

UNIT 0x04

ENTITY-RELATIONSHIP-MODELL II

# Übersicht

---

## Themen

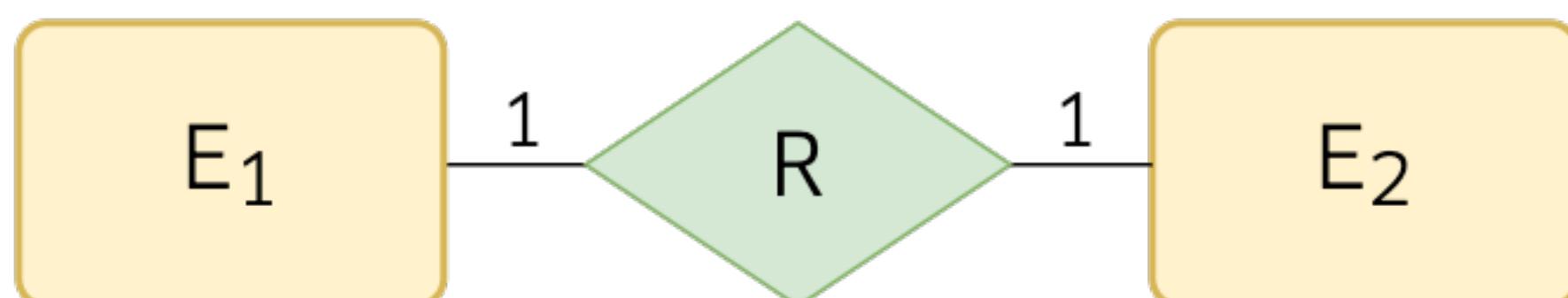
- Wiederholung der Elemente im ER-Diagramm
- Relationen
  - n-stellige Relationen
  - Existenzabhängige Entitätstypen
- Kardinalitäten
  - Min-Max-Notation
  - UML-Notation
- Generalisierung, Spezialisierung
  - Objekthierarchien

## Wiederholung

---

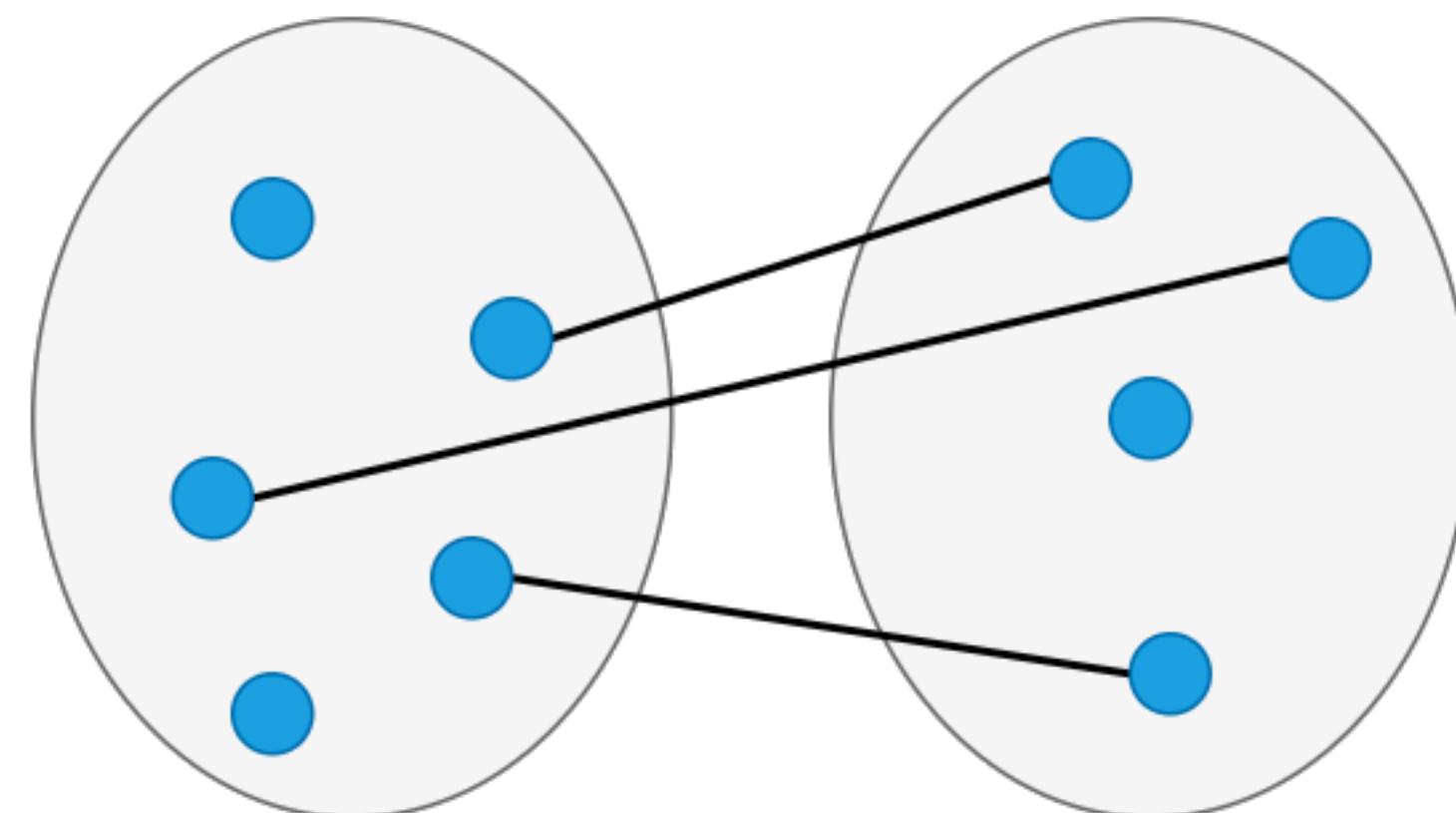
### 1:1

- Achtung: 0 oder 1, d.h. es existieren Entitäten ohne Partner.



### Beispiel

- Mini-Welt: Person hält genau ein Haustier und Haustier hat genau eine/n Besitzer/in.
- Person ( $E_1$ ) verantwortlich\_für Haustier ( $E_2$ ).
- (Max,Wuff)  $\in R$

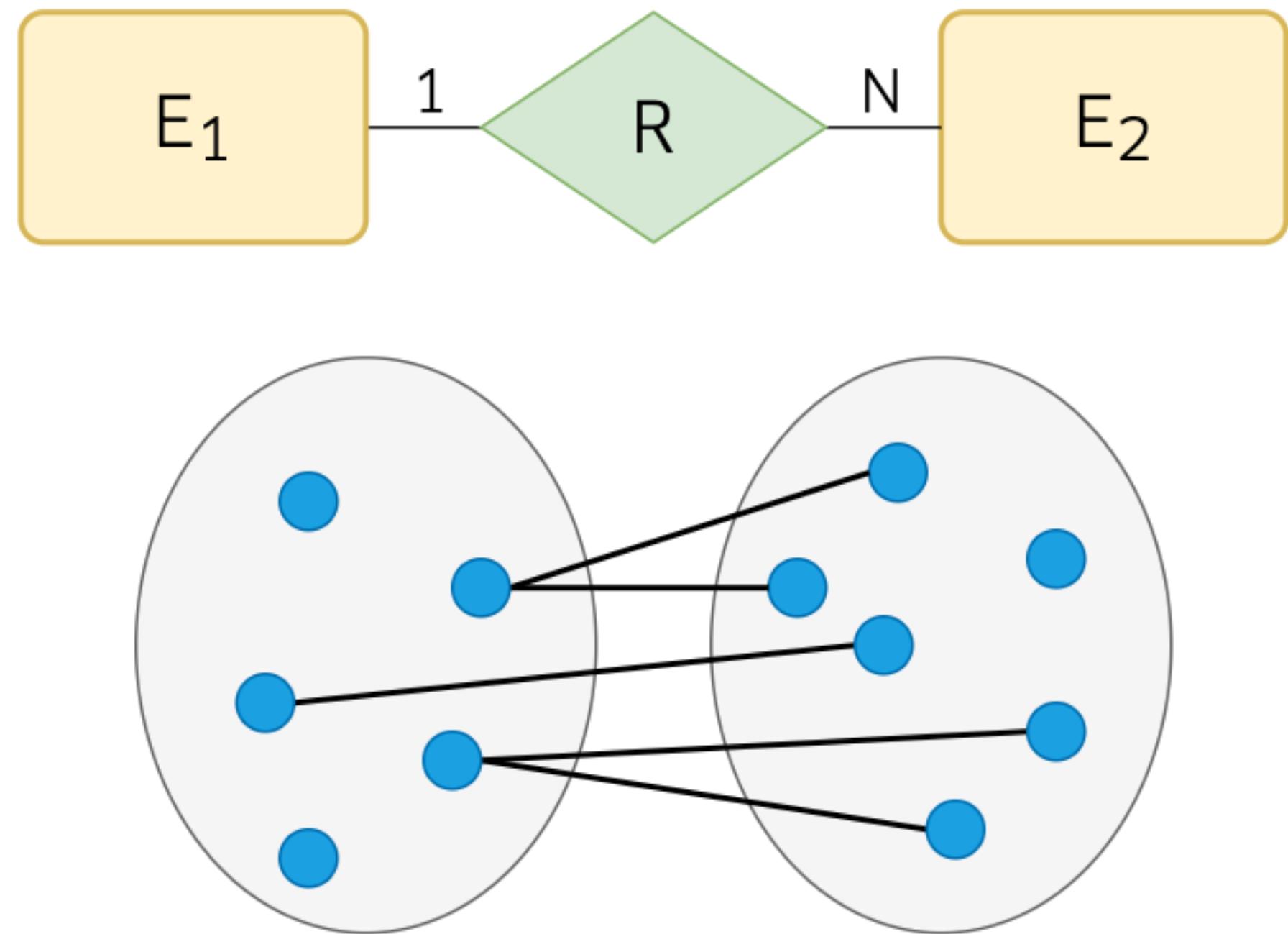


## Wiederholung

---

### 1:N und N:1

- Achtung: 0 oder 1, d.h. es existieren Entitäten ohne Partner.
- Für N:1 einfach die Rollen tauschen.
- Statt 'N' bzw. 'n' auch '\*'.



### Beispiel

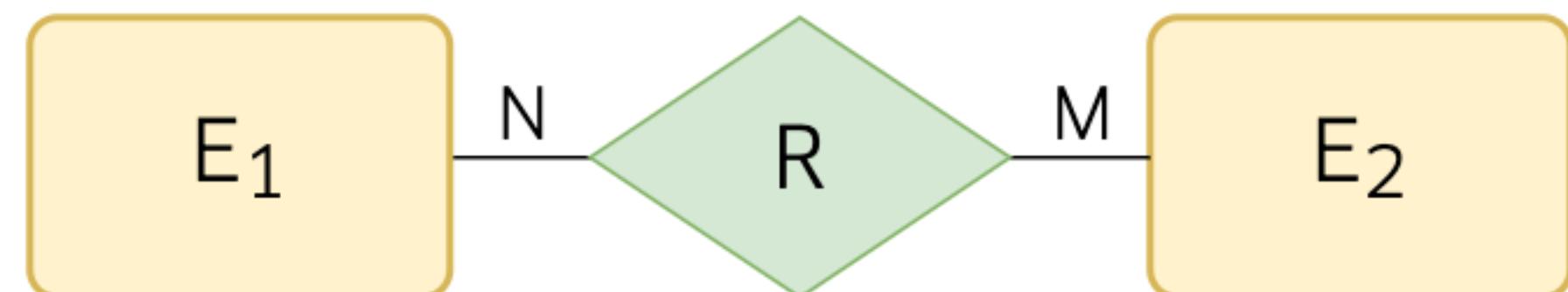
- Mini-Welt: Abteilung sind Mitarbeiter zugeordnet und ein/e Mitarbeiter/in arbeitet in genau einer Abteilung.
- Abteilung (E<sub>1</sub>) beschäftigt Mitarbeiter (E<sub>2</sub>).
- (Marketing,Max) ∈ R

## Wiederholung

---

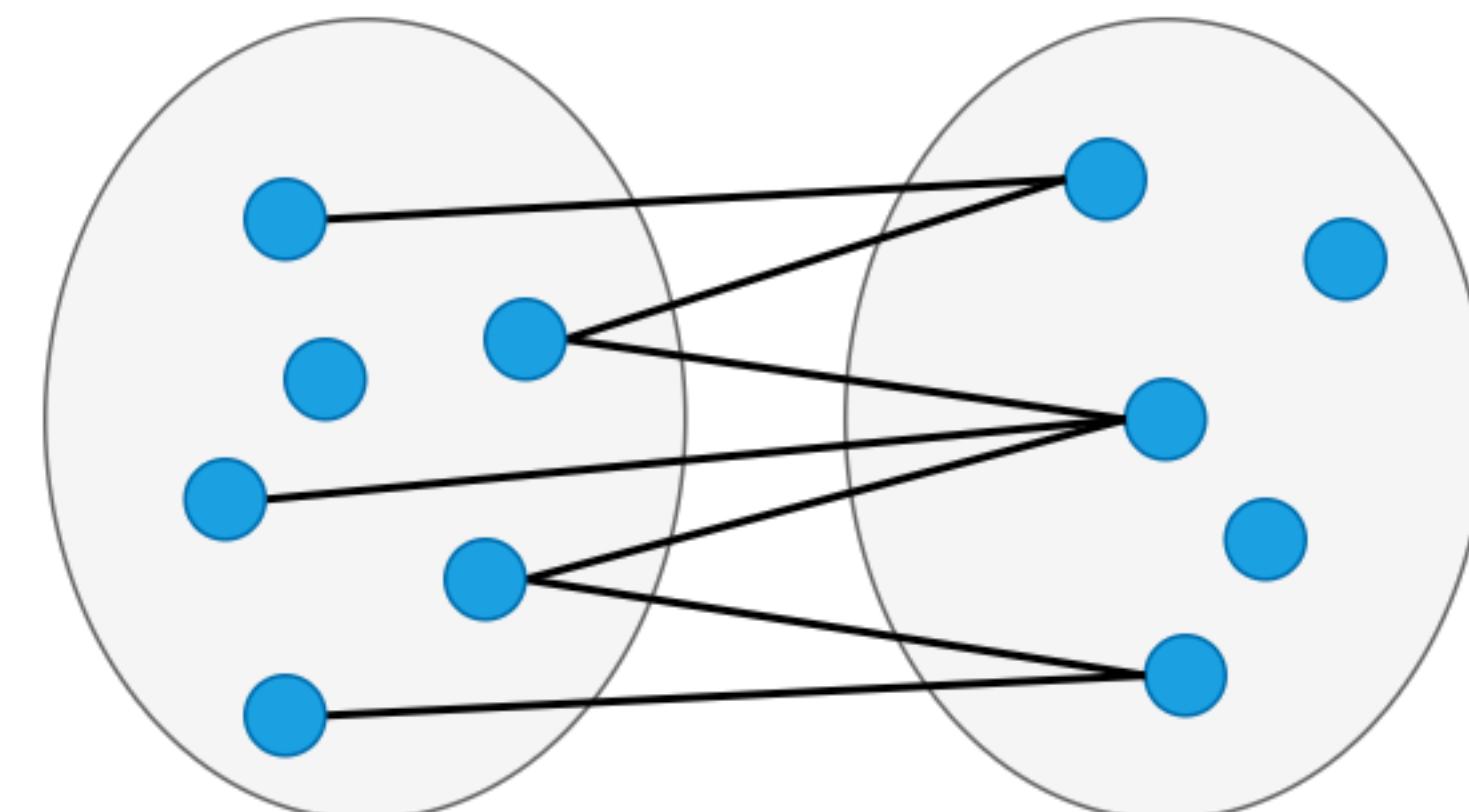
### N:M

- Achtung: 0 oder 1, d.h. es existieren Entitäten ohne Partner.
- Statt 'N' bzw. 'n' auch '\*'.



### Beispiel

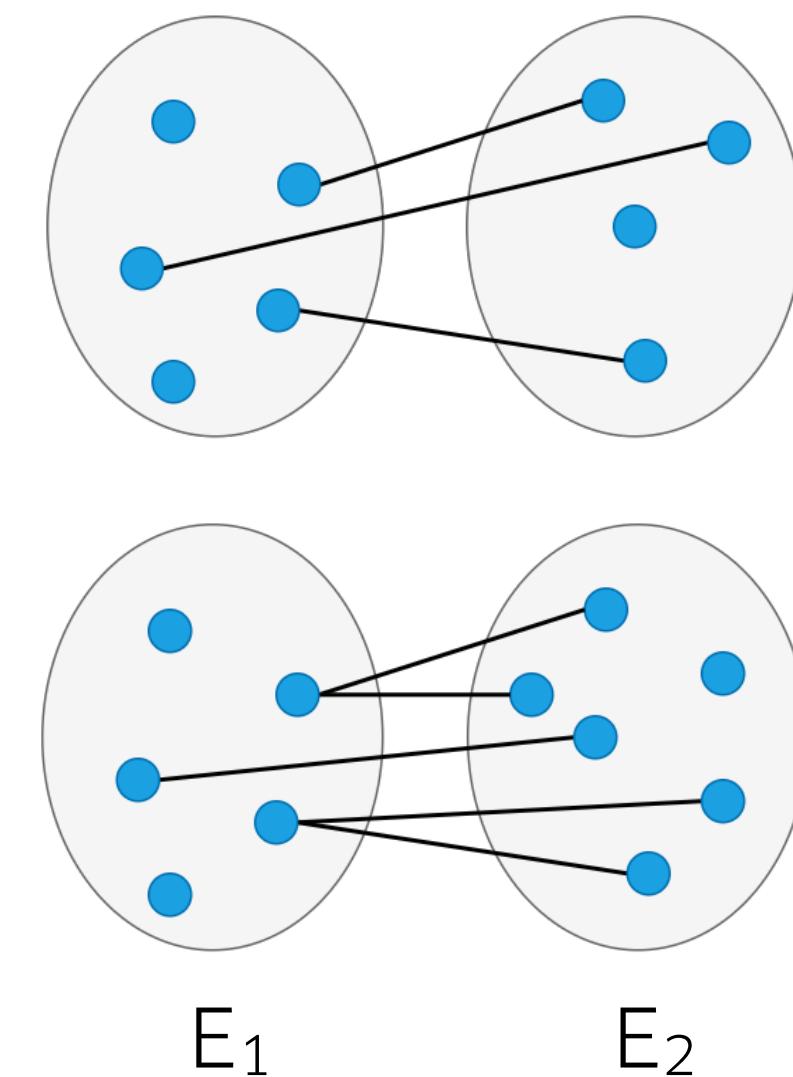
- Mini-Welt: Vorlesungen werden von Studierenden gehört.
- Studierende (E<sub>1</sub>) besuchen Vorlesungen (E<sub>2</sub>).
- (Max,Analysis 1) ∈ R



## Relationen

### Partielle und totale Funktion

- Analog zur Funktion  $g: \mathbb{R} - \{0\} \rightarrow \mathbb{R}, x \mapsto 1/x$  mit Definitionslücke  $\{0\}$  ist man an einem Funktionsbegriff interessiert, der Ausnahmen zulässt. Man nennt  $f: X - L \rightarrow Y$  und  $L \subseteq X$ 
  - **eine partielle Funktion**,  $f: X \rightarrow Y$ , falls  $L \neq \emptyset$ , oder klassisch
  - **eine totale Funktion**,  $f: X \rightarrow Y$ , falls  $L = \emptyset$ .
- So definiert eine 1:1-Relation implizit zwei partielle Funktionen  $f_1: E_1 \rightarrow E_2$  und  $f_2: E_2 \rightarrow E_1$ .
- Ebenso impliziert eine 1:N-Relation zumindest noch eine, nicht notwendigerweise injektive, partielle Funktion  $f_2: E_2 \rightarrow E_1$ . Die Abbildung von  $E_1$  nach  $E_2$  ist keine klassische Funktion mehr, da wir keine eindeutige Zuordnung haben.
- Letzteres gilt analog für N:M-Relation in beiden Richtungen.

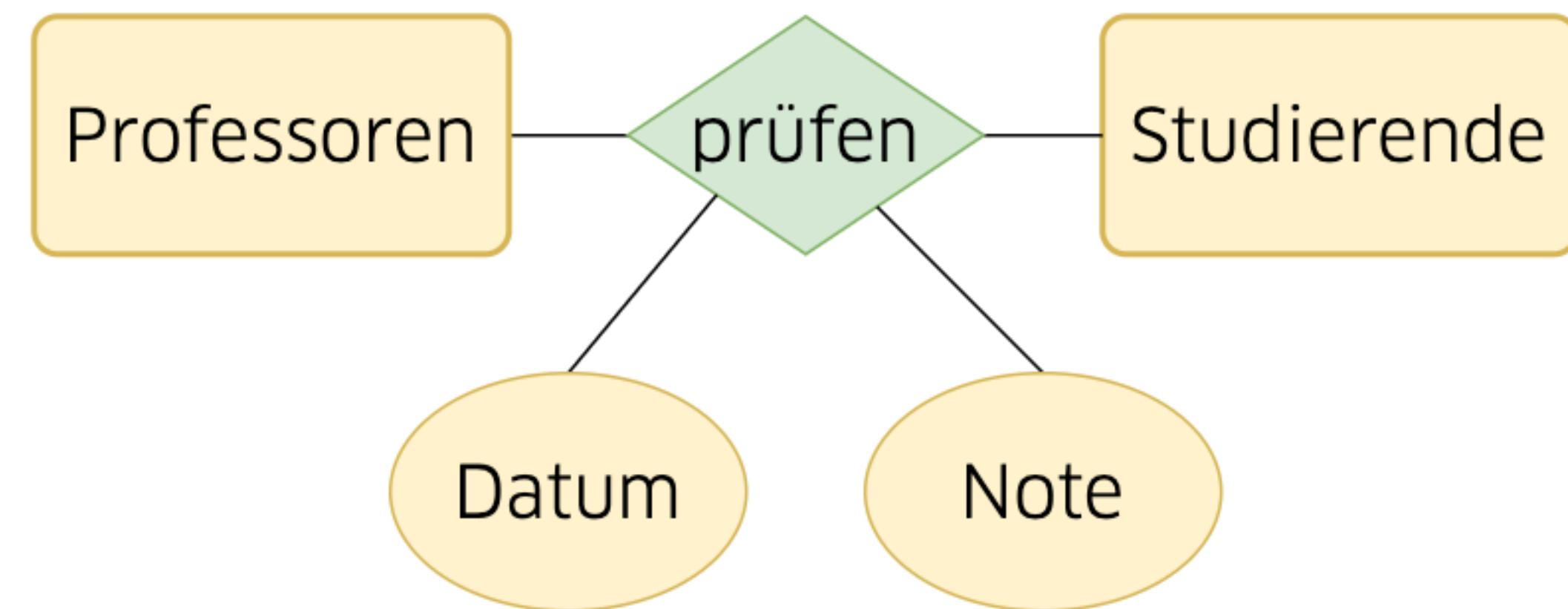


## Relationen

### Relationen mit Attributen

- Sei  $A=[A_1:D_1, \dots A_m:D_m]$  ein Tupel mit  $m$  Attributen  $A_1, \dots, A_m$  und  $E_1, \dots, E_n$  eine Menge von Entitätstypen. Dann ist  $R \subseteq E_1 \times \dots \times E_n \times D_1 \times \dots \times D_m$  eine mit  $A$  attributierte Relation.

### Beispiel $n=2, m=2$



- $\text{prüfen } (R) \subseteq \text{Professoren } (E_1) \times \text{Studierende } (E_2) \times \text{Datum } (D_1) \times \text{Note } (D_2)$

### Q&A

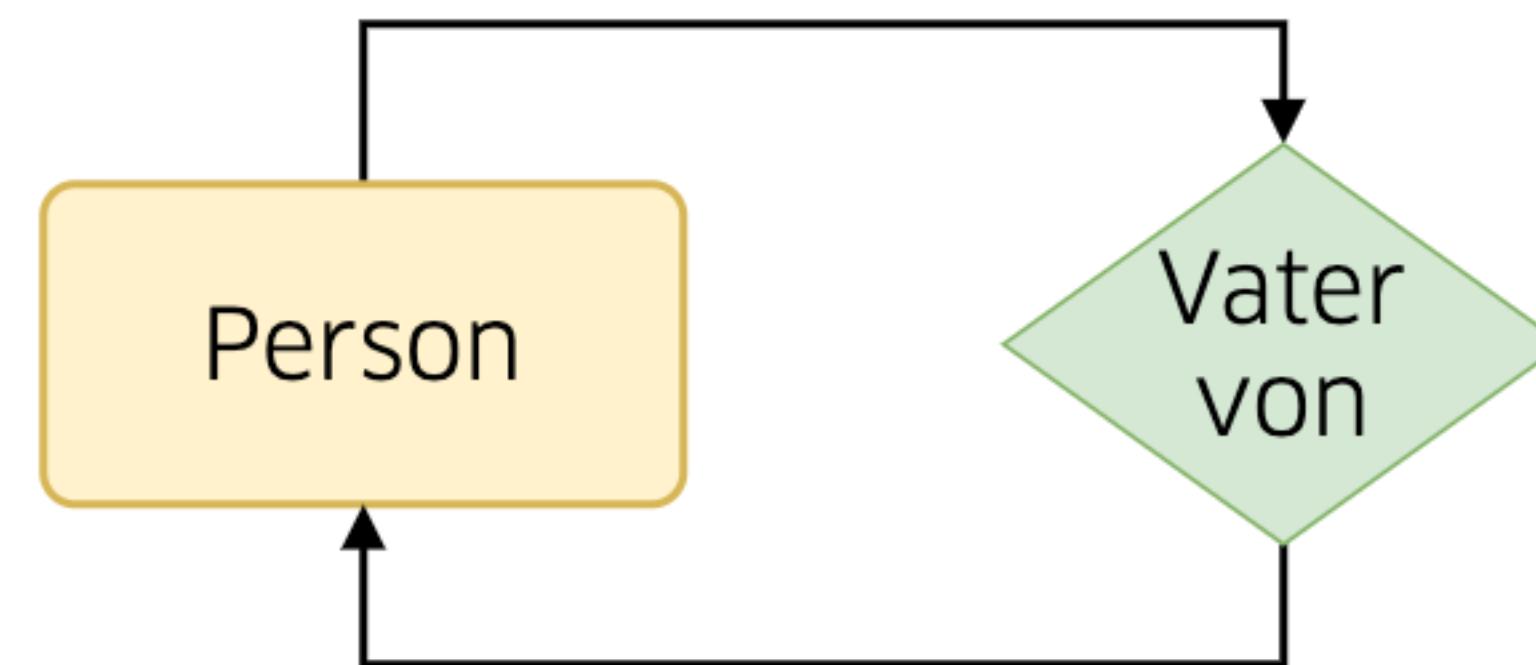
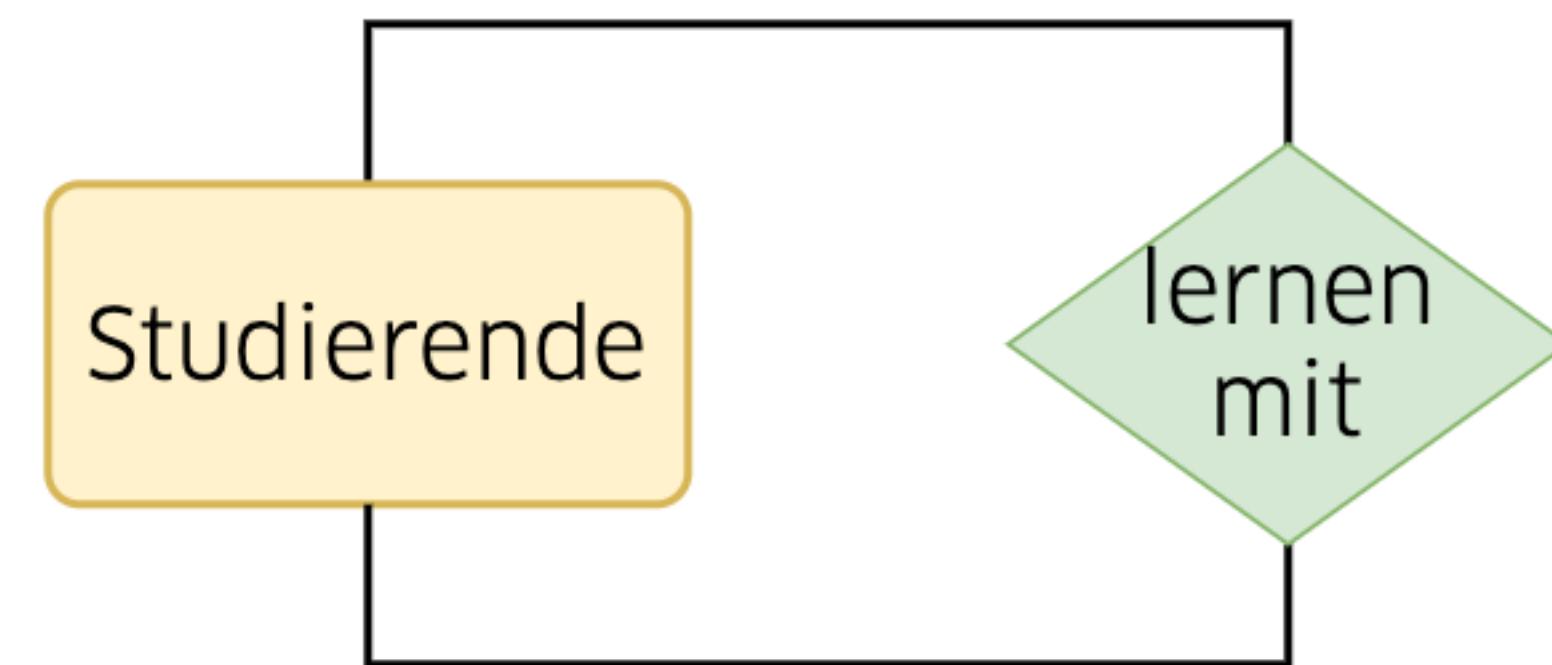
- Gehören Attribute an die Entitätstypen oder an die Relation?

## Relationen

---

### 1:1-, 1:N-, N:1-, N:M-Relationen zu sich selbst

- Auch wenn wir bislang allgemein von Mengen  $E_k$  in den Definitionen und Beispielen gesprochen haben, können diese natürlich identisch sein. Das bedeutet, dass eine Entitätsmenge zu sich selber in Relation stehen kann.
- Wenn es notwendig ist, können Pfeile die Beziehung verdeutlichen.

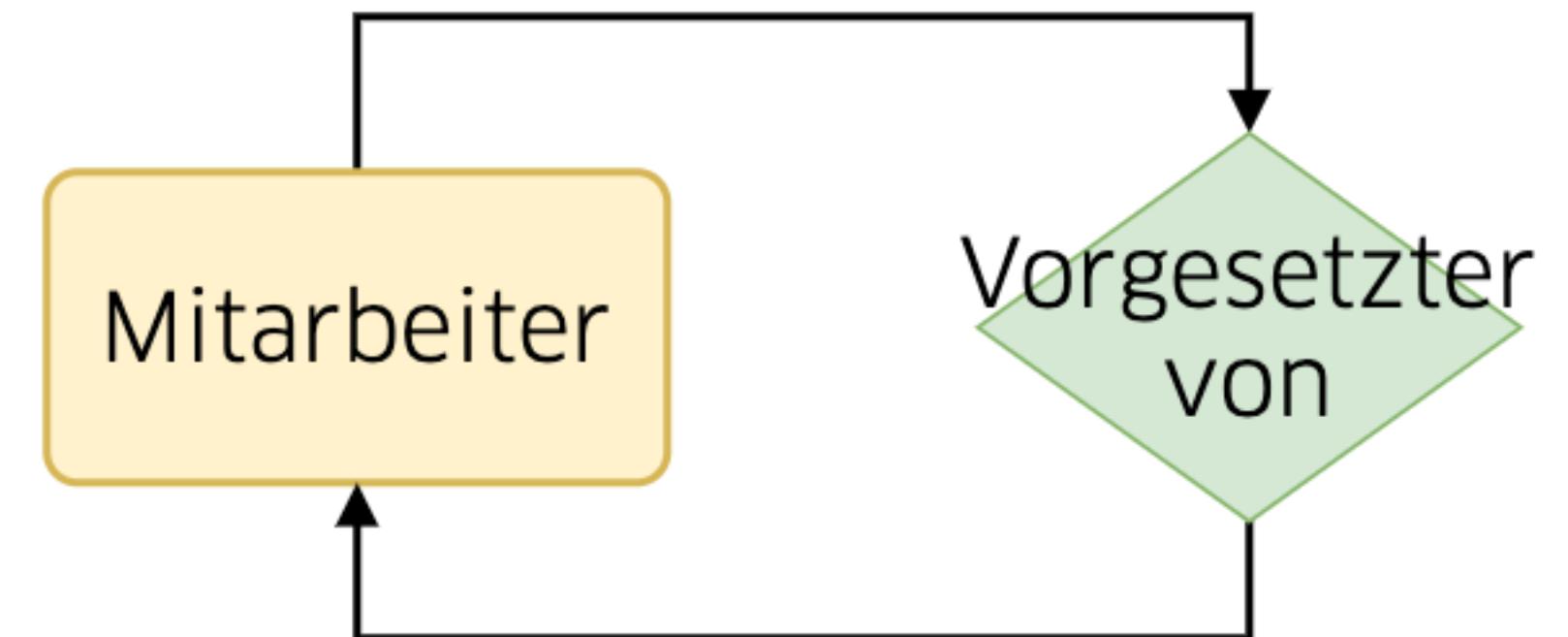


## Relationen

---

### 1:1-, 1:N-, N:1-, N:M-Relationen zu sich selbst

- Beispiel matse\_mhist
  - Relation vorgesetzter\_von zwischen mitarbeiter und mitarbeiter
  - aber in der Tabelle umgekehrt abgebildet über vorgesetzter\_mitarbeiter\_id



	id	name	jahresgehalt	vorgesetzter_mitarbeiter_id
1	1	Mia	110000.00	26
2	2	Ben	90000.00	26
3	3	Emma	70000.00	12
4	4	Paul	5000.00	27
5	5	Hannah	5000.00	27
6	6	Luka	5000.00	27
7	7	Sofia	50000.00	1
8	8	Jonas	50000.00	1
9	9	Anna	50000.00	2
10	10	Finn	50000.00	2

Beispiel: Mia (id 1) ist Vorgesetzte von Jonas (id 7)

**Q&A**

- Warum so?

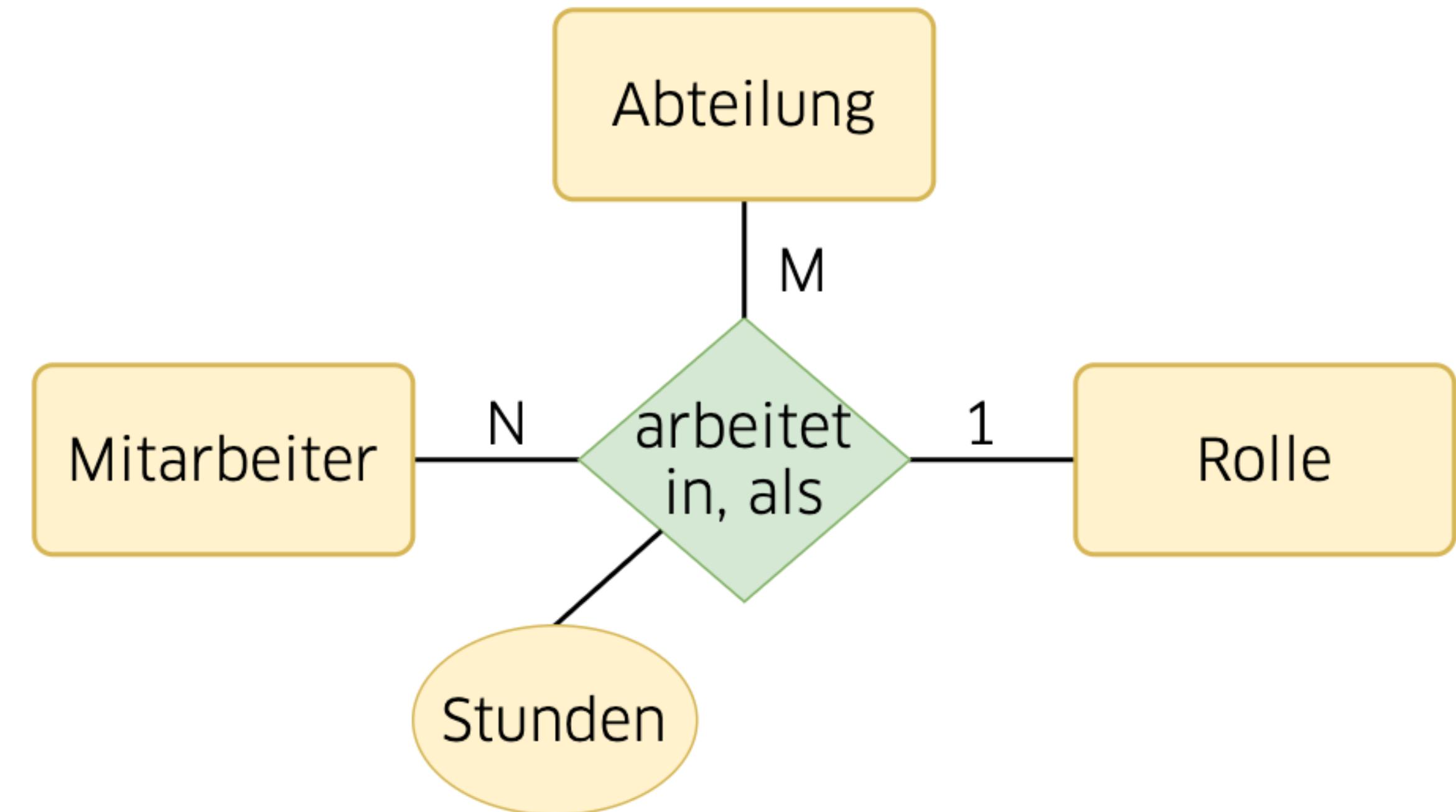
## Relationen

### n-stellige Relationen

- Es ist bekanntlich möglich, auch mehr als zwei Entitätstypen in Beziehung zu setzen, denn  $R \subseteq E_1 \times \dots \times E_n$ .

#### Q&A

- Wie kommt man hier auf die Chen-Kardinalitäten?
- Wie sehen die Tabellen aus?
- Ist die Situation äquivalent zu drei Relationen?

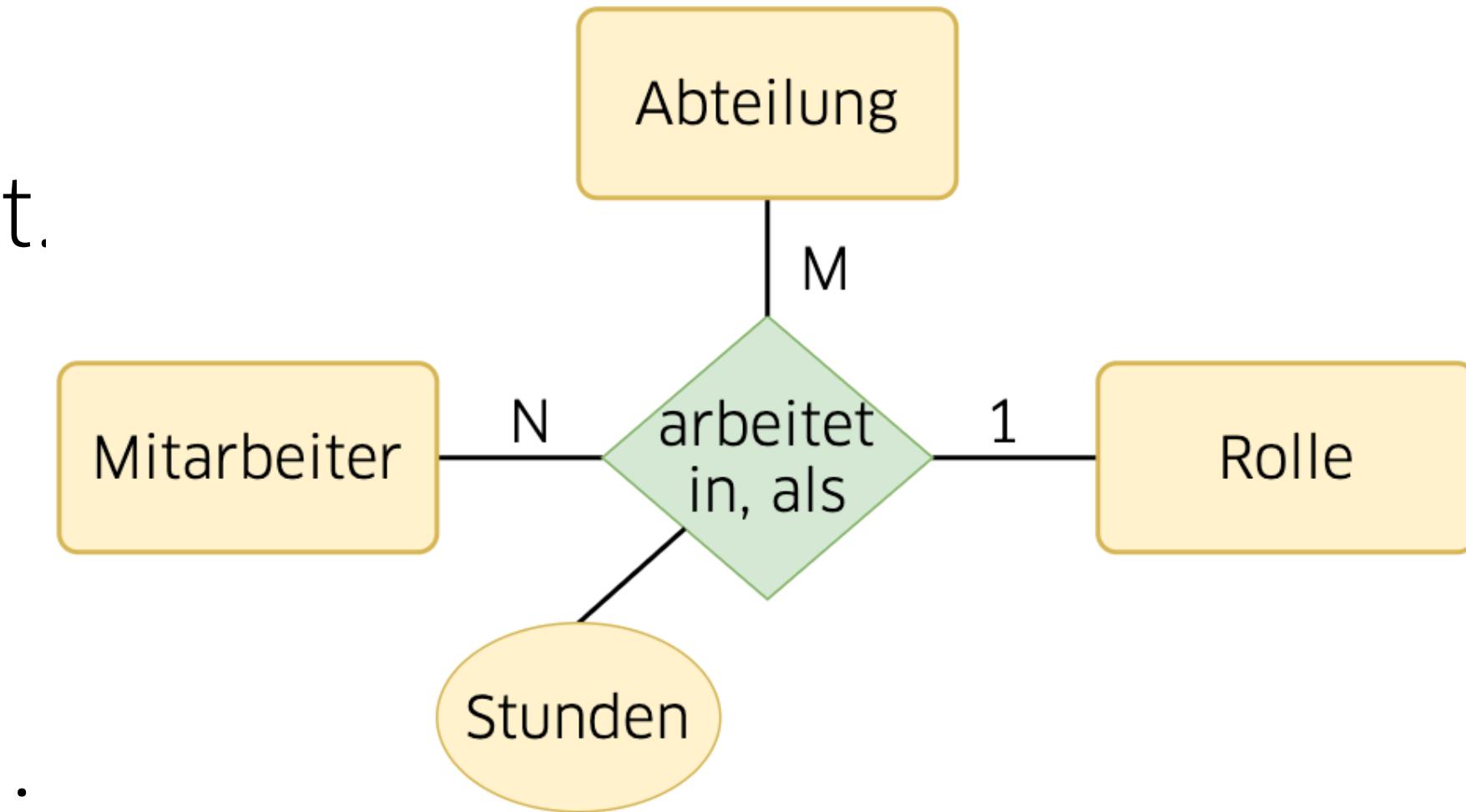


Beispiel: arbeitet\_in\_also aus matse\_mhist

## Relationen

### n-stellige Relationen - Chen-Notationen

- Zunächst ein Blick auf die Anforderungen/Mini-Welt.  
Hier könnte stehen, dass ein Mitarbeiter in einer Abteilung immer nur in einer Rolle arbeitet.
- Wie zuvor (siehe Chen-Notation) halten wir n-1 von n Entitäten fest (beliebig, aber fest) und notieren die Kardinalität 'gegenüber', d.h. hier bspw.:
  - Max (Mitarbeiter) arbeitet in der Forschung (Abteilung) nur als Berater (Rolle), d.h. Mitarbeiter fest, Abteilung fest (s.o.) → 1 Rolle.
  - In der Forschung (Abteilung) arbeiten als Berater (Rolle) Max und Mia, d.h. Abteilung fest, Rolle fest → N Mitarbeiter.
  - Mia (Mitarbeiter) kann als Berater (Rolle) in Forschung und Marketing helfen, d.h. Mitarbeiter fest, Rolle fest → M Abteilungen.

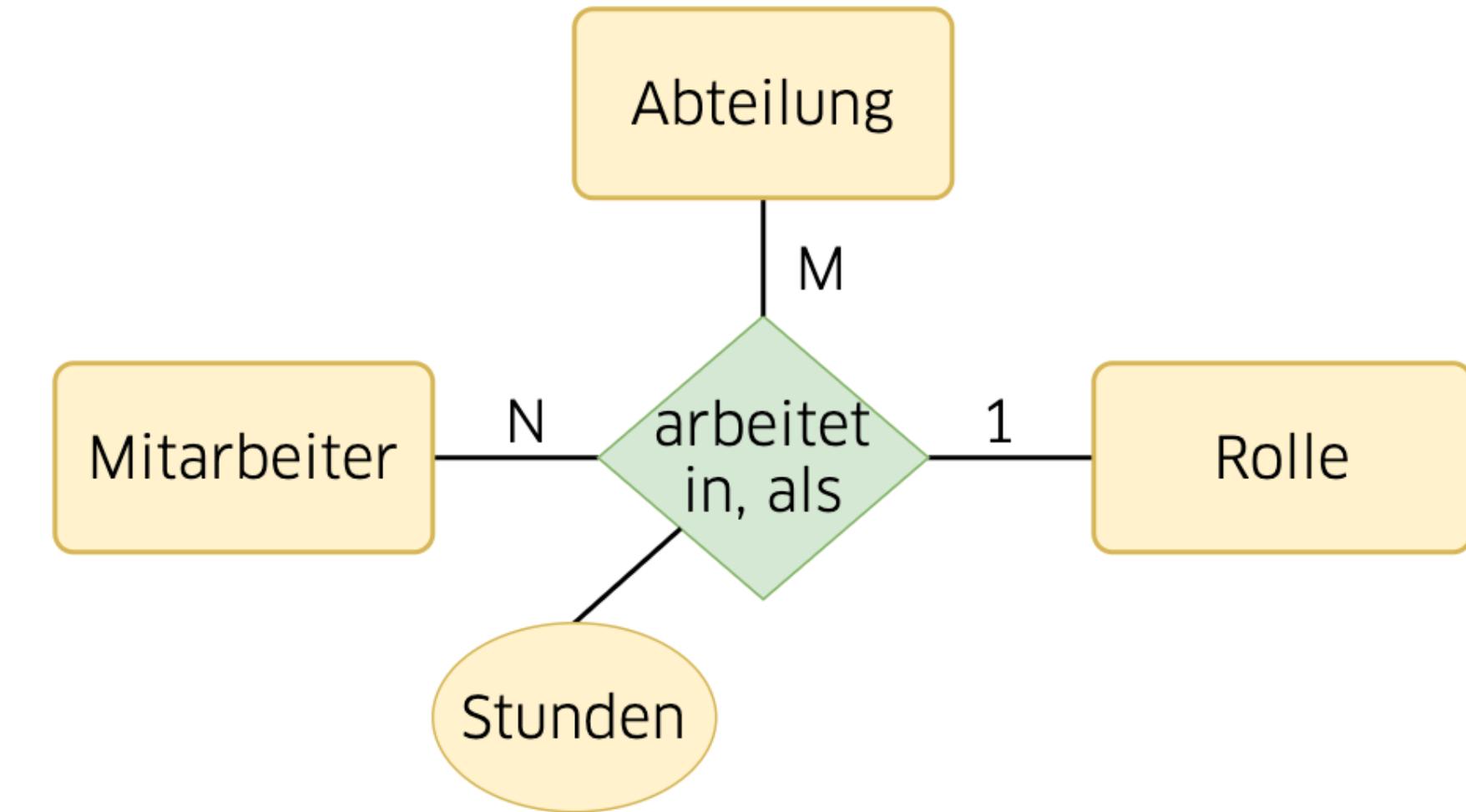


# Relationen

---

## n-stellige Relationen - Tabellen

- Ausser, dass in der Relation arbeitet\_in\_als hier mehrere Fremdschlüssel vorkommen, ändert sich nichts.
- Die Stunden (wochenstunden) sind ein Attribut der Relation – wie zuvor besprochen.



	<b>id</b>	<b>mitarbeiter_id</b>	<b>abteilung_id</b>	<b>rolle_id</b>	<b>wochenstunden</b>
1	1		1	3	10.00
2	2		1	4	5.00
3	3		1	3	10.00
4	4		2	4	15.00
5					

Beispiel: arbeitet\_in\_als aus matse\_mhist

	<b>id</b>	<b>name</b>
1	1	Mia
2	2	Ben
3	3	Emma
4	4	Paul
5		Hannah

mitarbeiter

	<b>id</b>	<b>name</b>
1	1	Vorstand
2	2	HR/Buchhalt
3	3	Vertrieb
4	4	Marketing
5		Einkauf

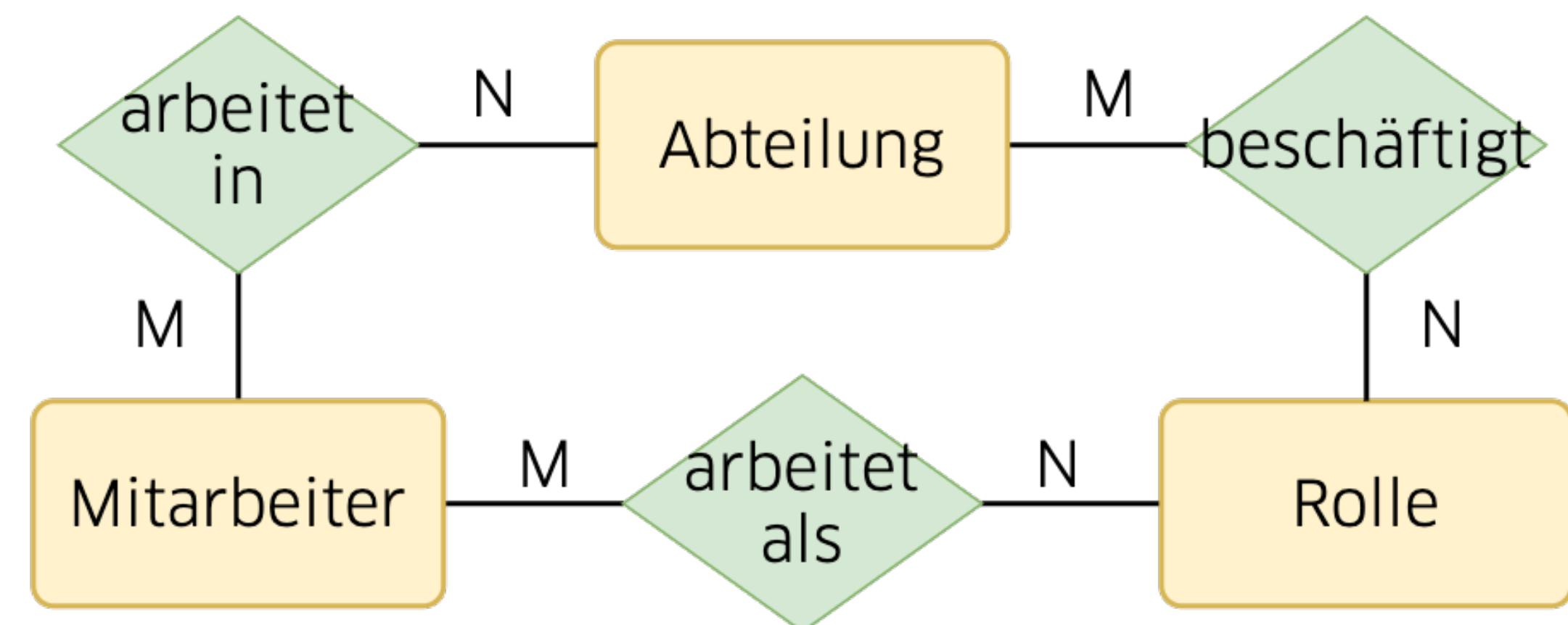
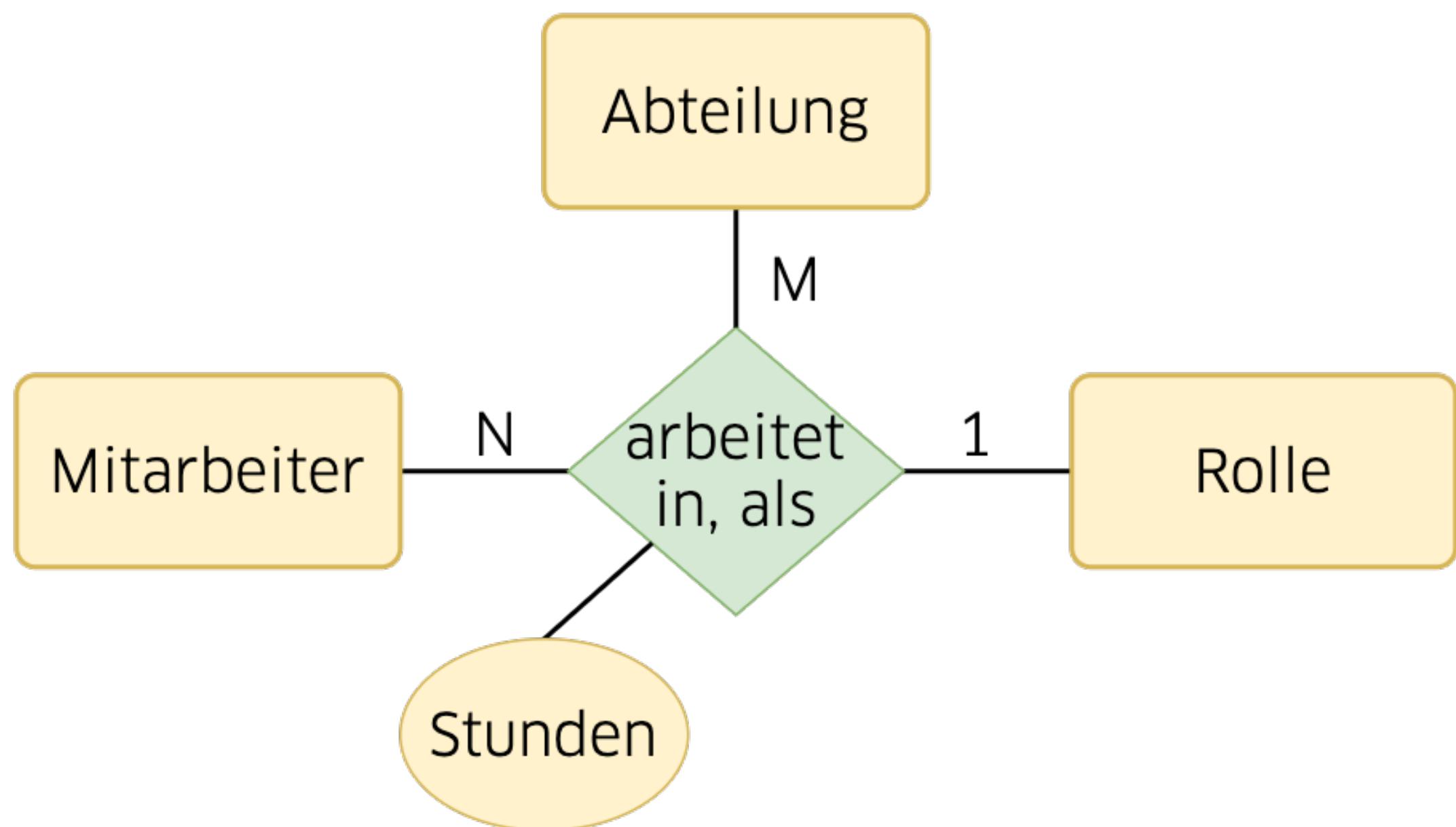
abteilung

	<b>id</b>	<b>beschreib</b>
1	3	Leitung
2	4	Mitarbeiter
3	7	Student
4	8	Auszubilden
5		Schüler

rolle

## Relationen

### n-stellige Relationen - keine Äquivalenzen



- In der rechten Version gehen Informationen verloren. Mitarbeiter arbeiten in Abteilungen, aber es ist unklar 'als was'.
- Die Anforderung aus der Mini-Welt (ein Mitarbeiter in einer Abteilung nur in einer Rolle) ist so nicht abzubilden.

#### Q&A

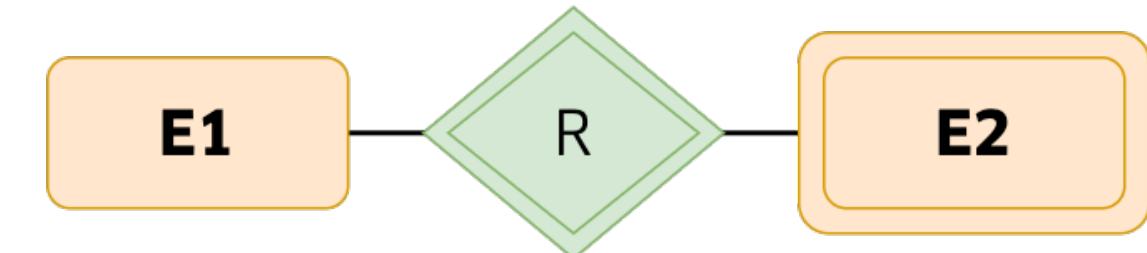
- Wo passen die Stunden hin?

## Relationen

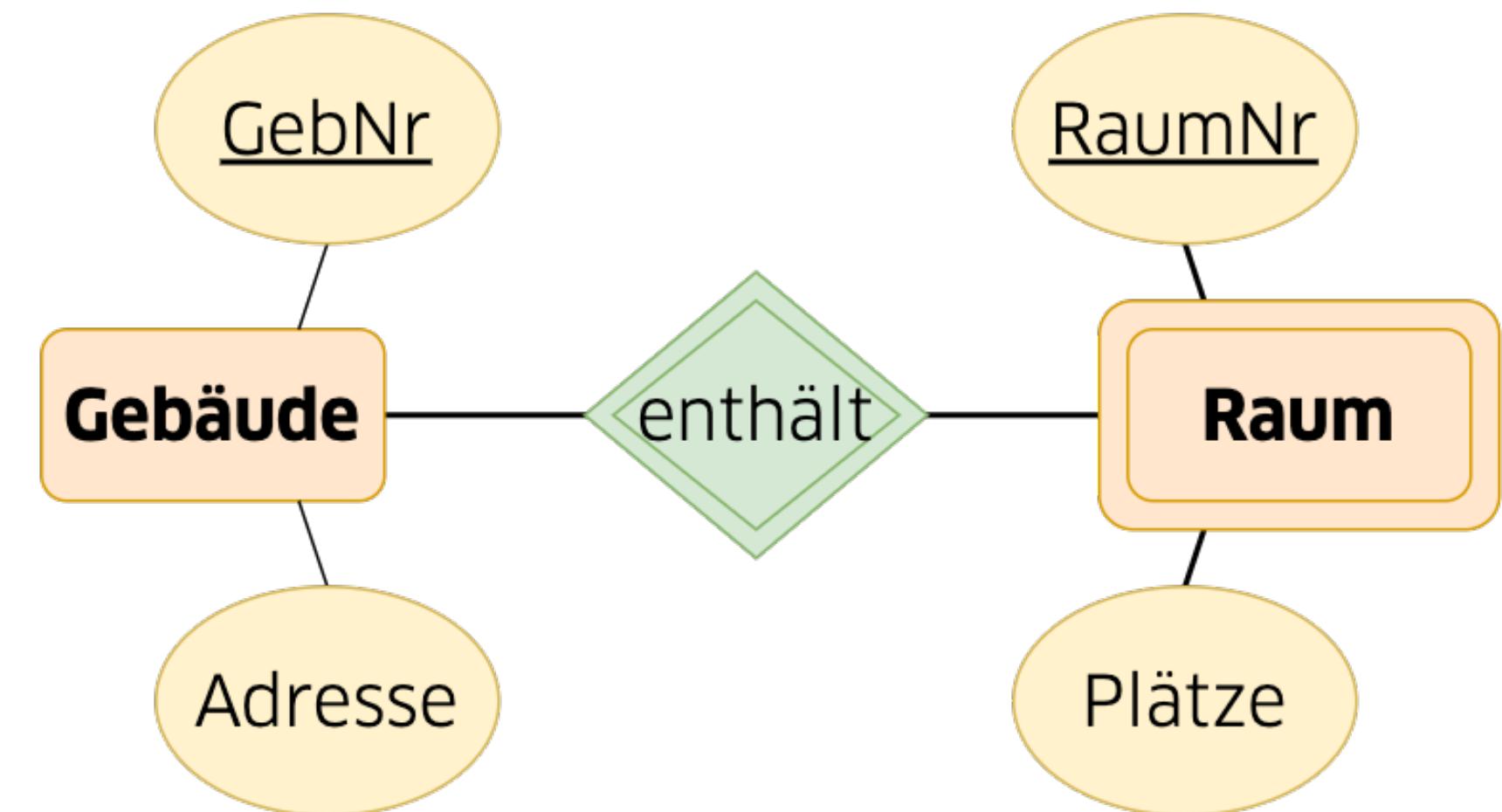
---

### Existenzabhängige Entitätstypen

- *Grundidee:* Entitäten existieren nur in Verbindung mit anderen Entitäten (wie UML Komposition).
- *Beispiel:* Ein Raum in einem Gebäude.
- Kein eigener Schlüsselkandidat (RaumNr alleine nicht geeignet), sondern Identifikation über Beziehung zur übergeordneten Entität, z.B. über Kombination (GebNr,RaumNr).
- Max. 1:1- oder 1:N-Relationen zwischen abhängigem (schwachem) Entitätstyp (Raum) und Vater-Entitätstyp (Gebäude), wobei die schwache Entität in Relation stehen muss!



Symbol eines existenzabhängigen Entitätstypen im ER-Diagramm



## Relationen und Kardinalität

---

### Chen-Notation

- Contra: Angabe ungenau, nur '1' (0 oder 1) und 'viele' (N,\*) möglich.

### Min-Max-Notation

- *Idee:* In der Min-Max-Notation wird angegeben, wieviele Relationen minimal bzw. maximal möglich sein sollen.
- Achtung: Hier werden die Elemente der Relation gezählt!

## Relationen und Kardinalität

---

### Mini-Welt Fotoshooting V

**Hintergrund:** Auf einer Familienfeier [...]

Auf den Fotos am Geburtstag sollen der Geselligkeit wegen auf jedem Foto mind. zwei aber max. vier Personen zu sehen sein. Umgekehrt soll eine Person mind. einmal aber auf nicht mehr als drei Fotos zu sehen sein.



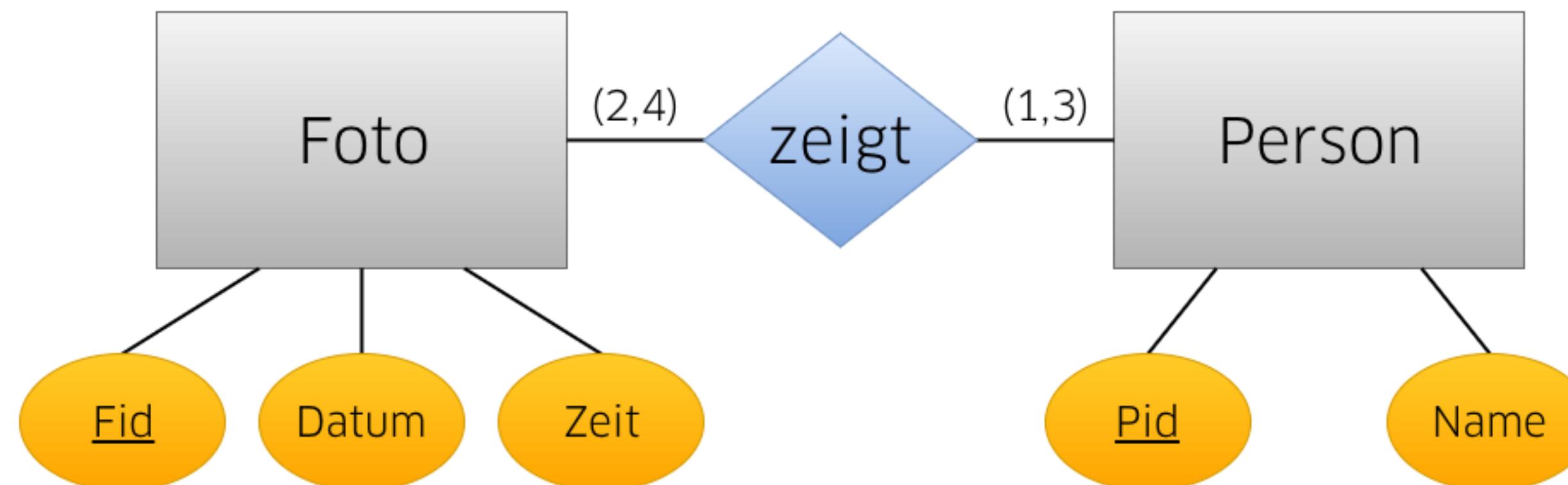
Ist so in Chen nicht abzubilden.

## Relationen und Kardinalität

---

### ER-Diagramm mit Min-Max-Kardinalität

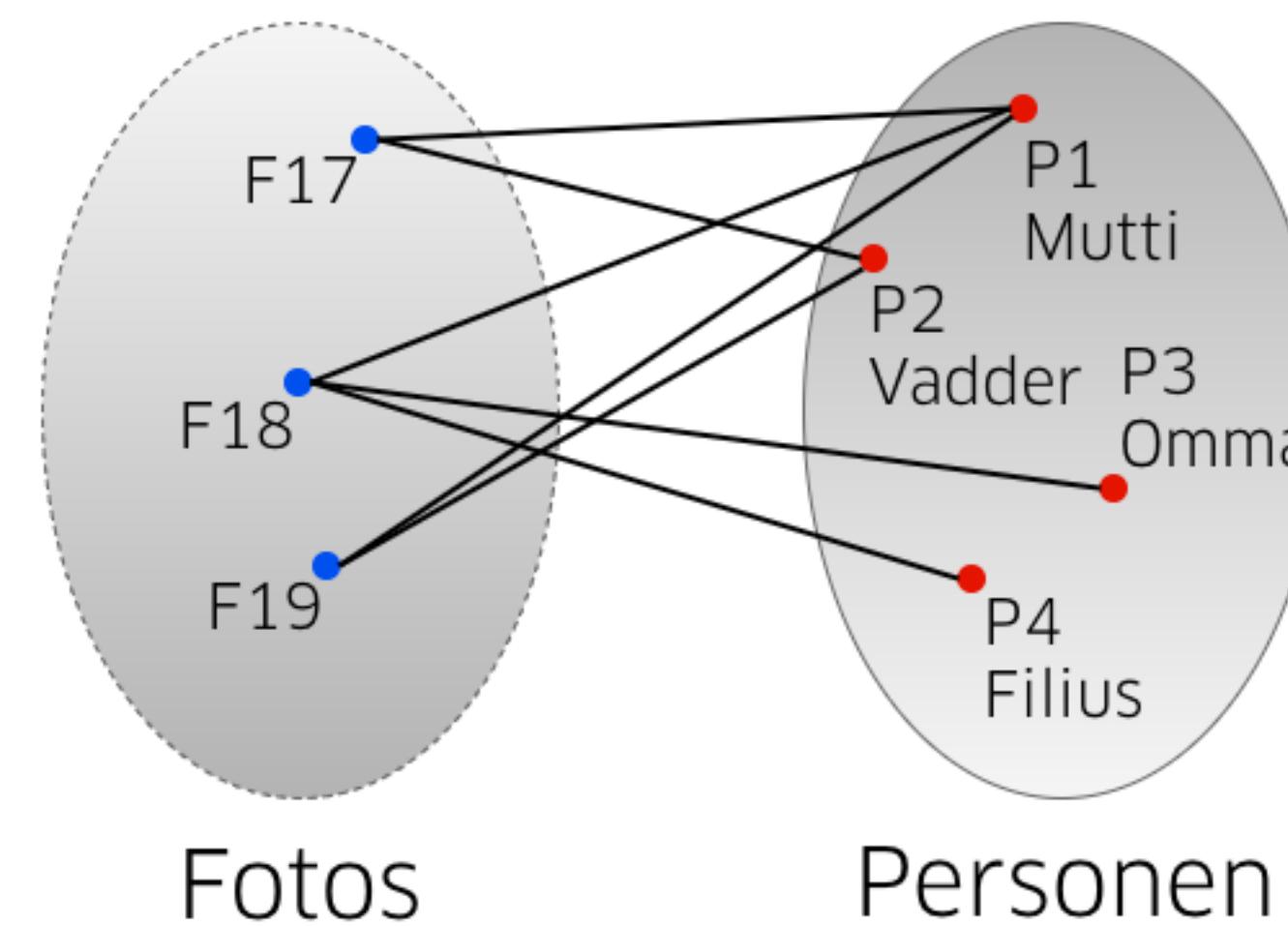
- Auf jedem Foto sind mind. zwei aber max. vier Personen zu sehen.
- ← Eine Person ist mind. einmal aber auf nicht mehr als drei Fotos.



## Relationen und Kardinalität

### ER-Diagramm mit Min-Max-Kardinalität

- Hier werden die Elemente der Relation gezählt!



Ein Foto zeigt mind. 2, max. 4 Personen und eine Person ist min. einmal, max. 3 mal auf Fotos.

Fid	Datum	Zeit
F17	29.02.16	21:01
F18	29.02.16	21:02
F19	29.02.16	21:03

Pid	Name
P1	Mutti
P18	P1
P18	P3
F18	P4
F19	P1
F19	P2



$$2 \leq |F_i| \leq 4, 1 \leq |P_j| \leq 3$$

## Relationen und Kardinalität

---

### Min-Max-Notation

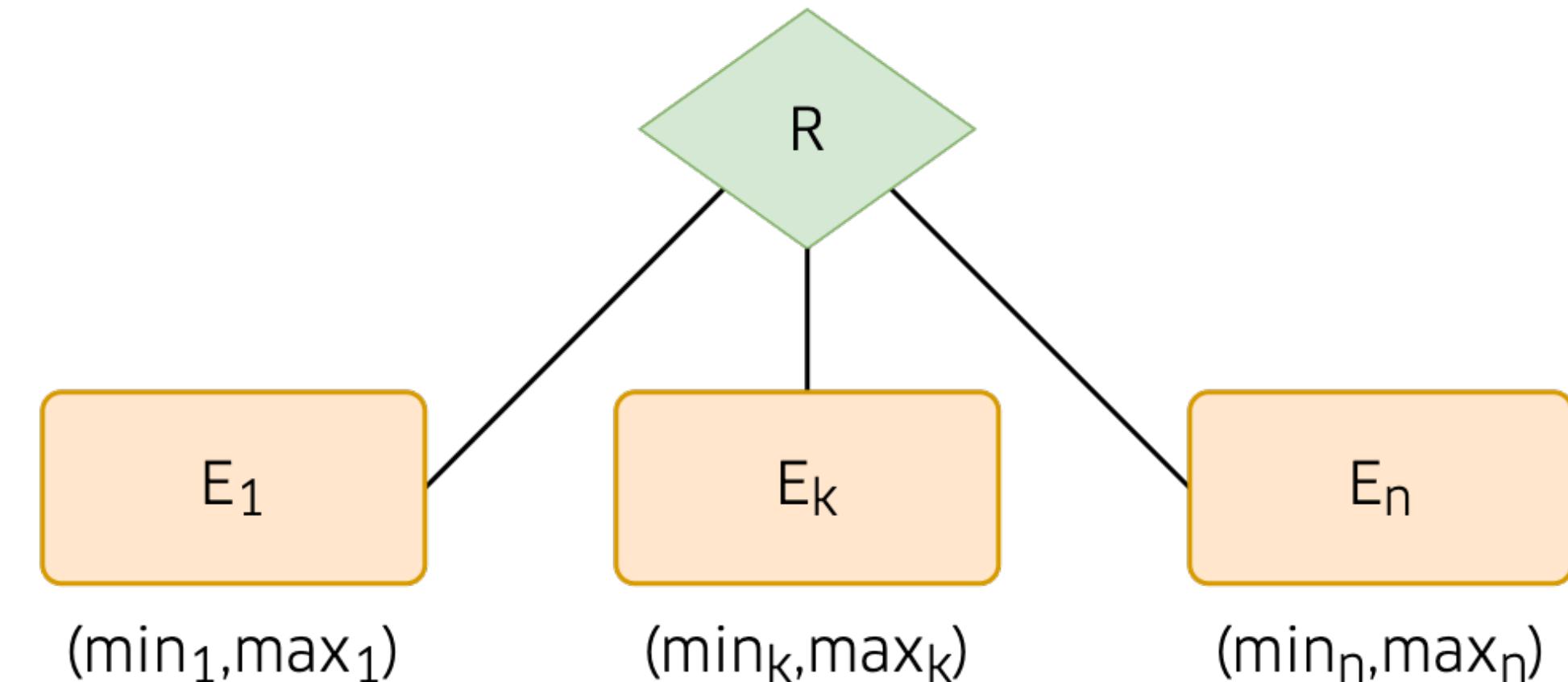
- Für jeden an einer Relation beteiligten Entitätstyp  $E_k$  wird angegeben, wie oft eine Entität minimal und maximal in der Relation enthalten sein darf (vgl. Beispiel).
- Sei  $R \subseteq E_1 \times \dots \times E_n$  und die Projektion auf das  $k$ -te Element bzw. die Anzahl eines Wertes an  $k$ -ter Stelle definiert durch

$$\text{proj}_k(e_1, \dots, e_k, \dots, e_n) = e_k,$$

$$\text{count}_k(e) = |\{r \in R \text{ mit } \text{proj}_k(r) = e\}|,$$

dann muss für alle  $k$  gelten:

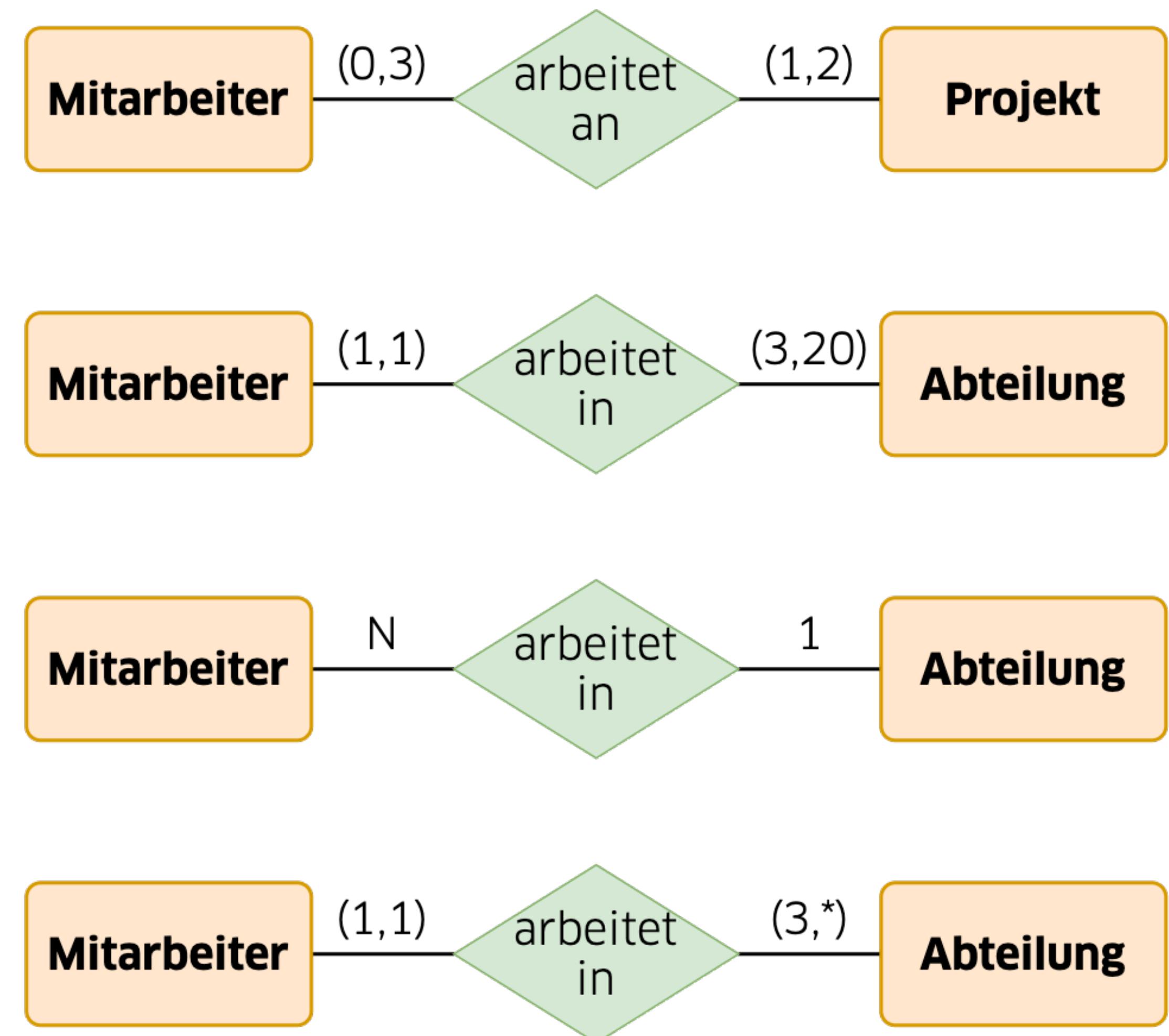
$$\forall e_k \in E_k : \min_k \leq \text{count}_k(e_k) \leq \max_k.$$



# Relationen und Kardinalität

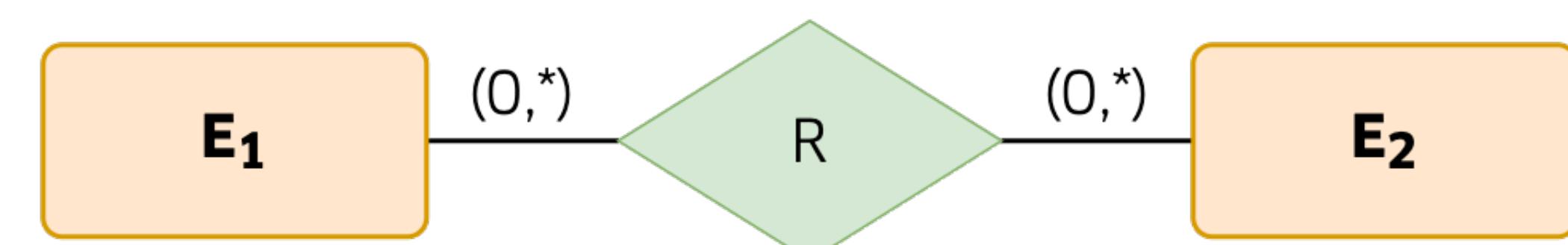
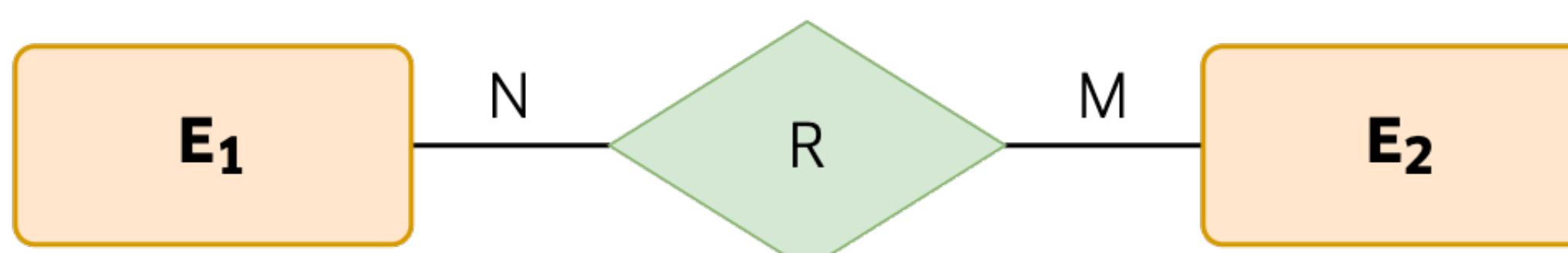
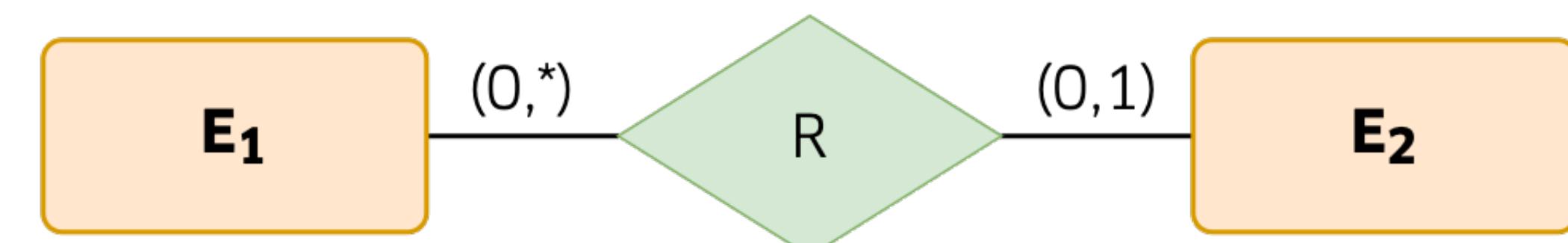
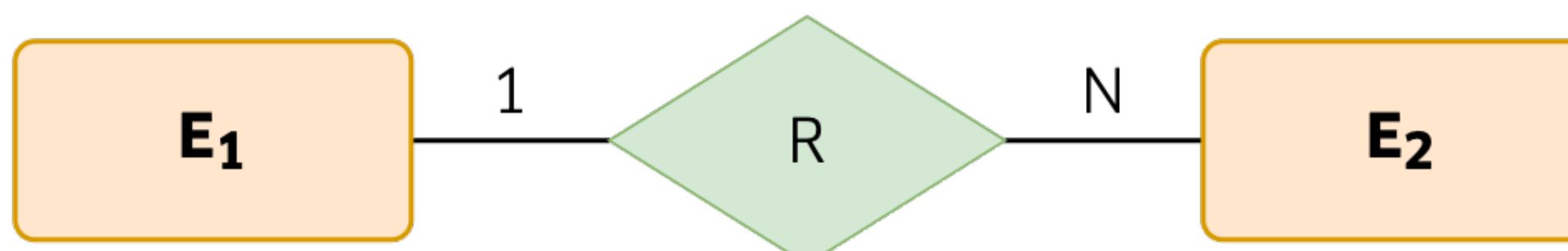
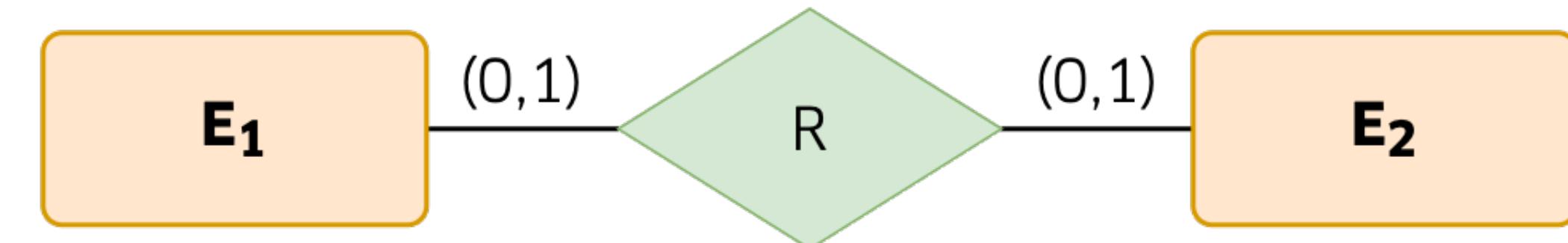
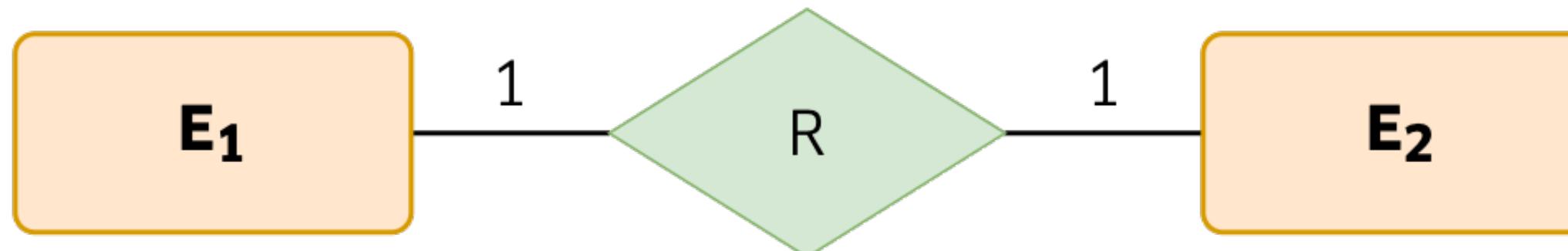
## Min-Max-Notation

- Jeder Mitarbeiter arbeitet an 0 bis 3 Projekten und ein Projekt wird von einem oder zwei Mitarbeitern bearbeitet.
- Jeder Mitarbeiter arbeitet in genau einer Abteilung und eine Abteilung besteht aus mind. 3, max. aber 20 Mitarbeitern.
- Gegenüber gestellt die Chen-Notation. Achtung: Umgekehrt!
- Wie bei Chen auch, darf ein '\*' oder 'N' verwendet werden, wenn eine Entität beliebig oft in einer Relation enthalten sein darf.



## Relationen und Kardinalität

### Chen-Notation vs. Min-Max-Notation



Hier unbedingt die Unterschiede und unterschiedlichen Ausprägungen in den Relationen-Tabellen verstehen.



## Relationen und Kardinalität

---

### Kardinalität in UML-Notation

- Idee: Kardinalität wie in der Chen-Notation, aber mit Min-Max Angaben.



... nur nicht verwirren lassen ...

### Hintergrund UML

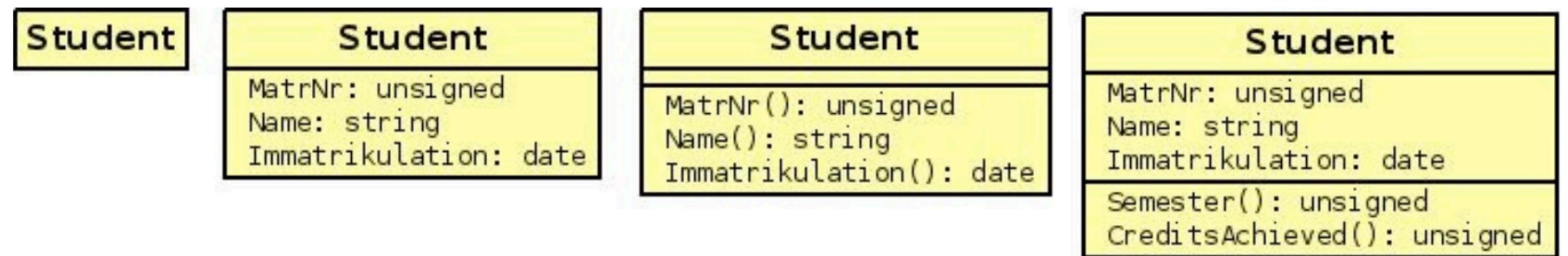
- Notation zur objektorientierten Modellierung (siehe z.B. Booch, Rumbaugh, Jacobson).
- Standardisierung 1997 Object Management Group (OMG).
- Verschiedene Diagrammtypen → Thema in SWE.

## UML Klassen

---

### Klassensymbol

- Rechteck mit drei Sektionen für
  - Klassenname,
  - Attribute (optional) und
  - Methoden (optional).
  - Hier werden häufig nur die relevanten Details gezeigt.



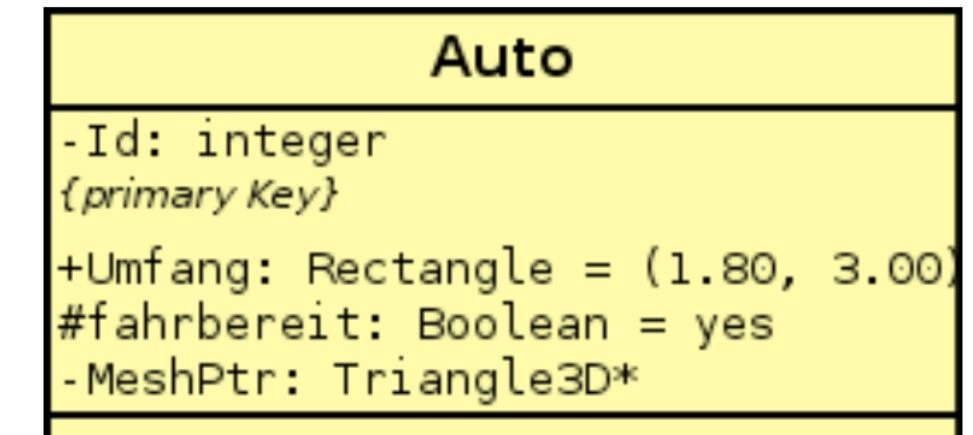
UML-Diagramme  
Prof. Striegnitz

## UML Klassen

---

### Attributspezifikation

- Sichtbarkeit: Name:Typ = Vorgabewert { Eigenschaften }



### Operationen/Methoden

- Sichtbarkeit: Name(Parameterliste): Rückgabetyp { Eigenschaften }

UML-Diagramme  
Prof. Striegnitz

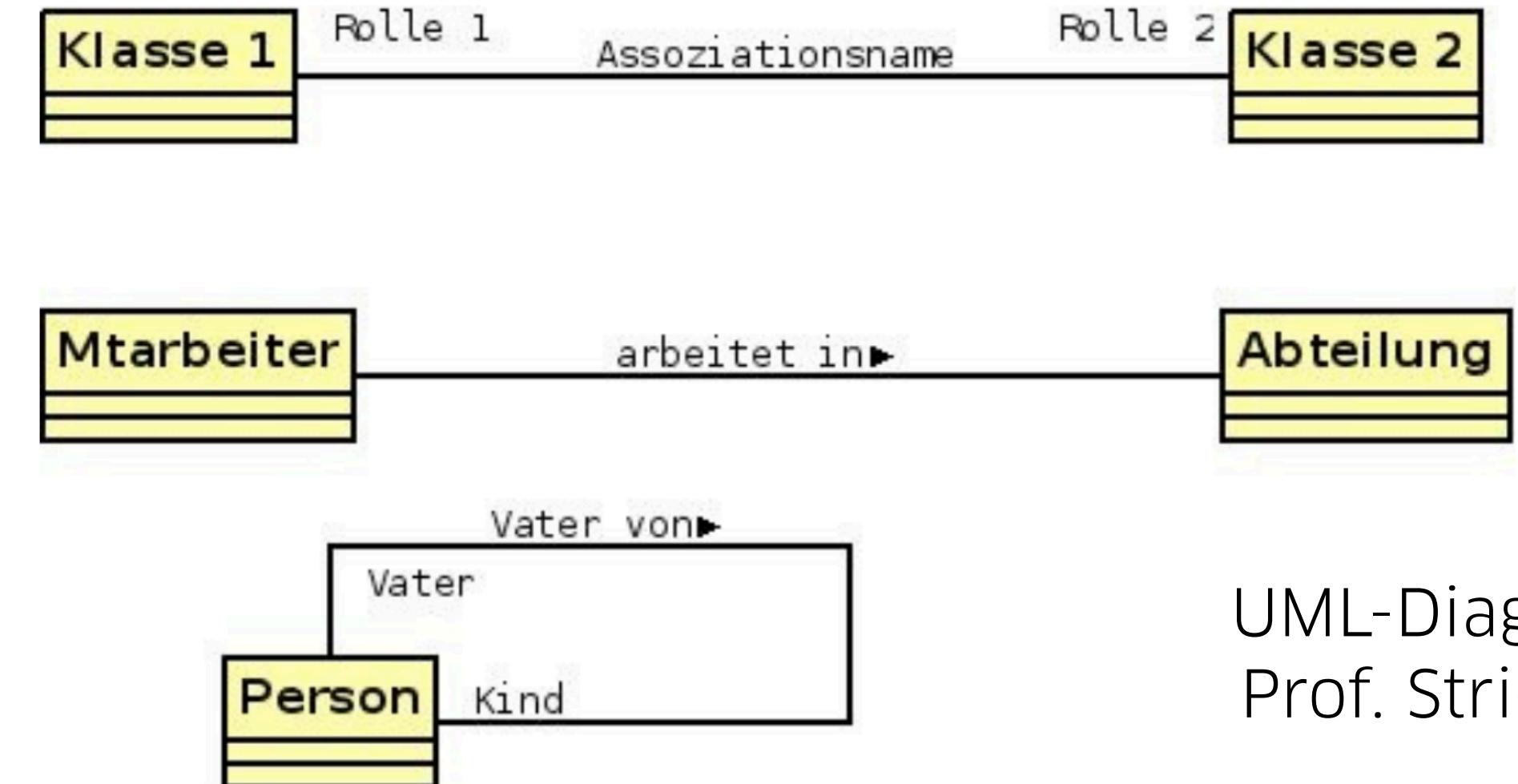
### Notationen

- Sichtbarkeit: +:public -:private #:protected
- Unterstrichene Attribute/Operationen stehen für Klassenattribute/Operationen.
- Eigenschaften können Bedingungen (constraints) sein.
- Schlüsselattribute z.B. als Kommentar / Eigenschaft (vgl. Attributierung).

## UML Assoziationen

### Repräsentation von Relationen

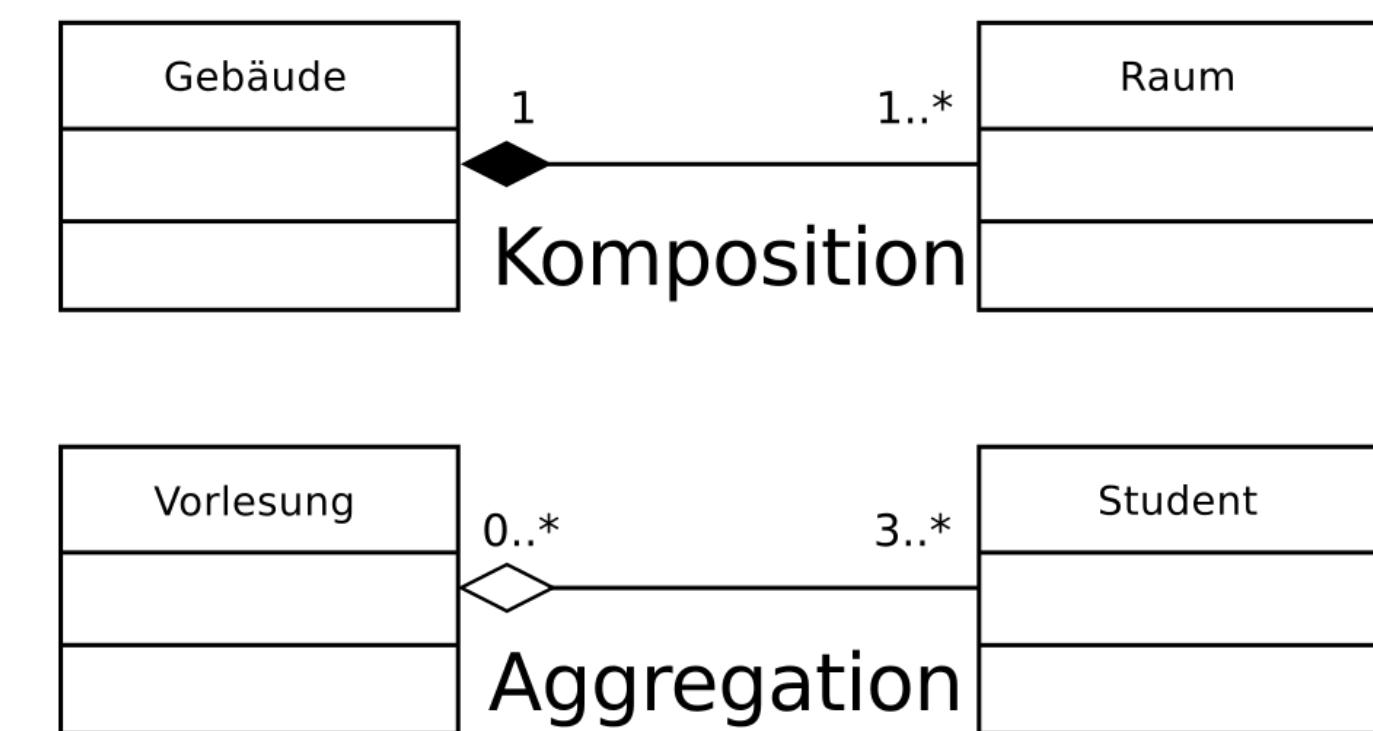
- Optional
  - Assoziationsname
  - Leserichtung ( $\blacktriangleright$  oder  $\blacktriangleleft$ )
  - Rollenname, Sichtbarkeit von Rollen
  - Kardinalitäten



UML-Diagramm  
Prof. Striegnitz

### Spezialfälle der Assoziation

- Komposition: von der Existenz des Ganzen abhängig
- Aggregation: eine "Teil von"-Beziehung

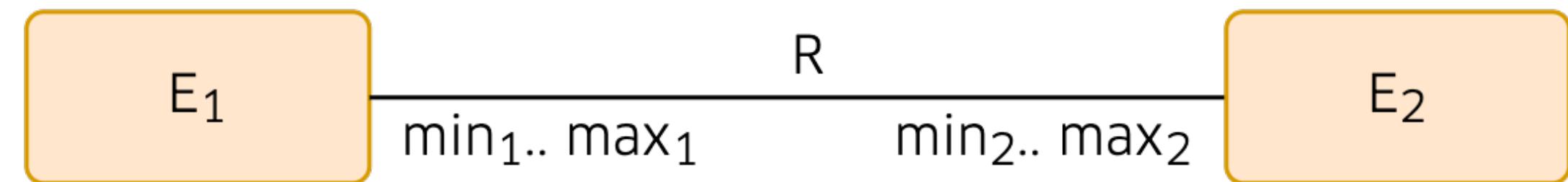


## Relationen und Kardinalität

---

### UML-Notation

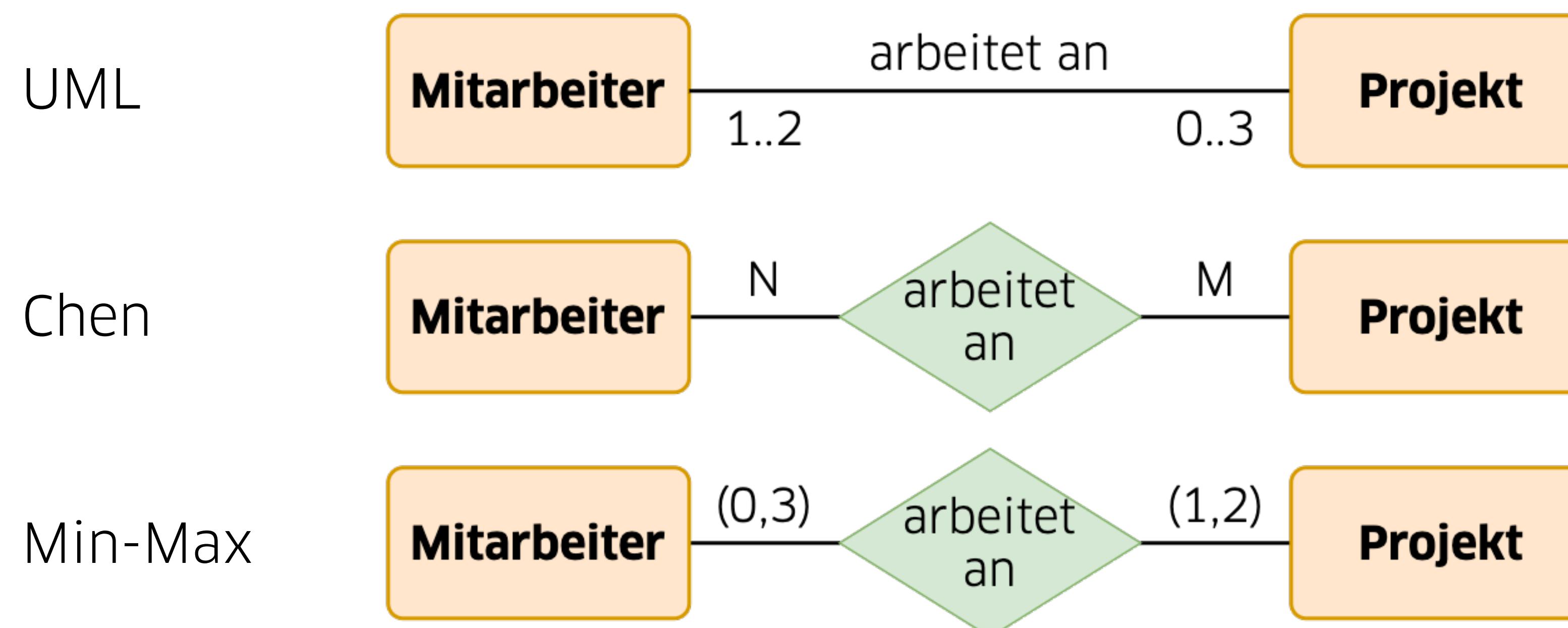
- Für jeden an einer Relation beteiligten Entitätstyp  $E_k$  wird angegeben, wie oft eine Entität minimal und maximal in  $E_k$  enthalten sein darf, wenn der andere Eintrag ( $n=2$ ) bzw. die anderen Einträge ( $n>2$ ) fest gewählt ist bzw. sind.
- Gleiche Sichtweise wie bei der Chen-Notation, nur mit Angabe der minimalen und maximalen Elemente in der Entitätenmenge.
- Dadurch ist eine Angabe der Fälle
  - *optional*, also Kardinalität=0, und
  - *obligatorisch*, also Kardinalität>0, möglich.



## Relationen und Kardinalität

### UML-Notation im Vergleich

- Jeder Mitarbeiter arbeiten an keinem, höchstens aber an 3 Projekten mit.
- Ein Projekt wird von einem oder zwei Mitarbeitern bearbeitet.

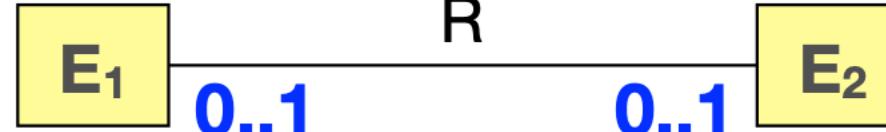
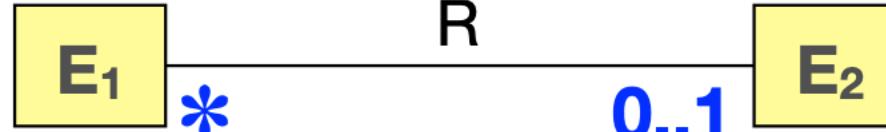
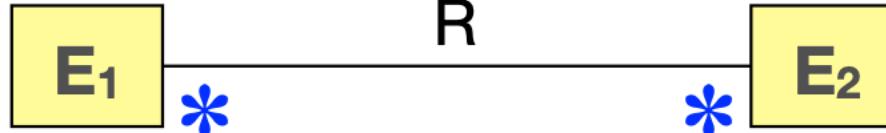


## Relationen und Kardinalität

---

### Vollständige vs. unvollständige UML-Relationen

- Vollständig, d.h. alle Entitäten nehmen teil.
- Unvollständig, d.h. es gibt Entitäten, nicht teilnehmen.
- Angabe in Kurzform, d.h. statt '0..\*' kurz '\*' und statt '1..1' kurz '1', Default ist '1'.

Typ	unvollständig	vollständig
<b>One-to-One</b>		
<b>Many-to-One</b>		
<b>Many-to-Many</b>		

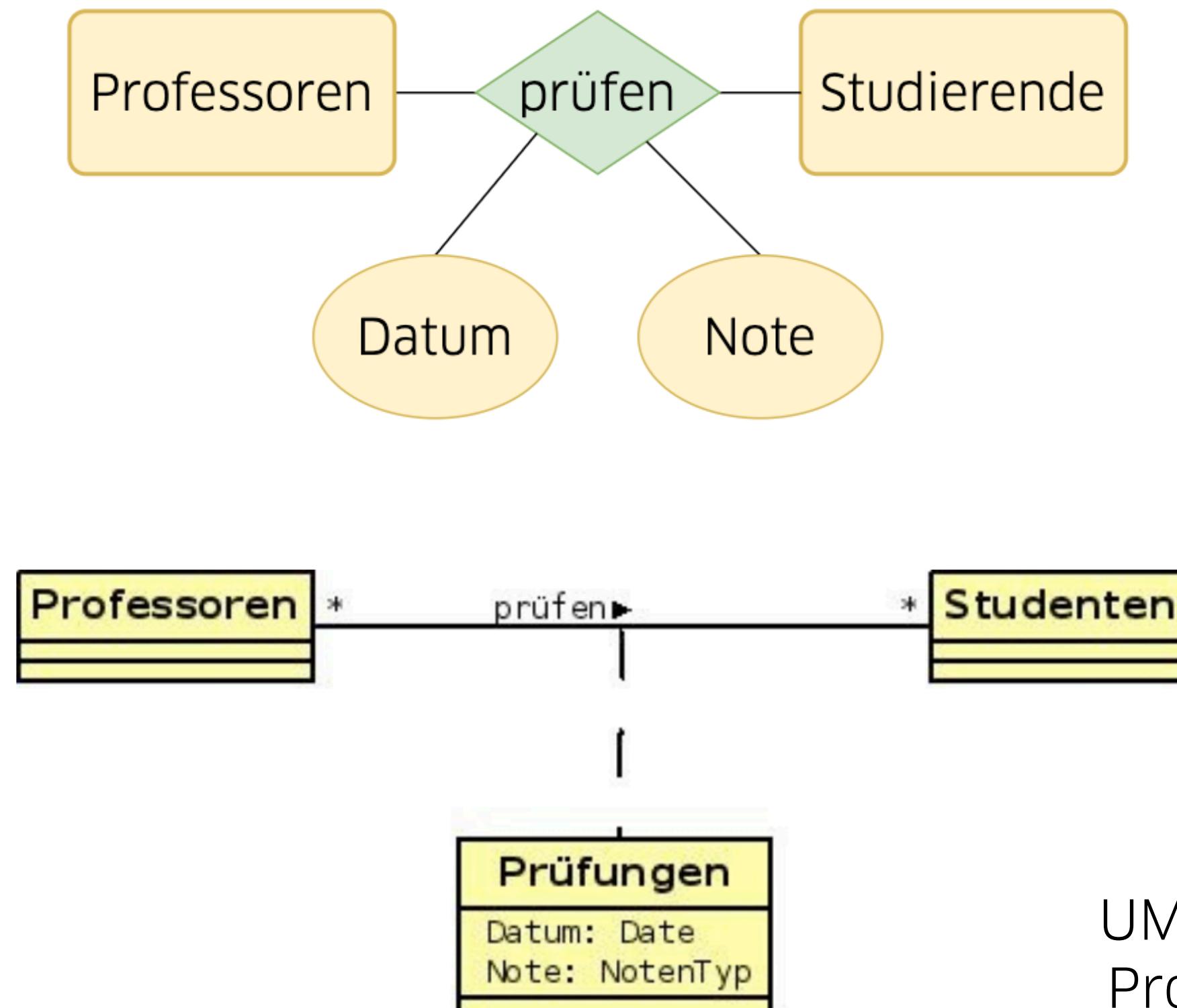
UML-Diagramm  
Prof. Striegnitz

## Relationen und Kardinalität

---

### UML-Relationen mit Attributen bzw. Relationen mit assoziierten Klassen

- Annotation von Relationen mit Attributen, in UML zusammengefasst in einer assoziierten Klasse.

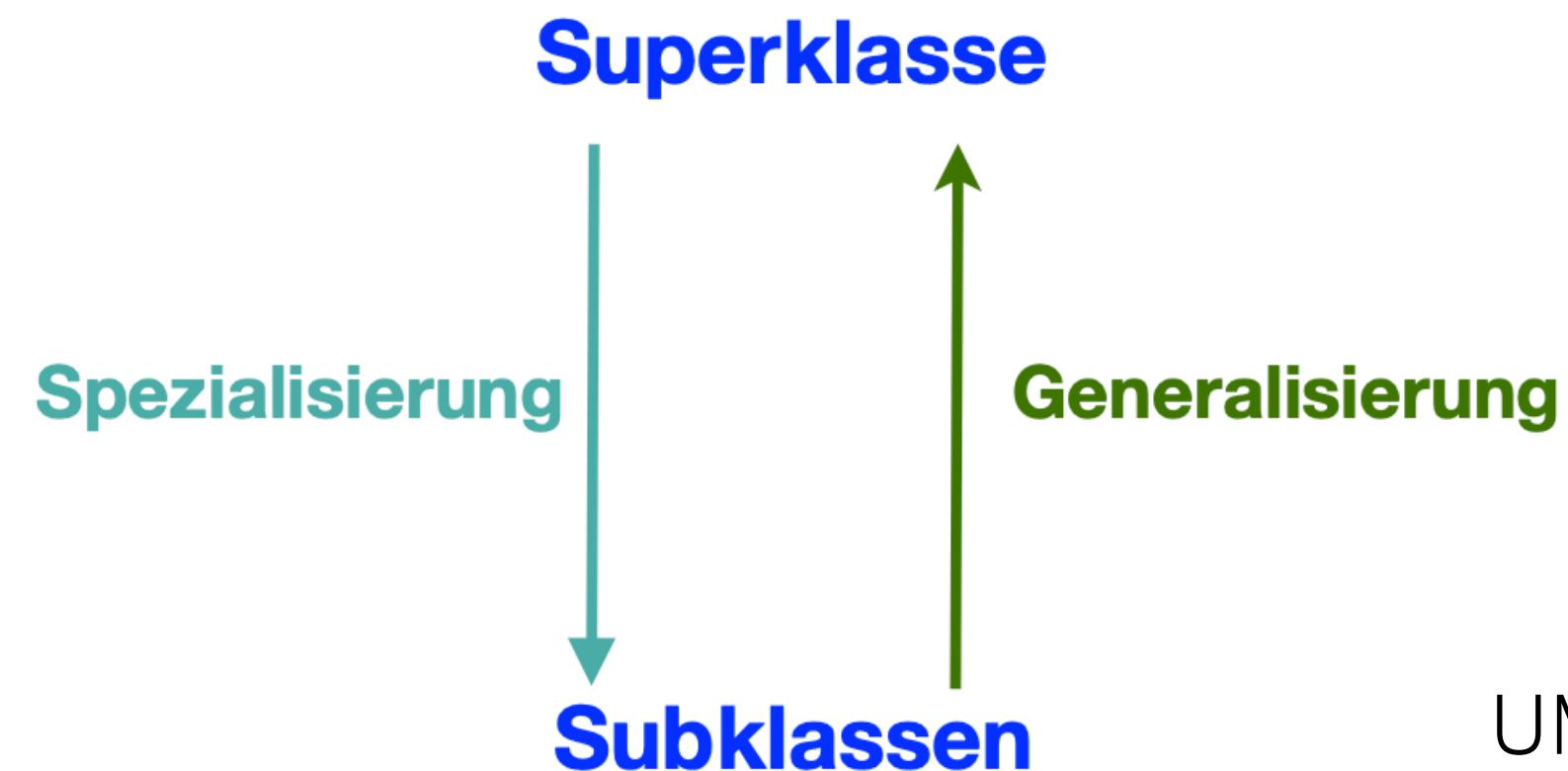
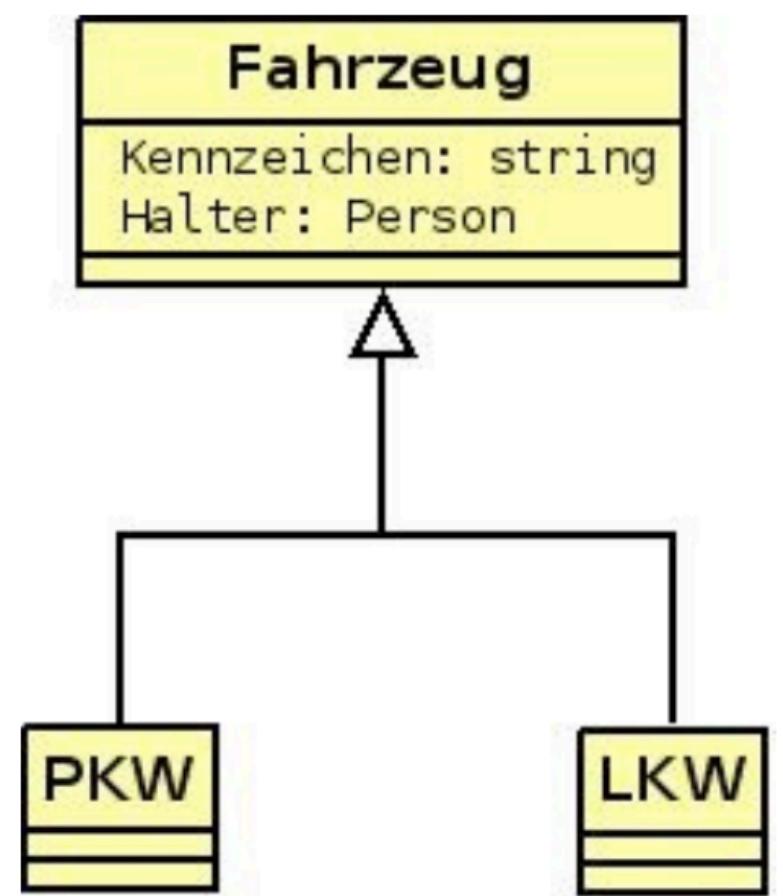


UML-Diagramm  
Prof. Striegnitz

## Generalisierung, Spezialisierung

### Ist-Ein-Beziehung

- Inklusionspolymorphie, d.h. jedes Objekt aus  $E_1$  ist auch ein Objekt aus  $E_2$ , kurz  $E_1$  'ist-ein'  $E_2$
- Vererbung von Eigenschaften bzw. Attributen der Superklasse an alle Subklassen



UML-Diagramm  
Prof. Striegnitz

## Generalisierung, Spezialisierung

### Inklusionspolymorphie

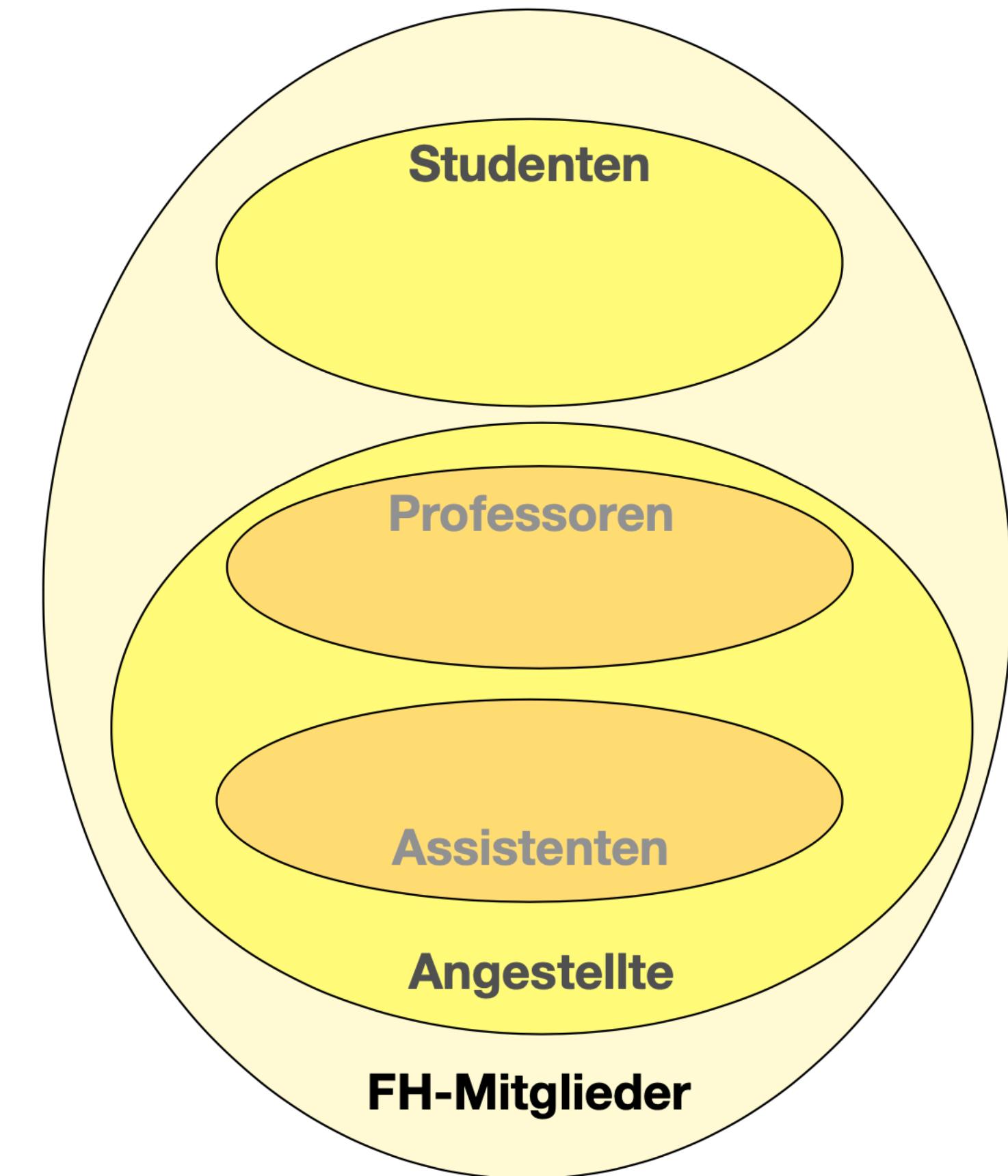
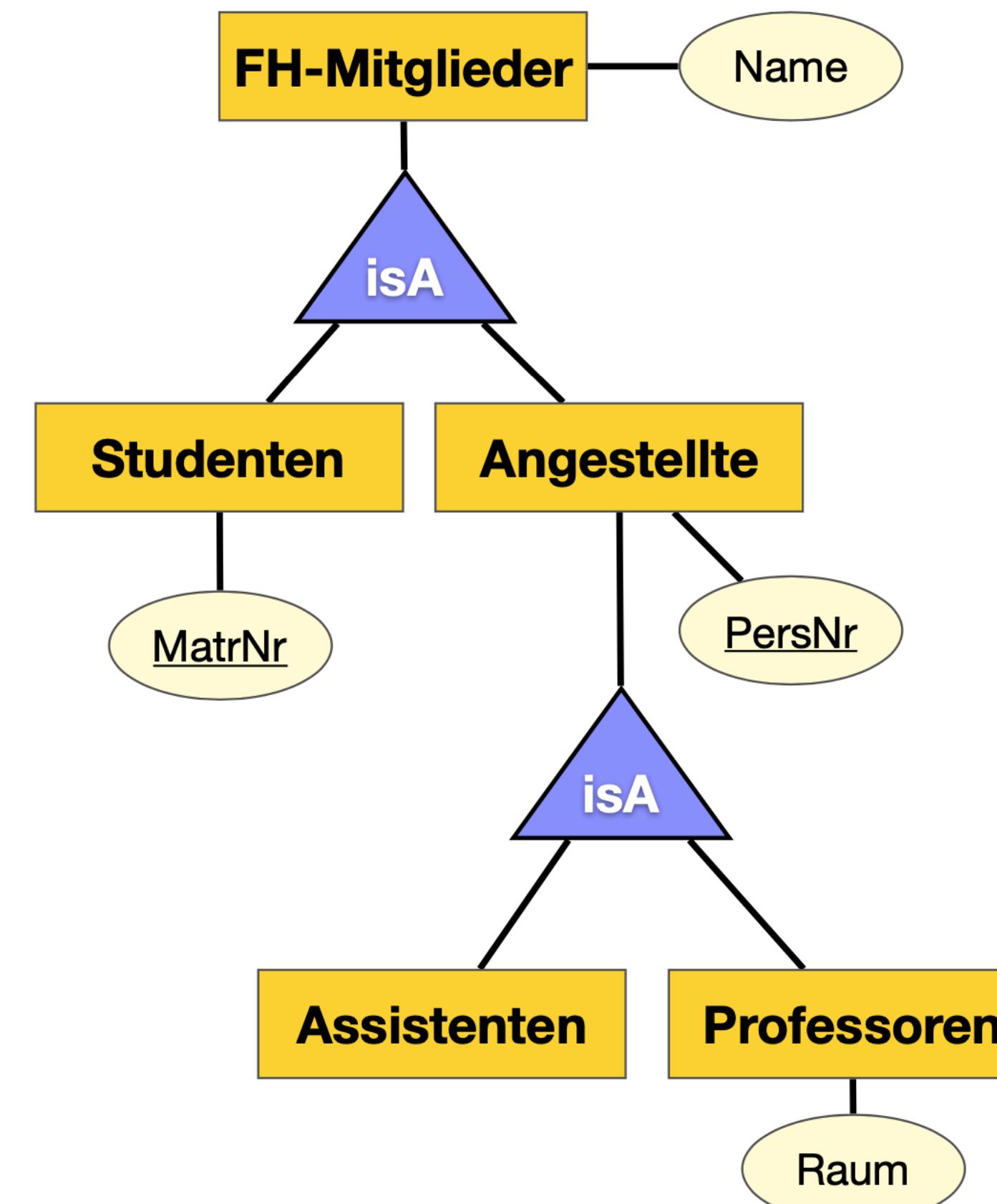


Diagramm  
Prof. Striegnitz

## Generalisierung, Spezialisierung

### Disjunkte / nicht disjunkte Vererbung

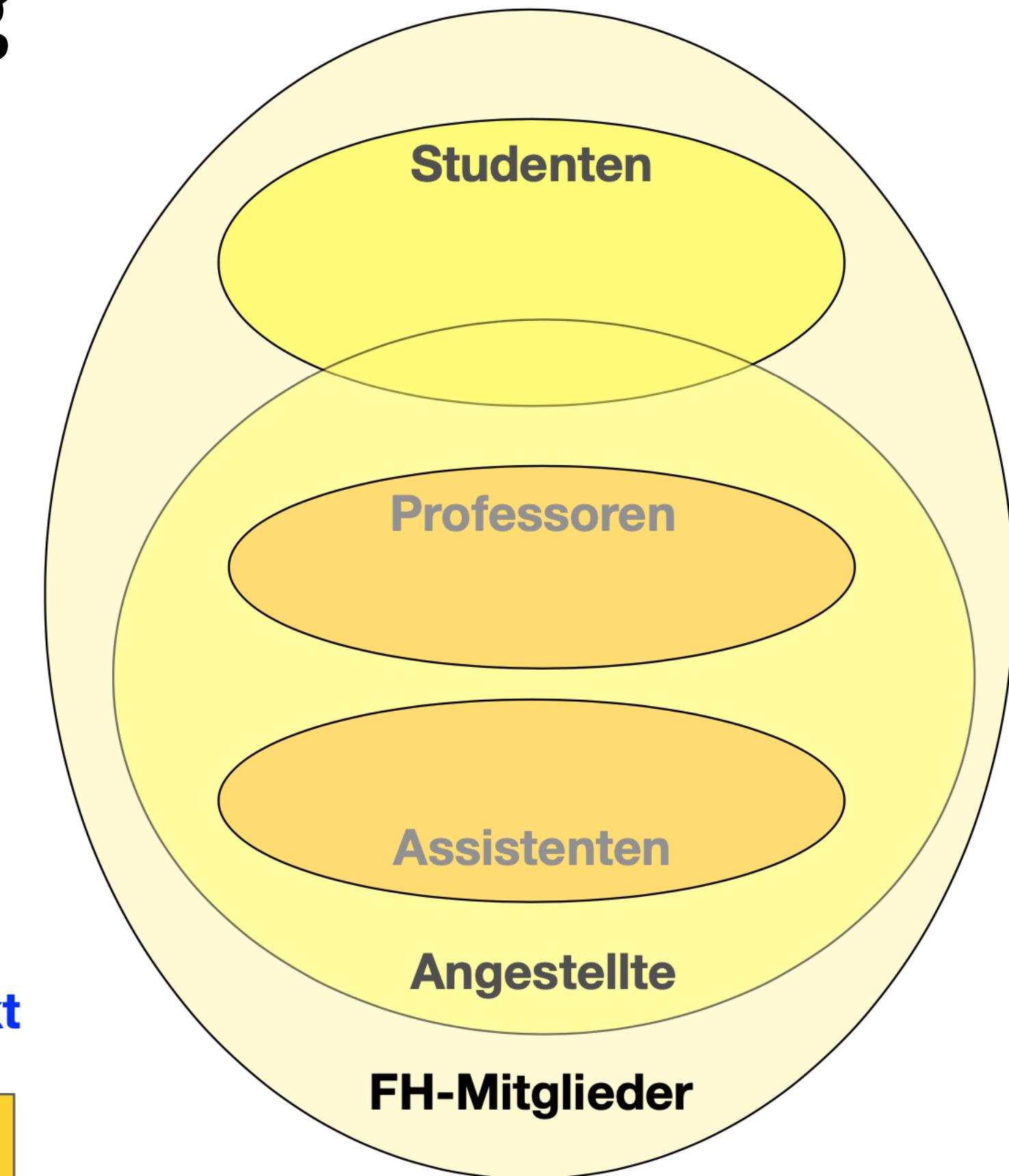
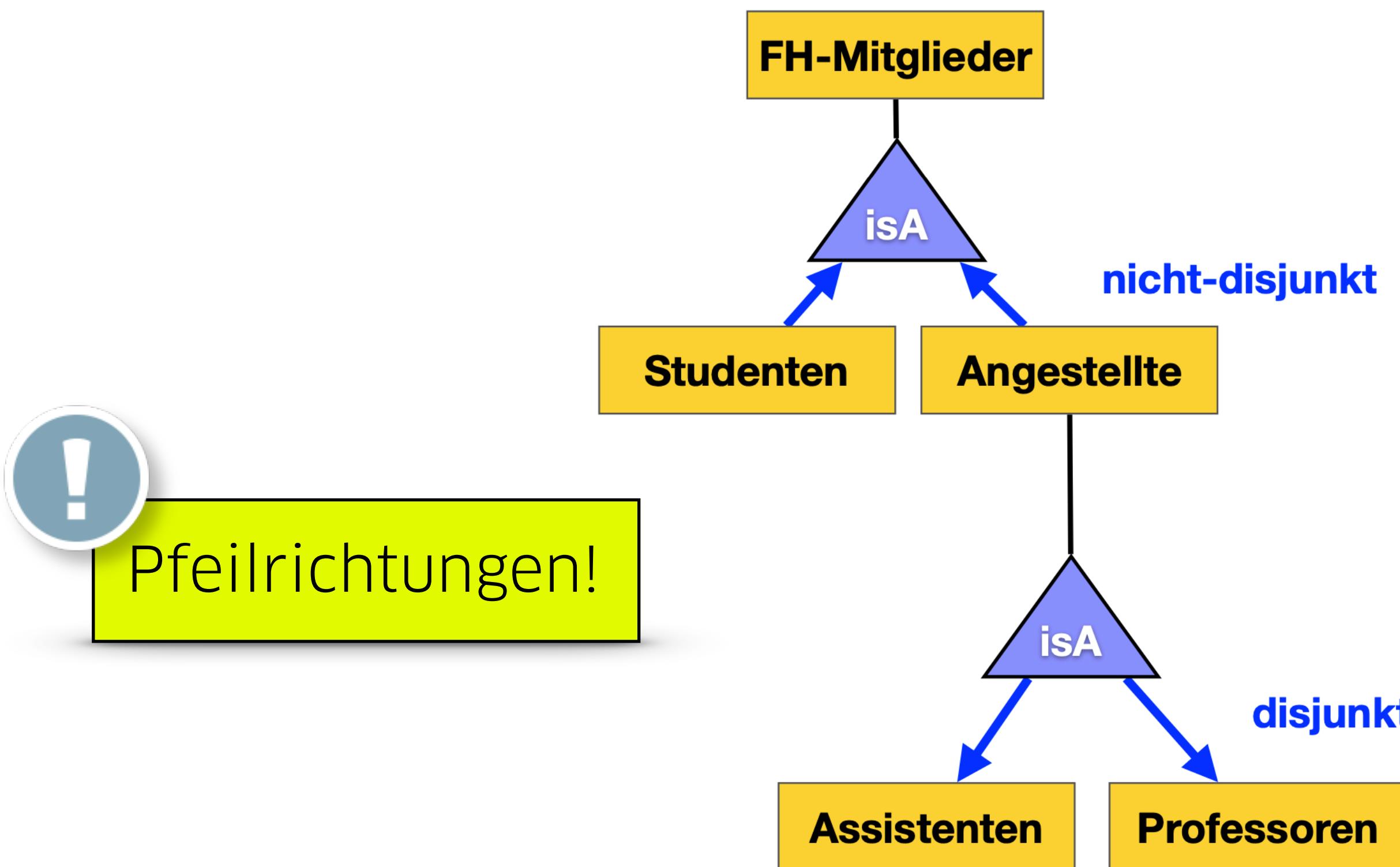


Diagramm  
Prof. Striegnitz

## Generalisierung, Spezialisierung

### Beispiel

- Randbedingungen
  - Nur Vierecke mit Innenwinkel von  $90^\circ$ .
  - Ein Quadrat sei kein Sonderfall eines Rechtecks.
- 't': *Totale Spezialisierung*, d.h. es gibt nur Rechtecke und Quadrate und sonst nichts, bzw.  
 $\text{Rechtecke} \cup \text{Quadrate} = \text{Vierecke}$
- *Disjunkte Spezialisierung*, d.h. keine Überlappung, bzw.  
 $\text{Rechtecke} \cap \text{Quadrate} = \emptyset$

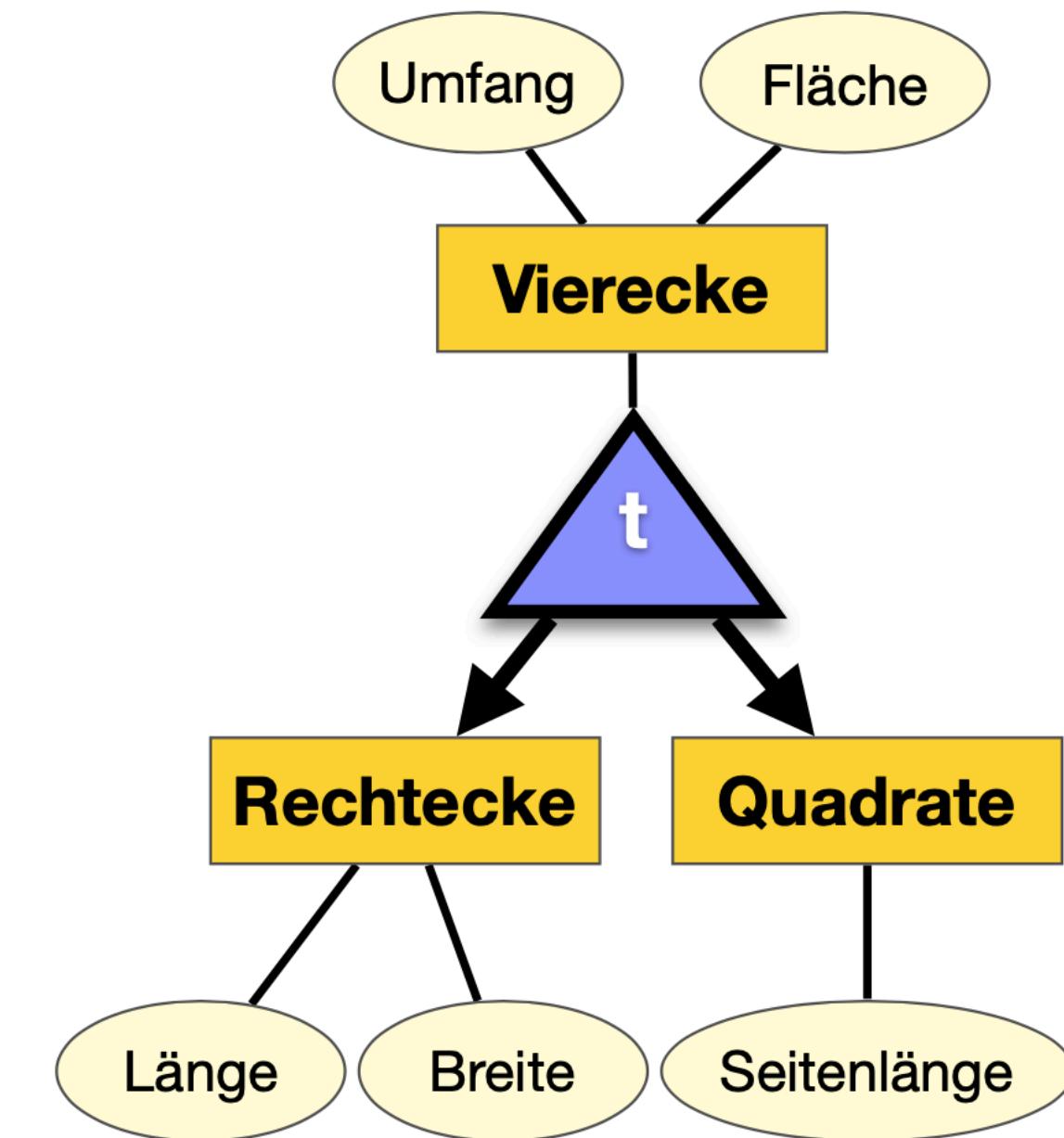


Diagramm  
Prof. Striegnitz

## Generalisierung, Spezialisierung

---

### Beispiel

- Randbedingungen
  - Nur Vierecke mit Innenwinkel von  $90^\circ$ .
  - Quadrat ist als Sonderfall eines Rechtecks erlaubt.
- 't': *Totale Spezialisierung*, d.h.  
 $\text{Rechtecke} \cup \text{Quadrate} = \text{Vierecke}$
- *Nicht-Disjunkte Spezialisierung*, d.h.  
 $\text{Rechtecke} \cap \text{Quadrate} \neq \emptyset$

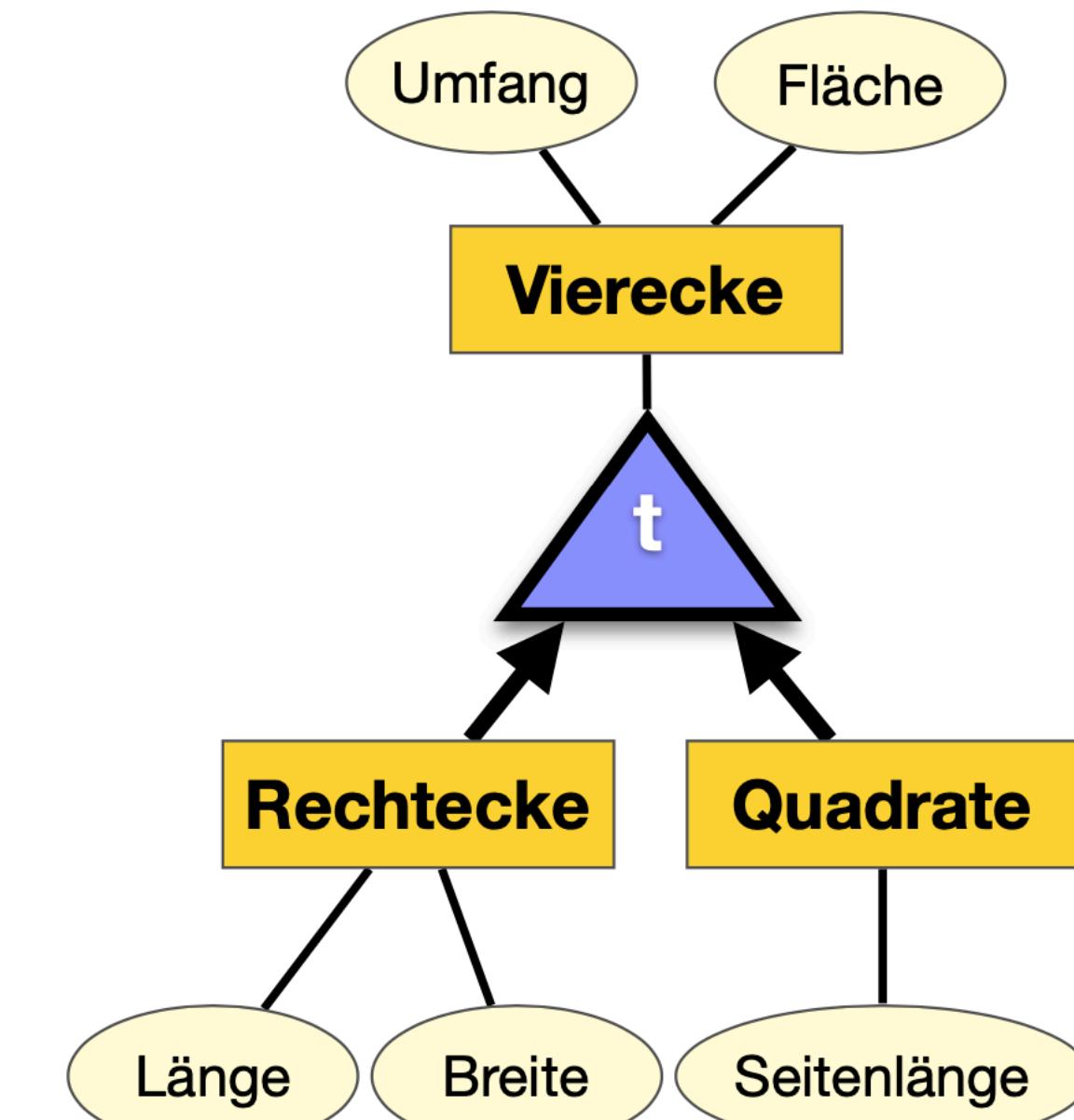


Diagramm  
Prof. Striegnitz

## Generalisierung, Spezialisierung

### Beispiel

- Randbedingungen
  - Beliebige Vierecke erlaubt, also auch Trapez, Parallelogram etc.
  - Ein Quadrat sei kein Sonderfall eines Rechtecks.
- 'p': *Partielle Spezialisierung*, d.h. es gibt noch mehr, was ggf. aber hier nicht aufgeführt ist, bzw.  
 $\text{Rechtecke} \cup \text{Quadrate} \subset \text{Vierecke}$
- *Disjunkte Spezialisierung*, d.h.  
 $\text{Rechtecke} \cap \text{Quadrate} = \emptyset$

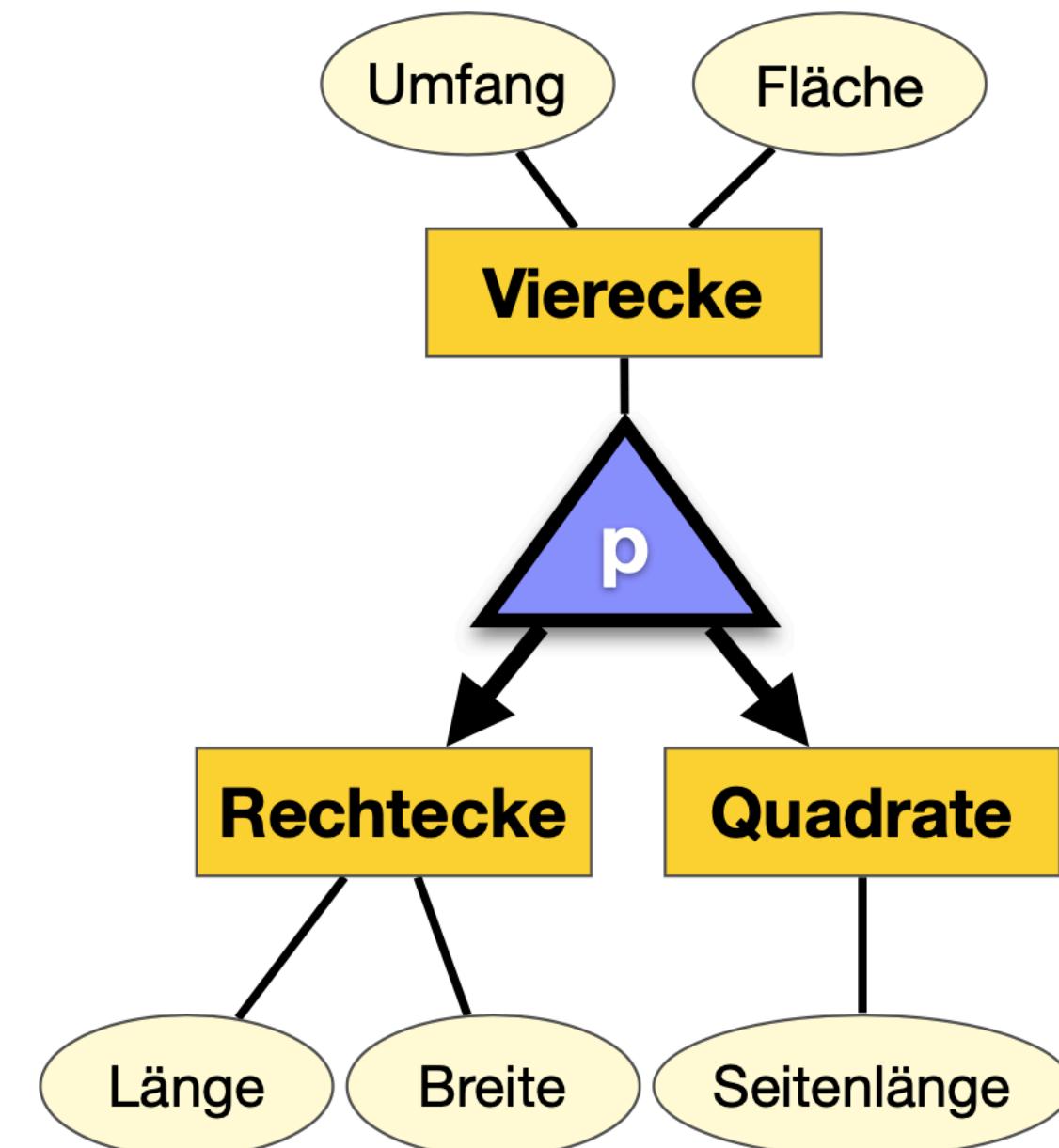


Diagramm  
Prof. Striegnitz

## Generalisierung, Spezialisierung

### Beispiel

- Randbedingungen
  - Beliebige Vierecke erlaubt, also auch Trapez, Parallelogram etc.
  - Quadrat ist als Sonderfall eines Rechtecks erlaubt.
- 'p': *Partielle Spezialisierung*, d.h.  
Rechtecke  $\cup$  Quadrate  $\subset$  Vierrecke
- *Nicht-Disjunkte Spezialisierung*, d.h.  
Rechtecke  $\cap$  Quadrate  $\neq \emptyset$

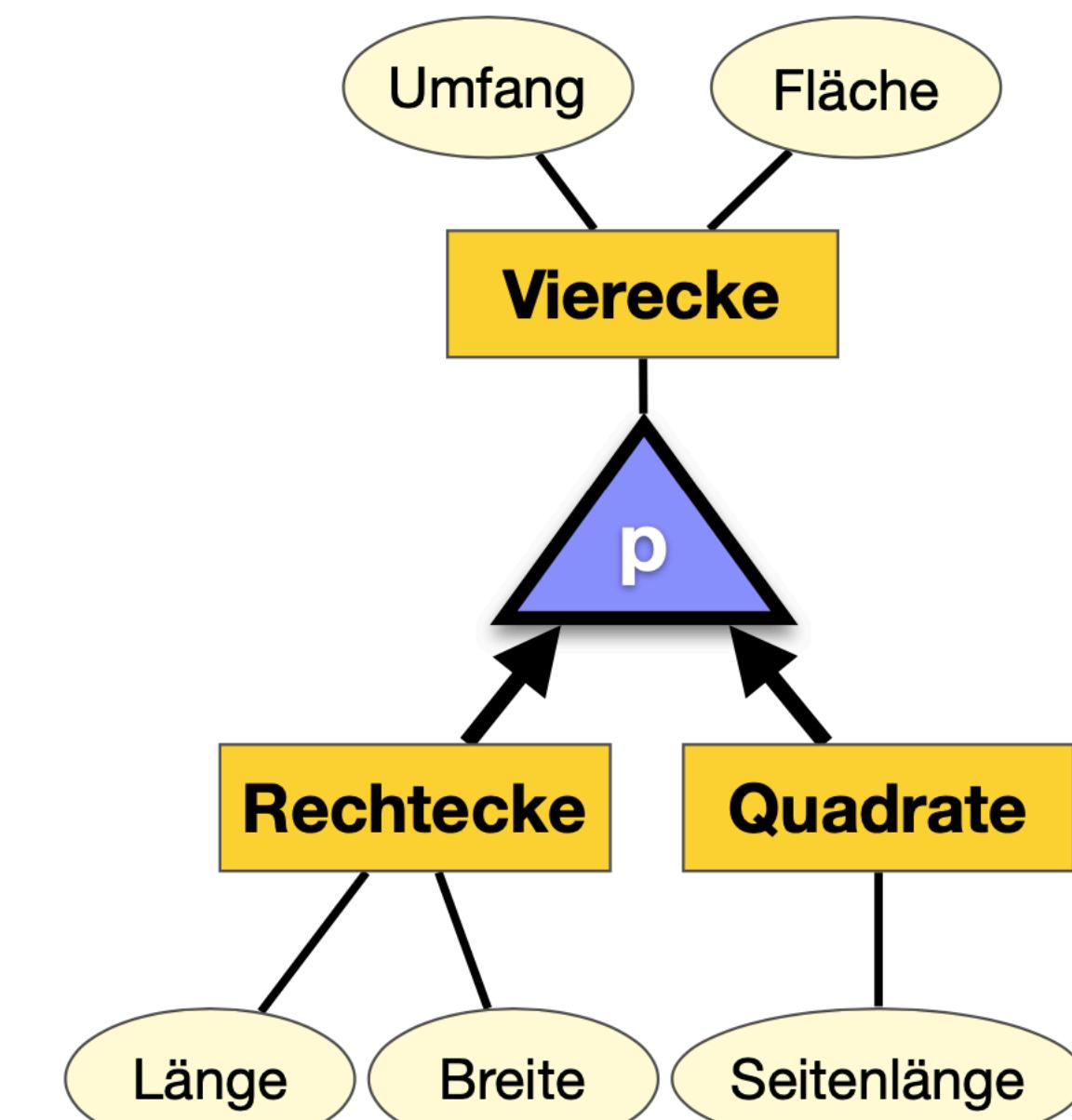


Diagramm  
Prof. Striegnitz

## Generalisierung, Spezialisierung

### Anmerkungen

- Behandlung von Spezialisierungshierarchien möglich, z.B. Modellierung typischer Objektwelten, aber:
- Abbildung in ein (relationales) DBMS nicht trivial.  
Mgl. Ansatz hier: sog. ORM (Objekt-relationale Mapper), ggf. sind aber nicht-relationale DBMS besser?

#### Q&A

- Wo liegt das Problem?

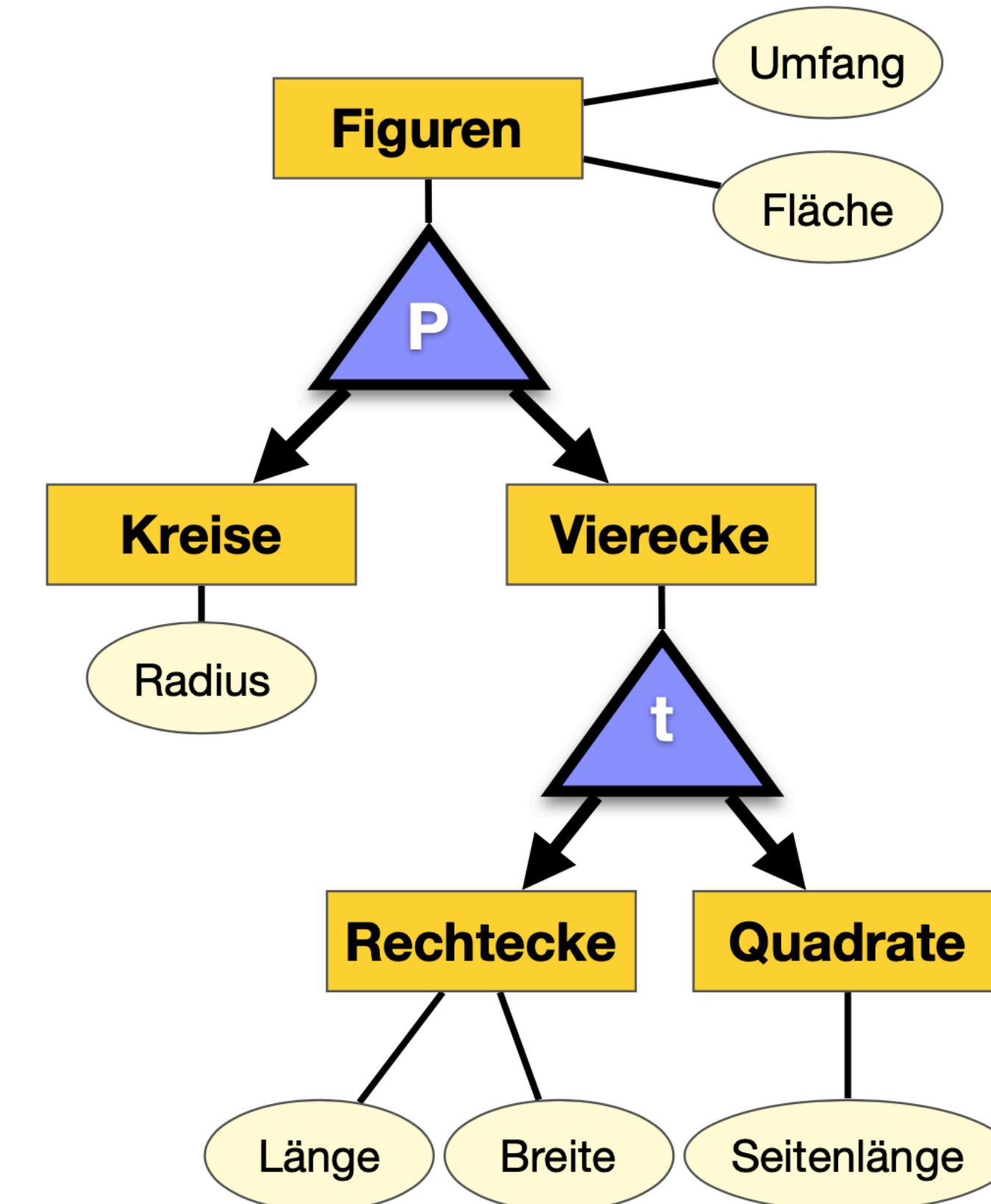


Diagramm  
Prof. Striegnitz

## Zusammenfassung

---

### Was ist wichtig?

- Weitere Elemente ER-Diagramme
  - Kardinalität in Chen-, Min-Max- und UML-Notation.
  - n-stellige Relationen
- Generalisierung, Spezialisierung

### Was folgt?

- Implementationsentwurf und 'Relationales Modell'.

## Fallbeispiel (Quelle Kemper/Eickler: Datenbanksysteme)

---

### Informationen aus Interviews

- Prüfungsamt: “Studierende erstellen Bachelorarbeiten zu einem bestimmten Thema und werden dabei von Assistenten betreut, die in der Regel an Ihrer Dissertation arbeiten. Das Thema der Dissertation wird mit dem betreuenden Professor abgestimmt.”
- Fachbereichsbibliothek: “Wir besitzen zahlreiche Dokumente, für die wir in einer Kartei Titel, Autoren und Jahr speichern. Zur eindeutigen Zuordnung besitzt jedes Dokument eine eindeutige Signatur. Geleitet wird die Bibliothek von Mitgliedern der Universität - das ist auch die Personengruppe, an welche Dokumente ausgeliehen werden können.”
- Dozenten: “Wir empfehlen für unsere Vorlesungen oft begleitende Literatur. Das sind meist Bücher. Den Studenten nennen wir jeweils den Titel, die Autoren, den Verlag und das Erscheinungsjahr.”

## Fallbeispiel

---

### Prüfungsamt

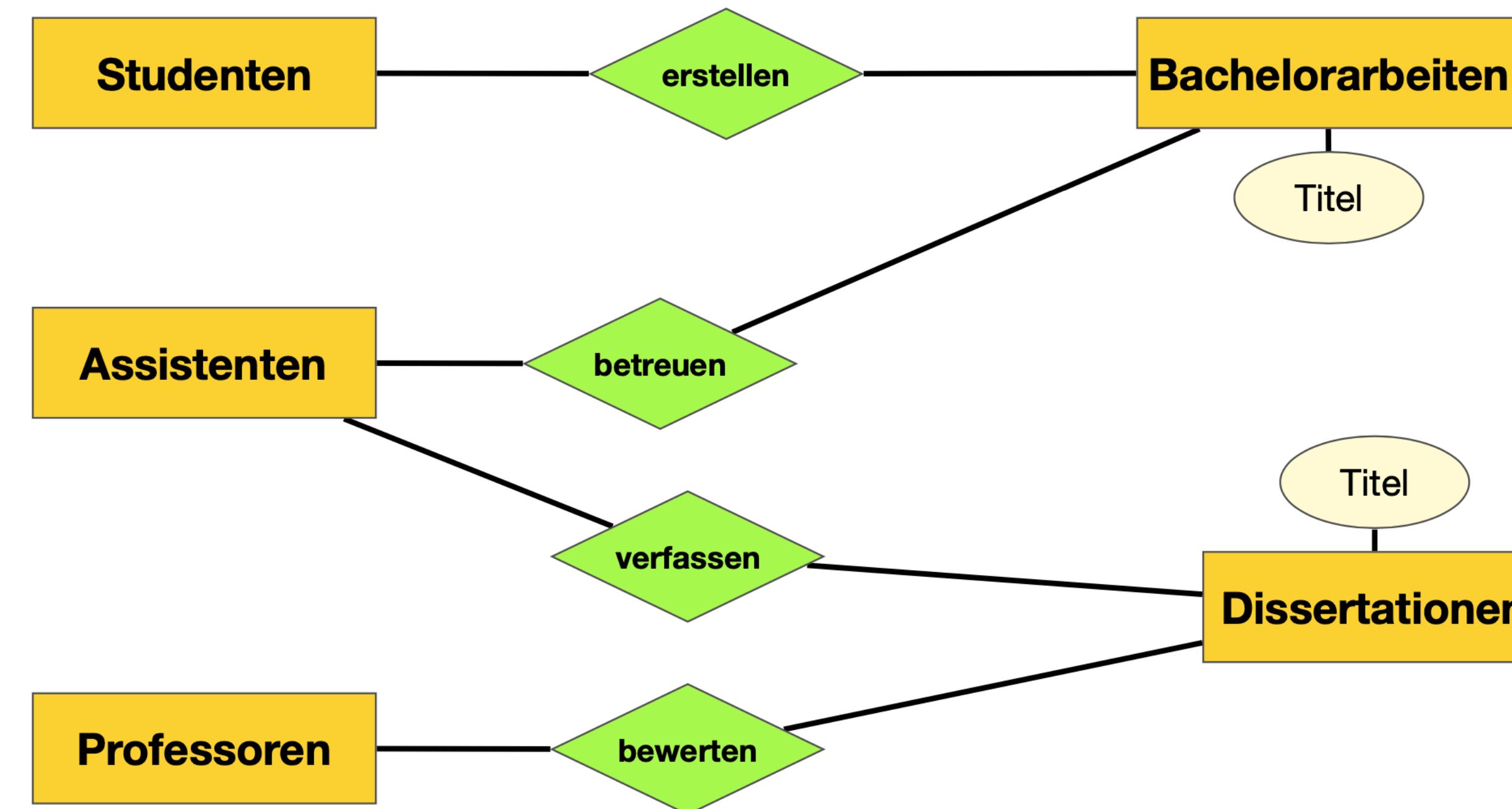


Diagramm Prof. Striegnitz  
Quelle Kemper/Eickler

## Fallbeispiel

---

### Fachbereichsbibliothek

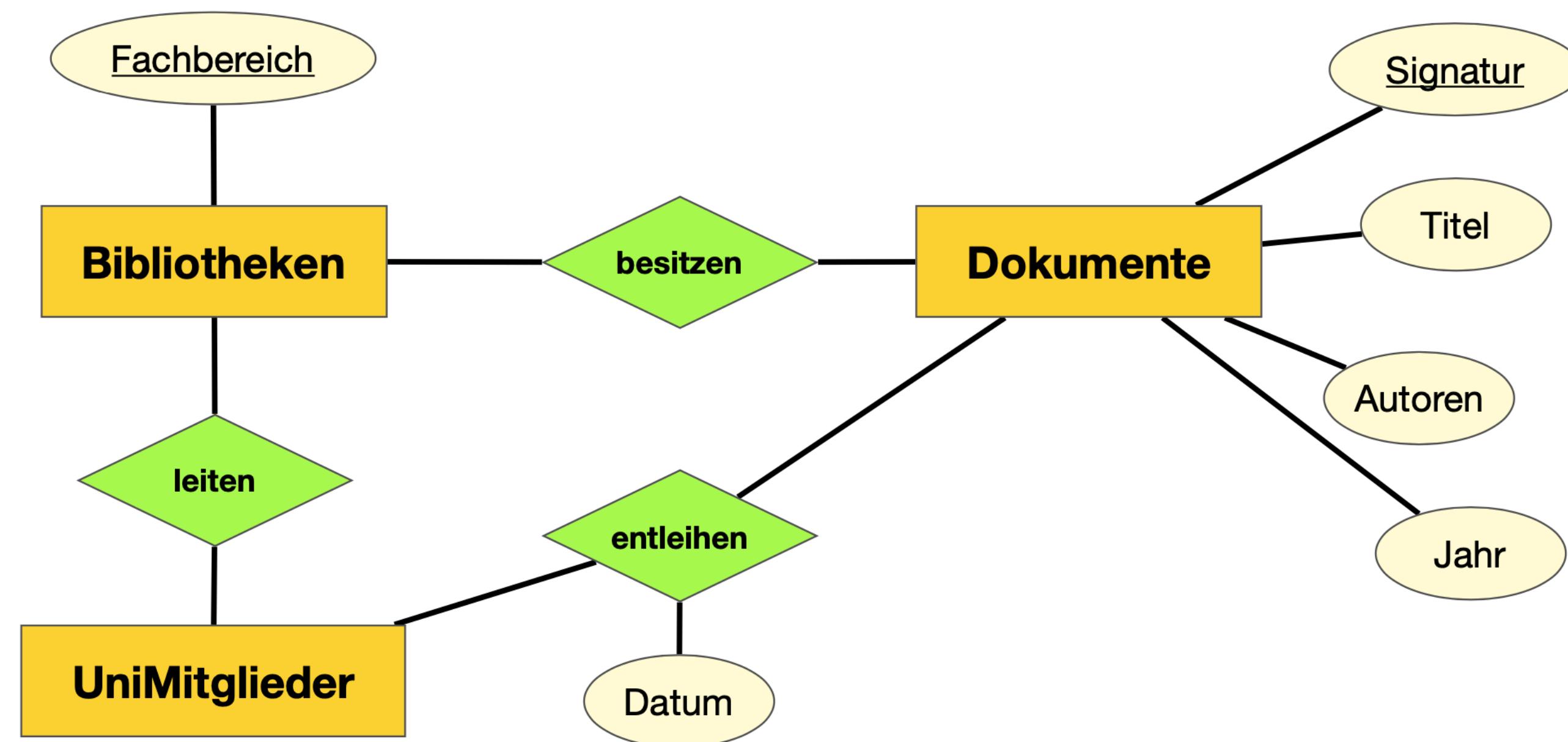


Diagramm Prof. Striegnitz  
Quelle Kemper/Eickler

## Fallbeispiel

---

### Dozenten

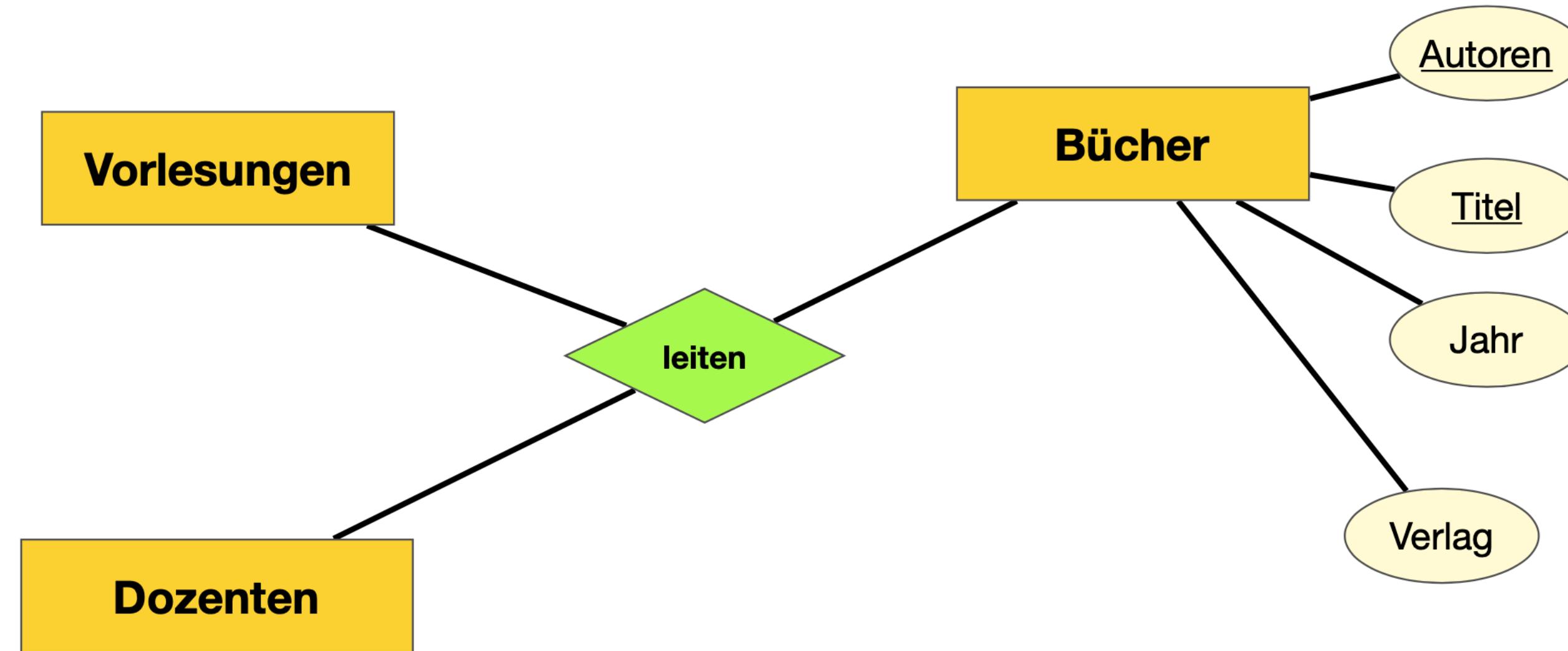


Diagramm Prof. Striegnitz  
Quelle Kemper/Eickler

## Fallbeispiel

---

### Beobachtungen I

- Begriffe *Dozenten* und *Professoren* sind synonym verwendet worden.
- Der Entitytyp *UniMitglieder* ist eine Generalisierung von *Studenten*, *Professoren* und *Assistenten*.
- Fakultätsbibliotheken werden von Angestellten geleitet.
- Die Beziehung *leiten* der Fachbereichsbibliothek ist revisionsbedürftig, sobald wir im globalen Schema ohnehin eine Spezialisierung von *UniMitglieder* in *Studierende* und *Angestellte* vornehmen - Studierende leiten keine Bibliotheken.
- Dissertationen, Diplomarbeiten und Bücher sind Spezialisierungen von Dokumenten, die in den Bibliotheken verwaltet werden.
- Wir können davon ausgehen, dass alle an der Universität erstellten Abschlussarbeiten und Dissertationen in Bibliotheken verwaltet werden.

## Fallbeispiel

---

### Beobachtungen II

- Die durch das Prüfungsamt festgelegten Beziehungen erstellen und verfassen modellieren denselben Sachverhalt wie das Attribut *Autoren* von Büchern der Sicht der Dozenten.
- Alle in einer Bibliothek verwalteten Dokumente werden durch eine Signatur identifiziert.



In einem konsolidierten Schema müssen die Begriffe eindeutig sein und es müssen sich die 'Partner' wiederfinden!

## Fallbeispiel

### Konsolidiertes Schema

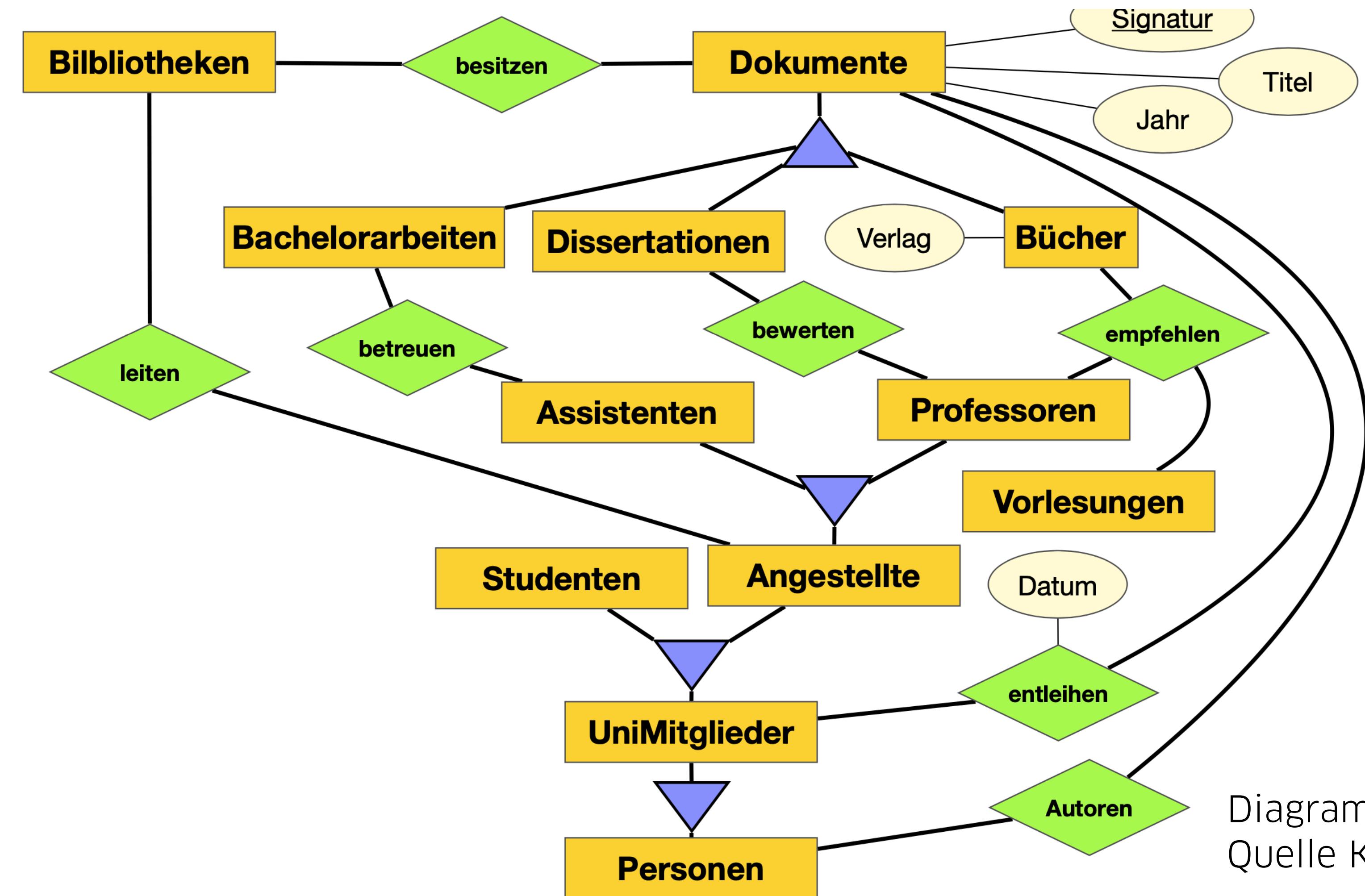


Diagramm Prof. Striegnitz  
Quelle Kemper/Eickler

## Vorschau

---

### **Relationen im RDBMS**

Bei der Umsetzung von 1:1, 1:N und N:M-Relationen eines ER-Diagramms (Modellierung Mini-Welt) in eine Datenbankstruktur gibt es, nimmt man mal hierarchische Strukturen aus, nicht so viele Möglichkeiten.

Diese sollen, für das Verständnis auch des folgenden Kapitels, im Folgenden kurz beispielhafte an Relationen aus `matse_mhist` vorgestellt werden. Hier ist noch wichtig, dass wir Entitäten immer über eine `id` identifizieren.

Hintergründe dann in den kommenden Kapiteln.

## Vorschau

---

### 1:N-Relation im RDBMS

- Eine Möglichkeit der Umsetzung sind zwei Tabellen für die Entitätstypen und die Realisierung der Relation als *Fremdschlüssel* in der Tabelle mit der N-Kardinalität !



	id	name
1	1	Mia
2	2	Ben
3	3	Emma
4	4	Paul
5	5	Hannah

mitarbeiter

	id	name	mitarbeiter_id
1	1	Petra	4
2	2	Mini	18
3	3	Rosa	<null>

tier

### Q&A

- Wieso ist der Fremdschlüssel beim Tier?

## Vorschau

### 1:N-Relation im RDBMS

- Eine andere Möglichkeit der Umsetzung sind zwei Tabellen für die Entitätstypen und eine eigene Tabelle für die Realisierung der Relation mit zwei Fremdschlüsseln.



	id	name
1	1	Vorstand
2	2	HR/Buchhaltung
3	3	Vertrieb
4	4	Marketing
5	5	Einkauf

abteilung

	abteilung_id	standort_id
1	1	1
2	2	1
3	3	2
4	3	1
5	3	3

sitzt\_am

	id	ort
1	1	Aachen
2	2	Jülich
3	3	Köln
4	4	Berlin

standort

### Q&A

- Hier gibt es einen Fehler...

## Vorschau

---

### N:M-Relation im RDBMS

- Die Relation enthält z.B. (Vorstand,Aachen) und (Buchhaltung, Aachen), d.h. ein Standort kann *mehrere* Abteilungen beherbergen → N:1, Richtig.
- Sie enthält aber auch (Vertrieb,Aachen) und (Vertrieb,Jülich)...  
→ N:M



	id	name
1	1	Vorstand
2	2	HR/Buchhaltung
3	3	Vertrieb
4	4	Marketing
5	5	Einkauf

abteilung

	abteilung_id	standort_id
1	1	1
2	2	1
3	3	2
4	3	1
5	3	3

sitzt\_am

	id	ort
1	1	Aachen
2	2	Jülich
3	3	Köln
4	4	Berlin

standort

## Vorschau

### N:M-Relation im RDBMS

- Die einzige Möglichkeit der Umsetzung sind zwei Tabellen für die Entitätstypen und eine eigene Tabelle für die Realisierung der Relation mit zwei Fremdschlüssen.



	id	vom
1	1	2014-09-12 00:00:00
2	2	2014-08-30 00:00:00
3	3	2014-09-11 00:00:00

bestellung

	bestellung_id	produkt_id
1	2	11
2	1	15
3	3	15
4	1	16
5	3	17

besteht\_aus

	id	bezeichnung
3	3	Spinatpizza
4	4	Fischstäbchen
5	5	Nudelpfanne
6	6	Möhren

produkt

## Vorschau

---

### 1:1-Relation im RDBMS

- Eine Möglichkeit der Umsetzung ist über eine Fremdschlüsselbeziehung in *einer der beiden Tabellen* (wie im ersten Beispiel).
- Oder über eine eigene Relationen-Tabelle (wie im letzten Beispiel). Hier darf aber natürlich dann jede Entität nur einmal vorkommen!
- In `matse_mhist` kommt keine 1:1-Relation vor, da man die Attribute auch immer zu einem Entitätstyp zusammenfassen kann, wenn es inhaltlich gerechtfertigt ist.

### Zusammenfassung der Grundidee

- Die Fälle 1:1 und 1:N (N:1) können über eine eigene Relationen-Tabelle abgebildet werden oder nicht. Eine N:M-Relation benötigt immer eine eigene Relationen-Tabelle.
- Vor- und Nachteile besprechen wir noch.