

# *Datenbanken:*

## Tutorium 5

Marvin Jahn

20.11.2019

# Das Relationale Schema

- Schema legt die Struktur der gespeicherten Daten fest

# Das Relationale Schema

- Schema legt die Struktur der gespeicherten Daten fest
- **Ausprägung**: der aktuelle Zustand der Datenbasis

# Das Relationale Schema

- Schema legt die Struktur der gespeicherten Daten fest
- **Ausprägung**: der aktuelle Zustand der Datenbasis
- **Schlüssel**: minimale Menge an Attributen, deren Werte ein Tupel (einen Eintrag der Tabelle) *eindeutig* bestimmen

# Das Relationale Schema

- Schema legt die Struktur der gespeicherten Daten fest
- **Ausprägung**: der aktuelle Zustand der Datenbasis
- **Schlüssel**: minimale Menge an Attributen, deren Werte ein Tupel (einen Eintrag der Tabelle) *eindeutig* bestimmen
- es kann mehrere Schlüssel geben, deshalb wählt man einen aus, dieser heißt **Primärschlüssel**

# Das Relationale Schema: Beispiel

Relationales Schema

Studenten		hören		Vorlesungen	
MatrNr	Name	MatrNr	VorlNr	VorlNr	Titel
26120	Fichte	25403	5022	5001	Grundzüge
25403	Jonas	26120	5001	5022	Glaube und Wissen
...	...	...	...	...	...

Notation: Studenten(MatrNr,Name), ...

Bei dieser Notation schreibt man die Datentypen der Attribute nicht dazu, denkt sie sich aber (z.B. *String* für *Name*).

# Vom Entity-Relationship Modell zum relationalen Schema

Vorgehen:

- Übersetzung der Entitäten

# Vom Entity-Relationship Modell zum relationalen Schema

Vorgehen:

- Übersetzung der Entitäten
- Übersetzung der Beziehungen



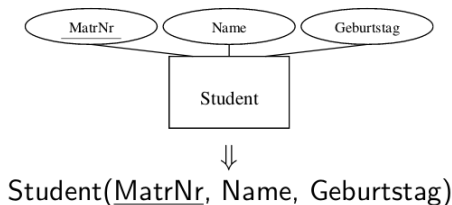
# Vom Entity-Relationship Modell zum relationalen Schema

Vorgehen:

- Übersetzung der Entitäten
- Übersetzung der Beziehungen
- Vereinfachung des Schemas

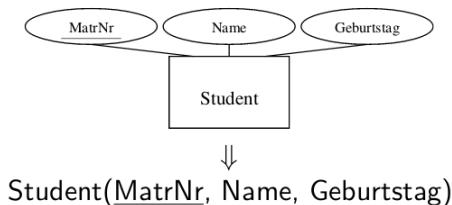
# Übersetzung von Entitäten

- Entität  $\rightarrow$  Relation (Tabelle)



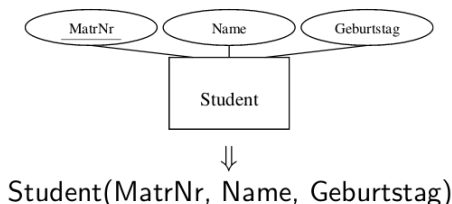
# Übersetzung von Entitäten

- Entität  $\rightarrow$  Relation (Tabelle)
- Attribut einer Entität  $\rightarrow$  Attribut (Spalte) der zugehörigen Relation

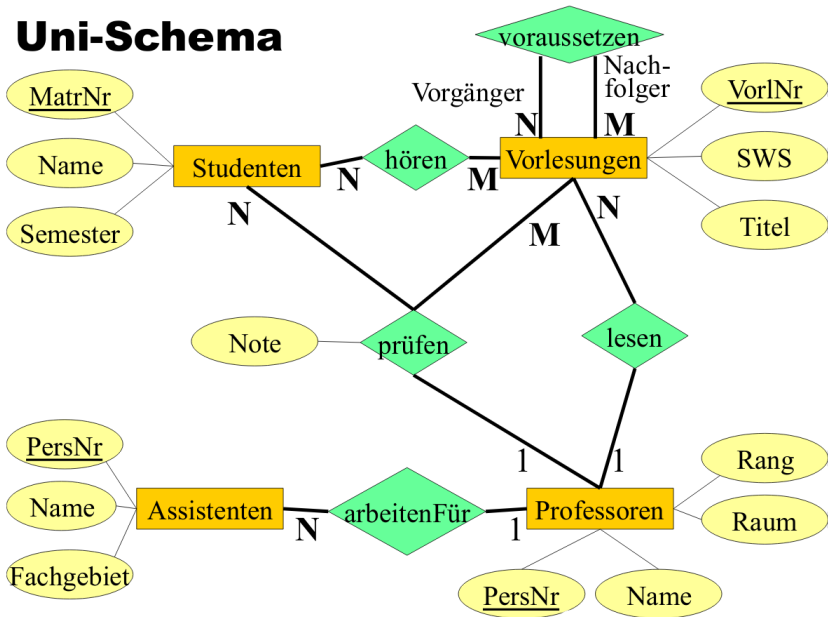


# Übersetzung von Entitäten

- Entität → Relation (Tabelle)
- Attribut einer Entität → Attribut (Spalte) der zugehörigen Relation
- Schlüssel der Entität → Primärschlüssel der Relation



# Uni-Schema



# Übersetzung von Entitäten: Beispiel

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

**Vorlesungen:** {[VorlNr:integer, *Titel: string*, *SWS: integer*]}

**Professoren:** {[PersNr:integer, *Name: string*, *Rang: string*,  
*Raum: integer*]}

**Assistenten:** {[PersNr:integer, *Name: string*, *Fachgebiet: string*]}

Die Datentypbeschreibungen können weggelassen werden.

# Übersetzung der Beziehungen

- Beziehung  $\rightarrow$  Relation

# Übersetzung der Beziehungen

- Beziehung  $\rightarrow$  Relation
- die Attribute der neuen Relation sind die Schlüssel der teilnehmenden Relationen und die Attribute der ursprünglichen Beziehung



# Übersetzung der Beziehungen

- Beziehung  $\rightarrow$  Relation
- die Attribute der neuen Relation sind die Schlüssel der teilnehmenden Relationen und die Attribute der ursprünglichen Beziehung
- die Schlüssel der neuen Relation hängen von der Art der Beziehung ab

# Übersetzung der Beziehungen

- Beziehung  $\rightarrow$  Relation
- die Attribute der neuen Relation sind die Schlüssel der teilnehmenden Relationen und die Attribute der ursprünglichen Beziehung
- die Schlüssel der neuen Relation hängen von der Art der Beziehung ab
- falls es sich um eine binäre Beziehung handelt:

# Übersetzung der Beziehungen

- Beziehung  $\rightarrow$  Relation
- die Attribute der neuen Relation sind die Schlüssel der teilnehmenden Relationen und die Attribute der ursprünglichen Beziehung
- die Schlüssel der neuen Relation hängen von der Art der Beziehung ab
- falls es sich um eine binäre Beziehung handelt:
  - ▶ N:M : Kombination der Entitätsschlüssel

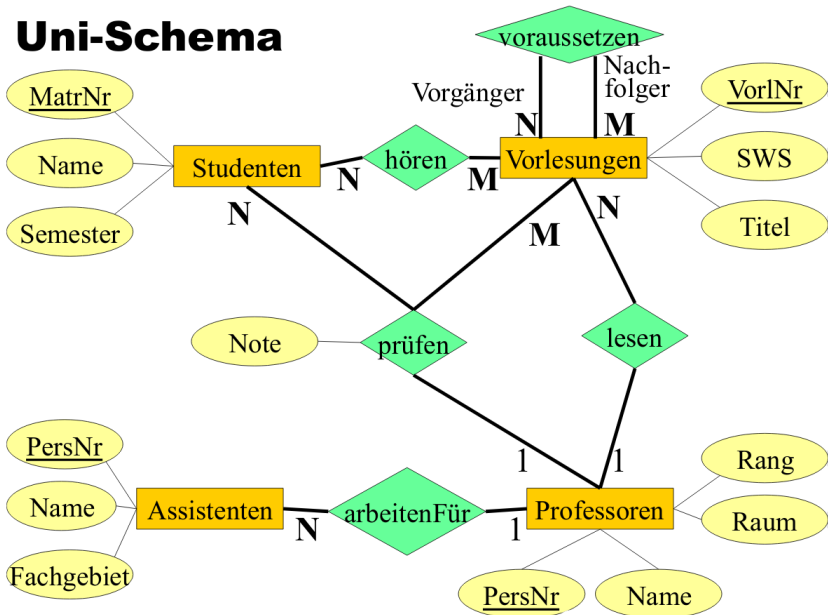
# Übersetzung der Beziehungen

- Beziehung  $\rightarrow$  Relation
- die Attribute der neuen Relation sind die Schlüssel der teilnehmenden Relationen und die Attribute der ursprünglichen Beziehung
- die Schlüssel der neuen Relation hängen von der Art der Beziehung ab
- falls es sich um eine binäre Beziehung handelt:
  - ▶ N:M : Kombination der Entitätsschlüssel
  - ▶ 1:N,N:1 : Schlüssel der N-Seite wird Schlüssel

# Übersetzung der Beziehungen

- Beziehung  $\rightarrow$  Relation
- die Attribute der neuen Relation sind die Schlüssel der teilnehmenden Relationen und die Attribute der ursprünglichen Beziehung
- die Schlüssel der neuen Relation hängen von der Art der Beziehung ab
- falls es sich um eine binäre Beziehung handelt:
  - ▶ N:M : Kombination der Entitätsschlüssel
  - ▶ 1:N,N:1 : Schlüssel der N-Seite wird Schlüssel
  - ▶ 1:1 : einer der beiden Schlüssel kann gewählt werden

# Uni-Schema



# Übersetzung der Beziehungen: Beispiel

**hören** : {[MatrNr: integer, VorlNr: integer]}

**lesen** : {[PersNr: integer, VorlNr: integer]}

**arbeitenFür** : {[AssistentenPersNr: integer, *ProfPersNr*: integer]}

**voraussetzen** : {[Vorgänger: integer, Nachfolger: integer]}

**prüfen** : {[MatrNr: integer, VorlNr: integer, PersNr: integer,  
Note: decimal]}

# Vereinfachung des Schemas

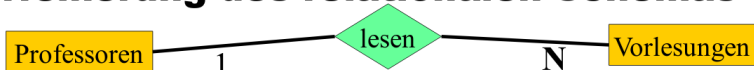
- falls Relationen mit dem gleichen Schlüssel existieren (dies passiert z.B. bei 1:N, N:1 oder 1:1 Beziehungen), werden diese zu einer einzigen Relation zusammengefasst



# Vereinfachung des Schemas

- falls Relationen mit dem gleichen Schlüssel existieren (dies passiert z.B. bei 1:N, N:1 oder 1:1 Beziehungen), werden diese zu einer einzigen Relation zusammengefasst
- zur besseren Lesbarkeit benennt man häufig Attribute bei dieser Zusammenfassung um

# Verfeinerung des relationalen Schemas



## 1:N-Beziehung

- Initial-Entwurf

**Vorlesungen** : {[VorlNr, Titel, SWS]}

**Professoren** : {[PersNr, Name, Rang, Raum]}

**lesen**: {[VorlNr, PersNr]}

- Verfeinerung durch Zusammenfassung

**Vorlesungen** : {[VorlNr, Titel, SWS, **gelesenVon**]}

**Professoren** : {[PersNr, Name, Rang, Raum]}

## Regel

Relationen mit gleichem Schlüssel kann man zusammenfassen  
**aber nur diese und keine anderen!**

# SQL: Beispiel 1

*“Gebe alle Informationen über alle Vorlesungen aus, die von Sokrates gelesen werden.”*

# SQL: Beispiel 1

*“Gebe alle Informationen über alle Vorlesungen aus, die von Sokrates gelesen werden.”*

```
select v.*  
from Vorlesungen v, Professoren p  
where v.gelesenVon = p.PersNr  
      and p.Name = 'Sokrates'
```

## SQL: Beispiel 2

*“Bestimme den Namen aller Studenten, die mind. im fünften Semester sind und mind. eine Vorlesung hören.”*

## SQL: Beispiel 2

*“Bestimme den Namen aller Studenten, die mind. im fünften Semester sind und mind. eine Vorlesung hören.”*

```
select distinct s.Name  
from Studenten s, hoeren h  
where s.Semester > 5 and s.MatrNr = h.MatrNr
```

# SQL: Beispiel 3

*“Bestimme die Anzahl der Studenten.”*

## SQL: Beispiel 3

*“Bestimme die Anzahl der Studenten.”*

```
select count(*)  
from Studenten
```

Statt *\** in *count(\*)* funktioniert auch ein beliebiges Attribut von Studenten.



## SQL: Beispiel 4

*“Generiere eine Liste, die alle Professoren und die Anzahl ihrer gelesenen Vorlesungen angibt. Ordne die Liste nach der Anzahl.”*

## SQL: Beispiel 4

*“Generiere eine Liste, die alle Professoren und die Anzahl ihrer gelesenen Vorlesungen angibt. Ordne die Liste nach der Anzahl.”*

### Falsche Lösung:

```
select p.PersNr, p.Name, count(VorINr) as anzahl
from Vorlesungen v, Professoren p
where v.gelesenVon = p.PersNr
group by p.PersNr, p.Name
order by anzahl desc
```

## SQL: Beispiel 4

*“Generiere eine Liste, die alle Professoren und die Anzahl ihrer gelesenen Vorlesungen angibt. Ordne die Liste nach der Anzahl.”*

### Falsche Lösung:

```
select p.PersNr, p.Name, count(VorINr) as anzahl
from Vorlesungen v, Professoren p
where v.gelesenVon = p.PersNr
group by p.PersNr, p.Name
order by anzahl desc
```

Was ist mit Professoren, die keine Vorlesung halten?

## SQL: Beispiel 4

*“Generiere eine Liste, die alle Professoren und die Anzahl ihrer gelesenen Vorlesungen angibt. Ordne die Liste nach der Anzahl.”*

### Richtige Lösung:

```
select p.PersNr, p.Name, count(VorINr) as anzahl
from professoren p left outer join vorlesungen v
on v.gelesenVon = p.PersNr
group by p.PersNr, p.Name
order by anzahl desc
```