

IBM DB2 Universal Database



Systemverwaltung: Konzept

Version 7

IBM DB2 Universal Database



Systemverwaltung: Konzept

Version 7

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen unter „Anhang F. Bemerkungen“ auf Seite 479 gelesen werden.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM DB2 Universal Database, Administration Guide Planning,
IBM Form SC09-2946-00,

herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2000

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 NLS Center
Kst. 2877
April 2000

Inhaltsverzeichnis

Zu diesem Handbuch	ix
Zielgruppe	x
Aufbau dieses Handbuchs	x
Kurzübersicht über die anderen Bände des Handbuchs zur Systemverwaltung	xii
Systemverwaltung: Implementierung	xii
Systemverwaltung: Optimierung	xiii

Teil 1. DB2 Universal Database. 1

Kapitel 1. Verwalten von DB2 Universal Database	3
--	----------

Teil 2. Datenbankkonzepte. 7

Kapitel 2. Allgemeine Konzepte relationaler Datenbanken.	9
Übersicht über Datenbankobjekte	9
Exemplare	10
Datenbanken.	11
Knotengruppen	11
Tabellen	11
Sichten.	12
Indizes.	12
Schemata	13
Systemkatalogtabellen.	14
Übersicht über Wiederherstellungsobjekte	14
Protokolldateien zur Wiederherstellung	15
Datei des Wiederherstellungsprotokolls	15
Übersicht über Speicherobjekte.	16
Tabellenbereiche	16
Behälter	20
Pufferpool	21
Übersicht über Systemobjekte	22
Konfigurationsparameter.	22
Geschäftsregeln für Daten	24
Wiederherstellen einer Datenbank.	28
Überblick über die Wiederherstellung	28
Faktoren mit Auswirkung auf die Wiederherstellung	35
Überlegungen zur Wiederherstellung nach Schäden	53
Begrenzen der Auswirkungen von Datenträgerfehlern	54

Begrenzen der Auswirkungen von Transaktionsfehlern	57
Synchronisation der Systemuhren in einem System mit partitionierten Datenbanken.	57
Reorganisieren von Tabellen in einer Datenbank	59
Übersicht über die Sicherheit unter DB2.	59
Authentifizierung	59
Berechtigung.	61
Übersicht über Authentifizierung und Berechtigung in einer zusammengesetzten Datenbank	62

Kapitel 3. Systeme zusammengesetzter Datenbanken	65
Einrichten eines Systems zusammengesetzter Datenbanken.	68

Kapitel 4. Parallele Datenbanksysteme	71
Knotengruppen und Datenpartitionierung	72
Arten der Parallelität	73
Ein-/Ausgabeparallelität	74
Abfrageparallelität	74
Dienstprogrammparallelität	77
Hardwareumgebungen	78
Einzelpartition auf einer Einzelprozessormaschine	78
Einzelpartition auf Mehrprozessormaschine	79
Konfigurationen mit mehreren Partitionen	81
Zusammenfassung der am besten geeigneten Arten von Parallelität für jede Hardwareumgebung	85

Kapitel 5. Data Warehouses	87
Was ist ein Data Warehouse?	87
Themenbereiche.	88
Warehouse-Quellen	88
Warehouse-Ziele	88
Warehouse-Agenten und Agenten-Sites	88
Schritte und Prozesse	89
Aufgaben bei der Warehouse-Erstellung.	91

Kapitel 6. Informationen zu Spatial Extender	93
Der Zweck von Spatial Extender	93

Daten zur Darstellung geografischer Merkmale	94
Wie Daten geografische Merkmale darstellen	94
Die Eigenschaften räumlicher Daten	96
Möglichkeiten zur Erstellung räumlicher Daten	96

Teil 3. Datenbankentwurf 99

Kapitel 7. Entwerfen des logischen Datenbankaufbaus 101

Festlegen der in der Datenbank aufzuzeichnenden Daten	101
Definieren von Tabellen für jede Art von Relation	103
Eins-zu-viele- und Viele-zu-eins-Beziehungen	103
Viele-zu-viele-Beziehungen.	104
Eins-zu-eins-Beziehungen	105
Definieren der Spalten für alle Tabellen	106
Definieren einer oder mehrerer Spalten als Primärschlüssel	108
Identifizieren möglicher Schlüsselspalten	110
Definieren von IDENTITY-Spalten	111
Sicherstellen, daß gleiche Werte die gleiche Entität darstellen	112
Normalisieren der Tabellen.	113
Erste Normalform.	113
Zweite Normalform	114
Dritte Normalform	116
Vierte Normalform	117
Planen der Integritätsbedingungen	118
Eindeutige Integritätsbedingungen	119
Referentielle Integrität	120
Prüfungen auf Integritätsbedingungen in Tabellen	126
Auslöser	127
Weitere Überlegungen zum Datenbankentwurf	127

Kapitel 8. Entwerfen der physischen Datenbank 131

Datenbankverzeichnisse.	131
Datenbankdateien.	132
Ermitteln des Platzbedarfs für Tabellen	134
Systemkatalogtabellen	135
Benutzertabellendaten	135
Langfelddaten	138
Daten großer Objekte (LOB-Daten)	138

Indexbereich	139
Zusätzlicher Platzbedarf	142
Speicherbereich für die Protokolldatei	142
Temporärer Arbeitsbereich	144
Entwerfen von Knotengruppen	144
Überlegungen zum Aufbau von Knotengruppen	146
Entwerfen und Auswählen von Tabellenbereichen	153
SMS-Tabellenbereich	157
DMS-Tabellenbereich.	161
Überlegungen zum Tabellenbereichs-entwurf	164
Überlegungen zum Entwerfen einer zusammengesetzten Datenbank.	178

Kapitel 9. Entwerfen verteilter Datenbanken 179

Verwenden einer einzelnen Datenbank in einer Transaktion	180
Verwenden mehrerer Datenbanken in einer Transaktion	181
Aktualisieren einer einzelnen Datenbank	181
Aktualisieren mehrerer Datenbanken	183
Weitere Überlegungen zur Konfiguration	187
Host- oder IBM AS/400-Anwendungen, die auf einen LAN-basierten DB2 Universal Database-Server zur Aktualisierung auf mehreren Systemen zugreifen	188
Prozeß der zweiphasigen Festschreibung	190
Beheben von Problemen bei der zweiphasigen Festschreibung	193
Resynchronisieren unbestätigter Transaktionen bei AUTORESTART=OFF	195

Kapitel 10. Entwerfen für Transaktionsmanager 197

X/Open-Modell für die verteilte Transaktionsverarbeitung	198
Anwendungsprogramm (AP)	198
Transaktionsmanager (TM).	200
Ressourcenmanager (RM)	201
Einrichten einer Datenbank als Ressourcenmanager	203
Verwenden der XA-Zeichenfolgen zum Öffnen und Schließen	203
Neues Format der XA-Zeichenfolge zum Öffnen für DB2 Version 7	203
Werte für TPM und TP_MON_NAME	206

Format der XA-Zeichenfolge zum Öffnen für frühere Versionen von DB2	209
Aktualisieren von Host- oder AS/400-Datenbank-Servern	209
Überlegungen zu Datenbankverbindungen	210
Treffen einer heuristischen Entscheidung	210
Überlegungen zur Sicherheit	213
Überlegungen zur Konfiguration.	214
Unterstützte XA-Funktion	216
Bestimmung von XA-Schnittstellenfehlern	218
Konfigurieren des XA-Transaktionsmanagers zur Verwendung von DB2 UDB	219
Konfigurieren von IBM TXSeries CICS	219
Konfigurieren von IBM TXSeries Encina	219
Konfigurieren von BEA Tuxedo	222
Konfigurieren von Microsoft Transaction Server.	224

Kapitel 11. Entwerfen für hohe Verfügbarkeit	233
Bereitschaftsmodus (Hot Standby)	234
Beispiele	235
Gegenseitige Übernahme	237
Beispiele	238
Wiederherstellen der Verbindung nach einer Funktionsübernahme.	240
Zusätzliche HACMP-Informationen.	241

Teil 4. Hohe Verfügbarkeit 243

Kapitel 12. High Availability Cluster Multi-Processing, Enhanced Scalability (HACMP ES) für AIX	245
Clusterkonfiguration.	246
Konfigurieren einer DB2-Datenbankpartition	251
Beispiel für eine Konfiguration im Bereitschaftsmodus (Hot Standby)	253
Beispiel für eine Konfiguration für gegenseitige Übernahme (Mutual Takeover)	253
Konfiguration eines NFS-Server-Knotens	254
Beispiel für eine NFS-Server-Übernahmekonfiguration	255
Überlegungen zur Konfiguration des SP-Switch	256
Beispiele für DB2-HACMP-Konfigurationen	257
Empfehlungen für den DB2-HACMP-Systemstart	266

HACMP ES-Ereignisüberwachung und benutzerdefinierte Ereignisse	267
HACMP ES-Prozedurdateien	271
Operationen der DB2-Wiederherstellungsprozeduren mit HACMP ES.	274
Andere Prozedurdienstprogramme	276
Überwachen von HACMP-Clustern.	277
Installation von SP HACMP ES unter DB2	279
Neuinstallation von SP HACMP ES unter DB2	279
Migration von SP HACMP ES für DB2	281
SP HACMP ES-Arbeitsblätter für DB2	283

Kapitel 13. Hohe Verfügbarkeit in der Windows NT-Umgebung 293

Konfigurationen für Funktionsübernahme	294
Konfiguration für Bereitschaftsmodus	295
Konfiguration für gegenseitige Übernahme	295
Verwenden des Dienstprogramms DB2MSCS	296
Angabe der Datei DB2MSCS.CFG	297
Einrichten der Funktionsübernahme für ein Datenbanksystem mit einer Partition	302
Einrichten einer Konfiguration zur gegenseitigen Übernahme für zwei Datenbanksysteme mit jeweils einer Partition	302
Einrichten mehrerer MSCS-Cluster für ein partitioniertes Datenbanksystem	303
Pflegen des MSCS-Systems.	304
Überlegungen zur Funktionsrückübertragung	305
Registrieren der Datenbanklaufwerkzuordnung für Konfigurationen zur gegenseitigen Übernahme in Umgebungen mit partitionierten Datenbanken	306
Ausgleichen der Datenbanklaufwerkzuordnung	308
Beispiel - Einrichten zweier Exemplare mit einer Partition für gegenseitige Übernahme	309
Vorbereitung	310
Ausführen des Dienstprogramms DB2MSCS	310
Beispiel - Einrichten eines in vier Knoten partitionierten Datenbanksystems zur gegenseitigen Übernahme	311
Vorbereitung	313
Ausführen des Dienstprogramms DB2MSCS	314

Registrieren der Datenbanklaufwerkzu- ordnung für ClusterA	315
Registrieren der Datenbanklaufwerkzu- ordnung für ClusterB	315
Verwalten von DB2 in einer MSCS- Umgebung	315
Starten und Stoppen von DB2-Ressourcen	315
Ausführen von Prozeduren	316
Überlegungen zu Datenbanken	321
Benutzer- und Gruppenunterstützung	321
Überlegungen zur Kommunikation	322
Überlegungen zur Systemzeit	323
Überlegungen zu Verwaltungs-Server und Steuerzentrale in Umgebungen mit parti- tionierten Datenbanken	323
Einschränkungen und Bedingungen.	325
Kapitel 14. DB2 und hohe Verfügbarkeit unter Sun Cluster 2.2	327
Hohe Verfügbarkeit	327
Fehlertoleranz und fortlaufende Verfüg- barkeit	330
Sun Cluster 2.2.	330
Unterstützte Systeme	330
Agenten	331
Logische Hosts.	332
Logische Netzschnittstellen	333
Plattengruppen und Dateisysteme	334
Steuermethoden	336
Konfiguration von Platten und Datei- systemen.	337
HA-NFS	338
Die Dienstprogramme cconsole und ctelnet	338
Campus-Clustering und kontinentale Clustering	338
Häufige Probleme.	339
Überlegungen zu DB2	339
Anwendungen, die eine Verbindung zu einem HA-Exemplar herstellen	339
Plattenlayout für EE- und EEE-Exemplare Ausgangsverzeichnis für EE- und EEE- Exemplare	341
Logische Hosts und DB2 UDB EEE	343
Speicherposition und Optionen der DB2- Installation	344
Konfigurationsparameter für die Daten- bank und den Datenbankmanager	345
Wiederherstellung nach einem Systemab- sturz	346

Hohe Verfügbarkeit durch Datenreplika- tion	346
Der DB2-Agent für hohe Verfügbarkeit.	346
Registrieren des Service hadb2	346
Die Datei hadb2tab	347
Steuermethoden	348
Benutzerprozeduren	350
Weitere Überlegungen	352
Fehlermonitor	352
Überlegungen zu EEE	353
Die Datei HA.config	355
Ausführen von DB2-Befehlen durch Steuermethoden	357
Installation und Konfiguration	357
Allgemeine Installationsschritte	357
Installation und Konfiguration von DB2 UDB Enterprise Edition.	358
Installation und Konfiguration von DB2 UDB Enterprise - Extended Edition	358
Der Befehl hadb2_setup.	358
Dauer der Funktionsübernahme	363
Fehlerbehebung	365

Teil 5. Anhänge und Schlußteil 373

Anhang A. Verwenden der DB2-Bibliothek 375	375
PDF-Dateien und gedruckte Bücher für DB2	375
Informationen zu DB2	375
Drucken der PDF-Handbücher	388
Bestellen der gedruckten Handbücher	389
DB2-Online-Dokumentation	391
Zugreifen auf die Online-Hilfefunktion	391
Anzeigen von Online-Informationen	393
Verwenden der DB2-Assistenten	395
Einrichten eines Dokument-Servers	397
Suchen nach Online-Informationen	398

Anhang B. Namenskonventionen	399
Datenbanknamen	399
Namen und Aliasnamen für Datenbanken	400
Benutzer-IDs und Kennwörter	400
Schemennamen	401
Gruppen- und Benutzernamen	402
Objektnamen	402
Objektnamen zusammengeschlüssener Datenbanken	404
Beibehaltung von Werten in Groß- /Kleinschreibung in einem System zusammengeschlüssener Datenbanken	405

Anhang C. Planen der Datenbankmigration	407
Überlegungen zur Migration	407
Migrationseinschränkungen	408
Sicherheit und Berechtigungen	408
Speicherbedarf	408
Durch Release-Wechsel bedingte Inkompatibilitäten	409
Umstellen einer Datenbank	409
Anhang D. Inkompatibilitäten zwischen Releases	413
Geplante Inkompatibilitäten von DB2 Universal Database	414
Schreibgeschützte Sichten in einer künftigen Version von DB2 Universal Database	414
PK_COLNAMES und FK_COLNAMES in einer künftigen Version von DB2 Universal Database	414
COLNAMES in einer künftigen Version von DB2 Universal Database nicht mehr verfügbar	415
Inkompatibilitäten in DB2 Universal Database Version 7	415
Anwendungsprogrammierung	415
SQL	418
Dienstprogramme und Tools	419
Konnektivität und Koexistenz	420
Inkompatibilitäten in DB2 Universal Database Version 6	420
Systemkatalogsichten	420
Anwendungsprogrammierung	427
SQL	432

Datenbanksicherheit und Optimierung	433
Dienstprogramme und Tools	435
Konnektivität und Koexistenz	436
Konfigurationsparameter	436

Anhang E. Unterstützung von Landes- sprachen	437
Unterstützung von Landescodes und Codepages	437
Ableiten der Werte von Codepages	452
Zeichensätze	453
Zeichensatz für Kennungen	453
Codieren von SQL-Anweisungen	454
CCSID-Unterstützung für bidirektionale Zeichen	454
Sortierfolge	459
Werte für Datum und Uhrzeit	462
Unicode-/UCS-2- und UTF-8-Unterstützung in DB2 UDB	468
Einführung	468
UCS-2-/UTF-8-Implementierung in DB2 UDB	470

Anhang F. Bemerkungen	479
Neue deutsche Rechtschreibung	482
Änderungen in der IBM Terminologie	482
Marken	483

Index	485
--------------	------------

Kontaktaufnahme mit IBM	493
Produktinformationen	493

Zu diesem Handbuch

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

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

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

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

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

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

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

Zielgruppe

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

Aufbau dieses Handbuchs

Das vorliegende Handbuch enthält Informationen zu folgenden Hauptthemen:

DB2 Universal Database

- Kapitel 1. Verwalten von DB2 Universal Database, enthält eine Einführung sowie eine Übersicht zu DB2 Universal Database.

Datenbankkonzepte

- Kapitel 2. Allgemeine Konzepte relationaler Datenbanken, enthält eine Übersicht über Datenbankobjekte, einschließlich Wiederherstellungsobjekte, Speicherobjekte und Systemobjekte.
- Kapitel 3. Systeme zusammengesetzter Datenbanken, behandelt Systeme zusammengesetzter Datenbanken, d. h. Datenbankverwaltungssysteme (DBMS), die Anwendungen und Benutzer unterstützen, die SQL-Anweisungen übergeben, die auf zwei oder mehr DBMSs oder Datenbanken in einer einzigen Anweisung verweisen.

- Kapitel 4. Parallele Datenbanksysteme, enthält eine Einführung in die Arten von Parallelität, die mit Hilfe von DB2 implementiert werden können.
- Kapitel 5. Data Warehouses, enthält eine Übersicht über den Einsatz von Data Warehouses und Data Warehouse-Funktionen.
- Kapitel 6. Informationen zu Spatial Extender, stellt Spatial Extender vor und enthält Erläuterungen zum Zweck des Produkts sowie zu den Daten, die von ihm verarbeitet werden.

Datenbankentwurf

- Kapitel 7. Entwerfen des logischen Datenbankaufbaus, behandelt die Konzepte und Richtlinien zum logischen Entwurf einer Datenbank.
- Kapitel 8. Entwerfen der physischen Datenbank, behandelt die Richtlinien zum physischen Entwurf einer Datenbank und enthält Überlegungen im Hinblick auf die Datenspeicherung.
- Kapitel 9. Entwerfen verteilter Datenbanken, beschreibt die Möglichkeit des Zugriffs auf mehrere Datenbanken in einer einzigen Transaktion.
- Kapitel 10. Entwerfen für Transaktionsmanager, behandelt die Verwendung von Datenbanken in einer Umgebung für verteilte Transaktionsverarbeitung wie CICS.
- Kapitel 11. Entwerfen für hohe Verfügbarkeit, bietet eine Übersicht über die von DB2 bereitgestellte Unterstützung von Funktionsübernahmen zur Implementierung hoher Verfügbarkeit.

Systeme mit hoher Verfügbarkeit

- Kapitel 12. High Availability Cluster Multi-Processing, Enhanced Scalability (HACMP ES) für AIX, behandelt die DB2-Unterstützung für Fehlerbehebung durch Funktionsübernahme unter AIX.
- Kapitel 13. Hohe Verfügbarkeit in der Windows NT-Umgebung, behandelt die DB2-Unterstützung für Fehlerbehebung durch Funktionsübernahme unter Windows NT.
- Kapitel 14. DB2 und hohe Verfügbarkeit unter Sun Cluster 2.2, behandelt die DB2-Unterstützung für Fehlerbehebung durch Funktionsübernahme in der Sun Solaris-Betriebssystemumgebung.

Anhänge

- Anhang B. Namenskonventionen, enthält die Regeln, die bei der Benennung von Datenbanken und Objekten zu beachten sind.
- Anhang C. Planen der Datenbankmigration, enthält Informationen zur Migration von Datenbanken auf Version 7.
- Anhang D. Inkompatibilitäten zwischen Releases, behandelt die entstandenen Inkompatibilitäten von Release zu Release bis einschließlich Version 7.

- Anhang E. Unterstützung von Landessprachen, gibt eine Einführung in die DB2-Unterstützung von Landessprachen und enthält Informationen zu Ländern, Sprachen und Codepages.
- Anhang A. Verwenden der DB2-Bibliothek, enthält Informationen zu der Struktur der DB2-Bibliothek, einschließlich Assistenten, Online-Hilfefunktion, Nachrichten und Handbücher.

Kurzübersicht über die anderen Bände des Handbuchs zur Systemverwaltung

Systemverwaltung: Implementierung

Der Band *Systemverwaltung: Implementierung* behandelt die Implementierung des entwickelten Datenbankentwurfs. Die einzelnen Kapitel und Anhänge des Bandes werden im folgenden kurz vorgestellt:

Systemverwaltung mit der Steuerzentrale

- Das Kapitel zum Verwalten von DB2 mit GUI-Tools behandelt die Tools der grafischen Benutzerschnittstelle (GUI), die zur Verwaltung der Datenbank verwendet werden.

Implementieren des Datenbankentwurfs

- Das Kapitel "Vor dem Erstellen einer Datenbank" behandelt die Voraussetzungen, die vor der Erstellung einer Datenbank erfüllt sein müssen.
- Das Kapitel zum Erstellen einer Datenbank behandelt die Aufgaben der Erstellung einer Datenbank sowie der zugehörigen Datenbankobjekte.
- Das Kapitel über das Ändern einer Datenbank behandelt, was vor der Änderung einer Datenbank zu tun ist und welche Aufgaben im Zusammenhang mit dem Ändern oder Löschen einer Datenbank oder zugehöriger Datenbankobjekte erledigt werden müssen.

Datenbanksicherheit

- Das Kapitel zur Steuerung des Datenbankzugriffs beschreibt die Möglichkeiten zur Steuerung des Zugriffs auf die Ressourcen einer Datenbank.
- Das Kapitel zur Protokollierung von DB2-Aktivitäten beschreibt Methoden zur Erkennung und Überwachung unerwünschten bzw. unvorhergesehenen Zugriffs auf Daten.

Versetzen von Daten

- Das Kapitel zu den Dienstprogrammen zum Versetzen von Daten ist eine kurze Einführung in die verschiedenen Möglichkeiten zum Versetzen von Daten, die Sie an das Handbuch *Versetzen von Daten Dienstprogramme und Referenz* verweist.

Wiederherstellung

- Das Kapitel zum Wiederherstellen einer Datenbank behandelt die Punkte, die bei der Auswahl der Methoden zur Wiederherstellung von Datenbanken und Tabellenbereichen zu beachten sind, und enthält Informationen zur Sicherung und Wiederherstellung einer Datenbank oder eines Tabellenbereichs sowie zur Methode der aktualisierenden Wiederherstellung.

Anhänge

- Der Anhang zur Verwendung der DCE Verzeichnisservices enthält Informationen zu den Einsatzmöglichkeiten der DC Verzeichnisservices.
- Der Anhang über den Benutzer-Exit zur Datenbankwiederherstellung behandelt die Verwendung von Benutzer-Exit-Programmen für Datenbankprotokolldateien und beschreibt einige Beispiele von Benutzer-Exit-Programmen.
- Der Anhang über das Absetzen von Befehlen an mehrere Datenbankpartitions-Server behandelt die Verwendung der Shell-Prozeduren *db2_all* und *rah* zum Senden von Befehlen an alle Partitionen in einer Umgebung mit partitionierten Datenbanken.
- Der Anhang über die Arbeitsweise von DB2 für Windows NT mit der Windows NT-Sicherheit beschreibt, wie DB2 mit der Windows NT-Sicherheit funktioniert.
- Der Anhang zur Verwendung des Windows NT-Systemmonitors enthält Informationen über das Registrieren von DB2 beim Windows NT-Systemmonitor und über die Nutzung der Leistungsdaten.
- Der Anhang zur Arbeit mit Windows NT- oder Windows 2000-Datenbankpartitions-Servern enthält Informationen über die verfügbaren Dienstprogramme zur Arbeit mit Datenbankpartitions-Servern unter Windows NT oder Windows 2000.
- Der Anhang zur Konfiguration mehrerer logischer Knoten beschreibt, wie mehrere logische Knoten in einer Umgebung mit partitionierten Datenbanken konfiguriert werden.
- Der Anhang zur Hochgeschwindigkeitskommunikation zwischen Knoten beschreibt, wie VIA (Virtual Interface Architecture) zur Verwendung mit DB2 Universal Database eingerichtet wird.
- Der Anhang den LDAP-Verzeichnisservices enthält Informationen zu den Einsatzmöglichkeiten der LDAP-Verzeichnisservices.
- Der Anhang zur Erweiterung 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.

Systemverwaltung: Optimierung

Der Band *Systemverwaltung: Optimierung* behandelt Themen zur Systemleistung. Damit sind die Themen gemeint, die das Einrichten, Testen und Optimieren der Leistung von Anwendungen sowie der Leistung des Produkts

DB2 Universal Database betreffen. Die einzelnen Kapitel und Anhänge des Bandes werden im folgenden kurz vorgestellt:

Einführung in die Optimierung

- Das Kapitel zu den Leistungselementen führt in die Konzepte und Überlegungen zur Verwaltung und Optimierung der Leistung von DB2 UDB ein.
- Die Übersicht zur Architektur und Verarbeitung stellt die zugrundeliegende Architektur und die Prozesse von DB2 Universal Database vor.

Optimieren der Anwendungsleistung

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

Optimieren und Konfigurieren des Systems

- Das Kapitel zur Leistung bei der Ausführung gibt einen Überblick über die Verwendung von Speicher durch den Datenbankmanager und enthält Informationen zu weiteren Faktoren, die sich auf die Leistung zur Laufzeit auswirken.
- Das Kapitel zum Programm Governor 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 einer Umgebung mit partitionierten Datenbanken bei der Neuverteilung von Daten auf die Partitionen zu beachten sind.
- Das Kapitel zu Vergleichstests (Benchmark-Tests) gibt einen Überblick über Vergleichstests und behandelt verschiedene Aspekte ihrer Durchführung.
- Das Kapitel zur DB2-Konfiguration behandelt die Dateien zur Konfiguration des Datenbankmanagers und der Datenbanken sowie die Werte für die Konfigurationsparameter.

Anhänge

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

Teil 1. DB2 Universal Database

Kapitel 1. Verwalten von DB2 Universal Database

Mit DB2 verfügen Sie über die Flexibilität, eine große Bandbreite von Hardwarekonfigurationen betreiben zu können. DB2 ermöglicht Ihnen, eine spezielle DB2-Produktkonfiguration optimal auf Ihre Hardware- und Anwendungsanforderungen abzustimmen.

DB2 unterstützt viele verschiedene Komplexitätsebenen in Datenbankumgebungen, und Sie erhalten Informationen und Anleitungen, die jeweils speziell auf eine bestimmte Umgebung zugeschnitten sind. Diese sind in detaillierter Form sowohl im Handbuch *Systemverwaltung* als auch in anderen Handbüchern der DB2-Bibliothek enthalten (siehe „Anhang A. Verwenden der DB2-Bibliothek“ auf Seite 375). In einigen Fällen treffen ganze Abschnitte dieser Handbücher nur eine bestimmte Umgebung zu. Im Vorwort zu diesem Handbuch („Zu diesem Handbuch“) erfahren Sie, welche Kapitel in diesem und den anderen Bänden des Handbuchs *Systemverwaltung* (*Systemverwaltung: Implementierung* und *Systemverwaltung: Optimierung*) zur Umsetzung Ihrer Geschäftsanforderung relevant sind.

Wenn Sie keine Erfahrung mit Verwaltungssystemen für relationale Datenbanken (RDBMS) oder mit DB2 haben, sind die Informationen im Abschnitt „Allgemeine Konzepte relationaler Datenbanken“ für Sie sicherlich hilfreich. Wenn Sie mit diesen Konzepten bereits vertraut sind oder keine Auffrischung benötigen, können Sie sich direkt den Abschnitten zuwenden, die weiterführende Themen wie zum Beispiel folgende im Einzelnen behandeln:

- Systeme zusammengesetzter Datenbanken. In diesen Abschnitten werden Datenbankverwaltungssysteme (DBMS) behandelt, die Anwendungen und Benutzer unterstützen, die SQL-Anweisungen übergeben, die auf zwei oder mehr DBMSs oder Datenbanken in einer einzigen Anweisung verweisen.
- Systeme paralleler Datenbanken. Dieser Abschnitt enthält eine Einführung in die Arten von Parallelität, die mit Hilfe von DB2 implementiert werden können. Komponenten einer Task, wie zum Beispiel einer Datenbankabfrage, können parallel ausgeführt werden, um die Leistung erheblich zu erhöhen.
- Verteilte Transaktionsverarbeitung. Dieser Abschnitt enthält Informationen darüber, wie in einer einzigen Transaktion auf mehrere Datenbanken zugegriffen werden kann und wie Datenbanken in einer Umgebung mit verteilter Transaktionsverarbeitung eingesetzt werden können.
- Systeme mit hoher Verfügbarkeit. Dieser Abschnitt bietet eine Übersicht über die von DB2 bereitgestellte Unterstützung von Funktionsübernahmen zur Implementierung hoher Verfügbarkeit. Durch die Möglichkeit einer

Funktionsübernahme kann im Falle eines Hardwarefehlers eine Auslastung von einem Prozessor automatisch auf einen anderen verlegt werden.

DB2 kann den spezialisiertesten Anforderungen des Datenmanagements Rechnung tragen, wie zum Beispiel folgenden:

- *Replikation*, die ein Kopieren von Daten auf mehrere ferne Datenbanken in regelmäßigen Abständen ermöglicht. Wenn Aktualisierungen aus einer Masterdatenbank automatisch in andere Datenbanken kopiert werden müssen, können Sie mit Hilfe der Replikationsfunktionen von DB2 angeben, welche Daten zu kopieren sind, in welche Datenbanktabellen die Daten zu kopieren sind und wie oft die Aktualisierungen zu kopieren sind. Falls Sie auf die Replikationseinrichtungen von DB2 zurückgreifen wollen, lesen Sie die Informationen im *Replikation Benutzer- und Referenzhandbuch*. Dieses Handbuch führt in die Konzepte der DB2-Datenreplikation ein und beschreibt die Planung, Konfiguration und Verwaltung einer Replikationsumgebung.
- *Data Warehouses*, die eine Erstellung von Speichern „informativer Daten“ bzw. von Daten ermöglichen, die aus den Betriebsdaten extrahiert und für Endbenutzer zur Entscheidungsfindung aufbereitet werden. Zum Beispiel könnte ein Data Warehouse-Tool alle Verkaufsdaten aus der Betriebsdatenbank kopieren, Berechnungen durchführen, um die Daten zusammenzufassen, und die zusammengefaßten Daten in eine Zieltabelle einer getrennten Datenbank schreiben. Die getrennte Datenbank (das sog. *Warehouse*) kann abgefragt werden, ohne die Betriebsdatenbanken zu beeinträchtigen. Detaillierte Informationen zu Data Warehouses finden Sie im Handbuch *Data Warehouse-Zentrale Verwaltung*.
- Ein *geografisches Informationssystem* (GIS - Geographic Information System), das durch den Spatial Extender erstellt werden kann. Ein GIS ist ein Komplex von Objekten, Daten und Anwendungen, der es ermöglicht, räumliche Informationen über geografische Merkmale zu generieren und zu analysieren. Im Spatial Extender kann ein geografisches Merkmal durch eine Zeile in einer Tabelle oder Sicht bzw. durch einen Teil einer solchen Zeile dargestellt werden. Detaillierte Informationen zur Verwendung des Spatial Extender finden Sie im Handbuch *Spatial Extender Benutzer- und Referenzhandbuch*.

Das Handbuch *Systemverwaltung: Konzept* behandelt auch das Thema des Datenbankentwurfs, der die für DB2 relevanten Aspekte des logischen und physischen Datenbankdesigns umfaßt. Andere konzeptionelle Themen, wie zum Beispiel die Planung einer Datenbankmigration, die Ermittlung von Inkompatibilitäten, die sich auf die Anwendungen auswirken können (eine *Inkompatibilität* ist ein Teil von DB2 Universal Database, das anders funktioniert als in einem früheren Release von DB2. Falls auf eine solche in einer vorhandenen Anwendung zurückgegriffen wird, führt sie zu einem unerwar-

teten Ergebnis, macht eine Änderung der Anwendung erforderlich oder verringert die Leistung) und die Nutzung der Unterstützung nationaler Zeichensätze werden ebenfalls behandelt.

Der Band *Systemverwaltung: Implementierung* enthält Informationen zur Implementierung des entwickelten Datenbankentwurfs. Zu den behandelten Themen gehören das Erstellen und Ändern einer Datenbank, Datenbanksicherheit, Wiederherstellen von Datenbanken und die Verwaltung von DB2 über die Steuerzentrale, eine grafische Benutzerschnittstelle für DB2.

Der Band *Systemverwaltung: Optimierung* behandelt Themen und Aspekte der Leistung, Leistungstests und Methoden der Leistungsoptimierung für Anwendungen und DB2.

Teil 2. Datenbankkonzepte

Kapitel 2. Allgemeine Konzepte relationaler Datenbanken

Dieser Abschnitt behandelt die folgenden Themen:

- „Übersicht über Datenbankobjekte“
- „Übersicht über Wiederherstellungsobjekte“ auf Seite 14
- „Übersicht über Speicherobjekte“ auf Seite 16
- „Übersicht über Systemobjekte“ auf Seite 22
- „Geschäftsregeln für Daten“ auf Seite 24
- „Wiederherstellen einer Datenbank“ auf Seite 28
- „Reorganisieren von Tabellen in einer Datenbank“ auf Seite 59
- „Übersicht über die Sicherheit unter DB2“ auf Seite 59

Übersicht über Datenbankobjekte

Dieser Abschnitt enthält eine Übersicht über die folgenden wichtigen Datenbankobjekte:

- Exemplare
- Datenbanken
- Knotengruppen
- Tabellen
- Sichten
- Indizes
- Schemata
- Systemkatalogtabellen

Abb. 1 auf Seite 10 verdeutlicht die Beziehungen zwischen einigen dieser Objekte. Sie zeigt außerdem, daß Tabellen, Indizes und lange Daten (LOBs) in Tabellenbereichen gespeichert werden.

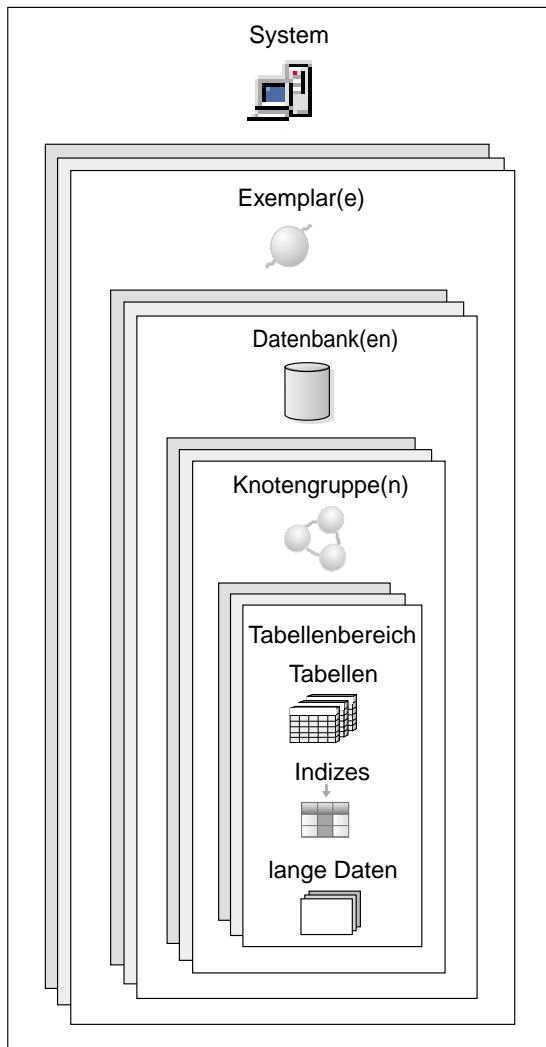


Abbildung 1. Beziehungen zwischen einigen Datenbankobjekten

Exemplare

Ein *Exemplar* (zuweilen auch als *Datenbankmanager* bezeichnet) ist der DB2-Code, der die Daten verwaltet. Er steuert, welche Operationen an den Daten ausgeführt werden können, und verwaltet die Systemressourcen, die ihm zugeordnet sind. Jedes Exemplar stellt eine vollständige Umgebung dar. Es enthält sämtliche Datenbankpartitionen, die für ein bestimmtes paralleles Datenbanksystem definiert sind (siehe „Kapitel 4. Parallele Datenbanksysteme“ auf Seite 71). Ein Exemplar verfügt über eigene Daten-

banken (auf die andere Exemplare keinen Zugriff haben), und alle zugehörigen Datenbankpartitionen benutzen gemeinsame Systemverzeichnisse. Es verfügt außerdem über ein von anderen Exemplaren auf derselben Maschine (System) getrenntes Sicherheitssystem.

Datenbanken

Eine *relationale Datenbank* stellt Daten als Sammlung von Tabellen dar. Eine Tabelle besteht auf einer definierten Anzahl von Spalten und einer beliebigen Anzahl von Zeilen. Jede Datenbank enthält einen Satz von Katalogtabellen, die die logische und physische Struktur der Daten beschreiben, eine Konfigurationsdatei, die die der Datenbank zugeordneten Parameterwerte enthält, und ein Wiederherstellungsprotokoll, das laufende Transaktionen und archivierbare Transaktionen enthält.

Knotengruppen

Eine *Knotengruppe* ist eine Gruppe aus einer oder mehreren Datenbankpartitionen. Wenn Sie Tabellen für die Datenbank erstellen wollen, erstellen Sie zunächst die Knotengruppe, in der die Tabellenbereiche gespeichert werden. Anschließend erstellen Sie die Tabellenbereiche, in denen die Tabellen gespeichert werden. Im Abschnitt „Knotengruppen und Datenpartitionierung“ auf Seite 72 finden Sie weitere Informationen zu Knotengruppen. Eine Definition des Begriffs Datenbankpartition finden Sie in „Kapitel 4. Parallele Datenbanksysteme“ auf Seite 71. Im Abschnitt „Tabellenbereiche“ auf Seite 16 finden Sie weitere Informationen zu Tabellenbereichen.

Tabellen

Eine relationale Datenbank stellt Daten als Sammlung von Tabellen dar. Eine *Tabelle* besteht aus Daten, die logisch in Spalten und Zeilen angeordnet sind. Alle Datenbank- und Tabellendaten werden Tabellenbereichen zugeordnet. Im Abschnitt „Tabellenbereiche“ auf Seite 16 finden Sie weitere Informationen zu Tabellenbereichen. Die Daten in der Tabelle besitzen eine logische Beziehung, und zwischen Tabellen können Abhängigkeitsbedingungen definiert werden. Daten können nach mathematischen Prinzipien und mit Hilfe als *Relationen* bezeichneter Operationen in Sichten zusammengefaßt und bearbeitet werden.

Der Zugriff auf Tabellendaten erfolgt mit Hilfe der Structured Query Language (SQL, siehe Handbuch *SQL Reference*), einer standardisierten Sprache zur Definition und Bearbeitung von Daten in einer relationalen Datenbank. Eine *Abfrage* wird in Anwendungen oder von Benutzern verwendet, um Daten aus einer Datenbank abzurufen. In der Abfrage wird unter Verwendung von SQL eine Anweisung in folgendem Format erstellt:

```
SELECT <datenname> FROM <tabellenname>
```

Sichten

Eine *Sicht* bietet eine effektive Methode zur Darstellung von Daten, ohne sie pflegen zu müssen. Eine Sicht ist keine wirkliche Tabelle und erfordert keine permanente Speicherung. Es wird eine „virtuelle Tabelle“ erstellt und verwendet.

Eine Sicht kann alle oder einige der Spalten oder Zeilen der Tabelle enthalten, auf der sie basiert. Zum Beispiel können Sie eine Tabelle mit Abteilungsdaten (DEPARTMENT) und eine Tabelle mit Mitarbeiterdaten (EMPLOYEE) in einer Sicht verknüpfen, sodaß Sie alle Mitarbeiter in einer bestimmten Abteilung auflisten können.

Abb. 2 zeigt die Beziehung zwischen Tabellen und Sichten.

Datenbank

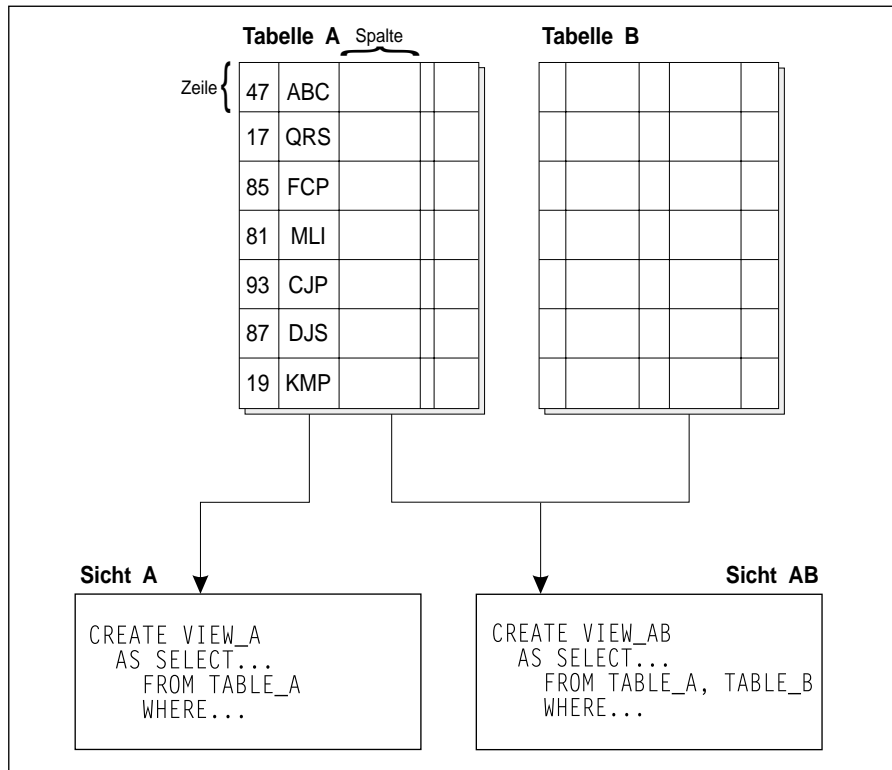


Abbildung 2. Beziehung zwischen Tabellen und Sichten

Indizes

Ein *Index* ist eine Menge von Schlüsseln, die jeweils auf Zeilen in einer Tabelle zeigen. Zum Beispiel hat Tabelle A in Abb. 3 auf Seite 13 einen Index,

der auf den Personalnummern in der Tabelle basiert. Dieser Schlüsselwert dient als Zeiger auf die Zeile in der Tabelle: die Personalnummer 19 zeigt auf Mitarbeiter KMP. Ein Index ermöglicht einen effizienteren Zugriff auf Zeilen einer Tabelle, indem er einen direkten Pfad zu den Daten mit Hilfe von Zeigern erstellt.

Das *SQL-Optimierungsprogramm* wählt automatisch den effizientesten Weg für den Zugriff auf Daten in Tabellen aus. Das Optimierungsprogramm bezieht Indizes bei der Ermittlung des schnellsten Pfads zu den Daten mit in Betracht.

Es können eindeutige Indizes erstellt werden, um die Eindeutigkeit des Indexschlüssels sicherzustellen. Ein *Indexschlüssel* ist eine Spalte oder eine geordnete Gruppe von Spalten, auf denen ein Index definiert wird. Mit Hilfe eines eindeutigen Index wird gewährleistet, daß der Wert jedes Indexschlüssels in der indizierten Spalte bzw. den indizierten Spalten eindeutig ist. Im Abschnitt „Geschäftsregeln für Daten“ auf Seite 24 werden Schlüssel und Indizes eingehender behandelt.

Abb. 3 illustriert die Beziehung zwischen einem Index und einer Tabelle.

Datenbank

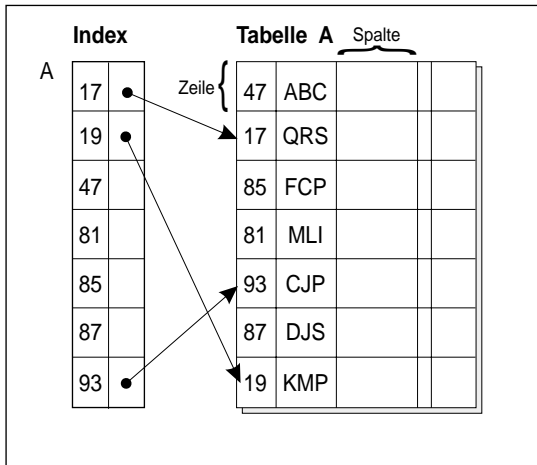


Abbildung 3. Beziehung zwischen einem Index und einer Tabelle

Schemata

Ein *Schema* ist eine Kennung, wie zum Beispiel eine Benutzer-ID, die bei der Gruppierung von Tabellen und anderen Datenbankobjekten hilfreich ist. Ein Schema kann einem Eigner zugeordnet sein, der den Zugriff auf die in ihm enthaltenen Daten und Objekte steuern kann.

Ein Schema ist außerdem ein Objekt in der Datenbank. Es kann automatisch erstellt werden, wenn das erste Objekt in einem Schema erstellt wird. Ein solches Objekt kann ein beliebiges Objekt sein, das durch einen Schemennamen qualifiziert werden kann, wie zum Beispiel eine Tabelle, ein Index, eine Sicht, ein Paket, ein einzigartiger Datentyp, eine Funktion oder ein Auslöser. Sie müssen über die Berechtigung `IMPLICIT_SCHEMA` verfügen, wenn das Schema automatisch erstellt werden soll. Sie können das Schema außerdem auch explizit erstellen.

Ein Schemenname wird als erster Teil eines zweiteiligen Objektens verwendet. Bei der Erstellung eines Objekts können Sie es einem bestimmten Schema zuordnen. Wenn Sie kein Schema angeben, wird das Objekt dem Standardschema zugeordnet, das in der Regel die Benutzer-ID der Person ist, die das Objekt erstellt hat. Der zweite Teil des Namens ist der Name des Objekts. Zum Beispiel könnte ein Benutzer namens Smith eine Tabelle des Namens `SMITH.PAYROLL` besitzen.

Systemkatalogtabellen

Jede Datenbank enthält einen Satz von *Systemkatalogtabellen*, die die logische und physische Struktur der Daten beschreiben. DB2 erstellt und pflegt einen umfangreichen Satz von Systemkatalogtabellen für jede einzelne Datenbank. Diese Tabellen enthalten Informationen über die Definitionen der Definitionen von Datenbankobjekten, wie zum Beispiel Benutzertabellen, Sichten und Indizes sowie Sicherheitsinformationen über die Berechtigungen, die Benutzer für diese Objekte besitzen. Sie werden bei der Erstellung der Datenbank erstellt und im Rahmen des normalen Datenbankbetriebs aktualisiert. Sie können nicht explizit erstellt oder gelöscht werden, jedoch kann ihr Inhalt mit Hilfe von Katalogsichten abgefragt und angezeigt werden.

Übersicht über Wiederherstellungsobjekte

Protokolldateien und die Datei des Wiederherstellungsprotokolls werden automatisch erstellt, wenn eine Datenbank erstellt wird (Abb. 4 auf Seite 15). Eine Protokolldatei oder die Datei des Wiederherstellungsprotokolls kann nicht direkt geändert werden. Jedoch spielen sie eine wichtige Rolle, wenn mit Hilfe des Sicherungsabbilds der Datenbank Daten wiederhergestellt werden müssen, die verloren oder beschädigt wurden.

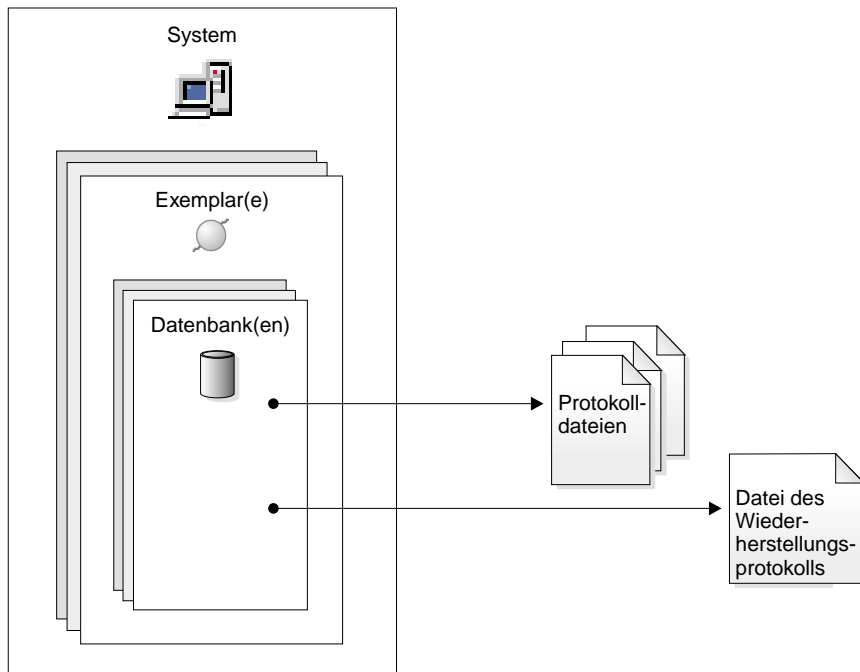


Abbildung 4. Protokolldateien und die Datei des Wiederherstellungsprotokolls

Protokolldateien zur Wiederherstellung

Jede Datenbank enthält *Wiederherstellungsprotokolle*, die zur Datenwiederherstellung nach Anwendungs- oder Systemfehlern verwendet werden. In Kombination mit den Datenbanksicherungen dienen sie zur Wiederherstellung der Konsistenz der Datenbank bis exakt zu dem Zeitpunkt, zu dem der Fehler auftrat. Die Datenbankwiederherstellung wird im Abschnitt „Wiederherstellen einer Datenbank“ auf Seite 28 eingehender behandelt.

Datei des Wiederherstellungsprotokolls

Die *Datei des Wiederherstellungsprotokolls* enthält eine Zusammenfassung der Sicherungsinformationen, die verwendet werden können, wenn die Datenbank insgesamt oder teilweise bis zu einem bestimmten Zeitpunkt wiederhergestellt werden muß. Sie dient zur Protokollierung wiederherstellungsrelevanter Ereignisse wie Sicherungs-, Wiederherstellungs- und Datenladeoperationen. Die Prozedur zur Sicherung und Wiederherstellung einer Datenbank wird im

Abschnitt „Wiederherstellen einer Datenbank“ auf Seite 28 behandelt. Das Dienstprogramm LOAD wird im Handbuch *Versetzen von Daten Dienstprogramme und Referenz* beschrieben.

Übersicht über Speicherobjekte

Mit Hilfe der folgenden Datenbankobjekte können Sie definieren, wie Daten auf dem System gespeichert werden und wie die (für den Zugriff auf die Daten relevante) Leistung verbessert werden kann:

- Tabellenbereich
- Behälter
- Pufferpool

Tabellenbereiche

Eine Datenbank wird in Bestandteilen verwaltet, die als *Tabellenbereiche* bezeichnet werden. Ein Tabellenbereich ist ein Platz zum Speichern von Tabellen. Bei der Erstellung einer Tabelle können Sie sich entschließen, bestimmte Objekte wie Indizes und große Objekte (LOB-Daten) von den übrigen Tabellendaten getrennt zu speichern. Ein Tabellenbereich kann außerdem über eine oder mehrere physische Speichereinheiten verteilt angelegt werden. Das folgende Diagramm veranschaulicht etwas von der Flexibilität, die Sie durch die Verteilung von Daten auf Tabellenbereiche erhalten:

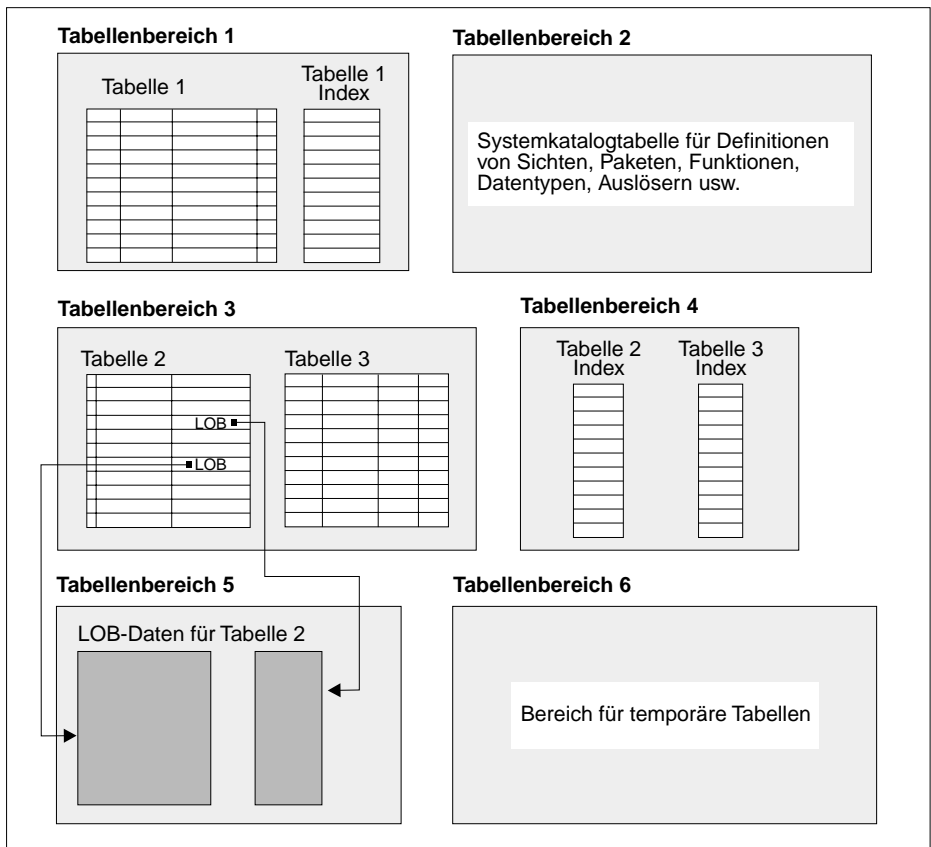


Abbildung 5. Tabellenbereiche

Tabellenbereiche befinden sich in Knotengruppen (siehe „Knotengruppen“ auf Seite 11). Die Definitionen und Attribute von Tabellenbereichen werden im Systemkatalog der Datenbank aufgezeichnet (siehe „Systemkatalogtabellen“ auf Seite 14).

Behälter werden Tabellenbereichen zugeordnet. Ein *Behälter* ist eine Zuordnung physischen Speichers (z. B. eine Datei oder eine Einheit).

Ein Tabellenbereich kann entweder ein vom System verwalteter Bereich (SMS - System Managed Space) oder ein von der Datenbank verwalteter Bereich (DMS - Database Managed Space) sein. Bei einem SMS-Tabellenbereich ist jeder Behälter ein Verzeichnis im Dateibereich des Betriebssystems, wobei der Speicherbereich vom Dateimanager des Betriebssystems gesteuert wird. Bei einem DMS-Tabellenbereich ist jeder Behälter entweder eine im voraus zugeordnete Datei fester Größe oder eine physische Einheit wie eine Platte, wobei der Datenbankmanager den Speicherbereich steuert.

Abb. 6 illustriert die Beziehung zwischen Tabellen, Tabellenbereichen und den beiden Typen von Speicherbereichen. Sie zeigt außerdem, daß Tabellen, Indizes und lange Daten (LOB-Daten) in Tabellenbereichen gespeichert werden.

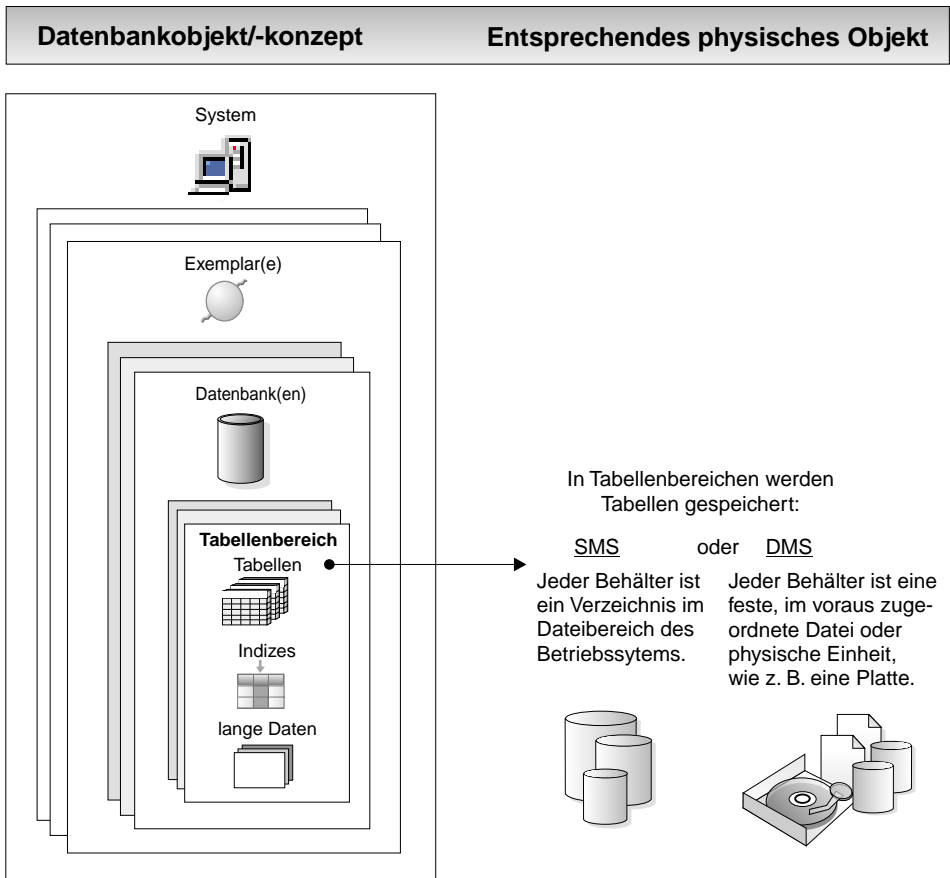


Abbildung 6. Tabellenbereiche und Tabellen

Abb. 7 auf Seite 19 zeigt die drei Arten von Tabellenbereichen: *regular*, *temporary*, und *long*.

Tabellen, die Benutzerdaten enthalten, werden in regulären Tabellenbereichen gespeichert. Der Standardtabellenbereich für Benutzer besitzt den Namen `USERSPACE1`. Indizes werden ebenfalls in regulären Tabellenbereichen gespeichert. Die Systemkatalogtabellen werden in einem regulären Tabellenbereich angelegt. Der Standardtabellenbereich für Systemkatalogtabellen heißt `SYS-CATSPACE`.

Tabellen, die Spalten mit sehr langen Zeichenfolgen (Long) oder lange Objekte (LOB) wie zum Beispiel Multimediaobjekte enthalten, werden in Tabellenbereichen für lange Objektdaten (LONG) gespeichert.

Temporäre Tabellenbereiche werden als Systemtabellenbereiche oder Benutzertabellenbereiche klassifiziert. *Temporäre Systemtabellenbereiche* dienen zur Speicherung interner temporärer Daten, die für SQL-Operationen wie Sortieren, Reorganisieren von Tabellen, Erstellen von Indizes und Verknüpfen von Tabellen erforderlich sind. Obwohl eine beliebige Anzahl von temporären Systemtabellenbereichen erstellt werden kann, ist es zu empfehlen, nur einen solchen Tabellenbereich zu erstellen und dabei die Seitengröße zu verwenden, die von der Mehrheit der Tabellen verwendet wird. Der Standardtabellenbereich für temporäre Systemdaten heißt TEMPSPACE1. *Temporäre Benutzertabellenbereiche* dienen zur Speicherung deklarierter globaler temporärer Tabellen, in denen temporäre Daten abgelegt werden. Temporäre Benutzertabellenbereiche werden *nicht* standardmäßig bei der Erstellung einer Datenbank erstellt.

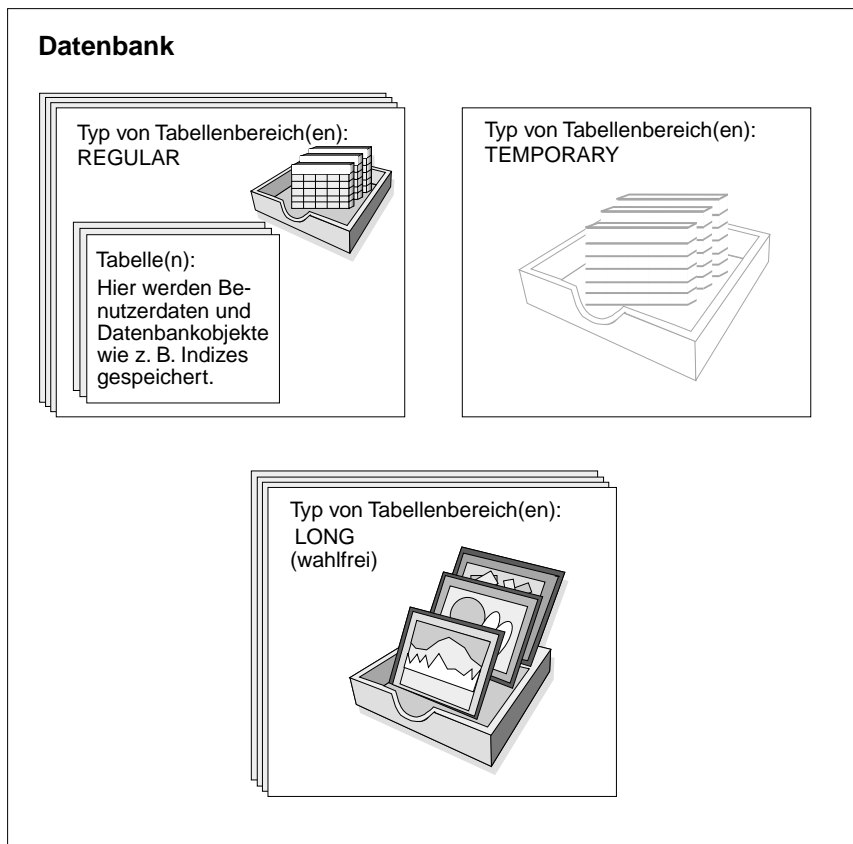


Abbildung 7. Drei Typen von Tabellenbereichen

Behälter

Ein *Behälter* ist eine physische Speichereinheit. Er kann durch einen Verzeichnisnamen, einen Einheitenamen oder einen Dateinamen bezeichnet werden.

Ein Behälter ist einem Tabellenbereich zugeordnet. Ein einzelner Tabellenbereich kann sich über viele Behälter erstrecken, jedoch kann ein Behälter jeweils nur zu einem Tabellenbereich gehören.

Abb. 8 veranschaulicht die Beziehung zwischen Tabellen und einer Tabelle innerhalb einer Datenbank und den zugeordneten Behältern und Platten.

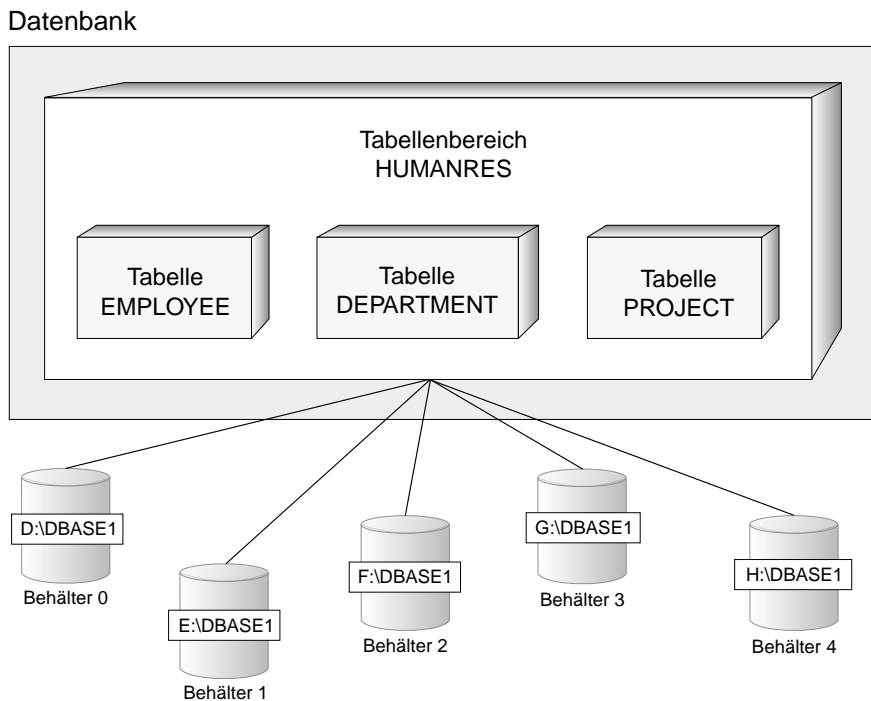


Abbildung 8. Tabellenbereiche und Tabellen in einer Datenbank

Die Tabellen **EMPLOYEE**, **DEPARTMENT** und **PROJECT** befinden sich im Tabellenbereich **HUMANRES**, der sich über die Behälter 0, 1, 2, 3 und 4 erstreckt. Dieses Beispiel zeigt jeden Behälter auf einer getrennten Platte.

Daten aller Tabellen werden in allen Behältern in einer Tabelle reihum verteilt gespeichert. Dies sorgt für eine gleichmäßige Verteilung der Daten auf die Behälter, die zu einem bestimmten Tabellenbereich gehören. Die Anzahl von

Seiten, die der Datenbankmanager in einen Behälter schreibt, bevor er zu einem anderen Behälter wechselt, wird als Speicherbereich *Speicherbereich* (Extentsize) bezeichnet.

Pufferpool

Ein *Pufferpool* ist die Menge an Hauptspeicher, die beim Lesen von Daten vom Datenträger oder beim Ändern von Daten zum Zwischenspeichern von Tabellen- und Indexdatenseiten zugeordnet wird. Der Pufferpool dient zur Erhöhung der Systemleistung. Auf Daten im Hauptspeicher kann wesentlich schneller zugegriffen werden als auf Daten auf einem Datenträger. Daher ist die Leistung umso höher, je weniger oft der Datenbankmanager Daten von einem Datenträger lesen bzw. auf ihn schreiben (E/A) muß. (Sie können mehr als einen Pufferpool erstellen, wenngleich für die meisten Fälle nur ein Pufferpool erforderlich ist.)

Die Konfiguration des Pufferpools ist der wichtigste Einzelbereich der Optimierung, weil hier die Verzögerung durch eine langsame Ein-/Ausgabe (E/A) verringert werden kann.

Abb. 9 auf Seite 22 veranschaulicht die Beziehung zwischen einem Pufferpool und Behältern.

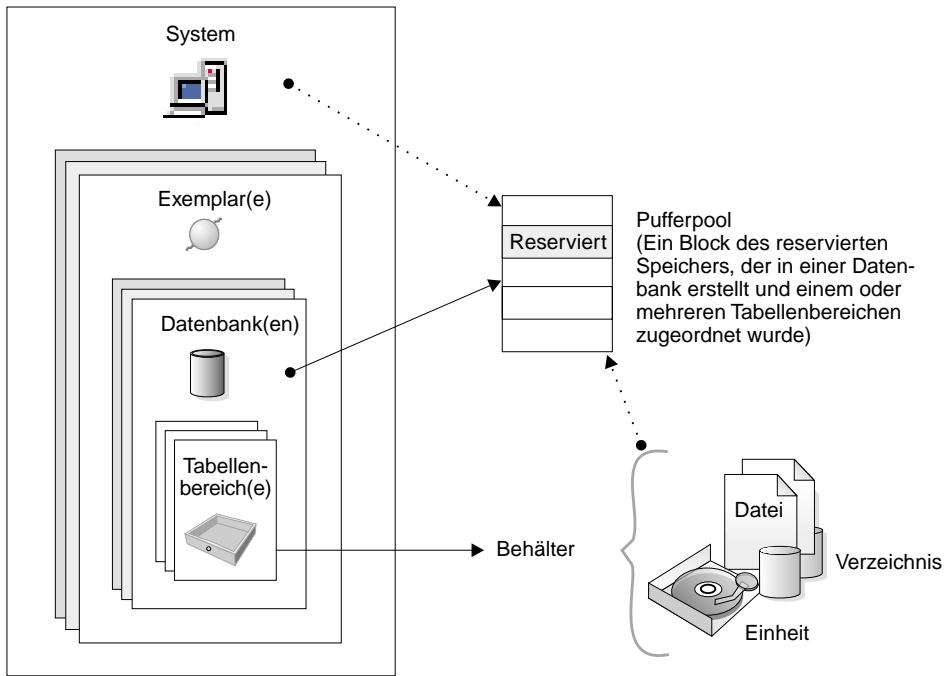


Abbildung 9. Pufferpool und Behälter

Übersicht über Systemobjekte

Bei der Erstellung eines DB2-Exemplars oder einer DB2-Datenbank wird eine entsprechende Konfigurationsdatei mit den Standardparameterwerten erstellt. Diese Parameterwerte können zur Verbesserung der Leistung geändert werden.

Konfigurationsparameter

Konfigurationsdateien enthalten Parameter, die Werte definieren, wie zum Beispiel die den DB2-Produkten zugeordneten Ressourcen und die Diagnoseebene. Es gibt zwei Arten von Konfigurationsdateien: eine Konfigurationsdatei des Datenbankmanagers für jedes DB2-Exemplar und eine Konfigurationsdatei der Datenbank für jede einzelne Datenbank (siehe Abb. 10 auf Seite 24).

Eine *Konfigurationsdatei des Datenbankmanagers* wird erstellt, wenn ein DB2-Exemplar erstellt wird. Die in ihr enthaltenen Parameter betreffen Systemressourcen auf Exemplarebene und sind von keiner Datenbank, die zu diesem Exemplar gehört, abhängig. Die Werte vieler dieser Parameter können von den Standardwerten des System zur Leistungsoptimierung oder Kapazitätserhöhung abhängig von der Konfiguration des Systems geändert werden.

Darüber hinaus gibt es auch für jede Client-Installation eine Konfigurationsdatei des Datenbankmanagers. Diese Datei enthält Informationen über den Client Enabler für eine bestimmte Workstation. Eine Untergruppe der für einen Server verfügbaren Parameter gelten für den Client.

Eine *Konfigurationsdatei der Datenbank* wird erstellt, wenn eine Datenbank erstellt wird. Sie befindet sich dort, wo sich die Datenbank befindet. Pro Datenbank gibt es eine Konfigurationsdatei. Die in ihr enthaltenen Parameter geben neben anderen Merkmalen die Menge der Ressourcen an, die der zugehörigen Datenbank zugeordnet sind. Die Werte vieler dieser Parameter können zur Leistungsoptimierung und Kapazitätserhöhung geändert werden. Abhängig von der Art der Aktivitäten in einer bestimmten Datenbank können unterschiedliche Änderungen erforderlich sein.

Datenbankobjekt/-konzept	Entsprechendes physisches Objekt
--------------------------	----------------------------------

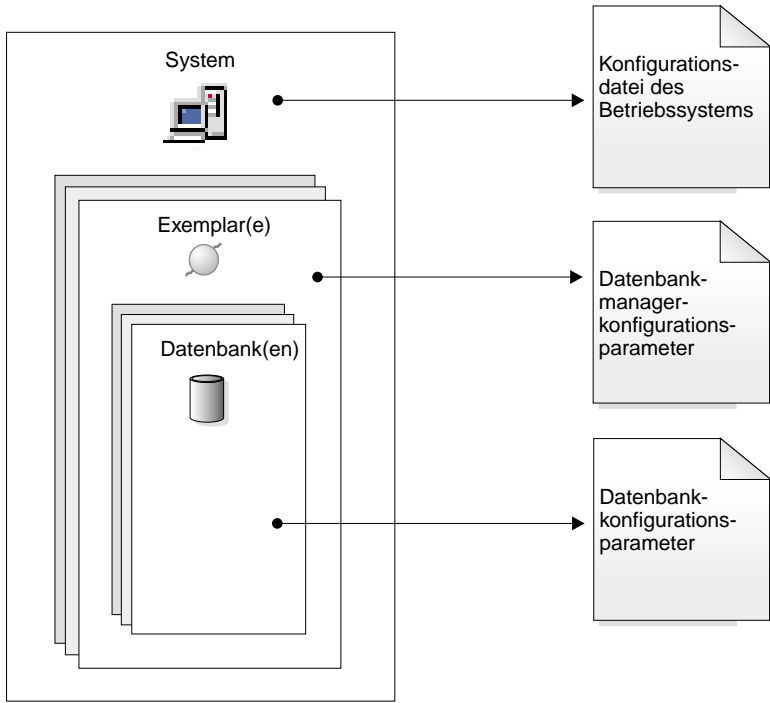


Abbildung 10. Konfigurationsparameterdateien

Geschäftsregeln für Daten

In jeder Geschäftsumgebung müssen Daten häufig bestimmten Rahmenbedingungen oder Regeln genügen. Zum Beispiel muss eine Personalnummer eindeutig sein. DB2 stellt *Integritätsbedingungen* bereit, die zur Implementierung solcher Regeln definiert werden können.

DB2 stellt die folgenden Arten von Integritätsbedingungen zur Verfügung:

- Integritätsbedingung NOT NULL
- Eindeutige Integritätsbedingung
- Integritätsbedingung über Primärschlüssel
- Integritätsbedingung über Fremdschlüssel
- Prüfung auf Integritätsbedingung

Integritätsbedingung NOT NULL

Die Integritätsbedingung NOT NULL verhindert, daß Nullwerte in eine Spalte eingegeben werden.

Eindeutige Integritätsbedingung

Eindeutige Integritätsbedingungen stellen sicher, daß die Werte in einer Gruppe von Spalten eindeutig sind und für alle Zeilen in der Tabelle nicht null sind. Zum Beispiel könnte eine typische eindeutige Integritätsbedingung in einer Tabelle DEPARTMENT mit Daten über Abteilungen darin bestehen, daß die Abteilungsnummer eindeutig und nicht null ist.

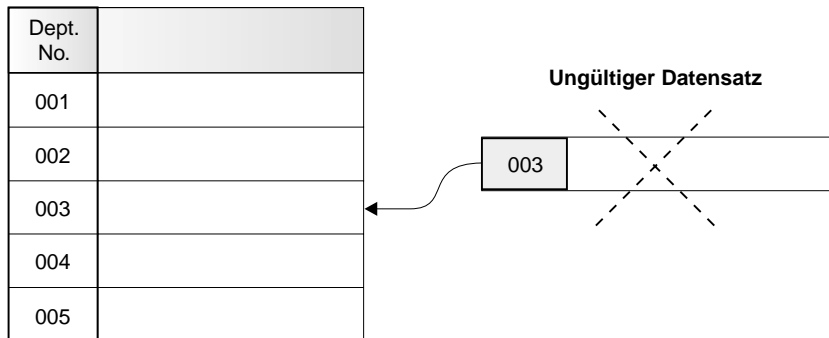


Abbildung 11. Eindeutige Integritätsbedingungen verhindern mehrfach auftretende Daten.

Der Datenbankmanager beachtet die Integritätsbedingung bei Operationen zum Einfügen und Aktualisieren von Daten, um die Datenintegrität zu gewährleisten.

Integritätsbedingung über Primärschlüssel

Jede Tabelle kann einen und nur einen Primärschlüssel besitzen. Ein Primärschlüssel ist eine Spalte oder eine Kombination von Spalten, die die gleichen Merkmale wie eine eindeutige Integritätsbedingung besitzen. Ein Primärschlüssel und Integritätsbedingungen über Fremdschlüssel dienen zur Definition von Beziehungen zwischen Tabellen.

Da ein Primärschlüssel zur Angabe einer Zeile in einer Tabelle verwendet wird, sollte er eindeutig sein und möglichst wenigen Hinzufüge- oder Löschoptionen unterliegen. Eine Tabelle kann nicht mehr als einen Primärschlüssel, jedoch mehrere eindeutige Schlüssel besitzen. Primärschlüssel sind wahlfrei und können definiert werden, wenn eine Tabelle erstellt oder geändert wird. Sie besitzen zudem den weiteren Vorteil, daß sie für eine Reihenfolge der Daten sorgen, wenn Daten exportiert oder reorganisiert werden.

In den folgenden Tabellen sind DEPTNO und EMPNO die Primärschlüssel der Tabellen DEPARTMENT und EMPLOYEE.

Tabelle 1. Tabelle DEPARTMENT

DEPTNO (Primär-schlüssel)	DEPTNAME	MGRNO
A00	Spiffy Computer Service Division	000010
B01	Planning	000020
C01	Information Center	000030
D11	Manufacturing Systems	000060

Tabelle 2. Tabelle EMPLOYEE

EMPNO (Primär-schlüssel)	FIRSTNAME	LASTNAME	WORKDEPT (Fremd-schlüssel)	PHONENO
000010	Christine	Haas	A00	3978
000030	Sally	Kwan	C01	4738
000060	Irving	Stern	D11	6423
000120	Sean	O'Connell	A00	2167
000140	Heather	Nicholls	C01	1793
000170	Masatoshi	Yoshimura	D11	2890

Integritätsbedingung über Fremdschlüssel

Integritätsbedingungen über Fremdschlüssel (die auch als referentielle Integritätsbedingungen bezeichnet werden) geben Ihnen die Möglichkeit, erforderliche Beziehungen zwischen und innerhalb von Tabellen zu definieren.

Zum Beispiel könnte eine typische Integritätsbedingung über Fremdschlüssel festlegen, daß jeder Mitarbeiter in der Tabelle EMPLOYEE ein Mitglied einer bestehenden, in der Tabelle DEPARTMENT definierten Abteilung sein muß.

Zur Herstellung dieser Beziehung würden Sie die Abteilungsnummer (WORKDEPT) der Tabelle EMPLOYEE als Fremdschlüssel und die Abteilungsnummer (DEPTNO) der Tabelle DEPARTMENT als Primärschlüssel definieren.

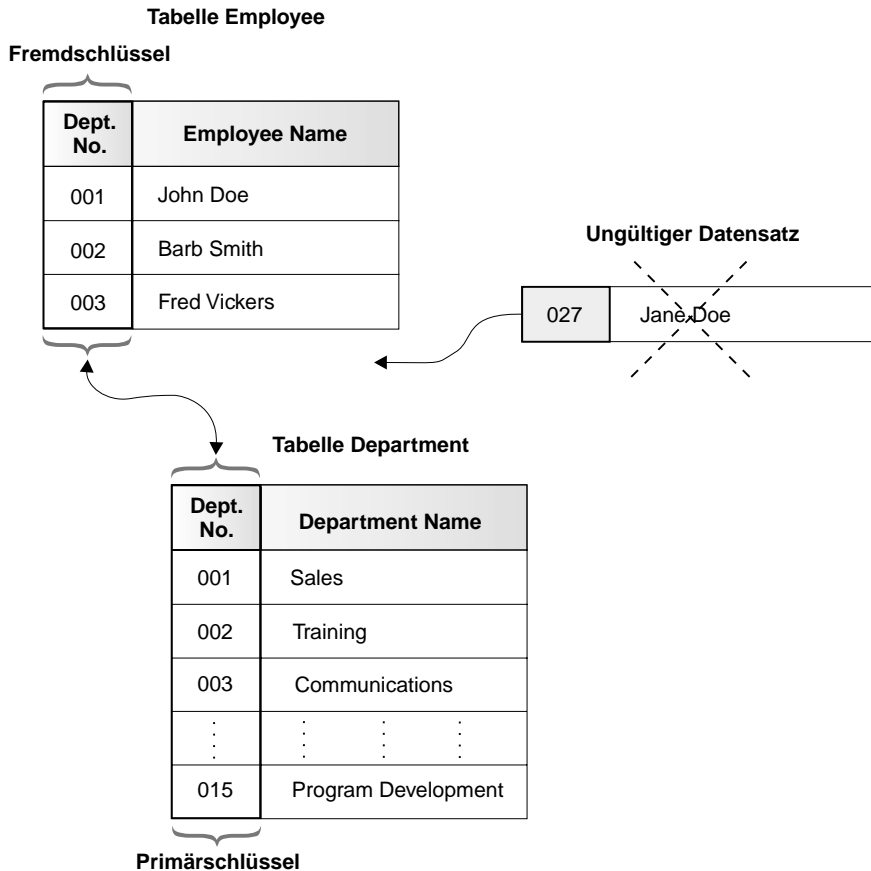


Abbildung 12. Integritätsbedingungen über Fremd- und Primärschlüssel definieren Beziehungen und schützen Daten.

Prüfung auf Integritätsbedingung

Eine Prüfung auf Integritätsbedingung ist eine Datenbankregel, mit der die zulässigen Werte in einer oder mehreren Spalten jeder Zeile einer Tabelle angegeben werden.

Zum Beispiel können Sie einer Tabelle EMPLOYEE definieren, daß die Spalte für die Jobbezeichnung nur die Werte "Sales", "Manager" oder "Clerk" enthalten kann. Unter dieser Integritätsbedingung ist jeder Datensatz mit einem anderen Wert in der Spalte für die Jobbezeichnung ungültig und würde zurückgewiesen, um die Regeln über den Typ der zulässigen Daten in der Tabelle durchzusetzen.

In einer Datenbank können außerdem *Auslöser* genutzt werden. Auslöser sind komplizierter und potentiell leistungsfähiger als Integritätsbedingungen. Sie definieren eine Reihe von Aktionen, die in Verbindung mit einer bzw. ausgelöst durch eine Klausel INSERT, UPDATE oder DELETE an einer angegebenen

Basistabelle ausgeführt wird. Auslöser können zur Unterstützung allgemeiner Formen der Datenintegrität und Geschäftsregeln verwendet werden. Zum Beispiel kann ein Auslöser die Kreditgrenze eines Kunden überprüfen, bevor ein Auftrag akzeptiert wird, oder er kann in einer Bankanwendung eingesetzt werden, um einen Alert auszulösen, wenn eine Geldentnahme aus einem Konto nicht dem gewöhnlichen Geldentnahmestandard eines Kunden entspricht. Weitere Informationen zu Auslösern (Triggers) finden Sie im Handbuch *Application Development Guide*.

Wiederherstellen einer Datenbank

Eine Datenbank kann aufgrund von Hardware- oder Softwarefehlern (oder beiden) unbrauchbar werden. Die verschiedenen Fehlerszenarien können verschiedene Maßnahmen zur Wiederherstellung erforderlich machen. Sie sollten eine erprobte Strategie zur Hand haben, um Ihre Datenbank gegen die Möglichkeit fehlerbedingter Ausfälle zu schützen.

In diesem Abschnitt werden die verschiedenen Wiederherstellungsmethoden behandelt und erläutert, wie sich die für eine gegebene Geschäftsumgebung am besten geeignete Wiederherstellungsmethode bestimmen läßt. Die folgenden Themen werden behandelt:

- „Überblick über die Wiederherstellung“
- „Faktoren mit Auswirkung auf die Wiederherstellung“ auf Seite 35
- „Überlegungen zur Wiederherstellung nach Schäden“ auf Seite 53
- „Begrenzen der Auswirkungen von Datenträgerfehlern“ auf Seite 54
- „Begrenzen der Auswirkungen von Transaktionsfehlern“ auf Seite 57
- „Synchronisation der Systemuhren in einem System mit partitionierten Datenbanken“ auf Seite 57

Überblick über die Wiederherstellung

Sie sollten mit den Strategien vertraut sein, die Ihnen bei Problemen mit Ihrer Datenbank zur Verfügung stehen. Zu diesen Problemen gehören Datenträger- und Speicherausfälle, Unterbrechungen der Stromversorgung und Anwendungsfehler. Sie haben die Möglichkeit, Ihre Datenbank bzw. einzelne Tabellenbereiche zu sichern und ggf. erneut zu erstellen, falls sie auf irgendeine Weise beschädigt oder unbrauchbar gemacht wurden. Das Konzept einer *Datenbanksicherung* ist mit jeder anderen Datensicherung vergleichbar: Erstellen einer Kopie der Daten und Speichern der Kopie auf einem anderen Datenträger für den Fall eines Ausfalls oder einer Beschädigung der Originaldaten. Den einfachsten Fall einer Sicherung bildet das Verfahren, bei dem zunächst die Datenbank gestoppt wird, um sicherzustellen, daß keine weiteren Transaktionen auftreten, und anschließend die Daten gesichert (Backup) werden.

Das erneute Erstellen der Datenbank wird als *Wiederherstellung* bezeichnet. Eine *Wiederherstellung nach einem Systemabsturz* versucht automatisch, die

Datenbank nach einem Fehler wiederherzustellen. Es gibt zwei Methoden, eine beschädigte Datenbank wiederherzustellen: *Versionswiederherstellung* und *aktualisierende Wiederherstellung* (Roll Forward).

Bei nicht wiederherstellbaren Datenbanken sind die beiden Konfigurationsparameter *logretain* und *userexit* der Datenbank inaktiviert. Dies bedeutet, daß lediglich die für Wiederherstellung nach einem Systemabsturz erforderlichen Protokolle beibehalten werden. Diese Protokolle werden als *aktive Protokoll-dateien* bezeichnet und enthalten aktuelle Transaktionsdaten. Die Versionswiederherstellung mit Hilfe von *Offline-Sicherungen* ist das primäre Mittel zur Wiederherstellung für eine nicht wiederherstellbare Datenbank. (Eine Offline-Sicherung bedeutet, daß während der Sicherungsoperation keine andere Anwendung die Datenbank verwenden kann.) Eine solche Datenbank kann nur offline wiederhergestellt werden. Sie wird in dem Status wiederhergestellt, den sie hatte, als das Sicherungsabbild erstellt wurde.

Bei wiederherstellbaren Datenbanken ist entweder der Konfigurationsparameter *logretain* der Datenbank auf den Wert "RECOVERY" gesetzt, oder der Konfigurationsparameter *userexit* der Datenbank aktiviert, oder beides. Aktive Protokolldateien sind weiterhin für Wiederherstellung nach einem Systemabsturz verfügbar, Ihnen stehen jedoch auch die *Archivprotokolldateien* zur Verfügung, die festgeschriebene Transaktionsdaten enthalten. Eine solche Datenbank kann nur offline wiederhergestellt werden. Sie wird in dem Status wiederhergestellt, den sie hatte, als das Sicherungsabbild erstellt wurde. Mit Hilfe der aktualisierenden Wiederherstellung (Roll Forward Recovery) können Sie jedoch die Datenbank *aktualisierend* wiederherstellen (d. h. über den Zeitpunkt hinaus, zu dem das Sicherungsabbild erstellt wurde), indem Sie die aktiven und archivierten Protokolle entweder bis zu einem bestimmten Zeitpunkt oder bis zum Ende der aktiven Protokolldateien anwenden.

Sicherungsoperationen für wiederherstellbare Datenbanken können entweder offline oder *online* (*online* heißt, daß andere Anwendungen während der Sicherungsoperation eine Verbindung zur Datenbank herstellen können) durchgeführt werden. Die Operationen zur Wiederherstellung und aktualisierenden Wiederherstellung der Datenbank müssen immer offline ausgeführt werden. Während einer Online-Sicherung stellt die aktualisierende Wiederherstellung sicher, daß *alle* Tabellenänderungen erfaßt und erneut angewendet werden, wenn diese Sicherungskopie wiederhergestellt wird.

Bei einer wiederherstellbaren Datenbank können Sie auch einzelne Tabellenbereiche sichern, wiederherstellen und aktualisierend wiederherstellen. Wenn Sie einen Tabellenbereich online sichern, kann er weiterhin verwendet werden, und gleichzeitig durchgeführte Änderungen werden in den Protokollen aufgezeichnet. Wenn Sie eine Online-Operation zur Wiederherstellung oder aktualisierenden Wiederherstellung für einen Tabellenbereich ausführen, kann der

Tabellenbereich selbst nicht mehr verwendet werden, bis die Operation beendet ist, Benutzer können jedoch weiterhin auf Tabellen in anderen Tabellenbereichen zugreifen.

Durch die Wiederherstellung nach einem Systemabsturz wird verhindert, daß eine Datenbank in einem inkonsistenten, d. h. unbrauchbaren, Status verbleibt. Bei einem Systemabsturz können Transaktionen (bzw. Arbeitseinheiten), die für die Datenbank ausgeführt werden, auf unerwartete Weise unterbrochen werden. Wenn eine Störung auftritt, bevor alle Änderungen, die Bestandteil der Arbeitseinheit sind, beendet und festgeschrieben wurden, verbleibt die Datenbank in einem inkonsistenten und unbrauchbaren Status.

Daraufhin muß die Datenbank wieder in einen konsistenten und brauchbaren Status versetzt werden. Dies geschieht durch Rückgängigmachen der unvollständigen Transaktionen und Beenden der festgeschriebenen Aktionen, die sich noch im Hauptspeicher befanden, als der Systemabsturz auftrat (Abb. 13). Wenn eine Datenbank in einem konsistenten und verwendbaren Zustand ist,

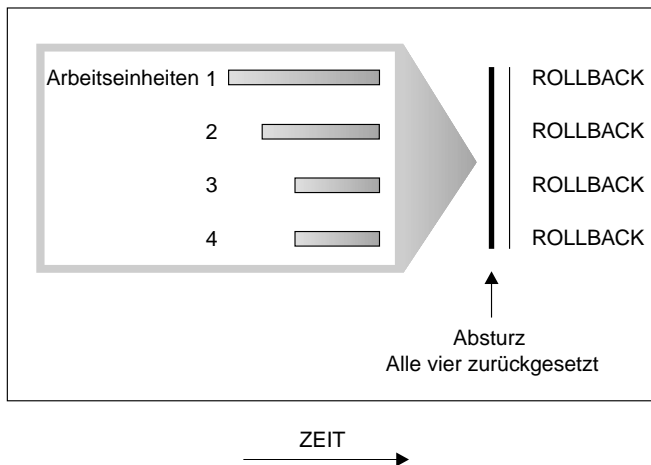


Abbildung 13. Rückgängigmachen von Arbeitseinheiten

hat sie einen Punkt erreicht, der als „Konsistenzzustand“ bezeichnet wird. Eine Offline-Datenbanksicherung stellt einen solchen Konsistenzzustand dar. Wenn ein Konsistenzzustand erreicht ist, sind alle Transaktionen aufgelöst und alle Daten stehen anderen Benutzern oder oder Anwendungen zur Verfügung.

Nach einem Absturz können Sie einen Konsistenzzustand erreichen, indem Sie den Befehl `RESTART DATABASE` ausführen (siehe Handbuch *Command Reference*). Wenn dies bei jedem Fehler geschehen soll, sollten Sie eine Verwendung des Konfigurationsparameters zur **Aktivierung des automatischen Neu-**

starts (autorestart) in Betracht ziehen. Die Standardfunktionsweise dieses Konfigurationsparameters der Datenbank ist jeweils das Aufrufen des Befehls `RESTART DATABASE`, wenn dies erforderlich ist. Wenn der Parameter `autorestart` aktiviert ist, wird durch die nächste Verbindungsanforderung an die Datenbank nach einem Fehler der Befehl `RESTART DATABASE` aufgerufen.

Durch die Wiederherstellung nach einem Systemabsturz wird die Datenbank in einen konsistenten und brauchbaren Status versetzt. Wenn eine Wiederherstellung nach einem Systemabsturz für eine Datenbank durchgeführt wird, für die die aktualisierende Wiederherstellung aktiviert ist (d. h. der Konfigurationsparameter `logretain` ist auf `RECOVERY` gesetzt, oder der Konfigurationsparameter `userexit` ist aktiviert) und während dieser Wiederherstellung ein Fehler auftritt, der auf einen einzelnen Tabellenbereich zurückzuführen ist, muß dieser Tabellenbereich offline genommen werden. Auf ihn kann erst dann wieder zugegriffen werden, nachdem er repariert wurde. Anschließend wird die Wiederherstellung nach dem Systemabsturz fortgesetzt. Nach der Beendigung der Wiederherstellung nach Systemabsturz sind die anderen Tabellenbereiche in der Datenbank weiterhin verwendbar, und es können Verbindungen zur Datenbank hergestellt werden. (Es gibt Ausnahmen bei Tabellenbereichen, die temporäre Tabellen oder Systemkatalogtabellen enthalten. Diese Ausnahmen werden in der Beschreibung der aktualisierenden Wiederherstellung behandelt.)

Wie bereits zuvor erwähnt, bietet DB2 zwei Methoden zur Wiederherstellung einer beschädigten Datenbank:

- Eine *Versionswiederherstellung* ist die Restaurierung einer früheren Version der Datenbank mit Hilfe eines Abbilds der Datenbank, das im Rahmen einer Sicherungsoperation erstellt wurde.

Durch die Operation einer Datenbankwiederherstellung wird die gesamte Datenbank mit Hilfe einer zu einem früheren Zeitpunkt erstellten Sicherungskopie rekonstruiert. Eine Sicherungskopie der Datenbank ermöglicht Ihnen, eine Datenbank in dem Status wiederherzustellen, in dem sie sich zum Zeitpunkt der Sicherung befand. Alle Arbeitseinheiten, die vom Zeitpunkt der Sicherung ausgehend bis zum Eintreten des Fehlers ausgeführt wurden, sind verloren (siehe Abb. 14 auf Seite 32).

Bei Einsatz der Methode der Versionswiederherstellung müssen Sie regelmäßige Gesamtsicherungen der Datenbank einplanen und ausführen.

In einer Umgebung mit partitionierten Datenbanken ist eine Datenbank auf viele Datenbankpartitions-Server (oder Knoten) verteilt. Sie müssen alle Partitionen wiederherstellen, und die Sicherungsabbilder, die Sie zur Wiederherstellung der Datenbank verwenden, müssen alle zum gleichen Zeitpunkt erstellt worden sein. (Jede Datenbankpartition wird getrennt gesichert und wiederhergestellt.) Eine Sicherung, bei der alle Sicherungs-

kopien der Datenbankpartitionen zur gleichen Zeit erstellt werden, wird als *Versionssicherung* bezeichnet.

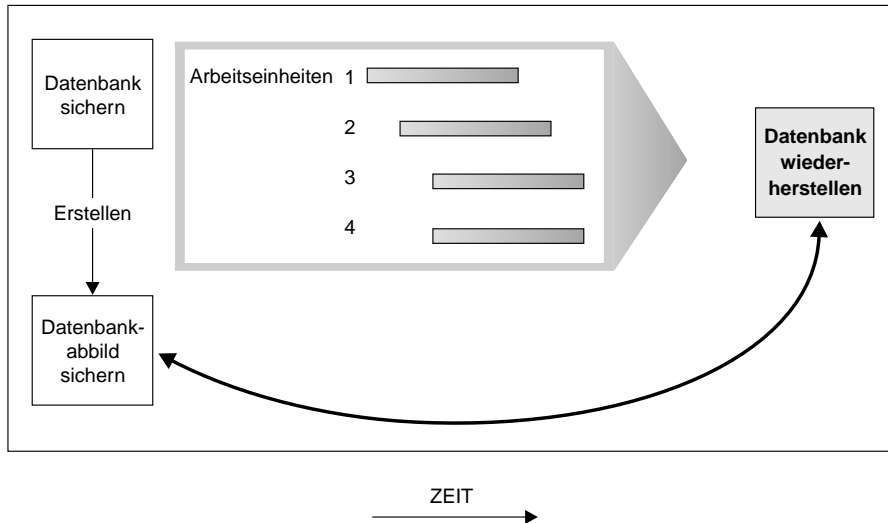


Abbildung 14. Wiederherstellen einer Datenbank mit Hilfe der Sicherungskopie

- Sie können die Methode der *aktualisierenden Wiederherstellung* nur dann einsetzen, wenn Sie eine Sicherungskopie der Datenbank erstellt und die Protokolldateien archiviert haben. (D. h., der Konfigurationsparameter *logretain* und/oder *userexit* der Datenbank ist aktiviert. Informationen zu den Entscheidungen hinsichtlich der zu verwendenden Protokollierprozedur finden Sie in „Datenbankprotokolle“ auf Seite 37.) Das Wiederherstellen der Datenbank unter Angabe der Option *WITHOUT ROLLING FORWARD* (d. h. ohne aktualisierende Wiederherstellung) entspricht der Methode der Versionswiederherstellung. Die Datenbank wird in einem Status wiederhergestellt, der dem Zeitpunkt entspricht, zu dem die Offline-Sicherung erstellt wurde. Wenn Sie die Datenbank wiederherstellen und für die Operation *RESTORE DATABASE* die Option *WITHOUT ROLLING FORWARD* *nicht* angeben, befindet sich die Datenbank nach Abschluß der Wiederherstellung im Status *Aktualisierende Wiederherstellung anstehend*. Dadurch kann die aktualisierende Wiederherstellung ausgeführt werden. Es können zwei Typen der aktualisierenden Wiederherstellung in Erwägung gezogen werden:
 - *Aktualisierende Wiederherstellung einer Datenbank*. Bei dieser Art der aktualisierenden Wiederherstellung werden in den Datenbankprotokollen aufgezeichnete Transaktionen im Anschluß an die Wiederherstellung der Datenbank angewendet (siehe Abb. 15 auf Seite 33). In den Datenbankprotokollen werden alle Änderungen aufgezeichnet, die an der Datenbank vorgenommen werden. Diese Methode vervollständigt die

Wiederherstellung der Datenbank bis zu ihrem Status zu einem bestimmten Zeitpunkt oder bis zu ihrem Status unmittelbar vor dem Fehler (das heißt bis zum Ende der aktiven Protokolle).

In einer Umgebung mit partitionierten Datenbanken ist eine Datenbank über viele Datenbankpartitionen verteilt. Wenn Sie eine aktualisierende Wiederherstellung bis zu einem bestimmten Zeitpunkt ausführen, müssen alle Datenbankpartitionen aktualisierend wiederhergestellt werden, um sicherzustellen, daß alle Partitionen den gleichen Stand haben. Wenn Sie eine einzelne Datenbankpartition wiederherstellen müssen, können Sie eine aktualisierende Wiederherstellung bis zum Ende der Protokolle durchführen, um sie auf den gleichen Stand wie die anderen Partitionen in der Datenbank zu bringen.

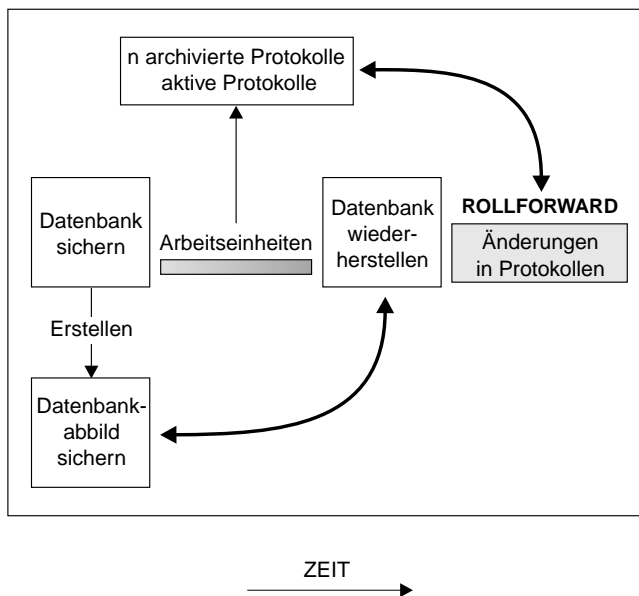


Abbildung 15. Aktualisierende Wiederherstellung einer Datenbank

- **Wiederherstellen und aktualisierende Wiederherstellung eines Tabellenbereichs.** Wenn für eine Datenbank die aktualisierende Wiederherstellung aktiviert ist, können auch Tabellenbereiche gesichert, wiederhergestellt und aktualisierend wiederhergestellt werden. Zur Ausführung der Operation zum Wiederherstellen und aktualisierenden Wiederherstellen eines Tabellenbereichs benötigen Sie ein Sicherungsabbild entweder der gesamten Datenbank (das heißt alle Tabellenbereiche) oder eines oder mehrerer Einzeltabellenbereiche. Außerdem benötigen Sie die Protokollsätze, die die wiederherzustellenden Tabellenbereiche betreffen. Sie können einen Tabellenbereich durch die Protokolle bis zu einem von zwei Punkten aktualisierend wiederherstellen:

- Das Ende der Protokolldateien
- Ein bestimmter Zeitpunkt (als Wiederherstellung *bis zu einem Zeitpunkt* bezeichnet)

Anmerkungen:

1. Die Tabellenbereiche, die zum Zeitpunkt der Sicherungsoperation (backup) nicht ausgewählt werden, befinden sich nach der Wiederherstellung nicht mehr in demselben Status wie die wiederhergestellten Tabellenbereiche.
2. Wenn Sie die aktualisierende Wiederherstellung auf Tabellenbereichsebene ausführen wollen, müssen Sie „Schlüsseltabellenbereiche“ in der Datenbank, die wiederhergestellt werden sollen, identifizieren und regelmäßige Sicherungen der Datenbank (oder der „Schlüsseltabellenbereiche“) terminieren und ausführen.

Eine aktualisierende Wiederherstellung von Tabellenbereichen kann in den beiden folgenden Situationen ausgeführt werden:

- Ein Tabellenbereich befindet sich nach der Wiederherstellung des Tabellenbereichs immer im Status *Aktualisierende Wiederherstellung anstehend* und muß aktualisierend wiederhergestellt werden. Rufen Sie dazu den Befehl `ROLLFORWARD DATABASE` (siehe Handbuch *Command Reference*) auf, um die Protokolle auf die Tabellenbereiche entweder bis zu einem Zeitpunkt oder bis zum Ende der Protokolle anzuwenden.
- Wenn sich ein oder mehrere Tabellenbereiche nach einer Wiederherstellung infolge eines Systemabsturzes im Status *Aktualisierende Wiederherstellung anstehend* befinden, beheben Sie zuerst das Problem mit dem Tabellenbereich. In einigen Fällen ist zur Behebung des Fehlers am Tabellenbereich keine Ausführung einer Operation `RESTORE DATABASE` erforderlich. Zum Beispiel kann ein Netzspannungsverlust den Tabellenbereich in den Status *Aktualisierende Wiederherstellung anstehend* versetzen. Wenn der Fehler vor der Wiederherstellung nach einem Systemabsturz behoben wird, reicht diese Wiederherstellung möglicherweise aus, um die Datenbank in einen konsistenten, verwendbaren Zustand zu bringen. Eine Operation `RESTORE DATABASE` ist in diesem Fall möglicherweise nicht erforderlich. Nachdem das Problem mit dem Tabellenbereich behoben ist, können Sie den Befehl `ROLLFORWARD DATABASE` verwenden, um die Protokolle auf die Tabellenbereiche bis zu einem bestimmten Zeitpunkt oder bis zum Ende der Protokolldateien anzuwenden.

Anmerkung: Wenn der fehlerhafte Tabellenbereich die Systemkatalogtabellen enthält, können Sie die Datenbank nicht starten. Sie müssen den Tabellenbereich `SYSCATSPACE` wiederherstellen und anschließend die aktualisierende Wiederherstellung bis zum Ende der Protokolle ausführen.

Wenn Sie in einer Umgebung mit partitionierten Datenbanken einen Tabellenbereich bis zu *einem bestimmten Zeitpunkt* aktualisierend wiederherstellen, müssen Sie die Liste der Knoten (Datenbankpartitionen) nicht angeben, auf die sich der Tabellenbereich verteilt. DB2 übergibt die Anforderung zur aktualisierenden Wiederherstellung an alle Partitionen. Dies bedeutet, daß der Tabellenbereich auf allen Datenbankpartitionen, auf denen der Tabellenbereich sich befindet, wiederhergestellt werden muß.

Wenn Sie in einer Umgebung mit partitionierten Datenbanken einen Tabellenbereich bis *zum Ende der Protokolle* aktualisierend wiederherstellen, müssen Sie die Liste der Datenbankpartitionen angeben, wenn Sie den Tabellenbereich *nicht* in allen Partitionen aktualisierend wiederherstellen wollen. Wenn Sie alle Tabellenbereiche in allen Partitionen, die sich im Status *Aktualisierende Wiederherstellung anstehend* befinden, bis zum Ende der Protokolle aktualisierend wiederherstellen wollen, müssen Sie die Liste der Datenbankpartitionen nicht angeben. Standardmäßig wird die Anforderung ROLLFORWARD DATABASE an alle Partitionen gesendet.

Faktoren mit Auswirkung auf die Wiederherstellung

Berücksichtigen Sie folgende Faktoren bei der Entscheidung über die zu verwendende Wiederstellungsmethode:

- Muß die Datenbank wiederherstellbar oder nicht wiederherstellbar sein?
- Wie nah müssen Sie beim Wiederherstellen der Datenbank an den Zeitpunkt des Fehlers herankommen (Wiederherstellungspunkt)?
- Wieviel Zeit steht für die Wiederherstellung der Datenbank zur Verfügung? Dazu gehört folgendes:
 - Zeitspanne zwischen den Sicherungen (wirkt sich auf die aktualisierende Wiederherstellung aus)
 - Zeitspanne, in dem die Datenbank verwendbar bzw. für den Zugriff bereit ist (Online- bzw. Offline-Sicherungen entsprechend der erforderlichen Datenverfügbarkeit)
- Wieviel Speicher kann für Sicherungskopien und Archivprotokolldateien zugeordnet werden?
- Werden Sicherungen auf Tabellenbereichsebene oder der gesamten Datenbank vorgenommen?

Allgemein sollte die Strategie zur Wartung und Wiederherstellung der Datenbank sicherstellen, daß alle Informationen verfügbar sind, wenn sie für eine Wiederherstellung der Datenbank benötigt werden. Die Strategie sollte einen regelmäßigen Plan zur Erstellung von Datenbanksicherungen vorsehen und Sicherungen einplanen, wenn eine Datenbank erstellt wird oder, wie im Fall eines Systems mit partitionierten Datenbanken, wenn das System durch Hinzufügen oder Löschen von Datenbankpartitions-Servern (Knoten) skaliert

wird. Über diese Grundanforderungen hinaus sollte eine gute Strategie Elemente enthalten, die die Wahrscheinlichkeit und die Auswirkungen eines Datenbankfehlers verringern.

Unter den folgenden Themen finden Sie weitere Informationen:

- „Wiederherstellbare und nicht wiederherstellbare Datenbanken“
- „Datenbankprotokolle“ auf Seite 37
- „Verringern der Protokollierung für Arbeitstabellen“ auf Seite 42
- „Wiederherstellungspunkt“ auf Seite 44
- „Häufigkeit der Sicherungen und erforderliche Zeit“ auf Seite 44
- „Zur Wiederherstellung benötigte Zeit“ auf Seite 46
- „Überlegungen zum Speicherbedarf“ auf Seite 46
- „Zusammenhalten zusammengehöriger Daten“ auf Seite 48
- „Einschränkungen für die Verwendung verschiedener Betriebssysteme“ auf Seite 48
- „Wiederherstellen beschädigter Tabellenbereiche“ auf Seite 49
- „Überlegungen zur Wiederherstellungsleistung“ auf Seite 51

Obwohl das Hauptgewicht dieses Abschnitts auf der Datenbank liegt, sollte Ihre Gesamtplanung zur Wiederherstellung auch die Wiederherstellung folgender Komponenten berücksichtigen:

- Das Betriebssystem und die ausführbaren DB2-Dateien
- Anwendungen, UDFs und den Code gespeicherter Prozeduren in Bibliotheken des Betriebssystems
- Befehle zur Erstellung von DB2-Exemplaren und Nicht-DB2Ressourcen
- Die Sicherheit des Betriebssystems
- Ladekopien aus einer LOAD-Operation (bei Angabe von COPY YES im Befehl LOAD)

Wiederherstellbare und nicht wiederherstellbare Datenbanken

Lassen sich Ihre Daten problemlos erneut erstellen, kann die Datenbank, die diese Daten enthält, als nicht wiederherstellbare Datenbank betrieben werden. Zum Beispiel:

- Es kann sinnvoll sein, Tabellen mit Daten einer externen Quelle, auf die Ihre Anwendungen lediglich zu Lesezwecken zugreifen (und die nicht mit vorhandenen Daten gemischt werden), in einer nicht wiederherstellbaren Datenbank zu speichern.
- Tabellen mit kleinen Datenmengen. In diesem Fall stellt die Wiederherstellung kein Problem dar. Es werden nicht genügend Protokollierungsaktivitäten für die Daten durchgeführt, um die zusätzliche Komplexität der

Verwaltung von Protokolldateien und der aktualisierenden Wiederherstellung nach einer Wiederherstellung von der Sicherungskopie zu rechtfertigen.

- Umfangreiche Tabellen, die in regelmäßigen Abständen durch eine geringe Anzahl von Zeilen ergänzt werden. Auch in diesem Fall sind die möglichen Verluste zu gering, um eine Verwaltung von Protokolldateien und eine aktualisierende Wiederherstellung im Anschluß an eine Wiederherstellungsoperation zu rechtfertigen.

Können Ihre Daten nicht problemlos erneut erstellt werden, sollte die Datenbank, die diese Daten enthält, als wiederherstellbare Datenbank betrieben werden. Nachfolgend sind Beispiele für Daten aufgeführt, die in einer wiederherstellbaren Datenbank gespeichert werden sollten:

- Daten, die Sie nicht erneut erstellen können. Hierzu gehören Daten, deren Quelle nach dem Laden der Daten zerstört wird, sowie Daten, die manuell in Tabellen eingegeben werden.
- Daten, die nach dem Laden in die Datenbank von Anwendungsprogrammen oder Workstation-Benutzern geändert werden.

Datenbankprotokolle

Jeder Datenbank sind Protokolldateien zugeordnet. In diesen Protokolldateien werden die Datenbankänderungen aufgezeichnet. Wenn eine Datenbank über den Stand der letzten vollständigen Offline-Sicherung hinaus wiederhergestellt werden muß, sind Protokolle erforderlich, um die Datenbank bis zu dem Zeitpunkt des Fehlers aktualisierend wiederherstellen zu können.

DB2 stellt zwei Arten von Protokollierung zur Verfügung: *Umlaufprotokolle* und *Archivprotokolle*. Jede Art stellt eine andere Stufe der Wiederherstellbarkeit bereit.

Umlaufprotokolle bilden die Standardprotokollierung, wenn eine neue Datenbank erstellt wird. Bei dieser Art der Protokollierung sind nur vollständige Offline-Sicherungen der Datenbank gültig. Wie der Name bereits andeutet, bilden Umlaufprotokolle einen „Ring“ von Online-Protokollen, die eine Wiederherstellung nach Transaktionsfehlern oder Systemabstürzen ermöglichen. Die Protokolle werden nur so lange verwendet und behalten, wie es erforderlich ist, um die Integrität der aktuellen Transaktionen zu gewährleisten. Bei der Verwendung von Umlaufprotokollen ist es nicht möglich, eine Datenbank durch frühere Transaktionen, die seit der letzten Gesamtsicherung durchgeführt wurden, aktualisierend wiederherzustellen. Die Wiederherstellung nach Datenträgerausfällen und Systembeschädigungen erfolgt mit Hilfe einer vollständigen Offline-Sicherung. Alle Änderungen, die seit der letzten Sicherung vorgenommen wurden, gehen verloren. Die Datenbank muß offline sein (d. h. für Benutzer nicht zugänglich), wenn eine Gesamtsicherung

erstellt wird. Da diese Art der Wiederherstellung die Daten auf dem Stand des bestimmten Zeitpunkts der Gesamtsicherung wiederherstellt, wird sie als *Versionswiederherstellung* bezeichnet.

Abb. 16 zeigt, daß die aktive Protokolldatei mit einem Ring aus Protokoll-dateien arbeitet, wenn die Umlaufprotokollierung aktiv ist.

Aktive Protokolldateien werden von der Wiederherstellung nach einem

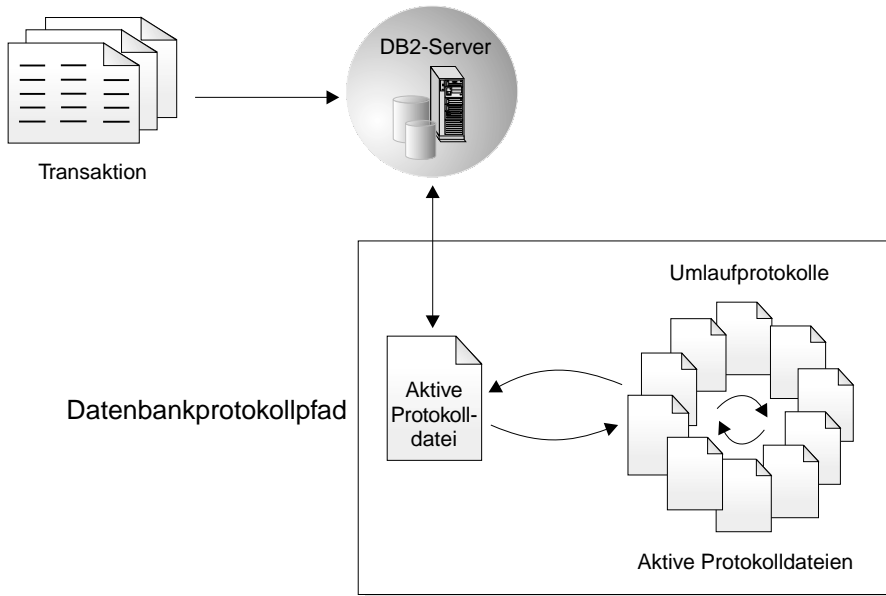


Abbildung 16. Umlaufprotokollierung

Systemabsturz verwendet, um zu verhindern, daß eine Datenbank nach einer Störung (Stromausfall oder Anwendungsfehler) in einem inkonsistenten Status zurückbleibt. Der Befehl `RESTART DATABASE` verwendet die aktiven Protokolle bei Bedarf, um die Datenbank in einen konsistenten und brauchbaren Status zu versetzen. Während einer Wiederherstellung nach einem Systemabsturz werden in diesen Protokollen aufgezeichnete Änderungen, die aufgrund der Störung nicht festgeschrieben wurden, rückgängig gemacht. Festgeschriebene, aber nicht physisch aus dem Hauptspeicher (Pufferpool) auf Platte (Datenbankbehälter) geschriebene Änderungen werden wiederholt. Durch diese Maßnahmen wird die Integrität der Datenbank gewährleistet. Der Befehl `ROLLFORWARD DATABASE` kann bei einer Wiederherstellung bis zu einem bestimmten Zeitpunkt oder bis zum Ende der Protokolle ebenfalls die aktiven Protokolle (sofern erforderlich) verwenden. Aktive Protokolldateien befinden sich im Verzeichnispfad der Datenbankprotokolle.

Archivierte Protokolldateien werden besonders für die aktualisierende Wiederherstellung verwendet. Dazu gehören folgende Arten von Dateien:

Online archivierte Protokolldateien

Wenn die Änderungen im aktiven Protokoll für die normale Verarbeitung nicht mehr benötigt werden, wird das Protokoll geschlossen und wird somit zu einem archivierten Protokoll. Ein archiviertes Protokoll wird als *online* bezeichnet, wenn es im Verzeichnispfad der Datenbankprotokolle gespeichert ist (siehe Abb. 17).

Offline archivierte Protokolldateien

Eine archivierte Protokolldatei wird als *offline* bezeichnet, wenn sie sich nicht mehr im Verzeichnispfad der Datenbankprotokolle befindet ist (siehe Abb. 18 auf Seite 40). Sie haben auch die Möglichkeit, mit Hilfe eines Benutzer-Exit-Programms archivierte Protokolle an einer anderen Speicherposition als im Verzeichnispfad der Datenbankprotokolle zu speichern. (Weitere Informationen finden Sie in "Benutzer-Exit zur Datenbankwiederherstellung" im Band *Systemverwaltung: Implementierung*.)

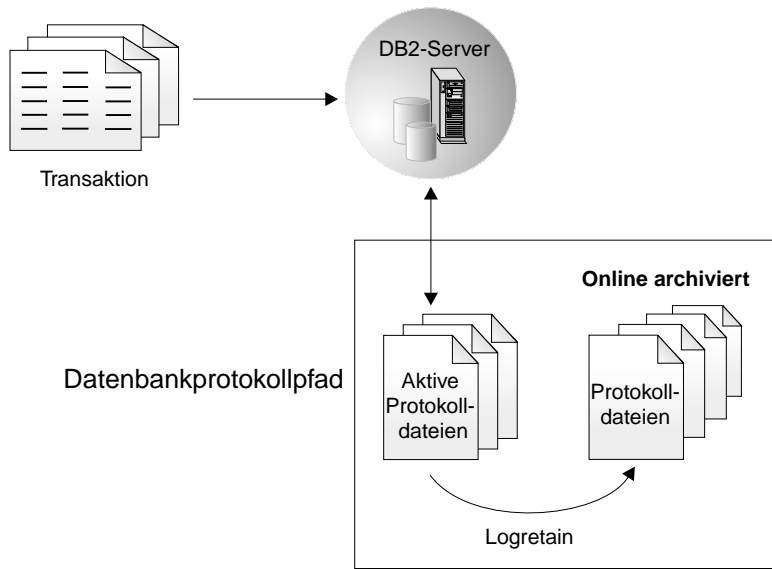


Abbildung 17. Online archivierte Protokolle

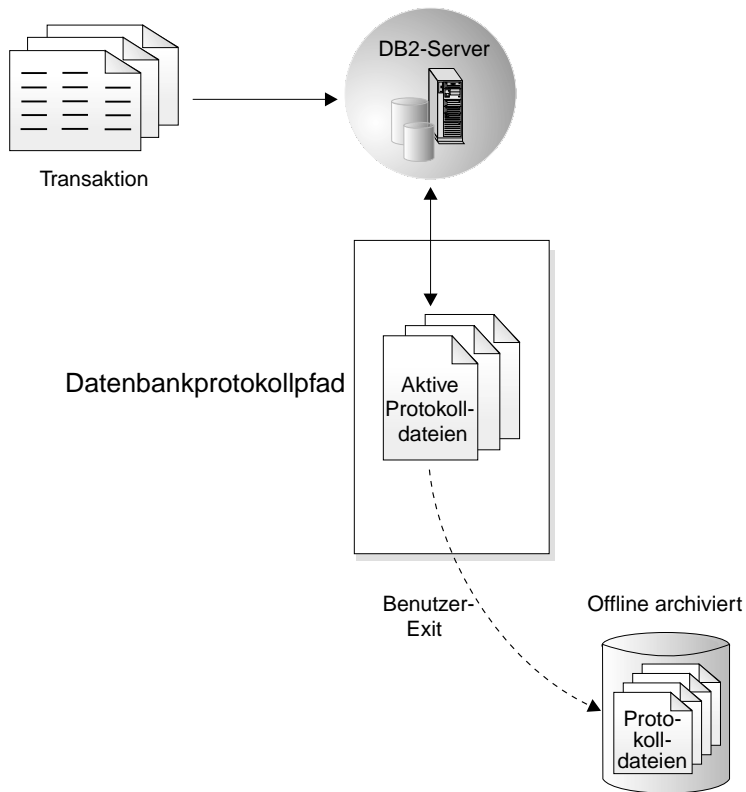


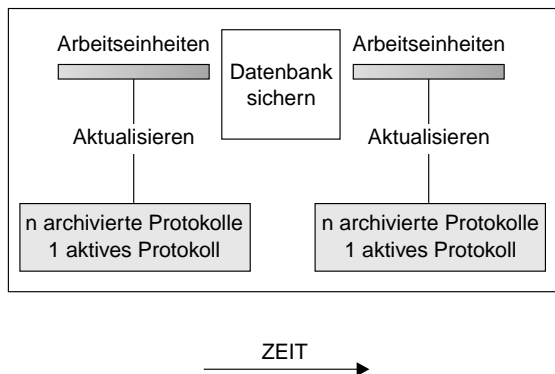
Abbildung 18. Offline archivierte Protokolle

Von der aktualisierenden Wiederherstellung können sowohl archivierte als auch aktive Protokolldateien verwendet werden, um eine Datenbank entweder bis zum Ende der Protokolle oder bis zu einem bestimmten Zeitpunkt wiederherzustellen. Die Funktion zur aktualisierenden Wiederherstellung erreicht dies, indem sie die in den archivierten und aktiven Protokolldateien gefunden festgeschriebenen Änderungen erneut auf die wiederhergestellte Datenbank anwendet.

Die aktualisierende Wiederherstellung kann mit Hilfe von Protokollen auch einen Tabellenbereich wiederherstellen, indem festgeschriebene Aktualisierungen in den archivierten und aktiven Protokolldateien erneut angewendet werden. Sie können einen Tabellenbereich bis zum Ende der Protokolle oder bis zu einem bestimmten Zeitpunkt wiederherstellen.

Während einer Online-Sicherung werden alle Aktivitäten an der Datenbank protokolliert. Wenn eine Online-Sicherung wiederhergestellt wird, muß mit Hilfe der Protokolle die Datenbank mindestens bis zu dem Zeitpunkt aktualisierend wiederhergestellt werden, zu dem die Sicherung abgeschlossen wurde.

Damit dies geschehen kann, müssen Sie die Protokolle archivieren und bereitstellen, wenn die Datenbank wiederhergestellt werden soll. Die Protokolldatei, die zum Zeitpunkt der Sicherung verwendet wurde, bleibt möglicherweise auch nach Abschluß der Sicherungsoperation noch geöffnet. Mit Hilfe der Option FLUSH LOG für die Online-Sicherung im Befehl BACKUP DATABASE wird die aktive Protokolldatei geschlossen, wenn die Online-Sicherung beendet ist. Dadurch kann die aktive Protokolldatei archiviert werden, sodaß Sie über eine vollständige Sicherung und sämtliche, für eine Wiederherstellung dieser Sicherung erforderlichen Protokolle verfügen.



Protokolle werden zwischen Sicherungen verwendet, um die Änderungen an den Datenbanken zu protokollieren.

Abbildung 19. Aktive und archivierte Datenbankprotokolle bei aktualisierender Wiederherstellung

Mit Hilfe zweier Konfigurationsparameter der Datenbank können Sie die Position ändern, an der die archivierten Protokolldateien gespeichert werden sollen: *newlogpath* und *userexit*. Änderungen am Parameter *newlogpath* wirken sich außerdem auf die Speicherposition aus, an der die aktiven Protokolle gespeichert werden. Weitere Informationen zu diesen Konfigurationsparametern finden Sie im Handbuch *Systemverwaltung: Optimierung*.

Welche *Protokollspeicherbereiche* (siehe „Behälter“ auf Seite 20) im Verzeichnispfad der Datenbankprotokolle archiviert Protokolldateien sind, können Sie am Wert des Konfigurationsparameters *loghead* der Datenbank ablesen. Dieser Parameter gibt das Protokoll mit der niedrigsten Nummer an, das aktiv ist. Bei Protokollen mit Folge-nummern, die niedriger als *loghead* sind, handelt es sich um archiviert Protokolldateien, die versetzt werden können. Die Werte dieser Parameter können Sie überprüfen, indem Sie die Steuerzentrale verwenden oder mit Hilfe des Befehlszeilenprozessors den Befehl GET DATABASE CONFIGURATION ausführen, um die „erste aktive Protokolldatei“

anzuzeigen. Weitere Informationen zu diesem Konfigurationsparameter finden Sie im Handbuch *Systemverwaltung: Optimierung*.

Anmerkungen:

1. Wenn Sie ein aktives Protokoll löschen, wird die Datenbank unbrauchbar. Eine Weiterverwendung der Datenbank ist dann nur möglich, wenn Sie die Datenbank wiederherstellen. Außerdem können Sie eine aktualisierende Wiederherstellung nur bis zum ersten gelöschten Protokoll durchführen.
2. Wenn Sie befürchten, daß Ihre aktiven Protokolle beschädigt werden könnten (aufgrund eines Plattenfehlers), sollten Sie ein zeitgleiches Spiegeln der Datenträger, auf denen die Protokolle gespeichert sind, in Betracht ziehen.

Verringern der Protokollierung für Arbeitstabellen

Wenn Ihre Anwendung Arbeitstabellen aus Originaltabellen erstellt und mit Werten füllt und Sie sich um die Wiederherstellbarkeit dieser Arbeitstabellen keine Gedanken machen müssen, weil sie leicht mit Hilfe der Originaltabellen wieder erstellt werden können, können Sie die Arbeitstabellen unter Angabe der Klausel NOT LOGGED INITIALLY in der Anweisung CREATE TABLE erstellen. Die Verwendung des Parameters NOT LOGGED INITIALLY hat den Vorteil, daß alle Änderungen an der Tabelle (einschließlich Operationen zum Einfügen, Löschen, Aktualisieren sowie Erstellen von Indizes) innerhalb derselben Arbeitseinheit, in der die Tabelle erstellt wird, nicht protokolliert werden. Dadurch werden nicht nur die Protokollieraktivitäten verringert, sondern auch eine bessere Leistung für Ihre Anwendung erzielt. Sie können das gleiche Ergebnis für vorhandene Tabellen erreichen, indem Sie die Anweisung ALTER TABLE mit dem Parameter NOT LOGGED INITIALLY verwenden.

Anmerkungen:

1. Sie können mehrere Tabellen mit der Klausel NOT LOGGED INITIALLY innerhalb derselben Arbeitseinheit erstellen.
2. Änderungen an den Katalogtabellen und anderen Benutzertabellen werden weiterhin protokolliert.

Da die Änderungen der Tabelle nicht protokolliert werden, sollten Sie bei der Entscheidung, die Klausel NOT LOGGED INITIALLY zu verwenden, folgendes beachten:

- Alle Änderungen an der Tabelle müssen zum Festschreibzeitpunkt auf Platte geschrieben werden. Das bedeutet, daß die COMMIT-Operation länger dauern kann.
- Ein Fehler, der für irgendeine Operation in der Arbeitseinheit, in der die Tabelle erstellt wird, zurückgegeben wird, führt dazu, daß die gesamte Arbeitseinheit rückgängig gemacht wird (SQLCODE -1476, SQLSTATE 40506).

- Diese Tabellen können Sie bei einer aktualisierenden Wiederherstellung nicht wiederherstellen. Wenn die aktualisierende Wiederherstellung eine Tabelle antrifft, die mit der Option NOT LOGGED INITIALLY erstellt wurde, wird diese Tabelle als nicht verfügbar markiert. Nach der Wiederherstellung der Datenbank führt jeder Versuch, auf die Tabelle zuzugreifen, zur Rückgabe der Nachricht SQL1477N.

Anmerkung: Bei der Erstellung einer Tabelle werden Zeilensperren für die Katalogtabellen aktiviert, bis eine COMMIT-Operation ausgeführt wird. Die Inaktivierung der Protokollfunktion ist nur dann von Vorteil, wenn Sie die Tabelle innerhalb derselben Arbeitseinheit mit Werten füllen, in der sie erstellt wird. Daraus ergeben sich Konsequenzen für den gemeinsamen Zugriff. Weitere Informationen finden Sie im Abschnitt über gemeinsamen Zugriff im Handbuch *Systemverwaltung: Optimierung*.

Weitere Informationen zum Erstellen von Tabellen finden Sie im Handbuch *SQL Reference*.

Wenn Sie planen, mit deklarierten temporären Tabellen zu arbeiten, ist folgendes zu beachten:

- Deklarierte temporäre Tabellen werden nicht in den Katalogen erstellt. Daher werden keine Sperren aktiviert.
- Für deklarierte temporäre Tabellen findet keine Protokollierung statt, auch nicht nach der ersten COMMIT-Operation.
- Geben Sie die Option ON COMMIT PRESERVE an, um die Zeilen in der Tabelle nach einer COMMIT-Operation zu behalten. Ansonsten werden alle Zeile gelöscht.
- Nur die Anwendung, durch die die deklarierte temporäre Tabelle erstellt, kann auf dieses Exemplar der Tabelle zugreifen.
- Die Tabelle wird implizit gelöscht, wenn die Verbindung der Anwendung zur Datenbank beendet wird.
- Fehler bei der Operation während einer Arbeitseinheit mit einer deklarierten temporären Tabelle bewirken nicht, daß die Arbeitseinheit vollständig rückgängig gemacht wird. Ein Fehler bei der Operation in einer Anweisung, die den Inhalt einer deklarierten temporären Tabelle ändert, führt jedoch dazu, daß alle Zeilen in dieser Tabelle gelöscht werden. Eine ROLLBACK-Operation der Arbeitseinheit (bzw. eines Sicherungspunkts) löscht alle Zeilen in deklarierten temporären Tabellen, die innerhalb dieser Arbeitseinheit (bzw. dieses Sicherungspunkts) geändert wurden.

Weitere Informationen zu deklarierten temporären Tabellen und ihren Einschränkungen finden Sie im Abschnitt zur Anweisung DECLARE GLOBAL TEMPORARY TABLE des Handbuchs *SQL Reference*.

Wiederherstellungspunkt

Die Methoden der Versionswiederherstellung und der aktualisierenden Wiederherstellung stellen unterschiedliche Wiederherstellungspunkte zur Verfügung. Bei der Versionsmethode ist eine Offline-Sicherung der gesamten Datenbank zu festgesetzten Zeiten erforderlich. Bei dieser Methode ist die wiederhergestellte Datenbank nur so aktuell, wie die Sicherungskopie, die wiederhergestellt wird. Wenn Sie beispielsweise am Ende jedes Tages eine Sicherungskopie anlegen und die Datenbank in der Mitte des folgenden Tages beschädigt wird, verlieren Sie die Änderungen eines halben Tages.

Bei der aktualisierenden Wiederherstellung werden die Änderungen, die an der Datenbank vorgenommen werden, in Protokolldateien aufgezeichnet. Bei dieser Methode stellen Sie zunächst die Datenbank bzw. die Tabellenbereiche mit Hilfe einer Sicherungskopie wieder her. Anschließend verwenden Sie die Protokolldateien, um die Änderungen erneut anzuwenden, die seit dem Erstellen der letzten Sicherungskopie an der Datenbank vorgenommen wurden.

Wenn die aktualisierende Wiederherstellung aktiviert ist, können Sie die Vorteile der Online-Sicherung und der Sicherung auf Tabellenbereichsebene nutzen. Bei einer aktualisierenden Wiederherstellung der gesamten Datenbank und aller Tabellenbereiche haben Sie die Wahl, ob die Wiederherstellung bis zum Ende der Protokolle oder bis zu einem bestimmten Zeitpunkt erfolgen soll. Wenn z. B. eine Anwendung die Datenbank beschädigt, könnten Sie von einer wiederhergestellten Kopie der Datenbank ausgehend die Änderungen bis kurz vor den Zeitpunkt, an dem die Anwendung gestartet wurde, aktualisierend wiederherstellen. Keine Arbeitseinheiten, die nach dem angegebenen Zeitpunkt in die Protokolle geschrieben wurden, werden erneut angewendet.

Sie können auch Tabellenbereiche bis zum Ende der Protokolle oder bis zu einem bestimmten Zeitpunkt aktualisierend wiederherstellen.

Häufigkeit der Sicherungen und erforderliche Zeit

In Ihrem Wiederherstellungsplan sollten regelmäßige Sicherungen vorgesehen sein, da das Sichern einer Datenbank Zeit und Systemressourcen in Anspruch nimmt.

Sie sollten in regelmäßigen Abständen Gesamtsicherungen der Datenbanken erstellen, selbst wenn Sie die Protokolle archivieren (die eine aktualisierende Wiederherstellung ermöglichen). Wenn in Ihrer Wiederherstellungsstrategie die aktualisierende Wiederherstellung vorgesehen ist, bedeutet eine erst kürzlich erstellte Gesamtsicherung der Datenbank, daß es weniger archivierte Protokolle gibt, die auf die Datenbank anzuwenden sind, und daß infolgedessen das Dienstprogramm ROLLFORWARD zur Wiederherstellung der Datenbank weniger Zeit benötigt.

Außerdem sollten Sie in Betracht ziehen, die Sicherungen und Protokoll-dateien nicht zu überschreiben, sondern eher mehrere Gesamtsicherungen der Datenbank mit zugehörigen Protokollen als weitere Vorsichtsmaßnahme zu erstellen.

Wenn die erforderliche Zeit für die Anwendung der Archivprotokolldateien bei der Wiederherstellung und der aktualisierenden Wiederherstellung einer sehr aktiven Datenbank wichtig ist, sollten Sie den Aufwand häufigerer Datenbanksicherungen berücksichtigen. Dadurch verringert sich die Anzahl der Archivprotokolldateien, die Sie bei einer aktualisierenden Wiederherstellung anwenden müssen.

Sie können eine Sicherung durchführen, während die Datenbank *online* oder *offline* ist. Wenn die Datenbank online ist, können andere Anwendungen oder Prozesse weiterhin Verbindungen zur Datenbank herstellen sowie Daten lesen und ändern, während die Sicherungsoperation ausgeführt wird. Bei einer offline durchgeführten Sicherung kann nur die Sicherungsoperation mit der Datenbank verbunden sein. Die übrige Organisation kann keine Verbindung zur Datenbank herstellen, solange die Sicherungsoperation ausgeführt wird.

Um den Zeitraum, in dem die Datenbank nicht verfügbar ist, möglichst kurz zu halten, sollten Sie Online-Sicherungen in Betracht ziehen. Online-Sicherungen werden nur unterstützt, wenn die aktualisierende Wiederherstellung aktiviert ist. Wenn die aktualisierende Wiederherstellung aktiviert ist und Sie über einen kompletten Satz von Protokollen verfügen, können Sie die Datenbank im Bedarfsfall rekonstruieren.

Anmerkungen:

1. Sie können eine Online-Sicherung nur dann verwenden, wenn Sie über die Datenbankprotokolle verfügen, die den Zeitraum für die Sicherungsoperation umfassen.
2. Offline-Sicherungen sind schneller als Online-Sicherungen.

Wenn eine Datenbank große Mengen von Daten der Typen LONG und LOB enthält, kann das Sichern der Datenbank sehr viel Zeit in Anspruch nehmen. Der Befehl BACKUP bietet die Möglichkeit, ausgewählte Tabellenbereiche zu sichern. Wenn Sie DMS-Tabellenbereiche verwenden, können Sie verschiedene Arten von Daten in speziellen Tabellenbereichen speichern, um die für Sicherungsoperationen benötigte Zeit zu verringern. Sie können die Tabellendaten in einem Tabellenbereich, die Langfeld- und LOB-Daten in einem anderen Tabellenbereich und die Indizes in einem dritten Tabellenbereich unterbringen. Wenn sich die Langfeld- und LOB-Daten in separaten Tabellenbereichen befinden, läßt sich der für das Sichern von Daten erforderliche Zeitaufwand dadurch verringern, daß die Tabellenbereiche mit den Langfeld- und LOB-Daten von der Sicherung ausgeschlossen werden. Handelt es sich bei Ihren LONG- und LOB-Daten um wichtige Unternehmensdaten,

sollten Sie das Sichern dieser Tabellenbereiche gegen den für die Wiederherstellungsoperation erforderlichen Zeitaufwand abwägen. Wenn die LOB-Daten von einer getrennten Quelle reproduziert werden können, sollten Sie beim Erstellen oder Ändern einer Tabelle zum Hinzufügen von LOB-Spalten die Option NOT LOGGED verwenden.

Wenn Sie eine Tabelle reorganisieren, sollten Sie die betroffenen Tabellenbereiche nach Beendigung der Operation sichern. Wenn dann der Fall eintritt, daß die Tabellenbereiche wiederhergestellt werden müssen, muß die Datenreorganisation nicht mehr aktualisierend wiederhergestellt werden.

Anmerkung: Wenn Sie einen Tabellenbereich sichern, der nicht alle Tabellendaten enthält, können Sie keine aktualisierende Wiederherstellung bis zu einem bestimmten Zeitpunkt für diesen Tabellenbereich ausführen. Alle Tabellenbereiche, die irgendeine Art von Daten für eine Tabelle enthalten, müssen gleichzeitig bis zum selben Zeitpunkt aktualisierend wiederhergestellt werden.

Zur Wiederherstellung benötigte Zeit

Die Zeit, die zur Wiederherstellung einer Datenbank benötigt wird, setzt sich aus zwei Perioden zusammen: die Zeit, die zur Wiederherstellung der Sicherungskopie benötigt wird, sowie, wenn für die Datenbank die aktualisierende Wiederherstellung aktiviert ist, die Zeit, die zur Anwendung der Protokolle während der aktualisierenden Wiederherstellung benötigt wird. Bei der Formulierung eines Wiederherstellungsplans sollten Sie diesen Wiederherstellungsaufwand und die dadurch verursachte Auswirkung auf den Geschäftsbetrieb berücksichtigen. Durch Testen des Gesamtwiederherstellungsplans können Sie ermitteln, ob die Zeit, die zur Wiederherstellung der Datenbank benötigt wird, in Anbetracht Ihrer Geschäftserfordernisse angemessen ist. Nach einem Test können Sie zu dem Schluß kommen, daß es vorteilhaft ist, die Häufigkeit der Sicherungen zu erhöhen. Wenn Ihre Strategie eine aktualisierende Wiederherstellung vorsieht, wird dieses Vorgehen die Anzahl der zwischen Sicherungen archivierten Protokolle reduzieren und auf diese Weise die für eine aktualisierende Wiederherstellung der Datenbank benötigte Zeit nach einer Wiederherstellung von der Sicherungskopie verringern.

Anmerkung: Die Einstellung des Konfigurationsparameters des Datenbankmanagers zur Aktivierung der partitionsinternen Parallelität (*intra_parallel*) wirkt sich auf die Leistung der Sicherungsoperationen und der Wiederherstellungsoperationen nicht aus. Es werden für beide Operationen unabhängig von der Einstellung des Parameters *intra_parallel* mehrere Prozesse verwendet.

Überlegungen zum Speicherbedarf

Bei der Entscheidung der zu verwendenden Wiederherstellungsmethode sollten Sie auch den erforderlichen Speicherplatz in Betracht ziehen.

Bei der Versionswiederherstellung wird Speicherplatz für die Sicherungskopie der Datenbank und für die wiederhergestellte Datenbank benötigt. Bei einer aktualisierenden Wiederherstellung wird Speicherplatz für die Sicherungskopie der Datenbank bzw. Tabellenbereiche, für die wiederhergestellte Datenbank sowie für die archivierten Datenbankprotokolle benötigt.

Enthält eine Tabelle LONG- oder LOB-Spalten, ist es u. U. ratsam, die betreffenden Daten in einen separaten Tabellenbereich zu stellen. Dies hat Auswirkungen auf Ihre Überlegungen zum Speicherbedarf sowie auf Ihren Wiederherstellungsplan. Wenn Sie einen separaten Tabellenbereich für LONG- und LOB-Daten verwenden und Ihnen der Zeitaufwand für das Sichern der LONG- und LOB-Daten bekannt ist, können Sie einen Wiederherstellungsplan verwenden, bei dem dieser Tabellenbereich nur gelegentlich gesichert wird. Beim Erstellen oder Ändern einer Tabelle, die LOB-Daten umfaßt, können Sie zudem angeben, daß Änderungen an den betreffenden Spalten nicht protokolliert werden sollen. Dadurch verringert sich die Größe der Protokolldateien und somit der erforderliche Protokollarchivierungsspeicher.

Die Sicherungskopie eines SMS-Tabellenbereichs mit LOB-Daten kann größer werden als der ursprüngliche Tabellenbereich. Die Sicherungskopie kann je nach Größe der LOB-Daten im Tabellenbereich bis zu 40% größer werden. Wenn Sie beispielsweise eine Sicherung eines 1 GB großen Tabellenbereichs (mit LOB-Daten) erstellen, benötigen Sie zur Wiederherstellung des Tabellenbereichs mehr als 1 GB Plattenspeicherplatz. Dies tritt nur bei Dateisystemen auf, die die Zuordnung von Dateien mit freien Bereichen (Sparse Allocation) unterstützen (z. B. auf UNIX basierende Betriebssysteme).

Um zu verhindern, daß ein Datenträgerfehler die Datenbank beschädigt und Ihnen die Wiederherstellung unmöglich macht, sollten Sie die Sicherungskopie der Datenbank, die Datenbankprotokolle sowie die Datenbank selbst auf unterschiedlichen Einheiten speichern. Aus diesem Grund wird ausdrücklich empfohlen, nach dem Erstellen der Datenbank die Datenbankprotokolle mit Hilfe des Konfigurationsparameters *newlogpath* auf eine separate Einheit umzuleiten. (Eine Beschreibung dieses sowie weiterer Konfigurationsparameter, die für das Protokollieren relevant sind, finden Sie in "Aktualisierendes Wiederherstellen von Änderungen in einer Datenbank" im Handbuch *Systemverwaltung: Implementierung*.)

Die Datenbankprotokolle können viel Speicherplatz in Anspruch nehmen. Wenn Sie beabsichtigen, eine aktualisierende Wiederherstellung auszuführen, müssen Sie sich daher für eine Verwaltung der archivierten Protokolle entscheiden. Ihnen stehen folgende Möglichkeiten zur Auswahl:

- Stellen Sie genügend Speicher im Verzeichnispfad der Datenbankprotokolle bereit, um die Protokolldateien aufzunehmen.

- Kopieren Sie die Protokolldateien manuell auf eine Speichereinheit bzw. in ein Verzeichnis, die/das nicht mit dem Verzeichnispfad der Datenbankprotokolle übereinstimmt, nachdem sie sich nicht mehr in der aktiven Protokollgruppe befinden.
- Kopieren Sie diese Protokolle mit Hilfe eines Benutzer-Exit-Programms auf eine andere Speichereinheit in Ihrer Umgebung. (Weitere Informationen finden Sie in "Benutzer-Exit zur Datenbankwiederherstellung" im Handbuch *Systemverwaltung: Implementierung*.)

Anmerkung: Unter OS/2 unterstützt DB2 ein Benutzer-Exit-Programm, das die Speicherung der Sicherungsabbilder und Protokolle von Datenbanken auf Standard- und Nicht-Standardeinheiten verwaltet. Weitere Informationen finden Sie in "Benutzer-Exit zur Datenbankwiederherstellung" im Handbuch *Systemverwaltung: Implementierung*.

Zusammenhalten zusammengehöriger Daten

Aufgrund Ihres Datenbankentwurfs sind Sie mit den Abhängigkeiten vertraut, die zwischen Tabellen bestehen. Diese Abhängigkeiten können auf Anwendungsebene ausgedrückt werden, wenn Transaktionen mehr als eine Tabelle aktualisieren, oder auf Datenbankebene, wenn es referentielle Integritätsbedingungen zwischen Tabellen gibt oder wenn sich Auslöser für eine Tabelle auf eine andere Tabelle auswirken. Diese Abhängigkeiten sollten bei der Entwicklung eines Wiederherstellungsplans berücksichtigt werden. Wahrscheinlich ist es sinnvoll, voneinander abhängige Mengen von Daten gemeinsam zu sichern. Solche zusammengehörigen Datenmengen können entweder auf Tabellenbereichsebene oder auf Datenbankebene eingerichtet werden. Wenn die zusammengehörigen Datenmengen zusammen gesichert werden, können sie bis zu einem Punkt wiederhergestellt werden, an dem alle Daten konsistent sind. Dies ist besonders wichtig, wenn Sie in der Lage sein wollen, eine aktualisierende Wiederherstellung bis zu einem bestimmten Zeitpunkt für Tabellenbereiche durchzuführen.

Einschränkungen für die Verwendung verschiedener Betriebssysteme

Wenn Sie in einer Umgebung mit mehreren Betriebssystemen arbeiten, müssen Sie berücksichtigen, daß die Sicherungs- und Wiederherstellungspläne nicht integriert werden können. Dies bedeutet, daß Sie den Befehl BACKUP DATABASE nicht unter dem einen Betriebssystem und den Befehl RESTORE DATABASE unter einem anderen Betriebssystem ausführen können. Sie sollten die Wiederherstellungspläne für die einzelnen Betriebssysteme getrennt halten und unabhängig voneinander ausführen.

Wenn Sie Tabellen von einem Betriebssystem auf ein anderes verschieben müssen, verwenden Sie den Befehl **db2move**, oder verwenden Sie den Befehl EXPORT und dann den Befehl IMPORT oder LOAD. Weitere Informationen finden Sie im Handbuch *Versetzen von Daten Dienstprogramme und Referenz*.

Wiederherstellen beschädigter Tabellenbereiche

Ein beschädigter Tabellenbereich enthält einen oder mehrere Behälter, auf den/die nicht zugegriffen werden kann. Die wird häufig durch Datenträgerprobleme verursacht, die entweder permanent (z. B. eine defekte Platte) oder temporär sind (z. B. eine Platte, die offline ist, oder ein abgehängtes Dateisystem).

Wenn der beschädigte Tabellenbereich der Bereich für die Systemkatalogtabellen ist, kann die Datenbank nicht neu gestartet werden. Wenn die Behälterprobleme nicht behoben werden können, ohne daß die ursprünglichen Daten erhalten bleiben, sind nur die folgenden Optionen verfügbar:

- Die Datenbank wiederherstellen
- Den Katalogtabellenbereich wiederherstellen

Anmerkung: Die Wiederherstellung des Tabellenbereichs ist nur für wiederherstellbare Datenbanken zulässig, da die Datenbank aktualisierend wiedergeherstellt werden muß.

Wenn der beschädigte Tabellenbereich kein Bereich für eine Systemkatalogtabelle ist, versucht DB2 so viel von der Datenbank wie möglich verfügbar zu machen. Der Erfolg hängt in diesem Fall von der Protokollstrategie ab.

Wenn der beschädigte Tabellenbereich der einzige Tabellenbereich für temporäre Tabellen ist, müssen Sie einen neuen Tabellenbereich für temporäre Tabellen erstellen, sobald eine Verbindung zur Datenbank hergestellt ist. Nach der Erstellung kann der neue Tabellenbereich für temporäre Tabellen verwendet werden, und der normale Datenbankbetrieb, der einen Tabellenbereich für temporäre Tabellen erfordert, kann fortgesetzt werden. Den offline befindlichen Tabellenbereich für temporäre Tabellen können Sie löschen, wenn Sie es wünschen. Für die Reorganisation von Tabellen, die mit einem Tabellenbereich für temporäre Systemtabellen arbeitet, sind die folgenden speziellen Überlegungen zu berücksichtigen:

- Wenn der Konfigurationsparameter *indexrec* der Datenbank oder des Datenbankmanagers auf RESTART gesetzt ist, müssen alle ungültigen Indizes während der Datenbankaktivierung neu erstellt werden. Dies schließt Indizes aus einer Reorganisation ein, die während der Erstellungsphase (Build) abgestürzt ist.
- Wenn in einem beschädigten Tabellenbereich für temporäre Tabellen unvollständige Reorganisationsanforderungen vorhanden sind, müssen Sie möglicherweise den Konfigurationsparameter *indexrec* auf ACCESS setzen, um Fehler beim Neustart zu verhindern.

Wiederherstellen von Tabellenbereichen: wiederherstellbare Datenbanken:

Der beschädigte Tabellenbereich wird in den Status *offline und nicht verfügbar* und *Aktualisierende Wiederherstellung anstehend* versetzt, da eine Wiederherstellung nach Systemabsturz erforderlich ist. Die Neustartoperation ist erfolg-

reich, falls kein weiterer Fehler vorliegt. Der beschädigte Tabellenbereich kann wieder verwendet werden, wenn folgendes geschehen ist:

- Die Fehler an den beschädigten Behältern wurden behoben, ohne die ursprünglichen Daten zu verlieren, und anschließend der Tabellenbereich aktualisierend wiederhergestellt (ROLLFORWARD). (Die ROLLFORWARD-Operation versucht zunächst, den Tabellenbereich wieder vom Offline-Status in den normalen Status zurückzubringen.)
- Nach der Reparatur der beschädigten Behälter (mit oder ohne Verlust der Originaldaten) wurden eine RESTORE-Operation für den Tabellenbereich und anschließend eine ROLLFORWARD-Operation durchgeführt.

Wiederherstellen von Tabellenbereichen: nicht wiederherstellbare Datenbanken: Da eine Wiederherstellung nach einem Systemabsturz notwendig ist und die Protokolle nicht unendlich lange gespeichert werden, kann die Neustartoperation nur erfolgreich sein, wenn der Benutzer bereit ist, die beschädigten Tabellenbereiche zu löschen. (Erfolgreiche Wiederherstellung bedeutet, daß die Protokollsätze, die zum Wiederherstellen eines konsistenten Zustands der beschädigten Tabellenbereiche erforderlich sind, verloren sind, und damit die einzige gültige Aktion für solche Tabellenbereiche das Löschen dieser Tabellenbereiche ist.)

Dies können Sie erreichen, indem Sie eine Operation RESTART DATABASE ohne Qualifikationsmerkmale durchführen. Wenn keine beschädigten Tabellenbereiche vorhanden sind, ist diese Operation erfolgreich. Wenn sie fehlschlägt (SQL0290N), können Sie die vollständige Liste der Tabellenbereiche, die momentan beschädigt sind, in der Datei db2diag.log überprüfen.

- Wenn Sie bereit sind, alle diese Tabellenbereiche zu löschen, sobald die Operation RESTART DATABASE beendet ist, können Sie eine weitere Operation RESTART DATABASE einleiten, wobei Sie alle beschädigten Tabellenbereiche unter der Option DROP PENDING TABLESPACES auflisten. Wenn ein beschädigter Tabellenbereich in der Liste DROP PENDING TABLESPACES enthalten ist, wird der Tabellenbereich in den Status *Löschen anstehend* versetzt, und die einzige Option nach der Wiederherstellung, besteht darin, den Tabellenbereich zu löschen. Die Neustartoperation wird ohne Wiederherstellung dieses Tabellenbereichs fortgesetzt. Wenn ein beschädigter Tabellenbereich *nicht* in der Liste DROP PENDING TABLESPACES aufgeführt ist, schlägt die Operation RESTART DATABASE mit Code SQL0290N fehl.
- Wenn Sie diese Tabellenbereiche nicht löschen (und so die darin enthaltenen Daten nicht verlieren) wollen, haben Sie folgende Möglichkeiten:
 - Sie können abwarten und die beschädigten Behälter reparieren (ohne die Originaldaten zu verlieren), und anschließend die Operation RESTART DATABASE erneut versuchen.
 - Sie können eine Operation RESTORE DATABASE durchführen.

Anmerkung: Wenn der Name eines Tabellenbereichs in der Liste DROP PENDING TABLESPACES aufgeführt wird, bedeutet dies nicht, daß der Tabellenbereich in den Status *Löschen anstehend* versetzt wird. Dies wäre nur der Fall, wenn der Tabellenbereich während des Neustarts beschädigt ist. Nach der erfolgreichen Ausführung der Neustartoperation müssen Sie jeden der Tabellenbereiche, die sich im Status *Löschen anstehend* befinden (mit Hilfe des Befehls LIST TABLESPACES können Sie die Tabellenbereiche in diesem Status feststellen), durch Ausführen von Anweisungen DROP TABLESPACE löschen. Dadurch kann der Speicherbereich wieder verfügbar gemacht oder die Tabellenbereiche erneut erstellt werden.

Überlegungen zur Wiederherstellungsleistung

Folgendes ist im Hinblick auf die Wiederherstellungsleistung zu beachten:

- Sie können die Leistung für Datenbanken, die häufig aktualisiert werden, verbessern, indem Sie die Protokolldateien auf einer separaten Einheit speichern. In einer OLTP-Umgebung (Online Transaction Processing - Online-Transaktionsverarbeitung) ist oftmals mehr E/A-Aufwand für das Schreiben von Daten in die Protokolle als für das Speichern einer Zeile von Daten erforderlich. Dadurch, daß die Protokolldateien auf einer separaten Einheit angelegt werden, wird die Bewegung des Plattenzugriffsarms minimiert, die zum Wechseln zwischen einer Protokolldatei und den Datenbankdateien erforderlich ist.

Berücksichtigen Sie dabei außerdem, welche anderen Dateien sich auf der Platte befinden. Wenn die Protokolldateien z. B. auf eine Platte gestellt werden, die auch für den Systemseitenwechsel in einem System verwendet wird, das nicht über genügend Realspeicher verfügt, werden dadurch Ihre Optimierungsversuche zunichte gemacht.

- Gehen Sie wie folgt vor, um die zur Durchführung einer Wiederherstellungsoperation benötigte Zeit verringern:
 - Passen Sie die Größe der Wiederherstellungspuffer an. Die Puffergröße muß ein Vielfaches der Puffergröße sein, die bei der Sicherungsoperation verwendet wurde.
 - Erhöhen Sie die Anzahl von Puffern.

Wenn Sie mehrere Puffer und E/A-Kanäle verwenden, sollten Sie zumindest doppelt so viele Puffer als Kanäle verwenden, um sicherzustellen, daß die Kanäle nicht auf Daten warten müssen. Die Größe der Puffer wirkt sich ebenfalls auf die Leistung der Wiederherstellungsoperation aus. Die ideale Größe der Wiederherstellungspuffer ist ein Vielfaches der Größe des durch EXTENTSIZE definierten Bereichs für die Tabellenbereiche.

Wenn Sie mehrere Tabellenbereiche mit verschiedenen EXTENTSIZE-Bereichen haben, geben Sie einen Wert an, der ein Vielfaches des größten Werts für EXTENTSIZE darstellt.

Die empfohlene *Mindestanzahl* von Puffern ist die Summe der Anzahl der externen Einheiten bzw. Behälter und der für die Option PARALLELISM angegebenen Zahl.

- Verwenden Sie mehrere Quelleneinheiten.
- Legen Sie die Option PARALLELISM für die Wiederherstellungsoperation auf mindestens eins (1) größer als die Anzahl Quelleneinheiten fest.
- Wenn eine Datenbank große Mengen von Daten der Typen LONG und LOB enthält, kann das Wiederherstellen der Datenbank sehr viel Zeit in Anspruch nehmen. Wenn für die Datenbank die aktualisierende Wiederherstellung aktiviert ist, bietet der Befehl RESTORE die Möglichkeit, ausgewählte Tabellenbereiche wiederherzustellen. Handelt es sich bei Ihren Langfeld- und LOB-Daten um wichtige Unternehmensdaten, sollten Sie das Wiederherstellen dieser Tabellenbereiche gegen den für die Sicherungsoperation dieser Tabellenbereiche erforderlichen Zeitaufwand abwägen. Wenn die Langfeld- und LOB-Daten in separaten Tabellenbereichen gespeichert werden, läßt sich der für das Wiederherstellen von Daten erforderliche Zeitaufwand dadurch verringern, daß die Tabellenbereiche mit den Langfeld- und LOB-Daten von der Wiederherstellung ausgeschlossen werden. Wenn die LOB-Daten von einer getrennten Quelle reproduziert werden können, sollten Sie beim Erstellen oder Ändern einer Tabelle zum Hinzufügen von LOB-Spalten die Option NOT LOGGED verwenden. Wenn Sie die Tabellenbereiche, die Langfeld- und LOB-Daten enthalten, nicht wiederherstellen wollen, aber die Tabellenbereiche wiederherstellen müssen, die die Tabelle enthalten, müssen Sie die aktualisierende Wiederherstellung bis zum Ende der Protokolle durchführen, so daß alle Tabellenbereiche, die die Tabellendaten enthalten, konsistent sind.

Anmerkung: Wenn Sie einen Tabellenbereich sichern, der Tabellendaten ohne die zugehörigen LONG- oder LOB-Felder enthält, können Sie keine aktualisierende Wiederherstellung bis zu einem bestimmten Zeitpunkt für diesen Tabellenbereich ausführen. Alle Tabellenbereiche für eine Tabelle müssen gleichzeitig bis zum selben Zeitpunkt aktualisierend wiederhergestellt werden.

- Folgendes gilt für Sicherungs- und Wiederherstellungsoperationen:
 - Es sollten mehrere E/A-Puffer und Einheiten verwendet werden.
 - Ordnen Sie mindestens zweimal so viele Puffer zu, wie Einheiten verwendet werden.
 - Überlasten Sie die Bandbreite der Steuereinheiten der E/A-Einheiten nicht.

- Verwenden Sie eher mehr Puffer kleinerer Größe als wenige große Puffer.
- Optimieren Sie die Anzahl und die Größe der Puffer entsprechend den Systemressourcen.

Überlegungen zur Wiederherstellung nach Schäden

Unter dem Begriff *Wiederherstellung nach Schäden* werden die Aktivitäten zusammengefaßt, die zur Wiederherstellung der Datenbank nach einem Brand, einem Erdbeben, nach Vandalismus oder anderen zerstörerischen Ereignissen erforderlich sind. Ein Plan zur Wiederherstellung nach Schäden kann folgendes vorsehen:

- Ein Zweitstandort, der im Notfall zur Verfügung steht
- Eine andere Maschine, auf der die Datenbank wiederhergestellt werden kann
- Aufbewahrung der Datenbanksicherungen und archivierten Protokolle an einem anderen Standort

Wenn Ihr Plan zur Wiederherstellung nach Schäden vorsieht, die gesamte Datenbank auf einer anderen Maschine wiederherzustellen, benötigen Sie zumindest eine vollständige Datenbanksicherung und alle archivierten Protokolldateien für die Datenbank. Es kann sinnvoll sein, eine Bereitschaftsdatenbank ebenfalls auf dem aktuellen Stand zu halten, indem die Protokolle, die archiviert werden, auf sie angewendet werden. Oder Sie könnten die Datenbanksicherung und Protokollarchive auf dem Bereitschaftssystem speichern und eine Wiederherstellung bzw. eine aktualisierende Wiederherstellung nur ausführen, wenn ein Schaden aufgetreten ist. (In diesem Fall ist eine möglichst aktuelle Sicherung von Vorteil.) Bei Eintritt eines unvorhergesehenen Schadens ist es gewöhnlich jedoch nicht möglich, alle Transaktionen bis zum Zeitpunkt des Unglücksfalls wiederherzustellen.

Der Nutzen einer Tabellenbereichssicherung zur Wiederherstellung nach einer Beschädigung hängt vom Ausmaß der Beschädigung ab. In der Regel ist für die Fehlerbehebung die Wiederherstellung der gesamten Datenbank erforderlich, das heißt, daß eine Gesamtsicherung der Datenbank am Bereitschaftsstandort zur Hand sein sollte. Selbst wenn Sie ein getrenntes Sicherungsabbild jedes Tabellenbereichs haben, können Sie sie nicht zur Wiederherstellung der Datenbank verwenden. Wenn die Beschädigung eine Platte betrifft, kann mit Hilfe einer Tabellenbereichssicherung jedes Tabellenbereichs auf dieser Platte die Wiederherstellung durchgeführt werden. Wenn Sie aufgrund eines Plattenfehlers (oder aus einem anderen Grund) keinen Zugriff auf einen Behälter mehr haben, können Sie den Behälter an einer anderen Position wiederherstellen. Weitere Informationen finden Sie in "Erneutes Definieren von Tabellenbereichsbehältern während der Wiederherstellung mit RESTORE" im Handbuch *Systemverwaltung: Implementierung*.

Sowohl Tabellenbereichssicherungen als auch vollständige Datenbank-sicherungen können in Ihrer Planung zur Schadensbehebung eingesetzt werden. Die zur Sicherung, Wiederherstellung und aktualisierenden Wiederherstellung von Daten verfügbaren DB2-Einrichtungen bieten die Grundlage für einen Plan zur Schadensbehebung. Sie sollten die eingerichteten Wiederherstellungsprozeduren getestet haben, um Ihr Unternehmen zu schützen.

Begrenzen der Auswirkungen von Datenträgerfehlern

Zur Verringerung der Wahrscheinlichkeit eines Datenträgerfehlers und zur Vereinfachung der Wiederherstellung der Daten nach einem solchen Fehler, sollten Sie folgende Maßnahmen in Betracht ziehen:

- Spiegeln oder duplizieren Sie die Platten, die die Daten und Protokolle für wichtige Datenbanken enthalten.
- Sehen Sie in einer Umgebung mit partitionierten Datenbanken für die Behandlung der Daten und der Protokolle auf dem Katalogknoten ein genaues Verfahren vor. Da dieser Knoten für die Verwaltung der Datenbank von entscheidender Bedeutung ist, sollten Sie folgendes beachten:
 - Stellen Sie sicher, daß er sich auf einer zuverlässigen Platte befindet.
 - Duplizieren Sie ihn.
 - Erstellen Sie häufig Sicherungen.
 - Speichern Sie keine Benutzerdaten auf ihm.

Schützen gegen Plattenfehler

Wenn Sie sich Gedanken über die Möglichkeit einer Beschädigung von Daten oder aktiven Protokollen aufgrund eines Plattenabsturzes machen, sollten Sie über den Einsatz von Systemen nachdenken, die eine gewisse Toleranz gegenüber Plattenfehlern gewährleisten. Im allgemeinen bietet sich hier die Verwendung einer *Platteneinheit* (Disk Array) an. Eine Platteneinheit besteht aus einem Verbund von Plattenlaufwerken, die einer Anwendung gegenüber wie ein einziges großes Plattenlaufwerk erscheinen.

Solche Platteneinheiten bieten *Striping*-Funktionen, d. h. Funktionen zur Verteilung einer Datei über mehrere Platten, Funktionen zur Spiegelung von Platten und zur Paritätsprüfung.

Platteneinheiten werden manchmal einfach als RAID (Redundant Array of Independent Disks) bezeichnet. Der spezielle Begriff RAID bezieht sich im allgemeinen nur auf Hardwareplatteneinheiten. Platteneinheiten können darüber hinaus durch Software auf Betriebssystem- oder Anwendungsebene implementiert werden. Das Unterscheidungsmerkmal zwischen Hardware- und Softwareplatteneinheiten ist die Art der CPU-Verarbeitung von E/A-Anforderungen. Bei Hardwareplatteneinheiten werden die E/A-Aktivitäten von Einheiten-Controllern verwaltet, während bei Softwareplatteneinheiten dies vom Betriebssystem bzw. von einer Anwendung übernommen wird.

Hardwareplatteneinheiten (RAID): Bei einer RAID-Platteneinheit werden mehrere Datenträger von einem Einheiten-Controller mit eigener CPU verwendet und verwaltet. Da sämtliche Logik, die zur Verwaltung der zu dieser Einheit gehörenden Platten erforderlich ist, im Einheiten-Controller enthalten ist, ist diese Implementierung vom Betriebssystem unabhängig.

Es gibt fünf Typen der RAID-Architektur, die mit RAID-1 bis RAID-5 bezeichnet werden und alle eine Toleranz gegenüber Plattenfehlern bieten. Jede variiert in Funktion und Leistung. Im allgemeinen bezeichnet RAID eine redundante Gruppe von Einheiten (Redundant Array). Die Spezifikation RAID-0, die nur Funktionen zum Daten-Striping (und keine fehlertolerante Redundanz) vorsieht, wird aus dieser Diskussion ausgeklammert. Obgleich die RAID-Spezifikation fünf Architekturen definiert, werden heutzutage gewöhnlich nur RAID-1 und RAID-5 verwendet.

RAID-1 ist auch als zeitgleiches Spiegeln (Mirroring) oder Duplizierung (Duplexing) von Platten bekannt. Beim zeitgleichen Spiegeln werden Daten (eine vollständige Datei) von einer Platte auf einer zweiten Platte mit Hilfe eines einzigen Einheiten-Controllers dupliziert. Bei der Plattenduplizierung (Disk Duplexing) geschieht dasselbe wie beim Spiegeln, jedoch sind die Platten an einen zweiten Einheiten-Controller (wie zwei SCSI-Adapter) angeschlossen. Der hierdurch implementierte Datenschutz ist gut. Es kann jede der beiden Platten ausfallen, und die Daten bleiben über die jeweils andere Platte verfügbar. Bei der Plattenduplizierung ist auch bei Ausfall eines Einheiten-Controllers der vollständige Schutz der Daten gewährleistet. Die Leistung bei einer RAID-1-Implementierung ist ebenfalls gut, allerdings muß mit ihr in Kauf genommen werden, daß die erforderliche Plattenkapazität doppelt so groß sein muß wie die tatsächliche Datenmenge, da die Daten auf Laufwerks-paaren dupliziert werden.

Durch RAID-5 wird das plattenübergreifende Lesen und Schreiben (Striping) von Daten und Parität nach Sektoren über alle Platten hinweg implementiert. Die Paritätsinformationen werden zusammen mit Daten, und nicht auf einem dedizierten Laufwerk gespeichert. Der hierdurch implementierte Datenschutz ist gut. Wenn eine Platte ausfällt, stehen die Daten über die Informationen von den anderen Platten zusammen mit den verteilten Paritätsinformationen immer noch zur Verfügung. Die Leseleistung ist gut, während die Schreibleistung beträchtlich schlechter als bei RAID-1 oder einer normalen Platte ist. Für eine RAID-5-Konfiguration sind mindestens drei identische Platten erforderlich. Die Menge des zusätzlich erforderlichen Speicherplatzes für den Systemaufwand variiert mit der Anzahl der Platten in der Platteneinheit. In Fall einer RAID-5-Konfiguration von fünf Platten, beträgt der Speicher-mehraufwand 20%.

Bei Verwendung einer RAID-Platteneinheit hindert eine ausgefallene Platte (außer bei RAID-0) Sie nicht daran, auf die Daten der Platteneinheit zuzugrei-

fen. Wenn direkt im Betrieb anschließbare oder austauschbare Platten in der Platteneinheit verwendet werden, kann die ausgefallene Platte während des Betriebs der Platteneinheit gegen eine Ersatzplatte ausgetauscht werden. Wenn bei einer RAID-5-Konfiguration zwei Platten gleichzeitig ausfallen, gehen alle Daten verloren (jedoch ist die Wahrscheinlichkeit eines gleichzeitigen Ausfalls zweier Platten sehr gering).

Für Ihre Protokolle können Sie eine RAID-1-Konfiguration oder durch Software gespiegelte Platten (siehe „Softwareplatteneinheiten“) in Betracht ziehen, da diese Möglichkeiten eine Wiederherstellbarkeit bis zu dem Punkt des Ausfalls bieten und eine gute Schreibleistung zeigen, die für Protokolle wichtig ist. In Fällen, in denen Zuverlässigkeit von essentieller Bedeutung ist (d. h., daß keine Zeit für eine Wiederherstellung der Daten nach einem Plattenfehler verloren gehen darf) und die Schreibleistung nicht ebenso wichtig ist, kann eine RAID-5-Konfiguration für die Platten in Betracht kommen. Wenn hingegen die Schreibleistung eine erhebliche Rolle spielt und Sie diese auch mit dem Aufwand zusätzlichen Speicherplatzes sicherstellen wollen, ziehen Sie eine RAID-1-Konfiguration für Ihre Daten und Ihre Protokolle in Betracht.

Softwareplatteneinheiten: Eine Softwareplatteneinheit leistet im wesentlichen dasselbe wie eine Hardwareplatteneinheit (siehe „Hardwareplatteneinheiten (RAID)“ auf Seite 55), jedoch wird die Verwaltung des Plattenverkehrs entweder durch eine Betriebssystemtask oder durch ein auf dem Server aktives Anwendungsprogramm erledigt. Wie alle anderen Programme auch steht die Softwareplatteneinheit bei der Nutzung der CPU- und Systemressourcen in einer Konkurrenzsituation. Dies ist keine gute Lösung für ein System mit knappen CPU-Ressourcen, und es ist zu bedenken, daß die Gesamtleistung der Platteneinheit von der Auslastung und Kapazität der CPU des Servers abhängig ist.

Eine typische Softwareplatteneinheit bietet Funktionen zur Spiegelung von Platten (siehe „Hardwareplatteneinheiten (RAID)“ auf Seite 55). Obwohl redundante Platten erforderlich sind, ist eine Softwareplatteneinheit relativ preiswert zu implementieren, da kostenintensive RAID-Einheiten-Controller nicht benötigt werden.

Anmerkung: Wenn sich das Boot-Laufwerk des Betriebssystems in der Platteneinheit befindet, kann Ihr System nicht starten, wenn dieses Laufwerk ausfällt. Wenn das Laufwerk ausfällt, bevor die Platteneinheit aktiv ist, kann die Platteneinheit keinen Zugriff auf das Laufwerk ermöglichen. Ein Boot-Laufwerk sollte von der Platteneinheit getrennt betrieben werden.

Begrenzen der Auswirkungen von Transaktionsfehlern

Zur Verringerung der Auswirkungen von Transaktionsfehlern versuchen Sie, folgendes sicherzustellen:

- Eine ununterbrochene Stromversorgung
- Angemessenen Plattenspeicherplatz für Datenbankprotokolle
- Zuverlässige Kommunikationsverbindungen unter den Datenbankpartitions-Servern in einer Umgebung mit partitionierten Datenbanken
- Synchronisation der Systemuhren in einer Umgebung mit partitionierten Datenbanken (siehe „Synchronisation der Systemuhren in einem System mit partitionierten Datenbanken“).

Synchronisation der Systemuhren in einem System mit partitionierten Datenbanken

Unter den Datenbankpartitions-Servern sollte für eine relative hohe Synchronisation der Systemuhren gesorgt werden, um einen reibungslosen Datenbankbetrieb und eine uneingeschränkte Möglichkeit zur aktualisierenden Wiederherstellung zu gewährleisten. Zeitunterschiede unter den Datenbankpartitions-Servern, einschließlich aller potentiellen betriebs- und übertragungsbedingten Verzögerungen, für eine Transaktion sollten kleiner sein als der für den Konfigurationsparameter *max_time_diff* des Datenbankmanagers angegebene Wert, mit dem die maximale Zeitdifferenz zwischen Knoten definiert wird.

Zur Sicherstellung, daß die Zeitmarken der Protokollsätze die Reihenfolge von Transaktionen in einem partitionierten Datenbanksystem richtig wiedergeben, verwendet DB2 die Systemuhr der jeweiligen Maschine als Basis für die Zeitmarken in den Protokollsätzen. Wenn die Systemuhr jedoch vorgestellt wird, wird die Protokolluhr automatisch mit vorgestellt. Obwohl die Systemuhr wieder zurückgestellt werden kann, kann die Uhr für die Protokolle dies nicht und bleibt auf dieser *vorgestellten* Zeit, bis die Systemuhr mit dieser Zeit übereinstimmt. Dann sind die Systemuhren synchron. Dies hat zur Konsequenz, daß ein kurzfristiger Systemuhrfehler auf einem Datenbankknoten einen langfristigen Effekt auf die Zeitmarken von Datenbankprotokollen haben kann.

Nehmen Sie zum Beispiel an, daß die Systemuhr auf Datenbankpartitions-Server A fälschlicherweise auf den 7. November 1999 gesetzt wird, obwohl das Jahr 1997 ist, und nehmen Sie weiter an, daß der Fehler korrigiert wurde, *nachdem* eine Transaktion zur Aktualisierung in der Partition auf diesem Datenbankpartitions-Server festgeschrieben wurde. Wenn die Datenbank ständig in Gebrauch ist und regelmäßig aktualisiert wird, bleibt jeder Zeitpunkt zwischen dem 7. November 1997 und dem 7. November 1999 durch die aktualisierende Wiederherstellung praktisch unerreichbar. Wenn die Festschreibung auf Datenbankpartitions-Server A erfolgt, wird die Zeitmarke im

Datenbankprotokoll auf 1999 gesetzt und die Uhr des Protokolls bleibt auf dem 7. November 1999 stehen, bis die Systemuhr diesen Zeitpunkt erreicht. Wenn Sie versuchen, eine aktualisierende Wiederherstellung bis zu einem Zeitpunkt innerhalb dieses Zeitrahmens durchzuführen, stoppt die Operation an der ersten Zeitmarke, die über den angegebenen Stoppzeitpunkt hinausgeht (d. h. 7. November 1997).

Obwohl DB2 die Aktualisierung der Systemuhr nicht beschränken kann, verringert der Konfigurationsparameter *max_time_diff* des Datenbankmanagers die Möglichkeit, daß diese Art von Problem auftritt:

- Die konfigurierbaren Werte für diesen Parameter reichen von 1 Minute bis zu 24 Stunden. Weitere Informationen zur Einstellung des Konfigurationsparameters *max_time_diff* finden Sie im Handbuch *Systemverwaltung: Optimierung*.
- Wenn die erste Verbindungsanforderung an einen Nichtkatalogknoten erfolgt, sendet der Datenbankpartitions-Server seine Zeit an den Katalogknoten für die Datenbank. Der Katalogknoten überprüft, ob die Zeit auf dem Knoten, der die Verbindung anfordert, und seine eigene Zeit innerhalb des durch den Parameter *max_time_diff* definierten Bereichs liegen. Falls dieser Bereich überschritten wird, wird die Verbindung verweigert.
- Eine aktualisierende Transaktion, die auf mehr als zwei Datenbankpartitions-Server in der Datenbank zugreift, muß überprüfen, ob die Systemuhren auf den beteiligten Datenbankpartitions-Servern synchron sind, bevor die Aktualisierung festgeschrieben werden kann. Wenn zwei oder mehr Datenbankpartitions-Server eine Zeitdifferenz aufweisen, die das durch den Parameter *max_time_diff* gesetzte Limit überschreitet, wird die Transaktion rückgängig gemacht, um zu verhindern, daß die falsche Zeit auf weitere Datenbankpartitions-Server übertragen wird.

Gehen Sie wie folgt vor, um eine falsche Zeitmarke in einem Datenbankprotokoll zu korrigieren und eine weitere Verbreitung der falschen Zeitmarke zu verhindern:

1. Stellen Sie die Systemuhr auf die richtige Zeit ein.
2. Stellen Sie die Datenbankpartition auf dem richtigen Datenbankpartitions-Server mit einer Sicherung wieder her, die vor dem Einstellen der falschen Systemzeit erstellt wurde.
3. Stellen Sie die Änderungen bis zum Ende des Protokolls für die Datenbankpartition aktualisierend wieder her.
4. Erstellen Sie sofort nach der aktualisierenden Wiederherstellung der Änderungen eine Sicherungskopie der Datenbankpartition.

Nach diesen Maßnahmen ist die Protokollzeit richtig eingestellt, und die falsche Zeitmarke wird nicht weitergegeben. Sie sind in der Lage, eine

Wiederherstellung bis zu einem bestimmten Zeitpunkt mit Hilfe der letzten Sicherung durchzuführen, die von der Datenbankpartition erstellt wurde.

Reorganisieren von Tabellen in einer Datenbank

Eine Tabelle kann infolge zahlreicher Aktualisierungen eine wachsende Fragmentierung aufweisen, wodurch die Leistung verringert wird. Wenn Sie Statistikdaten gesammelt haben und keine sichtbare Leistungsverbesserung festgestellt haben, kann eine Reorganisation der Tabellendaten hilfreich sein. Bei der Reorganisation von Tabellendaten werden die Daten in eine physische Reihenfolge gemäß einem angegebenen Index gebracht und der freie Speicherplatz, der fragmentierten Daten eigen ist, entfernt. Dies kann zu einem schnelleren Zugriff auf die Daten und somit zu einer Verbesserung der Leistung führen.

Vor der Reorganisation von Tabellen wird empfohlen, den Befehl REORGCHK auszuführen und Statistikdaten über die Tabelle zu sammeln. Die Ausführung dieses Befehls hilft bei der Bestimmung, ob eine Reorganisation der Tabellendaten angebracht ist. Informationen zum Befehl REORGCHK finden Sie im Handbuch *Command Reference*.

Übersicht über die Sicherheit unter DB2

Zum Schutz von Daten und Ressourcen auf einem Datenbank-Server verwendet DB2 eine Kombination aus externen Sicherheitsservices und internen Zugriffssteuerinformationen. Wenn Sie auf den Datenbank-Server zugreifen wollen, müssen Sie einige Sicherheitsüberprüfungen durchlaufen, bevor Ihnen Zugriff auf Datenbankdaten oder Ressourcen gewährt wird. Der erste Schritt in der Datenbanksicherheit wird als *Authentifizierung* (Identitätsüberprüfung) bezeichnet, bei der ein Benutzer nachweisen muß, daß er der ist, der er behauptet zu sein. Der zweite Schritt wird als *Berechtigung* (Authorization) bezeichnet, bei der der Datenbankmanager entscheidet, ob ein überprüfter Benutzer die angeforderte Aktion ausführen oder auf die angeforderten Daten zugreifen darf.

Authentifizierung

Die Authentifizierung (Identifikationsüberprüfung) wird mit Hilfe einer Sicherheitseinrichtung außerhalb von DB2 vollzogen. Die Sicherheitseinrichtung kann Teil des Betriebssystems, ein separates Produkt oder, in bestimmten Fällen, auch nicht vorhanden sein. Auf UNIX-Systemen ist die Sicherheitseinrichtung in das Betriebssystem selbst integriert. DCE-Sicherheitsservices (DCE Security Services) ist ein separates Produkt, das die Sicherheitseinrichtung in einer verteilten Umgebung zur Verfügung stellt. Unter den Betriebssystemen Windows 95 oder Windows 3.1 gibt es keine Sicherheitseinrichtungen.

Die Sicherheitseinrichtung benötigt zwei Elemente zur Authentifizierung eines Benutzers: eine Benutzer-ID und ein Kennwort. Die Benutzer-ID identifiziert den Benutzer der Sicherheitseinrichtung gegenüber. Durch die Eingabe des richtigen Kennworts (eine Information, die nur dem Benutzer und der Sicherheitseinrichtung bekannt ist) wird die Identität des Benutzers (entsprechend der Benutzer-ID) bestätigt.

Nach erfolgreicher Authentifizierung:

- Der Benutzer muß für DB2 mit Hilfe eines SQL-Berechtigungsnamens oder einer Berechtigungs-ID (*authid*) identifiziert werden. Der Name kann mit der Benutzer-ID übereinstimmen oder ein zugeordneter Wert sein. Beispielsweise wird auf UNIX-basierten Systemen eine DB2-Berechtigungs-ID (*authid*) abgeleitet, indem eine UNIX-Benutzer-ID, die den DB2-Namenskonventionen entspricht, in Großbuchstaben umgesetzt wird. Im Produkt der DCE-Sicherheitsservices ist die DB2-Berechtigungs-ID (*authid*) in der DCE-Registrierdatenbank enthalten und wird daraus extrahiert, wenn die Authentifizierung erfolgreich beendet wurde.
- Eine Liste von Gruppen, zu denen der Benutzer gehört, wird abgerufen. Die Gruppenzugehörigkeit kann bei der Berechtigung des Benutzers verwendet werden. Gruppen sind Definitionseinheiten von Sicherheitseinrichtungen, denen auch DB2-Berechtigungsnamen zugeordnet sein müssen. Diese Zuordnung erfolgt mit einer ähnlichen Methode wie für Benutzer-IDs.

DB2 ruft eine Liste von Gruppen bis zu einer maximalen Anzahl von 64 Gruppen ab. Wenn ein Benutzer Mitglied in mehr als 64 Gruppen ist, werden nur die ersten 64 Gruppen, denen gültige DB2-Berechtigungsnamen zugeordnet sind, der DB2-Gruppenliste hinzugefügt. Es wird kein Fehler gemeldet. Alle weiteren Gruppen über die ersten 64 hinaus werden von DB2 ignoriert.

DB2 verwendet die Sicherheitseinrichtung zur Authentifizierung von Benutzern auf eine von zwei Arten:

- DB2 verwendet eine erfolgreiche Anmeldung durch das Sicherheitssystem als Nachweis der Identität und läßt folgendes zu:
 - Verwendung lokaler Befehle zum Zugriff auf lokale Daten
 - Verwendung von Fernverbindungen, bei denen der Server die Authentifizierung auf dem Client akzeptiert
- DB2 akzeptiert eine Kombination aus Benutzer-ID und Kennwort. DB2 verwendet eine erfolgreiche Gültigkeitsprüfung dieses Paares durch die Sicherheitseinrichtung als Identitätsnachweis und erlaubt folgendes:
 - Verwendung von Fernverbindungen, bei denen der Server einen Beweis der Authentifizierung verlangt

- Verwendung von Operationen, bei denen der Benutzer einen Befehl unter einer anderen als der zur Anmeldung verwendeten Identität ausführen will

DB2-Administratoren können andere Personen auf EEE-Systemen (Enterprise - Extended Edition) unter AIX und Windows NT durch die Variable DB2CHGPWD_EEE der Profilregistrierdatenbank dazu berechtigen, Kennwörter zu ändern. Der Standardwert für diese Variable nicht definiert (d. h. NOT SET - inaktiviert.) DB2CHGPWD_EEE akzeptiert die Booleschen Standardwerte, die von anderen DB2-Profilvariablen verwendet werden.

Der DB2-Administrator ist für die zentrale Verwaltung der Kennwörter aller Knoten verantwortlich. Dazu kann er unter Windows NT einen Windows NT-Domänen-Controller oder unter AIX den NIS verwenden.

Anmerkung: Wenn die Kennwörter nicht zentral verwaltet werden, kann die Aktivierung der Variable DB2CHGPWD_EEE möglicherweise dazu führen, daß die Kennwörter nicht über alle Knoten hinweg konsistent sind. Wenn Sie also die Funktion für die Kennwortänderung verwenden, wird Ihr Benutzerkennwort nur auf dem Knoten geändert, mit dem Sie verbunden sind.

Mit DB2 UDB unter AIX können fehlgeschlagene Kennworteingaben beim Betriebssystem protokolliert werden. Dadurch läßt sich ermitteln, wann ein Client die Anzahl zulässiger Anmeldeversuche überschritten hat, die im Parameter LOGINRETRIES angegeben ist.

Zusätzliche Informationen zu Gültigkeitsprüfungen bei Systemanmeldungen, die besonders wichtig sind, wenn Sie ferne Clients haben, die auf die Datenbank zugreifen, finden Sie in "Auswählen einer Authentifizierungsart für den Server" im Handbuch *Systemverwaltung: Implementierung*.

Berechtigung

Die Berechtigung (Authorization) ist der Prozeß, durch den DB2 Informationen über einen überprüften DB2-Benutzer abrufen, die angeben, welche Datenbankoperationen der Benutzer durchführen und auf welche Datenobjekte er zugreifen darf. Bei jeder Benutzeranforderung kann es mehr als eine Berechtigungsprüfung geben, je nachdem, welche Objekte und Operationen beteiligt sind.

Die Berechtigung wird mit Hilfe von DB2-Einrichtungen durchgeführt. DB2-Tabellen und DB2-Konfigurationsdateien werden zur Speicherung der Berechtigungen verwendet, die jedem Berechtigungsnamen zugeordnet sind. Der Berechtigungsname eines authentifizierten Benutzers und die Namen der Gruppen, zu denen der Benutzer gehört, werden mit den gespeicherten Berechtigungen verglichen. Anhand dieses Vergleichs entscheidet DB2, ob der angeforderte Zugriff gewährt wird.

Es gibt zwei Arten der Berechtigungen, die von DB2 gespeichert werden: Zugriffsrechte und Berechtigungsstufen. Ein *Zugriffsrecht* definiert eine einzelne Berechtigung für einen Berechtigungsnamen, Datenbankressourcen zu erstellen oder auf sie zuzugreifen. Zugriffsrechte werden in den Datenbankkatalogen gespeichert. *Berechtigungsstufen* bilden eine Methode, Zugriffsrechte und Kontrolle über Operationen zur Wartung des Datenbankmanagers und über Dienstprogrammoperationen auf einer höheren Stufe in Gruppen zusammenzufassen. Datenbankspezifische Berechtigungen werden in den Datenbankkatalogen gespeichert, während Systemberechtigungen durch Gruppenzugehörigkeit zugeordnet und in der Konfigurationsdatei des Datenbankmanagers für das jeweilige Exemplar gespeichert werden.

Gruppen stellen eine zweckmäßige Methode dar, für einen Benutzerverbund Berechtigungen durchzuführen, ohne jedem Benutzer einzeln Zugriffsrechte erteilen bzw. entziehen zu müssen. Sofern nicht anders angegeben, können Gruppenberechtigungsnamen überall dort verwendet werden, wo Berechtigungsnamen zu Berechtigungszwecken verwendet werden. Im allgemeinen wird die Gruppenzugehörigkeit für dynamisches SQL und Berechtigungen für Nichtdatenbankobjekte (z. B. Befehle auf Exemplarebene und Dienstprogramme) und nicht für statisches SQL berücksichtigt. Ein Ausnahmefall zu diesem allgemeinen Fall ist das Erteilen von Zugriffsrechten für PUBLIC, die bei der Verarbeitung von statischem SQL berücksichtigt werden. Spezielle Fälle, in denen Gruppenzugehörigkeit nicht gültig ist, werden in der DB2-Dokumentation entsprechend vermerkt.

Weitere Informationen finden Sie in "Zugriffsrechte und Berechtigungen" im Handbuch *Systemverwaltung: Implementierung*.

Übersicht über Authentifizierung und Berechtigung in einer zusammengesetzten Datenbank

Da ein DB2-System zusammengesetzter Datenbanken auf Daten in mehreren Datenbankverwaltungssystemen zugreifen kann, sind möglicherweise zusätzliche Maßnahmen zur Absicherung der Daten erforderlich.

Bei der Planung der Lösung zur Authentifizierung müssen Sie berücksichtigen, daß die Benutzer sich möglicherweise bei den Datenquellen und bei DB2 authentifizieren müssen. In einem System zusammengesetzter Datenbanken kann die Authentifizierung auf DB2-Client-Workstations, auf DB2-Servern, an den Datenquellen (DB2, DB2 für OS/390, anderen DRDA-Servern, Oracle) oder einer Kombination von DB2 (Client oder DB2-Server) und Datenquellen stattfinden. Selbst in DCE-Umgebungen, können spezielle Maßnahmen erforderlich sein, wenn für die Anmeldung an den Datenquellen eine Benutzer-ID und ein Kennwort abgefragt wird. Weitere Informationen finden Sie in "Authentifizierungsverarbeitung in zusammengesetzten Datenbanken" im Handbuch *Systemverwaltung: Implementierung*.

| Entsprechend müssen die Benutzer eine Berechtigungsprüfung an Daten-
| quellen und für DB2 durchlaufen. Die jeweilige Datenquelle (DB2, Oracle,
| DB2 für OS/390 usw.) verwaltet die Sicherheit der Objekte, die von ihr
| gesteuert werden. Wenn ein Benutzer eine Operation mit einem Kurznamen
| ausführt, muß er für die Tabelle oder Sicht, auf die sich der Kurzname
| bezieht, eine Berechtigungsprüfung durchlaufen.

Kapitel 3. Systeme zusammengeschlossener Datenbanken

Ein *System zusammengeschlossener Datenbanken* ist ein Datenbankverwaltungssystem (DBMS), das Anwendungen und Benutzer unterstützt, die mit SQL-Anweisungen arbeiten, in denen auf zwei oder mehr Datenbankverwaltungssysteme oder Datenbanken in einer einzelnen Anweisung verwiesen wird. Ein Beispiel hierfür ist eine Verknüpfung zwischen Tabellen in zwei unterschiedlichen DB2-Datenbanken. Diese Anweisungsart wird als eine *verteilte Anforderung* bezeichnet.

Ein System zusammengeschlossener Datenbanken unter DB2 Universal Database unterstützt verteilte Anforderungen über Datenbanken und Datenbankverwaltungssysteme hinweg. Sie können z. B. eine UNION-Operation zwischen einer DB2-Tabelle und einer Oracle-Sicht ausführen. Zu den unterstützten Datenbankverwaltungssystemen gehören DB2, Produkte der DB2-Produktfamilie (wie z. B. DB2 für OS/390 und DB2 für IBM AS/400) und Oracle.

In DB2-Systemen zusammengeschlossener Datenbanken besteht für Datenbankobjekte *Positionstransparenz*. Wenn Informationen (in Tabellen und Sichten) versetzt werden, können Verweise auf diese Informationen (sogenannte *Kurznamen*) ohne Änderung an den Anwendungen, die die Informationen anfordern, aktualisiert werden. DB2-Systeme zusammengeschlossener Datenbanken bieten eine *Kompensation* für Datenbankverwaltungssysteme, die nicht das gesamte SQL von DB2 oder bestimmte Optimierungsfunktionen unterstützen. Operationen, die auf einem solchen Datenbankverwaltungssystem nicht ausgeführt werden können (z. B. rekursives SQL), werden unter DB2 ausgeführt.

Ein DB2-System zusammengeschlossener Datenbanken arbeitet *halbautonom*: DB2-Abfragen, die Verweise auf Oracle-Objekte enthalten, können übergeben werden, während Oracle-Anwendungen auf denselben Server zugreifen. Ein DB2-System zusammengeschlossener Datenbanken hat weder den alleinigen Zugriff auf Oracle oder andere DBMS-Objekte noch beschränkt es ihn (abgesehen von den Einschränkungen wegen der Integrität und durch Sperren).

Ein DB2-System zusammengeschlossener Datenbanken besteht aus einem Exemplar unter DB2 UDB, einer Datenbank, die als *zusammengeschlossene Datenbank* dient, und einer oder mehreren *Datenquellen*. Die zusammengeschlossene Datenbank enthält Katalogeinträge, die die Datenquellen und ihre Kenndaten angeben. Eine Datenquelle besteht aus einem DBMS und Daten. Anwendungen stellen eine Verbindung zur zusammengeschlossenen Daten-

bank her, wie zu jeder anderen DB2-Datenbank. In Abb. 20 ist eine Umgebung zusammenschlossener Datenbanken dargestellt.

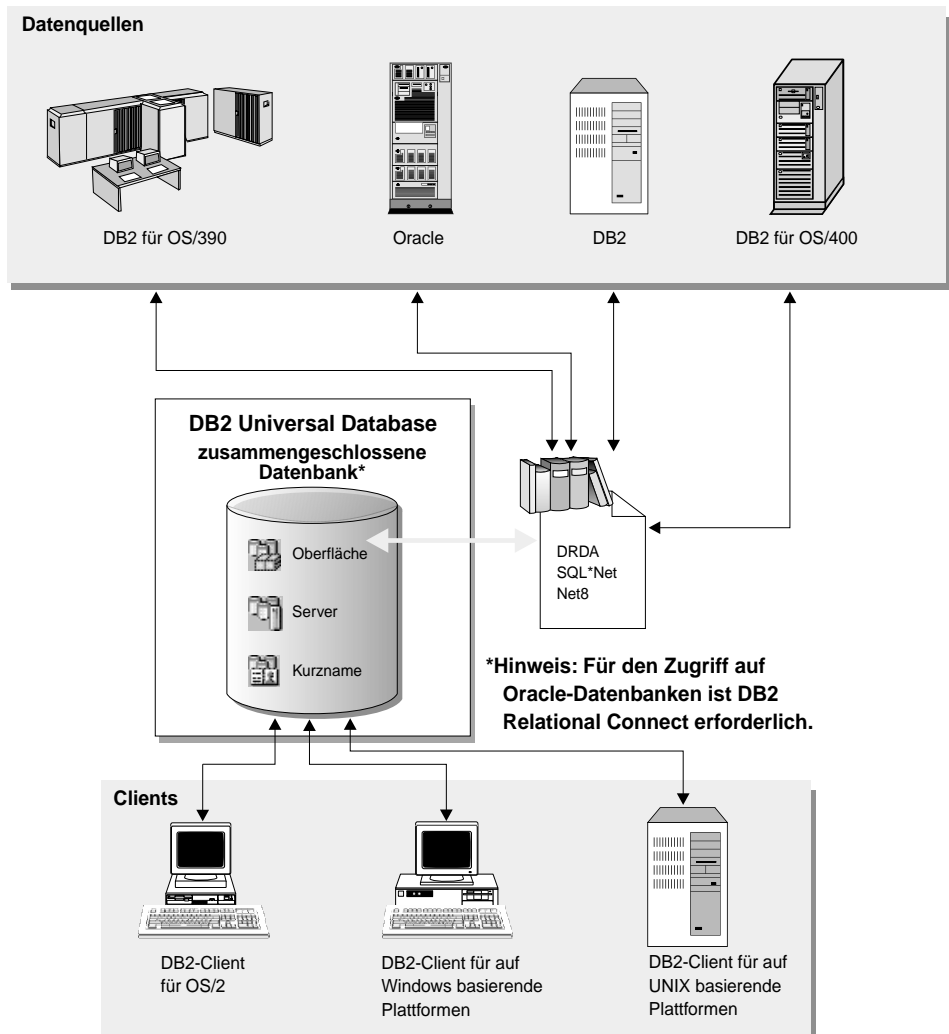


Abbildung 20. Ein System zusammenschlossener Datenbanken

Die Katalogeinträge der zusammenschlossenen DB2-Datenbank enthalten Informationen zu Datenquellenobjekten: den Namen dieser Objekte, die in diesen Objekten enthaltenen Informationen und die Bedingungen für die Verwendung der Objekte. Da dieser DB2-Katalog Informationen zu Objekten in vielen Datenbankverwaltungssystemen speichert, wird er auch *globaler Katalog* genannt. Objektattribute werden in dem Katalog gespeichert. Die tatsächlichen Datenbankverwaltungssysteme, auf die verwiesen wird, Module,

die für die Kommunikation mit der Datenquelle verwendet werden, und DBMS-Datenobjekte (wie z. B. Tabellen), auf die zugegriffen wird, befinden sich außerhalb der Datenbank. (Eine Ausnahme: Eine zusammengeschlossene Datenbank kann eine Datenquelle für das System zusammengeschlossener Datenbanken sein.) Sie können zusammengeschlossene Objekte mit Hilfe der Steuerzentrale oder mit Hilfe von DDL-Anweisungen in SQL erstellen. Folgende Objekte sind für zusammengeschlossene Datenbanken erforderlich:

Oberflächen

Geben die Module (DLL, library usw.) an, die für den Zugriff auf eine bestimmte Klasse oder Kategorie von Datenquelle verwendet werden.

Server Definieren Datenquellen. Zu den Server-Daten gehören der Name der Oberfläche, der Server-Name, die Server-Art, die Server-Version, Berechtigungsinformationen und die Server-Optionen.

Kurznamen

Kennungen, die in der zusammengeschlossenen Datenbank gespeichert sind und auf bestimmte Datenquellenobjekte (wie z. B. Tabellen, Aliasnamen, Sichten) verweisen. Anwendungen verweisen in Abfragen auf Kurznamen genauso, wie sie auf Tabellen und Sichten verweisen.

Sie können je nach Bedarf zusätzliche Objekte erstellen:

- Benutzerzuordnungen, um Authentifizierungsfunktionen zu implementieren.
- Datentypzuordnungen, um die Abhängigkeit zwischen einem Datenquellentyp und einem DB2-Typ individuell zu definieren.
- Funktionszuordnungen, um eine lokale Funktion einer Datenquellenfunktion zuzuordnen.
- Indexspezifikationen, um die Leistung zu erhöhen.

Wenn ein System zusammengeschlossener Datenbanken eingerichtet ist, kann auf die Informationen an Datenquellen zugegriffen werden, so als ob diese sich in einer einzigen großen Datenbank befänden. Benutzer und Anwendungen senden Abfragen an eine zusammengeschlossene Datenbank, die dann Daten von Systemen der DB2-Produktfamilie und Oracle-Systemen nach Bedarf abrufen. Benutzer und Anwendungen geben in Abfragen Kurznamen an. Diese Kurznamen stellen Verweise auf Tabellen und Sichten bereit, die sich in Datenquellen befinden. Aus der Perspektive eines Endbenutzers sind Kurznamen mit Aliasnamen vergleichbar.

Es gibt viele Faktoren, die sich auf die Leistung eines Systems zusammengeschlossener Datenbanken auswirken. Der wichtigste Faktor ist, sicherzustellen, daß korrekte und aktuelle Informationen zu Datenquellen und ihren Objekten im globalen Katalog der zusammengeschlossenen Datenbank gespeichert

werden. Diese Informationen werden vom DB2-Optimierungsprogramm verwendet und können entscheidend dafür sein, daß Auswertungsoperationen auf die ferne Quelle der zu bearbeitenden Daten ausgelagert und dort ausgeführt werden. Im Handbuch *Systemverwaltung: Optimierung* finden Sie zusätzliche Informationen zur Leistung von Systemen zusammengeschlossener Datenbanken.

Der Betrieb eines DB2-Systems zusammengeschlossener Datenbanken unterliegt einigen Einschränkungen. Verteilte Anforderungen sind auf Lesezugriffsoperationen beschränkt. Außerdem können Sie die Dienstprogrammoperationen (LOAD, REORG, REORGCHK, IMPORT, RUNSTATS usw.) nicht für Kurznamen ausführen.

Sie können allerdings eine Durchgriffsfunktion verwenden, um DDL- und DML-Anweisungen direkt an Datenbankmanager zu übergeben, die die SQL-Programmversion verwenden, die dieser Datenquelle zugeordnet ist.

Systeme zusammengeschlossener Datenbanken tolerieren parallele Umgebungen. Leistungsvorteile sind in dem Maße begrenzt, in dem eine Abfrage einer zusammengeschlossenen Datenbank semantisch in Verweise auf lokale Objekte (Tabellen, Sichten) und Verweise auf Kurznamen aufgesplittet werden kann. Anforderungen für Kurznamendaten werden sequentiell verarbeitet. Lokale Objekte können parallel verarbeitet werden. Zum Beispiel würde in einer Abfrage `SELECT * FROM A, B, C, D`, in der A und B lokale Tabellen, C und D aber Kurznamen sind, die auf Tabellen in Oracle-Datenquellen verweisen, ein möglicher Plan die Tabellen A und B mit einer parallelen Verknüpfung verknüpfen. Die Ergebnisse würden anschließend sequentiell mit den Kurznamen C und D verknüpft.

Einrichten eines Systems zusammengeschlossener Datenbanken

DB2 Enterprise Edition (EE) und DB2 Enterprise - Extended Edition (EEE) können zusammengeschlossene Datenbanken unterstützen. Gehen Sie wie folgt vor, um ein System zusammengeschlossener Datenbanken einzurichten:

1. Wählen Sie die Installationsoption *Distributed Join für DB2-Datenquellen* von DB2 EE oder EEE während der Installation aus.
2. Wenn Oracle-Datenbanken in das System zusammengeschlossener Datenbanken aufgenommen werden, installieren Sie DB2 Relational Connect. Weitere Informationen finden Sie im Handbuch *DB2 Installation und Konfiguration Ergänzung*.
3. Setzen Sie den Konfigurationsparameter *federated* des Datenbankmanagers auf "YES".
4. Erstellen Sie Oberflächen, Server und Kurznamen (weitere Informationen finden Sie in "Erstellen einer Datenbank" im Handbuch *Systemverwaltung: Implementierung*).

5. Erstellen Sie zusätzliche Objekte, oder legen Sie Optionen nach Bedarf fest (weitere Informationen finden Sie in "Implementieren des Datenbankentwurfs" im Handbuch *Systemverwaltung: Implementierung*).

Kapitel 4. Parallele Datenbanksysteme

DB2 erweitert den Datenbankmanager auf die parallele Mehrknoten-umgebung. Eine *Datenbankpartition* ist ein Teil einer Datenbank, der aus seinen eigenen Daten, Indizes, Konfigurationsdateien und Transaktionsprotokollen besteht. Eine Datenbankpartition wird manchmal auch als Knoten oder Datenbankknoten bezeichnet. (Knoten (Node) war der Begriff, der im Produkt DB2 Parallel Edition für AIX Version 1 verwendet wurde.)

Eine *Einzelpartitionsdatenbank* ist eine Datenbank, die nur über eine Datenbankpartition verfügt. Sämtliche Daten in der Datenbank werden in dieser Partition gespeichert. In diesem Fall bieten Knotengruppen (siehe „Knotengruppen“ auf Seite 11), obwohl sie vorhanden sind, keine zusätzlichen Leistungsvorteile.

Eine *partitionierte Datenbank* ist eine Datenbank mit zwei oder mehr Datenbankpartitionen. Die Tabellen können in einer oder mehreren Datenbankpartitionen gespeichert werden. Wenn eine Tabelle in einer Knotengruppe mit mehreren Partitionen gespeichert ist, heißt dies, daß einige ihrer Zeilen in einer Partition gespeichert werden, während sich andere Zeilen in anderen Partitionen befinden.

In der Regel existiert eine einzelne Datenbankpartition auf jedem physischen Knoten, und die Prozessoren auf jedem System werden vom Datenbankmanager in jeder Datenbankpartition zur Verwaltung des jeweiligen Teils der Gesamtdaten in der Datenbank verwendet.

Da die Daten auf Partitionen aufgeteilt sind, können Sie das Potential mehrerer Prozessoren auf mehreren physischen Knoten zur Erfüllung von Informationsanforderungen nutzen. Anforderungen zur Abfrage und Aktualisierung von Daten werden automatisch in Unteranforderungen zerlegt und in den betroffenen Datenbankpartitionen parallel ausgeführt. Die Tatsache, daß Datenbanken auf Datenbankpartitionen verteilt sind, ist für Benutzer, die SQL-Anweisungen ausführen, transparent.

Die Benutzerinteraktion erfolgt über nur eine Datenbankpartition, die als *Koordinator-knoten* für den jeweiligen Benutzer bezeichnet wird. Der Koordinator ist in derselben Datenbankpartition aktiv wie die Anwendung bzw. im Fall einer fernen Anwendung in der Datenbankpartition, mit der die Anwendung verbunden ist. Eine beliebige Datenbankpartition kann als Koordinator-knoten verwendet werden.

Knotengruppen und Datenpartitionierung

Sie können benannte Untergruppen einer oder mehrerer Datenbankpartitionen in einer Datenbank definieren. Jede Untergruppe, die Sie definieren, wird als *Knotengruppe* bezeichnet. Jede Untergruppe, die mehr als eine Datenbankpartition enthält, wird als *Mehrpartitions-knotengruppe* bezeichnet. Mehrpartitions-knotengruppen können nur mit Datenbankpartitionen definiert werden, die zum selben Exemplar gehören.

Abb. 21 auf Seite 73 zeigt ein Beispiel einer Datenbank mit fünf Partitionen mit folgenden Knotengruppen:

- Knotengruppe 1 umfaßt alle Datenbankpartitionen außer einer.
- Knotengruppe 2 enthält eine Datenbankpartition.
- Eine Knotengruppe enthält zwei Datenbankpartitionen.
- Die Datenbankpartition innerhalb Knotengruppe 2 wird gemeinsam mit Knotengruppe 1 (und überlappend) benutzt.
- Es gibt eine einzelne Datenbankpartition innerhalb Knotengruppe 3, die gemeinsam mit Knotengruppe 1 (und überlappend) benutzt wird.

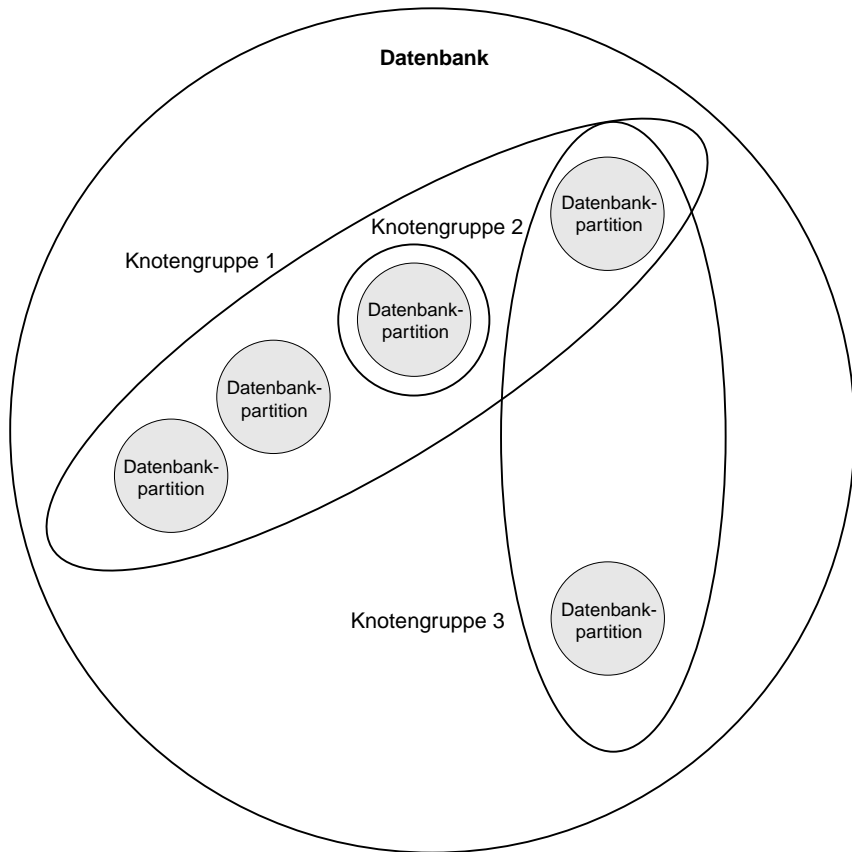


Abbildung 21. Knotengruppen in einer Datenbank

Eine neue Knotengruppe wird mit Hilfe der Anweisung `CREATE NODEGROUP` erstellt. Weitere Informationen finden Sie im Handbuch *SQL Reference*. Die Daten werden über alle Partitionen in einer Knotengruppe verteilt. Wenn Sie Knotengruppen mit mehreren Partitionen verwenden, müssen Sie eine Reihe von Gesichtspunkten im Hinblick auf den Aufbau der Knotengruppen beachten. Weitere Informationen finden Sie in „Entwerfen von Knotengruppen“ auf Seite 144.

Arten der Parallelität

Komponenten einer Task, wie zum Beispiel einer Datenbankabfrage, können parallel ausgeführt werden, um die Leistung erheblich zu erhöhen. Die Art der Task, die Datenbankkonfiguration und die Hardwareumgebung bestimmen, wie DB2 eine Task parallel ausführt. Diese Aspekte stehen in Wechselbeziehung zueinander und sollten bei der Arbeit am physischen und logischen

Entwurf einer Datenbank im Zusammenhang betrachtet werden. In diesem Abschnitt werden die folgenden Arten von Parallelität beschrieben, die von DB2 unterstützt werden:

- Ein-/Ausgabeparallelität
- Abfrageparallelität
- Dienstprogrammparallelität

Ein-/Ausgabeparallelität

Wenn mehrere Behälter für einen Tabellenbereich vorhanden sind, kann der Datenbankmanager die *parallele Ein-/Ausgabe* nutzen. Parallele E/A bezeichnet den Vorgang, bei dem Lese- bzw. Schreiboperationen mit zwei oder mehr Ein-/Ausgabeeinheiten gleichzeitig durchgeführt werden. Auf diese Weise können Verbesserungen des Durchsatzes erzielt werden.

Die E/A-Parallelität ist eine Komponente der jeweiligen Hardwareumgebungen, die im Abschnitt „Hardwareumgebungen“ auf Seite 78 beschrieben werden. In Tabelle 3 auf Seite 86 sind die Hardwareumgebungen aufgeführt, die am besten für parallele Ein-/Ausgabe geeignet sind.

Abfrageparallelität

Es gibt zwei Arten der Abfrageparallelität: abfrageübergreifende und abfrageinterne Parallelität.

Abfrageübergreifende Parallelität bezieht sich auf die Möglichkeit, daß mehrere Anwendungen gleichzeitig eine Datenbank abfragen können. Jede Abfrage wird unabhängig von allen anderen Abfragen ausgeführt, jedoch führt DB2 sie alle gleichzeitig aus. DB2 unterstützt diese Art der Parallelität seit jeher.

Abfrageinterne Parallelität bezeichnet die gleichzeitige Verarbeitung von Teilen einer einzelnen Abfrage mit Hilfe der *partitionsinternen Parallelität* bzw. der *partitionsübergreifenden Parallelität* (oder beiden).

Der Begriff *Abfrageparallelität* wird im gesamten Handbuch verwendet.

Partitionsinterne Parallelität

Der Begriff *partitionsinterne Parallelität* bezeichnet die Fähigkeit, eine Abfrage in mehrere Teile zu untergliedern. (Einige Dienstprogramme arbeiten ebenfalls mit dieser Art der Parallelität. Siehe „Dienstprogrammparallelität“ auf Seite 77.)

Bei der *partitionsinternen Parallelität* wird das, was im allgemeinen als eine einzige Datenbankoperation betrachtet wird (z. B. eine Indexerstellung, das Laden von Daten, SQL-Abfragen) in mehrere Teile unterteilt, von denen viele oder alle parallel *innerhalb einer einzigen Datenbankpartition* ausgeführt werden können.

Abb. 22 zeigt eine Abfrage, die in vier Teile aufgeteilt ist, die parallel ausgeführt werden können, wobei die Ergebnisse schneller geliefert werden als bei serieller Ausführung der Abfrage. Die Teile sind jeweils Kopien voneinander. Zur Nutzung der partitionsinternen Parallelität muß die Datenbank entsprechend konfiguriert werden. Sie können den Grad der Parallelität selbst auswählen oder ihn vom System festlegen lassen. Als Grad der Parallelität stellt die Anzahl der Teile einer Abfrage dar, die parallel ausgeführt werden.

In Tabelle 3 auf Seite 86 sind die Hardwareumgebungen aufgeführt, die am besten für partitionsinterne Parallelität geeignet sind.

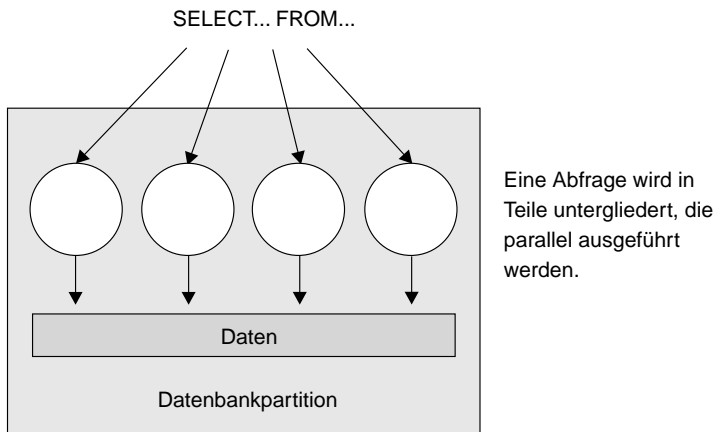


Abbildung 22. Partitionsinterne Parallelität

Partitionsübergreifende Parallelität

Der Begriff *partitionsübergreifende Parallelität* bezeichnet die Fähigkeit, eine Abfrage in mehreren Teilen über mehrere Partitionen einer partitionierten Datenbank auf einer oder mehreren Maschinen zu verteilen. Die Abfrage wird parallel ausgeführt. (Einige Dienstprogramme arbeiten ebenfalls mit dieser Art der Parallelität. Siehe „Dienstprogrammparallelität“ auf Seite 77.)

Bei der partitionsübergreifenden Parallelität wird das, was im allgemeinen als eine einzige Datenbankoperation betrachtet wird, (z. B. eine Indexerstellung, das Laden von Daten, SQL-Abfragen) in mehrere Teile unterteilt, von denen viele oder alle parallel über mehrere Partitionen einer partitionierten Datenbank hinweg auf einer oder mehreren Maschinen ausgeführt werden können.

Abb. 23 auf Seite 76 zeigt eine Abfrage, die in vier Teile aufgeteilt ist, die parallel ausgeführt werden können, wobei die Ergebnisse schneller zurückgeliefert werden als bei serieller Ausführung in einer Einzelpartition.

Der Grad der Parallelität wird im wesentlichen durch die Anzahl der Partitionen, die Sie erstellen, und die Art und Weise der Definition der Knotengruppen bestimmt.

In Tabelle 3 auf Seite 86 sind die Hardwareumgebungen aufgeführt, die am besten für partitionsübergreifende Parallelität geeignet sind.

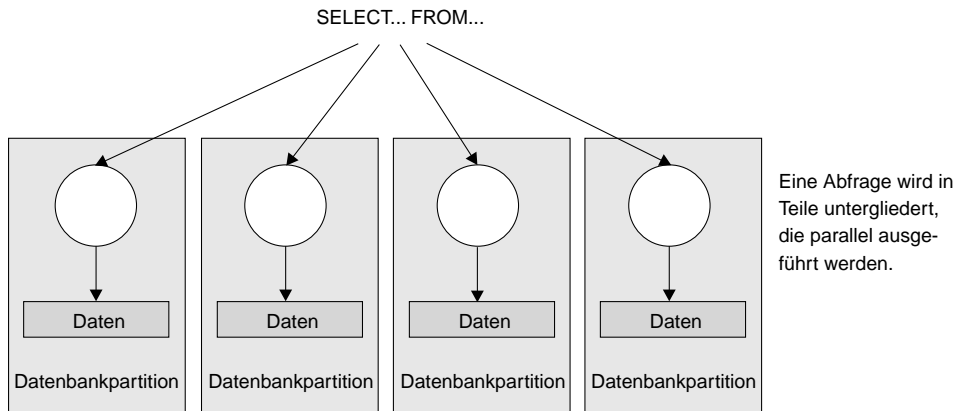


Abbildung 23. Partitionsübergreifende Parallelität

Gleichzeitiges Verwenden der partitionsinternen und partitionsübergreifenden Parallelität

Sie können die partitionsinterne Parallelität und die partitionsübergreifende Parallelität gleichzeitig nutzen. Diese Kombination bietet zwei Dimensionen der Parallelität, wodurch sich ein weiterer wesentlicher Anstieg der Verarbeitungsgeschwindigkeit für Abfragen ergibt:

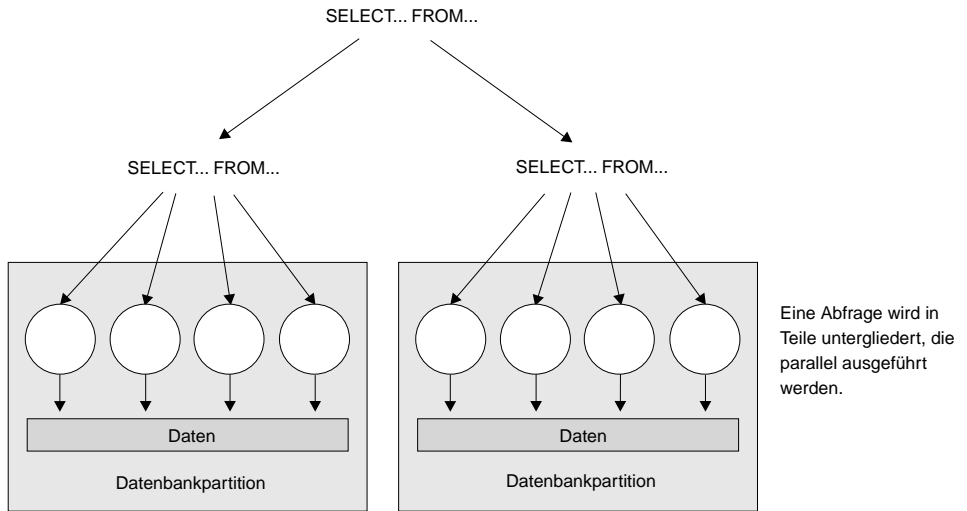


Abbildung 24. Gleichzeitige partitionsübergreifende und partitionsinterne Parallelität

Dienstprogrammparallelität

DB2-Dienstprogramme können mit partitionsinterner Parallelität arbeiten. Sie können außerdem die Vorteile der partitionsübergreifenden Parallelität nutzen. Wenn mehrere Datenbankpartitionen vorhanden sind, werden die Dienstprogramme in jeder der Partitionen parallel ausgeführt.

Das Dienstprogramm LOAD kann die Vorteile der partitionsinternen Parallelität und der E/A-Parallelität nutzen. Das Laden von Daten in eine Tabelle ist eine CDPUs-intensive Operation. Das Dienstprogramm LOAD nutzt die Möglichkeit mehrerer Prozessoren bei Operationen wie dem Analysieren und Formatieren von Daten. Darüber hinaus kann es parallele E/A-Server zum gleichzeitigen Schreiben der Daten in Behälter verwenden. Im Handbuch *Versetzen von Daten Dienstprogramme und Referenz* finden Sie Informationen darüber, wie Sie die Parallelität für das Dienstprogramm LOAD aktivieren können.

In einer Umgebung mit partitionierten Datenbanken nutzt das Dienstprogramm AutoLoader die Vorteile der partitionsinternen, partitionsübergreifenden und der E/A-Parallelität durch paralleles Aufrufen des Befehls LOAD in jeder Datenbankpartition, in der sich die Tabelle befindet. Weitere Informationen zum Dienstprogramm AutoLoader finden Sie im Handbuch *Versetzen von Daten Dienstprogramme und Referenz*.

Während der Indexerstellung erfolgt das Durchsuchen und das nachfolgende Sortieren der Daten parallel. DB2 nutzt sowohl die E/A-Parallelität als auch

die partitionsinterne Parallelität bei der Erstellung eines Index. Dadurch wird die Indexerstellung bei der Verarbeitung der Anweisung CREATE INDEX, beim Neustart (wenn ein Index als ungültig markiert ist) und bei der Reorganisation von Daten beschleunigt.

Sichern und Wiederherstellen von Daten sind Operationen, die wesentlich von der E/A-Leistung abhängig sind. DB2 nutzt sowohl die E/A-Parallelität als auch partitionsinterne Parallelität bei der Durchführung von BACKUP- und RESTORE-Operationen. Der Befehl BACKUP nutzt die E/A-Parallelität, indem er von mehreren Tabellenbereichsbehältern parallel liest und asynchron auf mehrere Sicherungsmedien parallel schreibt. In den Abschnitten zu den Befehlen BACKUP DATABASE und RESTORE DATABASE im Handbuch *Command Reference* finden Sie Informationen, wie Sie die Parallelität für diese beiden Dienstprogramme aktivieren können.

Hardwareumgebungen

Dieser Abschnitt enthält eine Übersicht über folgende Hardwareumgebungen:

- Einzelpartition auf Einzelprozessormaschine
- Einzelpartition auf Mehrprozessormaschine (SMP)
- Konfigurationen mit mehreren Partitionen
 - Partitionen mit einem Prozessor (MPP)
 - Partitionen mit mehreren Prozessoren (Gruppe von SMP-Systemen)
 - Logische Datenbankpartitionen (in DB2 Parallel Edition für AIX Version 1 auch als MLN (Multiple Logical Nodes) bezeichnet)

Für jede Umgebung werden die Kapazität und Skalierbarkeit behandelt. *Kapazität* bezieht sich hier auf die Anzahl der Benutzer und Anwendungen, die auf die Datenbank zugreifen können. Dies wird größtenteils durch Hauptspeicher, Agenten, Sperren, E/A-Operationen und Speicherverwaltung bestimmt. *Skalierbarkeit* bezeichnet die Fähigkeit einer Datenbank, bei zunehmender Größe weiterhin die gleichen Betriebsmerkmale und Antwortzeiten zu zeigen.

Einzelpartition auf einer Einzelprozessormaschine

Diese Umgebung verfügt über Hauptspeicher und Platte, enthält aber nur eine CPU (siehe Abb. 25 auf Seite 79). Für diese Umgebung werden viele verschiedene Bezeichnungen verwendet, wie zum Beispiel eigenständige Datenbank, Client-/Server-Datenbank, serielle Datenbank, Einprozessorsystem oder nicht parallele bzw. Einzelknotenumgebung.

Die Datenbank in dieser Umgebung erfüllt die Anforderungen einer Abteilung oder eines kleinen Büros, wobei die Daten und Systemressourcen (einschließlich des Einzelprozessors bzw. einer CPU) von einem einzelnen Datenbankmanager verwaltet werden.

In Tabelle 3 auf Seite 86 sind die Arten der Parallelität aufgeführt, die am besten in dieser Hardwareumgebung genutzt werden können.

Einzelprozessormaschine

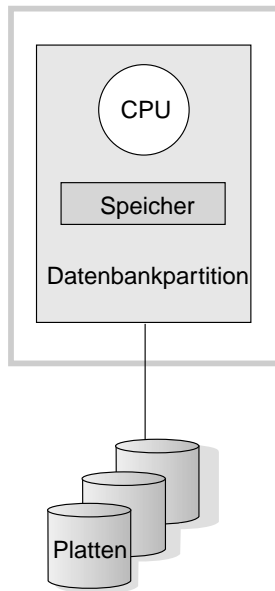


Abbildung 25. Einzelpartition auf einer Einzelprozessormaschine

Kapazität und Skalierbarkeit

In dieser Umgebung können Sie weitere Platten hinzufügen. Durch den Einsatz eines oder mehrerer E/A-Server für jede Platte kann mehr als eine E/A-Operation gleichzeitig stattfinden. Sie können außerdem weiteren Festplattenspeicher in dieser Umgebung hinzufügen.

Ein Einzelprozessorsystem ist durch die Menge an Plattenspeicherplatz, den der Prozessor handhaben kann, eingeschränkt. Jedoch kann, ungeachtet aller zusätzlichen Komponenten wie Arbeitsspeicher oder Festplattenspeicher, die Sie zur schnelleren Verarbeitung von Benutzeranforderungen vielleicht hinzufügen, eine einzelne CPU bei weiterer Zunahme der Auslastung Benutzeranforderungen nicht mehr schneller verarbeiten. Wenn Sie das Maximum an Kapazität und Skalierbarkeit erreicht haben, können Sie die Umrüstung auf ein Einzelpartitionssystem mit mehreren Prozessoren in Erwägung ziehen.

Einzelpartition auf Mehrprozessormaschine

Diese Umgebung besteht gewöhnlich aus mehreren gleich starken Prozessoren innerhalb derselben Maschine (siehe Abb. 26 auf Seite 80) und wird als *symmetrisches Mehrprozessorsystem (SMP-System)* bezeichnet (SMP - Symmetric Multi-Processor). Ressourcen wie Plattenspeicherplatz und Hauptspeicher werden *gemeinsam* benutzt.

Wenn mehrere Prozessoren verfügbar sind, können verschiedene Datenbankoperationen schneller durchgeführt werden. DB2 kann auch die Arbeit einer einzigen Abfrage auf die verfügbaren Prozessoren verteilen, um die Verarbeitungsgeschwindigkeit zu erhöhen. Andere Datenbankoperationen wie das Laden von Daten, das Sichern und Wiederherstellen von Tabellenbereichen sowie die Erstellung von Indizes auf vorhandenen Daten können die Verfügbarkeit mehrerer Prozessoren ausnutzen.

In Tabelle 3 auf Seite 86 sind die Arten der Parallelität aufgeführt, die am besten in dieser Hardwareumgebung genutzt werden können.

SMP-Maschine

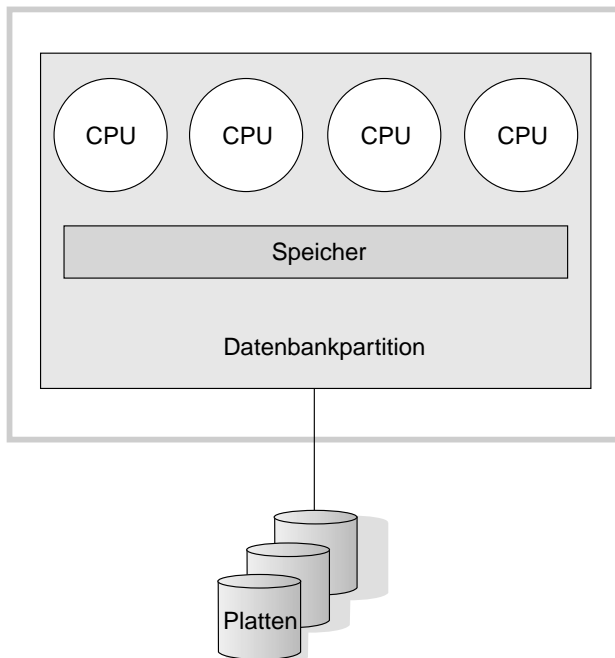


Abbildung 26. Einzelpartitionsdatenbank auf einem symmetrischen Mehrprozessorsystem (SMP)

Kapazität und Skalierbarkeit

In dieser Umgebung können Sie weitere Prozessoren hinzufügen. Da aber verschiedene Prozessoren versuchen könnten, gleichzeitig auf dieselben Daten zuzugreifen, können mit der Zunahme der Geschäftsoperationen Einschränkungen auftreten. Durch die gemeinsame Benutzung von Hauptspeicher und Platten werden effektiv sämtliche Datenbankdaten gemeinsam benutzt.

Sie können die E/A-Kapazität der Datenbankpartition, die Ihrem Prozessor zugeordnet ist, durch eine größere Anzahl von Platten erhöhen. Sie können E/A-Server speziell zur Verarbeitung von E/A-Anforderungen einrichten.

Durch den Einsatz eines oder mehrerer E/A-Server für jede Platte kann mehr als eine E/A-Operation gleichzeitig stattfinden.

Wenn Sie das Maximum an Kapazität und Skalierbarkeit erreicht haben, können Sie die Umrüstung auf ein System mit mehreren Partitionen in Erwägung ziehen.

Konfigurationen mit mehreren Partitionen

Sie können eine Datenbank in mehrere Partitionen aufteilen, die sich jeweils auf einer eigenen Maschine befinden. Mehrere Maschinen mit mehreren Datenbankpartitionen können in einer Gruppe zusammengefaßt werden. In diesem Abschnitt werden die folgenden Partitionskonfigurationen beschrieben:

- Partitionen auf Systemen mit einem Prozessor
- Partitionen auf Systemen mit mehreren Prozessoren
- Logische Datenbankpartitionen

Partitionen mit einem Prozessor

In dieser Umgebung gibt es viele Datenbankpartitionen. Jede Partition befindet sich auf einer eigenen Maschine mit eigenem Prozessor, eigenem Hauptspeicher und eigenen Platten (Abb. 27 auf Seite 82). Alle Maschinen sind über eine Kommunikationseinrichtung verbunden. Für eine solche Umgebung werden viele verschiedene Bezeichnungen gebraucht, wie zum Beispiel Cluster, Cluster von Einzelprozessoren, Umgebung mit exklusiver Parallelverarbeitung (MPP - Massively Parallel Processing) oder Konfiguration ohne gemeinsame Nutzung von Ressourcen. Die letztgenannte Bezeichnung charakterisiert die Anordnung der Ressourcen. Im Gegensatz zu einer SMP-Umgebung findet in einer MPP-Umgebung keine gemeinsame Nutzung von Hauptspeicher oder Platten statt. In der MPP-Umgebung fallen daher auch die Einschränkungen aufgrund der gemeinsamen Nutzung von Hauptspeicher und Platten fort.

Durch die Partitionierung in einer Umgebung kann eine Datenbank trotz ihrer physischen Verteilung auf mehrere Partitionen eine logische Gesamtheit bilden. Die Tatsache, daß die Daten auf Partitionen verteilt sind, bleibt für die Mehrheit der Benutzer transparent. Die Arbeit kann unter den Datenbankmanagern aufgeteilt werden. Die Datenbankmanager in den einzelnen Partitionen verrichten die Arbeit jeweils am eigenen Teil der Datenbank.

In Tabelle 3 auf Seite 86 sind die Arten der Parallelität aufgeführt, die am besten in dieser Hardwareumgebung genutzt werden können.

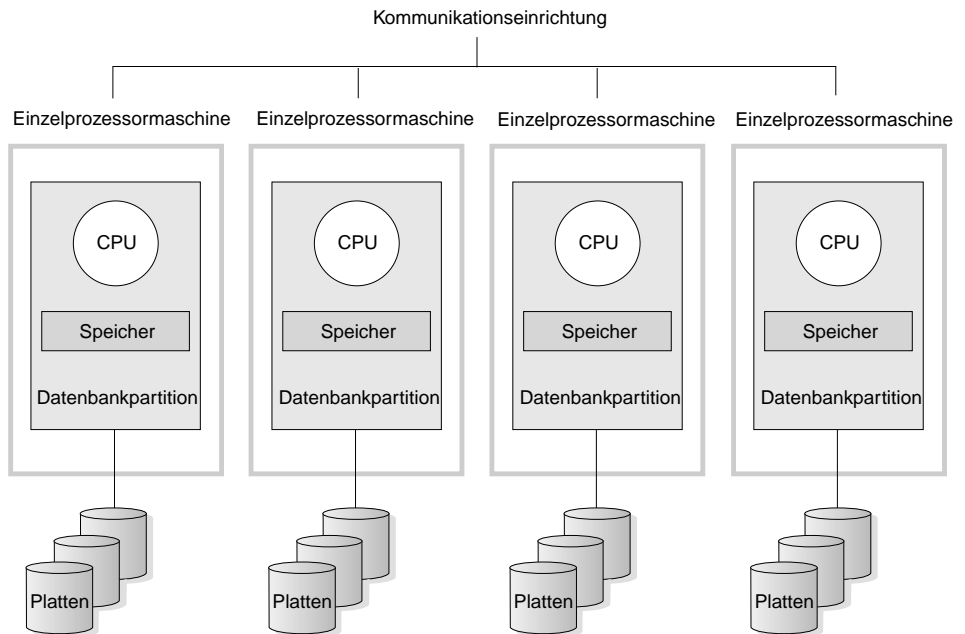


Abbildung 27. System zur exklusiven Parallelverarbeitung (MPP-System)

Kapazität und Skalierbarkeit: In dieser Umgebung können Sie weitere Datenbankpartitionen (Knoten) Ihrer Konfiguration hinzufügen. Auf einigen Plattformen, z. B. RS/6000 SP, ist die mögliche maximale Anzahl 512 Knoten. Jedoch kann es praktische Grenzen für die Verwaltung einer höheren Anzahl von Maschinen und Exemplaren geben.

Wenn Sie das Maximum an Kapazität und Systemgrößenflexibilität erreicht haben, können Sie die Umrüstung auf ein System in Erwägung ziehen, in dem jede Partition mehrere Prozessoren hat.

Partitionen mit mehreren Prozessoren

Eine Alternative zu einer Konfiguration, in der jede Partition einen einzigen Prozessor hat, ist eine Konfiguration, in der eine Partition über mehrere Prozessoren verfügt. Eine solche Konfiguration wird als *SMP-Cluster* bezeichnet (Abb. 28 auf Seite 83).

Diese Art der Konfiguration verbindet die Vorteile von SMP- und MPP-Parallelität. Dies bedeutet, daß eine Abfrage in einer Einzelpartition mit Hilfe mehrerer Prozessoren durchgeführt werden kann. Darüber hinaus kann eine Abfrage auch parallel in mehreren Partitionen durchgeführt werden.

In Tabelle 3 auf Seite 86 sind die Arten der Parallelität aufgeführt, die am besten in dieser Hardwareumgebung genutzt werden können.

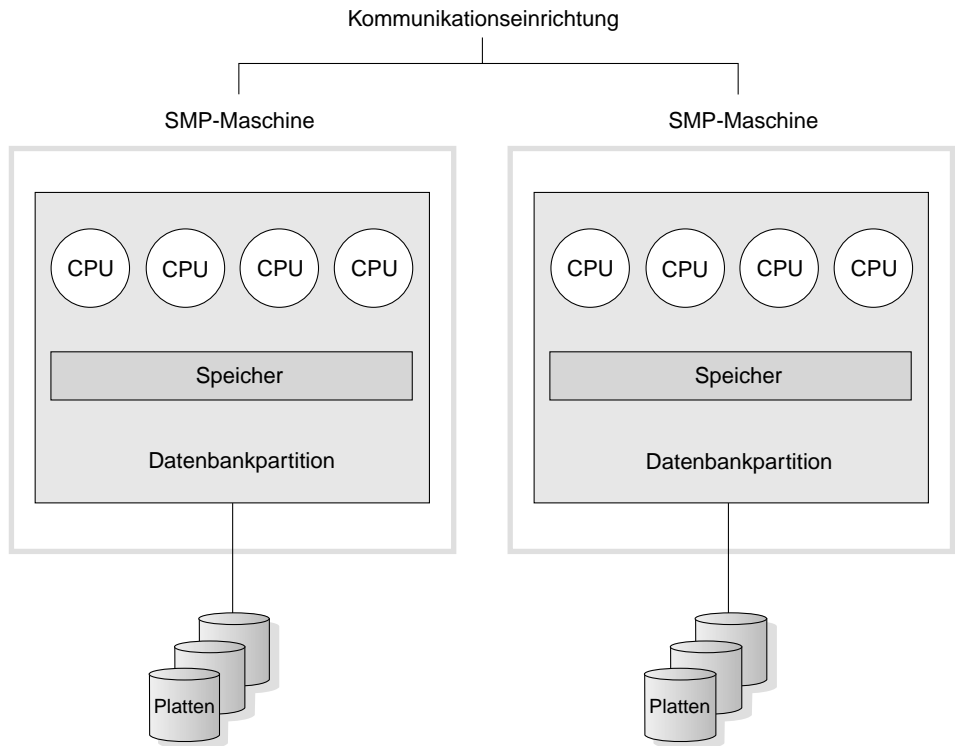


Abbildung 28. Gruppe von SMP-Maschinen

Kapazität und Skalierbarkeit: In dieser Umgebung können Sie weitere Datenbankpartitionen und den vorhandenen Datenbankpartitionen weitere Prozessoren hinzufügen.

Logische Datenbankpartitionen

Eine logische Datenbankpartition unterscheidet sich von einer physischen Partition darin, daß ihr nicht die Steuerung einer ganzen Maschine unterliegt. Obwohl die Maschine über gemeinsam benutzte Ressourcen verfügt, nutzen Datenbankpartitionen die Ressourcen nicht gemeinsam. Prozessoren werden gemeinsam benutzt, Platten und Hauptspeicher jedoch nicht.

Logische Datenbankpartitionen bieten Skalierbarkeit. Mehrere Datenbankmanager, die in mehreren logischen Partitionen ausgeführt werden, können die verfügbaren Ressourcen eventuell erschöpfender nutzen, als dies ein einzelner Datenbankmanager könnte. Abb. 29 auf Seite 84 veranschaulicht die Möglichkeit, mehr Skalierbarkeit auf einer SMP-Maschine zu gewinnen, indem weitere Partitionen hinzugefügt werden. Dies gilt insbesondere für Maschinen

mit vielen Prozessoren. Durch die Partitionierung der Datenbank können Sie jede Partition getrennt verwalten und wiederherstellen.

Große SMP-Maschine

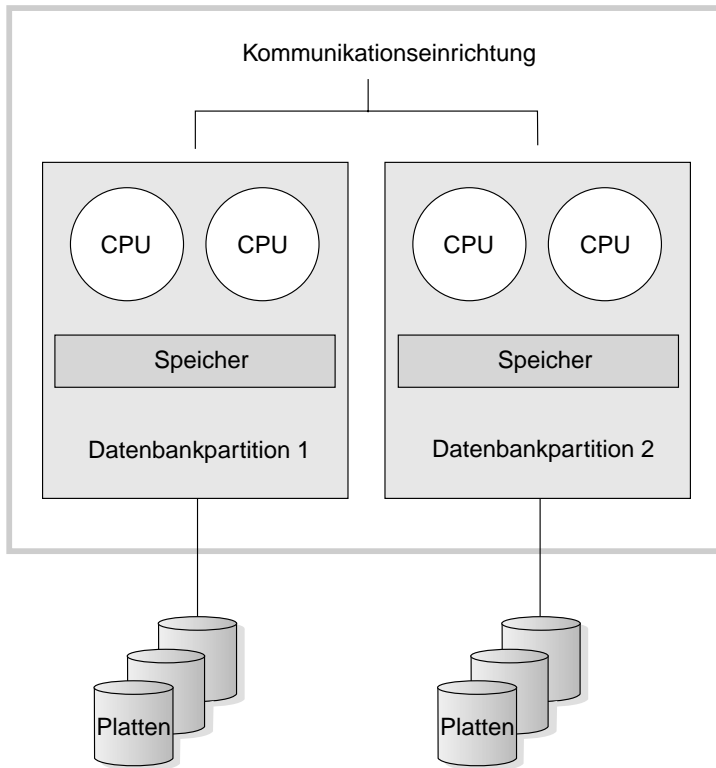


Abbildung 29. Partitionierte Datenbank auf einem symmetrischen Mehrprozessorsystem

Abb. 30 auf Seite 85 veranschaulicht die Möglichkeit, die in Abb. 29 gezeigte Konfiguration zu vervielfachen, um die Verarbeitungsleistung zu erhöhen.

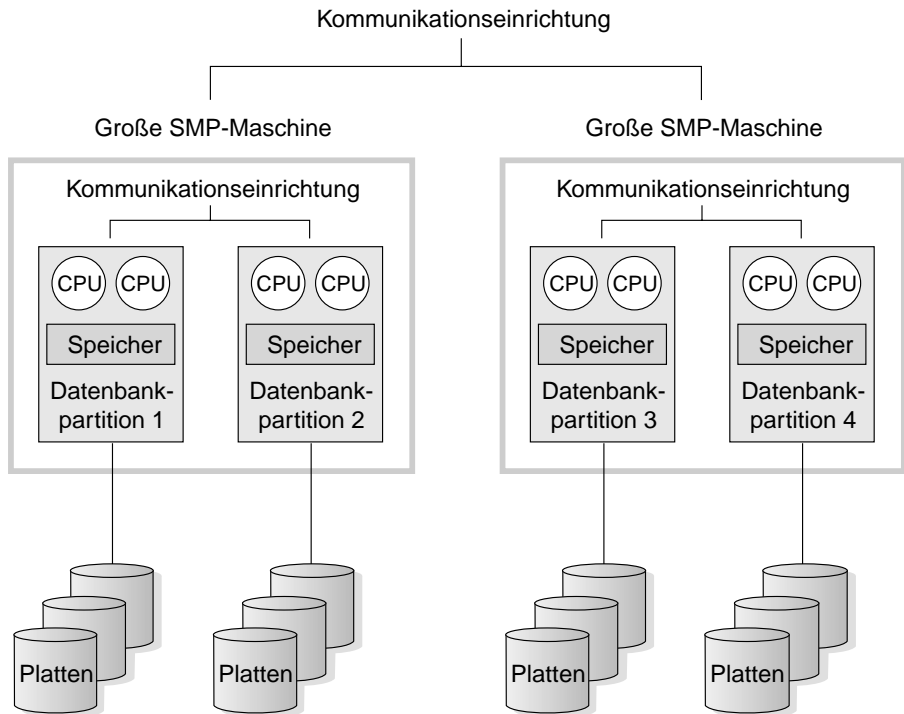


Abbildung 30. Partitionierte Datenbank auf in Gruppen zusammengefaßten SMP-Systemen

In Tabelle 3 auf Seite 86 sind die Arten der Parallelität aufgeführt, die am besten in dieser Hardwareumgebung genutzt werden können.

Anmerkung: Die Möglichkeit, zwei oder mehr Partitionen nebeneinander auf derselben Maschine zu haben (ungeachtet der Anzahl der Prozessoren), bietet eine größere Flexibilität bei der Einrichtung hochverfügbarer Konfigurationen und der Implementierung von Übernahmestrategien bei Systemausfällen. Bei Ausfall einer Maschine kann eine Datenbankpartition automatisch von einer anderen Maschine, die bereits eine andere Partition derselben Datenbank enthält, übernommen und wieder gestartet werden. Weitere Informationen finden Sie in „Kapitel 11. Entwerfen für hohe Verfügbarkeit“ auf Seite 233.

Zusammenfassung der am besten geeigneten Arten von Parallelität für jede Hardwareumgebung

Die folgende Tabelle gibt eine Übersicht über die Arten der Parallelität, mit denen die verschiedenen Hardwareumgebungen am besten genutzt werden können.

Tabelle 3. Mögliche Arten der Parallelität in jeder Hardwareumgebung

Hardwareumgebung	E/A-Parallelität	Abfrageinterne Parallelität	
		Partitionsinterne Parallelität	Partitionsübergreifende Parallelität
Einzelpartition, Einzelprozessor	Ja	Nein (1)	Nein
Einzelpartition, mehrere Prozessoren (SMP)	Ja	Ja	Nein
Mehrere Partitionen, ein Prozessor (MPP)	Ja	Nein (1)	Ja
Mehrere Partitionen, mehrere Prozessoren (Cluster von SMP-System)	Ja	Ja	Ja
Logische Datenbankpartitionen	Ja	Ja	Ja
<p>Anmerkung: (1) Es kann vorteilhaft sein, den Grad der Parallelität (mit Hilfe eines der Konfigurationsparameter) auch auf einem Einzelprozessorsystem auf einen Wert größer als 1 zu setzen, besonders wenn die von Ihnen ausgeführten Abfragen die CPU nicht voll auslasten (z. B. wenn die Abfragen E/A-lastig sind).</p>			

Kapitel 5. Data Warehouses

DB2 Universal Database stellt die Data Warehouse-Zentrale zur Verfügung, eine Komponente, die die Data Warehouse-Verarbeitung automatisiert. Mit Hilfe der Data Warehouse-Zentrale können Sie die Daten definieren, die in das Warehouse aufgenommen werden sollen. Anschließend können Sie über die Data Warehouse-Zentrale automatisch Aktualisierungen der Daten im Warehouse per Zeitplan terminieren.

Dieser Abschnitt enthält eine Übersicht über den Einsatz von Data Warehouses und Data Warehouse-Funktionen. Genauere Informationen zu Data Warehouses und Informationen zur Data Warehouse-Zentrale finden Sie im Handbuch *Data Warehouse-Zentrale Verwaltung* und in der Online-Hilfefunktion der Data Warehouse-Zentrale.

Was ist ein Data Warehouse?

Die Systeme mit *Betriebsdaten* (d. h. Daten, die für die täglichen Transaktionen des Geschäftsbetriebs benötigt werden) enthalten Informationen, die für Geschäftsanalysen nützlich sind. Zum Beispiel können Analytiker Informationen darüber nutzen, welche Produkte in welchen Gebieten zu welcher Jahreszeit verkauft wurden, um Abweichungen zu erkennen oder zukünftige Verkäufe zu projektieren.

Jedoch ergeben sich beim direkten Zugriff auf Betriebsdaten durch Analytiker verschiedene Probleme:

- Sie verfügen möglicherweise nicht über die Fachkenntnisse zum Abfragen der Betriebsdatenbank. Zum Beispiel ist zum Abfragen von IMS-Datenbanken ein Anwendungsprogramm erforderlich, das eine spezielle Art von Datenbearbeitungssprache verwendet. Im allgemeinen haben diejenigen Programmierer, die über die nötigen Fachkenntnisse zum Abfragen der Betriebsdatenbank verfügen, eine Vollzeitbeschäftigung mit der Verwaltung der Datenbank und der zugehörigen Anwendungen.
- Die Leistung ist für viele Betriebsdatenbanken, wie zum Beispiel Datenbanken einer Bank, von entscheidender Bedeutung. Das System kann Benutzer, die Sofortabfragen ausführen, nicht bedienen.
- Die Betriebsdaten liegen in der Regel nicht in dem zur Geschäftsanalyse günstigsten Format vor. Zum Beispiel sind Verkaufsdaten, die nach Produkt, Gebiet und Jahreszeit zusammengefaßt sind, wesentlich nützlicher für Analytiker als die Rohdaten.

Data Warehouses schaffen diesen Problemen Abhilfe. In *Data Warehouses* werden Speicher *informativer Daten* eingerichtet, d. h. Daten, die aus den Betriebsdaten extrahiert und zur Entscheidungsfindung für Endbenutzer aufbereitet werden. Zum Beispiel könnte ein Data Warehouse-Tool alle Verkaufsdaten aus der Betriebsdatenbank kopieren, Berechnungen durchführen, um die Daten zusammenzufassen, und die zusammengefaßten Daten in eine Zieltabelle einer von der Betriebsdatenbank getrennten Datenbank schreiben. Die getrennte Datenbank (das sog. *Warehouse*) kann dann von Endbenutzern abgefragt werden, ohne die Betriebsdatenbanken zu beeinträchtigen.

In den folgenden Abschnitten werden Objekte (Themenbereiche, Warehouse-Quellen, Warehouse-Ziele, Agenten, Agenten-Sites, Schritte und Prozesse) beschrieben, die zur Erstellung und Pflege eines Data Warehouse verwendet werden.

Themenbereiche

Ein *Themenbereich* identifiziert und gruppiert die Prozesse, die zu einem logischen Bereich des Geschäftsbetriebs gehören. Wenn Sie zum Beispiel ein Warehouse mit Marketing- und Verkaufsdaten aufbauen, können Sie einen Themenbereich Verkauf und einen Themenbereich Marketing definieren. Anschließend können Sie die Prozesse, die zum Verkauf gehören, unter dem Themenbereich Verkauf hinzufügen. Analog können Sie die Definitionen, die sich auf die Marketing-Daten beziehen, unter dem Themenbereich Marketing hinzufügen.

Warehouse-Quellen

Warehouse-Quellen geben die Tabellen und Dateien an, denen die Daten für das Warehouse entnommen werden. Die Data Warehouse-Zentrale verwendet die Angaben in den Warehouse-Quellen, um auf Daten zuzugreifen und sie auszuwählen. Die Quellen können beinahe jede relationale oder nicht relationale Quelle (Tabelle, Sicht oder Datei) sein, von der eine Verbindung zum Warehouse hergestellt werden kann.

Warehouse-Ziele

Warehouse-Ziele sind Datenbanktabellen oder Dateien, die Daten enthalten, die zur Verwendung durch Endbenutzer aufbereitet wurden. Ähnlich wie Warehouse-Quellen können auch Warehouse-Ziele Daten für Schritte der Data Warehouse-Zentrale bereitstellen.

Warehouse-Agenten und Agenten-Sites

Agenten der Data Warehouse-Zentrale verwalten den Datenfluß zwischen den Datenquellen und den Ziel-Warehouses. Agenten sind für die Betriebssysteme Windows NT, AIX, OS/2, OS/390, OS/400 und SUN Solaris verfügbar. Diese Agenten verwenden ODBC-Treiber (Open Database Connectivity) oder DB2 CLI zur Kommunikation mit verschiedenen Datenbanken.

Die Übertragung von Daten zwischen Quellen und Ziel-Warehouses kann von mehreren Agenten übernommen werden. Die Anzahl der Agenten, die Sie verwenden, hängt von der vorhandenen Konnektivitätskonfiguration und dem geplanten Volumen der Daten ab, die in das Warehouse fließen sollen. Zusätzliche Exemplare eines Agenten können generiert werden, wenn mehrere Prozesse, die den gleichen Agenten benötigen, gleichzeitig ausgeführt werden.

Agenten können lokal oder fern sein. Ein *lokaler Warehouse-Agent* ist ein Agent, der auf derselben Maschine wie der Warehouse-Server installiert ist. Ein *ferner Warehouse-Agent* ist ein Agent, der auf einer anderen Maschine installiert ist, die über eine Konnektivität zum Warehouse-Server verfügt.

Eine *Agenten-Site* ist ein logischer Name für eine Workstation, auf der Agentensoftware installiert ist. Der Name der Agenten-Site ist nicht mit dem TCP/IP-Host-Namen identisch. Eine einzelne physische Maschine kann nur einen TCP/IP-Host-Namen besitzen. Jedoch können auf einer einzelnen Maschine mehrere Agenten-Sites definiert werden. Jede Agenten-Site wird durch einen logischen Namen angegeben.

Die *Standardagenten-Site*, die den Namen Standard-VW-Agenten-Site besitzt, ist ein lokaler Agent auf Windows NT, der von der Data Warehouse-Zentrale während der Initialisierung der Warehouse-Steuerungsdatenbank definiert wird.

Schritte und Prozesse

Ein *Schritt* ist eine logische Definitionseinheit in der Data Warehouse-Zentrale, die folgendes definiert:

- Die Struktur der Ausgabetable oder Ausgabedatei
- Den Mechanismus (entweder SQL oder ein Programm) zum Füllen der Ausgabetable oder Ausgabedatei
- Den Zeitplan, nach dem die Ausgabetable bzw. Ausgabedatei gefüllt wird

Schritte übertragen Daten und setzen sie um, indem sie SQL-Anweisungen ausführen oder Programme aufrufen. Wenn Sie einen Schritt ausführen, findet die Datenübertragung zwischen der Warehouse-Quelle und dem Warehouse-Ziel sowie die Umsetzung dieser Daten statt.

Ein *Prozeß* enthält eine Reihe von Schritten, die Umsetzungs- und Datenübertragungstasks ausführen. In der Regel füllt ein Prozeß ein Warehouse-Ziel in einer Warehouse-Datenbank, indem er Daten aus einer oder mehreren Warehouse-Quellen extrahiert, bei denen es sich um Datenbanktabellen oder Dateien handeln kann. Jedoch können Sie einen Prozeß auch zum Starten von Programmen definieren, wobei weder Warehouse-Quellen noch Warehouse-Ziele angegeben werden.

Ein Schritt kann auf Anforderung ausgeführt oder zur Ausführung zu einem bestimmten Zeitpunkt terminiert werden. Ein Schritt kann zur einmaligen Ausführung oder zur wiederholten Ausführung, z. B. jeden Freitag, eingeplant werden. Schritte können auch so terminiert werden, daß sie nacheinander ausgeführt werden, d. h., wenn ein Schritt beendet wird, wird der nächste Schritt gestartet. Schritte können zudem abhängig von der (erfolgreichen oder nicht erfolgreichen) Beendigung eines anderen Schritts eingeplant werden. Wenn Sie einen Prozeß über einen Zeitplan terminieren, wird der erste Schritt des Prozesses zur definierten Zeit ausgeführt.

Wenn ein Schritt oder ein Prozeß ausgeführt wird, kann er Daten auf folgende Arten speichern:

- Ersetzen aller Daten im Warehouse-Ziel durch neue Daten
- Anhängen der neuen Daten an die vorhandenen Daten
- Anhängen einer getrennten Edition von Daten

Nehmen Sie zum Beispiel an, daß die Data Warehouse-Zentrale die folgenden Aufgaben ausführen soll:

1. Extrahieren von Daten aus verschiedenen Datenbanken
2. Konvertieren der Daten in ein einziges Format
3. Schreiben der Daten in eine Tabelle in einem Data Warehouse

In diesem Beispiel würden Sie einen Prozeß erstellen, der einzelne Schritte enthält. Jeder Schritt würde eine getrennte Task, wie zum Beispiel das Extrahieren der Daten aus den Datenbanken oder die Konvertierung der Daten in das richtige Format, ausführen. Dann würden Sie einen weiteren Schritt verwenden, um die Zieltabelle zu füllen, die die umgesetzten Daten enthält.

In den folgenden Abschnitten werden die verschiedenen Arten von Schritten beschrieben, die in der Data Warehouse-Zentrale verfügbar sind. Weitere Informationen zu Schritten finden Sie im Handbuch *Data Warehouse-Zentrale Verwaltung*.

SQL-Schritte

Ein SQL-Schritt arbeitet mit einer SQL-Anweisung SELECT, um Daten aus einer Warehouse-Quelle zu extrahieren, und generiert eine Anweisung INSERT, um die Daten in die Warehouse-Zieltabelle einzufügen.

Programmschritte

Es gibt verschiedene Typen von Programmschritten: DB2 for AS/400-Programme, DB2 for OS/390-Programme, DB2 for UDB-Programme, Visual Warehouse 5.2 DB2-Programme, OLAP Server-Programme, Dateiprogramme und Replikation. Diese Schritte führen vordefinierte Programme und Dienstprogramme aus.

Umsetzungsschritte

Umsetzungsschritte sind gespeicherte Prozeduren und benutzerdefinierte Funktionen, die Umsetzungen für das Warehouse oder statistische Funktionen angeben, die zur Umsetzung von Daten verwendet werden können. Mit Umsetzungsprogrammen können Sie Daten bereinigen, umkehren und umlagern, Primärschlüssel und Periodentabellen generieren sowie verschiedene statistische Berechnungen durchführen.

In einem Umsetzungsschritt geben Sie eines der statistischen oder Warehouse-Umsetzungsprogramme an. Wenn der Prozeß ausgeführt wird, schreibt der Umsetzungsschritt Daten in ein oder mehrere Warehouse-Ziele.

Benutzerdefinierte Programmschritte

Ein *benutzerdefinierter Programmschritt* ist eine logische Definitionseinheit innerhalb der Data Warehouse-Zentrale, die eine Anwendung darstellt, die von der Data Warehouse-Zentrale gestartet werden soll. Ein Warehouse-Agent kann einen benutzerdefinierten Programmschritt zu folgenden Zeiten starten:

- Während des Füllens eines Warehouse-Ziels
- Nach dem Füllen eines Warehouse-Ziels
- Unabhängig

Zum Beispiel können Sie ein benutzerdefiniertes Programm schreiben, das den folgenden Prozeß ausführt:

1. Exportieren von Daten aus einer Tabelle
2. Bearbeiten dieser Daten
3. Schreiben der Daten in eine vorläufige Ausgaberesource oder ein Warehouse-Ziel

Aufgaben bei der Warehouse-Erstellung

Zur Erstellung eines Data Warehouse gehören folgende Aufgaben:

- Definieren eines Themenbereichs, der die Prozesse angibt und gruppiert, die in dem Warehouse verwendet werden sollen
- Erkunden der Quelldaten (bzw. Betriebsdaten) und Definieren von Warehouse-Quellen
- Erstellen einer Datenbank zur Verwendung als Warehouse und Definieren von Warehouse-Zielen
- Angeben, wie Quelldaten übertragen und in das richtige Format für die Warehouse-Datenbank umgesetzt werden, durch Definieren eines Prozesses
- Testen der definierten Schritte und Festlegen eines Zeitplans zur automatischen Ausführung
- Verwalten des Warehouse durch Definieren von Sicherheit und Überwachen der Datenbanknutzung

- Erstellen eines Informationskatalogs der Daten im Warehouse, falls das Paket DB2 Warehouse Manager eingesetzt wird. Ein Informationskatalog ist eine Datenbank, die Metadaten über den Geschäftsbetrieb enthält. Die Metadaten unterstützen Benutzer bei der Angabe und Auffindung von Daten und Informationen, die ihnen in der Organisation zur Verfügung stehen. Endbenutzer des Warehouse können den Katalog durchsuchen, um festzustellen, welche Tabellen abzufragen sind.
- Definieren eines Sternschemenmodells für die Daten im Warehouse. Ein Sternschema ist ein spezieller Aufbau, der aus mehreren Dimensionstabellen (die Aspekte eines Unternehmens beschreiben) und einer Faktttabelle (die Fakten über das Unternehmen enthält) besteht. Wenn Sie zum Beispiel Getränke herstellen, enthalten einige Dimensionstabellen Daten zu Produkten, Märkten und Zeit. Die Faktttabelle enthält Transaktionsinformationen über die Produkte, die in jedem Gebiet je nach Jahreszeit bestellt werden.
- Verknüpfen der Faktttabelle und der Dimensionstabellen, um die Einzeldaten aus den Dimensionstabellen mit den Bestellinformationen zu kombinieren. Zum Beispiel könnten Sie die Produktdimension mit der Faktttabelle verknüpfen, um Informationen darüber hinzuzufügen, wie jedes Produkt entsprechend den Bestellungen verpackt wurde.

Weitere Informationen zu diesen Aufgaben und anderen Themen erhalten Sie, indem Sie das *Lernprogramm für das Informationsmanagement* durcharbeiten, sich, den *DB2 Universal Database Kurzüberblick* ansehen oder die Informationen im Handbuch *Data Warehouse-Zentrale Verwaltung* nachlesen.

Kapitel 6. Informationen zu Spatial Extender

In diesem Abschnitt wird Spatial Extender vorgestellt. Dazu werden der Zweck des Produkts erläutert und die Daten beschrieben, die von ihm verarbeitet werden. Detaillierte Informationen zur Verwendung von Spatial Extender finden Sie im Handbuch *Spatial Extender Benutzer- und Referenzhandbuch*.

Der Zweck von Spatial Extender

Mit Spatial Extender kann ein *geografisches Informationssystem* (GIS) erstellt werden: ein Komplex von Objekten, Daten und Anwendungen, die das Generieren und Analysieren räumlicher Informationen über geografische Merkmale ermöglicht. *Geografische Merkmale* sind zum Beispiel die Objekte, die die Oberfläche der Erde bilden, und Objekte, die sie bedecken. Sie stellen sowohl die natürliche Umwelt (zum Beispiel Flüsse, Wälder, Hügel und Wüsten) als auch die kulturelle Umwelt (Städte, Wohnhäuser, Bürogebäude, Landmarken usw.) dar.

Räumliche Informationen sind Fakten wie zum Beispiel folgende:

- Die Position geografischer Merkmale in Bezug auf ihre Umgebung (zum Beispiel Punkte in einer Stadt, an denen sich Krankenhäuser und Kliniken befinden, oder die Nähe von Wohngebieten der Stadt zu lokalen Erdbebenzonen)
- Arten von Beziehungen, in denen die geografische Merkmale miteinander stehen (zum Beispiel Informationen, daß ein bestimmtes Flußsystem in einer bestimmten Region eingeschlossen ist oder daß bestimmte Brücken in dieser Region Nebenflüsse des Flußsystems überqueren)
- Maßangaben, die für ein oder mehrere geografische Merkmale gelten (zum Beispiel die Entfernung zwischen einem Bürogebäude und seiner Parzellengrenze, oder die Länge der Grenzlinie eines Vogelschutzgebiets)

Räumliche Informationen können entweder als solche oder in Kombination mit herkömmlichen Datenbankausgaben bei der Entwicklung von Projekten sowie bei der geschäftlichen und geschäftspolitischen Entscheidungsfindung hilfreich sein. Nehmen Sie zum Beispiel an, daß ein leitender Mitarbeiter des Bezirkssozialamts überprüfen muß, ob Antragsteller und Empfänger von Sozialhilfe tatsächlich in dem Bezirk wohnen, für den das Amt zuständig ist. Spatial Extender kann diese Informationen aus der Lage des Bezirks und aus den Adressen der Antragsteller und Empfänger ableiten.

Oder nehmen Sie an, der Besitzer einer Restaurantkette plant, Geschäfte in nahegelegenen Städten zu eröffnen. Zur Bestimmung, wo die neuen Restau-

rants zu eröffnen sind, benötigt der Besitzer Antworten auf solche Fragen wie: In welchen Teilen dieser Städte gibt es Konzentrationen der Zielgruppe von Menschen, die solche Restaurants besuchen würden? Wo verlaufen die größten Straßen? Wo ist die Kriminalitätsrate am niedrigsten? Wo befinden sich Konkurrenzunternehmen? Spatial Extender kann räumliche Informationen in visuellen Anzeigen darstellen, um solche Fragen zu beantworten, und das zugrundeliegende Datenbankverwaltungssystem kann Beschriftungen und Text zur Erläuterung der Anzeigen generieren.

Daten zur Darstellung geografischer Merkmale

Dieser Abschnitt enthält eine Übersicht über die Daten, die zur Gewinnung räumlicher Informationen generiert, gespeichert und verarbeitet werden können. Folgende Themen werden behandelt:

- Wie Daten geografische Merkmale darstellen
- Die Eigenschaften räumlicher Daten
- Möglichkeiten zur Erstellung räumlicher Daten

Wie Daten geografische Merkmale darstellen

In Spatial Extender kann ein geografisches Merkmal durch eine Zeile in einer Tabelle oder Sicht bzw. durch einen Teil einer solchen Zeile dargestellt werden. Betrachten Sie zum Beispiel die beiden folgenden geografischen Merkmale: Bürogebäude und Wohnungen. In Abb. 31 auf Seite 95 stellt jede Zeile der Tabelle BRANCHES eine Zweigstelle einer Bank dar. Abweichend davon stellt jede Zeile der Tabelle CUSTOMERS als Ganzes einen Kunden der Bank dar. Jedoch kann ein Teil jeder Zeile, insbesondere die Zellen mit der

Adresse eines Kunden, als Darstellung der Wohnung des Kunden aufgefaßt werden.

Diese Tabellen enthalten Daten, die die Zweigstellen und Kunden der Bank

BRANCHES

ID	NAME	ADDRESS	CITY	STATE	ZIP
937	Airzone-Multern	92467 Airzone Blvd	San Jose	CA	95141

CUSTOMERS

ID	LAST NAME	FIRST NAME	ADDRESS	CITY	STATE	ZIP	CHECKING	SAVINGS
59-6396	Kriner	Endela	9 Concourt Circle	San Jose	CA	95141	A	A

Abbildung 31. Tabellenzeile, die ein geografisches Merkmal darstellt. Tabellenzeile, deren Adreßdaten ein geografisches Merkmal darstellen. Die Datenzeile in der Tabelle BRANCHES stellt eine Zweigstelle einer Bank dar. Die Zellen für Adreßdaten in der Tabelle CUSTOMERS stellen die Wohnung eines Kunden dar.

identifizieren und beschreiben. Solche Daten werden als *attributive Daten* bezeichnet.

Eine Untermenge der attributiven Daten (die Werte, die Zweigstellen- oder Kundenadressen darstellen) kann in Werte umgesetzt werden, die räumliche Informationen zur Verfügung stellen. Wie zum Beispiel in der Abbildung zu sehen ist, lautet eine Zweigstellenadresse 92467 Airzone Blvd., San Jose CA 95141. Eine Kundenadresse lautet 9 Concourt Circle, San Jose CA 95141. Spatial Extender kann diese Adressen in Werte umsetzen, die angeben, wo sich die Zweigstelle und die Wohnung des Kunden in Bezug auf die jeweilige Umgebung befinden. Abb. 32 zeigt die Tabellen BRANCHES und CUSTOMERS mit neuen Spalten, die solche Werte enthalten.

BRANCHES

ID	NAME	ADDRESS	CITY	STATE	ZIP	LOCATION
937	Airzone-Multern	92467 Airzone Blvd	San Jose	CA	95141	

CUSTOMERS

ID	LAST NAME	FIRST NAME	ADDRESS	CITY	STATE	ZIP	LOCATION	CHECKING	SAVINGS
59-6396	Kriner	Endela	9 Concourt Circle	San Jose	CA	95141		A	A

Abbildung 32. Tabellen mit hinzugefügten Spalten für räumliche Daten. In jeder Tabelle ist die Spalte LOCATION für Koordinaten vorgesehen, die den Adressen entsprechen.

Wenn Adressen und ähnliche Angaben als Ausgangspunkt für räumliche Informationen genutzt werden, werden sie als *Quellendaten* bezeichnet. Da die Werte, die aus ihnen abgeleitet werden, räumliche Informationen zur Verfügung stellen, werden diese abgeleiteten Werte als *räumliche Daten* bezeichnet. Im nächsten Abschnitt werden räumliche Daten beschrieben und die ihnen zugehörigen Datentypen vorgestellt.

Die Eigenschaften räumlicher Daten

Ein wesentlicher Teil räumlicher Daten besteht aus Koordinaten. Eine *Koordinate* ist eine numerische Angabe einer Position in Relation zu einem Bezugspunkt. Zum Beispiel sind Breitengrade Koordinaten, die Positionen in Relation zum Äquator angeben. Längengrade sind Koordinaten, die Positionen in Relation zum Nullmeridian von Greenwich angeben. Auf diese Weise ist die Lage des Yellowstone National Park durch die geografische Breite (44,45 Grad nördlich des Äquators) und die geografische Länge (110,40 Grad westlich des Nullmeridians von Greenwich) definiert.

Längen, Breiten und ihre Bezugspunkte sowie andere zugehörige Parameter werden kollektiv als *Koordinatensystem* bezeichnet. Es gibt außerdem Koordinatensysteme, die auf anderen Werten als geografische Länge und Breite basieren. Diese Koordinatensysteme verfügen über eigene Positionsmaße, Bezugspunkte und zusätzliche distinktive Parameter.

Das einfachste räumliche Datenelement besteht aus zwei Koordinaten, die die Position eines einzelnen geografischen Merkmals definieren. Der Ausdruck *Datenelement* bezieht sich auf den Wert oder die Werte, die eine Zelle in einer relationalen Tabelle einnehmen. Ein umfassenderes räumliches Datenelement besteht aus mehreren Koordinaten, die einen linearen Pfad, zum Beispiel eine Straße oder einen Flußlauf, definieren. Eine dritte Art besteht aus Koordinaten, die den Umfang eines Gebiets, zum Beispiel die Grenzlinie einer Landparzelle oder einer Schwemmebene definieren.

Jedes räumliche Datenelement ist ein Exemplar eines räumlichen Datentyps. Der Datentyp für zwei Koordinaten, die eine Position definieren, ist `ST_Point`, der Datentyp für Koordinaten, die lineare Pfade definieren, ist `ST_LineString` und der Datentyp für Koordinaten, die Umfänge definieren, ist `ST_Polygon`. Diese Typen sind, zusammen mit den anderen Datentypen für räumliche Daten, strukturierte Datentypen, die zu einer Hierarchie gehören.

Möglichkeiten zur Erstellung räumlicher Daten

Räumliche Daten können auf folgende Arten gewonnen werden:

- Ableiten aus attributiven Daten
- Ableiten aus anderen räumlichen Daten
- Importieren

Verwenden attributiver Daten als Quelldaten

Das Ableiten räumlicher Daten aus attributiven Daten (z. B. Adressen) wird als *geografische Codierung (Geocoding)* bezeichnet. In Abb. 32 auf Seite 95 sind zwei Spalten zu sehen, eine in der Tabelle BRANCHES und eine in der Tabelle CUSTOMERS, die für räumliche Daten vorgesehen sind. Stellen Sie sich vor, daß Spatial Extender die Adressen in diesen Tabellen geografisch codiert und das Ergebnis (Koordinaten, die den Adressen entsprechen) in die Spalten einfügt. Abb. 33 veranschaulicht das Ergebnis.

BRANCHES

ID	NAME	ADDRESS	CITY	STATE	ZIP	LOCATION
937	Airzone-Multern	92467 Airzone Blvd	San Jose	CA	95141	1653 3094

CUSTOMERS

ID	LAST NAME	FIRST NAME	ADDRESS	CITY	STATE	ZIP	LOCATION	CHECKING	SAVINGS
59-6396	Kriner	Endela	9 Concourt Circle	San Jose	CA	95141	953 1527	A	A

Abbildung 33. Tabellen, die aus Quelldaten abgeleitete räumliche Daten enthalten. Die Spalte LOCATION in der Tabelle CUSTOMERS enthält Koordinaten, die durch eine geografische Codierung aus der Adresse in den Spalten ADDRESS, CITY, STATE und ZIP abgeleitet wurden. Analog enthält die Spalte LOCATION in der Tabelle BRANCHES Koordinaten, die durch eine geografische Codierung aus der Adresse in den Spalten ADDRESS, CITY, STATE und ZIP dieser Tabelle abgeleitet wurden.

Spatial Extender arbeitet mit einer Funktion namens *geocoder*, um attributive Daten geografisch zu codieren und die resultierenden räumlichen Daten in Spalten abzulegen.

Verwenden anderer räumlicher Daten als Quelldaten

Räumliche Daten können nicht nur aus attributiven Daten, sondern auch aus anderen räumlichen Daten generiert werden. Nehmen Sie zum Beispiel an, daß die Bank, deren Zweigstellen in der Tabelle BRANCHES definiert sind, in Erfahrung bringen will, wie viele Kunden innerhalb einer Entfernung von fünf Meilen von einer Zweigstelle wohnen. Bevor die Bank diese Informationen aus der Datenbank erhalten kann, muß sie der Datenbank die Definition einer Zone hinzufügen, die innerhalb eines Radius von fünf Meilen um jede Zweigstelle liegt. Eine Spatial Extender-Funktion namens *ST_Buffer* kann eine solche Definition erstellen. Mit Hilfe der Koordinaten der einzelnen Zweigstellen als Eingabe kann diese Funktion Koordinaten generieren, die die Grenzlinien der gewünschten Zonen festlegen. Abb. 34 auf Seite 98 zeigt die Tabelle BRANCHES mit Informationen, die von *ST_Buffer* geliefert wurden.

BRANCHES

ID	NAME	ADDRESS	CITY	STATE	ZIP	LOCATION	SALES_AREA
937	Airzone-Multern	92467 Airzone Blvd	San Jose	CA	95141	1653 3094	1002 2001, 1192 3564, 2502 3415, 1915 3394, 1002 2001

Abbildung 34. Tabelle mit neuen räumlichen Daten, die aus vorhandenen räumlichen Daten abgeleitet wurden. Die Koordinaten in der Spalte SALES_AREA wurden durch die Funktion ST_Buffer aus den Koordinaten in der Spalte LOCATION abgeleitet.

Neben der Funktion ST_Buffer bietet Spatial Extender auch verschiedene andere Funktionen, die neue räumliche Daten aus vorhandenen räumlichen Daten ableiten.

Importieren räumlicher Daten

Eine dritte Methode zur Gewinnung räumlicher Daten ist das Importieren der Daten aus Dateien, die eines der von Spatial Extender unterstützten Formate besitzen. Diese Dateien enthalten Daten, die in der Regel auf Karten angewendet werden: Planungslinien, Schwemmebenen, Erdbebenverwerfungen usw. Durch die kombinierte Verwendung solcher Daten mit räumlichen Daten, die erstellt werden, können Sie die Ihnen zur Verfügung stehenden geografischen Informationen erweitern. Wenn zum Beispiel eine Bebauungsbehörde die Risiken bestimmen muß, denen ein Wohngebiet ausgesetzt ist, kann sie mit Hilfe von ST_Buffer eine Zone um das Wohngebiet definieren und anschließend Daten über Schwemmebenen und Erdbebenverwerfungen importieren, um festzustellen, welche dieser Problembereiche die Zone überlappen.

Teil 3. Datenbankentwurf

Kapitel 7. Entwerfen des logischen Datenbankaufbaus

Dieser Abschnitt behandelt die folgenden Schritte zum Entwurf einer Datenbank:

- „Festlegen der in der Datenbank aufzuzeichnenden Daten“
- „Definieren von Tabellen für jede Art von Relation“ auf Seite 103
- „Definieren der Spalten für alle Tabellen“ auf Seite 106
- „Definieren einer oder mehrerer Spalten als Primärschlüssel“ auf Seite 108
- „Sicherstellen, daß gleiche Werte die gleiche Entität darstellen“ auf Seite 112
- „Normalisieren der Tabellen“ auf Seite 113
- „Planen der Integritätsbedingungen“ auf Seite 118
- „Weitere Überlegungen zum Datenbankentwurf“ auf Seite 127

Das Ziel des Entwurfs einer Datenbank ist es, eine Darstellung der Umgebung zu finden, die leicht verständlich ist und als Basis für zukünftige Erweiterungen dienen kann. Darüber hinaus soll der Aufbau der Datenbank die Wahrung der Konsistenz und der Integrität der Daten unterstützen. Dies wird durch eine Datenbankstruktur erreicht, mit deren Hilfe Redundanz verringert und Anomalien eliminiert werden, die während der Aktualisierung der Datenbank auftreten können.

Diese Schritte sind Teil des *logischen* Datenbankentwurfs. Der Entwurf einer Datenbank ist kein linearer Prozeß. Wahrscheinlich müssen im Laufe des Entwurfs Schritte wiederholt ausgeführt oder überarbeitet werden.

Die *physische* Implementierung des Datenbankentwurfs wird in „Kapitel 8. Entwerfen der physischen Datenbank“ auf Seite 131 und „Implementieren des Datenbankentwurfs“ im Handbuch *Systemverwaltung: Implementierung* beschrieben.

Festlegen der in der Datenbank aufzuzeichnenden Daten

Der erste Schritt bei der Entwicklung eines Datenbankentwurfs ist die Festlegung, welche Typen von Daten in den Tabellen der Datenbank gespeichert werden sollen. Eine Datenbank umfaßt Informationen über die *Entitäten* innerhalb einer Organisation oder Firma sowie über die Beziehungen bzw. Abhängigkeiten zwischen ihnen. In einer relationalen Datenbank werden *Entitäten* als *Tabellen* dargestellt.

Eine *Entität* ist eine Person, ein Objekt oder ein Begriff, über die Informationen gespeichert werden sollen. Einige der Entitäten, die in den Beispiel-

tabellen beschrieben werden, sind zum Beispiel Mitarbeiter, Abteilungen und Projekte. (Eine Beschreibung der Beispieldatenbank finden Sie im Handbuch *SQL Reference*.)

In der Beispieltabelle EMPLOYEE hat die Entität "EMPLOYEE" (Mitarbeiter) bestimmte *Attribute* oder Merkmale, wie Personalnummer, Name, Abteilung (Work Department) und Gehalt (Salary). Diese Merkmale bilden die *Spalten* EMPNO, FIRSTNAME, LASTNAME, WORKDEPT und SALARY der Tabelle.

Ein konkretes *Vorkommen* der Entität "Mitarbeiter" (employee) besteht aus den Werten aller Spalten für einen Mitarbeiter. Jeder Mitarbeiter hat eine eindeutige Personalnummer (EMPNO), die zur Identifizierung dieses Vorkommens der Entität „Mitarbeiter“ verwendet werden kann. Jede Zeile in einer Tabelle stellt ein Vorkommen einer Entität oder Relation dar. Die Werte in der ersten Zeile der folgenden Tabelle beschreiben beispielsweise eine Mitarbeiterin namens Haas.

Tabelle 4. Vorkommen der Entität Mitarbeiter und zugehöriger Attribute

EMPNO	FIRSTNAME	LASTNAME	WORKDEPT	JOB
000010	Christine	Haas	A00	President
000020	Michael	Thompson	B01	Manager
000120	Sean	O'Connell	A00	Clerk
000130	Dolores	Quintana	C01	Analyst
000030	Sally	Kwan	C01	Manager
000140	Heather	Nicholls	C01	Analyst
000170	Masatoshi	Yoshimura	D11	Designer

Es gibt einen wachsenden Bedarf an Unterstützung für neuere Datenbankanwendungen wie zum Beispiel Multimediaanwendungen. Es kann sinnvoll sein, Attribute zur Unterstützung von Multimediaobjekten, z. B. Dokumenten, Videodaten oder gemischten Medien, Bildern und Tondaten, zu berücksichtigen.

Innerhalb einer Tabelle hat jede Spalte einer Zeile eine bestimmte Relation oder Beziehung zu allen anderen Spalten dieser Zeile. Einige Relationen, die in den Beispieltabellen zum Ausdruck kommen, sind folgende:

- Mitarbeiter sind Abteilungen zugeordnet.
 - Dolores Quintana gehört zu Abteilung C01 (WORKDEPT).
- Mitarbeiter führen eine Jobbezeichnung.
 - Dolores arbeitet als „Analyst“.
- Abteilungen berichten anderen Abteilungen.
 - Abteilung C01 berichtet an Abteilung A00.

- Abteilung B01 berichtet an Abteilung A00.
- Mitarbeiter arbeiten an Projekten.
 - Dolores und Heather arbeiten beide an Projekt IF1000.
- Mitarbeiter leiten Abteilungen.
 - Sally leitet die Abteilung C01.

"Mitarbeiter (Employee) und "Abteilung" (Department) sind Entitäten. Sally Kwan ist Teil eines Vorkommens der Entität "Mitarbeiter", und C01 ist Teil des Vorkommens der Entität "Abteilung". Für dieselben Spalten in jeder Zeile einer Tabelle gelten dieselben Arten von Relationen. Zum Beispiel gibt eine Zeile einer Tabelle die Relation an, daß Sally Kwan die Abteilung C01 leitet. Eine andere Zeile enthält die Relation, daß Sean O'Connell Sachbearbeiter (Clerk) in der Abteilung A00 ist.

Die Informationen innerhalb einer Tabelle sind von der Art der Relation, die dargestellt werden soll, vom Maß der benötigten Flexibilität und von der gewünschten Geschwindigkeit, mit der die Daten abgerufen werden können, abhängig.

Neben der Feststellung, welche Beziehungen zwischen Entitäten innerhalb Ihres Unternehmens in der Datenbank erfaßt werden sollen, müssen auch andere Arten von Informationen bestimmt werden, z. B. die Geschäftsregeln, denen die Daten unterliegen.

Definieren von Tabellen für jede Art von Relation

In einer Datenbank können verschiedene Arten von Beziehung definiert werden. Betrachten Sie zum Beispiel die möglichen Relationen zwischen Mitarbeitern (Employees) und Abteilungen (Departments). Ein Mitarbeiter kann in nur einer Abteilung arbeiten. Eine solche Relation ist *einwertig* für den Mitarbeiter. Auf der anderen Seite kann eine Abteilung mehrere Mitarbeiter haben. Diese Relation ist *mehrwertig* für die Abteilungen. Die Relation zwischen Mitarbeiter (einwertig) und Abteilungen (mehrwertig) ist eine *Eins-zu-viele-Beziehung*. In diesem Abschnitt werden die folgenden Arten von Beziehungen behandelt:

- „Eins-zu-viele- und Viele-zu-eins-Beziehungen“
- „Viele-zu-viele-Beziehungen“ auf Seite 104
- „Eins-zu-eins-Beziehungen“ auf Seite 105

Eins-zu-viele- und Viele-zu-eins-Beziehungen

Zur Definition von Tabellen für die einzelnen Eins-zu-viele- und Viele-zu-eins-Beziehungen empfiehlt sich folgende Methode:

1. Fassen Sie alle Beziehungen in Gruppen zusammen, bei denen jeweils die "Viele"-Seite der Relation dieselbe Entität ist.

2. Definieren Sie eine Tabelle für alle Relationen in der Gruppe.

Im folgenden Beispiel ist die Viele-Seite der ersten und zweiten Relation "Employees" (Mitarbeiter), so daß eine Tabelle für die Mitarbeiter (Tabelle EMPLOYEE) definiert wird.

Tabelle 5. Viele-zu-eins-Beziehungen

Entität	Beziehung	Entität
Mitarbeiter	sind zugeordnet zu	Abteilungen
Mitarbeiter	arbeiten als	Jobbezeichnung
Abteilungen	berichten an	(Verwaltungs-) Abteilungen

In der dritten Relation ist die Entität "Departments" (Abteilungen) auf der Viele-Seite, so daß eine Tabelle DEPARTMENT definiert wird.

Die folgenden Tabellen veranschaulichen, wie diese Relationen dargestellt werden:

Die Tabelle EMPLOYEE:

EMPNO	WORKDEPT	JOB
000010	A00	President
000020	B01	Manager
000120	A00	Clerk
000130	C01	Analyst
000030	C01	Manager
000140	C01	Analyst
000170	D11	Designer

Die Tabelle DEPARTMENT:

DEPTNO	ADMRDEPT
C01	A00
D01	A00
D11	D01

Viele-zu-viele-Beziehungen

Eine Relation, die in beiden Richtungen mehrwertig ist, wird als eine Viele-zu-viele-Beziehung bezeichnet. Ein Mitarbeiter (Employee) kann an mehr als einem Projekt arbeiten, und ein Projekt kann mehr als einen Mitarbeiter

beschäftigen. Auf die Fragen „Woran arbeitet Dolores Quintana?“ und „Wer arbeitet an Projekt IF1000?“ gibt es jeweils mehrere Antworten. Eine Viele-zu-viele-Beziehung kann in einer Tabelle mit einer Spalte für jede Entität (Mitarbeiter „EMPNO“ und Projekt „PROJNO“) wie im folgenden Beispiel dargestellt werden.

Die folgende Abbildung zeigt, wie eine Viele-zu-viele-Beziehung (ein Mitarbeiter kann an vielen Projekten arbeiten, und ein Projekt kann viele Mitarbeiter beschäftigen) in einer Tabelle dargestellt werden kann:

Die Tabelle EMP_ACT (Mitarbeitertätigkeiten):

EMPNO	PROJNO
000030	IF1000
000030	IF2000
000130	IF1000
000140	IF2000
000250	AD3112

Eins-zu-eins-Beziehungen

Eins-zu-eins-Beziehungen sind in beide Richtungen einwertig. Ein Manager leitet eine Abteilung. Eine Abteilung hat nur einen Manager. Auf die Fragen „Wer ist der Manager der Abteilung C01?“ und „Welche Abteilung wird von Sally Kwan geleitet?“ gibt es jeweils nur eine Antwort. Die Beziehung kann entweder der Tabelle DEPARTMENT oder der Tabelle EMPLOYEE zugeordnet werden. Da alle Abteilungen Manager haben, aber nicht alle Mitarbeiter Manager sind, ist die logische Konsequenz, die Manager der Tabelle DEPARTMENT wie im folgenden Beispiel zuzuordnen.

Die folgende Abbildung zeigt, wie eine Eins-zu-eins-Beziehung in einer Tabelle dargestellt wird:

Die Tabelle DEPARTMENT:

DEPTNO	MGRNO
A00	000010
B01	000020
D11	000060

Definieren der Spalten für alle Tabellen

Eine Spalte in einer relationalen Tabelle wird folgendermaßen definiert:

1. Wählen Sie einen Namen für die Spalte.

Jede Spalte in einer Tabelle muß einen für diese Tabelle eindeutigen Namen haben. Einzelheiten zu den Spaltennamen finden Sie in „Anhang B. Namenskonventionen“ auf Seite 399.

2. Definieren Sie die Art der Daten, die für die Spalte gültig sein sollen.

Der *Datentyp* und die *Länge* legen den Typ und die maximale Länge der Daten fest, die als Werte der Spalte gültig sind. Datentypen können aus den im Datenbankmanager vordefinierten Typen ausgewählt oder vom Benutzer als sogenannte benutzerdefinierte Typen selbst erstellt werden. Informationen über die von DB2 bereitgestellten Datentypen und über benutzerdefinierte Typen finden Sie im Handbuch *SQL Reference*.

Kategorien von Datentypen sind zum Beispiel: numerische Daten, Zeichenfolge, Doppelbytezeichenfolge (oder Grafikzeichenfolge), Datum/Uhrzeit und Binärzeichenfolge.

Der Datentyp *großes Objekt* (LOB) unterstützt Multimedia-Objekte wie Dokumente, Video-, Abbild- und Tondaten. Diese Objekte werden mit Hilfe der folgenden Datentypen implementiert:

- Eine Zeichenfolge des Typs *großes Binärobjekt* (BLOB). Beispiele für BLOBs sind Fotografien von Mitarbeitern, Stimmenaufnahmen und Videoaufnahmen.
- Eine Zeichenfolge des Typs *großes Zeichenobjekt* (CLOB), in der eine Folge von Zeichen entweder aus Einzelbyte- oder Mehrbytezeichen oder aus einer Kombination aus beidem bestehen kann. Ein Beispiel für ein CLOB ist ein Lebenslauf eines Mitarbeiters.
- Eine Zeichenfolge des Typs *Doppelbytezeichen für großes Objekt* (DBCLOB), in der eine Folge von Zeichen aus Doppelbytezeichen besteht. Ein Beispiel für ein DBCLOB ist ein Lebenslauf in japanischer Sprache.

Weitere Einzelheiten zur Unterstützung großer Objekte (LOBs) finden Sie im Handbuch *SQL Reference*.

Ein *benutzerdefinierter Datentyp* (UDT - User-Defined Type) ist ein Datentyp, der von einem vorhandenen Typ abgeleitet ist. Es können Datentypen definiert werden, die von vorhandenen Typen abgeleitet sind und über ähnliche Merkmale wie die vorhandenen Typen verfügen, die jedoch als separat und nicht kompatibel behandelt werden sollen.

Ein *strukturierter Typ* ist ein benutzerdefinierter Typ, dessen Struktur in der Datenbank definiert ist. Er enthält eine Reihe benannter *Attribute*, die jeweils über einen Datentyp verfügen. Ein strukturierter Typ kann als

untergeordneter Typ eines anderen strukturierten Typs definiert sein, der dementsprechend als zugehöriger *übergeordneter Typ* bezeichnet wird. Ein untergeordneter Typ übernimmt alle Attribute des dazugehörigen übergeordneten Typs und kann zusätzlich über weitere Attribute verfügen. Die Gruppe strukturierter Typen, die zu einem gemeinsamen übergeordneten Typ gehören, wird als *Typhierarchie* bezeichnet. Ein übergeordneter Typ, der keinem anderen Typ untergeordnet ist, wird als *Stammtyp* der Typhierarchie bezeichnet.

Ein strukturierter Typ kann als Typ einer Tabelle oder einer Sicht verwendet werden. Die Namen und Datentypen der Attribute der strukturierten Typen werden zusammen mit der Objektkennung als Namen und Datentypen für die Spalten dieser *typisierten Tabelle* oder *typisierten Sicht* verwendet. Zeilen einer typisierten Tabelle oder typisierten Sicht können als Darstellung von Exemplaren des strukturierten Typs verstanden werden.

Ein strukturierter Typ kann nicht als Datentyp für eine Spalte einer Tabelle oder Sicht verwendet werden. Außerdem besteht keine Möglichkeit, ein vollständiges Exemplar eines strukturierten Typs in eine Host-Variable eines Anwendungsprogramms zu übernehmen.

Ein *Referenztyp* ist ein Begleittyp zum strukturierten Typ. Wie ein einziger Datentyp ist auch der Referenztyp ein Skalartyp, dessen Darstellung mit der Darstellung eines integrierten Datentyps identisch ist. Diese Darstellung wird von allen Typen der Typhierarchie gemeinsam benutzt. Die Darstellung des Referenztyps wird beim Erstellen des Stammtyps einer Typhierarchie definiert. Bei Verwendung eines Referenztyps wird ein strukturierter Typ als Parameter des Typs angegeben. Dieser Parameter wird als *Zieltyp* der Referenz bezeichnet.

Das Ziel einer Referenz ist immer eine Zeile in einer typisierten Tabelle oder Sicht. Bei Verwendung eines Referenztyps kann ein *Bereich* (Scope) definiert werden. Der Bereich identifiziert eine Tabelle (die *Zieltabelle*) oder eine Sicht (die *Zielsicht*), die die Zielzeile eines Referenzwerts enthält. Die Zieltabelle oder -sicht muß denselben Typ aufweisen wie der Zieltyp des Referenztyps. Ein Exemplar eines Referenztyps mit Bereich identifiziert eindeutig eine Zeile in einer typisierten Tabelle oder Sicht, die als *Zielzeile* dieses Referenztyps bezeichnet wird.

Eine *benutzerdefinierte Funktion* (UDF - User-Defined Function) kann zu zahlreichen Zwecken verwendet werden, darunter auch zum Aufrufen von Routinen, die Vergleiche oder Umwandlungen zwischen benutzerdefinierten Datentypen ermöglichen. Benutzerdefinierte Funktionen erweitern und ergänzen die Unterstützung, die von integrierten SQL-Funktionen zur Verfügung gestellt wird, und können überall dort verwendet werden,

wo auch integrierte Funktionen verwendet werden können. Es gibt zwei Arten benutzerdefinierter Funktionen:

- Externe Funktionen, die in einer Programmiersprache geschrieben sind
- Quellenfunktionen, die zum Aufrufen anderer benutzerdefinierter Funktionen verwendet werden

Beispielsweise gibt es die beiden numerischen Datentypen der europäischen Schuhgröße und der amerikanischen Schuhgröße. Beide Typen stellen Schuhgrößen dar, aber sie sind nicht miteinander kompatibel, da die Maßeinheiten verschieden sind und nicht direkt miteinander verglichen werden können. Eine benutzerdefinierte Funktion kann aufgerufen werden, um die Maßangaben von einer Schuhgröße in eine andere umzuwandeln.

Weitere Einzelheiten zur Vertiefung der benutzerdefinierten und strukturierten Datentypen sowie der Referenztypen und benutzerdefinierten Funktionen finden Sie im Handbuch *SQL Reference*.

3. Geben Sie an, welche Spalten eventuell Standardwerte enthalten müssen.

Einige Spalten können nicht in allen Zeilen sinnvolle Werte enthalten. Dafür kann es folgende Gründe geben:

- Ein Spaltenwert ist für die Zeile nicht zutreffend.
Zum Beispiel kann eine Spalte, die die Mittelinitialen von Mitarbeiternamen enthält, keinen sinnvollen Wert enthalten, wenn der in der Zeile eingetragene Mitarbeiter keine Mittelinitiale hat.
- Ein Wert ist zwar zutreffend, aber er ist noch nicht bekannt.
Ein Beispiel wäre die Spalte MGRNO (Managernummer), die vorübergehend keine gültige Managernummer enthält, weil der vorige Manager versetzt wurde, aber bislang noch kein neuer Manager bestimmt wurde.

In beiden Fällen haben Sie die Wahl, einen Nullwert (einen speziellen Wert, der angibt, daß der Spaltenwert unbekannt oder unzutreffend ist) zuzulassen oder einen vom Nullwert abweichenden Standardwert durch den Datenbankmanager oder die Anwendung zuzuordnen zu lassen.

Einzelheiten zu Nullwerten (NULL Values) und Standardwerten (Default Values) finden Sie im Handbuch *SQL Reference*.

Definieren einer oder mehrerer Spalten als Primärschlüssel

Ein *Schlüssel* ist eine Gruppe von Spalten, deren Hilfe eine Zeile oder Zeilen identifiziert werden oder auf sie zugegriffen wird. Der Schlüssel wird in der Beschreibung einer Tabelle, eines Index oder einer referentiellen Integritätsbedingung angegeben. Eine Spalte kann zu mehreren Schlüsseln gehören.

Ein *eindeutiger Schlüssel* ist ein Schlüssel mit der Integritätsbedingung, daß nicht zwei seiner Werte gleich sein dürfen. Die Spalten eines eindeutigen Schlüssels können keine Nullwerte enthalten. Zum Beispiel kann eine Spalte mit der Personalnummer als eindeutiger Schlüssel definiert werden, da jeder Wert in der Spalte nur einen Mitarbeiter identifiziert. Innerhalb eines Unternehmens können nicht zwei Mitarbeiter dieselbe Personalnummer haben.

Der Mechanismus, mit dem die Eindeutigkeit sichergestellt wird, wird als *eindeutiger Index* bezeichnet. Der eindeutige Index einer Tabelle ist eine Spalte oder eine geordnete Gruppe von Spalten, deren Werte jeweils eine Zeile eindeutig identifizieren (funktionell bestimmen). Ein eindeutiger Index kann Nullwerte enthalten.

Der *Primärschlüssel* ist einer der für eine Tabelle definierten eindeutigen Schlüssel, der jedoch als Schlüssel mit der höchsten Priorität ausgewählt ist. Es kann nur einen Primärschlüssel für eine Tabelle geben.

Für den Primärschlüssel wird automatisch ein *Primärindex* erstellt. Der Primärindex wird vom Datenbankmanager zum effizienten Zugriff auf Tabellenzeilen verwendet und ermöglicht es dem Datenbankmanager, die Eindeutigkeit des Primärschlüssels zu erhalten. (Sie können außerdem Indizes für Spalten definieren, die nicht zum Primärschlüssel gehören, um bei der Verarbeitung von Abfragen einen effizienten Zugriff auf Daten bieten zu können.)

Wenn die Tabelle keinen „natürlichen“ eindeutigen Schlüssel besitzt, oder wenn zur Unterscheidung eindeutiger Zeilen die Eingangsfolge verwendet wird, kann auch die Verwendung einer Zeitmarke als Bestandteil des Schlüssels sinnvoll sein. (Siehe auch „Definieren von IDENTITY-Spalten“ auf Seite 111.)

Primärschlüssel für einige der Beispieltabellen sind folgende:

Tabelle	Schlüsselspalte
Tabelle EMPLOYEE	EMPNO
Tabelle DEPARTMENT	DEPTNO
Tabelle PROJECT	PROJNO

Das folgende Beispiel zeigt einen Teil der Tabelle PROJECT mit ihrem Primärschlüssel.

Tabelle 6. Ein Primärschlüssel in der Tabelle PROJECT

PROJNO (Primärschlüssel)	PROJNAME	DEPTNO
MA2100	Weld Line Automation	D01
MA2110	Weld Line Programming	D11

Wenn jede Spalte einer Tabelle doppelte Werte enthält, kann kein Primärschlüssel aus nur einer Spalte definiert werden. Ein Schlüssel, der aus mehreren Spalten besteht, wird als *zusammengesetzter Schlüssel* bezeichnet. Die Kombination der Spaltenwerte muß eine Entität eindeutig definieren. Wenn ein zusammengesetzter Schlüssel nicht einfach definiert werden kann, ist eventuell die Erstellung einer neuen Spalte mit eindeutigen Werten in Betracht zu ziehen.

Das folgende Beispiel zeigt einen Primärschlüssel, der mehrere Spalten (einen zusammengesetzten Schlüssel) enthält:

Tabelle 7. Ein zusammengesetzter Primärschlüssel in der Tabelle EMP_ACT

EMPNO (Primärschlüssel)	PROJNO (Primärschlüssel)	ACTNO (Primärschlüssel)	EMPTIME	EMSTDATE (Primärschlüssel)
000250	AD3112	60	1.0	1982-01-01
000250	AD3112	60	.5	1982-02-01
000250	AD3112	70	.5	1982-02-01

Identifizieren möglicher Schlüsselspalten

Ein Schlüsselkandidat wird durch die kleinste Anzahl von Spalten festgelegt, die eine Entität eindeutig definieren. Es kann mehrere Schlüsselkandidaten geben. In Tabelle 2 auf Seite 26 gibt es scheinbar mehrere Schlüsselkandidaten. Die Spalten EMPNO, PHONENO und LASTNAME identifizieren den Mitarbeiter eindeutig.

Bei der Auswahl eines Primärschlüssels aus einer Gruppe möglicher Schlüssel sollten die Kriterien Permanenz, Eindeutigkeit und Stabilität beachtet werden.

- Permanenz heißt, daß der Primärschlüssel für die Zeile stets vorhanden ist.
- Eindeutigkeit heißt, daß der Schlüsselwert für eine Zeile sich von den Schlüsselwerten aller anderen Zeilen unterscheidet.
- Stabilität heißt, daß die Werte für Primärschlüssel nie geändert werden.

Von den drei möglichen Schlüsselspalten im genannten Beispiel erfüllt nur die Spalte EMPNO alle Kriterien. Es kann zum Beispiel vorkommen, daß ein Mitarbeiter keine Telefonnummer (PHONENO) hat, wenn er in die Firma eintritt. Nachnamen (LASTNAME) können sich ändern und sind, wenn auch vielleicht zu einem bestimmten Zeitpunkt, so doch nicht unbedingt immer eindeutig. Die Personalnummer (EMPNO) ist die beste Wahl für den Primärschlüssel. Ein Mitarbeiter erhält nur einmal eine eindeutige Nummer, und diese Nummer wird in der Regel nicht geändert, solange der Mitarbeiter in der Firma bleibt. Da jeder Mitarbeiter unbedingt eine Nummer haben muß, gilt auch das Kriterium der Permanenz für die Werte dieser Spalte.

Definieren von IDENTITY-Spalten

Eine *IDENTITY-Spalte* stellt für DB2 eine Methode dar, automatisch einen eindeutigen numerischen Wert für jede Zeile in einer Tabelle zu generieren. Eine Tabelle kann eine einzelne Spalte besitzen, die mit dem Attribut IDENTITY definiert ist. Beispiele für IDENTITY-Spalten sind Auftragsnummern, Personalnummern, Bestandsnummern und Vorfallsnummern.

Werte für eine IDENTITY-Spalte können immer oder standardmäßig generiert werden.

- Die Eindeutigkeit einer IDENTITY-Spalte, die als *immer generiert* (generated always) definiert ist, wird von DB2 garantiert. Die Werte der Spalte werden immer von DB2 generiert. Die Angabe eines expliziten Werts durch Anwendungen wird nicht zugelassen.
- Eine IDENTITY-Spalte, die als *standardmäßig generiert* (generated by default) definiert ist, gibt Anwendungen eine Möglichkeit, einen Wert in die IDENTITY-Spalte explizit einzugeben. Wenn kein Wert angegeben wird, generiert DB2 einen, kann jedoch die Eindeutigkeit des Werts in diesem Fall nicht garantieren. DB2 garantiert die Eindeutigkeit nur für die Gruppe von Werten, die von DB2 generiert wird. Die Definition *standardmäßig generiert* ist zur Verwendung bei Datenweitergaben (Data Propagation), bei der der Inhalt einer vorhandenen Tabelle kopiert wird, oder zum Laden von Inhalt aus einer Tabelle bzw. erneutes Laden in eine Tabelle gedacht.

IDENTITY-Spalten eignen sich ideal zur Generierung eindeutiger Primärschlüsselwerte. Anwendungen können IDENTITY-Spalten nutzen, um Problemen beim gemeinsamen Zugriff und Leistungsproblemen vorzubeugen, die entstehen können, wenn eine Anwendung einen eigenen eindeutigen Zähler außerhalb der Datenbank generiert. Zum Beispiel besteht eine gängige Implementierung auf Anwendungsebene darin, eine 1-zeilige Tabelle mit einem Zähler zu pflegen. Jede Transaktion sperrt diese Tabelle, erhöht den Wert des Zählers und schreibt die Änderung fest. Das heißt, zu einem gegebenen Zeitpunkt kann nur eine Transaktion den Wert des Zählers erhöhen. Wenn im Unterschied dazu der Zähler durch eine IDENTITY-Spalte realisiert wird, können wesentliche höhere Ebenen des gemeinsamen Zugriffs erreicht werden, weil der Zähler nicht durch Transaktionen gesperrt wird. Eine nicht festgeschriebene Transaktion, die den Wert des Zählers erhöht hat, hindert nachfolgende Transaktionen nicht daran, den Zähler ebenfalls zu erhöhen.

Der Wert des Zählers für die IDENTITY-Spalte wird unabhängig von der Transaktion erhöht (oder verringert). Wenn eine bestimmte Transaktion einen IDENTITY-Zähler zweimal erhöht, stellt diese Transaktion möglicherweise einen Abstand zwischen den beiden generierten Nummern fest, weil vielleicht andere Transaktionen zur gleichen Zeit denselben IDENTITY-Zähler ebenfalls erhöhen (d. h. Zeilen in die gleiche Tabelle einfügen). Wenn eine Anwendung unbedingt einen aufeinander folgenden Bereich von Nummern erhalten muß,

sollte die Anwendung eine exklusive Sperre für die Tabelle definieren, in der die IDENTITY-Spalte enthalten ist. Diese Entscheidung muß jedoch gegen die daraus folgende Einbuße an gemeinsamen Zugriff abgewogen werden. Weiterhin ist es möglich, daß eine bestimmte IDENTITY-Spalte so aussehen kann, als habe sie Sprünge in den Nummern generiert, weil eine Transaktion, die einen Wert für die IDENTITY-Spalte generierte, rückgängig gemacht wurde, oder weil die Datenbank, die einen Bereich von Werten in einem Cache gespeichert hatte, deaktiviert wurde, bevor alle im Cache gespeicherten Werte zugeordnet werden konnten.

Die fortlaufenden Nummern, die durch die IDENTITY-Spalte generiert werden, besitzen die folgenden zusätzlichen Merkmale:

- Die Werte können jeden Exact Numeric-Datentyp ohne Kommastellen, d. h. SMALLINT, INTEGER, BIGINT oder DECIMAL mit null Kommastellen, besitzen. (Gleitkommazahlen mit einfacher und doppelter Genauigkeit werden als Approximate Numeric-Datentypen bezeichnet.)
- Fortlaufende Werte können sich durch ein beliebiges angegebenes INTEGER-Inkrement unterscheiden. Das Standardinkrement ist 1.
- Der Wert des Zählers für die IDENTITY-Spalte ist wiederherstellbar. Wenn ein Fehler auftritt, wird der Wert des Zählers aus den Protokollen rekonstruiert, wodurch sichergestellt wird, daß weiterhin eindeutige Werte generiert werden.
- Werte von IDENTITY-Spalten können im Cache gespeichert werden, um eine bessere Leistung zu erzielen.

Sicherstellen, daß gleiche Werte die gleiche Entität darstellen

Sie können die Attribute derselben Menge von Entitäten in mehreren Tabellen beschreiben. Zum Beispiel enthält die Tabelle EMPLOYEE die Nummer der Abteilung (WORKDEPT), der ein Mitarbeiter zugeteilt ist, und die Tabelle DEPARTMENT zeigt, welcher Manager jeder Abteilungsnummer (DEPTNO) zugeordnet ist. Um beide Mengen von Attributen gleichzeitig abzurufen, können die beiden Tabellen über die beiden übereinstimmenden Spalten verknüpft werden, wie im folgenden Beispiel gezeigt wird. Der Wert der Spalte WORKDEPT und der Wert der Spalte DEPTNO stellen dieselbe Entität dar und bilden daher einen *Verknüpfungspfad* zwischen den Tabellen DEPARTMENT und EMPLOYEE.

Die Tabelle DEPARTMENT:

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT
D21	Administration Support	000070	D01

Die Tabelle EMPLOYEE:

EMPNO	FIRSTNAME	LASTNAME	WORKDEPT	JOB
000250	Daniel	Smith	D21	Clerk

Beim Abrufen von Informationen über eine Entität aus mehreren Tabellen muß sichergestellt werden, daß gleiche Werte auch die gleiche Entität bezeichnen. Die verknüpfenden Spalten können unterschiedliche Namen haben (wie WORKDEPT und DEPTNO im vorangegangenen Beispiel), oder sie können auch denselben Namen haben (wie die Spalten mit dem Namen DEPTNO in den Tabellen DEPARTMENT und PROJECT).

Normalisieren der Tabellen

Eine *Normalisierung* hilft bei der Vermeidung von Redundanzen und Inkonsistenzen in Tabellendaten. Sie bezeichnet den Prozeß der Reduzierung von Tabellen auf eine Menge von Spalten, in der alle Spalten, die selbst zu keinem Schlüssel gehören, von der Primärschlüsselspalte abhängig sind. Wenn dies nicht der Fall ist, können die Daten durch Aktualisierungen inkonsistent werden.

In diesem Abschnitt werden kurz die Kriterien für die erste, zweite, dritte und vierte Normalform vorgestellt. Die fünfte Normalform einer Tabelle, die in zahlreichen Veröffentlichungen zum Entwurf von Datenbanken dargestellt wird, ist hier nicht beschrieben.

Form Beschreibung

- Erste* An jeder Zeilen- und Spaltenposition in der Tabelle existiert nur *ein* Wert und zu keiner Zeit eine Gruppe von Werten (siehe „Erste Normalform“).
- Zweite* Jede Spalte, die nicht Teil des Schlüssels ist, ist vom Schlüssel abhängig (siehe „Zweite Normalform“ auf Seite 114).
- Dritte* Jede Spalte ohne Schlüsselfunktion ist von anderen Spalten ohne Schlüsselfunktion unabhängig und nur vom Schlüssel abhängig (siehe „Dritte Normalform“ auf Seite 116).
- Vierte* Keine Zeile enthält zwei oder mehr unabhängige, mehrwertige Fakten über eine Entität (siehe „Vierte Normalform“ auf Seite 117).

Erste Normalform

Eine Tabelle ist in der *ersten Normalform*, wenn sich in jeder Zelle nur ein Wert und nie eine Gruppe von Werten befindet. Eine Tabelle in der ersten Normalform erfüllt nicht unbedingt auch die Kriterien für höhere Normalformen.

Die folgende Tabelle verstößt beispielsweise gegen die Kriterien der ersten Normalform, weil in der Spalte LAGER mehrere Werte für jedes Vorkommen von ARTIKEL auftreten.

Tabelle 8. Tabelle, die die erste Normalform verletzt

ARTIKEL (Primärschlüssel)	LAGER
P0010	Lager A, Lager B, Lager C
P0020	Lager B, Lager D

Das folgende Beispiel zeigt die gleiche Tabelle in der ersten Normalform.

Tabelle 9. Tabelle in der ersten Normalform

ARTIKEL (Primärschlüssel)	LAGER (Primärschlüssel)	BESTAND
P0010	Lager A	400
P0010	Lager B	543
P0010	Lager C	329
P0020	Lager B	200
P0020	Lager D	278

Zweite Normalform

Eine Tabelle ist in der *zweiten Normalform*, wenn jede Spalte, die nicht zum Schlüssel gehört, vom *gesamten* Schlüssel abhängig ist.

Die Kriterien der zweiten Normalform sind nicht erfüllt, wenn eine Spalte ohne Schlüsselfunktion von einem *Teil* eines zusammengesetzten Schlüssels abhängig ist, wie im folgenden Beispiel zu sehen ist:

Tabelle 10. Tabelle, die die zweite Normalform verletzt

ARTIKEL (Primärschlüssel)	LAGER (Primärschlüssel)	BESTAND	LAGER_ADRESSE
P0010	Lager A	400	Amselfelderstraße 24
P0010	Lager B	543	Industrieweg 6
P0010	Lager C	329	Kastanienallee 71
P0020	Lager B	200	Industrieweg 6
P0020	Lager D	278	Ulmenstraße 8

Der Primärschlüssel ist ein zusammengesetzter Schlüssel, der aus den Spalten ARTIKEL und LAGER besteht. Da die Spalte LAGER_ADRESSE jedoch nur vom Wert der Spalte LAGER abhängig ist, verstößt die Tabelle gegen die Regel der zweiten Normalform.

Dieser Tabellenentwurf hat folgende Nachteile:

- Die Lageradresse wird in jeden Datensatz für einen Artikel, der in dem entsprechenden Lagerhaus gelagert wird, wiederholt.
- Wenn die Adresse eines Lagerhauses geändert wird, muß jede Zeile, die sich auf einen Artikel in diesem Lagerhaus bezieht, aktualisiert werden.
- Aufgrund dieser Redundanz könnten die Daten inkonsistent werden, wenn für dasselbe Lagerhaus in verschiedenen Datensätzen unterschiedliche Adressen auftreten.
- Wenn zu einem Zeitpunkt keine Artikel in dem Lagerhaus gelagert werden, ist vielleicht keine Zeile mehr vorhanden, in der die Adresse des Lagerhauses gespeichert werden kann.

Die Lösung besteht in einer Aufspaltung der Tabelle in die beiden folgenden Tabellen:

Tabelle 11. Die Tabelle LAGERBESTAND erfüllt die zweite Normalform

ARTIKEL (Primärschlüssel)	LAGER (Primärschlüssel)	BESTAND
P0010	Lager A	400
P0010	Lager B	543
P0010	Lager C	329
P0020	Lager B	200
P0020	Lager D	278

Tabelle 12. Die Tabelle LAGERHAUS erfüllt die zweite Normalform

LAGER (Primärschlüssel)	LAGER_ADRESSE
Lager A	Amselfelderstraße 24
Lager B	Industrieweg 6
Lager C	Kastanienallee 71
Lager D	Ulmenstraße 8

Es muß beachtet werden, daß die Erstellung zweier Tabellen in der zweiten Normalform Auswirkungen auf die Leistung haben kann. Anwendungen, die Berichte über den Lagerort von Artikeln erstellen, müssen beide Tabellen verknüpfen, um die relevanten Informationen abrufen zu können.

Weitere Einzelheiten zur Datenbankleistung finden Sie im Abschnitt zum Optimieren der Anwendungsleistung im Handbuch *Systemverwaltung: Optimierung*.

Dritte Normalform

Eine Tabelle ist in der dritten Normalform, wenn jede Spalte ohne Schlüssel-funktion von anderen Spalten ohne Schlüsselfunktion unabhängig und nur vom Schlüssel abhängig ist.

Die erste Tabelle des folgenden Beispiels enthält die Spalten EMPNO und WORKDEPT. Nehmen Sie an, daß die Spalte DEPTNAME hinzugefügt wird (siehe Tabelle 14). Die neue Spalte ist von der Spalte WORKDEPT abhängig, während der Primärschlüssel nur die Spalte EMPNO umfaßt. Die Tabelle verstößt nun gegen die Kriterien der dritten Normalform. Durch Ändern der Spalte DEPTNAME für einen einzelnen Mitarbeiter, z. B. John Parker, wird nicht auch der Abteilungsname für andere Mitarbeiter dieser Abteilung geändert. Beachten Sie, daß es nun zwei unterschiedliche Abteilungsnamen (DEPTNAME) für Abteilungsnummer (WORKDEPT) E11 gibt. Die sich daraus ergebende Inkonsistenz zeigt sich in der aktualisierten Version der Tabelle.

Tabelle 13. Nicht normalisierte Tabelle EMPLOYEE_DEPARTMENT vor der Aktualisierung

EMPNO (Primär-schlüssel)	FIRSTNAME	LASTNAME	WORKDEPT	DEPTNAME
000290	John	Parker	E11	Operations
000320	Ramlal	Mehta	E21	Software Support
000310	Maude	Setright	E11	Operations

Tabelle 14. Nicht normalisierte Tabelle EMPLOYEE_DEPARTMENT nach der Aktualisierung. Die Daten der Tabelle sind nicht mehr konsistent.

EMPNO (Primär-schlüssel)	FIRSTNAME	LASTNAME	WORKDEPT	DEPTNAME
000290	John	Parker	E11	Installation Mgmt
000320	Ramlal	Mehta	E21	Software Support
000310	Maude	Setright	E11	Operations

Die Tabelle kann normalisiert werden, indem eine neue Tabelle mit Spalten für WORKDEPT und DEPTNAME erstellt wird. Eine Aktualisierung wie das

Ändern des Abteilungsnamens (DEPTNAME) ist nun wesentlich einfacher. Es muß nur eine Tabelle aktualisiert werden.

Eine SQL-Abfrage, mit der der Abteilungsname zusammen mit dem Namen des Mitarbeiters abgerufen wird, ist dagegen etwas komplizierter, da die beiden Tabellen verknüpft werden müssen. Sie wird wahrscheinlich eine etwas längere Zeit zur Ausführung benötigen als eine Abfrage auf eine einzelne Tabelle. Ferner ist zusätzlicher Speicherplatz erforderlich, da die Spalte WORKDEPT (Abteilung) in beiden Tabellen enthalten sein muß.

Die folgenden Tabellen werden als Ergebnis dieser Normalisierung definiert:

Tabelle 15. Tabelle EMPLOYEE nach Normalisierung der Tabelle EMPLOYEE_DEPARTMENT

EMPNO (Primärschlüssel)	FIRSTNAME	LASTNAME	WORKDEPT
000290	John	Parker	E11
000320	Ramlal	Mehta	E21
000310	Maude	Setright	E11

Tabelle 16. Tabelle DEPARTMENT nach Normalisierung der Tabelle EMPLOYEE_DEPARTMENT

DEPTNO (Primärschlüssel)	DEPTNAME
E11	Operations
E21	Software Support

Vierte Normalform

Eine Tabelle ist in der vierten Normalform, wenn keine Zeile zwei oder mehr unabhängige, mehrwertige Fakten bezüglich einer Entität enthält.

Betrachten Sie zum Beispiel folgende Entitäten: Mitarbeiter (Employee), Fachkenntnisse (Skill) und Sprachen (Language). Ein Mitarbeiter kann über Fachkenntnisse auf mehreren Gebieten verfügen und mehrere Sprachen beherrschen. Es gibt zwei Beziehungen, eine zwischen Mitarbeitern und Fachkenntnissen und eine zwischen Mitarbeitern und Sprachen. Eine Tabelle verstößt gegen die Kriterien der vierten Normalform, wenn sie beide Beziehungen darstellt, wie im folgenden Beispiel gezeigt:

Tabelle 17. Tabelle, die die vierte Normalform verletzt

EMPNO (Primärschlüssel)	SKILL (Primärschlüssel)	LANGUAGE (Primärschlüssel)
000130	Data Modelling	English

Tabelle 17. Tabelle, die die vierte Normalform verletzt (Forts.)

EMPNO (Primärschlüssel)	SKILL (Primärschlüssel)	LANGUAGE (Primärschlüssel)
000130	Database Design	English
000130	Application Design	English
000130	Data Modelling	Spanish
000130	Database Design	Spanish
000130	Application Design	Spanish

Statt dessen sollten die beiden Beziehungen in zwei Tabellen dargestellt werden:

Tabelle 18. Tabelle EMPLOYEE_SKILL in der vierten Normalform

EMPNO (Primärschlüssel)	SKILL (Primärschlüssel)
000130	Data Modelling
000130	Database Design
000130	Application Design

Tabelle 19. Tabelle EMPLOYEE_LANGUAGE in der vierten Normalform

EMPNO (Primärschlüssel)	LANGUAGE (Primärschlüssel)
000130	English
000130	Spanish

Wenn jedoch die Attribute gegenseitige Abhängigkeiten zeigen, (d. h., der Mitarbeiter verwendet bestimmte Sprachen nur im Zusammenhang mit bestimmten Fachgebieten), dann sollte die Tabelle *nicht* geteilt werden.

Eine gute Strategie bei der Entwicklung eines Datenbankentwurfs besteht darin, alle Daten in Tabellen in der vierten Normalform anzuordnen und anschließend zu entscheiden, ob das Ergebnis eine akzeptable Leistung liefert. Ist dies nicht der Fall, können Sie die Daten in Tabellen in der dritten Normalform neu anordnen und die Leistung erneut begutachten.

Planen der Integritätsbedingungen

Eine *Integritätsbedingung* ist eine Regel, die vom Datenbankmanager angewendet wird. Vier Arten der Behandlung von Integritätsbedingungen werden in diesem Abschnitt beschrieben:

Eindeutige Integritätsbedingungen

Stellen die Eindeutigkeit der Schlüsselwerte in einer Tabelle sicher. Alle Änderungen an den Spalten, die den Primärschlüssel bilden, werden auf Eindeutigkeit überprüft.

Referentielle Integrität

Implementiert referentielle Integritätsbedingungen für die Operationen INSERT (Einfügen), UPDATE (Aktualisieren) und DELETE (Löschen). Dies ist der Status einer Datenbank, in der alle Werte aller Fremdschlüssel gültig sind.

Prüfungen auf Integritätsbedingungen in Tabellen

Stellen sicher, daß keine geänderten Daten gegen Bedingungen verstoßen, die bei der Erstellung oder der Änderung der Tabelle angegeben wurden.

Auslöser

Definieren eine Reihe von Aktionen, die ausgeführt werden, wenn sie durch eine Aktualisierungs-, Lösch- oder Einfügeoperation für eine angegebene Tabelle aufgerufen werden.

Eindeutige Integritätsbedingungen

Eine *eindeutige Integritätsbedingung* ist die Regel, die sicherstellt, daß Schlüsselwerte innerhalb der Tabelle eindeutig sind. Jede Spalte, die Bestandteil des Schlüssels in einer eindeutigen Integritätsbedingung ist, muß mit NOT NULL (keine Nullwerte möglich) definiert werden. Eindeutige Integritätsbedingungen werden in der Anweisung CREATE TABLE oder ALTER TABLE mit Hilfe der Klausel PRIMARY KEY oder UNIQUE definiert.

Eine Tabelle kann eine beliebige Anzahl eindeutiger Integritätsbedingungen enthalten. Jedoch kann nur eine eindeutige Spaltenfolge als Primärschlüssel für eine Tabelle definiert werden. Außerdem kann eine Tabelle nicht mehr als eine eindeutige Integritätsbedingung für dieselbe Gruppe von Spalten besitzen.

Wenn eine eindeutige Integritätsbedingung definiert ist, erstellt der Datenbankmanager einen eindeutigen Index (falls erforderlich) und markiert ihn als primären oder eindeutigen systembedingten Index. Die Einhaltung der Integritätsbedingung wird mit Hilfe des eindeutigen Index gewährleistet. Wenn eine eindeutige Integritätsbedingung für eine Spalte definiert ist, wird die Überprüfung auf Eindeutigkeit bei Aktualisierung mehrerer Zeilen an das Ende der Aktualisierungsoperation verlegt.

Die Spalten, die einer eindeutigen Integritätsbedingung unterliegen, können auch als Primärschlüssel in einer referentiellen Integritätsbedingung verwendet werden.

Referentielle Integrität

Der Datenbankmanager sorgt für die Erhaltung der referentiellen Integrität mit Hilfe *referentieller Integritätsbedingungen*, die festlegen, daß alle Werte eines bestimmten Attributs oder einer Tabellenspalte auch in einer anderen Tabelle oder Spalte vorhanden sind. Eine referentielle Integritätsbedingung kann zum Beispiel fordern, daß jeder Mitarbeiter in der Tabelle EMPLOYEE in einer Abteilung sein muß, die in der Tabelle DEPARTMENT aufgeführt ist. Kein Mitarbeiter kann in einer Abteilung sein, die nicht existiert.

Referentielle Integritätsbedingungen können in einer Datenbank implementiert werden, um zu gewährleisten, daß die referentielle Integrität erhalten bleibt, und um dem Optimierungsprogramm zu ermöglichen, durch Ausnutzung dieser besonderen Abhängigkeitsbeziehungen Abfragen effizienter zu verarbeiten. Bei der Planung im Hinblick auf referentielle Integrität sollten Sie alle Beziehungen zwischen Datenbanktabellen angeben. Sie können eine Abhängigkeitsbeziehung durch Definition eines Primärschlüssels und referentieller Integritätsbedingungen festlegen.

Betrachten Sie die folgenden, durch eine Beziehung verbundenen Tabellen:

Tabelle 20. Tabelle DEPARTMENT

DEPTNO (Primärschlüssel)	DEPTNAME	MGRNO
A00	Spiffy Computer Service Div.	000010
B01	Planning	000020
C01	Information Center	000030
D11	Manufacturing Systems	000060

Tabelle 21. Tabelle EMPLOYEE

EMPNO (Primärschlüssel)	FIRSTNAME	LASTNAME	WORKDEPT (Fremdschlüssel)	PHONENO
000010	Christine	Haas	A00	3978
000030	Sally	Kwan	C01	4738
000060	Irving	Stern	D11	6423
000120	Sean	O'Connell	A00	2167
000140	Heather	Nicholls	C01	1793

Tabelle 21. Tabelle EMPLOYEE (Forts.)

EMPNO (Primär- schlüssel)	FIRSTNAME	LASTNAME	WORKDEPT (Fremd- schlüssel)	PHONENO
000170	Masatoshi	Yoshimura	D11	2890

Viele der folgenden Konzepte, die zum Verständnis der referentiellen Integrität hilfreich sind, werden mit Bezug auf diese Tabellen diskutiert.

Ein *eindeutiger Schlüssel* ist eine Spalte oder Spaltengruppe, in denen Werte einer Zeile in keiner anderen Zeile nochmals vorkommen. Für die Tabelle kann ein eindeutiger Schlüssel als Primärschlüssel definiert werden. Ein eindeutiger Schlüssel kann auch als *übergeordneter Schlüssel* bezeichnet werden, wenn sich ein Fremdschlüssel auf ihn bezieht.

Ein *Primärschlüssel* ist ein eindeutiger Schlüssel, der Teil der Definition der Tabelle ist. Jede Tabelle kann einen und nur einen Primärschlüssel besitzen. In den vorangegangenen Beispielen sind die Spalten DEPTNO und EMPNO Primärschlüssel der Tabellen DEPARTMENT und EMPLOYEE.

Ein *Fremdschlüssel* ist eine Spalte oder eine Spaltengruppe in einer Tabelle, die sich auf einen eindeutigen Schlüssel oder den Primärschlüssel derselben oder einer anderen Tabelle bezieht. Ein Fremdschlüssel wird zur Herstellung einer Abhängigkeitsbeziehung mit einem eindeutigen Schlüssel oder dem Primärschlüssel verwendet, um die referentielle Integrität zwischen Tabellen zu gewährleisten. Die Spalte WORKDEPT in der Tabelle EMPLOYEE ist ein Fremdschlüssel, weil sie auf den Primärschlüssel, die Spalte DEPTNO, in der Tabelle DEPARTMENT bezogen ist.

Ein *zusammengesetzter Schlüssel* ist ein Schlüssel, der aus mehr als einer Spalte besteht. Sowohl Primär- als auch Fremdschlüssel können zusammengesetzte Schlüssel sein. Wenn zum Beispiel Abteilungen eindeutig durch eine Kombination aus Unternehmensbereichsnummer und Abteilungsnummer identifiziert würden, wären zwei Spalten erforderlich, um den Schlüssel für die Tabelle DEPARTMENT zu bilden.

Ein *übergeordneter Schlüssel* ist ein Primärschlüssel oder eindeutiger Schlüssel einer referentiellen Integritätsbedingung. Als *Integritätsbedingung über Primärschlüssel* wird der übergeordnete Standardschlüssel der referentiellen Integritätsbedingung festgelegt, wenn keine Gruppe übergeordneter Schlüsselspalten angegeben wird.

Eine *übergeordnete Tabelle* ist eine Tabelle mit einem übergeordneten Schlüssel, auf den mindestens ein Fremdschlüssel in derselben oder einer anderen

Tabelle verweist. Eine Tabelle kann in beliebig vielen Beziehungen eine übergeordnete Tabelle sein. Die Tabelle DEPARTMENT, die einen Primärschlüssel in der Spalte DEPTNO hat, ist zum Beispiel eine übergeordnete Tabelle zur Tabelle EMPLOYEE, die den Fremdschlüssel WORKDEPT enthält.

Eine *übergeordnete Zeile* ist eine Zeile einer übergeordneten Tabelle, deren Wert des übergeordneten Schlüssels mit mindestens einem Wert eines Fremdschlüssels in einer abhängigen Tabelle übereinstimmt. Eine Zeile in einer übergeordneten Tabelle ist nicht unbedingt eine übergeordnete Zeile. Die vierte Zeile (D11) der Tabelle DEPARTMENT ist die übergeordnete Zeile der dritten und der sechsten Zeile in der Tabelle EMPLOYEE. Die zweite Zeile (B01) der Tabelle DEPARTMENT ist jedoch keine übergeordnete Zeile zu irgendeiner anderen Zeile.

Eine *abhängige Tabelle* ist eine Tabelle, die einen oder mehrere Fremdschlüssel enthält. Eine abhängige Tabelle kann gleichzeitig auch eine übergeordnete Tabelle sein. Eine Tabelle kann in einer beliebigen Anzahl von Beziehungen eine abhängige Tabelle sein. Die Tabelle EMPLOYEE enthält den Fremdschlüssel WORKDEPT und ist von der Tabelle DEPARTMENT abhängig, deren Primärschlüssel die Spalte DEPTNO ist.

Eine *abhängige Zeile* ist eine Zeile einer abhängigen Tabelle, die einen Nicht-nullwert im Fremdschlüssel enthält, der mit dem Wert eines übergeordneten Schlüssels übereinstimmt. Der Wert des Fremdschlüssels stellt einen Verweis der abhängigen Zeile auf die übergeordnete Zeile dar. Da Fremdschlüssel Nullwerte akzeptieren kann, ist eine Zeile einer abhängigen Tabelle nicht unbedingt auch eine abhängige Zeile.

Eine Tabelle ist eine *untergeordnete Tabelle* zu einer anderen Tabelle, wenn sie selbst eine abhängige Tabelle oder eine untergeordnete Tabelle zu einer abhängigen Tabelle ist. Eine untergeordnete Tabelle enthält einen Fremdschlüssel, der bis zu einem übergeordneten Schlüssel einer Tabelle zurückverfolgt werden kann.

Ein *Referenzzyklus* ist ein Pfad von Abhängigkeitsbeziehungen, durch den eine Tabelle mit sich selbst verknüpft wird. Wenn eine Tabelle direkt mit sich selbst verknüpft ist, handelt es sich um eine *auf sich selbst verweisende* Tabelle. Wenn die Tabelle EMPLOYEE eine weitere Spalte mit dem Namen MGRID enthielte, in der die EMPNO (Personalnummern) für jeden Manager eines Mitarbeiters aufgeführt wären, wäre die Tabelle EMPLOYEE eine auf sich selbst verweisende Tabelle. MGRID wäre ein Fremdschlüssel für die Tabelle EMPLOYEE.

Eine auf sich selbst verweisende Tabelle ist innerhalb derselben Beziehung sowohl übergeordnete als auch abhängige Tabelle. Eine auf sich selbst verweisende Zeile ist eine Zeile, die sowohl eine übergeordnete als auch eine abhängige Zeile zu sich selbst ist. Die Integritätsbedingung, die in diesem Fall vor-

handen ist, wird als auf sich selbst verweisende Integritätsbedingung bezeichnet. Wenn beispielsweise der Wert für den Fremdschlüssel in einer Zeile einer auf sich selbst verweisenden Tabelle mit dem Wert des eindeutigen Schlüssels in dieser Zeile übereinstimmt, dann verweist die Zeile auf sich selbst.

Eine *referentielle Integritätsbedingung* ist eine Festlegung, daß Nichtnullwerte eines bestimmten Fremdschlüssels nur gültig sind, wenn sie auch als Wert für einen eindeutigen Schlüssel einer bestimmten Tabelle vorkommen. Der Zweck referentieller Integritätsbedingungen ist die Gewährleistung, daß Datenbankbeziehungen erhalten bleiben und Dateneingaberegeln befolgt werden.

Auswirkungen auf SQL-Operationen

Die Implementierung referentieller Integritätsbedingungen hat besondere Auswirkungen für einige SQL-Operationen, bei denen unterschieden wird, ob es sich um übergeordnete oder abhängige Tabellen handelt. In diesem Abschnitt werden die Auswirkungen einer Wahrung der referentiellen Integrität auf SQL-Operationen mit den Anweisungen INSERT, DELETE, UPDATE und DROP behandelt.

DB2 sorgt *nicht* automatisch für die systemübergreifende Erhaltung referentieller Integritätsbedingungen. Wenn also referentielle Integritätsbedingungen systemübergreifend realisiert werden sollen, müssen die Anwendungen die erforderliche Logik enthalten.

Die folgenden Themen werden behandelt:

- „Regeln zur Anweisung INSERT“
- „Regeln zur Anweisung DELETE“ auf Seite 124
- „Regeln zur Anweisung UPDATE“ auf Seite 125

Regeln zur Anweisung INSERT: Sie können jederzeit eine Zeile in einer übergeordneten Tabelle einfügen, ohne daß eine Aktion in abhängigen Tabellen durchgeführt wird. In einer abhängigen Tabelle hingegen können Sie keine Zeile einfügen, sofern es keine Zeile in der übergeordneten Tabelle mit einem übergeordneten Schlüsselwert gibt, der mit dem Fremdschlüsselwert der einzufügenden Zeile übereinstimmt, wenn dieser Fremdschlüsselwert nicht NULL ist. Der Wert eines zusammengesetzten Fremdschlüssels ist NULL, wenn irgendeine Komponente des Werts NULL ist.

Diese Regel gilt implizit, wenn ein Fremdschlüssel definiert wird.

Bei dem Versuch, eine Zeile in einer Tabelle mit referentiellen Integritätsbedingungen einzufügen, ist die Operation INSERT nicht zulässig, wenn einer der Nichtnullwerte des Fremdschlüssels nicht im übergeordneten Schlüssel der

übergeordneten Tabelle vorkommt. Wenn die Operation INSERT bei dem Versuch, mehrere Zeilen einzufügen, aufgrund einer Zeile fehlschlägt, wird keine der Zeilen eingefügt.

Regeln zur Anweisung DELETE: Wenn eine Zeile aus einer übergeordneten Tabelle gelöscht wird, prüft DB2, ob es abhängige Zeilen in abhängigen Tabellen mit übereinstimmenden Fremdschlüsselwerten gibt. Werden abhängige Zeilen gefunden, sind mehrere Aktionen möglich. Sie können bestimmen, welche Aktion ausgeführt wird, indem Sie bei der Erstellung der abhängigen Tabelle eine *Löschbedingung* angeben.

Die Löschbedingungen, die für eine abhängige Tabelle (die Tabelle, die den Fremdschlüssel enthält) für den Fall, daß ein Primärschlüssel gelöscht wird, angegeben werden können, sind folgende:

RESTRICT

Diese Löschbedingung verhindert, daß eine Zeile in der übergeordneten Tabelle gelöscht wird, wenn eine oder mehrere abhängige Zeilen gefunden werden. Wenn sowohl die übergeordneten als auch die abhängigen Zeilen gelöscht werden sollen, müssen die abhängigen Zeilen zuerst gelöscht werden. Löschen zuerst der übergeordneten Zeilen verletzt die referentielle Integritätsbedingung und ist nicht zulässig.

NO ACTION

Sichert das Vorhandensein einer übergeordneten Zeile für jede untergeordnete Zeile, nachdem alle referentiellen Integritätsbedingungen angewendet wurden.

CASCADE

Diese Löschbedingung legt fest, daß durch Löschen einer Zeile der übergeordneten Tabelle automatisch auch alle abhängigen Zeilen in der abhängigen Tabelle gelöscht werden. Diese Löschbedingung ist sinnvoll, wenn eine Zeile in der abhängigen Tabelle ohne die Zeile in der übergeordneten Tabelle keine Bedeutung mehr hätte.

Das Löschen der übergeordneten Zeile bewirkt zunächst, daß die abhängigen Zeilen, die auf einen Primärschlüssel verweisen, automatisch gelöscht werden. Daher ist es nicht nötig, die abhängigen Zeilen zuerst zu löschen. Wenn einige dieser abhängigen Zeilen selbst übergeordnete Zeilen zu abhängigen Zeilen sind, wird die Löschbedingung auf diese

Abhängigkeitsbeziehungen ebenfalls angewendet. DB2 verwaltet gestaffelte Löschoptionen.

SET NULL

Diese Löschoption bewirkt, daß durch das Löschen einer Zeile der übergeordneten Tabelle die Werte des Fremdschlüssels in allen abhängigen Zeilen auf NULL gesetzt werden. Die anderen Spalten der Zeilen bleiben unverändert.

Wenn bei der Erstellung keine Löschoption explizit definiert wird, wird die Löschoption NO ACTION verwendet.

Jede Tabelle, die in eine Löschoption mit einbezogen werden kann, wird als durch übergreifendes Löschen verknüpft bezeichnet. Die folgenden Einschränkungen gelten für Verknüpfungen durch übergreifendes Löschen.

- Eine Tabelle kann nicht durch übergreifendes Löschen mit sich selbst in einem Referenzzyklus mit mehr als einer Tabelle verknüpft sein.
- Wenn eine Tabelle durch übergreifendes Löschen mit einer anderen Tabelle über mehr als eine Abhängigkeitsbeziehung verknüpft ist, müssen für diese Abhängigkeitsbeziehungen dieselben Löschoptionen, nämlich entweder CASCADE oder NO ACTION, gelten.
- Wenn eine auf sich selbst verweisende Tabelle von einer anderen Tabelle in einer CASCADE-Beziehung abhängig ist, muß für die auf sich selbst verweisende Beziehung auch die Löschoption CASCADE gelten.

In einer abhängigen Tabelle können zu jeder Zeit Zeilen gelöscht werden, ohne daß eine Aktion in der übergeordneten Tabelle erforderlich wird. Zum Beispiel könnte in der Abteilung-Mitarbeiter-Beziehung (DEPARTMENT-EMPLOYEE) ein Mitarbeiter in den Ruhestand gehen, so daß seine Zeile aus der Tabelle EMPLOYEE ohne Auswirkung auf die Tabelle DEPARTMENT gelöscht würde. (In der umgekehrten Mitarbeiter-Abteilung-Beziehung (EMPLOYEE-DEPARTMENT) ist die Abteilungsmanager-ID ein Fremdschlüssel mit Verweis auf den übergeordneten Schlüssel der Tabelle EMPLOYEE ist. Wenn ein Manager in den Ruhestand tritt, *hat* dies natürlich eine Auswirkung auf die Tabelle DEPARTMENT.)

Regeln zur Anweisung UPDATE: DB2 läßt nicht zu, daß ein eindeutiger Schlüssel einer übergeordneten Zeile aktualisiert wird. Wenn ein Fremdschlüssel in einer abhängigen Tabelle aktualisiert wird und der Fremdschlüssel nicht NULL ist, muß er mit einem Wert des übergeordneten Schlüssels der übergeordneten Tabelle der Abhängigkeitsbeziehung übereinstimmen. Wenn eine referentielle Integritätsbedingung durch eine Operation UPDATE verletzt wird, tritt ein Fehler auf, und keine Zeile wird aktualisiert.

Für die Aktualisierung eines Werts in einer Spalte des übergeordneten Schlüssels gilt folgendes:

- Wenn irgendeine Zeile in der abhängigen Tabelle mit dem Ursprungswert des Schlüssels übereinstimmt, wird die UPDATE-Operation zurückgewiesen, wenn als Aktualisierungsregel RESTRICT definiert ist.
- Wenn irgendeine Zeile der abhängigen Tabelle nach Ausführung der UPDATE-Anweisung keinen entsprechenden übergeordneten Schlüssel hat (außer nach Auslösern), wird die Aktualisierung zurückgewiesen, wenn als Aktualisierungsregel NO ACTION definiert ist.

Zur Aktualisierung des Werts eines übergeordneten Schlüssels, der sich in einer übergeordneten Zeile befindet, müssen Sie zunächst die Beziehung zu allen abhängigen Zeilen in den abhängigen Tabellen durch eine der folgenden Operationen beseitigen:

- Löschen der abhängigen Zeilen
- Aktualisieren der Fremdschlüssel in den abhängigen Tabellen, um sie mit anderen gültigen Schlüsselwerten zu versehen

Wenn es keine Abhängigkeit zum Schlüsselwert in der Zeile mehr gibt, ist die Zeile keine übergeordnete Zeile einer referentiellen Abhängigkeitsbeziehung mehr und kann aktualisiert werden.

Wenn ein Teil eines Fremdschlüssels aktualisiert wird und kein Teil des Fremdschlüssels einen Nullwert (NULL) enthält, muß der neue Wert des Fremdschlüssels als eindeutiger Schlüsselwert in der übergeordneten Tabelle vorhanden sein. Wenn kein Fremdschlüssel von einem bestimmten eindeutigen Schlüssel abhängig ist, d. h., wenn die Zeile mit dem eindeutigen Schlüssel *keine* übergeordnete Zeile ist, kann ein Teil des eindeutigen Schlüssels aktualisiert werden. Jedoch kann in diesem Fall nicht mehr als eine Zeile gleichzeitig zur Aktualisierung ausgewählt werden, da es sich um einen eindeutigen Schlüssel handelt und mehrere gleiche Werte in verschiedenen Zeilen nicht zulässig sind.

Prüfungen auf Integritätsbedingungen in Tabellen

Geschäftsregeln, die in Ihrem Datenbankentwurf identifiziert wurden, können mit Hilfe von Prüfungen auf Integritätsbedingungen in Tabellen (Table Check Constraints) implementiert werden. *Prüfungen auf Integritätsbedingungen für Tabellen* geben Suchbedingungen an, die für jede Zeile einer Tabelle angewendet werden. Diese Integritätsbedingungen werden automatisch aktiviert, wenn eine Anweisung UPDATE oder INSERT für eine Tabelle ausgeführt wird. Sie werden durch die Anweisung CREATE TABLE oder ALTER TABLE definiert.

Eine *Prüfung auf Integritätsbedingung* kann zur Gültigkeitsprüfung der Daten verwendet werden. Zum Beispiel müssen Werte für eine Abteilungsnummer innerhalb des Bereichs von 10 bis 100 liegen, oder die Jobbezeichnung für

einen Mitarbeiter (Employee) kann nur "Verkauf" (Sales), "Manager" oder "Sachbearbeiter" (Clerk) lauten, oder ein Mitarbeiter, der länger als acht Jahre in der Firma ist, muß ein Gehalt über 45.500 Dollar beziehen.

Informationen zu den Auswirkungen von Prüfungen auf Integritätsbedingungen in Tabellen auf die Befehle IMPORT und LOAD finden Sie im Handbuch *Versetzen von Daten Dienstprogramme und Referenz*.

Auslöser

Ein *Auslöser* (Trigger) ist eine definierte Gruppe von Aktionen, die immer dann ausgeführt werden, wenn eine DELETE-, INSERT- oder UPDATE-Operation für eine bestimmte Tabelle ausgeführt wird. Auslöser können zur Unterstützung von Geschäftsregeln definiert werden. Auslöser können außerdem zur automatischen Aktualisierung von Übersichts- oder Prüfdaten verwendet werden. Da ein Auslöser in der Datenbank gespeichert wird, müssen die durch den Auslöser ausgeführten Aktionen nicht mehr in jedem Anwendungsprogramm codiert werden. Der Auslöser wird einmal codiert, in der Datenbank gespeichert und automatisch durch DB2 je nach Bedarf aufgerufen, wenn eine Anwendung auf die Datenbank zugreift. Dadurch ist gewährleistet, daß die auf die Daten bezogenen Geschäftsregeln zu jeder Zeit in Kraft bleiben. Wenn eine Geschäftsregel geändert wird, müssen nur die entsprechenden Auslöser angepaßt werden.

Innerhalb einer ausgelösten SQL-Anweisung kann eine benutzerdefinierte Funktion (UDF) aufgerufen werden. Dadurch ist es möglich, daß die ausgelöste Aktion eine Nicht-SQL-Operation durchführen kann, wenn der Auslöser aktiviert wird. Es kann beispielsweise eine E-Mail-Nachricht als Alert-Mechanismus verschickt werden. Weitere Informationen zu Auslösern (Triggers) finden Sie in "Erstellen eines Auslösers" in Handbuch *Systemverwaltung: Implementierung* und im Handbuch *Application Development Guide*.

Weitere Überlegungen zum Datenbankentwurf

Beim Entwurf einer Datenbank muß vorausgeplant werden, auf welche Tabellen Benutzer Zugriff haben sollten. Der Zugriff auf Tabellen wird mit Hilfe von Berechtigungen erteilt oder entzogen. Die höchste Ebene der Berechtigung ist die Berechtigung zur Systemverwaltung (SYSADM). Ein Benutzer mit der Berechtigung SYSADM kann andere Berechtigungen, einschließlich der Berechtigung für den Datenbankadministrator (DBADM), erteilen.

Darüber hinaus gibt es weitere Gesichtspunkte, die im Datenbankentwurf berücksichtigt werden müssen, wie zum Beispiel *Prüfaktivitäten*, *Protokolldaten*, *Übersichtstabellen*, *Sicherheit*, *Datentypisierung*, und *Möglichkeiten zur Parallelverarbeitung*.

Zur Erfassung von *Prüfdaten* muß eventuell jede Aktualisierung an den Daten über einen bestimmten Zeitraum hinweg aufgezeichnet werden. Sie können zum Beispiel eine Prüfdatentabelle jedesmal dann aktualisieren, wenn das Gehalt eines Mitarbeiters geändert wird. Aktualisierungen dieser Tabelle könnten automatisch durchgeführt werden, wenn ein entsprechender Auslöser (Trigger) definiert wird. Prüffunktionen können außerdem über die DB2-Prüffunktion ausgeführt werden. Weitere Informationen finden Sie in "Protokollieren von DB2-Aktivitäten" im Handbuch *Systemverwaltung: Implementierung*.

Aus Gründen der Leistung kann es sinnvoll sein, nur auf eine ausgewählte Menge von Daten zuzugreifen, während die Basisdaten als *Protokoll (History)* verwaltet werden. Sie sollten in Ihren Datenbankentwurf die Anforderungen zur Verwaltung solcher Protokolldaten aufnehmen, z. B. die Anzahl der Monate oder Jahre, die Daten verfügbar sein müssen, bevor sie gelöscht werden können.

Es kann außerdem sinnvoll sein, *Übersichtsdaten* zu erstellen. Zum Beispiel könnten Sie über eine Tabelle verfügen, die Ihre gesamten Mitarbeiterdaten enthält. Angenommen, Sie möchten diese Daten nach Unternehmensbereichen oder Abteilungen in separate Tabellen aufteilen. In diesem Fall wäre es hilfreich, Übersichtstabellen für die einzelnen Unternehmensbereiche oder Abteilungen anzulegen, die auf den Daten der Originaltabelle basieren. Weitere Informationen zu Übersichtstabellen finden Sie in "Erstellen einer Übersichtstabelle" im Handbuch *Systemverwaltung: Implementierung*.

In Ihrem Datenbankentwurf sollten auch die *Sicherheitsaspekte* berücksichtigt werden. Sie können zum Beispiel den Entwurf so gestalten, daß der Zugriff von Benutzern auf bestimmte Arten von Daten über Sicherheitstabellen unterstützt wird. Sie können Zugriffsebenen für verschiedene Arten von Daten und verschiedene Benutzer mit Zugriffsberechtigung für diese Daten definieren. Vertrauliche Daten, wie Personal- und Gehaltsdaten, könnten strikten Sicherheitsbeschränkungen unterliegen. Weitere Informationen zur Sicherheit und zu Berechtigungen finden Sie in "Steuern des Datenbankzugriffs" im Handbuch *Systemverwaltung: Implementierung*.

Sie können Tabellen erstellen, denen ein *strukturierter Typ* zugeordnet ist. Mit solchen typisierten Tabellen können Sie eine hierarchische Struktur mit einer definierten Abhängigkeitsbeziehung zwischen diesen Tabellen erstellen, die als *Typhierarchie* bezeichnet wird. Eine Typhierarchie besteht aus einem Stammtyp sowie aus übergeordneten und untergeordneten Datentypen.

Die Darstellung eines *Referenztyps* wird beim Erstellen des Stammtyps einer Typhierarchie definiert. Das Ziel einer Referenz ist immer eine Zeile in einer typisierten Tabelle oder Sicht.

Weitere Informationen zur Implementierung eines Datenbankentwurfs mit typisierten Zeilen und Tabellen finden Sie in "Implementieren des Datenbankentwurfs" im Handbuch *Systemverwaltung: Implementierung*. Das Handbuch *Versetzen von Daten Dienstprogramme und Referenz* enthält Informationen zum Versetzen von Daten zwischen typisierten Tabellen in einer hierarchischen Struktur.

Mit wachsendem Geschäftsumfang können die zusätzlichen Kapazitäten und Leistungsvorteile von DB2 Enterprise - Extended Edition erforderlich werden. In dieser Umgebung sind die Partitionen Ihrer Datenbank über verschiedene Maschinen bzw. Systeme verteilt, wobei jedes für das Speichern und Abrufen nur eines Teils der Gesamtdatenbank zuständig ist. Jede Partition (oder Knoten) arbeitet bei der Verarbeitung von SQL- oder Dienstprogrammoperationen parallel.

Themen und Überlegungen zu parallelen Operationen werden an den entsprechenden Stellen im gesamten Handbuch berücksichtigt.

Kapitel 8. Entwerfen der physischen Datenbank

Nachdem Sie den logischen Datenbankentwurf entwickelt haben (siehe „Kapitel 7. Entwerfen des logischen Datenbankaufbaus“ auf Seite 101), gibt es eine Reihe von Themen, die hinsichtlich der physischen Umgebung, in der die Datenbank und die Tabellen angelegt werden, zu berücksichtigen sind. Hierzu gehören Kenntnisse bezüglich der Dateien, die für die Unterstützung und Verwaltung Ihrer Datenbank erstellt werden, eine Vorstellung davon, wieviel Speicherbereich zum Speichern Ihrer Daten erforderlich ist, und die Festlegung, wie die zum Speichern Ihrer Daten benötigten Tabellenbereiche verwendet werden sollen.

Die folgenden Themen werden behandelt:

- „Datenbankverzeichnisse“
- „Ermitteln des Platzbedarfs für Tabellen“ auf Seite 134
- „Zusätzlicher Platzbedarf“ auf Seite 142
- „Entwerfen von Knotengruppen“ auf Seite 144
- „Entwerfen und Auswählen von Tabellenbereichen“ auf Seite 153
- „Überlegungen zum Entwerfen einer zusammengeschlossenen Datenbank“ auf Seite 178

Datenbankverzeichnisse

Beim Erstellen einer Datenbank erstellt DB2 ein separates Unterverzeichnis, um Steuerdateien (z. B. Protokollkopfdateien) zu speichern und um Standardtabellenbereichen Behälter zuzuordnen. Objekte, die der Datenbank zugeordnet sind, werden nicht immer im Datenbankverzeichnis gespeichert. Sie können an unterschiedlichen Positionen gespeichert werden, zum Beispiel auf Einheiten.

Die Datenbank wird in dem Exemplar erstellt, das durch die Umgebungsvariable DB2INSTANCE definiert ist, oder in dem Exemplar, zu dem Sie explizit eine Verbindung hergestellt haben (mit dem Befehl ATTACH). Eine Einführung in Exemplare finden Sie in "Verwenden mehrerer Exemplare des Datenbankmanagers" im Handbuch *Systemverwaltung: Implementierung*.

Das auf UNIX-Plattformen verwendete Benennungsschema sieht folgendermaßen aus:

angegebener-pfad/\$DB2INSTANCE/NODEnnnn/SQL00001

Das unter den Betriebssystemen OS/2 und Windows verwendete Benennungsschema ist folgendes:

D:\\$DB2INSTANCE\NODEnnnn\SQL00001

Dabei gilt folgendes:

- angegebener-pfad ist die wahlfreie, benutzerdefinierte Speicherposition zum Installieren des Exemplars.
- NODEnnnn ist die Knotenkennung in einer partitionierten Datenbankumgebung. Der erste Knoten ist NODE0000.
- "D:" ist ein "Laufwerkbuchstabe", der den Datenträger angibt, auf dem das Stammverzeichnis angelegt ist.

SQL00001 enthält Objekte, die der ersten erstellten Datenbank zugeordnet sind. Nachfolgend erstellte Datenbanken erhalten höhere Nummern: SQL00002 usw.

Die Unterverzeichnisse werden in einem Verzeichnis erstellt, das denselben Namen trägt wie das Exemplar des Datenbankmanagers, mit dem Sie bei der Erstellung der Datenbank (durch ATTACH) verbunden sind. (In den Betriebssystemen OS/2 und Windows werden die Unterverzeichnisse unter dem Stammverzeichnis eines bestimmten Datenträgers erstellt, der durch den „Laufwerkbuchstaben“ identifiziert wird.) Diese Exemplar- und Datenbankunterverzeichnisse werden innerhalb des Pfads erstellt, der mit dem Befehl CREATE DATABASE angegeben wurde, und ihre Verwaltung erfolgt automatisch durch den Datenbankmanager. Je nach Plattform kann jedes Exemplar einem Exemplareigner gehören, der die Berechtigung des Systemadministrators (SYSADM) über die zu diesem Exemplar gehörenden Datenbanken hat.

Sie sollten, um potentielle Probleme zu vermeiden, keine Verzeichnisse mit demselben Benennungsschema erstellen und Verzeichnisse, die bereits vom Datenbankmanager erstellt wurden, nicht manuell ändern.

Datenbankdateien

Die folgenden Dateien gehören zu einer Datenbank:

Dateiname	Beschreibung
------------------	---------------------

SQLDBCON	In dieser Datei sind die Optimierungsparameter und Markierungen (Flags) für die Datenbank gespeichert. Informationen zur Änderung von Konfigurationsparametern der Datenbank finden Sie im Handbuch <i>Systemverwaltung: Optimierung</i> .
-----------------	--

SQLLOGCTL.LFH	Diese Datei dient dem Zweck, alle Datenbankprotokolldateien zu verfolgen und zu steuern.
----------------------	--

Syyyyyyy.LOG	Die Datenbankprotokolldateien werden von 0000000 bis
---------------------	--

9999999 durchnummeriert. Die Anzahl dieser Dateien wird durch die Konfigurationsparameter *logprimary* und *logsecond* der Datenbank gesteuert. Die Größe der einzelnen Dateien wird durch den Konfigurationsparameter *logfilesiz* der Datenbank bestimmt.

Bei der Umlaufprotokollierung werden die Dateien erneut verwendet, und die Nummern werden beibehalten. Bei der Archivprotokollierung erhöht sich die Zahl der Dateien in Folge, wenn Protokolle archiviert und neue Protokolle angelegt werden. Wenn 9999999 erreicht wird, beginnt die Nummerierung von vorn.

Standardmäßig werden die Protokolldateien im Verzeichnis `SQLLOGDIR` gespeichert. Das Verzeichnis `SQLLOGDIR` befindet sich im Unterverzeichnis `SQLnnnnn`.

- SQLINSLK** Diese Datei hilft bei der Sicherstellung, daß eine Datenbank nur von einem Exemplar des Datenbankmanagers verwendet wird.
- SQLTMPLK** Diese Datei hilft bei der Sicherstellung, daß eine Datenbank nur von einem Exemplar des Datenbankmanagers verwendet wird.
- SQLSPCS.1** Diese Datei enthält die Definition und den aktuellen Status aller Tabellenbereiche innerhalb der Datenbank.
- SQLSPCS.2** Diese Datei ist eine Sicherungskopie von `SQLSPCS.1`. Ohne eine dieser beiden Dateien ist kein Zugriff auf die Datenbank möglich.
- SQLBP.1** Diese Datei enthält die Definition aller Pufferpools, die in der Datenbank verwendet werden.
- SQLBP.2** Diese Datei ist eine Sicherungskopie von `SQLBP.1`. Ohne eine dieser beiden Dateien ist kein Zugriff auf die Datenbank möglich.
- DB2RHIST.ASC** Diese Datei ist die Protokolldatei der Datenbank. In ihr werden Verwaltungsoperationen an der Datenbank protokolliert, z. B. Sicherungs- und Wiederherstellungsoperationen.
- DB2RHIST.BAK** Diese Datei ist eine Sicherungskopie von `DB2RHIST.ASC`.

Anmerkungen:

1. Nehmen Sie *auf keinen Fall* direkte Änderungen an diesen Dateien vor. Der Zugriff auf diese Dateien kann nur indirekt über die dokumentierten APIs

und mit Tools erfolgen, die diese APIs implementieren (einschließlich des Befehlszeilenprozessors und der Steuerzentrale).

2. Verschieben Sie diese Dateien nicht.
3. Löschen Sie diese Dateien nicht.
4. Das Sichern einer Datenbank oder eines Tabellenbereichs über die API **sqlubkp** (Backup Database), einschließlich der Implementierungen dieser API durch den Befehlszeilenprozessor und die Steuerzentrale, ist die einzige Sicherungsmethode, die unterstützt wird.

Ermitteln des Platzbedarfs für Tabellen

Das Abschätzen der Größe von Datenbankobjekten ist ein ungenaues Unterfangen. Der Systemaufwand, der durch Datenträgerfragmentierung, freien Speicherbereich und die Verwendung von Spalten variabler Länge bedingt ist, macht eine Abschätzung schwierig, weil es eine große Bandbreite an Möglichkeiten für Spaltentypen und Zeilenlängen gibt. Erstellen Sie nach dem Abschätzen der Größe Ihrer Datenbank eine Testdatenbank, und füllen Sie sie mit repräsentativen Daten.

Über die Steuerzentrale können Sie auf eine Reihe von Dienstprogrammen zugreifen, die Sie bei der Ermittlung der Größenanforderungen verschiedener Datenbankobjekte unterstützen:

- Sie können ein Objekt auswählen und das Dienstprogramm "Größe schätzen" verwenden. Mit diesem Dienstprogramm können Sie die aktuelle Größe eines vorhandenen Objekts, wie z. B. einer Tabelle, ermitteln. Sie können dann das Objekt ändern, und das Dienstprogramm berechnet neue Schätzwerte für das Objekt. Dieses Dienstprogramm hilft bei der Abschätzung des ungefähren Speicherbedarfs unter Berücksichtigung zukünftiger Zuwächse. Es bietet mehr als einen einzelnen Schätzwert für die Größe des Objekts. Es gibt auch mögliche Größenbereiche für das Objekt an: die minimale Größe, die auf den aktuellen Werten beruht, und die mögliche Maximalgröße.
- Sie können die Abhängigkeiten zwischen Objekten bestimmen, indem Sie das Dialogfenster "Zugehörige anzeigen" verwenden.
- Sie können ein beliebiges Datenbankobjekt im Exemplar auswählen und "DDL generieren" anfordern. Diese Funktion verwendet das Dienstprogramm **db2look**, um Datendefinitionsanweisungen für die Datenbank zu generieren. Informationen zu diesem Dienstprogramm enthält das Handbuch *Command Reference*.

In allen diesen Fällen steht Ihnen entweder der Knopf "SQL anzeigen" oder der Knopf "Befehl anzeigen" zur Verfügung. Sie können die generierten SQL-Anweisungen oder Befehle auch in Prozedurdateien sichern, um diese später zu verwenden. In allen diesen Dienstprogrammen werden Sie durch eine Online-Hilfefunktion unterstützt.

Sie sollten diese Einrichtungen bei der Planung Ihrer physischen Datenbank-anforderungen berücksichtigen.

Bei der Abschätzung der Größe einer Datenbank müssen die Auswirkungen der folgenden Elemente berücksichtigt werden:

- „Systemkatalogtabellen“
- „Benutzertabellendaten“
- „Langfelddaten“ auf Seite 138
- „Daten großer Objekte (LOB-Daten)“ auf Seite 138
- „Indexbereich“ auf Seite 139

Der Platzbedarf folgender Elemente wird nicht behandelt:

- Datei für das lokale Datenbankverzeichnis
- Datei für das Systemdatenbankverzeichnis
- Bei der Dateiverwaltung entstehender Systemaufwand, den das Betriebssystem benötigt, zum Beispiel:
 - Dateiblockgröße
 - Speicherbereich für die Verzeichnissteuerung

Systemkatalogtabellen

Systemkatalogtabellen werden erstellt, wenn eine Datenbank erstellt wird. Der Umfang dieser Systemtabellen nimmt in dem Maße zu, wie der Datenbank Datenbankobjekte und Zugriffsrechte hinzugefügt werden. Zu Beginn belegen diese Tabellen einen Plattenspeicherplatz von ungefähr 3,5 MB.

Die Größe des Speicherbereichs, der für die Katalogtabellen zugeordnet wird, hängt von der Art des Tabellenbereichs und der Größe des durch EXTENT-SIZE definierten Bereichs des Tabellenbereichs mit den Katalogtabellen ab. Wenn z. B. ein DMS-Tabellenbereich mit einem EXTENTSIZE-Wert von 32 verwendet wird, wird dem Katalogtabellenbereich anfangs ein Speicherbereich von 20 MB zugeordnet. Weitere Informationen finden Sie in „Entwerfen und Auswählen von Tabellenbereichen“ auf Seite 153.

Anmerkung: Für Datenbanken mit mehreren Partitionen befinden sich die Katalogtabellen nur in der Partition, von der aus der Befehl CREATE DATABASE abgesetzt wurde. Plattenspeicherplatz für die Katalogtabellen ist nur für diese Partition erforderlich.

Benutzertabellendaten

Standardmäßig werden die Tabellendaten auf Seiten von jeweils 4 KB gespeichert. Jede Seite enthält (unabhängig von der Seitengröße) 76 Byte Systemaufwand für den Datenbankmanager. Dadurch bleiben 4020 Byte für die Benutzerdaten (oder Zeilen), obwohl keine Zeile auf einer 4-KB-Seite die

Länge von 4005 Byte überschreiten kann. Eine Zeile kann sich *nicht* über mehrere Seiten erstrecken. Wenn Sie Seiten mit jeweils 4 KB verwenden, sind maximal 500 Spalten möglich.

Tabellendatenseiten enthalten *keine* Daten für Spalten mit den Datentypen LONG VARCHAR, LONG VARCHARIC, BLOB, CLOB oder DBCLOB. Die Zeilen einer Tabellendatenseite enthalten jedoch einen Deskriptor für diese Spalten. (Informationen zur Ermittlung des Platzbedarfs für die Tabellenobjekte, die diese Datentypen enthalten, finden Sie in den Abschnitten „Langfelddaten“ auf Seite 138 und „Daten großer Objekte (LOB-Daten)“ auf Seite 138.)

Zeilen werden normalerweise an der ersten passenden Stelle in einer Tabelle eingefügt. Die Datei wird nach dem ersten verfügbaren Speicherbereich durchsucht (unter Verwendung einer Liste der freien Speicherbereiche), der groß genug ist, um die neue Zeile aufzunehmen. Das Aktualisieren einer vorhandenen Zeile erfolgt an der ursprünglichen Stelle, es sei denn, der freie Bereich der 4-KB-Seite reicht nicht aus, um die aktualisierte Zeile aufzunehmen. In diesem Fall wird an der Position, an der sich die Originalzeile befand, ein Datensatz erstellt, der auf die neue Speicherposition der aktualisierten Zeile in der Tabellendatei verweist.

Bei Verwendung der Anweisung ALTER TABLE APPEND ON werden Daten immer angehängt und es werden keine Informationen zum freien Speicherbereich der Datenseiten aufgezeichnet. Weitere Informationen zu dieser Anweisung finden Sie im Handbuch *SQL Reference*.

Die Anzahl der 4-KB-Seiten für jede Benutzertabelle in der Datenbank kann nach folgender Berechnung abgeschätzt werden:

$$\text{ABRUNDEN}(4020/(\text{durchschnittszeilengröße} + 10)) = \text{datensätze_pro_seite}$$

Das Ergebnis wird in folgende Berechnung eingefügt:

$$(\text{zahl_der_datensätze}/\text{datensätze_pro_seite}) * 1,1 = \text{anzahl_der_seiten}$$

Dabei ist die durchschnittliche Zeilenlänge die Summe der durchschnittlichen Spaltengrößen (Informationen über die Größe jeder Spalte finden Sie in der Beschreibung der Anweisung CREATE TABLE im Handbuch *SQL Reference*), und der Faktor "1,1" dient zur Kalkulation des Systemaufwands.

Anmerkung: Diese Formel ergibt nur einen Schätzwert. Die Genauigkeit des Schätzwerts nimmt ab, wenn die Datensatzlänge durch Fragmentierung und Überlaufsätze variiert.

Sie können auch Pufferpools oder Tabellenbereiche mit einer Seitengröße von 8 KB, 16 KB oder 32 KB erstellen. Alle Tabellen, die innerhalb eines Tabellenbereichs einer bestimmten Größe erstellt wurden, besitzen eine übereinstim-

mende Seitengröße. Die maximale Größe für ein einzelnes Tabellen- oder Indexobjekt beträgt 512 GB bei einer einer Seitengröße von 32 KB. Wenn Sie Seiten mit einer Größe von 8 KB, 16 KB oder 32 KB verwenden, sind maximal 1012 Spalten möglich. Für eine 4-KB-Seite beträgt die maximale Spaltenanzahl 500. Die maximalen Zeilenlängen variieren ebenfalls je nach Seitengröße:

- Bei einer Seitengröße von 4 KB beträgt die maximale Zeilenlänge 4005 Byte.
- Bei einer Seitengröße von 8 KB beträgt die maximale Zeilenlänge 8101 Byte.
- Bei einer Seitengröße von 16 KB beträgt die maximale Zeilenlänge 16 293 Byte.
- Bei einer Seitengröße von 32 KB beträgt die maximale Zeilenlänge 32 677 Byte.

Durch größere Seiten wird es möglich, die Zahl der Indexstufen in einem Index zu verringern. Wenn Sie mit OLTP-Anwendungen arbeiten, die wahlfreie Lese- und Schreibzugriffe durchführen, ist eine geringere Seitengröße empfehlenswert, weil dadurch weniger Pufferspeicher durch unerwünschte Zeilen belegt wird. Wenn Sie mit DSS-Anwendungen (Decision Support System) arbeiten, die jeweils auf eine große Anzahl aufeinanderfolgender Zeilen zugreifen, sind größere Seiten empfehlenswert, weil dadurch weniger E/A-Anforderungen erforderlich sind, um eine bestimmte Anzahl Zeilen zu lesen. Eine Ausnahme bildet der Fall, wenn die Zeilengröße kleiner ist als die Seitengröße dividiert durch 255. In diesem Fall wird auf jeder Seite ungenutzter Speicherbereich verschenkt. (Weil jede Seite maximal nur 255 Zeilen aufnehmen kann.) Zur Verringerung des unbenutzten Speicherbereichs kann eine geringere Seitengröße besser geeignet sein.

Eine Sicherung kann nicht in einer anderen Seitengröße wiederhergestellt werden.

Sie können keine IXF-Datendateien importieren, die mehr als 755 Spalten beinhalten. Weitere Informationen zum Importieren von Daten in Tabellen und IXF-Datendateien finden Sie im Handbuch *Versetzen von Daten Dienstprogramme und Referenz*.

Deklarierte temporäre Tabellen können nur in einem eigenen Tabellenbereichstyp für temporäre Benutzertabellen erstellt werden. Es gibt keinen Standardtabellenbereich für temporäre Benutzertabellen. Temporäre Tabellen können keine Daten mit LONG-Typen enthalten. Die Tabellen werden implizit gelöscht, wenn eine Anwendung die Verbindung zur Datenbank trennt. Schätzungen über den entsprechenden Platzbedarf sollten dies berücksichtigen.

Langfelddaten

Langfelddaten werden in einem separaten Tabellenobjekt gespeichert, das anders als bei anderen Datentypen strukturiert ist (siehe „Benutzertabellendaten“ auf Seite 135 und „Daten großer Objekte (LOB-Daten)“).

Die Daten werden in Bereichen von 32 KB gespeichert, die sich aus Segmenten zusammensetzen, deren Größen sich aus dem Produkt der Zweierpotenzen und 512 Byte errechnen. (Diese Segmente können also aus 512 Byte, 1024 Byte, 2048 Byte usw. bis 32 700 Byte bestehen.)

Langfelddatentypen (LONG VARCHAR oder LONG VARGRAPHIC) werden auf eine Weise gespeichert, die eine problemlose Neuverwendung freien Speicherbereichs ermöglicht. Die Informationen zur Zuordnung und zu freien Speicherbereichen werden in Zuordnungsseiten von jeweils 4 KB gespeichert, die gelegentlich im Objekt erscheinen.

Der Bereich des freien Speicherplatzes in den Objekten ist von der Größe der Langfelddaten sowie davon abhängig, ob diese Größe bei jedem Vorkommen der Daten relativ konstant ist. Bei Dateneinträgen von mehr als 255 Byte kann dieser nicht genutzte Speicherplatz bis zu 50 Prozent der Größe der Langfelddaten ausmachen.

Wenn Zeichendaten kleiner als die Seitengröße sind und sie zusammen mit dem Rest der Daten in den Datensatz passen, sollten anstelle der Datentypen LONG VARCHAR oder LONG VARGRAPHIC die Datentypen CHAR, GRAPHIC, VARCHAR oder VARGRAPHIC verwendet werden.

Daten großer Objekte (LOB-Daten)

Daten großer Objekte (LOB-Daten) werden in zwei separaten Tabellenobjekten gespeichert, die anders als bei anderen Datentypen strukturiert sind (siehe „Benutzertabellendaten“ auf Seite 135 und „Langfelddaten“).

Bei der Abschätzung des für LOB-Daten erforderlichen Speicherbereichs müssen Sie die beiden Tabellenobjekte berücksichtigen, die zum Speichern von Daten dieser Datentypen verwendet werden:

- **LOB-Datenobjekte**

Die Daten werden in Bereichen von 64 MB gespeichert, die sich aus Segmenten zusammensetzen, deren Größen sich aus dem Produkt der Zweierpotenzen und 1024 Byte errechnen. (Diese Segmente können also aus 1024 Byte, 2048 Byte, 4096 Byte usw. bis 64 MB bestehen.)

Zur Verringerung des für LOB-Daten erforderlichen Plattenspeicherplatzes können Sie den Parameter COMPACT in der Klausel für die *LOB-Optionen* der Anweisungen CREATE TABLE und ALTER TABLE angeben. Die Option COMPACT minimiert die Größe des für die LOB-Daten erforderlichen

Speicherplatzes dadurch, daß die LOB-Daten in kleinere Segmente aufgeteilt werden können. Dieser Prozess beinhaltet keine Datenkomprimierung, sondern verwendet lediglich den kleinstmöglichen Speicherbereich bis zur nächsten 1-KB-Grenze. Die Angabe der Option COMPACT kann zu einer geringeren Leistung führen, wenn Daten an LOB-Werte angehängt werden. Der Umfang des freien Speicherbereichs innerhalb der LOB-Datenobjekte wird vom Umfang der Aktualisierungs- und Löschartivitäten sowie von der Größe der eingefügten LOB-Werte beeinflusst.

- **LOB-Zuordnungsobjekte**

Die Informationen zur Zuordnung und zu freien Speicherbereichen werden in Zuordnungsseiten von jeweils 4 KB gespeichert, die von den eigentlichen Daten getrennt sind. Die Anzahl dieser 4-KB-Seiten ist vom Umfang der Daten (einschließlich des nicht genutzten Speicherbereichs) abhängig, die für die LOB-Daten zugeordnet wurden. Der Systemaufwand errechnet sich wie folgt: eine 4-KB-Seite pro 64 GB plus eine 4-KB-Seite pro 8 MB.

Wenn Zeichendaten kleiner als die Seitengröße sind und sie zusammen mit dem Rest der Daten in den Datensatz passen, sollten anstelle der Datentypen BLOB, CLOB oder DBCLOB die Datentypen CHAR, GRAPHIC, VARCHAR oder VARGRAPHIC verwendet werden.

Indexbereich

Für jeden Index kann der erforderliche Speicherbereich wie folgt abgeschätzt werden:

$$(\text{Durchschnittliche Indexschlüsselgröße} + 8) * \text{Anzahl der Zeilen} * 2$$

Dabei gilt folgendes:

- Die „durchschnittliche Indexschlüsselgröße“ ist die Bytezahl jeder Spalte im Indexschlüssel. Im Abschnitt zur Anweisung CREATE TABLE im Handbuch *SQL Reference* finden Sie Informationen zum Berechnen der Bytezahl für Spalten mit unterschiedlichen Datentypen. (Verwenden Sie bei der Abschätzung der durchschnittlichen Spaltengröße für VARCHAR- und VARGRAPHIC-Spalten die aktuelle Datengröße plus ein Byte. Verwenden Sie nicht die deklarierte Maximalgröße.)
- Der Faktor 2 ist für den Systemaufwand, z. B. für Nichtblattseiten und freien Speicherbereich.

Anmerkung: Fügen Sie für jede Spalte, die einen Nullwert (NULL) zuläßt, ein zusätzliches Byte für den Nullanzeiger hinzu.

Bei der Indexerstellung ist temporärer Speicherplatz erforderlich. Der maximal während der Indexerstellung erforderliche temporäre Speicherplatz kann folgendermaßen abgeschätzt werden:

$$(\text{Durchschnittliche Indexschlüsselgröße} + 8) * \text{Anzahl der Zeilen} * 3,2$$

Dabei wird der Faktor 3,2 für den Indexaufwand sowie für den Speicherbereich einkalkuliert, der für während der Indexerstellung anfallende Sortierungen erforderlich ist.

Anmerkung: Für nichteindeutige Indizes werden zum Speichern doppelter Schlüsseleinträge nur vier Byte benötigt. Die oben gezeigten Berechnungen gehen von eindeutigen Indizes aus. Der zum Speichern eines Index erforderliche Speicherbereich kann durch die oben gezeigte Formel eventuell zu groß abgeschätzt werden.

Die beiden folgenden Formeln können zum Abschätzen der Anzahl von Blattseiten eines Index verwendet werden. (Die zweite Formel liefert einen etwas genaueren Schätzwert.) Die Genauigkeit dieser Schätzwerte hängt im wesentlichen davon ab, wie gut die verwendeten Durchschnittswerte die tatsächlichen Daten widerspiegeln.

Anmerkung: Für einen SMS-Tabellenbereich ist der minimale Speicherbedarf 12 KB. Für DMS-Tabellenbereiche ist der minimale Speicherbedarf ein durch EXTENTSIZE definierter Bereich.

- Die Durchschnittszahl von Schlüsseln pro Blattseite kann mit folgender Formel grob abgeschätzt werden:

$$\frac{(0,9 * (U - (M*2))) * (D + 1)}{K + 6 + (4 * D)}$$

Dabei gilt folgendes:

- U, der verwendbare Speicherbereich auf einer Seite, entspricht ungefähr der Seitengröße minus 100. Bei einer Seitengröße von 4096 ist U gleich 3996.
- $M = U / (8 + \text{minimale Schlüsselgröße})$
- D = durchschnittliche Anzahl mehrfacher Werte pro Schlüsselwert
- K = *durchschnittliche Schlüsselgröße*

Beachten Sie, daß die Werte für *minimale Schlüsselgröße* und *durchschnittliche Schlüsselgröße* ein Byte extra für jeden Teil des Schlüssels haben müssen, der einen Nullwert enthalten kann, und ein zusätzliches Byte für die Länge jedes Teils des Schlüssels mit variabler Länge.

Wenn INCLUDE-Spalten vorhanden sind, müssen sie in den Werten für *minimale Schlüsselgröße* und *durchschnittliche Schlüsselgröße* berücksichtigt werden.

0,9 kann durch einen beliebigen, mit $(100 - \text{pctfree})/100$ berechneten Wert ersetzt werden, wenn während der Indexerstellung ein anderer Prozentwert für freien Speicherbereich (pctfree) angegeben wurde als der Standardwert zehn.

- Die Durchschnittszahl von Schlüsseln pro Blattseite kann mit folgender Formel etwas genauer abgeschätzt werden:

$$L = \text{Blattseitenzahl} = X / (\text{Durchschnittszahl von Schlüsseln pro Blattseite})$$

Dabei ist X die Gesamtzahl der Zeilen in der Tabelle.

Einen Schätzwert für die Originalgröße eines Indexes können Sie wie folgt berechnen:

$$L + 2L/(\text{Durchschnittszahl von Schlüsseln pro Blattseite}) * \text{Seitengröße}$$

Für DMS-Tabellenbereiche addieren Sie die Größen aller Indizes einer Tabelle zusammen, und runden Sie auf ein Vielfaches des Werts für EXTENTSIZE für den Tabellenbereich auf, in dem sich der Index befindet.

Sie sollten weiteren Speicherbereich für das Anwachsen des Index durch INSERT/UPDATE-Vorgänge bereitstellen, die zur Teilung von Seiten führen können.

Durch die folgenden Berechnungen erhalten Sie einen genaueren Schätzwert für die Originalgröße des Indexes sowie einen Schätzwert für die Anzahl der Indexstufen. (Dies ist möglicherweise von besonderem Interesse, wenn in der Indexdefinition INCLUDE-Spalten verwendet werden.) Die Durchschnittszahl von Schlüsseln pro Nichtblattseite kann mit folgender Formel grob abgeschätzt werden:

$$\frac{(0,9 * (U - (M*2))) * (D + 1)}{K + 12 + (8 * D)}$$

Dabei gilt folgendes:

- U, der verwendbare Speicherbereich auf einer Seite, entspricht ungefähr der Seitengröße minus 100. Bei einer Seitengröße von 4096 ist U gleich 3996.
- D ist die durchschnittliche Anzahl mehrfacher Werte pro Schlüsselwert auf Nichtblattseiten (dieser Wert ist deutlich kleiner als für Blattseiten. Sie können ihn zur Vereinfachung der Berechnung auf 0 setzen).
- M = $U / (8 + \text{minimale Schlüsselgröße für Nichtblattseiten})$
- K = *durchschnittliche Schlüsselgröße* für Nichtblattseiten

Die *minimale Schlüsselgröße* und die *durchschnittliche Schlüsselgröße* für Nichtblattseiten stimmen mit den Werten für Blattseiten überein, sofern keine INCLUDE-Spalten vorkommen. INCLUDE-Spalten werden auf Nichtblattseiten nicht gespeichert.

Sie sollten $0,9$ nur durch $(100 - \text{pctfree})/100$ ersetzen, wenn dieser Wert größer als $0,9$ ist, weil auf Nichtblattseiten bei der Indexerstellung maximal 10% Speicherbereich frei bleibt.

Die Zahl der Nichtblattseiten kann mit folgender Formel abgeschätzt werden:

```
if L > 1 then {P++; Z++;}
while (Y > 1)
{
  P = P + Y
  Y = Y / N
  Z++
}
```

Dabei gilt folgendes:

- P ist die Anzahl von Seiten(zunächst 0).
- L ist die Anzahl der Blattseiten.
- N ist die Anzahl von Schlüsseln pro Nichtblattseite.
- $Y = L / N$
- Z ist die Anzahl der Ebenen in der Indexbaumstruktur (zunächst 1).

Als Gesamtzahl der Seiten ergibt sich:

$$T = (L + P + 2) * 1,0002$$

Die zusätzlichen 0,02% sind für Systemaufwand (einschließlich Speicherabbildseiten).

Der für die Indexerstellung erforderliche Speicherbereich läßt sich folgendermaßen abschätzen:

$$T * \text{Seitengröße}$$

Zusätzlicher Platzbedarf

Folgende zusätzliche Speicherbereiche sind erforderlich:

- „Speicherbereich für die Protokolldatei“
- „Temporärer Arbeitsbereich“ auf Seite 144

Speicherbereich für die Protokolldatei

Der Speicherbereich (in Byte), der für Protokolldateien erforderlich ist, reicht von:

$$(\text{logprimary} * (\text{logfilsiz} + 2) * 4096) + 8192$$

bis:

$$((\text{logprimary} + \text{logsecond}) * (\text{logfilsiz} + 2) * 4096) + 8192$$

Dabei gilt folgendes:

- *logprimary* ist die Anzahl der primären Protokolldateien, die in der Konfigurationsdatei der Datenbank definiert sind.
- *logsecond* ist die Anzahl der sekundären Protokolldateien, die in der Konfigurationsdatei der Datenbank definiert sind.
- *logfilsiz* ist die Anzahl der Seiten in jeder Protokolldatei, die in der Konfigurationsdatei der Datenbank definiert sind.
- 2 ist die Anzahl der Kopfseiten, die für jede Protokolldatei erforderlich ist.
- 4096 ist die Anzahl der Byte einer Seite.
- 8192 ist die Größe (in Byte) der Protokollsteuerdatei.

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

Anmerkung: Der Gesamtspeicherbereich für die aktive Protokolldatei kann eine Größe von 32 GB nicht überschreiten.

Der obere Grenzwert für den Speicherbereich für Protokolldateien ist abhängig von der tatsächlichen Anzahl der sekundären Protokolldateien, die der Datenbankmanager zur Laufzeit benötigt. Dieser obere Grenzwert wird in der Regel nicht benötigt bzw. lediglich zu gelegentlich auftretenden Zeiten hoher Aktivität benötigt.

Wenn für die Datenbank die aktualisierende Wiederherstellung aktiviert ist, sollten spezielle Speicheranforderungen für Protokolle berücksichtigt werden:

- Bei aktiviertem Konfigurationsparameter *logretain* werden die Protokolldateien im Protokollpfadverzeichnis archiviert. Der Online-Plattenspeicherplatz wird schließlich belegt, sofern Sie die Protokolldateien nicht an eine andere Speicherposition verschieben.
- Bei aktiviertem Konfigurationsparameter *userexit* verschiebt ein BenutzerAusgangsprogramm die archivierten Protokolldateien an eine andere Speicherposition. Zusätzlicher Speicherbereich ist weiterhin erforderlich für folgende Dateien:
 - Online archivierte Protokolle, die darauf warten, vom BenutzerAusgangsprogramm verschoben zu werden
 - Neue Protokolle, die für künftige Zwecke formatiert werden

Temporärer Arbeitsbereich

Einige SQL-Anweisungen benötigen für die Verarbeitung temporäre Tabellen (z. B. eine Arbeitsdatei für Sortieroperationen, die nicht im Speicher vorgenommen werden können). Diese temporären Tabellen benötigen Platten-speicherplatz. Die Größe des erforderlichen Speichers hängt von den Abfragen sowie von der Größe der Ergebnistabellen ab und kann nicht abgeschätzt werden.

Mit Hilfe des Datenbanksystemmonitors und der APIs zum Abfragen des Tabellenbereichs können Sie verfolgen, wie groß der verwendete Arbeitsbereich während des normalen Betriebsablaufs ist.

Entwerfen von Knotengruppen

Eine *Knotengruppe* ist eine benannte Gruppe aus einem oder mehreren Knoten, die als zu einer Datenbank gehörig definiert sind. Jede Datenbankpartition, die Teil einer Datenbanksystemkonfiguration ist, muß bereits in einer *Partitionskonfigurationsdatei* namens `db2nodes.cfg` definiert sein. Eine Knotengruppe kann eine Datenbankpartition, mehrere Datenbankpartitionen und auch alle für das Datenbanksystem definierten Datenbankpartitionen enthalten.

Eine neue Knotengruppe wird mit Hilfe der Anweisung `CREATE NODEGROUP` erstellt und kann mit Hilfe der Anweisung `ALTER NODEGROUP` geändert werden. Es können eine oder mehrere Datenbankpartitionen einer Knotengruppe hinzugefügt oder aus ihr entfernt werden. Die Datenbankpartitionen müssen in der Datei `db2nodes.cfg` definiert sein, bevor die Knotengruppe geändert wird. Tabellenbereiche befinden sich innerhalb von Knotengruppen. Tabellen befinden sich innerhalb von Tabellenbereichen.

Wenn eine Knotengruppe erstellt oder geändert wird, wird ihr eine *Partitionierungszuordnung* zugeordnet. Die Partitionierungszuordnung wird vom Datenbankmanager in Verbindung mit einem *Partitionierungsschlüssel* und einem Hash-Algorithmus verwendet, um festzulegen, in welcher Datenbankpartition der Knotengruppe eine bestimmte Datenzeile gespeichert wird. Weitere Informationen zu Partitionierungszuordnungen finden Sie im Abschnitt „Partitionierungszuordnungen“ auf Seite 147. Weitere Informationen zu Partitionierungsschlüsseln enthält der Abschnitt „Partitionierungsschlüssel“ auf Seite 148.

in einer nichtpartitionierten Datenbank ist kein Partitionierungsschlüssel und keine Partitionierungszuordnung erforderlich. Wenn Sie eine nichtpartitionierte Datenbank verwenden, spielen Überlegungen zu Knotengruppen keine Rolle. Eine *Datenbankpartition* ist ein Teil einer Datenbank, mit eigenen Benutzerdaten, Indizes, Konfigurationsdateien und Transaktionsprotokollen. Vom Datenbankmanager werden Standardknotengruppen verwendet, die erstellt werden, wenn die Datenbank erstellt wird. `IBMCATGROUP` ist die

Standardknotengruppe für den Tabellenbereich, der die Systemkatalogtabellen enthält. IBMTEMPGROUP ist die Standardknotengruppe für die Tabellenbereiche mit den temporären Systemtabellen. IBMDEFAULTGROUP ist die Standardknotengruppe für die Tabellenbereiche, die die benutzerdefinierten Tabellen enthalten, die Sie dort speichern. Ein temporärer Benutzertabellenbereich für deklarierte temporäre Benutzertabellen kann in IBMDEFAULTGROUP bzw. in einer beliebigen benutzererstellten Knotengruppe, jedoch nicht in IBMTEMPGROUP erstellt werden.

Wenn Sie eine Knotengruppe mit mehreren Partitionen verwenden, beachten Sie die folgenden Gesichtspunkte:

- In einer Knotengruppe mit mehreren Partitionen können Sie nur dann einen eindeutigen Index erstellen, wenn er eine Obermenge des Partitionierungsschlüssels ist.
- Abhängig von der Anzahl der Datenbankpartitionen in der Datenbank können eine oder mehrere Knotengruppen mit einer Partition und eine oder mehrere Knotengruppen mit mehreren Partitionen vorhanden sein.
- Jeder Datenbankpartition muß eine eindeutige Partitionsnummer zugeordnet sein. Dieselbe Datenbankpartition kann in einer oder mehreren Knotengruppen enthalten sein.
- Zur Gewährleistung einer schnellen Wiederherstellung der Datenbankpartition, die die Systemkatalogtabellen enthält, sollten Sie keine Benutzertabellen in derselben Datenbankpartition anlegen. Dies wird erreicht, wenn die Benutzertabellen in Knotengruppen untergebracht werden, die nicht die Datenbankpartition der Knotengruppe IBMCATGROUP enthalten.

Kleine Tabellen sollten in Knotengruppen mit einer Einzelpartition angelegt werden, außer wenn Sie Nutzen aus der *Kollokation* mit einer größeren Tabelle ziehen wollen. Unter Kollokation (Zusammenfassung) ist die Platzierung von Zeilen aus verschiedenen Tabellen, die zusammengehörige Daten enthalten, in die gleiche Datenbankpartition zu verstehen. Zusammengefaßte Tabellen ermöglichen DB2 den Einsatz effizienterer Verknüpfungsstrategien. Durch Kollokation zusammengefaßte Tabellen können sich in einer Knotengruppe mit einer Einzelpartition befinden. Tabellen werden als durch Kollokation zusammengefaßt betrachtet, wenn sie sich in einer Knotengruppe mit mehreren Partitionen befinden, dieselbe Anzahl von Spalten im Partitionierungsschlüssel haben und die Datentypen der sich entsprechenden Spalten partititionskompatibel sind. Zeilen in zusammengefaßten Tabellen mit demselben Wert im Partitionierungsschlüssel werden in derselben Datenbankpartition gespeichert. Tabellen können sich in separaten Tabellenbereichen in derselben Knotengruppe befinden und trotzdem als durch Kollokation zusammengefaßt betrachtet werden.

Mittelgroße Tabellen sollten nicht über zu viele Datenbankpartitionen verteilt werden. Zum Beispiel ist es möglich, daß eine 100-MB-Tabelle in einer Knotengruppe mit 16 Partitionen zu besseren Leistungen führt als in einer Knotengruppe mit 32 Partitionen.

Sie können Knotengruppen dazu verwenden, OLTP-Tabellen (OLTP - Online-Transaktionsverarbeitung) von Entscheidungshilfetabellen (DSS-Tabellen - Decision Support System) zu trennen, um sicherzustellen, daß die Leistung von OLTP-Transaktionen nicht beeinträchtigt wird.

Überlegungen zum Aufbau von Knotengruppen

Der logische Datenbankentwurf und die Menge an Daten, die verarbeitet werden muß, legen eine Antwort auf die Frage nahe, ob eine Datenbank partitioniert werden muß. Dieser Abschnitt behandelt die folgenden Themen zur Datenbankpartitionierung:

- „Datenpartitionierung“
- „Partitionierungszuordnungen“ auf Seite 147
- „Partitionierungsschlüssel“ auf Seite 148
- „Tabellenkollokation“ auf Seite 151
- „Partitionskompatibilität“ auf Seite 151
- „Replizierte Übersichtstabellen“ auf Seite 152

Datenpartitionierung

DB2 unterstützt ein partitioniertes Speichermodell, das Ihnen ermöglicht, Daten über verschiedene Datenbankpartitionen in der Datenbank verteilt zu speichern. Dies bedeutet, daß die Daten zwar physisch über mehr als eine Datenbankpartition verteilt werden, jedoch so auf sie zugegriffen werden kann, als ob sie sich an derselben Speicherposition befänden. Anwendungen und Benutzer, die auf Daten in einer partitionierten Datenbank zugreifen, brauchen die physische Speicherposition der Daten nicht zu kennen.

Die Daten werden trotz der physischen Trennung als logische Einheit verwendet und verwaltet. Benutzer können wählen, wie Ihre Daten partitioniert werden, indem Sie Partitionierungsschlüssel festlegen. Benutzer können darüber hinaus bestimmen, auf welche und auf wie viele Datenbankpartitionen Ihre Tabellendaten verteilt werden können, indem sie den Tabellenbereich und die Knotengruppe auswählen, in der die Daten gespeichert werden sollen. Weiterhin wird eine aktualisierbare Partitionierungszuordnung zusammen mit einem Hash-Algorithmus verwendet, um die Zuordnung von Werten der Partitionierungsschlüssel zu Datenbankpartitionen anzugeben, wodurch das Plazieren und Abrufen der einzelnen Datenzeilen festgelegt wird. Dadurch können Sie die Auslastung für große Tabellen über eine partitionierte Datenbank hinweg verteilen, während kleinere Tabellen in einer oder mehreren Datenbankpartitionen gespeichert werden können. Jede Datenbankpartition

verfügt über lokale Indizes für die in ihr gespeicherten Daten, so daß für den lokalen Datenzugriff eine erhöhte Leistung erzielt wird.

Sie sind nicht darauf beschränkt, alle Tabelle über alle Datenbankpartitionen in der Datenbank verteilt zu speichern. DB2 unterstützt eine *Teilclusterung*, das heißt, Sie können die Tabellen und die zugehörigen Tabellenbereiche über eine Teilgruppe der Datenbankpartitionen im System (d. h. eine Knotengruppe) verteilen.

Als alternative Methode, wenn Sie Tabellen in den einzelnen Datenbankpartitionen angelegt haben wollen, sollten Sie die Verwendung von Übersichtstabellen in Erwägung ziehen, die anschließend repliziert werden. Sie können eine Übersichtstabelle mit den von Ihnen benötigten Informationen erstellen und diese dann auf jeden Knoten replizieren. Weitere Informationen finden Sie in „Replizierte Übersichtstabellen“ auf Seite 152.

Partitionierungszuordnungen

In einer Umgebung mit einer partitionierten Datenbank muß der Datenbankmanager eine Möglichkeit besitzen, zu wissen, welche Tabellenzeilen in welcher Datenbankpartition gespeichert werden. Der Datenbankmanager muß wissen, wo er die benötigten Daten findet. Zum Auffinden der Daten verwendet er ein Zuordnungsverzeichnis, das als *Partitionierungszuordnung* bezeichnet wird.

Eine Partitionierungszuordnung ist eine intern generierte Tabelle von entweder 4 096 Einträgen für Knotengruppen mit mehreren Partitionen oder einem einzigen Eintrag für Knotengruppen mit einer Einzelpartition. Für eine Knotengruppe mit nur einer Partition enthält die Partitionierungszuordnung nur einen Eintrag mit der Partitionsnummer der Datenbankpartition, in der alle Zeilen einer Datenbanktabelle gespeichert werden. Für Knotengruppen mit mehreren Partitionen werden die Partitionsnummern der Knotengruppe immer wiederholt reihum angegeben. Vergleichbar mit der Verwendung einer Stadtkarte, die durch Gitterlinien in Sektoren unterteilt ist, verwendet der Datenbankmanager einen *Partitionierungsschlüssel*, um die Speicherposition (die Datenbankpartition) zu bestimmen, in der die Daten gespeichert werden.

Nehmen Sie zum Beispiel an, daß Sie eine Datenbank in vier Datenbankpartitionen (mit den Nummern 0–3) erstellt haben. Die Partitionierungszuordnung für die Knotengruppe IBMDEFAULTGROUP dieser Datenbank wäre wie folgt:

0 1 2 3 0 1 2 ...

Wenn eine Knotengruppe in der Datenbank mit den Datenbankpartitionen 1 und 2 erstellt würde, wäre die Partitionierungszuordnung für diese Knotengruppe wie folgt:

1 2 1 2 1 2 1 ...

Wenn der Partitionierungsschlüssel für eine in die Datenbank zu ladende Tabelle eine ganze Zahl (Integer) mit möglichen Werten zwischen 1 und 500 000 ist, wird der Partitionierungsschlüssel mit einem Hash-Verfahren auf eine Partitionsnummer zwischen 0 und 4 095 abgebildet. Diese Nummer wird als Index für die Partitionierungszuordnung verwendet, um die Datenbankpartition für die betreffende Zeile auszuwählen.

Abb. 35 zeigt, wie der Zeile mit dem Partitionierungsschlüsselwert (c1, c2, c3) die Partitionsnummer 2 zugeordnet wird, die ihrerseits auf Datenbankpartition n5 verweist.

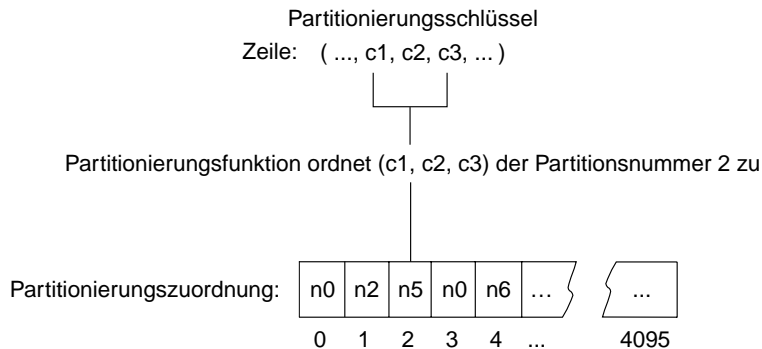


Abbildung 35. Datenverteilung mit einer Partitionierungszuordnung

Eine solche Partitionierungszuordnung ist eine flexible Steuermethode für die Speicherung von Daten in einer partitionierten Datenbank. Wenn zu einem zukünftigen Zeitpunkt die Notwendigkeit eintritt, die Datenverteilung auf die Datenbankpartitionen Ihrer Datenbank zu ändern, können Sie dazu das Dienstprogramm zur Datenumverteilung verwenden. Dieses Dienstprogramm ermöglicht Ihnen, die Datenverteilung neu auszugleichen oder bewußt ungleichmäßig zu gestalten. Weitere Informationen zu diesem Dienstprogramm finden Sie im Abschnitt zum Umverteilen von Daten über Datenbankpartitionen des Handbuchs *Systemverwaltung: Optimierung*.

Mit Hilfe der API **sqlugtpi** zum Abrufen von Informationen zur Tabellenpartitionierung können Sie eine Kopie der Partitionierungszuordnung abrufen, die Sie anzeigen können. Weitere Informationen zu dieser API finden Sie im Handbuch *Administrative API Reference*.

Partitionierungsschlüssel

Ein *Partitionierungsschlüssel* ist eine Spalte (oder Spaltengruppe), die dazu verwendet wird, die Partition zu bestimmen, in der eine bestimmte Datenzeile gespeichert ist. Ein Partitionierungsschlüssel wird in einer Tabelle mit Hilfe der Anweisung CREATE TABLE definiert. Wenn kein Partitionierungsschlüssel für eine Tabelle in einem Tabellenbereich, der über mehr als eine Daten-

bankpartition in einer Knotengruppe verteilt ist, definiert wird, wird ein Partitionierungsschlüssel standardmäßig aus der ersten Spalte des Primärschlüssels erstellt. Wenn kein Primärschlüssel angegeben ist, wird standardmäßig die erste Spalte der Tabelle, die keine langen Daten enthält, als Partitionierungsschlüssel angenommen. (*Lang* beinhaltet hier alle langen Datentypen (LONG) und alle LOB-Datentypen). Wenn Sie eine Tabelle in einem Tabellenbereich erstellen, der zu einer Knotengruppe mit nur einer Partition gehört, und einen Partitionierungsschlüssel erstellen wollen, müssen Sie den Partitionierungsschlüssel explizit definieren. In diesem Fall wird kein Standard-schlüssel erstellt.

Wenn keine Spalte die Voraussetzungen für einen standardmäßig erstellten Partitionierungsschlüssel erfüllt, wird die Tabelle ohne Partitionierungsschlüssel erstellt. Tabellen ohne Partitionierungsschlüssel sind nur in Knotengruppen mit einer Partition zulässig. Sie können Partitionierungsschlüssel auch später noch mit der Anweisung ALTER TABLE hinzufügen oder entfernen. Das Ändern des Partitionierungsschlüssels ist nur möglich in einer Tabelle, deren Tabellenbereich zu einer Knotengruppe mit einer Einzelpartition gehört.

Die Auswahl eines guten Partitionierungsschlüssels ist von großer Bedeutung. Sie sollten sich über folgendes im klaren sein:

- Wie der Zugriff auf Tabellen erfolgen soll
- Die Art der Abfrageauslastung
- Die vom Datenbanksystem angewendeten Verknüpfungsstrategien

Wenn Kollokation kein Haupt Gesichtspunkt ist, dann zeichnet sich ein guter Partitionierungsschlüssel für eine Tabelle dadurch aus, daß er die Daten gleichmäßig über alle Datenbankpartitionen in der Knotengruppe verteilt. Der Partitionierungsschlüssel jeder Tabelle in einem Tabellenbereich, der einer Knotengruppe zugeordnet ist, bestimmt, ob die Tabellen durch Kollokation zusammengefaßt werden. Tabellen werden als zusammengefaßt betrachtet, wenn folgende Bedingungen gelten:

- Die Tabellen liegen in Tabellenbereichen, die in derselben Knotengruppe sind.
- Die Partitionierungsschlüssel in jeder Tabelle haben dieselbe Anzahl von Spalten.
- Die Datentypen der entsprechenden Spalten sind partitionskompatibel.

Dadurch ist sichergestellt, daß Zeilen zusammengefaßter Tabellen mit denselben Werten für ihre Partitionierungsschlüssel in derselben Partition gespeichert werden. Weitere Informationen zur Partitionskompatibilität finden Sie in „Partitionskompatibilität“ auf Seite 151. Weitere Informationen zur Tabellenkollokation finden Sie in „Tabellenkollokation“ auf Seite 151.

Ein ungeeigneter Partitionierungsschlüssel kann zu einer ungleichmäßigen Datenverteilung führen. Spalten mit ungleichmäßig verteilten Daten und Spalten mit einer kleinen Anzahl unterschiedlicher Werte sollten nicht als Partitionierungsschlüssel ausgewählt werden. Die Anzahl der unterschiedlichen Werte muß ausreichend groß sein, um eine gleichmäßige Verteilung der Zeilen auf alle Datenbankpartitionen in der Knotengruppe sicherzustellen. Der Aufwand für die Anwendung des Hash-Algorithmus zur Partitionierung ist proportional zur Größe des Partitionierungsschlüssels. Der Partitionierungsschlüssel kann nicht mehr als 16 Spalten enthalten. Jedoch wird bei weniger Spalten eine bessere Leistung erzielt. Nicht benötigte Spalten sollten daher nicht in den Partitionierungsschlüssel aufgenommen werden.

Die folgenden Punkte sollten bei der Definition von Partitionierungsschlüsseln beachtet werden:

- Die Erstellung einer Mehrpartitionstabelle, die ausschließlich lange Datentypen (LONG VARCHAR, LONG VARGRAPHIC, BLOB, CLOB oder DBCLOB) enthält, wird nicht unterstützt.
- Die Definition des Partitionierungsschlüssels kann nicht geändert werden.
- Auch kann der Wert einer Spalte des Partitionierungsschlüssels für eine Zeile der Tabelle nicht aktualisiert werden.
- Werte für Spalten des Partitionierungsschlüssels können nur gelöscht oder eingefügt werden.
- Der Partitionierungsschlüssel sollte die am häufigsten an Verknüpfungen beteiligten Spalten enthalten.
- Der Partitionierungsschlüssel sollte außerdem aus Spalten bestehen, die häufig von Klauseln GROUP BY betroffen sind.
- Jeder eindeutige Schlüssel oder Primärschlüssel muß alle Spalten des Partitionierungsschlüssels enthalten.
- In einer OLTP-Umgebung (Online-Transaktionsverarbeitung) sollten alle Spalten des Partitionierungsschlüssels an der Transaktion über Gleichheitsvergleichselemente (=) mit Konstanten oder Host-Variablen beteiligt sein. Nehmen Sie zum Beispiel an, Sie haben eine Personalnummer *emp_no*, die in Transaktionen häufig wie folgt oder ähnlich verwendet wird:

```
UPDATE emp_table SET ... WHERE  
emp_no = host-variable
```

In diesem Fall wäre die Spalte EMP_NO ein guter einspaltiger Partitionierungsschlüssel für die Tabelle EMP_TABLE.

Hash-Partitionierung heißt die Methode, mit der die Speicherposition jeder Zeile in der partitionierten Tabelle ermittelt wird. Diese Methode funktioniert folgendermaßen:

1. Der Hash-Algorithmus wird auf den Wert des Partitionierungsschlüssels angewendet und generiert eine Partitionsnummer zwischen 0 und 4095.

2. Die Partitionierungszuordnung wird bei der Erstellung einer Knotengruppe erstellt. Die Partitionsnummern werden immer wieder in derselben Reihenfolge nacheinander wiederholt, bis die Partitionierungszuordnung gefüllt ist. Weitere Informationen zu Partitionierungszuordnungen finden Sie im Abschnitt „Partitionierungszuordnungen“ auf Seite 147.
3. Die Partitionsnummer wird als Index für die Position in der Partitionierungszuordnung verwendet. Die Nummer an dieser Position in der Partitionierungszuordnung ist die Nummer der Datenbankpartition, in der die Zeile gespeichert wird.

Tabellenkollokation

Sie stellen möglicherweise fest, daß zwei oder mehr Tabellen häufig zusammen Daten als Antwort für bestimmte Abfragen liefern. In diesem Fall ist es sinnvoll, zusammengehörige Daten aus solchen Tabellen möglichst nah beieinander zu speichern. In einer Umgebung, in der die Datenbank physisch auf zwei oder mehr Datenbankpartitionen verteilt ist, muß es eine Möglichkeit geben, zusammengehörige Teile der verteilten Tabellen so nah wie möglich beieinander zu halten. Diese Möglichkeit wird als *Tabellenkollokation* bezeichnet.

Tabellen sind in einer Kollokation zusammengefaßt, wenn sie in derselben Knotengruppe gespeichert sind und ihre Partitionierungsschlüssel kompatibel sind. Durch Speichern beider Tabellen in derselben Knotengruppe wird sichergestellt, daß sie eine gemeinsame Partitionierungszuordnung haben. Die Tabellen können sich in verschiedenen Tabellenbereichen befinden, jedoch müssen die Tabellenbereiche derselben Knotengruppe zugeordnet sein. Die Datentypen der entsprechenden Spalten in den jeweiligen Partitionierungsschlüsseln müssen *partitionskompatibel* sein. Informationen zur Partitionskompatibilität finden Sie in „Partitionskompatibilität“.

DB2 kann beim Zugriff auf mehr als eine Tabelle bei einer Verknüpfung oder einer Unterabfrage erkennen, daß sich die zu verknüpfenden Daten in derselben Datenbankpartition befinden. Wenn dies geschieht, kann DB2 entscheiden, die Verknüpfung oder Unterabfrage in der Datenbankpartition auszuführen, in der die Daten gespeichert sind, anstatt die Daten zwischen Datenbankpartitionen auszutauschen. Diese Möglichkeit, Verknüpfungen oder Unterabfragen in der Datenbankpartition auszuführen, bietet wesentliche Leistungsvorteile. Weitere Informationen finden Sie unter „Zusammengefaßte Verknüpfungen“ im Handbuch *Systemverwaltung: Optimierung*.

Partitionskompatibilität

Die Basisdatentypen entsprechender Spalten von Partitionierungsschlüsseln werden verglichen und können als *partitionskompatibel* deklariert werden. Partitionskompatible Datentypen haben die Eigenschaft, daß ein bestimmter Partitionierungsalgorithmus zwei Variablen mit jeweils einem dieser Datentypen dieselbe Partitionsnummer zuordnet, wenn sie denselben Wert haben.

Partitionskompatibilität hat folgende Merkmale:

- Ein Wert des Basisdatentyps ist mit einem anderen desselben Basistyps kompatibel.
- Interne Formate werden für die Datentypen DATE (Datum), TIME (Uhrzeit) und TIMESTAMP (Zeitmarke) verwendet. Sie sind untereinander nicht kompatibel, und keiner dieser Typen ist mit dem Zeichendatentyp (CHAR) kompatibel.
- Die Partitionskompatibilität wird von Spalten mit den Definitionen NOT NULL oder FOR BIT DATA nicht berührt.
- Nullwerte (NULL) kompatibler Datentypen werden auf identische Weise behandelt, Nullwerte nicht kompatibler Datentypen möglicherweise nicht.
- Die Basisdatentypen eines benutzerdefinierten Datentyps werden zur Analyse der Partitionskompatibilität verwendet.
- Dezimalzahlen desselben Werts im Partitionierungsschlüssel werden identisch behandelt, auch wenn ihre Anzahl an Kommastellen und ihre Genauigkeit unterschiedlich sind.
- Folgende Leerzeichen in Zeichenfolgen (CHAR, VARCHAR, GRAPHIC oder VARGRAPHIC) werden vom Hash-Algorithmus ignoriert.
- BIGINT, SMALLINT und INTEGER sind kompatible Datentypen.
- REAL und FLOAT sind kompatible Datentypen.
- CHAR und VARCHAR verschiedener Längen sind kompatible Datentypen.
- GRAPHIC und VARGRAPHIC sind kompatible Datentypen.
- Die Partitionskompatibilität gilt nicht für die Datentypen LONG VARCHAR, LONG VARGRAPHIC, CLOB, DBCLOB und BLOB, da sie in Partitionierungsschlüsseln nicht unterstützt werden.

Replizierte Übersichtstabellen

Eine *Übersichtstabelle* ist eine Tabelle, die durch eine Abfrage definiert ist, mit der auch die Daten für die Tabelle festgelegt werden. Durch Übersichtstabellen kann die Leistungsfähigkeit von Abfragen erhöht werden. Wenn DB2 erkennt, daß ein Teil einer Abfrage durch eine Übersichtstabelle aufgelöst werden könnte, kann der Datenbankmanager die Abfrage so abwandeln, daß die entsprechende Übersichtstabelle verwendet wird.

In einer partitionierten Datenbankumgebung können Sie Übersichtstabellen replizieren. Durch die Verwendung *replizierter Übersichtstabellen* können Sie die Leistung von Abfragen erhöhen. Eine replizierte Übersichtstabelle basiert auf einer Tabelle, die möglicherweise in einer Knotengruppe mit einer Einzelpartition erstellt wurde, die Sie jedoch über alle Datenbankpartitionen in der Knotengruppe replizieren wollen. Die replizierte Übersichtstabelle wird durch Ausführen der Anweisung CREATE TABLE mit dem Schlüsselwort REPLICATED erstellt. Das Schlüsselwort REPLICATED kann nur für eine Übersichtstabelle angegeben werden, die mit der Option REFRESH DEFERRED wurde.

Weitere Informationen zu Übersichtstabellen finden Sie in "Erstellen einer Übersichtstabelle" im Handbuch *Systemverwaltung: Implementierung*.

Mit Hilfe von replizierten Übersichtstabellen können Sie Kollokationen zwischen Tabellen herstellen, die normalerweise nicht kollokiert sind. Replizierte Übersichtstabellen sind besonders nützlich für Verknüpfungen großer Faktentabellen mit kleinen Dimensionstabellen. Zur Minimierung des erforderlichen zusätzlichen Speicherbedarfs sowie des Aufwands zum Aktualisieren aller Replikate, sollten zu replizierende Tabellen klein sein und nicht häufig aktualisiert werden.

Anmerkung: Sie sollten auch in Erwägung ziehen, umfangreiche Tabellen, die selten aktualisiert werden, zu replizieren: Der Zusatzaufwand für eine Replikation durch die mit Hilfe der Kollokation erzielten Leistungsvorteile wettgemacht.

Durch Angeben eines geeigneten Vergleichselements in der Unterauswahlklausel, die zum Definieren der replizierten Tabelle verwendet wird, können Sie ausgewählte Spalten, ausgewählte Zeilen (oder beides) replizieren.

Weitere Informationen zu replizierten Übersichtstabellen finden Sie im Abschnitt zur Anweisung CREATE TABLE im Handbuch *SQL Reference*. Weitere Informationen zu zusammengefaßten Verknüpfungen finden Sie unter „Zusammengefaßte Verknüpfungen“ im Handbuch *Systemverwaltung: Implementierung*.

Entwerfen und Auswählen von Tabellenbereichen

Ein Tabellenbereich ist ein Speichermodell, das eine Zwischenstufe zwischen einer Datenbank und den in ihr gespeicherten Tabellen zur Verfügung stellt. Tabellenbereiche befinden sich in Knotengruppen (Nodegroups). Mit ihrer Hilfe können Sie die Speicherposition der Datenbank- und Tabellendaten direkt bestimmten Behältern zuweisen. (Ein Behälter kann ein Verzeichnisname, Einheitenname oder Dateiname sein.) Die möglichen Vorzüge zeigen sich in einer besseren Leistung, einer flexibleren Konfiguration und einer höheren Integrität.

Informationen zum Erstellen oder Ändern eines Tabellenbereichs finden Sie in "Erstellen eines Tabellenbereichs" bzw. "Ändern eines Tabellenbereichs" im Handbuch *Systemverwaltung: Implementierung*.

Da sich Tabellenbereiche in Knotengruppen befinden, bestimmt der zur Speicherung einer Tabelle ausgewählte Tabellenbereich, wie die Daten dieser Tabelle auf die Datenbankpartitionen in einer Knotengruppe verteilt werden. Ein einzelner Tabellenbereich kann sich über mehrere Behälter erstrecken. Mehrere Behälter (eines oder mehrerer Tabellenbereiche) können auf derselben

physischen Platte (bzw. auf demselben Laufwerk) erstellt werden. Zur Erzielung einer optimalen Leistung sollte jeder Behälter eine andere Platte verwenden. Abb. 36 zeigt ein Beispiel für die Anordnungsbeziehung zwischen Tabellen und Tabellenbereichen innerhalb einer Datenbank und den Behältern, die dieser Datenbank zugeordnet sind.

Datenbank

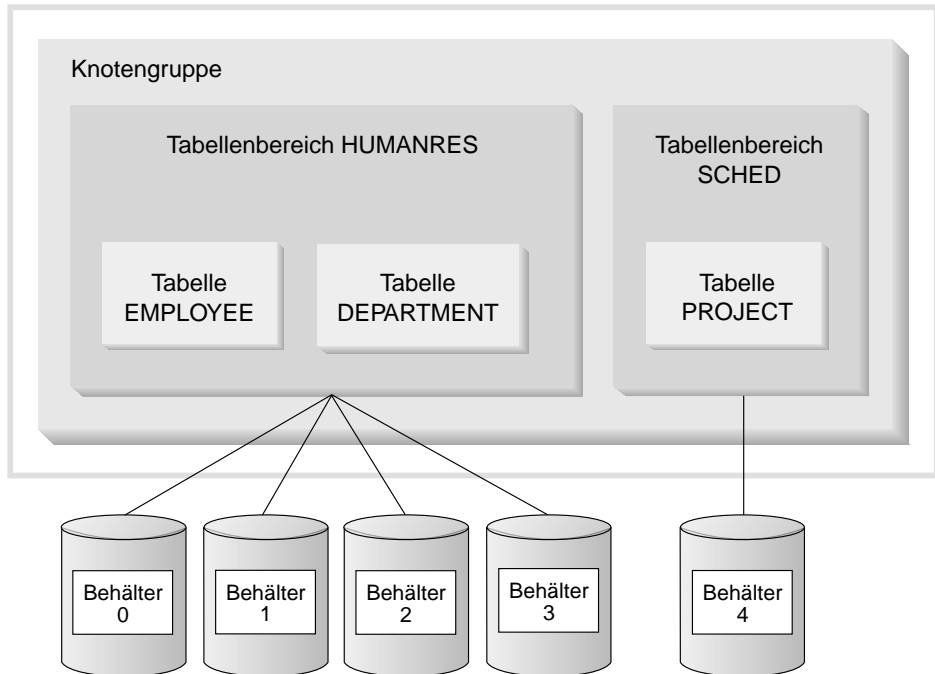


Abbildung 36. Tabellenbereiche und Tabellen in einer Datenbank

Die Tabellen EMPLOYEE und DEPARTMENT befinden Sie im Tabellenbereich HUMANRES, der sich über die Behälter 0, 1, 2 und 3 erstreckt. Die Tabelle PROJECT befindet sich im Tabellenbereich SCHED im Behälter 4. Dieses Beispiel zeigt jeden Behälter auf einer getrennten Platte.

Der Datenbankmanager versucht, die Datenmenge möglichst gleichmäßig über die Behälter zu verteilen. Das heißt, es werden alle Behälter zum Speichern der Daten verwendet. Die Anzahl der Seiten, die der Datenbankmanager in einen Behälter schreibt, bevor er einen anderen Behälter verwendet, wird mit dem Parameter *EXTENTSIZE* definiert. Der Datenbankmanager beginnt beim Speichern der Tabellendaten nicht immer mit dem ersten Behälter.

Abb. 37 auf Seite 155 zeigt den Tabellenbereich HUMANRES mit einem Wert von zwei 4-KB-Seiten für *EXTENTSIZE* und vier Behältern mit einer kleinen

Anzahl zugeordneter Speicherbereiche. Die Tabellen DEPARTMENT und EMPLOYEE haben jeweils sieben Seiten und verteilen sich über alle vier Behälter.

Tabellenbereich HUMANRES

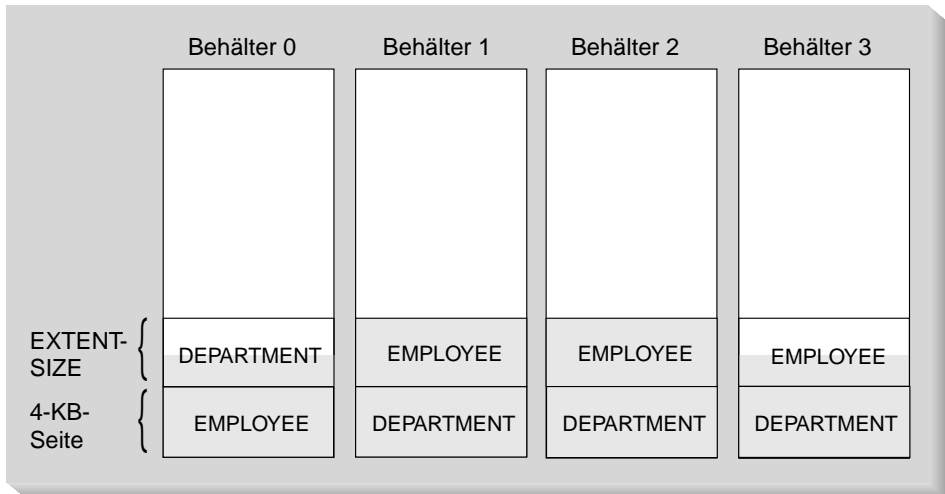


Abbildung 37. Behälter und EXTENT-SIZE-Bereiche

Eine Datenbank muß mindestens drei Tabellenbereiche enthalten:

- Einen *Katalogtabellenbereich*, der alle Systemkatalogtabellen für die Datenbank enthält. Dieser Tabellenbereich heißt SYSCATSPACE und kann nicht gelöscht werden. IBMCATGROUP ist die Standardknotengruppe für diesen Tabellenbereich.
- Einen oder mehrere *Benutzertabellenbereiche*, die alle benutzerdefinierten Tabellen enthalten. Standardmäßig wird ein Tabellenbereich namens USERSPACE1 erstellt. IBMDEFAULTGROUP ist die Standardknotengruppe für diesen Tabellenbereich.

Beim Erstellen einer Tabelle sollten Sie einen Tabellenbereichsnamen angeben, andernfalls kann es zu unerwünschten Ergebnissen kommen. Wenn Sie keinen Tabellenbereichsnamen angeben, wird die Tabelle nach folgenden Richtlinien gespeichert: Wenn benutzererstellte Tabellenbereiche existieren, wird derjenige mit der kleinsten Seitengröße, die für diese Tabelle ausreicht, ausgewählt. Ansonsten wird der Tabellenbereich USERSPACE1 verwendet, falls er über eine für die Tabelle ausreichende Seitengröße verfügt. Wenn keine Tabellenbereiche mit ausreichender Seitengröße vorhanden sind, wird die Tabelle nicht erstellt.

Die Seitengröße für eine Tabelle wird entweder durch die Zeilengröße oder durch die Anzahl von Spalten bestimmt. Die für eine Zeile maximal zulässige Länge hängt von der Seitengröße des Tabellenbereichs ab, in dem die

Tabelle erstellt wird. Gültige Werte für die Seitengröße sind 4 KB (Standardwert), 8 KB, 16 KB und 32 KB. Sie können einen Tabellenbereich mit einer Seitengröße für die Basistabelle und einen weiteren Tabellenbereich mit einer anderen Seitengröße für lange Daten (LONG) oder LOB-Daten verwenden. (Hier ist wiederum zu beachten, daß SMS keine Tabellen unterstützt, die sich über mehrere Tabellenbereiche erstrecken, im Gegensatz zu DMS.) Wenn die Anzahl der Spalten oder die Zeilengröße die Grenze für die Seitengröße eines Tabellenbereichs überschreitet, wird ein Fehler zurückgegeben (SQLSTATE 42997).

- Einen oder mehrere *temporäre Tabellenbereiche*, die temporäre Tabellen enthalten. Temporäre Tabellenbereiche können *temporäre Systemtabellenbereiche* oder *temporäre Benutzertabellenbereiche* sein. Eine Datenbank muß mindestens über einen temporären Tabellenbereich verfügen. Standardmäßig wird ein temporärer Systemtabellenbereich namens TEMPSPACE1 bei der Erstellung der Datenbank erstellt. IBMTEMPGROUP ist die Standardknotengruppe für diesen Tabellenbereich. Temporäre Benutzertabellenbereiche werden *nicht* standardmäßig bei der Erstellung einer Datenbank erstellt.

Wenn eine Datenbank mehr als einen temporären Tabellenbereich verwendet, werden temporäre Objekte reihum auf die temporären Tabellenbereiche verteilt.

Wenn Abfragen auf Tabellen in Tabellenbereichen ausgeführt werden, die mit einer größeren Seitengröße als dem Standardwert von 4 KB definiert sind (z. B. eine Klausel ORDER BY auf 1012 Spalten), schlagen manche dieser Abfragen möglicherweise fehl. Dies geschieht, wenn keine temporären Tabellenbereiche mit einer größeren Seitengröße definiert sind. Möglicherweise müssen Sie einen temporären Tabellenbereich mit einer größeren Seitengröße (8 KB, 16 KB oder 32 KB) erstellen. Jede DML-Anweisung (Datenbearbeitungssprache) könnte fehlschlagen, sofern kein temporärer Tabellenbereich mit derselben Seitengröße wie die größte in dem Benutzertabellenbereich verwendete Seitengröße vorhanden ist.

Sie sollten einen einzelnen temporären SMS-Tabellenbereich definieren, dessen Seitengröße der Seitengröße entspricht, die in den meisten der Benutzertabellenbereiche verwendet wird. Dies sollte für typische Umgebungen und Auslastungen angemessen sein. Siehe auch „Empfehlungen für temporäre Tabellenbereiche“ auf Seite 170.

In einer Umgebung mit partitionierten Datenbanken verfügt der Katalogknoten über alle drei Tabellenbereiche, während die anderen Datenbankpartitionen nur die Tabellenbereiche TEMPSPACE1 und USERSPACE1 enthalten.

Es gibt zwei Typen von Tabellenbereich, die beide in einer einzelnen Datenbank verwendet werden können:

- „SMS-Tabellenbereich“ auf Seite 157: Der Dateimanager des Betriebssystems steuert den Speicherbereich.

- „DMS-Tabellenbereich“ auf Seite 161: Der Datenbankmanager steuert den Speicherbereich.

SMS-Tabellenbereich

In einem vom Betriebssystem verwalteten Tabellenbereich (SMS - System Managed Space) ordnet der Dateisystemmanager des Betriebssystems den Speicherbereich zu, in dem die Tabelle gespeichert werden soll, und verwaltet diesen Bereich. Das Speichermodell enthält in der Regel viele Dateien, die Tabellenobjekte darstellen, die im Speicherbereich des Dateisystems gespeichert sind. Der Benutzer entscheidet über die Speicherposition der Dateien, DB2 verwaltet ihre Namen, und das Dateisystem ist für die Verwaltung auf dem System zuständig. Durch Steuern der Datenmenge, die in jede Datei geschrieben wird, verteilt der Datenbankmanager die Daten gleichmäßig auf die Behälter der Tabellenbereiche. Ein SMS-Tabellenbereich ist der Standardtabellenbereich.

Jeder Tabelle ist mindestens eine physische SMS-Datei zugeordnet. Eine Liste dieser Dateien sowie eine Beschreibung des Inhalts finden Sie in „Physische SMS-Dateien“ auf Seite 159.

In einem SMS-Tabellenbereich wird eine Datei mit dem Anwachsen des Objekts um jeweils eine Seite erweitert. Wenn Sie eine bessere Leistung benötigen, können Sie die Aktivierung der mehrseitigen Dateizuordnung in Betracht ziehen. Dadurch kann das System der Datei mehr als eine Seite gleichzeitig zuordnen. Zur Aktivierung der mehrseitigen Dateizuordnung müssen Sie das Dienstprogramm **db2empfa** ausführen. In einer Umgebung mit partitionierten Datenbanken muß dieses Dienstprogramm in jeder Datenbankpartition ausgeführt werden. Wenn die mehrseitige Dateizuordnung aktiviert ist, kann sie nicht inaktiviert werden. Weitere Informationen zum Dienstprogramm **db2empfa** finden Sie im Handbuch *Command Reference*.

Sie sollten SMS-Tabellenbereiche explizit mit der Option `MANAGED BY SYSTEM` im Befehl `CREATE DATABASE` oder in der Anweisung `CREATE TABLESPACE` definieren. Dabei müssen Sie zwei Schlüsselfaktoren beim Entwerfen Ihrer SMS-Tabellenbereiche beachten:

- Behälter für den Tabellenbereich

Sie müssen die Anzahl der Behälter angeben, die Sie für Ihren Tabellenbereich verwenden wollen. Es ist sehr wichtig, alle gewünschten Behälter zu ermitteln, weil Sie nach dem Erstellen des Tabellenbereichs keine Behälter löschen oder hinzufügen können. In einer partitionierten Datenbankumgebung kann die Anweisung `ALTER TABLESPACE` verwendet werden, um beim Hinzufügen einer neuen Partition zu einer Knotengruppe für einen SMS-Tabellenbereich Behälter für die neue Partition hinzuzufügen.

Jeder Behälter, der für einen SMS-Tabellenbereich verwendet wird, identifiziert einen absoluten oder relativen Verzeichnisnamen. Jedes dieser Ver-

zeichnisse kann sich auf einem anderen Dateisystem (oder einer anderen physischen Platte) befinden. Die Maximalgröße des Tabellenbereichs kann wie folgt abgeschätzt werden:

Anzahl von Behältern * (maximale Dateisystemgröße, die vom Betriebssystem unterstützt wird)

Diese Formel setzt voraus, daß jedem Behälter ein bestimmtes Dateisystem zugeordnet wird und daß für jedes Dateisystem der maximale Speicherbereich verfügbar ist. In der Praxis ist dies möglicherweise nicht der Fall, und die Maximalgröße des Tabellenbereichs kann sehr viel kleiner sein.

Anmerkung: Gehen Sie beim Definieren der Behälter mit besonderer Sorgfalt vor. Wenn die Behälter bereits Dateien oder Verzeichnisse enthalten, wird eine Fehlermeldung (SQL0298N) zurückgegeben.

- Parameter EXTENTSIZE für den Tabellenbereich

Der Wert für den Parameter EXTENTSIZE kann nur beim Erstellen des Tabellenbereichs angegeben werden. Da spätere Änderungen daran nicht möglich sind, ist es wichtig, einen geeigneten Wert für EXTENTSIZE anzugeben. Weitere Informationen finden Sie in „Auswählen eines Werts für EXTENTSIZE“ auf Seite 169.

Wenn Sie beim Erstellen eines Tabellenbereichs für den Parameter EXTENTSIZE keinen Wert angeben, erstellt der Datenbankmanager den Tabellenbereich mit dem Standardwert, der durch den Konfigurationsparameter *dft_extent_sz* der Datenbank definiert ist (Das Handbuch *Systemverwaltung: Optimierung* enthält weitere Informationen zu diesem Parameter). Dieser Konfigurationsparameter wird anfangs auf der Grundlage der Informationen gesetzt, die beim Erstellen der Datenbank angegeben werden. Wird der Parameter DFT_EXTENT_SZ nicht mit dem Befehl CREATE DATABASE angegeben, wird der Standardwert auf 32 gesetzt.

Um geeignete Werte für die Anzahl der Behälter und für den Parameter EXTENTSIZE für den Tabellenbereich festlegen zu können, benötigen Sie folgende Kenntnisse:

- Den oberen Grenzwert, den Ihr Betriebssystem für die Größe eines logischen Dateisystems festlegt

Zum Beispiel haben einige Betriebssysteme eine obere Begrenzung von 2 GB. Wenn Sie also ein Tabellenobjekt mit einer Größe von 64 GB erstellen möchten, benötigen Sie auf dieser Art System mindestens 32 Behälter.

Wenn Sie einen Tabellenbereich erstellen, können Sie Behälter angeben, die sich auf verschiedenen Dateisystemen befinden, und dadurch die Menge der Daten erhöhen, die in der Datenbank gespeichert werden können.

- Die Art und Weise, wie der Datenbankmanager die einem Tabellenbereich zugeordneten Datendateien und Behälter verwaltet

Die erste Datei mit Tabellendaten (SQL00001.DAT) wird im ersten für den Tabellenbereich angegebenen Behälter erstellt. Diese Datei darf soweit anwachsen, bis sie die durch den Wert für EXTENTSIZE festgelegte Größe erreicht. Nach Erreichen dieser Größe schreibt der Datenbankmanager Daten in die Datei SQL00001.DAT im nächsten Behälter. Dieser Prozeß wird fortgesetzt, bis alle Behälter Dateien des Namens SQL00001.DAT enthalten. Wenn dies eintritt, kehrt der Datenbankmanager zum ersten Behälter zurück. Dieser Prozeß (der auch als *Striping - einheitenübergreifendes Lesen und Schreiben von Daten* bezeichnet wird) wird über die Behälterverzeichnisse fortgesetzt, bis ein Behälter voll ist (SQL0289N) oder vom Betriebssystem kein weiterer Speicherbereich mehr zugeordnet werden kann (Fehlernachricht: Datenträger voll). Das Striping-Verfahren wird auch für Index- (SQLnnnnn.INX), Langfeld- (SQLnnnnn.LF) und LOB-Dateien (SQLnnnnn.LB und SQLnnnnn.LBA) verwendet.

Anmerkung: Der SMS-Tabellenbereich ist voll, sobald irgendeiner seiner Behälter voll ist. Daher ist es wichtig, jedem Behälter dieselbe Menge an Speicherbereich zuzuordnen.

Um die Daten gleichmäßiger über die Behälter zu verteilen, bestimmt der Datenbankmanager den Behälter, der zuerst verwendet werden soll, indem er den Wert der Tabellen-ID (1 im Beispiel oben) Modulo die Anzahl der Behälter ermittelt. Die Behälter werden beginnend mit dem Wert 0 durchnumeriert.

Weitere Informationen über die Dateien, die in einem SMS-Tabellenbereich verwendet werden, finden Sie in „Physische SMS-Dateien“.

Physische SMS-Dateien

Folgende Dateien befinden sich im Verzeichnisbehälter eines SMS-Tabellenbereichs:

Dateiname	Beschreibung
------------------	---------------------

SQLTAG.NAM	In jedem Behälterunterverzeichnis befindet sich jeweils eine dieser Dateien. Der Datenbankmanager verwendet sie, wenn Sie eine Verbindung zur Datenbank herstellen, um zu prüfen, ob die Datenbank vollständig und konsistent ist.
-------------------	--

SQLxxxxx.DAT	Dies ist eine Tabellendatei. Alle Tabellenzeilen werden hier gespeichert, ausgenommen Daten der Typen LONG VARCHAR, LONG VARGRAPHIC, BLOB, CLOB und DBCLOB.
---------------------	---

SQLxxxxx.LF	Eine Datei, die Daten der Typen LONG VARCHAR oder LONG VARGRAPHIC (auch als „Langfelddaten“ bezeichnet)
--------------------	---

enthält. Die Datei wird nur erstellt, wenn die Tabelle Spalten der Datentypen LONG VARCHAR oder LONG VARGRAPHIC enthält.

SQLxxxxx.LB Eine Datei, die Daten der Typen BLOB, CLOB oder DBCLOB (auch als „LOB-Daten“ bezeichnet) enthält. Die Datei wird nur erstellt, wenn die Tabelle Spalten der Datentypen BLOB, CLOB oder DBCLOB enthält.

SQLxxxxx.LBA Eine Datei, die Informationen zur Zuordnung und zu freien Speicherbereichen der Dateien SQLxxxxx.LB enthält.

SQLxxxxx.INX Eine Indexdatei für eine Tabelle. Alle Indizes für die entsprechende Tabelle werden in dieser einen Datei gespeichert. Die Datei wird nur erstellt, wenn Indizes definiert wurden.

Anmerkung: Beim Löschen von Indizes wird der von der Indexdatei (.INX) belegte Speicherbereich erst dann physisch freigegeben, wenn die Indexdatei gelöscht wird. Die Indexdatei wird gelöscht, wenn alle Indizes der Tabelle entfernt (und festgeschrieben) werden oder wenn die Tabelle reorganisiert wird. Wenn die Indexdatei nicht gelöscht wird, wird der Speicherbereich als frei markiert, sobald der Löschvorgang (DROP) festgeschrieben ist. Im Anschluß daran kann der Speicherbereich für künftige Indexerstellungs- und Indexverwaltungsoperationen erneut verwendet werden.

SQLxxxxx.DTR Eine temporäre Datendatei für das Reorganisieren einer DAT-Datei. Beim Reorganisieren einer Tabelle erstellt das REORG-Dienstprogramm (über den Befehl REORG TABLE) eine Tabelle in einem der temporären Systemtabellenbereiche. Diese temporären Tabellenbereiche können definiert werden, um andere Behälter als die für die benutzerdefinierten Tabellen angegebenen Behälter zu verwenden.

SQLxxxxx.LFR Eine temporäre Datendatei für das Reorganisieren einer LF-Datei. Beim Reorganisieren einer Tabelle erstellt das REORG-Dienstprogramm (über den Befehl REORG TABLE) eine Tabelle in einem der temporären Systemtabellenbereiche. Diese temporären Tabellenbereiche können definiert werden,

um andere Behälter als die für die benutzerdefinierten Tabellen angegebenen Behälter zu verwenden.

SQLxxxx.RLB

Eine temporäre Datendatei für das Reorganisieren einer LB-Datei. Beim Reorganisieren einer Tabelle erstellt das REORG-Dienstprogramm (über den Befehl REORG TABLE) eine Tabelle in einem der temporären Systemtabellenbereiche. Diese temporären Tabellenbereiche können definiert werden, um andere Behälter als die für die benutzerdefinierten Tabellen angegebenen Behälter zu verwenden.

SQLxxxx.RBA

Eine temporäre Datendatei für das Reorganisieren einer LBA-Datei. Beim Reorganisieren einer Tabelle erstellt das REORG-Dienstprogramm (über den Befehl REORG TABLE) eine Tabelle in einem der temporären Systemtabellenbereiche. Diese temporären Tabellenbereiche können definiert werden, um andere Behälter als die für die benutzerdefinierten Tabellen angegebenen Behälter zu verwenden.

Anmerkungen:

1. Nehmen Sie *auf keinen Fall* direkte Änderungen an diesen Dateien vor. Der Zugriff auf diese Dateien kann nur indirekt über die dokumentierten APIs und mit Tools erfolgen, die diese APIs implementieren (einschließlich des Befehlszeilenprozessors und der Steuerzentrale).
2. Verschieben Sie diese Dateien nicht.
3. Löschen Sie diese Dateien nicht.
4. Das Sichern einer Datenbank oder eines Tabellenbereichs über die API **sqlubkp** (Backup Database), einschließlich der Implementierungen dieser API durch den Befehlszeilenprozessor und die Steuerzentrale, ist die einzige Sicherungsmethode, die unterstützt wird.

DMS-Tabellenbereich

In einem DMS-Tabellenbereich (DMS - Database Managed Space) steuert der Datenbankmanager den Speicherbereich. Das Speichermodell besteht aus einer begrenzten Anzahl von Einheiten, deren Speicherbereich von DB2 verwaltet wird. Der Administrator entscheidet, welche Einheiten zu verwenden sind, und DB2 verwaltet den Speicherbereich auf diesen Einheiten. Der Tabellenbereich ist im wesentlichen eine Implementierung eines Dateisystems, das einem bestimmten Zweck dient und entworfen wurde, um die Anforderungen des Datenbankmanagers optimal zu erfüllen. Zur Tabellenbereichsdefinition gehört eine Liste der Einheiten oder Dateien, die zum Tabellenbereich gehören und in denen Daten gespeichert werden können.

Ein DMS-Tabellenbereich, der benutzerdefinierte Tabellen und Daten enthält, kann wie folgt definiert werden:

- Als *regulärer* Tabellenbereich zum Speichern normaler Tabellen- und Indexdaten
- Als *langer* Tabellenbereich (LONG) zum Speichern von Langfeld- oder LOB-Daten

Beachten Sie beim Entwerfen Ihrer DMS-Tabellenbereiche und Behälter folgendes:

- Der Datenbankmanager arbeitet mit einheitenübergreifendem Lesen und Schreiben von Daten (Striping), um eine gleichmäßige Verteilung von Daten auf alle Behälter sicherzustellen.
- Die Maximalgröße regulärer Tabellenbereiche beträgt 64 GB für 4-KB-Seiten, 128 GB für 8-KB-Seiten, 256 GB für 16-KB-Seiten und 512 GB für 32-KB-Seiten. Die Maximalgröße von Tabellenbereichen für lange Objektdateien beträgt 2 TB.
- Im Unterschied zu SMS-Tabellenbereichen müssen die Behälter, die einen DMS-Tabellenbereich bilden, nicht die gleiche Größe haben. Dies wird im Normalfall jedoch nicht empfohlen, da es zu ungleichmäßiger Verteilung (Striping) auf die Behälter und nicht optimaler Leistung führt. Wenn ein Behälter voll ist, wird in DMS-Tabellenbereichen jeder verfügbare freie Speicherbereich anderer Behälter genutzt.
- Da der Speicherbereich vorab zugeordnet wird, muß er zur Verfügung stehen, bevor der Tabellenbereich erstellt werden kann. Bei Verwendung von Einheitenbehältern muß die Einheit ebenfalls mit genügend Speicherbereich für die Definition des Behälters verfügbar sein. Auf jeder Einheit kann nur ein Behälter definiert werden. Um eine Verschwendung von Speicherbereich zu vermeiden, sollten die Größe der Einheit und die Größe des Behälters äquivalent sein. Wenn z. B. die Einheit mit 5 000 Seiten zugeordnet ist, und der Einheitenbehälter zum Zuordnen von 3 000 Seiten definiert ist, sind 2 000 Seiten der Einheit nicht verwendbar.
- Eine Seite in jedem Behälter ist für den Systemaufwand reserviert, und von den verbleibenden Seiten wird jeweils ein durch EXTENTSIZE definierter Speicherbereich gleichzeitig verwendet werden. Nur ganze, durch EXTENTSIZE definierte Speicherbereiche werden verwendet. Für eine optimale Speicherverwaltung können Sie daher beim Zuordnen eines Behälters eine geeignete Größe anhand der folgenden Formel bestimmen:

$$(\text{extent_size} * n) + 1$$

Dabei ist *extent_size* die Größe jedes durch EXTENTSIZE definierten Speicherbereichs im Tabellenbereich, und *n* ist die Anzahl dieser Speicherbereiche, die Sie in dem Behälter speichern wollen.

- Drei EXTENTSIZE große Speicherbereiche im Tabellenbereich sind für den Systemaufwand reserviert.
- Mindestens zwei EXTENTSIZE große Speicherbereiche sind erforderlich, um Tabellendaten des Benutzers zu speichern. (Diese zwei Speicherbereiche

sind für die regulären Daten einer Tabelle vorgesehen, und nicht für Index-, Langfeld- oder LOB-Daten, die eigene Speicherbereiche benötigen.)

- Einheitenbehälter müssen logische Datenträger mit einer „zeichenspezifischen Schnittstelle“ (d. h. keine physischen Datenträger) verwenden.
- Für DMS-Tabellenbereiche können auch Dateien anstelle von Einheiten (devices) verwendet werden. Zwischen einer Datei und einer Einheit gibt es keine betriebsbedingten Unterschiede. Jedoch kann eine Datei wegen der Laufzeit mit dem Dateisystem verbundenen Systemaufwands weniger effizient sein. Dateien sind in folgenden Situationen nützlich:
 - Wenn Einheiten nicht direkt unterstützt werden.
 - Wenn eine Einheit nicht verfügbar ist.
 - Wenn die Maximalleistung nicht erforderlich ist.
 - Wenn Sie keine Einheiten einrichten wollen.
- Wenn LOB- oder LONG VARCHAR-Daten verarbeitet werden, kann die Verwendung des Dateisystem-Cache Leistungsvorteile erbringen. Beachten Sie, daß LOB- und LONG VARCHAR-Daten nicht vom DB2-Pufferpool zwischengespeichert (gepuffert) werden.
- Einige Betriebssysteme erlauben den Betrieb physischer Einheiten, die größer als 2 GB sind. Sie sollten in Betracht ziehen, die physische Einheit in logische Einheiten zu partitionieren, damit kein Behälter größer als die vom Betriebssystem zugelassene Größe ist.

Hinzufügen von Behältern in DMS-Tabellenbereichen

Mit der Anweisung ALTER TABLESPACE können Sie einem bereits vorhandenen Tabellenbereich einen Behälter hinzufügen, um seine Speicherkapazität zu erhöhen. Der Inhalt des Tabellenbereichs wird dann über alle Behälter neu verteilt. Während dieser Neuverteilung wird der Zugriff auf den Tabellenbereich nicht eingeschränkt. Wenn mehr als ein Behälter hinzugefügt werden muß, sollten diese Behälter gleichzeitig in einer Anweisung ALTER TABLESPACE bzw. innerhalb derselben Transaktion hinzugefügt werden, um zu vermeiden, daß der Datenbankmanager den Inhalt der Behälter mehr als einmal neu verteilen muß.

Sie sollten die Auslastung der Behälter für einen Tabellenbereich mit Hilfe des Befehl LIST TABLESPACE CONTAINERS oder LIST TABLESPACES überprüfen. Neue Behälter sollten hinzugefügt werden, bevor die vorhandenen Behälter fast oder ganz voll sind. Der neue Speicherbereich aller Behälter wird erst nach der Neuverteilung der Daten verfügbar.

Durch das Hinzufügen eines Behälters, der kleiner als vorhandene Behälter ist, wird eine ungleichmäßige Verteilung der Daten verursacht. Dies kann dazu führen, daß parallele Operationen wie das Vorablesen von Daten weniger effizient arbeiten, als sie es auf Behältern gleicher Größe könnten.

Überlegungen zum Tabellenbereichsentwurf

Dieser Abschnitt behandelt die folgenden Themen:

- „Überlegungen zur Ein-/Ausgabe (E/A) von Tabellenbereichen“
- „Zuordnen von Tabellenbereichen zu Pufferpools“ auf Seite 166
- „Zuordnen von Tabellenbereichen zu Knotengruppen“ auf Seite 167
- „Zuordnen von Tabellen zu Tabellenbereichen“ auf Seite 168
- „Auswählen eines Werts für EXTENTSIZE“ auf Seite 169
- „Empfehlungen für temporäre Tabellenbereiche“ auf Seite 170
- „Empfehlungen für Katalogtabellenbereiche“ auf Seite 172
- „Überlegungen zur Art der Auslastung“ auf Seite 173
- „Auswählen eines SMS- oder DMS-Tabellenbereichs“ auf Seite 174
- „Optimieren von Leistung, wenn Daten auf RAID-Einheiten plaziert werden“ auf Seite 176

Überlegungen zur Ein-/Ausgabe (E/A) von Tabellenbereichen

Die Art und der Aufbau Ihres Tabellenbereichs bestimmen die Effizienz der Ein-/Ausgabeoperationen, die mit diesem Tabellenbereich erzielt werden kann. Es folgen einige Begriffe, mit denen Sie vertraut sein sollten, bevor Sie mit den weiteren Themen über Aufbau und Verwendung von Tabellenbereichen fortfahren:

Lesen großer Blöcke (Big Blocks)

Eine Leseoperation, bei der mehrere Seiten (in der Regel ein durch EXTENTSIZE definierter Bereich) in einer einzigen Anforderung abgerufen werden. Das Lesen mehrerer Seiten in einem Vorgang ist effizienter als das Lesen jeder Seite in getrennten Vorgängen.

Vorablesezugriff

Das Vorablesen der Seiten, auf die in einer Abfrage zugegriffen wird. Der Hauptzweck des Vorablesens ist die Verringerung von Antwortzeiten. Dies kann erreicht werden, wenn das Vorablesen von Seiten asynchron zur Ausführung der Abfrage stattfinden kann. Die besten Antwortzeiten werden erzielt, wenn entweder die CPU(s) oder das E/A-Subsystem mit maximaler Kapazität arbeiten.

Seitenlöschfunktion

Durch das Lesen und Ändern von Seiten sammeln diese sich im Pufferpool der Datenbank an. Wenn eine Datenseite eingelesen wird, wird sie in eine Seite des Pufferpools eingelesen. Wenn der Pufferpool ganz mit geänderten Seiten gefüllt ist, muß eine dieser geänderten Seiten auf die Platte geschrieben werden, bevor die neue Seite eingelesen werden kann. Bevor

nun der Pufferpool gänzlich gefüllt wird, treten Seitenlöschagenten in Aktion, die geänderte Seiten auf die Platte schreiben und im Pufferpool löschen, um die Verfügbarkeit von Pufferpoolseiten für zukünftige Leseanforderungen sicherzustellen.

Immer wenn DB2 das Lesen großer Blöcke (Big Blocks) als vorteilhaft erkennt, werden große Blöcke gelesen. Dies tritt in der Regel beim Abrufen von Daten, die sequentieller oder teilweise sequentieller Art sind. Die Menge der Daten, die in einer Leseoperation gelesen werden, hängt von der Größe des mit `EXTENTSIZE` definierten Bereichs ab. Je größer der Wert für `EXTENTSIZE` ist, desto mehr Seiten können in einem Vorgang gelesen werden.

Die Art, wie der Bereich auf der Platte gespeichert ist, hat Einfluß auf die E/A-Effizienz. In einem DMS-Tabellenbereich mit Einheitenbehältern werden die Daten eher zusammenhängend auf der Platte gespeichert und können mit minimaler Suchzeit und Plattenlatenzzeit gelesen werden. Wenn jedoch Dateien verwendet werden, können die Daten vom Dateisystem in Teile getrennt an mehr als einer Position auf der Platte gespeichert werden. Dies geschieht häufiger bei Verwendung von SMS-Tabellenbereichen, bei denen Dateien um jeweils eine Seite erweitert werden, wodurch die Fragmentierung wahrscheinlicher wird. Eine große Datei, die zur Verwendung durch einen DMS-Tabellenbereich vorab zugeordnet wurde, führt eher dazu, daß die Datei auf der Platte zusammenhängend gespeichert wird, insbesondere wenn die Datei in einem noch ungenutzten Speicherbereich zugeordnet wurde.

Sie können den Grad des Vorablesens durch Optimieren des Parameters `PREFETCHSIZE` in der Anweisung `CREATE TABLESPACE` steuern. (Der Standardwert für alle Tabellenbereiche in der Datenbank wird durch den Konfigurationsparameter `dft_prefetch_sz` der Datenbank festgelegt.) Der Parameter `PREFETCHSIZE` gibt DB2 an, wie viele Seiten zu lesen sind, wenn ein Vorableszugriff ausgelöst wird. Wenn der Wert des Parameters `PREFETCHSIZE` auf ein Vielfaches des Parameters `EXTENTSIZE` in der Anweisung `CREATE TABLESPACE` gesetzt wird, können mehrere durch `EXTENTSIZE` definierte Bereiche parallel gelesen werden. (Der Standardwert für alle Tabellenbereiche in der Datenbank wird durch den Konfigurationsparameter `dft_extent_sz` der Datenbank festgelegt.) Der Parameter `EXTENTSIZE` gibt die Anzahl der 4-KB-Seiten an, die in einen Behälter geschrieben werden, bevor zum nächsten Behälter übergegangen wird.

Nehmen Sie zum Beispiel an, Sie hätten einen Tabellenbereich, der drei Einheiten verwendet. Wenn Sie den Wert für `PREFETCHSIZE` auf das Dreifache des Werts für `EXTENTSIZE` setzen, kann DB2 einen Lesezugriff in großen Blöcken von jeder Einheit parallel durchführen, wodurch der E/A-Durchsatz erheblich erhöht wird. Voraussetzungen sind, daß jede Einheit eine getrennte physische Einheit ist und daß der Controller über eine ausreichende Band-

breite verfügt, um den Datenstrom von jeder Einheit zu verarbeiten. Beachten Sie, daß DB2 eventuell die Parameter für den Vorablesezugriff zur Laufzeit aufgrund der Abfragegeschwindigkeit, der Pufferpoolauslastung und anderer Faktoren dynamisch anpassen muß.

Einige Dateisysteme (z. B. Journaled File System unter AIX) verfügen über eine eigene Vorablesemethode. In einigen Fällen kann der Vorablesezugriff des Dateisystems auf größere Datenmengen eingestellt sein als der Vorablesezugriff von DB2. Dies kann dazu führen, daß der Vorablesezugriff für SMS- und DMS-Tabellenbereiche mit Dateibehältern scheinbar eine bessere Leistung zeigt als der Vorablesezugriff für DMS-Tabellenbereiche mit Einheitenbehältern. Dies ist jedoch irreführend, da es sich wahrscheinlich um das Ergebnis einer zusätzlichen Ebene des Vorablesezugriffs handelt, die innerhalb des Dateisystems wirksam ist. Normalerweise sollten DMS-Tabellenbereiche jeder äquivalenten Konfiguration im Hinblick auf die Leistung überlegen sein.

Für ein effizientes Vorablesen (oder auch nur Lesen) muß eine ausreichende Anzahl verfügbarer Pufferpoolseiten vorhanden sein. Zum Beispiel könnte es eine Anforderung zum parallelen Vorablesezugriff geben, mit dem drei (durch EXTENTSIZE bestimmte) Bereiche aus einem Tabellenbereich gelesen werden und durch den für jede einzulesende Seite eine geänderte Seite aus dem Pufferpool herausgeschrieben wird. Die Anforderung zum Vorablesezugriff könnte so weit verlangsamt werden, daß sie mit der Abfrage nicht Schritt halten kann. Daher sollten Seitenlöschfunktionen in ausreichender Anzahl konfiguriert werden, um die Anforderungen von Vorablesezugriffen erfüllen zu können. Für jede reale Platte, die von der Datenbank verwendet wird, sollte mindestens eine Seitenlöschfunktion definiert werden. Weitere Informationen zu diesen Themen finden Sie im Handbuch *Systemverwaltung: Optimierung*.

Zuordnen von Tabellenbereichen zu Pufferpools

Jeder Tabellenbereich wird einem bestimmten Pufferpool zugeordnet. Der Standardpufferpool ist IBMDEFAULTBP. Wenn ein anderer Pufferpool einem Tabellenbereich zugeordnet werden soll, muß der Pufferpool existieren (er wird mit der Anweisung CREATE BUFFERPOOL definiert). Die Zuordnung wird bei der Erstellung des Tabellenbereichs (mit der Anweisung CREATE TABLESPACE) definiert. Die Zuordnung zwischen dem Tabellenbereich und dem Pufferpool kann mit der Anweisung ALTER TABLESPACE geändert werden.

Durch die Verwendung mehr als eines Pufferpools haben Sie die Möglichkeit, die Verwendung von Hauptspeicher durch die Datenbank zu konfigurieren, um die allgemeine Leistung zu verbessern. Bei Tabellenbereichen mit einer oder mehreren umfangreichen Tabellen, auf die die Benutzer wahlfrei zugreifen, kann die Größe des Pufferpools begrenzt werden, da ein Zwischenspeichern (Caching) der Datenseiten vielleicht keine Vorteile bietet. Dem

Tabellenbereich für eine Online-Transaktionsanwendung kann ein größerer Pufferpool zugeordnet werden, so daß die Datenseiten, die von der Anwendung genutzt werden, länger im Cache zwischengespeichert und so schnellere Antwortzeiten erzielt werden können. Vorsicht ist aber bei der Konfiguration neuer Pufferpools geboten. Weitere Informationen zu diesem Thema finden Sie im Abschnitt zum Verwalten des Datenbankpufferpools im Handbuch *Systemverwaltung: Optimierung*.

Anmerkung: Wenn Sie ermittelt haben, daß für Ihre Datenbank eine Seitengröße von 8 KB, 16 KB oder 32 KB erforderlich ist, muß jeder Tabellenbereich mit einer dieser Seitengrößen einem Pufferpool mit derselben Seitengröße zugeordnet werden.

Der Speicher, der für alle Pufferpools benötigt wird, muß dem Datenbankmanager bereits beim Starten der Datenbank zur Verfügung stehen. Wenn DB2 den erforderlichen Speicherbereich nicht erhalten kann, startet der Datenbankmanager mit den Standardpufferpools (je einer mit 4-KB-Seiten, 8-KB-Seiten, 16-KB-Seiten und 32-KB-Seiten) und gibt eine Warnung aus.

In einer Umgebung mit partitionierten Datenbanken können Sie einen Pufferpool mit derselben Größe für alle Partitionen in der Datenbank erstellen. Sie können auch Pufferpools verschiedener Größen in verschiedenen Partitionen erstellen. Weitere Informationen zur Anweisung CREATE BUFFERPOOL finden Sie im Handbuch *SQL Reference*.

Zuordnen von Tabellenbereichen zu Knotengruppen

In einer Umgebung mit partitionierten Datenbanken wird jeder Tabellenbereich einer bestimmten Knotengruppe zugeordnet. Dadurch können die Merkmale des Tabellenbereichs auf jeden Knoten in der Knotengruppe übertragen werden. Die Knotengruppe muß vorhanden sein (sie wird mit der Anweisung CREATE NODEGROUP definiert). Die Zuordnung zwischen dem Tabellenbereich und der Knotengruppe wird bei der Erstellung des Tabellenbereichs mit der Anweisung CREATE TABLESPACE definiert.

Die Zuordnung zwischen dem Tabellenbereich und der Knotengruppe kann mit der Anweisung ALTER TABLESPACE nicht geändert werden. Sie können lediglich die Spezifikationen des Tabellenbereichs für einzelne Partitionen innerhalb der Knotengruppe ändern. In einer Umgebung mit einer Einzelpartition wird jeder Tabellenbereich der Standardknotengruppe zugeordnet. Die Standardknotengruppe bei der Definition eines Tabellenbereichs ist IBMDEFAULTGROUP, sofern kein temporärer Systemtabellenbereich definiert wird. In diesem Fall wird die Standardknotengruppe IBMTEMPGROUP verwendet. Weitere Informationen zur Anweisung CREATE NODEGROUP finden Sie im Handbuch *SQL Reference*. Weitere Informationen zu Knotengruppen und zum physischen Datenbankentwurf finden Sie in „Entwerfen von Knotengruppen“ auf Seite 144.

Zuordnen von Tabellen zu Tabellenbereichen

Bei der Festlegung, wie Tabellen Tabellenbereichen zugeordnet werden sollen, sollten Sie folgendes berücksichtigen:

- Die Partitionierung Ihrer Tabellen

Sie sollten mindestens sicherstellen, daß der Tabellenbereich, den sie auswählen, in einer Knotengruppe mit der gewünschten Partitionierung ist.

- Menge der Daten in der Tabelle

Wenn Sie planen, viele kleine Tabellen in einem Tabellenbereich zu speichern, sollten Sie dazu die Verwendung eines SMS-Tabellenbereichs in Betracht ziehen. Die Vorteile von DMS-Tabellenbereichen im Hinblick auf die Effizienz bei E/A-Operationen und Speicherbereichsverwaltung sind bei kleinen Tabellen nicht so bedeutsam. Die Vorteile von SMS-Tabellenbereichen hinsichtlich der Speicherzuordnung von einer Seite gleichzeitig und nur bei Bedarf sind bei kleinen Tabellen attraktiver. Wenn eine Ihrer Tabellen größer ist oder Sie einen schnelleren Zugriff auf die Daten in den Tabellen benötigen, sollte ein DMS-Tabellenbereich mit einem kleinen Wert für `EXTENTSIZE` in Betracht gezogen werden.

Eventuell empfiehlt es sich, einen separaten Tabellenbereich für jede umfangreiche Tabelle zu verwenden und kleine Tabellen gemeinsam in einem einzigen Tabellenbereich anzulegen. Diese Trennung ermöglicht Ihnen, anhand der Auslastung des Tabellenbereichs einen geeigneten Wert für den Parameter `EXTENTSIZE` auszuwählen. (Weitere Informationen hierzu finden Sie in „Auswählen eines Werts für `EXTENTSIZE`“ auf Seite 169.)

- Typ der Daten in der Tabelle

Sie könnten beispielsweise über Tabellen mit alten Daten verfügen, die relativ selten verwendet werden und bei denen der Endbenutzer eine längere Antwortzeit für Abfragen, die diese Daten betreffen, vielleicht akzeptiert. In diesem Fall könnten Sie für diese „historischen“ Tabellen einen anderen Tabellenbereich verwenden und diesem Tabellenbereich kostensparendere physische Einheiten mit einer langsameren Zugriffsgeschwindigkeit zuordnen.

Auf der anderen Seite können Sie einige wichtige Tabellen identifizieren, für die eine hohe Verfügbarkeit und schnelle Antwortzeiten erforderlich sind. Diese Tabellen könnten Sie in einen Tabellenbereich stellen, der einer schnellen physischen Einheit zugeordnet ist, die die Anforderungen für diese wichtigen Daten besser erfüllt.

Wenn Sie mit DMS-Tabellenbereichen arbeiten, können Sie Ihre Tabellendaten auf drei verschiedene Tabellenbereiche verteilen: einen für Indexdaten, einen für LOB- und Langfelddaten sowie einen für reguläre Tabellendaten. Auf diese Weise können Sie die Tabellenbereichsmerkmale und die physischen Einheiten der Tabellenbereiche auswählen, die für die Daten am besten geeignet sind. Sie könnten z. B. die Indexdaten auf die schnellste verfügbare Einheit stellen und dadurch eine erhebliche Verbesserung der

Leistung erzielen. Wenn Sie eine Tabelle auf DMS-Tabellenbereiche aufteilen, sollten Sie in Betracht ziehen, alle Teile der Tabelle zusammen zu sichern und wiederherzustellen, wenn die aktualisierende Wiederherstellung (Rollforward) aktiviert ist. SMS-Tabellenbereiche unterstützen diese Art der Datenverteilung auf Tabellenbereiche nicht.

- **Verwaltungserfordernisse**

Einige Verwaltungsfunktionen können auf Tabellenbereichsebene anstatt auf Datenbank- oder Tabellenebene ausgeführt werden. Wenn Sie beispielsweise eine Sicherungskopie eines Tabellenbereichs anstelle der Sicherungskopie einer Datenbank erstellen, können Sie Ihre Zeit und Ressourcen besser ausnutzen. Diese Vorgehensweise ermöglicht Ihnen, Tabellenbereiche mit umfangreichen Änderungen häufig zu sichern und von den Tabellenbereichen, die wenig geändert werden, nur gelegentlich neue Sicherungskopien anzulegen.

Sie können eine Datenbank oder einen Tabellenbereich wiederherstellen. Wenn Tabellen, die sich nicht aufeinander beziehen, keine gemeinsamen Tabellenbereiche benutzen, haben Sie die Option, kleinere Teile Ihrer Datenbank wiederherzustellen und den Aufwand verringern.

Ein gutes Verfahren besteht darin, Tabellen, die voneinander abhängig sind, in einer Gruppe von Tabellenbereichen zusammenzufassen. Diese Tabellen könnten über referentielle Integritätsbedingungen oder andere definierte Geschäftsregeln voneinander abhängig sein.

Falls Sie eine bestimmte Tabelle häufig löschen und erneut definieren müssen, können Sie die Tabelle in einem eigenen Tabellenbereich definieren, weil das Löschen eines DMS-Tabellenbereichs effizienter ist als das Löschen einer Tabelle.

Auswählen eines Werts für EXTENTSIZE

Der Wert des Parameters EXTENTSIZE für einen Tabellenbereich ist die Anzahl der Seiten mit Tabellendaten, die in einen Behälter geschrieben werden, bevor Daten in den nächsten Behälter geschrieben werden. Beim Auswählen eines Werts für EXTENTSIZE ist folgendes zu beachten:

- **Größe und Art der Tabellen im Tabellenbereich**

In DMS-Tabellenbereichen wird einer Tabelle gleichzeitig jeweils ein durch EXTENTSIZE definierter Speicherbereich zugeordnet. Wenn beim Füllen der Tabelle mit Daten ein durch EXTENTSIZE definierter Bereich voll ist, wird ein neuer Bereich dieser Größe zugeordnet.

Eine Tabelle besteht aus folgenden separaten Tabellenobjekten:

- Ein Datenobjekt. Hier werden die regulären Spaltendaten gespeichert.
- Ein Indexobjekt. Hier werden alle für die Tabelle definierten Indizes gespeichert.
- Ein Langfeldobjekt. Hier werden Langfelddaten gespeichert, wenn die Tabelle eine oder mehrere Spalten mit LONG-Datentypen besitzt.

- Zwei LOB-Objekte. Wenn die Tabelle eine oder mehrere LOB-Spalten hat, werden diese in folgenden beiden Tabellenobjekten gespeichert:
 - Ein Tabellenobjekt für die LOB-Daten
 - Ein zweites Tabellenobjekt für Metadaten, die die LOB-Daten beschreiben

Jedes Tabellenobjekt wird getrennt gespeichert und ordnet nach Bedarf neue (durch EXTENTSIZE definierte) Bereiche zu. Jedes Tabellenobjekt wird außerdem mit einem Metadatenobjekt verbunden, das als *Speicherbereichsmaske* bezeichnet wird und alle durch EXTENTSIZE definierten Bereiche im Tabellenbereich beschreibt, die zu dem Tabellenobjekt gehören. Der Speicherbereich für Speicherbereichsmasken wird ebenfalls jeweils in der Größe von EXTENTSIZE zugeordnet.

Die Erstzuordnung von Speicherbereich für eine Tabelle umfaßt daher zwei EXTENTSIZE-Bereiche für jedes Tabellenobjekt. Wenn Sie viele kleine Tabellen in einem Tabellenbereich haben, ist es möglich, daß eine relativ große Menge an Speicher für eine relativ kleine Menge von Daten zugeordnet ist. In einem solchen Fall sollten Sie einen kleinen Wert für EXTENTSIZE definieren oder einen SMS-Tabellenbereich verwenden, in dem jeweils eine Seite gleichzeitig zugeordnet wird.

Wenn Sie andererseits über eine umfangreiche Tabelle mit einer hohen Zuwachsrate verfügen und einen DMS-Tabellenbereich mit einem niedrigen Wert für EXTENTSIZE verwenden, könnte durch häufiges Zuordnen zusätzlichen Speicherbereichs unnötiger Systemaufwand entstehen.

- Art des Zugriffs auf die Tabellen
Wenn auf die Tabellen durch zahlreiche Abfragen oder Transaktionen zugegriffen wird, die große Datenmengen verarbeiten, kann das Vorablesen von Daten aus den Tabellen eine wesentliche Leistungsverbesserung bewirken. (Das Handbuch *Systemverwaltung: Optimierung* enthält Informationen zum Vorablesen von Daten sowie zur Beziehung zwischen dem Vorablesezugriff und dem Wert für EXTENTSIZE.)
- Minimal erforderliche Anzahl mit EXTENTSIZE definierter Speicherbereiche
Falls nicht genügend Platz für fünf durch EXTENTSIZE definierte Speicherbereiche des Tabellenbereichs in den Behältern vorhanden ist, wird der Tabellenbereich nicht erstellt.

Empfehlungen für temporäre Tabellenbereiche

Es wird empfohlen, einen einzelnen temporären SMS-Tabellenbereich zu definieren, dessen Seitengröße der Seitengröße entspricht, die in den meisten regulären Tabellenbereichen verwendet wird. Dies sollte für typische Umgebungen und Auslastungen geeignet sein. Es kann jedoch vorteilhaft sein, mit unterschiedlichen Konfigurationen für temporäre Tabellenbereiche und Auslastungen zu experimentieren. Sie sollten die folgenden Punkte beachten:

- Auf temporäre Tabellen wird meist gruppenweise und sequentiell zugegriffen. Das heißt, eine Gruppe von Zeilen wird eingefügt oder eine Gruppe sequentieller Zeilen wird abgerufen. Daher führt eine größere Seitengröße in der Regel zu einer besseren Leistung, weil weniger E/A-Anforderungen logischer oder physischer Seiten erforderlich sind, um eine bestimmte Datenmenge einzulesen. Dies ist nicht immer der Fall, wenn die durchschnittliche Zeilengröße einer temporären Tabelle kleiner ist als die Seitengröße dividiert durch 255. Ungeachtet der Seitengröße können maximal 255 Zeilen auf einer Seite vorhanden sein. Eine Abfrage, die z. B. eine temporäre Tabelle mit 15-Byte-Zeilen erfordert, könnte von einem temporären Tabellenbereich mit einer Seitengröße von 4 KB besser bedient werden, da alle 255 solcher Zeilen in eine 4-KB-Seite aufgenommen werden können. Eine 8-KB-Seite (oder größere Seite) ergäbe mindestens 4 KB (oder mehr) Byte ungenutzten Speicherbereich auf jeder Seite der temporären Tabelle und würde die Anzahl der erforderlichen E/A-Anforderungen nicht reduzieren.
- Wenn über fünfzig Prozent der regulären Tabellenbereiche in der Datenbank dieselbe Seitengröße verwenden, kann es vorteilhaft sein, die temporären Tabellenbereiche mit derselben Seitengröße zu definieren. Der Vorteil liegt darin, daß der temporäre Tabellenbereich mit dieser Konfiguration denselben Pufferpoolspeicherbereich mit den meisten oder allen regulären Tabellenbereichen gemeinsam benutzen kann. Dadurch wird die Optimierung des Pufferpools wiederum vereinfacht.
- Wenn Sie mit Hilfe eines temporären Tabellenbereichs eine Tabelle reorganisieren, muß die Seitengröße des temporären Tabellenbereichs mit der Seitengröße der Tabelle übereinstimmen. Deshalb sollten Sie sicherstellen, daß temporäre Tabellenbereiche vorhanden sind, die für jede Seitengröße definiert sind, die von vorhandenen Tabellen verwendet wird, die Sie vielleicht mit Hilfe eines temporären Tabellenbereichs reorganisieren.

Sie können eine Reorganisation auch ohne temporären Tabellenbereich ausführen, indem Sie die Tabelle „am Ort“ reorganisieren, das heißt, direkt im Zieltabellenbereich. Natürlich setzt eine Reorganisation „am Ort“ voraus, daß im Zieltabellenbereich zusätzlicher Speicherbereich für den Reorganisationsprozeß vorhanden ist. Weitere Informationen zur Reorganisation von Tabellen finden Sie im Handbuch *Systemverwaltung: Optimierung*.

- Im allgemeinen, wenn temporäre Tabellenbereiche mit unterschiedlichen Seitengrößen vorhanden sind, wählt das Optimierungsprogramm am häufigsten den temporären Tabellenbereich mit dem größten Pufferpool aus. In solchen Fällen empfiehlt es sich, einem der temporären Tabellenbereiche einen großen Pufferpool und den übrigen einen kleineren Pufferpool zuzuordnen. Eine solche Pufferpoolzuordnung hilft bei der Sicherstellung einer effizienten Auslastung des Hauptspeichers. Wenn z. B. Ihr Katalogtabellenbereich 4-KB-Seiten verwendet und die übrigen Tabellenbereiche 8-KB-Seiten, kann sich folgende Konfiguration für temporäre Tabellenbereiche am

besten eignen: ein einzelner temporärer 8-KB-Tabellenbereich mit einem großen Pufferpool und ein einzelner 4-KB-Tabellenbereich mit einem kleinen Pufferpool.

Anmerkung: Katalogtabellenbereiche sind auf die Verwendung von 4-KB-Seiten beschränkt. Daher erzwingt der Datenbankmanager immer das Vorhandensein eines temporären 4-KB-Tabellenbereichs, um die Reorganisation von Katalogtabellen zu ermöglichen.

- Es bringt im allgemeinen keinen Vorteil, mehr als einen temporären Tabellenbereich von jeder Seitengröße zu definieren.
- Für temporäre Tabellenbereiche sind SMS-Bereiche aus den folgenden Gründen DMS-Bereichen meist vorzuziehen:
 - Plattenspeicherplatz wird im SMS nach Bedarf zugeordnet, wohingegen er im DMS zuvor zugeordnet werden muß. Eine vorherige Zuordnung kann ein Problem darstellen: Temporäre Tabellenbereiche enthalten Übergangsdaten, die einen extremen Höchstspeicherbedarf und einen wesentlich geringeren durchschnittlichen Speicherbedarf besitzen können. Bei DMS muß der Höchstspeicherbedarf vorab zugeordnet werden, wohingegen bei SMS der zusätzliche Plattenspeicherplatz außerhalb der Spitzenlastzeiten für andere Zwecke verwendet werden kann.
 - Der Datenbankmanager versucht, temporäre Tabellenseiten im Speicher zu behalten und sie nicht auf die Platte auszulagern. Im Endeffekt sind die Leistungsvorteile von DMS weniger signifikant.
 - SMS-Behälter können das Zwischenspeichern von Dateisystemen ausnutzen. DMS-Behälter können dies nicht.

Empfehlungen für Katalogtabellenbereiche

Ein SMS-Tabellenbereich empfiehlt sich aus folgenden Gründen für Datenbankkataloge:

- Der Datenbankkatalog besteht aus zahlreichen Tabellen unterschiedlicher Größen. Bei der Verwendung eines DMS-Tabellenbereichs wird mindestens ein Speicherbereich in der Größe von zwei EXTENTSIZE-Bereichen für jedes Tabellenobjekt zugeordnet. Je nachdem, welcher Wert für EXTENTSIZE ausgewählt wurde, kann dies zu einer erheblichen Menge an zugeordnetem, aber ungenutztem Speicher führen. Wenn ein DMS-Tabellenbereich verwendet wird, sollte ein kleiner Wert für EXTENTSIZE (zwei bis vier Seiten) ausgewählt werden. Ansonsten sollte ein SMS-Tabellenbereich verwendet werden.
- In den Katalogtabellen gibt es Spalten mit großen Objekten (LOB-Spalten). LOB-Daten werden nicht mit anderen Daten im Pufferpool behalten, sondern jedesmal, wenn sie benötigt werden, von der Platte gelesen. Das Lesen von LOB-Daten von der Platte verlangsamt die Leistung. 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.

Unter diesen Gesichtspunkten erweist sich ein SMS-Tabellenbereich als die etwas günstigere Wahl für die Katalogtabellen.

Ein anderer Faktor, der zu berücksichtigen wäre, ist die Frage, ob der Katalogtabellenbereich in der Zukunft erweitert werden müßte. Obwohl einige Plattformen die Erweiterung des zugrundeliegenden Speichers für SMS-Behälter unterstützen und die Möglichkeit einer umgeleiteten Wiederherstellung zur Vergrößerung eines SMS-Tabellenbereichs gegeben ist, macht ein DMS-Tabellenbereich das Hinzufügen neuer Behälter einfacher.

Überlegungen zur Art der Auslastung

Der primäre Typ der Auslastung, die von DB2 in Ihrer Umgebung verwaltet wird, kann Ihre Wahl beeinflussen, welcher Typ von Tabellenbereich zu verwenden und welche Seitengröße anzugeben ist. Eine OLTP-Auslastung (Online-Transaktionsverarbeitung) ist durch Transaktionen charakterisiert, die einen wahlfreien Zugriff auf Daten benötigen und die in der Regel nur kleine Datenmengen zurückliefern. In Anbetracht dessen, daß der Zugriff wahlfrei nur auf eine bzw. wenige Seiten erfolgt, ist ein Vorablesezugriff nicht möglich.

In diesem Fall zeigen DMS-Tabellenbereiche mit Einheitenbehältern die beste Leistung. DMS-Tabellenbereiche mit Dateibehältern oder SMS-Tabellenbereiche sind ebenfalls sinnvolle Möglichkeiten für eine OLTP-Auslastung, wenn es nicht auf maximale Leistung ankommt. Wenn nur wenig oder gar keine sequentiellen E/A-Operationen zu erwarten sind, spielen die Werte der Parameter EXTENTSIZE und PREFETCHSIZE in der Anweisung CREATE TABLESPACE keine wichtige Rolle für die E/A-Effizienz.

Eine Abfrageauslastung wird durch Transaktionen charakterisiert, die einen sequentiellen oder teilweise sequentiellen auf Daten benötigen und in der Regel große Datenmengen zurückliefern. Ein DMS-Tabellenbereich mit mehreren Einheitenbehältern (wobei sich jeder Behälter auf einer separaten Platte befindet) bietet das größte Potential für einen effizienten parallelen Vorablesezugriff. Der Wert des Parameters PREFETCHSIZE in der Anweisung CREATE TABLESPACE sollte das Produkt aus dem Wert des Parameters EXTENTSIZE multipliziert mit der Anzahl der Einheitenbehälter sein. Dadurch kann DB2 von allen Behältern parallel vorablesen.

Eine sinnvolle Alternative für eine Abfrageauslastung ist die Verwendung von Dateien, wenn das Dateisystem über eine eigene Vorablesefunktion verfügt. Die Dateien können entweder zu DMS-Tabellenbereichen mit Dateibehältern gehören oder Dateien für SMS-Tabellenbereiche sein. Beachten Sie, daß Sie bei

Verwendung von SMS sicherstellen müssen, daß die Verzeichnisbehälter getrennten physischen Datenträgern zugeordnet sind, um E/A-Parallelität zu erreichen.

Das Ziel für eine gemischte Auslastung besteht darin, einzelne E/A-Anforderungen so effizient wie möglich für OLTP-Auslastungen zu machen und die Effizienz paralleler E/A-Operationen für Abfrageauslastungen zu maximieren.

Beim Ermitteln der Seitengröße für einen Tabellenbereich sind folgende Überlegungen zu berücksichtigen:

- Für OLTP-Anwendungen, die wahlfreie Lese- und Schreiboperationen durchführen, ist eine geringere Seitengröße empfehlenswert, weil dabei weniger Pufferspeicher durch unerwünschte Zeilen belegt wird.
- Für DSS-Anwendungen (Decision Support System), die jeweils auf eine große Anzahl aufeinanderfolgender Zeilen zugreifen, sind größere Seiten empfehlenswert, weil dadurch weniger E/A-Anforderungen erforderlich sind, um eine bestimmte Anzahl Zeilen zu lesen. Es gibt jedoch eine Ausnahme von dieser Regel. Diese betrifft den Fall, daß die Zeilengröße kleiner als folgender Wert ist:

$$\text{seitengröße} / 255$$

Bei dieser Größe bleibt auf jeder Seite Speicherbereich ungenutzt (weil jede Seite maximal 255 Zeilen enthalten kann). In diesem Fall ist eine geringere Seitengröße zu empfehlen.

- Durch größere Seiten können Sie möglicherweise die Zahl der Indexstufen reduzieren.
- Größere Seiten unterstützen längere Zeilen.
- Auf 4-KB-Standardseiten sind Tabellen auf 500 Spalten begrenzt, während auf größeren Seiten (8 KB, 16 KB und 32 KB) 1012 Spalten unterstützt werden.
- Die Maximalgröße des Tabellenbereichs ist proportional zur Seitengröße des Tabellenbereichs. Die Begrenzungen sind im Handbuch *SQL Reference* dokumentiert.

Auswählen eines SMS- oder DMS-Tabellenbereichs

Bei der Entscheidung, welcher Tabellenbereichstyp zum Speichern der Daten verwendet werden soll, sollten Sie das Pro und Kontra gut abwägen.

Vorteile eines SMS-Tabellenbereichs:

- Der Speicher wird vom System erst dann zugeordnet, wenn er benötigt wird.
- Beim Erstellen einer Datenbank sind weniger vorbereitende Arbeiten erforderlich, weil Sie keine Behälter vordefinieren müssen.

Vorteile eines DMS-Tabellenbereichs:

- Die Größe eines Tabellenbereichs kann durch Hinzufügen von Behältern erweitert werden (Anweisung ALTER TABLESPACE). Vorhandene Daten werden automatisch auf die neue Gruppe von Behältern verteilt, um die optimale E/A-Effizienz zu erhalten.
- Eine Tabelle kann nach dem Typ der zu speichernden Daten auf mehrere Tabellenbereiche verteilt werden:
 - Langfeld- und LOB-Daten
 - Indizes
 - Reguläre Tabellendaten

Vielleicht sollen die Tabellendaten zur Erhöhung der Leistung oder zur Vergrößerung der für eine Tabelle gespeicherten Datenmenge getrennt gehalten werden. Sie könnten beispielsweise eine Tabelle mit 64 GB für reguläre Tabellendaten, 64 GB für Indexdaten und 2 TB für Langfelddaten anlegen. Bei Verwendung von 8-KB-Seiten können die Tabellendaten und die Indexdaten bis zu 128 GB umfassen. Bei Verwendung von 16-KB-Seiten können sie bis zu 256 GB umfassen. Bei Verwendung von 32-KB-Seiten können die Tabellendaten und die Indexdaten bis zu 512 GB umfassen.

- Es ist möglich, die Speicherposition der Daten auf der Platte zu steuern, sofern das Betriebssystem dies zuläßt.
- Wenn sich alle Tabellendaten in einem einzigen Tabellenbereich befinden, ist der Systemaufwand beim Löschen und erneuten Definieren eines Tabellenbereichs geringer, als dies beim Löschen und erneuten Definieren einer Tabelle der Fall wäre.
- Im allgemeinen gilt, daß sich mit einer sinnvoll organisierten Gruppe von DMS-Tabellenbereichen eine bessere Leistung erzielen läßt als mit SMS-Tabellenbereichen.

Anmerkung: Bei Solaris und PTX (IBM NUMA-Q) wird die Verwendung von DMS-Tabellenbereichen mit unformatierten Einheiten für leistungskritische Auslastungen dringend empfohlen.

Generell lassen sich kleine Datenbanken, die von einem kleinen Personenkreis genutzt werden, am einfachsten mit SMS-Tabellenbereichen verwalten. Andererseits sollten Sie für umfangreiche, wachsende Datenbanken SMS-Tabellenbereiche nur für temporäre Tabellenbereiche sowie für jede Tabelle getrennte DMS-Tabellenbereiche mit mehreren Behältern verwenden. Außerdem ist es wahrscheinlich sinnvoll, Langfelddaten und Indizes in eigenen Tabellenbereichen zu speichern.

Wenn Sie sich entschließen, DMS-Tabellenbereiche mit Einheitenbehältern zu verwenden, müssen Sie bereit sein, Ihre Umgebung zu optimieren und zu

verwalten. Weitere Informationen finden Sie im Abschnitt über die Leistung von DMS-Einheiten im Handbuch *Systemverwaltung: Optimierung*.

Optimieren von Leistung, wenn Daten auf RAID-Einheiten plaziert werden

In diesem Abschnitt wird beschrieben, wie Sie die Leistung optimieren können, wenn Daten auf RAID-Einheiten plaziert werden. In allgemeinen sollten Sie folgende Maßnahmen für jeden Tabellenbereich ergreifen, der eine RAID-Einheit verwendet:

- Definieren Sie einen einzelnen Behälter für den Tabellenbereich (der eine RAID-Einheit verwendet).
- Setzen Sie den Wert für EXTENTSIZE des Tabellenbereichs so, daß er der Größe des einheitenübergreifend gespeicherten RAID-Datenblocks (Stripe Size) oder einem Vielfachen der Größe entspricht.
- Stellen Sie sicher, daß der Wert für PREFETCHSIZE des Tabellenbereichs die folgenden Bedingungen erfüllt:
 - Er entspricht der Größe des einheitenübergreifend gespeicherten RAID-Datenblocks (Stripe Size) multipliziert mit der Anzahl paralleler RAID-Einheiten (oder einem ganzen Vielfachen dieses Produkts).
 - Er ist ein Vielfaches des Werts für EXTENTSIZE.
- Verwenden Sie die Registrierungsvariable DB2_PARALLEL_IO (siehe unten), um die parallele Ein-/Ausgabe für den Tabellenbereich zu aktivieren.
- Verwenden Sie die Registrierungsvariable DB2_STRIPED_CONTAINERS (siehe unten), um sicherzustellen, daß die Grenzen der durch EXTENTSIZE definierten Bereiche im Tabellenbereiche ausgerichtet werden.

DB2_PARALLEL_IO

Wenn Daten aus Tabellenbereichsbehältern gelesen werden oder in diese geschrieben werden, kann DB2 die parallele Ein-/Ausgabe verwenden, falls die Anzahl der Behälter in der Datenbank größer als 1 ist. Es gibt jedoch Fälle, in denen es vorteilhaft wäre, die parallele Ein-/Ausgabe für einzelne Tabellenbereichsbehälter zu aktivieren. Wenn z. B. der Behälter auf einer einzelnen RAID-Einheit erstellt wird, die aus mehr als einer physischen Platte besteht, ist es vielleicht sinnvoll, Aufrufe zum parallelen Lesen und Schreiben absetzen zu können.

Sie können die Registrierungsvariable DB2_PARALLEL_IO verwenden, um die parallele Ein-/Ausgabe für einen Tabellenbereich zu erzwingen, der einen einzelnen Behälter besitzt. Diese Variable kann auf "*" (Stern) gesetzt werden, d. h. sich auf jeden Tabellenbereich beziehen, oder sie kann auf eine Liste von Tabellenbereichs-IDs gesetzt werden, die durch Komma voneinander getrennt sind. Zum Beispiel:

```

db2set DB2_PARALLEL_IO=*      {parallele E/A für alle
                               Tabellenbereiche aktivieren}
db2set DB2_PARALLEL_IO=1,2,4,8 {parallele E/A für die Tabellenbereiche 1, 2, 4
                               und 8 aktivieren}

```

Nach dem Setzen der Registrierungsvariablen muß DB2 gestoppt (**db2stop**) und anschließend erneut gestartet (**db2start**) werden, damit die Änderungen wirksam werden.

DB2_STRIPED_CONTAINERS

Zur Zeit wird beim Erstellen eines DMS-Tabellenbereichsbehälters (Einheit oder Datei) eine Kennung, die eine Seite lang ist, am Anfang des Behälters gespeichert. Die übrigen Seiten sind für die Datenspeicherung durch DB2 verfügbar und werden in Blöcke von der Größe des Werts für EXTENTSIZE gruppiert.

Wenn RAID-Einheiten für Tabellenbereichsbehälter verwendet werden, wird empfohlen, daß der Tabellenbereich mit einem Wert für EXTENTSIZE erstellt wird, der der Größe des einheitenübergreifend gespeicherten RAID-Datenblocks (Stripe Size) oder einem Vielfachen dieser Größe entspricht. Jedoch können die Speicherbereiche aufgrund der Behälterkennung, die eine Seite lang ist, nicht mit den einheitenübergreifend gespeicherten RAID-Datenblöcke ausgerichtet werden, und es kann während einer E/A-Anforderung notwendig werden, auf mehr physische Platten zuzugreifen als optimal wäre.

DMS-Tabellenbereichsbehälter können jetzt so erstellt werden, daß sich die Kennung in ihrem eigenen separaten Speicherbereich befindet. Dadurch wird das oben beschriebene Problem umgangen, aber es ist ein zusätzlicher Speicherbereich in Größe von EXTENTSIZE innerhalb des Behälters für den Systemaufwand erforderlich. Sie müssen die DB2-Registrierungsvariable DB2_STRIPED_CONTAINERS auf "ON" setzen. Anschließend stoppen Sie das Exemplar und starten es erneut, um Behälter auf diese Weise zu erstellen:

```

db2set DB2_STRIPED_CONTAINERS=ON
db2stop
db2start

```

Jeder DMS-Behälter, der mit der Anweisung CREATE TABLESPACE oder ALTER TABLESPACE erstellt wurde, enthält Kennungen, die einen Speicherbereich einnehmen, der einen ganzen EXTENTSIZE-Wert groß ist. Vorhandene Behälter bleiben unverändert.

Setzen Sie die Variable zurück. Anschließend stoppen Sie Ihr Exemplar und starten es erneut, um das Erstellen von Behältern mit diesem Attribut zu stoppen:

```
db2set DB2_STRIPED_CONTAINERS=  
db2stop  
db2start
```

Die Steuerzentrale und der Befehl LIST TABLESPACE CONTAINERS zeigen nicht an, ob ein Behälter als einheitenübergreifend (Striped) erstellt wurde. Sie verwenden die Bezeichnung "Datei" oder "Einheit", abhängig davon, wie der Behälter erstellt wurde. Zum Prüfen, ob ein Behälter als einheitenübergreifend erstellt wurde, können Sie die Option /DTSF von DB2DART verwenden, um Informationen zu Tabellenbereichen und Behältern auszulesen, und sich dann das Typfeld für den fraglichen Behälter ansehen. Sie können mit Hilfe der APIs zum Abfragen von Behältern (**sqlbftcq** und **sqlbtcq**) eine einfache Anwendung erstellen, die den Typ anzeigt.

Überlegungen zum Entwerfen einer zusammengeschlossenen Datenbank

Beim Entwerfen einer zusammengeschlossenen Datenbank müssen Sie die folgenden Aspekte berücksichtigen:

- Platzbedarf
- Prioritätensetzung im Netzwerk

Die Daten, auf die von einer zusammengeschlossenen Datenbank zugegriffen werden kann, sind in der Regel nicht in der Datenbank gespeichert. Verweise auf Datenquellentabellen und -sichten sind innerhalb des Systemkatalogs gespeichert, aber die tatsächlichen Daten befinden sich an der Datenquelle. Als solche kann eine zusammengeschlossene Datenbank eventuell weniger Speicherplatz als eine herkömmliche Datenbank erfordern. Diese allgemeine Regel ist möglicherweise nicht anwendbar, wenn die Abfragen aufgrund von Systemunterschieden bei Sortierfolgen oder fehlenden Funktionen an einer Datenquelle lokal ausgeführt werden müssen. In diesem Fall werden Tabellen für die Verarbeitung in DB2 erstellt.

Da sich die meisten Daten von Systemen zusammengeschlossener Datenbanken in der Regel an einer oder mehreren Datenquellen befinden, die in einem Netzwerk verteilt sind, ziehen Sie in Betracht, die Ressourcen zu ändern, die DB2 und Ihrem Netzwerksystem zugeordnet sind. Sie erzielen möglicherweise Leistungsverbesserungen, wenn Sie für das Netzwerk mit dem DB2-System mehr Ressourcen zuordnen als für den Datenbankmanager selbst.

Kapitel 9. Entwerfen verteilter Datenbanken

Eine Transaktion wird in DB2 allgemein als *Arbeitseinheit* bezeichnet. Eine Arbeitseinheit ist eine wiederherstellbare Folge von Operationen innerhalb eines Anwendungsprozesses. Der Datenbankmanager kann mit ihrer Hilfe sicherstellen, daß sich eine Datenbank in einem konsistenten Status befindet. Jeder Lese- oder Schreibvorgang in der Datenbank erfolgt innerhalb einer Arbeitseinheit.

Eine Banktransaktion könnte beispielsweise darin bestehen, daß eine Geldsumme von einem Sparkonto auf ein Girokonto überwiesen wird. Nachdem die Anwendung einen Betrag vom Sparkonto abgezogen hat, sind die beiden Konten inkonsistent und bleiben es, bis der Betrag auf dem Girokonto gutgeschrieben ist. Wenn *beide* Schritte zu Ende geführt sind, ist ein Konsistenzzustand erreicht. Die Änderungen können festgeschrieben und anderen Anwendungen zur Verfügung gestellt werden.

Eine Arbeitseinheit beginnt, wenn die erste SQL-Anweisung für die Datenbank ausgeführt wird. Die Anwendung muß die Arbeitseinheit beenden, indem sie die Anweisung COMMIT oder ROLLBACK absetzt. Die Anweisung COMMIT schreibt alle Änderungen, die innerhalb der Arbeitseinheit vorgenommen wurden, permanent fest. Die Anweisung ROLLBACK löscht diese Änderungen aus der Datenbank. Wenn bei einer normalen Beendigung der Anwendung keine dieser Anweisungen abgesetzt wird, wird die Arbeitseinheit automatisch festgeschrieben. Wenn die Anwendung inmitten einer Arbeitseinheit abnormal beendet wird, wird die Arbeitseinheit automatisch rückgängig gemacht. Sobald eine Anweisung COMMIT oder ROLLBACK abgesetzt wurde, kann sie nicht mehr gestoppt werden. Bei einigen Multithread-Anwendungen oder Betriebssystemen (z. B. Windows) wird die Arbeitseinheit, wenn eine Anwendung normal, jedoch ohne explizite Ausführung einer dieser Anweisungen beendet wird, automatisch rückgängig (ROLLBACK) gemacht. Es wird daher empfohlen, dafür zu sorgen, daß Anwendungen vollständige Arbeitseinheiten immer explizit mit einer Anweisung COMMIT festschreiben oder mit einer Anweisung ROLLBACK rückgängig machen. Wenn ein Teil einer Arbeitseinheit nicht erfolgreich beendet wird, werden die Aktualisierungen rückgängig gemacht, so daß die beteiligten Tabellen in dem Zustand verbleiben, in dem sie vor Beginn der Transaktion waren. Auf diese Weise wird sichergestellt, daß Anforderungen weder verloren gehen noch mehrfach ausgeführt werden.

Unter den folgenden Themen finden Sie weitere Informationen:

- „Verwenden einer einzelnen Datenbank in einer Transaktion“ auf Seite 180

- „Verwenden mehrerer Datenbanken in einer Transaktion“ auf Seite 181
- „Weitere Überlegungen zur Konfiguration“ auf Seite 187
- „Prozeß der zweiphasigen Festschreibung“ auf Seite 190
- „Beheben von Problemen bei der zweiphasigen Festschreibung“ auf Seite 193

Informationen zur Erstellung von Anwendungen für verteilte Datenbanken finden Sie in den Handbüchern *Application Development Guide* und *CLI Guide and Reference*.

Verwenden einer einzelnen Datenbank in einer Transaktion

Die einfachste Form der Transaktion besteht darin, innerhalb einer Arbeitseinheit lediglich für eine Datenbank Lese- und Schreibvorgänge auszuführen. Diese Art des Datenbankszugriffs wird als *ferne Arbeitseinheit* bezeichnet.

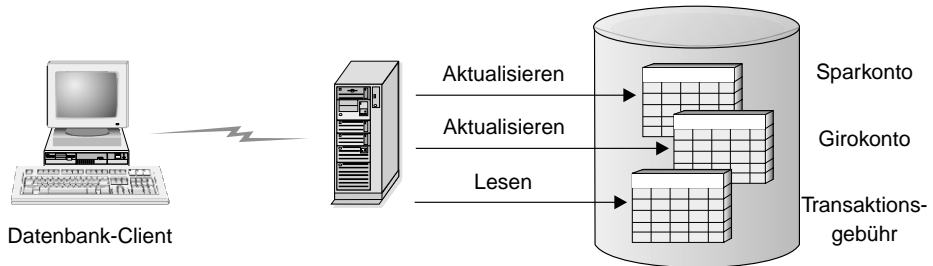


Abbildung 38. Verwenden einer einzelnen Datenbank in einer Transaktion

Abb. 38 zeigt einen Datenbank-Client, der eine Geldtransaktionsanwendung ausführt. Diese Anwendung greift auf eine Datenbank zu, die Tabellen für Giro- und Sparkonten sowie einen Plan für die Transaktionsgebühren enthält. Die Anwendung muß folgende Schritte ausführen:

- Akzeptieren des Betrags, der von der Benutzerschnittstelle überwiesen werden soll
- Subtrahieren des Betrags vom Sparkonto und Ermitteln des neuen Kontostands
- Lesen des Gebührenplans, um die Transaktionsgebühr für ein Sparkonto mit dem angegebenen Kontostand zu bestimmen
- Subtrahieren der Transaktionsgebühr vom Sparkonto
- Gutschreiben des Überweisungsbetrags auf dem Girokonto
- Festschreiben der Transaktion (Arbeitseinheit)

Zur Einrichtung einer solchen Anwendung sind folgende Schritte auszuführen:

1. Erstellen der Tabellen für das Sparkonto, das Girokonto und den Gebührenplan für die Transaktion innerhalb derselben Datenbank (siehe "Implementieren des Datenbankentwurfs" im Handbuch *Systemverwaltung: Implementierung*)
2. Wenn physisch fern, Einrichten des Datenbank-Servers für die Verwendung des geeigneten Übertragungsprotokolls (siehe Beschreibung im Handbuch *DB2 Installation und Konfiguration Ergänzung*)
3. Wenn physisch fern, Katalogisieren des Knotens und der Datenbank, um die Datenbank auf dem Datenbank-Server zu identifizieren (siehe die Beschreibung in den Handbüchern *Einstieg*)
4. Vorkompilieren Ihres Anwendungsprogramms, um eine Verbindung des Typs 1 anzugeben, d. h. Angeben von CONNECT 1 (Standardwert) im Befehl PRECOMPILE PROGRAM (siehe die Beschreibung im Handbuch *Application Development Guide*)

Verwenden mehrerer Datenbanken in einer Transaktion

Bei Verwendung mehrerer Datenbanken in einer einzelnen Transaktion gelten andere Anforderungen für das Einrichten und Verwalten Ihrer Umgebung, je nachdem, wie viele Datenbanken in der Transaktion aktualisiert werden.

Aktualisieren einer einzelnen Datenbank

Wenn Ihre Daten über mehrere Datenbanken verteilt sind, möchten Sie eventuell eine Datenbank aktualisieren, während Sie für eine oder mehrere andere Datenbanken Lesevorgänge ausführen. Diese Art des Zugriffs, die als *Aktualisierung auf mehreren Systemen* bzw. als *zweiphasige Festschreibung* bezeichnet wird, kann innerhalb einer einzelnen Arbeitseinheit (Transaktion) erfolgen. Ein weiteres Beispiel für eine Aktualisierung auf mehreren Systemen finden Sie in „Aktualisieren mehrerer Datenbanken“ auf Seite 183.

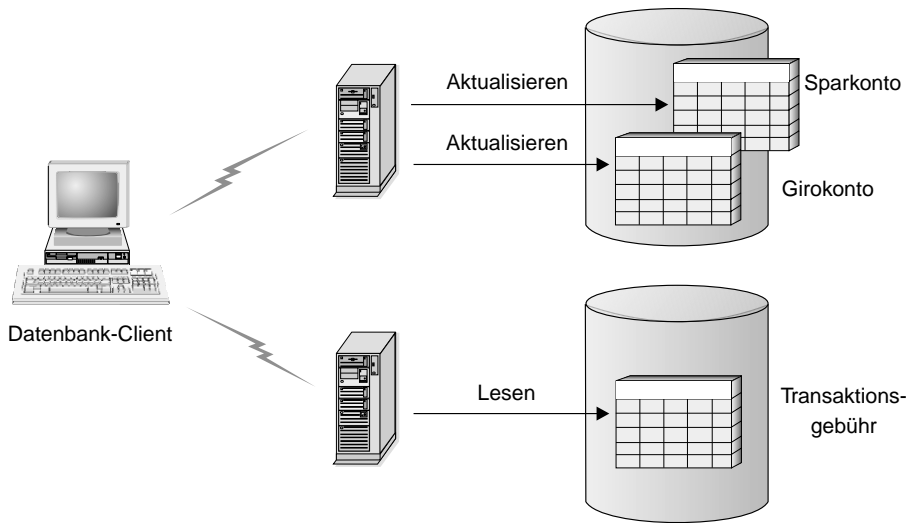


Abbildung 39. Verwenden mehrerer Datenbanken in einer Transaktion

Abb. 39 zeigt einen Datenbank-Client, der eine Geldtransaktionsanwendung ausführt, die auf zwei Datenbank-Server zugreift: einer enthält das Sparkonto und das Girokonto, der andere den Plan für die Transaktionsgebühren. Dieses Beispiel ähnelt dem in Abb. 38 auf Seite 180 gezeigten. Der Unterschied besteht in der Anzahl der Datenbanken und der Speicherposition der Tabellen.

Zur Einrichtung einer Geldtransaktionsanwendung für diese Umgebung sind folgende Schritte auszuführen:

1. Erstellen der erforderlichen Tabellen in den entsprechenden Datenbanken (siehe "Implementieren des Datenbankentwurfs" im Handbuch *Systemverwaltung: Implementierung*)
2. Wenn physisch fern, Einrichten der Datenbank-Server für die Verwendung der geeigneten Übertragungsprotokolle (siehe Beschreibung im Handbuch *DB2 Installation und Konfiguration Ergänzung*)
3. Wenn physisch fern, Katalogisieren der Knoten und der Datenbanken, um die Datenbanken auf den Datenbank-Servern zu identifizieren (siehe die Beschreibung in den Handbüchern *Einstieg*)
4. Vorkompilieren Ihres Anwendungsprogramms, um eine Verbindung des Typs 2, d. h. CONNECT 2 im Befehl PRECOMPILE PROGRAM, sowie das einphasige Festschreiben, d. h. SYNCPOINT ONEPHASE im Befehl PRECOMPILE PROGRAM, anzugeben (siehe die Beschreibung im Handbuch *Application Development Guide*)

Wenn sich Datenbanken auf einem Host oder einem AS/400-Datenbank-Server befinden, wird das Produkt DB2 Connect zur Herstellung der Konnektivität zu diesen Servern benötigt. Informationen zur Einrichtung von DB2

Connect finden Sie in einem der Handbücher DB2 Connect *Einstieg*. Informationen zur Verwendung von DB2 Connect enthält das *DB2 Connect Benutzerhandbuch*.

Aktualisieren mehrerer Datenbanken

Wenn Ihre Daten über mehrere Datenbanken verteilt sind, wollen Sie vielleicht mehrere Datenbanken in einer einzelnen Transaktion lesen und aktualisieren. Diese Art des Datenbankzugriffs wird als *Aktualisierung auf mehreren Systemen* bezeichnet.

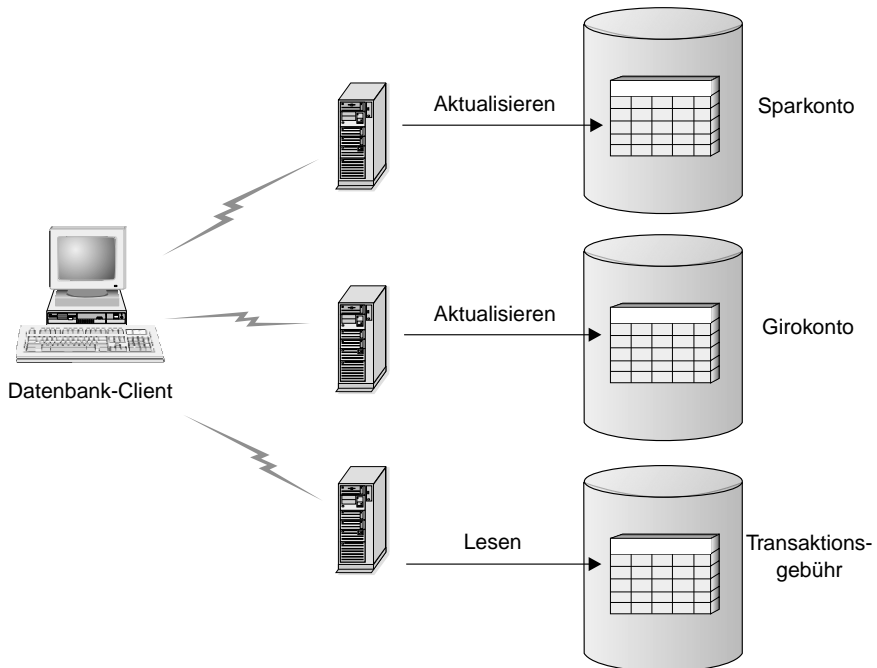


Abbildung 40. Aktualisieren mehrerer Datenbanken in einer Transaktion

Abb. 40 zeigt einen Datenbank-Client, der eine Geldtransaktionsanwendung ausführt, die auf drei Datenbank-Server zugreift: einer enthält das Girokonto, ein anderer das Sparkonto und der dritte den Plan für die Transaktionsgebühren.

Zur Einrichtung einer Geldtransaktionsanwendung für diese Umgebung sind folgende Schritte auszuführen:

1. Erstellen der erforderlichen Tabellen in den entsprechenden Datenbanken (siehe "Implementieren des Datenbankentwurfs" im Handbuch *Systemverwaltung: Implementierung*)

2. Wenn physisch fern, Einrichten der Datenbank-Server für die Verwendung der geeigneten Übertragungsprotokolle (siehe Beschreibung im Handbuch *DB2 Installation und Konfiguration Ergänzung*)
3. Wenn physisch fern, Katalogisieren der Knoten und der Datenbanken, um die Datenbanken auf den Datenbank-Servern zu identifizieren (siehe die Beschreibung in den Handbüchern *Einstieg*)
4. Vorkompilieren Ihres Anwendungsprogramms, um eine Verbindung des Typs 2, d. h. CONNECT 2 im Befehl PRECOMPILE PROGRAM, sowie das einphasige Festschreiben, d. h. SYNCPOINT ONEPHASE im Befehl PRECOMPILE PROGRAM, anzugeben (siehe die Beschreibung im Handbuch *Application Development Guide*)
5. Konfigurieren des DB2-Transaktionsmanagers (TM) (siehe die Beschreibung im Abschnitt „Verwenden des DB2-Transaktionsmanagers“).

Verwenden des DB2-Transaktionsmanagers

Der Datenbankmanager stellt Transaktionsmanagerfunktionen bereit, mit deren Hilfe das Aktualisieren mehrerer Datenbanken innerhalb einer einzelnen Arbeitseinheit koordiniert werden kann. Der Datenbank-Client koordiniert automatisch die Arbeitseinheit und verwendet eine *Transaktionsmanagerdatenbank*, um jede Transaktion zu registrieren und ihren Fertigstellungsstatus zu protokollieren.

Wenn Sie einen XA-konformen Transaktionsmanager wie IBM TXSeries, BEA Tuxedo oder Microsoft Transaction Server verwenden, finden Sie Anweisungen zur Integration in „Kapitel 10. Entwerfen für Transaktionsmanager“ auf Seite 197.

Wenn Sie zur Koordinierung Ihrer Transaktionen DB2 UDB für auf UNIX basierende Systeme, Windows-Betriebssysteme oder OS/2 verwenden, müssen Sie bestimmte Konfigurationsanforderungen erfüllen. Wenn Sie ausschließlich TCP/IP zur Kommunikation verwenden, und DB2 UDB und DB2 für OS/390 die einzigen Datenbank-Server sind, die an Ihren Transaktionen beteiligt sind, ist die Konfiguration recht einfach.

DB2 UDB und DB2 für OS/390 mit TCP/IP-Konnektivität: Wenn jeder der folgenden Punkte auf Ihre Umgebung zutrifft, gestalten sich die Konfigurationsschritte für eine Aktualisierung auf mehreren Systemen relativ einfach.

- Für die gesamte Kommunikation mit fernen Datenbank-Servern (einschließlich DB2 UDB für OS/390) wird ausschließlich TCP/IP verwendet.
- DB2 UDB für auf UNIX basierende Systeme, Windows-Betriebssysteme, OS/2 oder OS/390 sind die einzigen, an der Transaktion beteiligten Datenbank-Server.
- Der Synchronisationspunktmanager (SPM) von DB2 Connect ist nicht konfiguriert.

Der DB2 Synchronisationspunktmanager wird automatisch bei der Erstellung eines DB2-Exemplars konfiguriert und ist erforderlich, wenn:

- SNA-Konnektivität mit Host- oder AS/400-Datenbank-Servern für Aktualisierungen auf mehreren Systemen verwendet wird.
- Ein XA-konformer Transaktionsmanager (wie ein IBM TXSeries CICS) die zweiphasige Festschreibung koordiniert.

Dies gilt sowohl für die SNA- als auch für die TCP/IP-Konnektivität mit den Host- oder AS/400-Datenbank-Servern. Weitere Informationen finden Sie in „Kapitel 10. Entwerfen für Transaktionsmanager“ auf Seite 197.

Wenn in Ihrer Umgebung der DB2 Connect-

Synchronisationspunktmanager nicht erforderlich ist, können Sie ihn durch folgenden Befehl, der auf dem DB2 Connect-Server eingegeben wird, ausschalten: `db2 update dbm cfg using spm_name NULL`

Anschließend muß DB2 gestoppt und erneut gestartet werden.

Die Datenbank, die als Transaktionsmanagerdatenbank fungieren soll, wird auf dem Datenbank-Client durch den Konfigurationsparameter `tm_database` des Datenbankmanagers bestimmt. Weitere Informationen zu diesem Konfigurationsparameter finden Sie in "Konfigurieren von DB2" im Handbuch *Systemverwaltung: Optimierung*. Bei der Einstellung dieses Konfigurationsparameters sind folgende Faktoren zu beachten:

- Die Transaktionsmanagerdatenbank kann sein:

- Eine Datenbank unter DB2 UDB für auf UNIX basierende Systeme, Windows-Betriebssysteme oder OS/2

- Eine Datenbank unter DB2 für OS/390 Version 5 oder höher

Dies ist der empfohlene Datenbank-Server, der als Transaktionsmanagerdatenbank verwendet werden sollte. OS/390-Systeme sind in der Regel sicherer als Workstation-Server, da sie die Möglichkeit versehentlichen Ausschaltens, Neustartens usw. verringern. Aus diesem Grund sind die Wiederherstellungsprotokolle, die im Fall einer Resynchronisation verwendet werden, sicherer.

- Wenn der Wert `1ST_CONN` für den Konfigurationsparameter `tm_database` angegeben ist, wird die erste Datenbank, zu der eine Anwendung eine Verbindung herstellt, als Transaktionsmanagerdatenbank verwendet.

Bei der Verwendung von `1ST_CONN` ist besondere Sorgfalt geboten. Sie sollten diese Konfiguration nur verwenden, wenn es einfach ist, sicherzustellen, daß alle betroffenen Datenbanken korrekt katalogisiert werden, d. h. wenn folgende Bedingungen zutreffen:

- Der Datenbank-Client, der die Transaktion einleitet, befindet sich in demselben Exemplar, das die beteiligten Datenbanken, einschließlich der Transaktionsmanagerdatenbank, enthält.
- Sie verwenden die DCE Verzeichnisservices zum Katalogisieren und Verwalten des Zugriffs auf Ihre Datenbanken.

Falls Ihre Anwendung versucht, die Verbindung zu der Datenbank, die als Transaktionsmanagerdatenbank fungiert, zu trennen, wird eine Fehlermeldung ausgegeben. Die Verbindung bleibt in diesem Fall erhalten, bis die Arbeitseinheit festgeschrieben ist.

Andere Umgebungen: Wenn in Ihrer Umgebung folgendes zutrifft:

- Es wird nicht nur TCP/IP für die Kommunikation mit fernen Datenbank-Servern verwendet (es wird z. B. NETBIOS verwendet).
- Es wird auf DB2 für MVS Version 3 oder Version 4, DB2 für AS/400 oder DB2 für VM&VSE zugegriffen.
- Es wird auf DB2 für OS/390 über SNA zugegriffen.
- Der DB2 Connect-Synchronisationspunktmanager wird für den Zugriff auf Host- oder AS/400-Datenbank-Server verwendet.

In diesen Fällen sind die Konfigurationsschritte für die Aktualisierung auf mehreren Systemen etwas komplizierter.

Die Datenbank, die als Transaktionsmanagerdatenbank fungieren soll, wird auf dem Datenbank-Client durch den Konfigurationsparameter *tm_database* des Datenbankmanagers bestimmt. Weitere Informationen zu diesem Konfigurationsparameter finden Sie in "Konfigurieren von DB2" im Handbuch *Systemverwaltung: Optimierung*. Bei der Einstellung dieses Konfigurationsparameters sind folgende Faktoren zu beachten:

- Die Transaktionsmanagerdatenbank kann eine Datenbank unter DB2 UDB für auf UNIX basierende Systeme, Windows-Betriebssysteme oder OS/2 sein.
- Wenn der Wert 1ST_CONN für den Konfigurationsparameter *tm_database* angegeben ist, wird die erste Datenbank, zu der eine Anwendung eine Verbindung herstellt, als Transaktionsmanagerdatenbank verwendet.

Bei der Verwendung von 1ST_CONN ist besondere Sorgfalt geboten. Sie sollten diese Konfiguration nur verwenden, wenn es einfach ist, sicherzustellen, daß alle betroffenen Datenbanken korrekt katalogisiert werden, d. h. wenn folgende Bedingungen zutreffen:

- Der Datenbank-Client, der die Transaktion einleitet, befindet sich in demselben Exemplar, das die beteiligten Datenbanken, einschließlich der Transaktionsmanagerdatenbank, enthält.
- Sie verwenden die DCE Verzeichnisservices zum Katalogisieren und Verwalten des Zugriffs auf Ihre Datenbanken.

Falls Ihre Anwendung versucht, die Verbindung zu der Datenbank, die als Transaktionsmanagerdatenbank fungiert, zu trennen, wird eine Fehlermeldung ausgegeben. Die Verbindung bleibt in diesem Fall erhalten, bis die Arbeitseinheit festgeschrieben ist.

Weitere Überlegungen zur Konfiguration

Bei der Einrichtung der Umgebung sind die folgenden Konfigurationsparameter zu berücksichtigen: Weitere Informationen zur Einstellung dieser Parameter finden Sie im *DB2 Connect Benutzerhandbuch*.

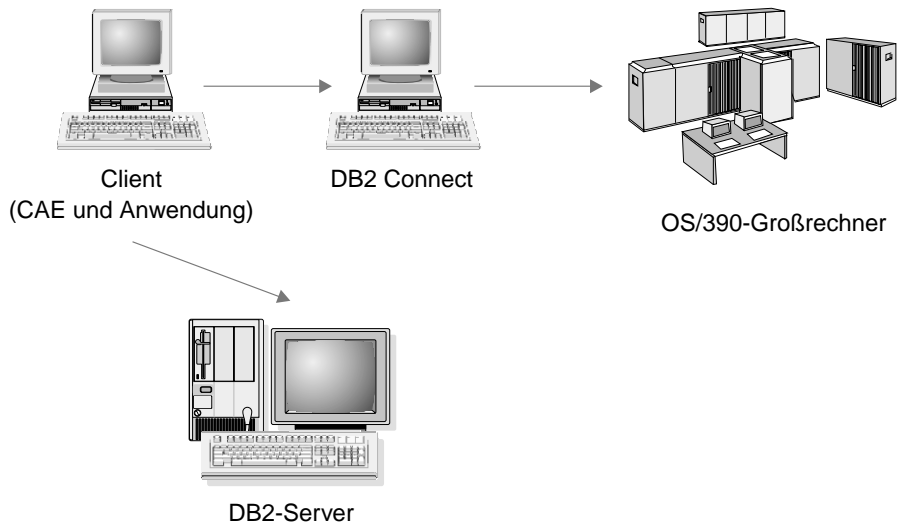


Abbildung 41. Überlegungen zur Konfiguration

Konfigurationsparameter des Datenbankmanagers

- *tm_database*
Mit diesem Parameter wird der Name der Transaktionsmanagerdatenbank für das jeweilige DB2-Exemplar definiert.
- *spm_name*
Mit diesem Parameter wird der Name des Exemplars mit dem DB2 Connect-Synchronisationspunktmanager für den Datenbankmanager definiert. Für eine erfolgreiche Resynchronisation muß der Name innerhalb Ihres gesamten Netzwerks eindeutig sein.
- *resync_interval*
Mit diesem Parameter wird das Zeitintervall (in Sekunden) angegeben, nach dem der DB2-Transaktionsmanager, der DB2-Server-Datenbankmanager und der DB2 Connect-Synchronisationspunktmanager den Versuch wiederholen sollen, alle ausstehenden unbestätigten Transaktionen wiederherzustellen.
- *spm_log_file_sz*
Mit diesem Parameter wird die Größe (in 4-KB-Seiten) der SPM-Protokolldatei angegeben.
- *spm_max_resync*

Mit diesem Parameter wird die Anzahl der Agenten angegeben, die gleichzeitig Resynchronisationsoperationen durchführen können.

- *spm_log_path*

Mit diesem Parameter wird der Protokollpfad für die SPM-Protokolldateien angegeben.

Konfigurationsparameter der Datenbank

- *maxappls*

Mit diesem Parameter wird die zulässige maximale Anzahl aktiver Anwendungen angegeben. Der Wert dieses Parameters muß gleich oder größer sein als die Summe der verbundenen Anwendungen plus die Anzahl dieser Anwendungen, die sich gleichzeitig im Prozeß einer zweiphasigen Festschreibung bzw. einer ROLLBACK-Operation befinden können, plus die angenommene Anzahl unbestätigter Transaktionen, die zu einem beliebigen Zeitpunkt gleichzeitig vorhanden sein können. Weitere Informationen zu unbestätigten Transaktionen finden Sie in „Beheben von Problemen bei der zweiphasigen Festschreibung“ auf Seite 193.

- *autorestart*

Mit diesem Konfigurationsparameter der Datenbank wird festgelegt, ob die Routine RESTART DATABASE bei Bedarf automatisch aufgerufen wird. Der Standardwert ist YES (d. h. aktiviert).

Für eine Datenbank, die unbestätigte Transaktionen enthält, ist eine RESTART DATABASE-Operation für den Neustart erforderlich. Wenn *autorestart* nicht aktiviert ist und die letzte Verbindung zur Datenbank getrennt wurde, schlägt die nächste Verbindungsanforderung fehl, so daß ein explizites Aufrufen des Befehls RESTART DATABASE erforderlich wird. Diese Bedingung bleibt solange bestehen, bis die unbestätigten Transaktionen entweder durch die Resynchronisationsoperation des Transaktionsmanagers oder durch eine vom Administrator eingeleitete heuristische Operation beseitigt wird. Wenn der Befehl RESTART DATABASE abgesetzt wird, wird eine Nachricht zurückgegeben, wenn unbestätigte Transaktionen in der Datenbank vorhanden sind. Der Administrator kann dann mit Hilfe des Befehls LIST INDOUBT TRANSACTIONS und anderer Befehle des Befehlszeilenprozessors Informationen über diese unbestätigten Transaktionen erhalten.

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

Host- oder IBM AS/400-Anwendungen, die auf einen LAN-basierten DB2 Universal Database-Server zur Aktualisierung auf mehreren Systemen zugreifen

DB2 Universal Database unterstützt keine Aktualisierung auf mehreren Systemen von Host- oder von IBM AS/400-Datenbank-Clients mit Hilfe der TCP/IP-Konnektivität. In diesem Fall wird nur die SNA-Konnektivität (Systems Network Architecture) unterstützt. Der DB2-

Synchronisationspunktmanager wird für Aktualisierungen auf mehreren Systemen benötigt. DB2 Connect wird in diesem Szenario nicht eingesetzt.

Der Datenbank-Server, auf den vom Host- oder vom IBM AS/400-Datenbank-Client zugegriffen wird, muß nicht lokal auf der Workstation vorhanden sein, auf der sich der DB2-Synchronisationspunktmanager befindet. Der Host- oder IBM AS/400-Datenbank-Client könnte eine Verbindung zu einem DB2 UDB-Server herstellen, indem er die Workstation mit dem DB2-Synchronisationspunktmanager als Übergangs-Gateway verwendet. Dadurch können Sie die Workstation mit dem DB2-Synchronisationspunktmanager in einer sicheren Umgebung isolieren, während die tatsächlichen DB2 UDB-Server in Ihrer Organisation ferne Server sind. Dadurch kann auch eine Datenbank auf der DB2 Server-Plattform der Version 2 an Aktualisierungen auf mehreren Systemen beteiligt sein, die von Host- oder IBM AS/400-Datenbank-Clients gestartet wurden.

Folgende Schritte sind auszuführen:

- Auf der Workstation, auf die durch die Host- oder AS/400-Anwendung direkt zugegriffen werden soll:
 1. Installieren der DB2 Universal Database Enterprise Edition oder Enterprise - Extended Edition, um die Aktualisierung auf mehreren Systemen mit Host- oder IBM AS/400-Datenbank-Clients bereitzustellen.
 2. Erstellen eines Datenbanke Exemplars auf demselben System. Sie können z. B. das Standardexemplar DB2 verwenden oder mit dem folgenden Befehl ein neues Exemplar erstellen:

```
db2icrt meinexemplar
```
 3. Eingeben der erforderlichen Lizenzinformationen.
 4. Sicherstellen, daß der Registrierungswert DB2COMM den Wert APPC enthält.
 5. Konfigurieren der erforderlichen SNA-Kommunikation. Wenn die unterstützten IBM SNA-Produkte eingesetzt werden, werden die für den DB2-Synchronisationspunktmanager erforderlichen SNA-Profile auf der Grundlage des für den Konfigurationsparameter *spm_name* des Datenbankmanagers definierten Werts automatisch erstellt. Jeder andere unterstützte SNA-Stapel (Stack) macht eine manuelle Konfiguration erforderlich. Genauere Informationen enthält das Handbuch *DB2 Installation und Konfiguration Ergänzung*.
 6. Bestimmen des Werts, der für den Konfigurationsparameter *spm_name* des Datenbankmanagers anzugeben ist. Dieser Parameter wird bei der Erstellung eines DB2-Exemplars mit einem aus dem TCP/IP-Host-Namen für die Maschine abgeleiteten Wert vorkonfiguriert. Falls dieser Wert annehmbar und in der Umgebung eindeutig ist, sollte er nicht geändert werden.

7. Falls erforderlich, Aktualisieren des Werts für *spm_name* auf dem DB2 Universal Database-Server mit Hilfe des Befehls UPDATE DATABASE MANAGER CONFIGURATION.
 8. Konfigurieren der für diese DB2-Workstation erforderlichen Kommunikation zur Verbindung zu fernen DB2 UDB-Servern, falls nötig.
 9. Konfigurieren der für ferne DB2 UDB-Server erforderlichen Kommunikation zur Verbindung mit diesem DB2-Server.
 10. Stoppen und erneutes Starten des Datenbankmanagers auf dem DB2 Universal Database-Server, um den SPM das erste Mal zu starten.
Sie sollten in der Lage sein, eine Verbindung von dieser Workstation aus zu fernen DB2 UDB-Servern herzustellen.
- Auf jedem fernen DB2 UDB-Server, auf den durch den Host- oder den IBM AS/400-Datenbank-Client zugegriffen wird:
 1. Konfigurieren der für die ferne DB2 UDB-Workstation mit dem DB2-Synchronisationspunktmanager erforderlichen Kommunikation zur Verbindung mit diesem DB2 UDB-Server.
 2. Stoppen und erneutes Starten des Datenbankmanagers.

Prozeß der zweiphasigen Festschreibung

Abb. 42 auf Seite 191 veranschaulicht die Schritte, die zu einer Aktualisierung auf mehreren Systemen gehören. Kenntnisse über die Art und Weise, wie eine Transaktion verwaltet wird, erweisen sich bei der Problemlösung als hilfreich, wenn während des Prozesses der zweiphasigen Festschreibung ein Fehler auftritt.

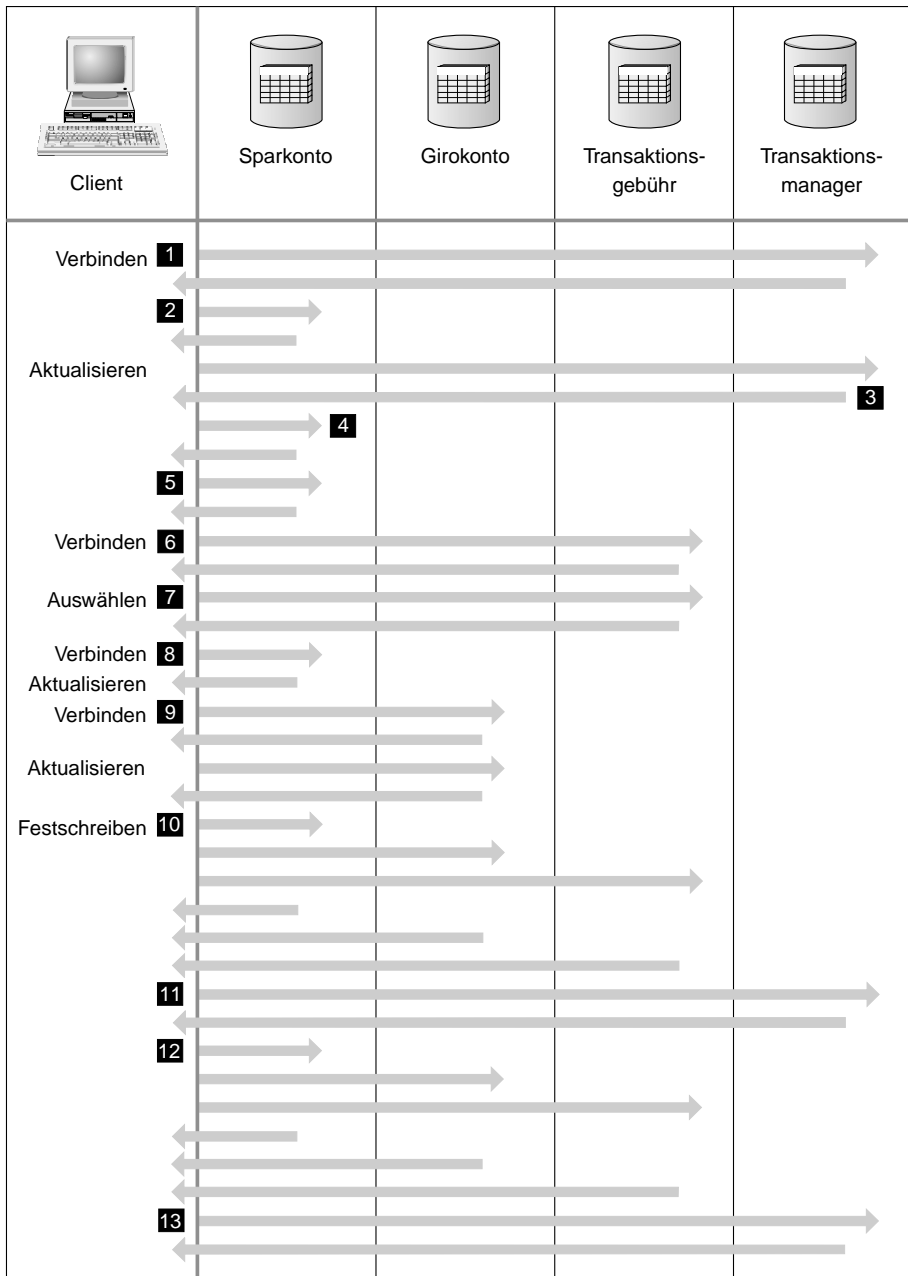


Abbildung 42. Aktualisieren mehrerer Datenbanken

- 0 Die Anwendung ist für die zweiphasige Festschreibung vorbereitet. Dies kann durch Vorkompileroptionen erreicht werden (Einzelangaben finden Sie im Handbuch *Application Development Guide*). Dies

kann auch durch eine Konfiguration von DB2 CLI (Call Level Interface) erreicht werden (Einzelangaben finden Sie im Handbuch *CLI Guide and Reference*).

- 1** Wenn der Datenbank-Client die Verbindung zur Datenbank SPARKO_DB herstellen will, stellt er zunächst intern die Verbindung zur Transaktionsmanagerdatenbank (TMD) her. Die TMD gibt eine Bestätigung an den Datenbank-Client zurück. Falls der Konfigurationsparameter *tm_database* des Datenbankmanagers auf den Wert *1ST_CONN* gesetzt ist, wird die Datenbank SPARKO_DB für die Dauer dieses Anwendungsexemplars zur Transaktionsmanagerdatenbank.
- 2** Die Verbindung zur Datenbank SPARKO_DB wird hergestellt und bestätigt.
- 3** Der Datenbank-Client beginnt mit der Aktualisierung der Tabelle SPAR_KONTO. Dies ist der Anfang der Arbeitseinheit. Die TMD reagiert auf den Datenbank-Client, indem sie eine Transaktions-ID für die Arbeitseinheit bereitstellt. Beachten Sie, daß die Registrierung einer Arbeitseinheit erfolgt, wenn die erste SQL-Anweisung in der Arbeitseinheit ausgeführt wird, und nicht während der Herstellung der Verbindung.
- 4** Nach dem Empfang der Transaktions-ID registriert der Datenbank-Client die Arbeitseinheit bei der Datenbank, die die Tabelle SPAR_KONTO enthält. Der Client erhält eine Antwort, die besagt, daß die Arbeitseinheit erfolgreich registriert wurde.
- 5** SQL-Anweisungen, die an der Datenbank SPARKO_DB ausgeführt werden, werden auf normale Weise verarbeitet. Die Antwort auf jede Anweisung wird im SQL-Kommunikationsbereich (SQLCA) zurückgegeben, wenn mit SQL-Anweisungen gearbeitet wird, die in ein Programm eingebettet sind. (Eine Beschreibung des SQL-Kommunikationsbereichs (SQLCA) finden Sie in den Handbüchern *Application Development Guide* und *SQL Reference*.)
- 6** Die Transaktions-ID wird in der Datenbank GEB_DB, die die Tabelle TRANSAKTION_GEB enthält, beim ersten Zugriff auf diese Datenbank innerhalb der Arbeitseinheit registriert.
- 7** SQL-Anweisungen, die für die Datenbank GEB_DB abgesetzt werden, werden auf normale Weise bearbeitet.
- 8** Zusätzliche SQL-Anweisungen können für die Datenbank SPARKO_DB ausgeführt werden, indem die Verbindung entsprechend eingerichtet wird. Da die Arbeitseinheit bei der Datenbank SPARKO_DB bereits registriert (**4**) wurde, muß der Datenbank-Client den Registrierungsschritt nicht wiederholen.

- 9** Für die Verbindung mit der Datenbank GIROKO_DB und ihre Verwendung gelten die gleichen Regeln, die unter **6** und **7** beschrieben sind.
- 10** Wenn der Datenbank-Client das Festschreiben der Arbeitseinheit anfordert, wird eine Nachricht *prepare* an alle Datenbanken gesendet, die an der Arbeitseinheit mitwirken. Jede Datenbank schreibt einen Satz "PREPARED" in ihre Protokolldatei und antwortet dem Datenbank-Client.
- 11** Nachdem der Datenbank-Client von allen Datenbanken eine positive Antwort erhalten hat, sendet er eine Nachricht an die Transaktionsmanagerdatenbank, die besagt, daß die Arbeitseinheit für das Festschreiben bereit (PREPARED) ist. Die Transaktionsmanagerdatenbank schreibt den Satz "PREPARED" in ihre Protokolldatei und teilt dem Client in einer Antwort mit, daß die zweite Phase der Festschreibung begonnen werden kann.
- 12** Während der zweiten Phase des Festschreibungsprozesses fordert der Datenbank-Client alle beteiligten Datenbanken in einer Nachricht zum Festschreiben auf. Jede Datenbank schreibt den Satz "COMMITTED" in ihre Protokolldatei und gibt die Sperren frei, die für diese Arbeitseinheit aktiv waren. Wenn die Datenbank das Festschreiben der Änderung beendet hat, sendet sie eine Antwort an den Client.
- 13** Nachdem der Datenbank-Client von allen beteiligten Datenbanken eine positive Antwort erhalten hat, sendet er eine Anforderung an die Transaktionsmanagerdatenbank, um ihr mitzuteilen, daß die Arbeitseinheit beendet wurde. Daraufhin schreibt die Transaktionsmanagerdatenbank einen Satz "COMMITTED" in ihre Protokolldatei, der anzeigt, daß die Arbeitseinheit beendet ist, und sendet eine Antwort an den Client, daß der Prozeß beendet wurde.

Beheben von Problemen bei der zweiphasigen Festschreibung

Das Beheben von Fehlerbedingungen ist eine normale Aufgabe bei der Anwendungsprogrammierung, Systemverwaltung, Datenbankverwaltung sowie beim Systembetrieb. Durch die Verteilung von Datenbanken über mehrere ferne Server erhöht sich das Potential von Fehlerbedingungen, die aus Netzwerk- oder Übertragungsstörungen resultieren können. Um die Datenintegrität zu gewährleisten, stellt der Datenbankmanager den Prozeß der zweiphasigen Festschreibung zur Verfügung. Eine Darstellung dieses Prozesses finden Sie in „Prozeß der zweiphasigen Festschreibung“ auf Seite 190 (siehe Punkte **10**, **11** und **12**). Im folgenden wird erläutert, wie Datenbankmanager Fehler während des Prozesses der zweiphasigen Festschreibung behandelt:

- **Fehler in der ersten Phase**

Wenn eine Datenbank mitteilt, daß sie die Festschreibung der Arbeitseinheit nicht vorbereiten (und kein "PREPARED" in die Protokolldatei eintragen) konnte, macht der Datenbank-Client die Arbeitseinheit in der zweiten Phase des Festschreibungsprozesses rückgängig. In diesem Fall wird *keine* PREPARE-Nachricht an die Transaktionsmanagerdatenbank gesendet.

Während der zweiten Phase der Festschreibung weist der Client alle beteiligten Datenbanken, die während der ersten Phase die Festschreibung erfolgreich vorbereitet haben, an, die Arbeitseinheit rückgängig (ROLL-BACK) zu machen. Jede Datenbank schreibt daraufhin einen Satz "ABORT" in ihre Protokolldatei und gibt die Sperren frei, die für diese Arbeitseinheit aktiv waren.

- **Fehler in der zweiten Phase**

Die Fehlerbehandlung ist in diesem Fall davon abhängig, ob die Transaktion in der zweiten Phase festgeschrieben oder rückgängig gemacht wird. Die zweite Phase macht die Transaktion nur dann rückgängig, wenn in der ersten Phase ein Fehler auftrat.

Wenn eine der beteiligten Datenbanken die Arbeitseinheit nicht festschreiben kann (u. U. aufgrund eines Übertragungsfehlers), versucht die Transaktionsmanagerdatenbank erneut, für die fehlgeschlagene Datenbank eine Festschreibung auszuführen. Die Anwendung wird jedoch über den SQL-Kommunikationsbereich (SQLCA) darüber informiert, daß die Festschreibung erfolgreich war. DB2 stellt sicher, daß die nicht festgeschriebene Transaktion im Datenbank-Server festgeschrieben wird. Mit Hilfe des Konfigurationsparameters *resync_interval* des Datenbankmanagers (siehe "Konfigurieren von DB2" im Handbuch *Systemverwaltung: Optimierung*) wird das Intervall festgelegt, das die Transaktionsmanagerdatenbank zwischen den einzelnen Versuchen, die Arbeitseinheit festzuschreiben, verstreichen läßt. Alle Sperren bleiben auf dem Datenbank-Server aktiv, bis die Arbeitseinheit festgeschrieben wird.

Fällt Transaktionsmanagerdatenbank aus, resynchronisiert sie die Arbeitseinheit, wenn sie erneut gestartet wird. Der Resynchronisationsprozeß versucht, alle *unbestätigten Transaktionen* zu beenden, d. h. die Transaktionen, die die erste Phase der Festschreibung beendet haben, deren zweite Phase jedoch nicht abgeschlossen wurde. Der Datenbankmanager, unter dem die Transaktionsmanagerdatenbank aktiv ist, führt zur Resynchronisation folgende Schritte aus:

1. Herstellen einer Verbindung zu den Datenbanken, die während der ersten Phase des Festschreibungsprozesses mitteilten, daß sie zur Festschreibung bereit ("PREPARED") waren
2. Versuch einer Festschreibung der unbestätigten Transaktionen in diesen Datenbanken. (Können die unbestätigten Transaktionen nicht gefunden werden, nimmt der Datenbankmanager an, daß die Datenbank die Transaktionen in der zweiten Phase des Festschreibungsprozesses erfolgreich festgeschrieben hat.)

3. Festschreiben der unbestätigten Transaktionen in der Transaktionsmanagerdatenbank, nachdem alle unbestätigten Transaktionen in den beteiligten Datenbanken festgeschrieben wurden

Wenn bei einer der beteiligten Datenbanken ein Fehler auftritt und die Datenbank erneut gestartet wird, fragt der Datenbankmanager für diese Datenbank die Transaktionsmanagerdatenbank nach dem Status dieser Transaktion ab, um festzustellen, ob die Transaktion rückgängig gemacht werden sollte. Wenn die Transaktion im Protokoll nicht gefunden wird, nimmt der Datenbankmanager an, daß die betreffende Transaktion rückgängig gemacht wurde, und macht die unbestätigte Transaktion in dieser Datenbank rückgängig. Andernfalls wartet die Datenbank auf eine Festschreibungsanforderung der Transaktionsmanagerdatenbank.

Wenn die Transaktion von einem Transaktionsverarbeitungsmonitor (XA-konformem Transaktionsmanager) koordiniert wurde, ist die Datenbank immer davon abhängig, daß die Resynchronisation vom TP-Monitor eingeleitet wird.

Wenn Sie aus einem bestimmten Grund nicht darauf warten können, daß der Transaktionsmanager unbestätigte Transaktionen automatisch auflöst, haben Sie die Möglichkeit, Maßnahmen zur manuellen Auflösung unbestätigter Transaktionen zu ergreifen. Dieser manuelle Prozeß wird manchmal als das „Treffen einer heuristischen Entscheidung“ bezeichnet. Weitere Informationen zur manuellen Wiederherstellung unbestätigter Transaktionen finden Sie in „Treffen einer heuristischen Entscheidung“ auf Seite 210.

Resynchronisieren unbestätigter Transaktionen bei AUTORESTART=OFF

Hinweise zur Konfiguration in einer Umgebung von DB2 Universal Database mit zweiphasiger Festschreibung finden Sie in „Weitere Überlegungen zur Konfiguration“ auf Seite 187.

Insbesondere ist zu beachten, daß, wenn der Konfigurationsparameter *autorestart* der Datenbank den Wert OFF hat und unbestätigte Transaktionen in der Transaktionsmanagerdatenbank oder den Ressourcenmanagerdatenbanken vorhanden sind, der Befehl `RESTART DATABASE` ausgeführt werden muß, um den Resynchronisationsprozeß zu starten. Wenn der Befehl `RESTART DATABASE` über den Befehlszeilenprozessor ausgeführt wird, verwenden Sie verschiedene Sitzungen. Wenn Sie eine andere Datenbank aus derselben Sitzung erneut starten, wird die Verbindung, die durch die vorherige Ausführung des Befehls `RESTART DATABASE` hergestellt wurde, beendet und muß erneut gestartet werden. Setzen Sie den Befehl `TERMINATE` ab, um die Verbindung zu beenden, wenn der Befehl `LIST INDOUBT TRANSACTIONS` keine weiteren unbestätigten Transaktionen mehr liefert.

Kapitel 10. Entwerfen für Transaktionsmanager

Wenn Sie neben den DB2-Datenbanken über weitere Ressourcen verfügen, die an einer zweiphasigen Festschreibungstransaktion beteiligt sein sollen, kann es sinnvoll sein, einen dem XA-Standard entsprechenden Transaktionsmanager zu verwenden. Wenn Ihre Transaktionen nur auf DB2-Datenbanken zugreifen, sollten Sie den DB2-Transaktionsmanager verwenden, der in „Aktualisieren mehrerer Datenbanken“ auf Seite 183 beschrieben wird.

Die folgenden Themen enthalten Informationen zur Verwendung des Datenbankmanagers mit einem XA-konformen Transaktionsmanager wie IBM TXSeries CICS, IBM TXSeries Encina, BEA Tuxedo oder Microsoft Transaction Server:

- „X/Open-Modell für die verteilte Transaktionsverarbeitung“ auf Seite 198
- „Einrichten einer Datenbank als Ressourcenmanager“ auf Seite 203
- „Verwenden der XA-Zeichenfolgen zum Öffnen und Schließen“ auf Seite 203
- „Neues Format der XA-Zeichenfolge zum Öffnen für DB2 Version 7“ auf Seite 203
- „Werte für TPM und TP_MON_NAME“ auf Seite 206
- „Format der XA-Zeichenfolge zum Öffnen für frühere Versionen von DB2“ auf Seite 209
- „Aktualisieren von Host- oder AS/400-Datenbank-Servern“ auf Seite 209
- „Überlegungen zu Datenbankverbindungen“ auf Seite 210
- „Treffen einer heuristischen Entscheidung“ auf Seite 210
- „Überlegungen zur Sicherheit“ auf Seite 213
- „Überlegungen zur Konfiguration“ auf Seite 214
- „Unterstützte XA-Funktion“ auf Seite 216
- „Bestimmung von XA-Schnittstellenfehlern“ auf Seite 218
- „Konfigurieren des XA-Transaktionsmanagers zur Verwendung von DB2 UDB“ auf Seite 219
- „Konfigurieren von Microsoft Transaction Server“ auf Seite 224

Wenn Sie einen dem XA-Standard entsprechenden Transaktionsmanager verwenden oder implementieren, stehen weitere Informationen auf unserer Web-Adresse für technische Unterstützung zur Verfügung:

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

Dort wählen Sie "DB2 Universal Database" aus, und durchsuchen anschließend die Web-Site mit dem Schlüsselwort "XA" nach den neuesten verfügbaren Informationen zu Transaktionsmanagern, die dem XA-Standard entsprechen.

X/Open-Modell für die verteilte Transaktionsverarbeitung

Das X/Open-Modell für die verteilte Transaktionsverarbeitung (DTP - Distributed Transaction Processing) umfaßt drei in Wechselbeziehung zueinander stehende Komponenten:

- „Anwendungsprogramm (AP)“
- „Transaktionsmanager (TM)“ auf Seite 200
- „Ressourcenmanager (RM)“ auf Seite 201

Abb. 43 veranschaulicht dieses Modell und zeigt die Beziehungen dieser Komponenten untereinander.

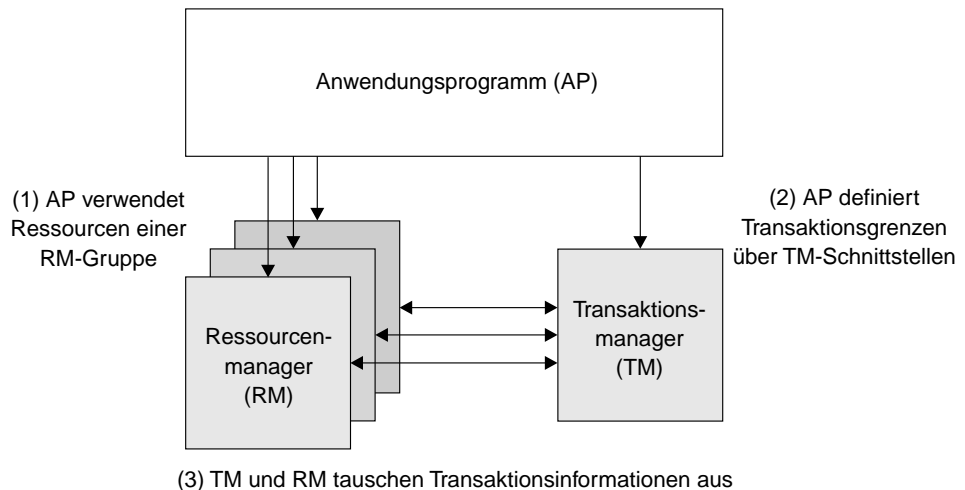


Abbildung 43. X/Open-Modell für die verteilte Transaktionsverarbeitung (DTP)

Anwendungsprogramm (AP)

Das Anwendungsprogramm (AP) definiert die Transaktionsgrenzen und gibt die anwendungsspezifischen Aktionen an, aus denen die Transaktion besteht.

Ein CICS*-Anwendungsprogramm möchte zum Beispiel auf Ressourcenmanager (RMs) wie eine Datenbank und eine CICS-Warteschlange mit Übergangsdaten (CICS Transient Data Queue) zugreifen und eine Programmlogik verwenden, um die Daten zu ändern. Jede Zugriffsanforderung wird an die entsprechenden Ressourcenmanager über für diesen Ressourcenmanager spezifische Funktionsaufrufe übergeben. Im Fall von DB2 können diese

Funktionsaufrufe vom DB2-Precompiler für jede SQL-Anweisung generiert worden sein, oder es kann sich um Datenbankaufrufe handeln, die direkt vom Programmierer mit Hilfe der APIs codiert wurden.

Zum Transaktionsmanagerprodukt (TM-Produkt) gehört in der Regel ein Transaktionsverarbeitungsmonitor (TP-Monitor) zur Ausführung der Benutzeranwendung. Der TP-Monitor stellt APIs bereit, die einer Anwendung die Möglichkeit geben, eine Transaktion zu starten und zu beenden, und die Funktionen zur Terminierung der zeitlichen Abläufe von Anwendungen sowie zum Lastausgleich unter den vielen Benutzern, die die Anwendung ausführen möchten, zur Verfügung stellen. Das Anwendungsprogramm (AP) in einer DTP-Umgebung ist im Grunde eine Kombination aus der Benutzeranwendung und dem TP-Monitor.

Um eine effiziente Online-Transaktionsverarbeitung (OLTP, Online Transaction Processing) zu ermöglichen, ordnet der TP-Monitor eine Reihe von Server-Prozessen beim Start im voraus zu und verwaltet und verwendet diese anschließend für die zahlreichen Benutzertransaktionen. Auf diese Weise werden Systemressourcen eingespart, da mehr Benutzer mit Hilfe einer geringeren Anzahl von Server-Prozessen und ihrer zugehörigen RM-Prozesse unterstützt werden. Durch die mehrfache Verwendung dieser Prozesse wird auch der Systemaufwand vermieden, der mit dem Starten eines Prozesses im Transaktionsmanager oder den Ressourcenmanagern (RM) für jede Benutzertransaktion bzw. jedes Benutzerprogramm verbunden ist. (Ein Programm ruft ein oder mehrere Transaktionen auf.) Dies bedeutet auch, daß die Server-Prozesse gegenüber dem Transaktionsmanager und den Ressourcenmanagern als die wirklichen „Benutzerprozesse“ auftreten. Hieraus ergeben sich bestimmte Konsequenzen für die Sicherheitsverwaltung und die Anwendungsprogrammierung. Nähere Informationen finden Sie in „Überlegungen zur Sicherheit“ auf Seite 213.

Die folgenden Arten von Transaktionen sind von einem TP-Monitor aus möglich:

- Nicht-XA-Transaktionen

Diese Transaktionen betreffen Ressourcenmanager, die für den Transaktionsmanager nicht definiert sind und daher nicht unter dem Protokoll für die zweiphasige Festschreibung des Transaktionsmanagers koordiniert werden. Diese Transaktionen können erforderlich werden, wenn eine Anwendung auf einen Ressourcenmanager zugreifen muß, der die XA-Schnittstelle nicht unterstützt. Der TP-Monitor stellt einfach effiziente Einrichtungen zur Zeitplanung und zum Lastausgleich bereit. Da der Transaktionsmanager den Ressourcenmanager nicht explizit für eine XA-Verarbeitung „öffnet“, behandelt der Ressourcenmanager diese Anwendung wie jede andere Anwendung, die in einer Nicht-DTP-Umgebung ausgeführt wird.

- Globale Transaktionen

Diese Transaktionen arbeiten mit Ressourcenmanagern, die gegenüber dem Transaktionsmanager definiert sind und der Steuerung der zweiphasigen Festschreibung des Transaktionsmanagers unterliegen. Eine globale Transaktion stellt eine Arbeitseinheit dar, die auf einen oder mehrere Ressourcenmanager zugreift. Eine *Transaktionsverzweigung* ist der Teil der Arbeit zwischen einem Transaktionsmanager und einem Ressourcenmanager, der die globale Transaktion unterstützt. Eine globale Transaktion kann mehrere Transaktionsverzweigungen umfassen, wenn auf mehrere Ressourcenmanager über einen oder mehrere vom Transaktionsmanager koordinierte Anwendungsprozesse zugegriffen wird.

Lose gekoppelte globale Transaktionen sind vorhanden, wenn jeder Prozeß einer Anzahl von Anwendungsprozessen auf die Ressourcenmanager so zugreift, als wäre er in einer getrennten globalen Transaktion, die Anwendungen jedoch alle der Koordination durch den Transaktionsmanager unterliegen. Jeder Anwendungsprozeß verfügt über eine eigene Transaktionsverzweigung innerhalb eines Ressourcenmanagers. Wenn eine COMMIT- oder ROLLBACK-Operation von einem der Anwendungsprogramme, dem Transaktionsmanager oder den Ressourcenmanagern angefordert wird, werden alle Transaktionsverzweigungen zusammen abgeschlossen. Es obliegt den Anwendungen, gegenseitige Sperren von Ressourcen unter den Transaktionsverzweigungen auszuschließen. (Beachten Sie, daß die Koordination der Transaktionen, die durch den DB2-Transaktionsmanager für Anwendungen implementiert wird, die mit der Option SYNCPOINT(TWOPHASE) vorbereitet (PREPARE) wurden, diesen lose gekoppelten globalen Transaktionen ungefähr entspricht. Siehe „Aktualisieren mehrerer Datenbanken“ auf Seite 183.)

Fest gekoppelte globale Transaktionen sind vorhanden, wenn mehrere Anwendungsprozesse abwechselnd dieselbe Transaktionsverzweigung in einem Ressourcenmanager verwenden, um Operationen auszuführen. Dem Ressourcenmanager gegenüber bilden die beiden Anwendungsprozesse eine Einheit. Der Ressourcenmanager muß dafür sorgen, daß keine gegenseitigen Ressourcensperren innerhalb der Transaktionsverzweigung auftreten.

Transaktionsmanager (TM)

Der Transaktionsmanager (TM) ordnet Transaktionen Kennungen zu, überwacht die Verarbeitung von Transaktionen und ist zuständig für die Beendigung bzw. für Fehler der Transaktionen. Die Kennungen für Transaktionsverzweigungen (XIDs) werden vom Transaktionsmanager zugeordnet, um die globale Transaktion und die spezielle Verzweigung innerhalb eines Ressourcenmanagers (RM) zu identifizieren. Diese Kennung ist das Korrelations-Token zwischen dem Protokoll in einem Transaktionsmanager (TM) und dem Protokoll in einem Ressourcenmanager (RM). Die XID wird für eine zweiphasige Festschreibung bzw. für eine ROLLBACK-Operation benötigt, um die *Resynchronisation* (abgekürzt *resync*) bei einem Systemstart durchzuführen oder um dem Administrator die Möglichkeit zu geben, bei Bedarf eine *heuristische* Operation (auch als *manueller Eingriff* bezeichnet) vorzunehmen.

Nach dem Start fordert ein TP-Monitor den Transaktionsmanager auf, alle Ressourcenmanager (RMs) zu öffnen, die in einer Gruppe von Anwendungsservern definiert sind. Der Transaktionsmanager übergibt die **xa_open**-Aufrufe an die Ressourcenmanager, damit diese für die DTP-Verarbeitung initialisiert werden können. Im Rahmen dieser Startprozedur führt der Transaktionsmanager eine Resynchronisation (resync) durch, um alle *unbestätigten Transaktionen* aufzulösen. Eine unbestätigte Transaktion ist eine globale Transaktion, die nicht vollständig zu Ende geführt wurde. Dies geschieht, wenn auf den Transaktionsmanager (oder mindestens auf einen Ressourcenmanager) nach einer erfolgreich abgeschlossenen ersten Phase (Vorbereitungsphase) des Protokolls für zweiphasige Festschreibung nicht mehr zugegriffen werden kann. Der Ressourcenmanager weiß solange nicht, ob seine Verzweigung der Transaktion festzuschreiben oder rückgängig zu machen ist, bis der Transaktionsmanager sein eigenes Protokoll mit den Protokollen der Ressourcenmanager abgleichen kann, wenn sie wieder verfügbar werden. Zur Durchführung der Resynchronisation setzt der Transaktionsmanager einen **xa_recover**-Aufruf an jeden Ressourcenmanager mindestens einmal ab, um alle unbestätigten Transaktionen zu identifizieren. Der Transaktionsmanager vergleicht die Antworten mit den Informationen im eigenen Protokoll, um zu bestimmen, ob die Ressourcenmanager angewiesen werden sollen, eine **xa_commit**-Operation oder eine **xa_rollback**-Operation für diese Transaktionen durchzuführen. Wenn ein Ressourcenmanager seine Verzweigung einer unbestätigten Transaktion aufgrund einer heuristischen Operation durch den Administrator bereits festgeschrieben oder rückgängig gemacht hat, setzt der Transaktionsmanager einen Aufruf **xa_forget** an diesen Ressourcenmanager ab, um die Resynchronisation abzuschließen.

Wenn eine Benutzeranwendung eine COMMIT- oder ROLLBACK-Operation anfordert, muß sie die vom TP-Monitor oder Transaktionsmanager bereitgestellte API verwenden, so daß der Transaktionsmanager die COMMIT- und ROLLBACK-Operation unter allen beteiligten Ressourcenmanagern koordinieren kann. Wenn zum Beispiel eine CICS-Anwendung die CICS-Anforderung SYNCPOINT absetzt, um eine Transaktion festzuschreiben, setzt der Transaktionsmanager CICS XA TM (im Encina-Server implementiert) seinerseits die entsprechenden XA-Aufrufe wie **xa_end**, **xa_prepare**, **xa_commit** oder **xa_rollback** ab, um den Ressourcenmanager anzuweisen, die Transaktion festzuschreiben oder rückgängig zu machen. Der Transaktionsmanager kann eine einphasige Festschreibung anstelle der zweiphasigen Festschreibung durchführen, wenn nur ein Ressourcenmanager beteiligt ist oder wenn ein Ressourcenmanager antwortet, daß seine Transaktionsverzweigung im Nur-Lese-Modus arbeitet.

Ressourcenmanager (RM)

Ein Ressourcenmanager (RM) stellt den Zugriff auf gemeinsame Ressourcen wie Datenbanken zur Verfügung.

DB2 als Ressourcenmanager einer Datenbank kann an einer *globalen Transaktion* beteiligt sein, die von einem dem XA-Standard entsprechenden Transaktionsmanager koordiniert wird. Wie für die XA-Schnittstelle erforderlich stellt der Datenbankmanager eine externe C-Variable *db2xa_switch* des Typs *xa_switch_t* bereit, um die XA-Schalterstruktur an den Transaktionsmanager zurückzugeben. Diese Datenstruktur enthält die Adressen der verschiedenen XA-Routinen, die vom Transaktionsmanager aufzurufen sind, und die Betriebsmerkmale des Ressourcenmanagers. Weitere Informationen zu XA-Funktionen, die vom Datenbankmanager unterstützt werden, finden Sie in „Unterstützte XA-Funktion“ auf Seite 216.

Es gibt zwei Methoden, mit denen der Ressourcenmanager seine Teilnahme an der jeweiligen globalen Transaktion registrieren kann: *statische Registrierung* und *dynamische Registrierung*:

- Für die statische Registrierung ist es erforderlich, daß der Transaktionsmanager (für jede Transaktion) die Folge von Aufrufen **xa_start**, **xa_end** und **xa_prepare** an alle Ressourcenmanager absetzt, die für die Server-Anwendung definiert sind, unabhängig davon, ob der einzelne Ressourcenmanager von der Transaktion verwendet wird. Dies ist nicht effizient, wenn nicht jeder Ressourcenmanager an jeder Transaktion beteiligt ist. Der Grad der Ineffizienz ist proportional zur Anzahl der definierten Ressourcenmanager.
- Die (von DB2 genutzte) dynamische Registrierung ist flexibel und effizient. Ein Ressourcenmanager registriert sich dem Transaktionsmanager gegenüber nur dann mit Hilfe eines **ax_reg**-Aufrufs, wenn der Ressourcenmanager eine Anforderung für seine Ressource empfängt. Beachten Sie, daß es keine Leistungsnachteile bei dieser Methode gibt, wenn lediglich ein Ressourcenmanager definiert ist oder wenn jeder Ressourcenmanager von jeder Transaktion verwendet wird, da die Aufrufe **ax_reg** und **xa_start** ähnliche Pfade im Transaktionsmanager haben.

Eine XA-Schnittstelle stellt eine Zweiwegeübertragung zwischen einem Transaktionsmanager und einem Ressourcenmanager her. Bei der XA-Schnittstelle handelt es sich um eine Systemebenesschnittstelle zwischen den beiden DTP-Softwarekomponenten, und nicht um eine gewöhnliche Schnittstelle für Anwendungsprogramme, für die ein Entwickler Aufrufe codiert. Anwendungsentwickler sollten jedoch mit den durch DTP-Softwarekomponenten bedingten Programmier einschränkungen vertraut sein. Informationen zu Gesichtspunkten, die bei Programmierung unter Verwendung der X/Open XA-Schnittstelle zu beachten sind, finden Sie im Handbuch *Application Development Guide*.

Obwohl die XA-Schnittstelle unveränderlich ist, kann jeder dem XA-Standard entsprechende Transaktionsmanager einen Ressourcenmanager auf eine eigene, produktspezifische Weise integrieren. Informationen zur Integration Ihres

DB2-Produkts als Ressourcenmanager mit einem bestimmten Transaktionsmanager finden Sie in der Dokumentation des entsprechenden Transaktionsmanagerprodukts. Informationen zur Integration in bezug auf die gängigsten TP-Monitore finden Sie in „Konfigurieren des XA-Transaktionsmanagers zur Verwendung von DB2 UDB“ auf Seite 219.

Einrichten einer Datenbank als Ressourcenmanager

Jede Datenbank wird als ein separater Ressourcenmanager (RM) beim Transaktionsmanager (TM) definiert, und die Datenbank muß durch eine XA-Zeichenfolge zum Öffnen identifiziert werden. Eine Beschreibung des DB2-Formats der XA-Zeichenfolge zum Öffnen finden Sie unter „Verwenden der XA-Zeichenfolgen zum Öffnen und Schließen“.

Verwenden der XA-Zeichenfolgen zum Öffnen und Schließen

Die XA-Zeichenfolge (`xa_open`) des Datenbankmanagers zum Öffnen besitzt zwei akzeptierte Formate. Ein Format wird in DB2 Version 7 neu eingeführt. Das zweite Format wird von früheren Versionen von DB2 verwendet und bleibt aus Gründen der Kompatibilität mit früheren Versionen erhalten. Neue Implementierungen sollten mit dem neuen Format arbeiten, und ältere Implementierungen sollten nach Möglichkeit auf das neue Format umgestellt werden. Zukünftige Versionen von DB2 werden möglicherweise das ältere Format für die XA-Zeichenfolge zum Öffnen nicht unterstützen. Informationen zum ursprünglichen Format der XA-Zeichenfolge zum Öffnen finden Sie in „Format der XA-Zeichenfolge zum Öffnen für frühere Versionen von DB2“ auf Seite 209.

Bei der Einrichtung einer Datenbank als Ressourcenmanager ist keine XA-Zeichenfolge zum Schließen (`xa_close`) erforderlich. Falls eine solche Zeichenfolge angegeben wird, wird sie vom Datenbankmanager ignoriert.

Neues Format der XA-Zeichenfolge zum Öffnen für DB2 Version 7

Das folgende XA-Zeichenfolgenformat wird mit DB2 Version 7 neu eingeführt:

```
parm_id1 = <parm-wert>, parm_id2 = <parm-wert>, ...
```

Es spielt keine Rolle, in welcher Reihenfolge diese Parameter angegeben werden. Die gültigen Werte für *parm_id* werden in der folgenden Tabelle beschrieben.

Tabelle 22. Gültige Werte für *parm_id*

Parametername	Wert	Verbindlich?	Groß-/Kleinschreibung beachtet?	Standardwert
DB	Aliasname der Datenbank	Ja	Nein	Kein

Tabelle 22. Gültige Werte für parm_id (Forts.)

Parametername	Wert	Verbindlich?	Groß- /Kleinschreibung beachtet?	Standardwert
Der Aliasname der Datenbank, der von der Anwendung für den Zugriff auf die Datenbank verwendet wird.				
UID	Benutzer-ID	Nein	Ja	Kein
Eine Benutzer-ID mit Berechtigung zur Verbindung mit der Datenbank. Erforderlich, wenn ein Kennwort angegeben wird.				
PWD	Kennwort	Nein	Ja	Kein
Ein zur Benutzer-ID gehöriges Kennwort. Erforderlich, wenn eine Benutzer-ID angegeben wird.				
TPM	Name des TP-Monitors	Nein	Nein	Kein
Name des verwendeten TP-Monitors. Unterstützte Werte siehe „Werte für TPM und TP_MON_NAME“ auf Seite 206. Dieser Parameter kann angegeben werden, um mehreren TP-Monitoren die Verwendung eines einzigen DB2-Exemplars zu ermöglichen. Der angegebene Wert überschreibt den im Konfigurationsparameter <i>tp_mon_name</i> des Datenbankmanagers definierten Wert.				
AXLIB	Bibliothek, die die Funktionen ax_reg und ax_unreg des TP-Monitors enthält	Nein	Ja	Kein
Dieser Wert wird von DB2 zum Abrufen der Adressen der erforderlichen Funktionen ax_reg und ax_unreg verwendet. Er kann genutzt werden, um aufgrund des Parameters TPM angenommene Werte zu überschreiben, oder er kann von TP-Monitoren genutzt werden, die nicht in der Liste für TPM aufgeführt sind.				
CHAIN_END	xa_end- Kettungsmarkierung. Gültige Werte sind T, F oder kein Wert.	Nein	Nein	F
Die XA_END-Kettung ist eine Optimierungsmaßnahme, die von DB2 zur Verringerung von Netzwerkübertragungen genutzt werden kann. Wenn die TP-Monitorumgebung so eingerichtet ist, daß gewährleistet werden kann, daß xa_prepare innerhalb desselben Thread oder Prozesses unmittelbar nach einem Aufruf von xa_end aufgerufen wird und CHAIN_END aktiviert ist, wird die xa_end-Markierung mit dem Befehl xa_prepare verkettet, um so eine Netzwerkübertragung einzusparen. Der Wert T bedeutet, daß CHAIN_END aktiviert ist. Der Wert F bedeutet, daß CHAIN_END inaktiv ist. Kein Wert bedeutet, daß CHAIN_END aktiviert ist. Dieser Parameter kann dazu dienen, die aus einem angegebenen TPM-Wert abgeleitete Einstellung zu überschreiben.				

Tabelle 22. Gültige Werte für parm_id (Forts.)

Parametername	Wert	Verbindlich?	Groß- /Kleinschreibung beachtet?	Standardwert
SUSPEND_CURSOR	Gibt an, ob Cursor beizubehalten sind, wenn ein Transaktions-Thread der Steuerung angehalten wird. Gültige Werte sind T, F oder kein Wert.	Nein	Nein	F
<p>TP-Monitore, die eine Transaktionsverzweigung anhalten, können den angehaltenen Thread oder Prozeß für andere Transaktionen wiederverwenden. In diesen Fällen müssen Cursor geschlossen werden, so daß die neue Transaktion diese nicht übernimmt. Wenn eine angehaltene Transaktion wieder aufgenommen wird, muß die Anwendung den Cursor erneut abrufen. Wenn SUSPEND_CURSOR aktiv ist, werden keine geöffneten Cursor geschlossen, jedoch kann der Thread oder Prozeß nicht für andere Transaktionen wiederverwendet werden. Es ist nur die Wiederaufnahme der unterbrochenen Transaktion zulässig. Der Wert T bedeutet, daß SUSPEND_CURSOR aktiviert ist. Der Wert F bedeutet, daß SUSPEND_CURSOR inaktiv ist. Kein angegebener Wert bedeutet, daß SUSPEND_CURSOR aktiviert ist. Dieser Parameter kann dazu dienen, die aus einem angegebenen TPM-Wert abgeleitete Einstellung zu überschreiben.</p>				
HOLD_CURSOR	Gibt an, ob Cursor über Fest-schreibungen von Transaktionen hinweg beibehalten werden. Gültige Werte sind T, F oder kein Wert.	Nein	Nein	F
<p>TP-Monitore verwenden Threads oder Prozesse in der Regel für mehrere Anwendungen wieder. Um sicherzustellen, daß neu geladene Anwendungen keine von einer früheren Anwendung geöffneten Cursor übernehmen, werden Cursor nach einer Festschreibung (COMMIT) geschlossen. Wenn HOLD_CURSOR aktiviert ist, werden Cursor über Festschreibungen von Transaktionen hinweg beibehalten. Der Wert T bedeutet, daß HOLD_CURSOR aktiviert ist. Der Wert F bedeutet, daß HOLD_CURSOR inaktiv ist. Kein angegebener Wert bedeutet, daß HOLD_CURSOR aktiviert ist. Dieser Parameter kann dazu dienen, die aus einem angegebenen TPM-Wert abgeleitete Einstellung zu überschreiben.</p>				

Werte für TPM und TP_MON_NAME

Der Parameter TPM für die XA-Zeichenfolge zum Öffnen (xa_open) und der Konfigurationsparameter *tp_mon_name* des Datenbankmanagers dienen dazu, für DB2 zu definieren, welcher TP-Monitor verwendet wird. Der Wert für *tp_mon_name* gilt für das gesamte DB2-Exemplar. Der Parameter TPM gilt nur für den speziellen XA-Ressourcenmanager. Der Wert für TPM überschreibt den Parameter *tp_mon_name*. Die folgenden Werte für die Parameter TPM und *tp_mon_name* sind gültig:

Tabelle 23. Gültige Werte für TPM und *tp_mon_name*

TPM-Wert	TP-Monitor- produkt	Interne Einstellungen
CICS	IBM TxSeries CICS	AXLIB=libEncServer (für Windows) =usr/lpp/encina/lib/libEncServer (für auf UNIX basierende Systeme) HOLD_CURSOR=T CHAIN_END=T SUSPEND_CURSOR=F
ENCINA	IBM TxSeries Encina Monitor	AXLIB=libEncServer (für Windows) =usr/lpp/encina/lib/libEncServer (für auf UNIX basierende Systeme) HOLD_CURSOR=F CHAIN_END=T SUSPEND_CURSOR=F
MQ	IBM MQSeries	AXLIB=mqmax (für Windows) =usr/mqm/lib/libmqmax.a (für AIX) =opt/mqm/lib/libmqmax.a (für Solaris) HOLD_CURSOR=F CHAIN_END=F SUSPEND_CURSOR=F
CB	IBM Component Broker	AXLIB=somtrxli (für Windows) =libsomtrxl (für auf UNIX basierende Systeme) HOLD_CURSOR=F CHAIN_END=T SUSPEND_CURSOR=F
SF	IBM San Francisco	AXLIB=ibmsfdb2 HOLD_CURSOR=F CHAIN_END=T SUSPEND_CURSOR=F
TUXEDO	BEA Tuxedo	AXLIB=libtux HOLD_CURSOR=F CHAIN_END=F SUSPEND_CURSOR=F

Tabelle 23. Gültige Werte für TPM und *tp_mon_name* (Forts.)

TPM-Wert	TP-Monitor- produkt	Interne Einstellungen
MTS	Microsoft Transaction Server	Es ist nicht nötig, DB2 für MTS zu konfigurieren. MTS wird vom ODBC-Treiber von DB2 automatisch erkannt.
JTA	Java Transaction API	Es ist nicht nötig, DB2 für Enterprise Java Servers (EJS) wie IBM WebSphere zu konfigurieren. Der JDBC-Treiber von DB2 erkennt diese Umgebung automatisch.

Beispiele

1. Sie verwenden IBM TxSeries CICS unter Windows NT. Der Dokumentation zu TxSeries ist zu entnehmen, daß Sie den Parameter *tp_mon_name* mit dem Wert `libEncServer:C` konfigurieren müssen. Dies ist immer noch ein akzeptables Format. Allerdings haben Sie mit DB2 UDB oder DB2 Connect Version 7 folgende Auswahl:

- Angeben des Wertes CICS für den Parameter *tp_mon_name* (empfohlen für dieses Szenario):

```
db2 update dbm cfg using tp_mon_name CICS
```

Für jede Datenbank, die in der Initialisierungszeichenfolge Region-> Resources-> Product-> XAD-> Resource manager für CICS definiert ist, geben Sie folgendes an:

```
db=dbalias,uid=benutzer-id,pwd=kennwort
```

- Für jede Datenbank, die in der Initialisierungszeichenfolge Region-> Resources-> Product-> XAD-> Resource manager für CICS definiert ist, geben Sie folgendes an:

```
db=dbalias,uid=benutzer-id,pwd=kennwort,tpm=cics
```

2. Sie verwenden IBM MQSeries unter Windows NT. Der Dokumentation zu MQSeries ist zu entnehmen, daß Sie den Parameter *tp_mon_name* mit dem Wert `mqmax` konfigurieren müssen. Dies ist immer noch ein akzeptables Format. Allerdings haben Sie mit DB2 UDB oder DB2 Connect Version 7 folgende Auswahl:

- Angeben des Wertes MQ für den Parameter *tp_mon_name* (empfohlen für dieses Szenario):

```
db2 update dbm cfg using tp_mon_name MQ
```

Für jede Datenbank, die in der Initialisierungszeichenfolge Region-> Resources-> Product-> XAD-> Resource manager für CICS definiert ist, geben Sie folgendes an:

```
uid=benutzer-id,db=dbalias,pwd=kennwort
```

- Für jede Datenbank, die in der Initialisierungszeichenfolge Region-> Resources-> Product-> XAD-> Resource manager für CICS definiert ist, geben Sie folgendes an:

```
uid=benutzer-id,db=dbalias,pwd=kennwort,tpm=mq
```

3. Sie verwenden sowohl IBM TxSeries CICS als auch IBM MQSeries unter Windows NT. Es wird ein einziges DB2-Exemplar verwendet. In diesem Szenario würden Sie wie folgt konfigurieren:
 - a. Für jede Datenbank, die in der Initialisierungszeichenfolge Region-> Resources-> Product-> XAD-> Resource manager für CICS definiert ist, geben Sie folgendes an:


```
pwd=kennwort,uid=benutzer-id,tpm=cics,db=dbalias
```
 - b. Für jede Datenbank, die als Ressource in den Eigenschaften des Warteschlangenmanagers definiert ist, geben Sie eine XA-Zeichenfolge zum Öffnen wie folgt an:


```
db=dbalias,uid=benutzer-id,pwd=kennwort,tpm=mq
```
4. Sie entwickeln einen eigenen XA-konformen Transaktionsmanager (XA TM) unter Windows NT, und Sie wollen DB2 mitteilen, daß die Bibliothek "meinaxlib" die erforderlichen Funktionen **ax_reg** und **ax_unreg** enthält. Die Bibliothek "meinaxlib" befindet sich in einem in der Anweisung PATH definierten Verzeichnis. Sie haben die folgende Auswahl:

- Angeben des Wertes *meinaxlib* für den Parameter *tp_mon_name*:

```
db2 update dbm cfg using tp_mon_name meinaxlib
```

Und für jede Datenbank, die für den XA TM definiert ist, Angeben einer XA-Zeichenfolge zum Öffnen:

```
db=dbalias,uid=benutzer-id,pwd=kennwort
```

- Für jede Datenbank, die für den XA TM definiert ist, Angeben einer XA-Zeichenfolge zum Öffnen:

```
db=dbalias,uid=benutzer-id,pwd=kennwort,axlib=meinaxlib
```

5. Sie entwickeln einen eigenen XA-konformen Transaktionsmanager (XA TM) unter Windows NT, und Sie wollen DB2 mitteilen, daß die Bibliothek "meinaxlib" die erforderlichen Funktionen **ax_reg** und **ax_unreg** enthält. Die Bibliothek "meinaxlib" befindet sich in einem in der Anweisung PATH definierten Verzeichnis. Außerdem wollen Sie die XA-END-Kettung aktivieren. Sie haben die folgende Auswahl:

- Für jede Datenbank, die für den XA TM definiert ist, Angeben einer XA-Zeichenfolge zum Öffnen:

```
db=dbalias,uid=benutzer-id,pwd=kennwort,axlib=meinaxlib,chain_end=T
```

- Für jede Datenbank, die für den XA TM definiert ist, Angeben einer XA-Zeichenfolge zum Öffnen:

```
db=dbalias,uid=benutzer-id,pwd=kennwort,axlib=meinaxlib,chain_end
```

Format der XA-Zeichenfolge zum Öffnen für frühere Versionen von DB2

Frühere Versionen von DB2 arbeiten mit dem im folgenden beschriebenen Format der XA-Zeichenfolge zum Öffnen (*xa_open*). Dieses Format wird aus Gründen der Kompatibilität weiterhin unterstützt. Anwendungen sollten nach Möglichkeit auf das neue Format (siehe „Neues Format der XA-Zeichenfolge zum Öffnen für DB2 Version 7“ auf Seite 203) umgestellt werden.

Jede Datenbank wird als ein separater Ressourcenmanager (RM) beim Transaktionsmanager (TM) definiert, und die Datenbank muß durch eine XA-Zeichenfolge zum Öffnen mit der folgenden Syntax identifiziert werden:

```
"db_alias<,benutzer-id,kennwort>"
```

Der *db_alias* ist erforderlich, um den Aliasnamen der Datenbank anzugeben. Der Aliasname ist derselbe Name wie der Datenbankname, sofern Sie nicht explizit einen Aliasnamen nach der Erstellung der Datenbank katalogisiert haben. Der Benutzername und das Kennwort sind wahlfrei und werden je nach Methode zur Identifikationsüberprüfung verwendet, um Informationen zur Identifikationsüberprüfung an die Datenbank weiterzugeben.

Bei der Einrichtung einer Datenbank als Ressourcenmanager ist keine XA-Zeichenfolge zum Schließen (*xa_close*) erforderlich. Falls eine solche Zeichenfolge angegeben wird, wird sie vom Datenbankmanager ignoriert.

Aktualisieren von Host- oder AS/400-Datenbank-Servern

Host- und AS/400-Datenbank-Server können aktualisierbar sein, je nach Architektur des XA-Transaktionsmanagers. Zur Unterstützung von Festschreibungssequenzen aus verschiedenen Prozessen muß der DB2 Connect-Konzentrator aktiviert werden. Zur Aktivierung des DB2 Connect EE-Konzentrators setzen Sie den Konfigurationsparameter *max_logicagents* des Datenbankmanagers auf einen Wert, der größer ist als *maxagents*. Beachten Sie, daß für den DB2 Connect EE-Konzentrator ein Client mit DB2 Version 7.1 zur Unterstützung von XA-Festschreibungssequenzen aus verschiedenen Prozessen erforderlich ist. Informationen zu SQL-Anweisungen, die in dieser Umgebung zulässig sind, finden Sie im Handbuch *Application Development Guide*. Informationen zum Konzentration finden Sie im *DB2 Connect Benutzerhandbuch*.

Wenn Sie die Host- oder AS/400-Datenbank-Server aktualisieren wollen, benötigen Sie DB2 Connect mit konfigurierbarem DB2-Synchronisationspunktmanager (SPM). Anweisungen finden Sie in einem der Handbücher *Einstieg*.

Überlegungen zu Datenbankverbindungen

Dieser Abschnitt behandelt die folgenden Themen:

- „Anweisung RELEASE“
- „Transaktionen, die auf partitionierte Datenbanken zugreifen“

Anweisung RELEASE

Wenn die Anweisung RELEASE verwendet wurde, um die Verbindung zu einer Datenbank freizugeben, sollte die Anweisung CONNECT und nicht die Anweisung SET CONNECTION zur erneuten Herstellung der Verbindung zu dieser Datenbank verwendet werden.

Transaktionen, die auf partitionierte Datenbanken zugreifen

In einer Umgebung mit partitionierten Datenbanken können Benutzerdaten auf mehrere Datenbankpartitionen verteilt werden. Eine Anwendung, die auf eine solche Datenbank zugreift, stellt die Verbindung her und sendet Anforderungen an eine der Partitionen der Datenbank (den Koordinatorknoten). Verschiedene Anwendungen können die Verbindung zu verschiedenen Datenbankpartitionen herstellen, und dieselbe Anwendung kann verschiedene Datenbankpartitionen für verschiedene Verbindungen auswählen.

Für Transaktionen, die an einer Datenbank in einer Umgebung mit partitionierten Datenbanken ausgeführt werden, muß der gesamte Zugriff über *gleiche* Datenbankpartition erfolgen. Das bedeutet, daß die gleiche Datenbankpartition vom Beginn der Transaktion bis zu dem Zeitpunkt (einschließlich) verwendet werden muß, zu dem die Transaktion festgeschrieben wird.

Jede Transaktion für die partitionierte Datenbank muß festgeschrieben werden, bevor die Verbindung getrennt wird.

Treffen einer heuristischen Entscheidung

Ein dem XA-Standard entsprechender Transaktionsmanager arbeitet mit einem ähnlichen zweiphasigen Festschreibeprozess wie der DB2-Transaktionsmanager, der in „Prozess der zweiphasigen Festschreibung“ auf Seite 190 beschrieben wird. Der Hauptunterschied zwischen diesen beiden Umgebungen besteht darin, daß hier der TP-Monitor die Funktionen zur Protokollierung und Steuerung zur Verfügung stellt, und nicht der DB2-Transaktionsmanager und die Transaktionsmanagerdatenbank.

Bei der Verwendung eines dem XA-Standard entsprechenden Transaktionsmanagers können Fehler auftreten, die den für den DB2-Transaktionsmanager beschriebenen Fehlern ähnlich sind (siehe „Beheben von Problemen bei der

zweiphasigen Festschreibung" auf Seite 193). Ähnlich wie der DB2-Transaktionsmanager versucht auch ein XA-Transaktionsmanager unbestätigte Transaktionen zu resynchronisieren.

Wenn Sie aus einem bestimmten Grund nicht darauf warten können, daß der Transaktionsmanager unbestätigte Transaktionen automatisch auflöst, haben Sie die Möglichkeit, Maßnahmen zur manuellen Auflösung unbestätigter Transaktionen zu ergreifen. Dieser manuelle Prozeß wird manchmal als das „Treffen einer heuristischen Entscheidung" bezeichnet.

Der Befehl LIST INDOUBT TRANSACTIONS (mit der Option WITH PROMPTING) bzw. eine entsprechende Gruppe von APIs ermöglicht es, unbestätigte Transaktionen abzufragen, festzuschreiben (COMMIT) oder rückgängig zu machen (ROLLBACK). Darüber hinaus können mit diesem Befehl Transaktionen ignoriert werden („FORGET"), die heuristisch festgeschrieben oder rückgängig gemacht wurden, indem die Protokollsätze entfernt und der Protokollspeicherbereich freigegeben werden. Zum Abfragen von Informationen über unbestätigte Transaktionen aus DB2 UDB auf UNIX-Systemen, auf dem Windows-Betriebssystem oder OS/2, stellen Sie eine Verbindung zu der Datenbank her und setzen den Befehl LIST INDOUBT TRANSACTIONS WITH PROMPTING ab bzw. verwenden die entsprechende API. Weitere Informationen zu diesem Befehl oder zu den entsprechenden administrativen APIs finden Sie im Handbuch *Command Reference* bzw. *Administrative API Reference*.

Zum Abrufen von Informationen zu unbestätigten Transaktionen in bezug auf Host- oder AS/400-Datenbank-Server stehen zwei Möglichkeiten zur Verfügung:

- Sie können Informationen zu unbestätigten Transaktionen direkt vom Host- oder AS/400-Server abrufen.

Zum Abrufen von Informationen zu unbestätigten Transaktionen direkt aus DB2 für OS/390 rufen Sie den Befehl DISPLAY THREAD TYPE(INDOUBT) auf. Mit Hilfe des Befehls RECOVER können Sie eine heuristische Entscheidung treffen. Zum direkten Abrufen von Informationen zu unbestätigten Transaktionen aus DB2 für OS/400 rufen Sie den Befehl **wrkcmtdfn** auf.

- Sie können Informationen zu unbestätigten Transaktionen direkt aus dem DB2 Connect-Server abrufen, der für den Zugriff auf den Host- oder AS/400-Datenbank-Server verwendet wird.

Zum Abrufen von Informationen zu unbestätigten Transaktionen vom DB2 Connect-Server stellen Sie zunächst eine Verbindung zum DB2-Synchronisationspunktmanager her, indem Sie eine Verbindung zu dem DB2-Exemplar herstellen, das durch den Wert des Konfigurationsparameters *spm_name* des Datenbankmanagers definiert ist. Anschließend setzen

Sie den Befehl LIST DRDA INDOUBT TRANSACTIONS WITH PROMPTING ab, um unbestätigte Transaktionen anzuzeigen und heuristische Entscheidungen zu treffen.

Verwenden Sie diese Befehle (bzw. die entsprechenden APIs) nur mit *äußerster Vorsicht* und nur als allerletzte Möglichkeit. Die beste Strategie ist, zu warten, bis der Transaktionsmanager den Prozeß der Resynchronisation durchgeführt hat. Es kann zu Problemen mit der Datenintegrität kommen, wenn Sie eine Transaktion in einer der beteiligten Datenbanken manuell festschreiben oder rückgängig machen und die entgegengesetzte Maßnahme für eine andere beteiligte Datenbank durchgeführt wird. Die Behebung von Problemen der Datenintegrität setzt voraus, daß Sie die Logik der Anwendung und die geänderten oder rückgängig gemachten Daten kennen, und Sie dann eine Wiederherstellung bis zu einem bestimmten Zeitpunkt durchführen bzw. die Datenbankänderungen manuell rückgängig machen oder erneut anwenden.

Wenn Sie nicht darauf warten können, bis der Transaktionsmanager den Prozeß zur Resynchronisation initialisiert, und die Ressourcen, die von einer unbestätigten Transaktion blockiert werden, freigeben müssen, sind heuristische Maßnahmen erforderlich. Diese Situation kann auftreten, wenn der Transaktionsmanager für einen längeren Zeitraum zur Durchführung der Resynchronisation nicht verfügbar ist und die unbestätigte Transaktion Ressourcen sperrt, die dringend benötigt werden. Eine unbestätigte Transaktion blockiert Ressourcen, die dieser Transaktion zugeordnet wurden, als der Transaktionsmanager oder die Ressourcenmanager noch verfügbar waren. Im Fall des Datenbankmanagers gehören zu diesen Ressourcen zum Beispiel die Sperren für Tabellen und Indizes, der Protokollspeicherbereich und der Speicher, der von der Transaktion belegt wird. Jede unbestätigte Transaktion verringert außerdem die maximale Anzahl gleichzeitiger Transaktionen (um den Wert 1), die von der Datenbank verarbeitet werden können.

Obwohl es keine absolut sichere Methode zur Durchführung heuristischer Maßnahmen gibt, werden im folgenden einige allgemeine Richtlinien beschrieben:

1. Stellen Sie die Verbindung zu der Datenbank her, für die alle Transaktionen abgeschlossen werden sollen.
2. Verwenden Sie den Befehl LIST INDOUBT TRANSACTIONS, um die unbestätigten Transaktionen anzuzeigen. Die *xid* stellt die ID der globalen Transaktion dar und ist mit der *xid* identisch, die vom Transaktionsmanager und von anderen Ressourcenmanagern, die an der Transaktion beteiligt sind, verwendet wird.
3. Bestimmen Sie mit Hilfe Ihrer Kenntnisse über die Anwendung und die Betriebsumgebung für jede unbestätigte Transaktion die anderen beteiligten Ressourcenmanager.
4. Stellen Sie fest, ob der Transaktionsmanager verfügbar ist:

- Wenn der Transaktionsmanager verfügbar ist und die unbestätigte Transaktion in einem Ressourcenmanager dadurch verursacht wurde, daß der Ressourcenmanager in der zweiten Phase der Festschreibung nicht verfügbar war, oder für einen früheren Resynchronisationsprozeß, sollten Sie das Protokoll des Transaktionsmanagers überprüfen, um festzustellen, welche Aktion für die anderen Ressourcenmanager durchgeführt wurde. Anschließend sollten Sie dieselbe Aktion für die Datenbank durchführen, d. h. mit dem Befehl LIST INDOUBT TRANSACTIONS die Transaktion entweder heuristisch festschreiben (COMMIT) oder heuristisch rückgängig machen (ROLLBACK).
- Wenn der Transaktionsmanager *nicht* verfügbar ist, müssen Sie den Status der Transaktion in den anderen beteiligten Ressourcenmanagern verwenden, um festzustellen, welche Aktion durchzuführen ist:
 - Wenn mindestens einer der anderen Ressourcenmanager die Transaktion festgeschrieben hat, dann sollten Sie die Transaktion in allen anderen Ressourcenmanagern heuristisch festschreiben (COMMIT).
 - Wenn mindestens einer der anderen Ressourcenmanager die Transaktion rückgängig gemacht hat, sollten Sie die Transaktion heuristisch rückgängig machen (ROLLBACK).
 - Wenn die Transaktion den Status "prepared" (unbestätigt) in allen beteiligten Ressourcenmanagern hat, sollten Sie die Transaktion heuristisch rückgängig machen.
 - Wenn einer oder mehrere der anderen Ressourcenmanager nicht verfügbar sind, sollten Sie die Transaktion heuristisch rückgängig machen.

Führen Sie die heuristische Funktion FORGET nicht aus, sofern nicht eine heuristisch festgeschriebene oder rückgängig gemachte Transaktion zu der Bedingung führt, daß das Protokoll voll ist, die in der Ausgabe des Befehls LIST INDOUBT TRANSACTIONS angezeigt wird. Die heuristische Funktion FORGET gibt den Protokollspeicherbereich frei, der durch eine unbestätigte Transaktion belegt wird. Dies könnte zur Folge haben, daß, wenn ein Transaktionsmanager schließlich einen Resynchronisationsprozeß für diese unbestätigte Transaktion durchführt, er eine falsche Entscheidung, andere Ressourcenmanager festzuschreiben oder rückgängig zu machen, treffen könnte, weil für die Transaktion in diesem Ressourcenmanager kein Protokollsatz mehr vorhanden wäre. Allgemein impliziert ein „fehlender“ Protokollsatz, daß der Ressourcenmanager die Transaktion rückgängig gemacht hat.

Überlegungen zur Sicherheit

Der TP-Monitor ordnet eine Gruppe von Server-Prozessen im voraus zu und führt die Transaktionen von verschiedenen Benutzern unter den IDs der Server-Prozesse aus. Für die Datenbank erscheint jeder Server-Prozeß wie eine große Anwendung, die viele Arbeitseinheiten vereinigt, die alle unter derselben, dem Server-Prozeß zugeordneten ID ausgeführt werden.

Wenn zum Beispiel in einer AIX-Umgebung mit CICS eine TXSeries CICS-Region gestartet wird, wird ihr der AIX-Benutzername zugeordnet, unter dem sie definiert ist. Alle CICS-Anwendungs-Server-Prozesse werden ebenfalls unter dieser TXSeries CICS-Haupt-ID ausgeführt, die in der Regel als "cics" definiert ist. CICS-Benutzer können CICS-Transaktionen unter ihrer DCE-Anmelde-ID aufrufen und, während sie in CICS arbeiten, ihre ID mit Hilfe der CESN-Anmeldetransaktion auch ändern. In beiden Fällen ist für den Ressourcenmanager die Endbenutzer-ID nicht verfügbar. Folglich kann ein CICS-Anwendungsprozeß Transaktionen für viele Benutzer ausführen, aber sie erscheinen dem Ressourcenmanager wie ein einziges Programm mit vielen Arbeitseinheiten von derselben ID "cics". Wahlfrei können Sie eine Benutzer-ID und ein Kennwort in der XA-Zeichenfolge zum Öffnen angeben, so daß diese Benutzer-ID anstelle der ID "cics" für die Verbindung mit der Datenbank verwendet wird.

Die Auswirkungen sind für statische SQL-Anweisungen nicht groß, da zum Zugriff auf die Datenbank die Zugriffsrechte der Person, die das Paket gebunden hat, und nicht die Zugriffsrechte des Endbenutzers verwendet werden. Das bedeutet aber, daß das Zugriffsrecht EXECUTE der Datenbankpakete der Server-ID, und nicht der Endbenutzer-ID erteilt werden muß.

Für dynamische Anweisungen, für deren Zugriff die Authentifizierung zur Laufzeit erfolgt, müssen die Zugriffsrechte der Datenbankobjekte der Server-ID, und nicht dem tatsächlichen Benutzer dieser Objekte erteilt werden. Anstatt sich bei der Steuerung des Zugriffs bestimmter Benutzer auf die Datenbank zu stützen, müssen Sie über das TP-Monitorssystem festlegen, welche Benutzer welche Programme ausführen können. Der Server-ID müssen alle Zugriffsrechte erteilt werden, die für die SQL-Benutzer erforderlich sind.

Um festzustellen, wer auf eine Datenbanktabelle oder -sicht zugegriffen hat, können Sie folgende Schritte unternehmen:

1. Erstellen Sie sich aus der Katalogsicht SYSCAT.PACKAGEDEP eine Liste aller Pakete, die von der Tabelle oder Sicht abhängig sind.
2. Bestimmen Sie die Namen der Server-Programme (z. B. CICS-Programme), die diesen Paketen aufgrund der in Ihrer Installation verwendeten Namenskonvention entsprechen.
3. Bestimmen Sie die Client-Programme (z. B. CICS-Transaktions-IDs), die diese Programme aufrufen konnten, und verwenden Sie anschließend das Protokoll des TP-Monitors (z. B. das CICS-Protokoll), um festzustellen, wer diese Transaktionen oder Programme wann ausgeführt hat.

Überlegungen zur Konfiguration

Bei der Einrichtung der TP-Monitorumgebung sollten die folgenden Konfigurationsparameter berücksichtigt werden:

- *tp_mon_name*

Mit diesem Konfigurationsparameter des Datenbankmanagers wird der Name des verwendeten TP-Monitorprodukts (z. B. "CICS" oder "ENCINA") angegeben.

- *tpname*

Mit diesem Konfigurationsparameter des Datenbankmanagers wird der Name des fernen Transaktionsprogramms angegeben, das der Datenbank-Client verwenden muß, wenn er eine Zuordnungsanforderung an den Datenbank-Server über das APPC-Übertragungsprotokoll absetzt. Der Wert wird in der Konfigurationsdatei auf dem Server definiert und muß mit dem im SNA-Transaktionsprogramm konfigurierten TP-Namen für den Transaktionsprozess übereinstimmen. Weitere Informationen finden Sie in den Handbüchern *Einstieg*.

- *tm_database*

Da DB2 Transaktionen in der XA-Umgebung *nicht* koordiniert, wird dieser Konfigurationsparameter des Datenbankmanagers für XA-koordinierte Transaktionen nicht verwendet.

- *maxappls*

Mit diesem Konfigurationsparameter der Datenbank wird die zulässige maximale Anzahl aktiver Anwendungen angegeben. Der Wert dieses Parameters muß gleich oder größer sein als die Summe der verbundenen Anwendungen plus die Anzahl dieser Anwendungen, die sich gleichzeitig im Prozeß einer zweiphasigen Festschreibung bzw. einer ROLLBACK-Operation befinden können. Diese Summe sollte dann um die vorab geschätzte Anzahl unbestätigter Transaktionen erhöht werden, die zu einem beliebigen Zeitpunkt gleichzeitig vorhanden sein könnten. Weitere Informationen zu unbestätigten Transaktionen finden Sie in „Beheben von Problemen bei der zweiphasigen Festschreibung“ auf Seite 193.

Für eine TP-Monitorumgebung (z. B. TXSeries CICS) müssen Sie den Wert des Parameters *maxappls* erhöhen. Dies hilft bei der Sicherstellung, daß alle TP-Monitorprozesse verarbeitet werden können.

- *autorestart*

Mit diesem Konfigurationsparameter der Datenbank wird festgelegt, ob die Routine RESTART DATABASE bei Bedarf aufgerufen wird. Der Standardwert ist YES (d. h. aktiviert).

Für eine Datenbank, die unbestätigte Transaktionen enthält, ist eine RESTART DATABASE-Operation für den Neustart erforderlich. Wenn *autorestart* nicht aktiviert ist und die letzte Verbindung zur Datenbank getrennt wurde, schlägt die nächste Verbindungsanforderung fehl, so daß ein explizites Aufrufen des Befehls RESTART DATABASE erforderlich wird. Diese Bedingung bleibt solange bestehen, bis die unbestätigten Transaktionen entweder durch die Resynchronisationsoperation des Transaktionsmanagers oder durch eine vom Administrator eingeleitete heuristische Operation beseitigt wird. Wenn der Befehl RESTART DATABASE abgesetzt wird, wird eine

Nachricht zurückgegeben, wenn unbestätigte Transaktionen in der Datenbank vorhanden sind. Der Administrator kann dann mit Hilfe des Befehls LIST INDOUBT TRANSACTIONS und anderer Befehle des Befehlszeilenprozessors Informationen über diese unbestätigten Transaktionen erhalten.

Unterstützte XA-Funktion

DB2 Universal Database unterstützt die Spezifikation XA91, die in *X/Open CAE Specification Distributed Transaction Processing: The XA Specification* definiert ist, mit folgenden Ausnahmen:

- **Asynchrone Services**

Die XA-Spezifikation ermöglicht der Schnittstelle die Verwendung asynchroner Services, so daß das Ergebnis einer Anforderung zu einem späteren Zeitpunkt überprüft werden kann. Für den Datenbankmanager müssen die Anforderungen im synchronen Modus aufgerufen werden.

- **Statische Registrierung**

Die XA-Schnittstelle ermöglicht zwei Methoden zur Registrierung eines Ressourcenmanagers: statische und dynamische Registrierung. DB2 Universal Database unterstützt nur die dynamische Registrierung, die ausgereifter und effizienter ist. Weitere Informationen zu diesen beiden Methoden finden Sie in „Ressourcenmanager (RM)“ auf Seite 201.

- **Migration von Zuordnungen**

DB2 Universal Database unterstützt die Transaktionsmigration zwischen Threads der Steuerung nicht.

Informationen zur Verwendung der XA-Zeichenfolgen zum Öffnen und zum Schließen (xa_open und xa_close) finden Sie in „Verwenden der XA-Zeichenfolgen zum Öffnen und Schließen“ auf Seite 203.

Syntax und Position der XA-Schalter

Wie für die XA-Schnittstelle erforderlich stellt der Datenbankmanager eine externe C-Variable *db2xa_switch* des Typs *xa_switch_t* bereit, um die XA-Schalterstruktur an den Transaktionsmanager zurückzugeben. Neben den Adressen verschiedener XA-Funktionen werden folgende Felder zurückgegeben:

Feld	Wert
name	Der Produktname des Datenbankmanagers. Zum Beispiel: DB2 für AIX.
flags	TMREGISTER TMNOMIGRATE Gibt explizit an, daß DB2 Universal Database die dynamische Registrierung verwendet und daß der TM keine Migration von Zuordnungen verwenden soll. Gibt implizit an, daß ein asynchroner Betrieb nicht unterstützt wird.
version	Muß null sein.

Verwenden des XA-Schalters von DB2 Universal Database

Die XA-Architektur verlangt, daß ein Ressourcenmanager (RM) einen *Schalter* bereitstellt, der dem XA-Transaktionsmanager (TM) Zugriff auf die *xa_*-Routinen des Ressourcenmanagers gibt. Ein RM-Schalter verwendet eine Struktur, die als *xa_switch_t* bezeichnet wird. Der Schalter enthält den Namen des RMs, Nicht-NULL-Zeiger auf die XA-Eingangspunkte des RMs, eine Markierung (Flag) und eine Versionsnummer.

Auf UNIX basierende Systeme und OS/2: Der DB2 UDB-Schalter kann durch eine der folgenden Methoden abgerufen werden:

- Über eine weitere Zwischenstufe. In einem C-Programm kann dies durch Definieren des Makros:

```
#define db2xa_switch (*db2xa_switch)
```

vor Verwendung von *db2xa_switch* erreicht werden.

- Durch Aufrufen von **db2xacic**

DB2 UDB stellt diese API zur Verfügung, die die Adresse der *db2xa_switch*-Struktur liefert. Der Prototyp dieser Funktion lautet:

```
struct xa_switch_t * SQL_API_FN db2xacic( )
```

Bei beiden Methoden müssen Sie die Anwendung mit `libdb2` (bei auf UNIX basierenden Systemen) oder `db2api.lib` (unter OS/2) verbinden ("linken").

Windows NT: Der Zeiger auf die *xa_switch*-Struktur, *db2xa_switch*, wird in Form von DLL-Daten exportiert (d. h. bereitgestellt). Dies heißt für eine Anwendung unter Windows NT, die diese Struktur verwendet, daß sie auf die Struktur mit Hilfe einer von drei Methoden zugreifen muß:

- Über eine weitere Zwischenstufe. In einem C-Programm kann dies durch Definieren des Makros:

```
#define db2xa_switch (*db2xa_switch)
```

vor Verwendung von *db2xa_switch* erreicht werden.

- Bei Verwendung des Microsoft Visual C++-Compilers kann *db2xa_switch* folgendermaßen definiert werden:

```
extern __declspec(dllimport) struct xa_switch_t db2xa_switch
```

- Durch Aufrufen von **db2xacic**

DB2 UDB stellt diese API zur Verfügung, die die Adresse der *db2xa_switch*-Struktur liefert. Der Prototyp dieser Funktion lautet:

```
struct xa_switch_t * SQL_API_FN db2xacic( )
```

Bei jeder dieser Methoden müssen Sie die Anwendung mit `db2api.lib` verbinden ("linken").

C-Beispielcode: Der folgende Code veranschaulicht die verschiedenen Methoden des Zugriffs auf *db2xa_switch* über ein C-Programm auf einer beliebigen DB2 UDB-Plattform. Stellen Sie sicher, daß die Anwendung mit der entsprechenden Bibliothek verbunden wird.

```
#include <stdio.h>
#include <xa.h>

struct xa_switch_t * SQL_API_FN db2xacic( );

#ifdef DECLSPEC_DEFN
extern __declspec(dllimport) struct xa_switch_t db2xa_switch;
#else
#define db2xa_switch (*db2xa_switch)
extern struct xa_switch_t db2xa_switch;
#endif

main( )
{
    struct xa_switch_t *foo;
    printf ( "%s \n", db2xa_switch.name );
    foo = db2xacic();
    printf ( "%s \n", foo->name );
    return ;
}
```

Bestimmung von XA-Schnittstellenfehlern

Wenn ein Fehler während einer XA-Anforderung vom Transaktionsmanager festgestellt wird, ist das Anwendungsprogramm eventuell nicht in der Lage, den Fehlercode vom Transaktionsmanager zu ermitteln. Wenn Ihr Programm abnormal beendet wird oder einen unklaren Rückkehrcode vom TP-Monitor oder Transaktionsmanager empfängt, sollten Sie das Serviceprotokoll des DB2-Diagnoseprogramms überprüfen, in dem XA-Fehlerinformationen aufgezeichnet werden, wenn Diagnosestufe 3 oder höher in Kraft ist. Weitere Informationen zum Serviceprotokoll des DB2-Diagnoseprogramms finden Sie im Handbuch *Troubleshooting Guide*.

Sie sollten auch die Informationen der Konsolnachricht, der Fehlerdatei des Transaktionsmanagers oder andere produktspezifische Informationen über die verwendete externe Software zur Verarbeitung von Transaktionen berücksichtigen.

Der Datenbankmanager schreibt alle XA-spezifischen Fehler mit dem SLQ-CODE -998 (Fehler bei Transaktion oder heuristischer Maßnahme) und den entsprechenden Ursachencodes in das Serviceprotokoll des DB2-Diagnoseprogramms. Es folgen einige der häufigeren Fehlerursachen:

- Ungültige Syntax in der XA-Zeichenfolge zum Öffnen
- Fehler bei der Verbindung mit der in der XA-Zeichenfolge zum Öffnen angegebenen Datenbank aus einem der folgenden Gründe:
 - Die Datenbank ist nicht katalogisiert.

- Die Datenbank wurde nicht gestartet.
- Der Benutzername bzw. das Kennwort der Server-Anwendung ist zur Verbindung mit der Datenbank nicht berechtigt.
- Übertragungsfehler

Das folgende Beispiel zeigt ein Fehlerprotokoll für einen xa_open-Fehler (wegen einer fehlender XA-Zeichenfolge zum Öffnen), das unter AIX generiert wurde:

```
Tue Apr  4 15:59:08 1995
toop pid(83378) process (xatest) XA DTP Support      sqlxa_open Probe:101
DIA4701E Datenbank "" konnte nicht für verteilte
Transaktionsverarbeitung geöffnet werden.
String Title : XA Interface SQLCA pid(83378)
SQLCODE = -998 REASON CODE: 4 SUBCODE: 1
Dump File : /u/toop/diagnostics/83378.dmp Data : SQLCA
```

Konfigurieren des XA-Transaktionsmanagers zur Verwendung von DB2 UDB

Beachten Sie, daß die Informationen in diesem Abschnitt einen ähnlichen Abschnitt im Handbuch *Systemverwaltung: Optimierung* ablösen.

In den folgenden Abschnitten wird beschrieben, wie bestimmte Produkte zur Verwendung von DB2 als Ressourcenmanager konfiguriert werden. Möglich sind:

- „Konfigurieren von IBM TXSeries CICS“
- „Konfigurieren von IBM TXSeries Encina“
- „Konfigurieren von BEA Tuxedo“ auf Seite 222
- „Konfigurieren von Microsoft Transaction Server“ auf Seite 224

Konfigurieren von IBM TXSeries CICS

Informationen zur Konfiguration von IBM TXSeries CICS zur Verwendung von DB2 als Ressourcenmanager finden Sie im Handbuch *IBM TXSeries CICS Administration Guide*. Die TXSeries-Dokumentation steht online unter http://www.transarc.com/Library/documentation/websphere/WAS-EE/en_US/html/ zur Verfügung.

Host- und AS/400-Datenbank-Server können an CICS-koordinierten Transaktionen teilnehmen.

Konfigurieren von IBM TXSeries Encina

Im folgenden werden die verschiedenen APIs und Konfigurationsparameter aufgeführt, die für die Integration von Encina Monitor und DB2 Universal Database-Servern oder DB2 für MVS, DB2 für OS/390, DB2 für AS/400 oder DB2 für VSE&VM erforderlich sind, wenn über DB2 Connect darauf zugegrif-

fen wird. Die TXSeries-Dokumentation steht online unter http://www.transarc.com/Library/documentation/websphere/WAS-EE/en_US/html/ zur Verfügung.

Host- und AS/400-Datenbank-Server können an CICS-koordinierten Transaktionen teilnehmen.

Konfigurieren von DB2

Gehen Sie wie folgt vor, um DB2 zu konfigurieren:

1. Jeder Datenbankname muß im DB2-Datenbankverzeichnis definiert werden. Wenn es sich bei der Datenbank um eine ferne Datenbank handelt, muß auch ein Eintrag im Knotenverzeichnis definiert werden. Sie können die Konfiguration mit dem Programm Client-Konfiguration - Unterstützung oder mit dem DB2-Befehlszeilenprozessor ausführen. Beispiel:

```
DB2 CATALOG DATABASE inventdb AS inventdb AT NODE host1 AUTH SERVER
DB2 CATALOG TCPIP NODE host1 REMOTE hostname1 SERVER svcname1
```

2. Der DB2-Client kann seine interne Verarbeitung für Encina optimieren, wenn er weiß, daß er es mit Encina zu tun hat. Sie können dies angeben, indem Sie den Konfigurationsparameter *tp_mon_name* des Datenbankmanagers auf ENCINA setzen. In der Standardeinstellung erfolgt keine spezielle Optimierung. Wenn *tp_mon_name* definiert ist, muß die Anwendung sicherstellen, daß der Thread, der die Arbeitseinheit ausführt, die Arbeit nach Beendigung auch sofort festschreibt. Es darf keine andere Arbeitseinheit gestartet werden. Wenn dies *nicht* Ihre Umgebung ist, stellen Sie sicher, daß der Wert von *tp_mon_name* NONE ist (oder der über den Befehlszeilenprozessor auf NULL gesetzt wird). Der Parameter kann über die Steuerzentrale oder den Befehlszeilenprozessor aktualisiert werden. Der Befehl für den Befehlszeilenprozessor (CLP) lautet:

```
db2 update dbm cfg using tp_mon_name ENCINA
```

Konfigurieren von Encina für jeden Ressourcenmanager

Bei der Konfiguration von Encina für jeden Ressourcenmanager (RM) muß ein Administrator die Zeichenfolge zum Öffnen, die Zeichenfolge zum Schließen und den Thread der Steuerungsvereinbarung für jede DB2-Datenbank als Ressourcenmanager definieren, damit der Ressourcenmanager für Transaktionen in einer Anwendung registriert werden kann. Die Konfiguration kann mit der Enconcole-Gesamtanzeigeschnittstelle oder mit der Encina-Befehlszeilenschnittstelle ausgeführt werden. Beispiel:

```
monadmin create rm inventdb -open "db=inventdb,uid=benutzer1,pwd=kennwort1"
```

Es gibt eine Ressourcenmanagerkonfiguration für jede DB2-Datenbank, und jede Konfiguration eines Ressourcenmanagers(RM) muß einen rm-Namen („logischen RM-Namen“) haben. Dieser sollte zur Vereinfachung mit dem Datenbanknamen übereinstimmen.

Die XA-Zeichenfolge zum Öffnen (xa_open) enthält Informationen, die erforderlich sind, um eine Verbindung zur Datenbank herzustellen. Der Inhalt der Zeichenfolge ist RM-spezifisch. Die XA-Zeichenfolge zum Öffnen von DB2 UDB enthält den Aliasnamen der Datenbank, die geöffnet werden soll, und wahlweise eine Benutzer-ID und ein Kennwort, die der Verbindung zugeordnet werden sollen. Beachten Sie, daß der hier definierte Datenbankname auch in dem normalen Datenbankverzeichnis katalogisiert werden muß, das für alle Datenbankzugriffe erforderlich ist. Informationen über die XA-Zeichenfolge zum Öffnen von DB2 finden Sie in „Einrichten einer Datenbank als Ressourcenmanager“ auf Seite 203.

Die XA-Zeichenfolge zum Schließen (xa_close) wird von DB2 nicht verwendet.

Der Thread der Steuerungsvereinbarung (Thread of Control Agreement) bestimmt, ob ein Anwendungsagent-Thread mehrere Transaktionen gleichzeitig verarbeiten kann. DB2 UDB unterstützt den Standardwert von TMXA_SERIALIZE_ALL_OPERATIONS, wonach ein Thread nur dann wiederverwendet werden kann, wenn eine Transaktion vollständig beendet ist.

Wenn Sie auf DB2 für OS/390, DB2 für MVS, DB2 für AS/400 oder DB2 für VSE&VM zugreifen, müssen Sie den DB2-Synchronisationspunktmanager verwenden. Weitere Informationen zur Konfiguration finden Sie im Handbuch *DB2 Connect Enterprise Edition für OS/2 und Windows Einstieg*.

Verweisen auf eine DB2-Datenbank aus einer Encina-Anwendung

Gehen Sie wie folgt vor, um auf eine DB2-Datenbank aus einer Encina-Anwendung zu verweisen:

1. Verwenden Sie die Encina Scheduling Policy-API, um anzugeben, wie viele Anwendungsagenten aus einem einzelnen TP-Monitoranwendungsprozeß ausgeführt werden können. Beispiel:

```
rc = mon_SetSchedulingPolicy (MON_EXCLUSIVE)
```

Für DB2 (DB2 Universal Database, Host- oder AS/400-Datenbank-Server) müssen Sie die Standardeinstellung MON_EXCLUSIVE verwenden. Damit wird sichergestellt, daß:

- Der Anwendungsprozeß während der Dauer der Transaktion gesperrt wird.
- Die Anwendung mit einem einzigen Tread arbeitet.

Anmerkung: Wenn Sie ODBC oder DB2 Call Level Interface verwenden, müssen Sie die Multithreading-Unterstützung inaktivieren. Sie können dazu den Konfigurationsparameter der Befehlszeilenschnittstelle folgendermaßen einstellen:
DISABLEMULTITHREAD = 1 (inaktiviert das Multithreading).
Der Standardwert für DB2 Universal Database ist

DISABLEMULTITHREAD = 0 (aktiviert das Multithreading).
Weitere Informationen finden Sie im Handbuch *CLI Guide and Reference*.

2. Verwenden Sie die Encina RM Registration-API, um den XA-Schalter und den logischen RM-Namen bereitzustellen, die von Encina zur Angabe des RMs in einem Anwendungsprozeß verwendet werden sollen. Beispiel:

```
rc = mon_RegisterRmi ( &db2xa_switch, /* xa-Schalter */  
                    "inventdb",      /* logischer RM-Name */  
                    &rmiId );        /* interne RM-ID */
```

Der XA-Schalter enthält die Adressen der XA-Routinen im RM, die der TM aufrufen kann. Er gibt auch die Funktionalität an, die vom RM bereitgestellt wird. Der XA-Schalter von DB2 Universal Database ist `db2xa_switch`, und er befindet sich in der DB2-Client-Bibliothek (`db2app.dll` auf Windows-Betriebssystemen und `OS/2` sowie `libdb2` auf UNIX-basierten Systemen).

Von Encina wird der logische RM-Name und nicht der eigentliche Name der Datenbank verwendet, die von der SQL-Anwendung verwendet wird, die unter Encina ausgeführt wird. Der tatsächliche Name der Datenbank wird in der XA-Zeichenfolge zum Öffnen in der Encina RM Registration-API angegeben. Der logische RM-Name in diesem Beispiel stimmt mit dem Datenbanknamen überein.

Der dritte Parameter gibt eine interne Kennung zurück, die vom TM verwendet wird, um auf diese Verbindung zu verweisen.

Anmerkung: Wenn Encina für die Transaktionsverarbeitung mit DB2 über die TM-XA-Schnittstelle verwendet wird, ist zu beachten, daß verschachtelte Encina-Transaktionen von der DB2-XA-Schnittstelle zur Zeit nicht unterstützt werden. Eine Verwendung dieser Transaktionen sollte nach Möglichkeit vermieden werden. Wenn dies nicht möglich ist, stellen Sie sicher, daß die SQL-Arbeit in nur einem Mitglied der Encina-Transaktionsfamilie durchgeführt wird.

Konfigurieren von BEA Tuxedo

Führen Sie die folgenden Schritte aus, um Tuxedo zur Verwendung von DB2 als Ressourcenmanager zu konfigurieren:

1. Installieren Sie Tuxedo, wie in der Dokumentation für das Produkt angegeben. Stellen Sie sicher, daß Sie die gesamte Basiskonfiguration von Tuxedo durchführen, einschließlich Protokolldateien und Umgebungsvariablen.

Sie benötigen auch einen Compiler und das DB2 Application Development Client. Installieren Sie diese, falls erforderlich.

2. Definieren Sie auf der Tuxedo-Server-ID die Umgebungsvariable DB2INSTANCE, so daß sie auf das Exemplar verweist, das die Datenbanken enthält, die Tuxedo verwenden soll. Definieren Sie die Variable PATH, so daß sie die Programmverzeichnisse von DB2 enthält. Bestätigen Sie, daß die Tuxedo-Server-ID eine Verbindung zu den DB2-Datenbanken herstellen kann.
3. Aktualisieren Sie den Konfigurationsparameter *tp_mon_name* des Datenbankmanagers mit dem Wert TUXEDO.
4. Fügen Sie eine Definition für DB2 zur Tuxedo-Ressourcenmanagerdefinitionsdatei hinzu. In den folgenden Beispielen ist UDB_XA der lokal definierte Tuxedo-Ressourcenmanagername für DB2, und *db2xa_switch* ist der DB2-definierte Name für eine Struktur des Typs *xa_switch_t*:

- Für AIX. Fügen Sie in der Datei `${TUXDIR}/udataobj/RM` folgende Definition hinzu:

```
# DB2 UDB
UDB_XA:db2xa_switch:-L${DB2DIR} /lib -ldb2
```

Dabei ist `{TUXDIR}` das Verzeichnis, in dem Sie Tuxedo installiert haben, und `{DB2DIR}` das Verzeichnis des DB2-Exemplars.

- Für Windows NT. Fügen Sie in der Datei `%TUXDIR%\udataobj\rm` folgende Definition hinzu:

```
# DB2 UDB
UDB_XA;db2xa_switch;%DB2DIR%\lib\db2api.lib
```

Dabei ist `%TUXDIR%` das Verzeichnis, in dem Sie Tuxedo installiert haben, und `%DB2DIR%` das Verzeichnis des DB2-Exemplars.

5. Erstellen Sie das Tuxedo-Transaktionsmonitor-Server-Programm für DB2:
 - Für AIX:


```
${TUXDIR}/bin/buildtms -r UDB_XA -o ${TUXDIR}/bin/TMS_UBD
```

Dabei ist `{TUXDIR}` das Verzeichnis, in dem Sie Tuxedo installiert haben.
 - Für Windows NT:


```
%TUXDIR%\bin\buildtms -r UDB_XA -o %TUXDIR%\bin\TMS_UBD
```
6. Erstellen Sie die Anwendungs-Server. In den folgenden Beispielen gibt die Option `-r` den Ressourcenmanagernamen, die Option `-f` (einmal oder mehrmals verwendet) die Dateien, die die Anwendungsservices enthalten, die Option `-s` die Anwendungsservicenamen für diesen Server und die Option `-o` den Ausgabe-Server-Dateinamen an:

- Für AIX:

```
${TUXDIR}/bin/buildserver -r UDB_XA -f svcfile.o -s SVC1,SVC2
-o UDBserver
```

Dabei ist {TUXDIR} das Verzeichnis, in dem Sie Tuxedo installiert haben.

- Für Windows NT:

```
%TUXDIR%\bin\buildserver -r UDB_XA -f svcfile.o -s SVC1,SVC2  
-o UDBserver
```

Dabei ist %TUXDIR% das Verzeichnis, in dem Sie Tuxedo installiert haben.

7. Richten Sie die Tuxedo-Konfigurationsdatei so ein, daß auf den DB2-Server verwiesen wird. Fügen Sie im Abschnitt *GROUPS der Datei UDBCONFIG einen Eintrag wie den folgenden hinzu:

```
UDB_GRP LMID=simp GRPNO=3  
TMSNAME=TMS_UDB TMSCOUNT=2  
OPENINFO="UDB_XA:db=sample,uid=db2_benutzer,pwd=db2_benutzer_kwt"
```

Dabei gibt der Parameter TMSNAME das Transaktionsmonitor-Server-Programm an, das Sie zuvor erstellt haben, und der Parameter OPENINFO gibt den Ressourcenmanagernamen an. Darauf folgen der Datenbankname und DB2-Benutzer und Kennwort, die für die Authentifizierung verwendet werden.

Die Anwendungs-Server, die Sie zuvor erstellt haben, werden im Abschnitt *SERVERS der Tuxedo-Konfigurationsdatei angegeben.

8. Wenn die Anwendung auf Daten zugreift, die sich in DB2 für OS/390, DB2 für OS/400 oder DB2 für VM&VSE befinden, ist der DB2 Connect XA-Konzentrator erforderlich. Einzelangaben zur Konfiguration und zu Einschränkungen finden Sie im *DB2 Connect Benutzerhandbuch*.
9. Starten Sie Tuxedo:

```
tboot -y
```

Nach Beendigung des Befehls sollten Tuxedo-Nachrichten angeben, daß die Server gestartet wurden. Außerdem sollten Sie, wenn Sie den DB2-Befehl LIST APPLICATIONS ALL absetzen, zwei Verbindungen sehen (in diesem Fall die vom Parameter TMSCOUNT in der UDB-Gruppe in der Tuxedo-Konfigurationsdatei UDBCONFIG angegebenen).

Konfigurieren von Microsoft Transaction Server

DB2 UDB Version 5.2 und spätere Versionen können vollständig mit Microsoft Transaction Server (MTS) Version 2.0 integriert werden. Anwendungen, die unter MTS auf 32-Bit-Windows-Betriebssystemen ausgeführt werden, können MTS verwenden, um eine zweiphasige Festschreibung mit mehreren DB2 UDB-, Host- und AS/400-Datenbank-Servern sowie mit anderen MTS-konformen Ressourcenmanagern zu koordinieren.

Aktivieren der MTS-Unterstützung in DB2

Die MTS-Unterstützung wird automatisch aktiviert. Obgleich Sie den Konfigurationsparameter *tp_mon_name* des Datenbankmanagers auf MTS setzen können, ist dies nicht notwendig und wird ignoriert.

Anmerkung: Zusätzliche technische Informationen stehen eventuell auf der IBM Web-Site bereit, um Ihnen bei der Installation und Konfiguration der MTS-Unterstützung von DB2 zu helfen. Geben Sie die URL-Adresse

<http://www.ibm.com/software/data/db2/library/> an, und suchen Sie nach einem DB2 Universal Database-Artikel ("Technote") mit dem Suchbegriff MTS.

MTS-Softwareanforderungen

Die MTS-Unterstützung benötigt DB2 Client Application Enabler (CAE) Version 5.2 oder später, und MTS muß Version 2.0 mit Hotfix 0772 oder höheren Releases haben.

Bei der Installation des DB2-ODBC-Treibers auf 32-Bit-Windows-Betriebssysteme wird automatisch ein neues Schlüsselwort in die Registrierdatenbank eingefügt:

```
HKEY_LOCAL_MACHINE\software\ODBC\odbcinit.ini\IBM DB2 ODBC Driver:  
Wert des Schlüsselworts: CTimeout  
Datentyp: REG_SZ  
Wert: 60
```

Installation und Konfiguration

Der folgende Abschnitt bietet eine Zusammenfassung von Überlegungen zur Installation und Konfiguration von MTS. Zur Verwendung der MTS-Unterstützung von DB2 müssen folgende Voraussetzungen erfüllt werden:

1. Installieren Sie MTS und den DB2-Client auf der gleichen Maschine, auf der die MTS-Anwendung ausgeführt wird.
2. Gehen Sie wie folgt vor, wenn Host- oder IBM AS/400-Datenbank-Server an einer Aktualisierung auf mehreren Systemen beteiligt sein sollen:
 - a. Installieren Sie DB2 Connect Enterprise Edition (EE) entweder auf Ihrer lokalen Maschine oder auf einer fernen Maschine. DB2 Connect EE ermöglicht Host- oder IBM AS/400-Datenbank-Servern an einer Aktualisierung auf mehreren Systemen teilzunehmen.
 - b. Stellen Sie sicher, daß Ihr DB2 Connect EE-Server für die Aktualisierung auf mehreren Systemen aktiviert ist. Informationen zum Aktivieren von DB2 Connect für eine Aktualisierung auf mehreren Systemen finden Sie im Handbuch *DB2 Connect Enterprise Edition Einstieg* für Ihre Plattform.

Wenn Sie CLI/ODBC-Anwendungen ausführen, dürfen die Standardwerte der folgenden Konfigurationsschlüsselwörter (die in der Datei `db2cli.ini` festgelegt sind) nicht geändert werden:

- Schlüsselwort `CONNECTTYPE` (Standardwert 1)
- Schlüsselwort `MULTICONNECT` (Standardwert 1)
- Schlüsselwort `DISABLEMULTITHREAD` (Standardwert 1)
- Schlüsselwort `CONNECTIONPOOLING` (Standardwert 0)
- Schlüsselwort `KEEPCONNECTION` (Standardwert 0)

CLI-Anwendungen, die für die Verwendung der MTS-Unterstützung geschrieben wurden, dürfen nicht die Attributwerte ändern, die den obigen Schlüsselwörtern entsprechen. Außerdem dürfen die Anwendungen nicht die Standardwerte der folgenden Attribute ändern:

- Attribut `SQL_ATTR_CONNECT_TYPE` (Standardwert `SQL_CONCURRENT_TRANS`)
- Attribut `SQL_ATTR_CONNECTION_POOLING` (Standardwert `SQL_C_P_OFF`)

Anmerkung: Zusätzliche technische Informationen stehen eventuell auf der IBM Web-Site bereit, um Ihnen bei der Installation und Konfiguration der MTS-Unterstützung von DB2 zu helfen. Geben Sie die URL-Adresse <http://www.ibm.com/software/data/db2/library/> an, und suchen Sie nach einem DB2 Universal Database-Artikel ("Technote") mit dem Suchbegriff MTS.

Prüfen der Installation

1. Konfigurieren Sie den DB2-Client und DB2 Connect EE für den Zugriff auf den DB2 UDB-, Host- oder IBM AS/400-Server.
2. Prüfen Sie die Verbindung von der DB2 CAE-Maschine zu den DB2 UDB-Datenbank-Servern.
3. Prüfen Sie die Verbindung von der DB2 Connect-Maschine zu Ihrem Host- oder AS/400-Datenbank-Server mit dem DB2-Befehlszeilenprozessor, und setzen Sie einige Abfragen ab.
4. Prüfen Sie die Verbindung von der DB2 CAE-Maschine über den DB2 Connect-Gateway zu Ihrem Host- oder AS/400-Datenbank-Server, und setzen Sie einige Abfragen ab.

Unterstützte DB2-Datenbank-Server

Die folgenden Server werden für die Aktualisierung auf mehreren Systemen mit Hilfe von MTS-koordinierten Transaktionen unterstützt:

- DB2 Universal Database Enterprise Edition Version 5.2
- DB2 Enterprise - Extended Edition Version 5.2

- DB2 für OS/390
- DB2 für MVS
- DB2 für AS/400
- DB2 für VM&VSE
- DB2 Common Server für SCO, Version 2
- DB2 Universal Database für AIX mit PTF U453782
- DB2 Universal Database für HP-UX mit PTF U453784
- DB2 Universal Database Enterprise Edition für OS/2 mit PTF WR09033
- DB2 Universal Database für SOLARIS mit PTF U453783
- DB2 Universal Database Enterprise Edition für Windows NT mit PTF WR09034
- DB2 Universal Database Enterprise - Extended Edition für UNIX oder Windows NT

MTS-Transaktionszeitlimit und DB2-Verbindungsverhalten

Sie können den Wert für das Transaktionszeitlimit im MTS Explorer-Tool festlegen. Weitere Informationen finden Sie online im *MTS Administrator Guide*.

Wenn eine Transaktion länger dauert als das Transaktionszeitlimit (der Standardwert ist 60 Sekunden), setzt MTS asynchron eine Abbruchanforderung an alle betroffenen Ressourcenmanager ab, und die gesamte Transaktion wird abgebrochen.

Für die Verbindung zu einem DB2-Server wird die Abbruchanforderung in eine DB2-Anforderung ROLLBACK übersetzt. Wie alle anderen Datenbankanforderungen wird die ROLLBACK-Anforderung auf der Verbindung serialisiert, um die Integrität der Daten auf dem Datenbank-Server zu gewährleisten.

Ergebnis:

- Wenn sich die Verbindung im Leerlauf befindet, wird die ROLLBACK-Operation sofort ausgeführt.
- Wenn eine SQL-Anweisung mit langer Ausführungsdauer verarbeitet wird, wartet die ROLLBACK-Anforderung, bis die SQL-Anweisung beendet ist.

Verbindungspool

Der Verbindungspool erlaubt es einer Anwendung, eine Verbindung aus einem Pool von Verbindungen zu verwenden, so daß die Verbindung nicht für jede Verwendung erneut erstellt werden muß. Nachdem eine Verbindung erstellt und einem Pool hinzugefügt wurde, kann eine Anwendung diese Verbindung erneut verwenden, ohne einen vollständigen Verbindungsprozeß ausführen zu müssen. Die Verbindung wird dem Pool hinzugefügt, wenn die

Anwendung die Verbindung zur ODBC-Datenquelle trennt, und einer neuen Verbindung übergeben, deren Attribute identisch sind.

Der Verbindungspool ist eine Funktion des ODBC-Treibermanagers 2.x. Beim neuesten ODBC-Treibermanager (Version 3.5), der mit MTS geliefert wurde, gibt es für den Verbindungspool einige Konfigurationsänderungen und eine neue Funktionsweise für ODBC-Verbindungen von MTS COM-Objekten für Transaktionen (siehe „Wiederverwenden von ODBC-Verbindungen zwischen COM-Objekten, die an der gleichen Transaktion teilnehmen“ auf Seite 229).

Der ODBC-Treibermanager 3.5 verlangt, daß der ODBC-Treiber ein neues Schlüsselwort in der Registrierdatenbank registriert, damit der Verbindungspool aktiviert werden kann. Das Schlüsselwort ist:

```
Schlüsselwort: SOFTWARE\ODBC\ODBCINST.INI\IBM DB2 ODBC DRIVER
Name: CTimeout
Typ: REG_SZ
Wert: 60
```

Der DB2-ODBC-Treiber ab Version 6 für das 32-Bit-Betriebssystem Windows unterstützt vollständig den Verbindungspool. Daher wird dieses Schlüsselwort registriert. Clients unter Version 5.2 müssen das FixPak 3 (WR09024) oder höher installieren.

Der Standardwert 60 bedeutet, daß eine Verbindung für 60 Sekunden im Pool behalten wird, bevor sie getrennt wird.

In einer Umgebung mit hoher Auslastung ist es besser, den Wert CTimeout möglichst hoch zu setzen (Microsoft schlägt für bestimmte Umgebungen 10 Minuten vor), um zu häufiges Herstellen und Trennen physischer Verbindungen zu verhindern, weil diese große Mengen an Systemressourcen, einschließlich Systemspeicher- und Übertragungstapelressourcen, beanspruchen.

Darüber hinaus müssen Sie auf einer Mehrprozessormaschine die Unterstützung für mehrere Pools pro Prozessor inaktivieren, um sicherzustellen, daß dieselbe Verbindung zwischen Objekten in der gleichen Transaktion verwendet wird. Dazu kopieren Sie die folgende Registrierdatenbankeinstellung in eine Datei namens `odbcpool.reg`, sichern diese Datei als Textdatei und setzen den Befehl **odbcpool.reg** ab. Das Windows-Betriebssystem importiert daraufhin diese Registrierdatenbankeinstellungen.

```
REGEDIT4
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\ODBC Connection Pooling]
"NumberOfPools"="1"
```

Wenn dieses Schlüsselwort nicht auf den Wert 1 gesetzt ist, kann MTS Verbindungen in verschiedenen Pools anlegen, so daß nicht dieselbe Verbindung wiederverwendet wird.

MTS-Verbindungspoolfunktion mit ADO 2.1 und späteren Versionen

Wenn die MTS-COM-Objekte mit Hilfe von ADO auf die Datenbank zugreifen, muß die Poolfunktion für OLEDB-Ressourcen inaktiviert werden, so daß der Microsoft-OLEDB-Anbieter für ODBC (MSDASQL) keinen störenden Einfluß auf die ODBC-Verbindungspoolfunktion hat. Diese Einrichtung wird mit dem Wert OFF in ADO 2.0, in ADO 2.1 jedoch mit dem Wert ON initialisiert. Zum Ausschalten der OLEDB-Ressourcenpoolfunktion kopieren Sie die folgenden Zeilen in eine Datei namens oledb.reg, sichern die Datei als Textdatei und setzen den Befehl **oledb.reg** ab. Das Windows-Betriebssystem importiert daraufhin diese Registrierdatenbankeinstellung.

```
REGEDIT4
```

```
[HKEY_CLASSES_ROOT\CLSID\{c8b522cb-5cf3-11ce-ade5-00aa0044773d}]  
@="MSDASQL"  
"OLEDB_SERVICES"=dword:ffffffffc
```

Wiederverwenden von ODBC-Verbindungen zwischen COM-Objekten, die an der gleichen Transaktion teilnehmen

Für ODBC-Verbindungen in MTS COM-Objekten ist der Verbindungspool automatisch eingeschaltet (unabhängig davon, ob es sich um ein COM-Objekt für Transaktionen handelt).

Wenn mehrere MTS COM-Objekte an der gleichen Transaktion teilnehmen, kann die Verbindung zwischen zwei oder mehr COM-Objekten auf folgende Weise wiederverwendet werden.

Angenommen, es gibt zwei COM-Objekte, COM1 und COM2, die eine Verbindung zur gleichen ODBC-Datenquelle herstellen und an der gleichen Transaktion teilnehmen.

Nachdem COM1 die Verbindung hergestellt und seine Arbeit ausgeführt hat, trennt es die Verbindung, und die Verbindung wird im Pool behalten. Diese Verbindung wird jedoch für die Verwendung durch andere COM-Objekte der gleichen Transaktion reserviert. Sie ist für andere Transaktionen erst dann verfügbar, wenn die aktuelle Transaktion beendet ist.

Wenn COM2 in der gleichen Transaktion aufgerufen wird, erhält es die gleiche Verbindung aus dem Pool. MTS stellt sicher, daß die Verbindung nur die COM-Objekte erhalten, die an der gleichen Transaktion teilnehmen.

Wenn andererseits COM1 die Verbindung nicht explizit trennt, belegt es die Verbindung, bis die Transaktion beendet ist. Wenn COM2 in der gleichen Transaktion aufgerufen wird, wird eine separate Verbindung hergestellt. Als Folge belegt diese Transaktion zwei Verbindungen statt einer.

Die Funktion zur Wiederverwendung von Verbindungen für COM-Objekte, die an der gleichen Transaktion teilnehmen, ist aus folgenden Gründen vorteilhaft:

- Sie verwendet weniger Ressourcen sowohl beim Client als auch beim Server. Es wird nur eine Verbindung benötigt.
- Durch sie wird die Möglichkeit ausgeschaltet, daß zwei Verbindungen, die an der gleichen Transaktion teilnehmen (auf den gleichen Datenbank-Server und auf die gleichen Daten zugreifen), sich gegenseitig sperren, weil DB2-Server verschiedene Verbindungen von MTS COM-Objekten als separate Transaktionen behandeln.

Optimieren von TCP/IP-Übertragungen

Wenn ein kleiner CTimeout-Wert in einer Umgebung mit hoher Auslastung verwendet wird, in der zu viele physische Verbindungen zur gleichen Zeit hergestellt und getrennt werden, kann es beim TCP/IP-Stapel zu Ressourcenknappheit kommen.

Zur Milderung dieses Problems sollten Sie die TCP/IP-Registrierungseinträge verwenden. Diese werden im Handbuch *Windows NT Resource Guide*, Volume 1 beschrieben. Die Registrierungsschlüsselwerte befinden sich in HKEY_LOCAL_MACHINE-> SYSTEM-> CurrentControlSet-> Services-> TCPIP-> Parameters.

Die Standardwerte und die vorgeschlagenen Einstellungen sind:

Name	Standardwert	Vorgeschlagener Wert
KeepAlive time	7200000 (2 Stunden)	Derselbe
KeepAlive interval	1000 (1 Sekunde)	10000 (10 Sekunden)
TcpKeepCnt	120 (2 Minuten)	240 (4 Minuten)
TcpKeepTries	20 (20 Wiederholungen)	Derselbe
TcpMaxConnectAttempts	3	6
TcpMaxConnectRetransmission	3	6
TcpMaxDataRetransmission	5	8
TcpMaxRetransmissionAttempts	7	10
Wenn der Registrierungswert nicht definiert ist, erstellen Sie ihn.		

Testen von DB2 mit der MTS-Beispielanwendung "BANK"

Sie können das mit MTS gelieferte Beispielprogramm "BANK" verwenden, um die Konfiguration der Client-Produkte und von MTS zu testen.

Führen Sie die folgenden Schritte aus:

1. Ändern Sie die Datei \Program Files\Common Files\ODBC\Data Sources\MTSSamples.dsn, so daß sie folgendermaßen aussieht:

```
[ODBC]
DRIVER=IBM DB2 ODBC DRIVER
UID=ihr_benutzer_id
PWD=ihr_kennwort
DSN=ihr_datenbank_aliasname
Description=MTS Samples
```

Dabei gilt folgendes:

- *ihr_benutzer_id* und *ihr_kennwort* sind die Benutzer-ID und das Kennwort für die Verbindung zum Host.
 - *ihr_datenbank_aliasname* ist der Aliasname für die Datenbank, der für die Verbindung zum Datenbank-Server verwendet wird.
2. Rufen Sie ODBC-Administrator in der **Systemsteuerung** auf, wählen Sie die Registerkarte **System-DSN** aus und fügen Sie die Datenquelle hinzu:
 - a. Wählen Sie IBM ODBC-Driver und anschließend **Fertigstellen** aus.
 - b. Wenn die Liste der Aliasnamen für die Datenbank angezeigt wird, wählen Sie den Namen aus, der zuvor angegeben wurde.
 - c. Wählen Sie **OK** aus.
 3. Stellen Sie mit Hilfe des DB2-Befehlszeilenprozessors unter der obigen ID *ihr_benutzer_id* eine Verbindung zu einer DB2-Datenbank her.
 - a. Binden Sie die Datei `db2cli.lst`:

```
db2 bind @C:\sqllib\bnd\db2cli.lst blocking all grant public
```
 - b. Binden Sie die Dienstprogramme.
Wenn der Server ein DRDA-Host-Server ist, binden Sie die Datei `ddcs-mvs.lst`, `ddcs400.lst` oder `ddcsvm.lst`, je nach Host, zu dem Sie eine Verbindung herstellen (OS/390, AS/400 oder VSE&VM). Beispiel:

```
db2 bind @C:\sqllib\bnd\ddcsmvs.lst blocking all grant public
```


Binden Sie andernfalls die Datei `db2ubind.lst`:

```
db2 bind @C:\sqllib\bnd\db2ubind.lst blocking all grant public
```
 - c. Erstellen Sie wie folgt die Beispieltabelle und -daten für die MTS-Beispielanwendung:

```
db2 create table account (accountno int, balance int)
db2 insert into account values(1, 1)
```
 4. Stellen Sie auf dem DB2-Client sicher, daß der Konfigurationsparameter *tp_mon_name* des Datenbankmanagers auf MTS gesetzt ist.
 5. Führen Sie die Anwendung "BANK" aus. Wählen Sie den Druckknopf **Account** und die Option **Visual C++** aus, und übergeben Sie dann die Anforderung. Andere Optionen verwenden möglicherweise SQL Server-spezifisches SQL und funktionieren vielleicht nicht.

Kapitel 11. Entwerfen für hohe Verfügbarkeit

DB2 Universal Database bietet *Unterstützung für Funktionsübernahme für hohe Verfügbarkeit* auf vielen Plattformen. Durch die Möglichkeit einer Funktionsübernahme kann im Falle eines Hardwarefehlers eine Auslastung von einem Prozessor automatisch auf einen anderen verlegt werden. Zum Beispiel unterstützt DB2 UDB unter AIX die Funktionsübernahme mit Hilfe von IBM HACMP (High Availability Cluster Multi-Processing). In diesem Abschnitt werden die Konzepte zur hohen Verfügbarkeit anhand von Beispielen aus AIX vorgestellt.

HACMP sorgt für eine erhöhte Systemverfügbarkeit mit Hilfe von Prozessorgruppen (Cluster), die Ressourcen wie Platten oder Netzwerkzugriff gemeinsam benutzen. Wenn ein Prozessor ausfällt, werden seine Funktionen von einem anderen Prozessor im Cluster übernommen.

Es gibt drei Modi der Funktionsübernahme:

Bereitschaftsmodus (Hot Standby)

In diesem Modus wird ein Prozessor für den Betrieb des DB2-Exemplars verwendet, während sich der zweite Prozessor im Bereitschaftsmodus befindet, um bei einem Ausfall des Betriebssystems oder der Hardware des ersten Prozessors das Exemplar zu übernehmen.

Gegenseitige Übernahme (Mutual Takeover)

In diesem Modus gilt:

- Beide Prozessoren werden zur Ausführung getrennter DB2-Exemplare verwendet.
- Ein Prozessor dient zur Ausführung eines DB2-Exemplars, während der andere Prozessor zur Ausführung von DB2-Anwendungen eingesetzt wird.

Bei einem Ausfall des Betriebssystems oder der Hardware eines der Prozessoren, übernimmt der andere Prozessor die Funktionen des ausgefallenen Prozessors und erledigt schließlich die Aufgaben beider Prozessoren.

Paralleler Zugriff (Concurrent Access)

In diesem Modus werden mehrere Prozessoren mit Hilfe des Produkts DB2 Universal Database Enterprise - Extended Edition (EEE) auf ein einziges DB2-Exemplar skaliert. Dies geschieht mit Hilfe eines Modells ohne gemeinsame Ressourcennutzung, indem die Daten so partitioniert werden, daß eine oder mehrere Partitionen auf jedem Prozessor im Cluster ausgeführt werden. Bei einem Ausfall des Betriebssystems

oder der Hardware eines der Prozessoren, übernimmt der andere Prozessor die Partitionen des ausgefallenen Prozessors. Bei DB2 UDB EEE ist kein paralleler Ressourcenmanager (Concurrent Resource Manager) zur Bereitstellung der Redundanz erforderlich. Die Redundanz wird mit Hilfe des Bereitschaftsmodus bzw. des Modus der gegenseitigen Übernahme verwaltet. Die Funktionen dieses Modus sind nur für Datenbankmanager erforderlich, die eine Architektur mit gemeinsam benutzten Systemen verwenden.

Jede dieser Konfigurationen kann verwendet werden, um eine oder mehrere Partitionen einer partitionierten Datenbank bei einem Ausfall zu übernehmen. Außerdem können sie jeweils ein vollständiges Exemplar einer Installation mit Einzelpartition übernehmen.

Bereitschaftsmodus (Hot Standby)

Der *Bereitschaftsmodus* (Hot hot) kann verwendet werden, um die automatische Übernahme des gesamten Exemplars einer Datenbank mit Einzelpartition oder einer Partition einer partitionierten Datenbankkonfiguration einzurichten. Wenn ein Prozessor ausfällt, kann ein anderer Prozessor im Cluster die Funktionen des ausgefallenen Prozessors durch eine automatische Übertragung des Exemplars übernehmen. Das Datenbankexemplar und die Datenbank selbst müssen sowohl für den primären Prozessor als auch für den Ersatzprozessor zugänglich sein.

- Der DB2-Installationspfad kann entweder ein Pfad sein, der von beiden Systemen gemeinsam benutzt wird, oder ein Pfad in einem nicht gemeinsam benutzten Dateisystem. Wenn er in einem nicht gemeinsam benutzten Dateisystem ist, müssen identische Programmversionen verwendet werden.
- Der DB2-Exemplarpfad kann sich entweder in einem gemeinsam benutzten Dateisystem oder in einem manuell gespiegelten Dateisystem befinden.
- Die Datenbank und zugehörige Behälter müssen sich in Dateisystemen (oder auf Einheiten) befinden, auf die beide Systeme Zugriff haben.
- Bei der Übernahme einer Partition in einer partitionierten Datenbankkonfiguration wird die Partition auf dem zweiten Prozessor neu gestartet: die Übernahmeprozedur ändert die Datei `db2nodes.cfg`, um auf diese Partition auf dem neuen Prozessor zu zeigen, und startet die Partition auf dem neuen Prozessor.
- Wenn eine Funktionsübernahme erfolgt, werden die externen Kommunikationsadressen für die unterstützten Kommunikationsprotokolle als Teil der Funktionsübernahmeprozedur transparent übertragen.

Detaillierte Informationen zu den tatsächlichen Installationsvoraussetzungen und zur Exemplarerstellung finden Sie im Handbuch *HACMP for AIX, Version 4.2: Installation Guide*, SC23-1940.

Beispiele

Für alle folgenden Beispiele gibt es jeweils eine Beispielprozedurdatei, die im Verzeichnis `sql1lib/samples/hacmp` von DB2 für AIX-Installationen gespeichert ist.

Exemplarübernahme

Das folgende Beispiel eines Szenarios mit Funktionsübernahme im Bereitschaftsmodus beschreibt einen HACMP-Cluster mit zwei Prozessoren, auf dem ein Datenbankexemplar mit einer Einzelpartition ausgeführt wird (Abb. 44). Informationen zum Konfigurieren eines HACMP-Clusters finden Sie in „Zusätzliche HACMP-Informationen“ auf Seite 241.

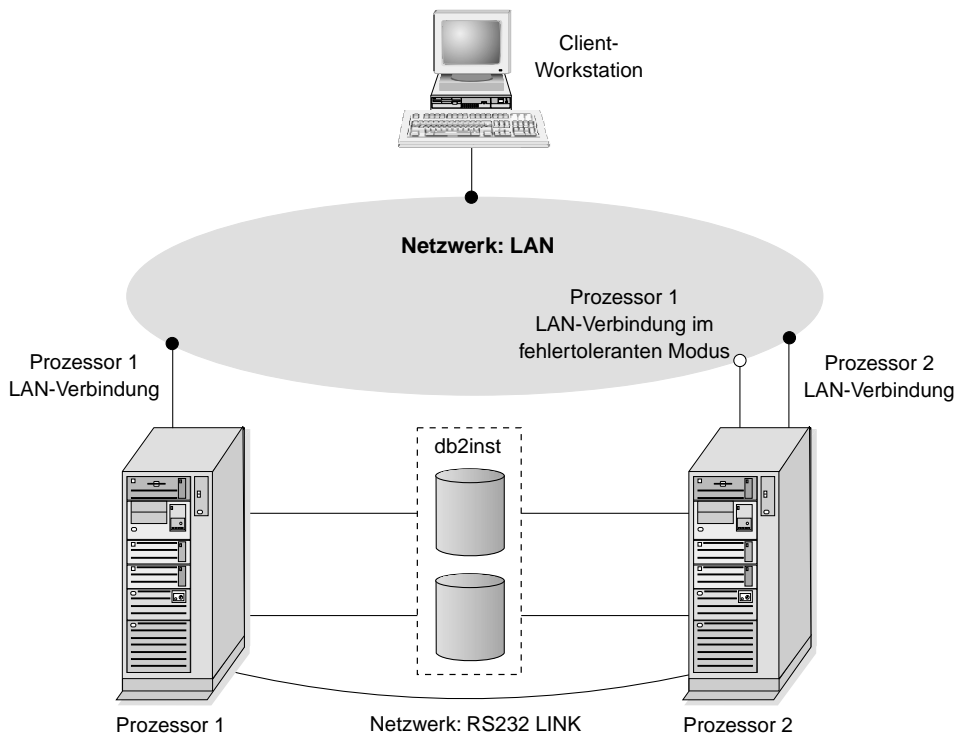


Abbildung 44. Beispiel für eine Funktionsübernahmekonfiguration im Bereitschaftsmodus

Beide Prozessoren haben Zugriff auf das Installationsverzeichnis, das Exemplarverzeichnis und das Datenbankverzeichnis. Das Datenbankexemplar "db2inst" wird aktiv auf Prozessor 1 ausgeführt. Prozessor 2 ist nicht aktiv, sondern wird als sofort einsatzfähiger Bereitschaftsprozessor verwendet. Auf Prozessor 1 kommt es zu einem Ausfall, und das Exemplar wird von Prozessor 2 übernommen. Sobald die Übernahme vollzogen ist, können sowohl ferne als auch lokale Anwendungen auf die Datenbank innerhalb des Exemplars "db2inst" zugreifen. Die Datenbank muß manuell erneut gestartet werden,

oder sie wird, falls der Parameter AUTORESTART aktiviert ist, durch die erste Verbindung zur Datenbank erneut gestartet. In der vorbereiteten Beispielprozedur wird angenommen, daß der Parameter AUTORESTART nicht aktiv ist (OFF) und daß die Übernahmeprozedur den Neustart für die Datenbank ausführt. Weitere Informationen zum Parameter AUTORESTART finden Sie in "Überblick über die Wiederherstellung" im Handbuch *Systemverwaltung: Implementierung*.

Beispielprozedur:

```
hacmp-s1.sh
```

Partitionsübernahme

Im folgenden Funktionsübernahmeszenario mit Bereitschaftsmodus (Hot Standby) wird eine Exemplarpartition anstelle des gesamten Exemplars verwendet. Dieses Szenario enthält wie im vorigen Beispiel einen HACMP-Cluster mit zwei Prozessoren, jedoch stellt die Maschine hier eine der Partitionen eines partitionierten Datenbank-Servers dar. Prozessor 1 betreibt eine einzelne Partition der Gesamtkonfiguration, während Prozessor 2 als übernehmender Prozessor fungiert. Wenn Prozessor 1 ausfällt, wird die Partition auf dem zweiten Prozessor neu gestartet. Bei der Übernahme wird die Datei `db2nodes.cfg` aktualisiert, so daß sie nun auf den Host-Namen und den Netznamen von Prozessor 2 zeigt, und anschließend die Partition auf dem neuen Prozessor gestartet.

Das folgende Beispiel zeigt einen Abschnitt aus der Datei `db2nodes.cfg` vor und nach der Funktionsübernahme. In diesem Beispiel wird Knotennummer 2 auf Prozessor 1 der HACMP-Maschine ausgeführt, die als Host-Namen und als Netznamen den Namen "node201" besitzt. Nach der Übernahme wird Knotennummer 2 auf Prozessor 2 der HACMP-Maschine mit dem Host-Namen und dem Netznamen "node202" ausgeführt.

Vorher:

```
1 node101 0 node101
2 node201 0 node201   <= HACMP
3 node301 0 node301
```

```
db2start nodenum 2 restart hostname node202 port 0 netname node202
```

Nachher:

```
1 node101 0 node101
2 node202 0 node202   <= HACMP
3 node301 0 node301
```

Beispielprozedur:

```
hacmp-s2.sh
```

Automatische Übernahme mehrerer logischer Knoten

Diese komplexere Variante als das vorige Beispiel beschreibt die automatische Übernahme mehrerer logischer Knoten von einem Prozessor durch einen anderen. Es wird wieder derselbe HACMP-Cluster mit zwei Prozessoren wie oben verwendet. In diesem Beispielszenario betreibt Prozessor 1 allerdings drei logische Partitionen. Die Ausgangssituation gleicht dem Szenario mit der einfachen Partitionsübernahme, jedoch muß in diesem Fall bei einem Ausfall von Prozessor 1 jede der logischen Partitionen auf Prozessor 2 gestartet werden. Jede logische Partition muß in der Reihenfolge gestartet werden, die in der Datei `db2nodes.cfg` definiert ist: die logische Partition mit der Anschlußnummer (Port) 0 muß immer als erste gestartet werden.

Das folgende Beispiel zeigt einen Abschnitt aus der Datei `db2nodes.cfg` vor und nach der Funktionsübernahme. In diesem Beispiel gibt es drei logische Partitionen auf Prozessor 1 in einem HACMP-Cluster mit zwei Prozessoren.

Vorher:

```
1 node101 0 node101
2 node201 0 node201   <= HACMP
3 node201 1 node201   <= HACMP
4 node201 2 node201   <= HACMP
5 node301 0 node301
```

```
db2start nodenum 2 restart hostname node202 port 0 netname node202
db2start nodenum 3 restart hostname node202 port 1 netname node202
db2start nodenum 4 restart hostname node202 port 2 netname node202
```

Nachher:

```
1 node101 0 node101
2 node202 0 node202   <= HACMP
3 node202 1 node202   <= HACMP
4 node202 2 node202   <= HACMP
5 node301 0 node301
```

Beispielprozedur:

```
hacmp-s3.sh
```

Gegenseitige Übernahme

Im Modus der *gegenseitigen Übernahme (Mutual Takeover)* kann ein Prozessor ein Exemplar mit einer Einzelpartitionsdatenbank oder die Partitionen einer partitionierten Datenbank übernehmen, während er ein anderes Exemplar oder andere Partitionen einer partitionierten Datenbankkonfiguration ausführt. Wie im Bereitschaftsmodus müssen die an der automatischen Übernahme beteiligten Prozessoren auf den Installationspfad, das Exemplarverzeichnis und die Datenbank zugreifen können. Die Installations- und Exemplarpfade können entweder auf einem gemeinsam benutzten Dateisystem oder gespiegelt auf getrennten Dateisystemen sein.

Wenn die Strategie der gegenseitige Übernahme von Exemplaren verwendet wird, müssen die Exemplare so definiert sein, daß beide Exemplare auf demselben Prozessor gleichzeitig ausgeführt werden können. Detaillierte Informationen zu den tatsächlichen Installationsvoraussetzungen und zur Exemplarerstellung finden Sie im Handbuch *HACMP for AIX, Version 4.2: Installation Guide, SC23-1940*.

Beispiele

Für alle folgenden Beispiele gibt es jeweils eine Beispielprozedurdatei, die im Verzeichnis `sql1lib/samples/hacmp` von DB2 für AIX-Installationen gespeichert ist.

Gegenseitige automatische Übernahme von DB2-Exemplaren

Das folgende Szenario einer gegenseitigen Exemplarübernahme enthält ein HACMP-System mit zwei Prozessoren mit den Namen "node10" und "node20".

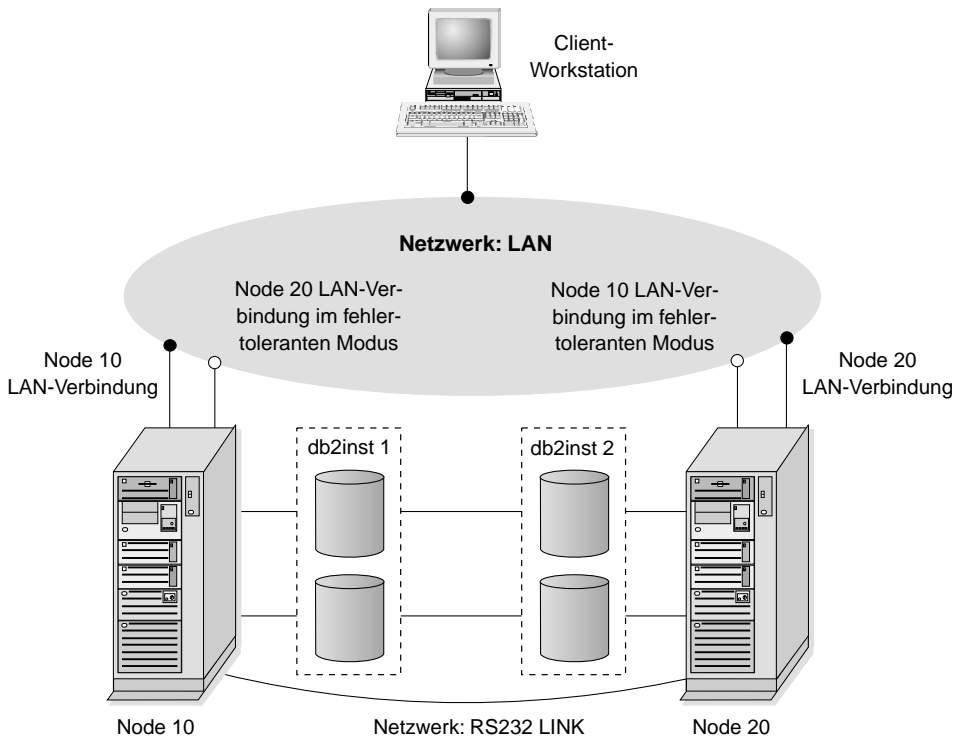


Abbildung 45. Konfigurationsbeispiel zur gegenseitigen Exemplarübernahme

Zwei Exemplare, "db2inst1" und "db2inst2", werden aus einem einzigen Installationspfad in einem gemeinsam benutzten Dateisystem erstellt. Das Exemplar "db2inst1" wird in `/u/db2inst1` und das Exemplar "db2inst2" in `/u/db2inst2` erstellt. Diese beiden Pfade liegen auf einem gemeinsam benutz-

ten Dateisystem, auf das beide Prozessoren Zugriff haben. Jedes Exemplar enthält eine einzelne Datenbank mit einem eindeutigen Pfad, der ebenfalls auf einer beiden Prozessoren zugänglichen gemeinsamen Ressource angelegt ist.

Auf beide Exemplare wird über ferne Clients mit Hilfe des TCP/IP-Protokolls zugegriffen: "db2inst1" verwendet den Servicenamen "db2inst1_port" (Portnummer 5500) und "db2inst2" den Servicenamen "db2inst2_port" (Portnummer 5550). In den Knotenverzeichnissen der fernen Clients, die auf das Exemplar "db2inst1" zugreifen, ist dieses Exemplar mit "node10" als Host-Namen katalogisiert. In den Knotenverzeichnissen der fernen Clients, die auf das Exemplar "db2inst2" zugreifen, ist dieses Exemplar mit "node20" als Host-Namen katalogisiert. Unter normalen Betriebsbedingungen werden "db2inst1" auf "node10" und "db2inst2" auf "node20" ausgeführt. Wenn „node10“ ausfallen sollte, startet die Übernahmeprozedur das Exemplar "db2inst1" auf "node20" und die externe IP-Adresse, die "node10" zugeordnet ist, wird auf "node20" umgeschaltet. Wenn das Exemplar von der Übernahmeprozedur gestartet und die Datenbank erneut gestartet wurde, können ferne Clients eine Verbindung zu der Datenbank in diesem Exemplar so herstellen, als würde es auf "node10" ausgeführt.

Beispielprozedur:

```
hacmp-s4.sh
```

Gegenseitige automatische Übernahme von DB2-Partitionen

Für die gegenseitige Übernahme von Partitionen in einer partitionierten Datenbank-Server-Umgebung ist es erforderlich, daß die übernommene Partition als logischer Knoten auf dem übernehmenden Prozessor auftritt. In einem Fall mit zwei Partitionen auf einem partitionierten Datenbank-Server, die auf getrennten Prozessoren eines zur gegenseitigen Übernahme konfigurierten HACMP-Clusters mit zwei Prozessoren ausgeführt werden, müssen die Partitionen zum Beispiel als logische Knoten übernommen werden. Die Standardpartition auf jedem Knoten muß als logischer Knoten 0 definiert sein, d. h., wenn eine Partition eines ausgefallenen Prozessors von einem anderen übernommen wird, wird sie als logischer Knoten gestartet, der keine direkten Empfangsfunktionen für ferne Übertragungsprotokolle hat. Eine solche Partition kann nicht als Koordinatorknoten verwendet werden.

Ein weiterer wichtiger Gesichtspunkt bei der Konfiguration eines Systems für gegenseitige Partitionsübernahme betrifft den Datenbankpfad der lokalen Partition. Wenn eine Datenbank in einer Umgebung mit partitionierten Datenbanken erstellt wird, wird sie in einem Stammfad erstellt, der unter den Servern der partitionierten Datenbank nicht gemeinsam benutzt wird. Betrachten Sie zum Beispiel den folgenden Befehl:

```
CREATE DATABASE db_a1 ON /dbpath
```

Dieser Befehl wird unter dem Exemplar "db2inst" ausgeführt und erstellt die Datenbank db_a1 im Pfad /dbpath. Jede Datenbankpartition wird in ihrem Dateisystem unter /dbpath/db2inst/nodexxxx erstellt, wobei xxxx die Knotennummer darstellt. Bei der HACMP-Übernahme wird versucht, das /dbpath-Dateisystem anzuhängen (Mount), was aber bereits vom anderen Prozessor benutzt wird. Daher muß die Übernahme-prozedur das Dateisystem unter einem anderen logischen Punkt anhängen und eine symbolische Verbindung von diesem Dateisystem zu dem richtigen Pfad /dpath/db2inst/nodexxxx definieren.

Das folgende Beispiel zeigt einen Abschnitt aus der Datei db2nodes.cfg vor und nach der Funktionsübernahme. In diesem Beispiel wird Knotennummer 2 auf Prozessor 1 der HACMP-Maschine ausgeführt, die als Host-Namen und als Netznamen den Namen "node201" besitzt. Knotennummer 3 wird auf Prozessor 2 auf der HACMP-Maschine ausgeführt, die als Host-Namen und als Netznamen den Namen "node202" besitzt.

Vorher:

```
1 node101 0 node101
2 node201 0 node201   <= HACMP
3 node202 0 node202   <= HACMP
4 node301 0 node301
```

```
db2start nodenum 2 restart hostname node202 port 1 netname node202
```

Nachher:

```
1 node101 0 node101
2 node202 1 node202   <= HACMP
3 node202 0 node202   <= HACMP
4 node301 0 node301
```

Nach der Übernahme muß jeder ferne Client, der versucht, direkt auf Knotennummer 2 als Koordinatorknoten zuzugreifen, den Knoteneintrag für die Datenbank erneut katalogisieren, um auf den Übernahmeknoten zu verweisen. Es wird nicht empfohlen, ein Szenario der gegenseitigen Funktionsübernahme für Koordinatorknoten zu verwenden. Wenn für Ihren Koordinatorknoten Redundanz erforderlich ist, verwenden Sie die Konfiguration für den Bereitschaftsmodus.

Beispielprozedur:

```
hacmp-s5.sh
```

Wiederherstellen der Verbindung nach einer Funktionsübernahme

Wenn ein Client mit Hilfe der Anweisung SET CLIENT eine Verbindung zu einem bestimmten Knoten herstellt und dieser Knoten bei einer Funktionsübernahme auf einen anderen Host verlegt wird, schlägt die nächste CONNECT-Anforderung des Clients fehl. Setzen Sie einen Befehl **db2stop**, gefolgt von einem Befehl **db2start knotennummer**, auf dem Knoten ab, auf dem

die Anweisung SET CLIENT ausgeführt wurde, und führen Sie die Anweisung erneut aus, damit sowohl der Client als auch der Server die neue physische Position den Zielknotens erkennen.

Zusätzliche HACMP-Informationen

Eingehende Informationen zu HACMP-Konzepten, zur Installation und zur Konfiguration finden Sie in folgenden Handbüchern:

- *HACMP for AIX, Version 4.2: Concepts and Facilities, SC23-1938*
- *HACMP for AIX, Version 4.2: Installation Guide, SC23-1940*
- *HACMP for AIX, Version 4.2: Planning Guide, SC23-1939.*

Teil 4. Hohe Verfügbarkeit

Kapitel 12. High Availability Cluster Multi-Processing, Enhanced Scalability (HACMP ES) für AIX

Enhanced scalability (ES - Erweiterte Skalierbarkeit) ist eine Einrichtung von High Availability Cluster Multi-processing (HACMP) für AIX Version 4.2.2, die zur Zeit nur auf RS/6000 SP-Knoten einsetzbar ist.

Diese Einrichtung stellt dieselbe Fehlerbehebung durch Funktionsübernahme wie HACMP bereit und besitzt dieselbe Ereignisstruktur wie frühere HACMP-Versionen (siehe *HACMP for AIX, V4.2.2, Enhanced Scalability Installation and Administration Guide*). Enhanced Scalability bietet außerdem folgende Merkmale:

- Größere HACMP-Cluster mit Skalierbarkeit auf bis zu 16 Knoten pro Cluster
- Zusätzliche Fehlerbehandlung durch *benutzerdefinierte Ereignisse (User-defined Events)*. Überwachte Bereiche können benutzerdefinierte Ereignisse auslösen, die so unterschiedlich sein können wie das Abbrechen eines Prozesses oder die Tatsache, daß Seitenwechselbereich bald an die Kapazitätsgrenze stößt. Solche Ereignisse umfassen vorher und nachher ausgeführte Ereignisse, die dem Funktionsübernahmeprozess bei Bedarf hinzugefügt werden können. Sonderfunktionen, die für verschiedene Implementierungen spezifisch sind, können in die Abläufe der vorher und nachher ausgeführten HACMP-Ereignisse integriert werden.

Eine *rules-Datei* (`/usr/sbin/cluster/events/rules.hacmprd`) enthält die Definitionen der HACMP-Ereignisse. Benutzerdefinierte Ereignisse werden dieser Datei hinzugefügt. Die Prozedurdateien (Skripte), die auszuführen sind, wenn Ereignisse eintreten, sind Teil dieser Definition.

Weitere Informationen zu benutzerdefinierten Ereignissen und der *rules-Dateien* finden Sie in „HACMP ES-Ereignisüberwachung und benutzerdefinierte Ereignisse“ auf Seite 267.

- HACMP-Client-Dienstprogramme zur Überwachung und Erkennung von Statusänderungen (in einem oder mehreren Clustern) von physischen, außerhalb des HACMP-Clusters befindlichen AIX-Knoten aus.

Die Knoten in HACMP ES-Clustern tauschen Nachrichten aus, die als *Heartbeats* oder *Keepalive-Pakete* bezeichnet werden und durch die jeder Knoten die anderen Knoten über seine Verfügbarkeit informiert. Ein Knoten, der nicht mehr reagiert, veranlaßt die übrigen Knoten im Cluster, die Wiederherstellung zu starten. Der Prozeß zur Wiederherstellung der Funktionalität wird als *node_down-Ereignis* oder auch als *Funktionsübernahme* (Failover) bezeichnet. Nach einem Wiederherstellungsprozeß folgt die erneute Integration des Knotens in den Cluster. Dieses Ereignis wird als *node_up-Ereignis* bezeichnet.

Es gibt zwei Arten von Ereignissen: Standardereignisse, die innerhalb der Funktionen von HACMP ES abgefangen werden, und benutzerdefinierte Ereignisse, die mit Überwachung von Parametern in Hardware- und Softwarekomponenten verbunden sind.

Eines der Standardereignisse ist das `node_down`-Ereignis. Bei der Planung, welche Maßnahmen als Teil des Wiederherstellungsprozesses durchzuführen sind, ermöglicht HACMP zwei Funktionsübernahmeoptionen: den reinen Bereitschaftsmodus (Hot bzw. Idle Standby) und den Modus der gegenseitigen Übernahme (Mutual Takeover).

Clusterkonfiguration

In einer Konfiguration im *Bereitschaftsmodus* (Hot Standby) hat der AIX-Prozessorknoten, der als Übernahmeknoten fungiert, *keine* andere Arbeitsbelastung. Bei einer Konfiguration zur *gegenseitigen Funktionsübernahme*, hat der AIX-Prozessorknoten, der als Übernahmeknoten fungiert, andere Arbeitsbelastungen.

Im allgemeinen arbeitet DB2 Universal Database Enterprise - Extended Edition (UDB EEE) im Modus der gegenseitigen Übernahme mit Partitionen auf jedem Knoten. Eine Ausnahme bildet eine Konstellation, in der der Katalogknoten Teil einer Bereitschaftskonfiguration ist.

Bei der Planung einer umfangreichen DB2-Installation auf einem RISC System/6000 SP mit HACMP ES müssen Sie besonderes Augenmerk auf die Verteilung der Knoten des Clusters innerhalb oder zwischen den RISC System/6000 SP-Frames legen. Wenn ein Knoten und der zugehörige Ausweichknoten in verschiedenen SP-Frames angelegt sind, ist eine Funktionsübernahme möglich, wenn ein ganzer Frame ausfällt (d. h. die Stromversorgung/Switch Boards des Frame ausfallen). Dies ist jedoch eine sehr seltene Ausnahme, weil es $N+1$ Stromversorgungen in jedem SP-Frame gibt und jeder SP-Switch redundante Pfade mit $N+1$ Ventilatoren und Stromzuführungen besitzt. Im Fall eines Frame-Ausfalls ist eventuell ein manueller Eingriff erforderlich, um die übrigen Frames wieder in Funktion zu setzen. Diese Wiederherstellungsprozedur ist im Handbuch *SP Administration Guide* dokumentiert. HACMP ES ermöglicht die Wiederherstellung nach SP-Knotenausfällen. Die Wiederherstellung nach Frame-Ausfällen ist von einer geeigneten Anordnung von Clustern innerhalb eines oder mehrerer SP-Frames abhängig.

Eine weitere Überlegung zur Planung ist die Verwaltung großer Cluster. Die Verwaltung eines kleinen Clusters ist einfacher als die eines großen. Aber die Verwaltung eines einzelnen großen Clusters ist immer noch einfacher als die vieler kleiner Cluster. Bei der Planung ist auch zu bedenken, wie die Anwendungen in der Clusterumgebung verwendet werden. Wenn zum Beispiel eine

einzig, umfangreiche und homogene Anwendung auf 16 Knoten betrieben wird, ist die Konfiguration als ein Cluster wahrscheinlich einfacher zu verwalten, als sie in acht Cluster mit jeweils zwei Knoten zu zerlegen. Wenn dieselben 16 Knoten viele verschiedene Anwendungen mit verschiedenen Netzwerken, Platten und Knotenanordnungsbeziehungen enthalten, ist es wahrscheinlich besser, die Knoten in kleinere Cluster zu gruppieren. Beachten Sie, daß Knoten nur nacheinander, d. h. nur einer gleichzeitig, starten und sich in einen HACMP-Cluster integrieren. Eine Konfiguration mit mehreren Clustern läßt sich schneller starten als ein einziger großer Cluster. HACMP ES unterstützt sowohl einzelne als auch mehrere Cluster, vorausgesetzt ein Knoten und der zugehörige Ausweichknoten befinden sich im selben Cluster.

Die Fehlerbehebung durch Funktionsübernahme von HACMP ES (*Failover*) ermöglicht eine vordefinierte Zuordnung einer Ressourcengruppe zu einem physischen Knoten (auch als hintereinanderschalten (*cascading*) bezeichnet). Die Funktionsübernahme ermöglicht außerdem eine gleitende (auch als *rotierende* bezeichnete) Zuordnung einer Ressourcengruppe zu einem physischen Knoten. IP-Adressen und externe Datenträgergruppen (Volume Groups) oder Dateisysteme (Filesystems) oder NFS-Dateisysteme und Anwendungs-Server innerhalb jeder Ressourcengruppe geben entweder eine Anwendung oder eine Anwendungskomponente an, die von HACMP ES zwischen den physischen Knoten durch Funktionsübernahme und Wiedereingliederung (Reintegration) beeinflußt werden können. Die Funktionsübernahme und Reintegration wird durch die Art der erstellten Ressourcengruppe und durch die Anzahl der in der Ressourcengruppe angelegten Knoten angegeben.

Betrachten Sie zum Beispiel eine DB2-Datenbankpartition (logischen Knoten). Wenn die Protokoll- und Tabellenbereichsbehälter auf externen Platten angelegt und andere Knoten mit diesen Platten verbunden würden, wäre es möglich, daß die anderen Knoten auf diese Platten zugreifen und die Datenbankpartition (auf einem Übernahmeknoten) neu starten. Gerade diese Art von Operation wird durch HACMP automatisiert. HACMP ES kann außerdem dazu verwendet werden, NFS-Dateisysteme wiederherzustellen, die von Hauptbenutzerverzeichnissen des DB2-Exemplars verwendet werden.

Lesen Sie sich im Rahmen Ihrer Planung für die Wiederherstellung mit DB2 UDB EEE die Dokumentation zu HACMP ES gut durch. Sie sollten die Handbücher zu Konzepten, Planung, Installation und Verwaltung lesen und anschließend die Wiederherstellungsarchitektur für Ihre Umgebung entwerfen. Ermitteln Sie für die einzelnen Subsysteme, die Sie für die Wiederherstellung anhand bekannter möglicher Fehlerpunkte bestimmt haben, die benötigten HACMP-Cluster und die Wiederherstellungsknoten (entweder im Bereitschaftsmodus oder im Modus der gegenseitigen Übernahme). Dies dient als Ausgangspunkt für das Ausfüllen der HACMP-Arbeitsblätter, die der Dokumentation beigelegt sind.

Es ist ausdrücklich zu empfehlen, sowohl Platten als auch Adapter in der externen Plattenkonfiguration zu spiegeln. Bei physischen DB2-Knoten, die für HACMP konfiguriert sind, muß sorgfältig sichergestellt werden, daß die Knoten in der Datenträgergruppe von den gemeinsamen externen Platten abweichen können. In einer Konfiguration für gegenseitige Übernahme macht diese Anordnung eine zusätzliche Planung erforderlich, so daß die zu Paaren verbundenen Knoten ohne Konflikt auf die Datenträgergruppen des jeweils anderen zugreifen können. Für DB2 UDB EEE bedeutet dies, daß alle Behälternamen über alle Datenbanken hinweg eindeutig sein müssen.

Eine Methode zur Sicherstellung der Eindeutigkeit besteht darin, die Partitionsnummer als Teil des Namens zu verwenden. Sie können einen Knotenausdruck für die Syntax von Behälterzeichenfolgen angeben, wenn Sie SMS- oder DMS-Behälter erstellen. Wenn Sie den Ausdruck angeben, kann die Knotennummer Teil des Behälternamens sein, oder, wenn Sie zusätzliche Argumente angeben, können die Ergebnisse dieser Argumente Teil des Behälternamens sein. Verwenden Sie das Argument " \$N" ([leerzeichen]\$N) zur Angabe des Knotenausdrucks. Das Argument muß an das Ende der Behälterzeichenfolge gesetzt werden und kann nur in einem der folgenden Formate verwendet werden:

Tabelle 24. Argumente zur Behältererstellung. Als Knotennummer wird fünf (5) angenommen.

Syntax	Beispiel	Wert
[leerzeichen]\$N	" \$N"	5
[leerzeichen]\$N+[zahl]	" \$N+1011"	1016
[leerzeichen]\$N%[zahl]	" \$N%3"	2
[leerzeichen]\$N+[zahl]%[zahl]	" \$N+12%13"	4
[leerzeichen]\$N%[zahl]+[zahl]	" \$N%3+20"	22
Anmerkungen:		
1. % bedeutet Modulus.		
2. In allen Fällen werden die Operatoren von links nach rechts ausgewertet.		

Es folgen einige Beispiele zur Erstellung von Behältern mit Hilfe dieses Spezialarguments:

- Erstellen von Behältern zur Verwendung auf einem Zweiknotensystem

```
CREATE TABLESPACE TS1 MANAGED BY DATABASE USING
(device '/dev/rcont $N' 20000)
```

Die folgenden Behälter würden verwendet:

```
/dev/rcont0 - auf Knoten 0
/dev/rcont1 - auf Knoten 1
```

- Erstellen von Behältern zur Verwendung auf einem Vierknotensystem

```
CREATE TABLESPACE TS2 MANAGED BY DATABASE USING  
  (file '/DB2/containers/TS2/container $N+100' 10000)
```

Die folgenden Behälter würden verwendet:

```
/DB2/containers/TS2/container100 - auf Knoten 0  
/DB2/containers/TS2/container101 - auf Knoten 1  
/DB2/containers/TS2/container102 - auf Knoten 2  
/DB2/containers/TS2/container103 - auf Knoten 3
```

- Erstellen von Behältern zur Verwendung auf einem Zweiknotensystem

```
CREATE TABLESPACE TS3 MANAGED BY SYSTEM USING  
  ('/TS3/cont $N%2, '/TS3/cont $N%2+2')
```

Die folgenden Behälter würden verwendet:

```
/TS3/cont0 - auf Knoten 0  
/TS3/cont2 - auf Knoten 0  
/TS3/cont1 - auf Knoten 1  
/TS3/cont3 - auf Knoten 1
```

Abb. 46 auf Seite 250 und Abb. 47 auf Seite 251 zeigen ein Beispiel für die Konfiguration eines DB2-SSA-E/A-Subsystems sowie einige Gesichtspunkte der Planung, die zur Sicherstellung einer hochverfügbaren Konfiguration externer Platten und der Möglichkeit des konfliktfreien Zugriffs auf alle Datenträgergruppen gehört.

DB2-SSA-E/A-Subsystemkonfiguration - Kein einzelner Fehlerpunkt

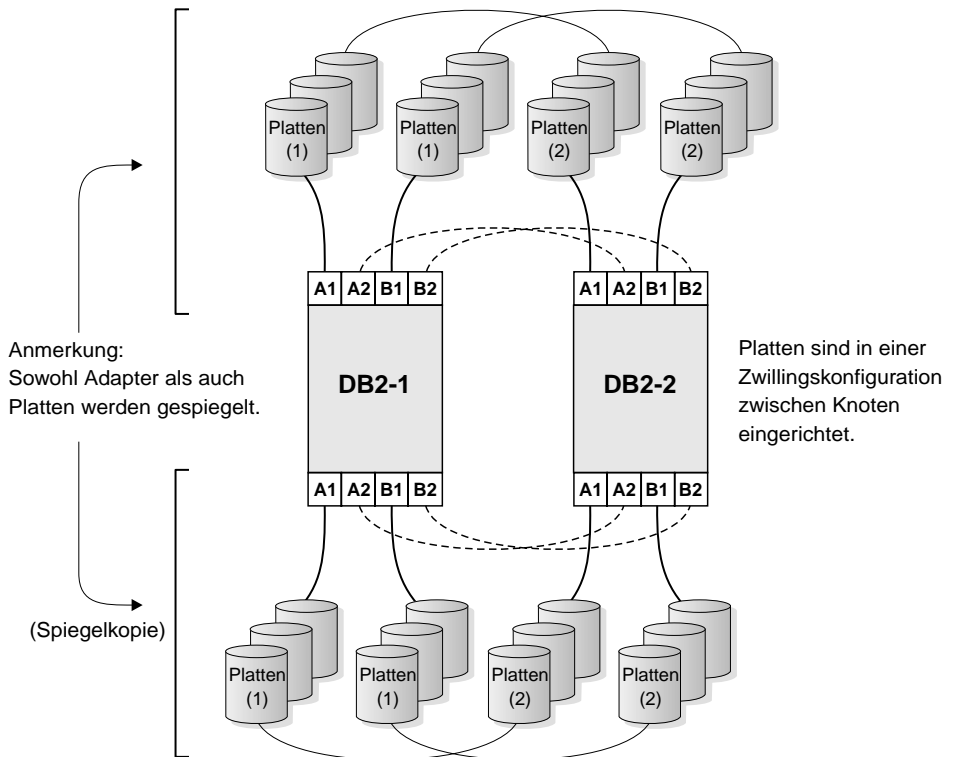


Abbildung 46. Kein einzelner Fehlerpunkt

DB2-SSA-E/A-Subsystemkonfiguration - Konfiguration von Datenträgergruppe (vg) und logischer Datenträger (lv)

DB2-Datenbank testdata für Dateisystem /database exemplarname powertp

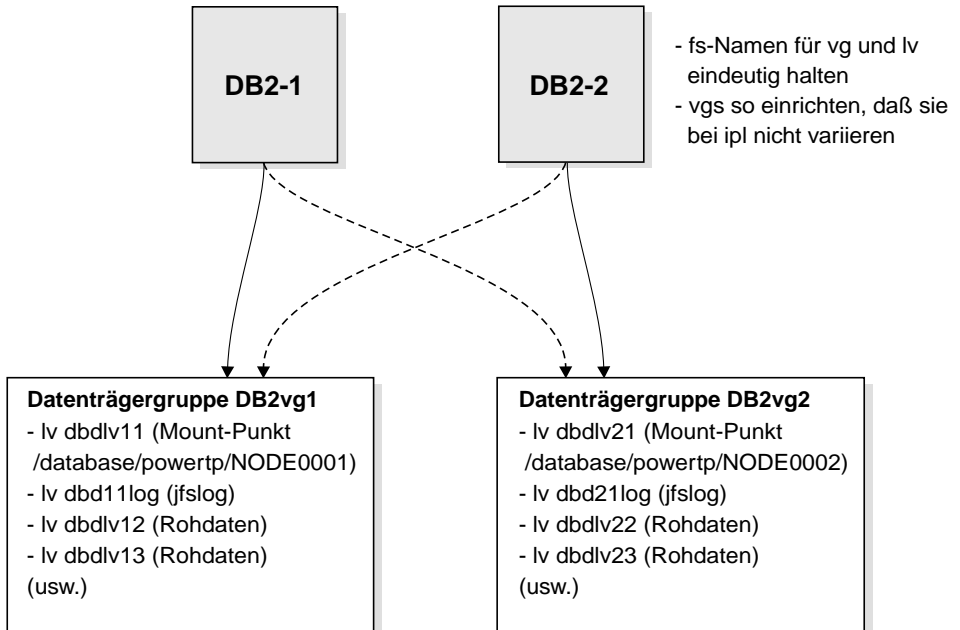


Abbildung 47. Konfiguration von Datenträgergruppe und logischem Datenträger

Konfigurieren einer DB2-Datenbankpartition

Nach der Konfiguration werden die Datenbankpartitionen in einem Exemplar durch HACMP ES nacheinander, ein physischer Knoten nach dem anderen gestartet. Mehrere Cluster werden empfohlen, wenn parallele DB2-Konfigurationen mit mehr als vier Knoten gestartet werden sollen. Beachten Sie, daß es in einer parallelen DB2-Konfiguration mit 64 Knoten schneller ist, 32 HACMP-Cluster mit zwei Knoten als vier Cluster mit 16 Knoten parallel zu starten.

Eine Prozedurdatei `rc.db2pe` gehört zum Lieferumfang von DB2 UDB EEE (und wird auf jedem Knoten in `/usr/bin` installiert), um Hilfestellung bei der Konfiguration für HACMP ES-Funktionsübernahme bzw. -Wiederherstellung auf Bereitschaftsknoten (Hot Standby) bzw. Knoten mit gegenseitiger Übernahme (Mutual Takeover) zu geben. Zusätzlich können aus `rc.db2pe` heraus DB2-Pufferpoolgrößen während der Funktionsübernahme bei Konfigurationen

mit gegenseitiger Übernahme angepaßt werden. (Pufferpoolgrößen müssen konfiguriert werden, um einen ordnungsgemäßen Betrieb sicherzustellen, wenn zwei Datenbankpartitionen auf demselben physischen Knoten ausgeführt werden.)

Wenn Sie einen Anwendungs-Server in einer HACMP-Konfiguration einer DB2-Datenbankpartition erstellen, geben Sie auf folgende Weise `rc.db2pe` als Start- und Stopprozedur an:

```
/usr/bin/rc.db2pe <exemplar> <dpn> <sekundäre dpn> start <switch>  
/usr/bin/rc.db2pe <exemplar> <dpn> <sekundäre dpn> stop <switch>
```

Dabei gilt folgendes:

<exemplar> ist der Exemplarname.

<dpn> ist die Datenbankpartitionsnummer.

<sekundäre dpn> ist die Datenbankpartitionsnummer des Pendants Konfigurationen mit gegenseitiger Übernahme. In Bereitschaftskonfigurationen gleich <dpn>.

<switch> ist in der Regel leer. Ist dies der Fall, zeigt dieser Parameter an, daß das SP-Switch-Netzwerk für das Feld *hostname* in der Datei *db2nodes.cfg* verwendet wird (der gesamte Verkehr für DB2 wird über den SP-Switch geleitet).

Falls nicht leer, ist der verwendete Name der Host-Name des zu verwendenden SP-Knotens.

Der DB2-Befehl `LIST DATABASE DIRECTORY` wird innerhalb von `rc.db2pe` zum Auffinden aller Datenbanken verwendet, die für diese Datenbankpartition konfiguriert sind. Die Prozedurdatei sucht anschließend die Dateien `/usr/bin/reg.parms.DATENBANK` und `/usr/bin/failover.parms.DATENBANK`, wobei `DATENBANK` jeweils die Datenbanken darstellt, die für diese Datenbankpartition konfiguriert sind. In einer Konfiguration zur gegenseitigen Übernahme ist es empfehlenswert, diese Parameterdateien `reg.parms.xxx` und `failover.parms.xxx` zu erstellen. In der Datei `failover.parms.xxx` sollten die Einstellungen für `BUFFPAGE`, `DBHEAP` und ggf. anderen Parametern, die die Pufferpools betreffen, angepaßt werden, um der Möglichkeit, daß mehr als ein Pufferpool vorhanden sein kann, Rechnung zu tragen. Die Beispieldateien `reg.parms.SAMPLE` und `failover.parms.SAMPLE` sind für Sie vorbereitet.

Einer der wichtigen Parameter in dieser Umgebung ist der Konfigurationsparameter `start_stop_time` des Datenbankmanagers, dessen Standardwert 10 Minuten ist. Die Prozedur `rc.db2pe` setzt diesen Parameter jedoch auf 2 Minuten. Sie sollten diesen Parameter durch `rc.db2pe` auf einen Wert von 10 Minuten oder etwas mehr setzen. In diesem Kontext ist die angegebene Zeitdauer das Intervall zwischen dem Ausfall der Partition und der Wiederherstellung der Partition. Wenn in den Anwendungen, die auf einer Partition ausgeführt werden, häufig `COMMIT`-Anforderungen abgesetzt werden, sollten zehn Minuten nach dem Ausfall einer Datenbankpartition ausreichen, um nicht festgeschriebene Transaktionen rückgängig zu machen (`ROLLBACK`) und

einen Konsistenzzustand in der Datenbank in dieser Partition zu erreichen. Wenn Sie eine hohe Auslastung oder viele Partitionen haben, müssen Sie die Zeitdauer eventuell verlängern, um die Wahrscheinlichkeit zu verringern, daß Zeitlimitüberschreitungen auftreten, bevor die ROLLBACK-Operation beendet ist.

Die folgenden Beispiele beschreiben eine Konfiguration im Bereitschaftsmodus (Hot Standby) und eine Konfiguration zur gegenseitigen Übernahme (Mutual Takeover). In beiden Beispielen enthalten die Ressourcengruppen eine Service-IP-Switch-Aliasadresse. Diese Adresse dient zu folgenden Zwecken:

- NFS-Zugriff auf einen Datei-Server für die Dateisysteme des DB2-Exemplareigners
- Sonstiger Client-Zugriff, der für den Fall einer Funktionsübernahme, einer TSM-Verbindung (Tivoli Storage Manager, früher ADSM) oder einer ähnlichen Operation aufrechterhalten werden muß

Wenn in Ihrer Implementierung diese Aliasnamen nicht erforderlich sind, können sie entfernt werden. Wenn sie entfernt werden, stellen Sie sicher, daß in der Prozedurdatei rc.db2pe der Parameter *MOUNT_NFS* auf den Wert *N0* gesetzt wird.

Beispiel für ein Konfiguration im Bereitschaftsmodus (Hot Standby)

Dieses Beispiel geht davon aus, daß eine Konfiguration im Bereitschaftsmodus zwischen den physischen Knoten 1 und 2 besteht und daß der Name des DB2-Exemplars POWERTP lautet. Die Datenbankpartition ist 1, der Name der Datenbank TESTDATA, die sich im Dateisystem /database befindet.

```
Resource group name: db2_dp_1
Node Relationship: cascading
Participating nodenames: node1_eth, node2_eth
Service_IP_label: nfs_switch_1 (<<< dies ist die Switch-Aliasadresse)
Filesystems: /database/powerntp/NODE0001
Volume Groups: DB2vg1
Application Servers: db2_dp1_app
Application Server Start Script: /usr/bin/rc.db2pe powerntp 1 1 start
Application Server Stop Script: /usr/bin/rc.db2pe powerntp 1 1 stop
```

Beispiel für eine Konfiguration für gegenseitige Übernahme (Mutual Takeover)

Dieses Beispiel geht davon aus, daß eine Konfiguration für gegenseitige Übernahme zwischen den physischen Knoten 1 und 2 besteht und daß der Name des DB2-Exemplars POWERTP lautet. Die Datenbankpartitionen sind 1 und 2, der Name der Datenbank ist TESTDATA, die sich im Dateisystem /database befindet.

```
Resource group name: db2_dp_1
Node Relationship: cascading
Participating nodenames: node1_eth, node2_eth
Service_IP_label: nfs_switch_1 (<<< dies ist die Switch-Aliasadresse)
Filesystems: /database/powerntp/NODE0001
```

```

Volume Groups: DB2vg1
Application Servers: db2_dp1_app
Application Server Start Script: /usr/bin/rc.db2pe powertp 1 2 start
Application Server Stop Script: /usr/bin/rc.db2pe powertp 1 2 stop

Resource group name: db2_pd_2
Node Relationship: cascading
Participating nodenames: node2_eth, node1_eth
Service_IP_label: nfs_switch_2 (<<< dies ist die Switch-Aliasadresse)
Filesystems: /database/powertp/NODE0002
Volume Groups: DB2vg2
Application Servers: db2_dp2_app
Application Server Start Script: /usr/bin/rc.db2pe powertp 2 1 start
Application Server Stop Script: /usr/bin/rc.db2pe powertp 2 1 stop

```

Konfiguration eines NFS-Server-Knotens

Die Prozedur rc.db2pe kann außerdem dazu verwendet werden, über NFS angehängte Verzeichnisse paralleler DB2-Exemplarbenutzerverzeichnisse verfügbar zu machen. Dies kann erreicht werden, indem der Parameter *MOUNT_NFS* in der Prozedurdatei rc.db2pe auf den Wert YES gesetzt wird und das NFS-Übernahme-Server-Paar wie folgt konfiguriert wird:

- Konfigurieren Sie das Benutzerverzeichnis (Home), und exportieren Sie es als "root" mit Hilfe von /etc/exports und des Befehls **exportfs** an die IP-Adresse, die auf den Knoten im selben Teilnetz wie die IP-Adresse des NFS-Servers verwendet wird. Geben Sie sowohl die HACMP-Boot-Adresse als auch die Serviceadresse an. Die IP-Adresse des NFS-Servers ist die gleiche Adresse wie die Serviceadresse in HACMP, die von einem Ausweichknoten übernommen werden kann. Das Benutzerverzeichnis (Home) des DB2-Exemplareigners sollte direkt über NFS angehängt (Mount) und nicht automatisch angehängt (Automount) werden. (Die Verwendung der Automount-Einrichtung wird von den Prozeduren nicht unterstützt, wenn es sich um das Benutzerverzeichnis des DB2-Exemplareigners handelt.)
- Erstellen Sie über SMIT oder eine Basiskonfiguration einen separaten Eintrag in /etc/filesystems für dieses Dateisystem, so daß alle Knoten in der parallelen DB2-Gruppierung, einschließlich des Datei-Servers, mit Hilfe des NFS-Dateisystembefehls (file system) hängen können.

Zum Beispiel kann ein JFS-Dateisystem /nfshome an alle Knoten als /dbhome exportiert werden. Jeder Knoten erstellt ein NFS-Dateisystem /dbname, das dem Dateisystem nfs_server:/nfshome entspricht. Daher wäre das Benutzerverzeichnis des DB2-Exemplareigners /dbhome/powertp, wenn der Exemplarname "powertp" wäre.

Stellen Sie sicher, daß in /etc/filesystems die NFS-Parameter "hard", "bg", "intr" und "rw" für das Anhängen (Mount) verwendet werden.

- Stellen Sie sicher, daß die Definitionen des DB2-Exemplareigners, die dem Benutzerverzeichnis /dbhome/powertp in /etc/passwd zugeordnet sind, auf allen Knoten identisch sind.

Die Benutzerdefinitionen in einer SP-Umgebung werden typischerweise auf der Steuer-Workstation (control workstation) erstellt, und "supper" oder "pcp" werden zur Verteilung von /etc/passwd, /etc/security/passwd, /etc/security/user und /etc/security/group an alle Knoten verwendet.

- Konfigurieren Sie *nicht* die zu exportierenden NFS-Dateisysteme ("nfs_filesystems to export") in HACMP-Ressourcengruppen für die Datenträgergruppe und das Dateisystem (Filesystem), das exportiert wird. Konfigurieren Sie diese Merkmale normal für NFS. Die Prozeduren für den NFS-Server steuern das Exportieren der Dateisysteme.
- Stellen Sie sicher, daß die Hauptgerätenummer der Datenträgergruppe, auf dem sich das Dateisystem befindet, auf dem Primärknoten und dem Übernahmeknoten gleich sind. Dies wird durch die Verwendung von **importvg** mit der Option -V erreicht.
- Überprüfen Sie, ob der Parameter *MOUNT_NFS* in der Prozedurdatei rc.db2pe auf den Wert YES gesetzt ist und ob jeder Knoten das anzuhängende NFS-Dateisystem in /etc/filesystems hat. Andernfalls ist rc.db2pe nicht in der Lage, das Dateisystem anzuhängen und DB2 zu starten.
- Wenn der DB2-Exemplareigner bereits erstellt wurde und Sie die Benutzerverzeichnisstruktur in das Dateisystem, das Sie erstellen, kopieren, stellen Sie sicher, daß Sie für das Verzeichnis den Befehl **tar** (-cvf) verwenden. Dadurch bleiben symbolische Verbindungen erhalten.
- Vergessen Sie nicht, sowohl die Adapter als auch die Platten für die logischen Datenträger sowie die Dateisystemprotokolle des Dateisystems, das Sie erstellen, zu spiegeln.

Beispiel für eine NFS-Server-Übernahmekonfiguration

Dieses Beispiel geht davon aus, daß es ein NFS-Server-Dateisystem /nfshome in der Datenträgergruppe nsvg über die IP-Adresse "nfs_server" gibt. Der DB2-Exemplarname ist POWERTP und das Ausgangsverzeichnis /dbhome/powertp.

```
Resource group name: nfs_server
Node Relationship: cascading
Participating nodenames: node1_eth, node2_eth
Service_IP_label: nfs_server (<<< dies ist die Switch-Aliasadresse)
Filesystems: /nfshome
Volume Groups: nsvg
Application Servers: nfs_server_app
Application Server Start Script: /usr/bin/rc.db2pe powertp NFS SERVER start
Application Server Stop Script: /usr/bin/rc.db2pe powertp NFS SERVER stop
```

In diesem Beispiel gilt:

- /etc/filesystems auf allen Knoten würde einen Eintrag für /dbhome als anzuhängendes nfs_server:/nfshome enthalten. nfs_server ist eine Service-IP-Switch-Aliasadresse.

- `/etc/exports` auf dem Knoten `nfs_server` und dem Ausweichknoten würde die Boot-Adresse und Serviceadresse einschließen und einen Eintrag für `/nfsfs -root=nfs_switch_1, nfs_switch_2, ...` enthalten.

Überlegungen zur Konfiguration des SP-Switch

Bei der Implementierung von HACMP ES mit dem SP-Switch sind folgende Punkte zu beachten:

- Auf dem SP-Switch gibt es "Basisadressen" und "Aliasadressen". Die Basisadressen sind diejenigen, die im SP System Data Repository (SDR) definiert sind und durch `rc.switch` konfiguriert werden, wenn das System "gebootet" wird. Die Aliasadressen sind IP-Adressen, die neben der Basisadresse in die `css0`-Schnittstelle mit Hilfe des Befehls **ifconfig** mit einem Aliasattribut konfiguriert werden. Beispiel:

```
ifconfig css0 inet alias sw_alias_1 up
```

- Bei der Konfiguration der DB2-Datei `db2nodes.cfg` sollten sowohl für das Feld "hostname" als auch für das Feld "netname" Namen der SP-Switch-Basis-IP-Adressen verwendet werden. Die Switch-IP-Aliasadressen dienen *nur* der Erhaltung der NFS-Konnektivität. Die DB2-Funktionsübernahme wird durch den Neustart von DB2 mit Hilfe des Befehls **db2 (RESTART) erreicht (der db2nodes.cfg aktualisiert)**.
- Verwechseln Sie die Switch-Adressen nicht mit den Aliasnamen in `etc/hosts`. Die SP-Switch-Adressen und die SP-Switch-IP-Aliasadressen sind entweder in `etc/hosts` oder in DNS reale Adressen. Die Aliasadressen für einen Switch sind nicht andere Namen für die SP-Switch-Basisadresse. Jeder hat eine eigene separate Adresse.
- Die SP-Switch-Basisadressen sind immer auf einem Knoten vorhanden, wenn er aktiv ist. HACMP ES konfiguriert und versetzt diese Adressen nicht zwischen Knoten.
- Wenn Sie beabsichtigen, SP-Switch-Aliasadressen zu verwenden, konfigurieren Sie diese für HACMP als Boot- und Serviceadressen für "Heartbeating" und IP-Adreßübernahme. Wenn Sie nicht beabsichtigen, SP-Switch-Aliasadressen zu verwenden, konfigurieren Sie die SP-Switch-Basisadresse für HACMP als Serviceadresse *nur* für "Heartbeating" (keine IP-Adreßübernahme). Konfigurieren Sie in keiner Konfiguration Aliasadressen *und* die Switch-Basisadresse. Eine solche Konfiguration wird von HACMP ES nicht unterstützt.
- Nur die SP-Switch-Aliasadressen (und nicht die SP-Switch-Basisadressen) werden zwischen Knoten bei einer IP-Übernahmekonfiguration versetzt.
- Der Bedarf an Aliasnamen für SP-Switch ergibt sich aus der Tatsache, daß es nur einen SP-Switch-Adapter pro Knoten geben kann. Die Verwendung von Aliasadressen ermöglicht es, daß ein Knoten die Switch-IP-Aliasadresse eines anderen Knotens übernehmen kann, ohne einen anderen Switch-Adapter hinzufügen zu müssen. Dies ist auf Knoten nützlich, auf denen Adapterplätze nur begrenzt vorhanden sind. Weitere Informationen zur

Durchführung von Wiederherstellungen nach Ausfällen von SP-Switch-Adaptoren finden Sie im Abschnitt zu Netzwerkausfällen unter „HACMP ES-Prozedurdateien“ auf Seite 271.

- Wenn Sie den SP-Switch zur Übernahme der IP-Adresse konfigurieren, müssen Sie zwei zusätzliche IP-Aliasadressen pro Knoten erstellen: eine als Boot-Adresse und eine als Serviceadresse.
- Vergessen Sie nicht, "HPS" in der Definition des HACMP ES-Netzwerknamens für eine IP-Basisadresse bzw. eine IP-Aliasadresse für einen SP-Switch zu verwenden.
- `rc.cluster` in HACMP führt beim Starten von HACMP automatisch **ifconfig** für die SP-Switch-Boot-Adresse aus. Außer der Erstellung von IP-Adresse und Name sowie deren Definition für HACMP sind keine weiteren Konfigurationsschritte erforderlich.
- Der Knoten Eprimary des SP-Switch ist der Server, der die Befehle Estart, Efence und Eunfence implementiert. Die HACMP-Prozeduren versuchen, die Befehle Eunfence oder Estart für einen Knoten auszuführen, wenn HACMP gestartet wird, und den Switch verfügbar zu machen, wenn er als einer seiner Netzwerke definiert sein sollte. Aus diesem Grund ist sicherzustellen, daß der Eprimary-Knoten beim Starten von HACMP verfügbar ist. Der HACMP-Code wartet bis zu 12 Minuten auf die Durchführung einer Eprimary-Funktionsübernahme, bevor er mit einem Fehler die Verarbeitung beendet.
- Der Eprimary-Knoten des SP-Switch wird von SP PSSP (Parallel System Support Program) zwischen Knoten versetzt, nicht von HACMP. Wenn ein Eprimary-Knoten offline geschaltet wird, übergibt PSSP die Aufgaben des Eprimary-Knotens automatisch an einen Ausweichknoten. Das Switch-Netzwerk ist von dieser Änderung nicht betroffen und bleibt aktiv.

Beispiele für DB2-HACMP-Konfigurationen

Die folgenden Beispiele zeigen verschiedene Konfigurationen zur Unterstützung von Funktionsübernahmen und erläutern, was im Fall eines Ausfalls geschieht.

Im Fall von DB2-HACMP-Konfigurationen für gegenseitige Übernahme (Abb. 48 auf Seite 259, Abb. 49 auf Seite 260 und Abb. 50 auf Seite 261) gilt:

- HACMP-Adapter werden für Ethernet- sowie für SP-Switch-Boot- und SP-Switch-Aliasadressen definiert, Basisadressen bleiben unberührt. Beachten Sie, daß eine Zeichenfolge "HPS" im HACMP-Netzwerknamen zu verwenden ist.
- Das Verzeichnis `NFS_server/nfshome` wird als `/dbhome` auf allen Knoten über Switch-Aliasadressen angehängt.

- Die Datei `db2nodes.cfg` enthält die SP-Switch-Basisadressen. Die Datei `db2nodes.cfg` wird durch den Befehl **db2start** (RESTART) nach einer Funktionsübernahme für eine DB2-Datenbankpartition (logischen Knoten) geändert.
- Die Boot-Aliasadressen für die SP-Switches werden nicht gezeigt.
- Knoten können sich in verschiedenen SP-Frames befinden.

DB2-HACMP-Konfiguration: Gegenseitige Übernahme mit NFS-Funktionsübernahme

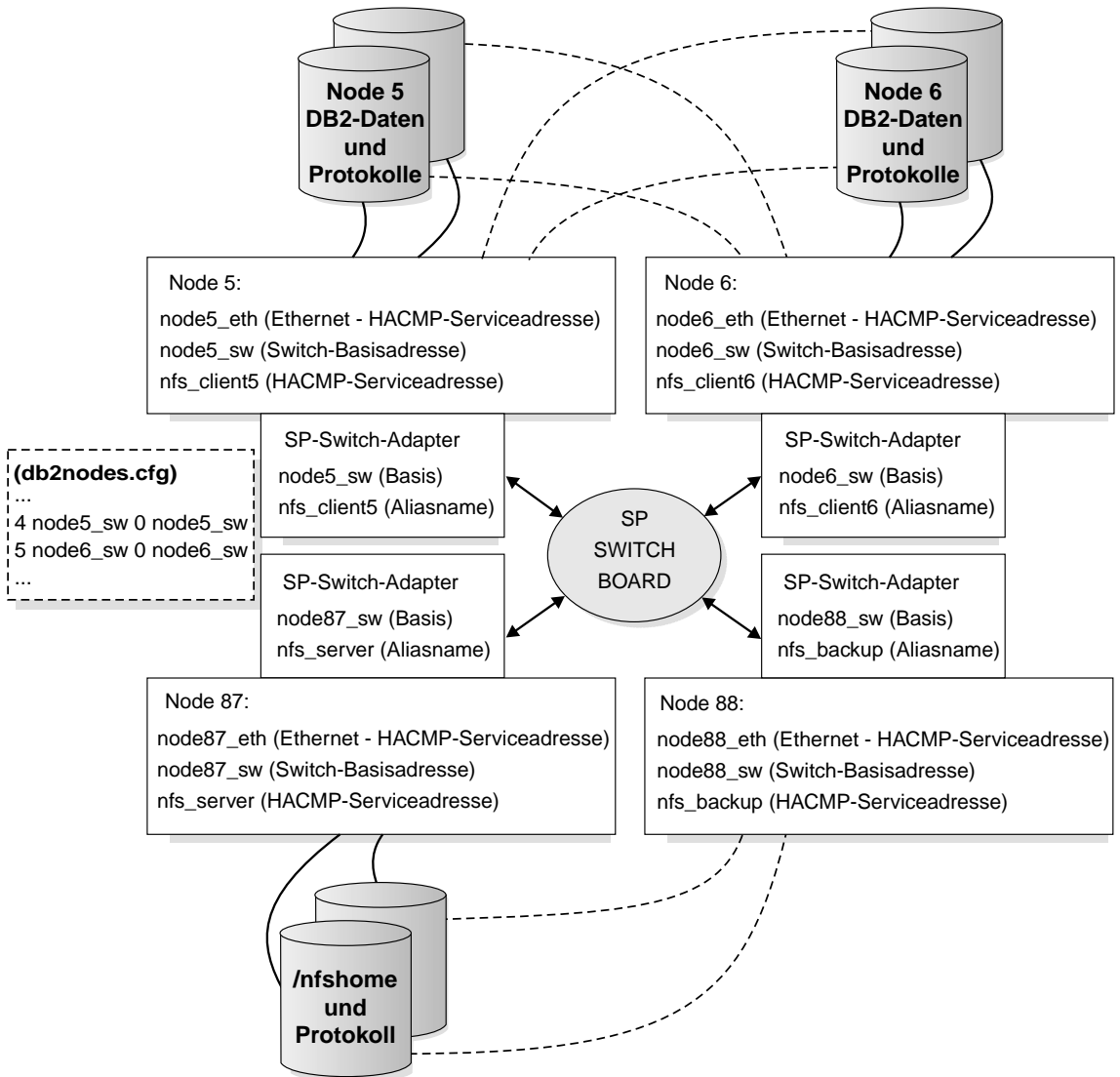


Abbildung 48. Gegenseitige Übernahme mit NFS-Funktionsübernahme - Normal

DB2-HACMP-Konfiguration: Gegenseitige Übernahme mit NFS-Funktionsübernahme - NFS-Funktionsübernahme

- nfs_server SP-Switch-Alias-IP-Adr. u. nfs angehängt. /nfshome von Node 87 auf 88 versetzt.
- SP-Switch-ARP-Code hat Funktionen zum Aktualisieren aller Switch-ARP-Caches bei dieser Übernahme.

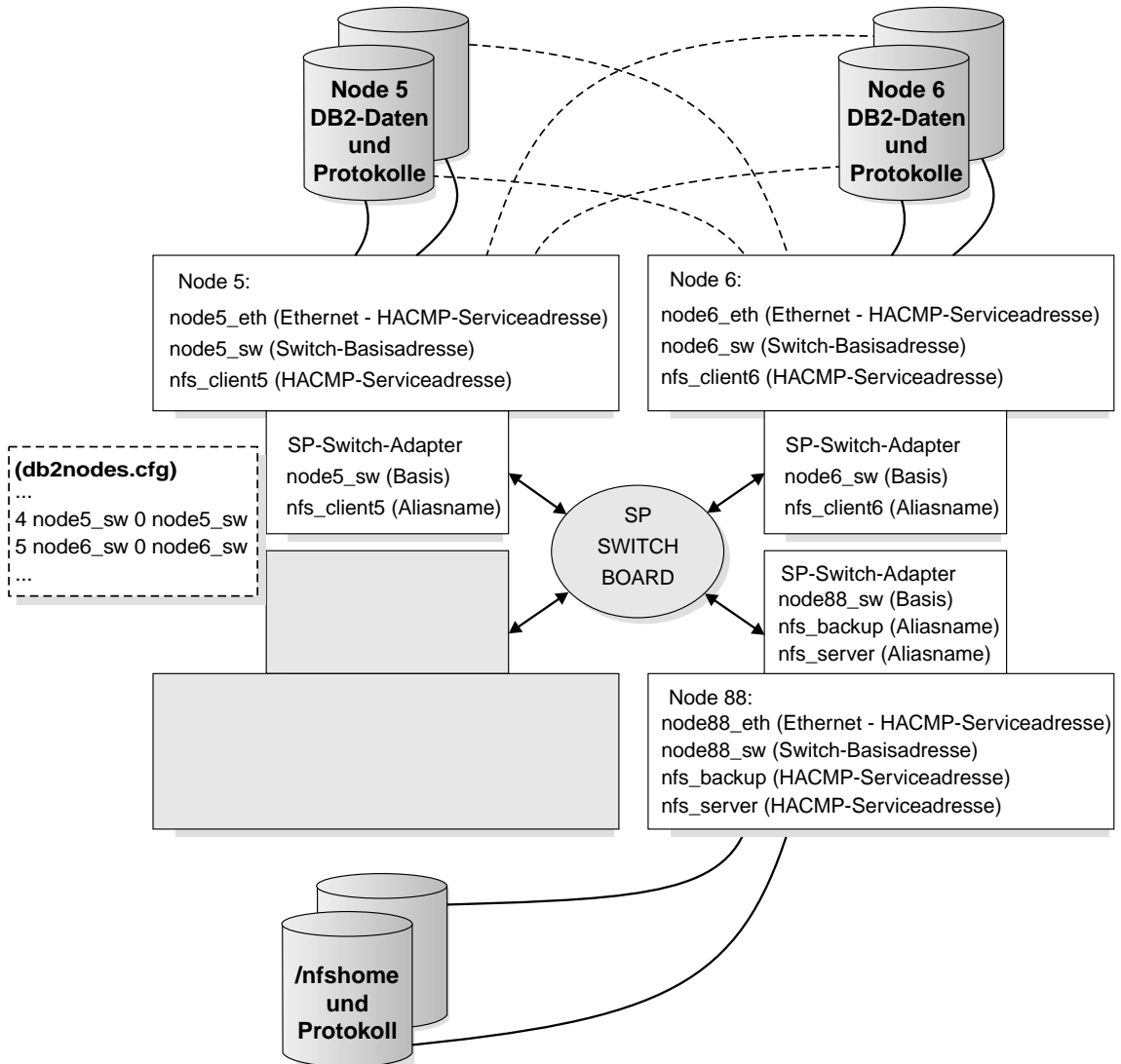


Abbildung 49. Gegenseitige Übernahme mit NFS-Funktionsübernahme - NFS-Funktionsübernahme

DB2-HACMP-Konfiguration:

Gegenseitige Übernahme mit NFS-Funktionsübernahme - DB2-Funktionsübernahme

- Übernahme der Switch-IP-Adresse ermöglicht anderen Servern (wie ADSM) das Beibehalten der Konnektivität.
- Node 5 führt zwei logische Knoten von DB2 aus.

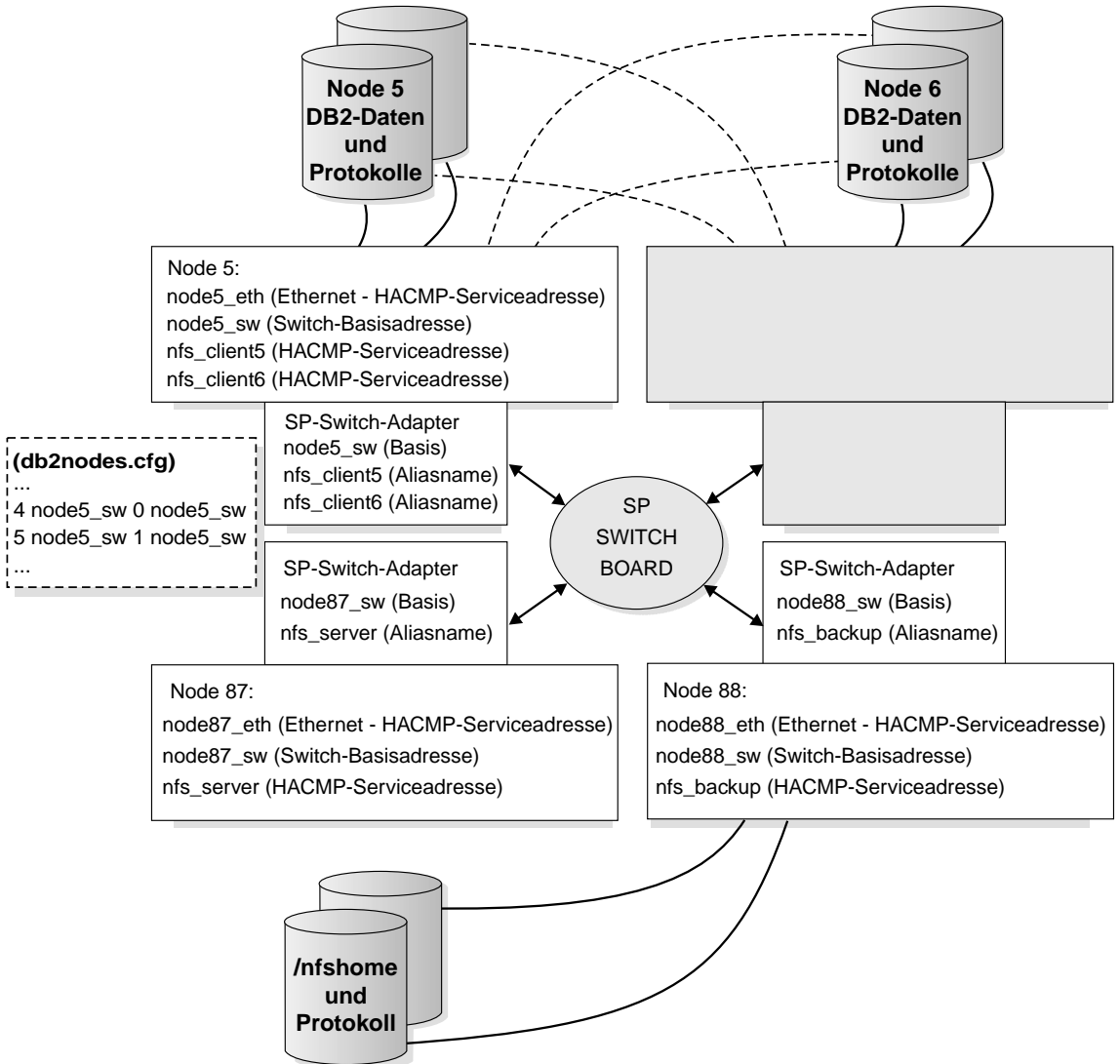


Abbildung 50. Gegenseitige Übernahme mit NFS-Funktionsübernahme - DB2-Funktionsübernahme

Im Fall von DB2-HACMP-Konfigurationen im Bereitschaftsmodus (Abb. 51 auf Seite 263 und Abb. 52 auf Seite 264) gilt:

- HACMP-Adapter werden für Ethernet- sowie für SP-Switch-Boot- und SP-Switch-Aliasadressen definiert, Basisadressen bleiben unberührt. Beachten Sie, daß eine Zeichenfolge "HPS" im HACMP-Netzwerknamen zu verwenden ist.
- Das Verzeichnis NFS_server/nfshome wird als /dbhome auf allen Knoten über Switch-Aliasadressen angehängt.
- Die Datei db2nodes.cfg enthält die SP-Switch-Basisadressen. Die Datei db2nodes.cfg wird durch den Befehl **db2start** (RESTART) nach einer Funktionsübernahme für eine DB2-Datenbankpartition (logischen Knoten) geändert.
- Die Boot-Aliasadressen für die SP-Switches werden nicht gezeigt.

DB2-HACMP-Konfiguration: Fehlertoleranter Modus mit NFS-Funktionsübernahme - Normal

Anmerkung: Ein Knoten im fehlertoleranten Modus kann je nach Plattenverkabelung mehrere Knoten sichern.

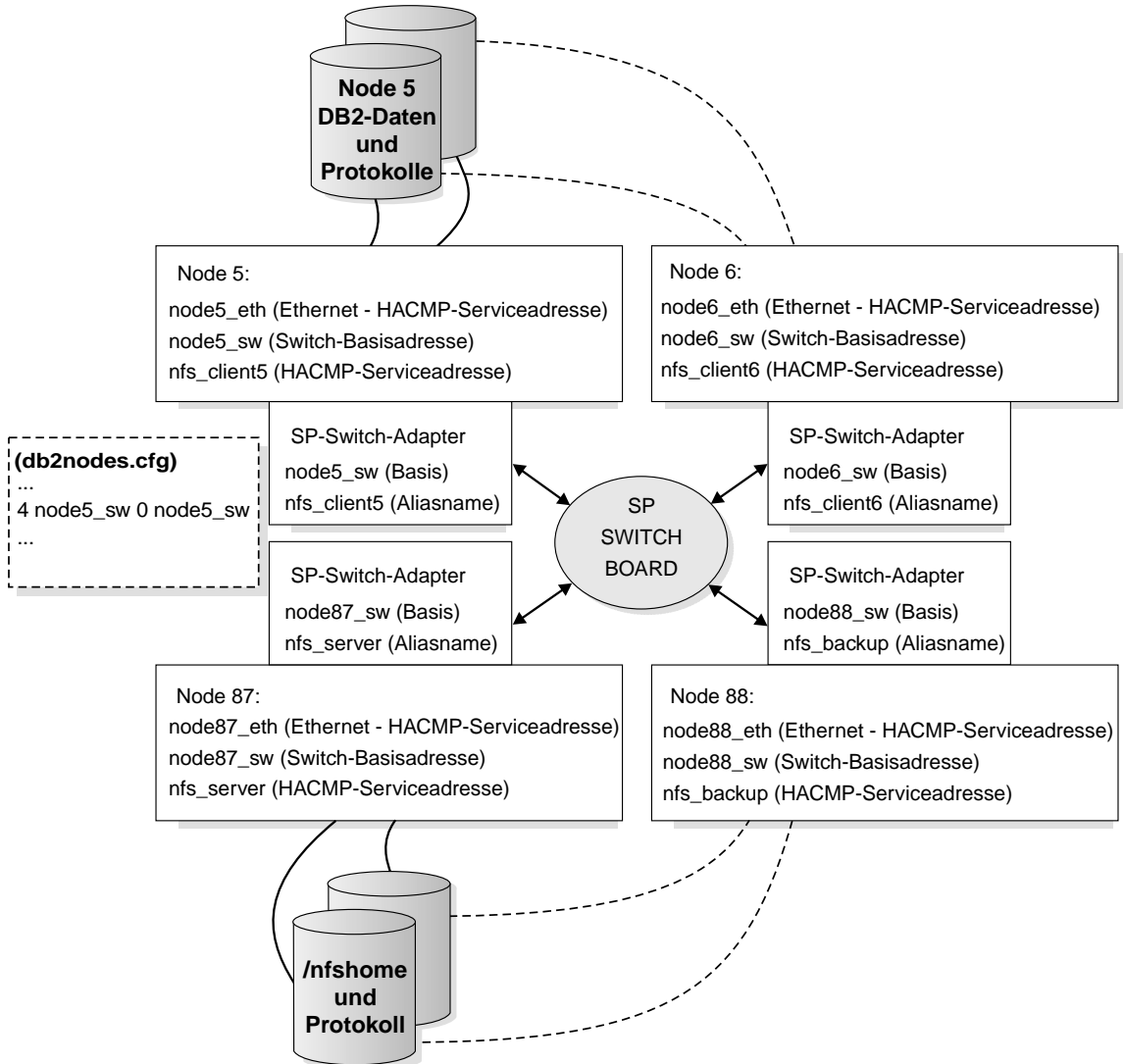


Abbildung 51. Bereitschaft mit NFS-Funktionsübernahme - Normal

DB2-HACMP-Konfiguration: Fehlertoleranter Modus mit NFS-Funktionsübernahme - DB2-Funktionsübernahme

Anmerkung: Ein Knoten im fehlertoleranten Modus kann je nach Plattenverkabelung mehrere Knoten sichern.

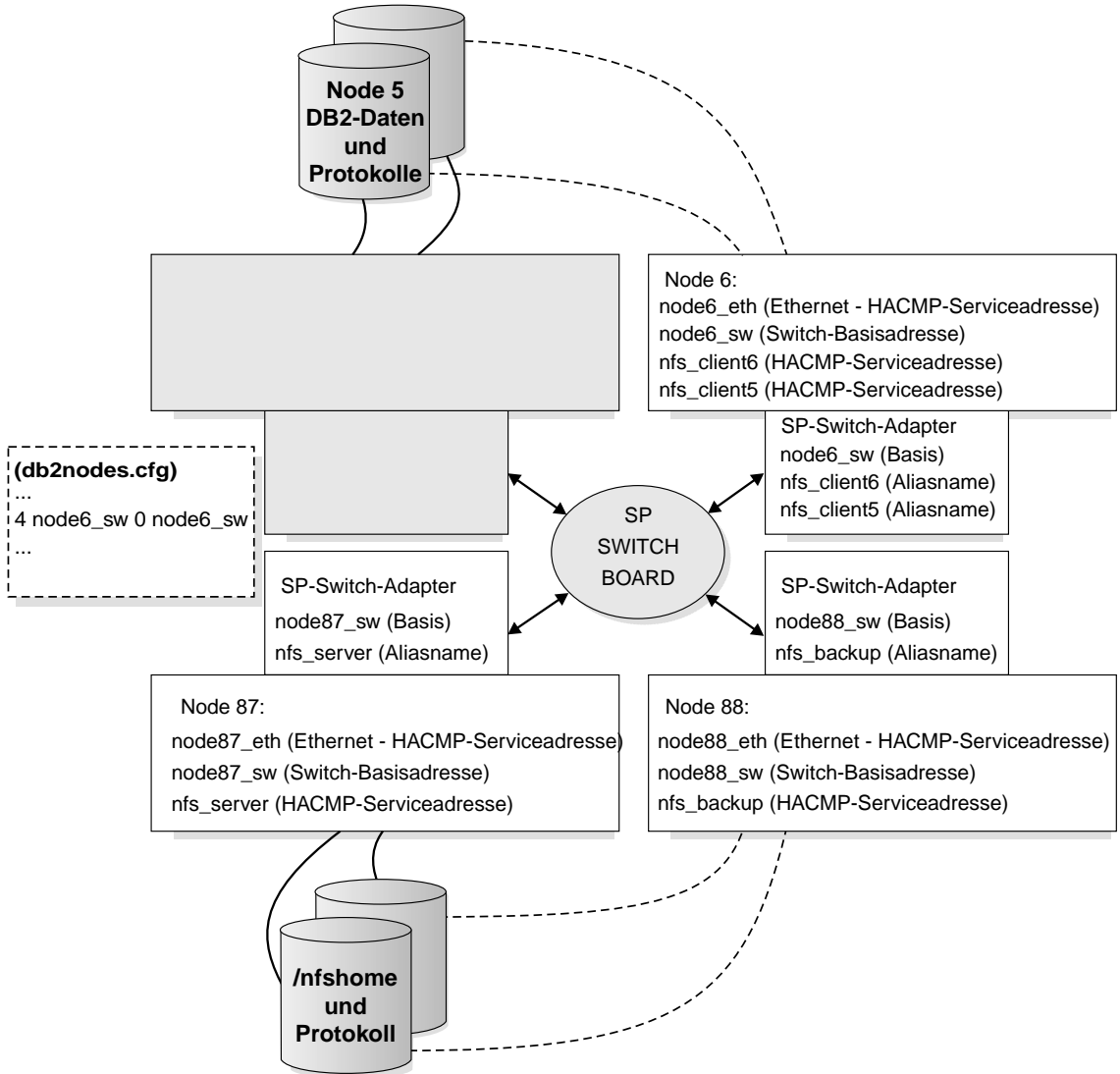


Abbildung 52. Bereitschaft mit NFS-Funktionsübernahme - DB2-Funktionsübernahme

Im Fall von DB2-HACMP-Konfigurationen für gegenseitige Übernahme ohne NFS-Übernahme (Abb. 53 auf Seite 265 und Abb. 54 auf Seite 266) gilt:

- HACMP-Adapter werden für Ethernet definiert und SP-Switch-Basisadressen. Beachten Sie, daß es, wenn Basisadressen als Serviceadressen für HACMP konfiguriert werden, keine Boot-Adresse (nur ein Überwachungssignal ("Heartbeat")) gibt. Vergessen Sie nicht, daß eine Zeichenfolge "HPS" im HACMP-Netzwerknamen für den SP-Switch zu verwenden ist.
- Die Datei `db2nodes.cfg` enthält die SP-Switch-Basisadressen. Die Datei `db2nodes.cfg` wird durch den Befehl **db2start** (RESTART) nach einer Funktionsübernahme für eine DB2-Datenbankpartition (logischen Knoten) geändert.
- Es werden keine NFS-Übernahmefunktionen gezeigt.
- Knoten können sich in verschiedenen SP-Frames befinden.

DB2-HACMP-Konfiguration: Gegenseitige Übernahme ohne NFS-Funktionsübernahme - Normal

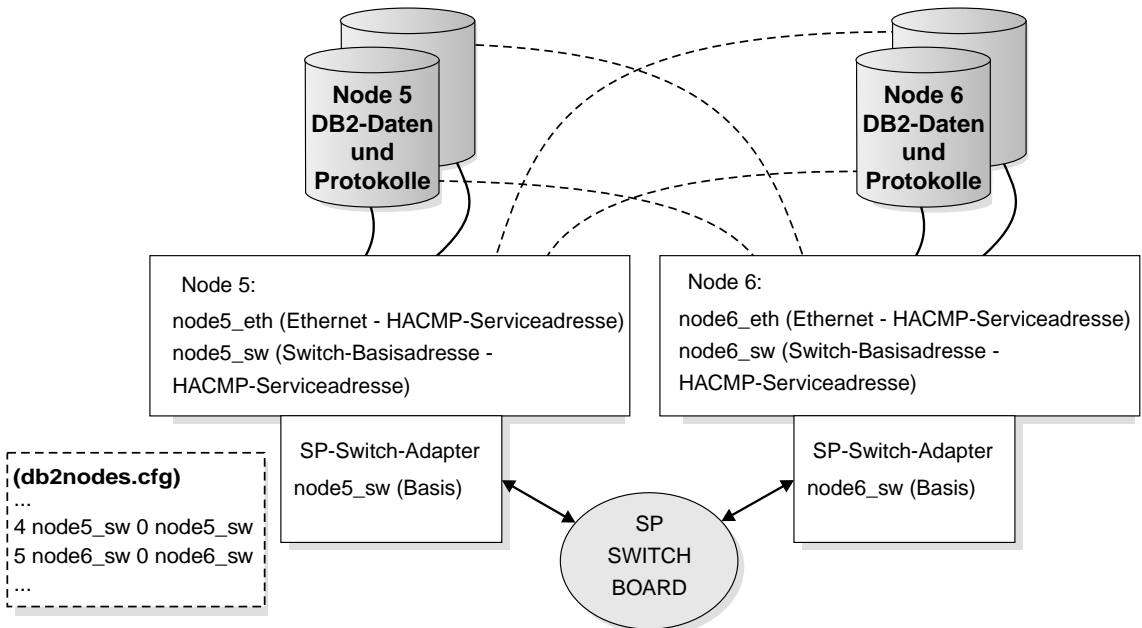


Abbildung 53. Gegenseitige Übernahme ohne NFS-Funktionsübernahme - Normal

DB2-HACMP-Konfiguration: Gegenseitige Übernahme ohne NFS Funktionsübernahme - DB2 Funktionsübernahme

- Node 5 führt zwei logische Knoten von DB2 aus.

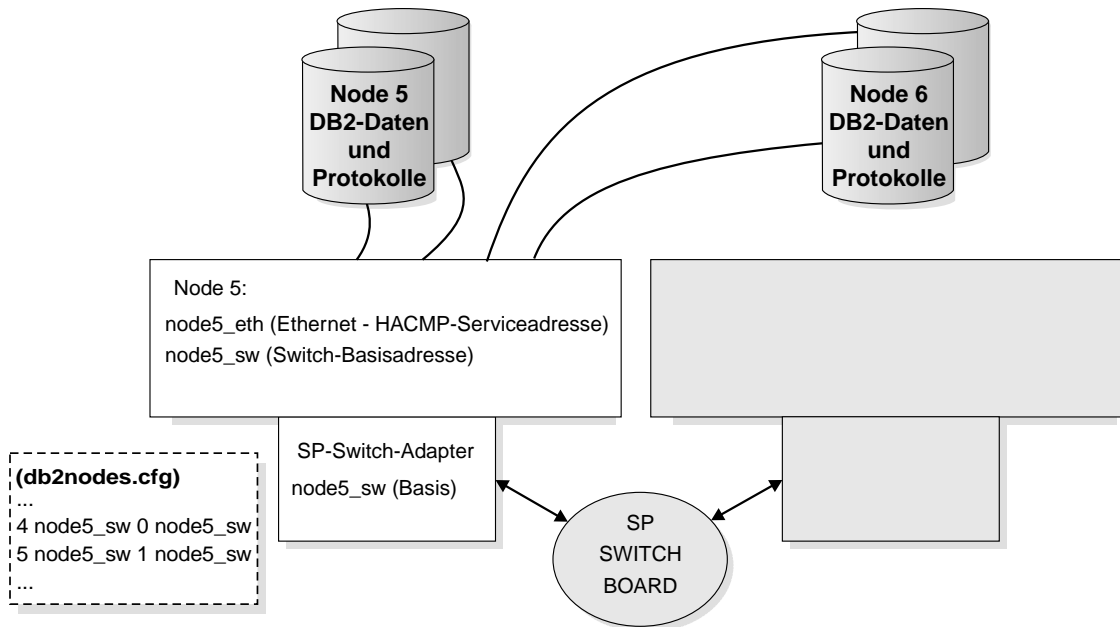


Abbildung 54. Gegenseitige Übernahme ohne NFS-Funktionsübernahme - DB2-Funktionsübernahme

Empfehlungen für den DB2-HACMP-Systemstart

Es wird empfohlen, HACMP nicht für den Start bei Systemstart in `/etc/inittab` anzugeben. HACMP sollte nach dem Booten der Knoten manuell gestartet werden. Dadurch wird eine ungestörte Wartung des ausgefallenen Knotens ermöglicht.

Betrachten Sie als Beispiel für eine störende Wartung den Fall, in dem auf einem Knoten ein Hardwarefehler mit Systemabsturz auftritt. Die Funktionsübernahme wird von HACMP automatisch eingeleitet und die Wiederherstellung erfolgreich durchgeführt. Allerdings muß der ausgefallene Knoten repariert werden. Wäre HACMP in `/etc/inittab` zum Starten bei erneutem Booten konfiguriert, würde dieser Knoten nach Abschluß des Bootens versuchen, sich erneut einzugliedern, was in diesem Fall jedoch nicht wünschenswert ist.

Zur unterbrechungsfreien Wartung sollte ein manuelles Starten von HACMP auf jedem Knoten in Betracht gezogen werden. Auf diese Weise können ausgefallene Knoten repariert und wieder integriert werden, ohne die anderen

Knoten zu beeinträchtigen. Die Prozedur `ha_cmd` wird zur Steuerung der HACMP-Befehle zum Starten und Stoppen über die Steuer-Workstation bereitgestellt.

Anmerkung: Wenn zum ersten Mal ein DB2-Exemplar erstellt wird, wird der folgende Eintrag an die Datei `/etc/inittab` angehängt:

```
rcdb2:2:once:/etc/rc.db2 > /dev/console 2>&1 # Autostart DB2 Services
```

Wenn HACMP oder HACMP ES aktiviert ist, aktualisieren Sie die Datei `/etc/inittab`, indem Sie die obige Zeile vor dem HACMP-Eintrag einordnen. Das folgende Beispiel zeigt einen HACMP-Eintrag in der Datei `/etc/inittab`:

```
clinit:a:wait:touch /usr/sbin/cluster/.telinit # HACMP for AIX
```

Der Eintrag muß der letzte Eintrag in der Datei `/etc/inittab` sein.

HACMP ES-Ereignisüberwachung und benutzerdefinierte Ereignisse

Das Herunterfahren von DB2-Datenbankpartitionen auf einem physischen AIX-Knoten, wenn der Seitenwechselbereich (Paging-Bereich) einen bestimmten Prozentsatz der Füllung erreicht, oder das erneute Starten einer DB2-Datenbankpartition bzw. die Einleitung einer Übernahmeoperation, wenn ein Prozeß auf einem bestimmten Knoten unterbrochen wird, sind zwei Beispiele für benutzerdefinierte Ereignisse. Beispiele, die benutzerdefinierte Ereignisse illustrieren, wie zum Beispiel das Herunterfahren einer Datenbankpartition und das Erzwingen eines Transaktionsabbruchs, um Seitenwechselbereich freizugeben, befinden sich im Unterverzeichnis `samples`.

Eine `rules`-Datei `/user/sbin/cluster/events/rules.hacmprd` enthält Definitionen von HACMP-Ereignissen. Jede Ereignisbeschreibung in dieser Datei besitzt die folgenden neun Komponenten:

- Ereignisname, der eindeutig sein muß.
- Status oder Qualifikationsmerkmal für das Ereignis. Der Ereignisname und der Status sind die Regelauslöser. HACMP ES Cluster Manager leitet die Wiederherstellung nur dann ein, wenn eine Regel mit einem Auslöser (Trigger) vorhanden ist, die dem Ereignisnamen und dem Status entsprechen.
- Ressourcenprogramm Pfad, eine vollständige Pfadangabe der Datei `xxx.rp`, die das Wiederherstellungsprogramm enthält.
- Wiederherstellungstyp. Diese Angabe ist für zukünftige Zwecke reserviert.
- Wiederherstellungsebene. Diese Angabe ist für zukünftige Zwecke reserviert.

- Ressourcenvariablenname, der für Ereignisse des Ereignismanagers (Event Manager) verwendet wird.
- Exemplarvektor (Instance vector), der für Ereignisse des Ereignismanagers verwendet wird. Dies ist eine Gruppe von Elementen der Form "name=wert". Die Werte identifizieren die Kopie der Ressource im System eindeutig und somit auch die Kopie der Ressourcenvariable.
- Vergleichselement, das für Ereignisse des Ereignismanagers verwendet wird. Dies ist ein Vergleichsausdruck zwischen einer Ressourcenvariablen und anderen Elementen. Wenn der Ausdruck wahr ist, generiert das Subsystem der Ereignisverwaltung (Event Management subsystem) ein Ereignis, um den Clustermanager und die entsprechende Anwendung zu benachrichtigen.
- Rearm-Vergleichselement, das für Ereignisse des Ereignismanagers verwendet wird. Dieses Vergleichselement dient zur Generierung eines Ereignisses, das den Status des primären Vergleichselements ändert. In der Regel stellt dieses Vergleichselement die Inversion des primären Vergleichselements dar. Es kann zusammen mit dem Ereignisvergleichselement auch dazu verwendet werden, einen oberen und einen unteren Grenzwert für eine bestimmte Bedingung festzulegen.

Jedes Objekt benötigt eine Zeile in der Ereignisdefinition, auch wenn die Zeile nicht verwendet wird. Wenn solche Zeilen gelöscht werden, kann HACMP ES Cluster Manager die Ereignisdefinition nicht korrekt analysieren, was zu einer Blockierung des Systems führen kann. Jede Zeile die mit einem Zeichen "#" beginnt, wird als Kommentarzeile behandelt.

Anmerkung: Die rules-Datei verlangt exakt neun Zeilen für jede Ereignisdefinition, wobei die Kommentarzeilen nicht mitgezählt werden. Beim Hinzufügen eines benutzerdefinierten Ereignisses am Ende der rules-Datei, muß unbedingt darauf geachtet werden, nicht benötigte Leerzeilen am Ende der Datei zu entfernen. Ansonsten kommt es zu einem Blockieren des Knotens.

Das folgende Beispiel zeigt eine Ereignisdefinition für das Ereignis node_up:

```
##### Beginning of the Event Definition: node_up
#
TE_JOIN_NODE
0
/usr/sbin/cluster/events/node_up.rp
2
0
# 6) Resource variable - only used for event management events

# 7) Instance vector - only used for event management events

# 8) Predicate - only used for event management events
```

```
# 9) Rearm predicate - only used for event management events  
##### End of the Event Definition: node_up
```

Dies ist nur ein Beispiel für die Ereignisdefinitionen, die sich in der Datei `rules.hacmprd` befinden können. In diesem Beispiel wird das Wiederherstellungsprogramm `/usr/sbin/cluster/events/node_up.rp` ausgeführt, wenn das Ereignis `node_up` eintritt. Werte sind für den Status, den Wiederherstellungstyp und die Wiederherstellungsebene angegeben. Es gibt vier leere Zeilen für Ressourcenvariable, Exemplarvariable, Vergleichselement (Predicate) und Rearm-Vergleichselement (Rearm predicate).

Sie können andere Ereignisse definieren, um auf andere als die Standardereignisse von HACMP ES zu reagieren. Zum Beispiel muß die Datei `rules.hacmprd` geändert werden, um das Ereignis zu definieren, daß das Dateisystem `/tmp` zu über 90 % gefüllt ist.

Zahlreiche Ereignisse sind in IBM Parallel System Support Program (PSSP) vordefiniert. Diese Ereignisse können (durch Verwendung innerhalb benutzerdefinierter Ereignisse) wie folgt genutzt werden:

1. Stoppen Sie den Cluster.
2. Editieren Sie die Datei `rules.hacmprd`. Fertigen Sie eine Sicherungskopie der Datei an, bevor Sie sie ändern. Fügen Sie das vordefinierte PSSP-Ereignis manuell hinzu. Wenn Sie Synchronisierungspunkte über alle Knoten im Cluster hinweg benötigen, verwenden Sie im Wiederherstellungsprogramm den Befehl **barrier**. (Weitere Informationen zum Befehl **barrier** und zur Synchronisierung von Wiederherstellungsprogrammen lesen Sie bitte in den HACMP-Handbüchern zu Konzepten, Installation und Verwaltung.)
3. Starten Sie den Cluster erneut. Die Datei `rules.hacmprd` wird im Hauptspeicher gespeichert, wenn Cluster Manager gestartet wird. Starten Sie alle Cluster erneut, um die Änderungen präzise zu implementieren. Es sollten sich keine inkonsistenten Angaben in der `rules`-Datei in einem Cluster befinden.
4. Cluster Manager verwendet alle Ereignisse in der Datei `rules.hacmprd`.

HACMP ES verwendet die PSSP-Ereigniserkennung zur Behandlung benutzerdefinierter Ereignisse. Das PSSP-Subsystem zur Ereignisverwaltung (Event Management) stellt eine umfassende Ereigniserkennung durch Überwachung verschiedener Hardware- und Softwareressourcen zur Verfügung.

Die Ressourcenstatus werden durch verschiedene Ressourcenvariablen dargestellt. Ressourcenbedingungen werden durch Ausdrücke wiedergegeben, die als Vergleichselemente (Predicates) bezeichnet werden.

Die Ereignisverwaltung (Event Management) empfängt Ressourcenvariablen vom Ressourcenmonitor (Resource Monitor), der den Status spezieller Systemressourcen beobachtet und diesen Status in verschiedene Ressourcenvariablen umsetzt. Diese Variablen werden in regelmäßigen Abständen an die Ereignisverwaltung (Event Management) übermittelt. Die Ereignisverwaltung wendet Vergleichselemente, die durch HACMP ES Cluster Manager in der Datei `rules.hacmprd` angegeben werden, für jede Ressourcenvariable an. Wenn das Vergleichselement als wahr ausgewertet wird, wird ein Ereignis generiert und an Cluster Manager gesendet. Cluster Manager initiiert das Wahlprotokoll, und die Wiederherstellungsprogrammdatei (`xxx.rp`) wird (nach Ereignispriorität) auf einer Gruppe von Knoten ausgeführt, die durch "node sets" im Wiederherstellungsprogramm angegeben sind.

Die Wiederherstellungsprogrammdatei (`xxx.rp`) besteht aus einer oder mehreren Wiederherstellungsprogrammzeilen. Jede Zeile ist auf das folgende Format festgelegt:

```
beziehung      auszuführender-befehl      erwarteter-status      NULL
```

Zwischen den einzelnen Werten in der Zeile muß mindestens ein Leerzeichen sein. "beziehung" ist ein Wert, der zu der Entscheidung herangezogen wird, welches Programm auf welchen Knoten auszuführen ist. Es werden drei Werte für die Beziehung unterstützt:

- All. Der angegebene Befehl oder das Programm wird auf allen Knoten des aktuellen HACMP-Clusters ausgeführt.
- Event. Der angegebene Befehl oder das Programm wird nur auf den Knoten ausgeführt, auf denen das Ereignis auftrat.
- Other. Der angegebene Befehl oder das Programm wird auf allen Knoten ausgeführt, auf denen das Ereignis *nicht* auftrat.

"auszuführender-befehl" ist eine in Anführungszeichen gesetzte Zeichenfolge mit oder ohne Angabe eines vollständigen Pfads zu einem ausführbaren Programm. Nur zum Lieferumfang von HACMP gehörende Ereignisprozeduren können eine relative Pfaddefinition verarbeiten. Andere Prozeduren oder Programme müssen eine vollständige Pfadangabe verwenden, auch wenn sie sich im selben Verzeichnis wie die HACMP-Ereignisprozeduren befinden.

"Erwarteter-status" ist der Rückkehrcode des angegebenen Befehls bzw. Programms. Dabei kann entweder ein Ganzzahlwert oder ein "x" angegeben werden. Wenn "x" verwendet wird, ignoriert Cluster Manager den Rückkehrcode. Alle anderen Codes müssen mit dem erwarteten Rückkehrcode übereinstimmen, ansonsten erkennt Cluster Manager einen Ereignisfehler. Die Behandlung dieses Ereignisses blockiert den Prozeß, bis eine Behebung (durch einen manuellen Eingriff) erfolgt. Ohne manuellen Eingriff wird der

Knoten nicht mit den anderen Knoten synchronisiert. Die Synchronisierung über alle Knoten hinweg ist eine Voraussetzung für Cluster Manager, um alle Knoten steuern zu können.

"NULL" ist ein für zukünftige Zwecke reserviertes Feld. Das Wort "NULL" muß am Ende jeder Zeile außer der barrier-Zeile stehen. Wenn mehrere Wiederherstellungsbefehle zwischen zwei barrier-Befehlen (oder vor dem ersten) angegeben werden, werden die Wiederherstellungsbefehle auf dem Knoten selbst und zwischen den Knoten parallel ausgeführt.

Der Befehl `barrier` dient zur Synchronisierung aller Befehle über alle Clusterknoten hinweg. Wenn ein Knoten auf die `barrier`-Anweisung im Wiederherstellungsprogramm trifft, initiiert Cluster Manager das `barrier`-Protokoll auf diesem Knoten. Da das `barrier`-Protokoll ein zweiphasiges Protokoll ist, werden alle Knoten benachrichtigt, daß beide Phasen abgeschlossen wurden, wenn alle Knoten die `barrier`-Anweisung im Wiederherstellungsprogramm erreicht und "gewählt" haben, das Protokoll zu bestätigen.

Der Prozeß kann folgendermaßen zusammengefaßt werden:

1. Entweder Group Services/ES (für vordefinierte Ereignisse) oder die Ereignisverwaltung (für benutzerdefinierte Ereignisse) informiert HACMP ES Cluster Manager über das Ereignis.
2. Cluster Manager liest die Datei `rules.hacmprd` und bestimmt das Wiederherstellungsprogramm, das dem Ereignis zugeordnet ist.
3. Cluster Manager führt das Wiederherstellungsprogramm aus, das aus einer Folge von Wiederherstellungsbefehlen besteht.
4. Das Wiederherstellungsprogramm führt die Wiederherstellungsbefehle aus, die Shell-Prozeduren oder Binärbefehle sein können. (In HACMP für AIX stimmen die Wiederherstellungsbefehle mit den HACMP-Ereignisprozeduren überein.)
5. Cluster Manager empfängt den Rückkehrstatus von den Wiederherstellungsbefehlen. Ein unerwarteter Status führt zum Blockieren des Clusters, bis ein manueller Eingriff (mit Hilfe von `sm` mit `cm_rec_aids` oder mit Hilfe des Befehls `/usr/sbin/cluster/utilities/clruncmd`) durchgeführt wird.

HACMP ES-Prozedurdateien

Die folgenden Beispielprozeduren zur Fehlerbehebung durch Funktionsübernahme und für benutzerdefinierte Ereignisse gehören zum Lieferumfang von DB2 UDB EEE. Die Prozedurdateien befinden sich im Verzeichnis `$INSTNAME/sqllib/samples/hacmp/es`. Diese Prozeduren sind in der vorliegenden Form funktionsfähig. Die Wiederherstellungsaktion kann aber auch angepaßt werden.

- Wiederherstellungsprozedur für DB2-Datenbankpartition `rc.db2pe`. Dies ist die Prozedurdatei, die zum Starten und Stoppen der HACMP-Konfiguration

in einer Datenbankpartition verwendet wird. Die Prozedur funktioniert auch als HACMP-Start-und-Stopp-Prozedur für einen NFS-Server des DB2-Exemplareigners.

- DB2-spezifische benutzerdefinierte Ereignisse für HACMP ES. Sechs Standardereignisse sind bereitgestellt: eines für Prozeßwiederherstellung, zwei für Seitenwechselbereich und drei für NFS- und Automount-Wiederherstellung (automatisches Anhängen von Verzeichnissen).
- Funktionsübernahme von einem NFS-Datei-Server des DB2-Exemplars. Diese Prozedur vollzieht die Funktionsübernahme und Wiederherstellung des Dateisystem-Servers für ein DB2-Exemplar durch einen Ausweich-Server.
- Netzwerkfunktionsübernahme. Die Prozeduren `network_up_complete`, `network_back`, `network_down_complete` und `network_down` ermöglichen SP-DB2-Datenbankpartitionen im Fall eines Ausfalls ihres SP-Switch-Adapters eine Funktionsübernahme.
- Prozeduren zur Definition von Überwachungsereignissen für SP GUI Perspectives. Die Überwachung der Funktionsübernahme und der benutzerdefinierten Wiederherstellung wird durch Event Perspectives und Hardware Perspectives ermöglicht. Weitere Informationen zu Perspectives finden Sie in der Dokumentation für PSSP-Verwaltung (PSSP Administration).
- Prozeduren zum Installieren und Entfernen von Kernprozeduren (Core Scripts) und Ereignissen auf den HACMP ES-Knoten
- Prozedurdateien zum Erstellen und Entfernen der Fehlerverwaltungsressourcen von SP Perspectives (pman) zur Überwachung der HACMP- und DB2-Konfiguration

Die Wiederherstellungsprozeduren müssen auf jedem Knoten installiert werden, der Wiederherstellungsoperationen ausführen soll. Die Prozedurdateien können zentral von der SP-Steuer-Workstation bzw. einem anderen zuvor festgelegten SP-Knoten aus installiert werden:

1. Kopieren Sie die Prozeduren aus dem Verzeichnis `$(INSTNAME)/sql1lib/samples/hacmp/es` entweder auf die SP-Steuer-Workstation oder auf einen anderen SP-Knoten, auf dem die Befehle **pcp** und **pexec** ausgeführt werden können. Diese Befehle sind für die Installationsoperation erforderlich.
2. Passen Sie die Dateien `reg.parms.SAMPLE` und `failover.parms.SAMPLE` an Ihre Umgebung an, indem Sie Werte für Schlüsselparameter (wie `BUFFPAGE`) für Übernahmekonfigurationen einstellen. Typischerweise gilt für Konfigurationen für gegenseitige Übernahme, daß die Einstellungen für den Störfall auf Werte unterhalb der Hälfte der normalen Einstellungen gesetzt werden. Außerdem müssen Sie eine in einen von Ihnen gewählten Namen (anstelle von "SAMPLE") umbenannte Kopie dieser Dateien verwenden.

3. Passen Sie (nach Bedarf) die fünf Parameter `NFS_RETRIES`, `START_RETRIES`, `MOUNT_NFS`, `STOP_RETRIES` und `FAILOVER` in der Datei `rc.db2pe` an. Die `RETRY`- und `FAILOVER`-Einstellungen sollten für die meisten Implementierungen angemessen sein. Die Einstellung `MOUNT_NFS` sollte abhängig davon konfiguriert werden, ob Sie das Paket für NFS-Server-Verfügbarkeit verwenden. Sie sollten diese Einstellung angeben, wenn Sie wollen, daß `rc.db2pe` das NFS-Benutzerverzeichnis des DB2-Exemplareigners für Sie anhängt (Mount) und überprüft. Die Einstellung des Parameters `FAILOVER` auf den Wert "YES" ruft `db2_proc_restart` auf und startet einen Versuch, eine DB2-Datenbankpartition erneut zu starten. Schlägt die `RESTART`-Operation fehl, schließt `HACMP` mit einer Funktionsübernahme.
4. Passen Sie `db2_paging_action`, `db2_proc_recovery` und `nfs_auto_recovery` in der Ereignisdatei an. Editieren Sie `pwq`, um diese Einstellung in den DB2-Exemplareigner zu ändern. Passen Sie `db2_paging_action` an, um die Aktion anzugeben, die auszuführen ist, wenn der Seitenwechsellbereich zu über 90 % gefüllt wird. (Wenn dies geschieht, wird die DB2-Datenbankpartition gestoppt.) Ändern Sie die Prozedur, wenn weitere Wiederherstellungsaktionen erforderlich sind.
5. Verwenden Sie `db2_inst_ha` zur Installation der Prozeduren und Ereignisse auf den Knoten, die Sie angeben. (`HACMP ES` muß auf den Knoten vorinstalliert sein, bevor Sie beginnen.) Die Syntax von `db2_inst_ha` lautet:

```
db2_inst_ha $INSTNAME/sqlllib/samples/hacmp/es <knotenliste> <DATENBANKNAME>
```

Dabei gilt folgendes:

```
$INSTNAME/sqlllib/samples/hacmp/es
    ist Verzeichnis mit den Prozeduren und den Ereignissen
<knotenliste> ist die Knotenangabe im pcp- oder pexec-Format;
    z. B. 1-16 oder 1,2,3,4
<DATENBANKNAME> ist der Name der Datenbank für die Dateien mit den
regulären Parametern und den Funktionsübernahmeparametern.
```

Die Dateien `reg.parms.SAMPLE` und `failover.parms.SAMPLE` werden auf jeden Knoten kopiert und in `reg.parms.DATENBANKNAME` umbenannt.

`db2_inst_ha` kopiert Dateien auf jeden Knoten in das Verzeichnis `/usr/bin` und aktualisiert die `HACMP`-Ereignisdateien:

```
/usr/sbin/cluster/events/rules.hacmprd
/usr/sbin/cluster/events/network_up_complete
/usr/sbin/cluster/events/network_down_complete
```

6. Konfigurieren Sie Ihr System und die Prozeduren mit `HACMP`.
7. Verwenden Sie den Befehl `create_db2_events`, um die Überwachungsereignisse für Fehlerverwaltungsressourcen (`pman`) und `SP GUI Perspectives` zu installieren. Es sind noch weitere Konfigurations- und Anpassungsschritte in `Perspectives` erforderlich. Weitere Informationen zu `Perspectives` finden Sie im Handbuch *PSSP Administration Guide*.

8. Verwenden Sie den Befehl `ha_db2stop`, um die Datenbankpartitionen herunterzufahren, ohne daß eine HACMP ES-Fehlerbehebung durch Funktionsübernahme stattfindet. Kopieren Sie zur Verwendung dieses Befehls die Datei in das Benutzerverzeichnis des Datenbankbenutzers, und stellen Sie sicher, daß Berechtigungen und Eigentumsrecht für den betreffenden Benutzer definiert sind. Geben Sie dann als der betreffende Benutzer zum Stoppen der Datenbank ohne Fehlerbehebung durch Funktionsübernahme folgenden Befehl ein:

```
ha_db2stop
```

Anmerkung: Sie müssen warten, bis der Befehl beendet ist. Durch Drücken der Tasten `Strg-C` oder durch Abbrechen des Prozesses kann die Fehlerbehebung durch Funktionsübernahme vorzeitig wieder aktiviert werden, und einige Datenbankpartitionen werden vielleicht nicht gestoppt.

Operationen der DB2-Wiederherstellungsprozeduren mit HACMP ES

HACMP ES ruft die DB2-Wiederherstellungsprozeduren auf folgende Weise auf:

- `node_up_local` (beim Start eines Knotens)
HACMP fordert mit der `node_up`-Sequenz Datenträgergruppen (Volume Groups), logische Datenträger (Logical Volumes), Dateisysteme (Filesystems) und IP-Adressen an, die in den Ressourcengruppen angegeben sind, deren Eigner (durch Hintereinanderschalten (Cascading)) der Knoten ist bzw. die diesem Knoten (durch Rotation) zugeordnet sind.
Wenn `node_up_local_complete` ausgeführt wird, wird die Anwendungs-Server-Definition, die `rc.db2pe` enthält, initiiert, um die Datenbankpartition zu starten, die in den Anwendungs-Server-Definitionen auf diesem physischen Knoten angegeben ist.

Anmerkung: `rc.db2pe` paßt bei Ausführung im Startmodus die in der Datei `reg.parms.DATENBANK` angegebenen DB2-Parameter für jede DATENBANK im Datenbankverzeichnis an, die einer Parameterdatei (`parms`-Datei) entspricht.

Auf jedem Knoten spielt sich diese Abfolge von Aktionen beim Start ab. Wenn Sie mehrere HACMP-Cluster haben und diese parallel starten, werden mehrere Knoten gleichzeitig hochgefahren.

- `node_down_remote` (Funktionsübernahme)
HACMP fordert die Datenträgergruppen, logischen Datenträger, Dateisysteme und IP-Adressen an, die in der Ressourcengruppe auf dem designierten Übernahmeknoten angegeben sind.
Bei Ausführung von `node_down_remote_complete` führt HACMP `rc.db2pe` als Startprozedur für den Anwendungs-Server aus, die in der Ressourcengruppe für diese Datenbankpartition angegeben ist.

Anmerkung: Bei Ausführung im Modus der gegenseitigen Übernahme stoppt rc.db2pe die auf dem Knoten ausgeführte DB2-Datenbankpartition, paßt die DB2-Parameter in failover.parms.DATENBANK für jede DATENBANK im Datenbankverzeichnis an, die einer Parameterdatei (parms) entspricht, und startet anschließend beide Datenbankpartitionen auf dem physischen Übernahmeknoten.

- node_up_remote (Reintegration eines ausgefallenen Knotens - Ressourcen-
gruppe für hintereinandergeschaltete gegenseitige Übernahme)

Bei der Ausführung von node_up_remote auf dem alten Übernahmeknoten bewirkt die Anwendungs-Server-Definition, daß rc.db2pe im Stoppmodus ausgeführt wird.

Anmerkung: Wenn rc.db2pe in einem Reintegrationsmodus (bei gegenseitiger Übernahme) ausgeführt wird, stoppt rc.db2pe beide auf dem Knoten aktiven Datenbankpartitionen, paßt die DB2-Parameter an, die in reg.parms.DATENBANK für jede Datenbank im Datenbankverzeichnis angegeben sind, die einer Parameterdatei (parms) entspricht, und startet anschließend nur die Datenbankpartition, die auf diesem physischen Übernahmeknoten behalten werden soll.

Der alte Übernahmeknoten gibt die Datenträgergruppen, logischen Datenträger, Dateisysteme und IP-Adressen frei, die in Ressourcen-
gruppen, deren Eigener der wieder integrierte Knoten sein soll, angegeben sind.

HACMP fordert erneut Datenträgergruppen, logische Datenträger, Dateisysteme und IP-Adressen an, die in der Ressourcen-
gruppe angegeben sind, deren Eigener jetzt der sich wieder integrierende Knoten ist.

Wenn node_up_local_complete ausgeführt wird, wird die Anwendungs-
Server-Definition initiiert, die rc.db2pe enthält, um die in der Anwendungs-
Server-Definition auf diesem wieder zu integrierenden physischen Knoten
angegebene DB2-Datenbankpartition zu starten.

Anmerkung: rc.db2pe paßt bei Ausführung im Startmodus die in der Datei
reg.parms.DATENBANK angegebenen DB2-Parameter für jede
DATENBANK im Datenbankverzeichnis an, die einer Parameterdatei (parms-Datei) entspricht.

- node_down_local (Knotenstopp oder Stopp mit Übernahme)

Wenn node_down_local auf dem zu stoppenden Knoten ausgeführt wird,
bewirkt die Anwendungs-Server-Definition, daß rc.db2pe im Stoppmodus
ausgeführt wird.

Anmerkung: Bei Ausführung im Stoppmodus paßt rc.db2pe die DB2-Parameter an, die in failover.parms.DATENBANK für jede DATENBANK im Datenbankverzeichnis angegeben sind, die einer Parameterdatei (parms) entspricht, und stoppt anschließend die DB2-Datenbankpartition (dies geschieht für die Übernahme).

HACMP gibt die Datenträgergruppen, logischen Datenträger, Dateisysteme und IP-Adressen frei, die in Ressourcengruppen angegeben sind, deren Eigner jetzt der Knoten ist.

- db2_proc_recovery (DB2-Prozeßabbruch)
Alle Knoten führen die Prozedur db2_proc_restart aus. Der Knoten, auf dem der Fehler auftrat, startet die richtige DB2-Datenbankpartition.
- db2_paging_recovery (Wiederherstellung von Seitenwechselbereich)
Alle Knoten führen die Prozedur db2_paging_action aus. Wenn auf einem Knoten mehr als 70 % des Seitenwechselbereichs gefüllt sind, wird ein wall-Befehl abgesetzt. Wenn auf einem Knoten mehr als 90 % des Seitenwechselbereichs gefüllt sind, werden die DB2-Datenbankpartitionen auf diesem physischen Knoten gestoppt und dann erneut gestartet.
- nfs_auto_recovery (Fehler in NFS-Prozeß oder beim automatischen Anhängen)
Alle Knoten führen die Prozedur rc.db2pe im NFS-Modus aus. Wird die Ausführung eines NFS-Prozesses gestoppt, wird er erneut gestartet. In ähnlicher Weise wird der Prozeß für automatisches Anhängen bei einem Ausfall erneut gestartet.
- network_down_complete (Netzwerkfehler - SP-Switch)
Die Prozedur net_down wird aufgerufen. Diese Prozedur überprüft, ob das Netzwerk ein SP-Switch-Netzwerk und außer Funktion ist. Ist dies der Fall, wartet die Prozedur ein benutzerdefiniertes Zeitintervall ab. Das Standardzeitintervall sind 100 Sekunden.
Ist das SP-Switch-Netzwerk wieder verfügbar, wie durch das Ereignis network_up_complete mitgeteilt wird, wird keine Wiederherstellung durchgeführt.
Wenn das Zeitlimit erreicht ist, wird HACMP mit Funktionsübernahme gestoppt.

Anmerkung: Alle Ereignisse können über die SP-Fehlerverwaltung (Problem Management) und die GUI SP Perspectives überwacht werden.

Andere Prozedurdienstprogramme

Es stehen noch weitere Prozedurdienstprogramme (Skripte) für Sie bereit:

- ha_cmd ist ein Befehl, der zum Starten von HACMP auf SP-Knoten über die Steuer-Workstation dient. Die Syntax ist wie folgt:

ha_cmd <knotenbereich> <START|STOP|TAKE|FORCE>

Dabei gilt folgendes:

<knotenbereich> ist ein SP-Knotenbereich im pcp- oder pexec-Format.
Z. B.: "ha_cmd 3-6 START" startet HACMP auf den Knoten 3,4,5,6.
"ha_cmd 5 TAKE" beendet HACMP auf Knoten 5
für gegenseitige Übernahme.

- ha_mon ist ein Befehl zur Überwachung von hacmp_out-Dateien von HACMP von der SP-Steuer-Workstation aus. Die Syntax ist wie folgt:

ha_mon <knoten>

Dabei gilt folgendes:

<knoten> ist der zu überwachende SP-Knoten.

ha_mon wendet "tail -f" auf die Datei /tmp/hacmp.out auf dem angegebenen Knoten an.

- db2_turnoff_recov ist ein Befehl zur temporären Inaktivierung sämtlicher HACMP-Wiederherstellung (ohne Funktionsübernahme), der für äußerst seltene Fälle gedacht ist. Es werden keine Wiederherstellungen für DB2-Prozesse, Seitenwechsel, NFS-Prozesse oder automatisches Anhängen (Auto-mount) eingeleitet. Diese Funktion entfernt die Ereigniszeilengruppen für die betreffende Wiederherstellung aus der rules-Datei von HACMP. HACMP muß gestoppt und erneut gestartet werden. Die Syntax ist wie folgt:

db2_turnoff_recov <knotenliste>

- db2_turnon_recov ist ein Befehl zur Wiederaktivierung der HACMP-Wiederherstellung (ohne Funktionsübernahme). Dieser Befehl dient nach Ausführung des Befehls db2_turnoff_recov zur Wiederherstellung der rules-Datei von HACMP, so daß die Wiederherstellung nach benutzerdefinierten Ereignissen wieder erfolgen kann. HACMP muß gestoppt und erneut gestartet werden. Die Syntax ist wie folgt:

db2_turnon_recov <knotenliste>

Überwachen von HACMP-Clustern

Es wurden Prozeduren vorbereitet, mit denen Ereignisse für die SP-Fehlerverwaltung (pman) erstellt werden können, um neben den in HACMP ES bereits vorhandenen Überwachungsprogrammen die HACMP-SP-Konfiguration für DB2 zu überwachen. Zur Überwachung des HACMP-Status von der SP-Steuer-Workstation aus sind folgende Schritte erforderlich:

- Installieren Sie den HACMP-Client-Code auf der Steuer-Workstation.
- Editieren Sie die Datei /usr/sbin/cluster/etc/clhosts, und fügen Sie die Ethernet-SP-IP-Adressen der Knoten ein, die Sie überwachen wollen.

- Rufen Sie den Befehl `startsrc -s clinfo` auf, um die Überwachung der Cluster zu starten.

HACMP stellt eine Schnittstelle zur Überwachung der Cluster bereit (`/usr/sbin/cluster/clstat`).

Die Überwachung der Fehlerverwaltung mit der GUI von SP Perspectives für HACMP ES-Ereignisse und benutzerdefinierte Ereignisse wird folgendermaßen verwendet:

1. Rufen Sie den Befehl `create_db2_events <knotenliste>` auf, wobei *nodelist* Knoten im `pcp-` oder `pexec-`Format enthält. Diese Prozedur erstellt fünf `pman-`Ereignisse für die Überwachung durch Perspectives.

Anmerkung: Die Ressourcenvariablen `PSSP.pm.User_state12-16` werden zur Erstellung dieser Ereignisse verwendet. Wenn diese Ressourcenvariablen bereits für einen anderen Zweck verwendet werden, müssen `create_db2_events` und `update_db2_events` zur Verwendung anderer Variablen aktualisiert werden.

2. Starten Sie Perspectives auf der Steuer-Workstation. Wählen Sie über die Klickstartleiste die Perspective für Ereignisse (Events) aus. Es sollten fünf Ereignisse angezeigt werden: `db2_hacmp_recovery`, `db2_process_recovery`, `db2_paging_err`, `db2_nfs_err` und `Errlog_PERM_entry`.
3. Klicken Sie jedes Ereignis doppelt an. Registrieren Sie in der daraufhin erscheinenden Anzeige (innerhalb der Definitionstabelle) eine Bedingung für das Ereignis. Klicken Sie neben dem Abwärtspfeil bei Name: "unnamed", und wählen Sie denselben Namen aus wie für das Ereignis, das Sie als Bedingung angeben. Wählen Sie die Indexzunge Response Options aus. Klicken Sie den Druckknopf oben in der Anzeige an ("Send Message to Perspectives event session"). Sie können Befehle, Errlog-Einträge sowie SNMP-Traps für Vorkommen dieser Ereignisse angeben. Die Ereignisprotokollanzeigen bleiben nur über Sitzungen von Perspective hinweg erhalten. Daher kann es wünschenswert sein, für jede Sitzung AIX-Fehlerprotokolleinträge zu erstellen. Wählen Sie **OK** aus, und schließen Sie das Fenster.
4. Wählen Sie über die Klickstartleiste von Perspectives die Perspective für Hardware aus.
5. Wenn die GUI für den Hardware-Frame angezeigt wird, wählen Sie "View" und anschließend "Monitor" aus. Sie erhalten eine Liste von Ereignissen, die für Ihr SP überwacht werden können. Blättern Sie zum Ende der Liste. Dort sind folgende zusätzliche Ereignisse aufgeführt: eines für die HACMP DB2-Wiederherstellung (`db2_ha_ind`) und das andere für SP-Knoten-PERM-Fehler (`Errlog_PERM_mon`). Wählen Sie die Ereignisse aus, die Sie überwachen wollen. (Wenn ein Ereignis eintritt, zeigt der Knoten ein rotes "X" an. Wenn alle überwachten Bedingungen in Ordnung sind, ist

die Knotenanzeige grün.) In der Regel werden `host_responds`, `switch_responds` und `node_power_LED` verwendet. Darüber hinaus können Sie die DB2-HACMP-Wiederherstellung und die PERM-Fehler auf dem Knoten überwachen.

Anmerkung: Die Variablen `db2_hacmp_mon` und `db2_hacmp_recovery` für `pman` und `Perspectives` geben nicht den HACMP-Clusterstatus wieder. Diese Variablen zeigen statt dessen den Status der `rc.db2pe`-Operation zum Starten bzw. Stoppen von DB2 an. Der „echte“ HACMP-Status wird vom HACMP-Monitor `clstat` gezeigt, der den HACMP-Clusterstatus wiedergibt. Wenn Sie wollen, daß `db2_hacmp_ind` Überwachungswerte liefert, die dem HACMP-Status nahe kommen, fügen Sie die folgende Zeile in Ihre Datei `/etc/inittab` ein:

```
haind:2:wait:/usr/bin/db2_update_events HAIND OFF 2>&1 >/dev/null
```

Wenn Sie die Verwendung von NetView für Ihre Implementierung planen, sollten Sie die Verwendung von HAVIEW (gehört zu HACMP) zur Überwachung Ihrer Konfiguration in Betracht ziehen. Informationen zur Konfiguration dieses Produkts entnehmen Sie bitte der NetView-Dokumentation.

Installation von SP HACMP ES unter DB2

Im folgenden finden Sie eine Übersicht in Einzelschritten über die Installations- und Migrationsprozesse, die Ihnen bei der Planung für die Installation von HACMP ES mit DB2 Universal Database Hilfestellung leisten soll.

Neuinstallation von SP HACMP ES unter DB2

Gehen Sie wie folgt vor, um HACMP ES zu installieren:

1. Installieren Sie das Betriebssystem AIX auf jedem SP-Knoten (nach den Anweisungen der Handbücher zur SP-Installation und SP-Verwaltung). Sorgen Sie dafür, daß ein angemessener Seitenwechselbereich auf der Steuer-Workstation und auf jedem der SP-Knoten verfügbar ist. Stellen Sie sicher, daß die Switch-Konfiguration zusammen mit allen anderen änderbaren Konfigurationsparametern durchdacht und implementiert ist. Richten Sie die SP-Überwachung (`Perspectives`) ein, die Sie verwenden wollen. Stellen Sie sicher, daß die SP-Befehle `dsh`, `pcp` und `pexec` funktionieren.
2. Legen Sie das Datenbanklayout fest. Dies beinhaltet mindestens die Anzahl der zu verwendenden Knoten, die Zuordnung von DB2-Datenbankpartitionen zu physischen Knoten, die Plattenvoraussetzungen pro Knoten oder Partition und Überlegungen zu Tabellenbereichen. Außerdem sollten Sie überlegen, wer der DB2-Hauptexemplareigner sein soll und welche Zugriffsberechtigungen für diesen und andere Benutzer erforderlich sein sollen.

3. Planen Sie Ihre externe SSA-Plattenkonfiguration, einschließlich redundanter Adapter, gespiegelter Platten und Zwillingskonfiguration von Platten.
4. Füllen Sie anhand Ihres Datenbanklayouts und Ihrer SSA-Konfiguration die HACMP-Arbeitsblätter in den Handbüchern zur HACMP-Planung, Installation und Verwaltung aus.
5. Implementieren Sie Ihre externe SSA-Plattenkonfiguration. Stellen Sie sicher, daß die Mikrocodeversionen über alle Laufwerke hinweg konsistent sind, und verwenden Sie das Dienstprogramm Maymap zum Prüfen und Ausfüllen etwaiger Lücken in Ihren Arbeitsblättern.
6. Installieren Sie DB2 UDB EEE auf jedem SP-Knoten.
7. Installieren Sie HACMP ES auf jedem SP-Knoten.
8. Installieren Sie das Paket DB2 UDB EEE HACMP ES on SP Package mit Hilfe des Befehls **db2_inst_ha**.
9. Erstellen Sie den DB2-Hauptexemplarbenutzer, und stellen Sie sicher, daß er auf alle Knoten zugreifen kann. Zu diesem Zeitpunkt ist dies kein hochverfügbarer Benutzer. Es kann sich vorläufig um einen SP-Benutzer auf der SP-Steuer-Workstation handeln.
10. Erstellen Sie Ihr DB2-Exemplar und Ihre Datenbank. Stellen Sie sicher, daß es betriebsbereit ist, indem Sie den Befehl **db2start** und anschließend den Befehl **db2stop** aufrufen, bevor Sie mit dem nächsten Schritt fortfahren.
11. Wenn Sie die Datenbank vor dem Hinzufügen von HACMP mit Daten füllen wollen, tun Sie dies jetzt.
12. Konfigurieren Sie anhand der HACMP-Arbeitsblätter und der Informationen in diesem Dokument HACMP ES in der SP-Knotentopologie und den Ressourcengruppen.
13. Ändern Sie den gegenwärtigen Benutzer, angefangen bei Ihrem NFS-Server-Knoten für den DB2-Hauptexemplarbenutzer, indem Sie auf allen Knoten `/etc/security/user` und `/etc/passwd` nach den Angaben in diesem Dokument ändern. Dieser Benutzer wird zu einem hochverfügbaren NFS-Benutzer. Dieser Knoten und der zugehörige Ausweichknoten aktualisieren `/etc/exports`. Alle Knoten werden in die Lage versetzt, dieses Verzeichnis mit Hilfe von NFS (mit einem Eintrag in `/etc/filesystems` auf jedem Knoten) über die IP-Aliasadressen des Switch anzuhängen (mount).
14. Packen Sie das Ausgangsverzeichnis des Hauptexemplareigners mit "tar", und entpacken Sie es an der neuen Speicherposition.
15. Erstellen Sie ein NFS-Dateisystem auf jedem der SP-Knoten, um ein neues Exemplarbenutzerverzeichnis anzuhängen.

16. Starten Sie HACMP auf dem NFS-Server-Knoten. Überprüfen Sie anhand der Datei /tmp/hacmp.out, ob HACMP erfolgreich anläuft. Der Befehl **ha_mon** kann zur Überwachung der Schreibvorgänge für diese Datei verwendet werden.
17. Starten Sie nacheinander die anderen Knoten, und überprüfen Sie den erfolgreichen Anlauf durch Untersuchen der Datei /tmp/hacmp.out. Der Befehl **ha_mon** kann zur Überwachung der Schreibvorgänge für diese Datei verwendet werden.
18. Richten Sie die wahlfreie Überwachung durch Perspectives und Fehlerverwaltung (Problem Management) ein.
19. Überprüfen Sie die Übernahmefunktionalität auf jedem Knoten, indem Sie eine parallele Wartungsaktion auf jedem Knoten simulieren. Der Befehl **ha_cmd** (unter Angabe der Option TAKE) kann dazu verwendet werden, HACMP auf ordnungsgemäße Weise mit Funktionsübernahme zu stoppen. Überprüfen Sie anhand der Datei /tmp/hacmp.out und mit Hilfe Ihrer Überwachungsprogramme den Erfolg der Funktionsübernahmen und Reintegrationen.

Migration von SP HACMP ES für DB2

Wenn Sie eine Migration von einer Nicht-HACMP-Installation zu einer mit HACMP durchführen wollen, machen Sie sich mit der folgenden Übersicht vertraut:

1. Wandeln Sie die Konfiguration Ihrer externen Platten in eine hochverfügbare, gespiegelte Zwillingskonfiguration um. Fügen Sie zusätzliche Hardware und Platten hinzu, um diese Konfiguration herzustellen. Beachten Sie, daß Namen verschiedener logischer Datenträger auf verschiedenen Knoten innerhalb einer Zwillingskonfiguration eindeutig sein *müssen*. Dies gilt für Datenträgergruppen, logische Datenträger und Dateisysteme.
2. Schließen Sie die HACMP-Planung ab, und füllen Sie die entsprechenden Arbeitsblätter in diesem Dokument aus.
3. Implementieren Sie die Änderungen an Ihrer externen SSA-Plattenkonfiguration. Stellen Sie sicher, daß die Mikrocodeversionen über alle Laufwerke hinweg konsistent sind, und verwenden Sie das Dienstprogramm Maymap zum Prüfen und Ausfüllen etwaiger Lücken in Ihren Arbeitsblättern.

Anmerkung: SSA-Platten in einer RAID-5-Konfiguration werden unterstützt. Die einzig zulässige Konfiguration sind zwei SSA-Adapter in derselben RAID-Schleife. Für eine HACMP-Konfiguration mit einer Zwillingskonfiguration der RAID-Platten wird nur ein Adapter pro Knoten unterstützt. In dieser Konfiguration ist der Adapter ein einzelner Fehlerpunkt beim Zugriff auf die Platten. Weitere Konfigurationsmaßnahmen zur Erkennung von Adapterausfällen und zu

deren Umstufung in ein HACMP-Funktionsübernahmeereignis werden deshalb empfohlen. AIX-Fehleraufzeichnung ist die einfachste Möglichkeit, einen Knoten für die Funktionsübernahme zu konfigurieren, falls der SSA-Adapter ausfallen sollte. Weitere Informationen zur AIX-Fehleraufzeichnung finden Sie im Handbuch *HACMP for AIX, V4.2.2, Enhanced Scalability Installation and Administration Guide*.

4. Installieren Sie HACMP ES auf jedem SP-Knoten.
5. Installieren Sie das Paket DB2 UDB EEE HACMP ES on SP Package mit Hilfe des Befehls **db2_inst_ha**.
6. Konfigurieren Sie anhand der HACMP-Arbeitsblätter und der Informationen in diesem Dokument HACMP ES in der SP-Knotentopologie und den Ressourcengruppen.
7. Ändern Sie den gegenwärtigen Benutzer, angefangen bei Ihrem NFS-Server-Knoten für den DB2-Hauptexemplarbenutzer, indem Sie auf allen Knoten `/etc/security/user` und `/etc/passwd` nach den Angaben in diesem Dokument ändern. Dieser Benutzer wird zu einem hochverfügbaren NFS-Benutzer. Dieser Knoten und der zugehörige Ausweichknoten aktualisieren `/etc/exports`. Alle Knoten werden in die Lage versetzt, dieses Verzeichnis mit Hilfe von NFS (mit einem Eintrag in `/etc/filesystems` auf jedem Knoten) über die IP-Aliasadressen des Switch anzuhängen (mount).
8. Packen Sie das Ausgangsverzeichnis des Hauptexemplareigners mit "tar", und entpacken Sie es an der neuen Speicherposition.
9. Erstellen Sie ein NFS-Dateisystem auf jedem der SP-Knoten, um ein neues Exemplarbenutzerverzeichnis anzuhängen.
10. Starten Sie HACMP auf dem NFS-Server-Knoten. Überprüfen Sie anhand der Datei `/tmp/hacmp.out`, ob HACMP erfolgreich anläuft. Der Befehl **ha_mon** kann zur Überwachung der Schreibvorgänge für diese Datei verwendet werden.
11. Starten Sie nacheinander die anderen Knoten, und überprüfen Sie den erfolgreichen Anlauf durch Untersuchen der Datei `/tmp/hacmp.out`. Der Befehl **ha_mon** kann zur Überwachung der Schreibvorgänge für diese Datei verwendet werden.
12. Richten Sie die wahlfreie Überwachung durch Perspectives und Fehlerverwaltung (Problem Management) ein.
13. Überprüfen Sie die Übernahmefunktionalität auf jedem Knoten, indem Sie eine parallele Wartungsaktion auf jedem Knoten simulieren. Der Befehl **ha_cmd** (unter Angabe der Option TAKE) kann dazu verwendet werden, HACMP auf ordnungsgemäße Weise mit Funktionsübernahme

zu stoppen. Überprüfen Sie anhand der Datei /tmp/hacmp.out und mit Hilfe Ihrer Überwachungsprogramme den Erfolg der Funktionsübernahmen und Reintegrationen.

SP HACMP ES-Arbeitsblätter für DB2

Die folgenden Arbeitsblätter sind für die Verwendung mit den HACMP-Arbeitsblättern vorgesehen, die bei der Vorbereitung der externen SSA-Plattenkonfiguration ausgefüllt werden sollten (und die in den HACMP-Handbüchern zur Planung, Installation und Verwaltung zu finden sind). Es ist jeweils ein ausgefülltes Beispiel und ein leeres Arbeitsblatt vorgesehen.

Die Datenbankkonfiguration auf externen Platten, die auf dem ersten Musterarbeitsblatt dokumentiert ist, wird in der folgenden Abbildung gezeigt. Die Anweisung zur Erstellung der Datenbank lautet:

```
db2 create database pwq on /newdata
```

Sowohl die externen SSA-Adapter als auch die externen SSA-Platten werden gespiegelt und in einer Zwillingskonfiguration für logische Datenträger ohne einen einzelnen Fehlerpunkt konfiguriert. Das Diagramm zeichnet eine Konfiguration in ähnlicher Weise wie die Ausgabe des Befehls **maymap**. Maymap ist ein Dienstprogramm (über AIXTOOLS verfügbar), das die externe SSA-Plattenkonfiguration anzeigt und bei der Planung der Konfiguration eingesetzt werden sollte.

Beispielkonfiguration für externe Platten einer DB2-Datenbank mit vier Knoten

- Darstellung der Zwillingskonfiguration für hohe Verfügbarkeit

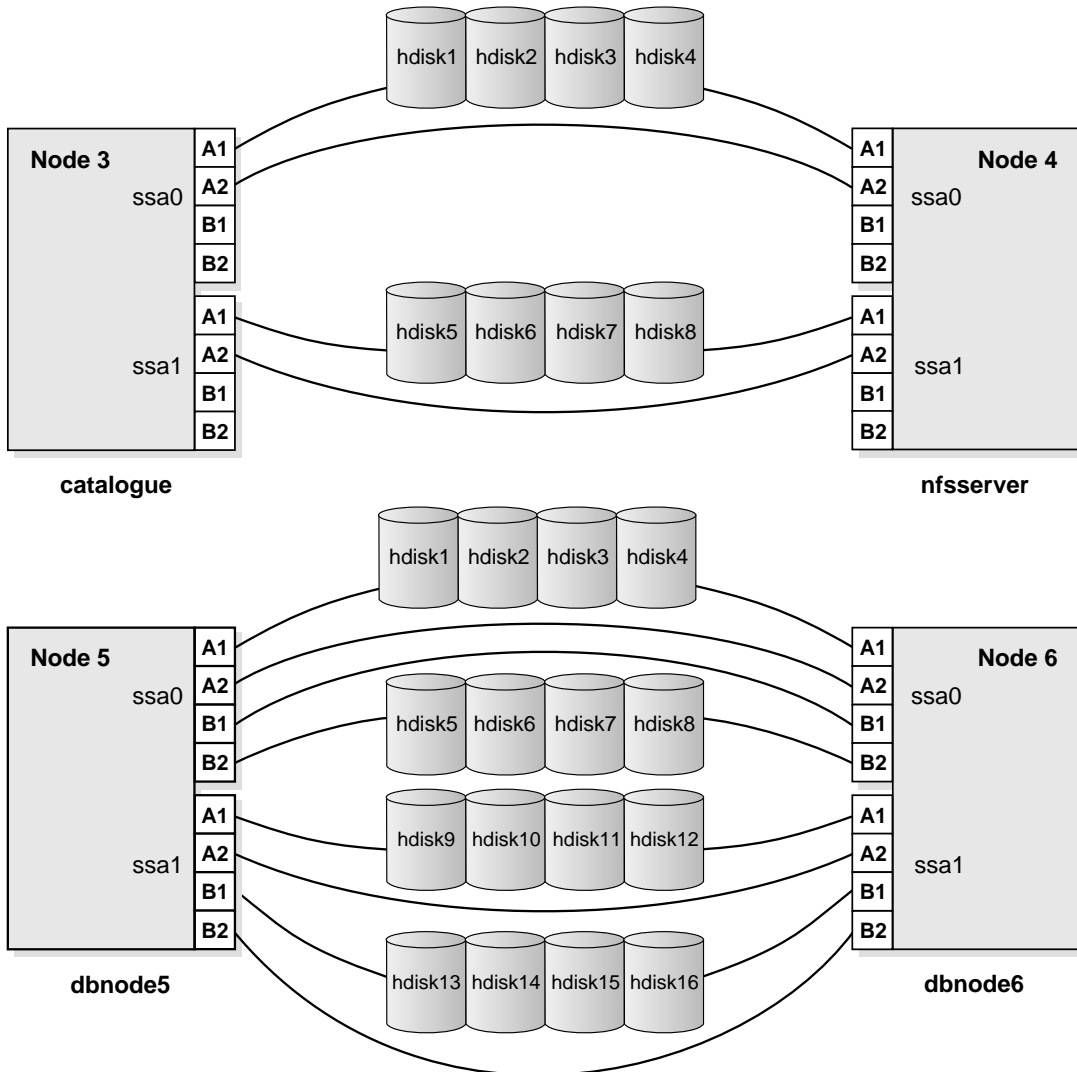


Abbildung 55. Beispielkonfiguration für externe Platten einer DB2-Datenbank mit 4 Knoten

Zum besseren Verständnis der folgenden Tabelle sollten Sie die HACMP-Dokumentation zu den Themen Quorum-Einstellungen (Quorum Settings) auf Datenträgergruppen und Konsistenz-Einstellungen für gespiegelte Schreibvorgänge auf logischen Datenträgern (Mirrored Write Consistency Settings On Logical Volumes) gut durchlesen. Die Einstellungen beider Berei-

che haben direkte Auswirkungen auf die Verfügbarkeit und Leistung Ihrer Konfiguration. Machen Sie sich unbedingt mit diesen Einstellungen und ihrer Bedeutung vertraut. Die typische Einstellung für "quorum" und "mirrored write consistency" ist "off".

Tabelle 25. Datenträgergruppen, logische Datenträger und Dateisysteme für HACMP

SP-Knoten	Name der Datenträgergruppe	PP-Größe (MB)	Name des logischen Datenträgers	# von Kopien	# von Platten (hplatten-Liste)	hdisk-Liste	Mount-Punkt für Dateisystem (MBsystemprotokoll)	Logischer Datenträger für Dateisystemprotokoll	Knotenbeschreibung und Ausweichknoten	Nutzer/Eigner der logischen /dev-Einheit
3	havg3	8	hlv300	10	2	hdisk1 hdisk5	/newdata /pwq /NODE0003	hlog301	Katalogknoten Knoten 4	newmount-Punkt;
3	havg3	8	hlog301	1	2	hdisk1 hdisk5	N/V	N/V	Katalogknoten jfslog; Knoten 4	root *
3	havg3	8	hlv301	10	2	hdisk2 hdisk6	N/V	N/V	Raw-Temp-Bereich für Katalogknoten; Knoten 4	pwq **
4	havg4	8	hlv400	10	2	hdisk3 hdisk7	/dbmnt	hlog401	nfserver pwq home; Knoten 3	root *
4	havg4	8	hlog401	1	2	hdisk3 hdisk7	N/V	N/V	nfserver jfslog; Knoten 3	root *
5	havg5	8	hlv500	10	2	hdisk1 hdisk9	/newdata/ pwq/ NODE0005	HLOG501	Dbnode5-Mount-Punkt; Knoten 6	root *
5	havg5	8	hlog501	1	2	hdisk1 hdisk9	N/V	N/V	Dbnode5 jfslog; Knoten 6	root *
5	havg5	8	hlv501	10	2	hdisk2 hdisk10	N/V	N/V	Raw-Temp-Bereich für Dbnode5; Knoten 6	pwq **
5	havg5	8	hlv502	100	2	hdisk2 hdisk10	N/V	N/V	Raw-Tabellenbereich für Dbnode5; Knoten 6	pwq **

Tabelle 25. Datenträgergruppen, logische Datenträger und Dateisysteme für HACMP (Forts.)

SP-Knoten	Name der PP-Datenträgergruppe	Größe (MB)	Name des logischen Datenträgers	# von PPs	Kopien (hplatten-Liste)	hdisk-Liste (hdisk-Liste)	Mount-Punkt für Dateisystem (MBsystemprotokoll)	Logischer Datenträger für Dateisystemprotokoll	Knotenbeschreibung und Ausweichknoten	Nutzer/Eigner der logischen /dev-Einheit
5	havg5	8	halv503	100	2	hdisk3 hdisk11	N/V	N/V	Raw-Tabellenbereich für Dbnode5; Knoten 6	pwq **
5	havg5	8	halv504	100	2	hdisk3 hdisk11	N/V	N/V	Raw-Tabellenbereich für Dbnode5; Knoten 6	pwq **
5	havg5	8	halv505	100	2	hdisk4 hdisk12	/dbdata5	hlog501	Dbnode6-Systemtabellenbereich; Knoten 6	root *
6	havg6	8	hlv600	10	2	hdisk5 hdisk13	/newdata/ pwq/ NODE0006	hlog601	Dbnode6-Mount-Punkt; Knoten 5	root *
6	havg6	8	hlog601	1	2	hdisk5 hdisk13	N/V	N/V	Dbnode6 jfslog; Knoten 5	root *
6	havg6	8	hlv601	10	2	hdisk6 hdisk14	N/V	N/V	Raw-Temp-Bereich für Dbnode6; Knoten 5	pwq **
6	havg6	8	hlv602	100	2	hdisk6 hdisk14	N/V	N/V	Raw-Tabellenbereich für Dbnode6; Knoten 5	pwq **
6	havg6	8	hlv603	100	2	hdisk7 hdisk15	N/V	N/V	Raw-Tabellenbereich für Dbnode6; Knoten 5	pwq **

Tabelle 25. Datenträgergruppen, logische Datenträger und Dateisysteme für HACMP (Forts.)

SP-Knoten	Name der PP-Datenträgergruppe	Größe (MB)	Name des logischen Datenträgers	# von Kopien (hplatten-Liste)	hdisk-Liste	Mount-Punkt für Dateisystem (MBsystemprotokoll)	Logischer Datenträger für Dateisystemprotokoll	Knotenbeschreibung und Ausweichknoten	Nutzer/Eigner der logischen /dev-Einheit	
6	havg6	8	hlv604	100	2	hdisk7 hdisk15	N/V	N/V	Raw-Tabellenbereich für Dbnod6; Knoten 5	pwq **
6	havg6	8	hlv605	100	2	hdisk8 hdisk16	/dbdata6	hlog601	System-tabellenbereich für Dbnod6; Knoten 5	root *

Anmerkungen:

- * Logische Datenträger und Protokolle für jfs-Dateisysteme behalten die root-Berechtigungen bei.
- ** Unformatierte (Raw) Datenbankbereiche erhalten die Datenbankbenutzerberechtigungen über die entsprechenden /dev-Dateieinträge (/dev/rxxxx).

Tabelle 26. Datenträgergruppen, logische Datenträger und Dateisysteme für HACMP (leer)

SP-Knoten	Name der PP-Datenträgergruppe	Größe (MB)	Name des logischen Datenträgers	# von Kopien (hplatten-Liste)	hdisk-Liste	Mount-Punkt für Dateisystem (MBsystemprotokoll)	Logischer Datenträger für Dateisystemprotokoll	Knotenbeschreibung und Ausweichknoten	Nutzer/Eigner der logischen /dev-Einheit

Tabelle 26. Datenträgergruppen, logische Datenträger und Dateisysteme für HACMP (leer) (Forts.)

SP-Knoten	Name der Daten-trägergruppe	PP-Größe (MB)	Name des logischen Daten-trägers	# von Kopien PPs	Kopien (hplatten-Liste)	hdisk-Liste (hplatten-Liste)	Mount-Punkt für Datei-system (MB)	Logischer Daten-träger für Datei-systemprotokoll	Knoten-beschreibung und Aus-weich-knoten	Benutzer/Eigner der logischen /dev-Einheit

Tabelle 26. Datenträgergruppen, logische Datenträger und Dateisysteme für HACMP (leer) (Forts.)

SP-Knoten	Name der Daten-trägergruppe	Größe (MB)	Name des logischen Daten-trägers	# von Kopien	hdisk-Liste (hplatten-Liste)	Mount-Punkt für Datei-system (MB)	Logischer Daten-träger für Datei-systemprotokoll	Knoten- und Aus- weich-knoten	benutzer/Eigner	logischer /dev-Einheit

Tabelle 27. Planung für HACMP-NFS-Server

SP-Knoten	Externes Dateisystem	Ausweichknoten	IP-Aliaspaare für SP-Switch-Boot- und Service-adressen (/etc/filesystems)	Anzuhängendes Dateisystem	Als Benutzerverzeichnis für Datenbank anzugebendes Dateisystem	Adressen zum Exportieren des Dateisystems (/etc/exports)
3	/dbmnt	4	nfs_boot_3 nfs_client_3	nfs_server:/dbmnt als /dbi	/dbi/pwq	nfs_boot_3 nfs_client_3 nfs_server_boot nfs_server nfs_boot_5 nfs_client_5 nfs_boot_6 nfs_client_6
4	/dbmnt	3	nfs_server_boot nfs_server	nfs_server:/dbmnt als /dbi	/dbi/pwq	nfs_boot_3 nfs_client_3 nfs_server_boot nfs_server nfs_boot_5 nfs_client_5 nfs_boot_6 nfs_client_6
5	N/V	N/V	nfs_boot_5 nfs_client_5	nfs_server:/dbmnt als /dbi	/dbi/pwq	N/V
6	N/V	N/V	nfs_boot_6 nfs_client_6	nfs_server:/dbmnt als /dbi	/dbi/pwq	N/V

Anmerkungen:

1. /etc/passwd muß auf allen Knoten identisch sein. Dies kann von der Steuer-Workstation aus synchronisiert werden.
2. Stellen Sie sicher, daß das externe Dateisystem die Berechtigung des Datenbankexemplareigners hat.
3. /etc/filesystems muß die Anhängparameter hard, bg, intr und rw haben.
4. /etc/exports hat

-root=ip1:ip2:ip3

nur auf dem Server und dem zugehörigen Ausweichknoten.

Tabelle 28. Planung für HACMP-NFS-Server (leer)

SP-Knoten	Externes Dateisystem	Ausweichknoten	IP-Aliaspaare für SP-Switch-Boot- und Service-adressen (/etc/filesystems)	Anzuhängendes Dateisystem	Als Benutzerverzeichnis für Datenbank anzugebendes Dateisystem	Adressen zum Exportieren des Dateisystems (/etc/exports)

Tabelle 28. Planung für HACMP-NFS-Server (leer) (Forts.)

SP-Knoten	Externes Dateisystem	Ausweichknoten	IP-Aliaspaare für SP-Switch-Boot- und Service-adressen	Anzuhängendes Dateisystem (/etc/filesystems)	Als Benutzerverzeichnis für Datenbank anzugebendes Dateisystem	Adressen zum Exportieren des Dateisystems (/etc/exports)

Kapitel 13. Hohe Verfügbarkeit in der Windows NT-Umgebung

Sie können Ihr Datenbanksystem so einrichten, daß bei Ausfall einer Maschine, der Datenbank-Server der ausgefallenen Maschine auf einer anderen Maschine ausgeführt werden kann. Unter Windows NT kann die Funktionsübernahme mit Microsoft Cluster Server (MSCS) implementiert werden. Zur Verwendung von MSCS wird Windows NT Version 4.0 Enterprise Edition mit installierter MSCS-Einrichtung benötigt.

MSCS leistet sowohl die Fehlererkennung und das Neustarten von Ressourcen in einer in einer Gruppe (Cluster) konfigurierten Umgebung als auch die Unterstützung für die Funktionsübernahme für physische Platten und IP-Adressen. (Wenn die ausgefallene Maschine wieder online ist, werden die Ressourcen nicht automatisch zurückübertragen, sofern sie dazu nicht vorher konfiguriert werden. Weitere Informationen finden Sie in „Überlegungen zur Funktionsrückübertragung“ auf Seite 305.)

Bevor Sie die Unterstützung der Funktionsübernahme von DB2-Exemplaren aktivieren können, müssen Sie die folgenden Planungsschritte ausführen:

1. Legen Sie fest, welche Platten für die Datenspeicherung verwendet werden sollen. Jedem Datenbank-Server sollte zumindest eine Platte zur eigenen Verfügung zugeordnet werden. Die Platte, die Sie zum Speichern von Daten einsetzen, muß an ein gemeinsames Plattensubsystem angeschlossen und als MSCS-Ressource konfiguriert werden.
2. Stellen Sie sicher, daß Sie eine IP-Adresse für jeden Datenbank-Server haben, der ferne Anforderungen unterstützen soll.

Wenn Sie die Unterstützung der Funktionsübernahme einrichten, kann dies für ein vorhandenes Exemplar geschehen, oder Sie können bei der Implementierung der Unterstützung der Funktionsübernahme ein neues Exemplar erstellen.

Zur Aktivierung der Unterstützung der Funktionsübernahme führen Sie die folgenden Schritte aus:

1. Erstellen Sie eine Eingabedatei für das Dienstprogramm DB2MSCS.
2. Rufen Sie den Befehl **db2mscs** auf.
3. Wenn Sie ein partitioniertes Datenbanksystem verwenden, registrieren Sie die Datenbanklaufwerkzuordnung zur Aktivierung der gegenseitigen Übernahme. Siehe „Registrieren der Datenbanklaufwerkzuordnung für

Konfigurationen zur gegenseitigen Übernahme in Umgebungen mit partitionierten Datenbanken” auf Seite 306.

Nach der Aktivierung der Unterstützung der Funktionsübernahme für das Exemplar sieht Ihre Konfiguration ungefähr wie in Abb. 56 aus.

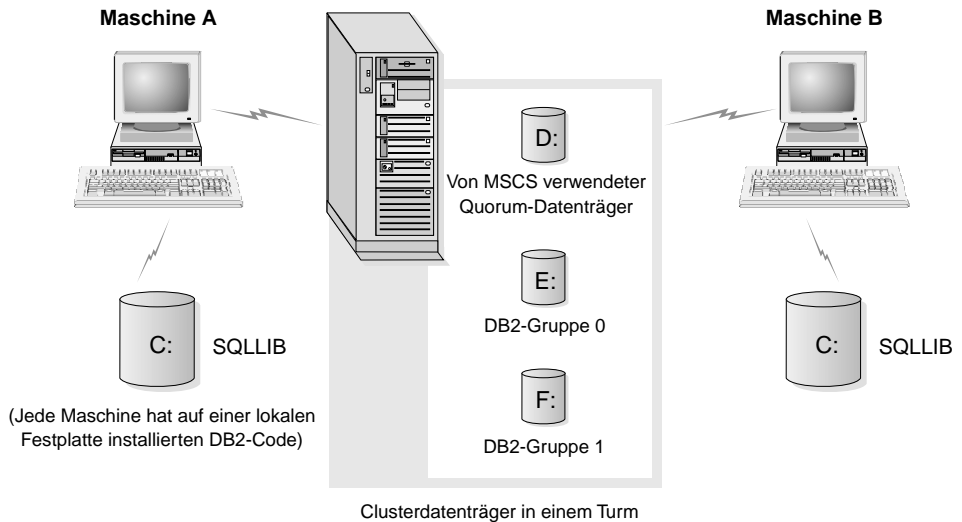


Abbildung 56. Beispiel für eine MSCS-Konfiguration

In den folgenden Abschnitten werden die verschiedenen Arten von Unterstützung für Funktionsübernahmen und ihre Implementierung beschrieben. Bevor Sie einen der im folgenden beschriebenen Schritte ausführen, müssen Sie die MSCS-Software bereits auf jeder Maschine installiert haben, die in einem MSCS-Cluster verwendet werden soll. Darüber hinaus muß auf jeder Maschine DB2 installiert sein.

Konfigurationen für Funktionsübernahme

Zwei Arten von Konfigurationen sind verfügbar:

- Bereitschaftsmodus (Hot Standby)
- Gegenseitige Übernahme

Zur Zeit unterstützt MSCS Cluster mit zwei Maschinen.

In einer partitionierten Datenbankanwendung müssen nicht alle Cluster die gleiche Konfigurationsart haben. Sie können Cluster besitzen, die im Bereitschaftsmodus (Hot Standby) konfiguriert sind, und andere, die für gegenseitige Übernahme konfiguriert sind. Wenn Ihr DB2-Exemplar zum Beispiel aus fünf Workstations besteht, können Sie zwei Maschinen zur gegenseitigen

Übernahme und zwei im Bereitschaftsmodus konfigurieren und eine Maschine aus der Funktionsübernahmekonfiguration ausnehmen.

Konfiguration für Bereitschaftsmodus

In einer Bereitschaftsmoduskonfiguration (Hot Standby) stellt eine Maschine im MSCS-Cluster eine dedizierte Übernahmeunterstützung bereit, und die andere Maschine ist Teil des Datenbanksystems. Wenn die zum Datenbanksystem gehörige Maschine ausfällt, wird ihr Datenbank-Server auf der Übernahmemaschine gestartet. Wenn Sie in einem partitionierten Datenbanksystem mehrere logische Knoten auf einer Maschine betreiben und die Maschine ausfällt, werden die logischen Knoten auf der Übernahmemaschine gestartet. Abb. 57 zeigt ein Beispiel einer Konfiguration im Bereitschaftsmodus.

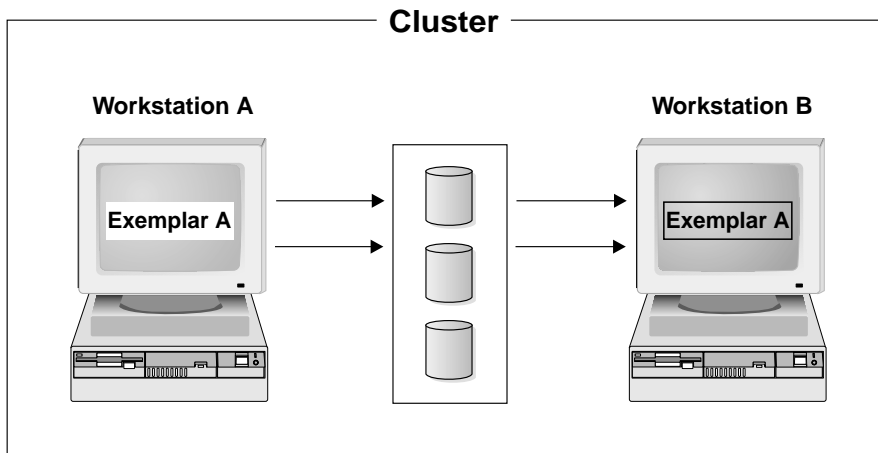


Abbildung 57. Konfiguration für Bereitschaftsmodus

Konfiguration für gegenseitige Übernahme

In einer Konfiguration zur gegenseitigen Übernahme sind beide Workstations an einem Datenbanksystem beteiligt (d. h., auf jeder Maschine wird mindestens ein Datenbank-Server ausgeführt). Wenn eine der Workstations im MSCS-Cluster ausfällt, wird der Datenbank-Server der ausgefallenen Maschine zur Ausführung auf der anderen Maschine gestartet. In einer Konfiguration zur gegenseitigen Übernahme kann ein Datenbank-Server auf einer Maschine unabhängig vom Datenbank-Server auf einer anderen Maschine ausfallen. Jeder Datenbank-Server kann zu einem beliebigen Zeitpunkt auf jeder Maschine aktiv sein. Abb. 58 auf Seite 296 zeigt ein Beispiel für die Konfiguration zur gegenseitigen Übernahme.

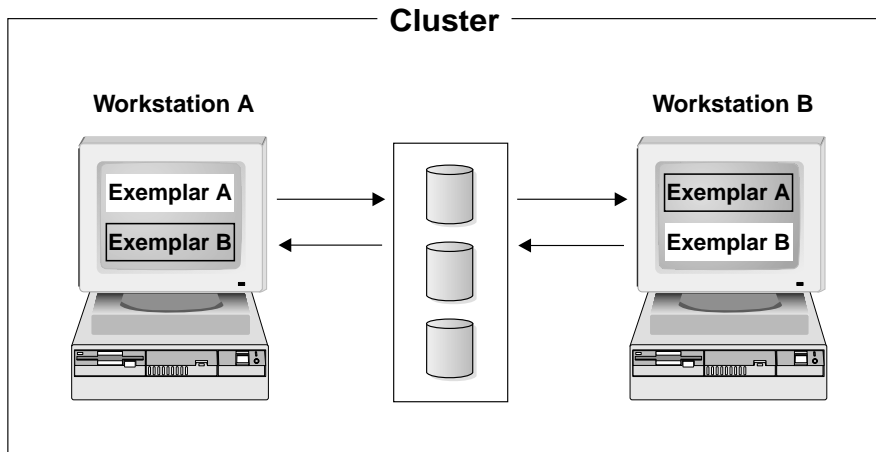


Abbildung 58. Konfiguration für gegenseitige Übernahme

Verwenden des Dienstprogramms DB2MSCS

Mit Hilfe des Dienstprogramms DB2MSCS können Sie die Infrastruktur für DB2 zur Unterstützung der Funktionsübernahme in der Windows NT-Umgebung durch die MSCS-Unterstützung erstellen. Sie können dieses Dienstprogramm zur Aktivierung der Funktionsübernahme in Umgebungen mit einer Partition und mit partitionierten Datenbanken verwenden.

Rufen Sie den Befehl **db2mcs** einmal für jedes Exemplar auf der Exemplar-eigenermaschine auf. Wenn es auf einer Maschine im MSCS-Cluster nur ein aktives DB2-Exemplar gibt, richtet dieses Programm eine Konfiguration im Bereitschaftsmodus ein. Wenn Sie auf jeder Maschine im MSCS-Cluster ein aktives Exemplar haben, führen Sie das Programm DB2MSCS einmal auf jeder Exemplareigenermaschine aus, um eine Konfiguration zur gegenseitigen Übernahme einzurichten.

Das Dienstprogramm DB2MSCS führt folgende Schritte aus:

1. Es liest die erforderlichen MSCS- und DB2-Parameter aus einer Eingabedatei namens DB2MSCS.CFG. Informationen über die gesamte Gruppe der Eingabeparameter finden Sie in „Angaben der Datei DB2MSCS.CFG“ auf Seite 297.
2. Es wertet die Parameter in der Eingabedatei aus.
3. Es registriert die DB2-Ressourcenart.
4. Es erstellt die MSCS-Gruppe (bzw. Gruppen) für die MSCS- und DB2-Ressourcen.
5. Es erstellt die IP-Ressource.

DB2_INSTANCE=DB2

Das Exemplar muß bereits vorhanden sein. Informationen zur Erstellung von Exemplaren finden Sie im Handbuch *DB2 Enterprise - Extended Edition für Windows Einstieg*.

DB2_LOGON_USERNAME

Der Name des Anmeldekontos für den DB2-Service (Dienst).

Dieser Parameter besitzt einen globalen Gültigkeitsbereich und wird in der Datei DB2MSCS.CFG nur einmal angegeben.

Dieser Parameter ist nur für Exemplare von DB2 Enterprise - Extended Edition erforderlich.

Beispiel:

```
DB2_LOGON_USERNAME=db2user
```

DB2_LOGON_PASSWORD

Das Kennwort des Anmeldekontos für den DB2-Service. Wenn der Parameter DB2_LOGON_USERNAME angegeben ist, aber der Parameter DB2_LOGON_PASSWORD nicht, fordert das Dienstprogramm DB2MSCS die Eingabe des Kennworts an. Das Kennwort wird bei der Eingabe in die Befehlszeile nicht angezeigt.

Dieser Parameter besitzt einen globalen Gültigkeitsbereich und wird in der Datei DB2MSCS.CFG nur einmal angegeben.

Dieser Parameter ist nur für Exemplare von DB2 Enterprise - Extended Edition erforderlich.

Beispiel:

```
DB2_LOGON_PASSWORD=xxxxxx
```

CLUSTER_NAME

Der Name des MSCS-Clusters. Alle Ressourcen, die nach dieser Zeile angegeben werden, werden in diesem Cluster erstellt, bis eine weitere Zeile mit dem Parameter CLUSTER_NAME angegeben wird.

Geben Sie diesen Parameter einmal pro Cluster an.

Dieser Parameter ist wahlfrei. Wenn er nicht angegeben wird, wird der Name des MSCS-Clusters auf der lokalen Maschine verwendet.

Beispiel:

```
CLUSTER_NAME=WOLFPACK
```

GROUP_NAME

Der Name der MSCS-Gruppe. Wenn dieser Parameter angegeben wird, wird eine neue MSCS-Gruppe erstellt, wenn keine vorhanden ist. Wenn die Gruppe bereits vorhanden ist, wird sie als Zielgruppe

verwendet. Jede MSCS-Ressource, die nach dieser Zeile erstellt wird, wird in dieser Gruppe erstellt, bis ein anderes Schlüsselwort GROUP_NAME angegeben wird.

Geben Sie diesen Parameter einmal für jede Gruppe an.

Dieser Parameter ist obligatorisch.

Beispiel:

```
GROUP_NAME=DB2 Gruppe
```

DB2_NODE

Die Knotennummer des Datenbankpartitions-Servers (Knotens), der in die aktuelle MSCS-Gruppe eingefügt werden soll. Wenn mehrere logische Knoten auf derselben Maschine vorhanden sind, benötigt jeder Knoten ein eigenes Schlüsselwort DB2_NODE.

Dieser Parameter wird nach dem Parameter GROUP_NAME angegeben, so daß die DB2-Ressourcen in der richtigen MSCS-Gruppe erstellt werden.

Dieser Parameter ist nur für Exemplare von DB2 Enterprise - Extended Edition erforderlich.

Beispiel:

```
DB2_NODE=0
```

IP_NAME

Der Name der IP-Adressenressource. Der Wert für IP_NAME kann willkürlich gewählt werden, muß jedoch eindeutig sein. Wenn dieser Parameter angegeben wird, wird eine MSCS-Ressource des Typs IP-Adresse (IP Address) erstellt.

Dieser Parameter wird für ferne TCP/IP-Verbindungen benötigt. In Umgebungen mit partitionierten Datenbanken muß dieser Parameter für die Exemplareignermaschine angegeben werden. Dieser Parameter ist in Datenbankumgebungen mit einer Partition wahlfrei.

Beispiel:

```
IP_NAME=IP-Adresse für DB2
```

Anmerkung: DB2-Clients sollten die TCP/IP-Adresse dieser IP-Ressource zum Katalogisieren des TCP/IP-Knoteneintrags verwenden. Durch Verwenden der MSCS-IP-Adresse können DB2-Clients auch, nachdem der Datenbank-Server von der anderen Maschine übernommen wurde, noch eine Verbindung zum Datenbank-Server herstellen, weil die IP-Adresse auf der Übernahmemaschine verfügbar ist.

Die Attribute der IP-Ressource sind folgende:

IP_ADDRESS

Die TCP/IP-Adresse der IP-Ressource. Geben Sie dieses Schlüsselwort an, um die TCP/IP-Adresse der vorangehenden IP-Ressource zu definieren.

Dieser Parameter ist erforderlich, wenn der Parameter IP_NAME angegeben wird.

Beispiel:

```
IP_ADDRESS=9.21.22.34
```

IP_SUBNET

Die Teilnetzmaske für die vorangehende IP-Ressource.

Dieser Parameter ist erforderlich, wenn der Parameter IP_NAME angegeben wird.

Beispiel:

```
IP_SUBNET=255.255.255.0
```

IP_NETWORK

Der Name des MSCS-Netzwerks, zu dem die vorangehende IP-Ressource gehört. Wenn dieser Parameter nicht angegeben wird, wird das erste vom System erkannte MSCS-Netzwerk verwendet.

Dieser Parameter ist wahlfrei.

Beispiel:

```
IP_NETWORK=Token Ring
```

NETNAME_NAME

Der Name der Netzwerknamensressource (Network Name). Geben Sie diesen Parameter an, um die Netzwerknamensressource zu erstellen.

Dieser Parameter ist in Datenbankumgebungen mit einer Partition wahlfrei. Für Umgebungen mit partitionierten Datenbanken ist er erforderlich.

Beispiel:

```
NETNAME_NAME=Netzwerkname für DB2
```

Die Attribute der Netzwerknamensressource sind folgende:

NETNAME_VALUE

Der Wert für den Netzwerknamen (Network Name).

Dieser Parameter ist erforderlich, wenn der Parameter NETNAME_NAME angegeben wird.

Beispiel:

NETNAME_VALUE=DB2SRV

NETNAME_DEPENDENCY

Die Abhängigkeitsliste für die Netzwerknamensressource. Jede Netzwerknamensressource muß eine Abhängigkeit von einer IP-Adressenressource haben. Wenn dieser Parameter nicht angegeben wird, besitzt die Netzwerknamensressource eine Abhängigkeit von der ersten IP-Ressource in der Gruppe.

Dieser Parameter ist wahlfrei.

Beispiel:

NETNAME_DEPENDENCY=IP-Adresse für DB2

DISK_NAME

Der Name der physischen Plattenressourcen, die in die aktuellen Gruppen versetzt werden sollen. Geben Sie so viele Plattenressourcen wie nötig an.

Anmerkungen:

1. Die Plattenressourcen müssen bereits vorhanden sein.
2. Wenn das Dienstprogramm DB2MSCS das DB2-Exemplar für MSCS-Unterstützung konfiguriert, wird das Exemplarverzeichnis auf die *erste* MSCS-Platte in der Gruppe kopiert. Soll eine andere MSCS-Platte für das Exemplarverzeichnis angegeben werden, ist der Parameter INSTPROF_DISK zu verwenden.

Beispiel:

DISK_NAME=Disk E:
DISK_NAME=Disk F:

INSTPROF_DISK

Ein wahlfreier Parameter zur Angabe einer MSCS-Platte, die das DB2-Exemplarverzeichnis aufnehmen soll. Wird dieser Parameter *nicht* angegeben, verwendet das Dienstprogramm DB2MSCS die *erste* MSCS-Platte, die zur gleichen Gruppe wie das Exemplarverzeichnis gehört.

Das DB2-Exemplarverzeichnis wird auf der MSCS-Platte unter dem Verzeichnis X:\DB2PROFS erstellt. (Dabei ist X der Laufwerksbuchstabe der MSCS-Platte.)

Beispiel:

INSTPROF_DISK=Disk E:

TARGET_DRVMAP_DISK

Ein wahlfreier Parameter zur Angabe der MSCS-Zielplatte für die Zuordnung von Datenbanklaufwerken. Wenn eine Datenbank auf einer MSCS-Platte erstellt wird, die nicht zur gleichen Gruppe wie der Knoten gehört, wird die Zielplatte der Laufwerkzuordnung zur Auf-

nahme der Datenbankpartition verwendet. Wenn dieser Parameter nicht angegeben wird, muß die Zuordnung der Datenbanklaufwerke manuell mit Hilfe des Dienstprogramms DB2DRVMP registriert werden.

Beispiel:

```
TARGET_DRVMAP_DISK = Disk E:
```

Einrichten der Funktionsübernahme für ein Datenbanksystem mit einer Partition

Wenn Sie das Dienstprogramm DB2MSCS für ein Datenbanksystem mit einer Partition ausführen, enthält eine MSCS-Gruppe DB2 und sämtliche abhängigen MSCS-Ressourcen (die IP-Adresse, Netzwerkname und Platten). Zum Beispiel kann der Inhalt der Datei DB2MSCS.CFG für ein Datenbanksystem mit einer Partition wie folgt aussehen:

```
#
# DB2MSCS.CFG für ein Datenbanksystem mit einer Partition
#
DB2_INSTANCE=DB2
CLUSTER_NAME=MSCS
GROUP_NAME=DB2 Gruppe
IP_NAME=...
IP_ADDRESS=...
IP_SUBNET=...
IP_NETWORK=...
NETNAME_NAME=...
NETNAME_VALUE=...
DISK_NAME=Disk E:
```

Einrichten einer Konfiguration zur gegenseitigen Übernahme für zwei Datenbanksysteme mit jeweils einer Partition

Sie können zwei Datenbanksysteme mit jeweils einer Partition, jede auf einer separaten Maschine einrichten, so daß ein Datenbanksystem bei einem Ausfall auf einer Maschine auf dem anderen MSCS-Knoten wieder gestartet wird.

Zur Einrichtung der Unterstützung der Funktionsübernahme für diese Konfiguration müssen Sie das Dienstprogramm DB2MSCS einmal auf jeder Exemplareignermaschine ausführen. Die Konfigurationsdatei muß für jedes Datenbanksystem angepaßt werden.

Nehmen Sie an, die DB2-Exemplare heißen DB2A und DB2B. Die Datei DB2MSCS.CFG für das Exemplar DB2A sähe folgendermaßen aus:

```
#
# DB2MSCS.CFG für das erste Datenbanksystem mit einer Partition
#
DB2_INSTANCE=DB2A
CLUSTER_NAME=MSCS
GROUP_NAME=DB2A Gruppe
IP_NAME=...
```



```

IP_ADDRESS=...
IP_SUBNET=...
IP_NETWORK=...
NETNAME_NAME=...
NETNAME_VALUE=...
DISK_NAME=Disk E:

```

Die Datei DB2MSCS.CFG für das Exemplar DB2B sähe folgendermaßen aus:

```

#
# DB2MSCS.CFG für das zweite Datenbanksystem mit einer Partition
#
DB2_INSTANCE=DB2B
CLUSTER_NAME=MSCS
GROUP_NAME=DB2B Gruppe
IP_NAME=...
IP_ADDRESS=...
IP_SUBNET=...
IP_NETWORK=...
NETNAME_NAME=...
NETNAME_VALUE=...
DISK_NAME=Disk F:

```

Ein vollständiges Beispiel finden Sie in „Beispiel - Einrichten zweier Exemplare mit einer Partition für gegenseitige Übernahme“ auf Seite 309.

Einrichten mehrerer MSCS-Cluster für ein partitioniertes Datenbanksystem

Wenn Sie das Dienstprogramm DB2MSCS für ein Datenbanksystem mit mehreren Partitionen ausführen, wird eine MSCS-Gruppe für jede physische Maschine erstellt, die Teil des Systems ist. Die Datei DB2MSCS.CFG muß mehrere Abschnitte enthalten, und jeder Abschnitt muß einen anderen Wert für den Parameter GROUP_NAME und andere Werte für alle erforderlichen abhängigen Ressourcen für diese Gruppe enthalten.

Außerdem müssen Sie den Parameter DB2_NODE für jeden Datenbankpartitions-Server in jeder MSCS-Gruppe angeben. Wenn Sie mehrere logische Knoten haben, ist für jeden logischen Knoten ein eigenes Schlüsselwort DB2_NODE erforderlich.

Nehmen Sie beispielsweise an, daß Sie ein Datenbanksystem mit mehreren Partitionen haben, das aus vier Datenbankpartitions-Servern auf vier Maschinen besteht, und daß Sie zwei MSCS-Cluster zur gegenseitigen Übernahme konfigurieren wollen. In diesem Fall würden Sie die Konfigurationsdatei DB2MSCS.CFG wie folgt einrichten:

```

#
# DB2MSCS.CFG für ein partitioniertes Datenbanksystem mit
# mehreren Clustern
DB2_INSTANCE=DB2MPP
DB2_LOGON_USERNAME=db2user
DB2_LOGON_PASSWORD=xxxxxx

```

```

CLUSTER_NAME=MSCS1
# Gruppe 1
GROUP_NAME=DB2 Gruppe 1
DB2_NODE=0
IP_NAME=...

...
# Gruppe 2
GROUP_NAME=DB2 Gruppe 2
DB2_NODE=1
IP_NAME=...

...

CLUSTER_NAME=MSCS2
# Gruppe 3
GROUP_NAME=DB2 Gruppe 3
DB2_NODE=2
IP_NAME=...

...
# Gruppe 4
GROUP_NAME=DB2 Gruppe 4
DB2_NODE=3
IP_NAME=...

...

```

Ein vollständiges Beispiel finden Sie in „Beispiel - Einrichten eines in vier Knoten partitionierten Datenbanksystems zur gegenseitigen Übernahme“ auf Seite 311.

Pflegen des MSCS-Systems

Bei der Ausführung des Dienstprogramms DB2MSCS wird die Infrastruktur für die Unterstützung der Funktionsübernahme für alle Maschinen im MSCS-Cluster erstellt. Zum Entfernen dieser Unterstützung von einer Maschine verwenden Sie den Befehl **db2iclus** mit der Option "drop". Soll die Unterstützung für eine Maschine wieder aktiviert werden, verwenden Sie die Option "add".

Der Befehl hat folgende Syntax:

```

➔ db2iclus add /i:—exemplarname /u:—kontoname,kennwort ➔
drop

```

```

/m:—maschinenname /c:—clustername ➔

```

Dabei gilt folgendes:

add	Aktiviert die Unterstützung der Funktionsübernahme auf der Maschine, indem sie einem MSCS-Cluster hinzugefügt wird. Die DB2-Ressource (Datenbank-Server) kann dann von dieser Maschine übernommen werden.
drop	Entfernt die Unterstützung der Funktionsübernahme von der Maschine, indem die Maschine aus einem MSCS-Cluster gelöscht wird.
/i: exemplarname	Der Name des Exemplars. (Dieser Parameter überschreibt die Einstellung der Umgebungsvariablen DB2INSTANCE.)
/u: kontoname, kennwort	Das Domänenkonto, das als Name des Anmeldekontos des DB2-Service (Dienstes) verwendet wird. Beispiel: <code>/u:domainA\db2nt,kennwort</code> Dieser Parameter ist nur im Zusammenhang mit der Option "add" erforderlich.
/m:machinename	Der Computername der Maschine, die einem MSCS-Cluster hinzugefügt oder aus ihm gelöscht werden soll. Sie müssen diese Option angeben, wenn Sie den Befehl von einer anderen Maschine aus ausführen als der, für die Sie die Funktionsübernahmeunterstützung ändern.
/c: clustername	Der Name des MSCS-Clusters, wie er im LAN registriert ist. Dieser Name wird bei der Ersterstellung des MSCS-Clusters angegeben.

Überlegungen zur Funktionsrückübertragung

Standardmäßig sind Gruppen so eingerichtet, daß sie nicht auf die ursprüngliche (ausgefallene) Maschine zurückübertragen werden. Sofern Sie eine DB2-Gruppe nicht manuell für eine Rückübertragung (Fallback) nach einer Funktionsübernahme konfigurieren, wird sie auch nach Beseitigung der Ursache für die Funktionsübernahme weiterhin auf dem alternativen MSCS-Knoten ausgeführt.

Wenn Sie eine DB2-Gruppe zur automatischen Rückübertragung auf die ursprüngliche Maschine konfigurieren, werden alle Ressourcen in der DB2-Gruppe, einschließlich der DB2-Ressource, sofort nach erneuter Verfügbarkeit

der ursprünglichen Maschine auf diese zurückübertragen. Wenn während der Rückübertragung eine Datenbankverbindung aktiv ist, kann die DB2-Ressource nicht offline genommen werden und der Rückübertragungsprozeß schlägt fehl.

Wenn Sie die Beendigung aller Datenbankverbindungen während der Rückübertragung erzwingen wollen, setzen Sie den Wert der Registrierungsvariablen DB2_FALLBACK auf ON. Diese Variable wird folgendermaßen definiert:

```
db2set DB2_FALLBACK=ON
```

Nach dem Einstellen dieser Registrierungsvariablen braucht das System nicht erneut gebootet und der Clusterdienst (Cluster-Service) nicht erneut gestartet zu werden.

Registrieren der Datenbanklaufwerkzuordnung für Konfigurationen zur gegenseitigen Übernahme in Umgebungen mit partitionierten Datenbanken

Bei der Erstellung einer Datenbank in einer Umgebung mit partitionierten Datenbanken können Sie einen Laufwerkbuchstabe angeben, um festzulegen, wo die Datenbank zu erstellen ist.

Anmerkung: Für Datenbankumgebungen mit einer Partition wird keine Datenbanklaufwerkzuordnung definiert.

Wenn der Befehl CREATE DATABASE ausgeführt wird, geht er davon aus, daß das Laufwerk, das Sie angeben, gleichzeitig für alle Maschinen verfügbar ist, die an dem Exemplar beteiligt sind. Da dies nicht möglich ist, verwendet DB2 eine Datenbanklaufwerkzuordnung, um demselben Laufwerk für jede Maschine einen anderen Namen zuzuweisen.

Nehmen Sie beispielsweise an, daß ein DB2-Exemplar namens DB2 zwei Datenbankpartitions-Server enthält:

```
NODE0 ist auf Maschine WOLF_NODE_0 aktiv  
NODE1 ist auf Maschine WOLF_NODE_1 aktiv
```

Nehmen Sie außerdem an, daß die freigegebene Platte E: (Share) zur selben Gruppe wie NODE0 und die freigegebene Platte F: zur selben Gruppe wie NODE1 gehören.

Der Befehl zur Erstellung einer Datenbank auf der freigegebenen Platte E:

```
db2 create database mppdb on e:
```

Damit der Befehl erfolgreich ausgeführt werden kann, muß Laufwerk E: auf beiden Maschinen verfügbar sein. In einer Konfiguration zur gegenseitigen Übernahme kann jeder Datenbankpartitions-Server auf einer anderen

Maschine aktiv und die Clusterplatte E: nur für eine Maschine verfügbar sein. In einem solchen Fall schlägt der Befehl CREATE DATABASE immer fehl.

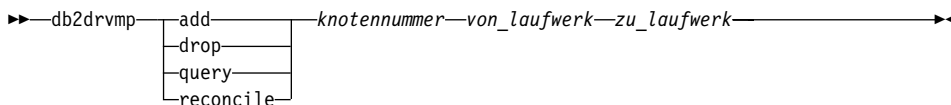
Zur Beseitigung dieses Problems sollte das Datenbanklaufwerk wie folgt zugeordnet werden:

Für NODE0 geht die Zuordnung von Laufwerk F: zu Laufwerk E:

Für NODE1 geht die Zuordnung von Laufwerk E: zu Laufwerk F:

Jeder Datenbankzugriff für NODE0 auf Laufwerk F: wird dann Laufwerk E: zugeordnet, und jeder Datenbankzugriff für NODE1 auf Laufwerk E: wird Laufwerk F: zugeordnet. Bei Verwendung der Laufwerkzuordnung erstellt der Befehl CREATE DATABASE Datenbankdateien auf Laufwerk E: für NODE0 und auf Laufwerk F: für NODE1.

Die Laufwerkzuordnung wird mit Hilfe des Befehls **db2drvmp** eingerichtet. Der Befehl hat folgende Syntax:



Der Befehl hat die folgenden Parameter:

- | | |
|---------------------|--|
| add | Fügt eine neue Datenbanklaufwerkzuordnung hinzu. |
| drop | Entfernt eine vorhandene Datenbanklaufwerkzuordnung. |
| query | Fragt eine Datenbankzuordnung ab. |
| reconcile | Repariert eine Datenbanklaufwerkzuordnung, wenn der Inhalt der Registrierdatenbank beschädigt ist. Weitere Informationen finden Sie in „Ausgleichen der Datenbanklaufwerkzuordnung“ auf Seite 308. |
| <i>knotennummer</i> | Die Knotennummer. Dieser Parameter ist für add- und drop-Operationen erforderlich. |
| <i>von_laufwerk</i> | Der Laufwerksbuchstabe, der zugeordnet wird. Dieser Parameter ist für add- und drop-Operationen erforderlich. |
| <i>zu_laufwerk</i> | Der Laufwerksbuchstabe, dem zugeordnet wird. Dieser Parameter ist für add-Operationen erforderlich. Für andere Operationen ist er nicht gültig. |

Zum Beispiel würde eine Datenbanklaufwerkzuordnung von F: zu E: für NODE0 durch folgenden Befehl definiert:

```
db2drvmp add 0 F E
```

Anmerkung: Die Datenbanklaufwerkzuordnung bezieht sich nicht auf Tabellenbereiche, Behälter oder andere Datenbankspeicherobjekte.

Mit dem folgenden, ganz ähnlichen Befehl würde die Datenbanklaufwerkzuordnung von Laufwerk E: zu Laufwerk F: für NODE1 definiert:

```
db2drvmp add 1 E F
```

Anmerkung: Alle Definitionen oder Änderungen der Datenbanklaufwerkzuordnung werden nicht sofort wirksam. Zur Aktivierung der Datenbanklaufwerkzuordnung muß die DB2-Ressource mit Hilfe des Clusterverwaltungsprogramms (Cluster Administrator) offline genommen und wieder online gebracht werden.

Mit Hilfe des Schlüsselworts TARGET_DRVMAP_DISK in der Datei DB2MSCS.CFG wird die Laufwerkzuordnung automatisch aktiviert.

Ausgleichen der Datenbanklaufwerkzuordnung

Wenn eine Datenbank auf einer Maschine erstellt wird, für die eine Datenbanklaufwerkzuordnung in Kraft ist, wird die Zuordnung auf dem Laufwerk in einer verdeckten Datei gesichert. Dadurch wird verhindert, daß das Datenbanklaufwerk nach der Erstellung der Datenbank entfernt wird. Ein Fall, in dem ein Ausgleichen (*reconcile*) der Datenbanklaufwerkzuordnung wünschenswert ist, liegt zum Beispiel vor, wenn die Datenbanklaufwerkzuordnung versehentlich gelöscht wurde. Zum Ausgleichen der Zuordnung muß der Befehl **db2drvmp reconcile** für jeden Datenbankpartitions-Server ausgeführt werden, der die Datenbank enthält. Der Befehl hat folgende Syntax:

```
►—db2drvmp reconcile—┐—————┐—————►  
                        └knotennummer—laufwerk┘
```

Der Befehl hat die folgenden Parameter:

- knotennummer* Die Knotennummer des zu reparierenden Knotens. Wenn *knotennummer* nicht angegeben wird, gleicht der Befehl die Zuordnung für alle Knoten aus.
- laufwerk* Das auszugleichende Laufwerk. Wenn kein Laufwerk angegeben wird, gleicht der Befehl die Zuordnung für alle Laufwerke aus.

Der Befehl **db2drvmp** durchsucht alle Laufwerke auf der Maschine nach Datenbankpartitionen, die von dem Datenbankpartitions-Server verwaltet werden, und wendet die Datenbanklaufwerkzuordnung erneut in der erforderlichen Weise auf die Registrierdatenbank an.

Beispiel - Einrichten zweier Exemplare mit einer Partition für gegenseitige Übernahme

Das Ziel in diesem Beispiel besteht in der Einrichtung zweier Datenbankexemplare mit jeweils einer Partition in einer Konfiguration zur gegenseitigen Übernahme. In diesem Beispiel werden vier Server in zwei MSCS-Cluster konfiguriert. Durch die Konfiguration zur gegenseitigen Übernahme wird bei Ausfall einer der Maschinen der für diese Maschine konfigurierte Datenbank-Server von der alternativen Maschine übernommen, wie es mit Hilfe der MSCS-Software konfiguriert wurde, und auf der alternativen Maschine ausgeführt.

In der entstehenden Konfiguration gibt es zwei MSCS-Cluster. Jeder Cluster verfügt über:

- Zwei Server, jeder mit 64 MB Speicher und einer lokalen SCSI-Platte von 2 GB
- Einen SCSI-Plattenturm, der drei gemeinsame, jeweils 2 GB große SCSI-Platten enthält

Darüber hinaus ist in jeder Maschine eine 100X-Ethernet-Adapterkarte installiert.

Auf jeder Maschine ist folgende Software installiert:

- Windows NT Version 4.0 Enterprise Edition mit installierter MSCS-Einrichtung
- DB2 Universal Database Enterprise Edition Version 7.

Es ergibt sich folgende Netzwerkkonfiguration:

Server 1: <ul style="list-style-type: none">• Machinenname: db2test1• TCP/IP-Host-Name: db2test1• IP-Adresse: 9.9.9.1 (Teilnetzmaske: 255.255.255.0)• MSCS-Clustername: ClusterA	Server 2: <ul style="list-style-type: none">• Machinenname: db2test2• TCP/IP-Host-Name: db2test2• IP-Adresse: 9.9.9.2 (Teilnetzmaske: 255.255.255.0)• MSCS-Clustername: ClusterA
---	---

Beide Maschinen im Netzwerk sind mit TCP/IP konfiguriert und über einen Ethernet 100 T-base Hub mit einem privaten LAN verbunden. Bei Abwesenheit eines DNS-Servers (Domain Name System) haben alle Maschinen eine lokale TCP/IP-Datei hosts, die folgende Einträge enthält:

```
9.9.9.1 db2test1 # für Server 1
9.9.9.2 db2test2 # für Server 2
9.9.9.3 ClusterA # für MSCS ClusterA
9.9.9.4 db2tcp1 # für ferne DB2-Client-Verbindung zu Server 1
9.9.9.5 db2tcp2 # für ferne DB2-Client-Verbindung zu Server 2
```

Vorbereitung

Bevor Sie die folgenden Schritte durchführen, wird angenommen, daß beide Maschinen zur gleichen Domäne namens DB2NTD gehören:

1. Erstellen Sie ein Domänenkonto für DB2, daß zur lokalen Administratorgruppe der Maschinen gehört, auf den DB2 aktiv sein wird. Verwenden Sie das Konto zur Durchführung aller Aufgaben:
 - Setzen Sie den Benutzernamen auf db2nt.
 - Setzen Sie das Kennwort auf db2nt.
2. Installieren Sie die MSCS-Einrichtung auf den Maschinen db2test1 und db2test2:
 - Nennen Sie den MSCS-Cluster ClusterA.
 - Die IP-Adresse des Clusters ist 9.9.9.3.
 - Die freigegebene Platte D: wird von der MSCS-Software verwendet.
 - Die freigegebenen Platten E: und F: werden von DB2 verwendet.
3. Installieren Sie DB2 Universal Database Enterprise Edition Version 7 auf Maschine db2test1. Installieren Sie die Software auf dem lokalen Laufwerk unter C:\SQLLIB.
4. Installieren Sie DB2 Universal Database Enterprise Edition Version 7 auf Maschine db2test2. Installieren Sie die Software auf dem lokalen Laufwerk unter C:\SQLLIB.

Im nächsten Schritt wird die Datei DB2MSCS.CFG für jedes Exemplar eingerichtet und das Dienstprogramm DB2MSCS für jedes Exemplar ausgeführt.

Ausführen des Dienstprogramms DB2MSCS

Zum Einrichten der Maschine db2test1 führen Sie folgende Schritte aus:

1. Melden Sie sich auf Maschine db2test1 als Benutzer db2nt an. Das Kennwort ist db2nt.
2. Erstellen Sie das DB2-Exemplar DB2A, wenn es nicht bereits vorhanden ist. Der Befehl zur Erstellung des Exemplars lautet:

```
db2icrt DB2A
```
3. Richten Sie die Datei DB2MSCS.CFG für das DB2-Exemplar auf Maschine db2test1 ein:

```
#
# DB2MSCS.CFG für Datenbanksystem
# auf Maschine db2test1
DB2_INSTANCE=DB2A
CLUSTER_NAME=ClusterA
#
# Gruppe 1
GROUP_NAME=DB2A Gruppe
IP_NAME=IP-Adresse für DB2A
IP_ADDRESS=9.9.9.4
IP_SUBNET=255.255.255.0
IP_NETWORK=ClusterA
```



```
NETNAME_NAME=Netzwerkname für DB2A
NETNAME_VALUE=DB2SRV1
NETNAME_DEPENDENCY=IP-Adresse für DB2A
DISK_NAME=Disk E:
INSTPROF_DISK=Disk E:
```

4. Führen Sie das Dienstprogramm DB2MSCS wie folgt aus:

```
db2mcs -f:DB2MSCS.CFG
```

5. Melden Sie sich vom Konto db2nt ab.
6. Melden Sie sich auf Maschine db2test2 als Benutzer db2nt an, der zur lokalen Administratorgruppe gehört. Das Kennwort ist db2nt.
7. Erstellen Sie das DB2-Exemplar DB2B, wenn es nicht bereits vorhanden ist. Der Befehl zur Erstellung des Exemplars lautet:

```
db2icrt DB2B
```

8. Richten Sie die Datei DB2MSCS.CFG für das DB2-Exemplar auf Maschine db2test2 ein:

```
#
# DB2MSCS.CFG für Datenbanksystem
# auf Maschine db2test2
DB2_INSTANCE=DB2B
CLUSTER_NAME=ClusterA
#
# Gruppe 1
GROUP_NAME=DB2B Gruppe
IP_NAME=IP-Adresse für DB2B
IP_ADDRESS=9.9.9.5
IP_SUBNET=255.255.255.0
IP_NETWORK=ClusterA
NETNAME_NAME=Netzwerkname für DB2B
NETNAME_VALUE=DB2SRV2
NETNAME_DEPENDENCY=IP-Adresse für DB2B
DISK_NAME=Disk F:
INSTPROF_DISK=Disk F:
```

9. Führen Sie das Dienstprogramm DB2MSCS wie folgt aus:

```
db2mcs -f:DB2MSCS.CFG
```

10. Melden Sie sich vom Konto db2nt ab.

Beispiel - Einrichten eines in vier Knoten partitionierten Datenbanksystems zur gegenseitigen Übernahme

Das Ziel in diesem Beispiel besteht in der Einrichtung eines in vier Knoten partitionierten Datenbanksystems in einer Konfiguration zur gegenseitigen Übernahme. In diesem Beispiel werden vier Server in zwei MSCS-Cluster konfiguriert. Durch die Konfiguration zur gegenseitigen Übernahme werden bei Ausfall einer der Maschinen die für diese Maschine konfigurierten Datenbankpartitions-Server von der alternativen Maschine übernommen, wie es mit Hilfe der MSCS-Software konfiguriert wurde, und als ein logischer Knoten auf der alternativen Maschine ausgeführt.

In der entstehenden Konfiguration gibt es zwei MSCS-Cluster. Jeder Cluster verfügt über:

- Zwei Server, jeder mit 64 MB Speicher und einer lokalen SCSI-Platte von 2 GB
- Einen SCSI-Plattenturm, der drei gemeinsame, jeweils 2 GB große SCSI-Platten enthält

Darüber hinaus ist in jeder Maschine eine 100X-Ethernet-Adapterkarte installiert.

Auf jeder Maschine ist folgende Software installiert:

- Windows NT Version 4.0 Enterprise Edition mit installierter MSCS-Einrichtung
- DB2 Universal Database Extended Enterprise Edition Version 7.

Es ergibt sich folgende Netzwerkkonfiguration:

Server 1: <ul style="list-style-type: none"> • Machinename: db2test1 • TCP/IP-Host-Name: db2test1 • IP-Adresse: 9.9.9.1 (Teilnetzmaske: 255.255.255.0) • MSCS-Clustername: ClusterA 	Server 2: <ul style="list-style-type: none"> • Machinename: db2test2 • TCP/IP-Host-Name: db2test2 • IP-Adresse: 9.9.9.2 (Teilnetzmaske: 255.255.255.0) • MSCS-Clustername: ClusterA
Server 3: <ul style="list-style-type: none"> • Machinename: db2test3 • TCP/IP-Host-Name: db2test3 • IP-Adresse: 9.9.9.3 (Teilnetzmaske: 255.255.255.0) • MSCS-Clustername: ClusterB 	Server 4: <ul style="list-style-type: none"> • Machinename: db2test4 • TCP/IP-Host-Name: db2test4 • IP-Adresse: 9.9.9.4 (Teilnetzmaske: 255.255.255.0) • MSCS-Clustername: ClusterB

Alle Maschinen im Netzwerk sind mit TCP/IP konfiguriert und über einen Ethernet 100 T-base Hub mit einem privaten LAN verbunden. Bei Abwesenheit eines DNS-Servers (Domain Name System) haben alle Maschinen eine lokale TCP/IP-Datei hosts, die folgende Einträge enthält:

```

9.9.9.1 db2test1 # für Server 1
9.9.9.2 db2test2 # für Server 2
9.9.9.3 db2test3 # für Server 3
9.9.9.4 db2test4 # für Server 4
9.9.9.5 ClusterA # für MSCS-Cluster 1
9.9.9.6 ClusterB # für MSCS-Cluster 2
9.9.9.7 db2tcp # für ferne DB2-Client-Verbindung

```

Vorbereitung

Bevor Sie die folgenden Schritte durchführen, wird angenommen, daß alle vier Maschinen zur gleichen Domäne namens DB2NTD gehören:

1. Erstellen Sie ein Domänenkonto für DB2, daß zur lokalen Administratorgruppe der Maschinen gehört, auf den DB2 aktiv sein wird. Verwenden Sie das Konto zur Durchführung aller Aufgaben:
 - Setzen Sie den Benutzernamen auf db2nt.
 - Setzen Sie das Kennwort auf db2nt.
2. Erstellen Sie ein zweites Domänenkonto mit dem Merkmal, daß das Kennwort nie abläuft. Dieses Konto wird den DB2-Services (Diensten) zugeordnet:
 - Setzen Sie den Benutzernamen auf db2mpp.
 - Setzen Sie das Kennwort auf db2mpp.
3. Installieren Sie die MSCS-Einrichtung auf den Maschinen db2test1 und db2test2:
 - Nennen Sie den MSCS-Cluster ClusterA.
 - Die IP-Adresse des Clusters ist 9.9.9.5.
 - Die freigegebene Platte D: wird von der MSCS-Software verwendet.
 - Die freigegebenen Platten E: und F: werden von DB2 verwendet.
4. Installieren Sie die MSCS-Einrichtung auf den Maschinen db2test3 und db2test4:
 - Nennen Sie den MSCS-Cluster ClusterB.
 - Die IP-Adresse des Clusters ist 9.9.9.6.
 - Die freigegebene Platte D: wird von der MSCS-Software verwendet.
 - Die freigegebenen Platten E: und F: werden von DB2 verwendet.
5. Installieren Sie DB2 Enterprise - Extended Edition auf Maschine db2test1:
 - Wählen Sie die Option *"Diese Maschine wird zum Datenbankpartitions-Server, der Eigner des Exemplars ist"* aus.
 - Das Konto für den DB2-Service ist db2mpp. Das Kennwort ist db2mpp.
 - Installieren Sie die Software auf dem lokalen Laufwerk unter C:\SQLLIB.
6. Installieren Sie DB2 Enterprise - Extended Edition auf den Maschinen db2test2, db2test3 und db2test4:
 - Wählen Sie die Option *"Diese Maschine wird ein neuer Knoten auf einem vorhandenen partitionierten Datenbanksystem"* aus.
 - Wählen Sie db2test1 als die Exemplareignermaschine aus.
 - Das Konto für den DB2-Service ist db2mpp. Das Kennwort ist db2mpp.
 - Installieren Sie die Software auf dem lokalen Laufwerk unter C:\SQLLIB.

Im nächsten Schritt wird die Datei DB2MSCS.CFG eingerichtet und das Dienstprogramm DB2MSCS ausgeführt.

Ausführen des Dienstprogramms DB2MSCS

Zum Einrichten der Maschine db2test1 führen Sie folgende Schritte aus:

1. Melden Sie sich als Benutzer db2nt an, der zur lokalen Administratorgruppe gehört. Das Kennwort ist db2nt.
2. Richten Sie die Datei DB2MSCS.CFG ein:

```
#
# DB2MSCS.CFG für ein partitioniertes Datenbanksystem mit
# mehreren MSCS-Clustern
DB2_INSTANCE=DB2MPP
CLUSTER_NAME=ClusterA
DB2_LOGON_USERNAME=db2mpp
DB2_LOGON_PASSWORD=db2mpp
# Gruppe 1
# für DB2 Knoten 0
GROUP_NAME=DB2NODE0
DB2_NODE=0
IP_NAME=IP-Adresse für DB2
IP_ADDRESS=9.9.9.7
IP_SUBNET=255.255.255.0
IP_NETWORK=Ethernet
NETNAME_NAME=Netzwerkname für DB2
NETNAME_VALUE=DB2WOLF
NETNAME_DEPENDENCY=IP-Adresse für DB2
DISK_NAME=Disk E:
INSTPROF_DISK=Disk E:
#

# Gruppe 2
# für DB2 Knoten 1
GROUP_NAME=DB2NODE1
DB2_NODE=1
DISK_NAME=Disk F:
#

CLUSTER_NAME=ClusterB
# Gruppe 3
# für DB2 Knoten 2
GROUP_NAME=DB2NODE2
DB2_NODE=2
DISK_NAME=Disk E:
#

# Gruppe 4
# für DB2 Knoten 3
GROUP_NAME=DB2NODE3
DB2_NODE=3
DISK_NAME=Disk F:
```

3. Führen Sie das Dienstprogramm DB2MSCS wie folgt aus:
db2mscs -f:DB2MSCS.CFG
4. Melden Sie sich vom Konto db2nt ab.

Der letzte Schritt ist das Registrieren der Datenbanklaufwerkzuordnung für die beiden MSCS-Cluster.

Registrieren der Datenbanklaufwerkzuordnung für ClusterA

Zum Registrieren der Datenbanklaufwerkzuordnung für MSCS-Cluster ClusterA sind folgende Schritte auszuführen:

1. Melden Sie sich auf Maschine db2test1 als Benutzer db2mpp an, d. h. mit dem den DB2-Services zugeordneten Konto. Das Kennwort ist db2mpp.
2. Geben Sie zum Registrieren der Datenbanklaufwerkzuordnung die folgenden Befehle ein:

```
db2drvmp add 0 F E
```

```
db2drvmp add 1 E F
```

3. Nehmen Sie alle DB2-Ressourcen offline, und bringen Sie sie dann wieder online.

Registrieren der Datenbanklaufwerkzuordnung für ClusterB

Zum Registrieren der Datenbanklaufwerkzuordnung für MSCS-Cluster ClusterB sind folgende Schritte auszuführen:

1. Melden Sie sich auf Maschine db2test3 als Benutzer db2mpp an, d. h. mit dem den DB2-Services zugeordneten Konto. Das Kennwort ist db2mpp.
2. Geben Sie zum Registrieren der Datenbanklaufwerkzuordnung die folgenden Befehle ein:

```
db2drvmp add 2 F E
```

```
db2drvmp add 3 E F
```

3. Nehmen Sie alle DB2-Ressourcen offline, und bringen Sie sie dann wieder online.

Verwalten von DB2 in einer MSCS-Umgebung

Der Einsatz von MSCS-Clustern bringt für das DB2-Exemplar zusätzliche Planung im Hinblick auf den täglichen Betrieb, die Datenbankverteilung und die Datenbankkonfiguration mit sich. Damit DB2 auf jedem MSCS-Knoten transparent ausgeführt werden kann, sind zusätzliche administrative Maßnahmen erforderlich. Alle von DB2 abhängigen Betriebssystemressourcen müssen auf allen MSCS-Knoten verfügbar sein. Einige dieser Betriebssystemressourcen gehören nicht zum Bereich von MSCS. Das heißt, sie können nicht als eine MSCS-Ressource definiert werden. Sie müssen daher sicherstellen, daß jedes System so konfiguriert ist, daß die gleichen Betriebssystemressourcen auf allen MSCS-Knoten zur Verfügung stehen. In den folgenden Abschnitten werden die zusätzlichen Maßnahmen beschrieben, die durchgeführt werden müssen.

Starten und Stoppen von DB2-Ressourcen

Sie müssen DB2-Ressourcen vom Clusterverwaltungsprogramm (Cluster Administrator) aus starten und stoppen. Es sind verschiedene Möglichkeiten

verfügbar, ein DB2-Exemplar zu starten, wie zum Beispiel der Befehl **db2start** und die Option **Dienste** in der Systemsteuerung. Wird DB2 jedoch nicht über Cluster Administrator gestartet, erkennt die MSCS-Software den Status des DB2-Exemplars nicht. Wenn ein DB2-Exemplar über Cluster Administrator gestartet und mit dem Befehl **db2stop** gestoppt wird, interpretiert die MSCS-Software den Befehl **db2stop** als Softwarefehler und versucht, DB2 erneut zu starten. (Die gegenwärtigen MSCS-Schnittstellen ermöglichen keine Benachrichtigung über einen *Ressourcenstatus*.)

Analog gilt, daß, wenn Sie ein DB2-Exemplar mit dem Befehl **db2start** starten, MSCS nicht erkennen kann, daß die Ressource online ist. Wenn ein Datenbank-Server ausfällt, bringt MSCS die DB2-Ressource in diesem Fall nicht auf der Übernahmemaschine im Cluster wieder online.

Drei Operationen können auf ein DB2-Exemplar angewendet werden:

Online

Diese Operation ist äquivalent zur Verwendung des Befehls **db2start**. Wenn DB2 bereits aktiv ist, kann diese Operation einfach dazu verwendet werden, MSCS darüber zu informieren, daß DB2 aktiv ist. Fehler während der Ausführung dieser Operation werden in das Windows NT-Ereignisprotokoll (Event Log) geschrieben.

Offline

Diese Operation ist äquivalent zur Verwendung des Befehls **db2stop**. Wenn es aktive Attach-Verbindungen zu einem Exemplar gibt, schlägt diese Operation fehl. Dies stimmt mit der Funktionsweise des Befehls **db2stop** überein.

Ressource abbrechen (Fail Resource)

Diese Operation ist äquivalent zur Verwendung des Befehls **db2stop** unter Angabe der Option **force**. DB2 trennt alle Anwendungen vom DB2-System und stoppt alle Datenbank-Server.

Ausführen von Prozeduren

Sie können Prozeduren ausführen, bevor eine DB2-Ressource online gebracht wird und nachdem Sie online gebracht wurde. Diese Prozeduren *müssen* sich im Exemplarprofilverzeichnis befinden, das für die Umgebungsvariable DB2INSTPROF angegeben ist. Dieses Verzeichnis ist der Verzeichnispfad, der durch den Parameter "-p" des Befehls **db2icrt** angegeben wird. Sie können diesen Wert durch folgenden Befehl abrufen:

```
db2set -i:exemplarname DB2INSTPROF
```

Dieser Dateipfad muß auf einer Platte im Cluster sein, so daß das Exemplarverzeichnis auf allen Clusterknoten verfügbar ist.

Diese Prozedurdateien sind nicht erforderlich und werden nur ausgeführt, wenn Sie im Exemplarverzeichnis gefunden werden. Sie werden vom MSCS-

Cluster-Service (Dienst) im Hintergrund gestartet. Die Prozedurdateien müssen Standardausgaben umleiten, um alle Ausgaben als Ergebnis von Befehlen innerhalb der Prozedurdateien aufzufangen. Die Ausgaben werden nicht auf dem Bildschirm angezeigt.

In einer partitionierten Datenbankumgebung wird standardmäßig die gleiche Prozedur von jedem Datenbankpartitions-Server im Exemplar verwendet. Wenn Sie zwischen verschiedenen Datenbankpartitions-Servern im Exemplar Unterschiede machen müssen, ordnen Sie unterschiedliche Werte der Umgebungsvariable DB2NODE zu, um bestimmte Zielknotennummern anzugeben. (Verwenden Sie dazu zum Beispiel die Anweisung IF in den Dateien db2cpre.bat und db2cpost.bat.)

Ausführen von Prozeduren, bevor DB2-Ressourcen online gebracht werden

Wenn Sie eine Prozedur ausführen wollen, *bevor* Sie eine DB2-Ressource online bringen, *muß* die Prozedur den Namen db2cpre.bat erhalten. DB2 ruft Funktionen auf, die diese Stapeldatei (Batch-Datei) über den Befehlszeilenprozessor (CLP) von Windows NT starten und auf die Beendigung der Ausführung durch den Befehlszeilenprozessor warten, bevor die DB2-Ressource online gebracht wird. Sie können diese Stapeldatei für Operationen wie das Ändern der DB2-Datenbankmanagerkonfiguration verwenden. Es ist zum Beispiel sinnvoll, einige Parameterwerte des Datenbankmanagers zu ändern, wenn das Übernahmesystem begrenzte Kapazitäten hat und die Systemressourcen, die von DB2 beansprucht werden, verringert werden müssen.

Die Befehle, die in die Prozedur db2cpre.bat gestellt werden, sollten synchron ausgeführt werden. Ansonsten könnte die DB2-Ressource online gebracht werden, bevor alle Operationen in der Prozedur beendet sind, wodurch sich unvorhergesehene Resultate ergeben können. Insbesondere sollte der Befehl **db2cmd** nicht in der Prozedur db2cpre.bat ausgeführt werden, weil er seinerseits einen anderen Befehlsprozessor startet, der DB2-Befehle asynchron zum Programm **db2cmd** ausführt.

Wenn Sie DB2-CLP-Befehle in der Prozedur db2cpre.bat ausführen wollen, sollten die Befehle in eine Datei gestellt und als CLP-Stapeldatei von einem Programm aus ausgeführt werden, das die DB2-Umgebung für den DB2-Befehlszeilenprozessor initialisiert und anschließend auf die Beendigung des DB2-Befehlszeilenprozessors wartet. Beispiel:

```
#include <windows.h>

int WINAPI DB2SetCLPEnv_api(DWORD pid);

void main ( int argc, char *argv [ ] )
{
    STARTUPINFO      startInfo  = {0};
    PROCESS_INFORMATION pidInfo  = {0};
```

```

char    title [32] = "Synchrone Ausführung";
char    runCmd [64] = "DB2 -z c:\\run.out -tvf c:\\run.clp";
/* Aufrufen der API zum Einrichten einer CLP-Umgebung */
if ( DB2SetCLPEnv_api (GetCurrentProcessId ()) == 0 ) 1
{
    startInfo.cb          = sizeof(STARTUPINFO);
    startInfo.lpReserved = NULL;
    startInfo.lpTitle    = title;
    startInfo.lpDesktop  = NULL;
    startInfo.dwX        = 0;
    startInfo.dwY        = 0;
    startInfo.dwXSize    = 0;
    startInfo.dwYSize    = 0;
    startInfo.dwFlags    = 0L;
    startInfo.wShowWindow = SW_HIDE;
    startInfo.lpReserved2 = NULL;
    startInfo.cbReserved2 = 0;
    if ( CreateProcessA( NULL,
                        runCmd, 2
                        NULL,
                        NULL,
                        FALSE,
                        NORMAL_PRIORITY_CLASS CREATE_NEW_CONSOLE,
                        NULL,
                        NULL,
                        &startInfo,
                        &pidInfo))
    {
        WaitForSingleObject (pidInfo.hProcess, INFINITE);
        CloseHandle (pidInfo.hProcess);
        CloseHandle (pidInfo.hThread);
    }
}
return;
}

```

1 Die API DB2SetCLPEnv_api wird durch die Importbibliothek DB2API.LIB aufgelöst. Diese API definiert eine Umgebung, die das Aufrufen von CLP-Befehlen ermöglicht. Wenn dieses Programm aus der Prozedur db2cpre.bat heraus aufgerufen wird, wird der Befehlsprozessor warten, bis die CLP-Befehle abgeschlossen sind.

2 runCmd ist der Name der Prozedurdatei, die die DB2-CLP-Befehle enthält.

Im Unterverzeichnis MISC des DB2-Installationspfads befindet sich ein Beispielprogramm namens db2clpex.exe. Diese ausführbare Datei ist dem angeführten Beispiel ähnlich, jedoch akzeptiert sie den DB2-CLP-Befehl als Befehlszeilenparameter. Wenn Sie dieses Beispielprogramm verwenden wollen, kopieren Sie es in das Unterverzeichnis BIN. Sie können diese ausführbare Datei in der Prozedur db2cpre.bat wie folgt verwenden (INSTHOME ist das Exemplarverzeichnis):


```
db2c1pex "DB2 -Z INSTHOME\pre.log -tvf INSTHOME\pre.clp"
```

Alle DB2-Befehle ATTACH oder Anweisungen CONNECT sollten explizit einen Benutzer angeben. Ansonsten werden sie unter dem Benutzerkonto ausgeführt, das dem Clusterdienst (Service) zugeordnet ist. CLP-Prozeduren sollten außerdem mit dem Befehl TERMINATE schließen, um den CLP-Hintergrundprozeß zu beenden.

Es folgt ein Beispiel einer Datei db2cpre.bat:

```
db2cpre.bat : 1
-----
db2c1pex "db2 -z INSTHOME\pre-%DB2NODE%.log -tvf INSTHOME\pre.clp" 2 - 5
-----

PRE.CLP 6
-----
update dbm cfg using MAXAGENTS 200;
get dbm cfg;
terminate;
-----
```

- 1** Die Prozedur db2cpre.bat wird unter dem Benutzerkonto ausgeführt, das dem Clusterdienst zugeordnet ist. Wenn DB2-Aktionen erforderlich sind, muß das dem Clusterdienst zugeordnete Benutzerkonto eine gültige SQL-Kennung sein, wie sie von DB2 definiert wird.
- 2** INSTHOME ist das Exemplarverzeichnis.
- 3** Der Name der Protokolldatei (Log) muß für jeden Knoten unterschiedlich sein, um Dateikonflikte zu vermeiden, wenn beide logischen Knoten gleichzeitig online gebracht werden.
- 4** db2c1pex.exe ist ein Beispielprogramm, das ein Befehlszeilenargument zur Angabe des auszuführenden CLP-Befehls akzeptiert.
- 5** Das Beispielprogramm db2c1pex.exe muß auf allen MSCS-Clusterknoten verfügbar gemacht werden.
- 6** Die CLP-Befehle in diesem Beispiel begrenzen die Anzahl von Agenten.

Ausführen von Prozeduren, nachdem DB2-Ressourcen online gebracht wurden

Wenn Sie eine Prozedur ausführen wollen, *nachdem* Sie eine DB2-Ressource online gebracht haben, *muß* die Prozedur den Namen db2cpost.bat erhalten. Die Prozedur wird von MSCS asynchron ausgeführt, wenn die DB2-Ressource erfolgreich online gebracht wurde. Der Befehl **db2cmd** kann in dieser Prozedur verwendet werden, um DB2-CLP-Prozedurdateien auszuführen. Verwenden Sie den Parameter *"-c"* des Befehls **db2cmd**, um anzugeben, daß das Dienstprogramm alle Fenster nach Beendigung der Operation schließen soll. Beispiel:

```
db2cmd -c db2 -tvf mycmds.clp
```

Der Parameter "-c" muß das erste Argument für den Befehl **db2cmd** sein, da er verwaiste Befehlsprozessoren im Hintergrund verhindert.

Die Prozedur `db2cpost.bat` ist nützlich, wenn Sie Datenbankaktivitäten unmittelbar, nachdem die DB2-Ressource von einer anderen Maschine übernommen und wieder aktiv wird, durchführen wollen. Zum Beispiel können Sie Datenbanken im Exemplar erneut starten oder aktivieren, so daß sie für den Benutzerzugriff vorbereitet sind.

Es folgt ein Beispiel einer Prozedur `db2cpost.bat`:

```
db2cpost.bat 1
-----
db2cmd -c db2 -z INSTHOME\post-%DB2NODE%.log -tvf INSTHOME\post.clp 2 - 4
-----

POST.CLP 5
-----
restart database SAMPLE;
connect reset;
activate database SAMPLE;
terminate;
-----
```

- 1** Die Prozedur `db2cpost.bat` wird unter dem Benutzerkonto ausgeführt, das dem Clusterdienst zugeordnet ist. Wenn DB2-Aktionen erforderlich sind, muß das dem Clusterdienst zugeordnete Benutzerkonto eine gültige SQL-Kennung sein, wie sie von DB2 definiert wird.
- 2** `INSTHOME` ist das Exemplarverzeichnis.
- 3** Der Name der Protokolldatei (Log) muß für jeden Knoten unterschiedlich sein, um Dateikonflikte zu vermeiden, wenn beide logischen Knoten gleichzeitig online gebracht werden.
- 4** Der Befehl **db2cmd** kann verwendet werden, weil die Prozedur `db2cpost.bat` asynchron ausgeführt werden kann. Der Parameter "-c" muß zur Beendigung des Befehlsprozessors verwendet werden.
- 5** Die CLP-Prozedur in diesem Beispiel enthält Befehle zum Neustarten und Aktivieren der Datenbank. Diese Prozedur versetzt die Datenbank unmittelbar nach dem Starten des Datenbankmanagers zurück in einen aktiven Status. In einem partitionierten Datenbanksystem sollten Sie den Befehl `ACTIVATE DATABASE` entfernen, weil mehrere DB2-Ressourcen gleichzeitig online gebracht werden. Der Befehl `RESTART DATABASE` kann fehlschlagen, weil ein anderer Knoten die Datenbank gerade aktiviert. Wenn dies eintritt, führen Sie die Prozedur erneut aus, um sicherzugehen, daß die Datenbank korrekt erneut gestartet wird.

Überlegungen zu Datenbanken

Stellen Sie bei der Erstellung einer Datenbank sicher, daß der Datenbankpfad eine freigegebene gemeinsame Platte angibt. Dadurch wird die Datenbank auf allen MSCS-Knoten sichtbar. Alle Protokolle und anderen Datenbankdateien müssen ebenfalls auf Platten im Cluster angelegt werden, damit die DB2-Funktionsübernahme erfolgreich durchgeführt werden kann. Wenn Sie diese Punkte nicht beachten, tritt ein DB2-Systemfehler auf, weil es für DB2 so aussieht, als ob Dateien gelöscht wurden oder nicht mehr verfügbar sind.

Stellen Sie darüber hinaus sicher, daß die Konfigurationsparameter des Datenbankmanagers und der Datenbank so eingestellt sind, daß die von DB2 beanspruchten Systemressourcen auf beiden MSCS-Knoten vorhanden sind. Der Konfigurationsparameter *autorestart* der Datenbank sollte auf ON gesetzt sein, so daß die erste Datenbankverbindung nach der Funktionsübernahme die Datenbank in einen konsistenten Status bringt. (Die Standardeinstellung für *autorestart* ist ON.) Die Datenbank kann auch mit Hilfe der Prozedur *db2cpost.bat* zum erneuten Starten und Aktivieren der Datenbank in einen bereiten Status versetzt werden. Diese Methode ist vorzuziehen, weil sie von *autorestart* unabhängig ist und die Datenbank unabhängig von einer Verbindungsanforderung eines Benutzers in einen bereiten Status versetzt wird.

Benutzer- und Gruppenunterstützung

DB2 ist für die Unterstützung der Benutzerauthentifizierung und Gruppen auf Windows NT angewiesen. Damit ein DB2-Exemplar eines MSCS-Knotens von einem anderen reibungslos übernommen werden kann, muß jeder MSCS-Knoten Zugriff auf die gleichen Sicherheitsdatenbanken von Windows NT haben. Sie können dies erreichen, indem Sie die Domänensicherheit von Windows NT (Domain Security) nutzen.

Definieren Sie alle DB2-Benutzer und DB2-Gruppen in einer Domänensicherheitsdatenbank. Die MSCS-Knoten müssen dieser Domäne angehören oder die Domäne muß eine akzeptierte Domäne (Trusted Domain) sein. DB2 verwendet dann die Domänensicherheitsdatenbank zur Authentifizierung und für die Gruppenunterstützung, unabhängig von dem MSCS-Knoten, auf dem DB2 aktiv ist.

Wenn Sie lokale Konten verwenden, müssen die Konten auf jedem MSCS-Knoten repliziert werden. Dieses Verfahren wird nicht empfohlen, weil es fehlerträchtig ist und eine doppelte Pflege verlangt.

DCE-Sicherheit ist ebenfalls eine unterstützte Identifikationsüberprüfungsart, wenn alle MSCS-Knoten Clients in der gleichen DCE-Zelle sind.

Sie sollten dem MSCS-Dienst ein Benutzerkonto zuordnen, das der DB2-Namenskonvention entspricht. Dadurch kann der MSCS-Dienst Aktionen für DB2 ausführen, die in den Prozeduren `db2cpre.bat` und `db2cpost.bat` aufgerufen werden.

Weitere Informationen zur Unterstützung von Benutzern und Gruppen unter Windows NT finden Sie in "Benutzerauthentifizierung mit DB2 für Windows NT" im Handbuch *Systemverwaltung: Implementierung*.

Überlegungen zur Kommunikation

DB2 unterstützt zwei LAN-Protokolle in einer MSCS-Umgebung:

- TCP/IP
- NetBIOS

TCP/IP wird unterstützt, weil es ein unterstützter Clusterressourcentyp ist. Zur Aktivierung von DB2 zur Verwendung von TCP/IP als Übertragungsprotokoll für ein partitioniertes Datenbanksystem, erstellen Sie eine IP-Adressenressource und platzieren sie in derselben Gruppe wie die DB2-Ressource, die den Datenbankpartitions-Server repräsentiert, den Sie als Koordinatorknoten für ferne Anwendungen verwenden wollen. Dann erstellen Sie eine Abhängigkeit mit Hilfe von Cluster Administrator, um sicherzustellen, daß die IP-Ressource online ist, bevor die DB2-Ressource gestartet wird. DB2-Clients können dann TCP/IP-Knotenverzeichniseinträge katalogisieren, um diese TCP/IP-Adresse zu verwenden.

Der TCP/IP-Anschluß, der dem Konfigurationsparameter `svcname` des Datenbankmanagers zugeordnet ist, muß zur Verwendung durch das DB2-Exemplar auf allen Maschinen reserviert werden, die am Exemplar beteiligt sind. Der Servicename, der der Anschlußnummer zugeordnet ist, muß außerdem in der Datei `services` auf allen Maschinen identisch sein.

Obwohl NetBIOS keine unterstützte Clusterressource ist, können Sie NetBIOS als LAN-Protokoll verwenden, weil das Protokoll sicherstellt, daß NetBIOS-Namen im LAN eindeutig sind. Wenn DB2 einen NetBIOS-Namen registriert, stellt NetBIOS sicher, daß der Name im LAN nicht bereits verwendet wird. Wenn DB2 im Szenario einer Funktionsübernahme von einem System auf ein anderes übertragen wird, wird der von DB2 verwendete `nname` von einer Partnermaschine im MSCS-Cluster deregistriert und auf der anderen Maschine registriert.

Die DB2-Unterstützung für NetBIOS verwendet NetBIOS-Frames (NBF). Dieser Protokollstapelspeicher kann verschiedenen logischen Adapternummern (LANA) zugeordnet werden. Damit ein konsistenter NetBIOS-Zugriff auf den Server sichergestellt ist, sollte die dem NBF-Protokollstapel zugeordnete LANA auf allen Knoten im Cluster gleich sein. Sie können dies mit Hilfe der

Option **Netzwerke** der Systemsteuerung konfigurieren. Sie sollten NBF der LANA 0 zuordnen, weil dies die von DB2 erwartete Standardeinstellung ist.

Überlegungen zur Systemzeit

DB2 verwendet die Systemzeit, um bestimmte Operationen mit einer Zeitmarke zu versehen. Alle MSCS-Knoten, die an einer DB2-Funktionsübernahme beteiligt sind, müssen eine synchronisierte Systemzeitzone und Systemzeit haben, um eine konsistente Funktionsweise von DB2 auf allen Maschinen sicherzustellen.

Stellen Sie die Systemzeitzone mit Hilfe der Option **Datum/Uhrzeit** der Systemsteuerung ein. MSCS hat einen Zeitdienst, der das Datum und die Uhrzeit synchronisiert, wenn die MSCS-Knoten zu einem Cluster verbunden werden. Der Zeitdienst synchronisiert die Zeit allerdings nur alle zwölf Stunden. Dadurch kann es zu Problemen kommen, wenn die Zeit auf einem System geändert wird und DB2 übernommen wird, bevor die Zeit synchronisiert wird.

Wenn die Zeit auf einem der MSCS-Clusterknoten geändert wird, sollte sie manuell auf den anderen Clusterknoten mit folgendem Befehl synchronisiert werden:

```
net time /set /y \\ferner_knoten
```

Dabei ist *ferner_knoten* der Maschinenname des Clusterknotens.

Überlegungen zu Verwaltungs-Server und Steuerzentrale in Umgebungen mit partitionierten Datenbanken

Der DB2-Verwaltungs-Server wird (wahlfrei) während der Installation von DB2 Universal Database erstellt. Er ist kein partitioniertes Datenbanksystem. Die Steuerzentrale verwendet vom Verwaltungs-Server zur Verfügung gestellte Dienste, um DB2-Exemplare und DB2-Datenbanken zu verwalten.

In einem partitionierten Datenbanksystem kann sich ein DB2-Exemplar auf mehreren Knoten befinden. Dies impliziert, daß ein DB2-Exemplar auf mehreren Systemen unter der Steuerzentrale katalogisiert sein muß, so daß der Zugriff auf das Exemplar erhalten bleibt, unabhängig davon, auf welchem MSCS-Knoten das DB2-Exemplar aktiv ist.

Das Exemplarverzeichnis des Verwaltungs-Servers wird nicht gemeinsam verwendet. Sie müssen alle benutzerdefinierten Dateien im Verzeichnis des Verwaltungs-Servers auf allen MSCS-Knoten spiegeln, um die gleiche Verwaltungsstufe auf allen MSCS-Knoten zur Verfügung zu stellen. Insbesondere müssen Benutzerprozeduren und nach Zeitplan terminierte ausführbare Dateien auf allen Knoten verfügbar sein. Außerdem müssen Sie sicherstellen, daß nach Zeitplan terminierte Aktivitäten auf allen Maschinen in einem MSCS-Cluster terminiert sind.

Alternativ zum Duplizieren des Verwaltungs-Servers auf allen Maschinen kann es wünschenswert sein, eine Funktionsübernahme für den Verwaltungs-Server zu ermöglichen. Nehmen Sie für das folgende Beispiel an, Sie hätten zwei MSCS-Knoten mit den Namen MACH0 und MACH1 im Cluster. MACH0 hat Zugriff auf eine Clusterplatte, die vom Verwaltungs-Server verwendet wird. Nehmen Sie außerdem an, daß MACH0 und MACH1 einen Verwaltungs-Server haben. Um den Verwaltungs-Server hochverfügbar zu machen, würden Sie folgende Schritte ausführen:

1. Stoppen Sie den Verwaltungs-Server auf beiden Maschinen, indem Sie den Befehl **db2admin stop** auf jeder Maschine ausführen.
2. Entkatalogisieren Sie auf allen Verwaltungs-Client-Maschinen alle Verweise auf die Verwaltungs-Server von MACH0 und MACH1 mit dem Befehl **UNCATALOG NODE**. (Mit Hilfe des Befehls **LIST NODE DIRECTORY** auf der Client-Maschine können Sie feststellen, ob Verweise auf den Verwaltungs-Server vorhanden sind.)
3. Löschen Sie den Verwaltungs-Server von MACH1, indem Sie von MACH1 aus den Befehl **db2admin drop** ausführen. (Dieser Schritt wäre nur erforderlich, wenn auf beiden Maschinen ein Verwaltungs-Server vorhanden wäre.)
4. Stellen Sie den Namen des Verwaltungs-Servers fest, indem Sie von MACH0 aus den Befehl **db2admin** ausführen. (Der Standardname ist **DB2DAS00**.)
5. Verwenden Sie das Dienstprogramm **DB2MSCS**, um die Unterstützung für die Funktionsübernahme des Verwaltungs-Servers einzurichten. Dies zieht die Erstellung einer DB2-Ressource auf MSCS namens **DB2DAS00** nach sich, die Abhängigkeiten von der IP-Ressource und den Plattenressourcen (**DISK**) hat. (Wenn Sie eine Konfiguration zur gegenseitigen Übernahme haben, würden Sie die Ressource in die Gruppe stellen, die die DB2-Ressource für **NODE0** enthält.) Diese Ressource wird als die MSCS-Ressource verwendet, die den Verwaltungs-Server unterstützt. Die Datei **DB2MSCS.ADMIN** sähe wie folgt aus:

```
#
# db2mscs.admin für Verwaltungs-Server
# run db2mscs -f:db2mscs.admin
#
DB2_INSTANCE=DB2DAS00
CLUSTER_NAME=CLUSTERA
DB2_LOGON_USERNAME=db2admin
DB2_LOGON_PASSWORD=db2admin
# den Verwaltungs-Server in die gleiche Gruppe wie DB2 Knoten 0 stellen
GROUP_NAME=DB2NODE0 1
DISK_NAME=DISK E:
INSTPROF_DISK=DISK E:
IP_NAME= IP-Adresse für Verwaltungs-Server
IP_ADDRESS=9.9.9.8
IP_SUBNET=255.255.255.0
IP_NETWORK=Ethernet
```

1 Diese Gruppe kann die gleiche wie die vorhandene sein. Auf diese Weise benötigen Sie keine zusätzliche Platte für das Exemplarprofilverzeichnis.

6. Führen Sie auf MACH1 den folgenden Befehl aus, um DB2DAS00 als den Verwaltungs-Server zu definieren:

```
db2set -g db2adminserver=DB2DAS00
```

7. Ändern Sie auf MACH0 die Startmerkmale von DB2DAS00 über das Programm Services, so daß er manuell, und nicht automatisch, gestartet wird, da DB2DAS00 nun von MSCS gesteuert wird.

Wenn der Verwaltungs-Server für die Funktionsübernahme eingerichtet ist, sollten alle fernen Zugriffe eine MSCS-IP-Ressource zur Kommunikation mit dem Verwaltungs-Server verwenden. Der Verwaltungs-Server hat nun die folgenden Merkmale:

- Das Exemplarverzeichnis des Verwaltungs-Servers wird zusammen mit dem Verwaltungs-Server übernommen.
- Clients katalogisieren nur einen einzigen Knoten zur Kommunikation mit dem Verwaltungs-Server, unabhängig davon, auf welchem MSCS-Knoten der Verwaltungs-Server aktiv ist.
- Jobs müssen nur einmal für den Verwaltungs-Server terminiert werden.
- Lokale Exemplare können vom Verwaltungs-Server nur gesteuert werden, wenn der Verwaltungs-Server auf demselben Knoten aktiv ist wie das lokale Exemplar.
- Der Zugriff auf den Verwaltungs-Server ist nicht möglich, wenn der Cluster-Service nicht aktiv ist.

Einschränkungen und Bedingungen

Bei der Ausführung von DB2 in einer MSCS-Umgebung gibt es folgende Einschränkungen:

- Sie können auf den gemeinsamen Platten keine physische E/A verwenden, sofern die gemeinsamen Platten nicht dieselbe physische Plattennummer auf beiden MSCS-Knoten haben. Sie können logische E/A verwenden, weil auf die Platte mit Hilfe einer Partitionskenung zugegriffen wird.
- Sie müssen alle DB2-Ressourcen für MSCS-Unterstützung konfigurieren. Andernfalls treten Systemfehler während der Laufzeit von DB2 auf (DB2 kann nicht ordnungsgemäß funktionieren, wenn Systemressourcen fehlen). Wenn zum Beispiel die Datenbankprotokolle nicht auf einer gemeinsamen MSCS-Platte gespeichert werden, kann DB2 die Datenbank nicht erneut starten.
- Sie müssen das DB2-Exemplar über das Programm Cluster Administrator verwalten. MSCS betrachtet andere Methoden, die zum Starten und Stoppen des Datenbankmanagers verwendet werden, als Software-unregelmäßigkeiten. Wenn Sie zum Beispiel DB2 über MSCS starten und den Befehl **db2stop** zum Stoppen von DB2 verwenden, erkennt MSCS dies

als Softwarefehler und startet das Exemplar erneut. Dies heißt natürlich auch, daß Sie die Steuerzentrale nicht zum Starten und Stoppen von DB2 verwenden sollten.

- Zum Deinstallieren von DB2 müssen Sie MSCS zuerst stoppen.

Kapitel 14. DB2 und hohe Verfügbarkeit unter Sun Cluster 2.2

Dieses Kapitel beschreibt detailliert, wie DB2 mit Sun Cluster 2.x (SC2.x) arbeitet, um hohe Verfügbarkeit zu erreichen, und enthält eine Beschreibung des Agenten für hohe Verfügbarkeit (HA-Agent), der als Mittler zwischen zwei Softwareprodukten fungiert (siehe Abb. 59).



Abbildung 59. DB2, Sun Cluster 2.x und hohe Verfügbarkeit

Hohe Verfügbarkeit

Computer-Systeme, die als Host für Datenbankservices dienen, enthalten viele unterschiedliche Komponenten, und jede Komponente besitzt eine für sie ermittelte „mittlere ausfallfreie Zeit“ (MTBF - Mean Time Before Failure). Die MTBF ist die durchschnittliche Dauer, die eine Komponente verwendbar bleibt. Die MTBF für ein Qualitätsfestplattenlaufwerk liegt in der Größenordnung von 1 Million Stunden (annähernd 114 Jahren). Obwohl dies wie ein langer Zeitraum erscheint, ist es wahrscheinlich, daß eine von 200 Festplatten innerhalb eines Zeitraums von sechs Monaten ausfällt.

Wenngleich es eine Reihe von Methoden zur Erhöhung der Verfügbarkeit für einen Datenbankservice gibt, ist die am häufigsten eingesetzte Methode ein HA-Cluster (High Availability-Cluster). Ein Cluster, wenn er zur Implementierung hoher Verfügbarkeit eingesetzt wird, besteht aus mindestens zwei Maschinen, einer Gruppe privater Netzchnittstellen, einer oder mehreren öffentlichen Netzchnittstellen sowie einigen gemeinsam benutzten Platten. Diese spezielle Konfiguration ermöglicht das Versetzen eines Datenbankservice von einer Maschine auf eine andere. Durch das Versetzen des Datenbankservice auf eine andere Maschine im Cluster wird es möglich, den Zugriff auf die zugehörigen Daten weiterhin aufrecht zu erhalten. Das Versetzen eines

Datenbankservice von einer Maschine auf eine andere wird als *Funktionsübernahme* (Failover) bezeichnet (siehe Abb. 60 zur Illustration). Die privaten Netzanschlüsse werden zum Senden von *Heartbeat*

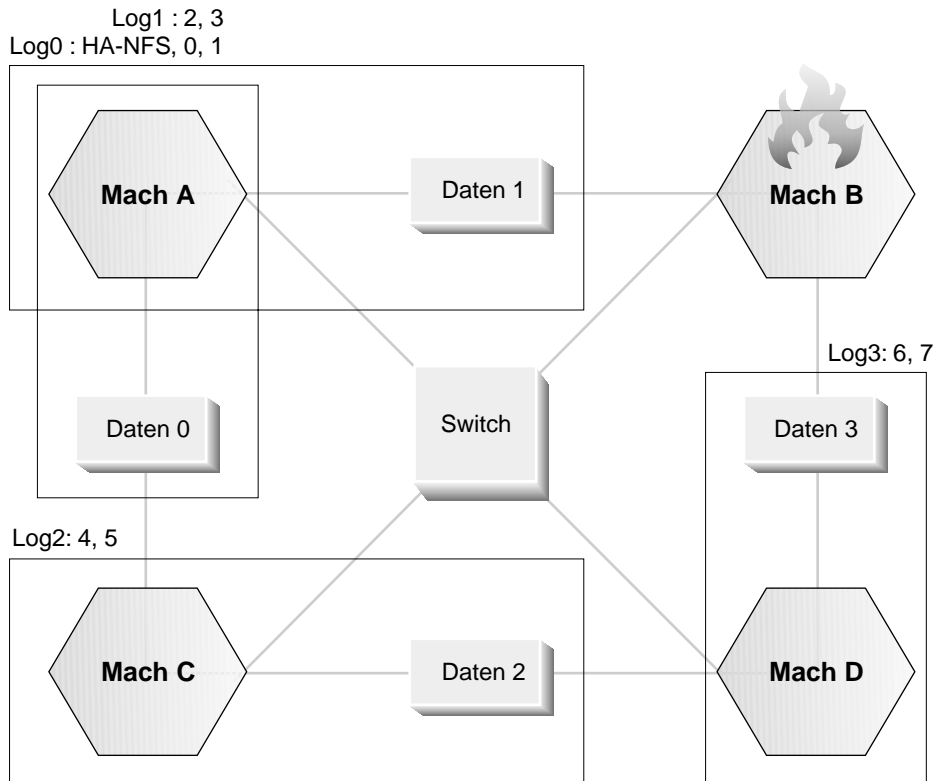


Abbildung 60. Funktionsübernahme

Nachrichten und von Steuernachrichten unter den Maschinen im Cluster verwendet. Die öffentlichen Netzanschlüsse dienen zur direkten Kommunikation mit Clients des HA-Clusters. Die Platten in einem HA-Cluster sind mit zwei oder mehr Maschinen im Cluster verbunden, so daß bei Ausfall einer Maschine eine andere Maschine auf sie zugreifen kann.

Ein Datenbankservice, der auf einem HA-Cluster aktiv ist, besitzt mindestens eine öffentliche Netzanschlüsse und eine Gruppe von Platten, die ihm zugeordnet sind. Die Clients eines HA-Datenbankservice stellen eine Verbindung über TCP/IP nur zu den logischen Netzanschlüssen des Datenbankservice her. Wenn es zu einer Funktionsübernahme kommt, werden der Datenbankservice zusammen mit den zugehörigen Netzanschlüssen und der Gruppe von Platten auf eine andere Maschine verlegt.

Einer der Vorteile eines HA-Clusters liegt darin, daß ein Datenbankservice ohne Hilfe durch Unterstützungspersonal wiederhergestellt werden kann, und daß dies zu jedem beliebigen Zeitpunkt möglich ist. Ein weiterer Vorteil ist die Redundanz. Alle Teile im Cluster sollten redundant sein, einschließlich der Maschinen selbst. Der Cluster sollte in der Lage sein, jeden einzelnen Fehlerpunkt überbrücken zu können.

Auch wenn sich hochverfügbare Datenbankservices in ihrer Art sehr unterscheiden können, besitzen sie einige allgemeine Anforderungen. Clients eines hochverfügbaren Datenbankservice erwarten, daß sich die Netzadresse und der Host-Name des Datenbankservice nicht ändern und daß sie die Möglichkeit haben, ihre Anforderungen unabhängig von der Maschine, auf der der Datenbankservice aktiv ist, auf dieselbe Weise zu tätigen.

Betrachten Sie einen Web-Browser, der auf einen hochverfügbaren Web-Server zugreift. Die Anforderung wird mit einer URL-Adresse (Uniform Resource Locator) abgesetzt, die sowohl einen Host-Namen als auch den Pfad zu einer Datei auf dem Web-Server enthält. Der Browser erwartet, daß sowohl der Host-Name als auch der Pfad auch nach einer Funktionsübernahme des Web-Servers gleich bleiben. Wenn der Browser gerade eine Datei vom Web-Server herunterlädt, wenn der Server durch eine Funktionsübernahme auf eine andere Maschine versetzt wird, muß der Browser die Anforderung erneut absetzen.

Die Verfügbarkeit eines Datenbankservice wird in der Zeitdauer gemessen, die der Datenbankservice seinen Benutzern zur Verfügung steht. Die am häufigsten verwendete Maßeinheit für Verfügbarkeit ist der Prozentsatz der Betriebszeit (Up Time). Diese wird oft als Anzahl von „Neunen“ angegeben:

99,99% => Service ist (höchstens) 52,6 Minuten / Jahr außer Betrieb
99,999% => Service ist (höchstens) 5,26 Minuten / Jahr außer Betrieb
99,9999% => Service ist (höchstens) 31,5 Sekunden / Jahr außer Betrieb

Beachten Sie beim Entwerfen und Testen eines HA-Clusters folgende Punkte:

1. Stellen Sie sicher, daß der Administrator des Clusters mit dem System vertraut ist und weiß, was geschehen soll, wenn eine Funktionsübernahme auftritt.
2. Stellen Sie sicher, daß jeder Teil des Clusters wirklich redundant ist und im Falle eines Ausfalls rasch ausgetauscht werden kann.
3. Erzwingen Sie die Funktionsübernahme eines Testsystems in einer kontrollierten Umgebung, und stellen Sie sicher, daß die Funktionsübernahme jedesmal ordnungsgemäß vollzogen wird.
4. Protokollieren Sie die Gründe für jede eingetretene Funktionsübernahme. Obwohl dies nicht oft geschehen sollte, ist es wichtig, alle Punkte zu untersuchen, die einen Cluster instabil machen. Wenn zum Beispiel eine

Komponente des Clusters fünfmal in einem Monat eine Funktionsübernahme erforderlich machte, stellen Sie fest, woran dies liegt, und beheben Sie das Problem.

5. Stellen Sie sicher, daß das Unterstützungspersonal für den Cluster benachrichtigt wird, wenn eine Funktionsübernahme eintritt.
6. Überlasten Sie den Cluster nicht. Stellen Sie sicher, daß die verbliebenen Systeme nach einer Funktionsübernahme die Auslastung weiterhin mit einer akzeptablen Leistung bewältigen können.
7. Überprüfen Sie ausfallanfälligere Komponenten (z. B. Festplatten) häufig, so daß sie ersetzt werden können, bevor Probleme auftreten.

Fehlertoleranz und fortlaufende Verfügbarkeit

Eine andere Methode zur Erhöhung der Verfügbarkeit eines Datenbankservice ist Fehlertoleranz. Bei einer *fehlertoleranten* Maschine ist sämtliche Redundanz integriert, so daß die Maschine in der Lage ist, einen Einzelausfall einer beliebigen Komponente, einschließlich CPU und Speicher, zu überbrücken. Fehlertolerante Maschinen werden hauptsächlich in Nischenmärkten eingesetzt und ihre Implementierung ist gewöhnlich mit hohem Kostenaufwand verbunden. Ein HA-Cluster mit Maschinen an verschiedenen geografischen Standorten hat den zusätzlichen Vorteil, einen Ausfall, der nur einen Teil dieser Standorte betrifft, überbrücken zu können.

Die *fortlaufende Verfügbarkeit* geht einen Schritt über die hohe Verfügbarkeit hinaus. Sie schützt die Clients sowohl gegen geplante als auch gegen ungeplante Ausfallzeiten. Bei einer Konfiguration für fortlaufende Verfügbarkeit bleibt der Client von Ausfällen einer der Maschinen, die den Datenbankservice beherbergen, bzw. von Wartungszeiten für einzelne Maschinen völlig unberührt. Konfigurationen für fortlaufende Verfügbarkeit sind komplex und kostenintensiver in der Implementierung.

Ein HA-Cluster ist die gängigste Lösung zur Erhöhung der Verfügbarkeit, weil sie skalierbar, einfach zu handhaben und relativ kostengünstig in der Implementierung ist.

Sun Cluster 2.2

Sun Cluster 2.2 (SC2.2) ist das Produkt zur Clusterung und Implementierung hoher Verfügbarkeit von Sun Microsystems. SC2.2 unterstützt bis zu vier Maschinen in einem einzigen Cluster. Bei Einsatz von vier Ultra Enterprise 10000s kann ein Cluster bis zu 256 CPUs und 256 GB RAM besitzen.

Unterstützte Systeme

System	UltraSPARC	Speicherkapazität	E/A
Ultra Enterprise 1	1	64 MB - 1 GB	3 SBus

Ultra Enterprise 2	1-2	64 MB - 2 GB	4 SBus
Ultra Enterprise 450	1-4	32 MB - 4 GB	10 PCI
Ultra Enterprise 3000	1-6	64 MB - 6 GB	9 SBus
Ultra Enterprise 4000	1-14	64 MB - 14 GB	21 SBus
Ultra Enterprise 5000	1-14	64 MB - 14 GB	21 SBus
Ultra Enterprise 6000	1-30	64 MB - 30 GB	45 SBus
Ultra Enterprise 10000	1-64	512 MB - 64 GB	64 SBus

Agenten

Die Sun Cluster-Software enthält eine Anzahl von Agenten für hohe Verfügbarkeit (HA-Agenten), die unterstützt und mit dem Produkt SC2.2 ausgeliefert werden. Andere HA-Agenten, wie zum Beispiel der für DB2, werden außerhalb von Sun entwickelt und nicht zusammen mit der Sun Cluster-Software ausgeliefert. Der HA-Agent für DB2 gehört zum Lieferumfang von DB2 und wird von IBM unterstützt.

Die Sun Cluster-Software arbeitet mit hochverfügbaren Datenbankservices, indem sie eine Möglichkeit gibt, Methoden (Prozeduren (Skripte) bzw. Programme) zu registrieren, die verschiedenen Komponenten der Sun Cluster-Software entsprechen. Mit Hilfe dieser Methoden kann die Software von SC2.2 einen Datenbankservice steuern, ohne über eingehende Kenntnisse über ihn zu verfügen. Zu diesen Methoden gehören folgende:

START

Dient zum Starten von Teilen des Datenbankservice, bevor die logischen Netzschnittstellen online sind.

START_NET

Dient zum Starten von Teilen des Datenbankservice, nachdem die logischen Netzschnittstellen online gebracht sind.

STOP Dient zum Stoppen von Teilen des Datenbankservice, nachdem die logischen Netzschnittstellen offline genommen wurden.

STOP_NET

Dient zum Stoppen von Teilen des Datenbankservice, bevor die logischen Netzschnittstellen offline genommen werden.

ABORT

Wie die Methode STOP, jedoch wird sie ausgeführt unmittelbar, bevor eine Maschine durch die Cluster-Software heruntergefahren wird. In

diesem Fall ist der einwandfreie Zustand der Maschine fraglich, und ein Datenbankservice könnte noch einige „letzte“ Anforderungen ausführen, bevor die Maschine heruntergefahren wird. Wird ausgeführt, nachdem die logischen Netzschnittstellen offline genommen wurden.

ABORT_NET

Wie die Methode ABORT, allerdings wird sie ausgeführt, bevor die logischen Netzschnittstellen offline genommen werden.

FM_INIT

Dient zur Initialisierung von Fehlermonitoren.

FM_START

Dient zum Starten von Fehlermonitoren.

FM_STOP

Dient zum Stoppen von Fehlermonitoren.

FM_CHECK

Wird durch den Befehl **hactl** aufgerufen. Liefert den aktuellen Status des entsprechenden Datenbankservice.

Der DB2-Agent besteht aus folgenden Prozeduren: START_NET, STOP_NET, FM_START und FM_STOP. Die folgenden Prozeduren werden bei der Neukonfiguration des Clusters nicht ausgeführt: ABORT, ABORT_NET und FM_CHECK.

Ein Agent für hohe Verfügbarkeit besteht aus mindestens einer dieser Methoden. Die Methoden werden bei SC2.2 durch den Befehl **hareg** registriert. Nach der Registrierung ruft die Sun Cluster-Software die entsprechende Methode auf, um den Datenbankservice zu steuern.

Es ist sehr wichtig, zu beachten, daß die Methoden ABORT und STOP eines Service möglicherweise nicht aufgerufen werden. Diese Methoden sind zum kontrollierten Herunterfahren eines Datenbankservice gedacht, und der Datenbankservice muß wiederhergestellt werden können, wenn eine Maschine ausfällt, ohne diese Methoden aufzurufen.

Weitere Informationen finden Sie in der Sun Cluster-Dokumentation.

Logische Hosts

Die SC2.2-Software arbeitet mit dem Konzept eines logischen Hosts. Ein *logischer Host* besteht aus einer Gruppe von Platten und einer oder mehreren öffentlichen Netzschnittstellen. Ein hochverfügbarer Datenbankservice wird einem logischen Host zugeordnet und benötigt die Platten, die in den Plattengruppen des logischen Hosts sind. Logische Hosts können auf verschiedenen Maschinen im Cluster untergebracht sein und sich die CPUs und den Hauptspeicher der Maschine „entleihen“, auf der sie aktiv sind.

Logische Netzchnittstellen

Ebenso wie andere auf UNIX basierende Betriebssysteme verfügt Solaris über die Möglichkeit, neben der primären IP-Adresse für eine Netzchnittstelle zusätzliche IP-Adressen zu besitzen. Die zusätzlichen IP-Adressen befinden sich auf einer logischen Schnittstelle in der gleichen Weise, wie sich die primäre IP-Adresse auf der physischen Netzchnittstelle befindet. Das folgende Beispiel zeigt die logischen Schnittstellen auf zwei Maschinen in einem Cluster. Es gibt zwei logische Hosts, die sich zur Zeit beide auf der Maschine "thrash" befinden.

```
scadmin@crackle(202)# netstat -in
Name Mtu Net/Dest Address Ipkts Ierrs Opkts Oerrs Collis Queue
lo0 8232 127.0.0.0 127.0.0.1 289966 0 289966 0 0 0
hme0 1500 9.21.55.0 9.21.55.98 121657 6098 764122 0 0 0
scid0 16321 204.152.65.0 204.152.65.1 489307 0 476479 0 0 0
scid0:1 16321 204.152.65.32 204.152.65.33 0 0 0 0 0 0
scid1 16321 204.152.65.16 204.152.65.17 347317 0 348073 0 0 0
```

1. lo0 ist die loopback-Schnittstelle
2. hme0 ist die öffentliche Netzchnittstelle (ethernet)
3. scid0 ist die erste private Netzchnittstelle (SCI oder Scalable Coherent Interface)
4. scid0:1 ist eine logische Netzchnittstelle, die von der Sun Cluster-Software intern verwendet wird
5. scid1 ist die zweite private Netzchnittstelle

```
scadmin@thrash(203)# netstat -in
Name Mtu Net/Dest Address Ipkts Ierrs Opkts Oerrs Collis Queue
lo0 8232 127.0.0.0 127.0.0.1 1128780 0 118780 0 0 0
hme0 1500 9.21.55.0 9.21.55.92 1741422 5692 757127 0 0 0
hme0:1 1500 9.21.55.0 9.21.55.109 0 0 0 0 0 0
hme0:2 1500 9.21.55.0 9.21.55.110 0 0 0 0 0 0
scid0 16321 204.152.65.0 204.152.65.2 476641 0 489476 0 0 0
scid0:1 16321 204.152.65.32 204.152.65.34 0 0 0 0 0 0
scid1 16321 204.152.65.16 204.152.65.18 348199 0 347444 0 0 0
```

1. hme0:1 ist eine logische Netzchnittstelle für einen logischen Host
2. hme0:2 ist eine log. Netzchnittstelle für einen anderen log. Host

Einem logischen Host können eine oder mehrere logische Schnittstellen zugeordnet sein. Diese logischen Schnittstellen werden mit dem logischen Host von einer Maschine zur anderen versetzt und dienen dem Zugriff auf den Datenbankservice, der dem logischen Host zugeordnet ist. Da diese logischen Schnittstellen mit den logischen Hosts versetzt werden, können Clients auf den Datenbankservice unabhängig von der Maschine, auf dem er sich befindet, zugreifen.

Ein hochverfügbarer Datenbankservice sollte an die TCP/IP-Adresse INADDR_ANY gebunden werden. Dadurch wird gewährleistet, daß jede IP-Adresse auf dem System Verbindungen für den Datenbankservice empfangen kann. Wenn ein Datenbankservice statt dessen an eine bestimmte IP-Adresse gebun-

den wird, muß er an die logische Schnittstelle gebunden werden, die dem logischen Host zugeordnet ist, auf dem der Datenbankservice aktiv ist. Das Binden an INADDR_ANY beseitigt die Notwendigkeit, den Service an eine neue IP-Adresse zu binden, wenn eine auf dem System eintrifft, die vom Datenbankservice benötigt wird.

Anmerkung: Clients eines HA-Exemplars sollten die Datenbank mit dem Host-Namen für die logische IP-Adresse eines logischen Hosts katalogisieren. Sie sollten nie den primären Host-Namen für eine Maschine verwenden, weil es keine Garantie gibt, daß DB2 auf dieser Maschine ausgeführt wird.

Plattengruppen und Dateisysteme

Platten für einen Datenbankservice werden einem logischen Host in Gruppen (bzw. Sätzen) zugeordnet. Wenn der Cluster mit Sun StorEdge Volume Manager (Veritas) arbeitet, verwendet die Sun Cluster-Software das Veritas-Dienstprogramm "vxdg", um die Plattengruppen für jeden logischen Host zu importieren und zu deportieren. Das folgende Beispiel zeigt die Plattengruppen für zwei logische Hosts "log0" und "log1", die auf einer Maschine namens "thrash" untergebracht sind. Die Maschine mit dem Namen "crackle" beherbergt zur Zeit keine logischen Hosts.

```
scadmin@crackle(206)# vxdg list
NAME STATE ID
rootdg enabled 899825206.1025.crackle

scadmin@thrash(205)# vxdg list
NAME STATE ID
rootdg enabled 924176206.1025.thrash
data0 enabled 925142028.1157.crackle=
data1 enabled 899826248.1108.crackle
```

Die Plattengruppen "data0" und "data1" entsprechen jeweils den logischen Hosts "log0" und "log1". Die Plattengruppe "data0" kann von "thrash" mit folgendem Befehl deportiert werden:

```
vxdg deport data0
```

Und mit folgendem Befehl auf "crackle" importiert werden:

```
vxdg import data1
```

Dies wird durch die Sun Cluster-Software automatisch durchgeführt und sollte auf einem im Betrieb befindlichen Cluster nicht manuell vorgenommen werden.

Jede Plattengruppe enthält eine Anzahl von Platten, die von zwei oder mehr Maschinen im Cluster gemeinsam benutzt werden können. Ein logischer Host kann nur zu einer anderen Maschine versetzt werden, die physischen Zugriff auf die Platten in den Plattengruppen hat, die ihm zugeordnet sind.

Zwei Dateien steuern die Dateisysteme für jeden logischen Host:

```
/etc/opt/SUNWcluster/conf/hanfs/vfstab.<logischer_host>
/etc/opt/SUNWcluster/conf/hanfs/dfstab.<logischer_host>
```

Dabei ist *logischer_host* der Name des zugeordneten logischen Hosts.

Die Datei *vfstab* ist der Datei */etc/vfstab* ähnlich, abgesehen davon, daß sie Einträge für die Dateisysteme enthält, die angehängt (mount) werden sollen, nachdem die Plattengruppen für einen logischen Host importiert wurden. Die Datei *dfstab* ist der Datei */etc/dfs/dfstab* ähnlich, jedoch enthält sie Einträge für die Dateisysteme, die über HA-NFS für einen logischen Host exportiert werden sollen. Jede Maschine verfügt über eine eigene Kopie dieser Datei, und es sollte sorgfältig darauf geachtet werden, daß sie auf jeder Maschine im Cluster den gleichen Inhalt haben.

Anmerkung: Die Pfade für die Dateien *vfstab* und *dfstab* eines logischen Hosts sind irreführend, weil sie das Verzeichnis *hanfs* enthalten. Nur die Datei *dfstab* für einen logischen Host wird für HA-NFS verwendet. Die Datei *vfstab* wird verwendet, selbst wenn HA-NFS nicht konfiguriert ist.

Die folgenden Beispiele stammen aus einem Cluster, auf dem DB2 Universal Database Enterprise - Extended Edition (EEE) in einer Konfiguration zur gegenseitigen Übernahme ausgeführt wird:

```
scadmin@thrash(217)# ls -l /etc/opt/SUNWcluster/conf/hanfs
total 8
-rw-r--r-- 1 root build 173 Apr 14 15:01 dfstab.log0
-rw-r--r-- 1 root build 316 Apr 26 12:07 vfstab.log0
-rw-r--r-- 1 root build 389 Apr 13 21:04 vfstab.log1
```

```
scadmin@thrash(218)# cat dfstab.log0
share -F nfs -o root=crackle:thrash:\
jolt:bump:crackle.torolab.ibm.com:thrash.torolab.ibm.com:\
jolt.torolab.ibm.com:bump.torolab.ibm.com /log0/home
```

Die Hosts, denen die Berechtigung zum Anhängen (mount) des Dateisystems */log0/home* erteilt wird, stammen von allen Netzschnittstellen (logischen und physischen) auf jeder Maschine im Cluster. Die Dateisysteme werden mit Root-Berechtigung exportiert.

```
scadmin@thrash(220)# cat vfstab.log0
#device to mount          device to fsck          mount          FS  fsck mount
#                          point                   type           type pass at boot

/dev/vx/dsk/data0/data1-stat /dev/vx/rdsk/data0/data1-stat /log0          ufs  2   no
/dev/vx/dsk/data0/vol01     /dev/vx/rdsk/data0/vol01     /log0/home    ufs  2   no
/dev/vx/dsk/data0/vol02     /dev/vx/rdsk/data0/vol02     /log0/data    ufs  2   no
```

```
scadmin@thrash(221)# cat vfstab.log1
#device to mount          device to fsck          mount          FS  fsck mount
```

#		point	type	pass	at boot
	/dev/vx/dsk/data1/data1-stat	/dev/vx/rdisk/data1/data1-stat	/log1	ufs	2 no
	/dev/vx/dsk/data1/vol01	/dev/vx/rdisk/data1/vol01	/log1/home	ufs	2 no
	/dev/vx/dsk/data1/vol02	/dev/vx/rdisk/data1/vol02	/log1/data	ufs	2 no
	/dev/vx/dsk/data1/vol03	/dev/vx/rdisk/data1/vol03	/log1/data1	ufs	2 no

Die Datei `vfstab.log0` enthält drei gültige Einträge für Dateisysteme unter dem Verzeichnis `/log0`. Beachten Sie, daß die Dateisysteme für den logischen Host `log0` logische Datenträgereinheiten (volume devices) verwenden, die Teil der Plattengruppe `data0` sind, die dem logischen Host zugeordnet ist.

Die Dateisysteme in den Dateien `vfstab` werden der Reihe nach von oben nach unten angehängt (mount), so daß es wichtig ist, sicherzustellen, daß die Dateisysteme in der richtigen Reihenfolge aufgelistet sind. Dateisysteme, die unterhalb eines bestimmten Dateisystems angehängt werden, müssen unter diesem Dateisystem aufgelistet werden. Die tatsächlich für einen logischen Host benötigten Dateisysteme hängen von den Anforderungen des Datenbankservice ab und werden sich von diesen Beispielen erheblich unterscheiden.

Während einer Funktionsübernahme ist die SC2.2-Software dafür zuständig, zu gewährleisten, daß die Plattengruppen und logischen Schnittstellen, die einem logischen Host zugeordnet sind, dem Host im Cluster von Maschine zu Maschine folgen. Der hochverfügbare Datenbankservice erwartet, daß ihm zumindest diese Ressourcen auf einem neuen System nach einer Funktionsübernahme zur Verfügung stehen. Tatsächlich ist es so, daß die meisten Datenbankservices selbst gar nicht erkennen, daß sie hochverfügbar sind, so daß diese Ressourcen auch nach einer Funktionsübernahme als exakt die gleichen erscheinen müssen.

Steuermethoden

Die Steuermethoden werden mit folgendem Befehl registriert:

```
hareg(1m)
```

Wenn ein HA-Service registriert ist, ist SC2.2 dafür zuständig, die Methoden, die für den HA-Service registriert wurden, zu gegebener Zeit während einer Änderung der Clusterkonfiguration oder einer Funktionsübernahme aufzurufen.

Während einer Änderung der Clusterkonfiguration (kontrollierten Funktionsübernahme) finden die folgenden Aktionen (in der angegebenen Reihenfolge) statt. Die Aktionen vor Schritt 5c werden nicht durchgeführt, falls die Maschine abstürzt. (Weitere Informationen zur Änderung der Clusterkonfiguration finden Sie in der SC2.2-Dokumentation.)

1. Methode FM_STOP wird ausgeführt.
 2. Methode STOP_NET wird ausgeführt.
 3. Logische Schnittstellen für den logischen Host werden offline gebracht.
 - ifconfig hme0:1 0.0.0.0 down
 4. Methode STOP wird ausgeführt.
 5. Plattengruppen und Dateisysteme werden versetzt.
 - a. Abhängen von Dateisystemen des logischen Hosts.
 - b. vxdg deportiert Plattengruppen auf einer Maschine.
- - Nur die folgenden Schritte werden ausgeführt, wenn eine Maschine abstürzt - -
- c. vxdg importiert Plattengruppen auf der anderen Maschine.
 - d. Ausführen von fsck für Dateisysteme des logischen Hosts.
 - e. Anhängen (Mount) der Dateisysteme des logischen Hosts.
 6. Methode START wird ausgeführt.
 7. Logische Schnittstellen für den logischen Host werden online gebracht.
 - ifconfig hme0:1 <ip address> up
 8. Methode START_NET wird ausgeführt.
 9. Methode FM_INIT wird ausgeführt.
 10. Methode FM_START wird ausgeführt.

Die Steuermethoden werden mit den folgenden Befehlszeilenparametern ausgeführt:

```
METHODE <untergebrachte logische Hosts> <nicht untergebrachte logische Hosts> <Zeitl
```

Der erste Parameter ist eine durch Kommas getrennte Liste logischer Hosts, die momentan auf der Maschine untergebracht sind, und der zweite Parameter eine durch Kommas getrennte Liste logischer Hosts, die nicht auf der Maschine untergebracht sind. Der letzte Parameter ist ein Zeitlimit für die Methode, d. h. die Zeit, die die Methode aktiv sein kann, bevor die SC2.2-Software sie abbricht.

Konfiguration von Platten und Dateisystemen

SC2.2 unterstützt zwei Datenträgermanager: Sun StorEdge Volume Manager (Veritas) und Solstice Disk Suite. Obwohl beide gut funktionieren, besitzt StorEdge Volume Manager einige Vorteile in einer Clusterumgebung. In einigen Clusterkonfigurationen kann die Controllernummer für eine Platteneinheit von Maschine zu Maschine im Cluster unterschiedlich sein. Wenn sich eine Controllernummer unterscheidet, unterscheiden sich auch die Pfade für die Platteneinheiten für den Controller. Da Disk Suite direkt mit den Pfaden der Plattengeräte arbeitet, funktioniert das Produkt in diesem Fall nicht gut. StorEdge Volume Manager arbeitet hingegen mit den Platten selbst, unabhängig von der Controllernummer, und wird durch unterschiedliche Controllernummern nicht beeinträchtigt.

Da das Ziel von HA in einer Erhöhung der Verfügbarkeit für einen Datenbankservice liegt, spielt es eine wichtige Rolle, sicherzustellen, daß alle Dateisysteme und Platteneinheiten gespiegelt werden oder in einer RAID-

Konfiguration angeordnet sind. Dadurch werden Funktionsübernahmen aufgrund einer ausgefallenen Platte vermieden und die Stabilität des Clusters erhöht.

HA-NFS

DB2 UDB EEE erfordert ein gemeinsam benutztes Dateisystem, wenn ein Exemplar über mehrere Maschinen konfiguriert wird. Eine typische DB2 UDB EEE-Konfiguration besitzt ein Ausgangsverzeichnis (home), das aus einer Maschine über NFS exportiert wird und auf allen Maschinen, die an dem EEE-Exemplar beteiligt sind, angehängt wird. Bei einer Konfiguration zur gegenseitigen Übernahme ist DB2 UDB EEE davon abhängig, daß HA-NFS ein gemeinsam benutztes, hochverfügbares Dateisystem bereitstellt. Einer der logischen Hosts exportiert ein Dateisystem durch HA-NFS, und jede Maschine im Cluster hängt das Dateisystem als Ausgangsverzeichnis (home) des EEE-Exemplars an. Weitere Informationen zu HA-NFS finden Sie in der Sun Cluster-Dokumentation.

Die Dienstprogramme cconsole und ctelnet

Zwei nützliche Dienstprogramme, die mit SC2.2 geliefert werden, sind `cconsole` und `ctelnet`. Diese Dienstprogramme können dazu verwendet werden, einen einzelnen Befehl an mehrere Maschinen in einem Cluster gleichzeitig abzusetzen. Wenn eine Konfigurationsdatei mit Hilfe dieser Dienstprogramme editiert wird, ist sichergestellt, daß sie auf allen Maschinen im Cluster gleich bleiben wird. Diese Dienstprogramme können außerdem dazu genutzt werden, Software in exakt der gleichen Weise auf jeder Maschine zu installieren. Weitere Informationen zu diesen Dienstprogrammen finden Sie in der Sun Cluster-Dokumentation.

Campus-Clusterung und kontinentale Clusterung

Ein Cluster wird als *Campus-Cluster* bezeichnet, wenn sich die zugehörigen Maschinen nicht im selben Gebäude befinden. Ein Campus-Cluster ist nützlich, um das Gebäude an sich als einzelnen Fehlerpunkt auszuschließen. Wenn sich zum Beispiel die Maschinen im Cluster alle im selben Gebäude befinden und das Gebäude abbrennt, ist der gesamte Cluster betroffen. Befinden sich die Maschinen hingegen in verschiedenen Gebäuden und ein Gebäude brennt ab, bleibt der Cluster bestehen.

Ein *kontinentaler Cluster* ist ein Cluster, dessen Maschinen über verschiedene Städte verteilt sind. In diesem Fall liegt das Ziel darin, die geografische Region als einzelnen Fehlerpunkt auszuschließen. Diese Art von Cluster bietet einen Schutz gegen Katastrophen wie Erdbeben und Tsunami-Wellen.

Zur Zeit kann Sun Cluster Maschinen unterstützen, die bis zu 10 km (ca. 6 Meilen) auseinander liegen. Dies macht ein Campus-Clusterung zu einer funktionsfähig Option für Benutzer, die Hochgeschwindigkeitsverbindungen zwischen zwei verschiedenen Standorten benötigen. Ein Cluster erfordert zwei private wechselseitige Verbindungen und eine Anzahl von Glasfaserkabeln für

die gemeinsam genutzten Platten. Die Kosten von Hochgeschwindigkeitsverbindungen zwischen zwei Standorten könnten die Vorteile relativieren.

Häufige Probleme

Die SC2.2-Software verwendet Cluster Configuration Database bzw. CCD(4), um eine einziges clusterweites Repository für die Clusterkonfiguration bereitzustellen. CCD besitzt eine private API und ist unter dem Verzeichnis `/etc/opt/SUNWcluster/conf` gespeichert. In seltenen Fällen kann CCD die Synchronisierung verlieren und muß dann eventuell repariert werden. Die beste Methode zur Reparatur von CCD in diesem Fall ist die Wiederherstellung von CCD von einer Sicherungskopie.

Zur Sicherung von CCD fahren Sie die Clustersoftware auf allen Maschinen im Cluster herunter, packen das Verzeichnis `/etc/opt/SUNWcluster/conf` mit "tar" zusammen und speichern die tar-Datei an einem sicheren Ort. Wenn die Clustersoftware bei der Erstellung der Sicherung nicht heruntergefahren ist, haben Sie vielleicht Schwierigkeiten bei der Wiederherstellung von CCD. Stellen Sie sicher, daß die Sicherungskopie auf aktuellem Stand gehalten wird, indem Sie die Sicherung nach jeder Änderung der Clusterkonfiguration aktualisieren. Zur Wiederherstellung von CCD fahren Sie die Clustersoftware auf allen Maschinen im Cluster herunter, versetzen (move) das Verzeichnis `conf` nach `conf.old` und entpacken (untar) die Sicherungskopie. Der Cluster kann anschließend mit der wiederhergestellten CCD-Datenbank gestartet werden.

Überlegungen zu DB2

Dieser Abschnitt behandelt die folgenden Themen:

- „Anwendungen, die eine Verbindung zu einem HA-Exemplar herstellen“
- „Plattenlayout für EE- und EEE-Exemplare“ auf Seite 341
- „Ausgangsverzeichnis für EE- und EEE-Exemplare“ auf Seite 343
- „Logische Hosts und DB2 UDB EEE“ auf Seite 344
- „Speicherposition und Optionen der DB2-Installation“ auf Seite 345
- „Konfigurationsparameter für die Datenbank und den Datenbankmanager“ auf Seite 345
- „Wiederherstellung nach einem Systemabsturz“ auf Seite 346
- „Hohe Verfügbarkeit durch Datenreplikation“ auf Seite 346

Anwendungen, die eine Verbindung zu einem HA-Exemplar herstellen

Anwendungen, die von einem hochverfügbaren DB2-Exemplar abhängig sind, müssen in der Lage sein, im Falle einer Funktionsübernahme eine neue Verbindung herzustellen. Da der Host-Name und die IP-Adresse eines logischen Host gleich bleiben, muß keine Verbindung zu einem anderen Host-Namen hergestellt oder die Datenbank erneut katalogisiert werden.

Betrachten Sie das Beispiel eines Clusters mit zwei Maschinen und einem Exemplar von DB2 Universal Database Enterprise Edition (EE). Das EE-Exemplar befindet sich im Normalfall auf einer der Maschinen im Cluster. Clients des HA-Exemplars stellen die Verbindung zur logischen IP-Adresse (oder zum Host-Namen) des logischen Hosts her, dem das HA-Exemplar zugeordnet ist.

Aus Sicht eines HA-Clients gibt es zwei Arten von Funktionsübernahme. Eine Art tritt auf, wenn die Maschine abstürzt, auf der das HA-Exemplar aktiv ist. Die andere Art tritt auf, wenn das HA-Exemplar eine Möglichkeit erhält, ordnungsgemäß herunterzufahren.

Wenn eine Maschine abstürzt und das HA-Exemplar außer Betrieb setzt, werden sowohl die vorhandenen Verbindungen als auch neue Verbindungen zur Datenbank blockiert. Die Verbindungen werden blockiert, weil es keine Maschinen im Netzwerk mit der IP-Adresse gibt, die die Clients für die Datenbank verwendeten. Wenn die Datenbank ordnungsgemäß heruntergefahren wird, unterbricht ein Befehl `db2stop force` vorhandene Verbindungen zur Datenbank und es wird eine Fehlernachricht zurückgegeben.

Während der Funktionsübernahme ist die der Datenbank zugeordnete logische IP-Adresse offline, entweder weil sie von der SC2.2-Software offline genommen wurde oder weil die Maschine, die den logischen Host beherbergt, abgestürzt ist. Zu diesem werden alle neuen Verbindungen zur Datenbank für eine kurze Zeit blockiert.

Die der Datenbank zugeordnete logische IP-Adresse wird schließlich auf einer anderen Maschine wieder funktionsfähig gemacht, bevor DB2 gestartet wird. In diesem Stadium wird eine Verbindung zur Datenbank nicht blockiert, jedoch empfängt sie einen Übertragungsfehler, weil DB2 im System noch nicht wieder gestartet wurde. DB2-Clients, die immer noch mit der Datenbank verbunden sind, erhalten nun ebenfalls Übertragungsfehler. Obwohl die Clients weiterhin annehmen, daß sie verbunden sind, sind der Maschine, die die logische IP-Adresse übernommen hat, keine vorhandenen Verbindungen bekannt. Die Verbindungen werden einfach zurückgesetzt, und der DB2-Client empfängt einen Übertragungsfehler. Nach einer kurzen Zeit wird DB2 auf der Maschine gestartet, so daß eine Verbindung zur Datenbank erfolgreich hergestellt werden kann. An diesem Punkt kann die Datenbank inkonsistent sein, und die Clients müssen möglicherweise warten, bis die Konsistenz wiederhergestellt ist.

Bei der Entwicklung einer Anwendung für eine HA-Umgebung ist es nicht nötig, speziellen Code für die Phasen zu schreiben, in denen die Datenbankverbindungen blockiert werden. Die Verbindungen sind nur kurze Zeit blockiert, während die Sun Cluster-Software die logische IP-Adresse versetzt. Jeder Datenbankservice, der unter Sun Cluster ausgeführt wird, erfährt die

gleichen blockierten Verbindungen während dieser Phase. Unabhängig davon, wie die Datenbank außer Betrieb gesetzt wird, empfangen die Clients eine Fehlernachricht und müssen versuchen, die Verbindung wiederherzustellen, bis sie erfolgreich sind. Aus der Perspektive des Clients sieht es so aus, als wäre das HA-Exemplar ausgefallen und auf derselben Maschine wieder verfügbar gemacht worden. Bei einer kontrollierten Funktionsübernahme erscheint es dem Client so, als ob er zwangsweise getrennt wurde und später eine neue Verbindung zur Datenbank auf derselben Maschine herstellen könnte. Bei einer unkontrollierten Funktionsübernahme hat der Client den Eindruck, daß der Datenbank-Server abgestürzt ist und auf derselben Maschine in kurzer Zeit wieder verfügbar gemacht wurde.

Plattenlayout für EE- und EEE-Exemplare

DB2 erwartet, daß die erforderlichen Platteneinheiten bzw. Dateisysteme auf jeder Maschine im Cluster gleich erscheinen. Um dies zu gewährleisten, sollten die erforderlichen Platten oder Dateisysteme so konfiguriert werden, daß sie dem logischen Host, dem das HA-Exemplar zugeordnet ist, folgen und auf jeder Maschine im Cluster die gleichen Pfadnamen besitzen.

Sowohl DMS- als auch SMS-Tabellenbereiche werden in einer HA-Umgebung unterstützt. Einheitenbehälter für DMS-Tabellenbereiche müssen vom Datenträgermanager erstellte unformatierte Einheiten (Raw Devices) verwenden, die entweder gespiegelt werden oder in einer RAID-Konfiguration eingerichtet sind. Reguläre Platteneinheiten (Disk devices), wie zum Beispiel `/dev/rdisk/c20t0d0s0`, sollten aus folgenden Gründen nicht eingesetzt werden:

- Sie erhöhen die Wahrscheinlichkeit, daß in Einheiten (device) aus mehr als einer Maschine gleichzeitig geschrieben werden könnte.
- Die Controllernummer könnte auf einer anderen Maschine anders lauten.

Wenn DB2 in dieser Situation von einer anderen Maschine übernommen wird, sehen die erforderlichen Platteneinheiten nicht genauso aus, wie auf der anderen Maschine, und DB2 wird nicht gestartet. Dateibehälter für DMS-Tabellenbereiche und Behälter für SMS-Tabellenbereiche müssen sich auf angehängten (mounted) Dateisystemen befinden. Die Dateisysteme für einen logischen Host werden automatisch angehängt, wenn sie in die Datei `vfstab` für den logischen Host aufgenommen wurden.

Die Datei `vfstab` für einen logischen Host befindet sich in folgendem Pfad:

```
/etc/opt/SUNWcluster/conf/hanfs/vfstab.<logischer_host>
```

Dabei ist *logischer_host* der Name des logischen Hosts, dem die Datei `vfstab` zugeordnet ist.

Jeder logische Host besitzt eine eigene Datei `vfstab`, die Dateisysteme enthält, die anzuhängen sind, nachdem die Plattengruppen für den logischen Host auf die aktuelle Maschine verlegt wurden, jedoch bevor die HA-Services gestartet

werden. Die Sun Cluster-Software versucht, jedes ordnungsgemäß definierte Dateisystem nach der Ausführung von **fsck** (Dateisystemprüfung) anzuhängen, um den einwandfreien Zustand des Dateisystems sicherzustellen. Wenn **fsck** fehlschlägt, wird das Dateisystem nicht angehängt und eine Fehlermeldung protokolliert.

Anmerkung: Wenn ein Prozeß eine geöffnete Datei besitzt oder sich das zugehörige aktuelle Arbeitsverzeichnis unter einem Mount-Punkt befindet, schlägt das Anhängen (Mount) fehl. Zur Vermeidung dieses Fehlers müssen Sie sicherstellen, daß keine Prozesse unter Mount-Punkten verbleiben, die in der Datei `vfstab` des logischen Hosts enthalten sind.

Als Dateisystemlayout für ein EEE-Exemplar kann bei Verwendung von SMS-Tabellenbereichen eine beliebige Konvention verwendet werden. Das folgende Beispiel zeigt die Konvention, die vom Dienstprogramm `hadb2_setup` verwendet wird:

```
scadmin@crack1e(190)# pwd
/export/ha_home/db2eee/db2eee
scadmin@crack1e(191)# ls -l
total 18
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0000 -> /log0/disks/db2eee/NODE0000
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0001 -> /log0/disks/db2eee/NODE0001
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0002 -> /log0/disks/db2eee/NODE0002
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0003 -> /log0/disks/db2eee/NODE0003
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0004 -> /log0/disks/db2eee/NODE0004
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0005 -> /log1/disks/db2eee/NODE0005
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0006 -> /log1/disks/db2eee/NODE0006
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0007 -> /log1/disks/db2eee/NODE0007
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0008 -> /log1/disks/db2eee/NODE0008
scadmin@crack1e(192)#
```

Der Exemplareigner ist `db2eee` und das Standarddatenbankverzeichnis für das Exemplar `db2eee` ist `/export/ha_home/db2eee`. Der logische Host `log0` beherbergt die Datenbankpartitionen 0, 1, 2,3 und 4, während der logische Host `log1` die Datenbankpartitionen 5, 6, 7 und 8 beherbergt.

Für jede Datenbankpartition gibt es ein entsprechendes Verzeichnis `NODExxxx`. Die Knotenverzeichnisse für die Datenbankpartitionen zeigen auf ein Verzeichnis unter dem Dateisystem des zugeordneten logischen Hosts.

Bei der Auswahl einer Pfadkonvention müssen Sie folgendes sicherstellen:

1. Die Platten für das Dateisystem befinden sich in einer Plattengruppe des logischen Hosts, der für die Datenbankpartitionen zuständig ist, die sie benötigen.
2. Die Dateisysteme, die Behälter enthalten, werden über die Datei `vfstab` des logischen Hosts angehängt.

Ausgangsverzeichnis für EE- und EEE-Exemplare

Für ein EE-Exemplar sollte das Ausgangsverzeichnis (home) ein Dateisystem sein, das in der Datei `vfstab` für einen logischen Host definiert ist. Dieses Verzeichnis wird verfügbar, bevor DB2 gestartet wird, und wird mit DB2 versetzt, wenn der logische Host im Cluster versetzt wird. Jede Maschine verfügt über eine eigene Kopie der Datei `vfstab`, und es sollte sorgfältig darauf geachtet werden, daß sie auf jeder Maschine im Cluster den gleichen Inhalt hat. Das folgende Beispiel zeigt das Ausgangsverzeichnis für ein EE-Exemplar:

```
/log0/home/db2ee
```

Dabei ist `/log0` das Dateisystem für den logischen Host `log0` und `db2ee` der Name des DB2-Exemplars. Dieser Ausgangsverzeichnispfad sollte in die Datei `/etc/passwd` auf jeder Maschine im Cluster eingefügt werden, die als Host für das Exemplar "db2ee" fungieren kann.

Für ein EEE-Exemplar kann das Ausgangsverzeichnis auf zwei Arten eingerichtet werden. Bei einer Konfiguration im Bereitschaftsmodus (Hot Standby) kann das Ausgangsverzeichnis auf die gleiche Art wie für ein EE-Exemplar eingerichtet werden. Bei einer Konfiguration zur gegenseitigen Übernahme (Mutual Takeover) muß HA-NFS für das Ausgangsverzeichnis verwendet und vor dem Einrichten des EEE-Exemplars ordnungsgemäß konfiguriert werden.

Eine der Maschinen im Cluster muß mit Hilfe der Datei `dfstab` für einen ausgewählten logischen Host das Dateisystem für das EEE-Exemplar exportieren. Die Datei `dfstab` enthält Dateisysteme, die durch NFS exportiert werden sollten, wenn eine Maschine einen logischen Host beherbergt. Jede Maschine verfügt über eine eigene Kopie der Datei `dfstab`, und es sollte sorgfältig darauf geachtet werden, daß sie auf jeder Maschine im Cluster den gleichen Inhalt hat.

Informationen für das HA-NFS-Dateisystem werden in der Datei `hadb2tab` (durch das Programm `hadb2_setup`) abgelegt. Wenn ein HA-Agent die Informationen für das Exemplar liest, hängt er das HA-NFS-Dateisystem für das Exemplar automatisch an (siehe „Die Datei `hadb2tab`“ auf Seite 347).

Der Mount-Punkt für das HA-NFS-Dateisystem ist in der Regel `/export/ha_home`. Auf jeder Maschine im Cluster wäre dies das NFS, das vom logischen Host angehängt wird, der das HA-NFS-Verzeichnis exportiert. Das Ausgangsverzeichnis des EEE-Exemplareigners wird unter diesem Verzeichnis angelegt und hat den folgenden Namen:

```
/export/ha_home/<exemplar>
```

Dabei ist *exemplar* der Name des Exemplareigners.

Es wäre möglich, ein Ausgangsverzeichnis für ein Exemplar auf jeder Maschine anzulegen, um ein Anhängen oder Abhängen dieses Verzeichnisses zu vermeiden. Dieses Verfahren erfordert einen zusätzlichen administrativen Aufwand, um sicherzustellen, daß die Ausgangsverzeichnisse auf allen Maschinen identisch bleiben. Wenn dies nicht gewährleistet wird, kann DB2 eventuell nicht ordnungsgemäß gestartet werden oder wird mit einer anderen Konfiguration gestartet. Dies ist *keine* unterstützte Konfiguration.

Logische Hosts und DB2 UDB EEE

Ein logischer Host wird in der Regel dazu ausgewählt, als Host für eine oder mehrere Datenbankpartitionen zu fungieren sowie dazu das HA-NFS-Dateisystem zu exportieren. Wenn es zum Beispiel vier Datenbankpartitionen und zwei Maschinen im Cluster gäbe, sollte es einen logischen Host für jede Maschine geben (siehe Abb. 61 auf Seite 345). Ein logischer Host würde als Host zweier Datenbankpartitionen fungieren und das HA-NFS-Dateisystem exportieren, während der andere logische Host die verbleibenden zwei Datenbankpartitionen beherbergt.

Standardmäßig ordnet ein DB2 UDB EEE-Exemplar genügend Ressourcen zum erfolgreichen Hinzufügen von bis zu zwei Datenbankpartitionen einer Maschine zu, die bereits eine oder mehrere im Betrieb befindliche Datenbankpartitionen für dieses Exemplar besitzt. Wenn es beispielsweise vier Datenbankpartitionen für ein einziges Exemplar in einem Cluster gibt, kann es nur dann zu Problemen kommen, wenn es eine Datenbankpartition pro logischen Host gibt oder ein logischer Host als Host von drei Datenbankenpartitionen fungiert. In beiden Fällen ist es möglich, daß drei Datenbankpartitionen von einer Maschine übernommen werden, die bereits für eine Datenbankpartition desselben Exemplars zuständig ist.

Die Registrierungsvariable `DB2_NUM_FAILOVER_NODES` kann dazu verwendet werden, die Ressourcenmenge zu erhöhen, die für zu übernehmende Datenbankpartitionen reserviert wird.

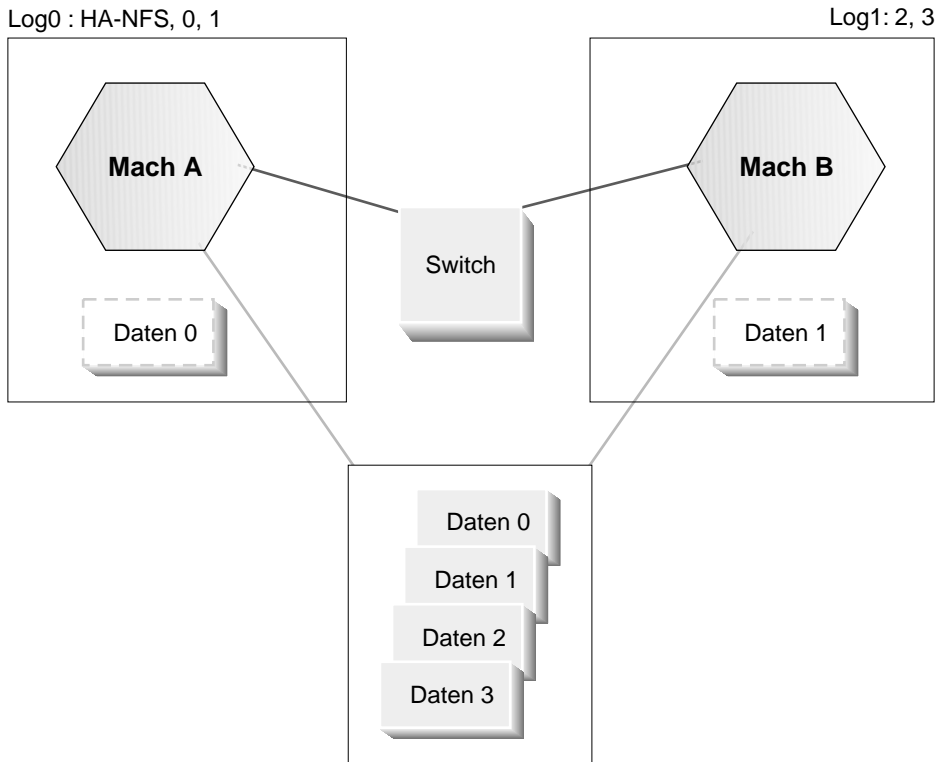


Abbildung 61. Ein logischer Host für jede Maschine

Speicherposition und Optionen der DB2-Installation

Das Dateisystem, in dem DB2 installiert wird, sollte gespiegelt werden oder zumindest in einer RAID-Konfiguration angelegt werden. Wenn DB2 auf normalen Platten installiert wird, ist ein Plattenausfall wahrscheinlicher. Die sich daraus ergebende Funktionsübernahme gilt als vermeidbar und verringert die Stabilität des Clusters.

DB2 kann nicht auf Platten in einer Plattengruppe für einen logischen Host installiert werden, weil der HA-Agent ständig Zugriff auf die DB2-Bibliotheken benötigt. Wenn die HA-Agenten keinen Zugriff auf die DB2-Bibliotheken haben, werden sie mit einem Fehler beendet. DB2 muß auf jeder Maschine im Cluster normal installiert werden.

Konfigurationsparameter für die Datenbank und den Datenbankmanager

Die Konfigurationsparameter des Datenbankmanagers können nach einer Funktionsübernahme und vor dem Starten von DB2 mit Hilfe der Prozedur `pre_db2start` (siehe „Benutzerprozeduren“ auf Seite 350) geändert werden. Diese ausführbare Prozedur (Skript) wird (falls vorhanden) unter dem Ver-

zeichnis `sqllib/ha` des Ausgangsverzeichnisses des Exemplareigners ausgeführt. Wie der Name verrät, wird sie unmittelbar vor **db2start** ausgeführt. Die gleichen Parameter, die an die Steuermethoden übergeben werden, werden auch an die Prozedur `pre_db2start` übergeben, wenn das Exemplar kein `EEE`-Exemplar ist. Für ein `EEE`-Exemplar wird der Prozedur `pre_db2start` außerdem die Knotennummer für den Befehl **db2start** übergeben.

Wiederherstellung nach einem Systemabsturz

Die Wiederherstellung nach einem Systemabsturz in einer HA-Umgebung erfolgt auf dieselbe Weise wie in einer regulären Umgebung. Selbst wenn das HA-Exemplar auf einer anderen Maschine als der Absturzmaschine wieder gestartet wird, sehen die Dateien und Platteneinheiten für das Exemplar gleich aus und die Aktionen, die zur Wiederherstellung der Datenbank durchzuführen sind, unterscheiden sich ebenfalls nicht. Weitere Informationen zur Wiederherstellung nach einem Systemabsturz finden Sie in "Wiederherstellen einer Datenbank" im Handbuch *Systemverwaltung: Implementierung*.

Obwohl eine Datenbank manuell (oder durch eine der Benutzerprozeduren) erneut gestartet werden kann, wird empfohlen, den Konfigurationsparameter *autorestart* der Datenbank, insbesondere für ein `EEE`-Exemplar, auf den Wert `ON` zu setzen. Dadurch wird der Zeitraum minimiert, den die Datenbank in einem inkonsistenten Zustand ist.

Hohe Verfügbarkeit durch Datenreplikation

Die Verfügbarkeit von Daten kann auch durch Replikation verbessert werden. Durch das Replizieren von Daten zwischen zwei Servern läßt sich eine Form hoher Verfügbarkeit erreichen. Wenn einer der Server ausfällt, kann der andere Server die Funktion übernehmen und den Datenbankservice weiterhin bereitstellen.

Da die Replikation jedoch asynchron erfolgt, wurden von einem Server, der ausfällt, möglicherweise einige Änderungen noch nicht auf den anderen Server weitergegeben.

Der DB2-Agent für hohe Verfügbarkeit

Der DB2-Agent für hohe Verfügbarkeit (HA-Agent) fungiert als Vermittler zwischen DB2 und SC2.x. Er bietet der Sun Cluster 2.2-Software eine Möglichkeit, DB2 in einer Clusterumgebung zu steuern, ohne sich mit den Einzelheiten von DB2 auskennen zu müssen. Es gibt einen Agenten für `EE`- und `EEE`-Exemplare. Der Agent unterstützt sowohl administrative Exemplare als auch Datenbankexemplare.

Registrieren des Service `hadb2`

Zur Arbeit mit SC2.2 muß der HA-Agent von DB2 registriert werden. Durch die Registrierung eines Datenbankservice wird SC2.2 mitgeteilt, welche Steuermethoden verfügbar sind und in welchem Verzeichnis sie sich befinden. Mit

dem HA-Agenten wird eine spezielle Prozedur namens `hadb2_reg` geliefert, die den Service `hadb2` für EE- und EEE-Exemplare registrieren kann. Die Prozedur `hadb2_reg` muß für den gesamten Cluster nur einmal ausgeführt werden.

Obwohl es nur einen Satz an Steuermethoden für den DB2-HA-Agenten gibt, hängt die Art, wie die Methoden registriert werden davon ab, ob ein EEE-Exemplar in einer Konfiguration zur gegenseitigen Übernahme verwendet wird oder nicht. Für ein EE-Exemplar bzw. ein EEE-Exemplar in einer Konfiguration im Bereitschaftsmodus (Hot Standby) wird HA-NFS nicht verwendet. Daher wird der Schalter `-d nfs`, der der SC2.2-Software mitteilt, daß der Service `hadb2` von HA-NFS abhängig ist, nicht benötigt.

Der tatsächliche Befehl, der von `hadb2_reg` zur Registrierung der Steuermethoden für ein EEE-Exemplar der DB2 Version 7.1 verwendet wird, sieht wie folgt aus:

```
hareg -r hadb2 -b /opt/IBMDb2/V7.1/ha -m
START=hadb2_start,START_NET=hadb2_startnet,STOP_NET=hadb2_stopnet,
FM_START=hadb2_fmstart,FM_STOP=hadb2_fmstop
-t START_NET=$TIMEOUT,STOP_NET=$TIMEOUT -d nfs
```

Der Schalter `-b` weist SC2.x an, alle Steuermethoden im Verzeichnis `opt/IBMDb2/V7.1/ha` zu suchen. Der Schalter `-m` definiert die eigentlichen Steuermethoden für den Service `hadb2`. Der Schalter `-t` definiert das Zeitlimit für die Steuermethoden `START_NET` und `STOP_NET`. Eine detaillierte Beschreibung der einzelnen Steuermethoden finden Sie in der Sun Cluster-Dokumentation.

Die Prozedur `hadb2_unreg` kann dazu genutzt werden, die Registrierung des Service `hadb2` zu löschen, und muß, wie `hadb2_reg`, nur einmal für den Cluster ausgeführt werden.

Die Datei `hadb2tab`

Die Datei `hadb2tab` ist die Hauptkonfigurationsdatei für den DB2-HA-Agenten. Jede Steuermethode greift auf diese Datei zurück, um zu ermitteln, welche Exemplare hochverfügbar sind. Die Datei `hadb2tab` befindet sich für DB2 UDB Version 7.1 unter dem Verzeichnis `/var/db2/v71/`. Die Datei unterstützt mehrere Exemplare, und jede Nichtkommentarzeile stellt ein anderes HA-Exemplar dar. Das folgende Beispiel zeigt eine Datei `hadb2tab`:

```
<scadmin@thrash(203)# cat hadb2tab
EEE DATA db2eee jolt ON /export/ha_home /log0/home #Added by DB2 HA software
EE ADMIN db2ee log1 ON - - #Added by DB2 HA software
```

Das erste Feld gibt dem DB2-HA-Agenten an, ob es sich bei dem Exemplar um ein EE- oder ein EEE-Exemplar handelt. Das zweite Feld gibt an, ob das Exemplar ein Datenexemplar oder ein administratives Exemplar ist. Das dritte Feld enthält den Benutzernamen für das HA-Exemplar. Das vierte Feld ist der

logische Host oder der HA-NFS-Host für das Exemplar, je nachdem, ob es sich um ein EE- oder ein EEE-Exemplar handelt. Das fünfte Feld gibt an, ob die Fehlerüberwachung für das Exemplar aktiviert ist (ON) oder nicht (OFF). Die beiden letzten Felder geben den lokalen Mount-Punkt bzw. das ferne HA-NFS-Verzeichnis an. Diese Felder sollten mit dem Wert "-" (Bindestrich) gefüllt werden, wenn sie nicht verwendet werden, und sollten außerdem nur bei einer EEE-Konfiguration zur gegenseitigen Übernahme verwendet werden. Kommentare sind in der Datei hadb2tab zulässig, wenn die Informationen in der Zeile vor einem "#" -Zeichen entweder die Länge null haben oder eine gültige Definition eines Exemplars darstellen.

Steuermethoden

Steuermethoden für SC2.2-Agenten können eine Gruppe von Prozeduren (Skripten) oder Programmen sein. Der Agent für DB2 auf Solaris besteht aus einer Gruppe von Programmen, die folgende Methoden enthält:

START_NET

hadb2_startnet dient zum Starten von DB2.

STOP_NET

hadb2_stopnet dient zum Stoppen von DB2.

FM_START

hadb2_fmstart dient zum Starten des Fehlermonitors für DB2.

FM_STOP

hadb2_fmstop dient zum Stoppen des Fehlermonitors für DB2.

Weitere Informationen zu diesen Steuermethoden finden Sie in der Sun Cluster-Dokumentation.

Für EE-Exemplare wird der logische Host, der dem Exemplar zugeordnet ist, direkt in der Datei hadb2tab definiert. Für EEE-Exemplare muß die Steuer-methode jedoch außerdem in folgende Datei schauen:

```
~<exemplar>/sql1lib/ha/hadb2-eee.cfg
```

Dabei ist ~<instance> das Ausgangsverzeichnis des Exemplareigners. Diese Datei enthält eine Zeile für jede Datenbankpartition und dient zur Zuordnung von Datenbankpartitionen zu logischen Hosts. Ein Beispiel für eine gültige Datei hadb2-eee.cfg sieht wie folgt aus:

```
crack1e % cat hadb2-eee.cfg
NODE:log0 0
NODE:log0 1
NODE:log1 2
NODE:log1 3
```

Das Exemplar oder die Datenbankpartitionen folgen dem entsprechenden logischen Host von Maschine zu Maschine im Cluster. Der logische Host kann auf eine beliebige Maschine im Cluster versetzt werden, die von der

zugrundeliegenden Hardware und SC2.2 unterstützt wird. Wenn die Konfiguration ordnungsgemäß eingerichtet ist, unterstützt DB2 jede Topologie, die von der SC2.2-Software unterstützt wird.

Nach dem Lesen aller Informationen für ein Exemplar weiß die Steuer- methode, welche logischen Hosts dem Exemplar zugeordnet sind. Nach der Syntaxanalyse der Befehlszeilenparameter weiß die Steuer- methode außerdem, welche logischen Hosts sich auf der aktuellen Maschine befinden und welche nicht.

Die folgende Tabelle zeigt die Aktionen, die unternommen werden, je nach- dem, welche Steuer- methode ausgeführt wird, und abhängig davon, ob die der Datenbankpartition oder dem Exemplar zugeordneten logischen Hosts auf der aktuellen Maschine aktiv sind.

Steuer- methode	Zugeordnete logische Host(s) auf Maschine	Zugeordnete logische Host(s) nicht auf Maschine
START_NET	Starten des DB2-Exemplars oder der Datenbankpartitionen	Keine Aktion
STOP_NET	Keine Aktion	Stoppen des DB2-Exemplars oder der Datenbankpartitionen
FM_START	Starten des Fehlermonitors für das Exemplar	Keine Aktion
FM_STOP	Keine Aktion	Stoppen des Fehlermonitors für das Exemplar

Die Steuer- methoden, die START- Aktionen ausführen, beziehen sich nur auf die logischen Hosts, die zur Zeit auf der Maschine aktiv sind, und die Steuer- methoden, die STOP- Aktionen ausführen, beziehen sich nur auf logische Hosts, nicht zur Zeit nicht auf der Maschine aktiv sind.

Die Steuer- methoden müssen darüber hinaus das HA-NFS- Verzeichnis auf spezielle Weise anhängen, wenn HA-NFS verwendet wird. Wenn der lokale Mount- Punkt und das Verzeichnis für HA-NFS nicht als "-" (Bindestrich) definiert sind, führt die Steuer- methode einen Befehl `statvfs(2)` auf dem lokalen Mount- Punkt aus. Wenn der Dateisystemtyp für den lokalen Mount- Punkt nicht `nfs` ist, versucht der Agent, das Dateisystem mit Hilfe der Infor- mationen aus der Zeile in der Datei `hadb2tab` anzuhängen. Wenn der Mount- Punkt und das Verzeichnis für HA-NFS als "-" (Bindestrich) definiert sind, ist die Datei `vfstab` für den entsprechenden logischen Host erforderlich, um das Dateisystem anzuhängen, das das Ausgangsverzeichnis für das Exemplar enthält. Der lokale Mount- Punkt und das ferne Verzeichnis für HA-NFS soll-

ten nur für EE- und EEE-Konfigurationen im Bereitschaftsmodus (Hot Standby) als "-" (Bindestrich) definiert werden.

Benutzerprozeduren

Diese Prozeduren (Skripte) werden über die Steuermethoden ausgeführt, um zusätzliche Funktionalität bereitzustellen. Sie erhalten die gleichen Befehlszeilenparameter wie die Steuermethoden und werden vom System- bzw. vom Datenbankadministrator geschrieben.

Wenn ein Programm innerhalb einer Prozedur ausgeführt werden muß, die nicht im Hintergrund ausgeführt wird, kann es sinnvoll sein, das Programm mit `nohup(1)` in den Hintergrund zu verlegen. Das Programm `nohup` schützt das ausgeführte Programm vor dem Signal `SIGHUP` (oder `Hangup`), d. h. vor einem nichtprogrammierten Stopp. Ohne `nohup` kann ein Programm, das über eine Prozedur im Hintergrund ausgeführt wird, infolge eines `SIGHUP`-Signals vorzeitig abgebrochen werden, wenn die Prozedur die Verarbeitung beendet hat.

Die Steuermethoden führen die folgenden Prozeduren aus:

- `/var/db2/v61/failover`
- `~<exemplar>/sqllib/ha/pre_db2start`
- `~<exemplar>/sqllib/ha/post_db2start`
- `~<exemplar>%s/sqllib/ha/post_failover`
- `~<exemplar>/sqllib/ha/pre_db2stop`
- `~<exemplar>/sqllib/ha/fm_warning`

Dabei ist `~exemplar` das Ausgangsverzeichnis des HA-Exemplars.

Mit Ausnahme der Prozedur `fm_warning` wird jede Benutzerprozedur mit den gleichen Befehlszeilenparametern ausgeführt wie die Methode, die sie aufruft. Bei Verwendung von EEE-Exemplaren wird ferner die Nummer der Datenbankpartition (als letzter Parameter) an die Benutzerprozedur übergeben.

Die Prozedur `/var/db2/v71/failover` wird zu Beginn der Methode `START_NET` aufgerufen und im Hintergrund ausgeführt. Eine solche Prozedur kann beispielsweise dazu genutzt werden, das Unterstützungspersonal per E-Mail über eine Funktionsübernahme in Kenntnis zu setzen. Das folgende Beispiel zeigt eine Failover-Prozedur:

```
#!/bin/ksh
```

```
# Nachricht über Funktionsübernahme an Personal per E-mail oder Pager
```

```
echo "Failover occurred on machine 'hostname':Running $0!" | /bin/mail admin@sphere.toro
```

Zur erfolgreichen Benachrichtigung durch eine Prozedur per E-Mail muß `sendmail(1m)` auf dem System ordnungsgemäß konfiguriert sein.

Wie der Name andeutet, wird die Prozedur `pre_db2start` unmittelbar vor dem Aufrufen von **db2start** ausgeführt. Diese Prozedur kann zu solchen Zwecken wie das Ändern von Konfigurationsparametern des Datenbankmanagers eingesetzt werden. Ihr werden maximal 20 Sekunden für die Ausführung eingeräumt. Für EEE-Exemplare wird diese Prozedur vor dem Aufrufen von **db2start** auf jeder Datenbankpartition ausgeführt. Diese Prozedur ist nur für Datenexemplare, nicht für administrative Exemplare von Bedeutung.

Analog wird die Prozedur `post_db2start` unmittelbar *nach* dem Aufrufen von **db2start** ausgeführt. Diese Prozedur kann zu solchen Zwecken wie das erneute Starten von Datenbanken eingesetzt werden. Sie wird im Hintergrund ausgeführt, um zu gewährleisten, daß ihre Ausführungszeit keine störenden Auswirkungen auf andere Exemplare hat. Diese Prozedur ist nur für Datenexemplare, nicht für administrative Exemplare von Bedeutung.

Die Prozedur `post_failover` unter dem Ausgangsverzeichnis des Exemplareigners wird nach der Verarbeitung des Exemplars ausgeführt. Diese Prozedur kann dazu verwendet werden, Client-Anwendungen zu benachrichtigen, daß DB2 nun betriebsbereit ist, Datenbanken zu aktivieren oder Administratoren eine Statusdatei zu senden. Sie wird im Hintergrund ausgeführt, um zu verhindern, daß ihre Ausführungszeit Aktionen an anderen HA-Exemplaren verzögert. Das folgende Beispiel zeigt eine `post_failover`-Prozedur:

```
#!/bin/ksh
#

# Statusdatei an Administrator senden.
mail admin@sphere.torolab.ibm.com </tmp/HA.info.db2eee
```

Sowohl die Methode `START_NET` als auch die Methode `STOP_NET` des DB2-HA-Agenten erstellen nach der Verarbeitung jedes Exemplars eine Statusdatei. Der Name der Statusdatei lautet:

```
/tmp/HA.info.<exemplar>
```

Dabei ist *exemplar* der Benutzername des Exemplareigners. Die Statusdatei enthält den Start- und Stoppbericht für das Exemplar sowie die Zeit, die zur Ausführung der Steuermethode benötigt wurde. Das folgende Beispiel zeigt eine Statusdatei:

```
scadmin@crackle(173)# cat /tmp/HA.info.db2eee
----- Elapsed Time: 00:00:18 -----
----- Elapsed Time: 00:00:00 (HA-NFS) -----

NODE      ACTION      RESULT      TRIES      RC
-----
4         stop        success      3           1064
5         stop        success      1           1064
6         stop        success      2           1064
```

7	stop	success	2	1064
8	stop	success	1	1064

Die Prozedur `pre_db2stop` wird unmittelbar vor dem Aufrufen von **db2stop** ausgeführt. Diese Prozedur kann dazu verwendet werden, Client-Anwendung davon zu benachrichtigen, daß DB2 gestoppt wird. Ihr werden maximal 20 Sekunden für die Ausführung eingeräumt. Diese Prozedur ist nur für Daten-exemplare, nicht für administrative Exemplare von Bedeutung.

Der Fehlermonitor (Fault Monitor) führt ebenfalls eine Benutzerprozedur aus, wenn DB2 aufgrund eines unerwarteten Herunterfahrens erneut gestartet wurde. Diese Prozedur heißt:

```
~<exemplar>/sqllib/ha/fm_warning
```

Die Prozedur `fm_warning` kann dazu genutzt werden, den Systemadministrator zu benachrichtigen, daß DB2 durch den Fehlermonitor erneut gestartet wurde. Der Systemadministrator sollte versuchen, herauszufinden, warum DB2 unerwartet heruntergefahren wurde, und die entsprechenden Maßnahmen ergreifen, um dieses in Zukunft zu vermeiden. Die Prozedur `fm_warning` wird im Hintergrund ausgeführt.

Weitere Überlegungen

Wenn ein HA-Datenbankservice ausgeschaltet wird, werden während einer Funktionsübernahme oder einer Änderung der Clusterkonfiguration nur die STOP-Methoden ausgeführt. Die übrigen Methoden werden nur ausgeführt, wenn der HA-Datenbankservice ordnungsgemäß registriert und aktiviert ist.

Stellen Sie sicher, daß jede Maschine im Cluster über ausreichend Ressourcen verfügt, um alle Datenbankservices auszuführen, für die sie gegebenenfalls zuständig ist. Ressourcen wie CPU-Auslastung, Arbeitsspeicher, Auslagerungs- und Kernel-Parameter müssen bedacht werden, bevor der Cluster als Geschäftssystem eingesetzt wird. Wenn zum Beispiel eine Maschine im Cluster zwei DB2-Exemplare ausführen muß, sind die Kernel-Parameteranforderungen für diese Maschine die Summe der Ressourcen, die für jedes Exemplar erforderlich sind.

Fehlermonitor

Wenn die Fehlerüberwachung aktiviert ist, wird der Fehlermonitor bei einer Änderung der Clusterkonfiguration oder bei einer Funktionsübernahme gestartet. Falls DB2 nicht durch die Prozedur `START_NET` gestartet wird, startet der Fehlermonitor DB2 selbst. Der Fehlermonitor kann erkennen, ob DB2 nicht gestartet wurde oder ob DB2 aus unbekanntem Gründen heruntergefahren wurde. Daher ist es wichtig, DB2 nicht manuell herunterzufahren, wenn der Fehlermonitor aktiviert ist. Der Fehlermonitor sieht dies als unerwartetes Herunterfahren an und startet DB2 erneut. Wenn dies zu häufig

geschieht, findet eine Funktionsübernahme des entsprechenden logischen Hosts durch eine andere Maschine statt.

Wenn die Fehlerüberwachung für ein Exemplar aktiviert ist, besteht die korrekte Art zum manuellen Starten oder Stoppen des Exemplars darin, zunächst die Fehlerüberwachung oder den Service `hadb2` zu inaktivieren. Beide Aktionen können durch den Befehl `hadb2_setup` mit den Schaltern `-f` und `-s` eingeleitet werden (siehe „Der Befehl `hadb2_setup`“ auf Seite 358).

Anmerkung: Verwenden Sie nicht mehr als ein Exemplar für den gleichen logischen Host. Wenn einem logischen Host mehr als ein Exemplar zugeordnet wird, kann das einwandfreie Exemplar zusammen mit dem defekten Exemplar bei einer Funktionsübernahme versetzt werden.

Überlegungen zu EEE

Bei der Entscheidung, welche Datenbankpartitionen einem logischen Host zuzuordnen sind, spielt die Art und Weise, wie die Funktionsübernahme stattfindet, eine wichtige Rolle. Betrachten Sie zum Beispiel einen Cluster mit zwei Maschinen, der mit vier Datenbankpartitionen zwischen zwei Maschinen eingesetzt werden soll, wie in Abb. 62 auf Seite 354 zu sehen ist.

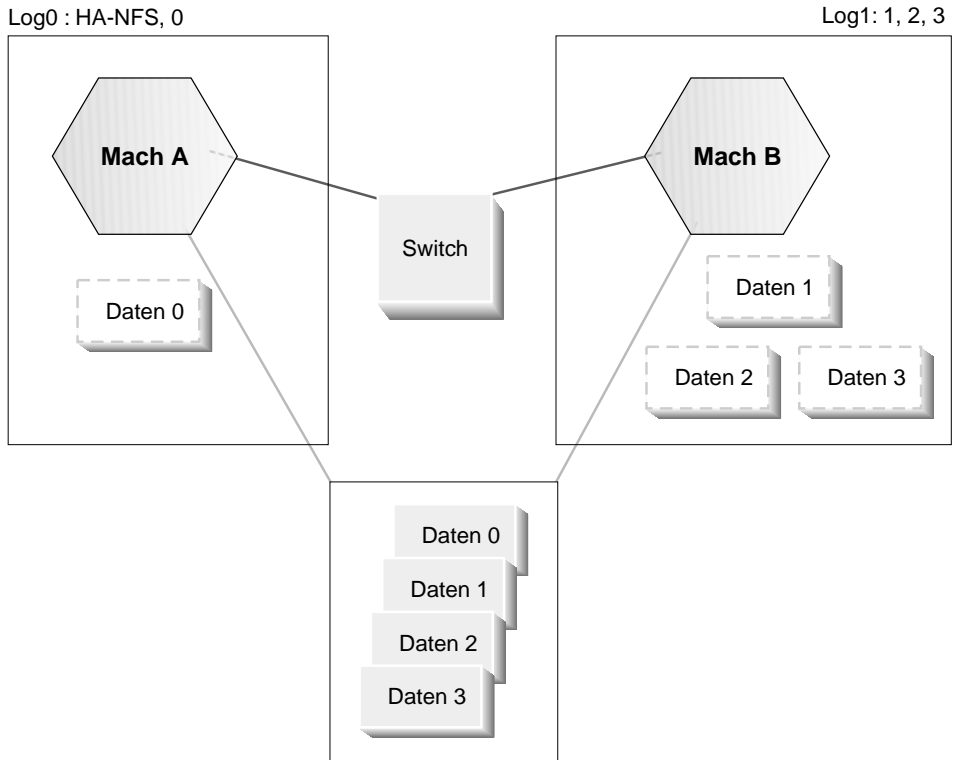


Abbildung 62. Cluster mit zwei Maschinen und vier Datenbankpartitionen

Sie könnten jede Datenbankpartition je einem logischen Host und HA-NFS ebenfalls einem logischen Host zuordnen. In diesem Fall kann ein Problem entstehen, falls alle logischen Hosts von einem System beherbergt werden. Wenn ein System ausfällt, müssen alle logischen Hosts gleichzeitig von diesem System versetzt werden. Leider versetzt die Sun Cluster-Software die logischen Hosts in keiner vorhersagbaren Reihenfolge, so daß es möglich ist, daß ein logischer Host für eine Datenbankpartition versetzt wird, bevor der logische Host mit HA-NFS versetzt wird. In der Regel ist es ein gutes Verfahren, Datenbankpartitionen je nachdem, was gewöhnlich auf einem einzelnen System untergebracht würde, zusammen zu gruppieren. Das bedeutet, daß zwei Datenbankpartitionen, die unter normalen Bedingungen auf einer Maschine untergebracht sind, einem einzigen logischen Host zugeordnet werden sollten.

Die Datei `db2nodes.cfg`, die von einem EEE-Exemplar verwendet wird, wird aktualisiert, um die Maschine anzugeben, auf der sich die Datenbankpartitionen befinden. Wenn zum Beispiel alle Datenbankpartitionen auf einer Maschine namens "crackle" untergebracht sind, sieht die Datei `db2nodes.cfg` in etwa wie folgt aus:

```

scadmin@crackle(193)# cat db2nodes.cfg
0 crackle 0 204.152.65.33
1 crackle 1 204.152.65.33
2 crackle 2 204.152.65.33
3 crackle 3 204.152.65.33
4 crackle 4 204.152.65.33
5 crackle 5 204.152.65.33
6 crackle 6 204.152.65.33
7 crackle 7 204.152.65.33
8 crackle 8 204.152.65.33

```

Wenn einige dieser Datenbankpartitionen auf eine Maschine namens "thrash" versetzt werden, wird die Datei db2nodes.cfg wie folgt aktualisiert:

```

scadmin@crackle(193)# cat db2nodes.cfg
0 crackle 0 204.152.65.33
1 crackle 1 204.152.65.33
2 crackle 2 204.152.65.33
3 crackle 3 204.152.65.33
4 thrash 0 204.152.65.34
5 thrash 1 204.152.65.34
6 thrash 2 204.152.65.34
7 thrash 3 204.152.65.34
8 thrash 4 204.152.65.34

```

Beachten Sie, daß sowohl der Host-Name als auch der Switch-Name geändert werden, um den Maschinennamen "thrash" anzugeben, und daß die Portnummern ebenfalls unterschiedlich sind.

Die Datei HA.config

Wenn sie vorhanden ist, kann die Datei /etc/HA.config eine Reihe von Konfigurationsoptionen, einschließlich der folgenden, enthalten:

```

scadmin@thrash(204)# cat /etc/HA.config
SYSLOG_FACILITY=LOG_LOCAL3
SYSLOG_LPRIORITY=LOG_INFO
SYSLOG_EPRIORITY=LOG_ERR
USE_INTERCONNECT=auto
SWITCH_NAME=204.152.65.18
DEBUG_LEVEL=2
FAILS_PER_HOUR=2
FAILS_PER_DAY=4
FAILS_PER_WEEK=10
FM_FAIL_SEV=soft
DB2START_TIMEOUT=60
DB2STOP_TIMEOUT=500
SCRIPT_USER=bin

```

Anmerkung: Wenn die Datei HA.config nicht vorhanden ist, werden Standardwerte verwendet.

Die Variable SYSLOG_FACILITY stellt die SYSLOG-Einrichtung auf das Protokollieren von Nachrichten und Fehlern ein. Die Variablen SYSLOG_LPRIORI-

TY und SYSLOG_EPRIORITY stellen die SYSLOG-Priorität auf das Protokollieren von Informationsnachrichten bzw. Fehlernachrichten ein.

Einige Änderungen sind zur Aktivierung des SYSLOG-Dämons zur Protokollierung von Informationen aus dem DB2-HA-Agent eventuell erforderlich. Zum Beispiel wird durch Einfügen einer der folgenden beiden Zeilen in der Datei `/etc/syslog.conf` der SYSLOG-Dämon angewiesen, Informationen in eine Protokolldatei zu schreiben.

```
*.notice /var/adm/SC.x
local3.info /var/adm/SC.LOG_LOCAL3
```

Ein Sun Cluster verfügt in der Regel über eine wechselseitige Hochgeschwindigkeitsverbindung. Um die Hochgeschwindigkeitsverbindung mit DB2 zu nutzen, muß `USE_INTERCONNECT` auf den Wert `auto` oder `override` gesetzt werden. Die Einstellung `auto` (Standardeinstellung) arbeitet mit der internen logischen Sun-Netzschnittstelle. Diese Schnittstelle wird auf eine andere physische Schnittstelle übertragen, wenn die erste Schnittstelle ausfällt. Wenn `USE_INTERCONNECT` auf den Wert `override` gesetzt ist, wird der Switch-Name der Variablen `SWITCH_NAME` entnommen. Eine weitere Option besteht darin, `USE_INTERCONNECT` auf `no` zu setzen, wodurch definiert wird, daß die Hochgeschwindigkeitsverbindung nicht zu verwenden ist.

Die Variable `DEBUG_LEVEL` gibt an, wie viele Informationen bei einer Funktionsübernahme zu protokollieren sind. Der Wert ist eine Zahl zwischen 0 und 10, wobei 10 die höchste Debug-Stufe darstellt. Die Informationen werden mit der angegebenen SYSLOG-Priorität und SYSLOG-Einrichtung protokolliert. Wenn Probleme auftreten, setzen Sie die Debug-Stufe auf den höchsten Wert, konfigurieren SYSLOG so, daß die Ausgabe aus den HA-Agenten protokolliert wird und senden die SYSLOG-Ausgabe an den IBM Kundendienst.

Drei der Variablen unterstützen den DB2-Fehlermonitor bei der Entscheidung, wann eine Funktionsübernahme für einen logischen Host durchzuführen ist: `FAILS_PER_HOUR`, `FAILS_PER_DAY` und `FAILS_PER_WEEK`. Jede HA-Umgebung ist unterschiedlich. Sie müssen entscheiden, wie viele DB2-Ausfälle (Fails) akzeptabel sind. Nach jedem akzeptablen Ausfall wird DB2 auf der gleichen Maschine wieder gestartet. Wenn einer dieser drei Schwellenwerte für Ausfälle überschritten wird, wird der logische Host, dem das Exemplar oder die Datenbankpartition zugeordnet ist, auf eine andere Maschine versetzt.

Die Variable `FM_FAIL_SEV` gibt an, ob die Funktionsübernahme "soft" oder "hard" ist. Weitere Informationen finden Sie in der Sun Cluster-Dokumentation über `hact1(1m)`.

Die Variablen `DB2START_TIMEOUT` und `DB2STOP_TIMEOUT` geben die maximale Anzahl von Sekunden an, die **db2start** und **db2stop** ausgeführt werden dürfen. Wenn das angegebene Intervall abgelaufen ist, betrachtet der HA-Agent die Operation als fehlgeschlagen und versucht, das Exemplar erneut zu starten.

Es gibt einige Benutzerprozeduren, die nicht einem bestimmten Exemplar zugeordnet sind. Normalerweise werden diese Prozeduren mit Root-Berechtigung ausgeführt. Dies kann mit Hilfe der Variablen `SCRIPT_USER`, mit der die Benutzer-ID angegeben wird, die diese Prozeduren ausführen kann, überschrieben werden.

Ausführen von DB2-Befehlen durch Steuermethoden

Der DB2-HA-Agent verwendet den Befehl **su**, um Befehle als Exemplareigner auszuführen. Der tatsächliche Befehl könnte in etwa wie folgt aussehen:

```
su - <exemplar> -c "db2stop"
```

Dabei ist *exemplar* der Benutzername des Exemplars.

Es ist wichtig, sicherzustellen, daß die Datei `.profile` des Exemplareigners **su**-"freundlich" ist. Wenn dies nicht der Fall ist, funktioniert der Befehl **su** eventuell nicht richtig. Rufen Sie den Befehl **su** manuell oder aus einer Prozedur auf, um zu prüfen, ob der Befehl erfolgreich ausgeführt wird.

Installation und Konfiguration

Bevor Sie diesen Abschnitt lesen, machen Sie sich mit der SC2.2-Software vertraut. Dieser Abschnitt setzt voraus, daß Sie wissen, wie SC2.2 und HA-NFS eingerichtet wird, und daß Sie wissen, wie der Datenträgermanager (Volume Manager) verwendet wird. Neben den anderen erforderlichen Programmkorrekturen (Patches) für DB2 sind für den HA-Agenten die folgenden Programmkorrekturen erforderlich:

```
Solaris 2.6:  
105210-17 (oder später)  
105786-05 (oder später)
```

Anmerkung: Für Solaris 7 (Solaris 2.7) sind keine Programmkorrekturen erforderlich.

Allgemeine Installationsschritte

1. Installieren Sie SC2.2 auf allen Maschinen im Cluster. Während der Installation fragt SC2.2, welche Agenten zu installieren sind. Da DB2 nicht mit SC2.2 ausgeliefert wird, ist DB2 nicht in der Agentenliste enthalten. Der Agent für DB2 wird mit DB2 installiert und über den Befehl **hadb2_reg** registriert.

2. Konfigurieren Sie die logischen Hosts mit Plattengruppen und logischen IP-Adressen.

Installation und Konfiguration von DB2 UDB Enterprise Edition

1. Erstellen Sie das Ausgangsverzeichnis (home) für das Exemplar unter dem Dateisystem eines logischen Hosts.
2. Installieren Sie DB2 auf allen Maschinen im Cluster.
3. Erstellen Sie das Exemplar auf der Maschine im Cluster, auf dem sich zur Zeit das Ausgangsverzeichnis für das Exemplar befindet.
4. Fügen Sie den Benutzer für das Exemplar den anderen Maschinen im Cluster hinzu, wobei sicherzustellen ist, daß die numerische Benutzer-ID identisch ist.
5. Registrieren Sie den Service hadb2 mit dem Befehl **hadb2_reg**.
6. Führen Sie den Befehl **hadb2_setup** aus, um HA für das Exemplar einzurichten.

Installation und Konfiguration von DB2 UDB Enterprise - Extended Edition

1. Erstellen Sie das Ausgangsverzeichnis für den HA-Exemplareigner:
 - a. Für den Bereitschaftsmodus (Hot Standby) erstellen Sie das Ausgangsverzeichnis für das Exemplar unter dem Dateisystem eines logischen Hosts.
 - b. Für gegenseitige Übernahme konfigurieren Sie HA-NFS und exportieren das Ausgangsverzeichnis aus einem der logischen Hosts. Hängen Sie auf einer der Maschinen das HA-NFS-Verzeichnis unter dem gewählten Mount-Punkt an.
2. Installieren Sie DB2 auf allen Maschinen im Cluster.
3. Erstellen Sie das Exemplar auf der Maschine, auf der das HA-NFS-Dateisystem angehängt ist.
4. Fügen Sie den Benutzer für das Exemplar den anderen Maschinen im Cluster hinzu, wobei sicherzustellen ist, daß die numerische Benutzer-ID identisch ist.
5. Registrieren Sie den Service hadb2 mit dem Befehl **hadb2_reg**.
6. Führen Sie den Befehl **hadb2_setup** aus, um HA für das Exemplar einzurichten.

Anmerkung: Die Verwendung von NIS zur Definition von Informationen für das HA-Exemplar ist nicht zu empfehlen, weil NIS einen einzelnen Fehlerpunkt bilden kann.

Der Befehl **hadb2_setup**

Der Befehl **hadb2_setup** ist der zentrale Punkt der Programme, die mit dem DB2-HA-Agenten geliefert werden. Er kann zur Einrichtung, Änderung und Löschung eines Exemplars verwendet werden. Ferner kann er dazu dienen,

den Service `hadb2_setup` zu aktivieren und zu inaktivieren. Bei Verwendung dieses Befehls ist es nicht erforderlich, die Datei `hadb2tab` manuell zu editieren.

Anmerkung: Der Befehl `hadb2_setup` führt Aktionen nur auf der Maschine aus, auf der er aufgerufen wird. Änderungen, die auf einer Maschine vorgenommen werden, sollten auch auf den anderen Maschinen im Cluster durchgeführt werden.

Folgende Argumente werden unterstützt:

Zum Hinzufügen eines EE-Exemplars:

```
-----  
hadb2_setup -a -i <exemplar> -f [on|off] -h <logischer_host> -p [DATA|ADMIN] -t EE
```

Beispiel:

```
hadb2_setup -a -i db2ee -f off -h log1 -p DATA -t EE
```

Zum Hinzufügen eines EEE-Exemplars:

```
-----  
hadb2_setup -a -i <exemplar> -f [on|off] -h <nfs_host> -l <mount_punkt> \  
-r <ha-nfs_verz> -p [DATA|ADMIN] -t EEE -n "<knoten_info>"
```

Beispiel:

```
hadb2_setup -a -i db2eee -f off -h ha-sun1 -l /export/ha_home \  
-r /log0/home -p DATA -t EEE -n "log0[0,10,20],log1[30,40,50]"
```

Zum Löschen eines Exemplars:

```
-----  
hadb2_setup -d -i <exemplar>
```

Zum Ändern eines Exemplars:

```
-----  
hadb2_setup -m -i <exemplar> [-f [on|off] | -l <mount_punkt> | \  
-h <host> | -p [DATA|ADMIN] | -r <ha-nfs_verz> | -t [EE|EEE] ]
```

Weitere Optionen:

```
-----  
-s <on|off>      Aktivieren/Inaktivieren von hadb2 (für alle HA-Exemplare)  
-y              Annehmen von YES für Sicherheitsprüfungen
```

Zur Aktivierung oder Inaktivierung des Service `hadb2` geben Sie den Schalter `-s` an. Dies ist mit der Verwendung des Befehls `harem` mit den Schaltern `-n` und `-y` und der Angabe des Service `hadb2` äquivalent. Weitere Informationen zum Befehl `harem(1m)` finden Sie in der Sun Cluster-Dokumentation.

Der Fehlermonitor für das Exemplar kann mit dem Schalter `-f` inaktiviert werden. Dies bewirkt, daß der Fehlermonitor für das Exemplar auf der lokalen Maschine gestoppt und die Datei `hadb2tab` geändert wird, um die Tatsache zu vermerken, daß die Fehlerüberwachung ausgeschaltet wurde.

Für EE-Exemplare wird das Ausschalten der Fehlerüberwachung auf allen Maschinen empfohlen, wenn das Exemplar von einer anderen Maschine übernommen wird. Für EEE-Exemplare muß die Fehlerüberwachung auf allen Maschinen inaktiviert werden, die Datenbankpartitionen für das Exemplar enthalten, bevor es manuell heruntergefahren wird.

Zum Löschen eines Exemplars dient der Schalter `-d`. Durch diesen Schalter wird das Exemplar nur aus der Datei `hadb2tab` entfernt. Andere Dateien oder Verzeichnisse werden nicht gelöscht oder geändert. Da die Datei `hadb2tab` die Hauptkonfigurationsdatei für den HA-DB2-Agenten ist, bewirkt das Löschen eines Exemplars aus dieser Datei, daß die Steuermethoden das Vorhandensein des Exemplars nicht mehr erkennen.

Zum Ändern eines Exemplars verwenden Sie den Schalter `-m`. Durch diesen Schalter werden Informationen in der Datei `hadb2tab` geändert. Andere Dateien oder Verzeichnisse werden nicht gelöscht oder geändert. Der Schalter `-m` kann mit jedem Schalter verwendet werden, der sich auf Informationen in der Datei `hadb2tab` bezieht. Die Datei `db2nodes.cfg` und die Datei `hadb2-eee.cfg` müssen im Anschluß an die Erstkonfiguration manuell geändert werden, weil der Befehl **hadb2_setup** eine Änderung dieser Dateien nicht unterstützt.

Das Hinzufügen eines Exemplars ist etwas komplizierter.

Für EE-Exemplare sind die folgenden Parameter erforderlich:

```
hadb2_setup -a -i <exemplar> -f <fm> -h <logischer_host> -t <EEE_oder_EE> -p <zweck>
```

Dabei gilt: *exemplar* ist der Name des hinzuzufügenden Exemplars, *fm* gibt an, ob die Fehlerüberwachung anfangs aktiv (ON) oder inaktiv (OFF) ist, *logical_host* ist der zugeordnete logische Host, *EEE_or_EE* wird auf EE gesetzt und *zweck* kann entweder DATA oder ADMIN sein.

Für EEE-Exemplare sind die folgenden Parameter erforderlich:

```
hadb2_setup -a -i <exemplar> -f <fm> -h <nfs_host> -t <EEE_oder_EE> -p  
<zweck> -l <mount_punkt> -r <HA-NFS_verzeichnis> -n <knoten_info>
```

Dabei gilt: *exemplar* ist der Name des hinzuzufügenden Exemplars, *fm* gibt an, ob die Fehlerüberwachung anfangs aktiv (ON) oder inaktiv (OFF) ist, *nfs_host* ist der Host-Name für den logischen Host, der das HA-NFS-Dateisystem exportiert, *EEE_or_EE* wird auf EEE gesetzt, *zweck* kann entweder DATA oder ADMIN sein, *mount_punkt* ist der lokale Mount-Punkt für das

HA-NFS-Verzeichnis *HA-NFS_verzeichnis* ist das HA-NFS-Verzeichnis und *knoten_info* sind die Informationen, durch die Datenbankpartitionen einem logischen Host zugeordnet werden. Beispiel:

```
hadb2_setup -a -i db2eee -f on -h jolt -l /export/ha_home -p DATA -t EEE -r  
/log1/home -n "log0[0,1],log1[2,3]"
```

Wenn ein EEE-Exemplar hinzugefügt wird, müssen die Knoteninformationen in Anführungszeichen gesetzt werden. In diesem Beispiel wird das Exemplar "db2eee" zwei logischen Hosts "log0" und "log1" zugeordnet. Die Datenbankpartitionen "0" und "1" des Exemplars "db2eee" werden dem logischen Host "log0" und die Datenbankpartitionen "2" und "3" dem logischen Host "log1" zugeordnet.

Verwenden Sie den Befehl **hadb2_setup**, um ein Exemplar allen Maschinen im Cluster hinzuzufügen. Das Exemplar kann anschließend gestartet werden, indem eine Änderung der Clusterkonfiguration erzwungen wird oder indem der Service hadb2 inaktiviert und wieder aktiviert wird. Dies kann entweder mit Hilfe des Befehls **hareg** oder mit Hilfe des Schalters **-s** des Befehls **hadb2_setup** geschehen. Wenn das Exemplar nicht startet, lesen Sie die Informationen in „Fehlerbehebung“ auf Seite 365.

Wenn der Befehl **hadb2_setup** ein EEE-Exemplar hinzufügt, werden die folgenden Aktionen transparent durchgeführt:

- Überprüfen der angegebenen Informationen. Darin enthalten ist auch eine Überprüfung, ob der Benutzer im System vorhanden ist und ob HA-NFS aktiv ist.
- Erstellen einer Datei `db2nodes.cfg`.
- Erstellen einer Datei `hadb2-eee.cfg`.
- Erstellen einer Datei `.rhosts` für das EEE-Exemplar.
- Erstellen symbolischer Verbindungen vom Standarddatenbankpfad zu Datenverzeichnissen der zugeordneten logischen Hosts.
- Einfügen einer Zeile in die Datei `hadb2tab`.

Zur Vermeidung von Konfigurationsfehlern und zur Sicherstellung, daß das HA-Exemplar nach der Ausführung des Befehls **hadb2_setup** gestartet werden kann, führt der Befehl eine beträchtliche Menge an Tests aus, bevor ein neues Exemplar hinzugefügt wird.

Die Datei `db2nodes.cfg` wird erstellt mit Startinformationen über den aktuellen Clusterstatus gefüllt. Wenn zum Beispiel der logische Host "log0" von der Maschine "crackle" beherbergt wird, enthalten die Einträge für die dem Host "log0" zugeordneten Datenbankpartitionen den Maschinennamen "crackle" und die Hochgeschwindigkeitsverbindung für "crackle":

```
scadmin@crack1e(193)# cat db2nodes.cfg
0 crack1e 0 204.152.65.33
1 crack1e 1 204.152.65.33
2 thrash 0 204.152.65.34
3 thrash 1 204.152.65.34
```

Die Datei `hadb2-eee.cfg` wird nur auf der Basis der im Befehl angegebenen Knoteninformationen erstellt. Für jede Datenbankpartition ist eine Zeile vorhanden:

```
sphere % cat hadb2-eee.cfg
NODE:log0 0
NODE:log0 1
NODE:log1 2
NODE:log1 3
```

Die Datei `.rhost` ist für DB2 UDB EEE erforderlich und sollte alle Host-Namen (oder IP-Adressen) für jede Maschine im Cluster enthalten. Beispiel:

```
crack1e db2eee
204.152.65.1 db2eee
204.152.65.17 db2eee
thrash db2eee
204.152.65.2 db2eee
204.152.65.18 db2eee
crack1e db2eee
jolt db2eee
bump db2eee
thrash.torolab.ibm.com db2eee
crack1e.torolab.ibm.com db2eee
```

In Übereinstimmung mit einem Dateisystemlayout für SMS-Tabellenbereiche richtet der Befehl **hadb2_setup** eine Reihe von Verzeichnissen und symbolischen Verbindungen ein. Dazu gehören:

- Ein Verzeichnis namens "data" unter dem Dateisystem des logischen Hosts für jeden logischen Host.
- Ein Knotenverzeichnis (unter diesem Verzeichnis "data") für jede Datenbankpartition, die dem logischen Knoten zugeordnet ist.
- Symbolische Verbindungen im Standarddatenbankpfad unter `~<exemplar>`, wobei `~<exemplar>` das Ausgangsverzeichnis des Exemplars ist. Für jede Datenbankpartition, die auf das entsprechende Knotenverzeichnis zeigt, ist eine symbolische Verbindung vorhanden. Weitere Informationen finden Sie in „Plattenlayout für EE- und EEE-Exemplare“ auf Seite 341.

Dauer der Funktionsübernahme

Die Dauer einer Funktionsübernahme wird von dem Zeitpunkt an gemessen, zu dem die Daten zuerst nicht mehr verfügbar sind, bis zu dem Zeitpunkt, zu dem sie wieder verfügbar werden. Eine Reihe von Ereignissen, die während einer Funktionsübernahme auftreten, können einen erheblichen Beitrag zur Dauer der Funktionsübernahme leisten:

- Deportieren und Importieren von Platten

Das Deportieren und Importieren von Platten benötigt in der Regel nicht sehr lange Zeit im Vergleich zu anderen Ereignissen, jedoch kann sich dadurch die allgemeine Ausfallzeit verlängern. Je mehr Platten von einer Maschine auf eine andere bei der Funktionsübernahme versetzt werden müssen, um so länger dauert der Prozeß. Wenn es fehlerhafte Platten gibt, kann der Prozeß noch länger dauern.

- Fskck-Prüfung der Dateisysteme, die für einen logischen Host angehängt sind

Bevor die Dateisysteme des logischen Hosts angehängt werden können, müssen sie eine fskck-Prüfung durchlaufen, die den einwandfreien Zustand des Dateisystems sicherstellt. Je größer das Dateisystem ist, desto länger dauert dieser Prozeß. Durch Verwendung eines Journal-Dateisystems kann diese Zeit erheblich verringert werden. Da in einer HA-Umgebung in der Regel Journal-Dateisysteme verwendet werden, spielt die fskck-Zeit gewöhnlich keine große Rolle.

- Benutzerprozeduren, die aus dem HA-Agent aufgerufen werden

Der HA-Agent ruft Benutzerprozeduren auf, wenn sie vorhanden und ausführbar sind. Einige dieser Prozeduren werden synchron ausgeführt und können die Zeit verlängern, die zum Starten von HA-Exemplaren benötigt wird. Stellen Sie sicher, daß sie so schnell wie möglich ausgeführt werden. Ziehen Sie in Betracht, externe Programme, die von diesen Prozeduren aufgerufen werden, im Hintergrund auszuführen.

- HA-NFS

Für ein einzelnes EEE-Exemplar in einer Konfiguration zur gegenseitigen muß Übernahme HA-NFS für das Ausgangsverzeichnis des Exemplar-eigners verwendet werden. HA-NFS verlängert die Funktionsübernahmedauer wegen der Aufschubzeit für *lockd* (definiert im HA-Agent für HA-NFS), die 90 Sekunden bei Ausführung von HA-NFS beträgt. Die Funktionsübernahmedauer ist betroffen, weil jeder Prozeß, der eine Datei im HA-NFS-Dateisystem nach einer Funktionsübernahme sperrt, warten muß, bis die Aufschubfrist abgelaufen ist. Der HA-Agent für DB2 ist der erste Prozeß, der eine Datei unter dem Ausgangsverzeichnis des Exemplar-eigners nach einer Funktionsübernahme sperrt und die Zeit aufzeichnet, die er benötigt, um die erste Sperre zu erhalten. Diese Zeit wird im Statusbericht nach einer Funktionsübernahme angegeben.

- Starten von DB2

Das Starten von DB2 trägt nur einen kleinen Teil zur Dauer der Funktionsübernahme bei. Für ein EE-Exemplar sind dies nur 5 bis 15 Sekunden im Durchschnitt. Für ein EEE-Exemplar trägt es ungefähr 10 Sekunden plus je fünf Sekunden pro Datenbankpartition bei, die übernommen wird. Wenn zum Beispiel drei Datenbankpartitionen übernommen werden, beträgt die Zeit, die durch das Starten dieser drei Datenbankpartitionen hinzugefügt wird, ungefähr 25 Sekunden. Diese Zeit enthält *nicht* die Zeit, die zur Wiederherstellung nach einem Systemabsturz des Exemplars benötigt wird.

- Wiederherstellung nach einem Systemabsturz der Datenbank

Die Wiederherstellung nach einem Systemabsturz ist häufig der größte Einzelposten in der Ausfallzeit bei einer Funktionsübernahme. Wie lange es dauert, eine Datenbank wiederherzustellen, hängt von einer Reihe von Faktoren ab:

- Client-Auslastung. Nur Änderungen an der Datenbank werden in den Transaktionsprotokollen aufgezeichnet. Wenn die Client-Auslastung überwiegend aus Leseoperationen besteht, müssen während der Wiederherstellung nach einem Systemabsturz relativ wenige Transaktionen auf die Datenbank angewendet werden.
- Platten- und Maschinengeschwindigkeit. Die Geschwindigkeit der Platten und der Maschine, auf der das HA-Exemplar ausgeführt wird, spielt ebenfalls eine Rolle für die Zeit, die eine Wiederherstellung der Datenbank benötigt. Je schneller das System arbeitet, desto kürzer ist die für eine Wiederherstellung nach einem Systemabsturz benötigte Zeit.
- Der Wert des Konfigurationsparameters *softmax* der Datenbank. Der Wert des Parameters *softmax* definiert den Prozentsatz der Protokolldateigröße, bei dem ein bedingter Prüfpunkt (soft checkpoint) vorzunehmen und eine Protokollsteuerdatei zu schreiben ist. Die Protokollsteuerdatei dient während der Wiederherstellung nach einem Systemabsturz dazu, zu bestimmen, welche Protokollsätze tatsächlich zur Wiederherstellung der Datenbank in einem konsistenten Zustand erforderlich sind. Eine Verringerung dieses Werts bewirkt, daß der Datenbankmanager die Seitenlöschfunktionen häufiger auslöst und häufigere bedingte Prüfpunkte ansetzt. Obwohl dadurch die Leistung herabgesetzt wird, erfolgt die Wiederherstellung der Datenbank schneller.
- Unterschied zwischen EE- oder EEE-Exemplar. Wenn es sich um ein EEE-Exemplar handelt, werden die Operationen zum Neustart der Datenbank parallel ausgeführt. Jede Datenbankpartition ist für den Neustart des eigenen Teils der Datenbank zuständig. Bei einer Datenmenge von 50 GB für eine Datenbank kann ein Exemplar mit vier Datenbankpartitionen die Datenbank ungefähr viermal schneller als ein EE-Exemplar wiederherstellen.

Fehlerbehebung

In der folgenden Tabelle werden Probleme aufgeführt, die auftreten können, ihre wahrscheinlichen Ursachen beschrieben und Maßnahmen zu ihrer Behebung erläutert.

Tabelle 29. Fehlerbehebung bei hoher Verfügbarkeit unter Sun Cluster 2.2

Symptom	Mögliche Ursache	Maßnahme
Datei-system von logischem Host läßt sich nicht anhängen.	Das Dateisystem des logischen Hosts wird normalerweise während der Funktionsübernahme eines logischen Hosts angehängt und abgehängt. Während der Funktionsübernahme sollten keine aktiven Prozesse oder geöffneten Dateien unter dem Dateisystem des logischen Hosts vorhanden sein. In seltenen Fällen haben Prozesse, die nicht zwangsweise beendet werden können, ihr aktuelles Arbeitsverzeichnis unter dem Dateisystem des logischen Hosts. Verwenden Sie <code>fuser(1m)</code> oder ein GNU-Dienstprogramm namens <code>lsof</code> , um herauszufinden, ob ein Prozeß unter dem Mount-Punkt vorhanden ist. Fehlernachrichten werden generiert, wenn das Dateisystem des logischen Hosts nicht angehängt werden kann. ^a	Starten Sie das System erneut oder versetzen Sie das Dateisystem des logischen Hosts in einen anderen Namen und erstellen Sie es erneut. Dadurch kann der eingefrorene Prozeß unter dem Verzeichnis bleiben (da er nicht abgebrochen werden kann), und das Anhängen kann stattfinden. ^b
Das Zeitlimit für <code>db2start</code> oder <code>db2stop</code> funktioniert nicht.	Ein Signal <code>SIGALRM</code> kann eventuell nicht aus einem blockierenden Systemaufruf ausbrechen. Statt dessen wird der Systemaufruf erneut gestartet so, als wäre die Markierung (Flag) <code>SA_RESTART</code> mit <code>sigaction()</code> gesetzt. Dies bewirkt, daß Zeitlimits für die DB2-HA-Agenten ignoriert werden und die Agentenmethode blockiert, anstatt einen blockierten Befehl <code>db2start</code> oder <code>db2stop</code> zu beheben.	Wenden Sie die erforderliche Programmkorrektur (Patch) 105210-17 (oder eine spätere) für Solaris 2.6 an.

Tabelle 29. Fehlerbehebung bei hoher Verfügbarkeit unter Sun Cluster 2.2 (Forts.)

Symptom	Mögliche Ursache	Maßnahme
Anmelden in Exemplar blockiert.	Obwohl es zahlreiche Ursachen hierfür geben kann, gehören zu den häufigsten Ursachen NFS-Probleme und das Programm <code>/usr/sbin/quota</code> .	Überprüfen Sie die angehängten NFS, um sicherzustellen, daß sie ordnungsgemäß funktionieren, und suchen Sie nach Quota-Prozessen, die dem Exemplareigner gehören. Das Problem kann eventuell durch eine Änderung des Quota-Programms in eine symbolische Verbindung zu <code>/bin/true</code> behoben werden, was jedoch im Ermessen des Systemadministrators liegt. Dies ist keine empfohlene Lösung, aber sie funktioniert vielleicht.
Nach der Einrichtung eines EEE-Exemplars läßt sich dieses nicht starten.	Der Befehl hadb2_setup fügt keine Ports in der Datei <code>/etc/services</code> ein. Es wird erwartet, daß der Administrator diese manuell hinzufügt. Es wird eine Fehlermeldung zurückgegeben. ^c	Stellen Sie sicher, daß Sie die richtigen Ports in der Datei <code>/etc/services</code> eingetragen haben.

Tabelle 29. Fehlerbehebung bei hoher Verfügbarkeit unter Sun Cluster 2.2 (Forts.)

Symptom	Mögliche Ursache	Maßnahme
Methode START_NET kann DB2 nicht starten.		<p>Schalten Sie die Fehlerüberwachung aus, um sicherzustellen, daß das Exemplar nicht übernommen wird. Melden Sie sich als Exempleigner an, und versuchen Sie DB2 manuell zu starten.</p> <ol style="list-style-type: none"> 1. Stellen Sie sicher, daß in der Konfigurationsdatei hadb2tab der korrekte Exemplartyp angegeben ist. Wenn zum Beispiel eine Datei db2nodes.cfg für ein administratives EE-Exemplar vorhanden ist, verursacht dies Probleme. Die HA-Agentenmethoden können diese Probleme nicht überwinden. 2. Stellen Sie sicher, daß die Datei .rhosts vorhanden ist und gültige Einträge enthält. 3. Stellen Sie sicher, daß das HA-NFS-Dateisystem mit Root-Berechtigung von allen Maschinen im Cluster gemeinsam benutzt werden kann. 4. Überprüfen Sie die Kernel-Parameter, und stellen Sie sicher, daß sie korrekt sind. 5. Stellen Sie sicher, daß die Datei /etc/services Einträge für das Exemplar enthält.

Tabelle 29. Fehlerbehebung bei hoher Verfügbarkeit unter Sun Cluster 2.2 (Forts.)

Symptom	Mögliche Ursache	Maßnahme
Das Exemplar funktioniert nur auf einer Maschine.	<ul style="list-style-type: none"> • Die numerische <i>uid</i> für das Exemplar ist vielleicht nicht auf allen Maschinen im Cluster gleich. • Die Kernel-Parameter sind vielleicht nicht auf jeder Maschine im Cluster gültig. • Die Datei <i>hadb2tab</i> ist vielleicht nicht auf jeder Maschine im Cluster gleich. • Andere Konfigurationsdateien, wie zum Beispiel die Datei <i>vfstab</i> für logische Hosts, sind vielleicht nicht auf jeder Maschine im Cluster gleich. 	Wenn keine dieser Ursachen zuzutreffen scheint, versuchen Sie, sich als Exemplareigner anzumelden und DB2 manuell zu starten. Für EE-Exemplare sollte dies funktionieren, wenn der logische Host, der dem Exemplar zugeordnet ist, auf der aktuellen Maschine aktiv ist. Für EEE-Exemplare sollte dies von jeder Maschine im Cluster aus funktionieren, die die Datenbankpartitionen übernehmen kann.
su - <exemplar> -c "db2start" funktioniert nicht.	<ul style="list-style-type: none"> • Die Datei <i>.profile</i> für das Exemplar ist eventuell nicht su-"freundlich". • Es gibt ein bekanntes Problem mit der Bourne-Shell (<i>/bin/sh</i>), in der der Befehl su manuell, jedoch nicht über den HA-Agenten funktioniert. 	<ul style="list-style-type: none"> • Versuchen Sie mit Root-Berechtigung diesen Befehl manuell auszuführen, und stellen Sie sicher, daß er funktioniert, bevor Sie den Versuch über den HA-Agenten wiederholen. • Wechseln Sie zur Korn-Shell (<i>/bin/ksh</i>), falls erforderlich.
Das EEE-Exemplar kann nicht starten, aber das Ausgangsverzeichnis wird angehängt.	Das HA-NFS-Verzeichnis wurde eventuell nicht mit der Root-Berechtigung an die Maschinen im Cluster exportiert. Sowohl für DB2 als auch für die HA-Agenten muß dieses Verzeichnis ordnungsgemäß arbeiten.	Versuchen Sie testweise, eine Datei (als Root) unter dem Ausgangsverzeichnis des Exemplareigners zu erstellen.

Table 29. Fehlerbehebung bei hoher Verfügbarkeit unter Sun Cluster 2.2 (Forts.)

Symptom	Mögliche Ursache	Maßnahme
Der Versuch eines Zugriffs auf das EEE-Exemplarverzeichnis liefert eine Fehler-nachricht "Stale NFS file handle".	Es gibt eventuell noch Prozesse unter dem Ausgangsverzeichnis des Exemplareigners.	Hängen Sie das Ausgangsverzeichnis des Exemplareigners ab, und lassen Sie den HA-Agenten es erneut anhängen. Der HA-Agent hängt es wieder an, wenn der Service hadb2 inaktiviert und wieder aktiviert wird (Siehe Beschreibung des Schalters -s in „Der Befehl hadb2_setup“ auf Seite 358).

Tabelle 29. Fehlerbehebung bei hoher Verfügbarkeit unter Sun Cluster 2.2 (Forts.)

Symptom	Mögliche Ursache	Maßnahme
Steuer- methoden werden durch SC2.2 nicht erfolgreich ausgeführt.	Der Service hadb2 ist eventuell nicht bei der Sun Cluster-Software registriert oder er ist inaktiviert.	<p>Wenn es scheint, als könnten die Steuermethoden über die Befehlszeile normal ausgeführt werden, prüfen Sie die SYSLOG-Dateien auf Fehlermeldungen, die helfen, das Problem zu erklären. Stellen Sie sicher, daß der Service hadb2 bei der Sun Cluster-Software registriert ist und daß er aktiviert ist.</p> <p>Die manuelle Ausführung der Methoden ist zur Lösung eines Problems hilfreich.^d</p> <p>Die Methoden sollten mit Root-Berechtigung und den entsprechenden Befehlszeilenparametern ausgeführt werden. Wenn die Liste logischer Host leer ist, sollte der Parameter als "" angegeben werden. Doppelte Anführungszeichen ohne Leerzeichen als Trenner bezeichnen ein leeres Argument. Beispiel:</p> <pre>hadb2_startnet log0,log1 "" 600</pre> <p>Das erste Argument log0,log1 teilt der Methode hadb2_startnet mit, daß die logischen Hosts log0 und log1 auf der aktuellen Maschine aktiv sind. Das zweite Argument ist leer, was der Methode hadb2_startnet anzeigt, daß es keine anderen logischen Hosts gibt, die auf anderen Maschinen im Cluster aktiv sind (alle sind auf der aktuellen Maschine aktiv). Das dritte Argument teilt der Methode mit, daß SC2.2 ein Zeitlimit von 600 Sekunden setzt.</p>

Table 29. Fehlerbehebung bei hoher Verfügbarkeit unter Sun Cluster 2.2 (Forts.)

Symptom	Mögliche Ursache	Maßnahme
Benutzerprozeduren lassen sich nicht ausführen.	Die Benutzerprozeduren können nur ausgeführt werden, wenn sie in den richtigen Verzeichnissen vorhanden und ausführbar sind.	Überprüfen Sie die Eigentumsrechte und Attribute der Dateien. Wenn eine Prozedur sich nicht ausführen läßt, wenden Sie sich an den IBM Kundendienst. Leiten Sie ein Verzeichnislisting der Prozedur, die sich nicht ausführen läßt und die SYSLOG-Ausgabe für eine Funktionsübernahme bzw. einer Änderung der Clusterkonfiguration weiter, durch die die Prozedur hätte ausgeführt werden müssen.
Informationen werden nicht in der in /etc/syslog.conf angegebenen Datei protokolliert.		Verwenden Sie touch(1) zur Erstellung der Datei, die in der Datei /etc/syslog.conf angegeben ist, und starten Sie den SYSLOG-Dämon erneut.

Tabelle 29. Fehlerbehebung bei hoher Verfügbarkeit unter Sun Cluster 2.2 (Forts.)

Symptom	Mögliche Ursache	Maßnahme
<p>^a Fehlernachrichten, die generiert werden, wenn das Dateisystem des logischen Hosts nicht angehängt werden kann, können etwa wie folgt aussehen:</p> <pre>Aug 17 11:14:01 rash ID[SUNWcluster.loghost.1170]: importing data1 Aug 17 11:14:06 rash ID[SUNWcluster.scnfs.3040]: mount -F ufs -o "" /dev/vx/dsk/data1/data1-stat /log1 failed. Aug 17 11:14:07 rash ID[SUNWcluster.ccd.cdd.5304]: error freeze cmd = /opt/SUNWcluster/bin/loghost_sync CCDSYNC_POST_ADDU LOGHOST_CM:log1:rash /etc/opt/SUNWcluster/conf/ccd.database 2 "0 1" 1 error code = 1</pre>		
<p>^b Beispiel:</p> <pre>scadmin@rash(218)# ps -fe egrep db2 db2ee 1984 1 0 0:01 <defunct></pre> <p>Lösung:</p> <pre>scadmin@rash(229)# cd / scadmin@rash(230)# mv /log1 /log1.bkp scadmin@rash(231)# mkdir /log1</pre>		
<p>^c Die Fehlernachricht kann etwa wie folgt aussehen:</p> <pre>SQL6030N Der Befehl START oder STOP DATABASE MANAGER ist fehlgeschlagen. Ursachencode: "13".</pre>		
<p>^d Wenn die Methode hadb2_startnet zum Beispiel libdb2.so.1 nicht finden kann, aber über die Sun Cluster-Software normal ausgeführt wird, werden keine Fehlernachrichten gemeldet. Eine manuelle Ausführung der Methode führt zu folgendem Ergebnis:</p> <pre>scadmin@crackle(213)# hadb2_startnet '''log0,log1' 600 ld.so.1: hadb2_startnet: fatal: libdb2.so.1: open failed: No such file or directory Killed</pre>		

Teil 5. Anhänge und Schlußteil

Anhang A. Verwenden der DB2-Bibliothek

Die Bibliothek für DB2 Universal Database besteht aus Online-Hilfe, Handbüchern (PDF und HTML) und Beispielprogrammen in HTML-Format. Im folgenden wird beschrieben, welche Informationen bereitgestellt werden und wie Sie darauf zugreifen können.

Über **Information - Unterstützung** können Sie online auf die Produktinformationen zugreifen. Weitere Informationen finden Sie in „Zugreifen auf Informationen mit "Information - Unterstützung"“ auf Seite 394. Sie können sich im Web Informationen zu Tasks und zur Fehlerbehebung sowie DB2-Bücher, Beispielprogramme und DB2-Informationen anzeigen lassen.

PDF-Dateien und gedruckte Bücher für DB2

Informationen zu DB2

In der folgenden Tabelle sind die DB2-Handbücher in vier Kategorien unterteilt:

DB2-Benutzerhandbücher und -Referenzinformationen

Diese Bücher enthalten die allgemeinen DB2-Informationen für alle Plattformen.

DB2-Installations- und -Konfigurationsinformationen

Diese Bücher gelten für DB2 auf einer bestimmten Plattform. So steht beispielsweise jeweils ein separates Handbuch *Einstieg* (Quick Beginnings) für DB2 für OS/2-, Windows- und UNIX-Plattformen zur Verfügung.

Plattformübergreifende Beispielprogramme in HTML

Bei diesen Beispielen handelt es sich um die HTML-Versionen der mit Application Development Client installierten Beispielprogramme. Sie dienen zur Information und können die Programme selbst nicht ersetzen.

Release-Informationen

Diese Dateien enthalten die neuesten Informationen, die in die DB2-Handbücher nicht mehr aufgenommen werden konnten.

Die Installationshandbücher, Release-Informationen und Lernprogramme können im HTML-Format direkt von der Produkt-CD-ROM angezeigt werden. Die meisten Handbücher stehen auf der Produkt-CD-ROM im HTML-Format zur Verfügung und können angezeigt werden. Auf der CD-ROM mit DB2-Veröffentlichungen stehen die Handbücher im PDF-Format zur Verfügung und können mit Adobe Acrobat angezeigt und gedruckt werden. Darüber hinaus können Sie gedruckte Veröffentlichungen bei IBM bestellen. Siehe hierzu „Bestellen der gedruckten Handbücher“ auf Seite 389. Die folgende Tabelle enthält eine Liste der Bücher, die bestellt werden können.

Auf OS/2- und Windows-Plattformen können Sie die HTML-Dateien im Verzeichnis `sql11ib\doc\html` installieren. Die DB2-Informationen werden in verschiedene Sprachen übersetzt, jedoch nicht alle Informationen in alle Sprachen. Sind bestimmte Informationen in einer Sprache nicht verfügbar, wird statt dessen die englische Version dieser Informationen zur Verfügung gestellt.

Auf UNIX-Plattformen können Sie die HTML-Dateien in mehreren Sprachen installieren, und zwar in den Unterverzeichnissen `doc/%L/html`, wobei `%L` für den Code der jeweiligen Landessprache steht. Weitere Informationen finden Sie im entsprechenden Handbuch *Einstieg*.

Es gibt verschiedene Möglichkeiten, auf DB2-Bücher und -Informationen zuzugreifen:

- „Anzeigen von Online-Informationen“ auf Seite 393
- „Suchen nach Online-Informationen“ auf Seite 398
- „Bestellen der gedruckten Handbücher“ auf Seite 389
- „Drucken der PDF-Handbücher“ auf Seite 388

Tabelle 30. Informationen zu DB2

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
DB2-Benutzerhandbücher und -Referenzinformationen			
<i>Systemverwaltung</i>	<i>Systemverwaltung: Konzept.</i> Dieses Handbuch enthält eine Übersicht über Datenbankkonzepte, Informationen zu Aspekten des Datenbankentwurfs (wie z. B. zum logischen und physischen Datenbankentwurf) sowie eine Erläuterung zu hohen Verfügbarkeit.	SC12-2879 db2d1g70 SC12-2877 db2d2g70	db2d0
	<i>Systemverwaltung: Implementierung.</i> Dieses Handbuch enthält Informationen zu Implementierungsaspekten, wie beispielsweise zur Implementierung des Datenbankentwurfs, zum Zugriff auf Datenbanken sowie zu Prüfungs-, Sicherungs- und Wiederherstellungsverfahren.	SC12-2878 db2d3g70	
	<i>Systemverwaltung: Optimierung.</i> Dieses Handbuch enthält Informationen zur Datenbankumgebung sowie zur Auswertung und Optimierung der Anwendungsleistung.		
	Sie können die drei Bände des Handbuchs <i>Systemverwaltung</i> in englischer Sprache in den USA und Kanada über die Formnummer SBOF-8934 bestellen.		

Tabelle 30. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>Administrative API Reference</i>	Dieses Handbuch enthält eine Beschreibung zu den DB2-Anwendungsprogrammierschnittstellen (APIs) und -Datenstrukturen, die Sie zum Verwalten Ihrer Datenbank verwenden können. Darüber hinaus wird in diesem Handbuch erläutert, wie Sie APIs von Ihren Anwendungen aus aufrufen können.	SC09-2947 db2b0e70	db2b0
<i>Application Building Guide</i>	Dieses Handbuch umfaßt Informationen zur Umgebungskonfiguration sowie Anweisungsschritte zum Kompilieren, Verbinden und Ausführen von DB2-Anwendungen auf Windows-, OS/2- und UNIX-Plattformen.	SC09-2948 db2axe70	db2ax
<i>APPC, CPI-C, and SNA Sense Codes</i>	Dieses Handbuch enthält Basisinformationen zu APPC-, CPI-DFV- und SNA-Prüfcodes, die bei der Arbeit mit DB2 Universal Database-Produkten ausgegeben werden können.	Keine Formnummer db2ape70	db2ap
	Nur im HTML-Format verfügbar.		
<i>Application Development Guide</i>	Dieses Handbuch enthält eine Erläuterung zur Entwicklung von Anwendungen, die mit Hilfe von eingebettetem SQL bzw. JAVA (JDBC und SQLJ) auf DB2-Datenbanken zugreifen. Unter anderem wird das Schreiben von gespeicherten Prozeduren, das Schreiben von benutzerdefinierten Funktionen, das Erstellen von benutzerdefinierten Typen, das Verwenden von Auslösern und das Entwickeln von Anwendungen in partitionierten Umgebungen oder mit Systemen zusammenschlossener Datenbanken beschrieben.	SC09-2949 db2a0e70	db2a0

Tabelle 30. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>CLI Guide and Reference</i>	Dieses Handbuch erklärt die Entwicklung von Anwendungen, die für den Zugriff auf DB2-Datenbanken DB2 Call Level Interface verwenden, eine aufrufbare SQL-Schnittstelle, die mit der Microsoft-ODBC-Spezifikation kompatibel ist.	SC09-2950 db2l0e70	db2l0
<i>Command Reference</i>	Dieses Handbuch enthält eine Erläuterung zur Verwendung des Befehlszeilenprozessors und eine Beschreibung der DB2-Befehle für die Datenbankverwaltung.	SC09-2951 db2n0e70	db2n0
<i>Konnektivität Ergänzung</i>	Dieses Handbuch enthält Konfigurations- und Referenzinformationen zur Verwendung von DB2 für AS/400, DB2 für OS/390, DB2 für MVS oder DB2 für VM als DRDA-Anwendungs-Requester mit DB2 Universal Database-Servern. Darüber hinaus enthält dieses Handbuch Informationen zur Verwendung von DRDA-Anwendungs-Servern mit DB2 Connect-Anwendungs-Requestern. Dieses Buch ist lediglich im HTML- und PDF-Format verfügbar.	Keine Form- nummer db2h1g70	db2h1
<i>Versetzen von Daten Dienstprogramme und Referenz</i>	Dieses Handbuch enthält eine Erläuterung zur Verwendung der DB2-Dienstprogramme, wie beispielsweise IMPORT, EXPORT, LOAD, AUTOLOADER und DPROF, die das Verschieben von Daten vereinfachen.	SC12-2881 db2dmg70	db2dm
<i>Data Warehouse-Zentrale Verwaltung</i>	Dieses Handbuch enthält Informationen zur Erstellung und Verwaltung eines Data Warehouse mit Hilfe der Data Warehouse-Zentrale.	SC12-2885 db2ddg70	db2dd
<i>Data Warehouse Center Application Integration Guide</i>	Dieses Handbuch enthält Informationen, die Programmierer bei der Integration von Anwendungen in die Data Warehouse-Zentrale sowie in den Information Catalog Manager unterstützen.	SC26-9994 db2ade70	db2ad

Tabelle 30. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>DB2 Connect Benutzerhandbuch</i>	Dieses Handbuch enthält eine Beschreibung der Konzepte der DB2 Connect-Produkte, allgemeine Informationen zur Verwendung sowie Informationen zur Programmierung dieser Produkte.	SC12-2880 db2c0g70	db2c0
<i>DB2 Query Patroller Administration Guide</i>	Dieses Handbuch enthält eine Übersicht über den Betrieb des DB2 Query Patroller-Systems, spezifische Informationen zum Systembetrieb und zur Verwaltung sowie Task-Informationen zu den GUI-Verwaltungsdienstprogrammen.	SC09-2958 db2dwe70	db2dw
<i>DB2 Query Patroller User's Guide</i>	In diesem Handbuch wird die Verwendung der Tools und Funktionen von DB2 Query Patroller beschrieben.	SC09-2960 db2wwe70	db2ww
<i>Glossar</i>	Dieses Handbuch enthält Definitionen zu den in DB2 und den zugehörigen Komponenten verwendeten Begriffen. Es ist im Handbuch <i>SQL Reference</i> enthalten und steht außerdem separat im HTML-Format zur Verfügung.	Keine Formnummer db2t0g70	db2t0
<i>DB2 UDB Image, Audio und Video Extender Verwaltung und Programmierung</i>	Dieses Handbuch enthält Basisinformationen zu DB2 Extender, Informationen zur Verwaltung und Konfiguration von IAV Extender sowie Informationen zur Programmierung mit Hilfe von IAV Extender. Es enthält Referenzinformationen, Diagnoseinformationen (mit Nachrichten) und Beispiele.	SC12-2892 dmbu7g70	dmbu7
<i>Information Catalog Manager Systemverwaltung</i>	Dieses Handbuch enthält eine Anleitung zur Verwaltung von Informationskatalogen.	SC12-2886 db2dig70	db2di
<i>Information Catalog Manager Programming Guide and Reference</i>	Dieses Handbuch enthält Definitionen für die Architekturschnittstellen für Information Catalog Manager.	SC26-9997 db2bie70	db2bi
<i>Information Catalog Manager Benutzerhandbuch</i>	Dieses Handbuch enthält Informationen zur Verwendung der Information Catalog Manager-Benutzerschnittstelle.	SC12-2887 db2aig70	db2ai

Tabelle 30. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>DB2 Installation und Konfiguration Ergänzung</i>	Dieses Handbuch enthält Anweisungen zur Planung, Installation und Konfiguration von plattformspezifischen DB2-Clients. Darüber hinaus enthält es Informationen zu Bindevorgängen, zum Einrichten der Client/Server-Kommunikation, zu DB2-GUI-Tools, zu DRDR-AS, zur verteilten Installation, zur Konfiguration von verteilten Anforderungen sowie zum Zugriff auf heterogene Datenquellen.	GC12-2864 db2iyg70	db2iy
<i>Fehlernachrichten</i>	Dieses Handbuch enthält eine Liste der Nachrichten und Codes, die von DB2, vom Information Catalog Manager und von der Data Warehouse-Zentrale ausgegeben werden, sowie eine Beschreibung der jeweils erforderlichen Benutzeraktionen. Sie können beide Bände des Handbuchs <i>Fehlernachrichten</i> in englischer Sprache in den USA und Kanada unter der Formnummer SBOF-8932 bestellen.	Band 1 GC12-2875 db2m1g70 Band 2 GC12-2888 db2m2g70	db2m0
<i>OLAP Integration Server Administration Guide</i>	Dieses Handbuch enthält eine Erläuterung zur Verwendung der Komponente Administration Manager von OLAP Integration Server.	SC27-0787 db2dpe70	n/v
<i>OLAP Integration Server Metaoutline User's Guide</i>	Dieses Handbuch enthält eine Erläuterung zum Erstellen und Ausfüllen von OLAP-Metastrukturen mit Hilfe der OLAP Metaoutline-Standard-schnittstelle (nicht mit Hilfe des OLAP Metaoutline Assistant).	SC27-0784 db2upe70	n/v
<i>OLAP Integration Server Model User's Guide</i>	Dieses Handbuch enthält eine Erläuterung zum Erstellen von OLAP-Modellen mit Hilfe der OLAP Model-Standard-schnittstelle (nicht mit Hilfe des OLAP Model Assistant).	SC27-0783 db2lpe70	n/v
<i>OLAP Konfiguration und Benutzerhandbuch</i>	Dieses Handbuch enthält Informationen zur Konfiguration und Einrichtung von OLAP Starter Kit.	SC12-2889 db2ipg70	db2ip

Tabelle 30. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>OLAP Tabellenkalkulations-Add-In Benutzerhandbuch für Excel</i>	Dieses Handbuch enthält eine Beschreibung zur Verwendung des Tabellenkalkulationsprogramms Excel zum Analysieren von OLAP-Daten.	SC12-2890 db2epg70	db2ep
<i>OLAP Tabellenkalkulations-Add-In Benutzerhandbuch für Lotus 1-2-3</i>	Dieses Handbuch enthält eine Beschreibung zur Verwendung des Tabellenkalkulationsprogramms Lotus 1-2-3 zum Analysieren von OLAP-Daten.	SC12-2891 db2tpg70	db2tp
<i>Replikation Benutzer- und Referenzhandbuch</i>	Dieses Handbuch enthält Informationen zur Planung, Konfiguration, Verwaltung und Verwendung der mit DB2 gelieferten Replikations-Tools.	SC12-2884 db2e0g70	db2e0
<i>Spatial Extender Benutzer- und Referenzhandbuch</i>	Dieses Handbuch enthält Informationen zur Installation, Konfiguration, Verwaltung, Programmierung und Fehlerbehebung für den Spatial Extender. Darüber hinaus enthält es zentrale Beschreibungen räumlicher Datenkonzepte sowie spezifische Referenzinformationen (Nachrichten und SQL) für den Spatial Extender.	SC12-2894 db2sbg70	db2sb
<i>SQL Erste Schritte</i>	Dieses Handbuch enthält eine Einführung in die SQL-Konzepte sowie Beispiele für eine Reihe von Konstrukten und Tasks.	SC12-2882 db2y0g70	db2y0
<i>SQL Reference, Band 1 und Band 2</i>	Dieses Handbuch beschreibt die Syntax, die Semantik und die Regeln von SQL. Darüber hinaus enthält das Handbuch Informationen zu Inkompatibilitäten zwischen Release-Ständen, Produkteinschränkungen und Katalogsichten. Sie können beide Bände des Handbuchs <i>SQL Reference</i> in englischer Sprache in den USA und Kanada unter der Formnummer SBOF-8933 bestellen.	Band 1 SC09-2974 db2s1e70 Band 2 SC09-2975 db2s2e70	db2s0

Tabelle 30. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>System Monitor Guide and Reference</i>	Dieses Handbuch enthält eine Beschreibung zum Sammeln unterschiedlicher Informationen zu Datenbanken und dem Datenbankmanager. In diesem Buch wird erläutert, wie Sie mit Hilfe dieser Informationen einen Einblick in Datenbankaktivitäten erhalten, die Leistung verbessern und Fehlerursachen feststellen können.	SC09-2956 db2f0e70	db2f0
<i>Text Extender Verwaltung und Programmierung</i>	Dieses Handbuch enthält Basisinformationen zu DB2 Extender, Informationen zur Verwaltung und Konfiguration von Text Extender sowie zur Programmierung mit Hilfe von Text Extender. Es bietet Referenzinformationen, Diagnoseinformationen (mit Nachrichten) und Beispiele.	SC12-2893 desu9g70	desu9
<i>Troubleshooting Guide</i>	Dieses Handbuch hilft Ihnen bei der Bestimmung von Fehlerquellen, bei der Fehlerbehebung sowie bei der Verwendung von Diagnose-Tools, wenn Sie den DB2-Kundendienst in Anspruch nehmen.	GC09-2850 db2p0e70	db2p0
<i>Neue Funktionen</i>	Dieses Handbuch enthält eine Beschreibung der neuen Einrichtungen, Funktionen und Erweiterungen in DB2 Universal Database Version 7.	SC12-2883 db2q0g70	db2q0
DB2-Installations- und -Konfigurationsinformationen			
<i>DB2 Connect Enterprise Edition für OS/2 und Windows Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation und Konfiguration für DB2 Connect Enterprise Edition unter OS/2 und 32-Bit-Windows-Betriebssystemen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2863 db2c6g70	db2c6

Tabelle 30. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>DB2 Connect Enterprise Edition für UNIX Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation, Konfiguration und Ausführung von Tasks für DB2 Connect Enterprise Edition auf UNIX-Plattformen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2862 db2cyg70	db2cy
<i>DB2 Connect Personal Edition Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation, Konfiguration und Ausführung von Tasks für DB2 Connect Personal Edition unter OS/2 und 32-Bit-Windows-Betriebssystemen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für alle unterstützten Clients.	GC12-2869 db2c1g70	db2c1
<i>DB2 Connect Personal Edition für Linux Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Migration und Konfiguration für DB2 Connect Personal Edition für alle unterstützten Linux-Varianten.	GC12-2865 db2c4g70	db2c4
<i>DB2 Data Links Manager Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Konfiguration und Ausführung von Tasks für DB2 Data Links Manager unter AIX und 32-Bit-Windows-Betriebssystemen.	GC12-2868 db2z6g70	db2z6
<i>DB2 Enterprise - Extended Edition für UNIX Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation und Konfiguration für DB2 Enterprise - Extended Edition auf UNIX-Plattformen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2867 db2v3g70	db2v3

Tabelle 30. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>DB2 Enterprise - Extended Edition für Windows Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation und Konfiguration für DB2 Enterprise - Extended Edition unter 32-Bit-Windows-Betriebssystemen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2866 db2v6g70	db2v6
<i>DB2 für OS/2 Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation und Konfiguration von DB2 Universal Database für das Betriebssystem OS/2. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2870 db2i2g70	db2i2
<i>DB2 für UNIX Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation und Konfiguration von DB2 Universal Database auf UNIX-Plattformen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2872 db2ixg70	db2ix
<i>DB2 für Windows Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Migration und Konfiguration für DB2 Universal Database unter 32-Bit-Windows-Betriebssystemen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2873 db2i6g70	db2i6
<i>DB2 Personal Edition Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Migration und Konfiguration für DB2 Universal Database Personal Edition unter OS/2 und 32-Bit-Windows-Betriebssystemen.	GC12-2871 db2i1g70	db2i1
<i>DB2 Personal Edition für Linux Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Migration und Konfiguration für DB2 Universal Database Personal Edition für alle unterstützten Linux-Varianten.	GC12-2874 db2i4g70	db2i4

Tabelle 30. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>DB2 Query Patroller Installation Guide</i>	Dieses Handbuch enthält Installationsinformationen zu DB2 Query Patroller.	GC09-2959 db2iwe70	db2iw
<i>DB2 Warehouse Manager Installation</i>	Dieses Handbuch enthält Installationsinformationen für Warehouse-Agenten, Warehouse- Umsetzungsprogramme und den Information Catalog Manager.	GC12-2876 db2ide70	db2id
Plattformübergreifende Beispielprogramme in HTML			
Beispielprogramme in HTML	Dieses Handbuch enthält die Beispiel- programme für die Programmier- sprachen auf allen von DB2 unterstützten Plattformen im HTML- Format. Die Beispielprogramme werden lediglich zu Informationszwecken zur Verfügung gestellt. Nicht alle Beispiele sind für alle Programmiersprachen verfügbar. Die HTML-Beispiele stehen nur dann zur Verfügung, wenn der DB2 Application Development Client installiert ist. Weitere Informationen zu den Programmen finden Sie im Handbuch <i>Application Building Guide</i> .	Keine Form- nummer	db2hs
Release-Informationen			
<i>DB2 Connect Release- Informationen</i>	Dieses Dokument enthält die neuesten Informationen, die in die DB2 Connect- Handbücher nicht mehr aufgenom- men werden konnten.	Siehe Anmerkung 2.	db2cr
<i>DB2 Installationsinformationen</i>	Dieses Dokument enthält die neuesten Informationen zur Installation, die in die DB2-Handbücher nicht mehr aufge- nommen werden konnten.	Nur auf der Produkt-CD- ROM verfügbar.	
<i>DB2-Release-Informationen</i>	Dieses Dokument enthält die neuesten Informationen zu allen DB2-Produkten und -Funktionen, die in die DB2- Handbücher nicht mehr aufgenommen werden konnten.	Siehe Anmerkung 2.	db2ir

Anmerkungen:

1. Das Zeichen an der sechsten Stelle des Dateinamens gibt die Landessprache eines Buchs an. So kennzeichnet der Dateiname db2d0e70 die englische Version des Handbuchs *Systemverwaltung*, der Dateinamen db2d0f70 kennzeichnet die französische Version des Buchs. Folgende Buchstaben werden an der sechsten Stelle des Dateinamens verwendet, um die Landessprache für ein Handbuch anzugeben:

Sprache	Kennung
Brasilianisches Portugiesisch	b
Bulgarisch	u
Dänisch	d
Deutsch	g
Englisch	e
Finnisch	y
Französisch	f
Griechisch	a
Italienisch	i
Japanisch	j
Koreanisch	k
Niederländisch	q
Norwegisch	n
Polnisch	p
Portugiesisch	v
Russisch	r
Schwedisch	s
Slowenisch	l
Spanisch	z
Trad. Chinesisch	t
Tschechisch	x
Türkisch	m
Ungarisch	h
Vereinf. Chinesisch	c

2. Kurzfristig verfügbare Informationen, die in die DB2-Handbücher nicht mehr aufgenommen werden können, sind in den Release-Informationen enthalten, die im HTML-Format und als ASCII-Datei verfügbar sind. Die HTML-Version steht über 'Information - Unterstützung' und auf den Produkt-CD-ROMs zur Verfügung. Gehen Sie wie folgt vor, um die ASCII-Dateien anzuzeigen:
 - Rufen Sie auf UNIX-Plattformen die Datei `Release.Notes` auf. Diese Datei befindet sich im Verzeichnis `DB2DIR/Readme/%L`. Dabei ist `%L` die länderspezifische Angabe und `DB2DIR` eine der folgenden Angaben:
 - `/usr/lpp/db2_07_01` (unter AIX)
 - `/opt/IBMdb2/V7.1` (unter HP-UX, PTX, Solaris und Silicon Graphics IRIX)
 - `/usr/IBMdb2/V7.1` (unter Linux)
 - Rufen Sie auf anderen Plattformen die Datei `RELEASE.TXT` auf. Diese Datei befindet sich in dem Verzeichnis, in dem das Produkt installiert ist. Auf OS/2-Plattformen können Sie auch den Ordner **IBM DB2** und anschließend das Symbol **Release-Informationen** doppelt anklicken.

Drucken der PDF-Handbücher

Wenn Sie eine gedruckte Version der Handbücher bevorzugen, können Sie die PDF-Dateien auf der CD-ROM mit DB2-Veröffentlichungen ausdrucken. Mit Adobe Acrobat Reader können Sie entweder das gesamte Handbuch oder bestimmte Teile des Handbuchs ausdrucken. Die Namen der einzelnen Handbücher in der Bibliothek finden Sie in Tabelle 30 auf Seite 377.

Die neueste Version von Adobe Acrobat Reader finden Sie auf der Adobe-Web-Site unter <http://www.adobe.com>.

Die PDF-Dateien befinden sich auf der CD-ROM mit DB2-Veröffentlichungen und haben die Dateierweiterung PDF. Führen Sie folgende Schritte aus, um auf die PDF-Dateien zuzugreifen:

1. Legen Sie die CD-ROM mit DB2-Veröffentlichungen in das CD-ROM-Laufwerk ein. Auf UNIX-Plattformen: Hängen Sie die CD-ROM mit den DB2-Veröffentlichungen an. Das Handbuch *Einstieg* enthält Anweisungen zu den Mount-Prozeduren.
2. Starten Sie Acrobat Reader.

3. Öffnen Sie die gewünschte PDF-Datei von einer der folgenden Positionen aus:

- Auf OS/2- und Windows-Plattformen:

Verzeichnis *x:\doc\sprache*. Dabei gibt *x* das CD-ROM-Laufwerk an, *sprache* den zweistelligen Landescode für die verwendete Sprache (z. B. EN für Englisch).

- Auf UNIX-Plattformen:

Verzeichnis */cdrom/doc/%L* auf der CD-ROM. Dabei gibt */cdrom* den Mount-Punkt der CD-ROM an, *%L* den Namen der gewünschten länderspezifischen Angaben.

Sie können die PDF-Dateien auch von der CD-ROM in ein lokales Laufwerk oder ein Netzlaufwerk kopieren und sie von dort aus lesen.

Bestellen der gedruckten Handbücher

Sie können die gedruckten DB2-Handbücher einzeln bestellen. In den USA und Kanada ist es außerdem möglich, mehrere Bücher als Paket unter einer SBOF-Nummer zu bestellen. Setzen Sie sich mit Ihrem IBM Vertragshändler oder Vertriebsbeauftragten in Verbindung, oder bestellen Sie die Handbücher telefonisch bei IBM Direkt unter der Nummer 0180/55 090. Darüber hinaus können Sie die Handbücher über die Web-Seite mit Veröffentlichungen unter <http://www.elink.ibm.com/pbl/pbl> bestellen.

Es sind zwei Gruppen von Handbüchern verfügbar. Die Gruppe mit der Formnummer SBOF-8935 umfaßt Referenzinformationen und Informationen zur Verwendung für DB2 Warehouse Manager. Die Gruppe mit der Formnummer SBOF-8931 umfaßt Referenzinformationen und Informationen zur Verwendung für alle anderen DB2 Universal Database-Produkte und -Funktionen. Der Inhalt der SBOF-Gruppen ist in der folgenden Tabelle aufgeführt.

Tabelle 31. Bestellen der gedruckten Handbücher

SBOF-Nummer	In dieser Gruppe enthaltene Handbücher	
SBOF-8931	<ul style="list-style-type: none"> • Administration Guide: Planning • Administration Guide: Implementation • Administration Guide: Performance • Administrative API Reference • Application Building Guide • Application Development Guide • CLI Guide and Reference • Command Reference • Data Movement Utilities Guide and Reference • Data Warehouse Center Administration Guide • Data Warehouse Center Application Integration Guide • DB2 Connect User's Guide • Installation and Configuration Supplement • Image, Audio, and Video Extenders Administration and Programming • Message Reference, Volumes 1 and 2 	<ul style="list-style-type: none"> • OLAP Integration Server Administration Guide • OLAP Integration Server Metaoutline User's Guide • OLAP Integration Server Model User's Guide • OLAP Integration Server User's Guide • OLAP Setup and User's Guide • OLAP Spreadsheet Add-in User's Guide for Excel • OLAP Spreadsheet Add-in User's Guide for Lotus 1-2-3 • Replication Guide and Reference • Spatial Extender Administration and Programming Guide • SQL Getting Started • SQL Reference, Volumes 1 and 2 • System Monitor Guide and Reference • Text Extender Administration and Programming • Troubleshooting Guide • What's New
SBOF-8935	<ul style="list-style-type: none"> • Information Catalog Manager Administration Guide • Information Catalog Manager User's Guide • Information Catalog Manager Programming Guide and Reference 	<ul style="list-style-type: none"> • Query Patroller Administration Guide • Query Patroller User's Guide

Zugreifen auf die Online-Hilfefunktion

Die Online-Hilfefunktion ist für alle DB2-Komponenten verfügbar. In der folgenden Tabelle werden die verschiedenen Hilfearten beschrieben.

Hilfearten	Inhalt	Zugriff
<i>Hilfe für Befehl</i>	Erklärt die Syntax von Befehlen im Befehlszeilenprozessor.	Geben Sie im interaktiven Modus des Befehlszeilenprozessors folgendes ein: ? <i>befehl</i> Dabei stellt <i>befehl</i> ein Schlüsselwort bzw. den vollständigen Befehl dar. So kann beispielsweise durch die Eingabe von ? catalog Hilfe für alle CATALOG-Befehle angezeigt werden, während mit ? catalog database lediglich Hilfe für den Befehl CATALOG DATABASE angezeigt wird.
Hilfe für Client-Konfiguration - Unterstützung	Erläutert die Tasks, die Sie in einem Fenster oder Notizbuch ausführen können. Die Hilfe umfaßt Übersichtsinformationen und unbedingt erforderliche Informationen sowie eine Beschreibung zur Verwendung der Steuerelemente im Fenster oder Notizbuch.	Klicken Sie in einem Fenster oder in einem Notizbuch den Druckknopf Hilfe an oder drücken Sie die Taste F1 .
<i>Hilfe für die Befehlszentrale</i>		
<i>Hilfe für die Steuerzentrale</i>		
<i>Hilfe für die Data Warehouse-Zentrale</i>		
Hilfe für Event Analyzer		
<i>Hilfe für Information Catalog Manager</i>		
<i>Hilfe für die Satellitenverwaltungszentrale</i>		
<i>Hilfe für die Prozedurenzentrale</i>		

Hilfearten	Inhalt	Zugriff
<i>Nachrichtenhilfe</i>	Beschreibt die Ursache von Nachrichten sowie die auszuführenden Benutzeraktionen.	<p>Geben Sie im interaktiven Modus des Befehlszeilenprozessors folgendes ein:</p> <pre>? XXXnnnnn</pre> <p>Dabei ist <i>XXXnnnnn</i> eine gültige Nachrichtenennung.</p> <p>Bei Eingabe von ? SQL30081 wird z. B. die Hilfe zur Nachricht SQL30081 angezeigt.</p> <p>Wenn Sie die Nachrichtenhilfe seitenweise anzeigen möchten, geben Sie den folgenden Befehl ein:</p> <pre>? XXXnnnnn more</pre> <p>Geben Sie folgenden Befehl ein, um die Nachrichtenhilfe in einer Datei zu speichern:</p> <pre>? XXXnnnnn > datei.erw</pre> <p>Dabei ist <i>datei.erw</i> die Datei, in der Sie die Nachrichtenhilfe speichern möchten.</p>
<i>Hilfe für SQL</i>	Erklärt die Syntax von SQL-Anweisungen.	<p>Geben Sie im interaktiven Modus des Befehlszeilenprozessors folgendes ein:</p> <pre>help anweisung</pre> <p>Dabei gibt <i>anweisung</i> eine SQL-Anweisung an.</p> <p>So kann beispielsweise durch die Eingabe von <code>help SELECT</code> die Hilfe zur Anweisung <code>SELECT</code> angezeigt werden.</p> <p>Anmerkung: Die Hilfe für SQL ist auf UNIX-Plattformen nicht verfügbar.</p>
<i>SQLSTATE-Hilfe</i>	Erklärt SQLSTATE-Werte und SQL-Klassencodes.	<p>Geben Sie im interaktiven Modus des Befehlszeilenprozessors folgendes ein:</p> <pre>? sqlstate oder ? klassencode</pre> <p>Datei ist <i>sqlstate</i> ein gültiger, fünfstelliger SQL-Status, und <i>klassencode</i> stellt die ersten zwei Ziffern des SQL-Statuswerts dar.</p> <p>So kann beispielsweise durch die Eingabe von ? 08003 Hilfe für den SQL-Statuswert 08003 angezeigt werden, während mit ? 08 Hilfe für den Klassencode 08 angezeigt wird.</p>

Anzeigen von Online-Informationen

Die zum Lieferumfang dieses Produkts gehörenden Handbücher werden als Softcopy im HTML-Format (HTML - Hypertext Markup Language) bereitgestellt. In einer Softcopy können Sie die Informationen auf einfache Art suchen und anzeigen und über Hypertextverbindungen auf zugehörige Informationen zugreifen. Außerdem wird die gemeinsame Nutzung der Bibliothek in Ihrem gesamten Unternehmen erleichtert.

Sie können die Online-Bücher und Beispielprogramme mit jedem Browser anzeigen, der den Spezifikationen von HTML Version 3.2 entspricht.

Führen Sie die nachfolgend beschriebenen Schritte aus, um Online-Bücher oder Beispielprogramme anzuzeigen:

- Wenn Sie DB2-Verwaltungs-Tools ausführen, verwenden Sie **Information - Unterstützung**.
- Klicken Sie in einem Browser **Datei**—>**Seite öffnen** an. Die geöffnete Seite enthält eine Übersicht über die DB2-Informationen und Verbindungen (Links) zu diesen Informationen:
 - Öffnen Sie auf UNIX-Plattformen die folgende Seite:

```
INSTHOME/sql11ib/doc/%L/html/index.htm
```

Dabei ist %L die länderspezifische Angabe.

- Öffnen Sie auf anderen Plattformen die folgende Seite:

```
sql11ib\doc\html\index.htm
```

Der Pfad befindet sich auf dem Laufwerk, auf dem DB2 installiert ist.

Wenn Sie **Information - Unterstützung** nicht installiert haben, können Sie die Seite öffnen, indem Sie das Symbol **DB2-Informationen** doppelt anklicken. Je nach verwendetem Betriebssystem befindet sich das Symbol im Hauptproduktordner bzw. unter Windows im Menü **Start**.

Installieren des Netscape-Browsers

Wenn Sie nicht bereits einen Web-Browser installiert haben, können Sie Netscape von der im Lieferumfang des Produkts enthaltenen Netscape-CD-ROM aus installieren. Führen Sie folgende Schritte aus, um ausführliche Informationen zur Installation zu erhalten:

1. Legen Sie die Netscape-CD-ROM ein.
2. Nur auf UNIX-Plattformen: Hängen Sie die CD-ROM an. Das Handbuch *Einstieg* enthält Anweisungen zu den Mount-Prozeduren.
3. Installationsanweisungen finden Sie in der Datei `CDNAVnn.txt`. Dabei ist *nn* die zweistellige Landeskennung. Die Datei befindet sich im Stammverzeichnis der CD-ROM.

Zugreifen auf Informationen mit "Information - Unterstützung"
Information - Unterstützung ermöglicht Ihnen den schnellen Zugriff auf DB2-Produktinformationen. **Information - Unterstützung** ist auf allen Plattformen mit DB2-Verwaltungs-Tools verfügbar.

Sie können 'Information - Unterstützung' öffnen, indem Sie das entsprechende Symbol doppelt anklicken. Abhängig vom verwendeten System befindet sich das Symbol im Hauptproduktordner im Ordner 'Information' bzw. unter Windows im Menü **Start**.

Sie können auf 'Information - Unterstützung' auch zugreifen, indem Sie die Funktionsleiste und das Menü **Hilfe** auf der DB2-Windows-Plattform verwenden.

Unter 'Information - Unterstützung' finden Sie sechs verschiedene Arten von Informationen. Klicken Sie die entsprechende Indexzunge an, um die für diese Informationsart verfügbaren Themen aufzurufen.

Funktionen Die Hauptfunktionen, die Sie mit DB2 ausführen können.

Referenz DB2-Referenzinformationen, wie beispielsweise Schlüsselwörter, Befehle und APIs.

Handbücher DB2-Handbücher.

Fehlerbehebung

Kategorien von Fehlermeldungen sowie die entsprechenden Benutzeraktionen.

Beispielprogramme

Beispielprogramme, die in DB2 Application Development Client enthalten sind. Wenn Sie DB2 Application Development Client nicht installiert haben, wird diese Indexzunge nicht angezeigt.

Web DB2-Informationen im World Wide Web. Sie müssen über Ihr System eine Verbindung zum Web herstellen können, um auf diese Informationen zugreifen zu können.

Wenn Sie einen Eintrag aus einer der Listen auswählen, startet **Information - Unterstützung** eine Funktion zum Anzeigen der Informationen. Bei der Anzeigefunktion kann es sich abhängig von der ausgewählten Informationsart um die Hilfeanzeige des Systems, einen Editor oder einen Web-Browser handeln.

In 'Information - Unterstützung' steht eine Suchfunktion zur Verfügung, mit der Sie nach einem bestimmten Thema suchen können, ohne in den Listen blättern zu müssen.

Rufen Sie über die Hypertextverbindung in 'Information - Unterstützung' das Suchformular **In DB2-Online-Informationen suchen** auf.

Der HTML-Such-Server wird normalerweise automatisch gestartet. Wenn eine Suche in HTML-Informationen fehlschlägt, müssen Sie möglicherweise mit einer der nachfolgend aufgeführten Methoden den Such-Server starten:

Unter Windows

Klicken Sie **Start** an und wählen Sie **Programme** —> **IBM DB2** —> **Informationen** —> **HTML-Such-Server starten** aus.

Unter OS/2

Klicken Sie den Ordner **DB2 für OS/2** und anschließend das Symbol für **HTML-Such-Server starten** doppelt an.

Falls andere Probleme bei der Suche in HTML-Informationen auftreten, finden Sie möglicherweise entsprechende Hinweise in den Release-Informationen.

Anmerkung: Die Suchfunktion steht in Linux-, PTX- und Silicon Graphics IRIX-Umgebungen nicht zur Verfügung.

Verwenden der DB2-Assistenten

Assistenten unterstützen Sie bei der Ausführung bestimmter Verwaltungsaufgaben, indem sie Sie Schritt für Schritt durch jede Aufgabe führen. Assistenten stehen über die Steuerzentrale und 'Client-Konfiguration - Unterstützung' zur Verfügung. In der folgenden Tabelle sind die einzelnen Assistenten und deren Verwendungszweck aufgeführt.

Anmerkung: In Umgebungen mit partitionierten Datenbanken sind die Assistenten **Datenbank erstellen**, **Index erstellen**, **Aktualisierung auf mehreren Systemen konfigurieren** und **Leistungskonfiguration** verfügbar.

Assistent	Verwendung	Zugriff
<i>Datenbank hinzufügen</i>	Katalogisieren einer Datenbank auf einer Client-Workstation.	Klicken Sie in Client-Konfiguration - Unterstützung die Option Hinzufügen an.
<i>Datenbank sichern</i>	Festlegen, Erstellen und Terminieren eines Sicherungsplans.	Klicken Sie in der Steuerzentrale die zu sichernde Datenbank mit der rechten Maustaste an und wählen Sie Sichern —> Datenbank mit Assistent aus.
<i>Aktualisierung auf mehreren Systemen konfigurieren</i>	Konfigurieren einer Aktualisierung auf mehreren Systemen, einer verteilten Transaktion oder einer zweiphasigen Fest-schreibung.	Klicken Sie in der Steuerzentrale den Ordner Datenbanken mit der rechten Maustaste an und wählen Sie Aktualisierung auf mehreren Systemen aus.

Assistent	Verwendung	Zugriff
<i>Datenbank erstellen</i>	Erstellen einer Datenbank und Ausführen einiger grundlegender Konfigurationsfunktionen.	Klicken Sie in der Steuerzentrale den Ordner Datenbanken mit der rechten Maustaste an und wählen Sie Erstellen —> Datenbank mit Assistent aus.
<i>Tabelle erstellen</i>	Auswählen eines Basisdatentyps und Erstellen eines Primärschlüssels für die Tabelle.	Klicken Sie in der Steuerzentrale das Symbol Tabellen mit der rechten Maustaste an und wählen Sie Erstellen —> Tabelle mit Assistent aus.
<i>Tabellenbereich erstellen</i>	Erstellen eines neuen Tabellenbereichs.	Klicken Sie in der Steuerzentrale das Symbol Tabellenbereiche mit der rechten Maustaste an und wählen Sie Erstellen —> Tabellenbereich mit Assistent aus.
<i>Index erstellen</i>	Hinweise zum Erstellen und Löschen von Indizes für Ihre Abfragen.	Klicken Sie in der Steuerzentrale das Symbol Index mit der rechten Maustaste an und wählen Sie Erstellen —> Index mit Assistent aus.
<i>Leistungskonfiguration</i>	Optimieren der Leistung einer Datenbank durch Aktualisieren der Konfigurationsparameter, so daß sie den Anforderungen Ihres Unternehmens entsprechen.	Klicken Sie in der Steuerzentrale die Datenbank, die optimiert werden soll, mit der rechten Maustaste an und wählen Sie Leistung mit Assistent konfigurieren aus. Klicken Sie in einer Umgebung mit partitionierten Datenbanken in der Sicht für Datenbankpartitionen die erste Datenbankpartition, die optimiert werden soll, mit der rechten Maustaste an und wählen Sie Leistung mit Assistent konfigurieren aus.
<i>Datenbank wiederherstellen</i>	Wiederherstellen einer Datenbank nach einem Fehler. Dieser Assistent hilft Ihnen, zu entscheiden, welche Sicherungskopie Sie verwenden und welche Protokolle Sie erneut abarbeiten.	Klicken Sie in der Steuerzentrale die Datenbank, die wiederhergestellt werden soll, mit der rechten Maustaste an und wählen Sie Wiederherstellen —> Datenbank mit Assistent aus.

Einrichten eines Dokument-Servers

Die DB2-Informationen werden standardmäßig auf Ihrem lokalen System installiert. Das bedeutet, daß alle Benutzer, die Zugriff auf DB2-Informationen benötigen, dieselben Dateien installieren müssen. Führen Sie folgende Schritte aus, um die DB2-Informationen an einer einzigen Position zu speichern:

1. Kopieren Sie alle Dateien und Unterverzeichnisse aus dem Verzeichnis `\sql11ib\doc\html` Ihres lokalen Systems auf einen Web-Server. Jedem Handbuch ist ein Unterverzeichnis zugeordnet, das alle erforderlichen HTML- und GIF-Dateien enthält, aus denen das Handbuch besteht. Stellen Sie sicher, daß die Verzeichnisstruktur erhalten bleibt.
2. Konfigurieren Sie den Web-Server so, daß er die Dateien an der neuen Speicherposition sucht. Informationen hierzu finden Sie im Anhang zu NetQuestion im Handbuch *DB2 Installation und Konfiguration Ergänzung*.
3. Wenn Sie die Java-Version von **Information - Unterstützung** verwenden, können Sie eine Basis-URL-Adresse für alle HTML-Dateien angeben. Sie sollten die URL-Adresse für das Bücherverzeichnis verwenden.
4. Wenn Sie die Buchdateien anzeigen können, ist es möglich, bei häufig aufgerufenen Themen Lesezeichen zu setzen. Es empfiehlt sich, folgende Seiten mit einem Lesezeichen zu versehen:
 - Bücherverzeichnis
 - Inhaltsverzeichnis häufig verwendeter Handbücher
 - Themen, auf die häufig verwiesen wird, wie beispielsweise zum Ändern von Tabellen
 - Suchformular

Informationen dazu, wie Sie die DB2 Universal Database-Online-Dokumentationsdateien auf einer zentralen Maschine zur Verfügung stellen können, finden Sie im Anhang zu NetQuestion im Handbuch *DB2 Installation und Konfiguration Ergänzung*.

Suchen nach Online-Informationen

Verwenden Sie eine der folgenden Methoden, um nach Informationen in den HTML-Dateien zu suchen:

- Klicken Sie im obersten Rahmen auf **Suchen**. Verwenden Sie das Suchformular, um nach einem bestimmten Thema zu suchen. Diese Funktion steht in Linux-, PIX- oder Silicon Graphics IRIX-Umgebungen nicht zur Verfügung.
- Klicken Sie im obersten Rahmen auf **Index**. Mit Hilfe des Indexes können Sie nach einem bestimmten Thema im Buch suchen.
- Rufen Sie das Inhaltsverzeichnis oder den Index der Hilfe oder des HTML-Buchs auf und verwenden Sie die Suchfunktion des Web-Browsers, um nach einem bestimmten Thema im Buch zu suchen.
- Mit Hilfe der Lesezeichenfunktion des Web-Browsers können Sie schnell zu einem bestimmten Thema zurückkehren.
- Mit Hilfe der Suchfunktion von **Information - Unterstützung** können Sie bestimmte Themen suchen. Weitere Informationen finden Sie in „Zugreifen auf Informationen mit "Information - Unterstützung"“ auf Seite 394.

Anhang B. Namenskonventionen

Beachten Sie die in diesem Anhang beschriebenen Regeln bei der Vergabe von Namen für die folgenden Datenbanken und Datenbankobjekte:

- Datenbanknamen
- Namen und Aliasnamen für Datenbanken
- Benutzer-IDs und Kennwörter
- Schemennamen
- Gruppen- und Benutzernamen
- Objektnamen

Verwenden Sie zur Benennung von Tabellen, Sichten, Spalten, Indizes und Berechtigungs-IDs keine für IBM SQL bzw. ISO/ANSI SQL92 reservierten Wörter. Eine Liste dieser Wörter finden Sie im Handbuch *SQL Reference*.

Lesen Sie die Informationen in den Handbüchern *Einstieg* zu den Namenskonventionen für Berechtigungs-IDs (einschließlich Benutzernamen und Gruppennamen) und Workstations sowie zu weiteren plattformspezifischen Einschränkungen.

Datenbanknamen

Jedesmal, wenn eine neue Datenbank erstellt wird, legt der Datenbankmanager ein separates Verzeichnis an, um die Steuerdateien und Datendateien für diese Datenbank zu speichern.

Das Benennungsschema für diese Verzeichnisse hat die Form SQL00001 - SQLnnnnn, wobei SQL00001 Steuerdateien enthält, die zur ersten erstellten Datenbank gehören, SQL00002 Steuerdateien, die zur zweiten erstellten Datenbank gehören, und so fort.

Diese Verzeichnisse werden automatisch verwaltet. Daher sollten Sie, um potentielle Probleme mit Verzeichnisnamen zu vermeiden, keine eigenen Verzeichnisse mit demselben Benennungsschema erstellen, und Verzeichnisse, die bereits vom Datenbankmanager erstellt wurden, nicht manuell bearbeiten.

Namen und Aliasnamen für Datenbanken

Datenbanknamen sind identifizierende Namen, die Sie oder Ihre Benutzer als Teil des Befehls CREATE DATABASE oder der gleichnamigen API angeben. Diese Namen müssen innerhalb der Position, in der sie katalogisiert werden, eindeutig sein. In auf UNIX basierenden Implementierungen von DB2 ist diese Position zum Beispiel ein Verzeichnispfad, während in OS/2-Implementierungen diese Position ein Laufwerkbuchstabe ist.

Aliasnamen für Datenbanken sind lokale Synonymnamen, die lokalen oder fernen Datenbanken zugeordnet werden. Diese Namen müssen innerhalb des Systemdatenbankverzeichnisses, in dem alle Aliasnamen für das spezifische Exemplar des Datenbankmanagers gespeichert werden, eindeutig sein. Wenn eine neue Datenbank erstellt wird, ist der Standardwert des Aliasnamens der Datenbankname. Daher können Sie keine Datenbank mit einem Namen erstellen, der bereits als Datenbankaliasname verwendet wird, selbst wenn es keine andere Datenbank mit diesem Namen gibt.

Bei der Festlegung eines Namens oder Aliasnamens für eine Datenbank gelten folgende Regeln:

- Die zulässige Länge ist 1 bis 8 Zeichen.
- Das erste Zeichen muß eines der folgenden Zeichen sein:
 - A - Z (Kleinbuchstaben werden in Großbuchstaben umgesetzt)
 - @, # oder \$
- Weitere Zeichen können sein:
 - A - Z (Kleinbuchstaben werden in Großbuchstaben umgesetzt)
 - 0 - 9
 - @, #, \$ und _ (Unterstreichungszeichen)

Anmerkung: Zur Vermeidung von Problemen sollten Sie die Sonderzeichen @, # und \$ nicht in einem Datenbanknamen benutzen, wenn Sie beabsichtigen, die Datenbank in einer Übertragungsumgebung zu verwenden. Darüber hinaus sollten Sie diese Zeichen nicht benutzen, wenn Sie die Datenbank in einem anderen Land verwenden möchten, weil diese Zeichen nicht auf allen Tastaturen in gleicher Weise verfügbar sind. Auf Systemen unter Windows NT ist außerdem sicherzustellen, daß der Exemplarname nicht mit einem Servicenamen identisch ist.

Benutzer-IDs und Kennwörter

Bei der Erstellung einer Benutzer-ID oder eines Kennworts gelten folgende Regeln:

- Nicht verwendet werden dürfen:

- USERS, ADMINS, GUESTS, PUBLIC, LOCAL oder eines der im Handbuch *SQL Reference* aufgeführten, für SQL reservierten Wörter.
- Erste Zeichen dürfen nicht sein:
 - SQL, SYS oder IBM
- Weitere Zeichen können sein:
 - A - Z

Anmerkung: Bei einigen Betriebssystemen sind Benutzer-IDs und Kennwörter mit Unterscheidung von Groß-/Kleinschreibung möglich. Prüfen Sie die Dokumentation zu Ihrem Betriebssystem, um festzustellen, ob dies auf Ihr Betriebssystem zutrifft.

- 0 - 9
- @, # oder \$
- Benutzer-IDs dürfen eine Länge von 30 Zeichen nicht überschreiten.

Anmerkung: Sie müssen möglicherweise Aufgaben zur Kennwortverwaltung ausführen. Da solche Aufgaben auf dem Server auszuführen sind und viele Benutzer mit Arbeit in der Server-Umgebung nicht vertraut sind oder sich dabei nicht wohl fühlen, kann diese Tätigkeit erhebliche Anstrengungen erfordern. DB2 UDB stellt eine Möglichkeit zur Aktualisierung und Überprüfung von Kennwörtern ohne Arbeiten auf dem Server zur Verfügung. Zum Beispiel unterstützt DB2 für OS/390 Version 5 diese Methode, das Kennwort eines Benutzers zu ändern. Wenn eine Fehlermeldung SQL1404N „Kennwort abgelaufen“ empfangen wird, ändern Sie mit der Anweisung CONNECT das Kennwort auf folgende Weise:

```
CONNECT TO <datenbank> USER <benutzer-id> USING <kennwort>
NEW <neues_kennwort> VERIFY <neues_kennwort>
```

Das Dialogfenster „Kennwort ändern“ der DB2-Client-Konfiguration - Unterstützung kann ebenfalls zur Änderung des Kennworts verwendet werden. Weitere Informationen zu diesen Methoden zur Änderung des Kennworts finden Sie im Handbuch *SQL Reference* und in der Online-Hilfefunktion der DB2-Client-Konfiguration - Unterstützung.

Schemennamen

Die folgenden Schemennamen sind reservierte Wörter und dürfen nicht verwendet werden:

- SYSCAT
- SYSFUN

- SYSIBM
- SYSSTAT

Im allgemeinen sollten Sie darauf achten, keine Schemennamen zu verwenden, die mit der Zeichenfolge SYS beginnen, um zukünftigen Migrationsproblemen vorzubeugen. Der Datenbankmanager läßt die Erstellung von Auslösern, benutzerdefinierten Datentypen bzw. benutzerdefinierten Funktionen mit Schemennamen, die mit SYS beginnen, nicht zu.

Darüber hinaus wird empfohlen, das Wort SESSION nicht als Schemennamen zu verwenden. Deklarierte temporäre Tabellen müssen mit SESSION qualifiziert werden. Daher ist es möglich, daß eine Anwendung eine temporäre Datei mit einem Namen deklariert, der mit dem einer permanenten Tabelle identisch ist. In einem solchen Fall kann die Anwendungslogik übermäßig kompliziert werden. Außer im Zusammenhang mit deklarierten temporären Tabellen sollte die Verwendung des Schemennamens SESSION also vermieden werden.

Gruppen- und Benutzernamen

In auf UNIX basierenden Systemen können Gruppen und Benutzer denselben Namen haben. Bei der Anweisung GRANT müssen Sie angeben, ob eine Gruppe oder ein Benutzer gemeint ist. Bei der Anweisung REVOKE hängt die Angabe eines Benutzers oder einer Gruppe davon ab, ob für GRANTEE mehrere Zeilen in den Tabellen des Berechtigungskatalogs mit verschiedenen Werten für GRANTEETYPE enthalten sind.

Unter OS/2 dürfen Gruppen und Benutzer nicht dieselben Namen haben.

Unter Windows NT dürfen lokale und globale Gruppen und Benutzer-IDs nicht dieselben Namen haben.

Gruppennamen dürfen nicht länger als 8 Zeichen sein.

Objektnamen

Es gibt folgende Datenbankobjekte:

- Schemata
- Tabellen
- Sichten
- Spalten
- Indizes
- Benutzerdefinierte Funktionen (UDFs)
- Benutzerdefinierte Datentypen (UDTs)

- Auslöser
- Aliasnamen
- Tabellenbereiche
- Gespeicherte Prozeduren
- Methoden
- Knotengruppen
- Pufferpools
- Ereignismonitore

Bei der Festlegung eines Namens für ein Datenbankobjekt gelten folgende Regeln:

- Die zulässige Länge ist 1 - 18 Zeichen (Byte)

Anmerkung: Es gibt jedoch Ausnahmen:

- Für Schemata und Spalten sind 1 bis 30 Zeichen zulässig.
- Für Tabellen, Sichten, Korrelationsnamen und Aliasnamen sind 1 bis 128 Zeichen zulässig.
- Das erste Zeichen muß eines der folgenden Zeichen sein:
 - A - Z (Kleinbuchstaben werden in Großbuchstaben umgesetzt)
 - Gültiger Buchstabe mit diakritischem Zeichen (z. B. ö)
 - Ein Mehrbytezeichen außer Mehrbyteleerzeichen (für Mehrbyteumgebungen)
- Weitere Zeichen können sein:
 - A - Z (Kleinbuchstaben werden in Großbuchstaben umgesetzt)
 - Gültiger Buchstabe mit diakritischem Zeichen (z. B. ö)
 - 0 - 9
 - @, #, \$ und _ (Unterstreichungszeichen)
 - Ein Mehrbytezeichen außer Mehrbyteleerzeichen (für Mehrbyteumgebungen)

Schlüsselwörter dürfen verwendet werden. Bei Verwendung des Schlüsselworts in einem Kontext, in dem es auch als SQL-Schlüsselwort interpretiert werden könnte, muß es als begrenzter Bezeichner angegeben werden. Im Handbuch *SQL Reference* finden Sie weitere Informationen zu begrenzten Bezeichnern (Delimited Identifier).

Zur Erreichung der größtmöglichen Übertragbarkeit verwenden Sie die für IBM SQL und ISO/ANSI SQL92 reservierten Wörter. Eine Liste dieser Wörter finden Sie im Handbuch *SQL Reference*.

Anmerkungen:

1. Mit Hilfe der begrenzten Bezeichner ist es möglich, ein Objekt zu erstellen, dessen Name gegen diese Namenskonventionen verstößt. Jedoch können bei nachfolgender Verwendung eines solchen Objekts Fehler auftreten. Zur Vermeidung möglicher Fehler bei der Verwendung und dem Betrieb der Datenbank sollten Sie diese Namenskonventionen **nicht** verletzen.

Wenn Sie zum Beispiel eine Spalte mit einem Namen erstellen, in dem ein Pluszeichen (+) oder ein Minuszeichen (-) vorkommt, und Sie in der Folge diese Spalte in einem Index verwenden, treten Probleme auf, wenn Sie versuchen, die Tabelle zu reorganisieren.

Objektnamen zusammenschlossener Datenbanken

Es gibt folgende Objekte zusammenschlossener Datenbanken:

- Indexspezifikationen
- Kurznamen
- Server
- Oberflächen
- Funktionszuordnungen
- Typzuordnungen
- Benutzerzuordnungen

Für die Benennung von Objekten zusammenschlossener Datenbanken gelten Einschränkungen. Eine vollständige Liste von Objektnamen und den zugehörigen Kennungsbegrenzungen und -anforderungen ist im Handbuch *SQL Reference* enthalten. Zusammenfassend gelten folgende Regeln für Objektnamen:

- Sie sind begrenzt. Kurznamen sowie Namen für Zuordnungen, Indexspezifikationen, Server und Oberflächen dürfen nicht länger als 128 Byte sein.
- Das erste Zeichen muß eines der folgenden Zeichen sein:
 - A bis Z (Namen ohne Anführungszeichen werden in Großschreibung umgewandelt)
 - Gültiger Buchstabe mit diakritischem Zeichen (z. B. ö)
 - Ein Mehrbytezeichen außer Mehrbyteleerzeichen (für Mehrbyteumgebungen)
- Sie müssen internen Namenskonventionen entsprechen. Weitere Zeichen können sein:
 - A - Z
 - Gültiger Buchstabe mit diakritischem Zeichen (z. B. ö)
 - 0 - 9
 - @, #, \$ und _ (Unterstreichungszeichen)

- Ein Mehrbytezeichen außer Mehrbyteleerzeichen (für Mehrbyteumgebungen)

Schlüsselwörter dürfen verwendet werden. Bei Verwendung des Schlüsselworts in einem Kontext, in dem es auch als SQL-Schlüsselwort interpretiert werden könnte, muß es als begrenzter Bezeichner angegeben werden. Im Handbuch *SQL Reference* finden Sie weitere Informationen zu begrenzten Bezeichnern (Delimited Identifier).

Zur Erreichung der größtmöglichen Übertragbarkeit verwenden Sie die für IBM SQL und ISO/ANSI SQL92 reservierten Wörter. Eine Liste dieser Wörter finden Sie im Handbuch *SQL Reference*.

Optionen (Server, Kurzname) und Optionseinstellungen sind auf 255 Byte begrenzt.

Beibehaltung von Werten in Groß-/Kleinschreibung in einem System zusammengesetzter Datenbanken

In verteilten Anforderungen müssen Sie manchmal Kennungen und Kennwörter angeben, deren Groß-/Kleinschreibung auf der Datenquelle unterschieden wird. Beachten Sie die folgenden Richtlinien, um sicherzustellen, daß die Schreibweise beim Übergeben an die Datenquelle korrekt ist:

- Geben Sie sie in der erforderlichen Schreibweise (Groß-/Kleinschreibung) an, und setzen Sie sie in doppelte Anführungszeichen.
- Wenn Sie eine Benutzer-ID angeben, setzen Sie die Server-Option 'fold_id' für diese Datenquelle auf 'n' („Nein, Groß-/Kleinschreibung nicht ändern“). Wenn Sie ein Kennwort angeben, setzen Sie die Server-Option 'fold_pw' für diese Datenquelle auf 'n'.

Es gibt eine Alternative für Benutzer-IDs und Kennwörter. Wenn bei einer Datenquelle eine Benutzer-ID in Kleinbuchstaben angegeben werden muß, können Sie sie in beliebiger Schreibweise angeben und die Server-Option 'fold_id' auf 'l' („Diese ID in Kleinbuchstaben an die Datenquelle senden“) setzen. Wenn bei der Datenquelle die Benutzer-ID in Großbuchstaben angegeben werden muß, können Sie sie in beliebiger Schreibweise angeben und die Server-Option 'fold_id' auf 'u' („Diese ID in Großbuchstaben an die Datenquelle senden“) setzen. Auf dieselbe Art und Weise können Sie, wenn bei einer Datenquelle ein Kennwort in Klein- oder in Großbuchstaben angegeben werden muß, diese Anforderung erfüllen, indem Sie die Server-Option 'fold_pw' auf 'l' bzw. 'u' setzen.

Weitere Informationen zu Server-Optionen finden Sie in "Vereinfachen der Datenquellendefinitionen und der Authentifizierungsverarbeitung durch Server-Optionen" in *Systemverwaltung: Implementierung*.

- Wenn Sie eine Kennung oder ein Kennwort mit zu beachtender Groß-/Kleinschreibung in einer Eingabeaufforderung des Betriebssystems in

doppelte Anführungszeichen setzen, müssen Sie sicherstellen, daß das System die doppelten Anführungszeichen syntaktisch korrekt analysiert. Gehen Sie dazu wie folgt vor:

- Bei einem auf UNIX basierendem Betriebssystem schließen Sie die Anweisung in einfache Anführungszeichen ein.
- Beim Betriebssystem Windows NT stellen Sie jedem Anführungszeichen einen umgekehrten Schrägstrich voran.

Viele begrenzte Bezeichner in Datenquellen der DB2-Systemfamilie zum Beispiel unterscheiden Groß-/Kleinschreibung. Angenommen, Sie wollen einen Kurznamen KURZ1 für eine Sicht von DB2 für CS, "mein_schema"."wöch_geh", in einer Datenquelle namens NORBASE erstellen.

In die Eingabeaufforderung eines auf UNIX basierenden Systems würden Sie folgendes eingeben:

```
db2 'create nickname kurz1 for norbase."mein_schema"."wöch_geh"'
```

In eine Eingabeaufforderung von Windows NT würden Sie folgendes eingeben:

```
db2 create nickname nick1 for norbase.\"mein_schema\".\"wöch_geh\"
```

Wenn Sie die Anweisung an der Eingabeaufforderung des DB2-Dialogmodus oder in einem Anwendungsprogramm eingeben, sind die einfachen Anführungszeichen oder die Schrägstriche nicht notwendig. In die DB2-Eingabeaufforderung eines auf UNIX basierenden Systems oder eines Windows NT-Systems würden Sie folgendes eingeben:

```
create nickname kurz1 for norbase."mein_schema"."wöch_geh"
```

Anhang C. Planen der Datenbankmigration

Dieser Abschnitt enthält einen Überblick über den Migrationsprozeß. Beachten Sie, daß Datenbanken von DB2 UDB Version 6 nicht nach Version 7 migriert werden müssen. Detaillierte Informationen zur Migration von Datenbanken aus DB2 UDB Version 5.x finden Sie im Handbuch *Einstieg* zu Ihrem Betriebssystem.

Wenn Sie eine Datenbank migrieren:

- Folgende Bestandteile der Datenbank werden migriert:
 - Konfigurationsdatei der Datenbank
 - Systemkatalogtabellen der Datenbank
 - Datenbankverzeichnisse
 - Kopfsatz der Datenbankprotokolldatei
- Die Systemkatalogtabellen werden folgendermaßen geändert:
 - Neue Spalten werden hinzugefügt.
 - Neue Tabellen werden erstellt.
 - Eine Gruppe von Katalogsichten wird migriert, und eine Gruppe neuer Katalogsichten wird im Schema SYSCAT erstellt.
 - Eine Gruppe aktualisierbarer Katalogsichten wird im Schema SYSSTAT erstellt.
 - Eine Gruppe von Skalarfunktionen für allgemeine Zwecke wird behalten, und eine Gruppe neuer Skalarfunktionen für allgemeine Zwecke wird im Schema SYSFUN erstellt. Nur die Skalarfunktion SYSFUN.DIFFERENCE wird während der Datenbankmigration entfernt und neu erstellt.
- Eine Datenbankprotokolldatei wird mit zugehöriger Spiegelkopie im Datenbankverzeichnis erstellt. Diese Datei enthält eine Zusammenfassung der Sicherungsinformationen, die verwendet werden können, wenn eine Datenbank wiederhergestellt werden muß. Sie wird immer dann aktualisiert, wenn spezifische Operationen an der Datenbank ausgeführt werden. Eine Zusammenfassung von Sicherungsinformationen wird auch für Sicherungs- und Wiederherstellungsoperationen von Tabellenbereichen gespeichert.

Überlegungen zur Migration

Zur erfolgreichen Migration einer Datenbank, die mit einer vorherigen Version des Datenbankmanagers erstellt wurde, sind folgende Faktoren zu berücksichtigen:

- „Migrationseinschränkungen“ auf Seite 408

- „Sicherheit und Berechtigungen“
- „Speicherbedarf“
- „Durch Release-Wechsel bedingte Inkompatibilitäten“ auf Seite 409

Migrationseinschränkungen

Für die Migration einer Datenbank auf Version 7 gelten bestimmte Bedingungen bzw. Einschränkungen, die zu beachten sind:

- Eine Migration wird nur von Version 5.x oder Version 6 unterstützt. Die Migration von DB2 Version 1.2 Parallel Edition wird nicht unterstützt. Frühere Version von DB2 (Datenbankmanager) müssen zunächst auf Version 5.x oder V6 umgestellt werden, bevor sie auf Version 7 migriert werden können.
- Das Absetzen des Migrationsbefehls von einem Client der Version 7 zur Migration einer Datenbank auf einen Server der Version 7 wird unterstützt. Jedoch wird das Absetzen eines Migrationsbefehls von älteren DB2-Clients zur Migration einer Datenbank auf einen Server der Version 7 nicht unterstützt.
- Migration zwischen Plattformen wird nicht unterstützt.
- Benutzerobjekte innerhalb Ihrer Datenbank können keine für Version 7 reservierte Schemennamen als Objektqualifikationsmerkmale haben. Zu diesen reservierten Schemennamen gehören: SYSCAT, SYSSTAT und SYSFUN.
- Benutzerdefinierte einzigartige Datentypen mit den Namen BIGINT, REAL, DATALINK bzw. REFERENCE müssen vor der Migration der Datenbank umbenannt werden.
- Es ist nicht möglich, eine Datenbank zu migrieren, die sich in einem der folgenden Status befindet:
 - Sicherung anstehend
 - Aktualisierende Wiederherstellung anstehend
 - Einer oder mehrere Tabellenbereiche nicht im Normalstatus
 - Transaktion inkonsistent
- Die Wiederherstellung von Datenbanksicherungen früherer Versionen (Version 5.x oder Version 6) wird unterstützt. Die aktualisierende Wiederherstellung (ROLLFORWARD) von Protokollen früherer Versionen wird jedoch nicht unterstützt.

Sicherheit und Berechtigungen

Sie benötigen die Berechtigung SYSADM, um Ihre Datenbank migrieren zu können.

Speicherbedarf

Während der Migration wird sowohl für die alten als auch für die neuen Kataloge Speicherplatz benötigt. Die Menge des erforderlichen Platten-

speicherplatzes hängt von der Größe und Komplexität der Datenbank sowie von der Anzahl und der Größe der Datenbankobjekte ab. Zu diesen Objekten gehören sämtliche Tabellen und Sichten. Sie sollten mindestens das Doppelte des Plattenspeicherplatzes verfügbar machen, den der Datenbankkatalog momentan einnimmt. Reicht der Plattenspeicherplatz nicht aus, schlägt die Migration fehl.

Wenn Ihr SYSCAT-Tabellenbereich ein SMS-Tabellenbereich ist, sollten Sie außerdem in Betracht ziehen, die Konfigurationsparameter der Datenbank zu aktualisieren, die sich auf die Protokolldateien auswirken. Sie sollten die Werte der Parameter *logfilesiz*, *logprimary* und *logsecond* erhöhen, um zu verhindern, daß der Speicherbereich für die Dateien aufgebraucht wird (SQL1704N mit Ursachencode 3). In diesem Fall erhöhen Sie die Werte der Parameter für den Protokollspeicherbereich und setzen den Befehl `MIGRATE DATABASE` erneut ab.

Durch Release-Wechsel bedingte Inkompatibilitäten

Berücksichtigen Sie bei der Planung der Migration einer Datenbank die Auswirkungen der Inkompatibilitäten zwischen den beiden Versionen des Produkts.

Um die Erweiterungen von Version 7 nutzen zu können, sollten Sie Ihre Datenbank- und Datenbankmanagerkonfiguration nach der Migration Ihrer Datenbanken optimieren. Zur Vereinfachung dieser Optimierung können Sie die Werte der Konfigurationsparameter vor und nach der Migration aufzeichnen und vergleichen. (Eine Beschreibung der Befehle `GET DATABASE CONFIGURATION` und `GET DATABASE MANAGER CONFIGURATION` finden Sie im Handbuch *Command Reference*.)

Umstellen einer Datenbank

Die folgenden Schritte müssen zur Migration einer Datenbank ausgeführt werden. Bevor Sie mit der Migration beginnen, müssen Sie den Datenbankmanager starten.

PRÄMIGRATION:

Anmerkung: Die Schritte der Prämigration müssen an einem vorigen Release (d. h. an dem aktuellen Release vor dem Umstellen auf das neue Release bzw. vor der Installation des neuen Release) ausgeführt werden.

1. Stellen Sie sicher, daß es keine geklärten Punkte im Sinne der „Migrationseinschränkungen“ auf Seite 408 gibt.
2. Trennen Sie alle Anwendungen und Endbenutzer von jeder zu migrierenden Datenbank (verwenden Sie nach Bedarf den Befehl `LIST APPLICATIONS` oder `FORCE APPLICATIONS`).

3. Verwenden Sie das Prämigrationsdienstprogramm DB2CKMIG, um festzustellen, ob die Datenbank migriert werden kann (detaillierte Informationen zu diesem Dienstprogramm finden Sie im Handbuch *Einstieg* für Ihre Plattform). Beachten Sie, daß Sie unter Windows NT oder OS/2 aufgefordert werden, dieses Tool während der Installation auszuführen, während es auf UNIX-basierten Systemen automatisch bei der Exemplarmigration aufgerufen wird.
4. Erstellen Sie eine Sicherungskopie der Datenbank.

Die Migration ist ein Prozeß, der sich nicht rückgängig machen läßt. Wenn Sie Ihre Datenbank sichern, bevor die unter Version 6 reservierten Schemennamen geändert werden, sind Sie nicht in der Lage, die Datenbank mit Hilfe von DB2 UDB Version 7 wiederherzustellen. Zur Wiederherstellung der Datenbank müssen Sie Ihre frühere Version des Datenbankmanagers verwenden.

Achtung! Wenn Sie keine Sicherungskopie Ihrer Datenbank angelegt haben und der Migrationsversuch fehlschlägt, gibt es keine Möglichkeit, Ihre Datenbank mit Hilfe von DB2 UDB Version 7 oder der vorherigen Version Ihres Datenbankmanagers wiederherzustellen.

Beachten Sie außerdem, daß alle Datenbanktransaktionen, die im Zeitraum zwischen der Erstellung der Sicherung und der Migration auf Version 7 erfolgen, nicht wiederhergestellt werden können. Das bedeutet, daß, wenn zu einem Zeitpunkt nach der Beendigung der Installation und Migration auf Version 7 die Datenbank wiederhergestellt werden muß (auf eine Stufe der Version 7), die Protokolldateien, die vor der Installation von Version 7 geschrieben wurden, bei der aktualisierenden Wiederherstellung nicht verwendet werden können.

MIGRATION:

5. Migrieren Sie die Datenbank mit einer der folgenden Methoden:
 - Befehl MIGRATE DATABASE
 - Befehl RESTORE DATABASE bei der Wiederherstellung einer Gesamtsicherung der Datenbank
 - API sqlcmd - API zur Datenbankmigration

Unter OS/2: Das Migrationshilfsprogramm DB2CIDMG, das in einer Umgebung mit CID-Architektur (CID - Configuration/Installation/Distribution) ausführbar ist, steht nur unter DB2 für OS/2 zur Verfügung. Es ermöglicht eine ferne nichtüberwachte Installation und Konfiguration auf Workstations in einem LAN. Ihr LAN muß über NetView DM/2 verfügen, damit die CID-Migration verwendet werden kann.

Auf UNIX-basierten Systemen: Das Handbuch *Einstieg* für Ihre Plattform beschreibt, was zu tun ist, wenn Sie nicht alle Datenbanken in einem bestimmten Exemplar migrieren wollen.

POSTMIGRATION:

6. Verwenden Sie wahlfrei das Dienstprogramm DB2UDDL, das die Verwaltung einer gestaffelten Migration eindeutiger Indizes nach eigenem Zeitplan erleichtert. (DB2-Datenbanken, die in Version 5 erstellt wurden, benötigen dieses Tool nicht, um eine verzögerte Überprüfung auf Eindeutigkeit nutzen zu können, da alle eindeutigen Indizes, die in Version 5 erstellt wurden, über diese Semantik bereits verfügen. Jedoch ist für Datenbanken, die zuvor auf Version 5 migriert wurden, diese Semantik nicht automatisch, sofern Sie nicht das Dienstprogramm DB2UDDL zum Ändern der eindeutigen Indizes verwenden.) Dieses Dienstprogramm generiert Anweisungen CREATE UNIQUE INDEX für eindeutige Indizes auf Benutzertabellen und schreibt diese in eine Datei. Das Ausführen dieser Datei als DB2-CLP-Befehlsdatei bewirkt, daß der eindeutige Index in die Semantik von Version 7 umgewandelt wird. Detaillierte Informationen zur Verwendung dieses Dienstprogramms finden Sie in einem der Handbücher *Einstieg*.
7. Führen Sie wahlfrei den Befehl RUNSTATS für Tabellen aus, die besonders entscheidend für die Leistung von SQL-Abfragen sind. Die alten Statistikdaten werden in der migrierten Datenbank beibehalten und erst aktualisiert, wenn Sie den Befehl RUNSTATS ausführen.
8. Verwenden Sie wahlfrei das Dienstprogramm DB2RBIND, um alle Pakete wieder gültig zu machen, oder lassen Sie die Gültigkeit von Paketen implizit bei der ersten Verwendung eines Pakets wiederherstellen.
9. Migrieren Sie wahlfrei EXPLAIN-Tabellen, wenn Sie beabsichtigen, sie in Version 7 zu verwenden. Weitere Informationen finden Sie in "SQL-EXPLAIN-Einrichtung" im Handbuch *Systemverwaltung: Optimierung*.
10. Optimieren Sie die Konfigurationsparameter Ihrer Datenbank und Ihres Datenbankmanagers, um die Erweiterungen von Version 7 zu nutzen.

Anhang D. Inkompatibilitäten zwischen Releases

Dieser Abschnitt beschreibt Inkompatibilitäten, die zwischen DB2 Universal Database und früheren Releases von DB2 bestehen.

Eine *Inkompatibilität* ist ein Bestandteil von DB2 Universal Database, dessen Funktionsweise im Vergleich zu einem früheren Release von DB2 geändert wurde. Bei Verwendung in einer vorhandenen Anwendung führt sie zu einem unerwarteten Ergebnis, macht eine Änderung in der Anwendung erforderlich oder beeinträchtigt die Leistung. In diesem Kontext bezieht sich die Bezeichnung *Anwendung* auf folgendes:

- Anwendungsprogrammcode
- Dienstprogramme anderer Hersteller
- Interaktive SQL-Abfragen
- Aufrufe von Befehlen oder APIs

Es werden Inkompatibilitäten beschrieben, die mit DB2 Universal Database Version 6 und Version 7 eingeführt wurden. Sie werden nach folgenden Kategorien untergliedert:

- Systemkatalogsichten
- Anwendungsprogrammierung
- SQL
- Datenbanksicherheit und Optimierung
- Dienstprogramme und Tools
- Konnektivität und Koexistenz
- Konfigurationsparameter

Jeder Inkompatibilitätsabschnitt enthält eine Beschreibung der Inkompatibilität, das Symptom oder den Effekt der Inkompatibilität und mögliche Lösungsmaßnahmen. Zu Anfang jeder Inkompatibilitätsbeschreibung gibt ein Indikator an, auf welches Betriebssystem sich die Inkompatibilität bezieht:

WIN Von DB2 unterstützte Microsoft Windows-Plattformen

UNIX Von DB2 unterstützte UNIX-basierte Plattformen

OS/2 OS/2

Anmerkung: Mit dem Stand von DB2 Universal Database Version 6 werden Clients der Version 1.x und Version 2.x, einschließlich der Cli-

ents, die mit DB2 Parallel Edition Version 1.2-Servern geliefert werden, nicht länger unterstützt.

Geplante Inkompatibilitäten von DB2 Universal Database

Dieser Abschnitt beschreibt zukünftige Inkompatibilitäten, die Benutzer von DB2 Universal Database bei der Codierung neuer Anwendungen bzw. der Änderung vorhandener Anwendungen berücksichtigen sollten. Dadurch wird die Migration auf künftige Versionen von DB2 UDB vereinfacht.

Schreibgeschützte Sichten in einer künftigen Version von DB2 Universal Database

WIN	UNIX	OS/2
-----	------	------

Änderung

Die Systemkatalogsichten werden schreibgeschützte Sichten. Die SYSSTAT-Sichten bleiben weiterhin aktualisierbar.

Symptom

UPDATE-Anweisungen, die bisher für Spalten in den SYSCAT-Sichten funktionierten, schlagen nun fehl.

Erläuterung

Tools oder Anwendungen sind so codiert, daß sie Werte im Katalog durch Aktualisieren der Spalte gemäß der Definition in der SYSCAT-Sicht ändern.

Maßnahme

Ändern Sie das Tool oder die Anwendung so, daß der Katalog durch Aktualisieren der Spalte gemäß der Definition in der SYSSTAT-Sicht geändert wird.

PK_COLNAMES und FK_COLNAMES in einer künftigen Version von DB2 Universal Database

WIN	UNIX	OS/2
-----	------	------

Änderung

Die SYSCAT.REFERENCES-Spalten PK_COLNAMES und FK_COLNAMES sind nicht mehr verfügbar.

Symptom

Die Spalte ist nicht mehr vorhanden, und es wird ein Fehler zurückgegeben.

Erläuterung

Tools oder Anwendungen sind so codiert, daß sie die veralteten Spalten PK_COLNAMES und FK_COLNAMES verwenden.

Maßnahme

Ändern Sie das Tool oder die Anwendung so, daß statt dessen die Sicht SYSCAT.KEYCOLUSE verwendet wird.

COLNAMES in einer künftigen Version von DB2 Universal Database nicht mehr verfügbar

WIN	UNIX	OS/2
-----	------	------

Änderung

Die SYSCAT.INDEXES-Spalte COLNAMES ist nicht mehr verfügbar.

Symptom

Die Spalte ist nicht mehr vorhanden, und es wird ein Fehler zurückgegeben.

Erläuterung

Tools oder Anwendungen sind so codiert, daß sie die veraltete Spalte COLNAMES verwenden.

Maßnahme

Ändern Sie das Tool oder die Anwendung so, daß statt dessen die Sicht SYSCAT.INDEXCOLUSE verwendet wird.

Inkompatibilitäten in DB2 Universal Database Version 7

Dieser Abschnitt beschreibt die Inkompatibilitäten, die mit DB2 Universal Database Version 7 eingeführt werden.

Anwendungsprogrammierung

Query Patroller Universal Client

WIN	UNIX	OS/2
-----	------	------

Änderung: Diese neue Version des Client Application Enabler (CAE) funktioniert nur mit Query Patroller Server Version 7, weil neue gespeicherte Prozeduren vorhanden sind. CAE ist die Anwendungsschnittstelle für DB2, die letzten Endes alle Anwendungen passieren müssen, um auf die Datenbank zuzugreifen.

Symptom: Wenn diese CAE-Version mit einem Server einer früheren Version ausgeführt wird, wird SQL29001 zurückgegeben.

Objektumsetzungsfunktionen und strukturierte Typen

WIN	UNIX	OS/2
-----	------	------

Änderung: Es gibt eine kleinere und entfernt mögliche Inkompatibilität zwischen einem Client einer Version vor Version 7 und einem Server der Version 7, die im Zusammenhang mit Änderungen am SQL-Deskriptorbereich (SQLDA) stehen. Wie im Handbuch *Application Development Guide* erläutert, kann Byte 8 des zweiten SQLVAR-Feldes nun den Wert X'12' (neben den Werten X'00' und X'01') annehmen. Anwendungen, die den neuen Wert nicht abfangen, können von dieser Erweiterung beeinträchtigt werden.

Maßnahme: Da es in zukünftigen Releases noch weitere Erweiterungen an diesem Feld geben kann, sind Entwickler gut beraten, dieses Feld nur auf explizit definierte Werte zu testen.

Von JVM verwendete Versionen von Class- und Jar-Dateien

WIN	UNIX	OS/2
-----	------	------

Änderung: Wenn früher eine gespeicherte Java-Prozedur oder eine benutzerdefinierte Funktion (UDF) gestartet wurde, sperrte Java Virtual Machine (JVM) alle Dateien, die in CLASSPATH angegeben waren (einschließlich der Dateien in sqllib/function). JVM verwendete diese Dateien, bis der Datenbankmanager gestoppt wurde. Abhängig von der Umgebung, in der Sie eine gespeicherte Prozedur oder UDF ausführen (d. h. abhängig vom Wert des Konfigurationsparameters *keepdari* des Datenbankmanagers und davon, ob die gespeicherte Prozedur abgeschirmt (fenced) wird), ermöglicht Ihnen das Aktualisieren (Refresh) von Klassen ein Ersetzen von CLASS- und JAR-Dateien, ohne den Datenbankmanager zu stoppen. Dies unterscheidet sich von der früheren Funktionsweise.

Geänderte Funktionalität der Befehle zum Installieren, Ersetzen und Entfernen eines JAR

WIN	UNIX	OS/2
-----	------	------

Änderung: In der Vergangenheit bewirkte die Installation eines JAR ein Abbrechen aller DARI-Prozesse (Database Application Remote Interface). Auf diese Weise wurde sichergestellt, daß eine neue gespeicherte Prozedurklasse garantiert mit dem nächsten Aufruf ausgewählt wurde. Zur Zeit brechen keine JAR-Befehle DARI-Prozesse ab. Um sicherzustellen, daß Klassen aus neu installierten oder ersetzten JARs ausgewählt werden, müssen Sie den Befehl `SQLJ.REFRESH_CLASSES` explizit absetzen.

Eine weitere Inkompatibilität, die durch das Nichtabbrechen von DARI-Prozessen eingeführt wird, ist die Tatsache, daß für abgeschirmte (fenced) gespeicherte Prozeduren, wenn der Wert des Konfigurationsparameters *kee-*

pdari des Datenbankmanagers auf "YES" gesetzt ist, Clients verschiedene Versionen der JAR-Dateien erhalten können. Betrachten Sie folgendes Szenario:

1. Benutzer A ersetzt ein JAR und aktualisiert (Refresh) die Klassen nicht.
2. Benutzer A ruft anschließend eine gespeicherte Prozedur aus dem JAR auf. Unter der Annahme, daß dieser Aufruf denselben DARI-Prozeß verwendet, erhält Benutzer A eine alte Version der JAR-Datei.
3. Benutzer B ruft dieselbe gespeicherte Prozedur auf. Dieser Aufruf arbeitet mit einer neuen DARI, was bedeutet, daß das neu erstellte Klassenladeprogramm (Class loader) die neue Version der JAR-Datei auswählt.

Mit anderen Worten, wenn Klassen nach JAR-Operationen nicht aktualisiert (Refresh) werden, kann eine gespeicherte Prozedur aus verschiedenen Versionen von JARs aufgerufen werden, je nachdem, welche DARI-Prozesse verwendet werden. Dies unterscheidet sich von der früheren Funktionsweise, bei der (durch Abbrechen (Flush) der DARI-Prozesse) sichergestellt war, daß neue Klassen immer verwendet wurden.

Inkompatibilität bei 32-Bit-Anwendungen

	UNIX	
--	------	--

Änderung: Ausführbare 32-Bit-Codedateien (DB2-Anwendungen) funktionieren mit der neuen 64-Bit-Datenbanksteuerkomponente nicht.

Symptom: Die Anwendung kann nicht verbunden ("gelinkt") werden. Wenn Sie versuchen, 32-Bit-Objekte mit der 64-Bit-Anwendungsbibliothek von DB2 zu verbinden, wird eine Betriebssystemfehlernachricht des Verbindungseditors (Linker) angezeigt.

Maßnahme: Die Anwendung muß als ausführbare 64-Bit-Codedatei neu kompiliert und wieder mit den neuen 64-Bit-Bibliotheken von DB2 verbunden (gelinkt) werden.

Ändern des Längenfelds des Arbeitspuffers

WIN	UNIX	OS/2
-----	------	------

Änderung: Jede benutzerdefinierte Funktion (UDF), die das Längenfeld des Arbeitspuffers ändert, der an die UDF übergeben wurde, empfängt nun einen SQLCODE-Wert -450.

Symptom: Eine UDF, die das Längenfeld des Arbeitspuffers ändert, schlägt fehl. Die aufrufende Anweisung empfängt den SQLCODE-Wert -450 mit Angabe des Schemas und des bestimmten Namens der Funktion.

Maßnahme: Schreiben Sie den Hauptteil der UDF um, so daß das Längfeld des Arbeitspuffers nicht geändert wird.

SQL

Anwendungen, die mit dem Schema SESSION qualifizierte reguläre Tabellen verwenden

WIN	UNIX	OS/2
-----	------	------

Änderung: Das Schema SESSION ist das einzig zulässige Schema für temporäre Tabellen und wird nun von DB2 verwendet, um anzuzeigen, daß ein mit SESSION qualifizierter Tabellename auf eine temporäre Tabelle verweisen kann. Jedoch ist SESSION kein reserviertes Schlüsselwort für temporäre Tabellen und kann als Schema für reguläre Basistabellen verwendet werden. Aus diesem Grund kann eine Anwendung vielleicht sowohl eine richtige Tabelle SESSION.T1 als auch eine deklarierte temporäre Tabelle SESSION.T1 gleichzeitig vorfinden. Wenn beim Binden eines Pakets eine statische Anweisung angebroffen wird, die einen (explizit oder implizit) durch SESSION qualifizierten Tabellenverweis enthält, wird weder ein Abschnitt (Section) noch eine Abhängigkeit für diese Anweisung in den Katalogen gespeichert. Statt dessen muß dieser Abschnitt zur Laufzeit inkrementell gebunden werden. Dadurch wird eine Kopie des Abschnitts in den Cache für dynamisches SQL gestellt, wo die zwischengespeicherte Kopie nur für das eindeutige Exemplar der Anwendung privat ist. Wenn zur Laufzeit eine deklarierte temporäre Tabelle mit übereinstimmendem Tabellennamen vorhanden ist, wird die deklarierte temporäre Tabelle verwendet, selbst wenn eine permanente Basistabelle desselben Namens existiert.

Symptom: In Version 6 (und früheren) verweisen alle Paketen mit statischen Anweisungen, die Tabellen mit durch SESSION qualifizierten Namen betreffen, immer auf eine permanente Basistabelle. Beim Binden des Pakets werden ein Abschnitt (Section) sowie relevante Abhängigkeitseinträge für diese Anweisung in den Katalogen gespeichert. In Version 7 werden diese Anweisungen nicht zur Bindezeit gebunden und können zur Laufzeit möglicherweise in eine deklarierte temporäre Tabelle des gleichen Namens aufgelöst werden. Daraus können folgende Situationen entstehen:

- Bei Migration von Version 5. Wenn ein solches Paket in Version 5 vorhanden war, wird es in Version 6 erneut gebunden, und die statischen Anweisungen werden nun inkrementell gebunden. Dies kann sich auf die Leistung auswirken, da diese inkrementell gebundenen Abschnitte (Sections) sich wie im Cache zwischengespeichertes dynamisches SQL verhalten, mit der Ausnahme, daß der im Cache zwischengespeicherte Abschnitt nicht mit anderen Anwendungen gemeinsam benutzt werden kann (nicht einmal von verschiedenen Exemplaren derselben ausführbaren Anwendungsdatei).

- Bei Migration von Version 6 zu Version 7. Wenn ein solches Paket in Version 6 vorhanden war, wird es in Version 7 nicht unbedingt erneut gebunden. Statt dessen werden die Anweisungen weiterhin als reguläres statisches SQL unter Verwendung des Abschnitts (Section) ausgeführt, der im Katalog zu ursprünglichen Bindezeit gespeichert wurde. Wenn dieses Paket jedoch erneut gebunden wird (implizit oder explizit), werden die Anweisungen im Paket mit durch SESSION qualifizierten Tabellenverweisen nicht mehr gespeichert und erfordern inkrementelles Binden. Dadurch kann die Leistung beeinträchtigt werden.

Zusammengefaßt läßt sich sagen, daß jedes Paket, daß in Version 7 mit statischen Anweisungen gebunden wird, die auf durch SESSION qualifizierte Tabellen verweisen, sich nicht länger wie statisches SQL verhalten, weil sie ein inkrementelles Binden erfordern. Wenn der Anwendungsprozeß eine Anweisung DECLARE GLOBAL TEMPORARY TABLE für eine Tabelle absetzt, die den gleichen Namen wie eine vorhandene, mit SESSION qualifizierte Tabelle, Sicht bzw. ein solcher Aliasname besitzt, werden Verweise auf diese Objekte immer auf die deklarierte temporäre Tabelle bezogen.

Maßnahme: Ändern Sie nach Möglichkeit die Schemennamen permanenter Tabellen, so daß nicht SESSION verwendet wird. Ansonsten gibt es keine Abhilfe, außer sich die Auswirkungen auf die Leistung und den möglichen Konflikt mit deklarierten temporären Tabellen bewußt zu machen.

Die folgende Abfrage kann zur Erkennung von Tabellen, Sichten und Aliasnamen verwendet werden, die betroffen sein könnten, wenn eine Anwendung mit temporären Tabellen arbeitet:

```
select tabschema, tablename from SYSCAT.TABLES where tabschema = 'SESSION'
```

Die folgende Abfrage kann zur Ermittlung von gebundenen Paketen der Version 7 dienen, für die statische Abschnitte (Sections) in den Katalogen gespeichert sind und deren Funktionsweise sich ändern könnte, wenn das Paket erneut gebunden wird (nur bei Migration von Version 6 zu Version 7 relevant):

```
select pkgschema, pkgname, bschema, bname from syscat.packagedep
where bschema = 'SESSION' and btype in ('T', 'V', 'I')
```

Dienstprogramme und Tools

Data Links File Manager und File System Filter unter Solaris

	UNIX	
--	------	--

Änderung: Data Links File Manager und File System Filter werden unter Solaris OS 2.5.1 nicht unterstützt.

db2set unter AIX und Solaris

	UNIX	
--	------	--

Änderung: Der Befehl "db2set -ul (user level)" und die zugehörigen Funktionen wurden nicht auf AIX oder Solaris portiert.

Konnektivität und Koexistenz

32-Bit-Client-Inkompatibilität

WIN	UNIX	OS/2
-----	------	------

Änderung: 32-Bit-Clients können keine Verbindung zu Exemplaren (Attach) oder Datenbanken (Connect) auf 64-Bit-Servern herstellen.

Symptom: Wenn sowohl der Client als auch der Server mit Code der Version 7 arbeiten, wird SQL1434N zurückgegeben. Ansonsten schlägt der Verbindungsversuch (Attach oder Connect) mit SQLCODE-Wert -30081 fehl.

Maßnahme: Verwenden Sie 64-Bit-Clients.

Inkompatibilitäten in DB2 Universal Database Version 6

Dieser Abschnitt beschreibt die Inkompatibilitäten, die mit DB2 Universal Database Version 6 eingeführt werden.

Systemkatalogsichten

Systemkatalogsichten in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: In den Systemkatalogsichten wurden neue Codes eingeführt: „U“ für typisierte Tabellen und „W“ für typisierte Sichten.

Symptom: Abfragen, die in den Systemkatalogen mit dem Typencode „T“ nach Tabellen bzw. „V“ nach Sichten suchen, finden keine typisierten Tabellen und Sichten mehr.

Erläuterung: Verschiedene Systemkataloge, einschließlich der Systemkatalogsichten TABLES, PACKAGEDEP, TRIGDEP und VIEWDEP, haben eine Spalte namens TYPE oder BTYPE, die einen aus einem Buchstaben bestehenden Typencode enthält. In Version 5.2 wurde der Typencode „T“ für alle Tabellen und der Typencode „V“ für alle Sichten verwendet. In Version 6 haben nicht typisierte Tabellen weiterhin den Typencode „T“. Typisierte Tabelle haben

dagegen den neuen Typencode „U“. Analog dazu haben nicht typisierte Sichten weiterhin den Typencode „V“. Typisierte Sichten haben dagegen den neuen Typencode „W“. Darüber hinaus wird eine neue Art von Tabelle, eine sogenannte Hierarchietabelle, die nicht direkt von Benutzern erstellt, sondern vom System zur Implementierung von Tabellenhierarchien verwendet wird, in den Systemkatalogtabellen mit dem Typencode „H“ angezeigt.

Maßnahme: Ändern Sie das Tool oder die Anwendung so, daß die Codes für typisierte Tabellen und Sichten erkannt werden. Wenn das Tool oder die Anwendung eine logische Sicht von Tabellen benötigt, sollten die Typencodes „T“, „U“, „V“ und „W“ verwendet werden. Benötigt das Tool oder die Anwendung eine physische Sicht von Tabellen, einschließlich Hierarchietabellen, sollten die Typencodes „T“ und „H“ verwendet werden.

Primär- und Fremdschlüsselspaltennamen in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: In zwei SYSCAT.REFERENCES-Spalten, PK_COLNAMES und FK_COLNAMES, wird der Datentyp von VARCHAR(320) in VARCHAR(640) geändert.

Symptom: Primärschlüssel- oder Fremdschlüsselspaltennamen werden abgeschnitten, sind nicht korrekt oder fehlen.

Erläuterung: Wenn in einem Primärschlüssel oder Fremdschlüssel Spaltennamen verwendet werden, die länger als 18 Byte sind, kann das Format, mit dem die Liste von Spaltennamen in diesen beiden Spalten gespeichert ist, nicht gleich bleiben. Die mit Leerzeichen begrenzten 20 Byte langen Spaltennamen, die auf die Spalte folgen, deren Länge (*n*) größer als 18 Byte ist, werden um *n*-18 Byte nach rechts verschoben. Wenn die Liste von Spaltennamen 640 Byte übersteigt, enthält die Spalte auch die leere Zeichenfolge.

Maßnahme: Die Sicht SYSCAT.KEYCOLUSE enthält die Liste der Spalten, aus denen ein Primär-, Fremd- oder eindeutiger Schlüssel besteht, und sollte anstelle der Spalten in SYSCAT.REFERENCES verwendet werden. Alternativ dazu können Benutzer die Länge der Spaltennamen auf 18 Byte oder die Gesamtlänge der Spaltenliste auf 640 Byte begrenzen.

SYSCAT.VIEWS-Spalte TEXT in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Sichttext in der SYSCAT.VIEWS-Spalte TEXT wird nicht mehr auf mehrere Zeilen verteilt. Der Datentyp wird von VARCHAR(3600) in CLOB(64K) geändert.

Symptom: Das Tool oder die Anwendung zeigt nicht den vollständigen Sichttext an.

Erläuterung: Tools oder Anwendungen, die so codiert sind, daß sie erwarten, daß höchstens 3600 (oder möglicherweise 3900) Byte auf einmal von der Spalte TEXT zurückgegeben werden, können die erhöhte Größe dieses Felds nicht handhaben. Der Mechanismus zum Abrufen mehrerer Zeilen und zum Wiederherstellen des Sichttexts über das Feld SEQNO ist nicht mehr notwendig. Der Wert für SEQNO ist immer 1.

Maßnahme: Ändern Sie das Tool oder die Anwendung so, daß Werte aus der Spalte TEXT gehandhabt werden können, die größer als 3600 Byte sind. Alternativ dazu könnte der Sichttext so umgeschrieben werden, daß er 3600 Byte nicht übersteigt.

SYSCAT.STATEMENTS-Spalte TEXT in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Anweisungstext in der SYSCAT.STATEMENTS-Spalte TEXT wird nicht mehr auf mehrere Zeilen verteilt. Der Datentyp wird von VARCHAR(3600) in CLOB(64K) geändert.

Symptom: Das Tool oder die Anwendung zeigt nicht den vollständigen Anweisungstext an.

Erläuterung: Tools oder Anwendungen, die so codiert sind, daß sie erwarten, daß höchstens 3600 (oder möglicherweise 3900) Byte auf einmal von der Spalte TEXT zurückgegeben werden, können die erhöhte Größe dieses Felds nicht handhaben. Der Mechanismus zum Abrufen mehrerer Zeilen und zum Wiederherstellen des Anweisungstexts über das Feld SEQNO ist nicht mehr notwendig. Der Wert für SEQNO ist immer 1.

Maßnahme: Ändern Sie das Tool oder die Anwendung so, daß Werte aus der Spalte TEXT gehandhabt werden können, die größer als 3600 Byte sind. Alternativ dazu könnte der Anweisungstext so umgeschrieben werden, daß er 3600 Byte nicht übersteigt.

SYSCAT.INDEXES-Spalte COLNAMES in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Der Datentyp der SYSCAT.INDEXES-Spalte COLNAMES wird von VARCHAR(320) in VARCHAR(640) geändert.

Symptom: Es fehlen Spaltennamen aus einem Index.

Erläuterung: Tools oder Anwendungen, die so codiert sind, daß sie Daten aus einer Spalte mit dem Datentyp VARCHAR(320) abrufen, können die erhöhte Größe dieses Felds nicht handhaben.

Maßnahme: Die Sicht SYSCAT.INDEXCOLUSE enthält die Liste von Spalten, aus denen ein Index besteht. Sie sollte anstelle der Spalte COLNAMES verwendet werden. Alternativ dazu können Sie eine Spalte aus dem Index entfernen oder die Größe des Spaltennamens verringern, so daß die Liste von Spaltennamen (mit führendem + oder -) 320 Byte nicht übersteigt.

SYSCAT.CHECKS-Spalte TEXT in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Der Datentyp der SYSCAT.CHECKS-Spalte TEXT wird von CLOB(32K) in CLOB(64K) geändert.

Symptom: Die Klausel für die Prüfung auf Integritätsbedingung ist unvollständig.

Erläuterung: Tools oder Anwendungen, die so codiert sind, daß sie Daten aus einer Spalte mit dem Datentyp CLOB(32K) abrufen, können die erhöhte Größe dieses Felds nicht handhaben.

Maßnahme: Ändern Sie das Tool oder die Anwendung so, daß Werte aus der Spalte TEXT gehandhabt werden können, die länger als 32 KB sind. Alternativ dazu können Sie die Klausel für die Prüfung auf Integritätsbedingung so umschreiben, daß sie weniger Zeichen verwendet und so 32 KB nicht übersteigt.

Spaltendatentyp in DB2 Universal Database Version 6 in BIGINT geändert

WIN	UNIX	OS/2
-----	------	------

Änderung: Bei mehreren Spalten von Systemkatalogsichten wurde der Datentyp von INTEGER in BIGINT geändert.

Symptom: Werte sind viel kleiner (oder größer) als erwartet, insbesondere bei statistischen Daten.

Erläuterung: Tools oder Anwendungen, die so codiert sind, daß sie Daten aus einer Spalte mit dem Datentyp INTEGER abrufen, können die erhöhte Größe dieses Felds nicht handhaben.

Maßnahme: Ändern Sie das Tool oder die Anwendung so, daß Werte gehandhabt werden können, die größer als der Höchstwert oder kleiner als Mindestwert sind, der in einem INTEGER-Feld gespeichert werden kann. Alternativ dazu können Sie die zugrundeliegende Struktur oder den zugrundeliegenden SQL-Code ändern, durch die/den der Wert aus dem Bereich herausfällt, der durch ein INTEGER-Feld dargestellt werden kann.

Spaltenabweichung in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Neue Spalten werden in der SYSCAT-Sichtdefinition nicht am Ende von Sichten eingefügt.

Symptom: Die erneute Vorverarbeitung schlägt mit mehreren Abweichungen der Spalte oder des Spaltendatentyps fehl.

Erläuterung: Neue Spalten werden in die Systemkatalogsichten eingeführt und für eine Sofortabfrageumgebung sinnvoll angeordnet, d. h. kürzere Spalten werden vor sehr langen Spalten angeordnet, und die Spalte REMARKS ist immer die letzte Spalte.

Maßnahme: Nennen Sie die Spalten in der SELECT-Liste explizit, statt „SELECT *” zu verwenden.

SYSCAT.COLUMNS und SYSCAT.ATTRIBUTES in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: SYSCAT.COLUMNS und SYSCAT.ATTRIBUTES enthalten jetzt Einträge für übernommene Spalten und Attribute.

Symptom: Abfragen auf SYSCAT.COLUMNS zum Abrufen der Spalten einer typisierten Tabelle oder Sicht sowie Abfragen auf SYSCAT.ATTRIBUTES zum Abrufen der Attribute eines strukturierten Typs können in Version 6 mehr Zeilen zurückgeben als in Version 5.2, wenn das Ziel der Abfrage eine untergeordnete Tabelle, eine untergeordnete Sicht oder ein Subtyp ist.

Erläuterung: In Version 5.2 enthielten die Kataloge COLUMNS und ATTRIBUTES für eine bestimmte Tabelle, eine bestimmte Sicht oder einen bestimmten strukturierten Typ nur Einträge für Spalten und Attribute, die von dieser

Tabelle, dieser Sicht oder diesem Typ eingeführt wurden. Spalten und Attribute, die von übergeordneten Tabellen oder Typen übernommen wurden, wurden nicht in den Katalogen dargestellt. In Version 6 enthalten jetzt die Kataloge COLUMNS und ATTRIBUTES jedoch Einträge für übernommene Spalten und Attribute.

Maßnahme: Ändern Sie das Tool oder die Anwendung so, daß die neuen Einträge in den Katalogen COLUMNS und ATTRIBUTES erkannt werden.

OBJCAT-Sichten in DB2 Universal Database Version 6 nicht mehr unterstützt

WIN	UNIX	OS/2
-----	------	------

Änderung: Die rekursiven Katalogsichten im Schema OBJCAT von Version 5.2 werden nicht mehr mit DB2 Universal Database mitgeliefert.

Symptom: Abfragen, die für die OBJCAT-Katalogsichten geschrieben sind, werden nicht mehr erfolgreich ausgeführt.

Maßnahme: Die meisten der Informationen, die bisher in den OBJCAT-Sichten enthalten waren, sind jetzt in normalen SYSCAT-Katalogsichten enthalten. In den meisten Fällen können Sie die Informationen aus den Systemkatalogsichten abrufen. Wenn Sie von Version 5.2 migrieren und die OBJCAT-Katalogsichten vorhanden sind, sollten diese gelöscht werden. Dies kann mit der Befehlszeilenprozedur `objcatdp.db2` im Unterverzeichnis `misc` des Verzeichnisses `sqllib` erfolgen.

Sie können auch Ihre eigenen OBJCAT-Sichten erstellen, die zu den in Version 5.2 unterstützten Katalogsichten äquivalent sind.

In Version 5.2 wurden die Benutzer in Anhang E des Handbuchs *SQL Reference* darauf hingewiesen, daß die OBJCAT-Katalogsichten temporär sind und in zukünftigen Versionen nicht mehr unterstützt werden.

Abhängigkeitscodes in DB2 Universal Database Version 6 geändert

WIN	UNIX	OS/2
-----	------	------

Änderung: In den Systemkatalogsichten werden die hierarchischen Abhängigkeiten, die bisher mit dem Code „H“ bezeichnet wurden, jetzt mit dem Code „O“ bezeichnet.

Symptom: Abfragen, die in den Katalogsichten mit dem Code „H“ nach hierarchischen Abhängigkeiten suchen, funktionieren nicht mehr korrekt.

Erläuterung: Verschiedene Systemkataloge, einschließlich der Systemkatalogsichten PACKAGEDEP, TRIGDEP und VIEWDEP, haben eine Spalte namens BTYPE. In Version 5.2 bezeichnen die OBJCAT-Sichten hierarchische Abhängigkeiten mit dem Code „H“. In Version 6 werden diese Abhängigkeiten mit dem Code „O“ bezeichnet.

Maßnahme: Überarbeiten Sie diese Abfragen, so daß sie nach Code „O“ suchen.

SYSIBM-Basiskatalogtabellen in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Die folgenden Änderungen wurden an den SYSIBM-Basiskatalogtabellen vorgenommen, die Sie vielleicht immer noch statt der SYSCAT-Sichten verwenden:

- Gelöschte Felder (jedoch in den SYSCAT-Sichten weiter vorhanden):
 - SYSSTMT.SEQNO
 - SYSVIEWS.SEQNO
- Umbenannte Katalogtabelle: SYSTRIGDEP wurde in SYSDEPENDENCIES geändert. Die Spalten BCREATOR und DCREATOR wurden in BSCHEMA bzw. DSCHEMA umbenannt. Die Sicht SYSCAT.TRIGDEP wurde nicht geändert.
- Gelöschte Felder (waren nie in den SYSCAT-Sichten enthalten):
 - SYSATTRIBUTES.DEFAULT_VALUE
 - SYSATTRIBUTES.NULLS
 - SYSCOLUMNS.SERVERTYPE
 - SYSDATATYPES.REFREP_TYPENAME
 - SYSDATATYPES.REFREP_TYPESHEMA
 - SYSDATATYPES.REFREP_LENGTH
 - SYSDATATYPES.REFREP_SCALE
 - SYSDATATYPES.REFREP_CODEPAGE
 - SYSINDEXES.TEXT
(War in der Sicht enthalten, aber nur zur künftigen Nutzung reserviert.)
 - SYSPLANDEP.PUBLICPRIV
 - SYSSECTION.SEQNO
 - SYSTABAUTH.UPDATE_BY_COLS
 - SYSTABAUTH.REF_BY_COLS
 - SYSTABLES.MINPDLENGTH
 - SYSTABLESPACES.READONLY

- SYSTABLESPACES.REMOVABLEMEDIA
- Datentypänderungen:
 - SYSSECTION.SECTION: von VARCHAR(3600) in CLOB(10M)
 - SYSPLANDEP.COLUSAGE: von VARCHAR(3000) FOR BIT DATA in BLOB(5K)

Anwendungsprogrammierung

Datentyp VARCHAR in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Die mögliche Maximalgröße des Datentyps VARCHAR (VARGRAPHIC) wurde in Version 6 von 4000 Zeichen (2000 Doppelbytezeichen) auf 32672 Zeichen (16336 Doppelbytezeichen) erhöht.

Symptom: Eine Anwendung, die Puffer mit einer festen Länge von 4000 Byte für den Datentyp VARCHAR (VARGRAPHIC) verwendet, überschreibt möglicherweise Puffer oder schneidet Daten ab, wenn sie ein VARCHAR-Feld mit mehr als 4000 Byte in einen zu kleinen Puffer abruft. Die CLI-Funktion SQLGetTypeInfo() gibt jetzt die Größe von VARCHAR als 32672 zurück. CLI-Anwendungen, die diesen Wert in Tabellen-DDLs verwenden, erhalten möglicherweise Fehler, weil keine Tabellenbereiche mit ausreichender Seitengröße verfügbar sind. Weitere Informationen zur Größe von Tabellenbereichen finden Sie in „Benutzertabellendaten“ auf Seite 135.

Maßnahme: Bei der Codierung der Anwendung ist zu empfehlen, zuerst die Spalten der Ergebnismenge (mit der Anweisung DESCRIBE) zu beschreiben und dann Puffer zu verwenden, deren Größe sich nach der von der Anweisung DESCRIBE zurückgelieferten Länge richtet.

Positionierte UPDATE- und DELETE-Anweisungen bei Java-Programmierung in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Beim Programmieren von Java in Version 6 verwenden positionierte UPDATE- und DELETE-Anweisungen die Standardberechtigungskennung der Person, die das Cursorpaket gebunden hat. Dies unterscheidet sich von Version 5.2, bei der die Berechtigungskennung der Person verwendet wird, die das Paket ausführt.

Symptom: Das Paket mit den positionierten UPDATE- und DELETE-Anweisungen wird möglicherweise nicht ausgeführt, da die Berechtigungskennung der Person, die das Paket gebunden hat, keine ausreichende Berechtigung hat.

Maßnahme: Der Berechtigungskennung der Person, die das Paket bindet, muß eine ausreichende Berechtigung zum Ausführen der positionierten UPDATE- und DELETE-Anweisungen im Paket erteilt werden. Erteilen Sie die korrekten Zugriffsrechte, und binden Sie das Paket dann erneut.

Syntaxänderung in der Klausel FOR UPDATE in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: In Version 5.2 kann in einem SQLJ-Programm die Klausel FOR UPDATE in einer SELECT-Anweisungen verwendet werden, um die Spalten anzugeben, die in nachfolgenden positionierten UPDATE-Anweisungen aktualisiert werden können. Die Syntax wurde für Version 6 geändert.

Symptom: Sie erhalten die Fehlermeldung SQJ0204E, wenn eine SELECT-Anweisung eine Klausel FOR UPDATE enthält.

Maßnahme: Entfernen Sie die Klausel FOR UPDATE aus der SELECT-Anweisung. Geben Sie einen aktualisierbaren Iterator über die Iteratordeklarationsklausel an. Beispiel:

```
#sql public iterator DelByName implements sqlj.runtime.ForUpdate(String EmpNo)
with updateColumns = (salary);
```

Wenn Sie explizit angeben wollen, welche Spalten aktualisierbar sind, geben Sie diese über das Schlüsselwort updateColumns in Verbindung mit der Klausel WITH an.

Weitere Informationen zur Deklaration positionierter Iteratoren finden Sie im Handbuch *Application Development Guide*.

Zeichennamenlängen in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: DB2 Universal Database Version 6 unterstützt 128 Byte lange Tabellen-, Sichten- und Aliasnamen sowie 30 Byte lange Spaltennamen. Bisher wurden 18 Byte lange Namen für alle diese Kategorien unterstützt.

Die Sonderregister USER und CURRENT SCHEMA wurden von CHAR(8) in VARCHAR(128) geändert. Das Sonderregister CURRENT EXPLAIN MODE wurde von CHAR(8) in VARCHAR(254) geändert. Die Ausgabe für die integrierten Funktionen TYPE_SCHEMA und TABLE_SCHEMA wurde von CHAR(8) in VARCHAR(128) geändert.

Symptom: Wenn Anwendungen, die vor Version 6 entwickelt wurden, für eine Datenbank der Version 6 ausgeführt werden, die die längeren Begrenzungen nicht verwendet, sollte sich das Anwendungsverhalten nicht ändern. Wenn Sie jedoch diese Anwendungen für eine Datenbank der Version 6 ausführen, die längere Namen verwendet, könnte es je nach Codierung dieser Anwendungen zu gewissen Nebeneffekten können.

Es folgen einige Beispiele:

- Eine vorhandene Anwendung ruft mit einer FETCH-Anweisung einen Tabellen- oder Spaltennamen (typischerweise aus einer Katalogsicht) in eine Host-Variable ab, die mit einer Länge von 18 Byte definiert wurde. Da 18 Byte bis Version 6 die maximale Größe für den Tabellen- oder Spaltennamen waren, prüft diese Anwendung möglicherweise nicht das `sqlwarn1`-Bit des SQL-Kommunikationsbereichs. Sie geht (fälschlicherweise) davon aus, daß ein Abschneiden der Daten niemals auftritt.
- Eine Anwendung ruft mit einer FETCH-Anweisung einen Tabellen- oder Spaltennamen (typischerweise aus einer Katalogsicht) in einen SQL-Deskriptorbereich (SQLDA) ab, bei dem die Größe des Felds `sqldata` auf der Basis des Feldes `sqllen` aus einer DESCRIBE-Operation der SELECT-Anweisung zugeordnet wurde. Dies führt dazu, daß das korrekte (nicht abgeschnittene) Ergebnis an die Anwendung zurückgegeben wird, obwohl die Größe der Tabellen- oder Spaltennamen sich möglicherweise erhöht hat. Wenn eine Anwendungslogik von der Annahme ausgeht, daß Spaltennamen auf 18 Byte begrenzt sind, können die zurückgegebenen längeren Namen auf unerwartete Weise behandelt werden. Zum Beispiel kann die Anzeige längerer Spaltennamen bei 18 Byte abgeschnitten werden.
- Da das Token-Feld des SQL-Kommunikationsbereichs (`sqlerrmc`) auf 70 Byte begrenzt ist, können vorhandene Anwendungen, die versuchen, eine Zeile in eine Tabelle einzufügen, betroffen sein. Als Reaktion auf die Fehlermeldung `SQL0204N` ermitteln solche Anwendungen den Namen der Tabelle aus dem Feld `sqlerrmc` des SQL-Kommunikationsbereichs und führen dann einige Operationen mit Hilfe dieses Objektnamens durch. Bei früheren Versionen von DB2 wurde durch die Begrenzung der Tabellen- oder Schemenkennungen sichergestellt, daß der gesamte Tabellenname in den SQL-Kommunikationsbereich aufgenommen wurde. Dies ist in Version 6 nicht der Fall.
- Eine Anwendung, die eine frühere Version einer API verwendet, erhält nur die ersten 18 Byte eines Tabellennamens.
- Auf vorhandene CLI- und ODBC-Anwendungen, die die Schemenfunktionen (wie `SQLTables()`, oder `SQLColumns()` u. a.) verwenden, ergibt sich eine Auswirkung, wenn eine Verbindung zu einem Server hergestellt wird, der längere Namen als 18 Byte unterstützt. Obwohl Warnungen ausgegeben werden, daß abgeschnitten wird, prüft die Anwendung möglicherweise diese Warnung nicht, und fährt mit einem abgeschnittenen Namen fort.

Maßnahme: Die beste Möglichkeit, Probleme dieser Art zu beheben, besteht darin, die Anwendung so neu zu codieren, daß sie längere Tabellen- und Spaltenname handhaben kann. Stellen Sie andernfalls sicher, daß diese Anwendungen nicht für Datenbanken der Version 6 ausgeführt werden, die Namen mit mehr als 18 Byte verwenden.

PC/IXF-Formatänderungen in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: DB2 Universal Database Version 6 unterstützt 128 Byte lange Tabellen-, Sichten- und Aliasnamen sowie 30 Byte lange Spaltennamen. Bisher wurden 18 Byte lange Namen für alle diese Kategorien unterstützt.

Symptom: Ein Client unter DB2 Universal Database Version 5 kann keine PC/IXF-Datei importieren, die von einem Client unter DB2 Universal Database Version 6 exportiert wurde (Fehlernachricht SQL3059N). Eine PC/IXF-Datei (exportiert aus einem Client unter DB2 Universal Database Version 6) kann nicht in eine Datenbank von DB2 Universal Database Version 5 geladen werden (Fehlernachricht SQL3059N).

Maßnahme: Verwenden Sie beim Importieren oder Laden von PC/IXF-Daten kompatible Versionen von DB2 Universal Database.

SQLNAME in nicht verdoppelten SQLVAR-Einträgen in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: DB2 Universal Database Version 6 unterstützt 30 Byte lange Spaltennamen. Bisher wurden 18 Byte lange Namen unterstützt. In Version 5 war die dokumentierte Funktionsweise, daß „0xFF“ in das 30. Byte eines Felds SQLNAME für nicht verdoppelte SQLVAR-Einträge plaziert wurde. Für systemgenerierte Namen und benutzerdefinierte Spaltennamen, die in einer Klausel AS angegeben wurden, wurde auch „0x00“ in das 30. Byte eingefügt.

In Version 6 wird „0xFF“ nur dann im 30. Byte zurückgegeben, wenn der Name systemgeneriert ist.

Symptom: Alle Anwendungen, die mit Hilfe des 30. Byte des Felds SQLNAME festzustellen, ob es sich um einen benutzerdefinierten Spaltennamen oder um einen vom System generierten Namen handelt, erhalten unerwartete Ergebnisse bei Logikprüfungen, wenn der benutzerdefinierte Spaltenname 30 Zeichen lang ist. Dies sollte selten vorkommen.

Maßnahme: Diese Anwendungen sollten so geändert werden, daß sie nur auf „0xFF“ im 30. Byte des Felds SQLNAME prüfen, wenn die Länge dieses Feldes kleiner als 30 ist. In diesem Fall ist der Name benutzerdefiniert.

Veraltete CLI-/ODBC-Konfigurationsschlüsselwörter in DB2 Universal Database Version 6

WIN		
-----	--	--

Änderung: Bei der Migration zu einer neuen Version von DB2 UDB können Sie die Funktionsweise des DB2-CLI-/ODBC-Treibers durch Angeben einer Gruppe wahlfreier Schlüsselwörter in der Datei `db2cli.ini` ändern.

In Version 6 sind die Schlüsselwörter TRANSLATEDLL und TRANSLATEOPTION veraltet.

Symptom: Diese Schlüsselwörter werden ignoriert, wenn sie noch angegeben sind. Das Wegfallen dieser Einstellungen kann zu erkennbaren Veränderungen in der Funktionsweise führen.

Maßnahme: Entscheiden Sie anhand der neuen Liste der gültigen Parameter, welche Schlüsselwörter und Einstellungen Ihrer Umgebung entsprechen. Informationen zu diesen Schlüsselwörtern finden Sie im Handbuch *CLI Guide and Reference*.

Ausgabedatenstromformat des Ereignismonitors in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Ausgabedatenströme des Ereignismonitors haben keine Versionssteuerung. Wenn daher Unterstützung für Tabellennamen hinzugefügt wird, die länger als 18-byte Byte sind, muß zu einem Ausgabedatenstromformat gewechselt werden.

Symptom: Anwendungen, die die Ausgabedatenströme des Ereignismonitors syntaktisch analysieren, funktionieren nicht mehr ordnungsgemäß.

Maßnahme: Es gibt zwei Möglichkeiten:

- Aktualisieren Sie die Anwendung, so daß sie den neuen Datenstrom verwendet.
- Setzen Sie die Registrierungsvariable

`DB2OLDEVMON=evmonname1,evmonname2,...`

Dabei ist *evmonname* der Name des Ereignismonitors (Event Monitor), der im alten Datenformat geschrieben werden soll. Beachten Sie, daß auf neue Felder im Ereignismonitor unter dem alten Datenformat nicht zugegriffen werden kann.

SQL

DATALINK-Spalten in DB2 Universal Database Version 6

	UNIX	
--	------	--

Änderung: Datenübertragungsverbindungswerte (DATALINK-Werte), die unter DB2 Universal Database Version 6 eingefügt werden, benötigen 4 Byte mehr Platz im Spaltenwertdeskriptor.

Symptom: Wenn in Version 5.2 erstellte DATALINK-Spalten aktualisiert werden, sind auf der Datenseite weitere 4 Byte erforderlich, um den neuen Spaltenwert zu speichern. Es ist daher möglicherweise nicht genügend Platz in der Datenseite vorhanden, um die Aktualisierung durchzuführen, und sie muß vielleicht auf eine neue Seite versetzt werden. Dadurch könnte der Aktualisierung der Platz ausgehen.

Maßnahme: Sie müssen auf Ihrem System mehr Platz für Aktualisierungen hinzufügen.

SYSFUN-Zeichenfolgefunktionskennungen in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Eine Reihe von Zeichenfolgefunktionen im Schema SYSFUN haben jetzt verbesserte Versionen, die im Schema SYSIBM definiert sind (integrierte Funktionen). Die Funktionsnamen sind LCASE, LTRIM, RTRIM und UCASE.

Symptom: Beim Vorbereiten von Anweisungen oder beim Erstellen von Sichten können sich die von diesen Funktionen zurückgegebenen Datentypen in Version 6 unterscheiden. Dies kommt daher, daß die integrierten Funktionen (unter dem Schema SYSIBM) normalerweise vor Funktionen im Schema SYSFUN aufgelöst werden.

Maßnahme: Es ist keine Maßnahme erforderlich. Normalerweise wird die integrierte Funktion vor der Funktion im Schema SYSFUN ausgewählt. Die frühere Funktionsweise kann durch Tauschen des SQL-Pfads (so daß SYSFUN vor SYSIBM steht) wiederhergestellt werden, jedoch wird die Leistung

dadurch beeinträchtigt. Die Funktion der früheren Version kann auch dadurch aufgerufen werden, daß der Funktionsname mit dem Schemenname SYSFUN qualifiziert wird.

Migrierte Pakete, Sichten, Übersichtstabellen, Auslöser und Integritätsbedingungen verwenden weiter die Version aus dem Schema SYSFUN, sofern keine explizite Aktion wie das Binden des Pakets oder das Neuerstellen der Sicht, der Übersichtstabelle, des Auslösers oder der Integritätsbedingung vorgenommen wird.

SYSTABLE-Spaltenänderung mit neuem Integritätsstatus in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Der Status „U“ in der Spalte CONST_CHECKED von SYSCAT.TABLES ändert sich anders, wenn eine Anweisung SET INTEGRITY ... OFF ausgeführt wird.

Symptom: Vor Version 6 änderte sich der Status „U“ in CONST_CHECKED in „N“, wenn eine Anweisung SET INTEGRITY ... OFF ausgeführt wurde. Jetzt ändert sich der Status „U“ in den Status „W“.

Maßnahme: Es ist keine Maßnahme erforderlich. Der neue Status „W“ in der Spalte CONST_CHECKED wird verwendet, um anzugeben, daß der Typ der Integritätsbedingung bisher durch den Benutzer geprüft wurde und daß einige Daten in der Tabelle eventuell auf Integrität geprüft werden müssen.

Der Status „N“ klärt nicht darüber auf, ob es alte Daten gibt, die noch nicht durch den Datenbankmanager überprüft wurden. In einer nachfolgenden Anweisung SET INTEGRITY ... IMMEDIATE CHECKED INCREMENTAL muß der Datenbankmanager einen Fehler zurückgeben, weil die Datenintegrität nicht gewährleistet werden kann, wenn nur neue Änderungen geprüft wurden. Andererseits kann der Status „W“ zurück in den Status „U“ geändert werden (wenn die Option INCREMENTAL angegeben ist), um anzugeben, daß der Benutzer weiterhin für die Integrität der Daten in der Tabelle verantwortlich ist. Wenn die Option INCREMENTAL nicht angegeben wird, wählt der Datenbankmanager die vollständige Verarbeitung, ändert den Status „W“ in den Status „Y“ und übernimmt die Verantwortung für die Aufrechterhaltung der Datenintegrität.

Datenbanksicherheit und Optimierung

Erstellen von Datenbanken mit Clients in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Die von den Clients verwendete Methode zum Erstellen einer Datenbank hat sich geändert.

Symptom: Die Verwendung eines Clients einer früheren Version zur Erstellung einer Datenbank führt zu Fehlern.

Maßnahme: Stellen Sie bei der Verwendung eines Clients zur Erstellung einer Datenbank sicher, daß der Client und der Server mit der gleichen Stufe des DB2-Codes arbeiten.

SELECT-Zugriffsrecht in DB2 Universal Database Version 6 in Hierarchie erforderlich

WIN 32-Bit	UNIX	OS/2
------------	------	------

Änderung: Die Angabe des Schlüsselworts ONLY (für eine Tabelle) setzt jetzt voraus, daß der Benutzer das SELECT-Zugriffsrecht für alle untergeordneten Tabellen der angegebenen typisierten Tabelle hat. Analog dazu setzt die Angabe des Schlüsselworts ONLY (für eine Sicht) jetzt voraus, daß der Benutzer das SELECT-Zugriffsrecht für alle untergeordneten Sichten der angegebenen typisierten Tabelle hat. In früheren Versionen von DB2 war das SELECT-Zugriffsrecht nur für die angegebene Tabelle oder Sicht erforderlich.

Symptom: Es gibt zwei mögliche Symptome:

- Ein Berechtigungsfehler (SQLCODE -551, SQLSTATE 42501) tritt beim erneuten Binden eines Pakets mit einer SQL-Anweisung auf, in der das Schlüsselwort ONLY in einer Klausel FROM angegeben ist, wenn die Berechtigungs-ID, unter der das Paket gebunden wurde, nicht das SELECT-Zugriffsrecht für die untergeordneten Tabellen der angegebenen typisierten Tabelle (oder Sicht) hat.
- Wenn die Definition einer Sicht oder eines Auslösers das Schlüsselwort ONLY in einer Klausel FROM enthält, arbeitet die Sicht oder der Auslöser weiter normal. Die Definition der Sicht oder des Auslösers kann jedoch nicht mehr verwendet werden, um eine neue Sicht oder einen neuen Auslöser zu erstellen, wenn der Ersteller nicht das SELECT-Zugriffsrecht für alle untergeordneten Tabellen der angegebenen Tabelle (oder Sicht) hat.

Maßnahme: Der Berechtigungs-ID, die ein Paket erneut binden oder eine neue Sicht oder einen neuen Auslöser erstellen soll, sollte das SELECT-Zugriffsrecht für alle untergeordneten Tabellen (und untergeordnete Sichten) der Tabelle (oder Sicht) erteilt werden, die nach dem Schlüsselwort ONLY angegeben ist.

Veraltete Profilregistrierdatenbank und Umgebungsvariablen in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Die folgenden Profilregistrierdatenbank- oder Umgebungsvariablen sind veraltet:

- DB2_VECTOR

Maßnahme: Diese Variablen werden nicht mehr benötigt.

Dienstprogramme und Tools

Aktueller EXPLAIN-Modus in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Der Typ des Sonderregisters CURRENT EXPLAIN MODE wurde von CHAR(8) in VARCHAR(254) geändert.

Symptom: Wenn die Anwendung davon ausgeht, daß der Typ immer noch CHAR(8) ist, kann der Wert von 254 auf 8 Byte abgeschnitten werden.

Maßnahme: Ändern Sie den Typ aller Host-Variablen, die das Sonderregister lesen, von CHAR(8) in VARCHAR(254).

Diese Änderung ist erforderlich, um zwei neue Werte im Sonderregister CURRENT EXPLAIN MODE unterzubringen. Diese neuen Werte sind EVALUATE INDEXES und RECOMMEND INDEXES.

Parameter USING und SORT BUFFER in DB2 Universal Database Version 6

WIN	UNIX	OS/2
-----	------	------

Änderung: Ab Version 6 werden die Parameter USING und SORT BUFFER des Befehls LOAD nicht mehr unterstützt. Diese Parameter werden ignoriert.

Symptom: Es wird eine Warnung zurückgegeben, die besagt, daß die Parameter USING und SORT BUFFER nicht länger unterstützt und vom Dienstprogramm LOAD ignoriert werden.

Maßnahme: Ignorieren Sie die Warnung. Weitere Informationen finden Sie im Handbuch *Versetzen von Daten Dienstprogramme und Referenz*.

Konnektivität und Koexistenz

Ersetzen von RUMBA durch PCOMM in DB2 Universal Database Version 6

WIN		
-----	--	--

Änderung: In Version 6 wird RUMBA durch PCOMM unter Windows NT, Windows 98 und Windows 95 (jedoch nicht unter Windows 3.1) ersetzt.

Symptom: Keine

Maßnahme: Keine

Konfigurationsparameter

Veraltete Datenbankkonfigurationsparameter

WIN	UNIX	OS/2
-----	------	------

Änderung: Die folgenden Datenbankkonfigurationsparameter sind veraltet:

- DL_NUM_BACKUP (ersetzt durch den Datenbankkonfigurationsparameter NUM_DB_BACKUP)

Maßnahme: Entfernen Sie alle Verweise auf diese Parameter aus Ihren Anwendungen.

Anhang E. Unterstützung von Landessprachen

Dieser Abschnitt enthält Informationen zur Unterstützung von Landessprachen (National Language Support, NLS), die von DB2 bereitgestellt wird, einschließlich Informationen zu den unterstützten Ländern, Sprachen und Codepages (codierten Zeichensätzen) sowie Angaben zu Konfiguration und Verwendung von DB2 NLS-Funktionen in Datenbanken und Anwendungen.

Unterstützung von Landescodes und Codepages

Tabelle 32 auf Seite 438 zeigt die von den Datenbank-Servern unterstützten Sprachen und codierten Zeichensätze sowie die Zuordnung dieser Werte zu den vom Datenbankmanager verwendeten Werten für den Landescode und die Codepage.

Es folgt eine Erläuterung zu jeder Spalte der Tabelle:

- Unter **Umsetztab.** wird die von IBM definierte Codepage gezeigt, wie sie vom codierten Zeichensatz des Betriebssystems zugeordnet wird.
- Unter **Gruppe** wird angegeben, ob eine Codepage Einzelbytezeichen („S“) oder Mehrbytezeichen („D“) enthält. Die Angabe „-n“ ist eine Zahl, die zur Erstellung einer Buchstabe-Zahl-Kombination verwendet wird. Übereinstimmende Kombinationen geben an, wo die Verbindung und die Umsetzung von DB2 erlaubt ist. Beispielsweise können alle „S-1“-Gruppen zusammenarbeiten.
- Unter **Codierter Zeichens.** wird der codierte Zeichensatz für die unterstützte Sprache aufgeführt. Der codierte Zeichensatz wird der DB2-Codepage zugeordnet.
- Unter **Geb.** ist die aus zwei Buchstaben bestehende Gebietskennung angegeben.
- Unter **Landescode** finden Sie den Landescode, den der Datenbankmanager intern zum Bereitstellen der länderspezifischen Unterstützung verwendet.
- Unter **Länderspez. Angaben** werden die Werte der vom Datenbankmanager unterstützten länderspezifischen Angaben angezeigt.
- Unter **BS** wird das Betriebssystem angezeigt, das die Sprachen und codierten Zeichensätze unterstützt.
- Unter **Landesname** wird der Name des Landes oder der Länder angezeigt.

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname	
----	-----	-----	--	---	----	-----	
437	S-1	IBM-437	AL	355	-	OS2	Albanien
850	S-1	IBM-850	AL	355	-	OS2	Albanien
819	S-1	ISO8859-1	AL	355	sq_AL	AIX	Albanien
850	S-1	IBM-850	AL	355	Sq_AL	AIX	Albanien
819	S-1	iso88591	AL	355	-	HP	Albanien
1051	S-1	roman8	AL	355	-	HP	Albanien
819	S-1	ISO8859-1	AL	355	-	Sun	Albanien
1252	S-1	1252	AL	355	-	WIN	Albanien
1275	S-1	1275	AL	355	-	Mac	Albanien
37	S-1	IBM-37	AL	355	-	HOST	Albanien
1140	S-1	IBM-1140	AL	355	-	HOST	Albanien
864	S-6	IBM-864	AA	785	-	OS2	Arabische Länder
1046	S-6	IBM-1046	AA	785	Ar_AA	AIX	Arabische Länder
1089	S-6	ISO8859-6	AA	785	ar_AA	AIX	Arabische Länder
1089	S-6	iso88596	AA	785	ar_SA.iso88596	HP	Arabische Länder
1256	S-6	1256	AA	785	-	WIN	Arabische Länder
420	S-6	IBM-420	AA	785	-	HOST	Arabische Länder
437	S-1	IBM-437	AU	61	-	OS2	Australien
850	S-1	IBM-850	AU	61	-	OS2	Australien
819	S-1	ISO8859-1	AU	61	en_AU	AIX	Australien
850	S-1	IBM-850	AU	61	En_AU	AIX	Australien
819	S-1	iso88591	AU	61	-	HP	Australien
1051	S-1	roman8	AU	61	-	HP	Australien
819	S-1	ISO8859-1	AU	61	en_AU	Sun	Australien
819	S-1	ISO8859-1	AU	61	en_AU	SCO	Australien
1252	S-1	1252	AU	61	-	WIN	Australien
1275	S-1	1275	AU	61	-	Mac	Australien
37	S-1	IBM-37	AU	61	-	HOST	Australien
1140	S-1	IBM-1140	AU	61	-	HOST	Australien
437	S-1	IBM-437	AT	43	-	OS2	Österreich
850	S-1	IBM-850	AT	43	-	OS2	Österreich
819	S-1	ISO8859-1	AT	43	ge_AT	AIX	Österreich
850	S-1	IBM-850	AT	43	Ge_AT	AIX	Österreich
819	S-1	iso88591	AT	43	-	HP	Österreich
1051	S-1	roman8	AT	43	-	HP	Österreich
819	S-1	ISO8859-1	AT	43	de_AT	SCO	Österreich
819	S-1	ISO-8859-1	AT	43	de_AT	Linux	Österreich
819	S-1	ISO8859-1	AT	43	de_AT	Sun	Österreich
1252	S-1	1252	AT	43	-	WIN	Österreich
1275	S-1	1275	AT	43	-	Mac	Österreich
37	S-1	IBM-37	AT	43	-	HOST	Österreich
1140	S-1	IBM-1140	AT	43	-	HOST	Österreich

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze (Forts.)

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname	
----	-----	-----	--	---	----	-----	
915	S-5	ISO8859-5	BY	375	-	OS2	Weißrußland
915	S-5	ISO8859-5	BY	375	be_BY	AIX	Weißrußland
1131	S-5	IBM-1131	BY	375	-	OS2	Weißrußland
1251	S-5	1251	BY	375	-	WIN	Weißrußland
1283	S-5	1283	BY	375	-	Mac	Weißrußland
1025	S-5	IBM-1025	BY	375	-	HOST	Weißrußland
437	S-1	IBM-437	BE	32	-	OS2	Belgien
850	S-1	IBM-850	BE	32	-	OS2	Belgien
819	S-1	ISO8859-1	BE	32	n _BE	AIX	Belgien
850	S-1	IBM-850	BE	32	N _BE	AIX	Belgien
819	S-1	iso88591	BE	32	-	HP	Belgien
819	S-1	ISO8859-1	BE	32	fr_BE	SCO	Belgien
819	S-1	ISO8859-1	BE	32	n _BE	SCO	Belgien
819	S-1	ISO-8859-1	BE	32	n _BE	Linux	Belgien
819	S-1	ISO8859-1	BE	32	n _BE	Sun	Belgien
1252	S-1	1252	BE	32	-	WIN	Belgien
1275	S-1	1275	BE	32	-	Mac	Belgien
500	S-1	IBM-500	BE	32	-	HOST	Belgien
1148	S-1	IBM-1148	BE	32	-	HOST	Belgien
855	S-5	IBM-855	BG	359	-	OS2	Bulgarien
915	S-5	ISO8859-5	BG	359	-	OS2	Bulgarien
915	S-5	ISO8859-5	BG	359	bg_BG	AIX	Bulgarien
915	S-5	iso88595	BG	359	bg_BG.iso88595	HP	Bulgarien
1251	S-5	1251	BG	359	-	WIN	Bulgarien
1283	S-5	1283	BG	359	-	Mac	Bulgarien
1025	S-5	IBM-1025	BG	359	-	HOST	Bulgarien
850	S-1	IBM-850	BR	55	-	OS2	Brasilien
850	S-1	IBM-850	BR	55	-	AIX	Brasilien
819	S-1	ISO8859-1	BR	55	pt_BR	AIX	Brasilien
819	S-1	ISO8859-1	BR	55	-	HP	Brasilien
819	S-1	ISO8859-1	BR	55	pt_BR	SCO	Brasilien
819	S-1	ISO8859-1	BR	55	pt_BR	Sun	Brasilien
819	S-1	ISO-8859-1	BR	55	pt_BR	Linux	Brasilien
1252	S-1	1252	BR	55	-	WIN	Brasilien
37	S-1	IBM-37	BR	55	-	HOST	Brasilien
1140	S-1	IBM-1140	BR	55	-	HOST	Brasilien

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze (Forts.)

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname	
----	-----	-----	--	---	----	-----	
850	S-1	IBM-850	CA	1	-	OS2	Kanada
850	S-1	IBM-850	CA	1	En_CA	AIX	Kanada
819	S-1	ISO8859-1	CA	1	en_CA	AIX	Kanada
819	S-1	iso88591	CA	1	fr_CA.iso88591	HP	Kanada
1051	S-1	roman8	CA	1	fr_CA.roman8	HP	Kanada
819	S-1	ISO8859-1	CA	1	en_CA	SCO	Kanada
819	S-1	ISO8859-1	CA	1	fr_CA	SCO	Kanada
819	S-1	ISO8859-1	CA	1	en_CA	Sun	Kanada
819	S-1	ISO8859-1	CA	1	en_CA	Sun	Kanada
819	S-1	ISO-8859-1	CA	1	en_CA	Linux	Kanada
1252	S-1	1252	CA	1	-	WIN	Kanada
1275	S-1	1275	CA	1	-	Mac	Kanada
37	S-1	IBM-37	CA	1	-	HOST	Kanada
1140	S-1	IBM-1140	CA	1	-	HOST	Kanada
863	S-1	IBM-863	CA	2	-	OS2	Kanada (Französisch)
1381	D-4	IBM-1381	CN	86	-	OS2	China (VRC)
1386	D-4	GBK	CN	86	-	OS2	China (VRC)
1383	D-4	IBM-eucCN	CN	86	zh_CN	AIX	China (VRC)
1386	D-4	GBK	CN	86	Zh_CN.GBK	AIX	China (VRC)
1383	D-4	hp15CN	CN	86	zh_CN.hp15CN	HP	China (VRC)
1383	D-4	eucCN	CN	86	zh_CN	SCO	China (VRC)
1383	D-4	eucCN	CN	86	zh_CN.eucCN	SCO	China (VRC)
1383	D-4	gb2312	CN	86	zh	Sun	China (VRC)
1381	D-4	IBM-1381	CN	86	-	WIN	China (VRC)
1386	D-4	GBK	CN	86	-	WIN	China (VRC)
935	D-4	IBM-935	CN	86	-	HOST	China (VRC)
1388	D-4	IBM-1388	CN	86	-	HOST	China (VRC)
852	S-2	IBM-852	HR	385	-	OS2	Kroatien
912	S-2	ISO8859-2	HR	385	hr_HR	AIX	Kroatien
912	S-2	iso88592	HR	385	hr_HR.iso88592	HP	Kroatien
912	S-2	ISO8859-2	HR	385	hr_HR.ISO8859-2	SCO	Kroatien
912	S-2	ISO-8859-2	HR	385	hr_HR	Linux	Kroatien
1250	S-2	1250	HR	385	-	WIN	Kroatien
1282	S-2	1282	HR	385	-	Mac	Kroatien
870	S-2	IBM-870	HR	385	-	HOST	Kroatien
852	S-2	IBM-852	CZ	421	-	OS2	Tschechische Republik
912	S-2	ISO8859-2	CZ	421	cs_CZ	AIX	Tschechische Republik
912	S-2	iso88592	CZ	421	cs_CZ.iso88592	HP	Tschechische Republik
912	S-2	ISO8859-2	CZ	421	cs_CZ.ISO8859-2	SCO	Tschechische Republik
912	S-2	ISO-8859-2	CZ	421	cs_CZ	Linux	Tschechische Republik
1250	S-2	1250	CZ	421	-	WIN	Tschechische Republik
1282	S-2	1282	CZ	421	-	Mac	Tschechische Republik
870	S-2	IBM-870	CZ	421	-	HOST	Tschechische Republik

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze (Forts.)

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname	
----	-----	-----	--	---	----	-----	
850	S-1	IBM-850	DK	45	-	OS2	Dänemark
819	S-1	ISO8859-1	DK	45	da_DK	AIX	Dänemark
850	S-1	IBM-850	DK	45	Da_DK	AIX	Dänemark
819	S-1	iso88591	DK	45	da_DK.iso88591	HP	Dänemark
1051	S-1	roman8	DK	45	da_DK.roman8	HP	Dänemark
819	S-1	ISO8859-1	DK	45	da	SCO	Dänemark
819	S-1	ISO8859-1	DK	45	da_DA	SCO	Dänemark
819	S-1	ISO8859-1	DK	45	da_DK	SCO	Dänemark
819	S-1	ISO8859-1	DK	45	da	Sun	Dänemark
819	S-1	ISO8859-1	DK	45	da	Sun	Dänemark
819	S-1	ISO-8859-1	DK	45	da_DK	Linux	Dänemark
1252	S-1	1252	DK	45	-	WIN	Dänemark
1275	S-1	1275	DK	45	-	Mac	Dänemark
277	S-1	IBM-277	DK	45	-	HOST	Dänemark
1142	S-1	IBM-1142	DK	45	-	HOST	Dänemark
922	S-10	IBM-922	EE	372	-	OS2	Estland
922	S-10	IBM-922	EE	372	Et_EE	AIX	Estland
922	S-10	IBM-922	EE	372	-	WIN	Estland
1122	S-10	IBM-1122	EE	372	-	HOST	Estland
437	S-1	IBM-437	FI	358	-	OS2	Finnland
850	S-1	IBM-850	FI	358	-	OS2	Finnland
819	S-1	ISO8859-1	FI	358	fi_FI	AIX	Finnland
850	S-1	IBM-850	FI	358	Fi_FI	AIX	Finnland
819	S-1	iso88591	FI	358	fi_FI.iso88591	HP	Finnland
819	S-1	ISO8859-1	FI	358	fi	SCO	Finnland
819	S-1	ISO8859-1	FI	358	fi_FI	SCO	Finnland
819	S-1	ISO8859-1	FI	358	sv_FI	SCO	Finnland
819	S-1	ISO8859-1	FI	358	-	Sun	Finnland
819	S-1	ISO-8859-1	FI	358	fi_FI	Linux	Finnland
1051	S-1	roman8	FI	358	-	HP	Finnland
1252	S-1	1252	FI	358	-	WIN	Finnland
1275	S-1	1275	FI	358	-	Mac	Finnland
278	S-1	IBM-278	FI	358	-	HOST	Finnland
1143	S-1	IBM-1143	FI	358	-	HOST	Finnland
855	S-5	IBM-855	MK	389	-	OS2	Mazedonien
915	S-5	ISO8859-5	MK	389	-	OS2	Mazedonien
915	S-5	ISO8859-5	MK	389	mk_MK	AIX	Mazedonien
915	S-5	iso88595	MK	389	-	HP	Mazedonien
1251	S-5	1251	MK	389	-	WIN	Mazedonien
1283	S-5	1283	MK	389	-	Mac	Mazedonien
1025	S-5	IBM-1025	MK	389	-	HOST	Mazedonien

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze (Forts.)

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname	
----	-----	-----	--	---	-----	-----	
437	S-1	IBM-437	FR	33	-	OS2	Frankreich
850	S-1	IBM-850	FR	33	-	OS2	Frankreich
819	S-1	ISO8859-1	FR	33	fr_FR	AIX	Frankreich
850	S-1	IBM-850	FR	33	Fr_FR	AIX	Frankreich
819	S-1	iso88591	FR	33	fr_FR.iso88591	HP	Frankreich
1051	S-1	roman8	FR	33	fr_FR.roman8	HP	Frankreich
819	S-1	ISO8859-1	FR	33	fr	Sun	Frankreich
819	S-1	ISO8859-1	FR	33	fr	SCO	Frankreich
819	S-1	ISO8859-1	FR	33	fr_FR	SCO	Frankreich
819	S-1	ISO-8859-1	FR	33	fr_FR	Linux	Frankreich
1252	S-1	1252	FR	33	-	WIN	Frankreich
1275	S-1	1275	FR	33	-	Mac	Frankreich
297	S-1	IBM-297	FR	33	-	HOST	Frankreich
1147	S-1	IBM-1147	FR	33	-	HOST	Frankreich
437	S-1	IBM-437	DE	49	-	OS2	Deutschland
850	S-1	IBM-850	DE	49	-	OS2	Deutschland
819	S-1	ISO8859-1	DE	49	de_DE	AIX	Deutschland
850	S-1	IBM-850	DE	49	De_DE	AIX	Deutschland
819	S-1	iso88591	DE	49	de_DE.iso88591	HP	Deutschland
1051	S-1	roman8	DE	49	de_DE.roman8	HP	Deutschland
819	S-1	ISO8859-1	DE	49	de	SCO	Deutschland
819	S-1	ISO8859-1	DE	49	de_DE	SCO	Deutschland
819	S-1	ISO8859-1	DE	49	de	Sun	Deutschland
819	S-1	ISO-8859-1	DE	49	de_DE	Linux	Deutschland
1252	S-1	1252	DE	49	-	WIN	Deutschland
1275	S-1	1275	DE	49	-	Mac	Deutschland
273	S-1	IBM-273	DE	49	-	HOST	Deutschland
1141	S-1	IBM-1141	DE	49	-	HOST	Deutschland
819	S-1	ISO8859-1	DE	49	De_DE.88591	SINIX	Deutschland
819	S-1	ISO8859-1	DE	49	De_DE.6937	SINIX	Deutschland
813	S-7	ISO8859-7	GR	30	-	OS2	Griechenland
869	S-7	IBM-869	GR	30	-	OS2	Griechenland
813	S-7	ISO8859-7	GR	30	e1_GR	AIX	Griechenland
813	S-7	iso88597	GR	30	e1_GR.iso88597	HP	Griechenland
813	S-7	ISO8859-7	GR	30	e1_GR.ISO8859-7	SCO	Griechenland
813	S-7	ISO-8859-7	GR	30	gr_GR	Linux	Griechenland
737	S-7	737	GR	30	-	WIN	Griechenland
1253	S-7	1253	GR	30	-	WIN	Griechenland
1280	S-7	1280	GR	30	-	Mac	Griechenland
423	S-7	IBM-423	GR	30	-	HOST	Griechenland
875	S-7	IBM-875	GR	30	-	HOST	Griechenland

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze (Forts.)

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname	
----	-----	-----	--	---	----	-----	
852	S-2	IBM-852	HU	36	-	OS2	Ungarn
912	S-2	ISO8859-2	HU	36	hu_HU	AIX	Ungarn
912	S-2	iso88592	HU	36	hu_HU.iso88592	HP	Ungarn
912	S-2	ISO8859-2	HU	36	hu_HU.IS08859-2	SCO	Ungarn
912	S-2	ISO-8859-2	HU	36	hu_HU	Linux	Ungarn
1250	S-2	1250	HU	36	-	WIN	Ungarn
1282	S-2	1282	HU	36	-	Mac	Ungarn
870	S-2	IBM-870	HU	36	-	HOST	Ungarn
850	S-1	IBM-850	IS	354	-	OS2	Island
819	S-1	ISO8859-1	IS	354	is_IS	AIX	Island
850	S-1	IBM-850	IS	354	Is_IS	AIX	Island
819	S-1	iso88591	IS	354	is_IS.iso88591	HP	Island
1051	S-1	roman8	IS	354	is_IS.roman8	HP	Island
819	S-1	ISO8859-1	IS	354	is	SCO	Island
819	S-1	ISO8859-1	IS	354	is_IS	SCO	Island
819	S-1	ISO8859-1	IS	354	-	Sun	Island
819	S-1	ISO-8859-1	IS	354	is_IS	Linux	Island
1252	S-1	1252	IS	354	-	WIN	Island
1275	S-1	1275	IS	354	-	Mac	Island
871	S-1	IBM-871	IS	354	-	HOST	Island
1149	S-1	IBM-1149	IS	354	-	HOST	Island
437	S-1	IBM-437	IE	353	-	OS2	Irland
850	S-1	IBM-850	IE	353	-	OS2	Irland
819	S-1	ISO8859-1	IE	353	en_IE	AIX	Irland
850	S-1	IBM-850	IE	353	En_IE	AIX	Irland
819	S-1	iso88591	IE	353	-	HP	Irland
1051	S-1	roman8	IE	353	-	HP	Irland
819	S-1	ISO8859-1	IE	353	en_IE	Sun	Irland
819	S-1	ISO8859-1	IE	353	en_IE.IS08859-1	SCO	Irland
819	S-1	ISO-8859-1	IE	353	en_IE	Linux	Irland
1252	S-1	1252	IE	353	-	WIN	Irland
1275	S-1	1275	IE	353	-	Mac	Irland
285	S-1	IBM-285	IE	353	-	HOST	Irland
1146	S-1	IBM-1146	IE	353	-	HOST	Irland
806	S-12	IBM-806	IN	91	hi_IN	-	Indien
1137	S-12	IBM-1137	IN	91	-	HOST	Indien
862	S-8	IBM-862	IL	972	-	OS2	Israel
916	S-8	ISO8859-8	IL	972	iw_IL	AIX	Israel
916	S-8	ISO-8859-8	IL	972	iw_IL	Linux	Israel
1255	S-8	1255	IL	972	-	WIN	Israel
424	S-8	IBM-424	IL	972	-	HOST	Israel

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze (Forts.)

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname	
----	-----	-----	--	---	----	-----	
437	S-1	IBM-437	IT	39	-	OS2	Italien
850	S-1	IBM-850	IT	39	-	OS2	Italien
819	S-1	ISO8859-1	IT	39	it_IT	AIX	Italien
850	S-1	IBM-850	IT	39	It_IT	AIX	Italien
819	S-1	iso88591	IT	39	it_IT.iso88591	HP	Italien
1051	S-1	roman8	IT	39	it_IT.roman8	HP	Italien
819	S-1	ISO8859-1	IT	39	it	SCO	Italien
819	S-1	ISO8859-1	IT	39	it_IT	SCO	Italien
819	S-1	ISO8859-1	IT	39	it	Sun	Italien
819	S-1	ISO-8859-1	IT	39	it_IT	Linux	Italien
1252	S-1	1252	IT	39	-	WIN	Italien
1275	S-1	1275	IT	39	-	Mac	Italien
280	S-1	IBM-280	IT	39	-	HOST	Italien
1144	S-1	IBM-1144	IT	39	-	HOST	Italien
932	D-1	IBM-932	JP	81	-	OS2	Japan
942	D-1	IBM-942	JP	81	-	OS2	Japan
943	D-1	IBM-943	JP	81	-	OS2	Japan
954	D-1	IBM-eucJP	JP	81	ja_JP	AIX	Japan
932	D-1	IBM-932	JP	81	Ja_JP	AIX	Japan
954	D-1	eucJP	JP	81	ja_JP.eucJP	HP	Japan
5039	D-1	SJIS	JP	81	ja_JP.SJIS	HP	Japan
954	D-1	eucJP	JP	81	ja	SCO	Japan
954	D-1	eucJP	JP	81	ja_JP	SCO	Japan
954	D-1	eucJP	JP	81	ja_JP.EUC	SCO	Japan
954	D-1	eucJP	JP	81	ja_JP.eucJP	SCO	Japan
954	D-1	eucJP	JP	81	ja	Sun	Japan
954	D-1	EUC-JP	JP	81	ja_JP	Linux	Japan
943	D-1	IBM-943	JP	81	-	WIN	Japan
930	D-1	IBM-930	JP	81	-	HOST	Japan
939	D-1	IBM-939	JP	81	-	HOST	Japan
5026	D-1	IBM-5026	JP	81	-	HOST	Japan
5035	D-1	IBM-5035	JP	81	-	HOST	Japan
1390	D-1		JP	81	-	HOST	Japan
1399	D-1		JP	81	-	HOST	Japan
949	D-3	IBM-949	KR	82	-	OS2	Korea, Süd-
970	D-3	IBM-eucKR	KR	82	ko_KR	AIX	Korea, Süd-
970	D-3	eucKR	KR	82	ko_KR.eucKR	HP	Korea, Süd-
970	D-3	eucKR	KR	82	ko_KR.eucKR	SGI	Korea, Süd-
970	D-3	5601	KR	82	ko	Sun	Korea, Süd-
1363	D-3	1363	KR	82	-	WIN	Korea, Süd-
933	D-3	IBM-933	KR	82	-	HOST	Korea, Süd-
1364	D-3	IBM-1364	KR	82	-	HOST	Korea, Süd-

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze (Forts.)

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname	
----	-----	-----	--	---	----	-----	
437	S-1	IBM-437	Lat	3	-	OS2	Lateinamerika
850	S-1	IBM-850	Lat	3	-	OS2	Lateinamerika
819	S-1	ISO8859-1	Lat	3	-	AIX	Lateinamerika
850	S-1	IBM-850	Lat	3	-	AIX	Lateinamerika
819	S-1	iso88591	Lat	3	-	HP	Lateinamerika
819	S-1	ISO8859-1	Lat	3	-	Sun	Lateinamerika
819	S-1	ISO-8859-1	Lat	3	-	Linux	Lateinamerika
1051	S-1	roman8	Lat	3	-	HP	Lateinamerika
1252	S-1	1252	Lat	3	-	WIN	Lateinamerika
1275	S-1	1275	Lat	3	-	Mac	Lateinamerika
284	S-1	IBM-284	Lat	3	-	HOST	Lateinamerika
1145	S-1	IBM-1145	Lat	3	-	HOST	Lateinamerika
921	S-10	IBM-921	LV	371	-	OS2	Lettland
921	S-10	IBM-921	LV	371	Lv_LV	AIX	Lettland
921	S-10	IBM-921	LV	371	-	WIN	Lettland
1112	S-10	IBM-1112	LV	371	-	HOST	Lettland
921	S-10	IBM-921	LT	370	-	OS2	Litauen
921	S-10	IBM-921	LT	370	Lt_LT	AIX	Litauen
921	S-10	IBM-921	LV	370	-	WIN	Litauen
1112	S-10	IBM-1112	LV	370	-	HOST	Litauen
437	S-1	IBM-437	NL	31	-	OS2	Niederlande
850	S-1	IBM-850	NL	31	-	OS2	Niederlande
819	S-1	ISO8859-1	NL	31	n1_NL	AIX	Niederlande
850	S-1	IBM-850	NL	31	N1_NL	AIX	Niederlande
819	S-1	iso88591	NL	31	n1_NL.iso88591	HP	Niederlande
1051	S-1	roman8	NL	31	n1_NL.roman8	HP	Niederlande
819	S-1	ISO8859-1	NL	31	n1	SCO	Niederlande
819	S-1	ISO8859-1	NL	31	n1_NL	SCO	Niederlande
819	S-1	ISO8859-1	NL	31	n1	Sun	Niederlande
819	S-1	ISO-8859-1	NL	31	n1_NL	Linux	Niederlande
1252	S-1	1252	NL	31	-	WIN	Niederlande
1275	S-1	1275	NL	31	-	Mac	Niederlande
37	S-1	IBM-37	NL	31	-	HOST	Niederlande
1140	S-1	IBM-1140	NL	31	-	HOST	Niederlande
850	S-1	IBM-850	NZ	64	-	OS2	Neuseeland
850	S-1	IBM-850	NZ	64	En_NZ	AIX	Neuseeland
819	S-1	ISO8859-1	NZ	64	en_NZ	AIX	Neuseeland
819	S-1	ISO8859-1	NZ	64	-	HP	Neuseeland
819	S-1	ISO8859-1	NZ	64	en_NZ	SCO	Neuseeland
819	S-1	ISO8859-1	NZ	64	en_NZ	Sun	Neuseeland
1252	S-1	1252	NZ	64	-	WIN	Neuseeland
37	S-1	IBM-37	NZ	64	-	HOST	Neuseeland
1140	S-1	IBM-1140	NZ	64	-	HOST	Neuseeland

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze (Forts.)

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname	
----	-----	-----	--	---	----	-----	
850	S-1	IBM-850	NO	47	-	OS2	Norwegen
819	S-1	ISO8859-1	NO	47	no_NO	AIX	Norwegen
850	S-1	IBM-850	NO	47	No_NO	AIX	Norwegen
819	S-1	iso88591	NO	47	no_NO.iso88591	HP	Norwegen
1051	S-1	roman8	NO	47	no_NO.roman8	HP	Norwegen
819	S-1	ISO8859-1	NO	47	no	SCO	Norwegen
819	S-1	ISO8859-1	NO	47	no_NO	SCO	Norwegen
819	S-1	ISO8859-1	NO	47	no	Sun	Norwegen
819	S-1	ISO-8859-1	NO	47	no_NO	Linux	Norwegen
1252	S-1	1252	NO	47	-	WIN	Norwegen
1275	S-1	1275	NO	47	-	Mac	Norwegen
277	S-1	IBM-277	NO	47	-	HOST	Norwegen
1142	S-1	IBM-1142	NO	47	-	HOST	Norwegen
852	S-2	IBM-852	PL	48	-	OS2	Polen
912	S-2	ISO8859-2	PL	48	pl_PL	AIX	Polen
912	S-2	iso88592	PL	48	pl_PL.iso88592	HP	Polen
912	S-2	ISO8859-2	PL	48	pl_PL.ISO8859-2	SCO	Polen
912	S-2	ISO-8859-2	PL	48	pl_PL	Linux	Polen
1250	S-2	1250	PL	48	-	WIN	Polen
1282	S-2	1282	PL	48	-	Mac	Polen
870	S-2	IBM-870	PL	48	-	HOST	Polen
860	S-1	IBM-860	PT	351	-	OS2	Portugal
850	S-1	IBM-850	PT	351	-	OS2	Portugal
819	S-1	ISO8859-1	PT	351	pt_PT	AIX	Portugal
850	S-1	IBM-850	PT	351	Pt_PT	AIX	Portugal
819	S-1	iso88591	PT	351	pt_PT.iso88591	HP	Portugal
1051	S-1	roman8	PT	351	pt_PT.roman8	HP	Portugal
819	S-1	ISO8859-1	PT	351	pt	SCO	Portugal
819	S-1	ISO8859-1	PT	351	pt_PT	SCO	Portugal
819	S-1	ISO8859-1	PT	351	pt	Sun	Portugal
819	S-1	ISO-8859-1	PT	351	pt_PT	Linux	Portugal
1252	S-1	1252	PT	351	-	WIN	Portugal
1275	S-1	1275	PT	351	-	Mac	Portugal
37	S-1	IBM-37	PT	351	-	HOST	Portugal
1140	S-1	IBM-1140	PT	351	-	HOST	Portugal
852	S-2	IBM-852	RO	40	-	OS2	Rumänien
912	S-2	ISO8859-2	RO	40	ro_RO	AIX	Rumänien
912	S-2	iso88592	RO	40	ro_RO.iso88592	HP	Rumänien
912	S-2	ISO8859-2	RO	40	ro_RO.ISO8859-2	SCO	Rumänien
912	S-2	ISO-8859-2	RO	40	ro_RO	Linux	Rumänien
1250	S-2	1250	RO	40	-	WIN	Rumänien
1282	S-2	1282	RO	40	-	Mac	Rumänien
870	S-2	IBM-870	RO	40	-	HOST	Rumänien

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze (Forts.)

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname	
----	-----	-----	--	---	----	-----	
866	S-5	IBM-866	RU	7	-	OS2	Rußland
915	S-5	ISO8859-5	RU	7	-	OS2	Rußland
915	S-5	ISO8859-5	RU	7	ru_RU	AIX	Rußland
915	S-5	iso88595	RU	7	ru_RU.iso88595	HP	Rußland
915	S-5	ISO8859-5	RU	7	ru_RU.IS08859-5	SCO	Rußland
915	S-5	ISO-8859-5	RU	7	ru_RU	Linux	Rußland
1251	S-5	1251	RU	7	-	WIN	Rußland
1283	S-5	1283	RU	7	-	Mac	Rußland
1025	S-5	IBM-1025	RU	7	-	HOST	Rußland
855	S-5	IBM-855	SP	381	-	OS2	Serbien/Montenegro
915	S-5	ISO8859-5	SP	381	-	OS2	Serbien/Montenegro
915	S-5	ISO8859-5	SP	381	sr_SP	AIX	Serbien/Montenegro
915	S-5	iso88595	SP	381	-	HP	Serbien/Montenegro
1251	S-5	1251	SP	381	-	WIN	Serbien/Montenegro
1283	S-5	1283	SP	381	-	Mac	Serbien/Montenegro
1025	S-5	IBM-1025	SP	381	-	HOST	Serbien/Montenegro
852	S-2	IBM-852	SK	422	-	OS2	Slowakei
912	S-2	ISO8859-2	SK	422	sk_SK	AIX	Slowakei
912	S-2	iso88592	SK	422	sk_SK.iso88592	HP	Slowakei
912	S-2	ISO8859-2	SK	422	sk_SK.IS08859-2	SCO	Slowakei
1250	S-2	1250	SK	422	-	WIN	Slowakei
1282	S-2	1282	SK	422	-	Mac	Slowakei
870	S-2	IBM-870	SK	422	-	HOST	Slowakei
852	S-2	IBM-852	SI	386	-	OS2	Slowenien
912	S-2	ISO8859-2	SI	386	sl_SI	AIX	Slowenien
912	S-2	iso88592	SI	386	sl_SI.iso88592	HP	Slowenien
912	S-2	ISO8859-2	SI	386	sl_SI.IS08859-2	SCO	Slowenien
912	S-2	ISO-8859-2	SI	386	sl_SI	Linux	Slowenien
1250	S-2	1250	SI	386	-	WIN	Slowenien
1282	S-2	1282	SI	386	-	Mac	Slowenien
870	S-2	IBM-870	SI	386	-	HOST	Slowenien
437	S-1	IBM-437	ZA	27	-	OS2	Südafrika
850	S-1	IBM-850	ZA	27	-	OS2	Südafrika
819	S-1	ISO8859-1	ZA	27	en_ZA	AIX	Südafrika
850	S-1	IBM-850	ZA	27	En_ZA	AIX	Südafrika
819	S-1	iso88591	ZA	27	-	HP	Südafrika
1051	S-1	roman8	ZA	27	-	HP	Südafrika
819	S-1	ISO8859-1	ZA	27	-	Sun	Südafrika
819	S-1	ISO8859-1	ZA	27	en_ZA.IS08859-1	SCO	Südafrika
1252	S-1	1252	ZA	27	-	WIN	Südafrika
1275	S-1	1275	ZA	27	-	Mac	Südafrika
285	S-1	IBM-285	ZA	27	-	HOST	Südafrika
1146	S-1	IBM-1146	ZA	27	-	HOST	Südafrika

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze (Forts.)

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname	
----	-----	-----	--	---	----	-----	
437	S-1	IBM-437	ES	34	-	OS2	Spanien
850	S-1	IBM-850	ES	34	-	OS2	Spanien
819	S-1	ISO8859-1	ES	34	es_ES	AIX	Spanien
850	S-1	IBM-850	ES	34	Es_ES	AIX	Spanien
819	S-1	iso88591	ES	34	es_ES.iso88591	HP	Spanien
1051	S-1	roman8	ES	34	es_ES.roman8	HP	Spanien
819	S-1	ISO8859-1	ES	34	es	Sun	Spanien
819	S-1	ISO8859-1	ES	34	es	SCO	Spanien
819	S-1	ISO8859-1	ES	34	es_ES	SCO	Spanien
819	S-1	ISO-8859-1	ES	34	es_ES	Linux	Spanien
1252	S-1	1252	ES	34	-	WIN	Spanien
1275	S-1	1275	ES	34	-	Mac	Spanien
284	S-1	IBM-284	ES	34	-	HOST	Spanien
1145	S-1	IBM-1145	ES	34	-	HOST	Spanien
437	S-1	IBM-437	SE	46	-	OS2	Schweden
850	S-1	IBM-850	SE	46	-	OS2	Schweden
819	S-1	ISO8859-1	SE	46	sv_SE	AIX	Schweden
850	S-1	IBM-850	SE	46	Sv_SE	AIX	Schweden
819	S-1	iso88591	SE	46	sv_SE.iso88591	HP	Schweden
1051	S-1	roman8	SE	46	sv_SE.roman8	HP	Schweden
819	S-1	ISO8859-1	SE	46	sv	SCO	Schweden
819	S-1	ISO8859-1	SE	46	sv_SE	SCO	Schweden
819	S-1	ISO8859-1	SE	46	sv	Sun	Schweden
819	S-1	ISO-8859-1	SE	46	sv_SE	Linux	Schweden
1252	S-1	1252	SE	46	-	WIN	Schweden
1275	S-1	1275	SE	46	-	Mac	Schweden
278	S-1	IBM-278	SE	46	-	HOST	Schweden
1143	S-1	IBM-1143	SE	46	-	HOST	Schweden
437	S-1	IBM-437	CH	41	-	OS2	Schweiz
850	S-1	IBM-850	CH	41	-	OS2	Schweiz
819	S-1	ISO8859-1	CH	41	de_CH	AIX	Schweiz
850	S-1	IBM-850	CH	41	De_CH	AIX	Schweiz
819	S-1	iso88591	CH	41	-	HP	Schweiz
1051	S-1	roman8	CH	41	-	HP	Schweiz
819	S-1	ISO8859-1	CH	41	de_CH	SCO	Schweiz
819	S-1	ISO8859-1	CH	41	fr_CH	SCO	Schweiz
819	S-1	ISO8859-1	CH	41	it_CH	SCO	Schweiz
819	S-1	ISO8859-1	CH	41	de_CH	Sun	Schweiz
819	S-1	ISO-8859-1	CH	41	de_CH	Linux	Schweiz
1252	S-1	1252	CH	41	-	WIN	Schweiz
1275	S-1	1275	CH	41	-	Mac	Schweiz
500	S-1	IBM-500	CH	41	-	HOST	Schweiz
1148	S-1	IBM-1148	CH	41	-	HOST	Schweiz

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze (Forts.)

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname	
----	-----	-----	--	---	----	-----	
938	D-2	IBM-938	TW	88	-	OS2	Taiwan
948	D-2	IBM-948	TW	88	-	OS2	Taiwan
950	D-2	big5	TW	88	-	OS2	Taiwan
950	D-2	big5	TW	88	Zh_TW	AIX	Taiwan
964	D-2	IBM-eucTW	TW	88	zh_TW	AIX	Taiwan
950	D-2	big5	TW	88	zh_TW.big5	HP	Taiwan
964	D-2	eucTW	TW	88	zh_TW.eucTW	HP	Taiwan
950	D-2	big5	TW	88	big5	Sun	Taiwan
964	D-2	cns11643	TW	88	zh_TW	Sun	Taiwan
950	D-2	big5	TW	88	-	WIN	Taiwan
937	D-2	IBM-937	TW	88	-	HOST	Taiwan
874	S-20	TIS620-1	TH	66	-	OS2	Thailand
874	S-20	TIS620-1	TH	66	Th_TH	AIX	Thailand
874	S-20	tis620	TH	66	th_TH.tis620	HP	Thailand
874	S-20	TIS620-1	TH	66	-	WIN	Thailand
838	S-20	IBM-838	TH	66	-	HOST	Thailand
857	S-9	IBM-857	TR	90	-	OS2	Türkei
920	S-9	ISO8859-9	TR	90	tr_TR	AIX	Türkei
920	S-9	iso88599	TR	90	tr_TR.iso88599	HP	Türkei
920	S-9	ISO8859-9	TR	90	tr_TR.ISO8859-9	SCO	Türkei
920	S-9	ISO-8859-9	TR	90	tr_TR	Linux	Türkei
1254	S-9	1254	TR	90	-	WIN	Türkei
1281	S-9	1281	TR	90	-	Mac	Türkei
1026	S-9	IBM-1026	TR	90	-	HOST	Türkei
437	S-1	IBM-437	GB	44	-	OS2	GB
850	S-1	IBM-850	GB	44	-	OS2	GB
819	S-1	ISO8859-1	GB	44	en_GB	AIX	GB
850	S-1	IBM-850	GB	44	En_GB	AIX	GB
819	S-1	iso88591	GB	44	en_GB.iso88591	HP	GB
1051	S-1	roman8	GB	44	en_GB.roman8	HP	GB
819	S-1	ISO8859-1	GB	44	en_UK	Sun	GB
819	S-1	ISO8859-1	GB	44	en_GB	SCO	GB
819	S-1	ISO8859-1	GB	44	en	SCO	GB
819	S-1	ISO-8859-1	GB	44	en_GB	Linux	GB
1252	S-1	1252	GB	44	-	WIN	GB
1275	S-1	1275	GB	44	-	Mac	GB
285	S-1	IBM-285	GB	44	-	HOST	GB
1146	S-1	IBM-1146	GB	44	-	HOST	GB
819	S-1	88591	GB	44	En_GB.88591	SINIX	GB
819	S-1	ISO8859-1	GB	44	En_GB.6937	SINIX	GB
1125	S-5	IBM-1125	UA	380	-	OS2	Ukraine
1124	S-5	IBM-1124	UA	380	uk_UA	AIX	Ukraine
1251	S-5	1251	UA	380	-	WIN	Ukraine
1123	S-5	IBM-1123	UA	380	-	HOST	Ukraine

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze (Forts.)

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname	
----	-----	-----	--	---	----	-----	
437	S-1	IBM-437	US	1	-	OS2	USA
850	S-1	IBM-850	US	1	-	OS2	USA
819	S-1	ISO8859-1	US	1	en_US	AIX	USA
850	S-1	IBM-850	US	1	En_US	AIX	USA
819	S-1	iso88591	US	1	en_US.iso88591	HP	USA
1051	S-1	roman8	US	1	en_US.roman8	HP	USA
819	S-1	ISO8859-1	US	1	en_US	Sun	USA
819	S-1	ISO8859-1	US	1	en_US	SGI	USA
819	S-1	ISO8859-1	US	1	en_US	SCO	USA
819	S-1	ISO-8859-1	US	1	en_US	Linux	USA
1252	S-1	1252	US	1	-	WIN	USA
1275	S-1	1275	US	1	-	Mac	USA
37	S-1	IBM-37	US	1	-	HOST	USA
1140	S-1	IBM-1140	US	1	-	HOST	USA
1163	S-11	IBM-1163	VN	84	-	OS2	Vietnam
1163	S-11	IBM-1163	VN	84	vi_VN	AIX	Vietnam
1258	S-11	1258	VN	84	-	WIN	Vietnam
1164	S-11	IBM-1164	VN	84	-	HOST	Vietnam

Übersicht der arabischen Länder (AA):

```

-----
/* Arabisch (Saudi-Arabien) */
/* Arabisch (Irak) */
/* Arabisch (Ägypten) */
/* Arabisch (Libyen) */
/* Arabisch (Algerien) */
/* Arabisch (Marokko) */
/* Arabisch (Tunesien) */
/* Arabisch (Oman) */
/* Arabisch (Jemen) */
/* Arabisch (Syrien) */
/* Arabisch (Jordanien) */
/* Arabisch (Libanon) */
/* Arabisch (Kuwait) */
/* Arabisch (Vereinigte Arabische Emirate) */
/* Arabisch (Bahrain) */
/* Arabisch (Katar) */

```

Übersicht über Englisch (US):

```

-----
/* Englisch (Jamaika) */
/* Englisch (Karibik) */

```

Tabelle 32. Unterstützte Sprachen und codierte Zeichensätze (Forts.)

Umsetz- tab.	Gruppe	Codierter Zeichens.	Landes- Geb. code	Länderspez. Angaben	BS	Landesname
----	-----	-----	--	---	----	-----

Übersicht über Lateinamerika (Lat):

```

-----
/* Spanisch (Mexiko) */
/* Spanisch (Guatemala) */
/* Spanisch (Costa Rica) */
/* Spanisch (Panama) */
/* Spanisch (Dominikanische Republik) */
/* Spanisch (Venezuela) */
/* Spanisch (Kolumbien) */
/* Spanisch (Peru) */
/* Spanisch (Argentinien) */
/* Spanisch (Ecuador) */
/* Spanisch (Chile) */
/* Spanisch (Uruguay) */
/* Spanisch (Paraguay) */
/* Spanisch (Bolivien) */

```

Anmerkung: Von der Codepage 950 für Solaris werden die folgenden Zeichen in IBM 950 nicht unterstützt:

Codebereich	Beschreibung	Sun Big-5	IBM Big-5
C6A1-C8FE	Symbole	Reservierter Bereich	Symbole
F9D6-F9FE	Erweiterung ETen	Reservierter Bereich	Erweiterung ETen
F286-F9A0	Von IBM ausgewählte Zeichen	Reservierter Bereich	Von IBM ausgewählt

Anmerkung: Die Unterstützung für das Euro-Symbol ist in dieser Version von DB2 UDB integriert. Die ANSI-Codepages von Microsoft Windows werden gemäß der neuesten Definition von Microsoft geändert. Diese enthalten nun Euro-Symbol an Position 0x80. Diese Position war bisher nicht definiert. Zusätzlich wurde die Definition von Codepage 850 geändert. Diese enthält anstelle des Zeichens "i ohne Punkt" (an Position 0xD5) das Euro-Symbol. DB2 UDB verwendet die neuen Definitionen dieser Codepages als Standardeinstellung für die Unterstützung des Euro-Symbols. Diese Implementierung ist der geeignete Standardwert für die derzeitigen DB2 UDB Kunden, die die Unterstützung des Euro-Symbols benötigen. Für Kunden, die das Euro-Symbol nicht benötigen, sollten sich keine Änderungen

ergeben. Wenn Sie jedoch die bisherigen Definitionen dieser Codepages weiter verwenden wollen, können Sie die folgenden Dateien:

- 12520850.cnv
- 08501252.cnv
- IBM00850.ucs
- IBM01252.ucs

aus dem Verzeichnis

sqllib/conv/alt/

in das Verzeichnis

sqllib/conv/

nach Ende der Installation kopieren. Es empfiehlt sich, die vorhandenen Dateien IBM01252.usc und IBM00850.ucs zu sichern, bevor Sie die Versionen ohne Unterstützung des Euro-Symbols darüber kopieren. Nach dem Kopieren der Dateien unterstützt DB2 UDB das Euro-Symbol nicht mehr.

Ableiten der Werte von Codepages

Die *Codepage der Anwendung* wird aus der aktiven Umgebung abgeleitet, wenn die Datenbankverbindung hergestellt wird. Wenn die Registrierungsvariable DB2CODEPAGE gesetzt ist, wird ihr Wert als Codepage der Anwendung verwendet. Es ist jedoch nicht notwendig, die Registrierungsvariable DB2CODEPAGE zu setzen, da DB2 den geeigneten Wert für die Codepage aus dem Betriebssystem ermittelt. Die Einstellung der Registrierungsvariablen DB2CODEPAGE auf falsche Werte kann zu unvorhersehbaren Ergebnissen führen.

Die *Codepage der Datenbank* wird aus dem Wert abgeleitet, der zur Zeit der Datenbankerstellung (explizit oder standardmäßig) angegeben wurde. Zum Beispiel wird im folgenden definiert, wie die *aktive Umgebung* in den unterschiedlichen Betriebsumgebungen bestimmt wird:

UNIX

Unter auf UNIX basierenden Betriebssystemen wird die aktive Umgebung durch die Einstellung für die länderspezifischen Angaben bestimmt, die Informationen zur Sprache, zu Gebiet (Territory) und zum codierten Zeichensatz umfassen.

OS/2

Unter OS/2 wird die primäre und sekundäre Codepage in der Datei CONFIG.SYS angegeben.

Mit dem Befehl **chcp** können Sie die Codepage anzeigen und innerhalb einer Sitzung dynamisch ändern.

Macintosh

Für das Macintosh Betriebssystem wird die Macintosh Codepage aus dem regionalen Versionscode des installierten Skripts abgeleitet, wenn die Registrierdatenbankvariable DB2CODEPAGE nicht gesetzt ist.

Windows

Für das Windows-Betriebssystem wird bei nicht gesetzter Registrierungsvariable DB2CODEPAGE die Windows-Codepage aus der Landes-ID abgeleitet, die im Wert iCountry im Abschnitt [int1] der Windows-Datei WIN.INI angegeben ist.

32-Bit-Windows-Betriebssysteme

Für alle 32-Bit-Windows-Betriebssysteme wird die Codepage aus der Einstellung für die ANSI-Codepage in der Registrierdatenbank abgeleitet, wenn die Registrierungsvariable DB2CODEPAGE nicht gesetzt ist.

Eine vollständige Liste der Umgebungszuordnungen für Codepage-Werte finden Sie in Tabelle 32 auf Seite 438.

Zeichensätze

Im allgemeinen wird der für eine Anwendung verfügbare Zeichensatz vom Datenbankmanager nicht eingeschränkt. Eine detaillierte Erläuterung zu Mehrbytezeichensätzen (MBCS), die von DB2 unterstützt werden, finden Sie im Handbuch *Application Development Guide*.

Zeichensatz für Kennungen

Der Standardzeichensatz, der für Datenbanknamen verwendet werden kann, besteht aus den großen und kleinen Einzelbytebuchstaben des lateinischen Alphabets (A...Z, a...z), den arabischen Ziffern (0...9) und dem Unterstrichungszeichen (_). Diese Liste wird noch um drei Sonderzeichen (#, @ und \$) erweitert, um Kompatibilität mit den Host-Datenbankprodukten zu gewährleisten. Allerdings sind diese Sonderzeichen in einer NLS-Umgebung mit Vorsicht zu verwenden, da sie nicht im unveränderlichen Zeichensatz für NLS-Hosts (EBCDIC) enthalten sind.

Beim Benennen von Datenbankobjekten (wie Tabellen und Sichten), Programmkennsätzen, Host-Variablen und Cursorsn können auch Elemente aus dem erweiterten Zeichensatz (z. B. Buchstaben mit diakritischen Zeichen) verwendet werden. Welche Zeichen im einzelnen verfügbar sind, hängt von

der verwendeten Codepage ab. Wenn Sie die Datenbank in einer Umgebung mit mehreren Codepages verwenden, müssen Sie darauf achten, daß alle Codepages die Elemente aus dem erweiterten Zeichensatz unterstützen, die Sie verwenden möchten. Informationen zu begrenzten Bezeichnern, die Zeichen außerhalb des erweiterten Zeichensatzes enthalten, jedoch in SQL-Anweisungen verwendet werden können, finden Sie im Handbuch *SQL Reference*.

Definition des erweiterten Zeichensatzes für DBCS-Kennungen

In DBCS-Umgebungen umfaßt der erweiterte Zeichensatz alle Zeichen des Standardzeichensatzes plus die folgenden Elemente:

- Alle Doppelbytezeichen in jeder DBCS-Codepage, mit Ausnahme des Doppelbyteleerzeichens, sind gültige Buchstaben.
- Das Doppelbyteleerzeichen ist ein Sonderzeichen.
- Die in jeder Misch-Codepage verfügbaren Einzelbytezeichen werden verschiedenen Kategorien zugeordnet:

Kategorie

Gültige Codepunkte in jeder Misch-Codepage

Ziffern

x30-39

Buchstaben

x23-24, x40-5A, x61-7A, xA6-DF (A6-DF nur für Codepages 932 und 942)

Sonderzeichen

Alle anderen gültigen Codepunkte für Einzelbytezeichen

Codieren von SQL-Anweisungen

Die Codierung von SQL-Anweisungen ist nicht sprachenabhängig. SQL-Schlüsselwörter können in Großbuchstaben, Kleinbuchstaben oder in gemischter Schreibweise eingegeben werden. Die Namen von Datenbankobjekten und Host-Variablen sowie Programmbezeichnungen in einer SQL-Anweisung dürfen keine Zeichen enthalten, die nicht in dem oben beschriebenen erweiterten Zeichensatz enthalten sind.

CCSID-Unterstützung für bidirektionale Zeichen

Die folgenden bidirektionalen Attribute sind erforderlich für die korrekte Behandlung bidirektionaler Daten auf unterschiedlichen Plattformen:

- Textart (LOGICAL oder VISUAL)
- Gestaltung (SHAPED oder UNSHAPED)
- Ausrichtung (RIGHT-TO-LEFT oder LEFT-TO-RIGHT)
- Zeichengestaltung (ARABIC oder HINDI)
- Symmetrische Auslagerung (YES oder NO)

Da die Standardeinstellungen auf den unterschiedlichen Plattformen nicht gleich sind, können Probleme auftreten, wenn DB2-Daten von einer Plattform auf eine andere übertragen werden. Das Windows-Betriebssystem verwendet beispielsweise Daten im Format LOGICAL UNSHAPED, während OS/390 im allgemeinen mit Daten im Format SHAPED VISUAL arbeitet. Daten, die ohne Unterstützung für bidirektionale Attribute von DB2 Universal Database für OS/390 auf DB2 UDB unter 32-Bit-Windows-Betriebssysteme übertragen werden, können daher falsch angezeigt werden.

Spezifische CCSIDs für bidirektionale Zeichen

DB2 unterstützt bidirektionale Datenattribute über spezielle bidirektionale IDs für codierte Zeichensätze (CCSIDs). Die folgenden CCSIDs für bidirektionale Zeichen wurden definiert und mit DB2 UDB implementiert:

CCSID (dez)	CCSID (hex)	Code- page	Zeichenfolge- Type
00420	x'01A4'	420	4
00424	x'01A8'	424	4
08612	x'21A4'	420	5
08616	x'21A8'	424	10
00856	x'0358'	856	5
00862	x'035E'	862	4
00864	x'0360'	864	5
00916	x'0394'	916	5
01046	x'0416'	1046	5
01089	x'0441'	1089	5
01255	x'04E7'	1255	5
01256	x'04E8'	1256	5
62208	x'F300'	856	4
62209	x'F301'	862	10
62210	x'F302'	916	4
62211	x'F303'	424	5
62213	x'F305'	862	5
62215	x'F307'	1255	4
62218	x'F30A'	864	4
62220	x'F30C'	856	6
62221	x'F30D'	862	6
62222	x'F30E'	916	6
62223	x'F30F'	1255	6
62224	x'F310'	420	6
62225	x'F311'	864	6
62226	x'F312'	1046	6
62227	x'F313'	1089	6
62228	x'F314'	1256	6
62229	x'F315'	424	8
62230	x'F316'	856	8
62231	x'F317'	862	8
62232	x'F318'	916	8
62233	x'F319'	420	8
62234	x'F31A'	420	9

62235	x'F31B'	424	6
62236	x'F31C'	856	10
62237	x'F31D'	1255	8
62238	x'F31E'	916	10
62239	x'F31F'	1255	10
62240	x'F320'	424	11
62241	x'F321'	856	11
62242	x'F322'	862	11
62243	x'F323'	916	11
62244	x'F324'	1255	11
62245	x'F325'	424	10
62246	x'F326'	1046	8
62247	x'F327'	1046	9
62248	x'F328'	1046	4
62249	x'F329'	1046	12
62250	x'F32A'	420	12

Dabei sind CDRA-Zeichenfolgearten wie folgt definiert:

Zeichen- folgeart	Text- art	Zeichen- gestaltung	Ausrichtung	Gestaltung	Symmetrische Auslagerungsfunktion
4	VISUAL	PASSTHRU	LTR	SHAPED	OFF
5	IMPLICIT	ARABIC	LTR	UNSHAPED	ON
6	IMPLICIT	ARABIC	RTL	UNSHAPED	ON
7(*)	VISUAL	ARABIC	CONTEXTUAL(*)	UNSHAPED-Lig	OFF
8	VISUAL	ARABIC	RTL	SHAPED	OFF
9	VISUAL	PASSTHRU	RTL	SHAPED	ON
10	IMPLICIT	PASSTHRU	CONTEXTUAL-L	UNSHAPED	ON
11	IMPLICIT	PASSTHRU	CONTEXTUAL-R	UNSHAPED	ON
12	IMPLICIT	ARABIC	RTL	SHAPED	ON

Anmerkung: (*) Die Feldausrichtung ist von links nach rechts (LTR), wenn das erste alphabetische Zeichen ein lateinisches Zeichen ist, und von rechts nach links (RTL), wenn es ein bidirektionales Zeichen ist. Zeichen sind UNSHAPED, LamAlef-Ligaturen werden jedoch beibehalten und nicht in Bestandteile getrennt.

Implementierung von Unterstützung bidirektionaler Zeichen in DB2 Universal Database

Bidirektionale Layoutumsetzungen werden in DB2 Universal Database mit den neuen CCSID-Definitionen implementiert. Bei den neuen spezifischen CCSIDs für bidirektionale Zeichen werden Layoutumsetzungen anstatt oder zusätzlich zur Umsetzung von Codepages durchgeführt. Damit diese Unterstützung verwendet werden kann, muß die Registrierungsvariable DB2BIDI auf YES gesetzt werden. Standardmäßig ist diese Variable nicht gesetzt. Sie wird vom Server für alle Umsetzungen verwendet und kann nur gesetzt werden, wenn der Server gestartet ist. Das Einstellen von DB2BIDI auf YES kann sich wegen der zusätzlichen Prüfung und der Layoutumsetzungen auf die Leistung auswirken.

Wenn Sie eine bestimmte CCSID für bidirektionale Zeichen in einer Nicht-DRDA-Umgebung angeben wollen, wählen Sie die CCSID (aus obiger Tabelle) aus, die den Kenndaten Ihres Clients entspricht, und stellen DB2CODEPAGE auf diesen Wert ein. Wenn Sie bereits eine Verbindung zur Datenbank haben, müssen Sie den Befehl TERMINATE absetzen, und die Verbindung anschließend erneut herstellen, damit die Einstellung von DB2CODEPAGE wirksam wird. Wenn Sie eine CCSID auswählen, die für die Codepage oder die Zeichenfolgeart Ihrer Client-Plattform nicht geeignet ist, erhalten Sie möglicherweise unerwartete Ergebnisse. Wenn Sie eine inkompatible CCSID (z.B. CCSID für Hebräisch zur Verbindung zu einer arabischen Datenbank) verwenden oder wenn DB2BIDI nicht für den Server eingestellt wurde, erhalten Sie eine Fehlermeldung, wenn Sie versuchen, die Verbindung herzustellen.

Wenn bei DRDA-Umgebungen die HOST-EBCDIC-Plattform diese CCSIDs für bidirektionale Zeichen ebenfalls unterstützt, müssen Sie nur den Wert für DB2CODEPAGE einstellen. Wenn die HOST-Plattform diese CCSIDs jedoch nicht unterstützt, müssen Sie eine CCSID-Überschreibung für den HOST-Datenbankserver angeben, zu dem Sie eine Verbindung herstellen. Dies ist notwendig, da in einer DRDA-Umgebung Umsetzungen von Codepages und Layoutumsetzungen vom Empfänger der Daten durchgeführt werden. Wenn der HOST-Server diese CCSIDs jedoch nicht unterstützt, wird keine Layoutumsetzung für die Daten durchgeführt, die er von DB2 UDB empfängt. Wenn Sie eine CCSID-Überschreibung verwenden, führt der Client unter DB2 UDB die Layoutumsetzung auch für die abgehenden Daten durch. Weitere Informationen zur Einstellung einer CCSID-Überschreibung finden Sie im *DB2 Connect Benutzerhandbuch*.

Die CCSID-Überschreibung wird nicht unterstützt, wenn die HOST-EBCDIC-Plattform der Client und DB2 UDB der Server ist.

Implementierung von Unterstützung bidirektionaler Zeichen in DB2 Connect

Wenn Daten zwischen DB2 Connect und einer Datenbank auf dem Server ausgetauscht werden, ist es normalerweise die Empfängermaschine, die die Umsetzung der ankommenden Daten ausführt. Dasselbe würde normalerweise auch auf bidirektionale Layoutumsetzungen zutreffen, die zusätzlich zur üblichen Codepage-Umsetzung erfolgen. DB2 Connect kann wahlfrei bidirektionale Layoutumsetzungen an den Daten ausführen, die gerade an die Server-Datenbank gesendet werden, zusätzlich zu den Daten, die von der Server-Datenbank empfangen wurden.

Damit DB2 Connect bidirektionale Layoutumsetzungen an abgehenden Daten für eine Server-Datenbank ausführen kann, muß die bidirektionale CCSID der Server-Datenbank überschrieben werden. Dies wird durch die Verwendung des Parameters BIDI im Feld PARMS des DCS-Datenbankverzeichniseintrags für die Server-Datenbank erreicht.

Anmerkung: Wenn Sie möchten, daß DB2 Connect eine Layoutumsetzung an den Daten ausführt, die gerade an die DB2-Host-Datenbank gesendet werden, müssen Sie ihre CCSID nicht überschreiben, aber Sie müssen den Parameter BIDI dem Feld PARMS des DCS-Datenbankverzeichnisses hinzufügen. In diesem Fall sollten Sie als CCSID die Standard-CCSID der DB2-Host-Datenbank angeben.

Der Parameter BIDI muß als neunter Parameter im Feld PARMS zusammen mit der bidirektionalen CCSID, mit der Sie die bidirektionale Standard-CCSID der Server-Datenbank überschreiben wollen, angegeben werden:

```
" , , , , , , , BIDI=xyz"
```

Dabei gilt folgendes: xyz ist die CCSID-Überschreibung.

Anmerkung: Die Registrierungsvariable DB2BIDI muß auf den Wert YES gesetzt werden, damit der Parameter BIDI wirksam werden kann.

Eine Liste der unterstützten bidirektionalen CCSIDs finden Sie zusammen mit ihren Zeichenfolgertypen in „Spezifische CCSIDs für bidirektionale Zeichen“ auf Seite 455.

Die Verwendung dieser Einrichtung wird am besten an einem Beispiel erläutert.

Angenommen, Sie haben einen hebräischen DB2-Client, der CCSID 62213 (bidirektionale Zeichenfolgertyp 5) ausführt, und Sie wollen auf eine DB2-Host-Datenbank zugreifen, die CCSID 00424 (bidirektionale Zeichenfolgertyp 4) ausführt. Sie wissen aber, daß die in der DB2-Host-Datenbank enthaltenen Daten auf der CCSID 08616 (bidirektionale Zeichenfolgertyp 6) basieren.

Hier gibt es zwei Probleme: Das erste Problem besteht darin, daß die DB2-Host-Datenbank den Unterschied zwischen den bidirektionalen Zeichenfolgertypen mit den CCSIDs 00424 und 08616 nicht kennt. Das zweite Problem ist, daß die DB2-Host-Datenbank die DB2-Client-CCSID (62213) nicht erkennt. Sie unterstützt nur CCSID 00862, die auf derselben Codepage wie CCSID 62213 basiert.

Sie müssen zunächst sicherstellen, daß die an die DB2-Host-Datenbank gesendeten Daten das Format der bidirektionalen Zeichenfolgertyp 6 haben. Ferner müssen Sie DB2 Connect bekanntgeben, daß eine bidirektionale Umsetzung der Daten ausgeführt werden soll, die von der DB2-Host-Datenbank empfangen werden. Sie müssen den folgenden Katalogisierungsbefehl für die DB2-Host-Datenbank verwenden:

```
db2 catalog dcs database nydb1 as telaviv parms " , , , , , , , BIDI=08616"
```

Mit diesem Befehl wird DB2 Connect mitgeteilt, daß die CCSID 00424 der DB2-Host-Datenbank mit der CCSID 08616 überschrieben werden soll. Diese Überschreibung wird wie folgt verarbeitet:

1. DB2 Connect stellt eine Verbindung zur DB2-Host-Datenbank mit CCSID 00862 her.
2. DB2 Connect führt eine bidirektionale Layoutumsetzung der Daten aus, die gerade an die DB2-Host-Datenbank *gesendet* werden sollen. CCSID 62213 (bidirektionale Zeichenfolgeart 5) wird in die CCSID 62221 (bidirektionale Zeichenfolgeart 6) umgesetzt.
3. DB2 Connect führt eine bidirektionale Layoutumsetzung der Daten aus, die von der DB2-Host-Datenbank *empfangen* werden. Diese Umsetzung erfolgt von CCSID 08616 (bidirektionale Zeichenfolgeart 6) in CCSID 62213 (bidirektionale Zeichenfolgeart 5).

Anmerkung: In einigen Fällen kann die Verwendung einer bidirektionalen CCSID die SQL-Abfrage selbst so ändern, daß sie nicht mehr vom DB2-Server erkannt wird. Sie sollten vor allem die Verwendung von CCSIDs mit IMPLICIT CONTEXTUAL und IMPLICIT RIGHT-TO-LEFT vermeiden, wenn eine andere Zeichenfolgeart verwendet werden kann. CONTEXTUAL-CCSIDs können zu unvorhersehbaren Ergebnissen führen, wenn die SQL-Abfrage Zeichenfolgen in Anführungszeichen enthält. Vermeiden Sie daher in SQL-Anweisungen Zeichenfolgen in Anführungszeichen. Verwenden Sie nach Möglichkeit überall Host-Variablen.

Wenn eine bestimmte bidirektionale CCSID Probleme hervorruft, die nicht durch das Befolgen dieser Empfehlungen korrigiert werden können, setzen Sie DB2BIDI auf den Wert NO.

Sortierfolge

Der Datenbankmanager vergleicht Zeichendaten mit Hilfe einer *Sortierfolge*. Dies ist eine Reihenfolge für eine Zeichengruppe, mit der festgelegt wird, ob ein bestimmtes Zeichen vor, nach oder gleichwertig mit einem anderen angeordnet wird. Mit einer Sortierfolge kann beispielsweise angegeben werden, daß Großbuchstaben und Kleinbuchstaben gleich bewertet werden sollen.

Die Sortierfolge wird bei der Datenbankeinstellung angegeben und kann später nicht mehr geändert werden.

Der Datenbankmanager ermöglicht das Erstellen von Datenbanken mit benutzerdefinierten Sortierfolgen über die API (Anwendungsprogrammierschnittstelle). Informationen zur Implementierung einer benutzerdefinierten Sortierfolgentabelle finden Sie im Handbuch *Application Development Guide*.

Anmerkung: Daten aus Zeichenfolgen, die mit dem Attribut FOR BIT DATA definiert wurden, und BLOB-Daten werden mit der Binärsortierfolge sortiert.

Allgemeine Überlegungen

Wenn eine Sortierfolge definiert ist, werden alle zukünftigen Zeichenvergleiche für diese Datenbank mit dieser Sortierfolge ausgeführt. Mit Ausnahme von Zeichendaten, die mit dem Attribut FOR BIT DATA oder BLOB definiert wurden, wird die Sortierfolge für alle SQL-Vergleiche und Klauseln ORDER BY verwendet, sowie für das Erstellen von Indizes und Statistiken. Weitere Informationen zur Verwendung der Datenbanksortierfolge finden Sie im Abschnitt über Vergleiche von Zeichenfolgen im Handbuch *SQL Reference*.

In folgenden Fällen können Probleme auftreten:

- Eine Anwendung mischt sortierte Daten aus einer Datenbank mit Anwendungsdaten, die mit einer anderen Sortierfolge sortiert wurden.
- Eine Anwendung mischt sortierte Daten aus einer Datenbank mit sortierten Daten aus einer anderen Datenbank, wobei beide unterschiedliche Sortierfolgen haben.
- Eine Anwendung geht von Voraussetzungen für die sortierten Daten aus, die für die betreffende Sortierfolge nicht stimmen. Für eine bestimmte Sortierfolge kann beispielsweise gelten, daß Zahlen nach Buchstaben angeordnet werden, oder nicht.

Ein letzter Punkt, der zu beachten ist, besteht darin, daß die Ergebnisse eines Sortiervorgangs, der auf einem direkten Vergleich von Zeichencodepunkten beruht, nur mit den Ergebnissen einer Abfrage übereinstimmen, die mit einer Identitätssortierfolge geordnet wurden.

Überlegungen zu zusammengeschlossenen Datenbanken

Die von Ihnen ausgewählte Datenbanksortierfolge kann sich auf die Leistung des Systems zusammengeschlossener Datenbanken auswirken. Wenn eine Datenquelle dieselbe Sortierfolge wie die zusammengeschlossene DB2-Datenbank verwendet, kann DB2 die von der Sortierfolge abhängige Verarbeitung von Zeichendaten in die Datenquelle auslagern. Verwendet eine Datenquelle eine von DB2 abweichende Sortierfolge, werden die Daten abgerufen, und die gesamte von der Sortierfolge abhängige Verarbeitung von Zeichendaten erfolgt lokal (dies kann die Leistung beeinträchtigen).

Beachten Sie die folgenden Punkte, wenn sie ermitteln wollen, ob DB2 und eine Datenquelle dieselbe Sortierfolge verwenden:

- Unterstützung von Landessprachen
Die Sortierfolge bezieht sich auf die auf einem Server unterstützte Sprache. Vergleichen Sie die NLS-Informationen von DB2 und der Datenquelle.
- Kenndaten der Datenquelle

Manche Datenquellen werden mit Sortierfolgen erstellt, die nicht zwischen Groß-/Kleinschreibung unterscheiden. Dadurch können von der Sortierfolge abhängige Operationen zu anderen Ergebnissen führen als in DB2.

- Anpassung

Einige Datenquellen bieten mehrere Optionen für Sortierfolgen bzw. ermöglichen deren Anpassung.

Wählen Sie die Sortierfolge für eine zusammengeschlossene DB2-Datenbank entsprechend der Mischung der Datenquellen aus, auf welche diese Datenbank zugreift. Beispiel:

- Wenn eine DB2-Datenbank überwiegend auf Oracle-Datenbanken mit derselben Codepage (NLS) wie DB2 zugreift, geben Sie bei der Erstellung der Datenbank die Identitätssortierfolge an (Oracle-Datenbanken verwenden eine gleichwertige Sortierfolge).
- Wenn eine DB2-Datenbank nur auf DB2-UDB-Datenbanken zugreift, stellen Sie sicher, daß die Sortierfolgenwerte übereinstimmen.

Informationen zur Einrichtung einer MVS-Sortierfolge finden Sie in den Beispielen unter der Beschreibung der API **sqlcrea** (Create Database) zur Erstellung von Datenbanken im Handbuch *Administrative API Reference*. Diese Beispiele enthalten Sortierfolgetabellen für die EBCDIC-Codepages 500, 37 und 5026/5035.

Denken Sie nach dem Festlegen der Sortierfolge für die DB2-Datenbank daran, die Server-Option *collating_sequence* für jeden Datenquellen-Server zu definieren. Diese Option gibt an, ob die Sortierfolge eines gegebenen Datenquellen-Servers mit der Sortierfolge der DB2-Datenbank übereinstimmt.

Setzen Sie die Option *collating_sequence* auf den Wert "Y", wenn die Sortierfolgen übereinstimmen. Diese Einstellung ermöglicht dem Optimierungsprogramm von DB2, von der Sortierfolge abhängige Verarbeitung in der Datenquelle vorzunehmen, was zu einer höheren Leistung führen kann. Unterscheidet sich jedoch die Sortierfolge der Datenquelle von der in DB2, erhalten Sie möglicherweise falsche Ergebnisse erhalten. Wenn Ihr Plan z. B. Mischverknüpfungen verwendet, lagert das DB2-Optimierungsprogramm Sortieroperationen soweit möglich in die Datenquellen aus. Wenn die Datenquelle eine andere Sortierfolge hat, können die Verknüpfungsergebnisse falsch sein.

Setzen Sie die Option *collating_sequence* auf den Wert "N", wenn die Sortierfolgen nicht übereinstimmen. Verwenden Sie diesen Wert, wenn sich die Sortierfolgen der Datenquelle von DB2 unterscheiden oder wenn die Sortieroperationen der Datenquelle möglicherweise die Groß-/Kleinschreibung nicht beachten. In einer Datenquelle mit einer deutschen Codepage ohne Beachtung von Groß-/Kleinschreibung z. B. werden die Begriffe KARUSSELL, KaRuSel

und karussell als identisch betrachtet. Setzen Sie die Option *collating_sequence* auf "N", wenn Sie nicht sicher sind, daß die Sortierfolgen der Datenquelle und DB2 übereinstimmen.

Werte für Datum und Uhrzeit

Die Datentypen für Datum und Uhrzeit werden im folgenden beschrieben. Werte für Datum und Uhrzeit können zwar in bestimmten arithmetischen Operationen und Zeichenfolgeoperationen verwendet werden und sind auch mit bestimmten Zeichenfolgen kompatibel, allerdings handelt es sich bei diesen Werten weder um Zeichenfolgen noch um Ziffern.

Datum

Datum ist ein Wert, der aus drei Teilen besteht (Jahr, Monat und Tag). Der Wertebereich für das Jahr liegt zwischen 0001 und 9999. Der Wertebereich für den Monat liegt zwischen 1 und 12. Der Wertebereich für den Tag liegt zwischen 1 und *x*, wobei *x* vom Monat abhängig ist.

Intern wird ein Datum durch eine Zeichenfolge aus 4 Byte dargestellt. Jedes dieser Byte besteht aus 2 gepackten Dezimalziffern. Die ersten zwei Byte stehen für das Jahr, das dritte Byte für den Monat und das letzte Byte für den Tag.

Die Länge einer Datumsspalte (DATE), wie im SQL-Deskriptorbereich beschrieben, beträgt 10 Byte. Dies ist die geeignete Länge für die Darstellung des Werts als Zeichenfolge.

Uhrzeit

Uhrzeit ist ein Wert, der aus drei Teilen besteht (Stunde, Minute und Sekunde) und eine Uhrzeit in der 24-Stunden-Zeiteinteilung bezeichnet. Der Wertebereich für die Stunde liegt zwischen 0 und 24. Der Wertebereich für die anderen Teile liegt zwischen 0 und 59. Wenn die Stunde 24 ist, sind die Minuten- und Sekundenangaben gleich null.

Intern wird Zeit durch eine Zeichenfolge aus 3 Byte dargestellt. Jedes dieser Byte besteht aus 2 gepackten Dezimalziffern. Das erste Byte steht für die Stunde, das zweite Byte für die Minute und das letzte Byte für die Sekunde.

Die Länge einer Zeitspalte (TIME), wie im SQL-Deskriptorbereich beschrieben, beträgt 8 Byte. Dies ist die geeignete Länge für die Darstellung des Werts als Zeichenfolge.

Zeitmarke

Eine *Zeitmarke* ist ein aus sieben Teilen bestehender Wert (Jahr, Monat, Tag, Stunde, Minute, Sekunde, Mikrosekunde), der einen Wert für ein Datum und eine Uhrzeit (wie oben definiert) bezeichnet, nur daß hier für die Zeit zusätzlich Mikrosekunden angegeben werden.

Intern wird die Zeitmarke durch eine Zeichenfolge aus 10 Byte dargestellt. Jedes dieser Byte besteht aus 2 gepackten Dezimalziffern. Die ersten vier Byte stehen für das Datum, die nächsten drei Byte für die Uhrzeit und die letzten drei Byte für die Mikrosekunden.

Die Länge einer Zeitmarkenspalte (TIMESTAMP), wie im SQL-Deskriptorbereich beschrieben, beträgt 26 Byte. Dies ist die geeignete Länge für die Darstellung des Werts als Zeichenfolge.

Darstellung von Werten für Datum und Uhrzeit als Zeichenfolge

Werte mit dem Datentyp DATE, TIME oder TIMESTAMP werden intern in einer Form dargestellt, die sie für den SQL-Benutzer transparent ist. Datum, Uhrzeiten und Zeitmarken können allerdings auch als Zeichenfolgen dargestellt werden. Diese Darstellungen betreffen den SQL-Benutzer direkt, da es keine Konstanten oder Variablen mit den Datentypen DATE, TIME oder TIMESTAMP gibt. Daher muß ein Wert für Datum und Uhrzeit, der abgerufen werden soll, einer Zeichenfolgenvariablen zugeordnet werden. Die Darstellung als Zeichenfolge entspricht normalerweise dem Standardformat der Werte für Datum und Uhrzeit, die dem Landescode des Clients zugeordnet sind, sofern diese nicht durch Angabe der Formatoption "F" bei der Vorkompilierung des Programms oder beim Binden an die Datenbank überschrieben werden. Eine Liste der Zeichenfolgenformate für die verschiedenen Landescodes können Sie Tabelle 35 auf Seite 466 entnehmen.

Wenn eine gültige Zeichenfolge zur Darstellung des Werts für Datum und Uhrzeit in einer Operation mit einem internen Wert für Datum und Uhrzeit verwendet wird, wird die Zeichenfolge vor Ausführung der Operation in das interne Format für Datum, Zeit oder Zeitmarke umgesetzt. Gültige Zeichenfolgendarstellungen von Werten für Datum und Uhrzeit werden in den folgenden Abschnitten definiert.

Zeichenfolgen für das Datum

Zur Darstellung des Datums können Zeichenfolgen verwendet werden, die mit einer Ziffer beginnen und aus mindestens 8 Zeichen bestehen. Abschließende Leerzeichen sind zulässig. Führende Nullen bei den Monats- und Tagesangaben des Datums können wegfallen.

Gültige Zeichenfolgenformate für Datumsangaben sind in Tabelle 33 aufgelistet. Für jedes Format wird der Name, eine zugehörige Abkürzung und ein Verwendungsbeispiel angegeben.

Tabelle 33. Formate zur Darstellung des Datums als Zeichenfolge

Formatname	Abkürzung	Datumsformat	Beispiel
International Standards Organization	ISO	jjj-mm-tt	1994-10-27

Tabelle 33. Formate zur Darstellung des Datums als Zeichenfolge (Forts.)

Formatname	Abkürzung	Datumsformat	Beispiel
IBM USA Standard	USA	mm/tt/iiii	10/27/1994
IBM Europäischer Standard	EUR	tt.mm.iiii	27.10.1994
Japanischer Industriestandard	JIS	iiii-mm-tt	1994-10-27
Standortabhängig (lokal)	LOC	Abhängig vom Landescode der Datenbank	—

Zeichenfolgen für die Uhrzeit

Zur Darstellung der Uhrzeit können Zeichenfolgen verwendet werden, die mit einer Ziffer beginnen und aus mindestens 4 Zeichen bestehen. Abschließende Leerzeichen sind zulässig. Eine führende Null kann bei den Stundenangaben wegfallen, und die Sekunden können ganz wegfallen. Wenn Sie die Sekunden nicht angeben, wird eine implizite Angabe von 0 Sekunden angenommen. Dementsprechend ist 13.30 äquivalent zu 13.30.00.

Gültige Zeichenfolgenformate für Uhrzeitangaben sind in Tabelle 34 aufgelistet. Für jedes Format wird der Name, eine zugehörige Abkürzung und ein Verwendungsbeispiel angegeben.

Tabelle 34. Formate zur Darstellung der Uhrzeit als Zeichenfolge

Formatname	Abkürzung	Zeitformat	Beispiel
International Standards Organization	ISO	hh.mm.ss	13.30.05
IBM USA Standard	USA	hh:mm AM oder PM	1:30 PM
IBM Europäischer Standard	EUR	hh.mm.ss	13.30.05
Japanischer Industriestandard	JIS	hh:mm:ss	13:30:05
Standortabhängig (lokal)	LOC	Abhängig vom Landescode der Anwendung	—

Anmerkungen:

1. In den Uhrzeitformaten ISO, EUR oder JIS ist .ss (bzw. :ss) wahlfrei.
2. Im Uhrzeitformat USA kann die Angabe der Minuten weggelassen werden, was implizit eine Angabe von 00 Minuten bedeutet. Demnach ist 1 PM äquivalent zu 1:00 PM.
3. Im Uhrzeitformat USA kann die Stundenangabe nicht größer als 12 sein und darf, abgesehen vom Spezialfall 00:00 AM, nicht 0 sein. Bei Verwen-

derung des ISO-Formats mit der 24-Stunden-Zeiteinteilung gelten die folgenden Entsprechungen zwischen dem Format USA und der 24-Stunden-Zeiteinteilung:

- 12:01 AM bis 12:59 AM entspricht 00.01.00 bis 00.59.00.
- 01:00 AM bis 11:59 AM entspricht 01.00.00 bis 11.59.00.
- 12:00 PM (Mittag) bis 11:59 PM entspricht 12.00.00 bis 23.59.00.
- 12:00 AM (Mitternacht) entspricht 24.00.00 und 00:00 AM (Mitternacht) entspricht 00.00.00.

Zeichenfolgen für Zeitmarken

Zur Darstellung einer Zeitmarke können Zeichenfolgen verwendet werden, die mit einer Ziffer beginnen und aus mindestens 16 Zeichen bestehen. Die vollständige Darstellung einer Zeitmarke hat das Format *jjjj-mm-tt-hh.mm.ss.nnnnnn*. Abschließende Leerzeichen sind zulässig. Führende Nullen können aus der Monats-, Tages- oder Stundenangabe der Zeitmarke weggelassen werden. Mikrosekunden können vollständig abgeschnitten oder weggelassen werden. Wenn Sie Ziffern in der Mikrosekundenangabe weglassen, wird eine implizite Angabe von 0 angenommen. Dementsprechend ist 1991-3-2-8.30.00 äquivalent zu 1991-03-02-08.30.00.000000.

Hinweise für MBCS

Zeichenfolgen für Datum und Zeitmarke dürfen nur aus Einzelbytezeichen und -ziffern bestehen.

Datums- und Uhrzeitformate: Die Zeichenfolgen für Datums- und Zeitformate entsprechen dem Standardformat der Werte für Datum und Uhrzeit, die dem Landescode der Anwendung zugeordnet sind. Dieses Standardformat kann durch Angeben der Formatoption "F" beim Vorkompilieren oder Binden an die Datenbank überschrieben werden.

Es folgt eine Beschreibung der Ein- und Ausgabeformate für Datum und Uhrzeit:

- Eingabezeitformat
 - Es gibt kein Standardformat für Uhrzeiteingaben.
 - Für jeden Landescode können alle Zeitformate eingegeben werden.
- Ausgabezeitformat
 - Der Standardwert für die Ausgabe des Zeitangaben entspricht dem lokalen Zeitformat.
- Eingabedatumsformat
 - Es gibt kein Standardformat für Datumseingaben.

- Wenn das lokale Format für das Datum und ein ISO-, JIS-, EUR- oder USA-Datumsformat nicht übereinstimmen, wird das lokale Format für die Datumseingabe angenommen. Siehe z. B. den Eintrag für Großbritannien in Tabelle 35.
- Ausgabedatumsformat
 - Das Standardformat für das Ausgabedatum wird in Tabelle 35 gezeigt.

Anmerkung: Tabelle 35 zeigt außerdem eine Liste der Zeichenfolgeformate für die verschiedenen Landescodes.

Tabelle 35. Datums- und Zeitformate mit Landescodes

Landescode	Lokales Datumsformat	Lokales Zeitformat	Standardausgabedatumsformat	Eingabedatumsformate
355 Albanien	jjjj-mm-tt	JIS	LOC	LOC, USA, EUR, ISO
785 Arabisch	tt/mm/jjjj	JIS	LOC	LOC, EUR, ISO
001 Australien (1)	mm-tt-jjjj	JIS	LOC	LOC, USA, EUR, ISO
061 Australien	tt-mm-jjjj	JIS	LOC	LOC, USA, EUR, ISO
032 Belgien	tt/mm/jjjj	JIS	LOC	LOC, EUR, ISO
055 Brasilien	tt.mm.jjjj	JIS	LOC	LOC, EUR, ISO
359 Bulgarien	tt.mm.jjjj	JIS	EUR	LOC, USA, EUR, ISO
001 Kanada	mm-tt-jjjj	JIS	USA	LOC, USA, EUR, ISO
002 Kanada (Französisch)	tt-mm-jjjj	ISO	ISO	LOC, USA, EUR, ISO
385 Kroatien	jjjj-mm-tt	JIS	ISO	LOC, USA, EUR, ISO
042 Tschechische Republik	jjjj-mm-tt	JIS	ISO	LOC, USA, EUR, ISO
045 Dänemark	tt-mm-jjjj	ISO	ISO	LOC, USA, EUR, ISO
358 Finnland	tt/mm/jjjj	ISO	EUR	LOC, EUR, ISO
389 FJR Republik Mazedonien	tt.mm.jjjj	JIS	EUR	LOC, USA, EUR, ISO
033 Frankreich	tt/mm/jjjj	JIS	EUR	LOC, EUR, ISO

Tabelle 35. Datums- und Zeitformate mit Landescodes (Forts.)

Landescode	Lokales Datumsformat	Lokales Zeitformat	Standardausgabedatumsformat	Eingabedatumsformate
049 Deutschland	tt/mm/jjj	ISO	ISO	LOC, EUR, ISO
030 Griechenland	tt/mm/jjj	JIS	LOC	LOC, EUR, ISO
036 Ungarn	jjj-mm-tt	JIS	ISO	LOC, USA, EUR, ISO
354 Island	tt-mm-jjj	JIS	LOC	LOC, USA, EUR, ISO
091 Indien	tt/mm/jjj	JIS	LOC	LOC, EUR, ISO
972 Israel	tt/mm/jjj	JIS	LOC	LOC, EUR, ISO
039 Italien	tt/mm/jjj	JIS	LOC	LOC, EUR, ISO
081 Japan	mm/tt/jjj	JIS	ISO	LOC, USA, EUR, ISO
082 Korea	mm/tt/jjj	JIS	ISO	LOC, USA, EUR, ISO
001 Lateinamerika (1)	mm-tt-jjj	JIS	LOC	LOC, USA, EUR, ISO
003 Lateinamerika	tt-mm-jjj	JIS	LOC	LOC, EUR, ISO
031 Niederlande	tt-mm-jjj	JIS	LOC	LOC, USA, EUR, ISO
047 Norwegen	tt/mm/jjj	ISO	EUR	LOC, EUR, ISO
048 Polen	jjj-mm-tt	JIS	ISO	LOC, USA, EUR, ISO
351 Portugal	tt/mm/jjj	JIS	LOC	LOC, EUR, ISO
086 VR China	mm/tt/jjj	JIS	ISO	LOC, USA, EUR, ISO
040 Rumänien	jjj-mm-tt	JIS	ISO	LOC, USA, EUR, ISO
007 Rußland	tt/mm/jjj	ISO	LOC	LOC, EUR, ISO
381 Serbien/ Montenegro	jjj-mm-tt	JIS	ISO	LOC, USA, EUR, ISO
042 Slowakei	jjj-mm-tt	JIS	ISO	LOC, USA, EUR, ISO

Tabelle 35. Datums- und Zeitformate mit Landescodes (Forts.)

Landescode	Lokales Datumsformat	Lokales Zeitformat	Standardausgabedatumsformat	Eingabedatumsformate
386 Slowenien	jjjj-mm-tt	JIS	ISO	LOC, USA, EUR, ISO
034 Spanien	tt/mm/jjjj	JIS	LOC	LOC, EUR, ISO
046 Schweden	tt/mm/jjjj	ISO	ISO	LOC, EUR, ISO
041 Schweiz	tt/mm/jjjj	ISO	EUR	LOC, EUR, ISO
088 Taiwan	mm-tt-jjjj	JIS	ISO	LOC, USA, EUR, ISO
066 Thailand (2)	tt/mm/jjjj	JIS	LOC	LOC, EUR, ISO
090 Türkei	tt/mm/jjjj	JIS	LOC	LOC, EUR, ISO
044 Großbritannien	tt/mm/jjjj	JIS	LOC	LOC, EUR, ISO
001 USA	mm-tt-jjjj	JIS	USA	LOC, USA, EUR, ISO
084 Vietnam	tt/mm/jjjj	JIS	LOC	LOC, EUR, ISO
<p>Anmerkungen:</p> <ol style="list-style-type: none"> 1. Ländern, die die standardmäßige länderspezifische Angabe C verwenden, wird der Landescode 001 zugeordnet. 2. jjjj in der buddhistischen Zeitrechnung entspricht der gregorianischen Zeitrechnung + 543 Jahre (nur Thailand). 				

Unicode-/UCS-2- und UTF-8-Unterstützung in DB2 UDB

Diese beiden Standards sind hier dokumentiert.

Einführung

Der Unicode-Zeichencodierungsstandard ist ein Zeichencodierungsschema mit fester Länge, das Zeichen fast aller lebenden Sprachen der Welt umfaßt. Unicode-Zeichen werden normalerweise mit "U+xxxx" dargestellt, wobei xxxx der hexadezimale Code des Zeichens ist.

Jedes Zeichen ist unabhängig von der Sprache 16 Bit (2 Byte) groß. Obwohl die sich ergebenden 65000 Codeelemente zur Codierung der meisten Zeichen der wichtigsten Sprachen der Welt ausreichen, bietet der Unicode-Standard außerdem einen Erweiterungsmechanismus, mit dem bis zu einer Million weiterer Zeichen codiert werden können. Diese Erweiterung reserviert einen

Bereich von Codewerten (U+D800 bis U+D8FF, als „Ersatzzeichen“ bezeichnet) zum Codieren einiger 32-Bit-Zeichen als zwei aufeinanderfolgende Codeelemente.

Der Standard 10646 (ISO/IEC 10646) der International Standards Organization (ISO) und der International Electrotechnical Commission (IEC) definiert den Universalzeichensatz Universal Multiple-Octet Coded Character Set (UCS), von dem es eine 2-Byte- (UCS-2) und eine 4-Byte-Version (UCS-4) gibt. Die 2-Byte-Version dieses ISO-Standards entspricht Unicode ohne Ersatzzeichen. ISO 10646 definiert außerdem eine Erweiterungstechnik zum Codieren einiger UCS-4-Codes in einer mit UCS-2 codierten Zeichenfolge. Diese Erweiterung wird als UTF-16 bezeichnet und entspricht Unicode mit Ersatzzeichen.

DB2 UDB unterstützt UCS-2, d. h. Unicode ohne Ersatzzeichen.

Die Verbindung eines UTF-8-Clients (Codepage 1208) zu einer Nicht-Unicode-Datenbank wird nicht unterstützt.

UTF-8

Bei der UCS-2- oder Unicode-Codierung sind ASCII- und Steuerzeichen ebenfalls zwei Byte lang, und das führende Byte ist Null. Zum Beispiel ist NULL gleich U+0000 und das große A gleich U+0041. Dies könnte für ASCII-basierte Anwendungen und ASCII-Dateisysteme ein größeres Problem sein, weil in einer UCS-2-Zeichenfolge fremde Nullen an beliebigen Stellen in der Zeichenfolge auftreten können. Mit einem Umsetzungsalgorithmus, der als UTF-8 bezeichnet wird, kann dieses Problem für Programme umgangen werden, die sich darauf verlassen, daß ASCII-Code invariant ist.

UTF-8 (UCS Transformation Format 8) ist eine algorithmische Umsetzung, mit der UCS-4-Zeichen mit fester Länge in Bytefolgen variabler Länge umgesetzt werden. In UTF-8 werden ASCII-Zeichen durch ihren gewöhnlichen Einzelbytecode dargestellt. Andere Zeichen in UCS-2 sind jedoch zwei oder drei Byte lang. UTF-8 setzt UCS-2-Zeichen also in einen Mehrbyte-Zeichensatz um, für den ASCII unveränderlich ist. Die Anzahl von Byte für jedes UCS-2-Zeichen im UTF-8-Format ist der folgenden Tabelle zu entnehmen:

UCS-2 (hex)	UTF-8 (binär)	Beschreibung
0000 - 007F	0xxxxxxx	ASCII
0080 - 07FF	110xxxxx 10xxxxxx	bis U+07FF
0800 - FFFF	1110xxxx 10xxxxxx 10xxxxxx	weitere UCS-2

HINWEIS: Der Bereich D800 bis DFFF ist von der Behandlung durch die dritte Zeile dieser Tabelle auszunehmen, die den UCS-4-Bereich 0000 0800 bis 0000 FFFF betrifft.

Im oben aufgeführten Angaben entspricht eine Reihe von x der UCS-Bitdarstellung des Zeichens. U0080 wird beispielsweise in 11000010 10000000 umgesetzt.

UCS-2-/UTF-8-Implementierung in DB2 UDB

Nummern von Codepages/IDs für codierten Zeichensatz

Bei IBM wurde die UCS-2-Codepage als Codepage 1200 registriert. Alle Codepages werden mit wachsenden Zeichensätzen definiert, d. h. beim Hinzufügen von neuen Zeichen zu einer Codepage ändert sich die Nummer der Codepage nicht. Codepage 1200 verweist immer auf die aktuelle Version von Unicode/UCS-2 und wird für UCS-2-Unterstützung in DB2 UDB verwendet.

Eine spezifische Version des UCS-Standards, wie durch Unicode 2.0 und ISO/IEC 10646-1 definiert, wurde auch bei IBM als ID für codierten Zeichensatz (CCSID) 13488 registriert. Diese ID für codierten Zeichensatz wird intern von DB2 UDB zum Speichern von Grafikzeichenfolgen in euc-Japan- und euc-Taiwan-Datenbanken verwendet. Die ID für codierten Zeichensatz 13488 und die Codepage 1200 verweisen beide auf UCS-2 und werden auf dieselbe Weise behandelt mit Ausnahme des Werts ihres „Doppelbyte“-Leerzeichens (DBCS):

CP/CCSID	Einzelbyte-Leerzeichen	Doppelbyte-Leerzeichen
-----	-----	-----
1200	N/V	U+0020
13488	N/V	U+3000

HINWEIS: In einer UCS-2-Datenbank hat U+3000 keine spezielle Bedeutung.

Bei Umsetzungstabellen werden die gleichen Tabellen (Implikation) für beide verwendet, da Codepage 1200 eine Implikation der ID für codierten Zeichensatz 13488 ist.

Bei IBM wurde UTF-8 als ID für codierten Zeichensatz 1208 mit wachsendem Zeichensatz registriert (manchmal auch als Codepage 1208 bezeichnet). Wenn neue Zeichen zum Standard hinzugefügt werden, ändert sich diese Nummer (1208) nicht. Diese Nummer 1208 wird als Nummer für die Mehrbyte-Codepage für die DB2-Unterstützung von UCS-2/UTF-8 verwendet.

DB2 UDB unterstützt UCS-2 als neue Mehrbyte-Codepage. Die MBCS-Codepage-Nummer ist 1208. Dies ist die Codepage-Nummer der Datenbank und die Codepage von Zeichenfolgedaten in der Datenbank. Die Doppelbyte-Codepage-Nummer für UCS-2 ist 1200. Dies ist die Codepage von Grafikzeichenfolgedaten in der Datenbank. Wenn eine Datenbank in UCS-2/UTF-8 erstellt wird, werden CHAR-, VARCHAR-, LONG VARCHAR- und CLOB-Daten in UTF-8 und GRAPHIC-, VARGRAPHIC-, LONG VARGRAPHIC- und DBCLOB-Daten in UCS-2 gespeichert. Dies wird hier einfach als UCS-2-Datenbank bezeichnet.

Erstellen einer UCS-2-Datenbank

Standardmäßig werden Datenbanken in der Codepage der Anwendung erstellt, die sie erstellt. Wenn Sie daher Ihre Datenbank von einem UTF-8-Client erstellen (z. B. länderspezifische Angaben UNIVERSAL von AIX) oder wenn die Registrierungsvariable DB2CODEPAGE auf dem Client auf 1208 gesetzt ist, wird Ihre Datenbank als UCS-2-Datenbank erstellt. Alternativ dazu können Sie explizit "UTF-8" als Namen für einen codierten Zeichensatz angeben und jeden gültigen aus zwei Buchstaben bestehenden Gebietscode verwenden, der von DB2 UDB unterstützt wird.

Setzen Sie z. B. den folgenden Befehl ab, um eine UCS-2-Datenbank mit dem Gebietscode für die Vereinigten Staaten zu erstellen:

```
DB2 CREATE DATABASE dbname USING CODESET UTF-8 TERRITORY US
```

Zum Erstellen einer UCS-2-Datenbank mit der API **sqlcrea** sollten Sie die Werte in *sqledbcountryinfo* entsprechend setzen. Setzen Sie z. B. SQLDBCODESET auf den Wert UTF-8 und SQLDBLOCALE auf einen gültigen Gebietscode (z. B. US).

Die Standardsortierfolge für eine UCS-2-Datenbank ist IDENTITY, die die UCS-2-Codepunkt-Reihenfolge bereitstellt. Daher werden standardmäßig alle UCS-2-/UTF-8-Zeichen gemäß ihrer UCS-2-Codepunktfolge geordnet und verglichen.

Alle länderspezifischen Parameter wie Datums- oder Uhrzeitformat, Dezimaltrennzeichen u. a. basieren auf dem aktuellen Gebiet des Clients.

Eine UCS-2-Datenbank erlaubt Verbindungen von jeder Einzelbyte- und Mehrbyte-Codepage, die von DB2 UDB unterstützt wird. Codepage-Zeichenumsetzungen zwischen der Codepage des Clients und UTF-8 werden automatisch vom Datenbankmanager durchgeführt. Daten in Grafikzeichenfolgetypen sind immer in UCS-2 und erfahren keine Codepage-Umsetzungen. Die Umgebung des Befehlszeilenprozessors (CLP) ist eine Ausnahme. Wenn Sie mit einer SELECT-Anweisung Grafikzeichenfolgedaten (UCS-2) über den Befehlszeilenprozessor auswählen, werden die zurückgegebenen Grafikzeichenfolgedaten (vom Befehlszeilenprozessor) von UCS-2 in die Codepage Ihrer Client-Umgebung umgesetzt.

Jeder Client ist durch die Menge von Zeichen, die Eingabemethode und die von seiner Umgebung unterstützten Schriftarten beschränkt. Die UCS-2-Datenbank selbst jedoch akzeptiert und speichert alle UCS-2-Zeichen. Jeder Client arbeitet also mit einer Untergruppe von UCS-2-Zeichen, der Datenbankmanager läßt jedoch die Gesamtheit von UCS-2-Zeichen zu.

Wenn Zeichen von einer lokalen Codepage in UTF-8 umgesetzt werden, wird möglicherweise die Byteanzahl erweitert. Es gibt keine Erweiterung für ASCII-

Zeichen, aber andere UCS-2-Zeichen werden um einen Faktor von zwei oder drei erweitert. Die Byteanzahl für jedes UCS-2-Zeichen im UTF-8-Format kann anhand der Tabelle in „UTF-8“ auf Seite 469 bestimmt werden.

Datentypen

Alle von DB2 UDB unterstützten Datentypen werden auch in einer UCS-2-Datenbank unterstützt. Insbesondere Grafikzeichenfolgedaten werden für eine UCS-2-Datenbank unterstützt und in UCS-2/Unicode gespeichert. Jeder Client, einschließlich Clients mit SBCS-Zeichensatz, kann mit Grafikzeichenfolgedatentypen in UCS-2/Unicode arbeiten, wenn er mit einer UCS-2-Datenbank verbunden ist.

Eine UCS-2-Datenbank ist wie jede beliebige MBCS-Datenbank, bei der die Zeichenfolgedaten in der Anzahl von Byte gemessen werden. Beim Arbeiten mit Zeichenfolgedaten in UTF-8 sollte nicht davon ausgegangen werden, daß jedes Zeichen ein Byte lang ist. Bei UTF-8-Mehrbytecodierung ist jedes ASCII-Zeichen ein Byte lang, andere Zeichen nehmen jedoch je zwei oder drei Byte ein. Dies sollte beim Definieren von CHAR-Feldern berücksichtigt werden. Je nach Verhältnis von ASCII-Zeichen zu anderen Zeichen kann ein CHAR-Feld der Größe von n Byte zwischen $n/3$ bis n Zeichen enthalten.

Die Verwendung von UTF-8-Zeichenfolgedecodierung statt dem UCS-2-Grafikzeichenfolgedatentyp wirkt sich ebenfalls auf die Gesamtspeicheranforderungen aus. Wenn die Mehrheit der Zeichen ASCII-Zeichen sind und einige andere Zeichen dazwischenstehen, kann das Speichern von UTF-8-Daten die günstigere Alternative sein, da der Speicherbedarf eher einem Byte pro Zeichen entspricht. Wenn dagegen die Mehrheit der Zeichen keine ASCII-Zeichen sind, die auf 3 Byte lange UTF-8-Folgen erweitert werden (z. B. Ideogramme), ist das UCS-2-Grafikzeichenfolgeformat möglicherweise die bessere Alternative, da jedes UCS-2-Zeichen genau zwei Byte statt drei Byte für jedes entsprechende Zeichen im UTF-8-Format benötigt.

In MBCS-Umgebungen operieren SQL-Skalarfunktionen, die mit Zeichenfolgen arbeiten, z. B. LENGTH, SUBSTR, POSSTR, MAX, MIN u. ä., mit der Anzahl von „Byte“ statt mit der Anzahl von „Zeichen“. Das Verhalten ist in einer UCS-2-Datenbank gleich. Sie sollten jedoch besonders vorsichtig sein, wenn Sie relative Positionen und Längen für eine UCS-2-Datenbank angeben, da diese Werte immer im Kontext der Datenbank-Codepage definiert werden. Im Fall einer UCS-2-Datenbank sollten diese relativen Positionen daher in UTF-8 definiert werden. Da einige Einzelbytezeichen in UTF-8 mehr als ein Byte benötigen, sind SUBSTR-Indizes, die für eine Einzelbyte-Datenbank gültig sind, möglicherweise für eine UCS-2-Datenbank nicht gültig. Wenn Sie falsche Indizes angeben, wird SQLCODE-Wert -191 (SQLSTATE 22504) zurückgegeben. Eine Beschreibung der Arbeitsweise dieser Funktionen finden Sie im Handbuch *SQL Reference*.

SQL-Datentypen CHAR werden in Benutzerprogrammen vom Datentyp char (in der Programmiersprache C) unterstützt. SQL-Datentypen GRAPHIC werden von sqldbcchar in Benutzerprogrammen unterstützt. Bei einer UCS-2-Datenbank sind sqldbcchar-Daten immer im Big-endian-Format (hohes Byte zuerst). Wenn ein Anwendungsprogramm mit einer UCS-2-Datenbank verbunden ist, werden Zeichenfolgedaten zwischen der Codepage der Anwendung und UTF-8 durch DB2 UDB umgesetzt, Grafikzeichenfolgedaten bleiben jedoch immer in UCS-2.

Bezeichner

In einer UCS-2-Datenbank sind alle Bezeichner im UTF-8-Mehrbyteformat. Es ist daher möglich, jedes UCS-2-Zeichen in Bezeichnern zu verwenden, bei denen die Verwendung eines Zeichens im erweiterten Zeichensatz (z. B. ein Zeichen mit Akzent oder ein Mehrbytezeichen) von DB2 UDB zugelassen wird. Informationen darüber, welche Bezeichner die Verwendung erweiterter Zeichen zulassen, finden Sie in „Anhang B. Namenskonventionen“ auf Seite 399.

Clients können alle Zeichen eingeben, die von ihrer SBCS- oder MBCS-Zeichensatzumgebung unterstützt werden. Alle Zeichen in den Bezeichnern werden vom Datenbankmanager in UTF-8 umgesetzt. Zwei Punkte müssen beim Angeben von Zeichen der Landessprache in Bezeichnern für eine UCS-2-Datenbank beachtet werden:

- Jedes Nicht-ASCII-Zeichen erfordert zwei oder drei Byte. Ein Bezeichner mit n Byte kann daher je nach Verhältnis von ASCII- zu anderen Zeichen nur zwischen $n/3$ und n Zeichen enthalten. Wenn Sie nur ein oder zwei Nicht-ASCII-Zeichen (z. B. Zeichen mit Akzent) haben, ist die Begrenzung näher an n Zeichen, während bei einer Kennung, die nur aus Nicht-ASCII-Zeichen besteht (z. B. in Japanisch), nur $n/3$ Zeichen verwendet werden können.
- Wenn Kennungen von unterschiedlichen Client-Umgebungen eingegeben werden sollen, sollten sie mit der gemeinsamen Untermenge von Zeichen definiert werden, die auf diesen Clients verfügbar sind. Wenn z. B. von Lateinisch-1-, arabischen oder japanischen Umgebungen auf eine UCS-2-Datenbank zugegriffen werden soll, sollten alle Bezeichner auf ASCII beschränkt sein.

UCS-2-Literale

UCS-2-Literale können auf zwei Arten angegeben werden:

- Als Grafikzeichenfolgekennkonstante mit dem Format G'...' oder N'...' gemäß der Beschreibung im Handbuch *SQL Reference*, Kapitel "Language Elements", Abschnitt "Graphic String Constants". Jedes auf diese Art angegebene Literal wird vom Datenbankmanager von der Codepage der Anwendung in UCS-2 umgesetzt.

- Als hexadezimale UCS-2-Zeichenfolge mit dem Format UX'....' oder GX'....': Die in Anführungszeichen nach UX oder GX angegebene Konstante muß ein Mehrfaches von 4 hexadezimalen Ziffern sein. Jede Gruppe aus vier Ziffern stellt jeweils einen UCS-2-Codepunkt dar.

Bei Verwendung des Befehlszeilenprozessors (CLP) ist die erste Methode einfacher, wenn das UCS-2-Zeichen in der Codepage der lokalen Anwendung vorhanden ist (z. B. zur Eingabe von Zeichen der Codepage 850 von einem Terminal, das Codepage 850 verwendet). Die zweite Methode sollte für Zeichen verwendet werden, die außerhalb des Umfangs der Codepage der Anwendung liegen (z. B. zum Angeben von japanischen Zeichen von einem Terminal, das Codepage 850 verwendet).

Mustererkennung in einer UCS-2-Datenbank

Mustererkennung ist ein Bereich, in dem sich vorhandene MBCS-Datenbanken leicht von einer UCS-2-Datenbank unterscheiden.

MBCS-Datenbanken in DB2 UDB verhalten sich wie folgt: Wenn der Übereinstimmungsausdruck MBCS-Daten enthält, kann das Muster sowohl SBCS- als auch MBCS-Zeichen enthalten. Die Sonderzeichen im Muster werden folgendermaßen interpretiert:

- Ein SBCS-Unterstrich verweist auf ein SBCS-Zeichen.
- Ein DBCS-Unterstrich verweist auf ein MBCS-Zeichen.
- Ein (SBCS- oder DBCS-) Prozentzeichen verweist auf eine Zeichenfolge von null oder mehr SBCS- oder MBCS-Zeichen.

Wenn der Übereinstimmungsausdruck DBCS-Grafikzeichenfolgedaten enthält, enthalten die Ausdrücke nur DBCS-Zeichen. Die Sonderzeichen im Muster werden folgendermaßen interpretiert:

- Ein DBCS-Unterstrich verweist auf ein DBCS-Zeichen.
- Ein DBCS-Prozentzeichen verweist auf eine Zeichenfolge von null oder mehr DBCS-Zeichen.

In einer UCS-2-Datenbank gibt es keine echte Unterscheidung zwischen „Einzelbyte“- und „Doppelbyte“-Zeichen. Jedes UCS-2-Zeichen nimmt zwei Byte ein. Obwohl das UTF-8-Format eine „Mischbyte“-Codierung von UCS-2-Zeichen ist, gibt es in UTF-8 keine echte Unterscheidung zwischen SBCS- und MBCS-Zeichen. Jedes Zeichen ist ein UCS-2-Zeichen, unabhängig von der Anzahl seiner Byte im UTF-8-Format. Bei der Angabe eines Zeichenfolge- oder Grafikzeichenfolgeausdrucks verweist ein Unterstrich auf ein UCS-2-Zeichen und ein Prozentzeichen auf eine Zeichenfolge von null oder mehr UCS-2-Zeichen.

Auf der Client-Seite verwenden die Zeichenfolgeausdrücke die Codepage des Clients und werden vom Datenbankmanager in UTF-8 umgesetzt. SBCS-

Codepages von Clients haben keine DBCS-Prozentzeichen oder -Unterstriche. Jede unterstützte Codepage enthält jedoch ein Einzelbyte-Prozentzeichen (entspricht U+0025) und einen Einzelbyte-Unterstrich (entspricht U+005F). Sonderzeichen werden in einer UCS-2-Datenbank folgendermaßen interpretiert:

- Ein SBCS-Unterstrich (entspricht U+0025) verweist auf ein UCS-2-Zeichen in einem Grafikzeichenfolgeausdruck oder einem UTF-8-Zeichen in einem Zeichenfolgeausdruck.
- Ein SBCS-Prozentzeichen (entspricht U+005F) verweist auf eine Zeichenfolge von null oder mehr UCS-2-Zeichen in einem Grafikzeichenfolgeausdruck oder eine Zeichenfolge von null oder mehr UTF-8-Zeichen in einem Zeichenfolgeausdruck.

DBCS-Codepages unterstützen außerdem ein DBCS-Prozentzeichen (entspricht U+FF05) und einen DBCS-Unterstrich (entspricht U+FF3F). Diese Zeichen haben für eine UCS-2-Datenbank keine besondere Bedeutung.

Für den wahlfreien „Escape-Ausdruck“, der ein Zeichen angibt, das verwendet wird, um die spezielle Bedeutung des Unterstrichs und des Prozentzeichens aufzuheben, werden nur ASCII-Zeichen oder Zeichen unterstützt, die in eine UTF-8-Doppelbytefolge erweitert werden. Wenn Sie ein Escape-Zeichen angeben, das in einen 3 Byte langen UTF-8-Wert erweitert wird, wird eine Fehlermeldung (SQL0130N, SQLSTATE 22019) zurückgegeben.

Überlegungen zu IMPORT/EXPORT/LOAD

Die Dateiformate DEL, ASC und PC/IXF werden für eine UCS-2-Datenbank gemäß der Beschreibung in diesem Abschnitt unterstützt. Das Format WSF wird nicht unterstützt.

Beim Export von einer UCS-2-Datenbank in eine ASCII-DEL-Datei werden alle Zeichendaten in die Codepage der Anwendung umgesetzt. Sowohl Zeichenfolge- als auch Grafikzeichenfolgedaten werden in dieselbe SBCS- oder MBCS-Codepage des Clients umgesetzt. Dies ist die für den Export erwartete Funktionsweise von Datenbanken und kann nicht geändert werden, weil die gesamte ASCII-DEL-Datei nur eine Codepage haben kann. Wenn Sie also in eine ASCII-DEL-Datei exportieren, werden nur die UCS-2-Zeichen gespeichert, die in der Codepage Ihrer Anwendung vorhanden sind. Andere Zeichen werden durch die Standardsubstitutionszeichen für die Codepage der Anwendung ersetzt. Bei UTF-8-Clients (Codepage 1208) entsteht kein Datenverlust, weil alle UCS-2-Zeichen von UTF-8-Clients unterstützt werden.

Beim Importieren aus einer ASCII-Datei (DEL oder ASC) in eine UCS-2-Datenbank werden Zeichenfolgedaten aus der Codepage der Anwendung in UTF-8 und Grafikzeichenfolgedaten aus der Codepage der Anwendung in UCS-2 umgesetzt. Es entsteht kein Datenverlust. Wenn Sie ASCII-Daten importieren wollen, die unter einer anderen Codepage gespeichert wurden,

sollten Sie die Codepage der Datendatei ändern, bevor Sie den Befehl `IMPORT` absetzen. Eine Möglichkeit hierzu besteht darin, `DB2CODEPAGE` auf die Codepage der ASCII-Datendatei zu setzen.

Der Bereich gültiger ASCII-Begrenzer für SBCS- und MBCS-Clients ist identisch mit dem derzeit von DB2 UDB für diese Clients unterstützten Bereich. Der Bereich gültiger Begrenzer für UTF-8-Clients ist 0x01 bis 0x7F mit den üblichen Einschränkungen. Eine vollständige Liste dieser Einschränkungen finden Sie im Anhang zu Dateiformaten für die Dienstprogramme `IMPORT/EXPORT/LOAD` im Handbuch *Versetzen von Daten Dienstprogramme und Referenz*.

Beim Export aus einer UCS-2-Datenbank in eine PC/IXF-Datei werden alle Zeichenfolgedaten in die SBCS-/MBCS-Codepage des Clients umgesetzt. Grafikzeichenfolgedaten werden nicht umgesetzt und in UCS-2 (Codepage 1200) gespeichert. Es entsteht kein Datenverlust.

Beim Importieren aus einer PC/IXF-Datei in eine UCS-2-Datenbank wird bei Zeichenfolgedaten davon ausgegangen, daß sie in der in den PC/IXF-Kopfdaten gespeicherten SBCS-/MBCS-Codepage vorliegen, und bei Grafikzeichenfolgedaten, daß sie in der in den PC/IXF-Kopfdaten gespeicherten DBCS-Codepage vorliegen. Zeichenfolgedaten werden vom Importdienstprogramm von der in den PC/IXF-Kopfdaten angegebenen Codepage in die Codepage des Clients und dann (von der Anweisung `INSERT`) von der Codepage des Clients in UTF-8 umgesetzt. Grafikzeichenfolgedaten werden vom Importdienstprogramm von der in den PC/IXF-Kopfdaten angegebenen DBCS-Codepage direkt in UCS-2 (Codepage 1200) umgesetzt.

Das Dienstprogramm `LOAD` setzt die Daten direkt in die Datenbank und geht standardmäßig davon aus, daß Daten in ASC- oder DEL-Dateien in der Codepage der Datenbank vorliegen. Es findet daher standardmäßig keine Codepage-Umsetzung für ASCII-Dateien statt. Wenn die Codepage für die Datendatei explizit (mit dem `codepage`-Wert) angegeben wurde, verwendet das Dienstprogramm `LOAD` diese Information zum Umsetzen der Daten aus der angegebenen Codepage in die Codepage der Datenbank, bevor es die Daten lädt. Bei PC/IXF-Dateien konvertiert das Dienstprogramm `LOAD` immer von den in den IXF-Kopfdaten angegebenen Codepages in die Codepage der Datenbank (1208 für `CHAR` und 1200 für `GRAPHIC`).

Die Codepage für `DBCLOB`-Dateien ist immer 1200 für UCS-2. Die Codepage von `CLOB`-Dateien entspricht der Codepage der importierten, geladenen oder exportierten Datendateien. Beim Laden oder Importieren von Daten mit dem Format PC/IXF wird beispielsweise angenommen, daß die `CLOB`-Datei in der in den PC/IXF-Kopfdaten angegebenen Codepage vorliegt. Wenn die `DBCLOB`-Datei das Format `ASC` oder `DEL` hat, nimmt das Dienstprogramm `LOAD` an, daß `CLOB`-Daten in der Codepage der Datenbank vorliegen (sofern

dies nicht explizit mit dem codepage-Wert anders angegeben wird), während das Importdienstprogramm annimmt, daß die Daten in der Codepage der Client-Anwendung vorliegen.

Der nochecklengths-Wert wird für eine UCS-2-Datenbank aus folgenden Gründen immer angegeben:

- Jeder SBCS-Zeichensatz kann mit einer Datenbank verbunden sein, für die keine DBCS-Codepage vorhanden ist.
- Zeichenfolgen im UTF-8-Format haben in der Regel andere Längen als die Zeichenfolgen in Client-Codepages.

Weitere Informationen über die Dienstprogramme LOAD, IMPORT und EXPORT finden Sie im Handbuch *Versetzen von Daten Dienstprogramme und Referenz*.

Inkompatibilitäten

Für Anwendungen, die mit einer UCS-2-Datenbank verbunden sind, sind die Grafikzeichenfolgedaten immer in UCS-2 (Codepage 1200). Für Anwendungen, die mit Nicht-UCS-2-Datenbanken verbunden sind, haben die Grafikzeichenfolgedaten die DBCS-Codepage der Anwendung oder sind unzulässig, wenn die Codepage der Anwendung SBCS ist. Wenn z. B. ein 932-Client mit einer japanischen Nicht-UCS-2-Datenbank verbunden ist, haben die Grafikzeichenfolgedaten die Codepage 301. Für die 932-Client-Anwendungen, die mit einer UCS-2-Datenbank verbunden sind, sind die Grafikzeichenfolgedaten in UCS-2.

Anhang F. 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. Anstelle der IBM Produkte, Programme oder Dienstleistungen können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb der Produkte, Programme oder Dienstleistungen in Verbindung mit Fremdprodukten und Fremddienstleistungen liegt beim Kunden, soweit nicht ausdrücklich solche Verbindungen erwähnt sind.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanfragen sind schriftlich an IBM Europe, Director of Licensing, 92066 Paris La Defense Cedex, France, zu richten. Anfragen an obige Adresse müssen auf englisch formuliert werden.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen bekanntgegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Web-Sites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Web-Sites dar. Das über diese Web-Sites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Web-Sites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne daß eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Handbuch aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt im Rahmen der Allgemeinen Geschäftsbedingungen der IBM, der Internationalen Nutzungsbedingungen der IBM für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer gesteuerten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Garantie, daß diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Informationen über Produkte anderer Hersteller als IBM wurden von den Herstellern dieser Produkte zur Verfügung gestellt, bzw. aus von ihnen veröffentlichten Ankündigungen oder anderen öffentlich zugänglichen Quellen entnommen. IBM hat diese Produkte nicht getestet und übernimmt im Hinblick auf Produkte anderer Hersteller keine Verantwortung für einwandfreie Funktion, Kompatibilität oder andere Ansprüche. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten der IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele der IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogrammes illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden, Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHT-LIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Beispielprogramme geschrieben werden. Die in diesem Handbuch aufgeführten Beispiele sollen lediglich der Veranschaulichung und zu keinem anderen Zweck dienen. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet.

Kopien oder Teile der Beispielprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. _Jahr/Jahre angeben_. Alle Rechte vorbehalten.

Neue deutsche Rechtschreibung

Durch die Einführung der neuen deutschen Rechtschreibung bei IBM zum 1. September 1999 kann es vorkommen, dass in dem vorliegenden Handbuch bestimmte Wörter sowohl nach der alten als auch nach der neuen Schreibweise verwendet werden, und zwar immer dann, wenn auf existierende Handbuchkapitel und/oder Programmteile zurückgegriffen wird.

Änderungen in der IBM Terminologie

Die ständige Weiterentwicklung der deutschen Sprache nimmt auch Einfluss auf die IBM Terminologie. Durch die daraus resultierende Umstellung der IBM Terminologie kann es u. U. vorkommen, dass in diesem Handbuch sowohl alte als auch neue Termini gleichbedeutend verwendet werden. Dies ist der Fall, wenn auf ältere existierende Handbuchkapitel und/oder Programmteile zurückgegriffen wird.

Aufgrund kurzfristiger Änderungen der Software, die in die Dokumentation nicht mehr aufgenommen werden konnten, entsprechen die in den Handbüchern aufgeführten Programmelemente möglicherweise nicht den im eigentlichen Programm angezeigten Elementen.

Marken

Folgende Namen sind in gewissen Ländern Marken der International Business Machines Corporation.

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
IBM System AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RS/6000
DataPropagator	IBM System /370
DataRefresher	SP
DB2	SQL/DS
DB2 Connect	SQL/400
DB2 Extenders	System/370
DB2 OLAP Server	IBM System /390
DB2 Universal Database	SystemView
Distributed Relational Database Architecture	VisualAge
DRDA	VM/ESA
eNetwork	VSE/ESA
Extended Services	VTAM
FFST	WebExplorer
First Failure Support Technology	WIN-OS/2

Folgende Namen sind in gewissen Ländern Marken oder eingetragene Marken anderer Unternehmen:

Microsoft, Windows und Windows NT sind Marken oder eingetragene Marken von Microsoft Corporation.

Java und alle auf Java basierenden Marken und Logos sowie Solaris sind in gewissen Ländern Marken von Sun Microsystems, Inc.

Tivoli und NetView sind in gewissen Ländern Marken von Tivoli Systems Inc.

UNIX ist eine eingetragene Marke und wird ausschließlich von der X/Open Company Limited lizenziert.

Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken anderer Unternehmen sein.

Index

A

Abfrageinterne Parallelität 74
Abfrageparallelität 74
Abfrageübergreifende Parallelität 74
Abhängige Tabelle 122
Abhängige Zeile 122
Abschätzen des Speicherbedarfs
 Großes Objekt (LOB-Daten) 138
 Indexbereich 139
 Langfelddaten 138
 Protokolldatei 142
 Tabellen 134
 temporärer Arbeitsbereich 144
Agent
 hohe Verfügbarkeit 331
 Warehouse 88
Agent-Site 89
Aktive Protokolle 38
Aktualisierende Wiederherstellung 32
 Datenbank 32
 Tabellenbereich 34
Aktualisierung auf mehreren Systemen 181, 183
 Host- oder AS/400-
 Anwendungen, Zugriff auf DB2
 UDB-Server 188
Aktualisierung auf mehreren Systemen konfigurieren, Assistent 395
Aktualisierungsregel 125
Aliasadresse 256
Aliasname
 Namenskonventionen 403
Aliasname der Datenbank 399
 Namenskonventionen 400
Ändern, Kennwörter 401
Anwendungsentwurf
 Sortierfolgen, Richtlinien 460
Anzeigen
 Online-Informationen 393
Arbeitseinheit 179
 fern 180
Archivierte Protokolldateien
 offline 38
 online 38
Archivprotokolle 37
Assistent
 Datenbank wiederherstellen 396

Assistenten
 Aktualisierung auf mehreren
 Systemen konfigurieren 395
 Assistenten 395
 Datenbank erstellen 395
 Datenbank hinzufügen 395, 396
 Datenbank sichern 395
 Index 396
 Leistungskonfiguration 396
 Tabelle erstellen 396
 Tabellenbereich erstellen 396
 Tasks ausführen 395
Attribut 102
Attributive Daten 95
Auf sich selbst verweisende Tabelle
 Integritätsbedingung 122
 Tabelle 122
 Zeile 122
Auslöser 27, 127
 Namenskonventionen 403
Auswählen von EXTENTSIZE 169
Authentifizierung 59
 zusammengeschlossene Daten-
 bank, Übersicht 62
Automatische Funktions-
übernahme 233, 245, 293, 327
 Bereitschaftsmodus (Hot Stand-
 by) 233, 234
 Exemplarübernahme 235
 Funktionsübernahme für mehrere
 logische Knoten 237
 gegenseitige Exemplar-
 übernahme 238
 gegenseitige Partitionsübernah-
 me 239
 Gegenseitige Übernahme (Mutual
 Takeover), Modus 233, 237
 Paralleler Zugriff (Concurrent
 Access) 233
 Partitionsübernahme 236
 Wiederherstellen der Verbindung
 nach Funktionsübernahme 240

B

Basisadresse 256
Begrenzen der Auswirkungen von
 Datenträgerfehlern 54
Begrenzen der Auswirkungen von
 Transaktionsfehlern 57

Behälter
 Übersicht 20
Beispielprogramme
 HTML 386
 plattformübergreifend 386
Benutzer-Exit-Programm
 Protokolle 47
 Sicherung 48
Benutzer-IDs
 benennen 400
Benutzerdefinierte Ereignisse 245,
 267
Benutzerdefinierte Funktionen
 (UDFs) 107
 Namenskonventionen 403
Benutzerdefinierter Datentyp (UDT)
 Namenskonventionen 403
 Spaltendefinition 106
Benutzerdefinierter Programm-
schritt 91
Benutzerprozeduren 350
Benutzertabelle
 Seitenbegrenzungen 135
Benutzertabellenbereich 155
Benutzertabellenbereich, temporär 156
Berechtigung 61
 Übersicht 127
 zusammengeschlossene Daten-
 bank, Übersicht 62
Berechtigungsstufe 61
Bereich 107
Beschädigter Tabellenbereich 49
Betriebsdaten 87
Beziehungen zwischen Tabellen 48
C
Campus-Clusterung 338
cconsole, Dienstprogramm 338
CCSID-Unterstützung für bidirektio-
nale Zeichen 454
 CCSID-Tabelle 455
 DB2 Connect-
 Implementierung 457
 DB2 UDB-Implementierung 456
Cluster
 Konfiguration 246
 Überwachung 277
 Verwaltung 246

- Clustering
 - Campus 338
 - kontinental 338
- Codepage
 - DB2CODEPAGE, Registrierdatenbankvariable 452
 - unterstützte Windows-Codepages 452
- ctelnet, Dienstprogramm 338
- D**
- Data Warehouse-Zentrale 87
- Datei
 - Datenbank 132
- Daten
 - Großes Objekt (LOB-Daten) 138
 - informative 88
 - Langfeld 138
 - operative 87
 - Partitionierung 146
- Daten- und Paritäts-Striping nach Sektoren (RAID-5) 55
- Datenbank
 - auf einem Host-System 182
 - Dateien 132
 - Namenskonventionen 400
 - nicht wiederherstellbar 36
 - Regeln für Objektnamen 399
 - Übersicht 11
 - verteilt 179
 - Verwenden mehrerer Datenbanken in einer Transaktion 181
 - Verzeichnisse 131
 - wiederherstellbar 36
- Datenbank, aktualisierende
 - Wiederherstellung 32
- Datenbank, Konfigurationsparameter
 - Übersicht 23
- Datenbank erstellen, Assistent 395
- Datenbank hinzufügen, Assistent 395, 396
- Datenbank sichern, Assistent 395
- Datenbankentwurf
 - logischer 101
 - physisch 131
- Datenbankmanager
 - Namenskonventionen 399
- Datenbankmanager, Konfigurationsparameter
 - Übersicht 22
- Datenbankmigration 407
- Datenbankobjekte
 - Datenbank 11
 - Exemplar 10
 - Index 12
- Datenbankobjekte (*Forts.*)
 - Knotengruppe 11
 - Namenskonventionen 403, 453
 - Schema 13
 - Sicht 12
 - Systemkatalogtabelle 14
 - Tabelle 11
 - Übersicht 9
- Datenbankpartition 71
 - Synchronisation 57
- Datenbankprotokolle 37
- Datenbanksystem
 - zusammengeschlossen 65
- Datenquelle 65
- Datenträgerfehler
 - Begrenzen der Auswirkungen 54
 - Protokolle 47
 - Überlegungen zu Katalogknoten 54
- Datentypisierung
 - Übersicht 128
- Datum
 - Definition 462
 - Formate 465
- DB2-Agent für hohe Verfügbarkeit 346
 - hadb2tab, Konfigurationsdatei 347
 - Registrierung 346
 - Steuermethoden 348
- DB2-Bibliothek
 - Assistenten 395
 - Dokument-Server einrichten 397
 - Drucken von PDF-Handbüchern 388
 - gedruckte Handbücher bestellen 389
 - Handbücher 375
 - Information - Unterstützung 394
 - neueste Informationen 388
 - Online-Hilfefunktion 391
 - Online-Informationen anzeigen 393
 - Online-Informationen suchen 398
 - Sprachenkennung für Bücher 387
 - Struktur 375
- DB2 Connect
 - für Datenbankaktualisierungen auf mehreren Systemen 182
- DB2-Modell ohne gemeinsame Benutzung 233
- DB2-Synchronisationspunktmanager (SPM) 188
- DB2CODEPAGE, Registrierdatenbankvariable 452
- DB2MCS, Dienstprogramm
 - Einrichten eines Datenbanksystems mit einer Partition 302
 - Einrichten eines partitionierten Datenbanksystems 303
 - Einrichten zweier Datenbanken mit einer Partition für gegenseitige Übernahme 302
 - Parameter in
 - DB2MCS.CFG 297
 - Übersicht 296
- Definieren von Tabellenspalten 106
- Dienstprogrammparallelität 77
- DMS (von der Datenbank verwalteter Bereich) 16, 161
 - Hinzufügen von Behältern 163
- DMS-Tabellenbereich 16, 161
- Dokument-Server einrichten 397
- Dritte Normalform 116
- Drucken von PDF-Handbüchern 388
- DTP (Distributed Transaction Processing) 198
- E**
- Ein-/Ausgabe (E/A), Überlegungen
 - Tabellenbereich 164
- Ein-/Ausgabeparallelität 74
- Eindeutig
 - Schlüssel 108
- Eindeutiger Index
 - Index 13
 - Integritätsbedingungen 119
 - Schlüssel 121
- Eindeutiges Identifizieren der Entitäten 112
- Eindeutigkeitsintegritätsbedingung 25
- Einschränkungen von Betriebssystemen 48
- Einzelpartition
 - Einzelprozessorumgebung 78
 - Mehrprozessorumgebung 79
- Einzelprozessorumgebung 78
- Empfehlungen für Katalogtabellenbereiche 172
- Enhanced Scalability (ES) 245
- Entclustering
 - teilweise 147
- Entität 101

Entwerfen und Auswählen von Tabellenbereichen 153
 Entwurf
 Tabellenbereich 164
 Zusammengeschlossene Datenbank 178
 Eprimary-Knoten des SP-Switch 257
 Ereignisüberwachung 267
 Erste Normalform 113
 Erste passende Stelle, Reihenfolge 135
 Erstellungsaufgaben
 Warehouses 91
 ES (Enhanced Scalability) 245
 Exemplar
 Übersicht 10
 EXTENTSIZE, Parameter 20, 154
 auswählen 169

F
 Fehlerbehebung 53
 Fehlermonitor 352
 Fehlertoleranz 330
 Ferne Arbeitseinheit 180
 Festschreiben
 Fehler beim zweiphasigen zweiphasig 193
 zweiphasig 190
 Flush Log 40
 Fortlaufende Verfügbarkeit 330
 Fremdschlüssel 121
 Fremdschlüssel, Integritätsbedingung 26
 Funktionsübernahme 245
 Übersicht 327
 Funktionsübernahme, Dauer 363

G
 Geografische Codierung 97
 Geografisches Informationssystem (GIS) 93
 Geschäftsmetadaten 91
 Geschäftsregeln
 Übersicht 24
 GIS (geografisches Informationssystem) 93
 Groß-/Kleinschreibung bei Namen, zusammengesetzte Datenbank 405
 Großes Objekt (LOB-Daten)
 Abschätzen des Speicherbedarfs 138
 Spaltendefinition 106

H
 HA.config, Datei 355
 HA-NFS 338
 HACMP (High Availability Cluster Multi-Processing) 233, 245
 HACMP
 ES-Konfigurationsbeispiele 257
 Handbücher 375, 389
 Hardwareplatteneinheiten 55
 Hardwareumgebungen 78
 Arten der Parallelität 85
 Einzelpartition, Einzelprozessor 78
 Einzelpartition, mehrere Prozessoren 79
 logische Datenbankpartitionen 83
 Partitionen mit einem Prozessor 81
 Partitionen mit mehreren Prozessoren 82
 Heartbeat 245, 328
 Heuristische Maßnahmen 212
 High Availability Cluster Multi-Processing (HACMP) 233, 245
 Hintereinanderschaltende Zuordnung 247
 Hinzufügen von Behältern zu DMS-Tabellenbereichen 163
 Hohe Verfügbarkeit 233, 293, 327
 Hohe Verfügbarkeit unter Sun Cluster 2.2
 Anwendungen, die Verbindung zu HA-Exemplar herstellen 339
 Ausgangsverzeichnislayout für EE- und EEE-Exemplare 343
 Datenbank und Datenbankmanager, Konfigurationsparameter 345
 Datenreplikation 346
 DB2-Agent für hohe Verfügbarkeit 346
 DB2-Installation, Speicherposition und Optionen 345
 Fehlerbehebung 365
 hadb2_setup, Befehl 358
 Installation und Konfiguration 357
 logische Hosts und DB2 UDB EEE 344
 Plattenlayout für EE- und EEE-Exemplare 341
 Wiederherstellung nach einem Systemabsturz 346

HTML
 Beispielprogramme 386

I
 IBMCATGROUP 155
 IBMDEFAULTGROUP 155
 IBMTEMPGROUP 156
 Identifizieren möglicher Schlüsselspalten 110
 IDENTITY-Spalte 111
 Index
 Namenskonventionen 403
 Übersicht 12
 Index, Assistent 396
 Indexbereich
 Abschätzen des Speicherbedarfs 139
 Indexschlüssel 13
 Information - Unterstützung 394
 Informationskatalog 91
 Informative Daten 88
 Inkompatibilitäten
 Beschreibung 413
 COLNAMES (geplant) 415
 Datenbanken erstellen 433
 FK_COLNAMES (geplant) 414
 Fremdschlüsselspaltennamen 421
 geplant 414
 PK_COLNAMES (geplant) 414
 Primärschlüsselspaltennamen 421
 schreibgeschützte Sichten (geplant) 414
 Spaltenabweichung 424
 Spaltentyp in BIGINT geändert 423
 SYSCAT.CHECKS-Spalte TEXT 423
 SYSCAT.INDEXES-Spalte COLNAMES 422
 SYSCAT.STATEMENTS-Spalte TEXT 422
 SYSCAT.VIEWS-Spalte TEXT 421
 Version 6 420
 Version 7 415
 Inkompatibilitäten für Version 6
 Abhängigkeitscodes 425
 aktueller EXPLAIN-Modus 435
 DATALINK-Spalten 432
 Ereignismonitor, Ausgabedatenstromformat 431
 FOR UPDATE-Syntax 428
 Java-Programmierung 427

- Inkompatibilitäten für Version 6 (Forts.)
 - OBJCAT-Sichten 425
 - PC/IXF-Formatänderungen 430
 - RUMBA 436
 - SELECT-Zugriffsrecht in Hierarchie 434
 - SQLNAME in nicht verdoppelten SQLVAR-Einträgen 430
 - SYSFUN-Zeichenfolgefunktionskennungen 432
 - SYSIBM-Basiskataloge 426
 - SYSTABLE-Spaltenänderung 433
 - USING und SORT BUFFER 435
 - VARCHAR, Datentyp 427
 - veraltete Konfigurationsparameter der Datenbank 436
 - veraltete Konfigurationsschlüsselwörter 431
 - Zeichennamenlängen 428
- INSERT, Regeln 123
- Installation
 - Netscape-Browser 393
- Integritätsbedingung 118
 - Eindeutigkeit 25
 - Fremdschlüssel 26
 - NOT NULL 25
 - Primärschlüssel 25
 - Prüfung 27
- K**
 - Kapazität 78
 - Katalogtabellenbereich 155
 - Keepalive-Pakete 245
 - Kennwörter
 - ändern 401
 - benennen 400
 - Knoten
 - Datenspeicherposition bestimmen 147
 - Knotengruppen 72
 - entwerfen 144
 - IBMCATGROUP 155
 - IBMDEFAULTGROUP 155
 - IBMTEMPGROUP 156
 - Übersicht 11
 - Knotensynchronisation 57
 - Kollokation, Tabelle 151
 - Kompatibilität
 - Partition 151
 - Konfiguration
 - mit mehreren Partitionen 81
 - Konfiguration für Bereitschaftsmodus 246
 - Beispiel 253
 - Konfiguration für gegenseitige Übernahme 246
 - Beispiel 253
 - Konfigurationsparameter
 - Überlegungen zum DB2-Transaktionsmanager 187
 - Übersicht 22
 - Konsistenzzustand 30
 - Kontinentale Clusterung 338
 - Konfigurationssystem 96
 - Koordinator-knoten 71
- L**
 - Langfelddaten
 - Abschätzen des Speicherbedarfs 138
 - Leistung
 - Wiederherstellung 51
 - Leistungskonfiguration, Assistent 396
 - LIST INDOUBT TRANSACTIONS, Befehl 210
 - LOB-Daten (großes Objekt)
 - Abschätzen des Speicherbedarfs 138
 - Spaltendefinition 106
 - Logische Datenbankpartitionen 83
 - Logische Netzschnittstellen 333
 - Logischer Datenbankentwurf 101
 - Beziehungen 103
 - Definieren von Tabellen 103
 - Festlegen der zu speichernden Daten 101
 - Logischer Host 332
 - Löschbedingungen 124
- M**
 - Mehrpartitions-knotengruppe 72
 - Mehrpartitions-konfigurationen 81
 - Metadaten 91
 - Methoden
 - Sun Cluster 331
 - Microsoft Cluster Server (MSCS) 293
 - Microsoft Transaction Server
 - DB2 mit Beispielanwendung testen 230
 - Installation und Konfiguration 225
 - ODBC-Verbindungen wieder verwenden 229
 - Prüfen der Installation 226
 - Softwarevorbereitungen 225
 - TCP/IP-Übertragungen optimieren 230
 - Microsoft Transaction Server (Forts.)
 - Transaktionszeitlimit und DB2-Verbindungsverhalten 227
 - unterstützte DB2-Datenbank-Server 226
 - Unterstützung in DB2 aktivieren 225
 - Verbindungspool 227
 - Verbindungspool mit ADO 2.1 und später 229
 - Migration
 - Datenbank 407
 - Migrationsaufgaben für HACMP ES 281
 - Modell ohne gemeinsame Benutzung 233
 - MPP-Umgebung 81
 - MSCS (Microsoft Cluster Server) 293
 - Multimediaobjekte 102
- N**
 - Netscape-Browser
 - Installation 393
 - Neueste Informationen 388
 - NFS-Server-Knoten 254
 - NFS-Server-Übernahme, Konfigurationsbeispiel 255
 - Nicht wiederherstellbare Datenbanken 36
 - node_down-Ereignis 245
 - node_up-Ereignis 245
 - Normalisieren von Tabellen 113
 - NOT NULL, Integritätsbedingung 25
 - Nullwert 108
- O**
 - Objektnamen, zusammengesetzte Datenbank 404
 - Offline archivierte Protokoll-dateien 38
 - Online archivierte Protokoll-dateien 38
 - Online-Hilfefunktion 391
 - Online-Informationen anzeigen 393
 - suchen 398
- P**
 - Paralleler Ressourcenmanager 233
 - Parallelität
 - Abfrage 74
 - Arten 73
 - AutoLoader, Dienstprogramm 77

- Parallelität (*Forts.*)
 - BACKUP und RESTORE, Datenbankdienstprogramme 77
 - Dienstprogramm 77
 - Ein-/Ausgabe 74
 - LOAD, Dienstprogramm 77
 - partitionsintern 74
 - partitionsübergreifend 75
 - Übersicht 71
 - und Indexerstellung 77
 - und verschiedene Hardwareumgebungen 85
 - Parallelverarbeitung
 - Übersicht 129
 - Partition
 - Datenbank 71
 - Partitionen mit einem Prozessor 81
 - Partitionen mit mehreren Prozessoren 82
 - Partitionierte Datenbank 71
 - Partitionierung
 - Daten 146
 - Schlüssel 148
 - Zuordnung 147
 - Partitionsinterne Parallelität 74
 - zusammen mit partitionsübergreifender Parallelität 76
 - Partitionskompatibilität 151
 - Partitionsübergreifende Parallelität 75
 - zusammen mit partitionsinterner Parallelität 76
 - PDF 388
 - Physische Dateien
 - SMS 159
 - Physischer Datenbankentwurf 131
 - Platte
 - Array 54
 - RAID (Redundant Array of Independent Disks) 54
 - Striping 54
 - Platteneinheiten
 - Hardware 55
 - Software 56
 - Plattenfehler
 - Schützen vor 54
 - Plattengruppen 334
 - Primärindex 108
 - Primärschlüssel 108, 121
 - Primärschlüssel, Integritätsbedingung 25, 121
 - Primärschlüsselwerte
 - Generieren eindeutiger 111
 - Programmschritt 90
 - Protokolldatei
 - Abschätzen des Speicherbedarfs 142
 - Protokolldaten
 - Übersicht 128
 - Protokolle
 - aktiv 38
 - Benutzer-Exit-Programm 47
 - Datenbank 37
 - erforderlicher Speicher 47
 - online archivierte Protokolldateien 38
 - Protokollieren
 - Archivprotokolle 37
 - Umlaufprotokolle 37
 - Prozedurdateien für HACMP
 - ES 271
 - Installation 272
 - Prozeß (bei Warehouses) 89
 - Prüfaktivitäten
 - Übersicht 127
 - Prüfung auf Integritätsbedingung 27
 - Pufferpool
 - IBMDEFAULTBP 166
 - Übersicht 21
- R**
- RAID (Redundant Array of Independent Disks) 54, 55
 - RAID-1 (Plattenspiegelung oder -duplizierung) 55
 - RAID-5 (Daten- und Paritäts-Striping nach Sektoren) 55
 - RAID-Einheiten
 - Leistungsoptimierung 176
 - Räumlich
 - Daten 96
 - Informationen 93
 - Redundant Array of Independent Disks (RAID) 54
 - Referentielle Integritätsbedingungen 120
 - Auswirkungen auf SQL-Operationen 123
 - durch übergreifendes Löschen verknüpft 125
 - SQL-Anweisung DELETE, Regeln 124
 - SQL-Anweisung INSERT, Regeln 123
 - SQL-Anweisung UPDATE, Regeln 125
 - Referenztyp 107
 - Übersicht 128
 - Referenzzyklus 122
 - Relation
 - eins-zu-eins 105
 - eins-zu-viele 103
 - viele-zu-eins 103
 - viele-zu-viele 104
 - Relationale Datenbank, Konzepte
 - Übersicht 9
 - Release-Informationen 388
 - Release-zu-Release, Inkompatibilitäten
 - Beschreibung 413
 - Reorganisieren von Tabellen 59
 - Replizierte Übersichtstabellen 152
 - Ressourcenmanager
 - Einrichten einer Datenbank als 203
 - Rotierende Zuordnung 247
 - rules-Datei 245
 - Einschränkung 268
 - für HACMP 267
- S**
- Schema
 - Namenskonventionen 401
 - Übersicht 13
 - Schlüssel 108
 - Partitionierung 148
 - Schlüsselspalten
 - identifizieren 110
 - Schritt (bei Warehouses) 89
 - Schützen gegen Plattenfehler 54
 - SDR (System Data Repository) 256
 - Sicherheit 59, 128
 - Sicherung
 - Benutzer-Exit-Programm 48
 - Häufigkeit 44
 - Informationen zum Speicher 46
 - offline 44
 - online 44
 - Sicht
 - Namenskonventionen 403
 - Übersicht 12
 - Skalierbarkeit 78, 245
 - SMP-Clusterumgebung 82
 - SMP-Umgebung 79
 - SMS, physische Dateien 159
 - SMS (vom System verwalteter Bereich) 16, 157
 - SMS-Tabellenbereich 16, 157
 - SNA (Systems Network Architecture) 188
 - Softwareplatteneinheiten 56
 - Sortierfolge
 - allgemeine Überlegungen 460

- Sortierfolge (*Forts.*)
 - collating_sequence, Option 461
 - Überlegungen zu zusammenge-
schlossenen Datenbanken 460
 - SP-Frame 246
 - SP-Switch-Konfiguration, Überlegun-
gen 256
 - Spalte
 - Definieren für eine Tabelle 106
 - Namenskonventionen 403
 - Spatial Extender
 - Übersicht 93
 - Speicher
 - Datenträgerfehler 47
 - für Sicherung und Wiederher-
stellung erforderlicher 46
 - Speicherobjekte
 - Behälter 20
 - Pufferpool 21
 - Tabellenbereich 16
 - Übersicht 16
 - Spiegeln oder Duplizieren von Plat-
ten (RAID-1) 55
 - Spiegeln von Platten 56
 - SPM (Synchronisationspunktmana-
ger) 184
 - Sprachenkennung
 - Handbücher 387
 - SQL-Optimierungsprogramm 13
 - SQL-Schritt 90
 - Stammtyp 106
 - Standardagenten-Site 89
 - Sternschema 92
 - Steuermethoden 336
 - Störende Wartung 266
 - Strukturierter Typ 106
 - Übersicht 128
 - Suche
 - Online-Informationen 398
 - Suchen
 - Online-Informationen 395
 - Sun Cluster 2.x 327
 - Switch-Aliasadresse 253
 - Synchronisation
 - Datenbankpartition 57
 - Knoten 57
 - Überlegungen zur Wiederher-
stellung 57
 - Synchronisationspunktmanager
(SPM) 184
 - SYSCATSPACE 155
 - System Data Repository (SDR) 256
 - Systemkatalogtabelle
 - Ermitteln der Anfangsgröße 135
 - Übersicht 14
 - Systemobjekte
 - Konfigurationsparameter 22
 - Übersicht 22
 - Systemprotokollfunktion
 - XA-Schnittstelle, Beispiel 219
 - Systems Network Architecture
(SNA) 188
 - Systemtabellenbereich, tempor-
är 156
- ## T
- Tabelle
 - Abschätzen des Speicher-
bedarfs 134
 - Benutzer 135
 - Beziehungen 48
 - Kollokation (Zusammenfassen
von Tabellen) 151
 - Namenskonventionen 403
 - Normalisierung 113
 - Prüfungen auf Integritätsbedin-
gung 126
 - reorganisieren 59
 - Systemkatalog 135
 - Übersicht 11
 - Zuordnen zu Tabellen-
bereichen 168
 - Tabelle erstellen, Assistent 396
 - Tabellenbereich
 - aktualisierende Wiederher-
stellung 34
 - Art der Auslastung, Überlegun-
gen 173
 - Auswählen von SMS oder
DMS 174
 - Benutzer 155
 - DMS-Tabellenbereich 161
 - Ein-/Ausgabe (E/A), Überlegun-
gen 164
 - Entwerfen und Auswählen 153
 - Entwurf 164
 - Katalog 155, 172
 - Knotengruppen zuordnen 167
 - Namenskonventionen 403
 - Pufferpools zuordnen 166
 - SMS-Tabellenbereich 157
 - SYSCATSPACE 155
 - temporär 156, 170
 - TEMPSPACE1 156
 - Übersicht 16
 - USERSPACE1 155
 - Wiederherstellen 33
 - Wiederherstellung 49
 - Tabellenbereich erstellen, Assi-
stent 396
 - Teilentclustering 147
 - Temporärer Arbeitsbereich
 - Abschätzen des Speicher-
bedarfs 144
 - Temporärer Tabellenbereich 156,
170
 - TEMPSPACE1 156
 - Themenbereich 88
 - TPM und TP_MON_NAME, Wer-
te 206
 - Transaktion
 - die auf partitionierte Daten-
banken zugreifen 210
 - fest gekoppelt 200
 - globale 199
 - lose gekoppelt 200
 - Nicht-XA 199
 - zweiphasige Festschreibung 199
 - Transaktionsfehler
 - Begrenzen der Auswirkun-
gen 57
 - Transaktionsmanager
 - mit IBM TXSeries CICS imple-
mentieren 219
 - mit IBM TXSeries Encina imple-
mentieren 219
 - Konfigurieren von DB2 220
 - Konfigurieren von Encina für
jeden Ressourcenmana-
ger 220
 - Verweisen auf DB2-Datenbank
aus Encina-
Anwendung 221
 - mit Microsoft Transaction Server
implementieren 224
 - mit Tuxedo implementieren 222
 - Transaktionsmanager (TM) 183, 184
 - Transaktionsverarbeitung
 - XA-Transaktionsmanager konfi-
gurieren 219
 - Typhierarchie 106
 - Übersicht 128
 - Typisiert
 - Sicht 107
 - Tabelle 107, 128
- ## U
- Übergeordnet
 - Schlüssel 121
 - Tabelle 121
 - Zeile 122
 - Übergeordneter Typ 106
 - Übersicht über Data Warehouses 87
 - Übersichtstabellen
 - repliziert 152
 - Übersicht 128

UDF (benutzerdefinierte Funktion) 107
 Uhrzeit
 Definition 462
 Formate 465
 Umlaufprotokolle 37
 Umsetzungsschritt 91
 Unbestätigte Transaktionen 210
 manuelle Wiederherstellung 211
 resynchronisieren 195
 wiederherstellen 194, 200
 Unterbrechungsfreie Wartung 266
 Untergeordnete Tabelle 122
 Untergeordneter Typ (Subtyp) 106
 Unterstützte DB2-Datenbank-Server für MTS-koo­rdinierte Transaktionen 226
 Unterstützung von Landessprachen (NLS)
 CCSID-Unterstützung für bidirektionale Zeichen 454
 Werte für Datum und Uhrzeit 462
 Zeichensätze 453
 USERSPACE1 155

V

Verbindungspool, MTS 227
 Verknüpfungspfad 112
 Verringern der Protokollierung für Arbeitstabellen 42
 Versionswiederherstellung 31
 Verteilte Anforderung 65
 Verteilte Datenbank 179
 Verteilte Transaktionsverarbeitung
 Anwendungsprogramm 198
 Fehlerbehandlung 210
 RELEASE, Anweisung 210
 Ressourcenmanager 201
 Transaktionsmanager 200
 Überlegungen zu Datenbankverbindungen 210
 Überlegungen zur Konfiguration 214
 Überlegungen zur Sicherheit 213
 Unterstützung für Host- und AS/400-Datenbank-Server 209
 Verzeichnis
 Datenbank 131
 Vierte Normalform 117
 Vorkommen 102

W

Warehouse 88
 Agent 88

Warehouse 88 (*Forts.*)
 Prozess 89
 Quelle 88
 Ziel 88
 Warehouse-Schritt 89
 benutzerdefiniertes Programm 91
 Programm 90
 SQL 90
 Umsetzung 91
 Warehouses
 Erstellungsaufgaben 91
 Objekte 88
 Übersicht 87
 Werte für Datum und Uhrzeit
 Darstellung als Zeichenfolge 463
 Übersicht 462
 Wiederherstellbare Datenbanken 36
 Wiederherstellen
 Datenbank 31
 Tabellenbereich 33
 Wiederherstellen, Assistent 396
 Wiederherstellung 28
 benötigte Zeit 46
 beschädigte Tabellenbereiche 49
 Datei des Wiederherstellungsprotokolls 15
 Einschränkungen von Betriebssystemen 48
 Ende der Protokolldateien 34
 Informationen zum Speicher 46
 Leistung 51
 Protokolldatei 15
 Punkt 44
 Verringern der Protokollierung für Arbeitstabellen 42
 Zeitpunkt 34
 zu berücksichtigende Faktoren 35
 Wiederherstellung auf Datenbank­ebene 31
 Wiederherstellung nach einem Systemabsturz 30
 Wiederherstellungsobjekte
 Datei des Wiederherstellungsprotokolls 15
 Protokolldatei 15
 Übersicht 14
 Wiederherstellungsprogrammdatei für HACMP ES 270
 Wiederherstellungsprozeduren für HACMP ES 274
 Wiederherstellungspunkt 44

Windows-Codepages
 DB2CODEPAGE, Registrierdatenbankvariable 452
 unterstützte Codepages 452
 Windows NT, Funktionsübernahme
 Arten 294
 Ausführen von Prozeduren, bevor DB2-Ressource online gebracht ist 317
 Ausführen von Prozeduren, nachdem DB2-Ressourcen online gebracht wurden 319
 Ausführen von Prozeduren, Übersicht 316
 Ausgleichen der Datenbanklaufwerkzuordnung 308
 Begrenzungen 325
 Benutzer- und Gruppenunterstützung 321
 Bereitschaftsmodus (Hot Standby) 295
 DB2MCS, Dienstprogramm
 Einrichten eines Datenbanksystems mit einer Partition 302
 Einrichten eines partitionierten Datenbanksystems 303
 Einrichten zweier Datenbanken mit einer Partition für gegenseitige Übernahme 302
 Parameter in
 DB2MCS.CFG 297
 Übersicht 296
 Einrichten eines partitionierten Datenbanksystems zur gegenseitigen Übernahme
 Ausführen des Dienstprogramms DB2MCS 314
 Registrieren der Datenbanklaufwerkzuordnung für ClusterA 315
 Registrieren der Datenbanklaufwerkzuordnung für ClusterB 315
 Vorbereitung 313
 Ziele 311
 Einrichten zweier Exemplare für gegenseitige Übernahme, Beispiel
 Ausführen des Dienstprogramms DB2MCS 310
 Vorbereitung 310
 Ziele 309
 Einschränkungen 325

Windows NT, Funktionsübernahme
(Forts.)
Einstellen der Datenbanklaufwerkzuordnung für gegenseitige Übernahme in Umgebung mit partitionierter Datenbank 306
gegenseitige Übernahme 295
Pflegen des MSCS-Systems 304
planen 293
Starten und Stoppen von DB2-Ressourcen 315
Überlegungen zu Datenbanken 321
Überlegungen zum Verwaltungs-Server 323
Überlegungen zur Funktionsrückübertragung 305
Überlegungen zur Kommunikation 322
Überlegungen zur Steuerzentrale 323
Überlegungen zur Systemzeit 323
Überlegungen zur Verwaltung von DB2 315

X

X/Open-
Transaktionsmanagerschnittstelle (XA)
verteilte Transaktionsverarbeitung (DTP) 198
XA-Transaktionsmanager
konfigurieren 219
XA-Zeichenfolge zum Öffnen 203

Z

Zeichenfolgen für das Datum
Definition 463
Zeichenfolgen für die Uhrzeit
Definition 464
Zeichenfolgen für Zeitmarke
Definition 465
Zeit, für Datenbankwiederherstellung benötigte 46
Zeitmarke
Definition 462
Ziel
Sicht 107
Tabelle 107
Typ 107
Zeile 107
Zugriffsrecht 61

Zuordnen
Tabellen zu Tabellenbereichen 168
Tabellenbereiche zu Knotengruppen 167
Tabellenbereiche zu Pufferpools 166
Zuordnung
Partitionierung 147
Zusammengeschlossene Datenbank
Authentifizierung 62
Berechtigung 62
Namen in Groß-/Kleinschreibung 405
Objektnamen 404
Sortierfolgen, Richtlinien 460
Systeme 65
Überlegungen zum Entwurf 178
Zusammengesetzter Schlüssel 110, 121
Zweiphasige Festschreibung 181, 183, 190
Fehlerbehandlung 193
Zweite Normalform 114

Kontaktaufnahme mit IBM

Bei technischen Problemen lesen Sie bitte die entsprechenden Korrekturmaßnahmen im Handbuch *Troubleshooting Guide* und führen Sie diese aus, bevor Sie sich mit der IBM Kundenunterstützung in Verbindung setzen. Mit Hilfe dieses Handbuchs können Sie Informationen sammeln, die die DB2-Kundenunterstützung zur Fehlerbehebung verwenden kann.

Wenn Sie weitere Informationen benötigen oder eines der DB2 Universal Database-Produkte bestellen möchten, setzen Sie sich mit einem IBM Ansprechpartner in einer lokalen Geschäftsstelle oder einem IBM Software-Vertriebspartner in Verbindung.

Telefonische Unterstützung erhalten Sie über folgende Nummern:

- Unter 0180 3/313 233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0190/772 243 erreichen Sie die DB2 Helpline, wo Sie Antworten zu DB2-spezifischen Problemen erhalten.

Produktinformationen

Telefonische Unterstützung erhalten Sie über folgende Nummern:

- Unter 0180 3/313 233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0180/55 090 können Sie Handbücher telefonisch bestellen.

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

Auf den DB2-World Wide Web-Seiten erhalten Sie aktuelle DB2-Informationen wie Neuigkeiten, Produktbeschreibungen, Schulungspläne und vieles mehr.

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

Mit **DB2 Product and Service Technical Library** können Sie auf häufig gestellte Fragen, Berichtigungen, Handbücher und aktuelle technische DB2-Informationen zugreifen.

Anmerkung: Diese Informationen stehen möglicherweise nur auf Englisch zur Verfügung.

<http://www.elink.ibm.com/pbl/pbl/>

Auf der Web-Site für die Bestellung internationaler Veröffentlichungen (International Publications) finden Sie Informationen zum Bestellverfahren.

<http://www.ibm.com/education/certify/>

Das 'Professional Certification Program' auf der IBM Web-Site stellt Zertifizierungstestinformationen für eine Reihe von IBM Produkten, u. a. auch DB2, zur Verfügung.

<ftp://software.ibm.com>

Melden Sie sich als *anonymous* an. Im Verzeichnis /ps/products/db2 finden Sie Demo-Versionen, Berichtigungen, Informationen und Tools zu DB2 und vielen zugehörigen Produkten.

<comp.databases.ibm-db2>, <bit.listserv.db2-1>

Über diese Internet-Newsgroups können DB2-Benutzer Ihre Erfahrungen mit den DB2-Produkten austauschen.

Für CompuServe: GO IBMDB2

Geben Sie diesen Befehl ein, um auf IBM DB2 Family Forums zuzugreifen. Alle DB2-Produkte werden über diese Foren unterstützt.

In Anhang A des Handbuchs *IBM Software Support Handbook* finden Sie Informationen dazu, wie Sie sich mit IBM in Verbindung setzen können. Rufen Sie die folgende Web-Seite auf, um auf dieses Dokument zuzugreifen:
<http://www.ibm.com/support/>. Wählen Sie anschließend die Verbindung zum IBM Software Support Handbook am unteren Rand der Seite aus.

Anmerkung: In einigen Ländern sollten sich die IBM Vertragshändler an die innerhalb ihrer Händlerstruktur vorgesehene Unterstützung wenden, nicht an die IBM Unterstützungsfunktion.

Antwort

**IBM DB2 Universal Database
Systemverwaltung: Konzept
Version 7**

IBM Form SC12-2879-00

Anregungen zur Verbesserung und Ergänzung dieser Veröffentlichung nehmen wir gerne entgegen. Bitte informieren Sie uns über Fehler, ungenaue Darstellungen oder andere Mängel.

Zur Klärung technischer Fragen sowie zu Liefermöglichkeiten und Preisen wenden Sie sich bitte entweder an Ihre IBM Geschäftsstelle, Ihren IBM Geschäftspartner oder Ihren Händler.

Unsere Telefonauskunft "HALLO IBM" (Telefonnr.: 01803/31 32 33) steht Ihnen ebenfalls zur Klärung allgemeiner Fragen zur Verfügung.

Kommentare:

Danke für Ihre Bemühungen.

Sie können ihre Kommentare betr. dieser Veröffentlichung wie folgt senden:

- Als Brief an die Postanschrift auf der Rückseite dieses Formulars
- Als E-Mail an die folgende Adresse: comment@tcvm.vnet.ibm.com

Name

Adresse

Firma oder Organisation

Rufnummer

E-Mail-Adresse

Antwort
SC12-2879-00



IBM Deutschland Informationssysteme GmbH
SW NLS Center

70548 Stuttgart



Teilenummer: CT7XVNA

Gedruckt in Deutschland

SC12-2879-00



CT7XVNA

