

Skript zur Vorlesung:

# Einführung in die Informatik: Systeme und Anwendungen

Sommersemester 2013

## Kapitel 3: Datenbanksysteme

Vorlesung: PD Dr. Peer Kröger

Übungen: Johannes Niedermayer

Skript © 2004 Christian Böhm, Peer Kröger

[http://www.dbs.ifi.lmu.de/cms/Einfuehrung\\_in\\_die\\_Informatik\\_Systeme\\_und\\_Anwendungen](http://www.dbs.ifi.lmu.de/cms/Einfuehrung_in_die_Informatik_Systeme_und_Anwendungen)



# Überblick

3.1 Einleitung

3.2 Das Relationale Modell

3.3 Die Relationale Algebra

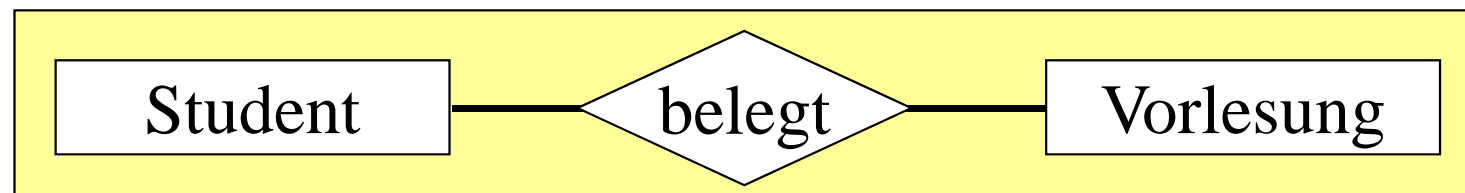
3.4 Mehr zu SQL

3.5 Das E/R-Modell

## 3.5 Das E/R-Modell

### • Schema-Entwurf

- Generelle Aufgabe:  
Finde eine formale Beschreibung (Modell) für einen zu modellierenden Teil der realen Welt
- Zwischenstufen:
  - Beschreibung durch natürliche Sprache (Pflichtenheft):  
Beispiel: *In der Datenbank sollen alle Studierenden mit den durch sie belegten Lehrveranstaltungen gespeichert sein*
  - Beschreibung durch abstrakte grafische Darstellungen:



- Beschreibung im relationalen Modell:  
*create table student (...);*  
*create table vorlesung (...);*

## 3.5 Das E/R-Modell

- Das Entity/Relationship Modell
  - Dient dazu, für einen Ausschnitt der realen Welt ein konzeptionelles Schema zu erstellen
  - Grafische Darstellung: E/R-Diagramm
  - Maschinenfernes Datenmodell
  - Hohes Abstraktionsniveau
  - Überlegungen zur Effizienz spielen keine Rolle
  - Das E/R-Modell muss in ein relationales Schema überführt werden
    - Einfache Grundregeln zur Transformation
    - Gewinnung eines **effizienten** Schemas erfordert tiefes Verständnis vom Zielmodell

## 3.5 Das E/R-Modell

- Elemente des E/R-Modells

- **Entities:**  
(eigentlich: Entity Sets)  
Objekttypen
- **Attribute:**  
Eigenschaften
- **Relationships:**  
Beziehungen zw. Entities

Student

Name

belegt

Entscheidende Aufgabe des Schema-Entwurfs:

- Finde **geeignete** Entities, Attribute und Relationships

## 3.5 Das E/R-Modell

- Entities

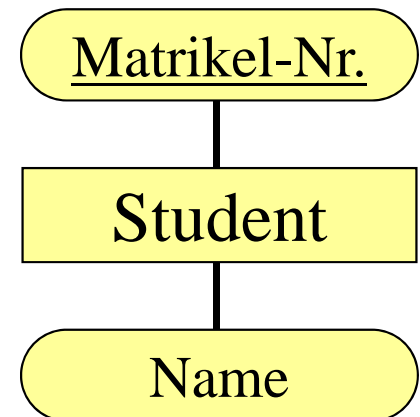
- Objekte, Typen, "Seiendes"
- Objekte der realen Welt, unterscheidbar
- Bsp: Mensch, Haus, Vorlesung, Bestellung, ...

- Attribute

- Entities durch charakterisierende Eigenschaften beschrieben
- Einfache Datentypen wie INT, STRING usw.
- Bsp: Farbe, Gewicht, Name, Titel, ...
- Häufig beschränkt man sich auf die wichtigsten Attribute

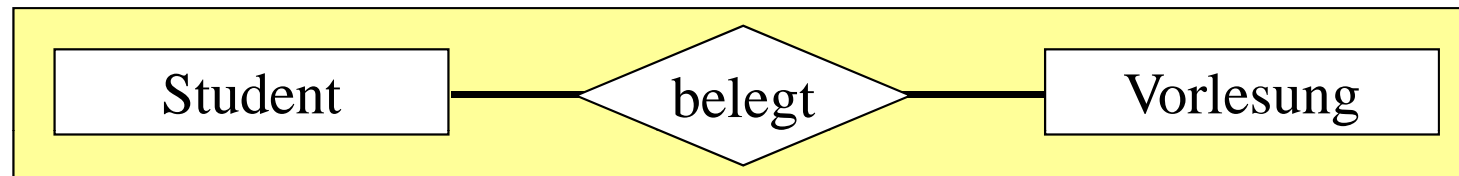
- Schlüssel

- ähnlich definiert wie im relationalen Modell
- Primärschlüssel-Attribute werden unterstrichen



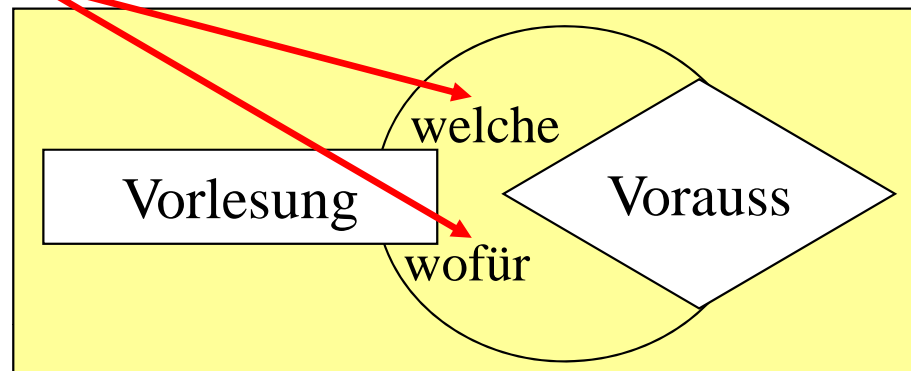
## 3.5 Das E/R-Modell

- Relationships (Beziehungen)  
Stellen Zusammenhänge zwischen Entities dar
- Beispiele:



Ausprägung: belegt (Anton, Einführung in die Informatik)  
 belegt (Berta, Einführung in die Informatik)  
 belegt (Caesar, Wissensrepräsentation)  
 belegt (Anton, Datenbanksysteme 1)

unterschiedliche *Rollen*



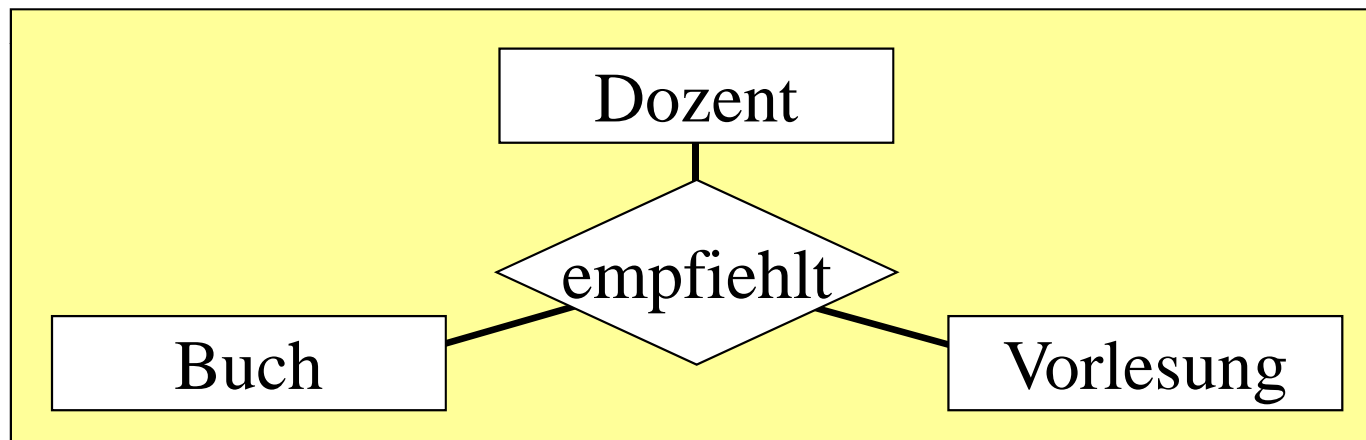
Ausprägung:  
 Vorauss (I 1, DBS 1)  
 Vorauss (I 1, Software-Eng.)  
 Vorauss (DBS 1, DBS 2)  
 Vorauss (I 1, Wissensrepr.)

## 3.5 Das E/R-Modell

- Relationships können eigene Attribute haben.

**Beispiel:** Student *belegt* Vorlesung  
*belegt* hat Attribut *Note*

- Mehrstellige (ternäre usw.) Relationships:

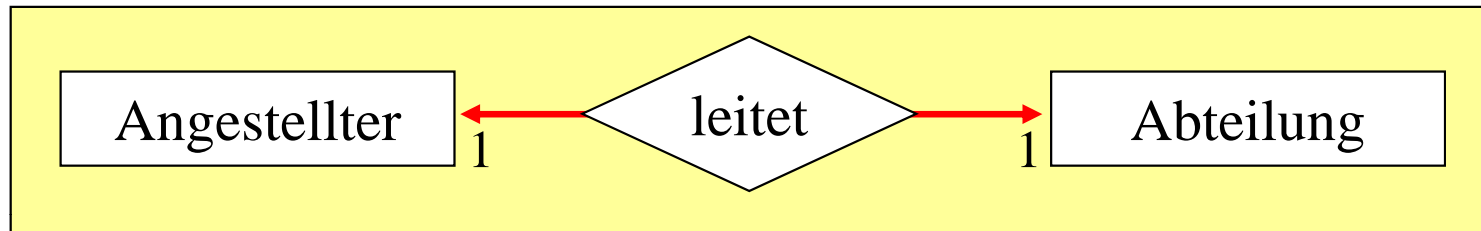


Ausprägung: empfiehlt (Kröger, Heuer&Saake, DBS 1)  
 empfiehlt (Kröger, Kemper&Eickler, DBS 1)  
 empfiehlt (Kröger, Bishop, Einführung in die Informatik)  
 empfiehlt (Kröger, Bishop, Programmierkurs)



## 3.5 Das E/R-Modell

- Funktionalität von Relationships
  - 1:1-Beziehung (one-to-one-relationship):

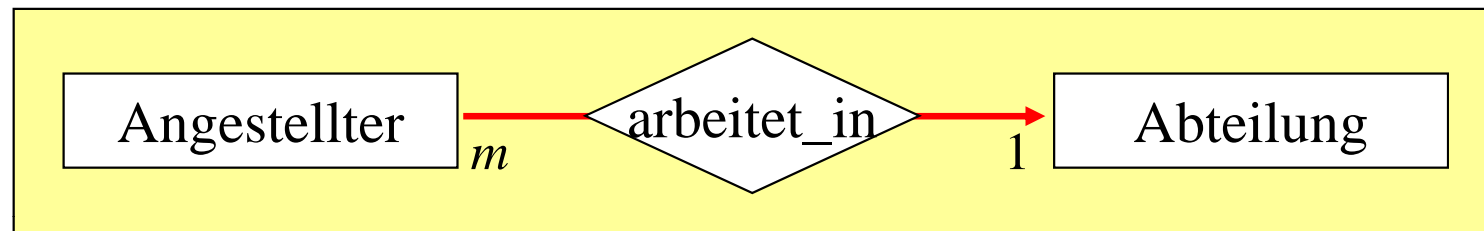


- Charakteristik:  
Jedes Objekt aus dem linken Entity steht in Beziehung zu **genau einem** Objekt aus dem rechten Entity **und umgekehrt**.
- Grafische Notation: Pfeile auf jeder Seite

Beispiel gilt unter der Voraussetzung, dass jede Abteilung max. einen Leiter hat und kein Angestellter mehrere Abteilungen leitet

## 3.5 Das E/R-Modell

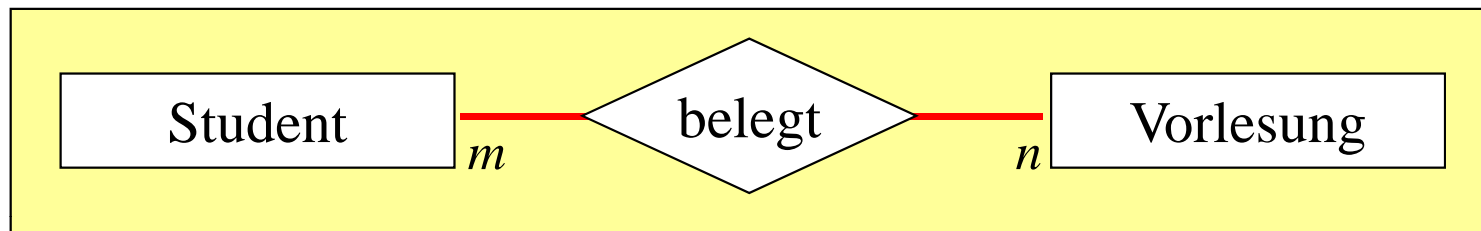
- Funktionalität von Relationships
  - $m:1$ -Beziehung (many-to-one-relationship)



- Charakteristik:  
Jedes Objekt auf der "*many*"-Seite steht in Beziehung zu ***höchstens*** einem Objekt auf der "*one*"-Seite (im Allgemeinen nicht umgekehrt)
- Grafische Notation:  
Pfeil in Richtung zur "*one*"-Seite

## 3.5 Das E/R-Modell

- Funktionalität von Relationships
  - $m:n$ -Beziehung (many-to-many-relationship)

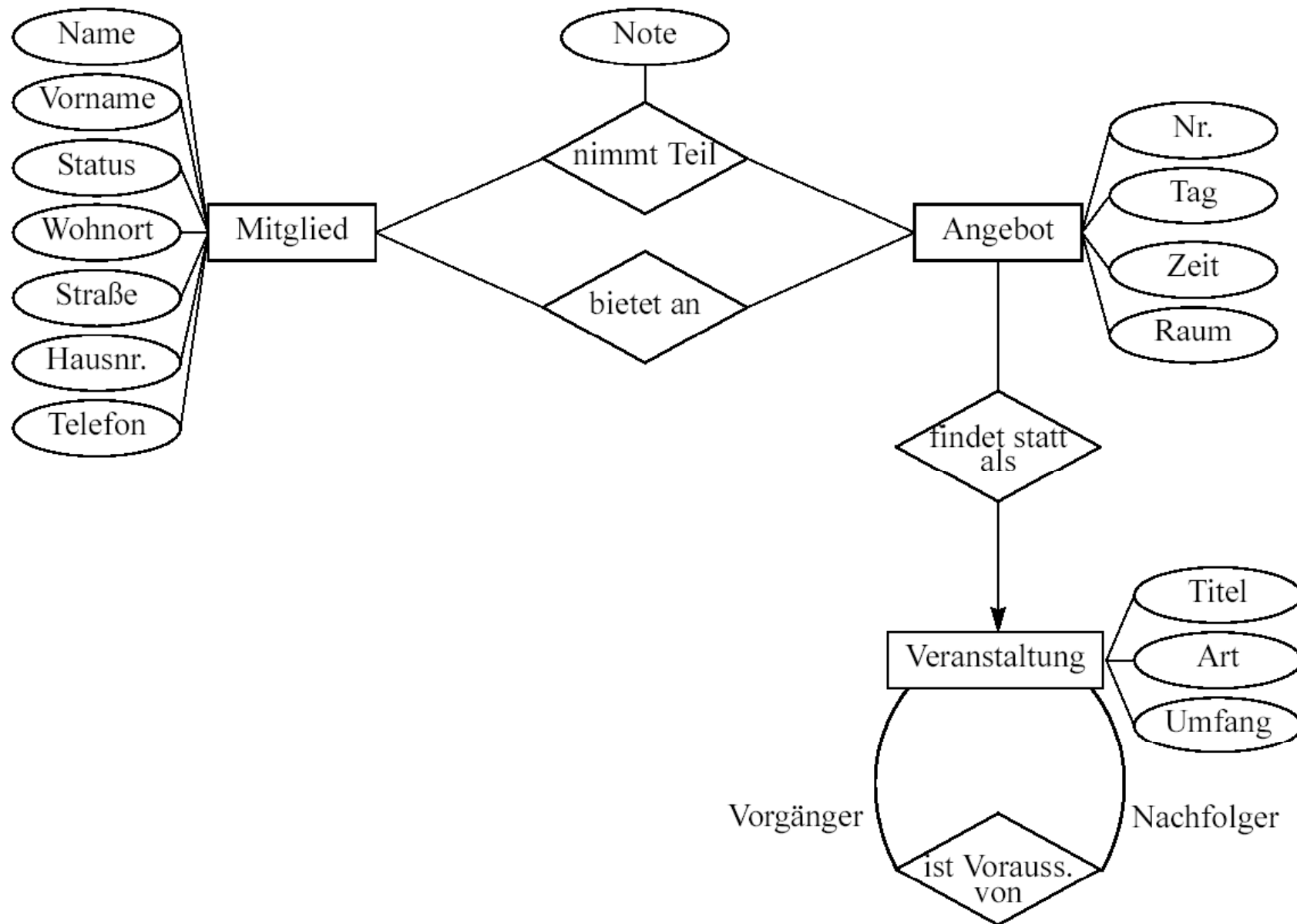


- Charakteristik:  
Jedes Objekt auf der linken Seite **kann** zu **mehreren** Objekten auf der rechten-Seite in Beziehung stehen (d.h. keine Einschränkung)
- Grafische Notation:  
Kein Pfeil

### Beispiel-Ausprägung:

belegt (Anton, Einführung in die Informatik)  
 belegt (Berta, Einführung in die Informatik)  
 belegt (Caesar, Wissensrepräsentation)  
 belegt (Anton, Datenbanksysteme 1)

## 3.5 Das E/R-Modell

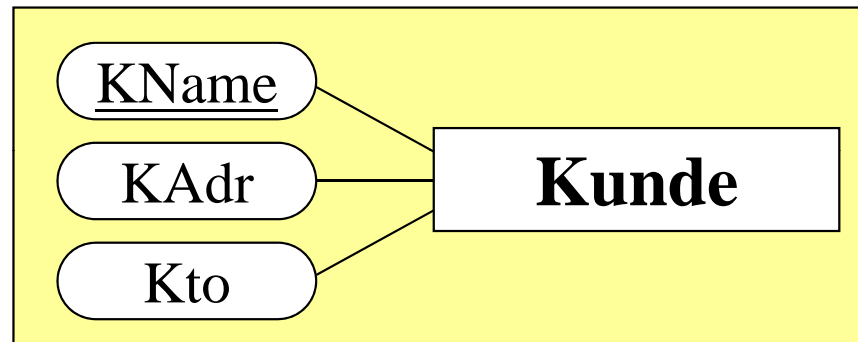


## 3.5 Das E/R-Modell

### • Vom E/R-Modell zur Relation

Einfache Umsetzungsregeln:

- Entities und Attribute:



- Jedem Entity-Typ wird eine Relation zugeordnet
- Jedes Attribut des Entity wird ein Attribut der Relation
- Der Primärschlüssel des Entity wird Primärschlüssel der Relation
- Attribute können im weiteren Verlauf dazukommen

**Kunde** (KName, KAdr, Kto)

## 3.5 Das E/R-Modell

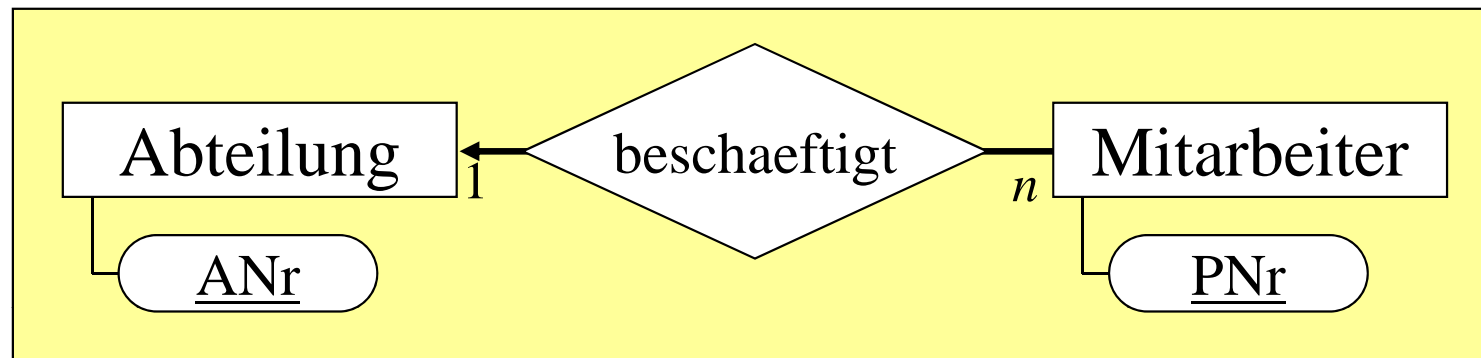
- Bei Relationships:  
Umsetzung abh. von Funktionalität/Kardinalität:
    - 1:1
    - 1:n
    - n:m
- } Zusätzliche Attribute in bestehende Relationen
- Erzeugung einer zusätzlichen Relation

Die ersten beiden Funktionalitäten sind Spezialfälle der dritten. Deshalb ist es immer **auch** möglich, zusätzliche Relationen einzuführen, jedoch nicht erforderlich.

Bei den ersten beiden kann man **alternativ** eine andere Lösung wählen.

## 3.5 Das E/R-Modell

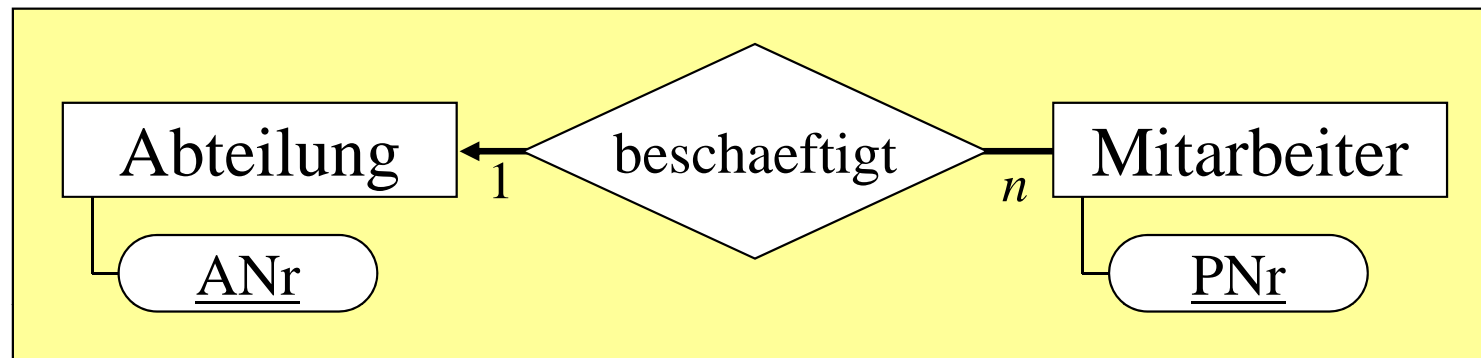
### – *Alternative* Lösung: One-To-Many Relationships:



- Keine zusätzliche Tabelle wird angelegt
- Der Primärschlüssel der Relation auf der **one**-Seite der Relationship kommt in die Relation der **many**-Seite (Umbenennung bei Namenskonflikten erforderlich!!!)
- Die neu eingeführten Attribute werden Fremdschlüssel
- Die Primärschlüssel der Relationen ändern sich nicht
- Attribute der Relationship werden ebenfalls in die Relation der **many**-Seite genommen (kein Fremdschlüssel)

## 3.5 Das E/R-Modell

- Beispiel One-To-Many-Relationship:



Abteilung (ANr, Bezeichnung, ...)

Mitarbeiter (PNr, Name, Vorname, ..., ANr)

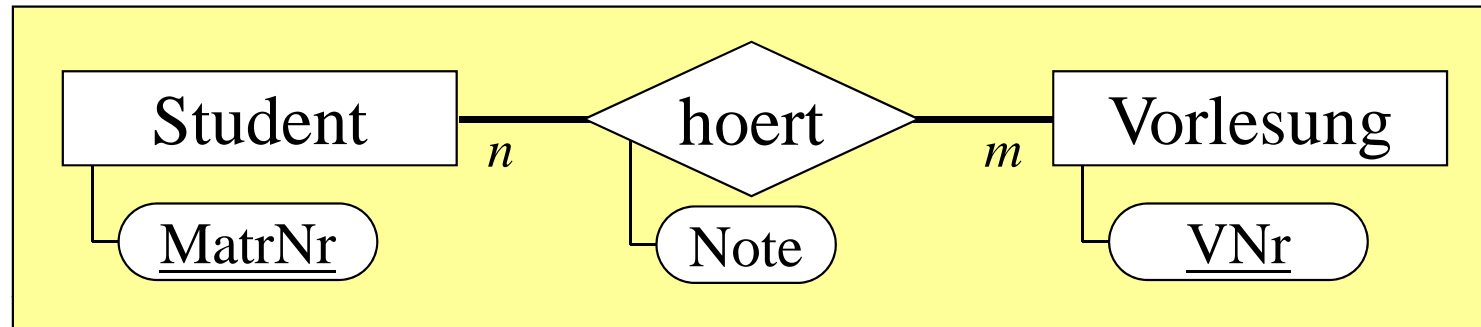
```

create table Mitarbeiter (
    PNr char(3) primary key,
    ...
    ANr char(3) references Abteilung (ANr) );
  
```



## 3.5 Das E/R-Modell

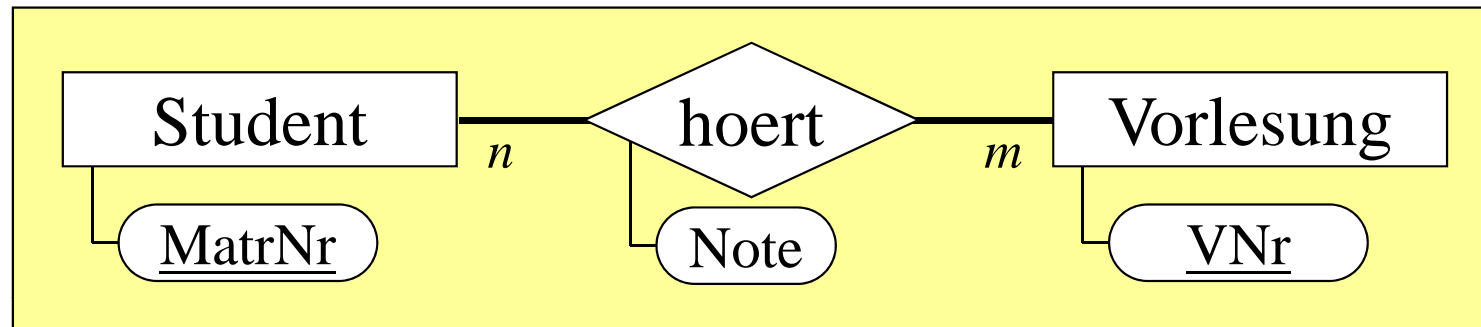
### – Many-To-Many-Relationships:



- Einführung einer zusätzlichen Relation mit dem Namen der Relationship
- Attribute: Die Primärschlüssel-Attribute der Relationen, die den Entities beider Seiten zugeordnet sind
- Diese Attribute sind jeweils Fremdschlüssel
- Zusammen sind diese Attribute Primärschlüssel der neuen Relation
- Attribute der Relationship ebenfalls in die neue Relation

## 3.5 Das E/R-Modell

- Beispiel: Many-To-Many-Relationships



Student (MatrNr, ...)

Vorlesung (VNr, ...)

hoert (MatrNr, VNr, Note)

...

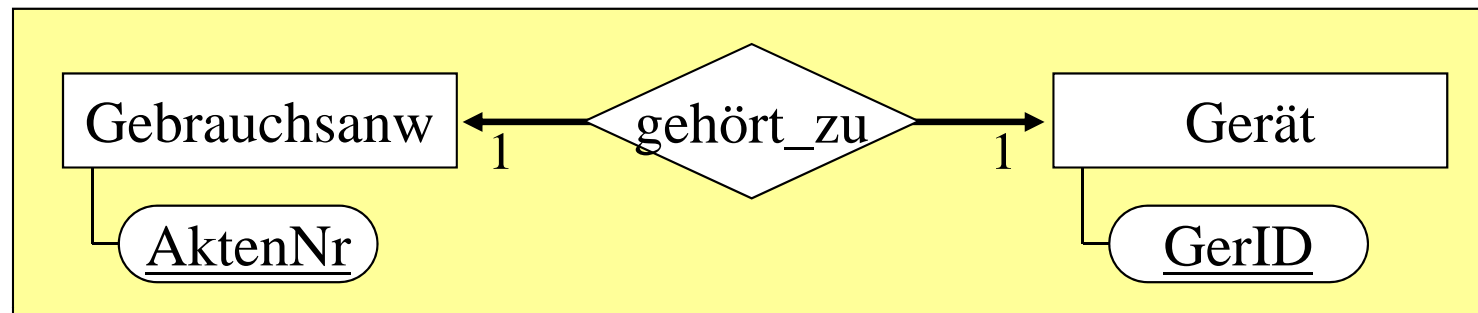
**primary key** (MatrNr, VNr),

**foreign key** MatrNr **references** Student,

**foreign key** VNr **references** Vorlesung...

## 3.5 Das E/R-Modell

- **Alternative** Lösung: One-To-One-Relationships:



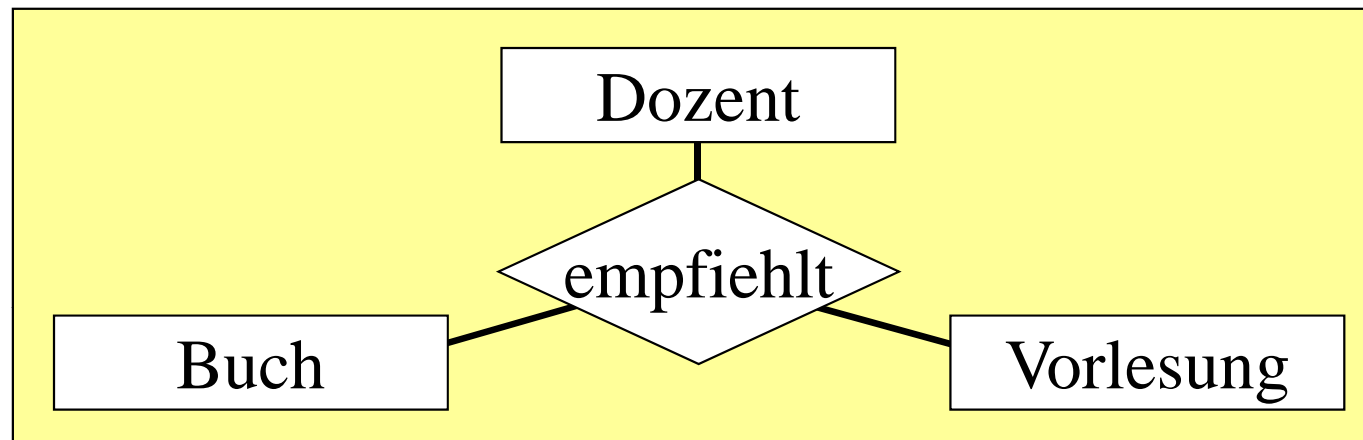
- Die beiden Entities werden zu einer Relation zusammengefasst
- Einer der Primärschlüssel der Entities wird Primärschlüssel der Relation

Geraet (GerID, ..., AktenNr, ...)

- Häufig auch Umsetzung wie bei 1:n-Beziehung (insbes. wenn eine Seite optional ist), wobei die Rollen der beteiligten Relationen austauschbar sind

## 3.5 Das E/R-Modell

### – Mehrstellige Relationen



- Eigene Relation für *empfiehlt*, falls mehr als eine Funktionalität **many** ist:

Dozent (PNr, ...)

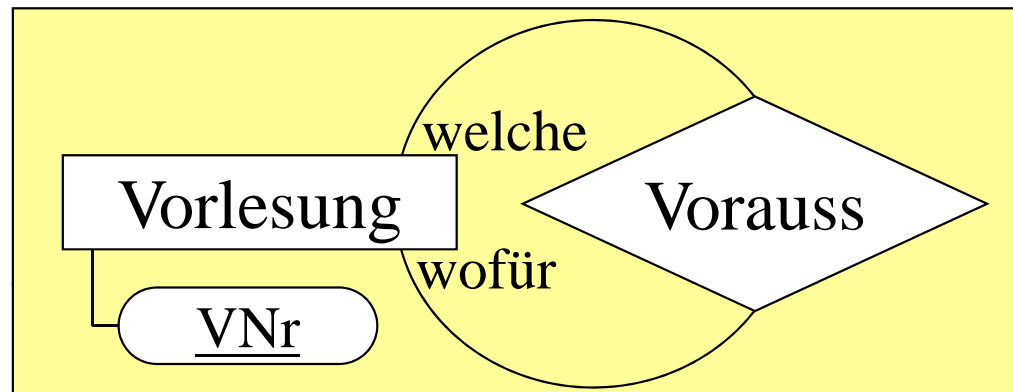
Buch (ISBN, ...)

Vorlesung (VNr, ...)

empfiehlt (PNr, ISBN, VNr)

## 3.5 Das E/R-Modell

### – Selbstbezug



Keine Besonderheiten:  
 Vorgehen je nach Funktionalität.

Vorlesung (VNr, ...)

Vorauss (Welche, Wofuer)

**foreign key** Welche **references** Vorlesung (VNr),

**foreign key** Wofuer **references** Vorlesung (VNr)