

EINSTIEG IN

DB2 Express-C

Ein Buch von der Community für die Community

RAUL CHONG, IAN HAKES, RAV AHUJA

VORWORT VON DR. ARVIND KRISHNA



ZWEITE AUSGABE

Zweite Ausgabe (April 2008)

Diese Ausgabe wurde für die IBM® DB2® Express-C Version 9.5 für Linux® UNIX® und Windows® aktualisiert.

© Copyright IBM Corporation, 2007, 2008. Alle Rechte vorbehalten.

Inhaltsverzeichnis

Über dieses Buch	8
Hinweise und Warenzeichen	8
Wer sollte dieses Buch lesen?	9
Wie ist dieses Buch strukturiert?	9
Ein Buch für die Community	9
Mitwirkende	11
Danksagungen	11
Vorwort	12
TEIL I – ÜBERBLICK UND INSTALLATION	14
Kapitel 1 – Was ist DB2 Express-C?	15
1.1 Frei für Entwicklung, Implementierung und Verteilung... grenzenlos!	16
1.2 Benutzerunterstützung und technischer Support	16
1.3 DB2 Server	17
1.4 DB2 Clients und Treiber	17
1.5 Unabhängig in der Anwendungsentwicklung	19
1.6 DB2 Versionen im Vergleich zu DB2 Editionen	19
1.7 Wechsel zu einer höheren DB2 Edition	20
1.8 Wartung von DB2 Express-C	20
1.9 Verwandte freie Software	21
1.9.1 IBM Data Studio	21
1.9.2 DB2 Net Search Extender	21
1.9.3 Starter Toolkit for DB2 on Rails	21
1.9.4 Web 2.0 Starter Toolkit for DB2	22
1.9.5 WebSphere Application Server – Community Edition	22
Kapitel 2 – Verwandte Funktionalitäten und Produkte	23
2.1 In der DB2 Express-C Subskription enthaltene Funktionalitäten	27
2.1.1 Fix Packs	27
2.1.2 Hochverfügbarkeit und Ausfallwiederherstellung (HADR)	27
2.1.3 Datenreplikation	28
2.2 Mit DB2 Express-C nicht verfügbare Funktionalitäten	28
2.2.1 Datenbankpartitionierung (DPF)	29
2.2.2 Verbindungsoptimierung (Connection Concentrator)	29
2.2.3 Geodetic Extender	29
2.2.4 Workload Management (WLM)	29
2.3 Kostenpflichtige, mit DB2 verwandte Produkte	30
2.3.1 DB2 Connect	30
2.3.2 WebSphere Federation Server	30
2.3.3 WebSphere Replication Server	32
Kapitel 3 – DB2 Installation	33
3.1 Installationsvoraussetzungen	33
3.2 Betriebssystemberechtigung für die Installation	33
3.3 Installationsassistent	34

3.4 Silent Installation	39
Übung 1: Installieren von DB2 Express-C und Erstellen der SAMPLE Datenbank	41
Kapitel 4 – DB2 Umgebung	45
Übung 2 – Erstellen einer neuen Datenbank	56
4.1 DB2 Konfiguration	57
4.1.1 Umgebungsvariablen	58
4.1.2 Datenbankmanager Konfigurationsdatei (DBM CFG)	58
4.1.3 Datenbankkonfigurationsdatei (DB CFG)	60
4.1.4 DB2-Registrierdatenbank	62
4.2 Der DB2 Administration Server	63
Übung 3 – Arbeiten mit Instanzen, Datenbanken und Konfiguration	65
Kapitel 5 – DB2 Dienstprogramme	67
5.1 Steuerzentrale	69
5.2 Befehlseditor	71
5.3 SQL Assist Assistent	73
5.4 SQL anzeigen Schaltfläche	75
Übung 4: Die EXPRESS Datenbank mit Skripten füllen	76
5.5 Scripting	78
5.5.1 SQL Skripte	78
5.5.2 Betriebssystem (Shell) Skripte	79
Übung 5: Erstellen eines Installationsskripts für die EXPRESS Datenbank	81
5.6 Taskzentrale	84
5.6.1 Die Toolskatalog Datenbank	84
5.7 Journal	85
5.8 Diagnosezentrale	87
5.8.1 Diagnosezentrale	87
TEIL II – DB2 LERNEN: DATENBANK ADMINISTRATION	91
Kapitel 6 – DB2 Architektur	93
6.1 DB2 Prozess Modell	93
6.2 DB2 Arbeitsspeichermodell	95
6.3 DB2 Datenspeicherungsmodell	96
6.3.1 Pages und Extents	97
6.3.2 Pufferpools	97
6.3.3 Tabellenbereiche	99
Kapitel 7 – DB2 Client Konnektivität	105
7.1 DB2 Verzeichnisse	105
7.2 Konfigurationsassistent	106
7.2.1 Auf dem Server erforderliche Einrichtung	107
7.2.2 Auf dem Client erforderliche Einrichtung	110
7.2.3 Client und Server Profile erstellen	114
Übung 6: Verwenden des Konfigurationsassistenten	117
Kapitel 8 – Mit Datenbankobjekten arbeiten	121
8.1 Schema	121

8.2 Tabellen	122
8.2.1 Datentypen	122
8.2.2 Identität Spalten	125
8.2.3 SEQUENZ Objekte	125
8.2.4 Systemkatalogtabellen	126
8.2.5 Deklarierte temporäre Tabellen	127
Übung 7: Erstellen einer neuen Tabelle	129
8.3 Sichten	132
8.4 Indizes	132
8.4.1 Designadvisor	132
8.5 Referentielle Integrität	134
Kapitel 9 – Die EXPORT und IMPORT Dienstprogramme	137
9.1 Das EXPORT Dienstprogramm	138
9.2 Das IMPORT Dienstprogramm	139
9.3 Das LOAD Dienstprogramm	140
9.4 Das db2move Dienstprogramm	142
9.5 Das db2look Dienstprogramm	142
Übung 8 – Extrahieren der DDL der EXPRESS Datenbank	145
Kapitel 10 – Datenbanksicherheit	149
10.1 Identitätsprüfung	150
10.2 Berechtigung	151
10.3 DBADM Berechtigung	154
10.4 Die PUBLIC Gruppe	154
10.5 Die GRANT und REVOKE Anweisungen	155
10.6 Prüfen der Berechtigungen und Zugriffsrechte	155
10.7 Überlegungen zu Gruppenberechtigungen	157
Übung 9 – Erteilen und entziehen von Benutzerberechtigungen	158
Kapitel 11 – Sicherung und Wiederherstellung	161
11.1 Datenbank Protokollierung	161
11.2 Protokollarten	162
11.3 Arten der Protokollierung	163
11.3.1 Umlaufprotokollierung	163
11.3.2 Archivprotokollierung oder Protokollspeicherung	163
11.4 Datenbankprotokollierung über die Steuerzentrale	164
11.5 Protokollparameter	166
11.6 Datenbanksicherung	166
Übung 10 – Planen einer Sicherung	169
11.7 Datenbankwiederherstellung	172
11.7.1 Wiederherstellungsarten	172
11.7.2 Datenbankwiederherstellung	173
11.8 Andere Funktionalitäten mit BACKUP und RESTORE	173
Kapitel 12 – Wartungsarbeiten	175
12.1 REORG, RUNSTATS, REBIND	175
12.1.1 Der REORG Befehl	176
12.1.2 Der RUNSTATS Befehl	176

12.1.3 BIND / REBIND	177
12.1.4 Wartungsarbeiten über die Steuerzentrale ausführen	178
12.2 Verwaltungsoptionen	179
Kapitel 13 – Gleichzeitiger Zugriff und Datensatzsperrern	185
13.1 Transaktionen	185
13.2 Gleichzeitiger Zugriff (Concurrency)	186
13.3 Probleme ohne Kontrolle des gleichzeitigen Zugriffs	187
13.3.1 Lost Update	188
13.3.2 Uncommitted read	188
13.3.3 Non-repeatable read	189
13.3.4 Phantom read	190
13.4 Isolationsstufen (Isolation Level)	190
13.4.1 Uncommitted Read	191
13.4.2 Cursor Stability	191
13.4.3 Read Stability	191
13.4.4 Repeatable Read	192
13.4.5 Vergleich der Isolationsstufen	192
13.4.6 Setzen der Isolationsstufe	193
13.5 Lock escalation	194
13.6 Überwachung von Sperrern (Lock monitoring)	195
13.7 Zeitlimit einer Sperrung (Lock wait)	196
13.8 Deadlock – Ursachen und Erkennung	197
13.9 Empfohlene Vorgehensweise bei gleichzeitigem Zugriff und Datensatzsperrern	198
TEIL III – DB2 LERNEN: ANWENDUNGSENTWICKLUNG	201
Kapitel 14 – SQL PL gespeicherte Prozeduren	203
14.1 Das IBM Data Studio	204
14.1.2 Erstellung einer gespeicherten Prozedur im Data Studio	205
14.2 Grundlagen SQL PL gespeicherter Prozeduren	209
14.2.1 Die Struktur gespeicherter Prozeduren	209
14.2.2 Optionale Eigenschaften gespeicherter Prozeduren	210
14.2.3 Parameter	210
14.2.4 Kommentare in einer SQL PL gespeicherten Prozedur	211
14.2.5 Zusammengesetzte Anweisungen	211
14.2.6 Variablendeklaration	212
14.2.7 Zuweisungsbefehle	212
14.3 Cursor	213
14.4 Flusskontrolle	213
14.5 Aufruf gespeicherter Prozeduren	214
14.6 Fehler und Zustandsverarbeitung	215
14.7 Dynamisches SQL	217
Kapitel 15 – Inline SQL PL, Trigger und benutzerdefinierte Funktionen	219
15.1 Inline SQL PL	219
15.2 Trigger	220
15.2.1 Arten von Triggern	221
Übung 12 – Erstellen eines Triggers über die Steuerzentrale	224

15.3 Benutzerdefinierte Funktionen (UDFs)	227
15.3.1 Skalarfunktionen	227
15.3.2 Tabellenfunktionen	228
Übung 13 – Erstellen einer benutzerdefinierten Funktion (UDF) mit dem IBM Data Studio	230
Kapitel 16 – DB2 pureXML	233
16.1 XML mit Datenbanken verwenden	234
16.2 XML Datenbanken	234
16.2.1 XML-aktivierte Datenbanken	234
16.2.2 Native XML Datenbanken	235
16.3 XML in DB2	236
16.3.1 pureXML Technologievorteile	237
16.3.2 XPath Grundlagen	240
16.3.3 XQuery definiert	243
16.3.4 XML Dokumente einfügen	245
16.3.5 XML Daten abfragen	248
16.3.6 Joins mit SQL/XML	254
16.3.7 Joins mit XQuery	255
16.3.8 Update und Delete Operationen	256
16.3.9 XML Indizierung	258
Übung 14 - SQL/XML und XQuery	260
Kapitel 17 – Entwickeln mit Java, PHP und Ruby	261
17.1 Anwendungsentwicklung mit Java	261
17.1.1 JDBC Typ 2 Treiber	261
17.1.2 JDBC Typ 4 Treiber	262
17.2 Anwendungsentwicklung mit PHP	263
17.2.1 DB2 Verbindungsoptionen für PHP	263
17.2.2 Zend Core for IBM	265
17.3 Anwendungsentwicklung mit Ruby on Rails	267
17.3.1 Starter Toolkit für DB2 on Rails	267
Anhang A – Problemlösung	269
A.1 Weitere Informationen über Fehlercodes erhalten	270
A.2 SQLCODE und SQLSTATE	270
A.3 Administratives DB2 Nachrichtenprotokoll	271
A.4 db2diag.log	271
A.5 CLI Fehlerprotokollierung	272
A.6 DB2 Fehler und Fixes	272
Ressourcen	273
Websites:	273
Bücher:	274

Über dieses Buch

Hinweise und Warenzeichen

© Copyright IBM Corporation 2007, 2008
Alle Rechte vorbehalten.
IBM Canada
8200 Warden Avenue
Markham, ON
L6G 1C7
Canada

Dieses Dokument oder Teile davon dürfen ohne die vorherige Zustimmung aller oben genannten Copyrighteigner weder kopiert noch in anderer Form reproduziert oder in eine andere Sprache übersetzt werden.

IBM übernimmt keine Gewährleistung für den Inhalt des Dokuments und lehnt insbesondere jede stillschweigende Gewährleistung für die Handelsüblichkeit oder Verwendungsfähigkeit für einen bestimmten Zweck ab. IBM übernimmt keine Verantwortung für Fehler, die in diesem Dokument enthalten sein könnten. Dies gilt auch ohne Einschränkung für Übersetzungsfehler. Die Informationen in diesem Dokument können Änderungen unterliegen, ohne dass diese angekündigt werden müssen. IBM behält sich vor, solche Änderungen vorzunehmen, und ist nicht verpflichtet, irgendeine Person über diese Änderungen zu informieren. IBM ist nicht verpflichtet, die Informationen in diesem Dokument auf dem jeweils aktuellsten Stand zu halten.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

IBM, das IBM logo, AIX, DB2, DB2 Connect, DB2 Universal Database, i5/OS, Informix, pureXML, Tivoli, WebSphere und z/OS sind Warenzeichen oder eingetragene Warenzeichen der International Business Machines Corporation in den Vereinigten Staaten, anderen Ländern, oder beides.

Java und alle Java-basierten Warenzeichen sind Warenzeichen der Sun Microsystems, Inc. in den Vereinigten Staaten, anderen Ländern, oder beides.

Microsoft und Windows sind Warenzeichen der Microsoft Corporation in den Vereinigten Staaten, anderen Ländern, oder beides.

Linux ist ein eingetragenes Warenzeichen von Linus Torvalds in den Vereinigten Staaten, anderen Ländern, oder beides.

UNIX ist ein eingetragenes Warenzeichen der The Open Group in den Vereinigten Staaten, anderen Ländern, oder beides.

Andere Firmen-, Produkt- oder Dienstleistungsamen können Warenzeichen oder Dienstleistungsmarken anderer sein.

Hinweise auf IBM Produkte, Programme und Services in dieser Veröffentlichung bedeuten nicht, dass IBM diese in allen Ländern, in denen IBM vertreten ist, anbietet.

Wer sollte dieses Buch lesen?

Dieses Buch ist für alle gedacht, welche mit Datenbanken arbeiten oder dies vorhaben, wie Datenbankadministratoren (DBAs), Anwendungsentwickler, Berater, Softwarearchitekten, Produktmanager, Ausbilder und Studenten.

Wie ist dieses Buch strukturiert?

Teil I, Überblick und Setup, erläutert worum es sich bei der DB2 Express-C Edition handelt, stellt die DB2 Produktpalette und deren Funktionsumfang vor, unterstützt bei der Installation und Erstellung von Datenbanken und erkundet die von DB2 zur Verfügung gestellten Dienstprogramme.

Teil II, DB2 lernen: Datenbankadministration ist ausgelegt um Sie mit der DB2 Umgebung, Architektur, Remote Verbindungsmöglichkeiten, Datenbankobjekten, Daten kopieren (Import/Export/Load), Sicherung und Wiederherstellung, gleichzeitiger Zugriff und Sperren, sowie anderen allgemeinen Verwaltungsaufgaben vertraut zu machen.

Teil III - DB2 lernen: Anwendungsentwicklung, beinhaltet gespeicherte Prozeduren, benutzerdefinierte Funktionen, Trigger, SQL/XML, XQuery, Entwicklung in Java™, PHP und Ruby.

Der Anhang enthält nützliche Informationen zur Fehlersuche.

Die meisten Kapitel enthalten Übungen. Die für diese Übungen benötigten Eingabedateien befinden sich in der diesem Buch beigelegten ZIP Datei `expressc_book_quicklabs_9.5.zip` oder werden auf der IBM® DB2 Express-C Website www.ibm.com/db2/express zur Verfügung gestellt.

Das Material in diesem Buch wird auch für als Teil des "DB2 on Campus" Programms verwendet und stimmen in etwa mit den unter www.channelDB2.com/oncampus zur Verfügung gestellten Videopräsentationen überein. Dieses Buch kann Ihnen auch bei der Vorbereitung auf die DB2 on Campus Prüfung helfen. Durch diese Prüfung erhalten Sie nach Beendigung des Programms eine Bestätigung, welche bescheinigt dass sie den Gegenwert von 16 Stunden DB2 Training erhalten haben. Auf der DB2 Express-C Website www.ibm.com/db2/express/students.html können Sie mehr über das Programm lesen.

Hinweis:

Für weitere Informationen über das DB2 on Campus Programm können Sie sich dieses Video ansehen
<http://www.channeldb2.com/video/video/show?id=807741:Video:3902>

Ein Buch für die Community

Dieses Buch wurde vom DB2 Express-C Team erstellt und für die DB2 Express-C Community kostenlos veröffentlicht. Zahllose Mitglieder der Community aus der ganzen

Welt haben das Buch in verschiedene Sprachen übersetzt. Für Rückmeldungen, Einreichen von neuem Material, die Verbesserung existierenden Materials oder Unterstützung bei der Übersetzung in eine andere Sprache senden Sie bitte eine E-Mail mit dem Betreff "DB2 Express-C book changes" an db2x@ca.ibm.com.

Mitwirkende

Wir möchten uns bei den freiwillig mitwirkenden Übersetzern der DB2 Community für ihre ausgezeichnete Arbeit Job bedanken! Sie verbrachten viele Stunden und Wochenenden mit der Realisierung der Übersetzung. In der Tabelle unten finden Sie die Beiträge der freiwillig Mitwirkenden:

Name des Beitragenden	Firma	Berufsbezeichnung	Beitrag
Raul F. Chong	IBM	DB2 on Campus Program Manager	Leiter des deutschen Übersetzungsteams.
Leo Elsenberg		Business data processing specialist	Übersetzung des gesamten Buchs in die deutsche Sprache.
Guenter Weber	IBM	IBM Accredited DB2 Specialist	Verifizierung der Fehlerfreiheit der Übersetzung.

Danksagungen

Ganz besonders möchten wir uns bei den nachfolgenden Personen für deren Hilfe und die Entwicklung von in diesem Buch verwendetem Material bedanken:

- Ted Wasserman, Clara Liu und Paul Yip vom IBM Toronto Labor, welche das die Grundlage dieses Buches ausmachende Material entwickelten.
- Don Chamberlin und Cindy Saracco für ihre IBM developerWorks Artikel über XQuery und Matthias Nicola für seine pureXML™ Präsentationen.
- Kevin Czap und Grant Hutchison für die Entwicklung des technischen DB2 Besprechungsmaterials.
- Katherine Boyachok für den Umschlagentwurf dieses Buchs.
- Susan Visser für ihre Hilfe bei der Veröffentlichung dieses Buchs.

Vorwort

Innovation ist der Eckpfeiler des technologischen Fortschritts. Bei IBM war Innovation immer Bestandteil in der Evolution unserer Datenserver. Nachdem in den 60er und 70er Jahren des letzten Jahrhunderts der Weg der Datenmanagement Techniken bereitet wurde, haben wir fortlaufend innovative Informationsmanagementtechnologien geliefert, welche sich in tausenden, von IBM Technologen verfassten, Datenmanagement Patenten widerspiegeln. Daraus resultiert, dass sich heute einige der größten Organisationen weltweit auf IBM Produkte, wie DB2, verlassen um deren höchst anspruchsvolle und Missionskritischen Datenmanagement Lösungen zu betreiben.

Wie dem auch sei, DB2 ist nicht mehr nur für große Unternehmen. Mit der Veröffentlichung von DB2 Express-C, ist die preisgekrönte DB2 Technologie jetzt verfügbar um den Anforderungen kleiner und mittelgroßer Firmen gerecht zu werden – und dies ohne obligatorische Kosten! Obwohl es andere freie oder Open-Source Datenserver gibt, bietet DB2 Express-C einzigartige Vorteile gegenüber diesen Alternativen.

DB2 Express-C enthält viele technologische Innovationen. Einige dieser Innovationen sind auf die fortgeschrittenen neuen Möglichkeiten ausgerichtet, andere zur Reduzierung des Administrationsaufwandes, einige zur Performancesteigerung und weitere zur Reduzierung der Infrastrukturkosten. Wir werden, in der Hoffnung dass Sie angeregt werden das Buch zu lesen, die meisten nicht hier besprechen – doch werden wir einen als Appetitanreger kurz beschreiben.

DB2 Express-C basiert auf der 'Viper' Technologie und macht sie zum ersten hybriden Datenserver für die Verwaltung relationaler Daten und XML Daten in den jeweiligen nativen Formaten. Dies macht DB2 ideal für den Betrieb einer neuen Gattung von SOA und Web 2.0 Anwendungen, in welchen XLM Daten im Überfluss fließen. Im Gegensatz zu Datenservern von anderen Firmen, limitiert DB2 Express-C nicht die Datenmenge, welche in einer Datenbank gespeichert werden kann, oder die Anzahl der Datenbanken, welche auf einem System erstellt werden können. Und Sie erhalten selbstverständlich Support, falls sie diesen benötigen – Hilfe ist nur einen Klick entfernt.

Dieses Buch dient als Leitfaden zur Einführung und der Arbeit mit DB2 Express-C. Es wird Ihnen dabei helfen die DB2 Konzepte zu verstehen und es ihnen ermöglichen das Handwerkszeug für die DB2 Administration und Anwendungsentwicklung zu erlernen. Das Können und Wissen, welches Sie mit Hilfe dieses Buchs gewonnen haben, ist sehr wichtig für andere fortgeschrittene DB2 Editionen unter Linux, UNIX und Windows.

Obwohl DB2 Express-C kein Open-Source Produkt ist, sind wir bei IBM fest davon überzeugt Community Initiativen zu unterstützen und zu pflegen. Ich bin erfreut, dass dieses Buch von Mitgliedern der DB2 Express-C Community entwickelt wird und jedem in der Community frei zugänglich ist. Damit andere von Ihrem Wissen profitieren können, möchte ich sie ermutigen das Buch mit ihrem Know-how und ihren Erfahrungen zu bereichern und zu aktualisieren, und bei der Übersetzung dieses Buchs in andere Sprachen behilflich zu sein.

A handwritten signature in black ink that reads "Arvind Krishna". The signature is written in a cursive, flowing style.

Arvind Krishna
Vice President, Data Servers
Information Management, IBM Software Group

TEIL I – ÜBERBLICK UND INSTALLATION

1

Kapitel 1 – Was ist DB2 Express-C?

DB2 Express-C gehört zur Familie leistungsstarker Datenserver Software zur Verwaltung von sowohl relationalen als auch XML Daten. DB2 Express-C ist eine freie, grenzenlose und leicht zu verwendende Edition von DB2. Das 'C' in DB2 Express-C steht für Community. Eine Community von DB2 Anwendern findet sich zur gegenseitigen Hilfe zusammen, sowohl Online als auch Offline. Die DB2 Express-C Community besteht aus den unterschiedlichsten Menschen und Firmen, welche Datenbanklösungen entwerfen, entwickeln, implementieren oder Datenbanklösungen verwenden. Zu den Mitgliedern der Community gehören:

- Anwendungsentwickler, welche eine offene Standarddatenbank für die Entwicklung von Einzelplatz, Client-Server, webbasierten und unternehmensweiten Anwendungen benötigen
- Unabhängige Softwareanbieter (ISVs), Hardwareanbieter, Infrastruktur Speicheranbieter und andere Anbieter von Lösungen, welche einen voll funktionsfähigen Datenserver mit ihren Lösungen bündeln oder einbetten wollen
- Berater, Datenbankadministratoren und IT Architekten, welche einen stabilen Datenserver für Training, Ausbau des fachlichen Könnens, Bewertung und Prototyping benötigen
- Neugründungen, kleine und mittelständische Unternehmen, welche einen zuverlässigen Datenserver für ihre Anwendungen und Geschäftsbereiche benötigen
- Datenbank Bastler und Enthusiasten mit einer Vorreiterrolle, welche einen einfach zu verwendenden Datenserver für die Entwicklung von Web 2.0 und zukunftsweisenden Anwendungen benötigen
- Studenten, Lehrer und andere akademische Enthusiasten, welche einen höchst vielseitigen Datenserver für den Unterricht, Kursmaterial, Projekte und Forschung benötigen

DB2 Express-C besitzt die gleiche Grundfunktionalität und Codebasis wie die anderen kostenpflichtigen DB2 Editionen für Linux, UNIX und Windows. DB2 Express-C ist unter Linux oder Windows sowohl auf 32-Bit als auch auf 64-Bit Systemen lauffähig. Es ist auf Systemen mit einer beliebigen Anzahl Prozessoren oder Speichermenge lauffähig und benötigt keinen speziellen Speicher oder Systeminstallationsvoraussetzungen. DB2 Express-C enthält außerdem pureXML ohne Zusatzkosten. Bei pureXML handelt es sich um DB2's einzigartige Technologie zur nativen Speicherung und Verarbeitung von XML Dokumenten.

1.1 Frei für Entwicklung, Implementierung und Verteilung... grenzenlos!

Dieser Satz summiert die Schlüsselideale hinter DB2 Express-C:

- **Frei zur Entwicklung:** Falls Sie ein Anwendungsentwickler sind und eine Datenbank für ihre Anwendung benötigen können sie DB2 Express-C verwenden.
- **Frei zur Implementierung:** Falls Sie in einer Produktionsumgebung arbeiten und ein Datenmanagementsystem für die Speicherung ihrer wichtigen Daten benötigen können sie DB2 Express-C verwenden.
- **Frei zur Verteilung:** Falls Sie eine Anwendung oder ein Dienstprogramm entwickeln, welches einen eingebetteten Datenserver erfordert können sie DB2 Express-C einschließen. Obwohl DB2 Express-C in Ihre Anwendung eingebettet ist und bei jedem Verkauf ihrer Anwendung mitgeliefert wird ist es immer noch kostenlos. Sie müssen sich bei IBM registrieren um DB2 Express-C weitergeben zu dürfen, wie auch immer, diese Registrierung ist ebenfalls kostenlos.
- **Keine Grenzen:** Während andere konkurrierende Datenbankangebote die Datenbankgröße, Anzahl der Datenbanken und Benutzer begrenzen, besitzt DB2 Express-C KEINE Datenlimitierung. Ihre Datenbank kann, ohne die Lizenzbestimmungen zu verletzen, wachsen und wachsen. Es gibt ebenfalls keine Begrenzung der Anzahl Verbindungen oder Benutzer je Server.

Hinweis:

Um mehr über DB2 Express-C und seine Rolle in der Information On Demand Welt und Web 2.0 zu erfahren, können Sie sich diese Videopräsentation ansehen:

<http://www.channeldb2.com/video/video/show?id=807741:Video:3922>

1.2 Benutzerunterstützung und technischer Support

Wenn Sie technische Fragen zu DB2 Express-C haben, können sie diese im DB2 Express-C Forum stellen. Dieses kostenlose Forum wird von einem engagierten DB2 Express-C Team betreut, letztlich ist aber die Community ist, welche die meisten der Antworten auf freiwilliger Basis liefert.

IBM gibt Benutzern auch die Möglichkeit zu geringen Kosten eine jährliche Subskription (auch bekannt als 12 Monatslizenz und Subskription oder Lizenz mit einer Festlaufzeit (Fixed Term License (FTL)) zu erwerben. Diese DB2 Express-C Subskription schließt IBM Unterstützung durch 24 * 7 technischen Support und Software Updates ein. Zusätzlich zu Support und Software Wartung, erhalten Sie mit den geringen jährlichen Subskriptionskosten (\$2,995 pro Server und Jahr in den Vereinigten Staaten – kann in anderen Ländern abweichen) zusätzlich zwei Schlüsselfunktionen: HADR (Clustering für Hochverfügbarkeit und Ausfallwiederherstellung), sowie SQL Replikation (um Daten mit anderen DB2 Servern zu replizieren). Weitere Informationen über die DB2 Express-C Subskription finden Sie unter www.ibm.com/db2/express/support.html.

1.3 DB2 Server

Alle DB2 Server enthalten dieselben Kernkomponenten, welche so verpackt sind, dass Benutzer die benötigte Funktionalität zum richtigen Preis wählen können. Abbildung 1.1 illustriert die unterschiedlichen DB2 Produkteditionen.

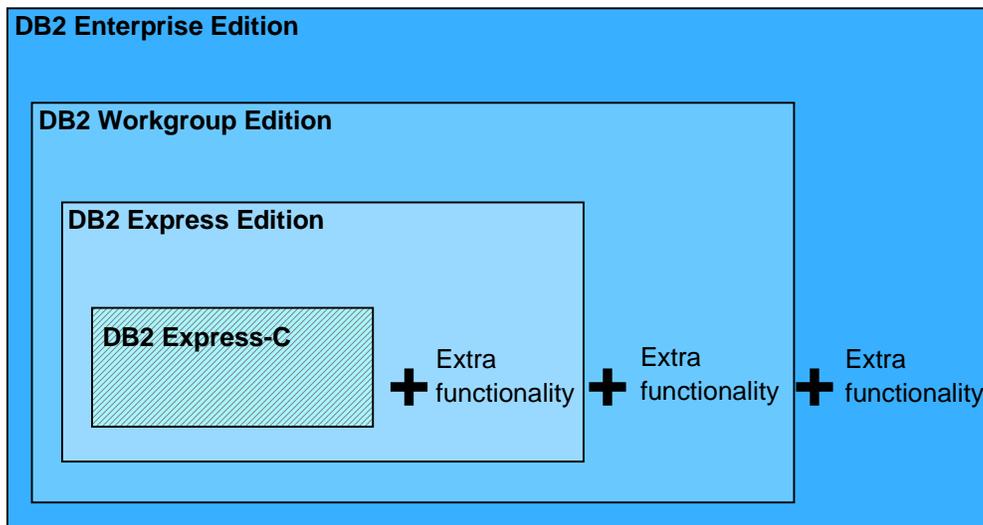


Abbildung 1.1 – DB2 Server

Wie aus Abbildung 1.1 hervorgeht ist DB2 Express-C, bis auf wenige Komponenten, mit DB2 Express identisch. DB2 Express-C wird der Community kostenlos zur Verfügung gestellt. Wie bereits zuvor erwähnt, ist technische Unterstützung über das Online Forum verfügbar oder Sie können durch den Erwerb der 12monatigen Subskriptionslizenz offiziellen 24 * 7 IBM DB2 technischen Support erhalten.

Abbildung 1.1 erklärt ebenfalls warum es so einfach ist ein Upgrade von DB2 Express-C durchzuführen. Falls Sie in Zukunft wünschen auf einen anderen DB2 Server zu aktualisieren, besitzen alle DB2 Server identische Kernkomponenten. Dies bedeutet ebenfalls, dass jedwede für eine Edition entwickelte Anwendung ohne Änderung unter einer anderen Edition läuft. Und alle unter einer Edition erlernten Fertigkeiten können auf andere Editionen angewandt werden.

1.4 DB2 Clients und Treiber

Ein DB2 Client enthält die erforderliche Funktionalität um eine Verbindung zu einem DB2 Server herzustellen; wie dem auch sei, ein DB2 Client muss nicht immer installiert werden. Eine JDBC Typ 4 Anwendung zum Beispiel kann sich direkt mit einem DB2 Server verbinden. DB2 Clients und Treiber sind in verschiedenen Ausführungen verfügbar:

- IBM Data Server Client: die umfangreichste, enthält grafische Dienstprogramme, Treiber.

- IBM Data Server Runtime Client: ein kleiner Client mit Basisfunktionalität, Treiber enthaltend
- DB2 Runtime Client Merge Modules for Windows: wird hauptsächlich verwendet um einen DB2 Runtime Client als Bestandteil einer Windows Anwendungsinstallation einzubetten
- IBM Data Server Driver for JDBC and SQLJ: erlaubt Java Anwendungen die direkte Verbindung zu DB2 Servern
- IBM Data Server Driver for ODBC and CLI: erlaubt ODBC und CLI Anwendungen die Verbindung zu einem DB2 Server ohne den Overhead der Installation eines Clients
- IBM Data Server Driver for ODBC, CLI and .NET: ein Windows spezifischer Treiber mit Unterstützung für .NET, zusätzlich zu ODBC und CLI

Abbildung 1.2 zeigt die verschiedenen verfügbaren DB2 Clients und Treiber.

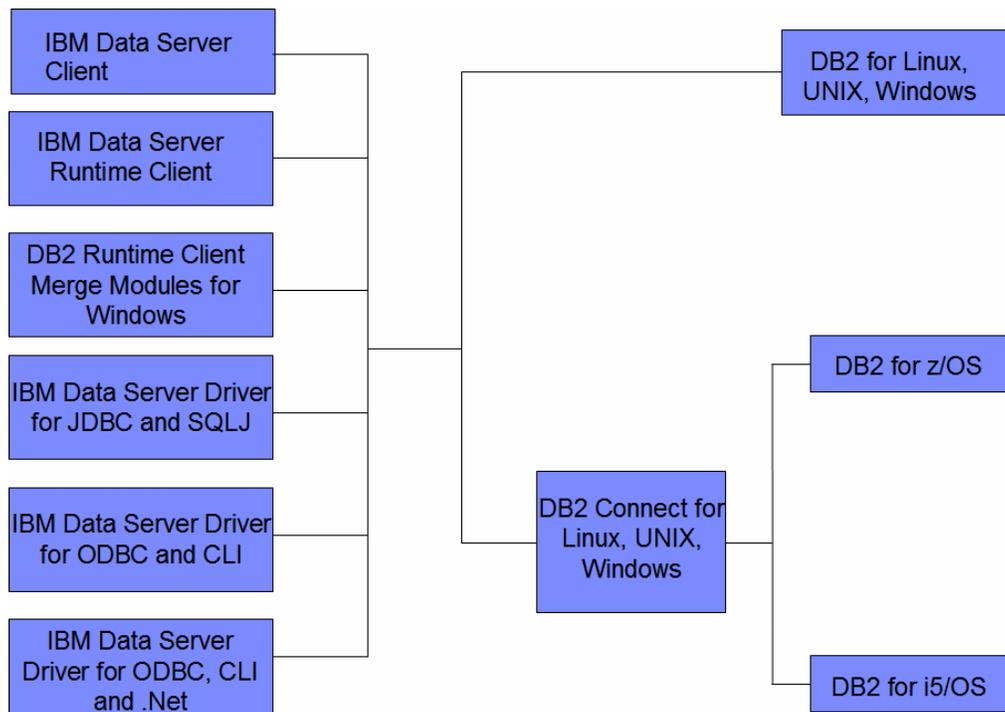


Abbildung 1.2 – DB2 Clients und Treiber

Auf der linken Seite der Abbildung 1.2 sind alle DB2 Clients und Treiber abgebildet. Obwohl alle DB2 Clients die erforderlichen Treiber beinhalten werden, beginnend mit DB2 9, die individuellen Treiber ebenfalls zu Verfügung gestellt. Alle DB2 Clients und Treiber sind kostenlos und stehen auf der DB2 Express-C Website zum Download zu Verfügung. Die Clients und Treiber können verwendet werden um eine Verbindung zu einem DB2 Server unter Linux, UNIX oder Windows herzustellen. Um eine Verbindung zu einem DB2 Server unter z/OS® oder DB2 for i5/OS® Server herzustellen müssen Sie über einen DB2

Verbindungsserver (Connect Server™, wie in der Mitte der Abbildung 1.2 gezeigt) gehen. Wir werden die DB2 Connect Software in Kapitel 2 diskutieren.

1.5 Unabhängig in der Anwendungsentwicklung

DB2 bietet eine Anwendungsentwicklungsumgebung, welche standardbasiert und transparent über die DB2 Familie ist. SQL Standardisierung über die DB2 Produktlinie bietet einen einheitlichen Satz von Datenzugriffsschnittstellen für die Anwendungsprogrammierung. Zusätzlich bietet jedes DB2 Produkt SQL Pre-Compiler, welche es Entwicklern erlauben statisches und dynamisches SQL in portablen Anwendungsprogrammen einzubetten. DB2 besitzt sogar einen nativen .NET verwalteten Anbieter und Integration in die Microsoft® Visual Studio Tools.

Sprachen und Normen mit welchen DB2 verwendet werden kann beinhalten:

- SQL, XQuery, XPath
- C/C++ (CLI, ODBC und eingebettetes SQL)
- Java (JDBC und SQLJ)
- COBOL
- PHP
- Perl
- Python
- Ruby on Rails
- .NET Sprachen
- OLE-DB
- ADO
- MS Office: Excel, Access, Word
- Web Dienste

1.6 DB2 Versionen im Vergleich zu DB2 Editionen

Falls Sie ein DB2 Neuling sind könnte sie die Unterscheidung zwischen einer DB2 Version und einer DB2 Edition verwirren.

Alle paar Jahre veröffentlicht IBM öffentlich eine neue DB2 Version. Eine Version enthält neue Funktionalitäten und signifikante Verbesserungen des Produkts. Zurzeit werden DB2 Version 8 und Version 9 offiziell von IBM unterstützt. Eine Version besitzt unter Umständen mehrere Ausgaben (Releases), welche einige neue Funktionalitäten enthalten können, jedoch normalerweise sind diese nicht signifikant genug eine neue Version zu rechtfertigen. So sind zum Beispiel 8.1 und 8.2 Release Level der DB2 Version 8. Wenn man die Vergangenheit betrachtet, sieht es so aus als ob IBM annähernd jedes Jahr eine neue DB2 Ausgabe freigibt, wie dem auch sei liegen zwischen neuen Versionen 2 bis 3 Jahre. Die aktuellste Ausgabe ist V9.5 (zuvor mit dem Codenamen DB2 'Viper 2' bezeichnet) und wurde im Oktober 2007 allgemein verfügbar (Generally Available (GA)) gemacht. Jede Ausgabe kann mehrere Änderungsstufen (Modification Level) beinhalten, welche typischerweise Fehlerbereinigungen beinhalten oder mit Fix Pack Leveln korrespondieren und nur selten neue Funktionalitäten beinhalten. Während dies geschrieben wird, ist die aktuellste Version, Release, Modification (V, R, M) Level von DB2

Express-C 9.5.0, welches mit einem Codelevel von 9.5 mit Fix Pack 0 korrespondiert, und was bedeutet, dass es sich um einen GA Level handelt.

Auf der anderen Seite sind Editionen ausgewählte Angebote oder Paketbündelungen innerhalb jeder Version. Wie zuvor diskutiert, handelt es sich bei einer Edition um ein Paket unterschiedlicher Funktionalität zu einem bestimmten Preis und einer Lizenz. DB2 Version 9.5 (auch bekannt als DB2 9.5) besitzt mehrere Editionen; wie zum Beispiel DB2 Express-C 9.5, DB2 Express 9.5, DB2 Workgroup 9.5, und DB2 Enterprise 9.5 (siehe Abbildung 1.1).

1.7 Wechsel zu einer höheren DB2 Edition

Wenn Ihre Datenbankanforderungen steigen möchten sie eventuell auf eine andere DB2 Edition aktualisieren, welche eine umfangreichere Hardwarekonfiguration unterstützt. Falls diese Situation auftritt, ist es einfach auf eine andere DB2 Edition zu aktualisieren:

- Falls Sie ein Upgrade auf eine andere DB2 Edition auf demselben Computersystem vornehmen, installieren sie die neue DB2 Edition über DB2 Express-C und die korrespondierende neue Lizenz. Ihre Datenbanken werden nicht gelöscht (eine Sicherung ist immer empfohlen).
- Falls Sie ein DB2 Upgrade durchführen bei welcher die neue Edition auf einem anderen größeren Computer mit dem gleichen Betriebssystem installiert wird, sichern sie ihre Datenbanken auf dem kleineren Computer, kopieren die Backup Images auf den größeren Computer und stellen die Datenbanken auf dem größeren Computer aus den Backup Images wieder her. Eventuell müssen Sie auch die Konfigurationseinstellungen der Instanz (DBM_CFG) sichern und auf den größeren Computer anwenden. Die Sicherungs- und Wiederherstellungsbefehle werden ausführlicher Kapitel 11, Sicherung und Wiederherstellung, besprochen. Die DBM CFG wird ausführlicher in Kapitel 5, Die DB2 Umgebung, besprochen.
- In beiden Fällen muss Ihre Anwendung nicht modifiziert werden.

1.8 Wartung von DB2 Express-C

Wie bereits zuvor besprochen gibt es zwei Supportoptionen für DB2 Express-C:

1. Kauf der 12monatigen Subskriptionslizenz. Diese bietet Ihnen 'Rund um die Uhr' Abdeckung durch den IBM DB2 technischen Support und gibt ihnen die Möglichkeit DB2 Software Updates (auch Fix Packs genannt) zu installieren.
2. Verwenden des Online DB2 Express-C Community Forums. Dies ist vollständig kostenlos, bietet jedoch keinen offiziellen Support durch IBM. Zusätzlich ist IBM bei dieser Option nicht verpflichtet neue Funktionalitäten und Fehlerbereinigungen zu einem bestimmten Zeitpunkt zur Verfügung zu stellen. Das Konzept eines Fix Packs, welches in Kapitel 2 besprochen wird, trifft ebenfalls nicht zu, da stattdessen Aktualisierungen des gesamten DB2 Images von Zeit zu Zeit verfügbar gemacht werden. Wenn neue Releases erscheinen können Sie normalerweise aktualisierte DB2 Express-C Images statt der alten erwarten.

1.9 Verwandte freie Software

Alle auf der DB2 Express-C Downloadseite (www.ibm.com/db2/express/download.html) zum Download angebotene Software ist kostenlos. Neben den Images für DB2 Express-C (für Linux und Windows, beide 32 und 64-Bit Architekturen), kann andere nützliche Software heruntergeladen und kostenlos verwendet werden:

- IBM Data Studio
- DB2 Net Search Extender
- DB2 Spatial Extender

Außerdem gibt es zusätzliche, Sie eventuell interessierende, auf DB2 Express-C basierende Starter Toolkits, welche über die IBM Alphaworks Website (www.alphaworks.ibm.com/datamgmt) heruntergeladen werden können:

- Starter Toolkit for DB2 on Rails
- Web 2.0 Starter Toolkit for DB2

Falls Sie einen leichtgewichtigen kostenlosen, von IBM angebotenen, Anwendungsserver suchen:

- WebSphere® Application Server – Community Edition (WAS CE)

1.9.1 IBM Data Studio

Beim IBM Data Studio handelt es sich um ein Eclipse basiertes Tool, welches es Ihnen erlaubt, ihre Daten, Datenbanken und Datenbank Anwendungen über den gesamten Lebenszyklus zu entwerfen, entwickeln, implementieren und zu verwalten. Data Studio ersetzt die zuvor erhältliche DB2 Developer Workbench 9.1 Lösung.

IBM Data Studio hilft Ihnen bei der Entwicklung von benutzerdefinierten Funktionen, gespeicherten Prozeduren, XQuery, SQL Anweisungen und enthält einen integrierten Debugger. Zusätzlich erlaubt Ihnen Data Studio die Arbeit mit Physical Data Modeling Diagrammen um die Entitätsbeziehungen zwischen Tabellen zu verstehen. Es kann Sie außerdem, ohne Programmierung, bei der Entwicklung und Veröffentlichung von Daten als Web Dienst unterstützen. Wir behandeln das Data Studio in Kapitel 14, SQL PL gespeicherte Prozeduren.

1.9.2 DB2 Net Search Extender

Mit dem DB2 Net Search Extender können Sie schnelle und detaillierte Volltextsuchen in Textdokumenten, einschließlich beliebiger nativ in DB2 gespeicherter XML Dokumente, ausführen.

1.9.3 [Starter Toolkit for DB2 on Rails](#)

Beim Starter Toolkit for DB2 on Rails handelt es sich um einen zweckdienlich verpackten Satz von Produkten und Technologien, welche die schnelle Erstellung einer Umgebung zur Erzeugung von Ruby on Rails verwendenden Webanwendungen ermöglicht. Die gesamte benötigte Software ist enthalten: DB2 Express-C; DB2 Treiber für Ruby; DB2 Adapter für

Rails; zusammen mit Anleitungen und anderen Lernmaterialien. Wir werden Ruby on Rails in Kapitel 17, Entwicklung in Java, PHP und Ruby, weiter besprechen.

1.9.4 [Web 2.0 Starter Toolkit for DB2](#)

Das Web 2.0 Starter Toolkit for DB2 vereinfacht den Einstieg in DB2, PHP und Dojo. Es hilft Ihnen die benötigte Software zu implementieren, verbindet sie mit Lerneinheiten und enthält Demoanwendungen. Bei zwei der Demoanwendungen handelt es sich um das Atom Feed Control Panel, welches Atom Feeds aus DB2 Tabellen generiert und das Web Services Control Panel, welches REST Web Service Wrapper um ihre DB2 Tabellen erstellt. Beide sind auf Grund der signifikanten Ajax und Widget Möglichkeiten, auf Dojo angewiesen.

1.9.5 WebSphere Application Server – Community Edition

Beim kostenlos verfügbaren IBM WebSphere Application Server Community Edition handelt es sich um einen einfachen Java EE 5 Anwendungsserver. Aufbauend auf der Apache Geronimo Technologie, nutzt er die neuesten Innovationen der Open-Source Community um eine integrierte, leicht zugängliche und flexible Grundlage für die Entwicklung und die Implementierung von Java Anwendungen zu Verfügung zu stellen. Optionaler technischer Support für WAS CE ist über eine jährliche Subskription verfügbar.

2

Kapitel 2 – Verwandte Funktionalitäten und Produkte

Dieses Kapitel beschreibt die DB2 Funktionalitäten, welche bei Erwerb der 12monatigen DB2 Express-C Lizenz verfügbar sind. Es beschreibt auch die in anderen Editionen, in einigen Fällen mit Zusatzkosten verbundenen, enthaltenen Funktionalitäten.

Mit der kostenlosen DB2 Express-C Edition zur Verfügung gestellten Funktionalitäten:

- DB2 Kernfunktionalität
- Steuerzentrale, Data Studio und andere Verwaltungstools
- pureXML
- Verwendung von bis zu 2 GB und 2 Prozessorkernen
- Verfügbar auf Linux, Windows und Solaris (x86)

Funktionalitäten, welche nicht in der kostenlosen Edition von DB2 Express-C, jedoch in der DB2 Express-C 12monatigen Subskriptionslizenz enthalten sind:

- Fix Packs
- Hochverfügbarkeit (High Availability)
- Datenreplikation (Homogenes SQL)
- Verwendung von bis zu 4 GB und 4 Prozessorkernen (auf 2 Sockeln)

In der folgenden Tabelle finden Sie eine Auflistung der Produktfunktionalitäten und deren Verfügbarkeit in den verschiedenen DB2 Editionen. Produktfunktionalitäten, welche Sie separat erwerben können sind bei den zutreffenden DB2 Edition mit Namen aufgeführt.

Funktionalität	Express-C Lizenz mit Festlaufzeit	DB2 Express Edition	DB2 Workgroup Server Edition	DB2 Enterprise Server Edition
Homogenous SQL Replication	Ja	Ja	Ja	Ja
Net Search Extender	Ja	Ja	Ja	Ja
Spatial Extender	Ja	Ja	Ja	Ja

Funktionalität	Express-C Lizenz mit Festlaufzeit	DB2 Express Edition	DB2 Workgroup Server Edition	DB2 Enterprise Server Edition
pureXML™ technology	Ja	pureXML Feature	pureXML Feature	pureXML Feature
High availability disaster recovery	Ja	High Availability Feature	Ja	Ja
Tivoli® System Automation	Ja	High Availability Feature	Ja	Ja
Advanced Copy Services	Nein	High Availability Feature	Ja	Ja
Online reorganization	Nein	High Availability Feature	Ja	Ja
Homogenous Federation	Nein	Homogeneous Federation Feature	Homogeneous Federation Feature	Homogeneous Federation Feature
MQT	Nein	Nein	Query Optimization Feature	Ja
MDC	Nein	Nein	Query Optimization Feature	Ja
Query parallelism	Nein	Nein	Query Optimization Feature	Ja
Connection concentrator	Nein	Nein	Nein	Ja
Table partitioning	Nein	Nein	Nein	Ja
DB2 Governor	Nein	Nein	Nein	Ja
Compression:	Nein	Nein	Nein	Storage

Funktionalität	Express-C Lizenz mit Festlaufzeit	DB2 Express Edition	DB2 Workgroup Server Edition	DB2 Enterprise Server Edition
Row Level				Optimization Feature
Compression: Backup	Nein	Nein	Nein	Storage Optimization Feature
Label-based access control (LBAC)	Nein	Nein	Nein	Advanced Access Control Feature
Geodetic Extender	Nein	Nein	Nein	Geodetic Data Management Feature
Query Patroller	Nein	Nein	Nein	Performance Optimization Feature
DB2 workload management	Nein	Nein	Nein	Performance Optimization Feature
Performance Expert	Nein	Nein	Nein	Performance Optimization Feature
Homogenous Q Replication	Nein	Nein	Nein	Homogeneous Replication Feature
Database Partitioning	Nein	Nein	Nein	Nein

Tabelle 2.1: DB2 Version 9.5 Editionen: Funktionalitäten und Funktionssupport

Mit anderen DB2 Editionen zur Verfügung stehende Funktionalitäten sind:

Kostenpflichtige DB2 Express Edition Funktionalitäten:

- pureXML

- Hochverfügbarkeit (High Availability)
- Homogene Föderierungseigenschaft (Homogenous Federation Feature)

In der DB2 Workgroup Edition ohne Zusatzkosten enthaltene Funktionalitäten:

- High Availability
- Verfügbarkeit auf AIX, Solaris und HP-UX, zusätzlich zu Linux und Windows

Kostenpflichtige DB2 Workgroup Edition Funktionalitäten:

- pureXML
- Abfrageoptimierung (Query Optimization Feature - MQT, MDC, query parallelism)
- Homogene Föderierungseigenschaft (Homogenous Federation Feature)

Die DB2 Enterprise Edition enthält ohne Zusatzkosten zusätzliche Funktionalität, wie:

- Tabellen Partitionierung (Range partitioning)
- Materialisierte Abfragetabellen (MQT - Materialized Query Tables)
- Multidimensionale Aufteilung der Tabellen (MDC - Multidimensionales Clustering)
- Hochverfügbarkeitsfunktionen (HADR - HighAvailability and Disaster Recovery , TSA- Tivoli® Systemautomatisierung)
- Verbindungsoptimierung (Connection Concentrator)

Kostenpflichtige DB2 Enterprise Edition Funktionalitäten:

- pureXML
- Speicheroptimierungsmöglichkeit (Storage Optimization Feature - enthält Zeilenkompression)
- Erweiterte Zugriffskontrolle (Advanced Access Control – granulare und erweiterte Sicherheit)
- Leistungsoptimierung (Performance Optimization Feature – enthält Workload Management, Performance Expert, Query Patroller)
- geographische Ortsanalyse (Geodetic Extender)
- Homogene Föderierungseigenschaft (Homogenous Federation Feature - Zugriff auf entfernte DB2 und Informix® Daten als lokale Tabellen)

Kostenpflichtige auf DB2 bezogene Produkte:

- DB2 Connect
- DB2 Warehouse Editionen
- WebSphere® Federation Server
- WebSphere Replication Server

2.1 In der DB2 Express-C Subskription enthaltene Funktionalitäten

Dieser Abschnitt umreißt DB2 Fix Packs, HADR und SQL Replikation.

2.1.1 Fix Packs

Ein DB2 Fix Pack besteht aus einem Satz von auf einem installierten DB2 Produkt angewandten Codeverbesserungen, welche der Behebung verschiedener nach dem Release des Produktes gemeldeter Fehler dienen. Mit einer installierten Subskriptionslizenz können Fix Packs kostenlos heruntergeladen und installiert werden. Sie werden typischerweise alle 3 Monate verfügbar gemacht.

Um das aktuellste Fix Pack herunterzuladen, prüfen Sie die DB2 Technische Support Site unter http://www.ibm.com/software/data/db2/support/db2_9/.

2.1.2 Hochverfügbarkeit und Ausfallwiederherstellung (HADR)

Bei der Hochverfügbarkeit und Ausfallwiederherstellung (HADR - High Availability Disaster Recovery) handelt es sich um Datenbankzuverlässigkeitsmerkmal, welches eine hohe Verfügbarkeit und Ausfallwiederherstellung für vollständige und partielle Site-Failures. Eine HADR Umgebung besteht im Allgemeinen aus zwei Datenservern, dem primären und dem sekundären (welche in geographisch unterschiedlichen Orten stehen können). Beim primären Server handelt es sich um den Server auf welchem sich die Quelldatenbank befindet und der Zugriff durch Clientanwendungen erfolgt. Wenn Transaktionen auf der primären Datenbank verarbeitet werden, werden Datenbankprotokolle automatisch über das Netzwerk zum sekundären Server gesandt. Auf dem sekundären Server befindet sich eine geklonte Kopie der primären Datenbank, was normalerweise dadurch erreicht wird, indem die primäre Datenbank gesichert und auf dem sekundären System wiederhergestellt wird. Wenn die primären Datenbankprotokolle empfangen werden, werden diese wieder eingespielt und auf die sekundäre Datenbank angewandt. Durch kontinuierliches Zurückspielen der Protokollsätze erhält die sekundäre Datenbank ein synchronisiertes Replikat der primären Datenbank und kann, falls die primäre Datenbank ausfällt, übernehmen.

Eine vollständig DB2-unterstützte HADR Lösung bietet Ihnen:

- Blitzschnelle Ausfallsicherungsfähigkeit, vollständig transparent für Kunden und Client Anwendungen
- Vollständige Atomarität von Transaktionen zur Vermeidung von Datenverlust
- Die Fähigkeit Systeme oder Anwendungen ohne sichtbare Störung der Dienste zu aktualisieren
- Remote Ausfallsicherheit bietet vollständige Wiederherstellung bei einem lokalen das Data Center treffenden Unglück
- Leichte Verwaltung mit DB2 grafischen Dienstprogrammen
- Dies alles mit vernachlässigbarer Auswirkung auf die gesamte Systemperformance

Hinweis:

Um sich eine Demonstration der HADR Funktionsweise anzusehen, besuchen Sie bitte diese Site:

<http://www.channeldb2.com/video/video/show?id=807741:Video:3902>

2.1.3 Datenreplikation

Diese Funktionalität erlaubt die Replikation von Daten zwischen einem Quellserver, auf welchem Datenänderungen vorgenommen werden, und einem Zielserver, auf welchem Datenänderungen angewandt werden. Abbildung 2.1 bietet einen Überblick über die Funktionsweise der Replikation.

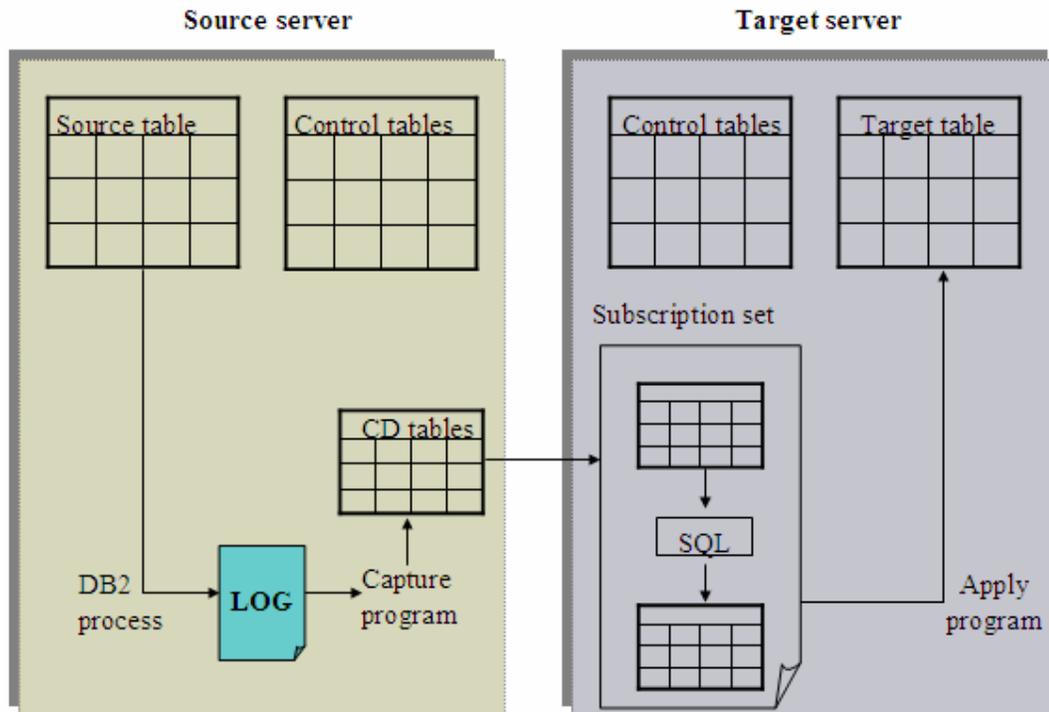


Abbildung 2.1 –SQL Replikation

In Abbildung 2.1 befinden sich zwei Server, ein Quellserver (Source Server) und ein Zielserver (Target Server). Auf dem Quellserver erfasst ein Erfassungsprogramm Änderungen an der Datenbank. Auf dem Zielserver wendet ein Apply Programm die Änderungen auf das Datenbankreplikat an. Replikation ist für verschiedene Datenreplikation benötigende Zwecke sinnvoll, einschließlich Kapazitätsentlastung, Beschickung von Data Warehouses und Data Marts; sowie die Revision der Änderungshistorie. Bei Verwendung der SQL Replikationsfunktion können Sie Daten zwischen DB2 Express-C und anderen DB2 Servern, einschließlich jenen auf Linux, UNIX, z/OS und i5/OS Systemen replizieren.

2.2 Mit DB2 Express-C nicht verfügbare Funktionalitäten

Dieser Abschnitt beschreibt einige der Funktionalitäten, welche in anderen Editionen von DB2, nicht jedoch in DB2 Express-C, verfügbar sind.

2.2.1 Datenbankpartitionierung (DPF)

Die Funktionalität Datenbankpartitionierung (DPF - Database Partitioning Feature) ist nur mit den DB2 Warehouse Editionen verfügbar. Es gestattet, dass Datenbanken über mehrere Partitionen, welche auf mehreren Computern residieren können, verteilt sind. DPF basiert auf einer Shared-Nothing Architektur.

Jeder Computer, sobald dieser zur Partitionsgruppe hinzugefügt wird, bringt zusätzliche Datenverarbeitungsleistung mit seiner eigenen CPU und Speicher. DPF ist besonders hilfreich in großen Datenserver Umgebungen, wie Data Warehouses, wo Abfragen von Entscheidungsunterstützungssystemen (DSS - Decision Support Systems) ausgeführt werden.

2.2.2 Verbindungsoptimierung (Connection Concentrator)

Die Funktionalität Verbindungsoptimierung (Connection Concentrator) unterstützt die Verbindung einer großen Anzahl gleichzeitiger Benutzer. Zuvor benötigte jede Datenbankverbindung einen Datenbankagenten. Der Connection Concentrator führt das Konzept eines "logischen Agenten" ein und erlaubt einem Agenten mehrere Verbindungen zu verwalten. Agenten werden ausführlicher in Kapitel 6, DB2 Architektur, behandelt.

2.2.3 Geodetic Extender

Der DB2 Geodetic Extender ist als kostenpflichtige Option für die DB2 Enterprise Edition erhältlich. Dieser Extender vereinfacht die Entwicklung von Geschäftsanalytik- (Business Intelligence) und e-Gouvernement Lösungen, welche geographische Ortsanalyse benötigen, erheblich. Der DB2 Geodetic Extender kann einen virtuellen Globus in jedem Maßstab konstruieren. Die meisten Ortsinformationen werden durch weltweite Systeme gesammelt, wie Global Positioning Satelliten (GPS) und können in Breitengrad/Längengrad Koordinaten (geocode) dargestellt werden. Geschäftsdaten, wie zum Beispiel Adressen, können vom DB2 Geodetic Extender zu einem geocode konvertiert werden und Firmenanwendungen arbeiten besser wenn diese die Daten in dieser nichtprojizierten Form speichern und die Kartenprojektionen (Erde nach flache Karte) belassen wo sie hingehören: in die Präsentationsschicht für Ausgabe und Druck der Karten.

2.2.4 Workload Management (WLM)

Verwaltet Auslastungen über eine Datenbank basierend auf Benutzer- und Anwendungsprioritäten kombiniert mit Ressourcenverfügbarkeit und Auslastungsschwellwerten. Er gestattet Ihnen die Datenbankauslastung und Abfragen zu regulieren so dass wichtige und mit hoher Priorität auszuführende Abfragen sofort ausgeführt werden, so dass 'schelmische' Abfragen gehindert werden die Systemressourcen in Anspruch zu nehmen und sicherzustellen, dass ihre Systeme effektiv laufen. WLM ist neu in DB2 9.5 und bietet leistungsfähigere Einsatzmöglichkeiten als Query Patroller und die DB2 Governor Tools, welche mit den vorherigen DB2 Versionen verfügbar waren.

2.3 Kostenpflichtige, mit DB2 verwandte Produkte

2.3.1 DB2 Connect

Bei DB2 Connect handelt es sich um eine kostenpflichtige Software, welche einem DB2 für Linux, UNIX oder Windows Client die Verbindung zu einem DB2 für z/OS oder DB2 für i5/OS Server, wie in Abbildung 2.2 gezeigt, erlaubt. DB2 Connect wird nicht benötigt wenn die Verbindung in die andere Richtung erfolgt, d.h. Sie verbinden von DB2 für z/OS oder DB2 für i5/OS zu einem DB2 für Linux, UNIX oder Windows Server. DB2 Connect wird nicht benötigt wenn die Verbindung in die andere Richtung erfolgt, d.h. Sie verbinden von DB2 für z/OS oder DB2 für i5/OS zu einem DB2 für Linux, UNIX oder Windows Server. DB2 Connect ist, abhängig von Ihren Erfordernissen, in zwei Haupteditionen verfügbar: Eine DB2 Connect Personal Edition und eine DB2 Connect Enterprise Edition.

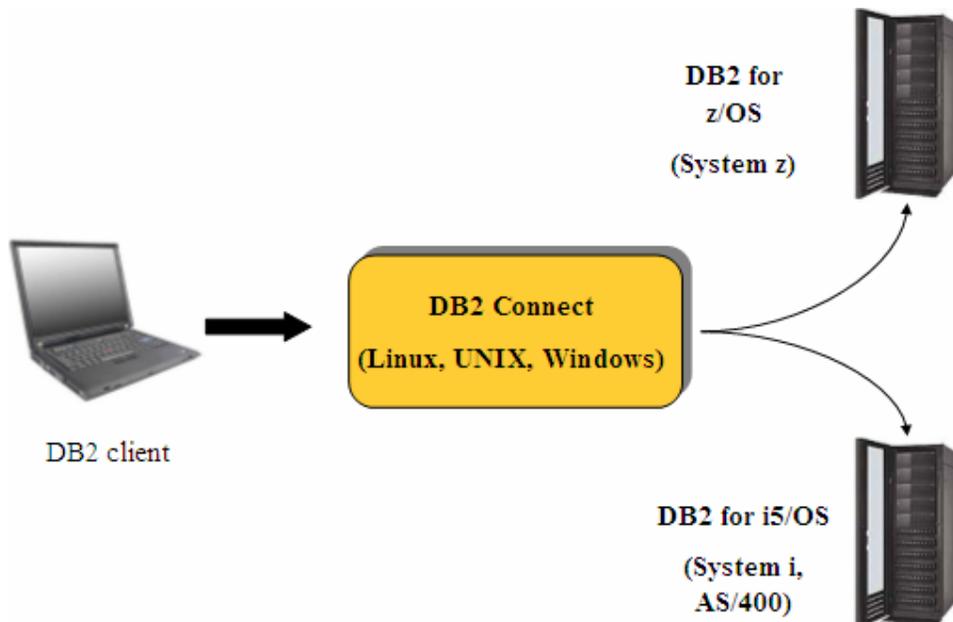


Abbildung 2.2 – DB2 Connect

2.3.2 WebSphere Federation Server

Früher bekannt als WebSphere Information Integrator (wegen der Föderationsunterstützung), gestattet der WebSphere Federation Server die Föderation von Datenbanken. Dies bedeutet dass Sie Datenbankabfragen, welche mit Objekten aus verschiedenen relationalen Datenbanksystemen arbeiten, ausführen können. Wenn Sie zum Beispiel den WebSphere Federation Server kaufen können sie die folgende Abfrage ausführen:

```

SELECT *
FROM   Oracle.Table1 A
       DB2.Table2 B
       SQLServer.Table3 C
WHERE  A.col1 < 100
       and B.col5 = 1000
       and C.col2 = 'Test'
    
```

Abbildung 2.3 gibt einen Überblick über die Einsatzmöglichkeiten des WebSphere Federation Servers.

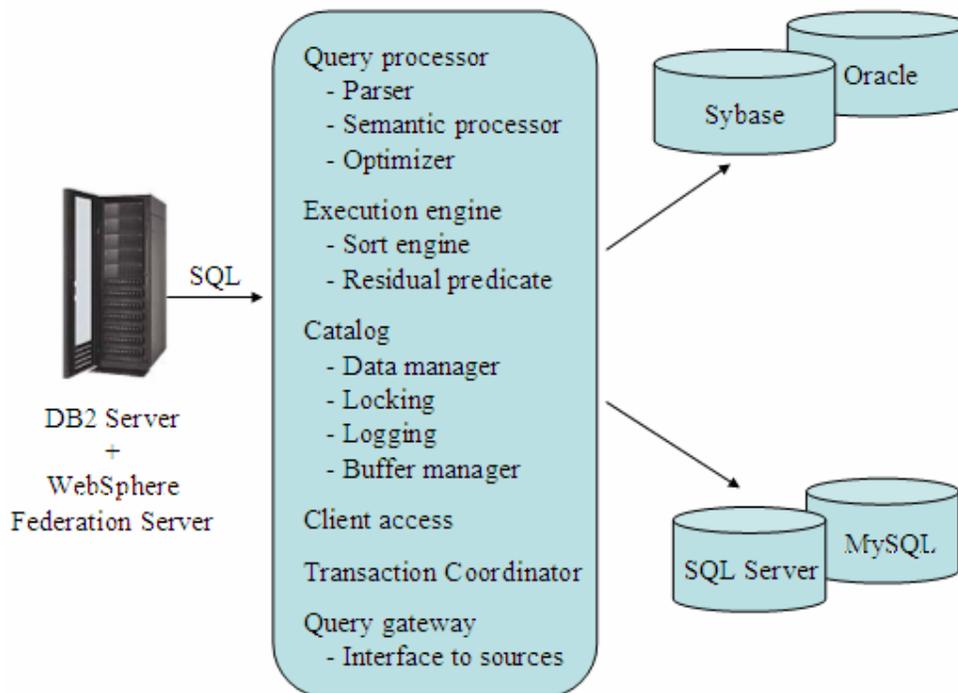


Abbildung 2.3 – WebSphere Federation Server

Für relationale Datenbankmanagementsysteme, welche Bestandteil der DB2 Familie sind, ist der Föderationsunterstützung (Federation Support) in DB2 Express-C eingebaut. Dies bedeutet, dass das WebSphere Federation Server Produkt nicht benötigt wird wenn Sie zum Beispiel eine Abfrage zwischen zwei unterschiedlichen DB2 Datenbanken ausführen wollen oder zwischen einer DB2 Datenbank und einer Informix® Datenbank (Informix ist Bestandteil der IBM Familie).

2.3.3 WebSphere Replication Server

Zuvor bekannt als WebSphere Information Integrator (zur Replikationsunterstützung), gestattet der WebSphere Replication Server die SQL Replikation von Datenbanksätzen wenn nicht-IBM Datenserver betroffen sind. Er enthält außerdem ein, als Q-Replication zur Datenreplikation unter Verwendung von Warteschlangen (Message Queues) bekanntes, Feature.

3

Kapitel 3 – DB2 Installation

Um die DB2 Express-C Edition unter entweder Linux oder Windows zu installieren, sollten Sie sicherstellen dass ihre Systeme die Installationsvoraussetzungen erfüllen.

3.1 Installationsvoraussetzungen

Unter Einbeziehung der Anforderungen an das Betriebssystem und den Level ist DB2 Express-C für Linux, Solaris (x64) und Windows 2000, 2003, XP und Vista verfügbar. Die verfügbaren Prozessorarchitekturen sind 32-Bit, 64-Bit und PowerPC (Linux). Falls es erforderlich ist DB2 auf einer anderen Plattform (wie UNIX) laufen zu lassen, sollten Sie eine der anderen, zuvor in diesem Buch beschrieben, Datenserver Editionen kaufen. Die Betriebssystemvoraussetzungen für alle DB2 Editionen sind in diesem Dokument beschrieben:

<http://www.ibm.com/software/data/db2/udb/sysreqs.html>

Bezogen auf Hardwareressourcen kann DB2 Express-C auf System mit einer beliebigen Anzahl von CPU Kernen und Speichermenge installiert werden. DB2 wird jedoch nur bis zu zwei Kerne und 2 GB Arbeitsspeicher in der kosten- und garantielosen Lizenzversion (DB2 Express-C) und bis zu 4 Kerne und 4 GB Arbeitsspeicher in der kostenpflichtigen Subskriptionsversion (DB2 Express) verwenden. Bei den Systemen kann es sich um physikalische Systeme oder durch Partitionierung erstellte virtuelle Systeme handeln bzw. den Betrieb in einer virtuellen Maschine. Falls Sie es vorziehen, kann selbstverständlich ein Betrieb auf einem kleineren System, wie zum Beispiel einem Einzelprozessorsystem mit 1 GB Arbeitsspeicher, erfolgen.

Für die neuesten Informationen bezüglich der DB2 Express-C Hardwareanforderungen konsultieren Sie diese DB2 Express-C Webseite:

<http://www-306.ibm.com/software/data/db2/express/getstarted.html>

3.2 Betriebssystemberechtigung für die Installation

Um DB2 Express-C unter Linux oder Windows zu installieren benötigen Sie die erforderliche Berechtigung.

Unter **Linux** benötigen Sie Root-Rechte (der Superuser) für die Installation von DB2 Express-C. Sie können DB2 Express-C auch als nicht-Root Benutzer installieren, werden jedoch insofern eingeschränkt sein, als das sie mit dem Produkt nicht alles machen können. So können Sie zum Beispiel unter einer nicht-Root Installation keine zusätzlichen Instanzen zusätzlich zur bereits während der Installation erstellten Instanz erstellen.

Unter **Windows** muss das Benutzerkonto auf der Maschine auf welcher die Installation vorgenommen werden soll zur Gruppe der Administratoren gehören. Alternativ kann ein nicht zur Gruppe der Administratoren gehörendes Konto verwendet werden, sofern ein Mitglied der Windows Administratorgruppe zuvor die Windows Rechteerhöhung (Elevation Privileges Settings) so konfiguriert, dass einem nicht zur Gruppe der Administratoren gehörenden Benutzerkonto die Durchführung einer Installation erlaubt wird.

Bei Windows Domänenkonten muss, zur Verifizierung der Benutzererkennung auf dem DB2 Server, die Installationsbenutzererkennung zur Gruppe der Domänenadministratoren auf der Domäne gehören auf welcher die Konten erstellt werden. Sie können auch das eingebaute lokale Systemkonto für den Start der Installation aller Produkte verwenden.

Das Benutzerkonto muss das Recht auf "Zugriff auf diesen Computer über das Netzwerk" besitzen.

Hinweis:

Eine Video Präsentation zur DB2 Express-C Installation können Sie sich unter diesem Link ansehen:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4442>

3.3 Installationsassistent

Obwohl es mehrere Möglichkeiten gibt DB2 zu installieren, handelt es sich bei dem auf eine grafische Oberfläche basierenden Installationsassistenten um die einfachste. Nach dem Download und dem Extrahieren des DB2 Express-C Images kann der die Installation handhabende Installationsassistent gestartet werden:

- Windows: Starten Sie die setup.exe Datei im Verzeichnis EXP/image
- Linux: Starten Sie den db2setup Befehl im Verzeichnis exp/disk1

DB2 Express-C ist, wenn Sie den Anweisungen des Installationsassistenten folgen, sehr leicht zu installieren. In den meisten Fällen sind die Standardeinstellungen ausreichend, so dass Sie nur noch die Lizenz akzeptieren müssen, mehrere Male auf die "Weiter" Schaltfläche klicken und dann auf die "Fertig stellen" Schaltfläche klicken. Nach ein paar Minuten ist Ihre Installation vollständig und DB2 wurde eingerichtet und gestartet!

Abbildung 3.1 zeigt das DB2 Setup Launchpad. Klicken Sie auf "Installieren eines Produkts" und wählen anschließend "Neue Installation", um eine neue DB2 Express-C Kopie auf ihrem System zu installieren.

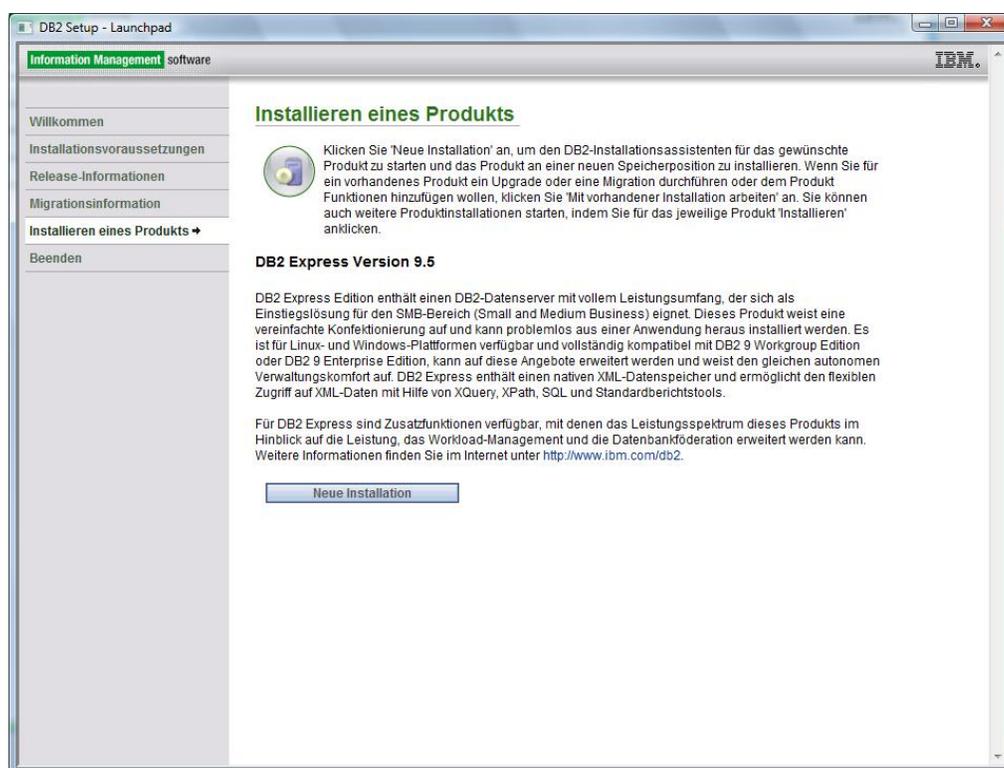
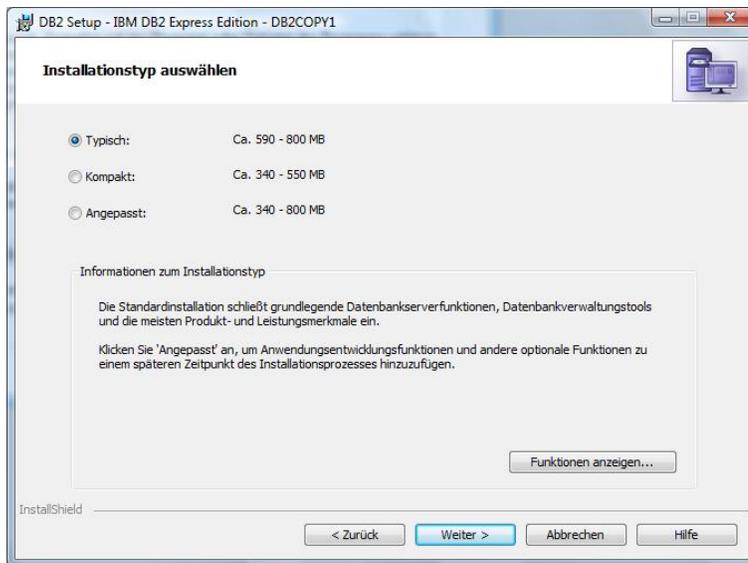


Abbildung 3.1 – Das DB2 Setup Launchpad

Nachdem die Lizenz akzeptiert wurde ist es meistens ausreichend den Installationstyp "Typisch" (Standardeinstellung), wie in Abbildung 3.2 gezeigt, zu wählen.

**Abbildung 3.2 – Installationstyp**

Wie aus Abbildung 3.3 hervorgeht, haben Sie die Auswahl zwischen der Installation des Produkts, die Erstellung einer Antwortdatei (response file), oder beides. Antwortdateien werden in Abschnitt 3.4, Silent Installation, besprochen. Durch Auswahl der Standardeinstellung (IBM DB2 Express Edition auf diesem Computer installieren und Einstellungen in einer Antwortdatei speichern) ist ausreichend.

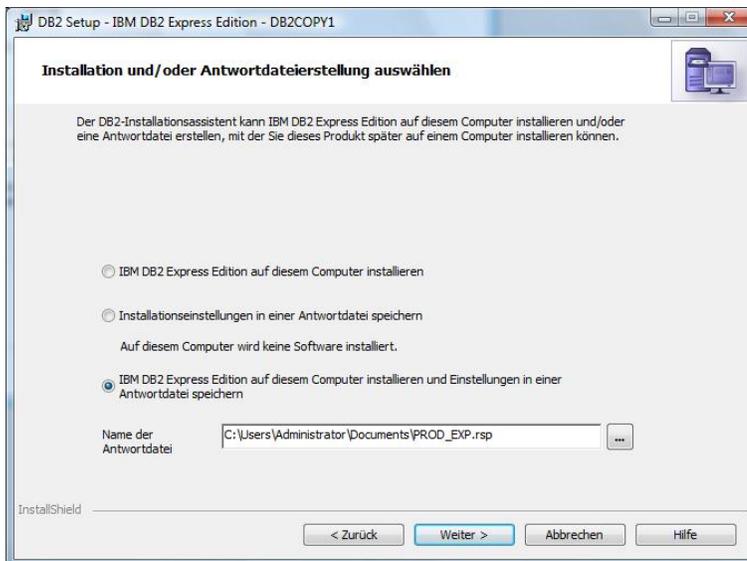


Abbildung 3.3 – Auswahl der Installation

Wählen Sie Standardwerte für die nächsten Dialoge. Wenn Sie den in Abbildung 3.4 gezeigten Dialog erreichen, können sie einen existierenden Benutzer angeben, welcher verwendet wird um mit der Instanz und anderen Diensten zu arbeiten. Dieser Benutzer muss unter Windows Mitglied der lokalen Administratorgruppe sein. Falls die Benutzererkennung nicht zu einem existierenden Benutzer gehört, wird dieser als lokaler Administrator angelegt. Das Feld Domäne können Sie, falls die Benutzererkennung keiner Domäne angehört, freilassen. Die unter Windows erstellte Standardbenutzererkennung lautet db2admin. Im Fall von Linux lautet die Standardbenutzererkennung db2inst1.



Abbildung 3.4 – Angabe der Benutzerinformation für die Standard DB2 Instanz

Abschließend zeigt der Installationsassistent, wie in Abbildung 3.5 gezeigt, eine Zusammenfassung über den Installationsumfang und die verschiedenen, in den vorherigen Schritten angegebenen, Konfigurationsoptionen. Wenn Sie auf die "Fertig stellen" Schaltfläche klicken startet die Installation und die Programmdateien werden auf Ihrem System gespeichert.

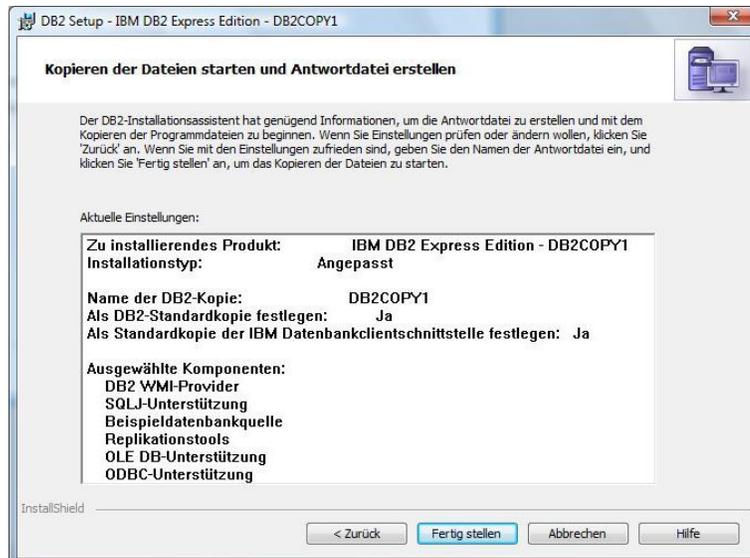


Abbildung 3.5 – Zusammenfassung der ausgewählten Komponenten

3.4 Silent Installation

Es gibt möglicherweise Situationen in welchen Sie einen DB2 Client auf mehreren Computern installieren, oder einen DB2 Datenserver als Bestandteil ihrer Anwendung einbetten, und es als Teil ihrer Anwendungsinstallation installieren. In derartigen Situationen ist eine sog. 'Silent Installation' der ideale Weg um DB2 zu installieren.

DB2 ermöglicht die Silent Installation durch die Verwendung von in Antwortdateien gespeicherten Installationsinformationen. Nachfolgend sehen Sie einen Ausschnitt aus einer Beispiellantwortdatei.

Codeausschnitt der während der Installation angelegten Beispiellantwortdatei:

```
PROD=UDB_EXPRESS_EDITION
LIC_AGREEMENT=ACCEPT
FILE=C:\Program Files\IBM\SQLLIB\
INSTALL_TYPE=TYPICAL
```

```
LANG=EN
```

```
INSTANCE=DB2
DB2.NAME=DB2
DEFAULT_INSTANCE=DB2
DB2.SVCENAME=db2c_DB2
DB2.DB2COMM=TCPIP
...
```

Es gibt mehrere Möglichkeiten der Erstellung einer Antwortdatei:

- ▶ Zuerst installieren Sie DB2 Express-C auf einem Computer unter Verwendung des Installationsassistenten. Eine der ersten Installationsoptionen erlaubt Ihnen durch Auswahl der Checkbox ihre Installationsantworten in einer Antwortdatei zu speichern. Bei Beendigung des Assistenten wird eine Antwortdatei in einem von Ihnen anzugebenden Verzeichnis und Dateinamen erstellt. Da es sich dabei um eine Textdatei handelt, können Sie diese in Anschluss bearbeiten. Dies wurde in Abbildung 3.3 gezeigt.
- ▶ Bearbeiten Sie die im DB2 Image enthaltene Beispielantwortdatei. Diese Beispieldatei (gekennzeichnet mit der Dateierweiterung `.rsp`) befindet sich im Verzeichnis `db2/platform/samples/`.
- ▶ Unter Windows können Sie auch das Programm `db2rspgn.exe` zur Generierung von Antwortdateien verwenden
`db2rspgn -d <output directory>`

Um dann unter Windows DB2 mittels einer Antwortdatei zu installieren führen Sie diesen Befehl aus:

```
setup -u <response filename>
```

Unter Linux führen Sie diesen Befehl aus:

```
db2setup -r <response filename>
```

Übung 1: Installieren von DB2 Express-C und Erstellen der SAMPLE Datenbank

Ziel

Bevor Sie mit dem Ausprobieren aller Funktionalitäten und Dienstprogramme beginnen, welche Bestandteile von DB2 Express-C sind, müssen sie es zuerst auf ihrem System installieren. In dieser Übung werden Sie eine einfache DB2 Express-C Installation unter Windows durchführen. Derselbe Installationsassistent steht auf Linux zur Verfügung; so dass die Schritte auf dieser Plattform fast analog sind.

Vorgehensweise

1. **Beschaffen des DB2 Express-C Images.** Laden Sie das zutreffende DB2 Express-C Image über das Internet oder bestellen sie die Discovery Kit DVD mit den Images über die DB2 Express-C Website (ibm.com/db2/express). Extrahieren Sie die Dateien in ein Verzeichnis ihrer Wahl.
2. **Dateien lokalisieren.** Navigieren zum Verzeichnis (oder Laufwerk) welches die extrahierten DB2 Produktinstallationsdateien enthält.
3. **Launchpad ausführen.** Starten Sie das DB2 Launchpad durch eine Doppelklick auf die Datei `setup.exe`. Unter Linux führen Sie den `db2setup` Befehl als Root aus. Im Launchpad klicken Sie auf die *Installieren eines Produkts* Option im linken Teilfenster.
4. **Ausführen des DB2 Installationsassistenten.** Der DB2 Installationsassistent prüft ob alle Systemvoraussetzungen erfüllt sind und schaut nach ob irgendwelche DB2 Installationen vorhanden sind. Klicken Sie auf die *Weiter* Schaltfläche um mit der Installation fortzufahren.
5. **Lesen der Lizenzvereinbarung.** Lesen und bestätigen Sie die Lizenzvereinbarung (wählen Sie das *„Ich akzeptiere...“* Optionsfeld) und klicken zum Fortsetzen auf die *Weiter* Schaltfläche.
6. **Installationstyp wählen.** Für diese Übung wählen Sie die Option *Typisch* (dies ist der Standard). Die Option *Kompakt* führt eine Basisinstallation durch, während die Option *Angepasst* Ihnen die spezifische Auswahl der zu installierenden Funktionalitäten erlaubt. Klicken Sie zum Fortsetzen auf die *Weiter* Schaltfläche.

- Auswahl des Installationsordners.** Dieser Bildschirm erlaubt Ihnen die Anpassung des Laufwerks und Verzeichnisses in welches der DB2 Code auf ihrem System installiert wird. Stellen Sie sicher dass ausreichend freier Speicherplatz für die Installation zur Verfügung steht. Wählen Sie für dieses Beispiel das Standardlaufwerk und Standardverzeichnis (wie unten gezeigt):

Laufwerk: C:
Verzeichnis: C:\Programme\IBM\SQLLIB (Windows XP)
C:\Program Files (x86)\IBM\SQLLIB (Windows Vista 32-Bit)
C:\Program Files\IBM\SQLLIB (Windows Vista 64-Bit)

Klicken sie zum Fortsetzen auf die *Weiter* Schaltfläche.

- Benutzerinformation festlegen.** Sobald DB2 Express-C installiert ist laufen bestimmte DB2 Prozesse als Systemdienste. Diese Dienste erfordern zur Ausführung ein Betriebssystemkonto. In der Windows Umgebung wird die Verwendung des Standardbenutzerkontos **db2admin** empfohlen. Falls das Benutzerkonto noch nicht existiert, erstellt DB2 dieses im Betriebssystem für Sie. Sie können auch ein existierendes Benutzerkonto angeben, doch dieses muss lokale Administrationsrechte besitzen. Wir empfehlen die Verwendung der empfohlenen Standardwerte. Stellen Sie sicher, dass ein Kennwort für das Konto vergeben wird. Unter Linux verwenden Sie die standardmäßige **db2inst1** Benutzerkennung für den Besitzer der Instanz, **db2fenc1** für den abgeschirmten Benutzer (Fenced User) und **dasusr1** für den DB2 Administrationsserverbenutzer. Klicken sie zum Fortsetzen auf die *Weiter* Schaltfläche.
- DB2 Instanz konfigurieren.** Eine DB2 Instanz kann mit einem Behälter für Datenbanken verglichen werden. Eine Instanz muss existieren bevor eine Datenbank in ihr erstellt werden kann. Während der Installation unter Windows wird eine Instanz mit dem Namen *DB2* automatisch erstellt. In einer Linux Umgebung lautet der Name der Standardinstanz *db2inst1*. Wir behandeln Instanzen später in diesem Buch.

Standardmäßig ist die *DB2* Instanz so konfiguriert, dass sie für TCP/IP Verbindungen auf Port 50000 reagiert. Beide, das Standardprotokoll und der Port, können durch Anklicken der *Protokolle* oder *Start* Schaltfläche geändert werden. Wir empfehlen für dieses Beispiel die Verwendung der Standardeinstellungen. Klicken sie zum Fortsetzen auf die *Weiter* Schaltfläche.

- Installation starten.** Prüfen Sie die zuvor gewählten Installationsoptionen. Klicken Sie auf die *Installieren* Schaltfläche um mit dem Kopieren der

Dateien in das Installationsverzeichnis zu beginnen. DB2 wird zusätzlich einige grundlegende Konfigurationseinstellungen vornehmen.

11. **Erste Schritte.** Nachdem die Installation abgeschlossen ist, werden die Optionen Start Dienstprogramm und Erste Schritte angezeigt. Erste Schritte kann auch über den Befehl `db2fs` gestartet werden.
12. Bei der SAMPLE Datenbank handelt es sich um eine Datenbank, welche Sie für Testzwecke verwenden können. Sie wird sofort automatisch nach der Installation von DB2 erstellt. Prüfen Sie, ob die Datenbank existiert, indem sie die DB2 Steuerzentrale öffnen. Um dieses Dienstprogramm über das Windows Start Menü zu öffnen wählen Sie: *Start -> Programme -> IBM DB2 -> DB2COPY1 (Standard) -> Allgemeine Verwaltungstools -> Steuerzentrale*
Sie können die Steuerzentrale auch über den Befehl `db2cc` öffnen.
13. Falls die SAMPLE Datenbank in der Steuerzentrale angezeigt wird können Sie direkt zu Schritt 16 gehen. Falls sie nicht angezeigt wird wählen Sie, um sicherzustellen dass die aktuellsten Informationen angezeigt werden, den *Aktualisieren* Menüpunkt des *Ansicht* Menüs der Steuerzentrale. Falls die SAMPLE Datenbank immer noch nicht angezeigt wird, wurde diese eventuell nicht erstellt. Sie können diese manuell über Erste Schritte erstellen. Wählen Sie im linken Teilfenster "Datenbankerstellung" und folgen dem Assistenten um die SAMPLE Datenbank zu erstellen. Stellen Sie sicher dass die Option *XML- und SQL-Objekte und -Daten* gewählt ist und klicken auf OK. Die letzte Option wird eine UNICODE Datenbank erstellen, was in Version 9.1 zur Unterstützung von pureXML erforderlich war, jedoch in DB2 9.5 nicht länger benötigt wird.
14. Die folgende Fortschrittmeldung wird während der Datenerstellung angezeigt. (Dieser Vorgang kann mehrere Minuten dauern). Wenn die Datenbankerstellung abgeschlossen ist, klicken Sie auf die *OK* Schaltfläche und beenden das Erste Schritte Dienstprogramm.



15. Gehen Sie zurück zur Steuerzentrale und verifizieren das die SAMPLE Datenbank nunmehr in der Objektsicht erscheint. Um die Änderungen zu sehen, müssen Sie die Ansicht der Steuerzentrale eventuell aktualisieren.

Hierzu wählen Sie den *Aktualisieren* Menüpunkt des *Ansicht* Menüs der Steuerzentrale.

16. **Neustart des Computers.** Obwohl dieser Schritt nicht in der offiziellen DB2 Installationsdokumentation erwähnt ist empfehlen wir, um sicherzustellen das alle Dienste erfolgreich starten und den Speicher von Überbleibseln der Installation zu bereinigen, einen Neustart des Systems (wenn möglich, zumindest unter Windows). Dies ist OPTIONAL.

4

Kapitel 4 – DB2 Umgebung

In diesem Kapitel besprechen wir die DB2 Umgebung. Abbildung 4.1 gibt einen Überblick über DB2, wobei die rote Ellipse den für dieses Kapitel relevanten Bereich kennzeichnet. Die linke Seite der Abbildung zeigt die verschiedenen DB2 Befehle, SQL, SQL/XML und XQuery Anweisungen, welche für die Interaktion mit einem DB2 Datenserver verwendet werden können. In der Mitte der Abbildung sehen Sie die Namen der verschiedenen Dienstprogramme, welche für die Interaktion mit einem DB2 Datenserver eingesetzt werden können. Die rechte Seite der Abbildung zeigt die fundamentale DB2 Umgebung, bestehend aus einer Instanz, einer Datenbank und den zugehörigen Konfigurationsdateien. **DB2 Commands & SQL/XML, Xquery Statements**

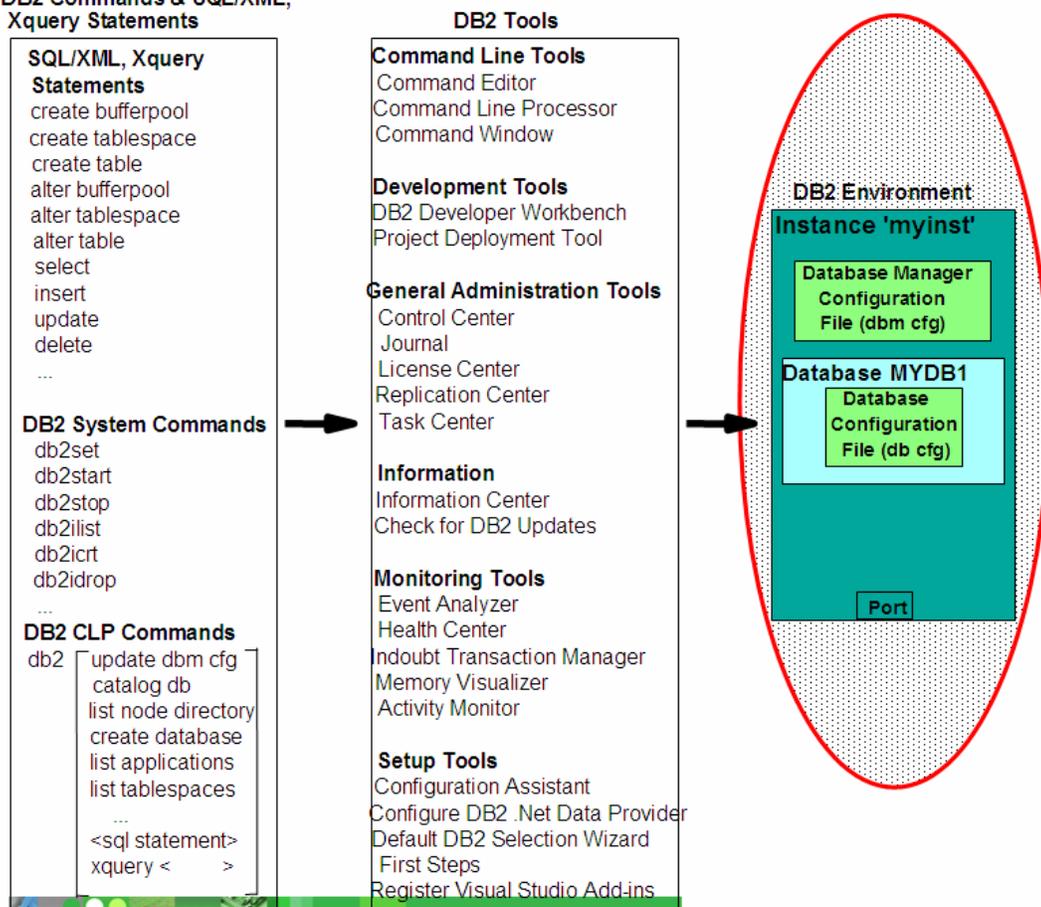


Abbildung 4.1 – DB2 Gesamtübersicht: DB2 Umgebung**Hinweis:**

Eine Video Präsentation über die DB2 Umgebung können Sie sich hier ansehen:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4029>

<http://www.channeldb2.com/video/video/show?id=807741:Video:4042>

Um die DB2 Umgebung zu beschreiben werden, Schritt für Schritt, die Bestandteile jeder Komponente erläutert. Abbildung 4.2 zeigt die Darstellung eines DB2 Datenservers nach der Installation von DB2 Express-C 9.5.

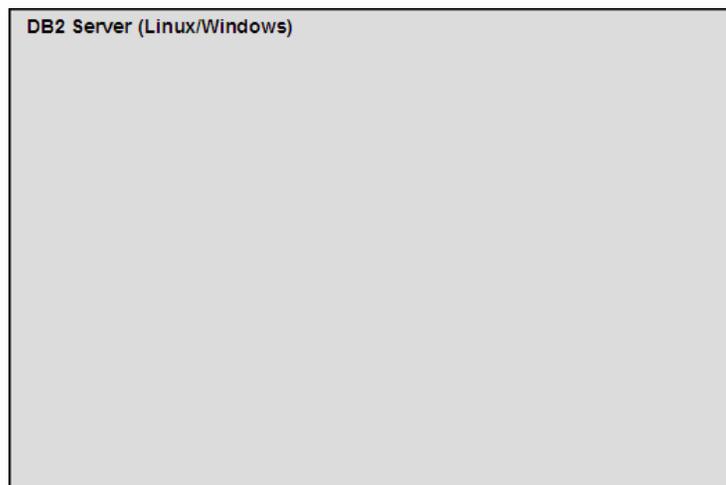


Abbildung 4.2 – Darstellung eines DB2 Servers nach der DB2 Express-C 9.5 Installation

Als Bestandteil der Installation unter Windows wird eine "DB2" genannte Standardinstanz ("db2inst1" unter Linux) erstellt. Diese wird durch den grünen Kasten in Abbildung 4.3 dargestellt. Einfacher gesagt handelt es sich bei einer Instanz um eine unabhängige Umgebung, in welcher Anwendungen laufen und Datenbanken erstellt werden können. Sie können mehrere Instanzen auf einem Datenserver anlegen und diese für verschiedene Zwecke verwenden, zum Beispiel eine Instanz für den Produktiveinsatz, eine Instanz für Datenbanken in einer Testumgebung und eine andere für eine Entwicklungsumgebung. Alle diese Instanzen sind unabhängig, mit der Folge, dass in einer Instanz ausgeführte Transaktionen die anderen Instanzen nicht beeinflussen.

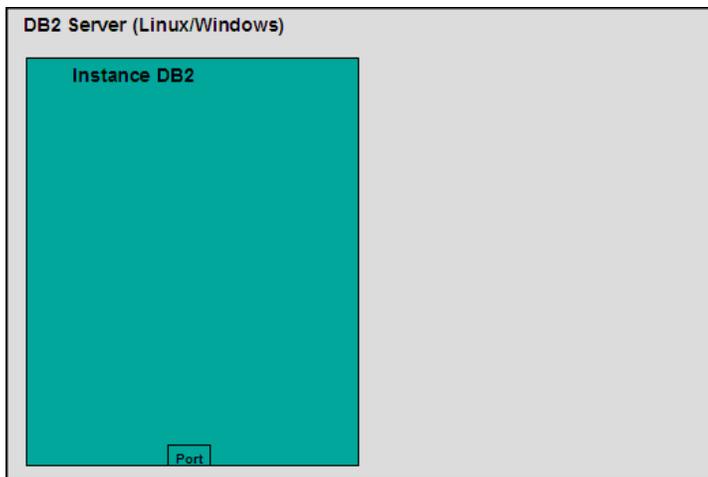


Abbildung 4.3 – Die erstellte Standard DB2 Instanz

Um eine neue DB2 Instanz anzulegen verwenden Sie den Befehl `db2icrt <instance name>`, wobei `<instance name>` durch einen 8 Zeichen langen Namen ersetzt wird. Um zum Beispiel die Instanz `myinst` zu erstellen verwenden wir diesen Befehl: `db2icrt myinst`.

Abbildung 4.4 zeigt eine neue, `myinst` genannte, Instanz in einem separaten grünen Kasten.

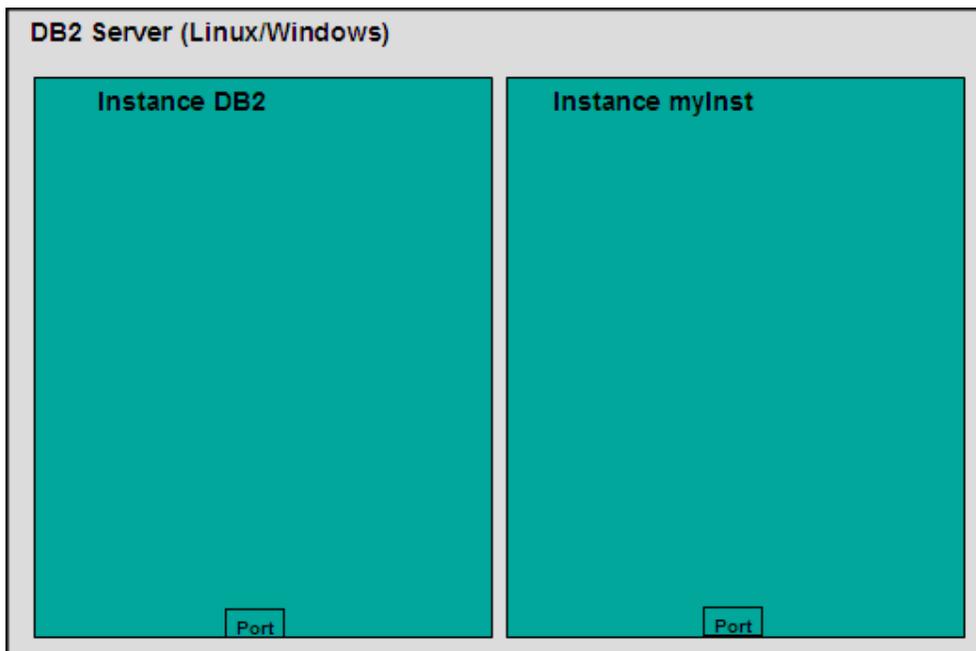


Abbildung 4.4 – Ein DB2 Server mit zwei Instanzen

Bitte beachten Sie, dass jede Instanz eine eindeutige Portnummer besitzt. Falls Sie von einem Remote Client eine Verbindung mit einer gegebenen Datenbank in einer gegebenen Instanz herstellen wollen, hilft dies zwischen den Instanzen zu unterscheiden. Falls Sie das DB2 Befehlsfenster verwenden, kann jede DB2 Instanz durch Verwendung der nachfolgenden Windows Befehlszeile zur Aktiven gemacht werden:

```
set db2instance=myinst
```

Wenn Sie jetzt eine Datenbank über das Befehlsfenster erstellen, wird diese in der Instanz `myinst` erstellt.

Um eine Liste aller Instanzen anzuzeigen, führen Sie diesen Befehl aus:

```
db2ilist
```

Unter Linux muss eine Instanz mit einem Linux Betriebssystembenutzer übereinstimmen. Um zwischen Instanzen zu wechseln kann unter Linux einfach der Benutzer mit dem `su` Befehl gewechselt werden.

Tabelle 4.1 zeigt einige nützliche Befehle auf Instanzebene.

Befehl	Beschreibung
db2start	Startet die aktuelle Instanz
db2stop	Stoppt die aktuelle Instanz
db2icrt	Erstellt eine neue Instanz
db2idrop	Löscht eine Instanz
db2ilist	Listet die auf Ihrem System vorhandenen Instanzen auf
db2 get instance	Gibt die derzeit aktive Instanz zurück

Tabelle 4.1 – Nützliche DB2 Befehle auf Instanzebene

Einige der oben aufgeführten Befehle können, wie in Abbildung 4.5 gezeigt, auch über die Steuerzentrale ausgeführt werden. In der Steuerzentrale können Sie zum Beispiel im Objektfenster den *Instanzen* Ordner erweitern, die erwünschte Instanz rechts anklicken, und *Starten* wählen, welches identisch mit dem im DB2 Befehlsfenster abgesetzten Befehl `db2start` ist, oder Sie wählen *Stoppen*, welches wiederum identisch mit dem Absetzen des Befehls `db2stop` ist.

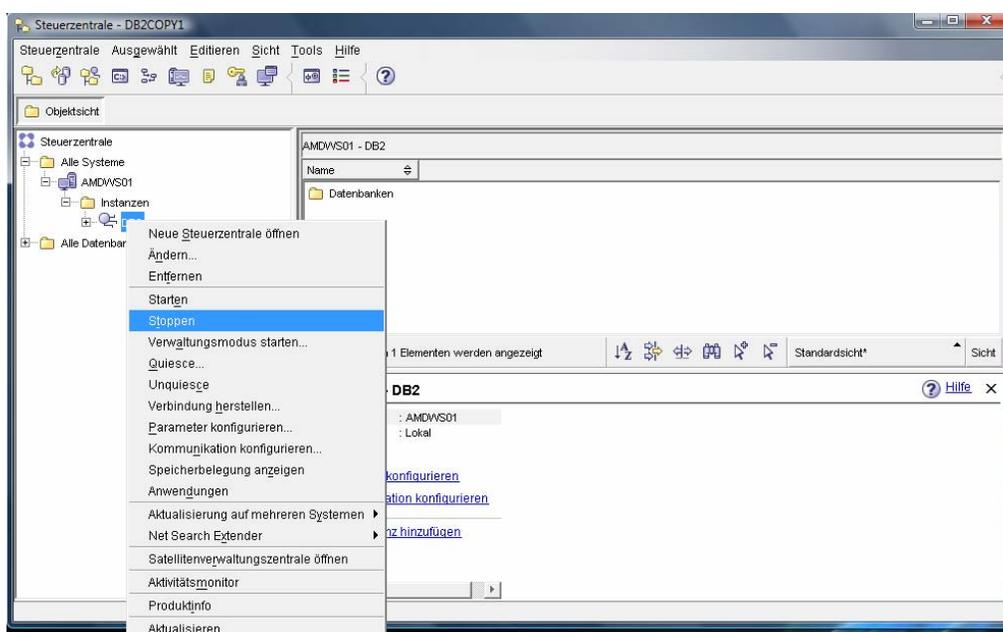


Abbildung 4.5 – Instanz Befehle über die Steuerzentrale aufrufen

Um eine Datenbank in der aktiven Instanz zu erstellen führen Sie diesen Befehl im DB2 Befehlsfenster aus:

```
db2 create database mydb1
```

Um eine Auflistung aller Datenbanken zu erhalten führen Sie diesen Befehl aus:

```
db2 list db directory
```

In einer Instanz können Sie viele Datenbanken erstellen. Eine Datenbank ist eine Sammlung von Objekten, wie Tabellen, Sichten, Indizes und so weiter. Datenbanken sind unabhängige Einheiten und teilen aus diesem Grund Objekte nicht mit anderen Datenbanken. Abbildung 4.6 zeigt eine Darstellung der in der Instanz "DB2" erstellten Datenbank "MYDB1".

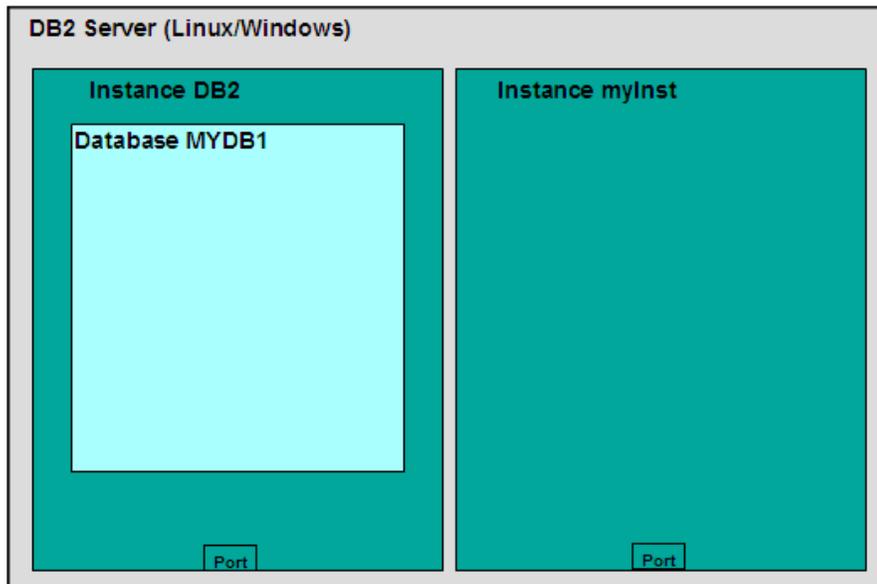


Abbildung 4.6 – In der Instanz “DB2” erstellte Datenbank “MYDB1”

Tabelle 4.2 zeigt einige Befehle welche auf Datenbankebene verwendet werden können.

Befehl/SQL Anweisung	Beschreibung
db2 create database	Erstellt eine neue Datenbank
db2 drop database	Löscht eine Datenbank
db2 connect to <database_name>	Verbindet mit einer Datenbank
db2 create table/create view/create index	SQL Anweisungen zur Erstellung von Tabellen, Sichten und Indizes

Tabelle 4.2 - Befehle/SQL Anweisungen auf Datenbankebene

Falls wir eine andere Datenbank mit identischem Namen (MYDB1) in der Instanz “myinst” erstellen wollten, müssten die folgenden Befehle über das DB2 Befehlsfenster abgesetzt werden:

```
db2 list db directory
set db2instance=myinst
db2 create database mydb1
set db2instance=db2
```

Abbildung 4.7 zeigt die in der Instanz “myinst” erstellte neue Datenbank “MYDB1”.

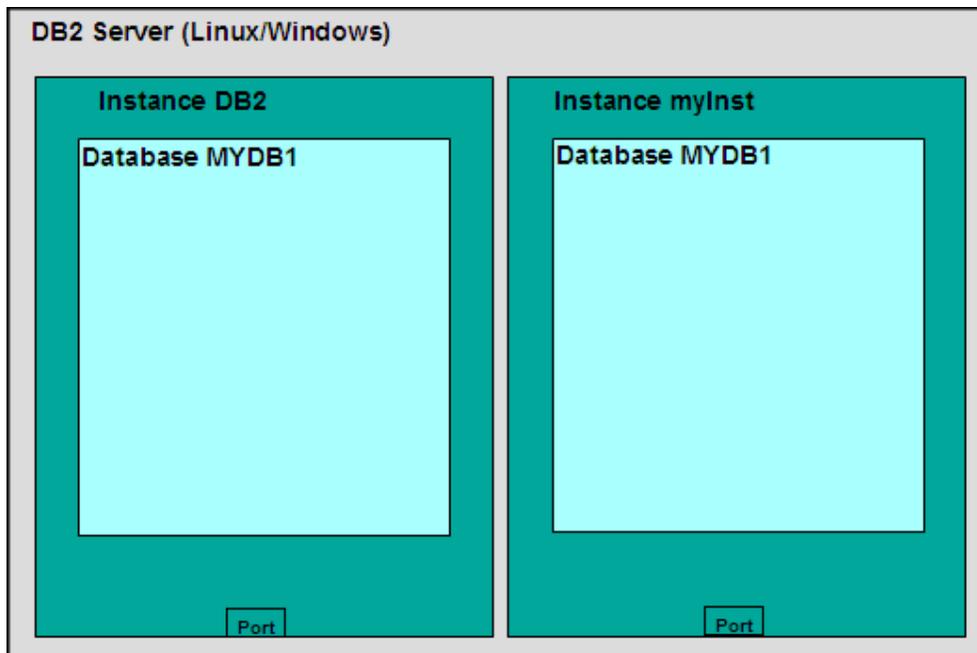


Abbildung 4.7 – In der Instanz “myInst” erstellte Datenbank “MYDB1”

Wenn eine Datenbank erstellt wird, werden standardmäßig verschiedene Objekte angelegt: Tabellenbereiche, Tabellen, ein Pufferpool und Protokolldateien. Der `create database` Befehl benötigt da die Erstellung der Objekte etwas Zeit in Anspruch nimmt, einige Minuten für die Verarbeitung. Abbildung 4.8 zeigt die drei standardmäßig erstellten Tabellenbereiche. Tabellenbereiche werden ausführlicher in Kapitel 6, DB2 Architektur, erläutert. Zu diesem Zeitpunkt betrachten Sie Tabellenbereiche als die logische Ebene zwischen logischen Tabellen und physikalischen Ressourcen, wie Festplatten und Speicher.

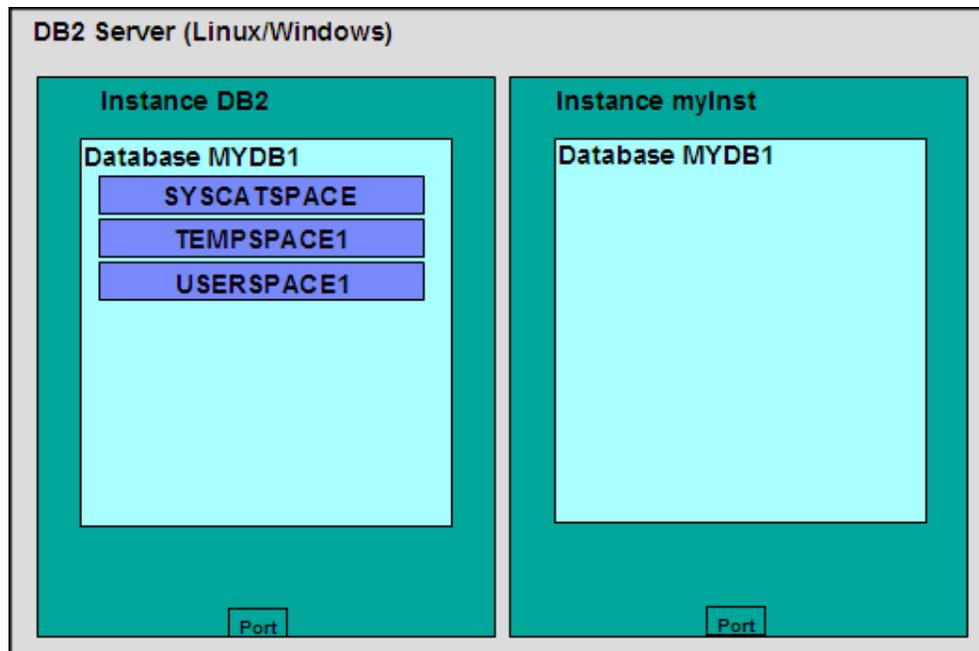


Abbildung 4.8 – Standardmäßig erstellte Tabellenbereiche wenn eine Datenbank angelegt wird

Der Tabellenbereich SYSCATSPACE enthält die Katalogtabellen. Der Katalog ist auch in anderen relationalen Datenbank Management Systemen als Datenwörterbuch (Data Dictionary) bekannt. Es enthält grundsätzlich Systeminformationen welche weder geändert noch gelöscht werden können, da die Datenbank andernfalls nicht einwandfrei arbeiten würde. Der Tabellenbereich TEMPSPACE1 wird von DB2 verwendet, wenn zusätzlicher Platz für die Ausführung bestimmter Aktionen, wie Sortiervorgänge, erforderlich ist. Der Tabellenbereich USERSPACE1 wird normalerweise verwendet um Datenbanktabellen des Benutzers zu speichern, sofern bei der Erstellung der Tabelle kein Tabellenbereich angegeben wurde.

Mit dem CREATE TABLESPACE Befehl können Sie auch ihren eigenen Tabellenbereich erstellen. Abbildung 4.9 zeigt das der Tabellenbereich MYTBLS1 in der Datenbank MYDB1 der Instanz DB2 erstellt wurde. Wenn Sie einen Tabellenbereich erstellen, geben sie die zu verwendenden Festplatten und den zu verwendenden Speicher (Pufferpool) an. Folglich, falls Sie eine "heiße" Tabelle haben, welches bedeutet, dass es sich um eine sehr oft verwendete Tabelle handelt, können sie die schnellsten Festplatten und den meisten Speicher zuordnen, indem sie einen Tabellenbereich mit diesen Charakteristika erstellen.

In Abbildung 4.9 zeigen wir zwei andere standardmäßig erstellte Objekte: Ein Pufferpool namens IBMDEFAULTBP und die Protokolldateien.

Bei einem Pufferpool handelt es sich im Grunde genommen um einen von der Datenbank verwendeten Cache. Sie können einen oder mehrere Pufferpools erstellen. Es sollte jedoch immer ein Pufferpool mit einer Seitengröße, welche mit der Seitengröße

existierender Tabellenbereiche übereinstimmt, erstellt werden. Seiten und Seitengrößen werden ausführlicher in Kapitel 6, DB2 Architektur, erläutert.

Die Protokolldateien werden für die Wiederherstellung verwendet. Wenn Sie mit einer Datenbank arbeiten werden die Informationen nicht nur auf Festplatten gespeichert, sondern zusätzlich werden, während sie mit der Datenbank arbeiten, alle auf Daten ausgeführten Vorgänge in Protokolldateien gespeichert. Protokolldateien können Sie als temporäre Dateien, in welchen eine "automatisches speichern" Operation durchgeführt wird, ansehen. Protokolldateien werden ausführlicher in Kapitel 11, Sicherung und Wiederherstellung, erläutert.

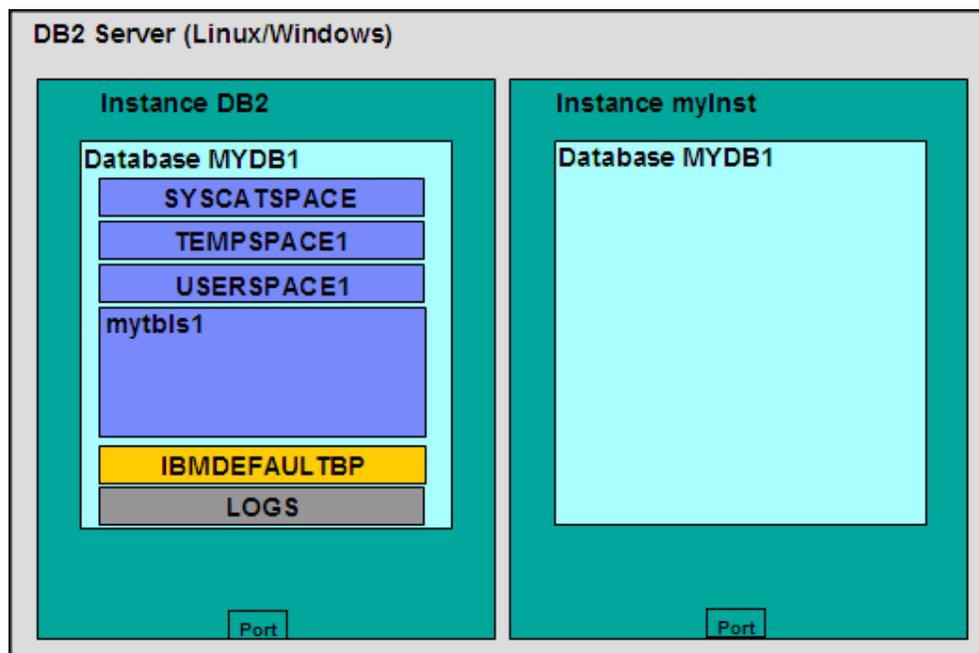


Abbildung 4.9 – Standardmäßig erstellter Pufferpool und Protokolle

Bereits zuvor besprochen wir, dass es sich bei Instanzen um unabhängige Umgebungen handelt und es aus diesem Grund möglich ist eine Datenbank mit identischem Namen in unterschiedlichen Instanzen zu erstellen. Wie bei Instanzen handelt es sich bei Datenbanken ebenfalls um unabhängige Einheiten; weshalb Objekte in einer Datenbank keine Beziehung zu Objekten einer anderen Datenbank besitzen. Abbildung 4.10 zeigt den Tabellenbereich "mytbls1" in den beiden Datenbanken MYDB1 und SAMPLE in der Instanz DB2. Da es sich bei Datenbanken um unabhängige Einheiten handelt ist dies zulässig. Bitte beachten Sie dass in Abbildung 4.10, da dafür der Platz in der Grafik nicht ausreicht, die anderen standardmäßig erstellten Objekte der Datenbank SAMPLE nicht dargestellt sind.

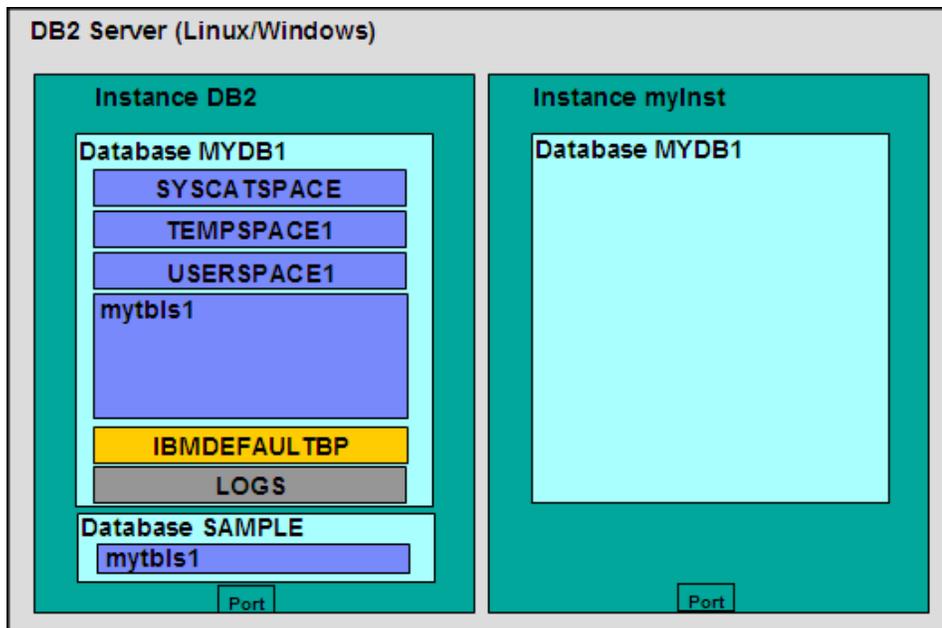


Abbildung 4.10 – Tabellenbereiche mit demselben Namen in unterschiedlichen Datenbanken.

Sobald Sie einen Tabellenbereich erstellt haben, können sie Objekte, wie Tabellen, Sichten und Indizes, in diesem Tabellenbereich erstellen. Dies wird in Abbildung 4.11 illustriert.

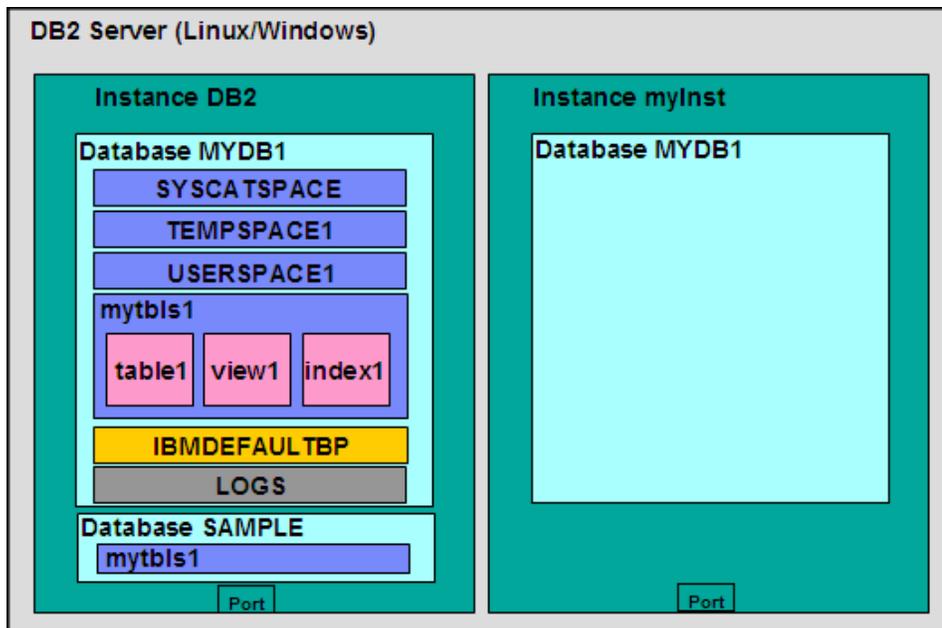


Abbildung 4.11 – Im Tabellenbereich erstellte Tabellen, Sichten und Indizes

Übung 2 – Erstellen einer neuen Datenbank

Ziel

In dieser Übung werden Sie eine neue Datenbank mit dem Datenbank erstellen Assistenten in der Steuerzentrale erstellen.

Vorgehensweise

1. Im Objektsicht Fenster der Steuerzentrale klicken Sie rechts auf *Alle Datenbanken* und wählen den Menüpunkt *Datenbank erstellen* und den Untermenüpunkt *Mit automatischer Verwaltung*. Hierdurch wird der Datenbank mit automatischer Verwaltung erstellen Assistent aufgerufen.
2. Geben Sie den Datenbanknamen und Speicherort auf der *Name* Seite des Assistenten ein. Verwenden Sie die folgenden Werte:

Datenbankname:	EXPRESS
Standardpfad (Windows):	C:
Standardpfad (Linux):	/home/db2inst1
Aliasname:	Hier wird, falls kein Aliasname eingetragen wird, per Default der Aliasname EXPRESS festgelegt
Kommentar:	Dieser ist optional und kann leer bleiben

Klicken Sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.

3. Auf der *Speicherposition für Ihre Daten angeben* Seite nehmen Sie keine Änderungen vor und klicken auf die Schaltfläche *Weiter*.
4. Auf der *Eine Verwaltungsstrategie auswählen* Seite ändern Sie den Standardwert (*Ja, ich kann ein Offlineverwaltungsfenster...*) nicht und klicken auf *Weiter*.
5. Spezifizieren Sie auf der nächsten Seite ein Offlineverwaltungsfenster. Geben Sie zwei oder mehr Stunden pro Woche, in welchen DB2 zur Sicherstellung der ordnungsgemäßen Funktion ihrer Datenbank automatische Verwaltungsaufgaben durchführen kann. Konfigurieren Sie das Zeitfenster so, dass es jeden Montag bis Donnerstag um 1:00 Uhr für eine Dauer von 6 Stunden startet. Klicken Sie auf die *Weiter* Schaltfläche.
6. Konfigurieren Sie die Benachrichtigung auf der *Mail-Server* Seite des Assistenten. DB2 kann automatisch eine E-Mail oder eine Seite senden, falls ein Problem oder ein die Funktionalität beeinträchtigender Zustand

festgestellt wird. Wenn Sie dies konfigurieren möchten, müssen sie einen gültigen SMTP Server, welchen DB2 für den Versand von E-Mails verwenden kann, angeben. In dieser Übung ist kein SMTP Server vorhanden, so dass Sie das Feld freilassen können und auf die Schaltfläche *Weiter* klicken.

7. Überprüfen Sie die gewählten Aktionen auf der *Zusammenfassung* Seite. Klicken Sie auf die *Fertig stellen* Schaltfläche um mit der Erstellung der Datenbank zu beginnen. Die Erstellung einer Datenbank dauert ein paar Minuten. Zu Ihrer Information wird während der Datenbankerstellung ein Fortschrittsbalken angezeigt.

4.1 DB2 Konfiguration

DB2 Parameter können mit dem Konfigurationsadvisor Dienstprogramm konfiguriert werden. Um auf den Konfigurationsadvisor über die Steuerzentrale zuzugreifen klicken Sie rechts auf eine Datenbank und wählen " Konfigurationsadvisor". Basierend auf die von Ihnen auf einige Fragen gegebenen Antworten über ihre Systemressourcen und die Auslastung wird der Konfigurationsadvisor eine Liste von DB2 Parametern anzeigen, welche in die jeweils empfohlenen Werte geändert werden sollten. Wenn Sie detailliertere Informationen über die DB2 Konfiguration benötigen sollten sie weiter lesen, ansonsten verwenden sie den Konfigurationsadvisor und der Arbeit mit DB2 steht nichts im Weg!

Ein DB2 Server kann auf vier verschiedenen Stufen konfiguriert werden:

- Umgebungsvariablen
- Datenbankmanager (DBM) Konfigurationsdatei (DBM CFG)
- Datenbank (DB) Konfigurationsdatei (DB CFG)
- DB2-Registrierdatenbank

Dies wird auch in Abbildung 4.12 gezeigt. Bitte beachten Sie in der Abbildung wo sich jeder Kasten befindet. Umgebungsvariablen werden zum Beispiel auf Betriebssystemebene gesetzt und Parameter der Datenbankmanager Konfigurationsdatei auf Instanzebene gesetzt. Datenbankkonfigurationsparameter werden auf Datenbankebene gesetzt und die DB2-Registrierdatenbank wird entweder auf Betriebssystemebene oder Instanzebene gesetzt.

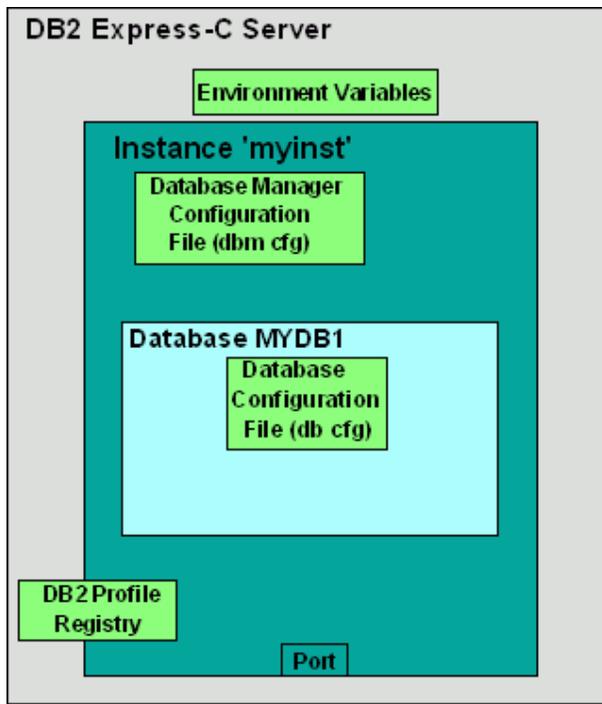


Abbildung 4.12 – DB2 Konfiguration

4.1.1 Umgebungsvariablen

Umgebungsvariablen werden auf Betriebssystemebene gesetzt. Eine Schlüsselumgebungsvariable ist DB2INSTANCE. Diese Variable gibt die aktive Instanz, mit welcher Sie arbeiten und auf welche DB2 Befehle zutreffen würden, an. Um zum Beispiel "myinst" unter Windows zur aktiven Instanz zu deklarieren können Sie diesen Betriebssystembefehl ausführen:

```
set db2instance=myinst
```

4.1.2 Datenbankmanager Konfigurationsdatei (DBM CFG)

Die Datenbankmanager Konfigurationsdatei (DBM CFG) enthält Parameter welche die Instanz und alle darin enthaltenen Datenbanken beeinflusst. Die Datenbankmanager Konfigurationsdatei kann über die Befehlszeile oder die DB2 Steuerzentrale angesehen oder geändert werden.

Um die DBM CFG über die Steuerzentrale zu bearbeiten, wählen Sie das Instanz Objekt aus dem Ordner Instanz, klicken rechts um das Kontextmenü anzuzeigen und wählen Parameter konfigurieren. Dies wird in Abbildung 4.13 gezeigt.

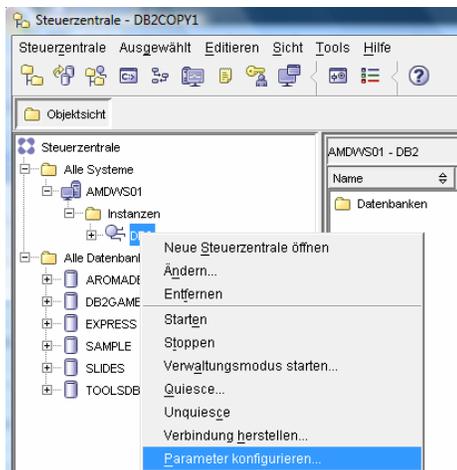


Abbildung 4.13 – Konfiguration der DBM CFG über die Steuerzentrale.

Nachdem Parameter konfigurieren gewählt wurde, wird der in Abbildung 4.14 gezeigte Bildschirm mit einer Liste der DBM CFG Parameter angezeigt.

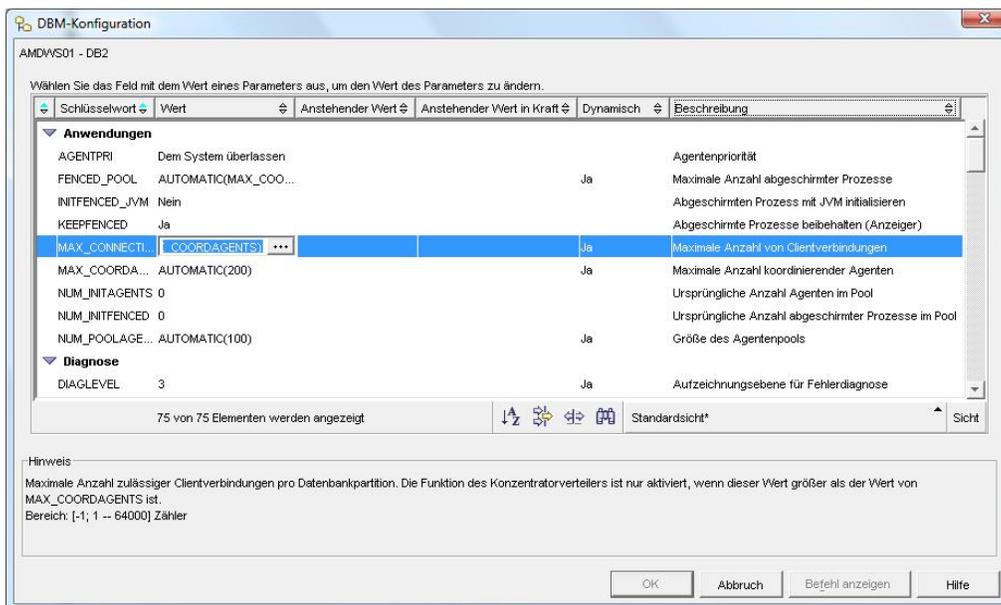


Abbildung 4.14 – Der DBM-Konfiguration Dialog

Viele Parameter sind dynamisch, was bedeutet, dass Änderungen sofort angewandt werden. Wie dem auch sei, die Änderungen an einigen Parametern werden eventuell erst nach dem Stoppen und erneuten Starten wirksam. Über die Befehlszeile kann dies mit den `db2stop` und `db2start` Befehlen erreicht werden.

Bevor eine Instanz gestoppt werden kann müssen alle verbundenen Anwendungen getrennt werden. Falls sie das Stoppen der Instanz erzwingen wollen kann der `db2stop force` Befehl verwendet werden.

Eine Instanz kann ebenfalls über die Steuerzentrale gestoppt werden, indem Sie das Instanz Objekt anklicken und entweder Stoppen oder Starten wählen.

Tabelle 4.3 zeigt einige nützliche Befehle um das DBM CFG über die Befehlszeile zu verwalten.

Befehl	Beschreibung
<code>db2 get dbm cfg</code>	Ruft Informationen über das dbm cfg ab
<code>db2 update dbm cfg using <parameter_name> <value></code>	Aktualisiert den Wert eines dbm cfg Parameters

Tabelle 4.3 – Befehle zur Verwaltung der DBM CFG

4.1.3 Datenbankkonfigurationsdatei (DB CFG)

Die Datenbankkonfigurationsdatei (DB CFG) enthält Parameter, welche eine bestimmte Datenbank beeinflussen. Die Datenbankkonfigurationsdatei kann über die Befehlszeile oder DB2 Steuerzentrale angesehen oder modifiziert werden.

Um mit der DB CFG über die Steuerzentrale zu arbeiten wählen Sie das Datenbank Objekt aus dem Datenbanken Ordner der Steuerzentrale, klicken rechts um das Kontextmenü anzuzeigen und wählen Parameter konfigurieren. Dies wird in Abbildung 4.15 gezeigt.

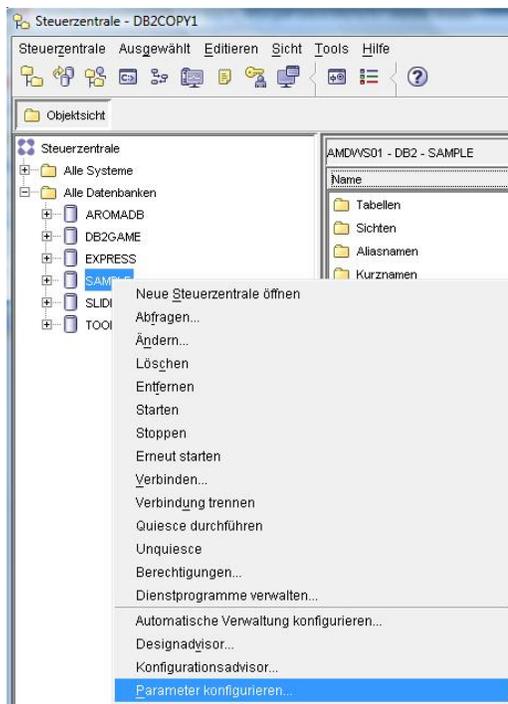


Abbildung 4.15 – Konfiguration des DB CFG über die Steuerzentrale.

Nachdem Sie Parameter konfigurieren gewählt haben wird, wie in Abbildung 4.16 gezeigt, ein Bildschirm mit einer Liste der DB CFG Parameter angezeigt.

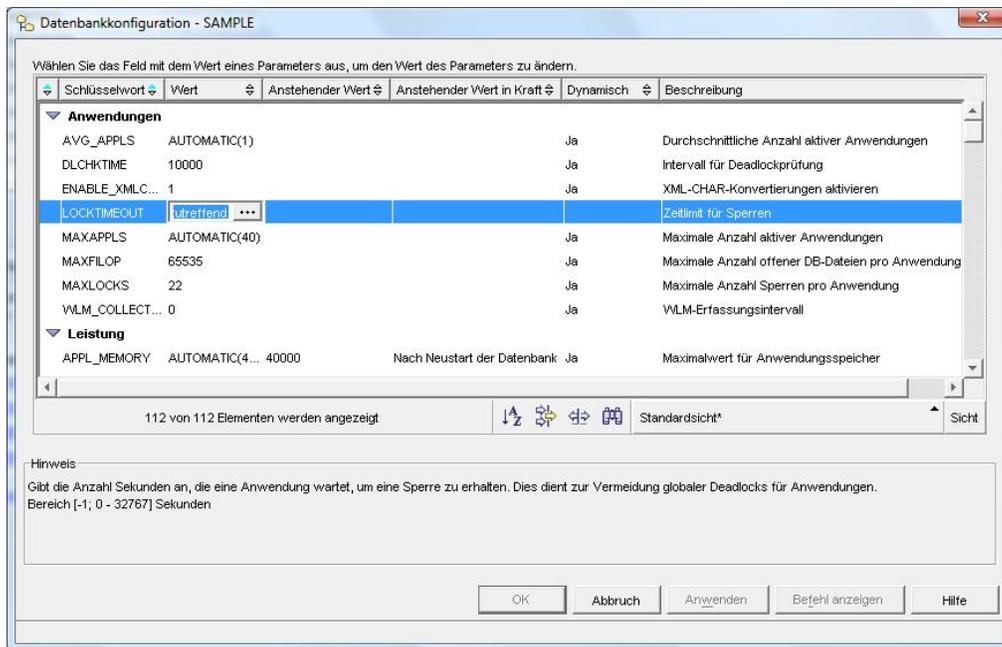


Abbildung 4.16 – Der Datenbankkonfiguration (DB CFG) Dialog

Tabelle 4.4 zeigt einige nützliche Befehle zur Verwaltung der DB CFG über die Befehlszeile.

Befehl	Beschreibung
<code>get db cfg for <database_name></code>	Ruft DB CFG Informationen für eine bestimmte Datenbank ab
<code>update db cfg for <database_name> using <parameter_name> <value></code>	Aktualisiert den Wert eines DB CFG Parameters

Tabelle 4.4 – Befehle zur Verwaltung der DB CFG

4.1.4 DB2-Registrierdatenbank

DB2-Registrierdatenbankvariablen enthalten Parameter welche eventuell plattformspezifisch sind und können global (beeinflussen alle Instanzen) oder auf Instanz-Ebene (beeinflusst eine bestimmte Instanz) gesetzt werden.

Tabelle 4.5 zeigt einige nützliche Befehl zur Verwaltung der DB2-Registrierdatenbank

Befehl	Beschreibung
<code>db2set -all</code>	Listet alle aktuell gesetzten DB2-Registrierdatenbankvariablen auf
<code>db2set -l r</code>	Listet alle DB2-Registrierdatenbankvariablen auf

<code>db2set <parameter>=<value></code>	Weist einem Parameter einen Wert zu
---	-------------------------------------

Tabelle 4.5 – Befehle zur Verwaltung der DB2-Registrierdatenbank

Tabelle 4.6 zeigt einige der meist verwendeten DB2 Registriervariablen

Registriervariable	Beschreibung
DB2COMM	Gibt den Kommunikationsmanager an, welcher beim Start der Datenbank gestartet wird.
DB2_EXTSECURITY	Verhindert unter Windows, durch das Sperren der DB2 Systemdateien, unberechtigten Zugriff auf DB2.
DB2_COPY_NAME	Speichert den Namen der aktuell verwendeten DB2 Kopie. Um zu einer anderen installierten DB2 Kopie zu wechseln führen Sie die Stapelverarbeitungsdatei Installationspfad\bin\db2envvars.bat aus. Diese Variable kann nicht für diesen Zweck verwendet werden.

Tabelle 4.6 – Oft verwendete DB2-Registriervariable

Um zum Beispiel TCPIP für die Kommunikation zuzulassen setzen Sie den DB2COMM Registriervariable wie unten gezeigt:

```
db2set db2comm=tcPIP
```

4.2 Der DB2 Administration Server

Beim DB2 Administration Server (DAS) handelt es sich um einen auf dem DB2 Server laufenden Hintergrundprozess, welcher es entfernten Clients gestattet den DB2 Server mit grafischen Werkzeugen zu verwalten. Es gibt, wie in Abbildung 4.16 gezeigt, nur einen DAS je physikalischen Computer.

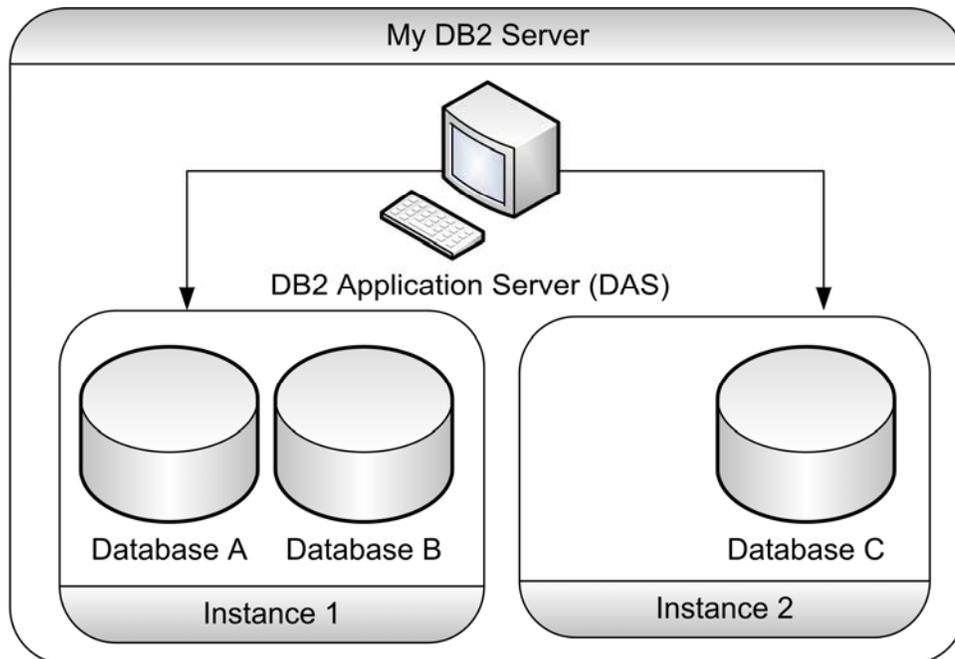


Abbildung 4.16 – Der DB2 Administration Server (DAS)

Übung 3 – Arbeiten mit Instanzen, Datenbanken und Konfiguration

Ziel

In dieser Übung erstellen Sie eine neue Instanz, Datenbank und ändern Konfigurationsparameter auf einem DB2 Server unter Windows. Dies können Sie entweder über die Steuerzentrale oder die das Befehlsfenster ausführen. Nachfolgend finden Sie die Anweisungen für das Befehlsfenster.

Vorgehensweise

1. Erstellen Sie über das Befehlsfenster eine neue, *newinst* genannte, Instanz
`db2icrt newinst`
2. In der neuen Instanz *newinst* erstellen Sie die Datenbank *newdb* mit Standardwerten
`set db2instance=newinst`
`db2start`
`db2 create database newdb`
3. Listen sie alle Instanzen auf ihrem Server auf
`db2ilist`
4. Wechseln Sie zur Instanz *DB2* und stellen sicher, dass sie wirklich gewechselt haben
`set db2instance=db2`
`db2 get instance`
5. Ändern Sie den DBM CFG Parameter FEDERATED von Nein auf Ja und stellen sie sicher, dass die Änderung vorgenommen wurde.
`db2 update dbm cfg using FEDERATED YES`
`db2 force applications all`
`db2 terminate`
`db2stop`
`db2start`
`db2 get dbm cfg`
6. Verbinden Sie sich mit der Datenbank *SAMPLE* unter Verwendung der Benutzerkennung/des Kennworts, welches zur Anmeldung am Betriebssystem verwendet wurde
`db2 connect to sample user <userID> using <psw>`
7. Überprüfen Sie wie viele Anwendungen in ihrer aktuellen Instanz laufen
`db2 list applications show detail`

8. Öffnen Sie ein weiteres DB2 Befehlsfenster und verbinden sich erneut mit der SAMPLE Datenbank ohne Angabe der Benutzerkennung/des Kennworts. Anschließend prüfen Sie die Anzahl der nunmehr vorhandenen Verbindungen.

```
db2 connect to sample
db2 list applications
```
9. Erzwingen Sie die Beendigung eines der DB2 Befehlsfenster

```
db2 force application
```

 (mit dem db2 Befehl `list applications` ermitteltes Anwendungskennzeichen der Anwendung mit dem Namen "db2bp.exe")
10. Löschen Sie die Instanz *newinst*

```
db2idrop newinst
```
11. Löschen Sie den DAS, erstellen ihn neu und starten ihn.

```
db2admin stop
db2admin drop
db2admin create
db2admin start
```
12. Setzen die DB2-Registriervariable in ihrer Instanz DB2COMM auf TCPIP und NPIPE

```
db2set db2comm=tcpip,npipe
db2stop
db2start
```
13. Setzen Sie die Registriervariable DB2COMM zurück

```
db2set db2comm=
db2stop
db2start
```
14. Prüfen Sie den aktuellen Wert des LOGSECOND DB CFG Parameters, ändern den Wert auf 5 und verifizieren den neuen Wert

```
db2 connect to sample
db2 get db cfg
db2 update db cfg using LOGSECOND 5
db2 get db cfg
```

5

Kapitel 5 – DB2 Dienstprogramme

In diesem Kapitel beschreiben wird einige der mit DB2 verwendbaren Dienstprogramme. Die rote Ellipse in Abbildung 5.1 zeigt worauf wir unser Augenmerk in diesem Kapitel richten.

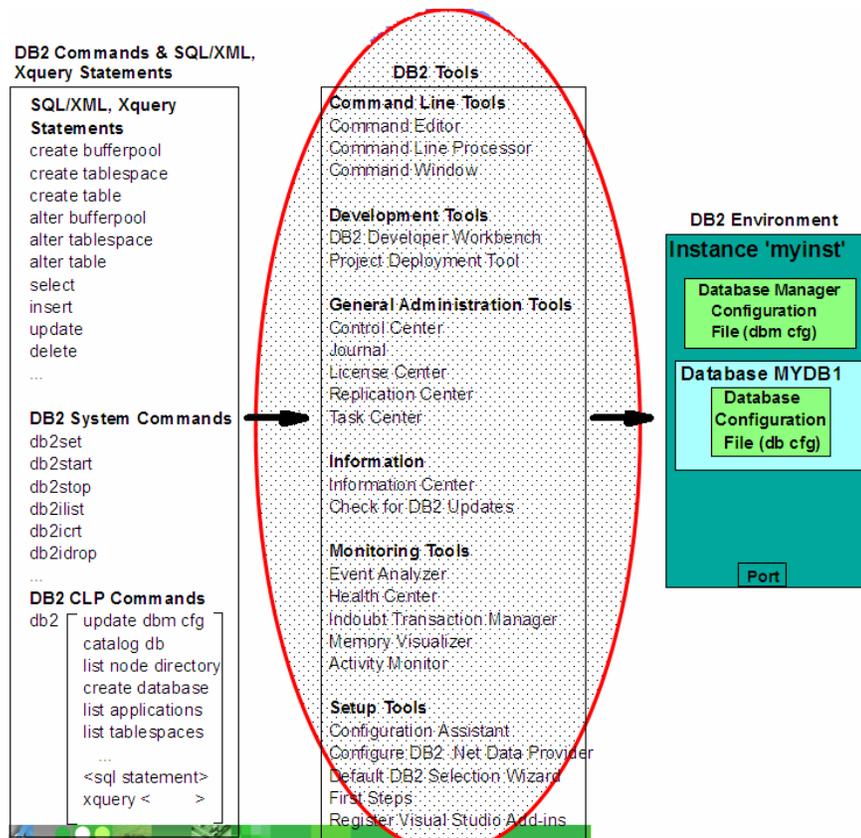


Abbildung 5.1 – Das DB2 Gesamtbild: DB2 Dienstprogramme

Hinweis:

Präsentationen über die DB2 Dienstprogramme und Scripting können Sie sich unter den folgenden Links ansehen:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4202>

<http://www.channeldb2.com/video/video/show?id=807741:Video:4182>

In Abbildung 5.2 sehen Sie die über den IBM DB2 Startmenüeintrag verfügbaren DB2 Dienstprogramme. Die meisten Dienstprogramme sind unter Linux und Windows identisch.

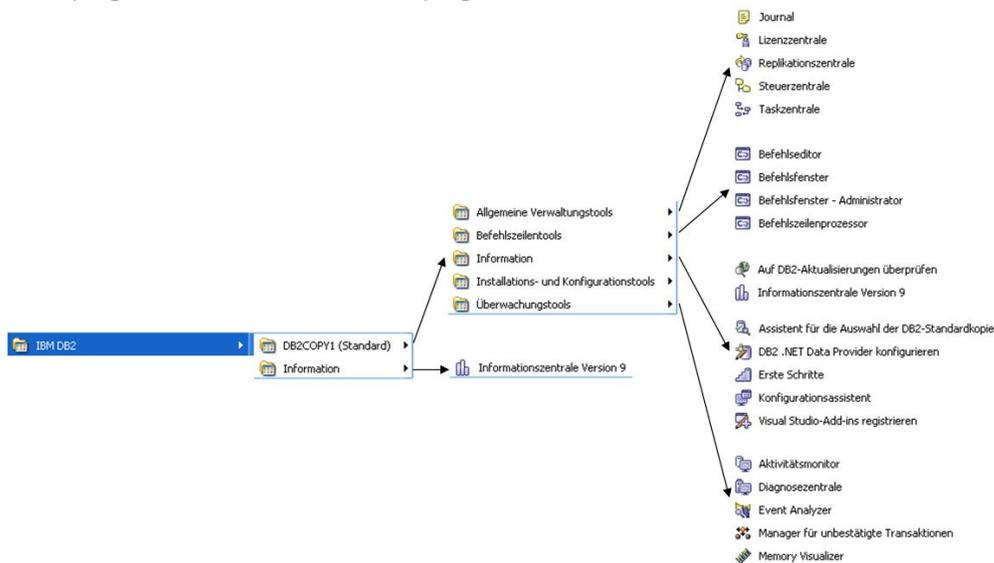


Abbildung 5.2 – DB2 Dienstprogramme über das Startmenü

Tabelle 5.1 gibt eine Übersicht über Befehlszeilenkommandos, welche zum Start der gebräuchlichsten Dienstprogramme unter Linux oder Windows verwendet werden können.

Name des Dienstprogramms	Befehl
Befehlseditor	db2ce
Befehlszeilenprozessor	db2
DB2 CLP (nur auf Windows Plattformen)	db2cmd
Steuerzentrale	db2cc
Taskzentrale	db2tc
Diagnosezentrale	db2hc
Konfigurationsassistent	db2ca
Erste Schritte	db2fs

Tabelle 5.1 – Befehlszeilenkommandos zum Start einiger DB2 Dienstprogramme

5.1 Steuerzentrale

Das elementare DB2 Dienstprogramm zur Datenbankadministration ist, wie in Abbildung 5.3 gezeigt, die Steuerzentrale.

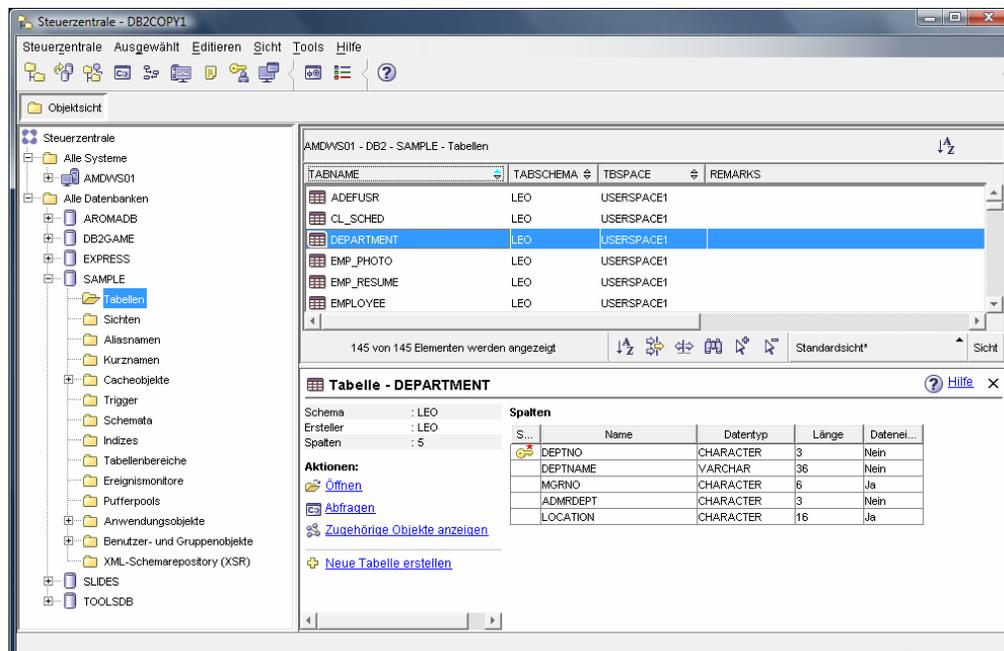


Abbildung 5.3 - Die DB2 Steuerzentrale

Bei der Steuerzentrale handelt es sich um ein zentralisiertes Administrationswerkzeug das Ihnen folgende Funktionen ermöglicht:

- Ansehen Ihrer Systeme, Instanzen, Datenbanken und Datenbankobjekte;
- Erstellen, modifizieren und verwalten von Datenbanken und Datenbankobjekten;
- Starten anderer grafischer DB2 Dienstprogramme

Im linken Teilfenster sehen Sie die visuelle Hierarchie der Datenbankobjekte auf ihrem System, mit "Ordnern" für Tabellen, Sichten usw. Wenn Sie auf einen Ordner doppelklicken (zum Beispiel den Tabellen Ordner, wie in Abbildung 5.3 gezeigt) zeigt das obere rechte Teilfenster alle zugehörigen Objekte, in diesem Fall alle mit der Datenbank SAMPLE verbundenen Tabellen. Wenn Sie eine bestimmte Tabelle im oberen rechten Teilfenster auswählen zeigt das untere rechte Teilfenster genauere Informationen zu dieser Tabelle an.

Ein Rechtsklick auf die verschiedenen Ordner/Objekte im Objektbaum öffnet ein Menü für den gewählten Ordner bzw. das gewählte Objekt. So bewirkt ein Rechtsklick auf eine Instanz und die Auswahl von "Parameter konfigurieren" dass Sie die Konfigurationsdatei

des Datenbankmanagers ansehen und aktualisieren können. Die DB2 Umgebung und Konfigurationsparameter werden ausführlicher in Kapitel 5, DB2 Umgebung, besprochen.

Wenn Sie die Steuerzentrale zum ersten Mal starten werden sie gefragt welche Sicht der Steuerzentrale sie verwenden möchten. Die Auswahl der Sicht legt fest welche Typen von Optionen und Datenbankobjekten angezeigt werden. Abbildung 5.4 zeigt den Sicht der Steuerzentrale Dialog.



Abbildung 5.4 – Der Sicht der Steuerzentrale Dialog

Die Sicht Basis bietet grundlegende DB2 Funktionalität.

Die Sicht Erweitert zeigt mehr Optionen und Funktionalitäten.

Die Sicht Angepasst gestattet die Auswahl der sichtbaren spezifischen Funktionalitäten, Optionen und Objekte.

Um den Sicht der Steuerzentrale Dialog erneut anzuzeigen wählen Sie, wie in Abbildung 5.5 gezeigt, aus dem Tools Menü der Steuerzentrale den Menüeintrag "Steuerzentrale anpassen" anpassen.

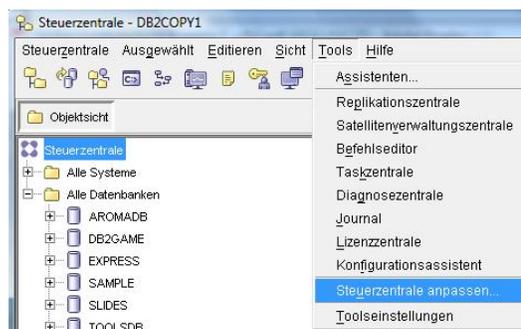


Abbildung 5.5 – Anpassen der Steuerzentrale

Starten der Steuerzentrale

Es gibt verschiedene Möglichkeiten die Steuerzentrale zu starten:

- Navigation durch das Windows *Start* Menü

- Ausführen von *db2cc* über die Befehlszeile
- Durch anklicken des Symbols Steuerzentrale  in der Werkzeugleiste eines anderen grafischen DB2 Dienstprogramms
- Über das DB2 Symbol im Windows Infobereich, wie Abbildung 5.6 zeigt (rechtsklick auf das grüne DB2 Symbol und Auswahl des Menüpunkts DB2-Steuerzentrale)



Klicken Sie rechts auf das grüne DB2 Symbol und wählen einen Menüpunkt.

Abbildung 5.6 – Starten der DB2 Steuerzentrale über den Windows Infobereich

5.2 Befehlseditor

Der DB2 Befehlseditor ermöglicht die Ausführung von DB2 Befehlen, SQL und XQuery Anweisungen, die Analyse des Zugriffsplans und die Ansicht oder Aktualisierung von Abfrageresultaten.

Abbildung 5.7 zeigt den Befehlseditor mit einer Beschreibung der Elemente.

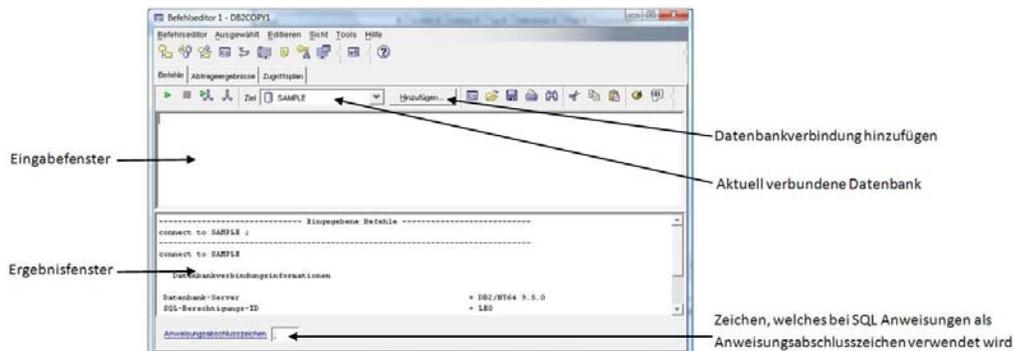


Abbildung 5.7 – DB2 Befehlseditor

Im Eingabefenster können Sie, unter der Voraussetzung dass jeder Befehl mit einem Anweisungstrennzeichen endet, eingeben. Wenn Sie die Ausführen Schaltfläche (siehe Abbildung 5.8) betätigen werden die Anweisungen nacheinander ausgeführt. Wenn Sie eine bestimmte Anweisung markieren wird nur die markierte Anweisung ausgeführt. Um

SQL Anweisungen auszuführen muss eine Verbindung zur Datenbank bestehen, weshalb einer der Befehle ein connect Befehl sein kann.

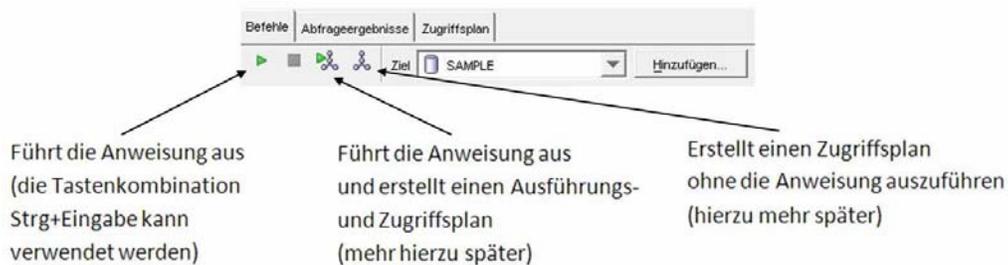


Abbildung 5.8 – Der Befehlseditor – Befehle Registerkarte

Starten des Befehlseditors

Sie können den Befehlseditor über verschiedene Wege starten:

- ▶ Über das Windows Start Menü:
Start -> Programme -> IBM DB2 -> DB2COPY1 (Standard) -> Befehlszeilentools -> Befehlseditor
- ▶ Durch Eingabe von db2ce auf der Befehlszeile
- ▶ Über das Tools Menü in der Steuerzentrale
- ▶ In die Steuerzentrale eingebettet
 - Im Objektsicht Teilfenster der Steuerzentrale klicken Sie rechts auf das Symbol der SAMPLE Datenbank und wählen den *Abfragen* Menüpunkt
- ▶ Wenn immer ein abfragefähiges Objekt ausgewählt wurde (Datenbank, Tabelle usw.) können Sie den Befehlseditor durch anklicken der *Abfragen* Verknüpfung im unteren rechten Teilfenster starten
- ▶ In der Steuerzentrale klicken Sie auf das Befehlseditor Symbol  in der Werkzeugleiste, wie in Abbildung 5.9 gezeigt



Abbildung 5.9 – Das Befehlseditor Symbol in der Steuerzentrale

Verbindung zu einer Datenbank hinzufügen

Um eine Verbindung zu einer Datenbank hinzuzufügen klicken Sie auf die Hinzufügen Schaltfläche (siehe Abbildung 5.7). Ein Dialog, wie in Abbildung 5.10 gezeigt, erscheint.

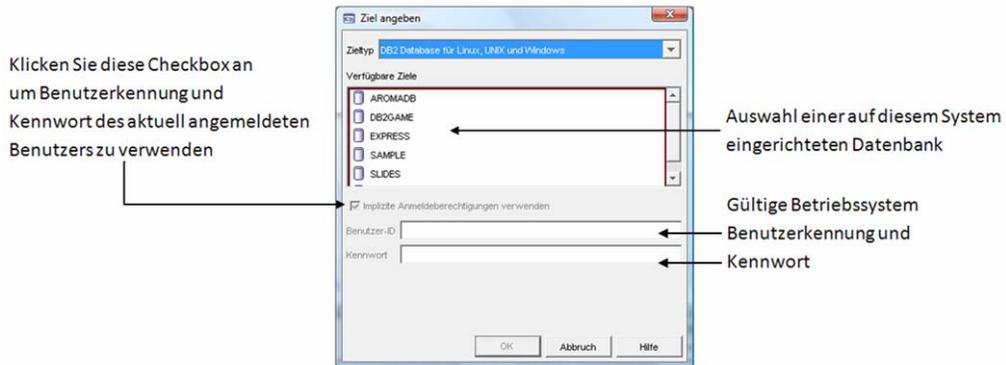


Abbildung 5.10 – Datenbankverbindung hinzufügen

5.3 SQL Assist Assistent

Falls Sie mit der SQL Abfragesprache nicht vertraut sind und einen Assistenten für die Generierung des SQL Codes verwenden möchten, kann Ihnen der über den Befehlseditor aufrufbare SQL Assist Assistent helfen. Wie in Abbildung 5.11 gezeigt rufen Sie ihn über den Befehlseditor auf, indem Sie auf das letzte Symbol in der Werkzeugleiste (in der Abbildung mit einem roten Kreis gekennzeichnet) klicken.

Der Assistent SQL Assist
kann auch über den Menüpunkt
Ausgewählt aufgerufen werden

Das Symbol des SQL Assist
Assistenten

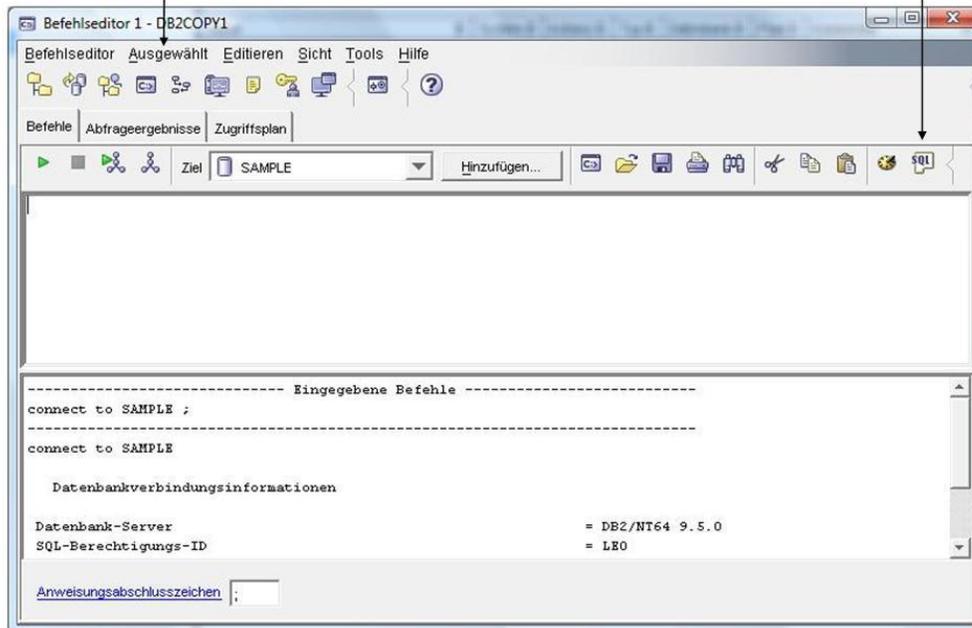


Abbildung 5.11 – Aufruf des Assistenten SQL Assist

Abbildung 5.12 zeigt den SQL Assist Assistenten. Er ist recht einfach zu verwenden. Zuerst geben Sie die Anweisungsart (SELECT, INSERT, UPDATE, DELETE), für welche sie Hilfe benötigen, an. Abhängig von der gewählten Anweisungsart werden unterschiedliche Optionen angezeigt. Im unteren Teilfenster können Sie sehen wie bei Auswahl unterschiedlicher Möglichkeiten des Assistenten die SQL Anweisung konstruiert wird.

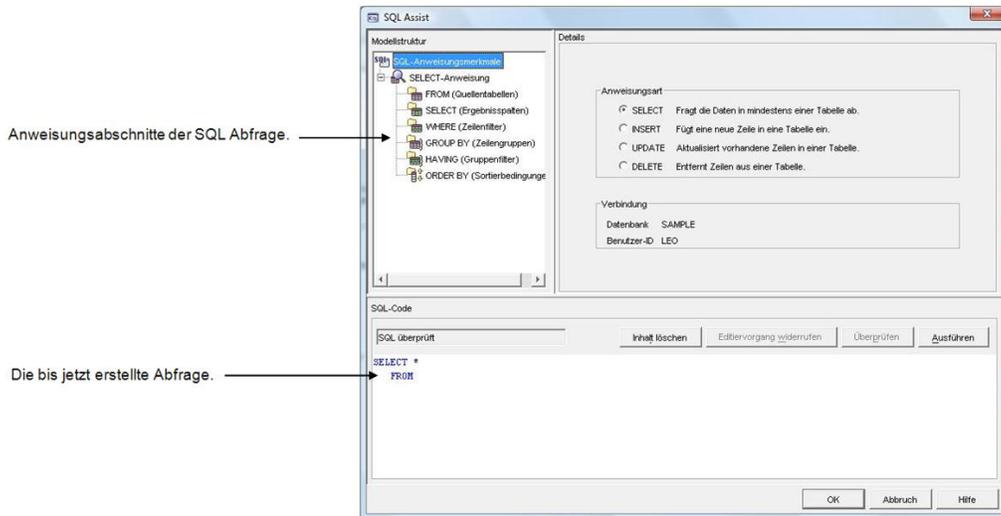


Abbildung 5.12 – Der SQL Assist Assistent

5.4 SQL anzeigen Schaltfläche

Die meisten grafischen DB2 Dienstprogramme und DB2 Assistenten erlauben es Ihnen den tatsächlichen Befehl oder die SQL Anweisung, generiert als Resultat eines Dienstprogramms oder Assistenten, zu prüfen. Um dies zu sehen klicken Sie, wie in Abbildung 5.13 und Abbildung 5.14 gezeigt, auf die SQL anzeigen Schaltfläche des Dienstprogramms mit welchem sie arbeiten.



Abbildung 5.13 – Die SQL anzeigen Schaltfläche

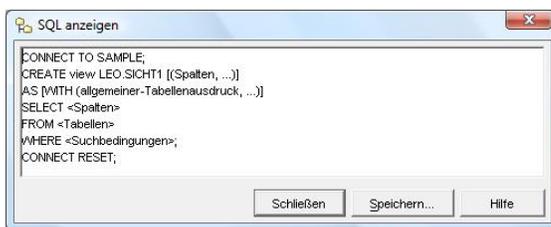


Abbildung 5.14 – Die Ausgabe einer SQL anzeigen Schaltfläche

Die Möglichkeit SQL Anweisungen und Befehle zu prüfen ist sehr hilfreich beim Erlernen der SQL Syntax oder zur Speicherung der Befehle oder Anweisungen zu späteren Wiederverwendung. Die generierten Befehle und Anweisungen können Sie auch zum Aufbau von Skripten verwenden.

Übung 4: Die EXPRESS Datenbank mit Skripten füllen

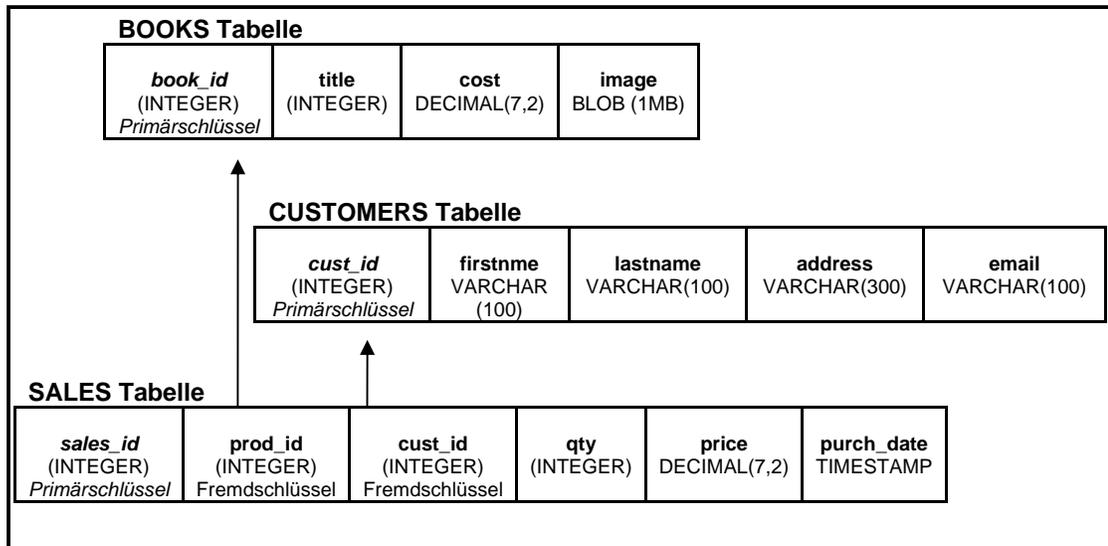
Ziel

In dieser Übung werden Sie die EXPRESS Datenbank mit dem Befehlseditor und zwei bereitgestellten Skripten füllen.

Vorgehensweise

1. In Übung 2 erstellten Sie die EXPRESS Datenbank. Nun müssen Sie diese nur noch mit einigen Tabellen und ein paar Daten füllen. Um Ihnen die Arbeit zu erleichtern wurden zwei Skripts (`quicklab4.db2` und `quicklab4.dat`) erstellt, welche das für sie erledigen. Das `quicklab4.db2` Skript enthält die für die Erstellung der Tabellen benötigten Befehle und muss aus diesem Grund zuerst ausgeführt werden. Das `quicklab4.dat` Skript enthält die zum Einfügen von Daten in die Tabellen erforderlichen Anweisungen. Beide Skripte in der dem Buch beiliegenden `quicklabs` zip Datei. Um die Skripte auszuführen öffnen Sie den Befehlseditor. Stellen Sie sicher, dass die neue Datenbank als Ziel im Kombinationsfeld der Werkzeugleiste angegeben ist. Falls die neue Datenbank nicht angezeigt wird, fügen Sie eine Verbindung zu dieser mittels der *Hinzufügen* Schaltfläche hinzu.
2. Klicken Sie in der Menüleiste des Befehlseditors auf *Ausgewählt* → *Öffnen* und navigieren zu dem Ordner in welchem die Skripte gespeichert sind. Wählen Sie die Datei `quicklab4.db2` und klicken auf die *OK* Schaltfläche. Der Dateiinhalt sollte jetzt im Eingabefenster des Befehlseditors angezeigt werden. Klicken Sie auf die *Ausführen* Schaltfläche um das Skript auszuführen. Stellen Sie sicher, dass während der Abarbeitung des Skripts keine Fehler auftraten.
3. Wiederholen Sie Schritt 2 für die Datei `quicklab4.dat`.

Die von Ihnen neu erstellte Datenbank verkörpert einen Internet Buchversand. Die *BOOKS* Tabelle enthält alle Informationen über die im Buchversand vorrätigen Bücher. Die *CUSTOMERS* Tabelle enthält Informationen über alle Kunden des Geschäfts. Die *SALES* Tabelle enthält, zu guter Letzt, Verkaufsdaten. Sobald ein Kunde ein Buch kauft wird ein Eintrag in die *SALES* Tabelle vorgenommen. Das untenstehende Diagramm zeigt das Design und die Beziehungen zwischen den Tabellen.



5.5 Scripting

Es ist immer hilfreich Skriptdateien, welche mehrere DB2 Befehle oder SQL Anweisungen wiederholt ausführen, zu erstellen. Ein DBA möchte zum Beispiel, um die Zeilenzahl wichtiger Tabellen festzustellen, ein bestimmtes Skript jeden Tag ausführen.

Es gibt zwei allgemeine Arten von Skripten:

1. SQL Skripte
2. Betriebssystem (Shell) Skripte.

5.5.1 SQL Skripte

SQL Skripte beinhalten Abfrageanweisungen und Datenbankbefehle. Diese Skripte sind relativ einfach zu verstehen und Plattformunabhängig. Wie dem auch sei, Variable oder Parameter werden nicht unterstützt.

So sind beispielsweise die nachfolgenden Befehle in der Datei script1.db2 gespeichert

```
CONNECT TO EXPRESS;
CREATE TABLE user1.mytable
    ( col1 INTEGER NOT NULL,
      col2 VARCHAR(40),
      col3 DECIMAL(9,2));
SELECT * FROM user1.mytable FETCH FIRST 10 ROWS ONLY;
COMMIT;
```

Datei script1.db2

Im obigen Skript handelt es sich bei allen Anweisungen um SQL Anweisungen und jede Anweisung durch ein Anweisungstrennzeichen, hier das Semikolon, getrennt. Die Dateierweiterung muss nicht "db2" lauten. Jede Erweiterung kann verwendet werden.

SQL Skripte ausführen

Ein SQL Skript kann entweder über die Befehlszeile oder das DB2 Befehlsfenster unter Windows oder die Shell unter Linux ausgeführt werden. Um das vorhergehende Skript über das DB2 Befehlsfenster oder die Linux Shell auszuführen können Sie den nachfolgenden Befehl verwenden:

```
db2 -t -v -f script1.db2 -z script1.log
```

oder:

```
db2 -tvf script1.db2 -z script1.log
```

In diesem Befehl:

- t gibt an, dass Anweisungen das standardmäßige Anweisungstrennzeichen (Semikolon) verwenden
- v gibt an, dass der verbose Modus verwendet wird. Dieser bewirkt das db2 alle ausgeführten Befehle anzeigt

- f gibt an, das es sich bei dem auf diesen Schalter folgenden Dateinamen um die Skriptdatei handelt.
- z gibt an, dass alle Bildschirmausgaben in der angegebenen Datei für die spätere Analyse gespeichert werden (dies ist optional, wird jedoch empfohlen)

Wenn der -t Schalter verwendet wird und kein Anweisungstrennzeichen spezifiziert wurde, wird das Semikolon als Anweisungstrennzeichen verwendet. Es könnte Situationen geben in welchen ein anderes Anweisungstrennzeichen erforderlich ist. So benötigt zum Beispiel SQL PL Code beinhaltendes Skript ein anderes, vom standardmäßigen Anweisungstrennzeichen (Semikolon) abweichendes, Anweisungstrennzeichen, da Semikola innerhalb der SQL PL Objektdefinitionen verwendet werden um Prozeduranweisungen zu trennen.

So sehen Sie zum Beispiel unten an der "functions.db2" benannten Skriptdatei, welche die erforderliche Datendefinitionssprache (DDL) zur Erstellung einer Funktion enthält, das innerhalb der Funktion ein Semikolon erforderlich ist um das Ende der SELECT Syntax anzuzeigen. Um die CREATE FUNCTION Anweisung zu terminieren verwenden wir das Ausrufezeichen (!). Hätten wir ein Semikolon als Anweisungstrennzeichen benutzt würde dies zu einem in eine DB2 Fehlermeldung mündenden Run-time Konflikt führen.

```
CREATE FUNCTION f1()
  SELECT ... ;
```

END!

Datei functions.db2

Um DB2 darüber zu informieren das ein anderes Anweisungstrennzeichen verwenden soll wird, wie unten gezeigt, der -d Schalter, gefolgt vom gewünschten Anweisungstrennzeichen, verwendet:

```
db2 -td! -v -f functions.db2 -z functions.log
```

Eine Beschreibung der verwendbaren Schalter kann über das Befehlsfenster oder die Linux Shell mit diesem Befehl angezeigt werden:

```
db2 list command options
```

5.5.2 Betriebssystem (Shell) Skripte

Betriebssystemskripte ermöglichen, da diese die Möglichkeit bieten zusätzliche Programmlogik zu integrieren, höhere Flexibilität und Leistung. Sie sind Plattformabhängig, unterstützen jedoch Parameter und Variable. Unten finden Sie ein Beispiel eines einfachen Windows Betriebssystem (Shell) Skripts.

```
set DBPATH=c:
set DBNAME=PRODEXPR
set MEMORY=25
db2 CREATE DATABASE %DBNAME% ON %DBPATH% AUTOCONFIGURE USING
```

```
MEM_PERCENT %MEMORY% APPLY DB AND DBM
db2 CONNECT TO %DBNAME% USER %1 USING %2
del schema.log triggers.log app_objects.log
db2 set schema user1
db2 -t -v -f schema.db2 -z schema.log
db2 -td@ -v -f triggers.db2 -z triggers.log
db2 -td@ -v -f functions.db2 -z functions.log
```

Datei create_database.bat

Um dieses Betriebssystem Skript über die Befehlszeile auszuführen würden sie unter Windows den folgenden Befehl absetzen:

```
create_database.bat db2admin ibmdb2
```

Unter Windows teilt die "bat" Dateierweiterung dem Betriebssystem mit das es sich um eine ausführbare Stapelverarbeitungsdatei handelt.

Unter Linux müssen Sie, zum Beispiel mit einem Befehl wie `chmod +x`, den Dateimodus der Datei dahingehend ändern, dass als Dateimodus ausführbar (executable) verwendet wird. Anschließend kann das Skript wie oben ausgeführt werden.

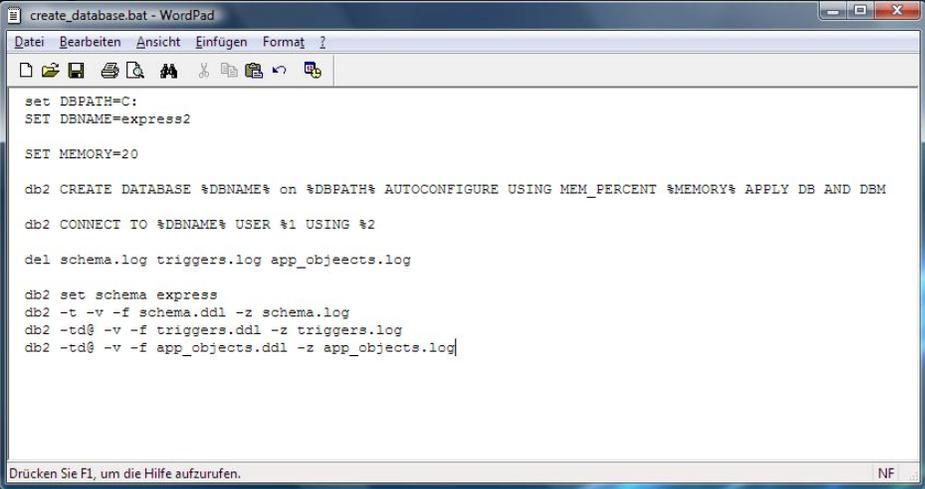
Übung 5: Erstellen eines Installationskripts für die EXPRESS Datenbank

Ziel

Bei Skripten handelt es sich um mächtige Mechanismen um sich wiederholende Vorgänge, wie das Sammeln von Datenbankstatistiken, das Durchführen von Sicherungen und die Datenbankimplementierung, auszuführen. Betriebssystemskripte haben den Vorteil, dass sie Parameter unterstützen und dadurch flexibler sind. In dieser Übung werden Sie ein Betriebssystemskript zur Implementierung der *EXPRESS* Datenbank als *EXPRESS2* Datenbank erstellen. Das Skript wird das zuvor erstellte SQL Skript für Datenbankobjekte aufrufen. Um Platz zu sparen werden in dieser Übung ausschließlich Skripte und Befehle für die Windows Plattform verwendet. Falls Sie die Arbeit mit Linux vorziehen müssen sie die untenstehenden Anweisungen entsprechend ändern.

Vorgehensweise

1. Öffnen Sie einen Texteditor wie Editor oder WordPad und geben die unten gezeigte Information ein.



```

set DBPATH=C:
SET DBNAME=express2

SET MEMORY=20

db2 CREATE DATABASE %DBNAME% on %DBPATH% AUTOCONFIGURE USING MEM_PERCENT %MEMORY% APPLY DB AND DBM

db2 CONNECT TO %DBNAME% USER %1 USING %2

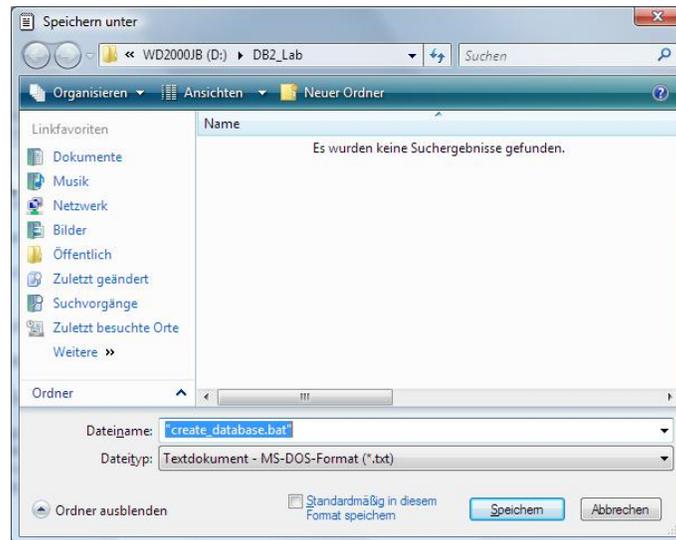
del schema.log triggers.log app_objects.log

db2 set schema express
db2 -t -v -f schema.ddl -z schema.log
db2 -td@ -v -f triggers.ddl -z triggers.log
db2 -td@ -v -f app_objects.ddl -z app_objects.log

```

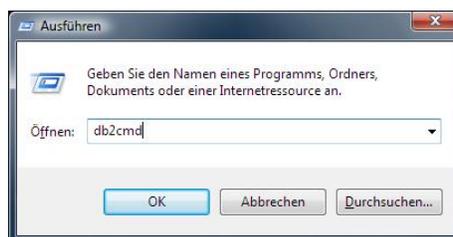
- 2.
3. Speichern Sie die Skriptdatei unter dem Namen `create_database.bat` in ein Verzeichnis. Im *Speichern unter* Dialogfenster stellen Sie sicher, dass sie die *MS-DOS Format* Option (in WordPad) gewählt haben. Falls Sie die Datei in einem anderen Format speichern fügt WordPad möglicherweise unsichtbare Zeichen ein, welche zu Problemen bei der Ausführung des Skripts führen können. Um sicherzustellen das Windows nicht die Dateierweiterung `.TXT` anhängt, schließen Sie zusätzlich, wie in

der Abbildung unten gezeigt, den Dateinamen in doppelte Anführungszeichen ein.



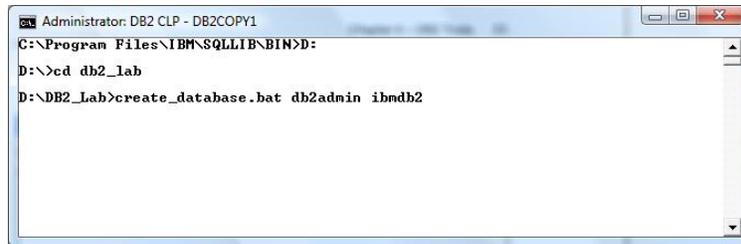
- Um mit DB2 in Interaktion tretende Skripts auszuführen benötigen Sie eine DB2 Befehlszeilenumgebung. Um ein DB2 Befehlsfenster zu öffnen gehen Sie nach *Start > Programme > IBM DB2 > DB2COPY1 (Standard) > Befehlszeilentools > Befehlsfenster*.

Alternativ können Sie, wie unten gezeigt, *Start > Ausführen* verwenden, geben `db2cmd` ein und betätigen die Eingabetaste.



- Um abschließend das Skript auszuführen geben Sie diese Befehle ein:

```
cd C:\express
create_database.bat db2admin ibmdb2
```



```
Administrator: DB2 CLP - DB2COPY1
C:\Program Files\IBM\SQLLIB\BIN>D:
D:\>cd db2_lab
D:\DB2_Lab>create_database.bat db2admin ibmdb2
```

6. Nehmen Sie sich etwas Zeit um sich mit dem soeben von ihnen erstellten Skript vertraut zu machen. Verstehen Sie was in jeder Zeile passiert?
7. Versuchen Sie die folgenden Fragen zu beantworten:
 - a. Wo wird die Verbindung zur Datenbank aufgebaut?
 - b. Was bedeuten %1 und %2?
 - c. Was macht die folgende Codezeile? Wo wird sie verwendet? Für was?
SET DBPATH=C:
 - d. Was bewirkt die folgende Codezeile?
del schema.log, triggers.log, app_objects.log
 - e. Was passiert wenn das Skript ohne Parameter aufgerufen wird?
 - f. Warum beinhalten die aufgerufenen SQL Skripte keine CONNECT TO Anweisungen? Wie verbinden sich die Skripte mit der Datenbank?

5.6 Taskzentrale

Das grafische Taskzentrale Dienstprogramm ermöglicht Ihnen das Anlegen von Tasks: ein Satz von Anweisungen, wie die Ausführung von DB2 Befehlen, Betriebssystembefehlen oder Skripten. Nachfolgende Aktionen können, abhängig vom Erfolg des Tasks, ausgeführt werden. Sollte zum Beispiel ein Task, welches eine wichtige Datenbank morgens um 3:00 Uhr sichert, fehlerfrei ausgeführt worden sein, kann eine diese Information beinhaltende E-Mail an den DBA gesandt werden. Falls jedoch die Sicherung fehlschlägt kann die Taskzentrale dem DBA eine Pager Nachricht senden. Abbildung 5.15 zeigt die Taskzentrale.

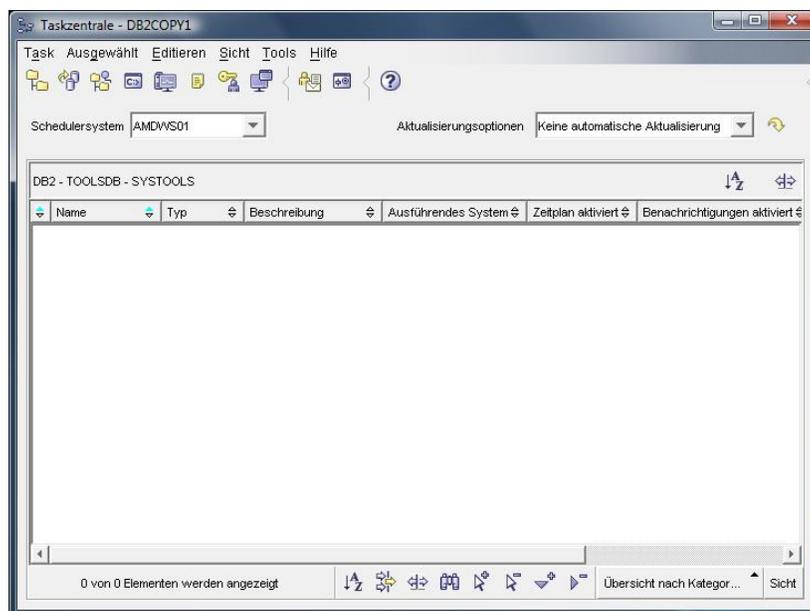


Abbildung 5.15 – Die Taskzentrale

5.6.1 Die Toolskatalog Datenbank

Alle Details über Ihre Tasks und Taskpläne werden in einer separaten, Toolskatalog genannten, DB2 Datenbank gespeichert. Um Tasks zu planen muss diese Datenbank bereits existieren. Um eine Toolskatalog Datenbank zu erstellen können Sie diesen Befehl verwenden:

```
CREATE TOOLS CATALOG systools CREATE NEW DATABASE toolsdb
```

Im obigen Beispiel ist `systools` das für alle Tabellen in der Datenbank verwendete Schema und der Name der Datenbank lautet `toolsdb`. In Kapitel 8, Arbeiten mit Datenbankobjekten, werden wir näher auf das Schema Thema eingehen.

Starten der Taskzentrale

Sie können die Taskzentrale über die Steuerzentrale starten, indem sie das *Tools* Menü öffnen und *Taskzentrale* wählen, wie in Abbildung 5.16 gezeigt. Alternativ können Sie dieses Dienstprogramm über das Windows Start Menü starten: *Start > Programme > IBM DB2 > DB2COPY1 (Standard) > Allgemeine Verwaltungstools > Taskzentrale*.

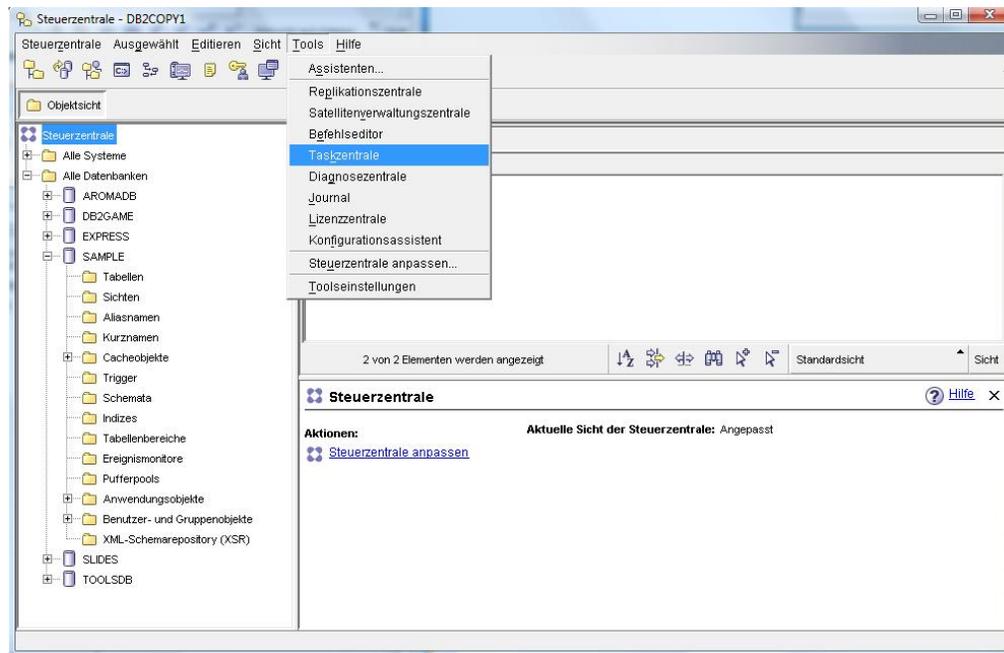


Abbildung 5.16 – Starten der Taskzentrale

Planen mit der Taskzentrale

Jede Art von Skript (ob mit einem grafischen DB2 Dienstprogramm erstellt oder nicht) kann mit der Taskzentrale als Aufgabe geplant werden. Aufgaben (Tasks) laufen zur vorgesehenen Zeit auf dem System, auf welchem Sie den Toolskatalog erstellt haben. Wir regen an, dass Sie die Taskzentrale selber erforschen. Das Anlegen eines neuen Tasks ist unkompliziert.

5.7 Journal

Das grafische DB2 Journal Dienstprogramm stellt einem DBA ein Journal der Aktivitäten in Online Form zur Verfügung. Abbildung 5.17 zeigt das Journal und Tabelle 5.2 die vom Journal zur Verfügung gestellte Information.

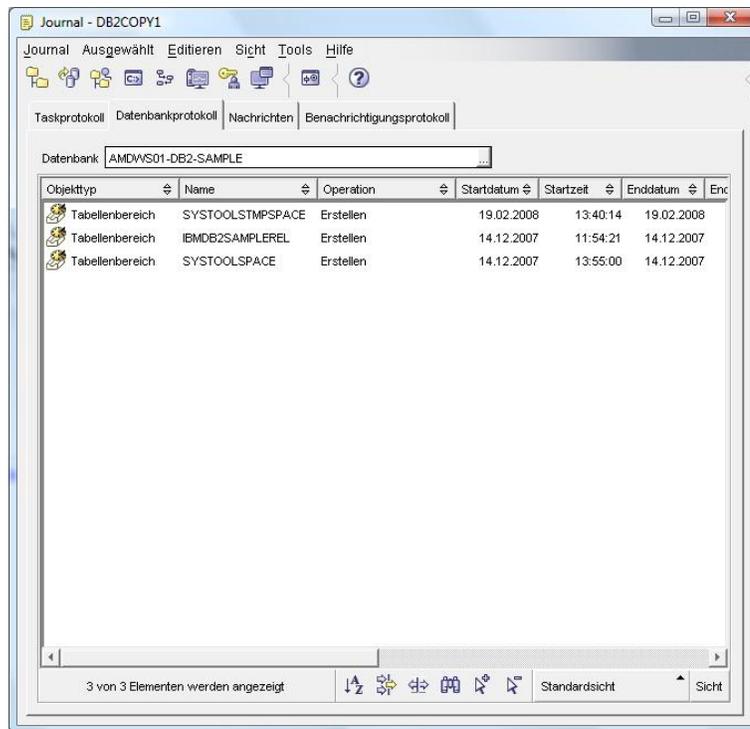


Abbildung 5.17 – Das Journal

Art der Information	Beschreibung
Taskprotokoll	Alle Versuche geplante Tasks auszuführen und deren Erfolgsstatus.
Datenbankprotokoll	Eine Aufzeichnung aller durchgeführten Datenbankaktivitäten (Sicherung, Wiederherstellung, Reorganisation usw.).
Nachrichten	Historie der von DB2 Dienstprogrammen zurückgegebenen Meldungen. Dies ist hilfreich wenn Sie alte Meldungen aufrufen und vergleichen wollen oder ein Dialog zu schnell oder aus Versehen geschlossen wurde.
Benachrichtigungsprotokoll	Enthält System-Level Nachrichten. Kritische Fehler werden hier aufgezeichnet.

Tabelle 5.2 – Vom Journal zur Verfügung gestellte Informationen

Das Journal starten

Sie können das Journal über die Steuerzentrale starten, indem sie das *Tools* Menü öffnen und *Journal* wählen, wie in Abbildung 5.18 gezeigt. Alternativ können Sie dieses Dienstprogramm über das Windows Start Menü starten: *Start > Programme > IBM DB2 > DB2COPY1 (Standard) > Allgemeine Verwaltungstools > Journal*.

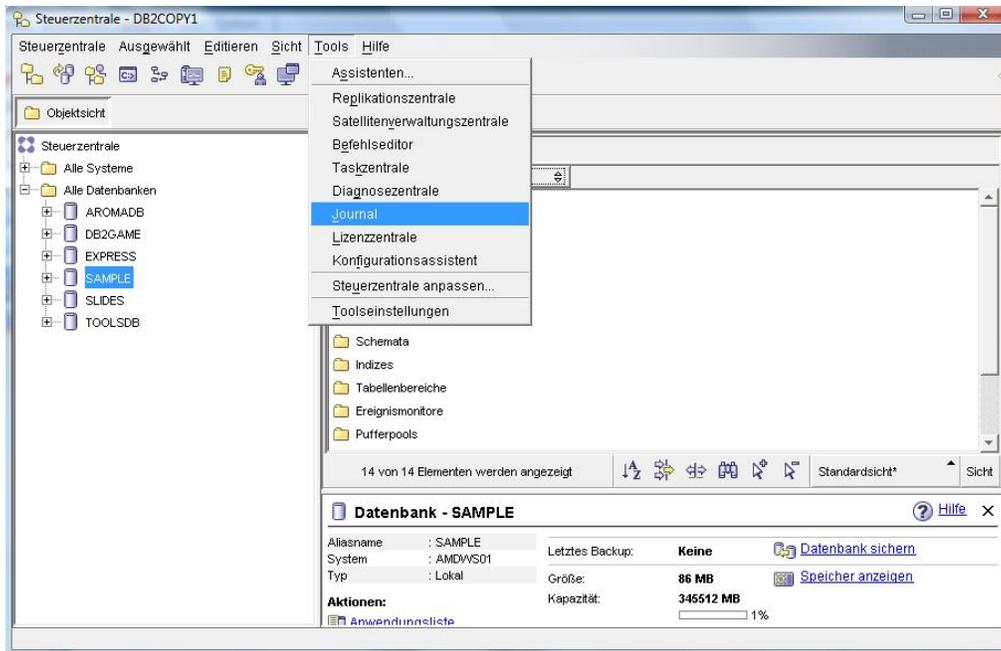


Abbildung 5.18 – Das Journal starten

5.8 Diagnosezentrale

Die Diagnosezentrale ist ein innerhalb des DB2 Prozesses laufender Agent, welcher alle Aspekte des Datenbankzustandes (Speicher, Speicherplatzmanagement, zuvor definierte automatische Aktivitäten usw.) überwacht. Sobald ein Aspekt von DB2 nicht innerhalb der gesetzten Parameter ist, wird eine Ausnahme ausgelöst und dem DBA eine Nachricht zugestellt. Es gibt drei Arten des Alarmstatus:

- ▶ Beliebiger Alarmstatus: ein nicht-normaler Zustand
- ▶ Alarm- oder Warnungsstatus: ein nicht-kritischer Zustand, bei welchem kein sofortiger Eingriff erforderlich ist, jedoch auf ein nicht-optimales System hinweisen könnte
- ▶ Alarmstatus: ein umgehendes Handeln verlangender kritischer Zustand

Die Diagnosezentrale kann mit dem Datenbankmanager Konfigurationsparameter HEALTH_MON ein- und ausgeschaltet werden.

5.8.1 Diagnosezentrale

Bei der Diagnosezentrale handelt es sich um ein mit dem Diagnosemonitor zusammenarbeitendes grafisches Dienstprogramm. Die Diagnosezentrale unterteilt Alerts auf einem System nach Instanz, Datenbank und Tabellenbereichen. Abbildung 5.19 zeigt die Diagnosezentrale.

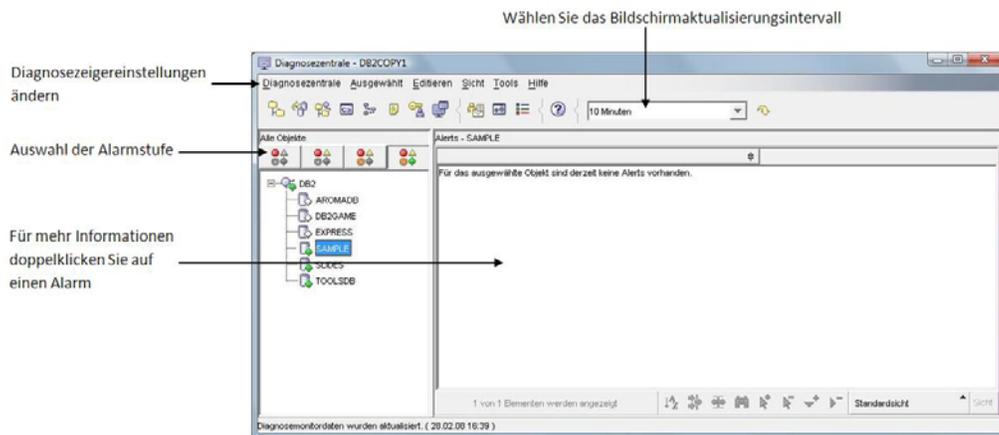


Abbildung 5.19 – Die Diagnosezentrale

Starten der Diagnosezentrale

Sie können die Diagnosezentrale über die Steuerzentrale starten, indem sie das *Tools* Menü öffnen und *Diagnosezentrale* wählen, wie in Abbildung 5.20 gezeigt. Alternativ können Sie dieses Dienstprogramm über das Windows Start Menü starten: *Start > Programme > IBM DB2 > DB2COPY1 (Standard) > Allgemeine Verwaltungstools > Diagnosezentrale*.

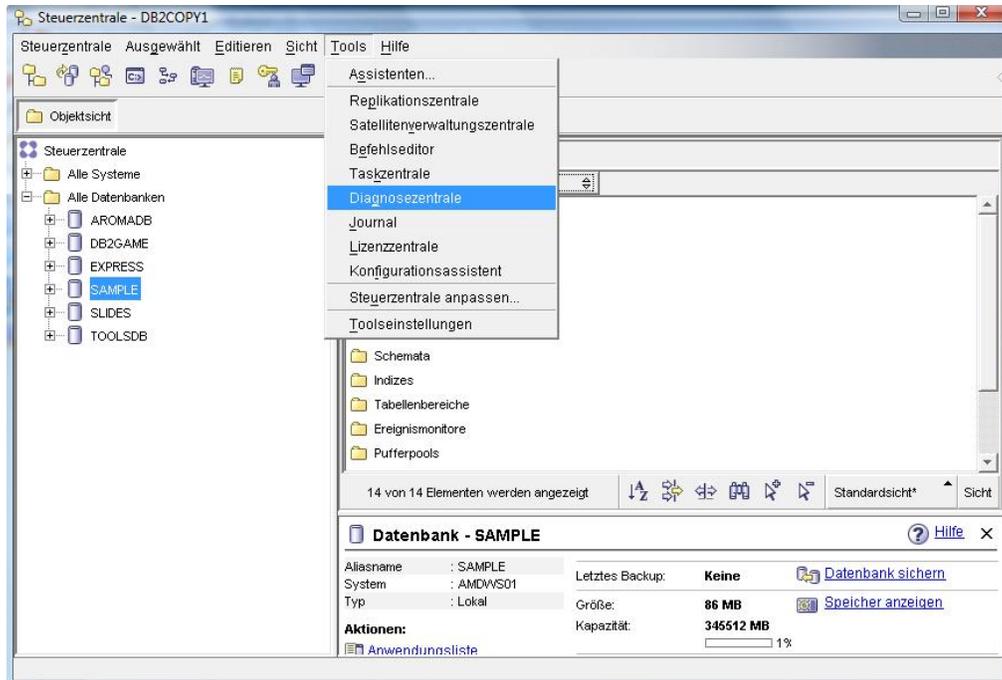


Abbildung 5.20 – Starten der Diagnosezentrale

Alertbenachrichtigung der Diagnosezentrale konfigurieren

Sobald die Diagnosezentrale gestartet wurde, können Sie die Alertbenachrichtigung konfigurieren. Öffnen Sie den *Diagnosezentrale* Menüpunkt, klicken auf *Konfigurieren* und wählen, wie in Abbildung 5.21 gezeigt *Alertbenachrichtigung*. Für den automatischen Versand von Nachrichten im Alarmfall, ermöglicht die Alertbenachrichtigung die Angabe von Ansprechpartnern mit E-Mail Adresse oder Page Nummer.

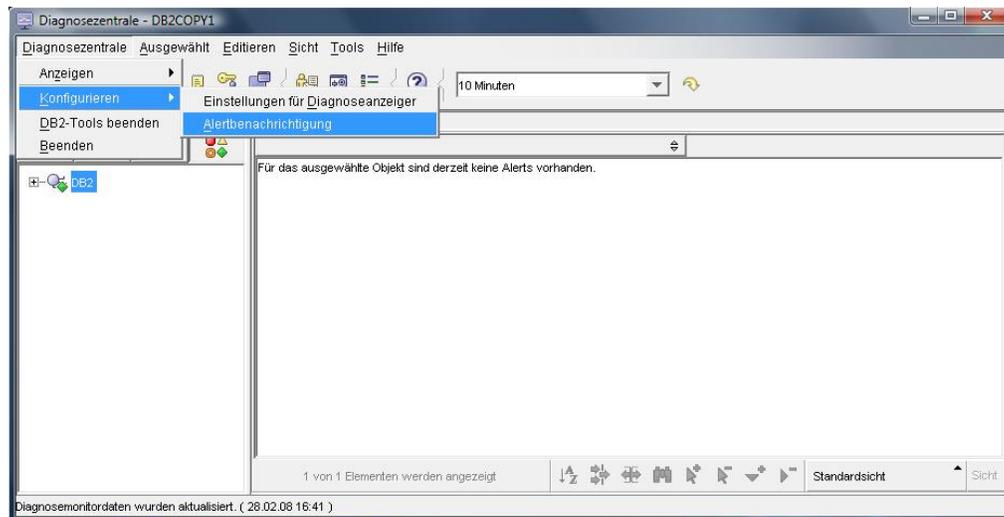


Abbildung 5.21 – Alertbenachrichtigung

**TEIL II – DB2 LERNEN:
DATENBANK ADMINISTRATION**

6

Kapitel 6 – DB2 Architektur

In diesem Kapitel stellen wir Ihnen die DB2 Architektur vor:

- Das DB2 Prozess Modell
- Das DB2 Arbeitsspeichermodell
- Das DB2 Datenspeicherungsmodell

Hinweis:

Für weitere Informationen zur DB2 Architektur können Sie sich dieses Video ansehen:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4482>

6.1 DB2 Prozess Modell

Abbildung 6.1 stellt das DB2 Prozess Modell dar. In dieser Abbildung repräsentieren Rechtecke Dienste wohingegen Ellipsen Threads darstellen. Der hauptsächliche DB2 Dienst heißt db2sysc. Unter diesem Dienst gibt es mehrere Threads, wovon der wichtigste ebenfalls db2sysc heißt. Hierbei handelt es sich um den wichtigsten Thread, welcher die anderen Threads erzeugt. Wenn eine Anwendung versucht sich mit dem Server mit einer SQL Anweisung per Fernzugriff zu verbinden, werden die Remote Listener des Kommunikationsprotokolls die Anfrage empfangen und den DB2 Coordinator Agenten (db2agent) kontaktieren. Ein DB2 Agent ist wie ein kleiner Arbeiter, welcher im Auftrag von DB2 Arbeiten ausführt. Wenn es sich um eine lokale Anwendung handelt, d.h. sie läuft auf demselben Server wie DB2, sind die Schritte sehr ähnlich, nur das der db2ipccm Agent die Anfrage statt des db2tcpdm Threads handhabt. In einigen Fällen, wie bei aktivierter Parallelität, erzeugt ein db2agent andere Agenten, welche als db2agntp Threads in Erscheinung treten. Andere in der Abbildung gezeigte Agenten, wie db2pfchr, db2loggr, db2dlock werden für andere Zwecke verwendet. Die gebräuchlichsten Dienste sind in Tabelle 6.1 beschrieben, die gebräuchlichsten Threads in Tabelle 6.2.

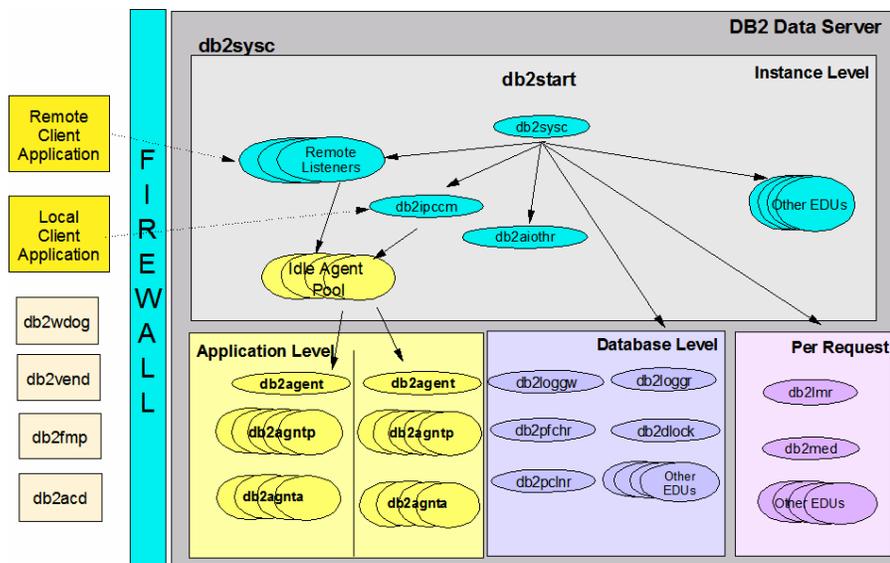


Abbildung 6.1 – Das DB2 Prozess Modell

Name des Dienstes	Beschreibung
db2sysc (Linux) db2syscs (Win)	Der Hauptprozess DB2 System-Controller oder -Engine. In DB2 9.5 existiert nur ein multi-threaded Hauptprozess je Partition. Alle Kontrollinstanzen (Engine Dispatchable Units (EDUs)) sind Threads innerhalb des Prozesses. Ohne diesen Prozess kann der Datenbankserver nicht funktionieren.
db2acd	Der Autonomic Computing Daemon. Er wird auf Seite des Clients für die Ausführung automatisierter Aufgaben, wie die Diagnosezentrale, automatische Wartungsdienstprogramme und dem Admin Scheduler verwendet. Dieser Prozess wurde früher <i>db2hmon</i> genannt.
db2wdog	Der DB2 watch dog. Der watch dog ist der Parent des wichtigsten Dienstes der Engine, db2sysc. Er bereinigt Ressourcen, falls der db2sysc Dienst abnormal endet.
db2vend	Der in DB2 9.5 eingeführte Dienst für abgeschirmte Programme. Der gesamte Code von Drittanbietern läuft unter diesem Dienst außerhalb der Engine. Bei Code von Drittanbietern handelt es sich um nicht-IBM Programme, welche mit DB2 zusammenarbeiten. So kann zum Beispiel die Protokollarchivierung von einem Programm eines Drittanbieters verwaltet werden indem eine <i>User Exit Routine</i> verwendet wird, welche auf diesen Code zeigt.
db2fmp	Abgeschirmte Prozesse, welche Benutzercode auf dem Server unter Umgehung der Firewall, sowohl für gespeicherte Prozeduren als auch benutzerdefinierte Funktionen, ausführt. Dieser Prozess ersetzt die

	beiden <i>db2udf</i> und <i>db2dari</i> Prozesse, welche in vorherigen DB2 Versionen verwendet wurden.
--	--

Tabelle 6.1 – Gebräuchliche DB2 Dienste

Name des Threads	Beschreibung
db2sysc	Der System Controller Thread. Dieser Thread ist verantwortlich für den Start, die Beendigung und Verwaltung der laufenden Instanz
db2tcpcm	TCP/IP Communication Listener
db2agent	DB2 Coordinator Agent, welcher Datenbankoperationen im Auftrag von Anwendungen ausführt (mindestens 1 je Verbindung, abhängig davon ob ein Connection Concentrator aktiviert ist).
db2agntp	Aktiver Sub-Agent, wenn der Instanzparameter INTRA_PARALLEL auf YES gesetzt ist. Der Thread <i>db2agent</i> koordiniert die Arbeit zwischen den unterschiedlichen <i>db2agntp</i> Sub-Agenten.
db2pfchr	Asynchroner DB2 I/O Data Prefetcher (NUM_IOSERVERS)
db2pclnr	Asynchroner DB2 I/O Data Writer (NUM_IOCLEANERS)

Tabelle 6.2 – Gebräuchliche DB2 Threads

6.2 DB2 Arbeitsspeichermodell

Das DB2 Arbeitsspeichermodell besteht aus verschiedenen Bereichen im Arbeitsspeicher auf der Stufe der Instanz, der Datenbank und der Anwendung und des Agenten, wie in Abbildung 6.2 gezeigt. Wir werden die verschiedenen Bereiche des Arbeitsspeichers in diesem Buch nicht detailliert behandeln, sondern nur einen kurzen Überblick geben.

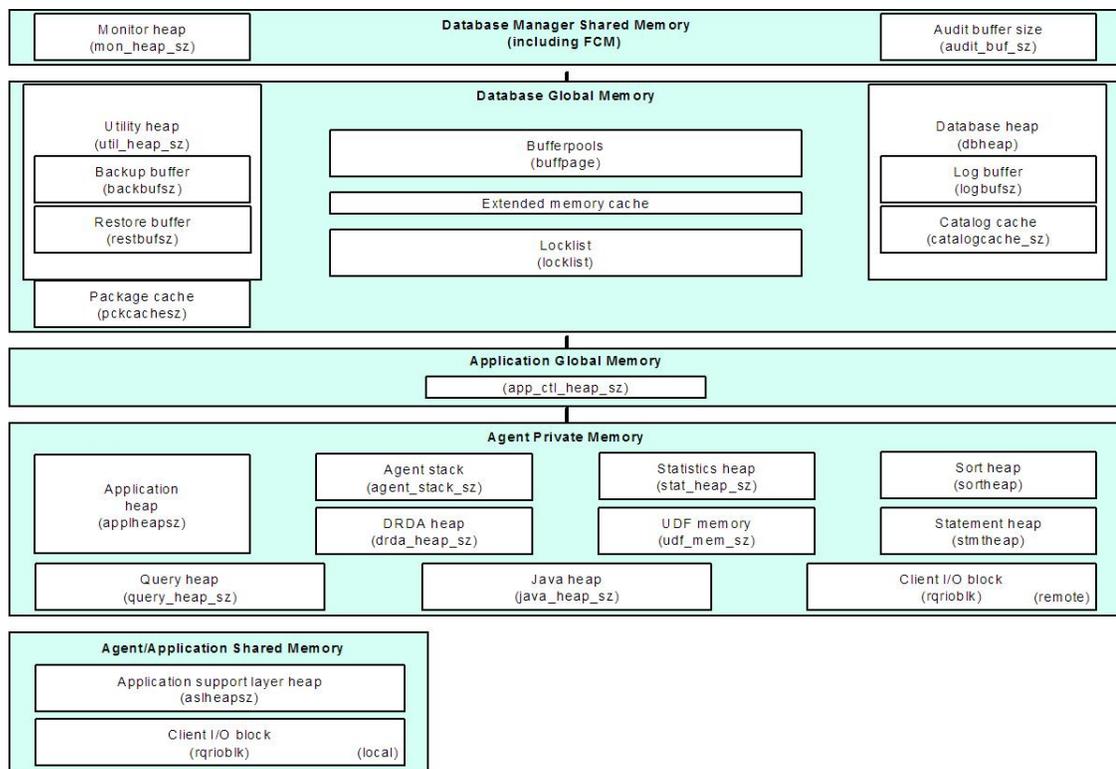


Abbildung 6.2 – Das DB2 Arbeitsspeichermodell

Sobald eine DB2 Instanz gestartet wird, wird dem Datenbankmanager geteilter Speicher (shared Memory) zugewiesen. Dieser Speicherbereich ist normalerweise sehr klein. Wenn Sie (oder eine Anwendung) sich das erste Mal mit einer Datenbank verbinden, wird der Datenbank globaler Speicher (global memory) zugewiesen. In diesem Block ist der Pufferpool einer der wichtigsten Bestandteile, insbesondere zur Verbesserung der Leistung von Abfragen. Die Größe des Pufferpools legt fest wie groß der gesamte globale Speicher der Datenbank sein wird.

Beim Agent Private Memory handelt es sich um den von jedem DB2 Agenten verwendeten Arbeitsspeicher. Ohne die Verwendung eines Connection Concentrators benötigt jede Verbindung einen Agenten. Üblicherweise kann ein Agent ungefähr 4 bis 5 MB verbrauchen. Mit Connection Concentrator können mehrere Verbindungen einen Agenten verwenden, was zu einer Reduzierung des erforderlichen Speicherbedarfs führt.

6.3 DB2 Datenspeicherungsmodell

In diesem Abschnitt beschreiben wir die folgenden Konzepte:

- Pages und Extents
- Pufferpool

- Tabellenbereich

6.3.1 Pages und Extents

Eine Seite (Page) ist die kleinste Speichereinheit in DB2. Zulässige Größen für Pages sind: 4K, 8K, 16K und 32K. Bei einem Extend handelt es sich um eine Gruppierung von Pages. Die Arbeit mit nur einer einzigen Seite in DB2 würde die Performance negativ beeinflussen und erklärt, warum DB2 stattdessen mit Extends arbeitet. Die Größe der Page und des Extends werden, wie wir in den nächsten Abschnitten sehen werden, bei der Arbeit mit Pufferpools und Tabellenbereichen definiert.

6.3.2 Pufferpools

Ein Pufferpool ist ein realer Cache im Arbeitsspeicher für Tabellen- und Indexdaten. Sie erhöhen die Performance indem sie die direkte Ein-/Ausgabe (I/O) reduzieren und das asynchrone Lesen (Zwischenspeichern (pre-fetching)) und Schreiben fördern. Dies bedeutet, dass DB2 präventiv Seiten liest und diese von der Festplatte in den Pufferpool für die Verwendung zwischenspeichert.

Pufferpools werden im Arbeitsspeicher in Einheiten von 4K, 8K, 16K und 32K Seiten angelegt. Es muss mindestens einen Pufferpool je Datenbank existieren und mindestens ein übereinstimmender Pufferpool für einen Tabellenbereich einer bestimmten Größe.

Erstellen eines Pufferpools

Um einen Pufferpool zu erstellen können Sie die `CREATE BUFFERPOOL` Anweisung verwenden. Alternativ können Sie in der Steuerzentrale rechts auf den Pufferpools Ordner innerhalb einer gegebenen Datenbank klicken und, wie in Abbildung 6.3 gezeigt, `Erstellen` wählen.

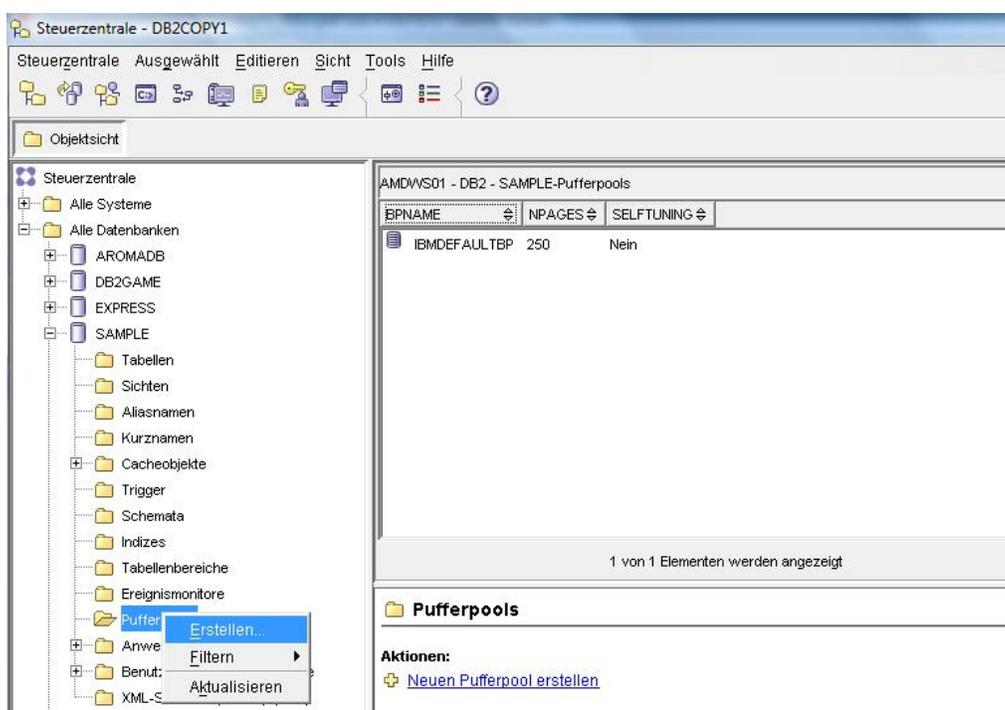


Abbildung 6.3 – Erstellen eines Pufferpools

Nach einem Klick auf Erstellen wird, wie in Abbildung 6.4 gezeigt, der Pufferpools erstellen Dialog angezeigt.

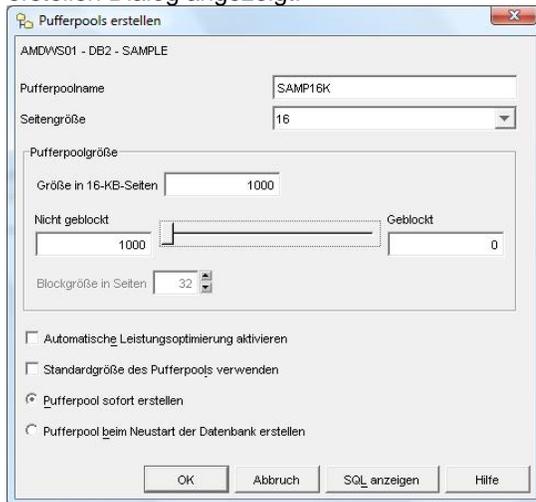


Abbildung 6.4 – Pufferpools erstellen Dialog

Die meisten Einträge in Abbildung 6.4 erklären sich selbst. Die Felder "Nicht geblockt" und "Geblockt" beziehen sich auf die Anzahl Pages (dt. Seiten), welche als nicht geblockt und

als geblockt existieren sollten. Blockbasierte Pufferpools stellen sicher, dass zusammenhängende Seiten der Festplatte auch zusammenhängend in den geblockten Bereich des Pufferpools kopiert werden. Dies kann die Performance erhöhen. Die Anzahl der Seiten darf nicht größer als 98 Prozent der Anzahl Seiten für den Pufferpool sein. Die Angabe von 0 deaktiviert geblockte Ein-/Ausgabe.

Sobald der Pufferpool erstellt wurde, wird er, wie in Abbildung 6.5 gezeigt, in der Steuerzentrale angezeigt.

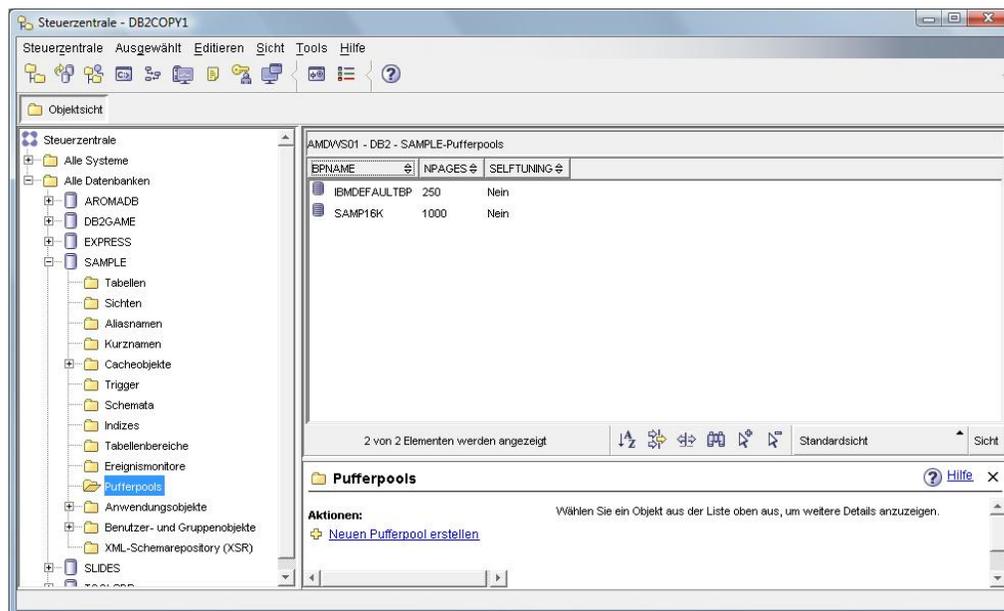


Abbildung 6.5 – Die Steuerzentrale nach der Erstellung des “SAMP16K” Pufferpools

6.3.3 Tabellenbereiche

Tabellenbereiche sind das logische Bindeglied zwischen logischen Tabellen und dem physikalischen Speicher des Systems (Pufferpool) und Containern (Festplatten).

Verwenden Sie die `CREATE TABLESPACE` Anweisung um einen Tabellenbereich anzulegen, wobei sie angeben können:

- Die Seitengröße (Page size) des Tabellenbereichs (4KB, 8KB, 16KB oder 32KB). Die Seitengröße muss mit einem Pufferpool der gleichen Größe übereinstimmen.
- Der Name der Pufferpools, welcher mit dem Tabellenbereich verbundenen ist.
- Eine Extend Größe.
- Eine Pre-Fetch Größe.

Arten von Tabellenbereichen

Es gibt drei Arten von Tabellenbereichen:

- Regulär

Diese werden für Benutzertabellen verwendet. So handelt es sich zum Beispiel bei dem standardmäßig erstellten Tabellenbereich USERSPACE1 um einen regulären Tabellenbereich.

- Groß
Diese werden optional verwendet um LOB Daten in ihrem eigenen Tabellenbereich zu speichern. Er wird für die Speicherung von XML Daten bei Datenbanken, welche mit pureXML Unterstützung generiert wurden, d.h. die Datenbank wurde als UNICODE generiert und verwendet den XML Datentyp für Spalten, verwendet. Große Tabellenbereiche sind der Standard.
- Temporär
Es gibt zwei Arten temporärer Tabellenbereiche:
 - ▶ Systemtemporär
Diese werden von DB2 für interne Vorgänge, wie das Sortieren, verwendet. So ist zum Beispiel der Tabellenbereich TEMPSPACE1, welcher immer angelegt wird wenn Sie eine Datenbank erstellen, ein systemtemporärer Tabellenbereich.
 - ▶ Benutzertemporär
Diese werden zur Erstellung benutzerdefinierter globaler temporärer Tabellen (temporäre Tabellen im Speicher) verwendet. Sie werden oft mit systemtemporären Tabellenbereichen verwechselt.

Speicherbereichsverwaltung

Tabellenbereiche können durch die Art der Verwaltung klassifiziert werden. Angegeben werden kann dies in der CREATE TABLESPACE Anweisung:

Verwaltet durch das System (SMS)

Dieser Tabellenbereichstyp ist auch als System Managed Storage (SMS) bekannt. Sie sind leicht zu verwaltet und Container sind Dateisystemverzeichnisse. Der Speicherplatz wird nicht vorbelegt (pre-allocated), jedoch können Dateien dynamisch in der Größe zunehmen. Sobald Sie die Container spezifiziert haben, sind diese zur Erstellungszeit festgelegt und andere Container können zu einem späteren Zeitpunkt nicht hinzugefügt werden, außer es wird eine umgeleitete Wiederherstellung (redirected restore) verwendet. Bei Verwendung von SMS Tabellenbereichen können die Tabellendaten, der Index und LOB Daten nicht über verschiedene Tabellenbereiche verteilt werden.

Verwaltet durch die Datenbank (DMS)

Dieser Tabellenbereichstyp ist auch als Database Managed Storage (DMS) bekannt. Dies bedeutet, dass DB2 die Speicherung verwaltet. Die Verwaltung des Speicherplatzes erfordert mehr manuelle Eingriffe durch einen DBA. Container können pre-allocated Dateien oder Raw Devices sein. Bei Raw Devices werden die Daten direkt, d.h. ohne Caching des Betriebssystems geschrieben.

Container können hinzugefügt, gelöscht oder in der Größe geändert werden. DMS Tabellenbereiche eignen sich zur Erzielung höchster Performance. Tabellendaten, Index und LOB Daten können in mehrere Tabellenbereiche aufgeteilt werden, was zu einer Erhöhung der Performance führt.

Automatische Verwaltung

Dieser Tabellenbereichstyp wird durch dynamischen Speicher (automatic storage) verwaltet und kann, ähnlich wie SMS Tabellenbereiche, von der leichten Verwendung profitieren, jedoch mit der besten Performance und Flexibilität von DMS Tabellenbereichen. Aus diesem Grund ist dieser Tabellenbereich, beginnend mit DB2 9, der Standardtyp für Tabellenbereiche. Für diesen Tabellenbereich spezifiziert ein Benutzer zuerst eine logische Gruppe von Massenspeichergeräten. Explizite Container Definitionen werden nicht vorgegeben. Container werden automatisch über die Speicherpfade erstellt. Die Vergrößerung existierender Container und die Hinzufügung neuer Container werden vollständig durch DB2 verwaltet.

Um dynamische Speicherung zu ermöglichen, muss eine Datenbank mit dynamischer Speicherung (hierbei handelt es sich um das Standardverhalten) erstellt und Speicherpfade definiert werden. Nach der Erstellung können, wenn gewünscht, die Speicherpfade unter Verwendung einer Datenbank RESTORE Operation neu definiert werden. Dann können Sie dynamische Speicherung verwendende Tabellenbereiche (auch hier handelt es sich um das Standardverhalten) erstellen.

Beispiel der dynamischen Speicherung

Zuerst erstellen Sie eine Datenbank mit aktivierter dynamischer Speicherung, wie in diesen Beispielen:

Dynamische Speicherung ist standardmäßig aktiviert:
`CREATE DATABASE DB1`

Dynamische Speicherung wird explizit angegeben:
`CREATE DATABASE DB1 AUTOMATIC STORAGE YES`

Dynamische Speicherung ist standardmäßig aktiviert, doch der Speicherpfad wird angegeben:
`CREATE DATABASE DB1 ON /data/path1, /data/path2`

Dynamische Speicherung ist explizit deaktiviert:
`CREATE DATABASE DB1 AUTOMATIC STORAGE NO`

Als nächstes erstellen Sie, wie in diesen Beispielen, die Tabellenbereiche mit aktivierter dynamischer Speicherung:

Dynamische Speicherung für Tabellenbereiche ist ebenfalls standardmäßig aktiviert:
`CREATE TEMPORARY TABLESPACE TEMPTS`

Dynamische Speicherung wird für den Tabellenbereich explizit angegeben:
`CREATE TABLESPACE TS2 MANAGED BY AUTOMATIC STORAGE`

Dynamische Speicherung wird implizit spezifiziert, die Anfangsgröße, unter Angabe der Zuwachsmenge und die maximale Größe, auf welche sie anwachsen kann, zugewiesen.

```
CREATE TABLESPACE TS1
  INITIALSIZE 500 K
  INCREASESIZE 100 K
  MAXSIZE 100 M
```

Wie Daten in Tabellenbereichen gespeichert werden

Standardmäßig wird DB2 jeweils in Disk Extents, verteilt (striped) über Container, schreiben. Wenn Sie zum Beispiel einen 4K Tabellenbereich mit einer Größe des Extents von 3 bei Verwendung von 3 Raw Containern bei einem DMS Tabellenbereich haben, bedeutet dies das 32K Daten (4K * 8 Pages je Extent = 32K) auf eine Festplatte geschrieben werden bevor in den nächsten Container geschrieben wird. Dies wird in Abbildung 6.6 gezeigt. Bitte beachten Sie, dass Tabellen keine Extents verwenden.

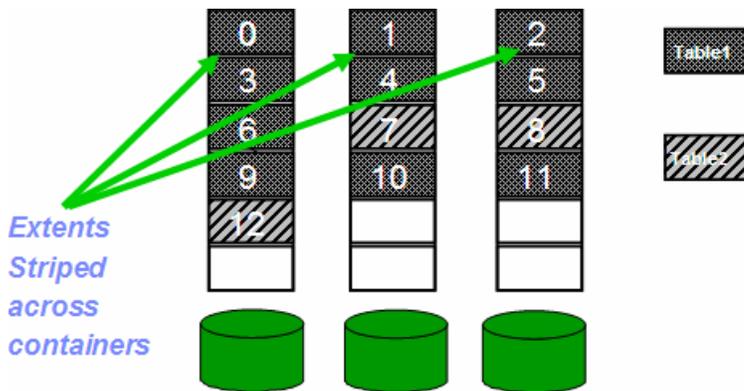


Abbildung 6.6 – Daten in Tabellenbereiche schreiben

Erstellen eines Tabellenbereichs über die Steuerzentrale

Um einen Tabellenbereich über die Steuerzentrale zu erstellen klicken Sie rechts auf den Tabellenbereiche Ordner innerhalb einer vorgegebenen Datenbank und wählen, wie in Abbildung 6.8. gezeigt, Create.

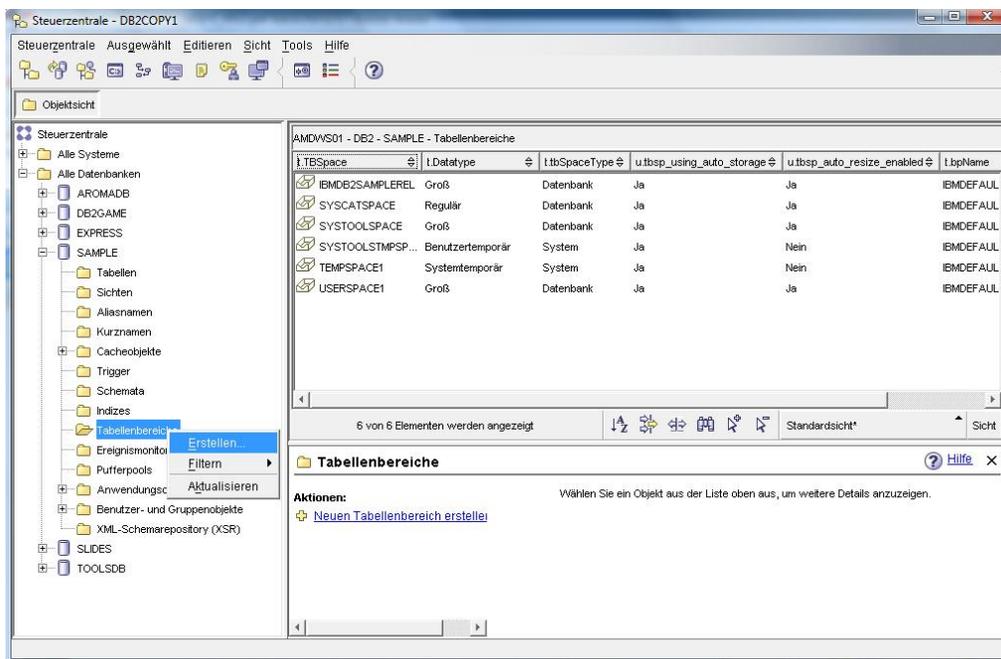


Abbildung 6.7 – Einen Tabellenbereich über die Steuerzentrale erstellen

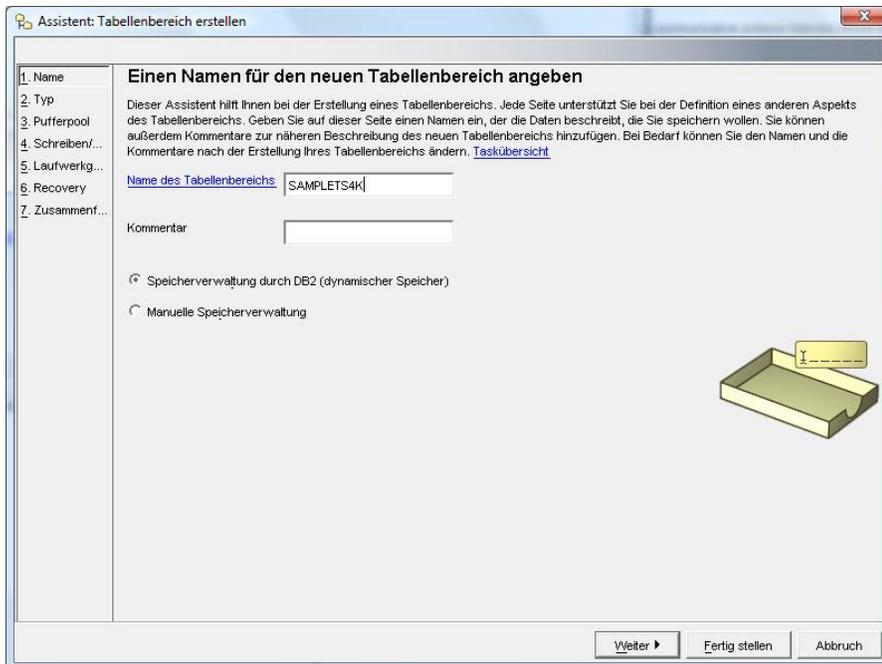


Abbildung 6.8 – Tabellenbereich erstellen Assistent

7

Kapitel 7 – DB2 Client Konnektivität

Dieses Kapitel beschreibt die für die Verbindungsherstellung eines DB2 Clients mittels TCPIP mit einem DB2 Server erforderliche Einrichtung. Bitte beachten Sie, dass ein DB2 Server bereits die Client Komponenten beinhaltet und ein DB2 Server auch als Client für die Verbindung mit einem anderen DB2 Server verwendet werden kann. Es gibt mehrere Möglichkeiten DB2 Client Verbindungen einzurichten. Dieses Kapitel beschreibt nur die einfachste Methode unter Verwendung des Konfigurationsassistenten.

Hinweis:

Für weitere Informationen zur DB2 Architektur können Sie sich dieses Video ansehen:
<http://www.channeldb2.com/video/video/show?id=807741:Video:4222>

7.1 DB2 Verzeichnisse

Bei DB2 Verzeichnissen handelt es sich um in Binärdateien gespeicherte Datenbankinformationen über Datenbanken mit welchen Sie sich mit ihrem Computer verbinden können. Es gibt vier Verzeichnisse:

1. Systemdatenbankverzeichnis

Dieses Verzeichnis ist wie das Inhaltsverzeichnis eines Buchs. Es zeigt alle Datenbanken, unabhängig davon ob es sich um entfernte oder lokale handelt, mit welchen Sie sich verbinden können. Bei einer lokalen Datenbank besitzt es einen Zeiger auf das *Lokale Datenbankverzeichnis*. Bei einer entfernten Datenbank besitzt es einen Zeiger auf einen Eintrag im *Knotenverzeichnis*. Um den Inhalt dieses Verzeichnisses zu sehen führen Sie diesen Befehl aus:

```
list db directory
```

2. Lokales Datenbankverzeichnis

Dieses Verzeichnis enthält Informationen zu lokal auf Ihrem Computer vorhandenen Datenbanken mit welchen sie sich verbinden können. Um den Inhalt dieses Verzeichnisses anzusehen führen Sie diesen Befehl aus:

```
list db directory on <drive/path>
```

3. Knotenverzeichnis

Dieses Verzeichnis schließt Informationen über den Verbindungsweg (wie wird verbunden) zu einer bestimmten Datenbank ein. Wenn z.B. das TCP/IP Protokoll verwendet wird, enthält ein TCP/IP Knoteneintrag die IP Adresse und den Port der Instanz des Servers auf

welchem die Datenbank, zu welcher Sie eine Verbindung herstellen wollen, liegt. Um den Inhalt dieses Verzeichnisses anzusehen führen Sie diesen Befehl aus:

```
list node directory
```

4. DCS Verzeichnis

Dieses Verzeichnis sehen Sie nur wenn sie die DB2 Connect Software zur Verbindung mit DB2 auf z/OS (Mainframe) oder DB2 auf i5/OS installiert haben. Um den Inhalt dieses Verzeichnisses anzusehen führen Sie diesen Befehl aus:

```
list dcs directory
```

Den Inhalt aller Verzeichnisse können Sie mit dem grafischen Konfigurationsassistent Dienstprogramm überprüfen und aktualisieren.

7.2 Konfigurationsassistent

Der grafische Konfigurationsassistent Dienstprogramms ermöglicht die spielend leichte Konfiguration von Verbindungen zwischen einem DB2 Client und einem DB2 Server.

Um den Konfigurationsassistenten unter Windows zu starten wählen Sie: *Start > Programme > IBM DB2 > DB2COPY1 (Standard) > Installations- und Konfigurationstools > Konfigurationsassistent.*

Über die Befehlszeile kann das Dienstprogramm mit dem Befehl `db2ca` gestartet werden. Abbildung 7.1 zeigt den Konfigurationsassistenten.

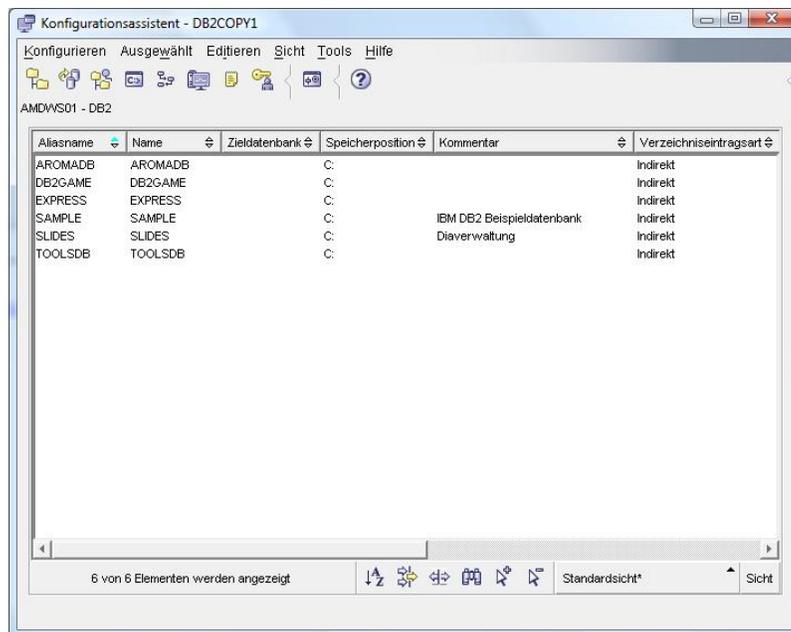


Abbildung 7.1 – Der Konfigurationsassistent

7.2.1 Auf dem Server erforderliche Einrichtung

Zwei Dinge müssen auf dem Server eingerichtet werden:

1) DB2COMM

Diese Registriervariable legt fest welches Kommunikationsprotokoll Anfragen von Clients überwacht. Üblicherweise wird als Kommunikationsprotokoll TCPIP verwendet. Wenn dieser Parameter geändert muss die Instanz neu gestartet werden. Um den DB2COMM zugewiesenen Wert im Konfigurationsassistenten anzusehen oder zu ändern wählen Sie *Konfigurieren -> DB2-Registrierdatenbank*, wie in Abbildung 7.2 und Abbildung 7.3 gezeigt.

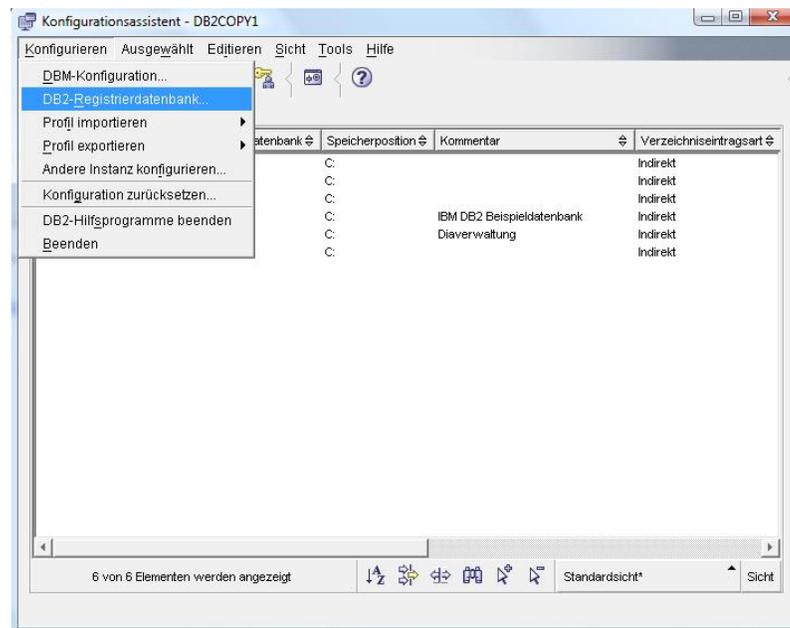


Abbildung 7.2 – Zugriff auf die DB2-Registrierdatenbank

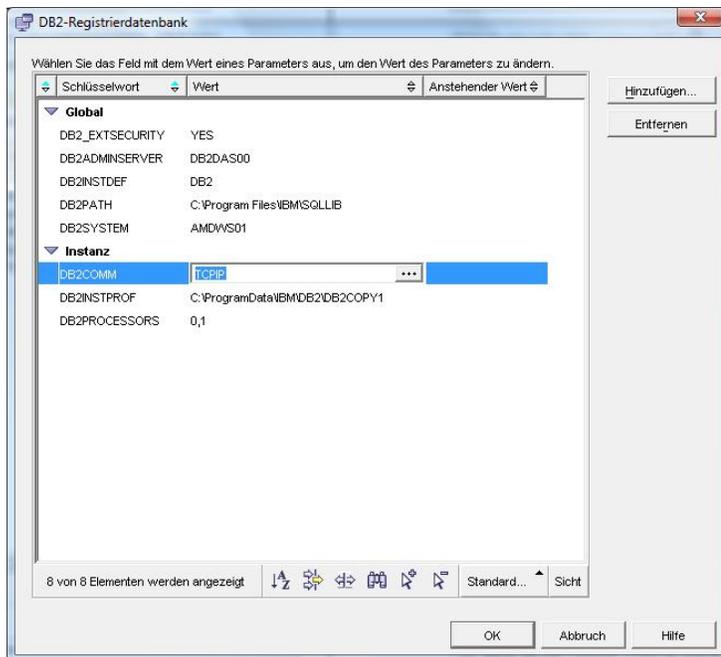


Abbildung 7.3 – Prüfen des Werts der DB2COMM DB2 Registrierungsvariablen

2) SVCENAME

Dieser DBM CFG Parameter sollte auf den Namen des Dienstes (wie in der TCPIP Servicedatei definiert) oder die bei Zugriff auf Datenbanken dieser Instanz zu verwendende Portnummer gesetzt werden. Im Konfigurationsassistenten wählen Sie, wie in Abbildung 7.4 gezeigt, *Konfigurieren > DBM-Konfiguration*.

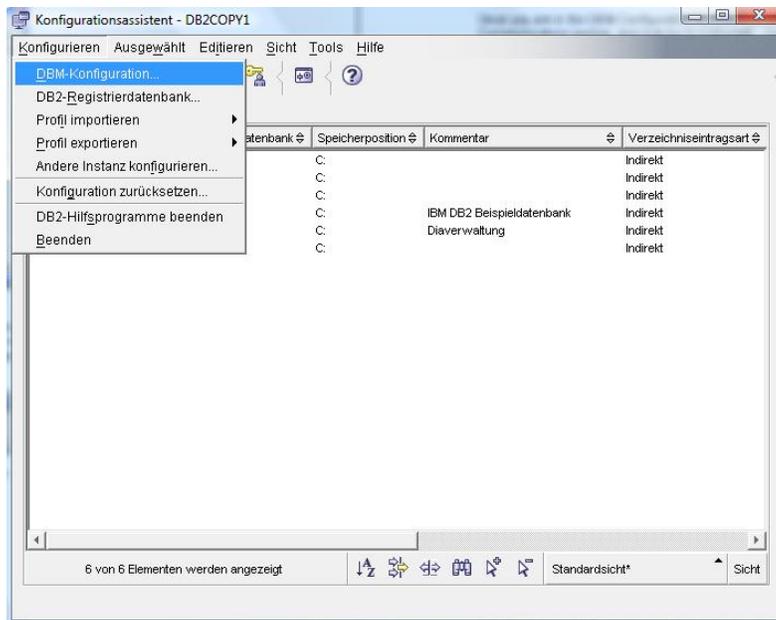


Abbildung 7.4 – Prüfen der DBM CFG mit dem Konfigurationsassistenten

Wenn Sie im DBM-Konfiguration Fenster sind lokalisieren sie den Abschnitt Kommunikation und suchen nach SVCENAME. Den Wert können Sie in eine Zeichenkette oder, falls erforderlich, sogar in eine Portnummer ändern. Dies wird in Abbildung 7.5 gezeigt.

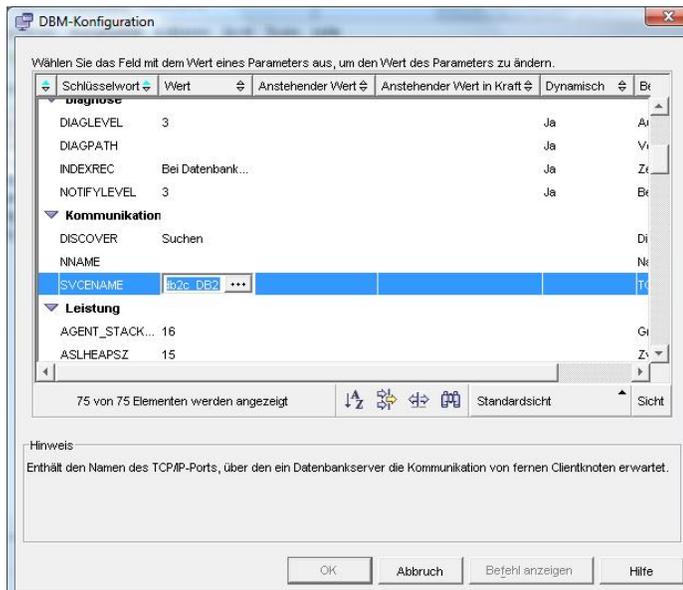


Abbildung 7.5 – Prüfen des SVCENAME DBM CFG Parameters

7.2.2 Auf dem Client erforderliche Einrichtung

Bevor Sie beginnen, benötigen sie auf dem Client folgende Informationen:

1. Der Name der Datenbank zu welcher eine Verbindung hergestellt werden soll,
2. Die Portnummer der DB2 Instanz auf dem Server auf welchem die Datenbank liegt. Sie können auch einen Servicenamen verwenden, wenn ein übereinstimmender Eintrag in der TCP/IP Servicedatei vorhanden ist.
3. Benutzererkennung und Kennwort des Betriebssystems um sich mit der Datenbank zu verbinden. Diese Benutzererkennung muss zuvor auf dem Server definiert werden.

Die obigen Informationen können vom DB2 Client über den Konfigurationsassistenten eingegeben werden. Als erstes starten Sie den Datenbank mit Assistenten hinzufügen Assistenten indem sie *Ausgewählt* klicken und *Datenbank mit Assistent hinzufügen* wählen, wie in Abbildung 7.6 gezeigt.

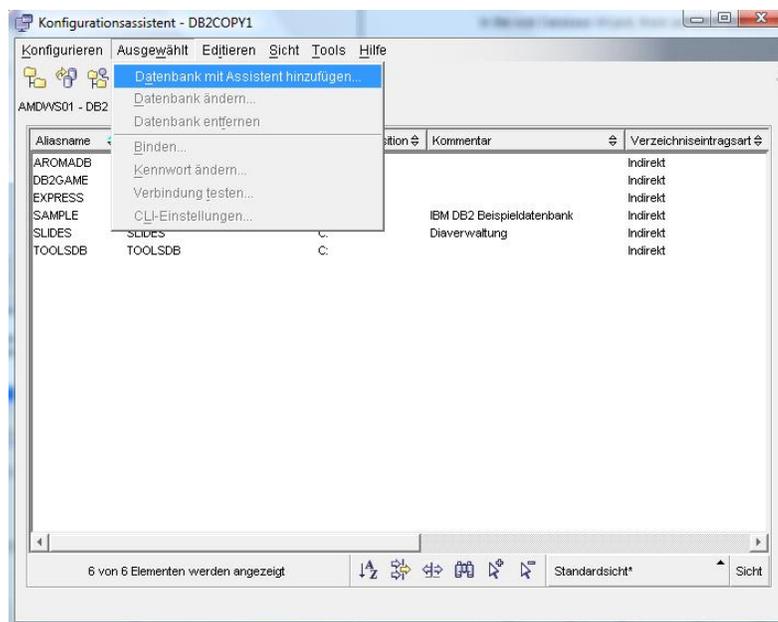


Abbildung 7.6 – Aufrufen des Datenbank mit Assistent hinzufügen Assistenten

Den Assistenten können Sie auch aufrufen indem sie in den leeren Bereich des Konfigurationsassistenten klicken und *Datenbank mit Assistent hinzufügen* wählen.

Abbildung 7.7 zeigt den Datenbank hinzufügen Assistent.



Abbildung 7.7 – Datenbank hinzufügen Assistent

Der Datenbank hinzufügen Assistent stellt drei Optionen zu Verfügung:

Profil verwenden

Es gibt möglicherweise Konstellationen in welchen Sie viele Clients für die Verbindung mit demselben DB2 Server einrichten müssen. In diesen Situationen ist es von Vorteil die vollständige Konfiguration auf einem Client durchzuführen und die Konfigurationen in einer sog. "Profil" Datei zu speichern. Diese Datei erlaubt die Verwendung dieser Informationen auf anderen Clients. In Abbildung 7.7, wenn Sie "Profil verwenden" wählen laden sie die Informationen aus einem existierenden "Profil". Weitere Details zur Erstellung von Client und Server Profilen finden Sie weiter unten in diesem Kapitel.

Netzwerk durchsuchen

Diese Methode, auch als "Discovery" bekannt, teilt DB2 mit, dass das Netzwerk nach einem bestimmten Server, Instanz und Datenbank durchsucht werden soll. Damit diese Methode funktioniert muss DAS (siehe Kap. 4) auf allen DB2 Servern, auf welchen Datenbanken gesucht werden sollen, laufen. Mit dieser Methode kann die Suche auf zwei Arten durchgeführt werden:

- Andere Systeme:
Durchsucht das gesamte Netzwerk. In einem großen Netzwerk mit vielen Hubs wird dies nicht empfohlen, da es sehr lange dauert die Informationen jedes Systems zu ermitteln.
- Bekannte Systeme:

Durchsuchen des Netzwerks nach einem bekannten Server mit einer von Ihnen angegebenen Adresse.

Die zwei Verfahren werden in Abbildung 7.8 illustriert.

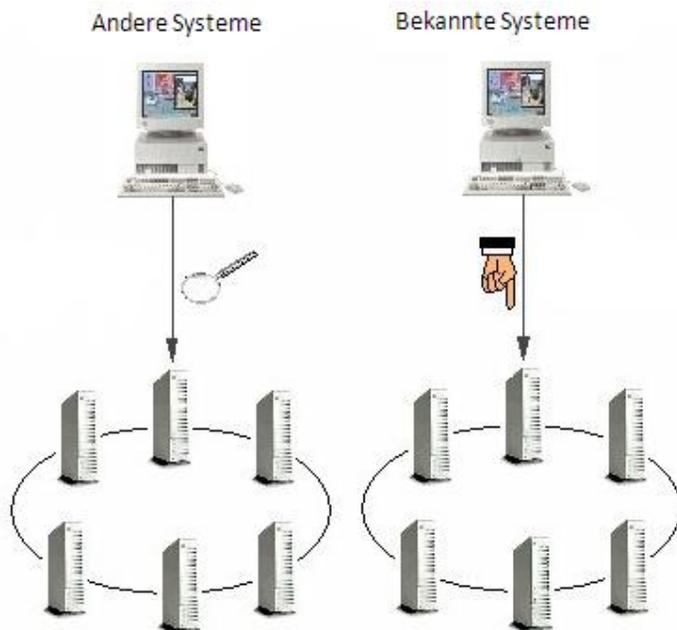


Abbildung 7.8 – Die Methoden der Suche nach anderen Systemen und bekannten Systemen (oder Discovery)

Es gibt möglicherweise Fälle in welchen ein Administrator verhindern möchte das Clients das Netzwerk nach Datenbanken mit vertraulichen Informationen durchsuchen. Dies kann durch den DAS, die Instanz oder auf Datenbankstufe verhindert werden. Abbildung 7.9 gibt darüber nähere Auskunft.

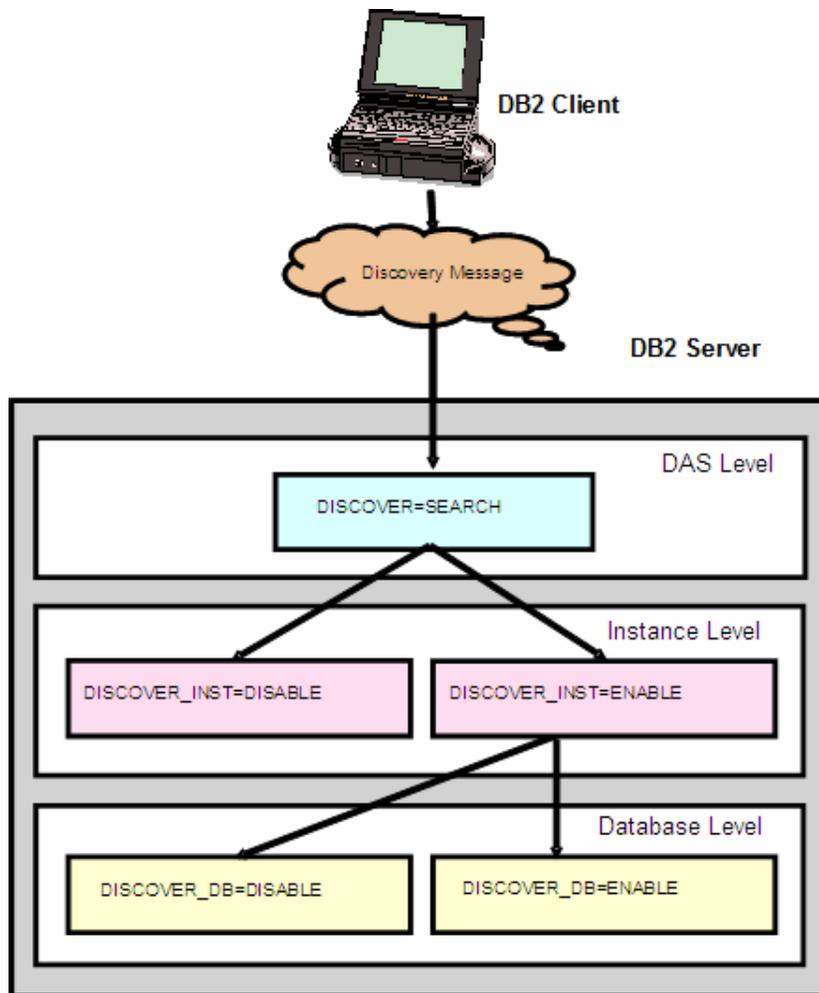


Abbildung 7.9 – Konfigurieren von die Suche beeinflussenden Parametern

Abbildung 7.9 zeigt die verschiedenen Stufen, auf welchen die Suche aktiviert bzw. deaktiviert werden kann. Auf DAS Stufe können Sie dem DISCOVER Parameter den Wert SEARCH oder KNOWN zuweisen. Auf Instanzebene kann der DISCOVER_INST DBM CFG Parameter auf DISABLE oder ENABLE gesetzt werden. Zu guter Letzt kann auf Datenbankebene der DISCOVER_DB Parameter auf ENABLE oder DISABLE gesetzt werden. Das Setzen dieser Parameter bietet Ihnen demzufolge die Granularität für die Datenbanksuche.

Verbindung zu Datenbank manuell herstellen

Bei Verwendung dieser Methode können Sie manuell Hostname, Portnummern und Datenbankinformationen im Konfigurationsassistent hinzufügen, welcher dann die "catalog" Befehle für die Ausführung der Verbindungskonfiguration generiert. Der Konfigurationsassistent überprüft nicht ob die Informationen korrekt sind. Dass diese

fehlerhaft sind werden sie erst feststellen, wenn sie sich nicht mit einem Server verbinden können. Sie müssen ebenfalls sicherstellen, dass die Benutzerkennung und das Kennwort mit welchem sie sich mit der entfernten Datenbank verbinden korrekt sind. In der Regel findet die Authentifizierung auf dem DB2 Server zu welchem Sie eine Verbindung herstellen wollen statt, weshalb es erforderlich ist eine auf diesem Server definierte Benutzerkennung und Kennwort anzugeben.

7.2.3 Client und Server Profile erstellen

Wenn Sie eine große Anzahl Server oder Clients konfigurieren können sie, statt jeden Client/Server individuell einzurichten, lediglich einen einrichten und das Profil exportieren (z.B. Konfigurationsdatei) und dieses Profil auf die anderen Clients/Server anwenden. Dies erspart sehr Administrationsaufwand bei der Einrichtung der Umgebung.

Um mit dem Konfigurationsassistenten ein benutzerdefiniertes Profil zu erstellen, klicken sie auf das *Konfigurieren* Menü und wählen, wie in Abbildung 7.10 gezeigt, *Profil exportieren* -> *Anpassen*.

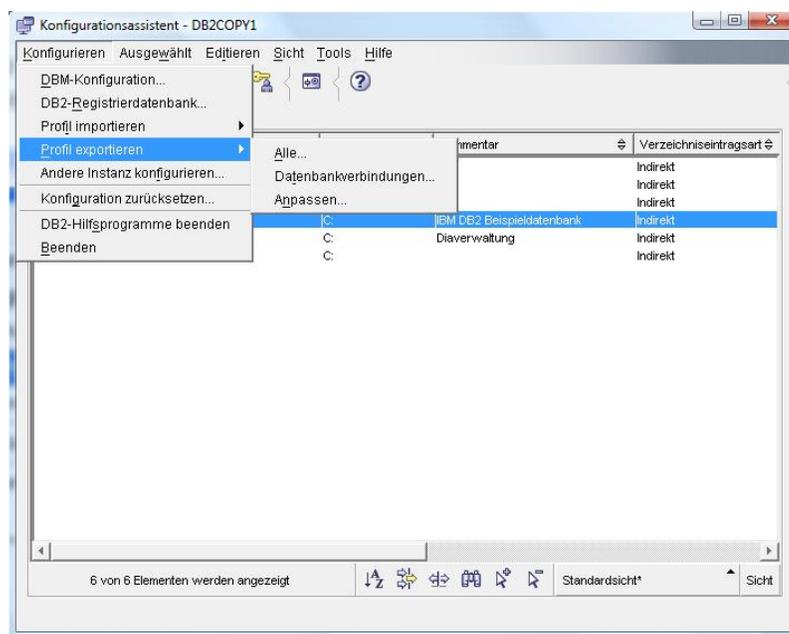


Abbildung 7.10 – Ein Profil exportieren

Abbildung 7.11 zeigt alle für den Export des Profils erforderlichen Felder.

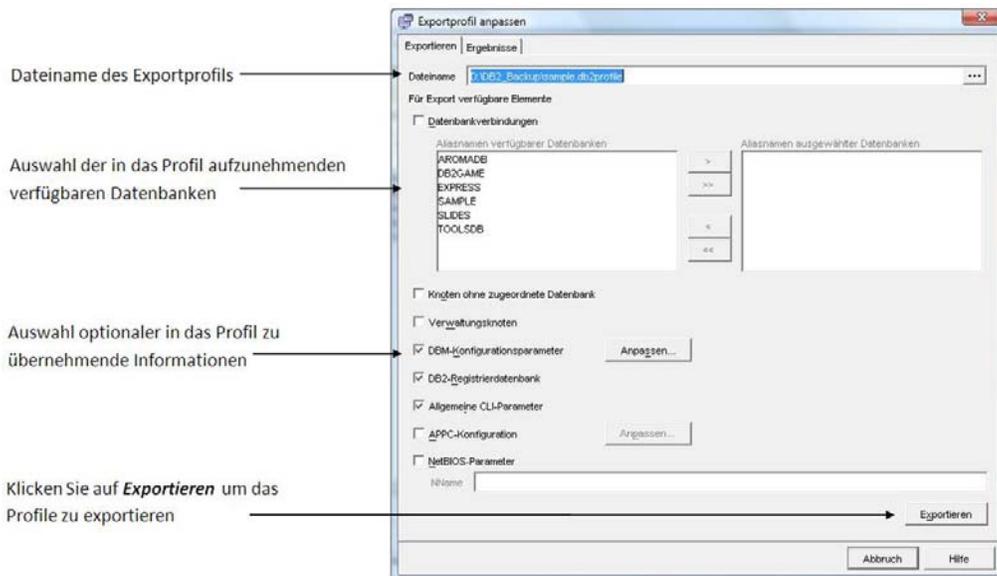


Abbildung 7.11 – Exportprofil anpassen Dialog

Abbildung 7.12 zeigt die Ergebnisse nachdem “Exportieren” im Exportprofil anpassen Dialog angeklickt wurde.

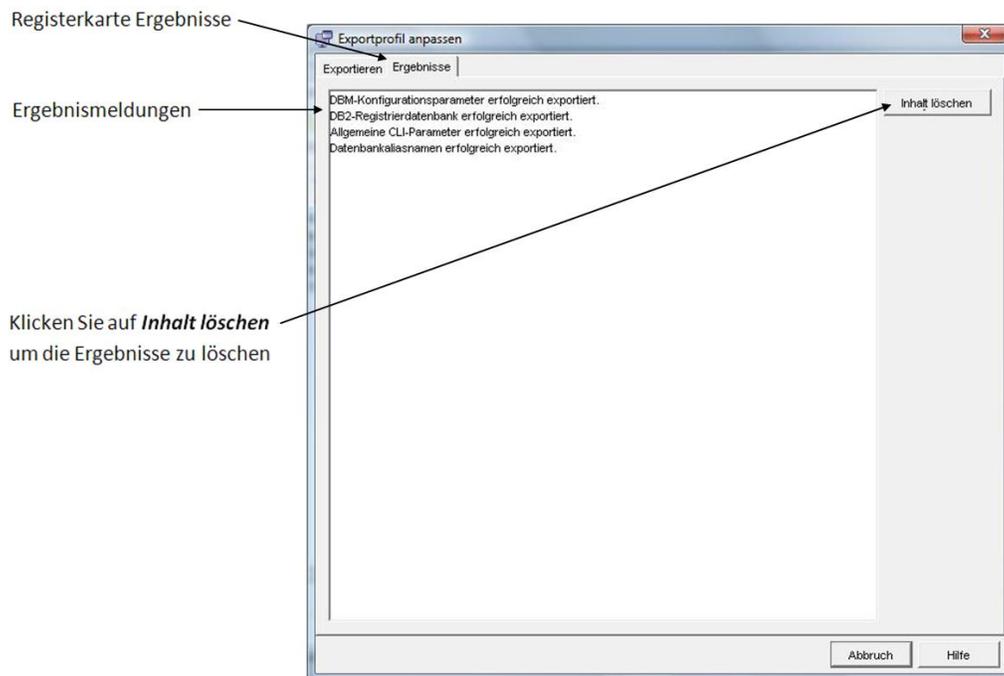


Abbildung 7.12 – Exportprofil Ergebnisse

Um ein benutzerdefiniertes Profil über den Konfigurationsassistenten zu importieren klicken sie auf das *Konfigurieren* Menü und wählen, wie in Abbildung 7.13 gezeigt, *Profil importieren -> Anpassen*.

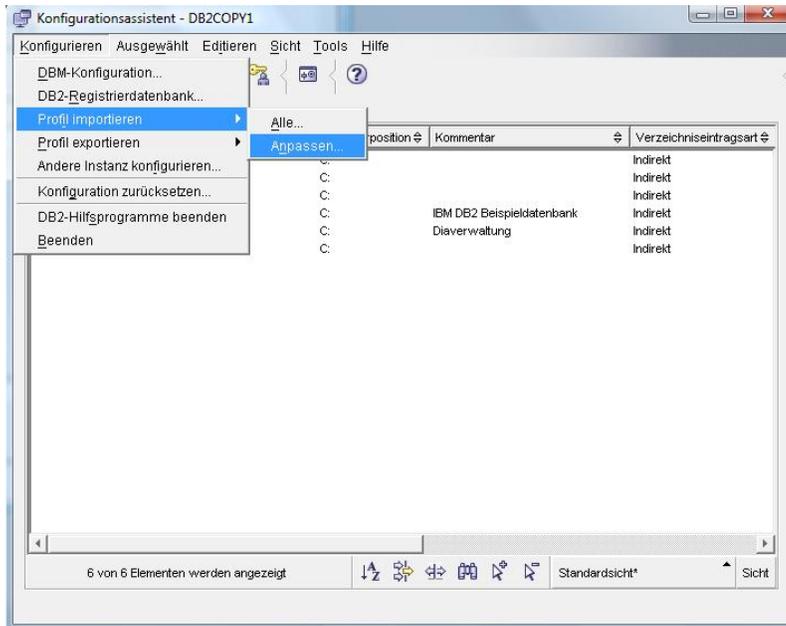


Abbildung 7.13 – Ein Profil importieren

Abbildung 7.14 zeigt alle für den Import eines Profils erforderlichen Felder.

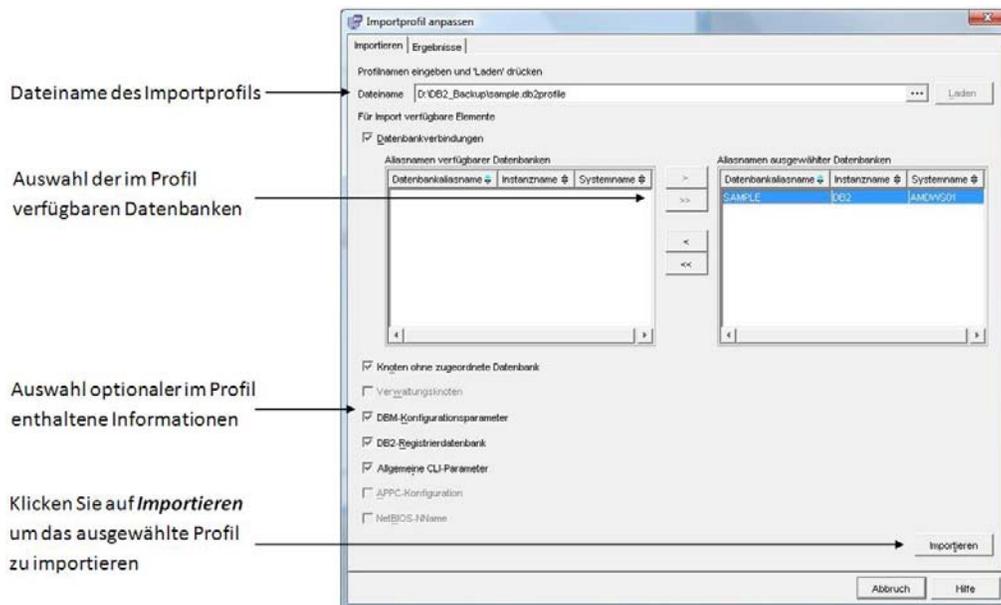


Abbildung 7.14 – Importprofil anpassen

Übung 6: Verwenden des Konfigurationsassistenten

Ziel

Der Konfigurationsassistent kann verwendet werden um schnell und einfach Verbindungen zu entfernten Datenbanken zu konfigurieren. In dieser Übung werden wir eine sich auf einem entfernten DB2 Server (repräsentiert durch Workstation ihres Nachbarn unter Verwendung sowohl der *Profil verwenden* als auch der *Netzwerksuche* Methode) befindende Datenbank katalogisieren. Sobald die Datenbank katalogisiert ist, können Sie auf diese genauso zugreifen wie auf eine auf ihrem System gespeicherte. DB2 führt die gesamte Kommunikationsverarbeitung "unter der Haube" durch.

Hinweis:

In dieser Übung wird davon ausgegangen dass Sie in einem Netzwerk arbeiten. Falls dies nicht der Fall ist, können Sie ihren eigenen Computer sowohl als Client als auch als Server verwenden und befolgen die Anweisungen weiter unten um sich mit dem eigenen System zu verbinden.

Vorgehensweise

1. Fragen Sie ihren Nachbar (oder Kursleiter) nach der folgenden Information:

Informationen zur entfernten Datenbank:

(PR)	Protokoll	<u>TCPIP</u>
(IP)	IP Adresse oder Hostname	_____
(PN)	Portnummer der Instanz	_____
(DB)	Datenbankname	<u>SAMPLE</u>

Hinweise:

- Um unter Windows den Hostnamen zu erfahren, geben Sie in der Eingabeaufforderung den Befehl *hostname* ein
- Um unter Windows die IP Adresse zu erfahren, geben Sie in der Eingabeaufforderung den Befehl *ipconfig* ein

2. Öffnen Sie den Konfigurationsassistenten. (Hinweis: er ist über das Start Menü erreichbar).
3. Öffnen Sie das *Ausgewählt* Menü und wählen den *Datenbank mit Assistent hinzufügen* Assistenten.
4. Auf der *Quelle* Seite des Assistenten wählen Sie die Option *Verbindung zur Datenbank manuell konfigurieren*. Klicken Sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.
5. Auf der *Protokoll* Seite des Assistenten wählen Sie die Option *TCP/IP*. Klicken Sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.
6. Auf der *TCP/IP* Seite des Assistenten geben Sie den vollständigen Hostnamen oder die in Schritt 1 notierte vollständige IP Adresse ein. Tragen Sie die in Schritt 1 aufgeschriebene Portnummer ein. Klicken Sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.

Hinweis: Die *Servicename* Option kann verwendet werden, wenn ein Eintrag in der *lokalen* Servicedatei mit einer Portnummer identisch mit dem Port auf welchem die entfernte Server Instanz horcht. Wenn Sie diese Option verwenden sieht DB2 in der Servicedatei auf den lokalen Computer nach, nicht auf dem Server. Um diese Option zu verwenden, müssen Sie einen Eintrag in der Servicedatei vornehmen.

7. Auf der *Datenbank* Seite des Assistenten geben Sie den in Schritt 1 ermittelten Namen der auf dem entfernten Server definierten Datenbank in das Feld *Datenbankname* ein. Sicherlich ist Ihnen aufgefallen, dass das Feld *Datenbankaliasname* automatisch mit dem gleichen Wert vorbelegt wird. Beim Datenbankaliasnamen handelt es sich um den von lokalen

Anwendungen für die Verbindung mit der Datenbank verwendeten Namen. Da Sie bereits eine lokale, *SAMPLE* genannte, Datenbank definiert haben wird DB2 es ihnen nicht gestatten eine andere Datenbank mit identischem Namen zu katalogisieren. Es ist deshalb erforderlich einen abweichenden Aliasnamen zu verwenden. In diesem Beispiel ändern Sie den Datenbankaliasnamen in *SAMPLE1*. Falls Sie möchten, können sie einen optionalen Kommentar zu dieser Datenbank eintragen. Klicken Sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.

8. Auf der *Datenquelle* Seite des Assistenten können Sie optional diese neue Datenbank (Datenquelle) als ODBC Datenquelle registrieren. Dies registriert die Datenbank automatisch im Windows ODBC Manager. Für dieses Beispiel entfernen Sie, da wir ODBC nicht verwenden, das Häkchen aus der *Datenbank für CLI/ODBC registrieren* Checkbox. Klicken Sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.
9. Auf der *Knotenoptionen* Seite des Assistenten wählen Sie das Betriebssystem der Servers, auf welchem sich die entfernte Datenbank befindet. Da alle Workstations in dieser Übung Microsoft Windows verwenden, stellen Sie sicher im Kombinationsfeld *Windows* gewählt zu haben. Das Feld Instanzname sollte auf *DB2* gesetzt sein. Falls dem nicht so ist, tragen Sie dort bitte *DB2* ein. Klicken Sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.
10. Die *Standardvorgaben* Seite des Assistenten gibt Ihnen die Möglichkeit sicherzustellen, dass der Systemname und Hostname richtig sind, und die Betriebssystem Einstellung zu überprüfen. Klicken Sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.
11. Die *Sicherheitsoptionen* Seite des Assistenten erlaubt Ihnen anzugeben wo und mit welcher Methode die Benutzerauthentifizierung stattfinden soll. Wählen Sie Option *Den in der DBM-Konfiguration des Servers angegebenen Authentifizierungswert verwenden*. Hierdurch wird die mit dem Parameter `AUTHENTICATION` in der entfernten Konfigurationsdatei spezifizierte Methode verwendet. Klicken Sie auf die *Fertig stellen* Schaltfläche um die entfernte Datenbank zu katalogisieren und den Assistenten zu schließen. Ein Bestätigungsdialog sollte angezeigt werden. Um sicherzustellen, dass Sie sich erfolgreich mit der Datenbank verbinden können klicken Sie auf die *Verbindung testen* Schaltfläche. Stellen Sie zudem sicher, dass es sich beim eingegebenen Benutzernamen und eingegebenen Kennwort um gültige *auf dem entfernten Server definierte* (da es wahrscheinlich ist, dass der Parameter `AUTHENTICATION` auf dem Server den Wert `SERVER` hat) handelt. Falls die Testverbindung erfolgreich ist, haben Sie die entfernte Datenbank erfolgreich katalogisiert. Falls der

Versuch fehlschlägt gehen Sie im Assistenten zurück und stellen sicher, dass die richtigen Werte angegeben sind. (Klicken Sie auf die *Zurück* Schaltfläche um in den Seiten des Assistenten zurückzublättern).

12. Öffnen Sie die Steuerzentrale und versuchen sie die verschiedenen Tabellen der neu katalogisierten Datenbank einzusehen.
13. Gehen Sie zurück zum Konfigurationsassistenten und versuchen sie eine andere Datenbank zu katalogisieren, doch dieses Mal verwenden sie die *Netzwerk durchsuchen* Option. Gehen Sie Schritt für Schritt durch den Assistenten, wie sie dies bereits bei der manuellen Konfiguration einer Verbindung taten. Bitte beachten Sie, dass die Netzwerksuche in großen Netzwerken sehr viel Zeit in Anspruch nehmen kann bevor Ergebnisse zurückgegeben werden.

8

Kapitel 8 – Mit Datenbankobjekten arbeiten

Dieses Kapitel behandelt Datenbankobjekte, wie Schemata, Tabellen, Sichten, Indizes, Sequenzen usw. Einige fortgeschrittene Datenbankanwendungsobjekte, wie Trigger, benutzerdefinierte Funktionen (UDFs) und gespeicherte Prozeduren werden in Kapitel 14, SQL PL gespeicherte Prozeduren, und Kapitel 15, Inline SQL PL, UDFs, Trigger, besprochen.

Hinweis:

Für weitere Informationen über die Arbeit mit Datenbankobjekten können Sie sich dieses Video anschauen:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4242>

8.1 Schema

Schemata sind *Namensbereiche* für eine Sammlung von Datenbankobjekten. Sie werden hauptsächlich verwendet für:

- Anzeige des Objektbesitzes oder der Beziehung zu einer Anwendung
- Logische Zusammengruppierung von Objekten

Alle DB2 Datenbankobjekte besitzen einen aus zwei Teilen zusammengesetzten vollqualifizierten Namen, wovon das Schema die erste Hälfte des Namens ist:

<schema_name>.<object_name>

Ein vollqualifizierter Name muss eindeutig sein. Wenn Sie sich ohne Angabe des Schemas mit einer Datenbank verbinden, verwendet DB2 die Benutzerkennung mit welcher sie sich verbinden als Schemaname. Wenn Sie sich zum Beispiel mit der Benutzerkennung "arfchong" mit der Datenbank verbinden und mit der CREATE TABLE Anweisung eine Tabelle erstellen:

```
CREATE TABLE artists ...
```

Lautet der vollqualifizierte Name der Tabelle tatsächlich arfchong.artists.

8.2 Tabellen

Eine Tabelle ist eine Sammlung von in Beziehung stehenden Daten, welche logisch in Spalten und Zeilen unterteilt sind. Die untenstehende Anweisung gibt Ihnen ein Beispiel wie eine Tabelle mit der `CREATE TABLE` Anweisung erstellt wird.

```
CREATE TABLE artists
(artno          SMALLINT    not null,
 name          VARCHAR(50) with default 'abc',
 classification CHAR(1)     not null,
 bio           CLOB(100K)   logged,
 picture       BLOB(2M)    not logged compact
)
IN mytbls1
```

In den folgenden Abschnitten beschreiben wir die wichtigsten Bestandteile dieser `CREATE TABLE` Anweisung.

8.2.1 Datentypen

Abbildung 8.1 listet die von DB2 unterstützten Datentypen auf

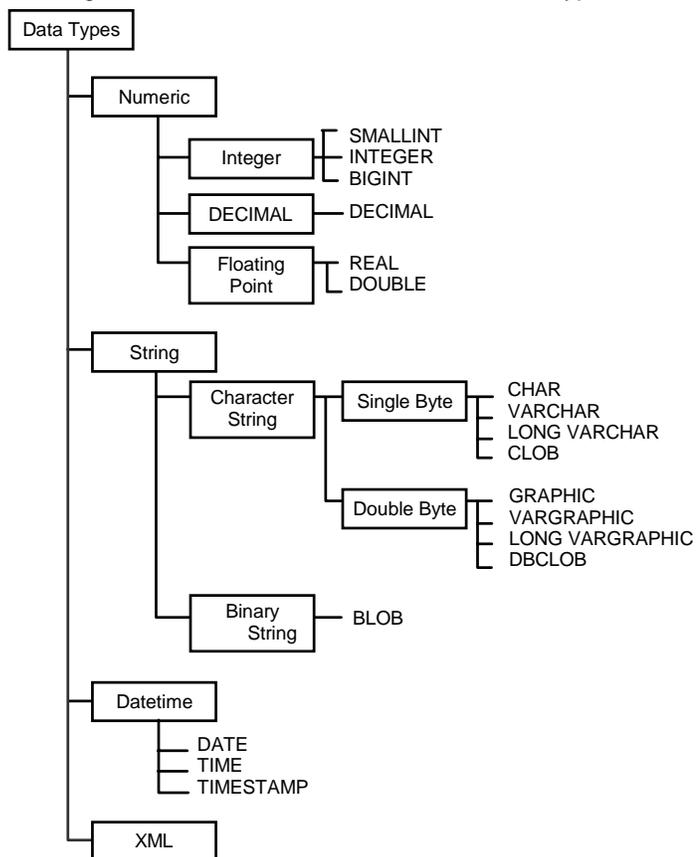


Abbildung 8.1 – DB2 eingebaute Datentypen

Large Object (LOB) Datentypen

Large Object Datentypen werden verwendet um große Zeichenketten, große Binärketten oder Dateien, wie in Abbildung 8.2 gezeigt, zu speichern.

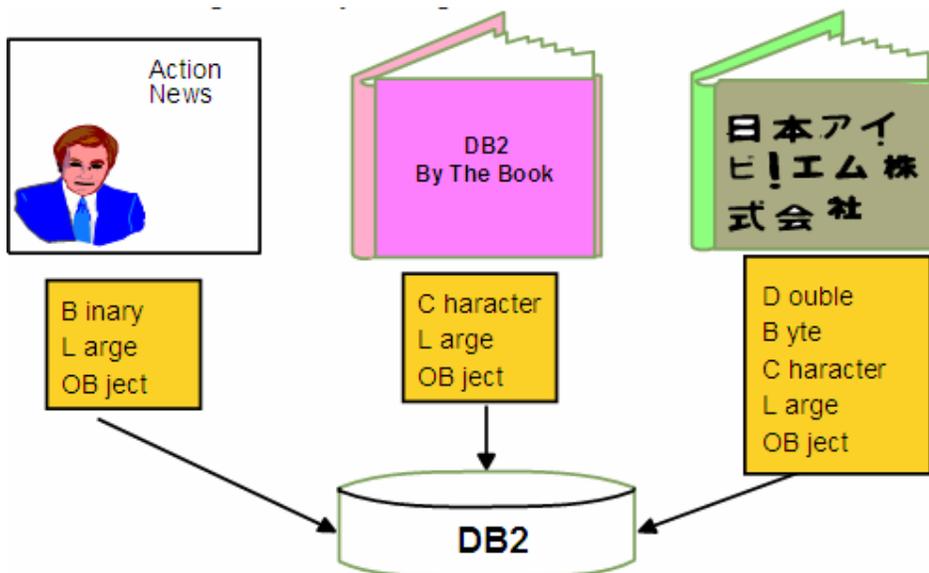


Abbildung 8.2 – LOB Datentypen

Diese großen Object Binaries werden gemeinhin der Deutlichkeit halber abgekürzt: ein großes binäres Objekt ist ein BLOB, ein großes Textdatenobjekt ist ein CLOB und ein Double-Byte Textdatenobjekt ist auch als DBCLOB bekannt.

Benutzerdefinierte Datentypen

DB2 erlaubt Ihnen die Definition eigener, auf die eingebauten Datentypen basierende, Datentypen. Diese sind bekannt als benutzerdefinierte Datentypen (UDTs). UDTs sind hilfreich wenn:

- Es ist erforderlich einen Kontext zu Werten herzustellen
- Es erforderlich ist, das DB2 Datentypisierung durchsetzt

Die folgenden Anweisungen veranschaulichen an Hand eines Beispiels wie und wann UDTs verwendet werden können:

```
CREATE DISTINCT TYPE POUND AS INTEGER WITH COMPARISONS
CREATE DISTINCT TYPE KILOGRAM AS INTEGER WITH
```

COMPARISONS

```
CREATE TABLE person
  (f_name    VARCHAR(30),
   weight_p  POUND NOT NULL,
   weight_k  KILOGRAM NOT NULL )
```

In diesem Beispiel werden zwei UDTs kreiert: POUND und KILOGRAM. Beide werden auf Basis des eingebauten Datentyps INTEGER aufgebaut. Die als Teil der Syntax definierte WITH COMPARISONS Klausel gibt an, dass zusätzlich Konvertierungsfunktionen (casting functions) mit demselben Namen wie der Datentyp erstellt werden.

Die Tabelle `person` verwendet die zwei neuen UDTs in den Spalten `weight_p` und `weight_k`. Wenn wir jetzt die folgende Anweisung ausführen:

```
SELECT F_NAME FROM PERSON
   WHERE weight_p > weight_k
```

Werden Sie eine Fehlermeldung erhalten, da zwei Spalten mit unterschiedlichen Datentypen verglichen werden. Obwohl `weight_p` und `weight_k` die POUND und KILOGRAM Datentypen verwenden, welche beide auf den Datentyp INTEGER basieren. Durch das Erstellen von UDTs machen sie diese Art des Vergleichs unmöglich. Das ist genau das was Sie wollen, denn was würde im realen Leben der Vergleich zwischen Pounds und Kilogramm bedeuten? Es ergäbe keinen Sinn.

Im nächsten Beispiel möchten Sie die Spalte `weight_p` mit einem Integer vergleichen doch da, wie auch immer, diese zwei Datentypen unterschiedlich sind, erhalten sie eine Fehlermeldung, außer sie verwenden eine Konvertierungsfunktion (casting function)).

Wie Sie an der unten stehenden Anweisung ersehen verwenden wir, um den Vergleich zu ermöglichen, die POUND() Konvertierungsfunktion. Wie bereits zuvor gesagt wurde die POUND() Konvertierungsfunktion mit dem UDT durch Verwendung der WITH COMPARISONS Klausel realisiert.

```
SELECT F_NAME FROM PERSON
   WHERE weight_p > POUND(30)
```

Null Werte

Der Wert Null stellt einen unbekanntem Status dar. Allerdings kann die CREATE TABLE Anweisung eine Spalte mit der NOT NULL Klausel definieren. Hierdurch wird sichergestellt, dass die Spalte einen bekannten Datenwert beinhaltet. Falls NOT NULL angegeben ist können Sie auch einen Standardwert angeben. Die nächste Anweisung gibt ein Beispiel dieses Verhaltens:

```
CREATE TABLE Staff (
  ID          SMALLINT NOT NULL,
  NAME        VARCHAR(9),
  DEPT        SMALLINT NOT NULL with default 10,
  JOB         CHAR(5),
  YEARS       SMALLINT,
```

```
SALARY    DECIMAL(7,2),
COMM      DECIMAL(7,2) with default 15
)
```

8.2.2 Identität Spalten

Bei einer Identitätsspalte handelt es sich um eine numerische Spalte welche automatisch einen einzigartigen numerischen Wert für jede neu eingefügte Zeile generiert. Je Tabelle ist nur eine Identitätsspalte zulässig.

Es gibt, abhängig davon wie es definiert wurde, zwei Möglichkeiten Werte für eine Identitätsspalte zu generieren:

- **Generated always:** Werte werden immer von DB2 generiert. Anwendungen ist es nicht gestattet einen expliziten Wert zu verwenden.
- **Generated by default:** Werte können explizit von einer Anwendung gesetzt werden oder, falls kein Wert zugewiesen wird, generiert DB2 diesen. DB2 kann keine Einzigartigkeit garantieren. Diese Option ist für die Datenübertragung und das Entladen und Laden von Tabellen gedacht.

Werfen wir einen Blick auf das folgende Beispiel:

```
CREATE TABLE subscriber(subscriberID INTEGER GENERATED ALWAYS AS
                           IDENTITY (START WITH 100
                                       INCREMENT BY 100),
                           firstname VARCHAR(50),
                           lastname  VARCHAR(50) )
```

In diesem Beispiel handelt es sich bei der INTEGER Spalte subscriberID um eine Identitätsspalte mit dem Attribut GENERATED ALWAYS. Der generierte Wert beginnt mit 100 und erhöht sich in 100er Schritten.

8.2.3 SEQUENZ Objekte

Sequenz Objekte generieren eine eindeutige Nummer über alle Datenbanken. Im Gegensatz zu Identitätsspalten sind Sequenzen unabhängig von Tabellen. Die folgenden Anweisungen geben ein Beispiel:

```
CREATE TABLE t1 (salary int)

CREATE SEQUENCE myseq
  START WITH 10
  INCREMENT BY 1
  NO CYCLE

INSERT INTO t1 VALUES (nextval for myseq)
INSERT INTO t1 VALUES (nextval for myseq)
INSERT INTO t1 VALUES (nextval for myseq)
```

```
SELECT * FROM t1
```

```
SALARY
```

```
-----  
      10  
      11  
      12
```

```
3 record(s) selected.
```

```
SELECT prevval for myseq FROM sysibm.sysdummy1
```

```
1
```

```
-----  
      12
```

```
1 record(s) selected
```

PREVVAL gibt den aktuellen Wert der Sequenz zurück, wohingegen NEXTVAL den nächsten Wert der Sequenz zurückgibt.

Das obige Beispiel verwendet zusätzlich SYSIBM.SYSDUMMY1. Hierbei handelt es sich um eine Systemkatalogtabelle, welche eine Spalte und eine Zeile enthält. Sie kann in Situationen verwendet werden wenn es erforderlich ist dass die Ausgabe einer Abfrage nur auf einen Wert basiert. Systemkatalogtabellen werden im nächsten Abschnitt näher beschrieben.

8.2.4 Systemkatalogtabellen

Jede Datenbank besitzt ihre eigenen Systemkatalogtabellen und Sichten. Diese speichern *Metadaten* über die Datenbankobjekte. Sie können diese Tabellen wie jede normale Tabelle abfragen. Drei Schemata werden zur Identifikation der Systemkatalogtabellen verwendet:

- SYSIBM: Basistabellen, optimiert für die DB2 Verwendung
- SYSCAT: auf SYSIBM Tabellen basierende Sichten, optimiert für leichte Verwendung
- SYSSTAT: Datenbankstatistiken

Nachfolgend einige Beispiele von Katalog Sichten:

- SYSCAT.TABLES
- SYSCAT.INDEXES
- SYSCAT.COLUMNS
- SYSCAT.FUNCTIONS
- SYSCAT.PROCEDURES

8.2.5 Deklarierte temporäre Tabellen

Deklarierte temporäre Tabellen werden im Speicher erstellt und werden von einer Anwendung verwendet und bei Beendigung der Anwendung automatisch gelöscht. Auf diese Tabellen kann nur von der Anwendung welche diese Tabellen erstellte zugegriffen werden. Es existiert kein Eintrag in einer DB2 Katalog Tabelle. Der Zugriff auf diese Tabellen bietet sehr effektive Performance da kein Konflikt mit dem Katalog existiert, keine Zeilen gesperrt werden, keine Standardprotokollierung (Protokollierung ist optional) und keine Benutzeridentitätsprüfung stattfindet. Diese temporären Tabellen unterstützen ebenfalls Indizes, d.h. jeder Standardindex kann in einer temporären Tabelle erstellt werden. Auch können Sie RUNSTATS gegen diese Tabellen ausführen.

Deklariert temporäre Tabellen werden im temporären Tabellenbereich des Benutzers angelegt, welcher vor der Erstellung einer deklarierten temporären Tabelle definiert werden muss. Die folgenden Anweisungen geben Ihnen ein Beispiel wie drei deklarierte temporäre Tabellen erstellt werden:

```
CREATE USER TEMPORARY TABLESPACE apptemps
    MANAGED BY SYSTEM USING ('apptemps');
```

```
DECLARE GLOBAL TEMPORARY TABLE employees
    LIKE employee NOT LOGGED;
```

```
DECLARE GLOBAL TEMPORARY TABLE tempdept
    (deptid CHAR(6), deptname CHAR(20))
    ON COMMIT DELETE ROWS NOT LOGGED;
```

```
DECLARE GLOBAL TEMPORARY TABLE tempprojects
    AS ( fullselect ) DEFINITION ONLY
    ON COMMIT PRESERVE ROWS NOT LOGGED
    WITH REPLACE IN TABLESPACE apptemps;
```

Wenn eine deklarierte temporäre Tabelle erstellt wird lautet ihr Schema SESSION und muss spezifiziert werden. Die Benutzerkennung, welche die temporäre Tabelle erstellt hat, besitzt alle Rechte an der Tabelle. Jede eine temporäre Tabelle erstellende Anwendung besitzt ihre eigene unabhängige Kopie, wie in Abbildung 8.5 gezeigt.



Abbildung 8.5 – Gültigkeitsbereich global deklarierter temporärer Tabellen

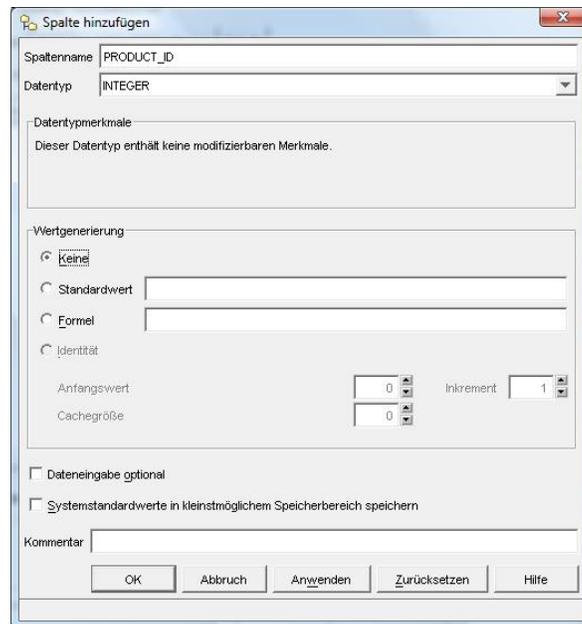
Übung 7: Erstellen einer neuen Tabelle

Ziel

Bis jetzt haben wir die *SAMPLE* Datenbank zur Erläuterung der Konzepte verwendet. Irgendwann müssen Sie ihre eigenen Tabellen in einer Datenbank erstellen. In dieser Übung verwenden wir für die Erstellung von zwei neuen Tabellen in der *SAMPLE* Datenbank den *Tabelle erstellen Assistenten*.

Vorgehensweise

1. Starten Sie, wie zuvor in dieser Präsentation, den *Tabelle erstellen Assistenten*. (Steuerzentrale -> Alle Datenbanken -> *SAMPLE* -> (rechtsklick) Tabellen Objekt -> Erstellen ... Option)
2. Definieren den Tabellennamen, Spaltendefinitionen und erforderliche Integritätsbedingungen (Constraints). Diese Tabelle wird verwendet, um Informationen bezüglich des von einem Projekt benötigten Bürobedarfs in der *SAMPLE* Datenbank zu speichern. Jedes Mal wenn Büromaterial gekauft wird, wird eine Zeile zu dieser Tabelle hinzugefügt. Die Tabelle wird sechs Spalten besitzen:
 - product_id: Eindeutiger Bezeichner des zu kaufenden Artikels
 - description: Beschreibung des Artikels
 - quantity: Die gekaufte Menge
 - cost: Der Artikelpreis
 - image: Ein Bild des Artikels (falls verfügbar)
 - project_num: Das Projekt, für welches der Artikel gekauft wurde
3. Auf der ersten Seite des Assistenten wählen Sie ihren Benutzerkennung, mit welcher sie eingeloggt sind, und verwenden den Tabellennamen *SUPPLIES*. Optional können sie einen Kommentar hinzufügen. Klicken sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.
4. Auf dieser Seite können Sie Spalten zur Tabelle hinzufügen. Klicken Sie auf *Hinzufügen* Schaltfläche um Spalten hinzuzufügen.



Tragen Sie den Spaltennamen “*product_id*” ein und wählen als Datentyp *INTEGER*. Entfernen Sie das Häkchen bei *Dateneingabe optional* (NOT NULL) und klicken auf die *Anwenden* Schaltfläche um die Spalte zu definieren.

Wiederholen Sie diesen Schritt für die verbleibenden Spalten der Tabellen unter Verwendung der in der Tabelle gezeigten Optionen. Wenn alle Spalten hinzugefügt (angewandt) wurden, klicken Sie auf die *OK* Schaltfläche und sie sollten eine zusammengefasste Liste aller soeben von ihnen erstellten Spalten sehen. Klicken Sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.

Spaltenname	Attribute
product_id (completed)	INTEGER, NOT NULL
description	VARCHAR, Länge 40, NOT NULL
quantity	INTEGER, NOT NULL
cost	DECIMAL, Genauigkeit 7, Anzahl Kommastellen 2, NOT NULL
image	BLOB, 1MB, NULLABLE, NOT LOGGED
project_num	CHAR, Länge 6, NOT NULL

Hinweis: Die NOT LOGGED Option kann bei der Deklaration von LOB Spalten angegeben werden. Für Spalten mit einer Größe über 1 GB ist diese zwingend erforderlich. Sie wird im Allgemeinen auch für LOBs mit einer Größe von über 10 MB empfohlen, da Änderungen an großen Spalten die Protokolldateien schnell füllen. Selbst wenn NOT LOGGED verwendet wird können Änderungen an LOB Dateien während einer Transaktion erfolgreich rückgängig gemacht (rolled back) werden. Bitte beachten Sie ebenfalls, dass es sich bei der Spalte *image* um die einzige Spalte handelt, welche als "NULLABLE" Spalte definiert wurde. Weshalb wurde Ihrer Meinung nach die Spalte so definiert?

5. Bis hierher wurden alle erforderlichen Informationen für die Tabellenerstellung angegeben. Indem die anderen Seiten übersprungen werden, wählen Sie die Standardoptionen für diese Optionen. Nach der Tabellenerstellung können Sie immer noch Schlüssel und Constraints hinzufügen.
6. Fügen Sie einen Constraint zur Begrenzung der Werte der Spalte *quantity* hinzu. Auf der *Integritätsbedingungen* Seite des Assistenten klicken Sie auf die Hinzufügen Schaltfläche. In das *Namen überprüfen* Feld schreiben Sie *valid_quantities*. In das *Bedingung für Prüfung auf Integritätsbedingung* Feld schreiben Sie *quantity > 0*.

Klicken Sie auf die *OK* Schaltfläche. Sie sollten eine Zusammenfassung der soeben von ihnen auf der *Integritätsbedingungen* Seite hinzugefügten Integritätsbedingung sehen. Klicken Sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.

7. Sie können den Assistenten weiter erforschen und weitere Parameter ändern. Alternativ können Sie zur *Zusammenfassung* Seite wechseln oder auf die *Fertig stellen* Schaltfläche klicken um die Tabelle zu erstellen.
8. In der Steuerzentrale klicken Sie im Objektsicht Teilfenster auf den *Tabellen* Ordner unter der *SAMPLE* Datenbank. Die soeben von Ihnen erstellte Tabelle sollte jetzt in der Liste erscheinen. Um die Änderungen zu sehen ist es eventuell erforderlich die Ansicht der Steuerzentrale zu aktualisieren.

8.3 Sichten

Bei einer Sicht handelt es sich um die Darstellung von Tabellendaten. Die Daten der Sicht werden nicht getrennt gespeichert, sondern bei Aufruf der Sicht abgerufen. Verschachtelte Sichten, d.h. eine erstellte Sicht basiert auf andere Sichten, werden unterstützt. Alle Informationen über Sichten werden in den folgenden DB2 Katalogsichten gespeichert: SYSCAT.VIEWS, SYSCAT.VIEWDEP und SYSCAT.TABLES. Nachfolgend ein Beispiel wie eine Sicht erstellt und verwendet wird.

```
CONNECT TO MYDB1;

CREATE VIEW MYVIEW1
  AS SELECT ARTNO, NAME, CLASSIFICATION
  FROM ARTISTS;

SELECT * FROM MYVIEW1;
```

Output:

ARTNO	NAME	CLASSIFICATION
10	HUMAN	A
20	MY PLANT	C
30	THE STORE	E
...		

8.4 Indizes

Bei einem Index handelt es sich um einen geordneten Satz von Schlüsseln, welche jeweils auf eine Spalte in eine Tabelle zeigen. Ein Index gestattet Eindeutigkeit und erhöht zusätzlich die Leistung. Einige der Charakteristika, welche bei einem Index definiert werden können:

- Die Sortierfolge des Indexes kann aufsteigend oder absteigend sein
- Die Indexschlüssel können eindeutig oder nicht eindeutig sein
- Mehrere Spalten können für den Index verwendet werden (dies wird zusammengesetzter Schlüssel genannt)
- Falls der Index und die physikalischen Daten in einer ähnlichen Indexreihenfolge gebündelt (clustered) sind, handelt es sich um einen Cluster Index

Zum Beispiel:

```
CREATE UNIQUE INDEX artno_ix ON artists (artno)
```

8.4.1 Designadvisor

Beim Designadvisor handelt es sich um hervorragendes Dienstprogramm, welches Sie beim optimalen Design ihrer Datenbank für eine bestimmte SQL Auslastung beraten kann. Der Designadvisor kann Ihnen beim Design ihrer Indizes, materialisierten Abfragetabellen (MQT - Materialized Query Tables), mehrdimensionaler Aufteilung (MDC - Multi-

Dimensional Clustering) und der Datenbankpartitionierung (DPF - Database Partitioning Feature) helfen. Der Designadvisor wird über die Steuerzentrale aufgerufen indem Sie die Datenbank rechts anklicken und, wie in Abbildung 8.6 gezeigt, "Designadvisor" wählen.

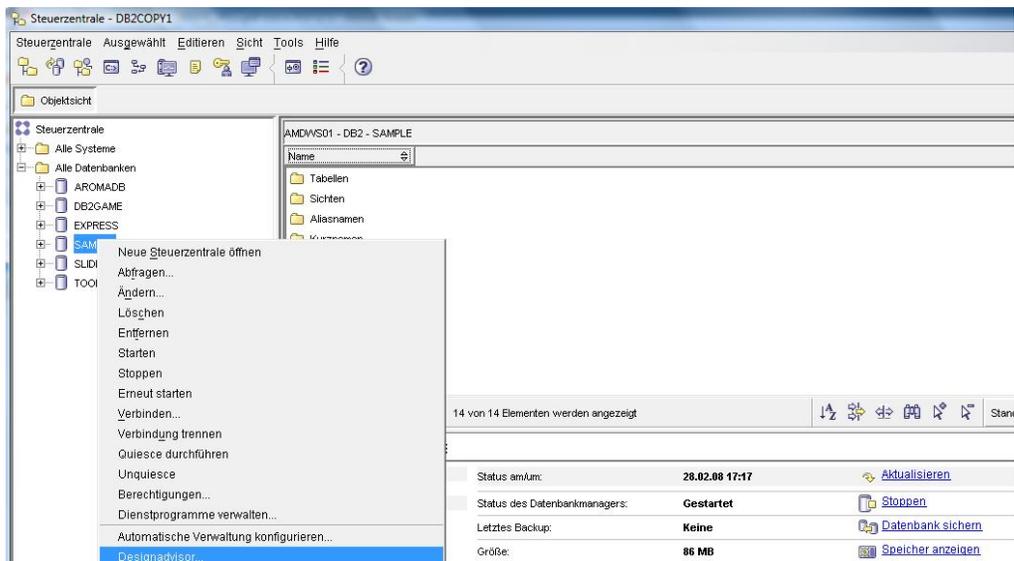


Abbildung 8.6 – Aufruf des Designadvisors über die Steuerzentrale

Abbildung 8.7 zeigt den Designadvisor. Folgen Sie den Schritten dieses Assistenten um Design Empfehlungen von DB2 zu erhalten.

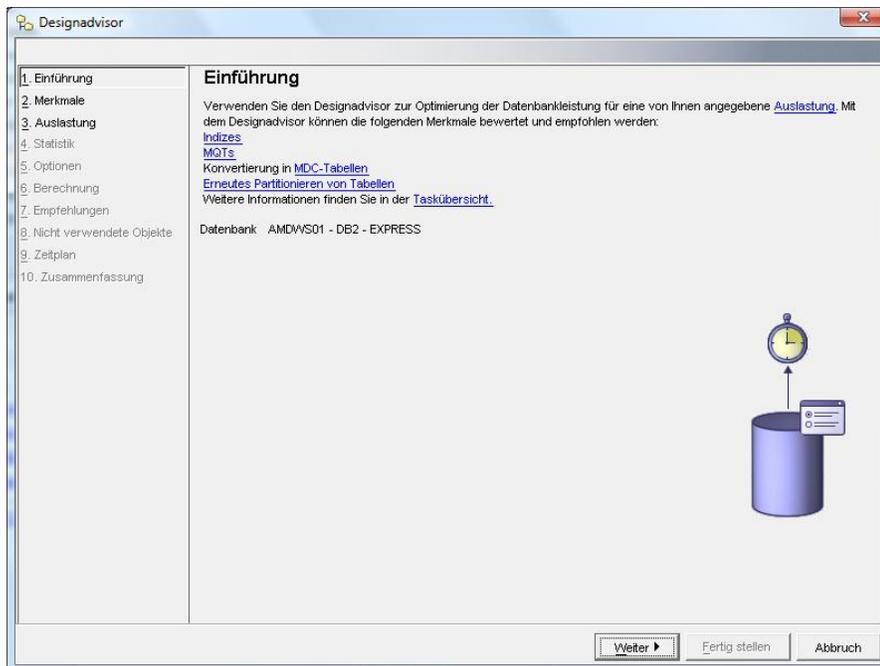


Abbildung 8.7 – Der Designadvisor

8.5 Referentielle Integrität

Referentielle Integrität ermöglicht Ihrer Datenbank Beziehungen zwischen Tabellen zu verwalten. Sie können, wie in Abbildung 8.8 gezeigt, Beziehungen vom Typ Eltern-Kind (Parent-Child) festlegen. In der Abbildung gibt es zwei Tabellen, DEPARTMENT und EMPLOYEE, welche über die Abteilungsnummer (DEPTNO / WORKDEPT) in Beziehung zueinander stehen. Die WORKDEPT Spalte der EMPLOYEE Tabelle kann ausschließlich Abteilungsnummern enthalten welche bereits in der DEPARTMENT Tabelle vorhanden sind. Dies ist darin begründet, dass es sich in diesem Beispiel bei der DEPARTMENT Tabelle um die Parent-Tabelle handelt und die EMPLOYEE Tabelle die Child, oder abhängige, Tabelle ist. Die Abbildung zeigt auch die für die Einrichtung der Beziehung erforderliche CREATE TABLE Anweisung der EMPLOYEE Tabelle.

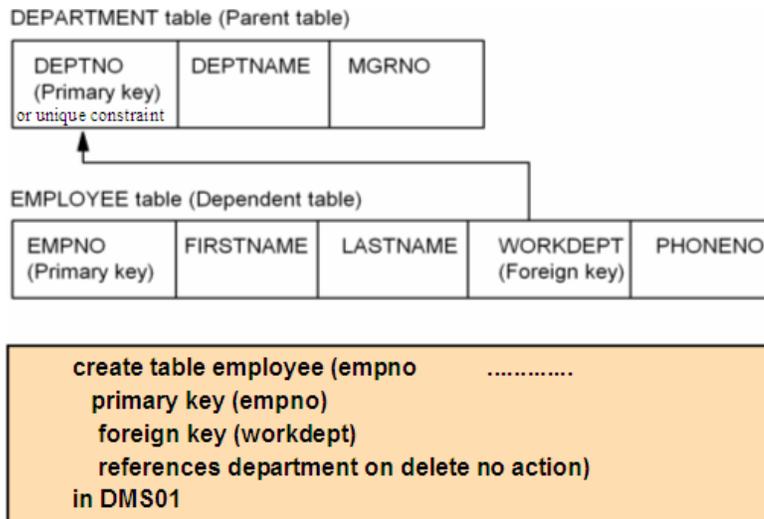


Abbildung 8.8 – Beispiel der referentiellen Integrität zwischen Tabellen

Bei der referentiellen Integrität werden die folgenden Konzepte oft verwendet

Konzept	Beschreibung
Parent-Tabelle	Eine kontrollierende Datentabelle in welcher der Hauptschlüssel existiert
Child-Tabelle	Eine von den Daten der Parent-Tabelle abhängige Tabelle. Sie enthält zusätzlich einen Fremdschlüssel. Damit eine Zeile in die abhängige Tabelle aufgenommen werden kann, muss eine übereinstimmende Zeile in der Parent-Tabelle bereits vorhanden sein.
Primärschlüssel	Definiert den Primärschlüssel der Parent-Tabelle. Dieser darf keinen NULL Wert beinhalten und muss eindeutig sein. Ein Primärschlüssel besteht aus einer oder mehreren Spalten in einer Tabelle.
Fremdschlüssel	Referenziert den Primärschlüssel der Parent-Tabelle

Daten in Tabellen können in Beziehung zu Daten in einer oder mehreren Tabellen mit referentieller Integrität stehen. Integritätsbedingungen (Constraints) können auch auf Datenwerten beruhen, welche bestimmten Eigenschafts- oder Geschäftsregeln entsprechen. Wenn zum Beispiel in einer Tabelle das Geschlecht einer Person gespeichert wird, kann die Integritätsbedingung sicherstellen, dass ausschließlich die Werte "M" für männlich und "F" für weiblich zugelassen werden.

9

Kapitel 9 – Die EXPORT und IMPORT Dienstprogramme

Die in diesem Kapitel beschriebenen Dienstprogramme und Befehle dienen dazu, Daten innerhalb derselben Datenbank oder in eine andere Datenbank derselben oder einer anderen Plattform zu kopieren. Abbildung 9.1 gibt einen Überblick über die Dienstprogramme EXPORT und IMPORT.

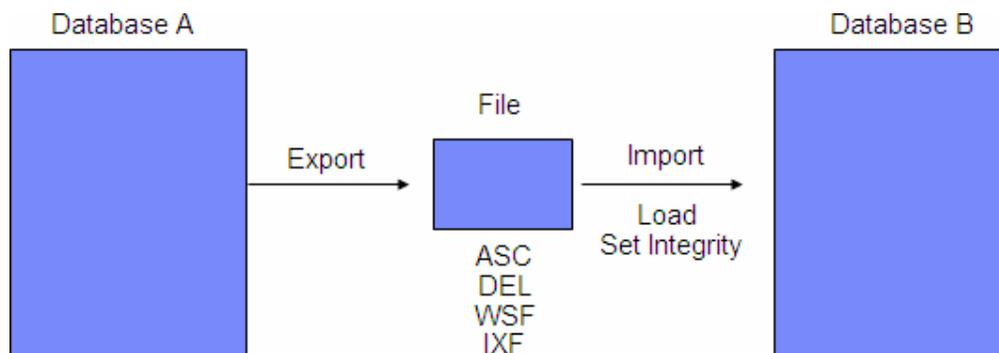


Abbildung 9.1 – Die EXPORT und IMPORT Dienstprogramme

In Abbildung 9.1 befinden sich die Datenbanken A und B. Mittels des EXPORT Dienstprogramms können Daten einer Tabelle in eine Datei gesichert werden. Erlaubt sind die nachfolgenden Dateiformate:

ASC = ASCII

DEL = Delimited ASCII (zeichenbegrenzte ASCII Datei)

WSF = Worksheet Format (Kalkulationsblatt/Arbeitsmappe)

IXF = Integrated Exchange Format (integriertes Austauschformat)

Bei ASC und DEL Dateien handelt es sich um Textdateien, welche mit jedem Texteditor geöffnet und editiert werden können. Das WSF Format dient dem Austausch mit Tabellenkalkulationsprogrammen, wie Excel oder Lotus® 1-2-3. Eine Besonderheit stellt das IXF Format dar, da dieses nicht nur die Daten, sondern auch die Data Definition Language (DDL) der in Frage kommenden Tabelle enthält. Dies ist günstig da, falls die Tabelle wiederhergestellt werden muss, die Wiederherstellung direkt mit der Datei im IXF Format durchgeführt werden kann, wohingegen dies mit anderen Formaten nicht möglich wäre.

Nachdem die Daten in eine Datei exportiert wurden, kann das IMPORT Dienstprogramm dazu verwendet werden die Daten der Datei in eine andere Tabelle zu importieren. Bei Sicherungen im ASC, DEL und WSF Format muss die Tabelle bereits existieren; wohingegen beim IXF Format die Tabelle nicht bereits vorhanden sein muss. Eine andere Methode besteht darin, die Daten mit dem LOAD Dienstprogramm zu laden. Das LOAD Dienstprogramm ist schneller, da es direkt, unter Umgehung von DB2, auf die Datenbankseiten zugreift. Hierbei ist zu beachten, dass diese Vorgehensweise keine Integritätsbedingungen (Constraints) prüft und Trigger nicht ausgelöst werden. Um die Konsistenz der mit dem LOAD Dienstprogramm geladenen Daten sicherzustellen wird im Anschluss durchweg der SET INTEGRITY Befehl verwendet.

Die nächsten Abschnitte beschreiben die EXPORT, IMPORT und LOAD Dienstprogramme detaillierter.

Hinweis:

Für weitere Informationen über die Arbeit mit den Dienstprogrammen EXPORT und IMPORT können Sie sich dieses Video ansehen:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4262>

9.1 Das EXPORT Dienstprogramm

Das EXPORT Dienstprogramm wird verwendet um, wie bereits zuvor erwähnt, Daten einer Tabelle in eine Datei zu extrahieren. Hinter den Kulissen wird eine SQL SELECT Anweisung ausgeführt. Das folgende Beispiel exportiert die ersten 10 Zeilen der Tabelle *employee* in die Datei *empolyee.ixf* im IXF Format.

```
EXPORT TO employee.ixf OF IXF
  SELECT * FROM employee
  FETCH FIRST 10 ROWS ONLY
```

Wir empfehlen Ihnen das obige Beispiel auszuprobieren. Da die *employee* Tabelle Teil der SAMPLE Datenbank ist, müssen Sie zuerst eine Verbindung zu dieser, in einem vorhergehenden Kapitel erstellten, Datenbank herstellen.

Falls Sie die Arbeit mit einer grafischen Benutzeroberfläche vorziehen, kann das EXPORT Dienstprogramm, wie aus Abbildung 9.2 ersichtlich, auch über die DB2-Steuerzentrale aufgerufen werden.

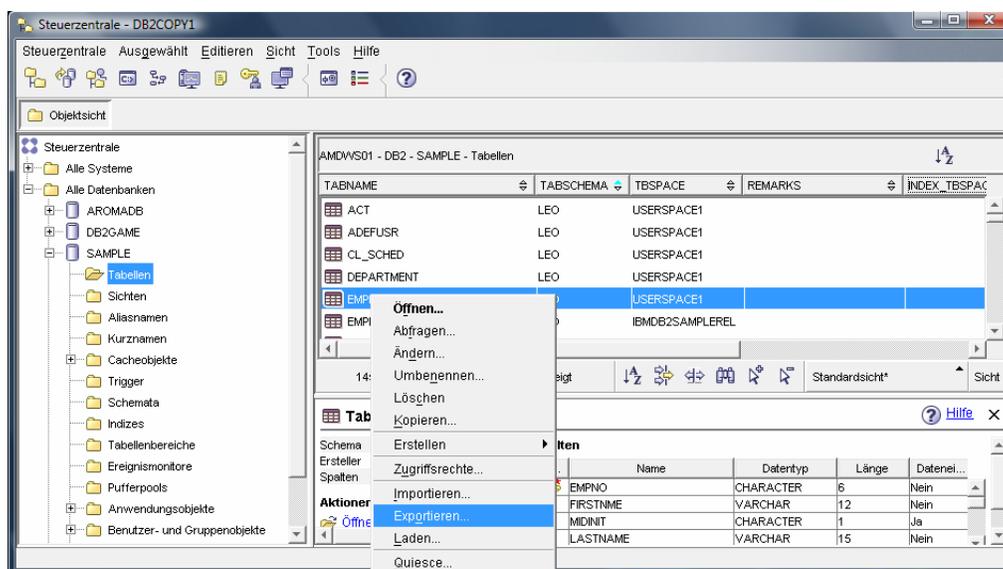


Abbildung 9.2 – Starten des Exportieren Notizbuchs

Wie aus der Abbildung hervorgeht, wählen Sie durch anklicken zuerst die employee Tabelle. Anschließend klicken Sie mit der rechten Maustaste auf die Tabelle um das Kontextmenü, über welches sie die Exportieren Option wählen können, zu öffnen. Nach Auswahl dieses Menüpunktes erscheint das Notizbuch Tabelle exportieren. Folgen Sie einfach den Schritten des Notizbuchs um den Vorgang fertigzustellen.

9.2 Das IMPORT Dienstprogramm

Das IMPORT Dienstprogramm wird, wie bereits zuvor beschrieben, verwendet um Daten aus einer Datei eine Tabelle zu laden. Tatsächlich wird im Hintergrund ein SQL INSERT Vorgang durchgeführt. Da ein INSERT Vorgang durchgeführt wird, sind Trigger aktiviert, alle Constraints werden sofort erzwungen und der Datenbank Pufferpool verwendet. Das folgende Beispiel lädt alle Daten der IXF formatierten Datei *employee.ixf* in die Tabelle *employee_copy*. Die REPLACE_CREATE Option ist eine von vielen Optionen des IMPORT Dienstprogramms. Diese Option ersetzt, falls diese Tabelle vor dem Einsatz des IMPORT Dienstprogramms bereits existiert, den Inhalt der Tabelle *employee_copy*. Wir empfehlen, das untenstehende Beispiel auszuprobieren, Sie sollten jedoch zuvor das EXPORT Dienstprogramm des vorhergehenden Abschnitts ausgeführt haben.

```
IMPORT FROM employee.ixf OF IXF
REPLACE_CREATE
INTO employee_copy
```

Falls Sie es vorziehen die DB2-Steuerzentrale zu verwenden, kann das IMPORT Dienstprogramm durch Auswahl einer Tabelle und anschließendem Rechtsklick auf die Tabelle, wie in Abbildung 9.3 gezeigt, gestartet werden.

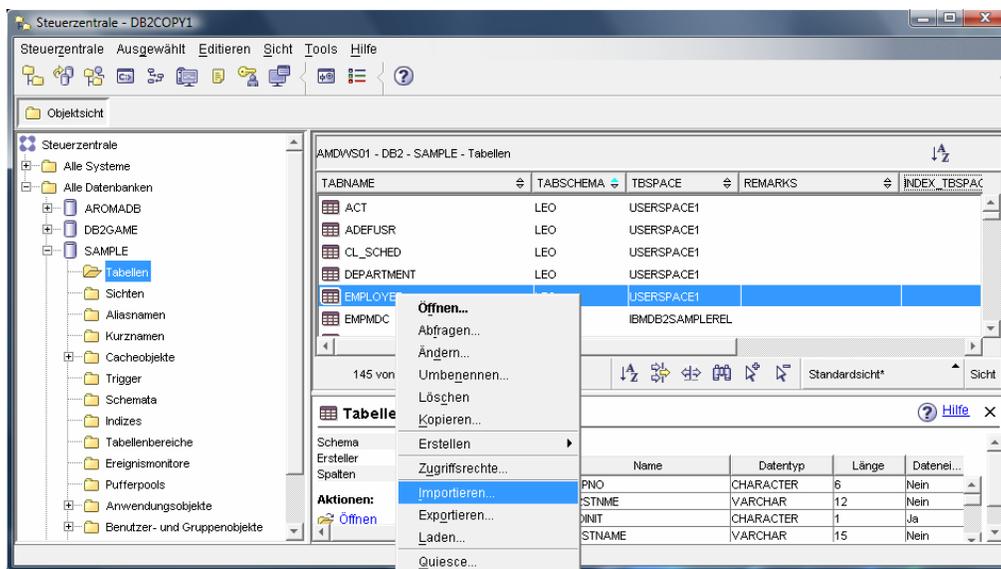


Abbildung 9.3 – Starten des Importieren Notizbuchs

9.3 Das LOAD Dienstprogramm

Das LOAD Dienstprogramm ist eine schnellere Alternative um Daten in eine Tabelle zu laden. Wie zuvor besprochen umgeht das LOAD Dienstprogramm DB2. Dies hat zur Folge das Trigger nicht aktiviert sind, der Pufferpool nicht verwendet wird und Constraints zwar erzwungen werden können, jedoch nur in einem separat auszuführenden Schritt. Andererseits ist ein LOAD Vorgang schneller als IMPORT, da dieser Daten Low-Level lädt und direkt auf die Datenseiten der Festplatte zugreift. Dies erfolgt in drei Phasen: LOAD, BUILD und DELETE.

Das nachfolgende Beispiel lädt alle Daten der IXF formatierten Datei *employee.ixf* in die Tabelle *employee_copy*. Die REPLACE Option ist eine von vielen verfügbaren LOAD Optionen. Hier wird REPLACE verwendet um alle Daten der Tabelle *employee_copy* zu ersetzen.

```
LOAD FROM employee.ixf OF IXF
  REPLACE INTO employee_copy
```

Nachdem Ausführen des obigen Befehls (was Sie ausprobieren können) ist der Tabellenbereich, in welchem ihre Tabelle definiert ist, möglicherweise in den Zustand CHECK PENDING gesetzt. Dies hat zur Folge, dass Sie zur Sicherstellung der Datenkonsistenz möglicherweise den Befehl SET INTEGRITY ausführen müssen. Das folgende Beispiel zeigt Ihnen die Vorgehensweise

```
SET INTEGRITY FOR employee_copy
  ALL IMMEDIATE UNCHECKED
```

Sollten Sie es vorziehen die DB2-Steuerzentrale zu verwenden können Sie die LOAD und SET INTEGRITY Notizbücher, wie aus den Abbildungen 9.4 bzw. 9.5 hervorgeht, über die DB2-Steuerzentrale starten.

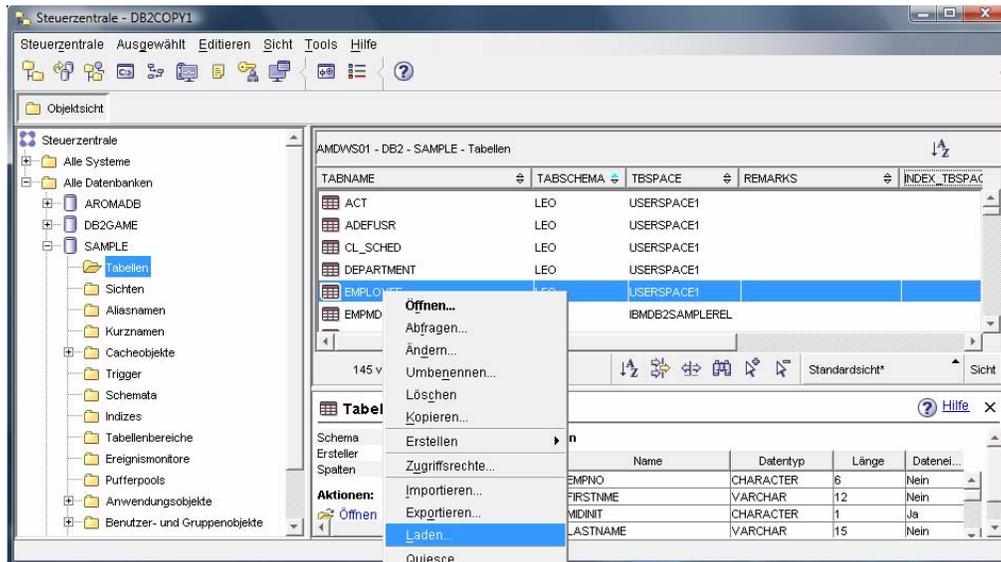


Abbildung 9.4 – Starten des Laden Notizbuchs

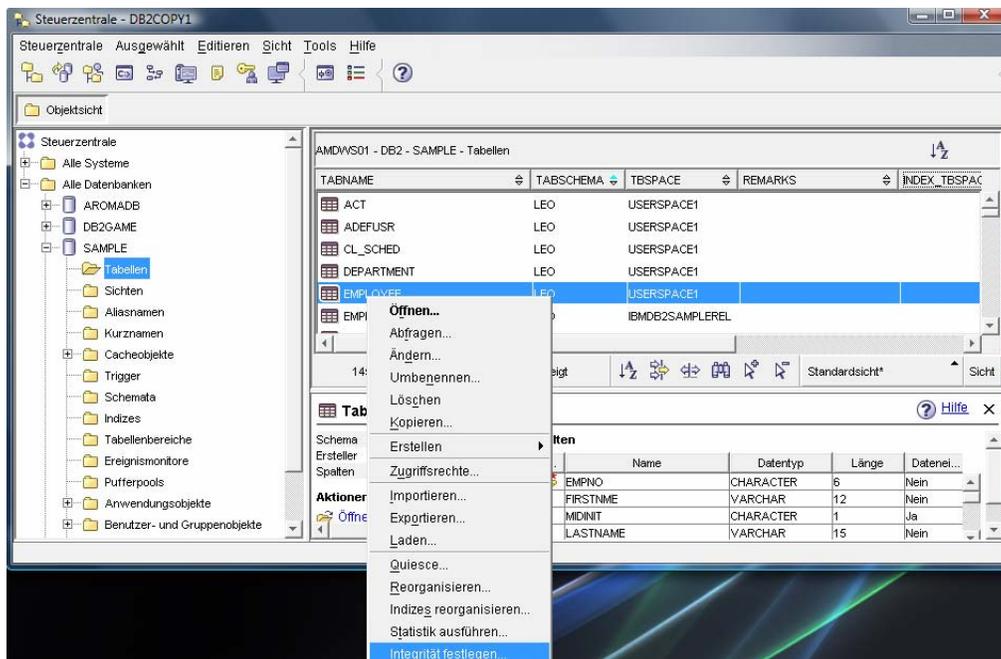


Abbildung 9.5 – Starten des Integrität festlegen Notizbuchs

9.4 Das db2move Dienstprogramm

Die EXPORT, IMPORT und LOAD Dienstprogramme verarbeiten nur einzelne Tabellen. Obwohl es möglich wäre ein Skript zu schreiben, welches diese Befehle für jede Tabelle in einer Datenbank enthält, ermöglicht dies das zusätzliche Dienstprogramm *db2move*. Das *db2move* Dienstprogramm arbeitet ausschließlich mit IXF Dateien und die Dateinamen werden automatisch von *db2move* generiert. Die Beispiele unten veranschaulichen wie *db2move* mit den *export* und *import* Optionen am Beispiel der Datenbank *SAMPLE* verwendet wird.

```
db2move sample export
db2move sample import
```

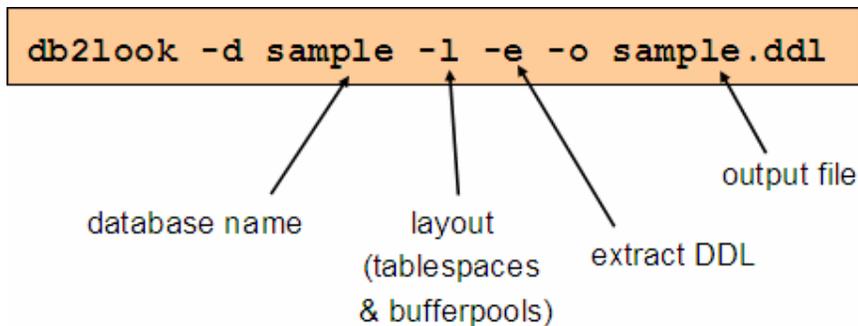
Die DB2-Steuerzentrale bietet keine *db2move* Option.

9.5 Das db2look Dienstprogramm

Während die EXPORT, IMPORT, LOAD und *db2move* Dienstprogramme es Ihnen erlauben Daten von einer Tabelle in eine andere Tabelle zu übernehmen, entweder innerhalb einer Datenbank oder über mehrere Datenbanken, kann das *db2look* Dienstprogramm verwendet werden um DDL Befehle, Datenbankstatistiken und die Eigenschaften des Tabellenbereichs einer Datenbank zu extrahieren um diese in einer Skript Datei zu speichern, um dieses später auf einem anderen System auszuführen. Falls Sie beispielsweise eine Datenbank, welche auf einem DB2 Server unter Linux läuft, klonen möchten um diese auf einem DB2 Server unter Windows zu verwenden können sie zuerst das *db2look* Dienstprogramm auf dem DB2 Server unter Linux ausführen um die Struktur der Datenbank in eine Skriptdatei zu speichern. Dann würden Sie das Skript auf den DB2 Windows Server kopieren und es ausführen um die geklonte Datenbank aufzubauen. Zu diesem Zeitpunkt wurde die Struktur der Datenbank geklont. Im nächsten Schritt würden Sie das *db2move* Dienstprogramm mit der *export* Option auf dem DB2 Linux Server ausführen und anschließend alle generierten Dateien auf den DB2 Windows Server kopieren und schließlich das *db2move* Kommando mit der *import* oder *load* Option ausführen. Nachdem dies durchgeführt wurde ist Ihre Datenbank vollständig von einem Server auf einen anderen Server einer anderen Plattform geklont.

Das obige Szenario wird benötigt wenn mit Datenbanken auf unterschiedlichen Plattformen, wie Linux und Windows, gearbeitet wird. Falls beide Server unter derselben Plattform laufen, werden Sie möglicherweise die Dienstprogramme *BACKUP* und *RESTORE* verwenden, da diese den Vorgang vereinfachen und nicht so komplex sind. Die *BACKUP* und *RESTORE* Dienstprogramme werden ausführlicher in einem späteren Kapitel dieses Buches besprochen.

Das nachfolgende Beispiel extrahiert die Tabellenbereiche und das Pufferpool Layout, zusammen mit den DDL Anweisungen aus der *SAMPLE* Datenbank und speichert diese in der Datei *sample.dll*. Wir ermuntern Sie den untenstehenden Befehl auszuführen und den Inhalt der Textdatei "sample.dll" anzusehen.



Das db2look Dienstprogramm besitzt zu viele Optionen um diese in diesem Buch zu beschreiben. Sie können den Schalter `-h` verwenden um eine Kurzbeschreibung der verfügbaren Optionen zu erhalten:

```
db2look -h
```

Das db2look Dienstprogramm kann auch, wie aus Abbildung 9.6 hervorgeht, über die DB2-Steuerzentrale gestartet werden.

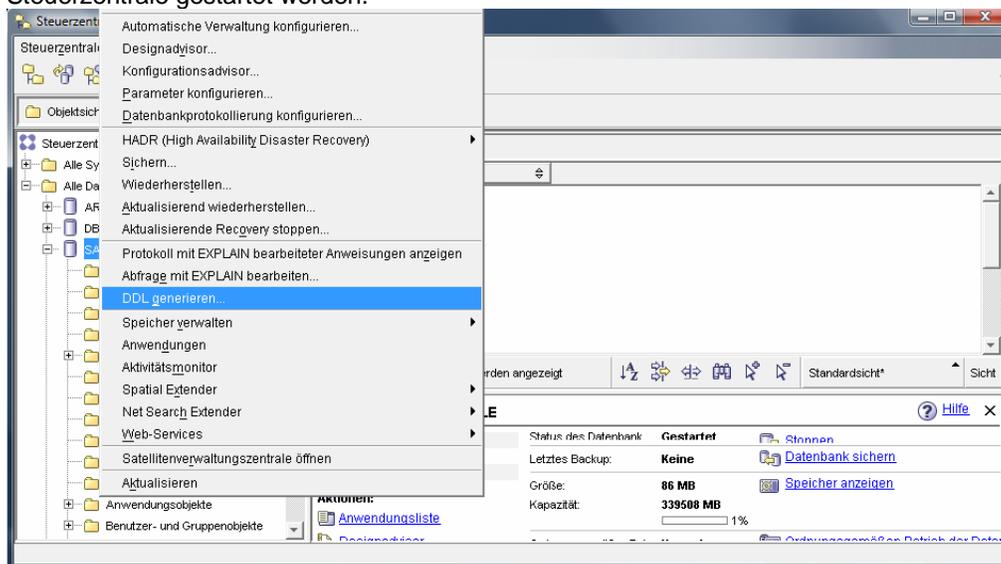


Abbildung 9.6 - DDL über die Steuerzentrale generieren

Wie aus Abbildung 9.6 ersichtlich, wählen Sie die Datenbank aus welcher DDL generiert werden soll, klicken rechts auf diese und wählen "DDL generieren". Das mehrere Optionen anzeigende DDL generieren Notizbuch wird, wie gezeigt in Abbildung 9.7, geöffnet.

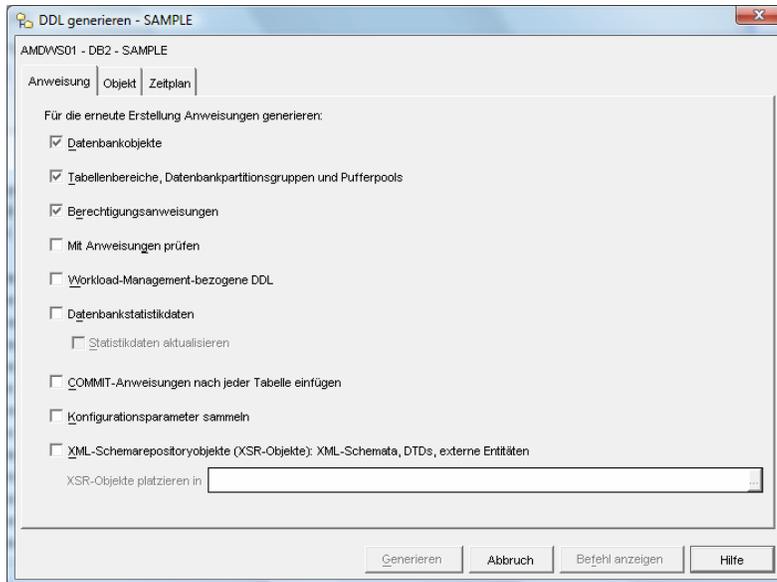


Abbildung 9.7 – Extrahieren von DDL über die Steuerzentrale

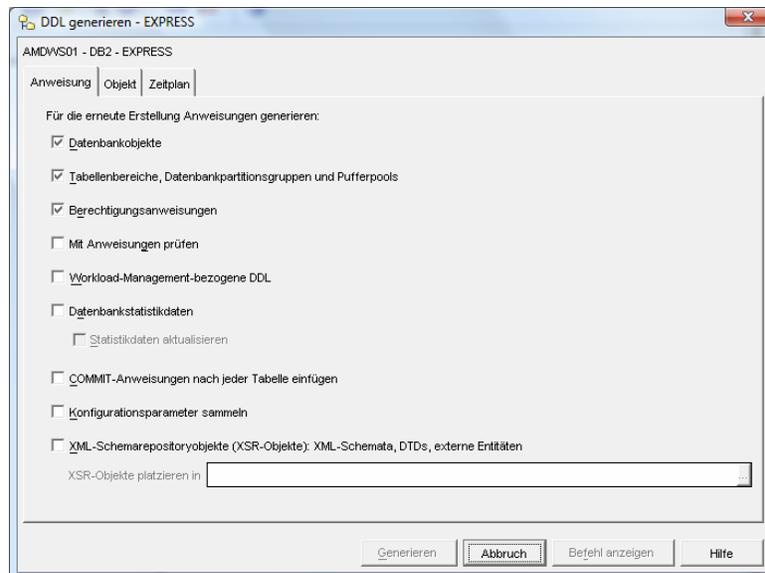
Übung 8 – Extrahieren der DDL der EXPRESS Datenbank

Ziel

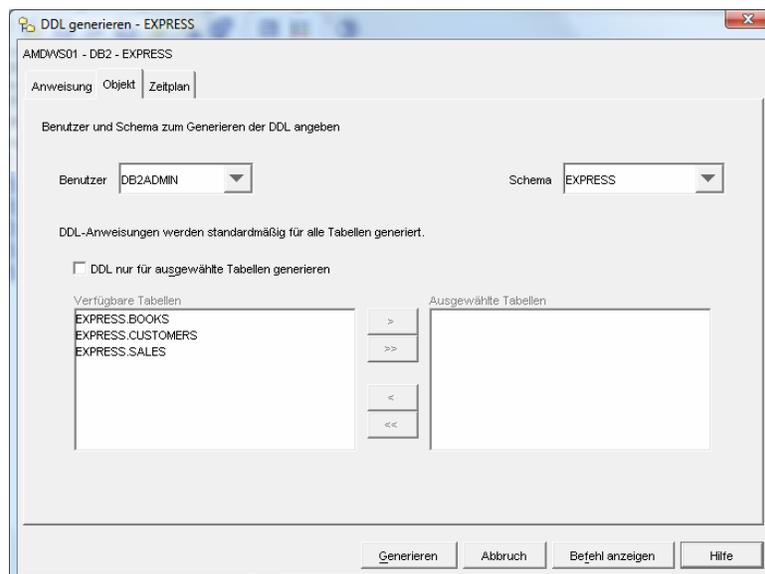
Wenn Sie eine Datenbank klonen sollte es ihr Ziel sein die Wiederherstellung der Datenbank so einfach und reproduzierbar wie möglich zu machen. Dies wird üblicherweise mit SQL Skripten erzielt, welche sofort nach der DB2 Installation ausgeführt werden können. In dieser Übung werden Sie die Objekt Definitionen der *EXPRESS* Datenbank (kreiert in Übung 2) mittels der DB2-Steuerzentrale extrahieren.

Vorgehensweise

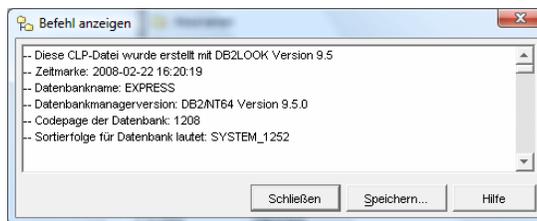
1. Öffnen Sie die DB2-Steuerzentrale.
2. Rechtsklicken Sie auf die *EXPRESS* Datenbank in der Objektsicht und wählen aus dem Kontextmenü *DDL generieren* um das *DDL generieren* Notizbuch zu öffnen.
3. Im *DDL generieren* Notizbuch geben Sie, wie unten gezeigt, die Optionen für das generierte DDL an. Falls Sie zusätzliche Objekte, wie z.B. Tabellenbereiche, Pufferpools usw., in ihrer Umgebung erstellt haben, werden diese hier ausgewählt. Da Sie keine Objekte dieses Typs erstellt haben, werden die Häkchen aus den Checkboxen entfernt. Datenbank Statistiken wurden nicht eingeschlossen, da die Produktionsumgebung möglicherweise einen anderen Satz Statistiken enthält als die Entwicklungsumgebung. Gleichmaßen werden Konfigurationsparameter ebenfalls unterschiedlich sein. In Ihrer eigenen Umgebung, falls alles exakt so wie es implementiert wird konfiguriert ist, können sie wahlweise zusätzliche Optionen angeben.



4. Gehen Sie zur *Objekt* Registerkarte. Hier besteht die Möglichkeit die Objekte, für welche DDL generiert werden soll, auszuwählen. In diesem Fall wählen Sie den Benutzer und das Schema welches sie für die Erstellung all ihrer Objekte verwendet haben und generieren DDL für alle Objekte in diesem Schema. Klicken Sie anschließend auf die *Generieren* Schaltfläche um die DDL Generierung zu starten.



5. Prüfen Sie das erstellte DDL. Das Resultat des vorhergehenden Schritts ist ein einziges Skript mit allen SQL Anweisungen der ausgewählten Objekte. Nunmehr werden wir das Skript in logische Gruppen organisieren.
6. Erstellen Sie ein neues Verzeichnis mit dem Namen `C:\express` im Dateisystem und speichern die generierte DDL Datei in dieses neue Verzeichnis unter dem Namen `schema.ddl`. (Klicken Sie auf die *Speichern* Schaltfläche)



7. Öffnen Sie die gespeicherte neue Datei im Befehlseditor. (Hinweis: Im Befehlseditor wählen Sie *Ausgewählt => Öffnen*)
8. Obwohl wir tatsächlich ausschließlich das DDL der Tabellen benötigen, werden Sie feststellen, dass ebenfalls DDL für andere Datenbankobjekte enthalten ist. Kopieren Sie alle CREATE TRIGGER Anweisungen in eine separate neue Datei mit dem Namen `triggers.ddl`. Auch wenn wir nur einen Trigger definiert haben empfiehlt es sich Objekte nach Typ zu gruppieren.
9. Fürs Erste empfehlen wir nachfolgenden Anweisungen zu entfernen:
 - CONNECT TO Datenbank Anweisungen
 - DISCONNECT Anweisungen

Nunmehr sollten zwei Skripte vorhanden sein:

DDL für Tabellen, Sichten, Indizes und Constraints
`C:\express\schema.ddl`

DDL für Trigger
`C:\express\triggers.ddl`

10. Bereinigen Sie das Skript für die Implementierung:
 - Entfernen Sie alle unnötigen Kommentare (z.B. `-- CONNECT TO...`)

- Teilen Sie Funktionen und Prozeduren in getrennte Dateien (hilfreich bei vielen Funktionen und Prozeduren) auf. Vielleicht wollen Sie diese nach Funktion oder Anwendung gruppieren (z.B. `billing.ddl`, `math.ddl`, `stringfunc.ddl` usw.)
11. Vielleicht haben Sie festgestellt das das Sonderzeichen `@` zur Kennzeichnung des Endes von Triggern, Funktionen und Prozeduren verwendet wird. Dies ist erforderlich um das Ende einer `CREATE <object>` Anweisung, im Gegensatz zu einer Prozeduranweisung, innerhalb des Objekts anzuzeigen.

10

Kapitel 10 – Datenbanksicherheit

Dieses Kapitel beschreibt wie Sicherheit in DB2 behandelt wird. Abbildung 10.1 gibt einen grundlegenden Überblick.

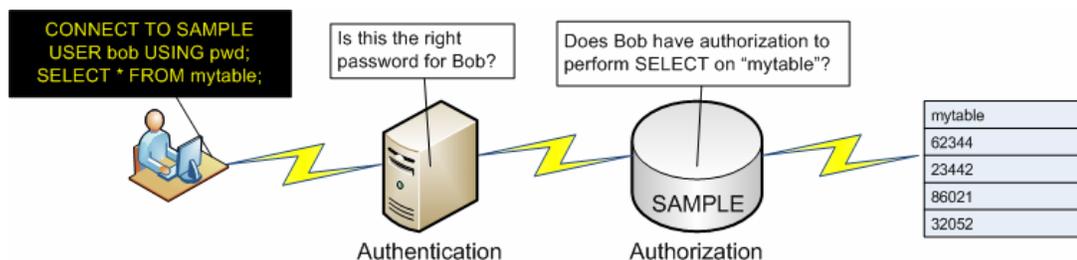


Abbildung 10.1 – Übersicht über die DB2 Sicherheit

Wie in Abbildung 10.1 gezeigt, besteht die DB2 Sicherheit aus zwei Teilen:

Identitätsprüfung

Dies ist der Vorgang welcher die Benutzeridentität validiert. Die Identitätsprüfung wird von Sicherheitsmechanismen außerhalb DB2 durchgeführt (typischerweise durch ein Betriebssystem, eine Identitätsprüfungsmethode oder ein speziell angefertigtes Plug-In zur Identitätsprüfung). Betriebssystembasierte Identitätsprüfung ist der Standard. Bei Verwendung betriebssystembasierter Identitätsprüfung werden die Benutzerkennung und Passwort an die Datenbank weitergereicht (z.B. als Teil einer connect Anweisung). Der Datenbankserver ruft dann die Betriebssystemidentitätsprüfung zur Validierung von Benutzerkennung und Passwort auf.

Berechtigungsprüfung

Zu diesem Zeitpunkt prüft DB2 ob der berechtigte Benutzer die angeforderte Operation ausführen darf. Die Berechtigungsinformationen werden im DB2 Katalog und in einer DBM Konfigurationsdatei gespeichert.

Wie Sie zum Beispiel in Abbildung 10.1 dargestellt, verbindet sich Benutzer "bob" mit der SAMPLE Datenbank unter Verwendung der nachfolgenden Anweisung:

```
CONNECT TO sample USER bob USING pwd
```

Beide, "bob" (Benutzerkennung) und "psw" (Passwort), werden an das Betriebssystem oder eine externe Authentifizierungsroutine weitergeleitet um die Authentifizierung zu

bestätigen und sicherzustellen, dass der Benutzer "bob" bereits definiert ist und das Passwort für den Benutzer übereinstimmt. Falls dieser Teil erfolgreich ist, übergibt das Betriebssystem die Sicherheitskontrolle an DB2. Als nächstes, wenn Benutzer "bob" eine Anweisung wie die folgende ausführt:

```
SELECT * FROM mytable
```

Nunmehr übernimmt DB2 die Sicherheitskontrolle zur Ausführung der Rechteüberprüfung und Sicherstellung dass Benutzer "bob" SELECT Berechtigung auf die Tabelle "mytable" besitzt. Falls die Rechteüberprüfung fehlschlägt gibt DB2 eine Fehlermeldung zurück, anderenfalls wird die Anweisung auf die Tabelle "mytable" angewandt.

Hinweis:
 Für weitere Informationen über die Arbeit mit dem DB2 Sicherheitssystem können Sie dieses Video ansehen:
<http://www.channeldb2.com/video/video/show?id=807741:Video:4267>

10.1 Identitätsprüfung

Obwohl die tatsächliche Identitätsprüfung durch das Betriebssystem vorgenommen (oder ein externes Sicherheitsdienstprogramm) entscheidet DB2 auf welcher Stufe die Identitätsprüfung erfolgt.

Der DBM CFG Parameter AUTHENTICATION, gesetzt auf dem DB2 Server, besitzt eine Reihe möglicher Werte. Sollte z.B. der Parameter auf Server (Standardeinstellung) gesetzt sein, erfolgt die Identitätsprüfung durch das Betriebssystem oder ein externes Dienstprogramm auf dem Server. Ist jedoch AUTHENTICATION auf CLIENT gesetzt, erfolgt die Identitätsprüfung durch das Betriebssystem oder ein externes Dienstprogramm auf dem Client. Dies zeigt Abbildung 10.2.

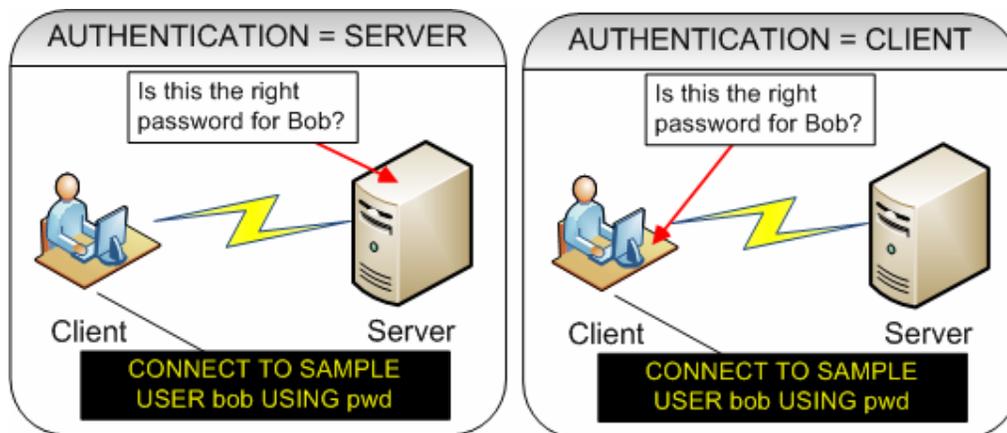


Abbildung 10.2 – Wo die Identitätsprüfung stattfindet

Der AUTHENTICATION Parameter kann auf jeden Wert der Tabelle 10.1 gesetzt werden.

Befehl	Beschreibung
SERVER (Standard)	Die Identitätsprüfung findet auf dem Server statt
CLIENT	Die Identitätsprüfung findet auf dem Client statt
SERVER_ENCRYPT	Wie SERVER, jedoch mit der Ausnahme das Benutzerkennungen und Passwörter verschlüsselt werden
KERBEROS	Identitätsprüfung unter Verwendung des Kerberos Sicherheitsmechanismus
SQL_AUTHENTICATION_DATAENC	Server Identitätsprüfung und die Verbindung muss zusätzlich Datenverschlüsselung verwenden
SQL_AUTHENTICATION_DATAENC_CMP	Wie oben, mit der Ausnahme das Datenverschlüsselung nur bei Verfügbarkeit verwendet wird
GSSPLUGIN	Die Identitätsprüfung verwendet einen externen GSS API-basierten Mechanismus

Tabelle 10.1 – Gültige AUTHENTICATION Parameter Werte

10.2 Berechtigung

Die Berechtigung besteht aus den in DB2 Systemtabellen gespeicherten Rechten und Berechtigungen, welche von DB2 verwaltet werden.

Eine Berechtigung gestattet einem Benutzer die Ausführung eines einzelnen Typs eines Vorgangs gegen die Datenbank, wie CREATE, UPDATE, DELETE, INSERT, usw.

Bei einem Recht handelt es sich um eine vordefinierte Rolle bestehend aus mehreren Berechtigungen. Abbildung 10.3 zeigt die verschiedenen Berechtigungen und Rechte in DB2.

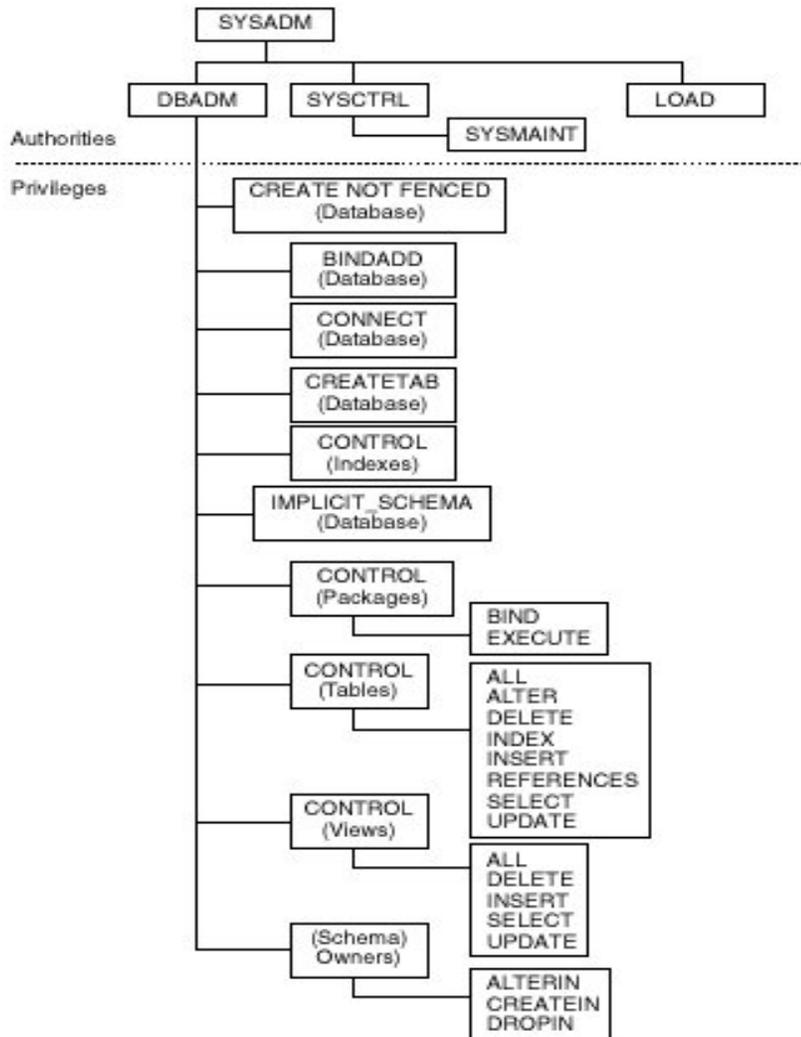


Abbildung 10.3 – Rechte und Berechtigungen

Tabelle 10.2 zeigt die abhängig von der Berechtigung ausführbaren Funktionen. Wie Sie sehen, besitzt SYSADM die meisten Berechtigungen, während SYSMON die wenigsten besitzt.

<i>Funktion</i>	SYSADM	SYSCTRL	SYSMAINT	SYSMON	DBADM	LOAD
Update DBM CFG	J					
Grant/revoke DBADM	J					
Establish/change SYSCTRL	J					
Establish/change SYSMAINT	J					
Establish/change SYSMON	J					
Force users off database	J	J				
Create/drop database	J	J				
Restore to new database	J	J				
Update DB CFG	J	J	J			
Backup database/table space	J	J	J			
Restore to existing database	J	J	J			
Perform roll-forward recovery	J	J	J			
Start/stop instance	J	J	J			
Restore table space	J	J	J			
Run trace	J	J	J	J		
Obtain monitor snapshots	J	J	J			
Query table space state	J	J	J			
Prune log history files	J	J	J			
Quiesce table space	J	J	J		J	J
LOAD tables	J				J	J
Set/unset check pending state	J				J	
Create/drop event monitors	J				J	

Tabelle 10.2 - DB2 Rechte und Berechtigungen

Um einer Gruppe die SYSADM, SYSCTRL oder SYSMAINT Berechtigung zuzuweisen können die DBM CFG Parameter SYSADM_GROUP, SYSCTRL_GROUP und SYSMAINT_GROUP einer Betriebssystemgruppe zugewiesen werden.

Um zum Beispiel der Betriebssystemgruppe "db2admns" die SYSADM Berechtigung zuzuweisen können Sie diesen Befehl verwenden:

```
update dbm cfg using SYSADM_GROUP db2admns
```

Jede DB2 Instanz besitzt ihre eigene Definition der Berechtigungsgruppen.

Unter Windows sind diese Parameter standardmäßig leer, was zur Folge hat, dass die lokale Gruppe der Windows Administratoren SYSADM sind. Unter Linux gehört die Gruppe der Besitzer der Instanz zur standardmäßigen SYSADM Gruppe.

10.3 DBADM Berechtigung

Bei der DBADM (DataBase Administrator) Berechtigung handelt es sich um den Super User für diese Datenbank. Da es sich nicht um eine Berechtigung auf Instanz-Ebene handelt wurde sie nicht im vorigen Abschnitt aufgeführt. Um die DBADM Berechtigung zu erteilen verwenden Sie die GRANT Anweisung, wie im Beispiel unten gezeigt.

```
connect to sample
grant DBADM on database to user <userid>
```

Im obigen Beispiel müssen Sie sich zuerst mit der Datenbank, hier die "sample" Datenbank, verbinden um DBADM einem Benutzer zuzuweisen. Um die DBADM Berechtigung zuzuweisen müssen Sie die SYSADM Berechtigung besitzen.

Bitte beachten Sie, dass ein DBADM keine Tabellenbereiche erstellen kann, obwohl es sich um Objekte innerhalb der Datenbank handelt, da Tabellenbereiche die physikalischen Bereiche Container (die Festplatte) und den Pufferpool (Speicher) des Systems verwenden.

10.4 Die PUBLIC Gruppe

DB2 definiert eine interne Gruppe namens PUBLIC. Jeder durch das Betriebssystem oder eine Netzwerkzugriffskontrolle identifizierte Benutzer ist implizit Mitglied der Gruppe PUBLIC. Wenn eine Datenbank erstellt wird, werden automatisch bestimmte Berechtigungen der Gruppe PUBLIC zugewiesen:

- CONNECT,
- CREATETAB,
- IMPLICIT SCHEMA,
- BINDADD

Für zusätzliche Sicherheit empfehlen wir der Gruppe PUBLIC alle Berechtigungen, wie nachfolgend aufgeführt, mit REVOKE zu entziehen:

```
REVOKE CONNECT          ON DATABASE FROM PUBLIC
REVOKE CREATETAB        ON DATABASE FROM PUBLIC
REVOKE IMPLICIT_SCHEMA  ON DATABASE FROM PUBLIC
REVOKE BINDADD          ON DATABASE FROM PUBLIC
```

10.5 Die GRANT und REVOKE Anweisungen

Sowohl GRANT als auch REVOKE Anweisungen sind Bestandteil des SQL Standards und werden verwendet um Benutzern oder Gruppen Berechtigungen zu erteilen oder zu entziehen. Unten finden Sie einige Beispiele dieser Anweisungen:

Um USER1 die SELECT Berechtigung auf die Tabelle T1 zu erteilen:

```
GRANT SELECT ON TABLE T1 TO USER user1
```

Um GROUP1 alle Berechtigungen auf die Tabelle T1 zu erteilen:

```
GRANT ALL ON TABLE T1 TO GROUP group1
```

Um GROUP1 alle Berechtigungen auf die Tabelle T1 zu entziehen:

```
REVOKE ALL ON TABLE T1 FROM GROUP group1
```

Um Benutzer USER1 die EXECUTE Berechtigung auf die Prozedur p1 zu erteilen:

```
GRANT EXECUTE ON PROCEDURE p1 TO USER user1
```

Um USER1 die EXECUTE Berechtigung auf die Prozedur p1 zu entziehen:

```
REVOKE EXECUTE ON PROCEDURE p1 FROM USER user1
```

10.6 Prüfen der Berechtigungen und Zugriffsrechte

Am einfachsten lassen sich Berechtigungen und Zugriffsrechte über die DB2-Steuerzentrale prüfen. Abbildung 10.4 zeigt wie das Tabellenzugriffsrechte Notizbuch für die Tabelle EMPLOYEE über die DB2-Steuerzentrale gestartet wird.

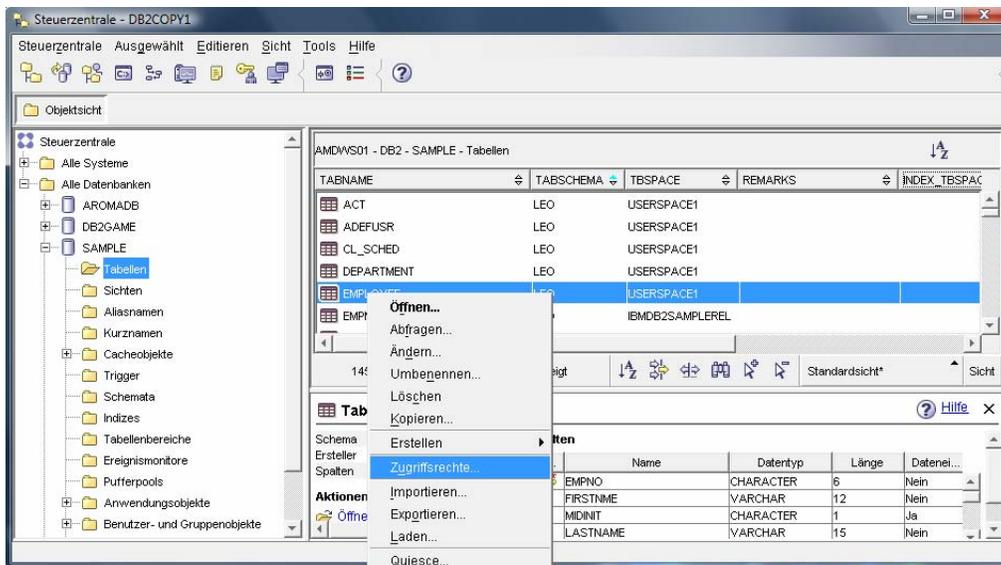


Abbildung 10.4 – Starten des Tabellenzugriffsrechte Notizbuchs

Wie in Abbildung 10.4 gezeigt, wählen Sie die in Frage kommende Tabelle, klicken rechts auf diese und wählen Tabellenzugriffsrechte. Nach der Auswahl erscheint, wie in Abbildung 10.5 gezeigt, der Tabellenzugriffsrechte Dialog. Diese Abbildung erläutert zusätzlich die verschiedenen Felder und Elemente des Dialogs.

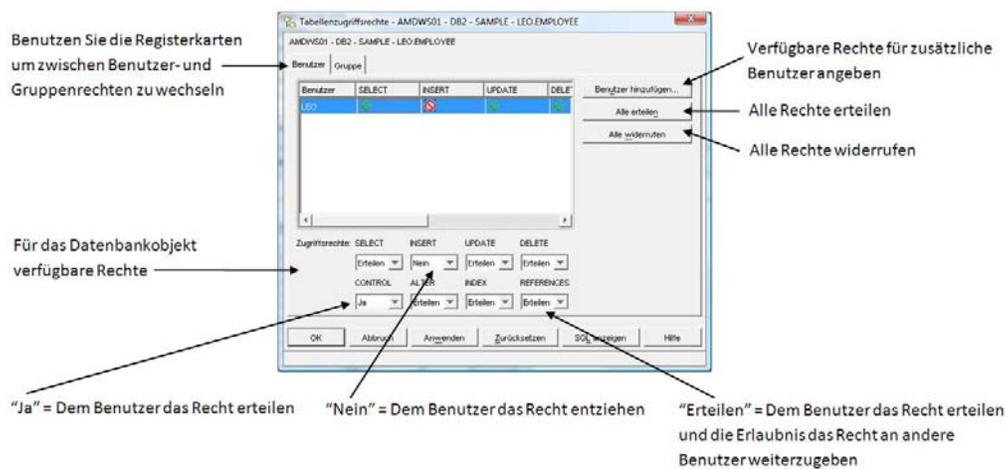


Abbildung 10.5 – Der Tabellenzugriffsrechte Dialog

Alternativ können Sie die DB2 SYSCAT Katalog Sichten, in welchen die die Zugriffsrechte betreffenden Informationen gespeichert sind, abfragen. Möchten Sie zum Beispiel wissen ob Benutzer DB2ADMIN SELECT Berechtigung auf die Tabelle T2 besitzt und zusätzlich herausfinden wer diese Berechtigung erteilt hat, können sie die folgende Abfrage verwenden:

```
SELECT grantor, grantee, selectauth
  FROM syscat.tabauth
 WHERE tabname = 'T2'
```

GRANTOR	GRANTEE	SELECTAUTH
ARFCHONG	DB2ADMIN	Y

Im obigen Beispiel erteilt Benutzer ARFCHONG die SELECT Berechtigung an Benutzer bzw. Gruppe DB2ADMIN.

10.7 Überlegungen zu Gruppenberechtigungen

Um die DB2 Administration einfacher zu machen, empfiehlt es sich Benutzer in Gruppen aufzunehmen und anschließend diesen Gruppen die erforderlichen Rechte zu erteilen.

Wenn einer Gruppe Rechte erteilt wurden erhalten alle Mitglieder der Gruppe implizite, durch die Gruppenmitgliedschaft vererbte, Rechte.

Wenn ein Benutzer aus seiner Gruppe entfernt wird, verliert dieser implizit alle Gruppenberechtigungen welche explizit erteilt wurden. Berechtigungen welche explizit einem Benutzer erteilt wurden müssen diesem auch explizit wieder entzogen werden.

Übung 9 – Erteilen und entziehen von Benutzerberechtigungen

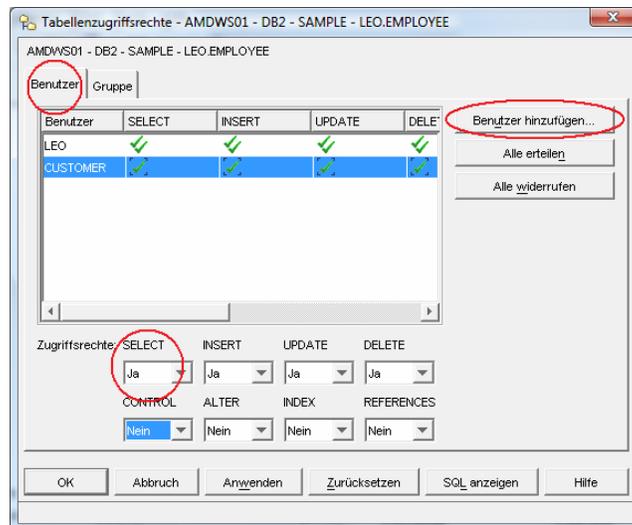
Ziel

Bisher haben wir das Instanz Administrator Konto (SYSADM) verwendet um alle Datenbank Befehle abzusetzen. Dieses Konto besitzt Vollzugriff auf alle Dienstprogramme, Daten und Datenbankobjekte. Es ist deshalb unumgänglich dieses Konto abzusichern, um unbeabsichtigtem oder vorsätzlichem Datenverlust vorzubeugen. In den meisten Fällen werden Sie Benutzerkonten oder Gruppen mit eingeschränkten Rechten anlegen. In dieser Übung werden Sie ein neues Benutzerkonto anlegen und diesem Benutzerkonto spezifische Rechte erteilen.

Vorgehensweise

1. Öffnen Sie die Windows Computerverwaltung durch einen Rechtsklick auf das *Mein Computer* Symbol auf dem Desktop und wählen aus dem Kontextmenü *Verwalten*.
2. Erweitern Sie den *System* Zweig im linken Fenster und anschließend öffnen Sie den Lokale *Benutzer und Gruppen* Ordner. Rechtsklicken Sie auf den *Benutzer* Ordner und wählen *Neuer Benutzer*.
3. Im *Neuer Benutzer* Dialog tragen Sie die folgenden Informationen ein: In das Feld *Benutzername* schreiben Sie "*customer*" und in das Feld *Vollständiger Name* schreiben Sie "*Customer1*". In das Feld *Beschreibung* schreiben Sie "*A typical bookstore customer*". In die Felder *Kennwort* und *Kennwort bestätigen* schreiben Sie "*ibmdb2*". Entfernen Sie das Häkchen vor der *Benutzer muss Kennwort bei der nächsten Anmeldung ändern* Option und klicken Sie auf die *Erstellen* Schaltfläche um den neuen Benutzer zu erstellen.

4. Bitte stellen Sie zuerst sicher, dass die erweiterte Ansicht der DB2-Steuerzentrale aktiviert ist. Um zur erweiterten Ansicht zu wechseln wählen Sie aus dem *Tools* Menü der DB2-Steuerzentrale den Eintrag *Steuerzentrale anpassen*. Wählen Sie dann *Erweitert* und klicken auf die *OK* Schaltfläche.
5. Erweitern Sie die Objektansicht der DB2-Steuerzentrale im linken Objektbaum bei *Alle Datenbanken > EXPRESS > Tabellen*.
6. Weisen Sie dem soeben erstellten neuen Benutzer die erforderlichen Benutzerrechte zu. In der Liste der Tabellen in der *EXPRESS* Datenbank klicken Sie rechts auf die *CUSTOMERS* Tabelle um den *Tabellenzugriffsrechte* Dialog anzuzeigen.
7. Klicken Sie auf *Benutzer hinzufügen* und wählen den soeben erstellten Benutzer *customer* aus der Liste der Benutzer. Klicken Sie auf die *OK* Schaltfläche um den *Benutzer hinzufügen* Dialog zu schließen.
8. Sie werden feststellen dass der Benutzer *customer* zwar zur Liste hinzugefügt, diesem jedoch noch keine Zugriffsrechte erteilt wurden. Um dem Benutzer *SELECT*, *INSERT*, *UPDATE* und *DELETE* Zugriffsrechte zu erteilen wählen Sie den Eintrag *Erteilen* in den Kombinationsfeldern. Ein Internet Kunde sollte seine Kontodaten ansehen, hinzufügen, aktualisieren und löschen können. Da der Benutzer keine weiteren Berechtigungen benötigt, werden diese nicht erteilt. Klicken Sie auf die *OK* Schaltfläche und den *Tabellenzugriffsrechte* Dialog zu schließen und die vorgenommenen Änderungen zu übernehmen.



9. Wiederholen Sie die Schritte 7 bis 9 für die Tabellen *BOOKS* und *SALES*. Für die *BOOKS* Tabelle erteilen Sie ausschließlich die *SELECT* Zugriffsberechtigung, da es dem Kunden nicht möglich sein sollte die Lagerbestandsdaten der Firma zu ändern. Bei der *SALES* Tabelle erteilen Sie die *SELECT* und *INSERT* Zugriffsrechte. Der Kunden sollte, da nur Firmenangestellte die Verkaufstransaktionen ändern können sollten, keine *DELETE* oder *UPDATE* Zugriffsrechte besitzen.

10. Verbinden Sie sich mit der Datenbank unter Verwendung der oben erstellten *customer* Benutzerkennung. Versuchen Sie mittels *SELECT* Daten aus der Tabelle *customers* auszuwählen. Was passiert? Versuchen Sie die Daten der *SALES* Tabelle mittels *DELETE* oder *UPDATE* zu aktualisieren. Was passiert?

In dieser Übung haben wir lediglich einen Benutzer erstellt, wohingegen Ihre Anwendung möglicherweise viele verschiedene Arten von Benutzern enthält. Experimenten Sie mit der Erstellung von Benutzern und der Zuweisung von Tabellenzugriffsrechten. Sie können ebenfalls Gruppen von Benutzern erstellen und diesen Tabellenzugriffsrechte erteilen, statt diese jedem Benutzer individuell zu erteilen.

11

Kapitel 11 – Sicherung und Wiederherstellung

In diesem Kapitel besprechen wir die DB2 Datenbank Protokollierung, wie eine vollständige oder partielle Kopie Ihrer Datenbank mit dem BACKUP Dienstprogramm erstellt, und ihre Daten mit dem RESTORE Dienstprogramm wiederhergestellt werden können.

Hinweis:

Für weitere Informationen bezüglich Protokollierung, Sicherung und Wiederherstellung können Sie sich dieses Video ansehen:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4282>

11.1 Datenbank Protokollierung

Falls Sie mit einem Texteditor arbeiten müssen sie, um sicherzustellen dass das Dokument gespeichert wird, auf die "Speichern" Schaltfläche klicken. In der Datenbank Welt bewirkt die COMMIT Anweisung genau dies. Jedesmal wenn eine COMMIT Anweisung ausgeführt wird, stellen Sie sicher dass alle Änderungen an den Daten irgendwo gespeichert werden.

Ähnlich ist es wenn Sie mit einem Textdokument arbeiten und manchmal in der Statusleiste unten rechts "Automatische Speicherung" sehen. In der Datenbank Welt geschieht dies ebenfalls, da alle auf die Datenbank durchgeführten Vorgänge, wie UPDATE, INSERT oder DELETE, irgendwo gespeichert werden.

Das "irgendwo" in den vorgehenden Absätzen bezieht sich auf die Datenbank Protokolle. Datenbank Protokolle werden auf der Festplatte gespeichert und verwendet um Aktionen von Transaktionen aufzuzeichnen. Im Falle eines System- oder Datenbankausfalls werden die Datenbankprotokolle benutzt um mit COMMIT abgeschlossene Transaktionen im Rahmen der Wiederherstellung der Datenbank nachzufahren und wieder einzuspielen.

Abbildung 11.1 zeigt eine grafische Übersicht über die Protokollierung während Sie mit einer Datenbank arbeiten.

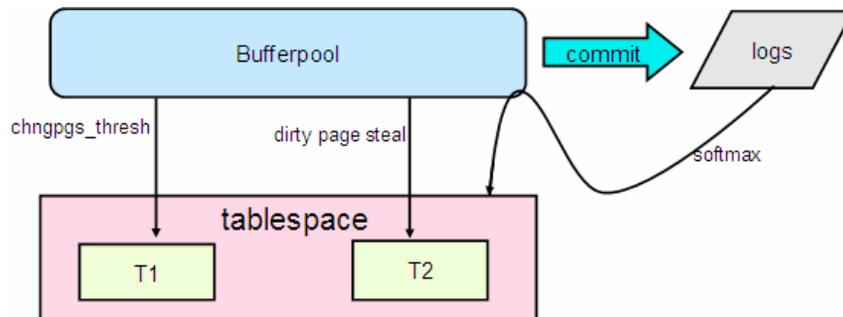


Abbildung 11.1 – Datenbank Protokollierung

In Abbildung 11.1 sehen wir einen Tabellenbereich und Protokolle. Beide befinden sich auf Festplatten, obwohl wir empfehlen diese nicht auf derselben Festplatte zu speichern. Falls z.B. ein UPDATE vorgenommen wird, werden die Speicherseiten der in Frage kommenden Spalte(n) in den Pufferpool (Speicher) übertragen. Die Aktualisierungsänderungen werden durch Pufferpool vorgenommen und die alten und neuen Werte werden in Protokollen gespeichert, manchmal sofort und manchmal wenn ein Protokollpuffer voll ist. Falls nach einem UPDATE ein COMMIT abgesetzt wurde, werden die alten und neuen Werte sofort in Protokollen gespeichert. Dieser Vorgang wiederholt sich für viele weitere auf die Datenbank abgesetzte SQL Anweisungen. Nur wenn bestimmte Bedingungen zutreffen, wie das Erreichen des Schwellenwerts für Seitenänderungen, wie im Parameter CHNGPGS_THRES spezifiziert, werden die Seiten im Pufferpool "externalisiert" oder auf die Festplatte des Tabellenbereichs geschrieben. Der CHNGPGS_THRES Parameter überwacht den Prozentsatz der "dirty pages", d.h. Pufferpoolseiten welche Änderungen beinhalten.

Aus Performancegründen macht es wenig Sinn zwei Schreibvorgänge für jedes COMMIT auszuführen: Einen um in das Protokoll zu schreiben und einen weiteren um in den Tabellenbereich zu schreiben. Dies ist der Grund warum eine "Externalisierung" von Daten auf die Festplatte des Tabellenbereichs nur dann stattfindet wenn Parameter wie z.B. der "CHNGPGS_THRES" Schwellenwert erreicht werden.

11.2 Protokollarten

Es existieren zwei Arten von Protokollen:

Primäre Protokolle

Diese werden im Voraus zugewiesen und die Anzahl der verfügbaren primären Protokolle wird durch den LOGPRIMARY DB CFG Parameter festgelegt.

Sekundäre Protokolle

Diese werden, falls von DB2 benötigt, dynamisch zugewiesen. Die maximale Anzahl der sekundären Protokolle wird durch den DB CFG Parameter LOGSECOND festgelegt. Die dynamische Zuweisung von Protokollen ist teuer, für Tag-für-Tag Operationen sollten Sie sicherstellen, dass innerhalb des zugewiesenen primären Protokolls bleiben. Sekundäre Protokolldateien werden bei Beendigung aller Verbindungen zur Datenbank gelöscht.

Unendliche Protokollierung ist möglich wenn Sie den LOGSECOND Wert auf -1 setzen. Wie dem auch sei, dies ist nicht empfehlenswert und der Dateisystemspeicherplatz wird möglicherweise vollständig verbraucht.

11.3 Arten der Protokollierung

Es existieren zwei Arten der Protokollierung: Umlaufprotokollierung (Standard) und Archivprotokollierung.

11.3.1 Umlaufprotokollierung

Abbildung 11.2 demonstriert die Funktionsweise der Umlaufprotokollierung.

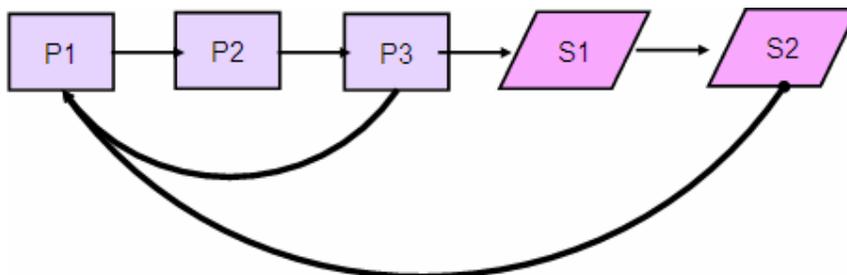


Abbildung 11.2 – Arbeiten mit primären und sekundären Protokollen

In Abbildung 11.2 sehen Sie 3 primäre Protokolldateien. Wir können daher annehmen, dass LOGPRIMARY auf den Wert 3 gesetzt wurde. Der Einfachheit halber gehen wir davon aus, dass nur eine Transaktion in diesem Beispiel durchgeführt wird. Während der Ausführung der Transaktion wird zuerst die Protokolldatei P1 und dann die Protokolldatei P2 gefüllt. Nachdem ein COMMIT der Transaktion erfolgt ist und die Information der Transaktion im Tabellenbereich auf Festplatte geschrieben wurde, kann P1 und P2 überschrieben werden, da diese Information dann nicht mehr für eine Absturz Wiederherstellung (Crash Recovery; welche später näher erläutert wird) benötigt wird. Falls die Transaktion aber weitergehen und P1, P2, P3 benutzt und immer noch Platz für Protokolle benötigt wird, weil weder ein COMMIT abgesetzt wurde noch die Daten externalisiert wurden, wird eine sekundäre Protokolldatei (S1 in dem Bild) dynamisch zugewiesen.

Falls die Transaktion andauert, werden solange weitere sekundäre Protokolldateien zugewiesen, bis das Maximum LOGSECOND erreicht ist. Sollten mehr Protokolle erforderlich sein, weist eine Meldung den Benutzer darauf hin, dass die Protokolldateien vollständig gefüllt sind und die Transaktion wird rückgängig gemacht (Rollback).

11.3.2 Archivprotokollierung oder Protokollspeicherung

Bei der Archivprotokollierung, auch bekannt als erhaltende Protokollspeicherung, werden die Protokolldateien nicht überschrieben, sondern Online oder Offline gehalten. Online archivierte Protokolldateien werden bei den aktiven Protokolldateien gehalten, welche für die Absturz Wiederherstellung (Crash Recovery) benötigt werden. Offline archivierte

Protokolldateien werden auf ein anderes Medium, z.B. Band, ausgelagert, welches mit USEREXIT Routinen realisiert werden kann. Um die Archivprotokollierung zu aktivieren setzen Sie den LOGRETAIN DB CFG Parameter auf YES.

Archivprotokollierung wird normalerweise auf Produktionssystemen benutzt und, da die Protokolldateien erhalten bleiben, ermöglicht diese in den meisten Fällen die Wiederherstellung der Datenbank auf den frühestmöglichen Stand der ältesten Protokolldatei. Mit Archivprotokollierung kann ein DBA eine Wiederherstellung (in bestimmtem Umfang) nach von Menschen verursachten Fehlern vornehmen. Sollte z.B. der Benutzer eines Systems versehentlich eine mehrere Tage dauernde fehlerhafte Transaktion starten, und das Problem wird erst zu einem späteren Zeitpunkt entdeckt, dann könnte der DBA das System auf den Zustand vor dem Auftreten des Fehlers wiederherstellen. Wie dem auch sei, sind wahrscheinlich weitere manuelle Eingriffe erforderlich um die Transaktion erneut korrekt zu starten.

Archivprotokollierung ist für die aktualisierende Wiederherstellung (roll forward recovery) und On-Line Sicherung (on-line backup) erforderlich. Abbildung 11.3 veranschaulicht die Archivprotokollierung.

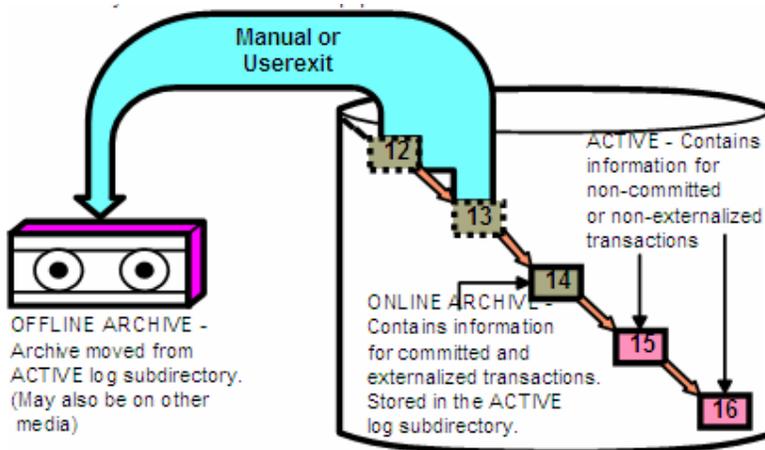


Abbildung 11.3 – Archivprotokollierung

11.4 Datenbankprotokollierung über die Steuerzentrale

Sie können die Datenbankprotokollierung über die DB2-Steuerzentrale konfigurieren indem sie einen Rechtsklick auf die in Frage kommende Datenbank ausführen und *Datenbankprotokollierung konfigurieren* auswählen. Dies wird in Abbildung 11.4 dargestellt.

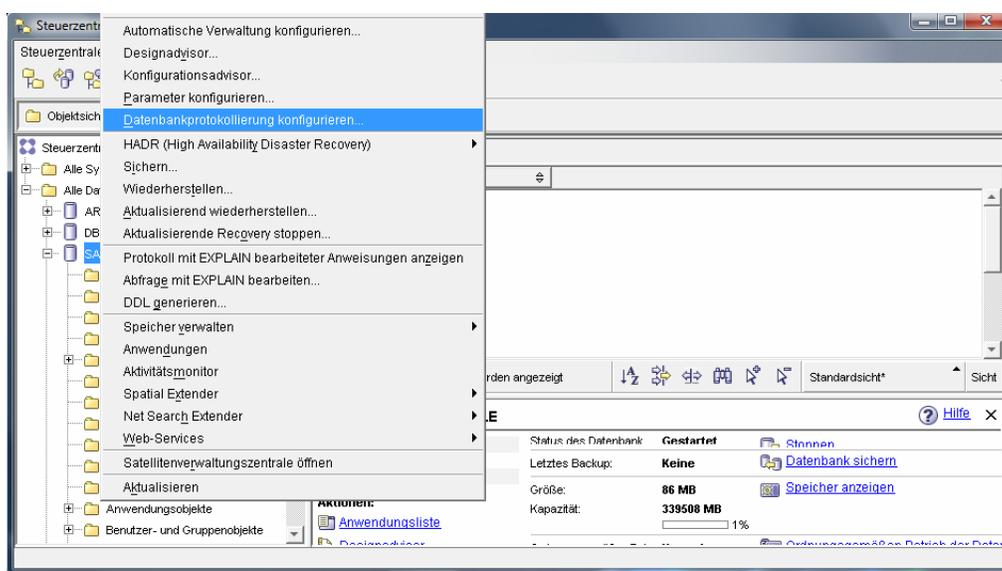


Abbildung 11.4 – Datenbankprotokollierung über die Steuerzentrale konfigurieren

Abbildung 11.5 zeigt den Assistenten Datenbankprotokollierung konfigurieren, wo Sie Umlaufprotokollierung oder Archivprotokollierung wählen können.

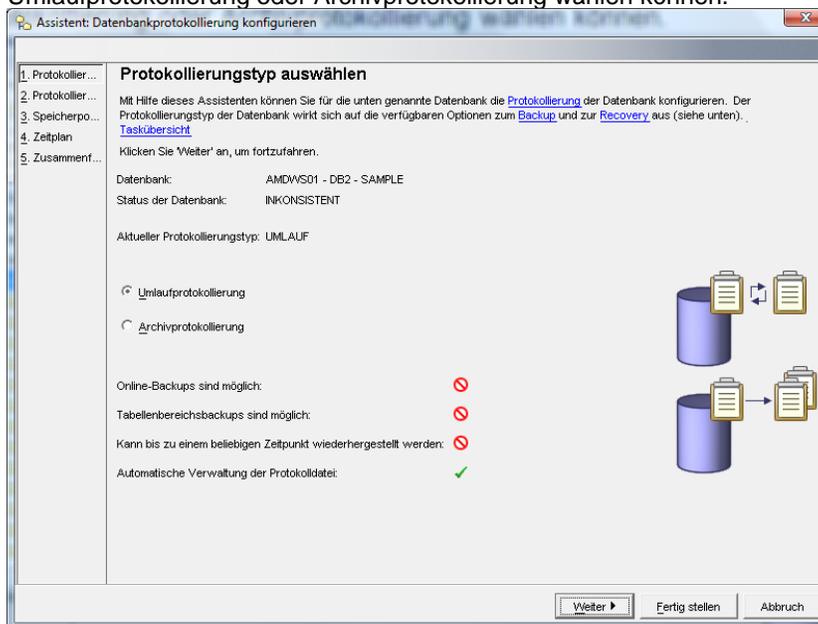


Abbildung 11.5 – Der Assistent Datenbankprotokollierung konfigurieren

11.5 Protokollparameter

Es existieren mehrere, die Protokollierung betreffende, DB CFG Parameter. Tabelle 11.1 listet die gebräuchlichsten Parameter auf.

Parameter	Beschreibung
LOGBUFSZ	Die Speichermenge, welche als Puffer für Protokollsätze verwendet wird, bevor diese auf Platte geschrieben werden
LOGFILSIZ	Die Größe jeder konfigurierten Protokolldatei in 4KB Seiten
LOGPRIMARY	Die Anzahl der zu erstellenden primären Protokolldateien in der Größe LOGFILSIZ.
LOGSECOND	Die Anzahl der sekundären Protokolldateien welche, falls erforderlich, erstellt und für die Wiederherstellung benötigt werden.
LOGPATH/ NEWLOGPATH MIRRORLOGPATH	Der Speicherort in welchem aktive Protokolldateien gespeichert werden. Um die primären Protokolle im primären Protokollspeicherpfad vor Plattenfehlern oder versehentlichem Löschen zu schützen kann angegeben werden, dass ein identischer Satz Protokolle in einem sekundären Pfad gespeichert (gespiegelt) wird.
LOGHEAD	Der Name der derzeit aktiven Protokolldatei.
USEREXIT	Aktiviert das USEREXIT Programm um Protokolldateien Offline zu kopieren.
LOGRETAIN OVERFLOWLOGPATH	Aktiviert den Archivprotokollierungsmodus. Entspricht der OVERFLOW LOG PATH Option des ROLLFORWARD Befehls; statt die OVERFLOW LOG PATH Option für jeden ROLLFORWARD Befehl zu setzen, kann dieser Parameter einmalig gesetzt werden.
BLK_LOG_DSK_FUL	Wird gesetzt um Platte voll Fehler zu vermeiden, falls DB2 eine neue Protokolldatei nicht im aktiven Log Pfad generieren kann. Stattdessen wird DB2 solange versuchen alle 5 Minuten die Logdatei zu erstellen bis dies erfolgreich ist. Nichtgeblocktes Nur-Lesen SQL kann eventuell fortgesetzt werden.
SOFTMAX MAX_LOG	Begrenzt die Kosten einer Ausfallwiederherstellung. Prozentsatz des maximal aktiven Protokollbereichs je Transaktion.
NUM_LOG_SPAN	Anzahl der aktiven Protokolldateien für eine aktive Transaktion (UOW).

Tabelle 11.1 – Protokollparameter

11.6 Datenbanksicherung

Der DB2 Backup Befehl ermöglicht Ihnen eine Schnapsschusskopie ihrer Datenbank zum Zeitpunkt der Ausführung des Befehls zu erstellen. Die einfachste Syntax, welche Sie zur Ausführung des Befehls benötigen, ist:

```
BACKUP DATABASE <dbname> [ TO <path> ]
```

Die meisten Befehle und Dienstprogramme können Online oder Offline ausgeführt werden. Online impliziert das, während Sie den Befehl ausführen, andere Benutzer möglicherweise mit der Datenbank verbunden sind und Operationen durchführen. Offline bedeutet, dass keine Benutzer mit der Datenbank verbunden sind während Sie den Befehl ausführen. Um eine Online Operation zu ermöglichen fügen Sie das Schlüsselwort ONLINE zur Befehlssyntax hinzu da sonst standardmäßig angenommen wird dass sie den Befehl Offline ausführen.

Falls Sie zum Beispiel die Datenbank *sample* in den Pfad C:\BACKUPS sichern wollen, können sie den folgenden Befehl im DB2 Befehlsfenster/der Linux Shell absetzen:

```
db2 BACKUP DB sample TO C:\BACKUPS
```

Bitte beachten Sie dass das Verzeichnis C:\BACKUPS vor dem Ausführen des Befehls vorhanden sein muss. Stellen Sie außerdem sicher, dass keine Verbindungen zu der Datenbank bestehen wenn sie den Befehl absetzen, da sie sonst eine Fehlermeldung erhalten, welche darauf hinweist, das eine Offline Sicherung bei bestehenden Verbindungen nicht möglich ist.

Um herauszufinden ob Verbindungen zu Datenbank in einer Instanz bestehen, setzen sie folgenden Befehl im DB2 Befehlsfenster/der Linux Shell ab:

```
db2 list applications
```

Um alle Verbindungen mit allen Datenbanken in einer Instanz zwingend zu beenden, führen Sie diesen Befehl über das DB2 Befehlsfenster oder die Linux Shell aus:

```
db2 force applications all
```

Es empfiehlt sich nicht den letzten Befehl in einer Produktionsumgebung mit vielen Benutzern verwenden, da Sie sonst viele Anrufe verärgelter Mitarbeiter erhalten! Beachten Sie außerdem, dass der letzte Befehl asynchron ausgeführt wird. Dies bedeutet, dass falls Sie versuchen direkt im Anschluss den Backup Befehl auszuführen, es immer noch nicht funktioniert. Warten Sie ein paar Sekunden und wiederholen den Backup Befehl falls sie beim ersten Mal eine Fehlermeldung erhielten.

Nach erfolgreicher Ausführung des Backup Befehls wurde eine neue Datei, die Sicherung (Backup) der Datenbank enthaltend, erstellt. Der Name der Datei folgt den in Abbildung 11.6 gezeigten Konventionen.

Linux/UNIX/Windows

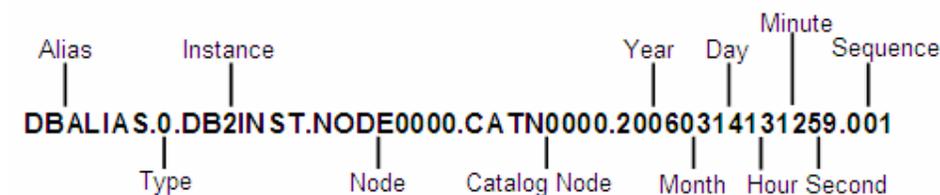


Abbildung 11.6 – Backupimage Namenskonvention

Der Type "0" bedeutet, dass es sich um ein vollständiges Backup handelt. So bedeutet der Type "3" zum Beispiel dass es sich nur um ein Backup des Tabellenbereichs handelt. Der Knoten (NODE) ist festgelegt mit NODE0000 bei nichtpartitionierten Datenbanken welches, mit Ausnahme der DB2 Enterprise Edition mit der DPF Funktionalität, bei allen DB2 Editionen der Fall ist. Der Katalog Knoten ist mit CATN0000 festgelegt. Ausführliche Details finden Sie in den DB2 Handbüchern.

Wenn mehrere Sicherungen erstellt und im gleichen Pfad gespeichert werden, unterscheiden sich die Sicherungen durch den Zeitstempel am Ende des Dateinamens. Wie wir im nächsten Kapitel sehen werden, kann der RESTORE Befehl die Zeitstempel nutzen um eine bestimmte Sicherung wiederherzustellen.

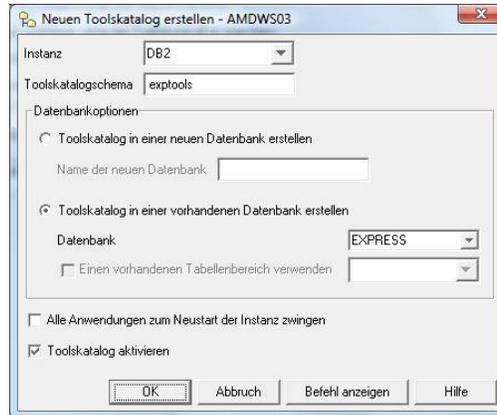
Übung 10 – Planen einer Sicherung

Ziel

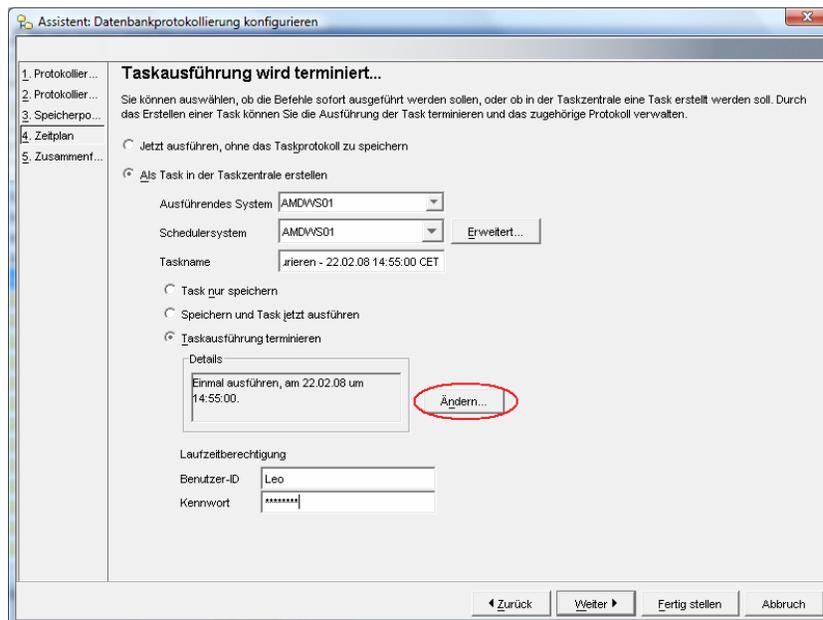
Obwohl DB2 in der Lage ist verschiedene Wartungsarbeiten zu automatisieren, möchten Sie diese anpassen falls bestimmte Aktivitäten auftreten. In dieser Übung werden sie eine angepasste nächtliche Sicherungsplanung für die *EXPRESS* Datenbank erstellen.

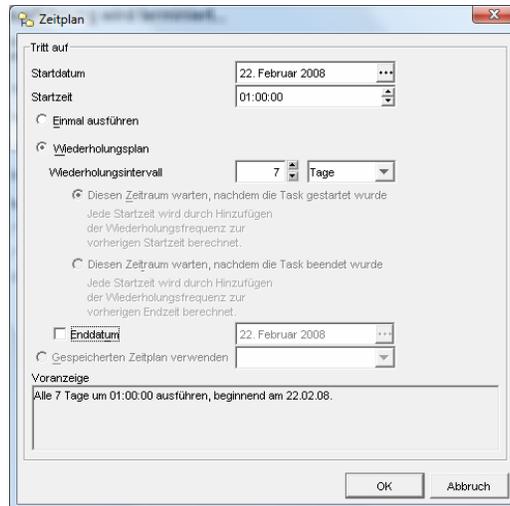
Vorgehensweise

1. Navigieren Sie in der Objektsicht der Steuerzentrale nach Alle Datenbanken. Rechtsklicken sie auf die *EXPRESS* Datenbank und wählen den *Sichern* Menüpunkt. Damit wird der *Sicherungsassistent* gestartet.
2. Die Seite *Einführung* des Assistenten summiert den aktuellen Status der Datenbank, einschließlich des Zeitpunkts der letzten Sicherung und den Protokollierungstyp. Klicken Sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.
3. Auf der *Image* Seite des Assistenten wählen Sie das Ziel des Sicherungimages. Sie werden normalerweise ein von Datenbankspeicherpfad abweichendes physikalisches Laufwerk wählen. Fürs Erste erstellen Sie einen neuen Ordner namens `C:\db2backup` im Dateisystem und wählen diesen Ordner als Sicherungsziel. Im Assistenten wählen Sie *Dateisystem* aus dem *Datenträger* Kombinationsfeld. Klicken Sie auf die *Hinzufügen* Schaltfläche und wählen den soeben erstellten Ordner. Klicken Sie anschließend auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.
4. Sie können die *Optionen* und *Leistung* Seiten erforschen, doch sind die Standardwerte meistens ausreichend, da DB2 die Sicherung automatisch optimal durchführt. Wechseln Sie zur Seite *Zeitplan* wenn ihre Entdeckungsreise beendet ist.
5. Auf der *Zeitplan* Seite können Sie, falls die Zeitplanung noch nicht aktiviert wurde, diese jetzt aktivieren. Wählen Sie das System auf welchem der Tools Katalog erstellt werden soll und erstellen einen neuen Tools Katalog. Geben Sie ein Schema für den Tools Katalog an und wählen Sie die existierende *EXPRESS* Datenbank, um den Katalog darin zu erstellen. Der Tools Katalog enthält Metadaten über alle geplanten Aufgaben. Nachdem der Tools Katalog erstellt wurde, klicken Sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.



6. Auf der *Zeitplan* Seite, wählen Sie dass das Task in der Taskzentrale gespeichert und die Taskausführung terminiert werden soll. Planen Sie eine tägliche, um 1 Uhr startende, Sicherung. Klicken Sie auf die *Weiter* Schaltfläche um zur nächsten Seite zu gelangen.





7. Auf der *Zusammenfassung* Seite können Sie die zu erstellenden geplanten Tasks prüfen. Nachdem Sie die Änderungen geprüft haben klicken sie auf die *Fertig stellen* Schaltfläche um das Task zu erstellen.
8. Starten Sie die Taskzentrale um das neu erstellte Sicherungstask anzusehen oder zu ändern.

11.7 Datenbankwiederherstellung

Eine Datenbankwiederherstellung impliziert die Wiederherstellung Ihrer Datenbank aus einer Sicherung, eventuell mit Protokolldateien. Wenn Sie die Datenbank aus einer Sicherung wiederherstellen, wird die Datenbank so wiederhergestellt wie Sie zum Zeitpunkt der Sicherung existierte.

Falls Archivprotokollierung vor der Sicherung aktiviert wurde, können nicht nur Sicherungen, sondern auch Protokolldateien, zur Wiederherstellung verwendet werden. Wie wir im nächsten Abschnitt sehen werden, erlaubt Ihnen eine aktualisierende Wiederherstellung (roll-forward recovery) eine Datenbank aus einer Sicherung wiederherzustellen, und anschließend die Protokolldateien einzuspielen um die Datenbank entweder bis zum Ende der Protokolldateien (End of logs) oder bis zu einem bestimmten Zeitpunkt (Point in time) wiederherzustellen.

Bitte beachten Sie, dass die Begriffe "Wiederherstellung" bzw. "recovery" in diesem Abschnitt oft verwendet werden, der zur Wiederherstellung verwendete Befehl jedoch 'RESTORE' lautet.

11.7.1 Wiederherstellungsarten

Es gibt drei Arten der Wiederherstellung:

- **Absturz oder Neustart Wiederherstellung**
Stellen Sie sich vor, Sie arbeiten auf einem Desktop Computer und lassen wichtige Transaktionen gegen die DB2 Datenbank laufen. Plötzlich kommt es zu einem Stromausfall oder jemand zieht versehentlich den Stromstecker: Was wird mit der Datenbank passieren?

Das nächste Mal wenn Sie den Computer hochfahren und DB2 starten, wird die Absturz Wiederherstellung (Crash Recovery) automatisch gestartet. Während der Absturz Wiederherstellung führt DB2 automatisch den Befehl RESTART DATABASE aus und macht, basierend auf die aktiven Protokolle, Transaktionen rückgängig bzw. stellt Transaktionen wieder her. Nach Abschluss des Befehls können Sie sicher sein, dass sich Ihre Datenbank in einem konsistenten Zustand befindet, das bedeutet, was auch immer committed wurde wird gespeichert und was auch immer nicht committed wurde wird mit roll back zurückgesetzt.
- **Versionswiederherstellung aus Sicherung**
Diese Art der Wiederherstellung impliziert, dass Sie ausschließlich von einer Sicherung wiederherstellen. Ihre Datenbank wird deshalb in den Zustand versetzt, in welchem sich diese bei der Erstellung der Sicherung befand. Alle nach der Sicherung erfolgten Transaktionen gehen verloren.
- **Aktualisierende Wiederherstellung**
Mit diesem Typ der Wiederherstellung können Sie nicht nur ein RESTORE von einer Sicherung durchführen, Sie können auch den ROLLFORWARD Befehl ausführen um

12

Kapitel 12 – Wartungsarbeiten

Dieses Kapitel behandelt einige der Arbeiten, welche erforderlich sind um Ihre Datenbank in einem gut gewarteten Zustand zu behalten. Im Allgemeinen automatisiert DB2 die meisten dieser Arbeiten. DB2 Express-C Edition, wie alle aktuellen DB2 Editionen, enthält diese Automatisierungsmöglichkeiten. Diese Selbstverwaltungsfähigkeit ist von großem Nutzen bei kleinen und mittleren Unternehmen, welche keinen Vollzeit DBA anstellen können um den Datenserver zu steuern. Andererseits, falls ein DBA angestellt ist, wird er oder sie mehr freie Zeit haben fortgeschrittenen Aktivitäten nachzugehen, was gewinnbringend für das Unternehmen sein kann.

Hinweis:

Für weitere Informationen bezüglich Wartungsarbeiten können Sie sich folgendes Video ansehen:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4302>

12.1 REORG, RUNSTATS, REBIND

Es gibt, wie aus Abbildung 12.1 ersichtlich, drei hauptsächliche Wartungsarbeiten in DB2: REORG, RUNSTATS und REBIND.

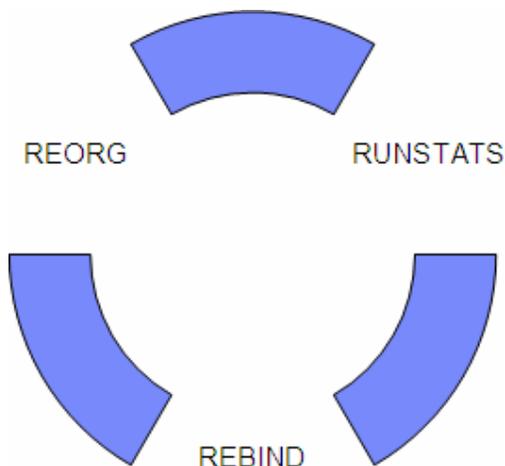


Abbildung 12.1 – Wartungsarbeiten: REORG, RUNSTATS, REBIND

Abbildung 12.1 zeigt, dass die Wartungsarbeiten im Umlaufverfahren durchgeführt werden. Wenn ein REORG durchgeführt wird, ist es empfohlen ebenfalls RUNSTATS, gefolgt von REBIND auszuführen. Nach einiger Zeit werden die Tabellen durch UPDATE, DELETE und INSERT Vorgänge geändert. Zu diesem Zeitpunkt startet der Zyklus, beginnend mit REORG, neu.

12.1.1 Der REORG Befehl

Nach einiger Zeit, wenn Sie INSERT, UPDATE und DELETE Vorgänge in ihrer Datenbank vornehmen werden die Daten mehr und mehr über die Datenbankseiten fragmentiert. Der REORG Befehl gibt verschwendeten Platz frei und reorganisiert die Daten um den Datenzugriff effektiver zu machen. Tabellen, welche häufig geändert werden, profitieren am meisten von REORG. Sie können REORG mit Indizes und Tabellen verwenden und REORG kann sowohl Online als auch Offline ausgeführt werden.

Offline REORG ist schneller und effizienter, erlaubt jedoch keinen Zugriff auf die Tabelle, wohingegen ein Online REORG Tabellenzugriff zulässt, kann allerdings viele Systemressourcen verbrauchen und ist daher am Besten für kleine Tabellen verwendbar.

Syntax:

```
REORG TABLE <tablename>
```

Beispiel:

```
REORG TABLE employee
```

Der REORGCHK Befehl kann, um festzustellen ob ein Index repariert werden muss, vor einem REORG benutzt werden.

12.1.2 Der RUNSTATS Befehl

Der DB2 Optimierer ist "das Gehirn" von DB2. Es findet den günstigsten Zugriffspfad um Daten zu lokalisieren und abzurufen. Der Optimierer beachtet die Systembelastung und verwendet Statistiken der Datenbankobjekte, welche in Systemtabellen gespeichert werden, um die Datenbankleistung zu optimieren. So beinhalten Katalogtabellen Statistiken über die Anzahl in einer Tabelle vorhandenen Spalten, die Anzahl vorhandener Zeilen, Anzahl und Typ der verfügbaren Indizes für jede Tabelle und so weiter.

Statistikinformationen werden nicht dynamisch aktualisiert. Dies ist das Grundkonzept, da Sie nicht möchten das DB2 die Statistiken fortlaufend für jede durchgeführte Operation aktualisiert, da dies die gesamte Datenbankleistung negativ beeinflusst. Stattdessen stellt DB2 den RUNSTATS Befehl zur Verfügung um die Statistiken zu aktualisieren. Es ist essentiell die Datenbankstatistiken auf einen aktuellen Stand zu halten. Der DB2 Optimierer kann, falls er denkt dass eine Tabelle 1 Zeile gegenüber 1 Million Zeilen hat, den Zugriffspfad radikal ändern. Wenn sich die Datenbankstatistiken auf einem aktuellen Stand befinden kann DB2 einen besseren Zugriffspfad wählen. Wie oft die Statistiken gesammelt werden sollte auf Grundlage der Anzahl Datenänderungen in einer Tabelle ermittelt werden.

Syntax:

```
RUNSTATS ON TABLE <schema.tablename>
```

Beispiel:

```
RUNSTATS ON TABLE myschema.employee
```

12.1.3 BIND / REBIND

Nachdem der RUNSTATS Befehl erfolgreich ausgeführt wurde, werden nicht alle Abfragen den letzten Stand der Statistiken verwenden. Statische SQL Zugriffspläne werden ermittelt wenn Sie den BIND Befehl absetzen, mit der Folge, dass die derzeit verwendeten Statistiken nicht identisch mit den aktuellen sind. Abbildung 12.2 hilft dies zu veranschaulichen.

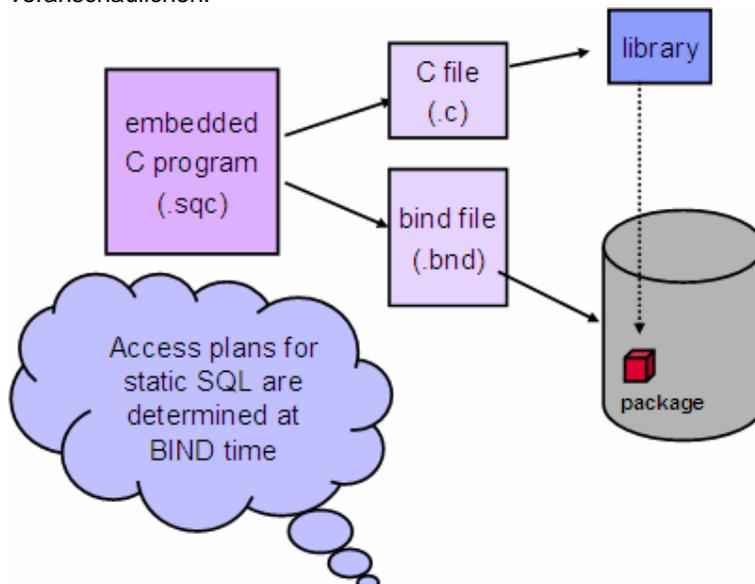


Abbildung 12.2 – Der Statische SQL Bindevorgang

In Abbildung 12.2 wird ein eingebettetes C Programm (gespeichert als eine Datei mit der Dateierweiterung "sqc") vorkompiliert. Nach der Vorkompilierung werden zwei Dateien generiert: eine den C Code beinhaltende ".c" Datei, bei welcher alle SQL Anweisungen auskommentiert wurden, und eine ".bnd" Datei, in welcher sich alle SQL Anweisungen befinden. Die C Datei mit der ".c" Erweiterung wird mit einem gebräuchlichen C Compiler kompiliert, wodurch eine Bibliothek ("library"), wie oben rechts in der Abbildung gezeigt, erstellt wird. Die ".bnd" Datei wird ähnlich gebunden und generiert ein in der Datenbank gespeichertes Paket. Das Binden ist gleichbedeutend mit dem kompilieren von SQL Anweisungen, bei welchen der beste Zugriffsplan basierend auf die aktuell verfügbaren Statistiken ermittelt wird, und anschließend in einem Paket gespeichert.

Doch was passiert, wenn 1 Million Zeilen in eine Tabelle eingefügt werden, die von oben genanntem C Programm referenziert wird? Falls nach dem Einfügen RUNSTATS ausgeführt wird, werden die Statistiken aktualisiert, das Paket jedoch nicht automatisch aktualisiert um den Zugriffsplan basierend auf die aktuellen Statistiken zu ändern. Der

db2rbind Befehl kann verwendet werden um alle vorhandenen Pakete unter Berücksichtigung der aktuellen Statistiken neu zu binden.

Syntax:

```
db2rbind database_alias -l <logfile>
```

Beispiel:

Um alle Pakete der *sample* Datenbank neu zu binden und Ausgabeprotokoll in die Datei mylog.txt zu schreiben, setzen Sie folgenden Befehl ab:

```
db2rbind sample -l mylog.txt
```

12.1.4 Wartungsarbeiten über die Steuerzentrale ausführen

Über die DB2-Steuerzentrale können Sie die Dienstprogramme REORG und RUNSTATS ausführen. Abbildung 12.3 zeigt Ihnen wie.

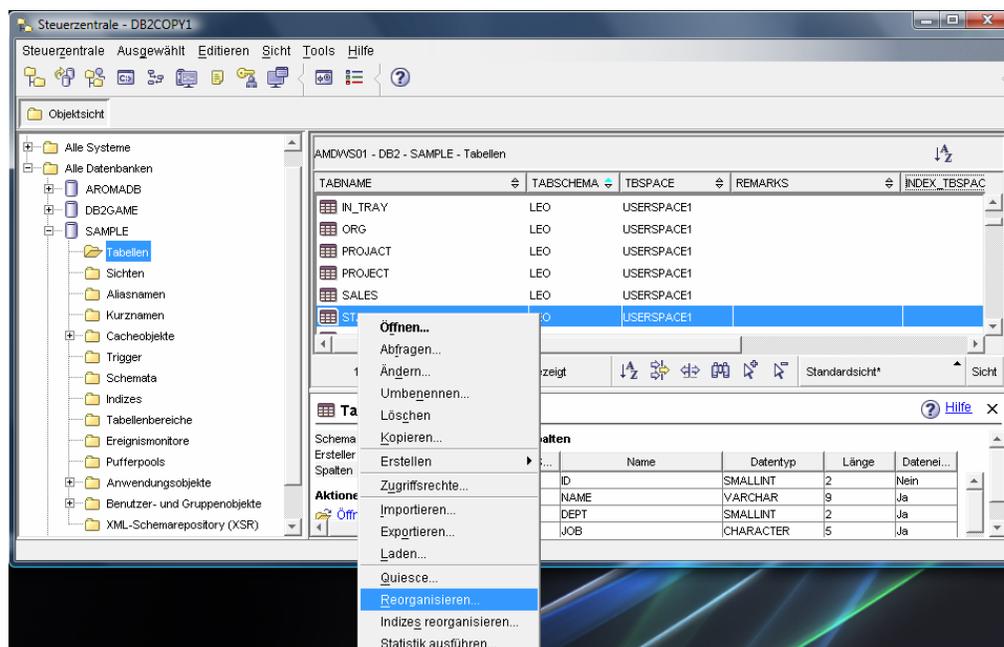


Abbildung 12.3 – Reorganisieren und Statistik über die Steuerzentrale ausführen

Sie wählen die Tabelle auf welche die Operationen durchgeführt werden sollen, klicken rechts auf diese und wählen *Reorganisieren* (für REORG) oder *Statistik ausführen* (für RUNSTATS).

Objektdetails einer Datenbank

Wenn Sie eine Datenbank auswählen, werden im rechten unteren Teilfenster der DB2-Steuerzentrale unten rechts die Objektdetails mit grundlegenden Datenbankinformationen, wie die Größe, wann sie zuletzt gesichert wurde, ob automatische Verwaltung verwendet

wird usw., angezeigt. Diese Sicht ermöglicht es Ihnen notwendige Wartungsaufgaben zu identifizieren. Abbildung 12.4 zeigt diese Information.

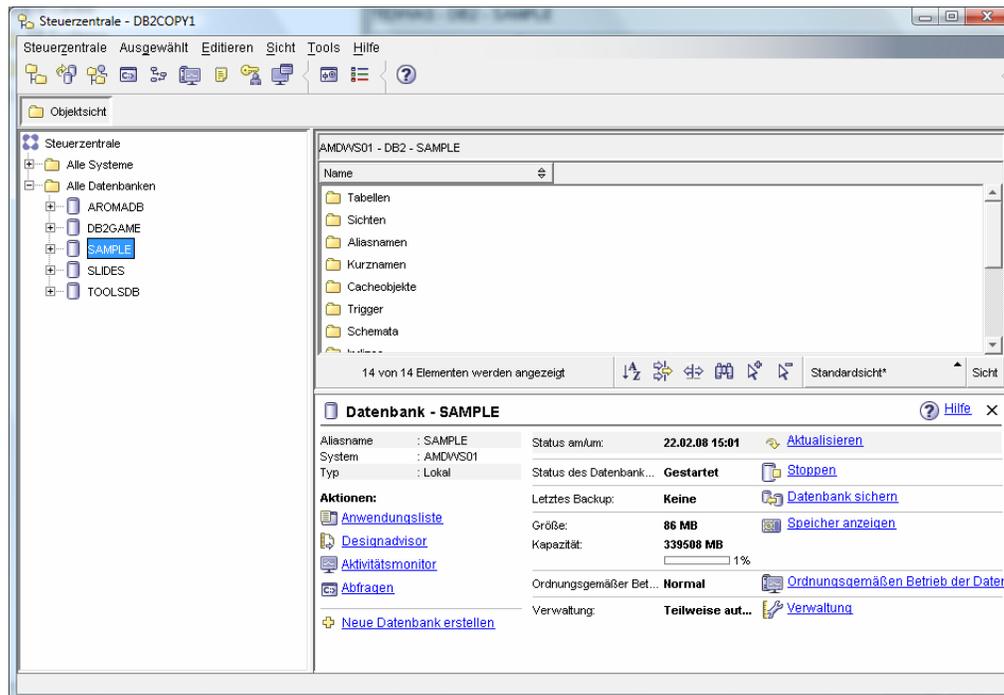


Abbildung 12.4 – Ansicht der Objektdetails einer Datenbank in der Steuerzentrale

12.2 Verwaltungsoptionen

Es gibt folgende Möglichkeiten, Verwaltungsaufgaben auszuführen:

1. Manuelle Verwaltung
Sie führen Verwaltungsarbeiten manuell bei Bedarf durch
2. Erstellen von Skripten um Verwaltungsaufgaben durchzuführen
Sie können Skripte mit Verwaltungsbefehlen erstellen und diese als Tasks zur regelmäßigen Ausführung vorsehen.
3. Automatische Verwaltung
DB2 führt die Verwaltung für Sie durch (REORG, RUNSTATS, BACKUP).

In diesem Abschnitt konzentrieren wir uns auf die automatische Verwaltung.

Die automatische Verwaltung besteht aus dem Folgenden:

- Der Benutzer definiert ein Verwaltungsfenster worin Tasks mit minimaler Störung ausgeführt werden. Sollte zum Beispiel das System Sonntags zwischen 1:00 Uhr

und 2:00 Uhr die geringste Aktivität aufweisen, würde sich dieser Zeitraum als Fenster für Verwaltungsaufgaben empfehlen.

- Es gibt zwei Verwaltungsfenster: eins für Online Operationen und ein anderes für Offline Operationen.
- DB2 wird Verwaltungsaufgaben nur bei Bedarf im Zeitfenster automatisch ausführen

Über die DB2-Steuerzentrale können Sie den Assistenten Automatische Verwaltung konfigurieren, wie in Abbildung 12.5 gezeigt, aufrufen.

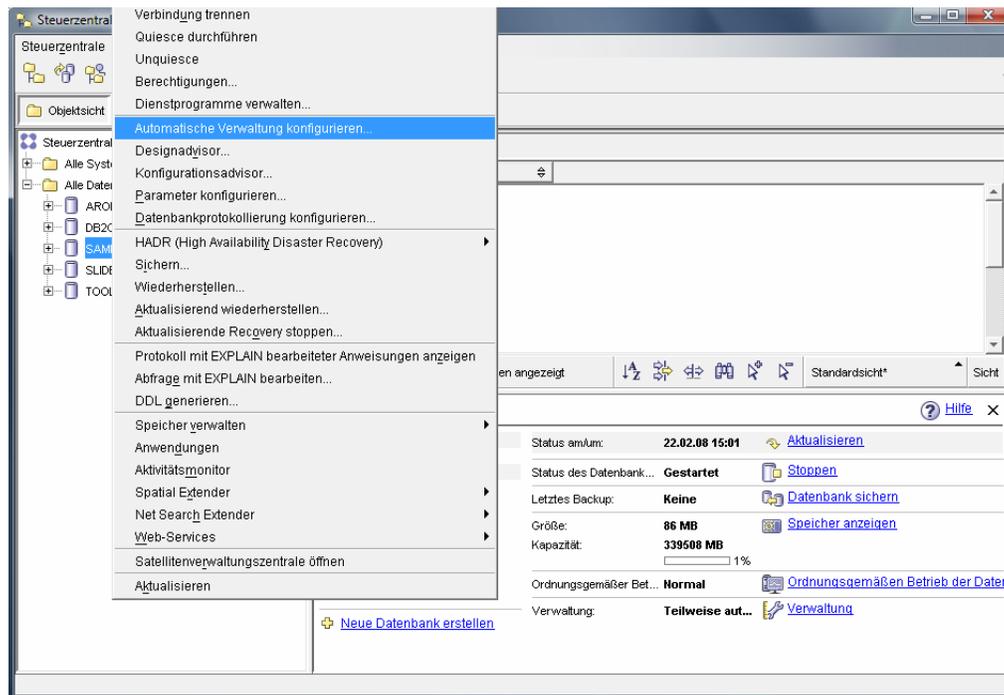


Abbildung 12.5 – Starten des Assistenten Automatische Verwaltung konfigurieren

Abbildung 12.6 zeigt den Assistenten Automatische Verwaltung konfigurieren.

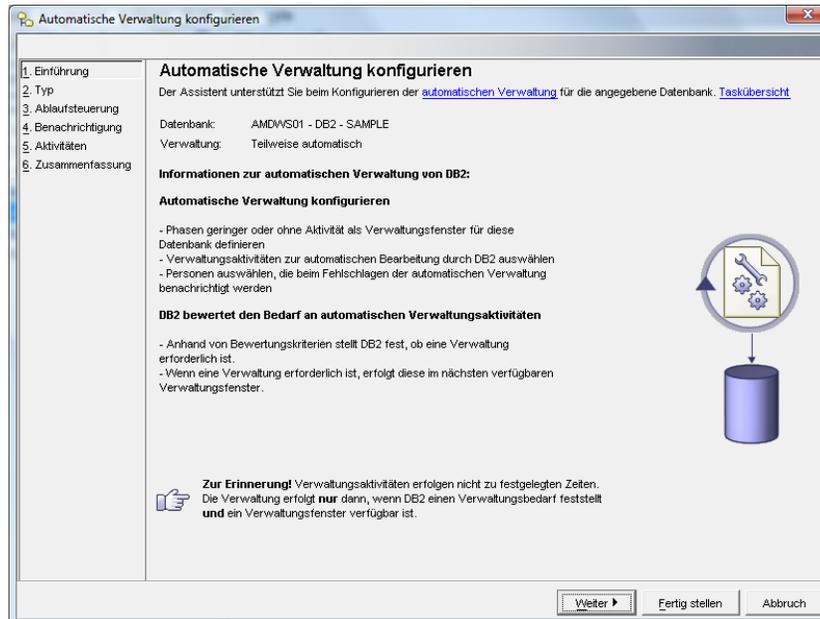


Abbildung 12.6 – Der Assistent Automatische Verwaltung konfigurieren

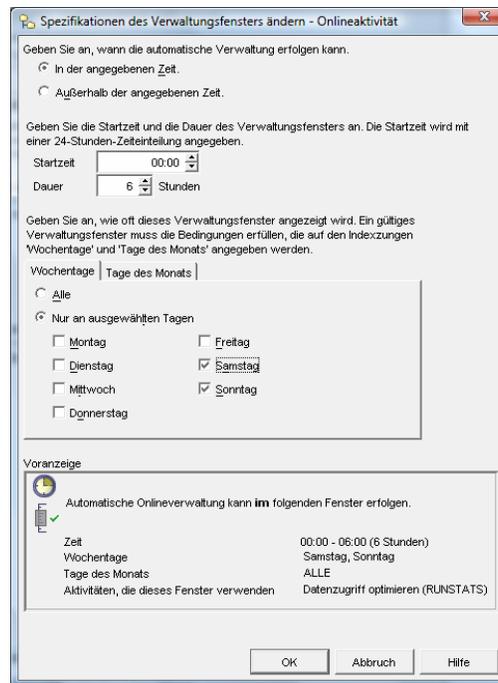
Übung 11 – Konfigurieren der automatischen Verwaltung

Ziel

In dieser Übung, mit einigen wenigen leichten Schritten, werden Sie die automatische Verwaltung der DB2 SAMPLE Datenbank konfigurieren.

Vorgehensweise

1. In der Objektsicht der DB2-Steuerzentrale klicken Sie rechts auf die *SAMPLE* Datenbank und wählen den *Automatische Verwaltung konfigurieren* Menüpunkt. Dies startet den *Automatische Verwaltung konfigurieren* Assistenten.
2. Die Seite *Einführung* des Assistenten zeigt die aktuellen Einstellungen der automatischen Verwaltung. Falls Sie eine Datenbank mit der automatischen Verwaltungsoption erstellt haben, ist die automatische Verwaltung bereits konfiguriert. Sie können diesen Assistenten verwenden um die automatischen Verwaltungsoptionen neu zu konfigurieren. Klicken sie auf die *Weiter* Schaltfläche um zur nächsten Seite des Assistenten zu gelangen.
3. Die *Typ* Seite des Assistenten erlaubt es Ihnen die automatische Verwaltung zu deaktivieren oder die Einstellungen der automatischen Verwaltung zu ändern. Wählen Sie die Option zur Änderung der aktuellen Automatisierungseinstellungen. Klicken Sie auf *Weiter*.
4. Die *Ablaufsteuerung* Seite des Assistenten verlangt von Ihnen die Angabe der Verwaltungsfenster. Konfigurieren Sie das Zeitfenster der *Offlineaktivität* so, dass es jeden Samstag zwischen Mitternacht und 6:00 Uhr liegt, wie unten gezeigt. Klicken Sie auf die *Ändern* Schaltfläche neben der Voransicht des Offlineverwaltungsfensters und wählen die gewünschten Zeiten. Nachdem die erforderlichen Angaben gemacht wurden klicken Sie auf die *OK* Schaltfläche um zum Assistenten zurückzukehren. Belassen Sie das Onlineverwaltungsfenster wie es ist (Onlineverwaltung kann jederzeit ablaufen). Klicken Sie auf die *Weiter* Schaltfläche.



5. Auf der *Benachrichtigung* Seite des Assistenten können Sie, für den Fall dass die automatische Verwaltung fehlschlägt, einen Kontakt angeben. Übergehen Sie diesen Schritt fürs Erste. Klicken Sie auf die *Weiter* Schaltfläche.
6. Auf der *Aktivitäten* Seite des Assistenten können Sie wählen ob bestimmte Aktivitäten aktiviert werden sollen oder nicht und ob sie über diese Aktivitäten informiert werden wollen. In diesem Beispiel stellen Sie bitte sicher, dass alle *Automatisieren* Kontrollkästchen mit einem Häkchen versehen sind und die *Benachrichtigen* Kontrollkästchen nicht mit einem Häkchen versehen sind. Klicken Sie auf die *Weiter* Schaltfläche.
7. Bevor Sie zur nächsten Seite gehen sollten sie die Speicherposition der Sicherungen ihrer Datenbank festlegen. Idealerweise werden Sie, um Probleme bei einem Ausfall der Festplatte vorzubeugen, die Sicherung auf einem anderen physikalischen Laufwerk speichern. Auf der *Aktivitäten* Seite wählen Sie die *Datenbank sichern* Option und klicken anschließend auf die *Einstellungen konfigurieren* Schaltfläche.
8. Auf der Registerkarte *Backup-Bedingungen* des Einstellungen konfigurieren Dialogs wählen Sie die Option *Wiederherstellbarkeit der*

Datenbank mit der Option Leistung ausgleichen. Auf der *Backup-Position* Registerkarte wählen Sie die vorhandene Backup-Position und klicken auf die *Ändern* Schaltfläche. Geben Sie eine abweichende Position zur Ausführung der Sicherung an (stellen sie sicher, das auf dem Laufwerk genügend freier Speicherplatz vorhanden ist). Auf der Registerkarte *Backup-Modus* stellen Sie sicher, dass *Offline-Backup* ausgewählt ist. Klicken Sie auf die *OK* Schaltfläche um den *Backup-Bedingungen* Dialog zu schließen. Klicken Sie auf die *Weiter* Schaltfläche.

9. Die *Zusammenfassung* Seite des *Automatische Verwaltung konfigurieren* Assistenten enthält eine Zusammenfassung der von Ihnen gewählten Optionen. Klicken Sie auf die *Fertigstellen* Schaltfläche um die vorgenommenen Änderungen zu akzeptieren um implementieren.

13

Kapitel 13 – Gleichzeitiger Zugriff und Datensatzsperrern

Dieses Kapitel erläutert wie sie mehreren Benutzern gleichzeitigen Zugriff auf die Datenbank erlauben können, ohne dass diese in Zugriffskonflikte geraten und dabei deren Operationen konsistent halten. Wir behandeln die Konzepte von Transaktionen, gleichzeitigem Zugriff (concurrency) und Datensatzsperrern (locking).

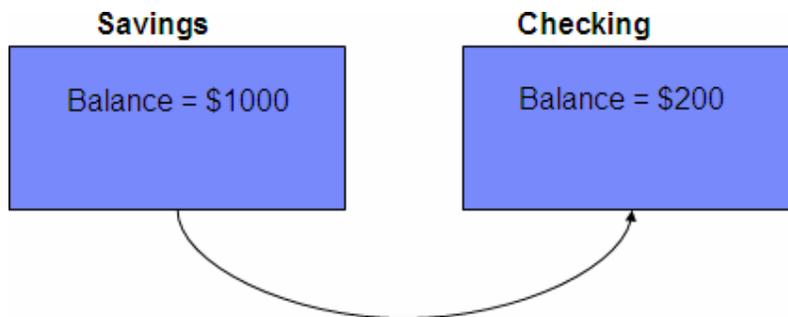
Hinweis:

Für weitere Informationen über Transaktionen, gleichzeitigem Zugriff und Datensatzsperrern können Sie sich dieses Video ansehen:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4322>

13.1 Transaktionen

Eine Transaktion oder Arbeitseinheit besteht aus einer oder mehreren SQL Anweisungen welche, wenn ausgeführt, als einzelne Einheit betrachtet werden sollten; dies bedeutet, dass falls eine Anweisung fehlschlägt, und dass alle bis zu dem Zeitpunkt des Fehlers verarbeiteten Anweisungen rückgängig gemacht (rolled back) werden. Eine Transaktion endet mit einer COMMIT Anweisung, welche auch den Start einer neuen Transaktion kennzeichnet. Abbildung 13.1 zeigt ein Beispiel einer Transaktion.



Transfer \$100 from Savings to Checking:

- Debit \$100 from Savings account
- Credit \$100 to Checking account

Abbildung 13.1 – Beispiel einer Transaktion

Sie möchten zum Beispiel, wie in Abbildung 13.1 gezeigt, 100 Dollar von ihrem Sparkonto auf ihr Bankkonto überweisen. Die folgende Abfolge von Ereignissen ist möglicherweise erforderlich um die Aufgabe zu erledigen:

- Belasten des Sparkontos mit 100\$
- Gutschrift von 100\$ auf dem Bankkonto

Wenn die obige Abfolge von Ereignissen nicht als eine einzelne Arbeitseinheit behandelt wird, eine Transaktion, können Sie sich sicher vorstellen was passiert, wenn es nach der Belastung des Sparkontos mit 100\$ zu einem Stromausfall kommt bevor dem Bankkonto 100\$ gutgeschrieben wurden. Sie werden 100\$ verlieren!

13.2 Gleichzeitiger Zugriff (Concurrency)

Gleichzeitiger Zugriff setzt voraus dass mehrere Benutzer zur gleichen Zeit mit denselben Datenbankobjekten arbeiten können. DB2 wurde als Multi-User Datenbank ausgelegt. Der Datenzugriff muss genau und transparent koordiniert werden und verwendet dazu einen Mechanismus welcher die Datenintegrität und Konsistenz sicherstellt. Betrachten Sie die Abbildung 13.2 als ein Beispiel.

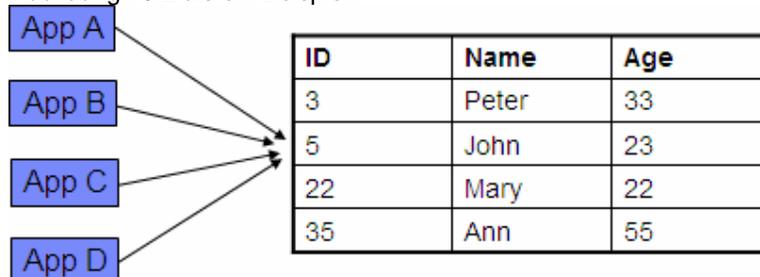


Abbildung 13.2 – Ein Beispiel gleichzeitigen Datenzugriffs und die Notwendigkeit der Kontrolle des gleichzeitigen Datenzugriffs

In Abbildung 13.2 sehen Sie vier Anwendungen: App A, App B, App C und App D, welche versuchen auf Daten derselben Zeile (Zeile 2) einer Tabelle zuzugreifen. Wenn der gleichzeitige Zugriff nicht kontrolliert wird, können alle Anwendungen Operationen auf dieselbe Zeile ausführen. Wenn wir davon ausgehen, dass alle Anwendungen die Age Spalte in Zeile 2 mit verschiedenen Werten aktualisieren, wird die Anwendung welche das letzte Update durchführt wahrscheinlich der "Gewinner" in dieser Situation sein. Dieses Beispiel soll deutlich machen dass irgendeine Art der Kontrolle des gleichzeitigen Zugriffs erforderlich ist um konsistente Ergebnisse zu garantieren. Die Kontrolle des gleichzeitigen Zugriffs wird über Datensatzsperrern realisiert.

Die Konzepte des Datensatzsperrerns und gleichzeitigen Zugriffs gehen Hand-in-Hand. Datensatzsperrern hindern andere Anwendungen solange temporär daran Operationen auszuführen, bis eine andere Operation beendet wurde. Je mehr Datensatzsperrern in einem System vorhanden sind, je weniger gleichzeitiger Zugriff ist möglich. Auf der anderen Seite ist bei weniger Datensatzsperrern in einem System mehr gleichzeitiger Zugriff möglich.

Datensatzsperrern werden bei Bedarf automatisch erlangt, um eine Transaktion zu unterstützen und nach Beendigung der Transaktion wieder freigegeben (durch Verwendung des COMMIT oder ROLLBACK Befehls). Datensatzsperrern können auf Tabellen oder Zeilen gesetzt werden. Es wird zwischen zwei grundlegenden Arten der Datensatzsperrern unterschieden:

- Freigabesperrern (S Sperrern) – werden gesetzt wenn eine Anwendung lesen möchte und andere Anwendungen daran hindern will die gleiche Zeile zu aktualisieren
- Exklusivsperrern (X Sperrern) – werden gesetzt wenn eine Anwendung eine Zeile aktualisiert, Daten einfügt oder eine Zeile löscht

Betrachten Sie jetzt die Abbildung 13.3, welche ähnlich der Abbildung 13.2 ist, jedoch jetzt eine Datensatzsperrern zeigt.

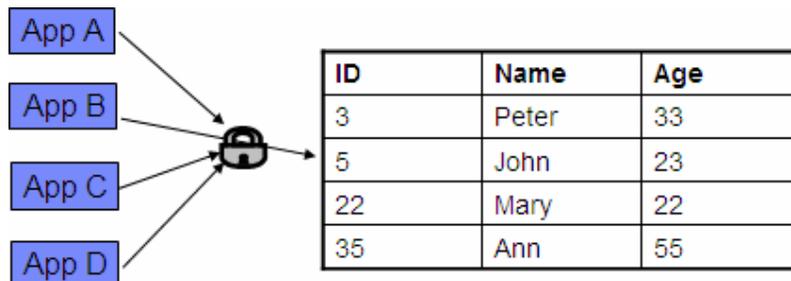


Abbildung 13.3 – Ein Beispiel gleichzeitigen Zugriffs und die Notwendigkeit von Datensatzsperrern

Wie sie am Beispiel der Abbildung 13.2 sehen geschieht folgendes: Falls App B die erste Anwendung ist welche auf Zeile 2 zugreift und ein UPDATE durchführt hält App B eine Exklusivsperrern (X lock) auf die Zeile. Wenn App A, App C und App D versuchen auf die gleiche Zeile zuzugreifen werden sie diese auf Grund der Exklusivsperrern (X lock) nicht aktualisieren (UPDATE) können. Diese Kontrolle stellt Konsistenz und Integrität der Daten sicher.

13.3 Probleme ohne Kontrolle des gleichzeitigen Zugriffs

Ohne eine irgendwie geartete Form der Kontrolle des gleichzeitigen Zugriffs könnten die folgenden Probleme auftreten

- ▶ Verlorenes Update (Lost update)
- ▶ Uncommitted read (Eine Transaktion liest einen nicht-festgelegten Zustand oder eine ohne mit Commit abgeschlossene Transaktion)
- ▶ Nichtwiederholbares Lesen (Non-repeatable read)
- ▶ Phantom read (Suchkriterien treffen während einer Transaktion auf unterschiedliche Datensätze zu)

13.3.1 Lost Update

Das lost update Problem ist ähnlich dem früher in diesem Abschnitt erläuterten Problem, wo die Anwendung, welche als letzte eine Aktualisierung (Update) durchführt, der "Gewinner" ist.

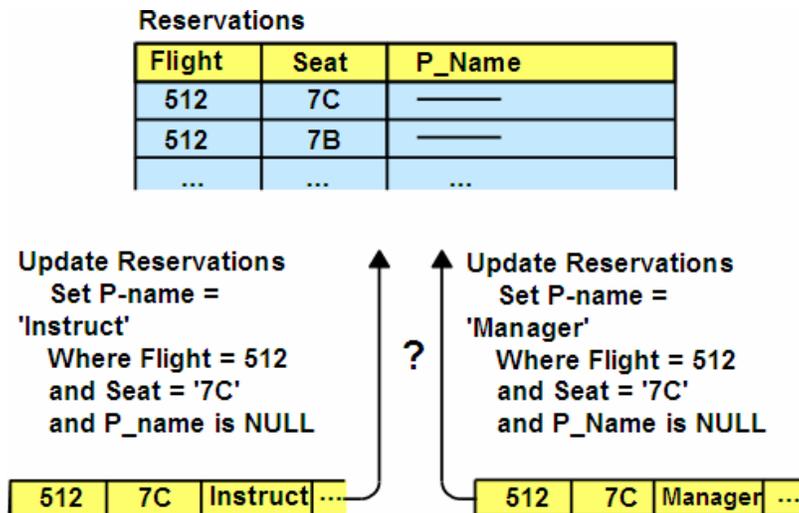


Abbildung 13.4 – Lost Update

In Abbildung 13.4 versuchen zwei Anwendungen die gleiche Zeile zu aktualisieren. Bei der Anwendung links handelt es sich um App1 und bei Anwendung rechts um App2. Die Reihenfolge der Ereignisse ist dann:

1. App1 aktualisiert eine Zeile
2. App2 aktualisiert dieselbe Zeile
3. App1 committed
4. App2 committed

Die von App1 vorgenommene Aktualisierung geht verloren sobald App2 eine Aktualisierung durchführt, woraus sich die Bezeichnung "Lost Update" ableitet.

13.3.2 Uncommitted read

Ein uncommitted read (d.h. eine Transaktion liest einen nicht-festgelegten Zustand oder eine ohne mit Commit abgeschlossene Transaktion), oder "dirty read", gestattet einer Anwendung Informationen zu lesen, welche noch nicht abgeschlossen (committed) wurden, und aus diesem Grund nicht zwingend akkurat sind.

Reservations

Flight	Seat	P_Name
512	7C	_____
512	7B	_____
...	...	

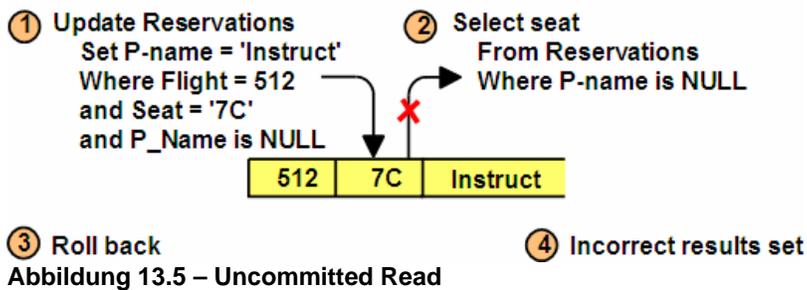


Abbildung 13.5 – Uncommitted Read

Die Reihenfolge der Ereignisse in Abbildung 13.5 ist wie folgt:

1. App1 aktualisiert eine Zeile
2. App2 liest den neuen Wert aus dieser Zeile
3. App1 macht die Änderungen an dieser Zeile rückgängig (rolls back)

App2 liest Daten, welche noch nicht committed wurden, und somit ungültige Daten. Dies erklärt warum das Problem "uncommitted read" genannt wird.

13.3.3 Non-repeatable read

Non-repeatable read impliziert, dass Sie nicht dasselbe Resultat erzielen können nach dem Sie denselben Lesevorgang in der gleichen Operation ausgeführt haben.

FLIGHT	SEAT	NAME	DESTINATION	ORIGIN
512	7B	_____	DENVER	DALLAS
....				
....				
814	8A	_____	SAN JOSE	DENVER
....				
134	1C	_____	HONOLULU	SAN JOSE
....			

Abbildung 13.6 – Non-repeatable Read

Erwägen Sie, wie in Abbildung 13.6 gezeigt, zu versuchen einen Flug von Dallas nach Honolulu zu buchen. Die Reihenfolge der Ereignisse ist wie folgt:

1. App1 öffnet einen Cursor (auch bekannt als Result Set) und erhält die in Abbildung 13.6 dargestellten Daten
2. App2 löscht eine für den Cursor qualifizierte Zeile (Zum Beispiel die Zeile mit dem Ziel (destination) "San Jose")
3. App2 committed die Änderungen
4. App1 schließt und öffnet den Cursor erneut

Da App1 in diesem Fall nicht die gleichen Daten bei repeated read erhalten wird, kann diese die Daten nicht reproduzieren; weshalb dieses Problem non-repeatable read genannt wird.

13.3.4 Phantom read

Das Phantom read Problem ist vergleichbar dem non-repeatable read Problem, jedoch besteht der Unterschied darin, dass Sie bei nachfolgenden Lesevorgängen (fetch) möglicherweise zusätzliche Zeilen statt weniger Zeilen erhalten. Abbildung 13.7 liefert ein Beispiel dieses Problems:

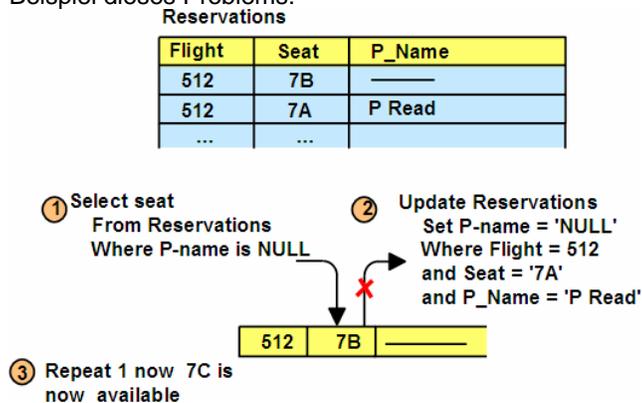


Abbildung 13.7 – Phantom read

Abbildung 13.7 zeigt die folgende Ereignisfolge:

1. App1 öffnet einen Cursor
2. App2 fügt eine Zeile zu der für den Cursor qualifizierten Datenbank hinzu
3. App2 committed die Änderungen
4. App1 schließt und öffnet den Cursor erneut

In diesem Fall wird App1 nicht dieselben Daten bei einem repeatable read erhalten, sondern mehr Zeilen; weshalb dieses Problem "phantom read" genannt wird.

13.4 Isolationstufen (Isolation Level)

Sie können Isolationstufen mit Sperregeln vergleichen welche, abhängig von der gewählten Isolationstufe, eventuell in einem unterschiedlichen Verhalten bei Datenbanksperren mit einer Anwendung resultieren.

DB2 bietet verschiedene Stufen um Daten zu isolieren:

- ▶ Uncommitted Read (UR) - Eine Transaktion liest einen nicht-festgelegten Zustand oder eine ohne mit Commit abgeschlossene Transaktion
- ▶ Cursor Stability (CS) - Cursorstabilität
- ▶ Read Stability (RS) - Lesestabilität
- ▶ Repeatable Read (RR) - Wiederholtes Lesen

13.4.1 Uncommitted Read

Uncommitted Read ist auch als Dirty Read bekannt. Es handelt sich dabei um die niedrigste Isolationsstufe und erlaubt die höchste Stufe des gleichzeitigen Zugriffs. Bei Lesevorgängen werden keine Sperrern eingeholt, außer eine andere Anwendung versucht eine Tabelle zu löschen (drop) oder zu ändern (alter), und Aktualisierungen (update) verhalten sich wie bei Verwendung der Cursorstabilitätsisolationsstufe.

Auf dieser Isolationsstufe weiterhin mögliche Probleme:

- ▶ Uncommitted read
- ▶ Nichtwiederholbares Lesen (Non-repeatable read)
- ▶ Phantom read (Suchkriterien treffen während einer Transaktion auf unterschiedliche Datensätze zu)

Durch diese Isolationsstufe verminderte Probleme:

- ▶ Verlust der Aktualisierung (Loss of update)

13.4.2 Cursor Stability

Bei Cursor Stability handelt es sich um die Standardisolationsstufe. Sie bietet einen minimalen Grad des Sperrern. Grundsätzlich wird mit dieser Isolationsstufe die "aktuelle" Zeile eines Cursors gesperrt. Falls die Zeile nur gelesen wird, wird die Sperre gehalten bis auf eine neue Zeile positioniert oder die Arbeitseinheit beendet wird. Falls die Zeile aktualisiert wird, bleibt die Sperre bis zum Abschluss der Arbeitseinheit bestehen.

Auf dieser Isolationsstufe weiterhin mögliche Probleme:

- ▶ Non-repeatable read
- ▶ Phantom read

Durch diese Isolationsstufe verminderte Probleme:

- ▶ Loss of update
- ▶ Uncommitted read

13.4.3 Read Stability

Bei Read Stability werden alle von einer Anwendung abgerufenen Zeilen einer Arbeitseinheit gesperrt. Bei einem gegebenen Cursor werden alle durch diesen qualifizierten Zeilen des Result Sets (n Zeilen einer Datenbank) gesperrt. Wenn Sie zum Beispiel eine Tabelle mit 10.000 Zeilen haben und die Abfrage 10 Zeilen zurückgibt, werden nur diese 10 Zeilen gesperrt. Read Stability verwendet einen moderaten Grad des Sperrern.

Auf dieser Isolationsstufe weiterhin mögliche Probleme:

- ▶ Phantom read

Durch diese Isolationsstufe verhinderte Probleme:

- ▶ Loss of update
- ▶ Uncommitted read
- ▶ Non-repeatable read

13.4.4 Repeatable Read

Bei Repeatable Read handelt es sich um die höchste Isolationsstufe. Sie bietet den höchsten Grad des Sperrens und die wenigsten gleichzeitigen Zugriffe. Sperrern werden auf alle Zeilen gehalten, welche zum Aufbau des Result Sets (n Zeilen einer Datenbank) erforderlich sind, mit der Folge, dass Zeilen, welche für das endgültige Result Set nicht benötigt werden möglicherweise ebenfalls gesperrt werden. Keine andere Anwendung kann eine Zeile aktualisieren (update), löschen (delete) oder einfügen (insert), welche eine Zeile des Result Sets beeinträchtigen würde, bis die Arbeitseinheit beendet ist. Repeatable Read stellt sicher dass, wenn dieselbe Abfrage während einer Arbeitseinheit mehrmals abgesetzt wird, jedesmal dasselbe Resultat zurückgegeben wird.

Auf dieser Isolationsstufe weiterhin mögliche Probleme:

- ▶ keine

Durch diese Isolationsstufe verhinderte Probleme:

- ▶ Loss of update
- ▶ Uncommitted read
- ▶ Non-repeatable read
- ▶ Phantom read

13.4.5 Vergleich der Isolationsstufen

Abbildung 13.8 vergleicht die verschiedenen Isolationsstufen für ein fetch. Aus der Abbildung können wir ersehen, dass die Isolationsstufe Uncommitted Read (UR); d.h. eine Transaktion liest einen nicht-festgelegten Zustand, keine Sperrern setzt. Die Isolationsstufe Cursor Stability (CS) setzt eine Sperre auf Zeile 1 wenn es diese liest und hebt diese auf sobald Zeile 2 gelesen wird, und so weiter. Bei den Isolationsstufen Read Stability (RS) oder Repeatable Read (RS) wird jede gelesene Zeile gesperrt und die Sperre wird erst nach Beendigung der Transaktion aufgehoben (Zeitpunkt des commits).

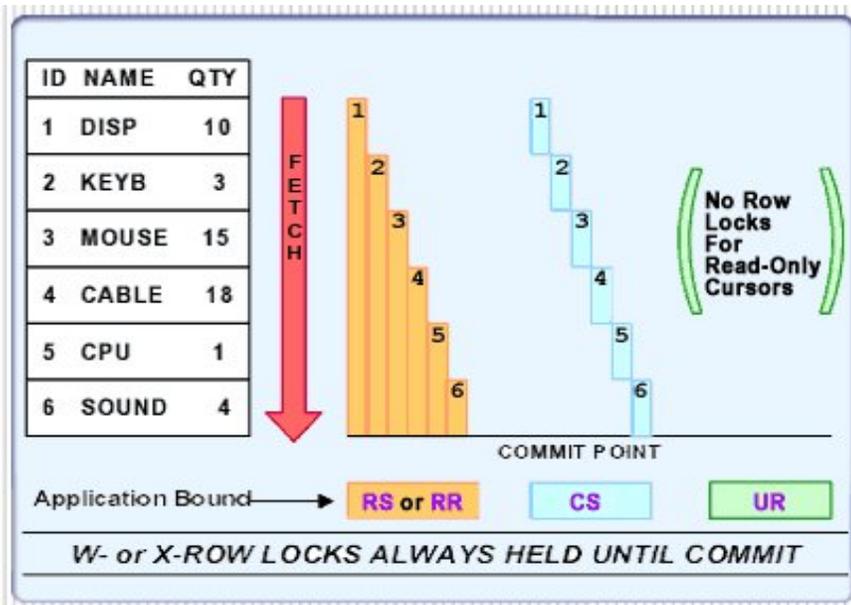


Abbildung 13.8 – Vergleich der Isolationsstufen für ein fetch

13.4.6 Setzen der Isolationsstufe

Isolationsstufen können auf mehreren Stufen spezifiziert werden:

- ▶ Sitzung (Session)
- ▶ Verbindung (Connection)
- ▶ Anweisung (Statement)

Die Isolationsstufe wird normalerweise auf Sitzungs- oder Anwendungsstufe definiert. Falls in Ihrer Anwendung keine Isolationsstufe spezifiziert wurde, wird standardmäßig Cursorstabilität verwendet. Tabelle 13.1. zeigt beispielsweise die möglichen Isolationsstufen für ein .NET oder JDBC Programm und wie diese Eigenschaften, falls gesetzt, den DB2 Isolationsstufen entsprechen.

DB2	.NET	JDBC
Uncommitted Read (UR)	ReadUncommitted	TRANSACTION_READ_UNCOMMITTED
Cursor Stability (CS)	ReadCommitted	TRANSACTION_READ_COMMITTED
Read Stability (RS)	RepeatableRead	TRANSACTION_REPEATABLE_READ
Repeatable Read (RR)	Serializable	TRANSACTION_SERIALIZABLE

Tabelle 13.1 – Vergleich der Isolationsstufenterminologie

In einer Anweisung kann die Isolationsstufe mit der WITH {isolation level} Klausel gesetzt werden. Zum Beispiel:

```
SELECT ... WITH {UR | CS | RS | RR}
```

Fallbeispiel:

Eine Anwendung benötigt eine "ungefähre" Zahl der in einer Tabelle vorhandenen Zeilen. Leistung ist von höchster Wichtigkeit. Mit Ausnahme einer SQL Anweisung ist die Isolationsstufe Cursorstabilität erforderlich:

```
SELECT COUNT(*) FROM tabl WITH UR
```

Bei eingebettetem (embedded) SQL wird die Isolationsstufe zum Zeitpunkt des Bindens und bei dynamischem SQL zur Laufzeit gesetzt.

Die Auswahl der zu verwendenden Isolationsstufe ist von Ihrer Anwendung abhängig. Falls Ihre Anwendung, wie im obigen Beispiel, keine genaue Anzahl benötigt, wählen sie die Isolationsstufe UR. Sollte Ihre Anwendung eine sehr enge Kontrolle über die Daten, mit welchen sie arbeitet, benötigen wählen sie die Isolationsstufe RR.

13.5 Lock escalation

Jede durch DB2 gesetzte Sperre verbraucht etwas Speicher. Wenn der Optimierer der Meinung ist das es besser wäre eine Sperre auf die ganze Tabelle statt auf mehrere Zeilen zu setzen tritt lock escalation (Sperre wurde konvertiert) auf. Abbildung 13.9 illustriert dies.

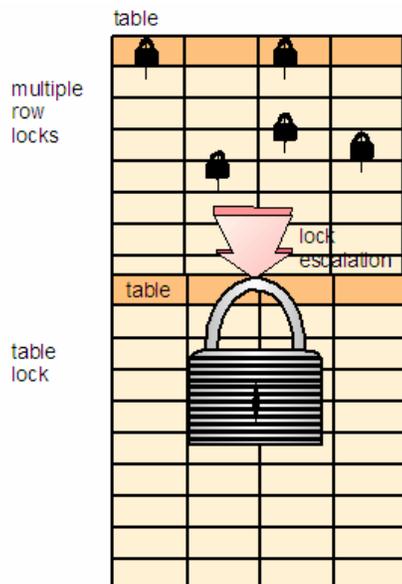


Abbildung 13.9 – Lock escalation

Es gibt zwei hauptsächliche auf lock escalation bezogene Datenbankkonfigurationsparameter:

- ▶ **LOCKLIST** – Die Speichermenge (in 4 KB Seiten), welche zur Verwaltung der Sperren für alle verbundenen Anwendungen verwendet wird. Der Standardwert beträgt fünfzig 4 KB Seiten (200 KB) unter Windows.
- ▶ **MAXLOCKS** – Der maximale Prozentsatz der gesamten Sperrliste, welche eine einzelne Anwendung verbrauchen darf. Der Standardwert beträgt 22%.

Deshalb wird, falls die Standardwerte verwendet werden, lock escalation auftreten, wenn eine einzelne Anwendung mehr als 44 KB des für Sperren verwendeten Speichers (200 KB * 22% = 44 KB) benötigt. Falls lock escalation regelmäßig auftritt, sollten die Werte für LOCKLIST und MAXLOCKS erhöht werden. Lock escalation beeinträchtigt die Leistung und reduziert die möglichen gleichzeitigen Zugriffe. Die DB2 Diagnose Log Datei (db2diag.log, welche sich normalerweise im Verzeichnis C:\Program Files\IBM\SQLLIB\DB2 befindet) kann verwendet werden um festzustellen ob lock escalation auftritt.

13.6 Überwachung von Sperren (Lock monitoring)

Sie können die Verwendung von Sperren mit dem DB2 Momentaufnahme Monitor überwachen. Um Momentaufnahmen (Snapshots) für Sperren zu aktivieren, verwenden Sie diesen Befehl:

```
UPDATE MONITOR SWITCHES USING LOCK ON
```

Nachdem dieser Schalter eingeschaltet wurde, werden Daten für Momentaufnahmen gesammelt. Um einen Bericht der Sperrern zu einem gegebenen Zeitpunkt zu erhalten führen Sie diesen Befehl aus:

```
GET SNAPSHOT FOR LOCKS FOR APPLICATION AGENTID <handle>
```

Abbildung 13.9 zeigt die Ausgabe eines Beispiels der Momentaufnahme einer Anwendungssperre.

```
Application Lock Snapshot

Snapshot timestamp           = 11-05-2002 00:09:08.672586

Application handle           = 9
Application ID                = *LOCAL.DB2.00B9C5050843
Sequence number              = 0001
Application name              = db2bp.exe
Authorization ID              = ADMINISTRATOR
Application status            = UOW Waiting
Status change time           = Not Collected
Application code page         = 1252
Locks held                    = 4
Total wait time (ms)         = 0

List Of Locks
Lock Name                    = 0x05000700048001000000000052
Lock Attributes               = 0x00000000
Release Flags                 = 0x40000000
Lock Count                    = 255
Hold Count                    = 0
Lock Object Name              = 98308
Object Type                   = Row
Tablespace Name               = TEST4K
Table Schema                  = ADMINISTRATOR
Table Name                    = T2
Mode                          = X
```

Abbildung 13.9 – Momentaufnahme einer Anwendungssperre

13.7 Zeitlimit einer Sperrung (Lock wait)

Wenn zwei oder mehr Anwendungen Operationen auf demselben Objekt ausführen müssen, muss eine der Anwendungen möglicherweise warten um die benötigte Sperre zu erhalten. Standardmäßig wird eine Anwendung unendlich warten. Die Zeit, wie lange eine Anwendung auf eine Sperre wartet, wird durch den Konfigurationsparameter LOCKTIMEOUT der Datenbank kontrolliert. Der Standardwert ist -1 (unendliches Warten).

Das CURRENT LOCK TIMEOUT Register kann für eine gegebene Verbindung verwendet werden. Standardmäßig ist dieses Register auf den Wert von LOCKTIMEOUT gesetzt. Zur Änderung des Wertes verwenden Sie den SET LOCK TIMEOUT Befehl. Sobald der Wert des Registers für eine Verbindung gesetzt ist, wird dieser während aller Transaktionen bestehen bleiben.

Beispiel:

```
SET LOCK TIMEOUT=WAIT n
```

13.8 Deadlock – Ursachen und Erkennung

Ein Deadlock tritt auf wenn zwei oder mehr Anwendungen, welche mit derselben Datenbank verbunden sind, unendlich auf eine Ressource warten. Das Warten wird niemals aufgelöst, da jede Anwendung eine Ressource hält, welche die andere Anwendung benötigt. Bei Deadlocks handelt es sich meistens um Fehler im Anwendungsentwurf. Abbildung 13.10 illustriert ein Deadlock.

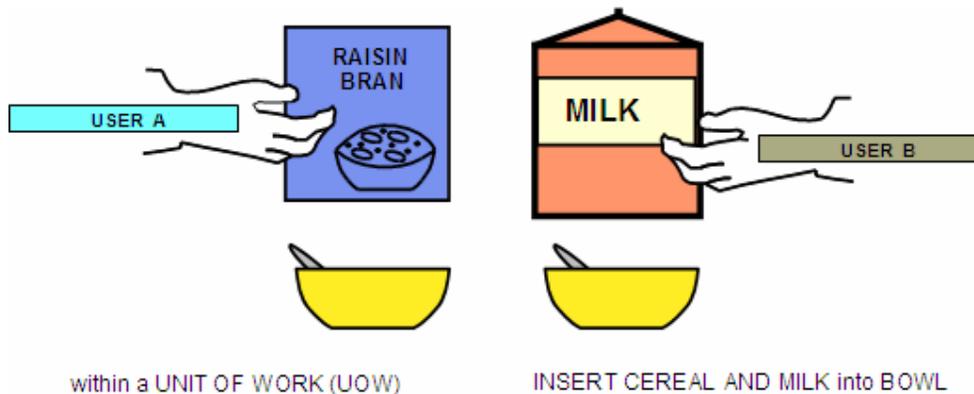


Abbildung 13.10 – Ein Deadlock Szenario

In Abbildung 13.10 hält Benutzer A das Rosinen Müsli (raisin bran) und wird es nicht loslassen bis er die Milch erhält. Auf der anderen Seite hält Benutzer B die Milch und wird diese solange nicht loslassen bis er das Rosinen Müsli (raisin bran) erhält. Folglich haben wir eine Deadlock Situation.

Um eine Deadlock Situation in DB2 zu simulieren, führen Sie die nachfolgenden Schritte aus:

1. Öffnen Sie zwei DB2 Befehlsfenster (welche wir "CLP1" und "CLP2" nennen). Diese repräsentieren zwei unterschiedliche sich mit der Datenbank verbindende Anwendungen.
2. Bei CLP1 setzen Sie diese Befehle ab:

```
db2 connect to sample
db2 +c update employee set firstnme = 'Mary' where empno =
'000050'
```

Zuerst verbinden wir uns mit der SAMPLE Datenbank und setzen anschließend einen update Befehl auf die Zeile mit "empno = 50000" der employee Tabelle ab. Die "+c" Option in dieser Anweisung gibt an, dass wir nicht wollen dass das DB2 Befehlsfenster den Befehl automatisch committed. Das machen wir, um die Sperren beizubehalten, mit Absicht.

3. Bei CLP2 setzen Sie diese Befehle ab:

```
db2 connect to sample
```

```
db2 +c update employee set firstnme = 'Tom' where empno =  
'000030'
```

Im CLP2 Fenster, welches die zweite Anwendung repräsentiert, verbinden wir uns ebenfalls mit der SAMPLE Datenbank, aktualisieren jedoch eine andere Zeile in der employee Tabelle.

4. Bei CLP1 setzen Sie diesen Befehl ab:

```
db2 +c select firstnme from employee where empno = '000030'
```

Nachdem Sie die Eingabetaste betätigt habe, um die obige SQL Anweisung auszuführen, könnte es so aussehen als ob das SELECT hinge. Tatsächlich hängt es nicht, sondern wartet auf die Freigabe der von CLP2 auf diese Zeile in Schritt 3 gesetzten exklusiven Sperrern. Zu diesem Zeitpunkt, falls LOCKTIMEOUT auf den Standardwert -1 belassen worden wäre, würde die CLP1 Anwendung für immer warten.

5. Bei CLP2 setzen Sie diesen Befehl ab:

```
db2 +c select firstnme from employee where empno = '000050'
```

Durch das Absetzen der obigen SQL Anweisung kreieren wir jetzt ein Deadlock. Diese SELECT Anweisung scheint ebenfalls zu hängen, da sie auf die Freigabe der exklusiven, von CLP1 in Schritt 2 gesetzten, Sperre wartet.

Im obigen Deadlock Szenario wird DB2 den Datenbank Konfigurationsparameter DLCHKTIME prüfen. Dieser Parameter setzt das Zeitintervall, in welchem Deadlocks geprüft werden. Sollte dieser Parameter zum Beispiel auf 10 Sekunden eingestellt sein, wird DB2 alle 10 Sekunden prüfen ob ein Deadlock vorliegt. Sollte es tatsächlich zu einem Deadlock gekommen sein, wird DB2 einen internen Algorithmus verwenden um herauszufinden welche der zwei Transaktionen zurückgesetzt (rolled back) und welche fortgesetzt wird.

Wenn Sie zahllose Deadlocks feststellen, sollten sie existierende Transaktionen erneut überprüfen um festzustellen, ob eine Umstrukturierung möglich ist.

13.9 Empfohlene Vorgehensweise bei gleichzeitigem Zugriff und Datensatzsperrern

Die folgenden, in dieser Reihenfolge zu beachtenden, Tipps ermöglichen den bestmöglichen gleichzeitigen Zugriff:

1. Halten Sie Transaktionen so kurz wie möglich. Dies kann durch das wiederholte Absetzen von COMMIT Befehlen (selbst für nur lesende Transaktionen) erreicht werden, falls ihre Anwendungslogik dies zulässt.
2. Protokollieren Sie Transaktionsinformationen nur bei Bedarf.

3. Säubern Sie Daten schnell mittels:

```
ALTER TABLE ACTIVATE NOT LOGGED INITIALLY WITH EMPTY TABLE
```

4. Führen Sie Datenänderungen in Stapeln/Gruppen aus. Zum Beispiel:

```
DELETE FROM (  
  SELECT * FROM tedwas.t1 WHERE c1 = ... FETCH FIRST 3000 ROWS  
  ONLY)
```

5. Verwenden Sie die Möglichkeiten des gleichzeitigen Zugriffs bei den DB2 Import- und Exportdienstprogrammen.
6. Setzen Sie den Datenbankparameter LOCKTIMEOUT (empfohlene Zeiten liegen zwischen 30 und 120 Sekunden). Belassen Sie diesen nicht auf den Standardwert von -1. Sie können auch den Sitzungsbasierten Wert verwenden.
7. Fragen Sie nicht mehr Daten ab als notwendig. Sie können zum Beispiel die FETCH FIRST n ROWS ONLY Klausel in einer SELECT Anweisung verwenden.

**TEIL III – DB2 LERNEN:
ANWENDUNGSENTWICKLUNG**

14

Kapitel 14 – SQL PL gespeicherte Prozeduren

In diesem Kapitel behandeln wir gespeicherte Prozeduren. Bei einer gespeicherten Prozedur handelt es sich um eine Datenbankanwendung, welche eingekapselte SQL Anweisungen und Geschäftslogik (Business Logic) beinhalten kann. Das halten Teiler der Anwendungslogik in der Datenbank liefert Performanceverbesserungen, da der Netzwerkverkehr zwischen der Anwendung und der Datenbank erheblich abnimmt. Zusätzlich bieten gespeicherte Prozeduren einen zentralen Ort für die Speicherung Ihres Codes, so dass andere Anwendungen dieselben Prozeduren wiederverwenden können.

DB2 gespeicherte Prozeduren können unter Verwendung von SQL PL, C/C++, Java, Cobol, CLR (Common Language Run-time) unterstützenden Sprachen und OLE entwickelt werden. In diesem Kapitel richten wir, auf Grund ihrer Popularität und Einfachheit, unser Hauptaugenmerk auf SQL PL Prozeduren.

Abbildung 14.1 illustriert die Funktionsweise von gespeicherten Funktionen.

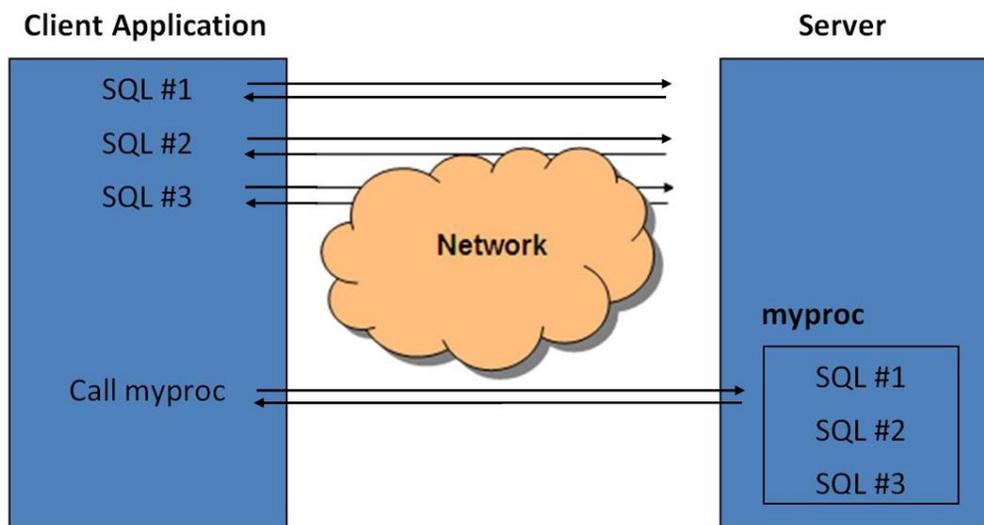


Abbildung 14.1 – Reduzierung des Netzwerkverkehrs mit gespeicherten Prozeduren

In der linken oberen Ecke der Abbildung sehen Sie mehrere nacheinander ausgeführte SQL Anweisungen. Jede SQL Anweisung wird vom Client zum Server gesandt und der

Server gibt das Ergebnis an den Client zurück. Wenn viele SQL Anweisungen so ausgeführt werden nimmt der Netzwerkverkehr zu. Andererseits sehen Sie in der rechten unteren Ecke die auf dem Server gehaltene gespeicherte Prozedur "myproc", welche dasselbe SQL enthält und auf dem Client (auf der linken Seite) wird die CALL Anweisung zum Aufruf der Prozedur verwendet. Diese zweite Methode des Aufrufs einer Prozedur ist effizienter, da lediglich eine Call Anweisung durch das Netz gesendet und ein Resultat an den Client zurückgegeben wird.

Gespeicherte Prozeduren können ebenfalls für Sicherheitsaspekte Ihrer Datenbank hilfreich sein. So können Sie zum Beispiel festlegen, dass Benutzer auf Tabellen oder Sichten nur über gespeicherte Prozeduren zugreifen. Dies hilft den Server abzusichern und Benutzer daran zu hindern auf Informationen zuzugreifen, welche nicht für sie bestimmt sind. Dies ist möglich, da Benutzer keine expliziten Rechte auf über gespeicherte Prozeduren zugegriffene Tabellen oder Sichten benötigen; sie benötigen lediglich ausreichend Rechte für den Aufruf der gespeicherten Prozedur.

Hinweis:

Für weitere Informationen über SQL PL gespeicherte Prozeduren können Sie sich dieses Video ansehen:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4343>

14.1 Das IBM Data Studio

Beim IBM Data Studio handelt es sich um eine umfassende Lösung, welche Ihnen hilft, Datenbankanwendungen während des gesamten Lebenszyklus der Daten zu entwickeln und verwalten. Einige der mit Data Studio durchführbaren Aufgaben sind:

- Erstellen, ändern und löschen von DB2 Datenbankobjekten (mit Auswirkungsanalyse)
- Durchsuchen und bearbeiten von Daten - Relational und XML
- Visuell SQL und XQuery Anweisungen aufbauen
- Abfragen mit Visual Explain optimieren
- Gespeicherte Prozeduren, SQL und Java™, entwickeln, debuggen und implementieren
- Entwickeln von benutzerdefinierten Funktionen (UDFs)
- Entwickeln von SQLJ Anwendungen
- Entwickeln von Abfragen und Routinen für pureXML Anwendungen
- Datenkopieraufgaben durchführen
- Zusammenarbeit und teilen von Projekten mit Teammitgliedern
- Schnelles erstellen von SOAP und REST Web Services
- Erkennen von Datenbankobjektbeziehungen mit physikalischen Datenmodellen (Diagramme)
- Visualisierung der Datenverteilung über Tabellen

IBM Data Studio basiert auf der Eclipse Plattform. Es handelt sich um ein separates Image (d.h. es ist nicht Bestandteil des DB2 Installationsimages), jedoch ebenfalls kostenlos erhältlich. Data Studio Images können über die "Download" Registerkarte auf der ibm.com/db2/express Website heruntergeladen werden. Abbildung 14.2 zeigt das IBM Data Studio.

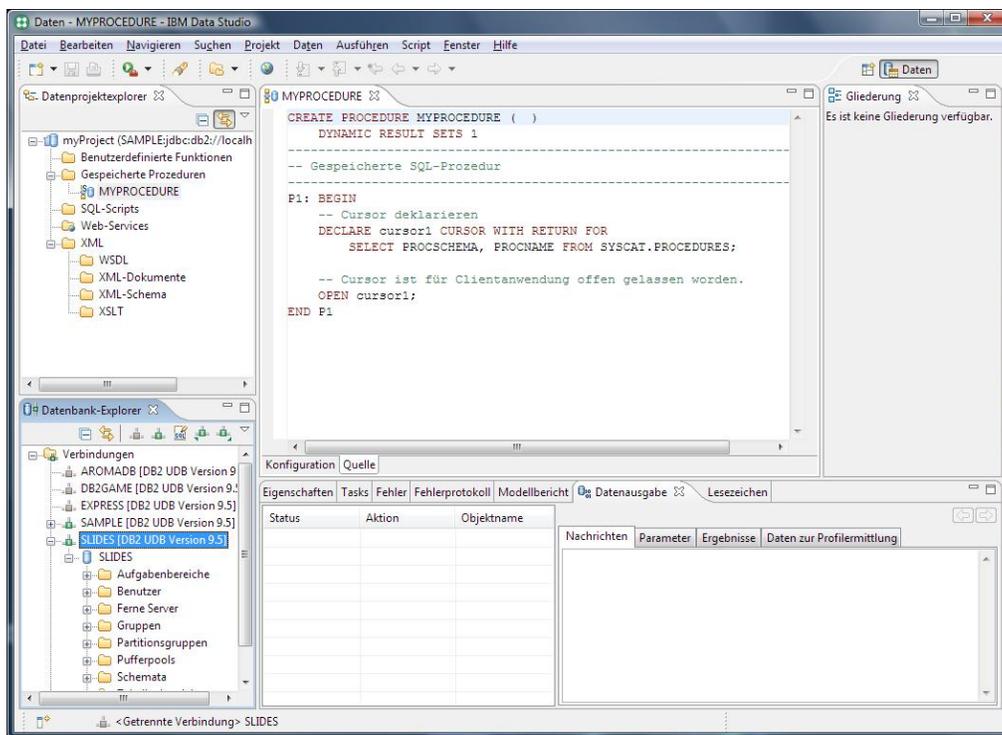


Abbildung 14.2 – Das IBM Data Studio

14.1.2 Erstellung einer gespeicherten Prozedur im Data Studio

Um eine Java oder SQL PL gespeicherte Prozedur im Data Studio zu erstellen, befolgen Sie die nachfolgenden Schritte. Bitte beachten Sie, dass in anderen Sprachen geschriebene Prozeduren nicht im Data Studio erstellt werden können.

Schritt 1: Erstellen eines Data Studio Projekts

Aus dem Data Studio Menü wählen Sie *Datei -> Neu -> Projekt* und wählen *Datenentwicklungsprojekt*. Dies wird Abbildung 14.3 gezeigt.

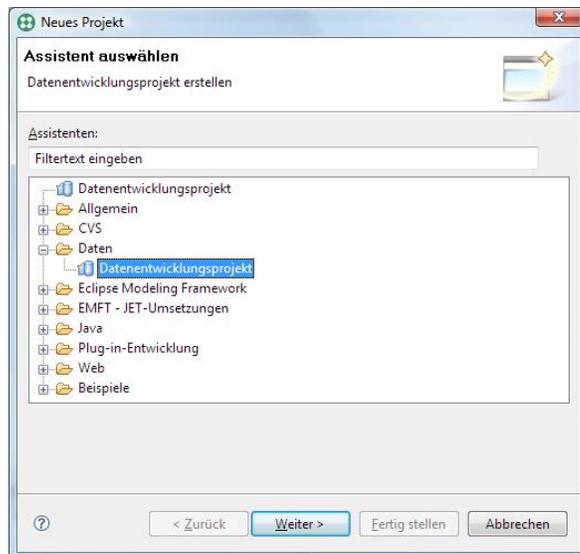


Abbildung 14.3 – Das Datenentwicklungsprojekt

Folgen Sie den Schritten des Assistenten zur Eingabe eines Namens für ihr Projekt, geben die Datenbank, zu welcher sie eine Verbindung aufbauen wollen, an und spezifizieren das JDK Verzeichnis (das standardmäßig vorgegebene ist normalerweise das richtige).

Schritt 2: Erstellen einer gespeicherten Prozedur

Sobald das Projekt erstellt wurde, zeigt die linke Seite der Datenperspektive Ihr Projekt. In Abbildung 14.4 sehen Sie das erstellte und erweiterte Projekt "myProject".

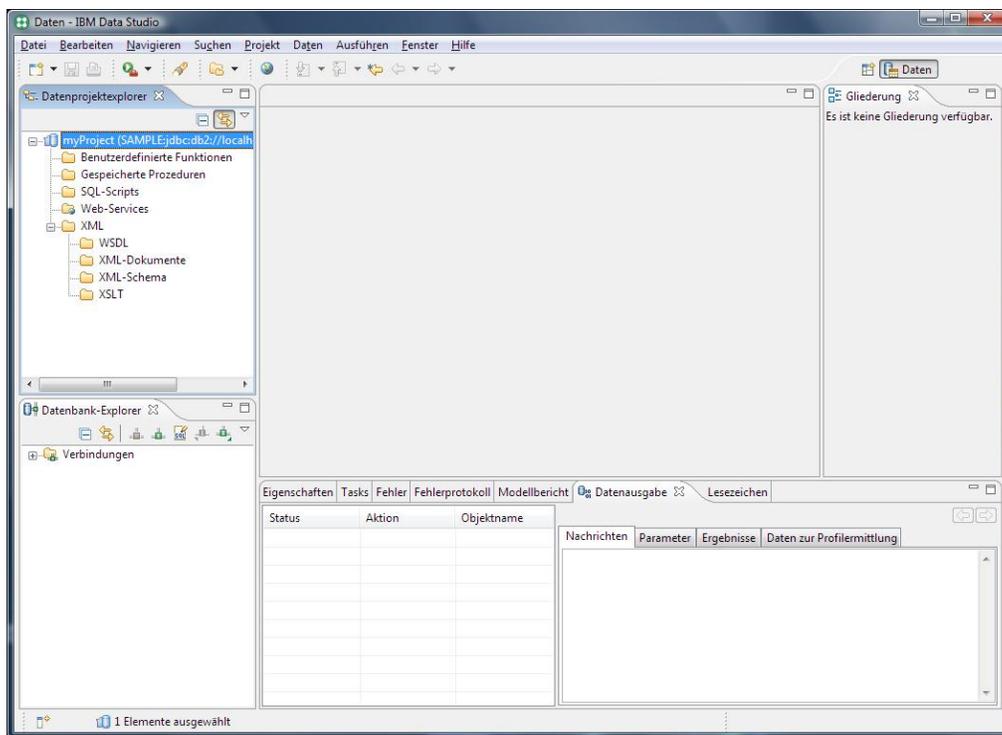


Abbildung 14.4 – Das “myProject” Projekt

Abbildung 14.4 zeigt verschiedene Ordner für ihr Projekt. Wenn Sie eine gespeicherte Prozedur erstellen möchten, klicken sie rechts auf den Gespeicherte Prozeduren Ordner und wählen *Neu -> Gespeicherte Prozedur*. Vervollständigen Sie vom Neue gespeicherte Prozedur Assistenten verlangten Informationen, wie das Projekt, mit welchem die Prozedur assoziiert ist, den Namen und die Sprache der Prozedur (beachten sie bitte, dass ausschließlich SQL PL und Java vom IBM Data Studio unterstützt werden) und die in der Prozedur zu verwendenden SQL Anweisungen. Standardmäßig gibt Data Studio Ihnen eine SQL Anweisung als Beispiel vor. An diesem Punkt angelangt, können Sie auf die *Fertig stellen* Schaltfläche klicken und eine gespeicherte Prozedur wird unter Verwendung von etwas Vorlagencode erstellt mitsamt der zuvor zur Verfügung gestellten SQL Anweisung als ein Beispiel. Dies ist in Abbildung 14.5 gezeigt.

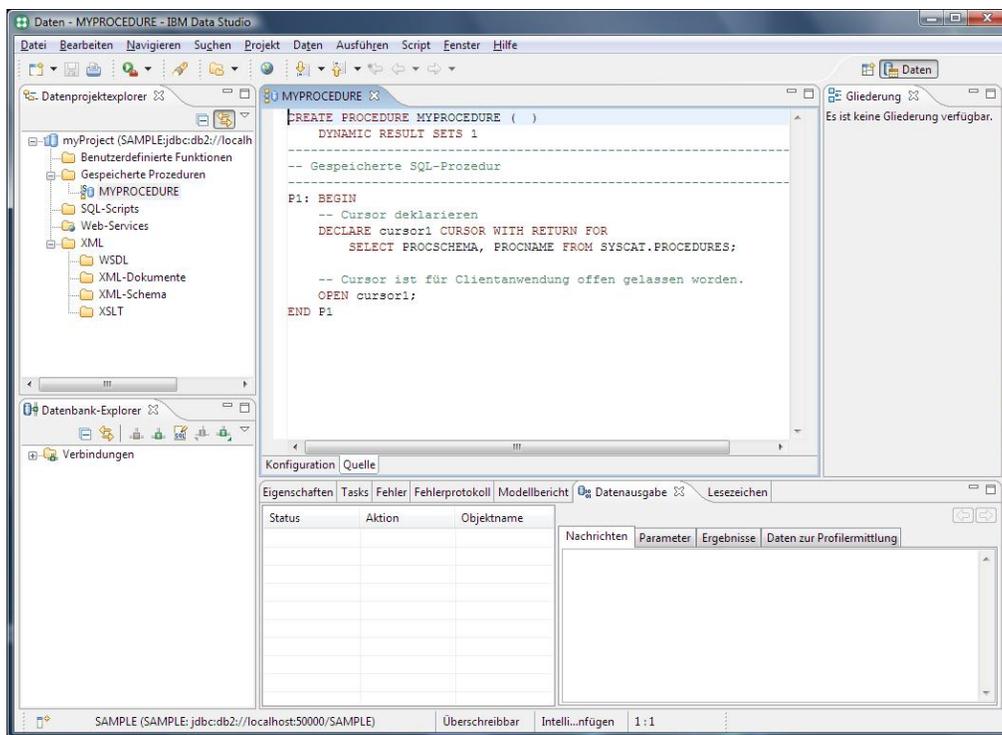


Abbildung 14.5 – Beispiel einer gespeicherten Prozedur

In Abbildung 14.5 wurde der Code des gespeicherten Prozedur Beispiels generiert. Sie können den gesamten Code mit ihrem eigenen Code ersetzen. Der Einfachheit halber werden wir in diesem Buch mit dem obigen Beispiel einer gespeicherten Prozedur fortfahren, wie wenn wir diese geschrieben hätten.

Schritt 3: Kompilieren (implementieren) einer gespeicherten Prozedur

Sobald die gespeicherte Prozedur erstellt ist, können Sie diese kompilieren und implementieren indem sie rechts auf das linke Teilfenster klicken und “Implementieren” wählen. Abbildung 14.6 illustriert diesen Schritt.

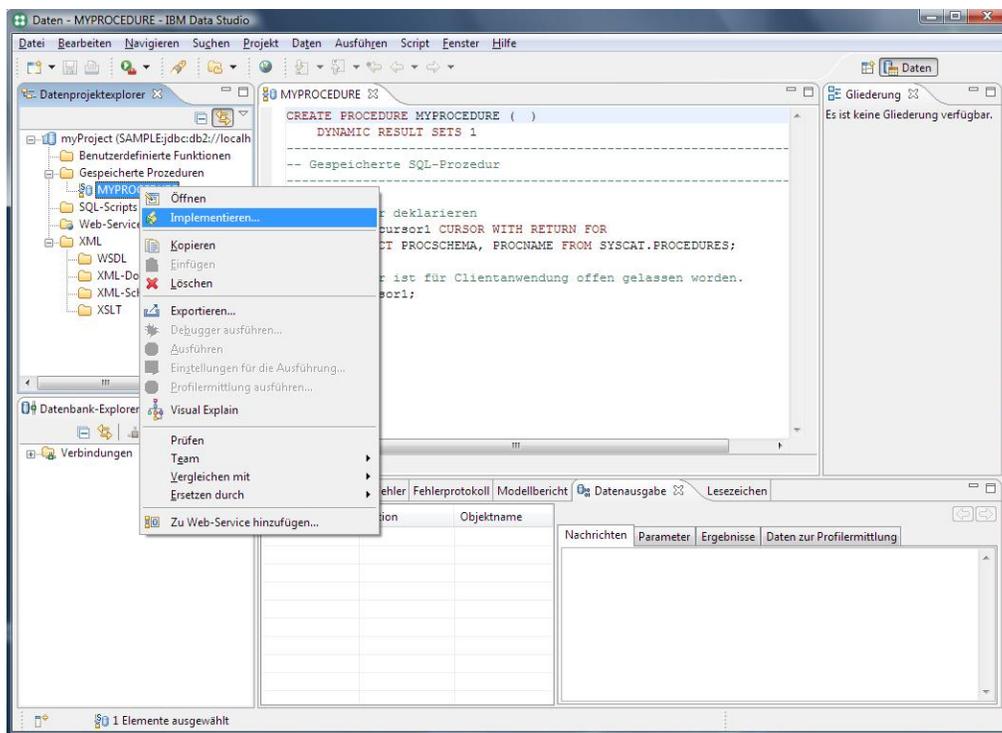


Abbildung 14.6 – Implementieren einer gespeicherten Prozedur

Schritt 4: gespeicherte Prozedur ausführen

Sobald die gespeicherte Prozedur implementiert wurde, kann diese durch einen Rechtsklick auf diese und die Auswahl "Ausführen" ausgeführt werden. Die Resultate werden in der "Ergebnisse" Registerkarte in der unteren rechten Ecke des Fensters angezeigt.

Um eine gespeicherte Prozedur über das Befehlsfenster oder den Befehlseditor auszuführen, können Sie den Befehl `CALL <procedure name>` verwenden.

14.2 Grundlagen SQL PL gespeicherter Prozeduren

SQL Procedural Language (SQL PL) gespeicherte Prozeduren sind leicht zu erstellen und erlernen. Sie besitzen die beste Performance in DB2. SQL PL gespeicherte Prozeduren (oder einfach "SQL gespeicherte Prozeduren") sind der Schwerpunkt dieses Kapitels.

14.2.1 Die Struktur gespeicherter Prozeduren

Die grundlegende Syntax zur Speicherung einer Prozedur wird hier gezeigt.

```
CREATE PROCEDURE proc_name [( {optional parameters} )]
                        [optional procedure attributes]      <statement>
```

Wobei es sich bei `<statement>` um eine alleinstehende Anweisung, oder einen Satz von durch `BEGIN [ATOMIC] ... END` gruppierte Anweisungen handelt.

14.2.2 Optionale Eigenschaften gespeicherter Prozeduren

Nachfolgend die Beschreibung einiger optionaler Eigenschaften gespeicherter Prozeduren:

- `LANGUAGE SQL`
Diese Eigenschaft bzw. Attribut gibt die verwendete Sprache der gespeicherten Prozedur an. `LANGUAGE SQL` ist der Standardwert. Für andere Sprachen, wie Java oder C, verwenden Sie `LANGUAGE JAVA` oder `LANGUAGE C`.
- `RESULT SETS <n>`
Dies ist erforderlich, falls Ihre gespeicherte Prozedur `n` Result Sets zurückgibt.
- `SPECIFIC my_unique_name`
Hierbei handelt es sich um einen eindeutigen Namen, welcher einer Prozedur gegeben werden kann. Eine gespeicherte Prozedur kann überladen (overloaded) werden, d.h. verschiedene gespeicherte Prozeduren können denselben Namen haben, jedoch mit einer unterschiedlichen Anzahl Parameter. Durch Verwendung des `SPECIFIC` Schlüsselworts können Sie einen einzigartigen Namen für jede dieser gespeicherten Prozeduren angeben und dadurch die Verwaltung der gespeicherten Prozeduren vereinfachen. Um zum Beispiel eine gespeicherte Prozedur unter Verwendung des `SPECIFIC` Schlüsselworts zu löschen, können Sie diesen Befehl absetzen: `DROP SPECIFIC PROCEDURE`. Falls das `SPECIFIC` Schlüsselwort nicht verwendet worden wäre, hätten Sie die `DROP PROCEDURE` Anweisung verwenden und den Namen der Prozedur mit den Parametern übergeben müssen, damit DB2 weiß welche der überladenen Prozeduren sie löschen wollen.

14.2.3 Parameter

Es gibt drei Arten von Parametern in einer SQL PL gespeicherten Prozedur:

- `IN` - Eingabeparameter
- `OUT` - Ausgabeparameter
- `INOUT` - Eingabe- und Ausgabeparameter

Zum Beispiel:

```
CREATE PROCEDURE proc(IN p1 INT, OUT p2 INT, INOUT p3 INT)
```

Wenn die Prozedur aufgerufen wird, müssen alle Parameter in der `CALL` Anweisung angegeben werden. Um zum Beispiel die obige gespeicherte Prozedur aufzurufen, würden Sie angeben:

```
CALL proc (10,?,4)
```

Das Fragezeichen (?) wird für Wertrückgabeparameter in der `CALL` Anweisung verwendet.

Nachfolgend ein weiteres Beispiel einer gespeicherten Prozedur, welches Sie ausprobieren können:

```
CREATE PROCEDURE P2 ( IN    v_p1 INT,
                    INOUT v_p2 INT,
                    OUT   v_p3 INT)

LANGUAGE SQL
SPECIFIC myP2
BEGIN
    -- my second SQL procedure
    SET v_p2 = v_p2 + v_p1;
    SET v_p3 = v_p1;
END
```

Um die Prozedur über den Befehlseditor auszurufen verwenden Sie:

```
call P2 (3, 4, ?)
```

14.2.4 Kommentare in einer SQL PL gespeicherten Prozedur

Es gibt zwei Möglichkeiten Kommentare in einer SQL PL gespeicherten Prozedur zu verwenden:

- Verwendung von zwei Bindestrichen. Zum Beispiel:
`-- This is an SQL-style comment`
- Verwendung eines Formats ähnlich der C Sprache. Zum Beispiel:
`/* This is a C-style coment */`

14.2.5 Zusammengesetzte Anweisungen

Eine zusammengesetzte Anweisung in einer gespeicherten Prozedur besteht aus mehreren Prozedurbefehlen und SQL Anweisungen, eingekapselt durch die Schlüsselwörter BEGIN und END. Wenn auf das BEGIN Schlüsselwort das ATOMIC Schlüsselwort folgt, wird die zusammengesetzte Anweisung als eine Einheit behandelt, d.h. alle Befehle und Anweisungen der zusammengesetzten Anweisung müssen erfolgreich sein damit die gesamte zusammengesetzte Anweisung erfolgreich ist. Falls eine Anweisung nicht erfolgreich war, werden alle Anweisungen zurückgesetzt (Roll-Back). Abbildung 14.7 zeigt die Struktur einer zusammengesetzten Anweisung.

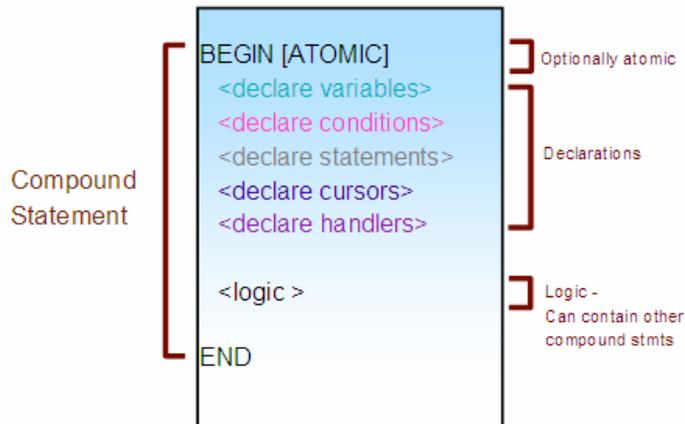


Abbildung 14.7 – Zusammengesetzte Anweisungen

14.2.6 Variablendeklaration

Um Variablen zu deklarieren verwenden Sie die DECLARE Anweisung:

```
DECLARE var_name <data type> [DEFAULT value];
```

Hier sind einige Beispiele:

```
DECLARE temp1 SMALLINT DEFAULT 0;
DECLARE temp2 INTEGER DEFAULT 10;
DECLARE temp3 DECIMAL(10,2) DEFAULT 100.10;
DECLARE temp4 REAL DEFAULT 10.1;
DECLARE temp5 DOUBLE DEFAULT 10000.1001;
DECLARE temp6 BIGINT DEFAULT 10000;
DECLARE temp7 CHAR(10) DEFAULT 'yes';
DECLARE temp8 VARCHAR(10) DEFAULT 'hello';
DECLARE temp9 DATE DEFAULT '1998-12-25';
DECLARE temp10 TIME DEFAULT '1:50 PM';
DECLARE temp11 TIMESTAMP DEFAULT '2001-01-05-12.00.00';
DECLARE temp12 CLOB(2G);
DECLARE temp13 BLOB(2G);
```

14.2.7 Zuweisungsbefehle

Um einer Variable einen Wert zuzuweisen verwenden Sie die SET Anweisung. Zum

Beispiel:

```
SET total = 100;
```

Die obige Anweisung ist gleichbedeutend mit

```
VALUES(100) INTO total;
```

Zusätzlich kann jede Variable auf NULL gesetzt werden:

```
SET total = NULL;
```

Ein Zustand wird ausgelöst wenn das Lesen von mehr als einer Zeile nur die erste Zeile einer Tabelle liest

```
SET total = (select sum(c1) from T1);
SET first_val = (select c1 from T1 fetch first 1 row only)
```

Sie können Variable auch gemäß externer Datenbankeigenschaften setzen:
 SET sch = CURRENT SCHEMA;

14.3 Cursor

Ein Cursor ist das Result Set, welches das Ergebnis einer SELECT Anweisung beinhaltet. Nachfolgend die Syntax um einen Cursor zu deklarieren, öffnen, Daten mit Fetch abzurufen oder zu schließen:

```
DECLARE <cursor name> CURSOR [WITH RETURN <return target>]
      <SELECT statement>;
OPEN <cursor name>;
FETCH <cursor name> INTO <variables>;
CLOSE <cursor name>;
```

Wenn ein Cursor deklariert wird, kann die WITH RETURN Klausel mit diesen Werten verwendet werden:

- CLIENT: Das Result Set wird an die Client Anwendung zurückgegeben
- CALLER: Das Result Set wird an den Client oder die gespeicherte Prozedur, welche den CALL aufrief, zurückgegeben

Nachfolgend das Beispiel einer einen Cursor verwendenden gespeicherten Prozedur:

```
CREATE PROCEDURE set()
DYNAMIC RESULT SETS 1
LANGUAGE SQL
BEGIN
DECLARE cur CURSOR WITH RETURN TO CLIENT
      FOR SELECT name, dept, job
      FROM staff
      WHERE salary > 20000;
OPEN cur;
END
```

14.4 Flusskontrolle

Wie in vielen anderen Programmiersprachen, besitzt SQL PL mehrere Befehle, welche zur Flusskontrolle der Logik verwendet werden können. Unten listen wir einige der unterstützten Flusskontrollbefehle auf:

```
CASE (wählt den Ausführungspfad (einfache Suche))
IF
FOR (führt den Rahmen für jede Zeile von Tabellen aus)
WHILE
ITERATE (erzwingt die nächste Iteration. Ähnlich zu CONTINUE in C)
```

LEAVE (verlässt einen Block oder eine Schleife. "Strukturiertes Goto")
 LOOP (unendliche Schleife)
 REPEAT
 GOTO
 RETURN
 CALL (Prozeduraufruf)

14.5 Aufruf gespeicherter Prozeduren

Der folgende Codeausschnitt zeigt, wie eine gespeicherte Prozedur bei Verwendung verschiedener Programmiersprachen aufgerufen werden kann.

Beispiel des Aufrufs einer gespeicherten Prozedur durch eine CLI/ODBC

Anwendung

```
SQLCHAR *stmt = (SQLCHAR *)
"CALL MEDIAN_RESULT_SET( ? )" ;
SQLDOUBLE sal = 20000.0; /* Bound to parameter marker in
stmt */
SQLINTEGER salind = 0; /* Indicator variable for sal */

sqlrc = SQLPrepare(hstmt, stmt, SQL_NTS);
sqlrc = SQLBindParameter(hstmt, 1, SQL_PARAM_OUTPUT,
SQL_C_DOUBLE, SQL_DOUBLE, 0, 0, &sal, 0, &salind);
SQLExecute(hstmt);

if (salind == SQL_NULL_DATA)
printf("Median Salary = NULL\n");
else
printf("Median Salary = %.2f\n\n", sal );

/* Get first result set */
sqlrc = StmtResultPrint(hstmt);
/* Check for another result set */
sqlrc = SQLMoreResults(hstmt);
if (sqlrc == SQL_SUCCESS) {
/* There is another result set */
sqlrc = StmtResultPrint(hstmt);
}
```

For more details, see the DB2 sample file: `sqllib/samples/sqlproc/rsultset.c`

Beispiel des Aufrufs einer gespeicherten Prozedur durch eine VB.NET Anwendung

```
Try
` Create a DB2Command to run the stored procedure
Dim procName As String = "TRUNC_DEMO"
Dim cmd As DB2Command = conn.CreateCommand()
Dim parm As DB2Parameter

cmd.CommandType = CommandType.StoredProcedure
cmd.CommandText = procName

` Register the output parameters for the DB2Command
```

```

    parm          = cmd.Parameters.Add("v_lastname",
DB2Type.VarChar)
    parm.Direction = ParameterDirection.Output
    parm          = cmd.Parameters.Add("v_msg",
DB2Type.VarChar)
    parm.Direction = ParameterDirection.Output

    ` Call the stored procedure
    Dim reader As DB2DataReader = cmd.ExecuteReader

Catch myException As DB2Exception
    DB2ExceptionHandler(myException)
Catch
    UnhandledExceptionHandler()
End Try

```

Beispiel des Aufrufs einer gespeicherten Prozedur durch eine Java Anwendung

```

try
{
    // Connect to sample database
    String url = "jdbc:db2:sample";
    con = DriverManager.getConnection(url);

    CallableStatement cs = con.prepareCall("CALL
trunc_demo(?, ?)");

    // register the output parameters
    callStmt.registerOutParameter(1, Types.VARCHAR);
    callStmt.registerOutParameter(2, Types.VARCHAR);

    cs.execute();
    con.close();
}
catch (Exception e)
{
    /* exception handling logic goes here */
}

```

14.6 Fehler und Zustandsverarbeitung

In DB2 werden die SQLCODE und SQLSTATE Schlüsselwörter verwendet um den Erfolg oder Misserfolg der Ausführung einer SQL Anweisung zu bestimmen. Diese Schlüsselwörter müssen explizit im äußeren Bereich der Prozedur wie folgt deklariert werden:

```

DECLARE SQLSTATE CHAR(5);
DECLARE SQLCODE INT;

```

DB2 wird die Werte der obigen Schlüsselworte automatisch nach jeder SQL Operation setzen. Bezüglich des SQLCODE, werden die Werte wie folgt gesetzt:

- = 0, erfolgreich
- > 0, erfolgreich mit Warnung
- < 0, nicht erfolgreich
- = 100, keine Daten gefunden (z.B.: FETCH Anweisung gab keine Daten zurück)

Bezüglich des SQLSTATE, werden die Werte wie folgt gesetzt:

- Erfolg: SQLSTATE '00000'
- Nicht gefunden: SQLSTATE '02000'
- Warnung: SQLSTATE '01XXX'
- Ausnahme: alle anderen Werte

Der SQLCODE ist spezifisch für ein RDBMS und detaillierter als der SQLSTATE. Der SQLSTATE ist Standard bei allen RDBMS, jedoch von Natur sehr allgemein. Verschiedene SQLCODEs können mit einem SQLSTATE übereinstimmen. SQLCODEs und SQLSTATEs werden ausführlicher in Kapitel 18, Problemlösung, besprochen.

Ein Zustand kann von jeder SQL Anweisung ausgelöst werden und stimmt mit einem SQLSTATE überein. So wird zum Beispiel ein bestimmter Zustand, wie SQLSTATE '01004', ausgelöst wenn ein Wert während einer SQL Operation abgeschnitten wird. Statt SQLSTATE '01004' für den Test dieses Zustands zu verwenden, können Namen zugewiesen werden. In diesem besonderen Beispiel kann der Name "trunc" dem Zustand SQLSTATE '01004', wie unten gezeigt, zugewiesen werden.

```
DECLARE trunc CONDITION FOR SQLSTATE '01004'
```

Andere vordefinierte allgemeine Zustände sind:

- SQLWARNING
- SQLEXCEPTION
- NOT FOUND

Zustandsverarbeitung (Condition handling)

Um einen Zustand zu verarbeiten können Sie einen Condition Handler erstellen, welcher spezifizieren muss:

- welche Zustände er behandelt
- wo die Ausführung fortgesetzt wird (basierend auf die Handler des Typs: CONTINUE, EXIT oder UNDO)
- die zur Behandlung des Zustands auszuführenden Aktionen. Bei diesen Aktionen kann es sich um irgendeinen Befehl, einschließlich Kontrollstrukturen, handeln.

Falls ein SQLEXCEPTION Zustand ausgelöst wird und es keinen Handler gibt, wird die Prozedur unter Zurückgabe eines Fehlers an den Client abgebrochen.

Arten von Handlern

Es gibt drei Arten von Handlern:

CONTINUE – Dieser Handler wird verwendet um anzugeben, das nach dem Auslösen einer Ausnahme der Handler die Ausnahme behandelt und mit der auf Fehler folgenden Anweisung fortgesetzt (CONTINUE) wird.

EXIT – Dieser Handler wird verwendet um anzugeben, das nach dem Auslösen einer Ausnahme der Handler die Ausnahme behandelt um zum Ende der Prozedur gesprungen wird.

UNDO – Dieser Handler wird verwendet um anzugeben, dass nach dem Auslösen einer Ausnahme der Handler die Ausnahme behandelt, zum Ende der Prozedur springt und alle ausgeführten Anweisungen rückgängig macht.

Abbildung 14.8 illustriert die unterschiedlichen Condition Handler und deren Verhalten.

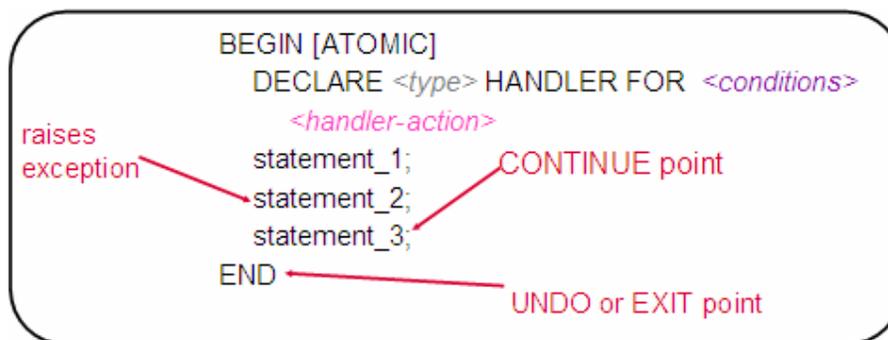


Abbildung 14.8 – Arten von Condition Handlern

14.7 Dynamisches SQL

Bei dynamischem SQL ist, im Gegensatz zu statischem SQL, die gesamte SQL Anweisung zur Laufzeit nicht bekannt. Wenn zum Beispiel `col1` und `tablename` Variablen in dieser Anweisung sind, haben wir es mit dynamischem SQL zu tun:

```
'SELECT ' || col1 || ' FROM ' || tablename;
```

Dynamisches SQL wird, zur Vermeidung von Abhängigkeitsproblemen und Paketungültigkeiten, auch für DDL empfohlen. Es ist ebenfalls für die Implementierung von Rekursion erforderlich.

Dynamisches SQL kann durch Verwendung zweier Vorgehensweisen ausgeführt werden:

- Verwendung der EXECUTE IMMEDIATE Anweisung – dies ist ideal für einmalig ausgeführtes SQL
- Verwendung der PREPARE Anweisung zusammen mit der EXECUTE Anweisung - ideal für mehrfach ausgeführtes SQL

Der folgende Codeausschnitt gibt ein Beispiel von dynamischem SQL unter Verwendung der zwei Vorgehensweisen. Das Beispiel geht davon aus, dass die Tabelle T2 mit der nachfolgenden Definition erstellt wurde:

```
CREATE TABLE T2 (c1 INT, c2 INT)

CREATE PROCEDURE dyn1 (IN value1 INT, IN value2 INT)
SPECIFIC dyn1
BEGIN
DECLARE stmt varchar(255);
DECLARE st STATEMENT;

SET stmt = 'INSERT INTO T2 VALUES (?, ?)';

PREPARE st FROM stmt;

EXECUTE st USING value1, value1;
EXECUTE st USING value2, value2;

SET stmt = 'INSERT INTO T2 VALUES (9,9)';
EXECUTE IMMEDIATE stmt;
END
```

15

Kapitel 15 – Inline SQL PL, Trigger und benutzerdefinierte Funktionen

In diesem Kapitel behandeln wir Inline SQL PL und andere Datenbankanwendungsobjekte, wie benutzerdefinierte Funktionen (UDFs) und Trigger.

Hinweis:

Für weitere Informationen über Trigger und UFDs können Sie sich diese Videos ansehen:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4367>

<http://www.channeldb2.com/video/video/show?id=807741:Video:4362>

15.1 Inline SQL PL

In Kapitel 14 besprachen wir die Erstellung von gespeicherten Prozeduren unter Verwendung der SQL PL Sprache. Die SQL PL Sprache kann auch "inline" verwendet werden, d.h ohne das Erfordernis eine gespeicherte Prozedur zu erstellen. Das in benutzerdefinierten Funktionen und Triggern verwendete SQL PL ist ebenfalls Inline, da der Code inline innerhalb des UDF/Trigger Codes hinzugefügt wird und von Natur dynamisch ist. Inline SQL PL unterstützt nur eine Untermenge aller SQL Anweisungen. Nachfolgend von Inline SQL PL unterstützte Schlüsselwörter:

```
DECLARE <variable>
SET
CASE
FOR
GET DIAGNOSTICS
GOTO
IF
RETURN
SIGNAL
WHILE
ITERATE
LEAVE
```

Nachfolgend die von Inline SQL PL nicht unterstützten Schlüsselwörter:

```
ALLOCATE CURSOR
ASSOCIATE LOCATORS
DECLARE <cursor>
DECLARE ...HANDLER
PREPARE
```

```
EXECUTE
EXECUTE IMMEDIATE
LOOP
REPEAT
RESIGNAL
CALL
COMMIT/ROLLBACK
```

Nachfolgend ein Beispiel von zusammengesetztem dynamischem SQL unter Verwendung von Inline SQL PL. Wenn Sie es ausprobieren möchten integrieren sie das Beispiel in eine Skriptdatei und stellen sicher dass die folgenden Tabellen existieren:

```
CREATE TABLE T1 (c1 INT)
CREATE TABLE T3 (c1 INT)

BEGIN ATOMIC
  DECLARE cnt          INT DEFAULT 0;
  DECLARE sumevens    INT DEFAULT 0;
  DECLARE err_msg     VARCHAR(1000) DEFAULT '';
  WHILE (cnt < 100) DO
    IF mod(cnt,2) = 0 THEN
      SET sumevens = sumevens + cnt;
    END IF;
    SET cnt=cnt+1;
  END WHILE;
  INSERT INTO T3 values (sumevens);
  SET cnt = (SELECT 0 FROM SYSIBM.SYSDUMMY1);
  FOR curl AS SELECT * FROM T1 DO
    IF curl.c1 > 100 THEN
      SET cnt = cnt + 1;
    END IF;
  END FOR;

  SET err_msg = 'Rows with values > 100 is:' ||
char(cnt);
  SIGNAL SQLSTATE '80000' SET MESSAGE_TEXT = err_msg;
END!
```

Wenn Sie das obige Inline SQL PL in eine Skriptdatei mit dem Namen "myScript.txt" speichern können sie es wie folgt ausführen:

```
db2 -td! -vf myScript.txt
```

15.2 Trigger

Trigger sind mit einer Tabelle verbundene Datenbankobjekte, welche nach einer auf eine Tabelle angewandten INSERT, UPDATE oder DELETE, stattfindende Aktionen definieren. Sie werden automatisch aktiviert (oder "ausgelöst"). Die einen Trigger auslösenden Operationen werden *auslösende* (triggering) SQL Anweisungen genannt.

15.2.1 Arten von Triggern

Wir unterscheiden drei Arten von Triggern: "Vor" Trigger, "Nach" Trigger und "Anstatt" Trigger.

"Vor" Trigger

Vor Trigger werden aktiviert bevor eine Zeile eingefügt, aktualisiert oder gelöscht wird. Die von diesem Trigger ausgelösten Aktionen können keine anderen Trigger aktivieren (so sind INSERT, UPDATE und DELETE Operationen nicht erlaubt)

Ein Beispiel eines einfachen Vor Triggers sehen Sie in Abbildung 15.1.

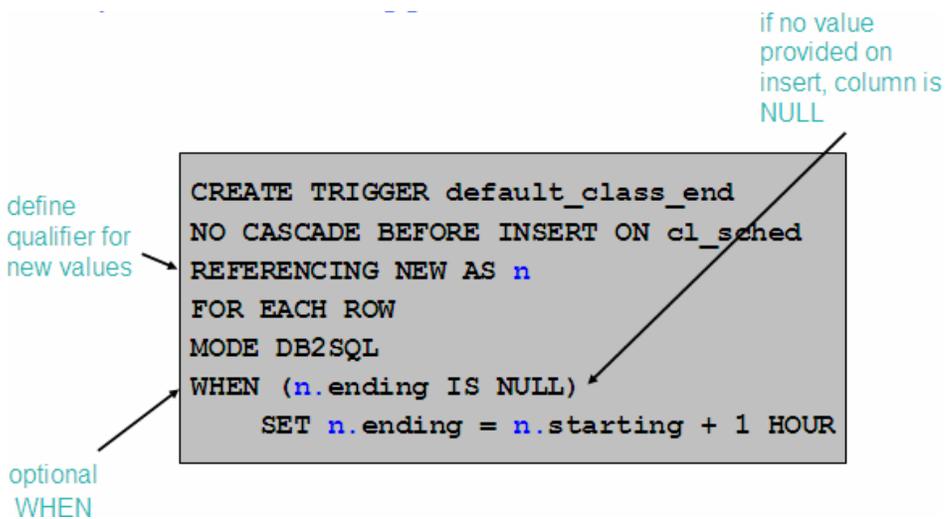


Abbildung 15.1 – Beispiel eines Vor Triggers

In Abbildung 15.1 wird der Trigger "default_class_end" vor einer INSERT SQL Anweisung auf die Tabelle cl_sched ausgelöst. Diese Tabelle ist Teil der SAMPLE Datenbank und Sie können, wenn sie sich mit der Datenbank verbinden, diesen Trigger selber testen. Die Variable "n" in der Trigger Definition stellt den neuen Wert in einem INSERT dar, d.h. der einzufügende Wert. Der Trigger führt eine Gültigkeitsprüfung der in die Tabelle einzufügenden Daten durch. Falls die Spalte "ending" während des Einfügens keinen Wert besitzt stellt der Trigger sicher, dass sie den Wert der Spalte starting plus 1 Stunde erhält.

Die folgenden Anweisungen zeigen Ihnen wie der Trigger getestet wird.

```
C:\Program Files\IBM\SQLLIB\BIN>db2 insert into cl_sched
(class_code, day, starting) values ('abc',1,current time)
DB20000I The SQL command completed successfully.

C:\Program Files\IBM\SQLLIB\BIN>db2 select * from cl_sched

CLASS_CODE DAY    STARTING ENDING
```

```

-----
042:BF          4 12:10:00 14:00:00
553:MJA          1 10:30:00 11:00:00
543:CWM          3 09:10:00 10:30:00
778:RES          2 12:10:00 14:00:00
044:HD          3 17:12:30 18:00:00
abc             1 11:06:53 12:06:53
    
```

```
6 record(s) selected.
```

Der untenstehende "validate_sched" Trigger erweitert die Funktionalität des zuvor beschriebenen "default_class_end" Triggers. Sie können diesen erstellen und gegen die SAMPLE Datenbank testen.

```

CREATE TRIGGER validate_sched
NO CASCADE BEFORE INSERT ON cl_sched
REFERENCING NEW AS n
FOR EACH ROW
MODE DB2SQL
BEGIN ATOMIC
-- supply default value for ending time if null
IF (n.ending IS NULL) THEN
    SET n.ending = n.starting + 1 HOUR;
END IF;

-- ensure that class does not end beyond 9pm
IF (n.ending > '21:00') THEN
    SIGNAL SQLSTATE '80000'
    SET MESSAGE_TEXT='class ending time is beyond 9pm';
ELSEIF (n.DAY=1 or n.DAY=7) THEN
    SIGNAL SQLSTATE '80001'
    SET MESSAGE_TEXT='class cannot be scheduled on a
weekend';
END IF;
END
    
```

"Nach" Trigger

Nach Trigger werden nach der erfolgreichen Ausführung der auslösenden SQL Anweisung aktiviert. Die von diesem Trigger durchgeführten Operationen können andere Trigger aktivieren (Kaskadierung bis zu 16 Ebenen ist erlaubt). Nach Trigger unterstützen INSERT, UPDATE und DELETE Operationen. Unten sehen Sie ein Beispiel eines Nach Trigger.

```

CREATE TRIGGER audit_emp_sal
AFTER UPDATE OF salary ON employee
REFERENCING OLD AS o NEW AS n
FOR EACH ROW
MODE DB2SQL
INSERT INTO audit VALUES (
    CURRENT_TIMESTAMP, ' Employee ' || o.empno || '
salary changed from ' || CHAR(o.salary) || ' to ' ||
    
```

```
CHAR(n.salary) || ' by ' || USER)
```

In diesem Beispiel wird der Trigger `audit_emp_sal` für die Revision der Spalte "salary" der "employee" Tabelle verwendet. Wenn jemand den Inhalt dieser Spalte ändert, wird der Trigger ausgelöst um Informationen über die an der salary vorgenommenen Änderungen in die Tabelle "audit" zu schreiben. Die "OLD as o NEW as n" Zeile besagt, das das Präfix "o" für die Darstellung des alten oder existierenden Wertes in der Tabelle verwendet wird und das Präfix "n" wird verwendet um den neuen, vom UPDATE herrührenden, Wert darzustellen. Deshalb stellt "o.salary" den alten oder existierenden Wert der salary und "n.salary" den aktualisierten Wert der Spalte salary dar.

"Anstatt" Trigger

Anstatt Trigger werden auf Sichten basierend definiert. Die im Trigger definierte Logik wird statt der auslösenden SQL Anweisung ausgeführt. Wenn Sie zum Beispiel eine Aktualisierung (UPDATE) auf eine Sicht vornehmen, wird der Anstatt Trigger ausgelöst um die Aktualisierung der Basistabellen der Sicht auszuführen.

Trigger können nicht mit dem IBM Data Studio erstellt werden. Sie können über die Steuerzentrale oder Befehlszeilentools (Befehlsfenster, Befehlszeilenprozessor oder Befehlseditor) erstellt werden.

Übung 12 – Erstellen eines Triggers über die Steuerzentrale

Ziel

Bei Triggern handelt es sich um Datenbankobjekte welche bei der Datenänderung zur Ausführung von Geschäftslogik verwendet werden. In dieser Übung werden Sie einen Trigger über die Steuerzentrale erstellen. Dieser Trigger protokolliert für Revisionszwecke alle an der Tabelle SALES vorgenommenen Änderungen. Sie werden die Benutzerkennung des die Änderung vornehmenden Benutzers und die Uhrzeit, an welcher die Änderung vorgenommen wurde, protokollieren.

Vorgehensweise

1. Öffnen Sie die Steuerzentrale.
2. Für diese Übung müssen Sie eine zusätzliche, für die Protokollierung benötigte, Tabelle erstellen. Erstellen Sie eine Tabelle mit den folgenden Eigenschaften:

Tabellenname: `saleslog`

Erste Spalte:

Name: `userid`
Datentyp: `VARCHAR(128)`
Andere Attribute: `NOT NULL`

Zweite Spalte:

Name: `daytime`
Datentyp: `TIMESTAMP`
Andere Attribute: `NOT NULL`

Hinweis: Erstellen Sie die Tabelle mit der `CREATE TABLE` Anweisung im Befehlseditor oder verwenden sie den *Tabelle erstellen* Assistenten.

3. In der Steuerzentrale erweitern Sie den *EXPRESS* Datenbankordner. Klicken rechts auf den *Trigger* Ordner und wählen die *Erstellen* Option. Das *Trigger erstellen* Dialogfenster wird geöffnet.
4. Im Dialogfenster geben Sie folgende Informationen ein:

Triggerschema: Benutzerkennung mit welcher Sie sich angemeldet haben (sollte die Vorgabe sein).

Triggername: *audit_sales*

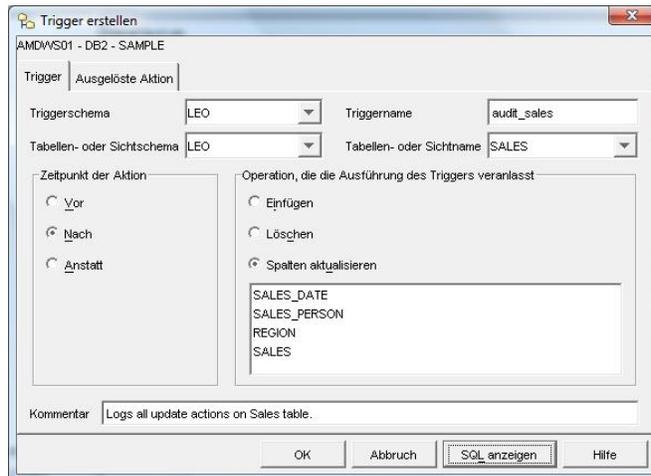
Tabellen- oder Sichtschemata: Benutzerkennung mit welcher Sie sich angemeldet haben (sollte die Vorgabe sein).

Tabellen- oder Sichtname: *SALES*

Zeitpunkt der Aktion: *After*

Operation, welche die Ausführung des Triggers veranlasst: *Spalten aktualisieren* (geben Sie keine Spalten an, da wir wollen das der Trigger bei der Aktualisierung einer der Spalten auslöst).

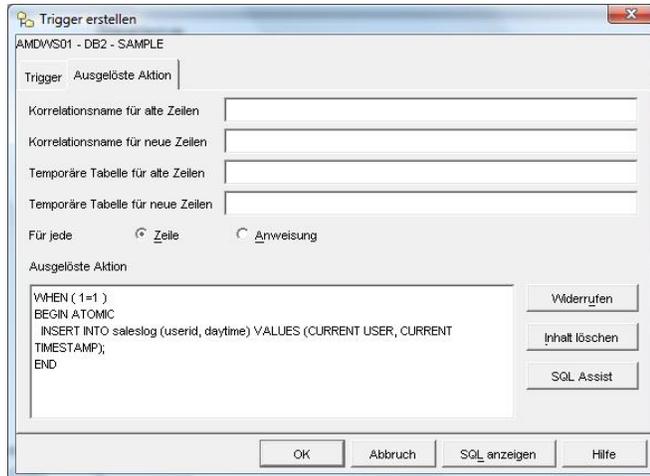
Kommentar: *Logs all update actions on Sales table.*



5. Auf der *Ausgelöste Aktion* Registerkarte wählen Sie die *Für jede Anweisung* Option. Verwenden Sie den folgenden Code für die ausgelöste Aktion:

```
WHEN ( 1=1 )
BEGIN ATOMIC
  INSERT INTO saleslog (userid, daytime) VALUES (CURRENT
USER, CURRENT TIMESTAMP);
END
```

(Hinweis: Ein Statement Trigger löst aus, nachdem die den Trigger auslösende Anweisung beendet wurde. Ein Row Trigger gibt an, dass die ausgelöste Aktion bei jeder eine Zeilenänderung auslösenden SQL Anweisung durchgeführt wird.)



Klicken Sie auf die **OK** Schaltfläche um den Trigger zu erstellen.

6. Sie sollten jetzt in der Lage sein den Trigger im *Trigger Ordner* der Steuerzentrale zu sehen.
7. Fragen Sie, um sicherzustellen dass diese keine Daten enthält, die *saleslog* Tabelle ab. Löschen Sie alle möglicherweise darin enthaltenen Zeilen (`DELETE FROM saleslog`).
8. Versuchen Sie eine Zeile der *sales* Tabelle zu aktualisieren. (Hinweis: Verwenden Sie den Befehlseditor oder den SQL Assist Assistenten).
9. Prüfen Sie den Inhalt der *saleslog* Tabelle erneut. Wie viele Zeilen enthält die Tabelle?

15.3 Benutzerdefinierte Funktionen (UDFs)

Bei einer benutzerdefinierten Funktion (UDF) handelt es sich um ein Datenbankanwendungsobjekt, welches einen Satz Eingabedatenwerte in einem Satz Ausgabedatenwerte abbildet. Eine Funktion könnte zum Beispiel als Eingabe Inches verwenden und als Resultat Zentimeter zurückgeben.

DB2 unterstützt die Erstellung von Funktionen mit SQL PL, C/C++, Java, CLR (Common Language Run Time) und OLE (Objekt Linking und Embedding). In diesem Buch konzentrieren wir uns auf SQL Funktionen, da diese einfach, populär und leistungsstark sind.

Es gibt vier Funktionstypen: Skalar-, Tabelle-, Zeilen- und Spaltenfunktionen. In diesem Kapitel behandeln wir ausschließlich Skalar- und Tabellenfunktionen.

15.3.1 Skalarfunktionen

Skalarfunktionen geben einen einzigen Wert zurück. Skalarfunktionen dürfen keine den Datenbankstatus ändernden SQL Anweisungen, d.h. INSERT, UPDATE und DELETE, enthalten. Einige eingebaute Skalarfunktionen sind SUM(), AVG(), DIGITS(), COALESCE() und SUBSTR().

DB2 erlaubt Ihnen maßgeschneiderte benutzerdefinierte Funktionen, in welche oft verwendete Logik eingekapselt ist, zu erstellen. Betrachten Sie zum Beispiel die Migration ihrer Anwendung von Oracle nach DB2. In Ihrer Anwendung rufen sie oft die Oracle Funktion NVL() auf. Die in DB2 eingebaute vergleichbare Funktion heißt COALESCE. Statt alle NVL Funktionen in Ihrer Anwendung in COALESCE umzubenennen, könnten Sie eine benutzerdefinierte Funktion in DB2 erstellen, diese NVL nennen und die COALESCE Funktion, wie unten gezeigt, aufrufen.

```
CREATE FUNCTION NVL (p_var1 VARCHAR(30),
                   p_var2 VARCHAR(30))
SPECIFIC nvlvarchar30
RETURNS VARCHAR(30)
RETURN COALESCE(p_var1, p_var2)
```

Die COALESCE Funktion gibt dann das erste Argument ungleich Null zurück.

Unten ist ein weiteres Beispiel einer Skalarfunktion. Die Funktion heißt "deptname" und gibt die Abteilungsnummer (department number) eines Angestellten (employee) basierend auf die Angestellten ID (employee id) zurück.

```
CREATE FUNCTION deptname(p_empid VARCHAR(6))
RETURNS VARCHAR(30)
SPECIFIC deptname
BEGIN ATOMIC
  DECLARE v_department_name VARCHAR(30);
  DECLARE v_err VARCHAR(70);
```

```

SET v_department_name = (
    SELECT d.deptname FROM department d, employee e
        WHERE e.workdept=d.deptno AND e.empno= p_empid);
SET v_err = 'Error: employee ' || p_empid || ' was not
found';
IF v_department_name IS NULL THEN
    SIGNAL SQLSTATE '80000' SET MESSAGE_TEXT=v_err;
END IF;
RETURN v_department_name;
END

```

Um die Funktion zu testen, versuchen Sie folgendes über das Befehlsfenster oder die Linux/UNIX Shell auszuführen:

```
db2 "values (deptname ('000300'))"
```

Der Aufruf von Skalarfunktionen

Skalarfunktionen können in einer einen Skalarwert erwartenden SQL Anweisung oder in einer VALUES Klausel aufgerufen werden. Nachfolgend zwei Beispiele wie die COALESCE Funktion aufgerufen werden kann:

```

SELECT DEPTNAME, COALESCE(MGRNO, 'ABSENT') FROM DEPARTMENT
VALUES COALESCE('A', 'B')

```

15.3.2 Tabellenfunktionen

Tabellenfunktionen liefern eine Menge von Tupeln (Zeilen einer Tabelle) als Ergebnis. Sie können Tabellenfunktionen in der FROM Klausel einer Abfrage aufrufen. Im Gegensatz zu Skalarfunktionen können Tabellenfunktionen den Datenbankstatus ändern, d.h. INSERT, UPDATE und DELETE Anweisungen sind erlaubt. Einige eingebaute Tabellenfunktionen sind SNAPSHOT_DYN_SQL() und MQREADALL(). Tabellenfunktionen sind ähnlich wie Sichten, da sie jedoch Anweisungen zur Datenänderung (INSERT, UPDATE und DELETE) unterstützen viel leistungsfähiger. Tabellenfunktionen werden normalerweise verwendet um eine Tabelle zurückzugeben und einen Datensatz für Revisionszwecke zu speichern.

Unten ist ein Beispiel einer einen Satz von Angestellten einer Abteilung (workdept) nummerierenden Tabellenfunktion:

```

CREATE FUNCTION getEnumEmployee(p_dept VARCHAR(3))
RETURNS TABLE
    (empno CHAR(6),
     lastname VARCHAR(15),
     firstnme VARCHAR(12))
SPECIFIC getEnumEmployee
RETURN
    SELECT e.empno, e.lastname, e.firstnme
    FROM employee e
    WHERE e.workdept=p_dept

```

Um die obige Funktion zu testen versuchen Sie:

```
db2 "SELECT * FROM table(getEnumEmployee('D11')) AS t"
```

Der Aufruf von Tabellenfunktionen

Eine Tabellenfunktion muss in der FROM Klausel einer SQL Anweisung aufgerufen werden. Die TABLE() Funktion muss angewandt und mit einem Alias versehen werden. Abbildung 15.2 zeigt ein Beispiel wie die bereits zuvor von uns getestete Funktion "getEnumEmployee" aufgerufen wird.

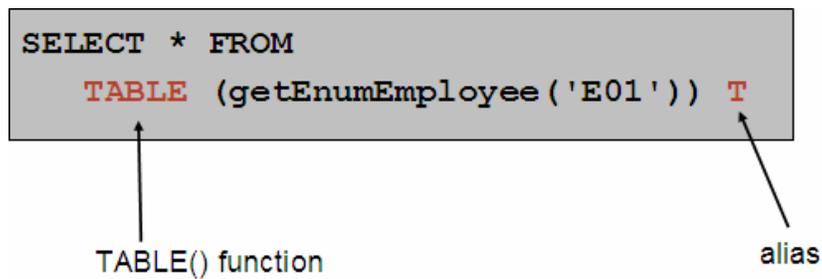


Abbildung 15.2 – Eine Tabellenfunktion aufrufen.

Übung 13 – Erstellen einer benutzerdefinierten Funktion (UDF) mit dem IBM Data Studio

Ziel

In dieser Übung werden Sie mit dem IBM Data Studio eine benutzerdefinierte Skalarfunktion erstellen. Hierdurch bekommen Sie mehr Erfahrung im Umgang mit dem Data Studio und verbessern außerdem ihre Kenntnisse der SQL PL Sprache für benutzerdefinierte Funktionen.

Vorgehensweise

1. Öffnen Sie das IBM Data Studio (Hinweis: Es ist über das Start Menü erreichbar).
2. Im Datenprojektexplorer Fenster markieren Sie das in der vorherigen Übung erstellte Projekt und wählen *Projekt öffnen*.
3. Klicken Sie rechts auf den Ordner *Benutzerdefinierte Funktionen*. Wählen Sie den Menüpunkt *Neu*. Wählen Sie den Menüpunkt *Benutzerdefinierte Funktion*. Im Assistent *Benutzerdefinierte Funktion* geben Sie folgende Werte ein:

Projekt: Stellen Sie sicher, dass es sich um einen bereits zuvor verwendeten Projektnamen handelt

Name: FUNCTION1

Sprache: SQL

4. Die *Editor* Sicht sollte mit einem wie folgt aussehenden Funktionsskelett geöffnet werden:

```
CREATE FUNCTION FUNCTION1( )
  RETURNS INTEGER
  NO EXTERNAL ACTION
-----
-- SQL UDF (Scalar)
-----
F1: BEGIN ATOMIC
      RETURN SELECT count(*) FROM SYSCAT.FUNCTIONS;
END
```

Ändern Sie den obigen Code wie folgt:

```
CREATE FUNCTION booktitle(p_bid INTEGER)
  RETURNS VARCHAR(300)
-----
SQL UDF (Scalar)
-----
SPECIFIC booktitle
```

```

F1: BEGIN ATOMIC
DECLARE v_book_title VARCHAR(300);
DECLARE v_err VARCHAR(70);
SET v_book_title = (SELECT title FROM books WHERE p_bid = book_id);
SET v_err = 'Error: The book with ID ' || CHAR(p_bid) || '
was not found.';
IF v_book_title IS NULL THEN SIGNAL SQLSTATE '80000' SET MESSAGE_TEXT=v_err;
END IF;
RETURN v_book_title;
END

```

5. Erstellen Sie die Funktion indem sie rechts auf diese klicken und *Implementieren* wählen. Falls ein zur Speicherung der Änderungen auffordernden Dialog angezeigt wird klicken Sie auf *Ja*. Im darauffolgenden Bildschirm können Sie Implementierungsoptionen angeben. Belassen Sie die Standardwerte und klicken auf *Fertig stellen*.
6. Starten Sie die Funktion durch einen Rechtsklick auf die Funktion und anschließender Auswahl von *Ausführen*.
7. Da die Funktion einen Eingabeparameter übernimmt, fordert Sie ein Dialogfenster zur Eingabe des Parameters auf.

Geben Sie diesen Wert ein: 80002

Was ist das Resultat?

Versuchen Sie es erneut mit diesem Wert: 1002

Was passiert diesmal? (Hinweis: Sehen Sie im *Nachrichten* Abschnitt des *Ausgabefensters* nach).

8. Beenden Sie das IBM Data Studio wenn sie fertig sind.

16

Kapitel 16 – DB2 pureXML

In diesem Kapitel behandeln wir pureXML, die neue, beginnend mit DB2 9 verfügbar gemachte, Technologie zur Unterstützung nativer XML Speicherung. Viele der in diesem Kapitel diskutierten Beispiel und Konzepte entstammen dem IBM Redbook *DB2 9: pureXML overview and fast start*. Im Abschnitt Ressourcen finden Sie weitere Informationen. Abbildung 16.1 skizziert, welcher Abschnitt des DB2 "Gesamtbilds" wir in diesem Kapitel diskutieren.

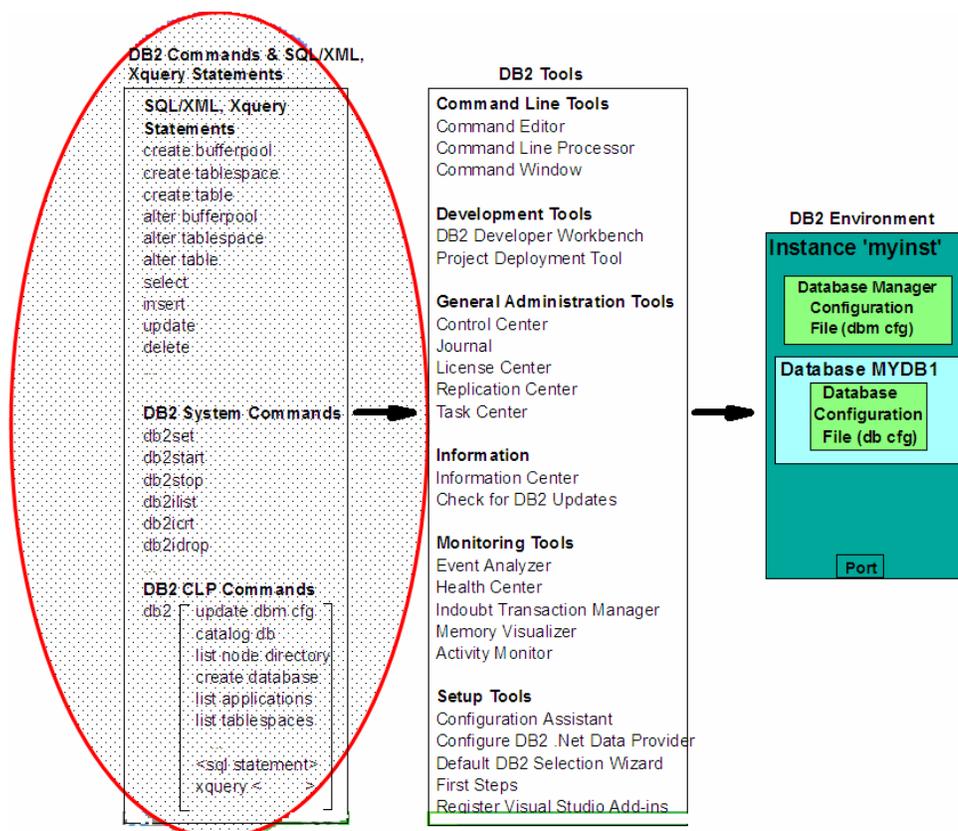


Abbildung 16.1 – Das DB2 Gesamtbild: DB2 Befehle, SQL/XML und XQuery**Hinweis:**

Für weitere Informationen über pureXML können Sie sich dieses Video ansehen:
<http://www.channeldb2.com/video/video/show?id=807741:Video:4382>

16.1 XML mit Datenbanken verwenden

XML kann in Textdateien, XML Repositorien oder Datenbanken gespeichert werden. Es gibt zwei Gründe, warum viele Firmen vorschlagen diese in Datenbanken zu speichern:

- Die Verwaltung großer Mengen XML Daten ist ein Datenbankproblem. Bei XML handelt es sich um Daten wie andere Daten, nur in einem unterschiedlichen umfassenden Format. Dieselben Kriterien zur Speicherung relationaler Daten in Datenbanken gelten auch für XML: Datenbanken bieten effiziente Suche und Datenbeschaffung, robuste Unterstützung der Datenpersistenz, Sicherung und Wiederherstellung, Unterstützung von Transaktionen, Leistung und Skalierbarkeit.
- Integration: durch das gemeinsame Speichern relationaler und XML Daten können Sie neue XML Daten in existierende relationale Daten integrieren und SQL mit XPath oder XQuery in einer Abfrage kombinieren und umgekehrt. Durch Integration können Datenbanken Webanwendungen, SOA und Webservices besser unterstützen.

16.2 XML Datenbanken

Es gibt zwei Arten von Datenbanken zur Speicherung von XML Daten:

- XML-aktivierte Datenbanken
- Native XML Datenbanken

16.2.1 XML-aktivierte Datenbanken

Eine XML-aktivierte Datenbank verwendet ein relationales Modell für das Kerndatenspeichermodell. Dies erfordert das Mapping zwischen dem XML (hierarchisch) Datenmodell und dem relationalen Datenmodell oder andernfalls die Speicherung der XML Daten als Character Large Object (CLOB). Obwohl dies als eine "alte" Technologie angesehen werden kann, wird es immer noch von vielen Datenbank Anbietern verwendet. Abbildung 16.2 erklärt die zwei Optionen für XML-aktivierte Datenbanken ausführlicher.

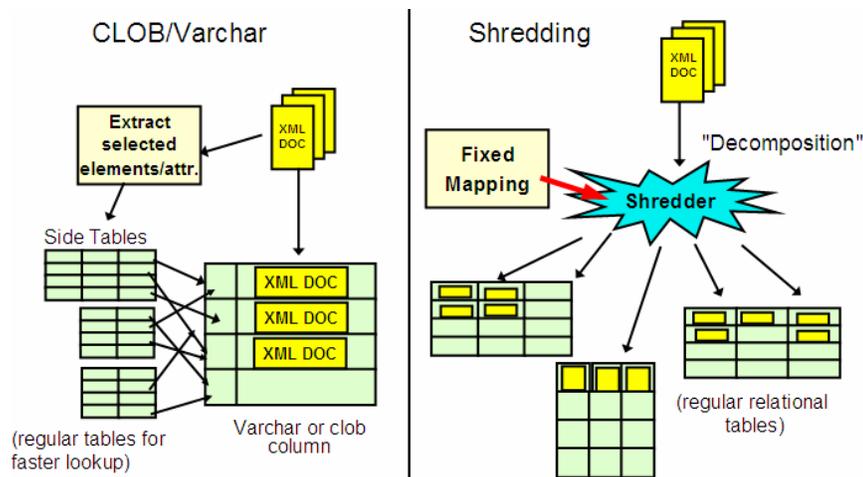


Abbildung 16.2 – Zwei Möglichkeiten zur Speicherung von XML in XML-aktivierten Datenbanken

Die linke Seite der Abbildung 16.2 zeigt die "CLOB und varchar" Methoden der Speicherung von XML Dokumenten in einer Datenbank. Bei Verwendung dieser Methode wird ein XML Dokument als nicht geparte Zeichenkette in entweder einer CLOB oder varchar Variablen in der Datenbank gespeichert. Falls das XML Dokument als Zeichenkette gespeichert wird muss, falls Sie einen Teil des Dokuments abrufen wollen, das Programm die Zeichenkette abrufen und das Parsen ausführen, damit sie finden was sie suchen. Diese Methode ist nicht sehr flexibel.

Die andere Option für XML-aktivierte Datenbanken nennt sich Schreddern (shredding) oder Dekomposition (decomposition) und wird auf der rechten Seite der Abbildung 16.2 illustriert. Bei Verwendung dieser Methode wird ein vollständiges XML Dokument in kleinere Teile zerlegt, welche in Tabellen gespeichert werden. Bei Verwendung dieser Methode wird das hierarchische Modell in das relationale Modell gezwungen. Dies ist ebenfalls nicht gut für die Flexibilität: eine Änderung im XML Dokument kann nicht einfach auf die betroffenen Tabellen angewandt werden und viele weitere Tabellen müssen eventuell erstellt werden. Diese Methode ist ebenfalls nicht gut für die Performance: wenn Sie wieder das originale XML Dokument benötigen müssen sie aufwändige SQL Operationen ausführen, was umso aufwändiger wird, je mehr Tabellen verbunden sind.

16.2.2 Native XML Datenbanken

Native XML Datenbanken verwenden das hierarchische XML Datenmodell um XML intern zu speichern und zu verarbeiten. Das Speicherformat ist identisch mit dem Verarbeitungsformat: es findet kein Mapping in das relationale Modell statt und XML Dokumente werden nicht als Images gespeichert. Wenn XPath oder XQuery Anweisungen verwendet werden, werden diese nativ durch die Engine verarbeitet und nicht nach SQL konvertiert. Aus diesem Grund sind diese Datenbanken bekannt als "native" XML Datenbanken. DB2, beginnend mit Version 9, ist zur Zeit der einzige diese Unterstützung bietende kommerzielle Datenserver.

16.3 XML in DB2

Abbildung 16.3 unten umreißt wie relationale Daten und hierarchische Daten (XML Dokumente) in DB2, beginnend mit Version 9, gespeichert werden. In dieser Abbildung nehmen wir an, dass die Tabelle dept wie folgt definiert ist:

```
CREATE TABLE dept (deptID CHAR(8),..., deptdoc XML);
```

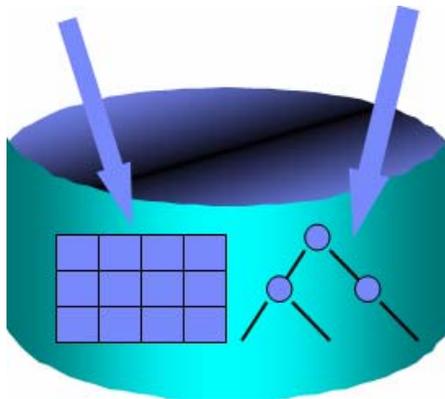


Abbildung 16.3 – XML in DB2

Bitte beachten Sie, dass die Tabellendefinition einen neuen Datentyp, XML, für die deptdoc Spalte verwendet.

Der linke Pfeil in der Abbildung zeigt an, dass die relationale Spalte deptID im relationalen Format (Tabellen) gespeichert wird, wohingegen die XML Spalte deptdoc im geparteten hierarchischen Format gespeichert wird.

Abbildung 16.4 illustriert dass es, beginnend mit DB2 9, nunmehr vier Wege des Datenzugriffs gibt:

- Verwenden von SQL zum Zugriff auf relationale Daten
- Verwenden von SQL mit XML Erweiterungen zum Zugriff auf XML Daten
- Verwenden von XQuery zum Zugriff auf XML Daten
- Verwenden von XQuery zum Zugriff auf relationale Daten

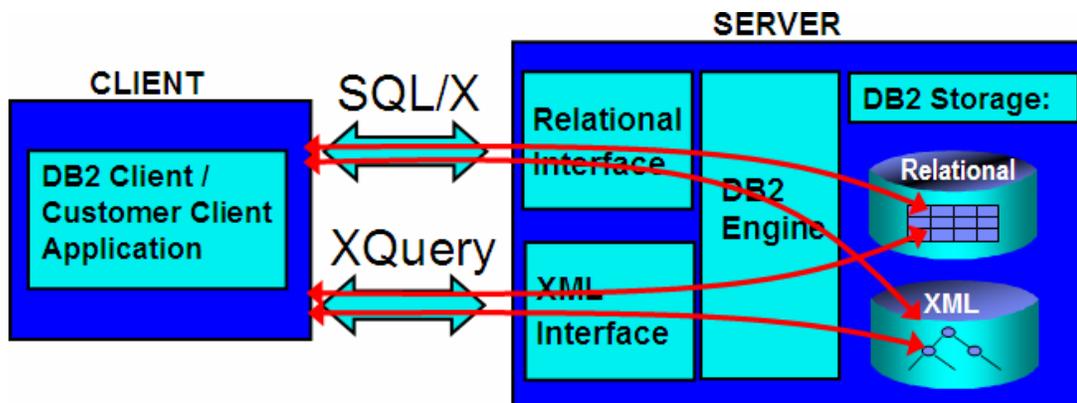


Abbildung 16.4 – Vier Wege des DB2 Datenzugriffs

Deshalb, während eine SQL Person DB2 vielleicht als ein RDBMS von Weltklasse betrachtet das zusätzlich XML unterstützt, könnte eine XML Person DB2 als XML Repositorium von Weltklasse betrachten das zusätzlich SQL unterstützt.

Bitte beachten Sie, dass IBM den Begriff pureXML statt "native XML" zur Beschreibung dieser Technologie verwendet. Während andere Anbieter immer noch die alten Technologien des CLOB/varchar oder Schredderns zur Speicherung von XML Dokumenten verwenden, nennen sie diese alten Technologien "native XML". Um Verwirrung zu vermeiden entschied IBM den neuen Begriff pureXML zu verwenden und diesen als Warenzeichen einzutragen, damit keine andere Datenbank oder XML Anbieter denselben Namen zur Kennzeichnung einer anderen Technologie verwenden kann. In DB2 9.1 wurde pureXML Unterstützung nur geboten wenn die Datenbank als eine Unicode verwendende Datenbank erstellt wurde. Diese Einschränkung wurde mit DB2 9.5 aufgehoben, wo Sie pureXML in einer Unicode oder nicht-Unicode Datenbank verwenden können.

16.3.1 pureXML Technologievorteile

Viele Vorteile werden durch die pureXML Technologie geboten.

1. Sie können direkt den Nutzen ihrer schon getätigten Investition in eine relationale Datenbanksoftware steigern, vorausgesetzt das XML Dokumente in Spalten einer Tabelle, unter Verwendung des neuen XML Datentyps, gespeichert werden.
2. Sie können die Codekomplexität reduzieren. Abbildung 16.5 zum Beispiel illustriert ein PHP Skript geschrieben sowohl mit als auch ohne die Verwendung von pureXML. Bei Verwendung von pureXML (der grüne Kasten) werden die Anzahl Codezeilen reduziert. Das bedeutet nicht nur, dass der Code weniger komplex ist, sondern die Gesamtleistung verbessert wird, da weniger Codezeilen zu parsen und verwalten sind.

```

<?php
$conn = db2_connect($dbname, $dbuser, $dbpass);

/* Insert Customer Documents */

$stmt = db2_prepare($conn, "VALUES (NEXT VALUE FOR
Cid)");
db2_execute($stmt);
list($Cid) = db2_fetch_array($stmt);

$fileContents = file_get_contents
("customers/c1.xml");

$stmt = db2_prepare($conn, "INSERT INTO xmlcustomer
(Cid, Info) VALUES (?, ?)");
if(!db2_execute($stmt, array($Cid, $fileContents)))

    echo db2_stmt_errormsg($stmt);

/* Insert Product Documents */

$fileContents = file_get_contents
("products/p1.xml");
$dom = simplexml_load_string($fileContents);

$prodID = (string) $dom["pid"];

$stmt = db2_prepare($conn, "INSERT INTO xmlproduct
(Pid, Description) VALUES (?, ?)");
if(!db2_execute($stmt, array($prodID,

    db2_execute($stmt);
    list($Cid) = db2_fetch_array($stmt);

    $fileContents = file_get_contents
    ("customers/c1.xml");
    $dom = simplexml_load_string($fileContents);

    $custName = (string) $dom->name;
    $custCountry = (string) $dom->addr["country"];
    $custStreet = (string) $dom->addr->street;
    $custCity = (string) $dom->addr->city;
    $custProvince = (string) $dom->addr->("prov-state");
    $custZip = (string) $dom->addr->("pcode-zip");
    $custPhone = (string) $dom->phone;

    $stmt = db2_prepare($conn, "INSERT INTO sqlcustomer
    (Cid, Name, Country, Street, City, Province, Zip,
    Phone, Info) VALUES (?, ?, ?, ?, ?, ?, ?, ?)");
    if(!db2_execute($stmt, array($Cid, $custName,
    $custCountry, $custStreet, $custCity, $custProvince,
    $custZip, $custPhone, $fileContents) )) {
        echo db2_stmt_errormsg($stmt);
    }

/* Insert Product Documents */

$fileContents = file_get_contents
("products/p1.xml");
$dom = simplexml_load_string($fileContents);

$prodID = (string) $dom["pid"];

```

Abbildung 16.5 – Codekomplexität mit und ohne pureXML

3. Änderungen an Ihrem Schema sind bei Verwendung von XML und der pureXML Technologie einfacher. Abbildung 16.6 illustriert ein Beispiel dieser erhöhten Flexibilität. In der Abbildung nehmen Sie an, dass sie eine Datenbank, bestehend aus den Tabellen Employee und Department, haben. Üblicherweise, bei einer nicht-XML Datenbank, falls Ihr Chef sie bitten würde nicht nur eine Telefonnummer je Angestellten (die private Telefonnummer), sondern auch eine zweite Telefonnummer (eine Mobiltelefonnummer) zu speichern, dann könnten sie eine zweite Spalte zur Employee Tabelle hinzufügen und die Mobiltelefonnummer in der neuen Spalte speichern. Wie dem auch sei, diese Vorgehensweise würde gegen die Normalisierungsregeln relationaler Datenbanken verstoßen. Wenn Sie diese Regeln einhalten wollen, sollten sie eine neue Zusatztabelle Phone erstellen, und alle Telefoninformationen in diese Tabelle verschieben. Sie könnten dann ebenfalls die Mobiltelefonnummern hinzufügen. Das Erstellen einer neuen "Phone" Tabelle ist aufwändig, nicht nur da große Mengen existierender Daten bewegt werden müssen, sondern auch da alle SQL Anweisungen in Ihren Anwendungen dahingehend geändert werden müssen, dass sie auf die neue Tabelle zeigen.

Stattdessen zeigen wir Ihnen in der linken Seite der Abbildung wie dies mit XML realisiert werden könnte. Falls die Angestellte "Christine" auch eine Mobiltelefonnummer hat, kann ein neuer Tag für die Aufnahme dieser Information hinzugefügt werden. Wenn der Angestellte "Michael" keine Mobiltelefonnummer besitzt, lassen wir es wie es ist.

```

<DEPARTMENT deptid="15" deptname="Sales">
  <EMPLOYEE>
    <EMPNO>10</EMPNO>
    <FIRSTNAME>CHRISTINE</FIRSTNAME>
    <LASTNAME>SMITH</LASTNAME>
    <PHONE>408-463-4963</PHONE>
    <PHONE>415-010-1234</PHONE>
    <SALARY>52750.00</SALARY>
  </EMPLOYEE>
  <EMPLOYEE>
    <EMPNO>27</EMPNO>
    <FIRSTNAME>MICHAEL</FIRSTNAME>
    <LASTNAME>THOMPSON</LASTNAME>
    <PHONE>406-463-1234</PHONE>
    <SALARY>41250.00</SALARY>
  </EMPLOYEE>
</DEPARTMENT>
  
```

Requires:

- Normalization of existing data !
- Modification of the mapping
- Change of applications

Phone

EMPNO	PHONE
27	406-463-1234
10	415-010-1234
10	408-463-4963

Department

DEPTID	DEPTNAME
15	Sales

Employee

DEPTID	EMPNO	FIRSTNAME	LASTNAME	PHONE	SALARY
15	27	MICHAEL	THOMPSON	406-463-1234	41250
15	10	CHRISTINE	SMITH	408-463-4963	52750

aufwändig!

Abbildung 16.6 – Erhöhte Datenflexibilität bei Verwendung von XML

4. Sie können ihre XML Anwendungsleistung erhöhen. Mit pureXML durchgeführte Tests zeigten erhebliche Leistungsverbesserungen bei mehreren XML Anwendungen. Abbildung 16.7 zeigt die Testergebnisse für eine Firma, welche von älteren Technologien auf XML wechselte. Die mittlere Spalte zeigt die Ergebnisse bei Verwendung der alten Methode der Arbeit mit XML und die rechte Spalte die Ergebnisse bei Verwendung von pureXML.

Aufgabe	Andere relationale DB	DB2 mit pureXML
Entwicklung eines Such- und Datengewinnungsgeschäftsprozesses	CLOB: 8 Std. Shred: 2 Std.	30 Min.
Relative Anzahl von I/O Codezeilen	100	35 (65% Reduktion)
Feld zum Schema hinzufügen	1 Woche	5 Min.
Abfragen	24 - 36 Std.	20 Sek. - 10 Min.

Abbildung 16.7 – Erhöhte Performance bei Verwendung der pureXML Technologie

16.3.2 XPath Grundlagen

XPath ist eine Sprache welche für die Abfrage von XML Dokumenten verwendet werden kann. Abbildung 16.8 zeigt ein XML Dokument und Abbildung 16.9 umschreibt dasselbe Dokument dargestellt im parsed-hierarchical (auch "node" oder "leaf" genannt) Format. Wir werden das parsed-hierarchical Format verwenden um zu erklären wie XPath funktioniert.

```

<dept bldg="101">
  <employee id="901">
    <name>John Doe</name>
    <phone>408 555 1212</phone>
    <office>344</office>
  </employee>
  <employee id="902">
    <name>Peter Pan</name>
    <phone>408 555 9918</phone>
    <office>216</office>
  </employee>
</dept>

```

Abbildung 16.8 – Ein XML Dokument

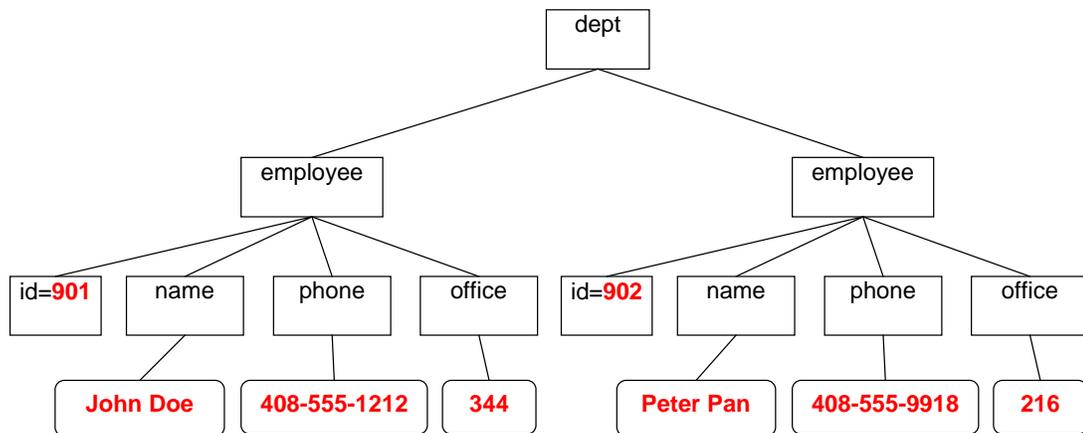


Abbildung 16.9 – XML Dokument aus Abbildung 16.8 in parsed-hierarchical Darstellung

Eine schnelle Möglichkeit XPath zu lernen besteht darin, einen Vergleich mit dem `change directory (cd)` Befehl unter MS-DOS oder Linux/UNIX anzustellen. Mit dem `cd` Befehl durchlaufen Sie ein Verzeichnis wie folgt:

```
cd /directory1/directory2/...
```

Ähnlich bei Xpath, wo Sie Schrägstriche verwenden um von einem Element zu einem anderen Element innerhalb des XML Dokuments zu gehen. Bei Verwendung des Dokuments aus Abbildung 16.9 können Sie zum Beispiel zum Auslesen der Namen aller Angestellten diese Abfrage verwenden:

```
/dept/employee/name
```

XPath Ausdrücke

XPath Ausdrücke (expressions) verwenden voll qualifizierte Pfade für die Spezifizierung der Elemente und/oder Attribute. Das “@” Zeichen wird zur Kennzeichnung eines Attributs verwendet. Um nur den Wert Textknoten (text node) eines Elementes auszulesen wird die “text()” Funktion verwendet. Tabelle 16.1 zeigt XPath Abfragen und die korrespondierenden Ergebnisse unter Verwendung des XML Dokuments aus Abbildung 16.9.

XPath	Ergebnis
/dept/@bldg	101
/dept/employee/@id	901 902
/dept/employee/name	<name>Peter Pan</name> <name>John Doe</name>

/dept/employee/name/text()	Peter Pan John Doe
----------------------------	-----------------------

Tabelle 16.1 – Beispiele einiger XPath Ausdrücke

XPath Maskenzeichen

Es gibt zwei hauptsächliche Maskenzeichen (Wild Cards) in XPath:

- "*" entspricht einem beliebigen Tag Namen
- "/" ist das "untergeordnete Knoten inklusive des Kontextknotens" (descendant-or-self) Maskenzeichen

Table 16.2 gibt weitere Beispiele unter Verwendung des XML Dokuments aus Abbildung 16.9.

XPath	Ergebnis
/dept/employee/*/text()	John Doe 408 555 1212 344 Peter Pan 408 555 9918 216
/dept*/@id	901 902
//name/text()	Peter Pan John Doe
/dept//phone	<phone>408 555 1212</phone> <phone>408 555 9918</phone>

Tabelle 16.2 – Beispiele einiger XPath Maskenzeichen

XPath Prädikate

Prädikate (predicates) sind in eckige Klammern [] eingeschlossen. Als Analogie können Sie diese mit der WHERE Klausel in SQL vergleichen. So kann zum Beispiel [@id="902"] gelesen werden als: "WHERE Attribut id ist gleich 902". Ein XPath Ausdruck kann aus mehreren Prädikaten bestehen. Zur Angabe eines Positionsprädikats verwenden Sie [n], womit das n^{te} Child gewählt würde. So bedeutet employee[2] zum Beispiel dass der zweite employee gewählt werden soll. Tabelle 16.3 liefert weitere Beispiele.

XPath	Ergebnis
/dept/employee[@id="902"]/name	<name>Peter Pan</name>
/dept[@bldg="101"]/employee[office >"300"]/name	<name>John Doe</name>
//employee[office="344" OR office="216"]/@id	901 902

/dept/employee[2]/@id	902
-----------------------	-----

Tabelle 16.3 – Beispiele einiger XPath Prädikate

XPath: Die Parent Achse

Ähnlich wie bei MS-DOS oder Linux/UNIX können Sie einen “.” (Punkt) verwenden, um im Ausdruck anzugeben, dass sie sich auf den aktuellen Kontext beziehen und ein “..” (Doppelpunkt) zur Bezugnahme auf den Parent Kontext. Table 16.4 bietet weitere Beispiele.

XPath	Ergebnis
/dept/employee/name[../@id="902"]	<name>Peter Pan</name>
/dept/employee/office[.>"300"]	<office>344</office>
/dept/employee[office > "300"]/office	<office>344</office>
/dept/employee[name="John Doe"]/../@bldg	101
/dept/employee[name[.="John Doe"]/../../@bldg	101

Tabelle 16.4 – XPath Parent Achse

16.3.3 XQuery definiert

Bei XQuery handelt es sich um eine für XML geschaffene Abfragesprache. XQuery unterstützt Pfadausdrücke (Path Expressions) um die durch native XML hierarchische Struktur zu navigieren. Tatsächlich ist XPath eine Untermenge von XQuery; deshalb trifft alles zuvor über XPath erlernte ebenfalls auf XQuery zu. XQuery unterstützt sowohl typisierte und nicht typisierte Daten. XQuery kennt keine NULL Werte, da XML Dokumente weder fehlende noch unbekannte Daten kennen. XQuery gibt eine Folge von XML Daten zurück.

Es ist wichtig zu beachten, dass XQuery und Xpath Ausdrücke Groß-/Kleinschreibung unterscheiden.

XQuery unterstützt den FLWOR Ausdruck. Wenn wir SQL als Analogie benutzen, ist es gleichbedeutend mit einem SELECT-FROM-WHERE Ausdruck. Der nächste Abschnitt beschreibt FLWOR ausführlicher.

XQuery: FLWOR Ausdruck

FLWOR bedeutet:

- FOR: iteriert durch eine Folge (Sequenz), bindet eine Variable an Elemente
- LET: bindet eine Variable an eine Folge (Sequenz)
- WHERE: nimmt Elemente von der Iteration aus
- ORDER: ordnet Elemente der Iteration neu
- RETURN: konstruiert Abfrageresultate

Es handelt sich um einen Ausdruck, welcher die Manipulation von XML Dokumenten erlaubt und Ihnen die Rückgabe eines anderen Ausdrucks ermöglicht. Nehmen wir zum Beispiel an, Sie haben eine Tabelle mit dieser Definition:

```
CREATE TABLE dept(deptID CHAR(8),deptdoc XML);
```

Und das folgende XML Dokument ist in die deptdoc Spalte eingefügt:

```
<dept bldg="101">
  <employee id="901">
    <name>John Doe</name>
    <phone>408 555 1212</phone>
    <office>344</office>
  </employee>
  <employee id="902">
    <name>Peter Pan</name>
    <phone>408 555 9918</phone>
    <office>216</office>
  </employee>
</dept>
```

Dann könnte die folgende, den FLOWR Ausdruck verwendende, XQuery Anweisung ausgeführt werden:

```
xquery
for $d in db2-fn:xmlcolumn('dept.deptdoc')/dept
let $emp := $d//employee/name
where $d/@bldg > 95
order by $d/@bldg
return
  <EmpList>
    {$d/@bldg, $emp}
  </EmpList>
```

Dies würde das folgende zurückgeben:

```
<EmpList bldg="101">
  <name>
    John Doe
  </name>
  <name>
    Peter Pan
  </name>
</EmpList>
```

16.3.4 XML Dokumente einfügen

Das Einfügen von XML Dokumenten in eine DB2 Datenbank kann mittels der SQL INSERT Anweisung oder dem IMPORT Dienstprogramm durchgeführt werden. XQuery kann für diesen Zweck nicht verwendet werden, da dies im Standard noch nicht definiert ist.

Untersuchen wir das folgende Skript welches, unter Verwendung der nachfolgenden Anweisungen, über das DB2 Befehlsfenster oder die Linux Shell ausgeführt werden kann:

```
db2 -tvf table_creation.txt
```

table_creation.txt

```
-- (1)
drop database mydb
;

-- (2)
create database mydb using codeset UTF-8 territory US
;

-- (3)
connect to mydb
;

-- (4)
create table items (
  id          int primary key not null,
  brandname   varchar(30),
  itemname    varchar(30),
  sku         int,
  srp         decimal(7,2),
  comments    xml
);

-- (5)
create table clients(
  id          int primary key not null,
  name        varchar(50),
  status      varchar(10),
  contact     xml
);

-- (6)
insert into clients values (77, 'John Smith', 'Gold',
  '<addr>111 Main St., Dallas, TX, 00112</addr>')
;

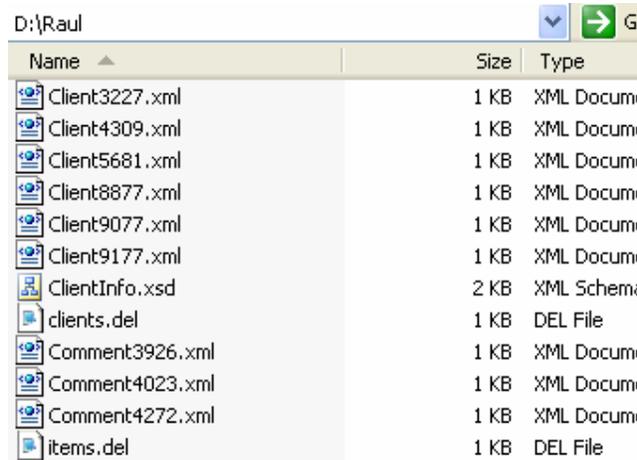
-- (7)
IMPORT FROM "D:\Raul\clients.del" of del xml from "D:\Raul" INSERT
INTO CLIENTS (ID, NAME, STATUS, CONTACT)
```

```
;  
-- (8)  
IMPORT FROM "D:\Raul\items.del" of del xml from "D:\Raul" INSERT  
INTO ITEMS (ID, BRANDNAME, ITEMNAME, SKU, SRP, COMMENTS)  
;
```

Bitte beachten Sie, dass die Skriptdatei und dazugehörigen Dateien Bestandteil der zum Buch gehörenden Zip Datei **expressc_book_quicklabs.zip** sind. Folgen Sie einfach den einzelnen Schritten in der Folge wie wir diese im Skript beschreiben.

1. Löschen der Datenbank "mydb". Dies wird normalerweise zur Bereinigung durchgeführt. Fall "mydb" nicht bereits existiert, werden Sie eine Fehlermeldung erhalten, welche sie ignorieren können.
2. Erstellen Sie die Datenbank "mydb" unter Verwendung des kodierten Zeichensatzes UTF-8. Eine UNICODE Datenbank wird für die Unterstützung von pureXML benötigt, so dass dieser Schritt erforderlich ist, um die Datenbank als UNICODE Datenbank zu erstellen.
3. Verbinden Sie sich mit der neu erstellten Datenbank "mydb". Dies ist erforderlich um Objekte in dieser Datenbank erstellen zu können.
4. Erstellen Sie die Tabelle "items". Bitte beachten Sie, dass die letzte Spalte der Tabelle (Spalte "comments") als XML Spalte, unter Verwendung des neuen XML Datentyps, definiert wurde.
5. Wir erstellen die Tabelle "clients". Bitte beachten Sie, dass die letzte Spalte der Tabelle (Spalte "contact") ebenfalls unter Verwendung des neuen Datentyps XML definiert wurde.
6. Bei Verwendung einer SQL INSERT Anweisung können Sie ein XML Dokument in eine XML Spalte einfügen. In der INSERT Anweisung geben Sie das XML Dokument als eine in einzelne Anführungszeichen gesetzte Zeichenkette an.
7. Bei Verwendung des IMPORT Befehls können Sie mehrere XML Dokumente zusammen mit relationalen Daten in die Datenbank einfügen oder importieren. In (7) importieren Sie die Daten der clients.del Datei (eine zeichenbegrenzte ASCII Datei) und geben ebenfalls an, wo sich die durch die clients.del Datei referenzierten XML Daten befinden (in diesem Beispiel im Verzeichnis D:\Raul).

Wir sehen uns die Datei clients.del noch genauer an. Doch zuerst sehen wir uns den Inhalt des Verzeichnisses D:\Raul (Abbildung 16.10) an.



Name	Size	Type
Client3227.xml	1 KB	XML Document
Client4309.xml	1 KB	XML Document
Client5681.xml	1 KB	XML Document
Client8877.xml	1 KB	XML Document
Client9077.xml	1 KB	XML Document
Client9177.xml	1 KB	XML Document
ClientInfo.xsd	2 KB	XML Schema
clients.del	1 KB	DEL File
Comment3926.xml	1 KB	XML Document
Comment4023.xml	1 KB	XML Document
Comment4272.xml	1 KB	XML Document
items.del	1 KB	DEL File

Abbildung 16.10 – Inhalt des D:\Raul Verzeichnisses mit XML Dokumenten

Nachfolgend der Inhalt der Datei clients.del:

clients.del

```
3227,Ella Kimpton,Gold,<XDS FIL='Client3227.xml' />,
8877,Chris Bontempo,Gold,<XDS FIL='Client8877.xml' />,
9077,Lisa Hansen,Silver,<XDS FIL='Client9077.xml' />
9177,Rita Gomez,Standard,<XDS FIL='Client9177.xml' />,
5681,Paula Lipenski,Standard,<XDS FIL='Client5681.xml' />,
4309,Tina Wang,Standard,<XDS FIL='Client4309.xml' />
```

In der clients.del Datei wird "XDS FIL=" verwendet um auf ein bestimmtes XML Dokument zu verweisen.

Abbildung 16.11 zeigt die Steuerzentrale nach der Ausführung des obigen Skripts.

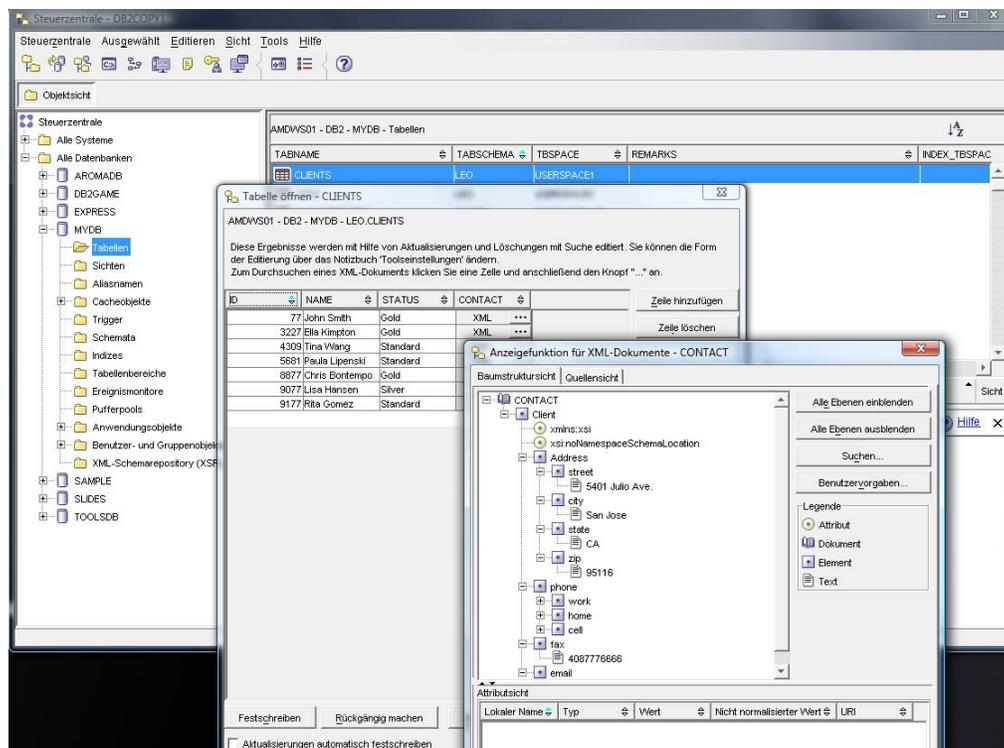


Abbildung 16.11 – Die Steuerzentrale nach dem Ausführen von table_creation.txt

Bitte beachten Sie dass wir in der Abbildung den Inhalt der CLIENTS Tabelle zeigen. Bei der letzten Spalte "Contact" handelt es sich um eine XML Spalte. Wenn Sie auf die Schaltfläche mit den drei Punkten klicken, wird ein anderes Fenster, den Inhalt des XML Dokuments zeigend, geöffnet. Dies wird in der unteren rechten Ecke der Abbildung 16.11 gezeigt.

16.3.5 XML Daten abfragen

Es gibt zwei Möglichkeiten XML Daten in DB2 abzufragen:

- Verwendung von SQL mit XML Erweiterungen (SQL/XML)
- Verwendung von XQuery

In beiden Fällen folgt DB2 internationalen XML Standards.

XML Daten mit SQL/XML abfragen

Die Verwendung einfacher SQL Anweisungen erlaubt Ihnen die Arbeit mit Zeilen und Spalten. Eine SQL Anweisung kann für die Arbeit mit vollständigen XML Dokumenten verwendet werden. Wie dem auch sei, dies reicht nicht aus um nur einen Teil des XML Dokuments auszulesen. In solchen Fällen müssen Sie SQL mit XML Erweiterungen (SQL/XML) verwenden.

Tabelle 16.5 beschreibt einige der mit dem SQL Standard 2006 verfügbaren SQL/XML Funktionen.

Funktionsname	Beschreibung
XMLPARSE	Parsed Character oder Large Object Binärdaten, produziert XML Wert
XMLSERIALIZE	Konvertiert einen XML Wert in Zeichen oder Large Object Binärdaten
XMLVALIDATE	Validiert den XML Wert gegen ein XML Schema und überprüft die Gültigkeit des XML Werts
XMLEXISTS	Stellt fest, ob eine XQuery Abfrage ein Ergebnis zurückgibt (z.B. eine Folge von einem oder mehreren Elementen)
XMLQUERY	Führt XQuery aus und gibt die resultierende Folge zurück
XMLTABLE	Führt XQuery aus und gibt die Ergebnisfolge als eine relationale Tabelle zurück (falls möglich)
XMLCAST	Überlagerung an oder von einem XML Typ

Tabelle 16.5 – SQL/XML Funktionen

Die folgenden Beispiele können mit der zuvor erstellten Datenbank "mydb" getestet werden.

Beispiel 1

Dies ist ein Beispiel eines exemplarischen Abfrageproblems. Stellen Sie sich vor, sie müssten alle Kunden, welche im einem bestimmten zip code Gebiet leben, finden. Die "clients" Tabelle speichert Kundenadressen, einschließlich zip codes, in einer XML Spalte. Bei Verwendung von XMLEXISTS können sie die XML Spalte nach der Zielpostleitzahl (zip code) durchsuchen und dann die Rückgabeergebnisse entsprechend begrenzen.

```
SELECT name FROM clients
WHERE xmlexists(
  '$c/Client/Address[zip="95116"]'
  passing clients.contact as "c"
)
```

Die erste Zeile einer SQL Klausel gibt an, dass Sie die Information in der "name" Spalte der "clients" Tabelle auslesen wollen.

Die WHERE Klausel ruft die XMLEXISTS Funktion, unter Angabe des XPath Ausdrucks, welcher DB2 veranlasst zum "zip" Element zu navigieren und auf einen Wert von 95116 zu prüfen, auf.

Die "\$c/Client/Address" Klausel gibt den Pfad innerhalb der XML Dokumenthierarchie an, wo DB2 das "zip" Element lokalisiert werden kann. Ein Dollar Zeichen (\$) wird zur Kennzeichnung einer Variablen verwendet; weshalb es sich bei "c" um eine Variable handelt. Diese Variable ist dann durch diese Zeile definiert: `passing clients.contact as "c"`. Hier ist "clients" der Name der Tabelle und "contact" der Name der Spalte mit

einem XML Datentyp. Anders gesagt übergeben wir das XML Dokument an die Variable "c".

DB2 untersucht die in der "contact" Spalte enthaltenen Daten, navigiert vom Wurzelknoten "Client" herab zum Knoten "address" und dann zum Knoten "zip" und stellt schließlich fest, ob der Kunde im zip code des Ziels lebt. Die XMLEXISTS Funktion erhält den Wert "true" und DB2 gibt den Namen des mit der Zeile assoziierten Kunden zurück.

In DB2 9.5 kann die obige Abfrage wie folgt vereinfacht werden:

```
SELECT name FROM clients
WHERE xmlexists(
  '$CONTACT/Client/Address[zip="95116"]'
)
```

Eine Variable mit demselben Namen wie eine XML Spalte wird von DB2 automatisch angelegt. Im obigen Beispiel wurde die Variable CONTACT automatisch von DB2 angelegt. Der Name stimmt mit dem Namen der XML Spalte CONTACT überein.

Beispiel 2

Betrachten wir wie das Problem gelöst werden kann, dass ein Bericht der E-Mail Adressen von Kunden mit dem Status "Gold" erstellt werden kann. Die folgende Abfrage könnte ausgeführt werden:

```
SELECT xmlquery('$c/Client/email' passing contact as "c")
FROM clients
WHERE status = 'Gold'
```

Die erste Zeile gibt an, dass wir die E-Mail Adresse, welche ein Element des XML Dokuments (keine relationale Spalte) ist, zurückgeben wollen. Wie im vorherigen Beispiel handelt es sich bei "\$c" um eine Variable, welche das XML Dokument enthält. In diesem Beispiel verwenden wir die XMLQUERY Funktion, welche nach einem SELECT verwendet werden kann, wohingegen XMLEXISTS Funktion nach einer WHERE Klausel benutzt werden kann.

Beispiel 3

Es kann Situationen geben, in welchen Sie XML Daten in Tabellenform darstellen wollen. Dies wird, wie unten gezeigt, mit der XMLTABLE Funktion ermöglicht.

```
SELECT t.comment#, i.itemname, t.customerID, Message
FROM items i,
xmltable('$c/Comments/Comment' passing i.comments as "c"
  columns Comment# integer path 'CommentID',
  CustomerID integer path 'CustomerID',
  Message varchar(100) path 'Message') AS t
```

Die erste Zeile gibt die in Ihrem Result Set einzubeziehenden Spalten an. Spalten mit vorangestellter "t" Variable basieren auf XML Elementwerten.

Die zweite Zeile ruft die XMLTABLE Funktion auf, um die die Zieldaten enthaltende DB2 XML Spalte ("i.comments") und den Pfad innerhalb des Dokuments zu spezifizieren, indem sich die interessierenden Elemente befinden.

Die "columns" Klausel in den Zeilen 4 bis 6 gibt die spezifischen XML Elemente an, welche in Ausgabespalten des in Zeile 1 angegebenen SQL Result Sets abgebildet werden. Ein Teil dieser Abbildung beinhaltet die Angabe der Datentypen, in welche die XML Elementwerte konvertiert werden. In diesem Beispiel werden alle XML Daten in traditionelle SQL Datentypen konvertiert.

Beispiel 4

Untersuchen wir jetzt ein einfaches Beispiel in welchem Sie einen XQuery FLOWR Ausdruck innerhalb einer XMLQUERY SQL/XML Funktion verwenden.

```
SELECT name, xmlquery(
  'for $e in $c/Client/email[1] return $e'
  passing contact as "c"
)
FROM clients
WHERE status = 'Gold'
```

Die erste Zeile gibt an, dass die Customer Names in der Ausgabe der XMLQUERY Funktion im Result Set eingeschlossen sind. Die zweite Zeile besagt, dass das erste "email" Sub-Element des "Client" Elements zurückgegeben werden soll. Die dritte Zeile identifiziert die Quelle Ihrer XML Daten ("contact" Spalte). Die vierte Zeile sagt uns, dass diese Spalte aus der "clients" Tabelle stammt und die fünfte Zeile zeigt an dass nur "Gold" Customers von Interesse sind.

Beispiel 5

Dieses Beispiel demonstriert erneut die XMLQUERY Funktion, welche einen XQuery FLOWR Ausdruck verwendet, jedoch dieses mal beachten Sie bitte, dass wir nicht nur XML, sondern auch HTML zurückgeben.

```
SELECT xmlquery('for $e in $c/Client/email[1]/text()
  return <p>{$e}</p>'
  passing contact as "c")
FROM clients
WHERE status = 'Gold'
```

Die XQuery return Anweisung erlaubt Ihnen, falls erforderlich, die Umwandlung der XML Ausgabe. Die Verwendung der text() Funktion in der ersten Zeile zeigt an, dass nur Textdarstellung der ersten E-Mail Adresse der in Frage kommenden Kunden von Interesse ist. Die zweite Zeile besagt, dass diese Information in HTML Paragraph-Tags eingeschlossen wird.

Beispiel 6

Das nachfolgende Beispiel verwendet die XMLELEMENT Funktion um eine Folge bzw. Serie von Postenelementen, wovon jede ein Sub-Element der aus den korrespondierenden Spalten der "items" Tabelle erhaltenen ID, Brand Name und Stock Keeping Unit (SKU)

enthält. Im Grunde genommen können Sie, wenn sie relationale Daten in XML Daten konvertieren müssen, die XMLELEMENT Funktion verwenden.

```
SELECT
  xmlelement (name "item", itemname),
  xmlelement (name "id", id),
  xmlelement (name "brand", brandname),
  xmlelement (name "sku", sku)
FROM items
WHERE srp < 100
```

Die obige Abfrage würde eine Ausgabe wie die folgende bieten:

```
<item>
  <id>4272</id>
  <brand>Classy</brand>
  <sku>981140</sku>
</item>
...
<item>
  <id>1193</id>
  <brand>Natural</brand>
  <sku>557813</sku>
</item>
```

XML Daten mit XQuery abfragen

Im vorhergehenden Abschnitt betrachteten wir, wie XML Daten unter Verwendung von SQL mit XML Erweiterungen abgefragt werden können. SQL war immer die primäre Abfragemethode und XPath in SQL eingebettet. In diesem Abschnitt besprechen wir, wie XML Daten mit XQuery abgefragt werden können. XQuery wird jetzt die primäre Abfragemethode und in einigen Fällen werden wir in XQuery eingebettetes SQL (verwenden der "db2-fn:sqlquery" Funktion) verwenden. Bei Verwendung von XQuery werden wir einige neue Funktionen aufrufen und ferner den FLOWR Ausdruck verwenden.

Beispiel 1

Ein einfaches XQuery zur Rückgabe der Kundenkontaktdaten.

```
xquery db2-fn:xmlcolumn('CLIENTS.CONTACT')
```

Einem XQuery Ausdruck muss, um DB2 mitzuteilen, dass es den XQuery Parser verwenden soll, immer der "xquery" Befehl vorangestellt werden. Andernfalls nimmt DB2 an, dass Sie einen SQL Anweisung ausführen wollen. Die **db2-fn:xmlcolumn** Funktion ist eine Funktion, welche die XML Dokumente aus den als Parameter angegebenen Spalten abruf. Er ist gleichbedeutend mit der folgenden SQL Anweisung, da er den gesamten Spalteninhalt abruf:

```
SELECT contact FROM clients
```

Beispiel 2

In diesem Beispiel verwenden wir den FLWOR Ausdruck um die Faxdaten den Kunden auszulesen.

```
xquery
  for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/fax
  return $y
```

Die erste Zeile ruft den XQuery Parser auf. Die zweite Zeile veranlasst DB2 durch Fax Sub-Elemente, enthalten in der CLIENTS.CONTACT Spalte, zu iterieren. Jedes Fax Element ist an die Variable \$y gebunden. Die dritte Zeile besagt, dass für jede Iteration der Wert \$y zurückgegeben wird.

Die Ausgabe dieser Abfrage ist ähnlich dieser (sie könnte standardmäßig den Namespace enthalten, doch wir zeigen dies unten nicht, da die Ausgabe, da sie mehrere Zeilen umfassen könnte, dann schwerer zu lesen wäre):

```
<fax>4081112222</fax>
<fax>5559998888</fax>
```

Beispiel 3

Das nächste Beispiel fragt XML Daten ab und gibt die Ergebnisse als HTML zurück.

```
xquery
  <ul> {
    for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/Address
    order by $y/zip
    return <li>{$y}</li>
  }
</ul>
```

Das Beispiel des zurückgegebenen HTML sieht wie folgt aus:

```
<ul>
<li>
<address>
  <street>9407 Los Gatos Blvd.</street>
  <city>Los Gatos</city>
  <state>ca</state>
  <zip>95302</zip>
</address>
</li>
<address>
<street>4209 El Camino Real</street>
  <city>Mountain View</city>
  <state>CA</state>
  <zip>95302</zip>
</address>
</li>
...
</ul>
```

Beispiel 4

Das folgende Beispiel zeigt wie durch Verwendung der db2-fn:sqlquery Funktion SQL in XQuery eingebettet wird. Die db2-fn:sqlquery Funktion führt eine SQL Abfrage aus und gibt nur die ausgewählten XML Daten zurück. Die an db2-fn:sqlquery übergebene SQL Abfrage muss ausschließlich XML Daten zurückgeben. Diese XML Daten können dann durch XQuery weiterverarbeitet werden.

```
xquery
  for $y in
    db2-fn:sqlquery(
      'select comments from items where srp > 100'
    )/Comments/Comment
  where $y/ResponseRequested='Yes'
  return (
    <action>
      {
        $y/ProductID
        $y/CustomerID
        $y/Message
      }
    </action>
  )
```

In dem Beispiel filtert die SQL Abfrage Zeilen basierend auf die Bedingung das die "srp" Spalte einen Wert größer 100 hat. Von diesen gefilterten Zeilen wählt es die "comments" Spalte, welche die XML Spalte ist. Als nächste wird XQuery (oder XPath) verwendet um zu den Sub-Elementen zu gehen.

Hinweis: DB2 unterscheidet Groß-/Kleinschreibung nicht und behandelt alle Tabellen- und Spaltennamen in Großschreibung, wohingegen XQuery Groß-/Kleinschreibung unterscheidet. Bei den obigen Quellfunktionen handelt es sich um XQuery Interface Funktionen, so dass alle Tabellen- und Spaltennamen an diese Funktionen in Großschreibung weitergegeben werden sollten. Weitergabe der Objektnamen in Kleinschreibung kann in einem undefinierten Objektnamefehler resultieren.

16.3.6 Joins mit SQL/XML

Dieser Abschnitt beschreibt, wie JOIN Operationen zwischen zwei XML Spalten unterschiedlicher Tabellen, oder zwischen einer XML Spalte und einer relationalen Spalte, durchgeführt werden. Nehmen wir an, Sie haben zwei Tabellen mit diesen Anweisungen erstellt:

```
CREATE TABLE dept (unitID CHAR(8), deptdoc XML)

CREATE TABLE unit (unitID CHAR(8) primary key not null,
                   name CHAR(20),
                   manager VARCHAR(20),
                   )
```

Sie können eine JOIN Operation auf einem von zwei Wegen ausführen:

Methode 1:

```

SELECT u.unitID
  FROM dept d, unit u
  WHERE XMLEXISTS (
    '$e//employee[name = $m]'
    passing d.deptdoc as "e", u.manager as "m")

```

Zeile 3 dieser Anweisung zeigt dass die JOIN Operation zwischen dem Element "name", welches ein Sub-Element der XML Spalte "deptdoc" ist, und der relationalen Spalte "manager" in der Tabelle "unit" stattfindet.

Methode 2:

```

SELECT u.unitID
  FROM dept d, unit u
  WHERE u.manager = XMLCAST(
    XMLQUERY('$e//employee/name '
    passing d.deptdoc as "e")
    AS char(20))

```

Bei dieser alternativen Methode befindet sich die relationale Spalte auf der linken Seite des JOINS. Falls sich die relationale Spalte auf der linken Seite des Vergleichs befindet kann ein relationaler Index anstelle eines XML Indexes verwendet werden.

16.3.7 Joins mit XQuery

Nehmen wir an, dass die folgenden Tabellen erstellt wurden:

```

CREATE TABLE dept(unitID CHAR(8), deptdoc XML)
CREATE TABLE project(projectDoc XML)

```

Wenn wir SQL/XML verwenden, würde die Abfrage wie folgt aussehen:

```

SELECT XMLQUERY (
  '$d/dept/employee' passing d.deptdoc as "d")
  FROM dept d, project p
  WHERE XMLEXISTS (
    '$e/dept[@deptID=$p/project/deptID]'
    passing d.deptdoc as "e", p.projectDoc as "p")

```

Unter Verwendung von XQuery würde dies die äquivalente Abfrage sein:

```

xquery
  for $dept in db2-fn:xmlcolumn("DEPT.DEPTDOC")/dept
  for $proj in db2-fn:xmlcolumn("PROJECT.PROJECTDOC")/project
  where $dept/@deptID = $proj/deptID
  return $dept/employee

```

Die zweite Methode ist leichter zu interpretieren – die Variable "\$dept" enthält das XML Dokument der XML Spalte "deptdoc" der Tabelle "dept". Die Variable "\$proj" enthält das XML Dokument der XML Spalte "projectdoc" der Tabelle "project". Dann führt Zeile 4 die

JOIN Operation zwischen einem Attribut des ersten XML Dokuments und einem Element des zweiten XML Dokuments aus.

16.3.8 Update und Delete Operationen

Update und Delete Operationen auf XML Daten können auf zwei unterschiedlichen Arten vorgenommen werden:

- ▶ Verwendung der SQL UPDATE und DELETE Anweisungen
- ▶ Verwendung des TRANSFORM Ausdrucks

Im Fall des ersten Wegs unter Verwendung der SQL UPDATE und DELETE Anweisungen, erfolgt das Update oder Delete auf Dokumentebene, d.h. das gesamte XML Dokument wird mit aktualisierten ersetzt. Sollten wir zum Beispiel im unteren Beispiel nur das <state> Element ändern wollen würde tatsächlich das gesamte XML Dokument ersetzt.

```
UPDATE clients SET contact=(
  xmlparse(document
    '<Client>
      <address>
        <street>5401 Julio ave.</street>
        <city>San Jose</city>
        <state>CA</state>
        <zip>95116</zip>
      </address>
      <phone>
        <work>4084633000</work>
        <home>4081111111</home>
        <cell>4082222222</cell>
      </phone>
      <fax>4087776666</fax>
      <email>newemail@someplace.com</email>
    </Client>')
  )
WHERE id = 3227
```

Beim zweiten Weg können Sie unter Verwendung des TRANSFORM Ausdrucks Sub-Dokument Aktualisierungen durchführen, was wesentlich effektiver ist. Dies erlaubt Ihnen Knoten in einem XML Dokument zu ersetzen, einzufügen, löschen oder umbenennen. Sie können auch den Wert eines Knotens ändern ohne den Knoten selbst ersetzen zu müssen, üblicherweise um eine Attribut- oder Elementwert zu ändern – wobei es sich um eine sehr oft vorkommende Art der Aktualisierung handelt. Das ist neu in DB2 9.5.

Da der TRANSFORM Ausdruck Bestandteil der XQuery Sprache ist, können Sie diesen überall verwenden wo sie normalerweise XQuery verwenden würden, z.B. in einem FLOWR Ausdruck oder in der XMLQUERY Funktion einer SQL/XML Anweisung. Am gebräuchlichsten ist eine SQL UPDATE Anweisung um in einem XML Dokument eine XML Spalte zu ändern.

Nachfolgend die Syntax des TRANSFORM Ausdrucks:

```
>>-transform--| copy clause |--| modify clause |--| return clause
|-><

copy clause

      .- ,-----
      v |
|--copy----$VariableName--:---CopySourceExpression-+-----|

modify clause

|--modify--ModifyExpression-----|

return clause

|--return--ReturnExpression-----|
```

Die `copy` Klausel wird verwendet, um einer Variablen die XML Dokumente, welche Sie verarbeiten wollen, zu übergeben.

In der `modify` Klausel können Sie einen `insert`, `delete`, `rename` oder `replace` Ausdruck angeben. Diese Ausdrücke erlauben Ihnen Aktualisierungen an ihrem XML Dokument vorzunehmen. Wenn Sie zum Beispiel einen neuen Knoten zum Dokument hinzufügen wollen, würden sie den `insert` Ausdruck verwenden, zum Löschen von Knoten aus dem XML Dokument verwenden sie den `delete` Ausdruck, zum umbenennen eines Elements oder Attributs in einem XML Dokument benutzen sie den `rename` Ausdruck und zur Ersetzung eines existierenden Knotens mit einem neuen Knoten oder einer Sequenz von Knoten verwenden sie den `replace` Ausdruck. Der `replace` Wert dieses Ausdrucks kann nur zur Änderung des Wert seines Elements oder Attributs verwendet werden. Die `Return` Klausel gibt das Ergebnis des Transform Ausdrucks zurück.

Hier ist das Beispiel einer UPDATE Anweisung unter Verwendung des TRANSFORM Ausdrucks.

```
(1)-- UPDATE customers
(2)-- SET contactinfo = xmlquery( 'declare default element namespace
(3)--                               "http://posample.org";
(4)--   transform
(5)--   copy $newinfo := $c
(6)--       modify do insert <email2>my2email.gm.com</email2>
(7)--       as last into $newinfo/customerinfo
(8)--   return $newinfo' passing contactinfo as "c")
(9)-- WHERE id = 100
```

Im obigen Beispiel sind die Zeilen (1), (2) und (9) Bestandteil der SQL UPDATE Syntax. In Zeile (2) wird die XMLQUERY Funktion aufgerufen, welche den Transform Ausdruck in Zeile (4) aufruft. Der Bereich des Transform Ausdrucks umfasst den Bereich der Zeilen (4) bis Zeile (8) und wird für das Einfügen eines neuen, das `email2` Element beinhaltenden,

Knotens in das XML Dokument verwendet. Bitte beachten Sie, dass die Aktualisierung der Elemente in einem XML Dokument durch eine Sicht nicht unterstützt wird.

Das Löschen vollständiger XML Dokumente aus Tabellen ist so einfach wie die Verwendung der SELECT Anweisung in SQL/XML. Verwenden Sie die SQL DELETE Anweisung und geben, falls erforderlich, WHERE Prädikate an.

16.3.9 XML Indizierung

In einem XML Dokument können Indizes für Elemente, Attribute oder Werte (Textknoten) erstellt werden. Unten finden Sie einige Beispiele. Nehmen wir an, die untenstehende Tabelle wurde erstellt:

```
CREATE TABLE customer(info XML)
```

Und unterstellen dies wäre eines der gespeicherten XML Dokumente:

```
<customerinfo Cid="1004">
  <name>Matt Foreman</name>
  <addr country="Canada">
    <street>1596 Baseline</street>
    <city>Toronto</city>
    <state>Ontario</state>
    <pcode>M3Z-5H9</pcode>
  </addr>
  <phone type="work">905-555-4789</phone>
  <phone type="home">416-555-3376</phone>
  <assistant>
    <name>Peter Smith</name>
    <phone type="home">416-555-3426</phone>
  </assistant>
</customerinfo>
```

1) Diese Anweisung erstellt einen Index auf das Attribut "Cid"

```
CREATE UNIQUE INDEX idx1 ON customer(info)
  GENERATE KEY USING
  xmlpattern '/customerinfo/@Cid'
  AS sql DOUBLE
```

2) Diese Anweisung erstellt einen Index auf das Element "name"

```
CREATE INDEX idx2 ON customer(info)
  GENERATE KEY USING
  xmlpattern '/customerinfo/name'
  AS sql VARCHAR(40)
```

3) Diese Anweisung erstellt einen Index auf alle Elemente "name"

```
CREATE INDEX idx3 ON customer(info)
  GENERATE KEY USING
```

```
xmlpattern '//name'  
AS sql VARCHAR(40);
```

4) Diese Anweisung erstellt einen Index auf alle Textknoten (alle Werte). Dies ist nicht empfohlen, da es sehr aufwändig ist den Index für die Aktualisierung, Löschung und Einfügung von Daten zu pflegen und der Index zu groß würde.

```
CREATE INDEX idx4 ON customer(info)  
GENERATE KEY USING  
xmlpattern '//text()'  
AS sql VARCHAR(40);
```

Übung 14 - SQL/XML und XQuery

Ziel

Sie haben mehrere Beispiele der SQL/XML Syntax gesehen und wurden in die Arbeit mit dem DB2 Befehlseditor und IBM Data Studio eingeführt. In dieser Übung testen Sie ihr SQL/XML und XQuery Wissen und erweitern ihre Erfahrung mit diesen Dienstprogrammen. Wir werden, wie bereits zuvor in diesem Kapitel beschrieben, die mit der `table_creation.txt` Skriptdatei erstellte Datenbank "mydb" verwenden.

Vorgehensweise

1. Erstellen Sie die "mydb" Datenbank und laden die XML Daten, wie zuvor in diesem Kapitel erläutert.
2. Verwenden Sie entweder den Befehlseditor oder das IBM Data Studio:
 - a) Lesen Sie alle Kommentare der XML Dokumente der ITEMS Tabelle unter Verwendung von XQuery auf zweierlei Art aus.
 - b) Warum würde das Absetzen dieser SQL Anweisung nicht exakt dieselbe Ausgabe zurückgeben?:
`SELECT comments FROM items`
 - c) Lesen Sie ID und BRANDNAME der Datensätze, deren XML Dokumente ein ResponseRequested Element mit einem Wert von "No" besitzen.

LÖSUNGEN:

2a)

```
xquery db2-fn:xmlcolumn('ITEMS.COMMENTS')
xquery db2-fn:sqlquery("select comments from
items")
```

2b)

Die Ausgabe ist unterschiedlich, da SQL, wenn ein Wert nicht vorhanden ist, einen NULL Wert zurückgibt, wohingegen XQuery nichts zurückgibt.

2c)

```
SELECT id, brandname FROM items WHERE
XML EXISTS ('$c/Comments/Comment[ResponseRequested="No"]')
passing ITEMS.COMMENTS as "c"
```

17

Kapitel 17 – Entwickeln mit Java, PHP und Ruby

Dieses Kapitel behandelt die Grundlagen der Anwendungsentwicklung in Java, PHP und Ruby on Rails unter Verwendung eines DB2 Servers. Es ist nicht das Ziel dieses Kapitels Ihnen diese Sprache zu beizubringen, sondern vielmehr relevante Informationen zu deren Verwendung mit DB2 zur Verfügung zu stellen.

Hinweis:

Für weitere Informationen zu diesem Thema können Sie sich dieses Video ansehen: <http://www.channeldb2.com/video/video/show?id=807741:Video:4402>

17.1 Anwendungsentwicklung mit Java

Der IBM DB2 Treiber für JDBC (auch bekannt als JCC Treiber) wurde für alle DB2 Server auf allen Plattformen optimiert. Die jar Datei db2jcc.jar (com.ibm.db2.jcc) enthält die Typ 2 und Typ 4 Treiber. Die db2jcc.jar Datei ist Bestandteil jedes DB2 Clients oder kann für sich allein (IBM DB2 Driver for JDBC and SQLJ) über die DB2 Express-C Website (ibm.com/db2/express) heruntergeladen werden.

17.1.1 JDBC Typ 2 Treiber

Der JDBC Typ 2 Treiber erfordert, dass ein DB2 Client dort installiert werden muss wo die JDBC Anwendung läuft. Abbildung 17.1 stellt eine den Typ 2 Treiber verwendende JDBC Anwendung dar.

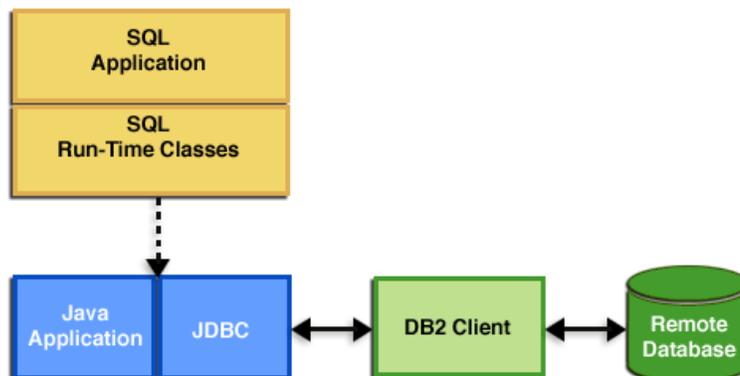


Abbildung 17.1 – Der JDBC Typ 2 Treiber

Abbildung 17.2 zeigt einen Auszug aus einem Programm, welcher zeigt, wie eine Verbindung mit dem JDBC Typ 2 Treiber hergestellt wird. Wie Sie sehen, beinhaltet die URL des von einem DB2 Klienten benutzten Programm weder Portinformationen noch Hostnamen.

```

...
public static final String DB_URL = "jdbc:db2:sample";
Properties connectProperties = new Properties();
connectProperties.put("user", "db2admin");
connectProperties.put("password", "ibmdb2");
Connection connection = null
try
{
    Class.forName("com.ibm.db2.jcc.DB2Driver").newInstance();
    connection = DriverManager.getConnection(url, connectProperties)
}
catch (Exception e)
throw e;
}
...

```

Abbildung 17.2 – Herstellen einer Verbindung unter Verwendung des JDBC Typ 2 Treibers

17.1.2 JDBC Typ 4 Treiber

Beim JDBC Typ 4 Treiber ist es nicht erforderlich, dass DB2 Client eine Verbindung zu einem DB2 Server herstellt. Abbildung 17.3 veranschaulicht eine den Typ 4 verwendende JDBC Anwendung.

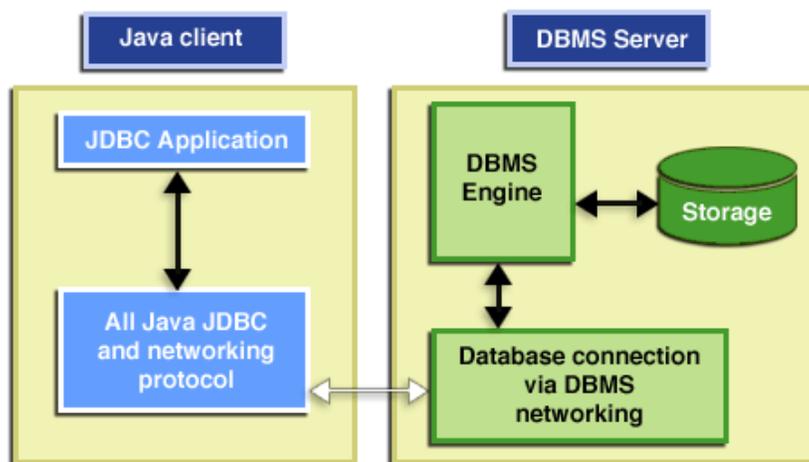


Abbildung 17.3 – Der JDBC Typ 4 Treiber

Abbildung 17.4 zeigt ein Listing wie eine Verbindung unter Verwendung des JDBC Typ 4 Treibers hergestellt wird. Bitte beachten Sie, dass die URL weder den Hostnamen, noch die Portinformationen einschließt.

```

...
public static final String DB_URL = "jdbc:db2://server1:50000/sample";
Properties connectProperties = new Properties();
connectProperties.put("user", "db2admin");
connectProperties.put("password", "ibmdb2");
Connection connection = null
try
{
    Class.forName("com.ibm.db2.jcc.DB2Driver").newInstance();
    connection = DriverManager.getConnection(url,connectProperties)
}
catch (Exception e)
    throw e;
}
...

```

Abbildung 17.4 – Herstellen einer Verbindung unter Verwendung des JDBC Typ 4 Treibers**17.2 Anwendungsentwicklung mit PHP**

Bei PHP (PHP Hypertext Preprocessor) handelt es sich um eine Open Source, plattformunabhängige Skriptsprache, welche für die Entwicklung von Webanwendungen bestimmt ist. Es handelt sich eine der Websprachen, mit dem zurzeit höchsten Verbreitungsgrad. Die Popularität von PHP basiert auf die folgenden Eigenschaften der Sprache:

- Schnelle, iterative Entwicklungszyklen mit geringem Lernaufwand.
- Robust, hohe Performanz und skalierbar.
- Stabil und sicher.
- Im Web eine Alternative zu J2EE™ und .NET.
- Leicht integrierbar in heterogene Umgebungen/Systeme.
- Erprobt durch verbreitete Implementierung
- Dynamisch und gut etablierte Community

PHP ist Teil des LAMP Stapels (Linux, Apache HTTP Server, MySQL, **PHP** / Perl / Python). Hierbei handelt es sich um einen Open Source Webtechnologiestapel, welche oft von ISPs gegen geringe monatliche Gebühren verfügbar ist.

17.2.1 DB2 Verbindungsoptionen für PHP

IBM unterstützt durch zwei Erweiterungen den Zugriff auf eine DB2 Datenbank durch PHP Anwendungen.

ibm_db2:

Die `ibm_db2` Erweiterung bietet, zusätzlich zum umfangreichen Zugriff auf Datenbank Metadaten, eine prozedurale Anwendungsprogrammierschnittstelle zum erstellen, lesen, aktualisieren und schreiben von Datenbankvorgängen. Es kann so kompiliert werden, dass es entweder mit PHP 4 oder PHP 5 arbeitet. Die Erweiterung ist unter der Apache 2.0 Lizenz über das PECL Repository erhältlich. Es wurde von IBM entwickelt und unterstützt. Es besitzt vollständige Unterstützung von gespeicherten Prozeduren und LOBs, ist schnell und wurde für DB2 optimiert.

PDO_ODBC:

`PDO_ODBC` ist ein Treiber für die PHP Data Objects (PDO) Erweiterung und bietet über eine, in PHP 5.1 eingeführte, objektorientierte Standardschnittstelle Zugriff auf DB2 Datenbanken. Er kann direkt gegen DB2 Bibliotheken kompiliert werden. Er bietet eine Standarddatenbankzugriffsschnittstelle für PHP. Er ist schnell, klein und objektorientiert. Die `PDO_ODBC` Erweiterung verwendet für den nativen Zugriff DB2 Bibliotheken und wurde in PHP 5.1 integriert. Für weitere Informationen können Sie sich diese Website ansehen:

- <http://pecl.php.net/package/pdo>
- http://pecl.php.net/package/PDO_ODBC

Verbinden mit einer nicht katalogisierten Datenbank

Listing 17.1 zeigt, wie eine Verbindung zu einer DB2 Datenbank mit einer der zuvor beschriebenen Erweiterungen hergestellt wird.

```
$host = 'localhost';
$port = 50000;
$DSN = "DRIVER={IBM DB2 ODBC DRIVER}; PORT=$port;
        HOSTNAME=$host; DATABASE=$database; PROTOCOL=TCPIP;
        USER=$user; PWD=$password";

-- If using the ibm_db2 extension --
$suconn = db2_connect($DSN, null, null);

-- If using the PDO_ODBC extension --
try {
    $suconn = new PDO("odbc:$DSN", null, null);
}
catch (PDOException $e) { print $e->errmsg(); }
```

Listing 17.1 – Verbindung zu einer nicht katalogisierten Datenbank

Listing 17.2 zeigt ein Beispiel einer einfachen, die `ibm_db2` Erweiterung verwendende, PHP Anwendung.

```
<?php
$sql = "SELECT name, breed FROM ANIMALS WHERE weight < ?";
$conn = db2_connect($database, $user, $password);
$stmt = db2_prepare($conn, $sql);
$res = db2_execute($stmt, array(10));
```

```
while ($row = db2_fetch_assoc($stmt)) {  
    print "{$row['NAME']} is a {$row['BREED']}.\\n";  
}  
?>
```

Listing 17.2 – Eine einfache, die ibm_db2 Erweiterungen verwendende, PHP Anwendung

PHP für ibm_db2 konfigurieren

Unter Linux oder UNIX müssen Sie die php.ini Datei unter Umständen wie folgt modifizieren:

```
extension=ibm_db2.so  
ibm_db2.instance_name=<instance name>
```

Unter Windows ändern Sie die php.ini Datei wie folgt:

```
extension=php_ibm_db2.dll
```

Alternativ können Sie die **Zend Core for IBM** Anwendungssuite, wie im nächsten Abschnitt beschrieben, downloaden und installieren und müssen sich nicht mehr um diese Konfigurationsproblematiken kümmern.

17.2.2 Zend Core for IBM

Bei Zend Core handelt es sich um eine nahtlose Out-Of-The-Box PHP Entwicklungs- und Produktionsumgebung für geschäftskritische Webanwendungen. Es liefert die für den Betrieb von PHP Anwendungen erforderliche Zuverlässigkeit, Produktivität und Flexibilität. Es kann unter der folgenden Adresse kostenlos heruntergeladen werden:

<http://ibm.com/software/data/info/zendcore>

Zend Core for IBM installiert DB2 und IDS Clients, sowie optional den Apache HTTP Server, PHP 5 und populäre PHP Erweiterungen einschließlich ibm_db2 und PDO_INFORMIX. Zend Core for IBM kann optional den DB2 Express-C Server, IBM Cloudscape™ Server, das vollständige PHP Handbuch und Beispielanwendungen für DB2 installieren. Es besitzt, wie in den Abbildungen 17.5, 17.6 und 17.7 gezeigt, eine leicht zu verwendende und konfigurierende PHP Umgebung.

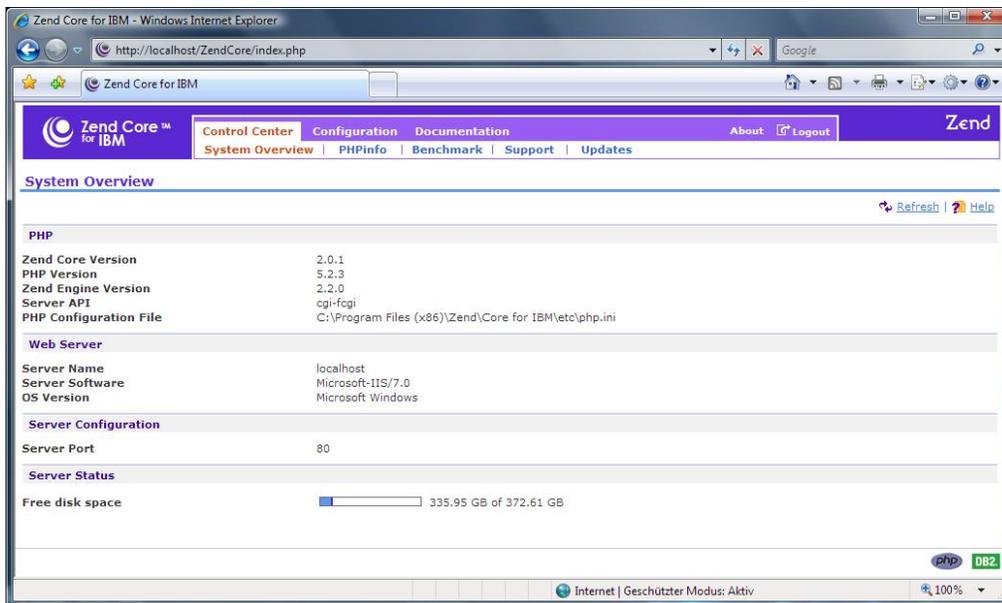


Abbildung 17.5 - Zend Core Verwaltungs- und Kontrollschnittstelle

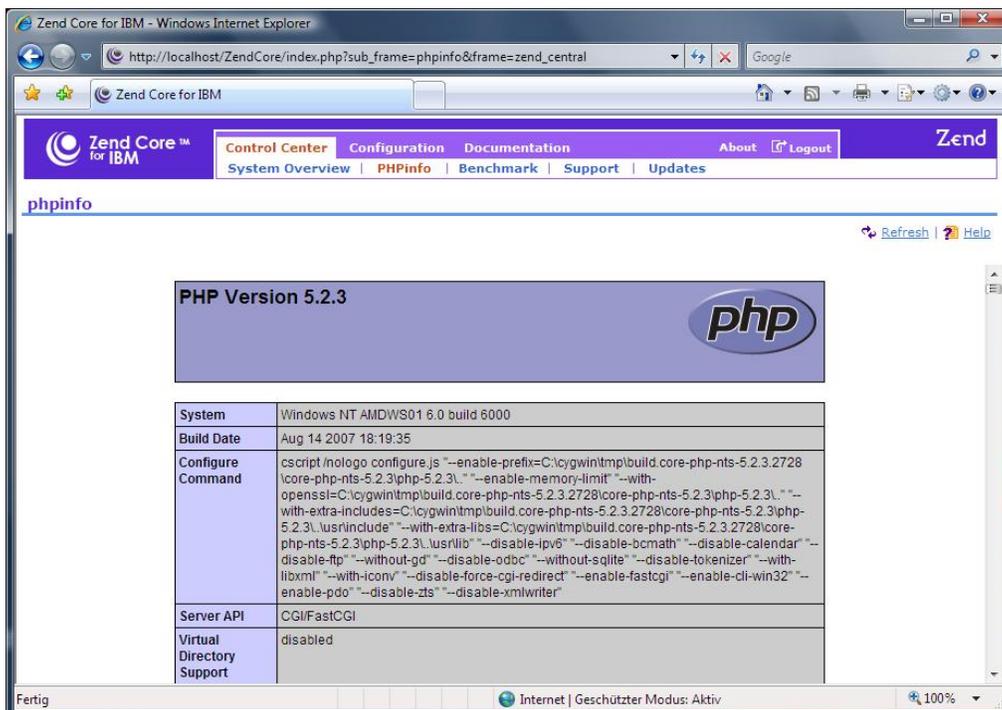


Abbildung 17.6 - Zend Core PHP Konfigurationsschnittstelle

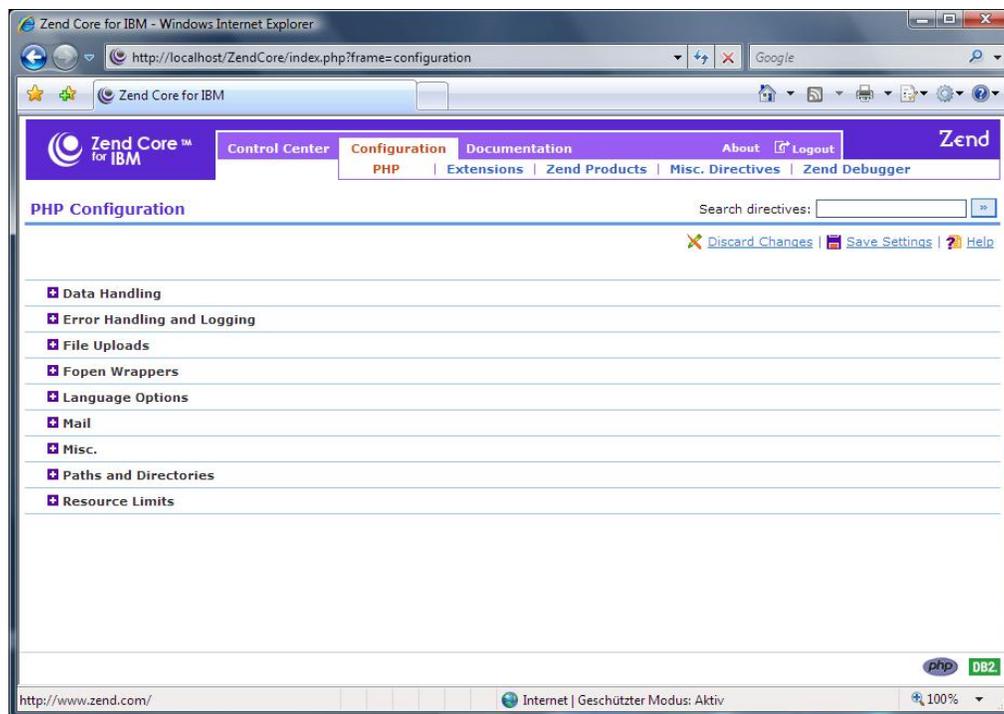


Abbildung 17.7 - Zend Core PHP Konfigurationsschnittstelle

17.3 Anwendungsentwicklung mit Ruby on Rails

Bei Ruby handelt es sich um eine objektorientierte, dynamische, plattformunabhängige Skriptsprache. Es bietet schnelle Entwicklung (Rapid Development) und enthält eine umfangreiche Bibliothek. Ruby ist eine einfache, 1995 von Yukihiro Matsumoto ("Matz") erfundene, spaßorientierte Programmiersprache.

Rails ist der vollständige Rahmen für in Ruby geschriebene Datenbankgestützte Webanwendungen. Es implementiert die Modell-Präsentation-Steuerung (MVC – Model-View-Control) Architektur. Es ist äußerst produktiv und leicht zu handhaben. Rails ist eines der am schnellsten Marktanteile gewinnende Rahmen für das Web, seit es 2004 von David Heinemeier Hansson erfunden wurde.

17.3.1 Starter Toolkit für DB2 on Rails

IBM erkennt die Wichtigkeit von Ruby on Rails für die Entwickler Community und hat aus diesem Grund ein Paket, **Startup Toolkit for DB2 on Rails** genannt, erstellt. Es handelt sich um einen integrierten Installer, welcher eine vollständige Entwicklungsumgebung aufbaut. Er kann über die IBM alphaWorks Website kostenlos heruntergeladen und verwendet werden: <http://www.alphaworks.ibm.com/tech/db2onrails>.

Das Starter Toolkit für DB2 on Rails:

- ▶ Enthält einen integrierten Installer
- ▶ Hilft Ihnen Ruby on Rails einfach zu installieren und konfigurieren
- ▶ Installiert DB2 Express – C 9 und Dienstprogramme
- ▶ Enthält einen von IBM entwickelten DB2 Ruby Treiber und einen DB2 Rails Adapter
- ▶ Enthält verschiedene Demos und Übungen

A

Anhang A – Problemlösung

Dieser Anhang behandelt wie Probleme, welche während der Arbeit mit DB2 auftreten können, gelöst werden. Abbildung A.1 gibt einen kurzen Überblick über die beim Auftreten eines Problems erforderlichen Schritte.

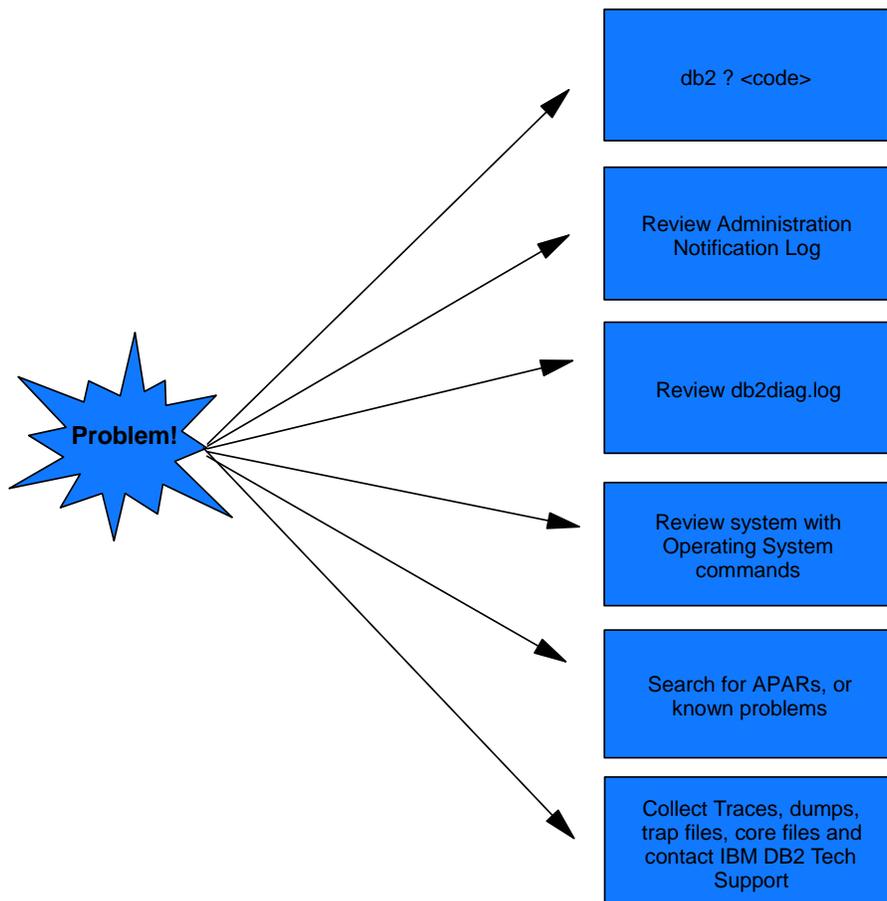


Abbildung A.1 – Problemlösungsübersicht

Hinweis:

Für weitere Informationen zur Fehlersuche können Sie sich dieses Video ansehen:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4462>

A.1 Weitere Informationen über Fehlercodes erhalten

Um weitere Informationen über zurückgegebene Fehlercodes zu erhalten, geben Sie den Fehlercode mit vorangestelltem Fragezeichen in das Eingabefenster des Befehlseditors ein und klicken auf die *Ausführen* Schaltfläche. Dies wird in Abbildung A.2 gezeigt.



Abbildung A.2 – Weitere Informationen über DB2 Fehlercodes erhalten

Das Fragezeichen (?) ruft den DB2 Help Befehl auf. Unten sehen Sie einige Beispiele wie er zur Hilfestellung aufgerufen wird falls sie zum Beispiel den SQL Fehlercode “-104” erhalten. Alle untenstehenden Beispiele sind äquivalent.

```
db2 ? SQL0104N
db2 ? SQL104N
db2 ? SQL-0104
db2 ? SQL-104
db2 ? SQL-104N
```

A.2 SQLCODE und SQLSTATE

Ein SQLCODE wird nach jeder Ausführung einer SQL Anweisung zurückgeben. Die Bedeutungen der Werte sind unten zusammengefasst:

SQLCODE = 0; der Befehl war erfolgreich
 SQLCODE > 0; der Befehl war erfolgreich, gab jedoch eine Warnung zurück
 SQLCODE < 0; der Befehl war nicht erfolgreich und gab einen Fehler zurück

Beim SQLSTATE handelt es sich um eine, mit dem ISO/ANSI SQL92 Standard konforme, fünf Zeichen lange Zeichenkette. Die ersten zwei Zeichen sind unter der Bezeichnung SQLSTATE Class Code bekannt:

Ein Class Code von 00 besagt das der Befehl erfolgreich war.
Ein Class Code von 01 impliziert eine Warnung.
Ein Class Code von 02 impliziert einen Not Found Zustand.
Alle anderen Class Codes werden als Fehler betrachtet.

A.3 Administratives DB2 Nachrichtenprotokoll

Das administrative DB2 Nachrichtenprotokoll stellt zum Zeitpunkt des Fehlers diagnostische Informationen über Fehler bereit. Unter Linux/UNIX Plattformen handelt es sich beim administrative Nachrichtenprotokoll um eine <instance name>.nfy genannte Textdatei (z.B. "db2inst.nfy"). Unter Windows werden alle administrativen Nachrichten in das Windows Ereignisprotokoll geschrieben.

Der DBM Konfigurationsparameter `notifylevel` erlaubt Administratoren die aufzuzeichnende Informationsebene anzugeben:

- 0 – Es werden keine administrativen Nachrichten aufgezeichnet (nicht empfohlen)
- 1 -- Fatale oder nicht behebbare Fehler
- 2 – Umgehender Eingriff erforderlich
- 3 – Wichtige Information, kein sofortiger Eingriff erforderlich (der Standard)
- 4 -- Informationsmeldungen

A.4 db2diag.log

Die Datei db2diag.log bietet ausführlichere Informationen als das administrative DB2 Nachrichtenprotokoll. Es wird normalerweise vom Technischen IBM DB2 Support oder erfahrenen DBAs verwendet. Im db2diag.log finden Sie folgende Informationen:

- Die DB2 Codeadresse, welche den Fehler meldet.
- Anwendungskennungen, welche es Ihnen erlauben Einträge der Datei db2diag.log auf Servern und Clients mit den zugehörigen Anwendungen in Zusammenhang zu bringen.
- Eine Diagnosenachricht (beginnend mit "DIA"), welche die Ursache des Fehlers erläutert.
- Andere verfügbare Daten, wie SQLCA Datenstrukturen und Zeiger auf die Speicherstelle von separaten Speicherauszugsdateien oder Trap Daten (Trap Level = Stufe, auf welcher Fehler abgefangen werden).

Unter Windows wird die Datei db2diag.log normalerweise im folgenden Verzeichnis gespeichert:

```
C:\Program Files\IBM\sqllib\\db2diag.log
```

Unter Linux/UNIX wird die Datei db2diag.log normalerweise im folgenden Verzeichnis gespeichert:

```
/home/<instance_owner>/sqllib/db2dump/db2diag.log
```

Der Umfang des Diagnosetextes wird durch den DBM CFG Konfigurationsparameter DIAGLEVEL festgelegt. Es sind Werte von 0 bis 4 zulässig, wobei 0 die wenigsten und 4 die meisten Informationen zurückgibt. Die Standardstufe ist 3.

A.5 CLI Fehlerprotokollierung

Für CLI (Call Level Interface oder Standard-API zum Zugriff auf Datenbanken) und Java Anwendungen können Sie die CLI Fehlerprotokollierung zur Fehlersuche in ihren Anwendungen einschalten. Dies wird dadurch erreicht, dass Änderungen an der Datei db2cli.ini auf dem Server auf welchem die Anwendung läuft vorgenommen werden. Typische Einträge in der db2cli.ini Datei sehen Sie unten.

```
[common]
trace=0
tracerefreshinterval=300
tracepathname=/path/to/writeable/directory
traceflush=1
```

Low-Level Fehlerprotokollierung (db2trc) ist ebenfalls verfügbar, jedoch üblicherweise nur für den Technischen DB2 Support verwendbar.

A.6 DB2 Fehler und Fixes

Manchmal begegnen Sie einem, auf einen Fehler in DB2 beruhenden, Problem. IBM gibt in regelmäßigen Zeitabständen sog. Fix Packs mit Fehlerbereinigungen (APARs) frei. Die Fix Pack Dokumentation enthält eine Liste der mit dem Fix Pack bereinigten Fehler. Bei der Entwicklung neuer Anwendungen empfehlen wir grundsätzlich die Verwendung des aktuellen Fix Packs, damit sie von den Verbesserungen profitieren. Um Ihre aktuelle Version und die Fix Pack Stufe zu sehen, öffnen sie in der Steuerzentrale das **Hilfe** Menü und wählen **Produktinfo** oder geben auf der Befehlszeile "db2level" ein. Bitte beachten Sie, dass nur Abonnenten der 12monatigen DB2 Express-C Subskriptionslizenz Fix Packs erhalten.

Ressourcen

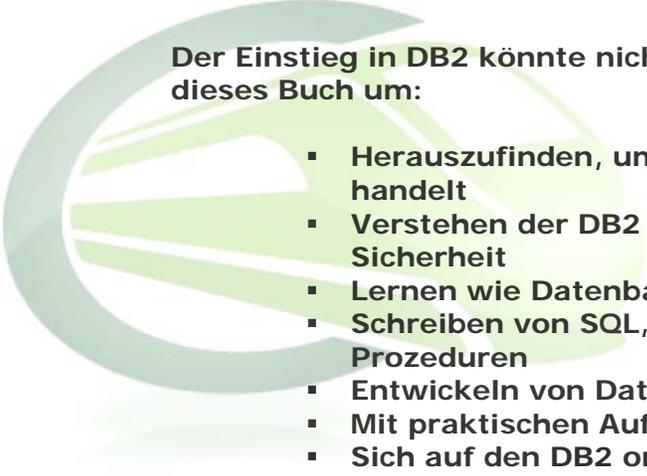
Websites:

1. DB2 Express-C Website:
www.ibm.com/db2/express
Verwenden Sie diese Website für den Download des DB2 Express-C Server Images, DB2 Clients, DB2 Treiber, Handbücher, Zugriff auf das Team Blog, Eintrag in Mailinglisten usw.
2. DB2 Express Forum:
www.ibm.com/developerworks/forums/dw_forum.jsp?forum=805&cat=19
Verwenden Sie das Forum um technische Fragen zu senden wenn sie selber in den Handbüchern keine Antwort finden.
3. DB2 Information Center
<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp>
Das Information Center bietet Zugriff auf die Online Handbücher. Es handelt sich dabei um die aktuellste Informationsquelle.
4. developerWorks
<http://www-128.ibm.com/developerworks/db2>
Die Website, kostenlosen Zugriff auf aktuelle Artikel und Übungen usw. bietend, ist eine hervorragende Quelle für Entwickler und DBAs.
5. alphaWorks
<http://www.alphaworks.ibm.com/>
Diese Website bietet direkten Zugriff auf IBMs zukünftige Technologie. Hier finden sie die neuesten Technologien von IBM Research.
6. planetDB2
www.planetDB2.com
Hierbei handelt es sich um einen Dienstleister für über DB2 bloggende Beitragsleistende.
7. DB2 Technical Support
Falls Sie die 12 monatige DB2 Express-C Subskriptionslizenz erworben haben, können sie Fix Packs von dieser Website laden.
http://www.ibm.com/software/data/db2/support/db2_9/
8. ChannelDB2
Bei ChannelDB2 handelt es sich um ein soziales Netzwerk für DB2 Community. Es stellt Inhalte wie mit DB2 verwandte Videos, Demos, Podcasts, Blogs, Diskussionen, Ressourcen usw. für Linux, UNIX, Windows, z/OS und i5/OS zur Verfügung.
<http://www.ChannelDB2.com/>

Bücher:

1. Free Redbook: DB2 Express-C: The Developer Handbook for XML, PHP, C/C++, Java, and .NET
Whei-Jen Chen, John Chun, Naomi Ngan, Rakesh Ranjan, Manoj K. Sardana,
August 2006 - SG24-7301-00
<http://www.redbooks.ibm.com/abstracts/sg247301.html?Open>
2. Understanding DB2 – Learning Visually with Examples V9.5
Raul F. Chong, et all. January 2008
ISBN-10: 0131580183
3. DB2 9: pureXML overview and fast start by Cynthia M. Saracco, Don Chamberlin,
Rav Ahuja June 2006 SG24-7298
<http://www.redbooks.ibm.com/abstracts/sg247298.html?Open>
4. DB2® SQL PL: Essential Guide for DB2® UDB on Linux™, UNIX®, Windows™,
i5/OST™, and z/OS®, 2nd Edition
Zamil Janmohamed, Clara Liu, Drew Bradstock, Raul Chong, Michael Gao, Fraser
McArthur, Paul Yip
ISBN: 0-13-100772-6
5. Free Redbook: DB2 pureXML Guide
Whei-Jen Chen, Art Sammartino, Dobromir Goutev, Felicity Hendricks, Ippei Komi,
Ming-Pang Wei, Rav Ahuja, Matthias Nicola. August 2007
<http://www.redbooks.ibm.com/abstracts/sg247315.html?Open>
6. Information on Demand - Introduction to DB2 9 New Features
Paul Zikopoulos, George Baklarz, Chris Eaton, Leon Katsnelson
ISBN-10: 0071487832
ISBN-13: 978-0071487832
7. Redbook: Developing PHP Applications for IBM Data Servers.
Whei-Jen Chen, Holger Kirstein, Daniel Krook, Kiran H Nair, Piotr Pietrzak
May 2006 - SG24-7218-00
<http://www.redbooks.ibm.com/abstracts/sg247218.html?Open>

Kontakt E-MailsAllgemeine DB2 Express-C Mailbox: db2x@ca.ibm.comAllgemeine DB2 on Campus Program Mailbox: db2univ@ca.ibm.com



Der Einstieg in DB2 könnte nicht einfacher sein. Lesen Sie dieses Buch um:

- **Herauszufinden, um was es sich bei DB2 Express-C handelt**
- **Verstehen der DB2 Architektur, Dienstprogramme, Sicherheit**
- **Lernen wie Datenbanken verwaltet werden**
- **Schreiben von SQL, XQuery, gespeicherten Prozeduren**
- **Entwickeln von Datenbankanwendungen für DB2**
- **Mit praktischen Aufgaben zu üben**
- **Sich auf den DB2 on Campus Test vorzubereiten**

Die schnelle Annahme von XML für die Anwendungsintegration, Web 2.0 und SAO erhöht die Nachfrage nach innovativen hybriden Datenservern. Bei DB2 Express-C von IBM handelt es sich um einen kostenlosen, grenzenlosen hybriden Datenbankserver, welcher in der Lage ist, sowohl XML als auch traditionelle relationale Daten mit Leichtigkeit zu verwalten. Kostenlos bedeutet, dass DB2 Express-C kostenlos heruntergeladen werden kann, kostenlos für die eigene Anwendungsentwicklung ist, frei für die Implementierung in der Produktion ist und kostenlos mit Ihrer Lösung weitergegeben werden kann. Und, DB2 begrenzt die Größe von Datenbanken, die Anzahl der Datenbanken und die Anzahl der Benutzer nicht künstlich.

DB2 Express-C läuft unter Windows und Linux Systemen und bietet Application-Driver eine Vielzahl von Programmiersprachen, einschließlich C/C++, Java, .NET, PHP, Perl und Ruby. Eine optionale kostengünstige Subskription und Support mit zusätzlichen Einsatzmöglichkeiten ist ebenfalls verfügbar. Wenn Sie noch größere Skalierbarkeit und mehr fortgeschrittene Funktionalität benötigen, können sie ihre mit DB2 Express-C erstellten Anwendungen nahtlos auf andere DB2 Editionen, wie DB2 Enterprise, übertragen.

Diese kostenlose DB2 Edition ist ideal für Entwickler, Berater, ISVs, DBAs, Studenten oder allen, die vorhaben Datenbankanwendungen zu entwickeln, testen, implementieren oder weiterzugeben. Treten Sie der wachsenden DB2 Express-C Benutzer Community bei und testen DB2 Express-C. Beginnen Sie zu entdecken, wie sie Anwendungen der nächsten Generation erstellen und innovative Lösungen liefern können.