die Unterstützung individuell anpassbarer Schriftarteigenschaften.

### Schriftarteneinstellungen

Über das Notizbuch 'Tools - Einstellungen' können Sie die Farbe, Größe und Schriftart des Textes in Menüs und Dialogfenstern auswählen.

Weitere Informationen zur Angabe von Schriftarteneinstellungen finden Sie unter " 'Ändern der Schriftarten für Menüs und Text: Gemeinsame GUI - Hilfe'.

### Unabhängigkeit von Farben

Zur Verwendung der Funktionen des vorliegenden Produkts ist es nicht erforderlich, zwischen unterschiedlichen Farben differenzieren zu können.

## Kompatibilität mit Unterstützungseinrichtungen

Die Schnittstellen der DB2-Tools unterstützen die Java Accessibility API. Hierdurch wird der Einsatz von Sprachausgabeprogrammen und anderen Unterstützungseinrichtungen für Personen mit Behinderungen mit den DB2-Produkten ermöglicht.

## Dokumentation im behindertengerechten Format

Die Dokumentation für DB2 steht im Format XHTML 1.0 zur Verfügung, das mit den meisten Webbrowsern geöffnet werden kann. XHTML ermöglicht das Aufrufen der Dokumentation mit den Anzeigeeinstellungen, die Sie in Ihrem Browser definiert haben. Darüber hinaus ist der Einsatz von Sprachausgabeprogrammen und anderen Unterstützungseinrichtungen möglich.

Syntaxdiagramme stehen in der Schreibweise mit Trennzeichen zur Verfügung. Dieses Format ist nur dann verfügbar, wenn Sie mit Hilfe eines Sprachausgabeprogramms auf die Onlinedokumentation zugreifen.

**Zugehörige Konzepte:**

- „Syntaxdiagramme in der Schreibweise mit Trennzeichen“ auf Seite 475

**Zugehörige Tasks:**

- „Direktaufrufe über die Tastatur: Gemeinsame GUI - Hilfe“
- „Ändern der Schriftarten für Menüs und Text: Gemeinsame GUI - Hilfe“

---

## Syntaxdiagramme in der Schreibweise mit Trennzeichen

Syntaxdiagramme stehen für Benutzer, die mit Hilfe eines Sprachausgabeprogramms auf 'DB2 Information - Unterstützung' zugreifen, in der Schreibweise mit Trennzeichen zur Verfügung.

In der Schreibweise mit Trennzeichen steht jedes Syntaxelement in einer separaten Zeile. Wenn zwei oder mehr Syntaxelemente stets gemeinsam angegeben (oder nicht angegeben) werden müssen, können sie in derselben Zeile stehen, da sie als ein zusammengesetztes Syntaxelement betrachtet werden können.

Jede Zeile beginnt mit einer Zahl in der Schreibweise mit Trennzeichen, zum Beispiel 3 oder 3.1 oder 3.1.1. Um diese Zahlen korrekt zu hören, müssen Sie sicherstellen, dass das Sprachausgabeprogramm so konfiguriert ist, dass die Interpunktion angesagt wird. Alle Syntaxelemente mit derselben Zahl in der Schreibweise mit Trennzeichen (z. B. alle Syntaxelemente mit der Zahl 3.1) stellen Alternativen dar, die sich gegenseitig ausschließen. Wenn Sie die Zeilen '3.1 USERID' und '3.1 SYSTEMID' hören, wissen Sie, dass die Syntax entweder USERID oder SYSTEMID enthalten kann, nicht jedoch beides.

Die Nummerierung bei der Schreibweise mit Trennzeichen gibt den Grad der Ausgliederung an. Beispiel: Wenn auf das Syntaxelement mit der Zahl 3 in der Schreibweise mit Trennzeichen eine Reihe von Syntaxelementen mit der Zahl 3.1 folgt, sind alle Syntaxelemente mit der Zahl 3.1 dem Syntaxelement mit der Zahl 3 untergeordnet.

Bestimmte Wörter und Symbole werden zusätzlich zu den Zahlen in der Schreibweise mit Trennzeichen verwendet, um weitere Informationen zu den Syntaxelementen anzugeben. In manchen Fällen können diese Wörter und Symbole am Anfang des Elements selbst stehen. Zur einfacheren Identifizierung wird dem Wort oder Symbol ein umgekehrter Schrägstrich (\) vorangestellt, wenn es Teil des Syntaxelements ist. Das Symbol \* (Stern) kann zusätzlich zu einer Zahl in der Schreibweise mit Trennzeichen verwendet werden, um anzugeben, dass das Syntaxelement wiederholt wird. Beispiel: Das Syntaxelement \*FILE mit der Zahl 3 in der Schreibweise mit Trennzeichen erhält das Format 3 \\* FILE. Format 3\* FILE gibt an, dass das Syntaxelement FILE wiederholt wird. Format 3\* \\* FILE gibt an, dass das Syntaxelement \* FILE wiederholt wird.

Zeichen wie beispielsweise Kommas, die bei einer Folge von Syntaxelementen als Trennzeichen verwendet werden, werden in der Syntax unmittelbar vor den Elementen dargestellt, die sie trennen. Diese Zeichen können in derselben Zeile stehen wie das jeweilige Element oder in einer separaten Zeile mit derselben Zahl in der Schreibweise mit Trennzeichen, die auch dem betreffenden Element zugeordnet ist. Die Zeile kann auch ein weiteres Symbol enthalten, das Informationen zu den

Syntaxelementen angibt. So bedeuten z. B. die Zeilen 5.1\*, 5.1 LASTRUN und 5.1 DELETE, dass, wenn Sie mehr als eines der Elemente LASTRUN und DELETE verwenden, diese Elemente durch Kommas voneinander getrennt werden müssen. Wenn kein Trennzeichen angegeben wird, verwendet das System zum Trennen der einzelnen Syntaxelemente ein Leerzeichen.

Wenn einem Syntaxelement das Symbol % vorangestellt ist, gibt dies einen Verweis an, der an anderer Stelle definiert ist. Die Zeichenfolge, die auf das Symbol % folgt, ist der Name eines Syntaxfragments und kein Literal. So gibt die Zeile 2.1 %OP1 beispielsweise einen Verweis auf das separate Syntaxfragment OP1 an.

Die nachfolgend aufgeführten Wörter und Symbole werden zusätzlich zu den Zahlen in der Schreibweise mit Trennzeichen verwendet:

- ? stellt ein optionales Syntaxelement dar. Eine Zahl in der Schreibweise mit Trennzeichen, auf die das Symbol ? (Fragezeichen) folgt, gibt an, dass alle Syntaxelemente mit einer entsprechenden Zahl in der Schreibweise mit Trennzeichen sowie alle untergeordneten Syntaxelemente optional sind. Ist nur ein Syntaxelement mit einer Zahl in der Schreibweise mit Trennzeichen vorhanden, wird das Symbol ? in derselben Zeile angezeigt wie das Syntaxelement (zum Beispiel 5? NOTIFY). Sind mehrere Syntaxelemente mit einer Zahl in der Schreibweise mit Trennzeichen vorhanden, wird das Symbol ? in einer separaten Zeile angezeigt, gefolgt von den optionalen Syntaxelementen. Wenn Sie beispielsweise die Zeilen 5 ?, 5 NOTIFY und 5 UPDATE hören, wissen Sie, dass die Syntaxelemente NOTIFY und UPDATE optional sind; das bedeutet, Sie können eines oder keines dieser Elemente auswählen. Das Symbol ? entspricht einer Umgehungslinie in einem Pfeildiagramm.
- ! stellt ein Standardsyntaxelement dar. Eine Zahl in der Schreibweise mit Trennzeichen, auf die das Symbol ! (Ausrufezeichen) und ein Syntaxelement folgen, gibt an, dass es sich bei diesem Syntaxelement um die Standardoption für alle Syntaxelemente handelt, denen dieselbe Zahl in der Schreibweise mit Trennzeichen zugeordnet ist. Nur für eines der Syntaxelemente, denen dieselbe Zahl in der Schreibweise mit Trennzeichen zugeordnet ist, darf das Symbol ! angegeben werden. Wenn Sie beispielsweise die Zeilen 2? FILE, 2.1! (KEEP) und 2.1 (DELETE) hören, wissen Sie, dass (KEEP) die Standardoption für das Schlüsselwort FILE ist. Wenn Sie in diesem Beispiel das Schlüsselwort FILE verwenden, jedoch keine Option angeben, wird die Standardoption KEEP verwendet. Eine Standardoption ist auch für die nächsthöhere Zahl in der Schreibweise mit Trennzeichen gültig. In diesem Beispiel bedeutet das: Wenn das Schlüsselwort FILE weggelassen wird, wird der Standardwert FILE(KEEP) verwendet. Wenn Sie jedoch die Zeilen 2? FILE, 2.1, 2.1.1! (KEEP) und 2.1.1 (DELETE) hören, gilt die Standardoption KEEP nur für die nächsthöhere Zahl in der Schreibweise mit Trennzeichen, 2.1 (der kein Schlüsselwort zugeordnet ist), nicht jedoch für 2? FILE. Wird das Schlüsselwort FILE weggelassen, wird kein Wert verwendet.
- \* stellt ein Syntaxelement dar, das keinmal, einmal oder mehrmals wiederholt werden kann. Eine Zahl in der Schreibweise mit Trennzeichen, auf die das Symbol \* (Stern) folgt, gibt an, dass dieses Syntaxelement keinmal, einmal oder mehrmals verwendet werden kann, d. h., es ist optional und kann wiederholt werden. Wenn Sie beispielsweise die Zeile 5.1\* Datenbereich hören, wissen Sie, dass Sie einen, mehrere oder keinen Datenbereich angeben können. Hören Sie die Zeilen 3\*, 3 HOST und 3 STATE, wissen Sie, dass Sie HOST, STATE, beide oder keines der Elemente angeben können.

### **Anmerkungen:**

1. Wenn neben einer Zahl in der Schreibweise mit Trennzeichen ein Stern (\*) angezeigt wird und nur ein Element mit dieser Zahl vorhanden ist, können Sie dieses Element mehrmals wiederholen.
  2. Wenn neben einer Zahl in der Schreibweise mit Trennzeichen ein Stern angezeigt wird und diese Zahl mehreren Elementen zugeordnet ist, können Sie mehrere Elemente aus der Liste verwenden, jedes davon jedoch nur einmal. Im vorhergehenden Beispiel könnten Sie HOST STATE angeben, nicht jedoch HOST HOST.
  3. Das Symbol \* entspricht einer zum Ausgangspunkt zurück führenden Linie in einem Pfeildiagramm.
- + stellt ein Syntaxelement dar, das mindestens einmal angegeben werden muss. Eine Zahl in der Schreibweise mit Trennzeichen, auf die das Symbol + (Pluszeichen) folgt, gibt an, dass dieses Syntaxelement mindestens einmal angegeben werden muss und wiederholt werden kann. Wenn Sie beispielsweise die Zeile 6.1+ Datenbereich hören, müssen sie mindestens einen Datenbereich angeben. Wenn Sie die Zeilen 2+, 2 HOST und 2 STATE hören, wissen Sie, dass Sie HOST, STATE oder beides angeben müssen. Wie auch für das Symbol \* gilt hier, dass mit dem Pluszeichen ein bestimmtes Element nur dann wiederholt werden kann, wenn es sich um das einzige Element mit dieser Zahl in der Schreibweise mit Trennzeichen handelt. Das Symbol + entspricht wie das Symbol \* einer zum Ausgangspunkt zurück führenden Linie in einem Pfeildiagramm.

### **Zugehörige Konzepte:**

- „Eingabehilfen“ auf Seite 473

### **Zugehörige Tasks:**

- „Direktaufrufe über die Tastatur: Gemeinsame GUI - Hilfe“

### **Zugehörige Referenzen:**

- „How to read the syntax diagrams“ in *SQL Reference, Volume 2*

---

## **Common Criteria-Zertifizierung von DB2 Universal Database-Produkten**

Für DB2 Universal Database läuft momentan der Bewertungsprozess für die Zertifizierung entsprechend den Richtlinien von Common Criteria Evaluation Assurance Level 4 (EAL4). Weitere Informationen zu Common Criteria finden Sie auf der Common Criteria-Website unter: <http://niap.nist.gov/cc-scheme/>.



---

## Anhang L. Bemerkungen

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Dienstleistungen von IBM verwendet werden können. An Stelle der IBM Produkte, Programme oder Dienstleistungen können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb der Produkte, Programme oder Dienstleistungen in Verbindung mit Fremdprodukten und Fremddienstleistungen liegt beim Kunden, soweit nicht ausdrücklich solche Verbindungen erwähnt sind.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Europe, Director of Licensing, 92066 Paris La Defense Cedex, France.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen bekanntgegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Canada Limited  
Office of the Lab Director  
8200 Warden Avenue  
Markham, Ontario  
L6G 1C7  
CANADA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Handbuch aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt im Rahmen der Allgemeinen Geschäftsbedingungen der IBM, der Internationalen Nutzungsbedingungen der IBM für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer gesteuerten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Garantie, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Informationen über Produkte anderer Hersteller als IBM wurden von den Herstellern dieser Produkte zur Verfügung gestellt, bzw. aus von ihnen veröffentlichten Ankündigungen oder anderen öffentlich zugänglichen Quellen entnommen. IBM hat diese Produkte nicht getestet und übernimmt im Hinblick auf Produkte anderer Hersteller keine Verantwortung für einwandfreie Funktion, Kompatibilität oder andere Ansprüche. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten der IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele der IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogrammes illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden, Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

#### COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Kopien oder Teile der Beispielprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. *Jahr/Jahre angeben*. Alle Rechte vorbehalten.



---

## Marken

Folgende Namen sind in gewissen Ländern Marken der International Business Machines Corporation und wurden in mindestens einem der Dokumente in der DB2 UDB-Dokumentationsbibliothek verwendet:

ACF/VTAM	iSeries
AISPO	LAN Distance
AIX	MVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
IBM System AS/400	NetView
BookManager	OS/390
C Set++	OS/400
C/370	PowerPC
CICS	pSeries
Database 2	QBIC
DataHub	QMF
DataJoiner	RACF
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/400
DB2 Extenders	SQL/DS
DB2 OLAP Server	System/370
DB2 Information Integrator	IBM System /390
DB2 Query Patroller	SystemView
DB2 Universal Database	Tivoli
Distributed Relational Database Architecture	VisualAge
DRDA	VM/ESA
eServer	VSE/ESA
Extended Services	VTAM
FFST	WebExplorer
First Failure Support Technology	WebSphere
IBM	WIN-OS/2
IMS	z/OS
IMS/ESA	zSeries

Folgende Namen sind in gewissen Ländern Marken oder eingetragene Marken anderer Unternehmen und wurden in mindestens einem der Dokumente in der DB2 UDB-Dokumentationsbibliothek verwendet.

Microsoft, Windows, Windows NT und das Windows-Logo sind in gewissen Ländern Marken der Microsoft Corporation.

Intel und Pentium sind in gewissen Ländern Marken der Intel Corporation.

Java und alle auf Java basierenden Marken sind in gewissen Ländern Marken von Sun Microsystems, Inc.

UNIX ist in gewissen Ländern eine eingetragene Marke von The Open Group.

Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken anderer Unternehmen sein.



---

# Index

## Sonderzeichen

\$RAHBUFDIR 384  
\$RAHBUFNAME 384  
\$RAHENV 391

## A

Abfragen  
  umschreiben, gespeicherte Abfrage-  
  tabelle 136  
Active Directory  
  DB2-Objekte 363  
  Erweitern des Verzeichniss-  
  schemas 361  
  Konfigurieren von DB2 344  
  Lightweight Directory Access Protocol  
  (LDAP) 341  
  Sicherheit 360  
  Unterstützung 343  
Aggregatfunktion 126  
Aktualisieren  
  DB2 Information - Unterstüt-  
  zung 461  
Aktualisieren der Daten in einer gespei-  
  cherten Abfragetabelle 208  
Aktualisieren von Sichtinhalten mit Aus-  
  lösern 125  
Aktualisierungen  
  DAS-Konfiguration 67  
  typisierte Tabelle 209  
Aliasnamen  
  Berechtigung 143  
  DB2 für z/OS und OS/390 143  
  erstellen 143  
  verwenden 143  
ALTER, Zugriffsrecht 259  
ALTER COLUMN 189  
ALTER TABLE, Anweisung  
  Hinzufügen eindeutiger Integritäts-  
  bedingungen, Beispiel 193  
  Hinzufügen einer Prüfung auf  
  Integritätsbedingung, Beispiel 196  
  Hinzufügen von Schlüssel, Bei-  
  spiel 195  
  Hinzufügen von Spalten, Bei-  
  spiel 188  
  Löschen einer eindeutigen Integritäts-  
  bedingung, Beispiel 197  
  Löschen einer Prüfbedingung, Bei-  
  spiel 199  
  Löschen von Schlüssel, Beispiel 198  
ALTER TABLESPACE, Anweisung  
  Beispiel 172  
ALTER VIEW, Anweisung  
  Beispiel 216  
Alternative Server  
  Beispiele 338  
Alternativer Server  
  identifizieren 75

Ändern  
  Datenbankkonfiguration 168  
  Datenbankpartitionsgruppe 171  
  IDENTITY-Spalte 193  
  Partitionierungsschlüssel 203  
  Sichten 216  
  Spalten 189  
  strukturierter Typ 208  
  Tabellenattribute 204  
  Tabellenbereiche 171  
Ändern, Integritätsbedingung 193  
Ändern, Tabelle 185  
Ändern der Merkmale einer gespeicher-  
  ten Abfragetabelle 207  
Ändern von Tabellen  
  mit gespeicherten Prozeduren 186  
Angeben  
  Datenbankpartitionsserver (logische  
  Knoten) 390  
Anwendungsprogrammierschnittstellen  
  (API)  
  Aktualisieren von Datenbank-  
  verzeichnissen 82  
Assistenten  
  Leistungskonfiguration 168  
ATTACH, Befehl 7  
Attributdefinitionen  
  Netscape LDAP 363  
audit\_buf\_sz, Konfigurations-  
  parameter 285  
Aufrufen  
  Hilfe für Befehle 471  
  Hilfe für Nachrichten 470  
  Hilfe zu SQL-Anweisungen 471  
Ausdrücke  
  NEXTVAL 111  
  PREVVAL 111  
Auslöser  
  Abhängigkeiten 124  
  Aktualisierungen  
  Aktualisieren von Sicht-  
  inhalten 125  
  erstellen 122  
  löschen 214  
  Vorteile 122  
Authentifizierung  
  Definition 234  
  Domänensicherheit 409  
  ferner Client 240  
  Gruppen 409  
  mit geordneter Domänenliste 410  
Typen  
  CLIENT 234  
  KERBEROS 234  
  KRB\_SERVER\_ENCRYPT 234  
  SERVER 234  
  SERVER\_ENCRYPT 234  
Überlegungen zu partitionierten  
  Datenbanken 240

Automatisch aktualisierte Übersicht-  
  stabellen  
  erstellen 136  
Automatische Clientweiterleitung  
  Beispiele 338  
  Beschreibung 335  
  einrichten 335  
  Einschränkungen 336

## B

Befehle  
  paralleles Ausführen 384  
Befehlszeilenprozessor (CLP)  
  Binden an eine Datenbank 80  
Behälter  
  DMS-Tabellenbereiche  
  Ändern von Behältern 173  
  Hinzufügen von Behältern 172  
  hinzufügen zu SMS-Tabellen-  
  bereichen 177  
Behinderung 473  
Beispiele  
  alternativer Server 338  
  automatische Clientweiterleitung 338  
Benutzer-IDs  
  auswählen 227  
  Namenskonventionen 330  
Benutzerauthentifizierung  
  Windows NT 405  
Benutzerdefinierte Datentypen (UDTs)  
  einzigartige Datentypen  
  erstellen 130  
  erstellen 129  
  löschen 215  
  strukturierte Typen 132  
Benutzerdefinierte Funktionen (UDFs)  
  Datenbankberechtigung zum Erstellen  
  nicht abgeschirmter 255  
  erstellen 126  
  löschen 214  
  Typen 126  
Benutzerdefinierte temporäre Tabellen  
  erstellen 108  
  löschen 213  
Benutzerdefinierter erweiterter Index-  
  typ 155  
Benutzertabellenbereiche 179  
Benutzertabellenbereiche, temporär  
  erstellen 87  
  löschen 182  
Berechtigung  
  gesicherter Client 234  
Berechtigungsnamen  
  Abrufen der Namen mit Berechtigung  
  DBADM 277  
  Abrufen für Informationen über  
  Zugriffsrechte 277  
  Abrufen von Namen mit Tabellenzu-  
  griffsberechtigung 278  
  Abrufen von Zugriffsrechten für 278

Berechtigungsnamen (*Forts.*)  
Erstellen einer Sicht für Informationen  
über Zugriffsrechte 279

Berechtigungsstufen  
Datenbankverwaltung  
(DBADM) 253, 257  
siehe Zugriffsrechte 245  
Systemmonitorberechtigung (SYS-  
MON) 253  
Systempflege (SYSMAINT) 252  
Systemsteuerung (SYSCTRL) 251  
Systemverwaltung (SYSADM) 250  
Widerrufen von DBADM für Benutzer  
mit SYSADM 250  
Widerrufen von DBADM für Benutzer  
mit SYSCTRL 251

Bereich  
hinzufügen 189  
Bereichsclustertabellen  
Beispiele 113  
Zugriffspfadsbestimmung 115  
Bestellen von DB2-Büchern 468  
Bestimmung von Fehlern mit rah 394  
BIND, Befehl  
OWNER, Option 268

BIND, Zugriffsrecht  
Definition 261  
BINDADD, Datenbankberechtigung  
Definition 255

Binden  
Datenbankdienstprogramme 80  
erneutes Binden ungültiger Pake-  
te 265  
Blockorientierte Einheiten 83

## C

Call Level Interface (CLI)  
Binden an eine Datenbank 80  
CATALOG DATABASE, Befehl  
Beispiel 81  
Client  
Kommunikationsfehler 335  
CLIENT, Authentifizierungstyp  
Sicherheit auf Clientebene 234  
Clients  
automatisches Weiterleiten 335  
Clientweiterleitung  
automatisch 335  
Beispiele 338  
Einschränkungen 336  
CONNECT, Datenbankberechtigung 255  
CONTROL, Zugriffsrecht  
Beschreibung 259  
implizites Ausführen 267  
Zugriffsrechte für Pakete 261  
CREATE ALIAS, Anweisung  
Beispiel 143  
CREATE DATABASE, Befehl  
Beispiel 69  
CREATE\_EXTERNAL\_ROUTINE,  
Datenbankberechtigung 255  
CREATE INDEX, Anweisung  
Beispiele 150  
eindeutiger Index 150  
Onlinereorganisation 145, 150  
Zugriff beschränken 150

CREATE\_NOT\_FENCED\_ROUTINE,  
Datenbankberechtigung 255  
CREATE TABLE, Anweisung  
Beispiel 95  
Definieren referenzieller Integritäts-  
bedingungen 101  
Definieren von Prüfungen auf  
Integritätsbedingung 105  
Verwenden mehrerer Tabellen-  
bereiche 119  
CREATE TABLESPACE, Anweisung  
Beispiel 83  
CREATE TRIGGER, Anweisung  
Beispiel 122  
CREATE VIEW, Anweisung  
Ändern von Spaltennamen 133  
Beispiel 133  
CHECK OPTION, Klausel 133  
CREATETAB, Datenbank-  
berechtigung 255  
CURRENT SCHEMA, Sonderregister 8,  
93

## D

DAS (DB2-Verwaltungsserver)  
FFDC (First Failure Data Capture) 67  
Java Virtual Machine, einrichten 56  
Daten  
Ändern der Verteilung 171  
Prüfung  
arbeiten mit, Übersicht 291  
Auswählen von Prüfdaten aus  
Tabellen 298  
Erstellen von Prüfdatendatei-  
en 294  
Erstellen von Tabellen 291  
Laden von Prüfdaten in Tabel-  
len 296  
Sicherheit des Systemkatalogs 279  
Steuerung des Datenbankzugriffs 227  
Überwachen des Zugriffs 273  
Datenbank 3  
ändern 170  
erstellen 69  
Überlegungen vor dem Ändern 163  
Überlegungen zum Erstellen 17  
vor dem Erstellen 3  
Zugriff  
Zugriffsrechte über Pakete bei  
SQL 268  
Datenbank hinzufügen, Assistent 82  
Datenbankadministratorberechtigung  
(DBADM)  
Definition 253  
Datenbankberechtigungen  
BINDADD 255  
CONNECT 255  
CREATE\_EXTERNAL\_ROUTINE 255  
CREATE\_NOT\_FENCED 255  
CREATETAB 255  
Datenbankmanager 255  
erteilen 255  
IMPLICIT\_SCHEMA 255  
LOAD 255  
PUBLIC 255  
QUIESCE\_CONNECT 255

Datenbankberechtigungen (*Forts.*)  
widerrufen 255  
Datenbanken  
Aktivieren von E/A-Parallelität 12  
Ändern der Verteilung von  
Daten 171  
Ändern einer Datenbankpartitions-  
gruppe 171  
Ermöglichen der Datenpartitionie-  
rung 13  
Erstellen in allen Datenbank-  
partitionen 13  
katalogisieren 81  
löschen 170  
Paketabhängigkeiten 221  
Datenbankkonfiguration  
ändern 168  
ändern in mehreren Partitionen 169  
Datenbankkonfigurationsdatei  
erstellen 42  
Datenbankmanager  
Binden von Dienstprogrammen 80  
Index 148  
starten unter UNIX 4  
starten unter Windows 5  
stoppen unter UNIX 15  
stoppen unter Windows 16  
Zugriffssteuerung 263  
Datenbankobjekte  
Eigentumsrecht und Zugriffs-  
rechte 249  
Erstellung und Zugriffsrechte 249  
modifizieren  
Anweisungsabhängigkeiten 221  
Namenskonventionen  
NLS 333  
Unicode 334  
Zugriffssteuerung 263  
Datenbankpartitionen  
ändern 421  
Ändern, Datenbankkonfiguration 169  
Erstellen einer Datenbank auf  
allen 13  
katalogisieren 13, 76  
Datenbankpartitionsgruppen  
ändern 171  
Erstdefinition 70  
erstellen 78  
IBMDEFAULTGROUP, Standard-  
tabelle 120  
Partitionierungsschlüssel, ändern 203  
Überlegungen zu Tabellen 120  
Datenbankpartitionsnummer 39  
Datenbankpartitionsserver  
Absetzen von Befehlen 381  
angeben 390  
Beschreibung 425  
löschen 423  
Windows 419  
Datenbankservers  
alternativ 75  
Datenbankverzeichnisse  
aktualisieren 82  
Datenbankwiederherstellungsprotokoll  
definieren 79  
Datenbankzugriff  
steuern 227

- Datensätze
    - Prüfung 283
  - Datentypen
    - Mehrbytezeichensatz 95
    - Spaltendefinition 95
  - Datenverschlüsselung
    - Beschreibung 273
  - Datenwiederherstellung xii
  - db2\_all, Befehl 381, 382, 383
    - Übersicht 381
  - DB2-Bücher
    - Drucken von PDF-Dateien 468
  - db2\_call\_stack 382
  - DB2 für Windows-Leistungszähler 413
  - DB2 für Windows NT, Szenario
    - Clientauthentifizierung
      - Windows 9x-Client 404
      - Windows NT-Client 403
    - Serverauthentifizierung 402
  - DB2 Information - Unterstützung 450
    - aktualisieren 461
    - aufrufen 459
    - installieren 452, 454, 457
    - verschiedene Sprachen 462
  - db2\_kill 382
  - DB2-Lernprogramme 472
  - DB2-Objekte
    - Namenskonventionen 327
  - DB2-Umgebung
    - automatische Einstellung
      - UNIX 20
    - manuelle Einstellung
      - UNIX 20
  - DB2-Verwaltungsserver (DAS)
    - Aktivieren der Discovery-Funktion 63
    - Aktualisieren der Konfiguration 67
      - aktualisieren unter UNIX 57
    - auflisten 49
    - Benachrichtigung und Kontaktliste, einrichten 55
    - Eigentumsregeln 37
    - einrichten mit einem partitionierten Datenbanksystem 59
      - Beispiel 59
    - entfernen 58
    - erstellen 47
    - Konfiguration 61
    - konfigurieren 50
    - Scheduler-Einrichtung und Konfiguration 50
    - starten und stoppen 48
    - Überlegungen zur Sicherheit 57
    - Übersicht 45
    - Verwenden des Konfigurationsassistenten und der Steuerzentrale 66
  - db2audit 287
  - db2audit.log 283
  - db2dmnbccktr 405, 408
  - db2gncol, Dienstprogramm 200
  - db2icrt, Befehl
    - Erstellen weiterer Exemplare 23
  - db2idrop, Befehl 167
  - db2ilist, Befehl 27
  - DB2INSTANCE, Umgebungsvariable
    - Definieren des Standardexemplars 6
  - db2iupdt, Befehl 164, 166
  - DB2LDAP\_CLIENT\_PROVIDER 342
  - db2ldcfg, Dienstprogramm 347
  - db2nchg, Befehl 421
  - db2ncrt, Befehl 420
  - db2ndrop, Befehl 423
  - db2nlist, Befehl 419
  - db2nodes.cfg, Datei 39
  - db2perfc 417
  - db2perfi 413
  - db2perfr 414
  - db2set, Befehl 29, 33
  - db2start, Befehl 4, 5
  - db2start ADDNODE 420
  - db2stop, Befehl 15, 16
  - DBADM-Berechtigung
    - Abrufen von Namen 277
  - DBCS (Double-Byte Character Set)
    - Namenskonventionen 333
  - DECLARE GLOBAL TEMPORARY TABLE 108
  - DEFAULT, Attribut 95
  - Definieren
    - eindeutige Integritätsbedingungen 100
    - Prüfung auf Integritätsbedingungen in Tabellen 105
    - referenzielle Integritätsbedingungen 101
  - Deklarieren
    - Registrier- und Umgebungsvariablen 33
    - Tabelle als flüchtig 203
  - DELETE, Zugriffsrecht 259
  - DETACH, Befehl
    - Übersicht 7
  - Dienstprogrammoperationen, Integritätsbedingungen 105
  - Dimensionen
    - definieren für eine Tabelle 116
  - Direktaufrufe über Tastatur
    - Unterstützung 473
  - Direkter Plattenzugriff 88
  - Discovery-Einrichtung
    - aktivieren 63
    - Festlegen von Parametern 65
    - Konfiguration 67
    - Verdecken von Serverexemplaren 64
  - DMS-Tabellenbereiche
    - erstellen 83
  - Dokumentation
    - anzeigen 459
  - Domänen
    - Vertrauensstellungen 407
  - Domänencontroller
    - Sicherungs- 405
  - Domänenliste
    - geordnet 410
  - Domänensicherheit
    - Authentifizierung 409
    - DB2 für Windows NT-Unterstützung 411
  - DROP, Anweisung
    - Indizes 220
    - Sichten
      - Beispiele 216
  - DROP, Anweisung (*Forts.*)
    - Tabellen
      - Beispiele 212
      - Tabellenbereiche 179
  - DROP DATABASE, Befehl
    - Beispiel 170
  - Drucken
    - PDF-Dateien 468
  - Dynamisches SQL
    - EXECUTE, Zugriffsrecht für den Datenbankzugriff 268
    - im Cache, als ungültig markiert 220
- ## E
- Ein-/Ausgabeparallelität
    - aktivieren 12
  - Eindeutige Integritätsbedingungen
    - definieren 100
    - hinzufügen 193
    - löschen 197
  - Eingabehilfen
    - Einrichtungen 473
    - Syntaxdiagramme in Schreibweise mit Trennzeichen 475
  - Einheiten, unformatiert 83
  - Einheiten, zeichenorientiert 83
  - Einschränkungen
    - Benennung
      - Windows NT 405
  - Einstellen
    - Schema 93
    - Standardumgebungsprofil für rah 393
  - Eliminieren doppelter Maschineneinträge 390
  - Entwurf implementieren 3
  - Erstellen
    - Aliasnamen 143
    - Auslöser 122
    - benutzerdefinierte Datentypen 129
    - benutzerdefinierte Funktionen 126
    - einzigartige benutzerdefinierte Datentypen 130
    - Exemplare
      - UNIX 24
      - Windows 25
    - Funktionsschablonen 128
    - Funktionszuordnungen 127
    - Hierarchietabellen 118
    - Indexerweiterungen 145
    - Indexspezifikationen 145
    - Indizes
      - Aktivieren von Parallelität 11
      - Übersicht 148
    - LDAP-Benutzer 346
    - Schemata 92
    - Sichten 133
    - Tabellen 95
    - Tabellen mit mehreren Tabellenbereichen 119
    - Tabellenbereiche 83
    - Typenzuordnungen 132
    - typisierte Sichten 136
    - typisierte Tabellen 118
  - Erstellungsaufgaben
    - Berechtigungen 274

- EXECUTE, Zugriffsrecht
  - Datenbankzugriff mit dynamischem SQL 268
  - Datenbankzugriff mit statischem SQL 268
  - Definition 261, 263
- Exemplarbenutzer
  - Einrichten der Umgebung 18
- Exemplare
  - Aktualisieren der Konfiguration
    - UNIX 164
    - Windows 166
  - ändern 163
  - auflisten 27
  - Auflisten der Datenbankpartitions-server 419
  - automatisch starten 28
  - Definition 18
  - Eigner 21
  - entfernen 167
  - erstellen 18
    - UNIX 24
    - Windows 25
  - Erstellen weiterer 23
  - Festlegen des aktuellen 27
  - Gründe zur Verwendung 18
  - hinzufügen 26
  - Hinzufügen von Partitions-servern 420
  - mehrere 6
  - mehrere ausführen 29
  - mehrere unter UNIX 21
  - mehrere unter Windows 22
  - Nachteile 18
  - Partitionsserver
    - ändern 421
    - löschen 423
  - Standard 18
  - starten unter UNIX 4
  - starten unter Windows 5
  - stoppen unter UNIX 15
  - stoppen unter Windows 16
  - Übersicht 6
  - Verzeichnis 18
- Exemplareigner 21
- Exemplarprofilregistrierdatenbank 29
- Explizite Schemaverwendung 8
- EXPORT, Dienstprogramm xi

## F

- Fast Communications Manager (FCM)
  - Service-Eintrag, Syntax 43
- FCM-Kommunikation 43
- Fehlerbehebung
  - Lernprogramme 472
  - Onlineinformationen 472
- Fehlerbestimmung
  - Lernprogramme 472
  - Onlineinformationen 472
- Fern
  - Leistung 416
  - Verwaltung 59
- FFDC (First Failure Data Capture)
  - auf DAS 67
- Firewalls
  - Anwendungsproxy 281

- Firewalls (*Forts.*)
  - Beschreibung 280
  - Circuit-Level 282
  - Screening-Router 281
  - Stateful Multi-Layer Inspection (SMLI) 282
- Fremdschlüssel
  - DROP FOREIGN KEY, Klausel in Anweisung ALTER TABLE 198
  - hinzufügen in einer Tabelle 195
  - IMPORT, Dienstprogramm, Auswirkungen der referenziellen Integrität 105
  - Integritätsbedingungsname 103
  - Load, Dienstprogramm, Auswirkungen der referenziellen Integrität 105
  - Regeln zur Definition 103
  - Zugriffsrechte zum Löschen 198
  - zusammengesetzter Schlüssel 103
- Fremdschlüssel, Integritätsbedingungen
  - referenzielle Integritätsbedingungen 103
  - Regeln zur Definition 103
- Füllen einer typisierten Tabelle 119
- Funktionen
  - DECRYPT 273
  - ENCRYPT 273
  - GETHINT 273
  - Löschen einer benutzerdefinierten 214
- Funktionsaufruf, Selektivität 159
- Funktionsschablonen
  - erstellen 128
- Funktionszugriffsrechte 263
- Funktionszuordnungen
  - erstellen 127

## G

- Gedruckte Bücher, bestellen 468
- Generierte Spalte
  - modifizieren 192
- Generierte Spalten
  - definieren für eine neue Tabelle 107
- Geordnete Domänenliste
  - Authentifizierung mit 410
- Gesicherte Clients
  - CLIENT, Sicherheitsebene 234
- Gespeicherte Abfragetabellen (MQTs)
  - benutzerverwaltet 139, 140
  - Daten aktualisieren 208
  - erstellen 136
  - füllen 140
  - löschen 218
  - Merkmale ändern 207
- Gespeicherte Prozeduren
  - Ändern, Tabelle 186
- Globale Gruppen, Unterstützung
  - Windows 404
- GRANT, Anweisung
  - Beispiel 264
  - implizites Ausführen 267
  - verwenden 264
- Große Objekte (LOB), Datentypen
  - Hinweise zu Spalten 98
- Gruppen
  - auswählen 227

- Gruppen (*Forts.*)
  - Namenskonventionen 330
- Gruppen- und Benutzerauthentifizierung
  - Windows 406
- Gruppeninformationen
  - Zugriffstoken 229

## H

- Hierarchietabellen
  - erstellen 118
  - löschen 212
- Hilfe
  - anzeigen 459, 462
  - für Befehle 471
  - für Nachrichten 470
  - für SQL-Anweisungen 471
- Hilfe für Befehle
  - aufrufen 471
- Hilfe zu SQL-Anweisungen
  - aufrufen 471
- Hinzufügen
  - Bereich 189
  - eindeutige Integritätsbedingungen 193
  - Fremdschlüssel 195
  - Primärschlüssel 194
  - Prüfung auf Integritätsbedingungen in Tabellen 196
- Hinzufügen, Integritätsbedingung 193

## I

- IBM eNetwork Directory, Objektklassen und Attribute 369
- IBMCATGROUP, Datenbankpartitionsgruppe 70
- IBMDEFAULTGROUP, Datenbankpartitionsgruppe 70
- IBMTEMPGROUP, Datenbankpartitionsgruppe 70
- Identitätsspalte
  - modifizieren 192
- Identitätsspalten
  - ändern 205
  - definieren für eine neue Tabelle 109
- IDENTITY, Identitätsspalten 112
  - modifizieren 193
- IMPLICIT\_SCHEMA
  - Berechtigung 92
  - Datenbankberechtigung 255
- Implizite Berechtigung
  - verwalten 267
- Implizite Schemaerstellungsberechtigung (IMPLICIT\_SCHEMA) 257
- Implizite Schemaverwendung 8
- IMPORT, Dienstprogramm xi
- INDEX, Zugriffsrecht 259
- Indexerweiterung 145
- Indexnutzung 157
- Indexpflege
  - Details 156
- Indexschlüssel 145
- Indexsuchen
  - Details 156

- Indextyp
  - eindeutiger Index 145
- Indexzugriffsrecht 262
- Indizes
  - Benutzerdefinierter erweiterter Index-  
typ 155
  - CREATE INDEX, Anweisung 150
  - CREATE UNIQUE INDEX, Anwei-  
sung 150
  - Definition 145
  - DROP INDEX, Anweisung 220
  - eindeutig 150
  - Eindeutigkeit für Primär-  
schlüssel 100
  - erstellen
    - Übersicht 148
  - Leistungstipps 149
  - löschen 220
  - nicht eindeutig 150
  - Nichtprimärindex 220
  - Onlineorganisation 145, 150
  - optimale Anzahl 145
  - Primärindex und benutzerdefinierter  
Index 145
  - Selektivität 159
  - Spezifikationen und Erweiterun-  
gen 145
  - umbenennen 209
  - Zugriffsrechte
    - Beschreibung 262
- Informative Integritätsbedingungen 106
- INSERT, Zugriffsrecht 259
- Installieren
  - DB2 Information - Unterstüt-  
zung 452, 454, 457
- Integritätsbedingung
  - ändern 193
  - hinzufügen 193
  - löschen 197
- Integritätsbedingungen
  - definieren 100
    - eindeutige Integritäts-  
bedingungen 100
    - Fremdschlüssel 103
    - referenzielle Integritäts-  
bedingungen 101
  - informativ 106
  - löschen
    - eindeutige Integritäts-  
bedingungen 197
  - Prüfung auf in Tabellen 105

## J

- Java Virtual Machine, einrichten auf  
DAS 56

## K

- Katalogknoten 13
- Katalogtabellen
  - gespeichert auf dem Katalogknoten  
der Datenbank 13
- Kerberos
  - Authentifizierungstyp 234

- Kerberos (*Forts.*)
  - Sicherheitsprotokolle
    - Authentifizierung durch Fremdan-  
bieter 234
- Knoten 9
- Knotengruppen (Datenbankpartitions-  
gruppen)
  - erstellen 78
- Knotenkonfigurationsdateien
  - erstellen 39
- Knotenverzeichnisse 76
- Known, Discovery 63
- Komprimierung
  - neue Tabellen 98
  - vorhandene Tabellen 185
- Konfigurationsparameter
  - partitionierte Datenbank 13
- Konfigurieren
  - LDAP 345
  - LDAP-Benutzer für Anwendun-  
gen 347
- KRB\_SERVER\_ENCRYPT, Authentifizie-  
rungstyp 234
- Kurznamen
  - Zugriffsrechte
    - indirekt durch Pakete 269

## L

- Laden
  - Daten
    - Aktivieren von Parallelität 11
- LDAP (Lightweight Directory Access Pro-  
tocol)
  - aktivieren 357
  - Aktualisieren von Einträgen 353
  - Aktualisieren von Protokoll-  
informationen 349
  - Beschreibung 341
  - DB2 Connect 359
  - Definieren von Registriervariab-  
len 357
  - Erstellen eines Benutzers 346
  - Erweitern des Verzeichnis-  
schemas 361
  - Herstellen einer fernen Verbin-  
dung 352
  - inaktivieren 358
  - Katalogisieren eines Knoten-  
eintrags 350
  - Konfigurieren von DB2 345
  - Objektclassen und Attribute 369
  - registrieren
    - Datenbanken 351
    - DB2-Server 347
    - Hostdatenbanken 355
  - Sicherheit 359
  - suchen
    - Verzeichnisdomänen 354
    - Verzeichnispartitionen 354
  - Unterstützung 342
  - Verzeichnisservice 77
  - Windows 2000 Active Directory 361
  - Zurücknehmen der Registrierung
    - Datenbanken 353
    - Server 351
- LDAP-Clients
  - weiterleiten 349
- Leistung
  - Anzeigen von Informationen 415
  - Einrichten des Fernzugriffs auf Infor-  
mationen 414
  - gespeicherte Abfragetabellen 136
  - Katalogdaten, Verringern der Zugriffs-  
konkurrenz 13
  - Werte zurücksetzen 417
  - Windows 415
  - Zugriff auf ferne Informationen 416
- Leistungskonfiguration, Assistent
  - aufrufen 161
  - in Konfigurationsadvisor umben-  
annt 168
- Lernprogramme 472
  - Fehlerbestimmung und Fehler-  
behebung 472
- LEVEL2 PCTFREE, Klausel 150
- Lightweight Directory Access Protocol  
(LDAP)
  - aktivieren 357
  - Aktualisieren von Einträgen 353
  - Aktualisieren von Protokoll-  
informationen 349
  - Beschreibung 341
  - DB2 Connect 359
  - Definieren von Registriervariab-  
len 357
  - Erstellen eines Benutzers 346
  - Erweitern des Verzeichnis-  
schemas 361
  - Herstellen einer fernen Verbin-  
dung 352
  - inaktivieren 358
  - Katalogisieren eines Knoten-  
eintrags 350
  - Konfigurieren von DB2 345
  - Objektclassen und Attribute 369
  - registrieren
    - Datenbanken 351
    - DB2-Server 347
    - Hostdatenbanken 355
  - Sicherheit 359
  - suchen
    - Verzeichnisdomänen 354
    - Verzeichnispartitionen 354
  - Unterstützung 342
  - Verzeichnisservice 77
  - Windows 2000 Active Directory 361
  - Zurücknehmen der Registrierung
    - Datenbanken 353
    - Server 351
- Lizenzzentrale
  - Verwalten von Lizenzen 29
- LOAD, Datenbankberechtigung 255
- LOAD, Dienstprogramm xi
- LOAD, Zugriffsrecht 254
- LOB (Large Object), Datentypen
  - Hinweise zu Spalten 98
- LOCK TABLE, Anweisung
  - bei CREATE INDEX 150
- Logische Knoten, siehe Datenbank-  
partitionsserver 390, 425

- Lokal
  - Datenbankverzeichnis
    - anzeigen 76
    - Beschreibung 74
  - Lokales Systemkonto 233
- Löschen
  - Auslöser 214
  - benutzerdefinierte Datentypen 215
  - benutzerdefinierte Funktionen 214
  - benutzerdefinierte Tabellen 213
  - Benutzertabellenbereiche 179
  - Datenbanken 170
  - eindeutige Integritätsbedingungen 197
  - Fremdschlüssel 198
  - gespeicherte Abfragetabellen 218
  - Indexerweiterungen 220
  - Indexspezifikationen 220
  - Indizes 220
  - Primärschlüssel 197
  - Prüfung auf Integritätsbedingungen in Tabellen 199
  - Schemata 182
  - Sequenzen 206
  - Sichten 216
  - Tabellen 212
  - Typenzuordnungen 215
  - Zeilen aus typisierten Tabellen 209
  - Zwischenspeichertabellen 218
- Löschen einer Integritätsbedingung 197

## M

- Maschinenliste, für partitionierte
  - Datenbankumgebung 389
- Mehrere Exemplare 6
  - UNIX 21
  - Windows 22
- Mehrere logische Knoten
  - konfigurieren 425
- Methodenzugriffsrechte 263
- MINPCTUSED, Klausel 150
- Modifizieren
  - Spalten 189
- Modifizieren, Tabelle 185
- MQTs (Materialized Query Tables)
  - benutzerverwaltet 139, 140
  - Daten aktualisieren 208
  - erstellen 136
  - füllen 140
  - löschen 218
  - Merkmale ändern 207

## N

- Nachrichten
  - Prüffunktion 300
- Nachrichtenhilfe
  - aufrufen 470
- Namenskonventionen
  - allgemeine 327
  - begrenzte Bezeichner und Objekt-namen 329
  - Benutzer, Benutzer-IDs und Grup-pen 330
  - DB2-Objekte 327

- Namenskonventionen (*Forts.*)
  - Einschränkungen 327
    - allgemeine 327
    - Windows NT 405
  - Landessprachen 333
  - Objekte und Benutzer 233
  - Objekte zusammengesetzter
    - Datenbanken 331
  - Schemanamen 331
  - Unicode 334
  - Workstations 332
- Netscape
  - LDAP-Verzeichnisunterstützung 363
- Neuausgleichen von Daten auf Behäl-ter 172
- Neuverteilen von Daten
  - über Partitionen 171
- NEXTVAL, Ausdruck 111
- Nichtprimärindizes, löschen 220
- Null
  - Spaltendefinition 95

## O

- Objekte
  - für Leistung unter Windows 415
  - modifizieren
    - Anweisungsabhängigkeiten 221
  - Schemata zum Gruppieren 8
- Online
  - Hilfe, Zugriff 469
  - Reorganisation von Indizes 145

## P

- PAGE SPLIT, Klausel 150
- Pakete
  - Eigner 268
  - löschen 220
  - unbrauchbar 221
  - ungültige
    - abhängig von gelöschten Indi-zes 220
    - nach Hinzufügen eines Fremd-schlüssels 194
  - Widerrufen von Zugriffsrechten 265
  - Zugriffsrechte 261
  - Zugriffsrechte bei SQL 268
- Parallelität
  - aktivieren 9
  - partitionsinterne
    - aktivieren 9
- Partitionen
  - Ändern in Datenbankpartitions-gruppe 171
- Partitionierung von Daten
  - Verwaltung 13
- Partitionierungsschlüssel
  - ändern 203
  - partitionierter Index 145
  - Überlegungen zu Tabellen 120
- Partitionsinterne Parallelität
  - aktivieren 9
- Partitionsübergreifende Abfrage-parallelität
  - aktivieren 9

## Plug-ins

- Architektur 427
- ausführen 428
- Basismenüaktionen 432
- Einstellen der Baumstrukturobjekt-attribute 440
- entwickeln 429
- Hinzufügen von Symbolleisten-knöpfen 430
- kompilieren 428
- Menüelemente, Einschränken der Anzeige 436
- Positionieren von Menü-elementen 433
- Richtlinien 427
- Separatoren für Basismenü-aktionen 434
- Portnummern
  - Bereich
    - definieren 420
- Präfix
  - Sequenzen 387
- PRECOMPILE, Befehl
  - OWNER, Option 268
- PREVVAL 111
- Primärschlüssel
  - DROP PRIMARY KEY, Klausel in Anweisung ALTER TABLE 197
  - erforderliche Zugriffsrechte zum Löschen 197
  - hinzufügen in einer Tabelle 194
  - Integritätsbedingungen 100
  - löschen
    - über Steuerzentrale 197
  - Primärindex 100, 145
  - Zeitpunkt der Erstellung 100
- Profilregistrierdatenbank 29
- Profilregistrierdatenbank auf Exemplar-ebene 29
- Profilregistrierdatenbank auf globaler Ebene 29
- Profilregistrierdatenbank auf Knoten-ebene 29
- Protokolle
  - Prüfung 283
- Protokollieren
  - Einheiten, zeichenorientiert 88
- Prozedurzugriffsrechte 263
- Prüfaktivitäten 283
- Prüffunktion
  - Aktionen 283
  - asynchrones Schreiben von Daten-sätzen 285
  - Beispiele 321
  - Berechtigungen/Zugriffsrechte 283
  - CHECKING-Ereignis, Gründe für Zugriffsgewährung 304
  - CHECKING-Ereignis, Typen von Zugriffsversuchen 305
  - CONTEXT, Ereignistabelle 317
  - CONTEXT, Prüfergebnisse 319
  - Ereignisse 283
  - ERRORTYPE, Parameter 285
  - Fehlerbehandlung 285
  - Funktionsweise 285
  - Nachrichten 300
  - OBJMAINT, Ereignistabelle 307



Prüffunktion (*Forts.*)  
 Parameterbeschreibung 287  
 Prüfdaten in Tabellen  
 Auswählen von Daten aus Tabellen 298  
 Erstellen von Prüfdatendateien 294  
 Erstellen von Tabellen für Prüfdaten 291  
 Laden von Prüfdaten in Tabellen 296  
 Übersicht 291  
 Prüfergebnistabelle 301  
 Satzaufbau 300  
 SECMAINT, Ereignistabelle 309  
 SECMAINT, Zugriffsrechte oder Berechtigungen 310  
 Steuern von Aktivitäten 321  
 synchrones Schreiben von Datensätzen 285  
 Syntax 287  
 SYSADMIN, Ereignistabelle 313  
 SYSADMIN, Prüfergebnisse 315  
 Tipps und Verfahrensweisen 319  
 Überwachen des Datenzugriffs 273  
 VALIDATE, Ereignistabelle 316  
 Verwendungsszenarios 287  
 Prüfliste, Protokoll 283  
 Prüfprotokoll 283  
 Prüfsatz  
 Objekttypen 303  
 Prüfungen auf Integritätsbedingung definieren 105  
 hinzufügen 196  
 löschen 199  
 PUBLIC, Klausel  
 Datenbankberechtigungen, Abbildung 255  
 Pufferpool  
 ändern 183  
 erstellen 72

## Q

Qualifizierte Objektnamen 8  
 QUIESCE\_CONNECT, Datenbankberechtigungen 255

## R

rah, Befehl  
 angeben  
 als Parameter oder Antwort 383  
 Liste von Datenbankpartitionsservern 389  
 Beschreibung 382  
 Definieren des Standardumgebungsprofils 393  
 Einführung 381  
 Fehlerbestimmung 394  
 paralleles Ausführen von Befehlen 384  
 Präfixsequenzen 387  
 RAHCHECKBUF, Umgebungsvariable 384

rah, Befehl (*Forts.*)  
 RAHDOTFILES, Umgebungsvariable 392  
 RAHOSTFILE, Umgebungsvariable 389  
 RAHOSTLIST, Umgebungsvariable 389  
 RAHWAITTIME, Umgebungsvariable 385  
 rekursiv aufgerufen 386  
 steuern 391  
 Übersicht 381  
 Überwachung von Prozessen 385  
 Umgebungsvariablen 391  
 RAHCHECKBUF, Umgebungsvariable 384  
 RAHDOTFILES, Umgebungsvariable 392  
 RAHOSTFILE, Umgebungsvariable 389  
 RAHOSTLIST, Umgebungsvariable 389  
 RAHTREETHRESH, Umgebungsvariable 386  
 RAHWAITTIME, Umgebungsvariable 385  
 REFERENCES, Klausel  
 Lösregeln 104  
 verwenden 104  
 REFERENCES, Zugriffsrecht 259  
 Referenzielle Integritätsbedingungen definieren 101  
 PRIMARY KEY, Klausel in Anweisung CREATE/ALTER TABLE 101  
 REFERENCES, Klausel in Anweisung CREATE/ALTER TABLE 101  
 Registriervariablen  
 Umgebungsvariablen 29  
 Reorganisation, Dienstprogramm  
 Binden an eine Datenbank 80  
 Replikation xi  
 REVOKE, Anweisung  
 Beispiel 265  
 implizites Ausführen 267  
 verwenden 265

## S

Scheduler  
 DB2-Verwaltungsserver (DAS) 50  
 Schemanamen  
 Beschreibung 331  
 Schemata  
 Beschreibung 8  
 einstellen 93  
 erstellen 92  
 löschen 182  
 SESSION 213  
 SEARCH-Discovery  
 in Discovery Parameter für Known-Discovery 63  
 SELECT, Zugriffsrecht 259  
 SELECT-Klausel  
 zum Erstellen einer Sicht 133  
 Selektivität 159  
 Sequenzen  
 ändern 205  
 erstellen 111  
 löschen 206

Sequenzen (*Forts.*)  
 Vergleich mit IDENTITY-Spalten 112  
 Zugriffsrechte 263  
 Server  
 alternativ 75, 335  
 SERVER, Authentifizierungstyp 234  
 SERVER\_ENCRYPT, Authentifizierungstyp 234  
 SET ENCRYPTION PASSWORD, Anweisung 273  
 Sicherheit  
 CLIENT, Ebene 234  
 planen 227  
 UNIX-Aspekte 233  
 Windows NT  
 Benutzer 232  
 Beschreibung 401  
 Services 407  
 Unterstützung der Domänensicherheit 411  
 Sichern von Daten xii  
 Sicherungsdomänencontroller  
 Installieren von DB2 408  
 Konfigurieren von DB2 405  
 Sichten  
 ändern 216  
 Auslöser zum Aktualisieren 125  
 Auswirkungen des Löschens auf Systemkataloge 216  
 Datenintegrität 133  
 Datensicherheit 133  
 Einschränkungen 216  
 Entfernen von Zeilen 191  
 erstellen 133  
 für Informationen über Zugriffsrechte 279  
 löschen 216  
 Spaltenzugriff 270  
 unbrauchbar 218  
 wiederherstellen, unbrauchbare 218  
 Zugriff auf Zeilen 270  
 Zugriffsrechte, Beispiele 270  
 Zugriffssteuerung für Tabelle 270  
 SIGTTIN, Nachricht 383  
 Skalarfunktionen  
 erstellen 126  
 SMS (vom System verwalteter Bereich)  
 Tabellenbereiche  
 erstellen 83  
 Hinzufügen von Behältern 177  
 Spalten  
 definieren 95  
 Definition  
 modifizieren 189  
 Spalten-UDFs 126  
 Speicherplatzkomprimierung  
 neue Tabellen 98  
 Tabellen 97  
 vorhandene Tabellen 185  
 SQL (Structured Query Language)  
 Schlüsselwörter 329  
 SQL-Anweisungen  
 unbrauchbar 221  
 Starten  
 DB2  
 UNIX 4  
 Windows 5

- Statisches SQL
    - EXECUTE, Zugriffsrecht für den Datenbankzugriff 268
  - stdin 383
  - Steuern des Befehls rah 391
  - Steuerzentrale
    - Erweiterungen
      - Ändern eines Objekts 445
      - Erstellen von Untermenüs 435
      - Hinzufügen der Aktion 'Entfernen' 443
      - Hinzufügen eines Beispielobjekts 438
      - Hinzufügen eines Objekts 442
      - Hinzufügen eines Ordners 436
      - Inaktivieren der Möglichkeit zum Ändern von Objekten 446
      - Inaktivieren von Konfigurationsfunktionen 446
      - Konfigurationsdialoge, Inaktivieren von Standardknöpfen 447
      - Plug-in-Architektur 427
      - Positionieren des Menüelements 433
      - Richtlinien für Plug-in-Entwickler 427
      - Schreiben von Plug-ins 429
  - Stoppen
    - DB2
      - UNIX 15
      - Windows 16
  - Stripesets 172
  - Strukturierte Typen
    - ändern 208
  - Suchen
    - DB2-Dokumentation 450
  - SWITCH ONLINE, Klausel 179
  - Synonyme
    - DB2 für OS/390 oder z/Series 143
  - Syntaxdiagramme in Schreibweise mit Trennzeichen 475
  - SYSADM, Berechtigung
    - Beschreibung 250
    - Zugriffsrechte 250
  - SYSCAT, Katalogsichten
    - für Sicherheitsaspekte 276
  - SYSCATSPACE, Tabellenbereiche 71
  - Systemdatenbankverzeichnis
    - anzeigen 76
    - Übersicht 75
  - Systemkataloge
    - abrufen
      - Berechtigungsnamen mit Zugriffsrechten 277
      - Namen erteilte Zugriffsrechte 278
      - Namen mit Berechtigung DBADM 277
      - Namen mit Tabellenzugriffsberechtigung 278
    - Liste der Zugriffsrechte 276
    - löschen
      - Auswirkungen auf Sichten 216
      - Tabellen 212
    - Sicherheit 279
  - Systemkatalogtabellen
    - Beschreibung 73
  - Systemmonitor
    - Windows 413
  - Systemmonitorberechtigung (SYS-MON) 253
  - Systempflegeberechtigung (SYS-MAINT) 252
  - Systemsteuerungsberechtigung (SYSC-TRL) 251
  - Systemtabellenbereiche, temporär 86
  - Szenarios
    - Definieren einer Indexerweiterung 159
- ## T
- Tabelle
    - ändern 185
  - Tabellen
    - Abrufen von Namen mit Zugriff 278
    - ALTER TABLE, Anweisung 188
    - ändern
      - Attribute 204
      - Partitionierungsschlüssel 203
    - ändern mit gespeicherten Prozeduren 186
    - Benennung 95
    - CREATE TABLE, Anweisung 95
    - definieren
      - Dimensionen 116
      - eindeutige Integritätsbedingungen 100
      - Prüfungen auf Integritätsbedingung 105
      - referenzielle Integritätsbedingungen 101
    - entfernen
      - Zeilen 191
    - erstellen 95
      - in partitionierten Datenbanken 120
    - flüchtig 203
    - generierte Spalten 200
    - Generierte Spalten 107
    - hinzufügen
      - Spalten, neu 188
    - Hinzufügen referenzieller Integritätsbedingungen 194, 195
    - Identitätsspalten 109
    - löschen 212
    - Tipps zum Hinzufügen von Integritätsbedingungen 194, 195
    - umbenennen 209
    - Widerrufen von Zugriffsrechten 265
  - Tabellen, benutzerdefinierte Funktionen (UDFs)
    - Beschreibung 126
  - Tabellenbereiche
    - Aktivieren von E/A-Parallelität 12
    - ändern 171
    - Behälter
      - Datei, Beispiel 83
      - Dateisystem, Beispiel 83
      - erweitern 173
    - Behältergröße anpassen 173
    - Benutzer, temporär 87
    - Einheitenbehälter, Beispiel 83
    - erste 71
  - Tabellenbereiche (*Forts.*)
    - erstellen
      - Beschreibung 83
      - in Datenbankpartitionsgruppen 88
    - hinzufügen
      - Behälter 172
    - löschen
      - Benutzer 179
      - Benutzer, temporär 182
      - System, temporär 180
    - SWITCH-Status 179
    - System, temporär 86
    - Trennen von Datentypen, Beispiel 119
    - umbenennen 178
    - Zugriffsrechte 259
  - Tabellenobjekte
    - ändern 185
    - erstellen 95
  - Temporäre Tabellen
    - benutzerdefiniert 108
    - Löschen einer benutzerdefinierten 213
  - TEMPSPACE1, Tabellenbereich 71
  - Tools
    - Katalogdatenbank 50
  - Typenzuordnung
    - erstellen 132
    - löschen 215
  - Typisierte Sichten
    - erstellen
      - CREATE VIEW, Anweisung 136
  - Typisierte Tabellen
    - Aktualisieren von Zeilen 209
    - erstellen 118
    - füllen 119
    - Löschen von Zeilen 209
- ## U
- Übersichtstabellen
    - wiederherstellen, unbrauchbare 219
  - Überwachung
    - rah-Prozesse 385
  - Umbenennen
    - Indizes 209
    - Tabellen 209
    - Tabellenbereiche 178
  - Umgebung mit partitionierten Datenbanken
    - Angaben einer Maschinenliste 389
    - doppelte Maschineneinträge eliminieren 390
  - Umgebungsvariablen
    - einstellen
      - UNIX 37
      - Windows 35
    - Profilregistrierdatenbank 29
    - rah 391
    - RAHDOTFILES 392
  - Unformatierte Ein-/Ausgabe
    - angeben 88
    - einrichten unter Linux 90
  - Unicode (UCS-2)
    - Bezeichner 334
    - Namenskonventionen 334

- Untermenüs
  - erstellen 435
- UPDATE, Zugriffsrecht 259
- USAGE, Zugriffsrecht 263
- USERSPACE1, Tabellenbereich 71

## V

- VARCHAR, Datentyp
  - in Tabellenspalten 189
- Verschlüsselung von Daten 273
- Versetzen von Daten xi
- Vertrauensstellungen 407
- Verwaltungsserver 45
- Verzeichnisse
  - aktualisieren 82
  - lokales Datenbankverzeichnis 74
  - Systemdatenbankverzeichnis 75
- Verzeichnisunterstützung
  - Netscape LDAP 363

## W

- Weiterleiten von Clients 335
  - LDAP 349
- Wiederherstellen
  - Datenbanken, Aktivieren der E/A-Parallelität 12
  - Tabellenbereiche, Aktivieren der E/A-Parallelität 12
- Wiederherstellung
  - Sichten, unbrauchbar 218
  - Übersichtstabellen, unbrauchbar 219
  - Zuordnen eines Protokolls bei der Datenbankerstellung 79
- Windows
  - Active Directory, DB2-Objekte
    - konfigurieren unter Windows NT 363
  - Active Directory, Objektklassen und Attribute
    - konfigurieren unter Windows NT 369
  - Erweitern des Verzeichnisschemas
    - Windows 2000 361
  - Systemmonitor 413
- Windows-Benutzergruppe
  - Zugriffstoken 229
- Windows Management Instrumentation (WMI)
  - Beschreibung 397
  - DB2 UDB-Integration 398
- Windows-Unterstützung
  - lokales Systemkonto (LSA) 233
- Workstations
  - (nname), Namenskonventionen 332

## Z

- Zeichenfolgen
  - Datentyp 95
- Zeichenorientierte serielle Einheiten 83
- Zugriffsrechte
  - abrufen
    - Berechtigungsnamen 277
    - für Namen 278

- Zugriffsrechte (*Forts.*)
  - ALTER 259
  - Aufgaben und erforderliche Berechtigungen 274
  - Beschreibung 245
  - CONTROL 259
  - DELETE 259
  - Eigentumszugriffsrecht (CONTROL) 245
  - Erstellen einer Sicht zu Informationszwecken 279
  - EXECUTE 263
  - GRANT, Anweisung 264
  - Hierarchie 245
  - implizite für Pakete 245
  - INDEX
    - Beschreibung 259, 262
  - indirekt 269
  - individuelle 245
  - INSERT 259
  - Objekteigentumsrecht 249
    - Paket
      - erstellen 261
  - REFERENCES 259
  - REVOKE, Anweisung 265
  - Schema 257
  - SELECT 259
  - Sicht 259
  - Systemkatalogliste 276
  - Tabelle 259
  - Tabellenbereich 259
  - UPDATE 259
  - USAGE 263
- Zugriffssteuerung
  - Authentifizierung 234
  - Datenbankmanager 263
  - Datenbankobjekte 263
  - Sicht auf Tabelle 270
- Zugriffstoken 229
- Zuordnung von Dateien mit freien Bereichen 98
- Zusammengeschlossene Datenbanken
  - Funktionsschablone, erstellen 128
  - Funktionszuordnung, erstellen 127
  - Indexspezifikation, erstellen 145
  - Objektnamenskonventionen 331
  - Typenzuordnung, erstellen 132
- Zwischenspeichertabellen
  - erstellen 141
  - löschen 218



---

## Kontaktaufnahme mit IBM

Telefonische Unterstützung erhalten Sie über folgende Nummern:

- Unter 0180 3 313233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0190 7 72243 erreichen Sie die DB2 Helpline, wo Sie Antworten zu DB2-spezifischen Problemen erhalten.

Informationen zur nächsten IBM Niederlassung in Ihrem Land oder Ihrer Region finden Sie im IBM Verzeichnis für weltweite Kontakte, das Sie im Web unter <http://www.ibm.com/planetwide> abrufen können.

---

## Produktinformationen

Informationen zu DB2 Universal Database-Produkten erhalten Sie telefonisch oder im World Wide Web unter <http://www.ibm.com/software/data/db2/udb>.

Diese Site enthält die neuesten Informationen zur technischen Bibliothek, zum Bestellen von Büchern, zu Produktdownloads, Newsgroups, FixPaks, Neuerungen und Links auf verfügbare Webressourcen.

Telefonische Unterstützung erhalten Sie über folgende Nummern:

- Unter 0180 3 313233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0180 5 5090 können Sie Handbücher telefonisch bestellen.

Informationen dazu, wie Sie sich mit IBM in Verbindung setzen können, finden Sie auf der globalen IBM Internet-Seite unter folgender Adresse:  
[www.ibm.com/planetwide](http://www.ibm.com/planetwide)







SC12-3059-01

