

Praktikum Datenbanken / DB2
Woche 4: Benutzung von DB2, SQL als DDL

Raum: LF 230

Nächste Sitzung: 3./6. November 2003

Einführung in IBM DB2

Im Praktikum wird als Datenbank-Managementsystem (DBMS) das kommerzielle System *IBM DB2 Universal Database* in der Version 8.1 verwendet. Dieses läuft auf dem Rechner **salz**, auf den übrigen Maschinen des Rechnerpools LF230 laufen *Administration Clients* für IBM DB2.

Die Dokumentation zu DB2 steht online zur Verfügung. Eine lokale Installation der Dokumentation findet sich unter der Adresse http://salz.is.informatik.uni-duisburg.de/db2doc/de_DE/index.htm.

Die englischsprachigen Handbücher für IBM DB2 V8.1 liegen als PDF Dateien lokal im Verzeichnis `/usr/projects/db2doc/`. Diese Handbücher sind sehr umfangreich und sollten nur zum Betrachten am Bildschirm herangezogen und nicht ausgedruckt werden. Die Referenzen in diesem Arbeitsblatt beziehen sich auf diese Handbücher. Die gleichen Informationen sind jedoch auch in der Online-Dokumentation zu finden.

Jede Kleingruppe im Praktikum besitzt eine eigene Instanz auf **salz** unter dem Namen ihres Accounts (z.B. `dbpw0301`). Auf dieser besitzt die Gruppe das Administratorrecht, und kann Datenbanken anlegen, ändern und löschen.

Um auf **salz** mit Eurer Datenbankinstanz zu arbeiten, könnt Ihr entweder die Verknüpfung *Command Line Processor* (CLP) im Desktop-Ordner *IBM DB2* benutzen, oder auf einer Befehlszeile das Kommando `ssh salz db2 -c -t` eingeben. Dies startet den CLP im interaktiven Modus.

Wichtige Befehle des CLP

In einer interaktiven Session mit dem CLP müssen alle Befehle mit einem Semikolon beendet werden. Ein `<RETURN>` ohne Semikolon setzt den Befehl in der nächsten Zeile fort. Weder bei Befehlen noch bei den Namen von Datenbanken oder Tabellen sind Groß- und Kleinschreibung wichtig.

`?:` liefert eine Liste möglicher Befehle

`? <BEFEHL>:` gibt Hilfe zu einem bestimmten Befehl, oder einer Fehlermeldung:

```
db2 => ? SQL1013N;
```

```
SQL1013N The database alias name or database name "<name>" could  
not be found.
```

`connect to <NAME>:` baut eine Verbindung zur Datenbank Name auf

`terminate:` beendet die Sitzung (Achtung: der Befehl `quit` beendet zwar den CLP, aber schließt nicht die Verbindung zur Datenbank)

Control Center

Angenehmer als über den CLP lässt sich IBM DB2 über das *Control Center* (CC) ansprechen. Auch für dieses existiert eine Verknüpfung im Desktop-Folder *IBM DB2*. Der Befehl aus einer Shell lautet `ssh -X salz db2cc`.

Das *Control Center* ist der zentrale Knoten des graphischen Benutzerinterfaces zu DB2. Von hier aus können alle Wizards, Werkzeuge und Konfigurationsprogramme aufgerufen werden. Das CC startet mit einer Ansicht aller katalogisierten Systeme, in der man in der bekannten Weise browsen kann. Es ist möglich, sich eigene Ordner mit direkten Verknüpfungen zu den für den Benutzer interessanten Datenbanken anzulegen:

→ Menu → Selected → Create Custom Folder

Das CC bietet eine Übersicht über alle katalogisierten Instanzen auf dem lokalen und allen remote administrierten Servern, sowie über die Datenbanken, Tabellen, Views, etc. innerhalb dieser Instanzen. Über das Kontextmenü (Rechtsklick) lassen sich bei einem Eintrag unterhalb der Datenbankebene Filter setzen, um etwa die Systemtabellen und -views auszublenden.

Ebenfalls über den Weg des Kontextmenü kann man über den Punkt **create** in diesen Ordnern neue Tabellen, Trigger, Views, etc. mit Hilfe von Wizards anlegen. In jedem Wizard und Dialog des CC gibt es einen Button *Show SQL*, über den man sich die SQL-Statements anzeigen lassen kann, die tatsächlich ausgeführt werden.

Command Center

Das *Command Center* ist das grafische Äquivalent zum CLP. Es wird über das *Tools*-Menü im CC gestartet. Es verfügt über bessere Editiermöglichkeiten als der CLP, sowie über eine History-Funktion. Ein Befehl wird durch einen Mausklick auf den Zahnrad-Button abgeschickt.

Grundlegende DB2 Struktur

Innerhalb einer DB2 Installation existieren die folgenden wichtigen Datenbanksystem-Objekte:

- Instanzen
- Datenbanken
- Schemata
- Tabellen
- Views

Instanz

Eine Instanz (auch: Datenbankmanager) ist ein Objekt, das Daten verwaltet. Es kontrolliert, was mit den Daten gemacht werden kann, und verwaltet die ihm zugewiesenen Systemressourcen. Jede Instanz ist eine vollständige Arbeitsumgebung, die ihre eigenen, für andere Instanzen unzugänglichen Datenbanken enthält. Alle Datenbankpartitionen einer Instanz teilen sich das gleiche Systemverzeichnis. Eine Instanz hat eine eigene Rechteverwaltung, die unabhängig von der anderer Instanzen ist.

Datenbank

Eine relationale Datenbank ist eine Sammlung von Tabellen. Eine Tabelle besteht aus einer festen Zahl von Spalten, die den Attribute des Relationenschemas entsprechen, sowie einer beliebigen Zahl an Reihen. Zu jeder Datenbank gehören Systemtabellen, welche die logische und physikalische Struktur der Daten beschreiben, eine Konfigurationsdatei mit den Parametern der Datenbank und ein Transaktionslog.

Tabellen

Eine relationale Datenbank speichert Daten als eine Menge von zweidimensionalen Tabellen. Jede Spalte der Tabelle repräsentiert ein Attribut des Relationenschemas, jede Zeile entspricht einer spezifischen Instanz dieses Schemas. Daten in diesen Tabellen werden mit Hilfe der Structured Query Language (SQL) manipuliert, einer standardisierten Sprache zur Definition und Manipulation von Daten in relationalen Datenbanken:

```
SELECT <Daten>
FROM <Tabelle>
WHERE <Bedingung>
```

Views

Ein View ist eine effiziente Art, um Daten wiederzugeben, ohne sie in einer Tabelle zu speichern. Ein View ist keine echte Tabelle, sondern eine Sicht auf einen Ausschnitt der Datenbank, und braucht keinen permanenten Platz. Eine virtuelle Tabelle wird beim Zugriff angelegt und genutzt. Ein View kann alle oder einige der Spalten oder Zeilen der Tabellen enthalten, auf denen die Sicht basiert.

Schemata

Ein Schema ist eine Sammlung von benannten Objekten (Tabellen, Sichten, Trigger, Funktionen,...). Jedes Objekt in der Datenbank liegt in einem Schema. Objektnamen müssen innerhalb eines Schemas eindeutig sein. Ein Objekt in der Datenbank wird durch einen *qualifizierten* Namen identifiziert. Dieser besteht aus dem Namen des Schemas, in dem das Objekt liegt, gefolgt von einem Punkt und dem Objektnamen: IMDB.FILME würde beispielsweise das Objekt FILME im Schema IMDB bezeichnen.

Wird nur ein Objektname angegeben, so qualifiziert DB2 diesen standardmässig durch den Benutzernamen. Will man diesen Standardwert ändern, kann man sich des Befehls SET SCHEMA bedienen:

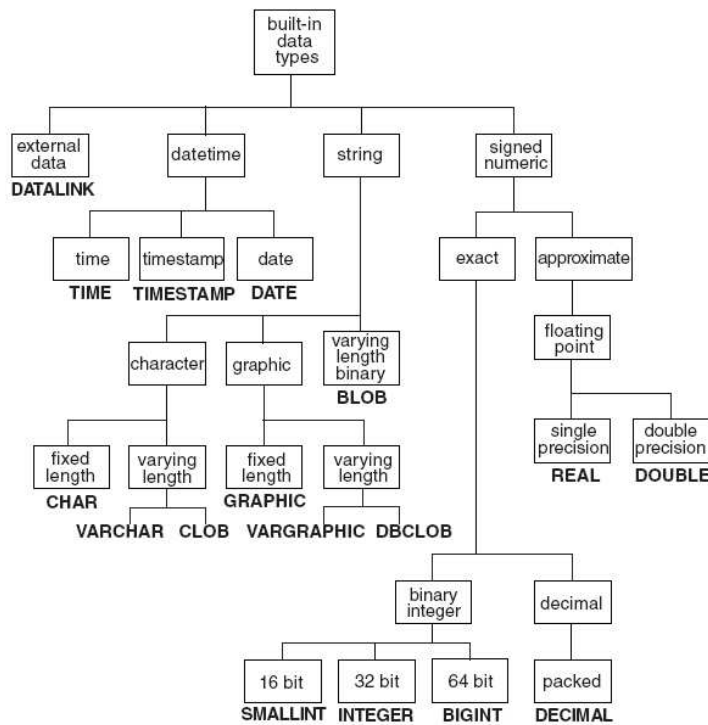
```
SET SCHEMA = IMDB;
```

Ein Benutzer, der über die Datenbankberechtigung IMPLICIT_SCHEMA (siehe unten) verfügt, kann ein Schema implizit erzeugen, wenn er in einem CREATE-Statement ein noch nicht existierendes Schema verwendet. Dieses neue, von DB2 erzeugte Schema, gehört standardmässig SYSTEM und jeder Benutzer hat das Recht, in diesem Schema Objekte anzulegen.

Datentypen

*db2s1e80.pdf,
S. 92ff*

DB2 kennt die in der nachstehenden Grafik abgebildeten Datentypen. Weitere Datentypen können vom Benutzer definiert werden (mehr dazu zu einem späteren Zeitpunkt).



Zu jedem Datentyp gehört ein NULL-Wert. Dieser ist von allen anderen Werten des Datentyps zu unterscheiden, da er nicht einen Wert an sich darstellt, sondern das Fehlen eines Wertes anzeigt. Der Wert 0 im Gehaltsfeld eines Angestellten könnte z.B. bedeuten, dass der Angestellte ehrenamtlich tätig ist, während der NULL-Wert bedeuten könnte, dass das Gehalt nicht bekannt ist. IBM DB2 bietet Prädikate an, die es z.B. in Anfragen erlauben, zu prüfen, ob ein NULL-Wert vorliegt oder eine andere Ausprägung des Datentyps.

Defaultwerte sind Werte, die vom DBMS eingetragen werden, wenn für das betreffende Attribut kein Wert angegeben wird. Sie bieten die Möglichkeit, Attribute automatisch mit Daten zu versehen, z.B. einer bestimmten Konstante, dem Namen des Benutzers (USER), der aktuellen Uhrzeit (CURRENT TIME), dem aktuellen Datum (CURRENT DATE) oder automatisch generierten Werten (GENERATE). Wird kein spezieller Defaultwert angegeben, so wird der NULL-Wert als Defaultwert verwendet.

Benutzerverwaltung

Das Authentisierungskonzept von DB2 sieht keine gesonderten Benutzernamen vor, sondern baut auf das Benutzerkonzept des Betriebssystems auf. Jede Kleingruppe

meldet sich unter dem Namen ihres Accounts bei der Datenbank an, zusätzlich gehört jede Kleingruppe zur Benutzergruppe **students**.

In einer DB2-Instanz gibt es drei Stufen der Zugriffsberechtigung. Als Instanzbesitzer hat jede Gruppe in ihrer Instanz das SYSADM-Recht.

SYSADM: Zugriff auf alle Daten und Ressourcen der Instanz

SYSCTRL: Verwalten der Instanz, aber kein direkter Zugriff auf Daten

SYSMAINT: niedrigste Berechtigungsstufe; kann z.B. Backups erstellen

Um Berechtigungen in der Datenbank, mit der man gerade verbunden ist, zu setzen oder zu verändern, bedient man sich des GRANT-Statements. Umgekehrt wird das REVOKE-Statement benutzt, um Rechte auf einer Datenbank zu entziehen. GET AUTHORIZATION zeigt die aktuellen Berechtigungen an. Beispiele:

*db2s2e80.pdf,
S. 569/642*

```
GRANT dbadm ON DATABASE  
TO USER dbpw0301;
```

```
GRANT connect, createtab, ON DATABASE  
TO GROUP students;
```

```
GRANT connect ON DATABASE  
TO public;
```

```
REVOKE connect ON DATABASE  
TO public;
```

Ein Benutzer oder eine Gruppe mit dem DBADM-Recht besitzt automatisch auch alle anderen Rechte auf dieser Datenbank, insbesondere auch das Recht IMPLICIT_-SCHEMA. Das Recht DBADM kann nicht an PUBLIC vergeben werden. Nur ein Benutzer mit SYSADM-Recht kann das DBADM-Recht an andere Benutzer oder Gruppen verleihen oder entziehen. Wird einem Benutzer ein Recht entzogen, bedeutet das nicht notwendig, dass dieser das Recht nun nicht mehr besitzt, da Rechte auch durch Zugehörigkeit zu Gruppen bestehen können.

Wer das Recht DBADM oder SYSADM besitzt kann Berechtigungen für ein Schema ändern. Dabei kann einem Benutzer auch das Recht gegeben werden, die erhaltenen Rechte an Dritte weiterzugeben. Der Befehl zum Vergeben bzw. Entziehen von Rechten auf Schemata ist GRANT/REVOKE ... ON SCHEMA.

*db2s2e80.pdf,
S. 583/657*

Um über das CC Rechte zu verwalten, muss eine Datenbank ausgewählt werden. Im Kontextmenü existiert ein Menüpunkt *Authorities*, der einen Dialog zum Ändern von Rechten öffnet. Eine weitere Möglichkeit zum Setzen von Rechten im CC ist der Ordner *User and Group Objects* unterhalb einer Datenbank.

SQL als DDL

SQL vereint in sich eine *Data Definition Language*, eine *Data Manipulation Language* und eine *Query Language*. In dieser Sitzung sollen zunächst nur die DDL-Funktionen von SQL zum Einsatz kommen.

Der folgende Befehl erstellt eine neue Datenbank in der aktuellen Instanz:

```
CREATE DATABASE <NAME>  
ON <PATH>
```

```

USING CODESET <CODESET>
TERRITORY DE
COLLATE USING SYSTEM

```

Die möglichen Codesets sind ISO-8859-1, ISO-885915 und UTF-8. Als Pfad für die Datenbank nehmt bitte `/home/dbpw03xx` (mit Eurer Gruppennummer statt `xx`). Im CC kann man eine neue Datenbank über das Kontextmenü des *Databases*-Ordner erstellen.

Erstellen von Tabellen

Jede Gruppe kann aufgrund der vorgegebenen Rechte in von ihnen erstellten Datenbanken neue Tabellen anlegen, eventuell mit impliziter Erstellung eines neuen Schemas. Zum Anlegen einer Tabelle benutzt man das `CREATE TABLE`-Statement. Der Benutzer, der die Tabelle erstellt, erhält automatisch das `CONTROL`-Recht auf dieser Tabelle.

db2s2e80.pdf,
S. 332

Beim Erstellen einer Tabelle wird auch ein Primärschlüssel angegeben. Ein Primärschlüssel besteht aus den angegebenen Attributen, die keine `NULL`-Werte zulassen dürfen und für die Einschränkungen bezüglich der möglichen Datentypen gelten. DB2 legt einen Index auf dem Primärschlüssel an. Jede Tabelle kann nur einen Primärschlüssel haben.

Fremdschlüssel werden benutzt, um zu erzwingen, dass ein oder mehrere Attribute einer abhängigen Tabelle nur Werte annehmen dürfen, die auch als Werte eines Schlüssels einer anderen Tabelle auftreten. Dieses Konzept wird auch Referentielle Integrität genannt. Der Benutzer muss das `REFERENCES`- oder ein übergeordnetes Recht auf der referenzierten Tabelle besitzen. Die Tabelle, auf die verwiesen wird, muss existieren und die referenzierten Attribute müssen dort als Schlüssel definiert sein.

```

CREATE TABLE employee (
  id          SMALLINT NOT NULL,
  name        VARCHAR(50),
  department  SMALLINT CHECK (department BETWEEN 1 AND 5),
  job         CHAR(10) CHECK (job IN
                             'Professor','WiMi','NichtWiMi'),
  hiredate    DATE,
  salary      DECIMAL(7,2),

  PRIMARY KEY (ID),

  CONSTRAINT yearsal CHECK (YEAR(hiredate) > 1996
                           OR SALARY > 1500)
)

```

Gelöscht werden Tabellen mit dem `DROP`-Statement. Dazu benötigt man das `CONTROL`-Recht für die Tabelle, das `DROPIN`-Recht im Schema der Tabelle, das `DBADM`- oder das `SYSADM`-Recht. Wird eine Tabelle gelöscht, so werden in allen Tabellen, die diese Tabelle referenzieren, die zugehörigen Fremdschlüsselbeziehungen gelöscht. Alle Indexe, die auf der Tabelle bestehen, werden gelöscht.

db2s2e80.pdf,
S. 512

Tabellen lassen sich auch direkt über das CC erstellen und löschen. Im Tabellenordner unterhalb einer Datenbank kann der Tabellen-Wizard über das Kontextmenü aufgerufen werden.

Aufgaben

- (a) Verschafft Euch zunächst Zugriff auf die existierende Datenbank IMDB. Diese enthält bereits viele Datensätze, die für unsere Fallstudie benötigt werden. Alle Gruppen haben Lesezugriff auf diese Datenbank. Geht dazu wie folgt vor:
- Öffnet das CC, und browsst hinunter auf die Ebene der Instanzen. Im Kontextmenü des *Instances*-Ordner wählt Ihr den Menüpunkt **Add**, um eine neue Instanz hinzuzufügen.
 - Ein Dialog öffnet sich. Hier wählt Ihr die *Discover*-Funktion. Diese sucht nach existierenden Instanzen. Die Beispieldatenbank liegt in der Instanz **db2inst**. Wählt diese an und bestätigt mit OK. Nun könnt Ihr dem Knoten noch einen eigenen Namen geben und die Instanz mit OK Euren katalogisierten Systemen hinzufügen.
 - Browsst nun innerhalb der Instanz **db2inst** auf die Datenbank-Ebene hinunter. Im Kontextmenü des *Databases*-Ordner wählt Ihr wieder **Add**, um eine der in dieser Instanz vorhandenen Datenbanken hinzuzufügen.
 - Im folgenden Dialog geht Ihr vor wie beim Hinzufügen einer Instanz und wählt diesmal die Datenbank IMDB.
- (b) Macht Euch mit den Funktionalitäten des CC vertraut. Öffnet die Beispieldatenbank IMDB und findet heraus, welche Nicht-Systemtabellen existieren. Setzt einen Filter, um die Systemtabellen (Schema **SYSIBM**) auszublenden. Über das Kontextmenü einer konkreten Tabelle lässt sich die Funktion *sample contents* aufrufen. Findet heraus, was in den Tabellen gespeichert ist.
- (c) Legt innerhalb Eurer Instanz eine neue Datenbank mit Namen **IMDBxx** an. (Ersetzt das **xx** durch Eure Gruppennummer.)
- (d) Erstellt in der neuen Datenbank eine Tabelle für Produktionen, Serien oder Filme (bzw. die Relation Eures Schemas, welche einer dieser Entitäten entspricht). Schreibt dazu das komplette **CREATE TABLE**-Statement auf. Vergesst nicht, Primärschlüssel anzugeben. Definiert auch mindestens einen sinnvollen *Check* für eines der Attribute. Benutzt anschließend entweder den CLP oder das *Command Center*, um die Tabelle anzulegen.
- (e*) Legt in der neuen Datenbank Tabellen für die restlichen Relationen Eurer Modellierung einer Film- und Serienminiwelt an. Nehmt dazu die in der letzten Woche definierten relationalen Schemata als Grundlage. Aus Zeitgründen könnt Ihr diese Tabellen mit Hilfe des Wizards anlegen.
- (f) Ändert die Zugriffsberechtigungen, so dass die Tutoren auf Eure Datenbank zugreifen und ein **SELECT** auf Euren Tabellen ausführen können. Die Tutoren gehören der Gruppe **dbtutors** an.

Die markierte (*) Aufgabe ist keine Präsenzaufgabe.