

## DB2 SQL, der Systemkatalog & Aktive Datenbanken

## Ziele

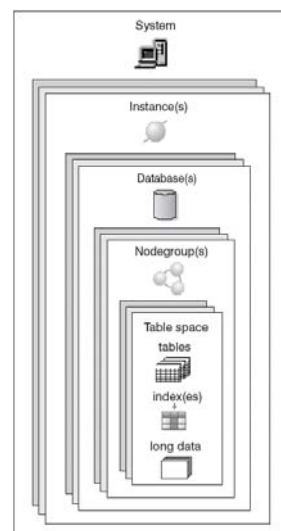
- Auf DB2 – Datenbanken zugreifen
- DB2 – Datenbanken benutzen
  - Abfragen ausführen
  - Den Systemkatalog benutzen
  - Import / Export von Daten
- Constraints kennen & benutzen
- Trigger kennen und benutzen

## Mit der DB2 verbinden

- Umgebung bereitstellen  
Bash: `source ~db2lehre/sql/lib/db2profile`  
C-Shell: `source ~db2lehre/sql/lib/db2cshrc`  
Windows: `db2cmd` aufrufen
- db2 – Kommandointerpreter starten  
`db2`  
`db2 -t SQL-Statement mit ; abschließen`
- Verbindung herstellen  
`connect to dbprak | terra | <datenbankname>`

## Struktur von DB2 – Datenbanken

- Eine Installation kann mehrere Instanzen haben
- Eine Instanz kann mehrere Datenbanken verwalten
- In einer Datenbank gibt es mehrere Schemata
- In jedem Schema gibt es DB-Objekte wie Tabellen, Views, ...



## Meine erste eigene Tabelle

- Tabelle anlegen  

```
create table person (  
  id integer not null primary key,  
  nachname varchar (50),  
  vorname varchar (50)  
)
```
- Werte einfügen  

```
insert into person values  
  (0, 'Meyer', 'Hans'),  
  (1, 'Müller', 'Franz')
```
- Tabelle abfragen  

```
select * from person
```

## Authentifizierung vs. Autorisierung

- Authentifizierung:
  - Feststellung der Identität des Nutzers, z.B. Name/Passwort
  - DB2-Authentifizierung über Betriebssystem-Mechanismen
- Autorisierung:
  - Feststellen, ob Zugriff auf Datenobjekte erlaubt ist
  - Unterschiedlichste Autorisierungsstufen, sehr ausgefeiltes Rechtemanagement

**GRANT SELECT ON <SCHEMA>.<TABNAME> TO USER <NAME>**

**GRANT CONNECT ON DATABASE TO USER <NAME>**

## Autorisierungen selbst ändern

- Schema-Rechte ändern  
*grant alterin | createin | dropin ON SCHEMA meyer  
TO user A | group B | public*
- DB2 – Hilfe: „GRANT (Schema privileges)“

## Der DB2 – Systemkatalog

- Alle Informationen über DB2-Objekte werden im Systemkatalog gehalten
- Systemkatalog folgt einem eigenen relationalen Modell im Schema syscat
- Die wichtigsten Tabellen:
  - syscat.tables
  - syscat.columns
- Struktur von Tables / Views:  
*describe table <tabellenname/viewname>*

## Import / Export

- EXPORT exportiert Daten. Syntax  
*EXPORT TO filename OF filetype*  
*SELECT \* FROM person*
- Filetype:
  - DEL: ASCII-delimited
  - WSF: Work Sheet Format (Lotus)
  - IXF: Integrated Exchange Format
    - Plattform-kompatibel
    - Enthält Daten zum Wiederherstellen der Tabelle

## Import / Export

- IMPORT importiert Daten. Syntax:  
*IMPORT FROM filename OF filetype*  
*CREATE INTO person*
- Siehe DB2-Hilfe.

## Aktive Datenbanken

- Datenintegrität sicherstellen
  - Foreign keys
  - Semantic data constraints
- Auf Schwellwerte reagieren
  - stop buy, stop loss – Aktienhandel
- Datenstruktur aktiv modellieren
  - Anlegen eines Nutzers legt auch seine Standardrechte / -rollen mit an
- Änderungen überwachen
  - Automatisch History von Änderungen protokollieren

## Constraints 1

- Sicherung der Datenintegrität
- Typen:
  - Primary key constraint: Definition des Primary Key
  - unique-constraint: Sicherung der Einzigartigkeit von Attributen (Unterschied zu primary key?)
  - referential-constraint: Sicherung der Beziehungen zu anderen Tabellen
  - check-constraint: Prüfung semantischer Integritätsregeln, z.B. Wertebereichsprüfung von Attributen
- „Sichtbarkeit“ für check-Constraints immer nur aktuelle Zeile einer Tabelle
- Immer wirksam: Implementierung eines Constraints auf einer den Constraint verletzenden Datenbasis ist nicht möglich!

## Constraints 2

- Syntax check-Constraint:

```
alter table <tablename>  
  add constraint <c-name> CHECK (condition)
```

- Beispiel

```
alter table gehalt  
  add constraint min_gehalt CHECK (gehalt>2000)
```

- Andere Constraint-Typen ähnlich; Details siehe DB2 – Hilfe

## Trigger

- Mächtiges Werkzeug zur Überwachung
- „reaktiv“: Wird durch bestimmte Ereignisse ausgelöst:
  - insert, update, delete
- Kaum Beschränkungen bezüglich der Trigger-Aktion
  - Insert/update/delete kann verändert oder zurückgewiesen werden
  - Zusätzliche Aktionen, wie z.B. Inserts / Updates / Deletes können angestoßen werden
  - Externe Routinen können ausgeführt werden
- Wirken erst, nachdem sie definiert wurden
- Syntax: Pflichtlektüre DB2-Hilfe

## Pointer in die Hilfe

- URL siehe dbprak-Homepage (Uni Jena)
  - <http://iibm01.inf.uni-jena.de/db2/index.htm>
- Stichworte (SQL Referenz):
  - create trigger
  - create table
  - alter table
  - signal sqlstate
  - grant

## Beispiel Constraints / Trigger

- Tabelle Gehalt(Name, Gehalt) erstellen
  - ```
CREATE TABLE GEHALT(  
    NAME VARCHAR(64) NOT NULL,  
    GEHALT INTEGER NOT NULL, PRIMARY KEY(NAME))
```
- Ein paar Daten laden (Import/Load)
  - ```
LOAD FROM file OF DEL MODIFIED BY COLDEL0x09  
INSERT INTO GEHALT(NAME, GEHALT)
```
- Constraint: Minimales Gehalt >2000
  - ```
ALTER TABLE GEHALT ADD CONSTRAINT min_gehalt  
CHECK (GEHALT > 2000)
```
- Testen ...
  - ```
INSERT INTO GEHALT VALUES('DORIS', 1000)
```
  - ```
INSERT INTO GEHALT VALUES('DORIS', 10000)
```



## Beispiel Constraints / Trigger

- Trigger: Budget für Gehälter max. 50000
  - CREATE TRIGGER BUDGET\_INSERT AFTER INSERT ON GEHALT  
FOR EACH ROW MODE DB2SQL  
WHEN ((SELECT SUM(GEHALT) FROM GEHALT) > 50000)  
SIGNAL SQLSTATE '75001' ('Budget von 50.000 ueberschritten')
  - CREATE TRIGGER BUDGET\_UPDATE AFTER UPDATE ON GEHALT ...
- Testen ...
  - INSERT INTO GEHALT VALUES('Katrin', 5000)
  - INSERT INTO GEHALT VALUES('Sylvia', 10000)

## TERRA

- **Achtung!** Qualität der Daten entspricht nicht immer den Erwartungen
- Beispiele
  - Städte (STADT) kommen mehrfach vor (z.B. Oslo)
  - Städte in LIEGT\_AN, die weder an einem Meer, See oder Fluss liegen
  - In Katar leben mehr Menschen in Städten, als im ganzen Land
  - ...

## TERRA (cont.)

```
SELECT * FROM TERRA.LAND WHERE L_ID = 'N'
```

| NAME         | L_ID | LT_ID | EINWOHN... | LAENGE | BREITE |
|--------------|------|-------|------------|--------|--------|
| Oslo         | N    | OS    | 449337     | 11     | 59     |
| Oslo         | N    | AK    | 449337     | 11     | 59     |
| Moss         | N    | OES   | 24517      | 10     | 59     |
| Hamar        | N    | HE    | 15685      | 11     | 60     |
| Lillehammer  | N    | OP    | 22118      | 10     | 61     |
| Drammen      | N    | BU    | 50855      | 10     | 59     |
| Toensberg    | N    | VE    | 8984       | 10     | 59     |
| Skien        | N    | TE    | 47010      | 9      | 59     |
| Arendal      | N    | AA    | 12174      | 8      | 58     |
| Kristiansand | N    | VA    | 62640      | 8      | 58     |
| Stavanger    | N    | RO    | 95089      | 5      | 59     |
| Bergen       | N    | HO    | 207916     | 5      | 60     |
| Hermannsverk | N    | SF    | 706        | 6      | 61     |
| Molde        | N    | MR    | 21448      | 7      | 62     |
| Trondheim    | N    | ST    | 134426     | 10     | 63     |
| Steinkjer    | N    | NT    | 20480      | 11     | 64     |
| Bodoe        | N    | NO    | 34479      | 14     | 67     |
| Tromsøe      | N    | TR    | 48109      | 19     | 69     |
| Vadsøe       | N    | FI    | 5961       | 29     | 70     |
| Hammerfest   | N    | FI    | 7089       | 23     | 70     |
| Svolvær      | N    | NO    | 4500       | 14     | 68     |
| Narvik       | N    | NO    | 18754      | 17     | 68     |

**PROBLEM:**

NAME, L\_ID, LT\_ID →  
EINWOHNER, LAENGE, BREITE

NAME, L\_ID → EINWOHNER

**FRAGE:**

*Wie viele Menschen leben in norwegischen Städten?*

## Ziele

- Auf DB2 – Datenbanken zugreifen
- DB2 – Datenbanken benutzen
  - Abfragen ausführen
  - Den Systemkatalog benutzen
  - Import / Export von Daten
- Constraints kennen & benutzen
- Trigger kennen und benutzen