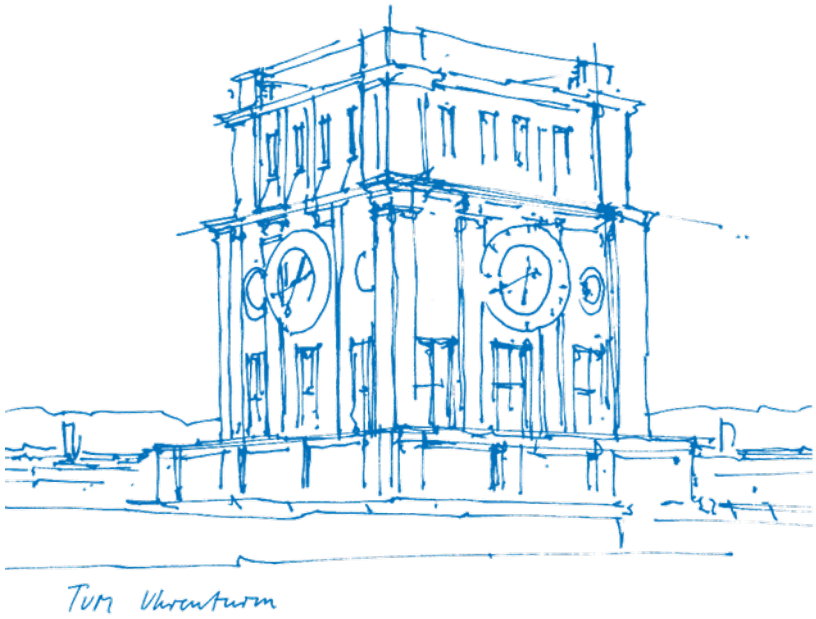
Grundlagen Datenbanken

Benjamin Wagner

29. November 2018



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/wagjamin/GDB2018`

SQL - Rekursion

- Unsere bisherigen Mittel reichen nicht ganz aus
- Beispiel: finde alle direkten und indirekten Vorgänger einer Vorlesung
- Hier hilft Rekursion
- **Idee:** definiere rekursiv eine Tabelle mit *with ... as*
- Nutze diese dann ganz normal weiter

SQL - Rekursion

- Rekursive Vorgänger-Nachfolger Relation
- Wir sehen: die Relation darf im *SELECT...* Teil verwendet werden

```
1 WITH RECURSIVE TransVorl(Vorg, Nachf) AS
2   (SELECT Vorgaenger, Nachfolger
3   FROM voraussetzen
4   UNION ALL
5   SELECT t.Vorg, v.Nachfolger
6   FROM TransVorl t, Voraussetzen v
7   WHERE t.Nachf = v.Vorgaenger)
```

SQL - Rekursion

- Und dann? Wir benutzen TransVorl ganz normal weiter...

```
1 SELECT Titel FROM Vorlesungen
2 WHERE VorlNr IN
3  (SELECT Vorg
4   FROM TransVorl where Nachf IN
5   (SELECT VorlNr FROM Vorlesungen
6    WHERE Titel= 'Der␣Wiener␣Kreis'))
```

Datenintegrität

- Wissen schon:

  Wie kann ich Schemata modellieren?

  Wie kann ich Anfragen an meine Datenbank formulieren?
- **Jetzt:** Wie stelle ich Korrektheit der Daten sicher?
- **Beispiel:** in einer Relationship soll immer auf einen existierenden Schlüssel verwiesen werden

Datenintegrität

- **Kandidatenschlüssel:** *unique*
- **Primärschlüssel:** *primary key*
- **Attribut darf nicht NULL sein:** *NOT NULL*
- **Referenz:** *references*

```
1 CREATE TABLE Studenten(
2 matrNr INTEGER PRIMARY KEY, (...)
3 );
4 CREATE TABLE Studentenausweis(
5 besitzer INTEGER REFERENCES Studenten,
6 (...))
```

Datenintegrität

- Was, wenn Referenzen gelöscht/geändert werden?
- **Änderung übernehmen:** *on update/delete cascade*
- **Referenz NULL setzen**: *on update/delete set null*

```
1 CREATE TABLE Studentenausweis(
2 besitzer INTEGER REFERENCES Studenten
3                  ON DELETE SET NULL,
4 (...))
```

Datenintegrität

- Es können kompliziertere Konsistenzbedingungen gefordert werden
- **Bedingung:** *check(...)*
- Wird vor Änderung am Datenbestand geprüft

```
1 CREATE TABLE Studentenausweis(
2 besitzer INTEGER REFERENCES Studenten
3          ON DELETE SET NULL,
4 CHECK(besitzer != 0)
5 (...))
```