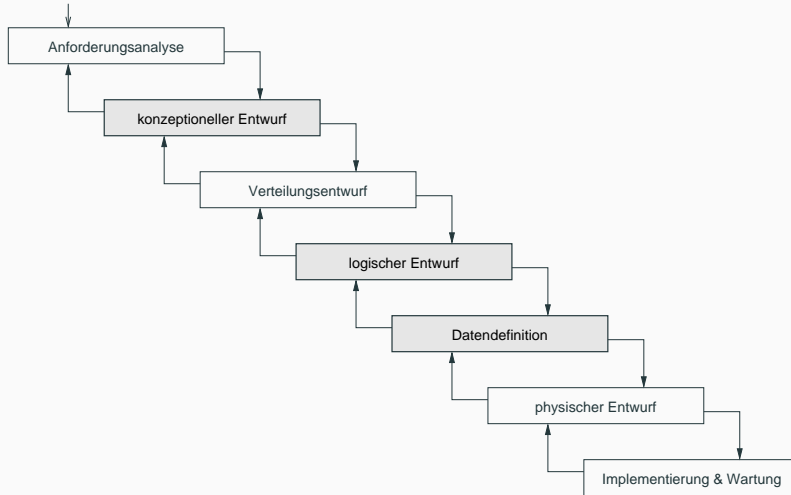


1. Datenbankmodelle für den Entwurf

1. Datenbankmodelle für den Entwurf

- Grundlagen von Datenbankmodellen
- Entity-Relationship-Modell
- Erweiterungen des ER-Modells

Phasen des Datenbankentwurfs



konzeptuelle Datenmodelle:

- implementierungsunabhängiges Modell zur Beschreibung von Objekten und ihren Beziehungen
- Beispiele: **Entity-Relationship-Modell**, UML-Diagramm

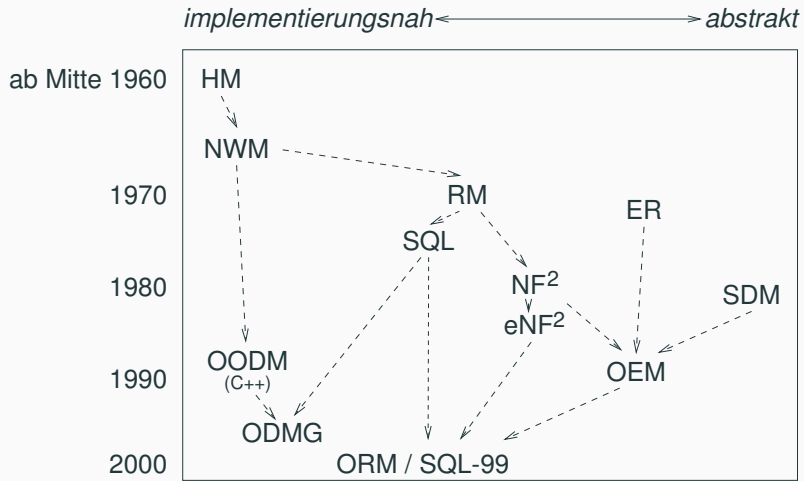
logische Datenmodelle:

- Darstellung von Instanzen, geeignet für Implementierung
- Beispiele: **Relationenmodell**, Hierarchisches Modell, Netzwerkmodell, ...

physische Datenmodelle:

- Speicherungs- und Zugriffstrukturen

Historische Einordnung und Bezüge



Datenbankmodell

Ein Datenbankmodell ist ein System von Konzepten zur Beschreibung von Datenbanken.

- Definition der Datenbankstruktur
- Operatoren zur Abfrage und Änderung von Daten
- Integritätsbedingungen

Datenbankschema (= Datenbankbeschreibung)

- Struktur von Datenobjekten (inkl. Datentypen) und Beziehungen zwischen Objekten
- Integritätsbedingungen

→ ändert sich selten

Datenbankinstanz (Datenbankausprägung, Datenbankzustand)

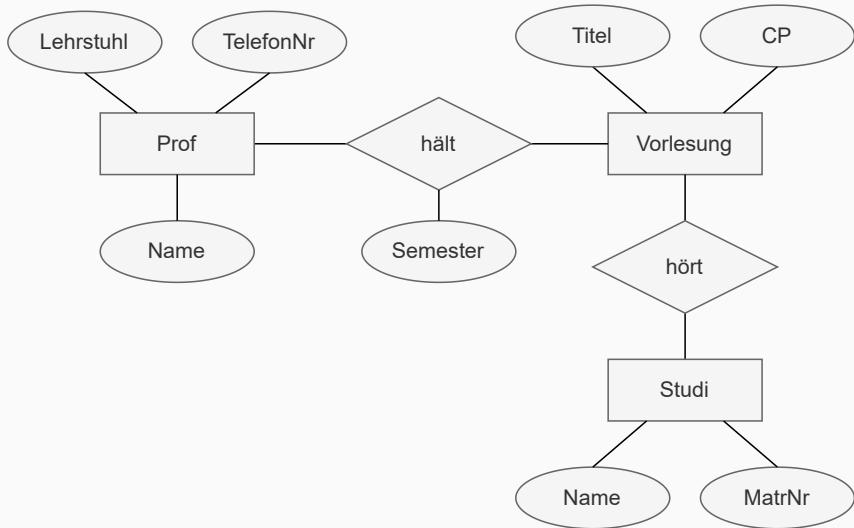
- Daten, die zu einem bestimmten Zeitpunkt in einer Datenbank gespeichert sind

→ ändert sich häufig

- P. P. Chen im Jahre 1976
- *konzeptuelles Datenmodell*
- Hauptbestandteile:
 - Entitäten/Gegenstände (entities)
 - Beziehungen (relationships)
 - Attribute

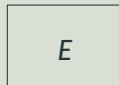
- **Entity:** Objekt der realen oder Vorstellungswelt, über das Informationen zu speichern sind
z.B. ein:e bestimmte:r Professor:in, eine bestimmte Vorlesung
- **Beziehung:** Beziehung zwischen Entities
z.B. ein:e bestimmte:r Professor:in *hält* eine bestimmte Vorlesung
- **Attribut:** Eigenschaften von Entities oder Beziehungen
z.B. Geburtstag, Adresse, Titel, Semester
- **Entity-Typ:** Zusammenfassung von Entities mit gleichen Attributen
z.B. Professor:in, Vorlesung, etc.
- **Beziehungstyp:** Zusammenfassung von Beziehungen
z.B. Professor:in *hält* Vorlesung

Ein einfaches Beispiel



Entities

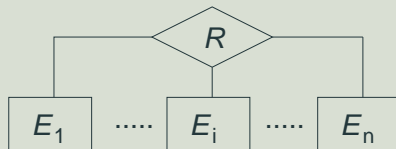
Entity-Typen, etwa E_1, E_2, \dots ; Notation:



- steht für die Menge der *möglichen* Entities vom Typ E
[wird hier nicht festgelegt; aber im Prinzip unendlich viele]
- In einem Datenbank-Zustand wird eine Menge von *aktuellen* Entities vom Typ E gespeichert
- Aktuelle Entities müssen immer mögliche Entities sein;
ferner gefordert: Menge der aktuellen Entities ist endlich

Beziehungen

Beziehungstypen; etwa R_1, R_2, \dots ; Notation n -stelliger Beziehungstypen:



- mögliche Ausprägungen:
jede Kombination möglicher Entities (je eines für jeden beteiligten Entity-Typ)
- aktuelle Beziehungen nur zwischen aktuellen Entities:
also nur zwischen in dem Datenbank-Zustand vorhandenen/gespeicherten Entities

Attribute



Ein Attribut A eines Entity-Typen E ordnet in jedem Datenbank-Zustand jedem aktuellen Entity einen Wert (aus dem Wertebereich D des Attributs) zu. In verschiedenen Datenbank-Zuständen können demselben Entity unterschiedliche Werte zugeordnet werden.

Werte

int : der Wertebereich \mathbb{Z} (die ganzen Zahlen)

string : der Wertebereich \mathcal{C}^* (Folgen von Zeichen aus der Menge \mathcal{C})

⋮

Beziehungsattribute



- Ein Beziehungsattribut A ordnet (in einem Datenbank-Zustand) jeder konkreten aktuellen Beziehung einen Wert zu.
- textuelle Notation für Attribute und Beziehungsattribute:
 - $E(A_1 : D_1, \dots, A_m : D_m)$ oder kurz $E(A_1, \dots, A_m)$
 - $R(E_1, \dots, E_n; A_1, \dots, A_p)$

Identifizierung durch Schlüssel

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

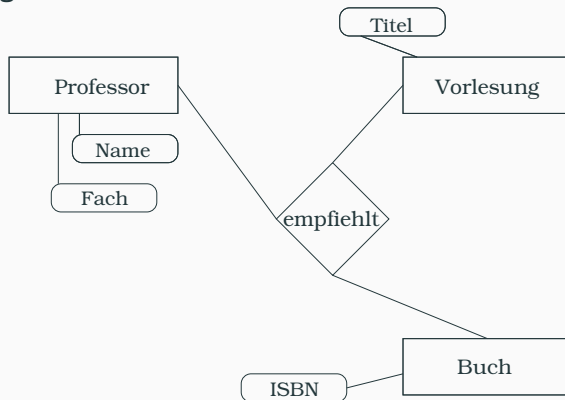
In jedem Datenbankzustand identifizieren die aktuellen Werte der Schlüsselattribute eindeutig Instanzen des Entity-Typs E (**Schlüsseleigenschaft**):

- In jedem möglichen Datenbank-Zustand gilt, dass für jedes Paar von Entities gilt, dass, wenn die beiden Entities in allen Attributwerten der Schlüsselattribute S_1, \dots, S_k übereinstimmen, es dann dasselbe Entity ist.

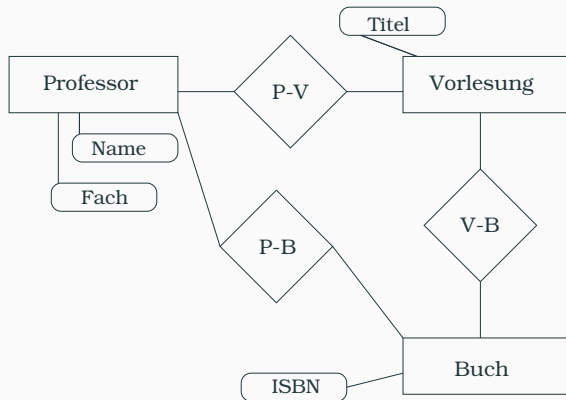
Notation: Markieren durch Unterstreichen:

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

Dreistellige Beziehung:



Mögliche Umwandlung in zweistellige Beziehungen:



Korrekte Ausprägung der dreistelligen Beziehung

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

Stelligkeit von Beziehungen iv

Ausprägungen der drei 2-stelligen Beziehungstypen

P-V	Prof.	Vorl.
	Heuer	DB 1
	Heuer	DB 2
	Saake	DB 1
	Saake	DB 2

P-B	Prof.	Buch
	Heuer	1-234-..
	Heuer	9-876-..
	Saake	9-876-..

V-B	Vorl.	Buch
	DB 1	1-234-..
	DB 2	9-876-..
	DB 1	9-876-..

...entsprechen aber auch:

empfeht	Prof	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-..

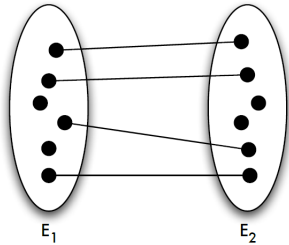
Jetzt außerdem möglich:

P-V	Prof.	Vorl.
	Heuer	DB 1
	Heuer	DB 2
	Saake	DB 1
	Saake	DB 2

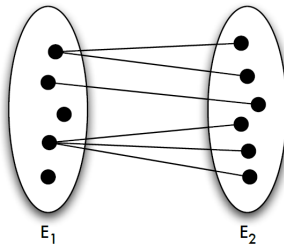
P-B	Prof.	Buch
	Heuer	1-234-..
	Heuer	9-876-..
	Saake	9-876-..

V-B	Vorl.	Buch
	DB 1	1-234-..
	DB 2	9-876-..
	DB 1	9-876-..
	DB 3	4-242-..

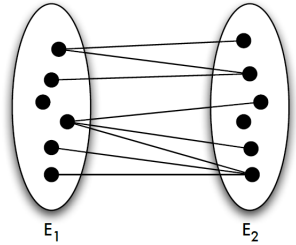
Kardinalitäten von Beziehungen i



1:1-Beziehung

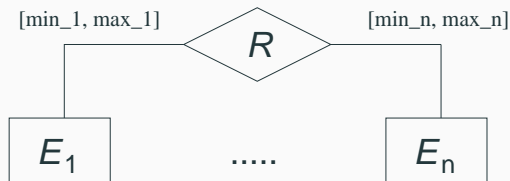


1:n-Beziehung



n:m-Beziehung

Kardinalitäten von Beziehungen ii



	$[min_1, max_1]$	$[min_2, max_2]$
1 : 1	$[0, \mathbf{1}]$	$[0, \mathbf{1}]$
1 : N	$[0, *]$	$[0, \mathbf{1}]$
N : M	$[0, *]$	$[0, *]$

- 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 $*$ (= beliebig)
- $[0, *]$ ist Standardannahme.
- *zwingende Teilnahme:* $min \geq 1$

arbeitet_in(Mitarbeiter:in[0,1], Raum[0,3])

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

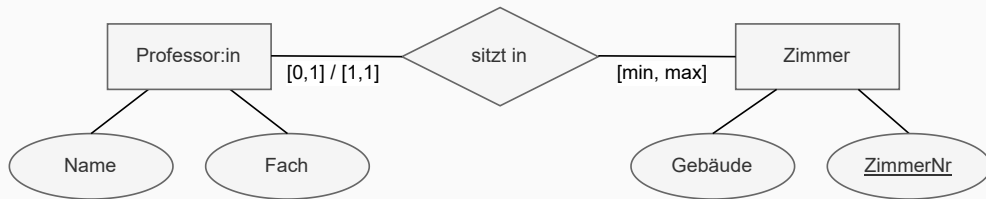
verantwortlich(Mitarbeiter:in[0,],Rechner[1,1])*

- Jedem Rechner ist genau ein:e Mitarbeiter:in zugeordnet, der:die für die Betreuung verantwortlich ist.

Funktionale Beziehung: $R : E_1 \rightarrow E_2$

- Bei n:1-Beziehungen
(bei 1:1-Beziehung liegen 2 funktionale Beziehungen vor)
- **partielle** funktionale Beziehung: durch $R(E_1[0, 1], E_2)$ modelliert
Jede Instanz aus E_1 ist *maximal* einer Instanz aus E_2 zugeordnet.
- **totale** funktionale Beziehung: durch $R(E_1[1, 1], E_2)$ modelliert
Jede Instanz aus E_1 ist *genau* einer Instanz aus E_2 zugeordnet.

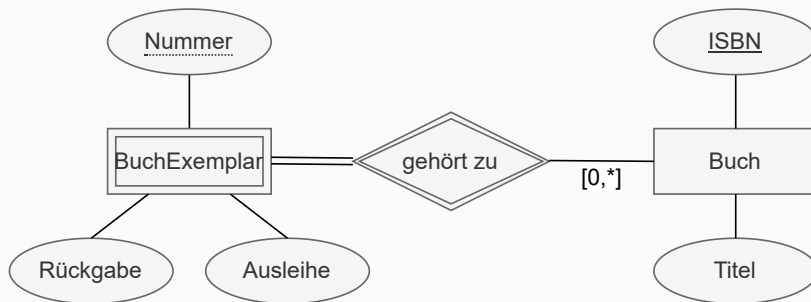
Beispiel: Professor:in \rightarrow Zimmer



- In jedem Datenbank-Zustand wird jeder aktuellen Entity vom Typ E_1 (hier: Professor:in) *maximal/genau* eine aktuelle Entity vom Typ E_2 (hier: Zimmer) zugeordnet.

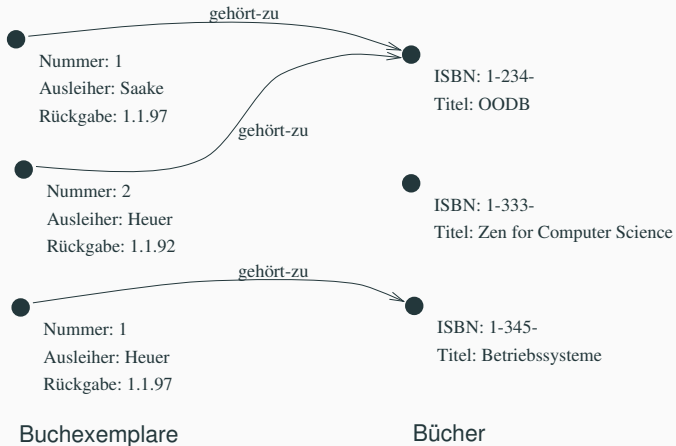
Abhängige Entity-Typen i

- auch *schwache* oder *existenzabhängige* Entity-Typen genannt
- funktionale Beziehung zu einem identifizierenden Entity-Typ
- **partiell**es Schlüsselattribut (gestrichelt unterstrichen)
- Identifikation zusammen mit Schlüssel des identifizierenden Entity-Typ

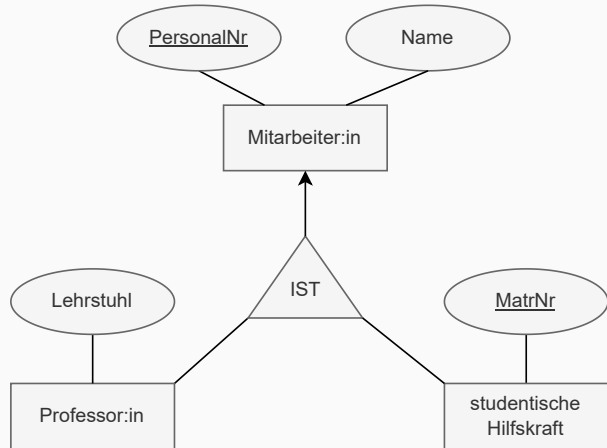


Abhängige Entity-Typen ii

- Mögliche Ausprägung für abhängige Entities



- Spezialfall einer funktionalen Beziehung:
Jede Professor:in-Instanz ist genau einer Mitarbeiter:in-Instanz zugeordnet.
- Aber nicht alle Mitarbeiter:innen sind auch Professor:innen



- Attribute des Obertyps (hier Mitarbeiter:in) vererben sich auf die Untertypen

Professor:in(*Name*, *PersonalNr*_{von Mitarbeiter:in}, *Lehrstuhl*)

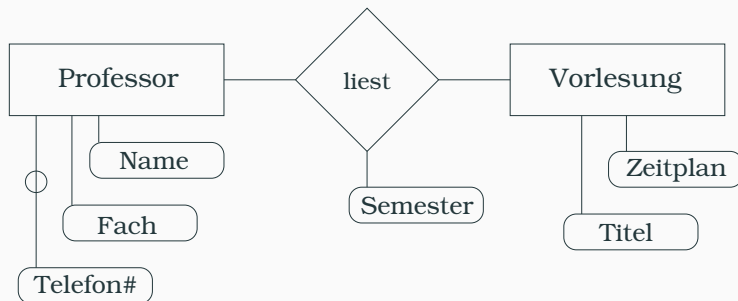
Nicht nur Deklarationen vererben sich, sondern auch aktuelle Werte.

- Die Menge der aktuellen Entities von Professor:in ist immer eine Teilmenge der aktuellen Entities von Mitarbeiter:in.

Für die Beziehung E_1 IST E_2 gilt: $\text{IST}(E_1[1, 1], E_2[0, 1])$

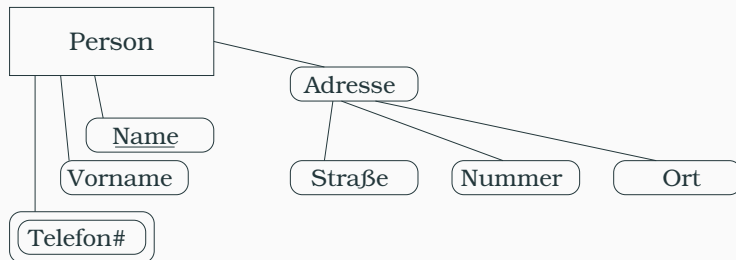
Erweiterungen des ER-Modells (exemplarisch)

- Optionalität von Attributen

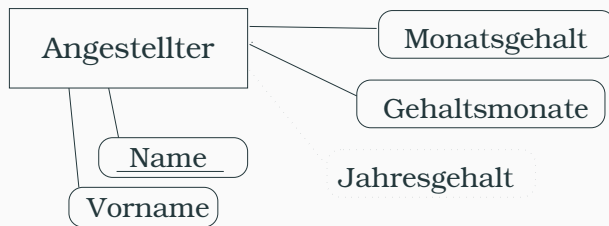


Verwenden wir auch nicht! Kann man anders (besser?) modellieren!

- strukturierte Attributwerte im ER-Modell



- abgeleitete Attributwerte im ER-Modell



$$\text{Jahresgehalt} := \text{Monatsgehalt} * \text{Gehaltsmonate}$$

- 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.

1.3. 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