Informations-
systeme

4. Grundlagen des Relationenmodells*

Stefan Deßloch

* *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.

Technische Universität Kaiserslautern

Informations-
systeme

- Übersicht
- Abbildung ERM → RM
- Relationen-algebra
- Optimierung
- Weitere Operationen

Gliederung

- 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
- Relationenalgebra – weitere Operationen
  - Division
  - Intervallverbund
  - Äusserer Verbund

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)

  mengenorientiert!
  abgeschlossen!

  ➥ 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

Informations-
systeme

Übersicht
Abbildung ERM → RM
Relationen-
algebra
Optimierung
Weitere
Operationen

Relationenmodell* und ER-Modell

ERM	**RM**
Einwertiges Attribut Definitionsbereich Primärschlüssel	wie im ERM
Zusammengesetztes Attribut Mehrwertiges Attribut	nur als unabhängige einwertige Attribute

Entity-Menge → Relation

Relationship-Menge → Relation

* 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.

Informations-
systeme

Übersicht

Abbildung ERM → RM

Relationen-
algebra

Optimierung

Weitere Operationen

Relationenmodell – Grundkonzepte

- Definition: Normalisierte Relation

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

D_i D_j D_k

Untermenge des kartesischen Produktes

- Darstellungsmöglichkeit für R: n-spaltige Tabelle
  - ➥ Jede **Relation** kann als Tabelle dargestellt werden

- Relation ist eine Menge: Garantie der Eindeutigkeit der Zeilen/Tupel
  - ➥ **Primärschlüssel** (und ggf. mehrere Schlüsselkandidaten)

Informations-
systeme

Übersicht

Abbildung ERM → RM

Relationen-
algebra

Optimierung

Weitere Operationen

Normalisierte Relationen in Tabellendarstellung

- 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

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

Beispiel-Relationen in Tabellendarstellung

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

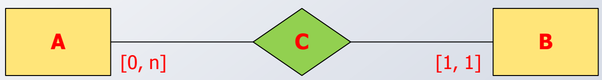
PROF	<u>PNR</u>	PNAME	FBNR	FACHGEBIET
	1234	Härder	FB 5	Datenbanksysteme
	5678	Wedekind	FB 9	Informationssysteme
	4711	Müller	FB 9	Operations Research
	6780	Nehmer	FB 5	Betriebssysteme
	2223	Richter	FB 5	Expertensysteme

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!

Relationenmodell – Grundkonzepte

- 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.

FB

FBNR	FBNAME	DEKAN
FB 9	Wirtschaftswissenschaften	4711
FB 5	Informatik	2223

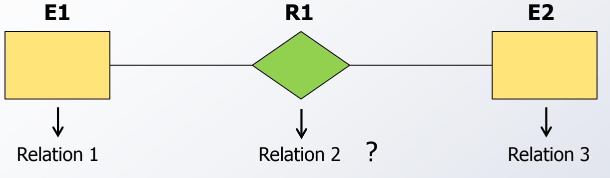
PROF

PNR	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

Fremdschlüssel - Bemerkungen

1. Fremdschlüssel und zugehöriger Primärschlüssel (Schlüsselkandidat) tragen wichtige interrelationale (manchmal auch intrarelationale) Informationen. Sie sind auf dem gleichen Wertebereich definiert (vergleichbar und vereinigungs-verträglich). 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

Informations-systeme

Übersicht

Abbildung ERM → RM

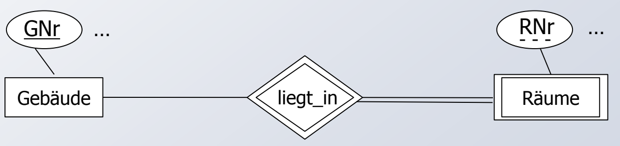
Relationen-algebra

Optimierung

Weitere Operationen

(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
    - Mengenwertige 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 existenz-abhängiger Entity-Typ ist und einen partiellen Schlüssel definiert



Darstellung der Entity-Typen im RM:

GEBÄUDE(GNR), ...) **Räume**(GNR, RNR, ...)

Informations-systeme

Übersicht

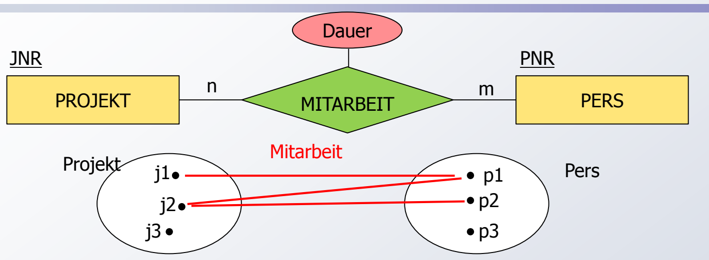
Abbildung ERM → RM

Relationen-algebra

Optimierung

Weitere Operationen

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



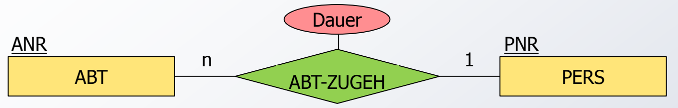
Verwendung von drei Relationen erforderlich:

PROJEKT (JNR, BEZEICH, ...)	PERS (PNR, PNAME, ...)	MITARBEIT (JNR, PNR, DAUER)
j1	p1	j1, p1 ...
j2	p2	j2, p1 ...
j3	p3	j2, p2 ...

- Regel:

  Die Relationship-Menge wird auf eine Relation abgebildet, wobei der Primär-schlüssel sich aus den Primärschlüsseln der beteiligten Entity-Mengen zu-sammensetzt, 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

ABT(<u>ANR</u>, ANAME, ...)
a1
a2

PERS (<u>PNR</u>, PNAME, ...)
p1
p2

ABT-ZUGEH (ANR, PNR, DAUER)
a1 p1
a1 p2

zwei Relationen

ABT(<u>ANR</u>, ANAME, ...)
a1
a2

PERS (<u>PNR</u>, PNAME, ..., ANR, DAUER)
p1, ... a1
p2, ... a1

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.

Informations-
systeme

Übersicht

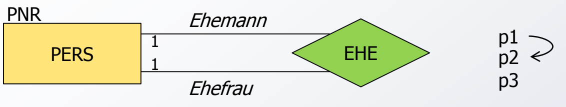
Abbildung ERM → RM

Relationen-algebra

Optimierung

Weitere Operationen

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



1.) Verwendung von zwei Relationen

PERS (<u>PNR</u>, PNAME, ...)
p1
p2
p3

EHE (PNR, GATTE, ...)
p2 p1

2. Verwendung von einer Relation

PERS (<u>PNR</u>, PNAME, ..., GATTE)
p1 -
p2 p1
p3 -

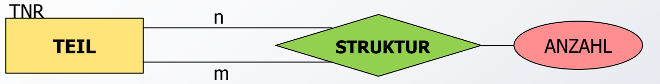
Vorsicht!

Durch das Relationenmodell wird hier nur 1:n-Semantik garantiert!

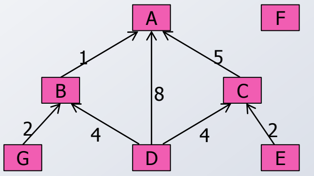
- Regel:

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

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



Stückliste



Darstellungsmöglichkeit im RM:

TEIL (<u>TNR</u>, BEZ, MAT, BESTAND)

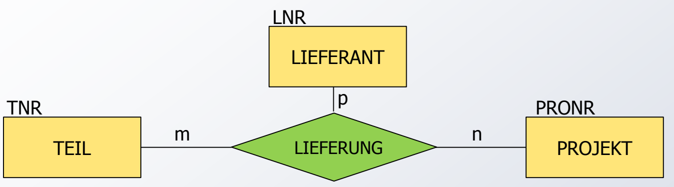
STRUKTUR (<u>OTNR</u>, <u>UTNR</u>, ANZAHL)

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 (<u>LNR</u>, LNAME, LORT, ...)

PROJEKT (<u>PRONR</u>, PRONAME, PORT, ...)

TEIL (<u>TNR</u>, TBEZ, GEWICHT, ...)

LIEFERUNG (<u>LNR, PRONR, TNR</u>, ANZAHL, DATUM)

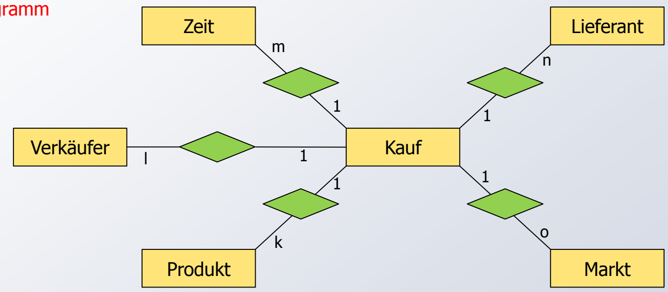
Informations-
systeme

Übersicht
Abbildung ERM → RM
Relationen-algebra
Optimierung
Weitere Operationen

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)

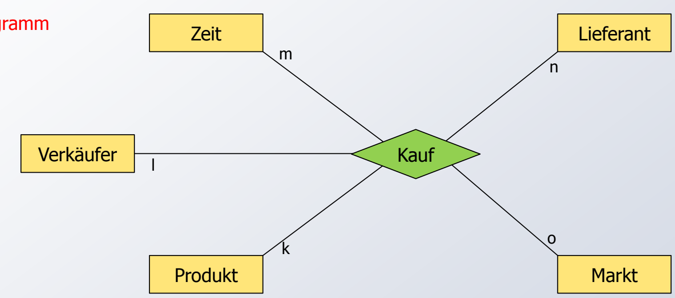
Informations-
systeme

Übersicht
Abbildung ERM → RM
Relationen-algebra
Optimierung
Weitere Operationen

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

- Data Warehousing – Kauf als 5-stellige Beziehung

ER-Diagramm

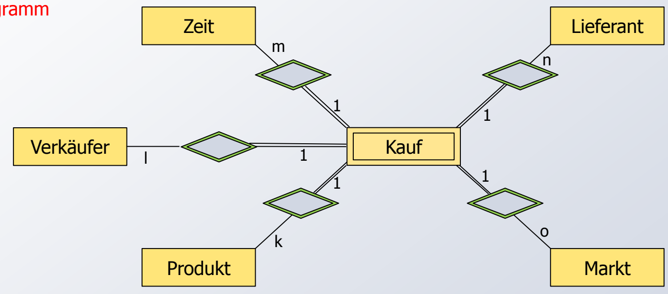


Produkt	(Pnr, Bezeichnung, …
Verkäufer	(Vnr, VName, …
Zeit	(Znr, Datum, Besonderheit, …)
Lieferant	(Lnr, LName, Ort, …)
Markt	(Mnr, Adresse, …)
Kauf	(Pnr, Vnr, Znr, Lnr, Mnr, Menge, Preis)

Beispiel – Data Warehousing (2)

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

ER-Diagramm

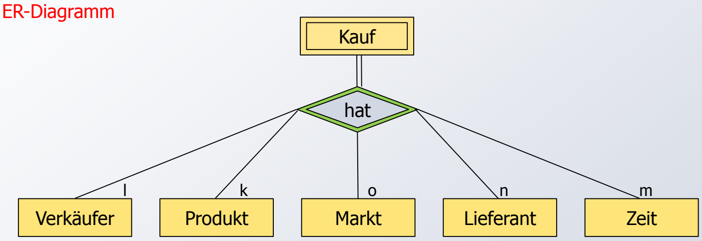


Produkt (Pnr, Bezeichnung, …
Verkäufer (Vnr, VName, …
Zeit (Znr, Datum, Besonderheit, …)
Lieferant (Lnr, LName, Ort, …)
Markt (Mnr, Adresse, …)
Kauf (Pnr, Vnr, Znr, Lnr, Mnr, Menge, Preis)

Beispiel – Data Warehousing

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

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

  ➥ ist eigentlich Aufgabe eines Sichtkonzeptes

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

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

  ➥ ist Aufgabe des internen Schemas und des Sichtkonzeptes

Informations-
systeme

Übersicht
Abbildung ERM → RM
Relationen-algebra
Optimierung
Weitere Operationen

Abbildungstypen innerhalb einer Entity-Menge (2)

- Abbildung mehrwertiger Attribute
  - Entity-Menge:

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

Darstellungsmöglichkeit im RM:

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

p1,	Müller
p2,	Schulz
...	

LIEBLINGSESSEN(<u>PNR, GERICHT,</u>)

p1,	Schnitzel
p1,	Braten
p1,	Rollmops
p2,	Pizza
.....	

KINDER (<u>PNR, VORNAME</u>, ALTER)

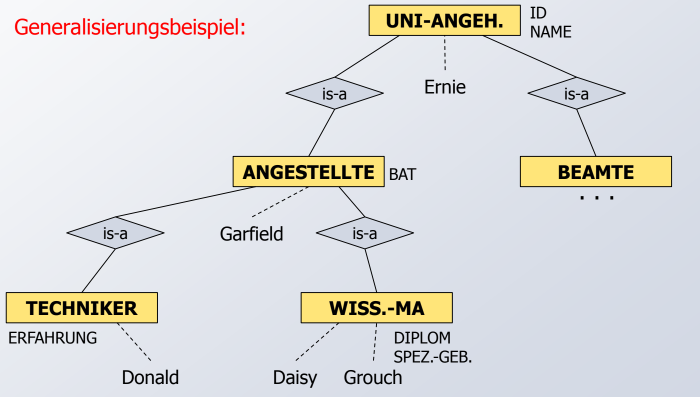
p2,	Nadine,	5
p2,	Philip,	2

Natürliche Abbildung?
Äquivalenz zu ER-Schema?

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:



Informations-
systeme

Übersicht

Abbildung ERM → RM

Relationen-
algebra

Optimierung

Weitere Operationen

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
	007	Informatik	Recovery	Daisy	IIa
	765	Mathematik	ERM	Grouch	IIa

- Eigenschaften:
  - niedrige Speicherungskosten und keine Änderungsanomalien
  - Retrieval kann rekursives Suchen in Unterklassen erfordern
  - explizite Rekonstruktion durch Relationenoperationen
- ➥ Beispiel: Finde alle ANGESTELLTE:
  TECHNIKER ∪ WISS.-MA ∪ ANGESTELLTE, beschränkt auf ID, NAME, BAT

Generalisierung – relationale Sicht (2)

- **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
	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.
	007	Informatik	Recovery
	765	Mathematik	ERM

- Eigenschaften
  - geringfügig erhöhte Speicherungskosten, aber hohe Aufsuch- und Aktualisierungs-kosten
  - Integritätsbedingungen: TECHNIKER.ID ⊆ 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 (3)

- **Lösung 3:** Volle Redundanz
  - Eine Instanz wird wiederholt in jeder Klasse, zu der sie gehört, gespeichert
  - Sie besitzt dabei die Werte der Attribute, die sie geerbt hat, zusammen mit den Werten der Attribute der Klasse

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

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

TECHNIKER	ID	NAME	BAT	ERFAHRUNG
	123	Donald	IVa	SUN

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

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

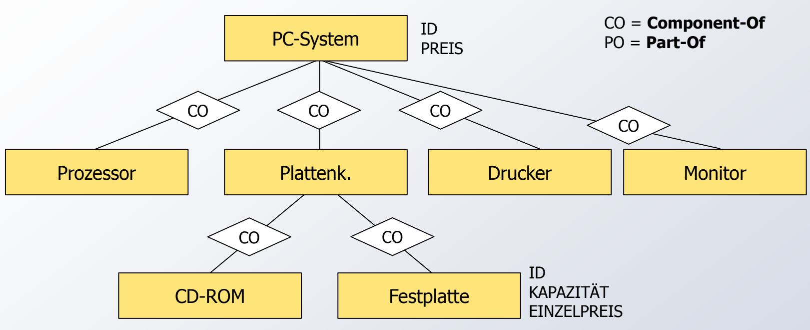
Generalisierung – relationale Sicht (4)

- **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	Angestellte	Garfield	Ia	-	-	-
111	Uni-Angeh.	Ernie	-	-	-	-
123	Techniker	Donald	Iva	SUN	-	-
333	Wiss.-MA	Daisy	IIa	-	Informatik	Recovery
765	Wiss.-MA	Grouch	IIa	-	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

ID	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

Informations-systeme

Übersicht

Abbildung ERM → RM

Relationen-algebra

Optimierung

Weitere Operationen

Aggregation – relationale Sicht

Plattenk.	ID	CD-ROM	Festplatte
	PK1	C1	F1
	PK2	C2	F2
	PK3	C3	F3

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

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

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

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

Drucker	ID	Hersteller	Typ	Preis
	Tintenstrahldrucker	HP	Desk Jet 5150	100,--
	Laserdrucker	HP	Laser 1300	360,--

➥ Eigenschaften der Aggregation werden durch relationale Operationen nicht unterstützt

Informations-systeme

Übersicht

Abbildung ERM → RM

Relationen-algebra

Optimierung

Weitere Operationen

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*:
  - Relationenalgebra (z. B. ISBL)
  - Relationenkalkül (z. B. Alpha)
  - Abbildungsorientierte Sprachen (z. B. SQL)
  - Graphikorientierte Sprachen (z. B. Query-by-Example)

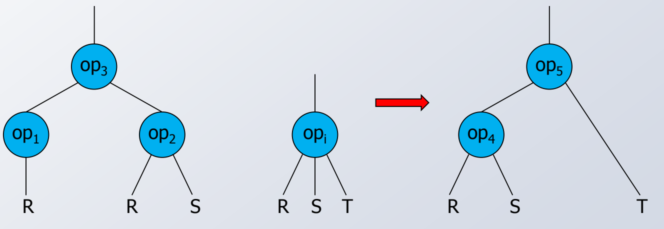
(*) The relational model separates data from the details of its physical storage so users and applications don't have to know where to look for the data they need.

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



Informations-
systeme

Übersicht
Abbildung ERM → RM
Relationen-algebra
Optimierung
Weitere Operationen

Relationenalgebra – Überblick (2)

- Klassische Mengenoperationen:
  - Vereinigung, Differenz, kartesisches Produkt
  - ableitbar: Durchschnitt
- Relationenoperationen:
  - Projektion, Restriktion (Selektion)
  - ableitbar: Verbund (Join), Division
  - ➥ Auswahlvermögen entspricht Prädikatenkalkül erster Ordnung („relational vollständig“)

© 2014 LG IS

Informations-
systeme

Übersicht
Abbildung ERM → RM
Relationen-algebra
Optimierung
Weitere Operationen

Selektion (Restriktion)

- Auswahl von Zeilen einer Relation über Prädikate, abgekürzt σ_P

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

a) Operanden — Konstanten, Attributnamen

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

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

- Definition:

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

P(t): Ersetze in der log. Formel P alle Attributnamen durch entsprechende Werte aus t.

Informations-
systeme

Übersicht
Abbildung ERM → RM
Relationen-algebra
Optimierung
Weitere Operationen

Selektion - Beipiele

- Beispiele:

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

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

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

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

- Ergebnis:

ERG	PNR	NAME	ALTER	GEHALT	ANR	MNR
	406	Coy	47	50 700	K55	123

Projektion

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

$$\boxed{\pi_A(R) = \{p \mid \exists\, t \in R : p = < t\,[A_1], \ldots, t\,[A_k] > \}}$$

! Duplikate entfernt !

- Alternativ: Benutzung der Spaltennummern j_i: $\pi_{j_1, j_2, \ldots, j_k}(R)$
- Beispiel: $\pi_{\text{NAME, GEHALT, ALTER}}(\text{PERS})$

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

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

ERG	ANR	MNR
	K55	123
	K51	-
	K53	777

Relationenalgebra – Beispiel-DB

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

1. Finde alle Abteilungsorte

   $T = \pi_{\text{AORT}}(\text{ABT})$

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

   $T = \pi_{\text{PNR, NAME}}(\sigma_{\text{ANR = K55 \& GEHALT > 40.000}}(\text{PERS}))$

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

$(A_1, A_2, \ldots, A_n)$ $(B_1, B_2, \ldots, B_n)$

D_i D_j $\ldots$ D_k

➥ $W(A_i) = W(B_i)$: $i = 1, n$

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

4. Symmetrische Differenz
   $R \Delta S = (R \cup S) - (R \cap S)$
   $= ((R \cup S) - (R - (R - S)))$
   $= \{t | t \in R \oplus t \in S\}$

(Erweitertes) Kartesisches Produkt

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

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

  $k = x|y = <x_1, \ldots, x_r, y_1, \ldots, y_s>$
  nicht $<<x_1, \ldots, x_r>, <y_1, \ldots, y_s>>$ wie übliches kartesisches Produkt

- Anwendung von: ABT x PERS

ABT	ANR	ANAME	AORT
	K51	Planung	KL
	K53	Einkauf	F
	K55	Vertrieb	F

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

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

➥ Iterative Schleifenmethode (Nested-Loop-Algorithmus)

Informations-
systeme

Übersicht

Abbildung ERM → RM

Relationen-algebra

Optimierung

Weitere Operationen

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
    - $\pi_{ABT.ANR,\ ANAME,\ PNR}\ (\sigma_{ABT.ANR\ =\ PERS.ANR}\ (ABT\ \mathbf{X}\ PERS))$

- Explizite Umbenennung durch Rename-Operator (ρ)
  - Umbenennung eines Attributs
    - $\pi_{ANR,\ ANAME,\ PNR}\ (\sigma_{ANR\ =\ ABTNR}\ (ABT\ \mathbf{X}\ \rho_{ABTNR \leftarrow ANR}(PERS)))$
  - Umbenennung von Relationen durch Angabe des neuen Relationennamens
  - Beispiel: Ermittle Paare von Personen, die in der gleichen Abteilung arbeiten
    - $\sigma_{P1.ANR\ =\ P2.ANR}\ (\rho_{P1}(PERS)\ \mathbf{X}\ \rho_{P2}(PERS))$

Informations-
systeme

Übersicht

Abbildung ERM → RM

Relationen-algebra

Optimierung

Weitere Operationen

Verbund (Join, Θ-Join)

- Grob:
  - Kartesisches Produkt zwischen zwei Relationen R (Grad r) und S (Grad s) eingeschränkt durch Q-Bedingungen zwischen i-Spalte von R und j-Spalte von S.
- Sei $\Theta \in \{<, =, >, \leq, \neq, \geq\}$ (arithmetischer Vergleichsoperator)

Θ-Verbund zwischen R und S:

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

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

- Bemerkungen:

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

(2) statt i und j: Attributnamen A und B, z. B.:

$$R \underset{A\,\Theta\,B}{\bowtie} S$$

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

Gleichverbund - Beispiel

- Anwendung von $ABT \underset{ANR = ANR}{\bowtie} PERS$

Fremdschlüssel

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

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

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

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

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

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

➥ π als Umkehroperation führt nicht auf ABT oder PERS

Informations-
systeme

Übersicht
Abbildung ERM → RM
Relationen-algebra
Optimierung
Weitere Operationen

Natürlicher Verbund (Natural Join)

- grob:
  Gleichverbund über <u>alle</u> gleichen Attribute und
  Projektion über die verschiedenen Attribute
- gegeben: $R(A_1, A_2, \ldots, A_{r-j+1}, \ldots, A_r)$
  $S(B_1, B_2, \ldots, B_j, \ldots, B_s)$

  o.B.d.A.: (sonst. Umsortierung)
  $B_1 = A_{r-j+1}$
  $B_2 = A_{r-j+2}$
  $\vdots$
  $B_j = A_r$

Natürlicher Verbund zwischen R und S:

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

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

- Bemerkung:
  Attribute sind durch Übereinstimmungsbedingung gegeben
  (nicht durch PK/FK-Paare!)

Informations-
systeme

Übersicht

Abbildung ERM → RM

Relationen-algebra

Optimierung

Weitere Operationen

Natürlicher Verbund - Beispiel

- Anwendung von: **ABT ⋈ PERS**

ABT	ANR	ANAME	AORT
	K51	Planung	KL
	K53	Einkauf	F
	K55	Vertrieb	F

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

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

➥ verlustfreier Gleichverbund: $\pi_{\text{ANR, ANAME, AORT}}(\text{AP})=\text{ABT}$

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

- Verlustbehafteter natürlicher Verbund analog zu Gleichverbund

Informations-
systeme

Übersicht

Abbildung ERM → RM

Relationen-algebra

Optimierung

Weitere Operationen

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

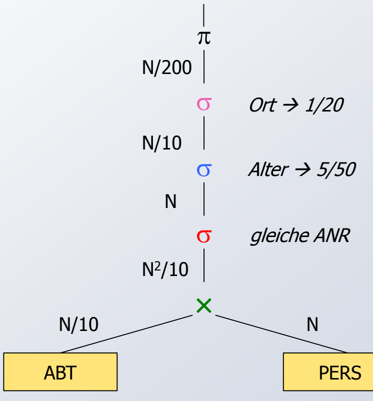
- Annahmen:
  - ABT: N/10 Tupel
  - PERS: N Tupel
  - Gleichverteilung der Attributwerte
  - AORT: 20 Werte
  - ALTER: 50 Werte (16-65)
  - Stochastische Unabhängigkeit der Werte verschiedener Attribute
  - Verlustfreie Verbunde von R1 und R2 über Primär-/Fremdschlüssel, mit Card(R1) < Card(R2): Card(R1 ⋈ R2) = Card(R2)
- Anfrage:
  Finde alle Angestellten (PNR, ALTER, ANAME), die in einer Abteilung in Frankfurt arbeiten und zwischen 30 und 34 Jahre alt sind.

Natürlicher Verbund - Lösungen

- Lösung 1:

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

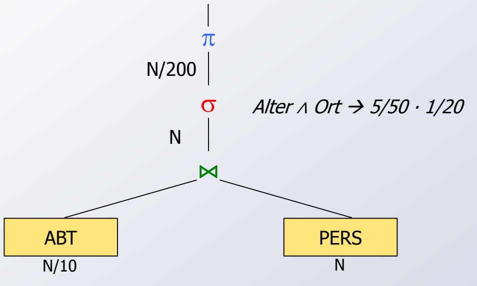
- Zugehöriger Operatorbaum



Natürlicher Verbund – Lösungen (2)

- Lösung 2:

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

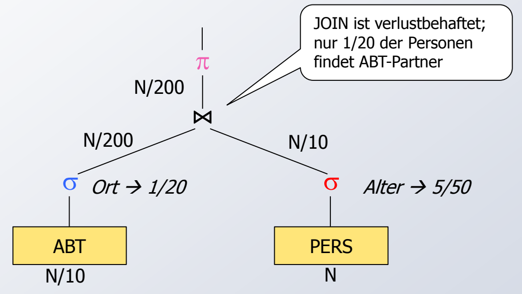


Natürlicher Verbund – Lösungen (3)

- Lösung 3:

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

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

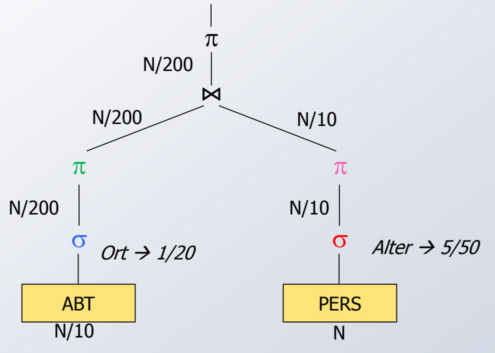


Natürlicher Verbund – Lösungen (4)

- Lösung 4:

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

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



➪ kleiner Tupel durch frühe Projektion

Natürlicher Verbund – Beispiel (3)

- Ist der Verbund ⋈ immer Umkehroperation zur Projektion (π)?
- Beispiel 1 (1:n): (AP = ABT ⋈ PERS)

AP1 = $\pi_{ANR,ANAME,AORT}$(AP) **AP2** = $\pi_{PNR,ALTER,ANR}$(AP)

AP3 = AP1 ⋈ AP2 $\stackrel{?}{=}$ (AP)

- Beispiel 2 (n:m):

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

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

DA1	PNR	A-ORT
	P1	MA
	P1	KL
	P2	MA

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

DA2	FIGUR	A-ORT
	Faust	MA
	Mephisto	KL
	Wallenst.	MA

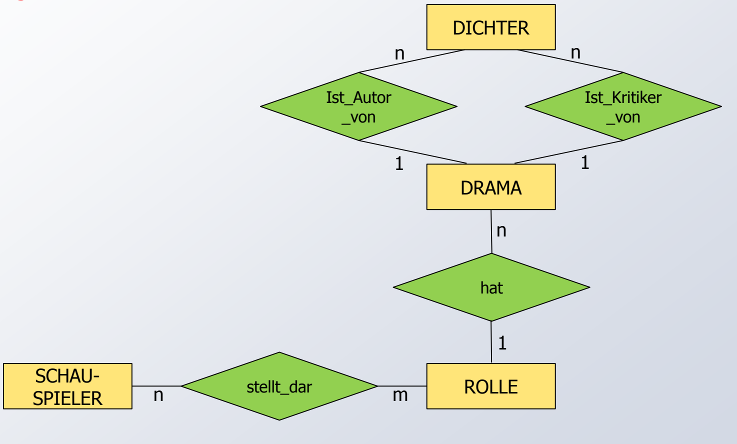
DA3 = (DA1 ⋈ DA2)

DA3	(PNR,	FIGUR,	A-ORT)
	P1	Faust	MA
	P1	Wallenst.	MA
	P1	Mephisto	KL
	P2	Faust	MA
	P2	Wallenst.	MA

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

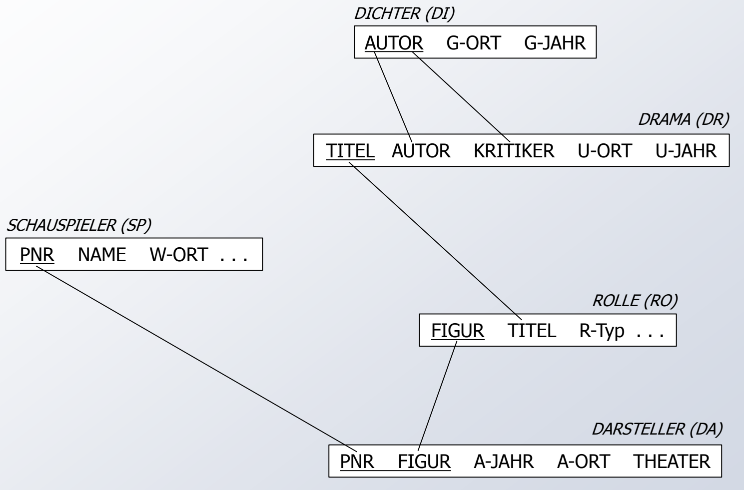
Beispiel-DB: BÜHNE

ER-Diagramm



Beispiel-DB: BÜHNE (2)

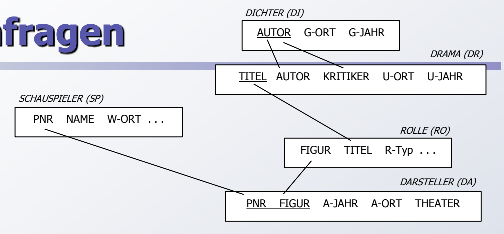
Relationales Schema



Informations-
systeme

Übersicht
Abbildung ERM → RM
Relationen-algebra
Optimierung
Weitere Operationen

Beispiel-DB: Anfragen



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

$$\pi_{NAME} (\sigma_{FIGUR=Faust} (SP \bowtie DA))$$

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

$$\pi_{NAME} (\sigma_{TITEL=Faust} (SP \bowtie DA \bowtie RO))$$

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

$$\pi_{NAME} (SP \bowtie DA \bowtie RO \bowtie (\sigma_{AUTOR=Schiller} DR))$$

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

$$SP \bowtie_{SP.NAME=DI.AUTOR} DI$$

$$\sigma_{SP.NAME=DR.AUTOR} (SP \bowtie DA \bowtie RO \bowtie DR)$$

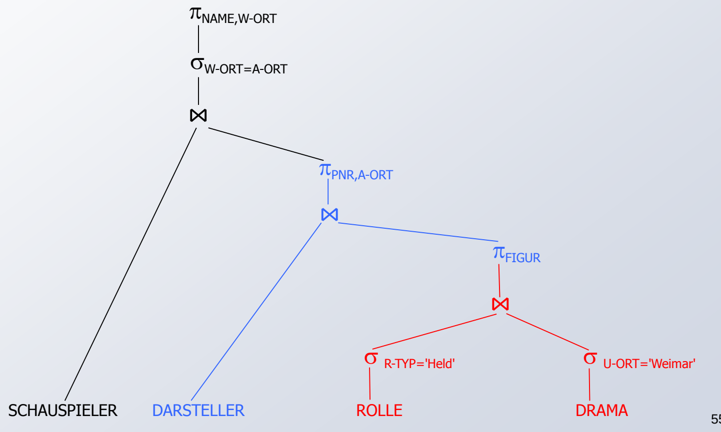
Anfragedarstellung als Operatorbaum

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

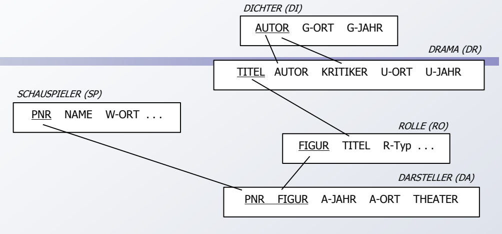
Q5= $\pi_{NAME,W\text{-}ORT}\ (\sigma_{W\text{-}ORT=A\text{-}ORT}(\text{SCHAUSPIELER} \bowtie ($

$\pi_{PNR,A\text{-}ORT}(\text{DARSTELLER} \bowtie (\pi_{FIGUR}($

$(\sigma_{R\text{-}TYP='Held'}\text{ROLLE}) \bowtie (\sigma_{U\text{-}ORT='Weimar'}\text{DRAMA}))))))$



Anfragen (2)



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

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

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

$\pi_{TITEL,\ KRITIKER}\ ((\sigma_{G\text{-}ORT=Weimar}\ DI\ )\ \underset{DI.AUTOR=DR.KRITIKER}{\bowtie}\ DR)$

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

$\pi_{PNR}\ SP\ -\ \pi_{PNR}\ DA$

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

$\pi_{PNR}\ (\sigma_{FIGUR=Faust\ \vee\ FIGUR=Wallenstein}\ DA)\ \ -\ \pi_{PNR}\ (\sigma_{FIGUR\neq Faust\ \wedge\ FIGUR\neq Wallenstein}\ DA)$

Relationenalgebra - Optimierung

- Relationenalgebraische Formulierungen spezifizieren Ausführungs-reihenfolge (prozedurale Elemente)
  - ➥ jedoch äquivalente Umformungen möglich

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

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

Relationenalgebra – Optimierung (2)

- Statistische Kenngrößen werden dem DB-Katalog entnommen
  - N_i = 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)

Anfrage:

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

Rewrite-Regeln der Relationenalgebra*

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

1. Kommutatives Gesetz für Verbunde und Produkte

$R1 \bowtie_F R2 \equiv R2 \bowtie_F R1$

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

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

2. Assoziatives Gesetz für Verbunde und Produkte

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

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

3. Folgen von Projektionen

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

4. Folgen von Selektionen

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

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

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

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

Rewrite-Regeln der Relationenalgebra (2)

5. Vertauschung von Selektionen und Projektionen

F enthält nur Attribute aus A . . . Z:

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

wenn F auch Attribute aus $B_1 \ldots B_m$ enthält:

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

6. Vertauschung von Selektion und Kartesischem Produkt

F enthält nur Attribute aus R1:

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

allgemeiner:

$F = F1 \wedge F2 \wedge F3$

F1: nur Attribute aus R1
F2: nur Attribute aus R2
F3: beides

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

Informations-
systeme

Übersicht
Abbildung ERM → RM
Relationen-algebra
Optimierung
Weitere Operationen

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) · Card (R)

- Selektivitätsfaktor SF bei:

$A_i = a_i$	SF =	$1/j_i$	wenn j_i bekannt (z.B. durch Index)
		1/10)	sonst
$A_i = A_k$	SF =	$1/Max(j_i,j_k)$	wenn j_i und j_k bekannt
		$1/j_i$	wenn j_i bekannt
		$1/j_k$	wenn j_k bekannt
		1/10	sonst
$A_i \geq a_i$ (oder $A_i > a_i$)	SF =	$(a_{max} – a_i) / (a_{max}-a_{min})$	wenn j_i bekannt und Wert inter-polierbar (Gleichvert. über Intervall)
		1/3	sonst
A_i BETWEEN a_i AND a_k	SF =	$(a_k–a_i) / (a_{max}-a_{min})$	wenn j_i bekannt und Wert inter-polierbar (Gleichvert. über Intervall)
		1/4	sonst

Informations-
systeme

Übersicht
Abbildung ERM → RM
Relationen-algebra
Optimierung
Weitere Operationen

Optimierung – Berechnungsgrundlagen (2)

- Berechnung von Ausdrücken
  - SF (p(A) ∧ p(B)) = SF (p(A)) · SF (p(B))
  - SF (p(A) ∨ p(B)) = SF (p(A)) + SF (p(B)) - SF (p(A)) · SF (p(B))
  - SF (¬ p(A)) = 1 - SF (p(A))
- Join-Selektivitätsfaktor (JSF)
  - Card (R ⋈ S) = JSF ∗ Card(R) ∗ Card(S)
    - JSF ergibt es aus SF für Joinbedingung (z.B. $A_i = A_k$)
  - bei (N:1)-Joins (verlustfrei): Card (R ⋈ S) = Max(Card(R), Card(S))

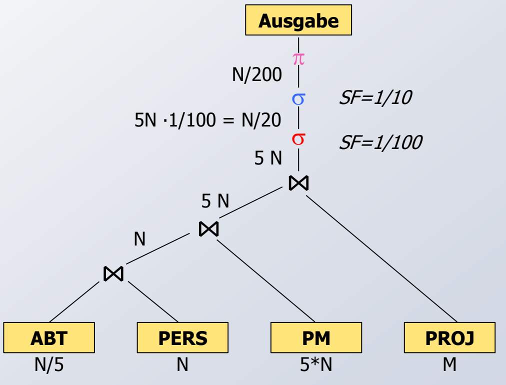
Algebraische Optimierung - Beispiel

Anfrage:

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

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

1. Ausgangslösung



- **Annahmen**

  Anzahl der Tupel in

  ABT: N/5

  PERS: N

  PM: 5*N (>M)

  PROJ: M

  Anzahl der Attributwerte von

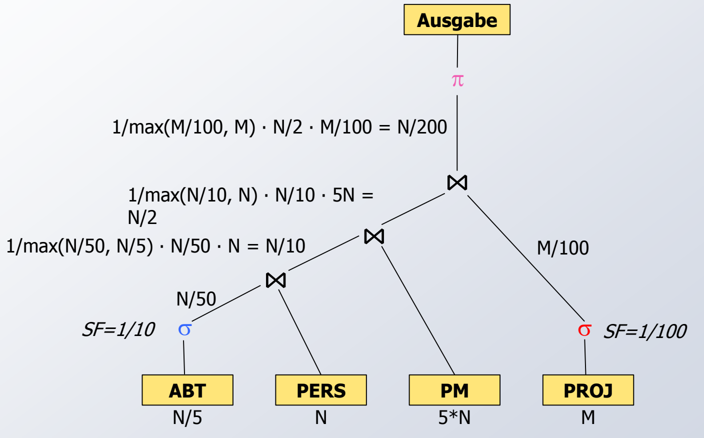
  A-ORT: 10

  P-ORT: 100

  Verlustfreie Verbunde über PS/FS

Algebraische Optimierung – Beispiel (2)

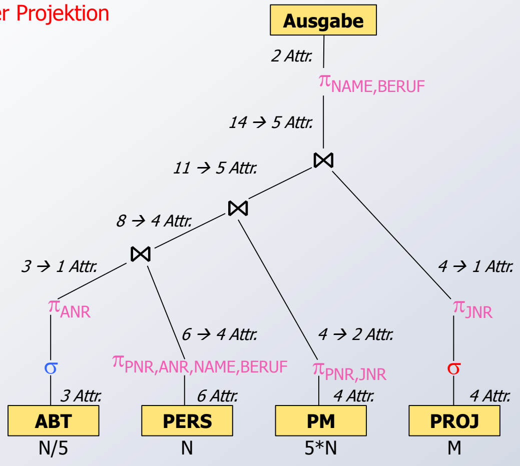
2. Verschieben der Selektion zu den Blattknoten



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

Algebraische Optimierung – Beispiel (3)

3. Verschieben der Projektion

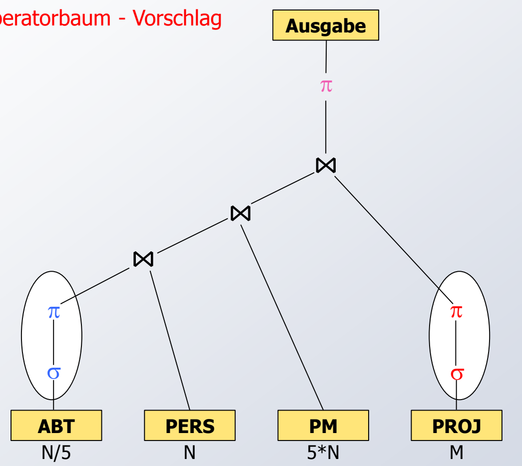


➥ II. *Führe Projektion (ohne Duplikateleminierung) frühzeitig aus!*
(Eliminierung von Duplikaten ist in der Regel sehr teuer)

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

Algebraische Optimierung – Beispiel (4)

4. Optimierter Operatorbaum - Vorschlag

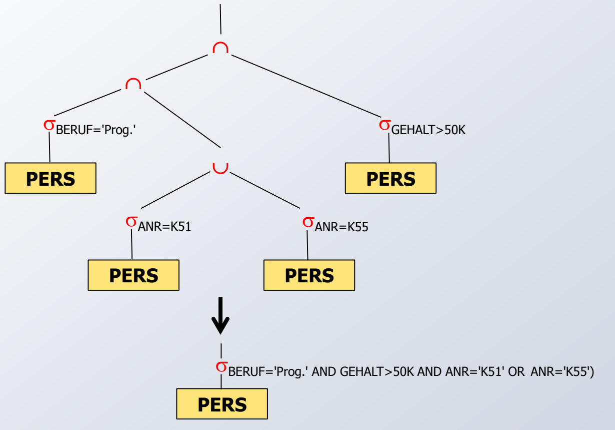


➥ III. *Verknüpfe Folgen von unären Operationen wie Selektion und Projektion (wenn diese tupelweise abgewickelt werden können)!*

Algebraische Optimierung – Beispiel (5)

5. Weitere Optimierungsmaßnahmen

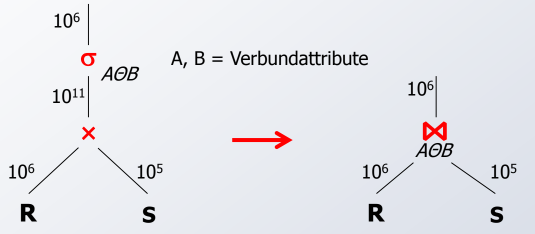
Ausdrucksauswertung: Alle Programmierer mit mehr als 50K Gehalt
und zwar aus Abteilungen K51 oder K55



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

Algebraische Optimierung – Beispiel (6)

6. Kartesisches Produkt mit Selektion



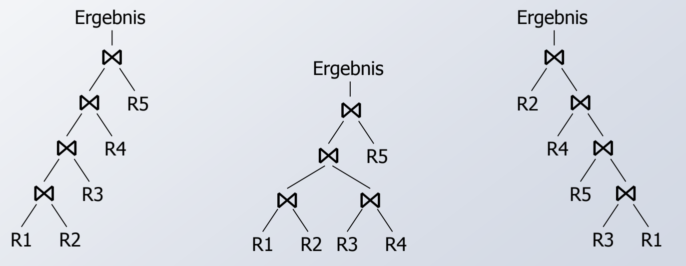
Leistungsbetrachtung bei:
a) Verlustfreier natürlicher Verbund
b) θ-Verbund (Annahme: 100 Ergebnistupel)

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

➥ VI. *Berechne gemeinsame Teilbäume nur einmal!*
(wenn die Zwischenspeicherung der Ergebnisse nicht zu teuer ist)

Algebraische Optimierung – Beispiel (7)

- Assoziativität und Kommutativität von Vereinigung, Durchschnitt, Verbund
  1. T = R3 ⋈ (R1 ⋈ R2)
  2. T = R2 ⋈ (R1 ⋈ R3)
  3. T = R1 ⋈ (R2 ⋈ R3)
- Allgemeines Problem bei binären Relationenoperationen
  - Was ist die beste Verknüpfungsreihenfolge?
  - Im allgemeinen Fall sind n! Reihenfolgen möglich
  - Die genaue Größe einer Zwischenrelation ergibt sich erst nach Ende der erzeugenden Operation
  - Einige Verknüpfungsreihenfolgen für den Verbund mit n=5



Algebraische Optimierung – Beispiel (8)

7. Kombination von Verbundoperationen

- Abschätzung mit j Werten des Verbundattributs und (N(R1) ≤ N(R2))

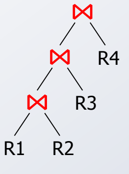
  N(T1) = N(R1) ∗ N(R2)/j

  beim kartesischen Produkt: analog zu j = 1 ⇒ N(T1) = N(R1) ∗ N(R2)

  beim (1:n)-Verbund: N(R1) = j ⇒ N(T1) = N(R2)

  beim (n:m)-Verbund: j < N(R1) ⇒ N(T1) > N(R2)

- Bestimmung der Verbundreihenfolge (Heuristik)



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

j ≤ N(R1) < N(R2) < N(R3) < N(R4)

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

- Dynamische Entscheidung aufwendiger, aber genauer

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

Informations-
systeme

Übersicht

Abbildung ERM → RM

Relationen-algebra

Optimierung

Weitere Operationen

Algebraische Optimierung – Beispiel (9)

8. Reihenfolgen von Mengenoperationen

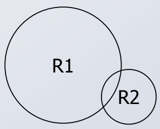
- Kardinalität der Vereinigung

  MAX(N(R1), N(R2)) ≤ N(R1 ∪ R2) ≤ N(R1) + N(R2)

- Kardinalität des Durchschnitts

  0 ≤ N(R1 ∩ R2) ≤ MIN(N(R1), N(R2))

Erwartung:



- Heuristische Regel

➥ VIII. *Verknüpfe bei Mengenoperationen immer zuerst die kleinsten Relationen!*

Informations-
systeme

Übersicht

Abbildung ERM → RM

Relationen-algebra

Optimierung

Weitere Operationen

Zusammenfassung: Algebraische Optimierung

- Heuristische Regeln

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

Division

- Ziel
  - Beantwortung von Fragen, bei denen eine „ganze Relation" zur Qualifikation herangezogen wird
  - Simulation des Allquantors $\Rightarrow$ ein Tupel aus R steht mit <u>allen</u> Tupeln aus S in einer bestimmten Beziehung
    - 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

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

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

  $R \div S = T - V$

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

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

Division - Beispiel

Datenbank:

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

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

Frage: Welche Schauspieler haben alle Rollen gespielt?

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

Auswertung:

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

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

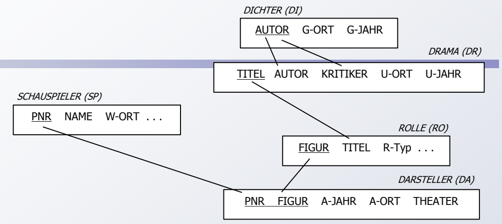
$\pi_{PNR,FIGUR}(DA)$	PNR	FIGUR
	P3	Faust
	P3	Nathan
	P3	Werther

? $\supseteq$

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

Ergebnis	PNR
	P3

Anfragen (3)



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

$$\pi_{NAME}\,(SP \bowtie (\underbrace{\pi_{PNR,FIGUR}\, DA}_{\textit{Rollen gespielt}} \;\div\; \underbrace{\pi_{FIGUR}\,(RO \bowtie (\sigma_{AUTOR=Goethe}\, DR))}_{\textit{alle Rollen in Goethe-Dramen}}\,))$$

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

$$\pi_{NAME}\,(SP \bowtie (\underbrace{(\pi_{PNR,FIGUR}\,(\sigma_{THEATER=Pfalztheater} DA))}_{\textit{Rollen am Pfalztheater gespielt}} \;\div\; \underbrace{(\pi_{FIGUR}\,(\sigma_{R\text{-}TYP=Narr}\, RO))}_{\textit{alle Narren-Rollen}}\,))$$

Intervallverbund (Band Join)

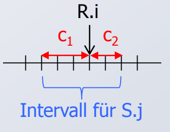
- Anstatt des arithmetischen Vergleichsoperators θ des θ -Joins wird hier eine Intervall-Bedingung überprüft.
- Grob:
  Kartesisches Produkt zwischen zwei Relationen R (Grad r) und S (Grad s) eingeschränkt durch eine Intervall-Bedingung zwischen i-Spalte von R und j-Spalte von S.
- Intervall I = $[c_1, c_2]$ mit c_1, c_2 sind positive Konstanten, wobei eine größer Null sein muss.

  Intervall-Verbund zwischen R und S:

$$V = R \underset{i\,I\,j}{\bowtie} S = \sigma_{i\,I\,j}(R \times S)$$

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

- 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.



Intervallverbund - Beipiel

- Beispiel:

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

PERS	PNR	ALTER
	P1	25
	P2	23
	P3	28

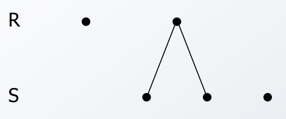
PERS'	PNR'	ALTER' ...
	P1	25
	P2	23
	P3	28

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

Äußerer Verbund (Outer Join)

- Ziel: Verlustfreier Verbund soll erzwungen werden

  Beispiel: Bei R ⋈ S sollen auch Teilobjekte als Ergebnis geliefert werden



- Bisher: R ⋈ S liefert nur vollständige Objekte
- Trick: Einfügen einer speziellen Leerzeile zur künstlichen Erzeugung von Verbundpartnern

Übersicht
Abbildung ERM → RM
Relationen-algebra
Optimierung
Weitere Operationen

Äußerer Verbund (2)

- Beispiel

SP (PNR, NAME, ...)

PNR	NAME
P1	x
P2	y
P3	≡

SP‘

DA (PNR, NAME, ...)

PNR	NAME
P1	F
P1	W
P3	M
P2	≡

DA‘

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

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

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

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

Informations-
systeme

Übersicht
Abbildung ERM → RM
Relationen-algebra
Optimierung
Weitere Operationen

Äußerer Verbund (3)

- Definition: Seien A die Verbundattribute, {≡} der undefinierte Wert und

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

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

Äußerer Gleichverbund

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

Äußerer natürlicher Gleichverbund

$$\boxed{R \mathbin{⟗} S := R' \bowtie S'}$$

Äußerer Verbund (4)

- 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

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

- Rechter äußerer Gleichverbund

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

Rechter äußerer Gleichverbund

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

Beispiele zum äußeren Gleichverbund

- Gleichverbund

R	A	B	C
	a_1	b_1	c_1
	a_2	b_2	c_2

$\bowtie$

S	C	D	E
	c_1	d_1	e_1
	c_3	d_2	e_2

=

ERG	A	B	C	D	E
	a_1	b_1	c_1	d_1	e_1

- Linker äußerer Gleichverbund

R	A	B	C
	a_1	b_1	c_1
	a_2	b_2	c_2

⟕

S	C	D	E
	c_1	d_1	e_1
	c_3	d_2	e_2

=

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

Beispiele zum äußeren Gleichverbund (2)

- Rechter äußerer Gleichverbund

R	A	B	C
	a_1	b_1	c_1
	a_2	b_2	c_2

⟖

S	C	D	E
	c_1	d_1	e_1
	c_3	d_2	e_2

=

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

- Äußerer Gleichverbund

R	A	B	C
	a_1	b_1	c_1
	a_2	b_2	c_2

⟗

S	C	D	E
	c_1	d_1	e_1
	c_3	d_2	e_2

=

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

Weitere äußere Operationen

- Äußere Vereinigung (*OUTER UNION*)

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

Beispiel:

STUDENT	MATNR	FBNR	SEM
	123	FB5	5
	789	FB9	9

HIWI	MATNR	FBNR	JOB
	456	FB5	Tutor
	987	FB9	Prog.

OUTER UNION

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

➡ Es können große Interpretationsprobleme beim Ergebnis entstehen

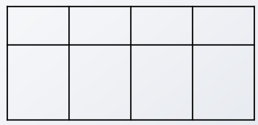
- In ähnlicher Weise lassen sich weitere Operationen einführen
  - OUTER INTERSECTION
  - OUTER DIFFERENCE

➡ Diese Operationen scheinen nicht besonders nützlich zu sein

Zusammenfassung: Abbildungskonzepte

- Datenstruktur

Relation (Tabelle)



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

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

Zusammenfassung: Relationenalgebra

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