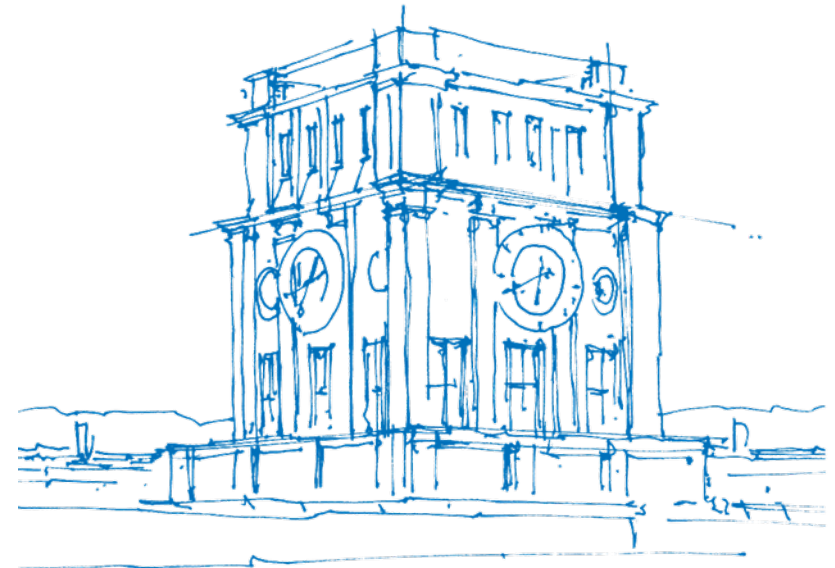


Grundlagen Datenbanken

Benjamin Wagner

22. 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>

SQL - Sprachkonstrukte

- **Gruppierung** - *group by*
- Bildet Gruppen von Tupeln mit den selben Werten in den Attributen der "group by" Klausel
- Auf den anderen Attributen können dann Aggregatsfunktionen aufgerufen werden
- **Aufgabe:** Wie viele Studenten studieren in welchem Semester?

SQL - Sprachkonstrukte

- **Gruppierung** - *group by*
- Bildet Gruppen von Tupeln mit den selben Werten in den Attributen der "group by" Klausel
- Auf den anderen Attributen können dann Aggregatsfunktionen aufgerufen werden
- **Aufgabe:** Wie viele Studenten studieren in welchem Semester?

```
1      SELECT semester , count (*)  
2      FROM Studenten  
3      GROUP BY semester
```

SQL - Sprachkonstrukte

- **Gruppierung** - *group by*
- Es gibt viele Aggregatsfunktionen: avg, max, min, count, sum
- Für Selektion auf Aggregaten: *having*
- **Aufgabe:** Welche Professoren halten mehr als 2 Vorlesungen?

SQL - Sprachkonstrukte

- **Gruppierung** - *group by*
- Es gibt viele Aggregatsfunktionen: avg, max, min, count, sum
- Für Selektion auf Aggregaten: *having*
- **Aufgabe:** Welche Professoren halten mehr als 2 Vorlesungen?

```
1      SELECT p.Name , count (*)
2      FROM Professoren p, Vorlesungen v
3      WHERE p.persNr = v.gelesenVon
4      GROUP BY v.gelesenVon
5      HAVING count (*) > 2
```

SQL - Sprachkonstrukte

- **Temporäre Relation** - *with ... as()*
- Komplexe Anfragen können u.U. modularisiert werden

```
1      WITH h AS (SELECT VorlNr ,
2                      count(*) AS AnzProVorl
3                      FROM hoeren
4                      GROUP BY VorlNr) ,
5
6      . . .
```

SQL - Sprachkonstrukte

- **String Vergleiche** - *like ...*
- `'_'` dient als Placeholder für ein Zeichen
- `'%'` dient als Placeholder für beliebig viele Zeichen

```
1      SELECT *  
2      FROM Studenten  
3      WHERE name like 'T%eophrastos';
```


SQL - Sprachkonstrukte

- **Fallunterscheidung** - *case when ...*
- Die erste passende Bedingung wird ausgewertet

```
1      SELECT MatrNr , (CASE
2          WHEN Note < 1.5 THEN 'sehr_gut'
3          WHEN Note < 2.5 THEN 'gut'
4          WHEN Note < 3.5 THEN 'befriedigend'
5          WHEN Note < 4.0 THEN 'ausreichend'
6          ELSE 'nicht_bestanden'
7          END)
8      FROM prüfen;
```