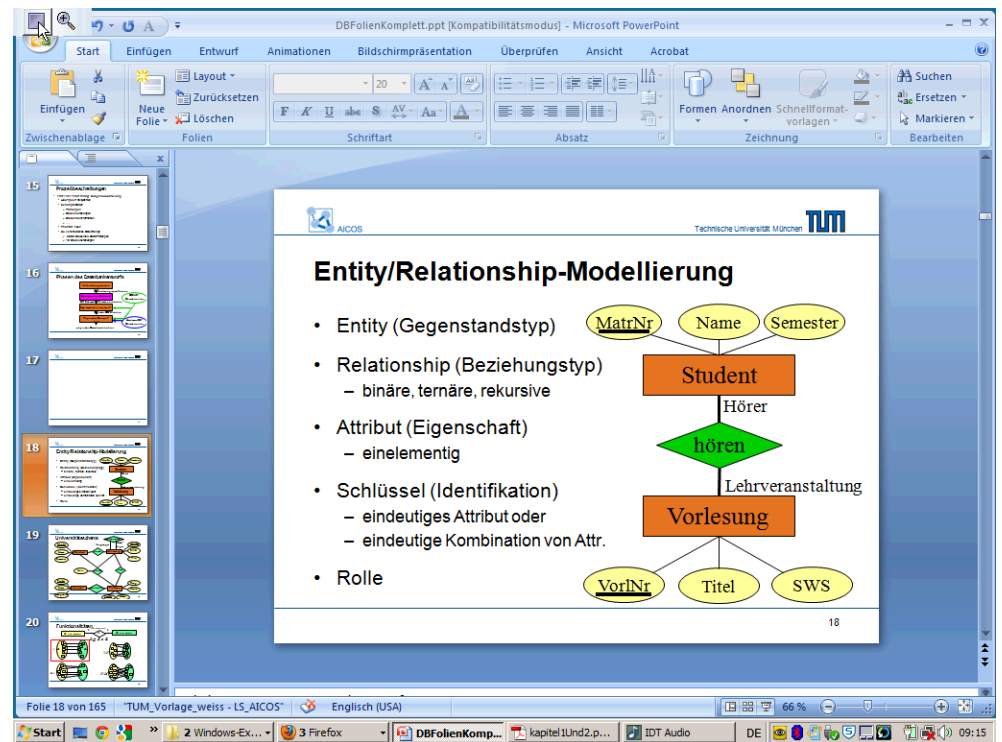


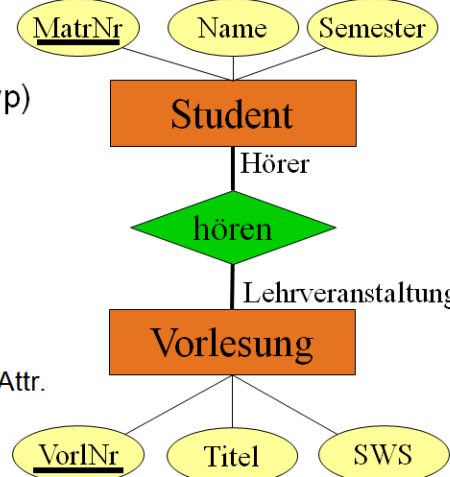
**Script** generated by TTT

Title: Groh: wzw\_2012 (27.04.2012)  
 Date: Fri Apr 27 09:15:49 CEST 2012  
 Duration: 92:10 min  
 Pages: 56

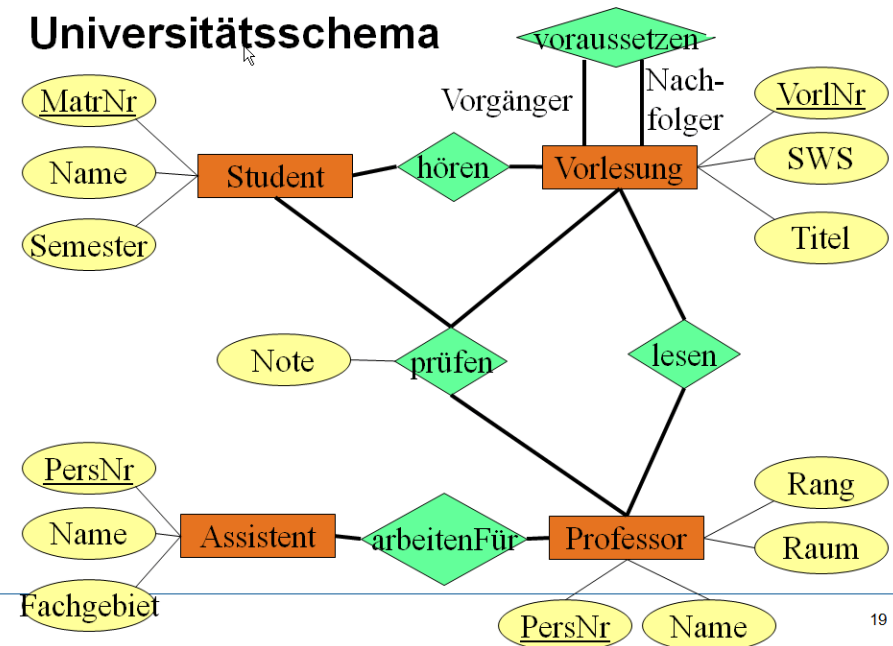


**Entity/Relationship-Modellierung**

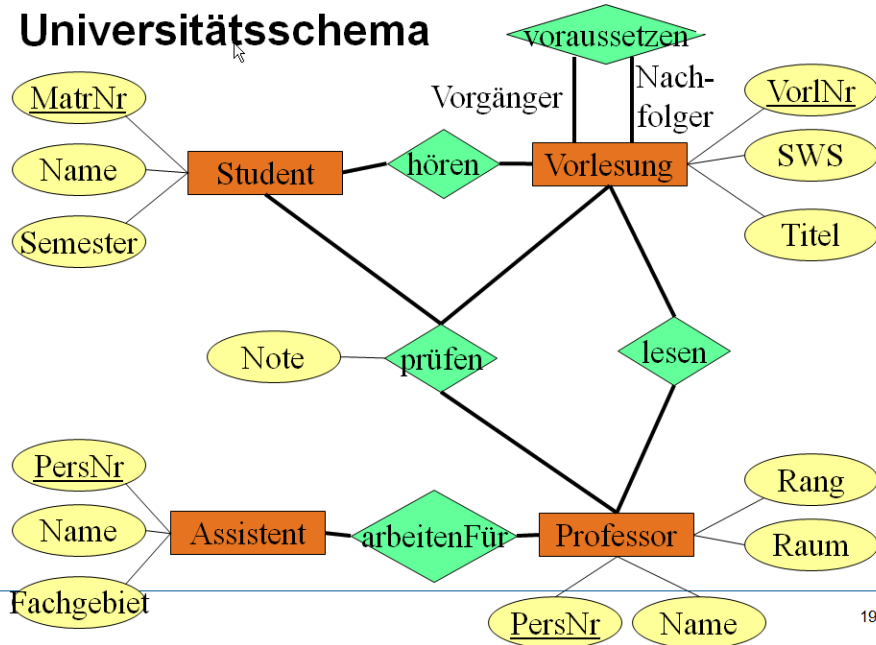
- Entity (Gegenstandstyp)
- Relationship (Beziehungstyp)
  - binäre, ternäre, rekursive
- Attribut (Eigenschaft)
  - einelementig
- Schlüssel (Identifikation)
  - eindeutiges Attribut oder
  - eindeutige Kombination von Attr.
- Rolle



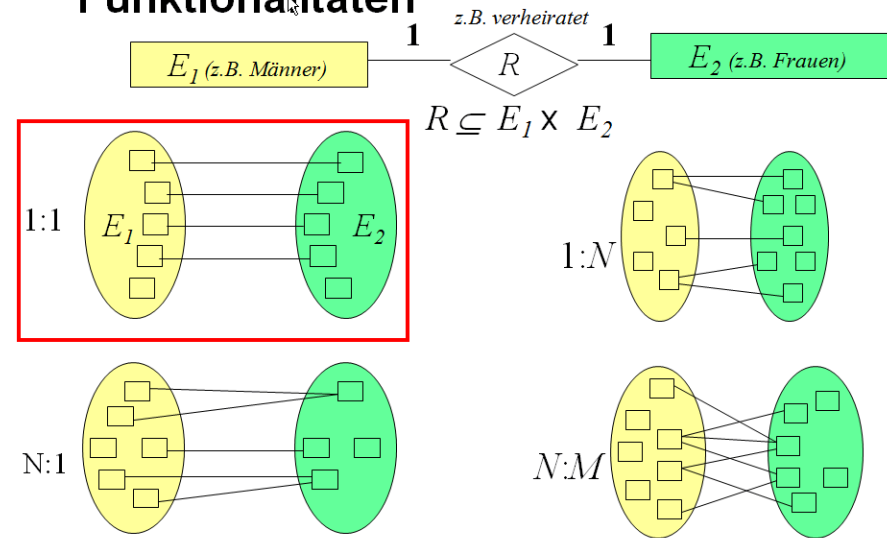
**Universitätsschema**



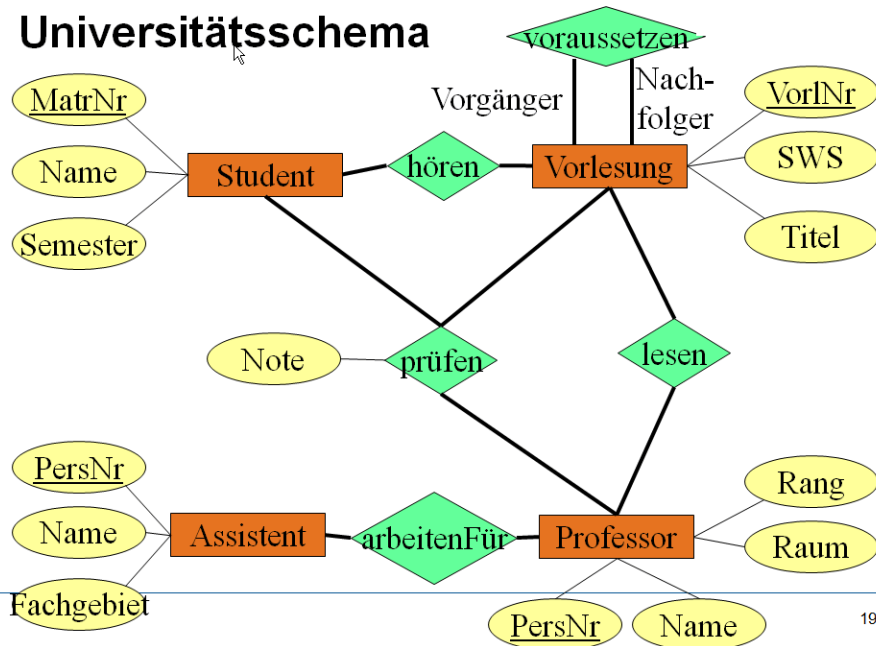
# Universitätsschema



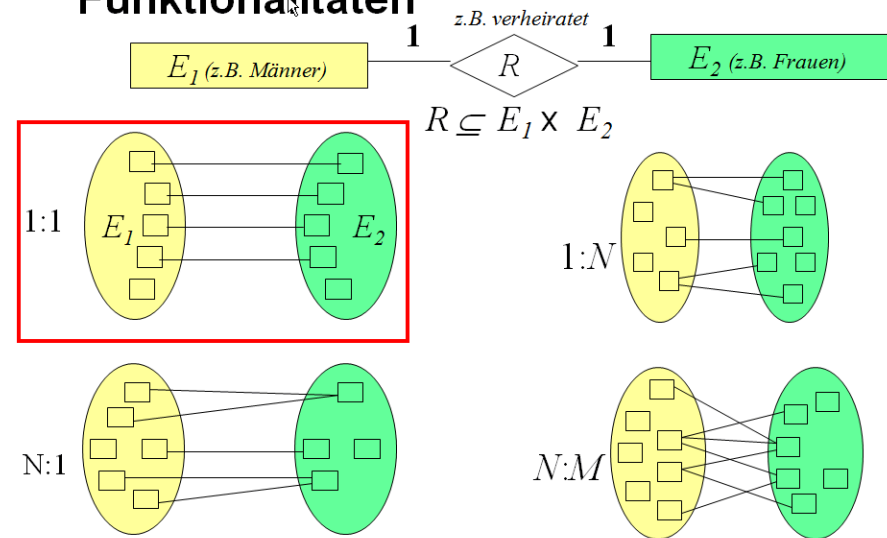
# Funktionalitäten



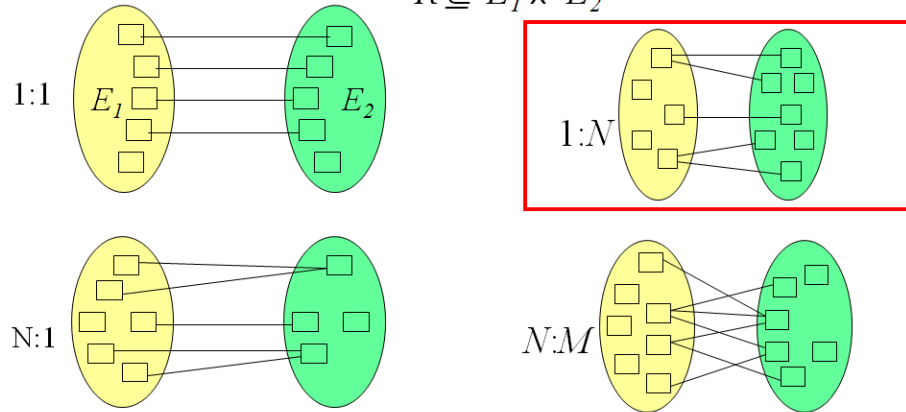
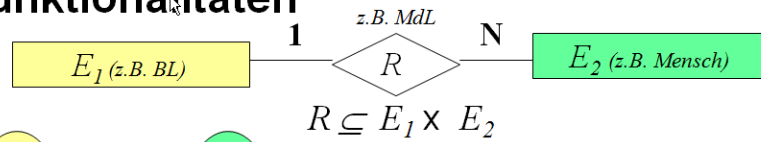
# Universitätsschema



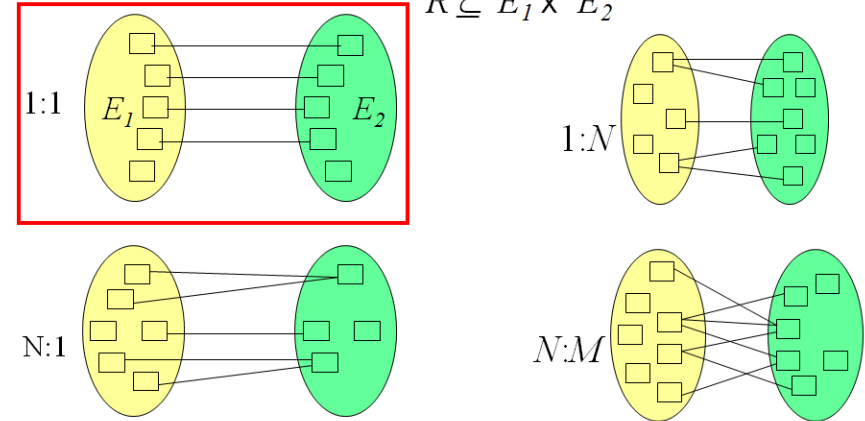
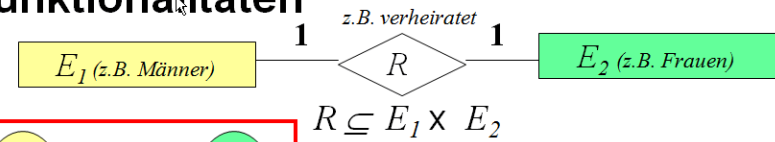
# Funktionalitäten



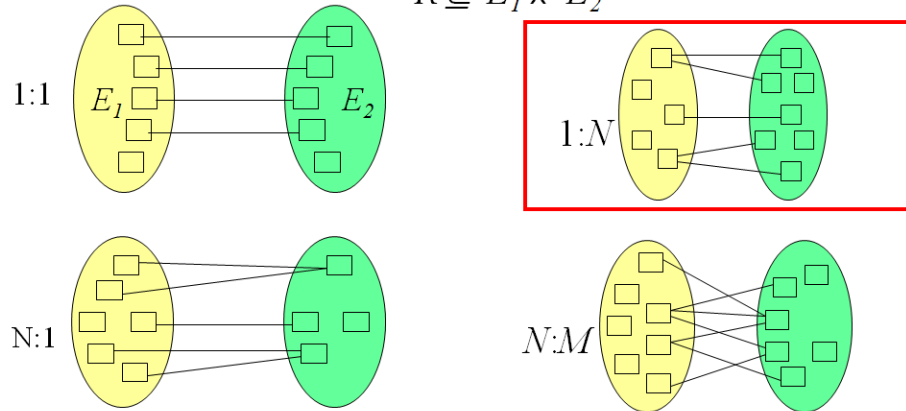
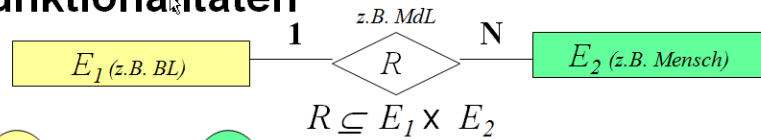
### Funktionalitäten



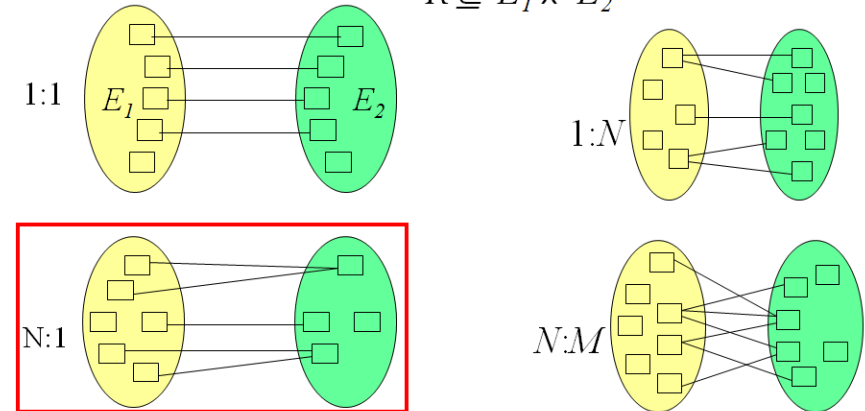
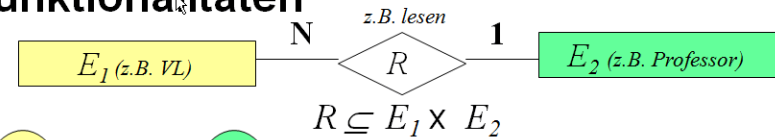
### Funktionalitäten



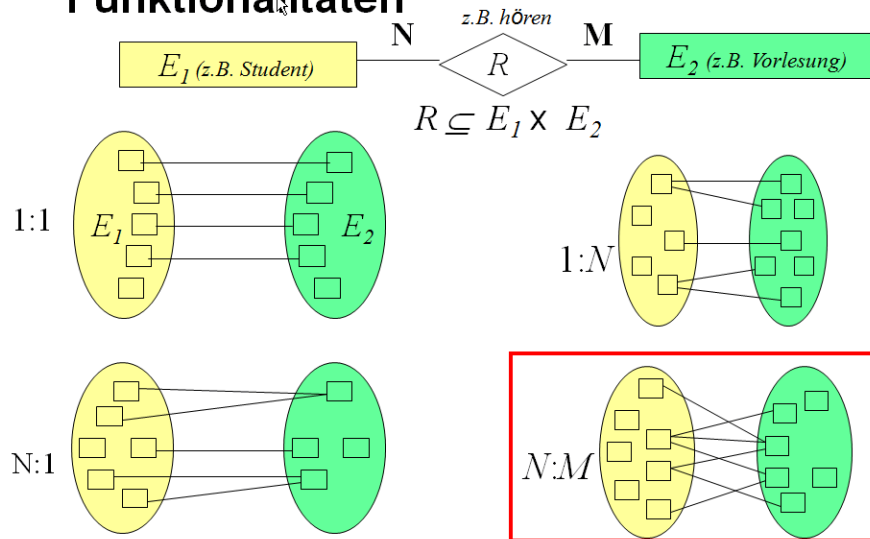
### Funktionalitäten



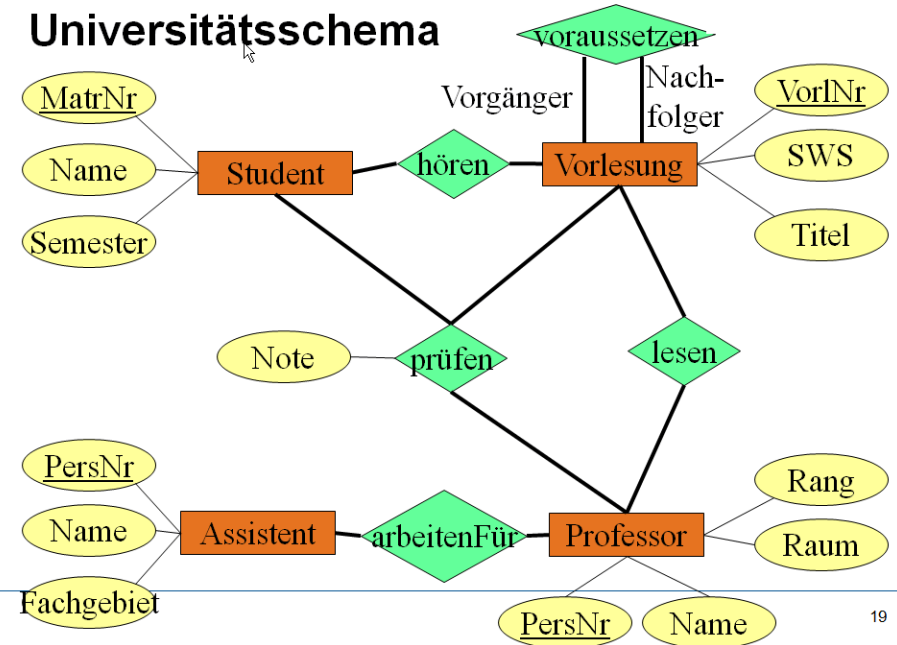
### Funktionalitäten



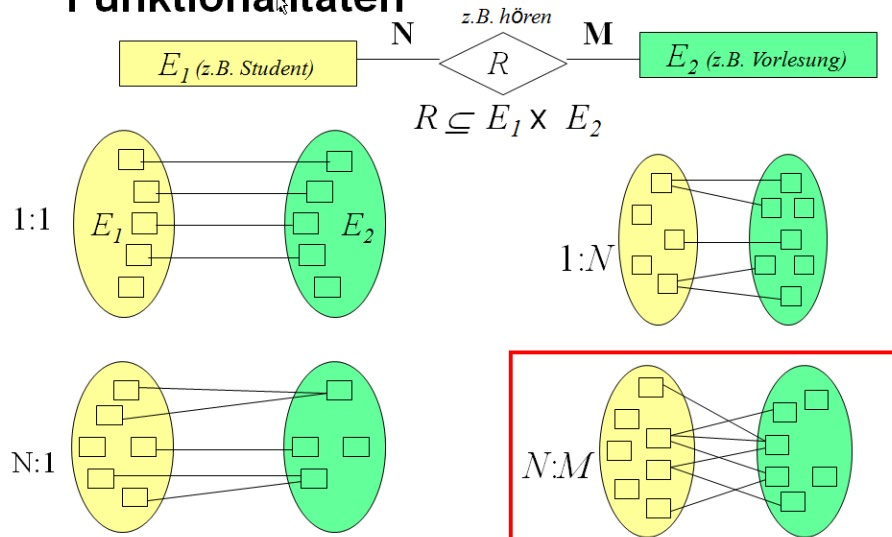
### Funktionalitäten



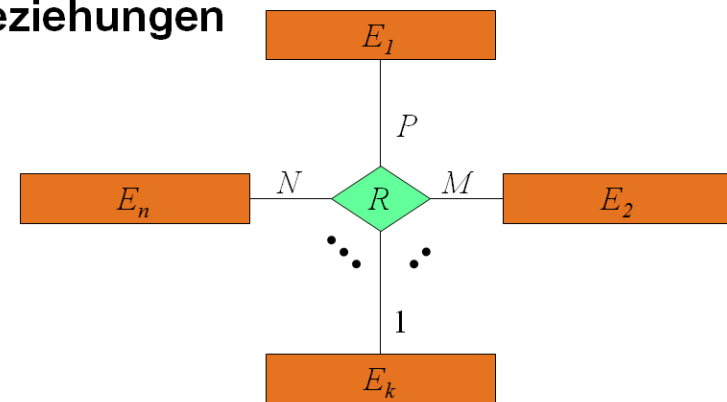
### Universitätsschema



### Funktionalitäten

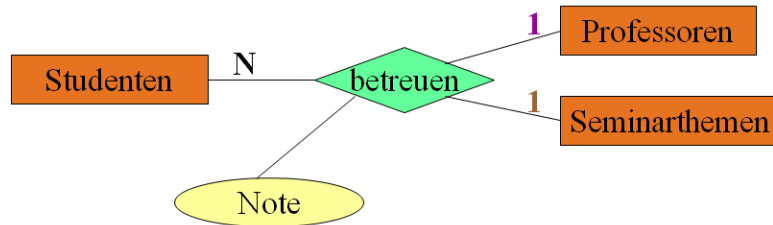


### Funktionalitäten bei n-stelligen Beziehungen



$$R: E_1 \times \dots \times E_{k-1} \times E_{k+1} \times \dots \times E_n \rightarrow E_k$$

## Beispiel-Beziehung: *betreuen*



dargestellt als partielle Funktionen:

*betreuen* : Seminarthemen  $\times$  Studenten  $\rightarrow$  Professoren

*betreuen* : Professoren  $\times$  Studenten  $\rightarrow$  Seminarthemen

25

## Dadurch erzwungene Konsistenzbedingungen

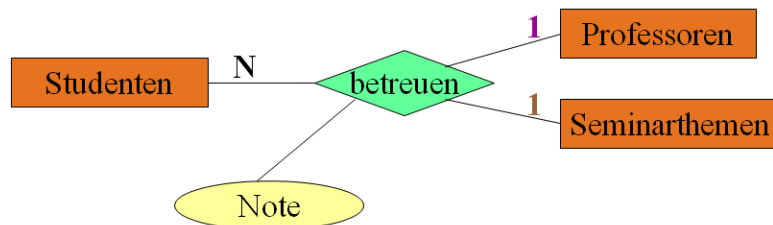
1. Studenten dürfen bei demselben Professor bzw. derselben Professorin **nur ein** Seminarthema "ableisten" (damit ein breites Spektrum abgedeckt wird). (vgl. 1 bei Seminarthema)
1. Studenten dürfen dasselbe Seminarthema **nur einmal** bearbeiten – sie dürfen also nicht bei anderen Professoren ein schon einmal erteiltes Seminarthema nochmals bearbeiten. (vgl. 1 bei Professor)

Es sind aber folgende Datenbankzustände nach wie vor möglich:

- Professoren können dasselbe Seminarthema „wiederverwenden“ – also dasselbe Thema auch mehreren Studenten erteilen.
- Ein Thema kann von mehreren Professoren vergeben werden – aber an unterschiedliche Studenten.

26

## Beispiel-Beziehung: *betreuen*



dargestellt als partielle Funktionen:

*betreuen* : Seminarthemen  $\times$  Studenten  $\rightarrow$  Professoren

*betreuen* : Professoren  $\times$  Studenten  $\rightarrow$  Seminarthemen

25

## Dadurch erzwungene Konsistenzbedingungen

1. Studenten dürfen bei demselben Professor bzw. derselben Professorin **nur ein** Seminarthema "ableisten" (damit ein breites Spektrum abgedeckt wird). (vgl. 1 bei Seminarthema)
1. Studenten dürfen dasselbe Seminarthema **nur einmal** bearbeiten – sie dürfen also nicht bei anderen Professoren ein schon einmal erteiltes Seminarthema nochmals bearbeiten. (vgl. 1 bei Professor)

Es sind aber folgende Datenbankzustände nach wie vor möglich:

- Professoren können dasselbe Seminarthema „wiederverwenden“ – also dasselbe Thema auch mehreren Studenten erteilen.
- Ein Thema kann von mehreren Professoren vergeben werden – aber an unterschiedliche Studenten.

26

## Dadurch erzwungene Konsistenzbedingungen

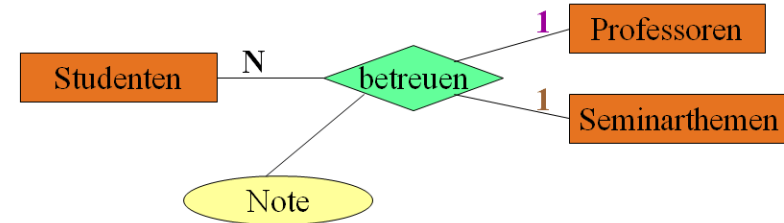
1. Studenten dürfen bei demselben Professor bzw. derselben Professorin **nur ein** Seminarthema "ableisten" (damit ein breites Spektrum abgedeckt wird). (vgl. 1 bei Seminarthema)
1. Studenten dürfen dasselbe Seminarthema **nur einmal** bearbeiten – sie dürfen also nicht bei anderen Professoren ein schon einmal erteiltes Seminarthema nochmals bearbeiten. (vgl. 1 bei Professor)

Es sind aber folgende Datenbankzustände nach wie vor möglich:

- Professoren können dasselbe Seminarthema „wiederverwenden“ – also dasselbe Thema auch mehreren Studenten erteilen.
- Ein Thema kann von mehreren Professoren vergeben werden – aber an unterschiedliche Studenten.

26

## Beispiel-Beziehung: *betreuen*



dargestellt als partielle Funktionen:

*betreuen* : Seminarthemen × Studenten → Professoren

*betreuen* : Professoren × Studenten → Seminarthemen

25

## Dadurch erzwungene Konsistenzbedingungen

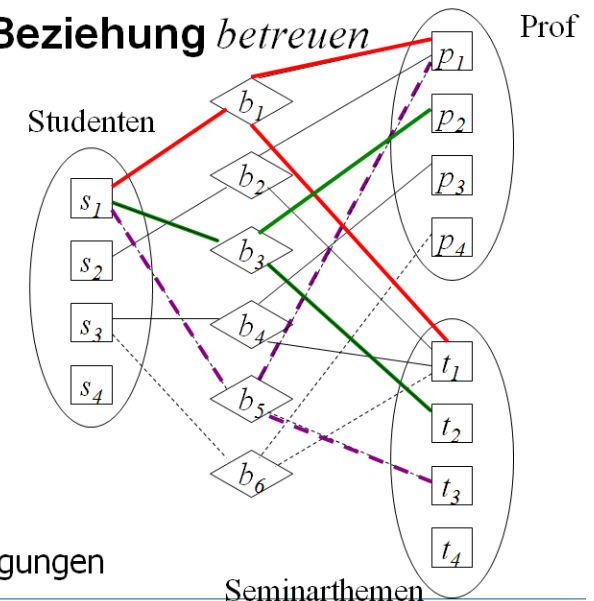
1. Studenten dürfen bei demselben Professor bzw. derselben Professorin **nur ein** Seminarthema "ableisten" (damit ein breites Spektrum abgedeckt wird). (vgl. 1 bei Seminarthema)
1. Studenten dürfen dasselbe Seminarthema **nur einmal** bearbeiten – sie dürfen also nicht bei anderen Professoren ein schon einmal erteiltes Seminarthema nochmals bearbeiten. (vgl. 1 bei Professor)

Es sind aber folgende Datenbankzustände nach wie vor möglich:

- Professoren können dasselbe Seminarthema „wiederverwenden“ – also dasselbe Thema auch mehreren Studenten erteilen.
- Ein Thema kann von mehreren Professoren vergeben werden – aber an unterschiedliche Studenten.

26

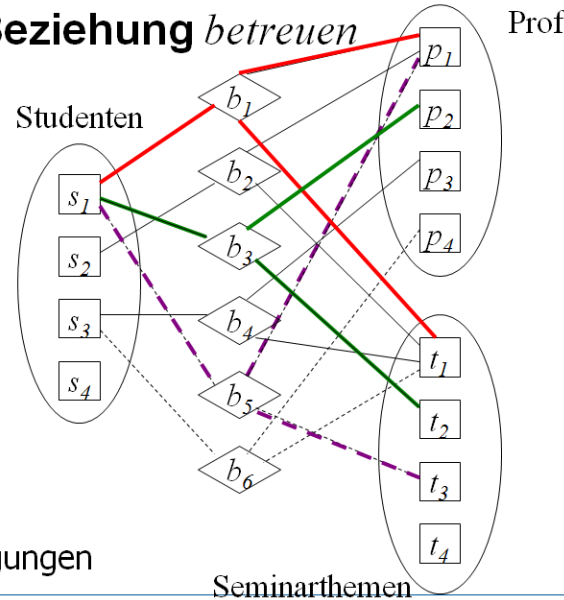
## Ausprägung der Beziehung *betreuen*



Gestrichelte Linien markieren illegale Ausprägungen

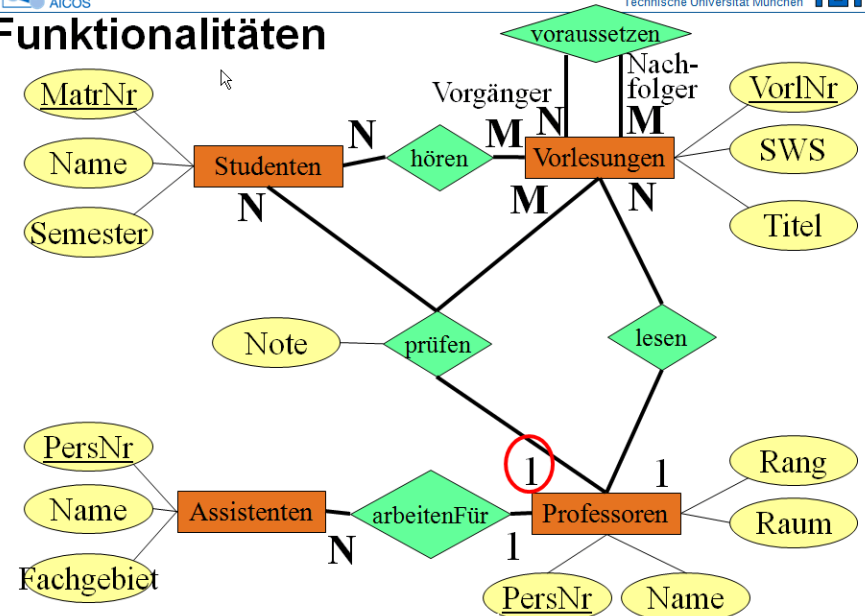
27

## Ausprägung der Beziehung *betreuen*



Gestrichelte Linien markieren illegale Ausprägungen

## Funktionalitäten



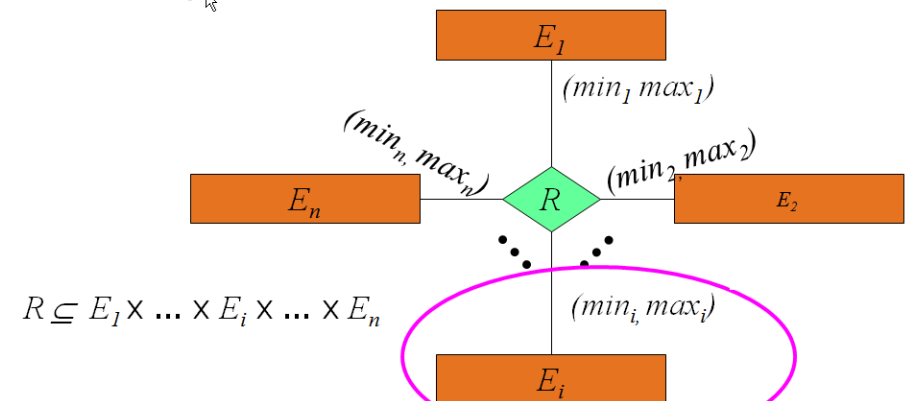
## Dadurch erzwungene Konsistenzbedingungen

1. Studenten dürfen bei demselben Professor bzw. derselben Professorin **nur ein** Seminarthema "ableisten" (damit ein breites Spektrum abgedeckt wird). (vgl. 1 bei Seminarthema)
1. Studenten dürfen dasselbe Seminarthema **nur einmal** bearbeiten – sie dürfen also nicht bei anderen Professoren ein schon einmal erteiltes Seminarthema nochmals bearbeiten. (vgl. 1 bei Professor)

Es sind aber folgende Datenbankzustände nach wie vor möglich:

- Professoren können dasselbe Seminarthema „wiederverwenden“ – also dasselbe Thema auch mehreren Studenten erteilen.
- Ein Thema kann von mehreren Professoren vergeben werden – aber an unterschiedliche Studenten.

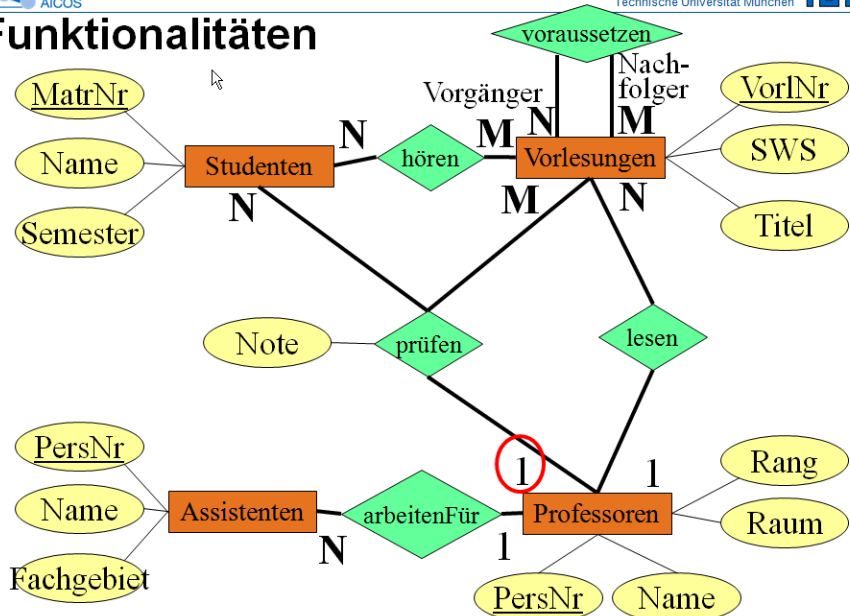
## (min, max)-Notation



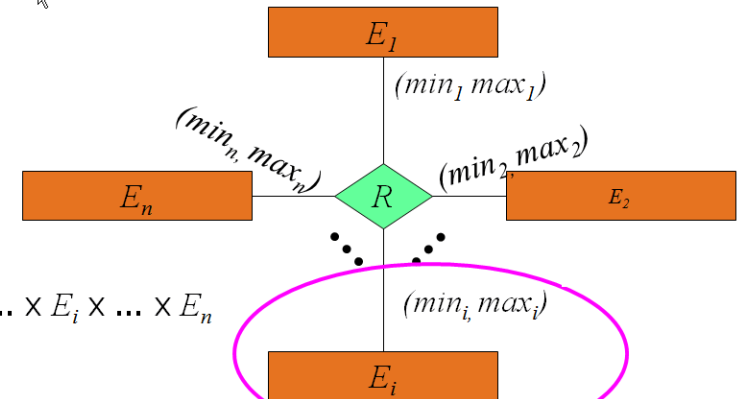
Für jedes  $e_i \in E_i$  gibt es

- Mindestens  $min_i$  Tupel der Art  $(\dots, e_i, \dots)$  und
- Höchstens  $max_i$  viele Tupel der Art  $(\dots, e_i, \dots) \in R$

# Funktionalitäten



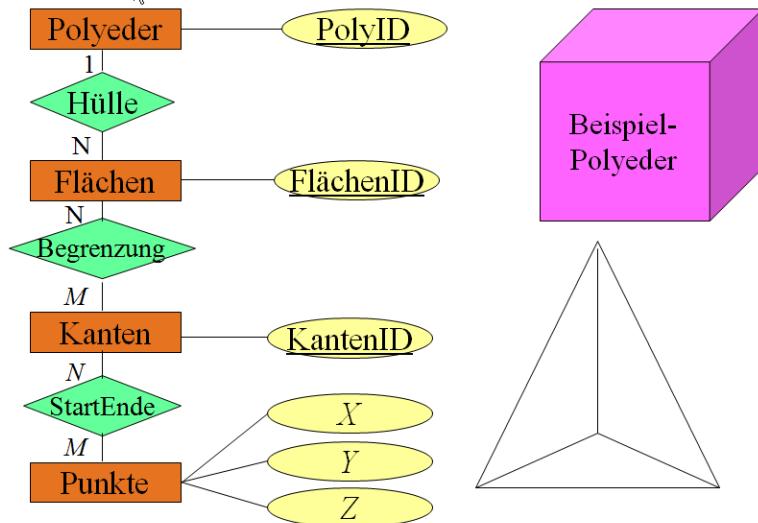
# (min, max)-Notation



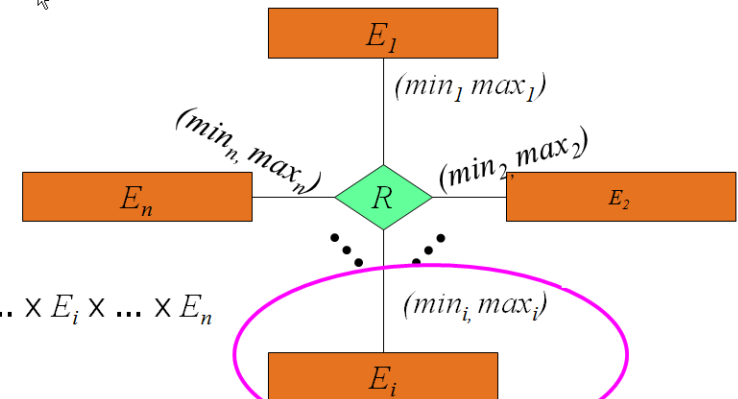
$$R \subseteq E_1 \times \dots \times E_i \times \dots \times E_n$$

- Für jedes  $e_i \in E_i$  gibt es
- Mindestens  $min_i$  Tupel der Art  $(\dots, e_i, \dots)$  und
  - Höchstens  $max_i$  viele Tupel der Art  $(\dots, e_i, \dots) \in R$

# Begrenzungsflächendarstellung



# (min, max)-Notation

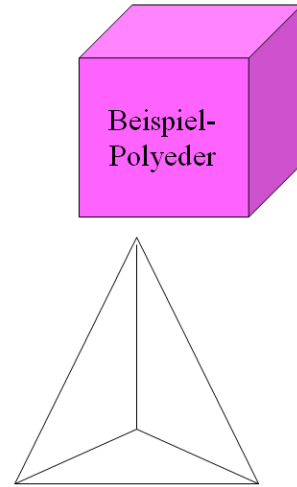
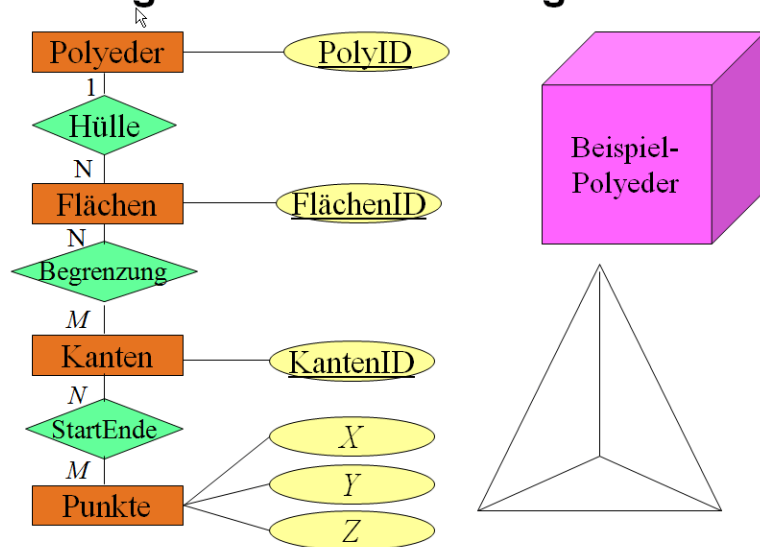


$$R \subseteq E_1 \times \dots \times E_i \times \dots \times E_n$$

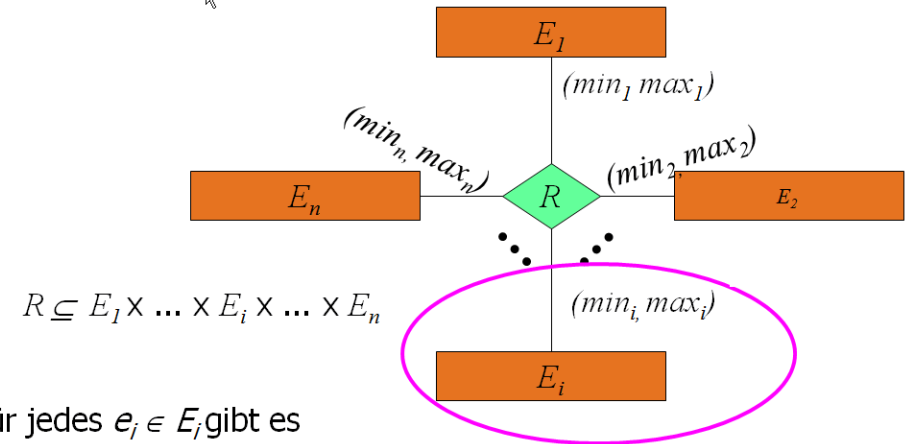
- Für jedes  $e_i \in E_i$  gibt es
- Mindestens  $min_i$  Tupel der Art  $(\dots, e_i, \dots)$  und
  - Höchstens  $max_i$  viele Tupel der Art  $(\dots, e_i, \dots) \in R$



# Begrenzungsflächendarstellung

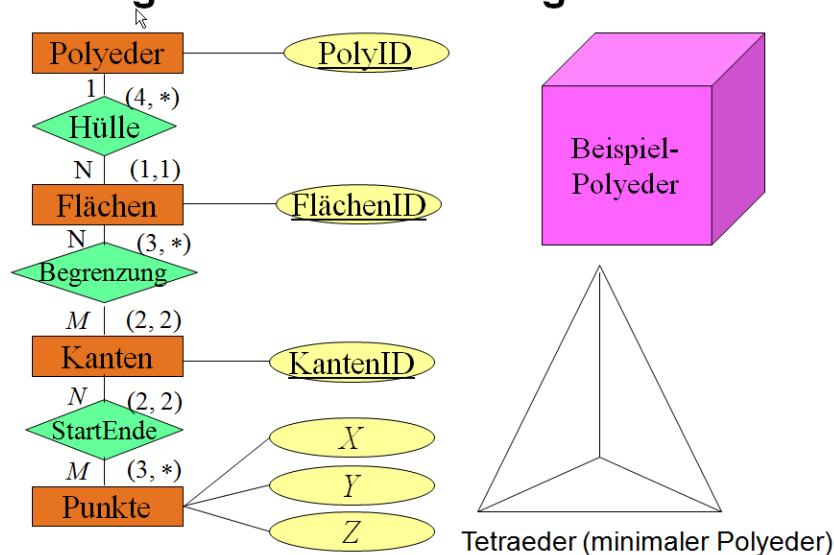


# (min, max)-Notation



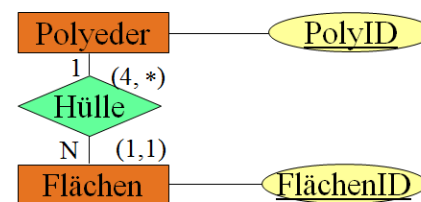
- Für jedes  $e_i \in E_i$  gibt es
- Mindestens  $min_i$  Tupel der Art  $(\dots, e_i, \dots)$  und
  - Höchstens  $max_i$  viele Tupel der Art  $(\dots, e_i, \dots) \in R$

# Begrenzungsflächendarstellung



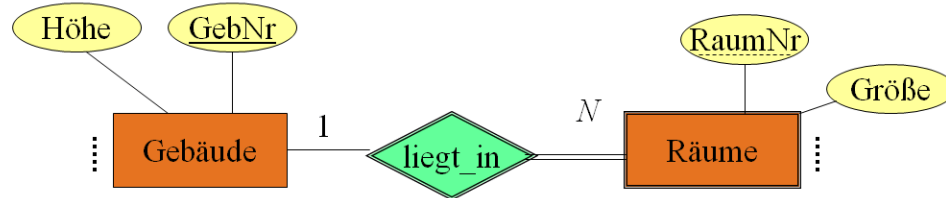
Tetraeder (minimaler Polyeder)

# Beziehung Hülle als Tabelle / Relation:



PolyID	FlächenID
Tetraeder_Horst	Fläche_1
Tetraeder_Horst	Fläche_2
Tetraeder_Horst	Fläche_3
Tetraeder_Horst	Fläche_4
Oktaeder_Heiner	Fl_a
Oktaeder_Heiner	Fl_b
Oktaeder_Heiner	Fl_c
...	...

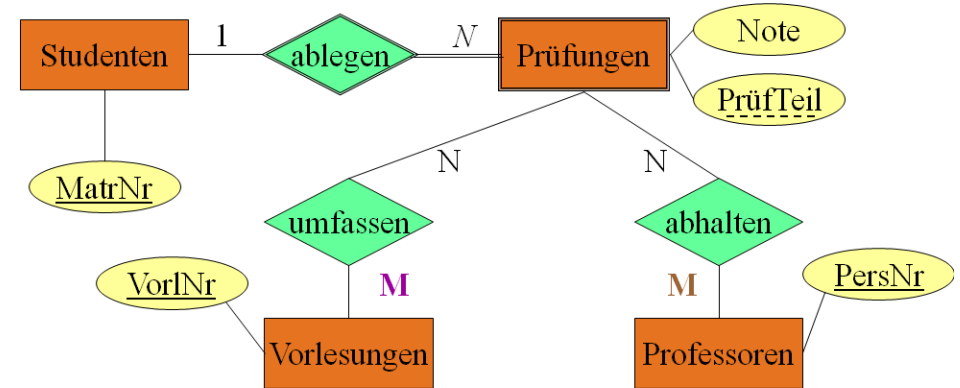
## Schwache, existenzabhängige Entities



- Beziehung zwischen "starken" und schwachem Typ ist immer 1:N (oder 1:1 in seltenen Fällen)
- Warum kann das keine N:M-Beziehung sein?
- RaumNr ist nur innerhalb eines Gebäudes eindeutig
- Schlüssel ist: GebNr **und** RaumNr

33

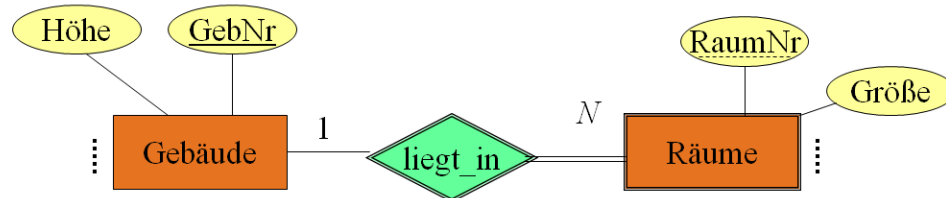
## Prüfungen als schwacher Entitytyp



- in einer Prüfung werden mehrere Vorlesungen abgefragt
- eine Prüfung kann von mehreren Prüfern abgehalten werden

34

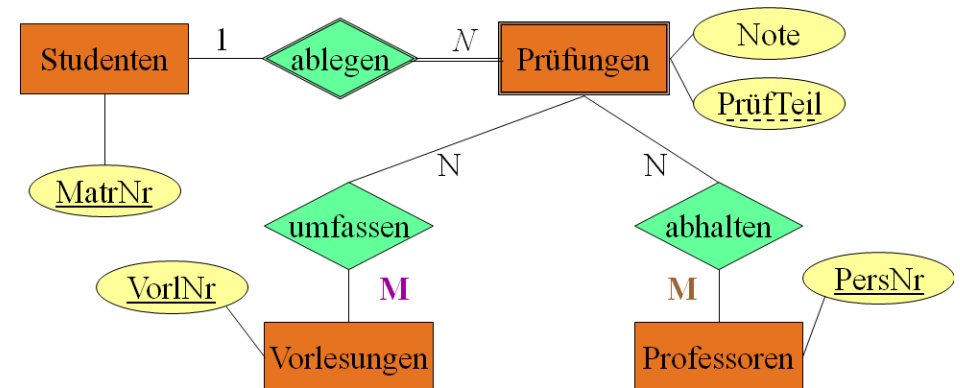
## Schwache, existenzabhängige Entities



- Beziehung zwischen "starken" und schwachem Typ ist immer 1:N (oder 1:1 in seltenen Fällen)
- Warum kann das keine N:M-Beziehung sein?
- RaumNr ist nur innerhalb eines Gebäudes eindeutig
- Schlüssel ist: GebNr **und** RaumNr

33

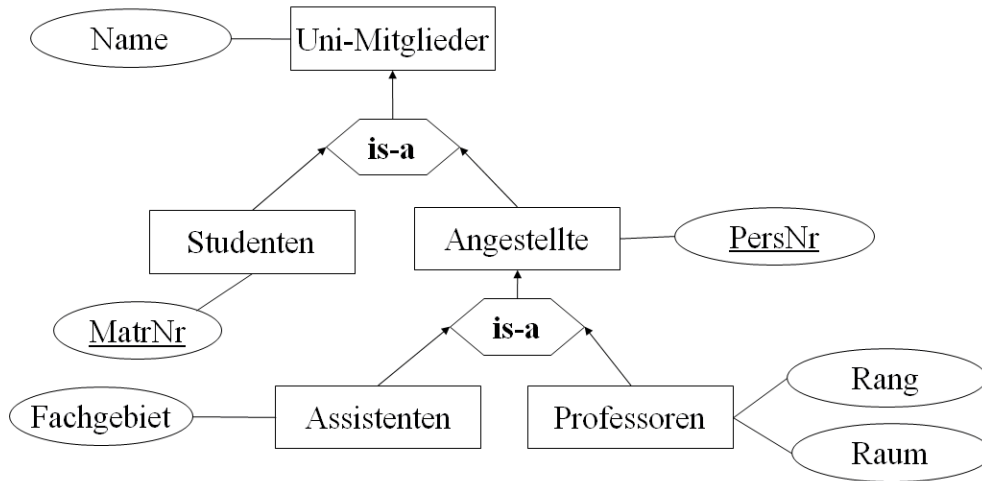
## Prüfungen als schwacher Entitytyp



- in einer Prüfung werden mehrere Vorlesungen abgefragt
- eine Prüfung kann von mehreren Prüfern abgehalten werden

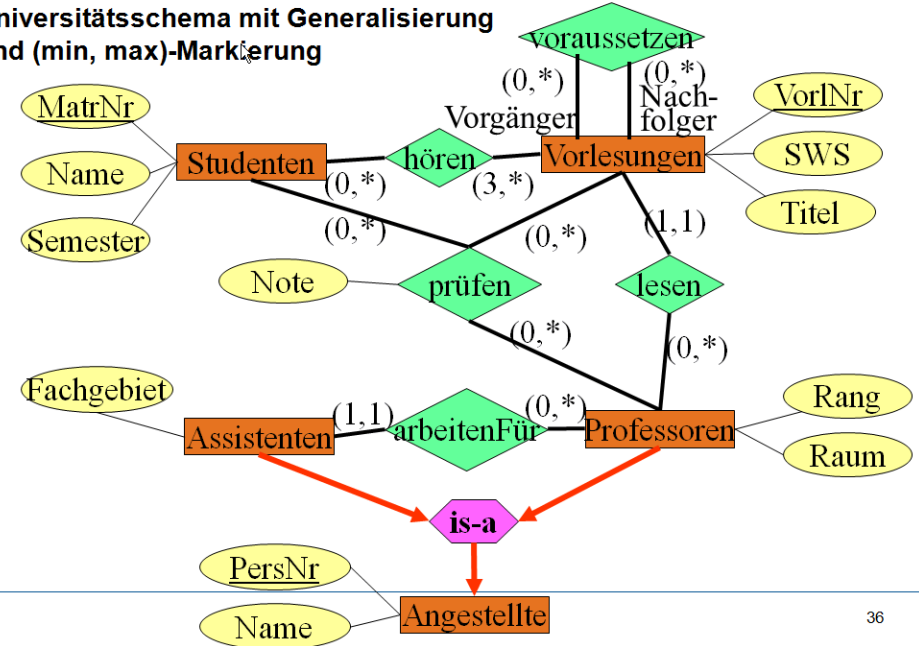
34

# Generalisierung/Spezialisierung



35

# Universitätsschema mit Generalisierung und (min, max)-Markierung



36

## Aufgabe 2.1

Charakterisieren Sie die 1:1-, 1:N-, N:1- und N:M-Beziehungstypen mittels der (min, max)-Notation. Für eine abstrakte binäre Beziehung R zwischen den beiden Entitätstypen E<sub>1</sub> und E<sub>2</sub> sollen jeweils die (min<sub>1</sub>, max<sub>1</sub>)- und (min<sub>2</sub>, max<sub>2</sub>)-Wertepaare angegeben werden, die sich aus den (größer) Funktionalitätsangaben herleiten lassen.



$F_1 : F_2$	$(min_1, max_1)$	$(min_2, max_2)$
1 : 1		
1 : N		
N : 1		
N : M		

40

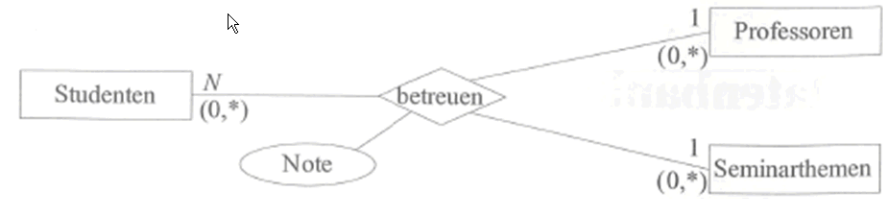


$F_1 : F_2$	$(min_1, max_1)$	$(min_2, max_2)$
1 : 1		
1 : N		
N : 1		
N : M		

41

### Aufgabe 2.2

Zeigen Sie, dass die Ausdruckskraft der Funktionalitätsangaben und der  $(min, max)$ -Angaben bei  $n$ -stelligen Beziehungen mit  $n > 2$  unvergleichbar ist: Finden Sie realistische Beispiele von Konsistenzbedingungen, die mit Funktionalitätsangaben, aber nicht mit  $(min, max)$ -Angaben ausdrückbar sind, und wiederum andere Konsistenzbedingungen, die mit der  $(min, max)$ -Angabe formulierbar sind aber nicht durch Funktionalitätseinschränkungen.



partielle Funktionen:

$$\begin{aligned} \text{betreuen} &: \text{Professoren} \times \text{Studenten} \rightarrow \text{Seminarthemen} \\ \text{betreuen} &: \text{Seminarthemen} \times \text{Studenten} \rightarrow \text{Professoren} \end{aligned}$$

Konsistenzbedingungen:

- Studenten dürfen bei einem Professor nur **ein** Seminarthema bearbeiten.
- Studenten dürfen dasselbe Thema nur bei **einem** Professor bearbeiten.

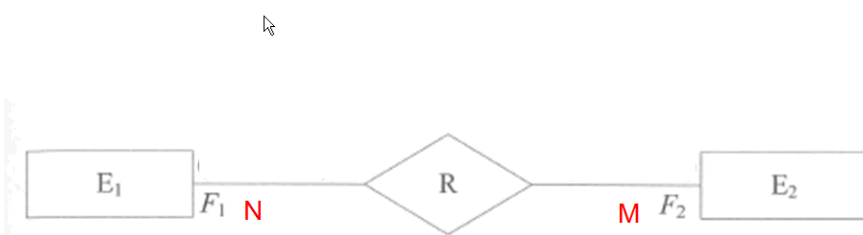


partielle Funktionen:

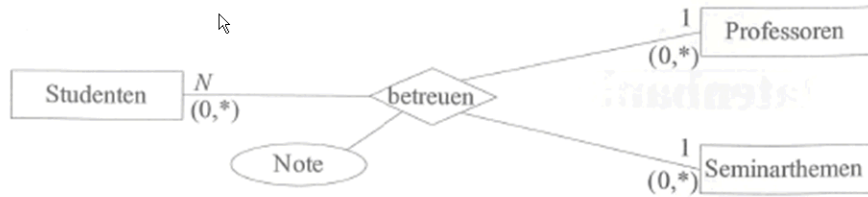
$$\begin{aligned} \text{betreuen} &: \text{Professoren} \times \text{Studenten} \rightarrow \text{Seminarthemen} \\ \text{betreuen} &: \text{Seminarthemen} \times \text{Studenten} \rightarrow \text{Professoren} \end{aligned}$$

Konsistenzbedingungen:

- Studenten dürfen bei einem Professor nur **ein** Seminarthema bearbeiten.
- Studenten dürfen dasselbe Thema nur bei **einem** Professor bearbeiten.



$F_1 : F_2$	$(min_1, max_1)$	$(min_2, max_2)$
1 : 1	(0, 1)	(0, 1)
1 : N	(0, *)	(0, 1)
N : 1	(0, 1)	(0, *)
N : M		



partielle Funktionen:

$\text{betreuen} : \text{Professoren} \times \text{Studenten} \rightarrow \text{Seminarthemen}$   
 $\text{betreuen} : \text{Seminarthemen} \times \text{Studenten} \rightarrow \text{Professoren}$

Konsistenzbedingungen:

- Studenten dürfen bei einem Professor nur **ein** Seminarthema bearbeiten.
- Studenten dürfen dasselbe Thema nur bei **einem** Professor bearbeiten.

51



partielle Funktionen:

$\text{betreuen} : \text{Professoren} \times \text{Studenten} \rightarrow \text{Seminarthemen}$   
 $\text{betreuen} : \text{Seminarthemen} \times \text{Studenten} \rightarrow \text{Professoren}$

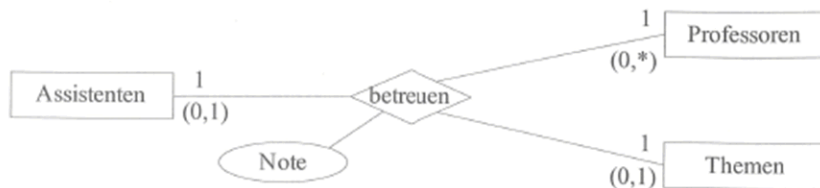
Konsistenzbedingungen:

- Studenten dürfen bei einem Professor nur **ein** Seminarthema bearbeiten.
- Studenten dürfen dasselbe Thema nur bei **einem** Professor bearbeiten.

Müssen aus Modellierungsgründen so sein (bspw: Profs verwenden Seminarthemen erneut), aber können alleine dargestellt zusammen die Konsistenzbedingungen verletzen

52

weiteres Bsp:

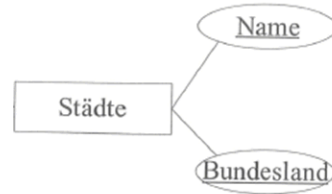
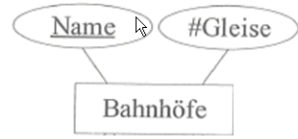


53

### Aufgabe 2.6

Modellieren Sie ein Zugauskunftssystem, in dem die wichtigsten Züge (z.B. die Intercity- und Eurocity-Züge) repräsentiert werden. Aus dem System sollen die Start- und Zielbahnhöfe und die durch den Zug verbundenen Bahnhöfe einschließlich Ankunfts- und Abfahrtszeiten ersichtlich sein. Geben Sie die Funktionalitäten der Beziehungstypen an.

54



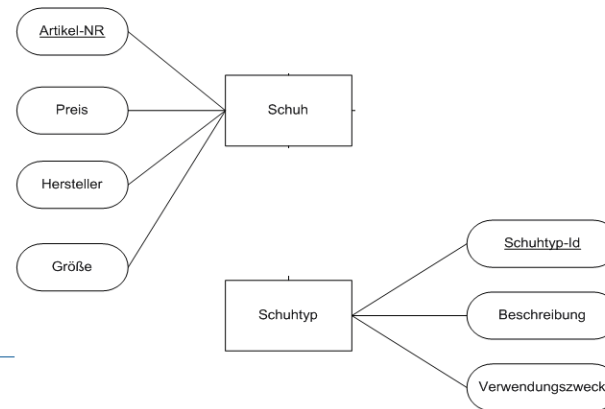
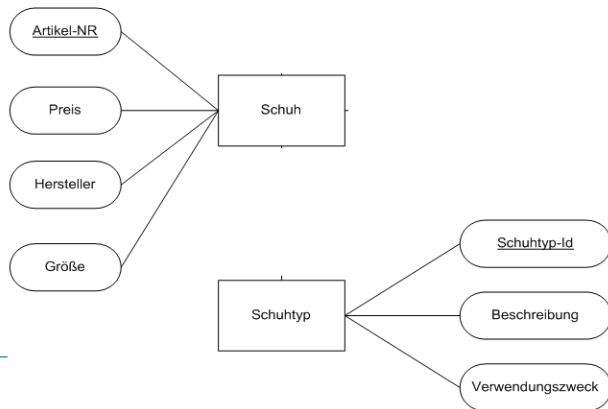
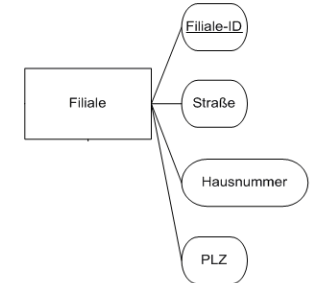
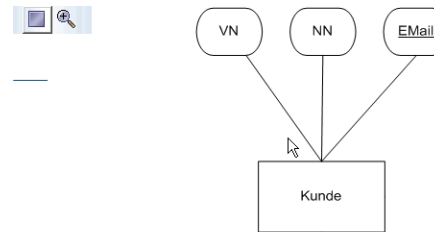
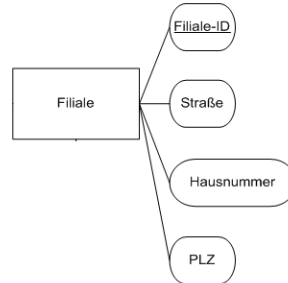
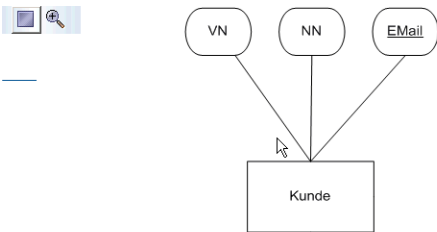
# Klausuraufgabe - April 2011

In der Datenbank einer Schuhladenkette sollen Informationen zu den Kunden, den einzelnen Filialen und den verfügbaren Schuhen gespeichert werden.

Erstellen Sie zunächst ein Entity-Relationship-Modell mit folgenden Entity- und Relationship-Typen und den zugehörigen Attributen:

- Kunde*: mit den Attributen Vorname, Nachname, E-Mail-Adresse
- Filiale*: mit den Attributen Filial-ID, Straße, Hausnummer, Postleitzahl
- Schuh*: mit den Attributen Artikelnummer, Preis, Hersteller, Größe
- Schuhtyp*: mit den Attributen Schuhtyp-ID, Beschreibung, Verwendungszweck
- besucht*: zwischen Kunde und Filiale (mit dem Attribut Datum)
- kauft*: eindeutige Zuordnung zwischen Kunde und Schuh
- verkauft*: eindeutige Zuordnung zwischen Filiale und Schuh
- von-Typ*: eindeutige Zuordnung zwischen Schuh und Schuhtyp

**Hinweis:** Vergessen Sie nicht, die Primärschlüssel zu markieren und die Funktionalitäten einzutragen! (6 P)



## Klausuraufgabe - April 2011

In der Datenbank einer Schuhladenkette sollen Informationen zu den Kunden, den einzelnen Filialen und den verfügbaren Schuhen gespeichert werden.

Erstellen Sie zunächst ein Entity-Relationship-Modell mit folgenden Entity- und Relationship-Typen und den zugehörigen Attributen:

*Kunde*: mit den Attributen Vorname, Nachname, E-Mail-Adresse

*Filiale*: mit den Attributen Filial-ID, Straße, Hausnummer, Postleitzahl

*Schuh*: mit den Attributen Artikelnummer, Preis, Hersteller, Größe

*Schuhtyp*: mit den Attributen Schuhtyp-ID, Beschreibung, Verwendungszweck

*besucht*: zwischen Kunde und Filiale (mit dem Attribut Datum)

*kauft*: eindeutige Zuordnung zwischen Kunde und Schuh

*verkauft*: eindeutige Zuordnung zwischen Filiale und Schuh

*von-Typ*: eindeutige Zuordnung zwischen Schuh und Schuhtyp

**Hinweis:** Vergessen Sie nicht, die Primärschlüssel zu markieren und die Funktionalitäten einzutragen! (6 P)