



## 3. Informationsmodellierung

### Inhalt

DB-Entwurf und Modellierung  
Entity-Relationship-Modell (ERM)  
Erweiterungen des ERM  
Kardinalitätsrestriktionen  
Generalisierung und Vererbung  
Aggregationen

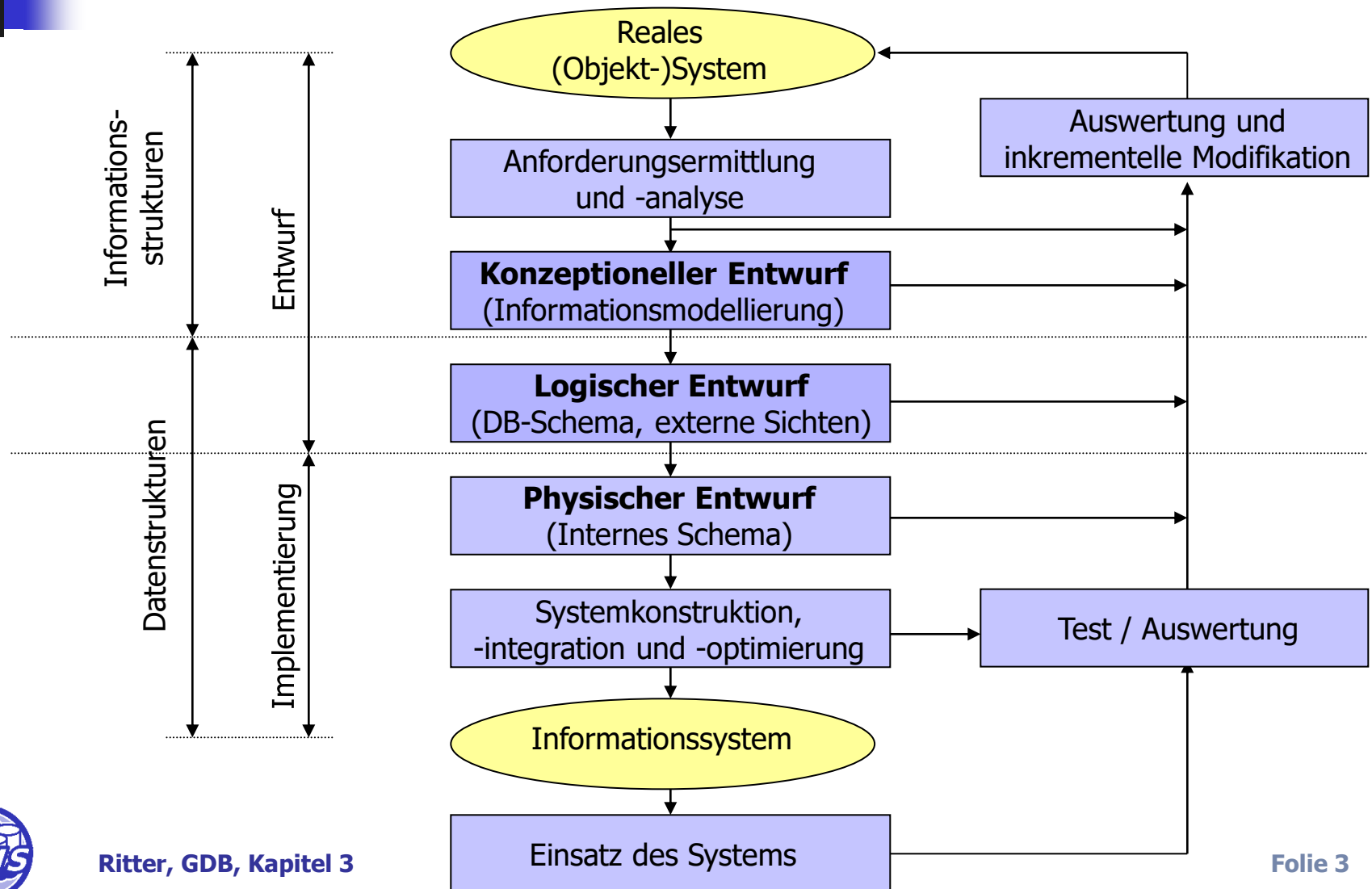




# DB-Entwurf und Modellierung (1)

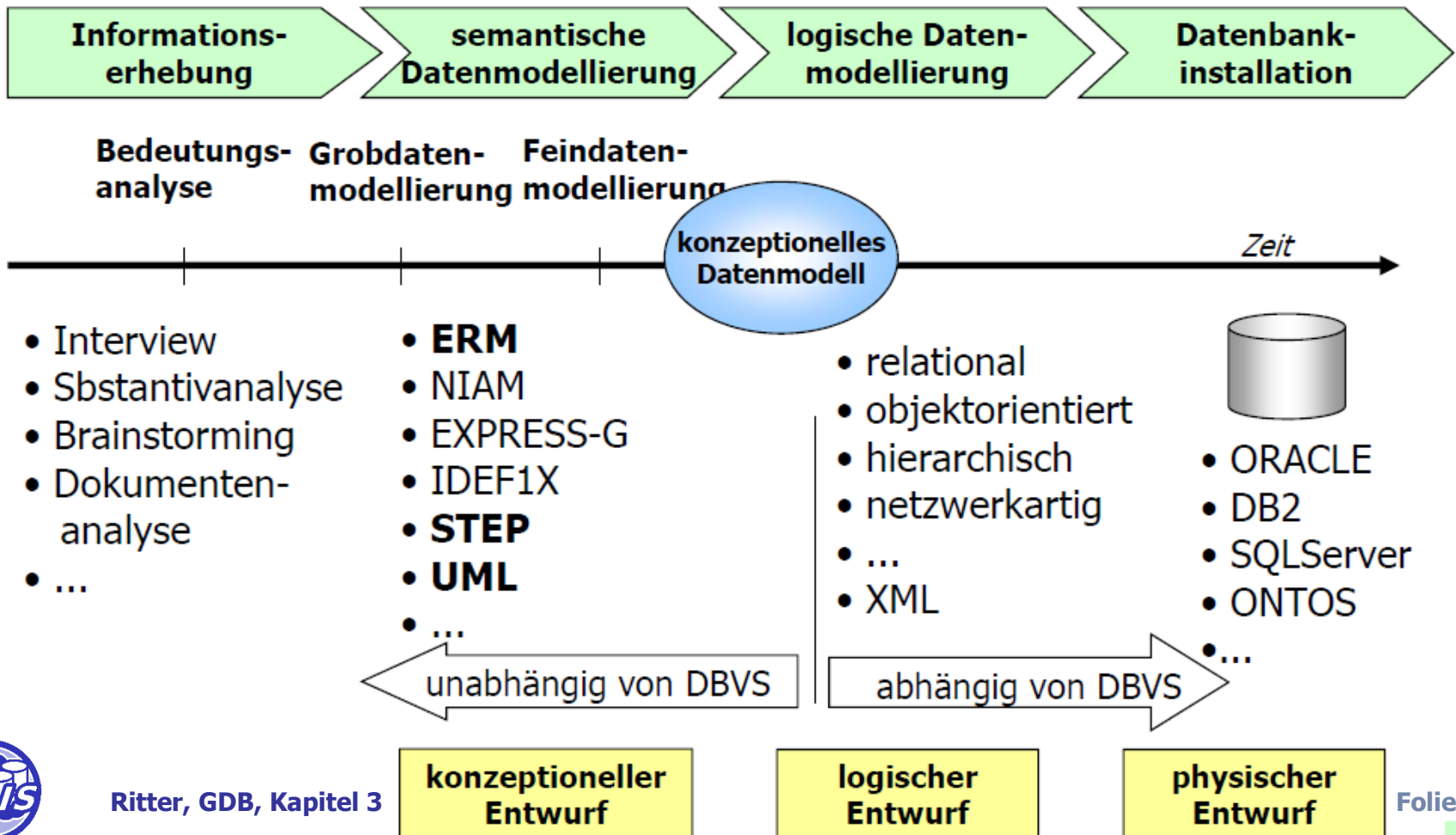
- **Ziel: Modellierung einer Miniwelt (Entwurf von DB-Schemata)**
  - modellhafte Abbildung eines anwendungsorientierten Ausschnitts der realen Welt (Miniwelt)
  - Nachbildung von Vorgängen durch **Transaktionen**
- **Nebenbedingungen:**
  - genaue Abbildung
  - hoher Grad an Aktualität
  - Verständlichkeit, Natürlichkeit, Einfachheit, ...
- **Zwischenziel:**
  - Erhebung der Information in der Systemanalyse (Informationsbedarf !)
  - **Informationsmodell** (allgem. Systemmodell)
- **Bestandteile:**
  - Objekte: Entities
  - Beziehungen: Relationships

# DB-Entwurf und Modellierung (2)

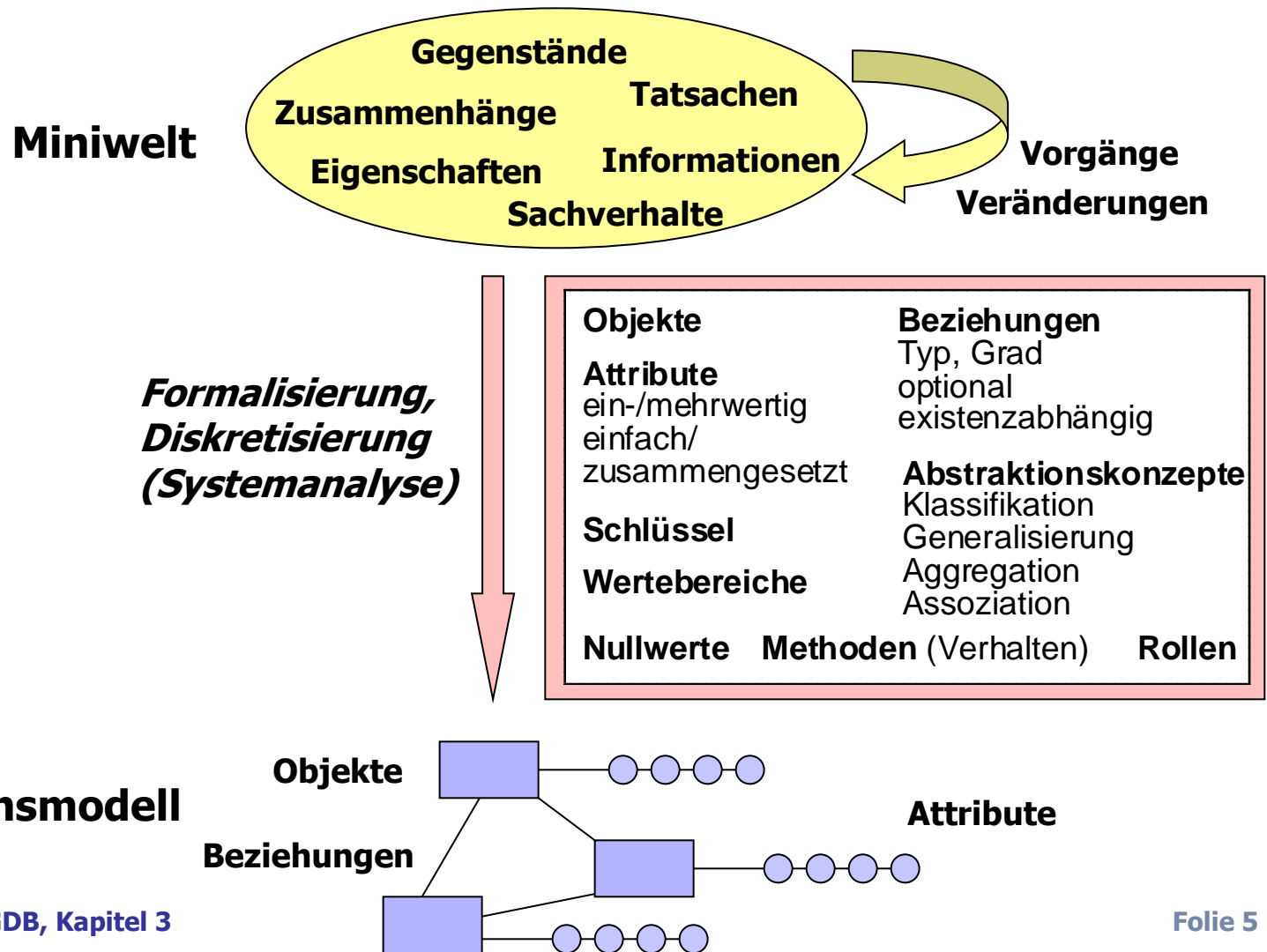


# DB-Entwurf und Modellierung (3)

## Prozessmodell:



# Miniwelt und Informationsmodell





# DB-Entwurf und Modellierung (4)

---

- **Informationsmodell**
  - Darstellungselemente & Regeln
  - eine Art formale Sprache, um Informationen zu beschreiben
- **Informationen über Objekte und Beziehungen nur, wenn:**
  - unterscheidbar und identifizierbar
  - relevant
  - selektiv beschreibbar

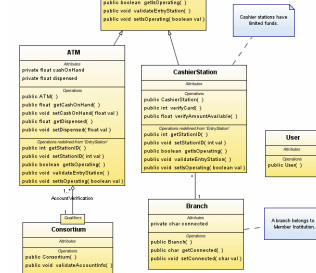
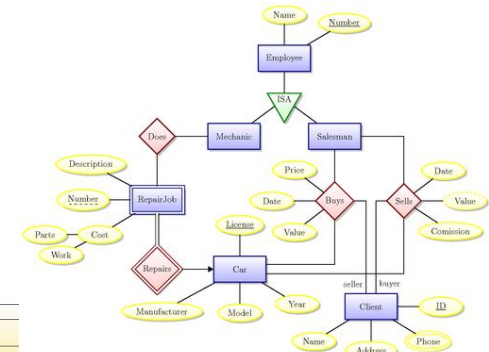
# Modellierungssprachen

## ■ ERM (Entity Relationship Model):

- generell einsetzbares Modellierungswerkzeug

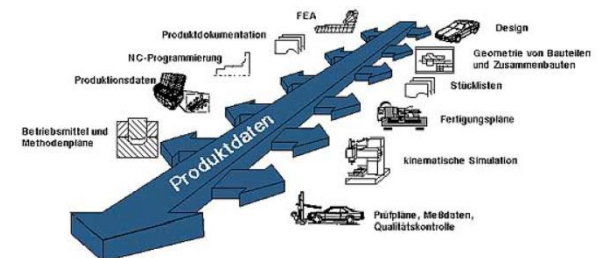
## ■ UML (Unified Modeling Language):

- Notation und Sprache zur Unterstützung objektorientierter Softwareentwicklung



## ■ STEP (STandard for the Exchange of Product Definition Data):

- Modellierung, Zugriff, Austausch von **produktdefinierenden Daten** über den gesamten Produktlebenszyklus





# Entity-Relationship-Modell (ERM) (1)

- **Modellierungskonzepte**

- Entity-Mengen (Objektmengen)
- Wertebereiche, Attribute
- Primärschlüssel
- Relationship-Mengen (Beziehungsmengen)

Chen, P. P.-S.: The Entity-Relationship Model  
- Toward a Unified View of Data,  
in: ACM TODS 1:1, March 1976, pp. 9-36.

- **Klassifikation der Beziehungstypen**

- benutzerdefinierte Beziehungen
- Abbildungstyp
  - 1 : 1
  - n : 1
  - n : m
- **Ziel:**
  - Festlegung von semantischen Aspekten
  - explizite Definition von strukturellen Integritätsbedingungen





# Entity-Relationship-Modell (2)

---

- **Beachte:**

- Das ERM modelliert die Typ-, nicht die Instanzenebene; es macht also Aussagen über Entity- und Relationship-Mengen, nicht jedoch über einzelne ihrer Elemente (Ausprägungen).
- Die Modellierungskonzepte des ERM sind häufig zu ungenau oder unvollständig. Sie müssen deshalb ergänzt werden durch Integritätsbedingungen (Constraints).



# Entity-Relationship-Modell (3)

## ■ **Entities**

- wohlunterscheidbare Dinge der Miniwelt (Diskurswelt)
- „A thing that has real or individual existence in reality or in mind“ (Webster)
- besitzen Eigenschaften, deren konkrete Ausprägungen als Werte bezeichnet werden

## ■ **Entity-Mengen (Entity-Sets)**

- Zusammenfassung von „ähnlichen“ oder „vergleichbaren“ Entities
- haben gemeinsame Eigenschaften
- Beispiele:
  - Abteilungen, Angestellte, Projekte, ...
  - Bücher, Autoren, Leser, ...
  - Studenten, Professoren, Vorlesungen, ...
  - Kunden, Vertreter, Wein, Behälter, ...



# Entity-Relationship-Modell (4)

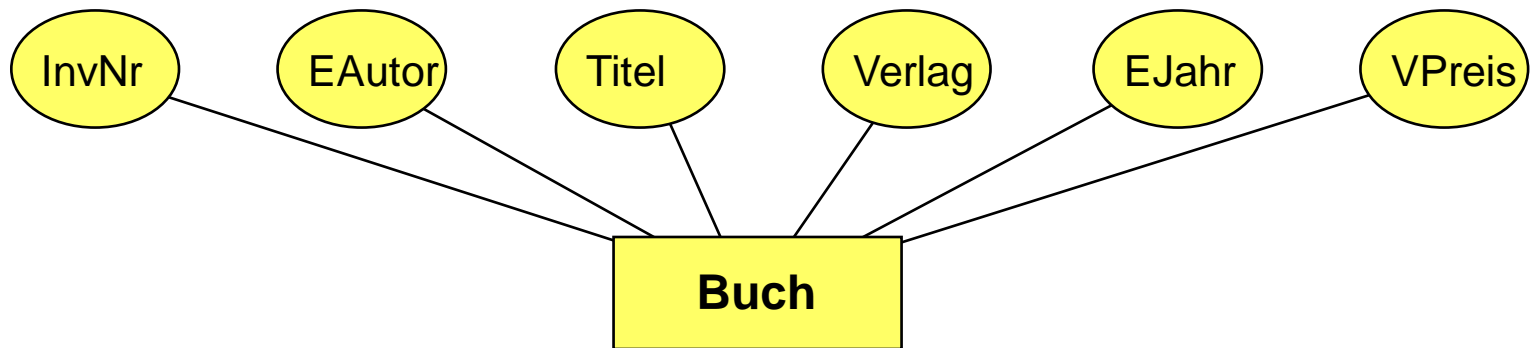
---

- **Wertebereiche und Attribute**

- Die möglichen oder „zulässigen“ Werte für eine Eigenschaft nennen wir Wertebereich (oder Domain)
- Die (bei allen Entities einer Entity-Menge auftretenden) Eigenschaften werden als Attribute bezeichnet
- Ein Attribut ordnet jedem Entity einer Entity-Menge einen Wert aus einem bestimmten Wertebereich (dem des Attributs) zu

# Entity-Relationship-Modell (5)

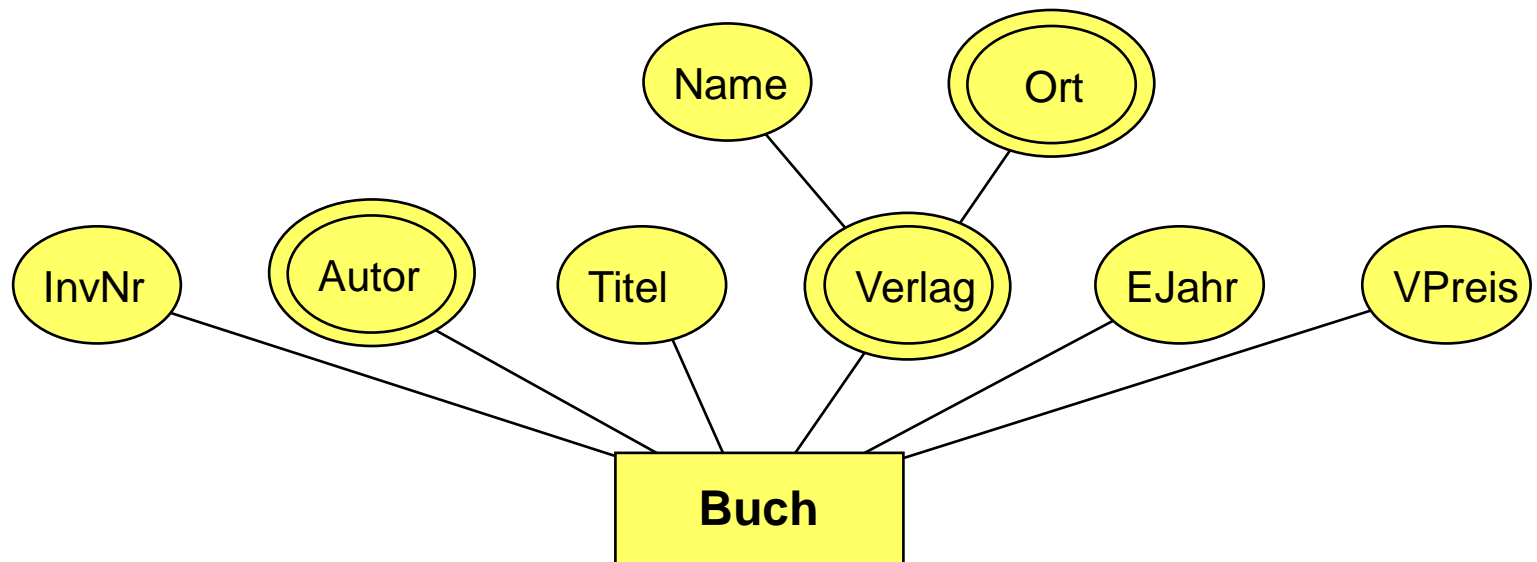
- **Beispiel: Entity-Typ „Buch“ (in Diagrammdarstellung)**



- jedem Attribut ist geeigneter Wertebereich zugeordnet
- Name der Entity-Menge sowie zugehörige Attribute sind **zeitinvariant**
- Entity-Menge und ihre Entities sind **zeitveränderlich**
  - e1 = (4711, Kemper, DBS, Oldenbourg, ...)
  - e2 = (0815, Date, Introd. To DBS, Addison, ...)
  - ...
  - e3 = (1234, Härder, DBS, Springer, ...)

# Entity-Relationship-Modell (6)

- **Erhöhung der Modellierungsgenauigkeit durch**
  - einwertige Attribute
  - mehrwertige Attribute (Doppelovale)
  - zusammengesetzte Attribute (hierarchisch angeordnete Ovale)
  - Verschachtelungen sind möglich



# Entity-Relationship-Modell (7)

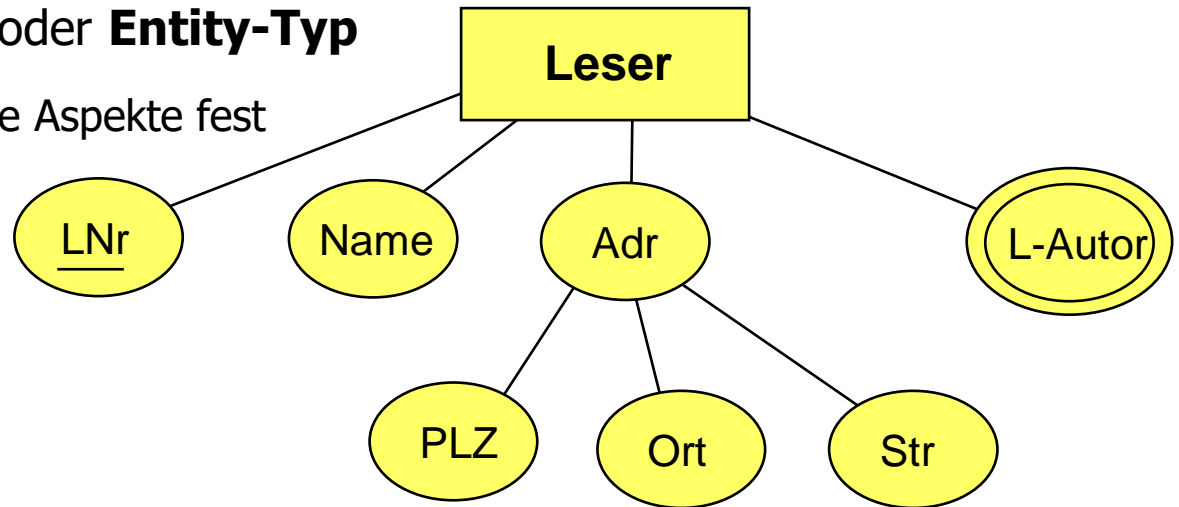
- **Wie wird ein Entity identifiziert?**
  - Entities müssen „wohlunterscheidbar“ sein
  - Information über ein Entity **ausschließlich** durch (Attribut-) Werte
- **Identifikation** eines Entities durch Attribut (oder Kombination von Attributen)
  - (1:1) - Beziehung
  - ggf. künstlich erzwungen (lfd. Nr.)
- $\{A_1, A_2, \dots, A_m\} = \mathbf{A}$  sei Menge der (einwertigen) Attribute zur Entity-Menge E
  - $\mathbf{K} \subseteq \mathbf{A}$  heißt **Schlüsselkandidat** von E
    - $\Leftrightarrow$   $\mathbf{K}$  irreduzibel (minimal) und
    - $e_i, e_j \in E: e_i \neq e_j \rightarrow \mathbf{K}(e_i) \neq \mathbf{K}(e_j)$
- mehrere Schlüsselkandidaten (SK) möglich  $\rightarrow$  **Primärschlüssel** auswählen
- **Beispiel:** Entity-Menge **Student** mit Attributen  
Matnr, SVNr, Name, Gebdat, FBNr

# Entity-Relationship-Modell (8)

- Entity-Deklaration oder **Entity-Typ**

- legt zeitinvariante Aspekte fest

- Entity-Typ**  
 $E = (X, K)$



- $\text{Leser} = ( \{ \text{LNr}, \text{Name}, \text{Adr} (\text{PLZ}, \text{Ort}, \text{Straße}), \{ \text{L-Autor} \} \}, \{ \text{LNr} \})$

- Wertebereiche**

- $W(\text{LNr}) = \text{int}(8), W(\text{Name}) = W(\text{L-Autor}) = \text{char}(30)$
- $W(\text{PLZ}) = \text{int}(5), W(\text{Ort}) = \text{char}(20), W(\text{Str}) = \text{char}(15)$
- $\text{dom}(\text{LNr}) = \text{int}(8)$
- $\text{dom}(\text{Adr}) = W(\text{PLZ}) \times W(\text{Ort}) \times W(\text{Str}) = \text{int}(5) \times \text{char}(20) \times \text{char}(15)$
- $\text{dom}(\text{L-Autor}) = 2^{W(\text{L-Autor})} = 2^{\text{char}(30)}$



# Entity-Relationship-Modell (9)

- **Definition *Entity-Typ***

- Ein Entity-Typ hat die Form  $E = (X, K)$  mit einem Namen  $E$ , einem Format  $X$  und einem Primärschlüssel  $K$ , der aus (einwertigen) Elementen von  $X$  besteht.

Die Elemente eines Formats  $X$  werden dabei wie folgt beschrieben:

- i) Einwertige Attribute :  $A$
- ii) Mehrwertige Attribute:  $\{A\}$
- iii) Zusammengesetzte Attribute:  $A (B_1, \dots, B_k)$

Definitionen aus:

G. Vossen: Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme, Oldenbourg.





# Entity-Relationship-Modell (10)

- **Definition *Wertebereich/Domain***

- $E = (X, K)$  sei ein Entity-Typ und  $\text{attr}(E)$  die Menge aller in  $X$  vorkommenden Attributnamen. Jedem  $A \in \text{attr}(E)$ , das nicht einer Zusammensetzung voransteht, sei ein Wertebereich  $W(A)$  zugeordnet. Für jedes  $A \in \text{attr}(E)$  sei  
 $\text{dom}(A) := W(A)$ , falls  $A$  einwertig;  
 $\text{dom}(A) := 2^{W(A)}$ , falls  $A$  mehrwertig;  
 $\text{dom}(A) := W(B_1) \times \dots \times W(B_k)$ ,  
falls  $A$  aus einwertigen  $B_1, \dots, B_k$  zusammengesetzt.  
Besteht  $A$  aus mehrwertigen oder zusammengesetzten Attributen, wird die Definition rekursiv angewendet.



# Entity-Relationship-Modell (11)

- **Definition *Entity* und *Entity-Menge***

- Es sei  $E = (X, K)$  ein Entity-Typ mit  $X = (A_1, \dots, A_m)$ .  
 $A_i$  sei  $\text{dom}(A_i)$  ( $1 \leq i \leq m$ ) zugeordnet.
  - Ein Entity  $e$  ist ein Element des Kartesischen Produkts aller Domains, d.h.  
 $e \in \text{dom}(A_1) \times \dots \times \text{dom}(A_m)$
  - Eine Entity-Menge  $E^t$  (zum Zeitpunkt  $t$ ) ist eine Menge von Entities, welche  $K$  erfüllt, d.h.  
 $E^t \subseteq \text{dom}(A_1) \times \dots \times \text{dom}(A_m)$   
 $E^t$  wird auch als der Inhalt bzw. der aktuelle Wert (Instanz) des Typs  $E$  zur Zeit  $t$  bezeichnet.



# Entity-Relationship-Modell (12)

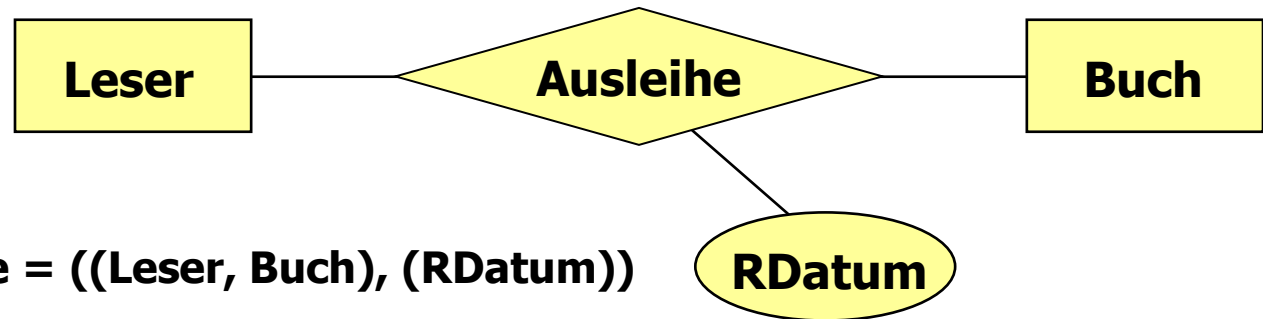
- **Definition *Relationship*, *Relationship-Typ* und *Relationship-Menge***
  - Ein Relationship-Typ hat die Form  $R = (\text{Ent}, Y)$ . Dabei ist  $R$  der Name des Typs,  $\text{Ent}$  bezeichnet die Folge der Namen der Entity-Typen, zwischen denen die Beziehung definiert ist, und  $Y$  ist eine (möglicherweise leere) Folge von Attributen der Beziehung.
  - Sei  $\text{Ent} = (E_1, \dots, E_k)$ , und für beliebiges, aber festes  $t$  sei  $E_i^t$  der Inhalt des Entity-Typs  $E_i$ ,  $1 \leq i \leq k$ . Ferner sei  $Y = (B_1, \dots, B_n)$ . Eine Relationship  $r$  ist ein Element des Kartesischen Produktes aus allen  $E_i^t$  und den Domains der  $B_j$ , d.h.  
$$r \in E_1^t \times \dots \times E_k^t \times \text{dom}(B_1) \times \dots \times \text{dom}(B_n) \text{ bzw.}$$
$$r = (e_1, \dots, e_k, b_1, \dots, b_n) \text{ mit } (e_1, \dots, e_k) \text{ eindeutig und}$$
$$e_i \in E_i^t \text{ für } 1 \leq i \leq k \text{ und } b_j \in \text{dom}(B_j) \text{ für } 1 \leq j \leq n.$$
  - Eine Relationship-Menge  $R^t$  (zur Zeit  $t$ ) ist eine Menge von Relationships, d.h.,  
$$R^t \subseteq E_1^t \times \dots \times E_k^t \times \text{dom}(B_1) \times \dots \times \text{dom}(B_n).$$

# Entity-Relationship-Modell (13)

- **Eigenschaften von Relationship-Mengen**

- Grad  $n$  der Beziehung (*degree*), gewöhnlich  $n=2$  oder  $n=3$
- Existenzabhängigkeit
- Beziehungstyp (*connectivity*)
- Kardinalität

- Beispiele



**Ausleihe = ((Leser, Buch), (RDatum))**

**Eigenschaften:**

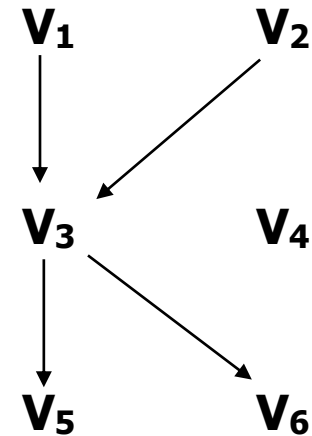
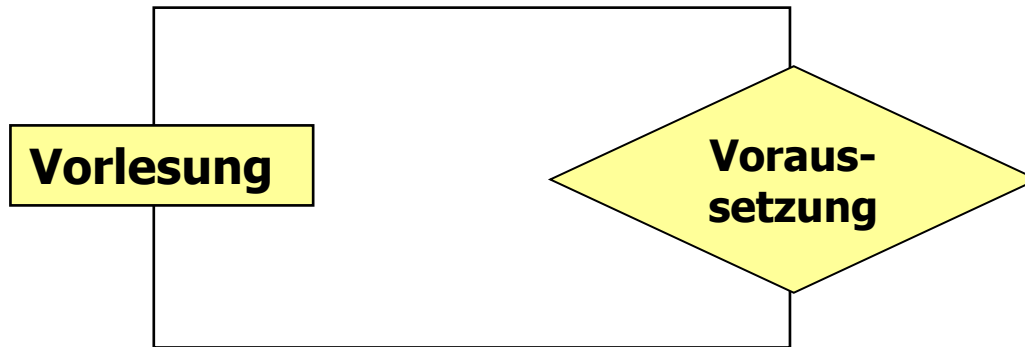
**Grad: 2**

**Existenzabhängigkeit: Nein**

**Beziehungstyp: n:m**

# Entity-Relationship-Modell (14)

## ■ Beispiele (Forts.)



**Voraussetzung** = ((**Vorgänger**/Vorlesung, **Nachfolger**/Vorlesung), ( $\emptyset$ ))

genauer: direkte Voraussetzung

Eigenschaften:

**Grad: 1**

**Existenzabhängigkeit: Nein**

**Beziehungstyp: n:m**

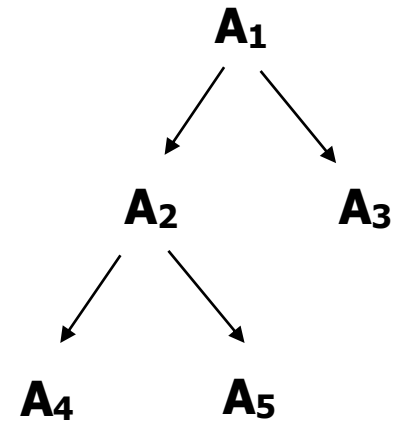
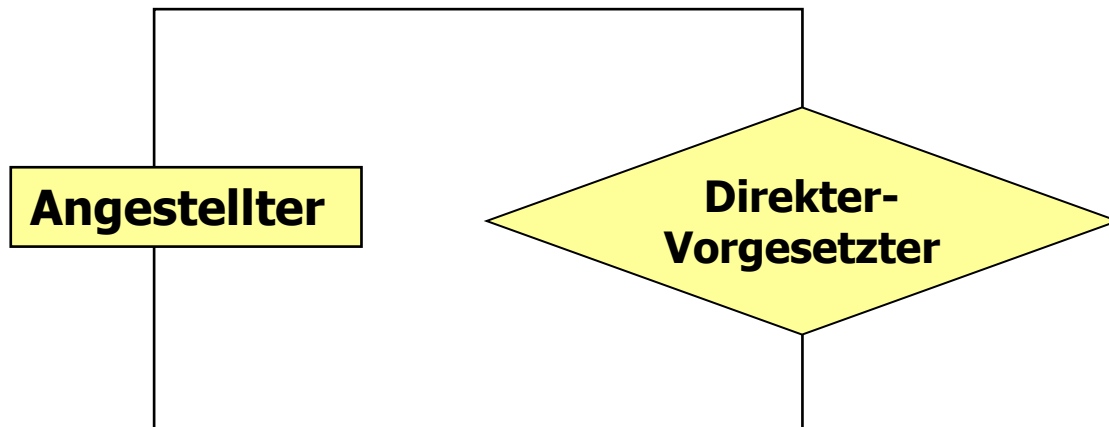
**Rollennamen**

Transitivität gilt im Allg. bei Selbstreferenz nicht (Beispiel: „liebt“ auf „Person“).

Keine Disjunktheit der an einer Relationship-Menge beteiligten Entity-Mengen gefordert.

# Entity-Relationship-Modell (15)

- Beispiele (Forts.)



**Direkter Vorgesetzter =**

**((Angestellter/Angestellter, Chef/Angestellter), (∅))**

**Eigenschaften:**

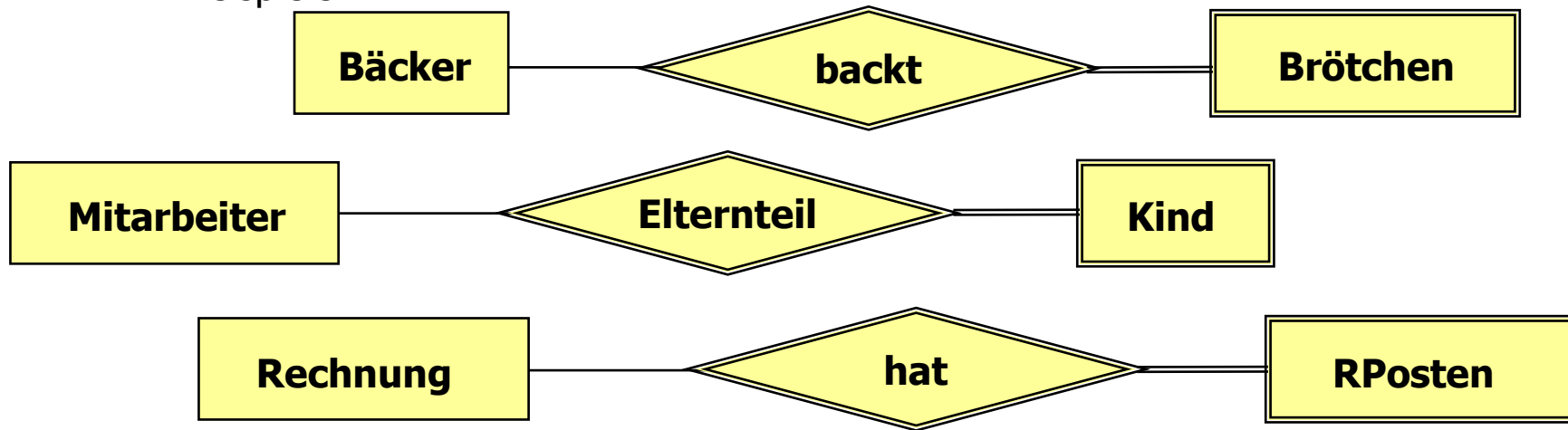
**Grad: 1**

**Existenzabhängigkeit: Nein**

**Beziehungstyp: 1:n**

# Entity-Relationship-Modell (16)

- Existenzabhängigkeit von Entity-Mengen
  - Existenzabhängigkeit: *Relationship begründet Existenz von*
  - Beispiele

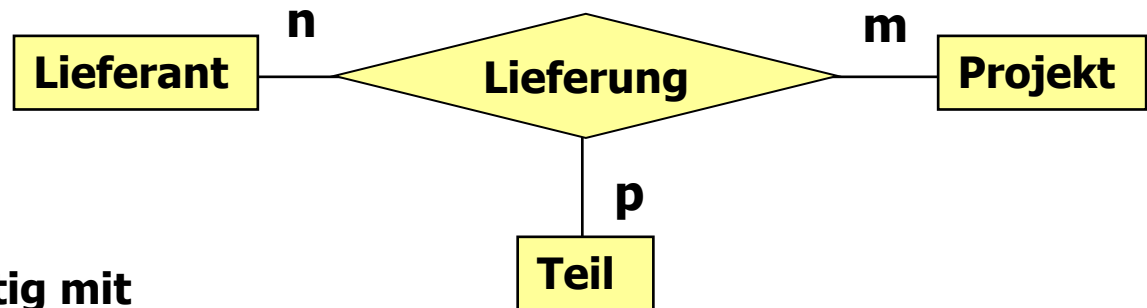


- **Eigenschaften**
  - Grad: 2
  - Existenzabhängig: ja
  - Beziehungstyp: 1 : n

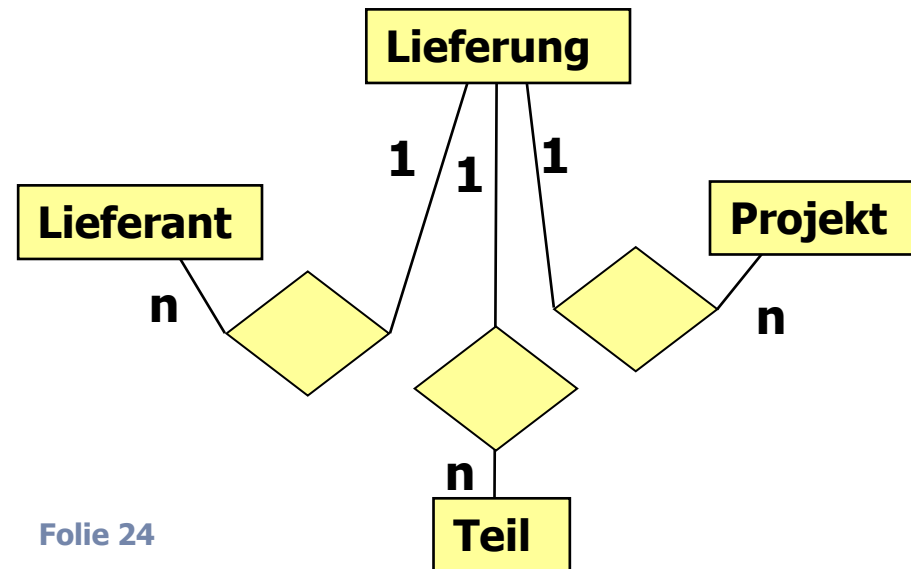
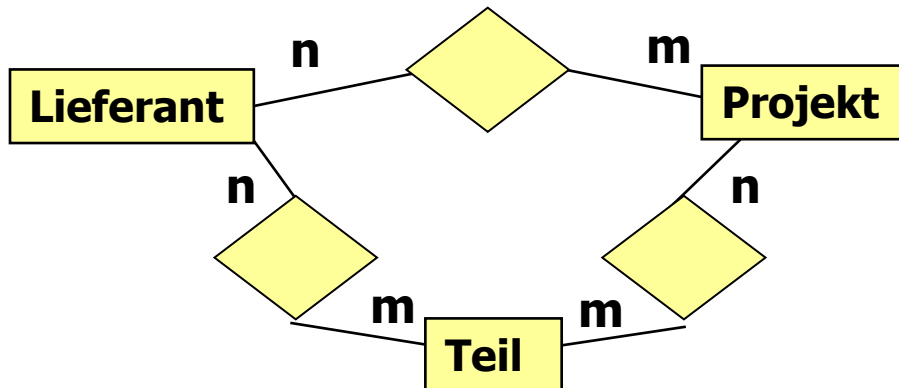
**Bem.:** Bei Mehrfachreferenzen ist eine „erzeugende“ von weiteren „referenzierenden“ Relationship-Mengen zu unterscheiden.

# Entity-Relationship-Modell (17)

- 3-stellige Relationship-Mengen



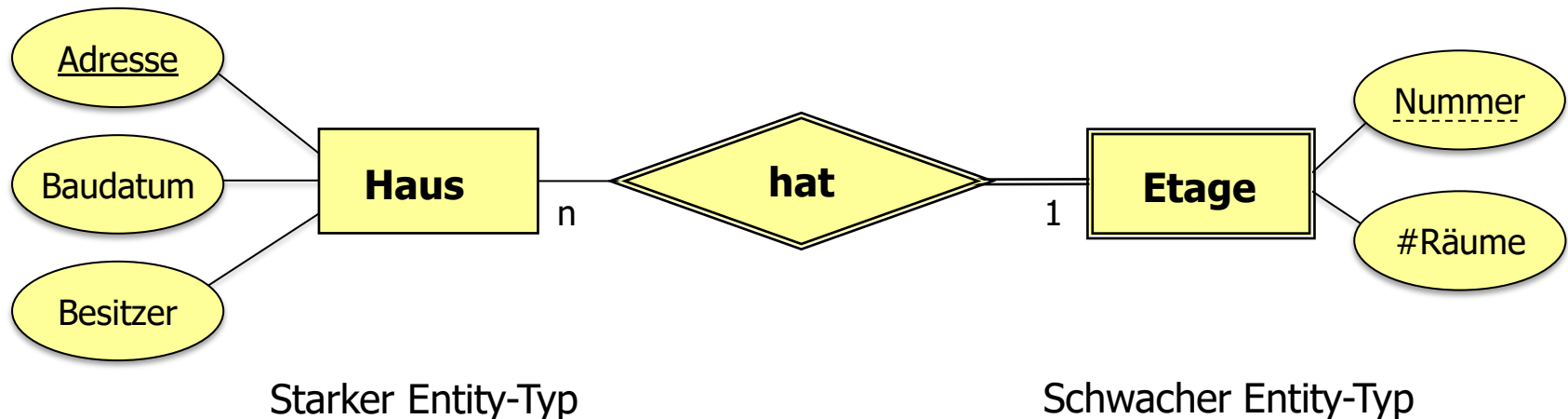
**ACHTUNG:** nicht gleichwertig mit





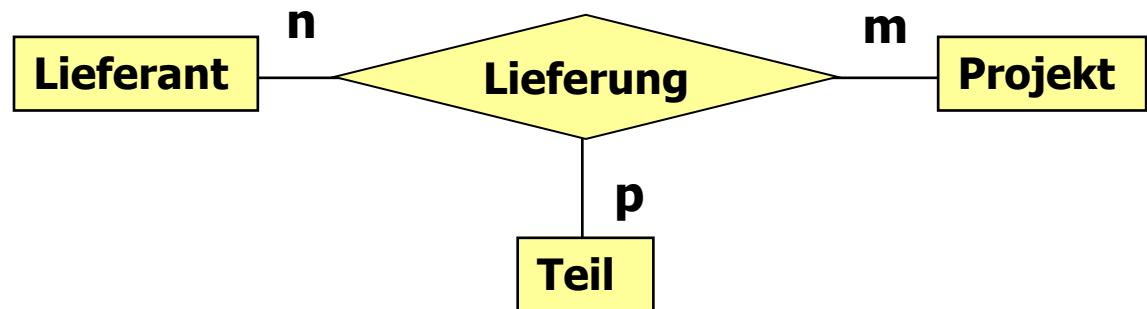
# Entity-Relationship-Modell (17)

- Existenzabhängigkeiten zwischen Entity-Typen
  - Starke Entity-Typen vs. Schwache Entity-Typen
  - Starker Primärschlüssel vs. Schwacher Primärschlüssel
  - Beispiel:

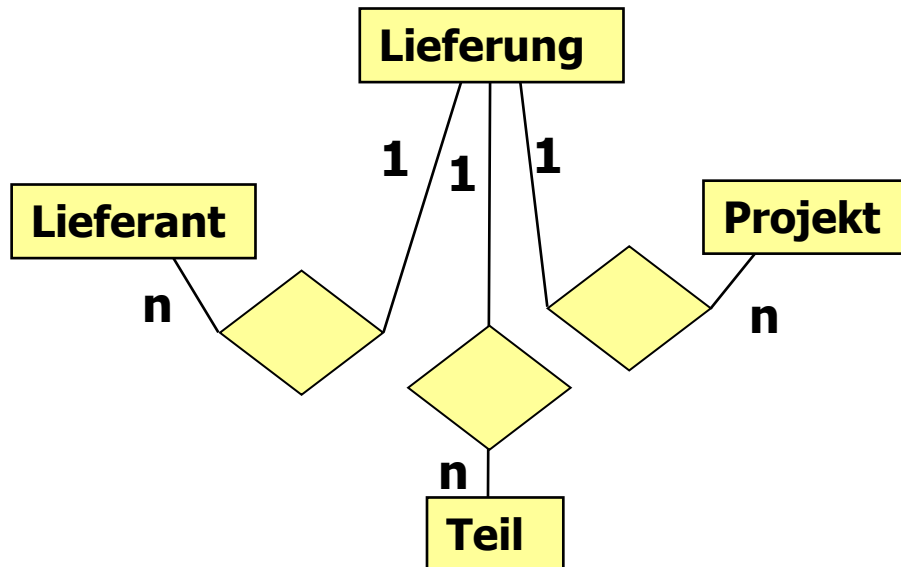


# Entity-Relationship-Modell (18)

- 3-stellige Relationship-Mengen (Grad  $k=3$ )

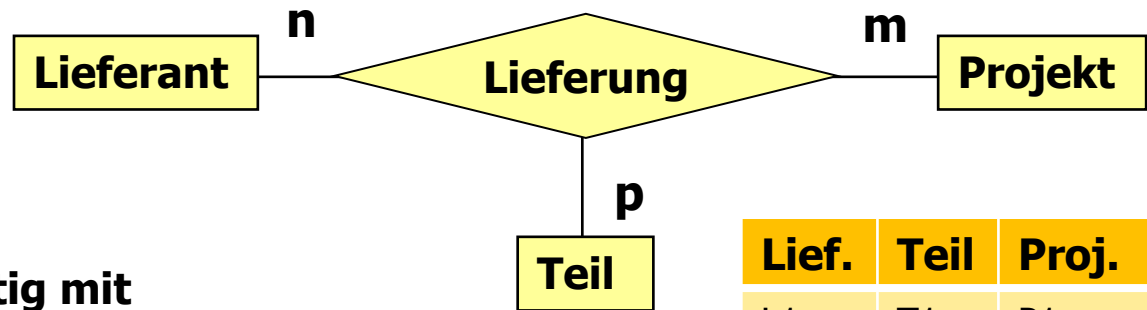


(fast) gleichwertig mit

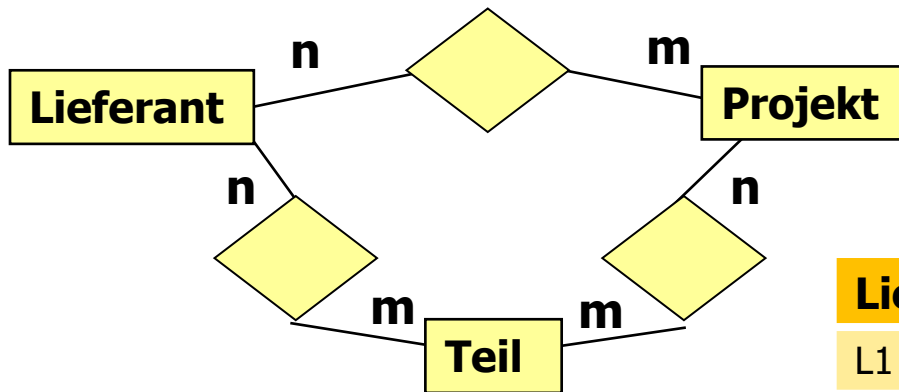


# Entity-Relationship-Modell (19)

- 3-stellige Relationship-Mengen (Grad  $k=3$ )



**ACHTUNG:** nicht gleichwertig mit



Lief.	Teil	Proj.
L1	T1	P1
L1	T2	P2
L2	T1	P2
L2	T2	P1

Lief.	Teil
L1	T1
L1	T2
L2	T1
L2	T2

Lief.	Proj.
L1	P1
L1	P2
L2	P1
L2	P2

Teil	Proj.
T1	P1
T1	P2
T2	P1
T2	P2



# Entity-Relationship-Modell (18)

- **Klassifikation von Datenabbildungen**
  - **ZIEL:**
    - Festlegung von semantischen Aspekten (hier: Beziehungstyp)
    - explizite Definition von strukturellen Integritätsbedingungen
  - **Unterscheidung von Beziehungstypen**
    - $E_i - E_j$
    - $E_i - E_i$
  - **Festlegung der Abbildungstypen**
    - 1:1 ... eindeutige Funktion (injektive Abbildung)
    - n:1 ... math. Funktion (funktionale oder invers funktionale Abbildung)
    - n:m ... math. Relation (komplexe Abbildung)
  - Abbildungstypen implizieren nicht, dass für jedes  $e_k \in E_i$  auch tatsächlich ein  $e_l \in E_j$  existiert

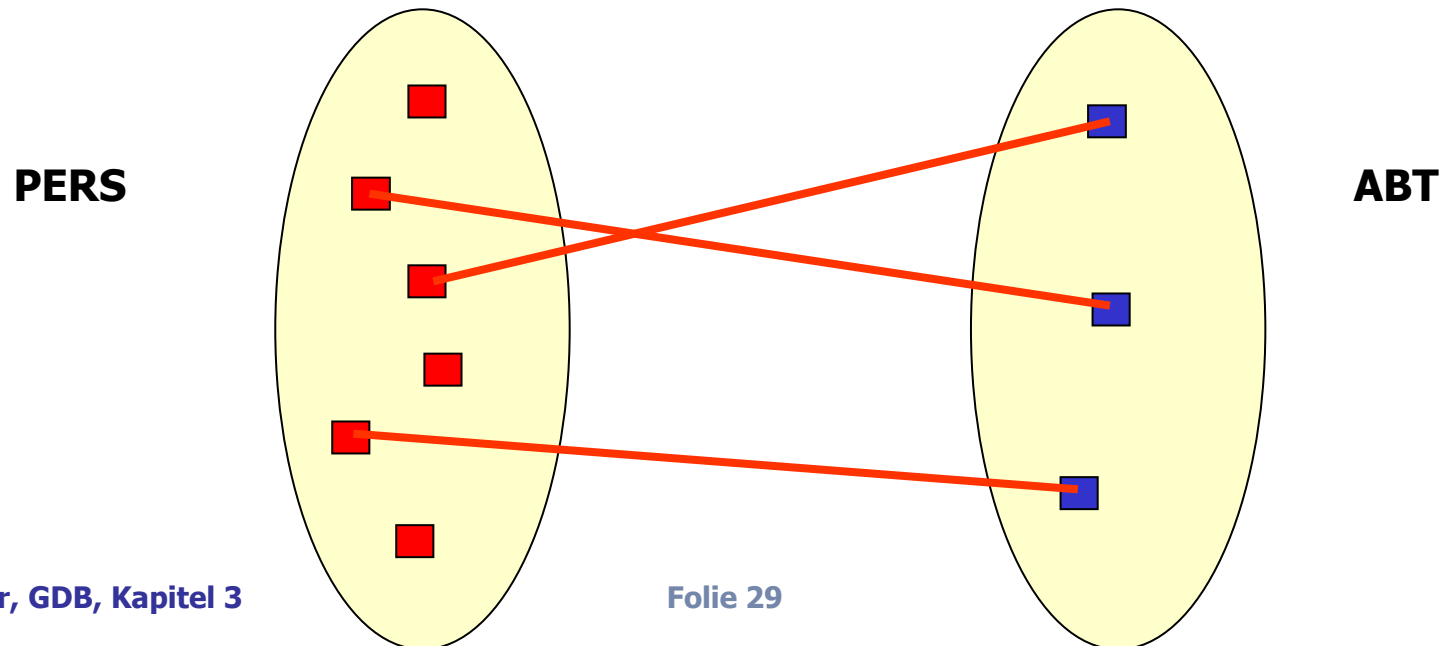
# Entity-Relationship-Modell (19)

- **Klassifikation von Datenabbildungen (Forts.)**

- **Beispiele**



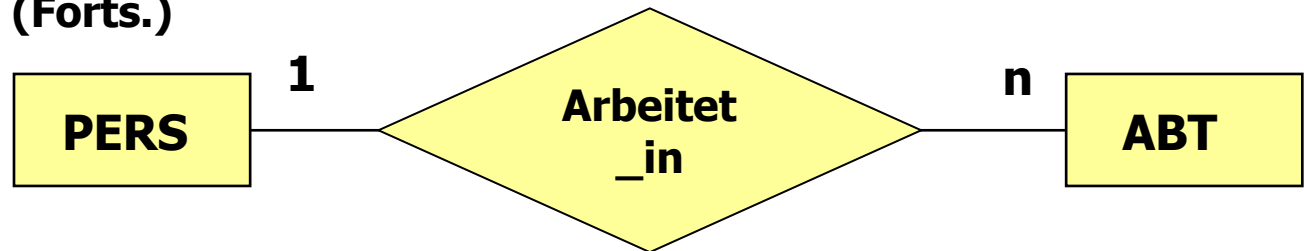
**Leitet / Wird\_geleitet\_von (1:1)**



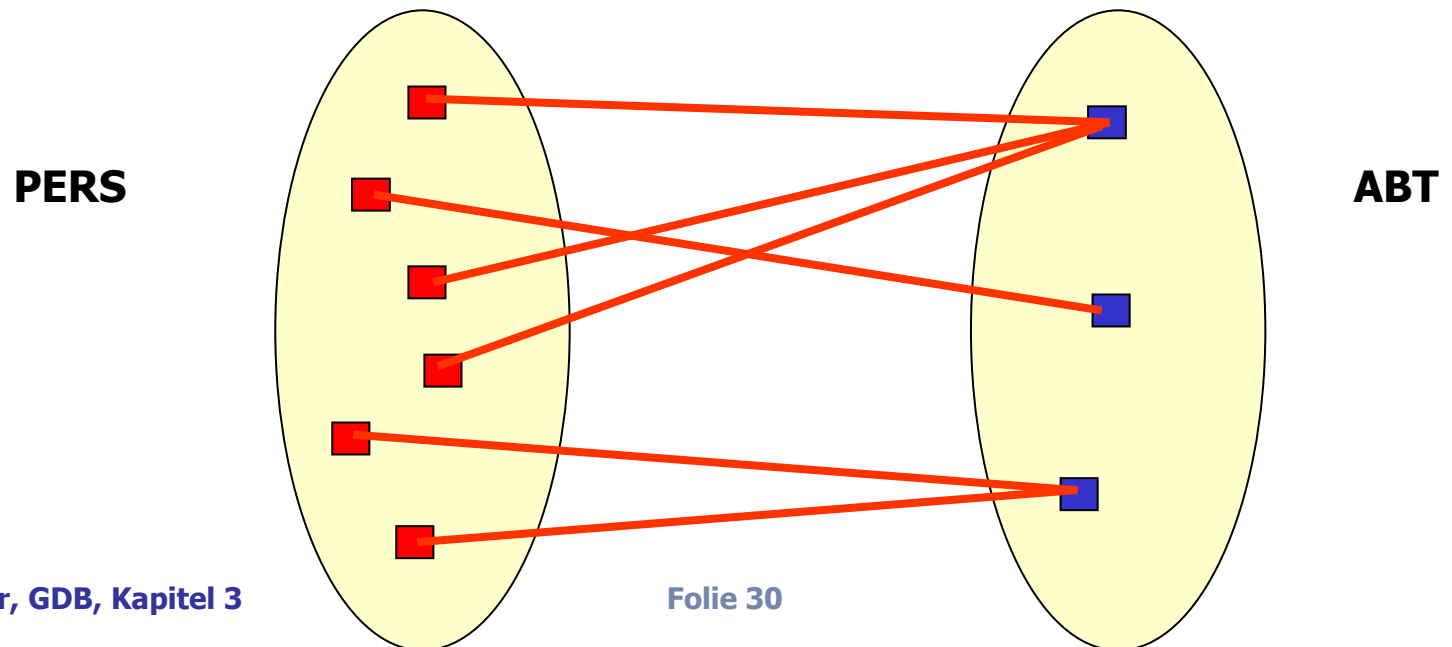
# Entity-Relationship-Modell (20)

- **Klassifikation von Datenabbildungen (Forts.)**

- **Beispiele (Forts.)**



**Arbeitet\_in / Hat\_Mitarbeiter (n:1)**



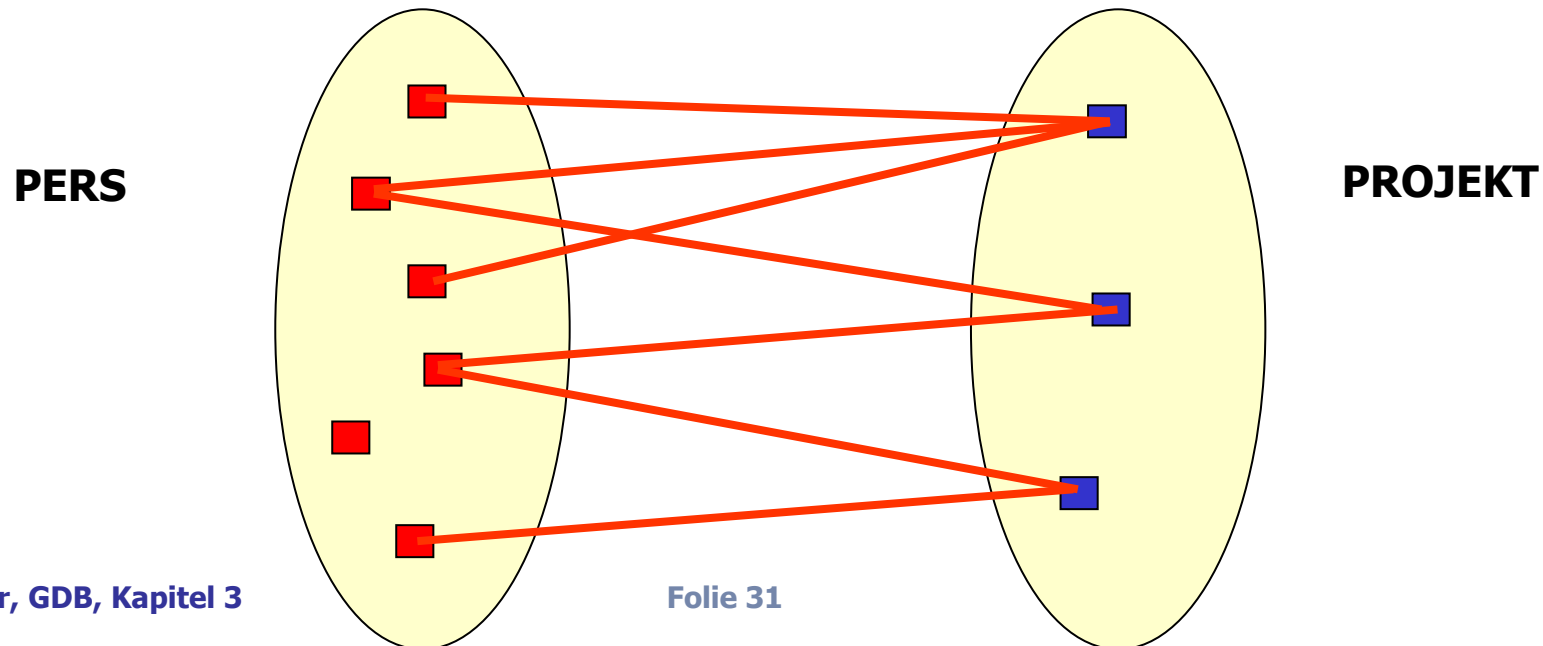
# Entity-Relationship-Modell (21)

- **Klassifikation von Datenabbildungen (Forts.)**

- **Beispiele (Forts.)**



**Arbeitet\_in / Mitarbeit (n:m)**





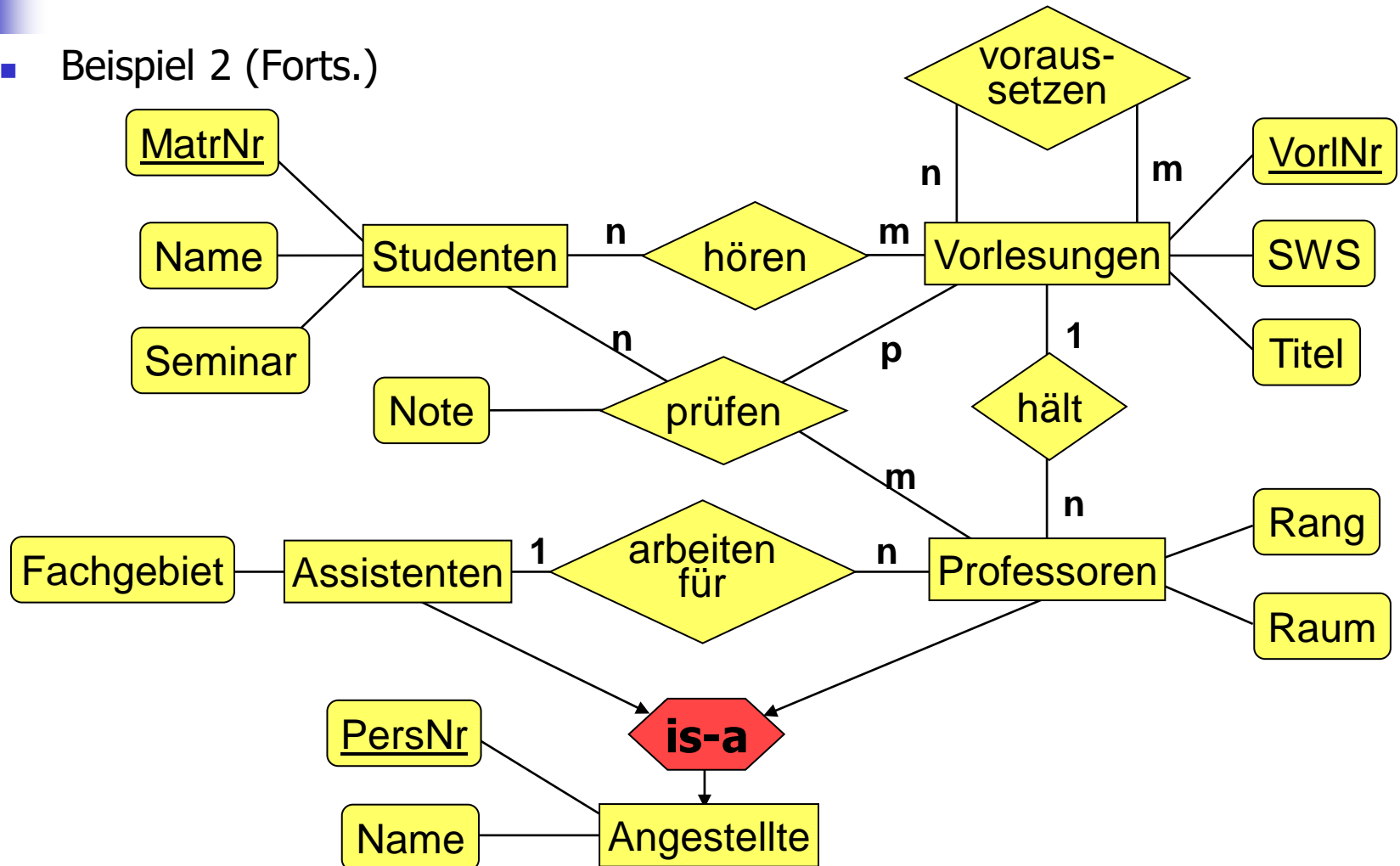
# Entity-Relationship-Modell (22)

- Beispiel 2: „Vorlesungsbetrieb“
  - Stellen Sie ein ER-Diagramm für folgende Miniwelt auf:
    - Jeder Professor **hält** mehrere seiner Vorlesungen und **prüft** Studenten jeweils über eine dieser Vorlesungen.
    - Mehrere Assistenten **arbeiten** jeweils für genau einen Professor.
    - Mehrere Studenten **hören** jeweils eine Reihe von Vorlesungen. Vorlesungen werden jeweils von mehreren Studenten **besucht**.
    - Der Besuch von Vorlesungen **setzt** i. allg. die Kenntnis anderer Vorlesungen **voraus**.
    - Sowohl Professoren als auch Assistenten sind Angestellte.



# Entity-Relationship-Modell (23)

- Beispiel 2 (Forts.)





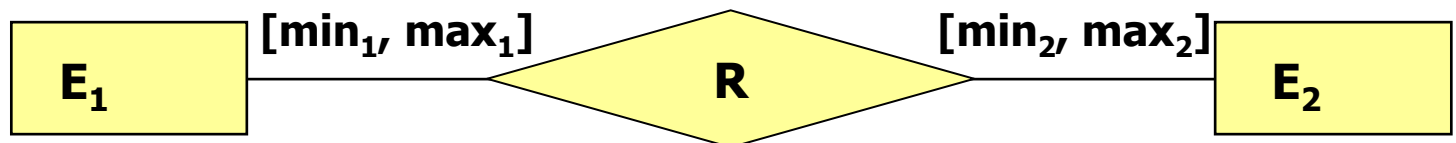
# Erweiterungen des ERM

---

- Ziel: Genauere Modellierung von Beziehungen
  - Verfeinerung der Abbildungen von Beziehungen durch **Kardinalitätsrestriktionen**
  - Generalisierung und Vererbung
  - Aggregationen
  - Einführung von systemkontrollierten Ableitungen (**Reasoning**)

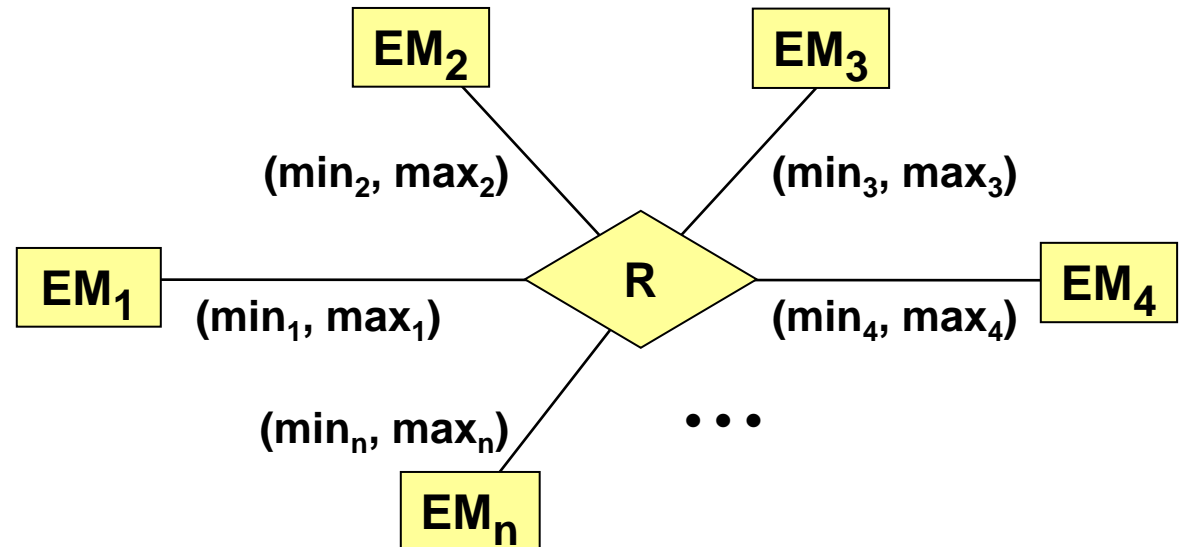
# Kardinalitätsrestriktionen (1)

- Verfeinerung der Datenabbildung
  - bisher: grobe strukturelle Festlegung der Beziehungen  
z. B.: 1:1 bedeutet „höchstens eins zu höchstens eins“
  - Verfeinerung der Semantik eines Beziehungstyps durch Kardinalitätsrestriktionen:  
sei  $R \subseteq E_1 \times E_2 \times \dots \times E_n$ ;  
Kardinalitätsrestriktion  $\text{kard}(R, E_i) = [\min, \max]$   
bedeutet, dass jedes Element aus  $E_i$  in wenigstens  $\min$  und höchstens  $\max$  Ausprägungen von  $R$  enthalten sein muss (mit  $0 \leq \min \leq \max$ ,  $\max \geq 1$ )
  - Grafische Darstellung
    - $e_1$  nimmt an  $[\min_1, \max_1]$  Beziehungen vom Typ  $R$  teil
    - $e_2$  nimmt an  $[\min_2, \max_2]$  Beziehungen vom Typ  $R$  teil



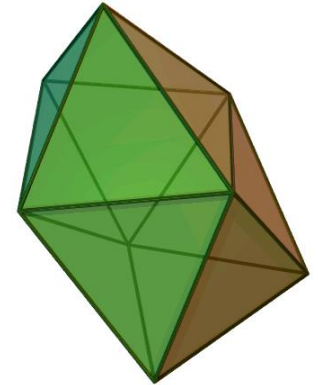
# Kardinalitätsrestriktionen (2)

- Verfeinerung der Datenabbildung (Forts.)
  - Verallgemeinerung:



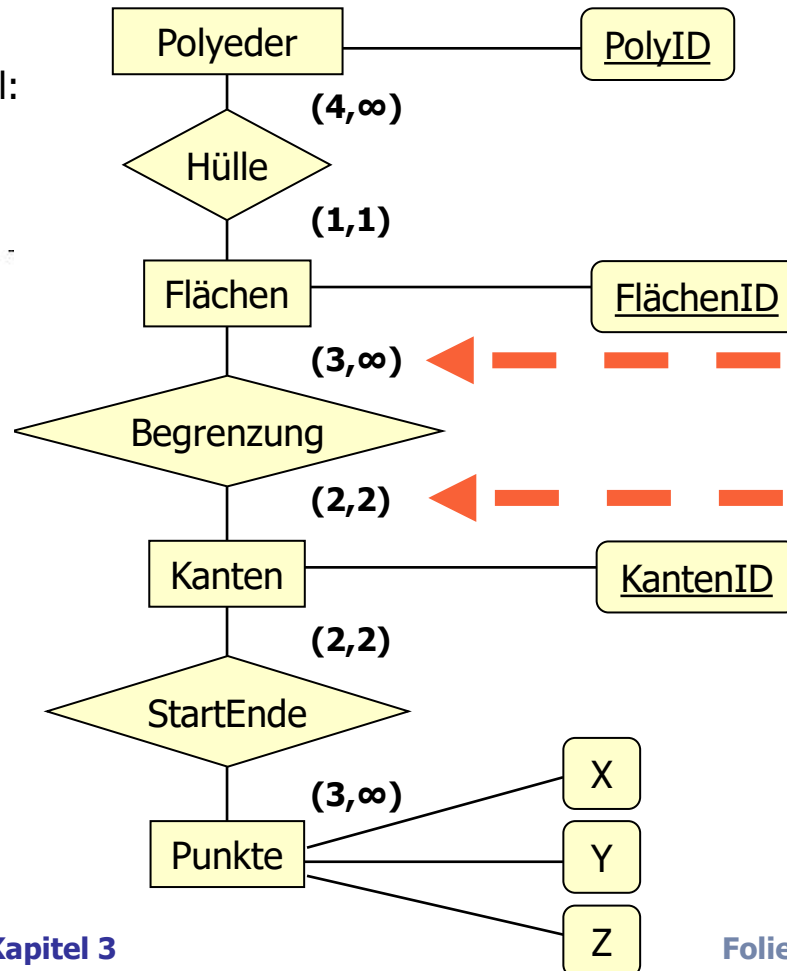
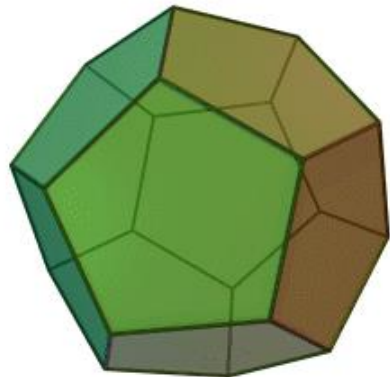
Für jedes  $e_i$  aus  $EM_i$  gibt es (in  $R$ )

- mindestens  $\min_i$  Tupel der Art  $(\dots, e_i, \dots)$
- höchstens  $\max_i$  Tupel der Art  $(\dots, e_i, \dots)$



## ■ Verfeinerung der Datenabbildung (Forts.)

■ Beispiel:

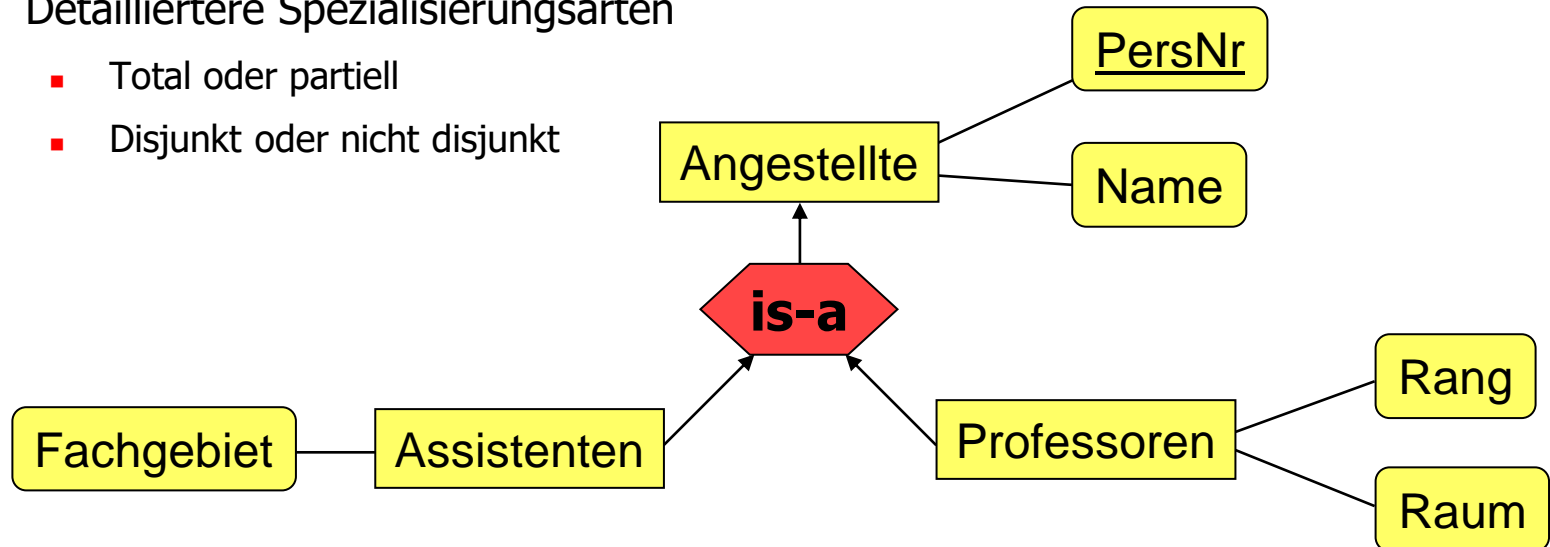


Jede Fläche wird von mindestens 3 Kanten begrenzt

Jede Kante begrenzt genau 2 Flächen

# Generalisierung und Vererbung

- **IS-A-Beziehung:** Obermenge vs. Untermenge (Abstraktion auf Typebene)
- Generalisierung (Untermenge → Obermenge):  
Zusammenfassen gleicher Eigenschaften mehrer Entity-Mengen
- Spezialisierung (Obermenge → Untermenge):  
Aufteilen einer Entity-Menge in detaillierter beschriebene Teilmengen
- Vererbung von Attributen, Schlüsseln und Beziehungen
- Detailliertere Spezialisierungsarten
  - Total oder partiell
  - Disjunkt oder nicht disjunkt

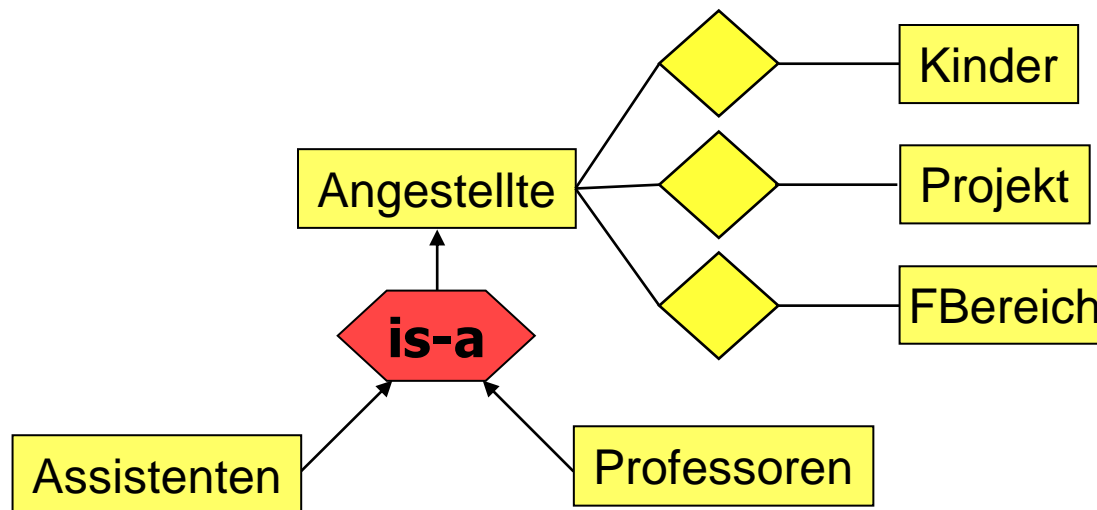


# Generalisierung und Vererbung

## ■ Vererbung von Beziehungen

### Beispiel:

- Professoren/Assistenten können Kinder haben
- Professoren/Assistenten können an Projekten teilnehmen
- Professoren/Assistenten können einem Fachbereich angehören

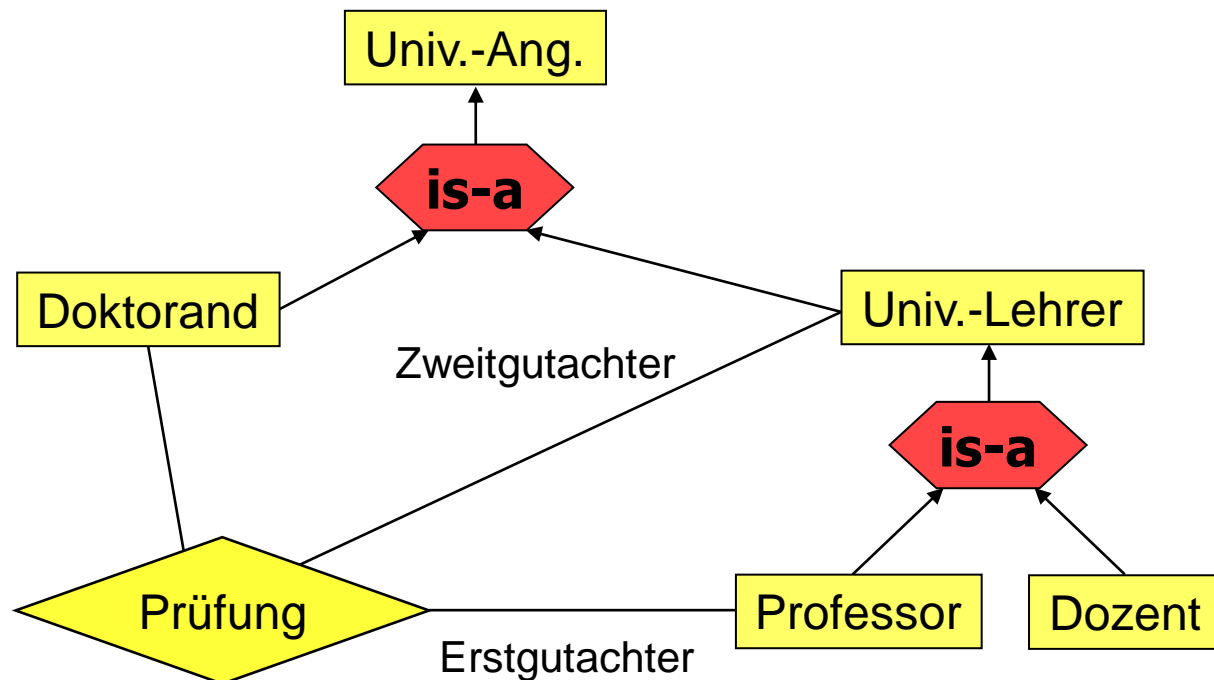


# Generalisierung und Vererbung

## ■ Vererbung von Rollen

### Beispiel:

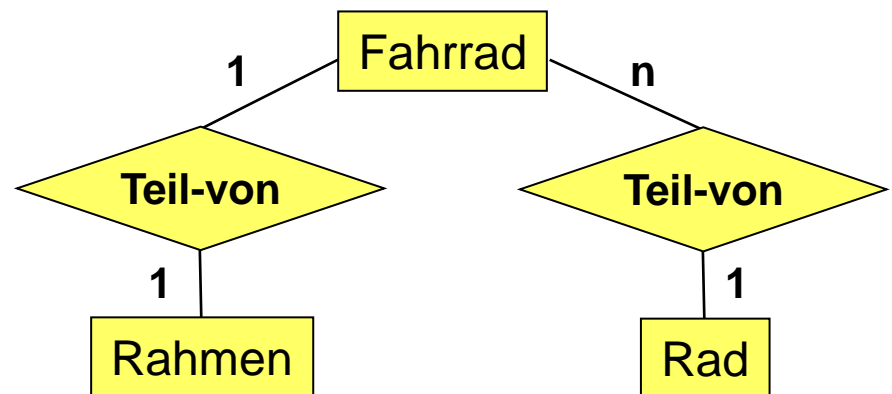
- Professoren und Dozenten können als Zweitgutachter fungieren





# Aggregationen

- **Teil-von**-Beziehung: Strukturierung unterschiedlicher Entity-Mengen
- Objekte der untergeordneten Entity-Menge sind Teile der Objekte der übergeordneten Entity-Menge
- Mögliche Abbildungstypen:
  - 1:1
  - n:1
- Häufige Verwendung von Existenzabhängigkeiten
- Aggregationshierarchien





# Zusammenfassung (1)

---

- **DB-Entwurf umfasst**

- Informationsbedarfsanalyse
- konzeptionelles DB-Schema (-> Informationsmodell)
- logisches DB-Schema (nicht diskutiert)
- physisches DB-Schema (nicht diskutiert)

- **ERM-Charakteristika**

- Modellierung bezieht sich auf die Typebene
- Relevante Zusammenhänge der Miniwelt werden durch Entity- und Relationship-Mengen modelliert; sie werden genauer durch Attribute, Wertebereiche, Primärschlüssel/Schlüsselkandidaten beschrieben
- Klassifikation von Beziehungstypen dient der Spezifikation von strukturellen Integritätsbedingungen
- anschauliche Entwurfsdarstellung durch ER-Diagramme
- relativ karges Informationsmodell



# Zusammenfassung (2)

---

- **Einführung weiterer Modellierungskonzepte**
  - Verfeinerung von Beziehungen durch Kardinalitätsrestriktionen
  - Generalisierung und Vererbung
  - Aggregation, implizierte Attribute und Beziehungen
  - Das erweiterte ERM ist sehr mächtig und umfasst viele bekannte Modellierungskonzepte
  - Integritätsbedingungen wurden hier nicht behandelt (-> Relationenmodell)