



### 3. Informationsmodelle

Inhalt  
DB-Entwurf und Modellierung  
Entity-Relationship-Modell (ERM)  
Erweiterungen des ERM  
Kardinalitätsrestriktionen  
Abstraktionskonzepte

---

---

---

---

---

---

---

---



### 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)

- **Schrittweise Ableitung: (Verschiedene Sichten)**
  - Information in unserer Vorstellung
  - Informationsstruktur: Organisationsform der Information
  - Logische Datenstruktur (zugriffspfadunabhängig, Was-Aspekt)
  - Physische Datenstruktur (zugriffspfadabhängig, Was- und Wie-Aspekt)

---

---

---

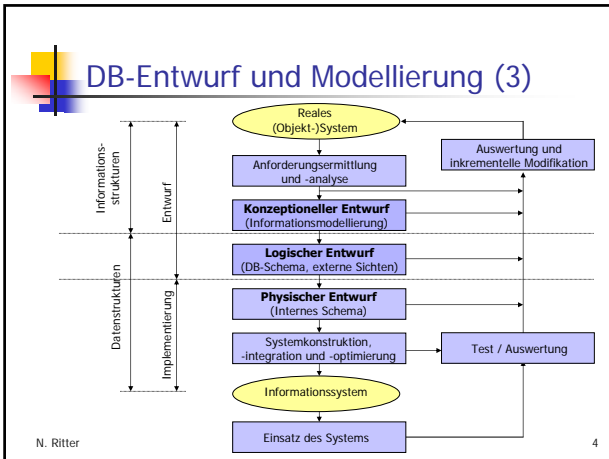
---

---

---

---

---




---

---

---

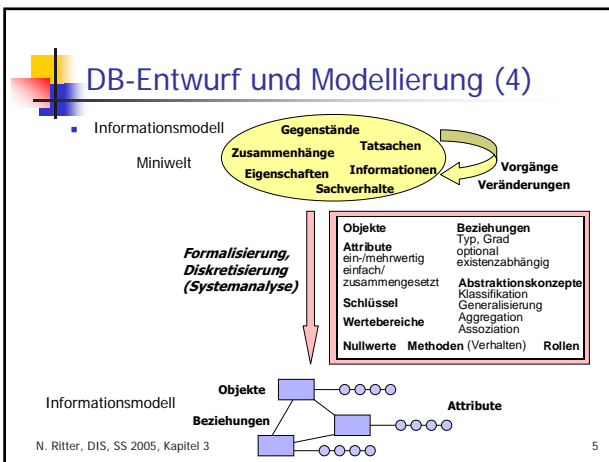
---

---

---

---

---




---

---

---

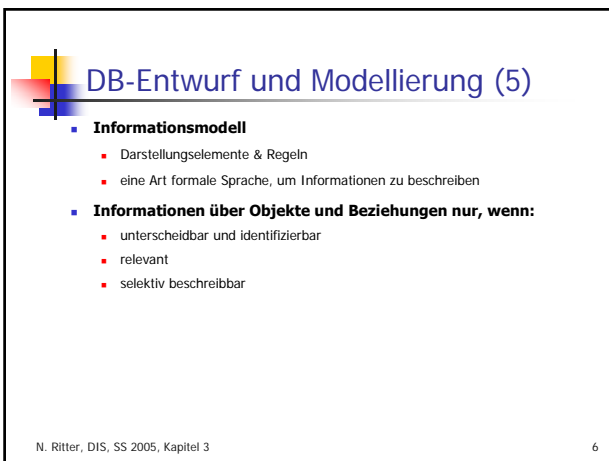
---

---

---

---

---




---

---

---

---

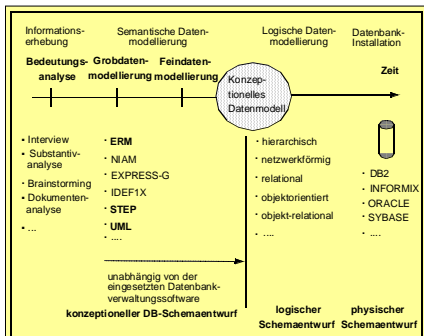
---

---

---

---

## DB-Entwurf und Modellierung (6)



N. Ritter, DIS

7

---

---

---

---

---

---

---

---

---

---

## DB-Entwurf und Modellierung (7)

- **ERM (Entity Relationship Model):**
  - generell einsetzbares Modellierungswerkzeug
- **STEP (STandard for the Exchange of Product Definition Data):**
  - Modellierung, Zugriff, Austausch von **produktdefinierenden Daten** über den gesamten Produktlebenszyklus
- **UML (Unified Modeling Language):**
  - Notation und Sprache zur Unterstützung objektorientierter Softwareentwicklung

N. Ritter, DIS, SS 2005, Kapitel 3

8

---

---

---

---

---

---

---

---

---

---

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

- **Modellierungskonzepte**
  - Entity-Mengen (Objektmengen)
  - Wertebereiche, Attribute
  - Primärschlüssel
  - Relationship-Mengen (Beziehungsmengen)
- **Klassifikation der Beziehungstypen**
  - benutzerdefinierte Beziehungen
  - Abbildungstyp
    - 1 : 1
    - n : 1
    - n : m
  - **Ziel:**
    - Festlegung von semantischen Aspekten
    - explizite Definition von strukturellen Integritätsbedingungen

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

N. Ritter, DIS, SS 2005, Kapitel 3

9

---

---

---

---

---

---

---

---

---

---

## Entity-Relationship-Modell (2)

### Beachte:

- Das ERM modelliert die Typ-, nicht die Instanzebene; 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 oder 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

---

---

---

---

---

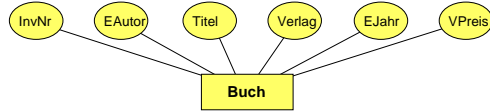
---

---

---

## 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, Oldenburg, ...)  
 e2 = (0815, Date, Introd. To DBS, Addison, ...)  
 ...  
 e3 = (1234, Härder, DBS, Springer, ...)

---

---

---

---

---

---

---

---

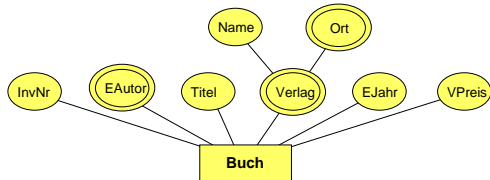
---

---

## 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.)
- $\{A_1, A_2, \dots, A_n\} = \mathbf{A}$  sei Menge der (einwertigen) Attribute zur Entity-Menge E
  - $\mathbf{K} \subseteq \mathbf{A}$  heißt **Schlüsselkandidat** von E
  - $\mathbf{K}$  irreduzibel (minimal) und  
 $\Leftrightarrow e_i, e_j \in E: e_i \neq e_j \rightarrow \mathbf{K}(e_i) \neq \mathbf{K}(e_j)$
- mehrere Schlüsselkandidaten (SK) möglich  $\rightarrow$  **Primärschlüssel** auswählen
- **Beispiel:** Entity-Menge **Student** mit Attributen  
 Matrnr, SVNr, Name, Gebdat, FBnr

---

---

---

---

---

---

---

---

---

---

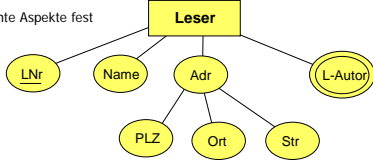
## Entity-Relationship-Modell (8)

### Entity-Deklaration oder Entity-Typ

- legt zeitinvariante Aspekte fest

### Entity-Typ

$E = (X, K)$



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

### Wertebereiche

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

---

---

---

---

---

---

---

---

## 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:

- Einwertige Attribute: A
- Mehrwertige Attribute: {A}
- Zusammengesetzte Attribute: A (B<sub>1</sub>, ..., B<sub>k</sub>)

Definitionen aus:  
G. Vossen: Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme, Oldenbourg, 4. Auflage, 2000.

---

---

---

---

---

---

---

---

## Entity-Relationship-Modell (10)

### Definition Wertebereich/Domain

- $E = (X, K)$  sei ein Entity-Typ und  $attr(E)$  die Menge aller in X vorkommenden Attributnamen. Jedem  $A \in attr(E)$ , das nicht einer Zusammensetzung voransteht, sei ein Wertebereich  $W(A)$  zugeordnet. Für jedes  $A \in attr(E)$  sei  
 $dom(A) := W(A)$ , falls A einwertig;  
 $dom(A) := 2^{W(A)}$ , falls A mehrwertig;  
 $dom(A) := W(B_1) \times \dots \times W(B_k)$ ,  
 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  $R = (Ent, Y)$ . Dabei ist  $R$  der Name des Typs,  $Ent$  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  $Ent = (E_1, \dots, E_k)$ , 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.  
 $r \in E_1^t \times \dots \times E_k^t \times \text{dom}(B_1) \times \dots \times \text{dom}(B_n)$  bzw.  
 $r = (e_1, \dots, e_k, b_1, \dots, b_n)$  mit  
 $e_i \in E_i^t$  für  $1 \leq i \leq k$  und  $b_j \in \text{dom}(B_j)$  für  $1 \leq j \leq n$ .
- Eine Relationship-Menge  $R^t$  (zur Zeit  $t$ ) ist eine Menge von Relationships, d.h.,  
 $R^t \subseteq E_1^t \times \dots \times E_k^t \times \text{dom}(B_1) \times \dots \times \text{dom}(B_n)$ .

---

---

---

---

---

---

---

---

---

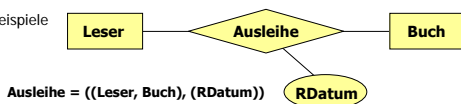
---

## Entity-Relationship-Modell (13)

### Eigenschaften von Relationship-Mengen

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

### Beispiele



### Eigenschaften:

- Grad: 2
- Existenzabhängigkeit: Nein
- Beziehungstyp: n:m

---

---

---

---

---

---

---

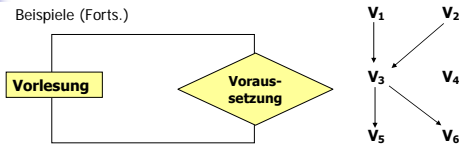
---

---

---

## Entity-Relationship-Modell (14)

- Beispiele (Forts.)



**Voraussetzung** = ((**Vorgänger/Vorlesung**, **Nachfolger/Vorlesung**), ( $\emptyset$ ))  
 genauer: direkte Voraussetzung  
**Eigenschaften:**  
 Grad: **1**  
 Existenzabhängigkeit: **Nein**  
 Beziehungstyp: **n:m**

Transitivität gilt im Allg. bei Selbstreferenz nicht (Beispiel: „liebt“ auf „Person“).  
 Keine Disjunktheit der an einer Relationship-Menge beteiligten Entity-Mengen gefordert.

---

---

---

---

---

---

---

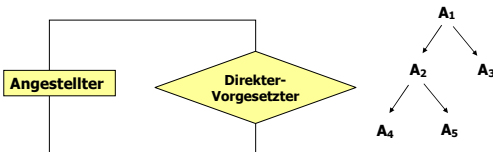
---

---

---

## Entity-Relationship-Modell (15)

- Beispiele (Forts.)



**Direkter Vorgesetzter** =  
 ((**Angestellter/Angestellter**, **Chef/Angestellter**), ( $\emptyset$ ))  
**Eigenschaften:**  
 Grad: **1**  
 Existenzabhängigkeit: **Nein**  
 Beziehungstyp: **1:n**

---

---

---

---

---

---

---

---

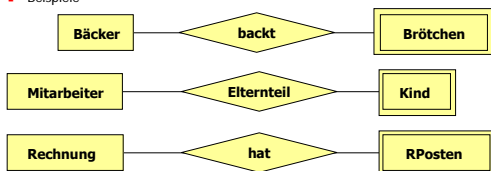
---

---

## 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.

---

---

---

---

---

---

---

---

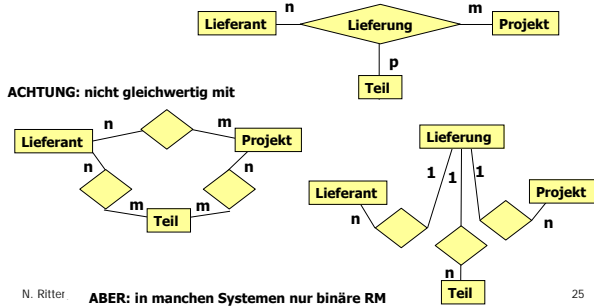
---

---



## Entity-Relationship-Modell (17)

- 3-stellige Relationship-Mengen




---

---

---

---

---

---

---

---

---

---

## 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

---

---

---

---

---

---

---

---

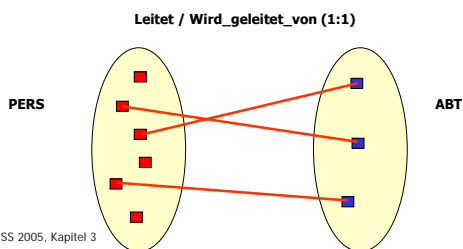
---

---

## Entity-Relationship-Modell (19)

- Klassifikation von Datenabbildungen (Forts.)

- Beispiele




---

---

---

---

---

---

---

---

---

---

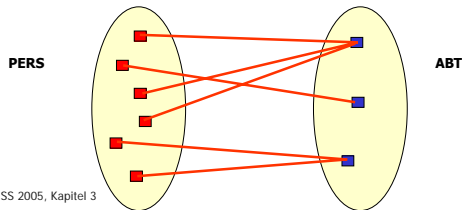
## Entity-Relationship-Modell (20)

### Klassifikation von Datenabbildungen (Forts.)

#### Beispiele (Forts.)



Arbeitet\_fuer / Hat\_Mitarbeiter (n:1)



N. Ritter, DIS, SS 2005, Kapitel 3

28

---

---

---

---

---

---

---

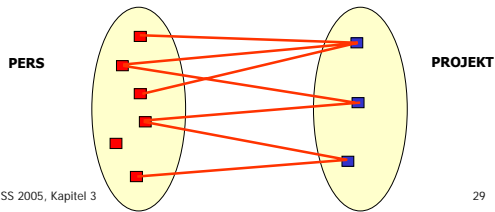
---

## Entity-Relationship-Modell (21)

### Klassifikation von Datenabbildungen (Forts.)

#### Beispiele (Forts.)

Arbeitet\_fuer / Mitarbeit (n:m)



N. Ritter, DIS, SS 2005, Kapitel 3

29

---

---

---

---

---

---

---

---

## Entity-Relationship-Modell (22)

### Beispiel 1

DECLARE VALUE-SETS	REPRESENTATION	ALLOWABLE-VALUES
PERSONAL-NR	INTEGER(5)	(1,10000)
VORNAMEN	CHARACTER(15)	ALL
NACHNAMEN	CHARACTER(25)	ALL
BERUFE	CHARACTER(25)	ALL
PROJEKT-NR	INTEGER(3)	(1,5000)
ANZ.-JAHRE	INTEGER(3)	(0,100)
ORTE	CHARACTER(15)	ALL
PROZENT	FIXED(5.2)	(0,100.00)
ANZ.-MONATE	INTEGER(3)	(0,100)

N. Ritter, DIS, SS 2005, Kapitel 3

30

---

---

---

---

---

---

---

---

## Entity-Relationship-Modell (23)

- Beispiel 1 (Forts.)

```
DECLARE REGULAR ENTITY RELATION PERSONAL  
ATTRIBUTE/VALUE-SET:  
  PNR/PERSONAL-NR  
  NAME/(VORNAMEN,NACHNAMEN)  
  KÜNSTLER-NAME/(VORNAMEN, NACHNAMEN)  
  BERUF/BERUFE  
  ALTER/ANZ.-JAHRE  
PRIMARY KEY:  
  PNR
```

---

---

---

---

---

---

---

---

---

---

## Entity-Relationship-Modell (24)

- Beispiel 1 (Forts.)

```
DECLARE REGULAR ENTITY RELATION PROJEKT  
ATTRIBUTE/VALUE-SET:  
  PRO-NR/PROJEKT-NR  
  PRO-ORT/ORTE  
PRIMARY KEY:  
  PRO-NR  
DECLARE RELATIONSHIP RELATION PROJEKT-MITARBEIT  
ROLE/ENTITY-RELATION.PK/MAX-NO-OF-ENTITIES  
  MITARBEITER/PERSONAL.PK/n  
  PROJEKT /PROJEKT.PK/m  
ATTRIBUTE/VALUE-SET:  
  ARBEITSZEITANTEIL/PROZENT  
  DAUER/ANZ.-MONATE
```

---

---

---

---

---

---

---

---

---

---

## Entity-Relationship-Modell (25)

- Beispiel 1 (Forts.)

```
DECLARE RELATIONSHIP RELATION PERS.-ANGEHÖRIGE  
ROLE/ENTITY-RELATION.PK/MAX-NO-OF-ENTITIES  
  UNTERHALTSPFLICHTIGER/PERSONAL.PK/1  
  KIND/ KINDER.PK/n  
EXISTENCE OF KIND DEPENDS ON  
EXISTENCE OF UNTERHALTSPFLICHTIGER  
DECLARE WEAK ENTITY RELATION KINDER  
ATTRIBUTE/VALUE-SET:  
  NAME/VORNAMEN  
  ALTER/ANZ.-JAHRE  
PRIMARY KEY:  
  NAME  
  PERSONAL.PK THROUGH PERS-ANGEHÖRIGE
```

---

---

---

---

---

---

---

---

---

---

## Entity-Relationship-Modell (26)

- 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 einen Professor und **halten** Übungen, die zu den entsprechenden Vorlesungen **gehören**.
    - Mehrere Studenten **hören** jeweils eine Reihe von Vorlesungen. Übungen und Vorlesungen werden jeweils von mehreren Studenten **besucht**.
    - Der Besuch von Vorlesungen **setzt** i. allg. die Kenntnis anderer Vorlesungen **voraus**.

---

---

---

---

---

---

---

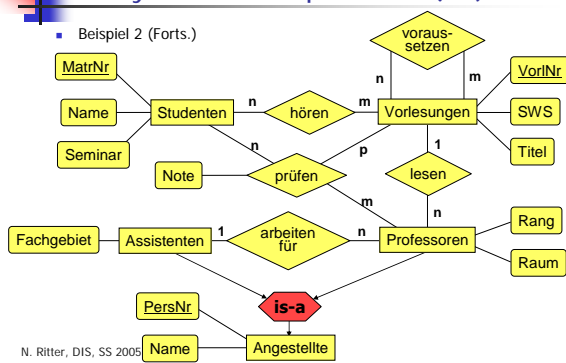
---

---

---

## Entity-Relationship-Modell (27)

- Beispiel 2 (Forts.)




---

---

---

---

---

---

---

---

---

---

## Erweiterungen des ERM

- Ziel: Genauere Modellierung von Beziehungen
  - Verfeinerung der Abbildungen von Beziehungen durch **Kardinalitätsrestriktionen**
  - Ausprägungen (Objekte) einer EM sollen im Modell explizit dargestellt werden; gleichartige Darstellung von Ausprägung und Typ (EM)
  - Einführung von systemkontrollierten Beziehungen (**Abstraktionskonzepte**)

---

---

---

---

---

---

---

---

---

---

## 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



N. Ritter, DIS, SS 2005, Kapitel 3

37

---

---

---

---

---

---

---

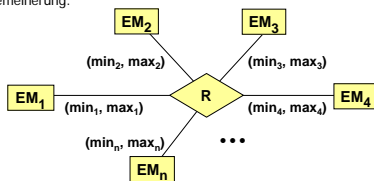
---

---

---

## 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  $(\dots, e_i, \dots)$

N. Ritter, DIS, SS 2005, Kapitel 3

38

---

---

---

---

---

---

---

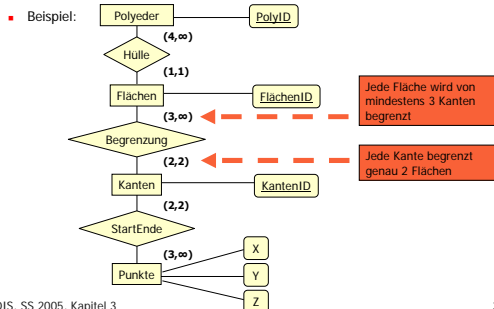
---

---

---

## Kardinalitätsrestriktionen (3)

- Verfeinerung der Datenabbildung (Forts.)



N. Ritter, DIS, SS 2005, Kapitel 3

39

---

---

---

---

---

---

---

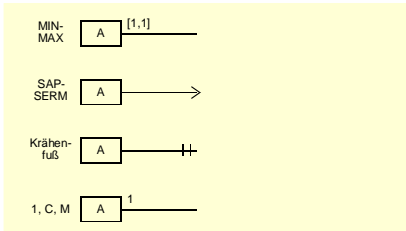
---

---

---

## Kardinalitätsrestriktionen (4)

- Verfeinerung der Datenabbildung (Forts.)
  - Notationen (beachte: viele Systeme erlauben nur 0, 1, n)
    - Jedes Element von A nimmt an genau einer Beziehung teil




---

---

---

---

---

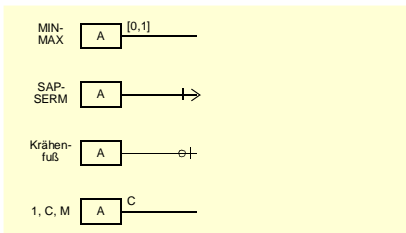
---

---

---

## Kardinalitätsrestriktionen (5)

- Verfeinerung der Datenabbildung (Forts.)
  - Notationen (Forts.)
    - Jedes Element von A nimmt an höchstens einer Beziehung teil




---

---

---

---

---

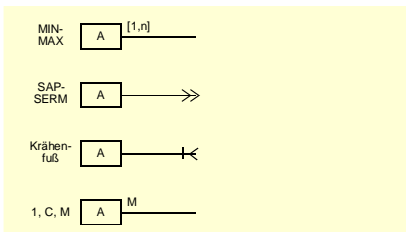
---

---

---

## Kardinalitätsrestriktionen (6)

- Verfeinerung der Datenabbildung (Forts.)
  - Notationen (Forts.)
    - Jedes Element von A nimmt an mindestens einer Beziehung teil




---

---

---

---

---

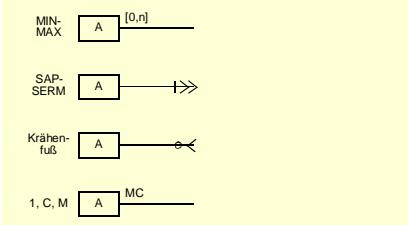
---

---

---

## Kardinalitätsrestriktionen (7)

- Verfeinerung der Datenabbildung (Forts.)
  - Notationen (Forts.)
    - Jedes Element von A kann an beliebig vielen Beziehungen teilnehmen




---

---

---

---

---

---

---

---

---

---

## Abstraktionskonzepte (1)

- Ziel:
  - Erfassung von noch mehr Semantik aus der Miniwelt durch das ERM
  - Entwicklung von (Beschreibungs-)Modellen zur adäquateren Wiedergabe der ausgewählten Miniwelt (Diskursbereich)
  - Definition von **systemkontrollierten Beziehungen**
- Aufgabe:
  - Identifikation von wesentlichen Konstrukten, die vom Menschen angewendet werden, wenn er seinen Diskursbereich beschreibt.
    - Anwendung von **Abstraktion**, um die Information zu organisieren: **"abstraction permits someone to suppress specific details of particular objects emphasizing those pertinent to the actual view"**

---

---

---

---

---

---

---

---

---

---

## Abstraktionskonzepte (2)

- Zwei Typen von Abstraktionen
  - von einfachen zu zusammengesetzten Objekten (*1-Ebenen-Beziehung*)
  - von zusammengesetzten zu (komplexer) zusammengesetzten Objekten (*n-Ebenen-Beziehungen*)
- Abstraktionskonzepte werden vor allem eingesetzt
  - zur *Organisation der Information* und damit auch
  - zur *Begrenzung des Suchraumes* beim Retrieval sowie
  - zu *systemkontrollierten Ableitungen* (Reasoning)
- **Unterschiedliche Konzepte:**
  - Klassifikation/Generalisierung
  - Assoziation
  - Aggregation

Matos, N.: An Approach to Knowledge Management, LNAI 513, Springer, 1991

---

---

---

---

---

---

---

---

---

---

## Abstraktionskonzepte (3)

- Klassifikation
  - entspricht der Bildung von Entity-Mengen
  - fasst Objekte (*Entities*) mit gemeinsamen Eigenschaften zu einem neuen zusammengesetzten Objekt (Entity-Typ, **Klasse**, Klassenobjekt) zusammen
  - eine Klasse ist definiert als Zusammenfassung von Objekten **gleichen Typs** (und gleicher Repräsentation) bzgl.
    - Attributnamen und -typen
    - Methoden
    - Integritätsbedingungen
  - es wird eine **'instance-of'**-Beziehung ('**io**') als 1-Ebenen-Beziehung zu den Objekten der Klasse aufgebaut

---

---

---

---

---

---

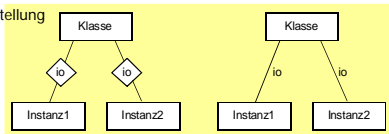
---

---

## Abstraktionskonzepte (4)

- Instantiation
  - inverses Konzept zur Klassifikation
  - wird benutzt, um zu Instanzen/Objekten zu gelangen, die den Eigenschaften der Klasse unterliegen
    - gleiche Struktur (Attribute)
    - gleiche Operationen
    - gleiche Integritätsbedingungen
- Klassifikation/Instantiation sind die primären Konzepte zur Objektbildung und -strukturierung

### ■ Grafische Darstellung (weitere Konzepte analog)



---

---

---

---

---

---

---

---

## Abstraktionskonzepte (5)

- Generalisierung
  - ergänzendes Konzept zur Klassifikation
  - allgemeinere Klasse (Superklassen) nehmen die Gemeinsamkeiten speziellerer Klassen (Subklassen) auf und unterdrücken deren Unterschiede
  - **'subclass-of'**-Beziehung ('**sc**'- oder '**is-a**'-Beziehung)
  - rekursiv anwendbar (n-Ebenen-Beziehung, Generalisierungshierarchie)
  - **Struktureigenschaften**
    - alle Instanzen einer Subklasse sind auch Instanzen der Superklasse
    - ein Objekt kann gleichzeitig Instanz verschiedener Klassen sein sowie auch Subklasse mehrerer Superklassen (→ Netzwerke, (n:m) !)
    - Zugehörigkeit eines Objektes zu einer Klasse/Superklasse wird im wesentlichen bestimmt durch **Struktur** (Attribute), **Verhalten** (Operationen) und **Integritätsbedingungen** der Klasse/Superklasse
    - Transitivität

---

---

---

---

---

---

---

---





## Abstraktionskonzepte (9)

- Spezialisierung
  - inverses Konzept zur Generalisierung
  - Entlang von Spezialisierungsbeziehungen **Vererbung von**
    - **Struktur**: Attribute, Konstante und Default-Werte
    - **Integritätsbedingungen**: Prädikate, Wertebereiche usw.
    - **Verhalten**: Operationen / Methoden
  - Integritätsbedingungen können **eingeschränkt**, Default-Werte können **überschrieben**, Methoden **überladen** werden
  - **Vorteile der Vererbung**
    - keine Wiederholung von Beschreibungsinformation
    - abgekürzte Beschreibung
    - Fehlervermeidung

---

---

---

---

---

---

---

---

## Abstraktionskonzepte (10)

- Spezialisierung (Forts.)
  - Subklasse: Klasse S, deren Entities eine Teilmenge einer Superklasse G sind:
    - $S \subseteq G$
    - d. h., jedes Element (Ausprägung) von S ist auch Element von G.
  - Spezialisierung:  $Z = \{S_1, S_2, \dots, S_n\}$   
Menge von Subklassen  $S_i$  mit derselben Superklasse G
    - Z heißt **vollständig (total)**, falls gilt  $G = \bigcup S_i$  ( $i = 1..n$ ), andernfalls **partiell**.
    - Z ist **disjunkt**, falls  $S_i \cap S_j = \{ \}$ , für  $i \neq j$  andernfalls **überlappend (nicht-disjunkt)**.

---

---

---

---

---

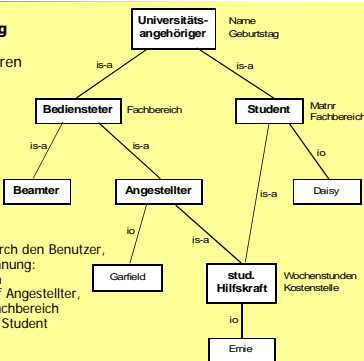
---

---

---

## Abstraktionskonzepte (11)

- **Mehrfach-Vererbung (multiple inheritance)** kann zu Konflikten führen



- Auflösung explizit durch den Benutzer, z. B. durch Umbenennung:  
Hiwi\_im\_Fachbereich  
→ Fachbereich of Angestellter,  
immatriculiert\_im\_Fachbereich  
→ Fachbereich of Student

---

---

---

---

---

---

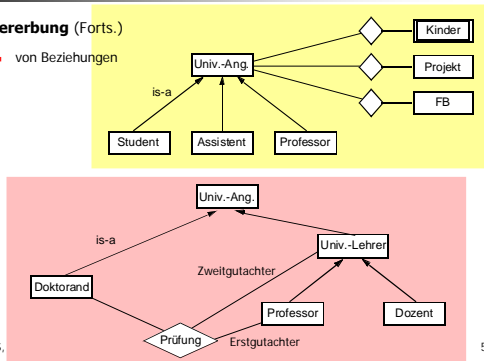
---

---

## Abstraktionskonzepte (12)

### Vererbung (Forts.)

- von Beziehungen



N. Ritter, DIS,

55

---

---

---

---

---

---

---

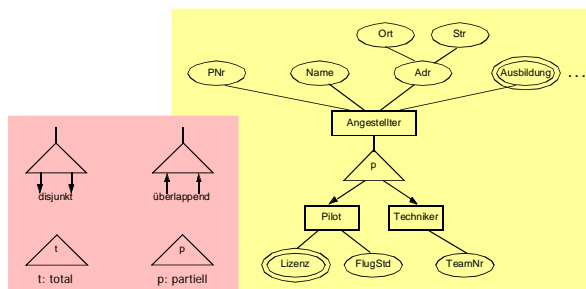
---

---

---

## Abstraktionskonzepte (13)

### Klassifikation und Generalisierung im ER-Diagramm



N. Ritter, DIS, SS 2005, Kapitel 3

56

---

---

---

---

---

---

---

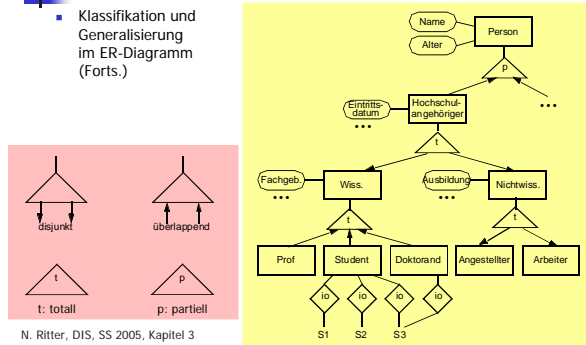
---

---

---

## Abstraktionskonzepte (14)

### Klassifikation und Generalisierung im ER-Diagramm (Forts.)



N. Ritter, DIS, SS 2005, Kapitel 3

---

---

---

---

---

---

---

---

---

---

## Abstraktionskonzepte (15)

### ■ Nutzung beim objektorientierten DB-Entwurf:

Vererbung von Typinformationen

- Strukturdefinitionen: Attribute, Defaultwerte, konstante Werte
- Integritätsbedingungen: Prädikate, Wertebereiche, Zusicherungen
- Verhalten: Operationen (Methoden) und ggf.
- Aspektdefinitionen: Kommentare, Einheiten u. a.

---

---

---

---

---

---

---

---

## Abstraktionskonzepte (16)

### ■ Element-Assoziation

- fasst Objekte (**Elemente**) zusammen, um sie im Rahmen einer Objektgruppe (**Mengenobjekt**) als Ganzes zu beschreiben
- dabei werden einerseits Details der einzelnen Elemente unterdrückt und andererseits bestimmte Eigenschaften, die die Objektgruppe charakterisieren, hervorgehoben.
- auch Gruppierung, Partitionierung, Überdeckungs-Aggregation genannt
- verkörpert eine **'element-of'**-Beziehung (**'eo'**) als 1-Ebenen-Beziehung
- es können auch heterogene Objekte zu einem Mengenobjekt zusammengefasst werden
- bei automatischer Ableitung müssen die Objekte das Mengenprädikat erfüllen; bei manuellem Aufbau wählt der Benutzer die Objekte aus und verknüpft sie mit dem Mengenobjekt (Connect)

---

---

---

---

---

---

---

---

## Abstraktionskonzepte (17)

### ■ Mengen-Assoziation

- ergänzendes Konzept zur Element-Assoziation
- drückt Beziehungen zwischen zusammengesetzten Mengenobjekten aus
- baut eine **'subset-of'**-Beziehung (**'ss'**) auf
- ist rekursiv anwendbar und organisiert die Mengenobjekte in einer Assoziations-Hierarchie (n-Ebenen-Beziehung)
- Ziel der Assoziation
  - **Zusammenfassung von Gruppen mit heterogenen Objekten** für einen bestimmten Kontext (Vergleiche Sichtkonzept)

---

---

---

---

---

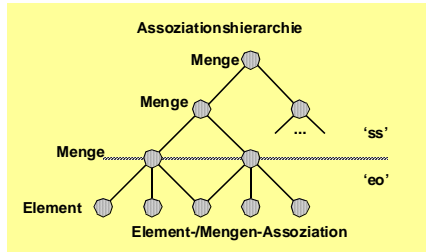
---

---

---

## Abstraktionskonzepte (18)

### Assoziation



---

---

---

---

---

---

---

---

## Abstraktionskonzepte (19)

### Struktureigenschaften der Assoziation

- alle Elemente eines Mengenobjekts sind auch Elemente der zugehörigen Supermenge
- Objekte können gleichzeitig Elemente verschiedener Mengenobjekte sein sowie auch Teilmenge von mehreren Supermengen
  - Netzwerke, (n:m) !

### Systemkontrollierte Ableitungen bei der Assoziation

- sie unterstützt keine Vererbung, da die Mengeneigenschaften keine Elementeigenschaften sind
- durch **Mitgliedschaftsimplication** lassen sich Eigenschaften bestimmen, die jedes gültige Element der Menge erfüllen muss
- Mengeneigenschaften** sind Eigenschaften der Menge, die über Elementeigenschaften abgeleitet sein können

---

---

---

---

---

---

---

---

## Abstraktionskonzepte (20)

### Assoziation

#### Operationen

- Erzeugen/Löschen
  - Create
  - Connect/Disconnect (manuell oder automatisch über Mengenprädikate)
  - Delete
- Suchen
- Schlussfolgerungen
  - Mitgliedschaftsimplication
  - Mengeneigenschaften

---

---

---

---

---

---

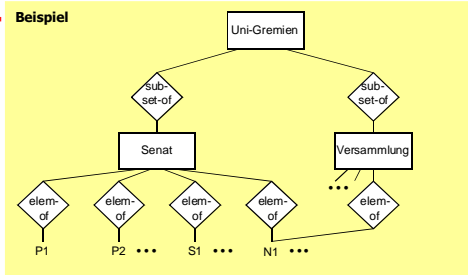
---

---

## Abstraktionskonzepte (21)

### Assoziation

#### Beispiel



---

---

---

---

---

---

---

---

## Abstraktionskonzepte (22)

### Aggregation

#### Teil-Ganze-Beziehung

- entweder **exklusiv** (kein anderes Objekt darf denselben Bestandteil haben) oder **gemeinsam** (derselbe Bestandteil wird in zwei oder mehr Objekten verwendet)
- entweder **abhängig** (Bestandteil kann nicht allein existieren und wird mit dem Objekt gelöscht oder **unabhängig** (Bestandteil kann auch für sich als Objekt existieren)
- Objekte mit exklusiven und/oder abhängigen Objekten heißen **zusammengesetzte Objekte** („composite objects“, „komplexe Objekte“) oder **Aggregate**

---

---

---

---

---

---

---

---

## Abstraktionskonzepte (23)

### Element-Aggregation

#### Aufgabe

- Element-Aggregation gestattet die Zusammensetzung von Objekten aus einfachen Objekten; sie stellt die 'Teil-Ganze'-Relation für solche nicht weiter zerlegbaren Objekte her

#### Anwendung

- Kollektion von einfachen Objekten (Element-Objekt, **Teil**) wird als zusammengesetztes Objekt (**Komponentenobjekt/Aggregatobjekt**) behandelt
- baut eine **'part-of'-Beziehung ('po')** auf (1-Ebenen-Abstraktion); typischerweise erzeugt der Benutzer ein Aggregat aus Teilen mit Hilfe von Connect-Anweisungen; dabei müssen Struktureigenschaften beachtet werden (z. B. Mannschaft besitzt 11 Spieler)
- Die Möglichkeit, heterogene Objekte zu aggregieren, erhöht die Anwendungsflexibilität

---

---

---

---

---

---

---

---

## Abstraktionskonzepte (24)

- Komponenten-Aggregation
  - **Aufgabe**
    - Komponenten-Aggregation dient als ergänzendes Konzept zur Element-Aggregation; durch sie wird die Teil-Ganze-Relation auf Komponenten angewendet
  - **Anwendung**
    - zwischen den Komponentenobjekten wird eine **'component-of'**-Beziehung (**'co'**) hergestellt (z. B. durch Connect-Anweisung)
    - rekursiv anwendbar und organisiert eine Aggregationshierarchie (n-Ebenen-Beziehung)

---

---

---

---

---

---

---

---

## Abstraktionskonzepte (25)

- Komponenten-Aggregation (Forts.)
  - **Struktureigenschaften bei der Aggregation**  
(Aggregation bedeutet auch 'besteht-aus'/'consists-of')
    - beschreibt *notwendige* Eigenschaften, die ein Objekt haben muss, um konsistent zu sein
    - Unterschied zu Klassen und Mengenobjekten, die ohne Instanzen existieren können, bzw. für die leere Mengen erlaubt sind
    - Elemente einer Subkomponente sind gleichzeitig auch Elemente aller Superkomponenten dieser Subkomponente
    - Objekte können gleichzeitig Elemente verschiedener Komponenten bzw. auch Subkomponente von mehreren Superkomponenten sein
    - Netzwerke, ((n:m) !)

---

---

---

---

---

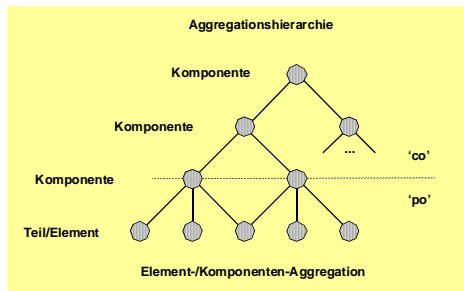
---

---

---

## Abstraktionskonzepte (26)

- Aggregation



---

---

---

---

---

---

---

---

## Abstraktionskonzepte (27)

### Aggregation (Forts.)

#### Operationen

- Erzeugen/Löschen
  - Create
  - Connect/Disconnect
  - Delete
- Integritätsbedingungen für Aggregatstrukturen
- Suchen (transitive Ableitung von komplexen Objekten)
- Schlussfolgerungen
  - implizierte Prädikate

---

---

---

---

---

---

---

---

## Abstraktionskonzepte (28)

### Aggregation (Forts.)

#### Systemkontrollierte Ableitungen: implizierte Prädikate

- Prädikate, die über der Aggregationshierarchie spezifiziert sind und gemeinsame Eigenschaften von Elementen/Aggregaten betreffen
- 'upward implied predicate'
  - Wenn  $P(x)$  wahr  $\Rightarrow$   $P(\text{Aggregatobjekte}(x))$  wahr
- 'downward implied predicate'
  - Wenn  $P(x)$  wahr  $\Rightarrow$   $P(\text{Komponentenobjekte}(x))$  wahr
- im Beispiel (nächste Folie):
  - 'upward implied predicate': Gewicht  $>$  x
  - 'downward implied predicate': Preis  $<$  y

---

---

---

---

---

---

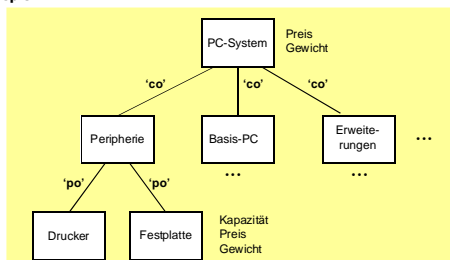
---

---

## Abstraktionskonzepte (29)

### Aggregation (Forts.)

#### Beispiel



---

---

---

---

---

---

---

---



## Abstraktionskonzepte (30)

- Objektorientierte Repräsentation
  - **Integration der Abstraktionskonzepte:**
    - ein Objekt kann mehrere Beziehungen verschiedener Typen aufbauen
    - entsprechend den verschiedenen Rollen, die in den Abstraktionen vorkommen
    - Objektsemantik wird bestimmt durch die Kontexte/Rollen eines Objektes
  - 3 Abstraktionskonzepte ermöglichen verschiedenartige Organisationsformen der modellierten Objekte und ihrer Beziehungen
  - können für Schlussfolgerungen benutzt werden:
    - um Aussagen über Objekte und ihre Eigenschaften abzuleiten
    - als Zusatz bei Manipulations- und Retrievaloperationen

---

---

---

---

---

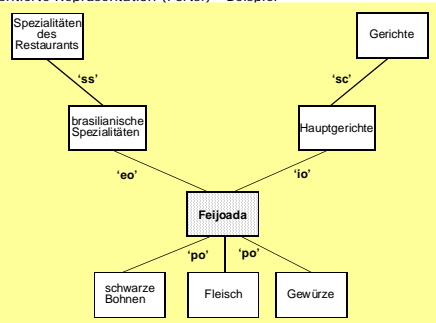
---

---

---

## Abstraktionskonzepte (31)

- Objektorientierte Repräsentation (Forts.) - Beispiel




---

---

---

---

---

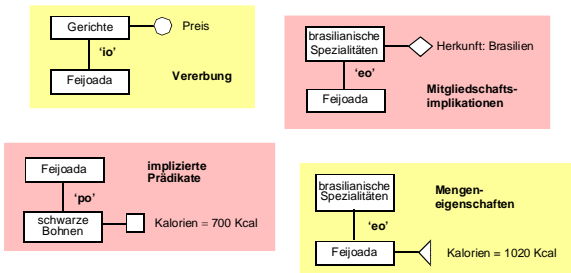
---

---

---

## Abstraktionskonzepte (32)

- Objektorientierte Repräsentation (Forts.) – Beispiel - Schlussfolgerungen




---

---

---


---

---

---

---

---



## Zusammenfassung (1)

- **DB-Entwurf umfasst**
  - Informationsbedarfsanalyse
  - konzeptionelles DB-Schema (-> Informationsmodell)
  - logisches DB-Schema
  - 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

N. Ritter, DIS, SS 2005, Kapitel 3 76

---

---

---


---

---

---

---

---



## Zusammenfassung (2)

- **Einführung weiterer Modellierungskonzepte**
  - Verfeinerung von Beziehungen durch Kardinalitätsrestriktionen und vor allem Abstraktionskonzepte
  - Das erweiterte ERM ist sehr mächtig und umfasst viele bekannte Modellierungskonzepte
  - Integritätsbedingungen wurden hier nicht behandelt (-> Relationenmodell)
  
- **Abstraktionskonzepte und deren Implikationen**
  - Generalisierung und Vererbung
  - Assoziation mit Mengeneigenschaften und Mitgliedschaftsimplikationen
  - Aggregation und implizierte Prädikate
  - Integration der Abstraktionskonzepte mittels objektzentrierter Darstellung

N. Ritter, DIS, SS 2005, Kapitel 3 77

---

---

---

---

---

---

---

---