



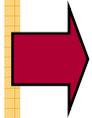
**Hasso
Plattner
Institut**

IT Systems Engineering | Universität Potsdam

Datenbanksysteme I
Relationale Algebra

Felix Naumann

2



- Einführung
- Basisoperatoren
- Operatoren auf Multimengen
- Erweiterte Operatoren



Einführung

3

Bisher

- Relationenschemata mit Basisrelationen, die in der Datenbank gespeichert sind

Jetzt

- „Abgeleitete“ Relationenschemata mit virtuellen Relationen, die aus den Basisrelationen berechnet werden
- Definiert durch Anfragen
- Basisrelationen bleiben unverändert

Kriterien für Anfragesprachen

4

Ad-Hoc-Formulierung

- Benutzer soll eine Anfrage formulieren können, ohne ein vollständiges Programm schreiben zu müssen

Deskriptivität

- Benutzer soll formulieren „Was will ich haben?“ und nicht „Wie komme ich an das, was ich haben will?“
- Deklarativ

Mengenorientiertheit

- Operationen auf Mengen von Daten
- Nicht navigierend nur auf einzelnen Elementen („tuple-at-a-time“)

Abgeschlossenheit

- Ergebnis ist wieder eine Relation und kann wieder als Eingabe für die nächste Anfrage verwendet werden.

Kriterien für Anfragesprachen

5

Adäquatheit

- Alle Konstrukte des zugrundeliegenden Datenmodells werden unterstützt

Orthogonalität

- Sprachkonstrukte sind in ähnlichen Situationen auch ähnlich anwendbar

Optimierbarkeit

- Sprache besteht aus wenigen Operationen, für die es Optimierungsregeln gibt

Effizienz

- Jede Operation ist effizient ausführbar
- Im relationalen Modell hat jede Operation eine Komplexität $\leq O(n^2)$, n Anzahl der Tupel einer Relation.

Kriterien für Anfragesprachen

6

Sicherheit

- Keine Anfrage, die syntaktisch korrekt ist, darf in eine Endlosschleife geraten oder ein unendliches Ergebnis liefern.

Eingeschränktheit

- Anfragesprache darf keine komplette Programmiersprache sein
- Folgt aus Sicherheit, Optimierbarkeit, Effizienz

Vollständigkeit

- Sprache muss mindestens die Anfragen einer Standardsprache (z.B. relationale Algebra) ausdrücken können.

Anfragealgebra

7

- Mathematik
 - Algebra definiert durch Wertebereich und auf diesem definierte Operatoren
 - Operand: Variablen oder Werte aus denen neue Werte konstruiert werden können
 - Operator: Symbole, die Prozeduren repräsentieren, die aus gegebenen werte neue Werte produzieren
- Für Datenbankankfragen
 - Inhalte der Datenbank (Relationen) sind Operanden
 - Operatoren definieren Funktionen zum Berechnen von Anfrageergebnissen
 - ◇ Grundlegenden Dinge, die wir mit Relationen tun wollen.
 - Relationale Algebra (Relationenalgebra, RA)
 - ◇ Anfragesprache für das relationale Modell

Mengen vs. Multimenge

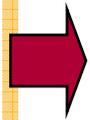
8

- Relation: Menge von Tupeln
- Datenbanktabelle: Multimenge von Tupeln
- Operatoren der relationalen Algebra: Operatoren auf Mengen
- Operatoren auf DBMS: SQL Anfragen
 - Rel. DBMS speichern Multimengen

- Motivation: Effizienzsteigerung
 - Beispiel:
 - ◇ Vereinigung als Multimenge
 - ◇ Vereinigung als Menge

9

- Einführung
- Basisoperatoren
- Operatoren auf Multimengen
- Erweiterte Operatoren



Klassifikation der Operatoren

10

- Mengenoperatoren
 - Vereinigung, Schnittmenge, Differenz
- Entfernde Operatoren
 - Selektion, Projektion
- Kombinerende Operatoren
 - Kartesisches Produkt, Join, Joinvarianten
- Umbenennung
 - Verändert nicht Tupel, sondern Schema

- Ausdrücke der relationalen Algebra: „Anfragen“ (queries)

Vereinigung (Union, \cup)

11

- Sammelt Elemente (Tupel) zweier Relationen unter einem gemeinsamen Schema auf.
- $R \cup S := \{t \mid t \in R \vee t \in S\}$
- Attributmengen beider Relationen müssen identisch sein.
 - Namen, Typen und Reihenfolge
 - Zur Not: Umbenennung
- Ein Element ist nur einmal in $(R \cup S)$ vertreten, auch wenn es jeweils einmal in R und S auftaucht.
 - Duplikatentfernung

Beispiel für Mengenoperatoren

12

R

Name	Adresse	Geschlecht	Geburt
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Mark Hamill	456 Oak Rd., Brentwood	M	8/8/88

S

Name	Adresse	Geschlecht	Geburt
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77

$R \cup S$

Name	Adresse	Geschlecht	Geburt
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Mark Hamill	456 Oak Rd., Brentwood	M	8/8/88
Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77

Differenz (Difference, $-$, $/$)

13

- Differenz $R - S$ eliminiert die Tupel aus der ersten Relation, die auch in der zweiten Relation vorkommen.
- $R - S := \{t \mid t \in R \wedge t \notin S\}$
- Achtung: $R - S \neq S - R$

Beispiel für Mengenoperatoren

14

R			
Name	Adresse	Geschlecht	Geburt
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Mark Hamill	456 Oak Rd., Brentwood	M	8/8/88

S			
Name	Adresse	Geschlecht	Geburt
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77

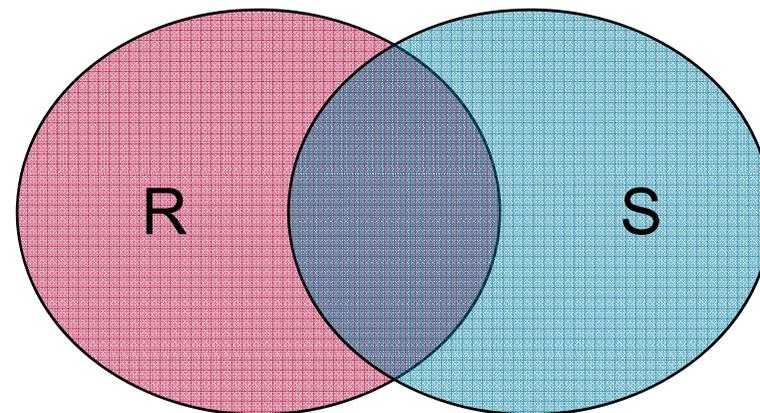
R - S

Name	Adresse	Geschlecht	Geburt
Mark Hamill	456 Oak Rd., Brentwood	M	8/8/88

Schnittmenge (Intersection, \cap)

15

- Durchschnitt $r1 \cap r2$ ergibt die Tupel, die in beiden Relationen gemeinsam vorkommen.
- $R \cap S := \{t \mid t \in R \wedge t \in S\}$
- Anmerkung: Durchschnitt ist überflüssig
 - $R \cap S = R - (R - S)$



Beispiel für Mengenoperatoren

16

R			
Name	Adresse	Geschlecht	Geburt
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Mark Hamill	456 Oak Rd., Brentwood	M	8/8/88

S			
Name	Adresse	Geschlecht	Geburt
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77

$R \cap S$

Name	Adresse	Geschlecht	Geburt
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99

Projektion (projection, π)

17

- Erzeugt neue Relation mit einer Teilmenge der ursprünglichen Attribute
- $\pi_{A_1, A_2, \dots, A_n}(R)$ ist eine Relation
 - mit den Attributen A_1, A_2, \dots, A_n
 - Üblicherweise in der aufgelisteten Reihenfolge
- Achtung: Es können Duplikate entstehen, die entfernt werden müssen.

Projektion – Beispiel

18

Filme					
Titel	Jahr	Länge	inFarbe	Studio	ProduzentID
Total Recall	1990	113	True	Fox	12345
Basic Instinct	1992	127	True	Disney	67890
Dead Man	1995	121	False	Paramount	99999

$\pi_{\text{Titel, Jahr, Länge}}(\text{Filme})$

Titel	Jahr	Länge
Total Recall	1990	113
Basic Instinct	1992	127
Dead Man	1995	121

$\pi_{\text{inFarbe}}(\text{Filme})$

inFarbe
True
False

Selektion (selection, σ)

19

- Erzeugt neue Relation mit gleichem Schema aber einer Teilmenge der Tupel.
- Nur Tupel, die der Selektionsbedingung C (condition) entsprechen.
 - Selektionsbedingung wie aus Programmiersprachen
 - Operanden der Selektionsbedingung sind nur Konstanten oder Attribute von R .
- Prüfe Bedingung für jedes Tupel

Selektion – Beispiel

20

Filme					
Titel	Jahr	Länge	inFarbe	Studio	ProduzentID
Total Recall	1990	113	True	Fox	12345
Basic Instinct	1992	127	True	Disney	67890
Dead Man	1995	90	False	Paramount	99999

$\sigma_{\text{Länge} \geq 100}(\text{Filme})$

Titel	Jahr	Länge	inFarbe	Studio	ProduzentID
Total Recall	1990	113	True	Fox	12345
Basic Instinct	1992	127	True	Disney	67890

Selektion – Beispiel

21

Filme					
Titel	Jahr	Länge	inFarbe	Studio	ProduzentID
Total Recall	1990	113	True	Fox	12345
Basic Instinct	1992	127	True	Disney	67890
Dead Man	1995	90	False	Paramount	99999

$\sigma_{\text{Länge} \geq 100 \text{ AND Studio} = \text{'Fox'}}(\text{Filme})$

Titel	Jahr	Länge	inFarbe	Studio	ProduzentID
Total Recall	1990	113	True	Fox	12345

Kartesisches Produkt (cartesian produkt, \times)

22

- Auch: Kreuzprodukt oder Produkt
- Auch: $R * S$ statt $R \times S$
- Kreuzprodukt zweier Relationen R und S ist die Menge aller Tupel, die man erhält, wenn man jedes Tupel aus R mit jedem Tupel aus S paart.
- Schema hat ein Attribut für jedes Attribut aus R und S
 - Achtung: Bei Namensgleichheit wird kein Attribut ausgelassen
 - Stattdessen: Umbenennen

Kartesisches Produkt – Beispiel

23

R

A	B
1	2
3	4

S

B	C	D
2	5	6
4	7	8
9	10	11

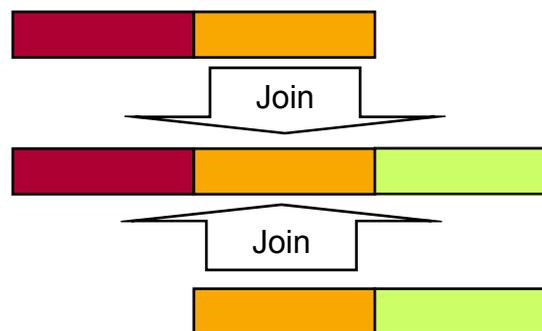
R × S

A	R.B	S.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

Natürlicher Join (natural Join, \bowtie)

24

- Motivation: Statt im Kreuzprodukt alle Paare zu bilden, sollen nur die Tupelpaare gebildet werden, deren Tupel irgendwie übereinstimmen.
 - Auch: „Verbund“
 - Beim natürlichen Join: Übereinstimmung in allen gemeinsamen Attributen.
 - Gegebenenfalls Umbenennung
 - Schema: Vereinigung der beiden Attributmengen



- Anmerkung: Dies war der Join zur Wiederherstellung nach Dekomposition

Natürlicher Join – Beispiel

25

R

A	B
1	2
3	4

S

B	C	D
2	5	6
4	7	8
9	10	11

R ⋈ S

A	B	C	D
1	2	5	6
3	4	7	8

R × S

A	R.B	S.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

Natürlicher Join – Beispiel

26

R

A	B	C
1	2	3
6	7	8
9	7	8

S

B	C	D
2	5	6
2	3	5
7	8	10

R ⋈ S

A	B	C	D
1	2	3	5
6	7	8	10
9	7	8	10

Anmerkungen

- Mehr als ein gemeinsames Attribut
- Tupel werden mit mehr als einem Partner verknüpft

Theta-Join (theta-join, \bowtie_{θ})

27

- Verallgemeinerung des natürlichen Joins
- Verknüpfungsbedingung kann selbst gestaltet werden.
- Konstruktion des Ergebnisses:
 - Bilde Kreuzprodukt
 - Selektiere mittels der Joinbedingung
 - Also: $R \bowtie_{\theta} S = \sigma_{\theta} (R \times S)$
- Schema: Wie beim Kreuzprodukt
- Natural Join ist ein Spezialfall des Theta-Joins
 - Aber: Schema des Ergebnisses sieht anders aus.

Theta-Join – Beispiel

28

R	A	B	C
	1	2	3
	6	7	8
	9	7	8

S	B	C	D
	2	5	6
	2	3	5
	7	8	10

$R \bowtie_{A < D} S$

A	R.B	R.C	S.B	S.C	D
1	2	3	2	5	6
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	7	8	10
9	7	8	7	8	10

$R \bowtie_{A < D \text{ AND } R.B \neq S.B} S$

A	R.B	R.C	S.B	S.C	D
1	2	3	7	8	10

Komplexe Ausdrücke

29

Idee: Kombination (Schachtelung) von Ausdrücken zur Formulierung komplexer Anfragen.

- Abgeschlossenheit der relationalen Algebra
 - Output eines Ausdrucks ist immer eine Relation.
- Darstellung
 - Als geschachtelter Ausdruck mittels Klammerung
 - Als Baum

Komplexe Ausdrücke – Beispiel

30

Filme

Titel	Jahr	Länge	Typ	StudioName
Total Recall	1990	113	Farbe	Fox
Basic Instinct	1992	127	Farbe	Disney
Dead Man	1995	90	s/w	Paramount

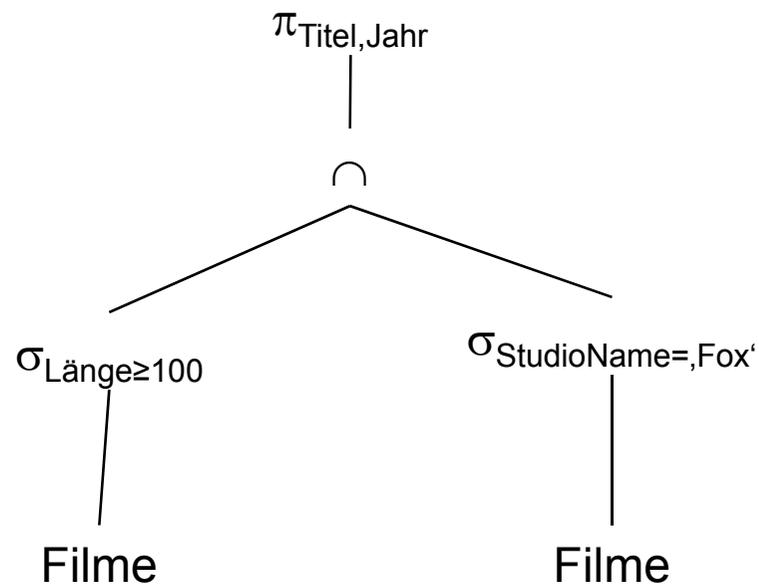
Gesucht: Titel und Jahr von Filmen, die von Fox produziert wurden und mindestens 100 Minuten lang sind.

- Suche alle Filme von Fox
- Suche alle Filme mit mindestens 100 Minuten
- Bilde die Schnittmenge der beiden Zwischenergebnisse
- Projiziere die Relation auf die Attribute Titel und Jahr.
- $\pi_{\text{Titel, Jahr}}(\sigma_{\text{Länge} \geq 100}(\text{Filme}) \cap \sigma_{\text{StudioName} = \text{'Fox'}}(\text{Filme}))$

Komplexe Ausdrücke – Beispiel

31

- $\pi_{\text{Titel, Jahr}}(\sigma_{\text{Länge} \geq 100}(\text{Filme}) \cap \sigma_{\text{StudioName} = \text{'Fox'}}(\text{Filme}))$



- Alternative: $\pi_{\text{Titel, Jahr}}(\sigma_{\text{Länge} \geq 100 \text{ AND StudioName} = \text{'Fox'}}(\text{Filme}))$

Komplexe Ausdrücke – Beispiel

32

Filme1

Titel	Jahr	Länge	Typ	StudioName
Total Recall	1990	113	Farbe	Fox
Basic Instinct	1992	127	Farbe	Disney
Dead Man	1995	121	s/w	Paramount

Filme2

Titel	Jahr	SchauspName
Total Recall	1990	Sharon Stone
Basic Instinct	1992	Sharon Stone
Total Recall	1990	Arnold
Dead Man	1995	Johnny Depp

Gesucht: Namen der Stars, die in Filmen spielten, die mindestens 100 Minuten lang sind.

- Verjoine beide Relationen (natürlicher Join)
- Selektiere Filme, die mindestens 100 Minuten lang sind.
- $\pi_{\text{SchauspName}}(\sigma_{\text{Länge} \geq 100}(\text{Filme1} \bowtie \text{Filme2}))$

Umbenennung (rename, ρ)

33

Motivation: Zur Kontrolle der Schemata und einfacheren Verknüpfungen

- $\rho_{S(A_1, \dots, A_n)}(R)$
 - Benennt Relation R in S um
 - Benennt die Attribute der neuen Relation A_1, \dots, A_n
- $\rho_S(R)$ benennt nur Relation um.

Durch Umbenennung ermöglicht

- Joins, wo bisher kartesische Produkte ausgeführt wurde
 - Unterschiedliche Attribute werden gleich benannt.
- Kartesische Produkte, wo bisher Joins ausgeführt wurden
 - Gleiche Attribute werden unterschiedlich genannt.
- Mengenoperationen
 - Nur möglich bei gleichen Schemata

Umbenennung - Beispiel

34

R	
A	B
1	2
3	4

S		
B	C	D
2	5	6
4	7	8
9	10	11

$R \times \rho_{S(X,C,D)}(S)$

A	B	X	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

- Alternativer Ausdruck: $\rho_{S(A,B,X,C,D)}(R \times S)$

Unabhängigkeit und Vollständigkeit

35

- Minimale Relationenalgebra:
 - $\pi, \sigma, \times, \rho, \cup$ und $-$
- Unabhängig:
 - Kein Operator kann weggelassen werden ohne Vollständigkeit zu verlieren.
- Natural Join, Join und Schnittmenge sind redundant
 - $R \cap S = R - (R - S)$
 - $R \bowtie_C S = \sigma_C(R \times S)$
 - $R \bowtie S = \pi_L(\sigma_{R.A1=S.A1 \text{ AND } \dots \text{ AND } R.An=S.An}(R \times S))$

Vorschau zu Optimierung

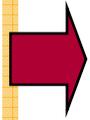
36

Beispiele für Algebraische Regeln zur Transformation

- $R \bowtie S = S \bowtie R$
- $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$
- $\pi_Y(\pi_X(R)) = \pi_Y(R)$
- $\sigma_{A=a}(\sigma_{B=b}(R)) = \sigma_{B=b}(\sigma_{A=a}(R)) [= \sigma_{B=b \wedge A=a}(R)]$
- $\pi_X(\sigma_{A=a}(R)) = \sigma_{A=a}(\pi_X(R))$
- $\sigma_{A=a}(R \cup S) = \sigma_{A=a}(R) \cup \sigma_{A=a}(S)$

37

- Einführung
- Basisoperatoren
- Operatoren auf Multimengen
- Erweiterte Operatoren



Motivation

38

- Mengen sind ein natürliches Konstrukt
 - ◇ Keine Duplikate
- Kommerzielle DBMS basieren fast nie nur auf Mengen
 - ◇ Sondern erlauben Multimengen
 - ◇ Duplikate sind erlaubt
- Multimenge
 - ◇ *bag, multiset*

A	B
1	2
3	4
1	2
1	2

Multimenge

Reihenfolge ist
weiter unwichtig

Effizienz durch Multimengen

39

- Bei Vereinigung
 - ◇ Direkt „aneinanderhängen“
- Bei Projektion
 - ◇ Einfach Attributwerte „abschneiden“
- Nach Duplikaten suchen
 - ◇ Jedes Tupel im Ergebnis mit jedem anderen vergleichen
- Effizienter nach Duplikaten suchen
 - ◇ Nach allen Attributen zugleich sortieren
- Bei Aggregation
 - ◇ Duplikateliminierung schädlich
 - ◇ $AVG(A) = ?$

Projektion auf
(A,B)

A	B	C
1	2	5
3	4	6
1	2	7
1	2	8

Vereinigung auf Multimengen

40

- Sei R eine Multimenge
 - ◇ Tupel t erscheine n-mal in R.
- Sei S eine Multimenge
 - ◇ Tupel t erscheine m-mal in S.
- Tupel t erscheint in $R \cup S$
 - ◇ $(n+m)$ mal.

R

A	B
1	2
3	4
1	2
1	2

S

A	B
1	2
3	4
3	4
5	6

$R \cup S$

A	B
1	2
3	4
1	2
1	2
1	2
3	4
3	4
5	6

Schnittmenge auf Multimengen

41

- Sei R eine Multimenge
 - Tupel t erscheine n-mal in R.
- Sei S eine Multimenge
 - Tupel t erscheine m-mal in S.
- Tupel t erscheint in $R \cap S$
 - $\min(n,m)$ mal.

R

A	B
1	2
3	4
1	2
3	4
1	2

S

A	B
1	2
3	4
3	4
5	6

$R \cap S$

A	B
1	2
3	4
3	4

Differenz auf Multimengen

42

- Sei R eine Multimenge
 - Tupel t erscheine n-mal in R.
- Sei S eine Multimenge
 - Tupel t erscheine m-mal in S.
- Tupel t erscheint in R - S
 - $\max(0, n-m)$ mal.
 - Falls t öfters in R als in S vorkommt, bleiben n-m t übrig.
 - Falls t öfters in S als in R vorkommt, bleibt kein t übrig.
 - Jedes Vorkommen von t in S eliminiert ein t in R.

R

A	B
1	2
3	4
1	2
1	2

S

A	B
1	2
3	4
3	4
5	6

R - S

A	B
1	2
1	2

S - R

A	B
3	4
5	6

Projektion und Selektion auf Multimengen

43

■ Projektion

- Bei der Projektion können neue Duplikate entstehen.
- Diese werden nicht entfernt

R	A	B	C
	1	2	5
	3	4	6
	1	2	7
	1	2	7

$\pi_{A,B}(R)$	A	B
	1	2
	3	4
	1	2
	1	2

■ Selektion

- Selektionsbedingung auf jedes Tupel einzeln und unabhängig anwenden
- Schon vorhandene Duplikate bleiben erhalten
 - ◇ Sofern sie beide selektiert bleiben

$\sigma_{C \geq 6}(R)$	A	B	C
	3	4	6
	1	2	7
	1	2	7

Kreuzprodukt auf Multimengen

44

- Sei R eine Multimenge
 - Tupel t erscheine n-mal in R.
- Sei S eine Multimenge
 - Tupel u erscheine m-mal in S.
- Das Tupel tu erscheint in $R \times S$ n·m mal.

R

A	B
1	2
1	2

S

B	C
2	3
4	5
4	5

A	R.B	S.B	C
1	2	2	3
1	2	2	3
1	2	4	5
1	2	4	5
1	2	4	5
1	2	4	5

Joins auf Multimengen

45

Keine Überraschungen

R

A	B
1	2
1	2

S

B	C
2	3
4	5
4	5

$R \bowtie S$

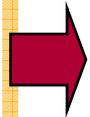
A	B	C
1	2	3
1	2	3

$R \bowtie_{R.B < S.B} S$

A	R.B	S.B	C
1	2	4	5
1	2	4	5
1	2	4	5
1	2	4	5

46

- Einführung
- Basisoperatoren
- Operatoren auf Multimengen
- Erweiterte Operatoren



Überblick über Erweiterungen

47

- Duplikateliminierung
- Aggregation
- Gruppierung
- Sortierung
- Erweiterte Projektion
- Outer Join
- Outer Union
- Semijoin
- Division

Duplikateliminierung (duplicate elimination, δ)

48

Wandelt eine Multimenge in eine Menge um.

- Durch Löschen aller Kopien von Tupeln
- $\delta(R)$

R

A	B
1	2
3	4
1	2
1	2

$\delta(R)$

A	B
1	2
3	4

Aggregation

49

- Aggregation fasst Werte einer Spalte zusammen.
 - Operation auf einer Menge oder Multimenge atomarer Werte (nicht Tupel)
 - Summe (SUM)
 - Durchschnitt (AVG)
 - Minimum (MIN) und Maximum (MAX)
 - ◇ Lexikographisch für nicht-numerische Werte
 - Anzahl (COUNT)
 - ◇ Doppelte Werte gehen auch doppelt ein.
 - ◇ Angewandt auf ein beliebiges Attribut ergibt dies die Anzahl der Tupel in der Relation.

R

A	B
1	2
3	4
1	2
1	2

- $SUM(B) = 10$
- $AVG(A) = 1,5$
- $MIN(A) = 1$
- $MAX(B) = 4$
- $COUNT(A) = 4$

Gruppierung

50

Partitionierung der Tupel einer Relation gemäß ihrer Werte in einem oder mehr Attributen.

- Hauptzweck: Aggregation auf Teilen einer Relation (Gruppen)

- Gegeben
 - Filme(Titel, Jahr, Länge, inFarbe, StudioName, ProduzentID)
- Gesucht: Gesamtminuten **pro** Studio
 - Gesamtminuten(StudioName, SummeMinuten)
- Verfahren:
 - Gruppieren nach StudioName
 - Summiere **in jeder Gruppe** die Länge der Filme

Gruppierung (grouping, γ)

51

- $\gamma_L(R)$ wobei L eine Menge von Attributen ist. Ein Element in L ist entweder
 - Ein Gruppierungsattribut nach dem gruppiert wird
 - Oder ein Aggregationsoperator auf ein Attribut von R (inkl. Neuen Namen für das aggregierte Attribut)
- Ergebnis wird wie folgt konstruiert:
 - Partitioniere R in Gruppen, wobei jede Gruppe gleiche Werte im Gruppierungsoperator hat
 - ◇ Falls kein Gruppierungsoperator angegeben: Ganz R ist die Gruppe
 - Für jede Gruppe erzeuge ein Tupel mit
 - ◇ Wert der Gruppierungsattribute
 - ◇ Aggregierte Werte über alle Tupel der Gruppe

Gruppierung – Beispiel

52

- Gegeben: SpieltIn(Titel, Jahr, SchauspName)
- Gesucht: Für jeden Schauspieler, der in mindestens 3 Filmen spielte, das Jahr des ersten Filmes.
- Idee
 - Gruppierung nach SchauspName
 - Minimum vom Jahr und Count von Titeln
 - Selektion nach Anzahl der Filme
 - Projektion auf Schauspielernamen und Jahr
- $\pi_{\text{SchauspName, MinJahr}}(\sigma_{\text{AnzahlTitel} \geq 3}(\gamma_{\text{SchauspName, MIN}(\text{Jahr}) \rightarrow \text{MinJahr, COUNT}(\text{Titel}) \rightarrow \text{AnzahlTitel}}(\text{SpieltIn})))$

Sortierung (sorting, τ)

53

- $\tau_L(R)$ wobei L eine Attributliste aus R ist.
 - Falls $L = (A_1, A_2, \dots, A_n)$ wir zuerst nach A_1 , bei gleichen A_1 nach A_2 usw. sortiert.
- Wichtig: Ergebnis der Sortierung ist keine Menge, sondern eine Liste.
 - Deshalb: Sortierung ist letzter Operator eines Ausdrucks. Ansonsten würden wieder Mengen entstehen und die Sortierung wäre verloren.
 - Trotzdem: Es macht manchmal auch Sinn zwischendurch zu sortieren.

Erweiterte Projektion

54

Motivation: Mehr Fähigkeiten in den Projektionsoperator geben.

- Vorher: $\pi_L(R)$ wobei L eine Attributliste ist
- Nun: Ein Element von L ist eines dieser drei Dinge
 - Ein Attribut von R (wie vorher)
 - Ein Ausdruck $X \rightarrow Y$ wobei X ein Attribut in R ist und Y ein neuer Name ist.
 - Ein Ausdruck $E \rightarrow Z$, wobei E ein Ausdruck mit Konstanten, arithmetischen Operatoren, Attributen von R und String-Operationen ist und Z ein neuer Name ist.
 - ◇ $A1 + A2 \rightarrow$ Summe
 - ◇ Vorname || Nachname \rightarrow Name

Erweiterte Projektion – Beispiel

55

R

A	B	C
0	1	2
0	1	2
3	4	5

$\pi_{A,B+C \rightarrow X}(R)$

A	X
0	3
0	3
3	9

Duplikate bleiben erhalten

$\pi_{B-A \rightarrow X, C-B \rightarrow Y}(R)$

X	Y
1	1
1	1
1	1

Neue Duplikate können entstehen

Semi-Join (\bowtie)

56

- Formal

- $R(A), S(B)$

- $R \bowtie S := \pi_A(R \bowtie_F S)$
 $= \pi_A(R) \bowtie_F \pi_{A \cap B}(S)$
 $= R \bowtie_F \pi_{A \cap B}(S)$
i.d.R. $= R \bowtie_F \pi_F(S)$

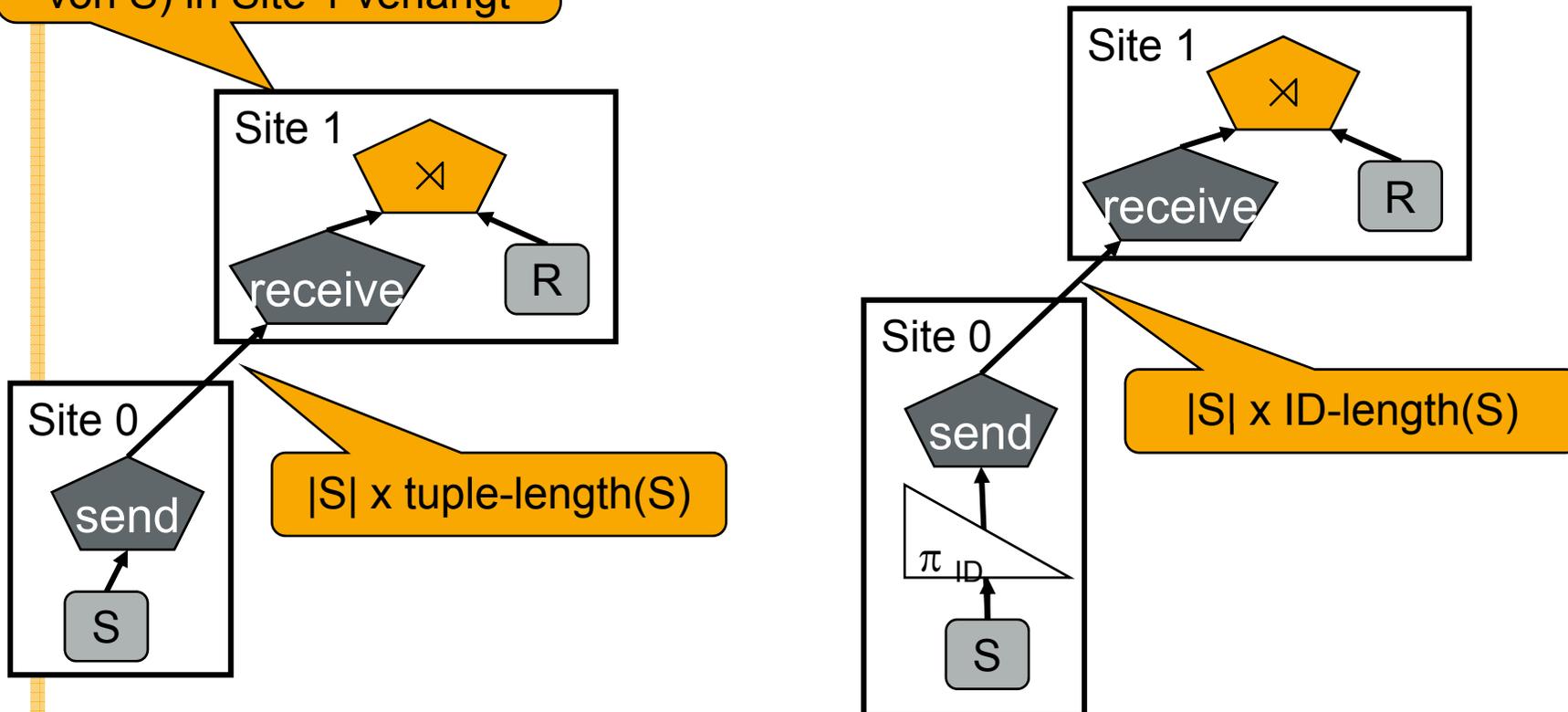
- In Worten: Join über R und S, aber nur die Attribute von R sind interessant.

- Nicht symmetrisch!

Semi-Join

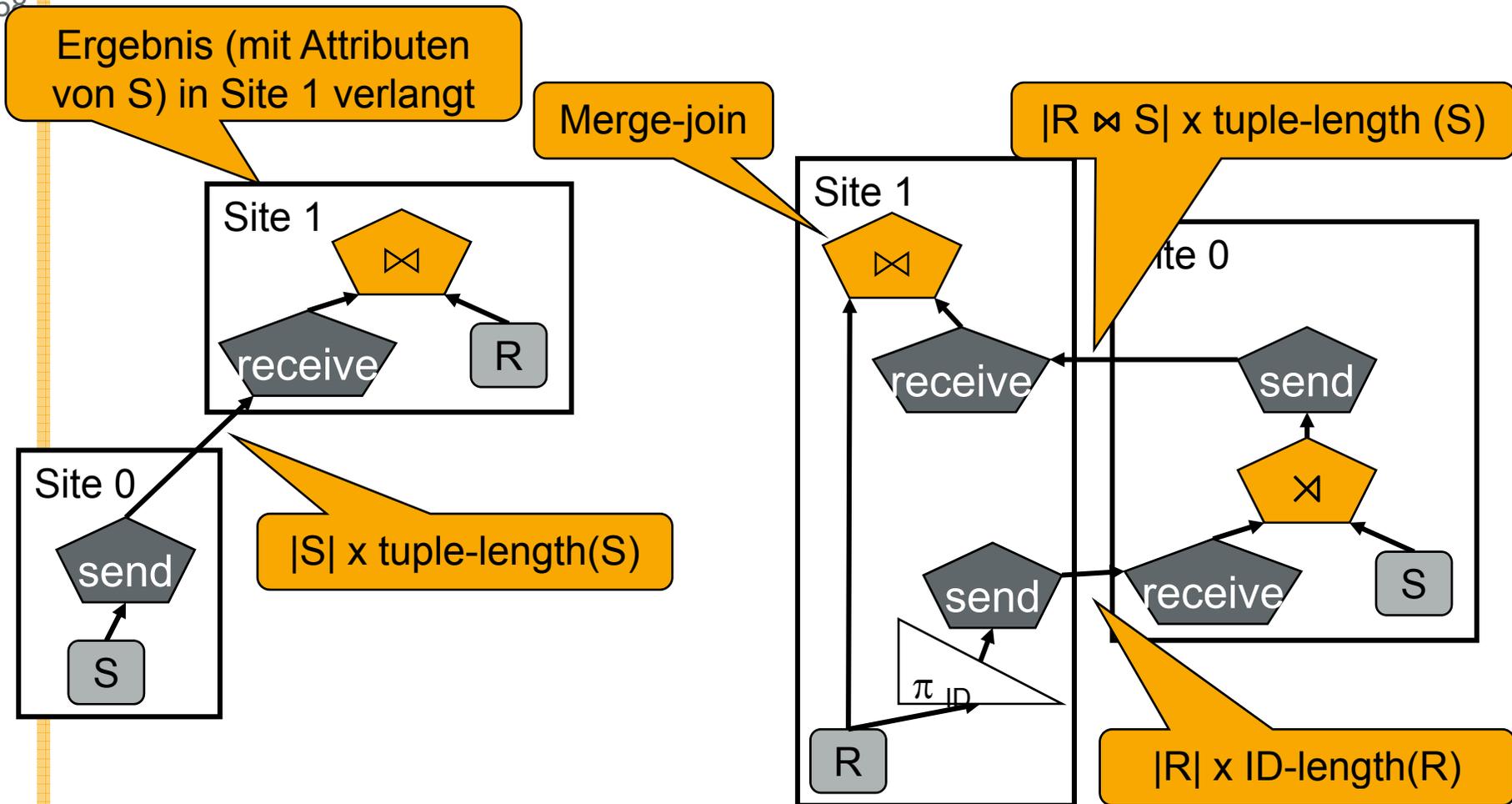
57

Ergebnis (ohne Attribute von S) in Site 1 verlangt



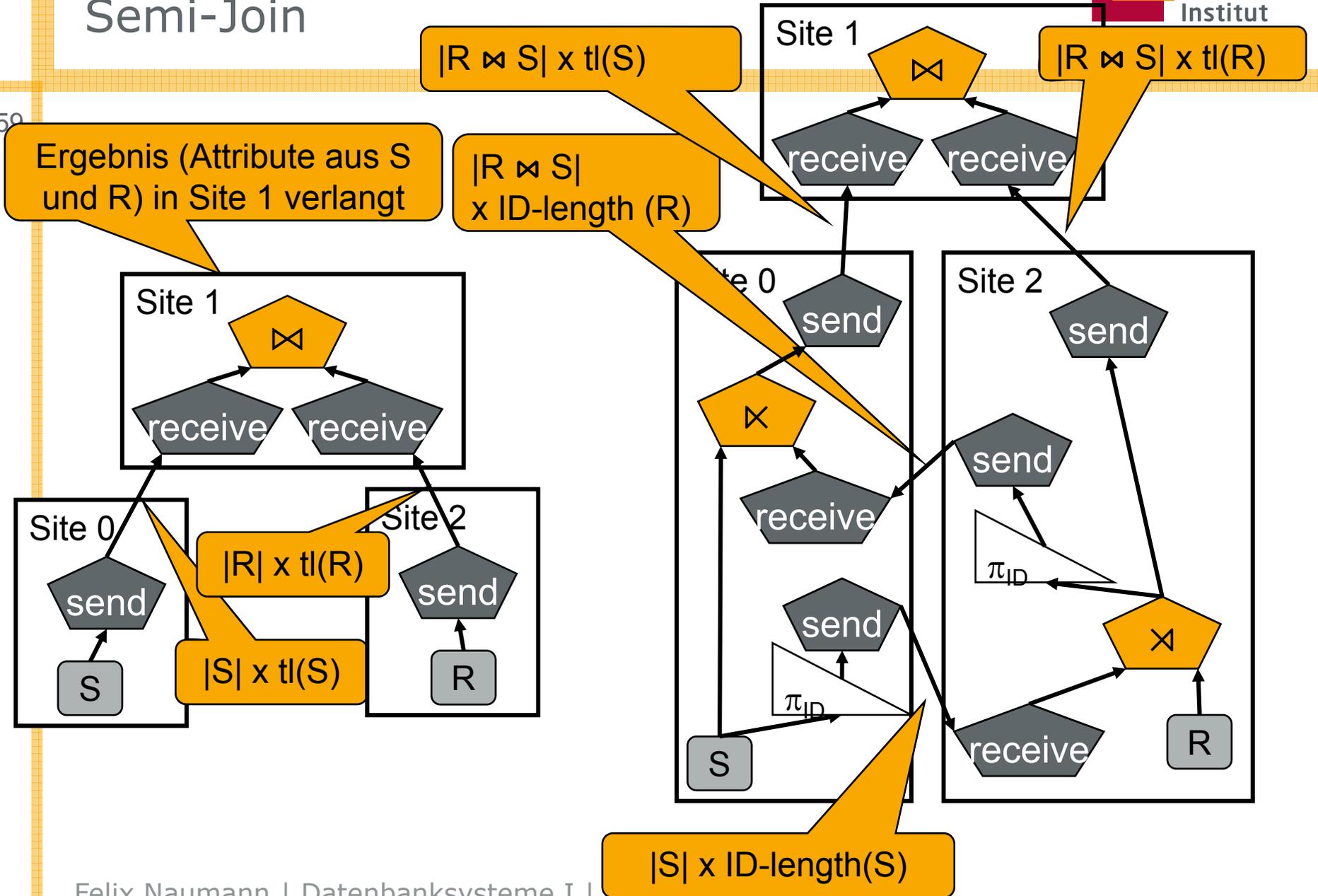
Semi-Join

58



Semi-Join

59



Outer Joins (Äußere Verbände, $|\bowtie|$)

60

- Übernahme von „*dangling tuples*“ in das Ergebnis und Auffüllen mit Nullwerten (*padding*)

- Full outer join
 - Übernimmt alle Tupel beider Operanden
 - $R |\bowtie| S$

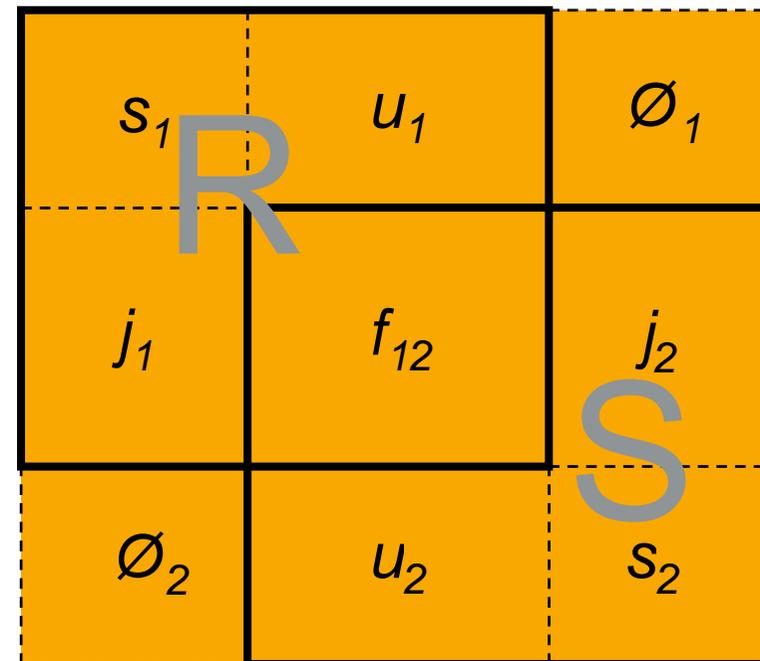
- Left outer join (right outer join)
 - Übernimmt alle Tupel des linken (rechten) Operanden
 - $R |\bowtie| S$ (bzw. $R \bowtie| S$)

- Andere Schreibweisen:
 - Herkömmlicher Join = „Inner join“

Outer Joins

61

- $R \bowtie S$
- $R \left| \bowtie S \right.$
- $R \bowtie \left| S \right.$
- $R \left| \bowtie \left| S \right. \right.$



Outer Joins

62

LINKS

A	B
1	2
2	3

RECHTS

B	C
3	4
4	5

\bowtie

A	B	C
2	3	4

$\bowtie\llcorner$

A	B	C
1	2	⊥
2	3	4
⊥	4	5

$\llcorner\bowtie$

A	B	C
1	2	⊥
2	3	4

$\lrcorner\bowtie$

A	B	C
2	3	4
⊥	4	5

Outer Joins und Informationsintegration

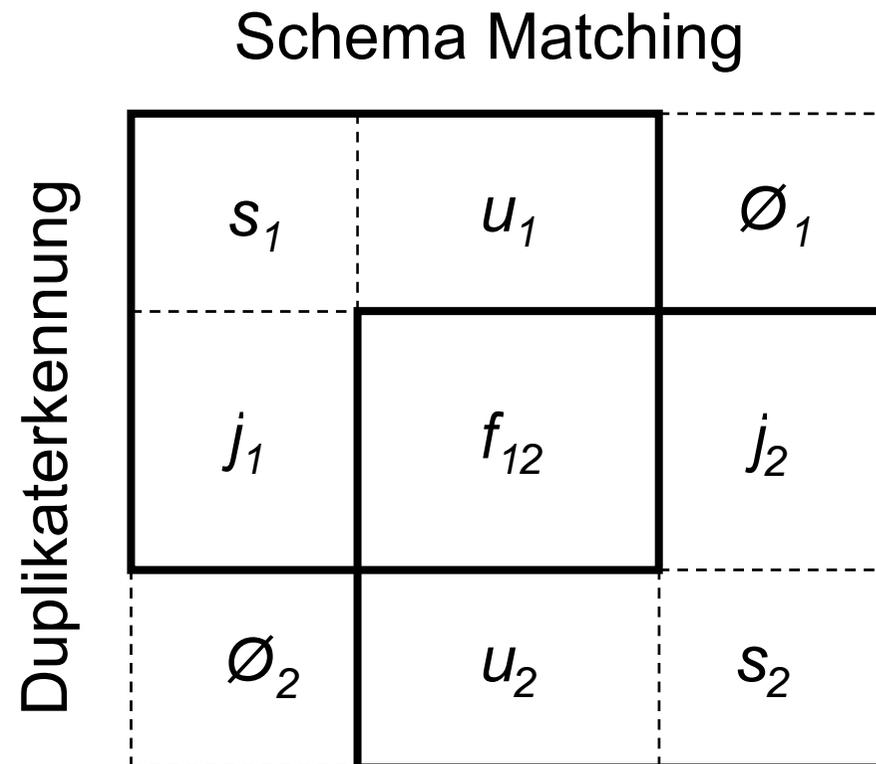
63

Ziel: Möglichst viele Informationen

- Viele Tupel
- Viele Attribute

Problem

- Überlappende Attribute erkennen
 - = Schema Matching
- Überlappende Tupel erkennen
 - = Duplikaterkennung



Outer Union (\cup)

64

Wie Vereinigung, aber auch mit inkompatiblen Schemata

- Schema ist Vereinigung der Attributmengen
- Fehlende Werte werden mit Nullwerten ergänzt.

R	A	B	C
	1	2	3
	6	7	8
	9	7	8

S	B	C	D
	2	5	6
	2	3	5
	7	8	10

R \cup S	A	B	C	D
	1	2	3	⊥
	6	7	8	⊥
	9	7	8	⊥
	⊥	2	5	6
	⊥	2	3	5
	⊥	7	8	10

Division (division, /)

65

- Nicht als primitiver Operator unterstützt.
- Finde alle Segler, die alle Segelboote reserviert haben.
- Relation $R(x,y)$, Relation $S(y)$
 - $R/S = \{ t \mid \exists x, y \in R \forall y \in S \}$
 - R/S enthält alle x -Tupel (Segler), so dass es für jedes y -Tupel (Boot) in S ein xy -Tupel in R gibt.
 - Andersherum: Falls die Menge der y -Werte (Boote), die mit einem x -Wert (Segler) assoziiert sind, alle y -Werte in S enthält, so ist der x -Wert in R/S .

Folie und Beispiel aus: Ramakrishnan, Gehrke „Database Management Systems“

Division – Beispiel

66

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

A

pno
p2

B1

sno
s1
s2
s3
s4

A/B1

pno
p2
p4

B2

sno
s1
s4

A/B2

pno
p1
p2
p4

B3

sno
s1

A/B3

Division ausdrücken

67

Division ist kein essentieller Operator, nur nützliche Abkürzung.

- Ebenso wie Joins, aber Joins sind so üblich, dass Systeme sie speziell unterstützen.
- Idee: Um R/S zu berechnen, berechne alle x -Werte, die nicht durch einen y -Wert in S „disqualifiziert“ werden.
 - x -Wert ist disqualifiziert, falls man durch Anfügen eines y -Wertes ein xy -Tupel erhält, das nicht in R ist.
- Disqualifizierte x -Werte: $\pi_x ((\pi_x(R) \times S) - R)$
- R/S : $\pi_x (R) -$ alle disqualifizierten Tupel

Division

68

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

A

pno
p2

B1

sno
s1
s2
s3
s4

A/B1

pno
p2
p4

B2

sno
s1
s4

A/B2

pno
p1
p2
p4

B3

sno
s1

A/B3

$$\pi_x(R) - \pi_x((\pi_x(R) \times S) - R)$$