## **Kapitel DB:V**

## V. Grundlagen relationaler Anfragesprachen

- □ Anfragen und Änderungen
- □ Relationale Algebra
- □ Anfragekalküle
- □ Relationaler Tupelkalkül
- □ Relationaler Domänenkalkül

DB:V-1 Relational Algebra & Calculus © STEIN 2004-2018

Ausgangspunkt: Basisrelationen, die in der Datenbank gespeichert sind.

Ziel: abgeleitete Relationen, die aus Basisrelationen berechnet werden.

Ableitung von Relationen mit drei unterschiedlichen Mechanismen:

1. Anfrage

2. Sicht

3. Snapshot

Ausgangspunkt: Basisrelationen, die in der Datenbank gespeichert sind.

Ziel: abgeleitete Relationen, die aus Basisrelationen berechnet werden.

Ableitung von Relationen mit drei unterschiedlichen Mechanismen:

### 1. Anfrage

Folge von Operationen, die aus Basisrelationen eine Ergebnisrelation berechnet. Die Ergebnisrelation kann angezeigt und interaktiv oder durch ein Programm weiterverarbeitet werden.

#### 2. Sicht

Folge von Operationen, die unter einem Sichtnamen *langfristig* gespeichert und unter diesem Namen wieder aufgerufen werden kann (Sichtrelation).

### 3. Snapshot

Ergebnisrelation einer Anfrage, die unter einem Snapshot-Namen abgelegt wird, aber nie ein zweites Mal (mit geänderten Basisrelationen) berechnet wird. Beispiel: Erstellung einer Jahresbilanz.

DB:V-3 Relational Algebra & Calculus ©STEIN 2004-2018

- Bei der Ableitung von Relationen bleiben die Basisrelationen unverändert.
- Update- und Änderungsoperationen verändern die Basisrelationen.
- □ Die Einbettung der Anfragesprache in eine Programmiersprache ermöglicht eine integrierte Weiterverarbeitung.

DB:V-4 Relational Algebra & Calculus © STEIN 2004-2018

Eigenschaften relationaler Anfragesprachen [Heuer/Scholl 1991]

- ad-hoc-formulierbar
- deklarativ

mengenbasiert

abgeschlossen

- orthogonal
- □ adäquat

Eigenschaften relationaler Anfragesprachen [Heuer/Scholl 1991]

#### ad-hoc-formulierbar

Man kann Anfragen formulieren, ohne ein Programm dafür zu schreiben.

#### deklarativ

Man formuliert im deklarativen Stil "Was will ich haben?" und nicht prozedural "Wie programmiere ich das, was ich haben will?"

### mengenbasiert

Die Operationen arbeiten auf Datenmengen – nicht navigierend auf einzelnen Elementen.

### abgeschlossen

Das Ergebnis einer Anfrage ist wieder vom Typ eines Operands (= Relation) und direkt als Eingabe für weitere Anfragen verwendbar.

### orthogonal

Die Operationen sind ohne Einschränkung kombinierbar.

### □ adäquat

Die Charakteristika des unterliegenden Datenmodells werden unterstützt.

DB:V-6 Relational Algebra & Calculus © STEIN 2004-2018

Eigenschaften relationaler Anfragesprachen (Fortsetzung)

□ vollständig

optimierbar

effizient

□ sicher

spezialisiert

Eigenschaften relationaler Anfragesprachen (Fortsetzung)

### vollständig

Die Anfragesprache bildet (mindestens) die Relationenalgebra oder den sicheren Relationenkalkül ab.

### optimierbar

Die Anfragesprache umfasst wenige Operationen, für die es leistungsfähige Optimierungsregeln gibt.

#### effizient

Die Anfragen sind effizient ausführbar.

#### sicher

Keine syntaktisch korrekte Anfrage gerät in eine Endlosschleife oder liefert ein unendliches Ergebnis.

### spezialisiert

Die Anfragesprache ist keine vollständige Programmiersprache. Diese Eigenschaft folgt aus Optimierbarkeit, Effizienz, Sicherheit.

DB:V-8 Relational Algebra & Calculus © STEIN 2004-2018

- □ Beispiel für Orthogonalität: an jeder Stelle, an der ein Basisrelationenname stehen kann, darf auch eine Anfrage stehen (die natürlich eine Relation zurück liefert).
- □ Orthogonalität ist in SQL-89 u.a. deshalb nicht erfüllt, weil in der From-Klausel keine Anfrage stehen darf.
- ☐ Beispiel für Optimierbarkeit: Auswertung von Select-Operationen vor Join-Operationen.
- $\square$  Beispiel für Effizienz: im Relationenmodell ist jede Operation in  $O(n^2)$ , mit n= Anzahl der Tupel einer Relation.

DB:V-9 Relational Algebra & Calculus © STEIN 2004-2018

Eine Algebra  $\mathcal{A} = \langle M, \Omega \rangle$  besteht aus

- 1. einem Grundbereich M sowie
- 2. einer Menge von Operationen  $\Omega$  mit  $\circ: M^n \to M$ ,  $\circ \in \Omega$ .

Eine Algebra  $\mathcal{A} = \langle M, \Omega \rangle$  besteht aus

- 1. einem Grundbereich *M* sowie
- 2. einer Menge von Operationen  $\Omega$  mit  $\circ: M^n \to M$ ,  $\circ \in \Omega$ .

### Bezogen auf die Relationenalgebra:

- 1. M ist die Menge aller Relationen über den Relationenschemata, die zu einer festen Menge von Attributen gebildet werden können.
- 2.  $\Omega$  ist eine Menge von Operationen auf Relationen.

### Folgende Menge von Operationen sind Teil der Relationenalgebra:

- Einstellige Operatoren: Selektion, Projektion, Umbenennung
- Mengenoperationen: Vereinigung, Durchschnitt, Differenz
- Kartesisches Produkt
- Verbundoperationen: natürlicher-, allgemeiner-, äußerer-, semi-Verbund
- relationale Division

DB:V-11 Relational Algebra & Calculus © STEIN 2004-2018

Einstellige Operationen: Selektion

Syntax:  $\sigma_{< \text{COND}>}(r)$ 

Semantik: Auswahl derjenigen Tupel in der Relation  $r(\mathcal{R})$ , die das

Selektionsprädikat < COND> erfüllen:

$$\{t \mid (t \in r) \ \land \ (\langle \mathsf{COND} \rangle(t) = \mathsf{TRUE})\}$$

Einstellige Operationen: Selektion

Syntax:  $\sigma_{< COND>}(r)$ 

Semantik: Auswahl derjenigen Tupel in der Relation  $r(\mathcal{R})$ , die das

Selektionsprädikat < COND> erfüllen:

$$\{t \mid (t \in r) \land (\langle COND \rangle(t) = TRUE)\}$$

### <COND> ist aus folgenden Elementen aufgebaut:

- 1. Operanden: Attributnamen aus dem Schema  $\mathcal{R}$ , Konstanten
- 2. arithmetische Vergleichsoperatoren:  $=, <, \leq, >, \geq, \neq$
- 3. logische Operatoren:  $\land, \lor, \neg$

Einstellige Operationen: Selektion

Syntax:  $\sigma_{< \text{COND}>}(r)$ 

Semantik: Auswahl derjenigen Tupel in der Relation  $r(\mathcal{R})$ , die das

Selektionsprädikat < COND> erfüllen:

$$\{t \mid (t \in r) \land (\langle COND \rangle(t) = TRUE)\}$$

### <COND> ist aus folgenden Elementen aufgebaut:

- 1. Operanden: Attributnamen aus dem Schema  $\mathcal{R}$ , Konstanten
- 2. arithmetische Vergleichsoperatoren:  $=, <, \leq, >, \geq, \neq$
- 3. logische Operatoren:  $\land, \lor, \neg$

### Beispiel:

Ausleihe		
InvNr	Name	
4711	Meyer	
1201	Schulz	
0007	Müller	
4712	Meyer	

$$\sigma_{\mathsf{Name} \leq \, '\mathsf{N}'} \ (\mathsf{Ausleihe}) \quad \leadsto \quad$$

InvNr	Name
4711	Meyer
0007	Müller
4712	Meyer

- Unmittelbar aufeinander folgende Selektionen lassen sich in ihrer Reihenfolge vertauschen, ohne dass sich die Ergebnisrelation ändert. Die Hintereinanderausführung von Selektionen besitzt eine konjunktive Semantik: es werden nur diejenigen Tupel berücksichtigt, die in der Schnittmenge aller selektierten Tupelmengen liegen. Die Schnittmengenbildung ist assoziativ.
- $\square$  Zur Bezeichnung einer Relation können r und  $r(\mathcal{R})$  gleichermaßen verwendet werden abhängig davon, ob auf das Relationenschema  $\mathcal{R}$  zu Unterscheidungszwecken Bezug genommen werden muss.

DB:V-15 Relational Algebra & Calculus © STEIN 2004-2018

Einstellige Operationen: Projektion

Syntax:  $\pi_{\alpha}(r)$ 

Semantik: Projektion aller Tupel in r bzgl. der Attribute in  $\alpha$ :  $\{t(\alpha) \mid t \in r\}$ 

Einstellige Operationen: Projektion

Syntax:  $\pi_{\alpha}(r)$ 

Semantik: Projektion aller Tupel in r bzgl. der Attribute in  $\alpha$ :  $\{t(\alpha) \mid t \in r\}$ 

## Beispiel:

Buch			
InvNr	Titel	ISBN	Autor
0007	Dr. No	3-125	James Bond
1201	Objektbanken	3-111	Heuer
4711	Datenbanken	3-765	Vossen
4712	Datenbanken	3-891	Ullman
4717	Pascal	3-999	Wirth

 $\pi_{\mathsf{InvNr},\mathsf{ISBN}}(\mathsf{Buch}) \quad \rightsquigarrow \quad$ 

InvNr	ISBN
0007	3-125
1201	3-111
4711	3-765
4712	3-891
4717	3-999

- Unmittelbar aufeinander folgende Projektionen lassen sich in ihrer Reihenfolge vertauschen, ohne dass sich die Ergebnisrelation ändert. Die Hintereinanderausführung von Projektionen besitzt eine konjunktive Semantik: es werden nur diejenigen Attribute berücksichtigt, die in der Schnittmenge aller projizierten Attributmengen liegen. Die Schnittmengenbildung ist assoziativ.
- □ Kombination von Selektion und Projektion: bilden die Attribute in <COND> eine Teilmenge der Attribute einer *nachfolgenden* Projektion, lassen sich Selektion und Projektion vertauschen. Falls nicht, muss die Selektion zuerst ausgeführt werden, um sicherzustellen, dass <COND> definiert ist.
- Gilt  $\alpha \subseteq \beta \subseteq \gamma$ , so ist die unmittelbare Hintereinanderausführung der Projektionen bzgl. dieser Attributmengen äquivalent zu der alleinigen Anwendung der Projektion bzgl.  $\alpha$ :  $\pi_{\alpha}\pi_{\beta}\pi_{\gamma}(r) \equiv \pi_{\alpha}(r)$
- $\alpha$  ist eine Menge; entsprechend müsste man z.B. bei  $\alpha = \{\text{InvNr}, \text{ISBN}\}\$  die Selektion  $\pi_{\alpha}$  als  $\pi_{\{\text{InvNr}, \text{ISBN}\}}$  notieren. Die Notation von Attributnamen ohne Mengenklammern im Index ist formal unsauber, hat sich in der Datenbankliteratur aber durchgesetzt.

DB:V-18 Relational Algebra & Calculus © STEIN 2004-2018

Einstellige Operationen: Umbenennung

Semantik: Umbenennung von Attribut  $A_1$  zu  $A_2$  in der Ergebnisrelation.

Einstellige Operationen: Umbenennung

Semantik: Umbenennung von Attribut  $A_1$  zu  $A_2$  in der Ergebnisrelation.

Syntax (b):  $\rho_s(r)$ 

Semantik: Umbenennung der Relation r zu s

Einstellige Operationen: Umbenennung

Semantik: Umbenennung von Attribut  $A_1$  zu  $A_2$  in der Ergebnisrelation.

```
Syntax (b): \rho_s(r)
```

Semantik: Umbenennung der Relation r zu s

## Beispiel:

```
zu (a) \rho_{\text{Buchtitel} \leftarrow \text{Titel}}(\text{Buch})
```

zu (b)  $\rho_{Dokument}(Buch)$ 

- Mit der Umbenennung kann man fehlende Voraussetzungen zur Anwendung von Mengenoperationen schaffen.
- □ Die Umbenennung ermöglicht natürliche Verbunde, wo ansonsten kartesische Produkte entstehen würden: verschieden benannte Attribute werden gleich benannt.
- □ Die Umbenennung ermöglicht kartesische Produkte, wo ansonsten natürliche Verbunde entstehen würden: gleiche Attribute werden verschieden benannt.
- □ Beispiel: Auswertung von Abhängigkeiten zwischen Vorlesungen.

```
\mathcal{R} = \text{voraussetzen} = \{\text{Nachfolgervorlesung}, \text{Vorgaengervorlesung}\}.
```

Bestimmung der Vorvorgängervorlesungen von Vorlesung 4711:

```
\pi_{\rm v2.Vorgaenger}( \sigma_{\rm v1.Nachfolger=4711~\wedge~v1.Vorgaenger=v2.Nachfolger}( \rho_{\rm v1}({\rm voraussetzen})\times\rho_{\rm v2}({\rm voraussetzen}) )
```

DB:V-22 Relational Algebra & Calculus © STEIN 2004-2018

Mengenoperationen: Vereinigung, Durchschnitt, Differenz

Syntax: 
$$r_1 \cup r_2, \ r_1 \cap r_2, \ r_1 - r_2$$
  
Semantik:  $= \{t \mid t \in r_1 \lor t \in r_2\}$   
 $= \{t \mid t \in r_1 \land t \in r_2\}$   
 $= \{t \mid t \in r_1 \land t \not\in r_2\}$ 

Mengenoperationen sind nur für Relationen mit gleichem Schema – d. h., Schemata mit gleichen Attributnamen und Domänen – definiert.

Mengenoperationen: Vereinigung, Durchschnitt, Differenz

Syntax:  $r_1 \cup r_2, r_1 \cap r_2, r_1 - r_2$ 

Semantik: 
$$r_1 \cup r_2 = \{t \mid t \in r_1 \lor t \in r_2\}$$
  
=  $\{t \mid t \in r_1 \land t \in r_2\}$   
=  $\{t \mid t \in r_1 \land t \not\in r_2\}$ 

Mengenoperationen sind nur für Relationen mit gleichem Schema – d. h., Schemata mit gleichen Attributnamen und Domänen – definiert.

### Beispiel:

Buch	
Autor	
James Bond	
Heuer	
Vossen	
Ullman	
Wirth	



 $\mathsf{Buch} \cup \rho_{\mathsf{Autor} \leftarrow \mathsf{Author}}(\mathsf{Book}) \quad \rightsquigarrow$ 



Mengenoperationen: Vereinigung, Durchschnitt, Differenz

Syntax:  $r_1 \cup r_2, r_1 \cap r_2, r_1 - r_2$ 

**Semantik:**  $r_1 \cup r_2 = \{t \mid t \in r_1 \lor t \in r_2\}$ 

$$r_1 \cap r_2 = \{t \mid t \in r_1 \land t \in r_2\}$$
$$= \{t \mid t \in r_1 \land t \notin r_2\}$$

Mengenoperationen sind nur für Relationen mit gleichem Schema – d. h., Schemata mit gleichen Attributnamen und Domänen – definiert.

### Beispiel:

Buch	
Autor	
James Bond	
Heuer	
Vossen	
Ullman	
Wirth	



$$\mathsf{Buch} \cap \rho_{\mathsf{Autor} \leftarrow \mathsf{Author}}(\mathsf{Book}) \quad \rightsquigarrow \quad$$

Autor Vossen Wirth

Mengenoperationen: Vereinigung, Durchschnitt, Differenz

Syntax:  $r_1 \cup r_2, r_1 \cap r_2, r_1 - r_2$ 

**Semantik:**  $r_1 \cup r_2 = \{t \mid t \in r_1 \lor t \in r_2\}$ 

 $r_1 \cap r_2 = \{t \mid t \in r_1 \land t \in r_2\}$ 

 $r_1 - r_2 = \{t \mid t \in r_1 \land t \not\in r_2\}$ 

Mengenoperationen sind nur für Relationen mit gleichem Schema – d. h., Schemata mit gleichen Attributnamen und Domänen – definiert.

### Beispiel:

Buch	
Autor	
James Bond	
Heuer	
Vossen	
Ullman	
Wirth	

Book
Author
Witt
Vossen
Silberschatz
Meier
Wirth

 $\mathsf{Buch} - \rho_{\mathsf{Autor} \leftarrow \mathsf{Author}}(\mathsf{Book}) \quad \rightsquigarrow \quad$ 

Autor
James Bond
Heuer
Ullman

 $lue{}$  Die Syntax A-B (anstelle von  $A\setminus B$ ) zur Notation der Mengendifferenz ist in der Relationenalgebra üblich.

DB:V-27 Relational Algebra & Calculus © STEIN 2004-2018

Kartesisches Produkt [natürlicher Verbund]

Syntax:  $r_1(\mathcal{R}_1) \times r_2(\mathcal{R}_2)$ 

Semantik: Sei  $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$ . Bildung aller  $|r_1| \cdot |r_2|$  Tupel über  $\mathcal{R}_1 \cup \mathcal{R}_2$ :

 $\{t \mid t \in r(\mathcal{R}_1 \cup \mathcal{R}_2) \land t(\mathcal{R}_1) \in r_1 \land t(\mathcal{R}_2) \in r_2\}$ 

Kartesisches Produkt [natürlicher Verbund]

Syntax:  $r_1(\mathcal{R}_1) \times r_2(\mathcal{R}_2)$ 

Semantik: Sei  $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$ . Bildung aller  $|r_1| \cdot |r_2|$  Tupel über  $\mathcal{R}_1 \cup \mathcal{R}_2$ :

$$\{t \mid t \in r(\mathcal{R}_1 \cup \mathcal{R}_2) \land t(\mathcal{R}_1) \in r_1 \land t(\mathcal{R}_2) \in r_2\}$$

### Beispiel:

Buch	
Autor	
James Bond	
Heuer	
Vossen	
Ullman	
Wirth	

Book
Author
Witt
Vossen
Silberschatz
Meier
Wirth

 $\mathsf{Buch} \times \mathsf{Book} \quad \rightsquigarrow \quad$ 

Autor	Author
James Bond	Witt
James Bond	Vossen
James Bond	Silberschatz
James Bond	Meier
James Bond	Wirth
Heuer	Witt
Heuer	Vossen

- Bei gleichen Attributnamen in den beteiligten Relationenschemata wird eine eindeutige Benennung dadurch erzwungen, dass ein qualifizierender Attributbezeichner r.A aus dem Namen der Relation r und dem Attributnamen A konstruiert wird.
- □ Das kartesische Produkt ist eine Operation, die quadratischen Platz (und folglich auch mindestens quadratische Rechenzeit) benötigt.

DB:V-30 Relational Algebra & Calculus © STEIN 2004-2018

#### Konstruktion von Ausdrücken

- 1. Jede Basisrelation ist ein relationaler Algebra-Ausdruck.
- 2. Seien  $E_1$  und  $E_2$  relationale Algebra-Ausdrücke (*Expressions*), dann sind auch folgende Ausdrücke relationale Algebra-Ausdrücke:
  - (a)  $E_1 \cup E_2$ ,  $E_1 \cap E_2$ ,  $E_1 E_2$ ,
  - (b)  $E_1 \times E_2$
  - (c)  $\sigma_{<\text{COND}>}(E_1)$ ,
  - (d)  $\pi_{\alpha}(E_1)$

(e)  $\rho_{A_2 \leftarrow A_1}(E_1)$ ,  $\rho_s(E_1)$ ,

#### Konstruktion von Ausdrücken

- 1. Jede Basisrelation ist ein relationaler Algebra-Ausdruck.
- 2. Seien  $E_1$  und  $E_2$  relationale Algebra-Ausdrücke (*Expressions*), dann sind auch folgende Ausdrücke relationale Algebra-Ausdrücke:
  - (a)  $E_1 \cup E_2$ ,  $E_1 \cap E_2$ ,  $E_1 E_2$ , wobei  $E_1$  und  $E_2$  das gleiche Relationenschema besitzen müssen.
  - (b)  $E_1 \times E_2$
  - (c)  $\sigma_{\langle \text{COND} \rangle}(E_1)$ , wobei  $\langle \text{COND} \rangle$  ein Prädikat über den Attributen des Relationenschemas von  $E_1$  ist.
  - (d)  $\pi_{\alpha}(E_1)$  mit einer Attributliste  $\alpha$ , deren Attribute in dem Relationenschema von  $E_1$  vorkommen.
  - (e)  $\rho_{A_2 \leftarrow A_1}(E_1)$ ,  $\rho_s(E_1)$ , wobei  $A_1$  ein Attributname in dem Relationenschema von  $E_1$  ist und  $A_2$  dort nicht als Attributname vorkommt.

Konstruktion von Ausdrücken (Fortsetzung)

### **Definition 1** (relational vollständig)

Eine Menge von Operationen  $\Omega$  ist relational vollständig bezüglich einer anderen Menge von Operationen  $\Omega'$ , wenn mit  $\Omega$  jede relationenalgebraische Operation ausgedrückt ("simuliert") werden kann, die sich mit  $\Omega'$  ausdrücken lässt.

DB:V-33 Relational Algebra & Calculus © STEIN 2004-2018

Konstruktion von Ausdrücken (Fortsetzung)

### **Definition 1** (relational vollständig)

Eine Menge von Operationen  $\Omega$  ist relational vollständig bezüglich einer anderen Menge von Operationen  $\Omega'$ , wenn mit  $\Omega$  jede relationenalgebraische Operation ausgedrückt ("simuliert") werden kann, die sich mit  $\Omega'$  ausdrücken lässt.

#### Satz 2

Die Menge der Operationen  $\Omega = \{\cup, -, \times, \sigma, \pi, \rho\}$  ist relational vollständig bezüglich der Menge  $\Omega' = \{\cup, \cap, -, \times, \bowtie, \div, \sigma, \pi, \rho\}$  von Operationen der Relationenalgebra.

Insbesondere lassen sich ausdrücken:

- $\neg r_1 \cap r_2$  durch  $r_1 (r_1 r_2)$
- $\Box$   $\bowtie$  durch die Kombination von  $\pi$ ,  $\sigma$  und  $\times$

Konstruktion von Ausdrücken (Fortsetzung)

### **Definition 1** (relational vollständig)

Eine Menge von Operationen  $\Omega$  ist relational vollständig bezüglich einer anderen Menge von Operationen  $\Omega'$ , wenn mit  $\Omega$  jede relationenalgebraische Operation ausgedrückt ("simuliert") werden kann, die sich mit  $\Omega'$  ausdrücken lässt.

#### Satz 2

Die Menge der Operationen  $\Omega = \{\cup, -, \times, \sigma, \pi, \rho\}$  ist relational vollständig bezüglich der Menge  $\Omega' = \{\cup, \cap, -, \times, \bowtie, \div, \sigma, \pi, \rho\}$  von Operationen der Relationenalgebra.

Insbesondere lassen sich ausdrücken:

- $\neg r_1 \cap r_2$  durch  $r_1 (r_1 r_2)$
- $\Box$   $\bowtie$  durch die Kombination von  $\pi$ ,  $\sigma$  und  $\times$
- $\neg r_1 \div r_2$  durch  $\pi_{(\mathcal{R}_1 \mathcal{R}_2)}(r_1) \pi_{(\mathcal{R}_1 \mathcal{R}_2)}(\pi_{(\mathcal{R}_1 \mathcal{R}_2)}(r_1) \times r_2) r_1$

#### Satz 3

Die Menge der Operationen  $\Omega$  in Satz 2 ist unabhängig: keine Operation kann weggelassen werden, ohne die Vollständigkeit zu verlieren.

Die Beschränkung des Selektionsprädikates <cond> auf die einfache Attribut- und Konstantenselektion gefährdet nicht die Vollständigkeitseigenschaft von  $\Omega$  in Satz 2. D.h., die booleschen Operatoren sind nicht notwendig.

DB:V-36 Relational Algebra & Calculus © STEIN 2004-2018

Konstruktion von Ausdrücken (Fortsetzung)

(a) Formulierung komplexer Operationen mittels Schachtelung:

 $\pi_{\mathsf{Nachname},\mathsf{Gehalt}}(\sigma_{\mathsf{AbteilungsNr}=5}(\mathbf{Angestellte}))$ 

Konstruktion von Ausdrücken (Fortsetzung)

#### (a) Formulierung komplexer Operationen mittels Schachtelung:

 $\pi_{\mathsf{Nachname},\mathsf{Gehalt}}(\sigma_{\mathsf{AbteilungsNr}=5}(\mathbf{Angestellte}))$ 

		Angestellte		
Vorname	Nachname	Abteilung	AbteilungsNr	Gehalt

DB:V-38 Relational Algebra & Calculus © STEIN 2004-2018

Konstruktion von Ausdrücken (Fortsetzung)

(a) Formulierung komplexer Operationen mittels Schachtelung:

#### Angestellte

		Angestellte		
Vorname	Nachname	Abteilung	AbteilungsNr	Gehalt
Derk	Smith	Research	5	6000
Peter	Sotelo	Research	5	5000
Pam	Brin	Accounting	3	5500

Konstruktion von Ausdrücken (Fortsetzung)

(a) Formulierung komplexer Operationen mittels Schachtelung:

$$\sigma_{\rm AbteilungsNr=5}({\rm Angestellte})$$

		Angestellte		
Vorname	Nachname	Abteilung	AbteilungsNr	Gehalt
Derk	Smith	Research	5	6000
Peter	Sotelo	Research	5	5000
Pam	Brin	Accounting	3	5500

Konstruktion von Ausdrücken (Fortsetzung)

#### (a) Formulierung komplexer Operationen mittels Schachtelung:

 $\pi_{\mathsf{Nachname},\mathsf{Gehalt}}(\sigma_{\mathsf{AbteilungsNr=5}}(\mathsf{Angestellte}))$ 

		Angestellte		
Vorname	Nachname	Abteilung	AbteilungsNr	Gehalt
Derk	Smith	Research	5	6000
Peter	Sotelo	Research	5	5000
Pam	Brin	Accounting	3	5500

DB:V-41 Relational Algebra & Calculus © STEIN 2004-2018

Konstruktion von Ausdrücken (Fortsetzung)

(b) Formulierung komplexer Operationen mittels benannten Ergebnisrelationen:

Abt5\_Angestellte  $\leftarrow \sigma_{\text{AbteilungsNr}=5}(\text{Angestellte})$ 

Abt5_Angestellte					
Vorname Nachname Abteilung AbteilungsNr Gehalt					
Derk	Smith	Research	5	6000	
Peter	Sotelo	Research	5	5000	

DB:V-42 Relational Algebra & Calculus © STEIN 2004-2018

Konstruktion von Ausdrücken (Fortsetzung)

(b) Formulierung komplexer Operationen mittels benannten Ergebnisrelationen:

 $\textbf{Abt5\_AngestelIte} \; \leftarrow \; \sigma_{\textbf{AbteilungsNr=5}}(\textbf{AngestelIte})$ 

Abt5_Angestellte					
Vorname Nachname Abteilung AbteilungsNr Gehalt					
Derk	Smith	Research	5	6000	
Peter	Sotelo	Research	5	5000	

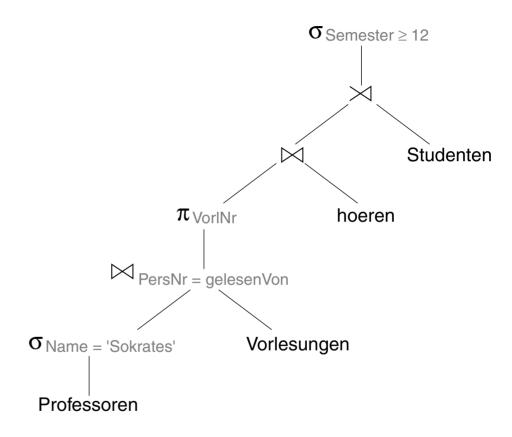
 $\textbf{Ergebnisrelation}(\textbf{Name}, \textbf{Einkommen}) \; \leftarrow \; \pi_{\textbf{Nachname}, \textbf{Gehalt}}(\textbf{Abt5\_Angestellte})$ 

Ergebnisrelation			
Name Einkommen			
Smith	6000		
Sotelo	5000		

DB:V-43 Relational Algebra & Calculus © STEIN 2004-2018

Konstruktion von Ausdrücken (Fortsetzung)

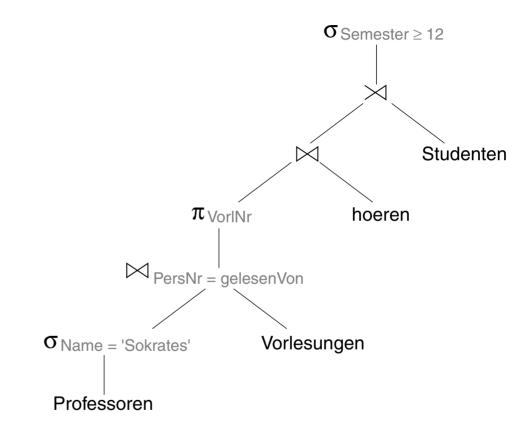
#### Baumdarstellung:



DB:V-44 Relational Algebra & Calculus © STEIN 2004-2018

Konstruktion von Ausdrücken (Fortsetzung)

#### Baumdarstellung:



```
Algebra: \sigma_{\text{Semester} \geq 12}((\pi_{\text{VorINr}}(\sigma_{\text{Name}='\text{Sokrates}'}(\text{Professoren}) \bowtie_{\text{PersNr}=\text{gelesenVon}} \text{Vorlesungen})) \bowtie \text{hoeren}) \bowtie \text{Studenten})
```

DB:V-45 Relational Algebra & Calculus © STEIN 2004-2018

Weitere Operationen: natürlicher Verbund (Natural-Join) [kartesisches Produkt]

Syntax:  $r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2)$ 

Semantik: Alle Tupel, die bei Projektion auf  $R_i$  ein Tupel aus  $r_i$ , i = 1, 2, liefern:

$$\{t \mid t \in r(\mathcal{R}_1 \cup \mathcal{R}_2) \land t(\mathcal{R}_1) \in r_1 \land t(\mathcal{R}_2) \in r_2\}$$

Weitere Operationen: natürlicher Verbund (Natural-Join) [kartesisches Produkt]

Syntax:  $r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2)$ 

Semantik: Alle Tupel, die bei Projektion auf  $\mathcal{R}_i$  ein Tupel aus  $r_i$ , i = 1, 2, liefern:

$$\{t \mid t \in r(\mathcal{R}_1 \cup \mathcal{R}_2) \land t(\mathcal{R}_1) \in r_1 \land t(\mathcal{R}_2) \in r_2\}$$

#### Beispiel:

Ausleihe				
InvNr	Name			
4711	Meyer			
1201	Schulz			
0007	Müller			
4712	Meyer			

Buch					
InvNr	Titel	ISBN	Autor		
0007	Dr. No	3-125	James Bond		
1201	Objektbanken	3-111	Heuer		
4711	Datenbanken	3-765	Vossen		
4712	Datenbanken	3-891	Ullman		
4717	Pascal	3-999	Wirth		

Ausleihe ⋈ Buch

Name	InvNr	Titel	ISBN	Autor
Müller	0007	Dr. No	3-125	James Bond
Schulz	1201	Objektbanken	3-111	Heuer
Meyer	4711	Datenbanken	3-765	Vossen
Meyer	4712	Datenbanken	3-891	Ullman

Weitere Operationen: natürlicher Verbund (Natural-Join) [kartesisches Produkt]

Syntax:  $r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2)$ 

Semantik: Alle Tupel, die bei Projektion auf  $\mathcal{R}_i$  ein Tupel aus  $r_i$ , i = 1, 2, liefern:

$$\{t \mid t \in r(\mathcal{R}_1 \cup \mathcal{R}_2) \land t(\mathcal{R}_1) \in r_1 \land t(\mathcal{R}_2) \in r_2\}$$

$$\underbrace{A_1, \ldots, A_n, B_1, \ldots, B_k}_{B_1, \ldots, B_k, C_1, \ldots, C_m}$$

$$r_1(\mathcal{R}_1)\bowtie r_2(\mathcal{R}_2) = \underbrace{\pi_{A_1,\ldots,A_n,\ r_1.B_1,\ldots,r_1.B_k,\ C_1,\ldots,C_m}}_{\text{Projektion}}\underbrace{\left(\underbrace{\sigma_{r_1.B_1=r_2.B_1,\ldots,r_1.B_k=r_2.B_k}}_{\text{Selektion}}\underbrace{\left(r_1\times r_2\right)}_{\text{kartesisches Produkt}}\right)$$

Weitere Operationen: natürlicher Verbund (Natural-Join) [kartesisches Produkt]

Syntax:  $r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2)$ 

Semantik: Alle Tupel, die bei Projektion auf  $R_i$  ein Tupel aus  $r_i$ , i = 1, 2, liefern:

$$\{t \mid t \in r(\mathcal{R}_1 \cup \mathcal{R}_2) \land t(\mathcal{R}_1) \in r_1 \land t(\mathcal{R}_2) \in r_2\}$$

$$egin{aligned} egin{aligned} egin{aligned} egin{aligned} egin{aligned} egin{aligned} egin{aligned} egin{aligned} egin{aligned} egin{aligned} A_1,\ldots,A_n,B_1,\ldots,B_k \ B_1,\ldots,B_k,C_1,\ldots,C_m \end{aligned}$$

$$r_1(\mathcal{R}_1)\bowtie r_2(\mathcal{R}_2) = \underbrace{\pi_{A_1,\ldots,A_n,\ r_1.B_1,\ldots,r_1.B_k,\ C_1,\ldots,C_m}}_{\text{Projektion}}\underbrace{\left(\underbrace{\sigma_{r_1.B_1=r_2.B_1,\ldots,r_1.B_k=r_2.B_k}}_{\text{Selektion}}\underbrace{\left(r_1\times r_2\right)}_{\text{kartesisches Produkt}}\right)$$

$r_1(\mathcal{R}_1)\bowtie r_2(\mathcal{R}_2)$				
$R_1 - R_2$	$\mathcal{R}_{1} \cap \mathcal{R}_{2}$	$\mathcal{R}_2 - \mathcal{R}_1$		

Weitere Operationen: natürlicher Verbund (Fortsetzung)

Eigenschaften des natürlichen Verbunds:

- $\Box$  Kommutativität:  $r_1 \bowtie r_2 = r_2 \bowtie r_1$
- Assoziativität:  $(r_1 \bowtie r_2) \bowtie r_3 = r_1 \bowtie (r_2 \bowtie r_3)$ Somit ist folgende Notation möglich:  $r_1 \bowtie r_2 \bowtie \ldots \bowtie r_p \equiv \bowtie_{i=1}^p r_i$
- $\square \mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset \Rightarrow r_1 \bowtie r_2 = r_1 \times r_2$

Weitere Operationen: natürlicher Verbund (Fortsetzung)

#### Eigenschaften des natürlichen Verbunds:

- $\Box$  Kommutativität:  $r_1 \bowtie r_2 = r_2 \bowtie r_1$
- □ Assoziativität:  $(r_1 \bowtie r_2) \bowtie r_3 = r_1 \bowtie (r_2 \bowtie r_3)$ Somit ist folgende Notation möglich:  $r_1 \bowtie r_2 \bowtie \ldots \bowtie r_p \equiv \bowtie_{i=1}^p r_i$
- $\square \quad \mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset \quad \Rightarrow \quad r_1 \bowtie r_2 = r_1 \times r_2$

#### Beispiel "3-Wege-Join":

(Ausleihe ⋈ Buch) ⋈ Verlag = Ausleihe ⋈ (Buch ⋈ Verlag)

#### Beispiel für Entartung zum kartesischen Produkt:

 $\pi_{\mathsf{InvNr}}(\mathsf{Ausleihe}) \bowtie \pi_{\mathsf{Autor}}(\mathsf{Buch})$ 

#### Bemerkungen:

- Die gemeinsamen Attribute der bei einem Join beteiligten Relationen werden auch als "Join-Attribute" bezeichnet.
- $\Box$  Die Umbennungsoperation  $\rho$  ermöglicht es, Relationen über zwei Attribute zu verbinden, welche die gleiche Bedeutung (Semantik), aber einen unterschiedlichen Namen haben.
- Tupel, die keinen Join-Partner finden, sogenannte "Dangling Tuples", werden eliminiert. Folglich ist die Projektion im Allgemeinen nicht die inverse Operation zum natürlichen Verbund. Es gilt:  $\pi_{\mathcal{R}_1}(r_1\bowtie r_2)\subseteq r_1$
- Der natürliche Verbund ist im Allgemeinen nicht die inverse Operation zu zwei Projektionen. Sei r eine Relation über  $\mathcal{R}$  und mit  $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ . Dann gilt der folgende Zusammenhang nur bei *Verbundtreue*:  $\pi_{\mathcal{R}_1}(r) \bowtie \pi_{\mathcal{R}_2}(r) = r$

DB:V-52 Relational Algebra & Calculus © STEIN 2004-2018

Weitere Operationen: allgemeiner Verbund (Theta-Join)

Der allgemeine Theta-Join-Operator,  $\bowtie_{\theta}$ , erlaubt die Spezifikation eines beliebigen Join-Prädikates  $\theta$ . Das Ergebnis des Theta-Joins enthält *alle* (bei Namensgleichheit: qualifizierten) Attribute der beteiligten Relationen:

$$r_1 \bowtie_{\theta} r_2 = \sigma_{\theta}(r_1 \times r_2)$$

#### Beispiel:

$$r_1 \bowtie_{A_1 > A_2} \land A_3 = A_4 \land A_5 < A_6 r_2$$

#### Bemerkungen:

- □ Einen Theta-Join der Form  $r_1 \bowtie_{A_{1_1}=A_{2_1},...,A_{1_n}=A_{2_n}} r_2$  nennt man auch Equi-Join. Im Unterschied zum Natural-Join werden beim Equi-Join alle Attribute übernommen.
- Die bislang eingeführten Join-Operatoren werden auch innere Joins genannt. Für sie gilt, dass diejenigen Tupel der Argumentrelationen verloren gehen, die keinen Join-Partner gefunden haben.
- □ Mit den äußeren Join-Operatoren können auch partnerlose Tupel der Argumentrelationen in die Ergebnisrelation übernommen werden: bei Anwendung des Left-Outer-Join bleiben die Tupel der linken Argumentrelation immer erhalten, bei Anwendung des Right-Outer-Join die Tupel der rechten Argumentrelation. Die nicht gegebenen Attributwerte der partnerlosen Tupel werden mit Nullwerten, in Zeichen: ⊥, aufgefüllt.

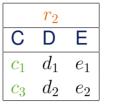
DB:V-54 Relational Algebra & Calculus © STEIN 2004-2018

Weitere Operationen: äußerer Verbund (Outer-Join)

□ Natural-Join:

	$r_1$	
Α	В	С
$a_1$	$b_1$	$c_1$
$a_2$	$b_2$	$c_2$

M

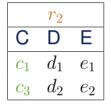


 $r_1 \bowtie r_2$ A B C D E  $a_1 \ b_1 \ c_1 \ d_1 \ e_1$ 

□ Left-Outer-Join:

$$egin{array}{c|cccc} & r_1 & & & \\ A & B & C & & \\ \hline a_1 & b_1 & c_1 & & \\ a_2 & b_2 & c_2 & & \\ \hline \end{array}$$

 $\supset$ 



Weitere Operationen: äußerer Verbund (Outer-Join)

□ Natural-Join:

	$r_1$	
Α	В	С
$a_1$	$b_1$	$c_1$
$a_2$	$b_2$	$c_2$

M

$$egin{array}{c|cccc} & r_2 & & & & \\ \hline C & D & E & & & \\ \hline c_1 & d_1 & e_1 & & & \\ c_3 & d_2 & e_2 & & & & \\ \hline \end{array}$$

□ Left-Outer-Join:

$$egin{array}{c|cccc} & r_1 & & & \\ A & B & C & & \\ a_1 & b_1 & c_1 & \\ a_2 & b_2 & c_2 & & \\ \end{array}$$

$$egin{array}{ccccc} & r_2 & & & & \\ \hline C & D & E & & & \\ c_1 & d_1 & e_1 & & \\ c_3 & d_2 & e_2 & & & \\ \end{array}$$

□ Right-Outer-Join:

$$egin{array}{c|cccc} r_1 & & & \\ \hline A & B & C & \\ \hline a_1 & b_1 & c_1 & \\ a_2 & b_2 & c_2 & \\ \hline \end{array}$$

$$egin{array}{ccccc} & r_2 & & & & \\ C & D & E & & & \\ c_1 & d_1 & e_1 & & \\ c_3 & d_2 & e_2 & & & \\ \end{array}$$

Weitere Operationen: äußerer Verbund (Outer-Join)

Natural-Join:

$$egin{array}{c|cccc} & r_1 & & & \\ A & B & C & & \\ \hline a_1 & b_1 & c_1 & & \\ a_2 & b_2 & c_2 & & \\ \hline \end{array}$$

 $r_1 \bowtie r_2$ 

Left-Outer-Join:

$$egin{array}{c|cccc} & r_1 & & & \\ A & B & C & & \\ \hline a_1 & b_1 & c_1 & & \\ a_2 & b_2 & c_2 & & \\ \hline \end{array}$$

 $\supset$ 

$$egin{array}{ccccc} r_2 & & & & & \\ \hline C & D & E & & & \\ c_1 & d_1 & e_1 & & & \\ c_3 & d_2 & e_2 & & & \\ \end{array}$$

Right-Outer-Join:

$$egin{array}{c|cccc} & r_1 & & & \\ \hline A & B & C & & \\ \hline a_1 & b_1 & c_1 & & \\ a_2 & b_2 & c_2 & & \\ \hline \end{array}$$

 $\bowtie$ 

Full-Outer-Join:

$$egin{array}{c|cccc} & r_1 & & & \\ A & B & C & & \\ \hline a_1 & b_1 & c_1 & & \\ a_2 & b_2 & c_2 & & \\ \hline \end{array}$$

 $\mathbb{X}$ 

$$egin{array}{cccc} & r_2 & & & \\ \hline C & D & E & & \\ \hline c_1 & d_1 & e_1 & & \\ c_3 & d_2 & e_2 & & \\ \hline \end{array}$$

 $r_1 \bowtie r_2$ В Ε  $c_1$  $a_1$  $c_2$  $d_2$  $c_3$  $e_2$ 

Weitere Operationen: Semi-Verbund (Semi-Join)

Die Semi-Verbundoperatoren projizieren die Tupel der Ergebnisrelation eines Natural-Join auf das Schema einer der Ausgangsrelationen:

$$r_1 \bowtie r_2 = \pi_{\mathcal{R}_1}(r_1 \bowtie r_2)$$
 bzw.  $r_1 \bowtie r_2 = \pi_{\mathcal{R}_2}(r_1 \bowtie r_2)$ 

Weitere Operationen: Semi-Verbund (Semi-Join)

Die Semi-Verbundoperatoren projizieren die Tupel der Ergebnisrelation eines Natural-Join auf das Schema einer der Ausgangsrelationen:

$$r_1 \bowtie r_2 = \pi_{\mathcal{R}_1}(r_1 \bowtie r_2)$$
 bzw.  $r_1 \bowtie r_2 = \pi_{\mathcal{R}_2}(r_1 \bowtie r_2)$ 

 $\Box$  Semi-Join von  $r_1$  mit  $r_2$ :

 $\Box$  Semi-Join von  $r_2$  mit  $r_1$ :

Es gilt folgende Identität:  $r_1 \rtimes r_2 = r_2 \ltimes r_1$ 

Weitere Operationen: relationale Division

Die bisher betrachteten Anfragebeispiele liefern diejenigen Tupel, die eine bestimmte Selektionsbedingung erfüllen.

Frage: Wie bestimmt man diejenigen Tupel, die *alle* Bedingungen – im Sinne von *gleichzeitig* – einer Menge von Selektionsbedingungen erfüllen?

Weitere Operationen: relationale Division

Die bisher betrachteten Anfragebeispiele liefern diejenigen Tupel, die eine bestimmte Selektionsbedingung erfüllen.

Frage: Wie bestimmt man diejenigen Tupel, die *alle* Bedingungen – im Sinne von *gleichzeitig* – einer Menge von Selektionsbedingungen erfüllen?

#### Beispiel:

Buecher		
Titel Verlag		
Harry Potter	Princeton	
Heuristics	Addison	
Glücksformel	dpunkt	
Datenbanken	Springer	

Buchhaendler			
Name	Stadt	PLZ	
Lehmann	Berlin	99011	
Meiersche	Aachen	42100	
Amazon	Köln	52100	

Angebote			
Titel	Haendler		
Harry Potter	Lehmann		
Harry Potter	Meiersche		
Harry Potter	Amazon		
Datenbanken	Amazon		
Glücksformel	Amazon		
Glücksformel	Lehmann		

#### Anfragen

- 1. "Welche Titel sind bei allen Buchhändlern im Angebot?"
- Nicht zu verwechseln mit "Welche Titel befinden sich (alle) im Angebot?"

DB:V-61 Relational Algebra & Calculus © STEIN 2004-2018

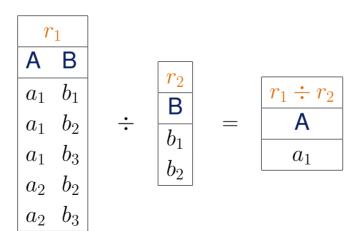
Weitere Operationen: relationale Division (Fortsetzung)

Syntax:  $r_1(\mathcal{R}_1) \div r_2(\mathcal{R}_2)$ Dividend Divisor

Semantik: Sei  $\mathcal{R}_2 \subseteq \mathcal{R}_1$ . Dann ist  $r_1 \div r_2$  definiert als:

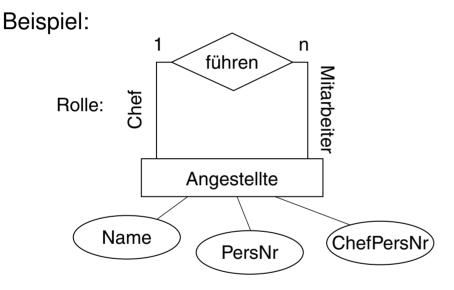
$$\{t \mid \forall t_2 \in r_2 \ \exists t_1 \in r_1 : t = t_1(\mathcal{R}_1 - \mathcal{R}_2) \land t_1(\mathcal{R}_2) = t_2\}$$

#### Beispiel:



#### **Rekursiver Abschluss**

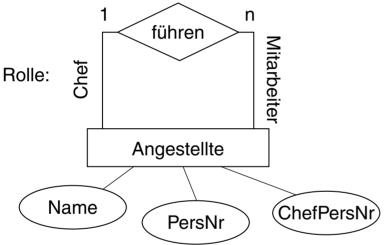
Der rekursive Abschluss kann mit Mitteln der relationalen Algebra nicht ausgedrückt werden.



#### **Rekursiver Abschluss**

Der rekursive Abschluss kann mit Mitteln der relationalen Algebra nicht ausgedrückt werden.





Angestellte			
Name	PersNr	ChefPersNr	
Franklin	333	888	
Smith	123	333	
Zelaja	999	987	
Ramesh	666	333	
Wallace	987	888	
Borg	888	$\perp$	
Jabbar	456	987	

#### Anfrage (rekursiv)

"Liefere alle direkten und indirekten Untergebenen von Borg."

#### Relationenalgebra

 $\sim$  TAFEL

#### Bemerkungen:

- □ Ansatz zur Auflösung der Rekursion in der Beispielanfrage:
  - 1. Ausgehend von 'Borg', Bestimmung der Untergebenen in jeder Stufe bzw. für soviel Stufen, für die angefragt ist.
  - 2. Vereinigung der Teilergebnisse aller Stufen.

DB:V-65 Relational Algebra & Calculus © STEIN 2004-2018

Operation	Argumente	Notation
SELECT	Relation $r$ , Auswahlbedingung $<$ COND $>$	$\sigma_{< {\sf COND}>}(r)$
PROJECT	Relation $r$ , Attributliste $\alpha$	$\pi_{lpha}(r)$
RENAME	Relation $r$ , Attributzuordnungen <a href="Mapping">Mapping</a> > Relation $r$ , Relationenname $s$	$ ho_{< ext{Mapping}>}(r) \  ho_s(r)$
UNION	Relationen $r_1, r_2$	$r_1 \cup r_2$
INTERSECTION	Relationen $r_1, r_2$	$r_1 \cap r_2$
DIFFERENCE	Relationen $r_1, r_2$	$r_1 - r_2$
CARTESIAN PRODUCT	Relationen $r_1, r_2$	$r_1 \times r_2$
NATURAL JOIN	Relationen $r_1, r_2$	$r_1 \bowtie r_2$
THETA JOIN	Relationen $r_1, r_2$ , Verbundbedingung $\theta$	$r_1 \bowtie_{\theta} r_2$
EQUI JOIN	Relationen $r_1, r_2,$ =-Verbundbedingung $\theta$	$r_1 \bowtie_{\theta} r_2$
OUTER JOIN (LEFT/RIGHT/FULL)	Relationen $r_1, r_2$	$r_1 \bowtie r_2, r_1 \bowtie r_2, r_1 \bowtie r_2$
SEMI JOIN (LEFT/RIGHT)	Relationen $r_1, r_2$	$r_1 \ltimes r_2, r_1 \rtimes r_2$
DIVISION	Relationen $r_1, r_2$	$r_1 \stackrel{\cdot}{\cdot} r_2$

Operation	Argumente	Notation
SELECT	Relation $r$ , Auswahlbedingung $<$ COND $>$	$\sigma_{< COND>}(r)$
PROJECT	Relation $r$ , Attributliste $\alpha$	$\pi_lpha(r)$
RENAME	Relation $r$ , Attributzuordnungen $<$ Mapping $>$ Relation $r$ , Relationenname $s$	$ ho_{< ext{Mapping}>}(r) \  ho_s(r)$
UNION	Relationen $r_1, r_2$	$r_1 \cup r_2$
INTERSECTION	Relationen $r_1, r_2$	$r_1 \cap r_2$
DIFFERENCE	Relationen $r_1, r_2$	$r_1 - r_2$
CARTESIAN PRODUCT	Relationen $r_1, r_2$	$r_1 \times r_2$
NATURAL JOIN	Relationen $r_1, r_2$	$r_1 \bowtie r_2$
THETA JOIN	Relationen $r_1, r_2$ , Verbundbedingung $\theta$	$r_1 \bowtie_{\theta} r_2$
EQUI JOIN	Relationen $r_1, r_2,$ =-Verbundbedingung $\theta$	$r_1 \bowtie_{\theta} r_2$
OUTER JOIN (LEFT/RIGHT/FULL)	Relationen $r_1, r_2$	$r_1 \bowtie r_2, r_1 \bowtie r_2, r_1 \bowtie r_2$
SEMI JOIN (LEFT/RIGHT)	Relationen $r_1, r_2$	$r_1 \ltimes r_2, r_1 \rtimes r_2$
DIVISION	Relationen $r_1, r_2$	$r_1 \stackrel{\cdot}{ locate} r_2$

Operation	Argumente	Notation
SELECT	Relation r, Auswahlbedingung <cond></cond>	$\sigma_{< COND>}(r)$
PROJECT	Relation $r$ , Attributliste $\alpha$	$\pi_{lpha}(r)$
RENAME	Relation $r$ , Attributzuordnungen <a href="Mapping">Mapping</a> > Relation $r$ , Relationenname $s$	$ ho_{< ext{Mapping}>}(r) \  ho_{s}(r)$
UNION	Relationen $r_1, r_2$	$r_1 \cup r_2$
INTERSECTION	Relationen $r_1, r_2$	$r_1 \cap r_2$
DIFFERENCE	Relationen $r_1, r_2$	$r_1 - r_2$
CARTESIAN PRODUCT	Relationen $r_1, r_2$	$r_1 \times r_2$
NATURAL JOIN	Relationen $r_1, r_2$	$r_1 \bowtie r_2$
THETA JOIN	Relationen $r_1, r_2$ , Verbundbedingung $\theta$	$r_1 \bowtie_{\theta} r_2$
EQUI JOIN	Relationen $r_1, r_2,$ =-Verbundbedingung $\theta$	$r_1 \bowtie_{\theta} r_2$
OUTER JOIN (LEFT/RIGHT/FULL)	Relationen $r_1, r_2$	$r_1 \bowtie r_2, r_1 \bowtie r_2, r_1 \bowtie r_2$
SEMI JOIN (LEFT/RIGHT)	Relationen $r_1, r_2$	$r_1 \ltimes r_2, r_1 \rtimes r_2$
DIVISION	Relationen $r_1, r_2$	$r_1 \stackrel{\circ}{ ext{-}} r_2$

Operation	Argumente	Notation
SELECT	Relation $r$ , Auswahlbedingung $<$ cond $>$	$\sigma_{< {\sf COND}>}(r)$
PROJECT	Relation $r$ , Attributliste $\alpha$	$\pi_{lpha}(r)$
RENAME	Relation $r$ , Attributzuordnungen < Mapping> Relation $r$ , Relationenname $s$	$ ho_{< ext{Mapping}>}(r) \  ho_s(r)$
UNION	Relationen $r_1, r_2$	$r_1 \cup r_2$
INTERSECTION	Relationen $r_1, r_2$	$r_1 \cap r_2$
DIFFERENCE	Relationen $r_1, r_2$	$r_1 - r_2$
CARTESIAN PRODUCT	Relationen $r_1, r_2$	$r_1 \times r_2$
NATURAL JOIN	Relationen $r_1, r_2$	$r_1 \bowtie r_2$
THETA JOIN	Relationen $r_1, r_2$ , Verbundbedingung $\theta$	$r_1 \bowtie_{\theta} r_2$
EQUI JOIN	Relationen $r_1, r_2,$ =-Verbundbedingung $\theta$	$r_1 \bowtie_{\theta} r_2$
OUTER JOIN (LEFT/RIGHT/FULL)	Relationen $r_1, r_2$	$r_1 \bowtie r_2, r_1 \bowtie r_2, r_1 \bowtie r_2$
SEMI JOIN (LEFT/RIGHT)	Relationen $r_1, r_2$	$r_1 \ltimes r_2, r_1 \rtimes r_2$
DIVISION	Relationen $r_1, r_2$	$r_1 \div r_2$