Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Datenbanksysteme I
Relationale Algebra

Felix Naumann

11.5.2011

Überblick

- Einführung
- Basisoperatoren
- Operatoren auf Multimengen
- Erweiterte Operatoren

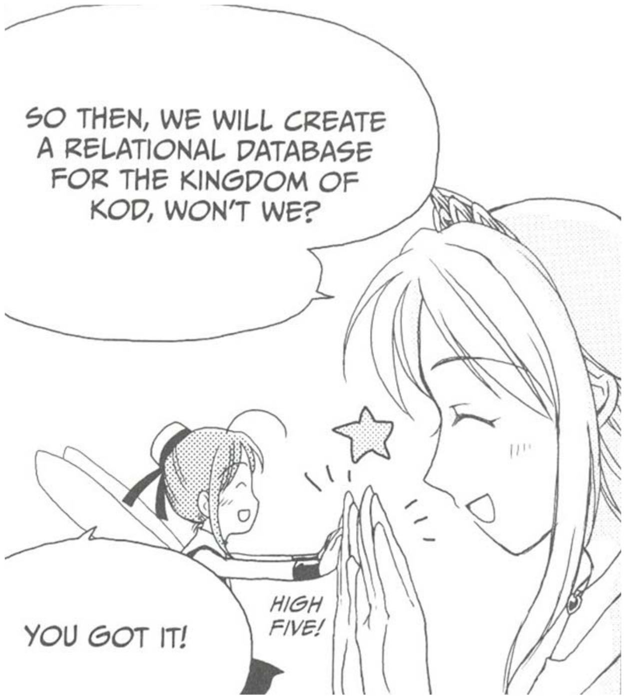
Einführung

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
  - Anfragesprache
- Basisrelationen bleiben unverändert



Kriterien für Anfragesprachen

- Ad-Hoc-Formulierung
  - Benutzer soll eine Anfrage formulieren können, ohne ein vollständiges Programm schreiben zu müssen.
- Deskriptivität / Deklarativität
  - Benutzer soll formulieren „Was will ich haben?" und nicht „Wie komme ich an das, was ich haben will?"
  - Deklarativ
- Optimierbarkeit
  - Sprache besteht aus wenigen Operationen, für die es Optimierungsregeln gibt
- Mengenorientiertheit
  - Operationen auf Mengen von Daten
  - Nicht navigierend nur auf einzelnen Elementen („tuple-at-a-time")

Kriterien für Anfragesprachen

- Abgeschlossenheit
  - Ergebnis ist wieder eine Relation und kann wieder als Eingabe für die nächste Anfrage verwendet werden.
- Adäquatheit
  - Alle Konstrukte des zugrundeliegenden Datenmodells werden unterstützt
- Orthogonalität
  - Sprachkonstrukte sind in ähnlichen Situationen auch ähnlich anwendbar
- 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

- Sicherheit
  - Keine Anfrage, die syntaktisch korrekt ist, darf in eine Endlosschleife geraten oder ein unendliches Ergebnis liefern.
- Eingeschränktheit
  - Anfragesprache darf keine komplette Programmiersprache sein
    - Aber: SQL Standard besteht aus > 1300 Seiten…
  - Folgt aus Sicherheit, Optimierbarkeit, Effizienz
- Vollständigkeit
  - Sprache muss mindestens die Anfragen einer Standardsprache (z.B. relationale Algebra) ausdrücken können.

Anfragealgebra

- 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 Datenbankanfragen
  - 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

- 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

Überblick

- Einführung
- Basisoperatoren
- Operatoren auf Multimengen
- Erweiterte Operatoren

Klassifikation der Operatoren

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

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

Vereinigung (union, $\cup$)

- 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

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, −, \)

- 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

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

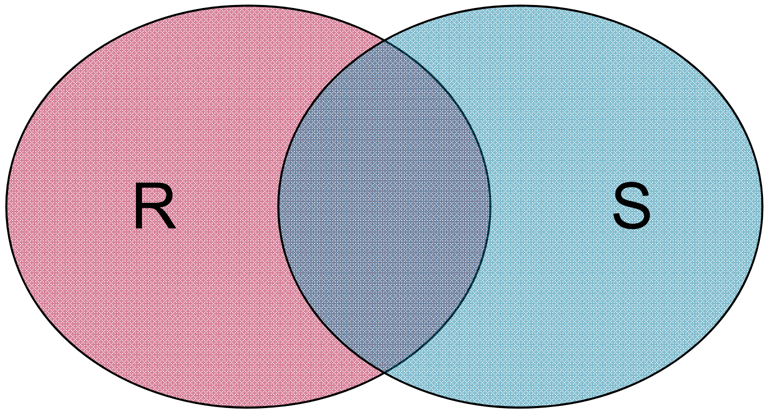
R - S

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

Schnittmenge (intersection, $\cap$)

- 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"
    - $R \cap S = R - (R - S)$
    - $ = S - (S - R)$



Beispiel für Mengenoperatoren

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
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99

Projektion (projection, π)

- Unärer Operator
- Erzeugt neue Relation mit einer Teilmenge der ursprünglichen Attribute
- $\pi_{A1,A2,\ldots,Ak}(R)$ ist eine Relation
  - mit den Attributen A1,A2,…,Ak
  - Üblicherweise in der aufgelisteten Reihenfolge
- Achtung: Es können Duplikate entstehen, die entfernt werden müssen.

Projektion – Beispiel

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

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

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

π_{inFarbe}**(Filme)**

inFarbe
True
False

Selektion (selection, σ)

- Unärer Operator
- 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*.
    - const = const (unnötig)
    - attr = const (typische Selektion)
    - attr = attr (join condition)
- Prüfe Bedingung für jedes Tupel

Selektion – Beispiel

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

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

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

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

Kartesisches Produkt
(*Cartesian product, cross product* $\times$)

- Binärer Operator
- 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

http://de.wikipedia.org/wiki/Ren%C3%A9_Descartes

Kartesisches Produkt – Beispiel

R

A	B
1	2
3	4

S

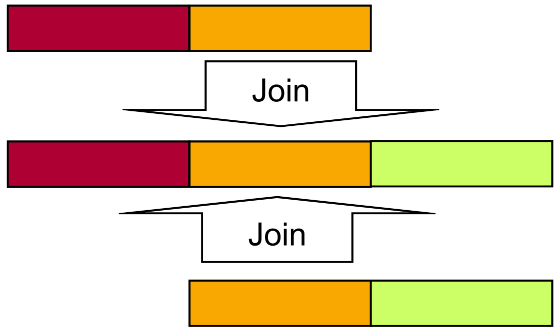
B	C	D
2	5	6
4	7	8
9	10	11

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, ⋈)

- Binärer Operator
- 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

R

A	B
1	2
3	4

S

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

$R \bowtie S$

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

$R \times S$

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

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

$R \bowtie S$

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, $\bowtie_\theta$)

- Verallgemeinerung des natürlichen Joins
- Verknüpfungsbedingung kann selbst gestaltet werden.
- Konstruktion des Ergebnisses:
  - Bilde Kreuzprodukt
  - Selektiere mittels der Joinbedingung
  - Also: $R \bowtie_\theta S = \sigma_\theta (R \times S)$
  - $\theta \in \{=, <, >, \leq, \geq, \neq\}$
- Schema: Wie beim Kreuzprodukt
- Natural Join ist ein Spezialfall des Theta-Joins
  - Aber: Schema des Ergebnisses sieht anders aus.
  - $R(A,B,C) \bowtie S(B,C,D)$
    $= \rho_{T(A,B,C,D)}(\pi_{A,R.B,R.C,D}(\sigma_{(R.B=S.B \text{ AND } \dots \text{ AND } R.C = S.C)} (R \times S)))$

Theta-Join – Beispiel

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

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

Filme

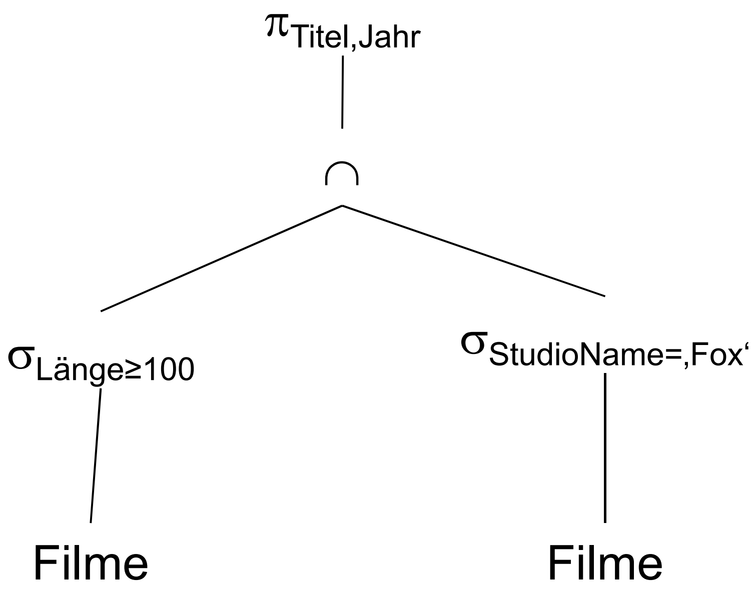
Titel	Jahr	Länge	Typ	StudioName
Total Recall	1990	113	Farbe	Fox
Basic Instinct	1992	127	Farbe	Disney
Dead Man	1995	90	s/w	Paramount

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

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



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

Komplexe Ausdrücke – Beispiel

Filme

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

Rollen

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

Gesucht: Namen aller Schauspieler, die in Filmen spielten, die mindestens 100 Minuten lang sind.

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

Umbenennung (rename, ρ)

Unärer Operator

Motivation: Zur Kontrolle der Schemata und einfacheren Verknüpfungen

- $\rho_{S(A1,\ldots,An)}(R)$
  - Benennt Relation R in S um
  - Benennt die Attribute der neuen Relation $A_1,\ldots,A_n$
- $\rho_S(R)$ benennt nur Relation um.

Durch Umbenennung ermöglicht

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

Umbenennung – Beispiel

R	
A	**B**
1	2
3	4

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

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

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

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

Vorschau zu Optimierung

Beispiele für algebraische Regeln zur Transformation

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

Jeweils: Welche Seite ist besser?

Überblick

- Einführung
- Basisoperatoren
- Operatoren auf Multimengen
- Erweiterte Operatoren

Motivation

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

A	B
1	2
3	4
1	2
1	2

Multimenge

Reihenfolge ist weiter unwichtig

Effizienz durch Multimengen

Projektion auf (A,B)

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

- 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) = ?

Vereinigung auf Multimengen

- 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
  - (*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 ∪ S

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

Schnittmenge auf Multimengen

- 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
  - 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 ∩ S

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

Differenz auf Multimengen

- 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* öfter in R als in S vorkommt, bleiben *n*-*m* *t* übrig.
  - Falls *t* öfter 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

- 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

- 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 × S *n*·*m*-mal.

R

A	B
1	2
1	2

S

B	C
2	3
4	5
4	5

R × S

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

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

Überblick

- Einführung
- Basisoperatoren
- Operatoren auf Multimengen
- Erweiterte Operatoren

Überblick über Erweiterungen

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

Duplikateliminierung (duplicate elimination, δ)

Wandelt eine Multimenge in eine Menge um.

- Durch Löschen aller Kopien von Tupeln
- δ(R)

R

A	B
1	2
3	4
1	2
1	2

δ(R)

A	B
1	2
3	4

Aggregation

- 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
- COUNT(B) = 4

Gruppierung

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:
  - Gruppiere nach StudioName
  - Summiere **in jeder Gruppe** die Länge der Filme

Gruppierung (group, γ)

- $\gamma_L(R)$ 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

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

Sortierung (sort, τ)

- $\tau_L(R)$ wobei L eine Attributliste aus R ist.
  - Falls $L = (A_1,A_2,\ldots,A_n)$ 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: In DBMS macht es manchmal auch Sinn zwischendurch zu sortieren.

Erweiterte Projektion

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→Y wobei X ein Attribut in R ist und Y ein neuer Name ist.
  - Ein Ausdruck E→Z, wobei E ein Ausdruck mit Konstanten, arithmetischen Operatoren, Attributen von R und String-Operationen ist, und Z ein neuer Name ist.
    - A1+A2 → Summe
    - Vorname || Nachname → Name

Semi-Join ($\rtimes$)

- Formal
  - R(A), S(B)
  - $R \ltimes S := \pi_A(R \bowtie_F S)$

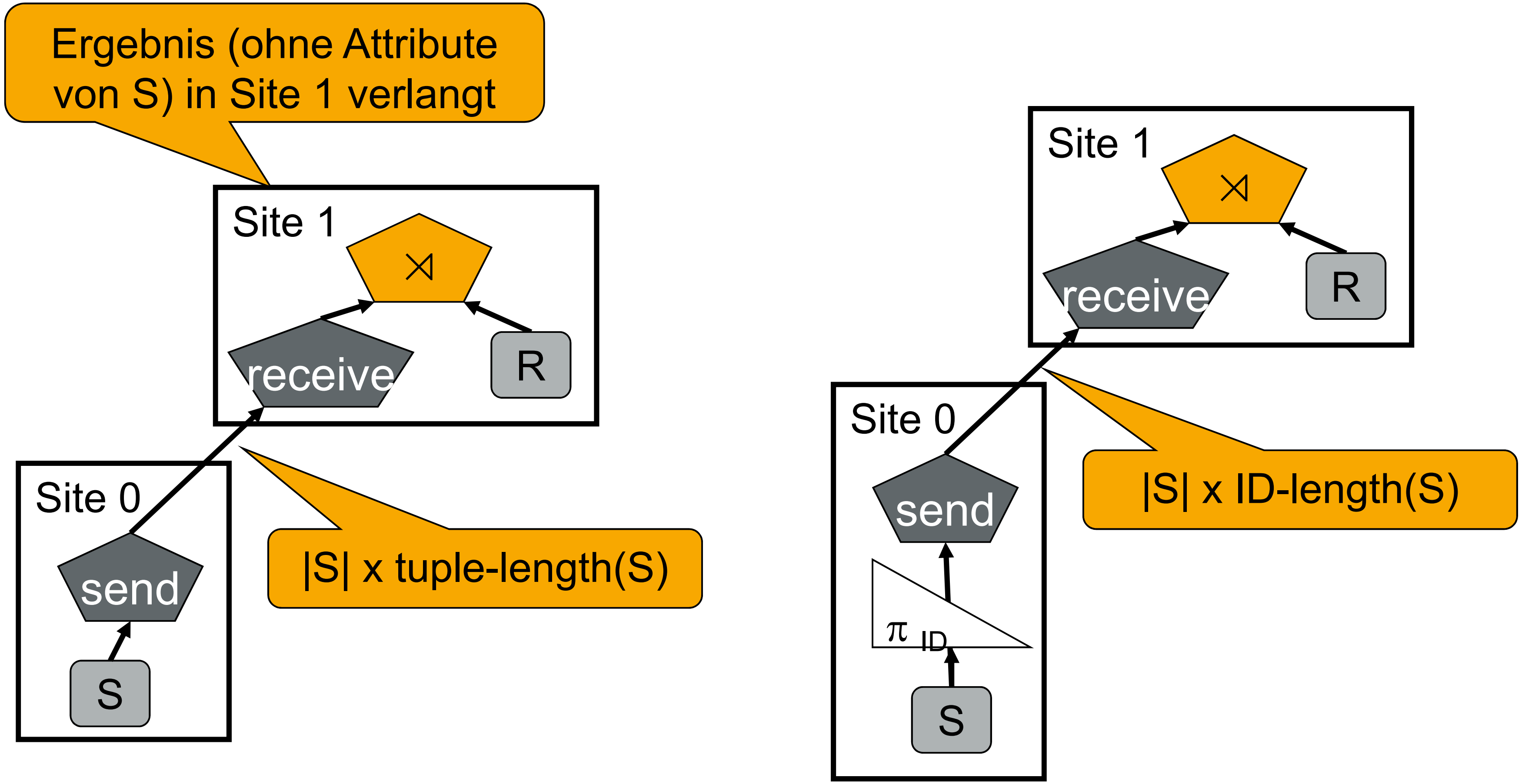
$$= \pi_A(R) \bowtie_F \pi_{A \cap B}(S)$$

$$= R \bowtie_F \pi_{A \cap B}(S)$$

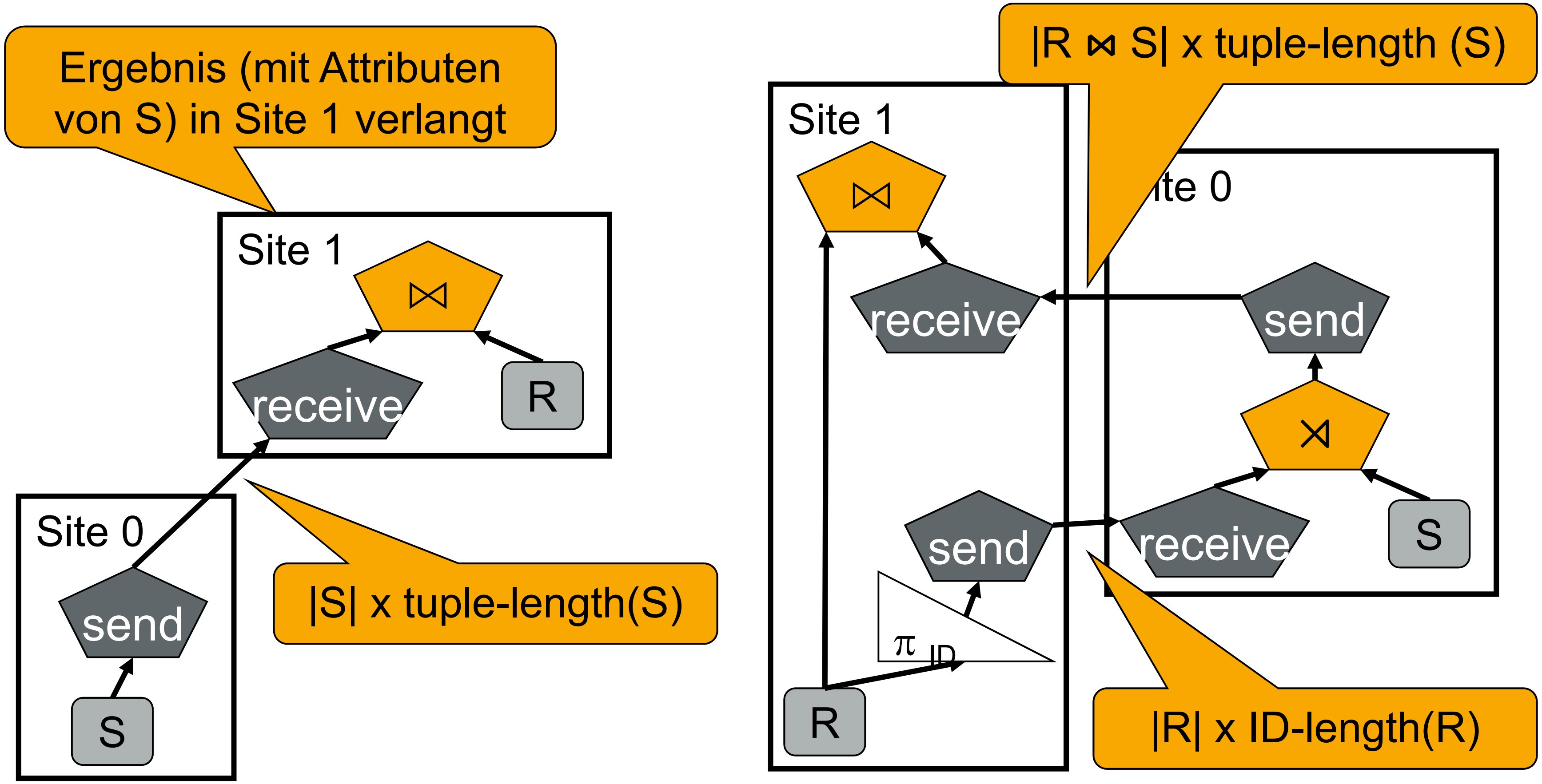
$$\text{i.d.R.} = R \bowtie_F \pi_F(S)$$

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

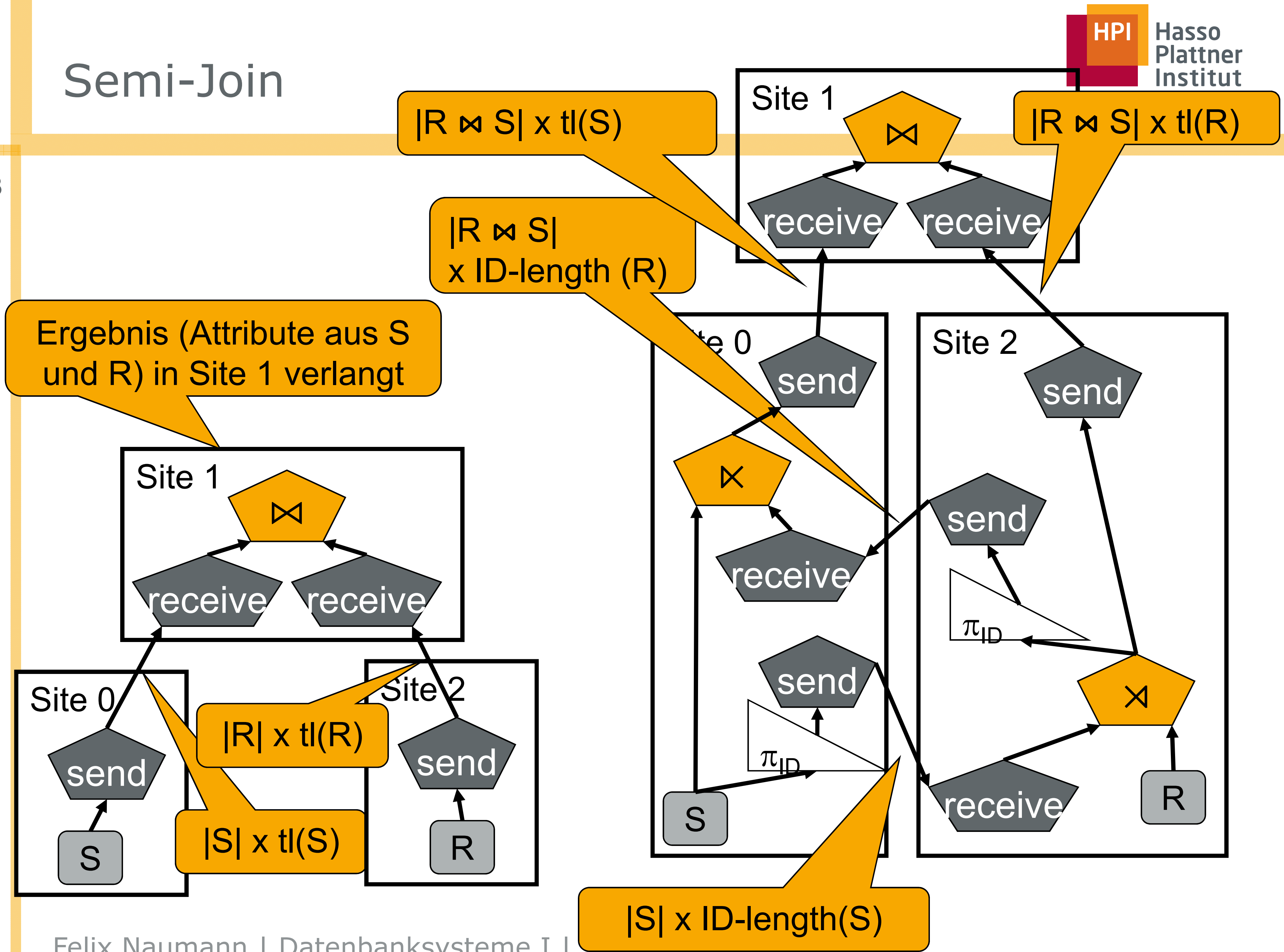
Semi-Join


Ergebnis (ohne Attribute von S) in Site 1 verlangt
Site 1
⋊
receive
R
Site 0
send
S
|S| x tuple-length(S)
Site 1
⋊
receive
R
Site 0
send
π ID
S
|S| x ID-length(S)

Semi-Join


Ergebnis (mit Attributen von S) in Site 1 verlangt
Site 1
⋈
receive
R
Site 0
send
S
|S| x tuple-length(S)
|R ⋈ S| x tuple-length (S)
Site 1
⋈
receive
send
π ID
R
ite 0
send
⋊
receive
S
|R| x ID-length(R)

Semi-Join



|R ⋈ S| x tl(S)
Site 1
⋈
|R ⋈ S| x tl(R)
receive
receive
|R ⋈ S|
x ID-length (R)
Ergebnis (Attribute aus S
und R) in Site 1 verlangt
Site 0
send
⋉
receive
send
π_{ID}
S
Site 2
send
send
π_{ID}
⋊
receive
R
Site 1
⋈
receive
receive
Site 0
send
S
|R| x tl(R)
|S| x tl(S)
Site 2
send
R
|S| x ID-length(S)

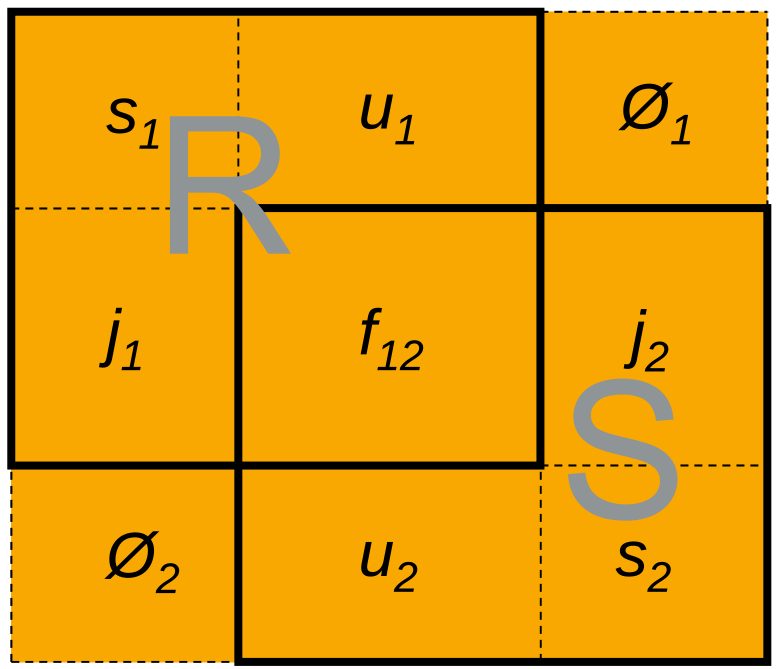
Outer Joins (Äußere Verbünde, |⋈|)

- Übernahme von „*dangling tuples*" in das Ergebnis und Auffüllen mit Nullwerten (*padding*)
  - Nullwert: ⊥ (≠ 0)
- Full outer join
  - Übernimmt alle Tupel beider Operanden
  - R |⋈| S
- Left outer join (right outer join)
  - Übernimmt alle Tupel des linken (rechten) Operanden
  - R |⋈ S (bzw. R ⋈| S)

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

Outer Joins

- R ⋈ S
- R |⋈ S
- R ⋈| S
- R |⋈| S



Outer Joins

LINKS

A	B
1	2
2	3

RECHTS

B	C
3	4
4	5

$\bowtie$

A	B	C
2	3	4

$⟗$

A	B	C
1	2	$\perp$
2	3	4
$\perp$	4	5

$⟕$

A	B	C
1	2	$\perp$
2	3	4

$⟖$

A	B	C
2	3	4
$\perp$	4	5

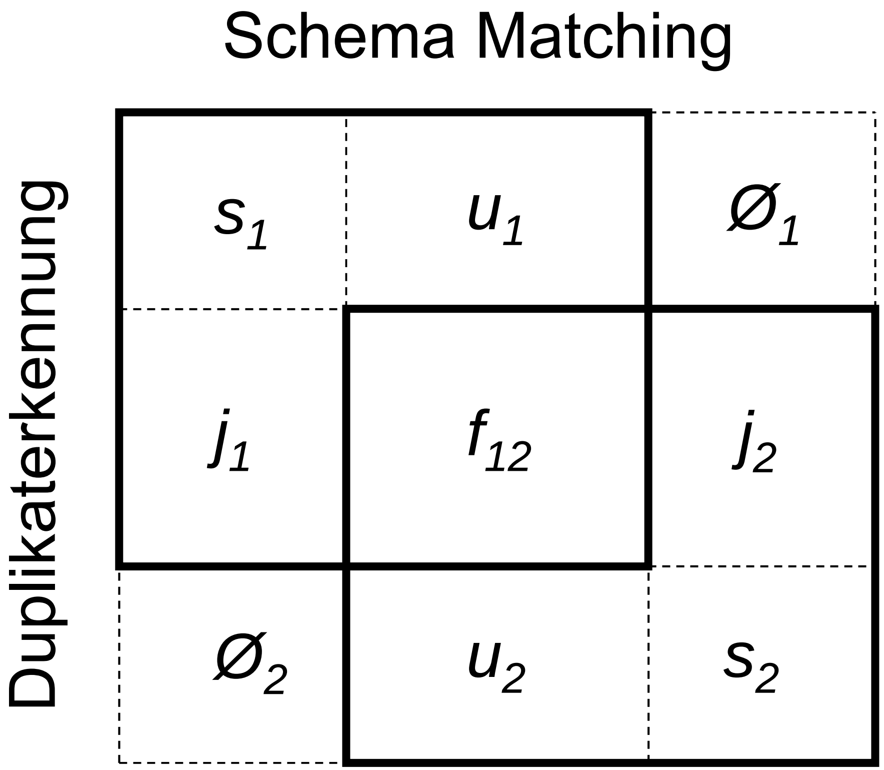
Outer Joins und Informationsintegration

Ziel: Möglichst viele Informationen

- Viele Tupel
- Viele Attribute

Problem

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



Outer Union (⊎)

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 ⊎ 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, /)

- 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

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

A

pno
p2

B1

pno
p2
p4

B2

pno
p1
p2
p4

B3

sno
s1
s2
s3
s4

A/B1

sno
s1
s4

A/B2

sno
s1

A/B3

Division ausdrücken

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

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

A

pno
p2

B1

pno
p2
p4

B2

pno
p1
p2
p4

B3

sno
s1
s2
s3
s4

A/B1

sno
s1
s4

A/B2

sno
s1

A/B3

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

Zusammenfassung

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

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