

# Kapitel 3

## Relationale Anfragen

## Bestandteile von DB-Sprachen

DB-Sprache =

- **DDL** (“Data Definition Language”)  
oder SDL (“Schema Definition Language”)
- und **DML** (“Data Manipulation Language”)

DML =

- **Änderungen (Updates)** – häufig einfach – *→ Kapitel 4*
- und **Anfragen (Queries)** – häufig komplex – *→ dieses Kapitel*

## Klassifikation von Anfragesprachen

- (a) **deskriptiv**: Die auszuwählenden Objekte werden als Gesamtheit beschrieben. (Was?)
- (b) **prozedural**: Das Ergebnis wird durch eine Folge von Operationen konstruiert. (Wie?)
  - (b1) Operationen auf Objektmengen, z. B. Relationen
  - (b2) Operationen auf einzelnen Objekten, z. B. Tupel/Records

### Beispiele:

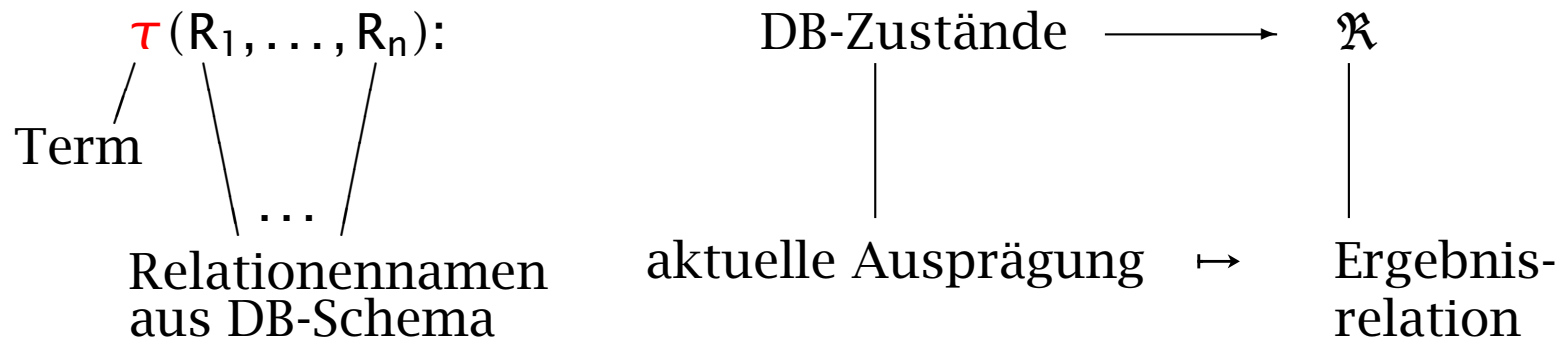
- zu (b1): **Relationenalgebra**
- zu (a): Relationenkalküle
- zu (a/b1): **SQL** (basiert auf Relationenalgebra und -kalkül)
- zu (b2): programmiersprachliche Schnittstellen

### 3.1 Relationenalgebra

= Menge  $\mathcal{R}$  aller endlichen Relationen (mit ihren Schemata) zusammen mit Operationen

$$\omega: \mathcal{R} [\times \dots \times \mathcal{R}] [\times \text{weitere Parameter}] \rightarrow \mathcal{R}$$

Aus diesen lassen sich Anfragen zusammensetzen:



z. B. “alle Ausleihen ?” (im jeweils aktuellen DB-Zustand):

$$(\text{AUSLEIHE\_ALT} - \text{RÜCKGABE\_HEUTE}) \cup \text{AUSLEIHE\_HEUTE}$$

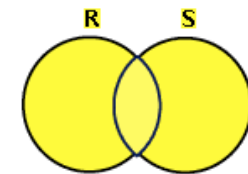
## Grundoperationen

Gegeben seien endliche Relationen  $R$  und  $S$  mit zugehörigen Schemata  $(A_1 : D_1, \dots, A_n : D_n)$  bzw.  $(B_1 : D'_1, \dots, B_m : D'_m)$ .

### 1. Vereinigung $R \cup S$

*Voraussetzung:*  $\text{Schema}(R) = \text{Schema}(S) = (A_1, \dots, A_n)$

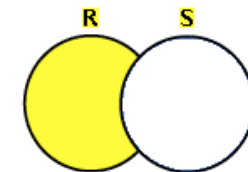
*Ergebnis:*            *Schema*     $(A_1, \dots, A_n)$   
                          *Relation*    $\{t \mid t \in R \vee t \in S\}$



### 2. Differenz $R - S$

*Voraussetzung und Ergebnis-Schema* wie oben

*Ergebnis:*             $\{t : (A_1, \dots, A_n) \mid t \in R \wedge \neg t \in S\}$



---

Die logischen Symbole bedeuten:  $\wedge$  “und”,  $\vee$  “oder”,  $\neg$  “nicht”,  
 $\Rightarrow$  “impliziert” bzw. “wenn...,dann...”,  $\Leftrightarrow$  “äquivalent” bzw. “genau dann...,wenn...”,  
 $\exists$  “es gibt”,  $\forall$  “für alle” .

## Grundoperationen (Forts.)

3. **(Einfache) Selektion**  $\sigma_{\varphi}(R)$  filtert Zeilen aus einer Tabelle:

*Voraussetzungen:*  $\varphi$  ist eine **“atomare Selektionsformel”**  $\theta(\alpha_1, \dots, \alpha_n)$ , d.h.:

$\alpha_1, \dots, \alpha_n$  sind **“Attributterme”** aus Datentyp-Konstanten, R-Attributnamen und Datentyp-Operationen, z.B.  $((A + 10) * 0.1)$

$\theta$  ist ein Datentyp-Prädikat, etwa ein Vergleichsoperator wie  $=, \neq, <$ .

*Ergebnis:* Relation  $\sigma_{\varphi}(R) := \{ r \mid (r \in R) \wedge r \text{ erfüllt } \varphi \}$   
mit unverändertem Schema

*Beispiel:*

R	A	B	C	D	$\sigma_{A=1}(R)$	A	B	C	D	$\sigma_{A>B}(R)$	A	B	C	D
	1	2	3	4		1	2	3	4		7	2	8	6
	1	5	3	6		1	5	3	6					
	7	2	8	6		1	9	8	10					
	1	9	8	10										

## Grundoperationen (Forts.)

4. **Projektion**  $\pi_{\bar{A}}(R)$  filtert Spalten aus einer Tabelle, ggfs. mit Umbenennung und tupelweisen Berechnungen:

*Voraussetzungen:*  $\bar{A} = (\alpha_1 C_1, \dots, \alpha_k C_k)$

mit Attributtermen  $\alpha_i$  und Aliasnamen  $C_i$

( $C_i$  kann entfallen, wenn  $\alpha_i$  ein Attributname  $A$  ist, d. h.  $C_i \equiv A$ .)

*Definition:* Zu einem Tupel  $r = (r_1, \dots, r_n)$  bezeichne  $\pi_{\alpha}(r)$  die Anwendung eines Attributterms  $\alpha$  auf  $r$ ,

z. B.  $\pi_{A_3}(r) = r_3$ ,  $\pi_{A_5+1}(r) = r_5 + 1$ .

*Ergebnis:*  $\pi_{\bar{A}}(R) := \{ t: (C_1, \dots, C_k) \mid \exists (r \in R) \ t = (\pi_{\alpha_1}(r), \dots, \pi_{\alpha_k}(r)) \}$   
mit Schema  $(C_1, \dots, C_k)$

*Beispiel:*

R	A	B	C	D	$\pi_{D, A, D*2, DD}(R)$	D	A	DD
	1	2	4	3		3	1	6
	1	5	6	3		8	7	16
	7	2	6	8		8	1	16
	1	9	10	8				

## Grundoperationen (Forts.): Beispielanfragen an die Bibliotheks-DB

(a) *Welche Bücher sind nach 2001 erschienen?*

$\sigma_{\text{Jahr} > 2001}$  (BUCH)

(b) Gegeben sei eine zusätzliche Relation

TAGUNGSBERICHT(DokNr, . . . , Tagungsjahr, Erscheinungsjahr).

*Welche Tagungsberichte sind bereits im Tagungsjahr erschienen?*

$\sigma_{\text{Tagungsjahr} = \text{Erscheinungsjahr}}$  (TAGUNGSBERICHT)

(c) *Namen und um 1 erhöhte Semesterzahlen aller Studenten?*

$\pi_{\text{SName}, \text{Semester}+1 \text{ Sem}}$  (STUDENT)



## Grundoperationen (Forts.)

### 5. (Kartesisches) Produkt $R \times S$

bildet alle Kombinationen von Tupeln aus R und S:

*Voraussetzungen:*  $\{A_1, \dots, A_n\} \cap \{B_1, \dots, B_m\} = \emptyset$   
(ggfs. Attribute vorher umbenennen, z. B. R.A und S.A)

*Definition:* Zu Tupeln  $r = (r_1, \dots, r_n)$  und  $s = (s_1, \dots, s_m)$  sei  
 $r \cdot s := (r_1, \dots, r_n, s_1, \dots, s_m)$  deren Konkatenation.

*Ergebnis:*  $R \times S := \{ t: (A_1, \dots, A_n, B_1, \dots, B_m) \mid \exists (r \in R, s \in S) t = r \cdot s \}$   
mit Schema  $(A_1, \dots, A_n, B_1, \dots, B_m)$

*Beispiel:*

R	A <sub>1</sub>	A <sub>2</sub>	S	B	$R \times S$	A <sub>1</sub>	A <sub>2</sub>	B
	1	2		3		1	2	3
	5	6		4		1	2	4
						5	6	3
						5	6	4

*Bsp.anfrage (d): Welche Studenten können welche Bücher ausleihen?*

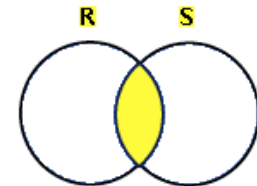
STUDENT  $\times$  BUCH

## Abgeleitete Operationen

### 6. **Durchschnitt** $R \cap S$

*Ergebnis:*  $R \cap S := \{t \mid t \in R \wedge t \in S\}$

*Ableitung aus Grundoperationen:*  $R \cap S = R - (R - S)$



## Abgeleitete Operationen (Forts.)

### 7. Verallgemeinerte Selektion $\sigma_{\varphi}(R)$

$\varphi$  ist eine “Selektionsformel”, d. h. eine Boolesche Verknüpfung von atomaren Selektionsformeln, also eine Zusammensetzung solcher Formeln mit den logischen Operatoren  $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$  und üblicher Klammersetzung.

*Ableitungen:*  $\sigma_{\varphi_1 \vee \varphi_2}(R) = \sigma_{\varphi_1}(R) \cup \sigma_{\varphi_2}(R)$

$\sigma_{\varphi_1 \wedge \varphi_2}(R) = \sigma_{\varphi_1}(R) \cap \sigma_{\varphi_2}(R) = \sigma_{\varphi_1}(\sigma_{\varphi_2}(R))$

$\sigma_{\neg \varphi}(R) = R - \sigma_{\varphi}(R)$  usw.

*Beispielanfragen:*

(e<sub>1</sub>) *Welche Informatik-Studenten sind noch nicht im 6. Semester?*

$\sigma_{\text{Fach}='Informatik' \wedge \text{Semester} < 6}(\text{STUDENT})$

(e<sub>2</sub>) *Alle Studenten, aber Philosophen erst ab 3. Semester?*

$\sigma_{\text{Fach}='Philosophie' \Rightarrow \text{Semester} > 2}(\text{STUDENT})$

## Abgeleitete Operationen (Forts.)

8. **Verbund (Join)**  $R \bowtie_{A_i \theta B_j} S$  oder  $R \bowtie_{A_i \theta B_j} S$  (“Equiverbund”, falls  $\theta \equiv =$ )

*Voraussetzungen:*  $\theta$  Vergleichsoperator,  $A_i/B_j$  Attribut(term) von  $R$  bzw.  $S$

*Definition:*  $R \bowtie_{A_i \theta B_j} S := \{ r \cdot s \mid (r \in R) \wedge (s \in S) \wedge (r.A_i \theta s.B_j) \}$

*Ableitung:*  $R \bowtie_{A_i \theta B_j} S = \sigma_{A_i \theta B_j} (R \times S)$  (ggfs. nach Umbenennung von Attributen)

*Beispiel:*

R	A	B	C	S	D	E	$R \bowtie_{B < D} S$	A	B	C	D	E
	1	2	3		3	1		1	2	3	3	1
	4	5	6		6	2		1	2	3	6	2
	7	8	9					4	5	6	6	2

## Abgeleitete Operationen (Forts.)

- *Notation:*

Um umbenennende Projektionen einzusparen, darf zu jedem Relationenoperand  $R$  ein **Alias** beliebigen Namens, z.B.  $X$ , vergeben werden. — Schreibweise:  $(R\ X)$

Auf zugehörige Attribute  $A_i$  kann man sich dann mit  $X.A_i$  beziehen:

*Ableitung:*  $(R\ X) = \pi_{A_1\ X.A_1, \dots, A_n\ X.A_n}(R)$

## Abgeleitete Operationen (Verbund, Forts.): Beispielanfragen

(f) *Autoren und bibliographische Angaben zu allen Büchern ?*

(AUTOREN A)  $\bowtie$  (BUCH B)  
 $A.DokNr = B.DokNr$

(g) Gegeben seien die Relationen

ANKUNFT(Zugnr, Zeit, Gleis) und ABFAHRT(Zugnr, Zeit, Gleis).

*Ermittle alle Umsteigemöglichkeiten*

*(am gleichen Tag mit mind. 5 Minuten Umsteigezeit):*

(ANKUNFT ANK)  $\bowtie$  (ABFAHRT ABF)  
 $ANK.Zeit + 5 \leq ABF.Zeit$

*Ermittle alle echten Umsteigemöglichkeiten:*

(ANKUNFT ANK)  $\bowtie$  (ABFAHRT ABF)  
 $ANK.Zeit + 5 \leq ABF.Zeit$   
 $\wedge ANK.Zugnr \neq ABF.Zugnr$

## Abgeleitete Operationen (Forts.)

### 9. Verallgemeinerter Verbund (Join) $R \bowtie_{\varphi} S$

Die Joinbedingung  $\varphi$  darf auch mehrere,  $\wedge$ -verknüpfte Vergleiche zwischen je einem Attribut(term) von R und S enthalten.

### 10. Natürlicher Verbund $R \bowtie S$ oder $R \bowtie_{\text{(namensgleiche Attribute)}} S$

*Definition:*  $R \bowtie S := \{ t \text{ mit den Attributen } \{A_1, \dots, A_n\} \cup \{B_1, \dots, B_m\} \mid \pi_{A_1, \dots, A_n}(t) \in R \wedge \pi_{B_1, \dots, B_m}(t) \in S \}$

*Ableitung:* Equiverbund über namensgleiche Attribute und anschl. Entfernung doppelter Attribute

*Beispiel:*

R	A	B	C	S	B	C	D	$R \bowtie S$	A	B	C	D	$= R \bowtie_{(B,C)} S$
	1	2	3		2	3	4		1	2	3	4	
	5	2	3		2	3	6		1	2	3	6	
	2	2	6		1	4	2		5	2	3	4	
	3	1	4						5	2	3	6	
									3	1	4	2	

*Bsp.anfr.:* Was liefert  $\text{ANKUNFT} \bowtie \text{ABFAHRT}$  ? – Alle Kurzstops von Zügen!

Im Spezialfall identischer Schemata stimmen natürlicher Verbund und Durchschnitt überein.

## Relationenalgebra als Anfragesprache

Relationale Operationen(namen) lassen sich zu **Termen**  $\tau$  über Relationen(namen) und konstanten Relationen zusammensetzen, z. B.

$$\tau = \pi_C (\sigma_{A=a} (R \bowtie S))$$

$\rightsquigarrow$  prozedurale Anfragesprache

*Beispielanfrage:*

(h) *Welche Titel haben die Bücher von Ullman?*

*in SQL:*

$$\pi_{(B.Titel \text{ Titel})} (\sigma_{A.AName='Ullman'} (\sigma_{B.DokNr=A.DokNr} (BUCH \times AUTOREN \ A)))$$

```
select B.Titel Titel
from   BUCH B, AUTOREN A
where  B.DokNr = A.DokNr
and    A.AName = 'Ullman'
```

$$= \pi_{Titel} (\sigma_{AName='Ullman'} (BUCH \bowtie AUTOREN))$$

```
select Titel
from   BUCH natural join AUTOREN
where  AName = 'Ullman'
```

$$= \pi_{Titel} (BUCH \bowtie \sigma_{AName='Ullman'} (AUTOREN))$$

eigentlich effizienteste Variante, da frühestmögliche Selektion;  
diese sollte aber der DBMS-Anfrageoptimierer selber finden



## Relationenalgebra als Anfragesprache (Forts.): weitere Beispiele

(i1) *Welche Studenten “lesen” welche Autoren?*

$$\pi_{AName, SName} ((AUTOREN \bowtie AUSLEIHE) \bowtie STUDENT)$$

(i2) *Welche Autoren liest der Student N.Neugierig?*

$$\pi_{AName} (\sigma_{SName='N.Neugierig'} ((AUTOREN \bowtie AUSLEIHE) \bowtie STUDENT))$$

(j) *Welche Bücher betreffen die Schlagworte “PR” und “DBS”?*

$$\begin{aligned} &BUCH \bowtie (\pi_{DokNr} (\sigma_{Schlagw='PR'} (DESKRIPTOREN))) \\ &\quad \cap \pi_{DokNr} (\sigma_{Schlagw='DBS'} (DESKRIPTOREN)) \end{aligned}$$

(k) *Welche Bücher wurden nicht ausgeliehen?*

$$BUCH - (\pi_{DokNr, Titel, Verlag, Ort, Jahr} (BUCH \bowtie AUSLEIHE))$$

*Konventionen:*

- Operationen: möglichst wenige verwenden, insbes. Joins statt Produkte
- Klammern: soweit nicht durch Op. ( $\sigma$ ,  $\pi$ ) erzwungen, nur nötig soweit Term eindeutig parsebar
- Äquivalente Alternativen: beliebig, (noch) ohne Rücksicht auf Laufzeitoptimierung

## Weitere abgeleitete Operationen der Relationenalgebra

11. **Semijoin**  $R \bowtie_{\varphi} S := \pi_{\text{Attribute}(R)} (R \Join_{\varphi} S)$

liefert alle R-Tupel, die mind. einen Join-Partner bzgl.  $\varphi$  in S haben  
(analog für den rechten Operanden definierbar)

(k1) *Welche Bücher wurden ausgeliehen?*

BUCH  $\bowtie_{\text{DokNr}=A.\text{DokNr}}$  (AUSLEIHE A)

12. **Anti[-Semi]join**  $R \bar{\bowtie}_{\varphi} S := R - (R \bowtie_{\varphi} S)$

$\rightsquigarrow$  alle R-Tupel, zu denen kein Join-Partner in S existiert (“R-Singles”)

(k2) *Welche Bücher wurden nicht ausgeliehen?*

BUCH  $\bar{\bowtie}_{\text{DokNr}=A.\text{DokNr}}$  (AUSLEIHE A)

*Ergebnis:*

BUCH	DokNr	Titel	Verlag	Ort	Jahr
	D8333	Betriebssysteme	...	...	...

## Weitere abgeleitete Operationen (Forts.)

$$13. \text{ Left Outer Join } R \bowtie_{\varphi} S := R \bowtie_{\varphi} S \cup (R \overline{\bowtie}_{\varphi} S) \times \{(\perp, \dots, \perp)\}$$

$\rightsquigarrow$  enthält ausser den Join-“Paaren” die “Singles” links (in R), jeweils passend(!) mit Nullwerten aufgefüllt

**Right Outer Join**  $\bowtie_{\varphi}$  analog

$$(\text{Full}) \text{ Outer Join } R \bowtie_{\varphi} S := R \bowtie_{\varphi} S \cup R \bowtie_{\varphi} S$$

(l) *Gesamtübersicht über alle Bücher, Ausleihen und Studenten?*

$$((\text{BUCH } B) \bowtie_{B.\text{DokNr}=A.\text{DokNr}} (\text{AUSLEIHE } A)) \bowtie_{A.\text{MatrNr}=S.\text{MatrNr}} (\text{STUDENT } S)$$

B.DokNr	B.Titel	...	A.DokNr	A.MatrNr	A.Datum	S.MatrNr	S.SName	...
B8501	Datenbanksysteme	...	B8501	3141593	11.10.2013	3141593	K. Klug	...
B8501	Datenbanksysteme	...	B8501	2718300	18.10.2014	2718300	N. Neugierig	...
...								
D8333	Betriebssysteme	...	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	1618034	W. Weise	...

## Gesetze der Relationen-Algebra: Verträglichkeit mit Selektionen

$R, R_1$  und  $R_2$  seien beliebige Relationen. In den folgenden Gleichungen seien auch die Selektionsformeln und Attributlisten beliebig, aber immer so gewählt, dass beide Seiten der Gleichung definiert sind. Attributreihenfolgen sind teilweise vernachlässigt.

$$(i) \sigma_{\varphi}(\pi_{\bar{A}}(R)) = \pi_{\bar{A}}(\sigma_{\varphi}(R))$$

$$(ii) \sigma_{\varphi}(R_1 \cup R_2) = \sigma_{\varphi}(R_1) \cup \sigma_{\varphi}(R_2)$$

$$(iii) \sigma_{\varphi}(R_1 - R_2) = \sigma_{\varphi}(R_1) - \sigma_{\varphi}(R_2)$$

$$(iv) \sigma_{\varphi}(R_1 \times R_2) = \sigma_{\varphi_0}(\sigma_{\varphi_1}(R_1) \times \sigma_{\varphi_2}(R_2))$$

falls  $\varphi \equiv \varphi_0 \wedge \varphi_1 \wedge \varphi_2$ ,  $\varphi_{1/2}$  nur mit  $R_{1/2}$ -Attributen

(iv)' wie (iv) für  $\bowtie$  statt  $\times$

$$(v) \sigma_{\varphi \wedge \psi}(R_1 \times R_2) = \sigma_{\varphi}(R_1 \bowtie_{\psi} R_2) \quad (\text{für eine Joinbedingung } \psi)$$

$$(vi) \sigma_{\varphi_1}(\sigma_{\varphi_2}(R)) = \sigma_{\varphi_1 \wedge \varphi_2}(R)$$

$$(vii) \sigma_{\varphi_1}(R) \cap \sigma_{\varphi_2}(R) = \sigma_{\varphi_1 \wedge \varphi_2}(R) \quad (\text{analog für } \cup, -)$$

$$(viii) \sigma_{\text{true}}(R) = R \quad \sigma_{\text{false}}(R) = \emptyset \quad \sigma_{\varphi}(\emptyset) = \emptyset$$

## Gesetze der Relationen-Algebra (Exemplarische Beweise)

zu (ii)  $\sigma_{\varphi}(R_1 \cup R_2) = \sigma_{\varphi}(R_1) \cup \sigma_{\varphi}(R_2)$ :

Die Aussage gilt, weil für beliebige Tupel  $t$  nach den Definitionen von  $\sigma/\cup$  gilt:

$$t \in \sigma_{\varphi}(R_1 \cup R_2)$$

$$\text{gdw. } t \in (R_1 \cup R_2) \wedge t \text{ erfüllt } \varphi$$

$$\text{gdw. } (t \in R_1 \vee t \in R_2) \wedge t \text{ erfüllt } \varphi$$

$$\text{gdw. } (t \in R_1 \wedge t \text{ erfüllt } \varphi) \vee (t \in R_2 \wedge t \text{ erfüllt } \varphi)$$

$$\text{gdw. } t \in \sigma_{\varphi}(R_1) \cup \sigma_{\varphi}(R_2)$$

zu (iv)  $\sigma_{\varphi}(R_1 \times R_2) = \sigma_{\varphi_0}(\sigma_{\varphi_1}(R_1) \times \sigma_{\varphi_2}(R_2))$ : ( $\varphi \equiv \varphi_0 \wedge \varphi_1 \wedge \varphi_2$  wie oben)

$$t \in \sigma_{\varphi}(R_1 \times R_2)$$

$$\text{gdw. } t \in (R_1 \times R_2) \wedge t \text{ erfüllt } \varphi$$

$$\text{gdw. } \exists(r \in R_1, s \in R_2) (t = r \cdot s \wedge t \text{ erfüllt } \varphi)$$

$$\text{gdw. } \exists(r \in R_1, s \in R_2) (t = r \cdot s \wedge r \text{ erfüllt } \varphi_1 \wedge s \text{ erfüllt } \varphi_2 \wedge t \text{ erfüllt } \varphi_0)$$

$$\text{gdw. } \exists(r \in \sigma_{\varphi_1}(R_1), s \in \sigma_{\varphi_2}(R_2)) (t = r \cdot s \wedge t \text{ erfüllt } \varphi_0)$$

$$\text{gdw. } t \in \sigma_{\varphi_0}(\sigma_{\varphi_1}(R_1) \times \sigma_{\varphi_2}(R_2))$$

## Gesetze der Relationen-Algebra: Verträglichkeit mit Projektionen

(i)  $\pi_{\bar{A}}(R) = R$  falls  $\bar{A} = \{\text{Attribute von } R\}$

(ii)  $\pi_{\bar{A}}(\pi_{\bar{B}}(R)) = \pi_{\bar{A}}(R)$  falls  $\bar{A} \subseteq \bar{B}$

(iii)  $\pi_{\bar{A}}(\sigma_{\varphi}(R)) = \pi_{\bar{A}}(\sigma_{\varphi}(\pi_{\bar{A} \cup \bar{F}}(R)))$  wobei  $\bar{F} = \{\text{Attribute in } \varphi\}$

(iv)  $\pi_{\bar{A}}(R_1 \cup R_2) = \pi_{\bar{A}}(R_1) \cup \pi_{\bar{A}}(R_2)$  (*gilt nicht für  $\times$  !*)

(v)  $\pi_{\bar{A}}(R_1 \times R_2) = \pi_{\bar{A}_1}(R_1) \times \pi_{\bar{A}_2}(R_2)$  wobei  $\bar{A}_i = \bar{A} \cap \{\text{Attribute von } R_i\} \neq \emptyset$

(vi)  $\pi_{\bar{A}}(R_1 \times R_2) = \pi_{\bar{A}_1}(R_1)$  falls  $\bar{A} = \bar{A}_1$  und  $R_2 \neq \emptyset$  (!)

(vii)  $\pi_{\bar{A}}(R_1 \bowtie_{\psi} R_2) = \pi_{\bar{A}}(\pi_{\bar{A}_1}(R_1) \bowtie_{\psi} \pi_{\bar{A}_2}(R_2))$

wobei  $\bar{A}_i = (\bar{A} \cup \{\text{Attribute in } \psi\}) \cap \{\text{Attribute von } R_i\}$

(viii)  $\pi_{\bar{A}}(R_1 \bowtie R_2) = \pi_{\bar{A}}(R_1 \bowtie R_2)$  falls  $\bar{A} \subseteq \{\text{Attribute von } R_1\}$

## Weitere Gesetze der Relationen-Algebra:

Einige Kommutativ-, Assoziativ- u.a. Regeln für binäre Operationen

(i) *vgl. Mengenlehre:*

$$R_1 \cup R_2 = R_2 \cup R_1$$

$$(R_1 \cup R_2) \cup R_3 = R_1 \cup (R_2 \cup R_3) = (R_1 \cup R_3) \cup R_2$$

$$R \cup R = R, \quad R \cup \emptyset = R$$

$$(ii) (R_1 \times R_2) \times R_3 = R_1 \times (R_2 \times R_3), \quad R \times \emptyset = \emptyset$$

Seien  $\bar{A}_i := \{\text{Attribute von } R_i\}$ ,  $A_i \in \bar{A}_i$ ,  $B_i \in \bar{A}_i$ .

$$(iii) R_1 \times R_2 = \pi_{\bar{A}_1, \bar{A}_2}(R_2 \times R_1), \quad (R_1 \times R_2) \times R_3 = \pi_{\bar{A}_1, \bar{A}_2, \bar{A}_3}((R_1 \times R_3) \times R_2)$$

(Die Projektionen korrigieren nur die Reihenfolge der Attribute!)

$$(ii)' R_1 \bowtie_{A_1=B_2} (R_2 \bowtie_{A_2=A_3} R_3) = (R_1 \bowtie_{A_1=B_2} R_2) \bowtie_{A_2=A_3} R_3$$

$$(iii)' R_1 \bowtie_{A_1=B_3} (R_2 \bowtie_{A_2=A_3} R_3) = \pi_{\bar{A}_1, \bar{A}_2, \bar{A}_3}((R_1 \bowtie_{A_1=B_3} R_3) \bowtie_{A_3=A_2} R_2)$$

$$\begin{aligned} \text{z. B. } \pi_{A_1}(\sigma_{A_1=A_3}(R_1 \times (R_2 \times R_3))) &= \pi_{A_1}(\sigma_{A_1=A_3}(R_1 \times R_3) \times R_2) \\ &= \pi_{A_1}((R_1 \bowtie_{A_1=A_3} R_3) \times R_2) \end{aligned}$$

## Regeln für das Rechnen mit Nullwerten:

*in Attributtermen und atomaren Selektionsformeln:*

Operationen und Vergleiche mit Nullwert( $\perp$ )-Argument liefern  $\perp$ ;  
z. B.  $\perp + 3$ ,  $\perp < 3$ , und auch  $\perp = 3$ ,  $\perp \neq 3$ .

(Deshalb braucht man spezielle Abfragen auf Nullwerte (Attribut[term] “is null”), die true/false liefern.)

*Sel.formeln:* Logische Verknüpfungen werden dreiwertig interpretiert:

$\wedge$	true	$\perp$	false	$\vee$	true	$\perp$	false	$\neg$	
true	true	$\perp$	false	true	true	true	true	true	false
$\perp$	$\perp$	$\perp$	false	$\perp$	true	$\perp$	$\perp$	$\perp$	$\perp$
false	false	false	false	false	true	$\perp$	false	false	true

*Selektionsergebnisse:* Ergibt sich bei der Auswertung einer Selektionsformel  $\varphi$  für ein Tupel insgesamt der “Wahrheitswert”  $\perp$ , so wird das Tupel nicht ausgegeben<sup>1</sup>.

---

<sup>1</sup>Vorgriff auf SQL: **where**  $\perp$  entspricht **where false**