

# Kapitel 6: Das relationale Modell

# Das Relationale Modell

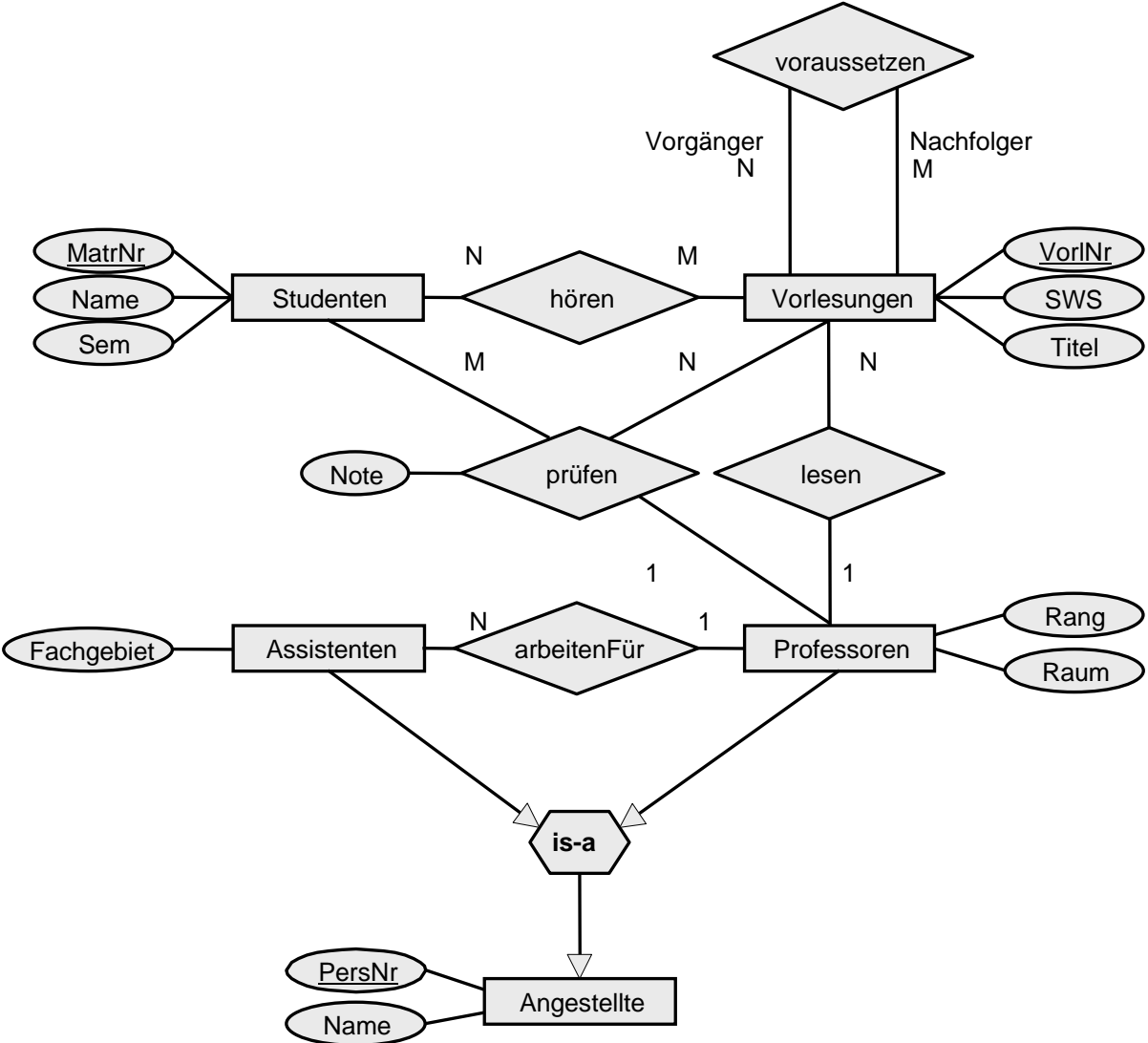
Wertebereiche (Domänen):	$D_1, D_2, \dots, D_n$
Relation:	$R \subseteq D_1 \times D_2 \times \dots \times D_n$
Wertebereich von Attribut A:	<b>dom</b> (A)
Relation	$R \subseteq \mathbf{dom}(A_1) \times \mathbf{dom}(A_2) \times \dots \times \mathbf{dom}(A_n)$
Element von R:	Tupel
Schema der Relation:	<b>sch</b> (R) = $A_1, A_2, \dots, A_n$
aktuelle Ausprägung:	R

bei Datenbanksystemen zusätzlich zum Wertebereich noch Bezeichner:

Telefonbuch : { [Name : string, Adresse: string, TelefonNr : integer] }

Telefonbuch : { [Name, Adresse, TelefonNr] }

# Konzeptuelles Schema der Universität



# Initial-Entwurf für Entity-Typen

Pro Entity-Typ eine Relation (Schlüssel unterstrichen):

Studenten : {[ MatrNr : integer, Name : string, Semester : integer] }

Vorlesungen : {[ VorlNr : integer, Titel : string, SWS : integer] }

Professoren : {[ PersNr : integer, Name:string, Rang : string, Raum : integer] }

Assistenten : {[ PersNr : integer, Name : string, Fachgebiet : string] }

# Initial-Entwurf für Relationship-Typen

Pro Relationship-Typ eine Relation:

hören : {[ MatrNr : integer, VorlNr : integer] }

lesen : {[ PersNr : integer, VorlNr : integer] }

arbeitenFür : {[ AssiPersNr : integer, ProfPersNr : integer] }

voraussetzen : {[ Vorgänger : integer, Nachfolger : integer] }

prüfen : {[ MatrNr:integer, VorlNr:integer, PersNr:integer, Note :decimal] }

Fremdschlüssel := Schlüsselattribut für referierte Entity-Typen

1:N-Beziehung entspricht einer Abbildung:

*lesen : Vorlesungen  $\rightarrow$  Professoren*

*prüfen : Studenten  $\times$  Vorlesungen  $\rightarrow$  Professoren*

# Elimination bei gleichen Schlüsseln

Vorlesungen : {[ VorlNr : integer, Titel : string, SWS : integer] }

Professoren : {[ PersNr : integer, Name : string, Rang : string, Raum:integer] }

lesen : {[ PersNr : integer, VorlNr : integer] }

Relationen mit gleichem Schlüssel können zusammengefaßt werden  
(ggf. Umbenennung erforderlich):

Vorlesungen : {[ VorlNr:integer, Titel:string, SWS:integer, gelesenVon:integer] }

Professoren : {[ PersNr : integer, Name :string, Rang :string, Raum :integer] }

# Elimination bei ungleichen Schlüsseln

Vorlesungen : {[ VorlNr : integer, Titel : string, SWS : integer] }

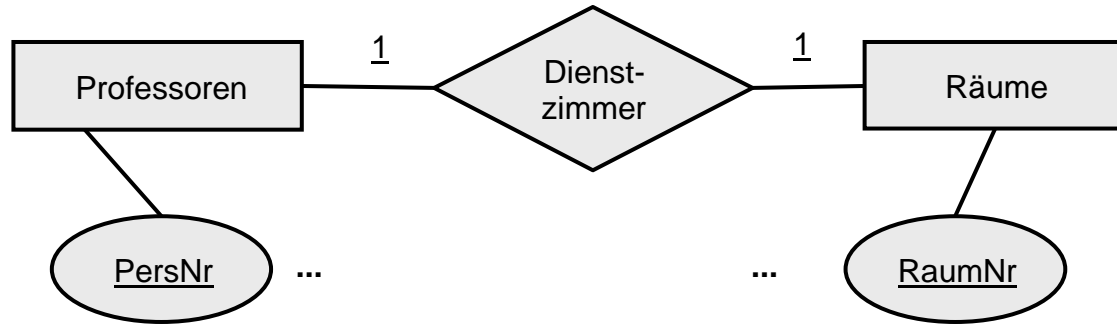
Professoren : {[ PersNr :integer, Name :string, Rang :string, Raum : integer] }

Relationen mit ungleichem Schlüssel sollten nicht zusammengefaßt werden:

Professoren' : {[ PersNr, liestVorl, Name, Rang, Raum ] }

PersNr	liestVorl	Name	Rang	Raum
2125	5041	Sokrates	C4	226
2125	5049	Sokrates	C4	226
2125	4052	Sokrates	C4	226

# Elimination bei 1:1-Beziehung



Professoren : { [ PersNr, Name, Rang ] }

Räume : { [ RaumNr, Größe, Lage ] }

Dienstzimmer: { [ PersNr, RaumNr ] }

Professoren : { [ PersNr, Name, Rang, Raum] }

Räume : { [ RaumNr, Größe, Lage ] }

Professoren : { [ PersNr, Name, Rang] }

Räume : { [ RaumNr, Größe, Lage, ProfPersNr ] }

Obacht:  
Nullwerte !

# Generalisierung

Professoren : {[ PersNr, Name, Rang, Raum] }

Assistenten : {[ PersNr, Name, Fachgebiet] }

Obertyp mit Gemeinsamkeiten:

Angestellte : {[ PersNr, Name] }

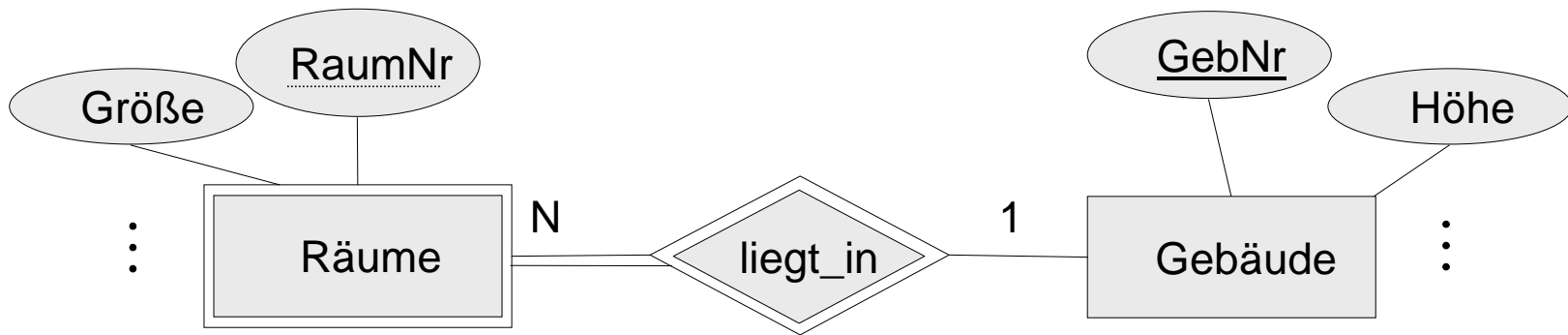
Aber: Die Information zu

[2125, Sokrates, C4, 226]

ist jetzt verteilt auf

[2125, Sokrates]    [2125, C4, 226]

# Schwacher Entity-Typ



Beziehung *liegt\_in* wird verlagert in den Entity-Typ *Räume*:

Räume : { [GebNr, RaumNr, Größe] }

Die Beziehung

*bewohnt* : Professoren  $\rightarrow$  Räume

erfordert drei Attribute

bewohnt : { [PersNr, GebNr, RaumNr] }

Alternative (bei geringer Gebäudeinformation):

Professoren: {[PersNr, Name, Rang, Raum]}

# Relationenschema

Studenten : {[ MatrNr : integer, Name : string, Semester : integer] }

Vorlesungen: {[ VorlNr:integer, Titel:string, SWS:integer, gelesenVon:integer] }

Professoren : {[ PersNr :integer, Name:string, Rang : string, Raum : integer] }

Assistenten : {[ PersNr :integer, Name:string, Fachgebiet :string, Boss:integer] }

hören : {[ MatNr : integer, VorlNr : integer] }

Voraussetzen : {[ Vorgänger : integer, Nachfolger : integer] }

prüfen : {[ MatrNr :integer, VorlNr :integer, PersNr :integer, Note :decimal] }

## 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

## Assistenten

PersNr	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	2134

## Vorlesungen

VorlNr	Titel	SWS	gelesenVon
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

## 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

Ausprägung hören, voraussetzen, prüfen

## hören

MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
27550	5041
28106	4052
28106	5216
28106	5259
27550	4630
29120	5041
29120	5049
29555	5022
25403	5022
29555	5001

## voraussetzen

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

## prüfen

MatrNr	VorlNr	PersNr	Note
28106	5001	2126	1.0
25403	5041	2125	2.0
27550	4630	2137	2.0

# Abfragesprachen

- Relationenalgebra (prozedural):  
konstruktive Verknüpfung; Operatoren:  $\cup, \cap, \dots$
- Relationenkalkül (deklarativ):  
Beschreibung des gewünschten Ergebnisses  
mit Formel der Prädikatenlogik 1. Stufe unter  
Verwendung von  $\exists, \forall, \wedge, \vee, \neg$
- SQL (kommerziell):  
umgangssprachliche Mischung aus  
Relationenalgebra und Relationenkalkül
- Query by Example (für Analphabeten):  
Ausfüllen eines Gerüstes mit Beispiel-Einträgen

# Relationenalgebra

Operanden = Relationen

Operatoren:

- Selektion
- Projektion
- Vereinigung
- Mengendifferenz
- Kartesisches Produkt
- Umbenennung

abgeleitete Operatoren:

- Verbund
- Durchschnitt
- Division

# Selektion

$\sigma_{Semester > 10}(Studenten)$

MatNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12

Selektionsprädikat durch Formel mit

- Attributnamen oder Konstanten als Operanden
- arithmetische Vergleichsoperatoren  $< = > \leq \neq \geq$
- logische Operatoren:  $\wedge \vee \neg$

# Projektion

$\Pi_{Rang} (Professoren)$

Rang
C4
C3

per definitionem keine Duplikate !

# Vereinigung

$$\Pi_{\text{PersNr, Name}}(\textit{Assistenten}) \cup \Pi_{\text{PersNr, Name}}(\textit{Professoren})$$

PersNr	Name
2125	Sokrates
3002	Platon
	.
	.
	.

# Mengendifferenz

$$\Pi_{MatrNr} (\textit{Studenten}) - \Pi_{MatrNr} (\textit{prüfen})$$

MatrNr
24002
26120
26830
.
.
.

# Kartesisches Produkt

*Professoren* × *hören*

PersNr	Name	Rang	Raum	MatrNr	VorlNr
2125	Sokrates	C4	226	26120	5001
...	...	...	...	...	...
2125	Sokrates	C4	226	29555	5001
...	...	...	...	...	...
2137	Kant	C4	7	29555	5001

**$\text{sch}(R \times S) := \text{sch}(R) \cup \text{sch}(S)$ .**

Ggf. durch Voranstellung des Relationennamens  
identifizieren:  $R.A$

# Umbenennung von Relationen und Attributen

$\rho_{Dozenten}(Professoren)$

$\rho_{Zimmer \leftarrow Raum}(Professoren)$

finde Vorgänger vom Vorgänger von Vorlesung 5216:

$\Pi_{V1.Vorgänger}(\sigma_{V1.Nachfolger = V2.Vorgänger \wedge V2.Nachfolger=5216}$   
 $(\rho_{V1}(voraussetzen) \times \rho_{V2}(voraussetzen)))$

V1		V2	
Vorgänger	Nachfolger	Vorgänger	Nachfolger
5001	5041	5001	5041
...	...	...	...
5001	5041	5041	5216
...	...	...	...
5052	5259	5052	5259

# Relationenalgebra

Operanden = Relationen

Operatoren:

- Selektion
- Projektion
- Vereinigung
- Mengendifferenz
- Kartesisches Produkt
- Umbenennung

abgeleitete Operatoren:

- Verbund
- Durchschnitt
- Division

# Natürlicher Verbund (Join)

$R$  habe  $m+k$  Attribute  $A_1, A_2, \dots, A_m, B_1, B_2, \dots, B_k$

$S$  habe  $n+k$  Attribute  $B_1, B_2, \dots, B_k, C_1, C_2, \dots, C_n$

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

*(Studenten  $\triangleright \triangleleft$  hören)  $\triangleright \triangleleft$  Vorlesungen*

*Studenten  $\triangleright \triangleleft$  (hören  $\triangleright \triangleleft$  Vorlesungen)*

*Studenten  $\triangleright \triangleleft$  hören  $\triangleright \triangleleft$  Vorlesungen*

MatrNr	Name	Semester	VorlNr	Titel	SWS	gelesenVon
26120	Fichte	10	5001	Grundzüge	4	2137
25403	Jonas	12	5022	Glaube und Wissen	2	2137
28106	Carnap	3	4052	Wissenschaftstheorie	3	2126
...	...	...	...	...	...	...

# Natürlicher Verbund mit Umbenennung

Vorlesungen der C4-Professoren:

Namen der C4-Professoren mit ihren Vorlesungstiteln:

$\Pi_{Name, Titel} (Professoren \triangleright \triangleleft \rho_{PersNr \leftarrow gelesenVon} (Vorlesungen))$

Name	Titel
Sokrates	Logik
Sokrates	Ethik
Sokrates	Mäeutik
Kant	Die 3 Kritiken
Kant	Grundzüge
...	...

# Theta-Join

Statt Gleichheit bei Attributen jetzt Prädikat  $\theta$ :

$$R \triangleright \triangleleft_{A_1 < B_1 \wedge A_2 = B_2 \wedge A_3 < B_5} S$$

gleichwertig zu

$$R \triangleright \triangleleft_{\theta} S := \sigma_{\theta} (R \times S)$$

Erweitere Professoren und Assistenten um ein Attribut *Gehalt*.

Verbinde Professoren mit höherverdienenden Assistenten:

$$\textit{Professoren} \triangleright \triangleleft_{\textit{Professoren.Gehalt} < \textit{Assistenten.Gehalt} \wedge \textit{Boss} = \textit{Professoren.PersNr}} \textit{Assistenten}$$

# Outer Join

Bisher:

Inner Join (Tupel ohne Partner gehen verloren)

Jetzt:

Outer Join (rette partnerlose Tupel):

- left outer join:  
Tupel der linken Argumentrelation bleiben erhalten
- right outer join:  
Tupel der rechten Argumentrelation bleiben erhalten
- full outer join:  
Tupel beider Argumentrelationen bleiben erhalten

# Outer Joins

L

A	B	C
$a_1$	$b_1$	$c_1$
$a_2$	$b_2$	$c_2$

R

C	D	E
$c_1$	$d_1$	$e_1$
$c_3$	$d_2$	$e_2$

inner Join

A	B	C	D	E
$a_1$	$b_1$	$c_1$	$d_1$	$e_1$

left outer Join

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

right outer Join

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

outer Join

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

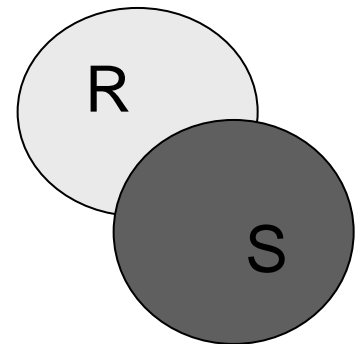
# Mengendurchschnitt

Personalnummer der C4-Professoren,  
die mindestens eine Vorlesung halten:

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

Äquivalenz:

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

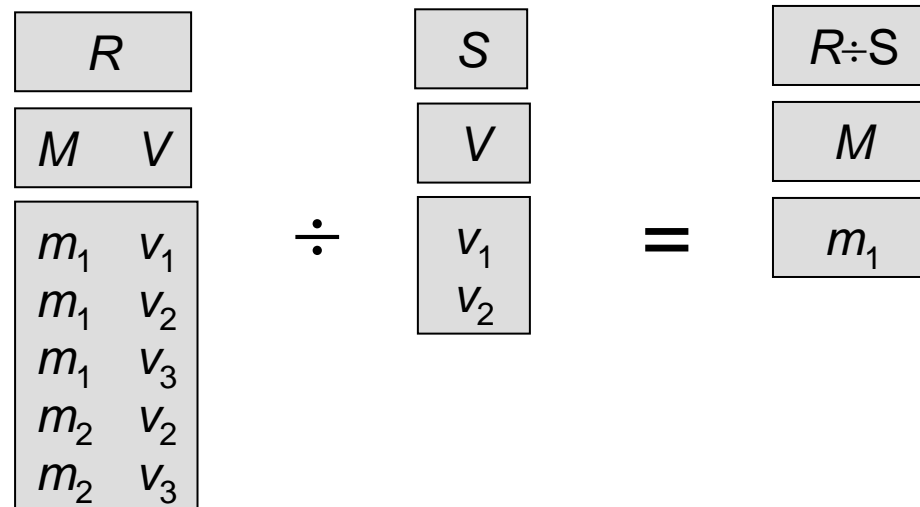


# Division

$R$  sei  $r$ -stellig,  $S$  sei  $s$ -stellig,  $\mathbf{sch}(S) \subseteq \mathbf{sch}(R)$

$R \div S := \{ t = t_1, t_2, \dots, t_{r-s} \mid \forall u \in S : tu \in R \}$

Anfangsstücke von  $R$ , zu denen sämtliche Verlängerungen mit Tupeln aus  $S$  in  $R$  liegen



Namen der Studenten, die alle 4-stündigen Vorlesungen hören:

$\Pi_{Name}(\text{Studenten} \triangleright \triangleleft (\text{Hören} \div \Pi_{VorlNr}(\sigma_{SWS=4}(\text{Vorlesungen}))))$

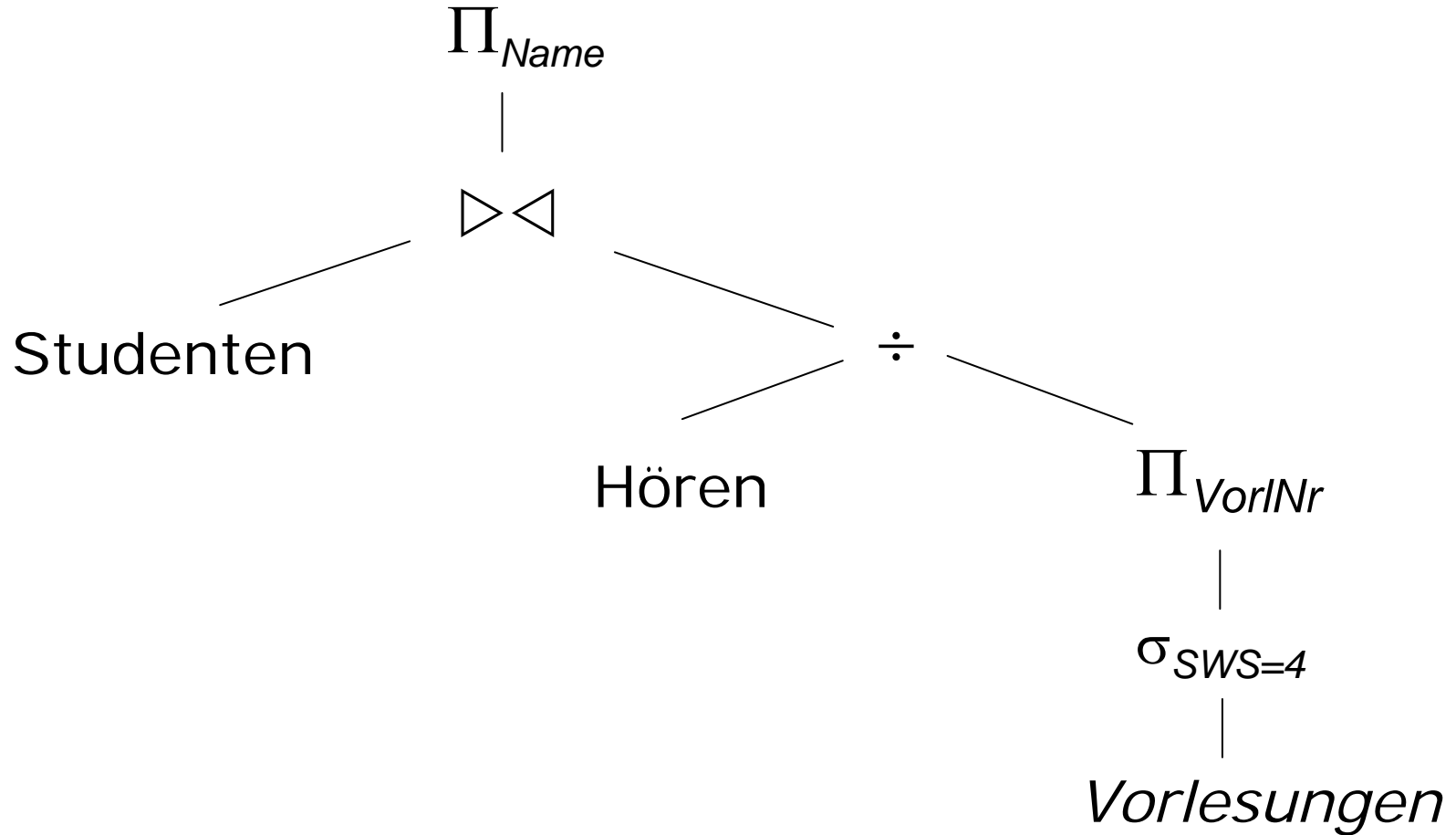
# Ableitung der Division

(Projektion über Index statt Namen)

$T := \pi_{1, \dots, r-s} (R)$  alle Anfangsstücke  
 $K := T \times S$  kombiniert mit allen Verlängerungen aus  $S$   
 $N := K \setminus R$  davon nur solche, die nicht in  $R$  sind  
 $V := \pi_{1, \dots, r-s} (N)$  davon die Anfangsstücke  
 $T \setminus V$  davon das Komplement

$$\pi_{1, \dots, r-s} (R) \setminus \pi_{1, \dots, r-s} ((\pi_{1, \dots, r-s} (R) \times S) \setminus R)$$

# Operatorbaum-Darstellung



# Relationenkalkül

Bisher:           Relationenalgebra (konstruktiv)

Jetzt:             Relationenkalkül (deklarativ)

- Der relationale Tupelkalkül  
(binde freie Variable an Tupel)
- Der relationale Domänenkalkül  
(binde freie Variable an Domäne)

# Der relationale Tupelkalkül

Sei  $t$  eine Tupelvariable (repräsentiert ein Tupel einer Relation)

sei  $P$  ein Prädikat unter Verwendung von  $\vee \wedge \neg \exists \forall \Rightarrow$

Ein Ausdruck im relationalen Tupelkalkül hat die Form

$$\{ t \mid P(t) \}$$

$t$  ist eine freie Variable, die unter Berücksichtigung des Prädikats sukzessive an die Tupel einer Relation gebunden wird

# Der relationale Tupelkalkül

Alle C4-Professoren:

$$\{ p \mid p \in \textit{Professoren} \wedge p.\textit{Rang} = \text{'C4'} \}$$

Alle Professoren mit den Personalnummern ihrer Assistenten:

$$\{ [ p.\textit{Name}, a.\textit{PersNr} ] \mid p \in \textit{Professoren} \wedge \\ a \in \textit{Assistenten} \wedge \\ p.\textit{PersNr} = a.\textit{Boss} \}$$

Alle Studenten, die sämtliche 4-stündigen Vorlesungen hören:

$$\{ s \mid s \in \textit{Studenten} \wedge \forall v \in \textit{Vorlesungen} ( v.\textit{SWS}=4 \Rightarrow \\ \exists h \in \textit{hören} ( h.\textit{VorlNr} = v.\textit{VorlNr} \wedge h.\textit{MatrNr} = s.\textit{MatrNr} ) ) \}$$

# Tupelkalkül versus Relationenalgebra

Sicherer Ausdruck: Ergebnis ist wieder Teilmenge der Domäne.

Z.B. nicht sicher:  $\{ n \mid \neg (n \in \textit{Professoren}) \}$

Bei Beschränkung auf sichere Ausdrücke sind Tupelkalkül und Relationenalgebra gleichmächtig.

# Der relationale Domänenkalkül

Seien  $v_1, v_2, \dots, v_n$  Domänenvariable (repräsent. Attributwerte)

Sei  $P$  ein Prädikat unter Verwendung von  $\vee \wedge \neg \exists \forall \Rightarrow$

Ein Ausdruck im relationalen Domänenkalkül hat die Form

$$\{ [v_1, v_2, \dots, v_n] \mid P(v_1, v_2, \dots, v_n) \}$$

$v_1, v_2, \dots, v_n$  sind freie Domänenvariable, die sukzessive unter Berücksichtigung des Prädikats an Wertebereiche der Attribute gebunden werden.

# Der relationale Domänenkalkül (Beispiel)

Alle Professorennamen mit den Personalnummern ihrer Assistenten:

$$\{ [n,a] \mid \exists p, r, t ( [p, n, r, t] \in \textit{Professoren} \wedge \exists v, w ( [a, v, w, p] \in \textit{Assistenten} ) ) \}$$

Bei Beschränkung auf sichere Ausdrücke sind die Relationenalgebra und der relationale Domänenkalkül gleichmächtig.

# QBE

Fordere Tabellenskelett an und fülle es exemplarisch:

<b>Vorlesungen</b>	VorlNr	Titel	SWS	gelesenVon
		<b>p._t</b>	<b>&gt;3</b>	
		Grundzüge Ethik Logik Die 3 Kritiken		

Im Domänenkalkül:

$\{ [t] \mid \exists v, s, r ( [v, t, s, r] \in \text{Vorlesungen} \wedge s > 3 ) \}$

# QBE Join

Liste alle Professoren, die Logik lesen:

Vorlesungen	VorlNr	Titel	SWS	gelesenVon
		Logik		_otto

Professoren	PersNr	Name	Rang	Raum
	_otto	<b>p._n</b>		
		Sokrates		

# QBE Condition Box

Liste alle Studenten, die in einem höheren Semester sind als Feuerbach:

Studenten	MatrNr	Name	Semester
		<b>p._s</b>	<b>_a</b>
		Feuerbach	<b>_b</b>

conditions

**\_a > \_b**

# QBE

## Gruppierung

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

Gruppierung: **g.**

Aggregatfunktionen: **sum. avg. min. max. all.**

Liste für jede Gehaltsgruppe den Namen des Professors mit der größten Personalnummer:

Professoren	PersNr	Name	Rang	Raum
	<b>p.min._x</b>	<b>p._x</b>	<b>p.g.</b>	
	2137	Kant	C4	
	2133	Popper	C3	

# QBE Gruppierung

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

Liste für jeden Professor die Summe seiner Vorlesungsstunden:

Vorlesungen	VorlNr	Titel	SWS	gelesenVon
			<b>p.sum.all._x</b>	<b>p.g.</b>
			10	2125
			8	2126
			2	2133
			2	2134
			8	2137

# QBE Einfügen

Füge neuen Studenten ein:

Studenten	MatrNr	Name	Semester
<b>i.</b>	<b>4711</b>	<b>wacker</b>	<b>5</b>

# QBE Ändern

Setze Semesterzahl von Feuerbach auf 3:

Studenten	MatrNr	Name	Semester
		<b>Feuerbach</b>	<b>u. 3</b>

# QBE Löschen

Entferne Sokrates und seine Vorlesungen:

Professoren	PersNr	Name	Rang	Raum
<b>d.</b>	<b>_x</b>	Sokrates		

Vorlesungen	VorlNr	Titel	SWS	gelesenVon
<b>d.</b>	<b>_y</b>			<b>_x</b>

hören	VorlNr	MatrNr
<b>d.</b>	<b>_y</b>	

# SQL

Die Namen der Studenten,  
die 4-stündige Vorlesungen hören:

```
select s.name
from   studenten s, hoeren h, vorlesungen v
where  s.matrnr = h.matrnr
and    h.vorlnr = v.vorlnr
and    v.sws     = 4
```

# Relationaler Tupelkalkül

Die Namen der Studenten,  
die jeweils alle 4-stündige Vorlesungen hören:

$$\{ s.name \mid s \in \textit{Studenten} \wedge \forall v \in \textit{Vorlesungen} \\ ( v.SWS=4 \Rightarrow \exists h \in \textit{hören} \\ (h.VorlNr = v.VorlNr \wedge h.MatrNr = s.MatrNr)) \}$$

# Äquivalenzen

$$\forall t \in R(P(t))$$

$$\neg (\exists t \in R(\neg P(t)))$$

$$A \Rightarrow B$$

$$\neg A \vee B$$

$$\neg (A \vee B)$$

$$\neg A \wedge \neg B$$

# Relationaler Tupelkalkül

Die Namen der Studenten,  
die jeweils alle 4-stündige Vorlesungen hören:

$\{ s.name \mid s \in \textit{Studenten} \wedge \forall v \in \textit{Vorlesungen}$

$\{ s.name \mid s \in \textit{Studenten} \wedge \neg (\exists v \in \textit{Vorlesungen}$

$(v.SWS=4 \Rightarrow \exists h \in \textit{hoeren}$

$\neg (v.SWS=4 \Rightarrow \exists h \in \textit{hoeren}$

$\neg (\neg v.SWS=4 \vee \exists h \in \textit{hoeren}$

$( v.SWS=4 \wedge \neg \exists h \in \textit{hoeren}$

$(h.VorlNr = v.VorlNr \wedge h.MatrNr = s.MatrNr)) \}$

$(h.VorlNr = v.VorlNr \wedge h.MatrNr = s.MatrNr)) \}$

# SQL

Die Namen der Studenten,  
die jeweils alle 4-stündige Vorlesungen hören:

$$\{ s.name \mid s \in \textit{Studenten} \wedge \neg (\exists v \in \textit{Vorlesungen} \\ (v.SWS=4 \Rightarrow \wedge \neg \exists h \in \textit{hoeren} \\ (h.VorlNr = v.VorlNr \wedge h.MatrNr = s.MatrNr))) \}$$

```
select s.name from Studenten s
where not exists
  (select * from vorlesungen v
   where sws=4 and not exists
     (select * from hoeren h
      where h.vorlnr = v.vorlnr
           and h.matrnur = s.matrnur))
```