

# 184.686 VU Datenbanksysteme

## Relationales Modell und relationale Algebra

Katja Hose

Institut für Logic and Computation

Sommersemester 2024



Informatics

# Lernziele

## Ziele

- Erklären des relationalen Modells
- Erstellen von nicht-trivialen Anfragen der relationalen Algebra
- Verwenden von verschiedenen Join-Typen in Ausdrücken der relationalen Algebra
- Erklären der Einschränkungen der relationalen Algebra

## Motivation

- SQL-Datenmodell ähnelt dem relationalen Modell
- SQL basiert teilweise auf der relationalen Algebra
- Die relationale Algebra hilft Ausführungspläne und Anfrageoptimierung zu verstehen

- 1 Das relationale Modell
  - Grundbegriffe
  - Eigenschaften
- 2 Relationale Algebra
  - Übersicht
  - Basisoperatoren
  - Natürlicher Verbund (Natural Join)
  - Zusätzliche Joinvarianten
  - Gruppierung und Aggregation
  - Relationale Division
  - Operatorbaumdarstellung

# Grundlagen des relationalen Modells

Seien  $D_1, D_2, \dots, D_n$  **Domänen** (=Wertebereiche)

## Relation

$$R \subseteq D_1 \times \dots \times D_n$$

- Beispiel:  $telephoneBook \subseteq string \times string \times integer$
- Wertebereiche dürfen identisch sein:  $D_i = D_j$  für  $i \neq j$
- Basierend auf Mengen

## Relationenschema

- legt die Struktur der gespeicherten Daten fest
- wird mit  $sch(R)$  oder  $\mathcal{R}$  bezeichnet
- Notation:  $R(A_1 : D_1, A_2 : D_2, \dots)$  mit  $A_i$  für Attribute
- Beispiel:

$telephoneBook(name : string, street : string, \underline{phoneNumber : integer})$

# Zulässige Relationen

Gegeben

- Domänen  $D_1 = \{1, 2, 3\}$ ,  $D_2 = \{x, y, z\}$ ,  $D_3 = \{\alpha, \beta, \gamma, \omega\}$
- Attribute auf diesen Domänen  $A : D_1$ ,  $B : D_2$ ,  $C : D_3$

Welche der folgenden Relationen sind zulässig?

A	B	C
1	y	$\alpha$
3	x	$\alpha$
3	y	$\alpha$

(a)

A	B	C
$\alpha$	y	$\alpha$
3	x	$\alpha$
2	y	$\beta$

(b)

A	B	C
1	y	$\alpha$
3	x	$\alpha$
1	y	$\alpha$

(c)

# Zulässige Relationen

Gegeben

- Domänen  $D_1 = \{1, 2, 3\}$ ,  $D_2 = \{x, y, z\}$ ,  $D_3 = \{\alpha, \beta, \gamma, \omega\}$
- Attribute auf diesen Domänen  $A : D_1$ ,  $B : D_2$ ,  $C : D_3$

Welche der folgenden Relationen sind zulässig?

A	B	C
1	y	$\alpha$
3	x	$\alpha$
3	y	$\alpha$

zulässig

A	B	C
$\alpha$	y	$\alpha$
3	x	$\alpha$
2	y	$\beta$

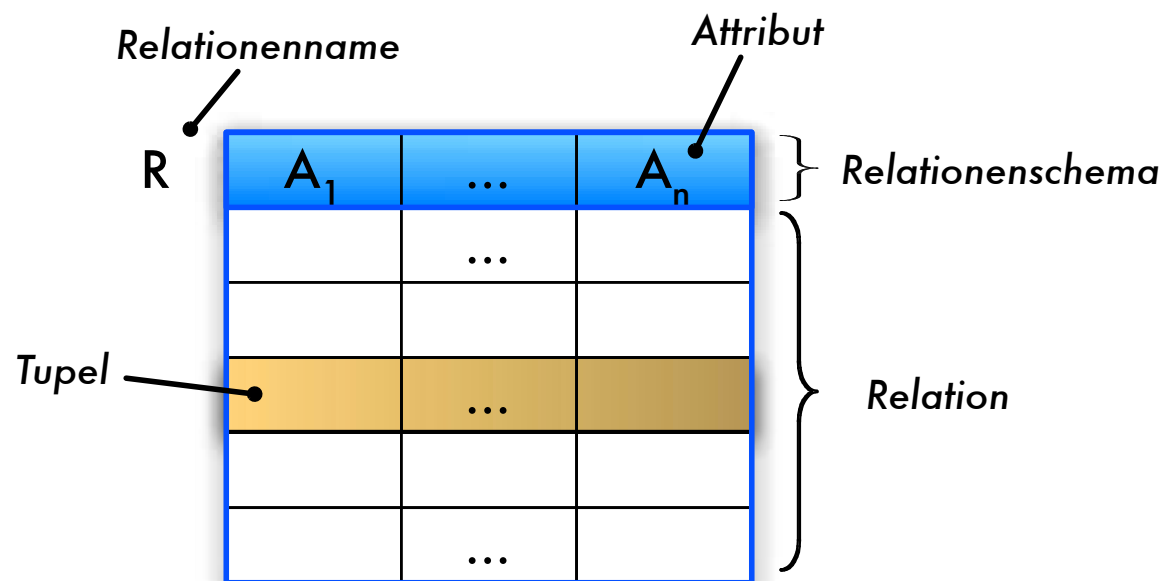
nicht zulässig

A	B	C
1	y	$\alpha$
3	x	$\alpha$
1	y	$\alpha$

nicht zulässig

# Veranschaulichung der Grundbegriffe

- Fettgeschriebene Kopfzeile: **Relationenschema**
- Spaltenüberschrift: **Attribut**
- Weitere Einträge in der Tabelle: **Relation**
- Eine Zeile der Tabelle: **Tupel**
- Ein Eintrag: **Attributwert**
- Unterstrichenes Attribut: **Primärschlüssel**



## Theorie vs. Realität

In der Realität meist keine Unterscheidung zwischen Ausprägung (Instanz) und Schema einer Relation.

# Relationenmodell

wines	<u>wineID</u>	name	color	year	vineyard → producer
	1042	La Rose Grand Cru	red	1998	Château La Rose
	2168	Creek Shiraz	red	2003	Creek
	3456	Zinfandel	red	2004	Helena
	2171	Pinot Noir	red	2001	Creek
	3478	Pinot Noir	red	1999	Helena

producer	<u>vineyard</u>	area	region
	Creek	Barossa Valley	South Australia
	Helena	Napa Valley	California
	Château La Rose	Saint-Emilion	Bordeaux
	Château La Pointe	Pomerol	Bordeaux

## Fremdschlüssel

- Eine Relation kann die Primärschlüsselattribute einer anderen Tabelle beinhalten.
- Werte der Fremdschlüsselattribute müssen im Primärschlüssel der referenzierten Tabelle vorkommen.



# Tupelreihenfolge

Tupel einer Relation haben keine **Reihenfolge**.

name	age
Pat	1
Fred	2
Sue	3
Pam	4

name	age
Sue	3
Pam	4
Fred	2
Pat	1

Diese Relationen beinhalten dieselben Informationen.

# Attributreihenfolge

Die **mathematische Definition** von Tupeln sieht eine **bestimmte Reihenfolge** der Attribute des Tuples/der Relation vor.

name	age
Pat	1
Fred	2
Sue	3
Pam	4

age	name
1	Pat
2	Fred
3	Sue
4	Pam

Diese Relationen beinhalten unterschiedliche Informationen.

# Attributreihenfolge

Die **mathematische Definition** von Tupeln sieht eine **bestimmte Reihenfolge** der Attribute des Tuples/der Relation vor.

name	age
Pat	1
Fred	2
Sue	3
Pam	4

age	name
1	Pat
2	Fred
3	Sue
4	Pam

## Aber...

- Die Reihenfolge der Attribute ist in den meisten Anwendungen bedeutungslos.
- Das Verwenden von Attributnamen statt einer bestimmten Reihenfolge ist praktischer.
- Das kartesische Produkt wird kommutativ.

# Atomare Werte

- Werte eines Tupels sind **atomar** (unteilbar).
- Ein Wert kann kein zusammengesetzter Datentyp (Liste, Array, ...) oder eine Relation sein.

name	age
Pat	1
Fred	2
Sue	3
Pam	4

Alle Werte sind atomar

name				age
	Pat		Jensen	1
	Fred	D.	Roosevelt	2
	Sue	H.M.I.	Knuth	3
	Pam	C.	Anderson	4

name ist nicht atomar

# Null Werte

Ein spezieller **Null** Wert wird verwendet um unbekannte bzw. für gewisse Tupel unanwendbare Werte zu repräsentieren

name	age
Pat	1
Fred	2
Sue	null
Pam	null

name	age
Pat	1
Fred	2
Sue	⊥
Pam	⊥

Alternative Notation

# Duplikate

Eine Relation folgt der **mathematischen Definition einer Menge**.

name	age
Pat	1
Fred	2
Sue	3
Pam	4

zulässig

name	age
Pat	1
Fred	2
Sue	3
Sue	3

unzulässig

Keine zwei Tupel einer Relation dürfen identische Werte in allen Attributen beinhalten.

# Zulässige Relationen

Welche der folgenden Relationen sind zulässig?

name	age
Pat	1
Fred	2
Sue	⊥
Sue	3

(a)

name	age
Pat	1
Fred	2
Sue	⊥
Sue	⊥

(b)

# Zulässige Relationen

Welche der folgenden Relationen sind zulässig?

name	age
Pat	1
Fred	2
Sue	⊥
Sue	3

gültig

name	age
Pat	1
Fred	2
Sue	⊥
Sue	⊥

gültig



- 1 Das relationale Modell
  - Grundbegriffe
  - Eigenschaften
- 2 Relationale Algebra
  - Übersicht
  - Basisoperatoren
  - Natürlicher Verbund (Natural Join)
  - Zusätzliche Joinvarianten
  - Gruppierung und Aggregation
  - Relationale Division
  - Operatorbaumdarstellung

# Die Operatoren der relationalen Algebra

- Projektion  $\pi$
- Selektion  $\sigma$
- Umbenennung  $\rho$
- Kreuzprodukt  $\times$
- Vereinigung  $\cup$
- Differenz  $-$
- Schnitt  $\cap$
- Join (Verbund)  $\bowtie$
- Linker äußerer Join  $\leftarrow \bowtie$
- Rechter äußerer Join  $\bowtie \rightarrow$
- Äußerer Join  $\leftarrow \bowtie \rightarrow$
- Semi-Join (linker)  $\ltimes$
- Semi-Join (rechter)  $\rtimes$
- Gruppierung  $\gamma$
- Division  $\div$

# Basisoperatoren

- **Projektion**  $\pi$
- **Selektion**  $\sigma$
- **Umbenennung**  $\rho$
- **Kreuzprodukt**  $\times$
- **Vereinigung**  $\cup$
- **Differenz**  $-$

## Basisoperatoren

Jede Anfrage in relationaler Algebra kann ausschließlich mit Basisoperatoren ausgedrückt werden.

Das Weglassen eines Basisoperators verringert die Ausdruckskraft.

## Unäre vs. binäre Operatoren

**Unäre Operatoren:**  $\sigma, \pi, \rho$

**Binäre Operatoren:**  $\times, \cup, -$

# Auswertung der Operatoren

## Operatoren und deren Verwendung

- Input: eine oder mehrere Relationen
- Output: eine Relation

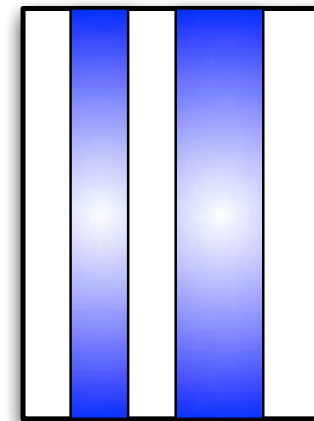
Operatoren können (nach bestimmten Regeln) kombiniert werden.

# Projektion

 $\pi_{name, dept\_name}(instructor)$ 

instructor			
ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000

Das Resultat ist eine Relation mit  $n$  Spalten, die durch das Weglassen der nicht angegeben Spalten entsteht.

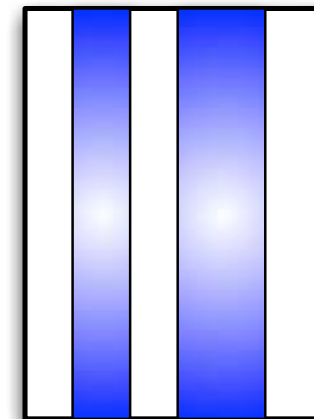


# Projektion

 $\pi_{name, dep\_name}(instructor)$ 

$\pi_{name, dep\_name}(instructor)$	
name	dep_name
Srinivasan	Comp. Sci.
Wu	Finance
Mozart	Music
Einstein	Physics

Das Resultat ist eine Relation mit  $n$  Spalten, die durch das Weglassen der nicht angegebenen Spalten entsteht.



# Erweiterte Projektion

$$\pi_{ID, name, dept\_name, salary \div 12}(instructor)$$

instructor			
ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000

# Erweiterte Projektion

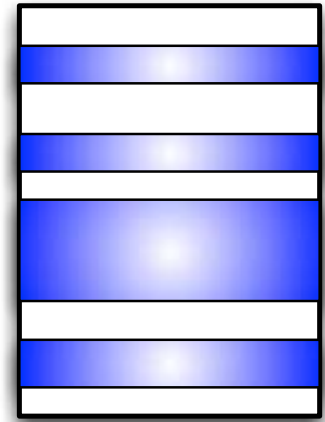
$$\pi_{ID, name, dept\_name, salary \div 12}(instructor)$$

instructor			
ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	5417
12121	Wu	Finance	7500
15151	Mozart	Music	3333
22222	Einstein	Physics	7917
32343	El Said	History	5000
33456	Gold	Physics	7250



# Selektion

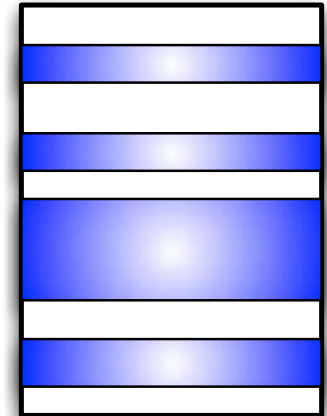
- Symbol:  $\sigma_F$
- Selektionsprädikat  $F$  besteht aus
  - Logischen Operatoren:  $\vee$  (oder),  $\wedge$  (und),  $\neg$  (nicht)
  - Arithmetischen Vergleichsoperatoren:  $<$ ,  $\leq$ ,  $=$ ,  $>$ ,  $\geq$ ,  $\neq$
  - ... und natürlich aus Attributnamen der Argumentrelation oder Konstanten als Operanden



Auswahl von Zeilen einer Tabelle anhand eines Selektionsprädikats

# Selektion

- Symbol:  $\sigma_F$
- Selektionsprädikat  $F$  besteht aus
  - Logischen Operatoren:  $\vee$  (oder),  $\wedge$  (und),  $\neg$  (nicht)
  - Arithmetischen Vergleichsoperatoren:  $<$ ,  $\leq$ ,  $=$ ,  $>$ ,  $\geq$ ,  $\neq$
  - ... und natürlich aus Attributnamen der Argumentrelation oder Konstanten als Operanden

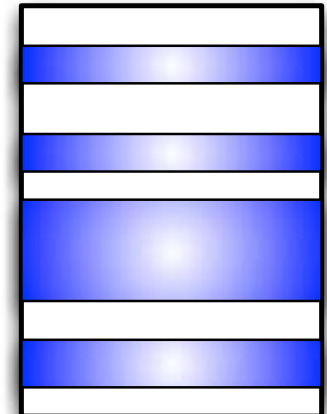


$\sigma_{salary > 80000}(instructor)$

instructor			
ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000

# Selektion

- Symbol:  $\sigma_F$
- Selektionsprädikat  $F$  besteht aus
  - Logischen Operatoren:  $\vee$  (oder),  $\wedge$  (und),  $\neg$  (nicht)
  - Arithmetischen Vergleichsoperatoren:  $<$ ,  $\leq$ ,  $=$ ,  $>$ ,  $\geq$ ,  $\neq$
  - ... und natürlich aus Attributnamen der Argumentrelation oder Konstanten als Operanden



$\sigma_{salary > 80000}(instructor)$

$\sigma_{salary > 80000}(instructor)$			
ID	name	dept_name	salary
12121	Wu	Finance	90000
22222	Einstein	Physics	95000
33456	Gold	Physics	87000

# Umbenennung

## Umbenennung einer Relation

$$\rho_S(R)$$

Relation  $R$  bekommt den neuen Namen  $S$ .

## Umbenennung von Spalten

$$\rho_{A \leftarrow B}(R)$$

Umbenennung von Attribut  $B$  in  $A$

In der Literatur wird oftmals auch das Symbol  $\beta$  für die Umbenennung verwendet.

Unterschiedliche Notationen in verschiedenen Lehrbüchern

# Kartesisches Produkt (Kreuzprodukt)

## Kreuzprodukt

Das Kreuzprodukt ( $R \times S$ ) zweier Relationen  $R$  und  $S$  besteht aus allen möglichen Kombinationen ( $|R| * |S|$  Paare) aus je einem Tupel der beiden Relationen.

Schema des Resultats:

$$sch(R \times S) = sch(R) \cup sch(S) = \mathcal{R} \cup \mathcal{S}$$

Enthält viele (oft auch viele unsinnige) Kombinationen!

Beim Referenzieren der Attribute des resultierenden Schemas wird  $R.A$  und  $S.A$  verwendet, insbesondere bei Überlappungen der einzelnen Schemata (Vermeidung von Unklarheiten, was gemeint ist).

# Was wird hier berechnet?

$$\pi_{V1.course}(\sigma_{V2.successor=5216 \wedge V1.successor=V2.course}(\rho_{V1}(prerequisite) \times \rho_{V2}(prerequisite)))$$

prerequisite	
course	successor
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

V1.course	V1.successor	V2.course	V2.successor
5001	5041	5001	5041
5001	5041	5001	5043
5001	5041	5001	5049
5001	5041	5041	5216
5001	5041	5043	5052
...	...	...	...
5001	5043	5001	5041
...	...	...	...
5052	5259	5052	5259

# Was wird hier berechnet?

$$\pi_{V1.course}(\sigma_{V2.successor=5216 \wedge V1.successor=V2.course}(\rho_{V1}(prerequisite) \times \rho_{V2}(prerequisite)))$$

prerequisite	
course	successor
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

V1.course	V1.successor	V2.course	V2.successor
5001	5041	5041	5216

V1.course
5001

Vorgänger zweiter Stufe der Vorlesung mit Nummer 5216

# Mengenoperationen

Die Mengenoperationen **Vereinigung**, **Schnitt**, und **Differenz** können auch auf Relationen angewendet werden.

## Voraussetzung

Beide beteiligten Relationen müssen **vereinigungskompatibel** sein:

- sie haben die gleiche Anzahl an Attributen.
- Der Wertebereich jedes Attributs in Spaltenreihenfolge ist für beide Relationen gleich.



# Vereinigung

## Vereinigung

Die Vereinigung ( $R \cup S$ ) von zwei Relationen  $R$  und  $S$  sammelt die Tupelmengen der beiden Relationen ohne Duplikate.

## Beispiel

$$\pi_{name, dep\_name}(instructor) \cup \pi_{name, dep\_name}(student)$$

# Differenz

## Differenz

Die Differenz ( $R - S$  bzw.  $R \setminus S$ ) von zwei Relationen  $R$  und  $S$  eliminiert die Tupel aus der ersten Relation, die auch in der zweiten Relation vorkommen.

# Differenz

instructor_departments
dept_name
Comp. Sci.
Music
History
Biology
Elec. Eng.

student_departments
dept_name
Comp. Sci.
History
Finance
Physics
Music

instructor\_departments – student\_departments

dept_name
Biology
Elec. Eng.

# Überblick über die Basisoperatoren

Ausdruck	Schema	Arität	Min. Kardinalität	Max. Kardinalität
$\sigma_F(R)$	$\mathcal{R}$	$ \mathcal{R} $	0	$ R $
$\pi_L(R)$	$L$	$\leq  \mathcal{R} $	0	$ R $
$R \cup S$	$\mathcal{R} (=S)$	$ \mathcal{R}  (= S )$	$\max( R ,  S )$	$ R  +  S $
$R - S$	$\mathcal{R} (=S)$	$ \mathcal{R}  (= S )$	0	$ R $
$R \times S$	$\mathcal{R} \circ S$	$ \mathcal{R}  +  S $	$ R  \cdot  S $	$ R  \cdot  S $
$\rho_S(R)$	$\mathcal{R}$	$ \mathcal{R} $	$ R $	$ R $

# Schnitt

Der Schnitt ist kein Basisoperator!

## (Durch-)schnitt

Der Schnitt ( $R \cap S$ ) von zwei Relationen  $R$  und  $S$  besteht aus der Menge der Tupel, die in beiden Relationen gemeinsam vorkommen.

Der Schnitt kann als Differenz ausgedrückt werden ( $R \cap S = R - (R - S)$ )

# Schnitt

instructor_departments
dept_name
Comp. Sci.
Music
History
Biology
Elec. Eng.

student_departments
dept_name
Comp. Sci.
History
Finance
Physics
Music

$\text{instructor\_departments} \cap \text{student\_departments}$

dept_name
Comp. Sci.
Music
History



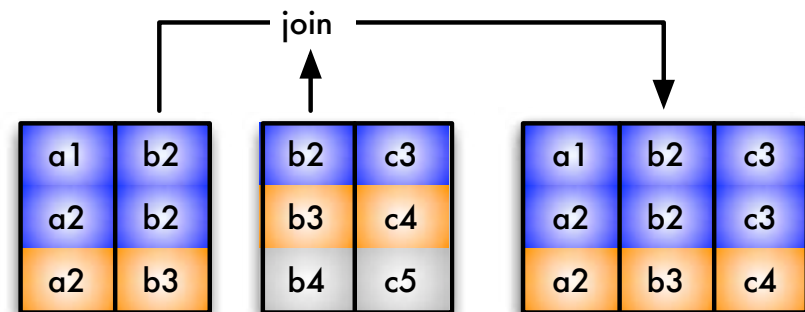
# Natürlicher Verbund (Natural Join)

## Natürlicher Verbund (Natural Join)

Der natürliche Verbund verknüpft Tabellen über **gleichbenannte Spalten**, indem jeweils zwei Tupel verschmolzen werden, falls sie dort **gleiche Werte** aufweisen.

Gegeben zwei Relationen (+ Schemata)

- $R(A_1, \dots, A_m, \mathbf{B_1}, \dots, \mathbf{B_k})$
- $S(\mathbf{B_1}, \dots, \mathbf{B_k}, C_1, \dots, C_n)$





# Natürlicher Verbund (Natural Join)

Der natürliche Join kann durch ein Kreuzprodukt gefolgt von Selektionen und Projektionen ausgedrückt werden.

Zum Beispiel:

$$R \bowtie S = \pi_{A_1, \dots, A_m, R.B_1, \dots, R.B_k, C_1, \dots, C_n} (\sigma_{R.B_1=S.B_1 \wedge \dots \wedge R.B_k=S.B_k} (R \times S))$$

$R \bowtie S$											
$\mathcal{R} - \mathcal{S}$				$\mathcal{R} \cap \mathcal{S}$				$\mathcal{S} - \mathcal{R}$			
$A_1$	$A_2$	...	$A_m$	$B_1$	$B_2$	...	$B_k$	$C_1$	$C_2$	...	$C_n$
...	...	...	...	...	...	...	...	...	...	...	...

Die Reihenfolge der Attribute wird durch die Attribute der gegebenen Relationen bestimmt, d.h. das resultierende Schema ist nicht unbedingt so wie im obigen Beispiel sortiert.

# Natürlicher Verbund (Natural Join)

wines ⋈ producer

wines	wineID	name	color	year	vineyard
	1042	La Rose Grand Cru	red	1998	Château La Rose
	2168	Creek Shiraz	red	2003	Creek
	2171	Pinot Noir	red	2001	Creek
	4711	Riesling Reserve	white	1999	Müller

producer	vineyard	area	region
	Creek	Barossa Valley	South Australia
	Helena	Napa Valley	California
	Château La Rose	Saint-Emilion	Bordeaux

Resultat:

wineID	name	...	vineyard	...	region
1042	La Rose Grand Cru	...	Ch. La Rose	...	Bordeaux
2168	Creek Shiraz	...	Creek	...	South Australia
2171	Pinot Noir	...	Creek	...	South Australia

Tupel, die keinen Partner finden (*dangling tuples*), werden “eliminiert”.

# Was wird hier berechnet?

$$\pi_{name,color,vineyard}(\sigma_{year>2000}(wines) \bowtie \sigma_{region='California'}(producer))$$

wines	wineID	name	color	year	vineyard
	1042	La Rose Grand Cru	red	1998	Château La Rose
	2168	Creek Shiraz	red	2003	Creek
	3456	Zinfandel	red	2004	Helena
	2171	Pinot Noir	red	2001	Creek
	3478	Pinot Noir	red	1999	Helena
	4711	Riesling Reserve	white	1999	Müller
	4961	Chardonnay	white	2002	Bighorn

producer	vineyard	area	region
	Creek	Barossa Valley	South Australia
	Helena	Napa Valley	California
	Château La Rose	Saint-Emilion	Bordeaux
	Château La Pointe	Pomerol	Bordeaux
	Müller	Rheingau	Hessen
	Bighorn	Napa Valley	California

# Was wird hier berechnet?

$$\pi_{name,color,vineyard}(\sigma_{year>2000}(wines) \bowtie \sigma_{region='California'}(producer))$$

wines	wineID	name	color	year	vineyard
	1042	La Rose Grand Cru	red	1998	Château La Rose
	2168	Creek Shiraz	red	2003	Creek
	3456	Zinfandel	red	2004	Helena
	2171	Pinot Noir	red	2001	Creek
	3478	Pinot Noir	red	1999	Helena
	4711	Riesling Reserve	white	1999	Müller
	4961	Chardonnay	white	2002	Bighorn

producer	vineyard	area	region
	Helena	Napa Valley	California
	Bighorn	Napa Valley	California

# Was wird hier berechnet?

$$\pi_{name,color,vineyard}(\sigma_{year>2000}(wines) \bowtie \sigma_{region='California'}(producer))$$

wines	wineID	name	color	year	vineyard
	2168	Creek Shiraz	red	2003	Creek
	3456	Zinfandel	red	2004	Helena
	2171	Pinot Noir	red	2001	Creek
	4961	Chardonnay	white	2002	Bighorn

producer	vineyard	area	region
	Helena	Napa Valley	California
	Bighorn	Napa Valley	California

Resultat:

wineID	name	...	vineyard	...	region
3456	Zinfandel	...	Helena	...	California
4961	Chardonnay	...	Bighorn	...	California

# Was wird hier berechnet?

$$\pi_{name,color,vineyard}(\sigma_{year>2000}(wines) \bowtie \sigma_{region='California'}(producer))$$

wines	wineID	name	color	year	vineyard
	2168	Creek Shiraz	red	2003	Creek
	3456	Zinfandel	red	2004	Helena
	2171	Pinot Noir	red	2001	Creek
	4961	Chardonnay	white	2002	Bighorn

producer	vineyard	area	region
	Helena	Napa Valley	California
	Bighorn	Napa Valley	California

Resultat:

name	color	vineyard
Zinfandel	red	Helena
Chardonnay	white	Bighorn

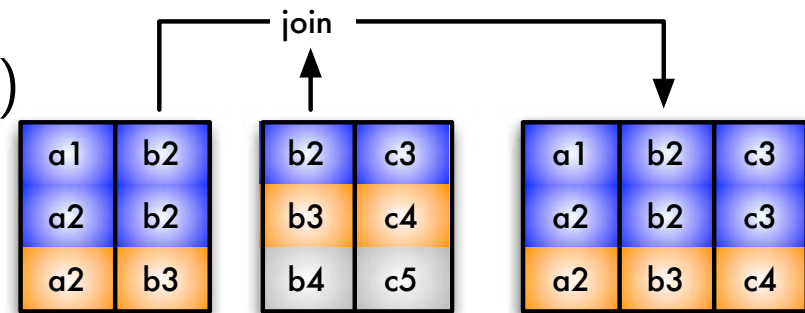
# Join Kommutativität

Wahr oder falsch?

$$R \bowtie S = S \bowtie R$$

Gegeben seien zwei Relationen (+ Schema)

- $R(A_1, \dots, A_m, \mathbf{B}_1, \dots, \mathbf{B}_k)$
- $S(\mathbf{B}_1, \dots, \mathbf{B}_k, C_1, \dots, C_n)$



Die Reihenfolge beeinflusst nicht “wirklich” das Resultat:  
Die Reihenfolge der Attribute ist unterschiedlich, aber der “Inhalt” der  
Tupel und Relationen ist “gleich”

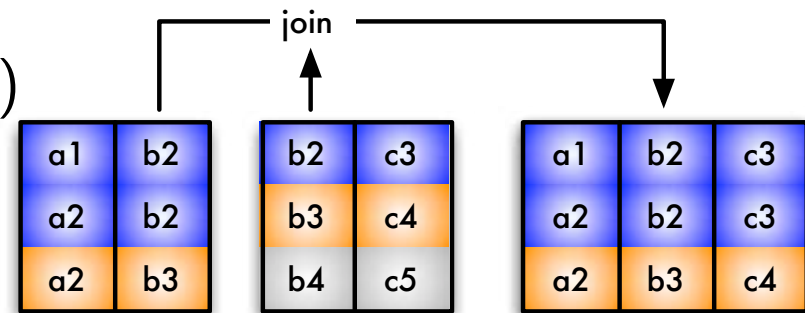
# Join Kommutativität

Wahr oder falsch?

$$R \bowtie S = S \bowtie R$$

Gegeben seien zwei Relationen (+ Schema)

- $R(A_1, \dots, A_m, \mathbf{B}_1, \dots, \mathbf{B}_k)$
- $S(\mathbf{B}_1, \dots, \mathbf{B}_k, C_1, \dots, C_n)$



Momentan betrachten wir Joins und Kreuzprodukte nicht als kommutativ.

Bei der Anfrageoptimierung (später in dieser LVA), betrachten wir aber natürliche Joins, sowie Kreuzprodukte und andere Join Varianten, als kommutativ



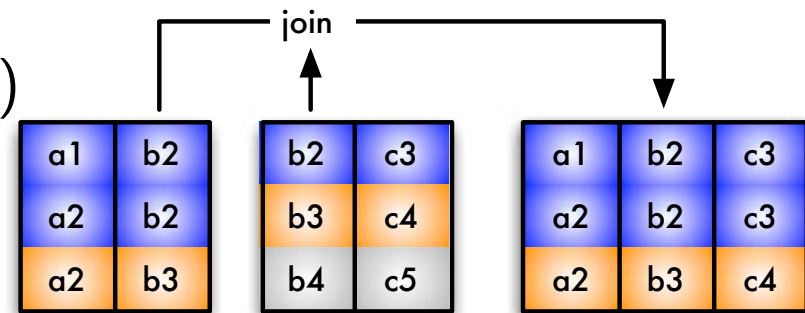
# Join Kommutativität

Wahr oder falsch?

$$R \bowtie S = S \bowtie R$$

Gegeben seien zwei Relationen (+ Schema)

- $R(A_1, \dots, A_m, \mathbf{B}_1, \dots, \mathbf{B}_k)$
- $S(\mathbf{B}_1, \dots, \mathbf{B}_k, C_1, \dots, C_n)$



Wenn wir die mathematische Definition von Tupeln einhalten und Joins trotzdem als kommutativ betrachten möchten, müssen wir eine Projektion vornehmen, um die Attribute entsprechend zu sortieren:

$$\pi_L(R \bowtie S) = \pi_L(S \bowtie R)$$

# Die Operatoren der relationalen Algebra

- Projektion  $\pi$
- Selektion  $\sigma$
- Umbenennung  $\rho$
- Kreuzprodukt  $\times$
- Vereinigung  $\cup$
- Differenz  $-$
- Schnitt  $\cap$
- **Join (Verbund)**  $\bowtie$
- **Linker äußerer Join**  $\bowtie\!\!\!\!\!\lrcorner$
- **Rechter äußerer Join**  $\rceil\!\!\!\!\!\bowtie$
- **Äußerer Join**  $\bowtie\!\!\!\!\!\lrcorner\!\!\!\!\!\rceil$
- **Semi-Join (linker)**  $\ltimes$
- **Semi-Join (rechter)**  $\rtimes$
- Gruppierung  $\gamma$
- Division  $\div$

# Theta-Join ( $\theta$ -join oder allgemeiner Join)

Gegeben zwei Relationen (+ Schemata) und  $\theta$

- $R(A_1, \dots, A_n)$
- $S(B_1, \dots, B_m)$
- $\theta$  ist ein beliebiges Prädikat über den beteiligten Attributen

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

$R \bowtie_{\theta} S$							
$\mathcal{R}$				$\mathcal{S}$			
$A_1$	$A_2$	...	$A_n$	$B_1$	$B_2$	...	$B_m$
...	...	...	...	...	...	...	...

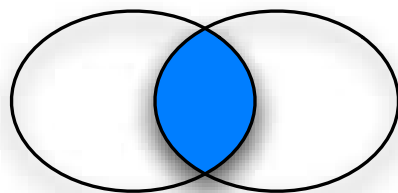
## Equi-Join

Join-Prädikat  $\theta$  prüft nur auf Gleichheit (=)

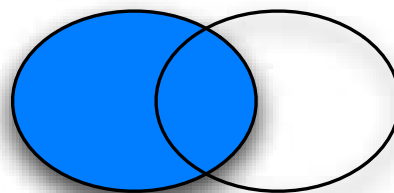
# Äußere Verbunde (Outer Joins)

Wie geht man mit Tupeln um, die keinen Joinpartner in der anderen Relation gefunden haben (“partnerlose” Tupel)?

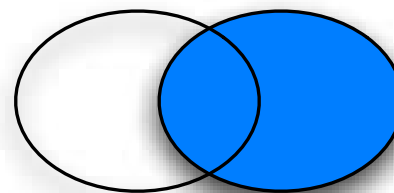
- ⌋⋈ Left outer Join (Linker äußerer Verbund):  
auch “partnerlose” Tupel der linken Relation bleiben erhalten
- ⋈⌋ Right outer Join (Rechter äußerer Verbund):  
auch “partnerlose” Tupel der rechten Relation bleiben erhalten
- ⌋⋈⌋ (Full) outer Join (Vollständiger äußerer Verbund):  
die “partnerlosen” Tupel beider Relationen bleiben erhalten



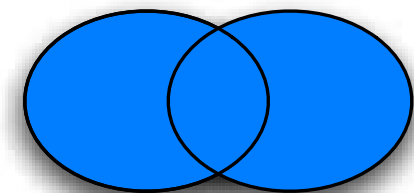
natural join



left outer join



right outer join



full outer join

# Natural Join und Left Outer Join

## Natural Join (Natürlicher Verbund)

L			R			Result				
A	B	C	C	D	E	A	B	C	D	E
$a_1$	$b_1$	$c_1$	$c_1$	$d_1$	$e_1$	$a_1$	$b_1$	$c_1$	$d_1$	$e_1$
$a_2$	$b_2$	$c_2$	$c_3$	$d_2$	$e_2$					

## Left Outer Join (Linker äußerer Verbund)

L			R			Result				
A	B	C	C	D	E	A	B	C	D	E
$a_1$	$b_1$	$c_1$	$c_1$	$d_1$	$e_1$	$a_1$	$b_1$	$c_1$	$d_1$	$e_1$
$a_2$	$b_2$	$c_2$	$c_3$	$d_2$	$e_2$	$a_2$	$b_2$	$c_2$	$\perp$	$\perp$

# Right Outer Join und (Full) Outer Join

Right Outer Join (Rechter äußerer Verbund)

L			R			Result				
A	B	C	C	D	E	A	B	C	D	E
$a_1$	$b_1$	$c_1$	$c_1$	$d_1$	$e_1$	$a_1$	$b_1$	$c_1$	$d_1$	$e_1$
$a_2$	$b_2$	$c_2$	$c_3$	$d_2$	$e_2$	$\perp$	$\perp$	$c_3$	$d_2$	$e_2$

Outer Join (Äußerer Verbund)

L			R			Result				
A	B	C	C	D	E	A	B	C	D	E
$a_1$	$b_1$	$c_1$	$c_1$	$d_1$	$e_1$	$a_1$	$b_1$	$c_1$	$d_1$	$e_1$
$a_2$	$b_2$	$c_2$	$c_3$	$d_2$	$e_2$	$a_2$	$b_2$	$c_2$	$\perp$	$\perp$
$\perp$	$\perp$	$c_3$	$d_2$	$e_2$		$\perp$	$\perp$	$c_3$	$d_2$	$e_2$

# Semi-Joins

Finde alle Tupel der linken Relation, die Joinpartner in der rechten Relation haben.

Linker Semi-Join:

$$L \bowtie R = \pi_{\mathcal{L}}(L \bowtie R)$$

$\mathcal{L}$  repräsentiert die Menge der Join-Attribute

Rechter Semi-Join

$$L \bowtie R = R \bowtie L = \pi_{\mathcal{R}}(L \bowtie R)$$

# Semi-Joins

Linker Semi-Join ( $L \ltimes R$ ):

L				R				Result		
A	B	C		C	D	E		A	B	C
$a_1$	$b_1$	$c_1$	$\ltimes$	$c_1$	$d_1$	$e_1$	$=$	$a_1$	$b_1$	$c_1$
$a_2$	$b_2$	$c_2$		$c_3$	$d_2$	$e_2$				

Rechter Semi-Join ( $L \ltimes R$ )

L				R				Result		
A	B	C		C	D	E		C	D	E
$a_1$	$b_1$	$c_1$	$\ltimes$	$c_1$	$d_1$	$e_1$	$=$	$c_1$	$d_1$	$e_1$
$a_2$	$b_2$	$c_2$		$c_3$	$d_2$	$e_2$				



# Relational algebra operations

- Projektion  $\pi$
- Selektion  $\sigma$
- Umbenennung  $\rho$
- Kreuzprodukt  $\times$
- Vereinigung  $\cup$
- Differenz  $-$
- Schnitt  $\cap$
- Join (Verbund)  $\bowtie$
- Linker äußerer Join  $\leftarrow \bowtie$
- Rechter äußerer Join  $\bowtie \rightarrow$
- Äußerer Join  $\leftarrow \bowtie \rightarrow$
- Semi-Join (linker)  $\leftarrow \ltimes$
- Semi-Join (rechter)  $\ltimes \rightarrow$
- Gruppierung  $\gamma$
- Division  $\div$

# Gruppierung und Aggregation

Tupel mit **gleichen Attributwerten** (für eine angegebene Liste von Attributnamen) werden gruppiert.

Auf jede Gruppe wird eine Aggregatfunktion angewendet (berechnet für jede Gruppe **einen** Wert).

## Typische Aggregatfunktionen

- count – Anzahl der Elemente pro Gruppe
- sum – Summe der Werte eines Attributs pro Gruppe
- min, max, avg – Minimum, Maximum, Durchschnittswert eines Attributs pro Gruppe

# Gruppierung und Aggregation

## Typische Aggregatfunktionen

- count – Anzahl der Elemente pro Gruppe
- sum – Summe der Werte eines Attributs pro Gruppe
- min, max, avg – Minimum, Maximum, Durchschnittswert eines Attributs pro Gruppe

## Notation

$$\gamma_{L;F}(R)$$

- $L$  Liste der Attribute für die Gruppierung
- $F$  Aggregatfunktion
- Alternative Symbole  $\mathcal{G}$  oder  $\beta$

# Gruppierung und Aggregation

Die Anzahl der Studierenden pro Semester

$$\gamma_{semester; count(*)}(student)$$

student		
studID	name	semester
24002	Nielsen	18
25403	Hansen	12
26120	Pedersen	10
26830	Andersen	6
27550	Larsen	6

$\gamma_{semester; count(*)}(student)$	
semester	count(*)
18	1
12	1
10	1
6	2

# Gruppierung und Aggregation

Die Anzahl der Studierenden pro Semester sowie die minimale studID pro Gruppe

$$\gamma_{semester; count(*), min(studID)}(student)$$

student		
studID	name	semester
24002	Nielsen	18
25403	Hansen	12
26120	Pedersen	10
26830	Andersen	6
27550	Larsen	6

$\gamma_{semester; count(*), min(studID)}(student)$		
semester	count(*)	min(studID)
18	1	24002
12	1	25403
10	1	26120
6	2	26830

Aggregatfunktionen werden auch auf Duplikaten evaluiert!

count vs. count-distinct, sum vs. sum-distinct, etc.

# Relational algebra operations

- Projektion  $\pi$
- Selektion  $\sigma$
- Umbenennung  $\rho$
- Kreuzprodukt  $\times$
- Vereinigung  $\cup$
- Differenz  $-$
- Schnitt  $\cap$
- Join (Verbund)  $\bowtie$
- Linker äußerer Join  $\boxtimes$
- Rechter äußerer Join  $\boxtimes$
- Äußerer Join  $\boxtimes$
- Semi-Join (linker)  $\ltimes$
- Semi-Join (rechter)  $\rtimes$
- Gruppierung  $\gamma$
- Division  $\div$

# Die relationale Division

## Beispiel

Finde die *studIDs* der Studierenden, die alle Vorlesungen mit 4 ECTS besuchen.

takes(*studID*, *courseID*)

course(*courseID*, *title*, *ects*, *teacher*)

$$takes \div \pi_{courseID}(\sigma_{ects=4}(course))$$

Formale Definition:

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

# Die relationale Division

$$takes \div \pi_{courseID}(\sigma_{ects=4}(course))$$

takes	
studID	courseID
26120	5001
27550	5001
27550	4052
28106	4052
28106	4630
28106	5001
28106	5041
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022

course			
courseID	title	ects	teacher
5001	DBS	4	2137
5041	Robotics	4	2125
5043	Software Engineering	3	2126
5049	Ethics	2	2125
4052	Logic	4	2125
5052	Theory of Science	3	2126
5216	Bioethics	2	2126
5259	Chemistry	2	2133
5022	Believe and Knowledge	2	2134
4630	Physics	4	2137



# Die relationale Division

$$takes \div \pi_{courseID}(\sigma_{ects=4}(course))$$

takes	
studID	courseID
26120	5001
27550	5001
27550	4052
28106	4052
28106	4630
28106	5001
28106	5041
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022

$\pi_{courseID}(\sigma_{ects=4}(course))$
courseID
5001
5041
4052
4630

# Die relationale Division

$$takes \div \pi_{courseID}(\sigma_{ects=4}(course))$$

takes	
studID	courseID
26120	5001
27550	5001
27550	4052
28106	4052
28106	4630
28106	5001
28106	5041
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022

$\pi_{courseID}(\sigma_{ects=4}(course))$
courseID
5001
5041
4052
4630

result
studID
28106

# Die relationale Division

$R$	
$M$	$V$
$m_1$	$v_1$
$m_1$	$v_2$
$m_1$	$v_3$
$m_1$	$v_4$
$m_2$	$v_1$
$m_2$	$v_2$
$m_3$	$v_1$
$m_3$	$v_3$
$m_4$	$v_3$

$S_1$
$V$
$v_1$

$S_2$
$V$
$v_1$
$v_2$

$S_3$
$V$
$v_1$
$v_2$
$v_3$

$R \div S_1$
$M$
$m_1$
$m_2$
$m_3$

$R \div S_2$
$M$
$m_1$
$m_2$

$R \div S_3$
$M$
$m_1$

# Ein weiteres Beispiel

wineRecommendation	<table> <tr> <th>wine</th><th>reviewer</th></tr> <tr> <td>La Rose Grand Cru</td><td>Parker</td></tr> <tr> <td>Pinot Noir</td><td>Parker</td></tr> <tr> <td>Riesling Reserve</td><td>Parker</td></tr> <tr> <td>La Rose Grand Cru</td><td>Clarke</td></tr> <tr> <td>Pinot Noir</td><td>Clarke</td></tr> <tr> <td>Riesling Reserve</td><td>Gault-Millau</td></tr> </table>	wine	reviewer	La Rose Grand Cru	Parker	Pinot Noir	Parker	Riesling Reserve	Parker	La Rose Grand Cru	Clarke	Pinot Noir	Clarke	Riesling Reserve	Gault-Millau
wine	reviewer														
La Rose Grand Cru	Parker														
Pinot Noir	Parker														
Riesling Reserve	Parker														
La Rose Grand Cru	Clarke														
Pinot Noir	Clarke														
Riesling Reserve	Gault-Millau														

wineGuide1	<table><tr><th>reviewer</th></tr><tr><td>Parker Clarke</td></tr></table>	reviewer	Parker Clarke	wineGuide2	<table><tr><th>reviewer</th></tr><tr><td>Parker Gault-Millau</td></tr></table>	reviewer	Parker Gault-Millau
reviewer							
Parker Clarke							
reviewer							
Parker Gault-Millau							

$wineRecommendation \div wineGuide1$

wine
La Rose Grand Cru
Pinot Noir

# Ein weiteres Beispiel

wineRecommendation	<table> <tr> <th>wine</th><th>reviewer</th></tr> <tr> <td>La Rose Grand Cru</td><td>Parker</td></tr> <tr> <td>Pinot Noir</td><td>Parker</td></tr> <tr> <td>Riesling Reserve</td><td>Parker</td></tr> <tr> <td>La Rose Grand Cru</td><td>Clarke</td></tr> <tr> <td>Pinot Noir</td><td>Clarke</td></tr> <tr> <td>Riesling Reserve</td><td>Gault-Millau</td></tr> </table>	wine	reviewer	La Rose Grand Cru	Parker	Pinot Noir	Parker	Riesling Reserve	Parker	La Rose Grand Cru	Clarke	Pinot Noir	Clarke	Riesling Reserve	Gault-Millau
wine	reviewer														
La Rose Grand Cru	Parker														
Pinot Noir	Parker														
Riesling Reserve	Parker														
La Rose Grand Cru	Clarke														
Pinot Noir	Clarke														
Riesling Reserve	Gault-Millau														

wineGuide1	reviewer
	Parker Clarke

wineGuide2	reviewer
	Parker Gault-Millau

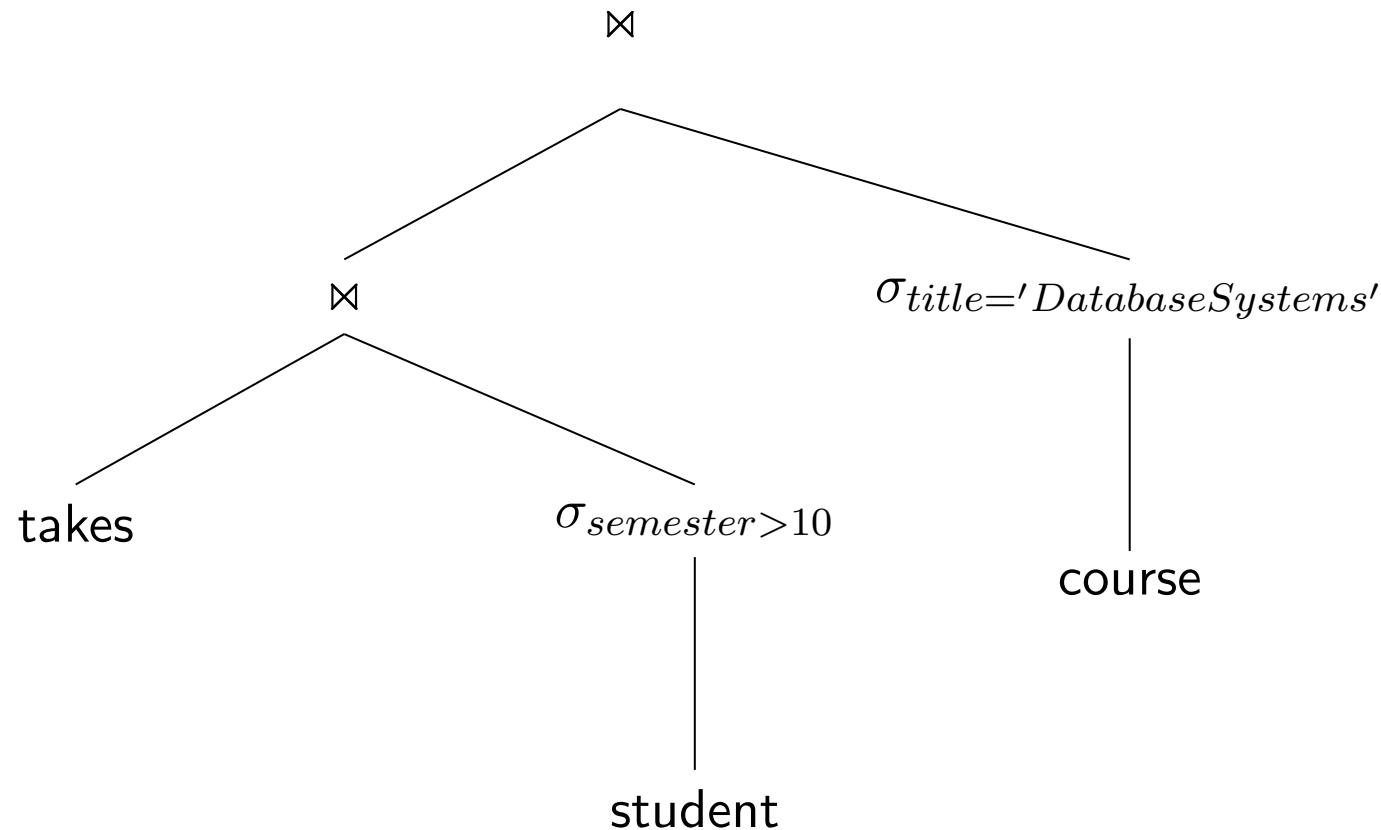
$wineRecommendation \div wineGuide2$

wine
Riesling Reserve

# Operatorbaumdarstellung

Alternative Darstellung von Ausdrücken der relationalen Algebra.

Beispiel:



# Einschränkungen der relationalen Algebra

- Eingeschränkte Arithmetik
  - Finde den Mietpreis bei einer angenommenen Erhöhung von 10%
- Aggregatfunktionen sind nicht in den Basisoperatoren der relationalen Algebra enthalten
  - Wie viele Filme hat jeder Kunde reserviert?
- Transitive Hülle kann nicht gebildet werden
  - Für eine `partOf(Part, ConstituentPart)` Relation, finde alle Teile um ein Auto zu bauen
- Kein Sortieren oder Ausgeben in verschiedenen Formaten
  - Anzeigen einer Zusammenfassung von Rechnungen, sortiert nach Kundenname
- Keine Modifikationen an einer Relation möglich
  - Erhöhe alle \$3.25 Preise auf \$3.50

## Erweiterte Relationale Algebra

Diese Einschränkungen werden in der **erweiterten relationalen Algebra** behoben.

# Zusammenfassung

- Relationenmodell
  - Keine Duplikate (mengenbasiert)
- Relationale Algebra
  - Jeder Operator erfordert Relationen als Input und erzeugt eine Relation
  - Operatoren
    - Sechs fundamentale Operatoren
    - Viele abgeleitete Operatoren
    - Operatoren erzeugen eine Anfragesprache
    - Vereinigungskompatibilität
  - Verschiedene Join-Arten
    - Sie werden die verschiedenen Join-Typen in der Praxis zu schätzen lernen, wenn Sie Anfragen an eine Datenbank stellen.
- Relationale Algebra ist die Basis für Anfragebearbeitung und -optimierung



# Appendix

- 3 Relationale Algebra
  - Relationale Division

# Die relationale Division

Formale Definition:

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

$\pi_{(\mathcal{R}-\mathcal{S})}(R)$
$M$
$m_1$
$m_2$
$m_3$
$m_4$

$S$
$V$
$v_1$
$v_2$

# Die relationale Division

Formale Definition:

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

$\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S$	
$M$	$V$
$m_1$	$v_1$
$m_1$	$v_2$
$m_2$	$v_1$
$m_2$	$v_2$
$m_3$	$v_1$
$m_3$	$v_2$
$m_4$	$v_1$
$m_4$	$v_2$

# Die relationale Division

Formale Definition:

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

$(\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R$	
$M$	$V$
$m_3$	$v_2$
$m_4$	$v_1$
$m_4$	$v_2$

# Die relationale Division

Formale Definition:

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

$(\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R$	
$M$	$V$
$m_3$	$v_2$
$m_4$	$v_1$
$m_4$	$v_2$

$R$	
$M$	$V$
$m_1$	$v_1$
$m_1$	$v_2$
$m_1$	$v_3$
$m_1$	$v_4$
$m_2$	$v_1$
$m_2$	$v_2$
$m_3$	$v_1$
$m_3$	$v_3$
$m_4$	$v_3$

$S$
$V$
$v_1$
$v_2$

# Die relationale Division

Formale Definition:

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

$\pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$
$M$
$m_3$
$m_4$

# Die relationale Division

Formale Definition:

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

$R$	
$M$	$V$
$m_1$	$v_1$
$m_1$	$v_2$
$m_1$	$v_3$
$m_1$	$v_4$
$m_2$	$v_1$
$m_2$	$v_2$
$m_3$	$v_1$
$m_3$	$v_3$
$m_4$	$v_3$

$\pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$
$M$
$m_3$
$m_4$

$S$
$V$
$v_1$
$v_2$



# Die relationale Division

Formale Definition:

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

$\pi_{(\mathcal{R}-\mathcal{S})}(R)$
$M$
$m_1$
$m_2$
$m_3$
$m_4$

$\pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$
$M$
$m_3$
$m_4$

# Die relationale Division

Formale Definition:

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

$\pi_{(\mathcal{R}-\mathcal{S})}(R)$
$M$
$m_1$
$m_2$
$m_3$
$m_4$

$\pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$
$M$
$m_3$
$m_4$

# Die relationale Division

Formale Definition:

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

$\pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$
$M$
$m_1$
$m_2$

# Die relationale Division

Was genau wurde gerade berechnet?

- 1 Alle Vorkommnisse von  $\mathcal{R} - \mathcal{S}$  in  $R$  ohne Duplikate

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

- 2 Alle möglichen Kombinationen davon mit  $S$

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

- 3 Alle **bereits in  $R$  existierenden** Kombinationen entfernen  
Ergebnis: die **nicht existierenden** Kombinationen

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

# Die relationale Division

Was genau wurde gerade berechnet?

- ④ **nicht-existierende** Tupel im Zielschema  
→ Tupel im Zielschema für die  $R$  nicht alle Kombinationen mit  $S$  enthält

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

- ⑤ Alle **existierenden** Tupel in  $R$  auf das Zielschema projizieren

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$

- ⑥ Alle Tupel, die auf der rechten Seite vorkommen, werden verworfen

$$R \div S = \pi_{(\mathcal{R}-\mathcal{S})}(R) - \pi_{(\mathcal{R}-\mathcal{S})}((\pi_{(\mathcal{R}-\mathcal{S})}(R) \times S) - R)$$