

# Relationale Algebra

# Operatoren der Relationalen Algebra

- $\sigma$  Selektion  $\leftarrow$
- $\pi$  Projektion  $\leftarrow$
- $\times$  Kreuzprodukt  $\leftarrow$
- $\bowtie$  Join (Verbund)
- $\rho$  Umbenennung  $\leftarrow$
- $-$  Mengendifferenz
- $\div$  Division
- $\cup$  Vereinigung
- $\cap$  Mengendurchschnitt
- $\ltimes$  Semi-Join (linker)
- $\ltimes$  Semi-Join (rechter)
- $\ltimes$  linker äußerer Join
- $\ltimes$  rechter äußerer Join

Operatoren stammen aus der Mengenlehre

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Vorlesungen			
VorlNr	Titel	SWS	gelesen Von
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

voraussetzen	
Vorgänger	Nachfolger
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

hören	
MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022

Assistenten			
PerslNr	Name	Fachgebiet	Boss
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2126

prüfen			
MatrNr	VorlNr	PersNr	Note
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

# Formale Definition der Algebra

## Basisausdrücke

- Relation der Datenbank oder
- konstante Relationen

• gleiche Attribute  
(Spalten)

gleichen Namen

## Operationen

- Selektion:  $\sigma_p (E_1)$
- Projektion:  $\Pi_S (E_1)$
- Kartesisches Produkt:  $E_1 \times E_2$
- Umbenennung:  $\rho_v (E_1), \rho_{A \leftarrow B} (E_1)$
- Vereinigung:  $E_1 \cup E_2$
- Differenz:  $E_1 - E_2$

$\cap$  - Schnittmenge



Schemagleichheit

Satz der Operatoren  
ist vollständig, d.h.  
alle anderen (hier  
nicht aufgeführten)  
Operatoren lassen  
sich mit ihnen  
ausdrücken

# Beispiel Vereinigung

Ermittle alle Namen der Unimitglieder:

- Unimitglieder befinden sich in den Tabellen Studenten, Professoren und Assistenten
- Die Schemen der Tabellen sind unterschiedlich  
→ Die Schemen müssen im ersten Schritt angeglichen werden (Projektion und Umbenennung (wenn erforderlich))

Wie kann das erfolgen?

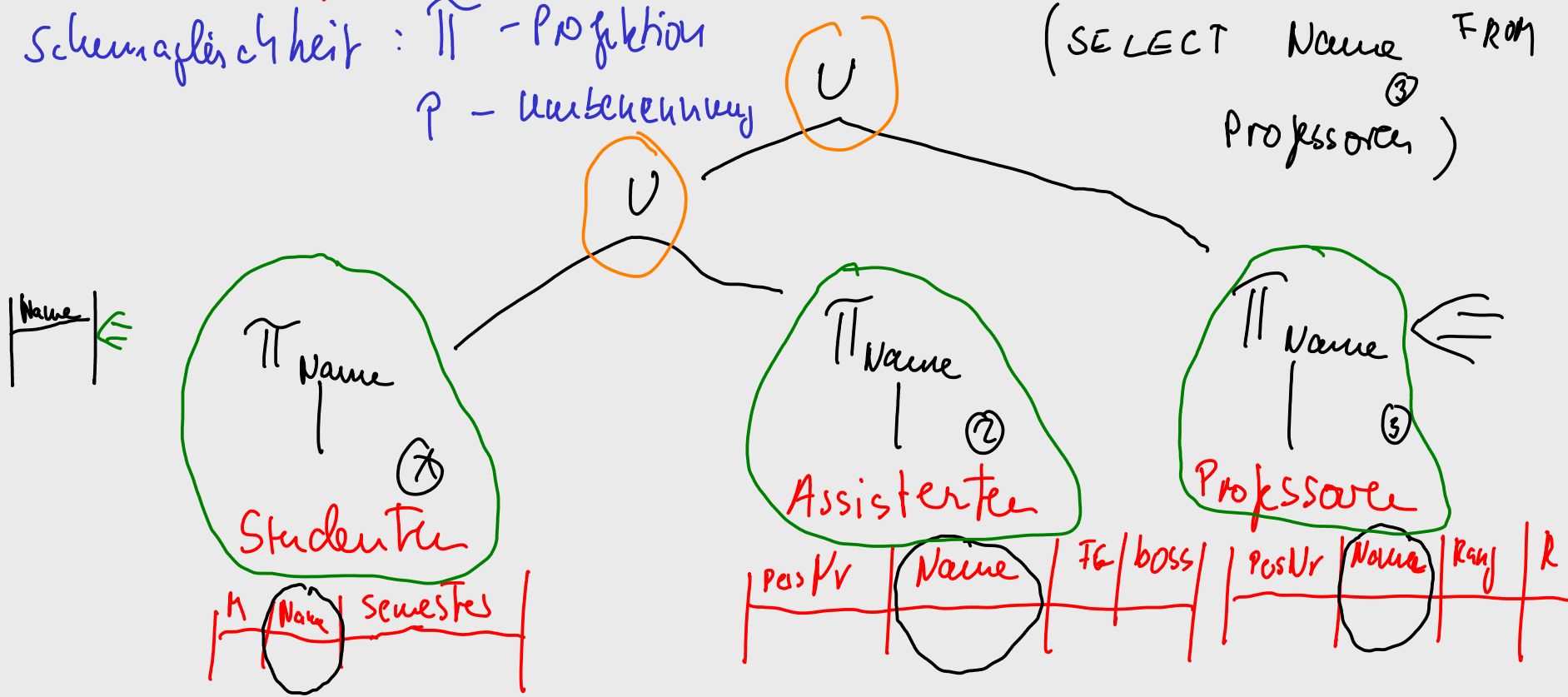
Durchführung einer Projektion auf den Namen, da

$$\text{Sch}(\Pi_{\text{Name}} (\text{Professoren})) = \text{Sch}(\Pi_{\text{Name}} (\text{Studenten})) = \text{Sch}(\Pi_{\text{Name}} (\text{Assistenten}))$$

$S \times A \times P$   
 $\dots | \text{Name} | \dots | \text{Name} | \dots | \text{Name} |$

① (SELECT Name FROM Studenten)  
           UNION  
 ② (SELECT Name FROM Assistenten)  
           UNION  
 (SELECT Name FROM Professoren)

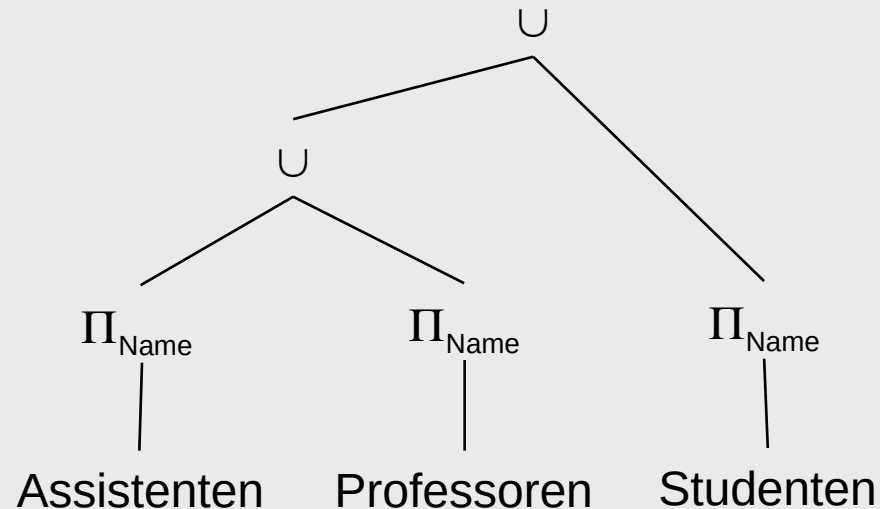
Schemaflichtigkeit:  $\Pi$  - Projektion  
 $\rho$  - Umbenennung



# Beispiel Vereinigung

Berechnen aller Unimitglieder:

$$\Pi_{\text{Name}}(\text{Assistenten}) \cup \Pi_{\text{Name}}(\text{Professoren}) \cup \Pi_{\text{Name}}(\text{Studenten})$$



**(SELECT Name FROM Professoren) UNION (SELECT Name FROM Assistenten) UNION (SELECT Name FROM Studenten)** (UNION eliminiert Duplikate, UNION ALL nicht!)

# Operator Differenz

## Definition Differenzmenge

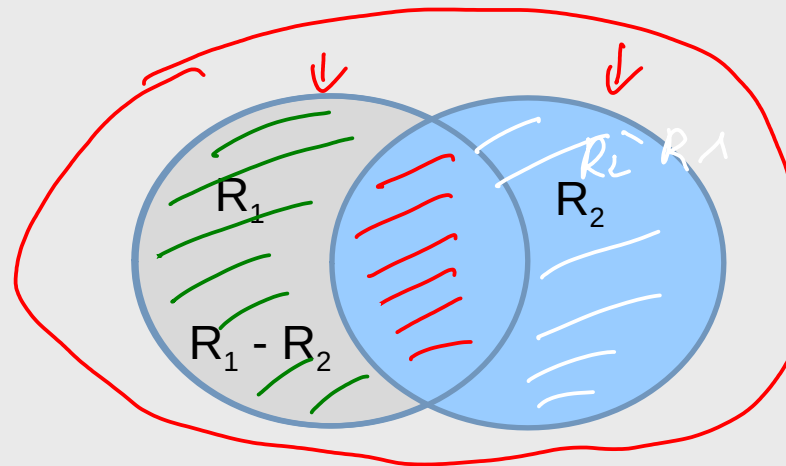
Zwischen zwei Relationen mit den gleichen Attributnamen und Attributtypen (Domänen) kann eine Mengen-Differenz gebildet werden. Dann sind in der Ergebnisrelation  $D = A - B$  alle Tupel aus A enthalten, die nicht in B enthalten sind.

- Symbol: -
- Verbindet zwei Relationen A und B mit gleichem (Attribut-)Schema:  
 $D = A - B$

$$R_1 \cup R_2$$

$$R_1 \cap R_2$$

$$R_1 - R_2$$



Reihenfolge ab  
 $R_1 - R_2$  Operanden  
 $\neq$   
 $R_2 - R_1$



Berechnung aller Studenten (MatrNr), die bisher noch keine Prüfung abgelegt haben.

~~SELECT <sup>Studenten.</sup> MatrNr FROM  
Studenten, prüfer where  
Studenten.MatrNr = prüfer.MatrNr~~

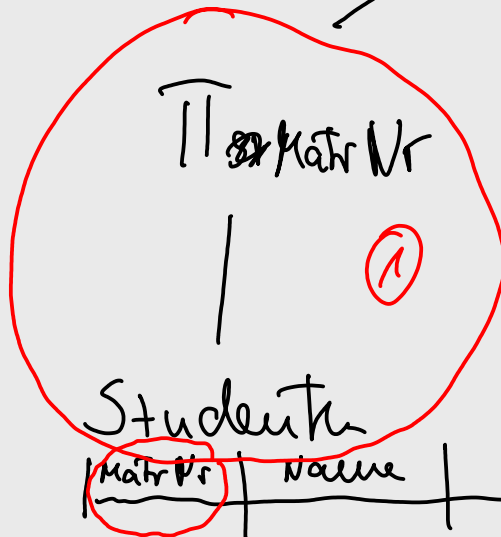
Tupel



Keuzenoperation

(SELECT MatrNr  
FROM Studenten)  
EXCEPT

(SELECT MatrNr  
FROM prüfer)



## Beispiel Differenz

Berechnung aller Studenten, die bisher noch keine Prüfung abgelegt haben.

$$\Pi_{\text{MatrNr}}(\text{Studenten}) - \Pi_{\text{MatrNr}}(\text{pruefen})$$

In SQL gibt es dafür den Befehl EXCEPT

(select MatrNr from Studenten) **EXCEPT**<sup>1</sup> (select MatrNr from pruefen)

<sup>1</sup>Except ist nicht in allen DB-Systemen implementiert (in PostgreSQL ja und in MySQL nein)

Berechnen aller Studenten (MatrNr, Name), die keine Vorlesung hören.

SELECT Studenten.MatrNr,  
NAME  
FROM Studenten,

(SELECT MatrNr FROM  
Studenten)  
EXCEPT

(SELECT MatrNr  
FROM hoeren) AS tmp  
where tmp.MatrNr  
= Studenten.  
MatrNr

Studenten		
M	N	S

$\Pi$  Studenten.MatrNr, Name  
 $\sigma$  Studenten.MatrNr =  
tmp.MatrNr

P tmp AS  
tmp

tmp	
MatrNr	

Studenten		
MatrNr	Name	Semester

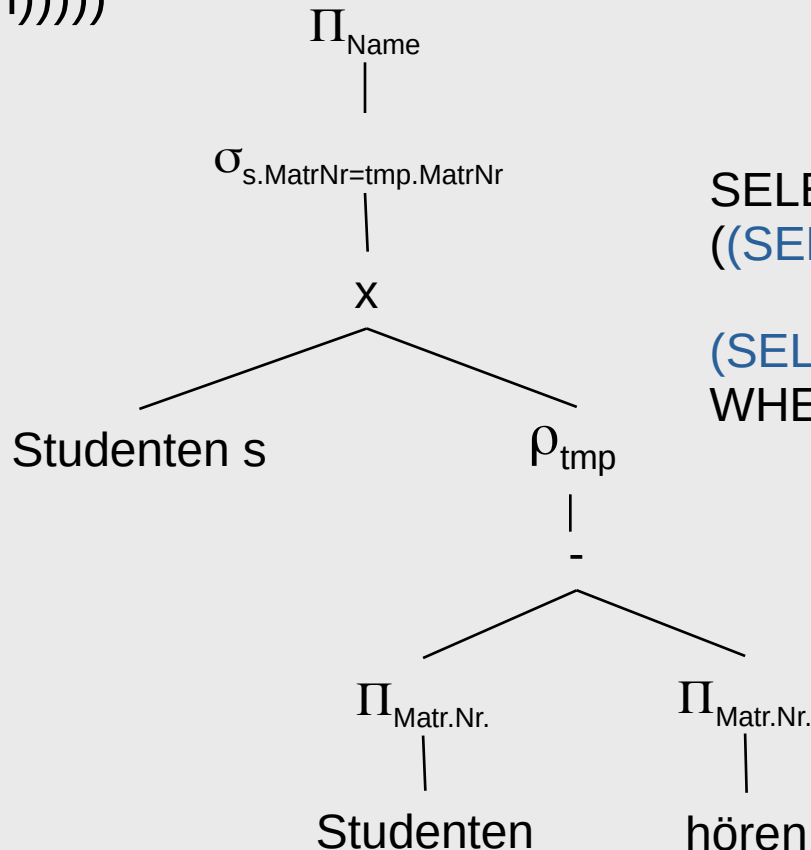
hoeren	
MatrNr	VorNr

MatrNr

# Beispiel Differenz

Berechnen aller Studenten, die keine Vorlesung hören.

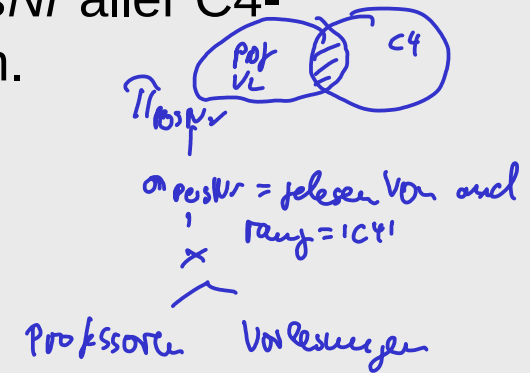
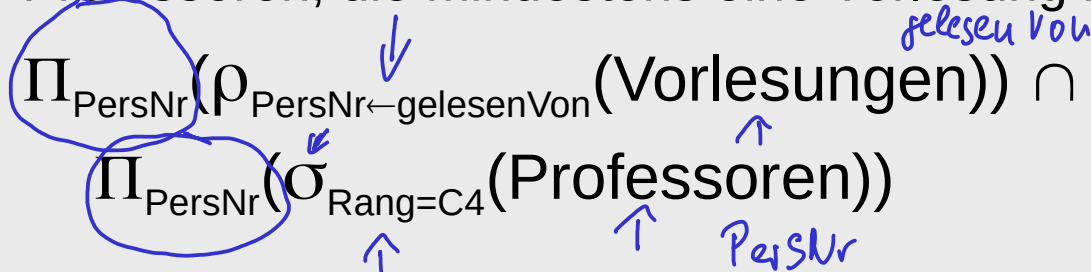
$\Pi_{\text{Name}}(\sigma_{s.\text{MatrNr}=\text{tmp.MatrNr}}(\text{Studenten } s \times \rho_{\text{tmp}}(\Pi_{\text{MatrNr}}(\text{Studenten}) - \Pi_{\text{MatrNr}}(\text{hören}))))$



SELECT Name FROM Studenten,  
((SELECT MatrNr FROM Studenten)  
EXCEPT  
(SELECT MatrNr FROM hoeren)) as tmp  
WHERE s.MatrNr=tmp.MatrNr

# Mengendurchschnitt (1/2)

Als Beispielanwendung für den Mengendurchschnitt (Operatorsymbol  $\cap$ ) betrachten wir folgende Anfrage: Finde die *PersNr* aller C4-Professoren, die mindestens eine Vorlesung halten.



- Mengendurchschnitt nur auf zwei Argumentrelationen mit gleichem Schema anwendbar
- Deshalb ist die Umbenennung des Attribute *gelesenVon* in *PersNr* in der Relation *Vorlesungen* notwendig
- Der Mengendurchschnitt zweier Relationen  $R \cap S$  kann durch die Mengendifferenz wie folgt ausgedrückt werden:

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

SQL: Befehl **INTERSECT** nicht in allen DBMS implementiert (in PostgreSQL ja in MySQL nein)

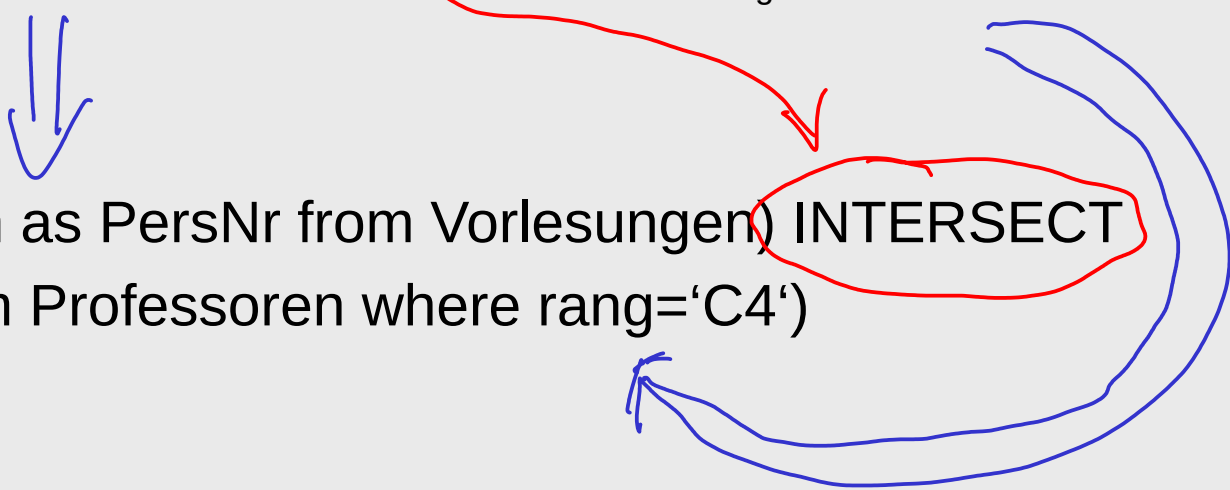
## Mengendurchschnitt (2/2)

Als Beispielanwendung für den Mengendurchschnitt (Operatorsymbol  $\cap$ ) betrachten wir folgende Anfrage: Finde die *PersNr* aller C4-Professoren, die mindestens eine Vorlesung halten.

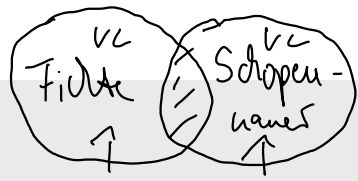
$\Pi_{\text{PersNr}}(\rho_{\text{PersNr} \leftarrow \text{gelesenVon}}(\text{Vorlesungen})) \cap \Pi_{\text{PersNr}}(\sigma_{\text{Rang}=\text{C4}}(\text{Professoren}))$

In SQL:

(select gelesenVon as PersNr from Vorlesungen) INTERSECT  
(select PersNr from Professoren where rang='C4')



Welche Vorlesungen werden von Fichte und Schopenhauer gehört?



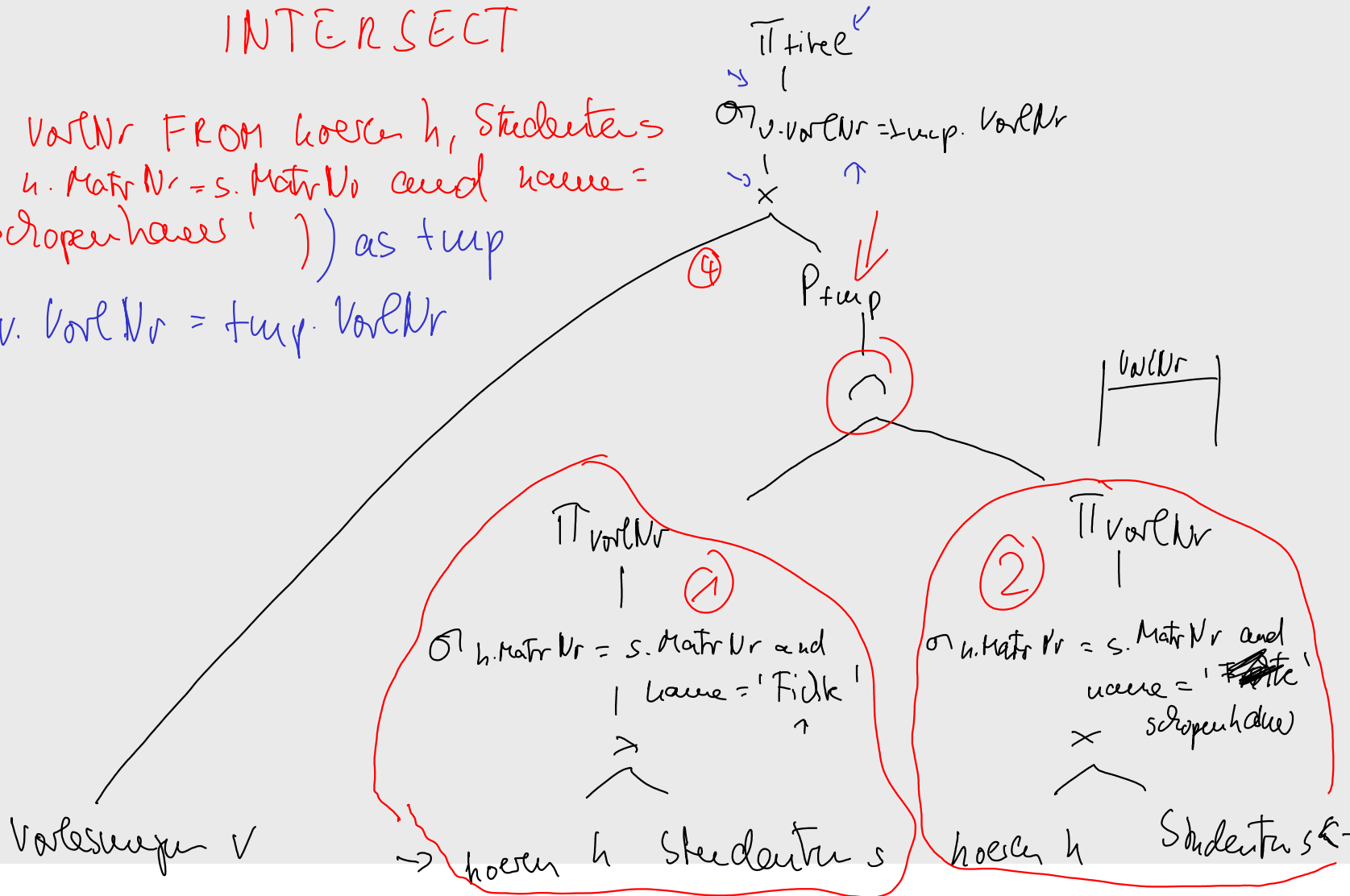
SELECT Titel FROM Vorlesungen V,

(SELECT VorNr FROM hoeren h, Studenten s where  
h.MatrNr = s.MatrNr and name = 'Fichte')

INTERSECT

(SELECT VorNr FROM hoeren h, Studenten s  
where h.MatrNr = s.MatrNr and name =  
'Schopenhauer') as temp

where v.VorNr = temp.VorNr

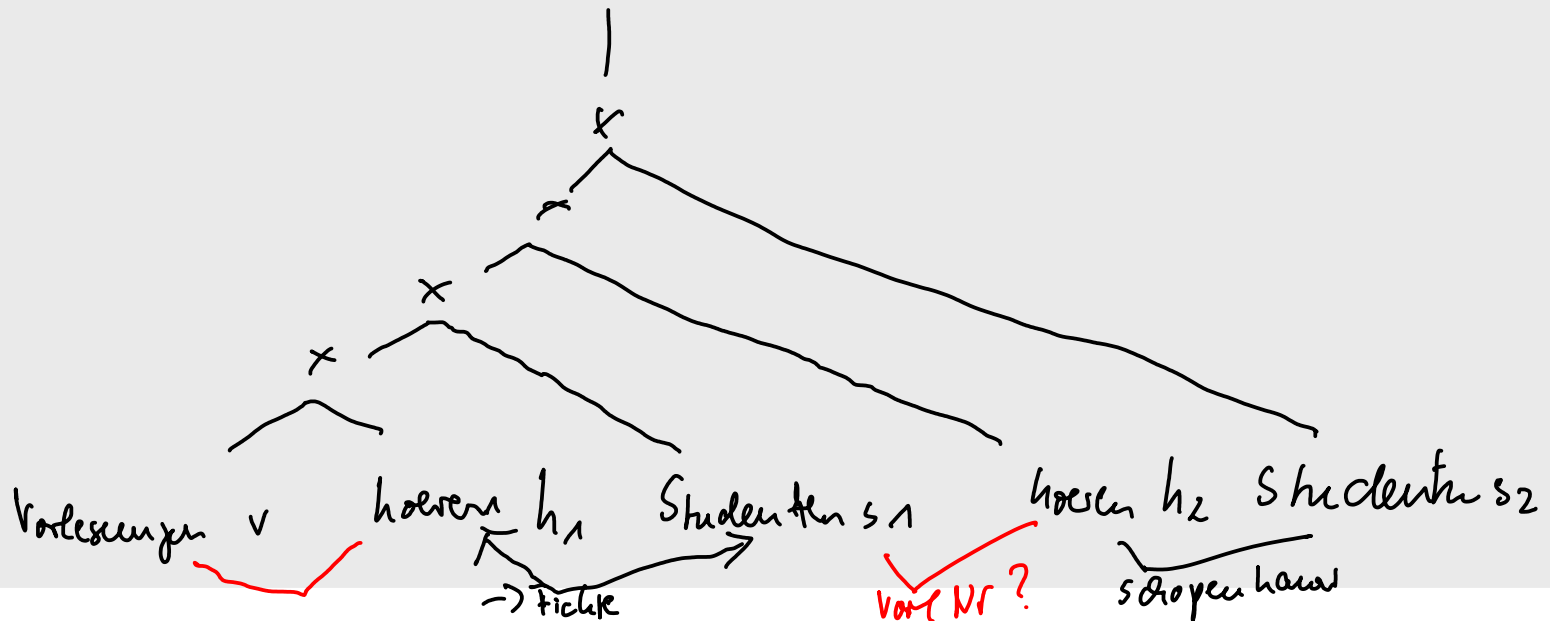


SELECT titel FROM Vorlesungen v, hoeren h1, hoeren h2,  
Studenten s1, Studenten s2

where

$\Pi$  titel  
|

$\sigma_{h1.MatrNr = h2.MatrNr \text{ and } h1.MatrNr = s2.MatrNr \text{ and } s1.name = 'Fichte'}$   
and  $s2.name = 'Schopenhauer'$  and  $h1.VorlNr = h2.VorlNr$  and  
and  $v.VorlNr = h1.VorlNr$

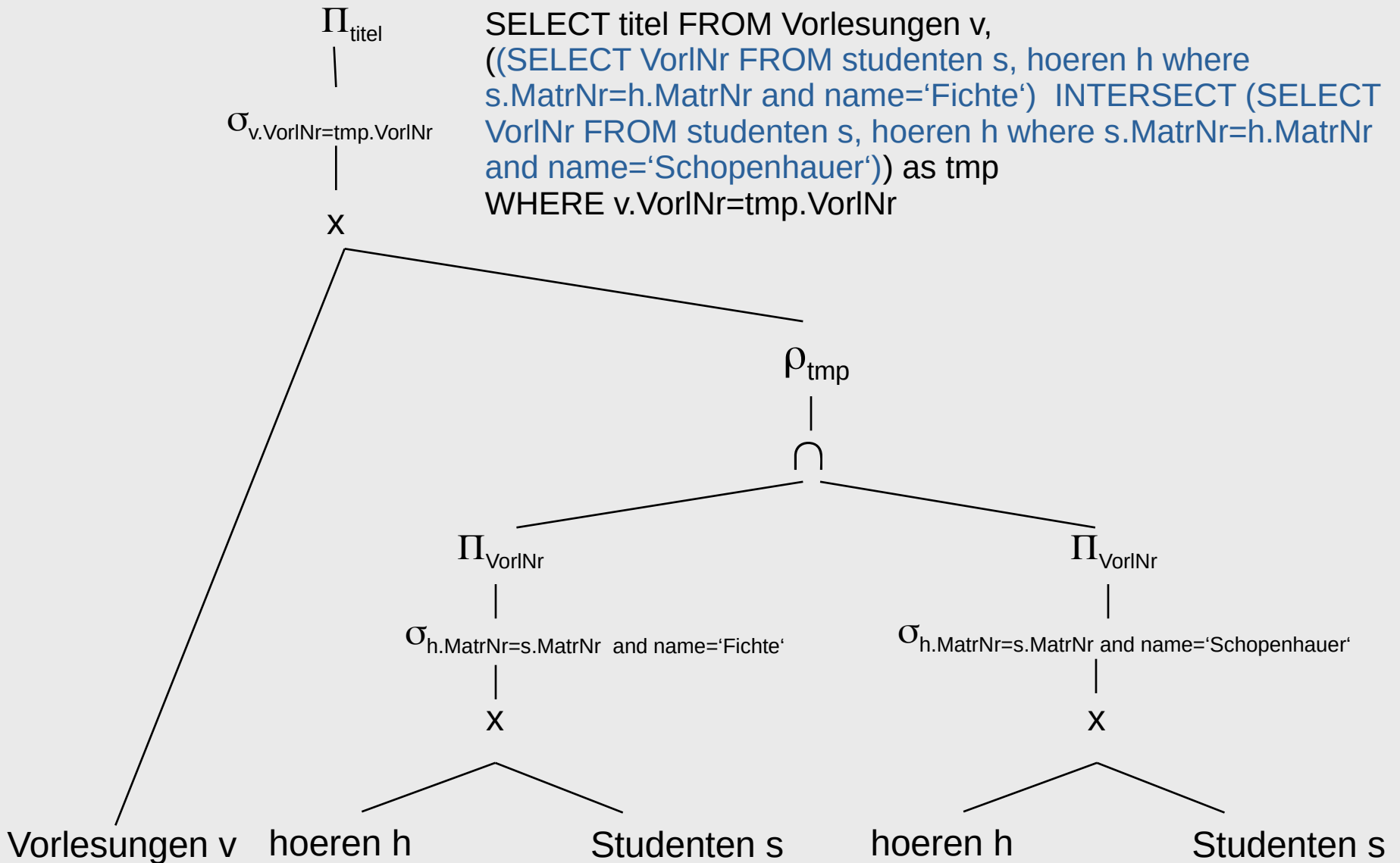




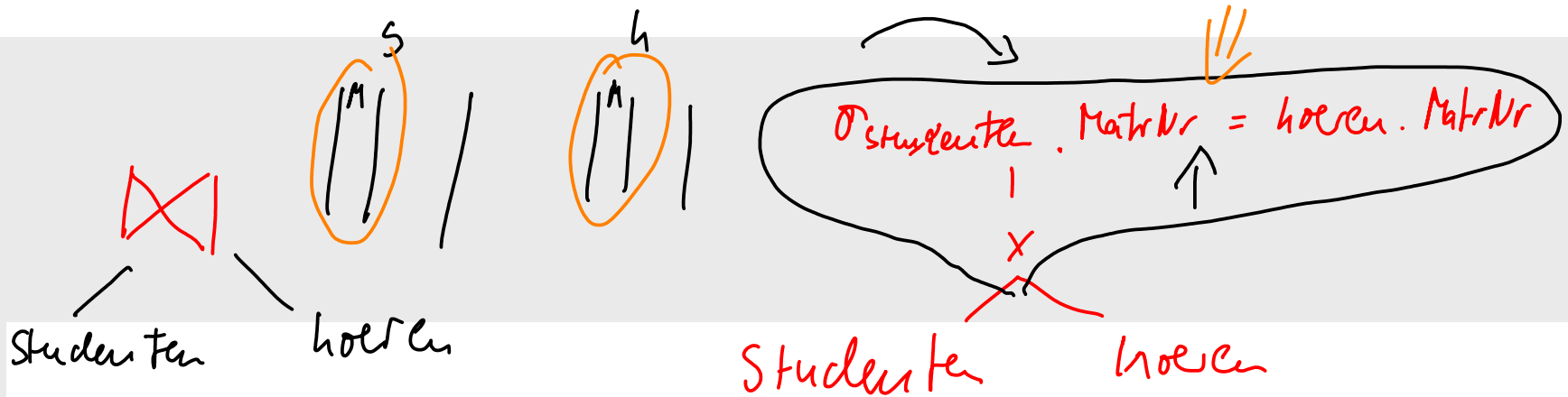
```

SELECT titel FROM Vorlesungen v,
((SELECT VorlNr FROM studenten s, hoeren h where
s.MatrNr=h.MatrNr and name='Fichte') INTERSECT (SELECT
VorlNr FROM studenten s, hoeren h where s.MatrNr=h.MatrNr
and name='Schopenhauer')) as tmp
WHERE v.VorlNr=tmp.VorlNr

```







# Join-Operatoren

SELECT \* FROM Studenten

Natural Join hoeren

# Der natürliche Verbund (Join)

Gegeben seien zwei Relationen:

$R(A_1, \dots, A_m, B_1, \dots, B_k)$

$S(B_1, \dots, B_k, C_1, \dots, C_n)$

↓

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

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

# Beispiel Join

hören	
MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5043

Vorlesungen			
VorlNr	Titel	SWS	gelesenVon
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125

hören ⋈ Vorlesungen

hören		Vorlesungen			
MatrNr	VorlNr	VorlNr	Titel	SWS	gelesenVon
26120	5001	5001	Grundzüge	4	2137
27550	5001	5001	Grundzüge	4	2137
28106	5043	5043	Erkenntnistheorie	3	2126

Einmal wird die Spalte VorlNr in der Relationalen Algebra eliminiert

## 3-Wege-Join

(Studenten ⋈ hören) ⋈ Vorlesungen							
MatrNr	Name	Semester	VorlNr	Titel	SWS	Gelesen Von → PersNr	...
26120	Fichte	10	5001	Grundzüge	4	2137	...
27550	Jonas	12	5022	Glaube und Wissen	2	2134	...
28106	Carnap	3	4052	Wissenschaftstheorie	3	2126	...
...	...	...	...	...	...	...	...

Wenn Attribute mit unterschiedlichen Name verknüpft werden sollen, müssen diese umbenannt werden (z.B. PersNr gelesenVon wenn die lesenden Professoren verknüpft werden sollen)

Studenten ⋈ Vorlesungen ⋈  $\rho_{\text{gelesenVon} \leftarrow \text{PersNr}}$  (Professoren)

# Allgemeiner Join (Theta-Join)

Gegeben seien folgende Relationen(-Schemata)

- $R(A_1, \dots, A_n)$  und
- $S(B_1, \dots, B_m)$

*Studenten*

*Studenten. MatNr. lesen  
lesen. MatrNr*

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

$$R \bowtie_{\theta} S$$

Angabe eines beliebigen Join-Prädikats  $\theta$

*Professoren ✓  
gelesen von = PostNr  
Vorlesungen*

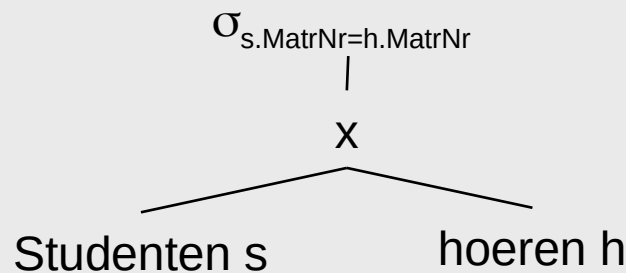
$R \bowtie_{\theta} S$							
R				S			
$A_1$	$A_2$	...	$A_n$	$B_1$	$B_2$	...	$B_m$

*SELECT \* FROM Professoren (INNER) Join Vorlesungen ON  
(gelesen von = PostNr)*

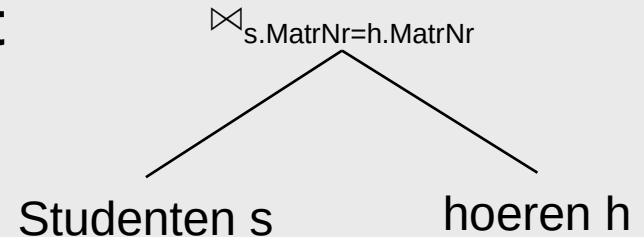
Beispiel: Assistenten, die mehr verdienen als die zugeordneten Profs

Professoren  $\bowtie_{\text{Professoren.Gehalt} < \text{Assistenten.Gehalt} \wedge \text{Boss} = \text{Professoren.PersNr}}$  Assistenten

# Vergleich Kartesisches Produkt/Join



**entspricht**



- Der Join Operator entspricht einem Kartesischen Produkt inklusive einer Selektion
- Join-Operatoren, die auf Gleichheit von Attributen selektieren wird Equi-Join genannt *Natural Join*
- Join-Operatoren mit beliebigem Selektionsprädikat wird Theta-Join genannt



# Join Operatoren in SQL

Beispiel Welche Studenten hören welche Vorlesungen:

```
select MatrNr, Name, Titel from (Studenten s join hoeren h  
on (s.MatrNr=h.MatrNr)) join Vorlesungen v on  
(v.VorlNr=h.MatrNr)
```

- im Prinzip sind die Join-Abfragen performanter, da weniger Zwischenergebnisse berechnet werden
- Da der Anfrageoptimierer der Datenbank vor dem Ausführen der Anfrage optimierenden Umformungen vornimmt, ist die Verwendung des natürlichen Joins nicht erforderlich

# Andere Join-Arten

Natürlicher Join:

L		
A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>

⋈

R		
C	D	E
c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>

=

Resultat				
A	B	C	D	E
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>



~~semi~~ Left Outer Join

Linker äußerer Join:

L		
A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>

⋈

R		
C	D	E
c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>

=

Resultat				
A	B	C	D	E
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>	-	-



↓ null

## Andere Join-Arten

Rechter äußerer Join:

*RIGHT Outer Join*

L		
A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>

⋈

R		
C	D	E
c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>

=

Resultat				
A	B	C	D	E
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
<i>null</i> -	<i>null</i> -	c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>

# Andere Join-Arten

Äußerer Join:

*Outer Join*

L		
A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>

⋈

R		
C	D	E
c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>

=

→

Resultat				
A	B	C	D	E
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>	<i>null</i>	<i>null</i>
<i>null</i>	<i>null</i>	c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>

Semi-Join von L mit R:

*Left Semi Join*

L		
A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>

⋈

R		
C	D	E
c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>

=

Resultat		
A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>

## Andere Join-Arten

Semi-Join von R mit L:

L		
A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>

⋈

R		
C	D	E
c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
c <sub>3</sub>	d <sub>2</sub>	e <sub>2</sub>

=

Resultat		
C	D	E
c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>

