



# Spaß mit Datenbanken



# Servus!

Ich bin Matthias.

CEO Frischluft Medien OG

Lehrender FH Hagenberg



# Fahrplan durch den Kurs

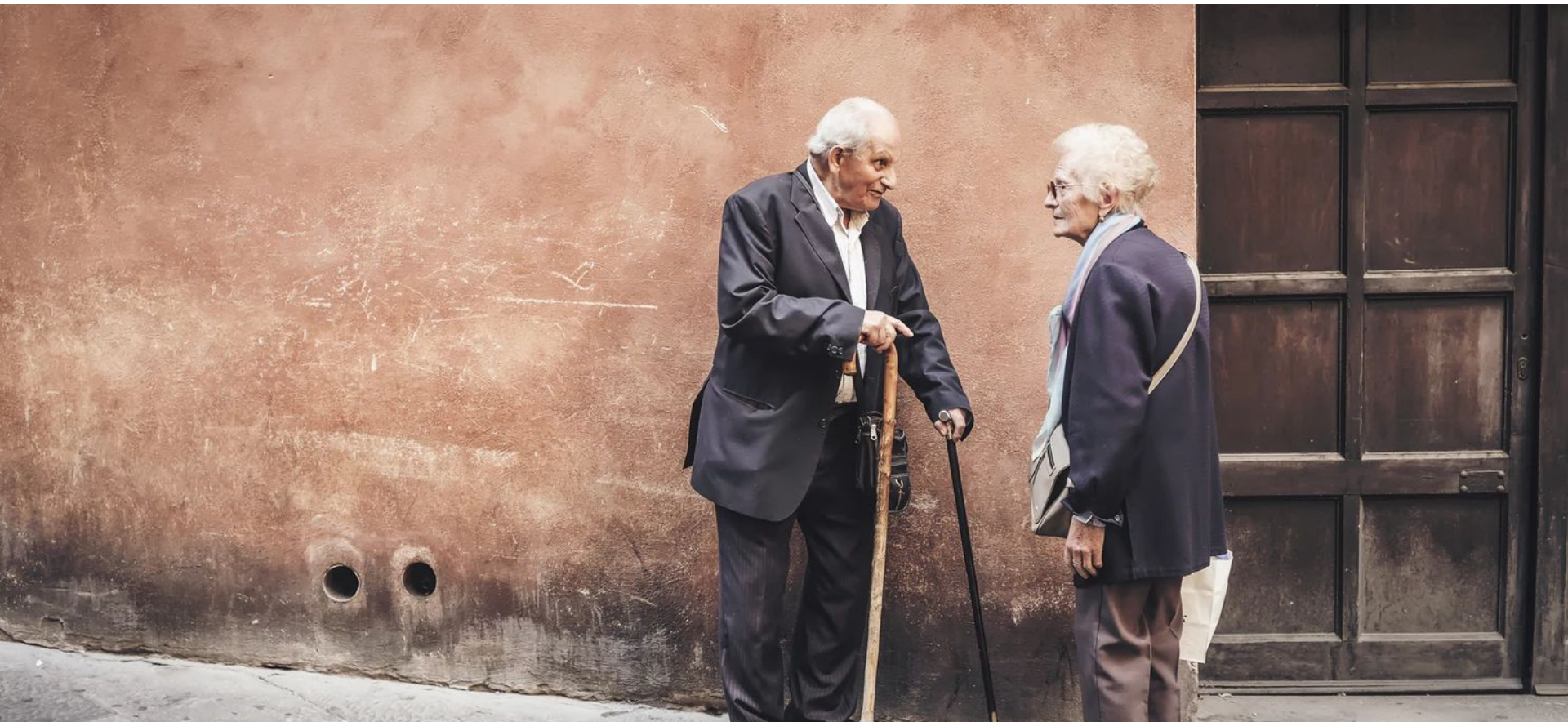


Unit 12:  
Datenbank-  
entwurf



Unit 13:  
Datenbanken  
und MySQL

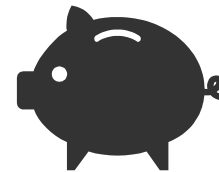
# Hallo sagen.



# Datenbankentwurf

Einführung

# Datenbank?



**Die Daten “auf die  
Bank bringen”**

Sicherheit vor Verlusten



Bank bedient **mehrere  
Kunden gleichzeitig**



Strukturierte  
Hinterlegung

# Die 9 Codd'schen Anforderungen

## Integration

einheitliche, nicht redundante Datenverwaltung

## Benutzersichten

Unterschiedliche Sichten für Benutzer

## Transaktionen

Mehrere DB-Operationen als Funktionseinheit

## Operationen

Speichern, Suchen, Ändern, Einfügen

## Integritätssicherung

Korrektheit des Datenbestandes

## Synchronisation

parallele Transaktionen mehrerer Benutzer koordinieren

## Katalog

Zugriffe auf Datenbeschreibungen  
(Data Dictionary)

## Zugriffskontrolle

Ausschluss von unberechtigtem Zugriff

## Datensicherung

Wiederherstellung von Daten nach Systemfehlern



I must to the **Datenbank.**





# Datenmodellierung



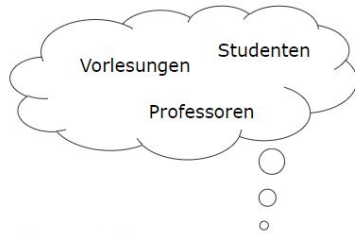
Ausschnitt



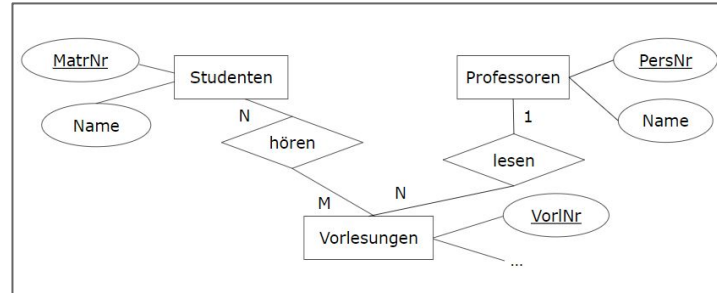
Konzeptuelles Schema



Logisches Schema



Reale Welt: Universität



Studenten	
MatrNr	Name
26120	Fichte
25403	Jonas
...	...

hören	
MatrNr	VorlNr
25403	5022
26120	5001
...	...

Vorlesungen	
VorlNr	Titel
5001	Grundzüge
5022	Glaube und Wissen
...	...

# Datenmodellierung



Logisches Schema

Abfrage & Manipulation

Studenten	
MatrNr	Name
26120	Fichte
25403	Jonas
...	...

hören	
MatrNr	VorlNr
25403	5022
26120	5001
...	...

Vorlesungen	
VorlNr	Titel
5001	Grundzüge
5022	Glaube und Wissen
...	...

```
SELECT Name
FROM Studenten, hören, Vorlesungen
WHERE Studenten.MatrNr = hören.MatrNr AND
       hören.VorlNr = Vorlesungen.VorlNr AND
       Vorlesungen.Titel = 'Grundzüge';
```

```
UPDATE Vorlesungen
SET Titel = 'Grundzüge der Logik'
WHERE VorlNr = 5001;
```

2,3,4

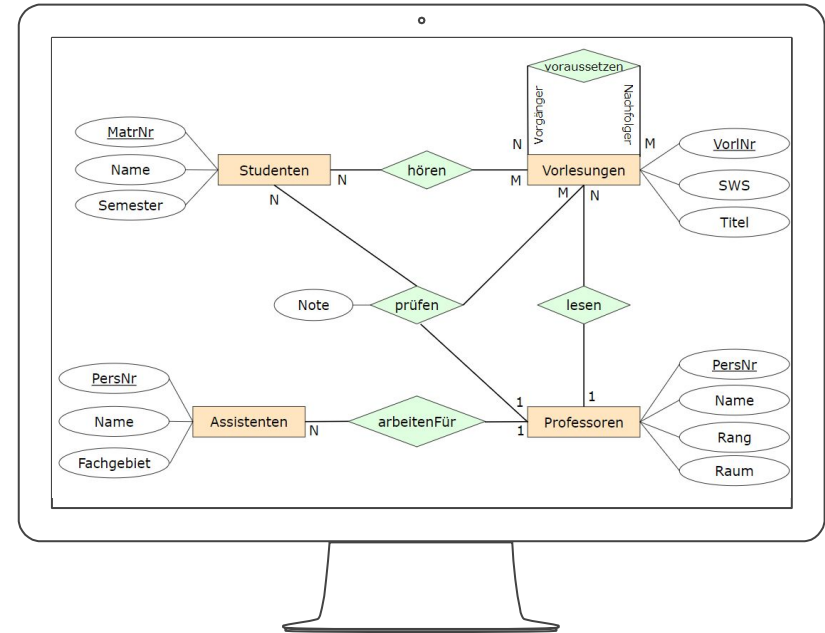
# Datenbankentwurf

ER Modell, Datentypen, Schlüsselfelder

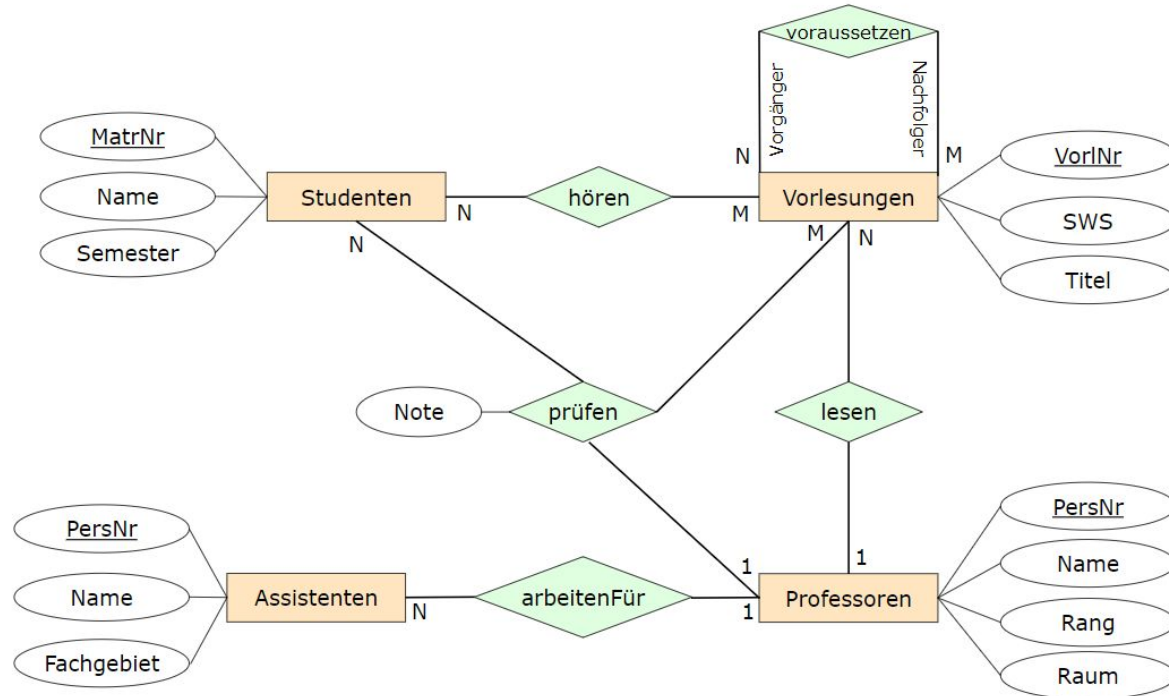
# Entity Relationship Modell

Modelliert **Gegenstände (Entities)** und die **Beziehungen (Relationships)** zwischen diesen.

- **Entity**
  - **Objekt**, über welches Informationen zu speichern sind (bspw. Vorlesung, Professor, Prüfung).
- **Relationship**
  - **Beziehung** zwischen Entities. z.B. Professor liest Vorlesung, Professor prüft Student.
- **Attribut**
  - **Eigenschaft** von Entities oder Beziehungen. z.B. Name von Professor, Titel einer Vorlesung.



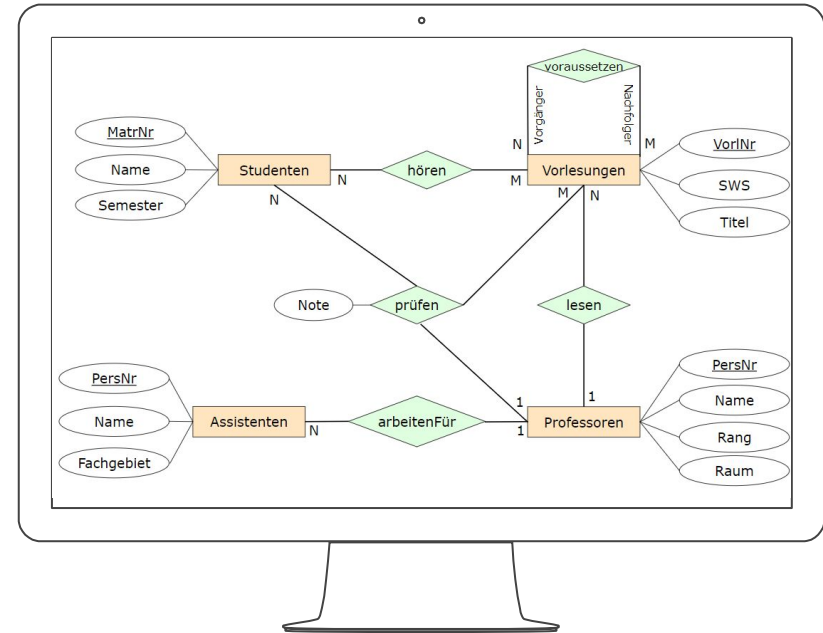
# ER-Modell - Chen Notation





# Entity Relationship Modell

- **Werte**
  - Primitive Datenelemente, die direkt darstellbar sind
  - Werden durch **Datentypen** beschrieben
- **Datentypen**
  - Vorgegebene Standard-Datentypen, wie etwa ganze Zahlen (**int**), Zeichenketten (**string**), Datumswerte (**date**) etc.



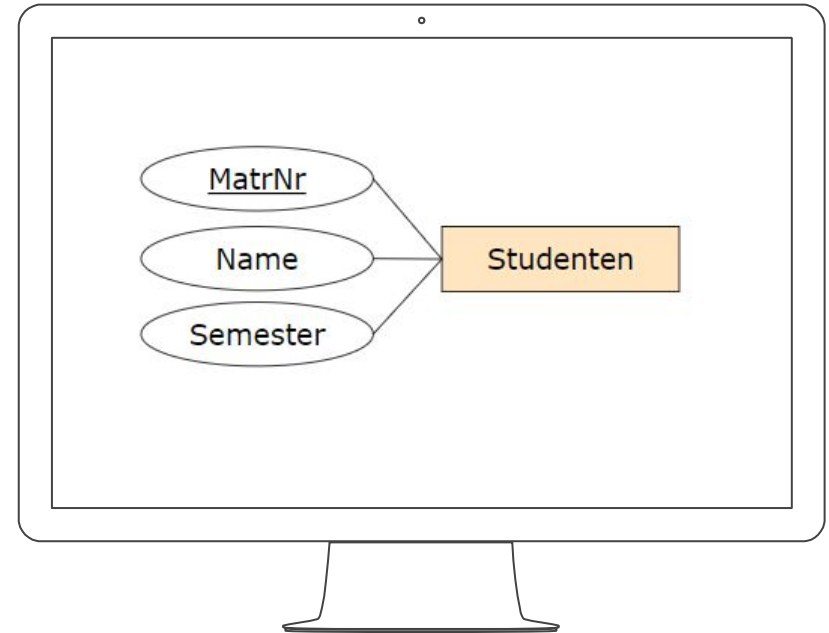






# Attribute

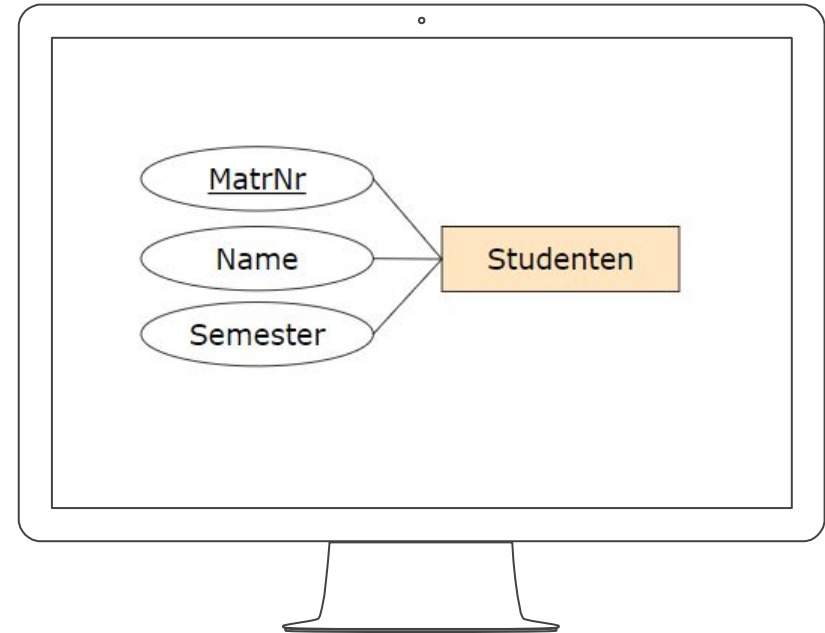
- Modellieren Eigenschaften von **Entities oder Beziehungen**
- **Alle Entities** eines Entity-Typen (hier bspw. "Studenten") **haben dieselben** Arten von **Eigenschaften**.
- Attribute werden somit für **Entity-Typen** deklariert
- textuelle Notation:  
E (A1 : D1 , ..., Ax : Dx )
- Attributen ist ein **Datentyp** zuzuweisen





# Schlüssel

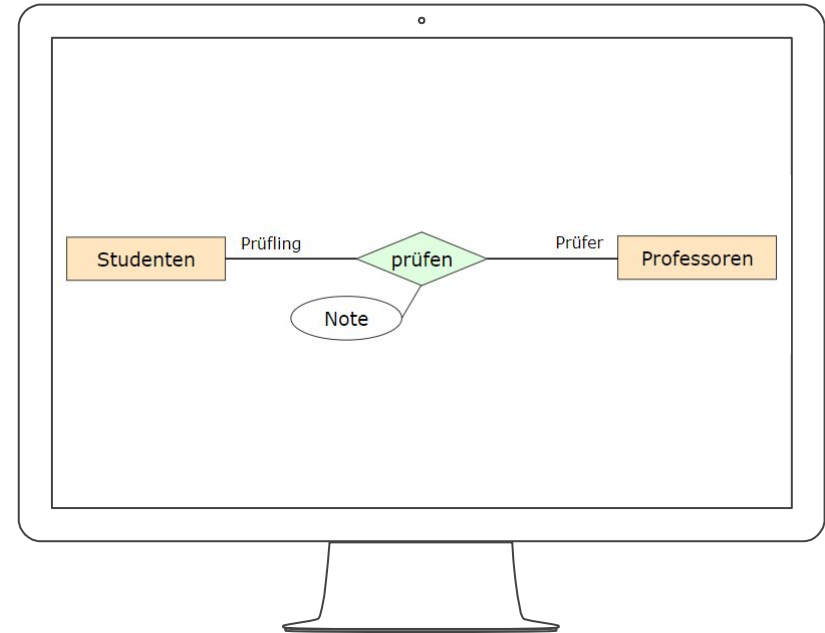
- Die Werte der Schlüsselattribute identifizieren Entities eindeutig
- Bei mehreren Schlüsselkandidaten ist ein **Primärschlüssel** zu wählen (im Modell unterstreichen)
- Beispiel rechts: Die Matrikelnummer kennzeichnet einen Studenten eindeutig. Zu jeder Matrikelnummer gibt es genau einen Studenten.





# Beziehungen

- Beziehungen zwischen Entities werden zu **Beziehungstypen** zusammengefasst.
- Beziehungen können ebenfalls **Attribute** besitzen (Note im Beispiel)



# Beziehungen

## Unterscheidung in Grade

Am häufigsten: **binär**

Zwei beteiligte Entitäten

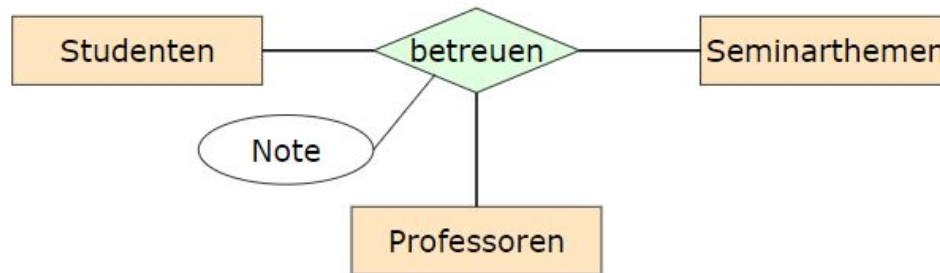


# Beziehungen

## Unterscheidung in Grade

Weiters: **ternär**

Drei beteiligte Entitäten

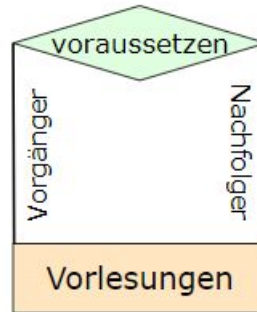


# Beziehungen

## Unterscheidung in Grade

Und dann noch: **unär**

Eine beteiligte Entität



# Beziehungen

## Unterscheidung Kardinalitäten

Beschreiben die Beziehung genauer.

Formen: 1:1, 1:N, N:1, M:N

Müssen immer eingehalten werden





# Zwischenübung

## Kardinalitäten

Definiere die Kardinalitäten für folgende Beziehungen:

- Mann ist verheiratet mit Frau
- Prospekt beschreibt Produkt
- Lehrer unterrichtet Fach
- Lieferant liefert Produkt
- Mitarbeiter arbeitet für Firma
- Bestellung umfasst Produkt
- Kino hat Kinosaal

# Zwischenübung

## Kardinalitäten

Überlege dir nun je ein eigenes Beispiel:

- 1:1
- 1:N
- N:1
- M:N

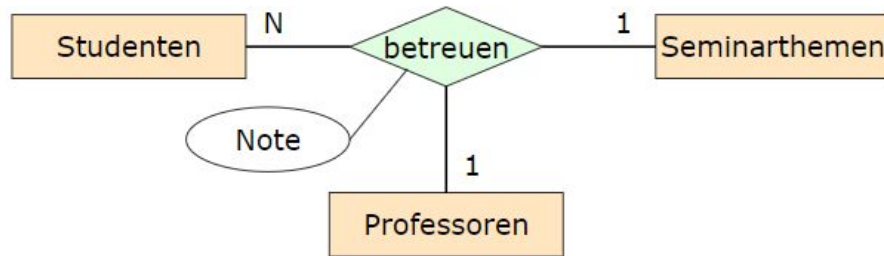
**Mal Pause.**



# Beziehungen

## Kardinalitäten bei ternären Beziehungen

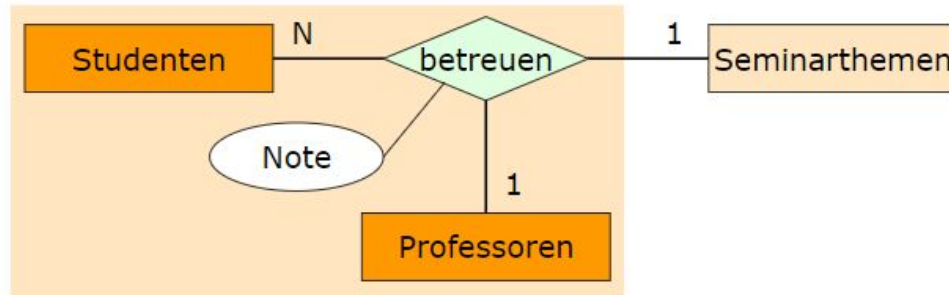
Welche Integritätsbedingungen werden hier festgelegt?



# Beziehungen

## Kardinalitäten bei ternären Beziehungen

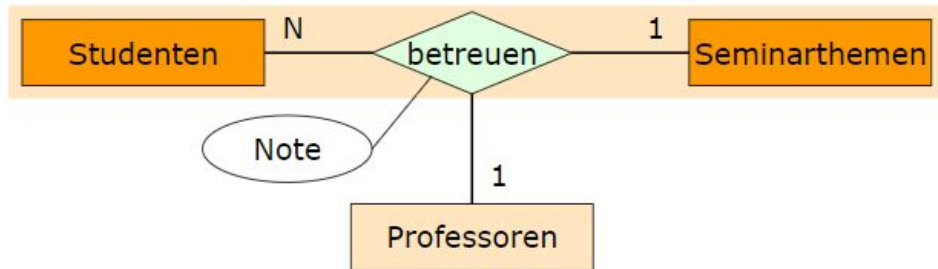
Studenten dürfen bei demselben Professor nur ein Seminarthema bearbeiten



# Beziehungen

## Kardinalitäten bei ternären Beziehungen

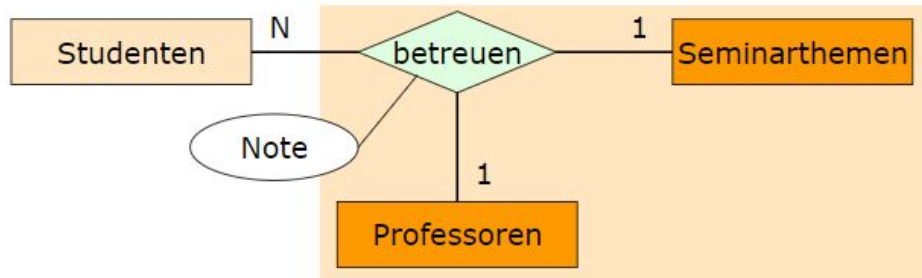
Studenten dürfen dasselbe Seminarthema nur bei einem Professor bearbeiten (also nicht wiederverwenden).



# Beziehungen

## Kardinalitäten bei ternären Beziehungen

Professoren können dasselbe Seminarthema an mehrere Studenten vergeben (= "wiederverwenden")





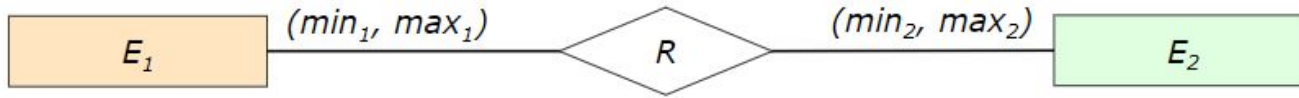


# Datenbankentwurf

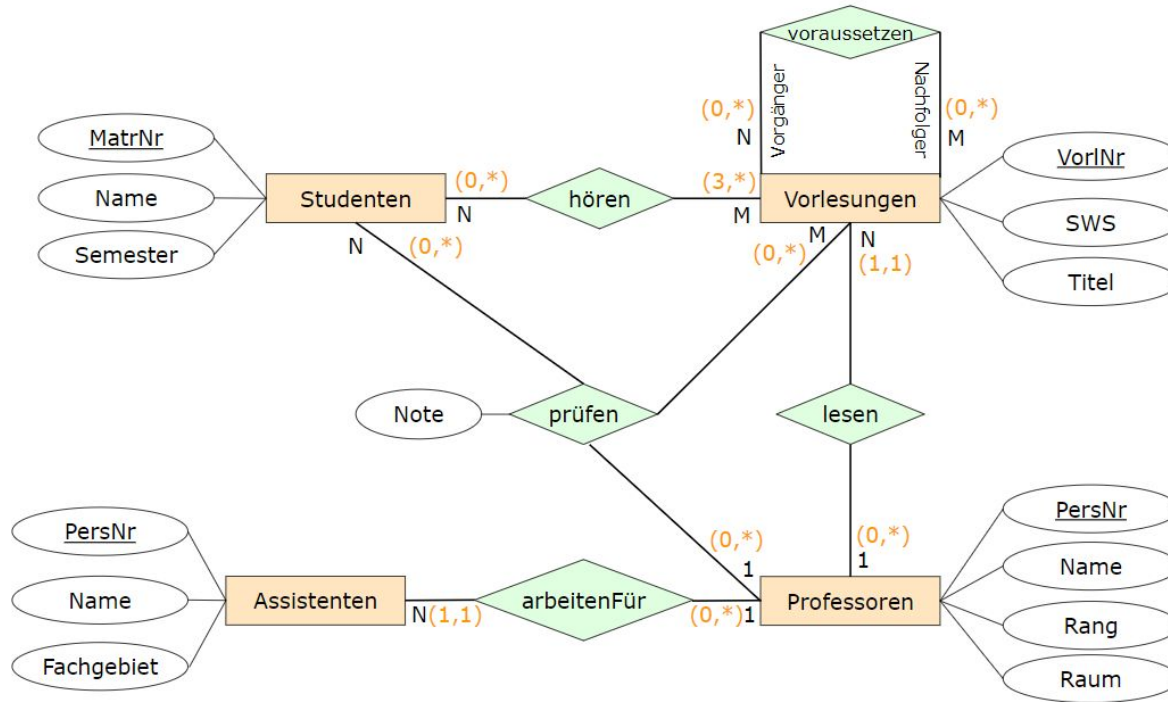
Weitere Konzepte

# (min, max)-Notation

- Verleiht einer Beziehung noch mehr Information
- Schränkt Teilnahme von Instanzen durch Vorgabe von minimal- und maximalwert.



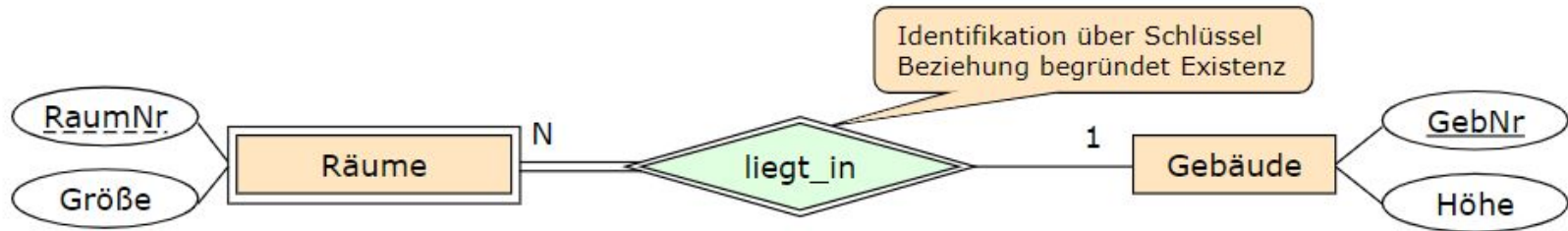
# Beispiel (min, max)-Notation



# Existenzabhängigkeit

## Starke und Schwache Entitäten

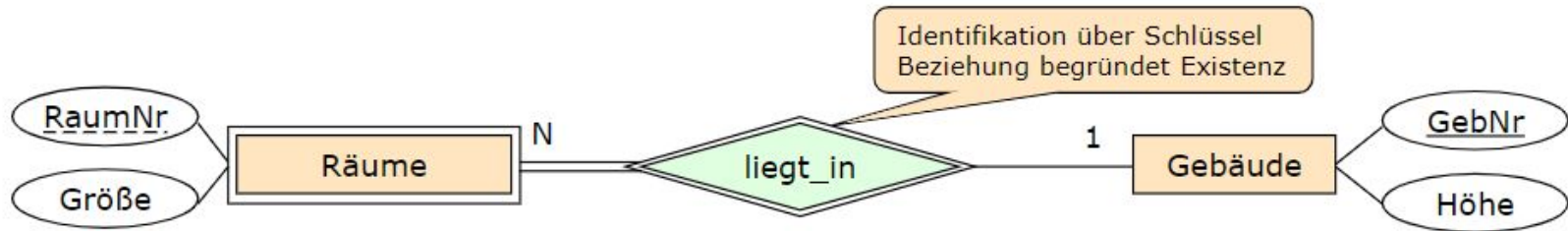
- Schwache Entities sind in ihrer Existenz von einer anderen (starken) abhängig
- Sind nur in Kombination mit dem Schlüssel der starken Entity eindeutig identifizierbar.



# Existenzabhängigkeit

## Starke und Schwache Entitäten

**Beispiel:** Ist die Nummer eines Raumes nur innerhalb eines Gebäudes eindeutig, so ist der Schlüssel von **Räume** eine Kombination aus RaumNr. und GebNr.



# Datenbankentwurf

Relationales Schema



# Relationenmodell

- Konzeptuell ist die Datenbank eine **Menge von Tabellen**
- Eine Tabelle = "**Relation**"
- Eine Zeile = "**Tupel**"

**Professoren**

<u>PersNr</u>	<u>Name</u>	<u>Rang</u>	<u>Raum</u>
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

**Vorlesungen**

<u>VorlNr</u>	<u>Titel</u>	<u>SWS</u>	<u>GelesenVon</u>
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

# Relationenmodell

## Grundregeln

1. Jede Zeile (Tupel) ist **eindeutig** und beschreibt ein Objekt bzw. Entität der Realität
2. Die **Ordnung der Zeilen** ist ohne Bedeutung. Durch die Reihenfolge wird keine relevante Information ausgedrückt.
3. Die **Ordnung der Spalten** ist ohne Bedeutung, da sie einen eindeutigen Namen (Attributnamen) tragen.
4. Jeder Datenwert innerhalb einer Relation ist ein **atomares Datenelement**.
5. Für die Attribute sind **NULL-Werte** erlaubt.
6. Es existieren **Primärschlüssel** und ggf. weitere **Schlüsselkandidaten**.

# Relationenmodell

## Schlüssel

- Attribute, deren Werte ein Tupel **eindeutig** identifizieren sind **Schlüssel**.
- Schlüssel soll **minimal** sein, also aus so wenig Attributen wie möglich bestehen.
- Schlüsselwert darf **niemals** NULL sein!

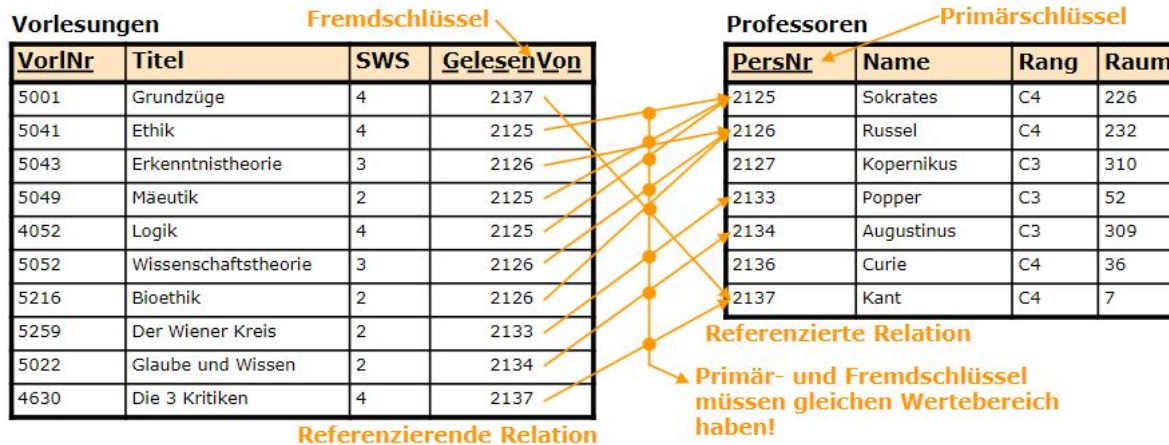
Professoren

<u>PersNr</u>	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

# Relationenmodell

## Fremdschlüssel

- Schlüssel einer Tabelle können in einer anderen (oder derselben) Tabelle als **eindeutige Verweise** genutzt werden.

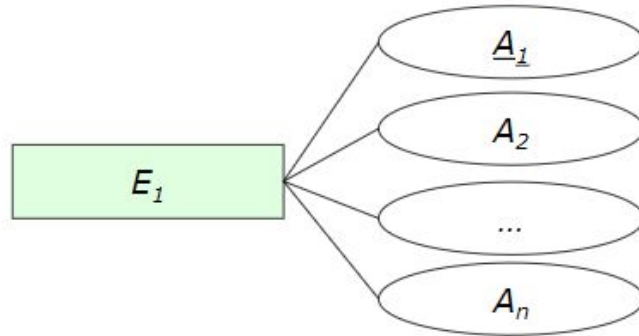


# Relationenmodell

## Fremdschlüssel

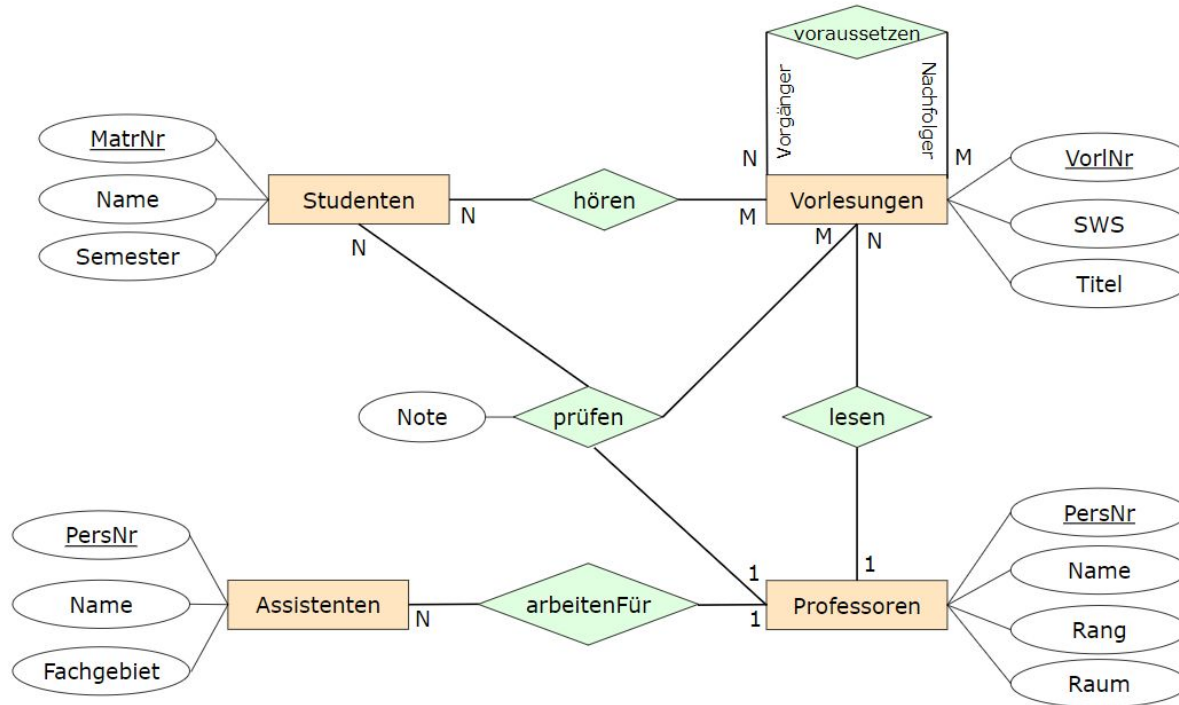
- Fremdschlüssel können **NULL-Werte** aufweisen (wenn sie nicht Teil eines Primärschlüssels sind und nicht explizit "NOT NULL" spezifiziert ist)
- Eine Relation kann **mehrere Fremdschlüssel** besitzen.
- Fremdschlüssel kann auch die **eigene Relation** referenzieren

# Relationale Darstellung von Entity-Typen



$E_1: \{[\underline{A_1:typ_1}, A_2:typ_2, \dots, A_n:typ_n] \}$

# Relationale Darstellung (Uni)



# Relationale Darstellung (Uni)

Logischer Entwurf:

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

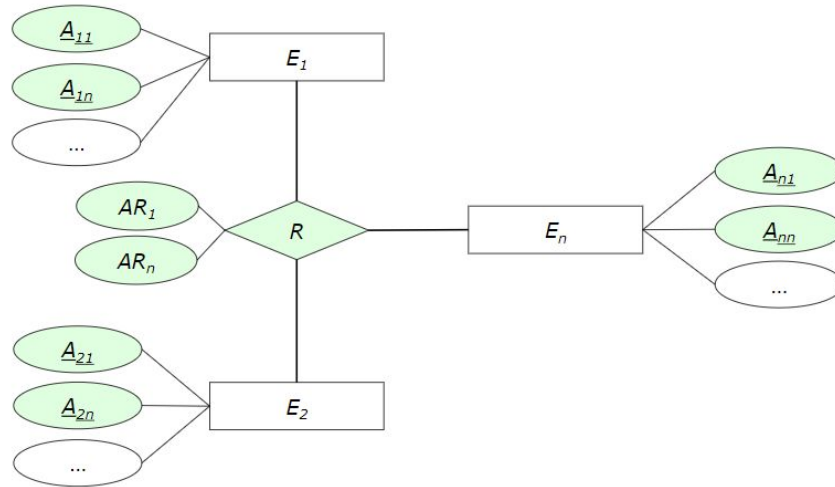
Vorlesungen: {[VorlNr:integer, Titel:string, SWS:integer]}

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

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



# Relationale Darstellung



$\mathcal{R}: \{[A_{11}:\text{typ}_{11}, A_{1n}:\text{typ}_{1n}, A_{21}:\text{typ}_{21}, A_{2n}:\text{typ}_{2n}, A_{n1}:\text{typ}_{n1}, A_{nn}:\text{typ}_{nn}, AR_1:\text{typ}_{R1}, AR_n:\text{typ}_{Rn}]\}$

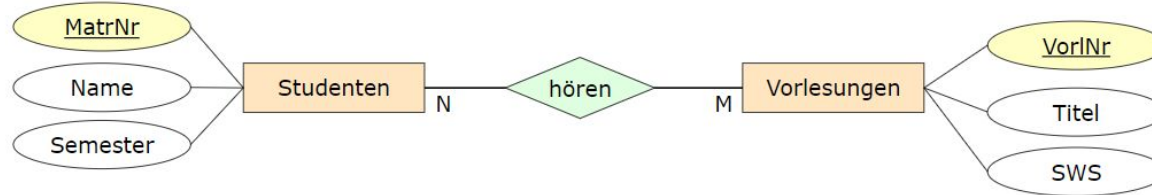
Schlüssel von  $E_1$

Schlüssel von  $E_2$

Schlüssel von  $E_n$

Attribute von  $R$

# Kreuztabellen für N:M Beziehung



Studenten

<u>MatrNr</u>	Name
24002	Xenokrates
25403	Jonas
26120	Fichte
26830	Aristoxenos
28106	Carnap
29555	Feuerbach
...	...

hören

<u>MatrNr</u>	<u>VorlNr</u>
26120	5001
24002	5001
24002	4052
...	...

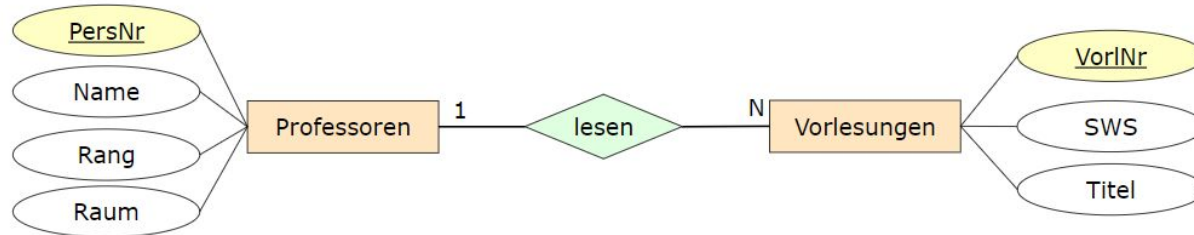
Vorlesungen

<u>VorlNr</u>	Titel	SWS
5001	Grundzüge	4
5041	Ethik	4
5043	Erkenntnistheorie	3
5049	Mäeutik	2
4052	Logik	4
5052	Wissenschaftstheorie	3
5216	Bioethik	2
5259	Der Wiener Kreis	2
5022	Glaube und Wissen	2
4630	Die 3 Kritiken	4

Logischer Entwurf:

hören: {[MatrNr:integer, VorlNr:integer]} (**N:M**)

# Kreuztabellen für 1:N Beziehung



**Professoren**

<u>PersNr</u>	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

**lesen**

<u>PersNr</u>	<u>VorlNr</u>
2137	5001
2125	5041
2125	4049
...	...

**Vorlesungen**

<u>VorlNr</u>	Titel	SWS
5001	Grundzüge	4
5041	Ethik	4
5043	Erkenntnistheorie	3
5049	Mäeutik	2
4052	Logik	4
5052	Wissenschaftstheorie	3
5216	Bioethik	2
5259	Der Wiener Kreis	2
5022	Glaube und Wissen	2
4630	Die 3 Kritiken	4