

# 4. Grundlagen des Relationenmodells<sup>1</sup>

- **Übersicht**

- Grundkonzepte
- Normalisierte Relationen
- Schlüssel

- **Abbildung ERM → RM**

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

- **Relationenalgebra**

- Klassische Mengenoperationen
- Relationenoperationen
- Anfragen

- **Relationenalgebra – Optimierung**

- Rewrite-Regeln
- Algebraische Optimierung: Beispiel

# 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

1. *Forbes* called the relational model one of the most important innovations of the past 85 years, placing it squarely in the company of more widely known inventions such as the polio vaccine, automatic transmissions, fast food, disk drives, the mouse, ATMs, CDs, microprocessors, index funds, the Internet, and the World Wide Web.

## Relationenmodell – Grundkonzepte<sup>2</sup>

### ERM

Einwertiges Attribut  
 Definitionsbereich  
 Primärschlüssel

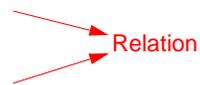
### RM

} wie im ERM

Zusammengesetztes Attribut  
 Mehrwertiges Attribut

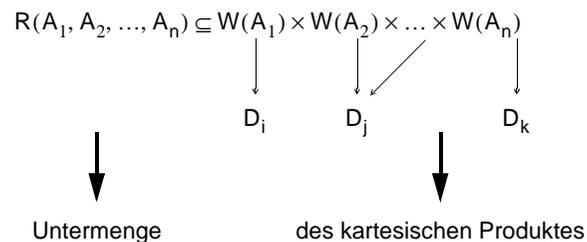
nur als unabhängige  
 einwertige Attribute

Entity-Menge



Relationship-Menge

#### • Definition: Normalisierte Relation



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

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

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

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

2. Dr. E. F. Codd described the relational model of data – one firmly grounded in predicate logic and mathematics – in a series of papers published between 1969 and 1981. He was named an IBM fellow in 1976 and received the highest technical honor in computing, the Turing award, in 1981.

## 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     |

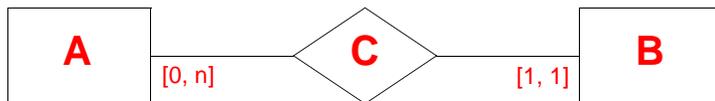
#### • Grundregeln<sup>3</sup>:

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

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

## Relationenmodell – Grundkonzepte (2)

- **Informationsdarstellung im RM**
  - ausschließlich durch Werte eines  $W(A_i)$  in Relationen
  - Reihenfolge von Zeilen und Spalten enthält keine Information
- **Wie werden Beziehungen dargestellt?**
  - Fremdschlüssel ist in Bezug auf Primärschlüssel / Schlüsselkandidat einer Relation definiert (gleicher Definitionsbereich)
  - Paare von Fremdschlüssel und Primärschlüssel / Schlüsselkandidaten
  - Beziehungen sind dadurch wertbasiert und symmetrisch!
- **Modellinhärente Integritätsbedingungen:  
Welche Zusicherungen werden vom Datenmodell garantiert?**
  - Mengeneigenschaft von Relationen
    - ➔ zur Abbildung von Entities/Relationships
  - Beziehungstypen (1:1, ..., n:m) ➔ mit Einschränkungen als (1:n)
  - Referentielle Integrität ➔ wertbasierte Beziehungen
  - Kardinalitätsrestriktionen? ➔ wünschenswert
  - Semantik der benutzerdefinierten Beziehung?
    - ➔ Es ist keine Systemunterstützung vorgesehen

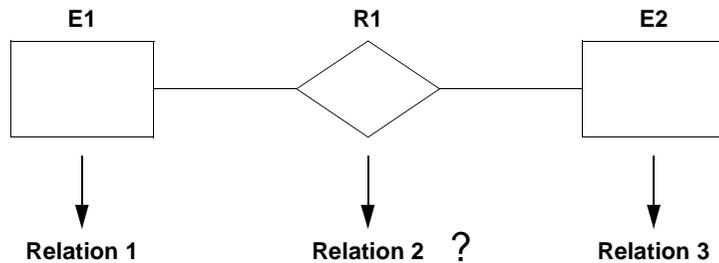


## Fremdschlüssel

- **Definition:**

Ein Fremdschlüssel bzgl. einer Relation R1 ist ein (ggf. zusammengesetztes) Attribut FS einer Relation R2, für das zu jedem Zeitpunkt gilt: **zu jedem Wert** (ungleich Null) **von FS** muss **ein gleicher Wert des Primärschlüssels** PS oder eines Schlüsselkandidaten SK in irgendeinem Tupel von Relation R1 vorhanden sein.
- **Bemerkungen:**
  1. Fremdschlüssel und zugehöriger Primärschlüssel (Schlüsselkandidat) tragen wichtige interrelationale (manchmal auch intrarelationale) Informationen. Sie sind auf dem gleichen Wertebereich definiert (vergleichbar und vereinigungsverträglich). Sie gestatten die Verknüpfung von Relationen mit Hilfe von Relationenoperationen.
  2. Fremdschlüssel können Nullwerte aufweisen, wenn sie nicht Teil eines Primärschlüssels sind oder wenn nicht explizit NOT NULL spezifiziert ist.
  3. Schlüsselkandidaten können Nullwerte aufweisen, wenn nicht explizit NOT NULL spezifiziert ist.
  4. Ein Fremdschlüssel ist zusammengesetzt, wenn der zugehörige Primärschlüssel (Schlüsselkandidat) zusammengesetzt ist.
  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**).

## Abbildung von Entity- und Relationship-Mengen



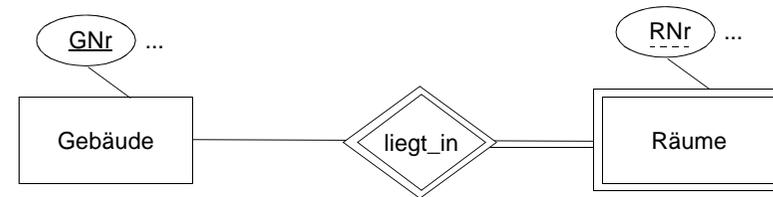
### Kriterien

- 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

## (Initiale) Abbildung von Entity-Mengen

### Jeder Entity-Typ E wird auf eine Relation R abgebildet

- R enthält alle einfachen Attribute von E
- R enthält alle einfachen Komponentenattribute der zusammengesetzten Attribute von E
  - Mengewertige Attribute erfordern gesonderte Behandlung
- Der Primärschlüssel von R ergibt sich aus
  - dem Primärschlüssel von E, bzw.
  - aus der Kombination der Primärschlüssel der bestimmenden Entity-Typen und des partiellen Schlüssels, falls E ein existenzabhängiger Entity-Typ ist und einen partiellen Schlüssel definiert

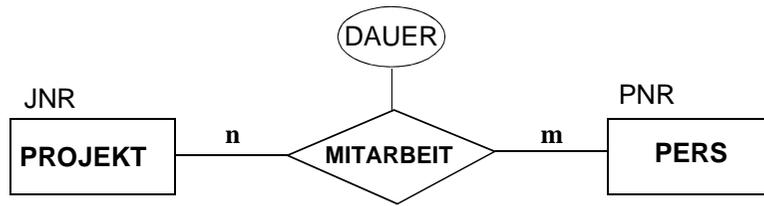


Darstellung der Entity-Typen im RM:

**GEBÄUDE**(GNR, ...)

**RÄUME**(RNR, ...)

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



Verwendung von drei Relationen erforderlich

PROJEKT (JNR, BEZEICH, ...)

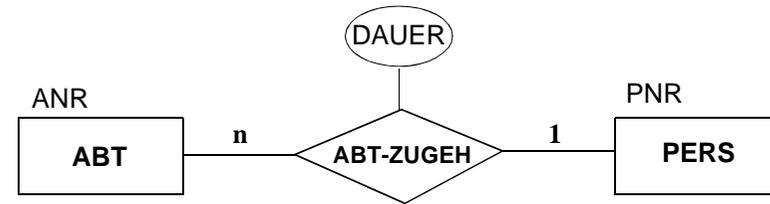
PERS (PNR, PNAME, ...)

MITARBEIT (JNR, PNR, DAUER)

- **Regel:**

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

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



Verwendung von  
drei Relationen                      zwei Relationen

ABT(ANR, ANAME, ...)

ABT(ANR, ANAME, ...)

PERS(PNR, PNAME, ...)

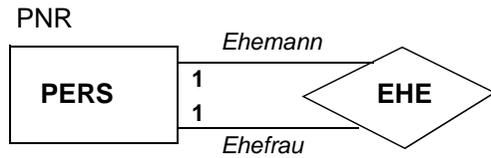
PERS(PNR, PNAME, ..., ANR, DAUER)

ABT-ZUEGH(ANR, PNR, DAUER)

- **Regel:**

(1:n)-Beziehungen lassen sich **ohne eigene Relation** darstellen. Hierzu wird in der referenzierenden Relation (mit Beziehungskardinalität 1) der Primärschlüssel der referenzierten Relation als Fremdschlüssel verwendet. Wenn eine **(1:n)-Beziehung eigene Attribute besitzt**, müssen dann auch diese Attribute in die referenzierende Relation aufgenommen werden. Das ist technisch möglich, jedoch kann die Verständlichkeit (Vermischung von Entity- und Relationship-Attributen) darunter leiden.

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



1.) Verwendung von zwei Relationen

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

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

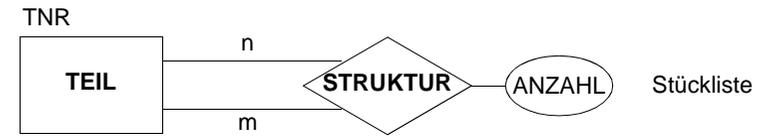
2.) Verwendung von einer Relation

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

• **Regel:**

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

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



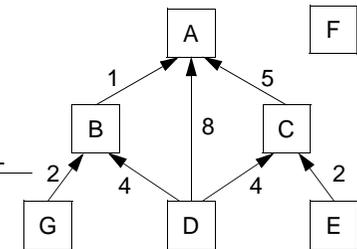
Darstellungsmöglichkeit im RM:

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

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

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

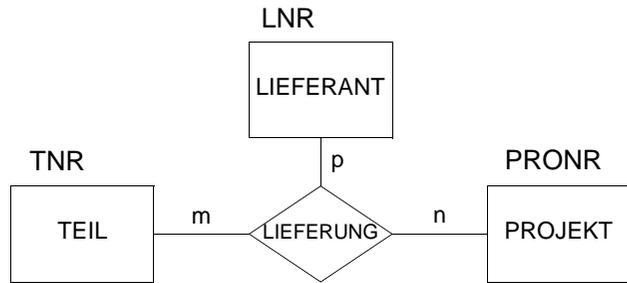
| STRUKTUR | <u>OTNR</u> | <u>UTNR</u> | ANZAHL |
|----------|-------------|-------------|--------|
| A        | B           | 1           |        |
| A        | C           | 5           |        |
| A        | D           | 8           |        |
| B        | D           | 4           |        |
|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



• **Regel:**

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

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



#### Darstellungsmöglichkeit im RM:

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

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

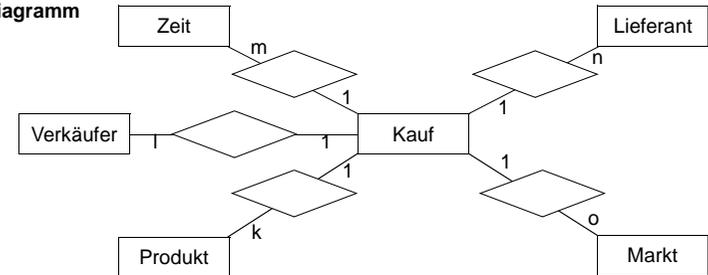
TEIL (TNR, TBEZ, GEWICHT, ...)

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

### Mehrere Entity-Mengen mit (n:m)-Verknüpfung (2)

- Data Warehousing – Kauf als Entity mit unabhängigen Beziehungen

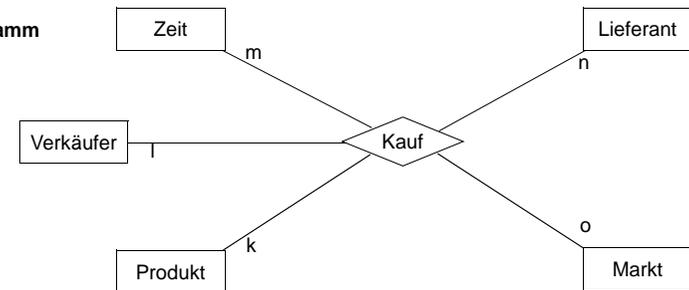
ER-Diagramm



- Produkt (Pnr, Bezeichnung, ...)
- Verkäufer (Vnr, VName, ...)
- Zeit (Znr, Datum, Besonderheit, ...)
- Lieferant (Lnr, LName, Ort, ...)
- Markt (Mnr, Adresse, ...)
- Kauf (Knr, Pnr, Vnr, Znr, Lnr, Mnr, Menge, Preis)

- Data Warehousing – Kauf als 5-stellige Beziehung

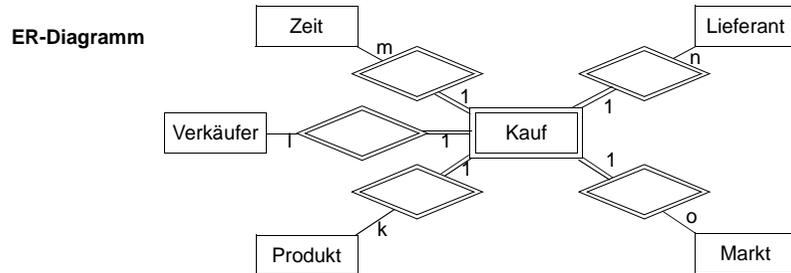
ER-Diagramm



- Produkt (Pnr, Bezeichnung, ...)
- Verkäufer (Vnr, VName, ...)
- Zeit (Znr, Datum, Besonderheit, ...)
- Lieferant (Lnr, LName, Ort, ...)
- Markt (Mnr, Adresse, ...)
- Kauf (Pnr, Vnr, Znr, Lnr, Mnr, Menge, Preis)

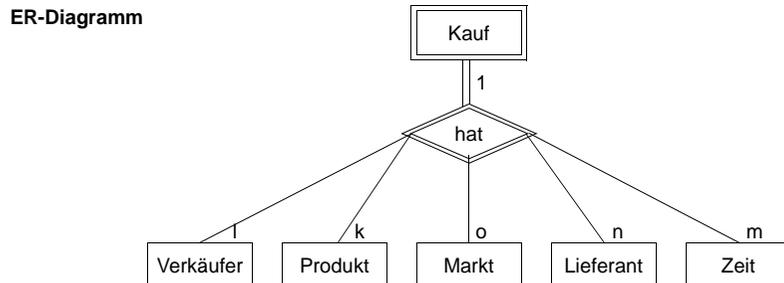
## Beispiel – Data Warehousing (2)

- Data Warehousing – Kauf als existenzabhängiges Entity mit 5 begründenden binären Beziehungen



Produkt (Pnr, Bezeichnung, ...)  
 Verkäufer (Vnr, VName, ...)  
 Zeit (Znr, Datum, Besonderheit, ...)  
 Lieferant (Lnr, LName, Ort, ...)  
 Markt (Mnr, Adresse, ...)  
 Kauf (Pnr, Vnr, Znr, Lnr, Mnr, Menge, Preis)  
 .....  
 .....

- Data Warehousing – Kauf als existenzabhängiges Entity mit einer begründenden 6-stelligen Beziehung



Produkt (Pnr, Bezeichnung, ...)  
 Verkäufer (Vnr, VName, ...)  
 Zeit (Znr, Datum, Besonderheit, ...)  
 Lieferant (Lnr, LName, Ort, ...)  
 Markt (Mnr, Adresse, ...)  
 Kauf (Pnr, Vnr, Znr, Lnr, Mnr, Menge, Preis)  
 .....  
 .....

## Abbildungstypen innerhalb einer Entity-Menge

- Entity-Menge kann viele Attribute besitzen

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

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

- Horizontale Partitionierung

- Klassenbildung anhand von Selektionsbedingungen

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

PERS (PNR, ..., VWL) GEHALT ≤ 100K

➔ ist eigentlich Aufgabe eines Sichtkonzeptes

- Vertikale Partitionierung

- zur leichteren Einhaltung von Leistungs- und Schutzaspekten:

PERS-ÖFF (PNR, PNAME, ADRESSE, ...)

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

➔ ist Aufgabe des internen Schemas und des Sichtkonzeptes

## Abbildungstypen innerhalb einer Entity-Menge (2)

- Abbildung mehrwertiger Attribute

- Entity-Menge:

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

P1, Müller, {Schnitzel, Braten, Rollmops}, -

P2, Schulz, {Pizza}, {(Nadine, 5), (Philip, 2)}

- Darstellungsmöglichkeit im RM:

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

**LIEBLINGSESSEN**(PNR, GERICHT, )

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

## Abbildung der Generalisierung im RM

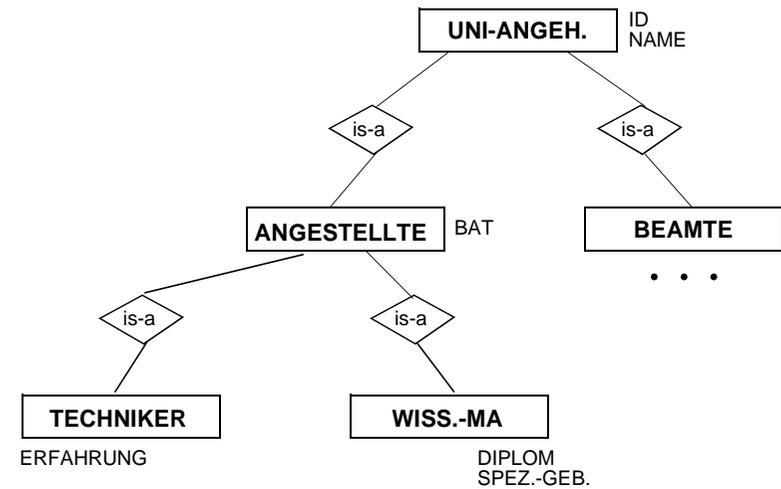
- Einschränkungen des Relationenmodells

- keine Unterstützung der Abstraktionskonzepte

- keine Maßnahmen zur Vererbung  
(von Struktur, Integritätsbedingungen, Operationen)

- „Simulation“ der Generalisierung eingeschränkt möglich

### Generalisierungsbeispiel:



## Generalisierung – relationale Sicht

### • LÖSUNG 1: Hausklassen-Modell:

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

| UNI-ANGEH. | ID  | NAME  |
|------------|-----|-------|
|            | 111 | Ernie |

| ANGESTELLTE | ID  | NAME     | BAT |
|-------------|-----|----------|-----|
|             | 007 | Garfield | Ia  |

| TECHNIKER | ID  | ERFAHRUNG | NAME   | BAT |
|-----------|-----|-----------|--------|-----|
|           | 123 | SUN       | Donald | IVa |

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

### • Eigenschaften:

- niedrige Speicherkosten und keine Änderungsanomalien
- Retrieval kann rekursives Suchen in Unterklassen erfordern
- explizite Rekonstruktion durch Relationenoperationen

➔ **Beispiel:** Finde alle ANGESTELLTE:

TECHNIKER  $\cup$  WISS.-MA  $\cup$  ANGESTELLTE, beschränkt auf ID, NAME, BAT

## Generalisierung – relationale Sicht

### • LÖSUNG 2: Partitionierungs-Modell

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

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

| TECHNIKER | ID  | ERFAHRUNG |
|-----------|-----|-----------|
|           | 123 | SUN       |

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

### • Eigenschaften

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

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

Verbinde TECHNIKER, ANGESTELLTE, UNI-ANGEH über gleiche IDs

## Generalisierung – relationale Sicht

### • Lösung 3: Volle Redundanz

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

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

| TECHNIKER | ID  | NAME   | BAT | ERFAHRUNG |
|-----------|-----|--------|-----|-----------|
|           | 123 | Donald | IVa | SUN       |

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

### • Eigenschaften

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

## Generalisierung – relationale Sicht

### • Lösung 4: Hierarchierelation

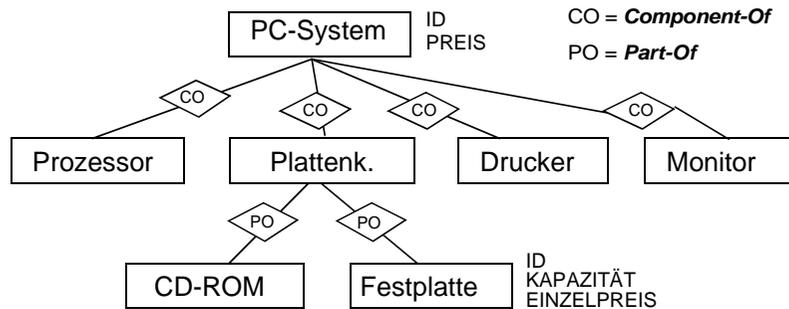
- Generalisierungshierarchie wird in einer einzigen Relation gespeichert.
- Attribut zur Typidentifikation (TT - type tag):  
Kennzeichnung der Klassenzugehörigkeit einer Instanz
- Nullwerte für Attribute, die in der zugeh. Klasse nicht vorhanden (definiert oder geerbt) sind

| ID  | TT                 | NAME     | BAT | ERFAHR. | DIPLOM     | SPEZ.-GEB. |
|-----|--------------------|----------|-----|---------|------------|------------|
| 007 | <i>Angestellte</i> | Garfield | Ia  | -       | -          | -          |
| 111 | <i>Uni-Angeh.</i>  | Ernie    | -   | -       | -          | -          |
| 123 | <i>Techniker</i>   | Donald   | IVa | SUN     | -          | -          |
| 333 | <i>Wiss.-MA</i>    | Daisy    | Ila | -       | Informatik | Recovery   |
| 765 | <i>Wiss.-MA</i>    | Grouch   | Ila | -       | Mathematik | ERM        |

### • Eigenschaften

- erhöhte Speicherkosten  
(Typidentifikator, Nullwerte für nicht zutreffende Attribute)
- Vermeidung von Redundanz
- Vermeidung von Verbund, Union
- Retrieval erfordert Projektion und mglw. zusätzliche Selektionsbedingungen über TT abhängig von der Zielklasse

## Aggregation - relationale Sicht



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

| Plattenk. | ID  | CD-ROM | Festplatte | Monitor | ID  | Hersteller | Preis |
|-----------|-----|--------|------------|---------|-----|------------|-------|
|           | PK1 | C1     | F1         |         | 15" | Belinea    | 320,- |
|           | PK2 | C2     | F2         |         | 17" | Belinea    | 420,- |
|           | PK3 | C3     | F3         |         | 17" | Belinea    | 420,- |

| CD-ROM | ID | Geschwindigk. | Preis | Festplatte | ID | Kapaz. | Preis |
|--------|----|---------------|-------|------------|----|--------|-------|
|        | C1 | DVD R         | 35.-  |            | F1 | 80 GB  | 85.-  |
|        | C2 | DVD R+CD RW   | 85.-  |            | F2 | 120 GB | 100.- |
|        | C3 | DVD R+RW      | 165.- |            | F3 | 160 GB | 150.- |

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

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

➔ Eigenschaften der Aggregation werden durch relationale Operationen nicht unterstützt

## Sprachen für das Relationenmodell

- Datenmodell = Datenobjekte + Operatoren
- DB-Sprachen wollen oft verschiedene Benutzerklassen unterstützen
  - Anwendungsprogrammierer
  - DB-Administratoren
  - anspruchsvolle Laien
  - parametrische Benutzer
  - gelegentliche Benutzer
- Im RM wird vereinheitlichte Sprache angestrebt für
  - alle Aufgaben der Datenverwaltung
    - Datendefinition
    - Anfragen (Queries)
    - Datenmanipulation
    - Zugriffs-, Integritäts- und Transaktionskontrolle
  - zur Nutzung
    - im Stand-Alone-Modus (Ad-hoc-Anweisungen) und
    - in einer Wirtssprache (eingebettete DB-Anweisungen)
- ➔ Die wichtigsten Eigenschaften von Anfragesprachen werden am Beispiel der Relationenalgebra diskutiert und anschließend zusammengefasst
- Vier verschiedene Grundtypen<sup>4</sup>:
  - Relationenalgebra (z. B. ISBL)
  - Relationenkalkül (z. B. Alpha)
  - Abbildungsorientierte Sprachen (z. B. SQL)
  - Graphikorientierte Sprachen (z. B. Query-by-Example)

4. The relational model separates data from the details of its physical storage so users and applications don't have to know where to look for the data they need.

## Relationenalgebra – Überblick

- Objekte:

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

→ Relationen sind Mengen.

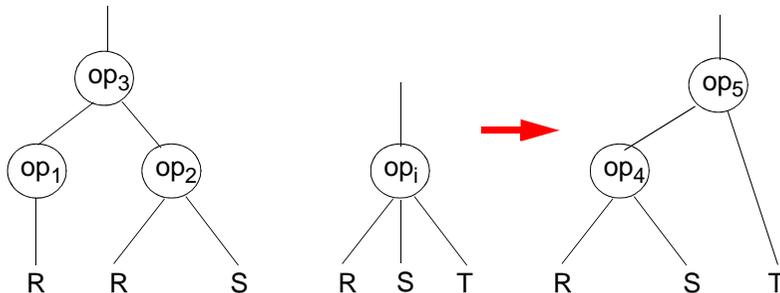
- Operationen:

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

Eigenschaften: **Ad-hoc-Sprache, Deskriptivität, Mengenorientierung, ...**

### Abgeschlossenheit

nur unäre und binäre Operationen



- Klassische Mengenoperationen:

- Vereinigung, Differenz, kartesisches Produkt
- ableitbar: Durchschnitt

- Relationenoperationen:

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

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

## Selektion (Restriktion)

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

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

- a) Operanden
  - Konstanten
  - Attributnamen
- b)  $\Theta \in \{<, =, >, \leq, \neq, \geq\}$
- c)  $\vee, \wedge, \neg$

- Definition:

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

- Beispiele:

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

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

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

| 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 |

- Ergebnis:

| ERG | <u>PNR</u> | NAME | ALTER | GEHALT | ANR | MNR |
|-----|------------|------|-------|--------|-----|-----|
|     |            |      |       |        |     |     |

## Projektion

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

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

! Duplikate entfernt !

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

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

- **Beispiel:**

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

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

| PERS | <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 |

**Ergebnis:**

| ERG | ANR | MNR |
|-----|-----|-----|
|     |     |     |

## Relationenalgebra – Beispiel-DB

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

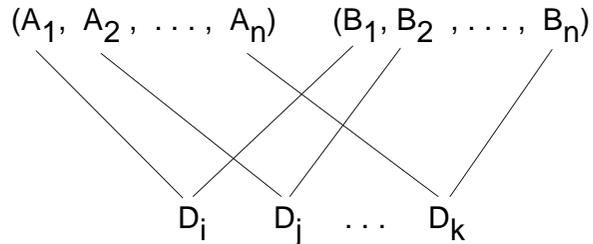
| PERS | <u>PNR</u> | NAME   | ALTER | GEHALT | ANR | MNR |
|------|------------|--------|-------|--------|-----|-----|
|      | 406        | Coy    | 47    | 50 700 | K55 | 123 |
|      | 123        | Müller | 32    | 43 500 | K51 | -   |
|      | 829        | Schmid | 36    | 45 200 | K53 | 777 |
|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

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

## Klassische Mengenoperationen

- **Voraussetzung: Vereinigungsverträglichkeit der beteiligten Relationen**

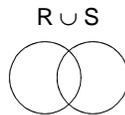
Gleicher Grad – Gleiche Bereiche



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

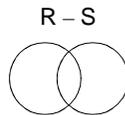
### 1. Vereinigung (UNION) von R und S

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



### 2. Differenz

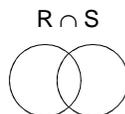
$$R - S = \{t | t \in R \wedge t \notin S\}$$



- **zusätzlich (redundante Mengenoperationen):**

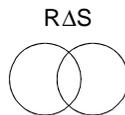
### 3. Durchschnitt (INTERSECTION)

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



### 4. Symmetrische Differenz

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



## (Erweitertes) Kartesisches Produkt

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

$$\begin{aligned} K &= R \times S \\ &= \{k | \exists x \in R, y \in S : (k = x|y)\} \end{aligned}$$

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

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

- **Anwendung von: ABT x PERS**

| ABT | ANR | ANAME    | AORT | PERS | PNR | ALTER | ANR |
|-----|-----|----------|------|------|-----|-------|-----|
|     | K51 | Planung  | KL   |      | 406 | 47    | K55 |
|     | K53 | Einkauf  | F    |      | 123 | 32    | K51 |
|     | K55 | Vertrieb | F    |      | 829 | 36    | K53 |
|     |     |          |      |      | 574 | 28    | K55 |

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

→ Iterative Schleifenmethode (Nested-Loop-Algorithmus)

## Umbenennung von Attributen und Relationen

- **Auflösung von Namenskonflikten kann erforderlich sein, wenn**

- ein Attributname in mehreren beteiligten Relationen auftaucht
- die gleiche Relation mehrmals in Operationen verwendet wird

- **Implizite Umbenennung von Attributen**

- Name der Ursprungsrelation wird als Präfix verwendet

Beispiel: Ermittle Kombinationen von Abteilungen mit Personen, die in den Abteilungen arbeiten

$$\pi_{\text{ABT.ANR, ANAME, PNR}} (\sigma_{\text{ABT.ANR} = \text{PERS.ANR}} (\text{ABT} \times \text{PERS}))$$

- **Explizite Umbenennung durch Rename-Operator ( $\rho$ )**

- Umbenennung eines Attributs

$$\pi_{\text{ANR, ANAME, PNR}} (\sigma_{\text{ANR} = \text{ABTNR}} (\text{ABT} \times \rho_{\text{ABTNR} \leftarrow \text{ANR}} (\text{PERS})))$$

- Umbenennung von Relationen durch Angabe des neuen Relationennamens

Beispiel: Ermittle Paare von Personen, die in der gleichen Abteilung arbeiten

$$\sigma_{\text{P1.ANR} = \text{P2.ANR}} (\rho_{\text{P1}} (\text{PERS}) \times \rho_{\text{P2}} (\text{PERS}))$$

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

- **Grob:**

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

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

$\Theta$ -Verbund zwischen R und S:

$$V = R \bowtie_{\Theta} S$$

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

- **Bemerkungen:**

- (1) Speziell  $\Theta = '=' : \text{Gleichverbund (Equi-Join)}$
- (2) statt i und j: Attributnamen A und B  
z. B.:  $R \bowtie_{\Theta} S \equiv R \bowtie_{A\Theta B} S$
- (3) Ein Gleichverbund zwischen R und S heißt **verlustfrei**, wenn alle Tupel von R und S am Verbund teilnehmen. Die inverse Operation Projektion erzeugt dann wieder R und S (**lossless join**)

## Gleichverbund – Beispiel

- Anwendung von:

**ABT** ⋈ **PERS**  
 ANR = ANR

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

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

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

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

- Verlustbehafteter Gleichverbund,**

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

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

## Natürlicher Verbund (Natural Join)

- grob:**

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

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

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

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

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

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

$\vdots$

$B_j = A_r$

### Natürlicher Verbund zwischen R und S:

$N = R \bowtie S$

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

$\bowtie = \text{Zeichen für Natural Join} \Rightarrow \Theta = '='$

- Bemerkung:**

Attribute sind durch Übereinstimmungsbedingung gegeben

## Natürlicher Verbund – Beispiel

- Anwendung von:

ABT  $\bowtie$  PERS

| ABT | ANR | ANAME    | AORT | PERS | PNR | ALTER | ANR |
|-----|-----|----------|------|------|-----|-------|-----|
|     | K51 | Planung  | KL   |      | 406 | 47    | K55 |
|     | K53 | Einkauf  | F    |      | 123 | 32    | K51 |
|     | K55 | Vertrieb | F    |      | 829 | 36    | K53 |
|     |     |          |      |      | 574 | 28    | K55 |

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

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

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

- Verlustbehafteter natürlicher Verbund analog zu Gleichverbund

## Natürlicher Verbund - Beispiel (2)

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

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

- Annahmen:

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

- Anfrage:

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

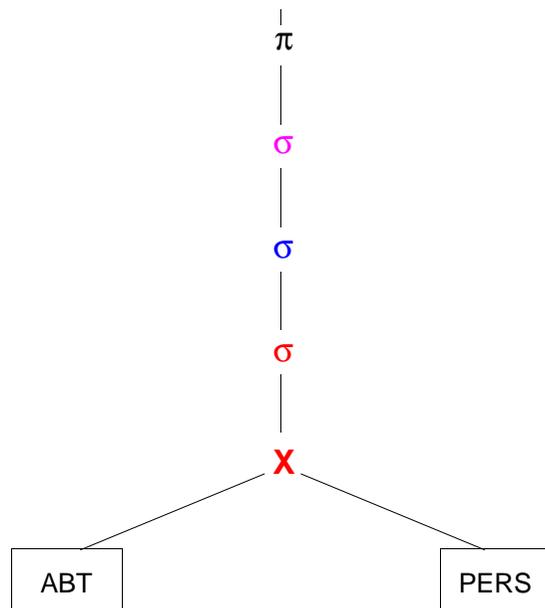
## Natürlicher Verbund – Lösungen

- Lösung 1:

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

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

- Zugehöriger Operatorbaum



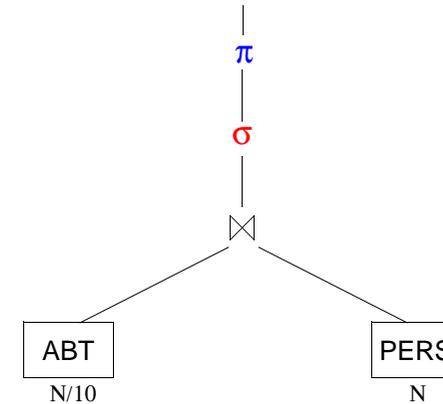
4 - 37

## Natürlicher Verbund – Lösungen (2)

- Lösung 2:

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

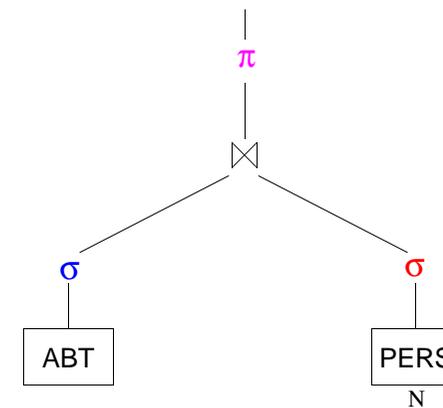
$(\sigma_{\text{ALTER} \geq 30 \wedge \text{ALTER} \leq 34 \wedge \text{AORT}='F'} (\text{ABT} \bowtie \text{PERS}))$



- Lösung 3:

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

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



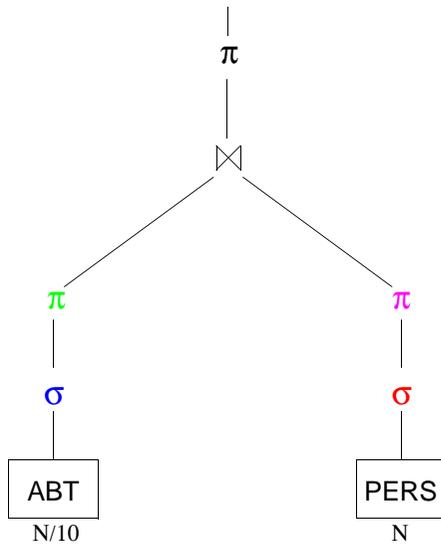
4 - 38

## Natürlicher Verbund – Lösungen (3)

- Lösung 4:

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

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



## Natürlicher Verbund - Beispiel (3)

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

- Beispiel 1 (1:n):

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

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

- Beispiel 2 (n:m):

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

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

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

| = DA1 | PNR | A-ORT |
|-------|-----|-------|
|       |     |       |
|       |     |       |
|       |     |       |

| = DA2 | FIGUR | A-ORT |
|-------|-------|-------|
|       |       |       |
|       |       |       |
|       |       |       |

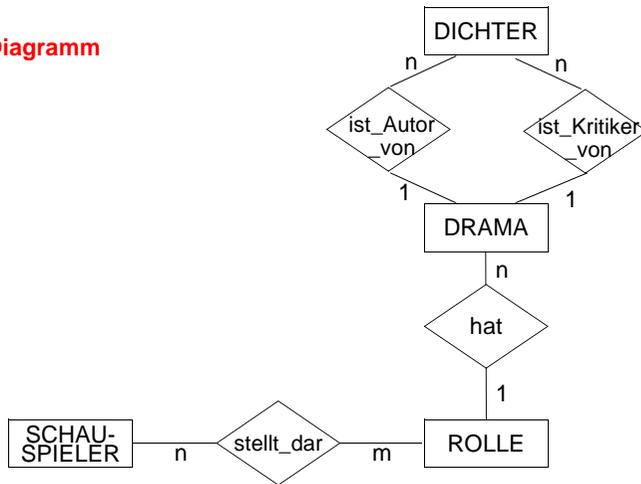
$\text{DA3} = \text{DA1} \bowtie \text{DA2}$

| = DA3 | PNR | FIGUR | A-ORT |
|-------|-----|-------|-------|
|       |     |       |       |
|       |     |       |       |
|       |     |       |       |

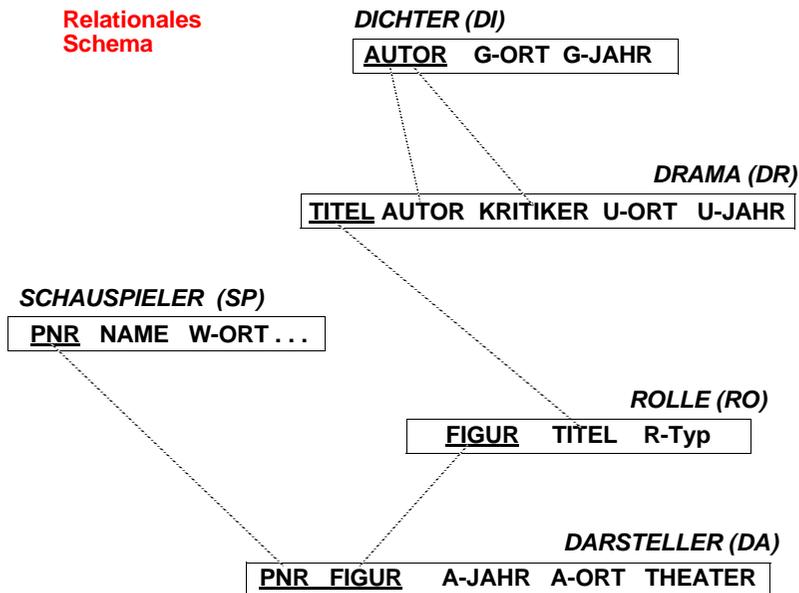
➔ „Connection Trap“ bei Projektion von Schlüsselteilen und nachfolgendem Verbund

## Beispiel-DB: BÜHNE

### ER-Diagramm



### Relationales Schema



## 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?

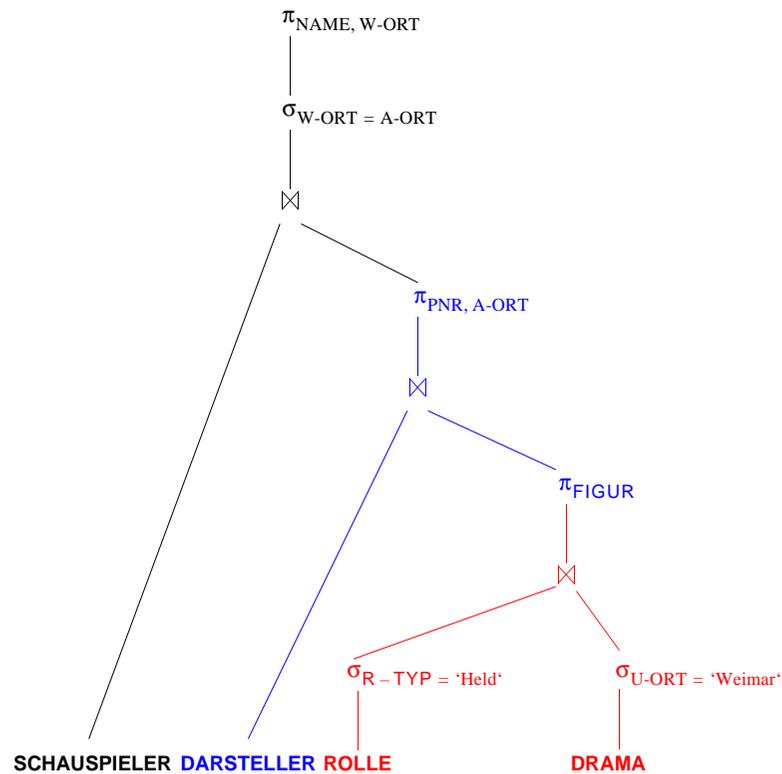
## Anfragedarstellung als Operatorbaum

## Anfragen (2)

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

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

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



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 Ausführungsreihenfolge (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:  $\pi, \sigma$
- binäre Operationen:  $\cap, \cup, -, \times, \bowtie, \div$

- Statistische Kenngrößen** werden dem DB-Katalog entnommen

- $N_i = \text{Card}(R_i)$
- $j_i =$  Anzahl der verschiedenen Werte eines Attributs  $A_i$

- Algebraische Optimierung - Beispiel:**

**Datenbank:**

ABT ( ANR, BUDGET, A-ORT )

PERS ( PNR, NAME, BERUF, GEHALT, ALTER, ANR )

PM ( PNR, JNR, DAUER, ANTEIL )

PROJ ( JNR, BEZEICHNUNG, SUMME, P-ORT )

## Rewrite-Regeln der Relationenalgebra<sup>5</sup>

### Äquivalenz von Ausdrücken

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

#### 1. Kommutatives Gesetz für Verbunde und Produkte

$$R_1 \bowtie R_2 \equiv R_2 \bowtie R_1$$

$$R_1 \bowtie R_2 \equiv R_2 \bowtie R_1$$

$$R_1 \times R_2 \equiv R_2 \times R_1$$

#### 2. Assoziatives Gesetz für Verbunde und Produkte

$$(R_1 \bowtie_{F_1} R_2) \bowtie_{F_2} R_3 \equiv R_1 \bowtie_{F_1} (R_2 \bowtie_{F_2} R_3)$$

$$(R_1 \times R_2) \times R_3 \equiv R_1 \times (R_2 \times R_3)$$

#### 3. Folgen von Projektionen

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

#### 4. Folgen von Selektionen

$$\sigma_{F_1}(\sigma_{F_2}(R)) = \sigma_{F_1 \wedge F_2}(R)$$

$$\text{Da } (F_1 \wedge F_2 = F_2 \wedge F_1):$$

$$\sigma_{F_1}(\sigma_{F_2}(R)) = \sigma_{F_2}(\sigma_{F_1}(R))$$

5. 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:

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

wenn F auch Attribute aus B<sub>1</sub> . . . B<sub>m</sub> enthält:

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

### 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

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

## Optimierung – Berechnungsgrundlagen

### • Allgemeine Annahmen

- Gleichverteilung der Attributwerte eines Attributes
- Stochastische Unabhängigkeit der Werte verschiedener Attribute

Mit Hilfe von statistischen Werten kann die Optimierungskomponente jedem Qualifikationsprädikat einen Selektivitätsfaktor ( $0 \leq SF \leq 1$ ) zuordnen (erwarteter Anteil an Tupeln, die das Prädikat erfüllen):  $Card(\sigma_p(R)) = SF(p) \cdot Card(R)$

### • Selektivitätsfaktor SF bei:

$$A_i = a_i \quad SF = \begin{cases} 1/j_i & \text{wenn Index auf } A_i \\ 1/10 & \text{sonst} \end{cases}$$

$$A_i = A_k \quad SF = \begin{cases} 1 / \text{Max}(j_i, j_k) & \text{wenn Index auf } A_i, A_k \\ 1 / j_i & \text{wenn Index auf } A_i \\ 1 / j_k & \text{wenn Index auf } A_k \\ 1/10 & \text{sonst} \end{cases}$$

$$A_i \geq a_i \quad (\text{oder } A_i > a_i) \quad SF = \begin{cases} (a_{\max} - a_i) / (a_{\max} - a_{\min}) & \text{wenn Index auf } A_i \\ & \text{und Wert interpolierbar} \\ 1/3 & \text{sonst} \end{cases}$$

$$A_i \text{ BETWEEN } a_i \text{ AND } a_k \quad SF = \begin{cases} (a_k - a_i) / (a_{\max} - a_{\min}) & \text{wenn Index auf } A_i \\ & \text{und Wert interpolierbar} \\ 1/4 & \text{sonst} \end{cases}$$

### • Berechnung von 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)

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

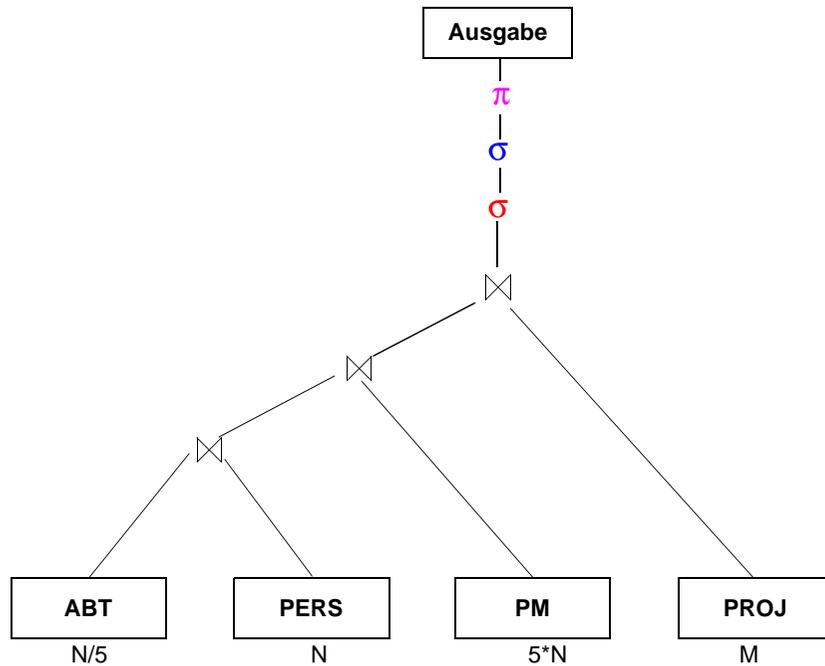
## Algebraische Optimierung – Beispiel

### Anfrage:

Finde Name und Beruf von Angestellten, deren Abteilung in KL ist und die in KL Projekte durchführen

$\pi_{\text{NAME, Beruf}}(\sigma_{\text{A-ORT}='KL'}(\sigma_{\text{P-ORT}='KL'}(\text{ABT} \bowtie \text{PERS} \bowtie \text{PM} \bowtie \text{PROJ})))$

### 1. Ausgangslösung



### • Annahmen:

Anzahl der Tupel in

ABT: N/5  
PERS: N  
PM: 5\*N  
PROJ: M

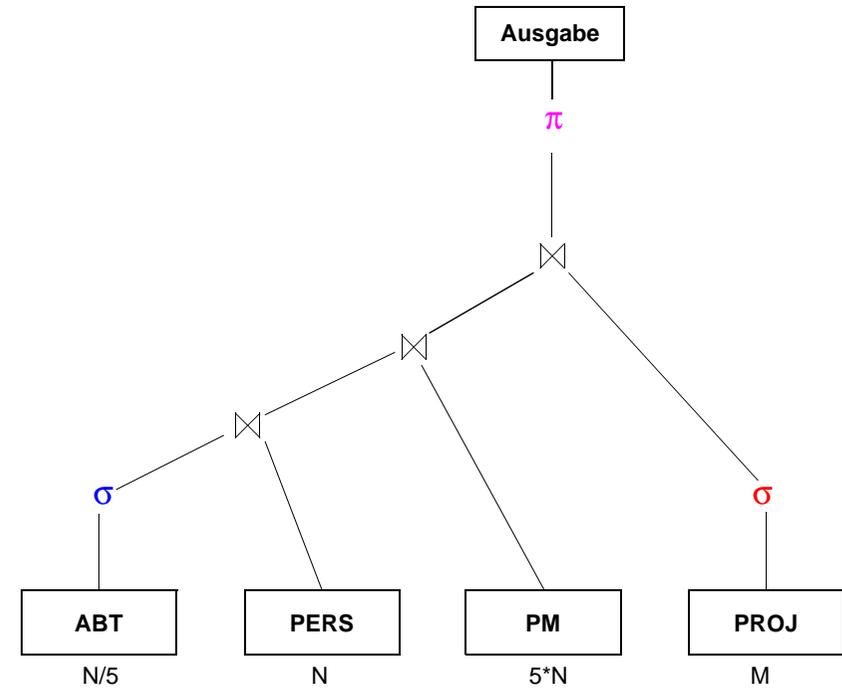
Anzahl der Attributwerte von

A-ORT: 10  
P-ORT: 100

Verlustfreie Verbunde über PS/FS

## Algebraische Optimierung – Beispiel (2)

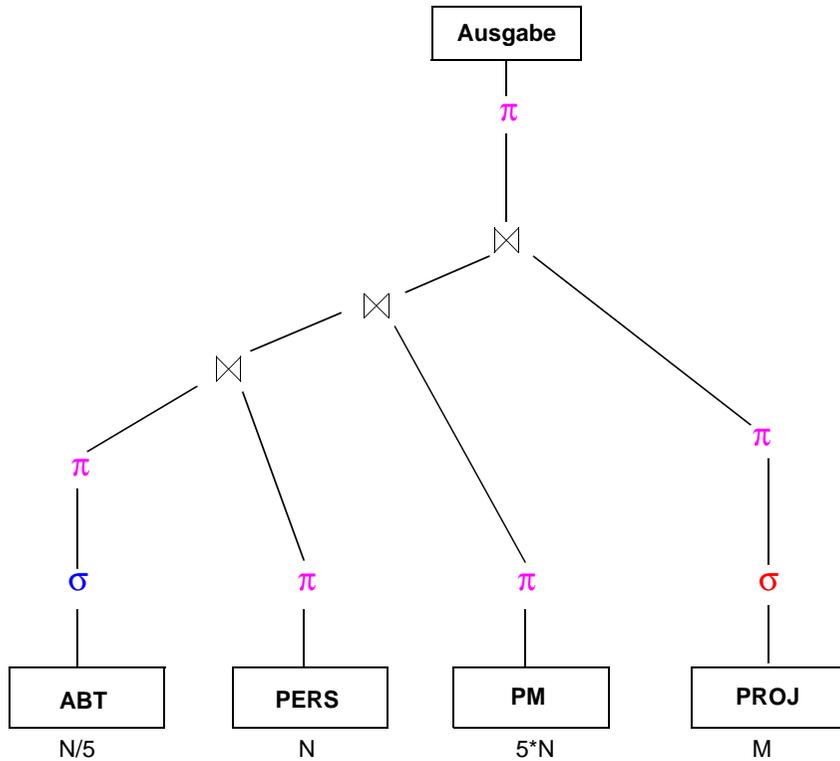
### 2. Verschieben der Selektion zu den Blattknoten



➔ 1. Führe Selektionen so früh wie möglich aus!

## Algebraische Optimierung – Beispiel (3)

### 3. Verschieben der Projektion



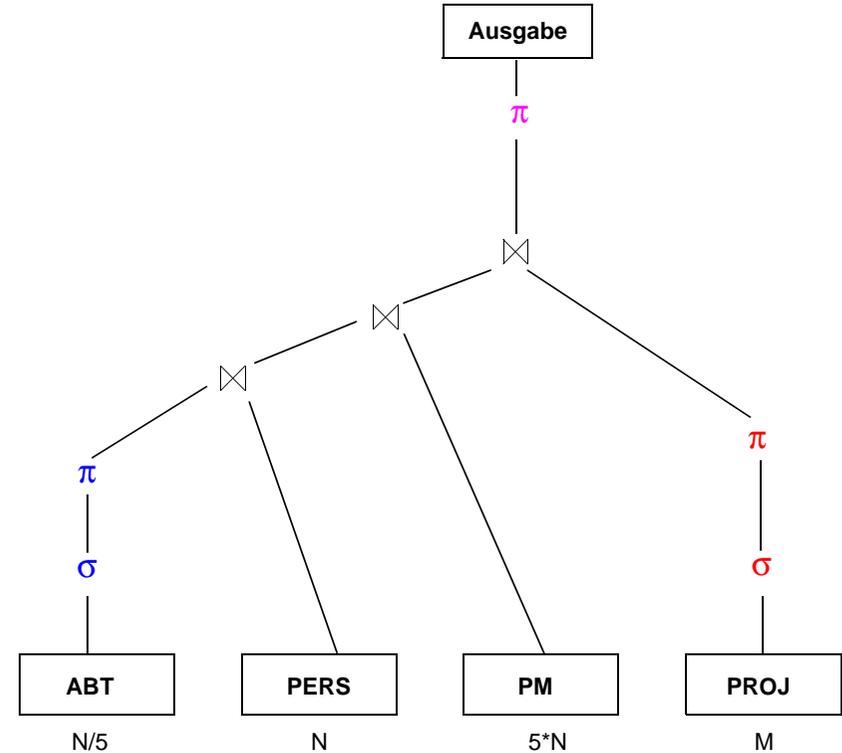
### ➔ II. Führe Projektion (ohne Duplikateliminierung) frühzeitig aus!

(Eliminierung von Duplikaten ist in der Regel sehr teuer)

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

## Algebraische Optimierung – Beispiel (4)

### 4. Optimierter Operatorbaum – Vorschlag



- Alternative Möglichkeit:

Zusammenfassen von

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

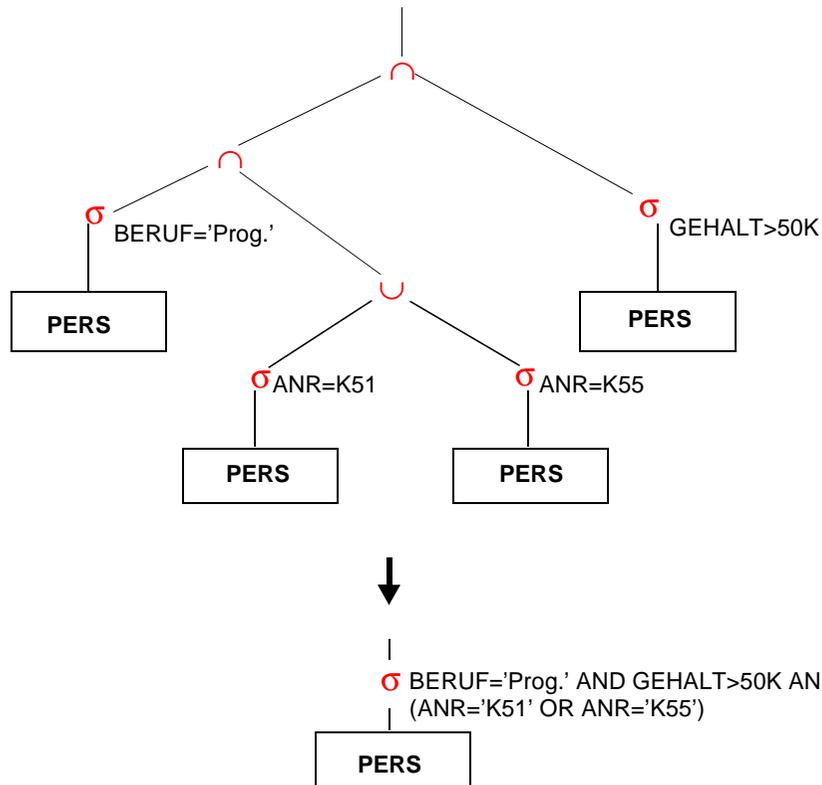
- ➔ III. Verknüpfe Folgen von unären Operationen wie Selektion und Projektion (wenn diese tupelweise abgewickelt werden können!)

## Algebraische Optimierung – Beispiel (5)

### 5. Weitere Optimierungsmaßnahmen

#### Ausdrucksauswertung:

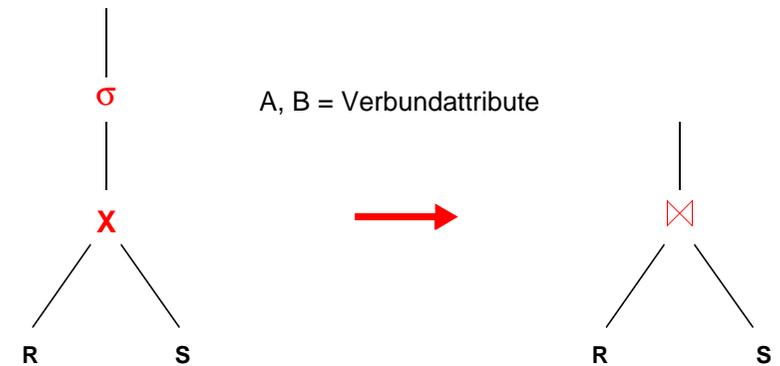
Alle Programmierer mit mehr als 50K Gehalt  
und zwar aus Abteilungen K51 oder K55



➔ **IV. Fasse einfache Selektionen auf einer Relation zusammen!**

## Algebraische Optimierung – Beispiel (6)

### 6. Kartesisches Produkt mit Selektion



A, B = Verbundattribute

Leistungsbetrachtung bei:

- a) verlustfreier natürlicher Verbund
- b)  $\Theta$ -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)

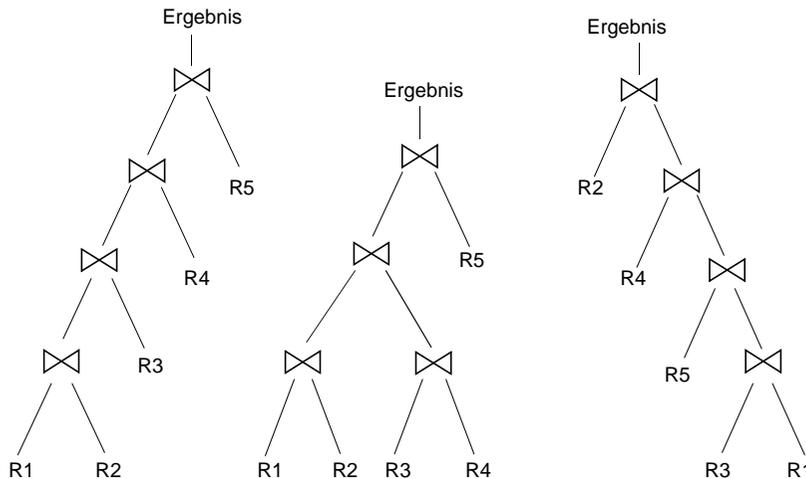
## Algebraische Optimierung – Beispiel (7)

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

1.  $T = R3 \bowtie (R1 \bowtie R2)$
2.  $T = R2 \bowtie (R1 \bowtie R3)$
3.  $T = R1 \bowtie (R2 \bowtie R3)$

- **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 mit  $j$  Werten des Verbundattributs und  $(N(R1) \leq N(R2))$**

$$N(T1) = N(R1) * N(R2)/j$$

beim kartesischen Produkt: analog zu  $j = 1$

$$\Rightarrow N(T1) = N(R1) * N(R2)$$

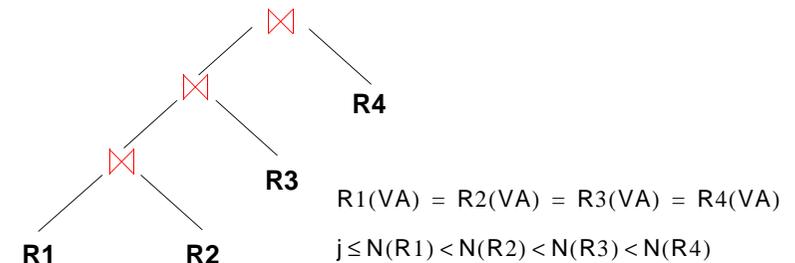
beim (1:n)-Verbund:  $N(R1) = j$

$$\Rightarrow N(T1) = N(R2)$$

beim (n:m)-Verbund:  $j < N(R1)$

$$\Rightarrow N(T1) > N(R2)$$

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



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

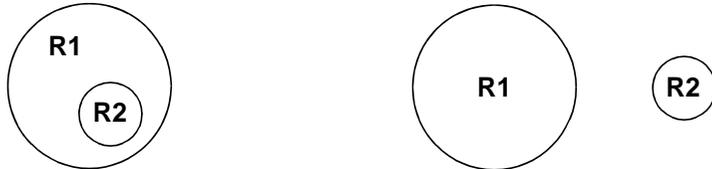
- **Dynamische Entscheidung aufwendiger, aber genauer**

Bei jedem Auswertungsschritt werden die momentan kleinsten (Zwischen-) Relationen ausgewählt

## Algebraische Optimierung – Beispiel (9)

### 8. Reihenfolgen von Mengenoperationen

#### • Kardinalität der Vereinigung



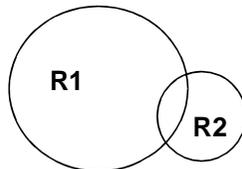
$$\text{MAX}(N(R1), N(R2)) \leq N(R1 \cup R2) \leq N(R1) + N(R2)$$

#### • Kardinalität des Durchschnitts



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

Erwartung:



#### • Heuristische Regel:

➔ VIII. Verknüpfte 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 allen 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:

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

- Beschreibung der Division mit den Grundoperationen**

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

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

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

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

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

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

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

## Division – Beispiel

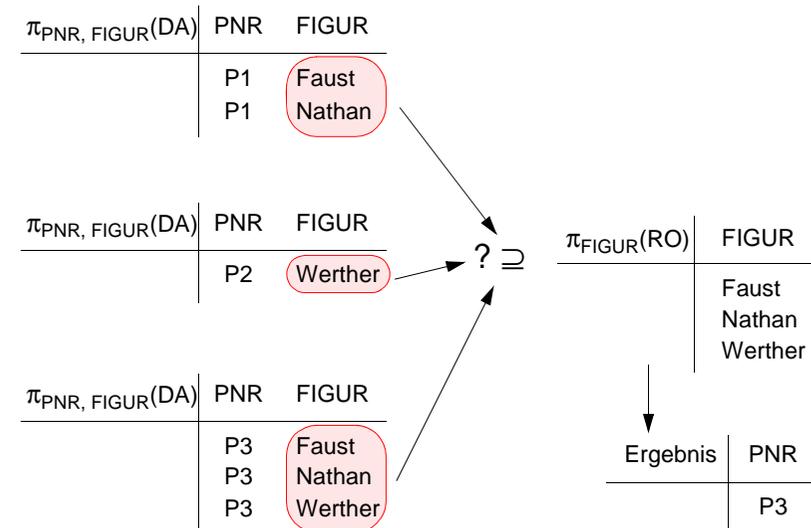
- Datenbank**

| DA | PNR | FIGUR   | A-Jahr ... | RO | FIGUR   | TITEL                | R-Typ |
|----|-----|---------|------------|----|---------|----------------------|-------|
|    | P1  | Faust   | 1999       |    | Faust   | Faust                |       |
|    | P1  | Nathan  | 1998       |    | Nathan  | Nathan der Weise ... |       |
|    | P2  | Werther | 1997       |    | Werther | Die Leiden ...       |       |
|    | P3  | Faust   | 1998       |    |         |                      |       |
|    | P3  | Nathan  | 1999       |    |         |                      |       |
|    | P3  | Werther | 1998       |    |         |                      |       |

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

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

- Auswertung:**



### 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  $\Theta$  des  $\Theta$ -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:

$$V = R \bowtie_{ij} S = \sigma_{ij}(R \times S)$$

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

- 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:

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

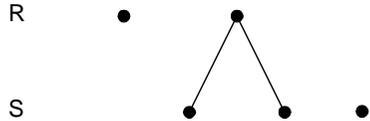
| PERS | PNR | ALTER | PERS' | PNR' | ALTER' ... |
|------|-----|-------|-------|------|------------|
|      | P1  | 25    |       | P1   | 25         |
|      | P2  | 23    |       | P2   | 23         |
|      | P3  | 28    |       | P3   | 28         |

| G | (PNR | ALTER | PNR' | ALTER') |
|---|------|-------|------|---------|
|   |      |       |      |         |

## Äußerer Verbund (Outer Join)

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

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



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

### Beispiel

| SP (PNR, NAME, ...) |   | DA (PNR, FIGUR, ...) |   |
|---------------------|---|----------------------|---|
| P1                  | x | P1                   | F |
|                     |   | P1                   | W |
| P2                  | y | P3                   | M |

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

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

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

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

## Äußerer Verbund (2)

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

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

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

### Äußerer Gleichverbund

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

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

### Äußerer natürlicher Verbund

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

- **Linker äußerer Gleichverbund**

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

### Linker äußerer Gleichverbund

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

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

- **Rechter äußerer Gleichverbund**

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

### Rechter äußerer Gleichverbund

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

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

## Beispiele zum äußeren Gleichverbund

- Gleichverbund

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

- Linker äußerer Gleichverbund

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

- Rechter äußerer Gleichverbund

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

- Äußerer Gleichverbund

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

## 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 |  | HIWI | MATNR | FBNR | JOB   |
|         | 123   | FB5  | 5   |  |      | 456   | FB5  | Tutor |
|         | 789   | FB9  | 9   |  |      | 987   | FB9  | Prog. |

### OUTER UNION

|               |       |      |     |       |
|---------------|-------|------|-----|-------|
| STUD-<br>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

## Zusammenfassung: Abbildungskonzepte

- **Datenstruktur**

Relation (Tabelle)

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |

- ➔ **einzig**e 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

