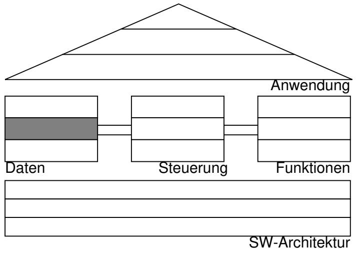
4. Grundlagen des Relationenmodells

- **GBIS-Rahmen: Einordnung**

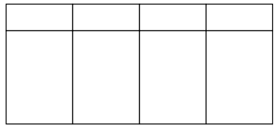


- **Übersicht**
  - Grundkonzepte
  - Normalisierte Relationen
  - Schlüssel
  - Sprachen für das Relationenmodell
- **Relationenalgebra**
  - Klassische Mengenoperationen
  - Relationenoperationen
  - Anfragen
- **Relationenalgebra – Optimierung**
  - Rewrite-Regeln
  - Algebraische Optimierung: Beispiel
- **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)
- **Entwurfstheorie**
  - Normalformenlehre (wünschenswerte und zweckmäßige Relationen)
  - Synthese von Relationen

Relationenmodell – Grundkonzepte[1]

ERM	RM
Einwertiges Attribut Definitionsbereich Primärschlüssel	} wie im ERM
Zusammengesetztes Attribut Mehrwertiges Attribut	nur als unabhängige einwertige Attribute
Entity-Menge Relationship-Menge	→ Relation

- **Def.: normalisierte Relation**

$$R(A_1, A_2, \ldots, A_n) \subseteq W(A_1) \times W(A_2) \times \ldots \times W(A_n)$$

$W(A_1) \to D_i$, $W(A_2) \to D_j$, $W(A_n) \to D_k$

Untermenge des kartesischen Produktes

- **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

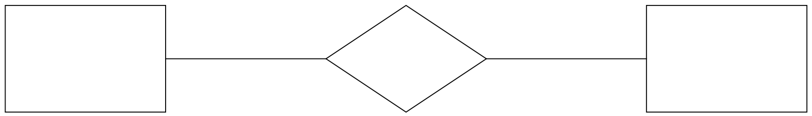
PROF	<u>PNR</u>	PNAME	FBNR	FACHGEBIET
	1234	Härder	FB 5	Datenbanksysteme
	5678	Wedekind	FB 9	Informationssysteme
	4711	Müller	FB 9	Operations Research
	6780	Nehmer	FB 5	Betriebssysteme
	2223	Richter	FB 5	Expertensysteme

- **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 (wertbasiert)
  - 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.
   (In SQL2 beschreibt die „exotische“ MATCH-Option die Semantik partiell definierter Fremdschlüssel).
5. Eine Relation kann mehrere Fremdschlüssel besitzen, welche die gleiche oder verschiedene Relationen referenzieren.
6. Referenzierte und referenzierende Relation sind nicht notwendig verschieden (**„self-referencing table“**).
7. 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)

- **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 σ_P**

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

a) Operanden
- Konstanten
- Attributnamen

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

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

- **Definition:**

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

Beispiele:

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

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

- **Anwendung von:** $\sigma_{ANR='K55' \wedge GEHALT > 50\,000}$ (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) $A_1, A_2, \ldots, A_k$ aus einer Relation R (Grad n >= k)**

$$\boxed{\begin{aligned} &\pi_{A_1, A_2, \ldots, A_k}(R) \\ &= \{ p \mid \exists t \in R : p = \langle t[A_1], \ldots, t[A_k] \rangle \} \end{aligned}}$$

! Duplikate entfernt !

- **Alternativ: Benutzung der Spaltennummern j_i**

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

- **Beispiel:**

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

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

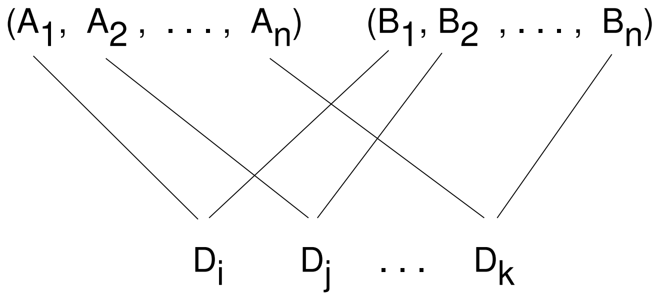
2. Find alle Angestellten (PNR, NAME) aus Abteilung K55, die mehr als 40.000 DM verdienen

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

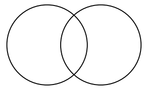


➥ $W(A_i) = W(B_i) \quad : \quad i = 1, n$

1. Vereinigung (UNION) von R und S

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

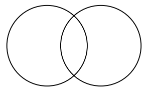
$R \cup S$



2. Differenz

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

$R - S$

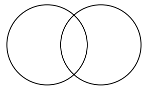


<u>zusätzlich</u> (redundante Mengenoperationen):

3. Durchschnitt (INTERSECTION)

$$R \cap S = R - (R - S) = \{t \mid t \in R \wedge t \in S\}$$

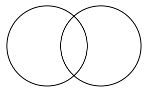
$R \cap S$



4. Symmetrische Differenz

$$R \Delta S = (R \cup S) - (R \cap S) = ((R \cup S) - (R - (R - S))) = \{t \mid t \in R \oplus t \in S\}$$

$R \Delta S$



(Erweitertes) Kartesisches Produkt

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

$$K = R \times S$$

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

$k = x|y = <x_1, \ldots, x_r, y_1, \ldots, y_s>$

<u>nicht</u> $<<x_1, \ldots, x_r>, <y_1, \ldots, y_s>>$ wie übliches kart. Produkt

- **Anwendung von: ABT x PERS**

ABT	<u>ANR</u>	ANAME	AORT
	K51	Planung	KL
	K53	Einkauf	F
	K55	Vertrieb	F

PERS	<u>PNR</u>	ALTER	ANR
	406	47	K55
	123	32	K51
	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, Θ-Join)

- **Grob:**

  Kartesisches Produkt zwischen zwei Relationen R (Grad r) und S (Grad s).

  eingeschränkt durch Θ-Bedingungen zwischen i-Spalte von R und j-Spalte von S.

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

Θ-Verbund zwischen R und S:

$$V = R \underset{i\Theta j}{\bowtie} S$$

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

- **Bemerkungen:**

(1) Speziell Θ = ’=’ : Gleichverbund (Equi-Join)

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

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

(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:**

$$\mathbf{ABT} \underset{ANR = ANR}{\bowtie} \mathbf{PERS}$$

ABT	<u>ANR</u>	ANAME	AORT
	K51	Planung	KL
	K53	Einkauf	F
	K55	Vertrieb	F

PERS	<u>PNR</u>	ALTER	ANR
	406	47	K55
	123	32	K51
	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:** $\pi_{ANR,\ ANAME,\ AORT}(R) = ABT$

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

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

➥ π als Umkehroperation führt nicht auf ABT oder PERS

Natürlicher Verbund (*Natural Join*)

- **grob:**

  Gleichverbund über <u>alle</u> gleichen Attribute und Projektion über die verschiedenen Attribute

- **gegeben:** $R(A_1, A_2, \ldots, A_{r-j+1}, \ldots, A_r)$
  $S(B_1, B_2, \ldots, B_j, \ldots, B_s)$

  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:

$$N = R \bowtie S$$

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

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

- **Bemerkung:**
  Attribute sind durch Übereinstimmungsbedingung gegeben

Natürlicher Verbund – Beispiel

- **Anwendung von:**

ABT ⋈ PERS

ABT	<u>ANR</u>	ANAME	AORT
	K51	Planung	KL
	K53	Einkauf	F
	K55	Vertrieb	F

PERS	<u>PNR</u>	ALTER	ANR
	406	47	K55
	123	32	K51
	829	36	K53
	574	28	K55

AP = ABT ⋈ 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:** $\pi_{\text{ANR, ANAME, AORT}}$ (AP) = ABT

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

- **Verlustbehafteter natürlicher Verbund analog zu Gleichverbund**

Natürlicher Verbund - Beispiel (2)

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

- **Annahmen:**
  - ABT: N/10 Tupel
  - PERS: N Tupel
  - Gleichverteilung der Attributwerte
    AORT: 20 Werte
    ALTER: 50 Werte
  - Stochastische Unabhängigkeit der Werte verschiedener Attribute
  - Verlustfreie Verbunde von R1 und R2 über Primär-/Fremdschlüssel, mit Card(R1) < Card(R2): Card(R1 ⋈ R2) = Card(R2)

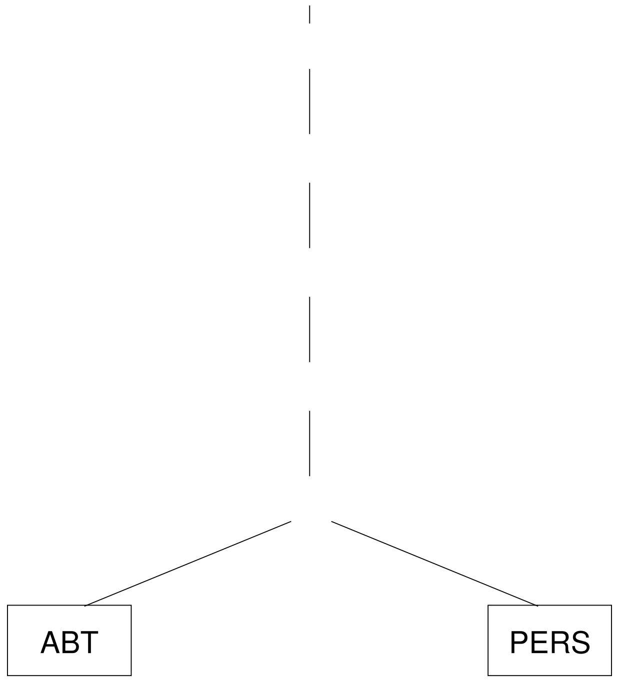
- **Anfrage:**
  Finde alle Angestellten (PNR, ALTER, ANAME), die in einer Abteilung in Frankfurt arbeiten und zwischen 30 und 34 Jahre alt sind.

Natürlicher Verbund – Lösungen

- **Lösung 1:**

  $\pi_{PNR,ALTER,ANAME}$
  $(\sigma_{AORT='F'}(\sigma_{ALTER \geq 30 \wedge ALTER \leq 34}(\sigma_{ABT.ANR=PERS.ANR}(ABT \times PERS))))$

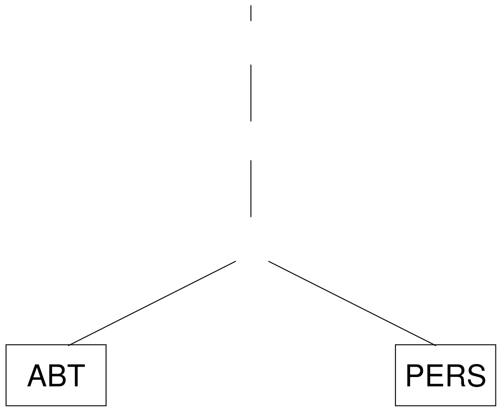
- **Zugehöriger Operatorbaum**



Natürlicher Verbund – Lösungen (2)

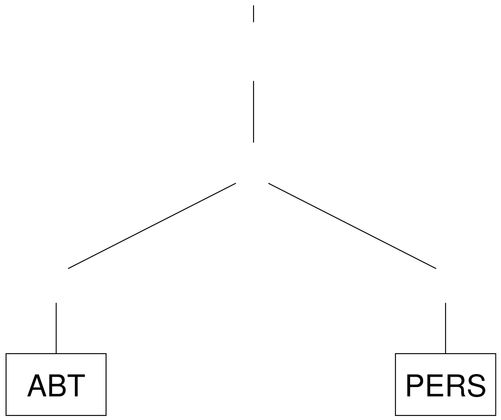
- **Lösung 2:**

  $\pi_{PNR,ALTER,ANAME}$
  $(\sigma_{ALTER \geq 30 \wedge ALTER \leq 34 \wedge AORT='F'}(ABT \bowtie PERS))$



- **Lösung 3:**

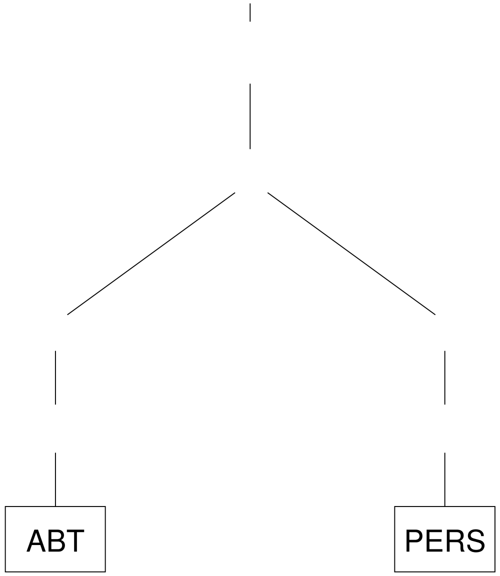
  $\pi_{PNR,ALTER,ANAME}$
  $((\sigma_{AORT='F'}ABT) \bowtie (\sigma_{ALTER \geq 30 \wedge ALTER \leq 34}PERS))$



Natürlicher Verbund – Lösungen (3)

- **Lösung 4:**

$\pi_{PNR,ALTER,ANAME}((\pi_{ANR,ANAME}$
$(\sigma_{AORT='F'}ABT))\bowtie(\pi_{PNR,ALTER,ANR}(\sigma_{ALTER \geq 30 \wedge ALTER \leq 34}PERS)))$



Natürlicher Verbund - Beispiel (3)

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

- **Beispiel 1 (1:n):**

AP1 $= \pi_{ANR,\ ANAME,\ AORT}$ (AP) **AP3** $=$ AP1 $\bowtie$ AP2 $\overset{!}{=}$ 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 $= \pi_{PNR,\ A\text{-}ORT}$ (DA)

= DA1	PNR	A-ORT

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

= DA2	FIGUR	A-ORT

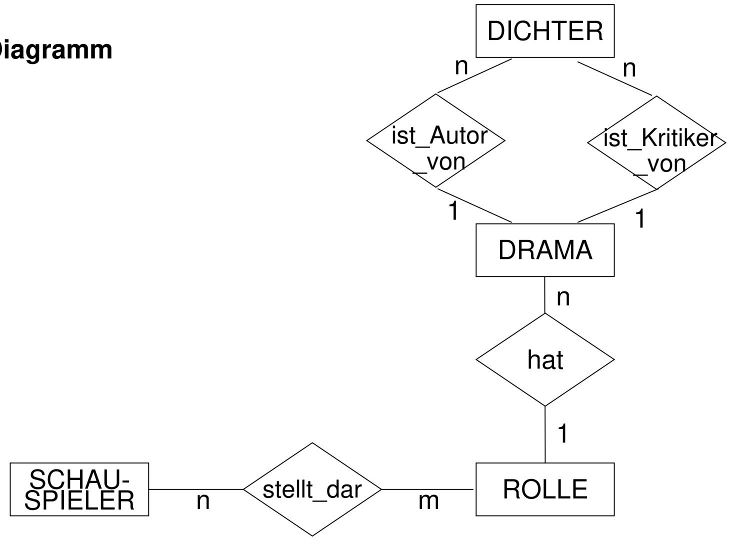
DA3 $=$ DA1 $\bowtie$ DA2

= DA3	PNR	FIGUR	A-ORT

➥ „Connection Trap“ bei Projektion von Schlüsselteilen und nachfolgendem Verbund

Beispiel-DB: BÜHNE

ER-Diagramm



Relationales Schema

DICHTER (DI)

AUTOR	G-ORT	G-JAHR

DRAMA (DR)

TITEL	AUTOR	KRITIKER	U-ORT	U-JAHR

SCHAUSPIELER (SP)

PNR	NAME	W-ORT . . .

ROLLE (RO)

FIGUR	TITEL	R-Typ

DARSTELLER (DA)

PNR	FIGUR	A-JAHR	A-ORT	THEATER

Beispiel-DB: Anfragen

Q1: Finde alle Schauspieler (NAME), die einmal den ’Faust’ (die Figur ’Faust’) gespielt haben.

Q2: Finde alle Schauspieler (NAME), die einmal im ’Faust’ (im Drama ’Faust’) mitgespielt haben.

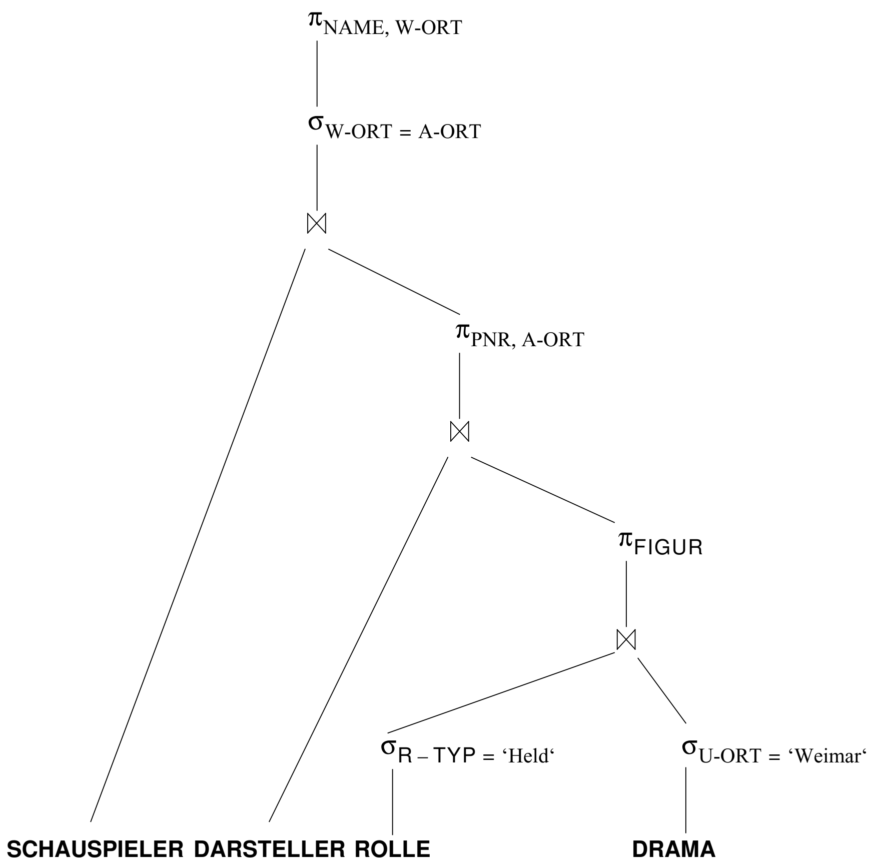
Q3: Finde alle Schauspieler (NAME), die in Dramen von Schiller mitgespielt haben.

Q4: Kann die Frage beantwortet werden: Welcher Dichter ist Schauspieler? oder: Welcher Dichter hat in einem seiner eigenen Stücke gespielt?

Anfragedarstellung als Operatorbaum

Q5: Finde alle Schauspieler (NAME, W-ORT), die bei in Weimar uraufgeführten Dramen an ihrem Wohnort als 'Held' mitgespielt haben.

Q5 = $\pi_{\text{NAME, W-ORT}}\ (\sigma_{\text{W-ORT= A-ORT}}\ (\text{SCHAUSPIELER} \bowtie ($
$\pi_{\text{PNR, A-ORT}}\ (\text{DARSTELLER} \bowtie (\pi_{\text{FIGUR}}\ ($
$(\sigma_{\text{R-TYP = 'Held}}\ \text{ROLLE}) \bowtie (\sigma_{\text{U-ORT = 'Weimar'}}\ \text{DRAMA})\ )\ )\ )\ )\ )\ )$



Anfragen (2)

Q6: Liste alle Dramen mit ihren Autoren (mit TITEL, AUTOR, G-JAHR) auf, die nach 1800 uraufgeführt wurden.

Q7: Liste alle Dramen mit ihren Kritikern, die in Weimar geboren wurden, (mit TITEL, KRITIKER) auf.

Q8: Finde die Schauspieler (PNR), die **nie** gespielt haben.

Q9: Finde die Schauspieler (PNR), die **nur** Faust oder Wallenstein gespielt haben.

Relationenalgebra – Optimierung

- Relationenalgebraische Formulierungen spezifizieren Ausführungs-reihenfolge (prozedurale Elemente)

  → jedoch äquivalente Umformungen möglich

- **Problem**
  - gegeben: Ausdruck der Relationenalgebra (RA)
  - gesucht: äquivalenter, möglichst effizient auszuführender Ausdruck der Relationenalgebra

- Bestimmung einer **möglichst guten Ausführungsreihenfolge** (Einsatz von Heuristiken) für
  - unäre Operationen: π, σ
  - binäre Operationen: $\cap$, $\cup$, -, $\times$, $\bowtie$, $\div$

- **Statistische Kenngrößen** werden dem DB-Katalog entnommen
  - Ni = Card(Ri)
  - ji = Anzahl der verschiedenen Werte eines Attributs Ai

- **Algebraische Optimierung - Beispiel:**

Datenbank:

ABT (<u>ANR</u>, BUDGET, A-ORT)

PERS (<u>PNR</u>, NAME, BERUF, GEHALT, ALTER, ANR)

PM (<u>PNR,JNR)</u>, DAUER, ANTEIL)

PROJ (<u>JNR</u>, BEZEICHNUNG, SUMME, P-ORT)

Rewrite-Regeln der Relationenalgebra[1]

Äquivalenz von Ausdrücken

- Umformung von Ausdrücken zur besseren Auswertung
- Zusammenstellung wichtiger Regeln
- Ri: Relationen (oder relationenalgebraische Ausdrücke)

1. Kommutatives Gesetz für Verbunde und Produkte

$R1 \underset{F}{\bowtie} R2 \equiv R2 \underset{F}{\bowtie} R1$

$R1 \bowtie R2 \equiv R2 \bowtie R1$

$R1 \times R2 \equiv R2 \times R1$

2. Assoziatives Gesetz für Verbunde und Produkte

$(R1 \underset{F1}{\bowtie} R2) \underset{F2}{\bowtie} R3 \equiv R1 \underset{F1}{\bowtie} (R2 \underset{F2}{\bowtie} R3)$

$(R1 \times R2) \times R3 \equiv R1 \times (R2 \times R3)$

3. Folgen von Projektionen

$\pi_{A, B, C}(\pi_{A, B, C, \ldots, Z}(R)) = \pi_{A, B, C}(R)$

4. Folgen von Selektionen

$\sigma_{F1}(\sigma_{F2}(R)) = \sigma_{F1 \wedge F2}(R)$

Da $(F1 \wedge F2 = F2 \wedge F1)$:

$\sigma_{F1}(\sigma_{F2}(R)) = \sigma_{F2}(\sigma_{F1}(R))$

1. Jarke, M., Koch, J.: Query Optimization in Database Systems, in: Computing Surveys 16:2, 1984, pp. 111-152.

Rewrite-Regeln der Relationenalgebra (2)

5. **Vertauschung von Selektionen und Projektionen**

   F enthält nur Attribute aus A . . . Z:

   $\sigma_F(\pi_{A, \ldots, Z}(R)) \equiv \pi_{A, \ldots, Z}(\sigma_F(R))$

   wenn F auch Attribute aus B_1 . . . B_m enthält:

   $\pi_{A, \ldots, Z}(\sigma_F(R)) \equiv \pi_{A, \ldots, Z}(\sigma_F(\pi_{A, \ldots, Z, B_1, \ldots, B_m}(R)))$

6. **Vertauschung von Selektion und Kartesischem Produkt**

   F enthält nur Attribute aus R1

   $\sigma_F(R1 \times R2) = \sigma_F(R1) \times R2$

   allgemeiner:

   F = F1 $\wedge$ F2 $\wedge$ F3
   F1: nur Attribute aus R1
   F2: nur Attribute aus R2
   F3: beides

   $\sigma_F(R1 \times R2) = \sigma_{F1}(R1) \underset{F3}{\bowtie} \sigma_{F2}(R2)$

Optimierung – Berechnungsgrundlagen

- **Allgemeine Annahmen**
  - Gleichverteilung der Attributwerte eines Attributes
  - Stochastische Unabhängigkeit der Werte verschiedener Attribute

Mit Hilfe von statistischen Werten kann der Optimizer jedem Qualifikationsprädikat einen Selektivitätsfaktor ($0 \le SF \le 1$) zuordnen (erwarteter Anteil an Tupeln, die das Prädikat erfüllen): Card (σ_p(R)) = SF(p) · Card (R)

- **Selektivitätsfaktor SF bei:**

$A_i = a_i$	SF =	$1/j_i$	wenn bekannt, z. B. Index auf A_i
		1/10	sonst
$A_i = A_k$	SF =	1 / Max(j_i, j_k)	wenn Index auf A_i, A_k
		1 / j_i	wenn Index auf A_i
		1/10	sonst
$A_i \ge a_i$	SF =	(high-key - a_i) / (high-key - low-key)	bei linearer Interpolation
		1/3	sonst
$A_i \ge a_i \wedge A_i \le a_k$	SF =	(a_k-a_i) / (high-key - low-key)	Index auf A_i
		1/4	sonst

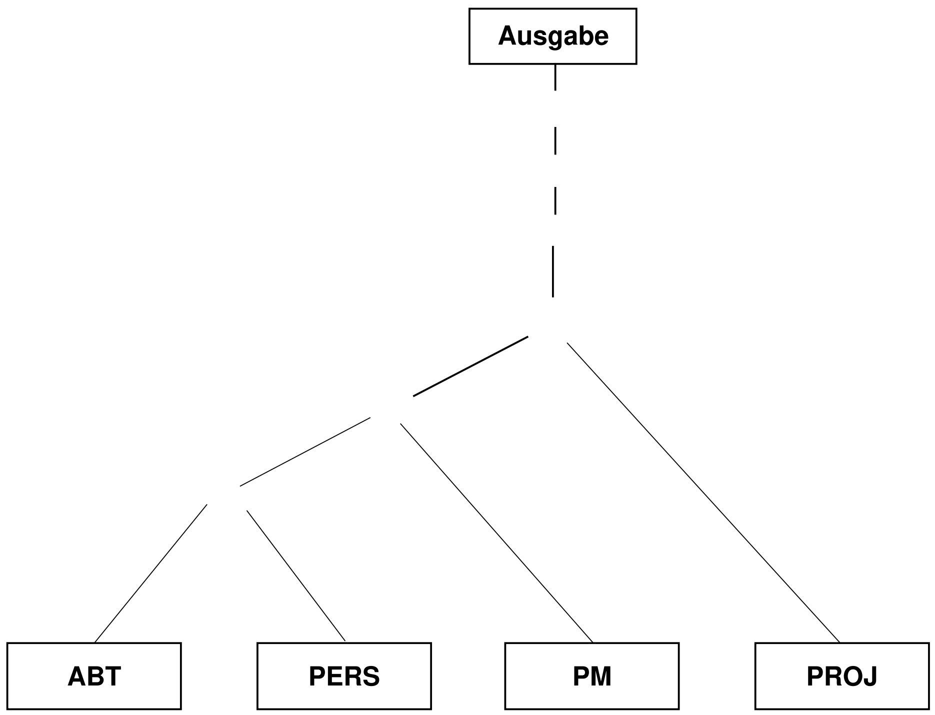
- **Berechnung von Ausdrücken**
  - SF (p(A) $\wedge$ p(B)) = SF (p(A)) · SF (p(B))
  - SF (p(A) $\vee$ p(B)) = SF (p(A)) + SF (p(B)) - SF (p(A)) · SF (p(B))
  - SF ($\neg$ p(A)) = 1 - SF (p(A))

- **Join-Selektivitätsfaktor (JSF)**
  - Card (R$\bowtie$S) = JSF * Card(R) * Card(S)
  - bei (N:1)-Joins (verlustfrei)
    Card (R$\bowtie$S) = Max(Card(R), Card(S))

Algebraische Optimierung – Beispiel

- **Anfrage:**
  Finde Name und Beruf von Angestellten, deren Abteilung in KL ist
  und die in KL Projekte durchführen

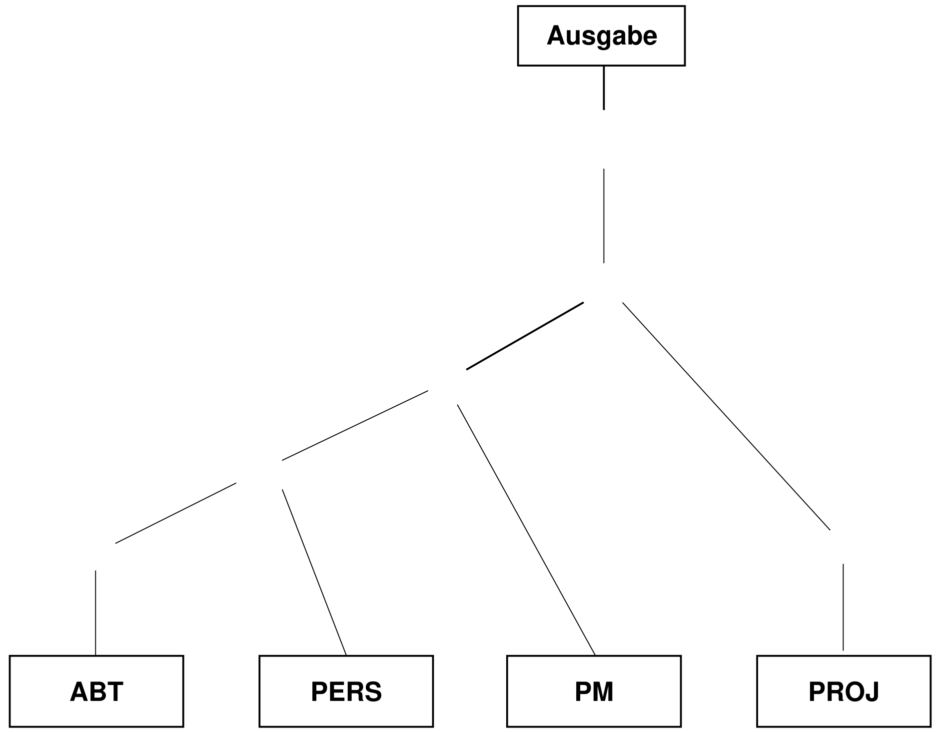
1. Ausgangslösung



- **Annahmen:**
  - ABT: N/5 Tupel
  - PERS: N Tupel
  - PM: 5·N Tupel
  - PROJ: M Tupel
  - Anzahl der Attributwerte von
    A-ORT: 10 Werte
    P-ORT: 100 Werte
  - Verlustfreie Verbunde von R1 und R2 über Primär-/Fremdschlüssel

Algebraische Optimierung – Beispiel (2)

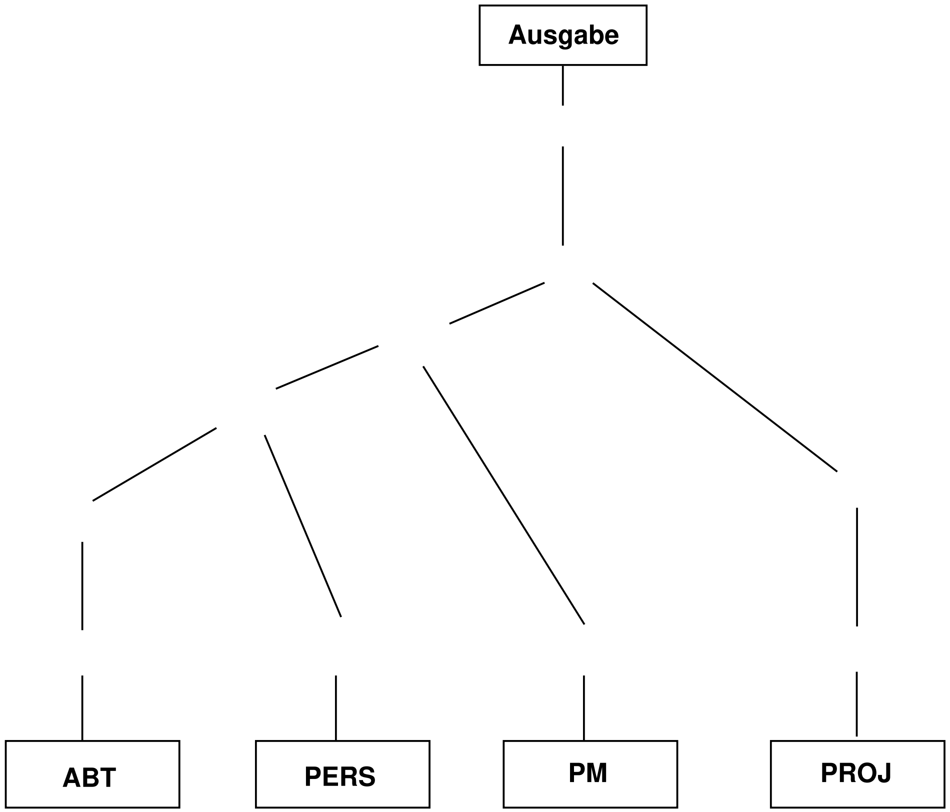
2. Verschieben der Selektion zu den Blattknoten



⇒ ***I. Führe Selektionen so früh wie möglich aus!***

Algebraische Optimierung – Beispiel (3)

3. Verschieben der Projektion



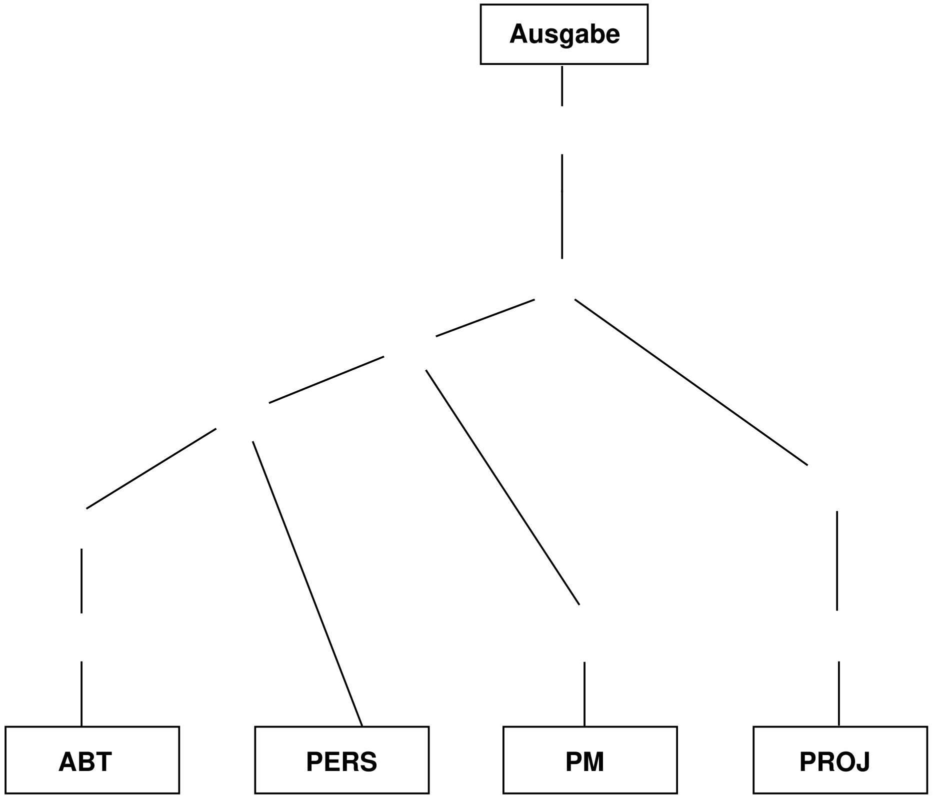
⇒ **II.** ***Führe Projektion (ohne Duplikateliminierung) frühzeitig aus***

(<u>Eliminierung</u> von Duplikaten ist in der Regel sehr teuer)

- Bem.: Der Nutzen einer frühzeitigen Projektionsausführung hängt von mehreren Faktoren ab.

Algebraische Optimierung – Beispiel (4)

4. Optimierter Operatorbaum – Vorschlag



- Alternative Möglichkeit:

  Zusammenfassen von

  $(PM) \bowtie (\pi_{JNR}(\sigma_{P\text{-}ORT = \text{'KL'}} PROJ))$

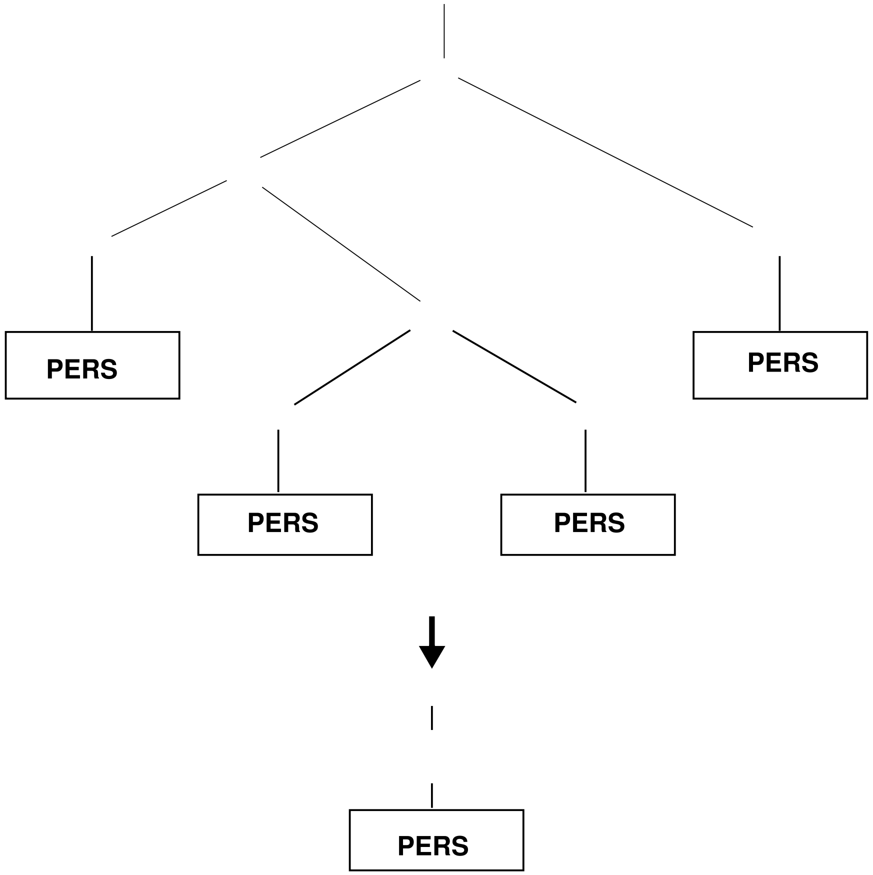
⇒ **III.** ***Verknüpfe Folgen von unären Operationen wie Selektion und Projektion*** **(wenn diese tupelweise abgewickelt werden können)**

Algebraische Optimierung – Beispiel (5)

5. Weitere Optimierungsmaßnahmen

Ausdruckauswertung:

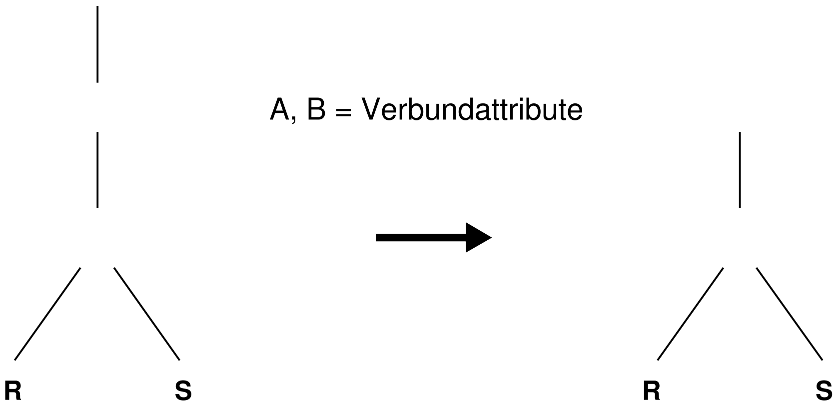
Alle Programmierer mit mehr als 50K Gehalt
und zwar aus Abteilungen K51 oder K55



⇒ **IV.** ***Fasse einfache Selektionen auf einer Relation zusammen***

Algebraische Optimierung – Beispiel (6)

6. Kartesisches Produkt mit Selektion



Leistungsbetrachtung bei:

a) verlustfreier natürlicher Verbund

b) Θ-Verbund (Annahme: 100 Ergebnistupel)

⇒ **V.** ***Verknüpfe bestimmte Selektionen mit einem vorausgehenden Kartesischen Produkt zu einem Verbund***

⇒ **VI.** ***Berechne gemeinsame Teilbäume nur einmal***

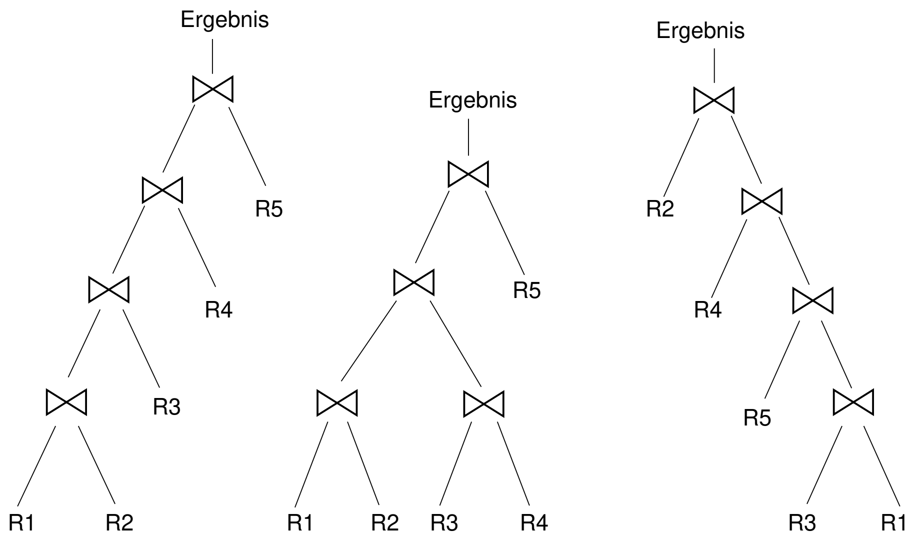
(wenn die Zwischenspeicherung der Ergebnisse nicht zu teuer ist)

Algebraische Optimierung – Beispiel (7)

- **Assoziativität und Kommutativität** von Vereinigung, Durchschnitt, Verbund

  1. T = R3 ⋈ (R1 ⋈ R2)
  2. T = R2 ⋈ (R1 ⋈ R3)
  3. T = R1 ⋈ (R2 ⋈ R3)

- **Allgemeines Problem bei binären Relationenoperationen**
  - Was ist die beste Verknüpfungsreihenfolge?
  - Im allgemeinen Fall sind n! Reihenfolgen möglich
  - Die genaue Größe einer Zwischenrelation ergibt sich erst nach Ende der erzeugenden Operation
  - Einige Verknüpfungsreihenfolgen für den Verbund mit n=5



Algebraische Optimierung – Beispiel (8)

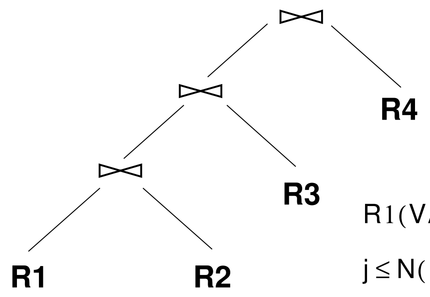
7. Kombination von Verbundoperationen

- **Abschätzung**

  beim (1:n)-Verbund: N(R1) = j

  ⇒ N(T1) = N(R2)

- **Bestimmung der Verbundreihenfolge (Heuristik):**



$R1(VA) = R2(VA) = R3(VA) = R4(VA)$

$j \leq N(R1) < N(R2) < N(R3) < N(R4)$

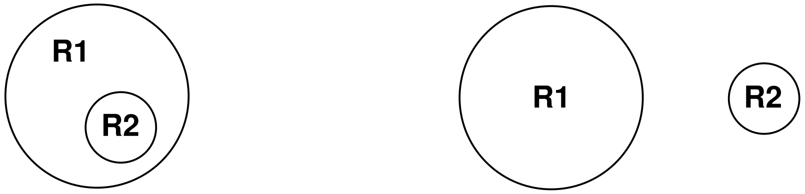
⇒ **VII.** ***Bestimme die Verbundreihenfolge so, daß die Anzahl und Größe der Zwischenobjekte minimiert wird***

- **Dynamische Entscheidung aufwendiger, aber genauer**
  Bei jedem Auswertungsschritt werden die momentan kleinsten (Zwischen-) Relationen ausgewählt

Algebraische Optimierung – Beispiel (9)

8. Reihenfolgen von Mengenoperationen

- **Kardinalität der Vereinigung**

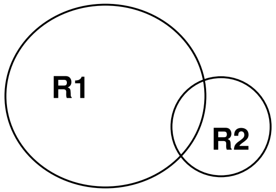


MAX(N(R1) , N(R2)) $\leq$ N(R1 $\cup$ R2) $\leq$ N(R1) + N(R2)

- **Kardinalität des Durchschnitts**

0 $\leq$ N(R1 $\cap$ R2) $\leq$ MIN(N(R1) , N(R2))

Erwartung:



- **Heuristische Regel:**

$\Longrightarrow$ **VIII.** ***Verknüpfe bei Mengenoperationen immer zuerst die kleinsten Relationen***

Zusammenfassung: Algebraische Optimierung

Heuristische Regeln:

1. Führe Selektion so früh wie möglich aus
2. Führe Projektion (ohne Duplikateliminierung) frühzeitig aus
3. Verknüpfe Folgen von unären Operationen wie Selektion und Projektion
4. Fasse einfache Selektionen auf einer Relation zusammen
5. Verknüpfe bestimmte Selektionen mit einem vorausgehenden Kartesischen Produkt zu einem Verbund
6. Berechne gemeinsame Teilbäume nur einmal
7. Bestimme Verbundreihenfolge so, daß die Anzahl und Größe der Zwischenobjekte minimiert wird
8. Verknüpfe bei Mengenoperationen immer zuerst die kleinsten Relationen

Division

- **Ziel:**
  - Beantwortung von Fragen, bei denen eine „ganze Relation“ zur Qualifikation herangezogen wird
  - Simulation des Allquantors $\Rightarrow$ ein Tupel aus R steht mit <u>allen</u> Tupeln aus S in einer bestimmten Beziehung
    - Def.:
      Sei R vom Grad r und S vom Grad s, $r > s$ und $s \neq 0$.
      t sei (r-s)-Tupel, u sei s-Tupel.
      S-Attribute $\subset$ R-Attribute

      Dann gilt:

      $$R \div S = \{ t \mid \forall u \in S : (t \mid u \in R) \}$$

- **Beschreibung der Division mit den Grundoperationen**

$$T = \pi_{1, 2, \ldots, r-s}(R)$$

$$W = (T \times S) - R$$

$$V = \pi_{1, 2, \ldots, r-s}(W)$$

$$R \div S = T - V$$

$$= \pi_{1, 2, \ldots, r-s}(R) - \pi_{1, 2, \ldots, r-s}((\pi_{1, 2, \ldots, r-s}(R) \times S) - R)$$

  - Es gilt: $(R \times S) \div S = R$

Division – Beispiel

- **Datenbank**

DA	PNR	FIGUR	A-Jahr ...
	P1	Faust	1999
	P1	Nathan	1998
	P2	Werther	1997
	P3	Faust	1998
	P3	Nathan	1999
	P3	Werther	1998

RO	FIGUR	TITEL	R-Typ
	Faust	Faust	
	Nathan	Nathan der Weise ...	
	Werther	Die Leiden ...	

- **Frage: Welche Schauspieler haben alle Rollen gespielt?**

$$(\pi_{PNR,\ FIGUR}(DA)) \div (\pi_{FIGUR}(RO))$$

- **Auswertung:**

$\pi_{PNR,\ FIGUR}(DA)$	PNR	FIGUR
	P1	Faust
	P1	Nathan

$\pi_{PNR,\ FIGUR}(DA)$	PNR	FIGUR
	P2	Werther

$\pi_{PNR,\ FIGUR}(DA)$	PNR	FIGUR
	P3	Faust
	P3	Nathan
	P3	Werther

$? \supseteq$

$\pi_{FIGUR}(RO)$	FIGUR
	Faust
	Nathan
	Werther

Ergebnis	PNR
	P3

Anfragen (3)

Q10: Finde alle Schauspieler (NAME), die **alle** Rollen in Dramen von Goethe gespielt haben.

Q11: Finde alle Schauspieler (NAME), die **alle** Narrenrollen am Pfalztheater gespielt haben.

Intervallverbund (*Band Join*)

- Anstatt des arithmetischen Vergleichsoperators Θ des Θ-Joins wird hier eine Intervall-Bedingung überprüft.
- Grob:

  Kartesisches Produkt zwischen zwei Relationen R (Grad r) und S (Grad s) **eingeschränkt durch eine Intervall-Bedingung** zwischen i-Spalte von R und j-Spalte von S.
- Intervall $I = [c_1, c_2]$ mit c_1, c_2 sind positive Konstanten, wobei eine größer Null sein muß.

  Intervall-Verbund zwischen R und S:

$$V = R \underset{iIj}{\bowtie} S = \sigma_{iIj}(R \times S)$$

$$= \sigma_{R.i - c_1 \leq S.j \leq R.i + c_2}(R \times S)$$

- Bemerkung:

  Ein Tupel s aus S ‘kombiniert’ mit einem Tupel r aus R nur, wenn der Wert der j-Spalte von S im Intervall der Größe $c_1 + c_2$ um den Wert der i-Spalte von R liegt.
- Beispiel:

  G = GLEICHALTRIGE = $\sigma_{PNR \neq PNR'}$ $\text{PERS} \underset{(\text{ALTER}[2,2]\text{ALTER'})}{\bowtie} \text{PERS'}$

PERS	PNR	ALTER
	P1	25
	P2	23
	P3	28

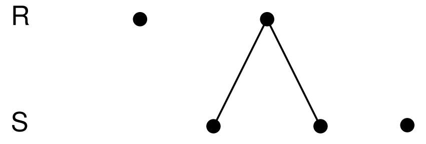
PERS’	PNR’	ALTER’ ...
	P1	25
	P2	23
	P3	28

G	(PNR	ALTER	PNR’	ALTER’)

Äußerer Verbund (*Outer Join*)

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

  Bsp.: 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, . . .)

PNR	NAME
P1	x
P2	y

DA (PNR, FIGUR, . . .)

PNR	FIGUR
P1	F
P1	W
P3	M

$$SP' = SP \cup ((\Pi_{PNR}\, DA - \Pi_{PNR}\, SP) \times \equiv \times \equiv \ldots)$$

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

$$DA' = DA \cup ((\Pi_{PNR}\, SP - \Pi_{PNR}\, DA) \times \equiv \times \equiv \ldots)$$

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

Äußerer Verbund (*2*)

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

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

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

Äußerer Gleichverbund

$$R \mathbin{⟗}_{R.A = S.A} S := R' \bowtie_{R'.A = S'.A} S'$$

Äußerer natürlicher Verbund

$$R \mathbin{⟗} S := R' \bowtie S'$$

- 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

$$R \mathbin{⟕}_{R.A = S.A} S := R \bowtie_{R.A = S'.A} S'$$

- Rechter äußerer Gleichverbund

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

Rechter äußerer Gleichverbund

$$R \mathbin{⟖}_{R.A = S.A} S := R' \bowtie_{R'.A = S.A} S$$

Beispiele zum äußeren Gleichverbund

- **Gleichverbund**

R	A	B	C
	a_1	b_1	c_1
	a_2	b_2	c_2

$\bowtie$

S	C	D	E
	c_1	d_1	e_1
	c_3	d_2	e_2

=

ERG	A	B	C	D	E
	a_1	b_1	c_1	d_1	e_1

- **Linker äußerer Gleichverbund**

R	A	B	C
	a_1	b_1	c_1
	a_2	b_2	c_2

⟕

S	C	D	E
	c_1	d_1	e_1
	c_3	d_2	e_2

=

ERG	A	B	C	D	E
	a_1	b_1	c_1	d_1	e_1
	a_2	b_2	c_2	--	--

- **Rechter äußerer Gleichverbund**

R	A	B	C
	a_1	b_1	c_1
	a_2	b_2	c_2

⟖

S	C	D	E
	c_1	d_1	e_1
	c_3	d_2	e_2

=

ERG	A	B	C	D	E
	a_1	b_1	c_1	d_1	e_1
	--	--	c_3	d_2	e_2

- **Äußerer Gleichverbund**

R	A	B	C
	a_1	b_1	c_1
	a_2	b_2	c_2

⟗

S	C	D	E
	c_1	d_1	e_1
	c_3	d_2	e_2

=

ERG	A	B	C	D	E
	a_1	b_1	c_1	d_1	e_1
	a_2	b_2	c_2	--	--
	--	--	c_3	d_2	e_2

Weitere äußere Operationen

- **Äußere Vereinigung (*OUTER UNION*)**

Diese Operation erlaubt die Vereinigung zweier Relationen, die nicht vereinigungsverträglich sind. Wenn zwei Relationen partiell verträglich sind, d.h., einige ihrer Attribute sind vereinigungsverträglich, dann kann OUTER UNION angewendet werden.

Beispiel:

STUDENT	MATNR	FBNR	SEM
	123	FB5	5
	789	FB9	9

HIWI	MATNR	FBNR	JOB
	456	FB5	Tutor
	987	FB9	Prog.

OUTER UNION

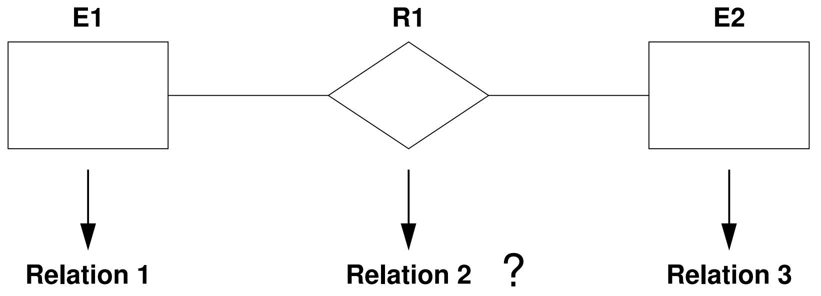
STUD-HIWI	MATNR	FBNR	SEM	JOB
	123	FB5	5	-
	789	FB9	9	-
	456	FB5	-	Tutor
	987	FB9	-	Prog.

→ Es können große Interpretationsprobleme beim Ergebnis entstehen

- In ähnlicher Weise lassen sich weitere Operationen einführen:
  - OUTER INTERSECTION
  - OUTER DIFFERENCE

→ Diese Operationen scheinen nicht besonders nützlich zu sein

Abbildung von Entity- und Relationship-Mengen

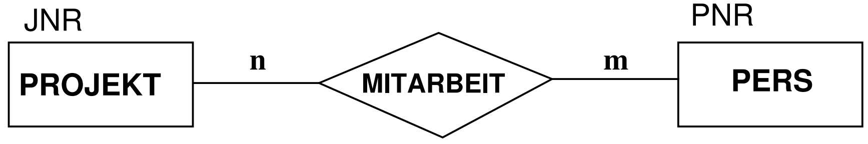


- **Kriterien**
  - Informationserhaltung
    - Abbildung der ER-Konzepte auf RM-Konzepte
    - möglichst genaue Übereinstimmung der Semantik (Übernahme aller spezifizierten Eigenschaften)

  ➥ 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 (<u>JNR</u>, BEZEICH, ...)

PERS (<u>PNR</u>, PNAME, ...)

MITARBEIT (<u>JNR, PNR</u>)

- **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. Die Attributnamen in einer Relation müssen eindeutig sein.

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

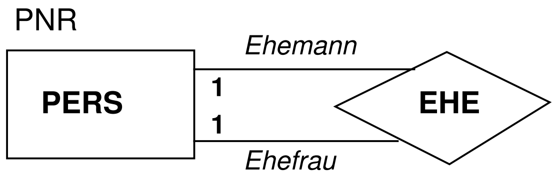
Verwendung von

drei Relationen	zwei Relationen
ABT (ANR, ANAME, ...)	**ABT** (ANR, ANAME, ...)
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, ...)

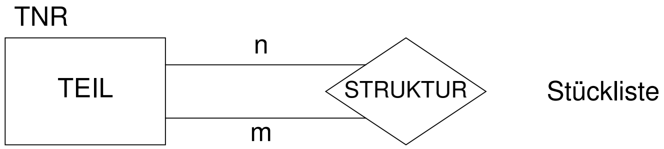
2.) Verwendung von einer Relation

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

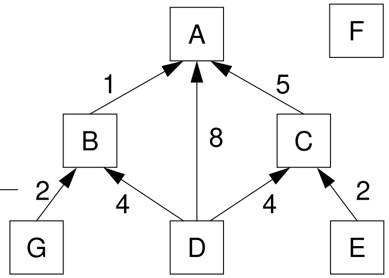


Darstellungsmöglichkeit im RM:

TEIL (TNR, BEZ, MAT, BESTAND)

STRUKTUR (OTNR, UTNR, ANZAHL)

TEIL	TNR	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

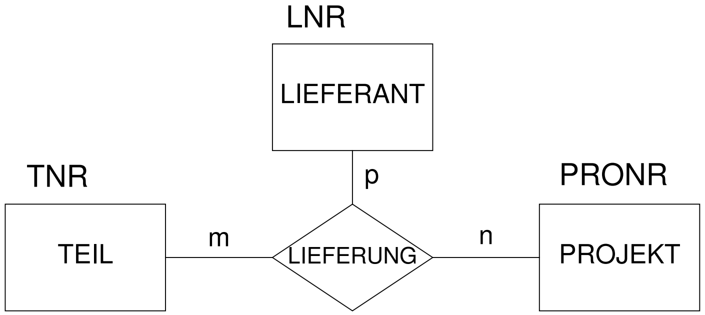


STRUKTUR	OTNR	UTNR	ANZAHL
	A	B	1
	A	C	5
	A	D	8
	B	D	4
	B	G	2
	C	D	4
	C	E	2

- **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:

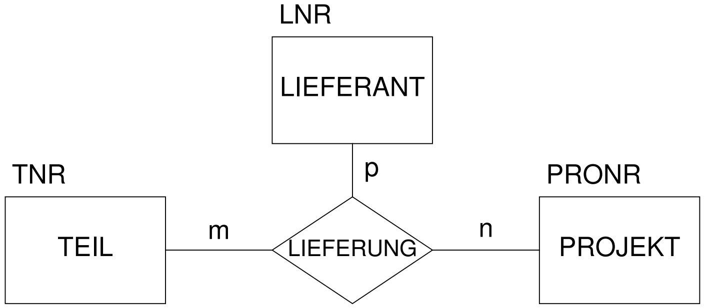
LIEFERANT (LNR, LNAME, LORT, ...)

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

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

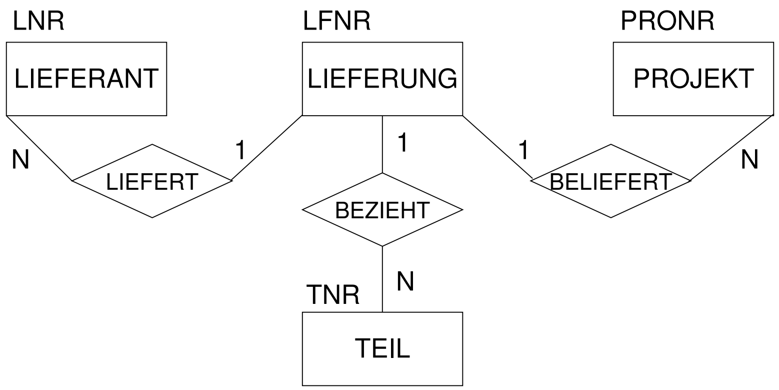
LIEFERUNG (LNR, PRONR, TNR, ANZAHL, DATUM)

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 (<u>PNR</u>, NAME, ADRESSE, ..., GEHALT, VWL)

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

- **Horizontale Partitionierung**
  - Klassenbildung anhand von Selektionsbedingungen

  PERS-VIP (<u>PNR</u>, ..., VWL) GEHALT > 100K

  PERS (<u>PNR</u>, ..., VWL) GEHALT ≤ 100K

  → ist eigentlich Aufgabe eines Sichtkonzeptes

- **Vertikale Partitionierung**
  - zur leichteren Einhaltung von Leistungs- und Schutzaspekten:

  PERS-ÖFF (<u>PNR</u>, PNAME, ADRESSE, ...)

  PERS-PRIV (<u>PNR</u>, 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 ...)

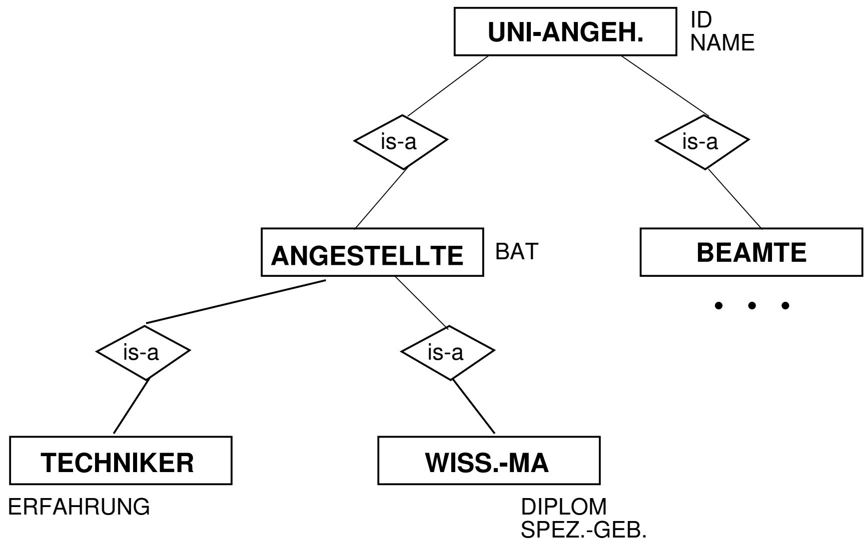
    L.ESSEN (PNR, GERICHT,)

    KINDER (PNR, VORNAME, ALTER)

Abbildung der Generalisierung im RM

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

Generalisierungsbeispiel:



- **LÖSUNG 1:** Hausklassen-Modell:
  - Jede Instanz ist genau einmal und vollständig in ihrer Hausklasse gespeichert
  - Es wird eine horizontale Partitionierung der DB-Instanzen erreicht
  - Einsatz in ORION und O_2

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	IIa
	765	Mathematik	ERM	Grouch	IIa

- **Eigenschaften:**
  - niedrige Speicherungskosten und keine Änderungsanomalien
  - Retrieval kann rekursives Suchen in Unterklassen erfordern
  - explizite Rekonstruktion durch Relationenoperationen (π, $\cup$)

    → 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
  - IRIS wendet dieses Speicherungsmodell an

UNI-ANGEH.	ID	NAME
	007	Garfield
	111	Ernie
	123	Donald
	333	Daisy
	765	Grouch

ANGESTELLTE	ID	BAT
	007	Ia
	123	IVa
	333	IIa
	765	IIa

TECHNIKER	ID	ERFAHRUNG
	123	SUN

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

- **Eigenschaften**
  - geringfügig erhöhte Speicherungskosten, 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 $\underset{ID=ID}{\bowtie}$ ANGESTELLTE $\underset{ID=ID}{\bowtie}$ 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
	007	Garfield
	111	Ernie
	123	Donald
	333	Daisy
	765	Grouch

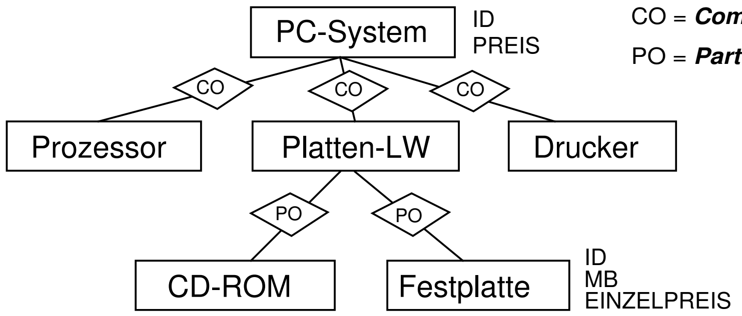
ANGESTELLTE	ID	NAME	BAT
	007	Garfield	Ia
	123	Donald	IVa
	333	Daisy	IIa
	765	Grouch	IIa

TECHNIKER	ID	NAME	BAT	ERFAHRUNG
	123	Donald	IVa	SUN

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



CO = ***Component-Of***

PO = ***Part-Of***

PC-System	PREIS	PROZESSOR	PLATTENK.	DRUCKER
PC-1	800,-	I-Celeron 1000 MHz	PK1	Tintenstrahldr.
PC-2	1400,-	I-Pentium IV 1,7 GHz	PK2	Laserdrucker
PC-3	2000,-	I-Pentium IV 2,2 GHz	PK2	Laserdrucker

Plattenk.	ID	CD-ROM	Festplatte
	PK1		F1
	PK2	C1	F2
	PK3	C2	F3

CD-ROM	ID	Geschwindigk.	Preis
	C1	48-fach Speed	35.-
	C2	DVD 16x/48x	79.-

Festplatte	ID	Kapaz.	Preis
	F1	20 GB	89.-
	F2	40 GB	119.-
	F3	80 GB	179.-

Prozessor	ID	SPEC int 2000	Preis
	I-Celeron 1000 MHz	407	105.-
	I-Pentium IV 1,7 GHz	573	249.-
	I-Pentium IV 2,2 GHz	771	649.-

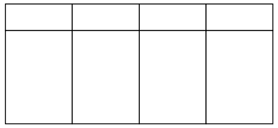
Drucker	ID	Hersteller	Typ	Preis
	Tintenstrahldr.	HP	Desk Jet 940C	149.-
	Laserdrucker	HP	Laser 1200	449.-

➥ Eigenschaften der Aggregation werden durch relationale Operationen nicht unterstützt

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

Zusammenfassung: Relationenalgebra

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