Leibniz Universität IT Services

Leibniz Universität Hannover

# S(tructured)Q(uery)L(anguage)
# Neue Datenbank erstellen

# Was ist eine „Datenbank“?

- Strukturierte Verwaltung und Sammlung von großen Datenmengen.
- Container für eine beliebige Anzahl von Tabellen.
- Abbildung von Listen, bei denen in einem Tabellenkalkulationsprogramm horizontal zur Seite geblättert werden muss.
- Speicherung auf einem SQL-Server. Ablage als Datei in einem bestimmten Format (wie zum Beispiel *.db für eine SQLite-Datei).

# Relationales Datenbankmodell

- Entwicklung in den 70er Jahren.
- In Tabellen (Relationen) werden spaltenweise Attribute von Dingen abgelegt.
- Jede Zeile in einer Tabelle beschreibt ein Ding. Die Beschreibung wird in einem Datensatz abgelegt.
- Jedes Datenfeld in einer Tabelle enthält einen Attribut-Wert. Jeder Datensatz unterscheidet sich in mindestens einem Wert von allen anderen.
- Jedes Ding kann eine Beziehung zu einem anderen Ding haben. Die Beziehungen werden über Schlüsselfelder abgebildet.

# Neue Datenbank anlegen

```
postgres =# CREATE DATABASE "dbsLandKontinent"
postgres =# WITH OWNER = postgres
postgres =# ENCODING = 'UTF8'
postgres =# TABLESPACE = pg_default
postgres =# CONNECTION LIMIT = -1;
```

- Jede SQL-Anweisung endet mit dem Semikolon.
- Jede SQL-Anweisung beginnt mit einem Verb, welches die gewünschte Aktion beschreibt.
- In diesem Beispiel wird mit Hilfe von `CREATE DATABASE` eine neue Datenbank erzeugt.

# Neue Datenbank

```
postgres =# CREATE DATABASE "dbsLandKontinent";
```

- Dem Befehl `CREATE DATABASE` folgt der Name der Datenbank.
- Der Name unterliegt den Regeln zu einem Verzeichnisnamen des verwendeten Betriebssystems sowie zu den Bezeichner in SQL.

# Regeln zu Bezeichnern

- Der Bezeichner beginnt mit einem Buchstaben.
- Ein Bezeichner besteht aus den lateinischen Groß- und Kleinbuchstaben und den Ziffern.
- Ein Bezeichner enthält als Sonderzeichen nur den Unterstrich.
- Der Schrägstrich „/“ ist in einem Datenbank-Namen nicht erlaubt.
- Ein Datenbank-Name ist maximal 64 Zeichen lang.
- SQL-Befehle können nicht als Bezeichner genutzt werden.

# Hinweis

- Sonderzeichen wie Umlaute verursachen in einer SQL-Anweisung keinen Fehler.
- Aber: Die Meta-Befehle von psql verarbeiten nur Buchstaben aus dem ASCII-Zeichensatz, Ziffern und den Unterstrich.

# Groß- und Kleinschreibung

```
postgres =# CREATE DATABASE "dbsLandKontinent";
postgres =# CREATE DATABASE dbsLandKontinent;
```

- Bezeichner können durch Anführungsstriche begrenzt werden.
- Wenn der Bezeichner nicht durch Anführungsstriche begrenzt wird, wird die Groß- und Kleinschreibung nicht beachtet.

# Beispiel

```
postgres=#
postgres=# CREATE DATABASE "dbsLandKontinent";
CREATE DATABASE
postgres=# CREATE DATABASE dbsLandKontinent;
CREATE DATABASE
postgres=# \l
                                                  Liste der Datenbanken
      Name        | Eigent³mer | Kodierung | Sortierfolge |
fsprivilegien
------------------+------------+-----------+--------------+
--------------
 chinook          | postgres   | UTF8      | C            |
 dbsLandKontinent | postgres   | UTF8      | C            |
 dbslandkontinent | postgres   | UTF8      | C            |
 northwind        | postgres   | UTF8      | C            |
 postgres         | postgres   | UTF8      | C            |
 template0        | postgres   | UTF8      | C            |
gres           +
                  |            |           |              |
s=CTc/postgres
 template1        | postgres   | UTF8      | C            |
gres           +
                  |            |           |              |
s=CTc/postgres
(7 Zeilen)
```

# Besitzer der Datenbank

```
postgres =# CREATE DATABASE "dbsLandKontinent"
postgres =# WITH OWNER = postgres;
```

```
postgres =# CREATE DATABASE "dbsLandKontinent"
postgres =# OWNER DEFAULT;
```

- Optional kann der Besitzer der Datenbank bei der Erstellung angegeben werden.

# … in PostgreSQL

- In beiden Beispiel wird ein Besitzer `postgres` erzeugt.
- Der Standardbesitzer `postgres` hat die Rolle Superuser. Nur der Superuser kann Datenbanken anlegen.
- Mit Hilfe von `postgres =# \du` können alle Besitzer zu einer Datenbank aufgelistet werden.

# Zeichenkodierung

```
postgres =# CREATE DATABASE "dbsLandKontinent"
postgres =# ENCODING = 'UTF8'
```

- Dem Attribut `ENCODING` wird der Name der gewünschten Zeichenkodierung für die Datenbank übergeben. In diesem Beispiel wird UTF8 für die Zeichenkodierung genutzt.
- PostgresSQL nutzt als Standardwert, die Zeichenkodierung der Datenbankvorlage.

# Anführungszeichen oder Apostroph?

```
postgres =# CREATE DATABASE "dbsLandKontinent"
postgres =# ENCODING = 'UTF8'
```

- In SQL-Anweisung werden Zeichenketten (Strings) durch Apostrophs begrenzt. Das Attribut, das Datenfeld erwartet einen Wert vom Type „Text“.
- Mit Hilfe der Anführungszeichen als Begrenzung bei Datenbank-Namen, Feldnamen etc. wird die Groß- und Kleinschreibung in der Bezeichnung erzwungen.

# Tablespace

```
postgres =# CREATE DATABASE "dbsLandKontinent"
postgres =# TABLESPACE = pg_default
```

```
postgres =# CREATE DATABASE "dbsLandKontinent"
postgres =# TABLESPACE DEFAULT;
```

- Logischer Speicherort für Tabellen.
- Kontrollierte Ablage von Tabellen in einer Datenbank.

# … in PostgreSQL

- `pg_Default` bezeichnet den Standardablageort für Tabellen und Indizies in PostgreSQL.
- Mit Hilfe von `postgres =# \l+` können alle Datenbanken und deren Tablespaces auf dem Server aufgelistet werden.

# Anzahl der Verbindungen zur Datenbank

```
postgres =# CREATE DATABASE "dbsLandKontinent"
postgres =# CONNECTION LIMIT = -1;
```

- Mit Hilfe des optionalen Attributs `CONNECTION LIMIT` können die maximalen Verbindungen zu einer Datenbank festgelegt werden.
- Der Standardwert `-1` ermöglicht unbegrenzt viele Verbindungen zu einer Datenbank.

# Auswahl einer Datenbankvorlage

```
postgres =# CREATE DATABASE "dbsLandKontinent"
postgres =# TEMPLATE = template0;
```

- Welche Datenbankvorlage wird für die neu zu erstellende Datenbank genutzt?
- Als Vorlage für eine PostgreSQL-Datenbank wird standardmäßig `template1` genutzt.

# Eingabegebietsschema

- Attribute, die mit `LC_` beginnen, beziehen sich immer auf ein Eingabegebietsschema.
- Länderspezifische Einstellungen wie Sortierreihenfolgen etc.
- Die Einstellungsmöglichkeiten sind abhängig vom Angebot des genutzten Betriebssystems.
- Einstellungen zum Eingabegebietsschema arbeiten nur mit der Datenbankvorlage `template0` zusammen.
- Attribute in PostgreSQL: https://www.postgresql.org/docs/9.6/static/locale.html

# Sortierreihenfolge

```
postgres =# CREATE DATABASE "dbsLandKontinent"
postgres =# TEMPLATE = template0
postgres =# LC_COLLATE = 'German_Germany.1252';
```

- Das optionale Attribut `LC_COLLATE` legt fest, wie Datenfelder sortiert werden.
- In diesem Beispiel wird der Zeichensatz „ISO 8859-1“ oder „Windows 1252“ für westeuropäische Länder genutzt.
- Unterstützte Zeichensätze in PostgreSQL: https://www.postgresql.org/docs/9.6/static/multibyte.html

# Vergleich von Zeichen

```
postgres =# CREATE DATABASE "dbsLandKontinent"
postgres =# TEMPLATE = template0
postgres =# LC_CTYPE = 'German_Germany.1252';
```

- Das Attribut `LC_CTYPE` legt fest, wie Zeichen bei einem Vergleich interpretiert werden.
- Zeichenklassifizierung und deren Umwandlung.
- In diesem Beispiel wird der Zeichensatz „ISO 8859-1“ oder „Windows 1252“ für westeuropäische Länder genutzt.

# Datenbanken auf den PostgreSQL-Server

- Mit Hilfe von `postgres =# \l` können alle Datenbanken auf dem Server aufgelistet werden.
- Die Datenbanken `template0` und `template1` sind Datenbankvorlagen auf einem PostgreSQL-Server.

# Öffnen von Datenbanken

| | | | |
|---|---|---|---|
| \c | | postgres | postgres |
| \c | | dbslandkontinent | |
| \c | | "dbsLandKontinent" | |
| \c | | Datenbank | Benutzer |

- Der Client psql verbindet mit Hilfe von `\c` einen Benutzer mit einer Datenbank.
- Wenn keine Angaben zum Benutzer gemacht werden, wird die Datenbank mit Administratorrechten geöffnet.

# Schließen von eigenen Datenbanken

| | | | |
|---|---|---|---|
| \c | | postgres | postgres |

- Durch das Öffnen einer anderen Datenbank wird die momentan geöffnete automatisch geschlossen.

# Löschen von Datenbanken

```
postgres =# CREATE DATABASE "dbsLandKontinent";
postgres =# DROP DATABBASE "dbsLandKontinent";
```

- Mit Hilfe des Befehls `DROP DATABASE` wird eine Datenbank vom Server gelöscht.
- Die Anweisung muss mit einem Semikolon abgeschlossen werden. Andernfalls wird die Datenbank nicht gelöscht.

# Hinweise

```
postgres =# DROP DATABBASE dbslandkontinent;
postgres =# DROP DATABBASE "dbsLandKontinent";
```

- Der Datenbankname muss exakt der Bezeichnung auf dem Server entsprechen.
- Die Groß- und Kleinschreibung von Namen muss beachtet werden.
- Vor dem Löschen muss die Datenbank geschlossen werden.

# Umbenennung der Datenbank

```
postgres =# ALTER DATABBASE "Flüsse"
postgres =# RENAME TO fluesse;
```

- Der Befehl `ALTER DATABASE` verändert eine Datenbank. In diesem Beispiel wird die Datenbank `"Flüsse"` bearbeitet.
- Mit Hilfe des Befehls `RENAME TO` wird die zu verändernde Datenbank umbenannt. Die neue Bezeichnung steht rechts vom Befehl.