4. Grundlagen des Relationenmodells

Inhalt

Grundkonzepte

Abbildung von ER-Diagrammen

Relationenalgebra

Algebraische Optimierung

Übersicht (1)

- **Datenstruktur**
  - Relation (Tabelle)
    - einzige Datenstruktur
    - alle Informationen ausschließlich durch (atomare) Werte dargestellt
    - zeitinvariante Typinformation: Relationenschema
- **Operatoren auf (mehreren) Relationen**
  - Vereinigung, Differenz (Schemata müssen kompatibel sein)
  - Kartesisches Produkt
  - Projektion
  - Selektion
  - zusätzlich: Grundoperationen (Einfügen, Löschen, Ändern)

Übersicht (2)

- **Beziehungen**
  - werden durch Werte dargestellt
    - Primär-/Fremdschlüssel
    - Gewährleistung von referentieller Integrität
    - können in SQL automatisch gewartet werden (referentielle Aktionen)
  - Fremdschlüssel verbindet zwei Relationen $\Rightarrow$ Beziehung binär
  - Komplexere Beziehungen durch Kombination mehrerer Fremdschlüssel

- **Entwurfstheorie**
  - Normalformenlehre (wünschenswerte und zweckmäßige Relationen)
  - Synthese von Relationen

Grundkonzepte (1)

ERM	RM
Einwertiges Attribut Definitionsbereich Primärschlüssel	wie im ERM
Zusammengesetztes Attribut Mehrwertiges Attribut	nur als unabhängige, einwertige Attribute
Entity-Typ/Menge Relationship-Typ/Menge	**Relation**

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

Grundkonzepte (2)

<u>Mathematische Notation</u>:

$D_1, D_2, \ldots, D_n$	**Definitionsbereiche (Domänen)**
$R \subseteq D_1 \times D_2 \times \ldots \times D_n$	**Relation (Beziehung)**
$t \in R$	**Tupel / Record**

<u>Notation für Datenbank-Relationen</u>:

$A_1, A_2, \ldots, A_n$	**Attribute**
$D_1, D_2, \ldots, D_n$	**(primitive) Datentypen**
$R \in Rel(A_1{:}D_1, \ldots, A_n{:}D_n)$	**Relation über den Attributen $A_1, \ldots, A_n$ mit den Domänen $D_1, \ldots, D_n$**

- **Relation** kann als **Tabelle** dargestellt werden
- **Relation** ist eine **Menge**
  - Garantie der Eindeutigkeit der Zeilen/Tupel durch **Primärschlüssel** (und ggf. mehrere Schlüsselkandidaten)

Grundkonzepte (3)

- **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

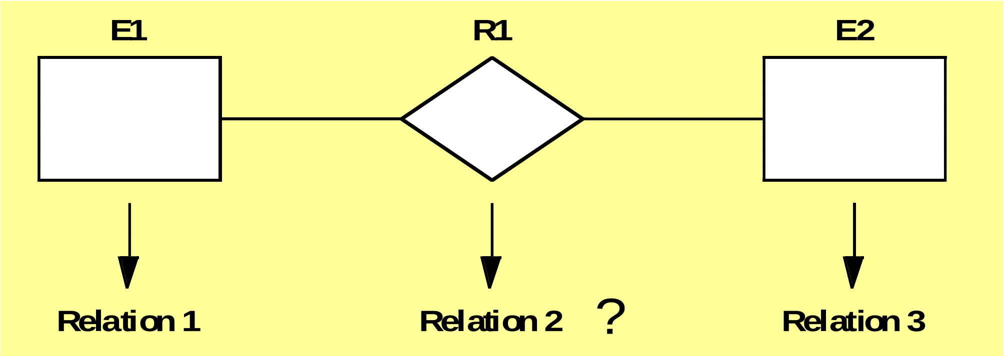
Grundkonzepte (4)

- **Wie wird „relationenübergreifende" Information dargestellt?**
  - **Fremdschlüssel,** Definition:
    - Ein Fremdschlüssel bzgl. einer Relation R1 ist ein Attribut oder eine Attributkombination **FS** einer Relation R2, für das/die 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 zu Fremdschlüssel:
    - Fremdschlüssel und zugehöriger Primärschlüssel (Schlüsselkandidat) tragen wichtige inter-relationale (manchmal auch intra-relationale) Informationen. Sie sind auf dem gleichen Wertebereich definiert (vergleichbar und vereinigungsverträglich). Sie gestatten die Verknüpfung von Relationen mit Hilfe von Relationenoperationen.
    - Fremdschlüssel können Nullwerte aufweisen, wenn sie nicht Teil eines Primärschlüssels sind und wenn nicht explizit NOT NULL spezifiziert ist.

Grundkonzepte (5)

- **Wie wird „relationenübergreifende“ Information dargestellt?** (Forts.)
  - Bemerkungen zu Fremdschlüssel (Forts.)
    - Schlüsselkandidaten können Nullwerte aufweisen, wenn nicht explizit NOT NULL spezifiziert ist.
    - Ein Fremdschlüssel ist zusammengesetzt, wenn der zugehörige Primärschlüssel (Schlüsselkandidat) zusammengesetzt ist.
    - Eine Relation kann mehrere Fremdschlüssel besitzen, welche die gleiche oder verschiedene Relationen referenzieren.
    - Referenzierte und referenzierende Relation sind nicht notwendigerweise verschieden.
    - Zyklen sind möglich.

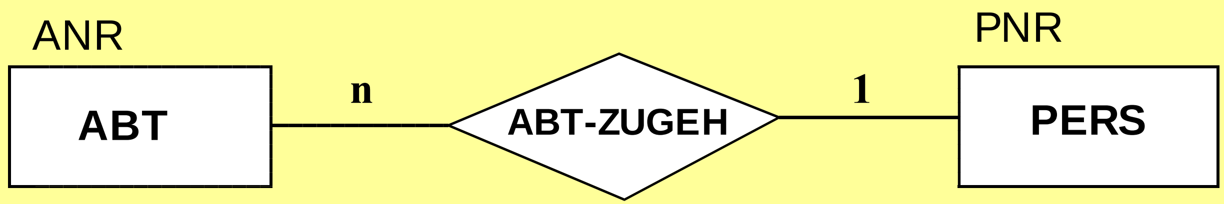
Abbildung ERM-RM (1)



- Kriterien
  - Informationserhaltung, d.h. möglichst genaue Übereinstimmung der Semantik (Übernahme aller spezifizierten Eigenschaften)
  - Minimierung der Redundanz
  - Minimierung des Verknüpfungsaufwandes
  - Natürlichkeit der Abbildung
  - Keine Vermischung von Objekten
  - Verständlichkeit

Abbildung ERM-RM (2)

- **2 Entity-Typen mit (1:n)-Verknüpfung**
  - **Regel:**
    - (1:n)-Beziehungen lassen sich ohne eigene Relation darstellen. Hierzu wird in der abhängigen Relation (mit Beziehungskardinalität 1) der Primärschlüssel der referenzierten Relation als Fremdschlüssel verwendet.

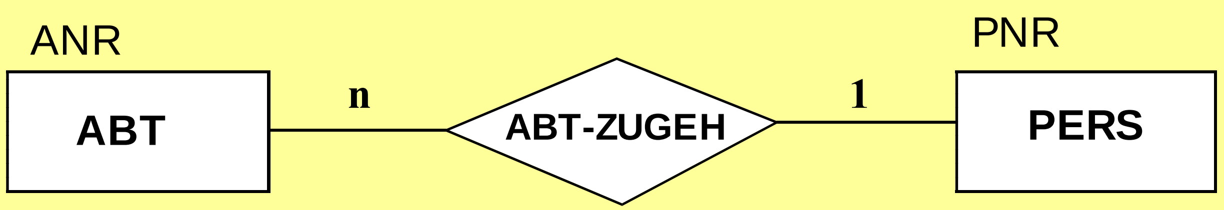


Verwendung von

drei Relationen	zwei Relationen
ABT (ANR, ANAME, ...)	**ABT** (ANR, ANAME, ...)
PERS (PNR, PNAME, ...)	**PERS** (PNR, PNAME, ..., ANR)
ABT-ZUGEH (ANR, PNR)	

Abbildung ERM-RM (2)

- **Notation Fremdschlüssel**
  - **Für Übungen:**
    - Notation bei einfachen Primär-/Fremdschlüsseln:

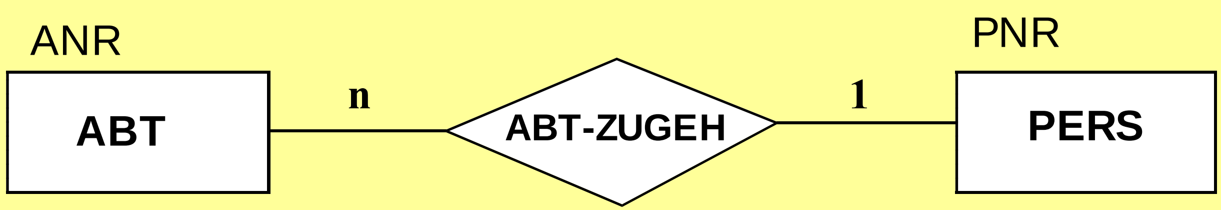


ABT (ANR, ...)

PERS (PNR, ... ANR → ABT.ANR, ...)

Abbildung ERM-RM (2)

- **Notation Fremdschlüssel**
  - **Für Übungen:**
    - Notation bei zusammengesetzten Primär-/Fremdschlüsseln:



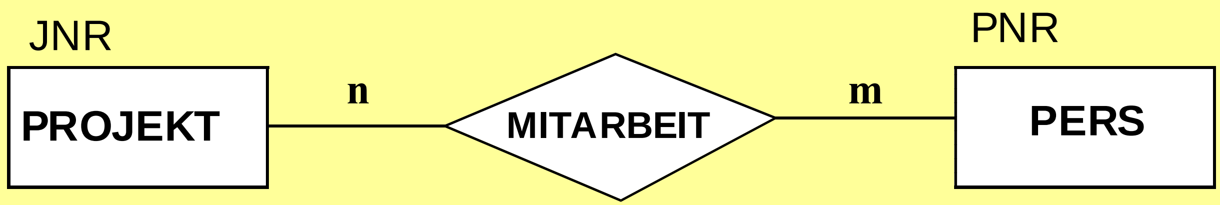
ABT (AName, AOrt, …)

PERS (VName, NName, …

(AName, AOrt) → (ABT.AName, ABT.AOrt))

Abbildung ERM-RM (3)

- **2 Entity-Typen mit (n:m)-Verknüpfung**
  - **Regel:**
    - Ein (**n:m**)-Relationship-Typ **muss** durch eine eigene Relation dargestellt werden. Der Primärschlüssel dieser Relation setzt sich aus den Primärschlüsseln der beteiligten Entity-Typen zusammen. Alle Namen können übernommen werden; es ist jedoch auch eine Umbenennung möglich. Die Attributnamen in einer Relation müssen eindeutig sein.



Verwendung von drei Relationen erforderlich:

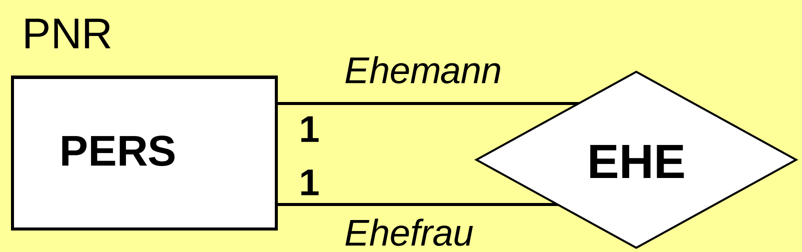
PROJEKT (JNR, BEZEICH, ...)

PERS (PNR, PNAME, ...)

MITARBEIT (JNR, PNR)

Abbildung ERM-RM (4)

- **1 Entity-Typ mit (1:1)-Verknüpfung**
  - **Regel:**
    - Der Primärschlüssel des zugehörigen Entity-Typs wird in zwei Rollen verwendet. Deshalb ist eine Umbenennung erforderlich.



1.) Verwendung von zwei Relationen

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

EHE (PNR, GATTE, ...)

- 2 Tupel pro Ehe
- Gatte ist ein Schlüsselkandidat
- Verletzung der Symmetrie möglich

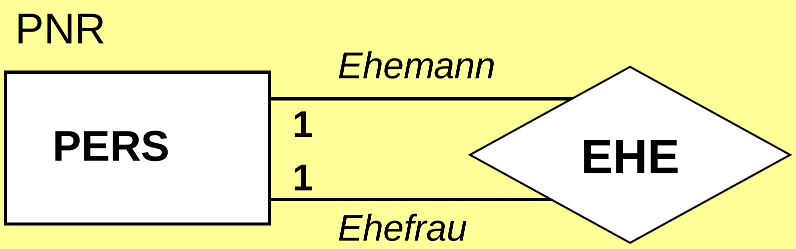
b) **PERS** (PNR, PNAME, ...)

EHE (Ehemann, Ehefrau, ...)

- 1 Tupel pro Ehe
- Ehefrau ist ein Schlüsselkandidat
- Symmetrie automatisch gewahrt

Abbildung ERM-RM (5)

- **1 Entity-Typ mit (1:1)-Verknüpfung**
  - **Regel:**
    - Der Primärschlüssel des zugehörigen Entity-Typs wird in zwei Rollen verwendet. Deshalb ist eine Umbenennung erforderlich.



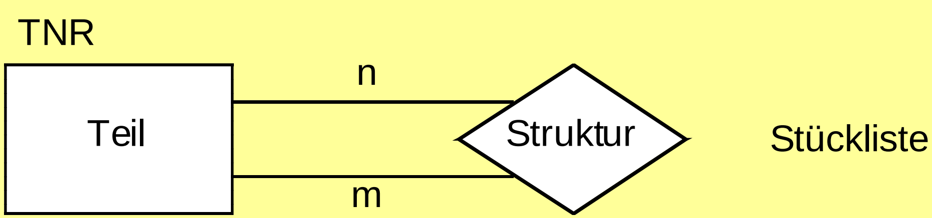
2.) Verwendung von einer Relation

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

- Ehe aus zwei versch. Perspektiven modelliert (d.h. redudante Speicherung)
- Gatte ist ein Schlüsselkandidat (keine 2 Personen haben denselben Gatten)
- Verletzung der Symmetrie möglich

Abbildung ERM-RM (6)

- **1 Entity-Typ mit (n:m)-Verknüpfung**
  - **Regel:**
    - Ein (**n:m**)-Relationship-Typ **muss** durch eine eigene Relation dargestellt werden. Der Primärschlüssel des zugehörigen Entity-Typs wird in zwei Rollen verwendet. Deshalb ist eine Umbenennung erforderlich.



Darstellungsmöglichkeit:

TEIL (TNR, BEZ, MAT, BESTAND)

STRUKTUR (OTNR, UTNR, ANZAHL)

Abbildung ERM-RM (7)

- **1 Entity-Typ mit (n:m)-Verknüpfung (Forts.)**

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

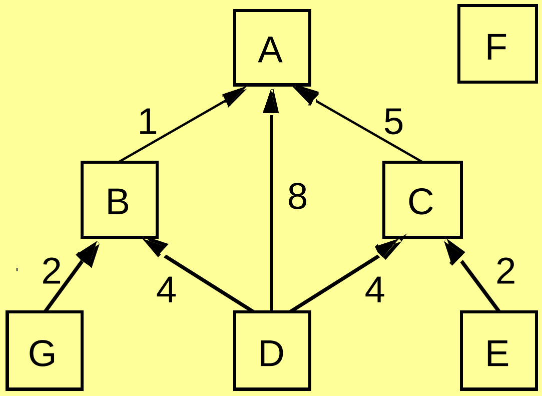
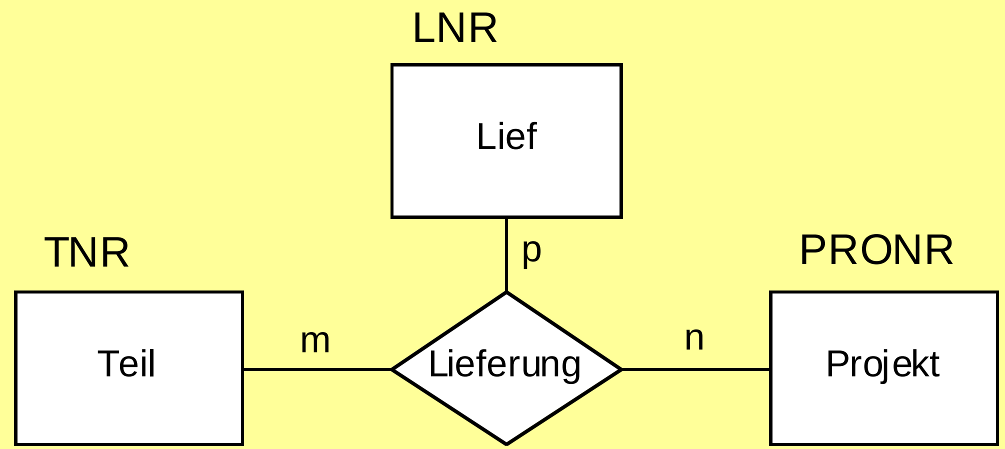


Abbildung ERM-RM (8)

- **mehrere (>2) Entity-Typen mit (n:m)-Verknüpfung**



Darstellungsmöglichkeit im RM:

LIEF (LNR, LNAME, LORT, ...)

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

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

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

Abbildung ERM-RM (9)

- **mehrere (>2) Entity-Typen mit (n:m)-Verknüpfung (Forts.)**

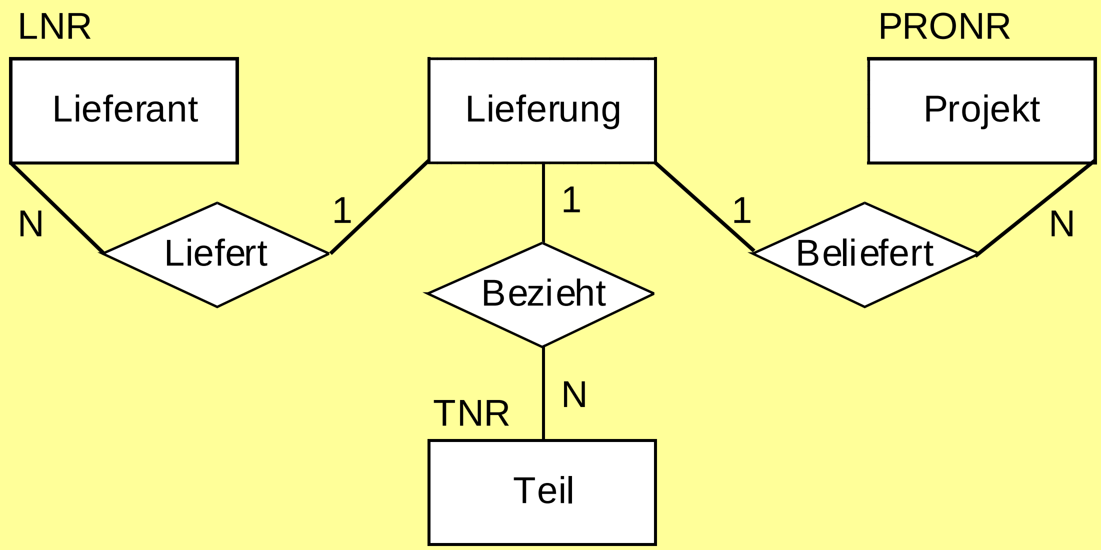
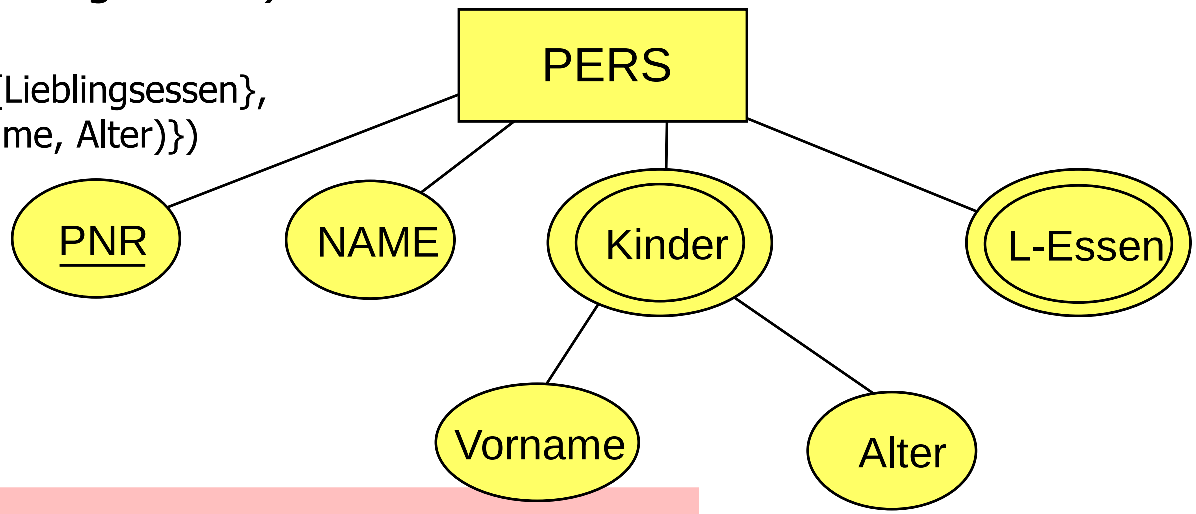


Abbildung ?

Abbildung ERM-RM (10)

- Mehrwertige (zusammengesetzte) Attribute

PERS (PNR, NAME, {Lieblingsessen}, {Kinder (Vorname, Alter)})



Achtung: Einwertige zusammengesetzte Attribute benötigen keine extra Relation

Darstellungsmöglichkeit:

PERS	(PNR, NAME ...)
L-ESSEN	(PNR, GERICHT)
KINDER	(PNR, VORNAME, ALTER)

Vorausgesetzt: Jedes Kind hat einen anderen Vornamen (sonst Alter mit in PK)

Abbildung ERM-RM (11)

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

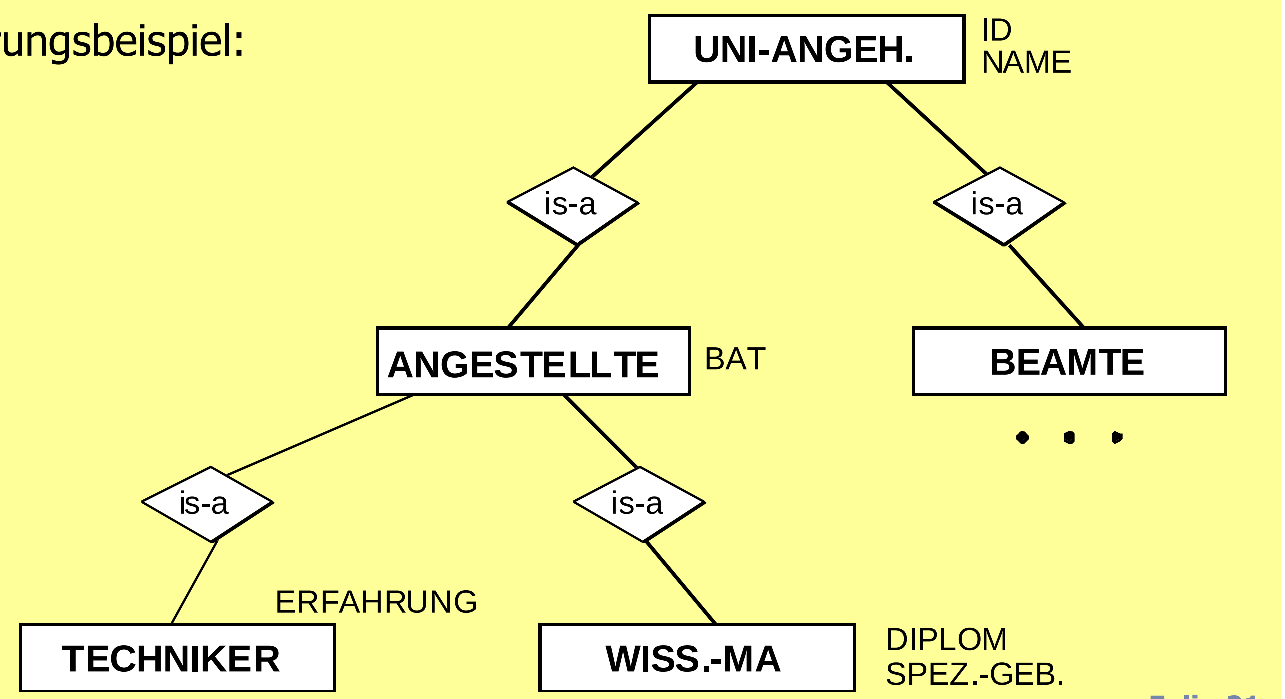


Abbildung ERM-RM (12)

- Generalisierung: Hausklassenmodell
  - 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	SPEZ.-GEB.	NAME	BAT
	333	Informatik	RECOVERY	Daisy	IIa
	765	Mathematik	ERM	Grouch	IIa

Abbildung ERM-RM (13)

- Generalisierung: Hausklassenmodell (Forts.)
  - **Eigenschaften:**
    - niedrige Speicherungskosten und keine Änderungsanomalien
    - Eindeutigkeit von IDs muss über Relationen hinweg gewährleistet werden
    - Retrieval kann rekursives Suchen in Unterklassen erfordern
    - explizite Rekonstruktion durch Relationenoperationen (π, $\cup$)
      - Beispiel: Finde alle ANGESTELLTE:

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

Projektion: unärer Operator der Relationenalgebra zur Auswahl von Spalten/Attributen; siehe unten.

Detaillierte Behandlung der relationalen Operatoren weiter hinten in diesem Kapitel!

Abbildung ERM-RM (14)

- Generalisierung: 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 (evtl. Verwendung als Fremdschlüssel)
  - Es wird eine vertikale Partitionierung in der DB erzielt

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	ERM
	765	Mathematik	MAD

Abbildung ERM-RM (15)

- Generalisierung: Partitionierungs-Modell (Forts.)
  - **Eigenschaften**
    - geringfügig erhöhte Speicherungskosten, aber hohe Aufsuch- und Aktualisierungskosten
    - Integritätsbedingungen: TECHNIKER.ID $\subseteq$ ANGESTELLTE.ID, usw. (Gewährleistung durch Verwendung von Fremdschlüsseln)
    - Instanzenzugriff erfordert implizite oder explizite Verbundoperationen ($\bowtie$)
      - Beispiel: Finde alle TECHNIKER-Daten

$$\text{TECHNIKER} \underset{\textbf{ID}}{\bowtie} \text{ANGESTELLTE} \underset{\textbf{ID}}{\bowtie} \text{UNI-ANGEH.}$$

Join/Verbund: binärer Operator der Relationenalgebra zur Verbindung der Tupeln der Argumentrelationen über gleiche Attributwerte; siehe unten.

Detaillierte Behandlung der relationalen Operatoren weiter hinten in diesem Kapitel!

Abbildung ERM-RM (16)

- Generalisierung: 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

Abbildung ERM-RM (17)

- Generalisierung: Volle Redundanz (Forts.)
  - **Eigenschaften**
    - sehr hoher Speicherplatzbedarf
    - Auftreten von Änderungsanomalien

    Zur Vermeidung: Aufwendige (evtl. synchrone) Änderungen von redundant gespeicherten Daten notwendig

    - sehr einfaches Retrieval, da nur die Zielklasse (z. B. ANGESTELLTE) aufgesucht werden muss

Abbildung ERM-RM (18)

- Generalisierung: Modellierung in einer einzigen Relation
  - Die Attribute aller Entity-Typen werden in einer Relation zusammengefasst
  - Spezielle Attribute um Typ-Zugehörigkeit zu modellieren

UNI-ANGEH.	ID	NAME	ANG?	BAT	TECH?	ERFAHRUNG	WMA?	DIPLOM	SPEZ.-GEB.
	007	Garfield	Ja	Ia	Nein	--	Nein	--	--
	111	Ernie	Nein	--	Nein	--	Nein	--	--
	123	Donald	Ja	IVa	Ja	SUN	Nein	--	--
	333	Daisy	Ja	IIa	Nein	--	Ja	Informatik	RECOVERY
	765	Grouch	Ja	IIa	Nein	--	Ja	Mathematik	ERM

Abbildung ERM-RM (19)

- Generalisierung: Modellierung in einer einzigen Relation (Forts.)
  - **Eigenschaften**
    - Sehr viele Nullwerte (⇒ großer Speicherbedarf)
    - Keine Redundanz
    - Retrieval kann Scan großer Tupelmengen erfordern inkl. Selektion (σ) auf Zugehörigkeitsattribut
      - Beispiel: Finde alle TECHNIKER:

        $\sigma_{\text{TECH=„Ja“}}$ (UNI-ANGEH)
    - Keine intuitive Modellierung da verschiedene Entity-Typen in einer Relation

Selektion: unärer Operator der Relationenalgebra zur Auswahl von Zeilen/Tupeln; siehe unten.

Detaillierte Behandlung der relationalen Operatoren weiter hinten in diesem Kapitel!

Abbildung ERM-RM (20)

- Generalisierung mit nicht-disjunkten Untertypen:
  - **Hausklassenmodell nicht mehr eindeutig definiert**
  - **Variante 1:** Eine Relation pro Entity-Typ
    - Eine Instanz muss in mehreren Relationen gespeichert werden
  - **Variante 2:** Eindeutige Relationen-Zugehörigkeit für jede Instanz
    - Eine Relation pro Kombination von nicht-disjunkten Untertypen

Abbildung ERM-RM (21)

- Generalisierung mit nicht-disjunkten Untertypen:
  - **Beispiel:**

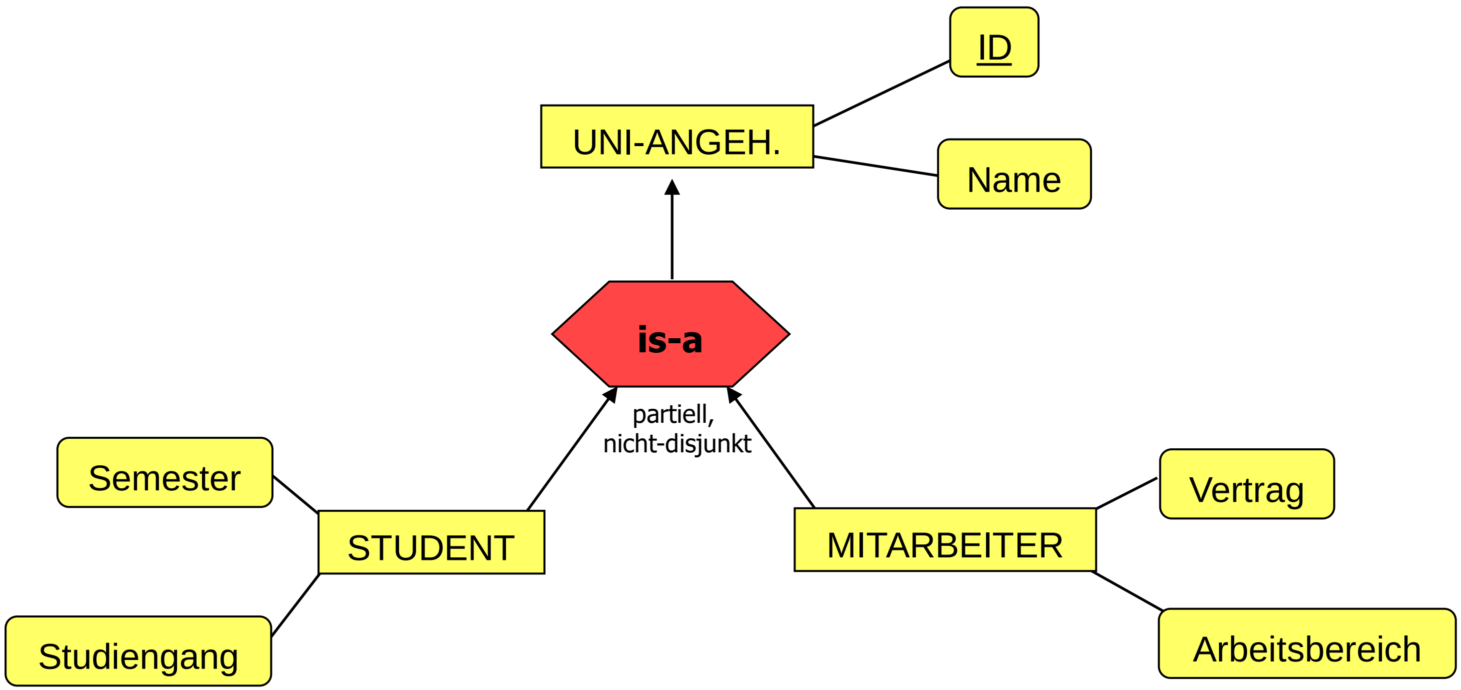


Abbildung ERM-RM (22)

- Generalisierung mit nicht-disjunkten Untertypen:
  - **Beispiel:** Eine Relation pro Entity-Typ

UNI-ANGEH.	ID	NAME
	3	Max

STUDENT	ID	NAME	STUDIENGANG	SEM
	1	Alice	Informatik, Prom.	3
	2	Bob	Informatik, M.Sc.	5

MITARBEITER	ID	NAME	VERTRAG	AB
	1	Alice	TV-L 13	ISYS

Abbildung ERM-RM (23)

- Generalisierung mit nicht-disjunkten Untertypen:
  - **Beispiel:** Eindeutige Relationen-Zugehörigkeit für jede Instanz

UNI-ANGEH.	ID	NAME
	3	Max

STUDENT	ID	NAME	STUDIENGANG	SEM
	2	Bob	Informatik, M.Sc.	5

MITARBEITER	ID	NAME	VERTRAG	AB

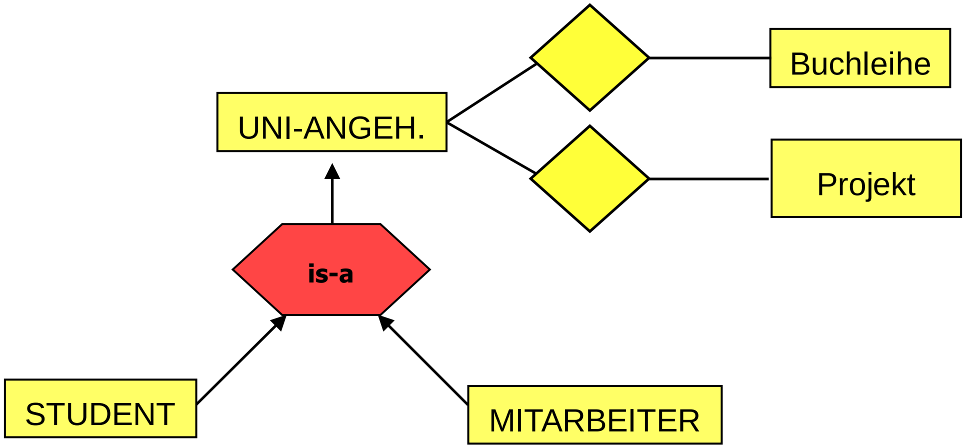
STUDENT & MITARBEITER	ID	NAME	STUDIENGANG	SEM	VERTRAG	AB
	1	Alice	Informatik, Prom.	3	TV-L 13	ISYS

Abbildung ERM-RM (24)

- Generalisierung mit vererbten n:m Relationship-Typen:
  - Modellierung durch Fremdschlüssel
  - Ein Fremdschlüssel zeigt auf eine konkrete Relation
- **Hausklassenmodell:**
  - Eine Relation für den Relationship-Typ pro Unterklasse
    (der Fremdschlüssel zeigt jeweils auf die zugehörige Unterklasse)
- **Partitionierungs-Modell oder volle Redundanz:**
  - Insgesamt eine Relation für den Relationship-Typen
    (der Fremdschlüsse zeigt auf die „oberste" Unterklasse)
- **Modellierung in einer einzigen Relation:**
  - Insgesamt eine Relation für den Relationship-Typen
  - Integritätsbedingungen, die gewährleisten, dass nur Tupel mit passenden Werten in den Zugehörigkeitsattributen im Fremdschlüssel referenziert werden können

Abbildung ERM-RM (25)

- Generalisierung mit vererbten n:m Relationship-Typen:
  - **Beispiel:**



- **Hausklassenmodell:**
  - Sechs Relationen (je 3 für jeden Relationship-Typen)
- **Partitionierungs-Modell oder volle Redundanz:**
  - Zwei Relationen (je 1 für jeden Relationship-Typen)

Abbildung ERM-RM (26)

- **Zusammenfassung der Abbildungskonzepte**
  - **Datenstruktur**: Relation (Tabelle)
    - einzige Datenstruktur (neben atomaren Werten)
    - alle Informationen ausschließlich durch Werte dargestellt
    - Integritätsbedingungen auf/zwischen Relationen
  - **Abbildung von Beziehungen durch PS/SK – FS**
    - Eine (n:1)-Verknüpfung wird in der Regel auf eine eigene Relation abgebildet
    - (n:m)-Verknüpfungen sind durch eine eigene Relation darzustellen
    - Verknüpfungen mit mehr als zwei Rollen müssen im Prinzip durch mehrere (n:1)-Verknüpfungen dargestellt werden
  - **Abstraktionskonzepte**
    - keine direkte Bereitstellung der Abstraktionskonzepte, z.B. Generalisierung
    - Verschiedene Modelle zur Simulation einer Generalisierung

Relationenalgebra - Operatoren (1)

- **Algebra: nicht leere Menge von Objekten + Familie von Operationen**
- **Operationen (unär, binär)**
  - **Klassische Mengenoperationen:**
    - Vereinigung, Differenz, kartesisches Produkt
    - ableitbar: Durchschnitt
  - **Relationenoperationen:**
    - Projektion, Restriktion (Selektion), Umbenennung
    - ableitbar: Verbund (Join), Division
- Auswahlvermögen entspricht Relationenkalkül („relational vollständig“)

Relationenalgebra - Operatoren (2)

- **Selektion (Restriktion): σ_P**
  - **Auswahl von Zeilen (Tupel) einer Relation über ein Prädikat**
  - P = log. Formel (ohne Quantoren!) bestehend aus Attributnamen, Konstanten, Vergleichsoperatoren ($<$, $=$, $>$, $\leq$, $\neq$, $\geq$) und logischen Verknüpfungen ($\vee$, $\wedge$, $\neg$)
  - $\sigma_P(R) = \{ t \mid t \in R \wedge P(t)\}$
  - Beispiel:

$$\sigma_{ANR='K55' \wedge GEHALT > 50\,000}\ (PERS)$$

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

Relationenalgebra - Operatoren (3)

- **Projektion: π**
  - **Auswahl von Spalten (Attribute) $A_1, A_2, \ldots, A_k$ aus einer Relation R (Grad n >= k)**
  - $\pi_{A_1, A_2, \ldots, A_k}(R) = \{ p \mid \exists t \in R : p = < t[A_1], \ldots, t[A_k] > \}$
    (Alternative: Benutzung von Spaltennummern)
  - Duplikateliminierung
  - Beispiel:

$\pi_{\text{ANR, MNR}}$ **(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~~

Relationenalgebra - Operatoren (4)

- **Umbenennung: ρ**
  - **Umbenennung von Relationen:** hilfreich, wenn eine Relation mehrfach (in unterschiedlichen Rollen) verwendet werden muss
  - **Umbenennung von Attributen:** hilfreich, wenn Attribute von zu verbindenden Relationen gleich heißen oder eine Schemakompatibilität zwischen verschiedenen Relationen hergestellt werden soll (z.B. bei Mengenoperationen)
  - Beispiel:

$$\rho_{\mathrm{ABTEILUNG} \leftarrow \mathrm{ANR}} \left(\rho_{\mathrm{ANGESTELLTER}} \left(\mathrm{PERS}\right)\right)$$

ANGESTELLTER / PERS	<u>PNR</u>	NAME	ALTER	GEHALT	ABTEILUNG / 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

Relationenalgebra - Operatoren (5)

- Klassische Mengenoperationen
  - **Voraussetzung: Gleicher Grad und Vereinigungsverträglichkeit der beteiligten Relationen**
  - **Basisoperatoren**

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

    Differenz: $R - S = \{t \mid t \in R \wedge t \notin S\}$
  - **Redundante Operatoren**

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

    Symmetrische Differenz: $R \, \Delta \, S = (R \cup S) - (R \cap S)$

Relationenalgebra - Operatoren (6)

- Erweitertes Kartesisches Produkt
  - $\mathbf{K = R \times S = \{ k \mid \exists x \in R, y \in S: (k = x|y)\}}$
    $\mathbf{mit\ x|y = <x_1, ..., x_r, y_1, ..., y_s>,}$

    nicht $<<x_1, ..., x_r>, <y_1, ..., y_s>>$
    wie ‚übliches' kartesisches Produkt!

PERS	PNR	ALTER	ANR
	406	47	K55
	123	32	K51
	829	36	K53

ABT	ANR	ANAME	ORT
	K51	PLAN.	KL
	K53	EINK.	F

ABT x PERS	ANR	ANAME	ORT	PNR	ALTER	ANR'
	K51	PLAN.	KL	406	47	K55
	K51	PLAN.	KL	123	32	K51
	K51	PLAN.	KL	829	36	K53
	K53	EINK.	F	406	47	K55
	K53	EINK.	F	123	32	K51
	K53	EINK.	F	829	36	K53

Relationenalgebra - Operatoren (7)

- Verbund, Join, Θ-Join
  - Seien R und S Relationen, $\Theta \in \{<, =, >, \leq, \neq, \geq\}$ (arithm. Vergleichsoperator), A Attribut von R und B Attribut von S. Θ-Verbund zwischen R und S:

$$\mathbf{V = (R \underset{A\,\Theta\,B}{\bowtie} S) = \sigma_{A\,\Theta\,B}(R \times S)}$$

  - Alternative Definition anhand Spaltennummern
    Annahme: R hat Grad r und S hat Grad s, $1 \leq i \leq r$, $1 \leq j \leq s$,

$$\mathbf{V = (R \underset{i\,\Theta\,j}{\bowtie} S) = \sigma_{i\,\Theta\,r+j}(R \times S)}$$

  - Gleichverbund (Θ = „=")
    - Ein Gleichverbund zwischen R und S heißt *verlustfrei*, wenn alle Tupel von R und S am Verbund teilnehmen (sonst *verlustbehaftet*). Die inverse Operation Projektion erzeugt dann wieder R und S (***lossless join***).

Relationenalgebra - Operatoren (8)

- Verbund, Join, Θ-Join (Forts.)
  - Definition ‚fortsetzbar' auf mehrere Join-Attribute
  - Natürlicher Verbund R $\bowtie$ S: Gleichverbund über alle übereinstimmenden (d.h. gleichnamigen) Attribute und anschließende Projektion, so dass keine Attribute doppelt vorkommt
  - Im Fall eines verlustfreien Verbund gilt:

    $\pi_{\text{ANR, ANAME, ORT}}$ (ABT $\bowtie$ PERS) = ABT

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

ABT	ANR	ANAME	ORT
	K51	PLAN	KL
	K53	EINK.	F
	K55	VERTR.	F

PERS	PNR	ALTER	ANR
	406	47	K55
	123	32	K51
	829	36	K53
	574	28	K55

ABT $\bowtie$ PERS	ANR	ANAME	ORT	PNR	ALTER
	K51	PLAN	KL	123	32
	K53	EINK.	F	829	36
	K55	VERTR.	F	406	47
	K55	VERTR.	F	574	28

Relationenalgebra - Operatoren (9)

- Definition *Natürlicher Verbund*
  - 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): $B_1 = A_{r-j+1}$, $B_2 = A_{r-j+2}$, ..., $B_j = A_r$
  - Natürlicher Verbund zwischen R und S:

$$N = R \bowtie S =$$

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

Relationenalgebra - Operatoren (10)

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

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

Relationenalgebra - Operatoren (11)

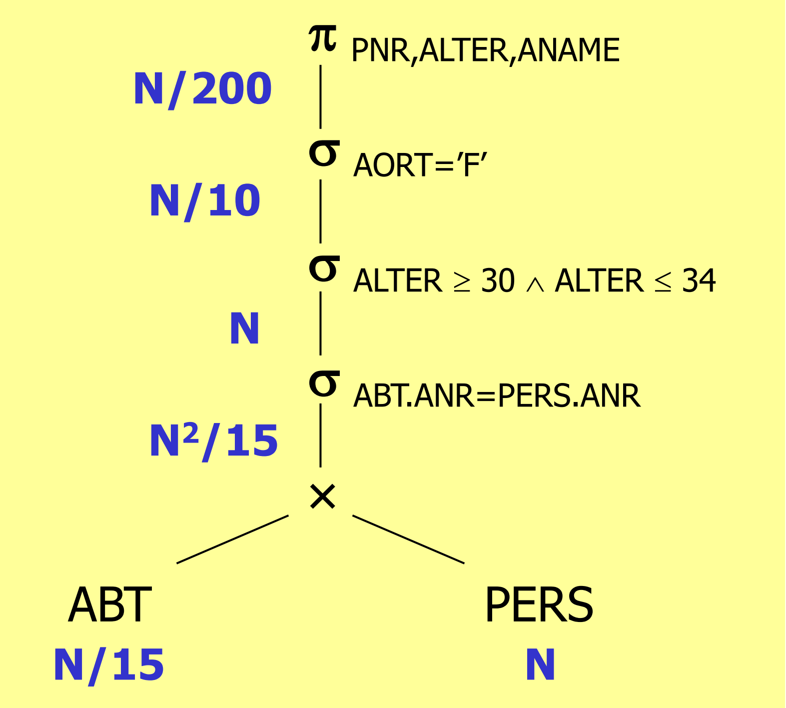
- Natürlicher Verbund – Beispiel (Forts.)
  - **Annahmen:**
    - ABT: N/15 Tupel
    - PERS: N Tupel
    - Gleichverteilung der Attributwerte
      AORT: 20 Werte
      ALTER: 50 Werte
    - Stochastische Unabhängigkeit der Werte verschiedener Attribute
    - Verlustfreie Verbunde von R1 und R2 über Primär-/Fremdschlüssel,
      mit Card(R1) < Card(R2): Card(R1 $\bowtie$ R2) = Card(R2)

Relationenalgebra - Operatoren (12)

- Natürlicher Verbund – Beispiel (Forts.)
  - **Lösung 1:**

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

zugehöriger Operatorbaum:

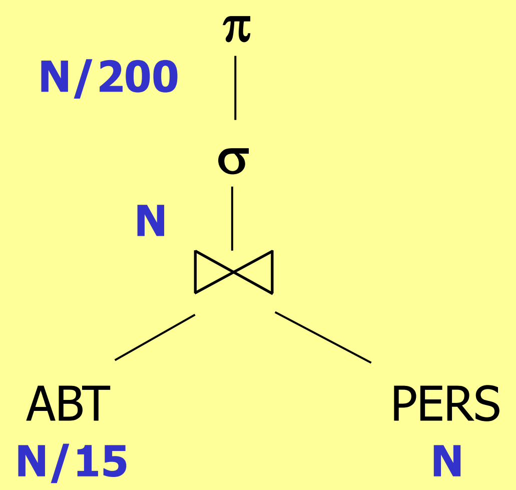


Relationenalgebra - Operatoren (13)

- Natürlicher Verbund – Beispiel (Forts.)
  - **Lösung 2:**

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

zugehöriger Operatorbaum:

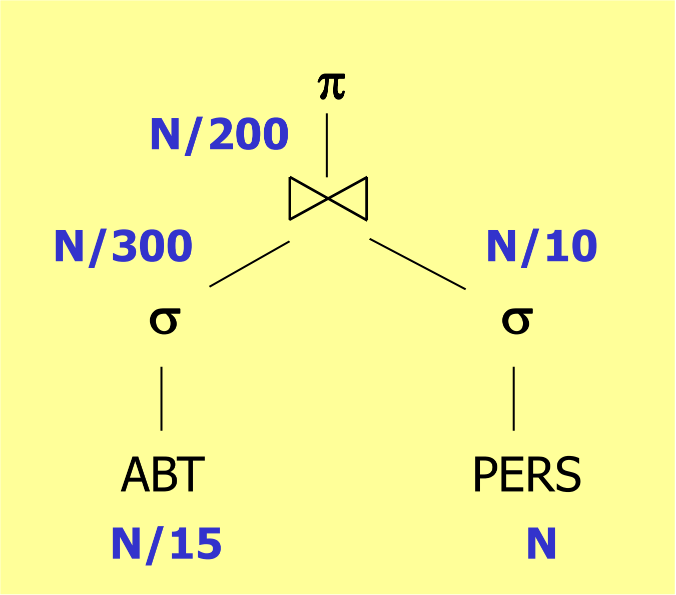


Relationenalgebra - Operatoren (14)

- Natürlicher Verbund – Beispiel (Forts.)
  - **Lösung 3:**

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

zugehöriger Operatorbaum:

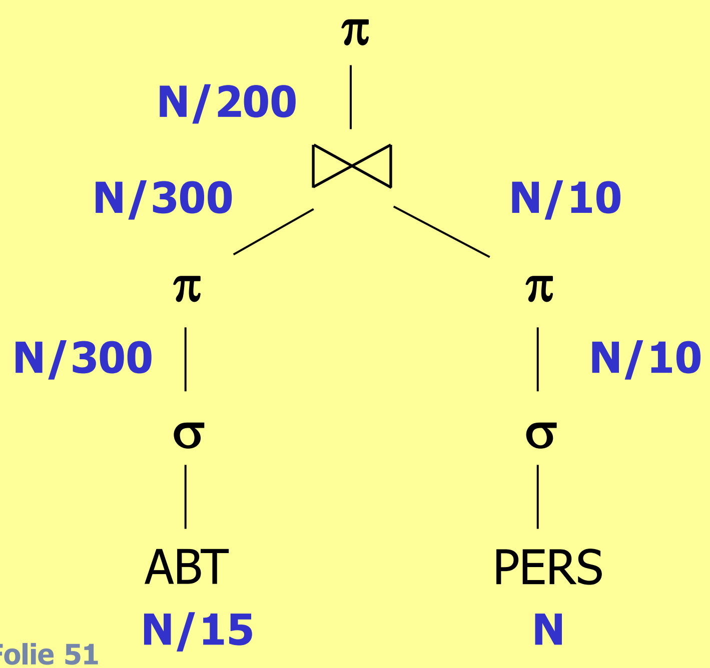


Relationenalgebra - Operatoren (15)

- Natürlicher Verbund – Beispiel (Forts.)
  - **Lösung 4:**

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

zugehöriger Operatorbaum:



Relationenalgebra - Operatoren (16)

- ACHTUNG: Connection Trap!
  - Verbund kann im Allg. nicht als Umkehroperation zur Projektion angesehen werden
  - Beispiel: DA1 und DA2 als Projektionen auf DA; DA3 als Verbund von DA1 und DA2 über A-ORT

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

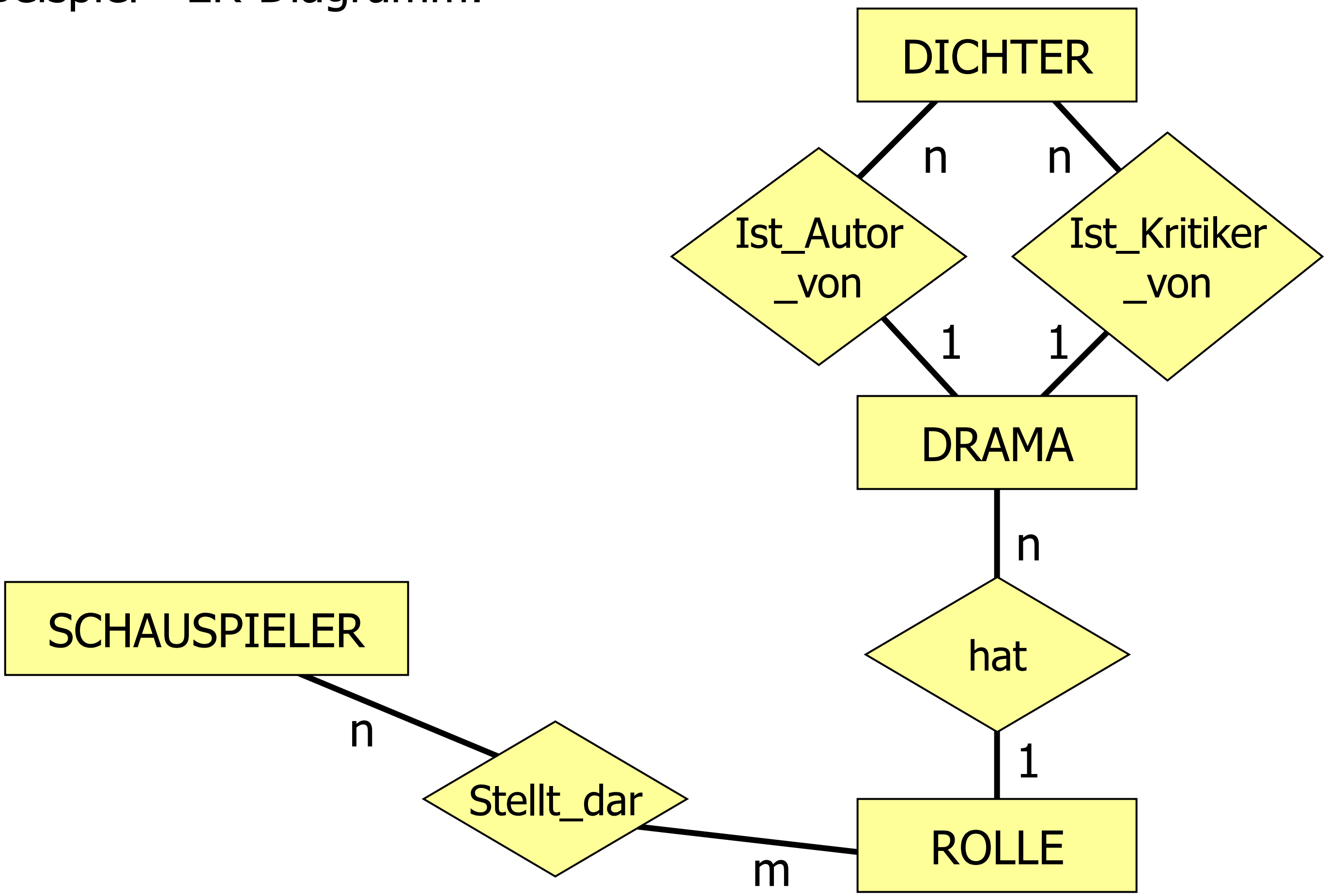
DA1	PNR	A-ORT
	P1	MA
	P1	KL
	P2	MA

DA2	FIGUR	A-ORT
	Faust	MA
	Mephisto	KL
	Wallenstein	MA

DA3	PNR	FIGUR	A-ORT
	P1	Faust	MA
	P1	Wallenstein	MA
	P1	Mephisto	KL
	P2	Faust	MA
	P2	Wallenstein	MA

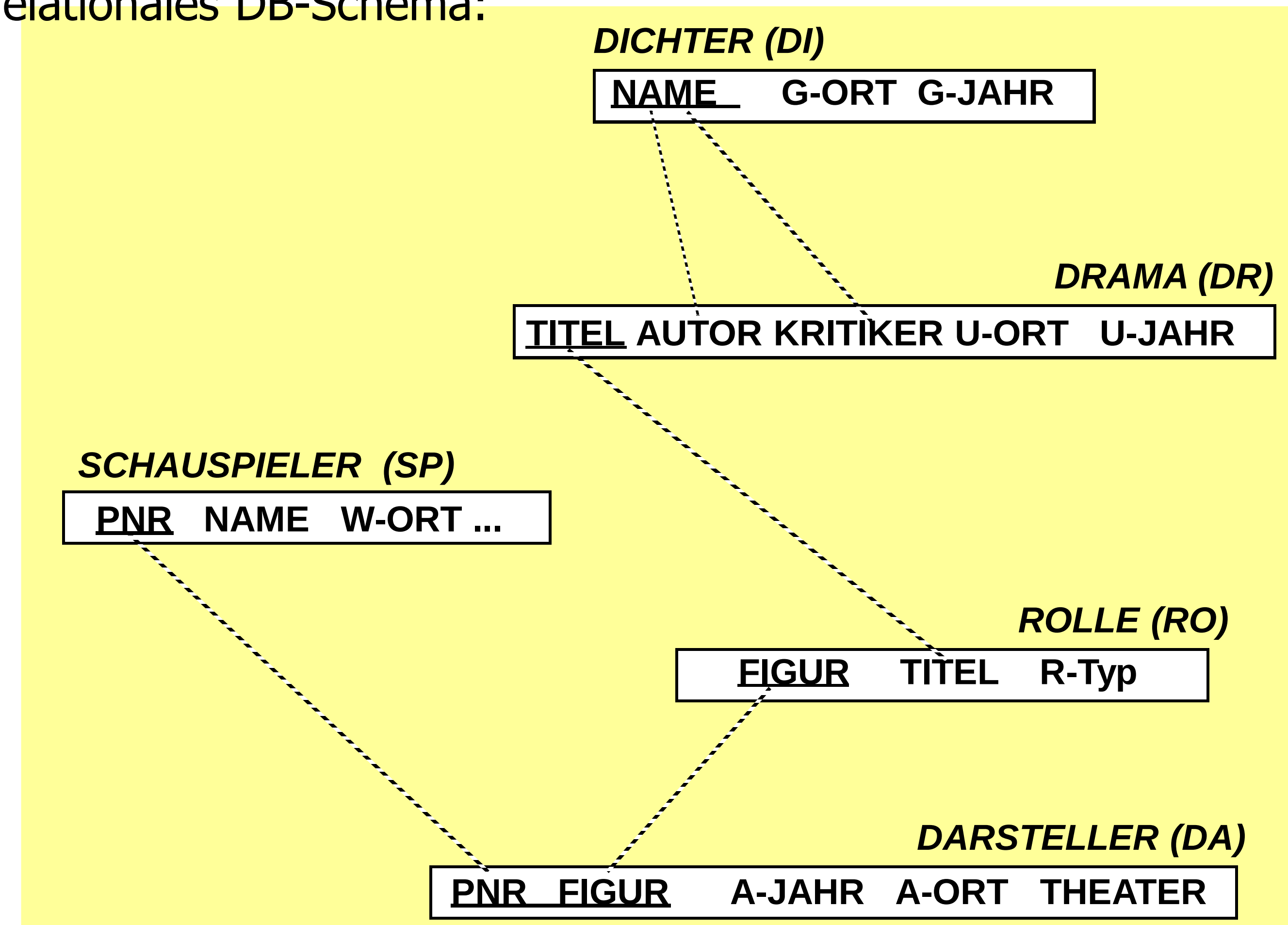
Relationenalgebra – Beispiel (1)

- Beispiel - ER-Diagramm:



Relationenalgebra – Beispiel (2)

- Beispiel – relationales DB-Schema:



Relationenalgebra – Beispiel (3)

- Beispiel – Anfragen:
  - Finde alle Schauspieler (NAME), die die Figur „Faust" gespielt haben.

$$\pi_{\text{NAME}} (\sigma_{\text{FIGUR=„FAUST"}} (\text{SP} \bowtie_{\text{PNR}} \text{DA}))$$

  - Finde alle Schauspieler (NAME), die im Drama „Faust" mitgespielt haben.

$$\pi_{\text{NAME}} (\sigma_{\text{TITEL = „Faust"}} (\text{SP} \bowtie_{\text{PNR}} \text{DA} \bowtie_{\text{FIGUR}} \text{RO}))$$

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

$$\pi_{\text{NAME}} (\sigma_{\text{AUTOR = „Schiller"}} (\text{SP} \bowtie_{\text{PNR}} \text{DA} \bowtie_{\text{FIGUR}} \text{RO} \bowtie_{\text{TITEL}} \text{DR}))$$

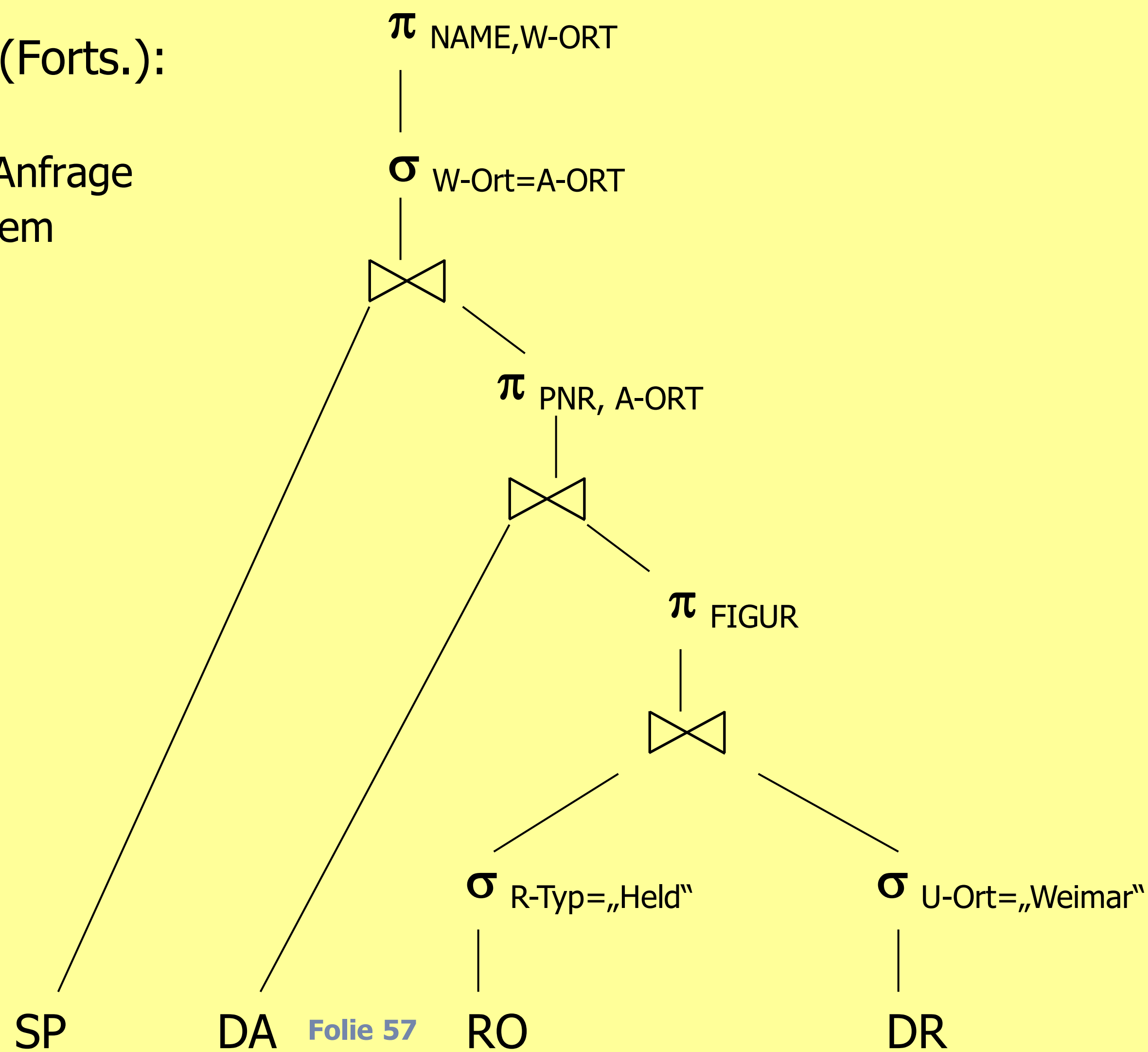
Relationenalgebra – Beispiel (4)

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

$$\pi_{\text{NAME, W-ORT}}\ (\sigma_{\text{W-ORT= A-ORT}}\ (\text{SP} \underset{\text{PNR}}{\bowtie} ($$
$$\pi_{\text{PNR, A-ORT}}\ (\text{DA} \underset{\text{FIGUR}}{\bowtie} (\pi_{\text{FIGUR}}\ ($$
$$(\sigma_{\text{R-TYP = 'HELD'}}\ \text{RO}) \underset{\text{TITEL}}{\bowtie} (\sigma_{\text{U-ORT = 'WEIMAR'}}\ \text{DR})\ )\ )\ )\ )\ )\ )$$

Relationenalgebra – Beispiel (5)

- Beispiel – Anfragen (Forts.):
  - Vorangegangene Anfrage entspricht folgendem Operatorbaum:



Relationenalgebra – Beispiel (6)

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

$$\pi_{\text{TITEL, AUTOR, G-JAHR}} (\sigma_{\text{U-JAHR} > 1800} (\text{DI} \underset{\text{NAME=AUTOR}}{\bowtie} \text{DR}))$$

  - Finde alle Schauspieler (NAME, W-ORT), die in Dramen von Schiller, die von in Weimar geborenen Dichtern kritisiert wurden, mitgespielt haben.

$$\pi_{\text{NAME, W-ORT}} (\text{SP} \underset{\text{PNR}}{\bowtie} \text{DA} \underset{\text{FIGUR}}{\bowtie} \text{RO} \underset{\text{TITEL}}{\bowtie}$$

$$(\rho_{\text{DNAME} \leftarrow \text{NAME}} (\sigma_{\text{AUTOR=„Schiller"}} (\text{DR})) \underset{\text{KRITIKER=NAME}}{\bowtie} (\sigma_{\text{G-ORT=„Weimar"}} (\text{DI}))))$$

Relationenalgebra – Beispiel (7)

- Beispiel – Anfragen (Forts.):
  - Finde die Schauspieler, die **nie** gespielt haben.

    π_{PNR} (SP) - π_{PNR} (DA))

  - Finde die Schauspieler, die **nur** Faust oder Wallenstein gespielt haben.

    π_{PNR} (DA) - π_{PNR} ($\sigma_{FIGUR \neq „FAUST“ \wedge FIGUR \neq „Wallenstein“}$ (DA)))

  - Anfragen wie „Welcher Dichter ist Schauspieler?“ oder „Welcher Dichter hat in einem seiner Stücke gespielt?“ können „eigentlich“ nicht beantwortet werden, da es keine systemkontrollierte Beziehung zwischen **Dichter** und **Schauspieler** gibt.

Algebraische Optimierung (1)

- Relationenalgebraische Formulierungen spezifizieren Ausführungs-reihenfolge (prozedurale Elemente), äquivalente Umformungen möglich
- **Optimierungsproblem**
  - gegeben: Ausdruck der Relationenalgebra (RA)
  - gesucht: äquivalenter, möglichst effizient auszuführender RA-Ausdruck
  - Bestimmung einer möglichst guten Ausführungsreihenfolge (Einsatz von Heuristiken)
- **Statistische Kenngrößen** werden dem DB-Katalog entnommen
  - $N_i = Card(R_i)$
  - j_i = Anzahl der verschiedenen Werte eines Attributs A_i

Algebraische Optimierung (2)

- **Rewrite-Regeln**
  - Kommutativgesetz für Produkte und Verbunde
    - $R1 \times R2 \equiv R2 \times R1$
    - $R1 \bowtie R2 \equiv R2 \bowtie R1$
  - Assoziativgesetz für Produkte und Verbunde
    - $(R1 \times R2) \times R3 \equiv R1 \times (R2 \times R3)$
    - $(R1 \bowtie R2) \bowtie R3 \equiv R1 \bowtie (R2 \bowtie R3)$
  - Zusammenfassung von Folgen von Projektionen
    - $\pi_{A,B,C} (\pi_{A,B,C,\ldots,Z} (SP)) \equiv \pi_{A,B,C} (SP)$
  - Zusammenfassung von Folgen von Selektionen
    - $\sigma_{F1} (\sigma_{F2} (R)) \equiv \sigma_{F1 \wedge F2} (R) \equiv \sigma_{F2 \wedge F1} (R) \equiv \sigma_{F2} (\sigma_{F1} (R))$

$\bowtie$ steht hier für beliebige Θ-Verbunde

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

Algebraische Optimierung (3)

- **Rewrite-Regeln (Forts.)**
  - Vertauschung von Selektionen und Projektionen
    - F enthält nur Attribute aus A, ..., Z:

      $\sigma_F(\pi_{A, ..., Z}(R)) \equiv \pi_{A, ..., Z}(\sigma_F(R))$
    - F enthält Attribute aus A, ..., Z, B1, ..., Bm:

      $\pi_{A, ..., Z}(\sigma_F(R)) \equiv \pi_{A, ..., Z}(\sigma_F(\pi_{A, ..., Z, B1, ..., Bm}(R)))$
  - Vertauschung von Selektion und Kartesischem Produkt
    - F enthält nur Attribute aus R1:

      $\sigma_F(R1 \times R2) \equiv \sigma_F(R1) \times R2$
    - allgemeiner: $F = F1 \wedge F2 \wedge F3$

      F1 nur auf R1, F2 nur auf R2, F3 auf beiden

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

Algebraische Optimierung (4)

- Annahmen
  - Gleichverteilung der Attributwerte eines Attributes
  - Stochastische Unabhängigkeit der Werte verschiedener Attribute
- Selektivitätsfaktor (SF)
  - basiert auf statistischen Werten
  - beschreibt hinsichtlich eines Qualifikationsprädikats den erwarteten Anteil an Tupeln, die das Prädikat erfüllen
  - $0 \leq SF \leq 1$
  - $Card\ (\sigma_p(R)) = SF(p) \cdot Card\ (R)$

Algebraische Optimierung (5)

- **SF-Berechnung**

j_i:Anzahl der Werte des Attributs A_i

- $A_i = a_i$ $$SF = \begin{cases} 1/j_i & \text{falls } j_i \text{ bekannt} \\ 1/10 & \text{sonst} \end{cases}$$
- $A_i = A_k$ $$SF = \begin{cases} 1/\max(j_i, j_k) & \text{falls } j_i \text{ und } j_k \text{ bekannt} \\ 1/j_i & \text{falls nur } j_i \text{ bekannt} \\ 1/10 & \text{sonst} \end{cases}$$

Algebraische Optimierung (6)

- **SF-Berechnung (Forts.)**

j_i:Anzahl der Werte des Attributs A_i

diskrete Domäne

- $A_i \geq a$ (oder $A_i > a$)

$$SF = \begin{cases} (\text{high-key} - a+1)/(\text{high-key} - \text{low-key}+1) & \text{falls keys bekannt} \\ 1/3 & \text{sonst} \end{cases}$$

- $A_i \geq a_i \wedge A_i \leq a_k$

$$SF = \begin{cases} (a_k - a_i+1)/ (\text{high-key} - \text{low-key}+1) & \text{falls keys bekannt} \\ 1/4 & \text{sonst} \end{cases}$$

stetige Domäne

- $A_i \geq a$ (oder $A_i > a$)

$$SF = \begin{cases} (\text{high-key} - a)/(\text{high-key} - \text{low-key}) & \text{falls keys bekannt} \\ 1/3 & \text{sonst} \end{cases}$$

- $A_i \geq a_i \wedge A_i \leq a_k$

$$SF = \begin{cases} (a_k - a_i)/ (\text{high-key} - \text{low-key}) & \text{falls keys bekannt} \\ 1/4 & \text{sonst} \end{cases}$$

Algebraische Optimierung (7)

- **SF-Berechnung bei Ausdrücken**
  - $SF\,(p(A) \wedge p(B)) = SF\,(p(A)) \cdot SF\,(p(B))$
  - $SF\,(p(A) \vee p(B)) = SF\,(p(A)) + SF\,(p(B)) - SF\,(p(A)) \cdot SF\,(p(B))$
  - $SF\,(\neg\, p(A)) = 1 - SF\,(p(A))$

- **Join-Selektivitätsfaktor (JSF)**
  - Card (RS) = JSF * Card(R) * Card(S)
  - bei (N:1)-Joins (verlustfrei): Card (RS) = Max(Card(R), Card(S))
  - bei verlustbehafteten (N:1)-Joins die Relationen über FK-PK Beziehungen verknüpfen: Card (RS) = Card(S), wenn S den Fremdschlüssel und R den zugehörigen Primärschlüssel besitzt (Annahme: keine Nullwerte im FK)
    - Beispiel: Relation ABT besitzt ein Attribut LEITER, welches auf den PK der Relation PERS referenziert

    Anmerkung: Wenn der FK-PK Join verlustfrei ist, gilt Card(S)>Card(R) und beide Regeln liefern das selbe Ergebnis

Algebraische Optimierung (8)

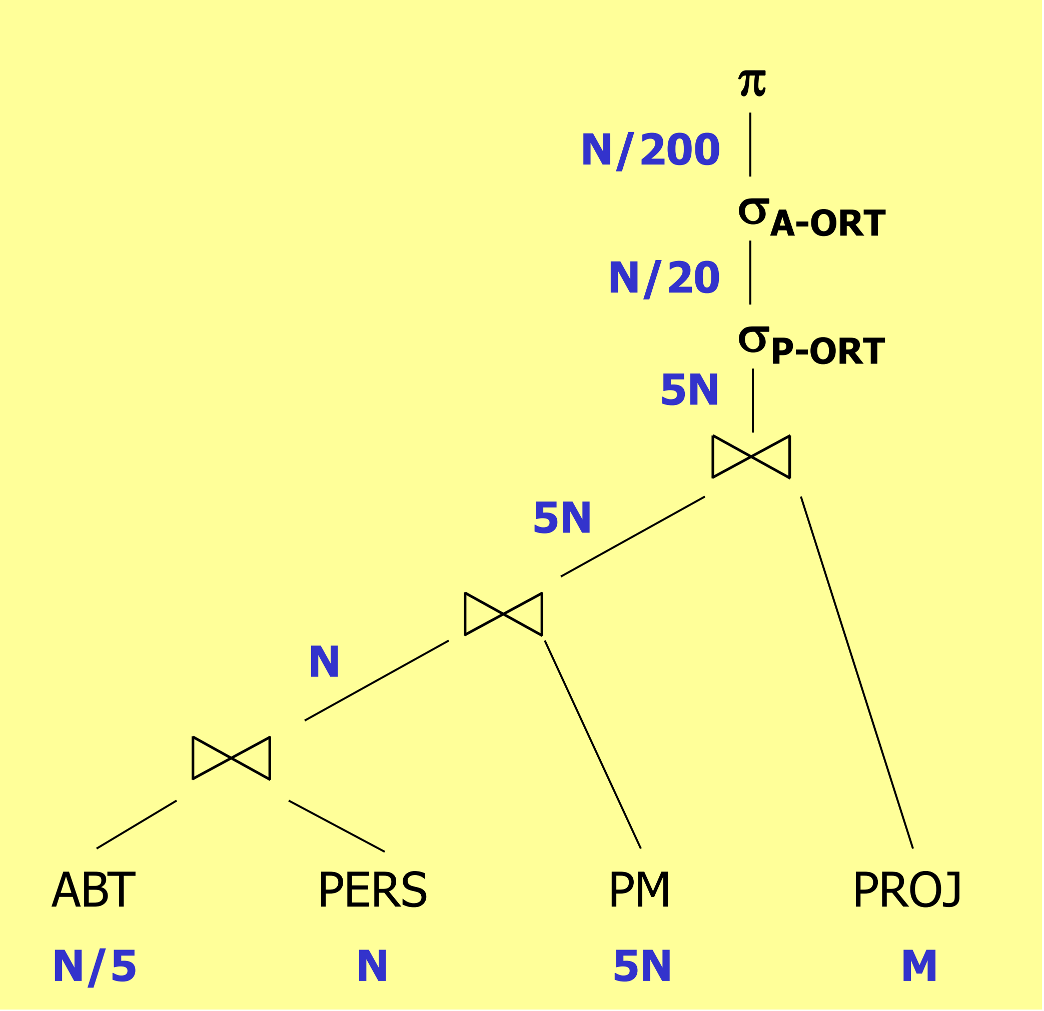
- **Beispiel**
  - **DB-Schema**

ABT	(ANR, BUDGET, A-ORT)
PERS	(PNR, NAME, BERUF, GEHALT, ALTER, ANR)
PM	(PNR, JNR, DAUER, ANTEIL)
PROJ	(JNR, BEZEICHNUNG, SUMME, P-ORT)

  - **Anfrage**: *Finde Name und Beruf von Angestellten, deren Abteilung in KL ist und die in KL Projekte durchführen.*
  - **Annahmen:**
    - ABT: N/5 Tupel
    - PERS: N Tupel
    - PM: 5N Tupel
    - PROJ: M Tupel
    - Anzahl der Attributwerte von A-ORT: 10, P-ORT: 100
    - Verlustfreie Verbunde von R1 und R2 über Primär-/Fremdschlüssel

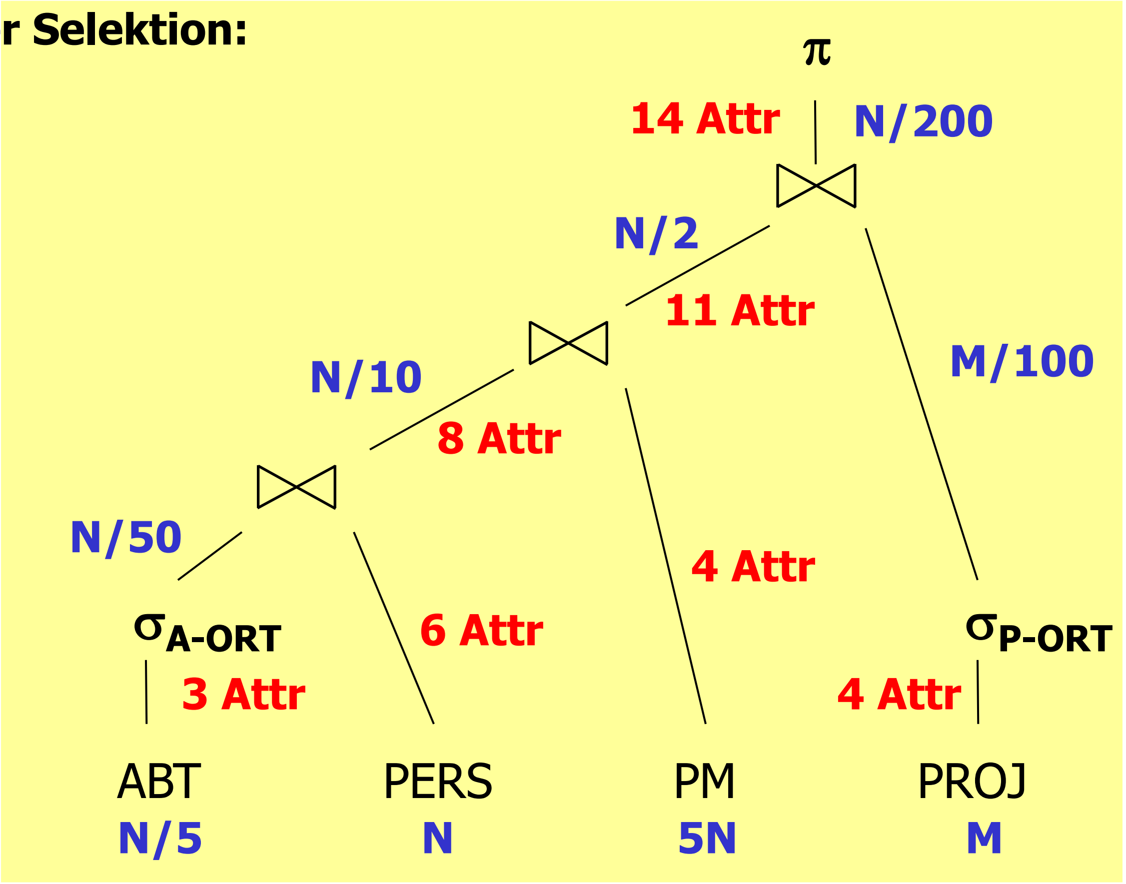
Algebraische Optimierung (9)

- **Beispiel (Forts.)**
  - **Ausgangslösung:**



Algebraische Optimierung (10)

- **Beispiel (Forts.)**
  - **Verschieben der Selektion:**



Algebraische Optimierung (10)

- **Beispiel (Forts.)**
  - **Verschieben der Selektion:**

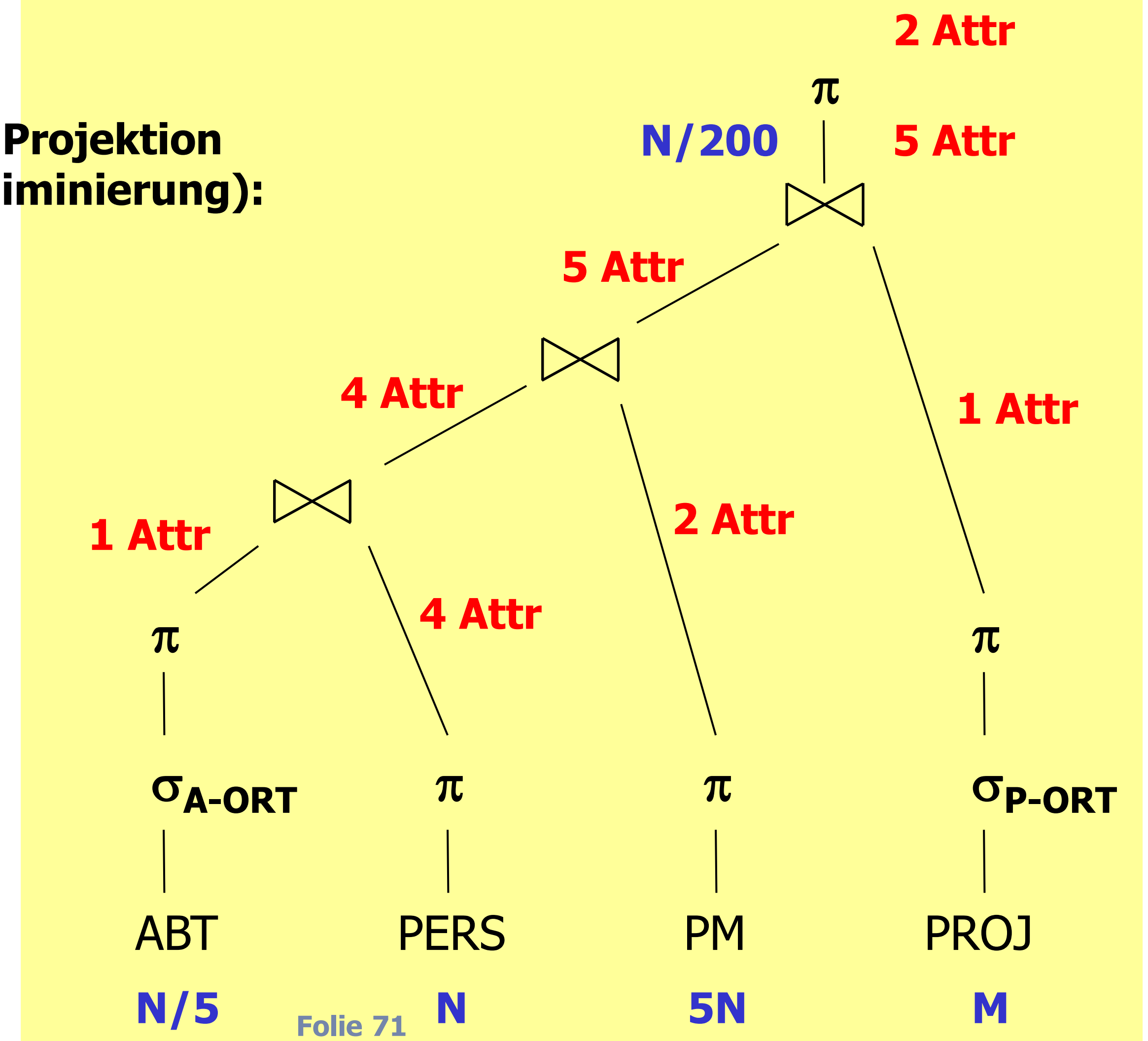
I.

Führe Selektion so früh

wie möglich aus!

Algebraische Optimierung (11)

- **Beispiel (Forts.)**
  - **Verschieben der Projektion (ohne Duplikateliminierung):**



Algebraische Optimierung (11)

- **Beispiel (Forts.)**
  - **Verschieben der Projektion (ohne Duplikateliminierung):**

II.

Führe Projektion

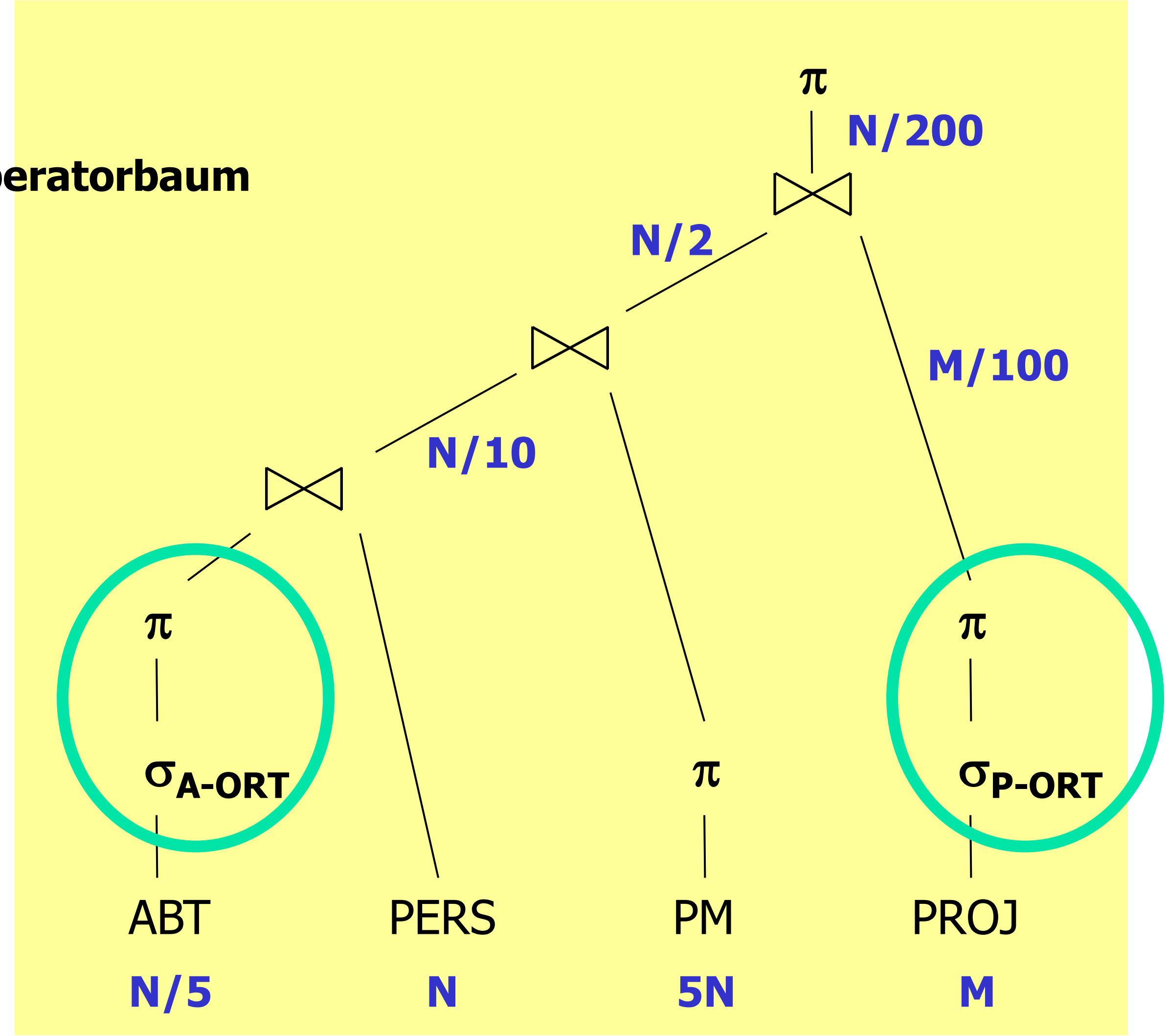
(ohne Duplikateliminierung)

so früh wie möglich aus!

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

Algebraische Optimierung (12)

- **Beispiel (Forts.)**
  - **Optimierter Operatorbaum (Vorschlag):**



Algebraische Optimierung (12)

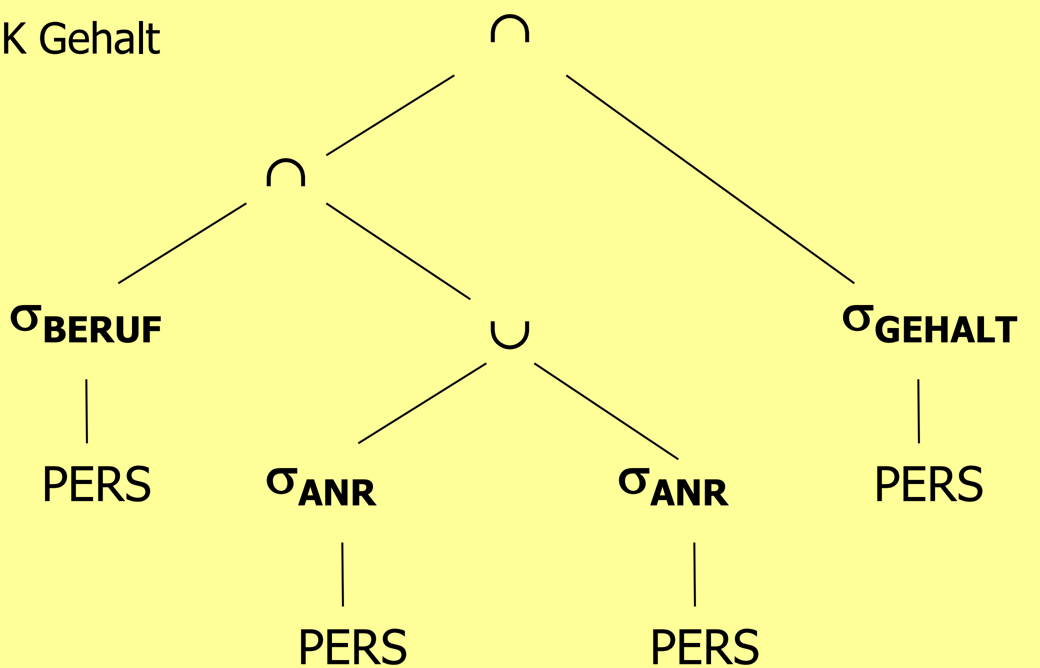
- **Beispiel (Forts.)**
  - **Optimierter Operatorbaum (Vorschlag):**

III.

Verknüpfe Folgen von unären Operatoren wie Selektion und Projektion (wenn diese tupelweise abgewickelt werden können)!

Algebraische Optimierung (13)

- **Weitere Optimierungsmaßnahmen**
  - **Ausdrucksauswertung**
    - Beispiel: Finde alle Programmierer aus Abteilung K51 oder K55 mit mehr als 50K Gehalt



Algebraische Optimierung (13)

- **Weitere Optimierungsmaßnahmen (Forts.)**
  - **Ausdrucksauswertung**
    - Beispiel: Finde alle Programmierer
      aus Abteilung K51 oder K55
      mit mehr als 50K Gehalt

$$\sigma_{\text{BERUF=„P“} \wedge \text{GEHALT>50K} \wedge (\text{ANR=„K51“} \vee \text{ANR=„K55“})}$$

PERS

Algebraische Optimierung (13)

- **Weitere Optimierungsmaßnahmen (Forts.)**
  - **Ausdrucksauswertung**
    - Beispiel: Finde alle Programmierer
      aus Abteilung K51 oder K55
      mit mehr als 50K Gehalt

IV.

Fasse einfache Selektionen

auf einer Relation zusammen!

Algebraische Optimierung (14)

- **Weitere Optimierungsmaßnahmen (Forts.)**

V.

Verknüpfe bestimmte Selektionen mit einem vorausgehenden Kartesischen Produkt zu einem Verbund!

VI.

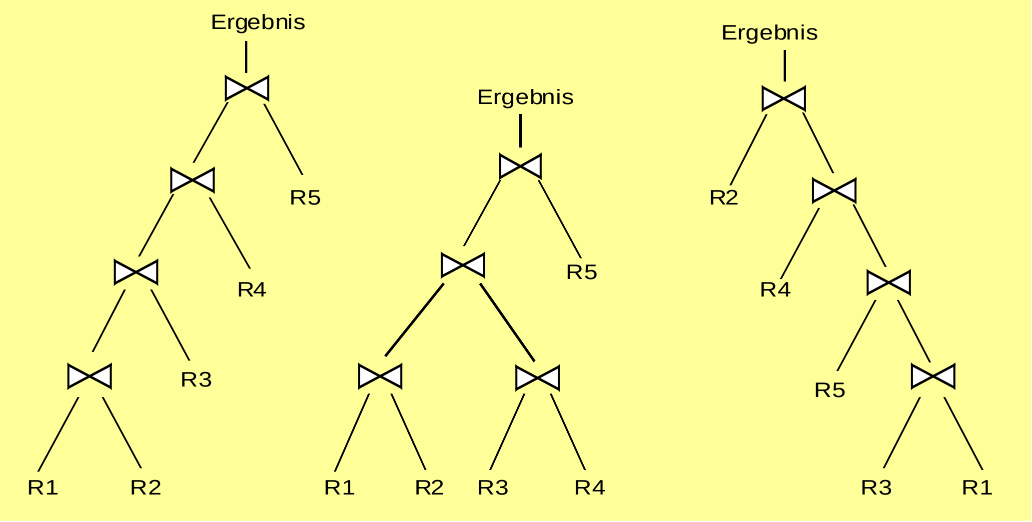
Berechne gemeinsame Teilbäume nur einmal (wenn die Zwischen-speicherung der Ergebnisse nicht zu teuer ist)!

Algebraische Optimierung (15)

- **Weitere Optimierungsmaßnahmen (Forts.)**
  - Kombination von Verbundoperationen
    - Assoziativität und Kommutativität von Verbundoperationen (gilt auch für Vereinigung und Durchschnitt)
    - **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
        - Dynamische Entscheidung aufwendiger, aber genauer als Abschätzung
        - Bei jedem Auswertungsschritt werden die momentan kleinsten (Zwischen-)Relationen ausgewählt

Algebraische Optimierung (16)

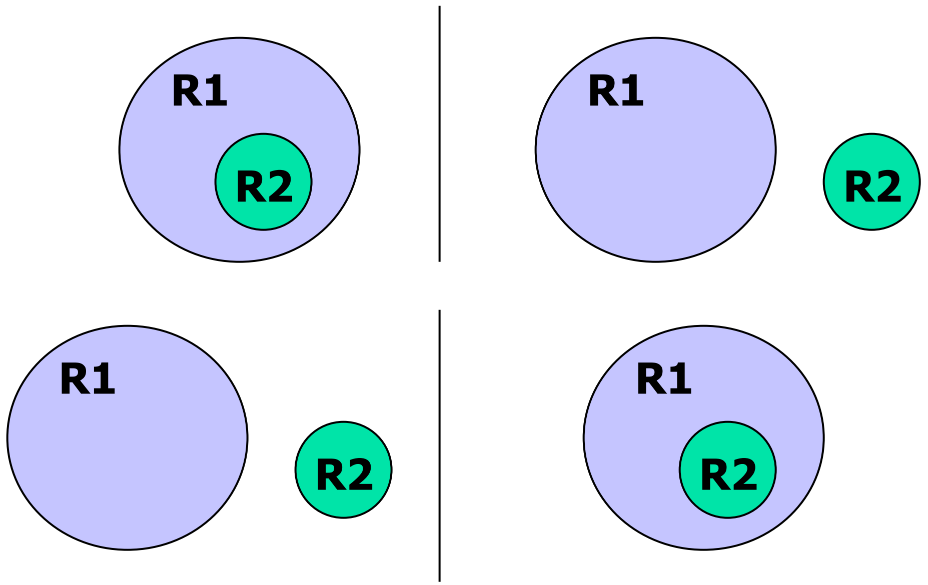
- **Weitere Optimierungsmaßnahmen (Forts.)**
  - Kombination von Verbundoperationen (Forts.)
    - **Allgemeines Problem bei binären Relationenoperationen (Forts.)**
      - Einige Verknüpfungsreihenfolgen für den Verbund mit n=5



VII. Bestimme die Verbundreihenfolge so, dass die Anzahl und Größe der Zwischenobjekte minimiert wird!

Algebraische Optimierung (17)

- **Weitere Optimierungsmaßnahmen (Forts.)**
  - Reihenfolge von Mengenoperationen
  - Kardinalität der Vereinigung:
    $\max(N(R1), N(R2))$
    $\leq N(R1 \cup R2)$
    $\leq N(R1) + N(R2)$
  - Kardinalität des Durchschnitts:
    0
    $\leq N(R1 \cap R2)$
    $\leq \min(N(R1), N(R2))$



VIII. Verknüpfe bei Mengenoperationen immer zuerst die kleinsten Relationen!

Algebraische Optimierung (18)

- **Heuristische Regeln:**
  - Führe Selektion so früh wie möglich aus
  - Führe Projektion (ohne Duplikateliminierung) frühzeitig aus
  - Verknüpfe Folgen von unären Operationen wie Selektion und Projektion
  - Fasse einfache Selektionen auf einer Relation zusammen
  - Verknüpfe bestimmte Selektionen mit einem vorausgehenden Kartesischen Produkt zu einem Verbund
  - Berechne gemeinsame Teilbäume nur einmal
  - Bestimme Verbundreihenfolge so, dass die Anzahl und Größe der Zwischenobjekte minimiert wird
  - Verknüpfe bei Mengenoperationen immer zuerst die kleinsten Relationen

Weitere Operatoren (1)

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

Weitere Operatoren (2)

- Division (Forts.)
  - Beispiel:

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

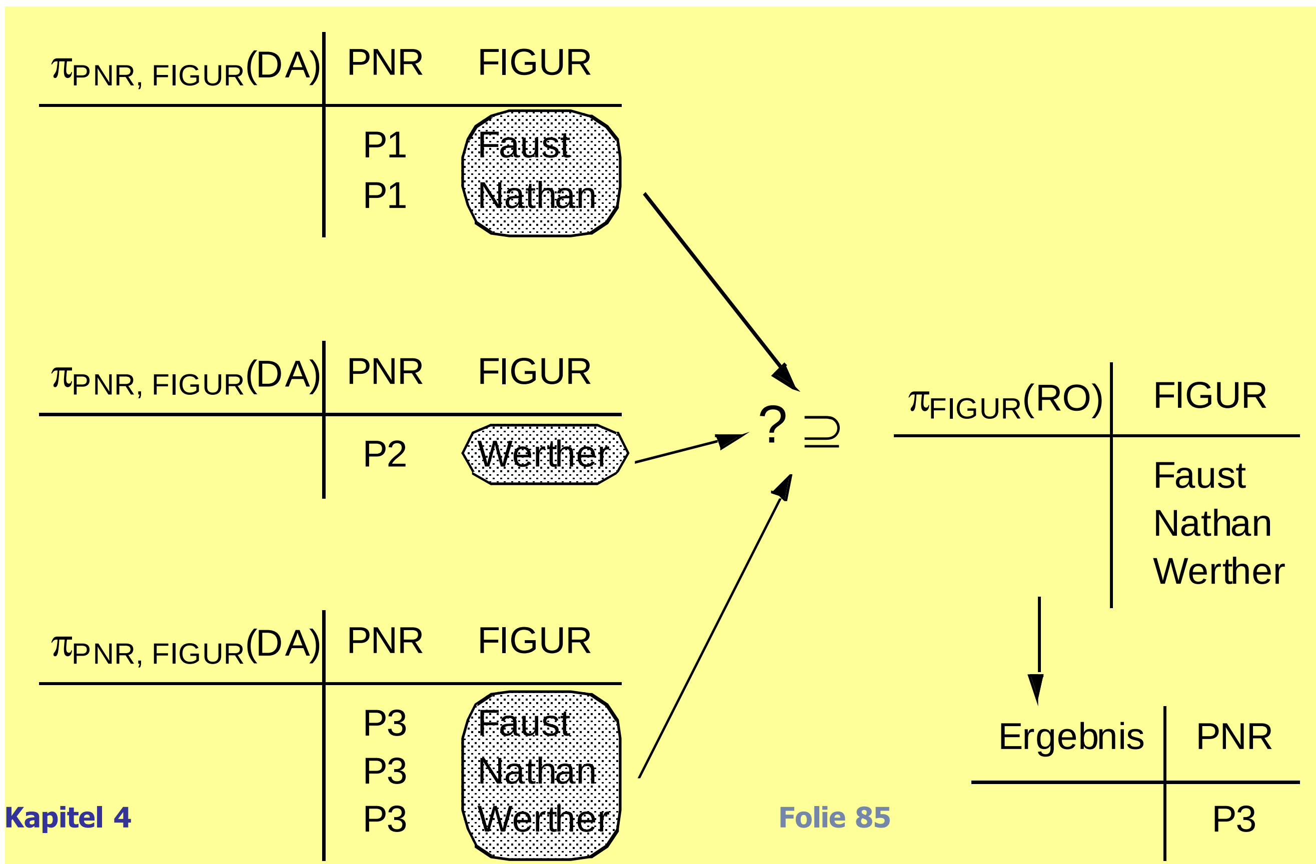
RO	FIGUR	TITEL	R-Typ
	Faust	Faust	
	Nathan	Nathan der Weise ...	
	Werther	Die Leiden ...	

- Welche Schauspieler haben alle Rollen gespielt:

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

Weitere Operatoren (3)

- Division (Forts.)
  - Beispiel (Forts.)
    - Welche Schauspieler haben alle Rollen gespielt: $(\pi_{\text{PNR, FIGUR}}(\text{DA})) \div (\pi_{\text{FIGUR}}(\text{RO}))$



Weitere Operatoren (4)

- Division (Forts.)
  - Beschreibung der Division mit den Grundoperatoren

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

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

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

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

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

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

Weitere Operatoren (5)

- Division (Forts.)
  - Weitere Beispiele
    - Finde alle Schauspieler (NAME), die **alle** Rollen in Dramen von Goethe gespielt haben.

$$\pi_{Name}(\ (SP \bowtie (\pi_{PNR,FIGUR}(DA))) \div (\pi_{FIGUR}(RO \bowtie (\sigma_{AUTOR=„Goethe“}(DR))))\ )$$

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

$$\pi_{Name}(\ (SP \bowtie (\pi_{PNR,FIGUR}(\sigma_{THEATER=...}(DA))) \div (\pi_{FIGUR}(\sigma_{R\text{-}Typ=...}(RO)))\ )$$

Weitere Operatoren (6)

- Teilverbund (Semi Join)
  - Berechnet den Anteil eines Verbundes (Θ oder natürlich), welcher nach einer Projektion auf die Attribute einer der beiden Relationen übrig bleibt.
  - Grob: Berechnet alle Tupel einer Relationen die einen Join-Partner in der anderen Relation besitzen.
  - Teilverbund von R zu S:

$$R \ltimes_{R.A=S.A} S = \pi_{R.*}(R \bowtie_{R.A=S.A} S)$$

PERS	PNR	ALTER	ANR
	406	47	K55
	123	32	K51
	829	36	K53
	574	28	K57

ABT	ANR	ANAME	ORT
	K51	PLAN	KL
	K53	EINK.	F
	K55	VERTR.	F

ABT $\rtimes$ PERS	PNR	ALTER	ANR
	406	47	K55
	123	32	K51
	829	36	K53

Weitere Operatoren (7)

- Anti-Verbund (Anti Join)
  - Berechnet das Komplement eines Teilverbundes (Θ oder natürlich).
  - Grob: Berechnet alle Tupel einer Relationen die **keinen** Join-Partner in der anderen Relation besitzen.
  - Anti-Verbund von R zu S:

$$R \underset{R.A=S.A}{\triangleright} S = R - (R \underset{R.A=S.A}{\ltimes} S)$$

PERS	PNR	ALTER	ANR
	406	47	K55
	123	32	K51
	829	36	K53
	574	28	K57

ABT	ANR	ANAME	ORT
	K51	PLAN	KL
	K53	EINK.	F
	K55	VERTR.	F

ABT $\triangleleft$ PERS	PNR	ALTER	ANR
	574	28	K57

Weitere Operatoren (8)

- 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 positive Konstanten, wobei eine größer Null sein muss.
  - Intervall-Verbund zwischen R und S:

$$V = R \underset{i|j}{\bowtie} S = \sigma_{i|j} (R \times S) = \sigma_{\mathbf{R.i-c_1} \leq \mathbf{S.j} \leq \mathbf{R.i+c_2}} (R \times S)$$

Weitere Operatoren (9)

- Intervallverbund (*Band Join*)
  - 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: $G = \sigma_{PNR \neq PNR'} (PERS \underset{ALTER\ [2,2]\ ALTER'}{\bowtie} PERS')$

PERS	PNR	ALTER
	P1	25
	P2	23
	P3	28

G	PNR	ALTER	PNR'	ALTER'
	P1	25	P2	23
	P2	23	P1	25

Weitere Operatoren (10)

- Äußerer Verbund (*Outer Join*)
  - Ziel: ‚Verlustfreiheit' soll erzwungen werden!
  - Trick: Einfügen spezieller Leerzeilen zur künstlichen Erzeugung von Verbundpartnern
    - Beispiel

SP	PNR	NAME
	P1	x
	P2	y

DA	PNR	FIGUR
	P1	F
	P1	W
	P3	M

$$SP' = SP \cup ((\pi_{PNR}(DA) - \pi_{PNR}(SP)) \times \equiv)$$

$$DA' = DA \cup ((\pi_{PNR}(SP) - \pi_{PNR}(DA)) \times \equiv)$$

Weitere Operatoren (11)

- Äußerer Verbund (Forts.)
  - Definition
    - Seien A die Verbundattribute, „ $\equiv$" der undefinierte Wert und

$$R\text{`} = R \cup ((\pi_A(S) - \pi_A(R)) \times \equiv \times \equiv \ldots)$$

$$S\text{`} = S \cup ((\pi_A(R) - \pi_A(S)) \times \equiv \times \equiv \ldots)$$

Äußerer Gleichverbund:

$$R \;⟗_{R.A=S.A}\; S := R\text{`} \bowtie_{R\text{`}.A=S\text{`}.A} S\text{`}$$

Äußerer natürlicher Verbund:

$$R \;⟗\; S := R\text{`} \bowtie S\text{`}$$

Weitere Operatoren (12)

- Äußerer Verbund (Forts.)
  - 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

      $$R \mathbin{\sqsupset\!\!\bowtie}_{R.A=S.A} S := R \bowtie_{R.A=S'.A} S'$$

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

      $$R \mathbin{\bowtie\!\!\sqsubset}_{R.A=S.A} S := R' \bowtie_{R'.A=S.A} S$$

Weitere Operatoren (13)

- Äußerer Verbund (Forts.)
  - Beispiele:

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

$\bowtie$

ERG	A	B	C	D	E
	a_1	b_1	c_1	d_1	e_1

⟖

ERG	A	B	C	D	E
	a_1	b_1	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	--	--

⟗

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 Operatoren (14)

- 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	MATRNR	FBNR	SEM
	123	FB5	5
	789	FB9	9

HIWI	MATRNR	FBNR	JOB
	465	FB5	TUTOR
	987	FB9	PROGR

STUD-HIWI	MATRNR	FBNR	SEM	JOB
	123	FB5	5	-
	789	FB9	9	-
	465	FB5	-	TUTOR
	987	FB9	-	PROGR

Achtung: Interpretationsprobleme beim Ergebnis!

In ähnlicher Weise lassen sich weitere Operationen einführen (Outer Intersection/Difference), die jedoch nicht besonders nützlich sind!

Relationenalgebra

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