



Kapitelübersicht

- ▶▶▶ **Grundlagen von Datenbankmodellen**
- ▶▶▶ **Entity-Relationship-Modelle**
- ▶▶▶ **Objektorientierte Modelle: UML**



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 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.



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)).**



Gegenüberstellung DB- und Programmiersprachenkonzepte:

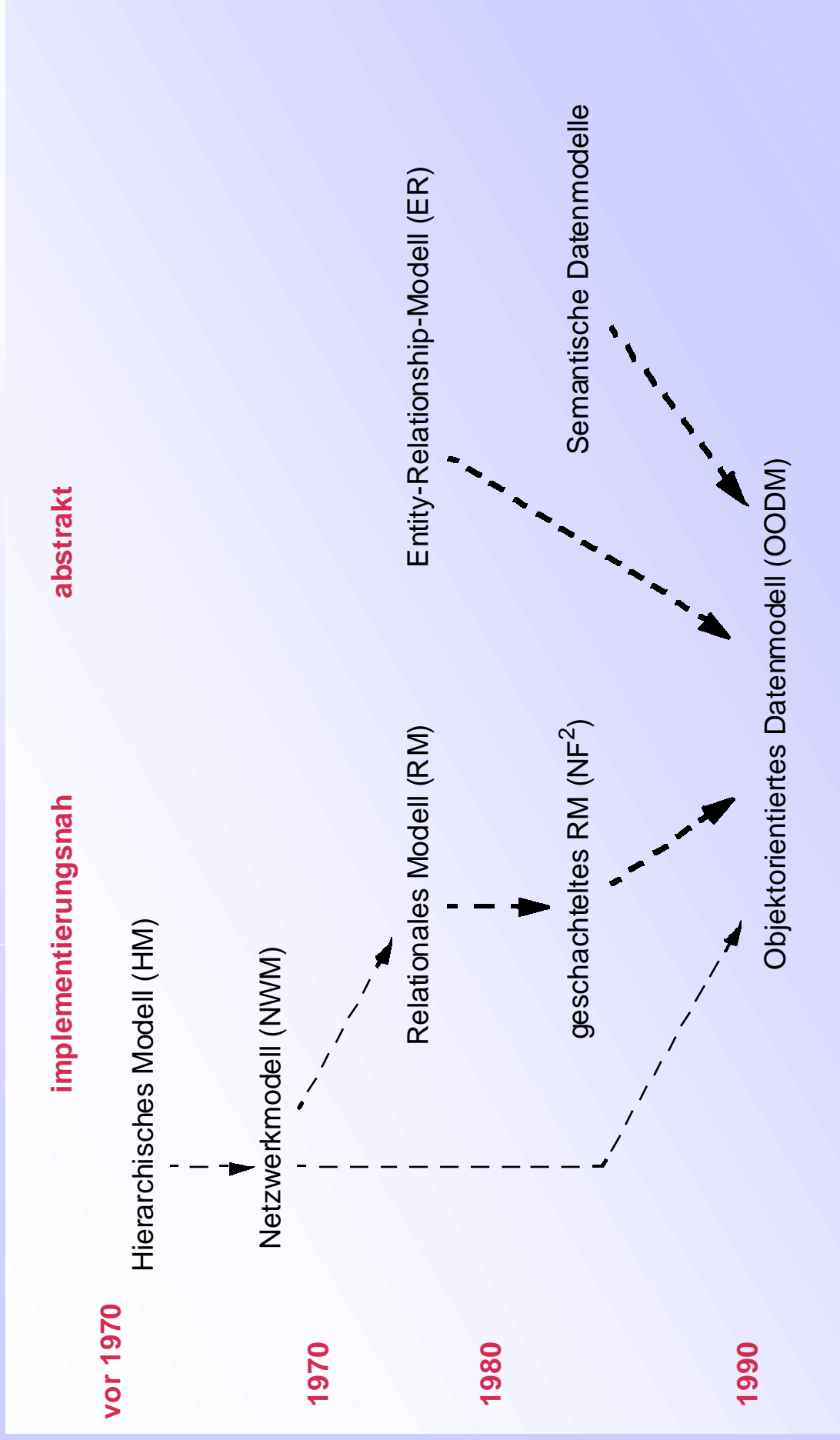
Datenbankkonzept	Typsystem einer Programmiersprache
Datenbank (DB)	Werte
Datenbankschema (DB-Schema)	Variablendeklaration mit Typdefinition
Datenbankmodell	Typsystem der Sprache
Datenbanksystem (DBS) = DB + DBMS	Programm mit Datenbestand
Datenbankmanagementsystem (DBMS)	Entwicklungs- und Laufzeitumgebung

Achtung:

Diese Gegenüberstellung vernachlässigt die Operationen auf Daten und gilt in erster Linie für den Vergleich klassischer DBS mit imperativen Programmiersprachen.



Historische Einordnung von Datenmodellen

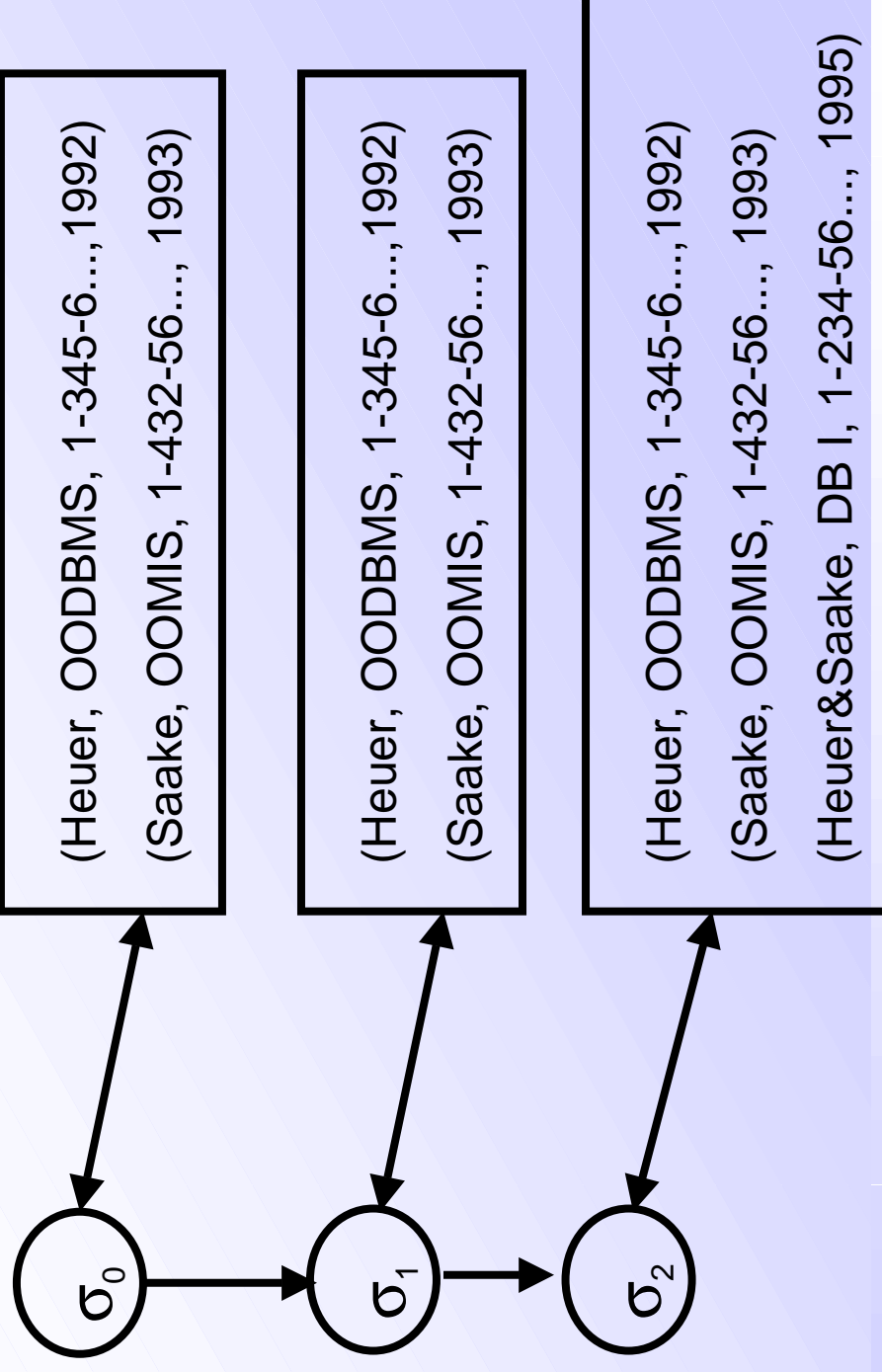




Semantikfestlegung für Datenbankmodelle

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

Formalisierung durch Zustandsfolgen





Semantikfestlegung am Beispiel I

Trägermengen für *mögliche* Werte: μ

- $\mu(\mathbf{integer}) = \mathbb{Z}$ (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 II

Datenbankentwicklung:

$$\underline{\sigma} = (\sigma_0, \sigma_1, \dots, \sigma_i, \dots).$$

Bedeutung einer DB-Variablen (T : Zeitachse):

$$\underline{\sigma}(\text{db}): T \rightarrow \mu(\text{typ}(\text{db}))$$



Semantikfestlegung am Beispiel III

Beispiel der Bücher-Datenbank:

$\underline{\sigma}$ (Bücher): $T \rightarrow 2^{C^* \times C^* \times C^* \times Z}$

Ein konkreter Zustandswert zum Zeitpunkt 42:

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



Entity-Relationship-Modelle

P.P. Chen im Jahre 1976

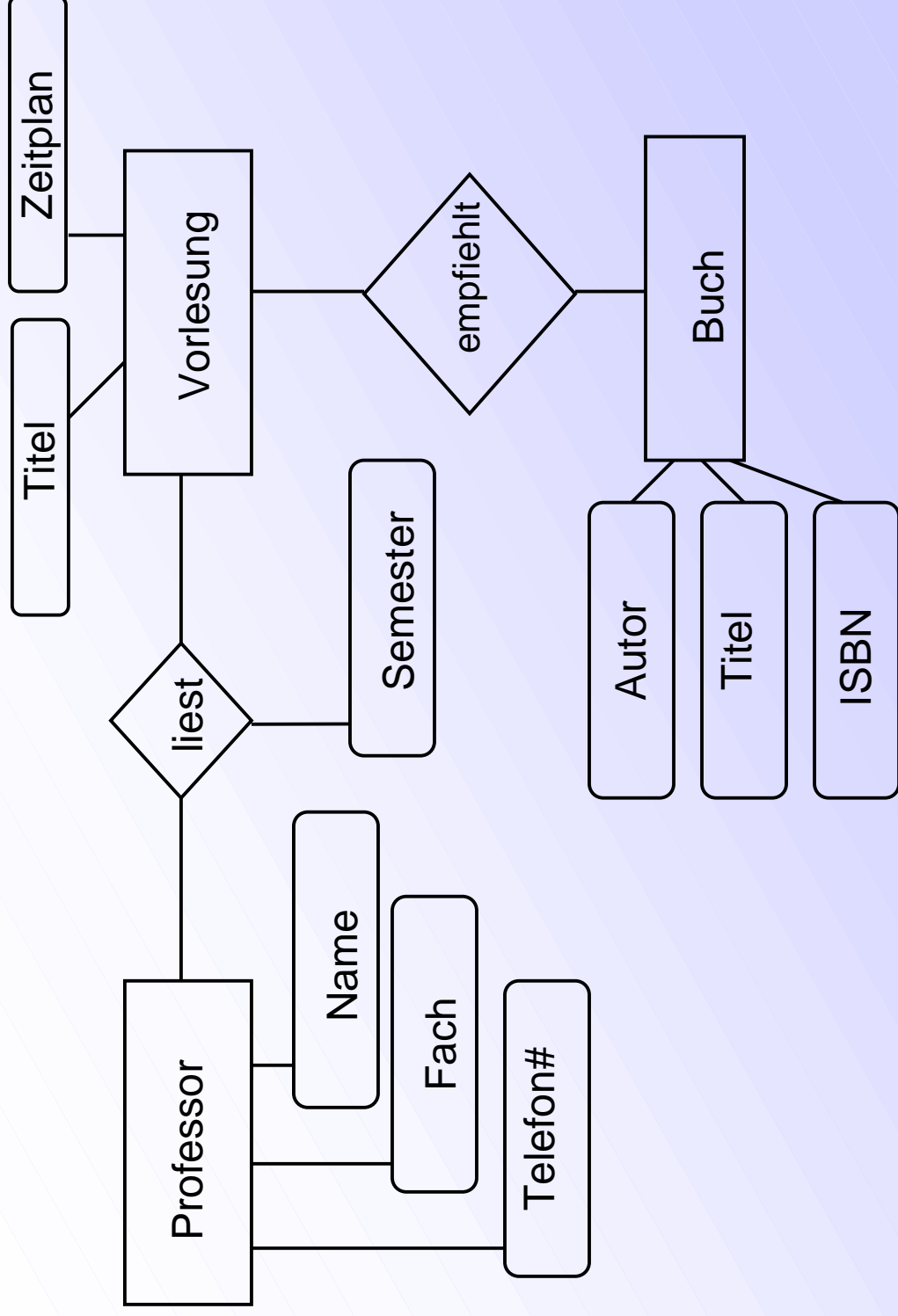
Entity: Objekt der realen oder der Vorstellungswelt, über das Informationen zu speichern sind
z. B. Vorlesungsveranstaltungen, Buch, Lehrperson, ...
Auch Informationen über Ereignisse: Prüfungen, ...

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

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



Ein einfaches Beispiel





ER-Modellierungskonzepte und ihre Semantik I

Werte

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

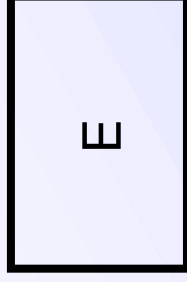
$\mu(\mathbf{string})$: der Wertebereich C^* (Folgen von Zeichen aus der Menge C) mit $+$, $=$, $<$, \dots

...

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

ER-Modellierungskonzepte und ihre Semantik II

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



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

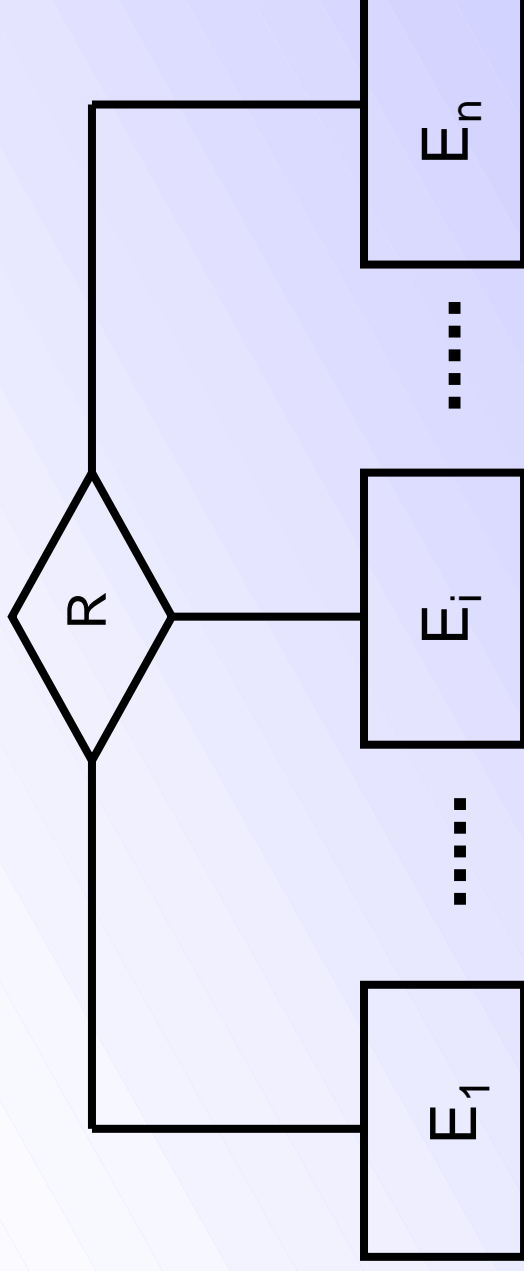
$\sigma(\mathbf{E})$ Menge der *aktuellen* Entities vom Typ E in einem Zustand σ
(σ , *Sigma*, für *state* (Zustand))

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

ER-Modellierungskonzepte und ihre Semantik III

Beziehungen: *Beziehungstypen*

Notation n -stelliger Beziehungstypen:





ER-Modellierungskonzepte und ihre Semantik IV

mögliche Ausprägungen:

$$\mu(\mathbf{R}) = \mu(\mathbf{E}_1) \times \dots \times \mu(\mathbf{E}_n)$$

aktuelle Beziehungen nur zwischen aktuellen Entities:

$$\sigma(\mathbf{R}) \subseteq \sigma(\mathbf{E}_1) \times \dots \times \sigma(\mathbf{E}_n)$$

Rollennamen:

verheiratet(Frau : Person, Mann : Person)

ER-Modellierungskonzepte und ihre Semantik V

Attribute:

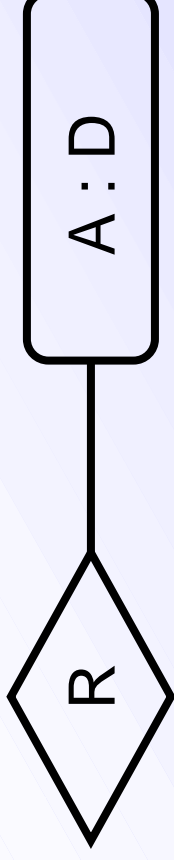


Semantik einer Attributdeklaration: Ein Attribut A eines Entity-Typen E ist im Zustand σ eine Abbildung

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

ER-Modellierungskonzepte und ihre Semantik VI

Beziehungsattribute:



Semantik:

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

Textuelle Notation für Attribute:

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

bzw.

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

Semantik eines ER-Schemas

Jeder Zustand σ eines ER-Schemas ist eine Zuordnung

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

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

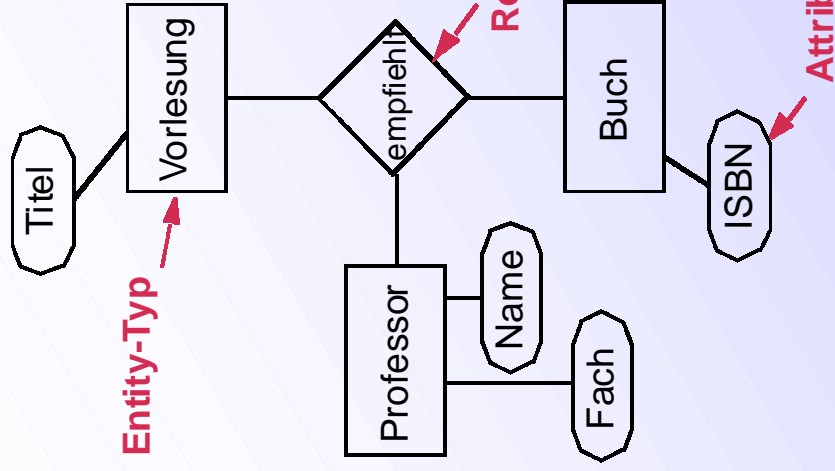
$$E(\dots, A_j : D, \dots) \longmapsto \sigma(A_j) : \sigma(E) \rightarrow \mu(D), \dots$$

$$R(\dots; \dots, A_j : D, \dots) \longmapsto \sigma(A_j) : \sigma(R) \rightarrow \mu(D), \dots$$

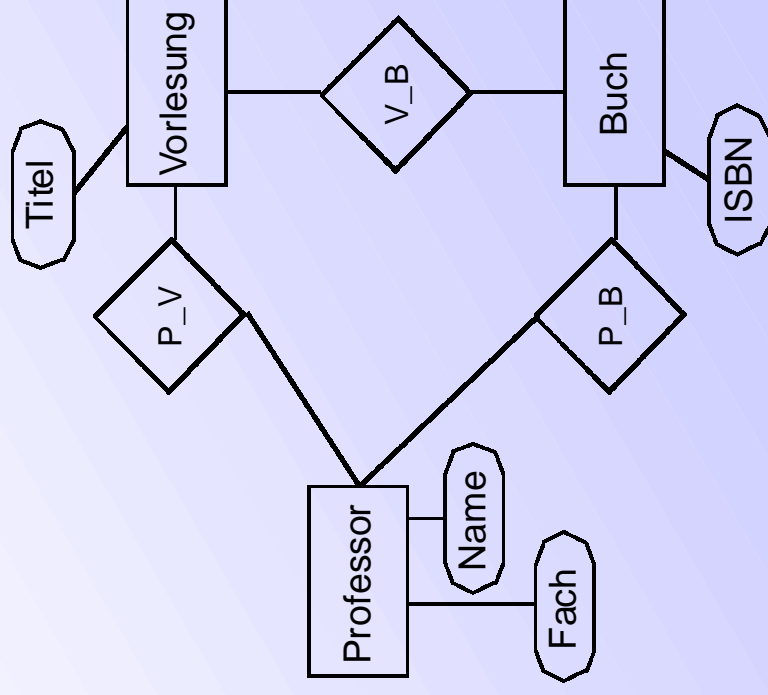
bei gegebener fester Interpretation μ der Datentypen durch Wertebereiche und der Entity-Typen durch vorgegebene Mengen möglicher Entities.

Zweistellige versus mehrstellige Beziehungen

ER-Diagramm mit dreistelliger Beziehung



mögliche Umwandlung in zweistellige Beziehungen





Problem mit Umwandlung n-stelliger in 2-stellige Beziehungen:

empfiehlt (korrekte Fassung)

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

Ausprägungen der drei zweistelligen Beziehungen:

P_V

Professor	Vorlesung
Heuer	DB 1
Heuer	DB 2
Saake	DB 1
Saake	DB 2

P_B

Professor	Buch
Heuer	1-234...
Heuer	9-876...
Saake	9-876...

V_B

Vorlesung	Buch
DB 1	1-234...
DB 2	9-876...
DB 1	9-876...



Problem mit Umwandlung n-stelliger in 2-stellige Beziehungen:

empfiehl (falsche, d.h. nicht beabsichtigte Fassung)

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...



Ausprägungen der drei zweistelligen Beziehungen:

P_V

Professor	Vorlesung
Heuer	DB 1
Heuer	DB 2
Saake	DB 1
Saake	DB 2

P_B

Professor	Buch
Heuer	1-234...
Heuer	9-876...
Saake	9-876...

V_B

Vorlesung	Buch
DB 1	1-234...
DB 2	9-876...
DB 1	9-876...



Problem mit Umwandlung n-stelliger in 2-stellige Beziehungen:

jetzt auch möglich:

P_V

Professor	Vorlesung
Heuer	DB 1
Heuer	DB 2
Saake	DB 1
Saake	DB 2

P_B

Professor	Buch
Heuer	1-234...
Heuer	9-876...
Saake	9-876...

V_B

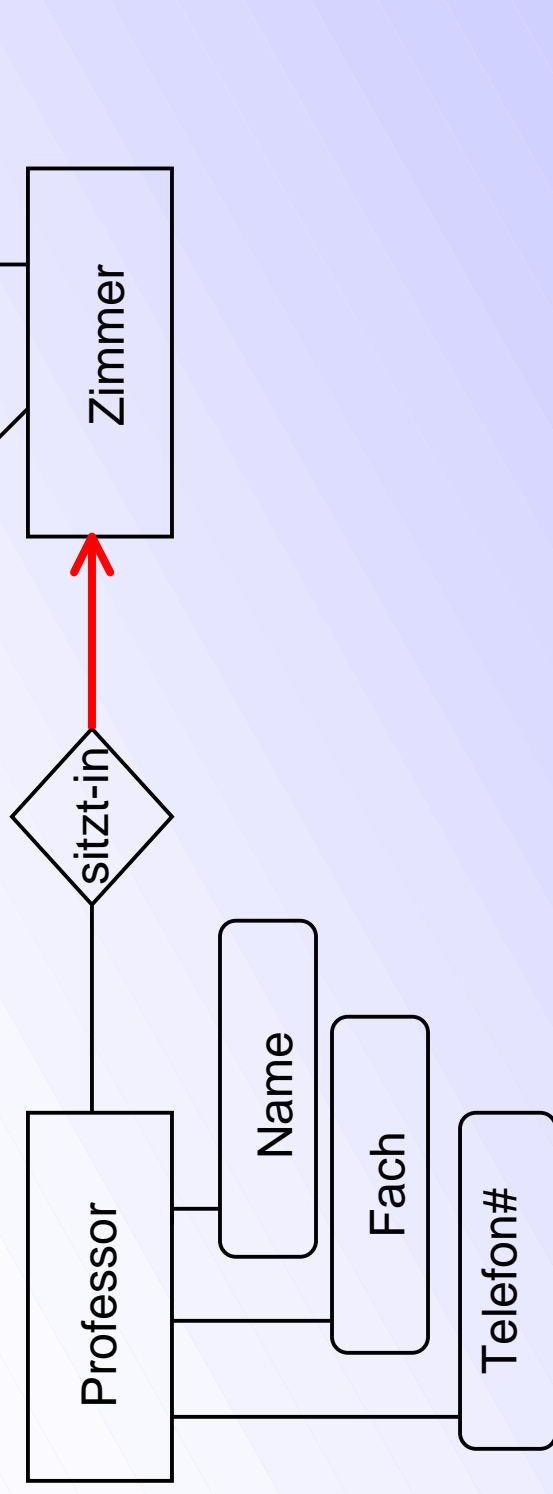
Vorlesung	Buch
DB 1	1-234...
DB 2	9-876...
DB 1	9-876...
DB3	4-242...

Funktionale Beziehungen

Textuell:

$R : E_1 \rightarrow E_2$

Graphisch:



Bedeutung:

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



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 :

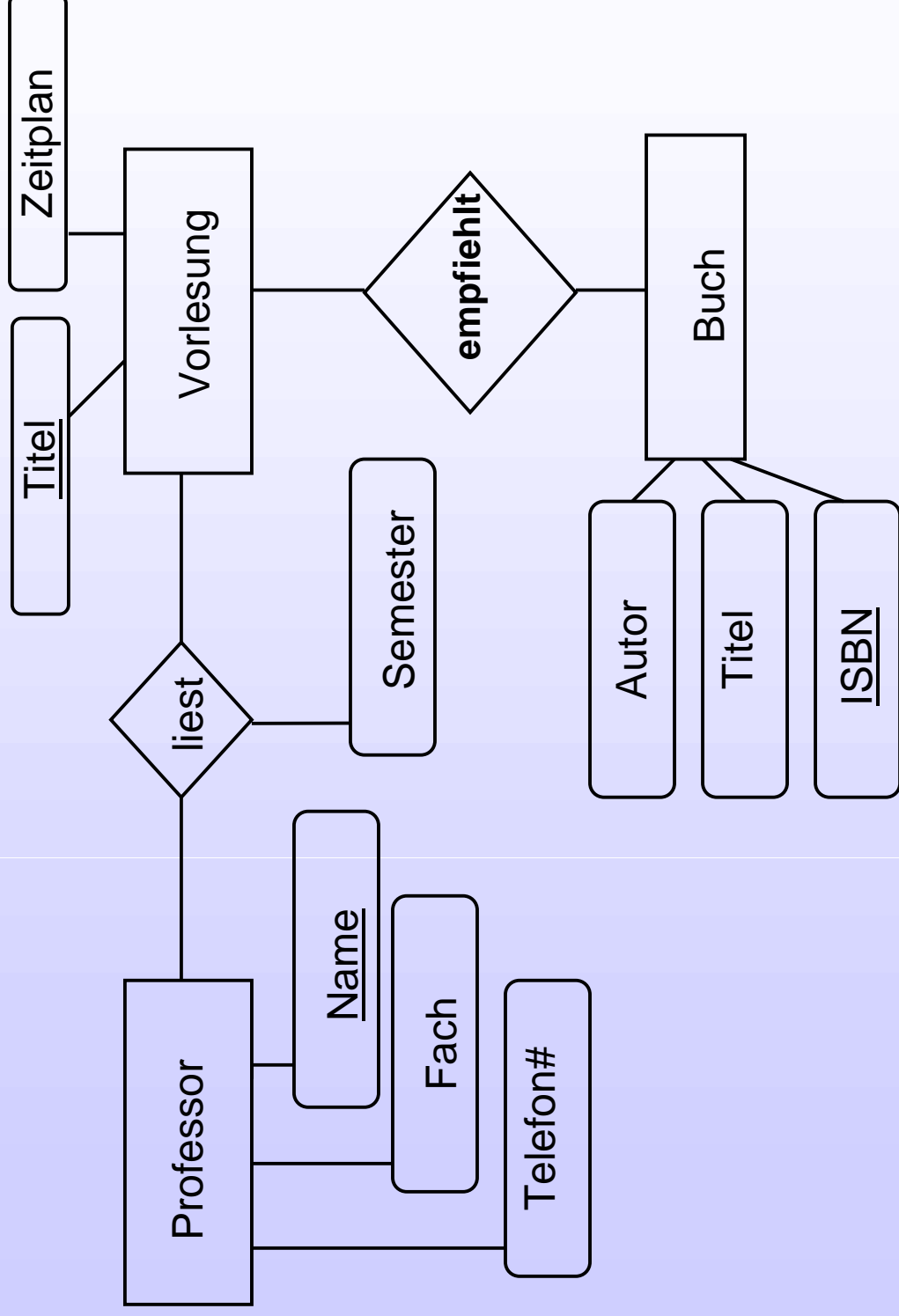
$$\forall e_1, e_2 \in \sigma(E) : (\sigma(S_1)(e_1) = \sigma(S_1)(e_2) \wedge \dots \wedge \sigma(S_k)(e_1) = \sigma(S_k)(e_2)) \Rightarrow (e_1 = e_2)$$

Notation: markieren durch Unterstreichung:

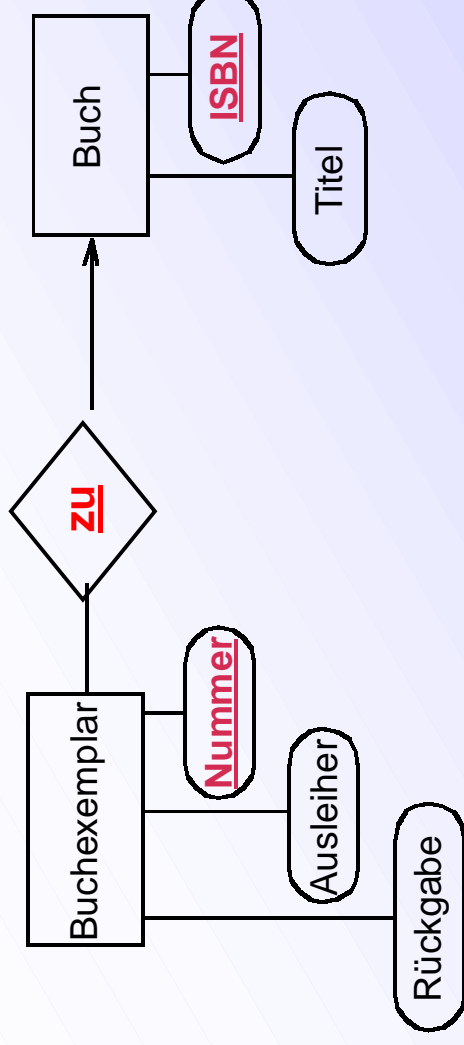
$$E(\dots, \underline{S}_1, \dots, \underline{S}_i, \dots)$$



Beispiel mit Schlüsselattributen

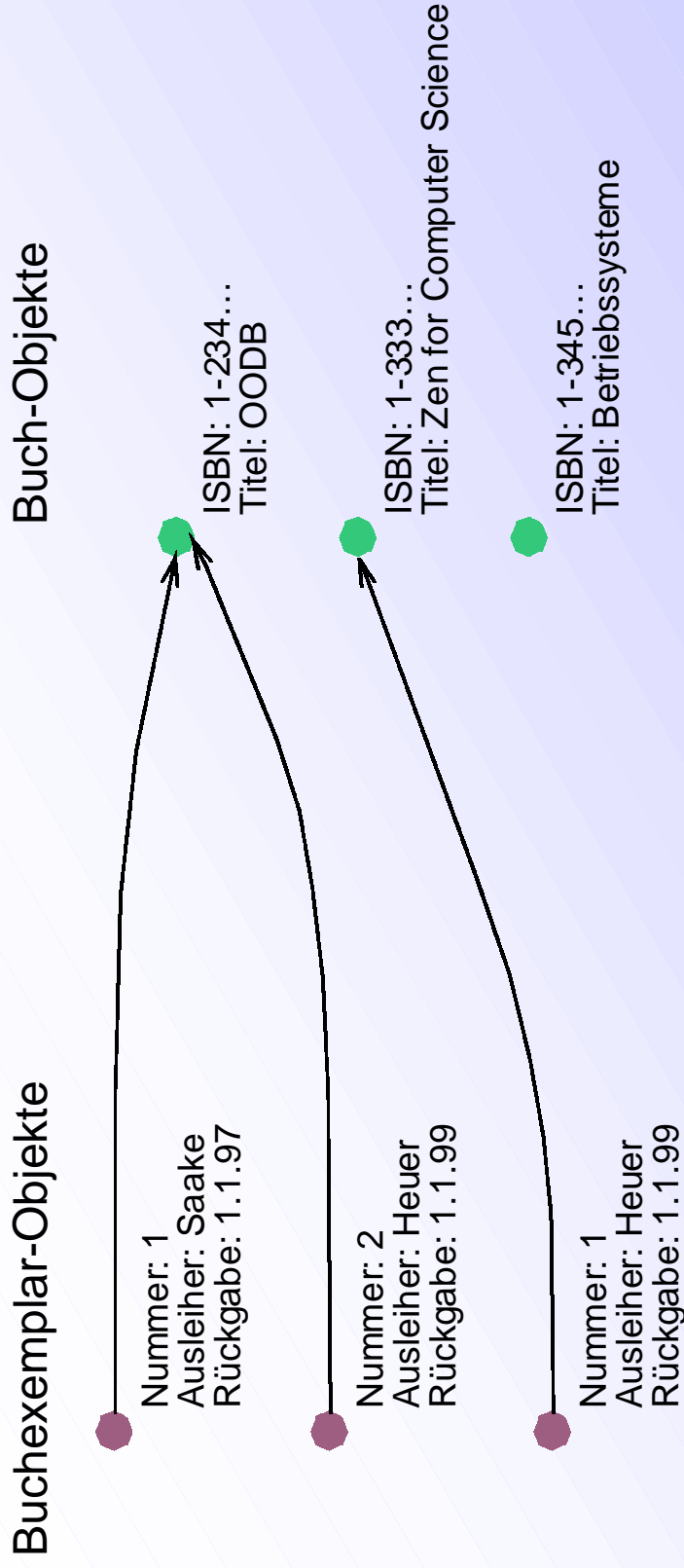


Abhängige Entity-Typen und Schlüsselattribute



- ☐ unterstrichene Attribute/funktionale Bez. sind die **Schlüssel** eines Entity-Typs:
 - ⇒ Buch hat den Schlüssel ISBN
 - ⇒ Buchexemplar hat den Schlüssel Nummer, zu - Verweis auf Buch
- ☐ Schlüsselattribute zweier Entities desselben Typs haben verschiedene Werte
- ☐ funktionale Beziehung als Schlüsselbestandteil legt **Existenzabhängigkeit** fest:
 - ⇒ Buchexemplar ohne zugehöriges Buch ist nicht erlaubt (= **weak Entity**)

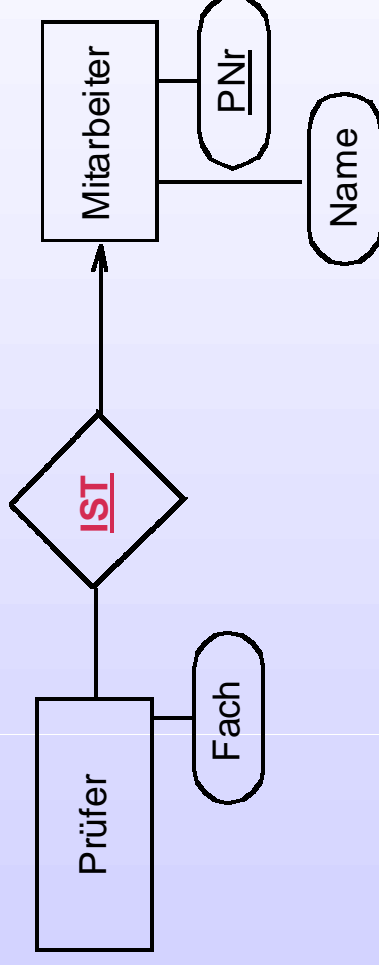
Ausprägungen abhängiger Entity-Typen:



Nicht erlaubt wäre z.B.:

- beide Buchexemplare zu OODB-Titel haben dieselbe Nummer 1
- Buchexemplar hat mehr als ein oder kein zugehöriges Buch

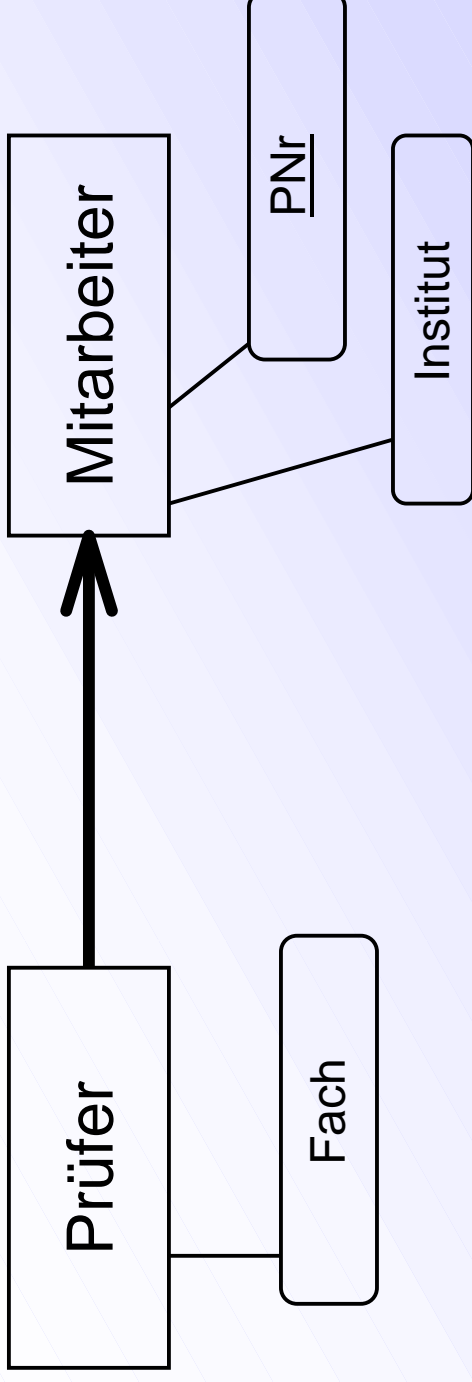
Die IST-Beziehung als Spezialfall einer funktionalen Beziehung



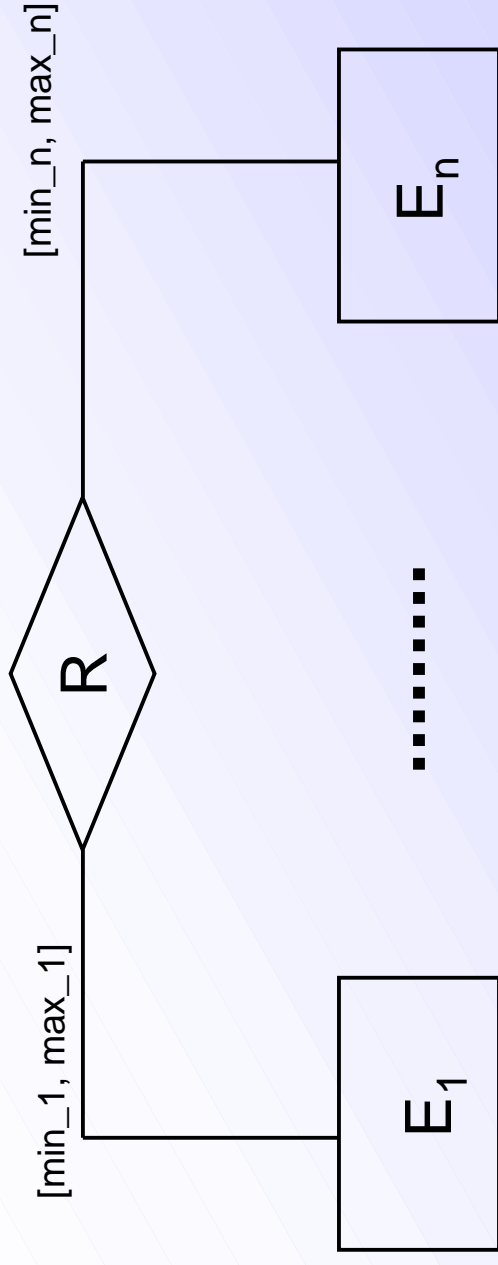
- ❑ jede **Prüfer** -Instanz ist genau einer **Mitarbeiter** -Instanz zugeordnet (Spezialfall eines abhängigen Entity-Typs)
- ❑ nicht jeder **Mitarbeiter** ist zugleich **Prüfer**, aber jeder **Prüfer** ist **Mitarbeiter**!
- ❑ **IST**-Beziehungstyp beschreibt also Mengeninklusion $\sigma(\mathbf{E}_1) \subseteq \sigma(\mathbf{E}_2)$
- ❑ Stichwort: semantische Vererbung (siehe TSE I)
- ❑ Attribute des Entity-Typs **Mitarbeiter** treffen auch auf **Prüfer** zu („vererbte Attribute“):
Prüfer(Name, PNr, Fach)
- ❑ nicht nur Attributdeklarationen, sondern auch konkrete Werte vererben sich



Die IST-Beziehung: Alternative Notation

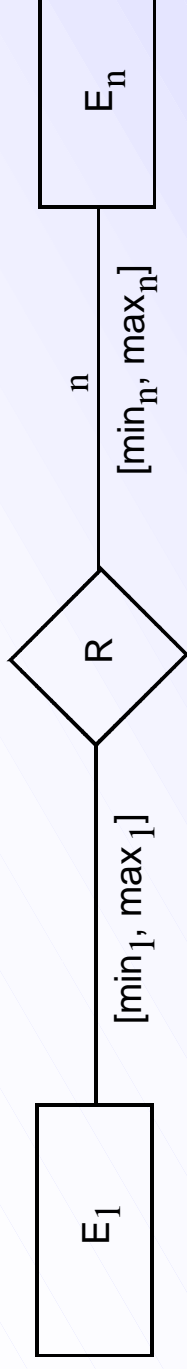


Kardinalitäten



- Notation für Kardinalitätsangaben an einem Beziehungstyp
 $R(E_1, \dots, E_i[*min_i, max_i*], \dots, E_n)$
- Spezielle Wertangabe für max_i ist *

2 verschiedene Semantikdefinitionen für Kardinalitäten



mit $\min_i, \max_i \in 0, 1, \dots$ und * für beliebig viele

□ Semantikmöglichkeit 1 (für den Rest der Vorlesung) :

die Anzahl von Beziehungen des Relationship-Typs R mit derselben Instanz e_i vom Typ E_i bewegt sich zwischen \min_i und \max_i :)

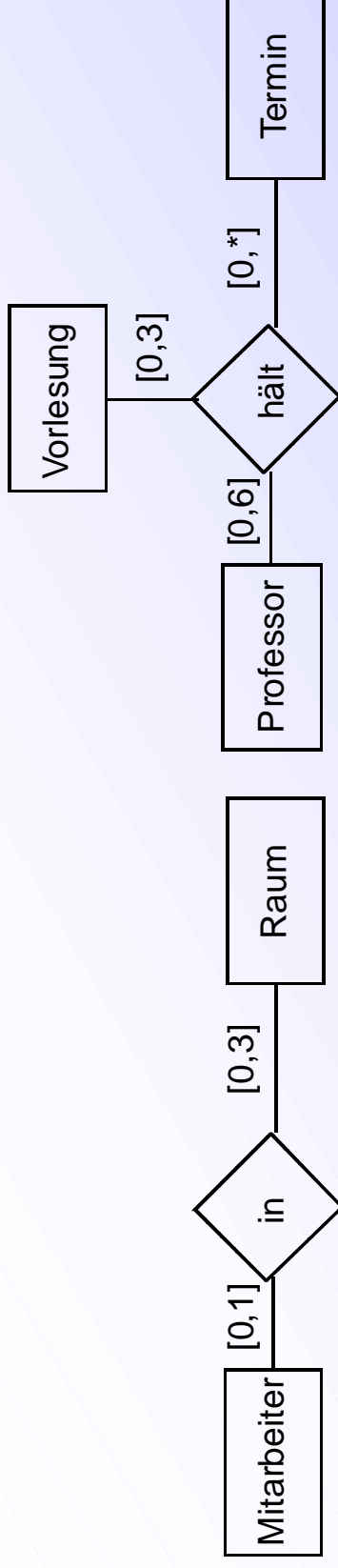
$$\forall e_i \in E_i : \min_i \leq |\{ r \in R \mid r.E_i = e_i \}| \leq \max_i$$

□ Semantikmöglichkeit 2 :

die Anzahl von Beziehungen des Relationship-Typs R , bei denen die Instanzen zu allen Entity-Typen bis auf E_i festgelegt sind, bewegt sich zwischen \min_i und \max_i :

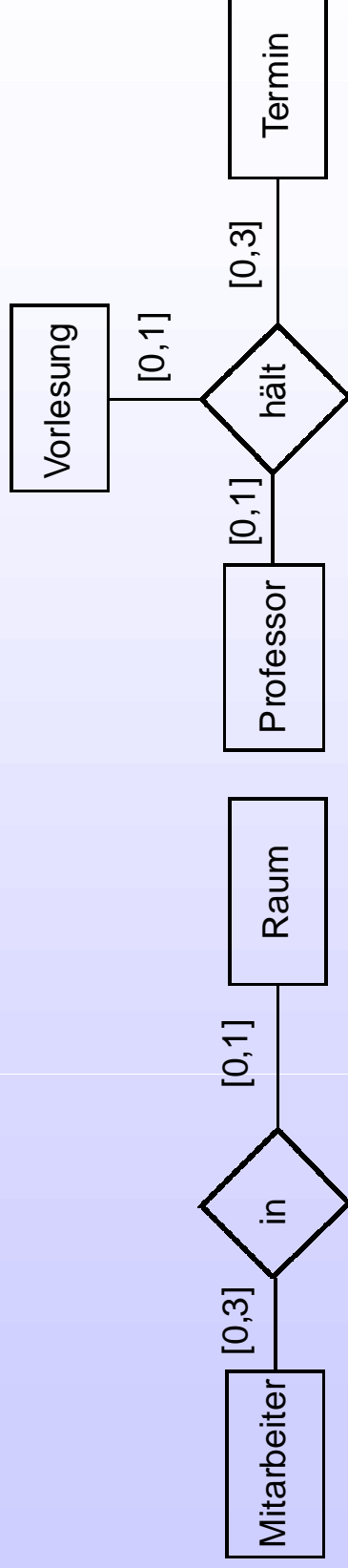
$$\forall i \in 1, \dots, n : \forall (e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_n) \in E_1 \times \dots \times E_{i-1} \times E_{i+1} \times \dots \times E_n : \\ \min_i \leq |\{ r \in R \mid \forall j \in 1, \dots, n : j \neq i \wedge r.E_j = e_j \}| \leq \max_i$$

Beispiele für Kardinalitätsangaben mit Semantikmöglichkeit 1



- ein Mitarbeiter arbeitet in höchstens einem Raum (der Uni)
- in einem Raum arbeiten höchstens 3 Mitarbeiter
- ein Professor hat maximal 6 Vorlesungstermine (pro Woche)
- eine Vorlesung wird an maximal drei Terminen (pro Woche) gehalten
- an einem Termin können beliebig viele Vorlesungen gehalten werden

Beispiele für Kardinalitätsangaben mit Semantikmöglichkeit 2

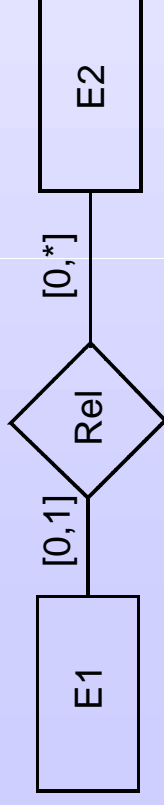


- ❑ ein Mitarbeiter arbeitet in höchstens einem Raum (der Uni)
- ❑ in einem Raum arbeiten höchstens 3 Mitarbeiter
- ❑ ein Professor hält an einem Termin höchstens eine Vorlesung
- ❑ eine Vorlesung wird an einem Termin höchstens von einem Professor gelesen
- ❑ ein Professor hält eine Vorlesung an maximal drei Terminen (pro Woche)

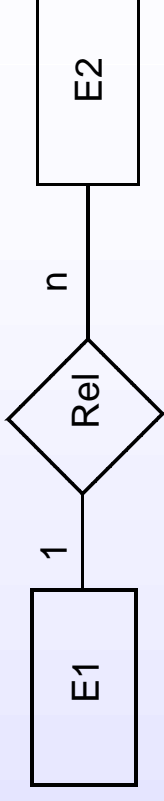


Vereinfachte Kardinalitätsangaben für binäre Beziehungen

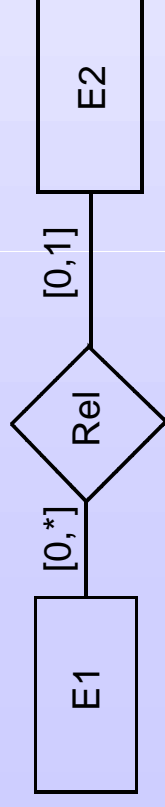
Kardinalitätsangabe nach Semantik 1:



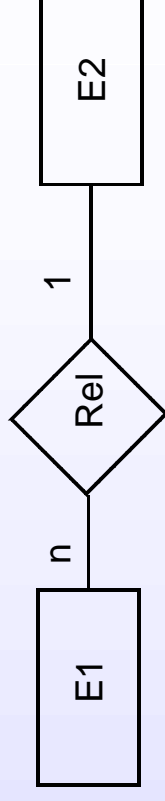
abkürzende Notation (Semantik 1):



Kardinalitätsangabe nach Semantik 2:

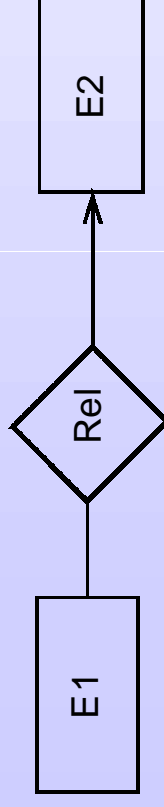


abkürzende Notation (Semantik 2):



(Standard-ER-Notation)

oder im ER-Modell ohne Kardinalitäten:



Fazit:

Vorsicht bei Kardinalitätsangaben in verschiedenen ER-Dialekten!

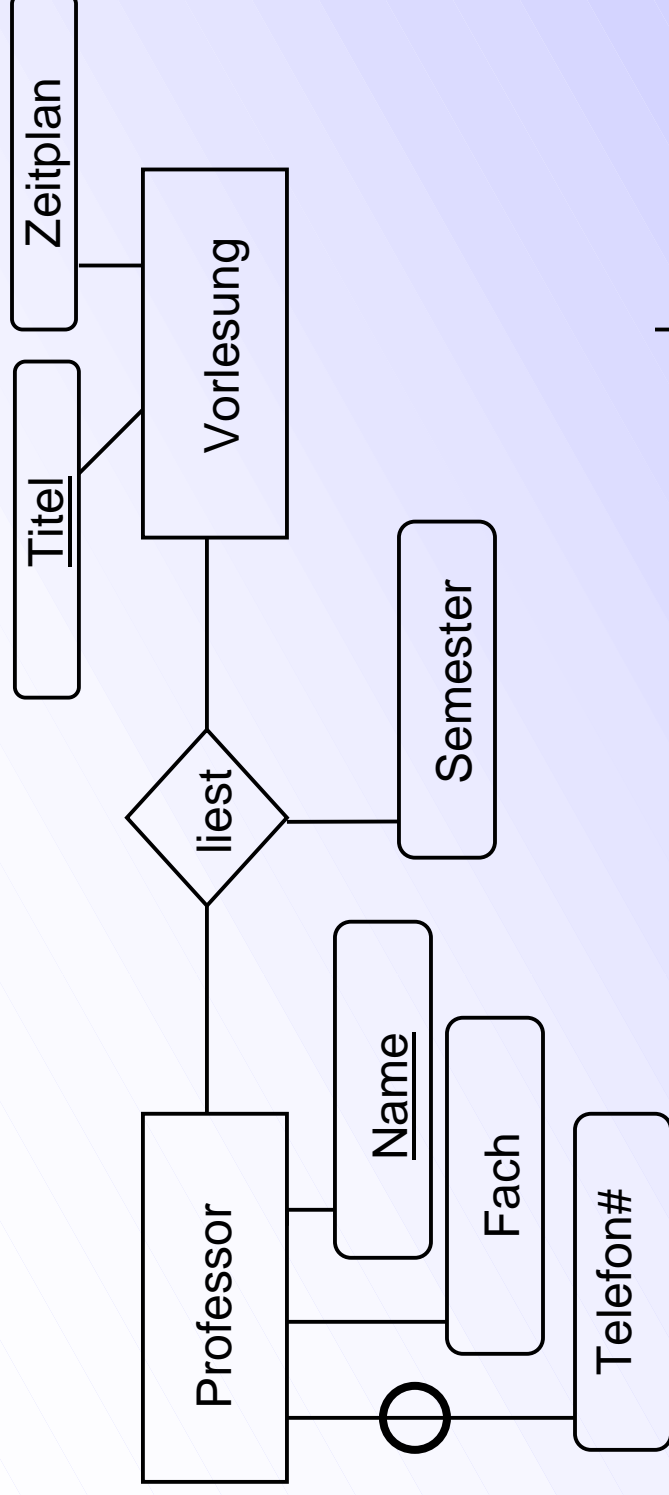


Kardinalitäten

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



Optionalität von Attributen und Beziehungen

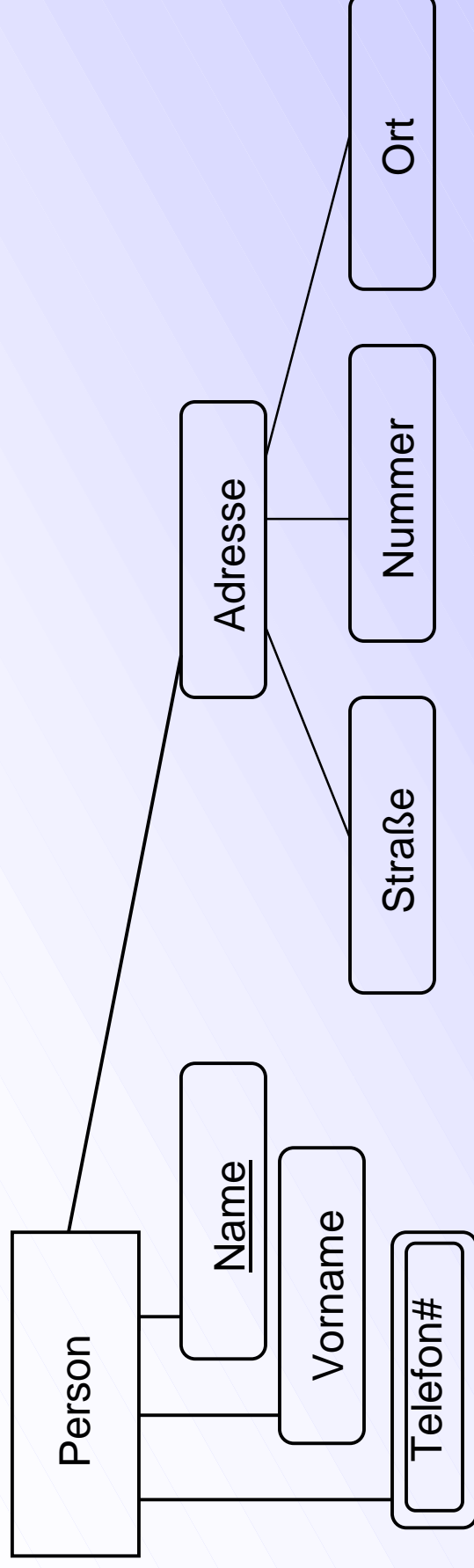


Optionale Attribute im ER-Modell



Weitere Konzepte

□ Strukturierte Attributwerte im ER-Modell





Strukturierte Werte im ER-Modell

Mengen- und tupelwertige Attribute im ER-Modell

prod: Tupelbildung

point = prod(real,real)

list: Listen/Folgen von Werten

polygon = list(point)

set: Mengen

holidays = set(date)

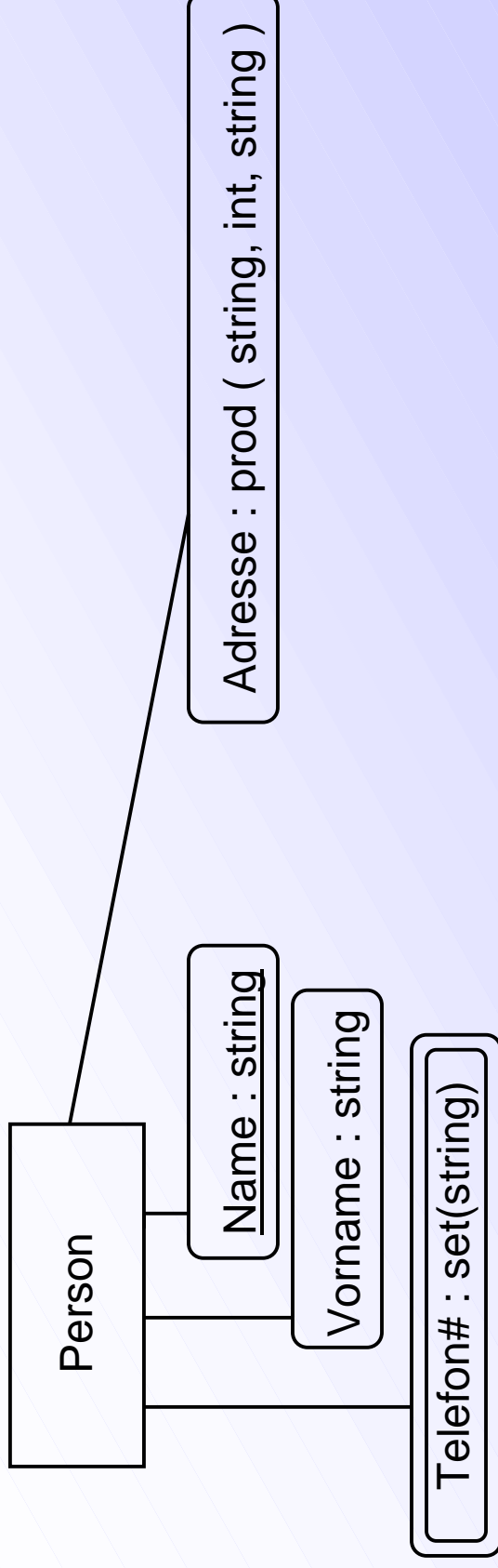
bag: Multimengen

birthdays_of_a_group = bag(date)



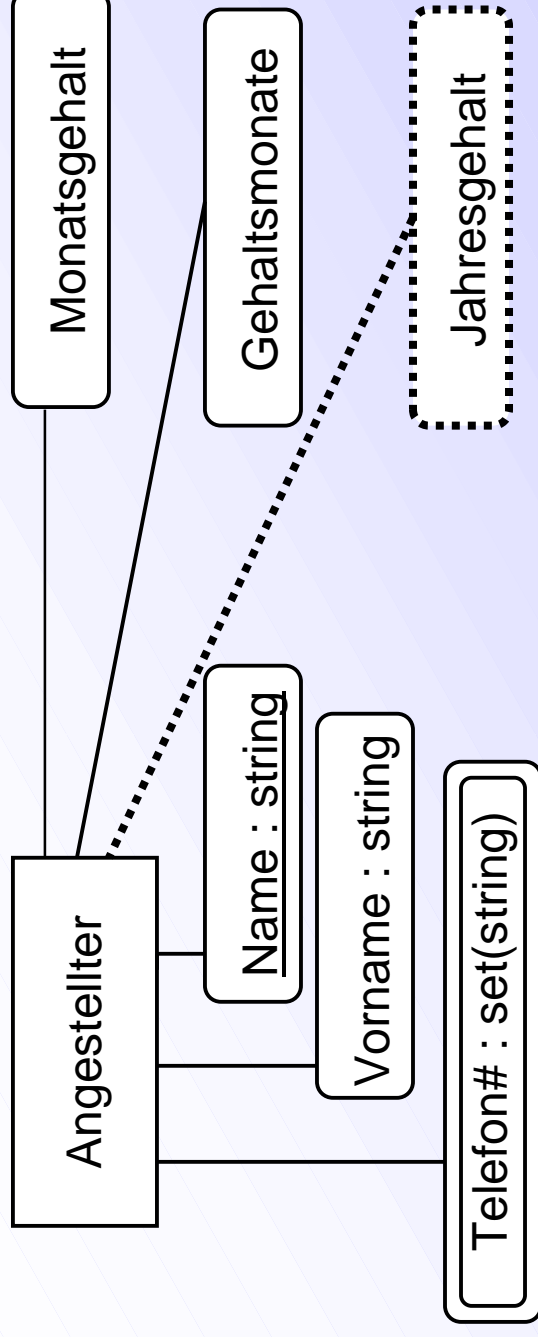
Weitere Konzepte

□ Strukturierte Attributwerte im ER-Modell (alternative Notation)



Weitere Konzepte

□ Abgeleitete Attributwerte im ER-Modell



Jahresgehalt := Monatsgehalt * Gehaltsmonate