

# 3. Grundlagen des Relationenmodells

- **Übersicht**

- Grundkonzepte
- Normalisierte Relationen
- Schlüssel
- Sprachen für das Relationenmodell

- **Relationenalgebra**

- Klassische Mengenoperationen
- Relationenoperationen
- Anfragen

- **Abbildung ERM → RM**

- Abbildung von Entity- und Relationship-Mengen
- Abbildung der Generalisierung
- Abbildung der Aggregation

# Relationenmodell – Übersicht

- **Datenstruktur**

Relation (Tabelle)


- ➔ einzige Datenstruktur (neben atomaren Werten)
- ➔ alle Informationen ausschließlich durch Werte dargestellt
- ➔ zeitinvariante Typinformation: Relationenschema
- ➔ Integritätsbedingungen auf/zwischen Relationen: relationale Invarianten

- **Operatoren auf (mehreren) Relationen**

- Vereinigung, Differenz
- Kartesisches Produkt
- Projektion
- Selektion
- zusätzlich: Grundoperationen (Einfügen, Löschen, Ändern)
- ➔ Tabellenverknüpfung und -manipulation

- **Beziehungen**

- sind stets **explizit, binär und symmetrisch**
- werden durch Werte dargestellt: Rolle von Primär-/Fremdschlüssel (Gewährleistung von referentieller Integrität)
- können in SQL automatisch gewartet werden (referentielle Aktionen)

# Relationenmodell – Grundkonzepte<sup>1</sup>

## ERM

Einwertiges Attribut  
Definitionsbereich  
Primärschlüssel

## RM

} wie im ERM

Zusammengesetztes Attribut  
Mehrwertiges Attribut

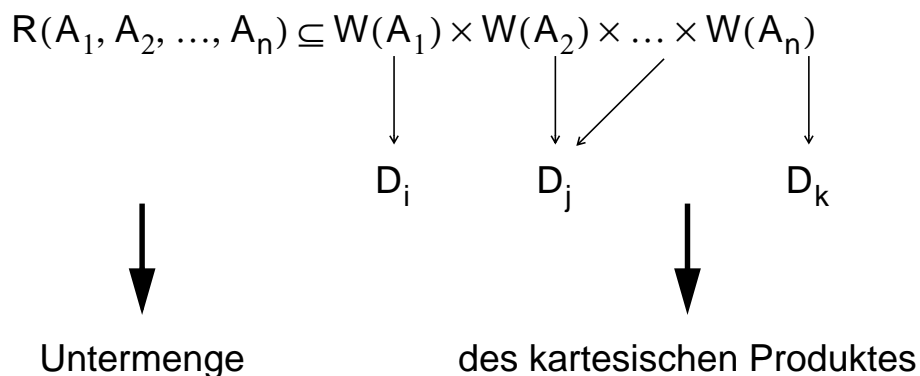
nur als unabhängige  
einwertige Attribute

Entity-Menge

Relation

Relationship-Menge

### • Definition: Normalisierte Relation



### • Darstellungsmöglichkeit für R: n-spaltige Tabelle

➔ Jede **Relation** kann als Tabelle dargestellt werden

### • Relation ist eine Menge: Garantie der Eindeutigkeit der Zeilen/Tupel

➔ **Primärschlüssel** (und ggf. mehrere Schlüsselkandidaten)

1. Codd, E.F.: A Relational Model of Data for Large Shared Data Banks, in: Comm. ACM 13:6, June 1970, pp. 377-387.

# Normalisierte Relationen in Tabellendarstellung

FB	<u>FBNR</u>	FBNAME	DEKAN
	FB 9	Wirtschaftswiss.	4711
	FB 5	Informatik	2223

PROF	<u>PNR</u>	PNAME	FBNR	FACHGEBIET
	1234	Härder	FB 5	Datenbanksysteme
	5678	Wedekind	FB 9	Informationssysteme
	4711	Müller	FB 9	Operations Research
	6780	Nehmer	FB 5	Betriebssysteme
	2223	Richter	FB 5	Expertensysteme

- **Grundregeln:**

1. Jede Zeile (Tupel) ist eindeutig und beschreibt ein Objekt der Miniwelt
2. Die **Ordnung der Zeilen** ist **ohne Bedeutung**; durch ihre Reihenfolge wird keine für den Benutzer relevante Information ausgedrückt
3. Die **Ordnung der Spalten** ist **ohne Bedeutung**, da sie einen eindeutigen Namen (Attributnamen) tragen
4. Jeder Datenwert innerhalb einer Relation ist ein **atomares** Datenelement
5. Alle für den Benutzer **bedeutungsvollen Informationen** sind **ausschließlich durch Datenwerte** ausgedrückt
6. Es existieren ein Primärschlüssel und ggf. weitere Schlüsselkandidaten

## Relationenmodell – Grundkonzepte (2)

- **Informationsdarstellung im RM**

- ausschließlich durch Werte  $W(A_i)$
- Reihenfolge von Zeilen und Spalten enthält keine Information

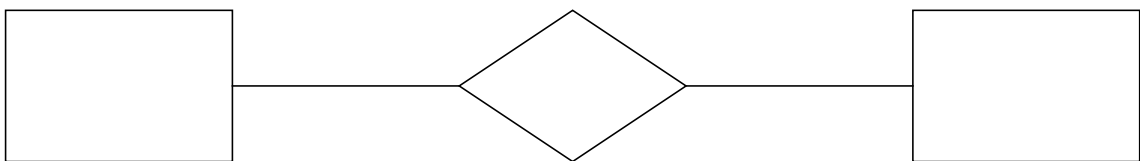
- **Wie wird „relationenübergreifende“ Information dargestellt?**

- Fremdschlüssel
  - Bezug auf den Primärschlüssel einer anderen (oder derselben) Relation definiert (gleicher Definitionsbereich)
  - trägt inter- oder intrarelationale Informationen
- Beziehungen werden durch Fremdschlüssel und zugehörigen Primärschlüssel oder Schlüsselkandidaten dargestellt!

- **Modellinhärente Integritätsbedingungen:**

- Welche Zusicherungen werden vom Datenmodell garantiert?**

- Mengeneigenschaft von Relationen
  - ↳ zur Abbildung von Entities/Relationships
- Beziehungstypen (1:1, ..., n:m)      ↳ mit Einschränkungen als (1:n)
- Referentielle Integrität              ↳ wertbasierte Beziehungen
- Kardinalitätsrestriktionen?          ↳ wünschenswert
- Semantik der benutzerdefinierten Beziehung?
  - ↳ Es ist keine Systemunterstützung vorgesehen



# Fremdschlüssel

- **Definition:**

Ein Fremdschlüssel bzgl. einer Relation R1 ist ein (ggf. zusammengesetztes) Attribut FS einer Relation R2, für das zu jedem Zeitpunkt gilt: **zu jedem Wert** (ungleich Null) **von FS** muss **ein gleicher Wert des Primärschlüssels PS** oder eines Schlüsselkandidaten SK in irgendeinem Tupel von Relation R1 vorhanden sein.

- **Bemerkungen:**

1. Fremdschlüssel und zugehöriger Primärschlüssel (Schlüsselkandidat) tragen wichtige interrelationale (manchmal auch intrarelationale) Informationen. Sie sind auf dem gleichen Wertebereich definiert (vergleichbar und vereinigungsverträglich). Sie gestatten die Verknüpfung von Relationen mit Hilfe von Relationenoperationen.
2. Fremdschlüssel können Nullwerte aufweisen, wenn sie nicht Teil eines Primärschlüssels sind oder wenn nicht explizit NOT NULL spezifiziert ist.
3. Schlüsselkandidaten können Nullwerte aufweisen, wenn nicht explizit NOT NULL spezifiziert ist.
4. Ein Fremdschlüssel ist zusammengesetzt, wenn der zugehörige Primärschlüssel (Schlüsselkandidat) zusammengesetzt ist.  
Eine Relation kann mehrere Fremdschlüssel besitzen, welche die gleiche oder verschiedene Relationen referenzieren.
5. Referenzierte und referenzierende Relation sind nicht notwendig verschieden („**self-referencing table**“).
6. Zyklen sind möglich (**geschlossener referentieller Pfad**).

# Sprachen für das Relationenmodell

- **Datenmodell = Datenobjekte + Operatoren**
- **Unterstützung verschiedener Benutzerklassen:**
  - Anwendungsprogrammierer
  - DBA
  - anspruchsvolle Laien
  - parametrische Benutzer
  - gelegentliche Benutzer
- **Im RM wird vereinheitlichte Sprache angestrebt für**
  - **alle Aufgaben der Datenverwaltung**
    - Datendefinition
    - Anfragen (*Queries*)
    - Datenmanipulation
    - Zugriffs-, Integritäts- und Transaktionskontrolle
  - **zur Nutzung**
    - im *Stand-Alone*-Modus (Ad-hoc-Anweisungen) und
    - in einer Wirtssprache (eingebettete DB-Anweisungen)
- ➔ **Die wichtigsten Eigenschaften von Anfragesprachen werden am Beispiel der Relationenalgebra diskutiert und anschließend zusammengefasst**
- **Vier verschiedene Grundtypen:**
  - Relationenalgebra (z. B. ISBL)
  - Relationenkalkül (z. B. Alpha)
  - Abbildungsorientierte Sprachen (z. B. SQL)
  - Graphikorientierte Sprachen (z. B. Query-by-Example)

# Relationenalgebra – Überblick

- **Objekte:**

Ein System, das aus einer nichtleeren Menge und einer Familie von Operationen besteht, heißt **Algebra**.

➔ Relationen sind Mengen.

- **Operationen:**

Operationen auf Relationen arbeiten auf einer oder mehreren Relationen als Eingabe und erzeugen eine Relation als Ausgabe.

Wichtige Eigenschaften: **Ad-hoc-Formulierung, Deskriptivität, Mengenorientiertheit, Abgeschlossenheit, . . .**

- **Klassische Mengenoperationen:**

- Vereinigung, Differenz, kartesisches Produkt
- ableitbar: Durchschnitt

- **Relationenoperationen:**

- Projektion, Restriktion (Selektion)
- ableitbar: Verbund (Join), Division

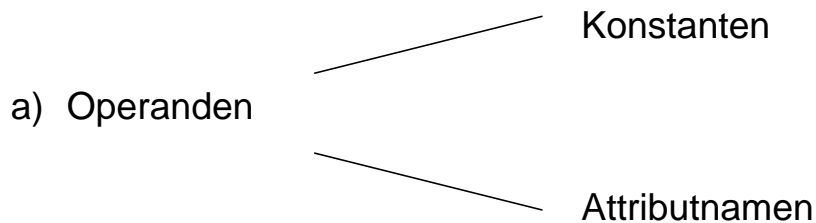
➔ Auswahlvermögen entspricht Prädikatenkalkül erster Ordnung („relational vollständig“)



## Selektion (Restriktion)

- Auswahl von Zeilen einer Relation über Prädikate, abgekürzt  $\sigma_P$

P = log. Formel (ohne Quantoren!) zusammengestellt aus:



b)  $\Theta \in \{<, =, >, \leq, \neq, \geq\}$

c)  $\vee, \wedge, \neg$

- Definition:

$$\sigma_P(R) = \{t \mid t \in R \wedge P(t)\}$$

### Beispiele:

$\sigma_{\text{NAME} = \text{'Schmid'} \wedge \text{ALTER} > 30}$  (PERS)

$\sigma_{\text{GEHALT} < \text{PROVISION}}$  (PERS)

- Anwendung von:  $\sigma_{\text{ANR} = \text{'K55'} \wedge \text{GEHALT} > 50\,000}$  (PERS)

PERS	PNR	NAME	ALTER	GEHALT	ANR	MNR
	406	Coy	47	50 700	K55	123
	123	Müller	32	43 500	K51	-
	829	Schmid	36	45 200	K53	777
	574	Abel	28	36 000	K55	123

### Ergebnis:

ERG	PNR	NAME	ALTER	GEHALT	ANR	MNR

# Projektion

- **Auswahl der Spalten (Attribute)**  $A_1, A_2, \dots, A_k$   
aus einer Relation R (Grad  $n \geq k$ )

$$\pi(R) = \{p \mid \exists t \in R : p = \langle t[A_1], \dots, t[A_k] \rangle\}$$

*! Duplikate entfernt !*

- **Alternativ: Benutzung der Spaltennummern  $j_i$**

$$\pi_{j_1, j_2, \dots, j_k}(R)$$

- **Beispiel:**

$$\pi_{\text{NAME, GEHALT, ALTER}}(\text{PERS})$$

- **Anwendung von:**  $\pi_{\text{ANR, MNR}}(\text{PERS})$

PERS	PNR	NAME	ALTER	GEHALT	ANR	MNR
	406	Coy	47	50 700	K55	123
	123	Müller	32	43 500	K51	-
	829	Schmid	36	45 200	K53	777
	574	Abel	28	36 000	K55	123

**Ergebnis:**

ERG	ANR	MNR

## Relationenalgebra – Beispiel-DB

ABT	<u>ANR</u>	ANAME	AORT
	K51	Planung	Kaiserslautern
	K53	Einkauf	Frankfurt
	K55	Vertrieb	Frankfurt

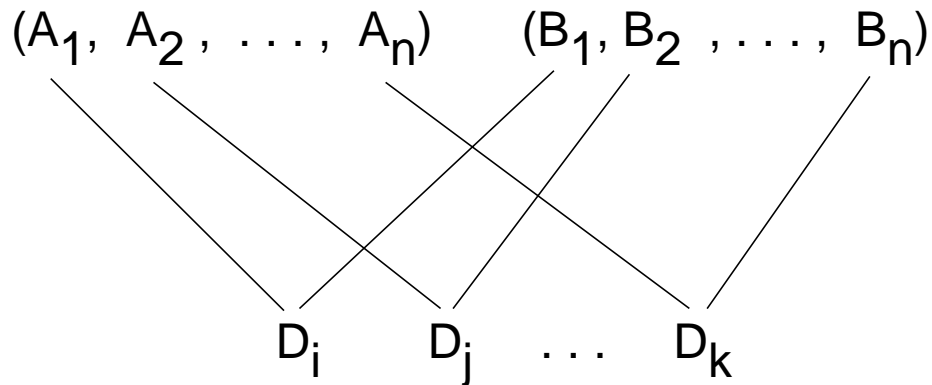
PERS	<u>PNR</u>	NAME	ALTER	GEHALT	ANR	MNR
	406	Coy	47	50 700	K55	123
	123	Müller	32	43 500	K51	-
	829	Schmid	36	45 200	K53	777
	574	Abel	28	36 000	K55	123

1. Finde alle Abteilungsorte
2. Find alle Angestellten (PNR, NAME) aus Abteilung K55, die mehr als 40.000 DM verdienen
3. Finde alle Angestellten (PNR, ALTER, ANAME), die in einer Abteilung in Frankfurt arbeiten und zwischen 30 und 34 Jahre alt sind.

# Klassische Mengenoperationen

- **Voraussetzung: Vereinigungsverträglichkeit der beteiligten Relationen**

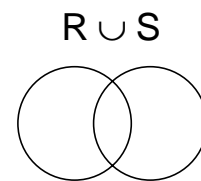
Gleicher Grad – Gleiche Bereiche



→  $W(A_i) = W(B_i) \quad : \quad i = 1, n$

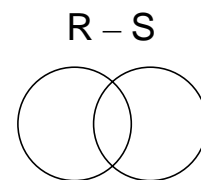
## 1. Vereinigung (UNION) von R und S

$$R \cup S = \{t | t \in R \vee t \in S\}$$



## 2. Differenz

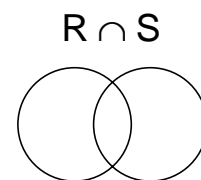
$$R - S = \{t | t \in R \wedge t \notin S\}$$



- **zusätzlich (redundante Mengenoperationen):**

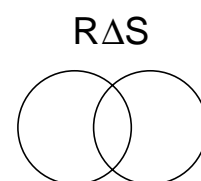
## 3. Durchschnitt (INTERSECTION)

$$\begin{aligned} R \cap S &= R - (R - S) \\ &= \{t | t \in R \wedge t \in S\} \end{aligned}$$



## 4. Symmetrische Differenz

$$\begin{aligned} R \Delta S &= (R \cup S) - (R \cap S) \\ &= ((R \cup S) - (R - (R - S))) \\ &= \{t | t \in R \oplus t \in S\} \end{aligned}$$



## (Erweitertes) Kartesisches Produkt

- R (Grad r) und S (Grad s) beliebig

$$\mathbf{K} = \mathbf{R} \times \mathbf{S}$$

$$= \{ \mathbf{k} \mid \exists \mathbf{x} \in \mathbf{R}, \mathbf{y} \in \mathbf{S} : (\mathbf{k} = \mathbf{x} \mid \mathbf{y}) \}$$

$$k = x \mid y = \langle x_1, \dots, x_r, y_1, \dots, y_s \rangle$$

nicht  $\langle \langle x_1, \dots, x_r \rangle, \langle y_1, \dots, y_s \rangle \rangle$  wie übliches kartesisches Produkt

- Anwendung von: **ABT x PERS**

ABT	<u>ANR</u>	ANAME	AORT	PERS	<u>PNR</u>	ALTER	ANR
	K51	Planung	KL		406	47	K55
	K53	Einkauf	F		123	32	K51
	K55	Vertrieb	F		829	36	K53
					574	28	K55

ABT x PERS	ANR	ANAME	AORT	PNR	ALTER	ANR'
	K51	Planung	KL	406	47	K55
	K51	Planung	KL	123	32	K51
	K51	Planung	KL	829	36	K53
	K51	Planung	KL	574	28	K55
	K53	Einkauf	F	406	47	K55
			• • •			

➔ Iterative Schleifenmethode (Nested-Loop-Algorithmus)

## Verbund (Join, $\Theta$ -Join)

- **Grob:**

Kartesisches Produkt zwischen zwei Relationen R (Grad r) und S (Grad s).  
eingeschränkt durch  $\Theta$ -Bedingungen zwischen i-Spalte von R und j-Spalte von S.

- Sei  $\Theta \in \{<, =, >, \leq, \neq, \geq\}$  (arithmetischer Vergleichsoperator)

$\Theta$ -Verbund zwischen R und S:

$$V = R \bowtie_{i\Theta j} S$$
$$= \sigma_{i\Theta r + j}(R \times S)$$

- **Bemerkungen:**

(1) Speziell  $\Theta = '='$  : Gleichverbund (Equi-Join)

(2) statt i und j: Attributnamen A und B

z. B.:  $R \bowtie_{i\Theta j} S \equiv R \bowtie_{A\Theta B} S$

(3) Ein Gleichverbund zwischen R und S heißt **verlustfrei**, wenn alle Tupel von R und S am Verbund teilnehmen. Die inverse Operation Projektion erzeugt dann wieder R und S (**lossless join**)

## Gleichverbund – Beispiel

- Anwendung von:

**ABT** ⋈ **PERS**  
 ANR = ANR

ABT	<u>ANR</u>	ANAME	AORT	PERS	<u>PNR</u>	ALTER	ANR
	K51	Planung	KL		406	47	K55
	K53	Einkauf	F		123	32	K51
	K55	Vertrieb	F		829	36	K53
					574	28	K55

R = ABT ⋈ PERS	ANR	ANAME	AORT	PNR	ALTER	ANR'
	K51	Planung	KL	123	32	K51
	K53	Einkauf	F	829	36	K53
	K55	Vertrieb	F	406	47	K55
	K55	Vertrieb	F	574	28	K55

→ **verlustfreier Gleichverbund:**  $\pi_{\text{ANR, ANAME, AORT}}(R) = \text{ABT}$

$\pi_{\text{PNR, ALTER, ANR'}}(R) = \text{PERS}$

- Verlustbehafteter Gleichverbund,**

wenn Tupeln in ABT oder PERS keine Verbundpartner finden,  
 z. B. (K56, Finanzen, M) in ABT oder (471, 63, -) in PERS

→  $\pi$  als Umkehroperation führt nicht auf ABT oder PERS

## Natürlicher Verbund (*Natural Join*)

- **grob:**

**Gleichverbund über alle gleichen Attribute** und Projektion über die verschiedenen Attribute

- **gegeben:**  $R(A_1, A_2, \dots, A_{r-j+1}, \dots, A_r)$

$S(B_1, B_2, \dots, B_j, \dots, B_s)$

o.B.d.A.: (sonst. Umsortierung)

$$B_1 = A_{r-j+1}$$

$$B_2 = A_{r-j+2}$$

$\vdots$

$$B_j = A_r$$

### Natürlicher Verbund zwischen R und S:

$$N = R \bowtie S$$

$$= \pi_{A_1, \dots, A_r, B_{j+1}, \dots, B_s} \sigma_{(R.A_{r-j+1} = S.B_1) \wedge \dots \wedge (R.A_r = S.B_j)} (R \times S)$$

$\bowtie$  = Zeichen für Natural Join  $\Rightarrow \Theta = '='$

- **Bemerkung:**

Attribute sind durch Übereinstimmungsbedingung gegeben



## Natürlicher Verbund – Beispiel

- Anwendung von:

ABT  $\bowtie$  PERS

ABT	<u>ANR</u>	ANAME	AORT	PERS	<u>PNR</u>	ALTER	ANR
	K51	Planung	KL		406	47	K55
	K53	Einkauf	F		123	32	K51
	K55	Vertrieb	F		829	36	K53
					574	28	K55

AP = ABT $\bowtie$ PERS	ANR	ANAME	AORT	PNR	ALTER
	K51	Planung	KL	123	32
	K53	Einkauf	F	829	36
	K55	Vertrieb	F	406	47
	K55	Vertrieb	F	574	28

→ **verlustfreier natürlicher Verbund:**  $\pi_{\text{ANR, ANAME, AORT}}(\text{AP}) = \text{ABT}$

$\pi_{\text{PNR, ALTER, ANR}}(\text{AP}) = \text{PERS}$

- Verlustbehafteter natürlicher Verbund analog zu Gleichverbund

## Natürlicher Verbund - Beispiel (2)

ABT	ANR	ANAME	AORT
	K51	Planung	Kaiserslautern
	K53	Einkauf	Frankfurt
	K55	Vertrieb	Frankfurt

PERS	PNR	NAME	ALTER	GEHALT	ANR	MNR
	406	Coy	47	50 700	K55	123
	123	Müller	32	43 500	K51	-
	829	Schmid	36	45 200	K53	777
	574	Abel	28	36 000	K55	123

- **Annahmen:**

- ABT: N/10 Tupel
- PERS: N Tupel
- **Gleichverteilung der Attributwerte**
  - AORT: 20 Werte
  - ALTER: 50 Werte (16-65)
- **Stochastische Unabhängigkeit** der Werte verschiedener Attribute
- Verlustfreie Verbunde von R1 und R2 über Primär-/Fremdschlüssel, mit  $\text{Card}(R1) < \text{Card}(R2)$ :  $\text{Card}(R1 \bowtie R2) = \text{Card}(R2)$

- **Anfrage:**

Finde alle Angestellten (PNR, ALTER, ANAME), die in einer Abteilung in Frankfurt arbeiten und 25 Jahre und jünger sind.

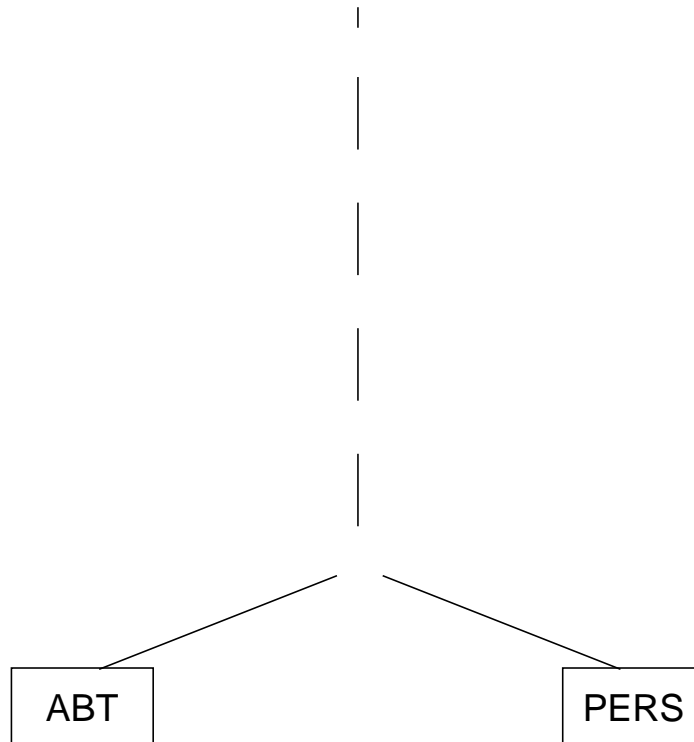
# Natürlicher Verbund – Lösungen

- Lösung 1:

$\pi_{\text{PNR,ALTER,ANAME}}$

$(\sigma_{\text{AORT}='F'} (\sigma_{\text{ALTER} \leq 25} (\sigma_{\text{ABT.ANR}=\text{PERS.ANR}} (\text{ABT} \times \text{PERS}))))$

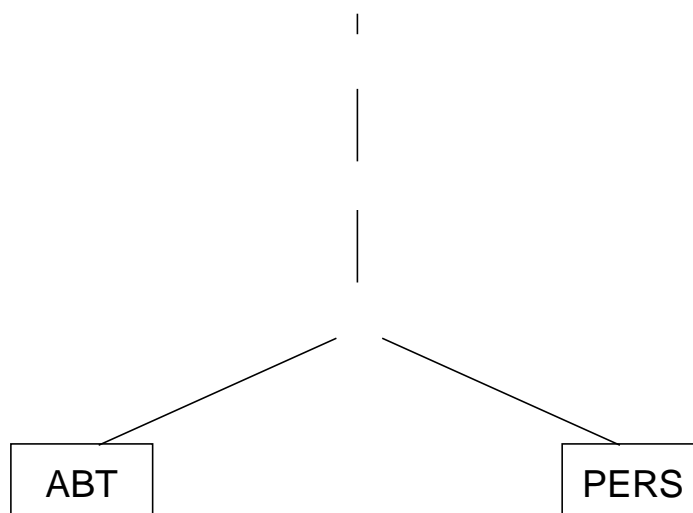
- Zugehöriger Operatorbaum



- Lösung 2:

$\pi_{\text{PNR,ALTER,ANAME}}$

$(\sigma_{\text{ALTER} \leq 25 \wedge \text{AORT}='F'} (\text{ABT} \bowtie \text{PERS}))$

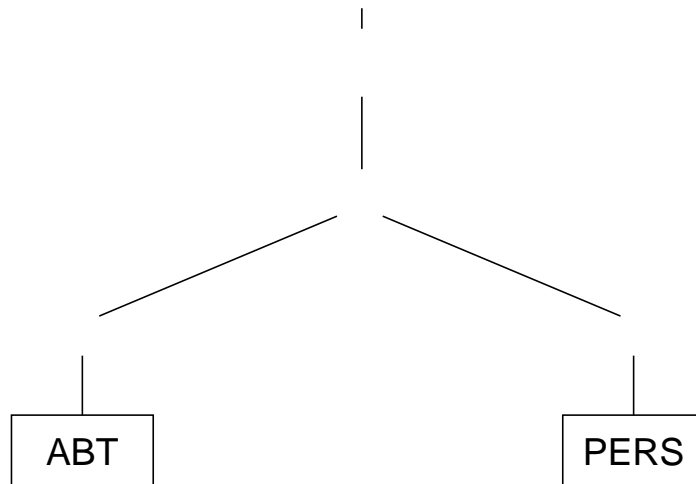


## Natürlicher Verbund – Lösungen (2)

- Lösung 3:

$\pi_{\text{PNR,ALTER,ANAME}}$

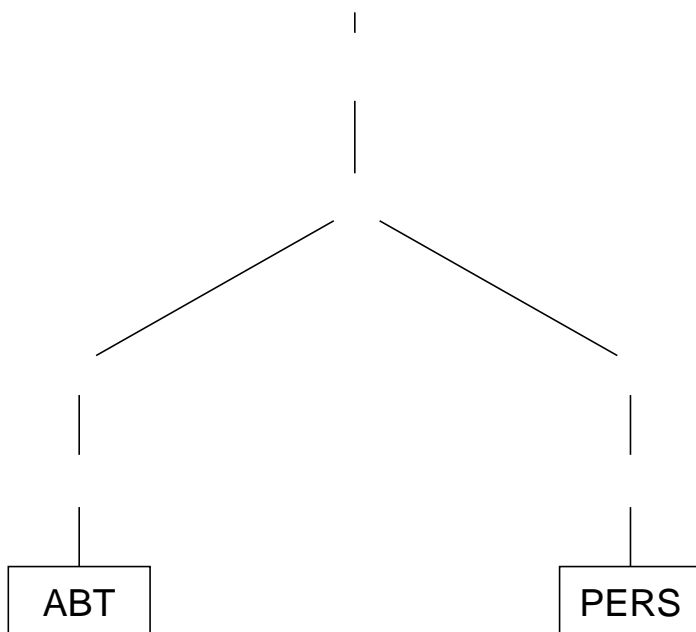
$((\sigma_{\text{AORT}='F'} \text{ ABT}) \bowtie (\sigma_{\text{ALTER} \leq 25} \text{ PERS}))$



- Lösung 4:

$\pi_{\text{PNR,ALTER,ANAME}} ((\pi_{\text{ANR,ANAME}}$

$(\sigma_{\text{AORT}='F'} \text{ ABT})) \bowtie (\pi_{\text{PNR,ALTER,ANR}} (\sigma_{\text{ALTER} \leq 25} \text{ PERS})))$



## Natürlicher Verbund - Beispiel (3)

- Ist der Verbund  $\bowtie$  immer Umkehroperation zur Projektion ( $\pi$ )?

- Beispiel 1 (1:n):

$$AP1 = \pi_{ANR, ANAME, AORT} (AP) \quad AP3 = AP1 \bowtie AP2 \stackrel{!}{=} AP$$

$$AP2 = \pi_{PNR, ALTER, ANR} (AP)$$

- Beispiel 2 (n:m):

DA	( PNR,	FIGUR,	A-ORT)
	P1	Faust	MA
	P1	Mephisto	KL
	P2	Wallenstein	MA

$$DA1 = \pi_{PNR, A-ORT} (DA)$$

$$DA2 = \pi_{FIGUR, A-ORT} (DA)$$

= DA1	PNR	A-ORT

= DA2	FIGUR	A-ORT

$$DA3 = DA1 \bowtie DA2$$

= DA3	PNR	FIGUR	A-ORT

- ➔ „Connection Trap“ bei Projektion von Schlüsselteilen und nachfolgendem Verbund

# Beispiel-DB: BÜHNE

## *DICHTER (DI)*

<u>AUTOR</u> G-ORT G-JAHR
---------------------------

## *DRAMA (DR)*

<u>TITEL</u> U-ORT U-JAHR AUTOR
---------------------------------

## *SCHAUSPIELER (SP)*

<u>PNR</u> NAME W-ORT
-----------------------

## *ROLLE (RO)*

<u>FIGUR</u> TITEL R-Typ
--------------------------

## *DARSTELLER (DA)*

<u>PNR</u> <u>FIGUR</u> A-JAHR A-ORT THEATER
--

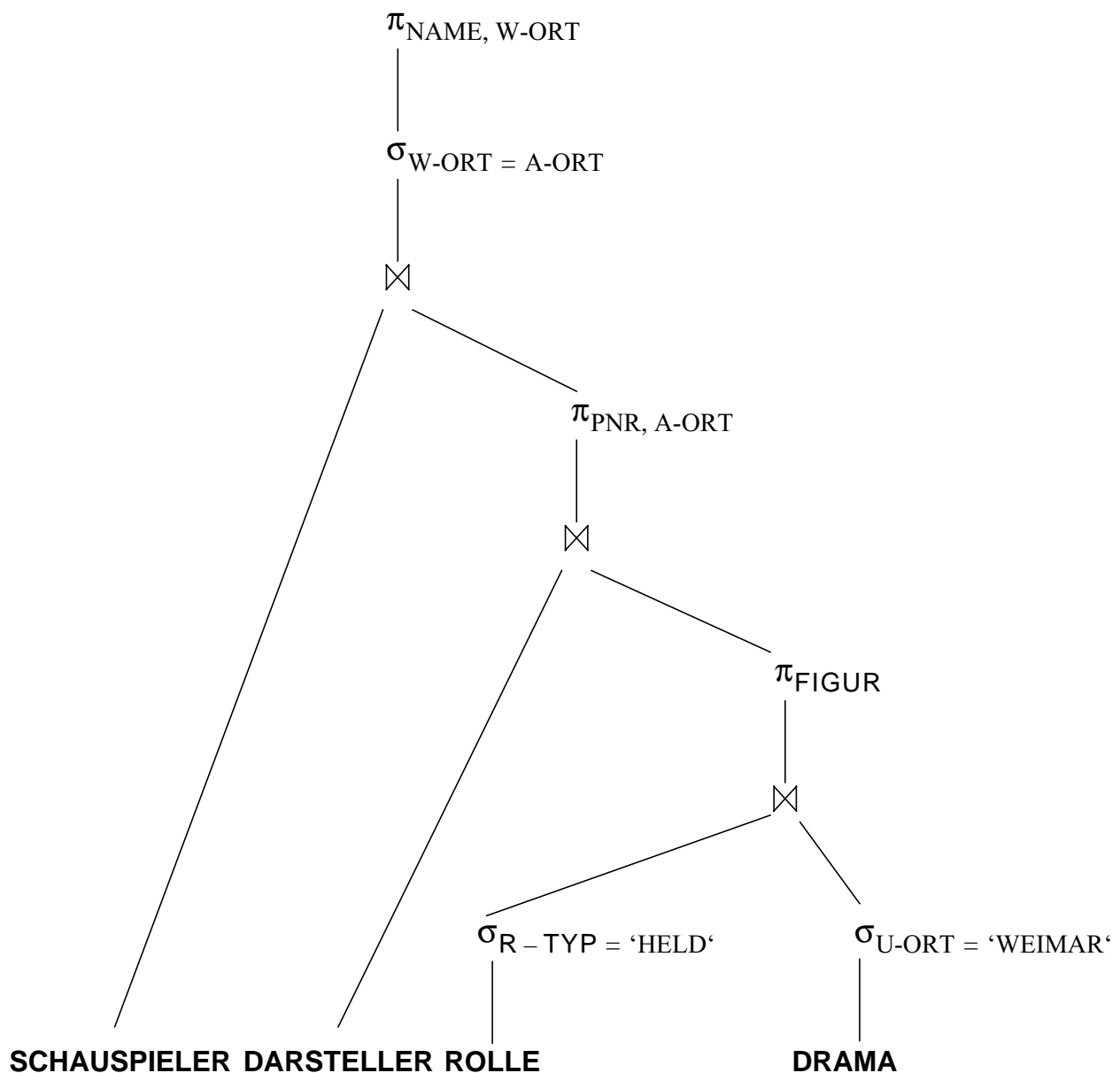
## Beispiel-DB: Anfragen

- Q1: Liste alle Dramen mit ihren Autoren (mit TITEL, AUTOR) auf, die nach 1800 uraufgeführt wurden.
- Q2a: Finde alle Schauspieler (NAME, W-ORT), die einmal den 'Faust' gespielt haben.
- Q2b: Finde alle Schauspieler (NAME, W-ORT), die einmal im 'Faust' mitgespielt haben.
- Q2c: Finde alle Schauspieler (NAME, W-ORT), die in Dramen von Schiller mitgespielt haben.

## Anfragedarstellung als Operatorbaum

Q2d: Finde alle Schauspieler (NAME, W-ORT), die bei in Weimar uraufgeführten Dramen an ihrem Wohnort als 'Held' mitgespielt haben.

Q2d =  $\pi_{\text{NAME, W-ORT}} (\sigma_{\text{W-ORT} = \text{A-ORT}} (\text{SCHAUSPIELER} \bowtie (\pi_{\text{PNR, A-ORT}} (\text{DARSTELLER} \bowtie (\pi_{\text{FIGUR}} (\sigma_{\text{R-TYP} = \text{'HELD'}} \text{ROLLE}) \bowtie (\sigma_{\text{U-ORT} = \text{'WEIMAR'}} \text{DRAMA}))))))$





## Anfragen (2)

Q3: Kann die Frage beantwortet werden: Welcher Dichter ist Schauspieler?  
oder: Welcher Dichter hat in einem seiner eigenen Stücke gespielt?

Q4: Finde die Schauspieler (PNR), die **nie** gespielt haben.

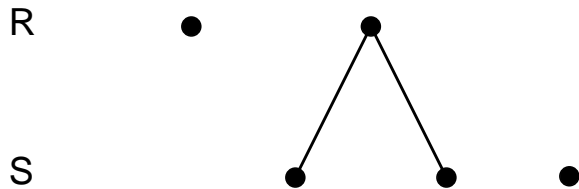
Q5a: Finde die Schauspieler (PNR), die Faust oder Wallenstein gespielt haben.

Q5b: Finde die Schauspieler (PNR), die **nur** Faust oder Wallenstein gespielt haben.

## Äußerer Verbund (Outer Join)

- **Ziel:** Verlustfreier Verbund soll erzwungen werden

Beispiel.: Bei  $R \bowtie S$  sollen auch Teilobjekte als Ergebnis geliefert werden



- **bisher:**  $R \bowtie S$  liefert nur „vollständige Objekte“
- **Trick:** Einfügen einer speziellen Leerzeile zur künstlichen Erzeugung von Verbundpartnern

- **Beispiel**

SP (PNR, NAME, ...)

P1 x

P2 y

DA (PNR, FIGUR, ...)

P1 F

P1 W

P3 M

$$SP' = SP \cup ((\Pi_{\text{PNR}} DA - \Pi_{\text{PNR}} SP) \times x \equiv x \equiv \dots)$$

$$= SP \cup \left( \left( \begin{pmatrix} P1 \\ P3 \end{pmatrix} - \begin{pmatrix} P1 \\ P2 \end{pmatrix} \right) \times x \equiv x \equiv \dots \right) = SP \cup (P3 \equiv)$$

$$DA' = DA \cup ((\Pi_{\text{PNR}} SP - \Pi_{\text{PNR}} DA) \times x \equiv x \equiv \dots)$$

$$= DA \cup \left( \left( \begin{pmatrix} P1 \\ P2 \end{pmatrix} - \begin{pmatrix} P1 \\ P3 \end{pmatrix} \right) \times x \equiv x \equiv \dots \right) = DA \cup (P2 \equiv)$$

## Äußerer Verbund (2)

- **Definition:** Seien A die Verbundattribute,  $\{\equiv\}$  der undefinierte Wert und

$$R' := R \cup ((\pi_A(S) - \pi_A(R)) \times \{\equiv\} \times \dots \times \{\equiv\})$$

$$S' := S \cup ((\pi_A(R) - \pi_A(S)) \times \{\equiv\} \times \dots \times \{\equiv\})$$

### Äußerer Gleichverbund

$$R \bowtie \sqsupset S := R' \bowtie S'$$

$$R.A = S.A \quad R'.A = S'.A$$

### Äußerer natürlicher Verbund

$$R \bowtie \sqsupset S := R' \bowtie S'$$

- **Linker äußerer Gleichverbund**

Bei bei dieser Operation bleibt die linke Argumentrelation verlustfrei, d. h., bei Bedarf wird ein Tupel durch „NULL“-Werte „nach rechts“ aufgefüllt.

### Linker äußerer Gleichverbund

$$R \bowtie \sqsupset S := R \bowtie S'$$

$$R.A = S.A \quad R.A = S'.A$$

- **Rechter äußerer Gleichverbund**

Dabei bleibt analog die rechte Argumentrelation verlustfrei; fehlende Partnertupel werden durch Auffüllen mit „NULL“-Werten „nach links“ ergänzt

### Rechter äußerer Gleichverbund

$$R \bowtie \sqsupset S := R' \bowtie S$$

$$R.A = S.A \quad R'.A = S.A$$

## Beispiele zum äußeren Gleichverbund

- Gleichverbund**

R	A	B	C	⊗	S	C	D	E	=	ERG	A	B	C	D	E
	a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>			c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>			a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
	a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>			c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>							

- Linker äußerer Gleichverbund**

R	A	B	C	⊗	S	C	D	E	=	ERG	A	B	C	D	E
	a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>			c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>			a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
	a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>			c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>			a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>	--	--

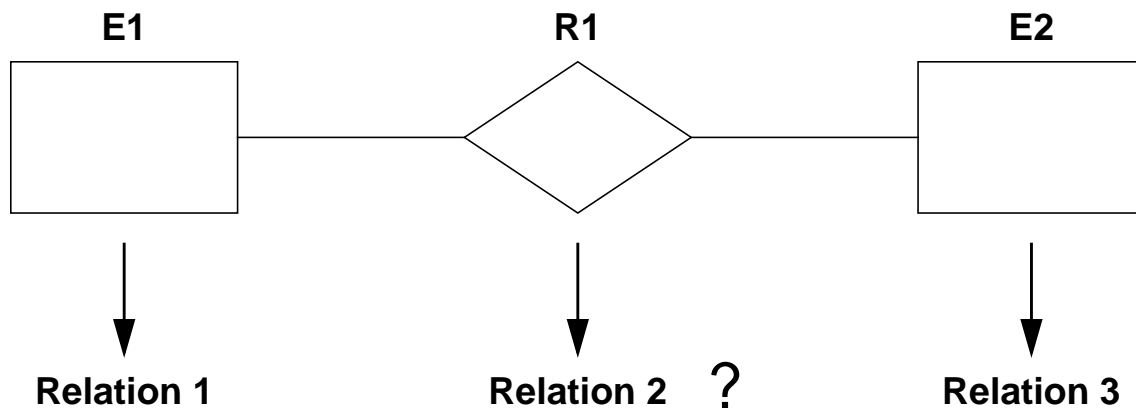
- Rechter äußerer Gleichverbund**

R	A	B	C	⊗	S	C	D	E	=	ERG	A	B	C	D	E
	a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>			c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>			a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
	a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>			c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>			--	--	c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>

- Äußerer Gleichverbund**

R	A	B	C	⊗	S	C	D	E	=	ERG	A	B	C	D	E
	a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>			c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>			a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
	a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>			c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>			a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>	--	--
											--	--	c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>

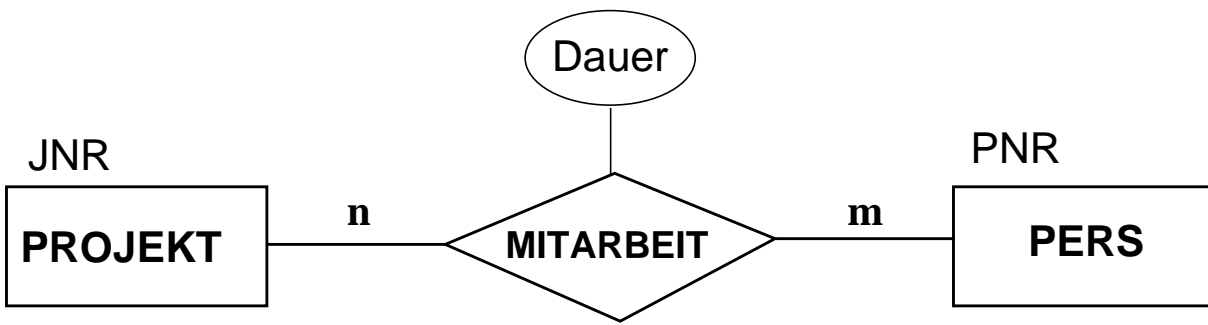
# Abbildung von Entity- und Relationship-Mengen



- **Kriterien**

- Informationserhaltung
  - Abbildung der ER-Konzepte auf RM-Konzepte
  - möglichst genaue Übereinstimmung der Semantik (Übernahme aller spezifizierten Eigenschaften)
- ➔ Die RM-Konzepte erreichen nicht das semantische Ausdrucksvermögen der ER-Konzepte. Deshalb gehen gewisse Aspekte der Semantik verloren. Es kann jedoch versucht werden, diese durch weitergehende Datenmodellkonzepte (siehe SQL) oder durch Anwendungsfunktionen nachzubilden.
- Minimierung der Redundanz
- Minimierung des Verknüpfungsaufwandes
- aber auch:
  - Natürlichkeit der Abbildung
  - keine Vermischung von Objekten
  - Verständlichkeit

## Zwei Entity-Mengen mit (n:m)-Verknüpfung



Verwendung von drei Relationen erforderlich

**PROJEKT** (JNR, BEZEICH, ...)

**PERS** (PNR, PNAME, ...)

**MITARBEIT** (JNR, PNR, Dauer)  
-----

- **Regel:**

Die Relationship-Menge wird auf eine Relation abgebildet, wobei der Primärschlüssel sich aus den Primärschlüsseln der beteiligten Entity-Mengen zusammensetzt. Alle Namen können übernommen werden; es ist jedoch auch eine Umbenennung möglich. Attributnamen in einer Relation müssen eindeutig sein.

## Zwei Entity-Mengen mit (1:n)-Verknüpfung



Verwendung von  
drei Relationen

**ABT** (ANR, ANAME, ...)

**ABT** (ANR, ANAME, ...)

zwei Relationen

**PERS** (PNR, PNAME, ...)

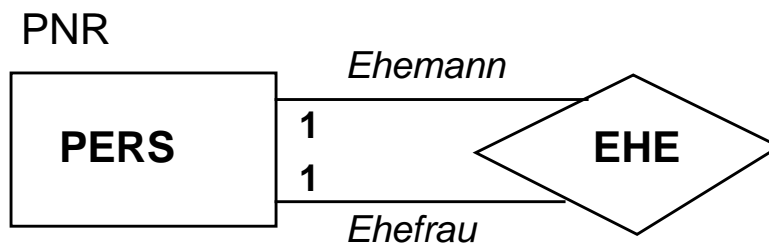
**PERS** (PNR, PNAME, ..., ANR)

**ABT-ZUGEH** (ANR, PNR)

- **Regel:**

(1:n)-Beziehungen lassen sich, wenn sie keine eigenen Attribute besitzen, ohne eigene Relation darstellen. Hierzu wird in der abhängigen Relation (mit Beziehungskardinalität 1) der Primärschlüssel der referenzierten Relation als Fremdschlüssel verwendet

## Eine Entity-Menge mit (1:1)-Verknüpfung



1.) Verwendung von zwei Relationen

**PERS** (PNR, PNAME, ...)

**EHE** (PNR, GATTE, ...)

2.) Verwendung von einer Relation

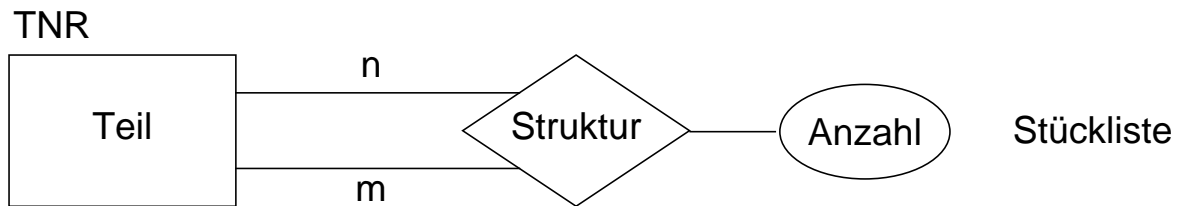
**PERS** (PNR, PNAME, ..., GATTE)

- **Regel:**

Der Primärschlüssel der zugehörigen Entity-Menge wird in zwei Rollen verwendet. Deshalb ist eine Umbenennung erforderlich.



## Eine Entity-Menge mit (m:n)-Verknüpfung



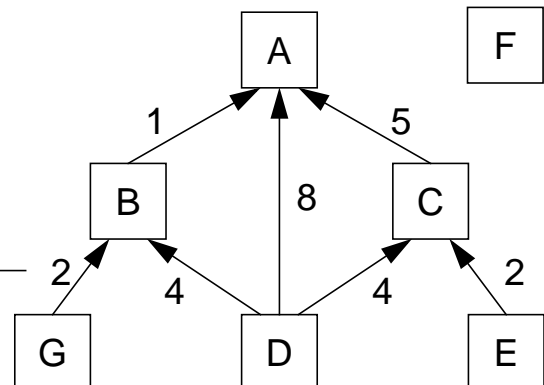
**Darstellungsmöglichkeit im RM:**

**TEIL** (TNR, BEZ, MAT, BESTAND)

**STRUKTUR** (OTNR, UTNR, ANZAHL)

TEIL	<u>TNR</u>	BEZ	MAT	BESTAND
	A	Getriebe	-	10
	B	Gehäuse	Alu	0
	C	Welle	Stahl	100
	D	Schraube	Stahl	200
	E	Kugellager	Stahl	50
	F	Scheibe	Blei	0
	G	Schraube	Chrom	100

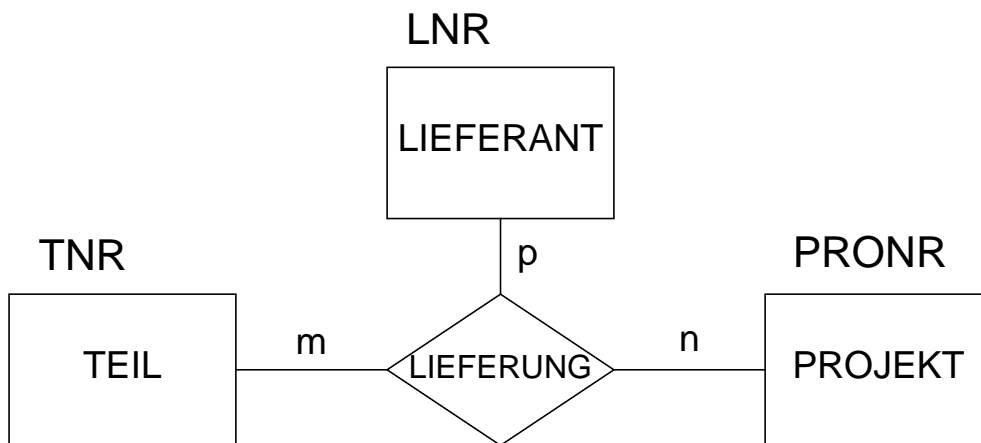
STRUKTUR	<u>OTNR</u>	<u>UTNR</u>	ANZAHL
	A	B	1
	A	C	5
	A	D	8
	B	D	4
	B	G	2
	C	D	4
	C	E	2



- Regel:**

Eine (n:m)-Relationship-Menge muss durch eine eigene Relation dargestellt werden. Der Primärschlüssel der zugehörigen Entity-Menge wird in zwei Rollen verwendet. Deshalb ist eine Umbenennung erforderlich.

## Mehrere Entity-Mengen mit (n:m)-Verknüpfung



### Darstellungsmöglichkeit im RM:

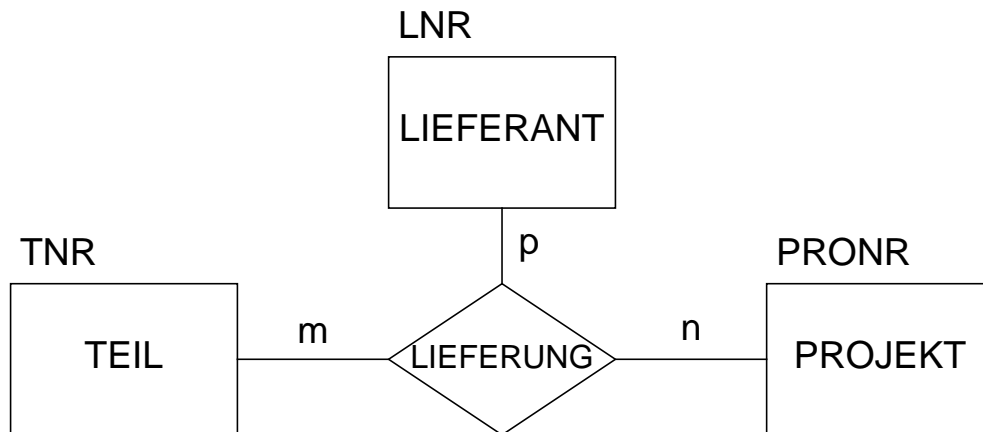
**LIEFERANT** (LNR, LNAME, LORT, ...)

**PROJEKT** (PRONR, PRONAME, PORT, ...)

**TEIL** (TNR, TBEZ, GEWICHT, ...)

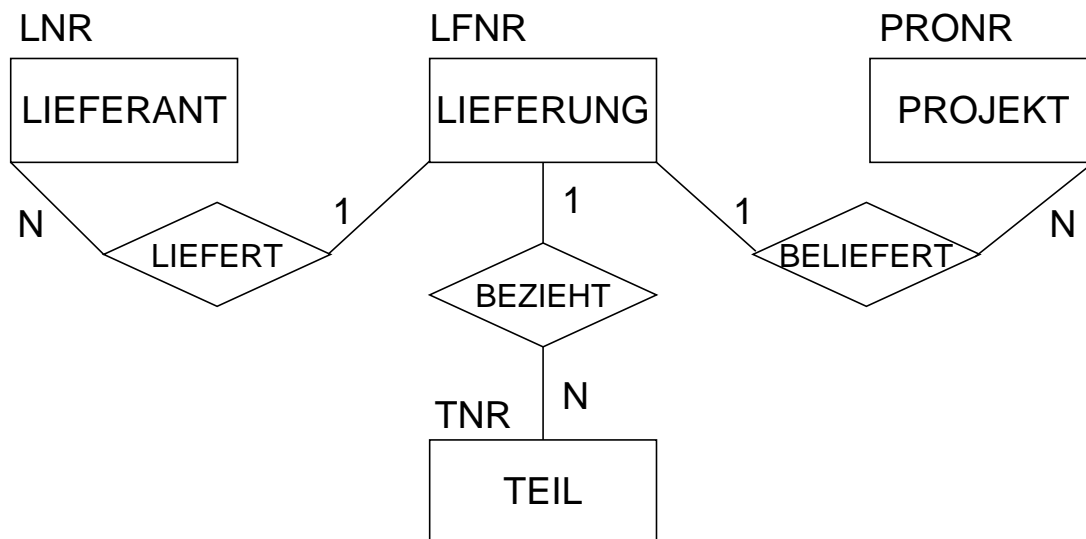
**LIEFERUNG** (LNR, PRONR, TNR, ANZAHL, DATUM)

## Mehrere Entity-Mengen mit (m:n)-Verknüpfung



➔ Manche Systeme erlauben nur die Modellierung binärer Relationship-Mengen!

- Wie lässt sich folgendes ER-Diagramm abbilden?



# Abbildungstypen innerhalb einer Entity-Menge

- Entity-Menge kann viele Attribute besitzen

**PERS** (PNR, NAME, ADRESSE, ..., GEHALT, VWL)

➔ ist weitere Zerlegung der Relation im DB-Schema sinnvoll?

- Horizontale Partitionierung

- Klassenbildung anhand von Selektionsbedingungen

**PERS-VIP** (PNR, ..., VWL)      GEHALT > 100K

**PERS** (PNR, ..., VWL)      GEHALT ≤ 100K

➔ ist eigentlich Aufgabe eines Sichtkonzeptes

- Vertikale Partitionierung

- zur leichteren Einhaltung von Leistungs- und Schutzaspekten:

**PERS-ÖFF** (PNR, PNAME, ADRESSE, ...)

**PERS-PRIV** (PNR, GEHALT, VWL, ...)

➔ ist Aufgabe des internen Schemas und des Sichtkonzeptes

## Abbildungstypen innerhalb einer Entity-Menge (2)

- Abbildung mehrwertiger Attribute

- Entity-Menge:

**PERS** (PNR, NAME, {Lieblingsessen}, {Kinder (Vorname, Alter)})  
P1, Müller, {Schnitzel, Braten, Rollmops}, -  
P2, Schulz, {Pizza}, {(Nadine, 5), (Philip, 2)}

- Darstellungsmöglichkeit im RM:

**PERS** (PNR, NAME ...)

**L.ESSEN** (PNR, GERICHT, )  
-----

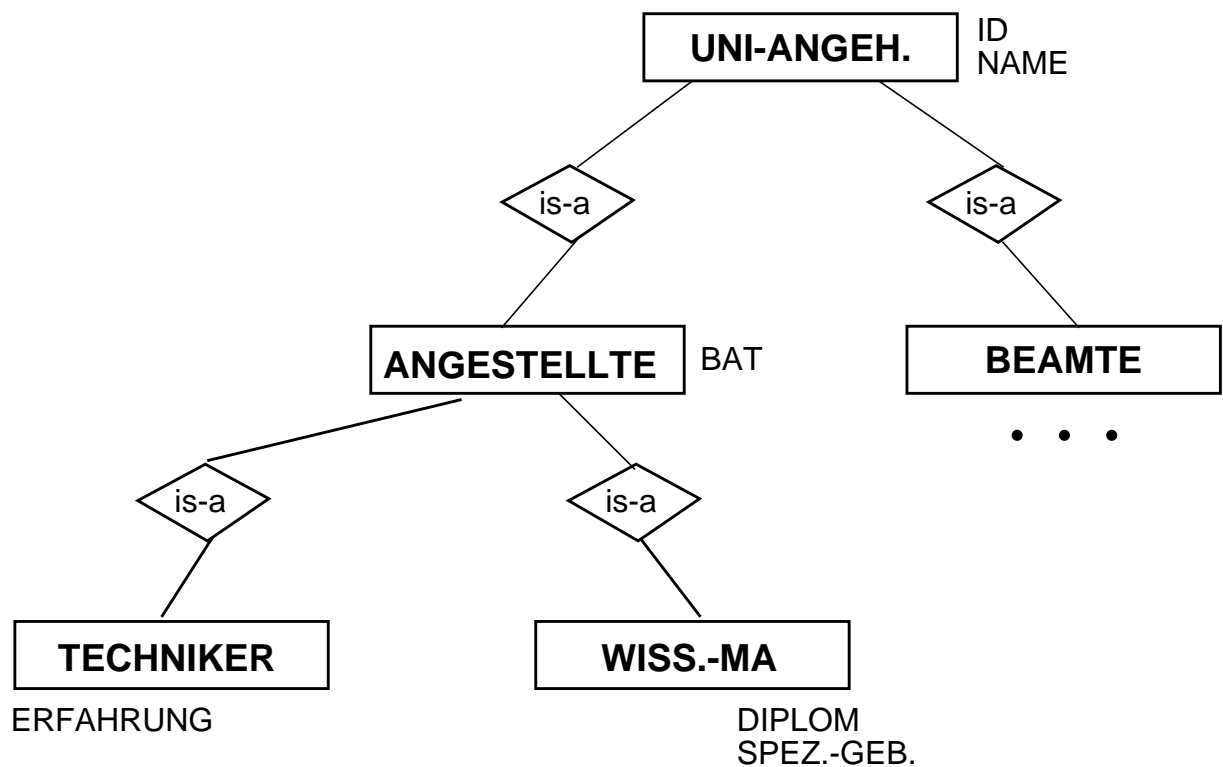
**KINDER** (PNR, VORNAME, ALTER)  
-----

## Abbildung der Generalisierung im RM

- **Einschränkungen des Relationenmodells**

- keine Unterstützung der Abstraktionskonzepte
- keine Maßnahmen zur Vererbung  
(von Struktur, Integritätsbedingungen, Operationen)
- „Simulation“ der Generalisierung eingeschränkt möglich

### Generalisierungsbeispiel:



## Generalisierung – relationale Sicht

- **LÖSUNG 1: Hausklassen-Modell:**

- Jede Instanz ist genau einmal und vollständig in ihrer Hausklasse gespeichert
- Es wird eine horizontale Partitionierung der DB-Instanzen erreicht

UNI-ANGEH.	ID	NAME
	111	Ernie

ANGESTELLTE	ID	NAME	BAT
	007	Garfield	Ia

TECHNIKER	ID	ERFAHRUNG	NAME	BAT
	123	SUN	Donald	IVa

WISS.-MA.	ID	DIPLOM	SEPZ.-GEB.	NAME	BAT
	333	Informatik	Recovery	Daisy	Ila
	765	Mathematik	ERM	Grouch	Ila

- **Eigenschaften:**

- niedrige Speicherkosten und keine Änderungsanomalien
- Retrieval kann rekursives Suchen in Unterklassen erfordern
- explizite Rekonstruktion durch Relationenoperationen ( $\pi$ ,  $\cup$ )

➔ **Beispiel: Finde alle ANGESTELLTE:**

$\pi_{ID, NAME, BAT}(TECHNIKER) \cup \pi_{ID, NAME, BAT}(WISS.-MA) \cup ANGESTELLTE$

## Generalisierung – relationale Sicht

- LÖSUNG 2: Partitionierungs-Modell**

- Jede Instanz wird entsprechend der Klassenattribute in der Is\_a-Hierarchie zerlegt und in Teilen in den zugehörigen Klassen gespeichert
- Es wird nur das ID-Attribut dupliziert
- Es wird eine vertikale Partitionierung in der DB erzielt

UNI-ANGEH.	ID	NAME	ANGESTELLTE	ID	BAT
	007	Garfield		007	Ia
	111	Ernie		123	IVa
	123	Donald		333	Ila
	333	Daisy		765	Ila
	765	Grouch			

TECHNIKER	ID	ERFAHRUNG
	123	SUN

WISS.-MA	ID	DIPLOM	SPEZ.-GEB
	333	Informatik	Recovery
	765	Mathematik	ERM

- Eigenschaften**

- geringfügig erhöhte Speicherkosten, aber hohe Aufsuch- und Aktualisierungskosten
- Integritätsbedingungen:  $TECHNIKER.ID \subseteq ANGESTELLTE.ID$  usw.
- Instanzenzugriff erfordert implizite oder explizite Verbundoperationen ( $\bowtie$ )

➔ **Beispiel: Finde alle TECHNIKER-Daten**

TECHNIKER  $\bowtie_{ID=ID}$  ANGESTELLTE  $\bowtie_{ID=ID}$  UNI-ANGEH.



## Generalisierung – relationale Sicht

- **Lösung 3: Volle Redundanz**

- Eine Instanz wird wiederholt in jeder Klasse, zu der sie gehört, gespeichert
- Sie besitzt dabei die Werte der Attribute, die sie geerbt hat, zusammen mit den Werten der Attribute der Klasse

UNI-ANGEH.	ID	NAME	ANGESTELLTE	ID	NAME	BAT
	007	Garfield		007	Garfield	Ia
	111	Ernie		123	Donald	IVa
	123	Donald		333	Daisy	Ila
	333	Daisy		765	Grouch	Ila
	765	Grouch				

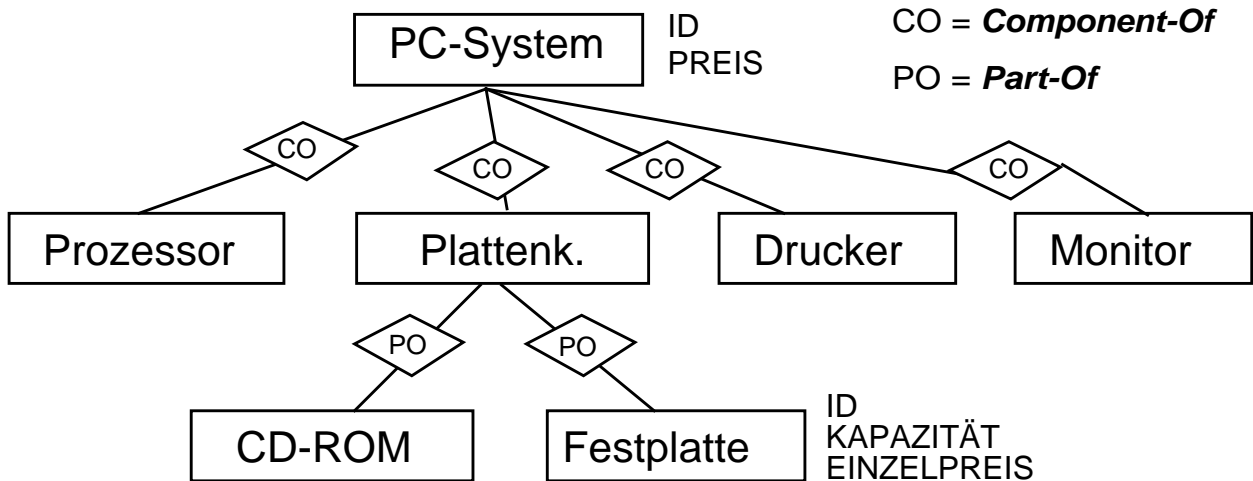
TECHNIKER	ID	NAME	BAT	ERFAHRUNG
	123	Donald	IVa	SUN

WISS.-MA	ID	NAME	BAT	DIPLOM	SPEZ.-GEB.
	333	Daisy	Ila	Informatik	Recovery
	765	Grouch	Ila	Mathematik	ERM

- **Eigenschaften**

- sehr hoher Speicherplatzbedarf und Auftreten von Änderungsanomalien
- sehr einfaches Retrieval, da nur die Zielklasse (z. B. ANGESTELLTE) aufgesucht werden muss

## Aggregation - relationale Sicht



PC-System	PREIS	PROZESSOR	PLATTENK.	DRUCKER	MONITOR
PC-1	900,-	I-Pentium IV 1,8 GHz	PK1	Tintenstrahldr.	15" TFT
PC-2	1700,-	I-Pentium IV 2,4 GHz	PK2	Laserdrucker	17" TFT
PC-3	2200,-	I-Pentium IV 3,0 GHz	PK3	Laserdrucker	17" TFT

Plattenk.	ID	CD-ROM	Festplatte
	PK2	C1	F1
	PK2	C2	F2
	PK3	C3	F3

Monitor	ID	Hersteller	Preis
	15"	Belinea	320,-
	17"	Belinea	420,-
	17"	Belinea	420,-

CD-ROM	ID	Geschwindigk.	Preis
	C1	DVD R	35.-
	C2	DVD R+CD RW	85.-
	C3	DVD R+RW	165.-

Festplatte	ID	Kapaz.	Preis
	F1	80 GB	85.-
	F2	120 GB	100.-
	F3	160 GB	150.-

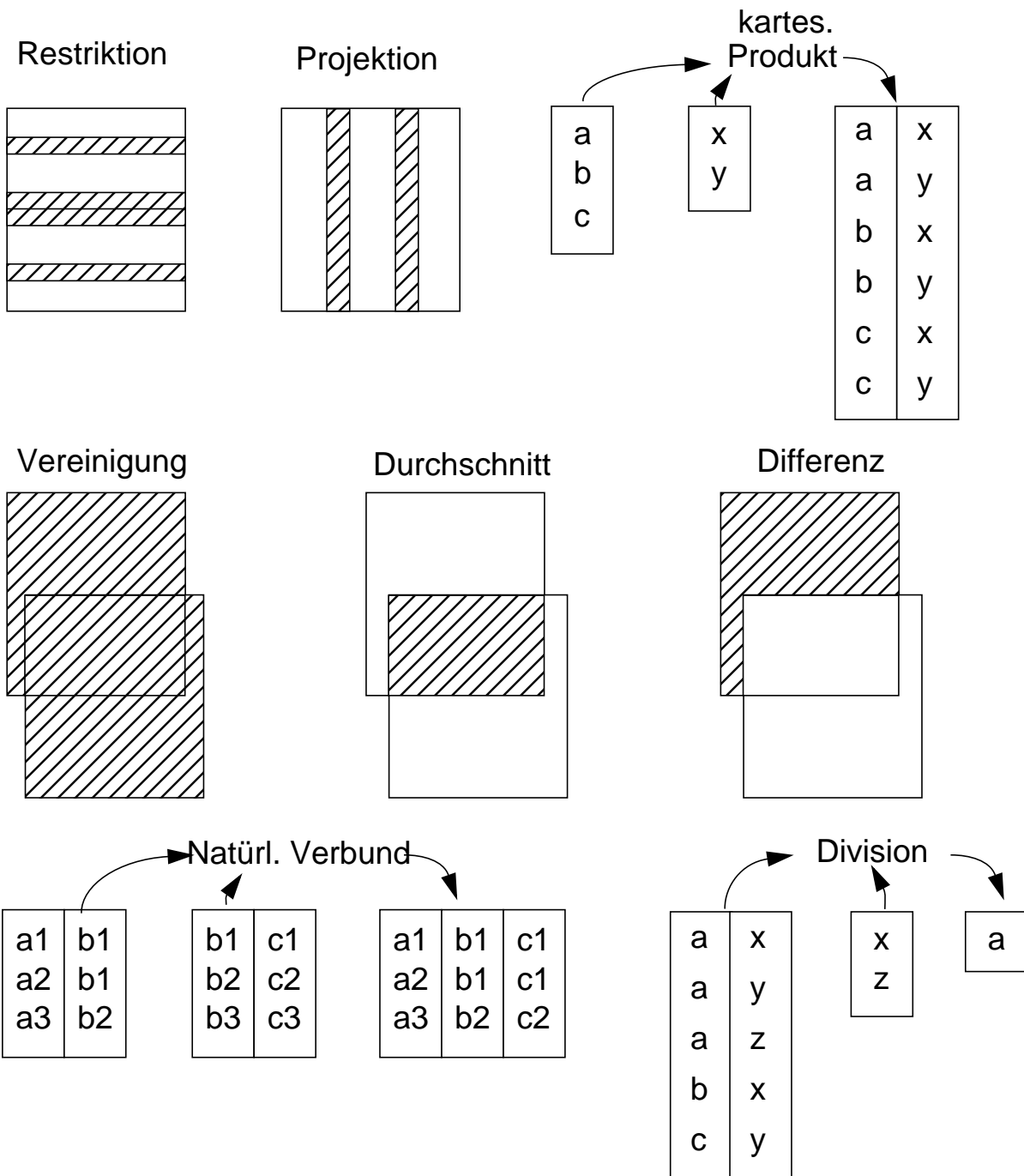
Prozessor	ID	SPEC int 2000	Preis
	I-Pentium IV 1,8 GHz	633	150.-
	I-Pentium IV 2,4 GHz	1034	190.-
	I-Pentium IV 3,0 GHz	1226	310.-

Drucker	ID	Hersteller	Typ	Preis
	Tintenstrahldr.	HP	Desk Jet 5150	100.-
	Laserdrucker	HP	Laser 1300	360.-

➔ Eigenschaften der Aggregation werden durch relationale Operationen nicht unterstützt

# Zusammenfassung: Relationenalgebra

- Algebra mit Auswahlvermögen der Prädikatenlogik 1. Stufe
- Abgeschlossenheit bzgl. der Algebraoperationen
- Klassische Mengenoperationen
- Relationenoperationen



# Zusammenfassung: Abbildungskonzepte

- **Datenstruktur**

Relation (Tabelle)


- ➔ einzige Datenstruktur (neben atomaren Werten)
  - ➔ alle Informationen ausschließlich durch Werte dargestellt
  - ➔ Integritätsbedingungen auf/zwischen Relationen: relationale Invarianten
- **Abbildung von Beziehungen durch PS/SK – FS**
    - alle Beziehungen sind explizit, binär und symmetrisch
    - alle Beziehungstypen müssen im Prinzip durch (n:1)-Beziehungen dargestellt werden
    - (n:m)-Beziehungstypen sind durch eine eigene Relation darzustellen
    - ein (n:1)-Beziehungstyp wird in der Regel nur dann auf eine eigene Relation abgebildet, wenn er beschreibende Attribute besitzt
  - **Bewertung hinsichtlich der Abstraktionskonzepte**
    - keine direkte Bereitstellung der Abstraktionskonzepte (nur Klassifikation der Tupeln in Relationen)
    - begrenzte Möglichkeiten zur Abbildung der Abstraktionskonzepte