



Datenbanken erstellen mit PostgreSQL

S. Baldes

Datenbank erstellen

Verbindung mit DBMS aufnehmen

- **psql -h localhost postgres postgres**
 - **Passwort postgres**
 - **Erstellt leere Datenbank**
- **\c postgres** //verbindet mit DB postgres
- **\l** //list db: gibt Liste mit allen Datenbanken aus
- **\dt** //describe tables: gibt alle Tabellen aus
- **\d <tabelle>** //beschreibt Struktur der Tabelle
- **\i <drag&drop>** //Datei einlesen
 - z.B. **\i myDatabase.sql**
- **\?** //help

Datenbank erstellen

DB in SQL-Datei erstellen

myDB.sql

-- Erzeuge die Tabelle test

drop table test;

CREATE TABLE test (name varchar(30), alter integer);

INSERT into test (name,alter) VALUES ('Tom',20);

INSERT into test (name,alter) VALUES ('Tina',18);

select * from test;

DBs an der Wara

- `psql -h localhost postgres postgres Passwort: postgres //leere DB`
 - oder `psql -h localhost postgres postgres -f <eigene datei>.sql`
- `psql -h localhost bundesliga postgres //Fußball password postgres`
- `psql -h localhost mondial postgres Passwort: postgres //Umfangreiche Geographie-DB`

DB anlegen

Create Table, Primary Key

```
CREATE TABLE Kunden(  
  KundenNr INTEGER,  
  Anrede VARCHAR(20),  
  Name VARCHAR(30),  
  Vorname VARCHAR(30),  
  eMail VARCHAR(50),  
  Passwort VARCHAR(8),  
  KundeSeit DATE,  
  CONSTRAINT PK_Kunden PRIMARY KEY (KundenNr));
```

DB anlegen Foreign Key

```
CREATE TABLE Bestellung(  
    BestellNr INTEGER NOT NULL,  
    KundenNr INTEGER NOT NULL,  
    Bestelldatum TIME NOT NULL,  
    CONSTRAINT PK_Bestellung PRIMARY KEY (BestellNr),  
    CONSTRAINT FK_Ku_Be FOREIGN KEY(KundenNr)  
        REFERENCES Kunden(KundenNr));
```

DB anlegen

Kurzform Primary & Foreign Key

```
CREATE TABLE Bestellung(  
  BestellNr INTEGER PRIMARY KEY,  
  KundenNr INTEGER REFERENCES Kunden(KundenNr),  
  Bestelldatum TIME NOT NULL);
```

Datentypen

Datentyp	Beschreibung
Integer, Int, Real, Float, Numeric	Numerischer Datentypen
Char(n) Varchar(n) Text	Text mit fester Länge (1-255) Text mit variabler Länge (1-4056) Variable Länge ohne Höchstgrenze
Date Time Timestamp	Datum, z.B. '2015-12-24' oder '24.12.2015' oder '12/24/2015' Uhrzeit, z.B. '22:9:59' Datum und Uhrzeit, z.B. '2015-12-24 22:9:59'
Boolean	True oder False
...	

Constraints

➤ PRIMARY KEY

- `CONSTRAINT PK_01 PRIMARY KEY (BestellNr)` oder
- `BestellNr PRIMARY KEY` // direkt bei der Spaltendefinition
- `CONSTRAINT PK_02 PRIMARY KEY (BestellNr, ArtikelNr)` // mehrere Spalten

➤ FOREIGN KEY

- `CONSTRAINT FK_01 FOREIGN KEY(KundenNr) REFERENCES Kunden(KundenNr)` oder
- `KundenNr INTEGER REFERENCES Kunden(KundenNr)` // direkt bei der Spaltendefinition

➤ NOT NULL

- `BestellNr INTEGER NOT NULL`

➤ DEFAULT

- `Erscheinungstermin DATE DEFAULT CURRENT_DATE`

➤ UNIQUE

- `ISBN VARCHAR(17) UNIQUE`

➤ CHECK

- `Statuscode CHAR(02) CHECK(Statuscode IN('AV','OP','IP','OR'))`

➤ AUTO_INCREMENT

- `ID INTEGER AUTO_INCREMENT` // in einigen DB wie MySQL
- `ID SERIAL` // in Postgres

Tabellen nachträglich ändern

- **ALTER TABLE** Bestellung **ADD**
Bestand INTEGER;
- **ALTER TABLE** Bestellung **ALTER COLUMN**
Bestand VARCHAR(100);
- **ALTER TABLE** Bestellung **DROP COLUMN**
Bestand;
- **ALTER TABLE** Bestellung **ADD CONSTRAINT**
FK_BEST_KUNDE FOREIGN KEY (KundenNr)
REFERENCES Kunden(KundenNr);
- **ALTER TABLE** Bestellung **DROP CONSTRAINT**
FK_BEST_KUNDE;

Datensätze einfügen

INSERT INTO

- Kompletten Datensatz (Zeile) eintragen:

INSERT INTO tabellenName

VALUES (wert1, wert2, ...);

- Nur Daten für einzelne Spalten eintragen

INSERT INTO tabellenName (spalte1, spalte2, ...)

VALUES (wert1, wert2, ...);

Beispiel: `INSERT INTO KundenDaten (ID, Name) VALUES (23412, 'Stegleiter');`

Zu Beachten:

- Text/Datum in einfache Anführungszeichen: 'Tom', ' 2015-12-24'
- Kommentare Einzeilig `--<space>` Mehrzeilig `/* ... */`

Datensätze ändern

UPDATE

Den Datensatz (Zeile) mit spalte=wert verändern:

UPDATE tabellenName

SET spalte1=wert1, spalte2=wert2, spalte3=wert3,...

WHERE spalte=wert;

Beispiel

```
UPDATE Kunde  
SET Name='Schmidt', City='Hamburg'  
WHERE KundeID=5342;
```

VORSICHT:

➤ Ohne oder mit nicht-eindeutiger WHERE-Klausel werden zuviele Datensätze verändert (evt. alle).

Datensätze löschen

DELETE FROM

Den Datensatz (Zeile) mit spalte=wert löschen:

DELETE FROM tabellenName

WHERE spalte=wert;

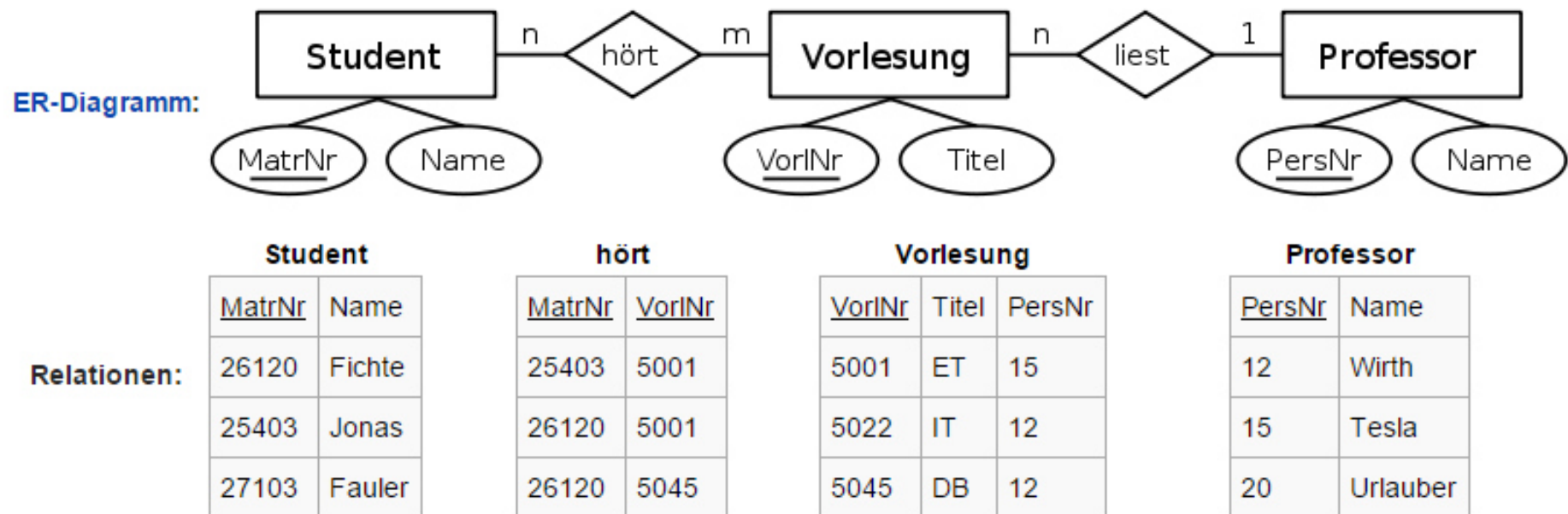
Beispiel

```
DELETE FROM Kunde  
WHERE Name='Schmidt, AND City='Hamburg';
```

Übung 1



Erstellen Sie eine SQL-Datei für folgende Datenbank:



Dokumentationen

- TEIA Lehrbuch:
<http://www.teialehrbuch.de/Kostenlose-Kurse/Datenbankentwicklung-mit-PostgreSQL-9/>
- W3schools.com:
<http://www.w3schools.com/sql/default.asp>
- Tutorial:
<http://workshop-postgresql.de/>