

Grundlagen Datenbanken

Benjamin Wagner

15. November 2018



TUM Uhrenturm

Allgemeines

- Folien von mir sollen unterstützend dienen. Sie sind nicht von der Übungsleitung abgesegnet und haben keinen Anspruch auf Vollständigkeit (oder Richtigkeit).
- Bei Fragen: wagnerbe@in.tum.de
- Vorlesungsbegleitendes Buch von Professor Kemper (Chemiebib)
- Mein Foliensatz ist online: <https://github.com/wagjain/GDB2018>

Wiederholung: Relationen

Professoren:

PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
...

- Jede Tabelle hat Spalten: **Attribute**
- Die einzelnen Zeilen nennt man: **Tupel**
- Jede Spalte hat einen: **Typ**
- **Schlüssel** markieren ein Tupel eindeutig

SQL

- Standard Anfragesprache für relationale Datenbanken
- Web-Interface: <http://hyper-db.de/interface.html>
- Möglichkeit, Anfragen auf Uni-Schema zu realisieren
- Läuft auf Hyper (Datenbank des Lehrstuhls)
- Grundstruktur einer SQL-Anfrage:

1 **SELECT** . . .

2 **FROM** . . .

3 **WHERE** . . .

SQL - Datentypen

- Es gibt eine Reihe von Datentypen in SQL
- Z.B: char(n), varchar(n), integer, blob, date ...
- Damit können Tabellen erstellt werden:

```
1 CREATE TABLE Customers (  
2     CustId      integer not null ,  
3     Name        varchar(30) not null ,  
4     Birthday    date  
5 );
```

SQL - Einfache Anfrage

- Suche Namen aller Professor*innen, deren Rang C4 ist

SQL - Einfache Anfrage

- Suche Namen aller Professor*innen, deren Rang C4 ist

```
1      SELECT Name
2      FROM Professoren
3      WHERE Rang = 'C4'
```

- Suche alle Studierenden, die seit mehr als vier Semestern studieren

SQL - Einfache Anfrage

- Suche Namen aller Professor*innen, deren Rang C4 ist

```
1      SELECT Name
2      FROM Professoren
3      WHERE Rang = 'C4'
```

- Suche alle Studierenden, die seit mehr als vier Semestern studieren

```
1      SELECT *
2      FROM Studenten
3      WHERE Semester > 4
```


SQL - Sprachkonstrukte

- **Kreuzprodukt von Relationen** - *from R1, R2*
- **Aufgabe:** Was ist der Name, des Professors, der 'Ethik' liest

SQL - Sprachkonstrukte

- **Kreuzprodukt von Relationen** - *from R1, R2*
- **Aufgabe:** Was ist der Name, des Professors, der 'Ethik' liest

```
1      SELECT Professoren.Name
2      FROM Professoren, Vorlesungen
3      WHERE Professoren.persNr
4              = Vorlesungen.gelesenVon
5      AND Vorlesungen.Titel = 'Ethik'
```

SQL - Sprachkonstrukte

- **Duplikateliminierung** - *select distinct*
- **Aufgabe:** Suche das Semester aller Studierenden, die Logik hören

SQL - Sprachkonstrukte

- **Duplikateliminierung** - *select distinct*
- **Aufgabe:** Suche das Semester aller Studierenden, die Logik hören

```
1      SELECT DISTINCT studenten.semester
2      FROM studenten, hoeren, vorlesungen
3      WHERE studenten.matrnr = hoeren.matrnr
4      AND hoeren.vorlNr = vorlesungen.vorlNr
5      AND vorlesungen.titel = 'Logik'
```

SQL - Sprachkonstrukte

- **Relation benennen** - *from Professoren p1, Professoren p2*
- **Mengenoperationen** - *union, intersects, minus*
- **Quantor** - *exists*

```
1      (SELECT p.Name
2         FROM Professoren p
3         WHERE NOT EXISTS (
4             SELECT *
5             FROM Vorlesungen v
6             WHERE v.gelesenVon = p.persNr);)
7
8      INTERSECT
9
10     (...)
```