

## Grundlagen von Datenbankmodellen

### Begriff Datenbankmodell

*Ein Datenbankmodell ist ein System von Konzepten zur Beschreibung von Datenbanken. Es legt Syntax und Semantik von Datenbankbeschreibungen für ein Datenbanksystem fest.*

Datenbankbeschreibungen = Datenbankschemata

## Datenbankmodelle II

Klassische Datenbankmodelle sind speziell geeignet für

- große Informationsmengen mit relativ starrer Struktur und
- die Darstellung statischer Eigenschaften und Integritätsbedingungen (also die Bereiche 1(a), 1(b) und 3(a)).

## 3. Datenbankmodelle für den Entwurf

- ➡ Grundlagen von Datenbankmodellen
- ➡ Entity-Relationship-Modelle
- ➡ Objektorientierte Modelle: UML

### Datenbankmodelle I

Ein Datenbankmodell legt fest

#### 1. *statische Eigenschaften*

- (a) Objekte
- (b) Beziehungen

inklusive der Standard-Datentypen, die Daten über die Beziehungen und Objekte darstellen können,

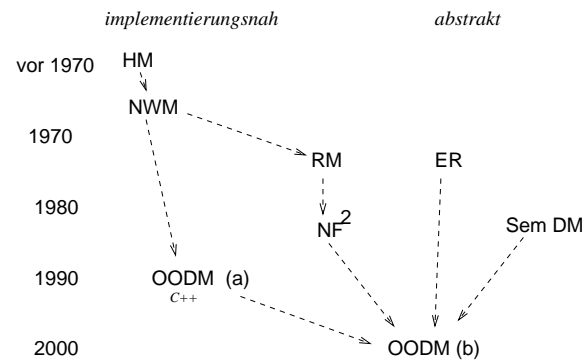
#### 2. *dynamische Eigenschaften* wie

- (a) Operationen
- (b) Beziehungen zwischen Operationen, sowie

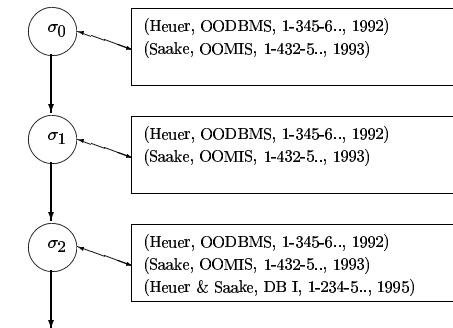
#### 3. *Integritätsbedingungen* an

- (a) Objekte
- (b) Operationen.

## Historische Einordnung und Bezüge zwischen einigen der vorgestellten Datenbankmodelle



## Formalisierung durch Zustandsfolgen



## Modelle für Daten und Algorithmen in verschiedenen Abstraktionsstufen

Modelle	Daten	Algorithmen
abstrakter	Entity-Relationship-Modell	Flußdiagramme
konkreter	Hierarchisches Modell	PASCAL
	Netzwerkmodell	FORTTRAN
	Relationenmodell	Ada

Neuere Modelle: deduktiv, objektorientiert

## Semantikfestlegung für Datenbankmodelle

- **Wertebereiche:**  
abstrakte Datentypen
- **Datenbankzustände:**  
Modell einer prädikatenlogischen Beschreibung
- **Gesamtsemantik:**  
Zustandsfolgen

## Semantikfestlegung am Beispiel II

Datenbankentwicklung:

$$\hat{\sigma} = \langle \sigma_0, \sigma_1, \dots, \sigma_i, \dots \rangle.$$

Bedeutung einer DB-Variablen ( $\mathcal{T}$ : Zeitachse):

$$\hat{\sigma}(db): \mathcal{T} \rightarrow \mu(\mathbf{typ}(db))$$

## Entity-Relationship-Modelle

P. P. Chen im Jahre 1976

**Entity:** Objekt der realen oder der Vorstellungswelt, über das Informationen zu speichern sind  
z.B. Vorlesungsveranstaltung, Buch, Lehrperson, ...  
Auch Informationen über Ereignisse: Prüfungen, ...

**Relationship:** Beziehung zwischen Entities, z.B. eine Lehrperson hält eine Vorlesung

**Attribut:** Eigenschaft von Entities oder Beziehungen, z.B. die ISBN eines Buchs, der Titel einer Vorlesung, oder das Semester, in dem eine Vorlesung gehalten wird

## Semantikfestlegung am Beispiel I

Trägermengen für *mögliche Werte*:  $\mu$

- $\mu(\mathbf{integer}) = \mathbb{Z}$  (die ganzen Zahlen  $\mathbb{Z}$ )
- $\mu(\mathbf{string}) = C^*$  (Folgen von Zeichen aus  $C = \{a, b, \dots, z, A, B, \dots, Z\}$ )
- $\mu(\mathbf{set}(z)) = 2^{\mu(z)}$

(die Potenzmenge über den Werten des Parameterdatentyps  $z$ , oder anders ausgedrückt die Mengen aller Teilmengen von möglichen Werten in  $\mu(z)$ )

- $\mu(\mathbf{tuple}(z_1, \dots, z_n)) = \mu(z_1) \times \dots \times \mu(z_n)$   
(das kartesische Produkt der Parameterwertebereiche)

## Semantikfestlegung am Beispiel III

Beispiel der Bücher-Datenbank:

$$\hat{\sigma}(\mathbf{Bücher}): \mathcal{T} \rightarrow 2^{C^* \times C^* \times C^* \times \mathbb{Z}}$$

Ein konkreter Zustandswert zum Zeitpunkt 42:

$$\hat{\sigma}(\mathbf{Bücher})(42) = \{(\text{Heuer}, 00DB, 1-453-, 1992), \\ (\text{Saake}, 00SIS, 1-321-, 1993)\}$$

## ER-Modellierungskonzepte und ihre Semantik I

### Werte

$\mu(\text{int})$  : der Wertebereich  $\mathbb{Z}$  (die ganzen Zahlen) mit  $+, -, \times, \div, =, <, \dots$

$\mu(\text{string})$  : der Wertebereich  $C^*$  (Folgen von Zeichen aus der Menge  $C$ ) mit  $+, =, <, \dots$

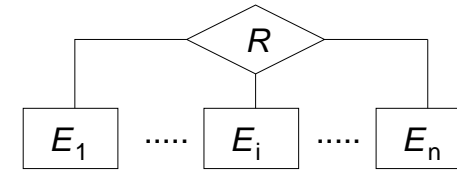
...

$\mu(D)$ : Interpretation von  $D$ , mögliche Werte einer Entity-Eigenschaft

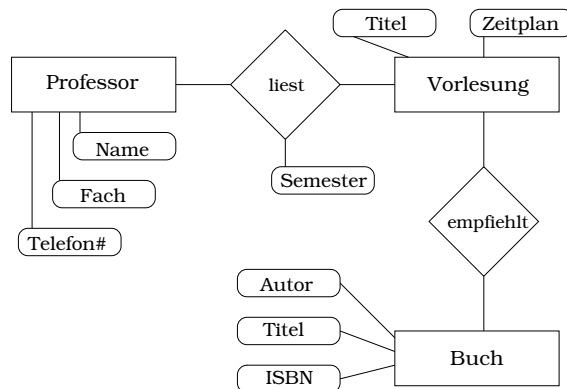
## ER-Modellierungskonzepte und ihre Semantik III

### Beziehungen: Beziehungstypen

Notation  $n$ -stelliger Beziehungstypen:



### Ein einfaches Beispiel



## ER-Modellierungskonzepte und ihre Semantik II

**Entities:** *Entity-Typen*, etwa  $E_1, E_2, \dots$



$\mu(E)$  Menge der *möglichen* Entities vom Typ  $E$   
wird hier nicht festgelegt (etwa Menge isomorph zu natürlichen Zahlen)

$\sigma(E)$  Menge der *aktuellen* Entities vom Typ  $E$  in einem Zustand  $\sigma$   
( $\sigma$ , *Sigma*, für *state* (Zustand))

Aktuelle Entities müssen mögliche Elemente sein:  $\sigma(E) \subseteq \mu(E)$   
Ferner gefordert:  $\sigma(E)$  endlich

## ER-Modellierungskonzepte und ihre Semantik V

Attribute:



Semantik einer Attributdeklaration : Ein Attribut  $A$  eines Entity-Typen  $E$  ist im Zustand  $\sigma$  eine Abbildung

$$\sigma(A) : \sigma(E) \rightarrow \mu(D)$$

## ER-Modellierungskonzepte und ihre Semantik IV

mögliche Ausprägungen:

$$\mu(R) = \mu(E_1) \times \dots \times \mu(E_n)$$

aktuelle Beziehungen nur zwischen aktuellen Entities:

$$\sigma(R) \subseteq \sigma(E_1) \times \dots \times \sigma(E_n)$$

Rollennamen:

`verheiratet(Frau : Person, Mann : Person)`

## Semantik eines ER-Schemas

Jeder Zustand  $\sigma$  eines ER-Schemas ist eine Zuordnung

$$E \mapsto \sigma(E) \subseteq \mu(E)$$

$$R(E_1, \dots, E_n; \dots) \mapsto \sigma(R) \subseteq \sigma(E_1) \times \dots \times \sigma(E_n)$$

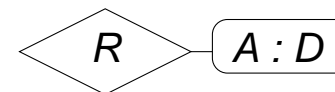
$$E(\dots, A_i : D, \dots) \mapsto \sigma(A_i) : \sigma(E) \rightarrow \mu(D), \dots$$

$$R(\dots; \dots, A_i : D, \dots) \mapsto \sigma(A_i) : \sigma(R) \rightarrow \mu(D), \dots$$

bei gegebener fester Interpretation  $\mu$  der Datentypen durch Wertebereiche und der Entity-Typen durch vorgegebene Mengen möglicher Entities.

## ER-Modellierungskonzepte und ihre Semantik VI

Beziehungsattribute:



Semantik:

$$\sigma(A) : \sigma(R) \rightarrow \mu(D)$$

Textuelle Notation:

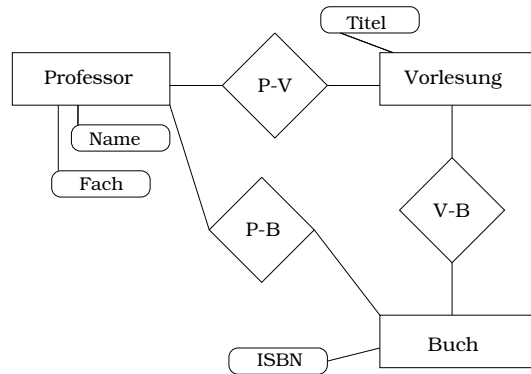
$$E(A_1 : D_1, \dots, A_m : D_m)$$

bzw.

$$R(E_1, \dots, E_n; A_1, \dots, A_p)$$

## Zweistellige versus mehrstellige Beziehungen II

Mögliche Umwandlung in zweistellige Beziehungen



## Ausprägungen im Beispiel II

Ausprägungen der drei 2-stelligen Beziehungstypen

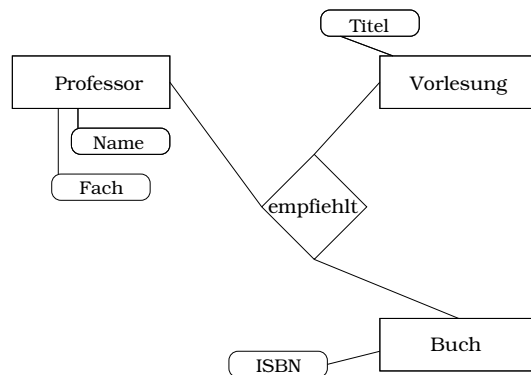
P-V	Prof.	Vorl.	P-B	Prof.	Buch	V-B	Vorl.	Buch
	Heuer	DB 1		Heuer	1-234-..		DB 1	1-234-..
	Heuer	DB 2		Heuer	9-876-..		DB 2	9-876-..
	Saake	DB 1		Saake	9-876-..		DB 1	9-876-..
	Saake	DB 2						

... entsprechen aber auch:

empfiehl	Professor	Vorlesung	Buch (ISBN)
	Heuer	DB 1	1-234-..
	Heuer	DB 1	9-876-..
	Heuer	DB 2	9-876-..
	Saake	DB 1	9-876-..
	Saake	DB 2	9-876-..

## Zweistellige versus mehrstellige Beziehungen

Dreistellige Beziehung



## Ausprägungen im Beispiel

Korrekte Ausprägung der dreistelligen Beziehung

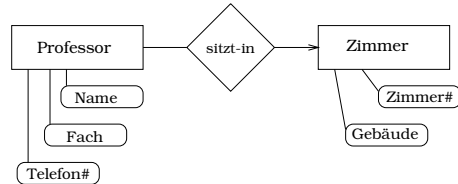
empfiehl	Professor	Vorlesung	Buch (ISBN)
	Heuer	DB 1	1-234-..
	Heuer	DB 2	9-876-..
	Saake	DB 1	9-876-..
	Saake	DB 2	9-876-..

## Funktionale Beziehungen

Textuell:

$$R : E_1 \rightarrow E_2$$

Graphisch:



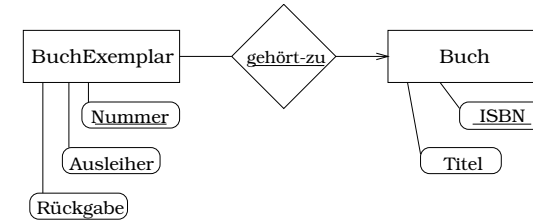
Bedeutung:

$$\sigma(R) : \sigma(E_1) \rightarrow \sigma(E_2)$$

## Abhängige Entity-Typen I

abhängiger Entity-Typ:

Identifikation über funktionale Beziehung



Abhängige Entities im ER-Modell:

Funktionale Beziehung als Schlüssel

## Ausprägungen im Beispiel III

Jetzt außerdem möglich:

P-V	Prof.	Vorl.	P-B	Prof.	Buch	V-B	Vorl.	Buch
	Heuer	DB 1		Heuer	1-234-..		DB 1	1-234-..
	Heuer	DB 2		Heuer	9-876-..		DB 2	9-876-..
	Saake	DB 1		Saake	9-876-..		DB 1	9-876-..
	Saake	DB 2					DB 3	4-242-..

## Identifizierung durch Schlüssel

Für Entity-Typ  $E(A_1, \dots, A_m)$  sei Teilmenge  $\{S_1, \dots, S_k\}$  der gesamten Attribute gegeben, die *Schlüsselattribute*.

Es gilt:  $\{S_1, \dots, S_k\} \subseteq \{A_1, \dots, A_m\}$ .

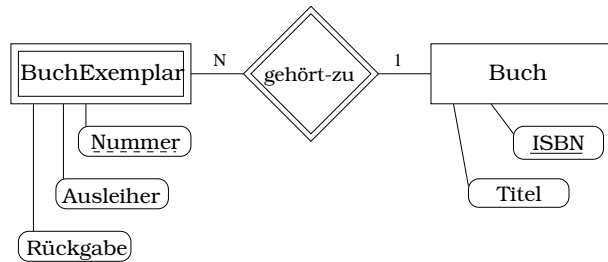
In jedem Datenbankzustand identifizieren die aktuellen Werte der Schlüsselattribute eindeutig Instanzen des Entity-Typs  $E$ :

$$\forall e_1, e_2 \in \sigma(E) : (\sigma(S_1)(e_1) = \sigma(S_1)(e_2) \wedge \dots \wedge \sigma(S_k)(e_1) = \sigma(S_k)(e_2)) \implies (e_1 = e_2)$$

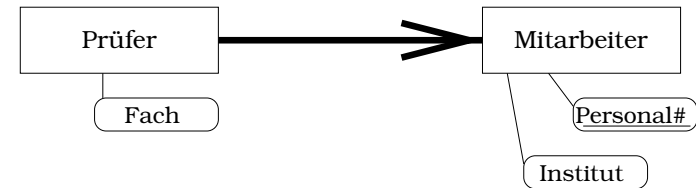
Notation: markieren durch Unterstreichung:

$$E(\dots, \underline{S_1}, \dots, \underline{S_i}, \dots)$$

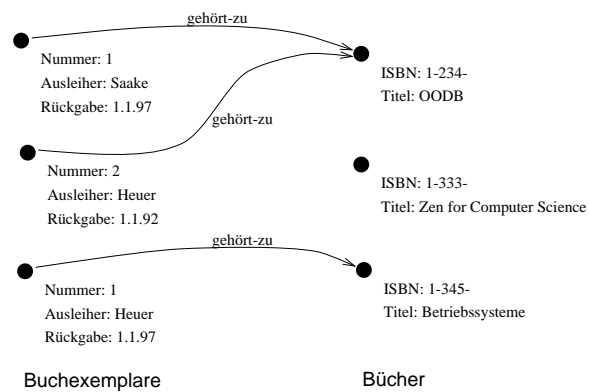
## Abhängige Entity-Typen: Alternative Notation



## Die IST-Beziehung: Alternative Notation

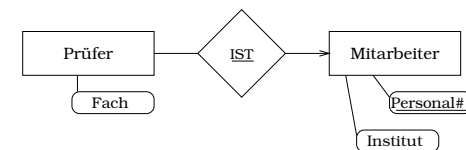


## Abhängige Entity-Typen II



## Mögliche Ausprägung für abhängige Entities

## Die IST-Beziehung



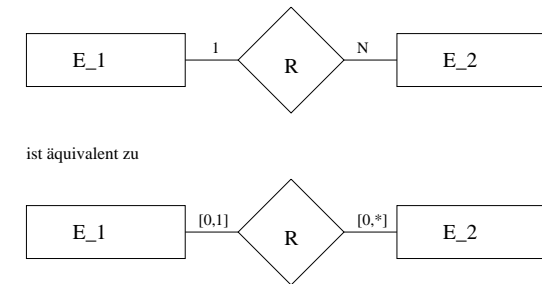
- Jeder Prüfer-Instanz ist genau einer Mitarbeiter-Instanz zugeordnet: Spezialfall eines abhängigen Entity-Typs!
- Nicht jeder Mitarbeiter ist zugleich Prüfer.
- Attribute des Entity-Typs Mitarbeiter treffen auch auf Prüfer zu: "vererbte" Attribute. Prüfer (Name, Personal#, Fach) *von Mitarbeiter*  
Nicht nur Deklarationen vererben sich, sondern auch aktuelle Werte.
- Bedeutung:

$$\sigma(E_1) \subseteq \sigma(E_2)$$

## Kardinalitäten II

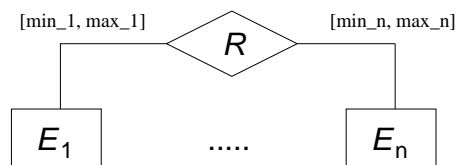
- $[0, *]$  ist Standardannahme.
  - Die Angabe  $R(E_1[0, 1], E_2)$  entspricht einer (partiellen) funktionalen Beziehung  $R : E_1 \rightarrow E_2$ , da jede Instanz aus  $E_1$  maximal einer Instanz aus  $E_2$  zugeordnet ist.
- Eine totale funktionale Beziehung wird durch  $R(E_1[1, 1], E_2)$  modelliert.

## Vereinfachte Kardinalitätsangaben für binären Beziehungstyp



Die Angabe  $N$  entspricht  $*$

## Kardinalitäten I



- Notation für Kardinalitätsangaben an einem Beziehungstyp

$$R(E_1, \dots, E_i[\min_i, \max_i], \dots, E_n)$$

- Kardinalitätsbedingung:  $\min_i \leq |\{r \mid r \in R \wedge r.E_i = e_i\}| \leq \max_i$
- Spezielle Wertangabe für  $\max_i$  ist  $*$

## Kardinalitäten: Beispiele

- Beispiele

`arbeitet_in(Mitarbeiter[0,1],Raum[0,3])`

Jedem Mitarbeiter ist in der Regel ein Raum zugeordnet, aber einige (externe) Mitarbeiter haben kein Arbeitszimmer. Pro Zimmer arbeiten maximal drei Mitarbeiter.

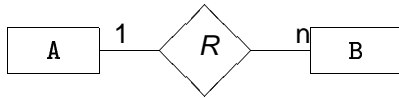
`verantwortlich_für(Mitarbeiter[0,*],Rechner[1,1])`

Jedem Rechner ist genau ein Mitarbeiter zugeordnet, der für die Betreuung verantwortlich ist.

- Für die Beziehung  $E_1$  IST  $E_2$  gilt:  $IST(E_1[1,1], E_2[0,1])$ .

## Alternative Notationen für Kardinalitäten II

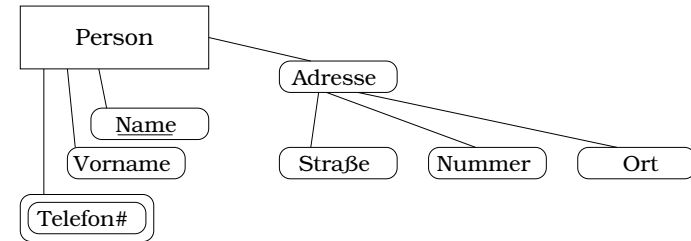
### ■ Kardinalitäten als Teilnahmezahl:



Achtung: Umgekehrt zum vorigen Fall!

## Weitere Konzepte

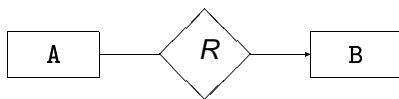
### ■ Strukturierte Attributwerte im ER-Modell



## Alternative Notationen für Kardinalitäten

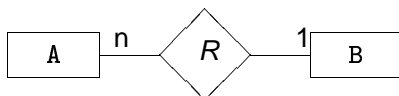
### ■ Standard-Entity-Relationship-Modell:

Pfeile

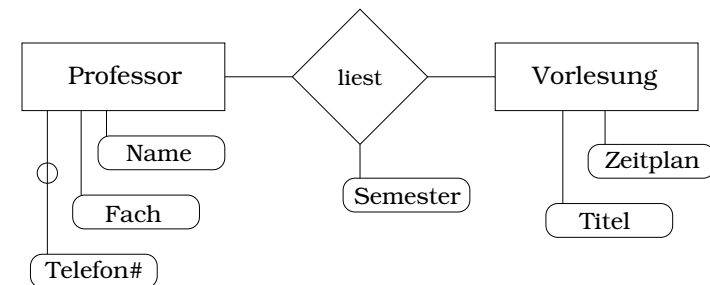


### ■ Standard-ER-Modell mit Kardinalitäten:

Numerische Werte statt Pfeilen



## Optionalität von Attributen und Beziehungen



### Optionale Attribute im ER-Modell

## Erweiterungen des ER-Modells I

### ■ Spezialisierung und Generalisierung

- ◆ *Spezialisierung* entspricht IST-Beziehung:  
Professor Spezialisierung von Mitarbeiter
- ◆ *Generalisierung*: Entities in einen *allgemeineren* Kontext.  
Person oder Institut als Ausleiher
- ◆ *Partitionierung*: Spezialfall der Spezialisierung, mehrere *disjunkte* Entity-Typen.  
Partitionierung von Büchern in Monographien und Sammelbänden.

## Ein erweitertes ER-Modell: Das EER-Modell

### ■ Übernommene Grundkonzepte des ER-Modells

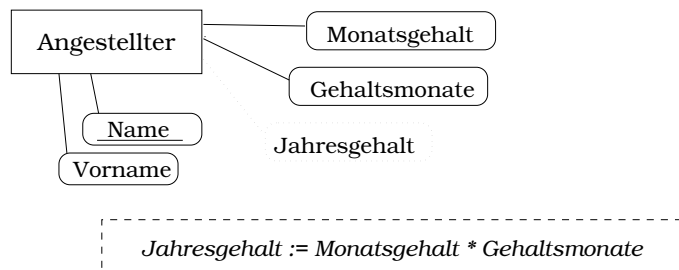
- ◆ *Werte*: Standard-Datentypen des ER-Modells.
- ◆ *Entities* bzw. *Entity-Typen*.
- ◆ *Beziehungen* bzw. *Beziehungstypen*.
- ◆ *Attribute*: unverändert.
- ◆ *Funktionale Beziehungen*: unverändert.
- ◆ *Schlüssel*: erweitertes Konzept, neue Notation.

### ■ Nicht übernommen:

- ◆ IST-Beziehung ersetzt durch *Typkonstruktor*.
- ◆ Abhängige Entity-Typen durch erweiterte Schlüsselkonzept objektwertige Attribute ersetzt.

## Weitere Konzepte

### ■ Abgeleitete Attributwerte im ER-Modell



## Erweiterungen des ER-Modells II

### ■ Komplexe Objekte

- ◆ *Aggregation*: Entity aus einzelnen Instanzen anderer Entity-Typen zusammengesetzt.  
Fahrzeug zusammengesetzt aus Motor, Karosserie...
- ◆ *Sammlung* oder *Assoziation*: Mengenbildung.  
Team als Gruppe von Personen.

### ■ Beziehungen höheren Typs

- ◆ *Spezialisierung* und *Generalisierung* auch für Beziehungstypen.  
Beispiel: Beziehung *Ausleihe* zu *Kurzausleihe* spezialisiert.
- ◆ *Beziehungen zwischen Beziehungsinstanzen*: Beziehungen zweiter und höherer Ordnung

## Komplexe Werte im EER-Modell

### Mengen- und tupelwertige Attribute im EER-Modells

<b>prod:</b> Tupelbildung	$\text{point} = \text{prod}(\text{real}, \text{real})$
<b>list:</b> Listen / Folgen von Werten	$\text{polygon} = \text{list}(\text{point})$
<b>set:</b> Mengen	$\text{holidays} = \text{set}(\text{date})$
<b>bag:</b> Multimengen	$\text{birthdays\_of\_a\_group} = \text{bag}(\text{date})$

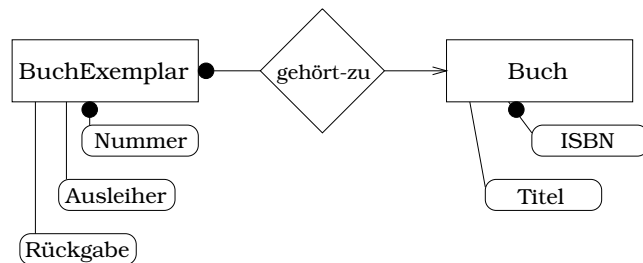
## Nicht-Standard-Datentypen im EER-Modell

Ein Nicht-Standard-Datentyp  $D$  stellt einen Wertebereich  $\mu(D)$  mit Operationen dar, z.B.:

$$\mu(\text{point}) = \mu(\text{real}) \times \mu(\text{real})$$

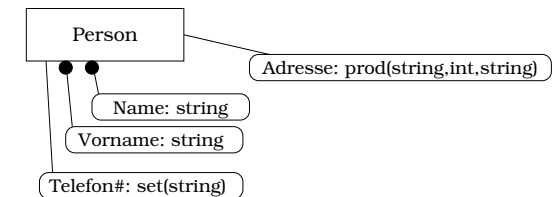
Für jeden Nicht-Standard-Datentyp können spezifische Operationen, etwa +, −, **distance** etc. für den Datentyp **point**, durch Gleichungen spezifiziert oder in einer imperativen, an Programmiersprachen angelehnten Notation definiert werden.

## Schlüsselnotation im EER-Modell

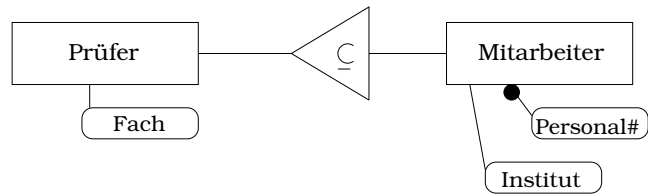


## Komplexe Werte im EER-Modell II

Graphische Notation:

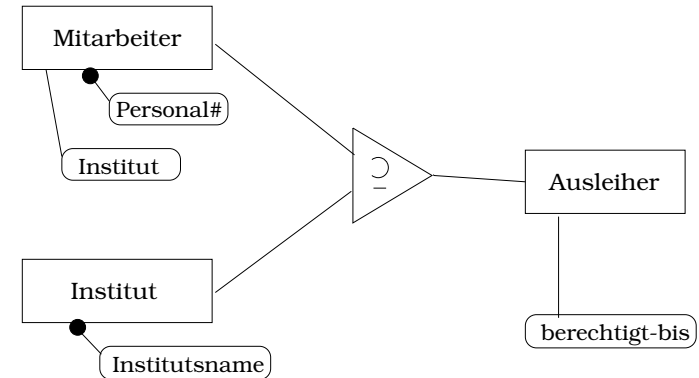


## Spezialisierung mit Typkonstruktor



Spezialisierung (IST-Beziehung) notiert mit dem Typkonstruktor des EER-Modells

## Beispiel für Generalisierung im EER-Modell



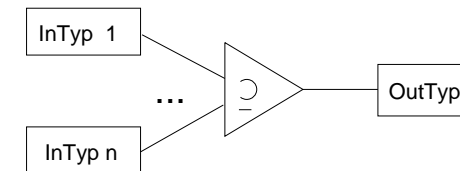
## Typkonstruktor: Spezialisierung, Generalisierung, Partitionierung

Ein Modellierungskonzept für

- Spezialisierung / IST-Beziehung
- Generalisierung
- Partitionierung

## Generalisierung

Notation: Typkonstruktor für Generalisierung

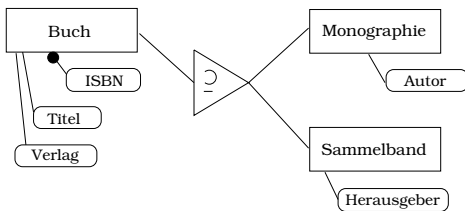


Bedeutung:

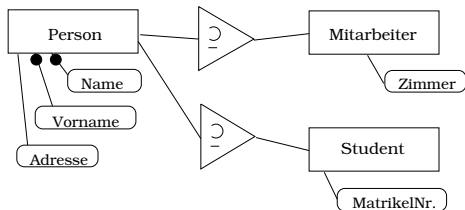
$$\bigcup_i \sigma(\text{InTyp}_i) \supseteq \sigma(\text{OutTyp})$$

## Partitionierung versus mehrfache Spezialisierung

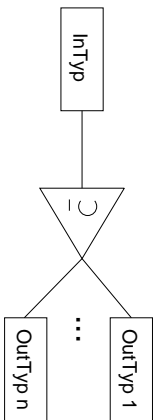
Beispiel für Partitionierung im EER-Modell



Mehrfache Spezialisierung im EER-Modell



## Partitionierung



Semantik:

- Teilmengenbeziehung:  $\sigma(\text{InTyp}) \supseteq \bigcup_i (\sigma(\text{OutTyp}_i))$
- Disjunktheit der Partitionen:

$$\forall i, j : i \neq j \implies \sigma(\text{OutTyp}_i) \cap \sigma(\text{OutTyp}_j) = \{\}$$

## Partitionierung und Generalisierung im Vergleich

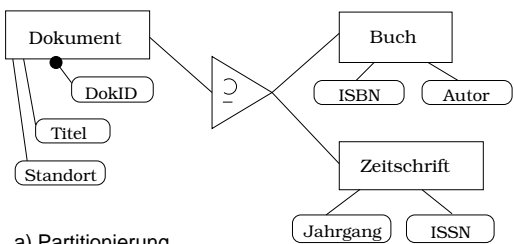
Partitionierung:

$$\sigma(\text{Dokument}) \supseteq \sigma(\text{Buch}) \cup \sigma(\text{Zeitschrift})$$

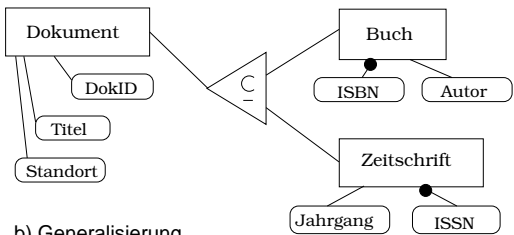
Generalisierung:

$$\sigma(\text{Dokument}) \subseteq \sigma(\text{Buch}) \cup \sigma(\text{Zeitschrift})$$

## Partitionierung versus Generalisierung



a) Partitionierung



b) Generalisierung

## Semantik des allgemeinen Typkonstruktors II

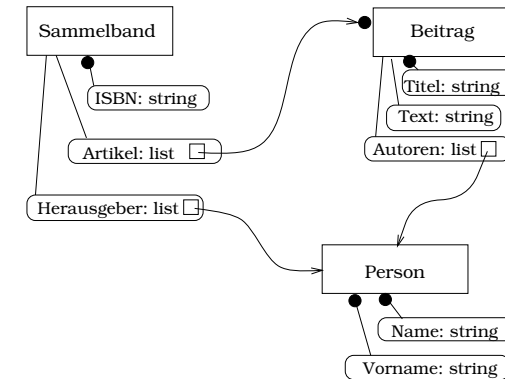
Falls  $X$  Gleichheitszeichen:

$$\bigcup_j (\sigma(\text{InTyp}_j)) = \bigcup_i (\sigma(\text{OutTyp}_i))$$

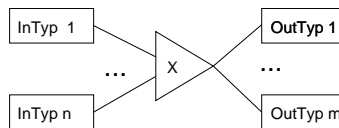
Ausgabetypen sind disjunkt:

$$\forall i, j: i \neq j \implies \sigma(\text{OutTyp}_i) \cap \sigma(\text{OutTyp}_j) = \{\}$$

## Aggregation und Sammlung mittels objektwertiger Attribute



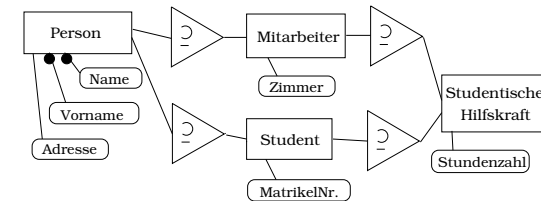
## Semantik des allgemeinen Typkonstruktors



Ausgabetypen sind Spezialisierungen der Eingabetypen:

$$\bigcup_j (\sigma(\text{InTyp}_j)) \supseteq \bigcup_i (\sigma(\text{OutTyp}_i))$$

## Mehrfachspezialisierung



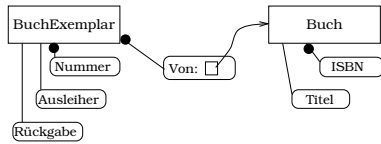
Mehrfachspezialisierung zu StudentischeHilfskraft

$$\sigma(\text{StudentischeHilfskraft}) \subseteq \sigma(\text{Student}) \cap \sigma(\text{Mitarbeiter})$$

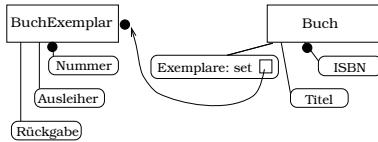
*Mehrfachspezialisierungen sind nur erlaubt, wenn die Eingabe-Typen direkt oder indirekt aus einer gemeinsamen Ausgangsklasse konstruiert wurden.*

## Erweitertes Schlüsselkonzept

- Objektwertige Attribute als Schlüsselattribute:



- Einsatz zur Modellierung abhängiger Entity-Typen:

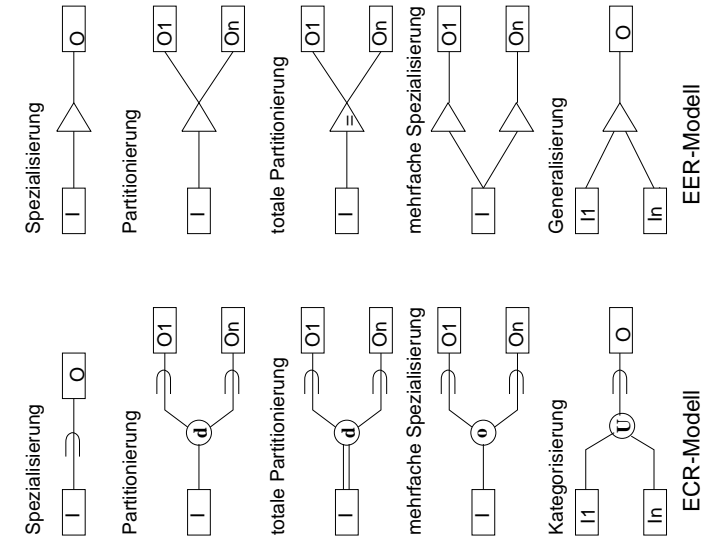


## Objektwertige Attribute im EER-Modell

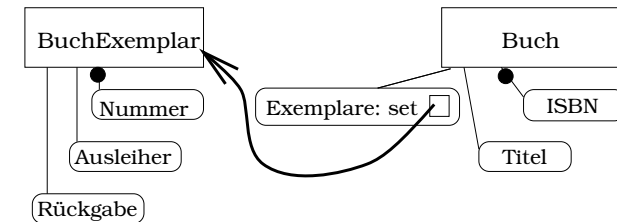
Ein objektwertiges Attribut  $A$  eines Entity-Typs  $E(\dots, A : E_A, \dots)$ , das Werte eines anderen Entity-Typs  $E_A$  annehmen kann, entspricht der Funktion:

$$\sigma(A) : \sigma(E) \rightarrow \sigma(E_A)$$

## Notationen im ECR- und EER-Modell



## Modellierung abhängiger Entity-Typen im EER-Modell (alternative Notation)



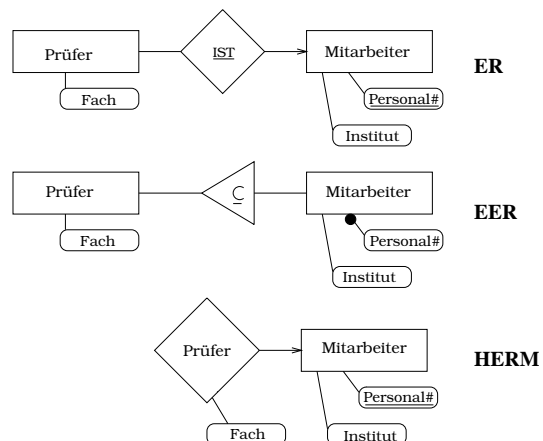
## Begriffe des ER- und EER-Modells I

Begriff	Informale Bedeutung	ER	EER
Entity	zu repräsentierende Informationseinheit	x	x
Entity-Typ	Gruppierung von Entities mit gleichen Eigenschaften	x	x
Beziehungstyp	Gruppierung von Beziehungen zwischen Entities	x	x
Attribut	datenwertige Eigenschaft eines Entities oder einer Beziehung	x	x
funktionale Beziehung	Beziehungstyp mit Funktionseigenschaft	x	x

## Begriffe des ER- und EER-Modells III

Begriff	Informale Bedeutung	ER	EER
Optionalität	Attribute oder funktionale Beziehungen als partielle Funktionen	x	x
strukturierte Attribute	zusammengesetzte Attributwerte		x
abgeleitete Attribute	durch eine Berechnungsvorschrift berechnete Attributwerte		x
Spezialisierung	Verfeinerung eines Entity-Typen zu einem spezielleren Entity-Typ		x
Generalisierung	Zusammenfassung von Entity-Typen zu einem allgemeineren Entity-Typ		x

## Spezialisierung im ER-Modell, im EER-Modell und HERM



## Begriffe des ER- und EER-Modells II

Begriff	Informale Bedeutung	ER	EER
Schlüssel	identifizierende Eigenschaft von Entities	x	x
abhängige Entities	Entities, die nur abhängig von anderen Entities existieren können	x	
IST-Beziehung	Spezialisierung von Entity-Typen	x	
Kardinalitäten	Einschränkung von Beziehungstypen bezüglich der mehrfachen Teilnahme von Entities an ihr	x	x

## Objektorientierte Modelle für den Entwurf

Seit Beginn der 90er Jahre: Ansätze zur objektorientierten Analyse und zum objektorientierten Entwurf

Ansätze der ersten Generation

- *Object Modelling Technique (OMT)* von Rumbaugh
- Methoden von Jacobson, Booch, ...

Ab Mitte der 90er Jahre: Vereinheitlichung zur *Unified Modeling Language (UML)* (Booch, Jacobson, Rumbaugh)

## Objektmodell von UML

- Erweiterung des EER-Modells
- Unterscheidung in
  - ◆ **Klassendiagramm:** entspricht Datenbankschema-Notation  $\rightsquigarrow$  beschreibt Typen von Kollektionen von Instanzen
  - ◆ **Objektdiagramm:** beschreibt Einzelobjekte
- Beschreibung von *strukturellen Aspekten* (Attribute, Beziehungen) und *Operationen*
- Formulierung von Integritätsbedingungen und Ableitungsregeln (textuell, Object Constraint Language)

## Begriffe des ER- und EER-Modells IV

Begriff	Informale Bedeutung	ER	EER
Partitionierung	mehrere disjunkte Spezialisierungen eines Entity-Typen		x
Aggregierung	Zusammensetzung von Entities aus anderen Entities		x
objektwertige Attribute	Attribute, die Entity-Typen als Wertebereich haben		x

## Entwurf mit UML

- **Objektmodell:** Anreicherung von EER-Modellen um objektorientierte Konzepte (Klassen, Beziehungen, Operationen, ...)
- **Dynamikmodell:** Beschreibung des dynamischen Verhaltens eines Objektes in Form von Übergangsautomaten (Übergänge entsprechen Systemereignissen)
- **Funktionenmodell:** Datenflußdiagramme zur Darstellung globaler Berechnungsabläufe
- **Anwendungsfälle**
- **Implementierungsdiagramme**
- ...

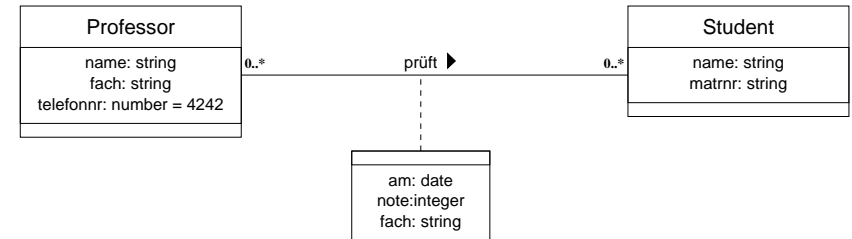
## Darstellung von Klassen II

### Weitere Angaben

- verschiedene Arten von Klassen, speziell *abstrakte Klassen* (Notation: { **abstract** }, *Metaklassen* und *parametrisierte Klassen*)
- Sichtbarkeit von Attributen (**private**, **protected**, **public**) sowie **readonly**-Attribute
- Abgeleitete (berechnete) Attribute durch vorangestelltes /-Symbol
- Klassenattribute durch Unterstreichung

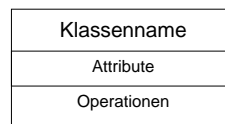
## Beziehungen mit Attributen

### Beziehungen mit Attributen als „degenerierte“ Klassen



## Darstellung von Klassen

### ■ Klassen



### ■ Attribute

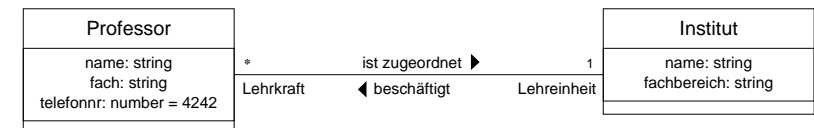
name: typ = initialerWert { Zusicherung }

### ■ Operationen

name { Parameterliste }

## Beziehungen

### Standardfall: binäre Beziehungen (*Assoziationen*)



### ■ Bezeichner der Beziehung mit Leserichtung (keine funktionale Beziehung)

### ■ Rollennamen für Implementierung von Referenzattributen

### ■ Kardinalitäten analog zum ER-Modell

### ■ optional: Zusicherung

## Weitere Beziehungen

### ■ Abgeleitete Beziehungen

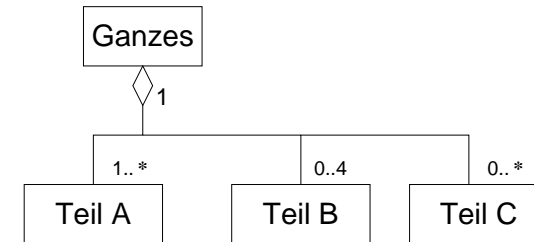
- ◆ Beispiel: berechnete Referenz zur Abkürzung von mehrstufigen Referenzpfaden
- ◆ Notation: Bezeichner mit vorangestelltem /-Symbol

### ■ n-stellige Beziehungen

- ◆ Notation wie im ER-Modell durch Raute

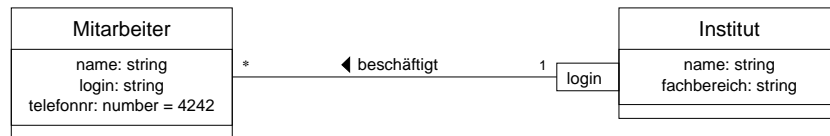
## Aggregation in UML II

Baumdarstellung für mehrere aggregierte Teile



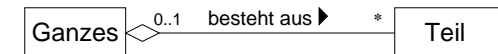
## Qualifizierende Beziehungen

Zugriffsschlüssel für spätere Implementierung

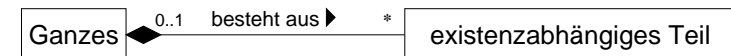


## Aggregation in UML

Aggregation über binäre Assoziation mit besonderer Notation

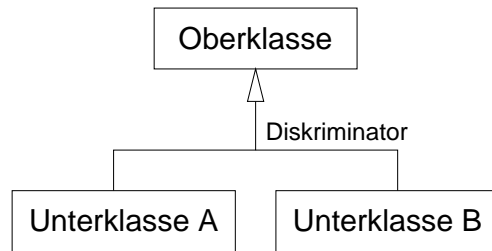


Komposition: Abhängige Objekte als Spezialfall



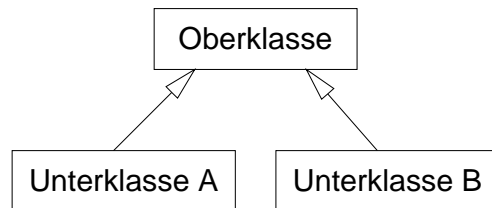
## Spezialisierung mit Diskriminator

- Spezielle Form der Spezialisierung
- Aufzählungsattribut (*Diskriminator*) teilt Instanzen in Unterklassen auf
- Wertebereich des Diskriminators: beteiligte Klassennamen



## Spezialisierung

Spezialisierung mit Vererbung



Angabe von Zusicherungen (**overlapping**, **disjoint**, **complete**, **incomplete**) möglich