

# Wiederholung VU Datenmodellierung

## VL Datenbanksysteme

Ingo Feinerer

Arbeitsbereich Datenbanken und Artificial Intelligence  
Institut für Informationssysteme  
Technische Universität Wien

# Kurzer Rückblick und Ausblick

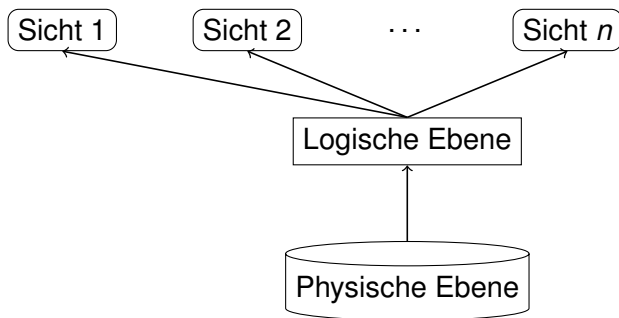
- ▶ Einführung: (relationale) DB-Management Systeme
- ▶ Datenbankentwurf, ER-Modell
- ▶ Relationales Datenmodell
- ▶ Datenintegrität
- ▶ Relationale Entwurfstheorie

# Motivation für den Einsatz eines DBMS

Typische Probleme bei Informationsverarbeitung ohne DBMS:

- ▶ Redundanz und Inkonsistenz
- ▶ Beschränkte Zugriffsmöglichkeiten
- ▶ Probleme beim Mehrbenutzerbetrieb
- ▶ Verlust von Daten
- ▶ Integritätsverletzung
- ▶ Sicherheitsprobleme
- ▶ Hohe Entwicklungskosten für Anwendungsprogramme

# Die Abstraktionsebenen eines Datenbanksystems



Datenunabhängigkeit:

- ▶ physische Datenunabhängigkeit
- ▶ logische Datenunabhängigkeit

# ER Diagram

Uni

Studenten

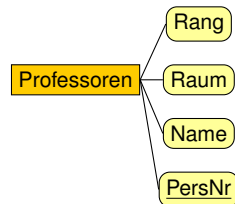
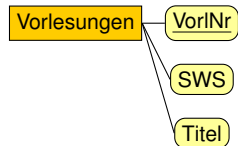
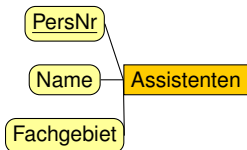
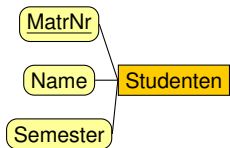
Vorlesungen

Professoren

Assistenten

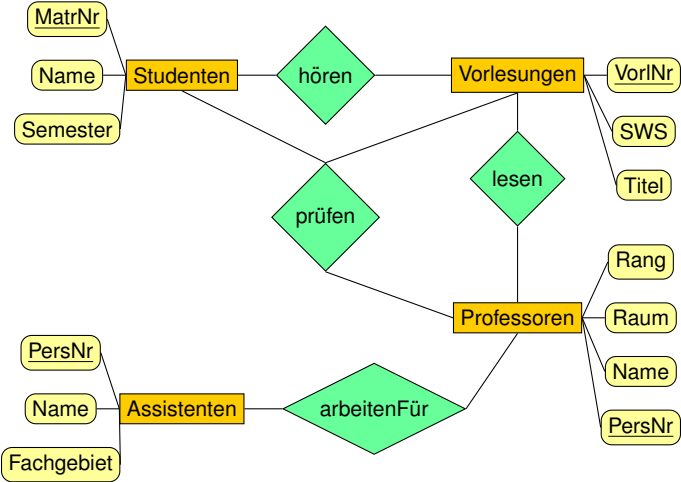
# ER Diagram

Uni



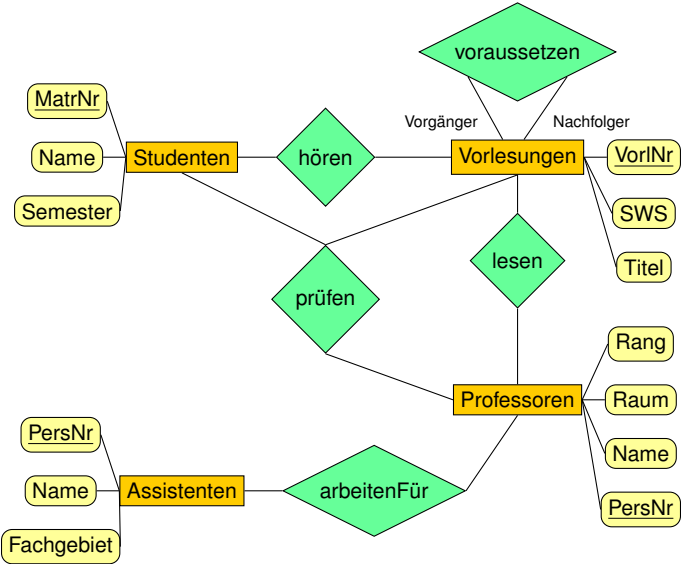
# ER Diagram

Uni



# ER Diagram

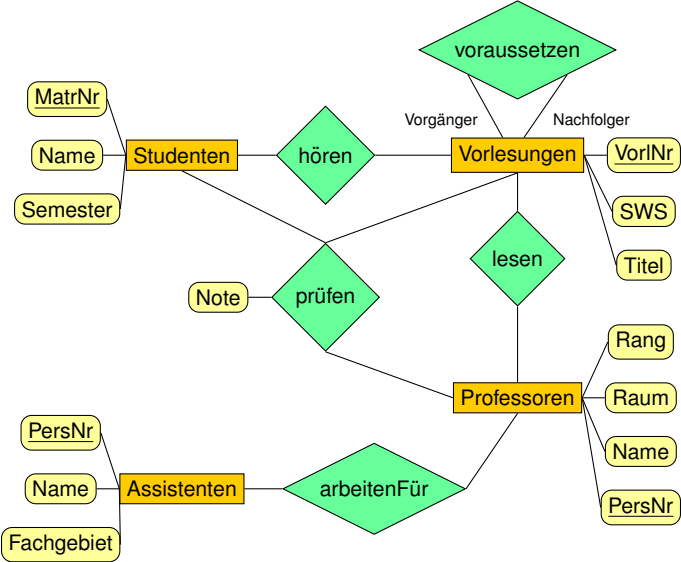
Uni





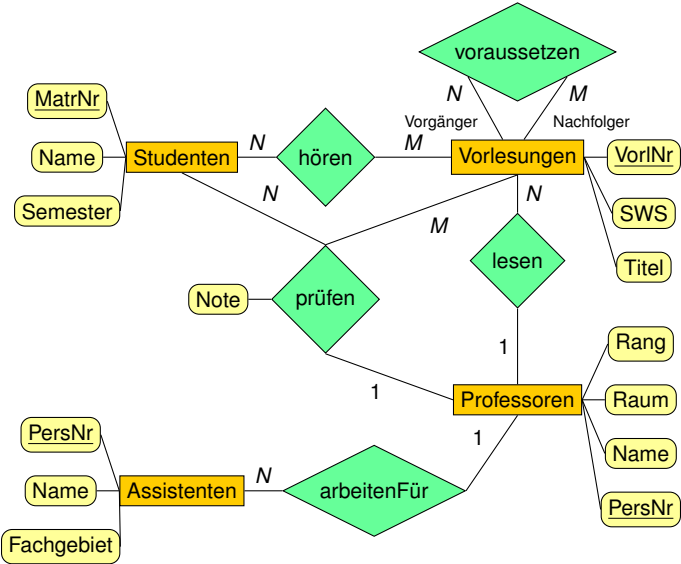
# ER Diagram

Uni



# ER Diagram

Uni



# Relationales Datenmodell

- ▶ Idee: Speicherung der Information ausschließlich in *Tabellen* (d.h. in *Relationen*)
- ▶ Schreibweise der Tabellen: Angabe des Tabellennamen, der Spaltennamen, der Datentypen und des Primärschlüssels

## Beispiel

Studenten: {[MatrNr: integer, Name: string, Semester: integer]}

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
...	...	...
29120	Theophrastos	2
29555	Feuerbach	2

# Relationale Anfragesprachen

Formale Grundlagen:

- ▶ Relationale Algebra: Operationen auf Mengen
- ▶ Relationaler Tupelkalkül: Logische Formeln, Variablen stehen für Tupel, d.h.: ganze Zeile einer Tabelle
- ▶ Relationaler Domänenkalkül: Logische Formeln, Variablen stehen für Wert aus Domäne, d.h.: einzelner Eintrag in Tabelle

Wichtigste Anfragesprache in der Praxis:

- ▶ SQL

# SQL

- ▶ DDL (Datendefinitionssprache): Anlegen einer Tabelle, einer View, eines Triggers, etc.

## Beispiel

```
CREATE TABLE Professoren  
  (PersNr INTEGER NOT NULL,  
   Name VARCHAR(30) NOT NULL,  
   Rang CHARACTER(2))
```

- ▶ DML (Datenmanipulationssprache): Einfügen, ändern, löschen

## Beispiel

```
INSERT INTO Studenten  
VALUES (28121, 'Archimedes', 2)
```

- ▶ Anfragesprache (Query)

# SQL Anfragen

- ▶ Einfache Anfragen auf einer Tabelle

## Beispiel

```
SELECT PersNr, Name  
FROM Professoren  
WHERE Rang = 'C4'
```

- ▶ Anfragen mit Verknüpfungen mehrerer Tabellen

## Beispiel

```
SELECT s.Name, v.Titel  
FROM Studenten s, hoeren h, Vorlesungen v  
WHERE s.MatrNr = h.MatrNr  
AND h.VorlNr = v.VorlNr
```

## Weitere Möglichkeiten in SQL

- ▶ Eliminierung von Duplikaten: **SELECT DISTINCT**
- ▶ Sortierung: **ORDER BY, ASC, DESC**
- ▶ Gruppierung: **GROUP BY, HAVING**
- ▶ Aggregatfunktionen: **COUNT, AVG, SUM, MIN, MAX**
- ▶ Mengenoperationen: **UNION, INTERSECT, EXCEPT (MINUS), IN**
- ▶ Quantoren: = **ALL, ≥ ANY, EXISTS, NOT EXISTS, ...**
- ▶ Behandlung von NULL-Werten: **IS NULL, IS NOT NULL**
- ▶ Geschachtelte Anfragen: **SELECT ... (SELECT ...)**
- ▶ Joins: **NATURAL JOIN, INNER JOIN, OUTER JOIN, etc.**

# Datenintegrität in SQL

- ▶ Kandidatenschlüssel: **UNIQUE**
- ▶ Primärschlüssel: **PRIMARY KEY**
- ▶ Fremdschlüssel (referentielle Integrität): **REFERENCES, FOREIGN KEY**
- ▶ Weitere Bedingungen: **NOT NULL, CHECK**

## Beispiel

```
CREATE TABLE Studenten  
  (MatrNr INTEGER PRIMARY KEY,  
   Name VARCHAR(30) NOT NULL,  
   Semester INTEGER  
   CHECK Semester BETWEEN 1 AND 13)
```



# Beispiel

## Fortsetzung

### Beispiel

```
CREATE TABLE Professoren
```

```
(PersNr INTEGER PRIMARY KEY,  
Name VARCHAR(30) NOT NULL,  
Rang CHARACTER(2),  
CHECK (Rang IN ( 'C2' , 'C3' , 'C4' )),  
Raum INTEGER UNIQUE)
```

```
CREATE TABLE Assistenten
```

```
(PersNr INTEGER PRIMARY KEY,  
Name VARCHAR(30) NOT NULL,  
Fachgebiet VARCHAR(30),  
Boss INTEGER,  
FOREIGN KEY (Boss) REFERENCES Professoren  
ON DELETE SET NULL)
```

# Relationale Entwurfstheorie

- ▶ Funktionale Abhängigkeiten (FDs):
  - ▶ Spezialfall: (Super-)schlüssel
  - ▶ Herleitung weiterer FDs (Armstrong Axiome)
- ▶ Normalformen:
  - ▶ Motivation: Anomalien durch schlechtes Schema
  - ▶ Normalformen: 1NF, 2NF, 3NF, BCNF
- ▶ Normalisierung durch Dekomposition:
  - ▶ Verlustlosigkeit
  - ▶ Abhängigkeitsbewahrung

# Ausblick

## Teil 1

- ▶ Einführung: (relationale) DB Management Systeme in VL Datenbanksysteme
- ▶ Datenbankentwurf, ER Modellierung im Laborübungsteil der VL Datenbanksysteme (Wiederholung)
- ▶ Relationales Datenmodell in VL Datenbanksysteme: Anfrageoptimierung auf der Basis der relationalen Algebra

# Ausblick

## Teil 2

- ▶ Relationale Anfragesprachen, SQL: in VL Datenbanksysteme (Vertiefung), im Laborübungsteil der VL Datenbanksysteme (Wiederholung), in VU Datenbanktheorie (weitere Anfragesprachen)
- ▶ Datenintegrität und relationale Entwurfstheorie: im Laborübungsteil der VL Datenbanksysteme (Tabellen in 3NF), in VU Datenbanktheorie (Vertiefung)
- ▶ Weitere Datenbanken LVAs des Arbeitsbereiches DBAI:
  - ▶ Semistrukturierte Daten (jeweils im SS)
  - ▶ Datenbanktheorie (jeweils im SS)
  - ▶ Advanced Database Systems (jeweils im SS)
  - ▶ Deduktive Datenbanken (jeweils im WS)